# Work-in-Progress: An ILP Framework for Energy Optimized Scheduling for Weakly-Hard Real-Time Systems

Niraj Kumar
IIT Patna, India
niraj.pcs15@iitp.ac.in

Jaishree Mayank
IIT Patna, India
jmayank@iitp.ac.in

Arijit Mondal
IIT Patna, India
arijit@iitp.ac.in

## ABSTRACT

In this work, we propose an integer linear programming (ILP) model to schedule a set of weakly-hard real-time tasks on a heterogeneous multiprocessor system to minimize energy consumption with the predictable performance. The ILP model provides an optimal solution in terms of energy consumption by selecting the appropriate frequency for each task with (m,k) weakly-hard deadline constraint.

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

## KEYWORDS

Scheduling, Real-time Systems, Weakly-hard, Non-preemptive

**ACM Reference Format:**
Niraj Kumar, Jaishree Mayank, and Arijit Mondal. 2019. Work-in-Progress: An ILP Framework for Energy Optimized Scheduling for Weakly-Hard Real-Time Systems. In *2019 International Conference on Embedded Software (EMSOFT'19), October 13–18, 2019, New York, NY, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3349568.3351546

## 1 INTRODUCTION

In practice, not all systems are hard real-time. The systems, referred to as weakly-hard real-time systems, allow occasional deadline misses in a familiar and predictable fashion such as multimedia applications. The weakly-hard real-time systems are often defined as $(m, k)$ model such that for any set of $m$ consecutive instances at least $k$ instances must meet the deadline otherwise significant degradation in the quality of service (QoS) may be observed. Moreover, with $(m, k)$ constraint satisfied, as number of accepted instances increases, the QoS such as reliability of the system also increases. For many cyber-physical systems, the non-preemptive scheduling is preferred, such as for the systems with very small memory. Moreover, the energy-efficient cyber-physical system design has received significant reserach attention due to the huge operational cost and energy constraint for the battery-powered systems, however, mainly for hard-real time systems. In contrast to the works with focus on the energy-constrained scheduling such as [1], [2], our focus is on the optimization of energy consumption. In this work, we address the problem of scheduling non-preemptive periodic weakly-hard real-time tasks on the heterogeneous multiprocessor system with the objective to minimize the energy consumption.


*EMSOFT'19, October 13–18, 2019, New York, NY, USA*

ACM ISBN 978-1-4503-6924-4/19/10…$15.00
https://doi.org/10.1145/3349568.3351546

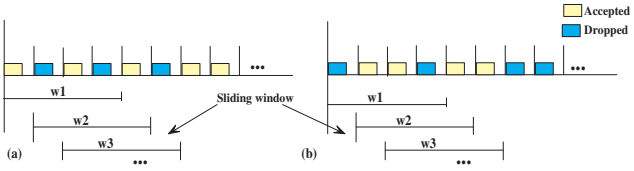


**Figure 1: Sliding window of (4,2)-(m,k) constraint for (a) Non-consecutive (b) Consecutive model**

## 2 PROBLEM FORMULATION AND PROPOSED APPROACH

*System Model:* Let $\mathbb{T} = \{T_i\}$ be the set of $n$ independent non-preemptive periodic real-time tasks. Each task $T_i \in \mathbb{T}$ is characterized by four parameters $\langle e_i, p_i, m_i, k_i \rangle$ where $e_i$ and $p_i$ $(\geq e_i)$ are the worst-case execution time and period (as well as deadline) of task $T_i$, respectively, whereas in any window of $m_i$ consecutive instances of $T_i$, at least $k_i$ instances must meet the deadline (which are referred to as *accepted* instances while remaining as *dropped* instances), in the sliding window as shown in Figure 1. Let $\mathbb{P}$ be the set of $\gamma$ heterogeneous Dynamic Voltage/Frequency Scaling (DVFS) enabled processors such that each processor $P_j \in \mathbb{P}$ can execute with a discrete set of $l$ frequency levels $F_j = \{F_j^k\} | F_j^k < F_j^{k+1}$. Without any loss of generality, we assume that the frequency values are normalized such that $F_j^l \leq 1.0$. Let $e_i$ be the execution time with the normalized frequency $F_j^l = 1.0$.

Let $T_{iq}$ represent the $q^{th}$ instance of task $T_i$ with the release time $(q-1)p_i$ and deadline $qp_i$. The hyper-period is computed as $\delta = lcm\{p_1, \ldots, p_n\}$ and the number of instances of $T_i$ in the hyper-period is $\beta_i = \delta / p_i$. The number of sliding windows corresponding to task $T_i$ is $w_i = \beta_i - m_i + 1$. Let $f_{iq}$ be the frequency with which $T_{iq}$ is executed. Thus, the effective execution time of $T_{iq}$ with $f_{iq}$ is $e_{iq} = e_i / f_{iq}$. Then, for a system-dependent constant $c$, energy consumption for the instance $T_{iq}$ is computed as $E_{iq} = c \times f_{iq}^2 \times e_i$. Let $E_j$ be the energy consumption on processor $P_j$ then the system-level energy consumption $E$ is

$$E = \sum E_j, \quad \forall P_j \in \mathbb{P} \tag{1}$$

*Problem Statement:* For a given set of non-preemptive periodic weakly-hard real-time tasks $\mathbb{T} = \{T_i\}$ and a heterogeneous DVFS-enabled multiprocessor system $\mathbb{P}$ such that each processor $P_j \in \mathbb{P}$ can execute with a discrete set of frequencies $F_j = \{F_j^k\}$, the problem is to schedule each instance $T_{iq}$ of the task $T_i$ and assign

an appropriate frequency level $f_{iq}$ (to $T_{iq}$) with the objective to *minimize* $E$ constrained to

- C1: A processor may execute at most one task at a time.
- C2: An instance can be executed on at most one processor.
- C3: Preemption, migration, or change in frequency are forbidden for an instance during execution.
- C4: Accepted instances must meet the deadline.
- C5: In every window of $m_i$ consecutive instances *any* $k_i$ (or more) or *consecutive* $k_i$ (or more) instances are accepted.

*Proposed Approach:* We have proposed an ILP model to obtain an optimal solution with a set of decision variables and corresponding constraints. Let $v_{ijkt}$ be a binary decision variable such that $v_{ijkt} = 1$ if $T_i$ start execution on $P_j$ at time $t$ with the frequency $F_j^k$, otherwise $v_{ijkt} = 0$. Let $i \in [1, n], j \in [1, \gamma], k \in [1, l]$, and $t \in [0, \delta]$ are the indices to the set of tasks, processors, frequency, and discrete-time unit in the hyper-period, respectively. Let $t_{iq} = [(q-1)p_i, qp_i]$ be the time interval in which $T_{iq}$ has been activated.

Let $T_i$ begin execution at time $t$ on the processor $P_j$ with $k^{th}$ frequency. Then, no other task can start execution at time $t$ on $P_j$.

$$C1 : \sum_i \sum_k v_{ijkt} \leq 1, \qquad \forall j, t$$

Being a weakly-hard real-time system, an instance may be executed (or *accepted*), however at most on one processor, that is to say

$$C2 : \sum_j \sum_k \sum_{t=(q-1)p_i}^{qp_i} v_{ijkt} \leq 1, \qquad \forall i, q \in \{1, \ldots, \beta_i\}$$

Let $T_{iq}$ begins execution at time $t \in t_{iq}$, then to ensure non-preemptive execution no other task can be executed on $P_j$ till $t + e_{ij}^k$ where $e_{ij}^k$ is the execution time of $T_i$ on $P_j$ with frequency $F_j^k$.

$$C3 : v_{ijkt} + \sum_{i'} \sum_{k'=1}^{l} \sum_{t'=t}^{t+e_{ij}^k} v_{i'jk't'} \leq 1, \ \forall i, i' \in [1, n], i \neq i', j, t \in t_{iq}$$

If $T_{iq}$ begins execution at time $t$ on processor $P_j$ with the frequency $F_j^k$, then it must finish by $qp_i$.

$$C4 : t \times v_{ijkt} + e_{ij}^k \leq \left\lceil \frac{t}{p_i} \right\rceil \times p_i$$

Let $z_{iq} = \sum_j \sum_k \sum_{t=(q-1)p_i}^{qp_i} v_{ijkt}, \forall i, q$ thus, if $T_{iq}$ executes successfully then $z_{iq} = 1$ otherwise $z_{iq} = 0$. The constraint C5 can be expressed as *Any k (or more) Instances* (AI) as C5a and *Consecutive k (or more) Instances* (CI) as C5b in the following.

$$C5a : \sum_{q=w}^{w+m_i-1} z_{iq} \geq k_i, \quad \forall i, w \in [1, w_i]$$

$$C5b : \sum_s y_{iws} = 1, \quad \sum_s \sum_{q=w}^{w+k_i-1} (y_{iws} \times z_{iq}) \geq k_i, \forall i, w \in [1, w_i]$$

where $y_{iws} = 1$ if the $s^{th}$ instance of $T_i$ in window $w$ start its execution, otherwise $y_{iws} = 0$ such that $w \in [1, (\beta_i - m_i + 1)]$, and $s \in [1, (m_i - k_i + 1)]$. The non-linear expression in C5b (product term) is converted into a linear expression using extra variables[1].

[1]We introduce additional variables $u$ (dropping indices for simplicity) to convert the non-linear term into linear one as follows: $u = y \times z$, $u \leq y$, $u \leq z$, $u \geq y + z - 1$

**Table 1: Performance Comparison**

| Tasks | Avg. Energy | | | | Observed Throughput | | | | Time (sec) | | Memory(GB) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CI | | AI | | CI | | AI | | CI | AI | CI | AI |
| | 4 | 8 | 4 | 8 | 4 | 8 | 4 | 8 | 4 | | 4 | |
| I | II | III | IV | V | VI | VII | VIII | IX | X | XI | XII | XIII |
| 5 | 1.58 | 0.94 | 1.47 | 0.65 | 11 | 17 | 14 | 19 | 4.3 | 1.3 | 1.4 | 0.8 |
| 6 | 2.14 | 1.87 | 1.9 | 1.43 | 20 | 26 | 23 | 29 | 8.6 | 3.6 | 2.4 | 1.6 |
| 7 | 2.68 | 2.17 | 2.16 | 1.55 | 27 | 36 | 34 | 43 | 15.3 | 4.7 | 3.6 | 2.5 |
| 8 | 3.12 | 2.58 | 2.59 | 2.38 | 36 | 53 | 42 | 62 | 27.6 | 6.3 | 4.7 | 3.4 |
| 9 | 3.89 | 3.15 | 3.42 | 2.93 | 47 | 61 | 55 | 67 | 35.4 | 9.7 | 6.1 | 5.7 |
| 10 | 5.27 | 4.72 | 4.78 | 3.21 | 53 | 68 | 64 | 74 | 42.8 | 15.2 | 8.3 | 7.1 |

CI - Consecutive $k$ Instances, AI - Any $k$ Instances

Then, the energy consumption to execute $T_{iq}$ on $P_j$ is

$$E_j^{iq} = \sum_k \sum_{t \in t_{iq}} v_{ijkt} \times \left\langle (F_j^k)^2 \times e_{ij}^k \right\rangle, \qquad \forall i, j \tag{2}$$

Effectively, the total energy consumption on $P_j$ is $E_j = \sum_i \sum_{q=1}^{\beta_i} E_j^{iq}, \ \forall i.$

Our final target is to minimize $E$ as described in Eq. 1.

## 3 SIMULATION

We have performed the experimental study with the IBM cplex tool to obtain an optimal solution for the problem formulated in section 2. Table 1 shows the average of the performance measures for 100 test-cases for CI as well as for AI with $n \in [5, 10]$ on 4 and 8 processors. As shown in Table 1, the energy consumption for CI is significantly larger than that of AI. However, we found that the total number of instances accepted (i.e., observed throughput) for execution corresponding to AI is significantly more than that of CI. The simulation time and memory usage both follow a similar pattern. We observe that with the increase in the number of processors, the energy consumption reduces and the observed throughput increases. Although, the non-linear constraints for CI are converted to linear (by cplex), however the number of constraints increases. Whereas the constraints for AI are linear; thus, the computation time for AI is comparatively smaller than that of CI. With the increasing number of tasks the memory usages increases. Moreover, during simulation study we observed that for 10 tasks approximately 7 – 8 GB memory is required. The proposed ILP formulation provides an optimal solution, however, scaling is the issue which can be addressed with heuristic approaches.

## 4 CONCLUSION AND FUTURE WORK

In this work, we have addressed the problem of scheduling a set of non-preemptive periodic weakly-hard real-time tasks on a heterogeneous multiprocessor system to optimize energy consumption. We have proposed an ILP formulation to obtain the optimal solution in terms of energy consumption. Due to significantly large computation time and memory requirement to solve the problem optimally, we are currently working towards efficient heuristic solutions. In addition, including reliability constraint is also a part of our future work.

## REFERENCES


[1] Tarek A AlEnawy and Hakan Aydin. 2005. Energy-Constrained Scheduling for Weakly-Hard Real-Time Systems. In *RTSS*. Citeseer, 376–385.

[2] Yeonhwa Kong and Hyeonjoong Cho. 2012. Energy-constrained scheduling for weakly-hard real-time tasks on multiprocessors. In *Computer Science and Convergence*. Springer, 335–347.