On Multiprocessor Temperature-Aware Scheduling Problems

Evripidis Bampis, Dimitrios Letsios, Giorgio Lucarelli, Evangelos Markakis, Ioannis Milis

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
CNRS UMR 7243

CAHIER DU LAMSADE

310

Mai 2011

On Multiprocessor Temperature-Aware Scheduling Problems

Evripidis Bampis, Dimitrios Letsios, Giorgio Lucarelli, Evangelos Markakis and Ioannis Milis

On Multiprocessor Temperature-Aware Scheduling Problems

Evripidis Bampis*†
LIP6, Université Pierre et Marie Curie, France
Evripidis.Bampis@lip6.fr

Dimitrios Letsios
IBISC, Université d' Évry, France
dimitris.letsios@ibisc.univ-evry.fr

Giorgio Lucarelli*
LAMSADE, Université Paris-Dauphine, France
lucarelli@lamsade.dauphine.fr

Evangelos Markakis and Ioannis Milis
Dept. of Informatics, Athens University of Economics and Business, Greece
{markakis,milis}@aueb.gr

Abstract

We study temperature-aware scheduling problems under the model introduced by Chrobak et al. in [6]. We consider a set of parallel identical processors and three optimization criteria: makespan, maximum temperature and (weighted) average temperature. On the positive side, we present polynomial time approximation algorithms for the minimization of the makespan and the maximum temperature, as well as, optimal polynomial time algorithms for minimizing the average temperature and the weighted average temperature. On the negative side, we prove that there is no $(\frac{4}{3} - \epsilon)$-approximation algorithm for the problem of minimizing the makespan for any $\epsilon > 0$, unless $\mathcal{P} = \mathcal{NP}$.

1 Introduction

The exponential increase in the processing power of recent (micro)processors has led to an analogous increase in the energy consumption of computing systems of any kind, from compact mobile devices to large scale data centers. This has

*Research supported by the French Agency for Research under the DEFIS program TODO, ANR-09-EMER-010.
†GDR Recherche Opérationnelle du CNRS.

also led to vast heat emissions and high temperatures affecting the processors' performance and reliability. Moreover, high temperatures reduce the lifetime of chips and may permanently damage the processors. For this reason, manufacturers have set appropriate thresholds in processors' temperature and use cooling systems working almost permanently. However, the energy consumption and heat emission of these cooling systems have to be added to that of the whole system.

The issues of the energy and thermal management, in the (micro)processor and system design levels, date back to the first computer systems. During the last few years these issues have been also addressed at the operating system's level, generating new interesting questions. In this context the operating system has to decide the order in which the jobs should be scheduled so that the system's temperature (and/or energy consumption) remains as low as possible, while at the same time some standard user or system oriented criterion (e.g. makespan, response time, throughput, etc) is optimized. Clearly, the minimization of the temperature and the optimization of the scheduling criteria are, in general, in conflict. Towards this direction several models have been proposed in the literature. A first model is based on the speed-scaling technique for energy saving and the Newton's law of cooling; see for example [4, 3] as well as recent reviews on speed scaling in [9, 1, 2]. In another model proposed in [12], a thermal RC circuit is utilized to capture the temperature profile of a processor.

In this paper we adopt the simplified model for cooling and thermal management introduced by Chrobak et al. in [6], which has been motivated by [11]. We consider a set of unit-length jobs (corresponding to slices of the processes to be scheduled), each one of a given heat contribution, and model the thermal behavior of the system as follows: If a job of heat contribution h is executed on a processor in a time interval $[t-1, t)$, $t \in \mathbb{N}$, and the temperature of the processor at time $t-1$ is Θ, then the processor's temperature at time t is $\frac{\Theta+h}{c}$, where c is a given cooling factor. We consider two natural variants of the model:

- the *threshold thermal model* in which a given threshold on the temperature of the processors cannot be violated. This makes necessary the introduction of idle times in a schedule.
- the *optimization thermal model* in which there is no explicit upper bound on the temperature of the processors. The lack of such an explicit bound is counterbalanced by the fact that the minimization of the (maximum or average) temperature becomes the goal of the scheduler.

The constraints that are introduced by such temperature management models give rise to interesting and technically challenging scheduling problems, which is the focus of our work. In particular, our goal is to schedule a set of jobs on a set of m parallel identical processors so as to minimize (i) the makespan in the threshold thermal model and (ii) the maximum or average temperature in the optimization thermal model.

Related results and our contribution. In [6], Chrobak et al. consider the threshold thermal model with $c = 2$ and a given temperature threshold θ. They

study the problem of scheduling a set of unit-length jobs with release dates and deadlines on a single processor so as to maximize the throughput, i.e. the number of jobs that meet their deadlines, without exceeding the temperature threshold θ at any time $t \in \mathbb{N}$. Extending the well-known three-field notation for scheduling problems, this problem is denoted as $1|r_i, p_i = 1, h_i| \sum U_i$. They prove that this problem is NP-hard even for the special case when all jobs are released at time 0 and their deadlines are equal, i.e. $1|p_i = 1, d_i = d, h_i| \sum U_i$. Furthermore, they study the on-line version of the throughput maximization problem in the presence of release dates and deadlines. They prove that a family of reasonable list scheduling algorithms, including *coolest first* and *earliest deadline first* algorithms, have a competitive ratio of at most two. In the negative side, they also give an instance that shows that there is no deterministic on-line algorithm with competitive ratio less than two. This result implies also an approximation factor of two for the off-line problem.

The analysis of the family of reasonable algorithms of [6] is generalized by Birks et al. in [5] for other values of c. In fact, for $c > 2$ they claim a competitive ratio of 2 and for $1 < c < 2$ they give a tight competitive ratio that tends to infinity as c tends to 1. Moreover, they present non-constant competitive ratios for the cases of multiple processors and weighted throughput.

We initiate the study of three additional optimization criteria under the model of [6] and under the optimization thermal model. Furthermore, we study all criteria for the case of multiple processors, unlike [6]. In Section 3 we address the problem of minimizing the schedule length (makespan) in the threshold thermal model ($P|p_i = 1, h_i, \theta|C_{max}$). We prove that this problem cannot be approximated within a factor less than 4/3 and we present a $\frac{7}{3} - \frac{1}{3m}$ approximation algorithm, where m is the number of processors. For the case of a single processor, this yields a 2 approximation. In Sections 4 and 5 we move to the optimization thermal model. In Section 4, we study the problem of minimizing the maximum temperature of a schedule ($P|p_i = 1, d_i = d, h_i|\Theta_{max}$), and we give a 4/3 approximation algorithm. In Section 5, we prove that the problem of minimizing the average temperature of a schedule ($P|p_i = 1, d_i = d, h_i| \sum \Theta_i$), as well as a time-dependent weighted version of this problem are both solvable in polynomial time. In order to avoid trivial solutions for the maximum and average temperature objectives, we assume a given common deadline for all jobs in the instances of these problems (see Section 2). We conclude in Section 6.

2 Notation and Preliminaries

We consider a set $J = \{J_1, J_2, \ldots, J_n\}$ of n jobs to be executed on a system of m identical processors. All jobs have unit processing times and for each one of them we are given a heat contribution h_i, $1 \leq i \leq n$. We consider each job J_i executed in a time interval $[t-1, t)$, $t \in \mathbb{N}$, which we call (time) slot t, on some processor. By Θ_t we denote the temperature of a processor at time t. As in [6] we consider the cooling factor to be $c = 2$ and, therefore, if we start

executing job J_i at time $t-1$, then $\Theta_t = \frac{\Theta_{t-1}+h_i}{2}$. The initial temperature of each processor (the ambient temperature) is considered to be zero, i.e., $\Theta_0 = 0$.

The threshold thermal model. In this model, the temperature is not allowed to exceed a threshold θ at any time $t \in \mathbb{N}$. It is clear that, for a given instance in this model, a feasible schedule may exist only if $h_i \leq 2 \cdot \theta$ for each job J_i. By normalizing the values of h_i's and θ we can assume w.l.o.g. that $0 < h_i \leq 2$ and $\theta = 1$. Moreover, if a processor at time $t-1$ has temperature Θ_{t-1} and it holds that $\frac{\Theta_{t-1}+h_i}{2} > 1$, for every job J_i that has not yet been scheduled, then this processor will remain idle for the slot $[t-1, t)$ and its temperature at time t will be reduced by half, i.e., $\frac{\Theta_{t-1}}{2}$. Note also that once a processor has executed some job(s) its temperature will never become exactly zero. Therefore, in this model, a feasible instance can not contain more than m jobs of heat contributions equal to 2, as there are m slots with $\Theta_0 = 0$ (the first slots in each one of the m available processors).

The optimization thermal model. In this model, no explicit temperature threshold is given and the problems we will study are the minimization of maximum and average temperature. For any instance in this model, any schedule of length at least $\lceil \frac{n}{m} \rceil$ is feasible, independently of the range of the jobs' heat contributions. However, the optimum value of our objectives depends on the time available to execute the given set of jobs: the maximum or average temperature of a schedule of length $\lceil \frac{n}{m} \rceil$ is, clearly, greater than that of a schedule of bigger length, where we are allowed to introduce idle slots. Due to this fact, we introduce in the instances of our problems a common deadline $d \geq \lceil \frac{n}{m} \rceil$ for all the jobs and we ask for a schedule that minimizes our objectives within this deadline. For these problems, we also consider only instances with $n = m \cdot d$, since for instances with $n < md$ we can simply add $md - n$ fictive jobs of heat contribution equal to zero. Thus, the number of jobs equals the total number of available slots of the m processors.

We close this section by elaborating on the complexity of the problems studied in the rest of the paper. It is already mentioned in [6] that even for a single processor, the NP-hardness of the maximum throughput problem for the case where jobs are released at time 0 and all deadlines are equal $(1|p_i = 1, d_i = d, h_i, \theta| \sum U_i)$ implies the NP-hardness of the makespan minimization problem $(1|p_i = 1, h_i, \theta|C_{max})$. In fact, the decision version of the latter problem asks for the existence of a feasible schedule where all jobs complete their execution by some given deadline C. Moreover, the decision version of the maximum temperature problem for a single processor $(1|p_i = 1, d_i = d, h_i|\Theta_{max})$ asks for the existence of a schedule where all jobs complete their execution by some given deadline d without exceeding a given temperature threshold θ. Therefore, the same reduction gives NP-hardness for both makespan and maximum temperature minimization problems. The NP-hardness for our problems on an arbitrary number of parallel processors follows trivially.

3 Makespan Minimization

In this section we study the makespan minimization under the threshold thermal model, that is $P|p_i = 1, h_i, \theta|C_{\max}$.

We start with a negative result on the approximability of our problem. The proof of the next theorem is based on adapting a reduction given in [6] for the complexity of throughput maximization under the same model.

Theorem 1. *It is NP-hard to approximate the minimum makespan problem ($P|p_i = 1, h_i, \theta|C_{\max}$) within a factor better than 4/3.*

Proof. We give a reduction from Numerical 3-Dimensional Matching (N3DM), where given three sets A, B, C of n integers each and an integer β, we ask for n triples $(a, b, c) \in A \times B \times C$ such that each integer belongs to exactly one triple and $a + b + c = \beta$ for each triple. Wlog, we assume that $\sum_{x \in A \cup B \cup C} x = \beta n$ and $x \leq \beta$ for each $x \in A \cup B \cup C$. Given instance of N3DM, we construct an instance of $P|p_i = 1, h_i, \theta|C_{\max}$ consisting of n processors and $3n$ jobs, one for each integer in $A \cup B \cup C$. Considering the function $f(x) = \frac{1}{25}\left(1 + \frac{x}{8\beta}\right)$, we set $h(a) = 8f(a) + 1$ for each $a \in A$, $h(b) = 4f(b) + 1$ for each $b \in B$ and $h(c) = 2f(c) + 1$ for each $c \in C$.

The hardness of approximation is obtained by the following claim:

Claim 1. *There is a* N3DM *if and only if there is a feasible schedule for $P|p_i = 1, h_i, \theta|C_{\max}$ of length three.*

Proof. ($\Rightarrow$) Assume that there is a solution for N3DM. For the i-th triple (a_i, b_i, c_i), $1 \leq i \leq n$, in this solution, we schedule in the i-th processor the jobs corresponding to a_i, b_i and c_i in the first, second and third slots, respectively. For the temperatures, $\Theta_{a_i}, \Theta_{b_i}, \Theta_{c_i}$, of the i-th processor after each one of those executions we have

$$\begin{aligned}
\Theta_{a_i} &= \tfrac{8f(a_i)+1}{2} \leq \tfrac{8f(\beta)+1}{2} = \tfrac{\frac{8}{25}\left(1+\frac{\beta}{8\beta}\right)+1}{2} = \tfrac{34}{50} \leq 1 \\
\Theta_{b_i} &= \tfrac{8f(a_i)+1}{4} + \tfrac{4f(b_i)+1}{2} = \tfrac{3}{4} + 2\left(\tfrac{1}{25}\left(1 + \tfrac{a_i}{8\beta}\right) + \tfrac{1}{25}\left(1 + \tfrac{b_i}{8\beta}\right)\right) \\
&\leq \tfrac{3}{4} + \tfrac{4}{25} + \tfrac{\beta}{100\beta} = \tfrac{92}{100} \leq 1 \\
\Theta_{c_i} &= \tfrac{8f(a_i)+1}{8} + \tfrac{4f(b_i)+1}{4} + \tfrac{2f(c_i)+1}{2} \\
&= \tfrac{7}{8} + \tfrac{1}{25}\left(1 + \tfrac{a_i}{8\beta}\right) + \tfrac{1}{25}\left(1 + \tfrac{b_i}{8\beta}\right) + \tfrac{1}{25}\left(1 + \tfrac{c_i}{8\beta}\right) \leq \tfrac{7}{8} + \tfrac{3}{25} + \tfrac{\beta}{200\beta} = 1
\end{aligned}$$

and hence there is a feasible schedule of length three.

($\Leftarrow$) Assume, now, that there is a feasible schedule of length three. In this schedule there are exactly three jobs in each processor, since there are $3n$ jobs in total.

If a job corresponding to an integer $a \in A$ is scheduled to the second slot of a processor, then the temperature threshold $\theta = 1$ is violated after the third slot of this processor. Indeed the temperature at this slot will be at least

$$\tfrac{2f(0)+1}{8} + \tfrac{8f(0)+1}{4} + \tfrac{2f(0)+1}{2} = \tfrac{7}{8} + \tfrac{1}{25}\left(1 + \tfrac{0}{8\beta}\right)\left(\tfrac{2}{8} + \tfrac{8}{4} + \tfrac{2}{2}\right) = \tfrac{201}{200} > 1.$$

In a similar way, we can show that a job corresponding to an integer $a \in A$ cannot be scheduled to the third slot of a processor:

$$\frac{2f(0)+1}{8} + \frac{2f(0)+1}{4} + \frac{8f(0)+1}{2} = \frac{7}{8} + \frac{1}{25}\left(1 + \frac{0}{8\beta}\right)\left(\frac{2}{8} + \frac{2}{4} + \frac{8}{2}\right) = \frac{213}{200} > 1.$$

Hence, each of the n jobs corresponding to one of the n integers $a \in A$ is scheduled to the first slot of a processor. Moreover, we can show that a job corresponding to an integer $b \in B$ cannot be scheduled to the third slot of a processor:

$$\frac{8f(0)+1}{8} + \frac{2f(0)+1}{4} + \frac{4f(0)+1}{2} = \frac{7}{8} + \frac{1}{25}\left(1 + \frac{0}{8\beta}\right)\left(\frac{8}{8} + \frac{2}{4} + \frac{4}{2}\right) = \frac{201}{200} > 1.$$

In all, in each processor exactly three jobs are scheduled: a job $a \in A$ in the first slot, a job $b \in B$ in the second slot, and a job $c \in C$ in the third slot. Therefore, the jobs of a processor correspond to a feasible triple for N3DM.

To finish our proof, we have to show that each triple sums up to β. If this does not hold then there is a triple (a, b, c) for which $a + b + c > \beta$, since $\sum_{x \in A \cup B \cup C} x = \beta n$. The temperature of the third slot of the processor in which the corresponding jobs to this triple are scheduled is

$$\frac{8f(a)+1}{8} + \frac{4f(b)+1}{4} + \frac{2f(c)+1}{2} = \frac{7}{8} + \frac{1}{25}\left(3 + \frac{a+b+c}{8\beta}\right) > \frac{7}{8} + \frac{1}{25}\left(3 + \frac{\beta}{8\beta}\right) > 1,$$

which is a contradiction that there is a feasible schedule. □

This completes the proof of Theorem 1 since an approximation ratio better than 4/3 would be able to decide the N3DM problem. □

In what follows in this section, we present an approximation algorithm for the minimum makespan problem. Note that, in order to respect the temperature threshold, a schedule may have to contain idle slots. To argue about the number of idle slots that are needed before the execution of each job, we will introduce first an appropriate partition of the set of jobs according to their heat contribution. In particular, for each $k \geq 0$, we can argue separately for jobs whose heat contribution belongs to the interval $(\frac{2^k-1}{2^{k-1}}, \frac{2^{k+1}-1}{2^k}]$; recall that $h_i \leq 2,\ 1 \leq i \leq n$. Moreover, the interval to which a job of heat contribution h_i belongs to is denoted by k_i, that is

$$k_i = \max\{k \mid h_i > \tfrac{2^k-1}{2^{k-1}}\}.$$

Our algorithm is based on the following proposition that bounds the number of idle slots needed before the execution of a job in a feasible schedule.

Proposition 1. *Any schedule in which every job J_i is executed after at least k_i idle slots is feasible.*

Proof. Let x be the number of idle slots that precede the execution of job J_i and Θ be the temperature of the processor before the first of these slots. This temperature becomes $\frac{\Theta}{2^x}$, after the last of these slots, and $\frac{\frac{\Theta}{2^x}+h_i}{2}$ after the execution of job J_i. Such a schedule is feasible if $\Theta \leq 1$ and $\frac{\frac{\Theta}{2^x}+h_i}{2} \leq 1$, that is

$2^x \geq \frac{\Theta}{2-h_i}$. Hence, for $\Theta = 1$ and $h_i > \frac{2^{k_i}-1}{2^{k_i-1}}$, it follows that $2^x > \frac{1}{2-\frac{2^{k_i}-1}{2^{k_i-1}}} = 2^{k_i-1}$, that is $x \geq k_i$. □

The algorithm works as follows: it first sorts the jobs in a non-increasing order of their heat contributions, $h_1 \geq h_2 \geq \ldots \geq h_n$, and schedules the m hottest of them to the first slot of each processor. Then, each one of the remaining jobs, $J_i,\ m+1 \leq i \leq n$, is scheduled by leaving before its execution *exactly* k_i idle slots, according to the Proposition 1. Essentially, our problem, for these $n-m$ coolest jobs, is transformed to an instance of the classical $P||C_{\max}$ scheduling where the processing time of such a job, equals the minimum number of idle slots of Proposition 1 plus its original unit processing time, i.e., $p_i = k_i + 1$. Then, these jobs are scheduled using the standard Longest Processing Time (LPT) rule. The LPT rule simply assigns the next job (in the non-increasing order of their processing times) to the first available processor.

Algorithm MAX_C

1: Sort the jobs in non-increasing order of their heat contributions: $h_1 \geq h_2 \geq ... \geq h_n$;
2: **for** $i = 1$ to m **do**
3: Schedule job J_i to the first slot of processor i;
4: **for** $i = m+1$ to n **do**
5: Let $p_i = k_i + 1$ be the processing time of job J_i;
6: Complete the schedule by running LPT for $J_{m+1}, J_{m+2}, \ldots, J_n$;

To analyze our Algorithm MAX_C, we first need a bound on the optimal makespan. The following proposition provides a lower bound to the number of slots that are required between two jobs, both of heat contributions greater than one, in an optimal schedule.

Proposition 2. *In an optimal schedule, between the execution on the same processor of jobs J_j and J_i of heat contributions $h_j, h_i > 1$, there are at least $k_i - 1$ slots, which are either idle or execute jobs of heat contribution at most one.*

Proof. Let Θ_t be the temperature of the processor before executing J_j. Next, after the execution of J_j we have $\Theta_{t+1} = \frac{\Theta_t + h_j}{2}$. Then, after x slots (idles or executing jobs of heat contribution $h \leq 1$) we get a temperature $\Theta_{t+x+1} \geq \frac{\Theta_t+h_j}{2} \cdot \frac{1}{2^x}$. In order J_i to be executed in the next slot, it should hold that $\Theta_{t+x+1} + h_i \leq 2$, that is $2^x \geq \frac{\Theta_t+h_j}{2(2-h_i)}$. Since, $\Theta_t \geq 0$, $h_j > 1$ and $h_i > \frac{2^{k_i}-1}{2^{k_i-1}}$ we get $2^x \geq \frac{\Theta_t+h_j}{2(2-h_i)} > \frac{1}{2(2-\frac{2^{k_i}-1}{2^{k_i-1}})} = \frac{1}{\frac{2}{2^{k_i-1}}} = 2^{k_i-2}$, that is $x \geq k_i - 1$. □

In the next lemma we obtain a lower bound on the optimal makespan, denoted by OPT, by bounding, using the previous proposition, the number of slots required for executing all the jobs.

Lemma 1. *For the optimal makespan it holds that* $OPT \geq 1 + \frac{\sum_{i=m+1}^{n}(k_i+1)}{2m}$, *where the jobs are indexed in non-increasing order of their heat contributions, i.e.,* $h_1 \geq h_2 \geq \ldots \geq h_n$.

Proof. A first trivial lower bound on the optimal makespan follows by considering all jobs requiring a single time slot for their execution, that is $OPT \geq \frac{n}{m}$.

However, this bound is too weak if there are jobs requiring a number of idle slots before their execution accordingly to Proposition 2. To take into account this fact, we obtain a second bound by considering only the jobs of heat contribution $h_i > 1$. Let A be the subset of these jobs assigned to any, but the first, slot of any processor and $L(A)$ be the length of the optimal schedule for these jobs, that is $OPT = 1 + L(A)$.

By Proposition 2, each job $J_i \in A$, requires at least k_i slots to be executed (one of them for its own execution). As jobs of heat contribution $h_i \leq 1$ have $k_i = 0$, it follows that

$$OPT = 1 + L(A) \geq 1 + \frac{\sum_{J_i \in A} k_i}{m} \geq 1 + \frac{\sum_{i=m+1}^{n} k_i}{m}.$$

Combining the two bounds given above by the standard average argument we get

$$OPT \geq \frac{m+\sum_{i=m+1}^{n} k_i + n}{2m} = 1 + \frac{\sum_{i=m+1}^{n}(k_i+1)}{2m}.$$

□

Theorem 2. *Algorithm* MAX_C *achieves an approximation ratio of* $\frac{7}{3} - \frac{1}{3m}$ *for the minimum makespan problem.*

Proof. Our proof follows the standard analysis given in [8], for the classical multiprocessor scheduling problem. Note, however, that for each job J_i not executed in the first slot of any processor, the Algorithm MAX_C devotes $k_i + 1$ slots, while the bound of Lemma 1 counts k_i slots for such a job of heat contribution greater than one.

Let J_ℓ be the job which finishes last in the schedule provided by Algorithm MAX_C. This job will start executed not later than $1 + \frac{\sum_{i=m+1, j\neq\ell}^{n}(k_i+1)}{m}$, and hence for the length, C, of the schedule provided by Algorithm MAX_C it holds that

$$C \leq 1 + \frac{\sum_{i=m+1, j\neq\ell}^{n}(k_i+1)}{m} + (k_\ell + 1) = 1 + \frac{\sum_{i=m+1}^{n}(k_i+1)}{m} + \left(1 - \frac{1}{m}\right)(k_\ell + 1).$$

Hence, by Lemma 1 we get $C \leq 2OPT - 1 + \left(1 - \frac{1}{m}\right)(k_\ell + 1)$.

If $k_\ell \leq OPT/3$, then the theorem follows directly.

If $k_\ell > OPT/3$, then we consider the subinstance, I', of the original problem that contains only the jobs of heat contribution at least h_ℓ, i.e., $J' = \{J_1, J_2, \ldots, J_\ell\}$. Obviously, $k_1 \geq k_2 \geq \ldots \geq k_\ell > \frac{OPT}{3} \geq 1$. Moreover, for the length of an optimal schedule, $OPT(I')$, of the subinstance I' it holds that $OPT(I') \leq OPT$ and the lengths of the schedules returned by Algorithm MAX_C for instances I and I' are equal, i.e., $C(I') = C$. Hence, $\frac{C}{OPT} \leq \frac{C(I')}{OPT(I')}$.

In an optimal schedule of I' there are at most three jobs in each processor, for otherwise, if there is a processor with four assigned jobs, the length of that

schedule will be, by Proposition 2, at least $1 + 3k_\ell > OPT$, a contradiction. Hence, there are at most $\ell \leq 3m$ jobs in I'.

The Algorithm MAX_C schedules the job J_i, $1 \leq i \leq m$, to the first slot of processor i, the job J_{m+i}, $1 \leq i \leq m$, to the second slot of processor i and the job J_{2m+i}, $1 \leq i \leq \ell - 2m$, to the next available slot of processor $m - i + 1$. In this schedule there is at least one processor executing three jobs and hence $C(I') \geq 1 + 2(k_\ell + 1) \geq 5$.

Consider now the Algorithm MAX_C running on instance I', but with $p_i = k_i$ (instead of $p_i = k_i + 1$) for each job $J_i \in J'$, $m \leq i \leq \ell$. The length of this schedule, say H, is at most two time slots shorter than $C(I')$, since at most three jobs are executed in any processor and one of them is assigned to its first slot, that is $H(I') \geq C(I') - 2$.

Consider now the optimal schedule for instance I'. We denote by $A \subseteq J'$ the jobs executed in any, but the first, slot of the m processors and by $L(A)$, the length of this schedule. Since the length of the first part of that schedule does not depend on the jobs in the set $I' \setminus A$, i.e., the jobs assigned to the first slot of the m processors, we get $OPT(I') = 1 + L(A) \geq 1 + L(A')$, where $A' = \{J_{m+1}, J_{m+2}, \ldots J_\ell\}$. As the jobs in A' are less than $2m$, they are scheduled in H in an optimal way (due to LPT rule) and also using the least possible number of slots according to Proposition 2, that is $OPT(I') \geq H(I')$.

Therefore,

$$\frac{C}{OPT} \leq \frac{C(I')}{OPT(I')} \leq \frac{C(I')}{H(I')} \leq \frac{C(I')}{C(I')-2} = 1 + \frac{2}{C(I')-2} \leq 1 + \frac{2}{3} = \frac{5}{3},$$

and the proof is completed. □

For the case of a single processor the approximation ratio of Algorithm MAX_C becomes equal to two. Moreover, in this case, a stronger result holds.

Theorem 3. *Any algorithm which executes the hottest job in an instance of the $1|p_i = 1, h_i, \theta|C_{\max}$ problem, in the first slot of the schedule achieves a 2-approximation ratio.*

Proof. For the case of a single processor, the lower bound for the optimal schedule proved in Lemma 1 becomes $OPT \geq 1 + \frac{\sum_{i=2}^{n}(k_i+1)}{2}$. Using Proposition 1, the length, C, of the schedule of an algorithm that executes the hottest job in the first slot is at most $1 + \sum_{i=2}^{n}(k_i + 1)$. Therefore, $\frac{C}{OPT} \leq \frac{1+\sum_{i=2}^{n}(k_i+1)}{1+\frac{\sum_{i=2}^{n}(k_i+1)}{2}} \leq 2$. □

4 Maximum Temperature Minimization

Now, we turn our attention to the optimization thermal model and to the problem of minimizing the maximum temperature, i.e., $P|p_i = 1, d_i = d, h_i|\Theta_{max}$. Recall that as we discussed in Section 2, we consider a common deadline $d \geq \lceil \frac{n}{m} \rceil$ for all jobs and that $n = m \cdot d$, by adding the appropriate number of fictive jobs. By Θ^*_{max} we denote the maximum temperature of an optimal schedule.

We start with the observation that any algorithm for this problem achieves a 2 approximation ratio. Indeed, it holds that $\Theta^*_{\max} \geq h_{max}/2$, no matter how

we schedule the job of maximum heat contribution. It also holds that for any algorithm, $\Theta_{\max} \leq h_{max}$, with $\Theta_{\max}$ being the maximum temperature of the algorithm's schedule. Therefore, $\Theta_{\max} \leq 2 \cdot \Theta^*_{\max}$.

To improve this trivial ratio we propose the following algorithm which is based on the intuitive idea of alternating the execution of hot and cool jobs.

Algorithm MAX_T

1: Sort the jobs in non-increasing order of their heat contributions: $h_1 \geq h_2 \geq ... \geq h_n$;
2: Using the order of Step 1, schedule the $\lceil \frac{d}{2} \rceil m$ hottest jobs to the *odd* slots of the processors using Round-Robin;
3: Using the reverse order of Step 1, schedule the $\lfloor \frac{d}{2} \rfloor m$ coolest jobs to the *even* slots of the processors using Round-Robin;

To elaborate a little more on how the algorithm works, note that processor 1 will be assigned the job J_1, followed by J_n, then followed by J_{m+1}, and then by J_{n-m} and this alternation of hot and cool jobs will continue till the end of the schedule. Similarly processor 2 will de assigned the jobs J_2, J_{n-1}, J_{m+2}, J_{n-m-1}, and so on. The schedule is illustrated further in Table 1.

1	J_1	J_n	J_{m+1}	J_{n-m}	J_{2m+1}	...
2	J_2	J_{n-1}	J_{m+2}	J_{n-m-1}	J_{2m+2}	...
...	...	...	...	...	...	...
m	J_m	J_{n-m+1}	J_{2m}	J_{n-2m+1}	J_{3m}	...

Table 1: The schedule produced by Algorithm MAX_T.

To analyze the Algorithm MAX_T, we start with the proposition below, which is implied by the Round-Robin scheduling of jobs in its Steps 2 and 3.

Proposition 3. *In the schedule returned by Algorithm* MAX_T*:*
(i) A job J_i, $i \geq \left(\lfloor \frac{d}{2} \rfloor + 1\right) m + 1$, is succeeded by the job $J_{n-i+m+1}$.
(ii) A job J_i, $m+1 \leq i \leq \lceil \frac{d}{2} \rceil m$, is preceded by the job $J_{n-i+m+1}$.

The maximum temperature may appear at various points of the schedule of Algorithm MAX_T. The next lemma states that one of these points satisfies a certain property regarding the heat contribution of the job executed right before.

Lemma 2. *In the schedule returned by Algorithm* MAX_T*, the maximum temperature is achieved after the execution of a job J_i, with $i \leq \left(\lfloor \frac{d}{2} \rfloor + 1\right) m$.*

Proof. Assume that all the points where the maximum temperature $\Theta_{\max}$ occurs are after the execution of a job J_i, with $i \geq \left(\lfloor \frac{d}{2} \rfloor + 1\right) m + 1$. By Proposition 3, such a job is succeeded by a job $J_{i'}$, $i' = n - i + m + 1$, in the schedule

returned by Algorithm MAX_T. It is easy to check that $i > i'$, hence $h_{i'} \geq h_i$. Let $\Theta, \Theta' \leq \Theta_{\max}$ be the temperatures before the execution of J_i and after the execution of $J_{i'}$, respectively. Then, $\Theta_{\max} = \frac{\Theta + h_i}{2}$ and $h_i \geq \Theta_{\max}$, since $\Theta_{\max} \geq \Theta$. Moreover, $\Theta' = \frac{\Theta_{\max} + h_{i'}}{2} \geq \Theta_{\max}$, since $h_{i'} \geq h_i$. This implies that $\Theta' = \Theta_{\max}$, since $\Theta' \leq \Theta_{\max}$. But this means that the maximum temperature is also achieved after the execution of job $J_{i'}$, which is a contradiction because

$$i' = n - i + m + 1 \leq m(d - \lfloor \tfrac{d}{2} \rfloor) \leq m \left(\lfloor \tfrac{d}{2} \rfloor + 1 \right)$$

contrary to what we assumed in the beginning of the proof. □

Lemma 3. *For the maximum temperature of an optimal schedule it holds that* $\Theta^*_{max} \geq \frac{h_{n-i+m+1}}{4} + \frac{h_i}{2}$*, for any* i*,* $m + 1 \leq i \leq \lceil \frac{d}{2} \rceil m$*.*

Proof. Consider a job J_i and let $J_{i'}$ be its previous job in the same processor in an optimal schedule S^*. The jobs executed in the first slot of each processor in S^* do not have a previous one. To simplify the presentation of our proof, we assume that they are preceded by hypothetical jobs J_{n+j}, $1 \leq j \leq m$.

If $i' \leq n - i + m + 1$, then $\Theta^*_{max} \geq \frac{h_{i'}}{4} + \frac{h_i}{2} \geq \frac{h_{n-i+m+1}}{4} + \frac{h_i}{2}$, since $h_{i'} \geq h_{n-i+m+1}$.

If $i' > n - i + m + 1$, then let $B = \{J_{n-i+m+2}, J_{n-i+m+3}, \ldots, J_n, J_{n+1}, \ldots, J_{n+m}\}$ and let A be the set of jobs that precede the jobs $J_1, J_2, \ldots, J_{i-1}$ in the optimal schedule. Clearly, $|B| = |A| = i - 1$, $J_{i'} \in B$ and $J_{i'} \notin A$ since $J_{i'}$ precedes J_i in S^*.

Therefore, there is a job $J_{k'} \in A$ such that $J_{k'} \notin B$, that is $k' < n-i+m+2$. For any $i \leq \lceil \frac{d}{2} \rceil m$, the job $J_{k'}$ precedes a job J_k in S^* and since $J_{k'} \in A$ it follows, by the definition of the set A, that $k < i$. Hence, $\Theta^*_{max} \geq \frac{h_{k'}}{4} + \frac{h_k}{2} \geq \frac{h_{n-i+m+1}}{4} + \frac{h_i}{2}$, since $h_k \geq h_i$ and $h_{k'} \geq h_{n-i+m+1}$. □

Theorem 4. *Algorithm* MAX_T *achieves a* $\frac{4}{3}$ *approximation ratio.*

Proof. By Lemma 2 the maximum temperature in the schedule, S, obtained by Algorithm MAX_T occurs after the execution of a job J_i, $i \leq \left(\lfloor \frac{d}{2} \rfloor + 1 \right) m$ (the maximum may be achieved in other timeslots as well).

If $1 \leq i \leq m$, then the maximum occurs at the first processor and $\Theta_{\max} = \frac{h_1}{2} \leq \Theta^*_{\max}$ and, hence, the algorithm returns an optimal schedule.

If $m + 1 \leq i \leq \lceil \frac{d}{2} \rceil m$ then by Proposition 3, the job J_i is preceded in the schedule S by the job $J_{n-i+m+1}$. Let Θ be the temperature before the execution of the job $J_{n-i+m+1}$. By Lemma 3, and since $\Theta \leq \Theta_{\max}$, $\Theta_{\max} = \frac{\Theta}{4} + \frac{h_{n-i+m+1}}{4} + \frac{h_i}{2} \leq \frac{\Theta_{\max}}{4} + \Theta^*_{\max}$. Hence, $\Theta_{\max} \leq \frac{4}{3} \cdot \Theta^*_{\max}$.

Note that if d is odd, then $\lceil \frac{d}{2} \rceil m = \left(\lfloor \frac{d}{2} \rfloor + 1 \right) m$. Hence the only remaining case is that d is even and $\lceil \frac{d}{2} \rceil m + 1 \leq i \leq \left(\lfloor \frac{d}{2} \rfloor + 1 \right) m$. For this case, let $\Theta' \leq \Theta_{\max}$ be the temperature before the execution of J_i. Then, $h_i \geq \Theta_{\max}$, since $\Theta_{\max} = \frac{\Theta' + h_i}{2}$ and $\Theta_{\max} \geq \Theta'$. Thus, there are at least $\lceil \frac{d}{2} \rceil m + 1$ jobs of heat contribution at least $\Theta_{\max}$. Note that, in any schedule, each processor can execute at most $\lceil \frac{d}{2} \rceil$ jobs without any pair of them scheduled in two consecutive slots. Hence, in an optimal schedule, there are at least two jobs J_p and J_q,

$p, q \leq i$, executed in consecutive slots in the same processor. Therefore, $\Theta^*_{\max} \geq \frac{h_p}{4} + \frac{h_q}{2} \geq \frac{\Theta_{\max}}{4} + \frac{\Theta_{\max}}{2} = \frac{3}{4} \cdot \Theta_{\max}$, that is $\Theta_{\max} \leq \frac{4}{3} \cdot \Theta^*_{\max}$. □

For the tightness of the analysis of Algorithm MAX_T consider an instance of m processors, mn^2 jobs and $d = n^2$; mn hot jobs of heat contribution $h = 2$ and $mn(n-1)$ cool jobs of heat contribution $h = \epsilon$. We consider n to be sufficiently large and that ϵ tends to 0. The algorithm in each processor alternates n hot jobs with $n-1$ cool jobs and schedules $n(n-2)+1$ cool jobs at the end. The maximum temperature of the algorithm's schedule is attained exactly after the execution of the last hot job on each processor. This job is executed at slot $2n-1$, and thus

$$\Theta_{max} = \frac{2}{2^{2n-1}} + \frac{\epsilon}{2^{2n-2}} + \frac{2}{2^{2n-3}} + \frac{\epsilon}{2^{2n-4}} + \ldots + \frac{\epsilon}{2^2} + \frac{2}{2^1} \simeq 2\frac{\frac{1}{2}}{1-\frac{1}{4}} = \frac{4}{3}.$$

On the other hand, the optimal solution alternates in each processor a hot job with $n-1$ cool jobs. The temperature before the execution of any hot job tends to zero and the maximum temperature is one.

5 Average Temperature Minimization

In this section, we look at the problem of minimizing the average temperature, that is $P|p_i = 1, d_i = d, h_i| \sum \Theta_i$, instead of the maximum temperature. We will again consider a common deadline d and assume that the number of jobs is $n = md$.

Contrary to the maximum temperature, we show that minimizing the average temperature of a schedule is solvable in polynomial time. Our algorithm is based on the following lemma.

Lemma 4. *In any optimal solution for the average temperature, jobs are scheduled in a coolest first order, i.e., for any pair of jobs J_i, J_j such that $h_i > h_j$ scheduled at slots t and t', respectively, it holds that $t' \leq t$, regardless of the processor they are assigned to.*

Proof. Consider the job J_i to be scheduled at slot t of some processor p in a schedule S. The contribution of job J_i to the temperature of the s-th slot of processor p (with $t \leq s \leq d$), is $\frac{h_i}{2^{s-t+1}}$, while this job does not affect the temperature of any other slot in any processor. Hence, the contribution of job J_i to the objective function, $\sum \Theta_i$, of schedule S is

$$\sum_{s=t}^{d} \frac{h_i}{2^{s-t+1}} = h_i \cdot \sum_{s=1}^{d-t+1} \frac{1}{2^s} = h_i \cdot \left(1 - \frac{1}{2^{d-t+1}}\right) = h_i \cdot \frac{2^{d+1} - 2^t}{2^{d+1}}.$$

Therefore, the later job J_i is scheduled, the smaller its contribution to the objective function becomes.

Assume, now, that in an optimal schedule S^* the job J_i is scheduled at slot t of some processor, while the job J_j at slot $t' > t$ in any processor. By swapping the execution of this pair of jobs the contribution of the job J_i to the objective function decreases by $h_i \cdot \frac{2^{t'} - 2^t}{2^{d+1}}$ and the contribution of job J_j increases by

$h_j \cdot \frac{2^{t'}-2^t}{2^{d+1}}$. As $h_i > h_j$, it follows that the resulting schedule contradicts the optimality of the schedule S^* and this completes the proof of the lemma. □

The previous lemma leads directly to the next simple algorithm.

Algorithm AVR_T

1: Sort the jobs in non-decreasing order of their heat contributions:$h_1 \leq h_2 \leq ... \leq h_n$;
2: According to this order schedule the jobs to processors using Round-Robin;

Algorithm AVR_T finds a schedule in $O(n \log n)$ time. The optimality of this schedule follows directly by the Round-Robin scheduling of the jobs in non-decreasing order of their heat contributions and Lemma 4.

Theorem 5. *An optimal schedule for the problem of minimizing the average temperature $(P|p_i = 1, d_i = d, h_i|\sum \Theta_i)$ can be found in polynomial time.*

5.1 Weighted Average Temperature Minimization

In what follows, we consider a time-dependent weighted version of average temperature minimization. In particular, we consider each slot of every processor to be associated with a given positive weight w_i, $1 \leq i \leq d$, and our problem is denoted as $P|p_i = 1, d_i = d, h_i|\sum w_i \cdot \Theta_i$. The weights w_i could represent the interest of the system manager to keep its processors/computers cool during specific time periods of peak loads. This leads to special, but more practical cases, of our formulation where the weights of some slots (e.g. the same or consecutive slots in all processors) could be considered equal.

To be more precise with our presentation we denote the weight of the t-th slot of processor p by w_t^p, $1 \leq t \leq d$, $1 \leq p \leq m$. Similarly with the unweighted case, we consider a job J_i of heat contribution h_i scheduled in the t-th slot of processor p in a schedule S. The contribution of this job to the weighted temperature of the s-th slot of processor p, with $t \leq s \leq d$, is $w_s^p \cdot \frac{h_i}{2^{s-t+1}}$, and this job does not affect the temperature of any other slot in any processor. Hence, the contribution of job J_i to the total weighted temperature of the schedule S is $\sum_{s=t}^{d} w_s^p \cdot \frac{h_i}{2^{s-t+1}} = h_i \cdot \sum_s^d \frac{w_s^p}{2^{s-t+1}}$. Clearly, the quantity $c_t^p = \sum_s^d \frac{w_s^p}{2^{s-t+1}}$ is a constant that depends only on the slot t of processor p and not on the job executed in this slot.

Based on this, we transform our problem to a weighted bipartite matching problem and we prove the next theorem.

Theorem 6. *The problem of minimizing the weighted average temperature $(P|p_i = 1, d_i = d, h_i|\sum w_i \cdot \Theta_i)$ is polynomially solvable.*

Proof. We transform the problem to a weighted bipartite matching problem. Consider a complete bipartite graph $G = (V, U; E)$ where the vertices in V correspond to the n jobs and the vertices in U to the $m \cdot d$ slots available in all processors. We set the weight of the edge between a job J_i and the slot t

of processor p to be equal to $h_i \cdot c_t^p$. Hence, the weight of this edge represents the contribution of job J_i to the objective function, if it is scheduled in slot t of processor p. A perfect matching in the graph G corresponds to a feasible schedule and the weight of such a matching to the value of the objective function for this schedule. Therefore, a minimum weight perfect matching corresponds to an optimal solution for our problem. Such a matching can be found in polynomial time (see for example [7]). □

6 Conclusions

We have provided algorithms as well as negative results for various optimization criteria in scheduling under thermal management models. There are many interesting open questions remaining, the major ones being to resolve the approximability status of minimizing the makespan and of minimizing the maximum temperature. Towards a different direction, one can also consider other objectives under the threshold thermal model, in line with the objectives that have been studied in the more traditional models of job scheduling. Resolving these questions seems technically more challenging than the classic scheduling problems due to the different nature of the constraints that are introduced by temperature management models. Note that scheduling problems under the threshold thermal model can be seen as scheduling problems with sequence-dependent setup times; such a setup time for a job corresponds to the idle slots required to respect the temperature threshold. In this context (see for example [10]), the set-up time of a job usually depends only on the job itself and the previous job in the schedule. However, in our case, the number of idle slots, required before executing a job, depends on all the jobs scheduled before as well as on their order, hence existing results from the literature cannot be applied.

References

[1] S. Albers. Energy-efficient algorithms. *Commun. ACM*, 53:86–96, 2010.

[2] S. Albers. Algorithms for dynamic speed scaling. In *STACS 2011*, volume 9 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

[3] L. Atkins, G. Aupy, D. Cole, and K. Pruhs. Speed scaling to manage temperature. In *TAPAS 2011*, volume 6595 of *LNCS*, pages 9–20. Springer, 2011.

[4] N. Bansal, T. Kimbrel, and K. Pruhs. Speed scaling to manage energy and temperature. *J. ACM*, 54(1):Article 3, 2007.

[5] M. Birks and S. P. Y. Fung. Temperature aware online scheduling with a low cooling factor. In *TAMC 2010*, volume 6108 of *LNCS*, pages 105–116. Springer, 2010.

[6] M. Chrobak, Ch. Dürr, M. Hurand, and J. Robert. Algorithms for temperature-aware task scheduling in microprocessor systems. In *AAIM 2008*, volume 5034 of *LNCS*, pages 120–130. Springer, 2008.

[7] H. N. Gabow. A scaling algorithm for weighted matching on general graphs. In *FOCS 1985*, pages 90–100. IEEE Computer Society, 1985.

[8] R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math.*, 17:416–426, 1969.

[9] S. Irani and K. R. Pruhs. Algorithmic problems in power management. *ACM SIGACT News*, 36:63–76, 2005.

[10] M. Pinedo. *Scheduling: Theory, Algorithms and Systems.* Prentice-Hall, 1995.

[11] J. Yang, X. Zhou, M. Chrobak, Y. Zhang, and L. Jin. Dynamic thermal management through task scheduling. In *ISPASS 2008*, pages 191–201. IEEE Computer Society, 2008.

[12] S. Zhang and K. S. Chatha. Approximation algorithm for the temperature-aware scheduling problem. In *ICCAD 2007*, pages 281–288. IEEE Press, 2007.