Theorem 1. *A system T of independent, preemptable tasks with relative deadlines equal to their respective periods can be feasibly scheduled on one processor if and only if its total utilization is equal to or less than 1.*

Proof. That the system is not feasible if its total utilization is larger than 1 is obvious so we focus on the *if* part of the proof. Since the EDF algorithm is optimal in the sense that it can surely produce a feasible schedule of any feasible system, it suffice for us to prove that the EDF algorithm can surely produce a feasible schedule of any system with a total utilization equal to 1. We prove this statement by showing that if according to an EDF schedule, the system fails to meet some deadlines, then its total utilization is larger than 1. To do so, let us suppose that the system begins to execute at time 0 and at time t, the job $J_{i,c}$ of task T_i misses its deadline.

For the moment, we assume that priori to t the processor never idles. We will remove this assumption at the end of the proof. There are two cases to consider: (1) The current period of every task begins at or after $r_{i,c}$, the release time of the job that misses its deadline, and (2) the current periods of some tasks begin before $r_{i,c}$. The two cases are illustrated in Figure 1. In this figure, we see that the current jobs of all tasks T_k, for all $k \neq i$, have equal of lower priorities than $J_{i,c}$ because their deadlines are at or after t.

Case (1): This case is illustrated by the time lines in Figure 1.(a); each tick on the time line of a task shows the release time of some job in the task. The fact that $J_{i,c}$ misses its deadline at t tells us that any current job whose deadline is after t is not given any processor time to execute before t and that the total processor time required to complete $J_{i,c}$ and all jobs with deadlines at or before t exceeds the total available time t. In other words,

$$t < \frac{(t - \phi_i)e_i}{p_i} + \sum_{k \neq i} \left\lfloor \frac{t - \phi_k}{p_k} \right\rfloor e_k \tag{1}$$

(You recall that ϕ_k, p_k and e_k are the phase, period, and execution time of task T_k, respectively.) $\lfloor x \rfloor$ ($x \geq 0$) denotes the largest integer less than or equal to x. The first term on the right-hand side of the inequality is the time required to complete all the jobs in T_i with deadlines before t and the job $J_{i,c}$. Each term in the sum gives the total amount of time before t required to complete jobs that are in a task T_k other than T_i and have deadlines at or before t. Since $\phi_k \geq 0$ and $e_k/p_k = u_k$ for all k, and $\lfloor x \rfloor \leq x$ for any $x \geq 0$,

$$\frac{(t - \phi_i)e_i}{p_i} + \sum_{k \neq i} \left\lfloor \frac{t - \phi_k}{p_k} \right\rfloor e_k \leq t\frac{e_i}{p_i} + t\sum_{k \neq i} \frac{e_k}{p_k} = t\sum_{k=1}^{n} u_k = tU$$

Combining this inequality with the one in Equation 1, we have $U > 1$.

Case (2): The time lines in Figure 1.(b) illustrate case (2). Let $\mathbf{T}'$ denote the subset of $\mathbf{T}$ containing all the tasks whose current jobs were released before

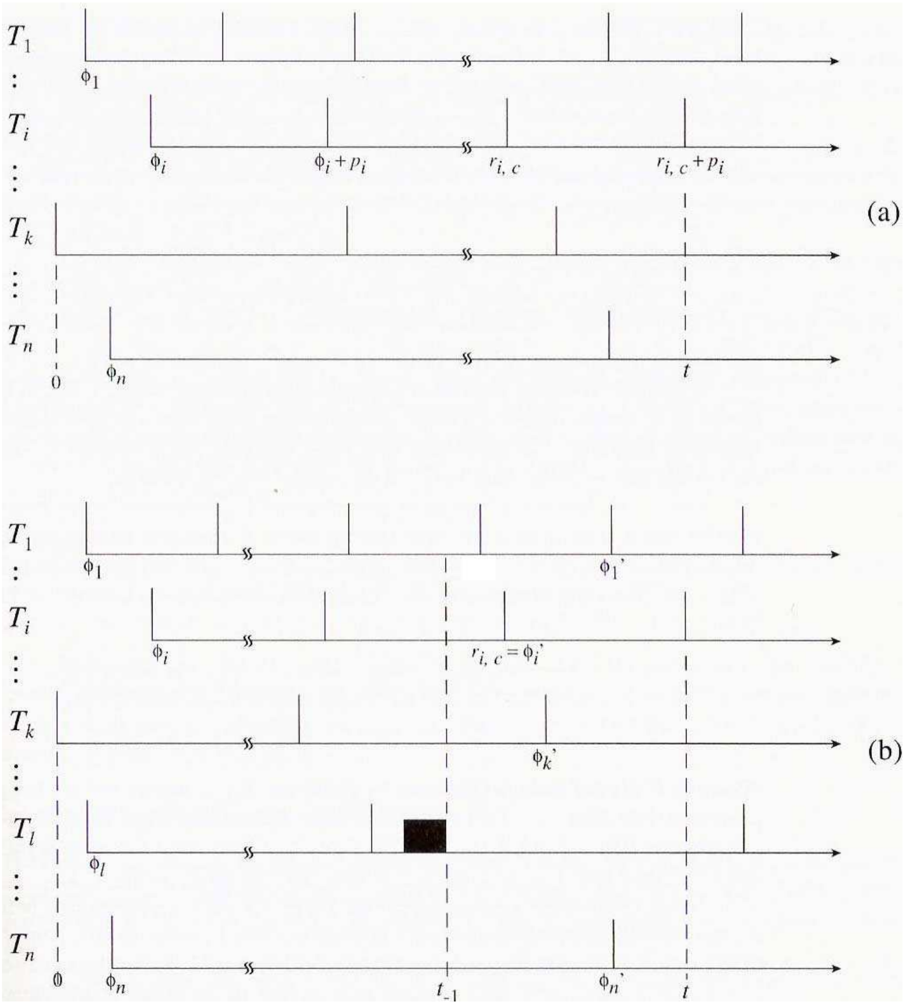

Figure 1: Infeasible EDF schedules.

$r_{i,c}$ and have deadlines after t. It is possible that some processor time before $r_{i,c}$ was given to the current jobs of some tasks in $\mathbf{T}'$. In the figure, T_l is such a task.

Let t_{-1} be the end of the latest time interval I (shown as a black box in Figure 1.(b)) that is used to execute some current job in $\mathbf{T}'$. We now look at the segment of the schedule starting from t_{-1}. In this segment, none of the current jobs with deadlines after t are given any processor time. Let ϕ_k' denote the release time of the first job of task T_k in $\mathbf{T} - \mathbf{T}'$ in this segment. Because $J_{i,c}$ misses its deadline at t, we must have

$$t - t_{-1} < \frac{(t - t_{-1} - \phi_i')e_i}{p_i} + \sum_{T_k \in \mathbf{T} - \mathbf{T}'} \left\lfloor \frac{t - t_{-1} - \phi_k'}{p_k} \right\rfloor e_k$$

(Tasks in $\mathbf{T}'$ are not included in the sum; by definition of t_{-1}, these tasks are not given any processor time after t_{-1}.) This inequality is the same as the one in Equation 1 except that t is replaced by $t - t_{-1}$ and ϕ_k is replaced by ϕ'_k. We can use the same argument used above to prove that $\sum_{T_k \in \mathbf{T} - \mathbf{T}'} u_k > 1$, which in turn implies that $U > 1$.

Now we consider the case where the processor idles for some time before t. Let t_{-2} be the latest instant at which the processor idles. In other words, from t_{-2} to the time t when $J_{i,c}$ misses its deadline, the processor never idles. For the same reason that Equation 1 is true, the total time required to complete all the jobs that are released at and after t_{-2} and must be completed by t exceeds the total available time $t - t_{-2}$. In other words, $U > 1$.

$\square$