Discrete and Continuous Optimal Control for Energy Minimization in Real-Time Systems

Bruno Gaujal, Alain Girault, **Stéphan Plassart**

25 September 2020

EBCCSP2020

Introduction

- Minimizing the energy consumption J of an embedded system under real-time (RT) constraints.
- A single-core processor executes a finite set of jobs.
- HRTS: Hard Real Time System.
- HRT constraints: Each job must finish before its deadline.
- The processor can change its speed s, with DVFS technology.

$$\begin{aligned} s \nearrow &\quad\Rightarrow\quad \text{job execution time} \searrow \\ s \nearrow &\quad\Rightarrow\quad J \nearrow \end{aligned}$$

Constrained Optimal Control Problem

Online Optimal Control Problem:

> *At time $\boldsymbol{t}$, the decision maker (processor) chooses the speed $s(\boldsymbol{t})$ based on the current state (jobs released before $\boldsymbol{t}$) and statistical information about the future job arrivals.*

Constrained Optimal Control Problem

Online Optimal Control Problem:

At time $\boldsymbol{t}$, the decision maker (processor) chooses the speed $s(\boldsymbol{t})$ based on the current state (jobs released before $\boldsymbol{t}$) and statistical information about the future job arrivals.

Goal of the decision maker:

Minimize the expected total energy cost while satisfying all the deadline constraints on the jobs.

Constrained Optimal Control Problem

Online Optimal Control Problem:

At time $\boldsymbol{t}$, the decision maker (processor) chooses the speed $s(\boldsymbol{t})$ based on the current state (jobs released before $\boldsymbol{t}$) and statistical information about the future job arrivals.

Goal of the decision maker:

Minimize the expected total energy cost while satisfying all the deadline constraints on the jobs.

- Continuous Time Control (CTC): at any time $t \in \mathbb{R}$ the controller chooses the current speed.

  $\Rightarrow \nexists$ general algorithm.

- Discrete Time Control (DTC): decisions can only be made at the release times of the jobs.

  $\Rightarrow \exists$ optimal solutions for this problem: Dynamic Programming (DP)

Our Solution to Solve the Continuous Time Control

- Proving that the continuous time control (CTC) solution is not better than the discrete time control (DTC) solution.

- As a consequence any optimal DTC solution is **also** an optimal CTC solution.

Our Solution to Solve the Continuous Time Control

- Proving that the continuous time control (CTC) solution is not better than the discrete time control (DTC) solution.

- As a consequence any optimal DTC solution is **also** an optimal CTC solution.

Outline of the talk:

1. Model description.
2. Discrete Time Control (DTC) solution.
3. Continuous Time Control (CTC) solution.

Model Presentation

HRTS Model Presentation

- A single uni-core processor, equiped with DVFS, running at speed $s(t)$, with $s(t) \in \mathcal{S} = \{0, \cdots, s_{\max}\}$.

- Finite set of **jobs** over a finite horizon T.

Jobs (r, c, d) characterized by :

- $r \in \mathbb{N}$: release time.
- $c \in \mathbb{N}$: execution time (size).
- $d \in \mathbb{N} \,|\, d \leq \Delta$: relative deadline.

HRTS Model Presentation

- A single uni-core processor, equiped with DVFS, running at speed $s(t)$, with $s(t) \in \mathcal{S} = \{0, \cdots, s_{\max}\}$.

- Finite set of **jobs** over a finite horizon T.

Jobs (r, c, d) characterized by :

- $r \in \mathbb{N}$: release time.
- $c \in \mathbb{N}$: execution time (size).
- $d \in \mathbb{N} \,|\, d \leq \Delta$: relative deadline.

- **Power consumption at speed s: $Q(s(t))$**

Q is a convex and increasing function in s.

HRTS Model Presentation

- A single uni-core processor, equiped with DVFS, running at speed $s(t)$, with $s(t) \in \mathcal{S} = \{0, \cdots, s_{\max}\}$.

- Finite set of **jobs** over a finite horizon T.

Jobs (r, c, d) characterized by :

- $r \in \mathbb{N}$: release time.
- $c \in \mathbb{N}$: execution time (size).
- $d \in \mathbb{N} \,|\, d \leq \Delta$: relative deadline.

- **Power consumption at speed** s**:** $Q(s(t))$

Q is a convex and increasing function in s.

- **Energy:** $J = \int_0^T Q(s(t))dt$

Problem Statement

1. What is the scheduling policy (j_π) of jobs to ensure that no job misses its deadline ?

   $\Rightarrow$ Choosing the job to execute at each instant t.

Problem Statement

1. What is the scheduling policy (j_π) of jobs to ensure that no job misses its deadline ?

   $\Rightarrow$ Choosing the job to execute at each instant t.

2. What is the speed policy (s_π) to choose at each instant so that the energy consumption is optimal ?

   $\Rightarrow$ Choosing the processor speed $s(t)$ at each instant t

Problem Statement

1. What is the scheduling policy (j_π) of jobs to ensure that no job misses its deadline ?

   $\Rightarrow$ Choosing the job to execute at each instant t.

2. What is the speed policy (s_π) to choose at each instant so that the energy consumption is optimal ?

   $\Rightarrow$ Choosing the processor speed $s(t)$ at each instant t

Optimization under constraint:

1. Each job finishes before its deadline.
2. The expected energy consumption is minimized.

Scheduling Policy Choice

- What is the scheduling policy (j_π) to choose to be certain that each job is executed before its deadline ?

$$\left.\begin{array}{l} \bullet \{s_\pi, j_\pi\} \text{ executes all jobs} \\ \text{before their deadlines.} \\ \\ \bullet \text{EDF policy is optimal} \\ \text{for scheduling jobs.} \end{array}\right\} \Rightarrow \{s_\pi, j_{EDF}\} \text{ also executes all jobs before their deadlines.}$$

$\Rightarrow$ Choice of Earliest Deadline First (EDF)

Scheduling Policy Choice

- What is the scheduling policy (j_π) to choose to be certain that each job is executed before its deadline ?

$$\left.\begin{array}{l} \bullet \{s_\pi, j_\pi\} \text{ executes all jobs} \\ \text{before their deadlines.} \\ \\ \bullet \text{ EDF policy is optimal} \\ \text{for scheduling jobs.} \end{array}\right\} \Rightarrow \{s_\pi, j_{EDF}\} \text{ also executes all jobs before their deadlines.}$$

$\Rightarrow$ Choice of Earliest Deadline First (EDF)

New Optimisation Problem

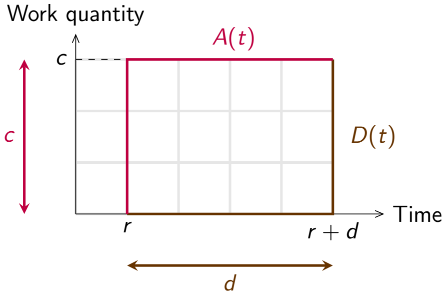
What is the optimal speed policy such that the expected energy consumption is minimized and no job misses its deadline ?

Discrete Time Control (DTC)

Work Quantity for Discrete Time Control

A single Job (r, c, d):

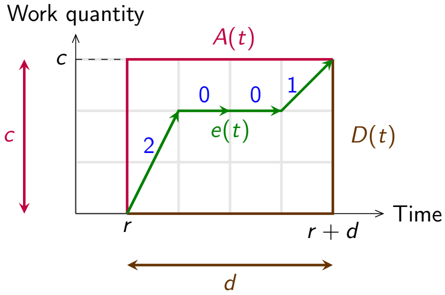
- A(t) $=$ work quantity arrived until time t.
- D(t) $=$ work quantity that must be executed before t.



Work Quantity for Discrete Time Control

A single Job (r, c, d):

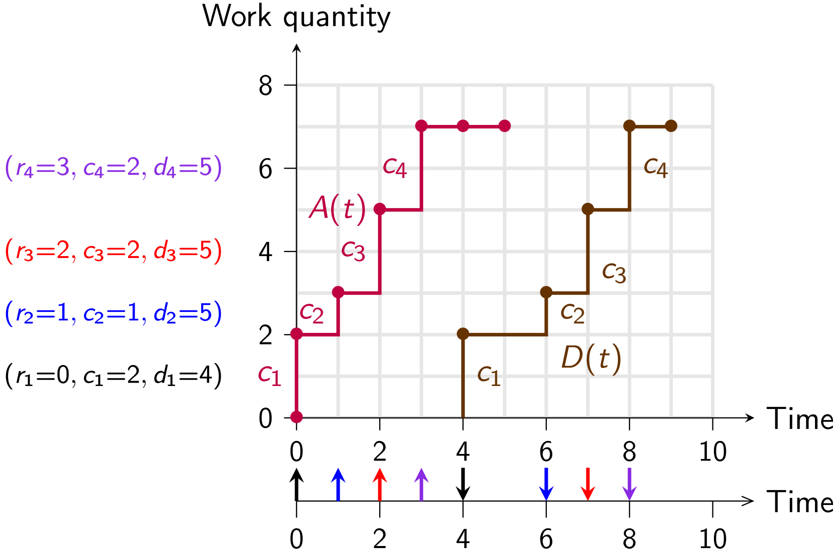
- $\mathrm{A(t)}$ = work quantity arrived until time t.
- $\mathrm{D(t)}$ = work quantity that must be executed before t.
- $e(t)$ = work executed by the processor until time t with speeds $s(u)$.



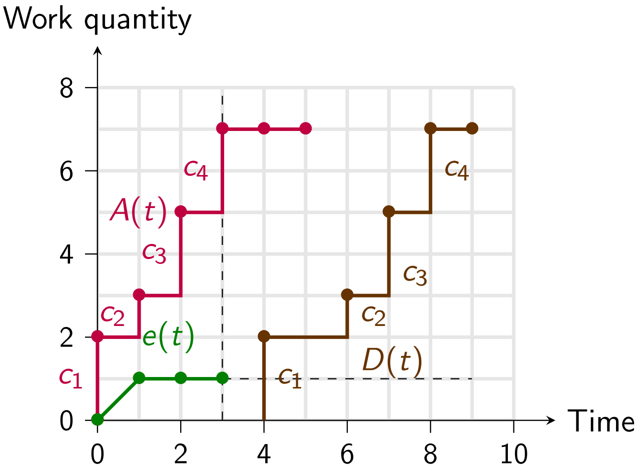
Two constraints on $e(t)$:

1. $e(t) \geq D(t)$: The job must be finished before its deadline.
2. $e(t) \leq A(t)$: The job can not be executed before it arrives.

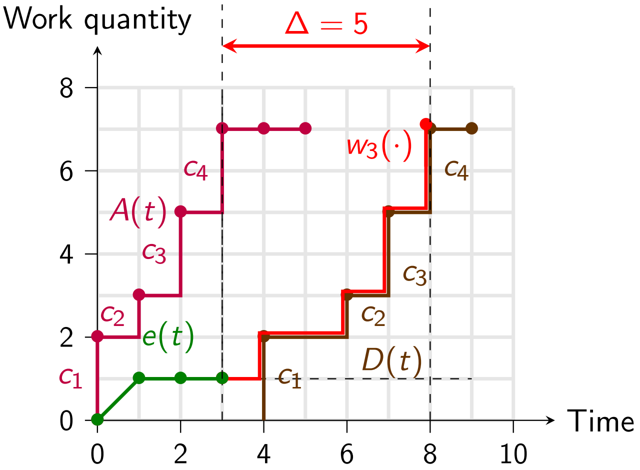
DTC: Remaining Work $w_t(\cdot)$ with $\Delta = 5$



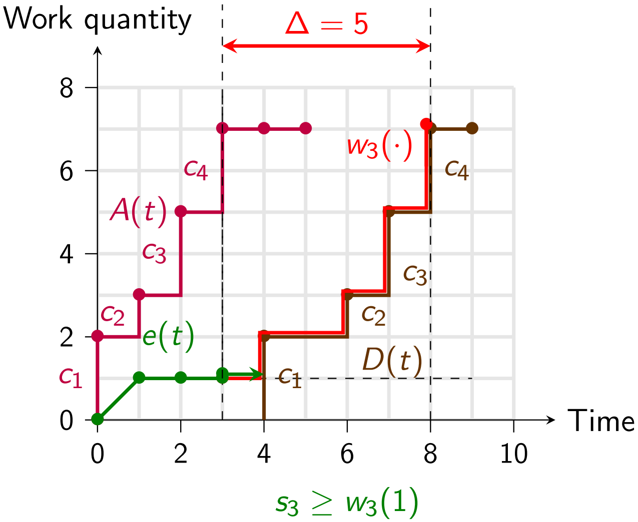
DTC: Remaining Work $w_t(\cdot)$ with $\Delta = 5$



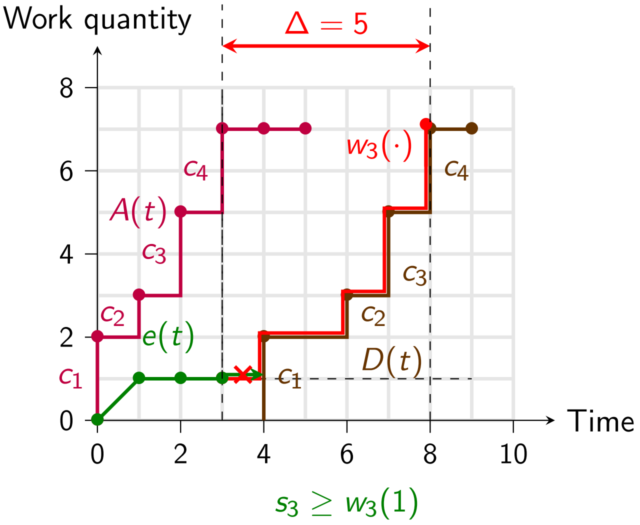
DTC: Remaining Work $w_t(\cdot)$ with $\Delta = 5$



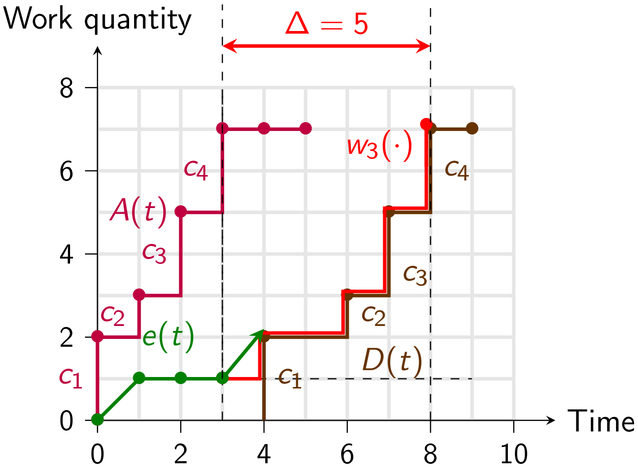
DTC: Remaining Work $w_t(\cdot)$ with $\Delta = 5$



DTC: Remaining Work $w_t(\cdot)$ with $\Delta = 5$

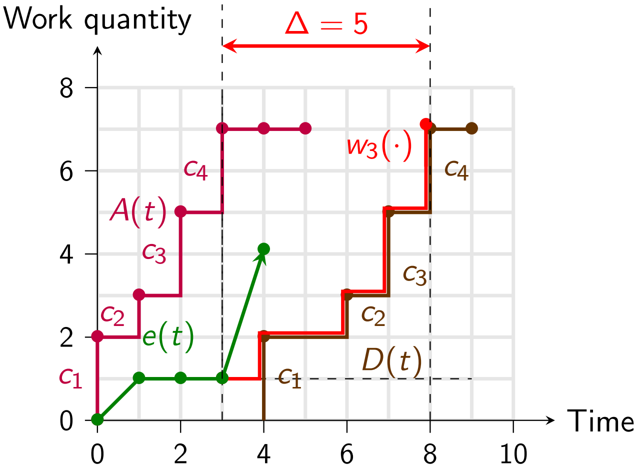


DTC: Remaining Work $w_t(\cdot)$ with $\Delta = 5$



$s_3 \geq w_3(1)$

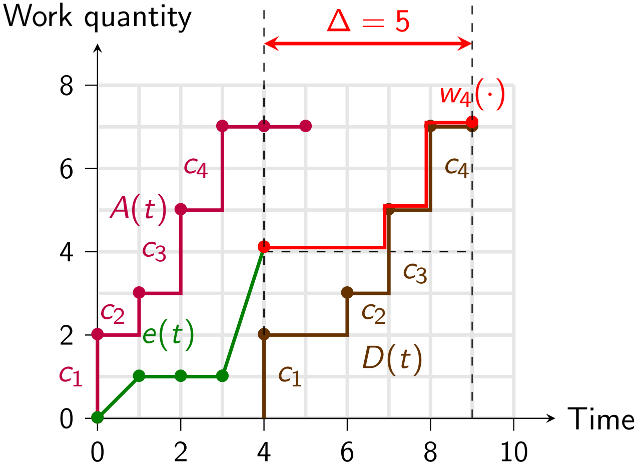
DTC: Remaining Work $w_t(\cdot)$ with $\Delta = 5$



$s_3 \geq w_3(1)$

$w_4 = Succ(w_3, s_3 = 3, a_4 = 0)$

DTC: Remaining Work $w_t(\cdot)$ with $\Delta = 5$



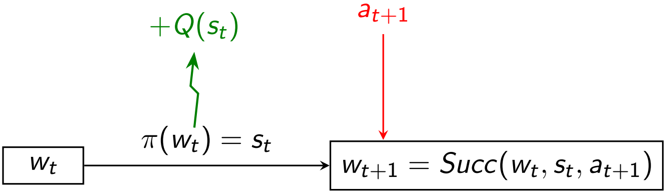
$s_4 \geq w_4(1)$

$w_4 = Succ(w_3, s_3 = 3, a_4 = 0)$

DTC: Optimal Control under HRT Constraints

- a_t = Work quantity arrived at t.
- w_t = System state at t.
- s_t = Speed used by the processor at t.

State evolution under speed policy π:



With the HRT constraint: $s_t \geq w_t(1)$

DTC: Finite Horizon Markov Decision Process (MDP)

- The minimal total expected processor energy consumption in discrete time ($J^{*,\mathbb{N}}$) from 0 to T:

$$J^{*,\mathbb{N}}(w_0) = \min_{\pi} \left\{ \mathbb{E} \left(\sum_{t=0}^{T} Q(\pi_t(w_t)) \right) \right\} \tag{1}$$

DTC: Finite Horizon Markov Decision Process (MDP)

- The minimal total expected processor energy consumption in discrete time ($J^{*,\mathbb{N}}$) from 0 to T:

$$J^{*,\mathbb{N}}(w_0) = \min_{\pi} \left\{ \mathbb{E} \left(\sum_{t=0}^{T} Q(\pi_t(w_t)) \right) \right\} \tag{1}$$

- Backward induction to compute $J^{*,\mathbb{N}}$:

$P_n(a)$ = occurrence probability of the work quantity arrival a.

$$J^{*,\mathbb{N}}_{n-1}(w) = \min_{s \in \mathcal{A}(w)} \left\{ Q(s) + \sum_{a} P_n(a) J^{*,\mathbb{N}}_{n} \Big(Succ(w, s, a) \Big) \right\} \tag{2}$$

- Admissible speed set $\mathcal{A}(w)$ to guarantee that each job is executed before its deadline:

$$\mathcal{A}(w) = \{ s \in \mathcal{S} \mid s \geq w(1) \}$$

Continuous Time Control (CTC)

Discrete Time Control (DTC)

- The processor may change its speed at integer time $n \in \{0, 1, \cdots, T\}$.
- Integer time evolution of the state:

$$\forall u \in \mathbb{N}, \quad w_{n+1}(u) = Succ(w_n(u), \pi(w_n), a_n(u))$$

Continuous Time Control (CTC)

- The processor may change its speed at any time $t \in [0, T]$.
- Continuous time evolution of the state:

(1) $n = 0$: $\forall u \geq 0$,

$$w_0^\pi(u) = 0.$$

(2) $n > 0$: $\forall t \in \,]n, n+1]$,

$$w_t^\pi(u) = \left(w_n^\pi(u) - \int_n^t \pi(w_v^\pi) dv \right)^+ + a_t(u).$$

CTC: Consecutive Speeds Set

Theorem

If the set $\mathcal{S}$ is made of consecutive speeds (i.e. $\mathcal{S} = \{0, 1, 2, \ldots, s_{\mathsf{max}}\}$) and the power function Q is increasing and convex, then there is no energy gain for the processor to change its speed at non-integer times.

CTC: Consecutive Speeds Set

Theorem

If the set $\mathcal{S}$ is made of consecutive speeds (i.e. $\mathcal{S} = \{0, 1, 2, \ldots, s_{\text{max}}\}$) and the power function Q is increasing and convex, then there is no energy gain for the processor to change its speed at non-integer times.

Proof.

- Consider that the processor can change its speed at times $t \in \mathbb{N}$ as well as at times $t + 1/2$.

CTC: Consecutive Speeds Set

Theorem

If the set $\mathcal{S}$ is made of consecutive speeds (i.e. $\mathcal{S} = \{0, 1, 2, \ldots, s_{\max}\}$) and the power function Q is increasing and convex, then there is no energy gain for the processor to change its speed at non-integer times.

Proof.

- Consider that the processor can change its speed at times $t \in \mathbb{N}$ as well as at times $t + 1/2$.
  - (i) Proving that changing speed at $t + 1/2$ does not improve the energy consumption with $s \in [0, s_{\max}]$.

CTC: Consecutive Speeds Set

Theorem

If the set $\mathcal{S}$ is made of consecutive speeds (i.e. $\mathcal{S} = \{0, 1, 2, \ldots, s_{\max}\}$) and the power function Q is increasing and convex, then there is no energy gain for the processor to change its speed at non-integer times.

Proof.

- Consider that the processor can change its speed at times $t \in \mathbb{N}$ as well as at times $t + 1/2$.
  (i) Proving that changing speed at $t + 1/2$ does not improve the energy consumption with $s \in [0, s_{\max}]$.
  (ii) Proving that choosing a non-integer speed at t does not improve the energy consumption, so $s(t) \in \{0, 1, \cdots, s_{\max}\} = \mathcal{S}$.

CTC: Consecutive Speeds Set

Theorem

If the set $\mathcal{S}$ is made of consecutive speeds (i.e. $\mathcal{S} = \{0, 1, 2, \ldots, s_{\max}\}$) and the power function Q is increasing and convex, then there is no energy gain for the processor to change its speed at non-integer times.

Proof.

- Consider that the processor can change its speed at times $t \in \mathbb{N}$ as well as at times $t + 1/2$.
  - (i) Proving that changing speed at $t + 1/2$ does not improve the energy consumption with $s \in [0, s_{\max}]$.
  - (ii) Proving that choosing a non-integer speed at t does not improve the energy consumption, so $s(t) \in \{0, 1, \cdots, s_{\max}\} = \mathcal{S}$.
- Generalization $\forall t \in \mathbb{N}/2^i$ with $\forall i \in \mathbb{N}$.

CTC: Consecutive Speeds Set

Theorem

If the set $\mathcal{S}$ is made of consecutive speeds (i.e. $\mathcal{S} = \{0, 1, 2, \ldots, s_{\max}\}$) and the power function Q is increasing and convex, then there is no energy gain for the processor to change its speed at non-integer times.

Proof.

- Consider that the processor can change its speed at times $t \in \mathbb{N}$ as well as at times $t + 1/2$.
  - (i) Proving that changing speed at $t + 1/2$ does not improve the energy consumption with $s \in [0, s_{\max}]$.
  - (ii) Proving that choosing a non-integer speed at t does not improve the energy consumption, so $s(t) \in \{0, 1, \cdots, s_{\max}\} = \mathcal{S}$.
- Generalization $\forall t \in \mathbb{N}/2^i$ with $\forall i \in \mathbb{N}$.
- By continuity of the total energy:

Changing its speed at integer times is always optimal with a consecutive set of speeds.

CTC: Sketch of the Proof (i)

Goal: proving that changing speed at $t + 1/2$, with $t \in \mathbb{N}$ does not improve the energy consumption.

- Recall of the dynamic programming equation:

$$J_t^*(w) = \min_{s \in \mathcal{A}(w)} \left\{ Q(s) + \sum_a P_t(a) J_{t+1}^* \Big(Succ(w, s, a) \Big) \right\} \tag{3}$$

CTC: Sketch of the Proof (i)

Goal: proving that changing speed at $t+1/2$, with $t \in \mathbb{N}$ does not improve the energy consumption.

- Recall of the dynamic programming equation:

$$J_t^*(w) = \min_{s \in \mathcal{A}(w)} \left\{ Q(s) + \sum_a P_t(a) J_{t+1}^* \Big(Succ(w, s, a) \Big) \right\} \tag{3}$$

- Combining the two dynamic programming equations at t and $t+1/2$:

$$\begin{aligned} J_t^*(w) \quad &= \quad \min_{u_{[0,1/2]} \in \mathcal{A}_1(w), u_{[1/2,1]} \in \mathcal{A}_2(w)} \left(Q \left(\frac{u_{[0,1/2]} + u_{[1/2,1]}}{2} \right) \right. \\ &+ \quad \left. \sum_a P_t(a) J_{t+1}^* \left(Succ \left(w, \frac{u_{[0,1/2]} + u_{[1/2,1]}}{2}, a \right) \right) \right) \end{aligned} \tag{4}$$

where $\mathcal{A}_2(w) = \{s \in \mathcal{S} : s \geq 2w(1)\}$, $\mathcal{A}_1(w) = \mathcal{S}$.

CTC: Sketch of the Proof (i)

Goal: proving that changing speed at $t + 1/2$, with $t \in \mathbb{N}$ does not improve the energy consumption.

- Combining the two dynamic programming equations at t and $t + 1/2$:

$$\begin{aligned} J_t^*(w) &= \min_{u_{[0,1/2]} \in \mathcal{A}_1(w), u_{[1/2,1]} \in \mathcal{A}_2(w)} \left(Q\left(\frac{u_{[0,1/2]} + u_{[1/2,1]}}{2} \right) \right. \\ &+ \left. \sum_a P_t(a) J_{t+1}^* \left(Succ\left(w, \frac{u_{[0,1/2]} + u_{[1/2,1]}}{2}, a \right) \right) \right) \quad (3) \end{aligned}$$

  where $\mathcal{A}_2(w) = \{s \in \mathcal{S} : s \geq 2w(1)\}$, $\mathcal{A}_1(w) = \mathcal{S}$.

- Changing speed at half times is equivalent to choosing half speeds at integer times.

CTC: Sketch of the Proof (ii)

- Combining the two dynamic programming equations at t and $t + 1/2$:

$$J_t^*(w) = \min_{u_{[0,1/2]} \in \mathcal{A}_1(w), u_{[1/2,1]} \in \mathcal{A}_2(w)} \Bigg(Q\left(\frac{u_{[0,1/2]} + u_{[1/2,1]}}{2}\right) + \sum_a P_t(a) J_{t+1}^*\left(Succ\left(w, \frac{u_{[0,1/2]} + u_{[1/2,1]}}{2}, a\right)\right)\Bigg) \quad (4)$$

  where $\mathcal{A}_2(w) = \{s \in \mathcal{S} : s \geq 2w(1)\}$, $\mathcal{A}_1(w) = \mathcal{S}$.

- The set of optimal speeds in Eq. (4) can be reduced to two cases:
  1. $u_{[1/2,1]} = u_{[0,1/2]}$
  2. $u_{[1/2,1]} = u_{[0,1/2]} + 1$

CTC: Sketch of the Proof (ii)

- The set of optimal speeds in Eq. (4) can be reduced to two cases:
  1. $u_{[1/2,1]} = u_{[0,1/2]}$
  2. $u_{[1/2,1]} = u_{[0,1/2]} + 1$ $\Longrightarrow$ Not optimal !

CTC: Sketch of the Proof (ii)

- The set of optimal speeds in Eq. (4) can be reduced to two cases:
  1. $u_{[1/2,1]} = u_{[0,1/2]}$ $\Longrightarrow$ Integer Speed at t, QED
  2. $u_{[1/2,1]} = u_{[0,1/2]} + 1$ $\Longrightarrow$ Not optimal !

CTC: Sketch of the Proof (ii)

- The set of optimal speeds in Eq. (4) can be reduced to two cases:
  1. $u_{[1/2,1]} = u_{[0,1/2]}$ $\Longrightarrow$ Integer Speed at t, QED
  2. $u_{[1/2,1]} = u_{[0,1/2]} + 1$ $\Longrightarrow$ Not optimal !

- By induction on t, we can show with $u = u_{[0,1/2]}$:

$$J\left(\text{Deterministic speed policy, using at } t,\ \frac{u + u + 1}{2}\right)$$

$$= J\left(\text{Randomized speed policy, using at } t,\ u \text{ w.p. } \frac{1}{2} \text{ and } u + 1 \text{ w.p. } \frac{1}{2}\right)$$

CTC: Sketch of the Proof (ii)

- The set of optimal speeds in Eq. (4) can be reduced to two cases:
  1. $u_{[1/2,1]} = u_{[0,1/2]}$ $\Longrightarrow$ Integer Speed at t, QED
  2. $u_{[1/2,1]} = u_{[0,1/2]} + 1$ $\Longrightarrow$ Not optimal !

- By induction on t, we can show with $u = u_{[0,1/2]}$:

$$J\left(\text{Deterministic speed policy, using at } t,\ \frac{u+u+1}{2}\right)$$
$$= J\left(\text{Randomized speed policy, using at } t,\ u \text{ w.p. } \frac{1}{2} \text{ and } u+1 \text{ w.p. } \frac{1}{2}\right)$$

- There exists an optimal deterministic speed policy by MDP theory.
  $\Rightarrow$ Here, it uses on $[t, t+1]$ either speed u or speed $u+1$.

CTC: Sketch of the Proof (ii)

- The set of optimal speeds in Eq. (4) can be reduced to two cases:
  1. $u_{[1/2,1]} = u_{[0,1/2]}$ $\Longrightarrow$ Integer Speed at t, QED
  2. $u_{[1/2,1]} = u_{[0,1/2]} + 1$ $\Longrightarrow$ Not optimal !

- By induction on t, we can show with $u = u_{[0,1/2]}$:

$$J\left(\text{Deterministic speed policy, using at } t,\ \frac{u+u+1}{2}\right)$$
$$= J\left(\text{Randomized speed policy, using at } t,\ u \text{ w.p. } \frac{1}{2} \text{ and } u+1 \text{ w.p. } \frac{1}{2}\right)$$

- There exists an optimal deterministic speed policy by MDP theory.
  $\Rightarrow$ Here, it uses on $[t, t+1]$ either speed u or speed $u+1$.

- $u_{[1/2,1]} = u_{[0,1/2]} + 1$ is not strictly better than $u_{[1/2,1]} = u_{[0,1/2]}$.

CTC: End of the Proof

- Changing speeds at half times does not improve the energy !

CTC: End of the Proof

- Changing speeds at half times does not improve the energy !
- Changing speeds at $t \in \mathbb{N}/2^i$ does not improve the energy $\forall i \in \mathbb{N}$.

CTC: End of the Proof

- Changing speeds at half times does not improve the energy !
- Changing speeds at $t \in \mathbb{N}/2^i$ does not improve the energy $\forall i \in \mathbb{N}$.
- By continuity of the total energy:

CTC: End of the Proof

- Changing speeds at half times does not improve the energy !
- Changing speeds at $t \in \mathbb{N}/2^i$ does not improve the energy $\forall i \in \mathbb{N}$.
- By continuity of the total energy:

Changing speeds at integer times is always optimal
with a consecutive set of speeds.

Extensions

1. Non consecutive speed set $\mathcal{S}$:

   Emulating non available integer speeds with V_{dd}*-hopping* by using two neighboring $s \in \mathcal{S}$.

2. Non convex power function $Q(\cdot)$:

   Replacing the power function $Q(\cdot)$ by its convex hull $\widehat{Q}(\cdot)$.

Conclusion

+ Minimizing in continuous time the energy consumption of a processor with a discrete set of speeds executing jobs with discrete features (arrival times, sizes and deadlines).

+ The continuous time optimization problem can be solved in discrete time.

+ In practice, the discrete optimal speed policy can be computed effectively using a finite Markov Decision Process.

    − The size of the state space of the MDP grows exponentially fast with Δ (the maximal deadline).

+ Can be implemented in an embedded system.