Nearly Optimal Strategies for Special Cases of On-line Capital Investment

Peter Damaschke

Computing Sciences, Chalmers University

41296 Göteborg, Sweden

ptr@cs.chalmers.se

Abstract

Suppose that some job must be done for a period of unspecified duration. The market offers a selection of devices that can do this job, each characterized by purchase and running costs. Which of them should we buy at what times, in order to minimize the total costs? As usual in competitive analysis, the cost of an on-line solution is compared to the optimum costs paid by a clearvoyant buyer. This problem which generalizes the basic rent-to-buy problem has been introduced by Y. Azar et al. In the so-called convex case where lower running costs always imply higher prices, a strategy with competitive ratio $4+2\sqrt{2} \approx 6.83$ has been proposed. Here we consider two natural subcases of the convex case in a continuous-time model where new devices can be bought at any time. For the static case where all devices are available at the beginning, we give a simple 4-competitive deterministic algorithm, and we show that 3.618 is a lower bound. (This is also the first non-trivial lower bound for the convex case, both for discrete and continuous time.) Furthermore we give a 2.88-competitive randomized algorithm. In the case that all devices have equal prices but are not all available at the beginning, we show that a very simple algorithm is 2-competitive, and we derive a 1.618 lower bound.

Keywors: on-line problems, competitive ratio, rent-to-buy

1 Problem Statement and Results

The *rent-to-buy* problem, also known as ski rental, leasing, spin-block problem etc., is a very basic on-line problem with only one positive real number x as its input. Assume that some resource (of whatever nature) is required for an unknown duration x. The on-line player has the option to rent it at cost r per time unit or to buy it at cost b. He may first rent the resource for a while and buy it later at any time he wants. The goal is to minimize the total cost. The optimal strategy of an off-line player (who knows x in advance) is to rent the resource if $x < b/r$, and to buy it at time 0 otherwise. It is an easy exercise to prove that the best deterministic on-line strategy has competitive ratio 2. This strategy is: Rent until time b/r, and then buy, in case that x is larger. (Note that the on-line player knows at time t that $x > t$.) As shown in [7], the best randomized strategy achieves excpected competitive ratio $\frac{e}{e-1} \approx 1.58$ against an oblivious adversary. (For basic concepts not explained here we refer to [2].)

One may consider r as any kind of running costs per time unit, rather than a rental fee. This gives reason to an equivalent formulation of the problem: We need some device for an unknown time x. There are two models available, one with $b_0 = 0$ (i.e. for free) but $r_0 > 0$, and a second one with $b_1 > 0$ but $r_1 = 0$ (i.e. without running costs). The player is obliged to buy some device at time 0, but he may buy any other device at an arbitrary later moment. That is, buying a device terminates accumulation of running costs of the previously used device.

Now we may consider a generalized version with $n + 1$ devices D_i, each characterized by non-negative numbers a_i, b_i, r_i, $i = 0, \ldots n$. Here a_i denotes the time when device D_i appears, that means, D_i is not available before time $a_i \geq 0$.

This problem, named *on-line capital investment*, has been introduced in [1]. It was shown that no algorithm for the general case can have constant competitive ratio, however, an algorithm with competitive ratio bounded by logarithmic functions of some natural instance parameters has been given, and these bounds are fairly tight, up to slowly growing factors.

On another front, competitive strategies exist for natural special cases of capital investment, one of which has been considered in [1]: Suppose that $r_i < r_j$ implies $b_i \geq b_j$, that is, a more "modern" device being cheaper at work is always more expensive in acquisition. An algorithm with competitive ratio $4 + 2\sqrt{2} \approx 6.83$ has been given for this so-called *convex case*.

It must be noticed that the problem was considered in [1] in the discrete-time model. In the present paper we adopt a continuous-time model where it is allowed to buy a device at any moment, unless otherwise stated. We achieve competitive ratios being not far from optimal, for two subcases of the convex case, as introduced below.

If all devices are available at time 0, and therefore only x is unknown to the on-line player, we speak of the *static case*. It is clear that we may without loss of generality assume $r_0 > \ldots > r_n$ and $b_0 < \ldots < b_n$, after removal of "redundant" devices. Hence the static case is included in the convex case. Note that the static case is also applicable to long phases in general instances where no new device is released (intervals between consecutive a_i), therefore this restriction is not too special.

Later we consider instances with arbitrary release times a_i, but equal prices b_i. Obviously, this is another subcase of the convex case.

We have the following results: In Section 2 we give a simple 4-competitive deterministic algorithm for the static case called DOUBLE. (It is similar to the 6.83-competitive convex case strategy of [1].) Despite its simplicity it is not far from optimal, as we will derive a 3.618 lower bound. The lower bound proof is the main technical contribution. Furthermore we propose a 2.88-competitive randomized version of DOUBLE. In Section 3 we show that a very simple algorithm has competitive ratio 2 in the case that all devices have the same price, but arbitrary times of appearance. For this case we have a lower bound of 1.618.

Other, different generalizations of rent-to-buy have been studied in a few papers. Rent-to-buy with interest rates is studied in [6]. In [8], an algorithm for a sequence of rent-to-buy decisions has been given, under the assumption that durations x are sampled from a fixed but unknown probability distribution. The closely related acknowledgement delay problem which arises in network communication protocols has been studied in [5]. In [3] we generalized rent-to-buy strategies to situations where several identical pieces of a resource are needed in overlapping time intervals. Some problems considered in mathematical finance (cf. [2] for a survey) are a little similar, but they have other cost assumptions and objectives. In [4] we studied another on-line problem of similar flavour as rent-to-buy.

2 The Static Case

We give a more formal definition of our problem in the static case. An instance consists of pairs (b_i, r_i), $i = 0, \ldots n$. A strategy specifies, for every i, a point t_i on the time axis which is either a non-negative real number or ∞. All finite t_i must be distinct, and one of the t_i has to be 0. Let $0 = u_1 < u_2 < u_3 < \ldots$ be the sorted sequence of these times t_i, and denote by $i(j)$ that index which satisfies $t_i = u_j$. Then the cost of the strategy until time $x \geq 0$ is defined as

$$\sum_{u_j \leq x} b_{i(j)} + r_{i(j)}(\min\{u_{j+1}, x\} - u_j).$$

Evidently, t_i is the time to buy device i, where $t_i = \infty$ means that device i will never be bought, and the cost is the accumulated purchase and running costs if, at any time, the device bought last is applied. One *must* have some device right from the beginning, therefore some must be bought already at time 0. It is required that some device must run at any moment in $[0, x]$. Now assume that an on-line player does not know the length x of this period, whereas an off-line player does and chooses an optimal strategy. Then the on-line player's goal is to minimize the competitive ratio, i.e. the ratio of his own costs and the off-line player's costs until time x. It is equivalent to say that we aim at a possibly small competitive ratio for any x (that is, to minimize the maximum over all $x \geq 0$).

An optimal off-line algorithm for fixed x is evident, due to the following observation:

Lemma 1 *Without loss of generality, an optimal off-line player OPT buys a device either immediately (when released) or never.*

Proof. Assume that OPT buys, at time t, a device D with costs (b, r), and that the currently used device has running cost r'. Clearly $r < r'$, otherwise there is no reason to buy D. But then it would be advantageous to buy D earlier: The price b is the same, and $r' - r$ running cost per time unit could be saved. This contradicts optimality. □

In particular, since in the static case all devices are present at time 0, the off-line player buys only one device, namely that with minimum $b_i + r_i x$. Hence the optimal solutions for all x are given by a piecewise linear monotone function being convex from above. In the following, this function is denoted f. Note that f and f^{-1} are very easy to compute.

No deterministic on-line algorithm can achieve a better competitive ratio than 2, since rent-to-buy is a special case. Next we give a 4-competitive on-line algorithm for the static case. As mentioned in the introduction, we assume $r_0 > \ldots > r_n$ and $b_0 < \ldots < b_n$. Clearly, the on-line player will successively buy some of the devices D_i, with i increasing in time. Note that f is bounded if and only if $r_n = 0$. We study the following on-line strategy:

Algorithm DOUBLE

At time 0, buy D_0. Then buy a new device when a certain condition specified below is met, and wait for a period specified below, without checking the condition. Then buy the next time the condition is met, wait again, etc. Now we give these specifications:

Let $y(t)$ be the total cost incurred until time t. The condition is that $y(t)$ reaches $2f(t)$. When this happens, find $u > t$ such that $f(u) = y(t)$. Find i such that the optimal off-line algorithm would buy D_i if $x = u$. That means $f(u) = b_i + r_i u$. (In the special case that the graph of f has a bend at u, choose e.g. the largest i satisfying this equation.) Buy this D_i at time t. Wait until time u, not buying anything.

If no u exists in the above situation (since f is bounded) then buy D_n. If $y(t)$ reaches b_n when still $y(t) < 2f(t)$, then buy D_n, too.

Theorem 2 *DOUBLE satisfies $y(t) < 4f(t)$ at any time t, hence it is 4-competitive.*

Proof. First let f be unbounded. Remember that both players have to buy something at time 0. DOUBLE buys D_0. As long as time t has not passed the first bend of f, we can assume that OPT also took D_0, since this is the optimal choice. Thus we have $y(t) = f(t)$ until the first bend of f. Function f is continuous, and as long as DOUBLE does not buy anything, y is also a continuous function, hence also y/f is continuous in any time interval between buy decisions. Therefore $y(t) \leq 2f(t)$ remains true until equality is reached.

If DOUBLE buys some device D_i at time t, let $y(t+) := y(t) + b_i$ denote the value of y immediately after the purchase. Now consider the first t with $y(t) = 2f(t)$, and let i be the index specified by the algorithm. Note that $b_i = y(t) - r_i u$. It follows $y(t+) < 4f(t)$. Furthermore observe that

$$y(u) = y(t+) + r_i(u - t) = 2y(t) - r_i t < 2f(u),$$

since DOUBLE buys no device in time interval $[t, u]$. In this time interval, y is linear and f is convex from above. Since $y < 4f$ holds at both endpoints, this inequality holds in the entire interval $[t, u]$. Moreover, since $y(u) \leq 2f(u)$, we can repeat the same argument for the next $t > u$, and so on.

If f is bounded then the above argument also works as long as $y(t) \leq b_n$. Otherwise DOUBLE buys D_n, thus

$$y(t+) = 2f(t) + b_n < 2f(t) + y(t) = 4f(t),$$

and y remains constant henceforth. This proves 4-competitiveness.

(We have added the last statement in the algorithm for sentimental reasons only: It ensures that DOUBLE yields the optimal competitive ratio 2 for rent-to-buy, but it is not relevant to the worst-case behaviour.) $\Box$

Algorithm DOUBLE as described above relies on the continuous-time assumption. It remains open whether the result carries over to the discrete-time setting. The on-line player can still ensure $y(u) \leq 2f(u)$ as above, but in general, he cannot take the exact moment t when $y(t) = 2f(t)$, hence the ratio may slightly exceed 4 in the beginning of every phase.

Next we raise the lower bound from the trivial ratio 2 to 3.618:

Theorem 3 *No deterministic algorithm for the static case (and hence for the convex case) of capital investment can achieve a better competitive ratio than* $c = \frac{5+\sqrt{5}}{2}$.

Proof. Consider a fixed c, and assume that there is an algorithm with competitive ratio less than $c - \delta$. Our final goal is to derive a contradiction, for c mentioned above and arbitrarily small $\delta > 0$, by presenting an instance that fools the on-line player. Then the theorem follows.

Our adversary constructs an instance with $b_0 = 0$, $r_0 = 1$, and $1 = b_1 < b_2 < \ldots < b_n$. We will specify our particular values b_i later on. Moreover suppose that a sequence of error terms ϵ_i has been fixed. They play a special role in our proof, for the moment just think of small positive numbers. Once the b_i and ϵ_i are decided, we can choose slopes $r_i > 0$ and times x_i, z_i such that the following conditions are fulfilled:
(1) A segment of every line $L_i = \{(x,y) : y = b_i + r_i x\}$ is part of graph F of the optimal cost function f, and x_i is the smallest x in $L_i \cap F$. That means, the off-line player would buy D_i if $x \in [x_i, x_{i+1})$.
(2) For every i, $z_i \in [x_i, x_{i+1}]$ is a time such that for every $j < i$, $b_j + r_j x$ will exceed $cf(x)$ before $x = z_i$, but it holds $f(x) < b_i + \epsilon_i$, for all $x \in [x_i, z_i]$.

To see that the claimed x_i, r_i, z_i exist, we argue as follows: We fix these values step by step, for increasing i. For every $i > 0$, we can obviously make

r_i small enough compared to the previous slopes such that condition (2) is satisfied for some $z_i > x_i$. Fix such a time z_i. Decreasing r_i further cannot violate (2), with the z_i just decided. Hence we can finally make r_i small enough to ensure that D_i still remains the optimal choice sometime after z_i (because b_{i+1} exceeds b_i plus the accumulated running costs of D_i). But then we can get some $x_{i+1} > z_i$, to satisfy (1).

The intuition behind this construction is to make the running cost of any new device D_i negligible, compared to all previous ones, such that the total cost becomes too high after a while if the on-line player hesitates to buy D_i, whereas it does not even noticeably increase up to that moment if D_i has been bought. In the following we only use properties (1) and (2), but no explicit values of r_i etc., therefore it is needless to give formulas for them.

Next we construct our sequence b_i explicitly, thereby adjusting our error terms. It is important to notice that our ϵ_i will only depend on c, δ, and previous b_j, but not on the *values* of x_i, r_i, z_i, hence there is no circulus vitiosus.

Remember that $b_1 = 1$.

Let $b_2 := c - 1$. Assume that the on-line player never buys D_1. Then we have $f < b_1 + \epsilon_1 = 1 + \epsilon_1$ after x_1 for a while, whereas the on-line cost increases up to c, due to (2). Therefore the on-line player must buy D_2 (or a more expensive device) before z_1, otherwise the competitive ratio would exceed $c - \delta$ for small enough ϵ_1. On the other hand, he cannot buy D_2 before x_1, as the competitive ratio would exceed c. Hence, after buying D_2, the on-line cost is at least c whereas we still have $f < 1 + \epsilon_1$. This yields a contradiction to competitive ratio $c - \delta$. This shows that D_1 must be bought. Once we know this, the best the on-line player can do is to buy D_1 at his earliest convenience, otherwise he pays unnecessarily for running the more expensive D_0. On the other hand, he cannot buy D_1 before f reaches $\frac{1}{c-1}$, since otherwise the competitive ratio would exceed c. Thus the on-line cost at time x_1 will be at least $\frac{c}{c-1}$.

In the following we extend this argument inductively to later times x_i, by suitable choice of the b_i and ϵ_i. However some more calculations are necessary.

Consider index i, and suppose that b_j is already fixed for all $j \leq i+1$. Let $p := b_{i+1}/b_i$, and let q be a coefficient such that the on-line cost at the time when D_i became most favourable for the off-line player (x_i) was at least qb_i. As for the induction base $i = 1$ we have shown above that we can choose $p = c - 1$ and $q = \frac{c}{c-1}$.

Now we fix $P := (c-1)p - (q-1)$ and $b_{i+2} := Pb_i$. We claim that the on-line player must buy D_{i+1} sometime. If the on-line player never buys D_{i+1} then, at time z_{i+1}, we have $f < b_{i+1} + \epsilon_{i+1}$ whereas the on-line cost has increased to at least cb_{i+1}. Therefore the on-line player must buy D_{i+2} (or a more expensive device) before that moment, otherwise the competitive ratio would exceed $c-\delta$ for small enough ϵ_{i+1}. On the other hand, he cannot have bought D_{i+2} before z_{i+1}, as the competitive ratio would exceed $c - \delta$: If the player buys D_{i+2} during $[x_{i+1}, z_{i+1}]$ then the on-line cost is at least $(q + p - 1 + P)b_i$ whereas we still have $f < b_{i+1} + \epsilon_{i+1} = pb_i + \epsilon_{i+1}$. Note that the ratio is above $c - \delta$ for small enough ϵ_{i+1}, and an earlier purchase would yield an even larger ratio, since on-line and off-line cost increase with the same slope r_i. This proves the claim.

The best the on-line player can do is to buy D_{i+1} at his earliest convenience, otherwise he pays too much for running D_i or an earlier device. But he cannot buy D_{i+1} before f reaches $(1 + y)b_i$, with $y := \frac{q+p-c}{c-1}$, otherwise the competitive ratio would exceed c again: To see this, verify that $\frac{q+y+p}{1+y} = c$.

If D_{i+1} is bought at the above mentioned moment then, obviously, the on-line cost at time x_{i+1} will be at least $(q+y+p)b_i$ which equals $\frac{(q+p-1)c}{c-1}b_i$.

We want to establish new coefficients p', q' for index $i+1$, having the same meaning as p, q have for index i. Define $Q := \frac{(q+p-1)c}{c-1}$. Then the on-line cost at time x_{i+1} is at least Qb_i. Furthermore remember that $b_{i+2} = Pb_i$.

Since $b_{i+1} = pb_i$, we just have $p' = P/p$ and $q' = Q/p$. This yields:

$$p' = c - \frac{p + q - 1}{p}$$

and

$$q' = \frac{c}{c-1} \cdot \frac{p + q - 1}{p}.$$

Note that $p' + q'\frac{c-1}{c} = c$. Since this relation holds for every i, we also obtain $p + q\frac{c-1}{c} = c$. We substitute p with $c - q(c-1)/c$ in the q' expression, and after some straightforward manipulation we get

$$q' = \frac{c^3 - c^2 + cq}{c^3 - c^2 - (c-1)^2 q}.$$

Define a sequence q_i by $q_1 = \frac{c}{c-1}$ and $q_{i+1} = (q_i)'$. Using this notion, we can summarize our hitherto discussion as follows: If $q_i > c$ for some i then a $(c - \delta)$-competitive algorithm cannot exist. Now the proof is completed by the following observations: For $c = \frac{5+\sqrt{5}}{2}$, the sequence q_i goes to the fixpoint $\sqrt{5}$. But if $c = \frac{5+\sqrt{5}}{2} - \zeta$ for any $\zeta > 0$, then the q_i sequence takes this hurdle and then grows to c and beyond. $\square$

As a consequence, the simple algorithm DOUBLE has competitive ratio less than 10/9 away from optimum. It seems possible that the true lower bound is 4, i.e. that DOUBLE is optimal. We leave this as an open problem. We remark that our lower bound proof constructs instances where $c = 3.6\ldots$ is reached only after an astronomic time x (compared to x_1). Thus the competitive ratio for bounded but unknown x/x_1 also deserves further study, since in real-world situations we are always interested in foreseeable periods of time.

Theorem 3 provides a lower bound also in the discrete-time model: One can start with the above construction and use the freedom in the choice of slopes to fine-tune them such that all x_i become integer.

The next result says that randomization beats the deterministic lower bound considerably. The competitive ratio is understood as expected value against an oblivious adversary; see [2] for fundamental notions.

Theorem 4 *There exists a 2.88-competitive randomized algorithm for the static case of capital investment.*

Proof. Basically we apply the deterministic algorithm DOUBLE, but to make analysis easier, we slightly aggravate the on-line costs: Whenever we get $y = 2f$, we pretend that y is doubled immediately and remains constant until f reaches this double value, and so on. (A moment of thinking reveals that this y is, in fact, not smaller than the cost incurred by DOUBLE.)

Moreover, prior to that, we choose some u between $f(x_1)/4$ and $f(x_1)/2$ (where x_1 is the first bend of f) and let y jump from u to $4u$ when $f = y = u$. This makes the evolution of costs particularly simple: DOUBLE and OPT alternatingly double their costs (instantly and over some period of time, respectively). Hence their ratio always ranges from 2 to 4. The only randomized part of our algorithm is the initial choice of u, which we specify now.

In the following, z denotes the final optimum cost (at time x). Without loss of generality let $2 \leq z \leq 4$, otherwise we may divide all costs by the suitable power of 2. Let k be the largest integer such that $v := 2^k u < 2$. Instead of choosing $u \in [f(x_1)/4, f(x_1)/2]$ at random, we may equivalently choose $v \in [1, 2]$ at random.

The crucial observation is: Since the ratio of on-line cost and optimum cost is always in $[2, 4]$, the final ratio is $8v/z$ if $2v < z$, and $4v/z$ if $2v \geq z$. Thus, if we sample v according to some density function h, the expected competitive ratio is

$$\frac{8}{z}\int_1^{z/2} vh(v)dv + \frac{4}{z}\int_{z/2}^{2} vh(v)dv = \frac{4}{z}\left(\int_1^{z/2} vh(v)dv + E[h]\right).$$

Choosing $h(v) := \frac{1}{v\ln 2}$ we obtain $\frac{4}{z\ln 2}(\frac{z}{2} - 1 + 2 - 1) = \frac{2}{\ln 2}$. This is the claimed bound. $\Box$

Note that this algorithm is barely random in the sense that it makes a random choice in the beginning, whereas the rest is deterministic. It remains open whether the result of Theorem 4 is optimal.

3 The Dynamic Case With Equal Prices

Recall that device D_i ($i = 0, \dots n$) is characterized by the triple (a_i, b_i, r_i), where the a_i, b_i, r_i are arbitrary non-negative numbers. The b_i and r_i have the same meaning as before, device D_i becomes available only at time a_i, and the on-line player learns about D_i only at that time. Some a_i must be 0. Instead of giving a total duration x as part of the input, we may equivalently add a dummy device with $a = x$ and $b = r = 0$.

Note that we assume that no offer disappears, i.e. device D_i can be bought at price b_i at any time $t \geq a_i$. Otherwise, i.e. if one can buy each device only during some time interval, no competitive algorithm exists, since the online search problem (see Section 14.1 in [2]) can be trivially reduced to this version. On the other hand, the present model captures reductions in prices: Instead of diminishing r_i we may introduce a new device with smaller price, such that the old one becomes redundant.

As shown in [1], the competitive ratio can be bounded in terms of some input parameters. Here we address the competitive ratio in the special case that all prices b_i are equal, without loss of generality $b_i = 1$.

The following result can be proved by reduction to the acknowledgement delay problem [5] which has an optimal deterministic competitive ratio 2. However a direct argument is even simpler:

Theorem 5 *There exists a 2-competitive algorithm for on-line capital investment if all prices b_i are equal.*

Proof. Without loss of generality let be $0 = a_0 < \dots < a_n$ and $r_0 > r_1 > \dots > r_n$. Obviously, other devices are redundant and may be removed from input.

Consider the following on-line algorithm: Buy the most recent device when the accumulated running costs since the last purchase reach 1, unless you are already using the most recent device.

Consider any phase, that is, a time interval $[x, y)$ between two purchases done by the on-line player. He pays 1 for the device and 1 unit of running costs. If the off-line player buys some device during the phase, he pays at least 1. If the off-line player does not buy anything during the phase, he has to pay only his running costs R. But since the on-line player bought the most recent device at time x, his running costs in the phase are not larger than R. This implies $R \geq 1$. In either case, the ratio is at most 2. $\square$

One may conjecture that Theorem 5 can be improved, exploiting the fact that both players always have to pay at least the running cost of the most recent device. (No analogous term occurs in the rent-to-buy or the acknowledgement delay problem where ratio 2 is optimal.) Actually there remains a gap between 2 and the following lower bound of 1.618:

Theorem 6 *No deterministic algorithm for on-line capital investment with equal prices can guarantee some competitive ratio better than* $c = \frac{1+\sqrt{5}}{2}$.

Proof. The basic idea is similar to Theorem 3, but the details are easier. Given an on-line strategy and any $\delta > 0$, we show how an adversary can construct an instance where that strategy has a competitive ratio at least $c - \delta$.

We consider instances where $0 = a_0 < a_1 < a_2 < \dots a_n$, with $r_0 = 1$, $r_n = 0$, where r_i is a rapidly decreasing sequence, as we will discuss below. Let y_i and z_i denote the total costs incurred until a_i by the on-line strategy and by the optimal off-line strategy, respectively. We get $y_0 = z_0 = 1$, since both players must buy D_0 at time 0. The adversary chooses some small a_1.

Suppose that all release times up to a_i have been fixed. Let d be the time when the on-line player would buy D_i, provided that no further device is released in between. Let t denote the running costs paid by the on-line player between a_i and d. We keep the invariant that the device used by the off-line player immediately before a_i is D_{i-1}. This is true for $i = 1$. Define $\phi = \frac{\sqrt{5}-1}{2}$.

Case $t \geq \phi$: Then the off-line player buys D_i at time a_i and places a_{i+1} immediately after d. This yields $y_{i+1} = y_i + t + 1$ and, on the other hand, $z_{i+1} = z_i + 1 + r_i(a_{i+1} - a_i)$.

Case $t < \phi$: Then the off-line player does not buy D_i and places a_{i+1} immediately after d. He buys D_{i+1} and defers a_{i+2} such that the on-line player must buy D_{i+1} sometime. This yields $y_{i+2} \geq y_i + t + 2$ and $z_{i+1} \leq z_i + t + 1 + r_i(a_{i+1} - a_i) + r_{i+1}(a_{i+2} - a_{i+1})$. Term t in the running costs is due to the fact that the off-line player has the second best device in use, and this invariant is recovered at a_{i+2} since he has bought D_{i+1}.

Note that $\frac{t+2}{t+1} > 1 + \phi$ for $t < \phi$. Ignoring the r_i terms for the moment, we see that the on-line player has always to add at least $1 + \phi$ times the optimum cost in one or two phases. Moreover, since the on-line player runs some device older than D_i before d, a_{i+1} can be bounded in terms of earlier slopes. (This bound is not important, just its existence.) Thus every new r_i can be chosen small enough such that $r_i(a_{i+1} - a_i)$ is smaller than any desired number. Therefore the actual ratio of added costs can be made larger than $1 + \phi - \delta = c - \delta$.

We conclude $\lim y_i/z_i > c - \delta$, hence $y_n/z_n > c - \delta$ for large enough n. $\square$

Concerning the relationship to the discrete-time model, similar remarks as in Section 2 apply.

Acknowledgement

I would like to thank an anonymous referee for a number of important remarks.

References

[1] Y. Azar, Y. Bartal, E. Feuerstein, A. Fiat, S. Leonardi, A Rosén: On capital investment, *Algorithmica* 25 (1999), 22-36

[2] A. Borodin, R. El-Yaniv: *Online Computation and Competitive Analysis*, Cambridge Univ. Press 1998

[3] P. Damaschke: Multiple spin-block decisions, *10th ISAAC'99*, *LNCS* 1741, 27-36

[4] P. Damaschke: Online strategies for backups, *4th CIAC'2000*, *LNCS* 1767, 63-71, to appear in a special issue of *Theoretical Computer Science*

[5] D.R. Dooly, S.A. Goldman, S.D. Scott: On-line analysis of the TCP acknowledgment delay problem, *J. of the ACM* 48 (2001), 243-273

[6] R. El-Yaniv, R. Kaniel, N. Linial: Competitive optimal on-line leasing, *Algorithmica* 25 (1999), 116-140

[7] A.R. Karlin, M.S. Manasse, L.A. McGeoch, S. Owicki: Competitive randomized algorithms for non-uniform problems, *Algorithmica* 11 (1994), 542-571

[8] P. Krishnan, P.M. Long, J.S. Vitter: Adaptive disk spindown via optimal rent-to-buy in probabilistic environments, *Algorithmica* 23 (1999), 31-56