Lower Bounds & Competitive Algorithms for Online Scheduling of Unit-Size Tasks to Related Machines*

[Extended Abstract] †

Spyros Kontogiannis
Max-Planck-Institut für Informatik,
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
spyros@mpi-sb.mpg.de

ABSTRACT

In this paper we study the problem of assigning unit-size tasks to related machines when only limited online information is provided to each task. This is a general framework whose special cases are the classical multiple-choice games for the assignment of unit-size tasks to identical machines. The latter case was the subject of intensive research for the last decade. The problem is intriguing in the sense that the natural extensions of the greedy oblivious schedulers, which are known to achieve near-optimal performance in the case of identical machines, are proved to perform quite poorly in the case of the related machines.

In this work we present a rather surprising lower bound stating that any oblivious scheduler that assigns an arbitrary number of tasks to n related machines would need $\Omega\left(\frac{\log n}{\log\log n}\right)$ polls of machine loads per task, in order to achieve a constant competitive ratio versus the optimum offline assignment of the same input sequence to these machines. On the other hand, we prove that the missing information for an oblivious scheduler to perform almost optimally, is the amount of tasks to be inserted into the system. In particular, we provide an oblivious scheduler that only uses $\mathcal{O}(\log\log n)$ polls, along with the additional information of the size of the input sequence, in order to achieve a constant competitive ratio vs. the optimum offline assignment. The philosophy of this scheduler is based on an interesting exploitation of the SLOWFIT concept ([1, 5, 3]; for a survey see [6, 9, 16]) for the assignment of the tasks to the related machines despite the restrictions on the provided online information, in combination with a layered induction argument for bounding the tails of the number of tasks passing from slower to faster machines. We finally use this oblivious scheduler as the core of an adaptive scheduler that does not demand the knowledge of the input sequence and yet achieves almost the same performance.

Keywords

Limited Information, Online Load Balancing, Related Machines.

*This work has been partially supported by the IST Program of the European Union under contract number IST-1999-14186 (ALCOM-FT)

†A full version of this paper is available at `www.mpi-sb.mpg.de/~spyros/`

STOC'02 May 19-21, 2002, Montreal, Quebec, Canada.

1. INTRODUCTION

The problem of the `Online Assignment of Tasks to Related Machines` is defined as follows: there are n machines possibly of different speeds, that are determined by a **speed vector** $\overline{\mathbf{c}} \equiv (\mathcal{C}_1, \ldots, \mathcal{C}_n) \in \mathbb{R}_+^n$, and an input sequence of m independent tasks to be assigned to these machines. The tasks arrive sequentially, along with their associated **weights** (positive reals) and have to be assigned immediately and uniquely to the machines of the system. The size of the input sequence as well as the weights of the tasks are determined by an oblivious adversary (denoted here by $\mathcal{ADV}$). Each task has to be assigned upon its arrival to one of the machines, using the following information:

- (possibly a portion of) the **online** information of current status of the system,
- the **offline** information of the machine speeds, and
- its own **weight**.

The tasks are considered to be of infinite duration (permanent tasks) and no preemption is allowed. The cost of an online scheduler $\mathcal{ALG}$ for the assignment of an input sequence of tasks σ (denoted by $\mathcal{ALG}(\sigma)$) is the maximum load eventually appearing in the system. In case that a randomized scheduler is taken into account, then the cost of the scheduler is the expectation of the corresponding random variable. The quality of an online scheduler is compared vs. the optimum offline assignment of the same input sequence to the n machines. We denote the optimum offline cost for σ by $\mathcal{ADV}(\sigma)$. That is, we consider the competitive ratio (or performance guarantee) to be the quality measure, (eg, see [6]):

DEFINITION 1.1. *An online scheduler $\mathcal{ALG}$ is said to achieve a* **competitive ratio of parameters** (a, β), *if for any* *input sequence* σ *the relation connecting its own cost* $\mathcal{ALG}(\sigma)$ *and the optimum offline cost of* $\mathcal{ADV}$*, are related by*

$$\mathcal{ALG}(\sigma) \leq a \cdot \mathcal{ADV}(\sigma) + \beta.$$

$\mathcal{ALG}$ *is* **strictly** a**-competitive** *if*

$$\forall \sigma,\ \mathcal{ALG}(\sigma) \leq a \cdot \mathcal{ADV}(\sigma).$$

In this work we study the consequences of providing only some portion of the online information to a scheduler. That is, we focus our interest on the case where each task is capable of checking the online status only by a (small wrt n) number d of polls from the n machines. In this case, the objective is to determine the trade-off between the number of polls that are available to each of the tasks and the performance guarantee of the online scheduler, or equivalently, to determine the minimum number of polls per task so that a **strictly constant competitive** ratio is achieved.

Additionally, we consider the case of unit-size tasks that are assigned to related machines. Thus, each task $t \in [m]$ has to be assigned to a machine $host(t) \in [n]$ using the following information that is provided to it: the current loads of d suitably chosen machines (the kind of the "suitable" selection is one of the basic elements of a scheduler and will be called the **polling strategy** from now on) and an **assignment strategy** that determines the host of t among these d selected candidates on behalf of t.

In what follows we shall consider **homogeneous** schedulers, ie, schedulers that apply exactly the same protocol on all the tasks that are inserted into the system. This choice is justified by the fact that no task is allowed to have access to knowledge concerning previous or forthcoming choices of other tasks in the system, except only for the current loads of those machines that have been chosen to be its candidate hosts. Additionally, we shall use the terms **(capacitated) bins** instead of (related) machines and **(identical) balls** instead of (unit-size) tasks interchangeably, due to the profound analogy of the problem under consideration with the corresponding `Balls & Bins` problem.

1.1 Polling Strategies

The way a scheduler $\mathcal{ALG}$ lets each newly inserted task choose its d candidate hosts is called a **polling strategy (PS)**. We call the strategies that poll candidate machines homogeneously for all the inserted tasks of the same size, **homogeneous polling strategies (HPS)**. In the present work we consider the tasks to be indistinguishable: Each task upon its arrival knows only the loads of the machines that it polls, along with the speed (or equiv. capacity wrt bins) vector $\overline{\mathbf{c}}$ of the system. This is why we focus our interest in schedulers belonging to HPS. Depending on the dependencies of the polls that are taken on behalf of a task, we classify the polling strategies as follows:

Oblivious polling strategies(HOPS)

In this case we consider that the polling strategy on behalf of a newly inserted task t consists of an independent (from other tasks) choice of a d-tuple from $[n]^d$ according to a fixed probability distribution $f : [n]^d \to [0, 1]$. This probability distribution may only depend on the common offline information provided to each of the tasks. It should be clear that any kind of d independent polls (with or without replacement) on behalf of each task, falls into this family of polling strategies. Thus the whole polling strategy is the sequence of m d-tuples chosen independently (using the same probability distribution f) on behalf of the m tasks that are to be inserted into the system. Clearly for any polling strategy belonging to HOPS, the d random polls on behalf of each of the m tasks could have been fixed prior to the start of the assignments.

Adaptive polling strategies (HAPS)

In this case the i^{th} poll on behalf of ball $t \in [m]$ is allowed to exploit the information gained by the $i-1$ previous polls on behalf of the same ball. That is, unlike the case of HOPS where the choice of d candidates of a task was oblivious to the current system state, now the polling strategy is allowed to direct the next poll to specific machines of the system according to the outcome of the polls up to this point. In this case all the polls on behalf of each task have to be taken at runtime, upon the arrival of the task.

Remark: It is commented that this kind of polling strategies are not actually helpful in the case of identical machines, where HOPS schedulers achieve asymptotically optimal performance (see [18]). Nevertheless, we prove here that this is not the case for the related machines. It will be shown that oblivious strategies perform rather poorly in this setting, while HAPS schedulers achieve actually asymptotically optimal performance.

1.2 Assignment Strategies

Having chosen the d-size set of candidate hosts for task $t \in [m]$, the next thing is to assign this task to one of these machines given their current loads and possibly exploiting the knowledge on the way that they were selected. We call this procedure the **assignment strategy**. The significant question that arises here is the following: *Given the polling strategy adopted and the knowledge that is acquired at runtime by the polled d-tuple on behalf of a task* $t \in [m]$*, which would be the optimal assignment strategy for this task so that the eventual maximum load in the system be minimized?*

In the `Unit Size Tasks-Identical Machines` case, when each of the d polls is chosen iur (with replacement) from $[n]$, Azar et al. ([2]) show that the best assignment strategy is the MINIMUM LOAD rule and requires $\mathcal{O}(\log n)$ polls per task for a strictly constant competitive ratio. Consequently Vöcking ([18]) has suggested the ALWAYS GO LEFT strategy, which (in combination with a properly chosen oblivious polling strategy) only requires a total number of $\mathcal{O}(\log\log n)$ in order to achieve a constant competitive ratio. In the same work it was also shown that one should not expect much by exploiting possible dependencies of the polls in the case of unit-size tasks that are placed into identical machines, since the load of the fullest machine is roughly the same as the one achieved in the case of non-uniform but independent polls using the ALWAYS GO LEFT rule.

Nevertheless, things are quite different in the `Related Machines` case: we show by our lower bound (section 3) that even if a scheduler $\mathcal{ALG}$ considers any oblivious polling strategy and the best possible assignment strategy, $\mathcal{ALG}$ has a strict competitive ratio of at least $\frac{2\sqrt[d]{n}}{4d-2}$, where d is the number of oblivious polls per task. This implies that in the case of the related machines there is still much space for the adaptive polling strategies until the lower bound of $\Omega(\log\log n)$ polls per task is matched.

1.3 Related Work

In the case of assigning unit-size tasks to identical machines, there has been a lot of intensive research during the last decade. If each task is capable of viewing the whole status of the system upon its arrival (we call this the `Full Information` case), then Graham's greedy algorithm assures a competitive ratio that asymptotically tends to $2 - \frac{1}{n}$ ([6]). Nevertheless, when the tasks are granted only a limited number of polls, things are much more complicated: In the case of unit-size tasks and a single poll per task, the result of Gonnet [10] has proved that for $m \approx n$ the maximum load is $(1 + o(1))\frac{\ln n}{\ln\ln n}$ when the poll of each task is chosen iur from the n machines, whp.[1] In [15] an explicit form for the expected maximum load is given for all combinations of n and m. From this work it easily seen that for $m \geq n \ln n$, the maximum load is $\frac{m}{n} + \Theta(\sqrt{m \ln n/n})$, which implies that by means of competitive ratio, $m \approx n$ is actually the hardest instance.

In the case of $d \geq 2$ polls per task, a bunch of new techniques had to be applied for the analysis of such schedulers. The main tools used in the literature for this problem have been the layered induction, the witness tree argument and the method of fluid models (a comprehensive presentation of these techniques may be found in the very good survey of Mitzenmacher et al. [14]). In the seminal paper of Azar Broder Karlin and Upfal [2] it was proved that the proposed scheduler ABKU that chooses each of the polls of a task iur from $[n]$ and then assigns the task to the candidate machine of minimum load, achieves a maximum load that is at most $\frac{m}{n} + \frac{\ln\ln n}{\ln d} \pm \Theta(1)$. This implies a strictly $\mathcal{O}\left(\frac{\ln\ln n}{\ln d}\right)$-competitive ratio, or equivalently, at least $\mathcal{O}(\ln n)$ polls per task would be necessary in order to achieve a strictly constant competitive ratio. In [18] the ALWAYS GO LEFT algorithm was proposed, which assures a maximum load of at most $\frac{m}{n} + \frac{\ln\ln n}{d \ln 2} \pm \Theta(1)$ and thus only needs an amount of $\mathcal{O}(\log\log n)$ polls per task in order to achieve a strictly constant competitive ratio. In addition it was shown that this is the best possible that one may hope for in the case of assigning unit-size tasks to identical machines with only d (either oblivious or adaptive) polls per task.

The `Online Assignment of Tasks to Related Machines` problem has been thoroughly studied during the past years for the `Full Information` case (eg, see chapter 12 in [6]). In particular, it has been shown that a strictly (small) constant competitive ratio can be achieved using the SLOWFIT-like algorithms that are based on the idea of exploiting the least favourable machines (this idea first appeared in [17]). The case of `Limited Information` has attracted little attention up to this point: some recent works ([12, 13, 8]) study the case of each task having a single poll, for its assignment to one of the (possibly related) machines when the probability distributions of the tasks comprise a Nash Equilibrium. For example, in [8] it was shown that in the `Related Machines` case a coordination ratio (ie, the ratio between the worst possible expected maximum load among Nash Equilibria over the offline optimum) of $\mathcal{O}\left(\frac{\log n}{\log\log\log n}\right)$. However, when all the task weights are equal then it was shown by Mavronicolas and Spirakis [13] that the coordination ratio is $\mathcal{O}\left(\frac{\log n}{\log\log n}\right)$. As for the case of $d > 1$ in the `Related Machines` problem, up to the author's knowledge this is the first time that this problem is dealt with.

[1]A probabilistic event A is said to hold **with high probability** (whp) if for some arbitrarily chosen constant $\gamma > 0$, $\mathbb{P}[A] \geq 1 - n^{-\gamma}$.

1.4 New results

In this work we show that any HOPS scheduler requires at least $\mathcal{O}\left(\frac{\log n}{\log\log n}\right)$ polls in order to achieve a strictly constant competitive ratio vs. an oblivious adversary. The key point in this lower bound argument is the construction of a system of $d+1$ groups of machines running at the same speed within each group, while the machine speeds (comparing machines of consecutive groups) fall by a fixed factor and on the other hand the cumulative capacities of the groups are raised by the same factor. Then it is intuitively clear that any HOPS scheduler cannot keep track of the current status within each of these $d+1$ groups while having only d polls per new task, and thus it will have to pay the cost of misplacing balls in at least one of these groups. More specifically, we show the following lower bound:

THEOREM 1. *$\forall d \geq 1$, the competitive ratio of any d-HOPS scheduler is at least $\frac{2\sqrt[d]{n}}{4d-2}$.*

Then we propose a new d-HOPS scheduler $\mathcal{OBL}^*$ which, if it is fortified with the additional knowledge of the total number of tasks to be inserted, then it achieves the following upper-bound:

THEOREM 2. *Let $\ln\ln n - \ln\ln\ln n > d \geq 2$ and suppose that the size of the input sequence is given as offline information. If $\mathcal{OBL}^*$ provides each task with (at most) $2d$ polls, then it has a strict competitive ratio that drops double-exponentially with d. In particular the cost of $\mathcal{OBL}^*$ is with high probability at most*

$$\mathcal{OBL}^*(m) \leq \left[(1+o(1))8\left(\frac{n}{d2^{d+3}}\right)^{1/(2^{d+1}-1)} + 1\right]\mathcal{ADV}(m)$$

It is commented that all the schedulers for the `Identical Machines-Limited Information` case up to now used MINIMUM LOAD as the profound assignment rule. On the other hand, $\mathcal{OBL}^*$ was inspired by the SLOWFIT approaches for the `Related Machines-Full Information` problem and the fact that a greedy scheduler behaves badly in that case. Up to the author's knowledge, the idea of using the slowest machine possible first appeared in [17]. Additionally, a layered induction argument is employed for bounding the amount of tasks that flow from the slower to the faster machines in the system.[2] This then allows the use of relatively simple arguments for bounding the maximum load of the tasks that end up in a small fraction of the system that consists of the fastest machines. Clearly this upper bound is near-optimal (up to a multiplicative constant), since it matches the $\Omega(\log\log n)$ lower bound of the `Unit Size Tasks-Identical Machines` problem ([18]) which is a subcase of our setting.

Finally we propose a HAPS scheduler ($\mathcal{ADAPT}$) that combines the previous HOPS scheduler with a classical guessing

[2]Note that this does not imply preemption of tasks which is not allowed in our setting, but rather the event that a task hits slower machines that are already overloaded and thus has to assign itself to a faster machine.

argument for the cost of $\mathcal{ADV}$ and assures a cost roughly 5 times the cost of $\mathcal{OBL}^*$:

THEOREM 3. *For any input sequence σ of identical tasks that have to be assigned to n related machines using at most $2d+1$ polls per task, the cost of $\mathcal{ADAPT}$ is (whp),*

$$\mathcal{ADAPT}(\sigma) < \mathcal{O}\left[\left(\frac{n}{d2^{d+3}}\right)^{1/(2^{d+1}-1)} + 1\right]\mathcal{ADV}(\sigma)$$

2. A SIMPLE LOWER BOUND ON HOMOGENEOUS SINGLE-POLL GAMES

This section contains a simple argument for the claimed lower bound on online schedulers that devote a single poll per new task, ie, $d = 1$. Clearly by their nature these are HOPS schedulers, since there is no actual option for the assignment strategy. The proof for the lower bound of these schedulers is rather simple but it will and shed some light to the essence of the construction for the proof of the general lower bound that will follow in the next section.

Let's assume that there exists a HOPS scheduler that only uses 1 poll per task and claims to be strict a-competitive against any oblivious adversary $\mathcal{ADV}$. Initially $\mathcal{ADV}$ chooses an arbitrary real number $r \le n$ which will be fixed in the end so as to maximize the lower bound on a. Let also the variables $\mathrm{C}_{\text{total}}, \mathcal{C}^{\max}$ denote the total capacity and the maximum possible polled capacity using one poll (ie, the maximum bin capacity in this case) in the system. Consequently $\mathcal{ADV}$ uses the following system of capacitated bins so that these values are preserved:

$$\begin{aligned}\mathcal{C}_1 &= \mathcal{C}^{\max},\\ \mathcal{C}_i &= \frac{\mathcal{C}^{\max}}{r},\ i = 2, \ldots, \frac{\mathrm{C}_{\text{total}} - \mathcal{C}^{\max}}{\frac{\mathcal{C}^{\max}}{r}} + 1\ (\equiv n)\end{aligned}$$

Observe that the capacity of bin $i \ge 2$ is r times smaller than $\mathcal{C}^{\max}$, while on the other hand, the cumulative capacity of the last $n-1$ machines is $\frac{n-1}{r}$ times larger than the capacity of the largest bin in the system. Consider also the following abbreviations of probabilities and events that may occur upon the arrival of a new ball:

$$\begin{aligned}\mathcal{E}_i &\equiv \text{“bin } i \text{ is hit by a ball”}\\ P_1 &\equiv \mathbb{P}[\mathcal{E}_1],\ \bar{P}_1 \equiv 1 - P_1\end{aligned}$$

Obviously due to the assumption of a-competitiveness,

$$a \le \frac{\mathrm{C}_{\text{total}}}{\mathcal{C}^{\max}} = \frac{\mathcal{C}^{\max} + \frac{(n-1)\mathcal{C}^{\max}}{r}}{\mathcal{C}^{\max}} = 1 + \frac{n-1}{r}$$

since $\mathcal{ALG}$ could choose to assign all the incoming balls to the largest bin in the system. The question that arises is whether there exists a 1-POLL scheduler that can do better than that. We consider the following input sequences:

<u>$|\boldsymbol{\sigma}| = \mathbf{1},\ \boldsymbol{w_1 = w}$</u>:

$$\left.\begin{aligned}\mathcal{ALG}(\sigma) = \mathbb{E}[L_{\max}(\sigma)] \ge \tfrac{P_1 \cdot w}{\mathcal{C}^{\max}} + \tfrac{\bar{P}_1 \cdot w}{\frac{\mathcal{C}^{\max}}{r}}\\ \mathcal{ADV}(\sigma) = \tfrac{w}{\mathcal{C}^{\max}}\end{aligned}\right\} \overset{\text{/* } a-comp. \text{ */}}{\Longrightarrow}$$

$$a \ge P_1 + r \cdot \bar{P}_1 = r - P_1(r-1) \Rightarrow$$
$$P_1 \ge \tfrac{r-a}{r-1}$$

<u>$|\boldsymbol{\sigma}| = \boldsymbol{\infty},\ \forall \boldsymbol{t \ge 1},\ \boldsymbol{w_t = w}$</u>: In this case the loads of all the bins will tend to their expected values, and thus

$$\left.\begin{aligned}\ell_{|\sigma|}(1) = \mathbb{E}[\ell_{|\sigma|}(1)] = \tfrac{P_1 \cdot |\sigma| \cdot w}{\mathcal{C}^{\max}}\\ \mathcal{ALG}(\sigma) = \mathbb{E}[L_{\max}(\sigma)] \ge \tfrac{P_1 \cdot |\sigma| \cdot w}{\mathcal{C}^{\max}}\\ \mathcal{ADV}(\sigma) = \tfrac{|\sigma| \cdot w}{\mathrm{C}_{\text{total}}}\end{aligned}\right\} \overset{\text{/* } a-comp. \text{ */}}{\Longrightarrow}$$

$$a \cdot \tfrac{|\sigma| \cdot w}{\mathrm{C}_{\text{total}}} \ge \tfrac{P_1 \cdot |\sigma| \cdot w}{\mathcal{C}^{\max}} \Rightarrow P_1 \le \tfrac{a\mathcal{C}^{\max}}{\mathrm{C}_{\text{total}}} = \tfrac{a}{\frac{n-1}{r}+1}$$

Combining the two bounds on P_1 we get:

$$\begin{aligned}\frac{a\mathcal{C}^{\max}}{\mathrm{C}_{\text{total}}} = \frac{a}{\frac{n-1}{r}+1} &\ge P_1 \ge \frac{r-a}{r-1}\\ a \cdot r - a &\ge n - 1 + r - a\left(\frac{n-1}{r} + 1\right)\\ a\left(r + \frac{n-1}{r}\right) &\ge r + n - 1\\ a &\ge \frac{r+n-1}{r + \frac{n-1}{r}} = \frac{r^2 + n \cdot r - r}{r^2 + n - 1}\end{aligned}$$

which is maximized for $r = \sqrt{n} + 1$ and assures a lower bound on a of $\frac{\sqrt{n}}{2}$.

Remark: It is worth noting that the lower bound completely depends on the number of bins in the system, and on the ratio $r = \frac{\mathcal{C}^{\max}}{\mathcal{C}^{\min}}$ and does not depend at all on the total capacity of the system, $\mathrm{C}_{\text{total}}$.

3. THE LOWER BOUND ON MULTI-HOPS SHCEDULERS

In this section we study the behaviour of homogeneous schedulers that adopt an oblivious polling strategy (ie, the polling strategy is from HOPS) and an arbitrary assignment strategy. We call these d-HOPS schedulers, since the choice of the d candidates on behalf of each ball is done independently for each ball, according to a common probability distribution $f : [n]^d \to [0,1]$. Recall that the choice of the candidate bins for each ball is oblivious to the current system state and thus could have been fixed prior to the beginning of the assignments.

THEOREM 1. *$\forall d \ge 1$, the competitive ratio of any d-HOPS scheduler is at least $\frac{2d\sqrt[2d]{n}}{4d-2}$.*

Proof: Let $f : [n]^d \to [0,1]$ be the adopted Oblivious polling strategy by an arbitrary d-HOPS scheduler, $\mathcal{ALG}$. Assume also that $\mathcal{ALG}$ uses the best possible assignment strategy given this polling strategy, that is, each ball chooses its own candidate bins according to f and then it may assign itself to an arbitrarily chosen host among its candidates, depending on the current loads of the candidate bins. Assume also that $\mathcal{ALG}$ claims a (strict) competitive ratio a against oblivious adversaries.

As parameters of the problem we consider again the quantities $\mathrm{C}_{\text{total}} = \sum_{i=1}^{n} \mathcal{C}_i$ and $\mathcal{C}^{\max}$: the total capacity of a system of n related machines and the maximum capacity that may be returned by a single poll. We shall describe an adversary $\mathcal{ADV}$ that initially chooses an arbitrary real number $1 \le r \le n$ and then considers the system of $(d+1$ groups of) n capacitated bins that is described in Table 1. Observe that this construction preserves the following two invariants when considering two successive groups of bins $F_\kappa, F_{\kappa+1} : 1 \le \kappa \le d$:

Group of Bins	Number of Bins in group	Capacity per Bin	Cumulative Group Capacity
F_1	1	$\mathcal{C}^{\max}$	$\mathcal{C}^{\max}$
F_2	$r(r-1)$	$\mathcal{C}^{\max}/r$	$(r-1)\mathcal{C}^{\max}$
F_3	$r^3(r-1)$	$\mathcal{C}^{\max}/r^2$	$r(r-1)\mathcal{C}^{\max}$
F_4	$r^5(r-1)$	$\mathcal{C}^{\max}/r^3$	$r^2(r-1)\mathcal{C}^{\max}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
F_d	$r^{2d-3}(r-1)$	$\mathcal{C}^{\max}/r^{d-1}$	$r^{d-2}(r-1)\mathcal{C}^{\max}$
F_{d+1}	$n-1-\frac{r}{r+1}(r^{2d-2}-1)\sim n-r^{2d-2}$	$\mathcal{C}^{\max}/r^d$	$(\frac{n}{r^d}-r^{d-2})\mathcal{C}^{\max}$

Table 1: The system of ($d+1$ groups of) capacitated bins considered by $\mathcal{ADV}$ for the proof of the Lower Bound on d-HOPS schedulers.

(I1) when going from one group to its successor, the bin capacities decrease by a factor of r, and

(I2) the cumulative capacity of the first $\kappa+1$ groups is larger than the cumulative capacity of the first κ groups by a factor of r.

We shall denote by $C[F]$ the cumulative capacity of any group of bins $F\subseteq[n]$.

Remark: The preservation of invariant (I2) when $\kappa=d$ implies that $C[F_{d+1}]\geq r^{d-1}(r-1)\mathcal{C}^{\max}\Rightarrow\left(\frac{n}{r^d}-r^{d-2}\right)\mathcal{C}^{\max}\geq r^{d-1}(r-1)\mathcal{C}^{\max}\Rightarrow n\geq r^{2d}-r^{2d-1}+r^{2d-2}$.

We fortify $\mathcal{ALG}$ by allowing a perfect balance of the bins of a group F_κ whenever at least one poll on behalf of a new ball goes to a bin of this group. This is actually in order to capture the notion of the "*perfect assignment strategy given the polling strategy*" claim stated above. Clearly this does not cause any problem since we are looking for a lower bound. Because $\mathcal{ALG}$ could lock its d choices to the first d groups of the system, it is obvious that its competitive ratio a is at most $a\leq\frac{\mathrm{C_{total}}}{C[\cup_{\nu=1}^{d}F_\nu]}=\frac{n}{r^{2d-1}}+1$.

Consider now the d events $\mathcal{E}_\kappa\equiv$ "*F_κ is hit by a ball*" ($1\leq\kappa\leq d$), while $P_\kappa\equiv\mathbb{P}[\mathcal{E}_\kappa]$ (call it the **hitting probability** of group F_κ) is the probability of at least one bin from F_κ being hit by a ball. We shall charge $\mathcal{ALG}$ according to the hitting probabilities that its polling strategy determines. Notice that these are fixed at the beginning of the assignments since the polling strategy of $\mathcal{ALG}$ is an oblivious strategy. Furthermore, the following **conditional hitting probabilities** are also determined uniquely by the polling strategy of $\mathcal{ALG}$: $\forall i,j\in[n]:i>j$,

$$\begin{aligned}P_{i|j} &\equiv \mathbb{P}[\mathcal{E}_i|\mathcal{E}_1\wedge\mathcal{E}_2\wedge\cdots\wedge\mathcal{E}_j],\\ Q_{i|j} &\equiv \mathbb{P}[\mathcal{E}_i|\neg\mathcal{E}_1\wedge\neg\mathcal{E}_2\wedge\cdots\wedge\neg\mathcal{E}_j].\end{aligned}$$

Finally, let $B_\kappa(\sigma)$ ($\kappa=1,\ldots,d$) denote the maximum number of balls that may be hosted by bins of the set $\cup_{\nu=1}^{\kappa}F_\nu$ without violating the assumption of a-competitiveness of $\mathcal{ALG}$, when the input sequence of tasks σ is chosen by $\mathcal{ADV}$. The following lemma states an inherent property of any d-HOPS scheduler:

LEMMA 3.1. *For any $\delta>1$, unless $\mathcal{ALG}$ admits a competitive ratio $a>\frac{(\delta-1)(1-r^{-2})}{d\delta^2-\delta}r$, the following property holds:*

$$\forall 1\leq\kappa\leq d,\quad P_{\kappa|\kappa-1}\geq 1-\frac{\delta}{\delta-1}\cdot\frac{a}{r}$$

Proof: We prove this lemma by considering the following input sequences of balls of the same (arbitrarily chosen) size w:

$\underline{|\boldsymbol{\sigma}|=\mathbf{1}}$: In this case we know that $\mathcal{ADV}(\sigma)=\frac{w}{\mathcal{C}^{\max}}$. The cost (ie, the expectation of maximum load) of $\mathcal{ALG}$ is:

$$\begin{aligned}\mathcal{ALG}(\sigma) \;\geq\; & P_1\frac{w}{\mathcal{C}^{\max}}+\\ & (1-P_1)\left[Q_{2|1}\frac{rw}{\mathcal{C}^{\max}}+(1-Q_{2|1})Q_{3|2}\frac{r^2w}{\mathcal{C}^{\max}}+\right.\\ & \left.+\cdots+(1-Q_{2|1})\cdots(1-Q_{d|d-1})\frac{r^dw}{\mathcal{C}^{\max}}\right]\end{aligned}$$

Due to the demand for a-competitiveness of $\mathcal{ALG}$ against $\mathcal{ADV}$, this then implies $\frac{a}{r}\geq 1-P_1\Rightarrow P_1\geq 1-\frac{a}{r}$.

$\underline{|\boldsymbol{\sigma}|=\boldsymbol{r^{2\kappa-2}},\ \boldsymbol{\kappa=2,\cdots,d}}$: In this case $\mathcal{ADV}$ will use only the bins of $\cup_{\nu=1}^{\kappa}F_\nu$ and thus he will pay a cost of $\mathcal{ADV}(\sigma)=\frac{r^{2\kappa-2}w}{r^{\kappa-1}\mathcal{C}^{\max}}=\frac{r^{\kappa-1}w}{\mathcal{C}^{\max}}$. As for the cost of $\mathcal{ALG}$, we shall only charge it for the input subsequence of balls that definitely hit groups $F_1,\ldots,F_{\kappa-1}$ (call it $\hat{\sigma}$). Our purpose is from this sequence of tasks to determine $P_{\kappa|\kappa-1}$, ie, the conditional hitting probability of group F_κ given that all the previous groups are hit by a ball. Clearly,

$$\mathbb{E}[|\hat{\sigma}|]=|\sigma|\cdot\prod_{\nu=1}^{\kappa-1}P_{\nu|\nu-1}=r^{2\kappa-2}\cdot\prod_{\nu=1}^{\kappa-1}P_{\nu|\nu-1}$$

(where for symmetry of representation we let $P_{1|0}=P_1$). Recall that $B_{\kappa-1}(\sigma)$ denotes the maximum number of balls that may be assigned to the bins of the first $\kappa-1$ groups, given the claimed competitive ratio a by $\mathcal{ALG}$ and the input sequence σ. Then we have: $\frac{wB_{\kappa-1}(\sigma)}{C[\cup_{\nu=1}^{\kappa-1}F_\nu]}\leq a\cdot\mathcal{ADV}(\sigma)\Rightarrow B_{\kappa-1}(\sigma)\leq a\cdot r^{2\kappa-3}$. Thus, there is a subsequence $\tilde{\sigma}$ of $\hat{\sigma}$ that consists of those tasks which cannot be assigned to the bins of the first $\kappa-1$ groups due to the a-competitiveness constraint. All these tasks have to exploit their remaining (at most) $d-\kappa+1$ polls among the bins of $[n]\setminus\cup_{\nu=1}^{\kappa-1}F_\nu$. It is clear that $\mathcal{ALG}$ has no reason to spoil more than one poll per group due to the optimal assignment strategy that it adopts. Thus we can safely assume that there remain exactly $d-\kappa+1$ polls for the remaining groups. Obviously

$$\begin{aligned}\mathbb{E}[|\tilde{\sigma}|] &\geq \mathbb{E}[|\hat{\sigma}|]-B_{\kappa-1}(\sigma)\geq r^{2\kappa-2}\prod_{\nu=1}^{\kappa-1}P_{\nu|\nu-1}-ar^{2\kappa-3}\\ &\geq r^{2\kappa-3}\cdot(r\gamma_{\kappa-1}-a),\end{aligned}$$

where for simplification of notation we use the bounding sequence $\gamma_{\kappa-1} = \gamma_{\kappa-2} \cdot \left(1 - \frac{\delta}{\delta-1}\frac{a}{r}\right) \Rightarrow \gamma_{\kappa-1} = \left(1 - \frac{\delta}{\delta-1}\frac{a}{r}\right)^{\kappa-1}$ and $\gamma_0 = 1$. This is true because $P_1 \geq 1 - \frac{a}{r} \geq \gamma_1 = 1 - \frac{\delta}{\delta-1}\frac{a}{r}$, $\forall \delta > 1$, while we assume inductively that $\prod_{\nu=1}^{\kappa-1} P_{\nu|\nu-1} \geq \gamma_{\kappa-1}$. By showing that $P_{\kappa|\kappa-1} \geq 1 - \frac{\delta}{\delta-1}\frac{a}{r}$ we shall also have assured that $\gamma_\kappa \leq \prod_{\nu=1}^{\kappa} P_{\nu|\nu-1}$. We apply the Markov Inequality (on the complementary random variable $|\sigma| - |\tilde{\sigma}|$) to find a lower bound on the size of $\tilde{\sigma}$:

$$\forall \delta > 1, \mathbb{P}[|\tilde{\sigma}| \geq r^{2\kappa-3}(r - r\delta + r\delta\gamma_{\kappa-1} - a\delta)] \geq 1 - \frac{1}{\delta}.$$

Now it is clear that if $\mathcal{ALG}$ claims a competitive ratio

$$a \leq \frac{(\delta-1)(1-r^{-2})}{\delta^2 d - \delta} r \leq \frac{(\delta-1)\left(1 - \frac{1}{r^{2\kappa-2}}\right)}{\delta^2 \kappa - \delta} r,$$

then at least one ball of σ will belong to $\tilde{\sigma}$ with probability at least $1 - \frac{1}{\delta}$. Thus, either $\mathcal{ALG}$ has $a > \frac{(\delta-1)(1-r^{-2})}{\delta^2 d - \delta} r$, or (by simply charging it only for this very specific ball)

$$\mathcal{ALG}(\sigma) \quad \geq \quad \left(1 - \frac{1}{\delta}\right)[P_{\kappa|\kappa-1} + (1 - P_{\kappa|\kappa-1})r]\frac{r^{\kappa-1}w}{\mathcal{C}^{\max}}$$

which, combined with the demand for a-competitiveness and the cost of $\mathcal{ADV}$ for σ, implies that

$$P_{\kappa|\kappa-1} \quad \geq \quad 1 - \frac{\delta}{\delta-1}\frac{a}{r}.$$

■

We finally try the following input sequence, in case that $\mathcal{ALG}$ still claims a competitive ratio $a \leq \frac{(\delta-1)(1-r^{-2})}{d\delta^2 - \delta} \cdot r$:
$|\boldsymbol{\sigma}| = \boldsymbol{\infty}$: For this input sequence it is clear that

$$\mathcal{ADV}(\sigma) = \frac{|\sigma| w}{\mathrm{C}_{\text{total}}} = \frac{|\sigma| w}{\left(r^{d-1} - r^{d-2} + \frac{n}{r^d}\right)\mathcal{C}^{\max}}.$$

For $\mathcal{ALG}$ we again consider the subsequence $\hat{\sigma}$ of balls that definitely hit the first $d-1$ groups of the system. Clearly $|\hat{\sigma}| = \mathbb{E}[|\hat{\sigma}|] \geq \gamma_{d-1} \cdot |\sigma|$ since we now consider an infinite sequence of incoming balls. As for the upper bound on the balls that the first $d-1$ groups can host, this is again given by the demand for a-competitiveness:

$$\begin{aligned}
\frac{wB_{d-1}(\sigma)}{C[\cup_{\nu=1}^{d-1} F_\nu]} = \frac{wB_{d-1}(\sigma)}{r^{d-2}\mathcal{C}^{\max}} \quad &\leq \quad \frac{a|\sigma| w}{\left(r^{d-1} - r^{d-2} + \frac{n}{r^d}\right)\mathcal{C}^{\max}} \\
B_{d-1}(\sigma) \quad &\leq \quad \frac{r^{d-1}}{r^{d-1} - r^{d-2} + \frac{n}{r^d}} \frac{a}{r} |\sigma|
\end{aligned}$$

The subsequence $\tilde{\sigma} \subseteq \hat{\sigma}$ that has to exploit a single poll among the bins of $[n] \setminus \cup_{\nu=1}^{d-1} F_\nu$ has size at least

$$\begin{aligned}
|\tilde{\sigma}| \quad &= \quad \mathbb{E}[|\tilde{\sigma}|] \geq \mathbb{E}[|\hat{\sigma}|] - B_{d-1}(\sigma) \\
&\geq \quad \left(\gamma_{d-1} - \frac{r^{d-1}}{r^{d-1} - r^{d-2} + \frac{n}{r^d}} \frac{a}{r}\right) \cdot |\sigma| \\
&\geq \quad \left(1 - \frac{\delta(d-1)}{\delta-1}\frac{a}{r} - \frac{r^{d-1}}{r^{d-1} - r^{d-2} + \frac{n}{r^d}}\frac{a}{r}\right)|\sigma| \\
&\geq \quad \left(1 - \frac{d\delta - 1}{\delta - 1}\frac{a}{r}\right)|\sigma|
\end{aligned}$$

where for the last inequality we consider that $n \geq r^{2d-2}$. Since we consider that $a \leq \frac{(\delta-1)(1-r^{-2})}{\delta^2 d - \delta} r$, we can be sure that $|\tilde{\sigma}| \geq \left(1 - \frac{1}{\delta} + \frac{1}{\delta r^2}\right)|\sigma|$ and thus, the cost of $\mathcal{ALG}$ will be lower bounded by the expected load of the bins in F_d due to the tasks of $\tilde{\sigma}$:

$$\begin{aligned}
a \cdot \mathcal{ADV}(\sigma) \geq \mathcal{ALG}(\sigma) \quad &\geq \quad \frac{P_{d|d-1} \cdot |\tilde{\sigma}| \cdot w}{(r^{d-1} - r^{d-2})\mathcal{C}^{\max}} \\
\frac{a}{r^{d-1} - r^{d-2} + \frac{n}{r^d}} \quad &\geq \quad \frac{P_{d|d-1}\left(1 - \frac{(\delta-1)(1-r^{-2})}{\delta^2 d - \delta}\right)}{r^{d-1} - r^{d-2}} \\
\frac{1}{\left(1 - \frac{(\delta-1)(1-r^{-2})}{\delta(\delta d-1)}\right)\left(1 + \frac{n}{r^{2d-2}(r-1)}\right)} \quad &\geq \quad 1 - \frac{\delta}{\delta - 1}\frac{a}{r} \\
&\geq \quad 1 - \frac{1 - r^{-2}}{\delta d - 1}
\end{aligned}$$

which is not possible for any $\delta > 1$ and $n \geq r^{2d}$. Thus we conclude that $\mathcal{ALG}$ cannot avoid a competitive ratio

$$a \geq \min\left\{\frac{\delta - 1}{\delta(\delta d - 1)} r, \frac{n}{r^{2d-1}} + 1\right\}$$

for any $\delta > 1$ and $n \geq r^{2d}$, which for $\delta = 2$ and $n = r^{2d}$ gives the desired bound. ■

4. DEALING WITH INPUT SEQUENCES OF KNOWN TOTAL SIZE

In this section we prove that the missing information for an oblivious scheduler to perform efficiently is the size of the input sequence. More specifically, considering that the input size is provided as offline information to each of the newly inserted tasks, we construct an oblivious scheduler that exploits this information along with a SLOWFIT assignment rule and a layered induction argument for the flow of balls from slower to faster bins, in order to achieve a strictly constant competitive ratio with only $\mathcal{O}(\log\log n)$ polls per task.

Assume that m unit-size balls are thrown into a system of n capacitated bins with capacities $\mathcal{C}^{\max} = \mathcal{C}_n \geq \mathcal{C}_{n-1} \geq \cdots \geq \mathcal{C}_1 = \mathcal{C}^{\min}$. Assume also that each ball is allowed to poll up to $2d$ bins and then it has to assign itself to one of these candidates. We additionally assume that $\mathcal{C}^{\max} \leq \frac{n}{2^{d+1}}$. As it will become clear later by the analysis, if this was not the case then it could only be in favour of the oblivious scheduler that we propose, because this would allow the absorption of the large additive constants in the performance guarantee of the scheduler.

We consider (wlog) that the capacity vector $\overline{\mathbf{c}}$ is normalized by $\frac{n}{\|\overline{\mathbf{c}}\|_1}$ so that $\sum_{i=1}^{n} \mathcal{C}_i = n$. We also assume that the total size m the input sequence is given to every newly inserted ball. This implies that each ball can estimate the cost $\mathcal{ADV}(m) \equiv$ opt (ie, the optimum offline assignment of the m unit-size balls to the n capacitated bins), and thus it can know a priori the subset of bins that may have been used by $\mathcal{ADV}$ during the whole process.[3] Having this in mind, we can assume that every bin in the system is **legitimate**, that is, it might have been used by the optimum solution, otherwise we could have each ball ignore all the illegitimate bins in the system. Thus, opt $\geq \max\left\{\frac{1}{\mathcal{C}^{\min}}, \frac{m}{\mathrm{C}_{\text{total}}}\right\} = \max\left\{\frac{1}{\mathcal{C}^{\min}}, \frac{m}{n}\right\}$. Finally, we assume that each of the legitimate bins of the system gets at

[3] A bin $i \in [n]$ may have been used by $\mathcal{ADV}$ if and only if $1/\mathcal{C}_i \leq$ opt.

least one ball in the optimum offline schedule. This does not affect the performance of $\mathcal{ADV}$, while it may only deteriorate the performance of an online scheduler. Nevertheless, it assures that $\frac{m}{n} \leq \text{opt} \leq \frac{m}{n} + 1$, meaning that the fractional load on the bins is actually a good estimation of opt.

Let the load of bin $i \in [n]$ at time t (that is, right after the assignment of the t^{th} ball of the input sequence) be denoted as $\ell_t(i) \equiv \frac{q_t(i)}{\mathcal{C}_i}$, where $q_t(i)$ is the number of balls assigned to bin i up to that time. The following definition refers to the notion of **saturated** bins in the system, ie, overloaded bins wrt the designed performance guarantee of an oblivious scheduler:

DEFINITION 4.1. *A bin $i \in [n]$ is called* **saturated upon the arrival of a new ball $t \leq m$**, *if and only if it has $\ell_{t-1}(i) > a \cdot \text{opt} \equiv \Phi(a)$, where $\Phi(a)$ is called the* **designed performance guarantee** *of the oblivious scheduler.*

Let $\forall r \in [d],\ i_r \equiv \min\{i \in [n] : \sum_{j=1}^{i} \mathcal{C}_j \geq \sum_{\nu=1}^{r} \frac{n}{2^\nu}\}$. Then, we consider the following partition of the set of bins $[n]$ into $d+1$ groups of (roughly geometrically) decreasing cumulative capacities:

$$\begin{aligned} F_1 &\equiv \{1, \ldots, i_1\}, \\ F_r &\equiv \{i_{r-1}+1, \ldots, i_r\}, \quad r = 2, \ldots, d, \\ F_{d+1} &\equiv \{i_d + 1, \ldots, n\}. \end{aligned}$$

Although the cumulative capacity of group F_r may vary from $\frac{n}{2^r} - \mathcal{C}^{\max}$ to $\frac{n}{2^r} + \mathcal{C}^{\max}$, for ease of the following computations we assume that asymptotically $\forall r \in [d],\ C[F_r] \approx \frac{n}{2^r}$ and $C[F_{d+1}] \approx \frac{n}{2^d}$. We denote by $\mathcal{C}_\kappa^{\min}$ the capacity of the smallest bin in F_κ, $\forall \kappa \in [d+1]$.

We now consider the following ideal scheduler that uses an oblivious polling strategy and an assignment strategy based on the SLOWFIT rule. This scheduler (we call it $\mathcal{OBL}^*$) initially discards all the illegitimate bins in the system, using the knowledge of m. Then first it normalizes the capacity vector of the remaining bins and afterwards it considers the grouping mentioned above and adopts the following pair of strategies:

POLLING: $\forall 1 \leq r \leq d$ group F_r gets exactly 1 poll, which is chosen among the bins of the group proportionally to the bin capacities. That is, $\forall r \in [d],\ \forall i \in F_r$,

$$\mathbb{P}[\text{bin } i \in F_r \text{ is a candidate host of a ball}] \equiv \frac{\mathcal{C}_i}{C[F_r]}.$$

The remaining d polls are assigned to the bins of group F_{d+1}, either to the d fastest bins, or according to the polling strategy of ALWAYS GO LEFT, depending only on the parameters $(\overline{\mathbf{c}}, d)$ of the problem instance.[4]

ASSIGNMENT: Upon the arrival of a new ball $t \in [m]$, the smallest polled bin from $\cup_{\kappa=1}^{d} F_\kappa$ (starting from F_1, to F_2, a.s.o.) that is unsaturated gets this ball (SLOWFIT rule). In case that all the first d polls of a ball are already saturated, then this ball has to be assigned to a bin of F_{d+1} using its remaining d candidates. Within group F_{d+1}, either the MINIMUM POST LOAD rule (ie, the bin of minimum load among the d choices from F_{d+1} is chosen, taking into account also the additional load of the new ball), or the SLOWFIT rule is applied, depending on the offline parameters $(\overline{\mathbf{c}}, d)$ of the problem instance. If all the $2d$ polled bins are saturated, then the MINIMUM POST LOAD rule is applied among them. Ties are always broken in favour of smaller bins (ie, slower machines).

The following theorem gives the performance of $\mathcal{OBL}^*$, when the additional information of the input size is also provided offline:

THEOREM 2. *For $\ln\ln n - \ln\ln\ln n > d \geq 2$, when the size of the input sequence $m \geq n$ is given as offline information, $\mathcal{OBL}^*$ has a strict competitive ratio that drops double-exponentially with d. In particular the cost of $\mathcal{OBL}^*$ is (whp) at most*

$$\mathcal{OBL}^*(m) \leq \left[(1+o(1)) 8 \left(\frac{n}{d 2^{d+3}}\right)^{1/(2^{d+1}-1)} + 1\right] \mathcal{ADV}(m)$$

Proof: Let $\tau + 1 \leq m$ be the first ball in the system that hits only saturated bins by its $2d$ polls. Our purpose is to determine the value of a in the designed performance guarantee $\Phi(a)$, so that the probability of ball $\tau+1$ existing to be polynomially small. As stated before, the technique that we shall employ is a layered induction argument on the number of balls that are passed to the right of group F_r, $r \in [d]$. For the assignment of the balls that end up in group F_{d+1} we use a slightly modified version of the ALWAYS GO LEFT scheduler of [18] that gives an upper bound on the maximum load in group F_{d+1} (we denote this by $\mathcal{L}_{\max}[F_{d+1}]$). This upper bound on $\mathcal{L}_{\max}[F_{d+1}]$ holds with high probability. This assignment is only used when it produces a smaller maximum load than the brute assignment of all the balls ending up in F_{d+1} to the d fastest bins of the group.

We shall consider a notion of time that corresponds to the assignments of newly arrived balls into the system: At time $t \leq m$, the t^{th} ball of the input sequence is thrown into the system and it has to be immediately assigned to one of its $2d$ candidates.

Consider the polls on behalf of a ball to be ordered according to the groups from which they are taken. Observe then that each ball $t \leq \tau$ is assigned to the first unsaturated bin that it hits from the first d groups, or to a bin in group F_{d+1}. Thus, $\forall \kappa \in [d+1]$, each ball that has been assigned to group F_κ up to (and including) ball τ, has definitely failed to hit an unsaturated bin in all the groups $F_1, \ldots, F_{\kappa-1}$. For any ball $t \leq m$ and $\kappa \in [d]$, let $\mathcal{Q}_t(\kappa)$ denote the number of balls that have been assigned to group F_κ up to time t (ie, right after the assignment of the t^{th} ball), while $\tilde{\mathcal{Q}}_t(\kappa)$ denotes the balls that have been assigned to the right of group F_κ, that is, to bins of $[n] \setminus \cup_{\nu=1}^{\kappa} F_\nu$. Thus, $\tilde{\mathcal{Q}}_t(\kappa) = \sum_{\nu=\kappa+1}^{d+1} \mathcal{Q}_t(\nu)$. Let also $S_t(\kappa)$ denote the set of saturated bins in F_κ at time t. Then, $\forall \kappa \in [d]$, $\forall t \leq \tau$,

$$\mathbb{P}[t \text{ hits a saturated bin in } F_\kappa] = \frac{C[S_{t-1}(\kappa)]}{C[F_\kappa]} \leq \frac{C[S_\tau(\kappa)]}{C[F_\kappa]}.$$

Observe now that $\forall \kappa \in [d]$,

$$\begin{aligned} \mathcal{Q}_\tau(\kappa) &= \sum_{i \in F_\kappa} q_\tau(i) \\ &\geq \sum_{i \in S_\tau(\kappa)} \mathcal{C}_i \frac{q_\tau(i)}{\mathcal{C}_i} > C[S_\tau(\kappa)]\Phi(a) \Rightarrow \\ \frac{C[S_\tau(\kappa)]}{C[F_\kappa]} &< \frac{\mathcal{Q}_\tau(\kappa)}{\Phi(a) \cdot C[F_\kappa]} \equiv \mathcal{P}_\kappa \end{aligned} \tag{1}$$

[4]Observe that this is offline information and thus this decision can be made at the beginning of the assignment process, for all the balls of the input sequence.

Recall that up to time τ, we can be sure that $\mathcal{Q}_\tau(\kappa) \leq \Phi(a) \cdot C[F_\kappa]$ (because all these balls are assigned to unsaturated bins), which in turn assures that $\mathcal{P}_\kappa \leq 1$. Inequality (1) implies that, had we known $\tilde{\mathcal{Q}}_\tau(\kappa-1)$, then the number $\tilde{\mathcal{Q}}_\tau(\kappa)$ of balls before ball $\tau+1$ that go to the right of set F_κ would be stochastically dominated by the number of successes in $\tilde{\mathcal{Q}}_\tau(\kappa-1)$ Bernoulli trials with success probability $\mathcal{P}_\kappa$. We shall denote this number by $B(\tilde{\mathcal{Q}}_\tau(\kappa-1), \mathcal{P}_\kappa)$.[5] In the following lemma, we apply the Chernoff-Hoeffding bound on these Bernoulli trials to get an upper bound on the amount of balls that have to be assigned beyond group F_κ, for $\kappa \in [d]$, as a function of the number of balls that have been assigned beyond group $F_{\kappa-1}$:

LEMMA 4.1. *$\forall \kappa \in [d]$ and for an arbitrary constant $\gamma > 1$, with probability at least $1 - n^{-\gamma}$*

$$\tilde{\mathcal{Q}}_\tau(\kappa) \leq \max\left\{\frac{2^{\kappa+1}[\tilde{\mathcal{Q}}_\tau(\kappa-1)]^2}{\Phi(a)n}, \sqrt{2\gamma \ln n \tilde{\mathcal{Q}}_\tau(\kappa-1)}\right\}.$$

Proof: Let's assume that we already know the number $\tilde{\mathcal{Q}}_\tau(\kappa-1)$ of balls that have already failed in $F_1, \ldots, F_{\kappa-1}$. Then $\tilde{\mathcal{Q}}_\tau(\kappa)$ is stochastically dominated by the random variable $B(\tilde{\mathcal{Q}}_\tau(\kappa-1), \mathcal{P}_\kappa)$. By applying Chernoff-Hoeffding bounds ([11], p. 198) on these Bernoulli trials, we get that

$$\begin{gathered}
\mathbb{P}[B(\tilde{\mathcal{Q}}_\tau(\kappa-1), \mathcal{P}_\kappa) \geq \tilde{\mathcal{Q}}_\tau(\kappa-1) \cdot (\mathcal{P}_\kappa + \varepsilon)] \\
\leq \exp(-2\tilde{\mathcal{Q}}_\tau(\kappa-1)\varepsilon^2), \quad \forall \varepsilon > 0 \quad \Rightarrow \\
\mathbb{P}\left[B(\tilde{\mathcal{Q}}_\tau(\kappa-1), \mathcal{P}_\kappa) \geq \tilde{\mathcal{Q}}_\tau(\kappa-1) \cdot \left(\mathcal{P}_\kappa + \sqrt{\tfrac{\gamma \ln n}{2\tilde{\mathcal{Q}}_\tau(\kappa-1)}}\right)\right] \\
\leq n^{-\gamma}, \quad \forall \gamma > 0
\end{gathered}$$

where we have set $\varepsilon = \sqrt{\frac{\gamma \ln n}{2\tilde{\mathcal{Q}}_\tau(\kappa-1)}}$. But recall that $\mathcal{P}_\kappa = \frac{\mathcal{Q}_\tau(\kappa)}{\Phi(a)\cdot C[F_\kappa]}$ and also that $C[F_\kappa] \approx \frac{n}{2^\kappa}$. Thus we conclude that with probability at least $1 - n^{-\gamma}$,

$$\left.\begin{array}{c}
\tilde{\mathcal{Q}}_\tau(\kappa) \leq B(\tilde{\mathcal{Q}}_\tau(\kappa-1), \mathcal{P}_\kappa) \leq \\
\leq \tilde{\mathcal{Q}}_\tau(\kappa-1) \cdot \left(\frac{2^\kappa \mathcal{Q}_\tau(\kappa)}{\Phi(a)n} + \sqrt{\frac{\gamma \ln n}{2\tilde{\mathcal{Q}}_\tau(\kappa-1)}}\right) \\
\mathcal{Q}_\tau(\kappa) = \tilde{\mathcal{Q}}_\tau(\kappa-1) - \tilde{\mathcal{Q}}_\tau(\kappa)
\end{array}\right\} \Rightarrow$$

$$\tilde{\mathcal{Q}}_\tau(\kappa) \leq \tilde{\mathcal{Q}}_\tau(\kappa-1)\left(\frac{2^\kappa[\tilde{\mathcal{Q}}_\tau(\kappa-1) - \tilde{\mathcal{Q}}_\tau(\kappa)]}{\Phi(a)n} + \sqrt{\frac{\gamma \ln n}{2\tilde{\mathcal{Q}}_\tau(\kappa-1)}}\right)$$

$$\begin{aligned}
&\Rightarrow \tilde{\mathcal{Q}}_\tau(\kappa) \cdot \left(1 + \frac{2^\kappa \tilde{\mathcal{Q}}_\tau(\kappa-1)}{\Phi(a)n}\right) \leq \\
&\leq \quad \tilde{\mathcal{Q}}_\tau(\kappa-1) \cdot \left(\frac{2^\kappa \tilde{\mathcal{Q}}_\tau(\kappa-1)}{\Phi(a)n} + \sqrt{\frac{\gamma \ln n}{2\tilde{\mathcal{Q}}_\tau(\kappa-1)}}\right) \\
\tilde{\mathcal{Q}}_\tau(\kappa) \quad &\leq \quad \frac{2^\kappa[\tilde{\mathcal{Q}}_\tau(\kappa-1)]^2 + \Phi(a)n\sqrt{\frac{\gamma \ln n}{2}\tilde{\mathcal{Q}}_\tau(\kappa-1)}}{\Phi(a)n + 2^\kappa \tilde{\mathcal{Q}}_\tau(\kappa-1)}
\end{aligned}$$

from which we get the desired bound. ∎

Consider now the following finite sequence

$$\begin{aligned}
&\hat{\mathcal{Q}}(\kappa) = \max\left\{\tfrac{2^{\kappa+1}[\hat{\mathcal{Q}}(\kappa-1)]^2}{\Phi(a)n}, \sqrt{2\gamma \ln n \hat{\mathcal{Q}}_\tau(\kappa-1)}\right\}, \kappa \in [d] \\
&\hat{\mathcal{Q}}(0) = m
\end{aligned}$$

[5]For the integrity of the representation we set $\tilde{\mathcal{Q}}_\tau(0) = \tau \leq m-1$.

We then bound the number of balls that end up in group F_{d+1} by the d^{th} term of this sequence:

LEMMA 4.2. *With probability at least $1 - dn^{-\gamma}$, at most $\hat{\mathcal{Q}}(d)$ balls end up in group F_{d+1}.*

Proof: The proof of this lemma is relatively simple and due to lack of space it is differed to the full version of the paper. ∎

Consequently we estimate a closed form for the first terms of the bounding sequence $\langle \hat{\mathcal{Q}}(\kappa),\ \kappa = 0, \ldots, d\rangle$ that was determined earlier:

LEMMA 4.3. *The first $\log\left(\frac{\log \frac{m}{\gamma \ln n}}{\log(a/8)}\right) + 2$ terms of the bounding sequence $\langle \hat{\mathcal{Q}}(\kappa),\ \kappa = 0, \ldots, d\rangle$ are given by the closed form $\hat{\mathcal{Q}}(\kappa) = \frac{m}{2^\kappa \cdot \left(\frac{a}{8}\right)^{2^\kappa - 1}}$.*

Proof: Assume that κ^* was the first element in the sequence for which

$$\frac{2^{\kappa^*+1}[\hat{\mathcal{Q}}(\kappa^*-1)]^2}{\Phi(a)n} < \sqrt{2\gamma \ln n \hat{Q}(\kappa^*-1)} \tag{2}$$

Then, up to term κ^*-1 the sequence is dominated by the right-hand term of the above inequality, and thus, $\forall r < \kappa^*$,

$$\begin{aligned}
\hat{\mathcal{Q}}(r) &= \frac{2^{r+1}}{\Phi(a)n}[\hat{\mathcal{Q}}(r-1)]^2 = \frac{2^{r+1}}{\Phi(a)n}\left(\frac{2^r}{\Phi(a)n}\right)^2 [\hat{\mathcal{Q}}(r-2)]^4 \\
&= \left(\frac{2^{r+1}}{\Phi(a)n}\right)^{2^0}\left(\frac{2^r}{\Phi(a)n}\right)^{2^1}\left(\frac{2^{r-1}}{\Phi(a)n}\right)^{2^2}[\hat{\mathcal{Q}}(r-3)]^{2^3} \\
&= \cdots = \prod_{\nu=1}^{r}\left(\frac{2^{2^{r+2-\nu}}}{\Phi(a)n}\right)^{2^{\nu-1}}[\hat{\mathcal{Q}}(0)]^{2^r} \\
&= 2^{\sum_{\nu=1}^r (r+2-\nu)\cdot 2^{\nu-1}} \cdot \left(\frac{1}{\Phi(a)n}\right)^{\sum_{\nu=1}^r 2^{\nu-1}} \cdot [\hat{\mathcal{Q}}(0)]^{2^r} \\
&= \frac{2^{3\cdot 2^r - r - 3}}{\left(\frac{\Phi(a)n}{m}\right)^{2^r-1}} \cdot m \leq \frac{m}{2^r\left(\frac{a}{8}\right)^{2^r-1}}
\end{aligned}$$

since $\frac{\Phi(a)n}{8m} \geq \frac{a}{8}$. We now plug in this closed form for $\hat{Q}(\kappa^*-1)$ in inequality (2) to get the following:

$$\begin{aligned}
\sqrt{2\gamma \ln n} &> \frac{2^{\kappa^*+1}}{\Phi(a)n}\left(\frac{m}{2^{\kappa^*-1}\left(\frac{a}{8}\right)^{2^{\kappa^*-1}-1}}\right)^{3/2} \\
\frac{m^{3/2}}{\sqrt{2\gamma \ln n}} &< \frac{\Phi(a)n}{2^{\kappa^*+1}} 2^{\frac{3}{2}(\kappa^*-1)}\left(\frac{\Phi(a)n}{8m}\right)^{2^{\kappa^*-1}-1} \\
\sqrt{\frac{m}{2\gamma \ln n}} &< 2^{\frac{1}{2}(\kappa^*+3)}\left(\frac{a}{8}\right)^{2^{\kappa^*-1}}
\end{aligned}$$

From the above and the definition of κ^*, it is easy to see that $\kappa^*-1 = \max\left\{r \in [d] : \frac{\ln m - \ln \gamma - \ln\ln n}{\ln 2} - 4 \geq r + 2^{r-2}\ln\frac{a}{8}\right\}$. By setting $A = \log m - \frac{\ln\ln n + \ln \gamma}{\ln 2} - 4$ and $B = \ln\frac{a}{8}$ we get the solution $\kappa^* \geq A - \frac{W\left[\frac{B \ln 2}{4}\exp(A \ln 2)\right]}{\ln 2} + 1$, where $W(x)$ is the Lambert W Function ([7]). By approximating this function by $\ln x - \ln\ln x$ (since $x = \frac{B \ln 2}{4}\exp(A \ln 2) \to \infty$) we conclude that

$$\kappa^* \sim \log\left(\frac{\log \frac{m}{\gamma \ln n}}{\log(a/8)}\right) + 3$$

∎

LEMMA 4.4. *Assume that* $m \geq n$ *and* $d < \frac{\ln\ln n - \ln\ln\ln n}{(\sqrt{2}+1)\ln 2}$. *If we set* $a = 8\left(\frac{n}{d2^{d+3}}\right)^{1/(2^{d+1}-1)}$ *the cost of* $\mathcal{OBL}^*$ *is upper bounded (whp) by* $\mathcal{OBL}^*(m) \leq (1+o(1))(a+1)\mathcal{ADV}(m)$.

Proof: The cost of $\mathcal{OBL}^*$ up to time τ is upper-bounded by $\max\{\mathcal{L}_{\max}[F_{d+1}], (a+1)\cdot \text{opt}\}$ since in the first d groups no saturated bin ever gets another ball and all the bins are legitimate. We choose a so that the cost in the first d groups is at least as large as $\mathcal{L}_{\max}[F_{d+1}]$ (whp), given the upper bound $\hat{\mathcal{Q}}(d)$ on the number of balls that end up in the last group. Thus the probability of $\tau+1$ existing will be then polynomially small because at least one poll in F_{d+1} will have to be unsaturated whp. This then implies the claimed upper bound on the performance of $\mathcal{OBL}^*$. Due to lack of space, the complete proof of this lemma is presented in the full version of the paper. ∎

Combining the statements of all these technical lemmas, we conclude with the desired bound on the competitive ratio of $\mathcal{OBL}^*$. ∎

5. A COMPETITIVE HAPS SCHEDULER

In the previous section we have proposed a HOPS scheduler that is based on the knowledge of the size of the input sequence and then assures that its own performance is never worse than $(a+1)$opt, whp. In this section we propose a HAPS scheduler (call it $\mathcal{ADAPT}$) whose main purpose is to "guess" the value opt of the optimum offline cost by a classical guessing argument and then let $\mathcal{OBL}^*$ do the rest of the assignments. This approach is in complete analogy with the online schedulers of the `Related Machines-Full Information` problem (see [6], pp. 208-210). The only difference is that $\mathcal{OBL}^*$ has a performance which holds whp, and this is why the final result also holds whp.

A significant difficulty of $\mathcal{ADAPT}$ is exactly this guessing mechanism that will have to be based on the limited information provided to each of the new tasks. Our goal is not to assume that any kind of additional information (eg, global environment variables) is provided to the balls, other than the capacity vector and the current loads of each ball's candidates. $\mathcal{ADAPT}$ sacrifices one of the available polls per ball, in order to create such a good guessing mechanism. Of course, a different approach that would be based on the outcome of some of the polls (eg, a constant fraction of the polls) in order to estimate a proper online prediction of opt would be more interesting in the sense that it would not create a communication bottleneck for the tasks. Nevertheless, the purpose of $\mathcal{ADAPT}$ is mainly to demonstrate the possibility of constructing such an adaptive scheduler whose performance is close to that of $\mathcal{OBL}^*$.

Let's assume that the system now has $n+1$ capacitated bins ($\mathcal{C}^{\min} = \mathcal{C}_1 \leq \mathcal{C}_2 \leq \cdots \leq \mathcal{C}_{n+1} = \mathcal{C}^{\max}$). Assume also that each new task is provided with $2d+1$ polls. Fix a number $r > 1$. Then an r**-guessing mechanism** proceeds in stages: Stage κ ($\kappa = 0, 1, \dots$) contains a (consecutive) subsequence of tasks from the input sequence that use the same prediction $\Lambda_\kappa = r^\kappa \Lambda_0$ for their assignment. We set $\Lambda_0 = \frac{1}{\mathcal{C}^{\max}}$. The following definitions refer to the notions of **eligible** and **saturated** machines upon the arrival of a new task $t \in [m]$ into the system, that are used in a similar fashion as in [5] where the concept of eligible-saturated machines was introduced:

DEFINITION 5.1. *Suppose that task* $t \leq m$ *belongs to stage* κ. *The set of* **eligible machines for t** *is*

$$E_\kappa \equiv \left\{ i \in [n] : \frac{1}{\mathcal{C}_i} \leq \Lambda_\kappa = r^\kappa \Lambda_0 \right\},$$

while a machine $i \in [n]$ *is considered to be* **saturated upon the arrival of task t**, *if* $\ell_{t-1}(i) > a_\kappa \Lambda_\kappa = r^\kappa a_\kappa \Lambda_0$, *where* $a_\kappa = 8 \cdot \max\left\{1, \left(\frac{|E_\kappa|}{d2^{d+3}}\right)^{1/(2^{d+1}-1)}\right\}$.

Notice that the static information of $\langle (E_\kappa, a_\kappa),\ \kappa = 0, 1, \dots \rangle$ only depends on the capacity vector $\overline{\mathbf{c}}$ and the number d of polls per task. Thus it can be easily computed by each of the tasks, or alternatively it can be provided a priori to all the tasks as additional offline information.

$\mathcal{ADAPT}$ proceeds in phases and works as follows: Upon the arrival of a new ball $t \in [m]$, first the fastest bin in the system is polled and the stage $s(t)$ to which this ball belongs is determined, according to the following rule:

$$s(t) = \kappa \in \mathbb{N} : \lceil r^{\kappa-1} a_{\kappa-1} + 1 \rceil < q_{t-1}(n+1) \leq \lceil r^\kappa a_\kappa + 1 \rceil$$

(obviously for stage 0 only the second inequality must hold). The remaining $2d$ polls of task t are taken from group $E_{s(t)}$ in a fashion similar to that of $\mathcal{OBL}^*$. The assignment strategy of $\mathcal{ADAPT}$ is exactly the same with that of $\mathcal{OBL}^*$, with the only difference that whenever the first $2d$ candidates of a task are already saturated, then this task is assigned to the fastest machine in the system, $n+1$. If the latter event causes machine $n+1$ to become also saturated, then by the definition of the stages this task is the last ball of the current stage and a new stage (with an r times larger prediction) starts from the next task on.

LEMMA 5.1. *Suppose that* $\mathcal{ADAPT}$ *uses* $r = 9/4$. *Let* h *denote the stage at which the prediction* Λ_h *of* $\mathcal{ADAPT}$ *reaches* opt *for the first time. Then the last stage of* $\mathcal{ADAPT}$ *is at most* $h+1$.

Proof: The case of $h = 0$ is easy since this implies that $\text{opt} = 1/\mathcal{C}^{\max}$ and then $\mathcal{ADAPT}$ cannot be worse than $\mathcal{OBL}^*$ which (having the right prediction) succeeds to assign all the incoming tasks (but for those that might fit in machine $n+1$ without making it saturated) below $(a_0+1)\cdot\text{opt}$, whp. So let's consider the case where $h \geq 1$.

Let E^* denote the set of legitimate machines in the system (ie, $E^* = \{i \in [n+1] : 1/\mathcal{C}_i \leq \text{opt}\}$). The amount of work inserted into the system during the whole assignment process is bounded by $C[E^*]\cdot\text{opt}$. By definition of h, $r^{h-1}/\mathcal{C}^{\max} < \text{opt} \leq r^h/\mathcal{C}^{\max}$. Stage $h+1$ ends when machine $n+1$ becomes saturated. Let $W(\kappa)$ denote the total amount of work assigned during stage κ, while $R(\kappa)$ denotes the amount of remaining work at the end of stage κ. We shall prove that the amount of remaining work at the end of stage h is not enough to make $\mathcal{OBL}^*$ fail within stage $h+1$.

Observe that at the end of stage h, no machine has exceeded a load of $(a_h+1)\Lambda_h = (a_h+1)r^h/\mathcal{C}^{\max}$ (by definition of stages). On the other hand, during stage $h+1$, each of the eligible machines $i \in E_{h+1}$ needs a load of more than $a_{h+1}\Lambda_{h+1} = a_{h+1}r^{h+1}/\mathcal{C}^{\max}$ in order to become saturated for this stage. We denote the additional **free space** of bin

$i \in E_{h+1}$ by $free_i(h+1) > \left[\frac{a_{h+1}r^{h+1}}{\mathcal{C}^{\max}} - \frac{(a_h+1)r^h}{\mathcal{C}^{\max}}\right]\cdot\mathcal{C}_i$, while

$$\begin{aligned} FREE(h+1) &\equiv \sum_{i\in E_{h+1}} free_i(h+1) \\ &> \left[\frac{a_{h+1}r^{h+1}}{\mathcal{C}^{\max}} - \frac{(a_h+1)r^h}{\mathcal{C}^{\max}}\right]\cdot C[E_{h+1}] \\ &> \frac{a_{h+1}r^{h+1}}{\mathcal{C}^{\max}}\cdot\left(1-\frac{a_h+1}{r\cdot a_{h+1}}\right)C[E_{h+1}] \\ &> (a_{h+1}+1)\text{opt}C[E_{h+1}]\left(\frac{ra_{h+1}}{a_{h+1}+1}-\frac{a_h+1}{a_{h+1}+1}\right) \quad (3) \end{aligned}$$

is the cumulative free space granted to the eligible bins of the system for stage $h+1$. It only remains to prove that the amount of work that has to be dealt with by $\mathcal{OBL}^*$ using only the bins of $[n]$ is less than $FREE(h+1)/(a_{h+1}+1)$. Observe now that $\mathcal{OBL}^*$ uses the bins of E_{h+1} and has to deal with an amount of work less than $W(h+1) - free_{n+1}(h+1) < \mathtt{R}(h) - a_{h+1}\frac{r^h}{\mathcal{C}^{\max}}\left(r-\frac{a_h+1}{a_{h+1}}\right)\mathcal{C}^{\max} < C[E^*]\text{opt} - a_{h+1}(r-2)\text{opt}\mathcal{C}^{\max} < \text{opt}(C[E^*]-(r-2)a_{h+1}\mathcal{C}^{\max}) < (C[E^*]-\mathcal{C}^{\max})\cdot$ opt if we set $r \geq 17/8$. This is because $a_{h+1} \geq a_h \geq 8$ and at the end of stage $h+1$ bin $n+1$ must have already become saturated. Observe also that $E_{h+1} \supseteq E_h \supseteq E^* \setminus \{n+1\}$ by definition of h. Thus, $C[E_{h+1}] \geq C[E^*] - \mathcal{C}^{\max}$. Thus, the amount of work that has to be served by $\mathcal{OBL}^*$ during stage $h+1$ is less than opt $\cdot\, C[E_{h+1}]$. Then, setting $r = 9/4$ in inequality (3) assures that $\left(\frac{ra_{h+1}}{a_{h+1}+1} - \frac{a_h+1}{a_{h+1}+1}\right) \geq 1$ (recall that $a_{h+1} \geq a_h \geq 8$) and thus the remaining work at the end of stage h is not enough to make $\mathcal{OBL}^*$ fail. ■

The following theorem is now a straightforward consequence of the previous lemma:

THEOREM 3. *For any input sequence σ of identical tasks that have to be assigned to n related machines using at most $2d+1$ polls per task, the cost of $\mathcal{ADAPT}$ is (whp),*

$$\mathcal{ADAPT}(\sigma) < \mathcal{O}\left[\left(\frac{n}{d2^{d+3}}\right)^{1/(2^{d+1}-1)}+1\right]\cdot\mathcal{ADV}(\sigma).$$

6. CONCLUSIONS

In this work we have studied the problem of exploiting limited online information for the assignment of a sequence of unit-size tasks to related machines. We have shown that the oblivious schedulers that perform asymptotically optimally in the case of identical machines, deteriorate significantly in this case. We have then determined an adaptive scheduler that actually mimics the behaviour of an ideal oblivious scheduler, in order to achieve roughly the asymptotically optimal performance similar to the case of the identical machines.

As for further research, the issue of providing only limited information to online algorithms is critical in many problems for which an objective is also the minimization of the communication cost. In this category fall most of the network design problems. Another interesting line of research would be the avoidance of communication bottlenecks for such limited information online algorithms. Additionally, it would be very interesting to study the case of tasks of arbitrary sizes being injected into the system.

7. ACKNOWLEDGMENTS

The author wishes to thank Paul Spirakis for valuable discussions during the write-up of the paper and also for suggesting an appropriate terminology on the categorization of polling strategies.

8. REFERENCES

[1] J. Aspens, Y. Azar, A. Fiat, S. Plotkin, O. Waarts. Online Machine Scheduling with Applications to Load Balancing and Virtual Circuit Routing. In Journal of ACM, Vol. 44 (1997), pp. 486-504.

[2] Y. Azar, A. Broder, A. Karlin, E. Upfal. Ballanced Allocations. In Proc. of 26th ACM-STOC (1994), pp. 593-602. Also in SIAM J. Computing, 29 (1999), pp. 180-200.

[3] P. Berman, M. Charikar, M. Karpinski. Online Load Balancing for Related Machines. In Proc. of 5th Int. Workshop on Algorithms and Data Structures (1997), LNCS 1272, Springer-Verlag 1997, pp. 116-125.

[4] P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking. Balanced Allocations: The Heavily Loaded Case. In Proc. of 32nd ACM-STOC (Portland, 2000), pp. 745-754.

[5] A. Bar-Noy, A. Freund, J. Naor. New Algorithms for Related Machines with Temporary Jobs. In Journal of Scheduling, Vol. 3 (2000), pp. 259-272.

[6] A. Borodin, R. El Yaniv. Online Computation and Competitive Analysis. Cambridge Univ. Press 1998.

[7] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, D. Knuth. On the Lambert W Function. In Advances in Computational Mathematics, Vol. 5 (1996), pp. 329-359.

[8] A. Czumaj and B. Vöcking. Tight Bounds for Worst-Case Equilibria. In Proc. of 13th ACM-SIAM SODA (San Francisco, 2002), pp. 413-420.

[9] Y. Azar. Online Load Balancing. In Online Algorithms: The State of the Art, A. Fiat, G. Woeginger, (Eds.). LNCS 1442, Springer 1998, pp. 178-195.

[10] G. Gonnet. Expected Length of the Longest Probe Sequence in Hash Code Searching. In J. of ACM, Vol. 28, No. 2 (1981), pp. 289-304.

[11] M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, B. Reed (Eds.). Probabilistic Methods for Algorithmic Discrete Mathematics. ISBN: 3-540-64622-1, Springer-Velrag 1998.

[12] E. Koutsoupias, C. Papadimitriou. Worst-case Equilibria. In Proc. of 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS), LNCS 1563, Springer-Verlag 1999, pp. 404-413.

[13] M. Mavronicolas, P. Spirakis. The Price of Selfish Routing. In Proc. of 33rd ACM-STOC (Krete, 2001), pp. 510-519.

[14] M. Mitzenmacher, A. Richa, R. Sitaraman. The Power of Two Random Choices: A Survey of Techniques and Results. In Handbook of Randomized Algorithms (to appear). Also available through `http://www.eecs.harvard.edu/ michaelm/`.

[15] A. Steger, M. Raab. “Balls into Bins” - A Simple and Tight Analysis. In Proc. of RANDOM’98, LNCS 1518, Springer Verlag, 1998, pp. 159-170.

[16] J. Sgall. Online Scheduling. In Online Algorithms: The State of the Art, A. Fiat, G. Woeginger (Eds.). LNCS 1442, Springer 1998, pp. 196-231.

[17] D. Shmoys, J. Wein, D. Williamson. Scheduling Parallel Machines Online. In Proc. of the 32nd IEEE-FOCS (1991), pp. 131-140.

[18] B. Vöcking. How Asymmetry Helps Load Balancing. In Proc. of 40th IEEE-FOCS (New York, 1999), pp. 131-140.