Scheduling aperiodic tasks

Date	@October 24, 2024

Aperiodic tasks

Aperiodic tasks are a infinite sequence of identical instances and their activations are not regularly interleaved. Scheduling algorithms are defined as $\alpha|\beta|\gamma$, where:

- α describes the machine environment on which the task set must be scheduled (uniprocessor, multiprocessor, and so on)
- β describes task and resource characteristics (preemptive, synchornous activations, and so on)
- γ indicates the optimality criterion to be followed in the schedule

For example:

- 3 | no preemp | $\sum f_i$ asks for scheduling a set of tasks on a three-processor machine, without using preemption and trying to minimize the sum of the finishing times.
- 2 | sync | $\sum Late_i$ asks for scheduling a set of tasks on a two-processor machine, considering tasks with synchronous arrival time, trying to minimize the number of late tasks.

Jackson's algorithm

The problem of Jackson's algorithm is 1 | sync | L_{max}, it means that is a scheduling problem on a one processor machine with synchronous arrival time tasks and the aim is to minimize the maximum lateness.

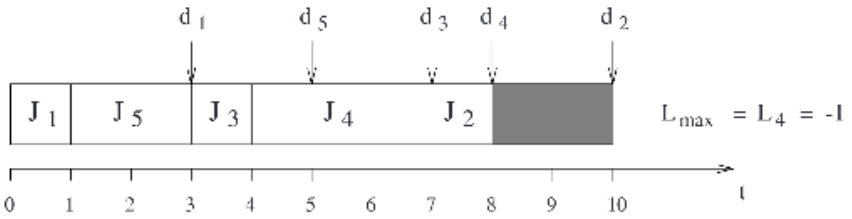
In this problem all the tasks arrive at the same time and there's no need of preemption.

The Earliest Due Date (EDD) theorem says that 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.

For example:

- 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



The lateness of a task is measured computing [time of deadline] - [end time] for each task.

Horn's algorithm

The problem of Horn's algorithm is 1 | preem | L_{max}, it means that is a scheduling problem on a one processor machine with synchronous arrival time tasks and the aim is to minimize the maximum lateness.

In this problem all the tasks arrive at the same time and there is preemption.

The Earliest Due Date (EDD) theorem says that 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.

For 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