that if deadlines are ever missed, the time of the earliest missed deadline will have a known upper bound. Then, we only need to check feasibility up to that point.

Just as with the RM algorithm, it is easy to show that the worst-case execution time of a task occurs when all task phasings are zero. So, if we verify schedulability for this case, it will hold for all task phasings.

The following is the finiteness result. We define \( u = \sum_{i=1}^{n} (e_i / P_i) \), \( d_{max} = \max_{1 \leq i \leq n} \{ d_i \} \), and \( P = \operatorname{lcm}(P_1, \ldots, P_n) \). Define \( h_T(t) \) to be the sum of the execution times of all tasks in set \( T \) whose absolute deadlines are less than, or equal to, \( t \).

**Theorem 12** A task set of \( n \) tasks is not EDF-feasible iff

- \( u > 1 \), or

- there exists

\[
    t < \min \left\{ P + d_{max}, \quad \frac{u}{1 - u} \max_{1 \leq i \leq n} \{ P_i - d_i \} \right\}
\]

such that

\[
    h_T(t) > t.
\]

Under this theorem, we only need to check for feasibility up to some finite time. We can build up the proof of Theorem 12 as the following series of lemmas.

**Lemma 9** A given set, \( T \), of periodic tasks, is not EDF-schedulable iff there exists some time \( t \) such that \( h_T(t) > t \).

**Proof:** Left to the reader.

**Lemma 10** Given a set, \( T \), of \( n \) periodic tasks, if \( u \leq 1 \),

\[
    h_T(t + P) > t + P \Rightarrow h_T(t) > t \quad \text{for all } t \geq d_{max}.
\]

**Proof:**

\[
    h_T(t) + P = \sum_{i=1}^{n} e_i \left( \left\lfloor \frac{t-d_i}{P_i} \right\rfloor + 1 \right) + P
\]
\[ \begin{align*}
& \geq \sum_{i=1}^{n} e_i \left( \frac{t - d_i}{P_i} \right) + 1 + P \sum_{i=1}^{n} e_i \\
& = \sum_{i=1}^{n} e_i \left( \frac{t - d_i + P}{P_i} \right) + 1 \quad \text{since } P \text{ is a multiple of } P_i \\
& = h_T(t + P)
\end{align*} \]

Hence,

\[ h_T(t + P) > t + P \Rightarrow h_T(t) > t \quad \text{Q.E.D.} \quad (3.54) \]

**Lemma 11** If task set \( T \) is not EDF-feasible and \( u \leq 1 \), then there exists \( t < P + d_{max} \) such that \( h_T(t) > t \).

**Proof:** Follows immediately from Lemma 10. Q.E.D.

**Lemma 12** Suppose \( T \) is not feasible, and \( u \leq 1 \). Then, \( h_T(t) > t \) implies

\[ t < d_{max} \quad \text{or} \quad t < \max_{1 \leq i \leq n} \{ P_i - d_i \} \frac{u}{1-u}. \]

**Proof:** Suppose that \( t > d_{max} \). We have

\[ \begin{align*}
h_T(t) & \leq \sum_{i=1}^{n} e_i \frac{t - d_i + P_i}{P_i} \\
& = t \sum_{i=1}^{n} \frac{e_i}{P_i} + \sum_{i=1}^{n} \frac{P_i - d_i}{P_i} \\
& \leq \sum_{i=1}^{n} \left[ \frac{e_i}{P_i} \left( t + \max_{1 \leq i \leq n} \{ P_i - d_i \} \right) \right] \quad (3.55)
\end{align*} \]

Hence, if \( h_T(t) > t \), we will have from (3.55),

\[ t < \sum_{i=1}^{n} \frac{e_i}{P_i} \left( t + \max_{1 \leq i \leq n} \{ P_i - d_i \} \right) \]

\[ \Rightarrow t < \max_{1 \leq i \leq n} \{ P_i - d_i \} \frac{u}{1-u} \quad \text{Q.E.D.} \quad (3.56) \]

### 3.2.3 Allowing for Precedence and Exclusion Conditions

We have assumed in the above sections that tasks are independent, and are always preemptible by other tasks. We will now relax both these assumptions, and present several scheduling heuristics.