LOADING BALANCE PROBLEM

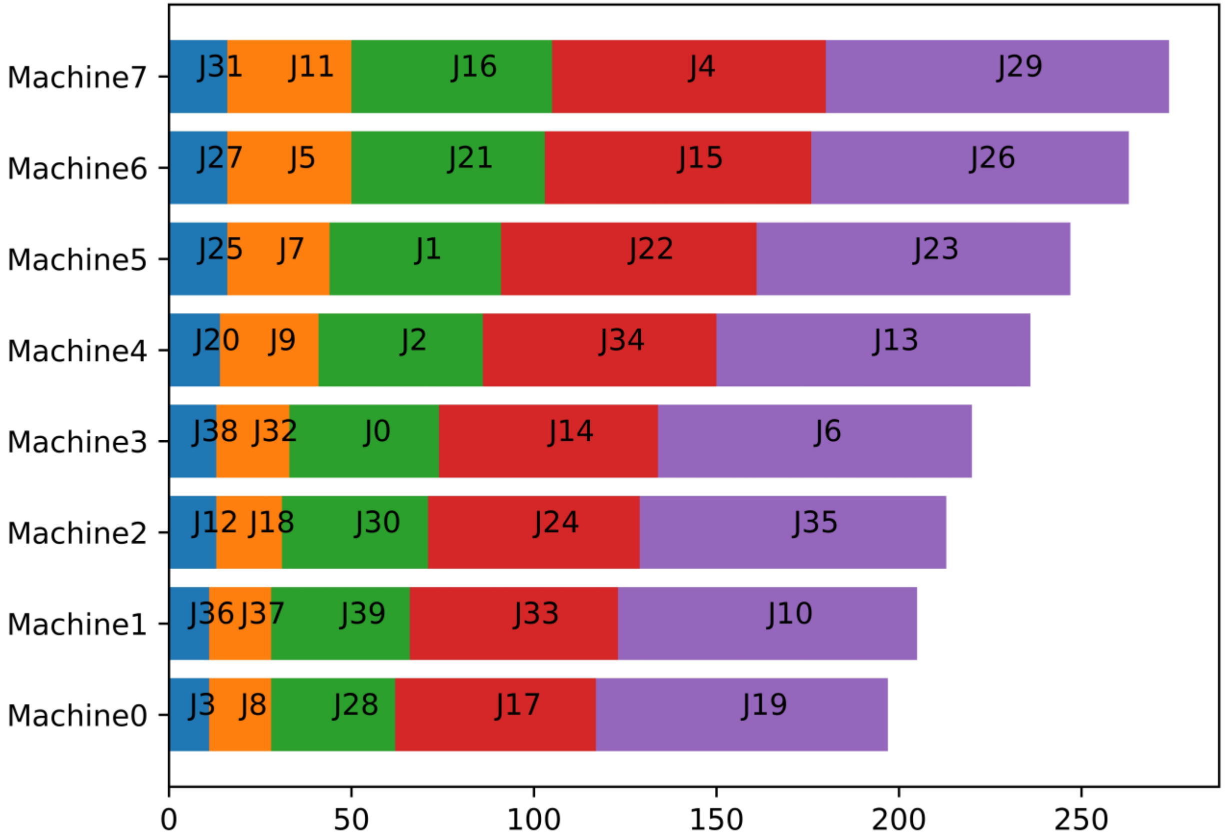
Greedy loading balance strategy

Zhiyuan Wang
12032878

Southern University of Science and Technology

LOADING BALANCE PROBLEM

This problem's target is assign jobs to computing machines to minimizes the total time consumption.



GREEDY LOADING BALANCE STRATEGY

This strategy will assign a job to the machine with the smallest load in an arbitrary order of jobs.

Algorithm 2 Loading Balance Problem

Require: m : number of machines, job_times : The time of each jobs
Ensure:

function $greedy\_workload\_balance(m, job\_time)$
    $M \leftarrow []$
    for i=1 to m **do**
        $L_i \leftarrow 0$
        $M(i) \leftarrow []$
    end for
    for j=1 to n **do**
        $i \leftarrow argmin_k L_k$
        $M(i) \leftarrow M(i).append(j)$
        $L_i \leftarrow \mathrm{L}_i + t_j$
    end for
    returnM
end function

EFFECT OF ALGORITHM

If the theoretical optimal makespan of the job queue is T^*, then the greedy algorithm's makespan T will not worse than $2T^*$.

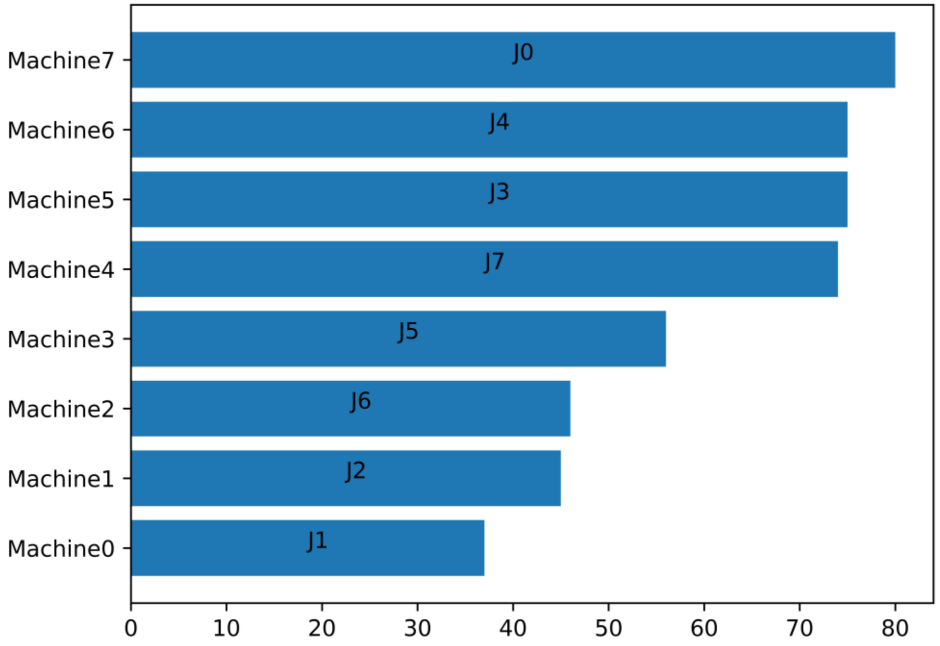
$$\boldsymbol{T^* \leq T \leq 2T^*}$$

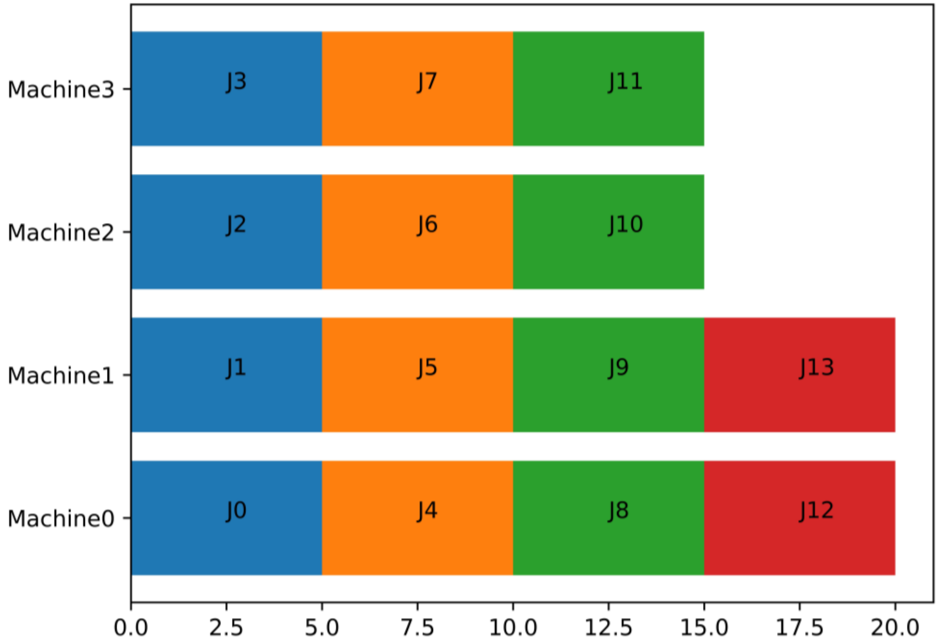
THE FIRST EXAMPLE

You want a example that the makespan T equal to the T^*? It's so easy!

What if the number of the jobs equal to the jobs of machine?

What if that each jobs need the time?





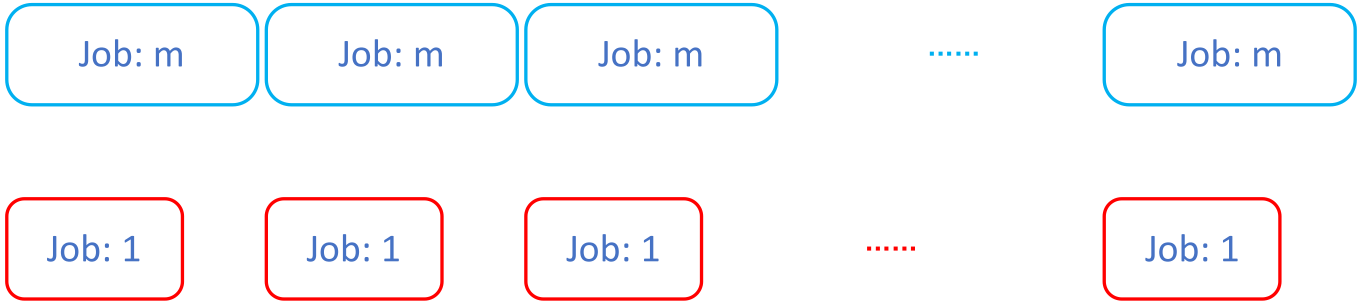
SUSTech Southern University of Science and Technology

THE SECOND AND THIRD EXAMPLE

These two question can be answered by one case.

Let’s suppose that there are m machines, and the minimum unit of time is 1.

What if we have m jobs with time m, and m jobs with time 1?



Let's consider two scenarios

1. Let the jobs queue be: [m, m, m, ……, m, 1, 1, ……, 1, 1]
2. Let the jobs queue be: [m, 1, m, 1, m, 1, ……, m, 1, m, 1]

THE SECOND AND THIRD EXAMPLE

These two question can be answered by one case.

Let’s suppose that there are m machines, and the minimum unit of time is 1.

What if we have m jobs with time m, and m jobs with time 1?

Let's consider two scenarios

1. Let the jobs queue be: [m, m, m, ……, 1, 1, ……, 1, 1, m]

2. Let the jobs queue be: [m, 1, m, 1, m, 1, ……, m, 1, m, 1]

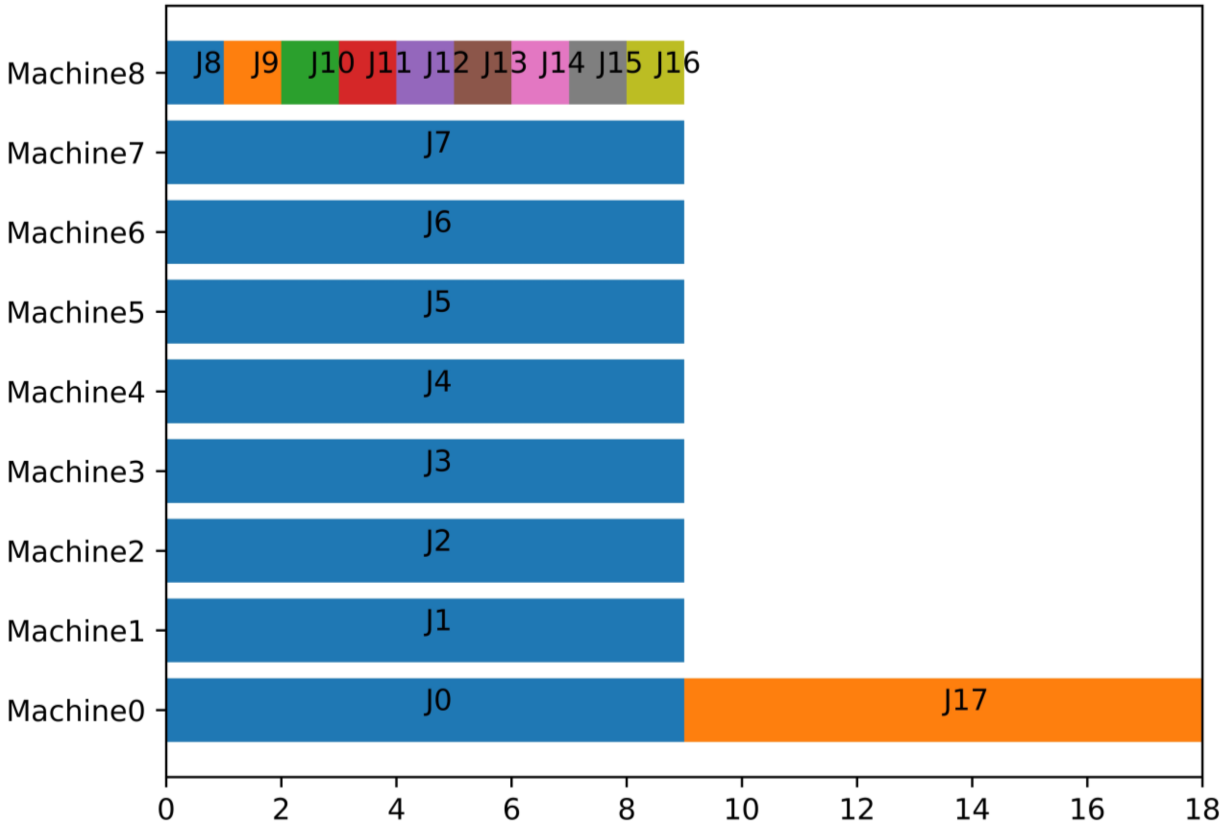
SUSTech Southern University of Science and Technology

THE SECOND AND THIRD EXAMPLE

These two question can be answered by one case.

Let's suppose that there are m machines, and the minimum unit of time is 1.

What if we have m jobs with time m, and m jobs with time 1?



Let the jobs queue be: [m, m, ……, m, 1, 1, ……, 1, 1, m]

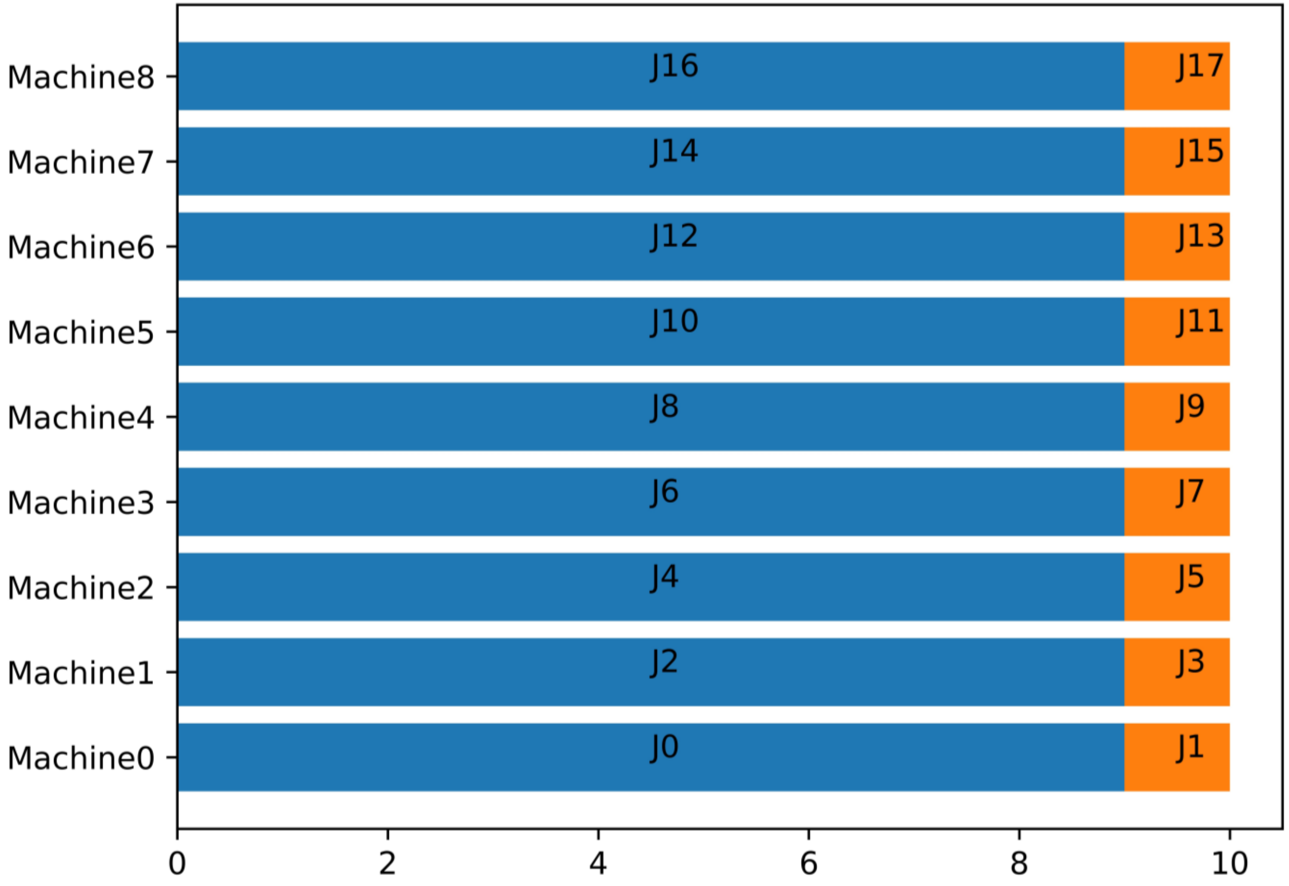
SUSTech Southern University of Science and Technology

THE SECOND AND THIRD EXAMPLE

These two question can be answered by one case.

Let's suppose that there are m machines, and the minimum unit of time is 1.

What if we have m jobs with time m, and m jobs with time 1?



Let the jobs queue be: [m, 1, m, 1, m, 1, ……, m, 1, m, 1]

SUSTech Southern University of Science and Technology

THE SECOND AND THIRD EXAMPLE

These two question can be answered by one case.

Let's suppose that there are m machines, and the minimum unit of time is 1.

What if we have m jobs with time m, and m jobs with time 1?

For the two examples above:

1. When the job queue is [m, 1, m, 1, m, 1, ……, m, 1, m, 1], the result of the greedy algorithm is optimal, $T=T^{*}=m+1$
2. When the job queue is [m, m, m, ……, m, 1, 1, ……, 1, 1, m], it will get the worst solution of the greedy algorithm, $T=2m$

So, for these jobs, the result can equal to T^{*}, and also can equal to $2T^{*}-2$, which is depend on the order of the jobs.

Southern University of Science and Technology

THANK YOU !

Q&A

Southern University of Science and Technology