Optimal Processor speeds

Jonatha Anselmi and Bruno Gaujal
Inria and Univ. Grenoble Alpes

Aussois, 2024

Problem Setting with Continous Speed

Input: One task with a deadline and one processor with variable speed $s \in [0, s_{\max}]$.
Output: speed of the processor minimizing energy consumption

Problem Setting with Continous Speed

Input: One task with a deadline and one processor with variable speed $s \in [0, s_{\max}]$.
Output: speed of the processor minimizing energy consumption

More precisely,
Task: deadline D and random size $W \in [W_{\min}, W_{\max}]$. The probability distribution $F(w) := \mathbb{P}(W \leq w)$ is known to the scheduler.
Energy: At speed $s \in [0, s_{\max}]$, the power consumption is $P(s)$ (increasing, convex).
Objective Function: Find a speed profile that minimizes the expected energy consumption to execute the task while satisfying its deadline.

Problem Setting with Discrete Speeds

Input: A task of random size with a deadline and N processors.
Output: Optimal activation (on/off) of the processors minimizing energy consumption

Problem Setting with Discrete Speeds

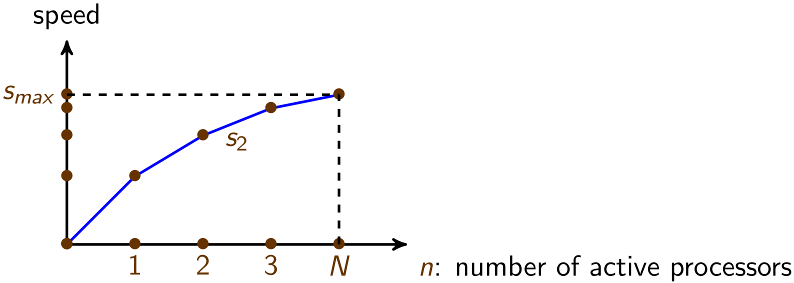
Input: A task of random size with a deadline and N processors.
Output: Optimal activation (on/off) of the processors minimizing energy consumption

Task: A task with deadline D and with random size $W \in [W_{\min}, W_{\max}]$. The probability distribution $F(w) := \mathbb{P}(W \leq w)$ is known to the scheduler.
Energy: When active (resp. inactive) , each processor power consumption is P_{on} (resp. P_{idle}).
Speed-up: When n processors are active, the effective speed (work executed per time unit) is s_n, **increasing concave** in n. The power function is

$$P(s_n) := nP_{\text{on}} + (N-n)P_{\text{idle}}.$$

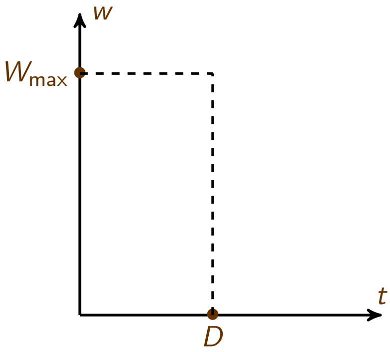
Objective Function: Find a processor activation profile that minimizes the expected energy consumption to execute the task while satisfying its deadline.

Speed-up function is concave



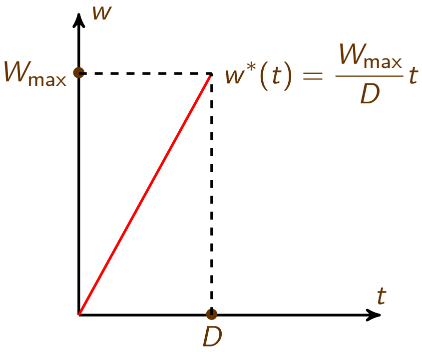
The function $n \to s_n$ is increasing and concave.
The function $P(s_n) : s_n \to nP_{\text{on}} + (N - n)P_{\text{idle}}$ is increasing and convex in s.

Deterministic Size: Constant Speed



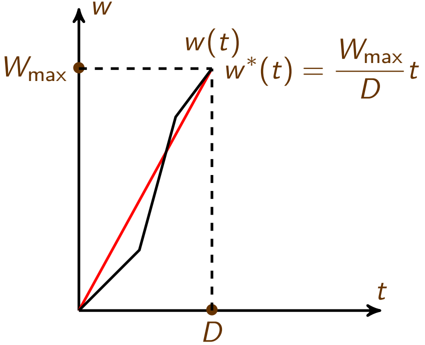
$W = W_{max}$.

Deterministic Size: Constant Speed



$W = W_{\max}$.

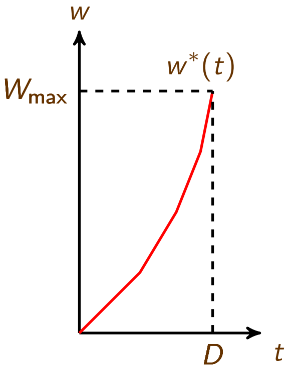
Deterministic Size: Constant Speed



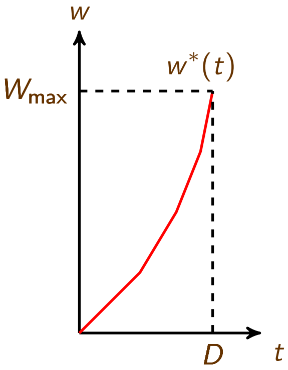
$W = W_{max}$.

By convexity, the optimal solution uses a constant speed: $s^* = \frac{W_{max}}{D}$.

Stochastic case: Procrastination

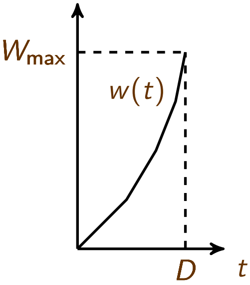


Stochastic case: Procrastination

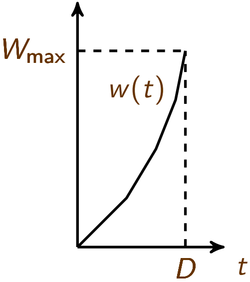


The optimal speed should increase over time, to take advantage of small sizes.

Speed selection as a function of the workload



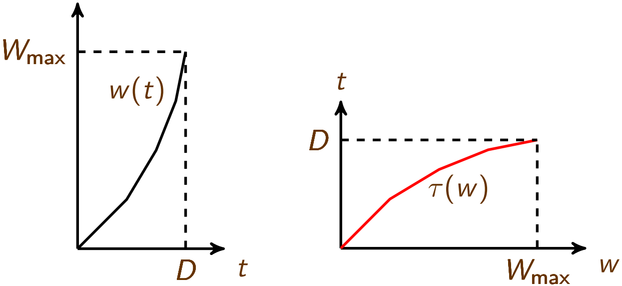
Speed selection as a function of the workload



$$\int_0^D s(t)dt = W_{max}$$

Speed selection as a function of the workload

Since the speed controller does not know the size of the task in advance, it is natural to take decisions as a function of w (the work already executed) instead as a function of time.



$$\int_0^D s(t)dt = W_{\max} \qquad \int_0^{W_{\max}} \frac{1}{s(w)} dw = D$$

Feasibility

The problem is feasible iff

$$s_{\max} \geq \frac{W_{\max}}{D}.$$

When the scheduler uses the maximal speed from the start, it must be able to complete the largest possible task $W_{\max}$ before the deadline D.

If there is no slack ($Ds_{\max} = W_{\max}$), then the unique solution is :
Use full speed from the start.

Also, we may assume w.l.o.o. that the deadline constraint is always met with equality (otherwise add an initial phase where nothing happens).

Continuous case: Mathematical Formulation

Power per work unit:

$$Q(s) := P(s)\frac{\mathrm{d}t}{\mathrm{d}w} = \frac{P(s)}{s}, \tag{1}$$

Since $P(0) = 0$ and P is convex, Q is non-decreasing.
The optimal speed is a measurable function
$s : [0, W_{\max}] \to [0, s_{\max}]$ minimizing

$$\text{Expected energy} = \int_0^{W_{\max}} \mathbb{P}(W \geq w)\, Q(s(w))\, \mathrm{d}w,$$

s.t.

$$\int_0^{W_{\max}} \frac{\mathrm{d}w}{s(w)} = D.$$

Continuous case: Mathematical Formulation

Power per work unit:

$$Q(s) := P(s)\frac{\mathrm{d}t}{\mathrm{d}w} = \frac{P(s)}{s}, \qquad (1)$$

Since $P(0) = 0$ and P is convex, Q is non-decreasing.
The optimal speed is a measurable function
$s : [0, W_{\max}] \to [0, s_{\max}]$ minimizing

$$\text{Expected energy} = \int_0^{W_{\max}} \mathbb{P}(W \geq w)\, Q(s(w))\, \mathrm{d}w,$$

s.t.

$$\int_0^{W_{\max}} \frac{\mathrm{d}w}{s(w)} = D.$$

This problem was solved by [Lorch, 2001] when $Q(s) = s^k, k > 1$ and $s_{\max} = +\infty$ using an abominable trick. In all cases, the solution uses unbounded speeds.

Constrained Optimisation: Pontryagin Principle

The Hamiltonian is $H(\tau(w), s(w), \lambda(w), w) = F^c(w)Q(s(w)) - \lambda(w)/s(w)$, where $\lambda(w)$ is the Lagrangian multiplier. The Pontryagin maximum principle says that the optimal solution s^*, λ^* satisfies

1. $H(\tau^*(w), s^*(w), \lambda^*(w), w) \leq H(\tau(w), s(w), \lambda(w), w)$ for all $w \leq W_{\max}$.
2. $\lambda^{*'}(w) = \lambda^*(w)\dfrac{\partial(1/s^*(w))}{\partial\tau} + \dfrac{\partial(F^c(w)Q(s^*(w)))}{\partial\tau}$.

Constrained Optimisation: Pontryagin Principle

The Hamiltonian is $H(\tau(w), s(w), \lambda(w), w) = F^c(w)Q(s(w)) - \lambda(w)/s(w)$, where $\lambda(w)$ is the Lagrangian multiplier. The Pontryagin maximum principle says that the optimal solution s^*, λ^* satisfies

1. $H(\tau^*(w), s^*(w), \lambda^*(w), w) \leq H(\tau(w), s(w), \lambda(w), w)$ for all $w \leq W_{\max}$.
2. $\lambda^{*'}(w) = \lambda^*(w)\dfrac{\partial(1/s^*(w))}{\partial\tau} + \dfrac{\partial(F^c(w)Q(s^*(w)))}{\partial\tau}$.

Using Eq. 2: $\lambda^*(w)$ is a constant (denoted λ^*).

Using Eq. 1: $(s^*(w))^2 Q'(s^*(w)) = \dfrac{\lambda^*}{F^c(w)}$ if it is in $[0, s_{\max}]$. Unique non-negative solution ($P(s)$ convex implies $s^2 Q'(s)$ is non-decreasing).

In the classical case ($Q(s) = Cs^2$),

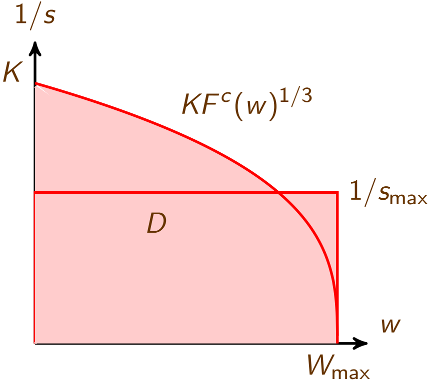
$$s^*(w) = K\, F^c(w)^{-1/3} \wedge s_{\max},$$

K is a constant s.t. $\displaystyle\int_0^{W_{\max}} 1/s^* = D$.

Pontryagin, continued

Illustration of the solution s^* for a uniform distribution of the size of the job. Here K satisfies

$$\int_0^{W_{\max}} \left(K\, F^c(w)^{1/3} \vee \frac{1}{s_{\max}} \right) dw = D.$$

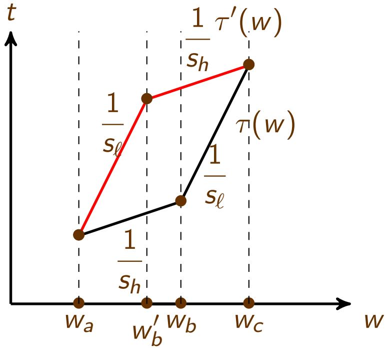


Discrete Case: A Convex Programming Solution (I)

Theorem

There exists an optimal profile that is increasing a.e.

Using Luzin's Theorem we can focus on a set of measure $1 - \varepsilon$ over which s is continuous and finish the proof by noticing that the set of bounded increasing functions is compact.



A Convex Programming Solution (II)

Focusing on increasing speed profiles, they only change at most $N-1$ times, at points $x_1, \ldots, x_{N-1}$. The problem becomes a convex program:

$$\text{Minimize} \sum_{i=1}^{N} Q(s_i) \int_{x_{i-1}}^{x_i} F^c(w)\,\mathrm{d}w \tag{2}$$

$$\text{s.t.} \sum_{i=1}^{N} \frac{x_i - x_{i-1}}{s_i} = D. \tag{3}$$

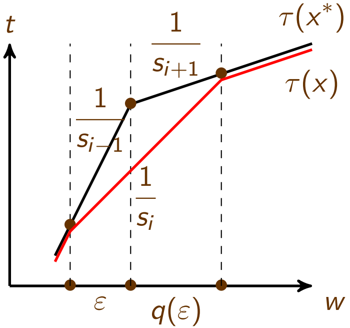
if F is continuous, then the objective function is differentiable and an optimizer can be computed by using, e.g., interior point methods (see Boyd, 2004).
Otherwise, one can rely on algorithms from derivative-free optimization theory (see Conn, Scheinberg, Vicente, 2009).
Can we do better than numerical computation via convex programming?

Structural Properties (I)

In the rest, we assume that F is continuous and P is strictly convex (both assumptions are almost always true in practice).

Proposition

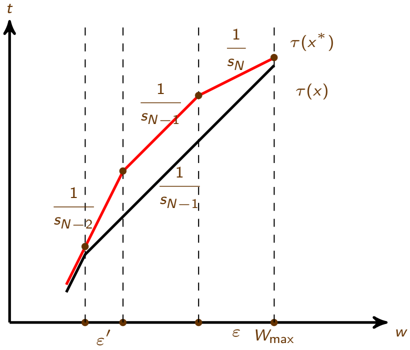
Let σ^* be an optimal schedule. Then, σ^* uses a consecutive set of speeds.



Structural Properties (II)

Proposition

The optimal schedule always uses the maximal speed v_N.



Lagrangian Relaxation

Using the multipliers $\nu \in \mathbb{R}_+^M$ and $\lambda \in \mathbb{R}_+$, the Lagrangian function is

$$L(x, \lambda, \nu) = \sum_{i=1}^{N} Q(s_i) \int_{x_{i-1}}^{x_i} F^c(w)\mathrm{d}w + \lambda \left(\sum_{i=1}^{N} \frac{x_i - x_{i-1}}{s_i} - D \right) + \sum_{i=0}^{N-1} \nu_i (x_i - x_{i+1}).$$

$$\frac{\partial L}{\partial x_i} = 0 \Leftrightarrow (Q(s_i) - Q(s_{i+1}))F^c(x_i) + \lambda \left(\frac{1}{s_i} - \frac{1}{s_{i+1}} \right) + \nu_i - \nu_{i-1} = 0.$$

Lagrangian Relaxation

Using the multipliers $\nu \in \mathbb{R}_+^M$ and $\lambda \in \mathbb{R}_+$, the Lagrangian function is

$$L(x,\lambda,\nu) = \sum_{i=1}^{N} Q(s_i) \int_{x_{i-1}}^{x_i} F^c(w)\mathrm{d}w + \lambda \left(\sum_{i=1}^{N} \frac{x_i - x_{i-1}}{s_i} - D \right) + \sum_{i=0}^{N-1} \nu_i (x_i - x_{i+1}).$$

$$\frac{\partial L}{\partial x_i} = 0 \Leftrightarrow (Q(s_i) - Q(s_{i+1}))F^c(x_i) + \lambda \left(\frac{1}{s_i} - \frac{1}{s_{i+1}} \right) + \nu_i - \nu_{i-1} = 0.$$

If only speeds $\geq s_m$ are used in the optimal solution then $\nu_i = 0$ for all $i \geq m$ by complementary slackness. The unique optimizer must satisfy for all $i \geq m$,

$$\frac{s_i P(s_{i+1}) - s_{i+1} P(s_i)}{s_{i+1} - s_i} F^c(x_i) = \frac{s_m P(s_{m+1}) - s_{m+1} P(s_m)}{s_{m+1} - s_m} F^c(x_m). \quad (4)$$

Algorithm Using Dichotomy

Computing the optimal schedule when some speeds are not used in the optimal solution;

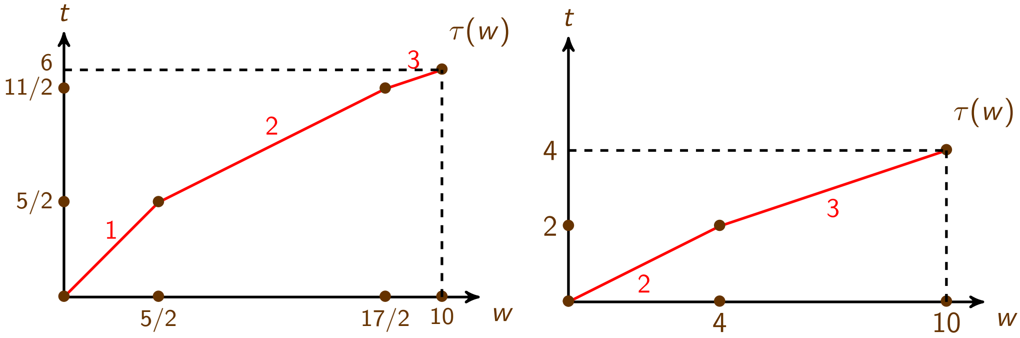
$U := N; L := 1;$
While $U > L$
do
  $m := \lfloor (U + L)/2 \rfloor;$
  Solve for x_m using speeds $s_m, \ldots, s_N$, where $x_{m+1}, \ldots, x_N$ are given by (4); (*)
  If $x_m \leq 0$ **Then** $L := m$; **Else** $U := m$;
od

Complexity: $\log_2(N)\times$ (complexity of solving one-dimensional eq. for x_m).

An explicit solution for (*) exists for specific distributions (uniform for ex.).

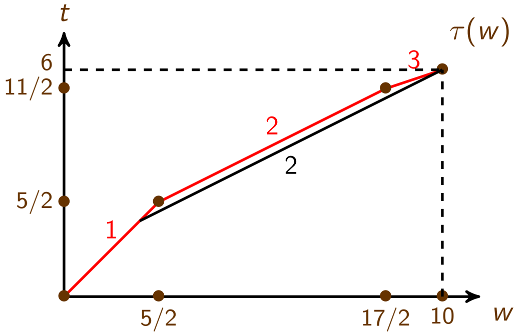
An example

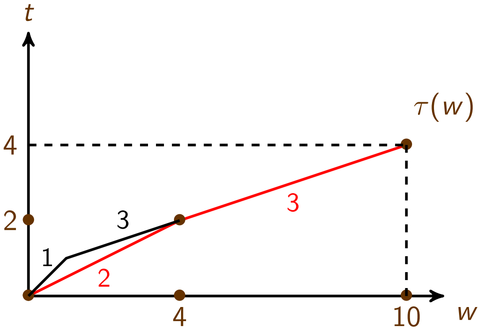
Let us consider the case where $Q(s) = s^2$, the job is uniform over $[0, 10]$ with deadline $D = 4$ or $D = 6$ and the possible speeds are $0, 1, 2, 3$.
In that case, the optimal solution is given in the following figure. (with $D = 6$ and $D = 4$ resp.)



An example

Let us consider the case where $Q(s) = s^2$, the job is uniform over $[0, 10]$ with deadline $D = 4$ or $D = 6$ and the possible speeds are $0, 1, 2, 3$.
In that case, the optimal solution is given in the following figure. (with $D = 6$ and $D = 4$ resp.)





Extensions

- Can be used for hardware design (via dynamic programming, (with E. Thierry));
- Several jobs (exponential growth, (with A. Girault & S. Plassart));
- Soft deadline alternatives (via rate decay, (with J. Anselmi & L.-S. Rebuffi)).

That's all folks!