Tighter Bounds of Speedup Factor of Partitioned EDF for Constrained-Deadline Sporadic Tasks

Xingwu Liu
School of Mathematical Sciences,
Dalian University of Technology
Email: liuxingwu@dlut.edu.cn

Zizhao Chen
SK-Lab of Media Convergence Production
Technology and Systems, Chinaso Inc.
Email: chenzizhao@chinaso.com

Xin Han*
School of Software Technology
Dalian University of Technology, China
Email: hanxin@dlut.edu.cn

Zhenyu Sun
SK-Lab of Computer Architecture, ICT, CAS
University of CAS, Beijing, China
Email: sunzhenyu@ict.ac.cn

Zhishan Guo*
Department of Electrical and Computer Engineering
University of Central Florida, USA
Email: zsguo@ucf.edu

2021 IEEE Real-Time Systems Symposium (RTSS) | 978-1-6654-2802-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/RTSS52674.2021.00046

***Abstract*—Even though earliest-deadline-first (EDF) is optimal in terms of uniprocessor schedulability, it is co-NP-hard to precisely verify uniprocessor schedulability for constrained-deadline task sets. The most efficient way to solve this problem in polynomial time is via a partially linear approximation of the demand bound function. Such approximation leads to a simple uniprocessor schedulability testing with speedup factor ρ. Such a result further leads to Deadline-Monotonic Partitioned-EDF on multi-processors with speedup factor of $1 + \rho - 1/m$ (where m is the number of processors). The current state of the art results indicate that ρ is within the range $[1.5, 14/9]$. Especially, it has been a conjecture that $\rho = 1.5$.**

This paper improves the range of ρ to $(1.5026, 1.5380)$. The improved lower bound disproves the conjecture of lower bound 1.5. A novel technique is to construct an auxiliary function that is larger than the approximate demand bound function but keeps the supremum ρ unchanged. It solves the dilemma that beating the lower bound 1.5 requires extremely large task sets, while the large size makes it difficult to check the schedulability. This technique not only enables us to disprove 1.5 by a task set of only eight tasks, but also sheds light on future work in transferring/downsizing task sets and deriving utilization bound based tests for various workload abstraction models, such as DAG tasks.

***Index Terms*—Sporadic tasks, resource augmentation bound, partitioned scheduling, demand bound function**

I. Introduction

Given a finite set of constrained-deadline sporadic tasks, each sequentially releasing infinitely many jobs, the mission of real-time scheduling is to allocate computing resources so that all the jobs complete their executions in a timely manner. It is long-known that EDF is optimal in terms of uniprocessor schedulability [1] [2], i.e., if a task set has a feasible schedule, then all deadlines can be met under EDF priority settings. However, conducting the schedulability test that judges whether the set is schedulable by EDF is co-NP-hard on a uniprocessor platform [3] (for constrained/arbitrary task sets), and even when total utilization is capped strictly below 1 [4]. Although there exists pseudo-polynomial-time approach [5], and attempts have been made in transforming the problem (e.g., [6]), the hardness result implies that it is impossible to decide EDF schedulability exactly in polynomial time, unless P=NP.

As a result, approximation algorithms that run in polynomial time have been actively studied [7] [8] [9] [10]. Usually, these approximation algorithms are *pessimistic*, i.e., always deciding "No" unless some sufficient condition of schedulability is met. The performance of such an algorithm can be measured by the *speedup factor* [11], also known as the resource augmentation bound, which is the smallest $s \geq 1$ such that whenever the algorithm decides "No" with processor speed 1, the task set is not schedulable with processor speed $1/s$. Although potential concerns were raised recently [12] [13] [14], the speedup factor has become a standard theoretical tool for evaluating schedulability tests [15].

Since a fully polynomial-time approximation scheme (FPTAS) with speedup factor $1 + \epsilon$ exists for uniprocessors [8], attention has been shifting towards multiprocessors, especially partitioned scheduling, where each task is scheduled on a dedicated processor. Although it remains NP-hard to determine schedulability of partitioned scheduling even for the implicit-deadline case (where the relative deadline of each task equals to its period), partitioned scheduling has been a hot topic due to its simplicity and wide-applicability to industry systems. Deadline-Monotonic Partitioned-EDF (P^{DM}-EDF in short) proposed by Baruah and Fisher [16] uses a partially linear approximation of the demand bound function to determine per-core schedulability. It is the best-known polynomial-time scheduling algorithm of partitioned style in terms of resource efficiency. However, determining its speedup factor has been a long-term challenge.

A milestone is that the speedup factor of P^{DM}-EDF for constrained deadline tasks on identical multi-processors is proven to be at most $1 + \rho - \frac{1}{m}$ [16]. Here m is the number of processors and ρ, called the relaxation factor as in [17], is the speedup factor of uniprocessor schedulability testing by the piece-wise linearly *approximated demand bound*

*Corresponding author

978-1-6654-2802-6/21/$31.00 ©2021 IEEE
DOI 10.1109/RTSS52674.2021.00046

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

function [8] (see Equation (2) for a formal definition). The connection between ρ and multi-processor scheduling was further strengthened by Chen and Chakraborty (Theorem 2, [18]), who implicitly proved that the lower bound of the speedup factor of P^{DM}-EDF with a random fitting strategy (instead of first-fit, as originally proposed by Baruah and Fisher) is asymptotically equal to $1+\rho$. Hence, ρ in itself deserves a thorough investigation.

Great progress has been made to estimate ρ. For constrained-deadline tasks, ρ is upper bounded by 2 by Baruah and Fisher [16] in 2005, by 1.6322 by Chen and Chakraborty [18] in 2011, and by $14/9$ by Han et al. [19] in 2018, which is the best known result so far. In [17], Liu et al. further showed that the upper bound $14/9$ remains valid for arbitrary-deadline tasks. The lower bound is more challenging. Since Chen and Chakraborty [18] obtained an asymptotic lower bound of $3/2$ a decade ago, no improvement has appeared. Actually, it has been conjectured that this lower bound is tight (oral at the presentation of [18] and written in [17] [19] [20]).

This paper tries to further narrow the range of the relaxation factor ρ, hence better estimating the speedup factor of Partitioned-EDF. Our main contribution lies in three aspects.

1) We prove a new lower bound of 1.5026 for ρ, which is greater than 1.5 that was established a decade ago [18]. Since it has been conjectured that $\rho = 1.5$, the significance of our finding mainly lies in that this conjecture is disproven and a reasonable new conjecture is called for. As a corollary, no matter which fitting strategy is used (including random ones), the asymptotic lower bound of the speedup factor of P^{DM}-EDF for constrained-deadline tasks is improved from 2.5 to 2.5026.
2) We improve the best existing upper bound of ρ from $14/9$ to 1.5380. Accordingly, the speedup factor of P^{DM}-EDF for constrained-deadline tasks is upper-bounded from above by $2.5380-\frac{1}{m}$ instead of the looser $23/9-\frac{1}{m}$. Compared to the best known results, the gap between the lower and upper bound is reduced by a fraction of 33%.
3) We propose a counter-intuitive approach for analyzing speedup factors of schedulability testing. An auxiliary function is constructed that enlarges the approximate demand bound function while preserving the supremum, enabling to observe much tighter lower bounds with small task sets. Using this method, we easily disprove the 1.5 lower bound of ρ by a set of eight tasks (found by brute-force search). On the contrary, to reach the same lower bound by the original demand bound function, the minimal task set we identify is as large as half a million tasks (such a big set is currently impossible to be found by brute-force search). The method may also shed light on estimating the resource augmenting factors of other scheduling problems/approaches in real-time systems, such as handling dependencies (e.g., DAG tasks).

The rest of the paper is organized as follows. Section II provides the models, notations, and preliminary results. Section III proves the lower bound 1.5026 of ρ, while Section IV improves the upper bound of ρ to 1.5380. Section V concludes the work and points out future directions.

II. System Model and Preliminaries

A. Notational Convention

$\mathbb{Z}_+$: the set of positive integers.
$\mathbb{R}_+$: the set of positive real numbers.
$\overrightarrow{\mathbf{p}}$: a vector, whose i-th entry is denoted by p_i.

Throughout this paper, $f(\cdot)$ is the function such that for any real number x,

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ \lfloor x \rfloor + 1 & \text{otherwise} \end{cases}.$$

B. System Model

We consider a set of sporadic real-time tasks, with each task being characterized by a triple (e, d, p)— meaning that the task sporadically releases jobs with inter-arrival time (also called period) at least p, the worst-case execution time (WCET) of any job is at most e, and any job has to be done within relative deadline d after it is released. Task (e, d, p) is called constrained-deadline provided that $d \le p$.

Our ultimate goal is to improve the speedup factor of *Deadline-Monotonic Partitioned-EDF* algorithm. When partitioning tasks to processors, it considers tasks in *non-decreasing order of their relative deadlines* (DM) while using any fit heuristic for bin-packing. The schedulability test applied to each processor leverages the *approximate demand-bound function* (see dbf^*-based test in formula (2)). Please refer to Section III in [16] or Algorithm 1 in [18] for more detail. Note that Baruah and Fisher restricted the fitting strategy to first-fit only, while Chen and Chakraborty [18] pointed out that the speedup factor results hold for any fitting strategy (even random), as far as the order for partitioning the tasks is restricted to DM. Similar properties and restrictions holds for this paper, and we refer to this specific approach as 'P^{DM}-EDF' in short in the rest of the paper—it has the best-known speedup factor and resource efficiency within the whole family of polynomial-time partitioned-EDF algorithms. It is well-known (Lemma 1) that the speedup factor of P^{DM}-EDF can be reduced to that of uniprocessor schedulability test. Hence, we will focus on the uniprocessor case from now on.

Given a finite set τ of tasks, it is called feasible if there exists a schedule such that all jobs can receive executions of their WCETs upon their deadlines (on a unit-speed uniprocessor). Let $n = |\tau|$, and suppose τ consists of tasks $\tau_i = (e_i, d_i, p_i)$, $1 \le i \le n$. According to the result in [5], τ is feasible if and only if its *demand bound function* (dbf) meets

$$dbf(\tau, t) \triangleq \sum_{i=1}^{n} f\left(\frac{t-d_i}{p_i}\right) \cdot e_i \le t \text{ for any } t \in \mathbb{R}_+. \quad (1)$$

Since EDF is *optimal* on uniprocessor [1], Condition (1) is also the exact (necessary and sufficient) schedulability test for uniprocessor EDF scheduling with constrained-deadline sporadic tasks. However, it is computationally hard to check

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

Condition (1), even when total system utilization is bounded strictly below 1 [4].

Hence, a schedulability test based on *approximate demand bound function* (dbf^*) has been proposed:

$$dbf^*(\tau,t) \triangleq \sum_{i=1}^{n} f^*\left(\frac{t-d_i}{p_i}\right)\cdot e_i \le t \text{ for any } t\in\mathbb{R}_+, \quad (2)$$

where $f^*(x) = x+1$ if $x \ge 0$ and $f^*(x)=0$ otherwise.

Then quantity $\rho(\tau)$ is defined as below:

$$\rho(\tau) \triangleq \frac{dbf^*(\tau,d)}{d} \text{ with } d = \max_{1\le i\le n} d_i. \quad (3)$$

Based on $\rho(\tau)$, we define the *relaxation factor* ρ to be

$$\rho \triangleq \sup_{n\in\mathbb{Z}_+} \sup_{|\tau|=n} \rho(\tau), \quad (4)$$

where τ ranges over feasible sets of sporadic tasks. Intuitively, ρ indicates how much the approximate demand bound function deviates from machine-capacity.

C. Preliminaries

Lemma 1 ([17], [19]): The speedup factor of P^{DM}-EDF on constrained-deadline task sets is $1+\rho-\frac{1}{m}$, where m is the number of processors.

Hence analyzing speedup factor of P^{DM}-EDF is reduced to computing ρ. Recently, Han et al. [19] and Liu et al. [17] have made progress on estimating ρ by introducing a series of lossless transformation, which leads to the following lemma.

Lemma 2: The value of ρ remains unchanged even if τ in (4) is further required to consist of tasks $\tau_i = (e_i, d_i, p_i)$ with

$$e_i = 1, d_i = i, p_i \in \mathbb{Z}_+ \text{ for any } 1\le i\le n, \quad (5)$$

where $n = |\tau|$.

Proof: . By Lemmas 2,3,7,8 in [17], ρ equals the optimal value of the following mathematical program:

$$\begin{aligned}
&\sup && \frac{dbf^*(\tau,d_n)}{d_n}, && (MP_3) && (6)\\
&\textit{subject to} && dbf(\tau,t)\le t, \quad \forall t>0 && && (7)\\
& && d_i + p_i > d_n, \quad 1\le i\le n-1, && && (8)\\
& && d_i = e_i + d_{i-1}, \quad 1\le i\le n, && && (9)\\
& && e_i = d_n/n, \quad 1\le i\le n, && && (10)\\
& && n\in\mathbb{Z}^+, e_i, d_i, p_i \in \mathbb{Q}^+, \quad 1\le i\le n. && && (11)
\end{aligned}$$

The **basic idea** of the rest of the proof: we sequentially impose two additional conditions (p_i is a multiple of e_i, and $e_i = 1$, respectively) on MP_3. These conditions preserve the optimum value and force the tasks to have the form as in (5). Specifically, we make two claims.

Claim 1: The optimum value of MP_3 remains unchanged if it is further required that

$$p_i/e_i \in \mathbb{Z}_+ \text{ for } 1\le i\le n. \quad (12)$$

In the proof of Lemma 8 in [17], the number δ can be chosen such that $e_i/\delta \in \mathbb{Z}_+$ and $p_i/\delta\in\mathbb{Z}_+$ for any $1\le i\le n$. That proof remain valid since any δ satisfying $e_i/\delta\in\mathbb{Z}_+$ works. **Claim 1 thus holds**.

Claim 2: The optimum value of MP_3 remains unchanged if it is even further required that

$$e_i = 1 \text{ for } 1\le i\le n. \quad (13)$$

To prove Claim 2, arbitrarily choose a task set $\tau = \{\tau_i = (e_i, d_i, p_i) : 1\le i\le n\}$ satisfying Conditions (7)-(12). Let $e = d_n/n$. Define $\tau' = \{\tau_i' = (e_i', d_i', p_i') : 1\le i\le n\}$ with

$$\begin{aligned}
e_i' &= e_i/e = 1,\\
d_i' &= d_i/e = i,\\
p_i' &= p_i/e \in \mathbb{Z}_+.
\end{aligned}$$

By definition, τ' satisfies Condition (5).

We proceed to show that τ' is a feasible solution to MP_3. It is obvious that Conditions (8)-(11) are satisfied. One only has to show that Condition (7) is met, namely,

$$dbf(\tau',t)\le t, \quad \forall t>0.$$

There are two cases.

Case 1: $t\le n$. Let $i\in\mathbb{Z}_+$ be such that $i-1\le t<i$. Then,

$$dbf(\tau',t) = \sum_{j=1}^{n} f\left(\frac{t-d_i'}{p_i'}\right)\cdot e_j' = \sum_{j=1}^{i-1} e_j' = i-1\le t.$$

Case 2: $t>n$. Arbitrarily fix $1\le i\le n$. Let j_i be the maximal integer such that $d_i' + j_i p_i' \le t$, which is equivalent to $d_i + j_i p_i \le et$. Then we have $dbf(\tau_i, et) = (1+j_i)e$ and $dbf(\tau_i', t) = 1 + j_i$. Recall that τ satisfies Condition (7), so

$$et \ge \sum_{i=1}^{n} dbf(\tau, et) = \sum_{i=1}^{n}(1+j_i)e.$$

Hence,

$$t \ge \sum_{i=1}^{n}(1+j_i) = \sum_{i=1}^{n} dbf(\tau_i', t) = dbf(\tau', t).$$

As a result, both Cases 1&2 hold, meaning that τ' is a feasible solution to MP_3.

Next we show that the objective value is preserved. This follows from the fact that for any $1\le i\le n$,

$$\begin{aligned}
\frac{dbf^*(\tau_i,d_n)}{d_n} &= \left(\frac{d_n-d_i}{p_i}+1\right)\frac{e_i}{d_n}\\
&= \left(\frac{d_n'-d_i'}{p_i'}+1\right)\frac{e_i'}{d_n'}\\
&= \frac{dbf^*(\tau_i',d_n')}{d_n'}.
\end{aligned}$$

Altogether, **Claim 2 also holds** and the lemma is proven. ∎

Lemma 2 enables us to discretize Condition (1). As shown in the following lemma, it suffices to only consider the demand at integer time points.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

Lemma 3: Given any task set τ satisfying (5), Condition (1) is equivalent to

$$\sum_{1\le i\le n} f\left(\frac{t-i}{p_i}\right) \le t \text{ for any } t \in \mathbb{Z}_+. \tag{14}$$

Proof: In the setting of (5), Condition (1) trivially implies Condition (14), so we only show that Condition (14) also leads to Condition (1).

Suppose τ satisfies Condition (14). Arbitrarily fix $t \in \mathbb{R}_+$. It holds that

$$\begin{aligned} & \sum_{1\le i\le n} f\left(\frac{t-i}{p_i}\right) \\ = & \sum_{1\le i\le n} f\left(\frac{\lfloor t\rfloor - i}{p_i}\right) \text{ (since } i, p_i \in \mathbb{Z}_+) \\ \le & \lfloor t \rfloor \text{ (since } \tau \text{ satisfies (14) and } \lfloor t\rfloor \in \mathbb{Z}_+) \\ \le & t \text{ (by definition of } \lfloor\cdot\rfloor). \end{aligned}$$

As a result, τ satisfies Condition (1). ■

In addition, Condition (5) makes it possible to expand Formula (3):

$$\begin{aligned} \rho(\tau) =& \frac{1}{d}\sum_{i=1}^{n} f^*\left(\frac{d-d_i}{p_i}\right)\cdot e_i \\ =& \frac{1}{n}\sum_{i=1}^{n}\left(\frac{n-i}{p_i}+1\right) \\ & \text{(by (Condition 5) and definition of } f^*(\cdot)) \\ =& 1+\sum_{i=1}^{n}\frac{n-i}{np_i}. \end{aligned}$$

This leads to a simplified form of ρ:

$$\rho = 1 + \sup_{n\in\mathbb{Z}_+}\sup_{\vec{\mathbf{p}}\in\mathbb{Z}_+^n}\sum_{1\le i\le n}\frac{n-i}{np_i}, \tag{15}$$

where $\vec{\mathbf{p}}$ satisfies Condition (14).

On this ground, for any $n \in \mathbb{Z}_+$, define

$$\xi_n \triangleq \sup_{\vec{\mathbf{p}}\in\mathbb{Z}_+^n}\sum_{1\le i\le n}\frac{n-i}{np_i}, \tag{16}$$

where $\vec{\mathbf{p}}$ satisfies Condition (14). Let $\xi = \sup_{n\in\mathbb{Z}_+}\xi_n$.

We immediately have a corollary.

Corollary 1: $\rho = \xi + 1$

Estimating ρ is thus reduced to estimating ξ. The rest of this paper is devoted to showing that $0.5026 < \xi < 0.5380$, establishing $1.5026 < \rho < 1.5380$.

III. Improved Lower Bound for ξ

A. Challenges and Insights

Let's begin with the challenges of analyzing ξ directly and insights that can be gained from them.

Recall that ξ is the supremum of the sequence in (16), upon all $\vec{\mathbf{p}}$'s that satisfy Condition (14). The known range of ξ is [0.5, 5/9] according to the state-of-the-art results, and it has been conjectured that 0.5 is the tight lower bound.

Note that for any $n \in \mathbb{Z}_+$, the all-n vector $(n, \cdots, n) \in \mathbb{Z}_+^n$ satisfies Condition (14). Since $\sum_{i=1}^{n}\frac{n-i}{n^2} = \frac{n-1}{2n}$ which is smaller than but converges to 0.5, all-n vectors serve as a certificate of the lower bound 0.5. However, these vectors are not worst for calculating ξ, because one can easily find $\vec{\mathbf{p}} \in \mathbb{Z}_+^n$ such that $\sum_{i=1}^{n}\frac{n-i}{np_i} > \sum_{i=1}^{n}\frac{n-i}{n^2}$. For example, $\vec{\mathbf{p}} = (8, 9, 5, 6, 7, 8, 12)$ satisfies Condition (14), and $\sum_{i=1}^{7}\frac{7-i}{7p_i} > 0.4308 > 3/7 = \sum_{i=1}^{7}\frac{7-i}{7^2}$. As a result, we conjecture that ξ *might* be greater than 0.5.

To search for a certificate, similar to [21], we wrote a computer program and exhaustively enumerated $\vec{\mathbf{p}} \in \mathbb{Z}_+^n$ for n up to 20. Unfortunately, we have failed to find any vector $\vec{\mathbf{p}}$ satisfying both Condition (14) and $\sum_{i=1}^{n}\frac{n-i}{np_i} > 0.5$. Note that the search space for $\vec{\mathbf{p}}$ is m^n, where m, set as 25 in our search, is the size of the search range for each period. Such an exponential growth makes it computationally difficult to search for larger task sets (with $n > 20$).

Such a fact leads to the key insight of this paper:

> As n gets larger, ξ_n is growing so slow that the search space for lower-bounding the supremum ξ becomes too large and computationally difficult to handle. If there is an auxiliary sequence (say, $\{\eta_n\}_{n\in\mathbb{Z}_+}$) *which is element-wise bigger than but has the same supremum with* $\{\xi_n\}_{n\in\mathbb{Z}_+}$, one can easily find a good lower bound of ξ by analyzing $\{\eta_n\}_{n\in\mathbb{Z}_+}$. See Fig. 1 for an intuitive illustration.

Suppose such an auxiliary sequence does exist. Since $\eta_n > \xi_n$ for any n, it might be easier to use brute-force search to find a small m such that $\eta_m > 0.5$. Then we know that $\xi = \sup_{n\in\mathbb{Z}_+}\xi_n = \sup_{n\in\mathbb{Z}_+}\eta_n \ge \eta_m > 0.5$, and the mission is fulfilled.

Hence, in the rest of this section, we will construct an auxiliary sequence which is element-wise bigger than $\{\xi_n\}_{n\in\mathbb{Z}_+}$, and prove that its supremum is exactly ξ.

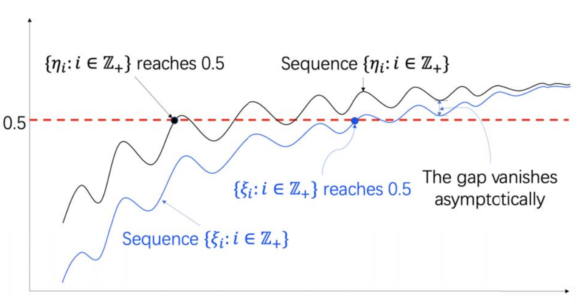

Fig. 1. Although the auxiliary sequence $\{\eta_i\}$ is bigger than $\{\xi_i\}$, both of them has the same supremum. Via $\{\eta_i\}$, it is easier to derive a good lower bound of the supremum.

B. An Auxiliary Sequence

We now propose the auxiliary sequence. $\forall n \in \mathbb{Z}_+$, let

$$\eta_n \triangleq \sup_{\vec{\mathbf{p}}\in\mathbb{Z}_+^n}\sum_{1\le i\le n}\frac{n-i+\frac{1}{2}}{np_i}, \tag{17}$$

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

where $\overrightarrow{\mathbf{p}}$ satisfies Condition (14). Let $\eta \triangleq \sup_{n\in\mathbb{Z}_+} \eta_n$.

One may immediately realize that this auxiliary sequence $\eta_n > \xi_n$ for any $n \in \mathbb{Z}_+$. We now proceed to prove the critical fact that the two sequences share the same supremum, i.e., $\eta = \xi$ (Lemma 6). As a basic step, we prove the following Lemma 4, which intuitively means that $\{\eta_n\}$ is nearly increasing and hence approaches its supremum asymptotically.

Lemma 4: $\eta_{kn} \geq \eta_n$ for any $n, k \in \mathbb{Z}_+$.

Proof: Arbitrarily fix $n, k \in \mathbb{Z}_+$ and $\overrightarrow{\mathbf{p}} \in \mathbb{Z}_+^n$ satisfying Condition (14). Define $\overrightarrow{\mathbf{q}} \in \mathbb{Z}_+^{kn}$ to be such that

$$q_j = kp_{\lceil \frac{j}{k} \rceil} \text{ for any } 1 \leq j \leq kn. \quad (18)$$

Equivalently,

$$kp_i = q_{(i-1)k+l} \text{ for any } 1 \leq i \leq n, 1 \leq l \leq k. \quad (19)$$

The rest of the proof consists of two steps.

Step 1. We prove that $\overrightarrow{\mathbf{q}}$ satisfies Condition (14).

For any $t \in \mathbb{Z}_+$, it holds that

$$\begin{aligned}
& \sum_{1\leq j\leq kn} f\left(\frac{t-j}{q_j}\right) \\
= & \sum_{1\leq l\leq k}\sum_{1\leq i\leq n} f\left(\frac{t-((i-1)k+l)}{q_{(i-1)k+l}}\right) \\
& \text{(since } j \text{ can be uniquely decomposed into} \\
& \quad j=(i-1)k+l \text{ with } 1\leq i\leq n, 1\leq l\leq k) \\
= & \sum_{1\leq l\leq k}\sum_{1\leq i\leq n} f\left(\frac{t+k-l-ik}{kp_i}\right) \text{(by (19))} \\
= & \sum_{1\leq l\leq k}\sum_{1\leq i\leq n} f\left(\frac{\frac{t+k-l}{k}-i}{p_i}\right) \\
= & \sum_{1\leq l\leq k}\sum_{1\leq i\leq n} f\left(\frac{\left\lfloor\frac{t+k-l}{k}\right\rfloor-i}{p_i}\right) \text{(by property of } f(\cdot)) \\
\leq & \sum_{1\leq l\leq k}\left\lfloor\frac{t+k-l}{k}\right\rfloor \text{(by Condition (14) of } \overrightarrow{\mathbf{p}}) \\
= & \ t.
\end{aligned}$$

Step 2. We show that $\overrightarrow{\mathbf{q}}$ makes η_{kn} big enough.

$$\begin{aligned}
\eta_{kn} \geq & \sum_{1\leq j\leq kn}\frac{kn-j+\frac{1}{2}}{knq_j} \text{(by Definition (17))} \\
= & \sum_{1\leq l\leq k}\sum_{1\leq i\leq n}\frac{kn-((i-1)k+l)+\frac{1}{2}}{knq_{(i-1)k+l}} \\
& \text{(since } j \text{ can be uniquely decomposed into} \\
& \quad j=(i-1)k+l \text{ with } 1\leq i\leq n, 1\leq l\leq k) \\
= & \sum_{1\leq l\leq k}\sum_{1\leq i\leq n}\frac{\left(kn+k-ik+\frac{1}{2}\right)-l}{k^2np_i} \text{(by (19))} \\
= & \sum_{1\leq i\leq n}\frac{\sum_{1\leq l\leq k}\left(kn+k-ik+\frac{1}{2}\right)-\sum_{1\leq l\leq k} l}{k^2np_i} \\
= & \sum_{1\leq i\leq n}\frac{\left(kn+k-ik+\frac{1}{2}\right)k-\frac{k(k+1)}{2}}{k^2np_i} \\
= & \sum_{1\leq i\leq n}\frac{n-i+\frac{1}{2}}{np_i}
\end{aligned}$$

Hence, $\eta_{kn} \geq \sum_{1\leq i\leq n}\frac{n-i+\frac{1}{2}}{np_i}$ holds for an any $\overrightarrow{\mathbf{p}}$ satisfying Condition (14). By definition of η_n in (17), we have $\eta_{kn} \geq \eta_n$. ■

Note that Lemma 3 is a very strong result due to arbitrary selection of k's and n's. It indicates that for any item η_j in the sequence, there is an infinite subsequence $\{\eta_{kj} : k \in \mathbb{Z}_+\}$ beyond the j^{th} item with all values no smaller than η_j. As a result, the supremum of the truncated sequence $\{\eta_n : n > j\}$ is also no smaller than η_j.

We then present one more lemma, which intuitively claims that Condition (14) precludes the vectors from having too many small entries.

Lemma 5 ([1]): For any $n \in \mathbb{Z}_+$, if $\overrightarrow{\mathbf{p}} \in \mathbb{Z}_+^n$ satisfies Condition (14), the total utilization satisfies $\sum_{i=1}^{n}\frac{1}{p_i} \leq 1$.

By definition of the sequences $\{\xi_n\}_{n\in\mathbb{Z}_+}$ and $\{\eta_n\}_{n\in\mathbb{Z}_+}$, Lemma 5 immediately implies that

$$\xi_n \geq \eta_n - \frac{1}{2n} \text{ for any } n \in \mathbb{Z}_+. \quad (20)$$

An interesting fact is that enlarging $\{\xi_n\}_{n\in\mathbb{Z}_+}$ to $\{\eta_n\}_{n\in\mathbb{Z}_+}$ preserves the supremum, as shown in Lemma 6.

Lemma 6: $\xi = \eta$.

Proof: Since $\xi_n \leq \eta_n$ for any $n \in \mathbb{Z}_+$, we have $\xi \leq \eta$.

Next we prove that $\xi \geq \eta$, which is equivalent to $\xi \geq \eta - \epsilon$ for any $\epsilon > 0$.

Arbitrarily fix $\epsilon > 0$. By definition of supremum, there is an $n_0 \in \mathbb{Z}_+$ such that $\eta_{n_0} \geq \eta - \frac{\epsilon}{2}$. By Lemma 4, for any $k \in \mathbb{Z}_+$, we have

$$\eta_{kn_0} \geq \eta_{n_0} \geq \eta - \frac{\epsilon}{2}. \quad (21)$$

Arbitrarily fix an integer k such that $kn_0 > \frac{1}{\epsilon}$. Then,

$$\begin{aligned}
\xi_{kn_0} \geq & \ \eta_{kn_0} - \frac{1}{2kn_0} \text{ (by Formula (20))} \\
> & \ \eta_{kn_0} - \frac{\epsilon}{2} \text{ (by definition of } k) \\
\geq & \ \eta - \epsilon. \text{ (by Formula (21))}
\end{aligned}$$

Therefore, $\xi \geq \eta - \epsilon$, as desired. ■

C. An Improved Lower Bound

Facilitated by the auxiliary sequence $\{\eta_n\}_{n\in\mathbb{Z}_+}$, we are ready to present our first main result of this paper.

Theorem 1: $\xi > 0.5026$.

Proof: By Lemma 6, it suffices to show that $\eta_n > 0.5026$ for some $n \in \mathbb{Z}_+$. This turns out to be true when $n = 8$.

Specifically, define $\overrightarrow{\mathbf{p}} \in \mathbb{Z}_+^8$ such that

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

$$p_i = \begin{cases} 12 & \text{if } i = 1 \text{ or } 8 \\ 8 & \text{if } i = 2, 4, \text{ or } 6 \\ 6 & \text{if } i = 3 \text{ or } 5 \\ 9 & \text{if } i = 7 \end{cases}.$$

A straightforward calculation indicates that

$$\sum_{1 \le i \le 8} \frac{8 - i + \frac{1}{2}}{8p_i} > 0.502601.$$

Then we prove that $\overrightarrow{\mathbf{p}}$ satisfies Condition (14). Observe that the least common multiple of $p_1, p_2, ..., p_8$ is 72. As suggested by Baruah et al. [5], it suffices to show that $\sum_{1 \le i \le 8} f\left(\frac{t-i}{p_i}\right) \le t$ for $1 \le t \le 72 + 8 = 80$. This can be verified computationally and straightforwardly via a simple code (available in the full version of this manuscript).

Hence this theorem holds. ∎

Combining Theorem 1 and Corollary 1, we get $\rho > 1.5026$, disproving the conjecture that ρ is 1.5. We can go one step further, showing that $\rho > 1.5026$ remains true for constrained-deadline task sets. This is immediately implied by the following corollary.

Corollary 2: There is $m \in \mathbb{Z}_+$ and $\overrightarrow{\mathbf{q}} \in \mathbb{Z}_+^m$ which satisfies Condition (14) and $q_j \ge j$ for any $1 \le j \le m$, such that

$$\xi_m \ge \sum_{1 \le j \le m} \frac{m - j}{mq_j} > 0.5026.$$

Proof: Let $\epsilon = 10^{-6}, n = 8, k = \frac{1}{2\epsilon n} = \frac{1}{16} \times 10^6, m = kn = \frac{1}{2} \times 10^6$. Recall the vector $\overrightarrow{\mathbf{p}} \in \mathbb{Z}_+^n$ in the proof of Theorem 1. As in Formula (18), define $\overrightarrow{\mathbf{q}} \in \mathbb{Z}_+^m$ such that

$$q_j = kp_{\lceil \frac{j}{k} \rceil} \text{ for any } 1 \le j \le m. \tag{22}$$

Specifically,

$$q_j = \begin{cases} 12k & \text{if } k(i-1) < j \le ki \text{ for } i = 1 \text{ or } 8 \\ 8k & \text{if } k(i-1) < j \le ki \text{ for } i = 2, 4, \text{ or } 6 \\ 6k & \text{if } k(i-1) < j \le ki \text{ for } i = 3 \text{ or } 5 \\ 9k & \text{if } 6k < j \le 7k \end{cases}.$$

Then we prove three facts.

First, by Step 1 of the proof of Lemma 4, $\overrightarrow{\mathbf{q}}$ satisfies Condition (14) because so does $\overrightarrow{\mathbf{p}}$.

Second, for any $1 \le j \le m$,

$$\begin{aligned} q_j &= kp_{\lceil \frac{j}{k} \rceil} \text{ (by definition of } \overrightarrow{\mathbf{q}} \text{ in (22))} \\ &\ge k \times \left\lceil \frac{j}{k} \right\rceil \text{ (since } p_i \ge i \text{ for any } i) \\ &\ge k \times \frac{j}{k} = j. \end{aligned}$$

Finally,

$$\begin{aligned} & \sum_{1 \le j \le m} \frac{m-j}{mq_j} \\ = & \sum_{1 \le j \le m} \frac{m - j + \frac{1}{2}}{mq_j} - \sum_{1 \le j \le m} \frac{1}{2mq_j} \\ \ge & \sum_{1 \le j \le m} \frac{m - j + \frac{1}{2}}{mq_j} - \frac{1}{2m} \quad \text{(by Lemma 5)} \\ = & \sum_{1 \le i \le n} \frac{n - i + \frac{1}{2}}{np_i} - \epsilon \\ & \text{(by Step 2 of the proof of Lemma 4).} \\ > & \ 0.502601 - 10^{-6} = 0.5026. \end{aligned}$$

As a result, m and $\overrightarrow{\mathbf{q}}$ meet the requirement. ∎

Remark 1: Even though η_n exceeds 0.5026 when $n = 8$, the smallest m we find such that $\xi_m > 0.5026$ is as big as $\frac{1}{2} \times 10^6$. This confirms the advantage of the auxiliary sequence $\{\eta_n\}$ in studying the lower bound of ξ.

IV. Improved Upper Bound for ξ

This section is devoted to showing that $\xi < 0.5380$. It suffices to prove that $\sum_{i=1}^{n} \frac{n-i}{np_i} < 0.5380$ for any $\overrightarrow{\mathbf{p}} \in \mathbb{Z}_+^n$ that satisfies Condition (14). This mission is challenging because it is hard to figure out the worst-case $\overrightarrow{\mathbf{p}} \in \mathbb{Z}_+^n$.

Our basic idea comes from the observation that

$$\sum_{i=1}^{n} \frac{n-i}{np_i} = \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{1}{np_i}. \tag{23}$$

The summation domain of the right hand side is the triangular area in Fig. 2(a). We can decompose the domain into three sub-areas A_1, A_2, A_3 via an additional controllable parameter $c < 1$ (as illustrated in Fig. 2(b)), by the curve $\phi_c(x)$ (defined below) and the line $i + k = cn$. To upper bound the value of (23), it suffices to upper bound the summation on each sub-area A_j, namely $B_j \triangleq \sum_{(i,k) \in A_j}$, for any $j \in \{1, 2, 3\}$. When c is close to 1, an easy upper bound of B_1 is precise enough. The curve $\phi_c(x)$ (also defined below) is chosen to be quadratic, since such a curve enables us to find out the vector $\overrightarrow{\mathbf{p}}$ which leads to the precise/tight upper bound of B_3. The main difficulty lies in estimating B_2, but its value gets very small when c is properly chosen. In this case, any not-too-bad upper bound of B_2 works without causing significant error to the upper bound of $\sum_{i=1}^{n} \frac{n-i}{np_i}$.

The flow of the proofs in this section is shown in Fig. 3.

First of all, we formalize the decomposition as in Lemma 7, where the three terms on the right hand side correspond to B_1, B_2, B_3 respectively. The decomposition involves two functions ϕ_c and ψ_c over domain $\mathbb{Z}_+$, where $c \in \mathbb{R}_+$ is a parameter ranging over the interval $(0, 1)$. For any $x \in \mathbb{R}_+$, these functions are defined such that

$$\phi_c(x) = \frac{c(n-x)^2}{n}, \quad \psi_c(x) = (2c - 1)x - \frac{cx^2}{n}.$$

Hereunder, arbitrarily fix $c \in \mathbb{R}_+$ with $\frac{1}{2} < c < 1$.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

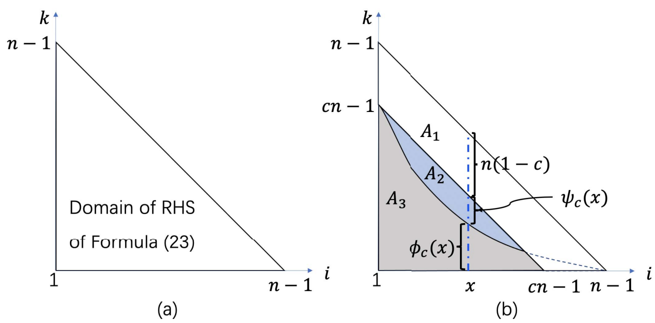

Fig. 2. (a) The domain of the right hand side of Formula (23) is a triangular area. (b) It is decomposed into three sub-areas by a line and a curve $\phi_c(x)$.

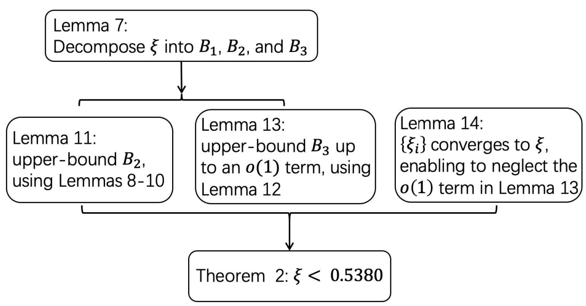

Fig. 3. The flow of the proofs in Section IV

Lemma 7: $\sum_{i=1}^{n} \frac{n-i}{np_i} \leq (1-c)+\sum_{i=1}^{\lfloor nc \rfloor} \frac{\psi_c(i)}{np_i}+\sum_{i=1}^{\lfloor nc \rfloor} \frac{\phi_c(i)}{np_i}$.

Proof: For any $1 \leq i \leq n$, we have

$$\begin{aligned} & n(1-c)+\phi_c(i)+\psi_c(i) \\ = \;& n(1-c)+cn-2ci+\frac{ci^2}{n}+(2c-1)i-\frac{ci^2}{n} \\ = \;& n-i. \end{aligned} \tag{24}$$

Hence,

$$\begin{aligned} & \sum_{i=1}^{n} \frac{n-i}{np_i} \\ = \;& \sum_{i=1}^{\lfloor nc \rfloor} \frac{n-i}{np_i} + \sum_{i=\lfloor nc \rfloor+1}^{n} \frac{n-i}{np_i} \text{ (split of sum)} \\ \leq \;& \sum_{i=1}^{\lfloor nc \rfloor} \frac{n-i}{np_i} + \sum_{i=\lfloor nc \rfloor+1}^{n} \frac{n(1-c)}{np_i} \text{ (range of } i \geq nc) \\ = \;& \sum_{i=1}^{\lfloor nc \rfloor} \frac{n(1-c)+\phi_c(i)+\psi_c(i)}{np_i} + \sum_{i=\lfloor nc \rfloor+1}^{n} \frac{n(1-c)}{np_i} \\ & \text{(from Formula (24))} \\ = \;& \sum_{i=1}^{n} \frac{1-c}{p_i} + \sum_{i=1}^{\lfloor nc \rfloor} \frac{\phi_c(i)}{np_i} + \sum_{i=1}^{\lfloor nc \rfloor} \frac{\psi_c(i)}{np_i} \text{ (merge sums)} \\ \leq \;& (1-c) + \sum_{i=1}^{\lfloor nc \rfloor} \frac{\phi_c(i)}{np_i} + \sum_{i=1}^{\lfloor nc \rfloor} \frac{\psi_c(i)}{np_i}. \\ & \text{(since } \textstyle\sum_{i=1}^{n} \frac{1}{p_i} \leq 1 \text{ by Lemma 5)} \end{aligned}$$

■

A. Upper bounding $\sum_{i=1}^{\lfloor nc \rfloor} \frac{\phi_c(i)}{np_i}$

Then, we will upper-bound $\sum_{i=1}^{\lfloor nc \rfloor} \frac{\phi_c(i)}{np_i}$ in Lemma 11. Two techniques will be used: (1) as demonstrated in Lemma 9, we are able to identify a lower bound of the periods p_i; (2) as illustrated in Lemma 10, a swapping technique is invented to enables us to figure out the worst-case $\overrightarrow{\mathbf{p}}$.

Before continuing, some notations have to be introduced. Arbitrarily fix $n \in \mathbb{Z}_+$ and $\overrightarrow{\mathbf{p}} \in \mathbb{Z}_+^n$, which will be used throughout this section. For any $1 \leq i \leq n$, define

$$S_i = \{j : 1 \leq j \leq n, j + p_j \leq i + p_i\},$$

and let α_i be the i-th smallest among the multiset $\{|S_j| : 1 \leq j \leq n\}$, where $|x|$ is the size of set x. Intuitively, $j \in S_i$ if and only if the deadline of the second job of τ_j is not later than that of τ_i. These concepts are exemplified as follows.

Example 1. *Consider $n = 5$ and $\overrightarrow{\mathbf{p}} = (60, 21, 20, 71, 70)$. Table I indicates the corresponding S_i and α_i values.*

TABLE I
AN EXAMPLE OF S_i AND α_i, WHERE $n = 5$.

i	1	2	3	4	5
p_i	60	21	20	71	70
$i + p_i$	61	23	23	75	75
S_i	{1,2,3}	{2,3}	{2,3}	{1,2,3,4,5}	{1,2,3,4,5}
$\lvert S_i \rvert$	3	2	2	5	5
α_i	2	2	3	5	5

We have an easy observation of α_i.

Lemma 8: For any $n \in \mathbb{Z}_+$ and $1 \leq i \leq n$, $\alpha_i \geq i$.

The proof is omitted. Hereunder, all the omitted proofs will appear in the full version of this manuscript.

We can lower-bound p_i in terms of S_i.

Lemma 9: For any $1 \leq i \leq n$, $p_i \geq n - i + |S_i|$.

The proof is omitted.

The following is a technical lemma which will enable us to figure out the worst-case $\overrightarrow{\mathbf{p}}$.

Lemma 10: For any $m \in \mathbb{Z}_+$ and $\overrightarrow{\mathbf{x}}, \overrightarrow{\mathbf{y}} \in \mathbb{R}_+^m$, assume that $\overrightarrow{\mathbf{x}}$ is increasing and the permutation $\vec{\pi}$ on $\{1, 2, ..., m\}$ sorts $\overrightarrow{\mathbf{y}}$ increasingly. Namely, $x_i \leq x_{i+1}$ and $y_{\pi_i} \leq y_{\pi_{i+1}}$ for any $1 \leq i < m$. Then

$$\sum_{i=1}^{m} \frac{1}{x_i + y_{\pi_i}} \geq \sum_{i=1}^{m} \frac{1}{x_i + y_i}, \tag{25}$$

$$\sum_{i=1}^{m} \frac{x_i^2}{x_i + y_i} \leq \sum_{i=1}^{m} \frac{x_i^2}{x_i + y_{\pi_{m-i+1}}}. \tag{26}$$

The proof is omitted.

We are ready to derive a good upper bound of $\sum_{i=1}^{\lfloor nc \rfloor} \frac{\phi_c(i)}{np_i}$.

Lemma 11: $\sum_{i=1}^{\lfloor cn \rfloor} \frac{\phi_c(i)}{np_i} \leq \frac{c(1-(1-c)^3)}{3}$.

Proof: For any $1 \leq i \leq \lfloor cn \rfloor$, let β_i be the i-th smallest among the multiset $\{|S_j| : 1 \leq j \leq \lfloor cn \rfloor\}$; obviously, $\beta_i \geq \alpha_i$.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

We have

$$
\begin{aligned}
& \sum_{i=1}^{\lfloor cn \rfloor} \frac{(n-i)^2}{n^2 p_i} \\
\leq \quad & \sum_{i=1}^{\lfloor cn \rfloor} \frac{(n-i)^2}{n^2(n-i+|S_i|)} \text{ (by Lemma 9)} \\
\leq \quad & \sum_{i=1}^{\lfloor cn \rfloor} \frac{(n-i)^2}{n^2(n-i+\beta_i)} \text{ (by Inequality (26))} \\
\leq \quad & \sum_{i=1}^{\lfloor cn \rfloor} \frac{(n-i)^2}{n^2(n-i+\alpha_i)} \text{ (since } \beta_i \geq \alpha_i) \\
\leq \quad & \sum_{i=1}^{\lfloor cn \rfloor} \frac{(n-i)^2}{n^3} \text{ (by Lemma 8)} \\
\leq \quad & \sum_{i=1}^{\lfloor cn \rfloor} \int_{\frac{i-1}{n}}^{\frac{i}{n}} (1-x)^2 dx \\
& \text{(since } (1-x)^2 \text{ is decreasing when } x \leq 1) \\
\leq \quad & \int_0^c (1-x)^2 dx = \frac{1-(1-c)^3}{3}.
\end{aligned}
$$

The lemma follows immediately. ■

B. Upper bounding $\sum_{i=1}^{\lfloor nc \rfloor} \frac{\psi_c(i)}{np_i}$

One more technical lemma is needed. It is a variant of Lemma 5, estimating the sum of $\frac{1}{p_i}$ over a general interval of i.

Lemma 12: For any real numbers $0 < a < b < 1$, we have

$$
\sum_{i=\lceil an \rceil}^{\lfloor bn \rfloor} \frac{1}{p_i} \leq \frac{1}{2} \ln \frac{1+b-2a}{1-b} + o(1).
$$

The proof is omitted.

Now we can derive a good upper bound of $\sum_{i=1}^{\lfloor nc \rfloor} \frac{\psi_c(i)}{np_i}$. The basic idea is to approximate it with a new summation over the integer points in the left/right part of Figure 4. The new summation is then estimated via calculus.

Lemma 13:

$$
\sum_{i=1}^{\lfloor nc \rfloor} \frac{\psi_c(i)}{np_i} \leq \frac{1}{2} \int_0^{\frac{(2c-1)^2}{4c}} g_c(x) dx + o(1), \text{ where}
$$

$$
g_c(x) \triangleq \ln \frac{1+3\sqrt{(2c-1)^2-4cx}}{1-\sqrt{(2c-1)^2-4cx}}.
$$

Proof: Let $C_1 = 2 - \frac{1}{c} \leq c < 1$ (since $1/2 < c < 1$). Note that $C_1 n$ is the right zero point of the quadratic function

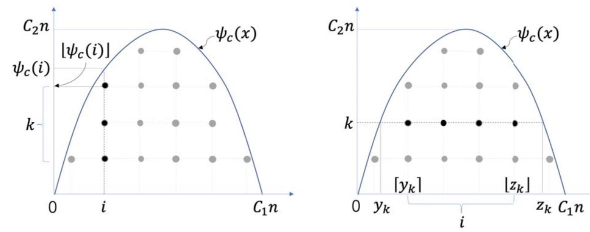

Fig. 4. The summation is over the integer points under the curve and above the horizontal axis. The left part illustrates $\sum_i \sum_k$, while the right for $\sum_k \sum_i$. The ranges of i and k are adjusted when the summation order changes.

$\psi_c(x)$. Hence, $\psi_c(x) < 0$ for $x > C_1 n$. Then,

$$
\begin{aligned}
& \sum_{i=1}^{\lfloor nc \rfloor} \frac{\psi_c(i)}{np_i} \\
\leq \quad & \sum_{i=1}^{\lfloor C_1 n \rfloor} \frac{\psi_c(i)}{np_i} \\
& \text{(since } C_1 \leq c \text{ and } \psi_c(x) < 0 \text{ for } x > C_1 n) \\
\leq \quad & \sum_{i=1}^{\lfloor C_1 n \rfloor} \frac{\lfloor \psi_c(i) \rfloor}{np_i} + \sum_{i=1}^{\lfloor C_1 n \rfloor} \frac{1}{np_i} \\
\leq \quad & \sum_{i=1}^{\lfloor C_1 n \rfloor} \frac{\lfloor \psi_c(i) \rfloor}{np_i} + \frac{1}{n} \sum_{i=1}^{n} \frac{1}{p_i} \text{ (since } C_1 < 1) \\
= \quad & \sum_{i=1}^{\lfloor C_1 n \rfloor} \frac{\lfloor \psi_c(i) \rfloor}{np_i} + o(1) \text{ (by Lemma 5)} \\
= \quad & \sum_{i=1}^{\lfloor C_1 n \rfloor} \sum_{k=1}^{\lfloor \psi_c(i) \rfloor} \frac{1}{np_i} + o(1).
\end{aligned}
$$

Let $C_2 = \frac{(2c-1)^2}{4c}$. Then $\max_{x \in \mathbb{R}_+} \psi_c(x) = C_2 n$. For any integer $k \leq \lfloor C_2 n \rfloor$, let $y_k \leq z_k$ be the two real roots of the equation $\psi_c(x) = k$, i.e.,

$$
y_k = \frac{(2c-1) - \sqrt{(2c-1)^2 - 4kc/n}}{2c/n}, \text{and}
$$

$$
z_k = \frac{(2c-1) + \sqrt{(2c-1)^2 - 4kc/n}}{2c/n}.
$$

The lemma holds since

$$
\begin{aligned}
& \sum_{i=1}^{\lfloor C_1 n \rfloor} \sum_{k=1}^{\lfloor \psi_c(i) \rfloor} \frac{1}{np_i} \\
= \quad & \sum_{k=1}^{\lfloor C_2 n \rfloor} \sum_{i=\lceil y_k \rceil}^{\lfloor z_k \rfloor} \frac{1}{np_i} \\
& \text{(by interchanging the order of sum—see Fig. 4)}
\end{aligned}
$$

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

$$
\begin{aligned}
&\le \sum_{k=1}^{\lfloor C_2 n\rfloor} \frac{1}{2n} \ln \frac{1+\frac{z_k}{n}-2\frac{y_k}{n}}{1-\frac{z_k}{n}} + o(1) \text{ (by Lemma 12)}\\
&= \sum_{k=1}^{\lfloor C_2 n\rfloor} \frac{1}{2n} \ln \frac{1+3\sqrt{(2c-1)^2-4c\frac{k}{n}}}{1-\sqrt{(2c-1)^2-4c\frac{k}{n}}} + o(1)\\
&\quad \text{(by definition of } y_k \text{ and } z_k)\\
&= \frac{1}{2}\sum_{k=1}^{\lfloor C_2 n\rfloor} \frac{1}{n} g_c\left(\frac{k}{n}\right) + o(1)\\
&\le \frac{1}{2}\sum_{k=1}^{\lfloor C_2 n\rfloor} \int_{\frac{k-1}{n}}^{\frac{k}{n}} g_c(x)dx + o(1)\\
&\quad \text{(since } g_c(x) \text{ is decreasing when } x \le C_2 \text{)}\\
&= \frac{1}{2}\int_0^{\frac{\lfloor C_2 n\rfloor}{n}} g_c(x)dx + o(1)\\
&\le \frac{1}{2}\int_0^{C_2} g_c(x)dx + o(1).\\
&\quad \text{(since } g_c(x) \ge 0 \text{ when } x \le C_2 \text{)}
\end{aligned}
$$

■

C. Upper bounding ξ

Another technical lemma is needed. It claims that the sequence $\{\xi_n\}_{n=1}^{+\infty}$ converges to ξ. It will enable us to neglect the $o(1)$ term in Lemma 13 when estimating ξ.

Lemma 14: $\xi = \lim_{n\to+\infty} \xi_n$.

The proof is omitted.

Theorem 2: $\xi < 0.5380$.

Proof: Choose $c = 0.8$.

$$
\begin{aligned}
&\sum_{i=1}^{n} \frac{n-i}{np_i}\\
\le\ & 1-c+\sum_{i=1}^{\lfloor nc\rfloor} \frac{\phi_c(i)}{np_i} + \sum_{i=1}^{\lfloor nc\rfloor} \frac{\psi_c(i)}{np_i} \text{ (by Lemma 7)}\\
\le\ & 1-c+\frac{c}{3}\left(1-(1-c)^3\right) + \frac{1}{2}\int_0^{\frac{(2c-1)^2}{4c}} g_c(x)dx + o(1)\\
& \text{(by Lemmas 11 and 13)}\\
\le\ & 0.2+0.2645+0.07341+o(1)\\
& \text{(computed by Mathematica)}\\
=\ & 0.53791+o(1).
\end{aligned}
$$

Then, by definition of ξ_n, we have

$$\xi_n \le 0.53791 + o(1).$$

By Lemma 14, it holds that

$$\xi = \lim_{n\to+\infty} \xi_n \le 0.53791 < 0.5380.$$

■

Remark 2: Combining Theorem 2 and Corollary 1, we get $\rho < 1.5380$. This upper bound of ρ holds even for arbitrary-deadline tasks, since so do Lemmas 2,3,7,8 in [17]. Then by Lemma 1, we get asymptotic upper bound 2.5380 of the speedup factor of P^{DM}-EDF on constrained-deadline tasks.

V. Conclusion and Future Work

We improved both upper and lower bounds of ρ, the speedup factor of uniprocessor schedulability testing by an approximate demand bound function. The new bounds of ρ hold for both constrained-deadline and constrained-deadline tasks. Accordingly, the potential range of the speedup factor of (deadline monotonically) partitioned-EDF (or, P^{DM}-EDF) gets tighter, due to its connection with ρ. However, the new range of the speedup factor are valid only for constrained-deadline tasks, since the connection with ρ was established only in this case.

We achieve the improvements via three techniques. First, we construct an auxiliary function which is bigger than the approximate demand bound function but has the same supremum. This makes it possible to beat the lower bound of 1.5 using a relatively small task set, bypassing the difficulty in checking schedulability of large task sets. Second, we show that the approximate demand bound function converges to ρ in a nearly increasing manner as the task set size goes to infinity. This enables us to estimate ρ asymptotically, precluding the effects of vanishing terms. Finally, a better lower bound of the periods is figured out for schedulable sets of tasks, which plays a critical role in deriving the upper bound of ρ.

The novel auxiliary function plays a critical role in analyzing ρ. We would like to point out that it seems to be a highly transferable technique for shaping and bounding demands for various types of real-time workloads, such as resource augmentation bounds in DAG task scheduling.

It is unlikely that our new lower bound or upper bound of ρ is tight. The small (0.0026) improvement of the lower bound is significant because it disproves the conjecture that $\rho = 1.5$. It is worthwhile to further improve it, or even just to make a new conjecture.

Acknowledgement

Xingwu Liu would like to thank Prof. Zhiwei Xu and Prof. Yungang Bao at the Institute of Computing Technology, Chinese Academy of Sciences. Their valuable support makes this work possible. The authors are also grateful for the anonymous reviewers on their constructive comments and suggestions which inspired us to greatly improve the readability of this manuscript.

This work is supported in part by the National Key Research and Development Program of China (Grant No. 2016YFB1000201), the National Natural Science Foundation of China (Grant No. 11971091, 62072433, 62090020), the Fundamental Research Funds for the Central Universities (Grant No. DUT21RC(3)102, DUT21LAB302), the Youth Innovation Promotion Association of Chinese Academy of Sciences (Grant No. 2013073), and the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDC05030200).

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.

References

[1] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[2] M. Dertouzos, "Control robotics: The procedural control of physical processors," in *the IFIP Congress*, 1974, pp. 807 – 813.

[3] F. Eisenbrand and T. Rothvoß, "Edf-schedulability of synchronous periodic task systems is conp-hard," in *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2010, pp. 1029–1034.

[4] P. Ekberg and W. Yi, "Uniprocessor feasibility of sporadic tasks remains conp-complete under bounded utilization," in *2015 IEEE Real-Time Systems Symposium*, 2015, pp. 87–95.

[5] S. K. Baruah, A. K. Mok, and L. E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," in *Real-Time Systems Symposium, 1990. Proceedings., 11th*. IEEE, 1990, pp. 182–190.

[6] E. Bini, "Uniprocessor edf feasibility is an integer problem," in *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.

[7] R. R. Devillers and J. Goossens, "Liu and layland's schedulability test revisited," *Inf. Process. Lett.*, vol. 73, no. 5-6, pp. 157–161, 2000.

[8] K. Albers and F. Slomka, "An event stream driven approximation for the analysis of real-time systems," in *Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on*. IEEE, 2004, pp. 187–195.

[9] S. K. Baruah and J. Goossens, "The EDF scheduling of sporadic task systems on uniform multiprocessors," in *Proceedings of the 29th IEEE Real-Time Systems Symposium, RTSS 2008, Barcelona, Spain, 30 November - 3 December 2008*, 2008, pp. 367–374.

[10] R. I. Davis, A. Thekkilakattil, O. Gettings, R. Dobrin, S. Punnekkat, and J. Chen, "Exact speedup factors and sub-optimality for non-preemptive scheduling," *Real-Time Systems*, vol. 54, no. 1, pp. 208–246, 2018.

[11] B. Kalyanasundaram and K. Pruhs, "Speed is as powerful as clairvoyance," *J. ACM*, vol. 47, no. 4, pp. 617–643, 2000.

[12] J.-J. Chen, G. von der Brüggen, W.-H. Huang, and R. I. Davis, "On the Pitfalls of Resource Augmentation Factors and Utilization Bounds in Real-Time Scheduling," in *29th Euromicro Conference on Real-Time Systems (ECRTS 2017)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), 2017, pp. 9:1–9:25.

[13] Z. Guo, "Regarding the optimality of speedup bounds of mixed-criticality schedulability tests," *Mixed Criticality on Multicore/Manycore Platforms (Dagstuhl Seminar Reports)*, vol. 17131, 2017.

[14] K. Agrawal and S. Baruah, "Intractability issues in mixed-criticality scheduling," in *the 30th Euromicro Conference on Real-Time Systems (ECRTS'18)*, vol. 106, 2018, pp. 11:1–11:21.

[15] S. Baruah, "Mixed-criticality scheduling theory: Scope, promise, and limitations," *IEEE Design Test*, vol. 35, no. 2, pp. 31–37, 2018.

[16] S. Baruah and N. Fisher, "The partitioned multiprocessor scheduling of sporadic task systems," in *Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International*. IEEE, 2005, pp. 321–329.

[17] X. Liu, X. Han, L. Zhao, and Z. Guo, "Narrowing the speedup factor gap of partitioned-EDF," *Information and Computation*, p. 104743, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0890540121000584

[18] J.-J. Chen and S. Chakraborty, "Resource augmentation bounds for approximate demand bound functions," in *Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd*. IEEE, 2011, pp. 272–281.

[19] X. Han, L. Zhao, Z. Guo, and X. Liu, "An improved speedup factor for sporadic tasks with constrained deadlines under dynamic priority scheduling," in *Real-Time Systems Symposium (RTSS), 2018 IEEE 39nd*. IEEE, 2018, pp. 447–455.

[20] X. Han and Z. Guo, "Resource augmentation bounds of EDF and partitioned-EDF for sporadic tasks with constrained deadlines," in *9th Real-Time Scheduling Open Problems Seminar (RTSOPS)*, 2018, pp. 9–10.

[21] P. Ekberg, "Dual Priority Scheduling is Not Optimal," in *31st Euromicro Conference on Real-Time Systems (ECRTS 2019)*, vol. 133. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 14:1–14:9.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on April 28,2023 at 17:30:23 UTC from IEEE Xplore. Restrictions apply.