# Constrained Programming and Applied Optimization (EEN025)

## Hand in Assignment - 3

Abdul Ahad Saeed

ahabdul@student.chalmers.se

Shreyas Mahadeva

shrmah@student.chalmers.se

October 29, 2023

# Task 1

The Tower of Hanoi is made up of three towers and a number of disks of various sizes that may be put onto any of the towers. The game's ultimate goal is to slide the full stack of disks onto another tower. The task is to find the number of steps required to transfer the disk stack from one tower to another. The decision variables used are.

- On(ds, tws, ts)
- Obj(ds, ts)
- From(tws, ts)
- To(tws, ts)

constraints required to encode the ToH:

1. At time t, only the smallest disk on a specific tower can be moved, Obj = false if a smaller disk is present on the same tower.
2. A disk cannot be put onto a tower that already holds a smaller disk. Hence towers are checked and if present Obj = false, to = false.
3. If a disk is to be relocated to another tower then From = true.
4. At time t for a tower To = true.
5. One disk can be moved at time t then Obj = true.
6. At time t+1, disk haven't moved stays in the same place.
7. It is not permitted to move a disk from a tower to same tower.
8. If disk is moved from one tower to another then On = true for the disk in the later tower at time t+1.
9. Ensures disks on first tower at time t moves to last tower at time t+n in n time steps.

Theoretical value for number of time steps required to move disks in a 3 tower configuration is given by.

$$t_s = 2^n - 1$$

| Disks | Time steps |
|---|---|
| 2 | 3 |
| 3 | 7 |
| 4 | 15 |
| 5 | 31 |
| 6 | 63 |
| 7 | 127 |

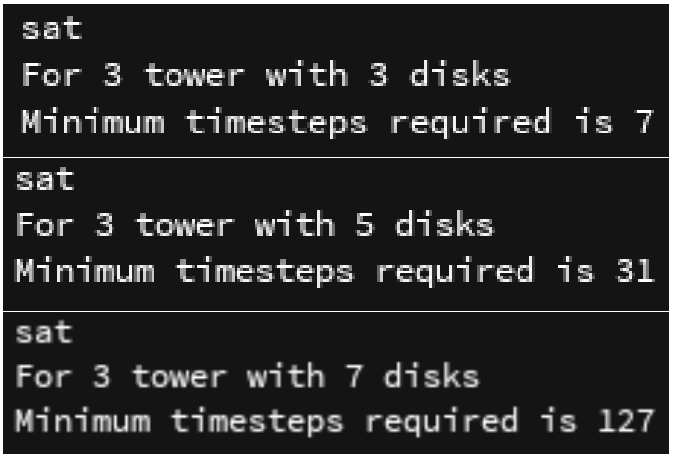


*Figure 1: TOH of 3 towers with different number of disks*

To convert it to optimisation problem we check minimum number of time-steps needed in a 'while' loop for given number of disks and towers, starting time-step from 1. The loop breaks when solution is found for certain value of time-steps.

# Task 2

A robot crane moving bricks on the platform has been taken into consideration. Following are the cases that have been taken into consideration:

## 2.1 Carrying out operation on same platform

In this scenario, robot is moving the bricks on the same platform that has been defined by the user. On considering the encoding from the previous task and taking into consideration the constraints that will be used in the scenario, variables are going to be defined in a similar way as has been done in the former task:

- On(b, pos, t)
- From(pos, t)
- To(pos, t)
- Obj(b, t)

Here b is brick, pos is position on the platform and t is the current time-step. The constraints required for this problem are defined as follows.

- $\boldsymbol{Unique - Starting - point}$: Variable $\boldsymbol{From}$ is going to be true only for the position from where the brick is going to be moved at a given time step.

$$Obj(b,t) \vee On(b,pos,t) \Rightarrow \neg From(1,t) \wedge \cdots \wedge From(pos,t) \wedge \cdots \wedge \neg From(npos,t) \\ (pos \in [1,npos], b \in [1,nb], t \in [1,ts]) \tag{1}$$

- $\boldsymbol{Unique - Destination}$: Variable $\boldsymbol{To}$ is going to be true only for the position to where the brick is going to be moved at a particular time step.

$$To(pos,t) \leftrightarrow \neg To(1,t) \wedge \cdots \wedge \neg To(pos-1,t) \wedge \neg To(pos+1,t) \wedge \cdots \wedge \neg To(npos,t) \\ (pos \in [1,npos], t \in [1,ts]) \tag{2}$$

- $\boldsymbol{Unique - Object}$: Only one brick $\boldsymbol{b}$ can be moved at a given time step.

$$Obj(b,t) \leftrightarrow \neg Obj(1,t) \wedge \cdots \wedge \neg Obj(b-1,t) \wedge \neg Obj(b+1,t) \wedge \cdots \wedge \neg Obj(nb,t) \\ (b \in [1,nb], t \in [1,ts]) \tag{3}$$

- $\boldsymbol{Non - moving - objects}$: Non-moving bricks will be in the same position in the next time step.

$$\neg Obj(b,t) \wedge On(b,pos,t) \Rightarrow On(b,pos,t+1) \wedge \neg On(b,1,t+1) \wedge \cdots \\ \wedge \neg On(b,pos-1,t+1) \wedge \neg On(b,pos+1,t+1) \wedge \cdots \wedge \neg On(b,npos,t+1) \\ (pos \in [1,npos], b \in [1,nb], t \in [1,ts]) \tag{4}$$

- ***Update − Process***: Once the brick has been moved to some other position, it will be on the same position in next step.

$$\begin{gathered}\neg Obj(b,t) \wedge From(pos,t) \wedge To(pos',t) \Rightarrow On(b,pos',t+1) \wedge \neg On(b,1,t+1) \\ \wedge \cdots \wedge \neg On(b,pos-1,t+1) \\ \wedge \neg On(b,pos+1,t+1) \wedge \cdots \wedge \neg On(b,npos,t+1) \\ (pos \in [1,npos], b \in [1,nb], t \in [1,ts])\end{gathered} \tag{5}$$

- ***Starting − point − and − destination − cannot − be − the − same*** Brick cannot be moved to the same position i.e. initial and final position should be different for a particular time-step.

$$\begin{gathered}From(pos,t) \Rightarrow \neg To(pos,t) \\ (pos \in [1,npos], t \in [1,ts])\end{gathered} \tag{6}$$

- ***Initial − Final − State*** Bricks are at initial positions at time-step = 1 but have reached their goal at the final time-step where initial and final positions are being defined by the user.

$$\begin{gathered}On(b,initial-pos,1) \wedge On(b,final-pos,ts) \\ (b \in [1,nb])\end{gathered} \tag{7}$$

- ***One − to − One − brick − position − mapping*** Two or more bricks cannot occupy the same position at a particular time step.

$$\begin{gathered}On(b,pos,t) \Rightarrow \neg On(1,pos,t) \wedge \cdots \wedge \neg On(b-1,pos,t) \\ \wedge \neg On(b+1,pos,t) \wedge \cdots \wedge \neg On(nb,pos,t) \\ (pos \in [1,npos], b \in [1,nb], t \in [1,ts])\end{gathered} \tag{8}$$

- ***Starting − position*** In the beginning, each brick is at its set initial position and cannot be found at any other location.

$$\begin{gathered}On(b,initial-pos,1) \wedge \neg On(b,1,1) \wedge \cdots \wedge \neg On(b,initial-pos-1,1) \\ \wedge \neg On(b,initial+pos,1) \wedge \cdots \wedge \neg On(b,npos,1) \\ (b \in [1,nb])\end{gathered} \tag{9}$$

The result is obtained as the number of steps required to reach the goal position while taking into consideration the above constraints.

## 2.2 Carrying out operation on same platform but bricks being categorized

In this section bricks are defined as categories. As compared to the last section the change is made in $\boldsymbol{Initial - Final - State}$ constraint. The final position can be at any place in the goal position of a particular category of bricks and the index of each brick remains the same. This results in the following equation:

$$\begin{gathered} On(b, initial - pos, 1) \wedge On(b, goal - pos_1, ts) \vee On(b, goal - pos_2, ts) \vee ... \\ On(b, goal - pos_{last}, ts) \\ (b \in [1, nb]) \end{gathered} \tag{10}$$

where $inital - pos$ and $goal - pos_i$ are the goal position in that particular category (to which the brick belongs) respectively. The results for this case are given in the figure below:

## 2.3 Moving bricks with multiple robots and platforms

This is the extended case of what we have done so far. Here we have to move bricks over multiple platforms with a shared platform between each pair of adjacent platforms. Considering this setup, a robot at a particular platform is supposed to put the brick on a shared platform and a robot from the other platform (which has a shared platform with the former robot) is supposed to pick it up from the shared platform and place it on its goal position if the goal position lies on the platform of this robot, in other case it will place it on the other shared platform and the process goes on unless the brick reaches its destination (goal position).

Now the total number of platforms (let it be nP) will include both the platforms on which the robots are working (let it be RP) and the shared platforms (let it be SP). So now we are going to have a platform ID in our variables (let it be $P_i$) and a robot at each platform (RP) can work independently at a given timestamp.

Considering platforms as a part of the scenario the modified variables are given as:

- On(b, P, pos, t)
- From(P, pos, t)

- To(P, pos, t)
- Obj(P',b, t)

Here b, pos and t are the same as in the previous scenario. P is the platform where we are considering the position pos and P' is the platform whose robot will pick up the brick at timestamp t.

Now consider 'N' as the number of RP (robots with platforms) and 'M' as the number of SP (shared platforms). A list of constraints which have been slightly modified due to the addition of platforms (multiple robots and shared platforms) or which are the same as in the previous case are given below:

- $\boldsymbol{Unique - Starting - point}$: Variable $\boldsymbol{From}$ is going to be true only for the position from where the brick is going to be moved at a given time step.

$$\begin{gathered} Obj(P_i, b, t) \vee On(b, P_i, pos, t) \Rightarrow \neg From(P_i, 1, t) \wedge \cdots \\ \wedge From(P_i, pos, t) \wedge \cdots \wedge \neg From(P_i, npos, t) \\ (pos \in [1, npos], b \in [1, nb], t \in [1, ts], P_i \in [RP_1, \cdots, RP_N, SP_1, \cdots, SP_M]) \end{gathered} \tag{11}$$

  The constraint will be defined for each platform (therefore we have used $P_i$).

- $\boldsymbol{Unique - Destination}$: Variable $\boldsymbol{To}$ is going to be true only for the position to where the brick is going to be moved at a particular time step.

$$\begin{gathered} To(P_i, pos, t) \leftrightarrow \neg To(P_i, 1, t) \wedge \cdots \\ \wedge \neg To(P_i, pos - 1, t) \wedge \neg To(P_i, pos + 1, t) \wedge \cdots \wedge \neg To(P_i, npos, t) \\ (pos \in [1, npos], t \in [1, ts], P_i \in [RP_1, \cdots, RP_N, SP_1, \cdots, SP_M]) \end{gathered} \tag{12}$$

- $\boldsymbol{Unique - Object}$: Only one brick $\boldsymbol{b}$ can be moved at a given time step.

$$\begin{gathered} Obj(P_i, b, t) \leftrightarrow \neg Obj(P_i, 1, t) \wedge \cdots \\ \wedge \neg Obj(P_i, b - 1, t) \wedge \neg Obj(P_i, b + 1, t) \wedge \cdots \wedge \neg Obj(P_i, nb, t) \\ (b \in [1, nb], t \in [1, ts], P_i \in [RP_1, \cdots, RP_N, SP_1, \cdots, SP_M]) \end{gathered} \tag{13}$$

- ***Non − moving − objects***: Non-moving bricks will be in the same position in the next time step.

$$\neg Obj(P_i, b, t) \wedge On(b, P_i, pos, t) \Rightarrow On(b, P_i, pos, t+1) \wedge \neg On(b, P_i, 1, t+1) \wedge \cdots$$
$$\wedge \neg On(b, P_i, pos-1, t+1) \wedge \neg On(b, P_i, pos+1, t+1) \wedge \cdots \wedge \neg On(b, P_i, npos, t+1)$$
$$(pos \in [1, npos], b \in [1, nb], t \in [1, ts], P_i \in [RP_1, \cdots, RP_N, SP_1, \cdots, SP_M]) \tag{14}$$

- ***Update − Process***: Once the brick has been moved to some other position, it will be on the same position in next step.

$$\neg Obj(P_i, b, t) \wedge From(P_i, pos, t) \wedge To(P_i, pos', t) \Rightarrow On(b, P_i, pos', t+1)$$
$$\wedge \neg On(b, P_i, 1, t+1) \wedge \cdots \wedge \neg On(b, P_i, pos-1, t+1)$$
$$\wedge \neg On(b, P_i, pos+1, t+1) \wedge \cdots \wedge \neg On(b, P_i, npos, t+1)$$
$$(pos \in [1, npos], b \in [1, nb], t \in [1, ts], P_i \in [RP_1, \cdots, RP_N, SP_1, \cdots, SP_M]) \tag{15}$$

- ***Starting − point − and − destination − cannot − be − the − same*** Brick cannot be moved to the same position i.e. initial and final position should be different for a particular time-step.

$$From(P_i, pos, t) \Rightarrow \neg To(P_i, pos, t)$$
$$(pos \in [1, npos], t \in [1, ts], P_i \in [RP_1, \cdots, RP_N, SP_1, \cdots, SP_M]) \tag{16}$$

- ***Starting − position*** In the beginning, each brick is at its set initial position and cannot be found at any other location.

$$On(b, P_i, initial - pos, 1) \wedge \neg On(b, P_i, 1, 1) \wedge \cdots \wedge \neg On(b, P_i, initial - pos - 1, 1)$$
$$\wedge \neg On(b, P_i, initial + pos, 1) \wedge \cdots \wedge \neg On(b, P_i, npos, 1)$$
$$(b \in [1, nb], P_i \in [RP_1, \cdots, RP_N, SP_1, \cdots, SP_M]) \tag{17}$$

- ***One − to − One − brick − position − mapping*** Two or more bricks cannot occupy the same position at a particular time step.

$$On(b, P_i, pos, t) \Rightarrow \neg On(1, P_i, pos, t) \wedge \cdots \wedge \neg On(b-1, P_i, pos, t)$$
$$\wedge \neg On(b+1, P_i, pos, t) \wedge \cdots \wedge \neg On(nb, P_i, pos, t)$$
$$(pos \in [1, npos], b \in [1, nb], t \in [1, ts], P_i \in [RP_1, \cdots, RP_N, SP_1, \cdots, SP_M]) \tag{18}$$

An important change will be made in the "Initial-Final-State" constraint. The final state of the brick, which will be transferred to another platform, will be set to shared platform $SP_i$ between those two adjacent platforms at some time t (where t > 1). Moreover, the initial position of a brick b (from the previous platform) at time t will be at a shared platform $SP_i$ and the final position will be set to goal platform $RP_j$ at time t' where t' > t.

## 2.4 Setting distance as cost function

Now we have to consider the distance as a cost function rather than the number of moves. By introducing an array d, having the same length as the number of robots, we can keep track of the distance moved by each robot. Consider a brick b moved from ($x_1$,$y_1$) to ($x_2$,$y_2$) in an $n \times n$ grid at time instant $t_i$ by a robot $r_j$. Considering both Manhattan and Cartesian distance between two points, the element $d(r_j, t_i)$ for movement of robot $r_j$ is given as:

$$d(r_j, t_i) = d(r_j, t_i) + |x_2 - x_1| + |y_2 - y_1| + \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{19}$$

This will be the new cost function.