Bandwidth Allocation for Fixed-Priority-Scheduled Compositional Real-Time Systems

FARHANA DEWAN* NATHAN FISHER

Abstract

Recent research in compositional real-time systems has focused on determination of a component’s real-time interface parameters. An important objective in interface-parameter determination is minimizing the bandwidth allocated to each component of the system while simultaneously guaranteeing component schedulability. With this goal in mind, in this paper we explore fixed-priority schedulability in compositional setting. First we derive an efficient exact test based on iterative convergence for sporadic task systems scheduled by fixed priority (e.g., deadline monotonic, rate monotonic) upon an Explicit-Deadline Periodic (EDP) resource. Then we address the time complexity of the exact test by developing a fully-polynomial-time approximation scheme (FPTAS) for allocating bandwidth to components. Our parametric algorithm takes the task system and an accuracy parameter $\epsilon > 0$ as input, and returns a bandwidth which is guaranteed to be at most a factor $(1 + \epsilon)$ times the optimal minimum bandwidth required to successfully schedule the task system. We perform thorough simulation over synthetically generated task systems to compare the performance of our proposed efficient-exact and the approximate algorithm and observe a significant decrease in runtime and a very small relative error when comparing the approximate algorithm with the exact algorithm and the sufficient algorithm.

1 Introduction

Recent research in real-time systems has focused on designing frameworks for enabling component-based design in real-time systems. State-of-the-art frameworks for compositional real-time systems include [10, 15, 31, 1]. Component-based design is highly desirable due to its well-known benefits of reducing overall system complexity and enhancing system designers’ understanding of the system. One of the major benefits of these systems is achieved by the goal of component abstraction, which hides the internal complexity and details of one component from developers of other components and only exposes information necessary to use the component via an interface. In most compositional real-time frameworks, a component uses a real-time interface to communicate with the other components of the system. The major difference of such interface with traditional functional interfaces for softwares or modules is the fact that the real-time interfaces represent the overall timing constraint of the component along with computational resource requirements. A component specifies its resource requirements to meet its real-time constraints by the attribute interface bandwidth. Thus, an important design issue of these compositional frameworks is addressing the problem minimization of interface bandwidth (MIB-RT).

One simple, yet flexible, real-time compositional framework is the explicit-deadline periodic resource (EDP) model [14, 31]. An EDP resource $\Omega$ is characterized by a three-tuple $(\Pi, \Theta, \Delta)$ where $\Pi$ is referred to as the period of repetition, $\Theta$ is the capacity, and $\Delta$ is the relative deadline. The interpretation of such a resource is that a component $C$ executed upon $\Omega$ is guaranteed $\Theta$ units of processing resource supply for successive $\Pi$-length intervals (given some initial starting time). Furthermore, the $\Theta$ units of resource supply must be provided within $\Delta$ ($\leq \Pi$) time units after the start of the $\Pi$-length interval. In compositional systems, since more than one component share the processing resource (single processor in our case), when one component receives the processing supply, all the remaining components have “no-supply period” at that instant of time. The no-supply period of $\Omega$ is the duration of

*F. Dewan and N. Fisher are with Department of Computer Science, Wayne State University in Detroit, MI, USA. Corresponding Author’s Email: farhanad@wayne.edu.
time at which the component $C$ does not receive any processing resource; more formally, $\Pi - \Theta$ units of time in any $\Pi$-interval of time in this case. The interface bandwidth of $C$ for this framework is the ratio of capacity and period ($\frac{\Theta}{\Pi}$). A system-level scheduling algorithm allocates the processor time among the different periodic resources that share the same processor, such that each resource receives (for every period) aggregate processor time equivalent to its capacity. A component’s tasks are then hierarchically scheduled by a component-level scheduling algorithm upon the processing time supplied to resource $\Omega$.

In this paper, we obtain solutions to MIB-RT for an EDP resource when components use fixed-priority as the component-level scheduling algorithm. (The system-level scheduling algorithm is not considered for this paper). Specifically, we consider the problem of determining the optimal choice of capacity parameter (i.e., $\Theta$) for an EDP resource $\Omega$ with a fixed period $\Pi$ and deadline $\Delta$ for component $C$. Algorithms exist for determining $\Pi$ and $\Delta$ by searching over possible values and using a capacity-determination algorithm as a subroutine (e.g., see [13, 16]); thus, since the search space may be quite large, efficient capacity-determination algorithms are necessary.

The MIB-RT problem for fixed-priority periodic resource model has been previously studied. An exact solution, based on exact schedulability techniques for uniprocessor real-time systems ([6, 20]) has been proposed by Easwaran et al. [14]; however, the proposed solution has pseudo-polynomial-time complexity. There is also a $O(n)$-time sufficient solution to MIB-RT for periodic resource ($\Pi$ equals $\Delta$) by Shin and Lee [31]. The exact resource allocation is computationally expensive (pseudo-polynomial in this case) and thus might be impractical for algorithms that search for optimal values of $\Pi$ and $\Delta$. On the other hand, though the sufficient resource allocation has lower (linear) computational complexity, these algorithms might provide over-estimated resource allocations and induce lower system utilization. This might be impractical for developing real-time systems in which resources are very scarce. However, in many real-time systems where tasks may be added or removed dynamically, it is important to provision resources efficiently at run-time and an efficient allocation algorithm is desirable. Our goal is to design an algorithm which is computationally efficient on real-time guarantee verification as well as to provide the system designer control over accuracy of resource allocation.

In the above setting, existing schedulability results for dedicated uniprocessor system [20, 17] can be applied to obtain schedulability test for fixed-priority systems scheduled upon periodic-resources. A component consisting of sporadic tasks [25] and scheduled upon periodic resource can be converted to an equivalent task system scheduled upon a dedicated resource by modeling the “no-supply period” of the periodic resource model as a special highest priority task [26]. However, the exact solution obtained in this approach is highly inefficient computationally, which makes this test impractical for real-time open system environment where application interface requires to change dynamically.

In our prior work [18], we devised approximate bandwidth allocation algorithm for EDP resource with component level scheduling algorithm EDF. In this paper we extend those results for fixed-priority scheduled components. However, the compositional results for EDF does not directly apply for fixed-priority, as we have to do maximum response time analysis for each task; this fundamentally differs from the demand-based approach of [18].

§Our Contribution. For EDP resource model with sporadic tasks [25] as components, first we derive a response time based exact schedulability condition similar to dedicated uniprocessor scheduling by considering the “no-supply period” of the EDP resource as special higher priority tasks. We derive heuristics (i.e., lower bound and upper bound for response time) to efficiently perform the schedulability test similar to [9]. A major contribution of this article is the development of a parametric approximation algorithm that addresses the current gap between computationally-expensive, exact solutions and computationally-inexpensive, sufficient solutions for MIB-RT problem:

Given $\Pi$, $\Delta$, task system $\tau$, and accuracy parameter $\epsilon > 0$, let $\Theta^*(\Pi, \Delta, \tau)$ be the optimal minimum capacity for $\tau$ to be fixed-priority-schedulable upon EDP resource $\Omega^* = (\Pi, \Theta^*(\Pi, \Delta, \tau))$. Our algorithm returns $\hat{\Theta}$ for the given parameters where $\Theta^*(\Pi, \Delta, \tau) \leq \hat{\Theta} \leq (1 + \epsilon) \cdot \Theta^*(\Pi, \Delta, \tau)$. Furthermore, the time complexity of our algorithm is polynomial in the number of tasks in $\tau$ and $\frac{1}{\epsilon}$.

In other words, our algorithm is a fully-polynomial-time approximation scheme (FPTAS) for the MIB-RT problem with approximation ratio $(1 + \epsilon)$. This implies that the system designer can pre-specify an arbitrary level of accuracy in obtaining solution to MIB-RT with the tunable algorithm. The second major contribution of this article is a comprehensive comparison of our approximate approach with the previously-existing exact and sufficient algorithms by means of simulation over randomly generated task systems. One application of our approximation scheme is in
thermally constrained real-time systems, where power-aware components dynamically tune the interface [19] to meet the temporal and thermal constrains.

Organization. The remainder of the paper is organized as follows. In Section 2, we briefly review the current literature on compositional real-time frameworks and MIB-RT problem for fixed-priority scheduling. In Section 3, we provide necessary notations required for the rest of the paper. For our capacity determination problem, we first derive an efficient exact schedulability test based on response time for fixed-priority scheduled components upon EDP resource in Section 4. In Section 5, we present an approximate algorithm for this setting based on testing set points (defined in Section 3), and prove its correctness. Then, we give the approximation ratio results for the proposed algorithm in Section 6. Simulation results comparing our algorithm with both previously-known exact and sufficient algorithms are given in Section 7. Finally, we conclude with discussion and future direction of this research in Section 8.

2 Related Work

The concept of compositional systems was first introduced by Deng and Liu [10] in their work real-time open environments and Rajkumar et al. [27] in their work resource kernels. Since then, researchers have proposed many different real-time compositional models and studied the MIB-RT problem of the proposed models. Feng and Mok [15] proposed the concept of temporal partitions to support hierarchical sharing of a processing resource. Shin et al. [29] proposed the related periodic resource model to characterize the supply guaranteed to any component in compositional system, which was generalized by Easwaran et al. [14] to explicit deadline periodic (EDP) resource model.

The resource allocation for fixed-priority scheduled periodic resource model has been previously studied for constrained deadline case (i.e., task deadline is less or equal task period). An exact solution based on exact schedulability techniques for uniprocessor real-time systems has been proposed by Easwaran et al. [14] for EDP resource. [31] proposed an $O(n)$-time sufficient solution for periodic resource (II equals $\Delta$). For the temporal partition models where components are scheduled by fixed-priority, [23] proposed exact, pseudo-polynomial time algorithm for resource allocation, and [3] proposed sufficient, polynomial-time resource allocation techniques.

The schedulability analysis of fixed-priority-scheduled components upon periodic resources can be compared to existing fixed-priority servers [21, 32, 33, 34] used to serve the aperiodic jobs in the system. When system consists of both periodic and aperiodic jobs, the periodic jobs require fraction of resource proportional to their utilization. A fixed-priority server reserves a fraction of resource for the upcoming aperiodic jobs, and serves them whenever such job arrives in the system depending on the server “budget”. Several fixed-priority servers have been proposed in the literature to schedule aperiodic and periodic jobs in system, for example, periodic server, polling server, deferrable server [21, 34], sporadic server [33] etc.

Traditional uniprocessor schedulability analysis for sporadic task system with dedicated resource can be applied in compositional setting to obtain a solution to MIB-RT, by modeling the “no-supply period” of the periodic resource model as special tasks (similar to the aperiodic server job) with higher priority than all the other tasks in the system. Using this approach, in [26], Okwudire et al. gave a linear-time sufficient schedulability test for fixed-priority scheduled component with harmonic task periods for two-level hierarchical system. Furthermore, response time bounds for such system can be derived similar to [9] to obtain efficient exact schedulability test.

Thus, for a variety of models, relatively efficient, sufficient algorithms for MIB-RT have been proposed; however, the existence of any work on obtaining polynomial-time algorithms (prior to our work) with constant-factor approximation ratios where components are scheduled by fixed-priority is unknown. Although research by [1] has developed parametric algorithms, without known approximation ratios, for the hierarchical event stream model. The goal of our work is to fill this needed gap by obtaining an FPTAS for MIB-RT in the periodic resource model with uniprocessor platforms. In our preliminary work [18], we obtained such ratios for the periodic resource model where components are scheduled by dynamic-priority (EDF). In this paper, our aim is to extend those results by developing an FPTAS for EDP framework where components are scheduled by fixed-priority (deadline-monotonic or rate-monotonic). The algorithm of [16] may be used in conjunction with the results of this paper to find an optimal period.
§Sporadic Task Model. A sporadic task $\tau_i = (e_i, d_i, p_i)$ is characterized by a worst-case execution requirement $e_i$, a (relative) deadline $d_i$, and a minimum inter-arrival separation $p_i$. Such a sporadic task generates a potentially infinite sequence of jobs, with successive job-arrivals separated by at least $p_i$ time units. A sporadic task system $\tau = \{\tau_1, \ldots, \tau_n\}$ is a collection of $n$ such sporadic tasks. We assume that the task system is constrained-deadline, that is, each task $\tau_i \in \tau$ has $d_i \leq p_i$. The task utilization for $\tau_i$ is defined as $u_i = e_i / p_i$ and the system utilization is denoted by $U(\tau) = \sum_{\tau_i \in \tau} u_i$, and $U(\tau) \leq 1$; otherwise, task system $\tau$ cannot be scheduled (by any algorithm) to meet all deadlines upon a dedicated preemptive uniprocessor.

We assume that task priorities are preassigned and each task has a fixed-priority. Tasks are indexed in non-increasing priority order. That is, $\tau_i$ has higher (or equal) priority than $\tau_j$, if and only if, $i \leq j$. As tasks generate jobs, each job inherits the priority of its generating task. For this paper, we assume each component uses fixed-priority scheduling as the component-level scheduling algorithm. Whenever component $C$ is allocated the processor, $C$ executes the highest-priority job with remaining execution; ties are broken in favor of the job generated from the lower task index. One noteworthy fixed-priority scheduling algorithm is deadline monotonic (DM) [22], which assigns each task a priority equal to the inverse of its relative deadline (i.e., tasks with shorter relative deadlines have priority greater than tasks with longer relative deadlines). DM is known to an optimal fixed-priority uniprocessor scheduling algorithm for constrained-deadline sporadic tasks in the following sense; if a constrained-deadline sporadic task system is schedulable upon a single processor by a fixed-priority assignment, then it will also meet all deadlines under the DM priority assignment.

§Workload Functions. To determine the schedulability of a sporadic task system, it is often useful to quantify the maximum amount of execution time requested by the task in its worst-case phasing over any given interval. For sporadic task systems, it is known that the worst-case phasing is the synchronous arrival sequence. The synchronous arrival sequence occurs when all tasks of a sporadic task system release jobs at the same time instant and subsequent jobs as soon as permissible. Researchers [20] have derived the request-bound function, as defined below.

**Definition 1 (Request-Bound Function).** For any $t > 0$ and sporadic task $\tau_i$, the request-bound function (RBF) quantifies the maximum cumulative execution requests that could be generated by jobs of $\tau_i$ arriving within a contiguous time-interval of length $t$. It has been shown that for sporadic tasks, RBF can be calculated as follows [20].

$$ RBF(\tau_i, t) \equiv \left\lceil \frac{t}{p_i} \right\rceil \cdot e_i. $$

(1)

Figure 1 shows the request-bound function for a sporadic task $\tau_i$, which is a right continuous function with discontinuities at time points of the form $t \equiv a \cdot p_i$ where $a \in \mathbb{N}$. The cumulative request-bound function for task $\tau_i$ is defined as follows:

$$ R_i(t) \equiv e_i + \sum_{j=1}^{i-1} RBF(\tau_j, t). $$

(2)
Audsley et al. [5] have given a necessary and sufficient condition for sporadic task system $\tau$ to be fixed-priority-schedulable upon a preemptive uniprocessor platform of unit speed: $\exists t \in (0, d_i]$ such that $R_i(t) \leq t, \forall i$. Furthermore, it has also been shown [4] that this condition needs to be verified at only time points in the following ordered set:

$$S_i(\tau) \overset{\text{def}}{=} \left\{ t = b \cdot p_a : a = 1, \ldots, i; b = 1, \ldots, \left\lfloor \frac{d_i}{p_a} \right\rfloor \right\} \cup \{ d_i \}.$$ (3)

The above set is known as the testing set for sporadic task $\tau_i$. The size of this set may be as large as $\sum_{j=1}^{i} \left\lfloor \frac{d_j}{p_j} \right\rfloor$ which is dependent on the task periods, and thus requires pseudo-polynomial time feasibility test. Fisher and Baruah [17] proposed the following approximation to RBF (inspired by a similar approximation for EDF due to Albers and Slomka [2]) to reduce the number of points in the testing set.

$$\delta (\tau_i, t, k) \overset{\text{def}}{=} \begin{cases} \frac{RBF(\tau_i, t)}{e_i + \frac{t \cdot \alpha_i}{p_i}}, & \text{if } t \leq (k-1)p_i \\ \text{otherwise.} & \end{cases}$$ (4)

This function tracks RBF for exactly $k - 1$ steps and after the $k-1$-th step, it uses linear interpolation of subsequent discontinuous points of RBF (with slope equal to $u_i$). The steps in Figure 1 correspond to $RBF(\tau_i, t, k)$, and the thick steps and the sloped-dashed line correspond to $\delta (\tau_i, t, k)$. The approximate cumulative request bound function is defined as follows:

$$\hat{R}_i(t) \overset{\text{def}}{=} e_i + \sum_{j=1}^{i-1} \delta(\tau_j, t, k).$$ (5)

For any fixed $k \in \mathbb{N}^+$, [17] showed that if for all $\tau_i \in \tau$ there exists a $t \in (0, d_i]$ such that $\hat{R}_i(t) \leq t$ then the sporadic task system $\tau$ is static priority schedulable upon a preemptive uniprocessor platform of unit speed. The testing set for this condition is as follows:

$$\hat{S}_i(\tau, k) \overset{\text{def}}{=} \left\{ t = b \cdot p_a : a = 1, \ldots, i-1; b = 1, \ldots, k-1; t \in (0, d_i] \right\} \cup \{ d_i \} \cup \{ 0 \}.$$ (6)

Let $t_a, t_{a+1}$ denote any pair of consecutive values in the above ordered set.

Next, we give the relation between the request bound function RBF and the approximate request bound function $\delta$.

**Lemma 1** (from [17]). Given a fixed integer $k \in \mathbb{N}^+$, $RBF(\tau_i, t) \leq \delta (\tau_i, t, k) \leq \frac{k+1}{k} RBF(\tau_i, t)$ for all $\tau_i \in \tau$ and $t \in \mathbb{R}_{\geq 0}$.

We will use this lemma in our approximation algorithm (Section 6).

Next, we define notation to represent the discontinuous line segments of the cumulative request bound function ($\hat{R}_i$). Let $\mathcal{L}_{t_a}^i \equiv \langle (t_a, D_{t_a}), (t_{a+1}, D_{t_{a+1}}), \alpha \rangle$ be a line segment in the Euclidean space, $\mathbb{R}^2$, originating at open left end point $(t_a, D_{t_a}) \in \mathbb{R}^2$ and ending at closed right end point $(t_{a+1}, D_{t_{a+1}}) \in \mathbb{R}^2$ with slope $\alpha \geq 0$ (Figure 1); more formally,

$$\mathcal{L}_{t_a}^i \overset{\text{def}}{=} \left\{ (x, y) \in \mathbb{R}^2 \mid (x \in [t_a, t_{a+1}]) \land (y = \alpha(x - t_a) + D_{t_a}) \right\}.$$ (7)

Please note the term $\alpha$ is included in the notation for convenience only; it is possible to determine the slope from points $(t_a, D_{t_a})$ and $(t_{a+1}, D_{t_{a+1}})$ alone. We denote any point in the line segment by $(t, D_t) \in \mathcal{L}_{t_a}^i$.

$$\alpha \overset{\text{def}}{=} \sum_{\tau_a \in \tau; (t_a \geq (k-1)p_a) \land (b < i)} u_b.$$ (8)

The connection between $\mathcal{L}_{t_a}^i$ and $\hat{R}_i$ is as follows. Consider a time $t_a \in \hat{S}_i(\tau, k)$. Define $D_{t_a}$ to be request bound function at time $t_a$, that is $\hat{R}_i(t_a)$ (Figure 1). At time $t_a$, some set of tasks with priority greater $\tau_i$ have job arrivals in the synchronous arrival sequence. Let $r_i(t)$ be the sum of the executions of these tasks. Formally,

$$r_i(t) \overset{\text{def}}{=} \sum_{\tau_j \in \tau; (j < i) \land (p_j \text{ divides } t)} e_j.$$ (9)
At time \( t_a \) there is a discontinuity in the function \( \hat{R}_i \) in which \( \hat{R}_i \) increases by \( r_i(t_a) \) and then is linear until the next discontinuity in \( \hat{R}_i \) (i.e., at time \( t_{a+1} \in \hat{S}_i(\tau,k) \)). Thus, \( \hat{R}_i \) is a line segment from \( t_a \) to \( t_{a+1} \) with slope equal to the total utilization of all task \( \tau_j \) such that \( j < i \) and \( t_a \geq (k-1)p_j \). We denote \( \bar{D}_{t_a} \) by the sum of request bound function at time point \( t_a \) and job release at time \( t_a \), that is, \( \bar{D}_{t_a} = D_{t_a} + r_i(t_a) \). From the above definitions of \( t_a \), \( t_{a+1} \), \( D_{t_a} \), \( \bar{D}_{t_a} \), \( D_{t_{a+1}} \), and \( \alpha \), it is straightforward to verify that the line segment \( L_{t_a} \) is equivalent to \( (t, \bar{R}_i(t)) \) for all \( t \in (t_a, t_{a+1}) \). The exception is at \( t_a \) where \( \bar{D}_{t_a} \) is not equal to \( \bar{R}_i(t_a) \); this difference exists for the notational and algebraic convenience throughout Section 5. From these definitions, the following lemma is apparent.

**Lemma 2.** For any consecutive pairs of values \((t_a, t_{a+1}) \in \hat{S}_i(\tau,k)\), \( \bar{R}_i(t) \leq D_t \) for all \( (t, D_t) \in L_{t_a} \).

### Explicit-Deadline Periodic (EDP) Resource Model

An EDP resource, denoted by \( \Omega = (\Pi, \Theta, \Delta) \), guarantees that a component \( C \) executed upon resource \( \Omega \) will receive at least \( \Theta \) units of execution between successive time points in \( \{t = t_0 + \ell \Pi \mid \ell \in \mathbb{N}\} \) where \( t_0 \) is some initial service start-time \( t_0 \) for the periodic resource. The “no-supply period” of \( \Omega \) is the duration of time at which the component \( C \) does not receive any processing resource; i.e., \( \Pi - \Theta \) units of time in any \( \Pi \)-interval of time in this case. Furthermore, the \( \Theta \) units of service must occur \( \Delta \) units after each successive time point in the aforementioned set. Obviously, \( \Theta \leq \Delta \); for this paper, we will make the simplifying assumption that \( \Delta \leq \Pi \), as well. Furthermore, we will assume in this paper that each component \( C \) is a sporadic task system \( \tau \) scheduled by fixed-priority upon \( \Omega \). (From now on, we use \( \tau \) in the context of component \( C \).)

#### Definition 2 (Supply-Bound Function)

For any \( t > 0 \), the supply-bound function \( \text{sbf} \) quantifies the minimum execution supply that a component executed upon periodic resource \( \Omega \) may receive over any interval of length \( t \). It is defined as follows [14]:

$$
\text{sbf}(\Omega,t) = \begin{cases} 
\gamma \Theta + \max(0, t - \gamma \Pi) & \text{if } t \geq \Delta - \Theta \\
0 & \text{otherwise}
\end{cases}
$$

where \( \gamma \) is the minimum capacity under \( \Omega \) for \( t \) given task system \( \tau \) by \( \Theta^*(\Pi, \Delta, \tau) \). We use the concept of \( \ell \)-feasibility region of \( \Omega \) similar to [18] to define the region under the \( \ell \)-th step of \( \text{sbf} \). For our convenience, we redefine \( \ell \)-feasibility region as follows:

**Definition 3 (\( \ell \)-Feasibility Region of \( \Omega \)).**

$$
\mathcal{F}_\ell(\Pi, \Theta, \Delta) = \left\{(t, D_t) \in \mathbb{R}_{\geq 0}^2 \mid \begin{array}{c}
\Theta \geq D_t + \ell (\Pi + \Delta) \\
\Theta \geq \frac{D_t}{\ell} \\
0 \leq t \leq \Pi \ell
\end{array} \right\}
$$

Figure 2 shows graphical depiction of the supply bound function \( \text{sbf} \) for EDP resource \( \Omega \). The shaded region in the figure corresponds to the \( \ell \)-feasibility region for some step \( \ell \in \mathbb{N} \) of the \( \text{sbf} \).

4 Determining Minimum Capacity Using Response Time

In this section, we derive an efficient exact schedulability test for fixed-priority scheduled component upon periodic resources similar to dedicated uniprocessor schedulability test. A fixed-priority scheduled component upon EDP
resource $\Omega$ can be considered as an equivalent dedicated uniprocessor system by adding special higher priority tasks to the original task system which corresponds to the “no-supply period” of $\Omega$. The traditional response time based fixed-priority schedulability test [20] can be applied to the modified task system.

Given a task system $\tau$ and EDP resource $\Omega \equiv (\Pi, \Theta, \Delta)$, we derive fixed-priority schedulability by considering two special periodic tasks $\tau_{-1}$ and $\tau_0$ with higher priority then all the tasks in task system $\tau$. Priority of $\tau_{-1}$ is higher than priority of $\tau_0$. Let $\tau' = \{\tau \cup \tau_{-1} \cup \tau_0\}$ be the new task system. Let $\tau_{-1}(e_{-1}, p_{-1}, d_{-1})$ have $e_{-1} = d_{-1} = \Delta - \Theta$, and $p_{-1} = \infty$. Let $\tau_0(e_0, p_0, d_0)$ have $e_0 = \Pi - \Theta, p_0 = d_0 = \Pi$, and a release jitter $j_0 = e_{-1}$. Note that for all the other task $\tau_i \in \tau'$, release jitter $j_i = 0$. This assumption enables us to correctly model the initial starvation period of $\Pi + \Delta - 2\Theta$ units of the periodic resource $\Omega$. Clearly, these two tasks together represent the “no-supply period” of $\Omega$, where $\tau_{-1}$ accounts for initial non-recurring starvation period $\Delta - \Theta$, and $\tau_0$ accounts for the resource unavailability in every $t - (\Delta - \Theta)$ interval. The exact fixed-priority schedulability test of $\tau$ with periodic resource $\Omega$.

Now, to solve our problem of obtaining minimum capacity $\Theta$, we use a binary search of $\Theta$ over the range $[0, \Pi]$, along with the exact fixed priority schedulability test of $\tau'$ shown in EXACTFPSCHEDULABILITY($\tau$, $\Omega$).

The request bound function for the two special tasks $\tau_{-1}$ and $\tau_0$ is as follows.

$\text{RBF}_0(t, \Omega) \equiv (\Delta - \Theta) + \max\left\{0, \left[\frac{t - (\Delta - \Theta)}{\Pi}\right] \cdot (\Pi - \Theta)\right\}.$ \hspace{1cm} (12)

Using this, we obtain response time\(^1\) for all tasks $\tau_i \in \tau$ as given by the following iterative equation.

$$R_i = e_i + \sum_{j \in hp(i)} \left[\frac{R_j}{p_j}\right] e_j + \text{RBF}_0(R_i, \Omega)$$ \hspace{1cm} (13)

\(^1\)We slightly abuse notation in this section by denoting response time for tasks in $\tau'$ as $R_i$. 

\begin{figure}[h]
\centering
\begin{lstlisting}
EXACTFPSCHEDULABILITY($\tau$, $\Omega$)
1 $\tau_{-1}(e_{-1}, p_{-1}, d_{-1}) = (\Delta - \Theta, \infty, \Delta - \Theta)$; $\tau_0(e_0, p_0, d_0) = (\Pi - \Theta, \Pi, \Delta)$
2 $u_{-1} \leftarrow 0, u_0 \leftarrow e_0/p_0$
3 $R_{0}^{ub} \leftarrow e_{-1} + e_0$
4 for Each task $\tau_i \in \tau$ in descending order of priority
5 \hspace{1cm} $R_{i}^{ub} \leftarrow \frac{e_i + \sum_{j \in hp(i)} e_j (1-u_j) + e_0 (1-u_0) + (1-u_{-1}) e_{-1}}{1 - \sum_{j \in hp(i)} u_j - u_{-1}}$
6 \hspace{1cm} if $R_{i}^{ub} > d_i$
7 \hspace{1cm} \hspace{1cm} $R_{i}^{ub} \leftarrow \frac{e_i + \Theta (\Delta - \Theta)}{\Pi - \Theta}$
8 \hspace{1cm} \hspace{1cm} $R_i \leftarrow \max\{(d_i + e_i)/2, d_i - R_{i-1}^{ub}, R_i^{ub}\}$
9 \hspace{1cm} \hspace{1cm} $\triangleright$ Solve the recurrence starting from initial response time
10 \hspace{1cm} \hspace{1cm} $R_i \leftarrow e_i + \sum_{j \in hp(i)} \left[\frac{R_j}{p_j}\right] e_j + \text{RBF}_0(R_i, \Pi, \Theta, \Delta)$
11 \hspace{1cm} \hspace{1cm} if $R_i > d_i$
12 \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} return Not Schedulable.
13 \hspace{1cm} \hspace{1cm} end if
14 \hspace{1cm} \hspace{1cm} end for
15 \hspace{1cm} return Schedulable.
\end{lstlisting}
\caption{Iterative convergence-based exact fixed-priority scheduling algorithm by modeling “no-supply period” of periodic resource $\Omega(\Pi, \Theta, \Delta)$ as higher priority tasks $\tau_{-1}, \tau_0$.}
\end{figure}
Thus, the exact schedulability test checks for each $\tau_i \in \tau'$, whether the response time $R_i$ is less or equal its relative deadline $d_i$. In the next two subsections, we derive lower bound and upper bound for the response time of tasks in $\tau'$, and use them to derive an efficient iterative schedulability test \textsc{ExactFPSchedulability}(\tau, \Omega), following the suggestions of [9].

4.1 Deriving Response Time Lower Bound

At any time $t > 0$, the response time lower bound $R_{i-1}^{lb}$ for $\tau_{i-1}$ must be equal to $e_{-1}$, which follows the response time lower bound for next highest priority task $\tau_0$ to be at least $R_{i-1}^{lb} = e_{-1} + e_0$ after it is released at time $t = \Delta - \Theta$. All the tasks in $\tau$ have lower priority than these two tasks, and we have two cases while determining their response time lower bound. When $t < \Delta - \Theta$, $R_i^{lb}$ must be greater than $R_{i-1}^{lb}$ and when $t \geq \Delta - \Theta$, $R_i^{lb}$ must be greater $R_0^{lb}, \forall \tau_i \in \tau$.

From Equation 13, we obtain the following inequality (by dropping the ceilings) when $t \geq \Delta - \Theta$:

$$R_i = e_i + \sum_{j \in hp(i)} \left[ \frac{R_j}{p_j} \right] e_j + (\Delta - \Theta) + \left[ \frac{R_i-(\Delta-\Theta)}{\Pi} \right] (\Pi - \Theta)$$

$$\geq e_i + \sum_{j \in hp(i)} \frac{R_j}{p_j} e_j + (\Delta - \Theta) + \frac{R_i-(\Delta-\Theta)}{\Pi} (\Pi - \Theta)$$

$$= e_i + \frac{R_i}{\Pi} U_{i-1} + \left( \frac{(\Delta-\Theta)\Pi + R_i(\Pi - \Theta) - (\Delta-\Theta)(\Pi - \Theta)}{\Pi} \right)$$

$$= \frac{\Pi e_i + \Theta \Pi U_{i-1}}{\Pi} + \left( \frac{(\Delta-\Theta)\Pi + R_i(\Pi - \Theta) - (\Delta-\Theta)(\Pi - \Theta)}{\Pi} \right)$$

(14)

Let $U_{i-1} = \sum_{j \in hp(i)} \frac{e_j}{p_j}$ (excluding $\tau_{-1}$ and $\tau_0$). Solving the above equation for $R_i$, we obtain response time lower bound $R_i^{lb}$. The last line of Equation 14 gives the lower bound $R_i^{lb}$ for the response time of task $\tau_i$.

4.2 Deriving Response Time Upper Bound

We derive the upper bound of response time for all $\tau_i \in \tau'$ using similar approach used by [7]. Using similar notation as [7], let $w_i(t)$ represents maximum amount of time that the processor executes $\tau_i$ in any interval length $t$, and $w_i^0(t)$ represents maximum amount of time that the processor executes $\tau_i$ in any interval length $t$ when $\tau_i$ is the only task in the system. Then the worst case workload at time $t$ for task $\tau_i$ is given by the following equation:

$$W_i(t) = \sum_{j=1}^{i} w_j(t) + w_0(t) + w_{-1}(t).$$

(15)

This implies:

$$W_i(t) \leq \sum_{j=1}^{i} w_j^0(t) + w_0^0(t) + w_{-1}^0(t).$$

(16)

Observe that for task $\tau_{-1}$, $w_{-1}^0(t)$ is equal to $\min\{\Delta - \Theta, t\}$. Therefore, we obtain following bound.

$$w_{-1}^0(t) \leq \Delta - \Theta.$$  

(17)

For $\tau_0$, $w_0^0(t) = \min\{t - (\Delta - \Theta) - (p_0 - e_0) \left[ \frac{t-(\Delta-\Theta)}{p_0} \right], \left[ \frac{t-(\Delta-\Theta)}{p_0} \right] e_0 \}$, which can be upper bounded by linear approximation of the step function.

$$w_0^0(t) \leq u_0 t + e_0 (1 - u_0) - (\Delta - \Theta) u_0.$$  

(18)

For all other tasks in $\tau'$, $w_i^0$ can be upper bounded by the following linear approximation.

$$w_i^0(t) \leq u_i t + e_i (1 - u_i).$$  

(19)
Combining Equation 17, 18, 19, we obtain workload upper bound for task $\tau_i$ at time $t$.

$$W_i(t) \leq \sum_{j=0}^{i} (u_j t + e_j (1 - u_j)) + (1 - u_0) (\Delta - \Theta).$$  \hspace{1cm} (20)

Now, using the steps similar to Theorem 2 of [7], we obtain response time upper bound from workload upper bound.

$$R_i \leq e_i + \sum_{j=0}^{i-1} e_j (1 - u_j) + (1 - u_0) (\Delta - \Theta) \frac{1 - \sum_{j=0}^{i-1} u_j}{1 - \sum_{j=0}^{i-1} u_j}. \hspace{1cm} (21)$$

Using Equation 14 and 21, we obtain initial value for our efficient iterative algorithm similar to [9]

$$R_i = \max \{(d_i + e_i)/2, d_i - R_{i-1}^{ub}, R_{i-1}^{lb}\} \hspace{1cm} (22)$$

In Figure 3, we give our efficient exact schedulability test for fixed-priority scheduled components upon periodic resources.

## 5 Determining Minimum Capacity Using Testing Set

### 5.1 Exact Test

The following theorem states the exact schedulability condition for EDP resource $\Omega$, where task system is scheduled using fixed priority scheduling algorithm [29, 30, 14]. It says that for the task system $\tau$ to be schedulable with EDP resource, each task $\tau_i$ in $\tau$ must have a fixed point $t$ before its deadline at which the cumulative request bound function for $\tau_i$ is less than the supply provided to the system at that point.

**Theorem 1** (from [14]). A sporadic task system $\tau$ is fixed priority schedulable upon an EDP resource $\Omega = (\Pi, \Theta, \Delta)$, if and only if,

$$\left( \forall i, \exists t \in (0, d_i] : R_i(t) \leq \operatorname{sbf}(\Omega, t) \right) \bigwedge \left( U(\tau) \leq \frac{\Theta}{\Pi} \right)$$  \hspace{1cm} (23)

In the next section we present an approximate algorithm to obtain minimum capacity for EDP resource when the component-level scheduling algorithm is fixed priority for the task system $\tau$. We consider fixed period ($\Pi$) and deadline ($\Delta$) for the EDP resource $\Omega$.

### 5.2 Approximate Test

In Figure 1, we present the pseudocode for our algorithm, FPMINIMUMCAPACITY$^2$. Given task system $\tau$ and an EDP resource with $\Pi$ and $\Delta$ as input, the algorithm returns approximate minimum capacity to correctly schedule the task system with the resource. The approximation parameter of the algorithm is the input $k \in \mathbb{N}^+ (k = \lceil \frac{1}{\epsilon} \rceil)$. For some fixed $k$ input, the algorithm returns the approximate minimum capacity; if $k$ is equal to $\infty$, it returns exact minimum capacity. If FPMINIMUMCAPACITY returns a value $\Theta^\text{min}$ that does not exceed $\Delta$, then $\tau$ can be fixed-priority scheduled to meet all deadlines upon $\Omega = (\Pi, \Delta, \Theta^\text{min})$. Note that the approximate capacity $\Theta^\text{min}$ can be at most $(1 + \epsilon)$ times the exact capacity. If FPMINIMUMCAPACITY returns a capacity greater than $\Delta$, then our algorithm cannot guarantee $\tau$ can be scheduled on any $\Omega$ with parameters $\Pi$ and $\Delta$. (Unless $k = \infty$, the algorithm is an approximation, and, thus, a returned capacity greater than $\Delta$ does not necessarily imply infeasibility of $\tau$).

$^2$We would like to thank the authors of [35] for identifying and correcting a small bug in our algorithm. In the original algorithm we did not check the condition in Line 10, and calculated $\Theta_{[l_1]}$ and $\Theta_{[l_2]}$ for all cases including the case $[l_2] > [l_1]$, which corresponds to an undefined interval (See Lemma 9, 10 and 11 for more details).
We will also show that we only need to consider the integer values of \( \tau \) such that there exist a \( \hat{a} \in \text{testing set} \) for \( \hat{a} \) in the testing set of \( \hat{a} \) is fixed-priority schedulable under EDP resource model. For each task \( \tau \) ((\( \Pi, \Delta, \tau \)) such that some point of the line segment is below the \( L \) for \( \ell \)-feasibility region with capacity \( R \)). Each \( R \) for \( \ell \) in our algorithm, the objective is to compute minimum capacity \( \Theta \) such that \( \tau \) is fixed-priority schedulable under EDP resource model. For each task \( \tau \) in \( \tau \), we find minimum capacity \( \Theta \) such that there exists a fixed point \( t \in (0, d, \Delta) \) at which the supply bound function \( \text{Sbf} \) exceeds the cumulative request bound function \( \hat{R} \) (Theorem 1). To calculate \( \Theta \), we determine, for each consecutive pair of values \( \tau \) in the testing set \( \hat{R} \), the minimum capacity \( \Theta \) required to guarantee that the line segment \( L \) is beneath \( \text{Sbf}((\Pi, \Theta, \Delta), t) \) for some \( t \in (t, t+1) \). Since \( L \) is equivalent to \( \hat{R} \) for all \( t \in (t, t+1) \), this implies that there exist a \( t \in (t, t+1) \) such that \( \hat{R} \) (\( t \leq \text{Sbf}((\Pi, \Theta, \Delta), t) \)). To determine \( \Theta \), we take specific steps of the \( \text{Sbf} \) (denote a selected step by \( \ell \)) and determine the minimum \( \Theta \) such that some point of the line segment is below the \( \ell \)-feasibility region with capacity \( \Theta \). Each \( \Theta \) for \( (t, t+1) \) is set in lines 8, 9, 11 and 12. The following functions are used to determine the values of \( \Theta \) in our algorithm.

\[
\Phi_1(L_{t_u}, \ell, \Pi, \Delta) \overset{def}{=} \frac{D_{t_{u+1}} - t_u + \ell\Pi + \Delta}{\ell + 1}, \\
\Phi_2(L_{t_u}, \ell, \Pi, \Delta) \overset{def}{=} \frac{D_{t_u}}{\ell}, \\
\Phi_3(L_{t_u}, \ell, \Pi, \Delta) \overset{def}{=} \frac{D_{t_u} + \alpha(\ell\Pi + 1)}{\ell + \alpha}.
\](24)

We will also show that we only need to consider the integer values of \( \ell \) given by the following equations.

\[
\ell_1 \overset{def}{=} \frac{(t_{u+1} - \Delta)^2 + 4\Pi D_{t_{u+1}}}{2\Pi},
\]

\[
\ell_2 \overset{def}{=} \frac{(t_u - \Delta)^2 + 4\Pi D_{t_u}}{2\Pi}.
\](26)
That is, we consider \(\lfloor \ell \tau \Theta \hat{\text{ }} \text{line segment of } \tau \text{ set } t \text{ n} \) for each task, thus Lemma 3. For any two consecutive pair of values \(\text{region.} \)

The complexity of \(\text{FPM}\) Algorithm Complexity.

Theorem 2. For all \(k \in \mathbb{N}^+ \cup \{\infty\}\), \(\text{FPM}\) Algorithm Correctness. To prove the correctness of \(\text{FPM}\) Algorithm, we prove the following theorem which states that the value returned by the algorithm (i.e., \(\Theta^\text{min}\)) is at least the optimal minimum capacity value \(\Theta^*(\Pi, \Delta, \tau)\). Furthermore, if the input \(k\) equals \(\infty\), then the returned capacity is optimal.

\(\text{FPM}\) Algorithm Complexity. The complexity of \(\text{FPM}\) Algorithm Complexity.

**Algorithm Correctness.** To prove the correctness of \(\text{FPM}\) Algorithm Complexity, we prove the following theorem which states that the value returned by the algorithm (i.e., \(\Theta^\text{min}\)) is at least the optimal minimum capacity value \(\Theta^*(\Pi, \Delta, \tau)\). Furthermore, if the input \(k\) equals \(\infty\), then the returned capacity is optimal.

\(\text{Algorithm Correctness.}\) To prove the correctness of \(\text{FPM}\) Algorithm Complexity, we prove the following theorem which states that the value returned by the algorithm (i.e., \(\Theta^\text{min}\)) is at least the optimal minimum capacity value \(\Theta^*(\Pi, \Delta, \tau)\). Furthermore, if the input \(k\) equals \(\infty\), then the returned capacity is optimal.

**Theorem 2.** For all \(k \in \mathbb{N}^+ \cup \{\infty\}\), \(\text{FPM}\) Algorithm Complexity. To prove the correctness of \(\text{FPM}\) Algorithm Complexity, we prove the following theorem which states that the value returned by the algorithm (i.e., \(\Theta^\text{min}\)) is at least the optimal minimum capacity value \(\Theta^*(\Pi, \Delta, \tau)\). Furthermore, if the input \(k\) equals \(\infty\), then the returned capacity is optimal.

We require some additional definitions similar to [18] for notational convenience. The next definition quantifies the minimum capacity \(\Theta(\leq \Delta)\) that is required for \(\text{sbf}\) to exceed the line segment \(L^i_{ta}\) at some point \((t, D_t)\). We will use the convention that \(\inf\) returns \(\infty\) on an empty set.

**Definition 4.** (Minimum Capacity for \(L^i_{ta}\)).

\[
\Theta^*(\Pi, \Delta, L^i_{ta}) \equiv \inf \left\{ \Theta \in \mathbb{R}^+ \mid \forall (\Theta \leq \Delta) \left( \exists (t, D_t) \in L^i_{ta} : D_t \leq \text{sbf}((\Pi, \Theta, \Delta), t) \right) \right\}.
\] (27)

The next function determines the minimum capacity for any given line segment \(L^i_{ta}\) to have a point in the \(\ell\)-feasibility region.

**Definition 5.** (\(\ell\)-Minimum Capacity for \(L^i_{ta}\)).

\[
\Theta^i_\ell(\Pi, \Delta, L^i_{ta}) \equiv \inf \left\{ \Theta(\leq \Delta) \in \mathbb{R}^+ \mid \exists (t, D_t) \in L^i_{ta} : D_t \leq \text{sbf}((\Pi, \Theta, \Delta), t) \right\}.
\] (28)

Note the two above definitions use infimum, since they are defined over infinite sets; however, we will later see (Corollary 4) that the infimum corresponds to the minimum (i.e., the value returned by \(\inf\) exists in the set specified in the right-hand side of Equations 27 and 28).

In order to prove Theorem 2, we must prove some additional lemmas. We start by presenting the three conditions on the value of \(\Theta\) that are necessary and sufficient condition for a line segment \(L^i_{ta}\) to have a point in the \(\ell\)-feasibility region.

**Lemma 3.** For any two consecutive pair of values \((t_a, t_{a+1}) \in \hat{S}_i(\tau, k)\), there exists \((t, D_t) \in L^i_{ta}\) such that \((t, D_t) \in F_{\ell}(\Pi, \Delta, \Theta)\) for some \(\ell \in \mathbb{N}^+\), if and only if, the following conditions hold:

\[
\Theta \geq \Phi_1(L^i_{ta}, \ell, \Pi, \Delta) \quad (29a)
\]
\[
\Theta \geq \Phi_2(L^i_{ta}, \ell, \Pi, \Delta) \quad (29b)
\]
\[
\Theta \geq \Phi_3(L^i_{ta}, \ell, \Pi, \Delta) \quad (29c)
\] (29)
Proof: For the “only if” direction, we must show if some point of the line segment is in the ℓ-feasibility region for any given ℓ ∈ N⁺ then the three conditions of Equation (29) hold. We will show this by contrapositive; that is, if any of the three conditions is violated, the line segment will not be in ℱℓ(Π, Δ, Θ) for that ℓ. We now consider the negation of the conditions of Equation (29). By negation, at least one of the Equations (29a), (29b), or (29c) must be violated. We will show that if any of the conditions is violated, then for all (t, Dℓ) ∈ ℓ∗(Π, Δ, Θ), (t, Dℓ) /∈ ℱℓ(Π, Δ, Θ).

Case 1: Equation (29a) is false. That is,

\[ \theta < \frac{D_{t_{a+1}} - t_{a+1} + \Pi + \Delta}{t_{a+1}} \]

⇒ \[ \theta < \frac{D_{t_{a}} + \alpha(t_{a+1} - t_{a}) - t_{a+1} + \Pi + \Delta}{t_{a+1}}. \]

The second inequality follows from the fact that \( D_{t_{a+1}} = \bar{D}_{t_{a}} + \alpha(t_{a+1} - t_{a}) \). Consider any \((t, D_{ℓ}) \in ℓ∗(Π, Δ, Θ)\). Let \( x \equiv t - t_{a} \) where \( 0 \leq x \leq t_{a+1} - t_{a} \); thus, \( t = t_{a} + x \) and \( D_{ℓ} = \bar{D}_{t_{a}} + \alpha x \). Consider the expression

\[ \frac{(\bar{D}_{t_{a}} + \alpha x) - (t_{a} + x) + \ell\Pi + \Delta}{\ell + 1} \]

Obviously, the above expression is non-increasing in \( x \), since \( U(τ) \leq 1 \) and \( \alpha \) is at most the utilization of tasks with higher priority than \( τ_{i} \). Therefore, \( \frac{D_{t_{a}} + \alpha(t_{a+1} - t_{a}) - t_{a+1} + \Pi + \Delta}{t_{a+1}} \leq \frac{(\bar{D}_{t_{a}} + \alpha x) - (t_{a} + x) + \ell\Pi + \Delta}{\ell + 1} \leq \frac{D_{t_{a+1}} - t_{a+1} + \Pi + \Delta}{t_{a+1}} \) for all \((t, D_{ℓ}) \in ℓ∗(Π, Δ, Θ)\). This implies that the first condition of ℓ-feasibility is violated for all \((t, D_{ℓ})\).

Case 2: Equation (29b) is false. That is, \( \theta < \frac{D_{t_{a}}}{\ell} \). Again, consider any \((t, D_{ℓ}) \in ℓ∗(Π, Δ, Θ)\). Observe that \( D_{ℓ} = \bar{D}_{t_{a}} + \alpha(t - t_{a}) \geq \bar{D}_{t_{a}} \), since \( t \geq t_{a} \) and \( \alpha \geq 0 \). Thus, \( \theta < \frac{D_{t_{a}}}{\ell} \) implies \( \theta < \frac{D_{t_{a}}}{\ell} \) for all \((t, D_{ℓ})\); this implies that the second condition of ℱℓ(Π, Δ, Θ) is violated.

Case 3: Equation (29c) is false. That is,

\[ \theta < \frac{\bar{D}_{t_{a}} + \alpha(\ell\Pi + \Delta - t_{a})}{\ell + 1}. \]

Consider any \((t, D_{ℓ}) \in ℓ∗(Π, Δ, Θ)\). We consider two further subcases based on the value of \( t \). We will show in both subcases, \((t, D_{ℓ}) \notin ℱℓ(Π, Δ, Θ)\).

Subcase 3a: \( t < \frac{\bar{D}_{t_{a}} + \alpha(\ell\Pi + \Delta - t_{a})}{\ell + 1} \).

By solving for \( \theta \), we obtain

\[ \theta < \frac{\bar{D}_{t_{a}} - \alpha t_{a} + \ell\Pi + \Delta}{t_{a} + 1}. \]

The implication follows from \( D_{ℓ} = \bar{D}_{t_{a}} + \alpha(t - t_{a}) \). The above inequality implies that the first condition of ℓ-feasibility is violated.

Subcase 3b: \( t > \frac{\bar{D}_{t_{a}} + \alpha(\ell\Pi + \Delta - t_{a})}{\ell + 1} \).

Again, solving for \( \theta \),

\[ \theta > \frac{\bar{D}_{t_{a}} - \alpha t_{a} + \ell\Pi + \Delta}{t_{a} + 1}. \]

Now consider the value of the first partial derivative of \( \Phi_{3} \) with respect to \( \alpha \); i.e., \( \frac{∂\Phi_{3}}{∂\alpha} \)

\[ \frac{\ell(\ell\Pi + \Delta - t_{a}) - \bar{D}_{t_{a}}}{(\ell + 1)^{2}}. \]

Note the sign of the above partial derivative is independent of the value of \( \alpha \); therefore, either \( \frac{∂\Phi_{3}}{∂\alpha} \leq 0 \), or \( \frac{∂\Phi_{3}}{∂\alpha} > 0 \) for any \( \alpha : 0 \leq \alpha \leq 1 \); in other words, the sign remains constant for all \( \alpha \). If \( \frac{∂\Phi_{3}}{∂\alpha} > 0 \), then \( \Phi_{3} \) is maximized when \( \alpha \) is as large as possible (i.e., \( \alpha \) equals one). By Equation (30), this implies that \( \theta < \frac{\bar{D}_{t_{a}} + \ell\Pi + \Delta - t_{a}}{t_{a} + 1} \) which is impossible due to Equation (31). Thus, \( \frac{∂\Phi_{3}}{∂\alpha} \leq 0 \) must be true. If the partial derivative is non-positive, then \( \Phi_{3} \) is maximized
when \( \alpha \) is as small as possible (i.e., \( \alpha \) equals zero). By Equation (30), \( \Theta < \frac{D_t}{t} \) which violates the second condition of \( \ell \)-feasibility.

Thus, we have proved that if the line segment has a point in the \( \ell \)-feasibility region, then the conditions in Equation (29) hold.

For the “if” direction, we need to show, if the conditions hold then there exists a point on the line segment that is included in the \( \ell \)-feasibility region. Again, we will show by contrapositive; that is, if the line segment is completely outside the \( \ell \)-feasibility region, then there is a condition of Equation (29) that is not satisfied. Assume that for all \( (t, D_t) \in \mathcal{L}_t \) that \( (t, D_t) \notin \mathcal{F}_t(\Pi, \Delta, \Theta) \). The previous statement implies that the first or the second condition of \( \ell \)-feasibility must be violated for each \( (t, D_t) \). We now consider two cases based on the “location” of the left end point of the line segment \( (t_a, \bar{D}_{t_a}) \).

**Case 1:** The second condition of \( \ell \)-feasibility is violated for \( (t_a, \bar{D}_{t_a}) \). In this case, \( \Theta < \frac{\bar{D}_{t_a}}{t_a} \). Indeed, this violates the condition of Equation (29b).

**Case 2:** The second condition of \( \ell \)-feasibility is not violated for \( (t_a, \bar{D}_{t_a}) \). In this case, \( \Theta \geq \frac{\bar{D}_{t_a}}{t_a} \). We now consider two further subcases regarding the “location” of \( (t_{a+1}, \bar{D}_{t_{a+1}}) \).

**Subcase 2a:** The first condition of \( \ell \)-feasibility is violated for the right end point of line segment \( (t_{a+1}, \bar{D}_{t_{a+1}}) \). In this case, \( \Theta < \frac{\bar{D}_{t_{a+1}} - t_{a+1} + t + \Delta}{t_{a+1}} \). This clearly violates the condition of Equation (29a).

**Subcase 2b:** The first condition of \( \ell \)-feasibility is not violated for \( (t_{a+1}, \bar{D}_{t_{a+1}}) \). In this subcase, \( \frac{\bar{D}_{t_{a+1}} - t_{a+1} + t + \Delta}{t_{a+1}} \leq \Theta \). Consider the function \( \theta(t) \equiv \frac{\bar{D}_{t_a} + \alpha(t - t_a) - t + \Pi + \Delta}{t - 1} \) for \( t \in [t_a, t_{a+1}] \). Thus, by this subcase and \( D_{t_{a+1}} = \bar{D}_{t_a} + \alpha(t_{a+1} - t_a) \) we obtain the following equation,

\[
\left( \theta(t_{a+1}) \equiv \frac{\bar{D}_{t_a} + \alpha(t_{a+1} - t_a) - t_{a+1} + \Pi + \Delta}{t_{a+1} - 1} \right) \leq \Theta. \tag{32}
\]

By Case 2, the second condition of \( \ell \)-feasibility is not violated for \( (t_a, \bar{D}_{t_a}) \). Thus, the first condition must be; i.e.,

\[
\left( \theta(t_a) \equiv \frac{\bar{D}_{t_a} - t_a + \Pi + \Delta}{t_a - 1} \right) > \Theta. \tag{33}
\]

Therefore, \( \Theta \in [\theta(t_{a+1}), \theta(t_a)] \). Observe that \( \theta(t) \) is continuous for all \( t \in [t_a, t_{a+1}] \). Therefore, the Intermediate Value Theorem implies that there exists a \( t' \in [t_a, t_{a+1}] \) such that \( \theta(t') = \Theta \). That is,

\[
\frac{\bar{D}_{t_a} + \alpha(t' - t_a) - t' + \Pi + \Delta}{t - 1} = \Theta. \tag{34}
\]

By the above equality, the first condition of \( \ell \)-feasibility is not violated for \( (t', D_{t'}) \); therefore, the second condition must be false:

\[
\frac{\bar{D}_{t_a} - t_a + \Pi + \Delta}{t - 1} > \Theta. \tag{35}
\]

Solving Equation (34) for \( t' \), we obtain

\[
t' = \frac{\bar{D}_{t_a} - \alpha t_a + \Pi + \Delta - (t - 1)\Theta}{1 - \alpha}.
\]

Substituting the above solution to \( t' \) into Equation (35) and solving for \( \Theta \), we obtain

\[
\Theta < \frac{\bar{D}_{t_a} - \alpha(t + \Pi + \Delta - t_a)}{t + \alpha}
\]

which indeed violates the condition of Equation (29c).
Thus, if the line segment is strictly above the \( \ell \)-feasibility region, at least one of the three conditions is violated. \( \square \)

The following lemma formalizes the equivalence between the concept of a line segment \( L_{t_a} \) being included in some \( \ell \)-feasibility region and the concept of a cumulative request-bound function \( \tilde{R}_t \) falling below a supply-bound function \( \text{sbf} \).

**Lemma 4.** For consecutive pair of values \((t_a, t_{a+1}) \in S_i(\tau, k)\) and \((t, D_t) \in L_{t_a}^i\) such that \( t_a < t \leq t_{a+1} \), the inequality \( \tilde{R}_t(t) \leq \text{sbf}(\Pi, \Theta, \Delta) \), \( t \) holds, if and only if, there exists \( \ell \in \mathbb{N}^+ \) such that \((t, D_t) \in F_\ell(\Pi, \Delta, \Theta)\).

**Proof:** For the “if” direction, we must show that if the point \((t, D_t) \in L_{t_a}^i\) satisfies \((t, D_t) \in F_\ell(\Pi, \Delta, \Theta)\), then there is sufficient supply over an interval of length \( \ell \) to satisfy the execution of a job of \( \tau_t \) and the approximated execution times of all higher-priority tasks (formally, \( \tilde{R}_t(t) \leq \text{sbf}(\Pi, \Theta, \Delta) \)). Observe that every point in \( F_\ell(\Pi, \Delta, \Theta) \) is below the \( \text{sbf} \) function (see Figure 2). Thus, if \((t, D_t) \in F_\ell(\Pi, \Delta, \Theta)\), then \( D_t \leq \text{sbf}(\Pi, \Theta, \Delta, t) \). Finally, Lemma 2 states that \( \tilde{R}_t(t) \leq D_t \) implying the “if” direction.

For the “only if” direction, observe that \( L_{t_a}^i \) and \( \tilde{R}_t(t) \) are equivalent for \( t \in (t_a, t_{a+1}) \). Thus, we must show that if line segment \( L_{t_a}^i \) has point \((t, D_t)\) contained below the \( \text{sbf} \) function for \( \Omega \), then there exists an \( \ell \in \mathbb{N}^+ \) such that \( \langle(t, D_t), a \rangle \in F_\ell(\Pi, \Theta, \Delta) \). Consider \( \ell = \lceil \frac{D_t}{\Theta} \rceil \). The second condition of \( \ell \)-feasibility (Equation (11)) is trivially satisfied for this \( \ell \). It also must be true that \( D_t > (\ell - 1) \Theta \). Thus, \((t, D_t)\) must be below of the line defined by \( y = x - (\ell \Pi + \Delta - (\ell + 1) \Theta) \) (otherwise, \((t, D_t)\) would be above the \( \text{sbf} \) function at \( t \)). This last constraint is equivalent to the first condition of \( \ell \)-feasibility region. Therefore, for \( \ell = \lceil \frac{D_t}{\Theta} \rceil \) we have satisfied the two conditions of Equation (11), implying that \( (t, D_t) \in F_{\lceil \frac{D_t}{\Theta} \rceil}(\Pi, \Theta, \Delta) \).

In the above lemma, we did not include \( t_a \) in the interval of time values where line segment inclusion in the \( \ell \)-feasibility region implies that the approximate request-bound function is below the supply-bound function. The exclusion of \( t_a \) from the above lemma is due to the fact that \( \tilde{R}_t \) is discontinuous at \( t_a \). However, notice that \( t_a \) is the right end point of the predecessor line segment immediately to the left of \( L_{t_a}^i \).

Lemma 3 equates the concept of finding \( t \) such that \( \tilde{R}_t(t) \) is below the \( \text{sbf} \) for a given \( \Theta \) and the concept of point \((t, D_t)\) of a line segment \( L_{t_a}^i \) being contained in some \( \ell \)-feasibility region for \( \Theta \). The next lemma uses Definitions 4 and 5 to show that if we can compute \( \Theta^*_\ell(\Pi, \Delta, L_{t_a}^i) \) for any \( \ell \in \mathbb{N}^+ \), then we can also compute \( \Theta^*(\Pi, \Delta, L_{t_a}^i) \).

**Lemma 5.**

\[
\Theta^*(\Pi, \Delta, L_{t_a}^i) = \inf_{\ell > 0} \{ \Theta^*_\ell(\Pi, \Delta, L_{t_a}^i) \}. \tag{36}
\]

**Proof:** Let \( \Theta_{\text{RHS}} \) denote the right-hand side of Equation (36). We will show that both \( \Theta_{\text{RHS}} \geq \Theta^*(\Pi, \Delta, L_{t_a}^i) \) and \( \Theta_{\text{RHS}} \leq \Theta^*(\Pi, \Delta, L_{t_a}^i) \) which will imply the lemma. First, we show \( \Theta_{\text{RHS}} \geq \Theta^*(\Pi, \Delta, L_{t_a}^i) \).

By definition of infimum, for any \( \delta > 0 \), there exists \( \ell \in \mathbb{N}^+ \) such that \( \Theta^*_\ell(\Pi, \Delta, L_{t_a}^i) \leq \Theta_{\text{RHS}} + \delta \). Definition 5 states that there exists \( (t, D_t) \in L_{t_a}^i \) such that \((t, D_t) \in F_\ell(\Pi, \Theta_{\text{RHS}} + \delta, \Delta) \) for this \( \ell \). Therefore, for all \( \delta > 0 \), \( \Theta_{\text{RHS}} + \delta \) must be in the set considered in the inf on the right-hand side of Equation (27) by Definition 4. Thus, \( \Theta_{\text{RHS}} \geq \Theta^*(\Pi, \Delta, L_{t_a}^i) \).

Next, we will show \( \Theta_{\text{RHS}} \leq \Theta^*(\Pi, \Delta, L_{t_a}^i) \). By Definition 4 and application of Lemma 2, there exist \((t, D_t) \in L_{t_a}^i \) such that

\[
\tilde{R}_t(t) \leq \text{sbf}(\Pi, \Delta, \Theta^*(\Pi, \Delta, L_{t_a}^i)), t).
\]

Lemma 4 implies that there exists \( \ell \in \mathbb{N}^+ \) such that \((t, D_t) \in F_\ell(\Pi, \Theta^*(\Pi, \Delta, L_{t_a}^i), \Delta) \). By Definition 5, this implies that \( \Theta^*(\Pi, \Delta, L_{t_a}^i) \) is in the set considered in the right-hand side of Equation (28) which implies the inequality. \( \square \)

In the next few lemmas, we derive the values \( \ell_1 \) and \( \ell_2 \) (Equations (25) and (26)), and prove that we only need to evaluate the \( \Phi \) functions at these \( \ell \) values to obtain minimum capacity. Consider the three conditions given in Equation (29) of Lemma 3. There are three possible cases. We invite the reader to verify that these cases are complete and mutually exclusive.

**Case I:** \( (\Phi_1(L_{t_a}^i, \ell, \Pi, \Delta) > \Phi_2(L_{t_a}^i, \ell, \Pi, \Delta)) \land (\Phi_1(L_{t_a}^i, \ell, \Pi, \Delta) > \Phi_3(L_{t_a}^i, \ell, \Pi, \Delta)) \);

**Case II:** \( (\Phi_2(L_{t_a}^i, \ell, \Pi, \Delta) > \Phi_3(L_{t_a}^i, \ell, \Pi, \Delta)) \land (\Phi_2(L_{t_a}^i, \ell, \Pi, \Delta) > \Phi_1(L_{t_a}^i, \ell, \Pi, \Delta)) \);
Lemma 6. For any $L_t^i$, and $\ell \in \mathbb{N}^+$, $\Pi$, $\Delta$, Case I holds, if and only if,

$$\ell \geq \left\lfloor \frac{(t_{a+1} - \Delta) + \sqrt{(t_{a+1} - \Delta)^2 + 4\Pi D_{t_{a+1}}}}{2\Pi} \right\rfloor + 1.$$  \hspace{1cm} (37)

Proof: Let us consider the “only if” direction of the lemma; that is, Case I holds. From Case I, we have that both $\Phi_1(L_t^i, \ell, \Pi, \Delta) > \Phi_2(L_t^i, \ell, \Pi, \Delta)$ and $\Phi_1(L_t^i, \ell, \Pi, \Delta) > \Phi_3(L_t^i, \ell, \Pi, \Delta)$. For $\Phi_1(L_t^i, \ell, \Pi, \Delta) > \Phi_2(L_t^i, \ell, \Pi, \Delta)$, solving for $\ell$,

$$\left\lfloor \frac{(t_{a+1} - \Delta) + \sqrt{(t_{a+1} - \Delta)^2 + 4\Pi D_{t_{a+1}}}}{2\Pi} \right\rfloor + 1.$$ \hspace{1cm} (38)

The bidirectional implication follows since Inequality (38) is a quadratic inequality with respect to $\ell$, defining a convex parabola $\Pi \ell^2 - ((1 - \alpha)t_{a+1} + \alpha t_a - \Delta) \ell - D_{t_a}$. The zeros of the parabola are

$$[(1 - \alpha)t_{a+1} + \alpha t_a - \Delta] \pm \sqrt{((1 - \alpha)t_{a+1} + \alpha t_a - \Delta)^2 + 4\Pi D_{t_a}}.$$  \hspace{1cm} (39)

Since the square-root term in the numerator is always greater than the term preceding the $\pm$, one root is positive and the other is negative. Inequality (38) implies that we are interested in values of $\ell \in \mathbb{N}^+$ such that the parabola strictly exceeds zero. Since the parabola is convex, all values of $\ell$ strictly greater than the positive root satisfy this inequality.

For $\Phi_1(L_t^i, \ell, \Pi, \Delta) > \Phi_3(L_t^i, \ell, \Pi, \Delta)$, solving for $\ell$,

$$\left\lfloor \frac{(t_{a+1} - \Delta) + \sqrt{(t_{a+1} - \Delta)^2 + 4\Pi D_{t_{a+1}}}}{2\Pi} \right\rfloor + 1.$$ \hspace{1cm} (40)

The bidirectional implication follows since Inequality (40) is a quadratic inequality with respect to $\ell$, defining a convex parabola $\Pi \ell^2 - (t_{a+1} - \Delta) \ell - (D_{t_a} + \alpha(t_{a+1} - t_a))$. By similar reasoning done for Inequality (38), all values of $\ell$ strictly greater than the positive root satisfy this inequality.

Combining Equations (38) and (39), we obtain

$$\ell > \max \left\{ \frac{[(1 - \alpha)t_{a+1} + \alpha t_a - \Delta] + \sqrt{((1 - \alpha)t_{a+1} + \alpha t_a - \Delta)^2 + 4\Pi D_{t_a}}}{2\Pi}, \frac{(t_{a+1} - \Delta) + \sqrt{(t_{a+1} - \Delta)^2 + 4\Pi D_{t_{a+1}}}}{2\Pi} \right\}.$$ \hspace{1cm} (41)

Observe that $(1 - \alpha)t_{a+1} + \alpha t_a - \Delta$ equals $t_{a+1} - \Delta - \alpha(t_{a+1} - t_a)$ which is at most $t_{a+1} - \Delta$, since $t_{a+1} > t_a$ and $0 \leq \alpha \leq 1$. Thus, we conclude that the second value of Equation (40) is the maximum of the two bounds obtained in this case. The lemma follows by observing that $\ell$ is an integer. The “if” direction follows by simply reversing the direction of each implication in the proof.

Similarly, we can prove the next two lemmas and bound the values of $\ell$ for Case II and III.

Lemma 7. For any $L_t^i$, and $\ell \in \mathbb{N}^+$, $\Pi$, $\Delta$, Case II holds, if and only if,

$$\ell \leq \left\lfloor \frac{(t_a - \Delta) + \sqrt{(t_a - \Delta)^2 + 4\Pi D_{t_a}}}{2\Pi} \right\rfloor - 1.$$ \hspace{1cm} (41)
Proof: Let us consider the “only if” direction of the lemma; that is, Case II holds. From Case II, we have that both \( \Phi_2(L^i_{ta}, \ell, \Pi, \Delta) > \Phi_3(L^i_{ta}, \ell, \Pi, \Delta) \) and \( \Phi_2(L^i_{ta}, \ell, \Pi, \Delta) \geq \Phi_1(L^i_{ta}, \ell, \Pi, \Delta) \). For \( \Phi_2(L^i_{ta}, \ell, \Pi, \Delta) > \Phi_3(L^i_{ta}, \ell, \Pi, \Delta) \), solving for \( \ell \),

\[
\frac{D_{ta}}{\ell} > \frac{D_{ta} + \alpha((\Pi + \Delta) - t_a)}{\ell + a} \\
\Leftrightarrow \ell < \frac{(t_a - \Delta) + \sqrt{(t_a - \Delta)^2 + 4\Pi D_{ta}}}{2\Pi}
\]

The bidirectional implication follows since Inequality (42) is a quadratic inequality with respect to \( \ell \), defining a convex parabola \( \Pi \ell^2 - (t_a - \Delta) \ell - D_{ta} \). The zeros of the parabola are

\[
(t_a - \Delta) \pm \sqrt{(t_a - \Delta)^2 + 4\Pi D_{ta}} \over 2\Pi.
\]

Since the square-root term in the numerator is always greater than the term preceding the \( \pm \), one root is positive and the other is negative. Inequality (42) implies that we are interested in values of \( \ell \in \mathbb{N}^+ \) such that the parabola is strictly below zero. Since the parabola is convex, all positive integer values of \( \ell \) strictly less than the positive root satisfy this inequality.

For \( \Phi_2(L^i_{ta}, \ell, \Pi, \Delta) \geq \Phi_1(L^i_{ta}, \ell, \Pi, \Delta) \), solving for \( \ell \),

\[
\frac{D_{ta}}{\ell} \geq \frac{D_{ta+1} - t_{a+1} + \Pi + \Delta}{\ell + 1} \\
\Leftrightarrow \ell \leq \frac{((1 - \alpha)t_{a+1} + \alpha t_a - \Delta) + \sqrt{((1 - \alpha)t_{a+1} + \alpha t_a - \Delta)^2 + 4\Pi D_{ta+1}}}{2\Pi}
\]

The bidirectional implication follows since Inequality (43) is a quadratic inequality with respect to \( \ell \), defining a convex parabola \( \Pi \ell^2 - ((1 - \alpha)t_{a+1} + \alpha t_a - \Delta) \ell - D_{ta+1} \). By similar reasoning done for Inequality (42), all positive integer values of \( \ell \) at most the positive root satisfy this inequality.

Now consider the following term: \((1 - \alpha)t_{a+1} + \alpha t_a - \Delta\) which equals \((1 - \alpha)(t_{a+1} - t_a) + t_a - \Delta\) which is at least \(t_a - \Delta\) since \(t_{a+1} > t_a\) and \(0 \leq \alpha \leq 1\). Thus, we conclude that the value on the right-hand-side of Equation (42) is the minimum of the two values obtained in this case. The lemma follows by observing that \( \ell \) must be an integer. The “if” direction of the lemma follows by simply reversing the implications of the proof. □

Lemma 8. For any \( L^i_{ta} \) and \( \ell \in \mathbb{N}^+ \), \( \Pi, \Delta \), Case III holds, if and only if,

\[
\left( t_a - \Delta \right) \pm \frac{\sqrt{(t_a - \Delta)^2 + 4\Pi D_{ta}}} {2\Pi} \leq \ell \leq \frac{((t_a + 1) - \Delta) + \sqrt{(t_a + 1) - \Delta)^2 + 4\Pi D_{ta+1}}}{2\Pi}
\]

Proof: Let us consider the “only if” direction of the lemma; that is, Case III holds. From Case III, we have that both \( \Phi_3(L^i_{ta}, \ell, \Pi, \Delta) \geq \Phi_1(L^i_{ta}, \ell, \Pi, \Delta) \) and \( \Phi_3(L^i_{ta}, \ell, \Pi, \Delta) \geq \Phi_2(L^i_{ta}, \ell, \Pi, \Delta) \). For \( \Phi_3(L^i_{ta}, \ell, \Pi, \Delta) \geq \Phi_1(L^i_{ta}, \ell, \Pi, \Delta) \), solving for \( \ell \),

\[
\frac{D_{ta} + \alpha((\Pi + \Delta - t_a)}{\ell + a} \geq \frac{D_{ta+1} - t_{a+1} + \Pi + \Delta}{\ell + 1} \\
\Leftrightarrow \ell \leq \frac{(t_a + 1 - \Delta) + \sqrt{(t_a + 1 - \Delta)^2 + 4\Pi D_{ta+1}}}{2\Pi}
\]

The bidirectional implication follows since Inequality (45) is a quadratic inequality with respect to \( \ell \), defining a convex parabola \( \Pi \ell^2 - (t_a + 1 - \Delta) \ell - (D_{ta} + \alpha(t_{a+1} - t_a)) \). The zeros of the parabola are

\[
(t_a + 1 - \Delta) \pm \frac{\sqrt{(t_a + 1 - \Delta)^2 + 4\Pi D_{ta+1}}}{2\Pi}.
\]

Since the square-root term in the numerator is always greater than the term preceding the \( \pm \), one root is positive and the other is negative. Inequality (45) implies that we are interested in values of \( \ell \in \mathbb{N}^+ \) such that the parabola is at most zero. Since the parabola is convex, all positive integer values of \( \ell \) at most the positive root satisfy this inequality.
For \( \Phi_3(L_{t_a}, \ell, \Pi, \Delta) \geq \Phi_2(L_{t_a}, \ell, \Pi, \Delta) \), solving for \( \ell \),

\[
\frac{D_{t_a} + \alpha(\Pi + \Delta - t_a)}{\ell + \alpha} \geq \frac{D_{t_a}}{\ell} \\
\iff \ell \geq \frac{(t_a - \Delta) + \sqrt{(t_a - \Delta)^2 + 4\Pi D_{t_a}}}{2\Pi}
\]

(46)

The bidirectional implication follows since Inequality (46) is a quadratic inequality with respect to \( \ell \), defining a convex parabola \( \Pi \ell^2 - (t_a - \Delta) \ell - D_{t_a} \). The zeros of the parabola are

\[
\ell \geq \frac{(t_a - \Delta) \pm \sqrt{(t_a - \Delta)^2 + 4\Pi D_{t_a}}}{2\Pi}.
\]

Since the square-root term in the numerator is always greater than the term preceding the \( \pm \), one root is positive and the other is negative. Inequality (46) implies that we are interested in values of \( \ell \in \mathbb{N}^+ \) such that the parabola is at least zero. Since the parabola is convex, all positive integer values of \( \ell \) at least the positive root satisfy this inequality.

The lemma follows by observing that \( \ell \) must be an integer. The “if” direction of the lemma follows by simply reversing the implications of the proof.

We now prove three lemmas and corollaries which show that for all \( \ell \in \mathbb{N}^+ \) not equal to the values \( \lfloor \ell_1 \rfloor, \lfloor \ell_1 \rfloor + 1, \lceil \ell_2 \rceil \) or \( \lceil \ell_2 \rceil - 1 \) will result in a larger minimum \( \Theta \). The first lemma, towards this goal, shows that if a point on the line segment is in an \( \ell \)-feasibility region and \( \ell' \) is at least \( \lfloor \ell_1 \rfloor + 1 \), then the point is also in the \( \lfloor \ell_1 \rfloor + 1 \)-feasibility region.

**Lemma 9.** For any \( t_a, t_{a+1} \in \hat{S}(\tau, k), (t, D_t) \in L_{t_a}, \ell' \in \mathbb{N}^+, \Pi, \Delta, \) and \( \Theta, \) if \( \ell' \geq \lfloor \ell_1 \rfloor + 1 \) and \( \Theta \leq \Delta \) then

\[
[(t, D_t) \in F_{\ell'}(\Pi, \Delta, \Theta)] \Rightarrow [(t, D_t) \in F_{\lfloor \ell_1 \rfloor + 1}(\Pi, \Delta, \Theta)].
\]

**Proof:** By Lemma 6 and \( \ell' \geq \lfloor \ell_1 \rfloor + 1 \), Case I must hold for all such \( \ell' \). Combining Case I and Lemma 3, we have that if \( (t, D_t) \in F_{\ell'}(\Pi, \Delta, \Theta) \), then

\[
\Theta \geq \Phi_1(L_{t_a}, \ell', \Pi, \Delta).
\]

Now consider the first partial derivative of \( \Phi_1 \) with respect to \( \ell \); i.e.,

\[
\frac{\partial \Phi_1}{\partial \ell} = -\frac{D_{t_a+1} + D_{t_a} - \Pi \Delta + \Pi (\ell + 1)}{(\ell + 1)^2}.
\]

Since \( \Pi \geq \Delta \), the second term in the numerator is positive. Consider the first term, \( t_{a+1} - D_{t_a+1} \). By \( (t, D_t) \in F_{\ell'}(\Pi, \Delta, \Theta) \) and the first condition of \( \ell' \)-feasibility,

\[
t \geq D_t + \ell \Pi + \Delta - (\ell + 1)\Theta \\
\Rightarrow t + (t_{a+1} - t) \geq D_t + \alpha(t_{a+1} - t) + \ell \Pi + \Delta - (\ell + 1)\Theta \\
\text{(since } \alpha < 1) \\
\Rightarrow t_{a+1} \geq D_{t_a+1} + \ell \Pi + \Delta - (\ell + 1)\Theta \\
\Rightarrow t_{a+1} \geq D_{t_{a+1}}.
\]

The second to last implication is due to \( D_t = \bar{D}_{t_a} + \alpha(t - t_a) \) and \( D_{t_{a+1}} = \bar{D}_{t_a} + \alpha(t_{a+1} - t_a) \). The last implication is due to \( \Theta \leq \Delta \). Therefore, the first term in the numerator of \( \frac{\partial \Phi_1}{\partial \ell} \) is also positive. Thus, \( \frac{\partial \Phi_1}{\partial \ell} \) is non-decreasing for all \( \ell' \). Thus, the \( \Phi_1 \) evaluated at \( \lfloor \ell_1 \rfloor + 1 \) is a lower bound; i.e., for all \( \ell' \geq \lfloor \ell_1 \rfloor + 1 \),

\[
\Phi_1(L_{t_a}, \ell', \Pi, \Delta) \geq \Phi_1(L_{t_a}, \lfloor \ell_1 \rfloor + 1, \Pi, \Delta).
\]

The above inequality implies that \( \Theta \geq \Phi_1(L_{t_a}, \lfloor \ell_1 \rfloor + 1, \Pi, \Delta) \), satisfying Equation (29a) of Lemma 3. For \( \lfloor \ell_1 \rfloor + 1 \), Case I holds, implying that Equations (29b) and (29c) must also hold. Thus, by Lemma 3, \( (t, D_t) \in F_{\lfloor \ell_1 \rfloor + 1}(\Pi, \Delta, \Theta) \).

The next corollary follows from the above lemma and the definition of \( \Theta_t^\ell \) (Definition 5).
Corollary 1. For any \( t_a, t_{a+1} \in \hat{S}_t(\tau, k) \), \( \ell' \in \mathbb{N}^+ \), and \( \Delta \), if \( \ell' \geq [\ell_1] + 1 \) then
\[
\Theta^*_t(\Pi, \Delta, L^i_t) \geq \Theta^*_t(\ell_{a+1}; \Pi, \Delta, L^i_t).
\]

The next lemma shows that if a point on the line segment is in an \( \ell' \)-feasibility region and \( \ell' \) is at most \( [\ell_2] - 1 \), then the point is also in the \([\ell_2] - 1\)-feasibility region.

Lemma 10. For any \( t_a, t_{a+1} \in \hat{S}_t(\tau, k) \), \( (t, D_t) \in L^i_t \), \( \ell' \in \mathbb{N}^+ \), \( \Pi, \Delta \), and \( \Theta \), if \( \ell' \leq [\ell_2] - 1 \) and \( \Theta \leq \Delta \) then
\[
[(t, D_t) \in \mathcal{F}_{\ell'}(\Pi, \Delta, \Theta)] \Rightarrow [(t, D_t) \in \mathcal{F}_{[\ell_2]-1}(\Pi, \Delta, \Theta)].
\]

Proof: By Lemma 7 and \( \ell' \leq [\ell_2] - 1 \), Case II must hold for all such \( \ell' \). Combining Case II and Lemma 3, we have that if \( (t, D_t) \in \mathcal{F}_{\ell'}(\Pi, \Delta, \Theta) \), then
\[
\Theta \geq \Phi_2(L^i_t, \ell', \Pi, \Delta).
\]

Now consider the first partial derivative of \( \Phi_2 \) with respect to \( \ell' \);
\[
\frac{\partial \Phi_2}{\partial \ell'} = -D_t \ell^2.
\]

Therefore, \( \Phi_2 \) is a decreasing function for all \( \ell' \in \mathbb{N}^+ \) such that \( \ell' \leq [\ell_2] - 1 \). Thus, the \( \Phi_2 \) evaluated at \([\ell_2] - 1 \) is an upper bound for all such \( \ell' \); i.e., for all \( \ell' \leq [\ell_2] - 1 \),
\[
\Phi_2(L^i_t, \ell', \Pi, \Delta) \leq \Phi_2(L^i_t, [\ell_2] - 1, \Pi, \Delta).
\]

The above inequality implies that \( \Theta \geq \Phi_2(L^i_t, [\ell_2] - 1, \Pi, \Delta) \), satisfying Equation (29b) of Lemma 3. For \([\ell_2] - 1 \), Case II holds, implying that Equations (29a) and (29c) must also hold. Thus, by Lemma 3, \( (t, D_t) \in \mathcal{F}_{[\ell_2]-1}(\Pi, \Delta, \Theta) \).

The next corollary follows from the above lemma and the definition of \( \Theta^*_t \) (Definition 5).

Corollary 2. For any \( t_a, t_{a+1} \in \hat{S}_t(\tau, k) \), \( \ell' \in \mathbb{N}^+ \), \( \Pi, \Delta \), if \( \ell' \leq [\ell_2] - 1 \) then
\[
\Theta^*_t(\Pi, \Delta, L^i_t) \geq \Theta^*_t([\ell_2] - 1; \Pi, \Delta, L^i_t).
\]

Lemma 11. For any \( t_a, t_{a+1} \in \hat{S}_t(\tau, k) \), \( (t, D_t) \in L^i_t \), \( \ell' \in \mathbb{N}^+ \), \( \Pi, \Delta \), and \( \Theta \), if \([\ell_2] \leq \ell' \leq [\ell_1] \) then
\[
[(t, D_t) \in \mathcal{F}_{\ell'}(\Pi, \Delta, \Theta)] \Rightarrow [(t, D_t) \in \mathcal{F}_{\ell_1}(\Pi, \Delta, \Theta)] \cup [(t, D_t) \in \mathcal{F}_{[\ell_2]}(\Pi, \Delta, \Theta)].
\]

Proof: By Lemma 8 and \([\ell_2] \leq \ell' \leq [\ell_1] \), Case III must hold for all such \( \ell' \). Combining Case III and Lemma 3, we have that if \( (t, D_t) \in \mathcal{F}_{\ell'}(\Pi, \Delta, \Theta) \), then
\[
\Theta \geq \Phi_3(L^i_t, \ell', \Pi, \Delta).
\]

Now consider the first partial derivative of \( \Phi_3 \) with respect to \( \ell' \);
\[
\frac{\partial \Phi_3}{\partial \ell'} = \frac{\alpha^2 \Pi - \hat{D}_t - \alpha \Delta + \alpha t_a}{(\ell + \alpha)^2}.
\]

Note the sign of the above partial derivative is independent of the value of \( \ell' \); therefore, either \( \frac{\partial \Phi_3}{\partial \ell'} \leq 0 \), or \( \frac{\partial \Phi_3}{\partial \ell'} > 0 \) for any \( \ell \in \mathbb{N}^+ \); in other words, the sign remains constant for all \( \ell \). If \( \frac{\partial \Phi_3}{\partial \ell'} > 0 \), then \( \Phi_3 \) is minimized when \( \ell \) is as small as possible; i.e., \( \ell \) equals \([\ell_2] \). In this case, the \( \Phi_3 \) evaluated at \([\ell_2] \) is a lower bound for all such \( \ell' \); i.e., for all \( \ell' \) such that \([\ell_2] \leq \ell' \leq [\ell_1] \),
\[
\Phi_3(L^i_t, \ell', \Pi, \Delta) \geq \Phi_3(L^i_t, [\ell_2], \Pi, \Delta).
\]

The above inequality implies that \( \Theta \geq \Phi_3(L^i_t, [\ell_2], \Pi, \Delta) \), satisfying Equation (29c) of Lemma 3. For \([\ell_2] \), Case III holds, implying that Equations (29a) and (29b) must also hold. Thus, by Lemma 3, \( (t, D_t) \in \mathcal{F}_{[\ell_2]}(\Pi, \Delta, \Theta) \) when \( \frac{\partial \Phi_3}{\partial \ell'} > 0 \).
If \( \frac{\partial \Phi_3}{\partial t} \leq 0 \), then \( \Phi_3 \) is minimized when \( t \) is as large as possible; i.e., \( t \) equals \( \lfloor \ell_1 \rfloor \). In this case, the \( \Phi_3 \) evaluated at \( \lfloor \ell_1 \rfloor \) is an upper bound for all such \( \ell' \); i.e., for all \( \ell' \) such that \( \lfloor \ell_2 \rfloor \leq \ell' \leq \lfloor \ell_1 \rfloor \),

\[
\Phi_3(L^i_{\tau, t}, \ell', \Pi, \Delta) \geq \Phi_3(L^i_{\tau, t}, \lfloor \ell_1 \rfloor, \Pi, \Delta).
\]

By the same argument for \( \frac{\partial \Phi_3}{\partial t} > 0 \), \( (t, D_\ell) \in \mathcal{F}_\ell(\Pi, \Delta, \Theta) \) when \( \frac{\partial \Phi_3}{\partial t} \leq 0 \).

The next corollary follows from the above lemma and the definition of \( \Theta_\ell^* \) (Definition 5).

**Corollary 3.** For any \( t_a, t_{a+1} \in \hat{S}_\ell(\tau, k), \ell' \in \mathbb{N}^+, \Pi, \text{ and } \Delta \), if \( (\lfloor \ell_2 \rfloor \leq \ell' \leq \lfloor \ell_1 \rfloor) \) then

\[
\Theta_\ell^*(\Pi, \Delta, L^i_{t_a}) \leq \min\{\Theta_\ell^*(\Pi, \Delta, L^i_{t_b}), \Theta_\ell^*(\Pi, \Delta, L^i_{t_c})\}.
\]

Combining Corollaries 1, 2, 3, and using Definitions 4 and 5, we obtain the following corollary.

**Corollary 4.**

\[
\Theta^*(\Pi, \Delta, L^i_{t_a}) = \min_{\ell \in \{\lfloor \ell_1 \rfloor, \lfloor \ell_2 \rfloor, \lfloor \ell_2 \rfloor - 1\}} \{\Theta_\ell^*(\Pi, \Delta, L^i_{t_a})\}.
\]

**Proof:** By Lemma 5, we may determine \( \Theta^*(\Pi, \Delta, L^i_{t_a}) \) by evaluating \( \Theta_\ell^*(\Pi, \Delta, L^i_{t_a}) \) for all possible \( \ell \in \mathbb{N}^+ \). The corollary follows by applying Corollaries 1, 2, and 3, respectively, for the following regions of \( \ell \): \( [1, \lfloor \ell_2 \rfloor - 1], \lfloor [\ell_2], [\ell_1] \rfloor \), and \( \lfloor [\ell_1] + 1, \infty \) by applying Lemmas 7 and 8, respectively.

The final lemma that we prove before providing a proof for Theorem 2 shows that a choice of \( \Theta \) based on the computation of \( \Theta^*(\cdot) \) is a “safe” choice in the sense that all tasks in \( \tau_\ell \) will complete by their deadline under an EDP resource \( \Omega = (\Pi, \Theta, \Delta) \).

**Lemma 12.** For any \( L^i_{t_a} \) and \( \ell \in \mathbb{N}^+ \),

\[
\Theta_\ell^*(\Pi, \Delta, L^i_{t_a}) = \begin{cases} 
\Phi_1(L^i_{t_a}, \ell, \Pi, \Delta), & \text{if } \ell \geq \lfloor \ell_1 \rfloor + 1; \\
\Phi_2(L^i_{t_a}, \ell, \Pi, \Delta), & \text{if } \ell \leq \lfloor \ell_2 \rfloor - 1; \\
\Phi_3(L^i_{t_a}, \ell, \Pi, \Delta), & \text{otherwise.}
\end{cases}
\]

**Proof:** From Definition 5, \( \Theta_\ell^*(\Pi, \Delta, L^i_{t_a}) \) is the minimum \( \Theta \leq \Delta \) such that there exists \( (t, D_\ell) \in L^i_{t_a} \) where \( (t, D_\ell) \in \mathcal{F}_\ell(\Pi, \Theta, \Delta) \). By Lemma 3, such a \( \Theta \) is also the minimum value that satisfied all three conditions of Equation (29). Since each of the conditions is a lower bound on \( \Theta \) (with equality permitted), \( \Theta \) must satisfy equality of at least one of the three conditions of Equation (29) and must exceed or equal the other two conditions. Notice that, if \( \ell \geq \lfloor \ell_1 \rfloor + 1 \), then by Lemma 6, \( \Theta \) equals \( \Phi_1(L^i_{t_a}, \ell, \Pi, \Delta) \). We can show an identical proof for intervals \( (0, \lfloor \ell_2 \rfloor - 1) \) and \( \lfloor [\ell_2], [\ell_1] \rfloor \), by applying Lemmas 7 and 8, respectively.

Now, consider the case when there exists a \( \tau_\ell \in \tau \) such that \( \hat{R}_\ell(t) > sfb((\Pi, \Theta, \Delta), t) \) for all \( t \in (0, d_\ell) \). By Lemma 4, this implies for all \( \ell \in \mathbb{N}^+ \), \( t_a, t_{a+1} \in \hat{S}_\ell(\tau, k) \), and \( (t, D_\ell) \in L^i_{t_a} \) that \( (t, D_\ell) \notin \mathcal{F}_\ell(\Pi, \Delta, \Theta) \). By Definition 5, it must be for all \( \ell \in \mathbb{N}^+ \) that \( \Theta < \Theta_\ell^*(\Pi, \Delta, L^i_{t_a}) \). By Lemma 5, this implies that \( \Theta < \Theta^*(\Pi, \Delta, L^i_{t_a}) \) for any \( t_a, t_{a+1} \in \hat{S}_\ell(\tau, k) \), which violates the inequality of Equation (48) due to the first term in the outer max. For
the "only if" direction of the lemma, we will also consider the contrapositive. The contrapositive will follow by simply reversing the implications of the proof for the "if" direction.

After proving the above conditions, we are ready to prove Theorem 2 which states that \textsc{FPMinimumCapacity} returns a valid value for finite \( k \) and an exact value for \( k = \infty \).

\textbf{Proof of Theorem 2} We will show that \( \Theta^\text{min} \) returned from \textsc{FPMinimumCapacity} corresponds to the value on the right-hand side of Equation (48) of Lemma 13. The loop from Line 4 to 17 iterates through each consecutive pair of values \( t_a \) and \( t_{a+1} \) in \( \hat{S}_i(\tau, k) \) to find optimal capacity for each line segment defined by the endpoints \((t_a, D_{t_a})\) and \((t_{a+1}, D_{t_{a+1}})\). It sets variables corresponding to \( \hat{R}_i(t_a) \) and \( \hat{R}_i(t_{a+1}) \) in Lines 5 and 6 respectively. Then, in the next few lines it sets four different values to \( \ell \) (based on \( \ell_1 \) and \( \ell_2 \), defined in Equations (25) and (26)) and evaluates \( \Phi_j(\cdot) \) according to Lemma 12 to compute \( \Theta^*_j(\cdot) \) for each of the four integer values of \( \ell \). Therefore, \( \Theta_{t_a}^\text{min} \), set in Line 15, equals \( \Theta^*(\Pi, \Delta, \tau, \bar{L}_{t_a}^i) \) by Lemma 5. At the end of this loop it sets \( \Theta_{t_a}^\text{min} \) to be the minimum of all \( \Theta_{t_a}^\text{min} \) values. The outer loop from Line 2 to Line 19 finds \( \Theta^\text{min} \) for all task \( \tau_i \) in \( \tau \). Finally, in Line 18, \( \Theta^\text{min} \) is set to the maximum of \( U(\tau) \cdot \Pi \) and \( \Theta_{t_a}^\text{min} \) over all values \( \tau_i \) in \( \tau \).

By Lemma 13, \( \hat{R}_i(t) \leq \text{sbff}(\Pi, \Theta^\text{min}, \Delta, t) \) for some \( t \in (0, d_i) \) and \( U(\tau) \leq \frac{\Theta}{\Pi} \). By Lemma 1, \( R_i(t) \leq \hat{R}_i(t) \). This implies \( R_i(t) \leq \text{sbff}(\Pi, \Theta^\text{min}, \Delta, t) \) which is the schedulability condition given by Theorem 1. Therefore, \( \tau \) will always meet all deadlines when scheduled by fixed-priority scheduling upon \( \Omega = (\Pi, \Theta^\text{min}, \Delta) \). When \( k = \infty \), \( \hat{R}_i(t) \) equals \( R_i(t) \) for all \( t \geq 0 \); in this case, \( \Theta^\text{min} \) equals \( \Theta^*(\Pi, \Delta, \tau) \) (i.e., \( \Theta^\text{min} \) is exact capacity).

\section{An Approximation Scheme}

In the previous section, we have shown that \textsc{FPMinimumCapacity} gives a valid answer when \( k \) is finite and an exact answer when \( k \) is infinite. In this section, we show that as \( k \) increases, the guaranteed accuracy of \textsc{FPMinimumCapacity} increases along with its running time. Theorem 3 presents the tradeoff between accuracy and computational complexity, in terms of \( k \).

\textbf{Theorem 3.} Given \( \Pi, \Delta, \tau \), and \( k \in \mathbb{N}^+ \), the procedure \textsc{FPMinimumCapacity} returns \( \Theta^\text{min} \) such that

\[ \Theta^*(\Pi, \Delta, \tau) \leq \Theta^\text{min} \leq \left( \frac{k+1}{k} \right) \cdot \Theta^*(\Pi, \Delta, \tau). \]

Furthermore, \textsc{FPMinimumCapacity} \((\Pi, \Delta, \tau, k)\) has time complexity \( O(kn^2 \log n) \).

The following corollary quantifying our FPTAS is immediately obtainable from Theorem 3, by substituting a value for \( k \) dependent on the accuracy parameter \( \epsilon \) (\( k = \left\lfloor \frac{1}{\epsilon} \right\rfloor \)).

\textbf{Corollary 5.} Given \( \Pi, \Delta, \tau \), and \( \epsilon > 0 \), the procedure \textsc{FPMinimumCapacity} \((\Pi, \Delta, \tau, \left\lfloor \frac{1}{epsilon} \right\rfloor)\) returns \( \Theta^\text{min} \) such that

\[ \Theta^*(\Pi, \Delta, \tau) \leq \Theta^\text{min} \leq (1 + \epsilon) \cdot \Theta^*(\Pi, \Delta, \tau). \]

Furthermore, \textsc{FPMinimumCapacity} \((\Pi, \Delta, \tau, \left\lfloor \frac{1}{epsilon} \right\rfloor)\) has time complexity \( O \left( \frac{n^2 \log n}{\epsilon} \right) \).

To prove Theorem 3, we need to prove two additional lemmas.

\textbf{Lemma 14.} Given \( \Pi, \Delta, \) and consecutive pair of values \( t_a, t_{a+1} \in \hat{S}_i(\tau, k) \), the following is true for all \( k, \ell(\in \mathbb{N}^+) \), and \( \alpha(\in [0,1]), \)

\[ \Theta^*_\ell(\Pi, \Delta, L_{t_a}^i) \leq \left( \frac{k+1}{k} \right) \cdot \Theta^*_\ell(\Pi, \Delta, \left\langle \left( t_a, k \frac{D_{t_a}}{k+1} \right), \left( t_{a+1}, k \frac{D_{t_{a+1}}}{k+1} \right), k \alpha, \frac{k+1}{k+\alpha} \right\rangle). \]  

\textbf{Proof:} By Lemma 12, \( \Theta^*_\ell(\Pi, \Delta, L_{t_a}^i) \) must be equal to one of \( \Phi_1, \Phi_2 \) or \( \Phi_3 \) according to the value of \( \ell \). We will show that for each of the three possibilities, Equation (49) must hold.

20
In this case, Equation (49) holds.

Lemma 15. Given \( \Pi, \Delta, \tau_i \in \tau \), and \( k \in \mathbb{N}^+ \), there exists consecutive pair of values \( t_a, t_{a+1} \in \hat{S}_i(\tau, k) \) such that,

\[
\Theta^*(\Pi, \Delta, \tau) \geq \Theta^* \left( \Pi, \Delta, \left\langle \left( t_a, \frac{k \cdot D_{t_a}}{k+1} \right), \left( t_{a+1}, \frac{k \cdot D_{t_{a+1}}}{k+1} \right), \frac{k \cdot \alpha}{k+1} \right\rangle \right).
\]  

(50)
Proof: Let $\Theta_{RHS}$ denote the right-hand side of Equation (50). By definition of $\Theta^*(\Pi, \Delta, \tau)$ and Theorem 1, for all $\tau_i \in \tau$, there exist $t \in (0, d_i]$ such that
\[
R_i(t) \leq \text{sbf}((\Pi, \Theta^*(\Pi, \Delta, \tau), \Delta), t).
\] (51)

Now consider any pair of consecutive values $t_a, t_{a+1} \in \widehat{S}_i(\tau, k)$. By Lemma 1, we have, for all $t \in (t_a, t_{a+1}]$,
\[
\left(\frac{k+1}{k}\right) \cdot R_i(t) = \left(\frac{k+1}{k}\right) \cdot (e + \sum_{j=1}^{i-1} \text{RBF}(\tau_j, t)) \\
\geq e + \left(\frac{k+1}{k}\right) \cdot \sum_{j=1}^{i-1} \text{RBF}(\tau_j, t) \\
\geq e + \left(\frac{k+1}{k}\right) \cdot \left(\sum_{j=1}^{i-1} \delta(\tau_i, t) \cdot \frac{k}{k+1}\right) \\
= \tilde{R}_i(t)
\] (52)

Combining the inequalities of Equations (51) and (52) gives us, for all $t \in (t_a, t_{a+1}]$,
\[
\text{sbf}((\Pi, \Theta^*(\Pi, \Delta, \tau), \Delta), t) \geq \frac{k}{k+1} \cdot \tilde{R}_i(t).
\] (53)

Lemma 4 and Equation (53) imply that there exists $\ell \in \mathbb{N}$ and $(t, D_\ell) \in \left\{\left(t_a, \frac{k \cdot D_a}{k+1}\right), \left(t_{a+1}, \frac{k \cdot D_{a+1}}{k+1}\right), \left(t_a, \frac{k \cdot D_a}{k+1}\right)\right\}$ such that
\[
(t, D_\ell) \in F(\Pi, \Theta^*(\Pi, \Delta, \tau), \Delta).
\]

The above expression and Definition 5 implies
\[
\Theta^*_\ell (\Pi, \Delta, \left(\left(t_a, \frac{k \cdot D_a}{k+1}\right), \left(t_{a+1}, \frac{k \cdot D_{a+1}}{k+1}\right), \left(t_a, \frac{k \cdot D_a}{k+1}\right)\right)) \leq \Theta^*(\Pi, \Delta, \tau).
\]

The lemma follows from the expression above and Lemma 5.

We find the following corollary by combining Lemmas 14, 15 and 5.

**Corollary 6.** Given $\Pi, \Delta, k \in \mathbb{N}^+$, and $\tau_i$, there exists consecutive pair of values $t_a, t_{a+1} \in \widehat{S}_i(\tau, k)$,
\[
\left(\frac{k+1}{k}\right) \cdot \Theta^*(\Pi, \Delta, \tau) \geq \inf_{\ell \in \mathbb{N}^+} \left\{\Theta^*_\ell (\Pi, \Delta, \mathcal{L}^i_a)\right\}.
\] (54)

Now, we are ready to give the proof of Theorem 3.

**Proof of Theorem 3** We already proved the first part in Theorem 2; now we must prove the second part of the inequality. From our algorithm, the value of $\Theta^{\min}$ can be either equal to $\Pi \cdot U(\tau)$ or greater than this term. If $\Theta^{\min} = \Pi \cdot U(\tau)$, Theorem 1 implies that $\Theta^*(\Pi, \Delta, \tau)$ must be at least $U(\tau) \cdot \Pi$. For this case, the second inequality follows, since $\frac{k+1}{k} \geq 1$ for all $k \in \mathbb{N}^+$. Now consider the case when $\Theta^{\min} > \Pi \cdot U(\tau)$.

\[
\Theta^{\min} = \max_{\tau_i \in \tau} \left\{\min_{t_a, t_{a+1} \in \widehat{S}_i} \Theta^*(\Pi, \Delta, \mathcal{L}^i_a)\right\}
\]

according to Theorem 2 and Lemma 13. By Lemma 5, this is equivalent to
\[
\Theta^{\min} = \max_{\tau_i \in \tau} \left\{\min_{t_a, t_{a+1} \in \widehat{S}_i} \left\{\inf_{\ell \in \mathbb{N}^+} \left\{\Theta^*_\ell (\Pi, \Delta, \mathcal{L}^i_a)\right\}\right\}\right\}.
\]

Applying Corollary 6, we find,
\[
\Theta^{\min} \leq \max_{\tau_i \in \tau} \left\{\left(\frac{k+1}{k}\right) \cdot \Theta^*(\Pi, \Delta, \tau)\right\}.
\]

From this and the definition of $\Theta^*(\Pi, \Delta, \tau)$ the second inequality of this theorem follows.
7 Simulations

In this section, we present simulation results and compare the performance of our proposed algorithms. We implemented six schedulability tests: exact test derived in Section 4 without any heuristics (i.e., iterative convergence to determine response time in Equation 13); exact test with heuristics (using response time lower bound and upper bound derived in Section 4.1 and 4.2); exact algorithm by [14]; our proposed approximate algorithm FP-MINIMUMCAPACITY; iterative convergence-based approximate test with heuristics and sufficient algorithm by [31]. We denote these algorithms as BS-E, BS-E-h, MC-E, MC-A, BS-A-h and Suff respectively in the plots. The simulation parameters and value ranges are shown below:

1. The number of tasks in a task system \( \tau \) is \([4, 60]\) at 4-increments.
2. The system utilization \( U(\tau) \) is taken from the range \([0.1, 0.9]\) at 0.05-increments and individual task utilizations \( u_i \) are generated using UUniFast algorithm [8].
3. Each sporadic task \( \tau_i = (e_i, d_i, p_i) \) has a period \( p_i \) uniformly drawn from the interval \([10, 10000]\). The execution time \( e_i \) is set to \( u_i, p_i \). We assume \( d_i \leq p_i \) and is uniformly drawn from the interval \([e_i, p_i]\).
4. The component level scheduling algorithm is FP.
5. \( k \) is taken from the range \([1, 25]\) at 2-increments. \( \Pi \) is set in the range \([10, 10000]\); \( \Delta \) is equal to \( \Pi \).
6. A 2.33 GHz Intel Core 2 Duo E6550 machine with 2.0GB RAM is used for simulations.
7. Each point in the plots represents mean of 1000 simulation runs with 95% confidence intervals.

For each simulation, given task system size \( n \) and system utilization \( U(\tau) \), we randomly generate taskset parameters \( u_i, p_i, e_i \) and \( d_i \) for each task \( \tau_i \). We execute three exact algorithms, two approximate algorithms, and the sufficient algorithm to generate exact, approximate and sufficient capacity, respectively. We first compare the relative error\(^4\) of our proposed approximate algorithm (MC-A) with the sufficient algorithm (Suff), and iterative convergence-based approximate algorithm (BS-A-h), with respect to the exact algorithm (MC-E)\(^5\).

7.1 Implicit deadline task system with component-level scheduler DM

For the simulation results shown in this section, we assume an implicit-deadline task system \( (d_i = p_i) \) with deadline-monotonic (DM) component-level scheduler. Since the sufficient algorithm Suff is defined for these constraints, in this section we compare the performances of all six algorithms.

---

\(^3\)Note that we can obtain the exact capacity from FP-MINIMUMCAPACITY with \( k = \infty \).

\(^4\)Relative error is defined as follows:\( \frac{\Theta^* - \Theta}{\Theta^*} \) where \( \Theta^* \) is the exact capacity and \( \Theta \) is either the sufficient capacity \( \bar{\Theta} \) or the approximate capacity \( \hat{\Theta} \).

\(^5\)Note that the relative error of BS-E, BS-E-h and BS-E-h are equal to the threshold (equals \( 10^{-6} \) in the simulations) of the binary search used to determine minimum capacity.
In Figure 11, the relative error in the calculation of capacity for our algorithm is plotted as a function of task system utilization ($n = 20; \Pi = 100; \Delta = \Pi; k = 3$). For MC-A, the mean relative error is less than 5%, whereas for Suff it ranges from 40% to 85%. For the sufficient algorithm, relative error is very high due to the fact that the algorithm overestimates capacity. The relative error for our approximation algorithm does not vary much with the increase in system utilization, where as it decreases for the sufficient algorithm. A potential explanation for this is that some of the functions of Suff for setting the capacity do not depend on the utilization, only the task and resource period parameters. Such functions will be constant over increasing utilization while the optimal capacity must increase as utilization increases. This results in a reduction in the relative error of these (non-utilization-dependent) functions. This observation continues to hold for the next two plots where we compare relative error by varying workload size (Figure 12) and resource period (Figure 13). We observe that for Suff, the relative error ranges from 25% – 95% and it increases with the increase of workload size and resource period. For MC-A, relative error for both these cases are below 5%, and it is independent of workload size and resource period. In Figure 14, we compare the relative error of the two approximation algorithms (MC-A and BS-A-h) by varying the approximation parameter $k$. For both the cases, we observe that the relative error is very low (below 1%) even for moderate value of $k$ ($\geq 5$). The relative error for BS-A-h is slightly higher than MC-A in all the above cases due to the fact that in the former case we have used a threshold of $10^{-6}$ while performing binary search of minimum capacity $\Theta$. Further, in BS-A-h, we approximate the special tasks which represent the resource unavailability period which results a slight overestimation of minimum capacity by this algorithm.

Next, we compare the execution time (in ms) for all six algorithms varying system utilization, workload size and resource period. The graphs show that for MC-A, the execution time is consistently lower than that of Suff, which overestimates capacity and hence takes more time. BS-A-h has a slightly higher execution time than MC-A but lower than Suff. In Figure 12, we see that the execution time increases with the increase in system utilization. Figure 13 shows that the execution time increases with the increase in workload size. Finally, Figure 14 shows that the execution time increases with the increase in resource period.
resource period. In Figure 8, execution time for these algorithms are plotted against system utilization. We observe that for the iterative convergence-based algorithms (BS-E, BS-E-h and BS-A-h), execution time decreases with increasing system utilization. This is due to the fact that the initial values of the response time for each task in task system $\tau$ (see Equation 13) is higher at the initial steps of the iterative convergence algorithm. This results in fewer number of iterations for the response time to converge, and thus reduces overall execution time of the algorithm. The execution time for the other three algorithms (i.e., MC-E, MC-A, Suff) does not vary much with respect to utilization. Notice that the comparison between the two heuristics based iterative algorithms: the approximate algorithm BS-A-h performs worse than the exact algorithm BS-E-h. This is due to the fact that the approximate response time obtained by BS-A-h is higher than that of the exact response time obtained by BS-E-h which results the former algorithm taking more time to converge. This is not trivial since at each iteration, the response time calculation for the approximate algorithm takes less time than the exact algorithm. We observe similar run-time performance of BS-A-h in the last two plots too.

In Figure 9, we compare execution time for all algorithms varying workload size, and observe that as the number of tasks in the system increases, the execution time for all the algorithms except the constant-time sufficient algorithm increases. However, the execution time for MC-E grows at higher rate than BS-E-h, and crosses it at around $n = 48$. This is due to the fact that MC-E calculates minimum capacity for each point in the testing set, and the size of testing set grows pseudo-polynomially with the workload size (Equation 3), whereas the growth of the iterative exact algorithms (BS-E, BS-E-h and BS-A-h) is proportional to the workload size. Thus, we may conclude that the heuristic-based exact algorithm (BS-E-h) is more suitable when workload size is high. Finally, in Figure 10 execution time is compared against resource period. Here we observe that MC-E, MC-A and Suff do not vary much with resource period as expected. However, execution time for iterative exact algorithms decrease with increasing period, due to large initial response time for larger value of $\Pi$.

### 7.2 Constrained deadline task system with component-level scheduler FP

In the next few plots, we demonstrate simulation results for arbitrary fixed priority scheduled constrained deadline ($d_i \leq p_i$) task system. Note that we excluded the sufficient algorithm in these plots.

The first four plots in this section compares relative error of the two approximation algorithms MC-A and BS-A-h. We observe that their behavior is very similar to the implicit deadline case (previous section), i.e., MC-A has better accuracy than BS-A-h.

In the last three plots, execution time of the exact and the approximate algorithms have been compared. In Figure 15, we observe that the execution time for the iterative algorithms are either indifferent (BS-E) or increases (BS-E-h and BS-A-h) with increasing utilization. Figure 16 and 17 exhibit similar behavior as of Figure 9 and 10. Although for moderate task system size the execution time of MC-E is very competitive to MC-A (Figure 8), while determining the interface parameters using the capacity determination algorithm as a subroutine of the period selection algorithm [16], it significantly adds up to the time required to determine interface parameters. Therefore, we can use the near-optimal approximate capacity determination algorithm MC-A with very low (less than 1% for $k = 5$) relative error. In this article, we have considered constrain-deadline sporadic tasks in which case the execution time of the exact algorithm (MC-E) is proportional to $d_{\text{max}}$ (i.e., maximum relative deadline among tasks). However, when task deadlines are arbitrary, this condition no longer holds and an approximation algorithm will perform much better than the exponential time exact algorithm [12].
8 Conclusions and Future Work

In this paper, we have addressed the minimization of interface bandwidth (MIB-RT) problem of explicit-deadline periodic (EDP) resource model in the context of compositional real-time systems with fixed-priority component level scheduling algorithms. In this setting, we have explored two approaches of fixed-priority scheduling: response time based analysis and testing set based analysis. For the former approach we derived an efficient bandwidth allocation scheme using dedicated uniprocessor schedulability test by modeling the resource unavailability period of the periodic resource model as special higher priority tasks. For the latter approach, we have addressed the time complexity of the exact test [14] by devising a fully polynomial time approximation scheme (FPTAS). In this model, given fixed period and deadline of the EDP resource, for any sporadic task system our algorithm returns bandwidth that is at most a factor of \(1 + \epsilon\) greater than the optimal minimum bandwidth, for any \(\epsilon > 0\). We showed that our algorithm has a polynomial time complexity in terms of the number of tasks in the task system \(n\) and the approximation parameter \(1/\epsilon\), whereas exact algorithms for MIB-RT problem on fixed-priority periodic resources may require pseudo polynomial time [31, 14] depending on the task parameters (i.e., task deadline or period) of the task system. We verified our result by running simulation over synthetically generated tasks, and showed that our approximation algorithm improves performance over the sufficient algorithm [31] by effectively reducing relative error. Further, the algorithm closely approximates the bandwidth from the exact algorithm regardless of the task parameters while maintaining polynomial time complexity.

Our result in this paper is for fixed-priority scheduling algorithms of constrained deadline sporadic task systems. Future direction from this paper may be to extend this work to more general task models such as fixed-priority task
system with arbitrary deadlines, hybrid priority task systems, generalized multiframed task model etc. Furthermore, the approximation algorithms for MIB-RT on uniprocessor frameworks may be applicable to multiprocessor compositional frameworks (e.g., [28]) as well as to other compositional resource models.

Acknowledgments

This article is a revise and extended version of [11] presented at the 16th Real-Time and Embedded Technology and Applications Symposium (RTAS). This research has been supported in part by a National Science Foundation CAREER Award (CNS-0953585) and a grant from Wayne State University’s Office of the Vice President for Research. We are sincerely thankful to anonymous reviewers for the detailed comments, corrections, and insightful directions which helped us to improve the quality of the article.

References


