



HAL
open science

Placement et exécution de tâches sous contraintes temporelles

Jean-Yves Baudais, Mohamad Oussayran, Jean-Christophe Prévotet, Ali Maiga

► **To cite this version:**

Jean-Yves Baudais, Mohamad Oussayran, Jean-Christophe Prévotet, Ali Maiga. Placement et exécution de tâches sous contraintes temporelles. 2023. hal-04169713

HAL Id: hal-04169713

<https://hal.science/hal-04169713>

Preprint submitted on 24 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Placement et exécution de tâches sous contraintes temporelles

Jean-YvesBaudais¹, MohamadOussayran^{1,2}, Jean-ChristophePrévolet¹, AliMaiga²

Résumé

Nous proposons une solution *offline*, analytique et algorithmique de l'exécution de tâches sur un ensemble de processeurs et sous plusieurs contraintes temporelles afin de minimiser la consommation d'énergie.

Abstract—We propose an offline, analytical and algorithmic method to minimise the energy consumption of tasks execution on multiple processors with delay constraints.

1 Introduction

Dans les systèmes embarqués temps réel, les approches DVFS (*dynamic voltage and frequency scaling*) et DPM (*dynamic power management*) sont deux techniques largement utilisées pour réduire la consommation d'énergie [1]. Schématiquement, l'approche DVFS, qui modifie la tension et la fréquence d'horloge des unités de calcul, est utilisée pour réduire la puissance dite dynamique, alors que l'approche DPM, qui bascule les unités de calcul dans un mode inactif à très basse consommation, est utilisée pour la puissance dite statique [1]. Les approches peuvent être *online*, c.-à-d. que l'optimisation des tension, fréquence et puissance est réalisée pendant l'exécution des tâches, ou *offline*. Nous nous intéressons à un système cyber-physique d'accès sécurisé composé d'un ensemble de noeuds hétérogènes, des capteurs-actionneurs effectuant également des calculs. L'ouverture d'un accès nécessite un ensemble de tâches séquentielles prédéfinies, initialement instanciées sur des noeuds et pouvant migrer sur n'importe quel noeud du réseau pour être effectuées. Ce cas d'étude nous permet de travailler dans un contexte *offline*. De nombreuses solutions algorithmes sont proposées pour réduire la consommation d'énergie [2] ou adresser plusieurs objectifs [3], et l'apprentissage machine est également appliqué [4]. Nous n'utilisons pas l'apprentissage machine mais proposons une étude analytique afin de réduire l'espace d'exploration des solutions pour, au besoin, adapter des algorithmes traitant des problèmes de complexité combinatoire [5]. Le problème traité dans cet article est le suivant.

Problème 1. *On considère la réalisation d'une application de N tâches séquentielles sur K processeurs, sous M contraintes de délais. Chaque tâche peut être exécutée avec sa propre fréquence. Quelles sont les fréquences et les affectations finales des processeurs qui minimisent la consommation d'énergie et respectent les contraintes temporelles?*

Nous commençons par définir le modèle de consommation utilisé au paragraphe 2. Le problème est résolu dans des configurations simplifiées, aux paragraphes 3, 4 et 5, avant que la solution ne soit présentée dans le cas général au paragraphe 6. L'algorithme d'optimisation, présenté au paragraphe 7, est basé sur la

méthode du point intérieur avec fonction barrière [6]. Un exemple numérique est donné au paragraphe 8. Le paragraphe 9 conclut l'article.

2 Modèle de consommation

En utilisant la dépendance linéaire entre la fréquence et la tension d'alimentation, la puissance consommée par un processeur s'écrit [7]

$$p = q + Cf^\alpha \quad (1)$$

où q est la puissance statique, Cf^α est la puissance dynamique liée à la commutation des transistors avec C la capacité équivalente des transistors, f la fréquence de fonctionnement et α un exposant de puissance dynamique compris entre 2 et 3 [7]. Les valeurs des paramètres q , C et α dépendent du système électronique considéré. La puissance statique q est fonction du mode actif ou non du processeur mais nous supposons, en première approximation, que la différence de puissance entre les modes est négligeable. Avec ce modèle de consommation, nous avons les hypothèses suivantes.

Hypothèse 2. *La consommation d'énergie de la tâche i qui s'exécute en $n_i \in \mathbb{N}^*$ cycle (ou coups d'horloge) et en un temps t_i s'écrit*

$$e_i = n_i g(f_i), \quad t_i = n_i h(f_i),$$

avec g et h des fonctions convexes, monotones, indépendantes de i , croissantes pour g et décroissantes pour h .

3 Un processeur, une contrainte

Nous commençons par traiter le problème simple d'exécution de N tâches sur un unique processeur avec une seule contrainte de temps d'exécution T . Le problème d'optimisation et sa solution sont alors les suivants.

Problème 3.

$$\min_{\{f_i\}_{i=1}^N} \sum_{i=1}^N n_i g(f_i) \quad t.q. \quad \forall i \in [1, N] \quad f_i \geq 0, \quad \sum_{i=1}^N n_i h(f_i) \leq T.$$

Proposition 4. *Sous l'hypothèse 2, la solution du problème 3 est $\forall i \in [1, N], f_i = f^*$ où f^* satisfait la contrainte de temps d'exécution.*

Démonstration. C'est un problème d'optimisation convexe. Le lagrangien s'écrit, sous forme canonique, $i \in [1, N]$,

$$\begin{aligned} L(\{f_i\}, \lambda, \{\mu_i\}) &= \sum_i e_i + \lambda \left(\sum_i t_i - T \right) - \sum_i \mu_i f_i \\ &= \sum_i n_i g(f_i) + \lambda \left(\sum_i n_i h(f_i) - T \right) - \sum_i \mu_i f_i, \end{aligned} \quad (2)$$

1. Univ Rennes, INSA Rennes, CNRS, IETR-UMR 6164, 35000 Rennes

1. FDI, 49300 Cholet

soit

$$\frac{\partial L}{\partial f_i} = n_i g'(f_i) + \lambda n_i h'(f_i) - \mu_i. \quad (3)$$

Les conditions KKT [6, § 4.4.1] pour un problème convexe conduisent à $\mu_i f_i = 0$, soit alors $\mathcal{S} = \{i | f_i > 0\}$ et ainsi

$$\forall i \in \mathcal{S}, \quad \lambda = -\frac{g'(f_i)}{h'(f_i)}. \quad (4)$$

Soit $l(x) = \frac{g'(x)}{h'(x)}$

$$l'(x) = \frac{g''(x)h'(x) - g'(x)h''(x)}{(h'(x))^2} \leq 0 \quad (5)$$

car $g'(x) \geq 0$, $g''(x) \geq 0$, $h'(x) \leq 0$ et $h''(x) \geq 0$. La fonction l est donc décroissante monotone et $l(x) + \lambda = 0$ admet au plus une solution. Si les tâches sont réalisables, alors la fréquence optimale f^* est donnée par

$$f^* = h^{-1}\left(\frac{T}{\sum_{i=1}^N n_i}\right). \quad (6)$$

Et si

$$\lim_{f \rightarrow +\infty} \sum_{i=1}^N n_i h(f) < T \quad (7)$$

alors les tâches sont toutes réalisables. \square

En pratique les dimensions du système sont telles que l'application d'accès sécurisé s'exécute correctement, ce qui nécessite que l'ensemble des tâches sont réalisables. Ainsi, la proposition 4 précise que si un ensemble de tâches doivent s'exécuter en un temps donné sur un même processeur, alors la consommation d'énergie est minimisée si toutes les tâches sont exécutées avec la même fréquence.

4 Un processeur, des contraintes

Avec M contraintes temporelles T_j , $j \in [1, M]$, le problème et sa solution sont les suivants.

Problème 5.

$$\min_{\{f_i\}_{i=1}^N} \sum_{i=1}^N n_i g(f_i) \quad t.q. \\ \forall i \in [1, N] \quad f_i \geq 0, \quad \forall j \in [1, M] \quad \sum_{i \in \mathcal{S}_j} n_i h(f_i) \leq T_j$$

Proposition 6. *Sous l'hypothèse 2, le nombre de fréquences différentes solution du problème 5 est donné par le nombre de sous-ensembles de tâches subissant le même sous-ensemble de contraintes.*

Démonstration. C'est toujours un problème d'optimisation convexe

$$L(\{f_i\}, \{\lambda_j\}, \{\mu_i\}) = \sum_{i=1}^N n_i g(f_i) + \sum_{j=1}^M \lambda_j \left(\sum_{i \in \mathcal{S}_j} n_i h(f_i) - T_j \right) - \sum_{i=1}^N \mu_i f_i, \quad (8)$$

soit

$$\frac{\partial L}{\partial f_i} = n_i g'(f_i) + \sum_{j=1}^M \lambda_j 1_{\mathcal{S}_j}(i) n_i h'(f_i) - \mu_i. \quad (9)$$

L'annulation du lagrangien s'écrit, pour des tâches réalisables,

$$\sum_{j=1}^M \lambda_j 1_{\mathcal{S}_j}(i) = -\frac{g'(f_i)}{h'(f_i)}. \quad (10)$$

S'il existe $\mathcal{A} \subset [1, N]$ tel que $\forall i \in \mathcal{A}, \sum_{j=1}^M \lambda_j 1_{\mathcal{S}_j}(i) = a$ alors $\forall i \in \mathcal{A}, f_i = f_a$, d'après la proposition 4.

Soit $V = (v_{i,j})_{i \in [1, N], j \in [1, M]}$ la matrice des contraintes, avec $v_{i,j} = 1_{\mathcal{S}_j}(i)$. Il suffit alors de compter le nombre de lignes différentes dans la matrice V pour avoir le nombre maximal de fréquences différentes. \square

5 Une tâche, des processeurs

Cette fois, la tâche peut migrer et être exécutée sur un des K processeurs. La migration a des coûts énergétique et temporel qui ne dépendent que de n_i . Il n'y a pas ici d'optimisation du débit pour le transfert des tâches et on suppose un réseau de communication homogène.

Hypothèse 7. *La consommation de la migration et de l'exécution d'une tâche i d'un processeur k sur le processeur k' à la fréquence f_i est*

$$e_i = g_0(n_i) \bar{\delta}_{kk'} + n_i g_{k'}(f_i)$$

et le temps de migration et d'exécution est

$$t_i = h_0(n_i) \bar{\delta}_{kk'} + n_i h_{k'}(f_i),$$

où $\bar{\delta}_{kk'} = 1 - \delta_{kk'}$ et $\delta_{kk'}$ est le symbole de Kronecker. Les propriétés des fonctions $g_{k'}$ et $h_{k'}$ sont données à l'hypothèse 2, les fonctions g_0 et h_0 sont monotones, croissantes.

À noter que si $k = k'$, alors il n'y a pas de migration et la consommation et le temps associés sont nuls. Dans ce cas

$$e_i = n_i g_k(f_i) \quad \text{et} \quad t_i = n_i h_k(f_i). \quad (11)$$

Avec l'hypothèse 7, le problème et sa solution sont alors les suivants.

Problème 8.

$$\min_{k' \in [1, K], f_i \geq 0} g_0(n_i) \bar{\delta}_{kk'} + n_i g_{k'}(f_i) \quad t.q. \\ \forall k' \in [1, K], \quad h_0(n_i) \bar{\delta}_{kk'} + n_i h_{k'}(f_i) \leq T.$$

Proposition 9. *Sous l'hypothèse 7, la solution du problème 8 est donnée par la comparaison des meilleures fréquences sur chaque processeur.*

Démonstration. La meilleure fréquence f_i sur le processeur k' est la fréquence solution de

$$h_{k'}(f_i) = \frac{T - h_0(n_i) \bar{\delta}_{kk'}}{n_i} \quad (12)$$

et la solution au problème est la fréquence f_i^* sur le processeur k^* telle que $\forall k' \in [1, K]$,

$$g_0(n_i) \bar{\delta}_{kk^*} + n_i g_{k^*}(f_i^*) \leq g_0(n_i) \bar{\delta}_{kk'} + n_i g_{k'}(f_i). \quad (13)$$

(8) Dans le cas où $k^* \neq k$, la fréquence solution doit vérifier $g_0(n_i) + n_i g_{k^*}(f_i^*) \leq n_i g_k(f_i)$, qui est un cas particulier de (13). \square

6 Cas général

Nous faisons les hypothèses supplémentaires suivantes pour traiter le cas général.

Hypothèse 10. *Les changements de fréquence nécessitent de l'énergie et du temps supplémentaires si deux tâches successives sont exécutées sur un même processeur*

$$\forall i \in [1, N], \begin{cases} e_i = g_f \bar{\delta}_{f_{i-1}f_i} \delta_{k_{i-1}k_i}, \\ t_i = h_f \bar{\delta}_{f_{i-1}f_i} \delta_{k_{i-1}k_i}, \end{cases}$$

avec $f_0 = k_0 = 0$.

Hypothèse 11. *Un mode veille est activé lorsqu'un processeur n'a pas de tâche à exécuter. Il est coûteux en énergie et en temps au réveil*

$$e_i = g_v \bar{\delta}_{k_{i-1}k_i}, \quad t_i = h_v \bar{\delta}_{k_{i-1}k_i}.$$

Le problème général est formalisé de la façon suivante.

Problème 12. *Soit N tâches i séquentielles initialement instantanées sur le processeur $k_i \in [1, K]$, le problème s'écrit*

$$\min_{\{f_i, k'_i\}_{i=1}^N} \sum_{i=1}^N \left(g_0(n_i) \bar{\delta}_{k_i k'_i} + n_i g_{k'_i}(f_i) + g_f \bar{\delta}_{f_{i-1}f_i} \delta_{k'_{i-1}k'_i} + g_v \bar{\delta}_{k'_{i-1}k'_i} \right),$$

t. q. $\forall j \in [1, M]$,

$$\sum_{i \in \mathcal{J}_j} \left(h_0(n_i) \bar{\delta}_{k_i k'_i} + n_i h_{k'_i}(f_i) + h_f \bar{\delta}_{f_{i-1}f_i} \delta_{k'_{i-1}k'_i} + h_v \bar{\delta}_{k'_{i-1}k'_i} \right) \leq T_j$$

et $\forall i \in [1, N]$ $f_i \geq 0$ et $k'_i \in [1, K]$.

La solution de ce problème est donnée par la proposition suivante.

Proposition 13. *Sous les hypothèses 2, 7 et 10, le nombre de fréquences différentes solutions du problème 12 est donné par le nombre de groupes de tâches subissant des jeux de contraintes différentes sur des processeurs différents.*

Démonstration. Soit une répartition donnée des tâches, c.-à-d. $\delta_{k_i k'_i}$ et $\delta_{k'_{i-1} k'_i}$ donnés. S'il n'existe pas deux tâches successives $\{i-1, i\}$ sur un même processeur ayant des contraintes différentes, alors la contribution de g_f est toujours nulle, sinon il faut distinguer le cas où les fréquences sont différentes ou pas pour dériver le lagrangien. On obtient alors, comme pour le problème 5, un système à $N+M$ inconnues $\{f_1, \dots, f_N, \lambda_1, \dots, \lambda_M\}$ et inéquations à résoudre.

Soit U le vecteur colonne de la configuration évaluée $U = (k'_i)_{i \in [1, N]}$. Le nombre maximal de fréquences différentes est donné par le nombre de lignes différentes de la matrice $[V \ U]$, avec la matrice V introduite dans la preuve de la proposition 6. Par exemple, si

$$U^T = (1 \ 1 \ 1 \ 3 \ 2 \ 3 \ 1 \ 3)$$

et

$$V^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix},$$

alors il y a 5 lignes différentes dans la matrice $[V \ U]$ et les fréquences optimales vérifient $f_1 = f_2 = f_7$ et $f_6 = f_8$. Il n'y a au plus que 5 fréquences optimales à rechercher dans cet exemple. \square

7 Algorithme

Le problème à M contraintes d'inégalité est résolu en utilisant une méthode de point intérieur à l'aide d'une fonction barrière [6, § 16.1]. Dans un premier temps, ces contraintes d'inégalité sont intégrées dans la fonction objective. On note $f = [f_1, \dots, f_N]^T$ le vecteur des fréquences des N tâches,

$$G(f) = \sum_{i=1}^N \left(g_0(n_i) \bar{\delta}_{k_i k'_i} + n_i g_{k'_i}(f_i) + g_f \bar{\delta}_{f_{i-1}f_i} \delta_{k'_{i-1}k'_i} + g_v \bar{\delta}_{k'_{i-1}k'_i} \right) \quad (14)$$

est la fonction de coût énergétique et

$$H(f) = - \sum_{j=1}^M \ln \Phi_j(f) \quad (15)$$

est la fonction barrière des contraintes, avec

$$\Phi_j(f) = T_j - \sum_{i \in \mathcal{J}_j} \left(h_0(n_i) \bar{\delta}_{k_i k'_i} + n_i h_{k'_i}(f_i) + h_f \bar{\delta}_{f_{i-1}f_i} \delta_{k'_{i-1}k'_i} + h_v \bar{\delta}_{k'_{i-1}k'_i} \right). \quad (16)$$

On peut également ajouter dans la fonction $H(f)$ les contraintes de positivité des variables. La fonction $H(f)$ s'écrit alors

$$H(f) = - \sum_{j=1}^M \ln \Phi_j(f) - \sum_{i=1}^N \ln f_i. \quad (17)$$

Le problème s'écrit

$$\min_f \beta G(f) + H(f) \quad (18)$$

avec $\beta \in \mathbb{R}_+^*$ et $\lim_{\beta \rightarrow \infty} f^*(\beta) = f^*$. La solution est recherchée en itérant β avec un pas $\mu > 1$: $\beta^{(i+1)} = \mu \beta^{(i)}$. Trouver $f^*(\beta)$ revient à résoudre

$$\beta \nabla G(f) + \nabla H(f) = 0 \quad (19)$$

où les dérivées sont calculées en considérant une configuration donnée, incluant des fréquences différentes si nécessaire. Ainsi, $\forall i \in [1, N]$, (19) s'écrit

$$\beta n_i g'_{k'_i}(f_i) + \sum_{j=1}^M \frac{1_{\mathcal{J}_j}(i) n_i h'_{k'_i}(f_i)}{\Phi_j(f)} - \frac{1}{f_i} = 0. \quad (20)$$

Pour résoudre numériquement (20), ou (18) à l'aide d'une descente de gradient, nous avons besoin de calculer la matrice hessienne de $\beta G(f) + H(f)$ dont les composantes sont

$$\frac{\partial^2 G(f)}{\partial f_i \partial f_j} = n_i g''_{k'_i}(f_i) \delta_{ij}, \quad (21)$$

$$\frac{\partial^2 H(f)}{\partial f_i^2} = \sum_{j=1}^M \frac{1_{\mathcal{J}_j}(i) \left(n_i h''_{k'_i}(f_i) \Phi_j(f) - n_i^2 h'^2_{k'_i}(f_i) \right)}{\Phi_j^2(f)} + \frac{1}{f_i^2} \quad (22)$$

et, $\forall i \neq l$,

$$\frac{\partial^2 H(f)}{\partial f_i \partial f_l} = - \sum_{j=1}^M \frac{1_{\mathcal{J}_j}(i) n_i h'_{k'_i}(f_i) 1_{\mathcal{J}_j}(l) n_l h'_{k'_l}(f_l)}{\Phi_j^2(f)}. \quad (23)$$

On note $e_U(f)$ l'énergie totale de la réalisation des tâches aux fréquences f d'une configuration $U = [k'_1, \dots, k'_N]^T$. L'algorithme 1 résout le problème 12.

Algorithme 1.

- 1: Initialiser $f^{(0)}$ réalisable, $\beta^{(0)} < 1$, $\mu > 1$, $f^* = +\infty$
- 2: **for** chaque configuration $U \in [1, K]^N$ **do**
- 3: $\forall i \in [1, N]$, calculer $\delta_{k_i k'_i}$, $\delta_{k'_{i-1} k'_i}$, δ_{f_{i-1}, f_i}
- 4: $j \leftarrow 0$
- 5: **while** conditions de convergence pas atteintes **do**
- 6: $f^{(j+1)} \leftarrow$ solution de (20), ou (18) par descente de gradient, initialisée avec $f^{(j)}$
- 7: $\beta^{(j+1)} \leftarrow \mu \beta^{(j)}$
- 8: $j \leftarrow j + 1$
- 9: **end while**
- 10: $f^* \leftarrow \min \{e_U(f^{(j)}), f^*\}$
- 11: **end for**

À noter que les fonctions barrières $\ln f_i$ (17) ne sont pas nécessaires car la recherche de zéro est initialisée avec un point réalisable $f^{(0)} > 0$ et les fonctions barrières $\ln \Phi_j$ permettent de garder les points $f^{(i)}$ dans le quadrant positif si le pas μ n'est pas trop grand. Par ailleurs, l'espace des configurations explorées $[1, K]^N$ est réduit en utilisant la proposition 13.

8 Simulations

On considère deux processeurs $K = 2$, deux contraintes $M = 2$ et quatre tâches $N = 4$, avec n_i une v.a. discrète uniforme sur $[10^5, 10^7]$, k_i une v.a. discrète uniforme sur $[1, K]$, b_i une v.a. discrète uniforme sur $[100, 1000]$, représentant les tailles des paquets transmis pour la migration des tâches, $g_0(b_i) = b_i \frac{6 \cdot 10^{-3}}{115000}$, $g_j(f_i) = 10^{-2} a_j f_i^2$ et $\{a_1, a_2\} = \{10^{-27}, 10^{-24}\}$, $g_f = 10^{-8}$, $g_v = 5 \cdot 10^{-3}$, $h_0(b_i) = \frac{b_i}{115000}$. Les processeurs ont un certain taux d'occupation x_j , $h_j(f_i) = \frac{1}{f_i(1-x_j)}$, avec $x_1 \in [0, 1[$ et $x_2 = 0$, $h_f = 10^{-8}$, $h_v = 10^{-3}$.

Les contraintes temporelles sont $\sum_{i=1}^N t_i \leq 0,3$ et $\sum_{i=1}^2 t_i \leq 0,05$ en seconde, soit

$$V^T = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}. \quad (24)$$

La figure 1 donne un exemple de consommation totale avec et sans optimisation, les 4 fréquences optimales et l'affectation finale optimale des 4 tâches sur les processeurs 1 ou 2 en fonction du taux d'occupation x_1 du processeur 1. Le vecteur d'affectation initiale est $U^T = [2 \ 1 \ 1 \ 1]$. Le processeur 1 consommant beaucoup moins que le processeur 2 à fréquence égale, pour un faible taux d'occupation l'optimum est alors d'effectuer les tâches sur ce processeur 1. À plus de 96 % de taux d'occupation, il est préférable de ne plus transférer la tâche 1, puis les autres tâches à plus de 97 % et 98 %.

9 Conclusion

Nous proposons une stratégie *offline* d'allocation optimale des fréquences et processeurs pour minimiser la consommation d'énergie. Elle peut avantageusement être appliquée dans un système cyber-physique d'accès contrôlé où la même application est répétée de nombreuses fois.

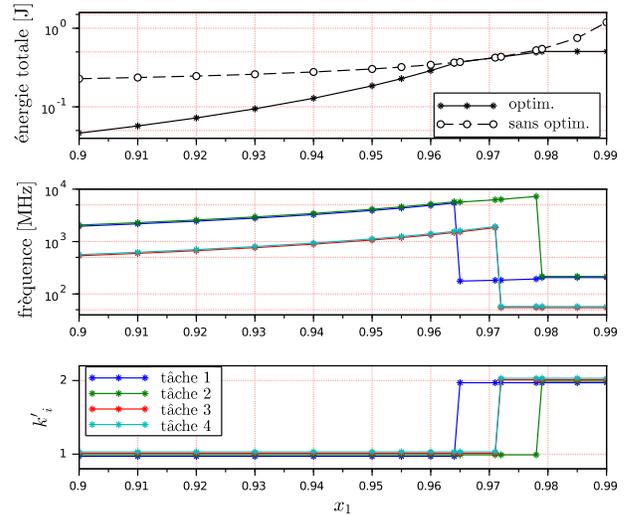


FIGURE 1 – Consommation en fonction du taux d'occupation.

Références

- [1] M. BAMBAGINI, M. MARINONI, H. AYDIN et G. BUTTAZZO, « Energy-aware scheduling for real-time systems: A survey », *ACM Transactions on Embedded Computing Systems*, vol. 15, p. 1–34, janvier 2016.
- [2] V. DEVADAS et H. AYDIN, « On the interplay of voltage/frequency scaling and device power management for frame-based real-time embedded applications », *IEEE Transactions on Computers*, vol. 61, p. 31–44, janvier 2012.
- [3] M. GADOU, S. MOGILI, T. BANERJEE et S. RANKA, « Multi-objective optimization on dvfs based hybrid systems », in *International Green and Sustainable Computing Conference*, (Alexandria, VA, U.S.A.), p. 1–7, octobre 2019.
- [4] C. ZHUO, D. GAO, Y. CAO, T. SHEN, L. ZHANG, J. ZHOU et X. YIN, « A dvfs design and simulation framework using machine learning models », *IEEE Design & Test*, vol. 40, p. 52–61, février 2023.
- [5] N. MAZYAVKINA, S. SVIRIDOV, S. IVANOV et E. BURNAEV, « Reinforcement learning for combinatorial optimization: A survey », *Computers & Operations Research*, vol. 134, p. 1–24, 2021.
- [6] J. GILBERT, « Fragments d'optimisation différentiable : théories et algorithmes ». INRIA, septembre 2021. Notes de cours.
- [7] D. ZHU et H. AYDIN, « Reliability-aware energy management for periodic real-time tasks », in *Real Time and Embedded Technology and Applications Symposium*, p. 225–235, avril 2007.