LOADING BALANCE PROBLEM

Sorted greedy loading balance strategy

Zhiyuan Wang
12032878

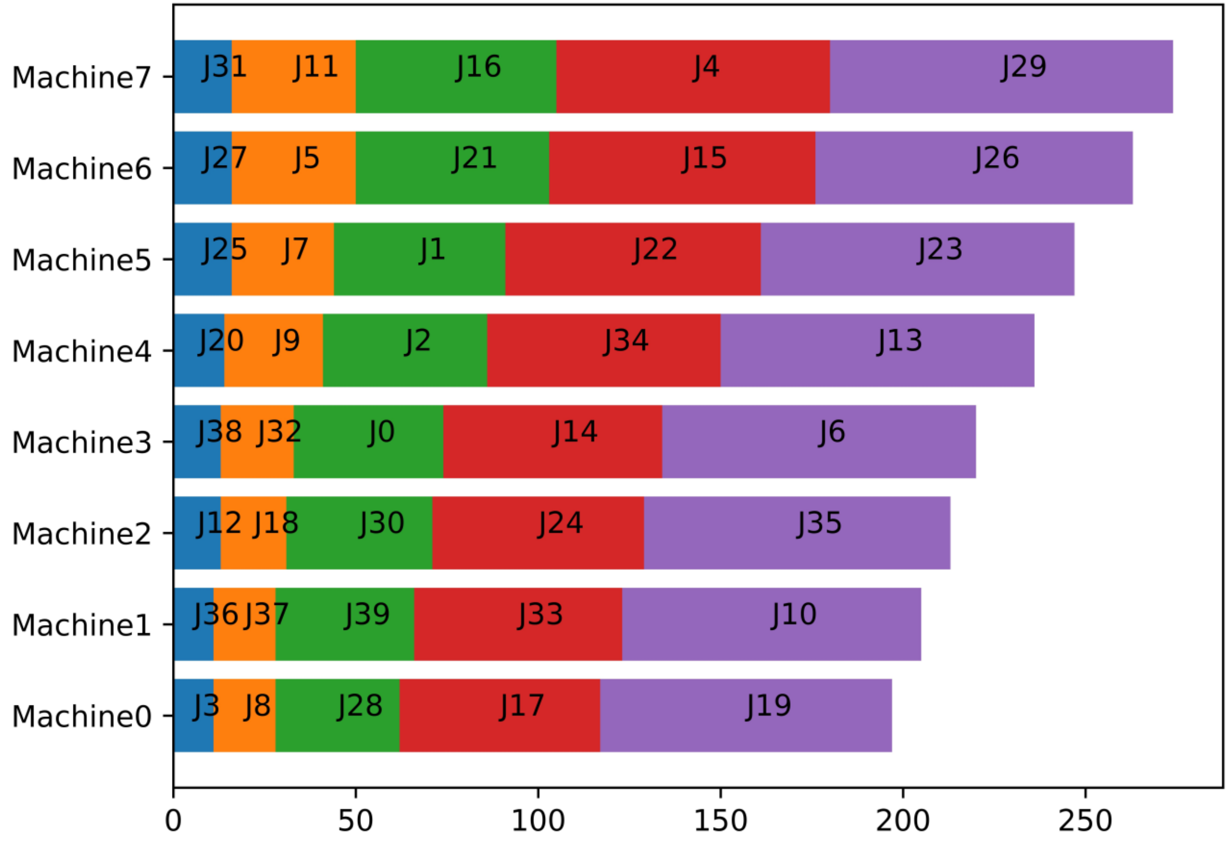
Southern University
of Science and
Technology

CONTENT

1. Loading Balance Problem
2. Sorted greedy loading balance strategy
3. The example where the obtained makespan T is very close to $\frac{4}{3}T*$
4. The algorithm for loading balance problem that some machines are more efficient than the others

LOADING BALANCE PROBLEM

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



SORTED GREEDY LOADING BALANCE STRATEGY

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

Algorithm 3 Sorted Loading Balance Problem

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

function $sorted\_greedy\_workload\_balance(m, job\_time)$
    Sort(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**
        $i \leftarrow argmin_k L_k$
        $M(i) \leftarrow M(i).append(j)$
        $L_i \leftarrow \mathrm{L}_i + job\_time[j]$
    end for
    returnM
end function

Southern University of Science and Technology

EFFECT OF ALGORITHM

If the theoretical optimal makespan of the job queue is T^*, then the sorted greedy algorithm's makespan T will not worse than $\frac{4}{3}T*$.

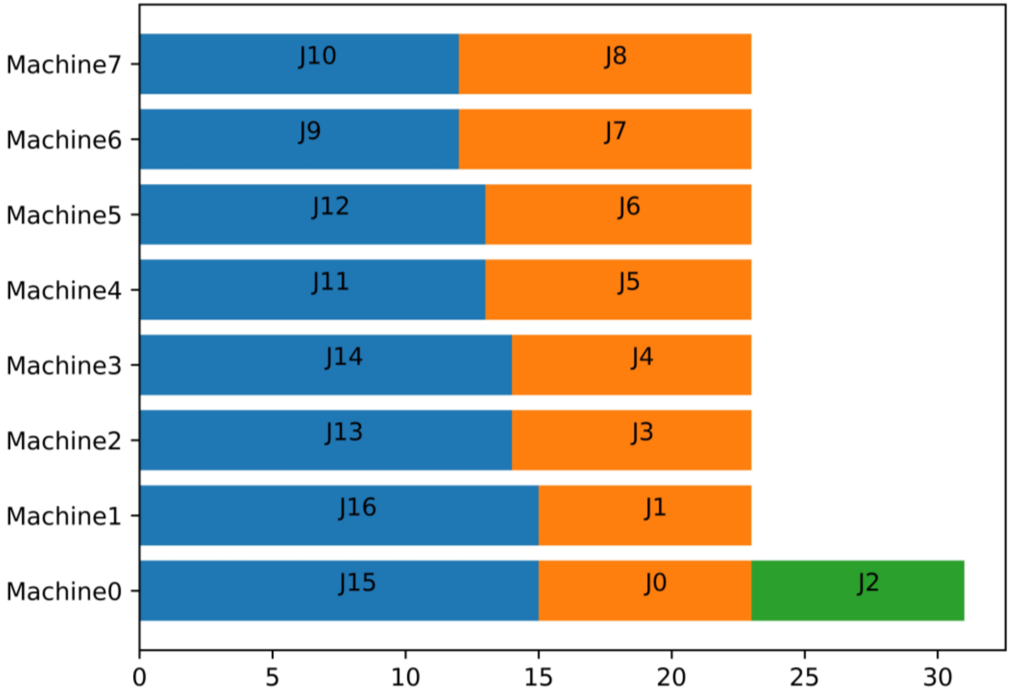
$$\boldsymbol{T^* \leq T \leq \frac{4}{3}T^*}$$

THE FIRST EXAMPLE

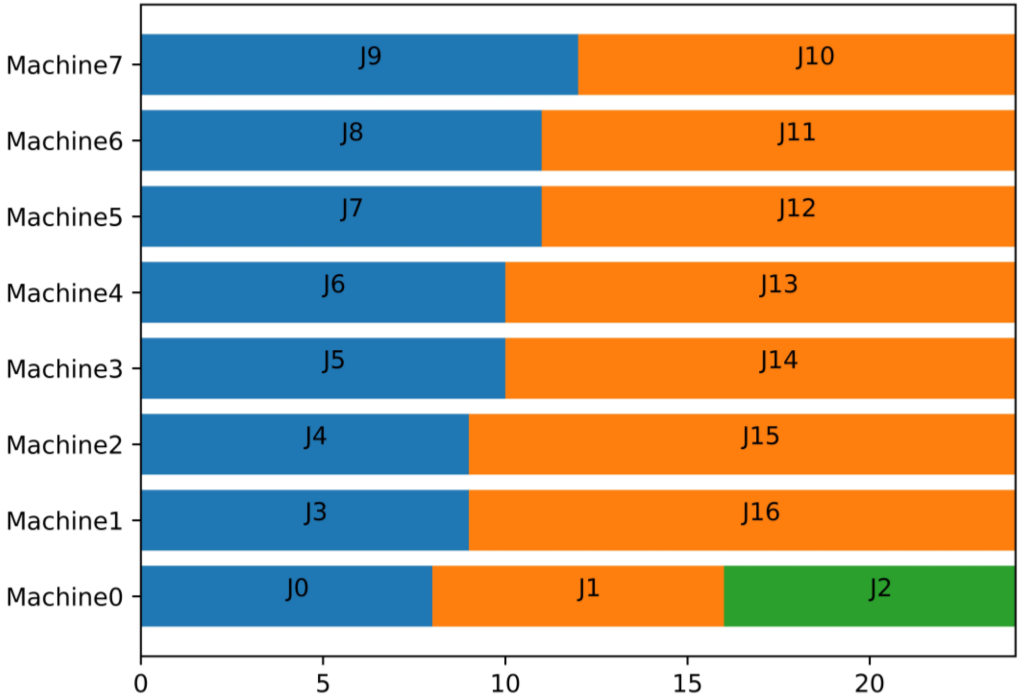
The first example should satisfy that the obtained makespan T is very close to $\frac{4}{3}T^*$.

How to build this example?

Suppose we have m machines, then we can create 3 jobs with time m, 2 jobs for time m+1, 2 jobs for time m+2,…, 2jobs for time 2m-1.



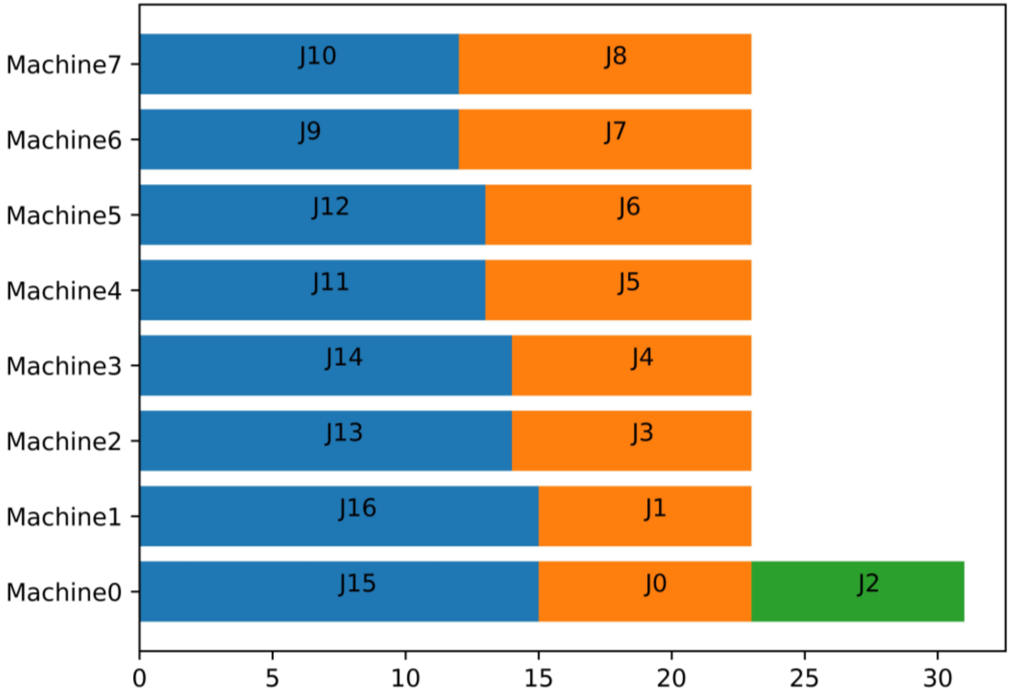
Result of sorted greedy algorithm



Optimal result

SUSTech Southern University of Science and Technology

THE FIRST EXAMPLE



Result of sorted greedy algorithm

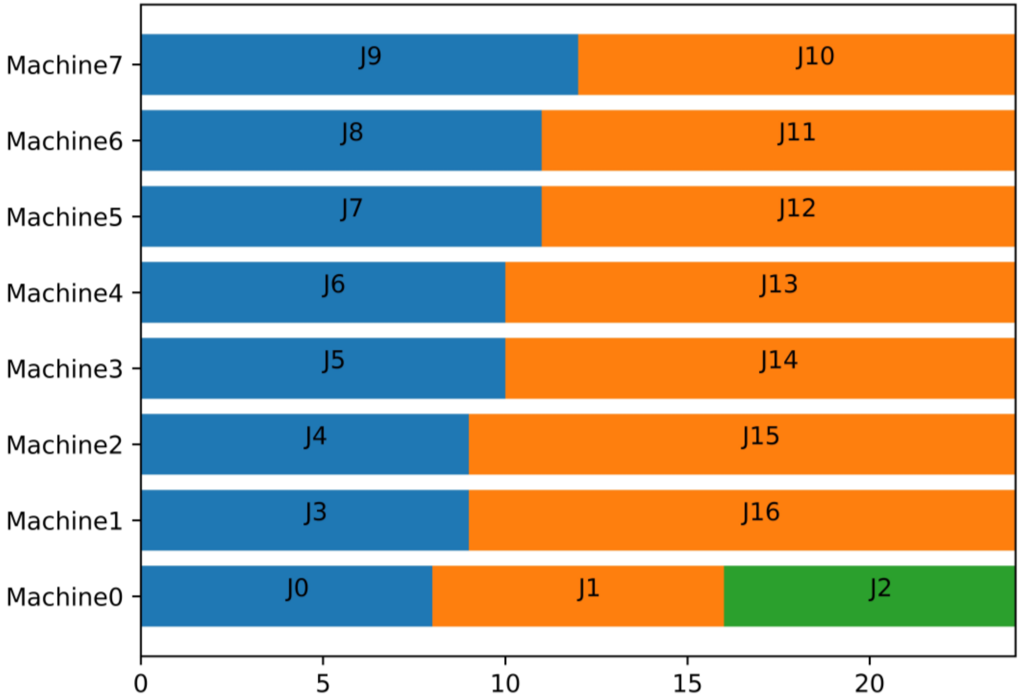
Is this way always work?

In the first two rounds, all the machines spend 3m-1 time. And in the third round, there is a machine spend m times. It need 4m-1 time total.

For the optimal solution. I combine jobs with (2m-1, m+1), (2m-2, m+2),…,(2m-floor($\frac{m}{2}$), m+ceil($\frac{m}{2}$)), (m,m,m). Which means it need 3m time total.

So the makespan T is $\frac{4m-1}{3m}T^*$.

Optimal result



SUSTech Southern University of Science and Technology

ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS

I use a greedy strategy like sorted greedy loading balance strategy.

In this algorithm, it will sort the task list by the average loading on the machines firstly.

Then, it will assign the ith job to the machine and let the machines solve the first i jobs fastest.

Algorithm 4 Sorted Heterogeneous Loading Balance Problem

Require: m : number of machines, job_times : The time of each jobs on each machine
Ensure:
function $sorted\_greedy\_workload\_balance(m, n, job\_time)$
    Sort job_time By $average\ loading$
    $M \leftarrow []$
    for i=1 to m **do**
        $L_i \leftarrow 0$
        $M(i) \leftarrow []$
    end for
    for j=1 to n **do**
        $i \leftarrow argmin_k(L_k + job\_time_{j,k})$
        $M(i) \leftarrow M(i).append(j)$
        $L_i \leftarrow \mathrm{L}_i + job\_time_{j,k}$
    end for
    returnM
end function

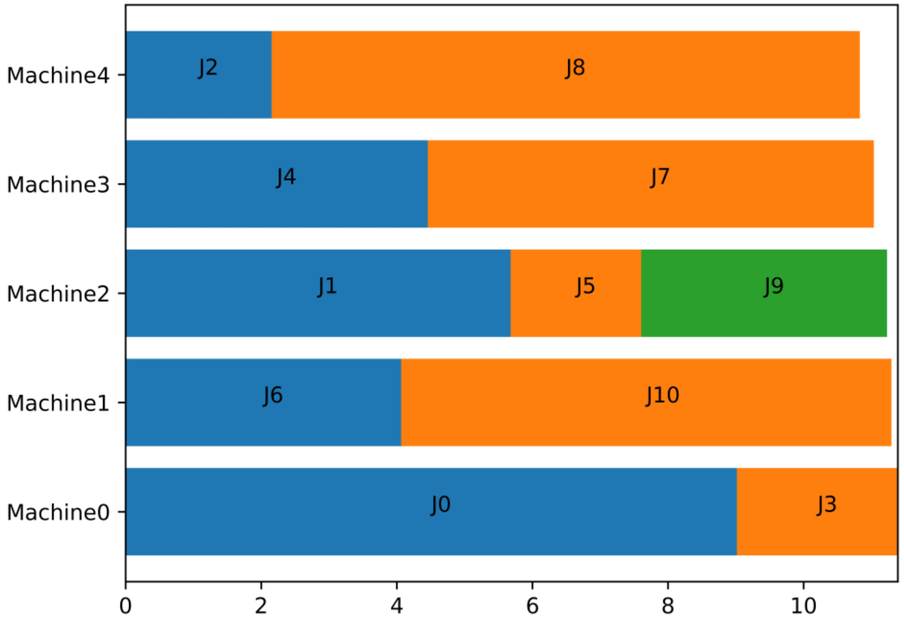
ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS

Experiment to verify the quality of the algorithm.

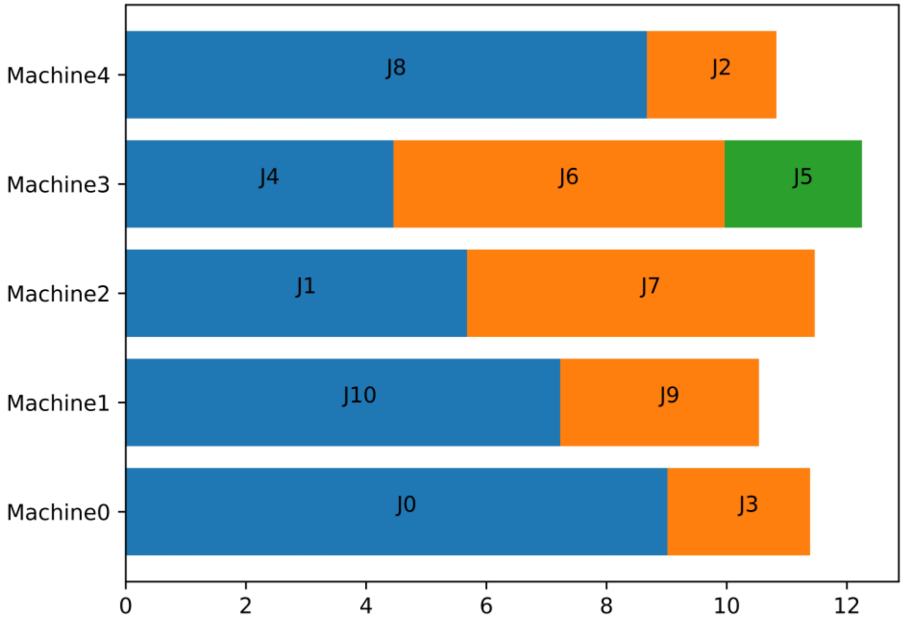
1. For the jobs, I generate the base cost for each job, and then I generate a factor between 0.5 to 1.5 to each job on each machine.
2. Generate 1000 cases respectively for 3 machines with 7 jobs, 3 machines with 8 jobs, 4 machines with 9 jobs and 5 machines with 11 jobs.
3. Give some examples
4. Do statistic for the quality of the cases

Job / Machine	J1	J2	J3
Base	2	3	4
M1	1	2.34	3.2
M2	1.81	3.12	4.21
m3	3	4.11	5.64

ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS

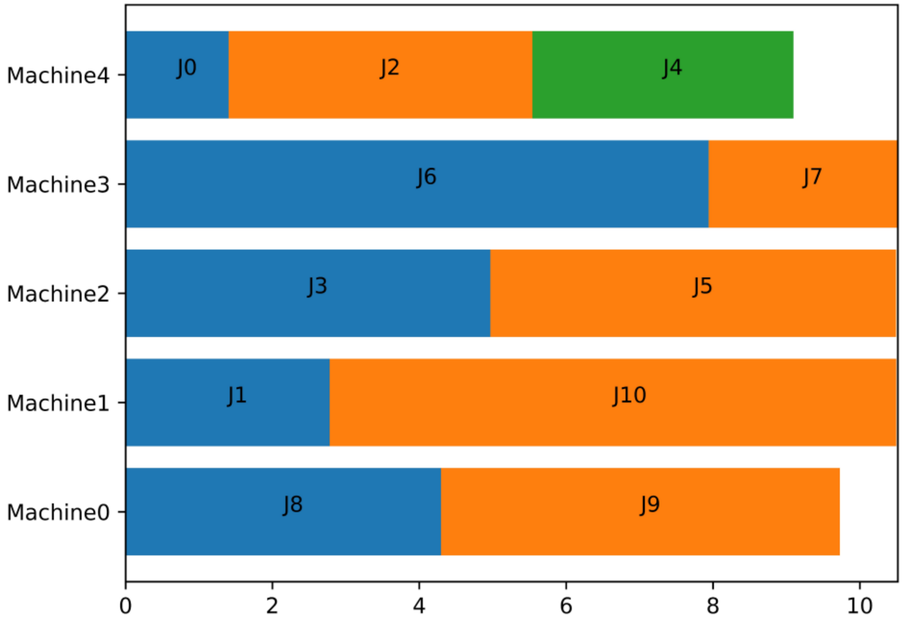


Optimal Solution

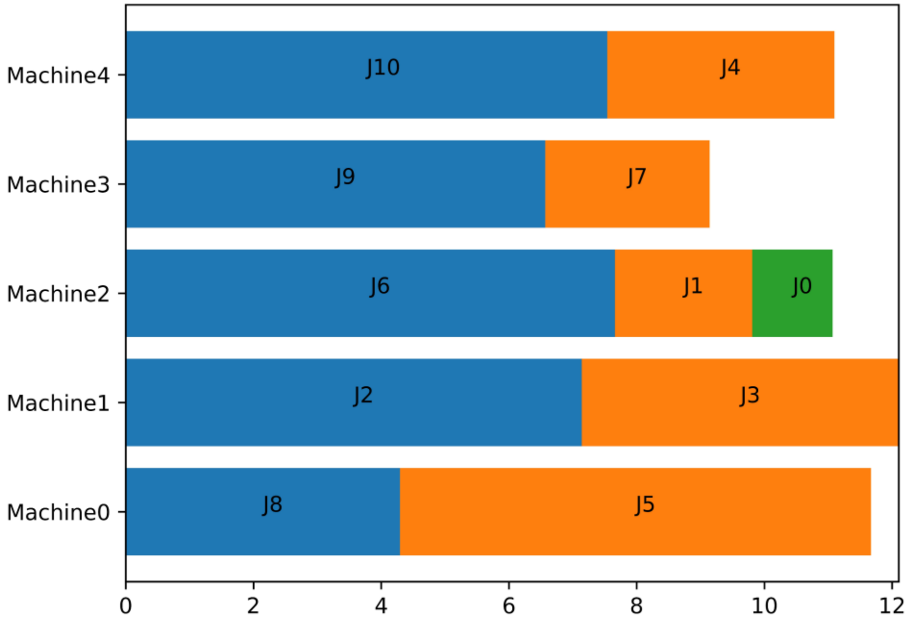


Greedy Solution

ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS



Optimal Solution



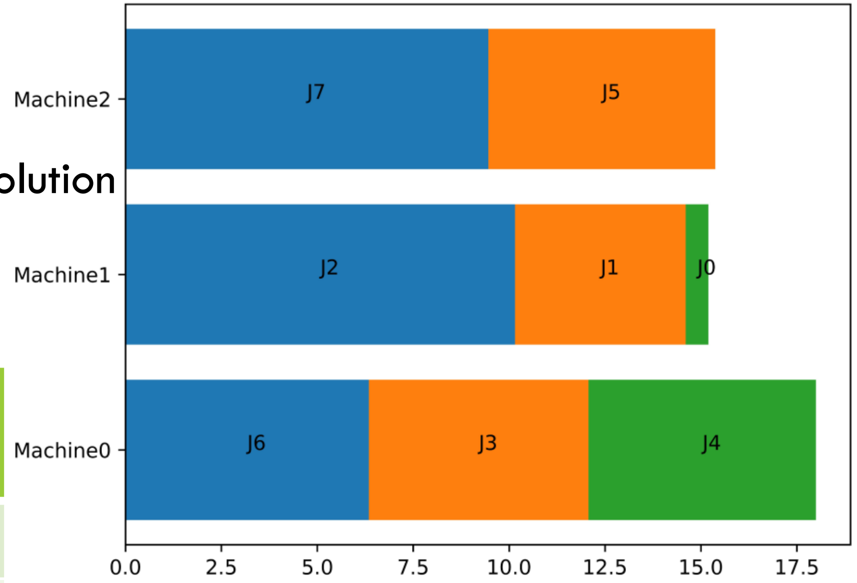
Greedy Solution

SUSTech Southern University of Science and Technology

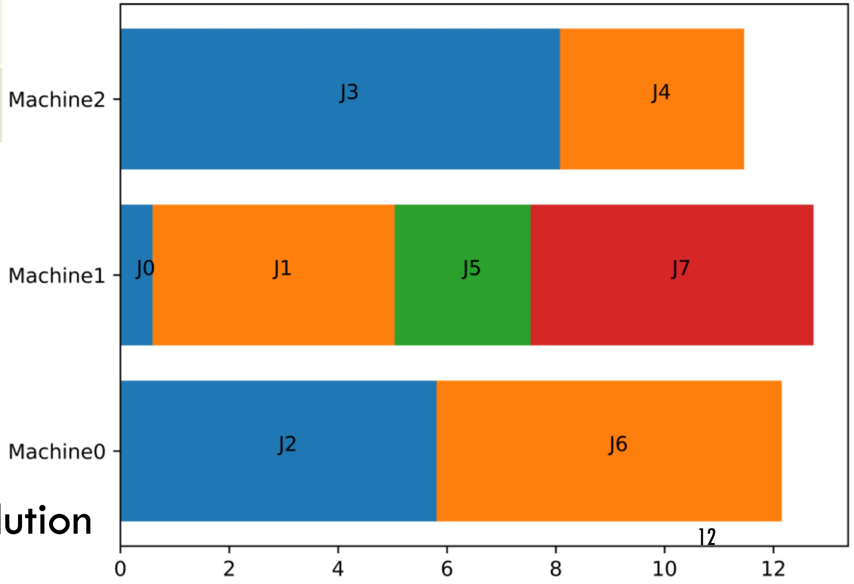
ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS

Job / Machine	J0	J1	J2	J3	J4	J5	J6	J7
M1	1.19	7.42	5.81	5.72	5.93	5.83	6.34	10.80
M2	0.59	4.45	10.15	10.26	4.46	2.50	8.60	5.20
m3	1.35	11.07	11.89	8.08	3.38	5.92	13.34	9.46

Greedy Solution



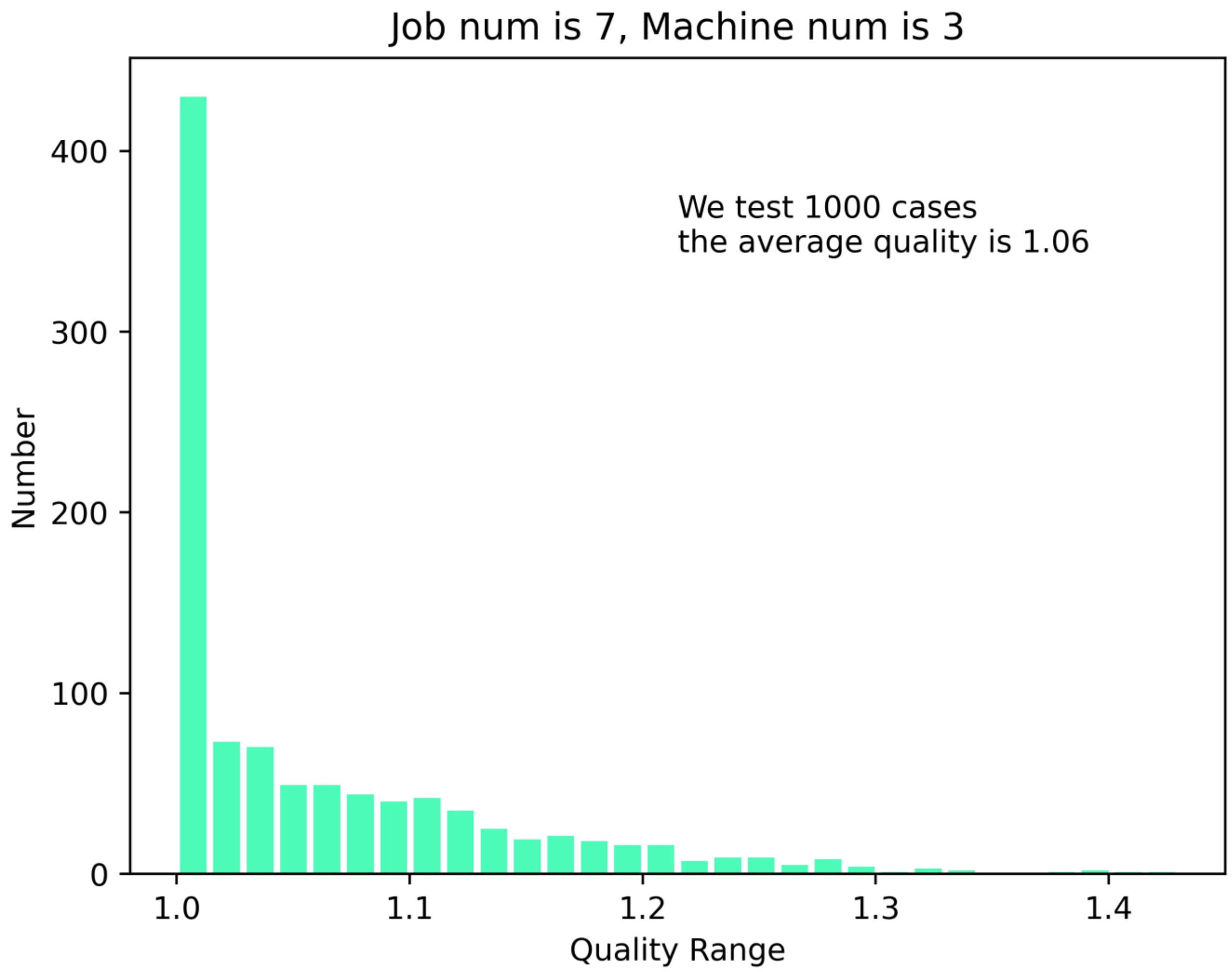
Quality is 1.4133



Optimal Solution

Southern University of Science and Technology

ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS



Southern University of Science and Technology

ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS

Southern University of Science and Technology

ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS

Southern University of Science and Technology

ALGORITHM FOR THE CASE THAT SOME MACHINES ARE MORE EFFICIENT THAN THE OTHERS

Southern University of Science and Technology

THANK YOU !

Q&A

Southern University of Science and Technology