Networked Embedded Systems WS 2016/17

Exercise 1: Real-time Scheduling

Marco Zimmerling

TECHNISCHE UNIVERSITÄT DRESDEN

cfaed CENTER FOR ADVANCING ELECTRONICS DRESDEN

Tasks

1. Scheduling Function and Parameters of Real-time Tasks
2. Earliest Deadline Due (EDD)
3. Earliest Deadline First (EDF)

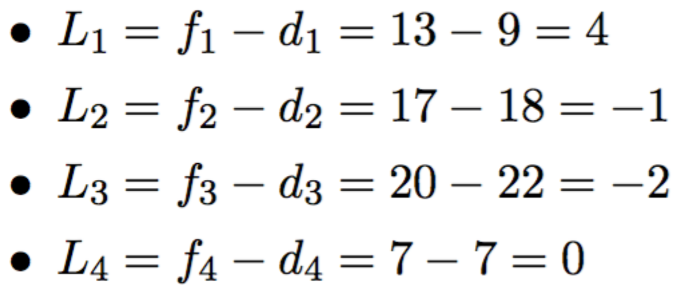
Aperiodic tasks

4. Fixed-priority Scheduling: Rate Monotonic (RM)
5. Dynamic-priority Scheduling: Earliest Deadline First (EDF)

Periodic tasks

Task 1 (a): Sample Solution

- Lateness of each task

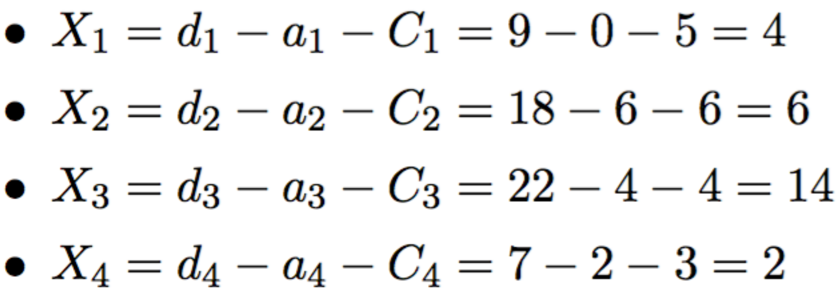


- $L_1 = f_1 - d_1 = 13 - 9 = 4$
- $L_2 = f_2 - d_2 = 17 - 18 = -1$
- $L_3 = f_3 - d_3 = 20 - 22 = -2$
- $L_4 = f_4 - d_4 = 7 - 7 = 0$

- Task J_4 induces the maximum lateness $L_4 = 4$. It is the only task that violates the specified timing constraints, and hence is the only task with a lateness greater than 0.

Task 1 (b): Sample Solution

- Laxity of each task



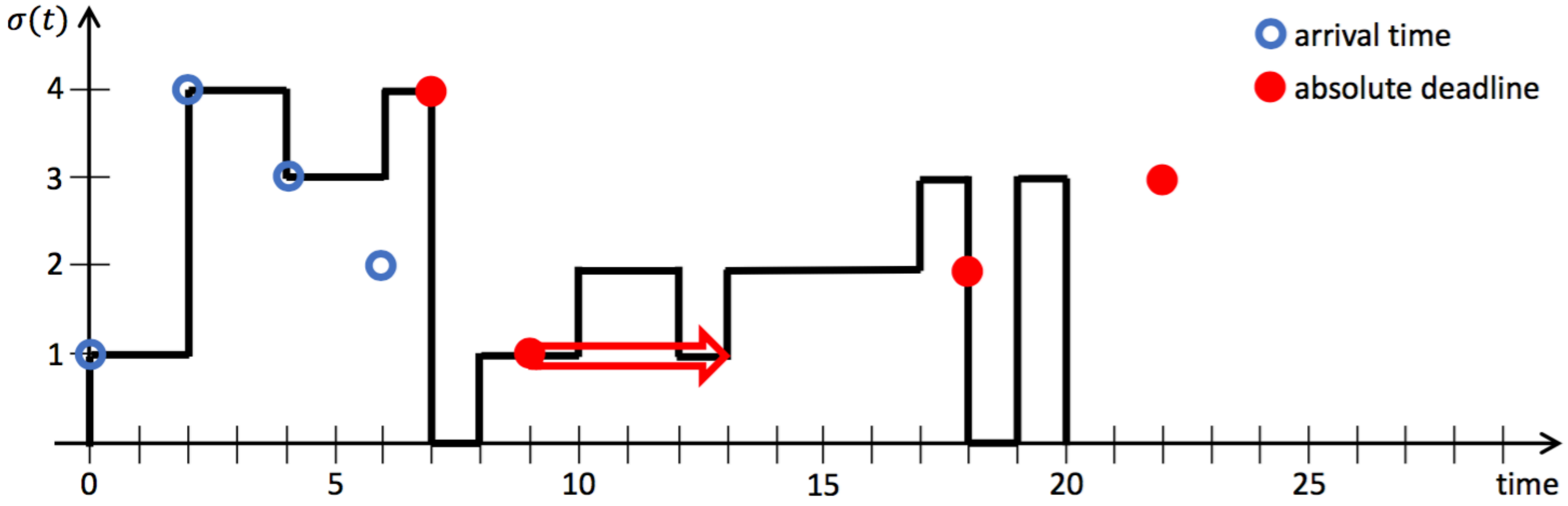
- $X_1 = d_1 - a_1 - C_1 = 9 - 0 - 5 = 4$
- $X_2 = d_2 - a_2 - C_2 = 18 - 6 - 6 = 6$
- $X_3 = d_3 - a_3 - C_3 = 22 - 4 - 4 = 14$
- $X_4 = d_4 - a_4 - C_4 = 7 - 2 - 3 = 2$

Task 1 (c): Sample Solution

- Processor utilization $U = \frac{18}{20} = 0.9$

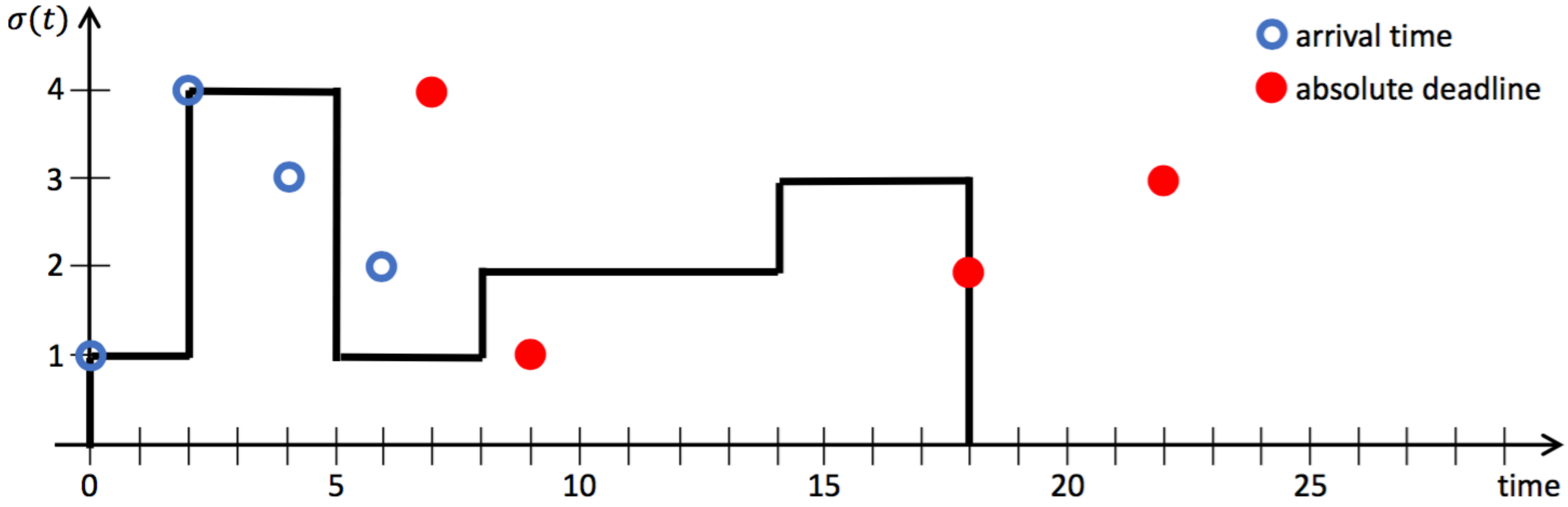
Task 1 (d): Sample Solution

- A schedule is said to be *feasible* if all tasks can be completed according to a set of specified constraints.
- The given scheduling function does *not* yield a feasible schedule.



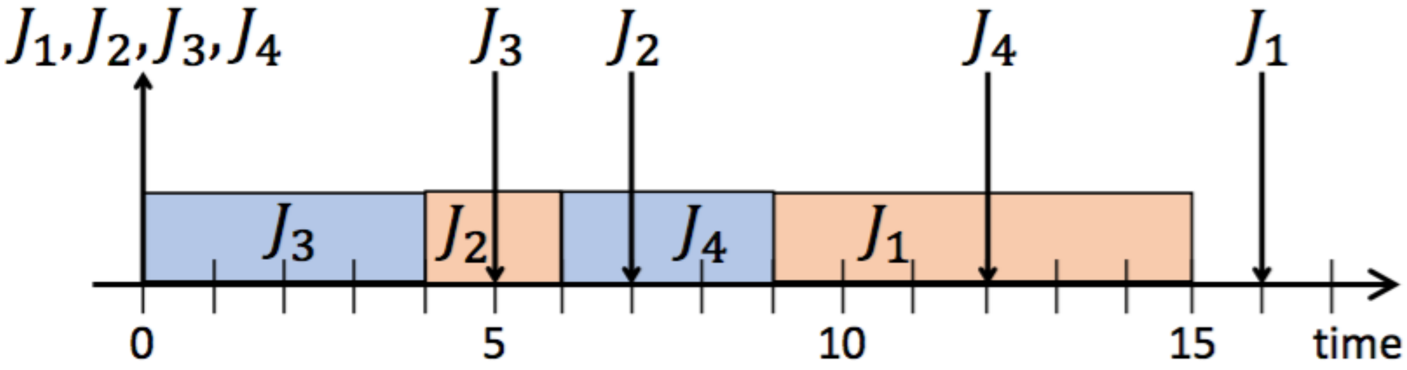
Task 1 (d): Sample Solution

- Below is one of the many possible modified scheduling functions that completes all tasks by their deadline (*i.e.*, produces a feasible schedule.)



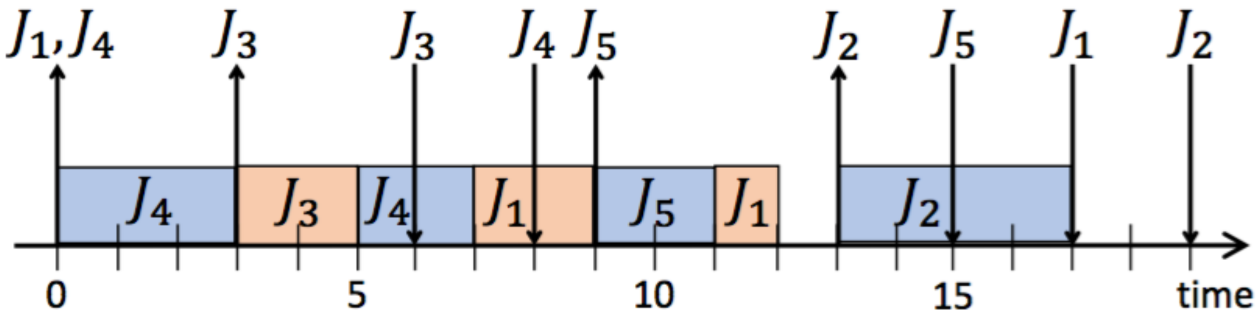
Task 2: Sample Solution

- EDD can schedule independent tasks with the same arrival time.
- EDD executes tasks in order of non-decreasing deadline.
- Applied to the given task set, EDD produces a feasible schedule, as shown below.



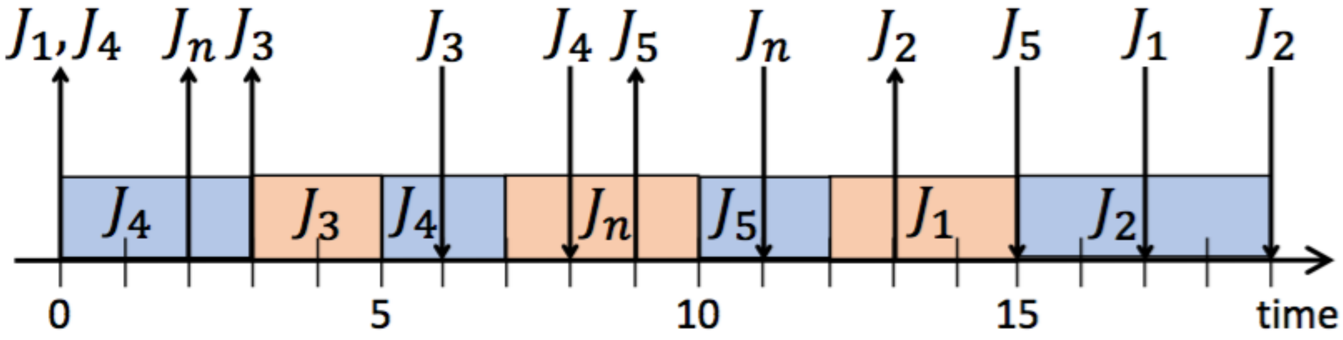
Task 3 (a): Sample Solution

- EDF can schedule independent tasks with the arbitrary arrival times in a preemptive fashion.
- EDF executes, at any point in time, the task with the earliest absolute deadline among all ready tasks. in order of non-decreasing deadline.
- Applied to the given task set, EDF produces a feasible schedule, as shown below.



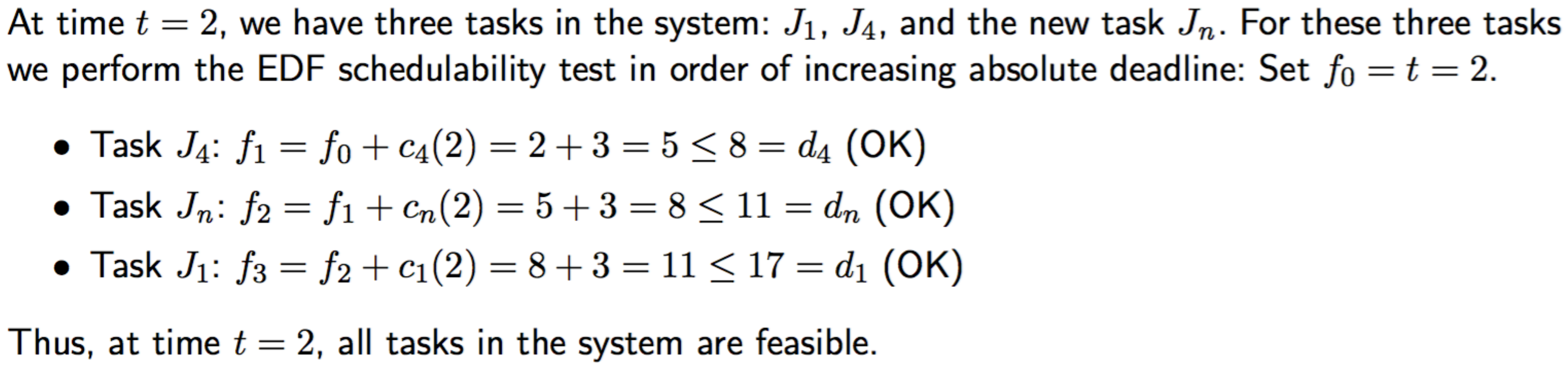
Task 3 (b): Sample Solution

- The new task can be accepted, because the resulting task set remains schedulable.



- This can be checked by computing at certain interesting points in time the worst-case finishing times of the tasks and comparing them to the absolute deadlines.
- We perform this check in an online fashion:
  - Conduct EDF schedulability test each time a new task arrives
  - Consider only those tasks that are currently present in the system
  - Process those tasks in order of increasing absolute deadline

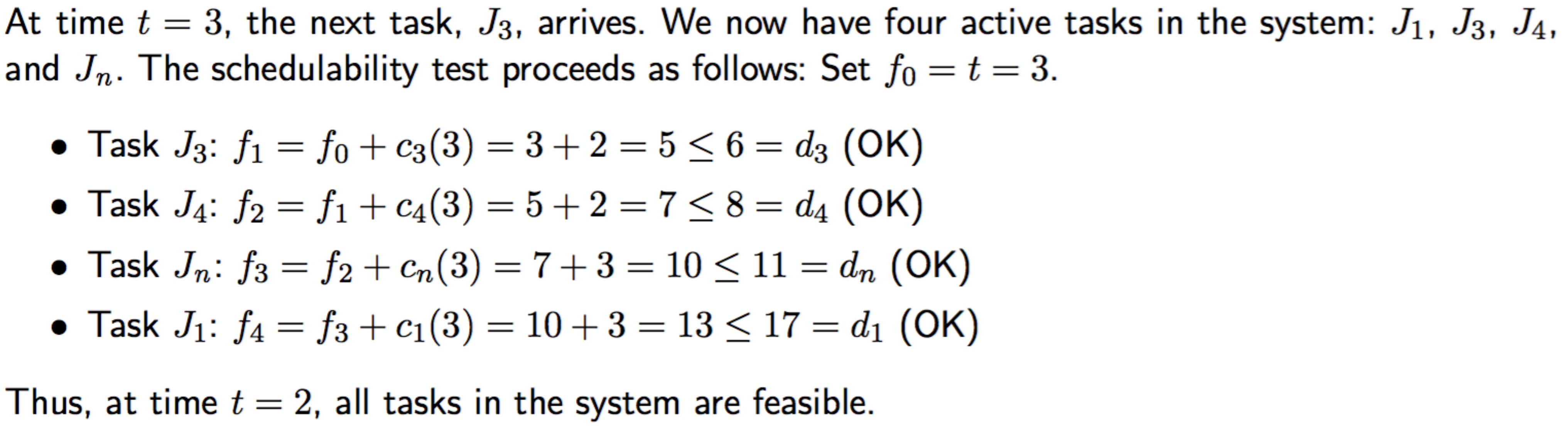
Task 3 (b): Sample Solution



At time $t = 2$, we have three tasks in the system: J_1, J_4, and the new task J_n. For these three tasks we perform the EDF schedulability test in order of increasing absolute deadline: Set $f_0 = t = 2$.

- Task J_4: $f_1 = f_0 + c_4(2) = 2 + 3 = 5 \leq 8 = d_4$ (OK)
- Task J_n: $f_2 = f_1 + c_n(2) = 5 + 3 = 8 \leq 11 = d_n$ (OK)
- Task J_1: $f_3 = f_2 + c_1(2) = 8 + 3 = 11 \leq 17 = d_1$ (OK)

Thus, at time $t = 2$, all tasks in the system are feasible.

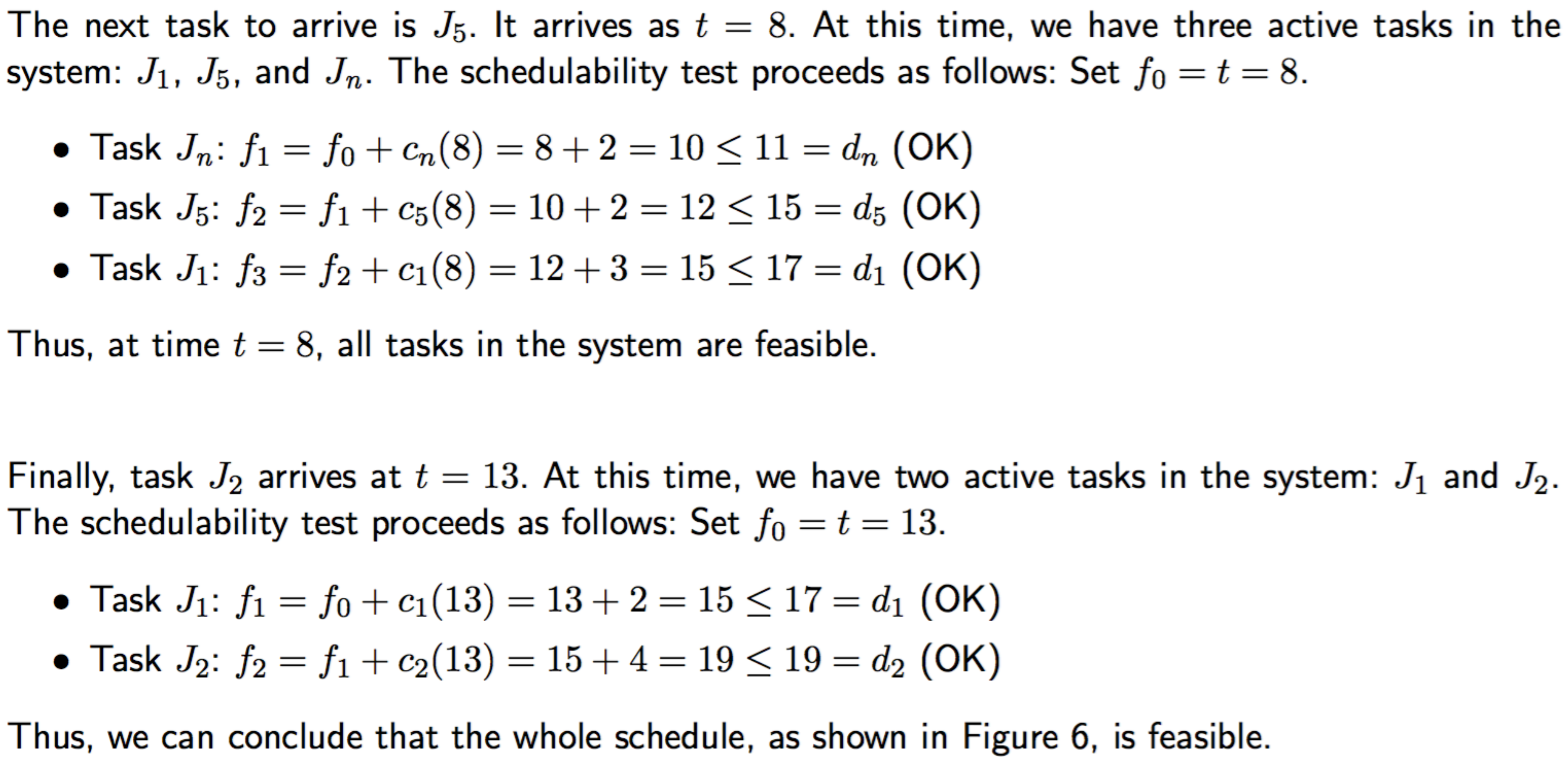


At time $t = 3$, the next task, J_3, arrives. We now have four active tasks in the system: J_1, J_3, J_4, and J_n. The schedulability test proceeds as follows: Set $f_0 = t = 3$.

- Task J_3: $f_1 = f_0 + c_3(3) = 3 + 2 = 5 \leq 6 = d_3$ (OK)
- Task J_4: $f_2 = f_1 + c_4(3) = 5 + 2 = 7 \leq 8 = d_4$ (OK)
- Task J_n: $f_3 = f_2 + c_n(3) = 7 + 3 = 10 \leq 11 = d_n$ (OK)
- Task J_1: $f_4 = f_3 + c_1(3) = 10 + 3 = 13 \leq 17 = d_1$ (OK)

Thus, at time $t = 2$, all tasks in the system are feasible.

Task 3 (b): Sample Solution



The next task to arrive is J_5. It arrives as $t = 8$. At this time, we have three active tasks in the system: J_1, J_5, and J_n. The schedulability test proceeds as follows: Set $f_0 = t = 8$.

- Task J_n: $f_1 = f_0 + c_n(8) = 8 + 2 = 10 \leq 11 = d_n$ (OK)
- Task J_5: $f_2 = f_1 + c_5(8) = 10 + 2 = 12 \leq 15 = d_5$ (OK)
- Task J_1: $f_3 = f_2 + c_1(8) = 12 + 3 = 15 \leq 17 = d_1$ (OK)

Thus, at time $t = 8$, all tasks in the system are feasible.

Finally, task J_2 arrives at $t = 13$. At this time, we have two active tasks in the system: J_1 and J_2. The schedulability test proceeds as follows: Set $f_0 = t = 13$.

- Task J_1: $f_1 = f_0 + c_1(13) = 13 + 2 = 15 \leq 17 = d_1$ (OK)
- Task J_2: $f_2 = f_1 + c_2(13) = 15 + 4 = 19 \leq 19 = d_2$ (OK)

Thus, we can conclude that the whole schedule, as shown in Figure 6, is feasible.

Task 4 (a): Sample Solution

- A set of n periodic real-time tasks is schedulable using RM if

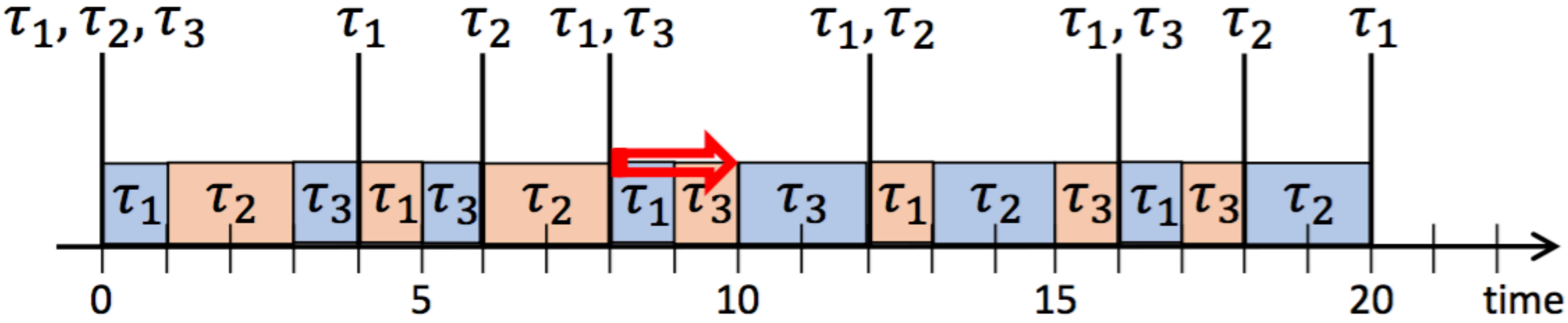
$$\sum_{i=1}^{n} C_i/T_i \leq n\left(2^{1/n} - 1\right)$$

- For the given task set, we have

$$U = \frac{1}{4} + \frac{2}{6} + \frac{3}{8} = 0.958 \nleq 3\left(2^{\frac{1}{3}} - 1\right) = 0.779$$

- Thus, the sufficient RM schedulability test failed.

Task 4 (b): Sample Solution



Task 5 (a): Sample Solution

- A set of n periodic real-time tasks, where $D_i = T_i$ for all tasks τ_i, is schedulable using EDF if and only if

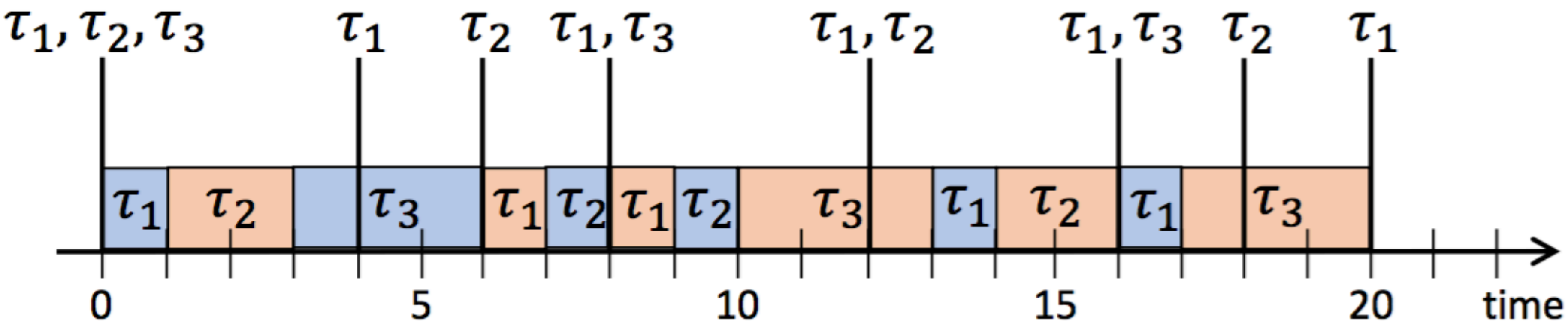
$$\sum_{i=1}^{n} C_i / T_i \leq 1$$

- For the given task set, we have

$$U = \frac{1}{4} + \frac{2}{6} + \frac{3}{8} = 0.958 \leq 1$$

- Thus, EDF definitely meets all deadlines.

Task 5 (b): Sample Solution



Note that, for example, at time $t = 4$ when the second instance of task τ_1 arrives, the running task τ_3 is *not* preempted, because both tasks have the same priority.

Task 5 (c): Sample Solution

- Schedulability test based on execution times and periods does not say which tasks are going to miss their deadline.
- Depending on the concrete arrival times (*i.e.*, phases), different tasks may miss deadlines.