SCHEDULING APERIODIC TASKS

STEFANO DI CARLO

APERIODIC TASKS

- Infinite sequence of identical instances
- Their activations are not regularly interleaved
- Scheduling algorithm defined using α|β|γ
  - **α** describes the <u>machine environment</u> on which the task set must be scheduled (uniprocessor, multiprocessor, distributed architecture, and so on)
  - β describes <u>task and resource characteristics</u> (preemptive, synchronous activations, and so on)
  - γ indicates the <u>optimality criterion</u> (performance measure) to be followed in the schedule

EXAMPLES

- 3 | no preem | Σf_i problem of scheduling a set of tasks on a three-processor machine. Preemption is not allowed, and the objective is to minimize the sum of the finishing times. Since no other constraints are indicated in the second field, tasks have arbitrary arrival times

- 2 | sync | $\Sigma Late_i$ problem of scheduling a set of tasks on a two-processor machine. Tasks have synchronous arrival times. The objective is to minimize the number of late tasks

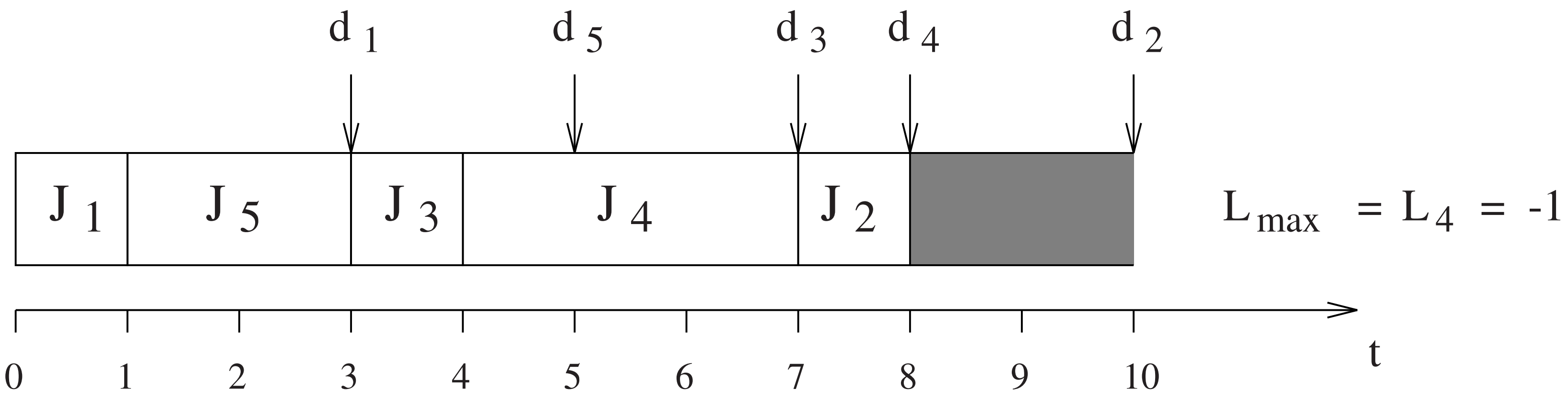
JACKSON'S ALGORITHM

- Problem: 1 | sync | Lmax
  - One processor, tasks with synchronous arrival time (can have different WCET and deadline)
  - Objective is to minimize the maximum lateness
- Observation:
  - All tasks arrive at the same time
  - No need for preemption
- Earliest Due Date (EDD)
  - Given a set of n independent tasks, any algorithm that executes the tasks in order of non decreasing deadlines is optimal with respect to minimizing the maximum lateness

EXAMPLE 1

- Arrival time = 0

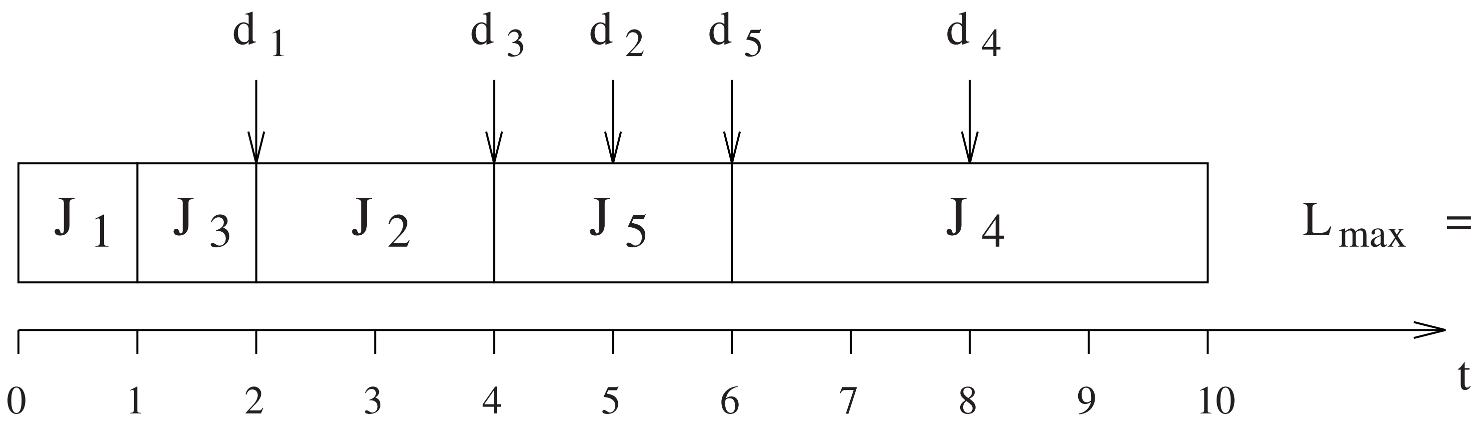
	J_1	J_2	J_3	J_4	J_5
C_i	1	1	1	3	2
d_i	3	10	7	8	5



EXAMPLE 2

- Arrival time = 0

	J_1	J_2	J_3	J_4	J_5
C_i	1	2	1	4	2
d_i	2	5	4	8	6



$L_{max} = L_4 = 2$

HORN'S ALGORITHM

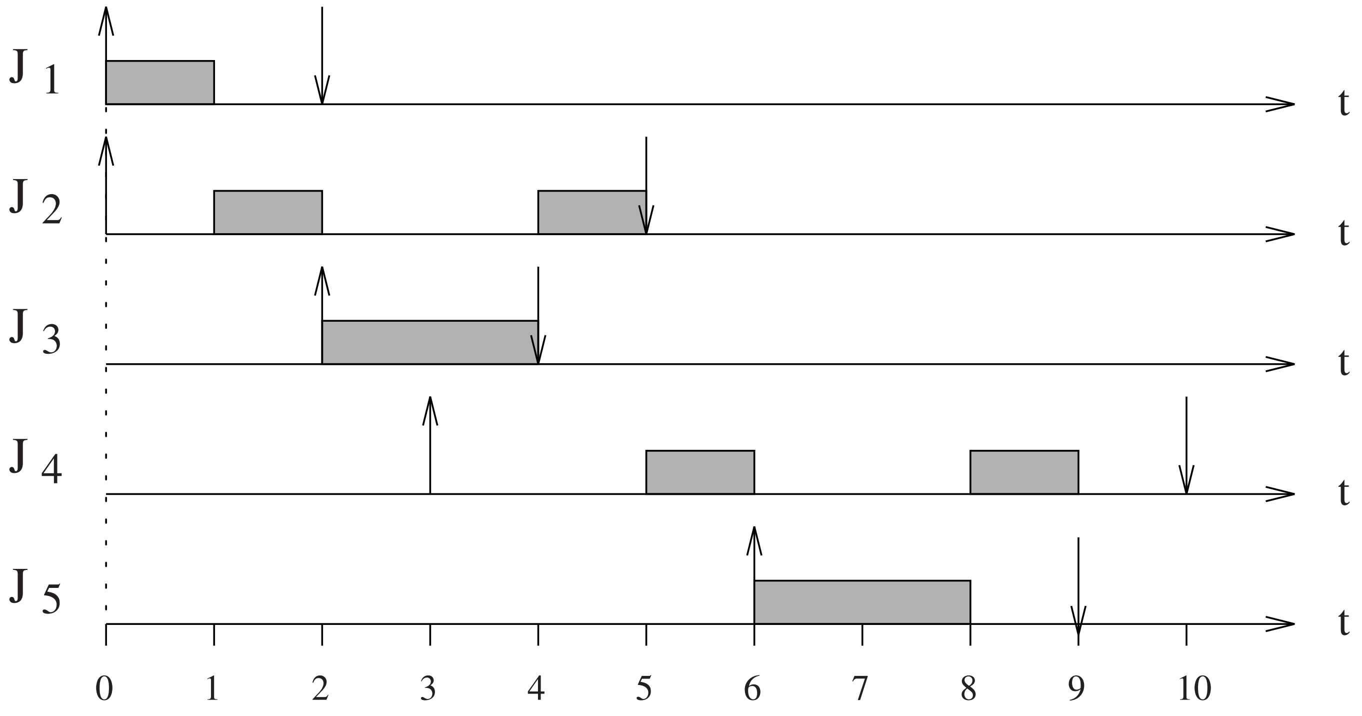
- Problem: 1 | preem| Lmax
  - One processor, tasks with asynchronous arrival time (can have different WCET and deadline)
  - Objective is to minimize the maximum lateness

- Observation:
  - Preemption

- Earliest Deadline First (EDF)
  - Given a set of n independent tasks with arbitrary arrival times, any algorithm that at any instant executes the task with the earliest absolute deadline among all the ready tasks is optimal with respect to minimizing the maximum lateness

EXAMPLE

	J_1	J_2	J_3	J_4	J_5
a_i	0	0	2	3	6
C_i	1	2	2	2	2
d_i	2	5	4	10	9

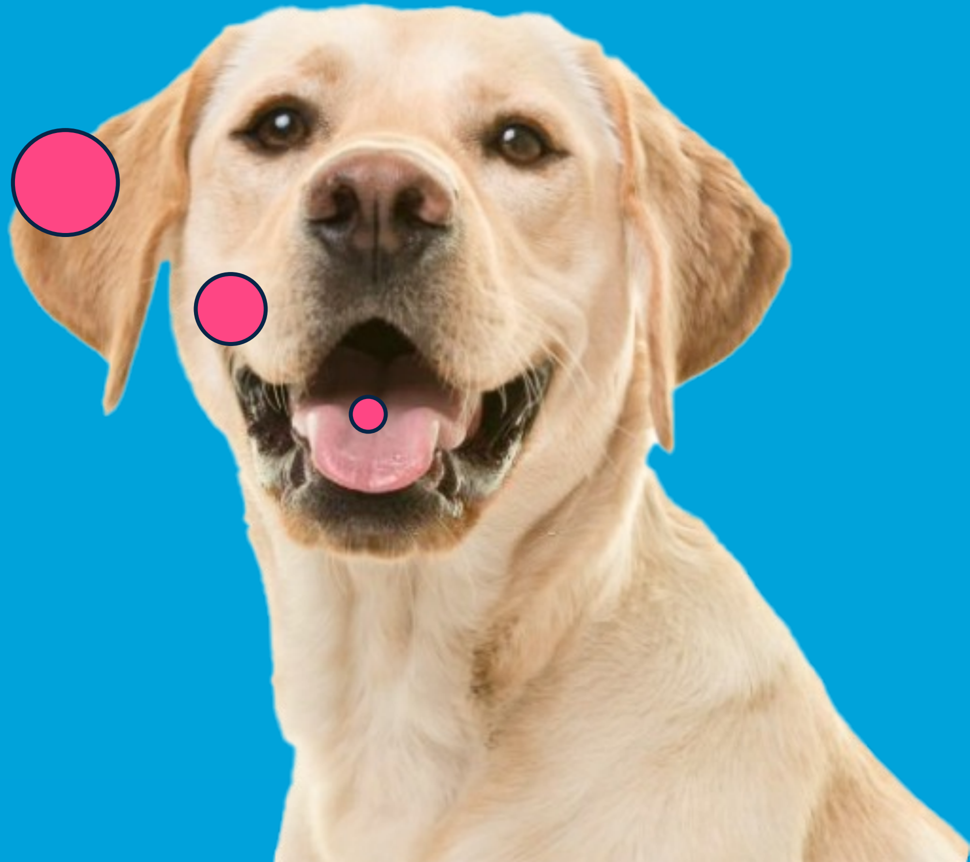


smilies

reSilient coMputer archItectures and LIfE Sciences

Politecnico di Torino
1859
Department of Control and Computer Engineering
DAUIN

QUESTIONS?



THANK YOU!