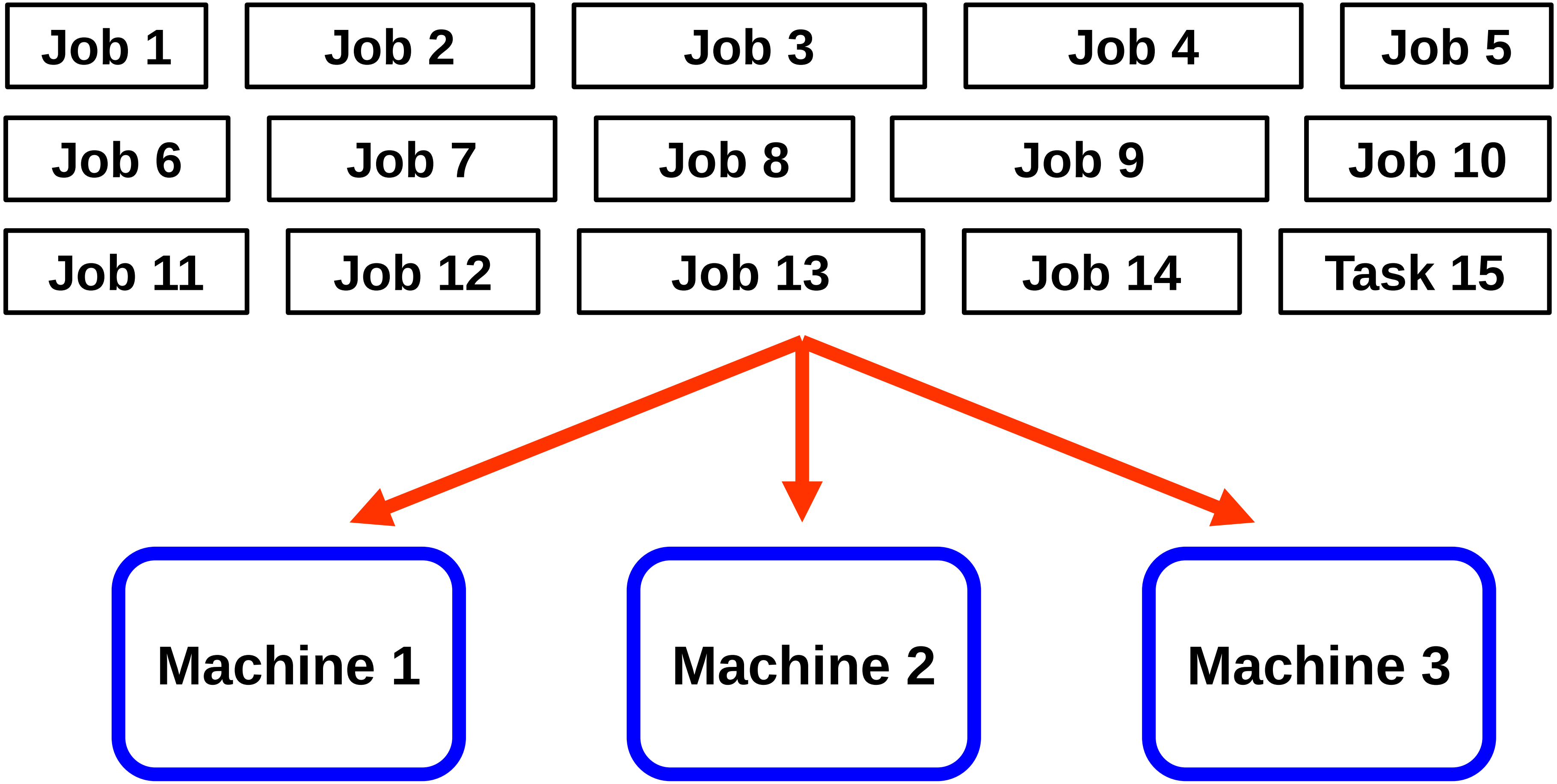
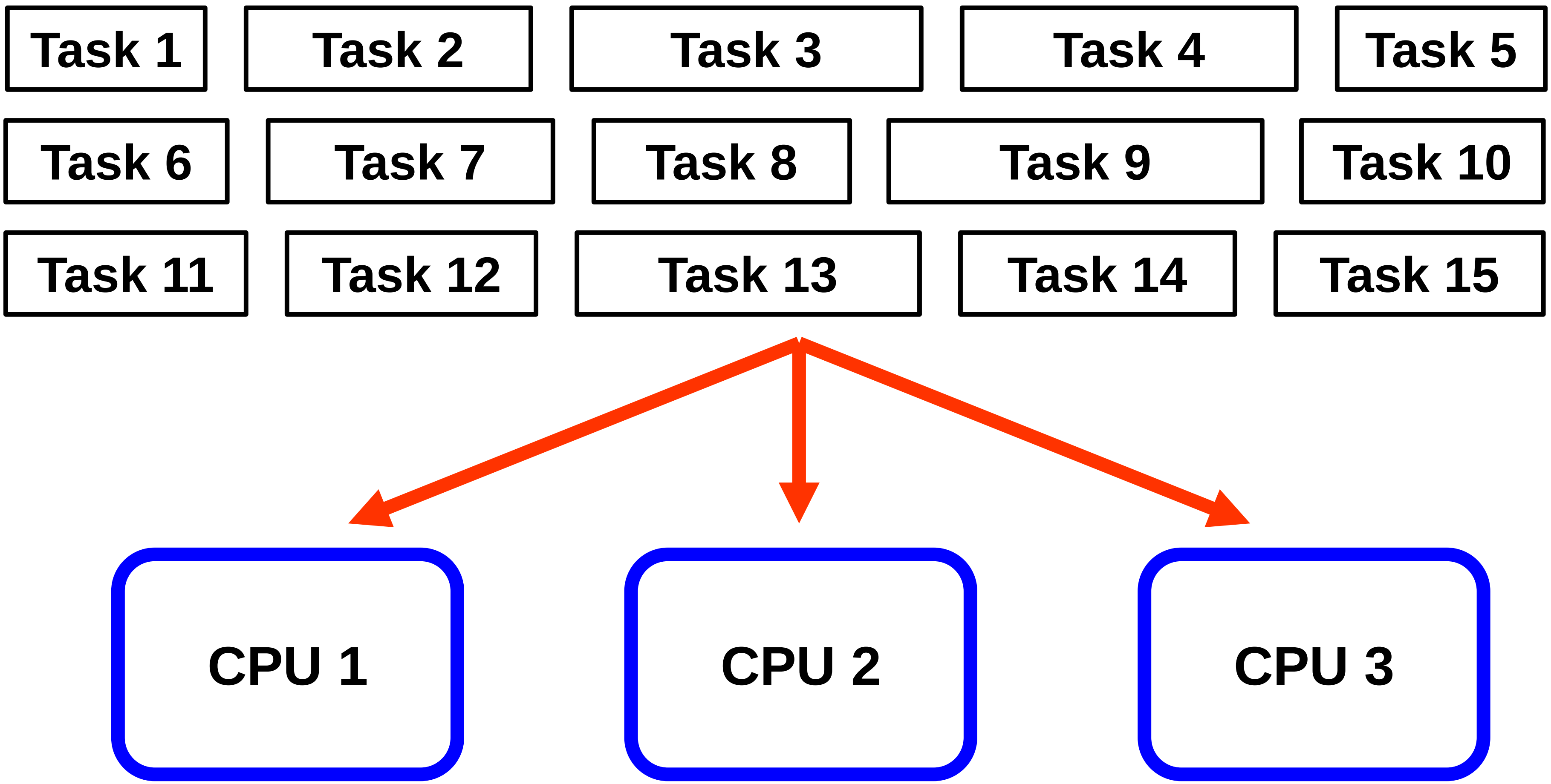
Topic 1: Load Balancing Problem

Q. When a number of jobs with different processing time and a number of identical machines are given, how do you assign the jobs to the machines? (Random?)



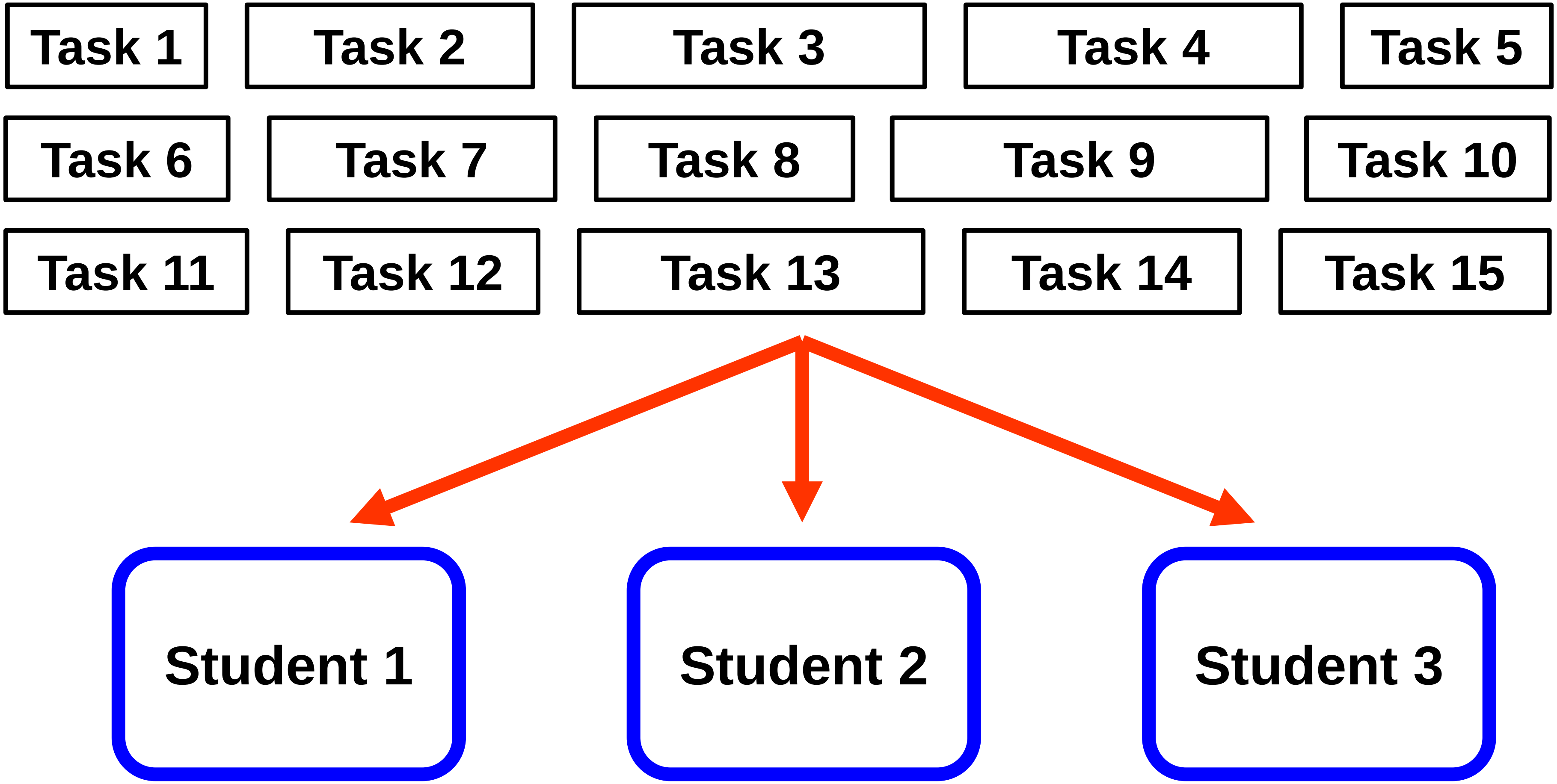
Topic 1: Load Balancing Problem

Q. When a number of tasks with different computation time and a number of identical CPUs are given, how do you assign the tasks to the CPUs? (Random?)



Topic 1: Load Balancing Problem

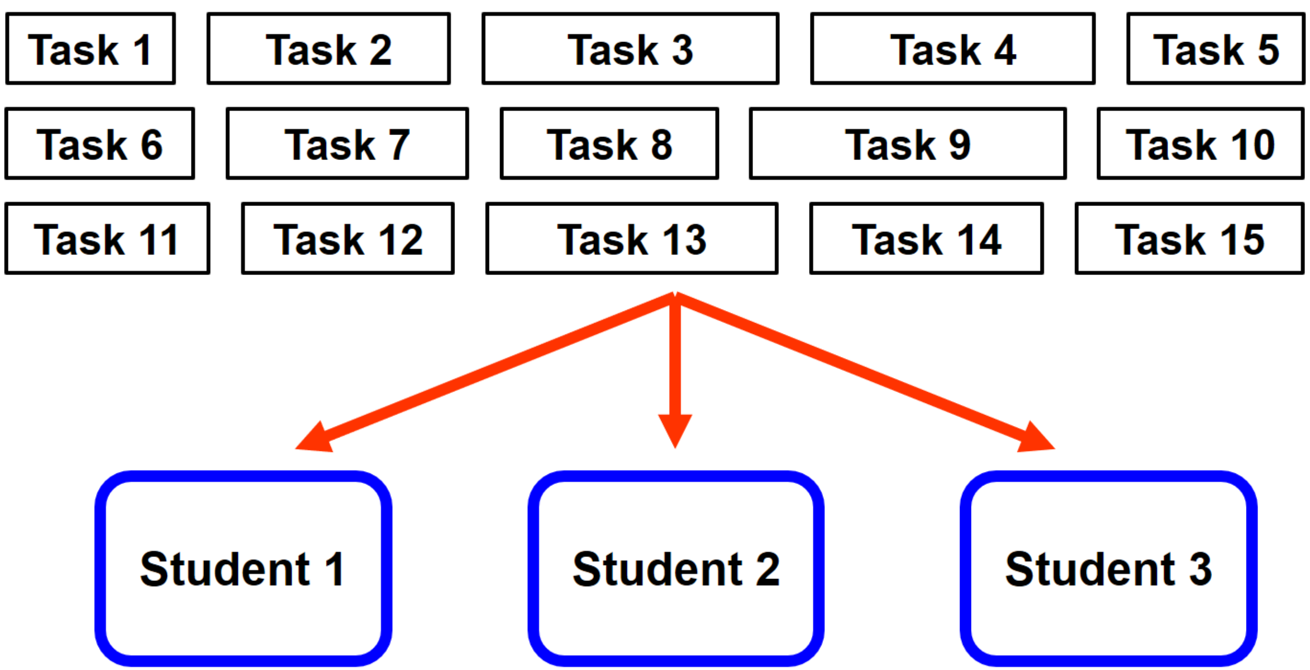
Q. **When we have a number of tasks with different work load (working hours) and a number of students, how should we assign the tasks to the students? (Random?)**



Topic 1: Load Balancing Problem

Q. When we have a number of tasks with different work load (working hours) and a number of students, how should we assign the tasks to the students? (Random?)

Q. What is our goal? (e.g., to assign all tasks to a single student, to assign almost the same amount of tasks to each student)



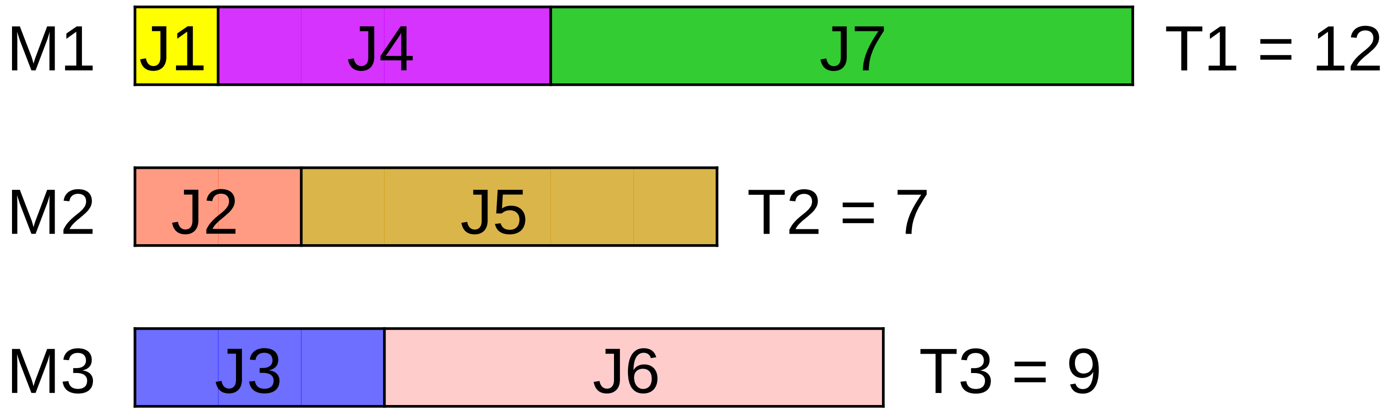
Topic 1: Load Balancing Problem

Input: m identical machines: M1, M2, ..., Mm

n jobs: J1, J2, ..., Jn

Processing time of each job: t_j (j = 1, 2, ..., n)

Example: 3 machines and 7 jobs (t_j = 1, 2, 3, 4, 5, 6, 7)



Makespan T = max {T1, T2, T3} = 12

Q: What is the best assignment ?

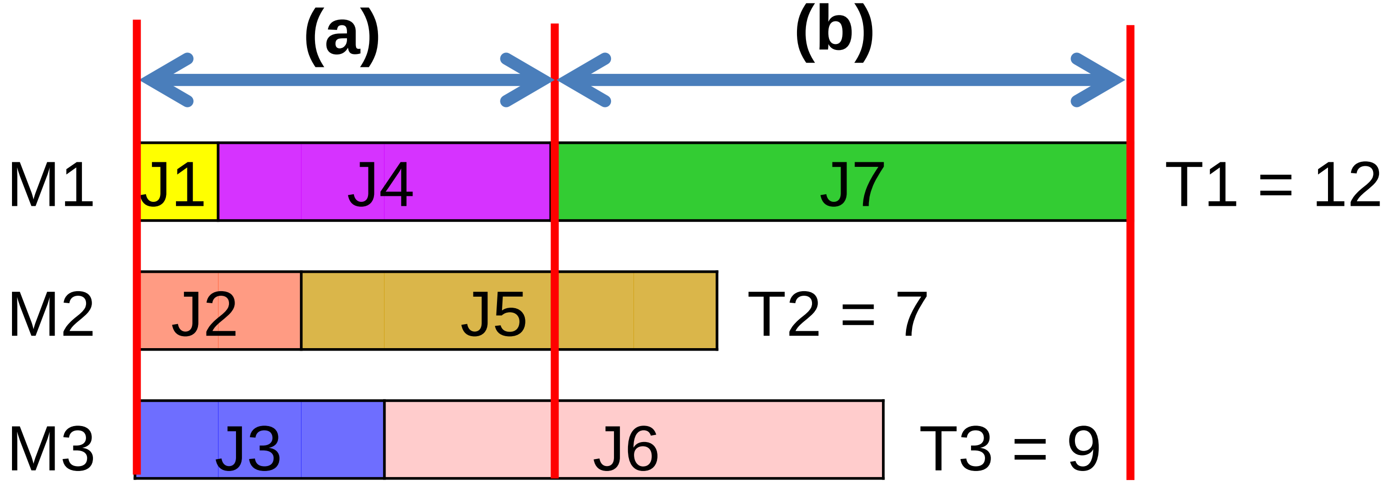
Greedy Algorithm

Assign a job to the machine with the smallest load in an arbitrary order of jobs.

Q: How good is this greedy algorithm?

The obtained makespan T is not worse than $2T^*$ where T^* is the optimal makespan ($T \leq 2T^*$): 2-approximation

$$\text{(a)} \;<\; \frac{1}{m}\sum_{j=1}^{n} t_j \;\leq\; T^* \qquad \text{(b)} \;\leq\; \max\{t_j\} \;\leq\; T^*$$

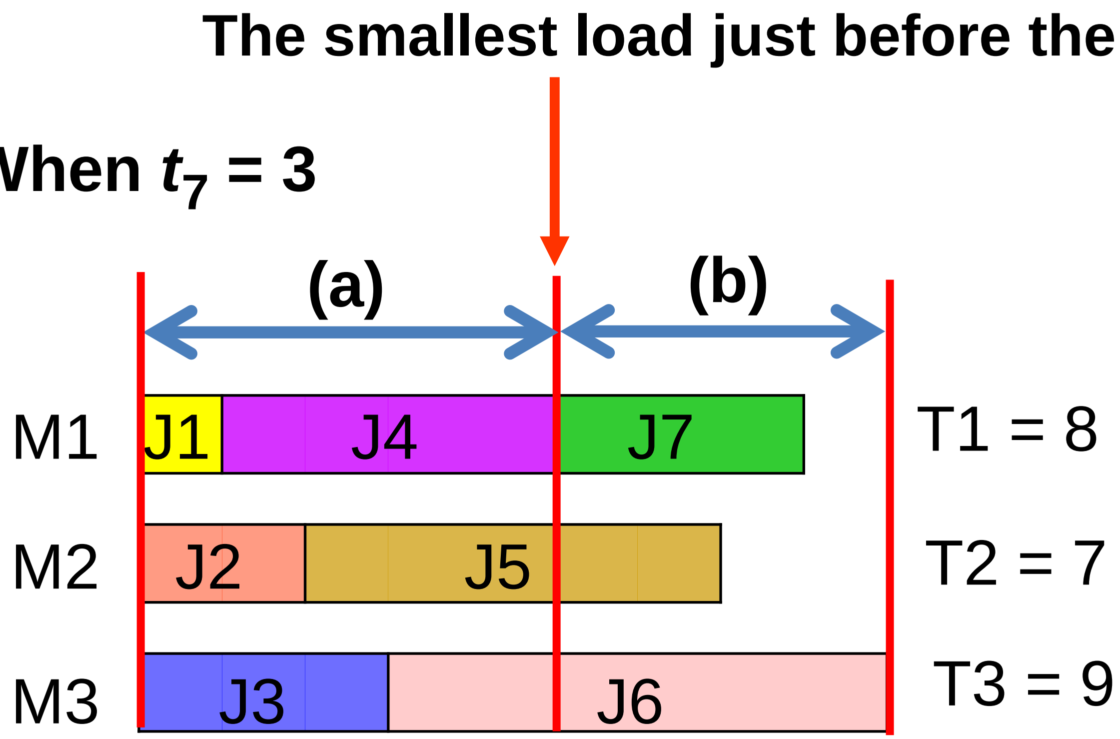


Q: How good is this greedy algorithm?

The obtained makespan T is not worse than $2T^*$ where T^* is the optimal makespan ($T \le 2T^*$): 2-approximation

$$\text{(a)} < \frac{1}{m}\sum_{j=1}^{n} t_j \le T^* \qquad \text{(b)} \le \max\{t_j\} \le T^*$$

When $t_7 = 3$



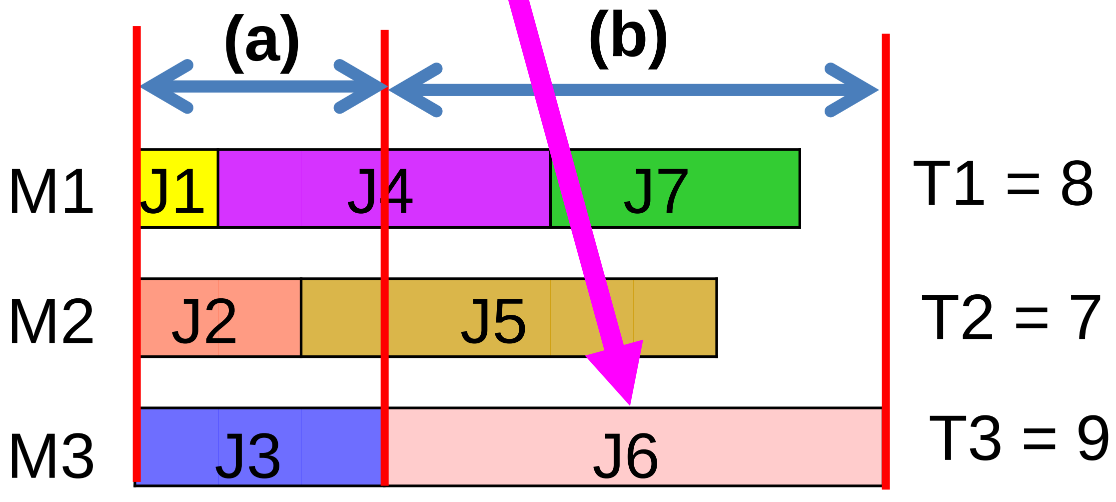
Q: How good is this greedy algorithm?

The obtained makespan T is not worse than $2T^*$ where T^* is the optimal makespan ($T \le 2T^*$): 2-approximation

$$\text{(a)} < \frac{1}{m}\sum_{j=1}^{n} t_j \le T^* \qquad \text{(b)} \le \max\{t_j\} \le T^*$$

The last job at the machine with the largest makespan.

When $t_7 = 3$



```
List-Scheduling(m, n, t_1,t_2,...,t_n) {
    for i = 1 to m {
        L_i ← 0              ← load on machine i
        J(i) ← ϕ             ← jobs assigned to machine i
    }

    for j = 1 to n {
        i = argmin_k L_k          ← machine i has smallest load
        J(i) ← J(i) ∪ {j}         ← assign job j to machine i
        L_i ← L_i + t_j           ← update load of machine i
    }
    return J(1), …, J(m)
}
```

procedure GREEDY-BALANCE
    1 pass through jobs in any order.
    Assign job j to machine with current smallest load.
end procedure

Q: How tight is this upper bound?

Exercise 1-1:

Try to create an example where the obtained makespan T is always the same as T^*. (Easy example)

Exercise 1-1:

Try to create an example where the obtained makespan T is very close to $2T^*$ for a particular order of jobs. (If there exists such an example, we can say that the upper limit $2T^*$ is tight.)

Exercise 1-3:

Try to create an example where the obtained makespan T strongly depends on the order of jobs (i.e., the obtained makespan is close to $2T^*$ for some orders of jobs and (almost) the same as T^* for some other orders of jobs).

Discussions:

Any real-world problems similar to load balancing?