An Improved Speedup Factor for Sporadic Tasks with Constrained Deadlines under Dynamic Priority Scheduling

Xin Han, Liang Zhao
Software School,
Dalian University of Technology, China

Zhishan Guo
Department of ECE,
University of Central Florida, US

Xingwu Liu*
University of Chinese Academy of Sciences,
Chinese Academy of Sciences, China

***Abstract*—Schedulability is a fundamental problem in real-time scheduling, but it has to be approximated due to the intrinsic computational hardness. As the most popular algorithm for deciding schedulability on multiprocess platforms, the speedup factor of partitioned-EDF is challenging to analyze and is far from being determined. Partitioned-EDF was first proposed in 2005 by Barush and Fisher [1], and was shown to have a speedup factor at most $3 - 1/m$, meaning that if the input of sporadic tasks is feasible on m processors with speed one, partitioned-EDF will always succeed on m processors with speed $3-1/m$. In 2011, this upper bound was improved to $2.6322 - 1/m$ by Chen and Chakraborty [2], and no more improvements have appeared ever since then. In this paper, we develop a novel method to discretize and regularize sporadic tasks, which enables us to improve, in the case of constrained deadlines, the speedup factor of partitioned-EDF to $2.5556 - 1/m$, very close to the asymptotic lower bound 2.5 in [2].**

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

I. Introduction

Scheduling is a hot topic in the real-time systems community. Basically, given a finite set of tasks, each sequentially releasing infinitely many jobs, the mission of real-time scheduling is to allocate computing resources so that all the jobs are done in a timely manner. The fundamental question of schedulability naturally arises: Is it possible at all to successfully schedule these tasks, such that all of them receive enough execution before their deadlines?

Unfortunately, answering this question is often not 'easy'; e.g., the schedulability of a set of constrained-deadline sporadic tasks, which is the focus of this paper, is co-NP-hard even on a uniprocessor platform [3]. For multiprocessor case, it remains NP-hard for partitioned paradigm, even if the relative deadline of each task equals to its period [4]. Here *partitioned paradigm* means that once a task is assigned to a processor, all the jobs released by the task will be scheduled on the dedicated processor. These hardness results imply that it is impossible to exactly decide schedulability in polynomial time, unless $P = NP$.

Due to the hardness, real-time schedulability problems are usually solved approximately by pessimistic algorithms which always answer 'No' unless some sufficient-only conditions for schedulability are met. To evaluate the performance of such an approximate algorithm (say, $\mathcal{A}$), the concept of *speedup factor*, also known as resource augmentation bound, has been proposed. Specifically, whenever a set of tasks is schedulable on a platform with speed one, algorithm $\mathcal{A}$ will return 'Yes' on the same platform with speed $r \geq 1$. The minimum such r is referred to as the speedup factor of $\mathcal{A}$. Despite of some recent discussion on potential pitfalls [5] [6] [7], speedup factor has been a major metric and standard theoretical tool for assessing scheduling algorithms since the seminal work in 2000 [8].

Recent years has witnessed impressive progress on finding scheduling algorithms with low speedup factors. For the preemptive case (i.e. running jobs might be interrupted by emergent ones), Global-EDF has a speedup factor $2-1/m$ [9] for scheduling tasks on m identical processors, and there is a polynomial-time algorithm for uniprocessors whose speedup factor is $1 + \epsilon$ [10], where $\epsilon > 0$ is sufficiently small for uniform processors, refer to [11]. For the non-preemptive case, there are also a variety of results, refer to [12], [13]. Except the speedup factor, there are many papers concerning about the utilization upper bound, refer to [14]–[16].

Although the speedup factor on uniprocessors is tight, the multiprocessor case remains open. Due to its simplicity, partitioned scheduling is of particular interest and has been attracting more and more attention from researchers and practitioners. Partitioned-EDF is the most popular scheduling algorithm of partitioned style, while the above-mentioned Global-EDF is not of partitioned paradigm. Breakthrough was made in the year of 2005, when Baruah and Fisher [1] established a $3-1/m$ $(4-2/m, respectively)$ upper bound for the speedup factor of partitioned-EDF on constrained-deadline (arbitrary-deadline, respectively) task sets, where m is the number of identical processors. A set of tasks is said to be constrained-deadline, if the relative deadline of each task is at most its period, otherwise is arbitrary-deadline. Then in 2011, Chen and Chakraborty [2] further improved the speedup factor to $2.6322 - 1/m$ ($3 - 1/m$, respectively) for the constrained-deadline case (arbitrary-deadline case, respectively). Also in the same paper, a lower bound 2.5 of the speedup factor

*Corresponding author, liuxingwu@ict.a.cn.

was established for the constrained-deadline case (via counter example). Throughout the last seven years, the bounds in [2] were never improved.

It is worth noting that deriving the upper bound of the speedup factor of partitioned-EDF relies heavily on a quantity about scheduling on uniprocessors, denoted by ρ which is formally defined in (1) of Section II. Roughly speaking, ρ measures how far the approximate demand bound function (defined in Section II) deviates from the actual deadline time-points. Baruah and Fisher [1] bridged ρ and the speedup factor of partitioned-EDF by showing that in case of constrained deadlines, the speedup factor is at most $1 + \rho - 1/m$. As a result, upper-bounding the speedup factor is reduced to upper-bounding ρ, and it is in this manner that both [1] and [2] obtained their estimations of the speedup factor. Hence, the quantity ρ itself deserves a deep investigation. Actually, Baruah and Fisher [1] upper-bounded it by 2, Chen and Chakraborty [2] narrowed its range into $[1.5, 1.6322]$.

On this ground, this paper will explore a better upper bound of ρ, and on this basis, provide a better estimation of the speedup factor of partitioned-EDF for sets of constrained-deadline sporadic tasks. The contributions are summarized into the following three aspects.

1) We improve the best existing upper bound of ρ for constrained-deadline tasks from 1.6322 to 1.5556 (Theorem 1), which is very close to the lower bound 1.5. The speedup factor of partitioned-EDF for the constrained-deadline case accordingly decreases from $2.632 - 1/m$ to $2.5556 - 1/m$ (Theorem 2).
2) We identified a way to discretize and regularize the constrained-deadline tasks with out losing tightness. As a result, the execution times of the tasks of interest becomes essentially all 1 and the deadlines are $1, 2, \cdots, n$ respectively, where n is number of tasks to be scheduled (Lemmas 3, 5, 6). The only parameter that varies is the period. The transformation is tight/lossless in the sense that the quantity ρ does not change although the parameters are extremely simplified.
3) We invent a method to further transform the tasks so that the period of each task ranges over integers between 1 to $2n$ (Lemma 7). Although tightness is no longer preserved during this second transformation step, the loss is negligible since we prove that ρ increases by at most 0.0556. We expect these transformation techniques may be further applied to real-time scheduling analysis or other problems.

The rest of the paper is organized as follows: Section II presents the model and preliminaries; Section III focuses on uniprocessor case and derives a new upper bound $(14/9)$ of ρ for feasible sporadic tasks; Section IV provides a new upper bound $(23/9 - 1/m)$ of the speedup factor for partitioned-EDF. Finally, Section V concludes the paper and mentions some potential future directions.

II. System Model and Preliminaries

We consider a finite set τ of sporadic tasks. Each task τ_i can be represented by a triple $\tau_i = (e_i, d_i, p_i)$, where e_i is the worst-case execution time, d_i is its relative deadline, and p_i is the minimum inter-arrival separation length (also known as period), respectively. The task τ_i is said to be constrained-deadline if $d_i \le p_i$.

This paper focuses on constrained-deadline tasks. Hereunder, every task has a constrained deadline unless otherwise mentioned.

Given a task τ_i, we can calculate its demand bound function $dbf(\tau_i, t)$ [17] and its approximate demand bound function $dbf^*(\tau_i, t)$ [10] in the following manner:

$$dbf(\tau_i, t) = \begin{cases} 0 & \text{if } t < d_i \\ \left(\left\lfloor \frac{t-d_i}{p_i} \right\rfloor + 1 \right) \cdot e_i, & \text{otherwise} \end{cases}$$

and

$$dbf^*(\tau_i, t) = \begin{cases} 0 & \text{if } t < d_i \\ \left(\frac{t-d_i}{p_i} + 1 \right) \cdot e_i, & \text{otherwise.} \end{cases}$$

Similarly for any set τ of tasks, we define

$$dbf(\tau, t) = \sum_{\tau_i \in \tau} dbf(\tau_i, t), \quad dbf^*(\tau, t) = \sum_{\tau_i \in \tau} dbf^*(\tau_i, t).$$

Lemma 1: [17] A set of tasks τ is feasible on uniprocessor if and only if $dbf(\tau, t) \le t$, for $\forall t \ge 0$.

To analyze the speedup factor of partitioned-EDF on multiprocessor platforms, the following quantity plays a critical role:

$$\rho = \sup_{\tau} \frac{dbf^*(\tau, d)}{d}, \tag{1}$$

where τ ranges over sporadic task sets that are feasible on uniprocessor platforms, and d is the largest relative deadline in τ. Here *feasible* means that there exists a *correct* schedule to the set of tasks, i.e., each task could receive up to e_i time units of execution after its release and before its deadline.

We will see that actually, ρ is the optimum value of the following math programming MP_0:

$$\sup \quad \frac{dbf^*(\tau, d_n)}{d_n}, \qquad (MP_0) \tag{2}$$

$$\textit{subject to} \quad dbf(\tau, t) \le t, \quad \forall t > 0 \tag{3}$$

$$d_i + p_i > d_n, \quad 1 \le i \le n-1, \tag{4}$$

$$d_1 \le d_2 \le \cdots \le d_n, \tag{5}$$

$$n \in \mathbb{Z}^+, e_i, d_i, p_i \in \mathbb{R}^+, \quad 1 \le i \le n. \tag{6}$$

where $\mathbb{Z}^+$ is the set of positive integers while $\mathbb{R}^+$ stands for the set of positive real numbers.

Lemma 2: ρ is the optimum value of MP_0.

Proof: Let $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \le i \le n\}$ be an arbitrary set of sporadic tasks that is feasible on a uniprocessor with speed 1. Assume that $d_1 \le d_2 \le \cdots \le d_n$. Apply the transformation proposed in [2]:

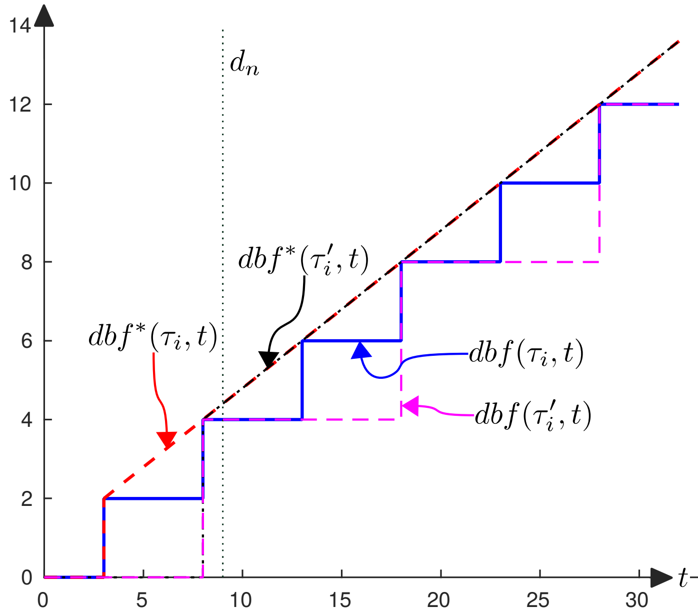

Fig. 1. An example for task transformation and dbf modifications, with task parameters of $e_i = 2, d_i = 3, p_i = 5$, and $d_n = 9$.

$$e_i^{'} = \left(\left\lfloor \frac{d_n - d_i}{p_i} \right\rfloor + 1\right) \cdot e_i, \tag{7}$$

$$p_i^{'} = \left(\left\lfloor \frac{d_n - d_i}{p_i} \right\rfloor + 1\right) \cdot p_i, \tag{8}$$

$$d_i^{'} = \left(\left\lfloor \frac{d_n - d_i}{p_i} \right\rfloor\right) \cdot p_i + d_i. \tag{9}$$

Let $\tau^{'} = \{\tau_1^{'}, \tau_2^{'}, \cdots, \tau_n^{'}\}$ with $\tau_i^{'} = (e_i^{'}, d_i^{'}, p_i^{'})$ for any $1 \leq i \leq n$.

Please refer to Figure 1 for an illustration of the above mentioned transformation.

For any $1 \leq i \leq n$, $p_i^{'} - d_i^{'} = p_i - d_i$. Hence $\tau^{'}$ is constrained-deadline since so is τ.

In [2], it was proven that the following results hold simultaneously:

i) $dbf^*(\tau, t) = dbf^*(\tau^{'}, t)$ for any $t \geq d_n$;
ii) $dbf(\tau, t) \geq dbf(\tau^{'}, t)$ for $t > 0$;
iii) $d_n^{'} < d_i^{'} + p_i^{'}$ for $1 \leq i \leq n$;
iv) $d_n^{'} = d_n$.

This immediately leads to our lemma. ■

III. Improved Bound for Uniprocessor Case

In order to estimate the speedup factor for multiprocessor partitioned scheduling, we first focus on the uniprocessor case. The main result of this section is Theorem 1, which establishes 14/9 as an upper bound of ρ for sporadic tasks.

The basic idea of our proof is to discretize the tasks into regular form, thus reducing the problem into an optimization one on bounded integers. Roughly speaking, Lemma 3 makes sure that ρ does not change if the parameters of the tasks are restricted to be rational numbers, Lemma 5 claims that further requiring $e_i = d_i - d_{i-1}$ for all i keeps ρ unchanged, the trend continues by Lemma 6 even if all the tasks are required to have the same worst-case execution time, and finally, Lemma 7 enables us to only consider tasks with bounded discrete periods. These transformations reduce estimating ρ to a simpler optimization problem which is solved approximately in Lemma 9. These results immediately lead to Theorem 1. The overall proof flow is illustrated in Figure 2.

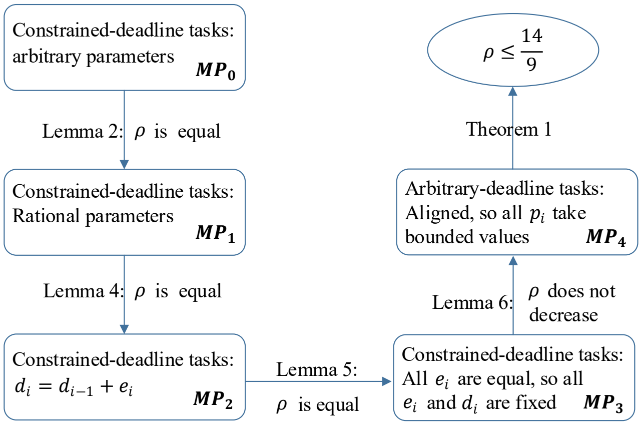
Constrained-deadline tasks: arbitrary parameters MP_0
Lemma 2: ρ is equal
Constrained-deadline tasks: Rational parameters MP_1
Lemma 4: ρ is equal
Constrained-deadline tasks: $d_i = d_{i-1} + e_i$ MP_2
Lemma 5: ρ is equal
Constrained-deadline tasks: All e_i are equal, so all e_i and d_i are fixed MP_3
Lemma 6: ρ does not decrease
Arbitrary-deadline tasks: Aligned, so all p_i take bounded values MP_4
Theorem 1
$\rho \leq \frac{14}{9}$

Fig. 2. The flow of the proofs of this section. The constraints are added incrementally, so each box only presents the new constraint. The overall constraints in each box is formulated into a math programming whose name MP_* is labeled at the lower-right corner of the box.

Specifically, we first observe that the optimum value of MP_0 remains unchanged even if the domain $\mathbb{R}^+$ is replaced by $\mathbb{Q}^+$, the set of positive rational numbers.

$$\sup \quad \frac{dbf^*(\tau, d_n)}{d_n}, \qquad (MP_1) \tag{10}$$

$$\textit{subject to} \quad dbf(\tau, t) \leq t, \quad \forall t > 0 \tag{11}$$

$$d_i + p_i > d_n, \quad 1 \leq i \leq n-1, \tag{12}$$

$$d_1 \leq d_2 \leq \cdots \leq d_n, \tag{13}$$

$$n \in \mathbb{Z}^+, e_i, d_i, p_i \in \mathbb{Q}^+, \quad 1 \leq i \leq n. \tag{14}$$

Lemma 3: MP_0 and MP_1 has the same optimum value.

Proof: The lemma immediately holds if both of the following claims are true:

1) The objective functions of MP_0 and MP_1 are the same and continuous.
2) The domain of MP_1 is a dense subset of that of MP_0. The term 'dense' means that for any $\epsilon > 0$ and any feasible solution $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \leq i \leq n\}$ to MP_0, there is a feasible solution $\tau' = \{\tau_i' = (e_i', d_i', p_i') : 1 \leq i \leq n\}$ to MP_1 such that for any $1 \leq i \leq n$,

$$|e_i' - e_i| < \epsilon, |d_i' - d_i| < \epsilon, |p_i' - p_i| < \epsilon. \tag{15}$$

It suffices to prove Claim 2 since the other is obvious.

Let $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \le i \le n\}$ be an arbitrary set of tasks that is a feasible solution to MP_0, and ϵ be an arbitrary positive real number. Without loss of generality, assume that $\epsilon < \min_{1 \le i \le n} e_i$. For any $1 \le i \le n$, arbitrarily choose

$$\begin{aligned} p'_i &\in (p_i + \frac{\epsilon}{2}, p_i + \epsilon) \cap \mathbb{Q}^+, \\ d'_i &\in (d_i + \frac{(i-1)\epsilon}{2n}, d_i + \frac{i\epsilon}{2n}) \cap \mathbb{Q}^+, \\ e'_i &\in (e_i - \epsilon, e_i) \cap \mathbb{Q}^+. \end{aligned}$$

Let τ' denote the set of tasks $\{\tau'_i = (e'_i, d'_i, p'_i) : 1 \le i \le n\}$. Obviously, τ' meets Conditions (14) and (15).

To proceed, arbitrarily fix an integer $1 \le i \le n$.

Note that $p'_i - d'_i > p_i + \frac{\epsilon}{2} - (d_i + \frac{i\epsilon}{2n}) \ge p_i - d_i$. This, together with the fact that τ_i is constrained-deadline, means τ'_i is also constrained-deadline.

Observe that

$$d'_i > d_i + \frac{(i-1)\epsilon}{2n} \ge d_{i-1} + \frac{(i-1)\epsilon}{2n} > d'_{i-1}.$$

Hence, τ' satisfies Condition (13) of MP_1.

Because

$$\begin{aligned} d'_i + p'_i &> d_i + \frac{(i-1)\epsilon}{2n} + p_i + \frac{\epsilon}{2} \\ &\ge d_i + p_i + \frac{\epsilon}{2} \\ &> d_n + \frac{\epsilon}{2} \quad \text{(since } \tau \text{ satisfies (4))} \\ &> d'_n, \end{aligned}$$

the task set τ' satisfies Condition (12).

As to Condition (11), arbitrarily fix $t > 0$.

When $t < d'_i$, $dbf(\tau'_i, t) = 0 \le dbf(\tau_i, t)$.

When $t \ge d'_i$, because $p'_i > p_i, d'_i > d_i, e'_i < e_i$, we have

$$\begin{aligned} dbf(\tau'_i, t) &= \left(\left\lfloor \frac{t - d'_i}{p'_i} \right\rfloor + 1 \right) \cdot e'_i \\ &\le \left(\left\lfloor \frac{t - d_i}{p_i} \right\rfloor + 1 \right) \cdot e_i = dbf(\tau_i, t). \end{aligned}$$

As a result, we always have $dbf(\tau', t) \le dbf(\tau, t)$. Then τ' satisfies Condition (11) since τ satisfies (3).

Altogether, τ' is a feasible solution to MP_1. ■

Now we present a technical lemma that will be frequently used.

Lemma 4: Suppose $d, p, d', p' \in \mathbb{R}^+$ are such that $d + p = d' + p'$ and $d > d'$. For any real number t,

$$\frac{t - d'}{p'} > \frac{t - d}{p}$$

if and only if $t < d + p$.

Proof: Let $\delta = d - d' = p' - p$. Then

$$\begin{aligned} & \frac{t - d'}{p'} > \frac{t - d}{p} \\ \Leftrightarrow \quad & p \cdot (t - d') > p' \cdot (t - d) \\ \Leftrightarrow \quad & p \cdot (t - d + \delta) > (p + \delta) \cdot (t - d) \\ \Leftrightarrow \quad & p \cdot \delta > \delta \cdot (t - d) \\ \Leftrightarrow \quad & p > t - d. \end{aligned}$$

■

Hereunder, let $d_0 = 0$. Then it is time to show that the optimum value of MP_1 remains unchanged even if we further require $e_i = d_i - d_{i-1}$ for all $i \ge 1$. We define a new math programming

$$\begin{aligned} \sup \quad & \frac{dbf^*(\tau, d_n)}{d_n}, & (MP_2) \quad (16) \\ \textit{subject to} \quad & dbf(\tau, t) \le t, \quad \forall t > 0 & (17) \\ & d_i + p_i > d_n, \quad 1 \le i \le n - 1, & (18) \\ & d_i = e_i + d_{i-1}, \quad 1 \le i \le n, & (19) \\ & n \in \mathbb{Z}^+, e_i, d_i, p_i \in \mathbb{Q}^+, \quad 1 \le i \le n. & (20) \end{aligned}$$

Lemma 5: MP_1 and MP_2 have the same optimum value.

Proof: For any feasible solution $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \le i \le n\}$ to MP_1, define $M(\tau) \triangleq |\{i : 1 \le i \le n, d_i \ne e_i + d_{i-1}\}|$. Obviously, τ is a feasible solution to MP_2 if and only if $M(\tau) = 0$.

Consider the following proposition: for any feasible solution τ to MP_1 with $M(\tau) > 0$, there is a feasible solution τ' to MP_1 such that $M(\tau') < M(\tau)$ and the objective value of τ' is at least that of τ. If it is true, one can easily prove the lemma by iteratively applying the proposition. Hence, the rest of the proof is devoted to proving this proposition.

Arbitrarily fix a feasible solution $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \le i \le n\}$ to MP_1. Suppose $M(\tau) > 0$. Assume k is the smallest index such that $e_k \ne d_k - d_{k-1}$, meaning that $e_i = d_i - d_{i-1}$ for all $i < k$. Then we have

$$\sum_{i=1}^{k-1} e_i = d_{k-1}. \qquad (21)$$

Since $d_k \ge d_i$ for any $i < k$, one has

$$\begin{aligned} \sum_{i=1}^{k} e_i &\le \sum_{i=1}^{k} dbf(\tau_i, d_k) \quad \text{(by definition of } dbf\text{)} \\ &\le dbf(\tau, d_k) \\ &\le d_k, \end{aligned}$$

where the last inequality holds because τ satisfies Condition (11). This, together with (21), leads to $e_k \le d_k - d_{k-1}$. By the assumption that $e_k \ne d_k - d_{k-1}$, we get

$$e_k < d_k - d_{k-1}. \qquad (22)$$

Construct $\tau' = \{\tau'_i = (e'_i, d'_i, p'_i) : 1 \le i \le n\}$ where

$$d'_i = d_i, p'_i = p_i, e'_i = e_i$$

for any $i \neq k$, and

$$e'_k = e_k, d'_k = d_{k-1} + e_k, p'_k = d_k + p_k - d'_k.$$

By (22), $d'_k < d_k$.

Obviously, $M(\tau') = M(\tau) - 1 < M(\tau)$, and τ' is constrained-deadline since so is τ.

Now we prove that τ' is a feasible solution to MP_1. Since τ satisfies Conditions (12)-(14), so does τ'. To show that Condition (11) is satisfied by τ', we arbitrarily choose $t > 0$ and proceed case by case.

Case 1: if $t < d'_k$. Then

$$\begin{aligned} dbf(\tau', t) &= \sum_{1 \le i \le n} dbf(\tau'_i, t) \\ &= \sum_{1 \le i < k} dbf(\tau'_i, t) \quad \text{(because } t < d'_j \text{ for } j \ge k) \\ &= \sum_{1 \le i < k} dbf(\tau_i, t) \quad \text{(because } \tau'_i = \tau_i \text{ for } i < k) \\ &\le dbf(\tau, t) \\ &\le t \quad \text{(because } \tau \text{ satisfies Condition (11)).} \end{aligned}$$

Case 2: if $d'_k \le t < d_k$.

$$\begin{aligned} dbf(\tau', t) &= \sum_{1 \le i \le n} dbf(\tau'_i, t) \\ &= \sum_{1 \le i \le k} \left(\left\lfloor \frac{t - d'_i}{p'_i} \right\rfloor + 1 \right) \cdot e'_i \\ &= \sum_{1 \le i \le k} e'_i \quad \text{(because } d'_i + p'_i > t \text{ for any } i) \\ &= \sum_{1 \le i \le k} e_i = d'_k \le t. \end{aligned}$$

Case 3: if $d_k \le t < d'_k + p'_k$. Then

$$\begin{aligned} dbf(\tau'_k, t) &= \left(\left\lfloor \frac{t - d'_k}{p'_k} \right\rfloor + 1 \right) \cdot e_k \\ &= e_k \quad \text{(because } d'_k < d_k \le t < d'_k + p'_k) \\ &= \left(\left\lfloor \frac{t - d_k}{p_k} \right\rfloor + 1 \right) \cdot e_k, \end{aligned}$$

where the last equality is due to $d_k \le t < d'_k + p'_k = d_k + p_k$.

For any $i \neq k$, $dbf(\tau'_i, t) = dbf(\tau_i, t)$ since $\tau'_i = \tau_i$.

As a result, $dbf(\tau', t) = dbf(\tau, t) \le t$ because τ satisfies Condition (11).

Case 4: if $t \ge d'_k + p'_k$. Because

$$d'_k < d_k \text{ and } p'_k + d'_k = d_k + p_k,$$

by Lemma 4, we have

$$\frac{t - d'_k}{p'_k} \le \frac{t - d_k}{p_k}.$$

Then

$$\begin{aligned} dbf(\tau', t) &= \sum_{1 \le i \le n} dbf(\tau'_i, t) \\ &= \sum_{i \neq k} dbf(\tau'_i, t) + \left(\left\lfloor \frac{t - d'_k}{p'_k} \right\rfloor + 1 \right) \cdot e_k \\ &\le \sum_{i \neq k} dbf(\tau'_i, t) + \left(\left\lfloor \frac{t - d_k}{p_k} \right\rfloor + 1 \right) \cdot e_k \\ &= \sum_{i \neq k} dbf(\tau_i, t) + dbf(\tau_k, t) \quad \text{(since } \tau'_i = \tau_i \text{ for } i \neq k) \\ &= dbf(\tau, t) \le t \quad \text{(since } \tau \text{ satisfies Condition (11)).} \end{aligned}$$

Altogether, τ' satisfies Condition (11), so it is a feasible solution to MP_1.

Finally, we show that

$$\frac{dbf^*(\tau, d_n)}{d_n} \le \frac{dbf^*(\tau', d'_n)}{d'_n}.$$

When $k < n$, we have $d'_n = d_n$, so it suffices to show $dbf^*(\tau, d_n) \le dbf^*(\tau', d'_n)$.

By definition of τ', for any $i \neq k$, $dbf^*(\tau_i, d_n) = dbf^*(\tau'_i, d'_n)$. Furthermore, note three facts:

1) $p'_k + d'_k = d_k + p_k$;
2) $d'_k < d_k$;
3) $d_n < d_k + p_k$ due to Conditions (12).

By Lemma 4, these facts mean

$$\frac{d_n - d_k}{p_k} \le \frac{d'_n - d'_k}{p'_k},$$

which implies $dbf^*(\tau_k, d_n) \le dbf^*(\tau'_k, d'_n)$.

As a result, $dbf^*(\tau, d_n) \le dbf^*(\tau', d'_n)$.

When $k = n$, we have $d'_n < d_n$. For any $i < n$,

$$\begin{aligned} \frac{dbf^*(\tau_i, d_n)}{d_n} &= \frac{e_i}{d_n} \left(1 + \frac{d_n - d_i}{p_i} \right) \\ &= \frac{e_i}{p_i} \left(1 + \frac{p_i - d_i}{d_n} \right) \\ &< \frac{e_i}{p_i} \left(1 + \frac{p_i - d_i}{d'_n} \right) \\ &= \frac{e_i}{d'_n} \left(1 + \frac{d'_n - d_i}{p_i} \right) \\ &= \frac{e'_i}{d'_n} \left(1 + \frac{d'_n - d'_i}{p'_i} \right) \quad \text{(because } \tau'_i = \tau_i) \\ &= \frac{dbf^*(\tau'_i, d'_n)}{d'_n} \end{aligned}$$

where the inequality is due to $d'_n < d_n$ and $p_i - d_i \ge 0$ (since τ is constrained-deadline). In addition,

$$\frac{dbf^*(\tau_n, d_n)}{d_n} = \frac{e_n}{d_n} < \frac{e'_n}{d'_n} = \frac{dbf^*(\tau'_n, d'_n)}{d'_n}.$$

Therefore, we also get $\frac{dbf^*(\tau, d_n)}{d_n} \le \frac{dbf^*(\tau', d'_n)}{d'_n}$, as desired. ■

We will impose further constraint on MP_2, without changing the optimum value. As presented in the math programming MP_3, the constraint is that all the e_i's are equal.

$$
\begin{aligned}
\sup \qquad & \frac{dbf^*(\tau, d_n)}{d_n}, & (MP_3) \qquad (23)\\
\mathit{subject\ to} \qquad & dbf(\tau, t) \le t, \quad \forall t > 0 & (24)\\
& d_i + p_i > d_n, \quad 1 \le i \le n-1, & (25)\\
& d_i = e_i + d_{i-1}, \quad 1 \le i \le n, & (26)\\
& e_i = d_n/n, \quad 1 \le i \le n, & (27)\\
& n \in \mathbb{Z}^+, e_i, d_i, p_i \in \mathbb{Q}^+, \quad 1 \le i \le n. & (28)
\end{aligned}
$$

Lemma 6: MP_2 and MP_3 have the same optimum value.

Proof: Let $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \le i \le n\}$ be an arbitrary feasible solution to MP_2. Due to Condition (20), we can choose $\delta \in \mathbb{Q}^+$ such that

$$k(i) \triangleq \frac{e_i}{\delta}$$

is an integer for any $1 \le i \le n$. Let $n' = \sum_{i=1}^{n} k(i)$.

For any $1 \le l \le n'$, define task $\tau'_l = (e'_l, d'_l, p'_l)$ as below, where $1 \le i \le n$ and $1 \le j \le k(i)$ are such that $l = m(i,j) \triangleq j + \sum_{1 \le h < i} k(h)$:

$$
\begin{aligned}
e'_l &= \delta,\\
d'_l &= d_{i-1} + \frac{j}{k(i)}(d_i - d_{i-1}) = d_{i-1} + j\delta,\\
p'_l &= p_i + d_i - d'_l.
\end{aligned}
$$

Let $\tau'(i) = \{\tau'_{m(i,j)} : 1 \le j \le k(i)\}$ for any $1 \le i \le n$, and $\tau' = \cup_{i=1}^{n} \tau'(i)$. Let $d'_0 = 0$. Next we will prove that τ' is a feasible solution to MP_3.

First of all, for any $1 \le i \le n$ and $1 \le j \le k(i)$, let $l = m(i,j)$. We have $d'_l \le d_i$ and $p'_l = p_i + d_i - d'_l \ge p_i$. Thus, τ' is constrained-deadline because so is τ.

Since τ' satisfies Conditions (25)-(28) by definition, now investigate Condition (24). Arbitrarily fix $t > 0$ and proceed case by case.

Case 1: $t < d'_{n'}$. Let integer $l \ge 0$ be such that $d'_l \le t < d'_{l+1}$. Then

$$
\begin{aligned}
&dbf(\tau', t) = \sum_{1 \le r \le n'} dbf(\tau'_r, t)\\
&= \sum_{1 \le r \le l} dbf(\tau'_r, t) \quad \text{(because } t < d'_{l+1}\text{)}\\
&= \sum_{1 \le r \le l} \left(\left\lfloor \frac{t - d'_r}{p'_r} \right\rfloor + 1 \right) \cdot e'_r\\
&= \sum_{1 \le r \le l} e'_r = d'_l \le t
\end{aligned}
$$

where the fourth equality holds due to the inequality $p'_r > t - d'_r$ which in turn follows from three facts:

1) For any $1 \le i \le n$ and $1 \le j \le k(i)$, we have

$$p'_{m(i,j)} = p_i + d_i - d'_{m(i,j)} \text{ by definition;}$$

2) From (18), $p_i + d_i > d_n$ holds $\forall i, 1 \le i \le n$;
3) $d_n = d'_{n'} > t$.

Case 2: $t \ge d'_{n'}$. It suffices to prove that $\forall i, 1 \le i \le n$,

$$dbf(\tau'(i), t) \le dbf(\tau_i, t).$$

Suppose $t < d_i + p_i$. We observe that

$$
\begin{aligned}
&dbf(\tau'(i), t) = \sum_{j=1}^{k(i)} dbf(\tau'_{m(i,j)}, t)\\
&= \sum_{j=1}^{k(i)} \left(\left\lfloor \frac{t - d'_{m(i,j)}}{p'_{m(i,j)}} \right\rfloor + 1 \right) \delta\\
&= k(i)\delta \quad \text{(because } t < d_i + p_i = d'_{m(i,j)} + p'_{m(i,j)}\text{)}\\
&= e_i \quad \text{(By definition of } k(i)\text{)}\\
&= dbf(\tau_i, t) \quad \text{(because } d_i \le t < d_i + p_i\text{)}
\end{aligned}
$$

Then consider $t \ge d_i + p_i$. For any $1 \le j \le k(i)$, since $d_i > d'_{m(i,j)}$ and $d_i + p_i = d'_{m(i,j)} + p'_{m(i,j)}$, Lemma 4 implies

$$\frac{t - d'_{m(i,j)}}{p'_{m(i,j)}} \le \frac{t - d_i}{p_i},$$

which further leads to

$$
\begin{aligned}
dbf(\tau'(i), t) &= \sum_{j=1}^{k(i)} \left(\left\lfloor \frac{t - d'_{m(i,j)}}{p'_{m(i,j)}} \right\rfloor + 1 \right) \delta\\
&\le \sum_{j=1}^{k(i)} \left(\left\lfloor \frac{t - d_i}{p_i} \right\rfloor + 1 \right) \delta\\
&= \left(\left\lfloor \frac{t - d_i}{p_i} \right\rfloor + 1 \right) e_i\\
&= dbf(\tau_i, t)
\end{aligned}
$$

Altogether, Condition (24) is satisfied in both cases, so τ' is a feasible solution to MP_3.

The rest of the proof is to show that

$$dbf^*(\tau', d'_{n'}) \ge dbf^*(\tau, d_n).$$

Note that

$$d'_{n'} = d_n < p_i + d_i = d'_{m(i,j)} + p'_{m(i,j)} \text{ and } d'_{m(i,j)} \le d_i$$

for any $1 \le i \le n, 1 \le j \le k(i)$. Lemma 4 implies that

$$\frac{d'_{n'} - d'_{m(i,j)}}{p'_{m(i,j)}} \ge \frac{d_n - d_i}{p_i}.$$

Then for any $1 \le i \le n$, we have

$$
\begin{aligned}
dbf^*(\tau'(i), d'_{n'}) &= \sum_{j=1}^{k(i)} \left(\frac{d'_{n'} - d'_{m(i,j)}}{p'_{m(i,j)}} + 1 \right) \delta\\
&\ge \left(\frac{d_n - d_i}{p_i} + 1 \right) e_i\\
&= dbf^*(\tau_i, d_n).
\end{aligned}
$$

Therefore, $dbf^*(\tau', d'_{n'}) \ge dbf^*(\tau, d_n)$. ■

It is still hard to find a good upper bound of the optimum value of MP_3, partly because Condition (24) is too strong and Condition (25) is too weak. It has to be modified accordingly.

On the one hand, we relax (24) by replacing the function $dbf(\cdot,\cdot)$ with $f(\cdot,\cdot)$: for any task $\tau_i = (e_i, d_i, p_i)$ and time $t > 0$,

$$f(\tau_i, t) = \begin{cases} dbf(\tau_i, t) & \text{if } t < d_i + p_i \\ 2e_i & \text{otherwise} \end{cases}$$

Note that $f(\tau_i, t) \le dbf(\tau_i, t)$ always holds. The first argument of f can be naturally extended to any set τ of tasks:

$$f(\tau, t) = \sum_{\tau_i \in \tau} f(\tau_i, t).$$

On the other hand, instead of (25), we require that the set of tasks τ should be aligned, as defined below:

Definition 1: Given a task set $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \le i \le n\}$, a permutation π over $\{1, 2, \cdots, n\}$ is called an *aligning permutation* of τ if

$$d_{\pi(i)} + p_{\pi(i)} = d_n + d_i$$

for any $1 \le i \le n$. τ is said to be *aligned* if it has an aligning permutation.

We will show that the optimum value of MP_3 does not decrease after the modification. Specifically, define a new math programming, where the tasks are *not* required to be constrained-deadline:

$$\begin{array}{lrr} \sup & \dfrac{dbf^*(\tau, d_n)}{d_n}, & (MP_4) \quad (29) \\ \textit{subject to} & f(\tau, t) \le t, \quad \forall t > 0 & (30) \\ & \tau \text{ is aligned}, & (31) \\ & d_i = e_i + d_{i-1}, \quad 1 \le i \le n, & (32) \\ & e_i = d_n/n, \quad 1 \le i \le n, & (33) \\ & n \in \mathbb{Z}^+, e_i, d_i, p_i \in \mathbb{Q}^+, \quad 1 \le i \le n. & (34) \end{array}$$

Lemma 7: The optimum value of MP_3 is not more than that of MP_4.

Proof: Arbitrarily choose a feasible solution $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \le i \le n\}$ to MP_3. Let π be a permutation over $\{1, 2, \cdots, n\}$ such that

$$d_{\pi(1)} + p_{\pi(1)} \le d_{\pi(2)} + p_{\pi(2)} \le ... \le d_{\pi(n)} + p_{\pi(n)}. \tag{35}$$

For any $1 \le i \le n$, construct a task $\tau'_i = (e'_i, d'_i, p'_i)$ where

$$e'_i = e_i, d'_i = d_i, p'_i = d_n + d_{\pi^{-1}(i)} - d'_i.$$

Let $\tau' = \{\tau'_i : 1 \le i \le n\}$.

We will show that τ' is a feasible solution to MP_4. Since Conditions (31)-(34) are satisfied by definition, it suffices to investigate Condition (30). Let's first derive an inequality as tool.

For any $1 \le i \le n$, let $j = \pi^{-1}(i)$, and we have

$$\begin{aligned} & d_i + p_i \\ \ge & dbf(\tau, d_i + p_i) \quad \text{(since } \tau \text{ satisfies Condition (24))} \\ = & \sum_{1 \le l \le j} dbf(\tau_{\pi(l)}, d_i + p_i) + \sum_{j < l \le n} dbf(\tau_{\pi(l)}, d_i + p_i) \\ \ge & \sum_{1 \le l \le j} 2e_{\pi(l)} + \sum_{j < l \le n} e_{\pi(l)} \\ = & \frac{2jd_n}{n} + \frac{(n-j)d_n}{n} \\ = & d_n + d_j \quad \text{(due to Conditions (26) and (27))}, \end{aligned}$$

where the second inequality is because

$$d_i + p_i = d_{\pi(j)} + p_{\pi(j)} \ge d_{\pi(l)} + p_{\pi(l)} \text{ for any } l \le j$$

and $d_i + p_i > d_n \ge d_{\pi(l)}$ for any l.

Hence, we have

$$d_i + p_i \ge d_n + d_{\pi^{-1}(i)} = d'_i + p'_i \tag{36}$$

by definition of τ'.

Now we continue to prove τ' satisfies Condition (30). For an arbitrary $t > 0$, this can be done case by case.

Case 1: $t < d_n + d_1$. Then for any $1 \le i \le n$,

$$\begin{aligned} & d_i + p_i \\ \ge & d'_i + p'_i = d_n + d_{\pi^{-1}(i)} \quad \text{by (36)} \\ \ge & d_n + d_1 > t \end{aligned}$$

This, together with the definition of τ', implies that

$$f(\tau', t) = dbf(\tau', t) = dbf(\tau, t).$$

Because τ satisfies Condition (24), we have $f(\tau', t) \le t$.

Case 2: $t \ge d_n + d_1$. Choose the biggest $1 \le i \le n$ such that $d_n + d_i \le t$. Then for any $j > i$,

$$p'_{\pi(j)} + d'_{\pi(j)} = d_n + d_j > t > d_n \ge d'_{\pi(j)} \ .$$

Thus

$$\begin{aligned} & f(\tau', t) \\ = & \sum_{1 \le j \le i} f(\tau'_{\pi(j)}, t) + \sum_{i < j \le n} f(\tau'_{\pi(j)}, t) \\ = & \sum_{1 \le j \le i} 2e'_{\pi(j)} + \sum_{i < j \le n} e'_{\pi(j)} \\ = & d_n + d_i \le t \end{aligned}$$

Altogether, Condition (30) is also satisfied.

Furthermore, for any $1 \le i \le n$, by (36) and $d_i = d'_i$, we have $p'_i \le p_i$. This, together with $e'_i = e_i, d'_i = d_i$ for any $1 \le i \le n$, implies $dbf^*(\tau', d'_n) \ge dbf^*(\tau, d_n)$. As a result,

$$\frac{dbf^*(\tau, d_n)}{d_n} \le \frac{dbf^*(\tau', d'_n)}{d'_n}.$$

The lemma thus holds. ■

We present a technical lemma before going on.

Lemma 8: For any $x_1, x_2, \cdots, x_n \in \mathbb{R}^+$ such that

$$\sum_{i=1}^{n} x_i = n^2,$$

we have

$$\sum_{i=1}^{n} \frac{i}{x_i} \geq \frac{4n}{9}.$$

Proof: By Cauchy's Inequality,

$$\left(\sum_{i=1}^{n} \frac{i}{x_i}\right)\left(\sum_{i=1}^{n} x_i\right) \geq \left(\sum_{i=1}^{n} \sqrt{i}\right)^2.$$

Note that

$$\begin{aligned}\sum_{i=1}^{n} \sqrt{i} &= n^{\frac{3}{2}}\left(\sum_{i=1}^{n} \frac{\sqrt{i}}{\sqrt{n}} \cdot \frac{1}{n}\right) \\ &\geq n^{\frac{3}{2}} \sum_{i=1}^{n} \int_{\frac{i-1}{n}}^{\frac{i}{n}} \sqrt{x}dx \\ &= n^{\frac{3}{2}} \int_{0}^{1} \sqrt{x}dx = \frac{2}{3}n^{\frac{3}{2}}.\end{aligned}$$

Therefore,

$$\sum_{i=1}^{n} \frac{i}{x_i} \geq \frac{\frac{4}{9}n^3}{n^2} = \frac{4n}{9}.$$

■

Lemma 9: The optimum value of MP_4 is at most $\frac{14}{9}$.

Proof: Arbitrarily choose a feasible solution $\tau = \{\tau_i = (e_i, d_i, p_i) : 1 \leq i \leq n\}$ to MP_4. Let $\delta = \frac{d_n}{n}$. By Conditions (32) and (33),

$$e_i = \delta \text{ and } d_i = i\delta$$

for any $1 \leq i \leq n$.

Let π be an aligning permutation of τ. Then we have

$$\sum_{i=1}^{n} p_{\pi(i)} = \sum_{i=1}^{n} (d_n + d_i - d_{\pi(i)}) = nd_n = n^2\delta,$$

which implies $\sum_{i=1}^{n} \frac{p_{\pi(i)}}{\delta} = n^2$. By Lemma 8,

$$\sum_{i=1}^{n} \frac{i\delta}{p_{\pi(i)}} \geq \frac{4n}{9}.$$

Hence,

$$\begin{aligned}\sum_{j=1}^{n} \frac{d_j + p_j - d_n}{p_j} &= \sum_{i=1}^{n} \frac{d_{\pi(i)} + p_{\pi(i)} - d_n}{p_{\pi(i)}} \\ &= \sum_{i=1}^{n} \frac{d_i}{p_{\pi(i)}} \quad \text{(since } \tau \text{ is aligned)} \\ &= \sum_{i=1}^{n} \frac{i\delta}{p_{\pi(i)}} \geq \frac{4n}{9}.\end{aligned}$$

As a result,

$$\begin{aligned}dbf^*(\tau, d_n) &= \sum_{j=1}^{n} dbf^*(\tau_j, d_n) \\ &= \sum_{j=1}^{n} \left(2 - \frac{p_j + d_j - d_n}{p_j}\right) e_j \\ &= \sum_{j=1}^{n} \left(2 - \frac{p_j + d_j - d_n}{p_j}\right) \delta \\ &\leq 2n\delta - \delta \sum_{j=1}^{n} \frac{p_j + d_j - d_n}{p_j} \\ &\leq 2n\delta - \frac{4n}{9}\delta = \frac{14}{9}d_n\end{aligned}$$

The lemma holds. ■

We are ready to present one of the main results of this paper, which claims that the quantity ρ is at most 1.5556.

Theorem 1: $\frac{dbf^*(\tau,d)}{d} \leq \frac{14}{9}$ for any set τ of constrained-deadline tasks such that $dbf(\tau, t) \leq t$ for all $t > 0$, where d is the maximum relative deadline of the tasks in τ.

Proof: It follows from Lemmas 3, 5, 6, 7, and 9. ■

IV. Partitioned Scheduling on Multiprocessors

This section is devoted to partitioning sporadic tasks on multiprocessors, where the tasks are assumed to have constrained deadlines. We adopt the algorithm of Deadline-Monotonic Partitioning in [1]. It is presented in Algorithm 1 to make this paper self-contained, where e_i and d_i stand for worst-case execution time and relative deadline of task τ_i, respectively.

Algorithm 1 Deadline-Monotonic Partitioning

```
Input: sporadic tasks $\tau = \{\tau_1, \ldots, \tau_n\}$ to be partitioned on $m$ identical unit-capacity processors;
    ▷ The tasks are indexed non-decreasingly according to their relative deadlines. For any $1 \leq k \leq m$, let $\tau(k)$ denote the set of tasks assigned to the $k$th processor.
$\tau(k) \leftarrow \emptyset$, for any $1 \leq k \leq m$;
for $i = 1$ to $n$ do
    if there exists $k$ such that $e_i + dbf^*(\tau(k), d_i) \leq d_i$ then
        Choose the smallest such $k$;
        $\tau(k) \leftarrow \tau(k) \cup \{\tau_i\}$;
    else
        return FAIL
    end if
end for
return feasible assignment $\tau(1), \tau(2), \ldots, \tau(m)$
```

Basically, Algorithm 1 assigns tasks sequentially in the order of non-decreasing relative deadlines. Suppose $\tau(k)$ is the set of tasks at processor k after the first $i - 1$ tasks has been assigned. Then task τ_i is assigned to the first processor (say, processor No. k) that can safely serve the task, namely $e_i + dbf^*(\tau(k), d_i) \le d_i$.

Remember that we have upper-bounded

$$\rho = \sup_{\tau} \frac{dbf^*(\tau, d)}{d}, \tag{37}$$

where τ ranges over constrained-deadline sporadic task sets that are feasible on uniprocessors, and d is the largest relative deadline in τ.

The following lemma is from references [1] and [2], so the proof is omitted.

Lemma 10: The speedup factor of Algorithm 1 is $1 + \rho - 1/m$, where m is the number of processors.

It is time to present the other main result of this paper.

Theorem 2: The speedup factor of Algorithm 1 is at most $2.5556 - 1/m$.

Proof: The theorem immediately follows from Theorem 1 and Lemma 10. ∎

V. Conclusion and Future Work

In this paper, we improve the upper bound of the speedup factor of partitioned-EDF from $2.6322 - 1/m$ to $2.5556 - 1/m$ for constrained-deadline sporadic tasks on m identical processors, narrowing the gap between the upper and the lower bounds from 0.1322 to 0.0556. This is an immediate corollary of our improvement of the upper bound of $\rho = \sup_{\tau} dbf^*(\tau, d)/d$ from 1.6322 to 1.5556.

Technically, our improvements root at a novel discretization that transforms the tasks into regular form without decreasing ρ. The discretization essentially restricts attentions to the tasks with fixed execution times and deadlines. Only the period parameter remains flexible to some extent—ranging over the set $\{1, 2, \cdots, 2n\}$, where n is the number of tasks to be scheduled. With such transformation, the estimation of ρ is reduced to a much simpler optimization problem. We believe that this knack may be applied to other problems or scenarios.

However, we have not yet proved that our transformation is lossless. This means that the discretization might strictly enlarge ρ. The good news is that the incurred loss, if not zero at all, is guaranteed to be no more than 0.0556.

As to future directions, we conjecture that Theorems 1 and 2 remain true if the constrained-deadline condition is removed. We also conjecture that a 1.5 upper bound for ρ can be derived with our method, thus closing the gap between the upper and the lower bounds. If this is the case, the speedup factor of partitioned-EDF becomes also fully determined, at least in the case of constrained deadlines.

Acknowledgment

The authors would like to thank Prof. Sanjoy Baruah from Washington University at St. Louis and Prof. Yungang Bao from Institute of Computing Technology, CAS for the fruitful discussions. This work is partially supported by the National Key Research and Development Program of China (Grant No. 2016YFB1000200), the National Natural Science Foundation of China (11101065, 61420106013), National Science Foundation (of the US, CNS-1837472 and CNS-1850851), State Key Laboratory of Computer Architecture Open Fund (CARCH3410).

References

[1] 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.

[2] 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.

[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] A. K.-L. Mok, "Fundamental design problems of distributed systems for the hard-real-time environment," Ph.D. dissertation, Massachusetts Institute of Technology, 1983.

[5] 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.

[6] 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.

[7] K. Agrawal and S. Baruah, "Intractability issues in mixed-criticality scheduling," in *the 30th Euromicro Conference on Real-Time Systems (ECRTS'18)*, 2018, to appear.

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

[9] C. A. Phillips, C. Stein, E. Torng, and J. Wein, "Optimal time-critical scheduling via resource augmentation," *Algorithmica*, vol. 32, no. 2, pp. 163–200, 2002.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] E. Bini and G. C. Buttazzo, "Measuring the performance of schedulability tests," *Real-Time Systems*, vol. 30, no. 1-2, pp. 129–154, 2005.

[15] E. Bini, "The quadratic utilization upper bound for arbitrary deadline real-time tasks," *IEEE Trans. Computers*, vol. 64, no. 2, pp. 593–599, 2015.

[16] J. Theis and G. Fohler, "Transformation of sporadic tasks for off-line scheduling with utilization and response time trade-offs," in *19th International Conference on Real-Time and Network Systems, RTNS '11, Nantes, France, September 29-30, 2011. Proceedings*, 2011, pp. 119–128.

[17] 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.