

Competitive Analysis of Dynamic Power Management Strategies for Systems with Multiple Power Saving States

Sandy Irani

Sandeep Shukla

Rajesh Gupta

Information and Computer Science

University of California, Irvine

Irvine, CA 92697-3425

E-mail: {irani, skshukla, rgupta}@ics.uci.edu

Abstract

We present strategies for “online” dynamic power management (DPM) based on the notion of the competitive ratio that allows us to compare the effectiveness of algorithms against an optimal strategy. This paper makes two contributions: it provides a theoretical basis for the analysis of DPM strategies for systems with multiple power down states; and provides a competitive algorithm based on probabilistically generated inputs that improves the competitive ratio over deterministic strategies. Experimental results show that our probability-based DPM strategy improves the efficiency of power management over the deterministic DPM strategy by 25%, bringing the strategy to within 23% of the optimal offline DPM.

1. Introduction

Dynamic Power Management (DPM) has gained considerable attention over the last few years as a way to save energy in devices that can be turned on and off by operating system control. Dynamic power management is an inherently *online* problem in that an algorithm governing power management must make decisions about the expenditure of resources before all the input to the system is available. For DPM strategies, the input is the length of an upcoming idle period and the decision to be made is whether to transition to a lower power dissipation state while the system is idle. Analytical solutions to online problems are best described in terms of a *competitive ratio* [9] that compares the cost of an online algorithm to the optimal offline solution which knows the input in advance.

Earlier work on competitive analysis of dynamic power management strategies presented bounds on the quality of DPM solutions [11, 5]. Competitive analysis has proven to be a powerful tool in providing a guarantee on the per-

formance of an algorithm for any input. We address here two chief limitations of this earlier work: real systems often have multiple idle states with transition energy costs that must be taken into account. This paper presents analytical bounds on the performance of strategies for systems with multiple idle states. Secondly, competitive analysis often gives overly pessimistic bounds for the behavior of algorithms. This is inherently the result of the fact that competitive analysis is a worst-case analysis. Competitive analysis still has great value in situations where it is impractical to obtain and process information for predicting future inputs. However, in many applications, there is structure in the input sequence that can be utilized to fine tune online strategies and improve their performance. Indeed, earlier work [1, 10] has relied on modeling the distribution governing interarrival times as an exponential distribution. In practice, such stochastic modeling seems to hold well for specific kinds of applications. However, these assumptions have led to complications in other settings due to such phenomena as the non-stationary nature of the arrival process, clustering, and the lack of independence between subsequent events. These problems have been addressed to some extent in [8, 3]. Our approach relies upon a probability distribution governing the arrival sequence of requests which is learned based on historical data. One of the strengths of this method is that we make no assumptions about the form of this distribution. Once the distribution is learned, we can automatically generate a probability-based DPM strategy that minimizes the expected power dissipation given that the input is generated according to that distribution. We compare the expected power dissipation of our online algorithm to that of the optimal offline algorithm to get a *probabilistic competitive ratio*. This method has been used in the context of two-state systems [5, 6]. We generalize this work for multi-state systems.

In the literature, one can find many strategies proposed and evaluated for DPM, such as predictive strategies [4, 12],

stochastic modeling based strategies [1, 10], session clustering and prediction strategies [8], on-line strategies [11], and adaptive learning based strategies [3]. In [7], one can find a quantitative comparison between various existing management strategies. Previous work on prediction based dynamic power management can be categorized into two groups: adaptive and non-adaptive. Non-adaptive strategies set a threshold on the idle time interval for transitioning from the active to the sleep state. For multiple state systems, there is a sequence of thresholds each of which indicates when to transition to the next lower power consumption state. In either case, non-adaptive strategies set these thresholds once and for all and do not alter them based on observed input patterns. Adaptive strategies, on the other hand, use the history of idle periods to guide the decisions of the algorithm for future idle periods. There have been a number of adaptive strategies proposed in the literature [4, 5, 1, 2]. In particular, Chung, Benini and De Micheli [3] address multiple idle state systems. In addition, they use a prediction scheme based on adaptive learning trees that proves to be robust under a variety of systems.

The work of Chung *et al.* (along with most of the adaptive algorithms in the literature) seeks to make a single prediction for what the upcoming idle period will be. The algorithm then behaves as if this prediction will hold and pays a price in the event that it is wrong. Since we use a probability distribution to predict the upcoming idle period length, we allow for some degree of uncertainty in our prediction for the future and optimize our algorithm in a way that takes the nature of this uncertainty into account. An interesting aspect of this work is that we present analytical as well as empirical bounds for the performance of our algorithms. This provides a baseline to compare different algorithms and even the effects of individual decisions made by an algorithm.

All of the previous work on competitive analysis for dynamic power management has concentrated on two-state systems [5, 6, 11]. We say that an algorithm is c -competitive if for any input, the cost of the online algorithm is bounded by c times the cost of the optimal offline algorithm for that input. The *competitive ratio* of an algorithm is the infimum over all c such that the algorithm is c -competitive. It has been proven that 2 is the best competitive ratio achievable by any deterministic online algorithm [9]. We extend this analysis to show a 2-competitive algorithm for the multi-state case. This result is tight in the sense that there is no constant $c < 2$ such that there is a deterministic c -competitive algorithm which works for all multiple power down state systems. However, it may be possible to have a competitive ratio less than c for a specific system, depending on the parameters of the system (e.g. number of states, power dissipation rates, start-up costs, etc.)

Probabilistic analysis for online DPM algorithms in two-

state systems has been given in [5, 6]. They assume, as we do here, that the distribution over the upcoming idle period is known and optimize the algorithm based on that distribution. They give a method for determining the best online algorithm given a distribution and show that for any distribution, the expected cost of this online algorithm is within a factor of $e/(e-1) \approx 1.58$ of the expected cost of the optimal offline algorithm. This result is tight in that there is a distribution for which the ratio is exactly $e/(e-1)$, although for some distributions the ratio may be less. We extend this work to the multi-state case, showing an algorithm which also achieves a ratio of $e/(e-1)$ for any system. Similar to the deterministic case, it is impossible to have a better bound for all systems, although it may be possible to have an algorithm which achieves a better bound for a particular system depending on the parameters of that system.

The idea of modeling idle period lengths by a probability distribution leads to two distinct questions:

1. If the upcoming idle period length will be generated according to a probability distribution known the algorithm, how can this information be used to optimize power consumption?
2. Given historical data for previous idle periods, how can we use this information to reliably construct a probability distribution describing future idles periods?

It should be noted that we only address the first question in this paper. The second question is a very important direction for future work which will complement the work presented here.

2. Online Algorithms for DPM

First we define some notation that will be useful in describing both the deterministic and probability-based algorithms. The number of states will be $k+1$: state k is the active state and state 0 is the completely powered down (or sleep) state. α_i is the power dissipation rate for state i . β_i is the total energy dissipated in moving from state i back to state k . We will assume that the states are ordered so that $\alpha_i \geq \alpha_{i-1}$ for all $1 \leq i \leq k$. Thus, we have that $\beta_k = 0$. We will also assume that $\alpha_0 = 0$, i.e. no energy is consumed in the deepest sleep state. If this is not the case, then any algorithm will dissipate power at a rate of at least α_0 regardless of its policy. Thus, we can subtract α_0 from the power dissipation rate of each state and isolate the power dissipation which is due to the choices made by the algorithm. Subtracting α_0 in this manner only serves to increase the competitive ratio.

Once an idle interval starts, the algorithm can choose to be in any of the $k+1$ states at any point in time. When a request for service arrives, the algorithm must immediately

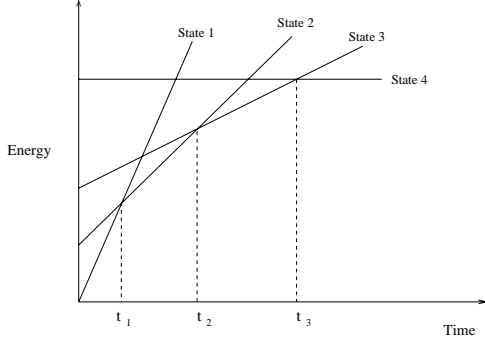


Figure 1. Energy consumption for each state for a four state system. Each state is represented by a line which indicates the energy used if an algorithm stays in that state as a function of the length of the idle period. For each state, the slope is the power dissipation rate and the y intercept is the energy required to power up from that state.

transition back to state k . The online algorithm does not know the length of the idle interval until it ends. It costs nothing for the algorithm to drop to a lower power dissipation state (i.e. a lower numbered state). This can be assumed because the power-down energy can be incorporated into the start-up energy without loss of generality. Note that, from the power dissipation point of view, there is no incentive for the algorithm to transition to a higher power dissipation state until the algorithm is forced to serve an incoming request in the active state.

2.1. The Deterministic Algorithm

To get the optimal cost, plot each line $c = \alpha_i t + \beta_i$. This is the cost of spending the entire interval in state i as a function of t , the length of the interval. Take the lower envelope of all of these lines. Call this function $LE(t)$. The optimal cost for an interval of length t is $LE(t) = \min_i \{\alpha_i t + \beta_i\}$. The online algorithm called the Lower Envelope Algorithm (LEA) will also follow the function LE . It will remain in the state which realizes the minimum in LE and will transition at the discontinuities of the curve. That is, LEA will remain in state j as long as $\alpha_j t + \beta_j = \min_i \{\alpha_i t + \beta_i\}$, for the current time t .

For $j = k$ to 1, let t_j be the solution to the equation $\alpha_j t + \beta_j = \alpha_{j-1} t + \beta_{j-1}$. t_j is the time that LEA will transition from state j to state $j-1$. We will assume here that we have thrown out all the lines which do not appear on the lower envelope at some point. This is equivalent to the assumption that $t_k < t_{k-1} < \dots < t_2 < t_1$. (See Figure 1).

Theorem 1 *The Lower Envelope algorithm is 2-competitive.*

The proof of Theorem 1, is provided in the appendix.

As emphasized earlier, this algorithm, does not take into account input patterns. Thus, the worst case scenario, obtained via Theorem 1, shows that the energy cost resulting from the on-line decisions can be no worse than 2 times the energy cost of the optimal offline strategy which knows the input sequence in advance. We show later that depending on request arrival patterns, this worst case bound may not really happen, and the empirical ratio of the online to offline costs may be much lower.

However, as shown in [10, 3, 1], input sequences are often interrelated, and hence modeling of the input pattern and exploiting that knowledge in the design of the algorithm can help bridge the gap between the performance of online strategy and that of the optimal offline strategy. In the next subsection, we discuss our probability-based algorithm and show that if the request interarrival time probability distribution is known before hand, the worst case competitive ratio can be improved by 21%, with respect to the deterministic case. Moreover, we show through experimental evaluation that this worst case bound is pathological. In fact, we can bring the energy cost of the online algorithm within 27% of the optimal offline one.

2.2. The Probability-based Algorithm

In this section, we assume that the length of the idle interval is generated by a fixed, known distribution whose density function is π . We first discuss systems with two states and then give our generalization to the multi-state case. let β be the start-up energy of the sleep state and α the power dissipation of the active state. Suppose that the online algorithm uses τ as the threshold at which time it will transition from the active state to the sleep state if the system is still idle. In this case, the expected energy cost for the algorithm for a single idle period will be

$$\int_0^\tau \pi(t)(\alpha t) dt + \int_\tau^\infty \pi(t)[\alpha \tau + \beta] dt.$$

The best online algorithm will select a value for τ which minimizes this expression. The offline algorithm which knows the actual length of an upcoming idle period will have an expected cost of

$$\int_0^{\beta/\alpha} \pi(t)(\alpha t) dt + \int_{\beta/\alpha}^\infty \pi(t)\beta dt.$$

It is known for the two state case, that the online algorithm can pick its threshold τ so that the ratio of its expected cost to the expect cost of the optimal algorithm is at most $e/(e-1)$ [5, 6]. That is, for any π , and any α and β ,

$$\frac{\min_\tau \left\{ \int_0^\tau \pi(t)(\alpha t) dt + \int_\tau^\infty \pi(t)[\alpha \tau + \beta] dt \right\}}{\int_0^{\beta/\alpha} \pi(t)(\alpha t) dt + \int_{\beta/\alpha}^\infty \pi(t)\beta dt} \leq \frac{e}{e-1}.$$

This is optimal in that for any α and β , there is a distribution π such that this ratio is at least $e/(e-1)$.

Let us now consider the multi-state case. As in the previous section, let t_j be the solution to the equation $\alpha_j t + \beta_j = \alpha_{j-1} t + \beta_{j-1}$. t_j is the time that LEA will transition from state j to state $j-1$. We will assume here that we have thrown out all the lines which do not appear on the lower envelope at some point. This is equivalent to the assumption that $t_k < t_{k-1} < \dots < t_2 < t_1$. For ease of notation, we will define t_{k+1} to be 0 and t_0 to be ∞ . The cost (expected energy consumption) of the optimal offline algorithm is:

$$\sum_{i=0}^k \int_{t_{i+1}}^{t_i} \pi(t) [\alpha_i t + \beta_i] dt.$$

Now to determine the online algorithm, we must determine k thresholds, where the threshold τ_i is the time at which the online algorithm will transition from state i to state $i-1$. In the spirit of the deterministic online algorithm for the multi-state case, we will let τ_i be the same as the threshold which would be chosen if i and $i-1$ were the only two states. We call this algorithm the Probability-based Lower Envelope Algorithm (PLEA). The proof of the following theorem appears in the appendix.

Theorem 2 *For any distribution, the expected cost of the Probability-based Lower Envelope Algorithm is within a factor of $e/(e-1)$ of the expected cost for the optimal offline algorithm.*

3. Experimental Results

To demonstrate the utility of our probability-based algorithm, we use a mobile harddrive from IBM [14]. This drive has four power down states, as shown in Figure 3. Here, the start-up energy refers to the energy cost in transitioning from a state to the active state. For application disk access data, we used trace data from auspex file server archive which is available at [13]. From this data, we collected the inter-arrival time for requests for disk access for 0.4 million disk accesses divided into multiple trace files, corresponding to different hours of the day.

To evaluate the deterministic algorithm, we ran it on fifteen different traces. For each trace, we calculated the total energy expenditure by the online algorithm and the total energy expenditure by the optimal offline algorithm and took the ratio between those two values. This data appears in the first column of numbers in Figure 3. Note that for all the traces, the ratio is significantly less than 2 (the theoretical upper bound). This demonstrates that the algorithm does much better in practice than the competitive ratio indicates.

To test the probability-based algorithm, we used each trace to build a distribution over idle period lengths. The probability distribution was constructed by first building a

State	Power Consumption in Watts	Start-up Energy in Joules	Transition Time to Active
Sleep	0	4.75	5 S
Stand-by	.2	1.575	1.5 S
Idle	.9	.56	40 mS
Active	2.4	0	0

Figure 2. Values for the power dissipation and start-up energy for the IBM mobile harddrive at [14] used in our experiments.

histogram in which all the idle period lengths were stored. The histogram was then used as the basis of a probability distribution as follows: each bin in the histogram represents a range of values for an idle period. Each bin was chosen in proportion to the weight of that bin in the histogram. An idle period length was then chosen by uniformly selecting a value from the range for that bin. We did not use uniform ranges for the bins since the data contained a few very long idle periods. Thus, we used a finer granularity for the bins representing shorter idle periods.

Once the probability distribution was constructed, we use this data to determine the thresholds for the Probability-based Lower Envelope Algorithm. Then we generated 10000 idle period lengths according to this distribution. For each idle period, we calculated the ratio between the total energy expenditure of PLEA and the total energy expenditure of the optimal offline algorithm. Thus, we assume for these experiments that we have perfect knowledge of the distribution generating the idle periods. Our goal is to examine how effectively we use this information in optimizing power management. The traces were useful as the basis for determining typical probability distributions. The second column of numbers in Figure 3 shows the ratio of the cost of the probability-based online algorithm to the offline algorithm on the data generated by the probability distributions based on each trace.

Figure 4 shows the average energy consumption for all four algorithms: deterministic online, deterministic offline, probability-cased online, probability-based offline. Thus, the deterministic column in Figure 3 is the ratio of the heights of the first two bars from Figure 4 for each trace. Similarly, the probability-based column in Figure 3 is the ratio of the values in the last two bars. The second bar for each trace in Figure 4 is the actual average offline cost for the idle periods in each trace. The fourth bar for each trace in Figure 4 is the average offline cost for idle periods generated according to a probability distribution based on each trace. Thus we expect the two values to be close, although not identical.

The results demonstrate that knowledge of the probabil-

ity distribution over idle periods can improve power consumption by 25% over the deterministic algorithm, bringing the power consumption down to 1.23 times the optimal offline algorithm on average. Earlier experiments for two-state models show that the deterministic algorithm compares favorably with respect to other adaptive algorithms in the literature [11]. Since the probability-based algorithm given here improves upon the deterministic algorithm, this comparison would be even more favorable for PLEA.

The worst case for the deterministic algorithm is a competitive ratio of 2. Thus, 2 is a theoretical upper bound for the the first column of numbers in Figure 3. In many cases, the results are actually much better than 2, showing that the theoretical bound is often a pessimistic estimate on the performance of the algorithm. In general, the deterministic online algorithm will do well for short idle periods in which case it does not drop to a lower energy consumption state. In these cases, the ratio of the cost of the online to the offline algorithms is close to one. The deterministic algorithm will do the most poorly for idle periods which are just longer than one of its threshold times. That is, it will do the most poorly in comparison to the optimal offline algorithm when a new request arrives just after it has transitioned to a lower power dissipation state. The very worst case which will force its ratio closest to 2 is when a request arrives just after it transitions to the sleep state. Thus, one would expect that on those traces where the deterministic competitive ratio is very close to two, the distribution of idle period lengths is tightly clustered just after the longest threshold for the algorithm.

The probability-based algorithm will, in general, perform better on those distributions that are highly clustered since those distributions give the most information about what idle periods lengths are more likely to occur. This is born out in our experimental results: those traces which result in ratios which are very close to two for the deterministic algorithm achieve ratios that are relatively close to one for the probability-based algorithm. Analysis of the two-state case tells us that the distributions that result in the worst ratio for the probability-based algorithm are exponentially distributed. They give the online algorithm very little information about the length of the upcoming idle period. The exponential distribution is the distribution that results in PLEA achieving a ratio of $e/(e-1) \approx 1.58$ for the two-state case. Notice that on the traces used in our experiments, the results are much better than the worst case $e/(e-1)$ since the distributions resulting from the traces give the online algorithm more information about the upcoming idle period length.

Trace	Deterministic Competitive Ratio	Probability-based Competitive Ratio Length	Trace
Trace 1	1.6556	1.3499	33626
Trace 2	1.4173	1.1595	67176
Trace 3	1.6277	1.2270	34006
Trace 4	1.8747	1.1540	7233
Trace 5	1.765	1.3032	1507
Trace 6	1.7328	1.3787	10304
Trace 7	1.9963	1.0124	558
Trace 8	1.7708	1.2252	7100
Trace 9	1.5256	1.2384	70280
Trace 10	1.6717	1.3792	43404
Trace 11	1.9688	1.0638	2167
Trace 12	1.5899	1.2891	24145
Trace 13	1.63277	1.1573	15839
Trace 14	1.7313	1.3313	17360
Trace 15	1.848	1.2479	12270
Average	1.7206	1.2345	

Figure 3. Ratios of the performance of online algorithms to the optimal offline algorithms in our experiments.

4 Conclusions

This paper presents two algorithms for dynamic power management for systems with multiple sleep states. Competitive analysis is used to guarantee a bound on the performance of the deterministic algorithm for any input sequence. Our empirical results show that it actually performs better than this bound on the harddisk data used in our study. The next algorithm assumes that the idle periods are generated according to a known probability distribution. In this case, we show, analytically and empirically, that this knowledge can be used to greatly enhance the performance of a DPM strategy.

The most important direction for future work is to develop methods for finding a probability distribution which will accurately predict future idle periods lengths based on past idle period lengths.

5 Acknowledgements

This work was supported by NSF grant CCR-0098335, SRC, and DARPA/ITO supported PADS project under the PAC/C programme. In addition, the first author is also partially supported by NSF grant CCR-0105498 and by ONR Award N00014-00-1-0617.

Comparison of Energy Dissipation between the Deterministic and Probability Based Strategies

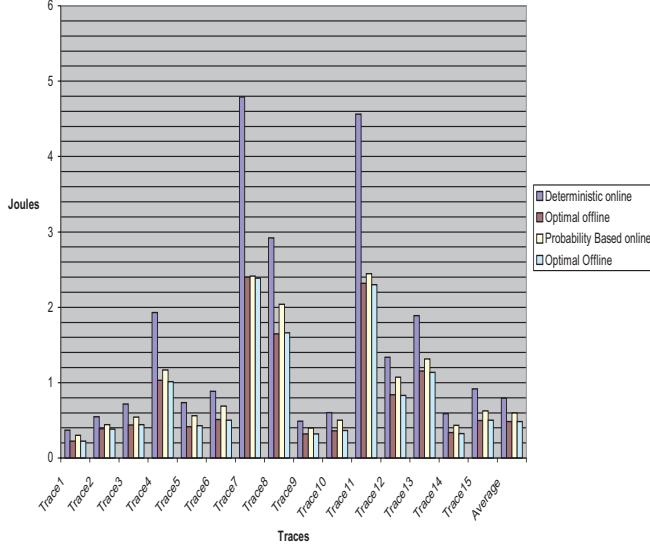


Figure 4. Average energy consumption in Joules per idle period for all four strategies. The averages at the end are a weighted average based on the trace length.

References

- [1] L. Benini, A. Bogliolo, G. Paleologo, and G. D. Micheli. Policy optimization for dynamic power management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(6):813–833, 1999.
- [2] E. Chung, L. Benini, A. Bogliolo, and G. DeMicheli. Dynamic power management for non-stationary service requests. In *DATE - Proceedings of the Design and Automation and Test in Europe Conference and Exhibition*, pages 77–81, 1999.
- [3] E.-Y. Chung, L. Benini, and G. D. Micheli. Dynamic power management using adaptive learning trees. In *Proceedings of ICCAD*, 1999.
- [4] C.-H. Hwang, C. Allen, and H. Wu. A predictive system shutdown method for energy saving of event-driven computation. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pages 28–32, 1996.
- [5] A. Karlin, M. Manasse, L. McGeoch, and S. Owicki. Randomized competitive algorithms for non-uniform problems. In *First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 301–309, 1990.
- [6] S. Keshav, C. Lund, S. Phillips, N. Reingold, and H. Saran. An empirical evaluation of virtual circuit holding time policies in ip-over-atm networks. *IEEE Journal on Selected Areas in Communications*, 13:1371–1382, 1995.

- [7] Y. Lu, E. Chung, t. Simunic, L. Benini, and G. DeMicheli. Quantitative comparison of power management algorithms. In *DATE - Proceedings of the Design and Automation and Test in Europe Conference and Exhibition*, 2000.
- [8] Y. Lu and G. DeMicheli. Adaptive hard disk power management on personal computers. In *Proceedings of the Great Lakes Symposium on VLSI*, 1999.
- [9] S. Phillips and J. Westbrook. *chapter 10 of Algorithms and Theory of Computation Handbook*, chapter On-line algorithms: Competitive analysis and beyond. CRC Press, Boca Raton, 1999.
- [10] Q. Qiu and M. Pedram. Dynamic Power Management Based on Continuous-Time Markov Decision Processes. In *Proceedings of the Design Automation Conference*, pages 555–561, June 1999.
- [11] D. Ramanathan, S. Irani, , and R. K. Gupta. Latency effects of system level power management algorithms. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, 2000.
- [12] M. B. Srivastava, A. P. Chandrakasan, and R. W. Broderson. Predictive Shutdown and Other Architectural Techniques for Energy Efficient Programmable Computation. *IEEE Trans. on VLSI Systems*, 4(1):42–54, Mar. 1996.
- [13] *Auspex File Traces from the NOW project*, available at <http://now.cs.berkeley.edu/Xfs/AuspexTraces/auspex.html>, 1993.
- [14] *Technical specifications of hard drive IBM Travelstar VP 2.5inch*, available at <http://www.storage.ibm.com/storage/oem/data/travvp.htm>, 1996.

A. Appendix

Proof of Theorem 1. First we establish that the worst case for the algorithm will always be just after a transition time. Consider the time $t_j + \gamma$, for some $k \geq j \geq 1$ and $0 \leq \gamma < t_{j-1} - t_j$. For any value of γ in the given range, the optimal cost will be

$$\alpha_{j-1}(t_j + \gamma) + \beta_{j-1}.$$

For any value of γ in the given range, the online cost will be:

$$\alpha_k t_k + \sum_{l=j}^k \alpha_{l-1}(t_{l-1} - t_l) + \alpha_{j-1}\gamma + \beta_{j-1}.$$

The ratio of these two will be maximized for $\gamma = 0$.

Now suppose that the interval ends just after t_j for some $k \geq j \geq 1$. Using the cost for the online and offline determined above, the ratio of the online cost to the offline cost will be

$$\begin{aligned} & \frac{\alpha_k t_k + \sum_{l=j}^k \alpha_{l-1}(t_{l-1} - t_l) + \beta_{j-1}}{\alpha_{j-1} t_j + \beta_{j-1}} \\ &= 1 + \frac{\alpha_k t_k + \sum_{l=j}^k \alpha_{l-1}(t_{l-1} - t_l) - \alpha_{j-1} t_j}{\alpha_{j-1} t_j + \beta_{j-1}} \end{aligned}$$

Thus, it is sufficient to prove that

$$\alpha_k t_k + \sum_{l=j}^k \alpha_{l-1}(t_{l-1} - t_l) - \alpha_{j-1} t_j \leq \alpha_{j-1} t_j + \beta_{j-1}.$$

Rearranging, we must prove that

$$\sum_{l=j}^k (\alpha_l - \alpha_{l-1})t_l - \alpha_j t_{j-1} \leq \alpha_{j-1} t_j + \beta_{j-1}. \quad (1)$$

Each t_l was chosen so that

$$(\alpha_l - \alpha_{l-1})t_l = \beta_{l-1} - \beta_l,$$

so we can substitute these values into Inequality 1 to get that

$$\begin{aligned} & (\beta_{k-1} - \beta_k) + \dots + (\beta_{j-1} - \beta_j) - \alpha_j t_{j-1} \\ & \leq \alpha_{j-1} t_j + \beta_{j-1}. \end{aligned}$$

Collapsing the telescoping sum, we get that

$$\beta_{j-1} - \beta_k - \alpha_j t_{j-1} \leq \alpha_{j-1} t_j + \beta_{j-1}.$$

Since all the α 's, β 's and t 's are non-negative, the equation holds. \blacksquare

Proof of Theorem 2. Consider a system in which there are only two states: i and $i-1$. Both online and offline must pay at least $\alpha_{i-1}t$ for an interval of length t . In addition, each must pay at least β_i for the start-up cost. These costs which are incurred by both algorithms, regardless of their choices will only serve to decrease the competitive ratio. In determining, τ_i , we will disregard these additional costs. Consider the system where the power consumption rate in the ON state is $\alpha_i - \alpha_{i-1}$ and is 0 in the OFF state. The energy required to transition from the ON to the OFF state is $\beta_{i-1} - \beta_i$. We will choose τ_i to be the transition time for the optimal online policy in this system. Thus, we will choose τ_i to be

$$\begin{aligned} & \arg \min_{\tau} \int_0^{\tau} \pi(t) t (\alpha_i - \alpha_{i-1}) dt \\ & + \int_{\tau}^{\infty} \pi(t) [\tau (\alpha_i - \alpha_{i-1}) + (\beta_{i-1} - \beta_i)] dt. \end{aligned}$$

The online cost for this new system is the above expression evaluated at $\tau = \tau_i$:

$$\begin{aligned} ON_i &= \int_0^{\tau_i} \pi(t) t (\alpha_i - \alpha_{i-1}) dt \\ &+ \int_{\tau_i}^{\infty} \pi(t) [\tau_i (\alpha_i - \alpha_{i-1}) + (\beta_{i-1} - \beta_i)] dt. \end{aligned}$$

Let t_i be defined to be $(\beta_{i-1} - \beta_i) / (\alpha_i - \alpha_{i-1})$. Note that this is the same definition in the previous proof: the point where the lines $\alpha_i t + \beta_i$ and $\alpha_{i-1} t + \beta_{i-1}$ meet. The offline cost for the new system is

$$\begin{aligned} OFF_i &= \int_0^{t_i} \pi(t) t (\alpha_i - \alpha_{i-1}) dt \\ &+ \int_{t_i}^{\infty} \pi(t) (\beta_{i-1} - \beta_i) dt. \end{aligned}$$

We are guaranteed that the ratio of the expected online to offline costs is at most $e/(e-1)$ [5, 6].

Since the ratio of ON_i to OFF_i is at most $e/(e-1)$ for each i , we know that

$$\frac{\sum_{i=1}^k ON_i}{\sum_{i=1}^k OFF_i} \leq \frac{e}{e-1}.$$

We will now prove that $\sum_{i=1}^k ON_i$ is exactly the expected cost for PLEA on the multi-level system. We will also prove that $\sum_{i=1}^k OFF_i$ is exactly the expected cost of the optimal algorithm for the multi-level system.

We will rephrase ON_i by separating the integral into the intervals from τ_{j+1} to τ_j . To simplify notation, τ_{k+1} will denote 0 and τ_0 will denote ∞ .

$$\begin{aligned} ON_i &= \sum_{j=i}^k \int_{\tau_{j+1}}^{\tau_j} \pi(t) [t (\alpha_i - \alpha_{i-1})] dt \\ &+ \sum_{j=0}^{i-1} \int_{\tau_{j+1}}^{\tau_j} \pi(t) [\tau_j (\alpha_i - \alpha_{i-1}) + (\beta_{i-1} - \beta_i)] dt. \end{aligned}$$

In the sum over all ON_i , we will group together all the contributions from each ON_i over the interval $[\tau_{j+1}, \tau_j]$ for $k \leq j \leq 1$. Note that this is the interval that the algorithm will spend in state j . This value will be:

$$\begin{aligned} & \sum_{i=j+1}^k \int_{\tau_{j+1}}^{\tau_j} \pi(t) [\tau_i (\alpha_i - \alpha_{i-1}) + (\beta_{i-1} - \beta_i)] dt \\ &+ \sum_{i=1}^j \int_{\tau_{j+1}}^{\tau_j} \pi(t) [t (\alpha_i - \alpha_{i-1})] dt. \end{aligned} \quad (2)$$

Thus, we have that

$$\sum_{i=1}^k ON_i = \sum_{j=0}^k f(j),$$

where

$$\begin{aligned} f(j) &= \sum_{i=j+1}^k \int_{\tau_{j+1}}^{\tau_j} \pi(t) [\tau_i (\alpha_i - \alpha_{i-1}) + (\beta_{i-1} - \beta_i)] dt \\ &+ \sum_{i=1}^j \int_{\tau_{j+1}}^{\tau_j} \pi(t) [t (\alpha_i - \alpha_{i-1})] dt \end{aligned}$$

Putting the summations inside the integrals and collapsing the telescoping sums, the expression in (2) becomes:

$$\int_{\tau_{j+1}}^{\tau_j} \pi(t) cost(t) dt,$$

where

$$\begin{aligned} cost(t) &= (\beta_j - \beta_k) + \tau_k \alpha_k \\ &+ \sum_{l=j+2}^k (\tau_{l-1} - \tau_l) \alpha_{l-1} + (t - \tau_{j+1}) \alpha_j. \end{aligned}$$

Note that

$$(\beta_j - \beta_k) + \tau_k \alpha_k + \sum_{l=j+2}^k (\tau_{l-1} - \tau_l) \alpha_{l-1} + (t - \tau_{j+1}) \alpha_j$$

is exactly the energy expended by PLEA if the idle period t is in the range $[\tau_{j+1}, \tau_j]$. Thus, the expected cost for PLEA is:

$$\sum_{j=0}^k \int_{\tau_{j+1}}^{\tau_j} \pi(t) cost(t) dt = \sum_{i=1}^k ON_i.$$

The proof that the expected offline cost is equal to $\sum_{i=1}^k OFF_i$ is the same as the proof for the online cost, except that the integrals are separated into intervals according to the t_i 's instead of the τ_i 's.