Optimization of power consumption and user impact based on point process modeling of the request sequence

Jean-Baptiste Durand, Stéphane Girard, Victor Ciriza and Laurent Donini

J.-B. Durand (corresponding author) and S. Girard are with Team Mistis, INRIA Rhône-Alpes and LJK, 655 avenue de l'Europe Montbonnot, 38334 Saint-Ismier Cedex, France (Jean-Baptiste.Durand@inria.fr)

V. Ciriza and L. Donini are with Xerox Research Centre Europe, 6 chemin de Maupertuis, 38240 Meylan, France.

Abstract. This article addresses the optimal choice of the waiting period (or $time\text{-}out$) that a device should respect before entering sleep mode, so as to optimize a trade-off between power consumption and user impact. The optimal timeout is inferred by appropriate statistical modeling of the times between user requests. In a first approach, these times are assumed to be independent, and a constant optimal timeout is inferred accordingly. In a second approach, some dependency is introduced through a hidden Markov chain, which also models specific activity states, like business hours or night periods. This model leads to a statistical framework for computing adaptive optimal timeout values. Different strategies are assessed using real datasets, on the basis of power consumption and user impact.

Keywords: hidden Markov chain modeling, optimal timeout, power management, renewal processes, statistical models for request processes

1. Introduction

The goal of this study is to determine a statistical method, based on the analysis of user behavior, achieving a compromise between low power consumption of devices and limited user impact. We describe this primarily with respect to the behavior of printers, however similar policies could also be applied to other devices such as disk drives and displays. Currently, in most printers the time period to wait before entering sleep mode is either set by the administrator or predefined by the device manufacturer according to Energy Star (`http://www.energystar.gov`) environmental standards. Current Energy Star criteria do not take into account observed printer usage patterns. Those criteria instead

set power consumption requirements depending on the device features (*e.g.* functionalities, estimated volume) and marking technology type (*e.g.* laser, solid ink, inkjet). In this paper, observed printer usage patterns are taken into account through the sequence of print job submissions (referred to as the print process). We model a device having several modes with different power consumptions. For a printer, these correspond to:

- *Print mode:* The device activates its marking engine, print path and controller and completes any print requests. Power consumption is typically the highest in this mode.
- *Idle mode:* The device is ready to print immediately and therefore a certain power consumption is required to maintain the device in a state of readiness.
- *Sleep (or standby, or power-save) modes:* The device is not ready to print immediately, which induces a delay between the user request and the actual beginning of the print job. Depending on the printer, one or several such modes are available.

In the sequel, transitions from sleep to idle modes are referred to as *wake-up*, while the reverse transitions are referred to as *shutdown*. Power consumption is typically the lowest in one of the sleep modes, and the difference in power consumption between idle mode and sleep modes is often as large as 40%. From the consumption point of view, the device features are summarized by the power consumption in each of these modes as well as the energy required to switch between these modes.

The goal of this study is to infer the optimal inactivity interval (or *timeout period*) before entering into sleep modes, given both the device power consumption model and observed usage patterns.

1.1. Consumption model and notations

Our approach relies on the following assumptions. Firstly, assuming that each print request is processed as soon as possible, the power consumption during print jobs cannot be reduced in any way, and the request queue must be emptied before power-saving can occur. Therefore, these consumptions and queues are ignored in our analysis, and the times between requests can be assumed positive. Secondly, power consumptions during idle and sleep modes are assumed to be constant. Finally, printing, shutdown and wake-up transitions are assumed to be instantaneous. Therefore, the optimization focuses on the power consumption in idle and sleep modes and on the energy consumption of the associated transitions.

Let m denote the number of sleep modes, a the power consumption (Watts) in idle mode, b_j the power consumption (Watts) in sleep mode j, c_j the energy (Joules) required to switch from sleep mode $j-1$ to sleep mode j and d_j the wake-up energy (Joules) required to switch from sleep mode j to print mode, with $j = 1, \ldots, m$. In this notation, sleep mode 0 corresponds to idle mode, and thus one can define $b_0 = a$. From the consumption point of view, the device features are summarized by the above quantities.

We limit ourselves to timeout strategies consisting of waiting a duration $\tau^{(j)}$ from the latest print onward, before switching into mode j. Since each print request must be processed immediately, the actual switch only occurs if the time between latest print job completion and the following print request is larger than $\tau^{(j)}$. This requires that the sequence $(\tau^{(1)}, \ldots, \tau^{(m)})$ is increasing. It is also assumed that the sequence $(b_0, \ldots, b_m)$ is decreasing, and that the only possible modes accessible from mode j are modes $j+1$ and 0. The transition graph between modes is illustrated in Figure 1.

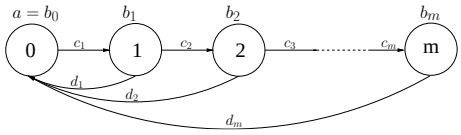

Figure 1. Possible transitions between idle and sleep modes.

1.2. Related work

The issue of power saving strategies has already been addressed in several studies. Although most of them present a very general framework for power management, their applications mainly focus on hardware devices (*e.g.* CPU, monitors, hard disk drives). A wide range of approaches are compared in Lu et al. (2000), using the following typology of methods:

Timeout: A timeout period is fixed either using a quantile of the residual time before next request, or using a parametric function of times between the last two requests and / or request and timeout (see Douglis et al., 1995; Golding et al., 1995; Lu et al., 2000; Cai and Lu, 2005).

L-shape: This is a variant of timeout approaches dedicated to request patterns where short busy periods tend to be followed by a long idle period (Srivastava et al., 1996).

Exponential average: This approach relies on a prediction of next idle period, based on an average of the previous idle periods with exponential weights (Hwang and Wu, 2000).

Stochastic model: These methods aim at finding an optimal probability distribution for the different actions to perform, given the past actions, states of the system and the expected power consumption for each action. The different levels of consumption are related to the notion of state. These approaches mainly rely on the theory of Markov decision processes (or MDPs – see Sutton and Barto, 1998) or their different variants (continuous time, semi-Markov or piecewise homogeneous Markov processes).
Competitive algorithm: A c-competitive power saving algorithm is such that the power consumption is less than c times that of an oracle algorithm (Karlin et al., 1994). An oracle algorithm considers all random variables, including future observations, as known, and achieves the minimal possible power consumption.
Learning tree: Adaptive learning trees transform sequences of idle periods into discrete events and store them in tree nodes. They predict idle periods using finite-state machines and select a path which resembles previous idle periods. At the beginning of an idle period, a learning tree determines an appropriate sleeping state; this algorithm is capable of controlling multiple sleeping states (Chung et al., 1999).

Our method belongs to the category of stochastic models, and combines the principles of continuous time modeling, piecewise identically distributed times between requests and MDPs (the exact connection of our approach with MDPs is discussed in the supplementary material). In Benini et al. (1999), a function of a homogeneous Markov process with discrete time and discrete state space is used to model the sequences of requests. The states represent different rates of requests to the device per time unit. Markov processes are directly used to model completion of those requests and the decision process.

An extension of this work was proposed in Chung et al. (2002) to take into account possible violation of the homogeneity assumption. This is addressed by a piecewise homogeneous Markov request process. This work was further extended in Šimunić (2002) and Bogliolo et al. (2004) using semi-Markov processes for modeling the dynamics of the states and events (typically the requests). These extensions are still discrete-time approaches. Continuous-time models were proposed in Qiu and Pedram (1999) to represent the requests process (by a homogeneous Poisson process), the service process and its queue (by Markov processes with discrete state space). In Ren et al. (2005), the requests process is modeled by a Markov-modulated Poisson process.

In Theocharous et al. (2006), the decision is based on machine learning (logistic regression, k-nearest-neighbors or classification trees). Learning is achieved from examples of actions to perform (turn the device on or off) and vectors (used as predictors) composed of characteristics that reflect the state of activity of the user, and the time since last request. These approaches mainly try to optimize system performance under constraints on power consumption.

In the context of power management for printers, the duration of a CPU cycle is negligible compared to the time between requests. Therefore, a continuous-time model is a natural way to model the request process. The contribution of this paper to power management algorithms is threefold: firstly, it considers the issue of modeling the sequence of requests from a statistical viewpoint, using continuous-time request processes (under the weak assumption of point processes). Secondly, it allows for a characterization of the optimal timeouts in multiple sleep mode devices, as the solutions of separate nonlinear equations. Explicit solutions of these equations are provided in the case of particular renewal process assumptions on the print process. For the sake of conciseness, we focus on the framework of one single sleep mode, and leave to supplementary material extensions to multiple sleep modes. Lastly, user impact can be accounted for in the optimization of the target function associated with this model, using a tractable extension of the basic framework, so that the corresponding optimal timeouts follow straightforwardly.

A general model for the request process is presented in Section 2, together with an assessment of user impact and the associated optimal timeouts. Several request processes are investigated in Section 3. Particular attention is given to parametric models, which allow fast update of the timeout. In Section 4, various power management strategies are compared based on experiments. The comparison criteria rely on out-of-sample prediction of power consumption and on the numbers of shutdowns. Possible extensions to our approach are provided in Section 5.

2. Stochastic model

The print process model is a particular case of a point process, similar to some reliability models, see for instance Rausand and Høyland (2004), Chapter 7. In our framework, the failure sequence is replaced by the print request sequence $\{T_i\}_{i\geq 1}$, with the convention $T_0 = 0$ and where i denotes the index of the print request. Equivalently, the print process can be described by $\{X_i\}_{i\geq 1}$, where $X_i = T_i - T_{i-1}$ is the time between the $(i-1)^{\text{th}}$ and the i^{th} print request. As a consequence of the previous assumptions, the print process is *simple*: there cannot be more than one print request at a time with probability 1. The print process is depicted in the supplementary material, Figure 1.

Recall that we consider a framework with one sleep mode, so that $m = 1$. Consequently, the dependence of all quantities on the sleep mode j will be omitted in the notation used in this section. The timeout period may be updated after each print request i and thus will be denoted by τ_i. Given a probabilistic model for the print process $\{X_i\}_{i\geq 1}$, we aim at optimizing over τ_i the expectation of the energy consumption (denoted by $h(X_i, \tau_i)$) given the past of the

print process $X_{1:i-1} := (X_1, \ldots, X_{i-1})$, between two successive print jobs $i-1$ and i.

Let us denote by f_i the probability density function (pdf) of X_i given $X_{1:i-1}$, let $\bar{F}_i$ be its survival distribution function, and $z_i = f_i/\bar{F}_i$ be its hazard rate function. This is referred to as the failure rate function in reliability theory (Barlow and Proschan (1981), Chapter 2). In our case, it can be interpreted as a *printing rate function*. In the sequel, z_i is assumed to be monotonic. Under this assumption, one can define the asymptotic hazard rate $\ell_i = \lim_{x\to+\infty} z_i(x) \in [0, +\infty]$. Formally, an optimal timeout period is defined by $\hat{\tau}_i \in \arg\min_\tau \mathbb{E}(h(X_i, \tau)|X_{1:i-1})$. To compute the energy consumption $h(X_i, \tau)$, two cases arise:

a) Either the time X_i between two successive printings is larger than τ_i. Then the printer stays in idle mode for τ_i before switching into sleep mode. After a delay $X_i - \tau_i$, the print job is processed and the printer returns to idle mode. Consequently, the energy consumption in this case is $a\tau_i+c+b(X_i-\tau_i)+d$.

b) Or X_i is smaller than or equal to τ_i. Then the printer stays in idle mode for X_i before processing the job. Consequently, the energy consumption in this case is aX_i.

These two cases are illustrated in the supplementary material, Figure 2. Let us define $\Delta t = (c+d)/(a-b)$. In a static analysis of the printer energy consumption, Δt is the time after which switching into sleep mode is less expensive than staying in idle mode (see the supplementary material, Figure 3). If the times of the print requests were known, the optimal strategy would be to enter into sleep mode if $X_i > \Delta t$. For this reason, Δt is frequently called the *break-even* time.

The expected consumption between two successive printings $\mathbb{E}(h(X_i, \tau)|X_{1:i-1})$ is derived in Lemma 1 of the supplementary material. The optimal timeout can then be computed based on the following result (see the supplementary material for a proof):

PROPOSITION 1. *Two situations are examined:*
a) Supposing that the hazard rate function $z_i(x)$ is strictly decreasing in x, three cases occur:

- *If $1/\Delta t < \ell_i$, then $\hat{\tau}_i = +\infty$.*
- *If $\ell_i \leq 1/\Delta t \leq z_i(0)$, then $\hat{\tau}_i$ is the unique root of the equation $z_i(\hat{\tau}_i) = 1/\Delta t$.*
- *If $z_i(0) < 1/\Delta t$, then $\hat{\tau}_i = 0$.*

b) Supposing that z_i is strictly increasing or constant, four cases occur:

- *If* $1/\Delta t < z_i(0)$, *then* $\hat{\tau}_i = +\infty$.
- *If* $z_i(0) \leq 1/\Delta t \leq \min\left(\ell_i, 1/\mathbb{E}(X_i|X_{1:i-1})\right)$, *then* $\hat{\tau}_i = +\infty$.
- *If* $\max\left(z_i(0), 1/\mathbb{E}(X_i|X_{1:i-1})\right) < 1/\Delta t \leq \ell_i$, *then* $\hat{\tau}_i = 0$.
- *If* $\ell_i < 1/\Delta t$, *then* $\hat{\tau}_i = 0$.

It appears that three situations are possible. Either the times between printings are so small on average that the printer should not enter into sleep mode ($\hat{\tau}_i = \infty$), or they are so large on average that the printer should enter into sleep mode immediately ($\hat{\tau}_i = 0$). The intermediate case provides non-degenerate optimal timeouts defined by the equation $z_i(\hat{\tau}_i) = 1/\Delta t$. This result highlights the separate roles of the printer characteristics (summarized by Δt) and the user behavior (modeled through the hazard rate function z_i). The extension of this result to multiple sleep mode printers is given in Proposition 1 of the supplementary material.

In reality, transitions between sleep and idle modes may delay printing. The more frequently the system switches between sleep and idle modes, the more the user will be impacted. We thus propose to model this impact by a penalty term in the energy consumption. We further assume that user impact is proportional to the number of shutdown transitions. With such a model, the consumption between two successive print requests $h(X_i, \tau_i)$ is replaced by the cost $g(X_i, \tau_i) = h(X_i, \tau_i) + \delta \mathbb{1}_{\{X_i > \tau_i\}}$, where $\delta > 0$ is the weight assigned to user impact in the energy consumption. The expected consumption including user impact is given in Lemma 3 of the supplementary material. It turns out that penalizing the consumption by the number of shutdowns can be interpreted as increasing the transition consumption $c + d$ by δ. As a consequence, Proposition 1 still holds with Δt replaced by $\tilde{\Delta} t = (c+d+\delta)/(a-b)$. In particular, when the hazard rate is a decreasing function and there is a non-degenerate optimal timeout period such that $z_i(\hat{\tau}_i) = 1/\tilde{\Delta} t$, the optimal timeout period is an increasing function of δ. Moreover, this property also allows user impact to be accounted for in the break-even time. In practice, $\tilde{\Delta} t$ can be seen as the optimal timeout if X_i follows a particular Pareto distribution – see Section 3.1.

3. Modeling the print process

According to the previous Section, the optimal timeout period depends on the model for the print process through the hazard rate function. Four different print process models are proposed hereafter. In the first three approaches, times between printings are assumed to be independent. In the last approach, a hidden Markov chain (HMC) is used to model dependencies between printing

times. The HMC states can be interpreted as specific states of activity like business hours or night periods. While we only consider a single sleep mode, extension to multiple sleep modes is straightforward.

3.1. Renewal process

In this section, the times between print requests are assumed to be independent. The print process is then a particular case of renewal process (see Rausand and Høyland, 2004, Chapter 7). The random variable modeling the times between printings is denoted by X, since its distribution does not depend on the index i of the print job. Similarly, the optimal timeout period in sleep mode j is denoted by $\hat{\tau}^{(j)}$. In the following, the optimal timeout is studied under the assumptions that X is Weibull, Gamma or Pareto distributed. These three distributions were chosen either for their relevance on the particular datasets we had, or for theoretical reasons in the case of Pareto distributions.

3.1.1. Weibull distribution

The pdf of the two-parameter Weibull distribution is parametrized as $f_X(x) = \alpha\lambda^\alpha x^{\alpha-1}\exp[-(\lambda x)^\alpha]$ for $x > 0$, where $\lambda > 0$ is a scale parameter and $\alpha > 0$ is referred to as the shape parameter. The hazard rate function $z_X(x) = \alpha\lambda^\alpha x^{\alpha-1}$, $x \geq 0$, is strictly decreasing if $\alpha \in (0, 1)$, strictly increasing if $\alpha > 1$, and constant if $\alpha = 1$ (exponential distribution). Proposition 1 yields

$$\hat{\tau} = \left| \begin{array}{ll} (\alpha\lambda^\alpha \Delta t)^{\frac{1}{1-\alpha}} & \text{if } \alpha \in (0,1) \\ 0 & \text{if } \alpha \geq 1 \text{ and } \Delta t < \Gamma(1+1/\alpha)/\lambda \\ +\infty & \text{if } \alpha \geq 1 \text{ and } \Delta t > \Gamma(1+1/\alpha)/\lambda. \end{array} \right.$$

In practical situations, the parameters α and λ are replaced by their maximum likelihood estimates, see Johnson et al. (1995), Chapter 21 for their computation and Section 4 for examples.

3.1.2. Gamma distribution

The pdf of the two-parameter Gamma distribution is parametrized as $f_X(x) = \beta^{-\alpha}\Gamma(\alpha)^{-1}x^{\alpha-1}\exp(-x/\beta)$ for $x > 0$, where $\beta > 0$ is a scale parameter and $\alpha > 0$ is the shape parameter. In this case, no closed-form expression is generally available for the hazard rate function. Nevertheless, it can be shown (Barlow and Proschan, 1981) that, similarly to the Weibull case, the hazard rate function is strictly decreasing if $\alpha \in (0, 1)$, strictly increasing if $\alpha > 1$ and constant if

$\alpha = 1$ (exponential distribution). Thus from Proposition 1,

$$\hat{\tau} = \begin{cases} z_X^{-1}(1/\Delta t) & \text{if } \alpha \in (0,1) \\ 0 & \text{if } \alpha \geq 1 \text{ and } \Delta t < \alpha\beta \\ +\infty & \text{if } \alpha \geq 1 \text{ and } \Delta t > \alpha\beta, \end{cases}$$

where the hazard rate function $z_X(x)$ has to be evaluated numerically through the use of the incomplete gamma function. The maximum likelihood estimates of α and β are computed following Johnson et al. (1995), Chapter 17 and the computation of $\hat{\tau}$ is achieved with a dichotomy procedure.

3.1.3. Pareto distribution

The pdf of the two-parameter Pareto distribution is parametrized as $f_X(x) = \alpha\beta^\alpha x^{-\alpha-1}$ for $x \geq \beta$, where β and α are two positive parameters. The associated hazard rate function is $z_X(x) = \alpha/x$, which yields $\hat{\tau} = \alpha\Delta t$ if $\Delta t \geq \beta/\alpha$ (and $\hat{\tau} = 0$ otherwise). This result is consistent with that in Cai and Lu (2005). As a consequence, the break-even time can be seen as the optimal timeout for a Pareto distribution with $\alpha = 1$. The maximum likelihood estimators of the parameters are given in Johnson et al. (1995), Chapter 20.

3.2. Hidden Markov model

The assumption of a constant hazard rate throughout day and night does not seem realistic *a priori*. It can be expected that during given periods, users will tend to print more often or less often than average. Such periods can be interpreted in terms of activity levels, which yield different levels of hazard rates. Denoting S_i the activity level at ith print request, and assuming a Markovian dependence between the $(S_i)_{i\geq 1}$, leads to an HMC model for the print process.

This corresponds to a heterogeneous distribution of the times between printings, such that there exist some homogeneous periods $(i, \ldots, i+k)$ where $(X_i, \ldots, X_{i+k})$ have the same distribution. These can be interpreted as activity periods, defined by non-visible factors, such as the amount of users at a given time in the printer network (which is related to working hours and can vary with the company), the type of users, country or even site specificities.

Formally (see Ephraim and Merhav, 2002), an HMC is defined by two processes $X_{1:n} = (X_1, \ldots, X_n)$ (observed process) and $S_{1:n} = (S_1, \ldots, S_n)$ (hidden process), such that:

a) $S_{1:n}$ is a homogeneous Markov chain with finite state space $\{1, \ldots, K\}$, with transition matrix A and a distribution $\pi = (\pi_1, ..., \pi_K)$ for the initial state S_1. Here, $S_{1:n}$ is assumed stationary and ergodic. Thus, π also corresponds

to the marginal distribution of S_i. In our case, S_i represents the state of the process at ith printing request, which is not directly observed.

b) Given $S_{1:n} = s_{1:n}$, the X_i are mutually independent, and independent on the $(S_{i'})_{i' \neq i}$, with conditional pdf $f_{\theta_{s_i}}$ (called *emission distributions*), where $(f_\theta)_{\theta \in \Theta}$ is a parametric family of pdf.

In HMC modeling, the non-visible factors underlying the state definition are related to the print requests by the emission distributions, so that the unknown states $S_{1:n}$ can be accessed indirectly, through the print process.

The HMC process aims at modeling both dependence and heterogeneity in the print process. Indeed, although $(X_i)_{i \geq 1}$ is a stationary process, X_i has conditional pdf $f_{\theta_{s_i}}$ given $S_{1:n} = s_{1:n}$. Since this distribution depends on i, the conditional distribution of the print process given a state sequence containing transitions is non-stationary.

As an alternative, non-stationarity can be modeled by a sliding window approach inspired by Chung et al. (2002). The model parameters of the renewal process in Section 3.1 are reestimated after each request i, using dataset $X_{i-\mathcal{L}+1:i}$, where $\mathcal{L}$ is called the window length.

The parameters of the HMC model, which consist in π, A and $(\theta_1, \ldots, \theta_K)$ are estimated by maximum likelihood, using the EM algorithm (see a detailed description in Ephraim and Merhav (2002)).

Adaptive timeout period using HMCs: In the sequel, different strategies are proposed to derive adaptive timeout periods $\hat{\tau}_i$ (updated after each printing job i), taking advantage of the dependence in the print process. Those strategies basically consist of predicting the time to the next print request X_i, from the past observed values $X_{1:i-1}$. Firstly, we propose two approaches based on a prediction $\hat{S}_i$ of the next state value from the past of the process $X_{1:i-1}$ (using two variants for the prediction). The predicted distribution for X_i is then $f_{\theta_{\hat{S}_i}}$. Our third approach considers all possible values of S_i, and thus takes into account the uncertainty about its value.

Viterbi-based approach: In this approach the next state value $\hat{S}_i$ is predicted as

$$\arg\max_k (\max_{s_{1:i-1}} \mathbb{P}(S_{1:i-1} = s_{1:i-1}, S_i = k | X_{1:i-1})).$$

This value is deduced from the Viterbi algorithm (Ephraim and Merhav, 2002).

Filtering-based approach: This approach consists in predicting the next state value $\hat{S}_i$ as

$$\tilde{S}_i = \arg\max_k \beta_i(k),$$

where $\beta_i(k) = \mathbb{P}(S_i = k | X_{1:i-1})$ is the filtered probability. This quantity is deduced from $\mathbb{P}(S_{i-1} = j | X_{1:i-1})$ (forward recursion in Ephraim and Merhav,

2002).

Approach based on full conditional distribution: This approach consists in computing the hazard rate function of X_i given $X_{1:i-1}$. The pdf f_i (respectively the survival distribution function $\bar{F}_i$) of this distribution is a mixture of the pdf $(f_{\theta_k})_{1\leq k\leq K}$ (respectively $(\bar{F}_{\theta_k})_{1\leq k\leq K}$) with weights $(\beta_i(k))_{1\leq k\leq K}$. The hazard rate function follows immediately.

Each of the three approaches results into an estimated pdf for the predictive distribution of X_i, namely $f_{\theta_{\hat{s}_i}}$ in the first two cases and f_i in the third one. Each pdf is associated with a hazard rate function z_i. The optimal timeout period $\hat{\tau}_i$ is given by Proposition 1. In the approach based on full conditional distributions, even in the case of Weibull, Gamma or Pareto observation distribution families $(f_\theta)_{\theta\in\Theta}$, we could not derive general conditions on the parameters $(\theta_k)_{k=1,\ldots,K}$, under which equation $z_i(\hat{\tau}_i) = 1/\Delta t$ has a unique solution. Thus, numerical methods have to be used, to determine whether the optimal timeout period is null, positive or infinite.

4. Experiments

Our methodology is illustrated, in the sequel, by experiments on two real datasets. The efficiencies of the timeout strategies introduced in Section 3 are compared in terms of energy consumption. These strategies are also compared with four alternatives called *Energy star method*, *oracle method*, *c-competitive algorithm* and *exhaustive search method*. In the *Energy star method*, the timeout period is fixed so as to comply with the Energy Star standard, depending on the printer features. In the *oracle method*, the future of the print process is assumed to be known. The printer switches into sleep mode j before print job i if $X_i > \Delta t_j$. Let us highlight that this reference method provides a lower bound on the consumption but cannot be used in practice. The strategy consisting of setting the timeout at Δt_j is referred to in Lu et al. (2000) and Cai and Lu (2005) as a *c-competitive* algorithm (in the sense of Karlin et al., 1994). Since this is a deterministic algorithm, here $c = 2$. Finally, the *exhaustive search method* consists of finding the timeout that minimizes the actual consumption on the dataset at hand, without parametric assumption. This approach assumes a constant timeout. To determine this timeout, a grid of possible values is considered. The actual consumption is computed for each possible timeout, the optimal one being retained. Moreover, the method mentioned in Section 3.1 will be called *static method*, while the HMC-based method described in Section 3.2 and resorting to the Viterbi algorithm for state restoration will be referred to as the *Viterbi method*. The results provided by the two variants for HMCs, referred to as *filtering method* and *conditional method*, were very close to those of the

Viterbi method; thus their detailed results are given in supplementary material only (Table 2). The *sliding window method*, described in Section 3.2, is also considered in the experiments.

In both datasets, times between printings were deduced from the print logs, recorded during the whole of the year 2006 on XRCE print infrastructure, which is composed of 14 printers and involves 155 users. The first printer is a *Xerox WorkCentre 238* model with two sleep modes ($a_1 = 270W$, $b_1 = 150W$, $b_2 = 50W$, $c_1 = c_2 = 0$, $d_1 = 40kJ$ and $d_2 = 200kJ$) and the second printer is a *Phaser 4500* model with one single sleep mode ($a_1 = 80W$, $b_1 = 16W$, $c_1 = 0$ and $d_1 = 25.3kJ$).

Static, Viterbi and sliding window methods require the selection of a family of distributions for the times between print requests. Given the histograms and statistics (skewness and kurtosis), available in supplementary material (Figures 4, 5 and Table 1), parametric families with fat tails that contain either decreasing pdf, or unimodal pdf with positive skewness were considered: Gamma, Weibull, Lognormal and Pareto. We also included the two classical Gaussian and Cauchy families. The final choice was based on the Bayesian Integrated Criterion BIC (Schwarz, 1978), whose values are summerized in Table 1. The selected model has a maximal BIC value.

Table 1. Values of BIC for selection of a distribution for the times between printings. The values of the selected models are indicated in bold.

Distribution	BIC	
	WorkCentre 238	Phaser 4500
Weibull	**13,324**	**7,065**
Gamma	13,253	**7,069**
Lognormal	13,253	6,976
Pareto	10,989	5,806
Cauchy	9,821	4,569
Normal	8,226	3,984

It appears that Weibull is the most appropriate distribution for the first dataset, and that Gamma and Weibull are both appropriate for the second dataset. In the following, a Weibull distribution is adopted in both cases to model the distribution of the times between printings, when those are assumed to be independent. Its advantage over Gamma distributions is the derivation of an explicit timeout (see paragraph 3.1.2). The considered HMC model also has Weibull emission distributions, and three states, which can be interpreted as rush, normal and calm periods, from the point of view of the print requests. Note that the number of states or the family of emission distributions could also be selected using penalized likelihood criteria (Gassiat, 2002) or cross-

validation (Celeux and Durand, 2008). The M step for parameter estimation by the EM algorithm is given in the supplementary material.

4.1. Cross-validated assessment of the strategies

The goal of this experiment is to investigate the methods' performance on future data, and thus to assess their generalization capacities. We focus on the *Xerox WorkCentre 238* dataset ($n = 3910$ print jobs) and user impact is not considered. Its predefined timeouts according to Energy Star environmental standards are 900 s for the first sleep mode and $1,800$ s for the second. The test procedure is multi-fold cross-validation (Zhang, 1993), as follows: the dataset is divided into L contiguous sub-samples of equal size. Then for each sub-sample $\ell \leq L-1$, the method parameters are estimated on this sub-sample while the consumption is computed on sub-sample $\ell+1$. The length of the sliding window $\mathcal{L}_\ell$ is one of the parameters; this is also estimated on sub-sample ℓ only, by minimizing the consumption over $\mathcal{L}_\ell$. Three cases are considered: $L = 10$ sub-samples of size 361, $L = 30$ sub-samples of size 121 and $L = 60$ sub-samples of size 61. Results are summarized in Tables 2 and 3. The computation times include the computation of the actual consumption for the dataset.

It appears on Table 2 that exhaustive search, static and sliding window methods are the most efficient in terms of consumption. The consumption associated with these methods is about 12% larger than the lower bound given by the Oracle method. This slight increase of the optimal consumption confirms that the Weibull distribution achieves a proper fit to the sample. Besides, exhaustive search, static and sliding window methods are quite robust since they yield a constant consumption, whatever the subdivision. Moreover, the standard deviation of the consumption represents less than 2% of the total consumption. Among these four methods, the static one is at least one thousand time faster than the other ones. Experiments were conducted in Matlab on an Intel Pentium Dual Core running at 2.5 GHz.

The 3-state HMC model provides a better fit to the data than an independent 3-state mixture model, which itself fits the data better than an independent Weibull model (with the following values of BIC, respectively: 13,696; 13,503 and 13,324). This highlights that the times between requests are dependent, and that changes in the hazard rate occur. However, the above results show that the better fit of the HMC model does not directly translate into a better consumption.

Focusing on Table 3, it appears that the timeout periods provided by the static and exhaustive search methods are approximately independent of the subdivision of the sample, for both methods. Let us emphasize that static timeouts benefit from small standard deviation whereas exhaustive search timeouts suffer

Table 2. Energy consumption and mean computation time associated to the different strategies.

	Total consumption (kWh)			Standard deviation of consumption			Mean computation time by sample (ms)		
Sample size	361	121	61	361	121	61	361	121	61
Energy Star	500	500	500	7.99	4.73	3.04	1.2e+00	1.0e+00	2.0e+00
$\tau^{(1)} = \tau^{(2)} = 0$	498	498	498	6.77	3.76	2.55	2.0e+00	1.0e+00	1.0e+00
Exhaustive search	446	446	447	6.86	4.17	2.76	6.6e+04	1.5e+05	2.7e+05
Oracle	399	399	399	7.13	4.11	2.72	2.0e+00	2.0e+00	2.0e+00
c-competitive	471	471	471	7.66	4.54	2.94	5.0e-01	1.0e+00	2.0e+00
Static	446	446	446	7.02	4.13	2.77	2.0e+01	5.0e+01	9.0e+01
Sliding window	445	445	444	7.02	4.18	2.77	5.2e+05	1.5e+05	6.6e+04
Viterbi	471	464	462	8.07	3.97	2.64	1.0e+04	4.3e+03	2.8e+03

from a high variability. As a conclusion, static method seems to be an accurate, reliable and fast method to select the optimal timeouts. A decrease of about 12% of power consumption can be achieved with regard to the Energy Star method. The gain with regard to the competitive algorithm is about 6%.

Table 3. Timeout associated to the different strategies.

		Mean timeouts (s)			Standard deviation of timeouts		
Sample size		361	121	61	361	121	61
Exhaustive search	$\tau^{(1)}$	26	25	24	13	17	21
	$\tau^{(2)}$	203	208	218	55	108	121
Static	$\tau^{(1)}$	11	12	12	2	5	7
	$\tau^{(2)}$	179	188	192	33	64	84

4.2. Assessment of user impact

In what follows, the behavior of the methods is compared when taking user impact into account on the Phaser 4500 printer. Our test procedure is the following: The dataset ($n = 2,320$) is divided into 2 sub-samples with the same size. Parameters of each method are estimated on the first sub-sample, while the total consumption is computed on the second one as the penalty δ varies.

The variations of the number of shutdowns (or equivalently wake-ups) as a function of the penalty δ are depicted in Figure 2. Given a delay of 8 s caused by each transition on this printer model, the y-axis in Figure 2 also corresponds to an upper bound of the total delay, from the users' point of view. Even though, for a fixed penalty δ, the different methods yield different numbers of

shutdowns and different consumptions, it appears on Figure 3 that, for a fixed number of shutdowns, there exists some value of the penalty (that depends on each method) such that the consumption is identical for every method. This shows that the different methods are globally equivalent from the point of view of user impact, up to a rescaling of the penalty. Keeping in mind the conclusions of the previous paragraph, it seems that the static methods should be preferred since they are the simplest and most robust ones.

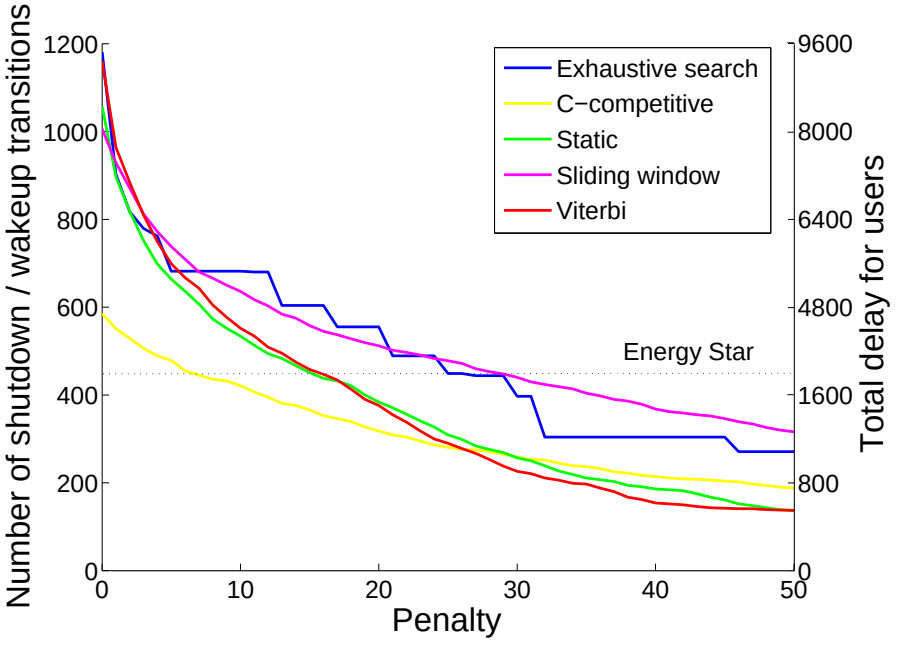

Figure 2. Number of shutdown transitions (left vertical axis) and total delay for users (right vertical axis) as the penalty δ increases.

4.3. Real-world implementation

In power saving issues, it is important to consider the consumption induced by the power management infrastructure itself (hardware and software). Indeed, there is a risk that the power savings may be eroded by the extra consumption caused by our algorithm. In the case of printing infrastructures, large device fleets are usually managed by dedicated servers in charge of several tasks. One can propose to extend their capabilities of device management software by per-

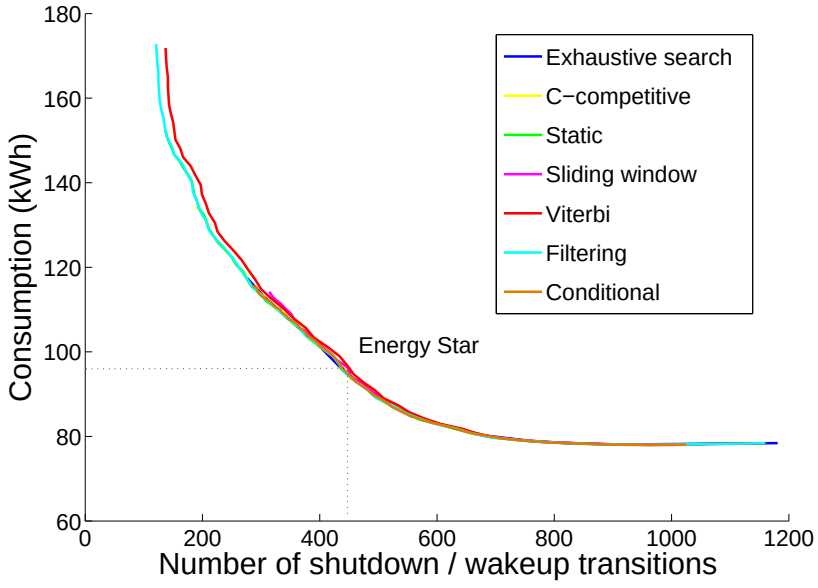

Figure 3. Consumption as a function of the number of shutdown transitions obtained with the different methods.

forming the timeout optimization. In the case of Xerox, the static method was implemented in the production prototype in .NET, using a high performance mathematical library. The execution time of the .NET implemented prototype with respect to Matlab is on average 10 times slower. One run of the static method on a 400 W server represents 400 ms of CPU time, *i.e.* $4.4\text{e}-5$ kWh, which is negligible with respect to the energy consumption of a printer.

A real-world experiment was conducted on 100 Xerox Phaser 4500 printers where 47,000 print jobs were collected. On this basis, a predictive model was built, based on the renewal process approach. Using this model, the probability of shutdown of the printer was computed, as well the associated consumption and timeout, for any value of the penalty. Figure 4 shows the expected decrease of the consumption as a function of the expected increase of the number of transitions. The reference model is a renewal process without penalty (using the estimated optimal timeout for $\delta = 0$). To illustrate the variability of these gains, their values is represented on a histogram (Figure 5), corresponding to the gain on each of the 100 printers. Consequently, the users can now specify

which increase (in %) in the number of shutdowns they are ready to accept (or equivalently which increase of an upper bound of the time they are ready to wait). Then the model deduces the corresponding penalty, timeout and consumption.

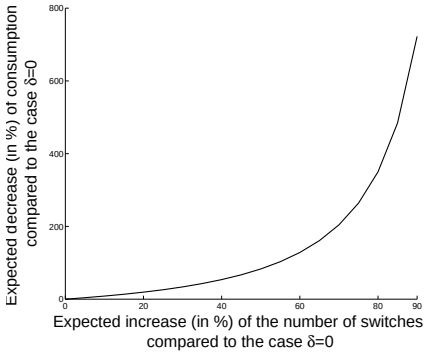

Figure 4. Expected decrease (in %) of the consumption as a function of the expected increase (in %) of the number of transitions. The reference model is a renewal process without penalty.

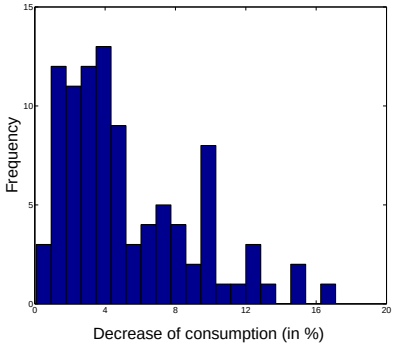

Figure 5. Histogram of consumption gains (on each of the 100 printers) obtained with a number of switches twice larger than the Energy Star standard.

5. Conclusion and discussion

In this paper, we have proposed a statistical cost-based analysis to determine optimal timeout period for devices. The theoretical formulation of power consumption in terms of a print process can be considered as a stepping stone for more complex models (*e.g.* incorporating covariates) that will allow the model to progressively gain completeness in the consideration of other several cost factors, as for example device aging due to increased transitions from power saving mode due to a more dynamic power saving policy. We have also established the foundations to develop in the future a power saving strategy capable of performing accurate prediction of power saving entry as described in this article, but also of optimal power saving exit.

A further extension of this work is the challenging issue of optimal redirection of print jobs and power saving policy within a network of printers managed by a server. Given a printing request, this consists in determining on which printer the job has to be processed, and after what delay each printer has to be turned into sleep mode, so as to minimize the global consumption. Modeling this problem should take into account constraints due to user impact, that are partially related to network connectivity.

Finally, our approach deals separately with model identification (parameter estimation from trajectories of user requests) and computation of the optimal timeout periods (in a framework with fixed parameters). As an alternative, a unified model for handling both model identification and decision taking would be provided by the Bayesian Partially-Observed Markov Decision Processes (POMDPs) in Poupart and Vlassis (2008). Here the non-observed part of the MDP would consist in, firstly, the unknown parameters, considered as stochastic in a Bayesian framework, and secondly, potential unknown states as in the HMC models. The benefit of Bayesian POMDPs to our application would come from taking into account simultaneously the different sources of uncertainty: states of the printer and of the user, value of the parameter and of the reward.

References

Barlow, R. and Proschan, F. (1981) *Statistical theory of reliability and life testing; Probability models.* To Begin With, Silver Spring.

Benini, L., Bogliolo, A., Paleologo, G. and Micheli, G. D. (1999) Policy Optimization for Dynamic Power Management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **18**(6), 813–833.

Bogliolo, A., Benini, L., Lattanzi, E. and Micheli, G. D. (2004) Specification

and Analysis of Power–Managed systems. *Proceedings of the IEEE*, **92**(8), 1308–1346.

Cai, L. and Lu, Y.-H. (2005) Joint power management of memory and disk. In *DATE '05: Proceedings of the conference on Design, automation and test in Europe*, pp. 86–91. Washington, DC, USA: IEEE Computer Society.

Celeux, G. and Durand, J.-B. (2008) Selecting Hidden Markov Model State Number with Cross-Validated Likelihood. *Computational Statistics*, **23**, 541–564.

Chung, E.-Y., Benini, L., Bogliolo, A., Lu, Y.-H. and Micheli, G. D. (2002) Dynamic Power Management for Nonstationary Service Requests. *IEEE Transactions on Computers*, **11**(51), 1345–1361.

Chung, E.-Y., Benini, L. and Micheli, G. D. (1999) Dynamic Power Management Using Adaptive Learning Tree. In *International Conference on Computer-Aided Design (ICCAD '99)*, pp. 274–279.

Douglis, F., Krishnan, P. and Bershad, B. (1995) Adaptive disk spin-down policies for mobile computers. In *Proc. 2nd USENIX Symp. on Mobile and Location-Independent Computing*.

Ephraim, Y. and Merhav, N. (2002) Hidden Markov processes. *IEEE Transactions on Information Theory*, **48**, 1518–1569.

Gassiat, E. (2002) Likelihood ratio inequalities with application to various mixtures. *Annales de l'Institut Henri Poincaré*, **38**, 897–906.

Golding, R., Bosch, P., Staelin, C., Sullivan, T. and Wilkes, J. (1995) Idleness is not sloth. In *TCON'95: Proceedings of the USENIX 1995 Technical Conference Proceedings*, pp. 17–17. Berkeley, CA, USA: USENIX Association.

Hwang, C.-H. and Wu, A. C. (2000) A predictive system shutdown method for energy saving of event-driven computation. *ACM Trans. Des. Autom. Electron. Syst.*, **5**, pages 226–241.

Johnson, N., Kotz, S. and Balakrishnan, N. (1995) *Continuous Univariate Distributions, 2nd edition, vol.1.* Wiley Series in Probability and Statistics.

Karlin, A. R., Manasse, M. S., McGeoch, L. A. and Owicki, S. (1994) Competitive randomized algorithms for nonuniform problems. *Algorithmica*, **11**, 542–571.

Lu, Y.-H., Chung, E.-Y., Šimunić, T., Benini, L. and Micheli, G. D. (2000) Quantitative Comparison of Power Management Algorithms. In *DATE '00: Proceedings of the conference on Design, automation and test in Europe*, pp. 20–26. IEEE Computer Society.

Poupart, P. and Vlassis, N. (2008) Model-based Bayesian Reinforcement Learning in Partially Observable Domains. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM), Fort Lauderdale, Florida, USA*.

Qiu, Q. and Pedram, M. (1999) Dynamic Power Management Based on Continuous-Time Markov Decision Processes. In *DAC '99: Proceedings of the 36th ACM/IEEE conference on Design Automation, New Orleans, Louisiana (USA)*, pp. 555–561. New York, NY, USA: ACM.

Rausand, M. and Høyland, A. (2004) *System Reliability Theory: Models, Statistical Methods, and Applications, 2nd Edition*. Wiley–Interscience.

Ren, Z., Krogh, B. and Marculescu, R. (2005) Hierarchical adaptive dynamic power management. *IEEE Transactions on Computers*, **54**(4), 409–420.

Schwarz, G. (1978) Estimating the dimension of a model. *The Annals of Statistics*, **6**, 461–464.

Šimunić, T. (2002) *Dynamic management of power consumption*, pp. 102–125. Graybill, R., Melhem, R., eds.: Power Aware Computing. Kluwer Academic.

Srivastava, M. B., Chandrakasan, A. P. and Brodersen, R. W. (1996) Predictive system shutdown and other architectural techniques for energy efficient programmable computation. *IEEE Trans. Very Large Scale Integr. Syst.*, **4**(1), 42–55.

Sutton, R. S. and Barto, A. G. (1998) *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts.

Theocharous, G., Mannor, S., Shah, N., Gandhi, P., Kveton, B., Siddiqi, S. and Yu, C.-H. (2006) Machine Learning for Adaptive Power Management. *Intel Technology Journal*, **10**(4), 298–311.

Zhang, P. (1993) Model selection via multifold cross validation. *The Annals of Statistics*, **21**(1), 299–313.