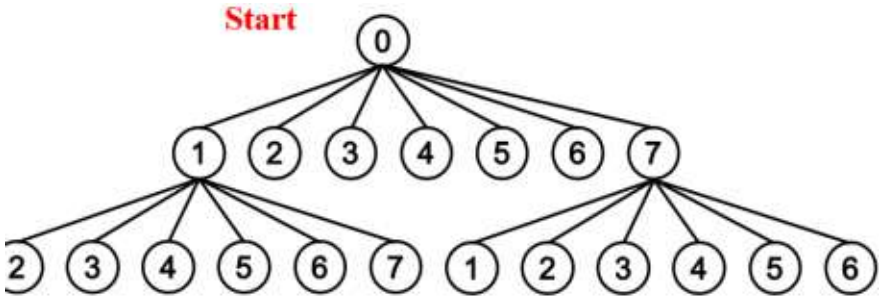
Optimization Methods
Lab 6 Session

By Shi Jimao

Task1

```
5.3825424421026655 7.295309061734091 8.681915936465941 7.0931712199461305 4.538533624653502 6.477851503836757
8.477851503836757 7.295309061734091 10.477851503836757 8.538533624653503 5.0931712199461305 6.681915936465942
10.477851503836757 7.3825424421026655 12.390618123468183 10.386342439892262 6.538533624653502 7.295309061734091
8.0644583785686O8 6.765084884205331 8.590618123468182 7.312975122664688 5.303146344170309 7.295309061734091
7.723213908544128 6.073267251517407 7.833842686387594 8.56047536916002 4.538533624653502 6.55064659066564
9.016385128490O6 6.475713662048797 7.833842686387594 10.3984554059044 8.386342439892262 6.40828418439878
10.955703007673513 8.064458378568608 8.590618123468182 12.390618123468183 10.3984554059044 6.40828418439878
(2 4)    (3 1)    (3 4)    (3 5)    (6 4)    (7 1)    (7 4)    (7 5)
```

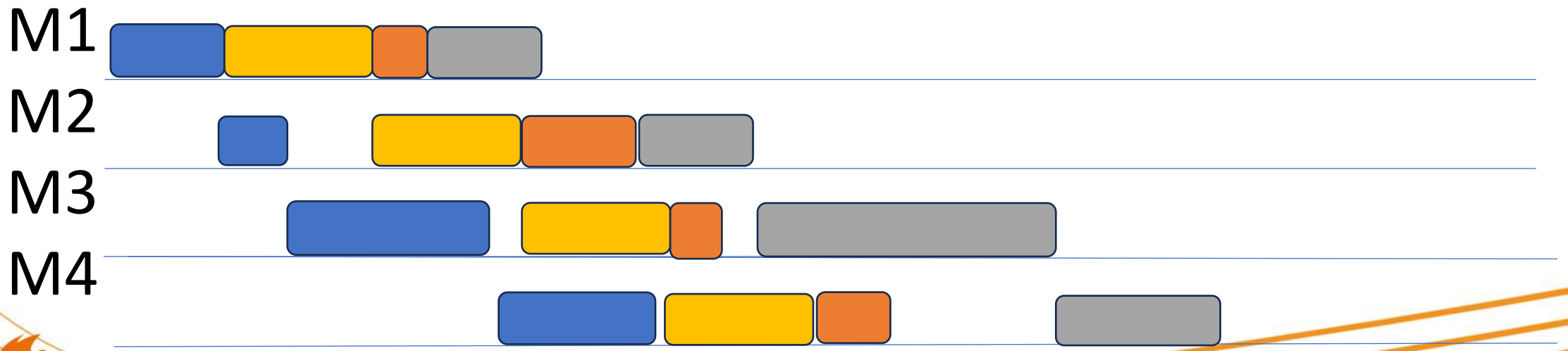


In depth 2 nodes, 8 nodes will be terminated using the greedy solution of tour length 10.

Task2

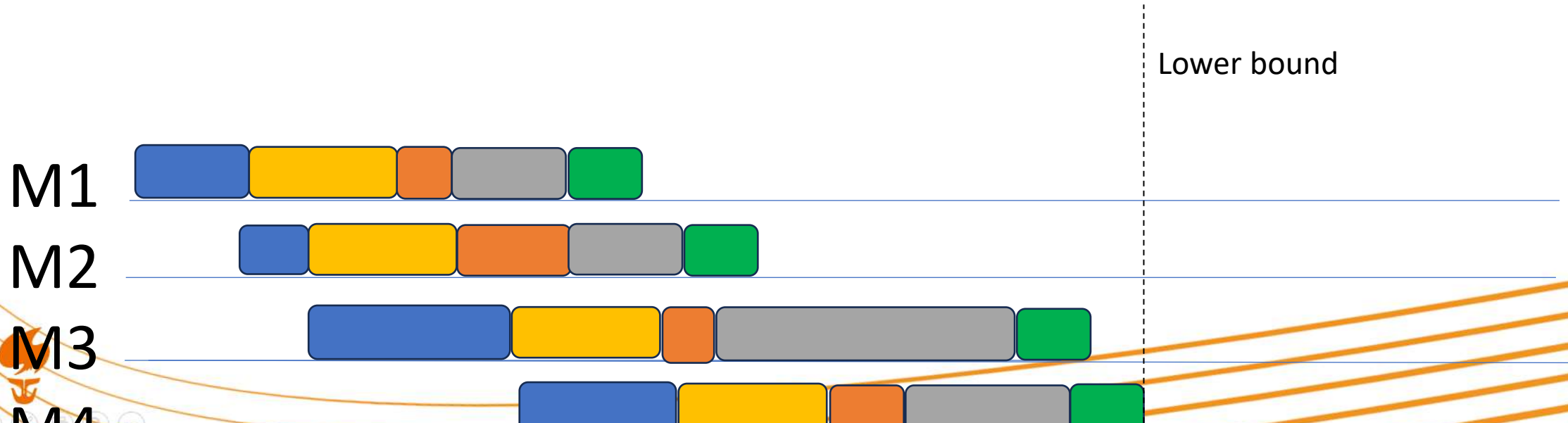
I will describe my opinion within an example.

Here are 4 jobs and 4 machines, and I select  as the first job. The schedule is shown in followings:



Task2

As for lower bound, I compress the wait time for all tasks except the first for all machines. And then add a period which is the shortest time in the remains jobs of the last machine( ). Then select the maximum total time from all machines.



Task3

I use a code to estimate the time of m-machine-n-jobs flow task. I set a series of n and record their running time and shown in the following picture.

For n is from 200000 to 400000, it needs about 10s. Therefore, about one hour, it can solve the problem with n = 20000000