# Variable Voltage Task Scheduling Algorithms for Minimizing Energy

Ali Manzak
Arizona State University
Tempe, AZ 85287-5706
manzak@asu.edu

Chaitali Chakrabarti
Arizona State University
Tempe, AZ 85287-5706
chaitali@asu.edu

## ABSTRACT

In this paper we propose variable voltage task scheduling algorithms (periodic as well as aperiodic) that minimize energy. We first apply the existing task scheduling algorithms to obtain a feasible schedule and then distribute the available slack using an iterative algorithm that satisfies the theoretically obtained relation for minimum energy. We show experimentally that the voltage assignment obtained by our algorithm is very close (0.1% error) to that of the optimal assignment.

## 1. INTRODUCTION

Energy consumption is a very important performance metric in todays world full of portable devices. Substantial energy minimization can be achieved by lowering the supply voltage instead of idling, when the computational load of a task is low. Thus determining the supply voltage of the processor as it executes tasks with different computational loads such that the energy consumption is minimum (without sacrificing performance) is indeed an important problem.

In this paper we consider a system which consists of a processor that is capable of operating over a range of supply voltages and a DC-DC converter (located off chip) that is capable of supplying these voltages. Examples include Intel's XScale processor and Transmeta's Crusoe processor. The input to our system consists of task arrival times, deadline times, execution times, switching activities, periods and/or the time constraint $T_{total}$ (defined as the time to complete all computations). Our aim is to determine the operating voltage of the processor as it executes each task such that the energy consumption is minimum.

There are several on-line and off-line algorithms for the variable voltage task scheduling problem [1]-[9]. The scheduling problem for the case when there are no dependencies between the tasks is addressed in [1] and solved using integer linear programming. A two phase scheduling algorithm is presented in [2] where in the first phase, all tasks are scheduled at the nominal voltage and in the second phase, the voltages of each task are adjusted till no further reduction in power is obtained. The scheduling algorithm in [3] schedules the critical jobs at the highest speed and then constructs a subproblem for the remaining jobs and solves it recursively. The algorithm gives a minimum energy schedule for off-line preemptive scheduling. In contrast, the algorithm in [5] schedules tasks with relaxed deadlines and few overlaps first. Fixed priority scheduling method that considers best and worst case execution time is presented in [4]. If, in addition, release times and deadlines are known apriori, the energy savings increase [6]. Systems where the voltage can be varied during the execution of a task exploit the slack in workload variation and result in significant energy reduction [7], [8].

Our approach to solving the task scheduling problem is to first develop a relationship between the task voltages for minimizing energy consumption, and then to develop a simple heuristic to satisfy the relation. We use the Lagrange multiplier method to show that the minimum energy configuration corresponds to the case when $\alpha(V - V_t)^3$ is the same for all the tasks for both periodic and aperiodic schedules, where $V$ is the operating voltage, $\alpha$ is the switching activity, and $s$ is the execution time in terms of control cycles. In our procedure, we first apply the existing task scheduling algorithm to obtain a feasible schedule, and then distribute the available extra slack using an iterative algorithm that satisfies the minimum energy relation. Our procedure has been applied to $10,000$ randomly generated task configurations as well as some real-life cases. For the randomly generated cases, the energy savings are between $43.5 - 45.7\%$ when $T_{total} = 1.5T_{crit}$ for aperiodic schedules. If the delay to change the converter voltage is $0.05T_{total}$, then the savings drop to $40 - 41.7\%$. This paper is an extension of our previous work in [9], where we had only considered non-preemptive aperiodic tasks.

The rest of the paper is organized as follows. Section 2 formalizes the problem and presents the iterative algorithm. Section 3 presents the algorithms EDD and EDF for aperiodic task assignment and RM and EDF for periodic task assignment. Section 4 includes the results on user generated as well as real life examples. Section 5 concludes the paper.

## 2. FORMALIZATION

### 2.1 Problem Definition

Given a set of $n$ tasks $(\gamma_1, \gamma_2, ...., \gamma_n)$ and a computation time $T_{total}$, our aim is to schedule the tasks in time $T_{total}$ such that the energy consumption is minimum. Associated with task $j$ are the following parameters: $a_j$: the arrival time of task $j$, $d_j$: the deadline time of task $j$, $e_j$: the execution time of task $j$ with the maximum voltage $V_{ref}$ $p_j$: the period of task $j$, $\alpha_j$: the average switching activity of task $j$.

Scheduling the tasks is equivalent to determining the supply voltage of the processor during execution of each of the tasks. In our model, the supply voltage remains constant during the execution of the entire task.

## 2.2 Aperiodic (Single) Tasks

In this section, we consider aperiodic tasks, which includes single tasks and identical jobs activated at irregular intervals. If $\tau_j$ is the execution time of task $j$, and $n$ is the number of tasks, then

$$T_{total} \geq \sum_{j=1}^{n} \tau_j = \sum_{j=1}^{n} s_j * k' C_L \frac{V_j}{(V_j - V_t)^2}, \qquad (1)$$

where $s_j$ is the number of control cycles required to execute task $j$, $V_j$ is the circuit supply voltage, $V_t$ is the threshold voltage, $C_L$ is the load capacitance and $k'$ is a device parameter which depends on the transconductance and the width to length ratio of the transistors.

### 2.2.1 Formulation for energy minimization

The energy of task $j$ is given by $\alpha_j C_L V_j^2 s_j$, where $\alpha_j$ is the average switching activity of task $j$ and the terms $C_L$, $V_j$, $s_j$ are the same as defined before. The total energy consumption is thus $E_{total} = \sum_{j=1}^{n} \alpha_j C_L V_j^2 s_j$.

Our aim is to minimize $E_{total}$ subject to the constraint $T_{total} = constant$. We use the Lagrange multiplier method to determine the supply voltage of each task.

We find $E_{total}$ is minimum when the following condition is satisfied.

$$\frac{\alpha_1 V_1 (V_1 - V_t)^3}{V_1 + V_t} = ..... = \frac{\alpha_n V_n (V_n - V_t)^3}{V_n + V_t}$$

For $V_j > 3V_t$, this is approximated to

$$\alpha_1 (V_1 - V_t)^3 = ..... = \alpha_n (V_n - V_t)^3 \qquad (2)$$

The error in energy calculation is $\approx 0.1\%$ (see Figure 1, $E_{alg}$, $E_{opt}$ difference).

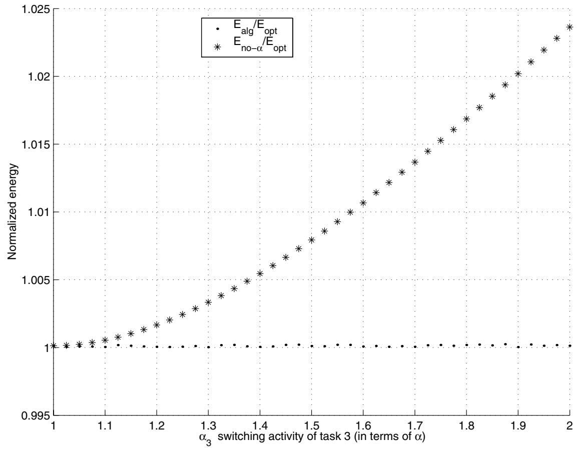

**Figure 1: Normalized energy vs. $\alpha_3$ for three task assignment problem when $\alpha_1 = \alpha_2 = \alpha$, and $s_1 = s_2 = s_3$.**

### 2.2.2 Graphical illustration of the formulations

The energy minimization result is graphically illustrated in Figure 1 for the case when there are three independent tasks (i.e. there is no dependency between the tasks, tasks are soft and activated simultaneously) that have to be computed in time $T_{total}$. In Figure 1, $\alpha_1 = \alpha_2 = \alpha$, $s_1 = s_2 = s_3$, and $\alpha_3$ is varied from $\alpha$ to $2\alpha$. $E_{opt}$ is the lowest energy when switching activities are considered, $E_{alg}$ is the energy using our formulation in eqn. 2 , $E_{no-\alpha}$ is the lowest energy when the switching activities are ignored. $E_{alg}$ is very close to $E_{opt}$ (0.1% error) and $E_{no-\alpha}$ is also close to $E_{opt}$ when the switching activity ratio of the tasks is at the most 2 (3% error).

## 2.3 Periodic Task Assignment

In this section, we consider periodic tasks, that is tasks that consist of a sequence of identical jobs (instances), activated at regular intervals. If $p_j$ is the period of task j, and $n$ is the number of tasks, then $\frac{\tau_j}{p_j}$ is the fraction of processor time spent in execution of task j. So, the utilization factor $U = \sum_{j=1}^{n} \frac{\tau_j}{p_j} \leq 1$.

In the energy minimization formula, we use $U = constant$ as our constraint. The total energy consumption in time $T_{total}$ is

$$E_{total} = \sum_{j=1}^{n} \frac{T_{total}}{p_j} \alpha_j C_L V_j^2 s_j$$

Using the same approximations as in aperiodic task assignment, we find that $E_{total}$ is minimum when

$$\alpha_1 (V_1 - V_t)^3 = ..... = \alpha_n (V_n - V_t)^3 \qquad (3)$$

Thus to minimize energy, the task voltages have to be chosen according to eqn.(3). Note that the condition for periodic and aperiodic tasks is the same.

## 2.4 Slack adjustment for minimum energy

The task voltages are iteratively adjusted so that the Lagrange relation is satisfied. Let $m$ be the task with the minimum $\alpha$ value. Then in each iteration the task voltages are adjusted with respect to task $m$.

$$V_j = (\frac{\alpha_{min}}{\alpha_j})^{1/3} (V_m - V_t) + V_t \qquad (4)$$

In the first iteration, task $m$ is set to $V_{start}$, where

$$V_{start} = (\frac{\alpha_{max}}{\alpha_{min}})^{1/3} (V_{max} - V_t) + V_t \qquad (5)$$

In the iterative procedure, in each step, the voltage of task $m$ decreases by $\Delta V$, where $\Delta V$ is the converter sensitivity. The maximum number of iterations is then given by $k_{max} = \frac{1}{\Delta V}(V_{start} - V_{min}) = \frac{1}{\Delta V}((\frac{\alpha_{max}}{\alpha_{min}})^{1/3}(V_{max} - V_t) + V_t - V_{min})$. By doing a binary search, the number of iterations can be reduced to $log(k_{max})$.

## 3. TASK ASSIGNMENTS

## 3.1 Aperiodic EDD (Jackson's) Algorithm

The Earliest Due Date (EDD) algorithm schedules tasks with earlier due date first. It has a complexity of $O(nlogn)$, since the procedure sorts the tasks by increasing deadlines.

If after the application of the EDD algorithm, a feasible solution exists and unused slack is available, then the following algorithm is invoked to decrease the energy consumption. In each iteration, the voltage of the critical task (or task $m$) is decreased, the voltages of the other tasks adjusted, and the finishing time of the task, $f$, compared with its deadline. If there is a violation of task $j$ at step $k$, i.e. there is a violation in the assignment $V_j(k)$, then the previous voltage value ($V_j(k-1)$) is the optimal voltage value. Furthermore, since all tasks with earlier due date than task $j$ could have caused deadline violation of task $j$, all these tasks are assigned voltages corresponding to iteration $k-1$, $V_l = V_l(k-1)$ for $l = 1$ to $j$. The algorithm continues until voltages for all the tasks are determined.

```
for $k = 1$ to $k_{max}$
    update $V_m(k)$, $\tau_m(k)$ for the critical task
    for each unscheduled task $\gamma_j \in \Gamma$
        update $V_j(k)$, $\tau_j(k)$ for the tasks using eqn.(4)
        $f_j = f_{j-1} + \tau_j(k)$
        if $f_j > d_j(k)$
            for $l = 1$ to $j$
                Schedule: $V_l = V_l(k-1)$.
                return end, if all the tasks are scheduled.
```

**Complexity:** The worst case complexity of the algorithm is $O(nk_{max})$. This is because there are at most $k_{max}$ iterations, and in each iteration, at most $n$ task voltages are calculated.

## 3.2 Aperiodic EDF (Earliest Deadline First) Algorithm

Earliest Deadline First (EDF) is a dynamic scheduling algorithm that at any instant executes the task with the earliest absolute deadline among all the ready tasks. In our algorithm, whenever a new task arrives, the voltage values of the unscheduled tasks are updated according to the minimum energy equation provided that the deadline constraints are not violated.
**Complexity:** In each cycle, the task with the earliest deadline is scheduled with complexity $O(nlog(k_{max}))$, using a binary search. The overall complexity of the algorithm is $O(n^2 log(k_{max}))$.

## 3.3 Periodic RM (Rate Monotonic) Algorithm

The Rate Monotonic (RM) scheduling algorithm assigns priorities to tasks according to their request rates. In addition, deadline of a task ($d_j$) is equal to its period. The sufficient but not necessary guarantee test is $U \leq U_{lub} = n(2^{1/n} - 1)$. In our algorithm, the voltage values of the unscheduled tasks are updated according to the minimum energy equation provided $U \leq U_{lub}$.
**Complexity:** The worst case complexity of the algorithm is $O(nlog(k_{max}))$. This is because at most $log(k_{max})$ iterations are needed, and in each iteration at most $n$ calculations are done.

## 3.4 Periodic EDF Algorithm

The Earliest Deadline First (EDF) algorithm is a dynamic scheduling rule that selects tasks according to their absolute deadlines. Deadlines of the tasks are assumed to be equal to their periods for simplicity. The guarantee test is $U = \sum_{i=1}^{n} \frac{\tau_i}{p_i} \leq 1$

In our algorithm, whenever a new task arrives, the voltage values of the unscheduled tasks are updated according to the minimum energy equation provided $U \leq 1$.

| Alg. | | Energy Saving | | |
|---|---|---|---|---|
| | | $D_c = 0$ | $D_c = 0.05$ | $D_c = 0.1$ |
| EDD | $T_t = 1.5T_c$ | 43.5 | 40 | 35.7 |
| | $T_t = 2T_c$ | 63.5 | 61.2 | 58.8 |
| Aper. EDF | $T_t = 1.5T_c$ | 43.5 | 40 | 35.7 |
| | $T_t = 2T_c$ | 57.4 | 55.8 | 54.2 |
| RM | | $U_c = 0$ | $U_c = 0.05$ | $U_c = 0.1$ |
| | $U_{lub} = 1.5U_{3.3}$ | 45.7 | 41.7 | 37.6 |
| | $U_{lub} = 2U_{3.3}$ | 62.8 | 60.3 | 57.8 |
| Per. EDF | $U = 1.5U_{3.3}$ | 45.7 | 41.7 | 37.6 |
| | $U = 2U_{3.3}$ | 62.8 | 60.3 | 57.8 |

**Table 1: Energy saving change for different converter delays for aperiodic schedules and different converter utilizations for periodic schedules ($T_t = T_{total}, T_c = T_{crit}$).**

**Complexity:** In each cycle, the task with the earliest deadline is scheduled with complexity $O(nlog(k_{max}))$. The overall complexity of the algorithm is $O(n^2 log(k_{max}))$.

## 4. RESULTS

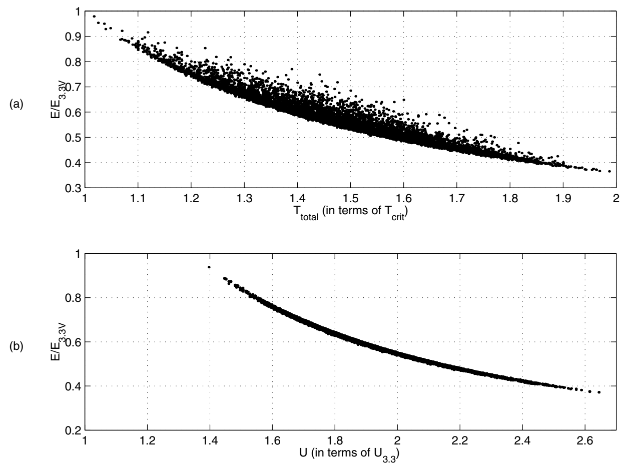

**Figure 2: Five task assignment problem for $\alpha_{max} = 2\alpha_{min}$, $s_{max} = 2s_{min}$, $p_{max} = 2p_{min}$.(a) EDD: Normalized energy vs. normalized delay. (b) RM: Normalized energy vs.normalized utilization**

## 4.1 Randomly generated task configurations

We experiment with 10,000 different task configurations, where each configuration consists of 5 different tasks with task execution times randomly chosen between $s_{min}$ and $10s_{min}$ and switching activities randomly chosen between $\alpha_{min}$ and $2\alpha_{min}$.

**EDD algorithm:** The task execution times are distributed normally between $T_{crit}$ to $2T_{crit}$, with a mean of $1.5T_{crit}$ and standard deviation of $0.16T_{crit}$. Figure 2(a) plots the normalized energy as a function of $T_{total}$. We see that as the delay increases, tasks can be assigned to lower voltages and as a result the normalized energy reduces.

**Aperiodic EDF algorithm:** The energy savings for the aperiodic EDF algorithm are almost the same as EDD, since the variation in $\alpha$ is the same.

| Appl. | Alg. | $\alpha$ | Energy Saving % | |
|---|---|---|---|---|
| | | | $U_c = 0$ | $U_c \neq 0$ |
| CNC | RM | same | 44.7 | 44.7 |
| | | diff. | 46 | 43.3 |
| | EDF | same | 64.7 | 64.7 |
| | | diff. | 65.4 | 64.3 |
| INS | EDF | same | 37.3 | 37.3 |
| | | diff. | 45 | 44.4 |
| Avio. | EDF | same | 32.2 | 32.3 |
| | | diff. | 25.9 | 25.5 |

**Table 2: Energy savings for real-life examples.**

**RM schedule:** The task utilizations are distributed normally between $U_{lub} = 0.74$ to $0.5U_{lub} = 0.37$. Figure 2(b) plots the normalized energy as a function of the normalized utilization factor. We see that as the normalized utilization factor increases, tasks can be assigned to lower voltages, and as a result the normalized energy reduces.

**Periodic EDF algorithm:** The energy curve are similar to that obtained by the RM schedule. However the energy savings are much higher for EDF, compared to RM, since the slack is fully utilized.

## 4.2 Effect of Converter Delay

The delay to change the voltage in the converter and the delay to change the clock frequency are important issues to be considered in the variable task scheduling problem. To model the effect of converter delay, we consider the ratio of $D_c$, the converter delay due to voltage and frequency change, and $T_{total}$ for aperiodic assignment. Similarly, for aperiodic assignment, we consider the utilization factor of the converter, $U_c$, defined as the slack that gets absorbed because of voltage and frequency change.

The average energy savings for different converter delays and utilizations are summarized in Table 1. For instance, for EDD, when $T_{total} = 1.5T_{crit}$, the saving is 43.5% when $D_c = 0$ and 40% when $D_c = 0.05T_{total}$. For RM, when $U_{lub} = 1.5U_{3.3}$, the saving is 45.7% when $U_c = 0$ and 41.7% when $U_c = 0.05$. Thus the effect of converter delay and utilization cannot be ignored.

## 4.3 Real Life Examples

We have also considered some real-time applications, CNC (Computerized Numerical Control), INS (Inertial Navigation System) and Avionics task set. The deadlines of the tasks are assumed to be equal to their periods (for simplification). The worst case delay to vary the clock frequency and the supply voltage is taken to be $10\mu s$ [8].

We consider 2 cases. In the first case, switching activities are assumed the same for all the tasks and in the second case, switching activities are chosen between $\alpha_{min}$ and $2\alpha_{min}$. When switching activities are the same, the minimum energy solution corresponds to the case where all the task voltages are the same. So, there is no delay in the converter and energy savings do not change. When switching activities are different, the energy savings change with the converter delay. Table 2 summarizes our results for those three cases. For instance, when EDF is applied to INS, the energy savings drop from 45% to 44.4%, when switching activities are taken into account ($\alpha$ diff.).

| Algorithm | | | Complexity | |
|---|---|---|---|---|
| | | | before | after |
| sta. | per. | RM | $O(nlogn)$ | $O(nlogk) + O(nlogn)$ |
| | aper. | EDD | $O(nlogn)$ | $O(nk) + O(nlogn)$ |
| dyn. | per. | EDF | $O(n^2)$ | $O(n^2logk)$ |
| | aper. | EDF | $O(n^2)$ | $O(n^2logk)$ |

**Table 3: Complexity of the algorithms before and after energy minimization, where $n$ is the number of tasks and $k$ is the number of iterations.**

## 5. CONCLUSIONS

In this paper we present an extension of existing task scheduling algorithms to minimize energy. The main features of our scheme are

1. It can be used widely for different classes of task scheduling problems.
2. It finds a near-optimal (0.1% error) solution.
3. It has a low complexity; the asymptotic complexity of the scheduling algorithms increases mildly when the heuristic is applied (see Table 3).
4. It considers the effect of the delay to change the converter voltage and the clock frequency.
5. It is more realistic, since switching activities of the tasks are also considered.

**Acknowledgements:** This work was partly sponsored by the Center for Low Power Electronics.

## 6. REFERENCES

[1] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processor," ISLPED, pp 197-202, 1998.
[2] I. Hong, D. Kirovski G. Qu, M. Potkonjak and M-B. Srivastava, "Power Optimization of Variable Voltage Core-Based Systems" DAC, pp 176-181, 1998.
[3] F. Yao, A. Demers and S. Shenker, "A scheduling model for reduced CPU energy," FOCS, pp 374-82, 1995.
[4] Y. Shin and K. Choi "Power conscious fixed priority scheduling for hard real-time systems," DAC, pp 134-139, 1999.
[5] J. Pouwelse, K. Langendoen and H. Sips, "Energy Priority Scheduling for Variable Voltage Processors," ISLPED 2001.
[6] G. Quan and X. Hu, "Energy Efficient Fixed-Priority Scheduling for Real-Time Systems on Variable Voltage Processors," DAC 2001.
[7] S. Lee and T. Sakurai, "Run-time Voltage Hopping for Low Power Real-Time Systems," DAC, pp. 806-809, 2000.
[8] T. Pering, T. Burd and R. Brodersen, "The simulation and evaluation of dynamic voltage scheduling algorithms," ISLPED, pp. 76-81, 1998.
[9] A. Manzak and C. Chakrabarti "Variable voltage task scheduling for minimizing energy or minimizing power" ICASSP, vol 6. pp. 3239-42, 2000.