LETTER *Special Section on Formal Approach*

Stochastic Power Minimization of Real-Time Tasks with Probabilistic Computations under Discrete Clock Frequencies*

Hyung Goo PAEK[†], **Jeong Mo YEO**[†], *Nonmembers*, **Kyong Hoon KIM**[††], *and* **Wan Yeon LEE**[†††a)], *Members*

SUMMARY The proposed scheduling scheme minimizes the mean power consumption of real-time tasks with probabilistic computation amounts while meeting their deadlines. Our study formally solves the minimization problem under finitely discrete clock frequencies with irregular power consumptions, whereas state-of-the-arts studies did under infinitely continuous clock frequencies with regular power consumptions.
key words: *power consumption, minimization, real-time task, scheduling*

1. Introduction

In dynamic voltage and frequency scaling (DVFS) mechanism [1], processor speed is proportional to supplied clock frequency and power consumption is approximately proportional to a polynomial function of the clock frequency. Exploiting the convex relationship between clock frequency and power consumption, many scheduling schemes have been suggested to reduce power consumption by decreasing clock frequency to the lower bound completing the worst-case (maximum) computation amount exactly at deadlines of real-time tasks.

A few recent studies [2]–[4] considered varying computation amount, instead of a fixed but the worst-case computation amount. Actual computation amount is smaller than the worst-case in most cases and uncertain until the completion. These studies translate the varying computation amount into a probabilistic computation amount and minimize the mean power consumption of the probabilistic computation amount. Lower clock frequency with lower power consumption is assigned to the computation parts with higher probability, and vice versa. When actual computation amount is smaller than the worst-case computation amount, this stochastic approach consumes less power than the conservative approach that assigns a fixed frequency completing the worst-case computation amount exactly at the deadline. All these literatures solved the minimization problem of power consumption of real-time tasks over infinitely continuous frequencies with an enforced formula between clock frequency and power consumption. However, in real-life DVFS-enabled processors [1], only a finite set of discrete frequencies are available and the relationship between available discrete frequencies and their power consumptions is irregular. Furthermore, most of previous studies [1]–[3] dealt with the minimization problem only for a single real-time task.

The proposed novel scheme formally solves the minimization problem of *multiple* real-time tasks over *finitely discrete* clock frequencies with *irregular* power consumptions. The scheduling scheme minimizes the mean power consumption of multiple real-time tasks with probabilistic computation amounts while meeting their deadlines. The scheme is designed to operate with a polynomial time complexity.

2. Preliminaries

In the considered processor, M periodic tasks are given. The m^{th} task is denoted as T^m. Each task T^m should complete its computation within its arrival period, which becomes its deadline D^m. The required computation amount is uncertain until the completion, but varying computation amount could be estimated from statistical models of the variation sources, on-line profiling, or off-line profiling [2]–[4].

Figure 1 shows the statistical models of varying computation amount, where task computation amount is represented by the number of processor clock cycles required to complete the task computation. Figure 1 (a) shows a probability distribution of required cycles, and Fig. 1 (b) shows the tail cumulative distribution of the probability shown in Fig. 1 (a). When the probability at the c^{th} cycle is denoted as p_c, the cumulative probability at the c^{th} cycle is $\sum_{i=1}^{c} p_i$, and its tail cumulative probability at the c^{th} cycle is $(1 - \sum_{i=1}^{c} p_i)$. Henceforth we denote the tail cumulative probability at the c^{th} cycle of T^m as $\Phi_c^m = (1 - \sum_{i=1}^{c} p_i^m)$. In

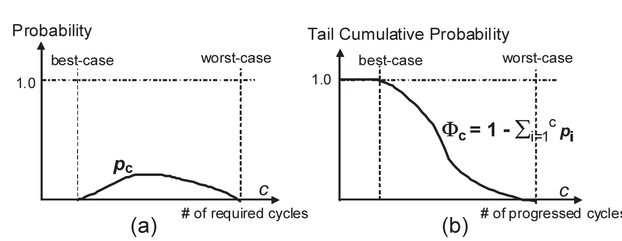

Fig. 1 Statistical distributions of varying computation cycles.

Manuscript received July 14, 2011.
Manuscript revised October 17, 2011.
[†]The authors are with Pukyong National University, S. Korea.
[††]The author is with Gyeongsang National University, S. Korea.
[†††]The author is with Dongduk Women's University, S. Korea.
*This research was supported by Basic Science Research Program through the NRF of Korea funded by the Ministry of Education, Science and Technology (2009-0064347), and by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C1090-1131-0007).
a) E-mail: wanlee@dongduk.ac.kr
DOI: 10.1587/transinf.E95.D.1380

other words, Φ_c^m is the probability that T^m is still running at the c^{th} progressed cycle. Note that $\Phi_{c_1}^m \geq \Phi_{c_2}^m$ for $c_1 < c_2$, because the cumulative distributions of all probability functions are non-decreasing and thus their tail distributions are non-increasing. The worst-case number of required computation cycles for T^m is denoted as W^m. Tasks have different D^m, W^m and Φ_c^m, which are known to the scheduler in advance.

Available K discrete frequencies are denoted as $F_1, \cdots, F_K$ in increasing order. The power consumption at F_i is denoted as P_i. If $F_i < F_j$, then $P_i < P_j$. There is no specific relationship between F_i and P_i. The execution time of each cycle at F_i is $\frac{1}{F_i}$. It is assumed that overhead of switching the supplied clock frequency is negligible.

3. Proposed Scheme

The instant frequency assigned to the c^{th} cycle of T^m is denoted as f_c^m where $f_c^m \in \{F_1, \cdots, F_K\}$. Its power consumption is denoted as $P(f_c^m)$ where $P(F_1) = P_1, \cdots, P(F_K) = P_K$. Then the problem of minimizing the mean power consumption of a single task T^m while meeting its deadline D_m can be formulated as follows:

$$\begin{aligned} &\text{Minimize} && \textstyle\sum_{c=1}^{W^m} P(f_c^m) \cdot \Phi_c^m, \\ &\text{subject to} && \frac{\sum_{c=1}^{W^m} \frac{1}{f_c^m}}{D^m} \leq 1. \end{aligned} \tag{1}$$

The extended problem for M tasks can be formulated as follows:

$$\begin{aligned} &\text{Minimize} && \textstyle\sum_{m=1}^{M} \sum_{c=1}^{W^m} P(f_c^m) \cdot \Phi_c^m, \\ &\text{subject to} && \textstyle\sum_{m=1}^{M} \frac{\sum_{c=1}^{W^m} \frac{1}{f_c^m}}{D^m} \leq 1. \end{aligned} \tag{2}$$

A derived solution determines f_c^m for each m and $1 \leq c \leq W^m$. We refer to a schedule producing a solution of Eq. (2) as *Optimal Schedule*.

3.1 Optimal Schedule of a Single Task

Optimal Schedule has the following properties, proved in Appendix.

Lemma 1: The *power-inefficient* frequency F_y such that $\frac{P_y - P_x}{F_y - F_x} > \frac{P_z - P_y}{F_z - F_y}$ for $F_x < F_y < F_z$ is excluded.

Lemma 2: $f_{c_1} \leq f_{c_2}$ for $1 \leq c_1 < c_2 \leq W$.

By Lemma 1, we hereafter discard the *power-inefficient* frequencies that are not in accordance with a convex function of their power consumptions. After excluding the power-inefficient frequencies, the indexes of the remaining clock frequencies are renumbered in increasing order of their frequency values. For example, let's consider F_1, F_2, F_3 and F_4 are given as inputs. If F_2 is the power-insufficient frequency, then $K = 4$ is updated with $K = 3$ and the old F_3 and F_4 are renumbered as F_2 (= the old F_3) and F_3 (= the old F_4), respectively. Calculating $\frac{P_k - P_{k-1}}{F_k - F_{k-1}} > \frac{P_{k+1} - P_k}{F_{k+1} - F_k}$ for $1 < k < K$ can select all power-inefficient frequencies. Then the power consumptions of the remaining clock frequencies construct a convex function with the input of their frequency values, i.e., $\frac{P_k - P_{k-1}}{F_k - F_{k-1}} \leq \frac{P_{k+1} - P_k}{F_{k+1} - F_k}$ for $1 < k < K$.

By Lemma 2, we consider only switchings of the assigned instant frequency from a lower instant frequency to a higher instant frequency. Hereafter the switching point of the assigned instant frequency from F_{k-1} to F_k (the index of the starting cycle to assign F_k) is denoted as π_k. Lemma 2 means that $\pi_k \leq \pi_{k+1}$ in Optimal Schedule. If F_{k-1} is switched into F_{k+1} instead of F_k, $\pi_{k+1} = \pi_k$. If $F_k, \cdots, F_K$ are not used, $\pi_k = \cdots = \pi_K = (W + 1)$. Then Eq. (1) can be reformulated as follows:

$$\begin{aligned} &\text{Minimize} && P_1 \cdot \textstyle\sum_{c=\pi_1}^{(\pi_2-1)} \Phi_c + P_2 \cdot \sum_{c=\pi_2}^{(\pi_3-1)} \Phi_c \\ & && + \cdots + P_K \cdot \textstyle\sum_{c=\pi_K}^{W} \Phi_c, \\ &\text{subject to} && \frac{\pi_2 - \pi_1}{F_1} + \frac{\pi_3 - \pi_2}{F_2} \cdots + \frac{(W+1) - \pi_K}{F_K} \leq D \end{aligned} \tag{3}$$

where $\pi_1 = 1$ because F_1 is the lowest frequency.

If we know the values of all π_ks, we can directly obtain Optimal Schedule. Unfortunately, however, the values of π_ks depend on the given D. In order to find the values of π_ks, we first examine the relationship among the values of π_ks and next exploit it to obtain the values of π_ks associated with the value of D. The following Theorem 1, proved in Appendix, verifies the relationship among the values of π_ks.

Theorem 1: The values of π_ks in Optimal Schedule obey the following relationship:

$$\Phi_{\pi_x} \cdot \frac{P_x - P_{x-1}}{1/F_{x-1} - 1/F_x} = \Phi_{\pi_y} \cdot \frac{P_y - P_{y-1}}{1/F_{y-1} - 1/F_y}$$

for $2 \leq x < y \leq K$.

From a fixed value π_K, we can calculate deterministic values of $\pi_{K-1}, \cdots, \pi_2$ satisfying the relationship in Theorem 1. Exhaustive searching of $\pi_{K-1}, \cdots, \pi_2$ from $\pi_K = 1$ to $\pi_K = W$ enables us to find the exact π_ks matching D. The procedure for finding π_ks works as follows; Initially, it assigns $\pi_K = 1$ and searches for the switching point π_k such that

$$\Phi_{\pi_k} = \Phi_{\pi_K} \cdot \frac{P_K - P_{K-1}}{1/F_{K-1} - 1/F_K} \cdot \frac{1/F_{k-1} - 1/F_k}{P_k - P_{k-1}} \text{ for } K > k > 1.$$

Because $0 \leq \Phi_{\pi_k} \leq 1$ by the definition of Φ_c, π_k such that $P_{\pi_k} > 1$ is set to 1. It calculates the execution time of the searched schedule and compares it with the given deadline D. If the execution time of the searched schedule is smaller than the deadline, it increases the value of π_K and again searches for the switching point π_k for $K > k > 1$. If $\pi_K = W$ but the execution time is still smaller than the deadline, it increases the value of π_{K-1} after fixing π_K to $(W + 1)$ and searches for the remaining point π_k for $(K - 1) > k > 1$. Similarly, if $\pi_j = W$ but the execution time is still smaller than the deadline, it increases the value of π_{j-1} instead of π_j after fixing π_j to $(W+1)$. This procedure is repeated until the execution of the searched schedule is equal to the deadline.

The computational time complexity of the above procedure is $O(K \cdot (\log_2 W)^2)$ in the average case. If the execution time when $\pi_K = W$ is smaller than D, the replacement of the base value π_K with π_j such that $j < K$ is repeated at most $(K-2)$ times until the execution when $\pi_j = W$ is larger than D. With a base value of π_j, the bisectional search operation of the remaining point π_i satisfying the relationship in Theorem 1 requires $O(\log_2 W)$ steps for each i such that $j > i > 1$. When the base value of π_j is selected in a bisectional manner, the selection operation of the fixed π_j is repeated at most $O(\log_2 W)$ times. Then $O((K-2)\cdot\log_2 W + K\cdot\log_2 W\cdot\log_2 W) = O(K\cdot(\log_2 W)^2)$.

3.2 Optimal Schedule of Multiple Tasks

Given Φ_c^m and D^m, the exhaustive search procedure described in Sect. 3.1 can derive the minimum of Eq. (3), which depends only on the allocated time $\delta^m \cdot D^m$ where $0 < \delta^m \leq 1$ and $\sum_{m=1}^{M} \delta^m \leq 1$. If we can obtain the minimum of Eq. (3) for any allocated time $\delta^m \cdot D^m$, the result of Eq. (2) is entirely dominated by the decision of δ^ms. When Ψ^m denotes the minimum value derived from Eq. (3) with a decided δ^m, Eq. (2) can be reformulated as follows:

$$\text{Mimimize } \sum_{m=1}^{M} \Psi^m \text{ subject to } \sum_{m=1}^{M} \delta^m \leq 1. \tag{4}$$

As the allocated time $A^m = \delta^m \cdot D^m$ increases, the power consumption Ψ^m decreases and the decrement ratio of power consumption per unit time $\frac{\Psi^m}{\partial A^m}$ decreases for each T^m by Lemma 3 in Appendix. In this case, the minimum of Eq. (4) can be achieved by incrementally allocating an additional unit time to the task providing the largest decrement of power consumption with the additional unit time. The minimum of Eq. (4) occurs when $\sum_{m=1}^{M} \frac{A^m}{D^m} = \sum_{m=1}^{M} \delta^m = 1$.

The following numerical procedure allocates the available time to M tasks, so as to maximize the total decrements of power consumptions obtained with the allocated times; Initially, $\frac{W^m}{F_K}$ is assigned to each A^m in order to provide at least the fastest frequency to all computations. Next, it calculates the power decrement of each task when allocating the remaining available time evenly to all tasks. If x denotes the maximum time simultaneously and additionally allocatable to each task, then $\frac{A^1+x}{D^1} + \cdots + \frac{A^M+x}{D^M} = 1$ and thus $x = \frac{1-\sum_{m=1}^{M} A^m/D^m}{\sum_{m=1}^{M} 1/D^m}$. It calculates the decrement of each T^m between power consumption when assigning time A^m and that when assigning time $(A^m + x)$. It selects the task having the largest power decrement and actually allocates the additional time x only to the task (i.e., $A^m \leftarrow (A^m + x)$). This process is repeated until there is no more time allocatable to any task (i.e., $\sum_{m=1}^{M} \frac{A^m}{D^m} = \sum_{m=1}^{M} \delta^m = 1$).

The computational complexity of the above numerical procedure is $O(\log_{\frac{M+1}{M}}(\sum_{m=1}^{M} D^m)\cdot\sum_{m=1}^{M}\{K\cdot(\log_2 W^m)^2 + W^m\})$ in the average case. The operation to search for the values of π_ks with an increased time requires $O(K\cdot(\log_2 W^m)^2)$ steps for each task, as explained in Sect. 3.1. The operation to calculate the power decrement obtained with the additional time requires $O(W^m)$ steps for each task. The operations to allocate addition time to the task with the largest power decrement are repeated at most $O(\log_{\frac{M+1}{M}}(\sum_{m=1}^{M} D^m))$ times, where the total available time is smaller than $\sum_{m=1}^{M} D^m$ and $(1+\frac{1}{M})^\alpha \leq \sum_{m=1}^{M} D^m$.

The proposed scheduling scheme determines f_c^m for each T^m at off-line time according to the above numerical procedure. The scheduler preferentially executes the task T^e with the earliest deadline among multiple tasks, and assigns the determined frequency f_c^e of T^e until the completion of T^e.

4. Conclusions

Our study formally solves a problem of minimizing the mean power consumption of real-time tasks with probabilistic computations under finitely discrete frequencies with irregular power consumptions, whereas state-of-the-arts studies [4] did under infinitely continuous frequencies with regular power consumptions. Our solution can be applied directly to real-life DVFS-enabled processor, whereas the previous solutions cannot. Performance evaluation on our scheme is omitted because the previous studies [2]–[4] verified well that, when actual computation is smaller than the worst-case computation, the stochastic approach designed with probabilistic computation saves more power than the conservative approach designed with the worst-case computation. The problem of minimizing the mean power consumption of probabilistic computations with overhead of switching the supplied clock frequency remains for our further study.

References

[1] R. Xu, C. Xi, R. Melhem, and D. Mossé, "Practical PACE for embedded systems," Int'l Conf. on Embedded Software (EMSOFT), pp.54–63, 2004.

[2] W. Yuan and K. Nahrstedt, "Energy-efficient soft real-time CPU scheduling for mobile multimedia systems," ACM Symp. on Operating Systems Principles (SOSP), pp.149–163, 2003.

[3] J.R. Lorch and A.J. Smith, "PACE: a new approach to dynamic voltage scaling," IEEE Trans. Comput., vol.53, no.7, pp.856–869, July 2004.

[4] C. Xian, Y.H. Lu, and Z. Li, "Dynamic voltage scaling for multitasking real-time systems with uncertain execution time," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.27, no.8, pp.1467–1478, Aug. 2008.

[5] D. Luenberger, Linear and Nonlinear Programming, Addison-Wesley, 1984.

Appendix

Lemma 1: The *power-inefficient* frequency F_y such that $\frac{P_y-P_x}{F_y-F_x} > \frac{P_z-P_y}{F_z-F_y}$ for $F_x < F_y < F_z$ is excluded in Optimal Schedule.

proof: Assume that Optimal Schedule uses the frequency F_y to execute $C_y = (C_x + C_z)$ cycles. We will show that another schedule assigning F_x to C_x cycles and F_z to C_z cycles consumes less power with the same execution time than the assumed Optimal Schedule. When $\frac{C_x+C_z}{F_y} = \frac{C_x}{F_x} + \frac{C_z}{F_z}$, the two schedules have the same execution time. Then $C_z \cdot \frac{F_x}{F_x-F_y} = C_x \cdot \frac{F_z}{F_y-F_z}$. If $\frac{P_y-P_x}{F_y-F_x} > \frac{P_z-P_y}{F_z-F_y}$ and $C_z \cdot \frac{F_x}{F_x-F_y} = C_x \cdot \frac{F_z}{F_y-F_z}$, then $C_x \cdot (\frac{P_x}{F_x} - \frac{P_y}{F_y}) - C_z \cdot (\frac{P_y}{F_y} - \frac{P_z}{F_z}) = (P_x \cdot \frac{C_x}{F_x} + P_z \cdot \frac{C_z}{F_z}) - P_y \cdot \frac{(C_x+C_z)}{F_y} < 0$. That is, the power consumption using F_y to execute (C_x+C_z) cycles, $P_y \cdot \frac{(C_x+C_z)}{F_y}$, is larger than that using F_x to execute C_x cycles and F_z to execute C_z cycles, $(P_x \cdot \frac{C_x}{F_x} + P_z \cdot \frac{C_z}{F_z})$. Hence there is no Optimal Schedule using the frequency F_y. □

Lemma 2: In Optimal Schedule, $f_{c_1} \leq f_{c_2}$ for $1 \leq c_1 < c_2 \leq W$.

proof: Assume that $f_{c_1} > f_{c_2}$ for $c_1 < c_2$ in Optimal Schedule. Let $F_a = f_{c_1}$ and $F_b = f_{c_2}$. Then $F_a > F_b$. By the definition of Φ_c in Sect. 2, $\Phi_{c_1} \geq \Phi_{c_2}$ for $c_1 < c_2$. If $\Phi_{c_1} \geq \Phi_{c_2}$ and $F_a > F_b$, another schedule assigning F_b to the c_1^{th} cycle and F_a to the c_2^{th} cycle consumes less power with the same execution time than the assumed Optimal Schedule assigning F_a to the c_1^{th} cycle and F_b to the c_2^{th} cycle. This is a contradiction on the definition of Optimal Schedule. Hence $f_{c_1} \leq f_{c_2}$ in Optimal Schedule. □

Theorem 1: The values of π_ks in Optimal Schedule obey the following relationship:

$$\Phi_{\pi_x} \cdot \frac{P_x - P_{x-1}}{1/F_{x-1} - 1/F_x} = \Phi_{\pi_y} \cdot \frac{P_y - P_{y-1}}{1/F_{y-1} - 1/F_y}$$

for $2 \leq x < y \leq K$.

proof: Let us apply the Lagrange Multiplier Method [5] to Eq. (3) in Sect. 3.1. Because $P_k \cdot \sum_{c=\pi_k}^{(\pi_{k+1}-1)} \Phi_c = \{(P_k - P_{k-1}) + (P_{k-1} - P_{k-2}) + \cdots + (P_1 - 0)\} \cdot \sum_{c=\pi_k}^{(\pi_{k+1}-1)} \Phi_c$ and $(P_k - P_{k-1}) \cdot (\sum_{c=\pi_k}^{(\pi_{k+1}-1)} \Phi_c + \sum_{c=\pi_{k+1}}^{(\pi_{k+2}-1)} \Phi_c + \cdots + \sum_{c=\pi_K}^{W} \Phi_c) = (P_k - P_{k-1}) \cdot \sum_{c=\pi_k}^{W} \Phi_c$,

$$\begin{aligned} &P_1 \cdot \textstyle\sum_{c=\pi_1}^{(\pi_2-1)} \Phi_c + P_2 \cdot \sum_{c=\pi_2}^{(\pi_3-1)} \Phi_c + \cdots + P_K \cdot \sum_{c=\pi_K}^{W} \Phi_c \\ &\quad = (P_1 - 0) \cdot \textstyle\sum_{c=\pi_1}^{W} \Phi_c + (P_2 - P_1) \\ &\qquad \cdot \textstyle\sum_{c=\pi_2}^{W} \Phi_c + \cdots + (P_K - P_{K-1}) \cdot \sum_{c=\pi_K}^{W} \Phi_c. \end{aligned}$$

Also

$$\begin{aligned} &\tfrac{\pi_2-\pi_1}{F_1} + \tfrac{\pi_3-\pi_2}{F_2} \cdots + \tfrac{W-\pi_K}{F_K} \\ &\quad = -\pi_1 \cdot \tfrac{1}{F_1} + \pi_2 \cdot \left(\tfrac{1}{F_1} - \tfrac{1}{F_2}\right) + \cdots + \pi_K \\ &\qquad \cdot \left(\tfrac{1}{F_{K-1}} - \tfrac{1}{F_K}\right) + W \cdot \tfrac{1}{F_K}. \end{aligned}$$

When $L(\pi_2, \cdots, \pi_K, \lambda) = \sum_{k=2}^{K}\{(P_k - P_{k-1}) \cdot \sum_{c=\pi_k}^{W} \Phi_c\} + P_1 \cdot \sum_{c=\pi_1}^{W} + \lambda \cdot \{D - (\sum_{k=2}^{K} \pi_k \cdot (\frac{1}{F_{k-1}} - \frac{1}{F_k}) + (\frac{W}{F_K} - \frac{\pi_1}{F_1}))\}$,

$$\begin{aligned} \tfrac{\partial L}{\partial \lambda} &= D - \left(\textstyle\sum_{k=2}^{K} \pi_k \cdot \left(\tfrac{1}{F_{k-1}} - \tfrac{1}{F_k}\right) + \left(\tfrac{W}{F_K} - \tfrac{\pi_1}{F_1}\right)\right) \\ &= 0, \end{aligned}$$

and

$$\begin{aligned} \tfrac{\partial L}{\partial \pi_k} &= \tfrac{(P_k - P_{k-1}) \cdot \sum_{c=\pi_k}^{W} \Phi_c}{\partial \pi_n} - \lambda \cdot \left(\tfrac{1}{F_{k-1}} - \tfrac{1}{F_k}\right) \\ &= 0 \quad \text{for } 2 \leq k \leq K. \end{aligned}$$

From the above equation

$$\frac{\sum_{c=\pi_k}^{W} \Phi_c}{\partial \pi_k} = \lambda \cdot \frac{1/F_{k-1} - 1/F_k}{P_k - P_{k-1}}$$

and from solving the differential formula

$$\frac{\sum_{c=\pi_k}^{W} \Phi_c}{\partial \pi_k} = -\Phi_{\pi_k}.$$

Because $-\lambda = \Phi_{\pi_k} \cdot \frac{P_k - P_{k-1}}{1/F_{k-1} - 1/F_k}$ for each $2 \leq k \leq K$,

$$\Phi_{\pi_x} \cdot \tfrac{P_x - P_{x-1}}{1/F_{x-1} - 1/F_x} = \Phi_{\pi_y} \cdot \tfrac{P_y - P_{y-1}}{1/F_{y-1} - 1/F_y}$$

for $2 \leq x < y \leq K$. □

Lemma 3: The power consumption of Optimal Schedule constructs a convexly decreasing function with the input of $\delta \cdot D$.

proof: The switching point π_k of F_k increases with increasing $\delta \cdot D$, because $\delta \cdot D = \sum_{k=2}^{K} \pi_k \cdot (\frac{1}{F_{k-1}} - \frac{1}{F_k}) + (\frac{W}{F_K} - \frac{\pi_1}{F_1})$. Let π_k^a and π_k^b denote the switching points of F_k when using the time $\delta \cdot D$ and the time $(\delta \cdot D + \Delta t)$ for arbitrary $\Delta t > 0$, respectively. Then $\pi_k^a < \pi_k^b$. Because the power consumptions of Optimal Schedule with $\delta \cdot D$ and $(\delta \cdot D + \Delta t)$ are $\sum_{k=2}^{K}\{(P_k - P_{k-1}) \cdot \sum_{c=\pi_k^a}^{W} \Phi_c\} + P_1 \cdot \sum_{c=\pi_1}^{W}$ and $\sum_{k=2}^{K}\{(P_k - P_{k-1}) \cdot \sum_{c=\pi_k^b}^{W} \Phi_c\} + P_1 \cdot \sum_{c=\pi_1}^{W}$ respectively, the decrement of power consumption gained with the additional time Δt is

$$\sum_{k=2}^{K} (P_k - P_{k-1}) \cdot \sum_{c=\pi_k^a}^{\pi_k^b} \Phi_c, \tag{A· 1}$$

where $\Delta t = \sum_{k=2}^{K}(\frac{1}{F_{k-1}} - \frac{1}{F_k}) \cdot (\pi_k^b - \pi_k^a)$. Because $\Phi_{c_1} \geq \Phi_{c_2}$ for $c_1 < c_2$, the value ratio of $\sum_{c=\pi_k^a}^{\pi_n^b} \Phi_c$ to $(\pi_k^b - \pi_k^a)$ decreases with increasing $\delta \cdot D$ for each k. Accordingly, the value ratio of Eq. (A· 1) to Δt decreases with increasing $\delta \cdot D$. This means that the power consumption of Optimal Schedule constructs a convexly decreasing function with the input of $\delta \cdot D$. □