Temperature Minimization and Thermal-Driven Scheduling for Real-Time Periodic Tasks

Article *in* Journal of Signal Processing Systems · June 2018

DOI: 10.1007/s11265-018-1390-7

Tiantian Li
Northeastern University (Shenyang, China)
14 PUBLICATIONS **12** CITATIONS

SEE PROFILE

Jie Song
Northeastern University (Shenyang, China)
24 PUBLICATIONS **109** CITATIONS

SEE PROFILE

Journal of Signal Processing Systems
https://doi.org/10.1007/s11265-018-1390-7

Temperature Minimization and Thermal-Driven Scheduling for Real-Time Periodic Tasks

Tiantian Li[1] · Tianyu Zhang[1] · Ge Yu[1] · Jie Song[2]

Received: 1 January 2018 / Revised: 20 April 2018 / Accepted: 13 June 2018

Abstract

With the increasing power density of processors due to the continuous shrinking of chip size, thermal issue of real-time systems has become more and more urgent since it can greatly affect the systems' reliability and safety. However, due to the difficulty in quantitative analysis of temperature, existing thermal-aware research either considers temperature as a constraint to minimize energy consumption treating leakage power as a constant, or minimizes the temperature qualitatively through approaches like cool-hot task execution pattern. Even the latest work also only analyzes the average temperature in thermal steady state. To this end, this paper aims at quantitatively minimizing the temperature in both steady and transient states while considering the effect of temperature on leakage power. The following contributions have been made: presents a method of task construction, based on which the minimal temperature condition for any scheduling algorithm like GPS (Global Processor Sharing) and EDF (Earliest Deadline First) under both steady and transient states is proved; discovers a thermal law exhibited by tasks' power consumption, based on which a thermal-aware approximate algorithm of GPS with a relatively longer scheduling length is designed to mitigate the higher switching overhead of GPS scheduling while guaranteeing the tasks' timing constraints. Sufficient experiments validate the minimal temperature condition and thermal-aware scheduling algorithm. The two important conclusions obtained from this work are: (1) when the optimal condition can be achieved, all of the scheduling algorithms like GPS and EDF result in the same minimal temperature traces since all of the tasks share the same power consumption value and the processor is fully utilized; (2) when not, the executing sequence of tasks in a scheduling interval of the approximate GPS scheduling, which is determined based on the thermal law, exhibits a better thermal-aware feature.

Keywords Temperature minimization · Task construction · Thermal law · Thermal-driven task scheduling · Real-time periodic task

1 Introduction

Embedded real-time systems are widely used to monitor and control the dynamics of the underlying physical processes in many cyber-physical applications. With the ever-increasing power density of processors caused by the aggressive shrinking of chip size and the integration of various processing elements on a single die to further improve the system's computing capacity, both energy and thermal issues have become very urgent to cope with. Increase in energy may cause serious technical and economic problems especially for battery-powered real-time embedded system, while increase in temperature may greatly damage the

✉ Tiantian Li
litiantian_neu@163.com

Tianyu Zhang
tyzhang@stumail.neu.edu.cn

Ge Yu
yuge@mail.neu.edu.cn

Jie Song
songjie@mail.neu.edu.cn

[1] School of Computer Science and Engineering of Northeastern University, Shenyang, 110169, China

[2] Software College of Northeastern University, Shenyang, 110169, China

Published online: 23 June 2018

performance, reliability and safety of the systems. Further, there exists a vicious circle between power and temperature: higher power results in higher temperature which results in higher leakage power. Therefore, it is necessary to seek task scheduling approaches with taking both energy and temperature minimization into consideration.

In the last decade, researchers in this domain have developed many optimization approaches for meeting the time and temperature constraints. They deal with different processor models (unicore, multicore, or multiprocessor), different application models (preemptive, nonpreemptive, resource constrained, energy constrained), different task models (periodic, aperiodic, sporadic), and different task needs (hard, soft) of the timing constraints. However, due to the difficulty in quantitative analysis of temperature, existing work generally has one or more of the following limitations: (1) Adopts a restrictive power model, not taking leakage power into consideration or not considering the vastly different power characteristics exhibited by different tasks which can greatly affect processor temperature; (2) Temperature optimization methods proposed are mostly qualitative rather than quantitative, which usually consider temperature threshold as a constraint to solve a given power or throughput optimization problem rather than decrease the temperature to a largest extent; (3) Existing scheduling algorithms usually take time as the metric to determine the task priorities, barely considering temperature as the metric, which can be proved to be able to further decrease the temperature.

This work addresses the above three problems. For the first two problems, we aim to find the minimal temperature condition for multi-tasks by considering both the temperature-dependent leakage power and the different task power features. For the last problem, we explore the thermal law for real-time tasks and based on which propose an approximate GPS scheduling which can further minimize the temperature while guaranteeing the tasks' timing constraints.The main contributions of this work are:

- Propose a method of end-temperature-equivalently task construction, laying the theoretical foundation to prove the presented minimal temperature condition;
- Prove the minimal temperature condition for GPS scheduling under both steady and transient states, and further generalize it to other scheduling algorithms like EDF;
- Discover the thermal law of real-time systems: the end processor temperature can be decreased when tasks are executed in a decreasing order by their dynamic power consumption;
- Design and implement a thermal-aware approximate GPS scheduling algorithm with temperature as the metric to determine the task priorities based on the thermal law and minimal temperature condition presented in this paper.

We obtain two important conclusions from this work: (1) when the optimal temperature condition can be achieved, all of the scheduling algorithms like GPS and EDF result in the same minimal temperature traces since all of the tasks share the same power consumption value and the processor is fully utilized; (2) when not, the executing sequence of tasks in a scheduling interval of the approximate GPS scheduling that is determined based on the thermal law, exhibits a better thermal-aware feature. When the optimal temperature condition cannot be achieved, GPS scheduling results in lower temperature than other scheduling algorithms due to its ability to eliminate the oscillating feature of the processor energy and temperature. However, GPS scheduling causes much task switching overhead due to its execution pattern. To mitigate this problem, the approximate GPS scheduling with a longer scheduling length is proposed, in which the executing sequence of tasks in a scheduling interval matters. Further, as mentioned above, the executing sequence determined by the thermal law we find shows a better thermal-aware feature.

The rest of this article is organized as follows: Section 2 covers the related work addressing the thermal issues. Section 3 describes the three basic models adopted to conduct this research. Section 4 first depicts a method of task construction, and then using this method deduces the minimal temperature condition with GPS scheduling, and generalizes it to any scheduling algorithm. Section 5 presents the thermal law we find and proposes a thermal-aware approximate algorithm of GPS scheduling. Validation of the minimal temperature condition and thermal-aware scheduling algorithm has been shown in Section 6 through plenty of experiments. This work is summarized in Section 7.

2 Related Work

For any research in this domain, determination of the system models is usually the first thing to do. They include three models: power model, thermal model and task model. However, the power models adopted by existing works usually fail to capture the critical factors like the temperature-dependent leakage power and the heterogeneity of tasks in the sense of dynamic power consumption, which can significantly affect the optimization effect of their proposed approaches. For instance, Xie et al. [1] treats the leakage power as a constant, ignoring its changeability with temperature; Lee et al. [2] only considers dynamic power, ignoring leakage power which has taken quite a large account due to the increasing power density; Niu et al. [3] proposes a hybrid approach to find the energy-efficient speeds for real-time tasks, but it also treats the leakage power as a constant rather than a variant changing with temperature; Jejurikar et al. [4] considers dynamic power to be only a simple function of processor speed, not considering the power differences of real-time tasks due to their different computing features. About the leakage power model, the one proposed by Liao et al. [5] is accurate enough but not easy to use during optimization research. Therefore, Liu et al. [6] and Yang et al. [7] simplify it as linear model and quadratic model respectively. The linear model has been widely adopted by many works [8, 9] due to its simplicity and accuracy, which is also adopted by this work. As to the thermal model, the basis to conduct thermal-aware research, RC [10] model is widely used, which is also adopted in this work. Task model is different from studies; it depends on the research object, which can be periodic, aperiodic or sporadic. This work aims at periodic task model.

As to temperature aware optimization studies, they can be roughly categorized into two types: single task optimization and multi-task scheduling. However, existing works usually only consider temperature as a constraint to solve a power or throughput optimization problem rather than minimize it. As to the former type, TALK [11], PB [7] and MO [12] are the representative ones. All of them try to decide a better running pattern for the processor to satisfy the temperature threshold, such as PB's work-sleep running pattern and MO's high-low speed running pattern. As to the latter type, Jayaseelan et al. [13] and Huang et al. [8] first divide tasks into hot ones and cool ones based on their corresponding processor temperature in steady state, and then execute tasks in pairs of a hot task and a cool task to meet temperature constraint. Other related works [14–16] propose several coarse-grained task-to-core allocation strategies to decrease temperature. The basic idea is to allocate hot tasks to cores that have little effects on the other ones, and cool tasks to cores that have much effects on the other ones. Hanumaiah et al. [17] integrates online DVFS, task migration and fan speed controlling techniques to guarantee temperature constraint. Quan et al. [18] presents necessary and sufficient conditions for real-time schedules to guarantee the maximal temperature constraint. However, as mentioned above, all these works only consider temperature as a constraint rather than minimize it, which cannot fully explore the energy and temperature optimization space due to vicious circle between power and temperature.

Following researches consider temperature-dependent leakage power and try to minimize it through slack distribution. Zhou et al. [19] proposed a two-stage task scheduling, which minimizes the dynamic energy consumption by a heuristic task assignment approach in the first stage, and minimizes the leakage energy consumption by reducing the processor temperature through slack distribution. However, this work does not fully exploit the potentials to minimize the temperature and energy consumption. First, the task assignment is designed to minimize only the dynamic energy consumption, rather than the overall energy consumption or the system temperature; Second, it fully utilizes the processors' computing resources when assigning tasks to processors, thus reserving little optimization room for either slack distribution or speed scaling; Third, the slack distribution is designed to minimize the temperature thus decreasing the leakage energy consumption, while its efficiency is lower compared with the dynamic frequency/speed scaling technique; Fourth, the RM scheduling it adopts on one hand is not temperature friendly in the sense of its oscillating feature on temperature caused by the different task powers, and on the other hand has a complex feasibility analysis. Zhou et al. [20] assigned the tasks to processors in a decreasing order of the difference of steady state temperature on different processors to minimize the peak temperature among processors, and split the tasks with slacks to further minimize the peak temperature of a processor. However, the task assignment heuristic proposed is based on the thermal profiles of tasks rather than the processors, which may result in huge difference when applied in practice; the task splitting optimization aims at the tasks one by one rather than treating them as a whole; further, the last three problems described above also exist in this work. Further, Zhou et al. [21] minimize the energy consumption of processors with dynamic voltage/frequency scaling by transforming the processor into a virtual multi-processor model supporting only one frequency level. However, only the second problem mentioned above is addressed while the other two problems still remain to be exploited. In this work, we intend to find the optimal speed assignment to average the dynamic power consumption of all tasks so as to minimize both the temperature and energy consumption.

The studies by Ahmed et al. [22, 23] are the most related to this work. Inspired by computation utilization, they address the problem of thermal utilization. They derive the necessary and sufficient conditions for thermal feasibility of periodic task sets on a unicore system. However, they only analyze the temperature profile under steady state, while this work intends to derive the minimal temperature condition by speed scaling for any scheduling algorithm that can fully utilize the processor under both steady and transient states.

In addition, scheduling algorithm (SA) is another study aspect of real-time systems. All of the optimization works have to be finally implemented based on a specific scheduling algorithm. Existing SAs, like RM (Rate Monotonic) with fixed priority, EDF (Earliest Deadline First) and LLF (Least Laxity First) with dynamic priority, usually take time as the metric to determine the task priorities, barely considering temperature as the metric, which can be proved to be able to further decrease the temperature in this work.

The limitations of current works will be addressed by this work through the following three aspects: Adopt a system model that not only take temperature-dependent leakage power into consideration but also considers the different power characteristics exhibited by different tasks which can greatly affect processor temperature; Quantitatively analyze the temperature so as to find the minimal temperature condition by speed scaling for multiple periodic real-time tasks; Take temperature as the metric to determine the task executing sequence to further minimize the temperature.

3 System Models and Problem Definition

This section introduces the system models we use, including power model, thermal model and task model. The symbols used in this work are described in Table 1.

3.1 Power Model

A uniprocessor is considered in this research, whose power consumption consists of two parts: (i) dynamic power (P_{dyn}) due to switching of gate when executing tasks; (ii) static power (P_{lkg}) due to leakage current which is related with processor temperature and is unavoidable as long as the processor is active. The power model adopted here is as follows:

$$\begin{aligned} P_{\mathrm{dyn}}(t) &= A(t)s(t)^3,\ P_{\mathrm{lkg}}(\Theta(t)) = \alpha\Theta(t) + \beta, \\ P(t, \Theta(t)) &= P_{\mathrm{dyn}}(t) + P_{\mathrm{lkg}}(\Theta(t)), \end{aligned} \tag{1}$$

where $s(t)$ is the processor speed at moment t, $A(t)$ is task-related switching factor, $\Theta(t)$ is the processor temperature at moment t, α and β are linear fitting constants that are related to processor voltage. In practice, α and β can be treated as constants due to the tight changing range of voltage.

Table 1 Symbols notation.

Category	Symbol	Meaning	Unit
Task related	τ_i	Real-time task τ_i	–
	C_i	Worst case execution time of τ_i	ms
	T_i	Period of τ_i	ms
	D_i	Relative deadline of τ_i	ms
	A_i	Switching activity of τ_i	–
	s_i	Speed assigned to τ_i	/s
	P_{dyn}^i	Dynamic power consumption of τ_i	W
System related	Θ	Temperature of the processor	K
	R_{th}	Thermal resistence	K/W
	C_{th}	Thermal capacitance	J/K
	Θ_a	Ambiant temperature	K
	Θ_δ	Constrained temperature	K
Leakage power related	α	Temperature related coefficient	W/K
	β	Temperature irrelevant coefficient	W

Here, P_{lkg} is modeled as a linear function of temperature, which has been proved accurate enough within typical processor temperature range (25 °C–120 °C) under 65 nm manufacturing technology by Liu et al. [6] and has been widely adopted by many research works [8, 9, 22, 23].

3.2 Thermal Model

Due to the duality between heat and electronic transfer, processor temperature can be characterized as the following differential equation, which is usually called RC thermal model [10] and has been widely adopted by lots of researches.

$$\frac{d\Theta(t)}{dt} = \frac{P(t)}{C_{\mathrm{th}}} - \frac{\Theta(t) - \Theta_a}{R_{\mathrm{th}}C_{\mathrm{th}}} \tag{2}$$

For the sake of conciseness, $\Theta(t)$ can be adjusted to simplify the RC model as follows:

$$\begin{aligned} \theta(t) &= C_{\mathrm{th}}\Theta(t) - \frac{C_{\mathrm{th}}(R_{\mathrm{th}}\beta + \Theta_a)}{1 - R_{\mathrm{th}}\alpha}, \\ \theta'(t) &= P_{\mathrm{dyn}}(t) - \lambda\theta(t), \lambda = \frac{1}{R_{\mathrm{th}}C_{\mathrm{th}}} - \frac{\alpha}{C_{\mathrm{th}}}. \end{aligned} \tag{3}$$

Note that λ is related to R_{th}, C_{th} and linear fitting constant α. Then, the temperature function can be derived as follows:

$$\theta(t) = \int_0^t P_{\mathrm{dyn}}(t)e^{\lambda(u-t)}du + \theta(0)e^{-\lambda t}, \tag{4}$$

where $\theta(0)$ is the adjusted initial temperature at moment 0. Hereafter, temperature refers to the adjusted temperature unless specified otherwise.

3.3 Task Model

This research considers periodic real-time tasks. Each periodic task τ_i is usually characterized through a three-way tuple, i.e. $\tau_i =< C_i, T_i, D_i >$, indicating the task's worst case execution time under maximal speed ($s_{\max} = 1$ when normalized), period and deadline respectively. Since we make the assumption of $D_i = T_i$ without loss of generality, D_i is omitted here. In the mean-time, the difference in dynamic power consumption due to different switching activity across different tasks is taken into consideration through a variable A_i, and the speed assigned to task τ_i is denoted as s_i. Thus, a periodic task can be depicted as $\tau_i =< C_i/s_i, T_i, A_i >$. In addition, for the convenience to use, it can be simplifies as $\tau_i =< C_i/s_i, T_i, P_{\mathrm{dyn}}^i >$, combing variables A_i and s_i into dynamic power consumption of τ_i.

Suppose that all instances of task τ_i will be executed at the same speed s_i, then by Eq. 4, the temperature during executing the jth instance of τ_i can be deduced as:

$$\begin{aligned}\theta_{i,s_i}(t) &= \frac{P^i_{\text{dyn}}}{\lambda}(1 - e^{\lambda(t_0 - t)}) + \theta(t_0)e^{\lambda(t_0-t)} \\ &= \frac{P^i_{\text{dyn}}}{\lambda} - \left(\frac{P^i_{\text{dyn}}}{\lambda} - \theta(t_0)\right) e^{\lambda(t_0-t)}, \end{aligned} \qquad (5)$$

where t_0 is the referential time.

3.4 Preliminaries

Different scheduling algorithms would result in different task execution patterns, thus causing different temperature profiles over time and hence different dynamic power and leakage power. Therefore, this section briefly introduces the scheduling algorithms mentioned in this work to provide insights into the scheduling problem aimed by this work.

Rate Monotonic (RM) Scheduling: at each moment select the task with the shortest period to execute.

Earliest Deadline (EDF) Scheduling: at each moment select the task with the earliest deadline to execute.

Global Processor Sharing (GPS) Scheduling [24]: at each moment execute all of the tasks with fixed execution rates (such as $\omega_i = C_i/T_i s_i$, which can make the task finish just before its deadline). GPS scheduling is also called as fluid scheduling, and has been adopted recently by many works [25, 26].

For all of the there scheduling algorithms, the key problem to minimize processor temperature and energy consumption is to determine the optimal execution speeds for all of the tasks. In particular, for GPS scheduling, we also need to determine the corresponding execution rates for tasks, which can significantly affect the optimization results.

3.5 Problem Definition

For a given independent periodic task set $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$, with each task $\tau_i =< C_i/s_i, T_i, A_i >$, find the optimal speed assignment and the corresponding execution rates for all tasks under the GPS scheduling, with which the processor temperature can be minimized hence minimizing the energy consumption while guaranteeing the tasks' timing constraints.

4 Minimal Temperature Condition

This section first describes a method of task construction, then using this method to deduce the minimal temperature condition while meeting the timing constraints. We first deduce it with GPS (Generalized Processor Sharing) scheduling under both steady and transient states and then generalize it with other scheduling algorithms. Since when the initial temperature is relatively higher, the temperature at the end of each hyperperiod will keep decreasing, thus making it unnecessary to explore the minimal temperature condition. Hence, this paper aims at a lower one. Finally, we also make a feasible solution analysis of the minimal condition.

4.1 Task Construction

Before we introduce the method of task construction, we first give the definition of task temperature feature, which reflects a task's thermal effect under different setups of speed and initial temperature.

Definition 1 (**Task Temperature Feature (TemF)**) The end processor temperature when task τ_i is executed for Δ_i time units nonpreemptively with constant speed s_i and initial temperature θ_0, denoted as $\text{TemF}_{i,s_i}|_{\theta_0}$.

According to Eq. 5, $\text{TemF}_{i,s_i}|_{\theta_0}$ can be calculated as:

$$\begin{aligned}\text{TemF}_{i,s_i}|_{\theta_0} &= \frac{P^i_{\text{dyn}}}{\lambda}(1 - e^{-\lambda\Delta_i}) + \theta_0 e^{-\lambda\Delta_i} \\ &= \frac{P^i_{\text{dyn}}}{\lambda} - \left(\frac{P^i_{\text{dyn}}}{\lambda} - \theta_0\right) e^{-\lambda\Delta_i}. \end{aligned} \qquad (6)$$

From Eq. 6, we can see that $\text{TemF}_{i,s_i}|_{\theta_0}$ is the weighted summation of P^i_{dyn}/λ and θ_0, and it increases with both of them.

Now, we describe the task construction method first for two tasks then for multi-tasks. We aim at the following scenario:

> **Scenario I:** For $\tau_i =< C_i/s_i, T_i, P^i_{\text{dyn}} >$ and $\tau_j =< C_j/s_j, T_j, P^j_{\text{dyn}} >$, consider a situation of nonpreemptively executing τ_i and τ_j for Δ_i and Δ_j time units consecutively with initial temperature of θ_0.

Here, although tasks are executed preemptively, we treat the jobs executing in the nonpreemtively time segments separately although they may belong to the same task. For the sake of simplicity, we still call the nonpreemptively executed jobs as tasks. Denote $\theta_e^{x_1,x_2,...,x_n}|_{\theta_0}$ as the end processor temperature where $< x_1, x_2, ..., x_n >$ is a permutation of $< 1, 2, ..., n >$, corresponding to a schedule sequence of $< \tau_{x_1}, \tau_{x_2}, ..., \tau_{x_n} >$.

Lemma 1 (**Task Construction**) *Without considering the periodicity of tasks, τ_i and τ_j can be end-temperature-equivalently (i.e. $\theta_e^{i,j}|_{\theta_0} = \theta_e^{j,i}|_{\theta_0}$) constructed as one single* *task τ_{ij} with P^{ij}_{dyn} executing for Δ_{ij} time units, where $\Delta_{ij} = \Delta_i + \Delta_j$ and*

$$\begin{aligned} P^{ij}_{\text{dyn}} &= \frac{e^{-\lambda\Delta_j}(1-e^{-\lambda\Delta_i})}{1-e^{-\lambda(\Delta_i+\Delta_j)}}P^i_{\text{dyn}} + \frac{(1-e^{-\lambda\Delta_j})}{1-e^{-\lambda(\Delta_i+\Delta_j)}}P^j_{\text{dyn}}, \\ &\min_{\tau_i,\tau_j} P^{\tau}_{\text{dyn}} \leqslant P^{ij}_{\text{dyn}} \leqslant \max_{\tau_i,\tau_j} P^{\tau}_{\text{dyn}}. \end{aligned} \tag{7}$$

When $P^i_{\text{dyn}} = P^j_{\text{dyn}}$, $P^{ij}_{\text{dyn}} = P^i_{\text{dyn}} = P^j_{\text{dyn}}$.

Proof By Eq. 6, $\theta_{\text{e}}^{i,j}|_{\theta_0}$ can be calculated and transformed as follows:

$$\begin{aligned} \theta_{\text{e}}^{i,j}|_{\theta_0} &= \frac{P^j_{\text{dyn}}}{\lambda} - \left[\frac{P^j_{\text{dyn}}}{\lambda} - \left(\frac{P^i_{\text{dyn}}}{\lambda} - \left(\frac{P^i_{\text{dyn}}}{\lambda} - \theta_0\right)e^{-\lambda\Delta_i}\right)\right] \\ &\quad \times e^{-\lambda\Delta_j} \\ &= \frac{P^j_{\text{dyn}} + \Delta_{P_j}}{\lambda} - \left(\frac{P^j_{\text{dyn}} + \Delta_{P_j}}{\lambda} - \theta_0\right)e^{-\lambda(\Delta_i+\Delta_j)} \\ &= \frac{P^i_{\text{dyn}} - \Delta_{P_i}}{\lambda} - \left(\frac{P^i_{\text{dyn}} - \Delta_{P_i}}{\lambda} - \theta_0\right)e^{-\lambda(\Delta_i+\Delta_j)} \\ &= \frac{P^{ij}_{\text{dyn}}}{\lambda} - \left(\frac{P^{ij}_{\text{dyn}}}{\lambda} - \theta_0\right)e^{-\lambda\Delta_{ij}} = \theta_{\text{e}}^{ij}|_{\theta_0}, \\ P^{ij}_{\text{dyn}} &= P^i_{\text{dyn}} - \Delta_{P_i} = P^j_{\text{dyn}} + \Delta_{P_j},\ \Delta_{ij} = \Delta_i + \Delta_j, \\ \Delta_{P_i} &= \frac{(P^i_{\text{dyn}} - P^j_{\text{dyn}})(1-e^{-\lambda\Delta_j})}{1-e^{-\lambda\Delta_{ij}}}, \\ \Delta_{P_j} &= \frac{(P^i_{\text{dyn}} - P^j_{\text{dyn}})e^{-\lambda\Delta_j}(1-e^{-\lambda\Delta_i})}{1-e^{-\lambda\Delta_{ij}}}. \end{aligned} \tag{8}$$

We can easily reach the declaration in the lemma through the above transformation. □

For multi-tasks, the following scenario is considered to describe the method of task construction:

> **Scenario II:** For a task set $\Gamma_n = \{\tau_1, \tau_2, ..., \tau_n\}$, $\tau_\tau =< C_\tau/s_\tau, T_\tau, P^\tau_{\text{dyn}} >$, consider a situation of nonpreemptively executing each task τ_τ for Δ_τ time units consecutively with initial temperature of θ_0.

Corollary 1 (**Task Construction**) *The n tasks can be end-temperature-equivalently (i.e. $\theta_{\text{e}}^{1,2,...,n}|_{\theta_0} = \theta_{\text{e}}^{12...n}|_{\theta_0}$) constructed as one single task $\tau_{12...n}$ executing for $\Delta_{12...n}$ time units ($\Delta_{12...n} = \sum_1^n \Delta_i$) and $P^{12...n}_{\text{dyn}}$ can be recursively calculated by* Eq. 7.

$$\min_{\Gamma_n} P^\tau_{\text{dyn}} \leqslant P^{12...n}_{\text{dyn}} \leqslant \max_{\Gamma_n} P^\tau_{\text{dyn}}. \tag{9}$$

Proof This lemma can be easily proved through recursively leveraging Lemma 1. Hence, the detailed proof is omitted due to the space limit. □

4.2 The Case of GPS Scheduling

Lemma 1 and Corollary 1 do not consider the periodicity of tasks due to the nondeterminacy of the adjacently executed tasks combination in different time interval when RM or EDF is used. However, this problem is eliminated with GPS scheduling [24].

For the convenience of description, we assume a scheduling length Δ which approaches 0 to depict a GPS scheduling. During any time interval $[k\Delta, (k+1)\Delta]$ in the mth hyperperiod, each task τ_τ will be nonpreemptively executed for $\Delta \cdot \omega_\tau (\omega_\tau = C_\tau/T_\tau s_\tau)$ time units with a fixed relative start time of st_τ, and the execution sequence always keeps the same. This execution rates assignment guarantees the tasks completing right at their deadlines (which are assumed to be equal to their periods in this work). Based on the method of task construction presented above, we further propose another two improved lemmas as shown below.

Lemma 2 *With GPS scheduling, adjacently executed tasks τ_i and τ_j can be end-temperature-equivalently constructed as one periodic task τ_{ij} featured as*

$$\begin{aligned} \tau_{ij} &= < C_{ij}/s_{ij}, T_{ij}, P^{ij}_{\text{dyn}} > \\ &= < L_{ij}\omega_{ij}, L_{ij}, \left(\frac{\omega_i}{\omega_{ij}}P^i_{\text{dyn}} + \frac{\omega_j}{\omega_{ij}}P^j_{\text{dyn}}\right) >, \\ L_{ij} &= LCM\{T_i, T_j\},\ \omega_{ij} = \omega_i + \omega_j. \end{aligned} \tag{10}$$

Proof With GPS scheduling, τ_i and τ_j will nonpreemptively execute for $\Delta \cdot \omega_i$ and $\Delta \cdot \omega_j$ time units consecutively with initial temperature $\theta(k\Delta + st_{i,j})$ during any time interval $[k\Delta, (k+1)\Delta]$. By Lemma 1, τ_i and τ_j can be end-temperature-equivalently constructed as one single task τ_{ij} executing $\Delta \cdot \omega_{ij}$ time units, where $\Delta \cdot \omega_{ij} = \Delta \cdot \omega_i + \Delta \cdot \omega_j$ and P^{ij}_{dyn} is shown as follows:

$$\begin{aligned} P^{ij}_{\text{dyn}} &= \frac{e^{-\lambda\Delta\omega_j}(1-e^{-\lambda\Delta\omega_i})}{1-e^{-\lambda(\Delta\omega_i+\Delta\omega_j)}} \times P^i_{\text{dyn}} \\ &\quad + \frac{(1-e^{-\lambda\Delta\omega_j})}{1-e^{-\lambda(\Delta\omega_i+\Delta\omega_j)}} \times P^j_{\text{dyn}}. \end{aligned} \tag{11}$$

When Δ approaches 0, by L'Hospital's rule, we have

$$\begin{aligned} &\lim_{\Delta\to 0} \frac{e^{-\lambda\Delta\cdot\omega_j}(1-e^{-\lambda\Delta\cdot\omega_i})}{1-e^{-\lambda\Delta(\omega_i+\omega_j)}} = \frac{\omega_i}{\omega_i+\omega_j} \\ &\lim_{\Delta\to 0} \frac{(1-e^{-\lambda\Delta\cdot\omega_j})}{1-e^{-\lambda\Delta(\omega_i+\omega_j)}} = \frac{\omega_j}{\omega_i+\omega_j}, \\ &P^{ij}_{\text{dyn}} = \frac{\omega_i}{\omega_{ij}} \times P^i_{\text{dyn}} + \frac{\omega_j}{\omega_{ij}} \times P^j_{\text{dyn}}. \end{aligned} \tag{12}$$

Considering that the execution of τ_i and τ_j is periodic as a whole, period of τ_{ij} can be naturally constructed as the least common multiply of T_i and T_j, i.e. $T_{ij} = L_{ij} =$

LCM$\{T_i, T_j\}$, and C_{ij} can be also constructed as $(C_i + C_j)$. Then, considering that $\omega_{ij} = C_{ij}/s_{ij}T_{ij} = (\omega_i + \omega_j)$, we can calculate that $C_{ij}/s_{ij} = (\omega_i + \omega_j) \cdot T_{ij}$, and $s_{ij} = (C_i + C_j)/\omega_{ij}T_{ij}$. Now, τ_{ij} shown in Eq. 10 has been end-temperature-equivalently constructed. □

Corollary 2 *Consider task set* $\Gamma_n = \{\tau_1, \tau_2, ..., \tau_n\}$ *with each task* $\tau_\tau =< C_\tau/s_\tau, T_\tau, P_{\text{dyn}}^{\tau} >$, *then, with GPS scheduling, all of the adjacently executed n tasks can be end-temperature-equivalently constructed as one single task* $\tau_{12...n}$ *featured as*

$$\begin{aligned}
\tau_{12\cdots n} &= < C_{12\cdots n}/s_{12\cdots n}, T_{12\cdots n}, P_{\text{dyn}}^{12\cdots n} > \\
&= < L_{12\cdots n}\omega_{12\cdots n}, L_{12\cdots n}, \sum_{i=1}^{n} \frac{\omega_i}{\omega_{12\cdots n}} P_{\text{dyn}}^{i} >, \\
L_{12\cdots n} &= LCM\{T_{\Gamma_n}\},\ \omega_{12\cdots n} = \sum_{i=1}^{n} \omega_i, \qquad (13)
\end{aligned}$$

Proof This lemma can be easily proved through recursively leveraging Lemma 2. Hence, the detailed proof is omitted due to the space limit. □

Denote $\theta(mL)$ and $\theta_{\max}(mL)$ as the start and maximal temperatures respectively of the m_{th} heperperiod, where $L = \text{LCM}\{T_{\Gamma_n}\}$ and integer $m \geqslant 0$, and μ_0 as the computation utilization which is equal to $\sum_\Gamma \omega_\tau = \sum_\Gamma (C_\tau/T_\tau s_\tau)$. Now, we will deduce the expressions of $\theta(mL)$ and $\theta_{\max}(mL)$. First, we deduce $\theta(mL + k\Delta)$ and $\theta_{\max}(mL + k\Delta)$, where integer $k \geqslant 1$. In particular, we let $\theta_{\max}(0) = \theta(0)$ without loss of generality.

Theorem 1 *With GPS scheduling, during any hyperperiod of* $[mL, (m+1)L]$, $\theta(mL + k\Delta)$ *and* $\theta_{\max}(mL + k\Delta)$ *can be calculated as*

$$\begin{aligned}
\theta(mL + k\Delta) &= (1 - e^{-\lambda(L\cdot m + \Delta\cdot k)})\frac{\mu_0 P_{\text{dyn}}^{\Gamma}}{\lambda} \\
&\quad + e^{-\lambda(L\cdot m + \Delta\cdot k)}\theta(0), \\
\theta_{\max}(mL + k\Delta) &= (1 - e^{-\lambda\Delta\cdot(k+1)} + e^{-\lambda\Delta\cdot(k+\mu_0)} \\
&\quad - e^{-\lambda(L\cdot m + \Delta\cdot(k+\mu_0))})\frac{\mu_0 P_{\text{dyn}}^{\Gamma}}{\lambda} \\
&\quad + e^{-\lambda(L\cdot m + \Delta\cdot(k+\mu_0))}\theta(0). \qquad (14)
\end{aligned}$$

Proof Considering that the total utilization is μ_0, we can construct an idle periodic task $\tau_{\text{idle}} =< (1-\mu_0)L, L, 0 >$. Then, there are two tasks τ_Γ and τ_{idle} for GPS to schedule. Without loss of generality, we execute τ_Γ first, and the case of executing τ_{idle} first can be proved in a similar way. Then, the maximal temperature during each interval of $[mL + k\Delta, mL + (k+1)\Delta]$ is achieved at time moment $mL + (k+\mu_0)\Delta$, i.e. $\theta_{\max}(mL + k\Delta) = \theta(mL + (k+\mu_{)}\Delta)$. Here, a relatively lower initial temperature is considered in this paper, therefore, with GPS scheduling, the temperature keeps increasing when the system is active to execute tasks. In the kth interval of Δ in the mth hyperperiod, $\theta_{\max}(mL + k\Delta)$ and $\theta(mL + k\Delta)$ can be recursively calculated by Eq. 10 as follows:

$$\begin{aligned}
\theta(mL + k\Delta) &= \frac{(1 - e^{-\lambda\mu_0\Delta})e^{-\lambda(1-\mu_0)\Delta}(1 - e^{-\lambda\Delta\cdot k})}{1 - e^{-\lambda\Delta}} \\
&\quad \times \frac{P_{\text{dyn}}^{\Gamma}}{\lambda} + e^{-\lambda\Delta\cdot k}\theta(mL), \\
\theta_{\max}(mL + k\Delta) &= \theta(mL + (k+\mu_0)\Delta) \\
&= \frac{(1 - e^{-\lambda\mu_0\Delta})(1 - e^{-\lambda\Delta\cdot(k+1)})}{1 - e^{-\lambda\Delta}} \frac{P_{\text{dyn}}^{\Gamma}}{\lambda} \\
&\quad + e^{-\lambda\Delta\cdot(k+\mu_0)}\theta(mL), \qquad (15)
\end{aligned}$$

When Δ is infinitesimal, by L'Hospital's rule, we have

$$\begin{aligned}
&\lim_{\Delta\to 0} \frac{(1-e^{-\lambda\mu_0\Delta})e^{-\lambda(1-\mu_0)\Delta}(1-e^{-\lambda\Delta\cdot k})}{1-e^{-\lambda\Delta}} = \mu_0(1-e^{-\lambda\Delta\cdot k}), \\
&\lim_{\Delta\to 0} \frac{(1 - e^{-\lambda\mu_0\Delta})(1 - e^{-\lambda\Delta\cdot(k+1)})}{1 - e^{-\lambda\Delta}} = \mu_0(1 - e^{-\lambda\Delta\cdot(k+1)}), \\
&\theta(mL + k\Delta) = (1 - e^{-\lambda\Delta\cdot k})\frac{\mu_0 P_{\text{dyn}}^{\Gamma}}{\lambda} + e^{-\lambda\Delta\cdot k}\theta(mL), \\
&\theta_{\max}(mL + k\Delta) = (1 - e^{-\lambda\Delta\cdot(k+1)})\frac{\mu_0 P_{\text{dyn}}^{\Gamma}}{\lambda} \\
&\qquad + e^{-\lambda\Delta\cdot(k+\mu_0)}\theta(mL). \qquad (16)
\end{aligned}$$

Then, $\theta_{\max}(mL)$ and $\theta(mL)$ can be recursively deduced as follows:

$$\begin{aligned}
\theta_{\max}(mL) &= \theta(mL) \\
&= (1 - e^{-\lambda L\cdot m})\frac{\mu_0 P_{\text{dyn}}^{\Gamma}}{\lambda} + e^{-\lambda L\cdot m}\theta(0) \\
&= \frac{\mu_0 P_{\text{dyn}}^{\Gamma}}{\lambda} - \left(\frac{\mu_0 P_{\text{dyn}}^{\Gamma}}{\lambda} - \theta(0)\right) e^{-\lambda L\cdot m}, \qquad (17)
\end{aligned}$$

Combining Eqs. 16 and 17, $\theta(mL+k\Delta)$ and $\theta_{\max}(mL+k\Delta)$ can be deduced as shown in Eq. 14. □

In addition, we can get that $\theta(0) < 0$ during typical temperature range ($\theta(0) = -0.036$ when $\Theta_a = 25$ °C for state of the art processors [22]). Therefore, the following three critical conclusions can be summarized:

Conclusion 1: From Eq. 17, we can get that $\theta_{\max}(mL) \leqslant \theta_{\max}((m+1)L) < \mu_0 P_{\text{dyn}}^{\Gamma}/\lambda$ and $\theta(mL) \leqslant \theta((m+1)L) < \mu_0 P_{\text{dyn}}^{\Gamma}/\lambda$, i.e. both $\theta_{\max}(mL)$ and $\theta(mL)$ increase with m;
Conclusion 2: From Eq. 16, we can get that $\theta_{\max}(mL + k\Delta) \leqslant \theta_{\max}(mL+(k+1)\Delta)$ and $\theta(mL+k\Delta) \leqslant \theta(mL+(k+1)\Delta)$, i.e. both $\theta_{\max}(mL + k\Delta)$ and $\theta(mL + k\Delta)$ increase with m and k;

Conclusion 3: From Eq. 17, we can get that both $\theta_{\max}(mL)$ and $\theta(mL)$ achieve the lowest value when $\mu_0 P^{\Gamma}_{\text{dyn}}$ is minimized.

Furthermore, we can conclude our two new theorems:

Theorem 2 *With GPS scheduling, the system will finally reach a thermal steady state, and the temperature during this state is a constant like* $\theta_{ss} = \mu_0 P^{\Gamma}_{\text{dyn}}/\lambda = \sum_{\Gamma}(\omega_\tau P^{\tau}_{\text{dyn}})/\lambda$.

Proof By Corollary 2, we have

$$\begin{aligned}P^{\Gamma}_{\text{dyn}} &= \sum_{i=1}^{n} \frac{\omega_i}{\omega_{12\cdots n}} P^{i}_{\text{dyn}} \\ &= \left(1/\sum_{i=1}^{n}\omega_i\right)\sum_{i=1}^{n}\omega_i P^{i}_{\text{dyn}} = \frac{1}{\mu_0}\sum_{i=1}^{n}\omega_i P^{i}_{\text{dyn}}. \end{aligned} \tag{18}$$

By Theorem 1, when m is large enough, $\theta_{\max}((m+1)L)$ and $\theta((m+1)L)$ can be approximated as

$$\begin{aligned}&\lim_{m\to\infty}\theta_{\max}((m+1)L) = \lim_{m\to\infty}\theta((m+1)L) \\ &= \lim_{m\to\infty}\theta(mL) = \mu_0 P^{\Gamma}_{\text{dyn}}/\lambda = \sum_{i=1}^{n}\omega_i P^{i}_{\text{dyn}}/\lambda. \end{aligned} \tag{19}$$

□

Theorem 3 *With GPS scheduling, no matter the system is in a steady state or in a transient state,* $\theta_{\max}(mL + k\Delta)$ *and* $\theta(mL + k\Delta)$ *always achieve their minimal values when the speeds of tasks are assigned as:*

$$s_i = A_i{}^{-1/3}\sum\nolimits_{1}^{n} A_i{}^{1/3} \cdot C_i/T_i. \tag{20}$$

Proof According to Theorem 1, both $\theta_{\max}(mL + k\Delta)$ and $\theta(mL + k\Delta)$ increases with $\mu_0 P^{\Gamma}_{\text{dyn}}/\lambda$. Therefore, the problem of minimizing $\theta_{\max}(mL + k\Delta)$ and $\theta(mL + k\Delta)$ can be transformed as minimizing the value of $\mu_0 P^{\Gamma}_{\text{dyn}}/\lambda$. Considering that $\mu_0 P^{\Gamma}_{\text{dyn}} = \sum_{\tau_i\in\Gamma}(\omega_i P^{i}_{\text{dyn}}) = \sum_{\tau_i\in\Gamma}(A_i s_i^2 \cdot C_i/T_i)$ and the constrained condition of this optimization problem is $\sum_{\tau_i\in\Gamma}(\omega_i) = \sum_{\tau_i\in\Gamma} C_i/T_i s_i = \mu_0 \leqslant 1$, by the Lagrange Multiplier method and the famous KKT (Karush-Kuhn-Tucker) Condition, we can simply get the minimal condition as Eq. 20 through the following deduction:

$$\begin{aligned}&g(s_1, s_2, \cdots, s_n) = \sum\nolimits_{\tau_i\in\Gamma} C_i/T_i s_i - 1; \\ &F(s_1, s_2, \cdots, s_n, \eta) = \sum\nolimits_{\tau_i\in\Gamma}(A_i s_i^2 \cdot C_i/T_i) \\ &\qquad\qquad + \eta g(s_1, s_2, \cdots, s_n); \\ &\frac{\partial F}{\partial s_i} = 0, \forall\tau_i \in \Gamma;\ \frac{\partial F}{\partial \eta} = 0. \end{aligned} \tag{21}$$

□

4.3 Generalization of Minimal Temperature Condition

Actually, the minimal temperature condition we deduced with GPS scheduling (Theorem 3) can also be generalized with any other periodic scheduling algorithms as long as they can guarantee the task deadlines under the minimal condition, i.e. any periodic scheduling algorithms that can fully utilize the processor, such as EDF and LLF, could find a possible schedule to satisfy the minimal condition. We will prove this generalization in this section.

Consider periodic task set $\Gamma_n = \{\tau_1, \tau_2, ..., \tau_n\}$ with each task $\tau_\tau = < C_\tau/s_\tau, T_\tau, P^{\tau}_{\text{dyn}} >$. Denote the hyperperiod of Γ_n as L (L is equal to the least common multiple of the periods of all tasks in Γ_n). Let $\Pi(t)$ be the cyclic schedule of Γ_n, then $\Pi(t) = \Pi(t + mL)$ where $m = 0, 1, 2, ...$ $\Pi(t)$ identifies one task in Γ_n executing at time t with speed s, or $\emptyset$ if the system is idle at this moment.

By Corollary 2, whatever the scheduling algorithm is, all of the preemptible instances of tasks in Γ_n, including possible idle tasks during one hyperperiod, can be end-temperature-equivalently constructed as one single task $\tau_s = < C_s/s_s, T_s, P^{s}_{\text{dyn}} >$. When the schedule is cyclic, τ_s keeps the same during any hyperperiod, where $C_s/s_s = T_s = L$ and P^{s}_{dyn} is uniquely determined by the schedule $\Pi(t)$ which is related with the scheduling algorithm and the speed assignment. The expression of P^{s}_{dyn} is too complex to show, but it can be computed recursively according to Eq. 7.

Let $\theta^{\Pi}(mL + t)$ be the processor temperature during the periodic schedule $\Pi(t)$ in the mth hyperperiod, $\theta^{\Pi}_{\max}(mL)$ and $\theta^{\Pi}_{\text{e}}(mL)$ be the corresponding maximal and end temperatures separately. Let $\theta^{s}(mL + t)$ be the processor temperatures during the execution of the single constructed task τ_s in the mth hyperperiod, $\theta^{s}_{\max}(mL)$ and $\theta^{s}_{\text{e}}(mL)$ be the corresponding maximal and end temperatures separately.

Lemma 3 $\theta^{s}_{\max}(mL) = \theta^{s}_{\text{e}}(mL)$.

Proof $\theta^{s}(mL + t)$ can be deduced as:

$$\begin{aligned}\theta^{s}(mL+t) &= \frac{P^{s}_{\text{dyn}}}{\lambda} - \left(\frac{P^{s}_{\text{dyn}}}{\lambda} - \theta(0)\right)e^{-\lambda(mL+t)}, \\ &\quad 0 \leqslant t \leqslant L. \end{aligned} \tag{22}$$

Considering that $\theta(0) < 0$ during typical temperature range, we can get that $\theta^{s}(mL + t)$ increases with t. Hence the lemma. □

Lemma 4 $\theta^{\Pi}(mL+t) = \theta^{s}(mL+t)$ *under condition* (20).

Proof Under condition (20), we can get that $\mathrm{U}_{\mathrm{tot}} = 1$, $P^i_{\mathrm{dyn}} = P^j_{\mathrm{dyn}} = P_{\mathrm{dyn}}$ for arbitrary i and j and $P^s_{\mathrm{dyn}} = P_{\mathrm{dyn}}$. It is easy to conclude that $\theta^{\Pi}(mL+t)$ and $\theta^s(mL+t)$ keep the same during the hyperperiod. □

Lemma 5 *$\theta^{\Pi}_{\max}(mL) \geqslant \theta^s_{\max}(mL)$, and the equality holds under condition* (20).

Proof Since τ_s is end-temperature-equivalently constructed for Γ_n, $\theta^{\Pi}_{\mathrm{e}}(mL) = \theta^s_{\mathrm{e}}(mL)$. Therefore, $\theta^{\Pi}_{\max}(mL) \geqslant \theta^{\Pi}_{\mathrm{e}}(mL) = \theta^s_{\max}(mL)$. Furthermore, by Lemma 4, $\theta^{\Pi}_{\max}(mL) = \theta^{\Pi}_{\mathrm{e}}(mL)$ under condition (20). Hence, the lemma. □

Theorem 4 *$\theta^s_{\max}(mL)$ is minimized when s_i is assigned as* Eq. 20.

Proof According to the research work by Ahmed et al. [22, 23], when the system is in a thermal steady state, the average temperature within the hyperperiod θ_{avg} is minimized when the speeds of tasks are assigned according to Eq. 20. With condition (20), by Lemmas 4 and 5, $\theta^{\Pi}(mL+t) = \theta^s(mL+t) = \theta^s_{\max}(mL) = P^s_{\mathrm{dyn}}/\lambda$ when m is infinite. Thus, we can say that $P^s_{\mathrm{dyn}}/\lambda$ is minimized with condition (20). By Lemma 3, during any hyperperiod, $\theta^s_{\max}(mL) = \theta^s_{\mathrm{e}}(mL)$, which increases with $P^s_{\mathrm{dyn}}/\lambda$ according to Eq. 22. Therefore, $\theta^s_{\max}(mL)$ is minimized with condition (20). □

Theorem 5 *$\theta^{\Pi}_{\max}(mL)$ is minimized if s_i is assigned according to* Eq. 20. *That is to say, the maximal temperature is minimized during any hyperperiod with any periodic schedule if s_i is assigned according to* Eq. 20.

Proof With condition (20), by Lemma 5, we can get that $\theta^{\Pi}_{\max}(mL) = \theta^s_{\max}(mL)$. By Theorem 4, $\theta^s_{\max}(mL)$ is minimized with condition (20). Then, we can say that $\theta^{\Pi}_{\max}(mL)$ is minimized if s_i is assigned according to Eq. 20. □

4.4 Feasible Solution Analysis of Minimal Condition

Through analysis, we find that the necessary condition to find a feasible solution of minimal condition (20) is as follows:

$$\mathrm{U}_{\mathrm{tot}} = \sum\nolimits_i C_i/T_i \in [S_{\min}, 1]; \quad A_{\min} \geqslant {S_{\min}}^3 \cdot A_{\max}. \tag{23}$$

where $\mathrm{U}_{\mathrm{tot}}$ here is the original total utilization of the system, $A_{\min}$ is the minimal switching activity value while $A_{\max}$ is the maximal, $S_{\min}$ is the normalized minimal speed.

Proof As all we know, $\mathrm{U}_{\mathrm{tot}} \leqslant 1$, so we only prove it by contradiction when $\mathrm{U}_{\mathrm{tot}} < S_{\min}$ or $A_{\min} < S^3_{\min} \cdot A_{\max}$, under which,

$$\sum\nolimits_i \frac{C_i}{T_i s_i} \leqslant \frac{1}{S_{\min}} \sum\nolimits_i \frac{C_i}{T_i} < 1,$$
$$\text{or } s_{\min} > \frac{s_{\max}}{S_{\min}} \geqslant 1. \tag{24}$$

Obviously, condition (20) has no feasible solution if Eq. 23 is not satisfied. □

5 Thermal-Aware Scheduling Algorithm

This section proposes a thermal-aware scheduling algorithm with a novel priority assignment approach based on the optimal speed condition deduced in Section 4 and the thermal law we find to deal with the situation when optimal speeds cannot be achieved due to reasons like additional performance constraints, deadline constraints, system speeds constraints (eg., [0.625, 1.0]) and its discreteness, or resources reservation for other non-real-time tasks.

When the optimal speeds can be achieved, it can be easily proved that all of the scheduling algorithms like GPS and EDF result in the same minimal temperature traces (see Theorem 3 and Theorem 5) since all of the tasks share the same power consumption value and the processor is fully utilized (see Eq. 25); when not, GPS scheduling results in lower temperature than other scheduling algorithms due to its ability to eliminate the oscillating feature of the processor energy and temperature. However, GPS scheduling causes much task switching overhead due to its execution pattern. To mitigate this problem, an approximate GPS scheduling algorithm with a longer scheduling length is proposed, in which the executing sequence of tasks in a scheduling interval matters. The thermal law shown below helps to determine the thermal-aware task executing sequence.

$$P^i_{\mathrm{dyn}} = A_i {s_i}^3 = \left(\sum\nolimits_1^n \left(A_i^{1/3} C_i/T_i\right)\right)^3 \forall \tau_i \in \Gamma,$$
$$U_{\mathrm{tot}} = \sum\nolimits_1^n (C_i/T_i s_i) = 1. \tag{25}$$

5.1 Determination of the Task Executing Sequence

This section presents the thermal law exhibited by tasks' power consumption we find, which can be used to determine the thermal-aware task executing sequence.

In a similar way, we describe the thermal law first for two tasks then for multi-tasks. Scenarios I and II are used respectively.

Lemma 6 (Thermal Law) *When $P_{\mathrm{dyn}}^{i} \geqslant P_{\mathrm{dyn}}^{j}$, $\theta_{\mathrm{e}}^{i,j}|_{\theta_0} \leqslant \theta_{\mathrm{e}}^{j,i}|_{\theta_0}$.*

Proof By Eq. 6, we have

$$\theta_{\mathrm{e}}^{i,j}|_{\theta_0} - \theta_{\mathrm{e}}^{j,i}|_{\theta_0} = \left(\frac{P_{\mathrm{dyn}}^{j} - P_{\mathrm{dyn}}^{i}}{\lambda}\right)(1-e^{-\lambda\Delta_i})(1-e^{-\lambda\Delta_j}). \tag{26}$$

Obviously, the second and third multipliers in the above equation are greater than zero. In addition, λ can be proved a positive number in practice (e.g. $\lambda = 3.471$ for state of the art processors where $R_{\mathrm{th}} = 0.36$, $C_{\mathrm{th}} = 0.8$, $\alpha = 0.001$ and $\beta = 0.1$ [22]). Hence, the lemma. □

Corollary 3 (Thermal Law) *When $P_{\mathrm{dyn}}^{1} \geqslant P_{\mathrm{dyn}}^{2} \geqslant \ldots \geqslant P_{\mathrm{dyn}}^{n}$, $\theta_{\mathrm{e}}^{1,2,\ldots,n}|_{\theta_0}$ is no greater than any other $\theta_{\mathrm{e}}^{x_1,x_2,\ldots,x_n}|_{\theta_0}$.*

Proof Let $< y_1, y_2 >$ be the permutation of $< x_1, x_2 >$ satisfying that $P_{\mathrm{dyn}}^{y_1} \geqslant P_{\mathrm{dyn}}^{y_2}$, then by Lemma 6, $\theta_{\mathrm{e}}^{y_1,y_2}|_{\theta_0} \leqslant \theta_{\mathrm{e}}^{x_1,x_2}|_{\theta_0}$. Thus, by Eq. 6, we have

$$\begin{aligned}&\theta_{\mathrm{e}}^{y_1,y_2,\cdots,x_n}|_{\theta_0} = \theta_{\mathrm{e}}^{x_3,x_4,\cdots,x_n}|_{\theta_{\mathrm{e}}^{y_1,y_2}|_{\theta_0}}\\ &\leqslant \theta_{\mathrm{e}}^{x_3,x_4,\cdots,x_n}|_{\theta_{\mathrm{e}}^{x_1,x_2}|_{\theta_0}} = \theta_{\mathrm{e}}^{x_1,x_2,\cdots,x_n}|_{\theta_0}.\end{aligned} \tag{27}$$

By analogy, let $< y_1, y_2, \ldots, y_k >$ be the permutation of $< x_1, x_2, \ldots, x_k >$ satisfying that $P_{\mathrm{dyn}}^{y_1} \geqslant P_{\mathrm{dyn}}^{y_2} \geqslant \ldots \geqslant P_{\mathrm{dyn}}^{y_k}$, and suppose $\theta_{\mathrm{e}}^{y_1,y_2,\ldots,y_k,x_{k+1},\ldots,x_n}|_{\theta_0} \leqslant \theta_{\mathrm{e}}^{x_1,x_2,\ldots,x_n}|_{\theta_0}$, then if (a) $P_{\mathrm{dyn}}^{y_k} \geqslant P_{\mathrm{dyn}}^{x_{k+1}}$, we have

$$\begin{aligned}&\theta_{\mathrm{e}}^{y_1,\cdots,y_k,x_{k+1},\cdots,x_n}|_{\theta_0} = \theta_{\mathrm{e}}^{x_{k+1},\cdots,x_n}|_{\theta_{\mathrm{e}}^{y_1,y_2,\cdots,y_k}|_{\theta_0}}\\ &\leqslant \theta_{\mathrm{e}}^{x_{k+1},\cdots,x_n}|_{\theta_{\mathrm{e}}^{x_1,x_2,\cdots,x_k}|_{\theta_0}} = \theta_{\mathrm{e}}^{x_1,x_2,\cdots,x_n}|_{\theta_0};\end{aligned} \tag{28}$$

If (b) $P_{\mathrm{dyn}}^{y_j} \geqslant P_{\mathrm{dyn}}^{x_{k+1}} \geqslant P_{\mathrm{dyn}}^{y_{j+1}}(j \leqslant k-1, k \leqslant n-1$, $P_{\mathrm{dyn}}^{0} = \infty$, $P_{\mathrm{dyn}}^{n+1} = 0)$, by Corollary 3, $\tau_{y_{j+1}\ldots y_k}$ can be end-temperature-equivalently constructed for $\tau_{y_{j+1}}, \ldots, \tau_{y_k}$ and $P_{\mathrm{dyn}}^{x_{k+1}} \geqslant P_{\mathrm{dyn}}^{y_{j+1}} \geqslant P_{\mathrm{dyn}}^{y_{j+1}\ldots y_k} \geqslant P_{\mathrm{dyn}}^{y_k}$; then, by Lemma 6, $\theta_{\mathrm{e}}^{x_{k+1},y_1 y_2\ldots y_{j+1}}|_{\theta_0} \leqslant \theta_{\mathrm{e}}^{y_1 y_2\ldots y_{j+1},x_{k+1}}|_{\theta_0}$; finally, by Eq. 6, we have

$$\begin{aligned}&\theta_{\mathrm{e}}^{y_1,\cdots,y_j,x_{k+1},y_{j+1},\cdots,y_k,x_{k+2},\cdots,x_n}|_{\theta_0}\\ &= \left(\theta_{\mathrm{e}}^{x_{k+2},\cdots,x_n}|_{\theta_{\mathrm{e}}^{x_{k+1},y_{j+1}\cdots y_k}|_{\theta_{\mathrm{e}}^{y_1,y_2,\cdots,y_j}|_{\theta_0}}}\right)\\ &\leqslant \left(\theta_{\mathrm{e}}^{x_{k+2},\cdots,x_n}|_{\theta_{\mathrm{e}}^{y_{j+1}\cdots y_k,x_{k+1}}|_{\theta_{\mathrm{e}}^{y_1,y_2,\cdots,y_j}|_{\theta_0}}}\right)\\ &\leqslant \left(\theta_{\mathrm{e}}^{x_{k+2},\cdots,x_n}|_{\theta_{\mathrm{e}}^{x_{j+1},\cdots,x_k,x_{k+1}}|_{\theta_{\mathrm{e}}^{x_1,x_2,\cdots,x_j}|_{\theta_0}}}\right)\\ &= \theta_{\mathrm{e}}^{x_1 x_2\cdots x_n}|_{\theta_0}.\end{aligned} \tag{29}$$

Combination of (a) and (b) prove that when the first k tasks satisfy Corollary 3, then the first $(k+1)$ tasks also satisfy it. When $k = n-1$, it can be proved that $\theta_{\mathrm{e}}^{1,2,\ldots,n}|_{\theta_0}$ is no greater than any other $\theta_{\mathrm{e}}^{x_1,x_2,\ldots,x_n}|_{\theta_0}$. □

Now, consider a special case when $P_{\mathrm{dyn}} = 0$. Idle time interval t_{idle} can be constructed as a task τ_{idle} with $P_{\mathrm{dyn}}^{\mathrm{idle}} = 0$, and it's trivial to prove that it also satisfies Corollary 3.

What deserves to be noticed here is that Corollary 3 explores the minimal end temperature condition rather than the peak temperature studied by Jayaseelan et al. [13]. However, compared with the method of qualitatively adopting cool-hot task pair execution [13], Corollary 3 quantitatively minimizes the end temperature, which can in turn decrease the temperature in the next time interval due to its increasing property with the initial temperature. In particular, when the task sequence determined based on the thermal law is applied in each scheduling interval of the GPS scheduling, temperature can be better optimized due to such virtuous circle in each scheduling interval with small enough length.

5.2 Thermal-Aware Scheduling Algorithm

Based on the task construction method proposed in Section 4.1, we deduce the optimal speed condition (see Theorem 3) for GPS scheduling and generalize it to other scheduling methods (see Theorem 5). Based on the thermal law proposed in Section 5.1, we address the scenarios when the optimal speed condition cannot be achieved due to reasons like the tasks' timing constraints or the processor speeds limitation. Based on the optimal speed condition and thermal law proposed above, this section implements a thermal-aware task scheduling algorithm while taking the case when optimal speed condition cannot be achieved into consideration.

Existing scheduling algorithms, such as RM, EDF, LSF, etc., generally take time as the metric to determine the executing priorities of tasks, while temperature is barely considered. According to the thermal law presented above, we have sufficient reasons to believe that the key to further minimize temperature is to consider it as the metric to determine the task executing sequence. Then, how do we guarantee the timing constraints? GPS scheduling just provides a good guarantee, by which, all task instances will finish exactly at their absolute deadlines. However, GPS scheduling demands that the system should be fluid which cannot be always satisfied. Further, with GPS scheduling, the system may have to switch speed too frequently thus causing extraordinary overhead. Given the above reasons, based on the thermal law and minimal condition presented in this paper, we design and implement a thermal-aware scheduling algorithm with the power-related temperature as the metric to determine the task priorities (see Algorithm 1). It is an approximation of GPS and can be applied in a system of harmonic tasks. The maximal scheduling interval length can be set as the minimal period of the tasks.

Algorithm 1 GPS-TA (Approximate GPS scheduling with Thermal-Aware priority assignment)

```
   Input: A task set $\Gamma_n = \{\tau_1, \tau_2, ..., \tau_n\}$ with each task
          $\tau_\tau =< C_\tau, T_\tau, A_\tau >$, $U_{tot}$ and thermal related
          parameters shown in Table 1.
   Output: The processor temperature values during the
           execution of the tasks.
1  Set tasks' speeds according to the optimal speed
   condition (see Theorem 3, Eq. 20);
2  Adjust the tasks' speeds according to the constraints
   like tasks' deadlines and processor's speed range;
3  Set the tasks' priorities according to their dynamic
   power consumption based on the thermal law proposed
   in Section 5.1;
4  repeat
5      $releaseTasks()$;
6      repeat
7          $runningTask = getNextTask()$;
8          $runTask()$; //run tasks for $IL \cdot \omega_\tau (C_\tau / T_\tau s_\tau)$
           time units;
9          $updateTemperature()$;
10         if $runningTask.finish()$ then
11             $removeTask(runningTask)$;
12         else
13             $runningTask.setVisibility(False)$;
14         end
15     until $\Delta t < IL$;
16     $resetTasks()$;//set all tasks visible;
17 until $k \cdot IL < TI$;
```

Algorithm 1 can be divided into the following four parts. The first part (line 1) is responsible to set the tasks' speeds based on the optimal speed condition deduced in Theorem 3. With this speed assignment, the processor's temperature and energy consumption can be minimized. The second part (line 2) is responsible to address the case when the optimal speed condition cannot be achieved due to reasons like tasks' timing constraints and processor's speed range. In specific, when the optimal task speed is lower than the minimal speed that meets its deadline, it will be adjusted as the minimal speed; when the optimal task speed is larger than the maximal processor speed, say 1, it will be adjusted as the maximal processor speed. Once the speed of one task is adjusted, the speeds of the remained tasks also need to be adjusted so as to get the best sub-optimal solution under the constraint of schedulability condition. This procedure will be recursively executed until all of the tasks' speeds satisfy the constraints. The third part (line 3) is responsible to set the tasks' priorities according to their dynamic power consumption based on the thermal law proposed in Section 5.1, i.e. a task with a higher dynamic power consumption will be assigned a higher priority. The fourth part (lines 4–17) is responsible to simulate the task scheduling process of our proposed approximate GPS with a fixed scheduling length. Here, IL refers to the scheduling interval length to approximate GPS, and TI is the time interval we set to observe the processor temperature and power consumption. The critical step of GPS-TA is that during each IL, tasks will be executed in a decreasing order according to their dynamic power consumption. About IL, the smaller it is, the closer the approximation is to GPS, and the lower the temperature is; the longer it is, the less the task switching overheads are. Therefore, a better tradeoff should be cautiously made to balance the temperature and switching overheads. Usually, to meet timing constraints, IL should be no larger than the greatest common divisor of tasks' periods ($IL \leqslant \gcd\{T_i\}$), and particularly IL can be set as large as the minimal task period when it comes to harmonic real-time tasks. Hereafter, we call GPS-TA as TA if no confusions can be made.

6 Experiments

In this section, we design four groups of experiments to validate the approaches proposed in this work. The first two groups of experiments are designed to validate the correctness of the proposed task construction method and the optimal conditions for GPS and EDF scheduling separately. The third experiment is designed to validate the efficiency of the thermal law we proposed; while the last experiment is designed to compare our proposed approximate GPS scheduling, which is implemented based on the optimal condition and the thermal law we propose, with the state-of-the-art approach SD-RM presented by Zhou et al. [19] (we call this approach SD-RM because it adopts slack distribution to minimize the peak temperature and is implemented based on RM scheduling).

The system considered in this work is assumed DVFS capable with normalized speed varying between [0.625, 1.0] for the state of the art processors. System thermal parameters of the processor are set as: $R_{th} = 0.36$, $C_{th} = 0.8$, $\alpha = 0.001$, $\beta = 0.1$, $\Theta_a = 25$ °C [22]. Experiments are conducted for hard periodic real-time tasks with varying total utilization ($U_{tot} = 0.75, 0.85, 0.95$) and variance of switching activity ($\delta_{Acti} = 0.1, 0.2, 0.3$). For all of the three sets of experiments, we separately generate three kinds of test cases: S_{opt} refers to the optimal speeds under given parameters; $S_1, S_2, ..., S_6$ refer to the sub-optimal speeds randomly generated with conditions of $U_{Scale} = 1.0$ (the total utilization after speeds scaling), and they are ordered increasingly according to the variance of tasks' dynamic power consumption determined by them; $S_{\max}$ refers to the speeds of 1.0 without any optimization. Different task parameters result in different speed assignments $S_{\text{opt}}, S_1, S_2, ..., S_6$. In addition, since the six sub-optimal speed assignments result in similar temperature traces, for the sake of visibility, we only choose three of them (S_1, S_2, S_6) as representatives to show their inferiority to the optimal speed assignment S_{opt}.

6.1 Optimal Speeds Verification of GPS Scheduling

In the first set of experiments, we set tasks' periods in 0.01 second grade, and validate the optimal condition for GPS scheduling proposed in this research. The interval length of GPS is 10 ms, and the system achieves steady state in its fifth hyperperiod. We conduct 9 groups of experiment for all the parameters. All of the results show the same regularities that are exactly consistent with the theorems and conclusions proved for GPS in Section 4.2. As representatives, results of $\text{U}_{\text{tot}} = 0.75$ are shown in Fig. 1.

Figure 1 shows that, with GPS scheduling, temperature keeps increasing all the time and the maximal temperature is achieved at the end of each hyperperiod ($400 \cdot k$) under all of the parameter setups (different U_{tot}s and different δ_{Acti}s). In addition, S_{opt} results in the lowest temperature while $S_{\max}$ results in the highest temperature for all parameters. About $S_1, S_2, ..., S_6$, they result in the medium temperature compared with S_{opt} and $S_{\max}$. First, we guess that temperature increases with the variance of tasks' dynamic power consumption, but through the experimental results, we can see that this guess is not exactly right: the temperatures with $S_1, S_2, ..., S_6$ do not always result in an increasing order (you can see it more clearly in Fig. 2). This means that, the relationship between temperature and δ_{Acti} is much more complex in practice than we thought. This phenomenon deserves to be studied in our future work.

Next, we analyze the relationship between the maximal temperature and the constructed dynamic power consumption (P^s_{dyn}/λ) for each set of speeds. Here, the maximal temperature is sampled with the same length as IL. As representatives, results of $\delta_{\text{Acti}} = 0.3$ are shown in Fig. 2.

Figure 2 shows that with GPS scheduling, the maximal temperature has a strong positive correlation with P^s_{dyn}/λ for all parameter setups, i.e. maximal temperature is minimized when P^s_{dyn}/λ is minimized. In the meantime, with the increasing of U_{tot}, temperature and P^s_{dyn}/λ both increase while the benefit of S_{opt} decreases. The results shown in Figs. 1 and 2 validate the conclusions proposed for GPS in Section 4.2.

6.2 Optimal Speeds Verification with EDF Scheduling

In the second set of experiments, we set tasks' periods in 0.1 second grade, and validate the optimal condition for EDF scheduling proposed in this research. The system achieves steady state in its second hyperperiod. The reason we choose 0.1-second-grade periods rather than 0.01 is that the temperature oscillating features of EDF are more obvious in this situation. We conduct 9 groups of experiment for all the parameters. All of the results show the same regularities that are exactly consistent with the theorems and conclusions proved for EDF in Section 4.3. As a representative, results of 0.75 are shown in Fig. 3.

Figure 3 shows that although temperature shows oscillating feature under EDF scheduling, it keeps increasing in general for all of the parameter setups. Similarly, S_{opt} results in the lowest temperature, $S_{\max}$ results in the highest temperature, while $S_1, S_2, ..., S_6$ result in the medium temperature for all parameter setups compared with S_{opt} and $S_{\max}$. In addition, we can also see that with optimal speeds S_{opt}, EDF has no difference with GPS, also resulting in the same smoothing temperature curve. However, there exist exceptions which can be seen from Fig. 3c. This is caused by the speed range limitation: the optimal speeds have to be in [0.625, 1.0], otherwise they need to be modified to sub-optimal values, meaning that the dynamic power

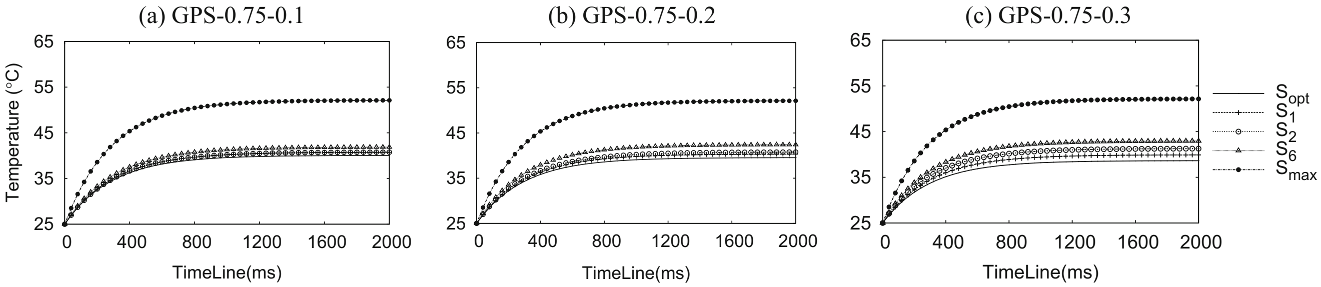


Figure 1 Temperature variation trends with GPS scheduling.

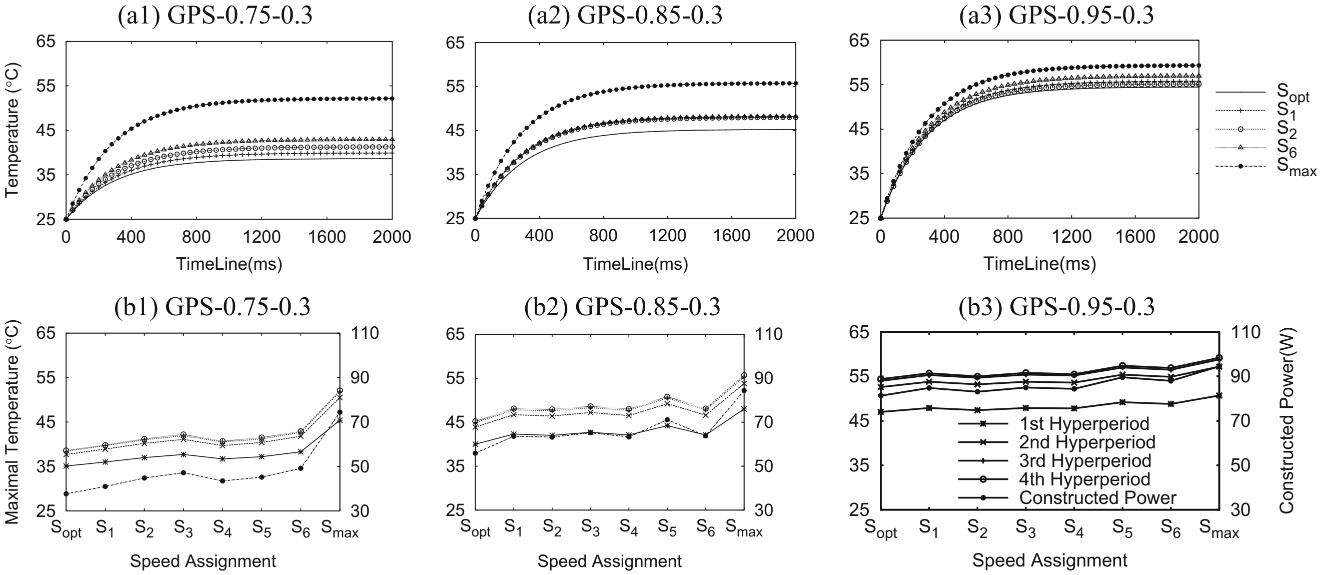


Figure 2 Optimal speeds verification with GPS scheduling.

consumptions of the real-time periodic tasks are not the same any more. Furthermore, this set of experiments also show that the relationship between temperature and δ_{Acti} is much more complex than we thought.

Next, by the same way we analyze the relationship between the maximal temperature and the constructed dynamic power consumption ($P^{s}_{\mathrm{dyn}}/\lambda$) for each set of speeds. As representatives, results of $\delta_{\mathrm{Acti}} = 0.3$ are shown in Fig. 4. Figure 4 shows that with EDF scheduling, the maximal temperature has a strong positive correlation with $P^{s}_{\mathrm{dyn}}/\lambda$ for all parameter setups, i.e. maximal temperature is minimized when $P^{s}_{\mathrm{dyn}}/\lambda$ is minimized. In the meantime, with the increasing of $\mathrm{U_{tot}}$, temperature and $P^{s}_{\mathrm{dyn}}/\lambda$ both increase while the benefit of S_{opt} decreases. This conclusion is consistent with that proposed for EDF in Section 4.3.

6.3 Efficiency Analysis of Thermal-Aware Algorithm

In Section 5, we have denoted that although the execution sequence has no effect on temperature with optimal condition, we usually cannot easily achieve the optimal speeds in practice due to many reasons, such as additional performance constraints, deadline constraints, or the system speed constraints. In this case, the executing sequence of tasks determined by the thermal law we propose shows better thermal feature. To validate the efficiency of the thermal-aware priority assignment algorithm (Algorithm 1, denoted as TA for short, in opposition to non-thermal-aware scheduling algorithm NTA) proposed in this paper, the three sets of task speed assignments are used to scheduling the tasks. We set tasks' periods in second grade, and study the efficiency of TA under interval lengths with different orders of magnitude. Here, we only show part of the results as an example.

Figure 5 shows that TA is more efficient than NTA in peak temperature minimization for all parameter setups, which is consistent with the thermal law we find in our research. The longer the interval length is, the larger the benefit of TA; the larger the variance of tasks' dynamic power consumption is, the larger the benefit of TA. That is

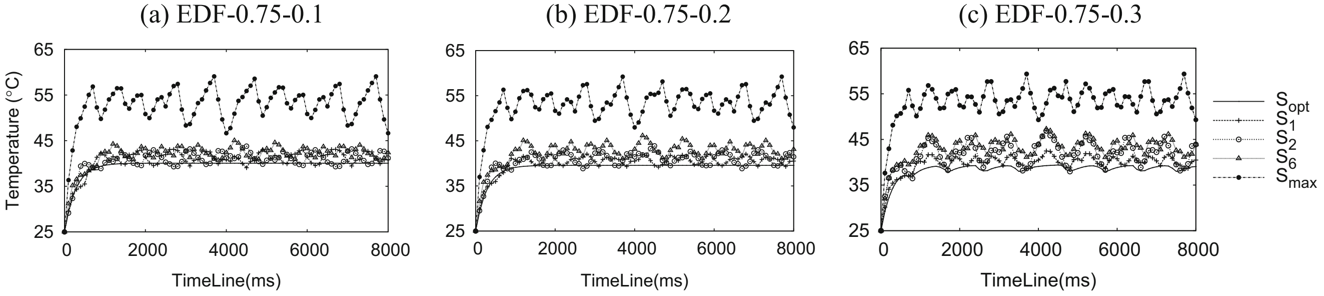


Figure 3 Temperature variation trends with EDF scheduling.

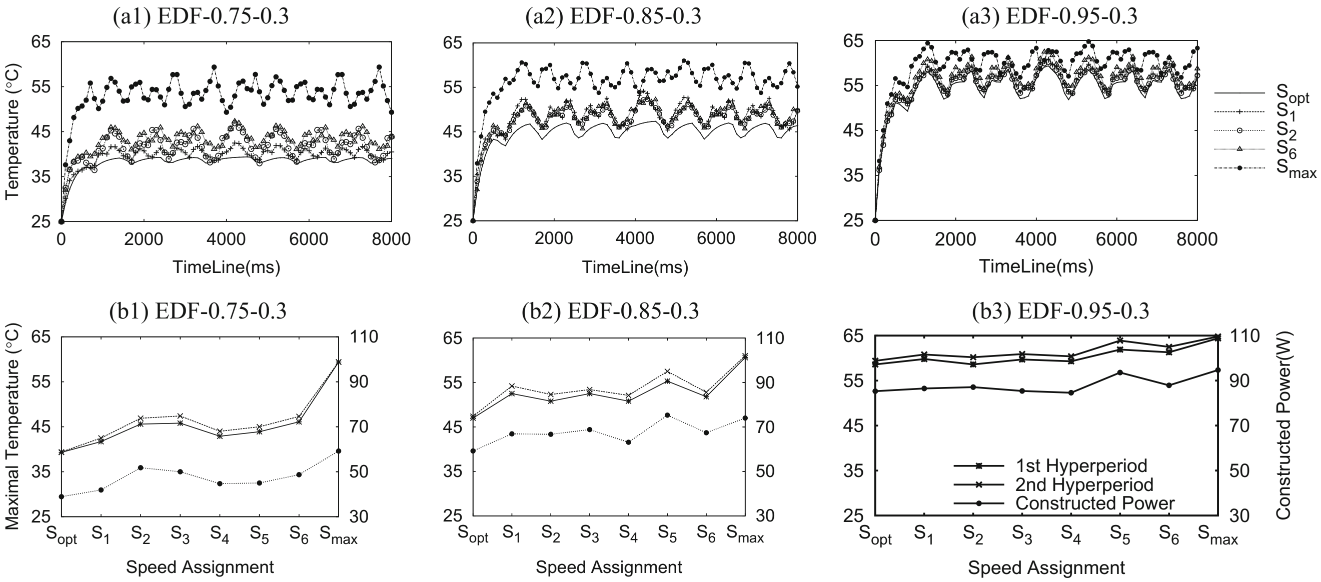


Figure 4 Optimal speeds verification with EDF scheduling.

too say, TA is applicable for longer interval length and larger variance of tasks' dynamic power consumption.

6.4 Comparison with the state-of-the-art Approaches

This group of experiment is designed to validate the efficiency on both temperature and energy consumption minimization of our proposed approximate GPS scheduling which is implemented based on the optimal speed condition (see Theorem 3 in Section 4.2) and the thermal law (see Corollary 3 in Section 5.1) we proposed. We compare it with the state-of-the-art approach SD-RM presented in [19]. We conduct this experiment under three setups of system utilization. Experimental results are shown in Fig. 6.

Figure 6 shows that, compared with the state-of-the-art approach (SD-RM), our proposed approximate GPS scheduling (denoted as GPS in the figure without confusion) can significantly decrease both temperature and energy consumption under all of the experimental setups. Through analysis, we find that the deficiency of SD-RM lies in the following two aspects: (1) SD-RM does not conduct speed scaling, which will result in higher dynamic power consumption, thus causing higher temperature compared with our approach; (2) Although SD-RM minimizes peak temperature by slack distribution, its efficiency is largely limited by the schedulability constraint of RM scheduling. To be more specific, temperature is affected by both the dynamic power and the leakage power, so only using slack distribution to minimize temperature is not sufficient; for RM scheduling, tasks with lower periods will be executed first, which may be not in accordance with the goal of balancing the thermal profiles of tasks, thus weakening its effect on temperature minimization. In fact, in our experiments, the slacks assigned to tasks are usually much small than their thermal optimal slacks due to the tasks' timing constraints in RM scheduling. On the contrary, our proposed approach on the one hand can make the

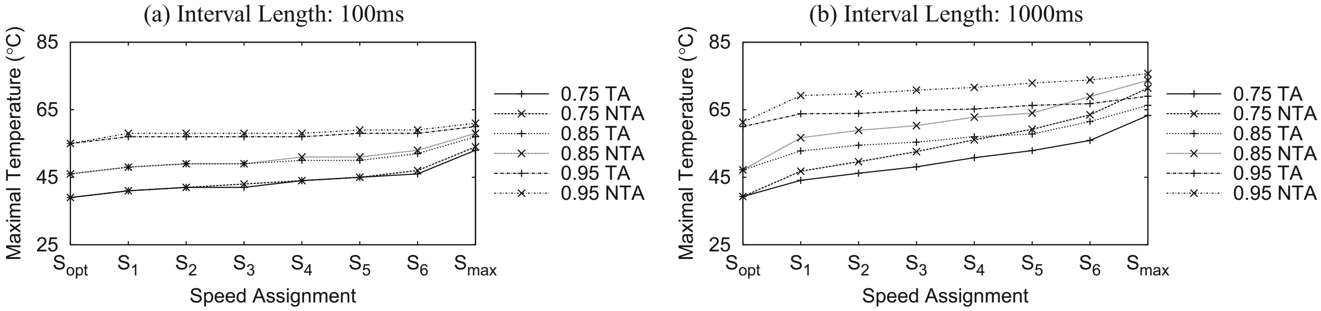
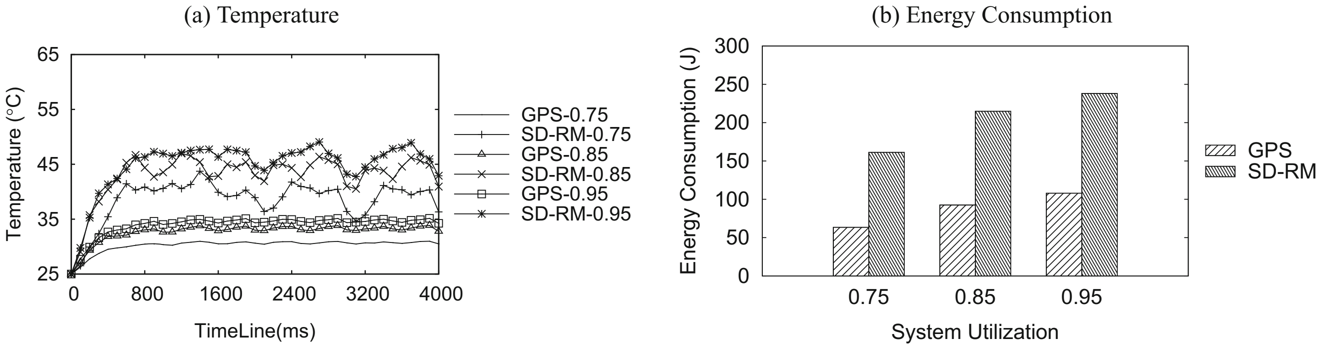


Figure 5 Comparisons between Thermal-Aware(TA) and Non-Thermal-Aware(NTA) algorithms.


(a) Temperature

(b) Energy Consumption

Figure 6 Temperature and energy consumption comparisons with the state-of-the-art.

tasks finish right before their deadlines by executing tasks with fixed execution rates in each scheduling interval, thus leaving the maximal optimization space for the subsequent speed scaling, and on the other hand can balance the thermal load of each task to minimize both the processor temperature and energy consumption through the optimal speed condition we deduced.

7 Conclusions

Through the new method of end-temperature-equivalently task construction, this work proposes and proves the minimal temperature condition for periodic tasks with any scheduling algorithms under both steady and transient states. In the meantime, based on the thermal law we found, this work proposes a novel scheduling algorithm by considering thermal feature as the metric to determine tasks' priorities. Experimental results further validate the task construction method, the minimal temperature condition and thermal-aware scheduling algorithm proposed in this work. Two important conclusions can be obtained from this work: (1) when the optimal temperature condition can be achieved, all of the scheduling algorithms like GPS and EDF result in the same minimal temperature traces since all of the tasks share the same power consumption value and the processor is fully utilized; (2) when the optimal temperature condition cannot be achieved due to some reasons like tasks' timing constraints, the task executing sequence in a scheduling interval of the approximate GPS scheduling determined by the thermal law, exhibits a better thermal-aware feature.

In our future work, we will conduct multi-processor extension for the proposed theorems and conclusions. The challenge in this extension lies in the complex thermal interaction between cores and the effect of cores' layout on the temperature.

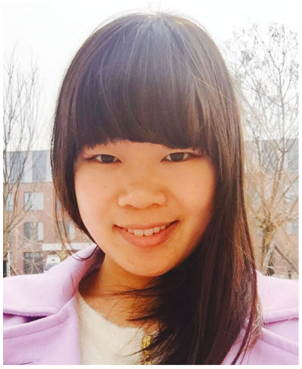
Acknowledgments This work is supported by the National Natural Science Foundation of China under Grant No. 61672143, 61433008, U1435216, 61662057, 61502090, and the Fundamental Research Funds for the Central Universities under Grant No. N161602003.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Xie, G., Jiang, J., Liu, Y., Li, R., Li, K. (2017). Minimizing energy consumption of real-time parallel applications using downward and upward approaches on heterogeneous systems. *IEEE Transactions on Industrial Informatics*, *13*(3), 1068–1078.
2. Lee, C.-H., & Shin, K.G. (2004). On-line dynamic voltage scaling for hard real-time systems using the EDF algorithm. In *Proceedings of the 25th IEEE real-time systems symposium (RTSS 2004), 5–8 December 2004, Lisbon, Portugal* (pp. 319–327).
3. Niu, L., & Li, W. (2017). Energy-efficient scheduling for embedded real-time systems using threshold work-demand analysis. *Journal of Circuits, Systems, and Computers*, *26*(6), 1–36.
4. Jejurikar, R., Pereira, C., Gupta, R.K. (2004). Leakage aware dynamic voltage scaling for real-time embedded systems. In *Proceedings of the 41th design automation conference, DAC 2004, San Diego, CA, USA, June 7–11, 2004* (pp. 275–280).
5. Liao, W., He, L., Lepak, K.M. (2005). Temperature and supply voltage aware performance and power modeling at microarchitecture level. *IEEE Transactions on CAD of Integrated Circuits and Systems*, *24*(7), 1042–1053.
6. Liu, Y., Dick, R.P., Shang, L., Yang, H. (2007). Accurate temperature-dependent integrated circuit leakage power estimation is easy. In *2007 design, automation and test in Europe conference and exposition, DATE 2007, Nice, France, April 16–20, 2007* (pp. 1526–1531).
7. Yang, C.-Y., Chen, J.-J., Thiele, L., Kuo, T.-W. (2010). Energy-efficient real-time task scheduling with temperature-dependent leakage. In *Design, automation and test in Europe, DATE 2010, Dresden, Germany, March 8–12, 2010* (pp. 9–14).
8. Huang, H., Chaturvedi, V., Quan, G., Fan, J., Qiu, M. (2014). Throughput maximization for periodic real-time systems under the maximal temperature constraint. *ACM Transactions on Embedded Computing Systems*, *13*(2s), 70:1–70:22.
9. Chaturvedi, V., Huang, H., Quan, G. (2010). Leakage aware scheduling on maximum temperature minimization for periodic hard real-time systems. In *10th IEEE international conference on computer and information technology, CIT 2010, Bradford, West Yorkshire, UK, June 29–July 1, 2010* (pp. 1802–1809).
10. Skadron, K., Stan, M.R., Sankaranarayanan, K., Huang, W., Velusamy, S., Tarjan, D. (2004). Temperature-aware microarchitecture: modeling and implementation. *TACO*, *1*(1), 94–125.

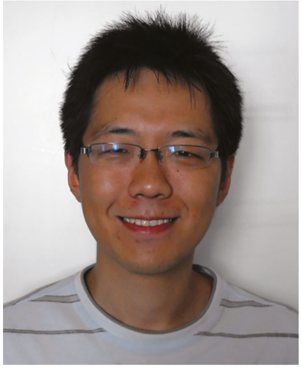
11. Yuan, L., Leventhal, S., Gu, J., Qu, G. (2011). Talk: a temperature-aware leakage minimization technique for real-time systems. *IEEE Transactions on CAD of Integrated Circuits and Systems*, *30*(10), 1564–1568.
12. Huang, H., Chaturvedi, V., Liu, G., Quan, G. (2012). Leakage aware scheduling on maximum temperature minimization for periodic hard real-time systems. *Journal of Low Power Electronics*, *8*(4), 378–393.
13. Jayaseelan, R., & Mitra, T. (2008). Temperature aware task sequencing and voltage scaling. In *2008 international conference on computer-aided design, ICCAD 2008, San Jose, CA, USA, November 10–13, 2008* (pp. 618–623).
14. Fan, M., Chaturvedi, V., Sha, S., Quan, G. (2013). Thermal-aware energy minimization for real-time scheduling on multi-core systems. *SIGBED Review*, *10*(2), 27.
15. Chaturvedi, V. (2013). *Leakage temperature dependency aware real-time scheduling for power and thermal optimization*. Fiu electronic theses and dissertations, Florida International University, Miami.
16. Awan, M.A., Yomsi, P.M., Nelissen, G., Petters, S.M. (2016). Energy-aware task mapping onto heterogeneous platforms using DVFS and sleep states. *Real-Time Systems*, *52*(4), 450–485.
17. Hanumaiah, V., & Vrudhula, S.B.K. (2014). Energy-efficient operation of multicore processors by dvfs, task migration, and active cooling. *IEEE Transactions on Computers*, *63*(2), 349–360.
18. Quan, G., Zhang, Y., Wiles, W., Pei, P. (2008). Guaranteed scheduling for repetitive hard real-time tasks under the maximal temperature constraint. In *Proceedings of the 6th international conference on hardware/software codesign and system synthesis, CODES+ISSS 2008, Atlanta, GA, USA, October 19–24, 2008* (pp. 267–272).
19. Zhou, J., Wei, T., Chen, M., Yan, J., Hu, X.S., Ma, Y. (2016). Thermal-aware task scheduling for energy minimization in heterogeneous real-time mpsoc systems. *IEEE Transactions on CAD of Integrated Circuits and Systems*, *35*(8), 1269–1282.
20. Zhou, J., Yan, J., Chen, J., Wei, T. (2016). Peak temperature minimization via task allocation and splitting for heterogeneous mpsoc real-time systems. *Signal Processing Systems*, *84*(1), 111–121.
21. Zhou, J., Yan, J., Cao, K., Tan, Y., Wei, T., Chen, M., Zhang, G., Chen, X., Hu, S. (2018). Thermal-aware correlated two-level scheduling of real-time tasks with reduced processor energy on heterogeneous mpsocs. *Journal of Systems Architecture*, *82*(1), 1–11.
22. Ahmed, R., Ramanathan, P., Saluja, K.K. (2013). On thermal utilization of periodic task sets in uni-processor systems. In *2013 IEEE 19th international conference on embedded and real-time computing systems and applications, RTCSA 2013, Taipei, Taiwan, August 19–21, 2013* (pp. 267–276).
23. Ahmed, R., Ramanathan, P., Saluja, K.K. (2016). Necessary and sufficient conditions for thermal schedulability of periodic real-time tasks under fluid scheduling model. *ACM Transactions on Embedded Computing Systems*, *15*(3), 49:1–49:26.
24. Parekh, A.K., & Gallager, R.G. (1993). A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking*, *1*(3), 344–357.
25. Baruah, S.K., Easwaran, A., Guo, Z. (2015). Mc-fluid: simplified and optimally quantified. In *2015 IEEE real-time systems symposium, RTSS 2015, December 1–4, 2015, San Antonio, Texas, USA* (pp. 327–337).
26. Ramanathan, S., & Easwaran, A. (2015). Mc-fluid: rate assignment strategies. In *Proceedings of the 3rd workshop on mixed criticality systems (WMC), 1st December 2015, San Antonio, Texas, USA* (pp. 6–11).



Tiantian Li received the B.S. and M.S. degrees from Northeastern University, China, in 2011 and 2013, respectively. She is currently pursuing the PhD degree of Computer Science and Theory in Northeastern University and supervised by Prof. Ge Yu. Her field of interest is energy-efficient computing, distributed computing, and dynamic thermal and power management in real-time systems.

Tianyu Zhang received the B.S. degree from Qingdao University in 2010, the M.S. degree from Northeastern University, China, in 2013. He is currently pursuing the PhD degree with the School of Computer Science and Engineering, Northeastern University, China. His research interests include real-time embedded systems, cyber-physical systems and wireless sensor networks.

Ge Yu received the PhD degree in computer science from Kyushu University of Japan in 1996. He is currently a professor in Northeastern University of China. His research interests include distributed and parallel database, big data storage and management, real-time systems optimization, etc. He has published more than 200 papers in refereed journals and conferences. He is a member of the IEEE Computer Society, IEEE, ACM, and a Senior Member of the China Computer Federation.



Jie Song received the PhD degree from Northeastern University of China in 2010. He is currently an associate professor of Software College, Northeastern University. His research interest includes green computing, big data management and machine learning.