On Capital Investment

Yossi Azar[1], Yair Bartal[2], Esteban Feuerstein[3], Amos Fiat[1], Stefano Leonardi[4] and Adi Rosén[5]

[1] Department of Computer Science, Tel Aviv University. §
[2] International Computer Science Institute, Berkeley. ¶
[3] Depto. de Computacion, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires & Instituto de Ciencias, Universidad de General Sarmiento. ‖
[4] International Computer Science Institute, Berkeley, & Dipartimento di Informatica Sistemistica, Università di Roma "La Sapienza". **
[5] Department of Computer Science, University of Toronto. ††

Abstract. We deal with the problem of making capital investments in machines for manufacturing a product. Opportunities for investment occur over time, every such option consists of a capital cost for a new machine and a resulting productivity gain, i.e., a lower production cost for one unit of product. The goal is that of minimizing the total production and capital costs when future demand for the product being produced and investment opportunities are unknown. This can be viewed as a generalization of the ski-rental problem and related to the mortgage problem [3].

If all possible capital investments obey the rule that lower production costs require higher capital investments, then we present an algorithm with constant competitive ratio.

If new opportunities may be strictly superior to previous ones (in terms of both capital cost and production cost), then we give an algorithm which is $O(\min\{\log C, \log\log P, \log M\})$ competitive, where C is the ratio between the highest and the lowest capital costs, P is the ratio between the highest and the lowest production costs, and M is the number of investment opportunities. We also present a lower bound on the competitive ratio of any on-line algorithm for this case which is $\Omega(\max\{\log C, \frac{\log\log P}{\log\log\log P}, \frac{\log M}{\log\log M}\})$. This shows that the competitive ratio of our algorithm is tight (up to constant factors) as a function of C, and not far from the best achievable as a function of P and M.

§ Research supported in part by a grant from the Israel Academy of Sciences. e-mail: {azar,fiat}@math.tau.ac.il.

¶ Research supported in part by the Rothchild Postdoctoral fellowship. email: yairb@icsi.berkeley.edu.

‖ Research supported in part by EEC project KIT-DYNDATA. email: efeuerst@dc.uba.ar.

** Research supported in part by EU Esprit Long Term Research project ALCOM IT under contract 20244, and by Italian Ministry of Scientific Research Project 40% "Algoritmi, Modelli di Calcolo e Strutture Informative". email: leon@dis.uniroma1.it.

†† email: adiro@cs.toronto.edu.

1 Introduction

We consider the problem of manufacturing costs versus capital investment. A factory uses machines for producing units of some product. The production of each unit requires some fixed cost for using the machine (electricity, raw material, etc.). Over time opportunities for investment in new machines, that would replace the old ones, become available. Such opportunities could be the result of technological improvement, relocation to a cheaper market, or any other investment that would replace the facilities of the factory and would lead to lower production costs. We model all these opportunities as machines that can be bought, and then used to produce the units of the product. The factory must decide if to invest in buying new machines to reduce production costs while neither future demand for the product nor future investment opportunities are known.

Many financial problems require to take decisions without having knowledge, or while having only partial knowledge, of future opportunities. Competitive analysis of financial problems has received an increasing attention during the last years, for instance for currency exchange problems [2] or asset allocation [5].

The problem considered in this paper is a generalization of one of the basic on-line problems, the *ski-rental* problem due to L. Rudolph (see [4]), a model for the well known practical problem "rent or buy?". The ski-rental problem can be stated as follows: you don't know in advance how many times you will go skiing; renting a pair of skis costs \$$r$; to purchase your own pair costs \$$p$. When do you buy? It is not hard to see that the best competitive ratio is obtained if you buy when the total rental cost (thus far) is equal to the cost of buying your own pair. Another problem considered in this model in the past is the so-called mortgage problem [3], where a fluctuating mortgage rate and associated re-financing charges lead to the question, re-finance or not?

While for the ski-rental problem the only possible capital expenditure is to purchase a pair of skis, and then the "production" costs drop to zero, in the capital investment problem there may be many future capital expenditure options and the resulting productivity gains are unknown. Unlike the mortgage problem where the future demand is the servicing the entire debt (which is a known fixed value), and a capital investment has a fixed cost, in the capital investment problem future demand is unknown and capital investments may have arbitrary costs.

We consider two models for our problem, and call the first one the *convex case*. Here, we assume that to get a lower production cost, one must spend more as capital expenditures. In this case we get a constant competitive ratio. This scenario is usually true in manufacturing: purchasing a better machine costs more. However, sometimes technological breakthroughs are achieved, after which both machine costs and production costs are reduced. This matches our second model, the *non-convex* case, which allows both capital and production costs to drop.

In contrast to the convex case, for the non-convex case we present a lower bound on the competitive ratio of any on-line algorithm for the problem which is $\Omega(\max\{\log C, \log\log P/\log\log\log P, \log M/\log\log M\})$, where C is the ratio between the highest and the lowest capital costs, P is the ratio between the highest and the lowest production costs, and M is the number of investment opportunities. We complement this lower bound with an algorithm for general capital investment scenarios which is $O(\min\{\log C, \log\log P, \log M\})$ competitive.

2 The On-line Capital Investment Problem

Imagine a factory whose goal is to produce units of some commodity at low cost. From time to time, orders for units of the commodity arrive, and at times new machines become available. Every such machine is characterized by its *production cost*, and by its *capital cost*. The production cost is the cost of producing one unit of commodity using this machine. The capital cost is the capital investment necessary to buy the machine. We assume that once a machine becomes available, then it is available forever. We also assume that one can produce an unlimited number of units with any machine. An algorithm for this problem has to decide what machines to buy and when to do so, as to minimize the total cost (capital costs plus production costs).

More formally, an instance of the problem consists of a sequence of machines, and a sequence of orders of demand. Machine m_i is defined by the triplet (t_i, c_i, p_i), where t_i is the time at which the machine becomes available, c_i is its capital cost, and p_i is its production cost. Every order is defined by its arrival time. Without loss of generality we may assume that the j'th order appears at time j. Moreover, since any reasonable algorithm will not buy a new machine when there is no order pending, we can assume that for any machine i such that $j < t_i \leq j+1$, $t_i = j+1$. At any time t, the algorithm can buy any of the available machines (those with $t_i \leq t$), and then produce one unit of the commodity.

We say that machine m_i *dominates* machine m_j if both the production cost and the capital cost of m_i are lower than those of m_j. We call an instance of the problem *convex* if no machine presented dominates another. I.e., an instance is convex if for any two machines i, j such that $p_i < p_j$ it holds that $c_i \geq c_j$. To help distinguish between the two versions of the problem, we call the general case *non-convex*.

We note that if all machines are available at the very beginning, then all machines that are dominated by others can be removed. Thus, whenever all machines are available in advance, we are left with the convex setting. The non-convex setting only makes sense if machines appear over time and it is possible that a better machine (in terms of both capital cost and production cost criteria) will appear later.

2.1 Performance Measures

We measure the performance of an on-line algorithm for this problem by its competitive ratio [6]. Let σ be a sequence of offers of machines and orders of demand for units of the commodity to be produced.

We denote by $\mathrm{ON}(\sigma)$ the cost of the on-line algorithm ON for the problem over the sequence σ, and with $\mathrm{OPT}(\sigma)$ the cost of an *optimal* off-line algorithm that knows the entire sequence σ in advance. We parameterize the sequences by the ratio between the cost of the most expensive and cheapest machines (denoted by C), by the ratio between the highest and the lowest production cost (denoted by P), and by the total number of machines presented during the sequence (denoted by M). Denote by $\Sigma(C, P, M)$ the set of sequences that obey the above restrictions.

The competitive ratio of an algorithm may be a function of the above parameters. An on-line algorithm ON is $\rho(C, P, M)$-competitive for a set $\Sigma(C, P, M)$ of sequences if

$$\sup_{\sigma \in \Sigma(C,P,M)} \frac{\mathrm{ON}(\sigma)}{\mathrm{OPT}(\sigma)} \leq \rho(C, P, M).$$

3 Upper Bound for the Convex Case

In this section we study the convex case in which a machine with a lower production cost cannot be cheaper than a machine with a higher production cost. We present an on-line algorithm for the convex case with competitive ratio 7.

3.1 The Algorithm

The algorithm is defined as follows: before producing the first unit the algorithm buys the machine m_i that minimizes $p_i + c_i$ amongst all machines available at the beginning of the sequence. It then produces the first unit of commodity. The initial cost $p_i + c_i$ is considered a *production* cost.

Let α and β be positive constants satisfying $2/\alpha \leq 1$ and $1/\alpha + 2\beta \leq 1$. In particular we choose $\alpha = 2$ and $\beta = 1/4$.

Before producing any subsequent unit of commodity the algorithm considers buying a new machine. However, it is not always allowed to buy a new machine. When an amount of c is spent as capital cost to buy a machine, it is not allowed to buy another machine until the algorithm spends at least $\beta \cdot c$ on production.

When it is allowed to buy a machine, the algorithm buys the machine m_i that minimizes production cost p_i amongst all machines of capital cost at most α times the total production cost incurred since the beginning of the sequence. If no such machine is available, the algorithm does not buy a new machine.

3.2 Analysis

We prove that the competitive ratio of the above algorithm is $1 + \alpha + 1/\beta = 7$.

We use the following notation. Fix the sequence σ. Denote by $\mathrm{ON} = \mathrm{ON}^c + \mathrm{ON}^p$ the total cost of the algorithm that is equal to the sum of the total capital cost ON^c and the total production cost ON^p. Let p^t be the production cost incurred by the on-line algorithm to produce unit number t. Let ON_t^p be the production cost incurred by the algorithm to produce the first t units, i.e., $\mathrm{ON}_t^p = \sum_{i=1}^t p^i$. Let OPT_t be the optimal total (capital and production) cost to produce the first t units. We start by proving a bound on the total cost spent on purchasing machines, in terms of the total production cost incurred.

Lemma 1. *The total capital cost* ON^c *incurred by the on-line algorithm is at most* $(\alpha + 1/\beta)$ *its total production cost* ON^p*.*

Proof. The capital cost of the last machine bought is at most α times the total production cost. For every other machine, the production costs in the interval between the time this machine has been bought, and the time the next machine is bought, is at least β times the capital cost of the machine. These intervals do not overlap, and thus the total capital cost of all the machines except the last one sums to at most $1/\beta$ times the total production cost. ■

We now relate the production cost of the on-line algorithm to the total cost of the off-line algorithm.

Lemma 2. *At any time t the production cost* ON_t^p *of the on-line algorithm is at most the total cost* OPT_t *of the off-line algorithm.*

Proof. We prove the claim by induction on the number of units produced.

For $t = 1$ the claim holds since the on-line production cost of the first unit (defined as the sum of the capital and the production costs of the first machine bought) is the minimum possible expense to produce the first unit. Therefore $\mathrm{ON}_1^p \leq \mathrm{OPT}_1$.

Consider unit t for $t > 1$, and assume the claim holds for any unit $t' < t$. Let m be the machine used by the on-line algorithm to produce unit t. Let m' be the machine used by the optimal off-line solution to produce unit t, p' its production cost, and c' its capital cost.

If $p' \geq p^t$ then we have $\mathrm{ON}_t^p = \mathrm{ON}_{t-1}^p + p^t \leq \mathrm{OPT}_{t-1} + p' \leq \mathrm{OPT}_t$.

If $p' < p^t$ then the on-line algorithm did not buy machine m' just before producing unit t. Let the capital cost of the last machine bought by the on-line algorithm (i.e. m) be $\bar{c}$, and assume it was bought just before unit $\bar{t}$ was produced. Since we consider the convex case we have that $p' < p^t = p^{\bar{t}}$ implies $c' \geq \bar{c}$.

As we assume that the on-line algorithm did not buy m' just before producing unit t, one of the following holds:

1. The capital cost of machine m' was too high, i.e., less than $\frac{1}{\alpha}c'$ was spent on production since the start of the sequence.
2. It was not allowed to buy any machine at this time: less than $\beta \cdot \bar{c}$ was spent on production since machine m was bought, and until unit number $t-1$ is produced.

We consider each of these cases.

For the first case we have that $\mathrm{ON}_t^p = \mathrm{ON}_{t-1}^p + p^t \leq 2 \cdot \mathrm{ON}_{t-1}^p \leq \frac{2}{\alpha}c' \leq \mathrm{OPT}_t$. For the second case we have

$$\mathrm{ON}_t^p = \mathrm{ON}_{\bar{t}-1}^p + \sum_{i=\bar{t}}^{t-1} p^i + p^t \leq \mathrm{ON}_{\bar{t}-1}^p + 2\sum_{i=\bar{t}}^{t-1} p^i < \mathrm{ON}_{\bar{t}-1}^p + 2\beta \cdot \bar{c}\ .$$

We now distinguish between two cases, depending on whether machine m' is available before unit $\bar{t}$ is produced. The first case is that machine m' becomes available only after unit $\bar{t}$ is produced. In this case we have

$$\mathrm{ON}_t^p < \mathrm{ON}_{\bar{t}-1}^p + 2\beta \cdot \bar{c} \leq \mathrm{OPT}_{\bar{t}-1} + 2\beta \cdot \bar{c} \leq \mathrm{OPT}_{\bar{t}-1} + \bar{c} \leq \mathrm{OPT}_{\bar{t}-1} + c' \leq \mathrm{OPT}_t\ .$$

The second case is when machine m' is available before unit number $\bar{t}$ is produced. We have that its capital cost, c', is higher than $\alpha \cdot \mathrm{ON}_{\bar{t}-1}^p$, otherwise the on-line algorithm would have bought this (or a better) machine at time $\bar{t}$, which contradicts $p^{\bar{t}} > p'$. Therefore we have

$$\mathrm{ON}_t^p < \mathrm{ON}_{\bar{t}-1}^p + 2\beta \cdot \bar{c} \leq \mathrm{ON}_{\bar{t}-1}^p + 2\beta \cdot c' \leq (1/\alpha)c' + 2\beta \cdot c' = (1/\alpha + 2\beta)c' \leq \mathrm{OPT}_t\ .$$

■

Combining Lemma 1 and Lemma 2 we get the following theorem.

Theorem 3. *The algorithm presented above for the convex case of the on-line capital investment problem is $(1 + \alpha + 1/\beta)$-competitive.*

4 Lower Bound for the Non-Convex Case

In contrast to the constant upper bound proved in the previous section, in this section we prove a lower bound of $\Omega(\max\{\log C, \log\log P/\log\log\log P, \log M/\log\log M\})$ on the competitive ratio of any on-line algorithm, where C is the ratio between the highest and the lowest capital costs, P is the ratio between the highest and the lowest production costs, and M is the number of presented machines.

We now describe the instance of the problem on which the lower bound is achieved. We let C be some large power of 2. The capital costs of all the machines in the instance are powers of 2 between 1 and C, and their production costs will be of the form $1/\log^k C$, for some integer k.

We assign a *level* between 0 and $\log C$ to each machine; machines of level i have capital cost $c_i = 2^i$.

We say that a *phase of level* i starts when a machine of level i is presented. A phase of level i ends when one of the following occurs:

1. The on-line algorithm buys a machine of level i.
2. The on-line algorithm has reached a global cost (production and capital) in the phase greater or equal to
   - 1 for $i = 0$;
   - 2 for $i = 1$;
   - $\frac{i}{2}c_i$ for $i \geq 2$.
3. A phase of level higher than i ends.

Immediately after the end of the phase a new machine of the same level is presented and a new phase of the same level starts.

Let $n_k(i) = \frac{i!}{k!}$ for $i = 1, \ldots, \log C$, $k = 1, \ldots, \log C - 1$, and let $n_0(i) = 2i!$. When a phase of level i with an associated machine of production cost p ends in Case 1 or Case 2, a set of $i+1$ machines are presented, one for each level $j = 0, \ldots, i$. The production cost of the appropriate machine of level j is defined to be

$$p_j = \frac{p}{(\log C)^{1+\sum_{k=0}^{j-1} n_k(j)}}.$$

At the beginning we assume that a phase of level $i = \log C$ and $p = 1$ ends, so that a first set of machines is presented, with capital and production costs as defined above.

The sequence will be over with the end of the phase of level $\log C$ associated with the machine of capital cost C presented at the beginning. The sequence is built so that there is only one machine of capital cost C presented in the whole sequence, and that machine's production cost is at most $1/\log C$ the production cost of any other machine presented in the sequence.

We define a relation of inclusion between phases. A phase of level i contains all the phases of level $j < i$ that start simultaneously or during the level i phase. Note that no phase of level $j > i$ starts during a phase of level i.

We call a phase *active* if it is not yet ended. At every point in time one phase is active at every level.

We call a phase that ends in Case 1 or Case 2 a *complete phase* and a phase that ends in Case 3 an *incomplete phase*. If a phase of level i is complete then the i phases at lower levels that have ended as a consequence of the end of this level i phase are incomplete.

Lemma 4. *At most i machines of level $i-1$ are presented during a phase of level i for $i \geq 2$, and at most 2 machines of level 0 are presented during a phase of level 1.*

Proof. For $i = 1$, the cost incurred by the on-line algorithm in every complete phase of level 0 is at least 1, thus after at most 2 level 0 phases the on-line cost will reach 2, and the level 1 phase will end in Case 2. For $i \geq 2$, a new machine of level $i-1$ is presented when the on-line algorithm buys the previous one of that level or when its cost reaches $\frac{i-1}{2} c_{i-1}$. In any case, the on-line algorithm's cost for the phase of level $i-1$ is at least c_{i-1}. Hence, the maximum number x of phases of level $i-1$ is restricted to be $x c_{i-1} \leq \frac{i}{2} c_i$, which implies $x \leq i$. ■

The production costs defined above were chosen so as to obey the property stated in the following lemma.

Lemma 5. *A machine of level i has production cost less or equal to $1/\log C$ times the production cost of any machine of level $k \leq i$ presented before the starting of the phase, and of any machine of level $k < i$ presented during the phase.*

Proof. A new machine of level i is presented when a phase of level $j \geq i$ ends. Let p be the production cost of the machine associated with the phase of level j that has just ended. The production cost of the machine associated with the new phase of level i is $p_i = \frac{p}{(\log C)^{1+\sum_{k=0}^{i-1} n_k(i)}}$. We prove the claim by induction. If a new phase of level i starts then a previous phase of level i has just ended. Say $\tilde{p}_i$ is the production cost of the associated machine. We know that $\tilde{p}_i \geq p \geq p_i \ \log C$. Since, by induction, the claim is true for the previous phase of level i, with production cost $\tilde{p}_i$, then the production cost p_i of the machine presented in the new phase of level i is less or equal than $1/\log C$ times the production cost of any machine of level $k \leq i$ presented before the start of the phase.

Let us prove the second part of the claim, i.e. that every machine presented in the phase has production cost at least $p_i \ \log C$. First we prove it for a phase of level $i = 1$. It contains at most 2 machines of level 0, with production cost $\frac{p}{\log C}$ and $\frac{p}{\log^2 C}$. Since $p_i = \frac{p}{\log^3 C}$, the claim is proved.

Finally, we prove the claim for $i > 1$. We prove it for the machine associated to the last phase of level $i-1$ contained in the phase of level i, by induction the machine with lowest production cost presented in the phase. In a phase of level i at most i machine of level $i-1$ are presented . A new machine is presented when the previous phase of level $i-1$ is stopped because of Case 1 or Case 2. (Recall that no phase of level higher than i ends during a phase of level i.) Hence, the production cost p'_{i-1} of the last machine of level $i-1$ is $p'_{i-1} = \frac{p}{(\log C)^{i(1+\sum_{k=0}^{i-2} n_k(i-1))}} \geq \frac{p}{(\log C)^{\sum_{k=0}^{i-1} n_k(i)}} \geq p_i \ \log C$.
■

Consider a phase of level i. Let O_i and z_i be respectively the global cost and the production cost of the on-line algorithm during that phase. The global on-line cost in a phase is given by the production cost during the phase plus the capital cost charged to the on-line algorithm for buying machines of level not higher than i (possibly including the machine of level i if the phase ends in Case 1).

We will denote by A_i the global cost of the adversary in the case in which it is committed to buy either machines presented at the beginning of the phase of level not higher than i or machines presented during the phase that, by definition of the sequence, have level lower than i. In fact this is not a restriction since the cost paid by the adversary during the unique phase of level $\log C$ is equal to the global cost of the adversary over all the sequence.

First, we state two upper bounds on the global cost A_i of the adversary during a phase. The first upper bound considers the case in which the adversary only buys the machine of level i. Observe that the machines of lower level presented at the beginning or during the phase have production cost higher than the machine of level i, and hence, in this case, can be ignored by the adversary.

Lemma 6. *If the adversary buys the machine of level i, then $A_i \le \frac{3}{2}c_i$.*

Proof. Since the on-line algorithm has not bought the machine of level i, then the adversary produces with a production cost that is at most $\frac{1}{\log C}$ times the on-line production cost during the phase. Therefore the adversary's production cost during the phase is at most $\frac{z_i}{\log C}$. We can assume $\log C \ge 2$. For $i = 0$, we have that $A_0 \le c_0 + \frac{z_0}{\log C} \le c_0 + \frac{1}{\log C} \le \frac{3}{2}c_0$. For $i = 1$, $A_1 \le c_1 + \frac{z_1}{\log C} \le c_1 + \frac{2}{\log C} \le 3 = \frac{3}{2}c_1$. Finally, for $i \ge 2$, the phase ends as soon as the global cost of the algorithm during the phases reaches the value $\frac{i}{2}c_i$. Then we have that $z_i \le \frac{i}{2}c_i \le \frac{\log C}{2}c_i$. Therefore $A_i \le c_i + \frac{z_i}{\log C} \le c_i + \frac{c_i}{2} = \frac{3}{2}c_i$. ∎

The second upper bound on A_i considers the case in which the adversary does not buy the machine of level i, and its global cost is composed by the sum of the costs of the phases of level $i-1$ contained in the phase of level i. A phase of level i (complete or incomplete) is partitioned into a sequence of phases of level $i-1$, whose number we indicate with s_i. The last one of those phases is possibly incomplete, while the first $s_i - 1$ are complete. Thus, we get the following lemma.

Lemma 7. *Let A^j_{i-1}, $j = 1, \ldots, s_i$ be the global cost of the adversary during the j-th phase of level $i-1$. Then $A_i \le \sum_{j=1}^{s_i} A^j_{i-1}$.*

Theorem 8. *If an algorithm for the non-convex on-line capital investment problem is ρ-competitive then $\rho = \Omega(\log C)$.*

Proof. We first show that for any algorithm, the (unique) phase of level $\log C$ arrives to an end. For this it is enough to show that the global cost incurred by the on-line algorithm will eventually reach the value $\frac{\log C}{2}C$. This follows immediately from the fact that the production costs of all the machines that are presented in this instance are lower-bounded by the production cost of the machine of level $\log C$, that is strictly positive.

We will now show that any on-line algorithm pays a global cost (over the sequence) of at least $\frac{1}{6}\log C$ times the cost of the adversary.

We focus our attention on a phase of level i. The phase starts when a machine of level i is presented. By definition, one machine for each lower level is simultaneously presented. Observe that during this phase the on-line algorithm does not buy any machine of level higher that i. Otherwise the phase immediately ends, a new machine of level i is presented and a new phase of level i starts.

We prove the following inductive claim:

- $O_i \ge \frac{i}{6}A_i$ for a complete phase;
- $O_i \ge \frac{i}{6}A_i - \frac{c_i}{2}$ for an incomplete phase.

We prove the claim for each of the three cases in which a phase ends. Recall that in Case 1 and Case 2 the phase is complete and the first part of the claim must be proved, while in Case 3 the phase is incomplete and the second part of the claim must be proved. For Case 1 and Case 3 the proof is by induction on i. We assume that the claim holds for phases of level $i-1$. The claim is obviously true for $i = 0$.

1. In Case 1, the on-line algorithm buys the machine of level i before the global production cost has reached the value $\frac{i}{2}c_i$. Then, the global cost of the on-line algorithm in the phase is given by the sum of the costs for each of the s_i phases of level $i-1$ contained in the phase of level i, plus the capital cost c_i for buying the machine of level i that ends the phase. Let O_{i-1}^j be the global cost of the on-line algorithm during the j-th phase of level $i-1$. Without loss of generality we consider that the last phase of level $i-1$ is an incomplete phase (the inductive hypothesis is otherwise stronger). Then

$$\begin{aligned} O_i &= \sum_{j=1}^{s_i} O_{i-1}^j + c_i \geq \sum_{j=1}^{s_i} \frac{i-1}{6} A_{i-1}^j - \frac{c_{i-1}}{2} + c_i \\ &\geq \frac{i-1}{6} A_i + \frac{3}{4} c_i \geq \frac{i}{6} A_i \ . \end{aligned}$$

   The first inequality stems by applying the inductive hypothesis. The second inequality is obtained from Lemma 7 and the relation $c_i = 2c_{i-1}$. Finally, the last inequality follows from Lemma 6.
2. In Case 2 the global cost of the on-line algorithm has reached the value $\frac{i}{2}c_i$. Then, applying Lemma 6, it follows that

$$O_i = \frac{i}{2} c_i \geq \frac{i}{6} A_i .$$

3. In Case 3 the phase ends because a new machine of the same level is presented, i.e., a phase of a higher level ends in Case 1 or Case 2. The global cost of the on-line algorithm for an incomplete phase is obtained by summing up the global cost for every phase of level $i-1$ contained in the incomplete phase of level i. Clearly, the capital cost of the machine of level i is not paid by the on-line algorithm. Note that in this case the last phase of level $i-1$ is also incomplete. The claim is proved as follows:

$$\begin{aligned} O_i &= \sum_{j=1}^{s_i} O_{i-1}^j \geq \sum_{j=1}^{s_i} \frac{i-1}{6} A_{i-1}^j - \frac{c_{i-1}}{2} \\ &\geq \frac{i-1}{6} A_i - \frac{c_i}{2} + \frac{c_i}{4} \geq \frac{i}{6} A_i - \frac{c_i}{2} \end{aligned}$$

   The first equality indicates the on-line global cost in the phase, while the first inequality is derived by applying the inductive hypothesis. The second inequality is obtained from Lemma 7 and the relation between the capital costs of machines of level i and $i-1$, while the final inequality is derived from the upper bound on the adversary's global cost of Lemma 6.

Since the unique phase of level $\log C$ is a complete phase and its completion ends the sequence, then the theorem follows from the claim on complete phases. ∎

The following corollary states the lower bound as a function of the ratio P between the highest and the lowest production costs, and of the maximum number of presented machines M.

Theorem 9. *If an algorithm for the non-convex on-line capital investment problem is ρ-competitive then $\rho = \Omega(\frac{\log\log P}{\log\log\log P})$ and $\rho = \Omega(\frac{\log M}{\log\log M})$.*

Proof. The claim follows by observing that in the sequence for the $\Omega(\log C)$ lower bound, the ratio between the maximum and the minimum production cost is $P = (\log C)^{(1+\sum_{k=0}^{\log C-1} n_k(\log C))} = ((\log C)^{O(\log C)!})$ and the number of machines presented is $M \leq 1 + \sum_{k=0}^{\log C-1} n_k(\log C) = O((\log C)!)$. ■

5 Upper Bound for the Non-Convex Case

In this section we present an algorithm for the general (non-convex) case of the problem. This algorithm achieves a competitive ratio of $O(\min\{\log C, \log\log P, \log M\})$.

5.1 The Algorithm

Given any new machine with production cost p_i, and capital cost c_i, our algorithm first rounds these costs up to the nearest power of two, i.e., if $2^{j-1} < c_i \leq 2^j$ then set $c_i = 2^j$, and if $2^{k-1} < p_i \leq 2^k$ then set $p_i = 2^k$.

The algorithm is defined as follows. Before producing the first unit buy the machine m_i that minimizes $p_i + c_i$ amongst all machines available at the beginning of the interval. It then produces the first unit of commodity. The initial cost $p_i + c_i$ is considered a *production* cost.

Before producing any subsequent unit, order all available machines by increasing production cost and (internally) increasing capital cost. Number the machines by index i, and let p_i, c_i be the production costs and capital costs, respectively. For all i, $p_i \leq p_{i+1}$, and if $p_i = p_{i+1}$, then $c_i \leq c_{i+1}$. Buy the machine with least i that satisfies the two following conditions:

- Its production cost p_i is smaller than the production cost of the current machine.
- A production cost of at least c_i has been spent since the last time a machine with capital cost c_i has been bought (or since the beginning of the run, if no such machine has been previously bought).

5.2 Analysis

We prove that the above algorithm has competitive ratio of $O(\min\{\log C, \log\log P, \log M\})$. In the following analysis we assume that all capital and production costs are indeed powers of 2, as rounded by the on-line algorithm. Clearly, an adversary that uses this modified sequence incurs a cost of at most twice the cost incurred by the real adversary that uses the real sequence.

Denote by $\mathrm{ON} = \mathrm{ON}^c + \mathrm{ON}^p$ the total cost of the algorithm which is equal to the sum of the total capital cost ON^c and of the total production cost ON^p.

Lemma 10. *The total capital cost* ON^c *is at most* $O(\log C)$ *times the total production cost* ON^p.

Proof. For a given j, consider all the machines of cost 2^j that are bought. A machine of cost 2^j can be bought only after an amount of 2^j has been spent on production since the last time a machine of the same cost has been bought (or since the beginning of the sequence, if not such machine was previously bought). It follows that for any j, the total cost of the algorithm for buying machines of cost 2^j is at most ON^p. Since there are at most $\lceil \log C \rceil$ different costs for the machines, $\mathrm{ON}^c = O(\mathrm{ON}^p \cdot \log C)$. ■

Lemma 11. *The total capital cost* ON^c *is at most* $O(\log M')$ *times the total production cost* ON^p, *where* M' *is the total number of machines bought.*

Proof. Let 2^l be the cost of the cheapest machine, and let 2^k be the cost of the most expensive machine such that $2^k \leq \mathrm{ON}^p$. All machines bought by the algorithm have costs between 2^l and 2^k. For any j, $l \leq j \leq k$, let b_j be the number of machines of cost 2^j bought by the algorithm. An upper bound on the capital cost spent by the algorithm is the maximum of $Z = \sum_{j=l}^{k} b_j 2^j$ as a function of the variables $b_j, j = l, \ldots, k$ subject to constraints $b_j 2^j \leq \mathrm{ON}^p$, and $\sum_{j=l}^{k} b_j = M'$.

We relax the problem by allowing the variables b_j to assume non-integer values. Clearly the solution to this relaxed problem is also an upper bound on ON^c. Denote by $b^r_j, j = l, \ldots, k$, the variables of the relaxed problem. For the optimal solution of the relaxed problem, there are no h and h' such that $l \leq h < h' \leq k$, $b_h > 0$ and $b_{h'} < \frac{\mathrm{ON}^p}{2^{h'}}$. Otherwise, there would have been a solution with higher value of the objective function Z of the relaxed problem, achieved by reducing b_h and increasing $b_{h'}$ by the same amount, until either $b_h = 0$, or $b_{h'} = \frac{\mathrm{ON}^p}{2^{h'}}$.

From the above observation we derive an upper bound on the maximum of the objective function (and thus an upper bound on ON^c). If $\frac{\mathrm{ON}^p}{2^k} \geq M'$ then the maximum is achieved by setting $b^r_k = M'$ and $b^r_j = 0$ for $l \leq j \leq k-1$. In this case $\sum_{j=l}^{k} b_j 2^j \leq \mathrm{ON}^p$, and the lemma clearly holds.

If $\frac{\mathrm{ON}^p}{2^k} < M'$, let h^* be the maximum integer such that $\sum_{j=h^*}^{k} \frac{\mathrm{ON}^p}{2^j} \geq M'$. An upper bound on the maximum of the objective function is obtained by assigning $b^r_j = \frac{\mathrm{ON}^p}{2^j}$, $j = h^*+1, \ldots, k$, $b^r_{h^*} = M' - \sum_{j=h^*+1}^{k} b^r_j \leq \frac{\mathrm{ON}^p}{2^{h^*}}$, and $b^r_j = 0$, for $j = l, \ldots, h^*-1$. The upper bound on the value of the objective function is

$$\sum_{j=l}^{k} b^r_j 2^j \leq \sum_{j=h^*}^{k} b^r_k 2^{k-j} 2^j = \sum_{j=h^*}^{k} b^r_k 2^k \leq \mathrm{ON}^p \cdot (k - h^* + 1) .$$

It remains to show that $k - h^* + 1 = O(\log M')$. By the definition of h^*, $\sum_{j=h^*+1}^{k} 2^{k-j} b^r_k = \sum_{j=h^*+1}^{k} b^r_j = \sum_{j=h^*+1}^{k} \frac{\mathrm{ON}^p}{2^j} < M'$. Therefore, we get $\sum_{j=h^*}^{k} 2^{k-j} b^r_k \leq 3M'$ and thus $(2^{k-h^*+1} - 1) \leq \frac{3M'}{b^r_k}$. Since $\mathrm{ON}^p \geq 2^k$, it follows that $b^r_k \geq 1$, and we obtain $2^{k-h^*+1} - 1 \leq 3M'$. Since $M' \geq 1$, we obtain $k - h^* + 1 = O(\log M')$. ■

Corollary 12. *The total capital cost* ON^c *is at most* $O(\log M)$ *times the total production cost* ON^p.

Corollary 13. *The total capital cost* ON^c *is at most* $O(\log \log P)$ *times the total production cost* ON^p.

Proof. The algorithm buys a machine only if the production cost decreases. Since all production costs are powers of 2, the algorithm buys at most $O(\log P)$ machines. ■

Lemma 14. *At any time the total production cost* ON^p *of the on-line algorithm is at most twice the total cost of the off-line algorithm.*

Proof. Let p^t be the production cost incurred by the on-line algorithm to produce unit number t. Let ON^p_t be the production cost incurred by the algorithm to produce the first t units, i.e., $\mathrm{ON}^p_t = \sum_{i=1}^{t} p^i$. Let OPT_t be the lowest (optimal) cost to produce the first t units.

We prove by induction on t that $\mathrm{ON}^p_t \leq 2 \cdot \mathrm{OPT}_t$.

To produce the first unit the on-line algorithm buys the machine that minimizes the sum of production and capital costs. This is the minimum possible cost to produce the first unit. Thus, $\mathrm{ON}_1^p \leq \mathrm{OPT}_1$.

Consider unit t for $t > 1$, and assume that the claim holds for every unit number t', $t' < t$. Let m be the machine used by ON to produce unit t. Let m' be the machine used by OPT to produce unit t, p' its production cost and c' its capital cost.

If $p^t \leq p'$ then we have

$$\mathrm{ON}_t^p = \mathrm{ON}_{t-1}^p + p^t \leq 2 \cdot \mathrm{OPT}_{t-1} + p^t \leq 2 \cdot \mathrm{OPT}_{t-1} + p' \leq 2 \cdot \mathrm{OPT}_t \ .$$

We now consider the case in which $p' < p^t$. It follows that the on-line algorithm did not buy machine m' although it was available before unit t is produced. If this happens one of the following holds:

- The production cost incurred by the on-line algorithm by time $t-1$ is less than c'. On the other hand, the optimal off-line algorithm buys machine m', incurring a cost of c'. It follows that
$$\mathrm{ON}_t^p = \mathrm{ON}_{t-1}^p + p^t \leq 2 \cdot \mathrm{ON}_{t-1}^p \leq 2c' \leq 2 \cdot \mathrm{OPT}_t \ .$$
- Some machine of cost c' was previously bought by the on-line algorithm, but the production cost incurred by the algorithm since then is less than c'. Assume that such machine was bought just before unit $\bar{t}$ was produced. As unit t is produced with production cost higher than p', we can conclude that m' was not available before unit $\bar{t}$ was produced. Thus, m' was bought by the off-line algorithm after unit $\bar{t}$ is produced. On the other hand, the on-line production cost since the production of unit $\bar{t}$ is less than c'. Therefore, we have

$$\mathrm{ON}_t^p = \mathrm{ON}_{\bar{t}-1}^p + \sum_{i=\bar{t}}^{t-1} p^i + p^t \leq \mathrm{ON}_{\bar{t}-1}^p + 2\sum_{i=\bar{t}}^{t-1} p^i \leq 2 \cdot \mathrm{OPT}_{\bar{t}-1} + 2c' \leq 2 \cdot \mathrm{OPT}_t \ .$$

■

We conclude with the following theorem, whose proof is straightforward from the previous lemmata.

Theorem 15. *The competitive ratio of the on-line capital investment algorithm described above is* $O(\min\{\log C, \log\log P, \log M\})$.

References

1. A. Chou, J. Cooperstock, R. El-Yaniv, M. Klugerman and F. Leighton. The Statistical Adversary Allows Optimal Money-making Trading Strategies. In *Proceedings of the 6th Annual ACM/SIAM Symposium on Discrete algorithms*, 1995.
2. R. El-Yaniv, A. Fiat, R. Karp and G. Turpin. Competitive Analysis of Financial Games. In *Proc. of the 33rd IEEE Annual Symposium on Foundations of Computer Science*, 1992.
3. R. El-Yaniv and R.M. Karp. The Mortgage Problem. In *Proceedings of the 2nd Israeli Symposium on Theory of Computing and Systems*, pp. 304–312, June 1993.
4. R.M. Karp, On-line Algorithms Versus Off-line Algorithms: How Much is it Worth to Know the Future?. In *Proc. World Computer Congress*, 1992.
5. P. Raghavan. A Statistical Adversary for On-line Algorithms. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol 7:79-83, 1992.
6. D.D. Sleator and R.E. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Communications of the ACM*. 28:202–208, February 1985.

This article was processed using the LaTeX macro package with LLNCS style