




# Decrease and reset for power-down

James Andro-Vasko<sup>1</sup> · Wolfgang Bein<sup>1</sup>  · Hiro Ito<sup>2</sup> · Shoji Kasahara<sup>3</sup> · Jun Kawahara<sup>3</sup>

Received: 30 December 2020 / Accepted: 16 August 2021 / Published online: 3 September 2021  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

## Abstract

We consider the classical power management problem: There is a system or “device” which has two states—ON and OFF—and one has to develop a control algorithm for changing between these states as to minimize cost (energy or some other hybrid cost) when given a sequence of service requests. We analyze this problem in terms of online competitive analysis to obtain worst-case guarantees. Although an optimal 2-competitive algorithm exists, that algorithm does not result in good performance in many practical situations, especially in case the device is not used frequently. To take the frequency of device usage into account, we construct an algorithm based on the concept of “slackness degree”. Then by relaxing the worst-case competitive ratio of our online algorithm to  $2 + \varepsilon$ , where  $\varepsilon$  is an arbitrary small constant, we make the algorithm flexible to slackness. The algorithm thus automatically tunes itself to slackness degree and gives better performance than the optimal 2-competitive algorithm for real world inputs. In addition to worst-case competitive ratio analysis, a queueing model analysis is given and computer simulations are reported, confirming that the performance of the algorithm is high. We show how the approach can be generalized to a situation where the system has a number of intermediate states. Our model can be used to facilitate renewable energy integration into the electrical grid and we highlight that an online competitive approach can yield techniques for grid resiliency.

**Keywords** Power-down problems · Online competitive analysis · Decrease and reset scheduling · Queueing analysis · Monte Carlo simulation · Renewable energy management

---

A preliminary version appeared in the Proceedings of the 9th Workshop on Approximation and Online Algorithms.

---

Wolfgang Bein: Research supported by NSF grant CCR-9821009. Shoji Kasahara: Part of JST ERATO MINATO Project.

---

✉ Wolfgang Bein  
wolfgang.bein@unlv.edu

Extended author information available on the last page of the article

# 1 Introduction

## 1.1 The power-down problem

We consider a device, which has two power states, ON and OFF, and a number of intermediate states  $\mathcal{I} = \{\text{INT}_j | j = 1, \dots, \ell\}$ . For example, such a device could be a cell phone, a workstation, an HVAC unit, or a power generation system. If  $\mathcal{I} = \emptyset$ , the device is called a two-state device, else it is called an  $\ell + 2$  state device, i.e. ON/OFF plus the number of intermediate states in  $\mathcal{I}$ . In the OFF-state the device consumes zero amount of energy (or amount of resource in the case of a power generation system) and the running cost of the device in the ON state is proportional to the time of usage. The intermediate states serve as sleep states, where the running cost is also proportional to time, but at only smaller cost  $r_j \in (0, 1)$  per time unit. Thus the cost to remain in  $\text{INT}_j$  for time  $\Delta t$  is  $\Delta t \cdot r_j$ . There is no cost to switch from ON to OFF or any of the intermediate states. But a fixed power cost of  $a > 0$  occurs to switch to ON from OFF and a smaller cost  $a_j \in (0, 1)$  occurs to switch to ON from any of the intermediate states  $\text{INT}_j$ . Table 1 summarizes this system. In this paper we will mainly focus on two- and three-state systems.

At any time the device may be in any state but it must be turned to the ON state when service is requested. More precisely, for some  $n \in \mathbb{N}$ , let  $t_1^b, \dots, t_n^b$  and  $t_1^e, \dots, t_n^e$  be non-negative real values with  $t_1^b < t_1^e < t_2^b < t_2^e \dots < t_n^b < t_n^e$  that represent requests for service  $S_i$  between begin of service times  $t_i^b$  and end of the service times  $t_i^e$  ( $i = 1, \dots, n$ ) which means that at time  $t_i^b$  the state of the device must be in ON until time  $t_i^e$ . Between requests the device can remain in the ON state, go to the OFF state or any of the intermediate states. If the device goes to an intermediate state for a certain duration, we say it is *standing by* and refer to this duration as the *standby time*. The duration from the end of service  $S_i$  to the beginning of service  $S_{i+1}$  (i.e.  $t_{i+1}^b - t_i^e$ ) is called the *waiting time*. Clearly, if the waiting time is small, rather than switching off, it may be better stand by in anticipation  $S_{i+1}$ .

It is not known in advance when and how many service requests occur, thus an algorithm has to decide whether to switch states without knowing what the next request will be. Such an algorithm is called *online*; an online algorithm makes its decisions without knowledge of future input. This is in contrast to an offline algorithm which operates under full knowledge of the entire request sequence in advance. We say that an online algorithm  $\mathcal{A}$  has competitive ratio  $C$  for a given request sequence  $\sigma$ , if

**Table 1** Costs when running and for switching

| State          | Costs            |                  |
|----------------|------------------|------------------|
|                | Running/Standby  | Switching        |
| OFF            | 0                | $a$              |
| ...            | ...              | ...              |
| $\text{INT}_j$ | $s_j \in (0, 1)$ | $a_j \in (0, 1)$ |
| ...            | ...              | ...              |
| ON             | 1                | 0                |

$$\text{cost}_{\mathcal{A}}(\sigma) \leq C \cdot \text{cost}_{\text{opt}}(\sigma)$$

where  $\text{cost}_{\mathcal{A}}(\sigma)$  is the cost of  $\mathcal{A}$  to serve  $\sigma$  and  $\text{cost}_{\text{opt}}(\sigma)$  is the cost of the optimal offline algorithm on that same request sequence. (As mentioned above, for the power-down problem  $\sigma = t_1^b < t_1^e < t_2^b < t_2^e \dots < t_n^b < t_n^e$  for some  $n$ .) If the inequality holds for all possible input sequences, then we say that the online algorithm is  $C$ -competitive; furthermore the competitive ratio for a given algorithm  $\mathcal{A}$  is defined as the smallest such  $C$  possible. (For a comprehensive overview over online competitive analysis see [14].) The goal is thus to find an online algorithm with competitive ratio best possible, *i.e.* an online algorithm with the smallest competitive ratio. We refer to such an algorithm as an ‘optimal worst-case competitive ratio algorithm’ denoted here by  $\text{OWCR}$  with the stipulation that we break ties arbitrarily if more than one optimal online algorithm exists.

Under competitive analysis, for the two-state problem, an optimal online algorithm  $\text{OWCR}$  is well known for this problem [18, 24, 25]:  $\text{OWCR}$  switches to ON at a service request. After the service,  $\text{OWCR}$  remains on to stand by for time  $a$ . This means that if another service is requested within the standby period,  $\text{OWCR}$  simply remains in state ON. Only if the waiting time is larger than  $a$ , will  $\text{OWCR}$  go to OFF after a standby time of  $a$ . Algorithm  $\text{OWCR}$  is 2-competitive, which is optimal.

This situation is analogous to the noted ski rental problem [25]. In the ski rental problem a “friend” invites an invitee to go skiing, one can rent skis at cost  $\$r$  or purchase them at cost  $\$a = k \cdot \$r$ , for some  $k \in \mathbb{N}$ . Unlike the inviter, the invitee does not know how often they will be invited. Thus the invitee has to decide how often to rent and when to buy. It is easy to see that the best competitive ratio of the invitee cost versus the inviter cost achievable by any online algorithm for the ski rental problem is 2; the strategy is to rent  $k$  times before buying. The ski rental problem is analogous to the two-state power-down problem in the following way. Both problems are about when to commit to a higher cost in anticipation of future savings. In the ski-rental problem, this commitment is when buying skis, in the power-down problem the commitment is to accept a (large one-time) switching-on cost next when the system is switched off.

Though  $\text{OWCR}$  has the best possible worst-case competitive ratio this does not mean that it will perform well in most situations. For example, if the service requests  $\mathcal{S}_i$  are infinitesimally short and all  $\mathcal{S}_i$  are spaced far apart, then  $\text{OWCR}$  remains in the ON state for an extra  $a$  units for each service. This is quite wasteful as the cost for  $\text{OWCR}$  is  $2a$  (the switching cost and running cost are both  $a$ ), whereas the optimal offline algorithm “OPT” turns off immediately after each service is finished. OPT pays only the switching cost  $a$  (and the infinitesimal running cost). Although  $\text{OWCR}$  does not seem clever at all, under worst-case competitive analysis this algorithm has the best performance, namely a ratio of 2.

*Basic concept of our algorithm* Since  $\text{OWCR}$  is optimal it appears there is no improvement possible, unless we leave the online competitive model altogether. This is undesirable as described above. Our idea, instead, is to consider algorithms with competitive ratio only worse from best by a small positive constant  $\varepsilon$ : such an approach allows for the design of an entire spectrum of algorithms with

“almost-best” competitiveness. We call such algorithms  $\varepsilon$ -OPT. In this paper then we present such  $\varepsilon$ -OPT algorithms—algorithms which maintain close to an optimally competitive ratio while additionally performing better for a multitude of instances. The type of algorithms we design react adaptively to fluctuations: they gradually decrease the duration of the standby time when waiting times are large (in a two state system this means longer than  $a$ ) until the waiting time becomes short again (a value within  $a$  for two-states) at which point standby times are reset. Informally this can be described as “decrease and reset” – standby times are successively tapered down in an “off peak” situation when the frequency of requests is low and then reset if the frequency goes up.

We introduce a parameter called “slackness degree,” which represents the frequency of serviced requests and we construct  $\varepsilon$ -OPT algorithms, which have superior competitiveness (*i.e.* lower than  $2 + \varepsilon$ ) in the presence of slackness. The competitive ratio improves when the frequency of requests within any time period is small—something very desirable in practice, while maintaining a guarantee of competitiveness within  $\varepsilon$  of best possible competitiveness in the worst-case. We emphasize that the algorithm is online and, as such, need not know the actual slackness degree.

## 1.2 Related algorithmic work

As mentioned above, the tight optimal competitive ratio of the two-state power-down problem is 2; this has been observed in various contexts, see [22, 24, 25, 32]. In that sense the problem is settled and therefore there are no further results. When a probability distribution for the service requests is given, the expected cost is minimized; these algorithms are called probability-based. Such an algorithm utilizes knowledge of the probability distribution and the competitive ratio improves to  $e/(e - 1) \approx 1.582$ ; this bound is tight for probability-based algorithms [22–24]. It is important to note that such an algorithm is not defined for arbitrary data and an algorithm designed for a specific distribution will perform poorly for arbitrary request sequences. In contrast, our algorithm performs with close to optimal competitiveness while performing well for a range of sequences which we parameterize using a slackness parameter.

There is a body of work regarding multiple intermediate states as declared in Table 1. Irani et al. [21] show that the ski-rental approach for two state systems can be generalized to multiple states both in the deterministic case using a “lower envelope” techniques resulting in a 2-competitive algorithms as well as in the probability-based situation with tight competitiveness  $e/(e - 1)$ .

There are customized algorithms for a specific number of intermediate states, see our papers [7] for systems with one extra state, *i.e.* three-state systems, [8] for systems with finitely many but few states, [6] for systems with a continuous number of states, and our survey paper [9]. More generally, Augustine et al. [10, 11] give an algorithm that, given a system, produces a deterministic strategy whose competitive ratio is arbitrarily close to optimal.

We note that online competitive algorithms for power-down are related to earlier work on “speed-scaling” of CPUs [4, 22]. Furthermore, there is a body of work from the embedded systems community where a number of adaptive strategies were proposed in the literature. In particular, Chung et al. [15] address multiple-state systems in this way.

### 1.3 Electric power systems and renewable energy management

Power-down mechanisms are common in electronic control, such as power optimization for hand-held devices, laptop computers, work stations and data centers [2, 3, 30, 33]. But the model is also useful to handle power-down phenomena in a future electrical grid, which predominantly relies on renewable energy, see our paper [12] for a survey. In the traditional energy grid, when renewables produce a surplus of energy, such surplus generally does not affect the operation of traditional power plants. Instead, renewables are throttled down or the surplus is simply ignored. But in the future when the majority of power is generated by renewables this is not tenable. Rather, traditional power plant output needs be throttled down or switched off in response to less predictable renewable supplies. Online competitive models have the advantage that little statistical insight is needed. We note that online competitive algorithms were developed to model an emerging internet where requests are not driven by a well-defined distribution but requests which are largely unpredictable. Similarly energy management in a distributed smart grid will benefit from models which are rooted in such online competitive models; for example, network design has been studied extensively in the online competitive setting; see Meyerson [29] for work on dynamic network design.

It could be argued that a game-theoretic approach which assumes an omniscient adversary may not be so realistic for modeling the grid, however this kind of modeling gives performance guarantees in the absence of reliable forecasting. For example, climate scientists have noted unusual weather patterns related to a change in the Arctic Oscillation (OA) and North Atlantic Oscillation (NAO) [19]. Recently, unprecedented winter storms across Texas in February 2021 caused wide-spread power outages [40]. See also Maimó-Far et al. [26] for unpredictability issues around renewables. In order to guarantee a resilient grid, worst-case assumptions must be taken into account.

There is a large body of work on the unit commitment problem in electrical power production, see Padhy [31] for a bibliography, and Huang, Pardalos and Zhang [20] for a comprehensive review of unit commitment problems in electrical power generation. The introduction of renewables has significantly increased the underlying uncertainty in the system, an aspect which has been recognized increasingly in the literature, see Tahanan et al. [1, 36] for a comprehensive survey on recent work and algorithmic techniques. A recent survey on integrating renewables and the unique challenges involved is in [34]. Ben-Tal, Bertsimas, and Brown give models which provide some measure of robustness against distributional variation [13]. However, renewable energy management can benefit from online competitive modeling where algorithms allocate resources in hard

to predict circumstances and where reliable distributional assumptions cannot be made.

Budischak et al. [16], for example, model many combinations of renewable electricity sources (inland wind, offshore wind, and photovoltaics) with electrochemical storage (batteries and fuel cells), incorporated into a large grid system. One of the components available to manage variable generation is (limited) fossil backup where conventional power is cycled as needed to fill gaps. More precisely, cycling refers to the operation of electric generating units at varying load levels, including on/off, load following, and minimum load operation, in response to changes in system load [28]. There are technical limits and costs associated with cycling conventional power plants. Van den Bergh and Delarue [38] point out the importance of including full cycling costs in unit commitment scheduling. They also discuss the different ways of quantifying various types of cycling costs, such as direct start costs, indirect start costs, forced outage costs, ramping costs and efficiency costs. Very recently, de Mars et al. [17] have estimated the impact of variable renewable energy on base-load cycling in the UK and it is expected that such impact will only be larger in the future. As a result there are numerous studies on the costs of cycling itself [27, 35, 38, 41], including Integrated Solar Combined Cycle Power Plants [5]. We also mention a recent study conducted at Sandia National Laboratories, which described implications of power plant idling and cycling on water use intensity [37]

Cycling costs can be avoided by the obvious method of not cycling a unit and that may include staying on at a loss and we are modeling this tradeoff by abstracting the problem as a power-down problem in the online competitive setting. While we are mainly concerned with decrease and reset techniques in the two-state situation (Sects. 3 and 4) our method can be adapted to situations where there is partial load operation [38], *i.e.* where there are “intermediate states” other than on/off (Sect. 5).

## 1.4 Organization of the paper

Though we have described the general problem earlier in the introduction we give a concise statement of the two-state problem (and further notations) in Sect. 2. Section 3 presents the decrease and reset algorithm and gives the concept of slackness degree. We study the competitive ratio under this concept. In Sect. 4 we analyze the cost performance of the decrease and reset algorithm using queueing theory. After deriving the model, we run the model with specific parameters. Furthermore, we report results of experiments conducted using Monte Carlo simulation in order to validate the results obtained through the analysis. We also compare our approach to a number of simplified reset algorithms. The decrease and reset concept can be generalized to systems with more than two states, *i.e.* systems with intermediate states. We discuss the use of the decrease and reset approach in a three state system in Sect. 5. We conclude and give further outlook in Sect. 6.

## 2 Problem statement

We now define our system in detail. In this paper we consider a device or system with infinite capacity that has two states, an *ON-state* and an *OFF-state* (simply ON and OFF), for which we design a control algorithm for changing between ON and OFF.

Let  $t_1^b, \dots, t_n^b, t_1^e, \dots, t_n^e$  be non-negative real values that represent the service times  $t_-^b$  and end-of-service times  $t_-^e$  for  $n$  requests and which satisfy  $0 \leq t_1^b < t_1^e < t_2^b < t_2^e < \dots < t_n^b < t_n^e$ . The input for this problem is given as  $\sigma = \langle (t_1^b, t_1^e), (t_2^b, t_2^e), \dots, (t_n^b, t_n^e) \rangle$ . For this input, the device must be kept ON between times  $t_i^s$  and  $t_i^e$  for each  $i = 1, \dots, n$ .

The state of the device can be switched at an arbitrary time. There is no cost for switching from ON to OFF, while there is a *switching cost*  $a (> 0)$  when switching from OFF to ON. For keeping the ON-state, it takes *running cost* of one unit per unit time.

The strategy of the optimal offline algorithm (OPT) for this problem is clear: If the period between the current request and next one is less than  $a$ , the device is kept ON. Otherwise, it is turned off immediately. Therefore OPT's total cost for the input  $\sigma = \langle (t_1^b, t_1^e), (t_2^b, t_2^e), \dots, (t_n^b, t_n^e) \rangle$  is  $a + \sum_{i=1}^n (t_i^e - t_i^b) + \sum_{i=1}^{n-1} \min\{t_{i+1}^b - t_i^e, a\}$ .

For some  $i$ -th request  $(t_i^s, t_i^e)$ , we consider the sum of the  $i$ -th running cost and the next  $(i + 1)$ -th switching cost. Let ALG be any algorithm. Let  $u$  be the period between the end of the request and the start of the next request (i.e.,  $u = t_{i+1}^s - t_i^e$ ). Then the optimal offline algorithm pays  $t_i^e - t_i^s + \min\{u, a\}$ . If ALG turns the device off after  $v (< u)$  standby time, the cost is  $t_i^e - t_i^s + v + a$ . Otherwise, it must pay  $t_i^e - t_i^s + u$ . In each case, the smaller  $t_i^e - t_i^s$  is, the worse the competitive ratio becomes. Therefore, from the standpoint of competitive analysis, it is enough to consider that usage times of the device (i.e.,  $t_i^e - t_i^s$  for each  $i$ ) are tiny. We call this the *short-job assumption*. On the basis of the above discussion, we redefine this problem as follows:

Let  $t_1, \dots, t_n$  be non-negative real values satisfying  $0 \leq t_1 < \dots < t_n$  representing the time of service of the device for  $n$  requests. An input is given as  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$ . We do nothing if the state is ON at  $t_i$  ( $i = 1, \dots, n$ ), and should turn a device ON if it is OFF at that time.

For a given input  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$  ( $n$  may be  $\infty$ ), the action of an algorithm is determined by a sequence  $\langle w_1, w_2, \dots, w_n \rangle$ , where  $w_i$  is standby time after  $i$ th request is leaving. In other words, the problem is how to determine  $w_i$  from  $\langle t_1, t_2, \dots, t_i \rangle$ . For each  $i = 2, \dots, n$ , let  $u_i = t_i - t_{i-1}$  be an *idle period*. OPT's cost for this redefined problem is  $\text{OPT}(\sigma) = a + \sum_{i=1}^{n-1} \min\{t_{i+1} - t_i, a\}$ . We denote ALG's cost for input  $\sigma$  by  $\text{ALG}(\sigma)$  and the competitive ratio of ALG for  $\sigma$  by  $\mathcal{R}_{\text{ALG}}(\sigma) = \text{ALG}(\sigma)/\text{OPT}(\sigma)$ . Let  $\Sigma$  be the set of whole inputs  $\sigma$ . For a subset  $\Sigma' \subseteq \Sigma$ , we define  $\mathcal{R}_{\text{ALG}}(\Sigma') = \sup_{\sigma \in \Sigma'} \mathcal{R}_{\text{ALG}}(\sigma)$ . And we represent  $\mathcal{R}_{\text{ALG}} = \mathcal{R}_{\text{ALG}}(\Sigma)$ , which is the (worst-case) competitive ratio of ALG.

### 3 The decrease and rest approach

#### 3.1 Decrease and reset algorithm (DRA)

We propose an algorithm which decreases the standby time gradually when the frequency of the device usage becomes low.

*Decrease and Reset Algorithm (DRA)*

Let  $x_1, x_2, \dots$ , be an infinite non-increasing sequence of non-negative values. In DRA,  $w_i = x_{f(i)}$  such that

$$f(i) = \begin{cases} f(i - 1) + 1 & \text{if } u_i \geq a \text{ and } i \neq 1, \\ 1 & \text{otherwise.} \end{cases}$$

If  $x_i = a$  for all  $i$ , DRA is equivalent to *OWCR*. Setting  $x_i$  be larger than  $a$  is clearly wasteful, and hence we consider cases such that  $x_i \leq a$  for all  $i = 1, 2, \dots$ . From a simple observation we see that  $x_1$  gives a lower bound of  $\mathcal{R}_{\text{DRA}}$ :

**Observation 1**  $\mathcal{R}_{\text{DRA}} \geq 1 + a/x_1$ .

**Proof** Let  $m$  be an integer. For an input  $\sigma = \langle x_1, 2x_1, \dots, mx_1 \rangle$ , *OPT*'s total cost is  $a + (m - 1)x_1$  and DRA's total cost is  $m(a + x_1)$ . Thus the competitive ratio of them is the following:

$$\mathcal{R}_{\text{DRA}} \geq \mathcal{R}_{\text{DRA}}(\sigma) = \frac{m(a + x_1)}{a + (m - 1)x_1} \xrightarrow{(m \rightarrow \infty)} 1 + \frac{a}{x_1}.$$

□

From this observation, it follows that  $x_1$  cannot be much smaller than  $a$ , otherwise  $\mathcal{R}_{\text{DRA}}$  becomes very large. In other words, if the difference between  $a$  and  $x_1$  is small, the effect to  $\mathcal{R}_{\text{DRA}}$  is not so large. Thus we relax the worst-case competitive ratio from 2 to  $2 + \epsilon$  for small  $\epsilon > 0$ , i.e., we let  $x_1 = a/(1 + \epsilon)$ .

The above observation is easily extended to the other values  $x_2, x_3, \dots$ , as follows.

**Observation 2** For any integer  $k$ ,

$$\mathcal{R}_{\text{DRA}} \geq \frac{ka + \sum_{i=1}^k x_i}{(k - 1)a + x_k}.$$

**Proof** Let  $m$  be an integer and  $t_1, t_2, \dots, t_{mk}$  be a sequence such that

$t_1 = a$  and if  $i = 1 \pmod k$  then  $t_i = t_{i-1} + x_k$ , otherwise  $t_i = t_{i-1} + a$ . For input  $\sigma = \langle t_1, t_2, \dots, t_{mk} \rangle$ , DRA sets  $w_i = x_{g(i)}$  for all  $i$ , where  $g(i) = ((i - 1) \pmod k) + 1$ . *OPT*'s total cost is  $a + m(k - 1)a + (m - 1)x_k$  and DRA's total cost is  $mka + m \sum_{i=1}^k x_i$ . Thus the competitive ratio of them is the following:



$$\mathcal{R}_{\text{DRA}} \geq \mathcal{R}_{\text{DRA}}(\sigma) = \frac{mka + m \sum_{i=1}^k x_i}{a + m(k-1)a + (m-1)x_k} \xrightarrow{(m \rightarrow \infty)} \frac{ka + \sum_{i=1}^k x_i}{(k-1)a + x_k}.$$

So this observation is satisfied. □

Our upper bound of the competitive ratio is  $2 + \epsilon$ , and thus, the following inequalities must be satisfied for every  $k = 1, 2, \dots$ :

$$\frac{ka + \sum_{i=1}^k x_i}{(k-1)a + x_k} \leq 2 + \epsilon.$$

Solving this equation for  $x_k$ , we have

$$x_k \geq \frac{1}{1 + \epsilon} \left( (2 + \epsilon)a + \sum_{i=1}^{k-1} x_i \right) - ka.$$

By elementary induction, we obtain

$$x_k \geq -\epsilon \left( \frac{2 + \epsilon}{1 + \epsilon} \right)^k a + (1 + \epsilon)a. \tag{1}$$

This is a necessary condition for keeping the competitive ratio less than or equal to  $2 + \epsilon$ . But this condition is not sufficient to guarantee optimality within the  $\epsilon$  bound. We propose next an algorithm that sets exact values for  $x_2, x_3, \dots$ , to guarantee optimality within the  $\epsilon$  bound.

### 3.2 How to set the coefficients for “optimality”

Before turning to this problem, we need to define what “optimal” means here. Our motivation is to give a better algorithm for slack systems. Thus we introduce a measure, “slackness degree” for representing the slackness of input sequences. For an input sequence  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$ , request  $i$  is called a *busy request* if  $u_i \leq a$  or a *slack request*, otherwise. The first request is neither busy nor slack one. We denote the number of slack requests in  $\sigma$  by  $s(\sigma)$ , and that of busy requests in  $\sigma$  by  $b(\sigma)$ .

**Definition 1** For an input  $\sigma$ , if  $s(\sigma)/b(\sigma) \geq d$  ( $b(\sigma) \neq 0$ ) for a real number  $d \geq 0$ ,  $\sigma$  is called  $d$ -slack. The slackness degree  $d(\sigma)$  is defined as the maximum  $d$  such that  $\sigma$  is  $d$ -slack.

The slackness degree describes how busy the inputs are. Clearly if  $d(\sigma)$  is larger,  $\sigma$  has more slack. We will optimize DRA under the assumption that an input is  $d$ -slack without knowing the value of  $d$ .

We consider asymptotic performance, and assume that  $\sigma$  is large enough. In other words  $\sigma$  has a sufficient number of busy requests, i.e.,  $b(\sigma) = \omega(1)$  if  $b(\sigma) \neq 0$ .

Note that for  $b(\sigma) = 0$  (i.e., all arrivals are slack), we can easily get the upper bound of the competitive ratio of  $1 + \frac{\sum_{i=1}^{|\sigma|} x_i}{|\sigma|a}$ , which is close to 1 when  $|\sigma|$  is large and  $\lim_{i \rightarrow \infty} x_i = 0$ . This case is so particular that we ignore it in the following. We will show that it is sufficient to consider inputs which end with a busy request:

For a detailed analysis, let us separate an input  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$  into some  $(b(\sigma)$  or  $b(\sigma) + 1)$  blocks as follows. Assume that  $t_{b_1}, t_{b_2}, \dots, t_{b_{b(\sigma)}}$  ( $0 \leq b_1 < b_2 < \dots < b_{b(\sigma)} \leq n$ ) are the busy requests in  $\sigma$ . The blocks are defined as  $B_1 = \{t_1, \dots, t_{b_1}\}$ ,  $B_2 = \{t_{b_1+1}, \dots, t_{b_2}\}$ ,  $\dots$ ,  $B_{b(\sigma)} = \{t_{b_{b(\sigma)-1}+1}, \dots, t_{b_{b(\sigma)}}\}$ . If  $b_{b(\sigma)} < n$ , then  $B_{b(\sigma)+1} = \{t_{b_{b(\sigma)}+1}, \dots, t_n\}$  also exists. For analyzing the worst-case competitive ratio we will show that the final block  $B_{b(\sigma)+1}$  can be ignored even if it exists. Let  $\beta(\sigma)$  be the number of blocks in  $\sigma$ . (Then  $\beta(\sigma) = b(\sigma)$  or  $b(\sigma) + 1$ .) Let  $s(B_i)$  be the number of slack requests in block  $B_i$ .

**Lemma 1** *If  $s(B_{b(\sigma)}) \leq s(B_{b(\sigma)+1}) - 2$  holds, then  $\mathcal{R}_{\text{DRA}}(\sigma) \leq \mathcal{R}_{\text{DRA}}(\sigma')$ , where  $\sigma'$  is obtained from  $\sigma$  by exchanging  $t_{b_{b(\sigma)}}$  with  $t_{b_{b(\sigma)+1}}$ , i.e.,  $\sigma' = \langle t_1, \dots, t_{b_{b(\sigma)}-1}, t_{b_{b(\sigma)+1}}, t_{b_{b(\sigma)}}, t_{b_{b(\sigma)}+2}, \dots, t_n \rangle$ . (Note that  $t_{b_{b(\sigma)+1}}$  is a slack request from  $s(B_{b(\sigma)}) (\geq 2)$ .)*

**Proof** The competitive ratio of DRA for  $\sigma'$  is

$$\mathcal{R}_{\text{DRA}}(\sigma') = \frac{\text{DRA}(\sigma) - x_{s(B_{b(\sigma)+1})} + x_{s(B_{b(\sigma)})+2}}{\text{OPT}(\sigma) - x_{s(B_{b(\sigma)})+1} + x_{s(B_{b(\sigma)})+2}}$$

Since  $x_1, x_2, \dots$  is a non-increasing sequence and  $s(B_{b(\sigma)}) \leq s(B_{b(\sigma)+1}) - 2$ ,  $x_{s(B_{b(\sigma)})+2} \geq x_{s(B_{b(\sigma)+1})}$  and  $x_{s(B_{b(\sigma)})+1} \geq x_{s(B_{b(\sigma)})+2}$  hold. Thus we get

$$\mathcal{R}_{\text{DRA}}(\sigma') = \frac{\text{DRA}(\sigma) - x_{s(B_{b(\sigma)+1})} + x_{s(B_{b(\sigma)})+2}}{\text{OPT}(\sigma) - x_{s(B_{b(\sigma)})+1} + x_{s(B_{b(\sigma)})+2}} \geq \frac{\text{DRA}(\sigma)}{\text{OPT}(\sigma)} = \mathcal{R}_{\text{DRA}}(\sigma).$$

□

**Lemma 2** *For any  $d \geq 0$  and sufficiently long inputs, there exists an input which finishes with a busy request and gives the worst competitive ratio in the same slackness degree  $d$ .*

**Proof** By Lemma 1 for a  $d$ -slack input  $\sigma$  we can shift the last busy request later as long as the last two blocks satisfy  $s(B_{b(\sigma)}) \leq s(B_{b(\sigma)+1}) - 2$  without decreasing the competitive ratio (Operation 1). We can clearly exchange the two subsequences in  $\sigma$  which begin with a slack request and end with a busy request without changing the competitive ratio (Operation 2).

When we apply Operation 1 and Operation 2 for a sufficiently long input  $\sigma$  repeatedly and let the result be  $\sigma^*$ , which is  $d$ -slack and gives the worst competitive ratio, approximately we can assume that  $\sigma^*$  finishes with a busy request. □

**Lemma 3** *For any input  $\sigma$ , if each  $x_i$  satisfies inequality (1) then  $\mathcal{R}_{\text{DRA}}(\sigma) \leq 2 + \varepsilon$ .*

**Proof** In Observations 1 and 2, the given input is clearly the worst for the competitive ratio among one-block input  $\sigma$  (i.e.,  $\sigma$  includes one busy requests at the end) and the slackness degree is fixed. This means  $\mathcal{R}_{\text{DRA}} \leq 2 + \epsilon$  for any one-block input. From Lemma 2,  $\mathcal{R}_{\text{DRA}} \leq 2 + \epsilon$  for any long enough input.  $\square$

**Lemma 4** For a sufficiently long input  $\sigma$ , if  $\sigma$  is  $d$ -slack ( $d > 0$ ) and  $x_1, x_2, \dots (x_i \leq a)$  satisfy inequality (1), the following inequality holds:

$$\mathcal{R}_{\text{DRA}}(\sigma) \leq 1 + \frac{1}{d} + \frac{\sum_{i=1}^{\infty} x_i}{ad}. \tag{2}$$

And the equality holds for  $d \geq h - 1$  where  $h = \min\{i \mid x_i = 0\}$ .

**Proof** To analyze the worst-case input, we define  $\sigma(k)$  as an input in which one busy request arrives after  $(k - 1)$  slack requests, where  $k = 1, 2, \dots$ , is any positive integer. Then we find that all the worst-case input instances are described as the combination of  $\sigma(k)$  by Lemma 2. Let the combination of them be  $\sigma_w$ , which can be represented by a sequence of  $\sigma(\cdot)$ , i.e.,  $\sigma_w = \sigma(f(1))\sigma(f(2)) \dots \sigma(f(n))$  where  $n = b(\sigma_w)$ , and each  $f(i)$  is a positive integer ( $i = 1, \dots, n$ ).

Against this input, DRA must pay the switching cost for all the requests. The cost of DRA for  $\sigma_w$  is  $a + \sum_{i=1}^n \left\{ \sum_{j=1}^{f(i)} (x_j + a) \right\} + x_1$ . OPT keeps the ON-state during  $x_{f(i)}$  for the last input in each  $\sigma(f(i))$  and switches to OFF immediately for the other inputs. The cost is  $a + \sum_{i=1}^n x_{f(i)} + \sum_{i=1}^n (f(i) - 1)a$ . Therefore the competitive ratio is

$$\begin{aligned} \mathcal{R}_{\text{DRA}}(\sigma_w) &= \frac{a + \sum_{i=1}^n \left( \sum_{j=1}^{f(i)} (x_j + a) \right) + x_1}{a + \sum_{i=1}^n x_{f(i)} + \sum_{i=1}^n (f(i) - 1)a} \\ &\leq \frac{a + \sum_{i=1}^n \sum_{j=1}^{f(i)-1} x_j + \sum_{i=1}^n f(i)a + x_1}{a + \sum_{i=1}^n (f(i) - 1)a} \\ &\leq \frac{a + n \sum_{i=1}^{\infty} x_i + \sum_{i=1}^n (f(i) - 1)a + an + x_1}{a + \sum_{i=1}^n (f(i) - 1)a}. \end{aligned}$$

Since  $\sum_{i=1}^n (f(i) - 1)/n = s(\sigma_w)/b(\sigma_w) \geq d$  and  $\sigma_w$  is  $d$ -slack, we have

$$\begin{aligned} \mathcal{R}_{\text{DRA}}(\sigma) \leq \mathcal{R}_{\text{DRA}}(\sigma_w) &\leq 1 + \frac{\sum_{i=1}^{\infty} x_i + a + x_1/n}{a/n + ad} \\ &\xrightarrow{(n \rightarrow \infty)} 1 + \frac{1}{d} + \frac{\sum_{i=1}^{\infty} x_i}{ad}. \end{aligned}$$

The inequalities are tight when  $\sum_{i=1}^n x_{f(i)} = 0$ , and such input exists only when  $\sum_{i=1}^n f(i)/n \geq h$ . Thus for a sufficiently long input when  $d \geq h - 1$ , we find that the bound is tight.  $\square$

Note that even for 0-slack inputs ( $s(\sigma) = 0$ ), if  $x_1, x_2, \dots$  satisfy (1), the competitive ratio is guaranteed to be  $2 + \epsilon$  according to Observation 1.

We get the upper bound of the worst competitive ratio with parameter  $d$ . To minimize it, we should minimize each  $x_i$  such that they satisfy (1).

**Theorem 1** *We set the coefficients  $x_i$  as  $x_i = \max\{-\epsilon((2 + \epsilon)/(1 + \epsilon))^i a + (1 + \epsilon)a, 0\}$ . Then for any sufficiently long  $d$ -slack input  $\sigma$ , DRA guarantees the following competitive ratio:*

$$\mathcal{R}_{\text{DRA}}(\Sigma_d) = \min \left\{ 1 + \frac{1}{d} + \frac{\sum_{i=1}^{h-1} x_i}{ad}, 2 + \epsilon \right\}, \tag{3}$$

where  $\Sigma_d$  is the set of sufficiently long  $d$ -slack inputs, and  $h = \lfloor (\log(1 + \epsilon) - \log \epsilon) / (\log(2 + \epsilon) - \log(1 + \epsilon)) \rfloor + 1$ .

**Proof** Let  $h$  be defined as in Lemma 4. Then the value of  $h$  is obtained as shown above. From Lemmas 4 and 3 we get

$$\mathcal{R}_{\text{DRA}}(\Sigma_d) \leq \min \left\{ 1 + \frac{1}{d} + \frac{\sum_{i=1}^{\infty} x_i}{ad}, 2 + \epsilon \right\}.$$

To optimize the competitive ratio we should minimize each  $x_i$  in range of satisfying inequality (1). So we get

$$x_i = \begin{cases} -\epsilon \left( \frac{2 + \epsilon}{1 + \epsilon} \right)^i a + (1 + \epsilon)a & \text{if } i < h, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

This means  $\sum_{i=1}^{\infty} x_i = \sum_{i=1}^{h-1} x_i$ .

Furthermore, from Lemma 4, when  $d \geq h - 1$  there are inputs which hold the equation in (2) tightly. On the other hand, from Lemma 3, when  $d < h - 1$ , there are inputs such that DRA uses only  $x_1, \dots, x_d$  (i.e., they satisfy (1) tightly.) and then achieve  $2 + \epsilon$ -competitive ratio tightly. Therefore we obtain the desired equation (3). □

From this, we will call the DRA satisfying the condition of Theorem 1 *the optimal DRA (ODRA)*.

**Corollary 1** *For the value that  $0 < \epsilon < 0.2$ ,*

$$\mathcal{R}_{\text{ODRA}} \leq \min \left\{ 1 + \frac{(1 + \epsilon)^2 + 2(1 + \epsilon) \log \frac{1}{\epsilon}}{d}, 2 + \epsilon \right\}.$$

We also get such a heuristic bound, but skip the details of proof. If  $d \rightarrow \infty$  then  $\mathcal{R}_{\text{ODRA}} \rightarrow 1$ . Therefore we confirm that the competitive ratio is close to 1 when the frequency of requests within any time period is small enough.

## 4 Queuing analysis

### 4.1 Analysis

In this section, we analyze the cost performance of DRA using queuing theory. We consider a huge space in a public building, called a system hereafter. Customers arrive at the system, stay for a while, and then leave the system. An on-off device is equipped with the system. At first, the device is turned off, and then turned on if the first customer arrives at the system. The device is kept on as long as there exists at least one customer in the system. When all customers leave the system, the device is turned off according to DRA whose coefficients  $\{x_i : i = 1, 2, \dots\}$  are given by (4). In the following, we say the system is in ON (resp. OFF) state when the device is on (resp. off).

We assume that customers arrive at the system according to a Poisson process with rate  $\lambda$ . The sojourn time of a customer is independently and identically distributed (i.i.d.) with a general distribution with mean  $1/\mu$ . As we mentioned before, the system capacity is infinity. Then the system we consider here is an M/G/ $\infty$  queuing model.

In the M/G/ $\infty$  model, the busy period is defined as the time interval during which the number of customers in the system is greater than zero, while in the idle period, no customers are in the system. For analytical simplicity, we assume that the system is in equilibrium at time 0, and that the first busy period starts at time 0. Let  $B_n$  and  $I_n$  denote the  $n$ th busy period of the system and the  $n$ th idle period, respectively. Note that both busy periods and idle periods are i.i.d., and hence independent of  $n$ . The mean busy period and the mean idle period of the M/G/ $\infty$  system are given by

$$E[B_n] = \frac{e^\rho - 1}{\lambda} (\equiv E[B]), \quad E[I_n] = \frac{1}{\lambda} (\equiv E[I]), \tag{5}$$

respectively, where  $\rho = \lambda/\mu$ . We define the  $n$ th cycle as the time interval consisting of  $B_n$  and  $I_n$ .

The power control process under DRA with coefficients (4) evolves as follows. When the first busy period  $B_1$  starts, the initial power cost  $a$  is required. During the busy period, the power cost per unit time is one. When  $B_1$  ends, the system is kept ON state for the standby time of  $x_1$ . Note that  $x_1$  is the power cost of the first idle period  $I_1$ . If  $I_1 > a$ , the next standby time for  $I_2$  is set to  $x_2$ . If  $I_1 \leq a$ , then the standby time for  $I_2$  is initialized to  $x_1$ . Similarly, if  $I_1 > a$  and  $I_2 > a$ , then the the standby time for  $I_3$  is set to  $x_3$ , while if  $I_1 > a$  and  $I_2 \leq a$ , the standby time for  $I_3$  is initialized to  $x_1$ , and so on. In the following, the time interval from the beginning of the busy period with  $x_1$  standby time to the end of the idle period which is smaller than  $a$  is referred to as the reset interval.

Let  $L (\geq 1)$  denote the number of cycles in a reset interval. Consider the amount of power consumption during a reset interval. When the number of cycles in the reset interval is  $L = k$ , the amount of power consumption is given by the power

consumption for  $k$  busy periods and  $k$  standby times. Let  $T_k$  denote the total amount of power consumption of the reset interval consisting of  $k$  cycles. We obtain

$$T_k = \begin{cases} \sum_{i=1}^{k-1} x_i + I_k \cdot 1_{\{I_k \leq x_k\}} + x_k \cdot 1_{\{x_k \leq I_k \leq a\}}, & k = 1, 2, \dots, h - 1, \\ \sum_{i=1}^{h-1} x_i, & k \geq h, \end{cases}$$

where  $1_\chi$  is the indicator function of event  $\chi$ . Then we have the following lemma.

**Lemma 5** *The mean of the total amount of power consumption of a reset interval  $E[T_L]$  is given by*

$$\begin{aligned} E[T_L] &= \epsilon a(2 + \epsilon)e^{-\lambda a} (1 - e^{-\lambda a})^{h-2} \\ &\quad - \epsilon a(2 + \epsilon)(1 - e^{-\lambda a}) \frac{(2 + \epsilon)e^{-\lambda a}}{1 + \epsilon - (2 + \epsilon)e^{-\lambda a}} \left\{ 1 - \left( \frac{2 + \epsilon}{1 + \epsilon} \cdot e^{-\lambda a} \right)^{h-2} \right\} \\ &\quad + (1 + \epsilon)a \cdot \frac{e^{-\lambda a}}{1 - e^{-\lambda a}} \left\{ 1 - (h - 1)e^{-\lambda a(h-2)} + (h - 2)e^{-\lambda a(h-1)} \right\} \\ &\quad + \frac{1}{\lambda} \cdot \frac{1 - e^{-\lambda a(h-1)}}{1 - e^{-\lambda a}} - \sum_{k=1}^{h-1} e^{-\lambda a(k-1)} \left( \frac{1}{\lambda} e^{-\lambda x_k} + x_k e^{-\lambda a} \right) \\ &\quad + \left[ \epsilon a(2 + \epsilon) \left\{ 1 - \left( \frac{2 + \epsilon}{1 + \epsilon} \right)^{h-1} \right\} + (1 + \epsilon)a(h - 1) \right] \cdot e^{-\lambda a(h-1)}. \end{aligned} \tag{6}$$

**Proof** Note that the event  $\{L = k\}$  ( $k = 1, 2, \dots$ ) is such that the first to  $(L - 1)$ st idle periods are greater than  $a$  and the  $L$ th idle period is smaller than or equal to  $a$ . Define the following joint probability density function

$$\pi_k(y)dy = \Pr \{L = k, y < I_k \leq y + dy\}, \quad k = 1, 2, \dots \tag{7}$$

From the Poisson arrival assumption, we obtain

$$\pi_k(y)dy = e^{-\lambda a(k-1)} \cdot \lambda e^{-\lambda y} dy, \quad k = 1, 2, \dots \tag{8}$$

Let  $W(k)$  denote the total amount of power consumption for a reset interval which consists of  $k$  cycles.  $W(k)$  is given by

$$W(k) = (k - 1)a + a \cdot 1_{\{x_k < I_k \leq a\}} + \sum_{i=1}^k B_i + T_k.$$

Taking the mean of  $W(L)$  yields

$$\begin{aligned} E[W(L)] &= E \left[ a(L - 1) + a \cdot 1_{\{x_k < I_k \leq a\}} + \sum_{i=1}^L B_i + T_L \right] \\ &= a(E[L] - 1) + aE[1_{\{x_k < I_k \leq a\}}] + E[L]E[B] + E[T_L]. \end{aligned} \tag{9}$$

From (8),  $E[L]$  is obtained as

$$\begin{aligned}
 E[L] &= \sum_{k=1}^{\infty} k \Pr\{L = k\} \\
 &= \sum_{k=1}^{\infty} k \int_0^a \pi_k(y) dy \\
 &= \sum_{k=1}^{\infty} k e^{-\lambda a(k-1)} (1 - e^{-\lambda a}) \\
 &= \frac{1}{1 - e^{-\lambda a}}.
 \end{aligned}
 \tag{10}$$

We also have

$$\begin{aligned}
 E[1_{\{x_k < I_k \leq a\}}] &= \sum_{k=1}^{\infty} \int_{x_k}^a e^{-\lambda a(k-1)} \lambda e^{-\lambda x} dx \\
 &= \sum_{k=1}^{h-1} e^{-\lambda\{a(k-1)+x_k\}} - \frac{e^{-\lambda a} - e^{-\lambda a(h-1)}}{1 - e^{-\lambda a}}.
 \end{aligned}
 \tag{11}$$

$E[T_L]$  is expressed as

$$\begin{aligned}
 E[T_L] &= \sum_{k=1}^{\infty} \int_0^a \left\{ \sum_{i=1}^{k-1} x_i + y \cdot 1_{\{I_k \leq x_k\}} + x_k \cdot 1_{\{x_k \leq I_k \leq a\}} \right\} \pi_k(y) dy \\
 &= \sum_{k=2}^{h-1} \left( \sum_{i=1}^{k-1} x_i \right) e^{-\lambda a(k-1)} (1 - e^{-\lambda a}) + \sum_{k=1}^{h-1} \int_0^{x_k} y \cdot \lambda e^{-\lambda y} \cdot e^{-\lambda a(k-1)} dy \\
 &\quad + \sum_{k=h}^{\infty} \left( \sum_{i=1}^{k-1} x_i \right) e^{-\lambda a(k-1)} (1 - e^{-\lambda a}) + \sum_{k=1}^{h-1} \int_{x_k}^a x_k \cdot \lambda e^{-\lambda y} \cdot e^{-\lambda a(k-1)} dy.
 \end{aligned}
 \tag{12}$$

Substituting (4) into (12) yields (6). □

Let  $Q_{\text{DRA}}$  denote the mean power-consumption cost per unit time. Then we obtain the following theorem.

**Theorem 2**  $Q_{\text{DRA}}$  is given by

$$Q_{\text{DRA}} = \frac{1}{e^\rho} \left[ \lambda a \left\{ (1 - e^{-\lambda a}) \cdot \sum_{k=1}^{h-1} e^{-\lambda\{a(k-1)+x_k\}} + e^{-\lambda a(h-1)} \right\} + e^\rho - 1 + \lambda (1 - e^{-\lambda a}) E[T_L] \right],
 \tag{13}$$

where  $E[T_L]$  is given by (6).

**Proof** Let  $R$  denote the reset interval. Using  $L$ , the number of cycles in a reset interval, we obtain

$$R = \sum_{n=1}^L (B_n + I_n),$$

where  $B_n$  and  $I_n$  are the  $n$ th busy period and  $n$ th idle period, respectively. Note that  $\{B_n\}$  and  $\{I_n\}$  are independent.

Suppose  $L = k$ . Let  $I_n^*(s)$  denote the Laplace-Stieltjes transform (LST) of  $I_n$ . Since  $I_n$  for  $n = 1, 2, \dots, k - 1$  is greater than  $a$  and  $I_k$  is smaller than or equal to  $a$ ,  $I_n^*(s)$  is yielded as

$$I_n^*(s) = \begin{cases} \frac{\lambda}{s+\lambda} e^{-(s+\lambda)a}, & n = 1, 2, \dots, k - 1, \\ \frac{\lambda}{s+\lambda} (1 - e^{-(s+\lambda)a}), & n = k. \end{cases} \tag{14}$$

We define  $I^*(k, s)$  as

$$I^*(k, s) = \int_0^\infty e^{-sx} \Pr \left\{ L = k, x < \sum_{n=1}^k I_n \leq x + dx \right\}$$

$I^*(k, s)$  is the LST of the total idle period of the reset interval which consists of  $k$  cycles. Using  $I_n^*(s)$ , we obtain

$$I^*(k, s) = \prod_{n=1}^k I_n^*(s) = \left( \frac{\lambda}{s + \lambda} \right)^k e^{-(s+\lambda)a(k-1)} (1 - e^{-(s+\lambda)a}). \tag{15}$$

Then, the mean amount of idle periods in a reset interval is given by

$$E \left[ \sum_{n=1}^L I_n \right] = \sum_{k=1}^\infty \left( -\frac{d}{ds} I^*(k, s) \right)_{s=0} = \frac{1}{\lambda(1 - e^{-\lambda a})}. \tag{16}$$

The mean amount of busy periods in a reset cycle is yielded as

$$E \left[ \sum_{n=1}^L B_n \right] = \sum_{k=1}^\infty kE[B] e^{-\lambda a(k-1)} (1 - e^{-\lambda a}) = \frac{e^\rho - 1}{\lambda(1 - e^{-\lambda a})}, \tag{17}$$

where  $\rho = \lambda/\mu$ .

Finally, the mean reset interval  $E[R]$  is given by

$$E[R] = E \left[ \sum_{n=1}^L B_n \right] + E \left[ \sum_{n=1}^L I_n \right] = \frac{e^\rho}{\lambda(1 - e^{-\lambda a})}. \tag{18}$$

Now consider the mean power consumption per unit time,  $Q_{\text{DRA}}$ . Note that reset intervals are i.i.d., and that the amount of power consumption during the reset interval is also i.i.d. Therefore, we have  $Q_{\text{DRA}} = E[W(L)]/E[R]$  from the renewal-reward theorem [39]. From (9), (10), (11) and (17), we obtain (13).  $\square$



### 4.2 Simplified reset algorithms

It may be advantageous to consider simplified variants of DRA; such algorithms may have worse competitive ratio, but still may work well on average. We consider here the following family of algorithms: Given parameter  $k$  let

$ALG(k)$  be the DRA algorithm where  $x_1 = x_2 = \dots = x_k = a$ ,  $x_{k+1} = x_{k+2} = \dots = 0$ .

The worst-case competitive ratio of  $ALG(k)$  is  $2 + \frac{1}{k}$ , which can be obtained easily. In what follows we consider three cases:

- (a)  $ALG(\infty)$  (all  $x_i$  are equal to  $a$ )
- (b)  $ALG(1)$
- (c)  $ALG(10)$ .

Note that  $ALG(\infty)$  is equal to  $OWCR$ .

Let  $Q_\xi$  denote the mean of the power-consumption cost per unit time when the algorithm of  $\xi \in \{OPT, ALG(\infty), ALG(1), ALG(10)\}$  is employed.  $Q_\xi$ 's can be derived in a straightforward manner, and we obtain

$$\begin{aligned}
 Q_{OPT} &= 1 - e^{-(a\lambda+\rho)}, \\
 Q_{ALG(\infty)} &= 1 + (a\lambda - 1)e^{-(a\lambda+\rho)}, \\
 Q_{ALG(1)} &= 1 + 2(a\lambda - 1)e^{-(a\lambda+\rho)} - (a\lambda - 1)e^{-(2a\lambda+\rho)}, \\
 Q_{ALG(10)} &= 1 + (a\lambda - 1)e^{-(a\lambda+\rho)} + (a\lambda - 1)e^{-(ka\lambda+\rho)} - (a\lambda - 1)e^{-(k+1)a\lambda+\rho}.
 \end{aligned}$$

In Table 3 we display competitive ratios of  $ALG(\infty)$ ,  $ALG(1)$ ,  $ALG(10)$ , as well as  $ODRA$  for  $\epsilon = 0.1, 0.01$  and  $0.001$  given different degrees of slackness. We use the basic set of parameters shown in Table 2 and calculated the average power consumption per unit time of each algorithm and the competitive ratio with  $Q_{opt}$ .

Next, we conducted experiments of Monte Carlo simulation in order to validate the results obtained through the analysis. Our simulation program was

**Table 2** Basic parameters

| Parameter                        | Value                       |
|----------------------------------|-----------------------------|
| Value of $a$                     | 1, 3, 10 [unit]             |
| $ALG(10)$ parameter $k$          | 10                          |
| Value of $\epsilon$              | 0.1, 0.01, 0.001            |
| Consuming cost while in ON-state | 1 [unit]                    |
| Customer arrival rate $\lambda$  | 0.001, 0.01, 0.1, 0.5, 0.99 |
| Mean sojourn time $1/\mu$        | 1                           |
| Number of events                 | 100,000                     |
| Number of simulations            | 100                         |

**Table 3** Analytical results of competitive ratios

| $\rho$       | $a$ | $\frac{Q_{ALG(\infty)}}{Q_{OPT}}$ | $\frac{Q_{ALG(1)}}{Q_{OPT}}$ | $\frac{Q_{ALG(10)}}{Q_{OPT}}$ | $\frac{Q_{ODRA}}{Q_{OPT}}$ |                   |                    |
|--------------|-----|-----------------------------------|------------------------------|-------------------------------|----------------------------|-------------------|--------------------|
|              |     |                                   |                              |                               | $\epsilon = 0.1$           | $\epsilon = 0.01$ | $\epsilon = 0.001$ |
| $CR_{worst}$ | 2   | 3                                 | 2.1                          | 2.1                           | 2.01                       | 2.001             |                    |
| 0.001        | 1   | 1.499500167                       | 1.000748917                  | 1.005217540                   | 1.001271437                | 1.002655881       | 1.004223251        |
| 0.001        | 3   | 1.748501000                       | 1.003363768                  | 1.023213298                   | 1.005700441                | 1.011875363       | 1.018824733        |
| 0.001        | 10  | 1.904100076                       | 1.013501416                  | 1.090154187                   | 1.022737696                | 1.046923893       | 1.073583298        |
| 0.01         | 1   | 1.495016667                       | 1.007392352                  | 1.049361599                   | 1.012449439                | 1.025691967       | 1.040288636        |
| 0.01         | 3   | 1.735099997                       | 1.032642545                  | 1.198858383                   | 1.054009659                | 1.108550126       | 1.165205808        |
| 0.01         | 10  | 1.860007391                       | 1.123442676                  | 1.560542526                   | 1.192391007                | 1.354364280       | 1.491272831        |
| 0.1          | 1   | 1.451665557                       | 1.064830612                  | 1.294390205                   | 1.101041447                | 1.186107865       | 1.258010593        |
| 0.1          | 3   | 1.609973435                       | 1.241087434                  | 1.585182262                   | 1.322390871                | 1.483578065       | 1.560710753        |
| 0.1          | 10  | 1.498960659                       | 1.498960659                  | 1.498960659                   | 1.468155678                | 1.493303595       | 1.498345837        |
| 0.5          | 1   | 1.290988353                       | 1.176493358                  | 1.289716429                   | 1.207633090                | 1.268587916       | 1.286176758        |
| 0.5          | 3   | 1.234776464                       | 1.295573382                  | 1.234776547                   | 1.244383268                | 1.234341587       | 1.234673050        |
| 0.5          | 10  | 1.020517708                       | 1.036821277                  | 1.020517708                   | 1.030159311                | 1.021349348       | 1.020598886        |
| 0.99         | 1   | 1.158584136                       | 1.157577489                  | 1.158584000                   | 1.148352929                | 1.156732850       | 1.158383505        |
| 0.99         | 3   | 1.057717729                       | 1.094037776                  | 1.057717729                   | 1.070640985                | 1.058742869       | 1.057806889        |
| 0.99         | 10  | 1.000184576                       | 1.000350501                  | 1.000184576                   | 1.000426862                | 1.000201666       | 1.000186226        |

written in C. For each parameter setting, we performed 100 different simulation runs, each one consisting of 100000 arrival events. Then,  $\hat{Q}_\xi$  ( $\xi \in \{OPT, ALG(\infty), ALG(1), ALG(10), ODRA\}$ ), the estimate of  $Q_\xi$ , was calculated by taking the average of the 100 simulation results.

From Tables 3 and 4 we observe that the differences between the analysis and the simulations ratios are extremely small, which is summarized in Table 5.

We mention that there are two natural algorithms, namely the algorithm “always-on”, which always continues to standby in the on state, and the algorithm “always-off”, which switches off immediately after each request. However, both of these algorithms are not competitive.

### 4.3 Analysis of results

We now provide further analysis of the algorithms in the previous subsection. It is noted that in Sect. 2 “Problem Statement” the power-down problem was formalized under the “short-job assumption” and this assumption was used in Sect. 3 to develop the DRA approach. Figure (1a) depicts competitive ratios of the algorithms over  $\rho$  in the case where the value of  $a$  is large relative to the mean sojourn time  $1/\mu$  (cf. Table 2), thus modeling the “short-job assumption”. To this end, in Fig. 1a we have  $a = 10, 1/\mu = 1$  and  $\epsilon = 0.1$ .

Comparing  $ALG(\infty), ALG(10), ALG(1)$ , and  $ODRA$ , it is clear that  $ODRA$  outperforms both  $ALG(\infty)$  and  $ALG(10)$  for small values of  $\rho$ . When  $\rho$  becomes larger, the

**Table 4** Simulation results of competitive ratios

| $\rho$ | $a$ | $\frac{\hat{Q}_{ALG(\infty)}}{\hat{Q}_{OPT}}$ | $\frac{\hat{Q}_{ALG(1)}}{\hat{Q}_{OPT}}$ | $\frac{\hat{Q}_{ALG(10)}}{\hat{Q}_{OPT}}$ | $\frac{\hat{Q}_{ODRA}}{\hat{Q}_{OPT}}$ |                  |                   |                    |
|--------|-----|---|--|---|--|------------------|-------------------|--------------------|
|        |     |   |  |   |  | $\epsilon = 0.1$ | $\epsilon = 0.01$ | $\epsilon = 0.001$ |
| 0.001  | 1   | 1.499492963                                   | 1.000750893                              | 1.005234781                               | 1.001278115                            | 1.002673568      | 1.004253776       |                    |
| 0.001  | 3   | 1.748494731                                   | 1.003364077                              | 1.022938486                               | 1.005706731                            | 1.011885855      | 1.018841590       |                    |
| 0.001  | 10  | 1.904080771                                   | 1.013526162                              | 1.086857557                               | 1.022823295                            | 1.04704002       | 1.073740246       |                    |
| 0.01   | 1   | 1.495001195                                   | 1.007416410                              | 1.047625321                               | 1.012511045                            | 1.025785455      | 1.040428928       |                    |
| 0.01   | 3   | 1.735080811                                   | 1.032678371                              | 1.180575779                               | 1.054337550                            | 1.108750303      | 1.165435414       |                    |
| 0.01   | 10  | 1.860076853                                   | 1.123428691                              | 1.463281096                               | 1.195298066                            | 1.354960147      | 1.491342028       |                    |
| 0.1    | 1   | 1.451629102                                   | 1.064873144                              | 1.243433987                               | 1.102648834                            | 1.186478676      | 1.258064006       |                    |
| 0.1    | 3   | 1.609907886                                   | 1.241255251                              | 1.499596828                               | 1.335282596                            | 1.486253375      | 1.561186174       |                    |
| 0.1    | 10  | 1.499079473                                   | 1.499134714                              | 1.499052517                               | 1.499094624                            | 1.499025049      | 1.499089089       |                    |
| 0.5    | 1   | 1.290981807                                   | 1.176359960                              | 1.265689059                               | 1.219014586                            | 1.270510235      | 1.286351517       |                    |
| 0.5    | 3   | 1.234641100                                   | 1.295229088                              | 1.242249087                               | 1.257543221                            | 1.236841254      | 1.234870023       |                    |
| 0.5    | 10  | 1.020479491                                   | 1.036753389                              | 1.022105558                               | 1.030423560                            | 1.021337539      | 1.020612829       |                    |
| 0.99   | 1   | 1.158652448                                   | 1.157644732                              | 1.158482857                               | 1.158265291                            | 1.158607584      | 1.158659617       |                    |
| 0.99   | 3   | 1.057636604                                   | 1.093908755                              | 1.061441260                               | 1.071619214                            | 1.058930370      | 1.057789034       |                    |
| 0.99   | 10  | 1.000200917                                   | 1.000380917                              | 1.000200917                               | 1.000538819                            | 1.000305787      | 1.000301569       |                    |

**Table 5** Difference between analysis and simulation

| $\rho$ | $a$ | $\frac{Q_{ALG(\infty)}}{Q_{OPT}}$ | $\frac{Q_{ALG(1)}}{Q_{OPT}}$ | $\frac{Q_{ALG(10)}}{Q_{OPT}}$ | $\frac{Q_{ODRA}}{Q_{OPT}}$ |                  |                   |                    |
|--------|-----|-----------------------------------|------------------------------|-------------------------------|----------------------------|------------------|-------------------|--------------------|
|        |     |                                   |                              |                               |                            | $\epsilon = 0.1$ | $\epsilon = 0.01$ | $\epsilon = 0.001$ |
| 0.001  | 1   | 7.20367E-06                       | 1.97565E-06                  | 1.72414E-05                   | 6.67816E-06                | 1.76866E-05      | 3.05254E-05       |                    |
| 0.001  | 3   | 6.26900E-06                       | 3.09299E-07                  | 2.74812E-04                   | 6.28975E-06                | 1.04919E-05      | 1.68566E-05       |                    |
| 0.001  | 10  | 1.93047E-05                       | 2.47463E-05                  | 3.29663E-03                   | 8.55985E-05                | 1.16127E-04      | 1.56948E-04       |                    |
| 0.01   | 1   | 1.54716E-05                       | 2.40582E-05                  | 1.73628E-03                   | 6.16060E-05                | 9.34883E-05      | 1.40292E-04       |                    |
| 0.01   | 3   | 1.91863E-05                       | 3.58258E-05                  | 1.82826E-02                   | 3.27891E-04                | 2.00177E-04      | 2.29606E-04       |                    |
| 0.01   | 10  | 6.94621E-05                       | 1.39855E-05                  | 9.72614E-02                   | 2.90706E-03                | 5.95867E-04      | 6.91967E-05       |                    |
| 0.1    | 1   | 3.64546E-05                       | 4.25324E-05                  | 5.09562E-02                   | 1.60739E-03                | 3.70811E-04      | 5.34132E-05       |                    |
| 0.1    | 3   | 6.55485E-05                       | 1.67817E-04                  | 8.55854E-02                   | 1.28917E-02                | 2.67531E-03      | 4.75421E-04       |                    |
| 0.1    | 10  | 1.18814E-04                       | 1.74055E-04                  | 9.18580E-05                   | 3.09389E-02                | 5.72145E-03      | 7.43252E-04       |                    |
| 0.5    | 1   | 6.54643E-06                       | 1.33398E-04                  | 2.40274E-02                   | 1.13815E-02                | 1.92232E-03      | 1.74759E-04       |                    |
| 0.5    | 3   | 1.35364E-04                       | 3.44294E-04                  | 7.47254E-03                   | 1.31600E-02                | 2.49967E-03      | 1.96973E-04       |                    |
| 0.5    | 10  | 3.82174E-05                       | 6.78883E-05                  | 1.58785E-03                   | 2.64249E-04                | 1.18085E-05      | 1.39428E-05       |                    |
| 0.99   | 1   | 6.83124E-05                       | 6.72425E-05                  | 1.01143E-04                   | 9.91236E-03                | 1.87473E-03      | 2.76112E-04       |                    |
| 0.99   | 3   | 8.11251E-05                       | 1.29021E-04                  | 3.72353E-03                   | 9.78229E-04                | 1.87501E-04      | 1.78547E-05       |                    |
| 0.99   | 10  | 1.63405E-05                       | 3.04164E-05                  | 1.63405E-05                   | 1.11957E-04                | 1.04121E-04      | 1.15343E-04       |                    |

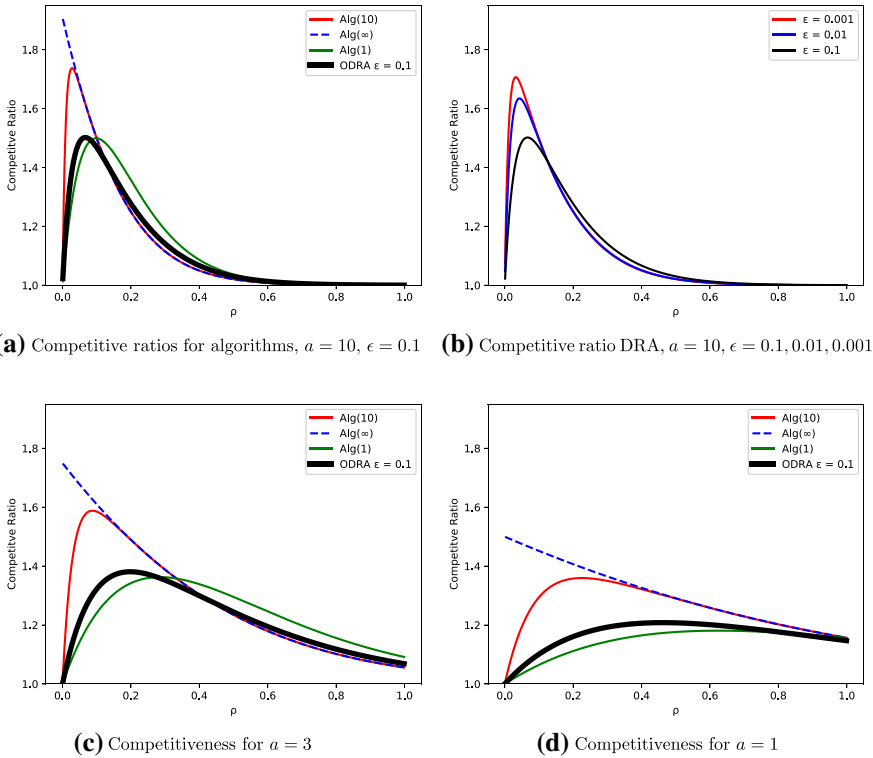


Fig. 1 Competitiveness of algorithms

differences among all the algorithms decrease to the point where there is almost no difference. It is important to recall that we desire the algorithms to be close to optimally-competitive even for worst-case request sequence instances. While *ALG*(1) performs quite well, its worst-case competitive ratio is 3. Only *ODRA* combines a close-to-optimal worst-case competitiveness with good competitiveness across all values of  $\rho$ . Finally, observe the values in the third column of Table 6 which gives averages (i.e. the integrals of the curves): Indeed, *ODRA* exhibits the smallest average, though—as discussed—this is not its most significant feature.

Figure 1b shows the effect of tightening the value of  $\epsilon$  for *ODRA*. To obtain a better worst-case guarantee *ODRA* gives up competitiveness across ranges of  $\rho$  when

Table 6 Average competitiveness

|                         | $a = 1$     | $a = 3$     | $a = 10$    |
|-------------------------|-------------|-------------|-------------|
| <i>ODRA</i>             | 1.168412309 | 1.223852092 | 1.116581260 |
| <i>ALG</i> (1)          | 1.144922345 | 1.237722127 | 1.129922058 |
| <i>ALG</i> (10)         | 1.260466135 | 1.268950391 | 1.127783885 |
| <i>ALG</i> ( $\infty$ ) | 1.303473642 | 1.291111016 | 1.135928400 |

**Table 7** Three State Taper Down Values for  $s_1 = 0.6$  and  $a_1 = 0.4, \mathcal{R} = 1.8$

| $\epsilon = 0.001$ |        | $\epsilon = 0.01$ |        | $\epsilon = 0.1$ |        |
|--------------------|--------|-------------------|--------|------------------|--------|
| $u$                | $q$    | $u$               | $q$    | $u$              | $q$    |
| 0.4993             | 0.4996 | 0.4938            | 0.4959 | 0.4444           | 0.4615 |
| 0.4970             | 0.4996 | 0.4708            | 0.4959 | 0.2460           | 0.4615 |
| 0.4918             | 0.4996 | 0.4193            | 0.4959 | 0                | 0.3417 |
| 0.4799             | 0.4996 | 0.3044            | 0.4959 | 0                | 0      |
| 0.4534             | 0.4996 | 0.0474            | 0.4959 |                  |        |
| 0.3936             | 0.4996 | 0                 | 0.1433 |                  |        |
| 0.2591             | 0.4996 | 0                 | 0      |                  |        |
| 0                  | 0.4708 |                   |        |                  |        |
| 0                  | 0.0390 |                   |        |                  |        |
| 0                  | 0      |                   |        |                  |        |

the value of  $\epsilon$  is chosen to be smaller. Thus there is a tradeoff between the demand on how close to optimality the algorithm in the worst-case should be, and what its overall competitiveness is.

Finally, Fig. 1c, d show characteristics for  $a = 3$  and  $a = 1$ . As is expected, the competitiveness behavior is similar, though competitiveness is somewhat better for small values of  $\rho$ .

### 5 Systems with intermediate states

The use of multiple intermediate states, sometimes called “sleep states” affords better competitiveness; see our papers [7] for systems with one extra state, *i.e.* three-state systems, [8] for systems with finitely many but few states, [6] for systems with a continuous number of states, and our survey paper [9]. In this section we focus on three-state systems; for simplicity we assume that  $a = 1$  and consider one extra state  $INT_1$  with standby cost  $s_1$  and switching cost  $a_1$ , as in Table 1. As shown in [7] for  $s_1 = 0.6$  and  $a_1 = 0.4$ , for example, the competitive ratio  $\mathcal{R}$  for this system is 1.8. (The values  $s_1 = 0.6$  and  $a_1 = 0.4$  are significant, as among all parameter values  $s_1$  and  $a_1$  for  $INT_1$  this choice gives the lowest competitiveness.) Whereas for a two state system  $\mathcal{O}WCR$  switches to OFF after time  $a=1$  to obtain ratio 2, for this three-state system the  $\mathcal{O}WCR$  switches to  $INT_1$  after idle time  $x_1 = 0.9037$  and to OFF after idle time  $x_2 = 1$ . Just as in the two-state case,  $\mathcal{O}WCR$  does not result in good performance in many practical situations, especially in case the device is not used frequently. We will now describe how the DRA approach can be adapted to a system with three states (see also Table 7).

Let  $u_1, u_2, u_2, \dots$  be an infinite sequence of standby times in the ON state and let  $q_1, q_2, q_3, \dots$  be an infinite sequence of standby times in the INT state. So  $u_i$  is the duration of the machine in the ON state and  $q_i$  is the duration in the INT state and after  $u_i + q_i$  time units the machine switches to the OFF state. Since this will be a three state machine, there are two switch times when we have the worst-case cost which occurs

immediately at the switch times. Assuming competitive ratio  $\mathcal{R}$  we utilize two equations that we will use to compute these times.

$$\frac{\sum_{i=1}^{k-1} (u_i + s_1 q_i + 1) + u_k + a_1}{\sum_{i=1}^{k-1} 1 + u_k} = \mathcal{R} + \epsilon \tag{19}$$

$$\frac{\sum_{i=1}^{k-1} (u_i + s_1 q_i + 1) + u_k + s_1 q_k + 1}{\sum_{i=1}^{k-1} 1 + u_k + q_k} = \mathcal{R} + \epsilon \tag{20}$$

Equation (19) is used to compute  $u_k$  after  $k - 1$  slack requests, and that  $u_k$  value is used to compute  $q_k$  in Eq. (20). In the two equations, it is assumed that for state INT,  $s_1 + a_1 \geq 1$ , which can be seen by the offline cost since constants  $a$  and  $d$  are absent in the offline cost. Similarly to the 2 state DRA, we increase the competitive ratio by a small constant  $\epsilon$ , and we attempt to taper the standby duration of the ON and INT state after every slack request, a request that occurs after  $u_i + q_i$ , and the durations reset back to  $u_1$  and  $q_1$  otherwise. We substitute  $R = \mathcal{R} + \epsilon$  and we solve for  $u_k$  and  $q_k$ , we get

$$u_k = \max \left\{ \frac{R(k - 1) - \sum_{i=1}^{k-1} (u_i + s_1 q_i + 1) - a_1}{1 - R}, 0 \right\} \tag{21}$$

$$q_k = \max \left\{ \frac{R(k - 1) + u_k(R - 1) - \sum_{i=1}^{k-1} (u_i + s_1 q_i + 1) - 1}{s_1 - R}, 0 \right\} \tag{22}$$

Since the value of  $u_k$  from (21) is needed to compute the value of  $q_k$  in (22), if we assume that  $u_k > 0$  to get:

$$q_k = \frac{1 - a_1}{R - s_1} \tag{23}$$

So  $q_k$  does not taper down as long as  $u_k > 0$ . Initially, only the duration of the ON state will taper after each slack request while the duration of the machine in state INT does not change. When  $u_k = 0$ , several slack requests have arrived consecutively, and then the value of  $q_k$  will begin tapering down. We will obtain a closed form for  $u_k$ , as was done for the two state system. We can derive (19) to have

$$\sum_{i=1}^{k-1} u_i + s_1 \sum_{i=1}^{k-1} q_i - (1 - R)u_k = (R - 1)(k - 1) - a_1 \tag{24}$$

Since we know that if any  $u_i > 0$ , then the values for  $q_i$  will never taper down so we can substitute  $q_i = \frac{1 - a_1}{R - s_1}$  and then we have

$$\sum_{i=1}^k u_i - Ru_k = (R - 1)k - R + (R - s_1)k \frac{1 - a_1}{R - s_1} \tag{25}$$

Now we can simply substitute  $k = k - l$  in (25) to have

$$\sum_{i=1}^{k-1} u_i - Ru_{k-1} = (R - 1)(k - 1) - R + (R - s_1)(k - 1) \frac{1 - a_1}{R - s_1} \tag{26}$$

If we subtract (26) from (25) we have the following recurrence

$$(1 - R)u_k + Ru_{k-1} = (R - 1) - s_1 \frac{1 - a_1}{R - s_1} \tag{27}$$

Let us substitute the right hand side to  $\phi = (R - 1) - s_1 \frac{1 - a_1}{R - s_1}$

$$u_k - \phi = \frac{R}{R - 1}(u_{k-1} - \phi) \tag{28}$$

In (28), we can see that  $(R)/(R - 1)$  is multiplied  $k$  times so we can rewrite (28)

$$u_k - \phi = \left( \frac{R}{R - 1} \right)^{k-1} (u_1 - \phi) \tag{29}$$

From Eq. (21) from the last section, we can compute  $u_1 = d/(R - 1)$ , substitution that for  $u_1$  we obtain

$$u_k = \max \left\{ \left( \frac{R}{R - 1} \right)^{k-1} (a_1/(R - 1) - \phi) + \phi, 0 \right\} \tag{30}$$

For all  $i < l$  if  $u_i > 0$ , then we know that  $q_i$  does not change, let  $l$  be an index such that  $u_l = 0$  and  $u_{l-1} > 0$ , which can be computed from (30), so the index  $l$  is the first instance such that the duration becomes zero, in which this case the value of  $q_l \leq q_{l-1}$  and this holds for all indices larger than  $l$ . So we will derive a formula for the tapering values for  $q_k$ , similar to how we derived the formula for  $u_k$ , we will start by deriving (20) by replacing  $k$  with  $l$

$$\sum_{i=1}^l u_i + s_1 \sum_{i=1}^l q_i - Ru_l - Rq_l = R(l - 1) - l \tag{31}$$

As done earlier, we will substitute  $l = l - 1$

$$\sum_{i=1}^{l-1} u_i + s_1 \sum_{i=1}^{l-1} q_i - Ru_{l-1} - Rq_{l-1} = R(l - 2) - l + 1 \tag{32}$$

Once again we subtract (31) from (32)

$$u_l + s_1 q_l + R(u_{l-1} - u_l) + R(q_{l-1} - q_l) = R - 1 \tag{33}$$

So as we assumed before,  $u_l = 0$  so we make that substitution into (33)

$$\begin{aligned} s_1 q_l + R u_{l-1} + R(q_{l-1} - q_l) &= R - 1 \\ q_l &= \frac{R - 1 - R(u_{l-1} + q_{l-1})}{s_1 - r} \end{aligned} \quad (34)$$

This  $q_l$  from (34), will be used to solve the recurrence for  $q_l$  values. In order to derive this formula, we will begin with (33) which was obtained by subtracting (31) from (32). We will substitute  $l$  for  $t$  and  $t \geq l$ , so we have

$$s_1 q_t + R q_{t-1} - R q_t = R - 1 \quad (35)$$

The difference between (35) and (33) is that in this case both  $u_t = 0$  and  $u_{t-1} = 0$  since  $t > l$  in which at this point  $u_l = 0$  and  $u_{l+1} = 0$  must be true and  $t > l$  so  $u_t = 0$  and  $u_{t-1} = 0$  must be true as well. Now when we solve for  $q_t$ , we have the following recurrence

$$q_t = \frac{R - 1}{s_1 - R} + \left( \frac{R}{R - a} \right) q_{t-1} \quad (36)$$

Let us substitute  $\chi = \frac{R}{R-s_1}$ , when we solve the recurrence we have

$$q_t = \max \left\{ \chi^{t-l} q_l + \left( \frac{R - 1}{s_1 - R} \right) \left( \frac{\chi^{t-l} - 1}{\chi - 1} \right), 0 \right\} \quad (37)$$

## 6 Conclusions and outlook

With the proliferation of IT devices and the adoption of a smart grid powered by renewables, power-down methods are becoming increasingly relevant. The decrease and reset approach affords close to optimal performance while guaranteeing good performance over a large spectrum of request types. The different types of analysis employed in this paper show that the decrease and reset approach achieves significant improvements for power-down problems. We suggest that decrease and reset should be used in practice across various applications but especially in the context of the electrical grid. To ensure resiliency of a majority-renewables grid our work on the power-down problem demonstrates that online competitive analysis and methods from game theory can be useful tools for existing and future modeling. Further work is under way to see how decrease and reset approach can be adapted to various multi-state systems.

We could further refine our approach by utilizing a budget-based method which keeps a tally of gains and losses as requests are processed. This approach would be similar conceptually to decrease and reset in terms of how the system tapers down, but the adjusted wait times in the budget-based approach would calculate the switch time to be even more cost efficient.



**Acknowledgements** This research has been carried out in collaboration with “Consumer Electronics Network Eco Management” project sponsored by Panasonic Corporation. We would like to thank the project members, Mr. Toshiya Naka, Mr. Hideyuki Yoshida and Mr. Kazuhiro Aizu. We also would like to thank Prof. Hiroshi Fujiwara of Toyohashi University of Technology for his valuable comments on power consumption problems and helpful discussions. Author Wolfgang Bein acknowledges a sabbatical granted by the University of Las Vegas, Nevada to conduct research in Japan. He also acknowledges NSF grant CCR-9821009. The authors are grateful to two anonymous referees for a number of helpful comments and insights that improved the paper.


## References

1. Ackooij, W., Danti Lopez, I., Frangion, A., Lacalandra, F., Tahanan, M.: Large-scale unit commitment under uncertainty: an updated literature survey. *Ann. Oper. Res.* **271**(1), 11–85 (2018)
2. Agarwal, Y., Hodges, S., Chandra, R., Scott, J., Bahl, P., Gupta, R.: Somniloquy: augmenting network interfaces to reduce pc energy usage. In: Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI’09, pp. 365–380, Berkeley, CA, USA, 2009. USENIX Association
3. Agarwal, Y., Savage, S., Gupta, R.: Sleepserver: a software-only approach for reducing the energy consumption of PCs within enterprise environments. In: Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, USENIXATC’10, pp. 22–22, Berkeley, CA, USA, (2010). USENIX Association
4. Albers, S.: Energy-efficient algorithms. *Commun. ACM* **53**(5), 86–96 (2010)
5. Alqahtani, B.J., Patiño-Echeverri, D.: Integrated solar combined cycle power plants: paving the way for thermal solar. *Appl Energy* **169**, 927–936 (2016)
6. Andro-Vasko, J., Avasarala, S.R., Bein, W.: Continuous state power-down systems for renewable energy management. In: Latifi, S. (ed.) *Information Technology: New Generations: 15th International Conference on Information Technology*, vol. 738, pp. 701–707. Springer International Publishing (2018)
7. Andro-Vasko, J., Bein, W., Nyknahad, D., Ito, H.: Evaluation of online power-down algorithms. In: 2015 12th International Conference on Information Technology—New Generations, vol. 1346, pp. 473–478 (2015)
8. Andro-Vasko, J., Bein, W., Pathak, G.: A heuristic for state power down systems with few states. In: Latifi, S. (ed.) *Information Technology: New Generations: 14th International Conference on Information Technology*, vol. 1, pp. 877–882. Springer International Publishing (2018)
9. Andro-Vasko, J., Bein, W., Ito, H.: Energy efficiency and renewable energy management with multi-state power-down systems. *Information* **10**(2), 44 (2019)
10. Augustine, J., Irani, S., Swamy, C.: Optimal power-down strategies. In: *IEEE Symposium on Foundations of Computer Science*, pp. 530–539. Cambridge University Press, (2004)
11. Augustine, J., Irani, S., Swamy, C.: Optimal power-down strategies. *SIAM J. Comput.* **37**(5), 1499–1516 (2008)
12. Bein, W., Madan, B.B., Bein, D., Nyknahad, D.: Algorithmic approaches for a dependable smart grid. In: Latifi, S. (ed.) *Information Technology: New Generations: 13th International Conference on Information Technology*, vol. 448, pp. 677–687. Springer International Publishing (2016)
13. Ben-Tal, A., Bertsimas, D., Brown, D.B.: A soft robust model for optimization under ambiguity. *Oper. Res.* **58**(4, Part 2 of 2), 1220–1234 (2010)
14. Borodin, A., El-Yaniv, R.: *Online Computation and Competitive Analysis*. Cambridge University Press (1998)
15. Chung, E., Benini, L., Bogliolo, A., Lu, Y., De Micheli, G.: Dynamic power management for nonstationary service requests. *IEEE Trans. Comput.* **51**(11), 1345–1360 (2002)
16. Budischak, C., Sewell, D.A., Thomson, H., Mach, L., Veron, D.E., Kempton, W.: Cost-minimized combinations of wind power, solar power and electrochemical storage, powering the grid up to 99.9% of the time. *J. Power Sources* **225**, 60–74 (2013)
17. de Mars, P., O’Sullivan, A., Keppo, I.: Estimating the impact of variable renewable energy on base-load cycling in the GB power system. *Energy* **195**, 9 (2020)
18. Eggers, S.J., Katz, R.H.: Evaluating the performance of four snooping cache coherency protocols. In: *Proc. 16th International Symp. on Computer Architecture (ISCA)*. IEEE, pp. 2–15 (1989)
19. Hall, R., Erdélyi, R., Hanna, E., Jones, J.M., Scaife, A.A.: Drivers of North Atlantic Polar front jet stream variability. *Int. J. Climatol.* **35**, 1697–1720 (2014)
20. Huang, Y., Pardalos, P.M., Zheng, Q.P.: *Electrical Power Unit Commitment*. Springer, Boston (2017)

21. Irani, S., Gupta, R., Shukla, S.: Competitive analysis of dynamic power management strategies for systems with multiple power savings states. In: DATE '02: Proceedings of the conference on Design, automation and test in Europe, p. 117, Washington, DC, USA, (2002). IEEE Computer Society
22. Irani, S., Pruhs, K.R.: Algorithmic problems in power management. In: ACM SIGACT News, vol. 36, pp. 63–76 (2005)
23. Karlin, A.R., Kenyon, C., Randall, D.: Dynamic tcp acknowledgement and other stories about  $e/(e-1)$ . In: Proc. 33rd STOC, pp. 502–509. ACM (2001)
24. Karlin, A., Manasse, M., McGeoch, L., Owicki, S.: Competitive randomized algorithms for nonuniform problems. *Algorithmica* **11**, 542–571 (1994)
25. Karlin, A., Manasse, M., Rudolph, L., Sleator, D.: Competitive snoopy caching. *Algorithmica* **3**, 79–119 (1988)
26. Maimo-Far, A., Tantet, A., Homar, V., Drobinski, P.: Predictable and unpredictable climate variability impacts on optimal renewable energy mixes: the example of Spain. *Energies* **13**(9), 25 (2020)
27. Marini, B.: Are simple cycles or combined cycles better for renewable power integration? *Power* **159**, 72–76 (2015)
28. Kumar, N., Besuner, P., Lefton, S., Agan, D., Hilleman, D.: Power Plant Cycling Costs. NREL (2012)
29. Meyerson, A.: Online algorithms for network design. In: SPAA 2004: Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, June 27–30, Barcelona, Spain, pp. 275–280 (2004)
30. Nedeveschi, S., Chandrashekar, J., Liu, J., Nordman, B., Ratnasamy, S., Taft, N.: Skilled in the art of being idle: Reducing energy waste in networked systems. In: Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI'09, pages 381–394, Berkeley, CA, USA, (2009). USENIX Association
31. Padhy, N.P.: Unit commitment—a bibliographical survey. *IEEE Trans. Power Syst.* **19**(2), 1196–1205 (2004)
32. Phillips, S., Westbrook, J.: Competitive analysis and beyond. In: Atallah M.J. (ed.) *Algorithms and Theory of Computation Handbook*. Chapter 10, CRC Press, ISBN 0849326494 (1999)
33. Reich, J., Goraczko, M., Kansal, A., Padhye, J., Padhye, J.: Sleepless in Seattle no longer. Technical report, Microsoft. 2010. <https://www.microsoft.com/en-us/research/publication/sleepless-in-seattle-no-longer/>. Accessed 7 June 2021
34. Ringkjøb, H.-K., Haugan, P.M., Solbrenke, I.M.: A review of modelling tools for energy and electricity systems with large shares of variable renewables. *Renew. Sustain. Energy Rev.* **96**, 440–459 (2018). (ISSN 1364-0321)
35. Sällberg, E., Lind, A., Velut, S., Åkesson, J., Yances, S.G., Link, K.: Start-up optimization of a combined cycle power plant. In: Proceedings of the 9th International Modelica Conference, Linköping Electronic Conference Proceedings, vol. 76, pp. 619–630 (2012)
36. Tahanan, M., van Ackooij, W., Frangioni, A., Lacalandra, F.: Large-scale unit commitment under uncertainty. *4OR-Q J Oper Res* **13**, 115–171 (2015)
37. Vincent, T., Calvin, S., Katherine, C., Geoffrey, K., Fields, F., Clement, Z., Bauer, D.: Implications of power plant idling and cycling on water use intensity. *Environ. Sci. Technol.* **53**, 4657–4666 (2019)
38. Van den Bergh, K.: Cycling of conventional power plants: technical limits and actual costs. *Energy Convers. Manag.* **97**, 70–77 (2015)
39. Wolff, R.W.: *Stochastic Modeling and the Theory of Queues*. Prentice Hall (1989)
40. Wu, D., Zheng, X., Xu, Y., Olsen, D., Xia, B., Singh, C., Xie, L.: An open-source model for simulation and corrective measure assessment of the 2021 Texas Power Outage. 05 April (2021), PREPRINT (Version 1) available at Research Square <https://doi.org/10.21203/rs.3.rs-384535/v1>
41. Yoshida, Y., Yoshida, T., Enomoto, Y., Osaki, N., Nagahama, Y., Tsuge, Y.: Start-up optimization of combined cycle power plants: a field test in a commercial power plant. *J. Eng. Gas Turbines Power.* **141**(3), 9 (2018)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

**James Andro-Vasko**<sup>1</sup> · **Wolfgang Bein**<sup>1</sup>  · **Hiro Ito**<sup>2</sup> · **Shoji Kasahara**<sup>3</sup> · **Jun Kawahara**<sup>3</sup>

<sup>1</sup> Department of Computer Science, Center for Information Technology and Algorithms, University of Nevada, Las Vegas, Las Vegas, NV, USA

<sup>2</sup> School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan

<sup>3</sup> Division of Information Science, Graduate School of Science and Technology, Nara Institute of Science and Technology, Nara, Japan