Push Forward: Global Fixed-Priority Scheduling of Arbitrary-Deadline Sporadic Task Systems — Efficient Implementation of Theorem 4.4 —

Jian-Jia Chen[1], Georg von der Brüggen[2], and Niklas Ueter[3]

1 **TU Dortmund University, Germany**
jian-jian.chen@tu-dortmund.de
2 **TU Dortmund University, Germany**
georg.von-der-brueggen@tu-dortmund.de
3 **TU Dortmund University, Germany**
niklas.ueter@tu-dortmund.de

Abstract

This document provides the detailed discussions on how to implement a schedulability test in Theorem 4.4 in the paper.

▶ **Theorem 4.4.** *Task τ_k is schedulable by the given global fixed-priority scheduling if*

$$\forall \ell \in \mathbb{N}, \exists 1 \geq \rho \geq \ell C_k/((\ell-1)T_k + D_k)$$
$$\frac{\ell C_k}{D'_k} + \sum_{\tau_i \in \mathbf{T}^{carry-approx}} \frac{\gamma_i U_i D_i}{D'_k} + \sum_{i=1}^{k-1} \left(\frac{C_i - C_i U_i}{D'_k} + U_i \right) \leq \mu_k, \tag{1}$$

where $\mu_k = M - (M-1)\rho$ with $1 \geq \rho \geq \ell C_k/((\ell-1)T_k + D_k)$, D'_k is $(\ell-1)T_k + D_k$,

$$\gamma_i = \begin{cases} 1 & \text{if } U_i > \rho \\ 0 & \text{if } U_i \leq \rho \end{cases} \tag{2}$$

and $\mathbf{T}^{carry-approx}$ is the set of the $\lceil \mu_k \rceil - 1$ tasks among the $k-1$ higher-priority tasks with the largest values of $\gamma_i U_i D_i$. Note that $|\mathbf{T}^{carry-approx}|$ can be smaller than $\lceil \mu_k \rceil - 1$ if the number of tasks with $U_i > \rho$ is less than $\lceil \mu_k \rceil - 1$. If $D_k \leq T_k$, we only need to consider $\ell = 1$. ◻

In Theorem 4.4 we need to find a ρ with $1 \geq \rho \geq \ell C_k/((\ell-1)T_k + D_k)$ for each $\ell \in \mathbb{N}$ such that

$$\frac{\ell C_k}{D'_k} + \sum_{\tau_i \in \mathbf{T}^{carry-approx}} \frac{\gamma_i U_i D_i}{D'_k} + \sum_{i=1}^{k-1} \left(\frac{C_i - C_i U_i}{D'_k} + U_i \right) \leq \mu_k \tag{1}$$

holds, where $\mu_k = M - (M-1)\rho$ and $D'_k = (\ell-1)T_k + D_k$. Furthermore $\gamma_i = 1$ if $U_i > \rho$ and $\gamma_i = 0$ if $U_i \leq \rho$. As C_i, D_i, T_i, and U_i are given for all tasks, whether Eq. (1) holds or not only depends on the values of ρ (and hence μ_k), ℓ, and $\mathbf{T}^{carry-approx}$. Let $\sum_{\tau_i \in \mathbf{T}^{carry-approx}} \frac{\gamma_i U_i D_i}{D'_k}$ be denoted as $G(\mu)$. Note that $\mathbf{T}^{carry-approx}$ depends on ρ and hence μ_k. If we assume μ_k and hence $G(\mu)$ to be a constant, the left hand side of Eq. (1), denoted as $F(\ell, \mu)$, is either an increasing or a non-increasing function with respect to ℓ, i.e., maximized either for $\ell = 1$ or $\ell \to \infty$. We will use ∞ as a value here for notational brevity, where $F(\infty, \mu)$ is the limit of the function $F(\ell, \mu)$ when $\ell \to \infty$. Knowing this, for a given μ_k we have the following cases:

- $F(\ell, \mu)$ is increasing with respect to ℓ
  - $F(1, \mu) > \mu_k \Rightarrow$ Eq. (1) never holds.
  - $F(\infty, \mu) \leq \mu_k \Rightarrow$ Eq. (1) always holds.
  - $F(1, \mu) \leq \mu_k$ and $F(\infty, \mu) > \mu_k$. This means, we can calculate a value $l_{\mu_k} \in \mathbb{R}$ with $F(l_{\mu_k}, \mu) = \mu_k \Rightarrow$ Eq. (1) holds for $1 \leq \ell \leq l_{\mu_k}$ but not for $\ell > l_{\mu_k}$.
- $F(\ell, \mu)$ is not increasing with respect to ℓ
  - $F(1, \mu) \leq \mu_k \Rightarrow$ Eq. (1) always holds.
  - $F(\infty, \mu) > \mu_k \Rightarrow$ Eq. (1) never holds.
  - $F(1, \mu) > \mu_k$ and $F(\infty, \mu) \leq \mu_k$. This means, we can calculate a value $l_{\mu_k} \in \mathbb{R}$ with $F(l_{\mu_k}, \mu) = \mu_k \Rightarrow$ Eq. (1) holds for $l_{\mu_k} \leq \ell$ but not for $\ell < l_{\mu_k}$.

As a result, for a given ρ we can calculate the interval where Eq. (1) holds by calculating the values for $\ell = 1$, $\ell = \infty$, and $l_{\mu_k} \in \mathbb{R}$ with $F(l_{\mu_k}, \mu) = \mu_k$. Note that this interval must be further reduced due to the condition $1 \geq \rho \geq \ell C_k/((\ell - 1)T_k + D_k)$, i.e., some (or all) values of ℓ are not allowed for a given μ_k. However, when each $\ell \in \mathbb{N}$ is covered by at least one interval, the task is schedulable according to Theorem 4.4. By testing only a finite number of μ_k values, we can implement the schedulability condition in Theorem 4.4 efficiently.

When we only look at the right hand side of Eq. (1), we would want to reduce ρ as much as possible to get the largest possible value for μ_k, thus making the condition easier. However, increasing μ_k will lead to a larger value of $G(\mu)$, i.e., a bigger left hand side. This happens either due to an additional summand in the summation or due to new tasks available to be summed up, i.e, the reduction of ρ leads to $U_i > \rho$. For the number of summands, due to the ceiling function, we only have to test integer values of μ_k, as they maximize the right hand side of Eq. (1) for a given number of summands. As $\mu_k = M - (M - 1)\rho$, the number of integer values for μ_k is bounded by the number of processors, i.e., we only have to test a finite number of ρ values to cover the situation where the number of summands in $G(\mu)$ increases. In addition, only tasks τ_i with $U_i > \rho$ are allowed in $G(\mu)$. As ρ gets smaller, the number of tasks with $U_i > \rho$ increases and vice versa. However, if we test all values with $U_i = \rho$, where τ_i that has higher priority than τ_k, in an increasing order, we only have to test a finite number of additional ρ values, depending on the number of tasks.

Therefore, we only have to test those $O(M + k)$ possible ρ values. As discussed above, each of them forms an interval I_ρ of the integer values of ℓ that can be covered by the specified ρ value. For each interval $I_\rho = [left_\rho, right_\rho)$, Eq. (1) holds when $\ell = left_\rho, left_\rho + 1, \ldots, right_\rho - 1$, where $left_\rho$ is an positive integer and $right_\rho$ is either positive integer or ∞. Note that such an interval I_ρ does not exist if Eq. (1) never holds, and such ρ values are discarded from further considerations. Deriving all these *valid* intervals needs $O(M + k)$ time in the amortized manner, provided that the higher-priority tasks are sorted by their utilization in $O(k \log k)$ and stored in a list in advance. We need to pay some attention if an interval I_ρ does not have a limited upper bound, called an *unbounded interval* here, i.e., Eq. (1) holds for any $\ell \geq left_\rho$. Note that we do need the existence of at least such an unbounded interval to cover sufficiently large ℓ. Among the unbounded intervals, we take the minimum left endpoint, called ℓ_{max}. This step takes $O(M + k)$. The remaining intervals I_ρ that are not unbounded are called *bounded intervals*. Verifying whether $\ell = 1, 2, \ldots, \ell_{max} - 1$ are covered can be done by checking whether the union of these bounded intervals provides the coverage, which is achievable in $O((M + k) \log(M + k))$ with Klee's algorithm [1].

Due to the above discussions, we can efficiently implement the schedulability test in Theorem 4.4 with a time complexity of $O((M + k) \log(M + k))$.

References

1 V. Klee. Can the measure of $\cup_{i=1}^{n}[a_i, b_i]$ be computed in less than $O(nlogn)$ steps? *The American Mathematical Monthly*, 84(4):284–285, 1977.