Journal of Global Optimization **24:** 449–462, 2002.

On the On-line Number of Snacks Problem

WEIMIN MA[1,2], JANE YOU[2], YINFENG XU[1], JAMES LIU[2] and KANLIANG WANG[1]
[1]*School of Management, Xi'an Jiaotong University, Xi'an, Shaanxi, P.R. China 710049;*
[2]*Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: cswmma@comp.polyue.edu.hk)*

Abstract. In the *number of snacks problem* (*NSP*), which was originally proposed by our team, an on-line player is given the task of deciding how many shares of snacks his noshery should prepare each day. The on-line player must make his decision and then finish the preparation before the customers come to his noshery for the snacks; in other words, he must make decision in an on-line fashion. His goal is to minimize the competitive ratio, defined as $\inf_\sigma C_A(\sigma)/C_{OPT}(\sigma)$, where σ denotes a sequence of numbers of customers, $C_{OPT}(\sigma)$ is the cost of satisfying σ by an optimal off-line algorithm, and $C_A(\sigma)$ is the cost of satisfying σ by an on-line algorithm. In this paper we give a competitive algorithm for on-line number of snacks problem $P1$, the *Extreme Numbers Harmonic Algorithm* (*ENHA*), with competitive ratio $1 + p \cdot (M - m)/(M + m)$, where M and m are two extreme numbers of customers over the total period of the game, and p is a ratio concerning the cost of the two types of situations, and then prove that this competitive ratio is the best one if an on-line player chooses a fixed number of shares of snacks for any sequence of numbers of customers. We also discuss several variants of the *NSP* and give some results for it. Finally, we propose a conjecture for the on-line *NSP*.

Key words: On-line number of snacks problem, Competitive algorithms, Competitive ratio

1. Introduction

FOUNDATIONS

Many ongoing decision-making activities, such as currency exchange, stock transactions or mortgage financing, must be carried out in an on-line fashion, with no secure knowledge of future events. However, that knowledge often influences the decision result in a fatal way. Faced with this lack of knowledge, players of these decision-making games often have two choices. One is to use models based on assumptions about the future distribution of relevant quantities such as exchange rates or mortgage rates, and aim for acceptable results on the average. The other is to analyze the worst case and then make some decision. Unfortunately, these two approaches may give some on-line solutions that are far from the relevant optimal solutions.

An alternate approach in such situations — and the one we explore here — is to use *competitive analysis* (first applied to on-line algorithms by Sleator and Tarjian [1]). In this approach, the performance of an on-line strategy is measured against that of an optimal off-line strategy having full knowledge of future events.

For some measure of cost or of profit, we try to minimize the worst-case ratio of on-line cost to optimal cost or of optimal profit to on-line profit; if this ratio is bounded for all event sequences, we deem the on-line strategy to be *competitive* and call the supremum of this ratio for profit problem and the infimum of this ratio for cost problem the *competitive ratio* of this on-line strategy. An advantage of this performance measure over the traditional average-case measure is that for most nontrivial decision-making activities it is extremely difficult to come up with an accurate probabilistic model.

In devising a competitive strategy, the on-line player will not be able to escape the necessity of making some assumptions, or having some knowledge about future events, but these need not be probabilistic in nature. For example, instead of knowledge about the distribution of future numbers of customers, an on-line strategy might be based only on knowledge of the bounds on the possible numbers of customers over the period in question, and should work well no matter how erratically (unfortunately) the number of customers varies from day to day.

An on-line algorithm receives the input incrementally, one piece at a time. In response to each input portion the algorithm must generate output, not knowing the future input. In a competitive analysis an on-line algorithm A is compared to an optimal off-line algorithm *OPT*. An optimal off-line algorithm knows the entire input sequence in advance and can process it optimally. Given an input sequence I, let $C_A(I)$ and $C_{OPT}(I)$ denote the costs incurred by A and *OPT* in processing I, respectively. Algorithm A is called *á*-competitive if there exist a constant α and β such that,

$$C_A(I) \leqslant \alpha \cdot C_{OPT}(I) + \beta,$$

for all input sequences I. An analogous definition can be given for on-line maximization problems. We note that a competitive algorithm must perform well on all input sequences.

RELATED WORK

Over the past two decades, *on-line problems* and their *competitive analysis* have received considerable interest. On-line problems had been investigated already in the 1970s and early 1980s but an extensive, systematic study started only when Sleator and Tarjian [1] suggested comparing an on-line algorithm to an optimal off-line algorithm and Karlin et al. [2] coined the term competitive analysis. In the late 1980s and early 1990s, three basic on-line problems were studied extensively, namely paging, the k-server problem and metrical task systems. The k-server problem, introduced by Manasse et al. [3], generalizes paging as well as more general caching problems. The problem consists of scheduling the motion of k mobile servers that reside on the points of a metric space S. The metrical task system, introduced by Borodin et al. [4], can model a wide class of on-line problems. An on-line algorithm deals with events that require an immediate response. Future events are unknown when the current event is dealt with. The task system [4], the *k*-server problem [5], and on-line/off-line games [6] all attempt to model on-line problems and algorithms. During the past few years, apart from the three basic problems, many on-line problems have been investigated in application areas such as data structures, distributed data management, scheduling and load balancing, routing, robotics, financial games, graph theory, and a number of problems arising in computer systems.

The adversary method for deriving lower bounds on the competitive ratio has been implicitly used by Woodall [7] in the analysis of the so-called Bay Restaurant Problem. Kierstead and Trotter [8] use the adversary method in their investigation of on-line interval graph coloring. Yao [9] formulates a theorem that starts with the words “For any on-line algorithm...” and which proves the impossibility of an on-line bin-packing algorithm with a competitive ratio strictly better than 3/2. This seems to be the first result stated on the class of all on-line algorithms for a certain optimization problem, thus exploiting the distinction between on-line and off-line algorithms. The on-line *NSP* was originally proposed by us, and there is not much discussion in the literature.

OUR CONTRIBUTION

We originally proposed the on-line number of snacks problem. The problem objective is to decide how many shares of snacks should be prepared, without knowing the numbers of customers coming in. We use the adversary method to solve this problem. Namely, for any on-line algorithm, we suppose an off-line adversary — an off-line player — to design a sequence of numbers of customers in order to let the relevant competitive ratio be as great as possible, namely, to let the on-line algorithm which is chosen by on-line player performance be as bad as possible. We investigate different versions of this problem, given by varying the on-line player’s knowledge. For different versions of this on-line problem, we show that some surprisingly small competitive ratios can be achieved under very moderate assumptions about the on-line player’s knowledge concerning the numbers of customers: only the upper and lower bounds on the possible numbers of customers need to be known.

2. The number of snacks problem

2.1. PROBLEM STATEMENT

In the *number of snacks problem* an on-line player is given the job of deciding the number of snacks, without knowing how many customers will come to his noshery for these snacks over some period of time. On any given day, the on-line player must tell his staff to prepare a certain number of shares of snacks in the morning, and the customers will come to his noshery for the snacks later, e.g., at noon. The problem is designated as *on-line* because the player must determine his snack number without knowing what the future numbers of customers will be. There are then three cases as follows,

- The noshery prepares m shares of snacks, and then there are m buyers. In other words, the on-line player can sell all the snacks out without any more cost. In this situation we assume that the on-line player supplies the snack with the cost c per share, so that the total cost is $m \cdot c$.
- The noshery prepares m shares of snacks, but there are only m_1 buyers, where $m_1 < m$. Thus, the on-line player must cost c_1 per snack to produce and to deal with the $m - m_1$ shares of snacks. In this situation, the on-line player cost is $m_1 \cdot c + (m - m_1) \cdot c_1$. Considering the economic meaning, if we assume that $c_1 > c$ and let $c_1 = pc$, where $p \geqslant 1$, the above formula is changed to $m_1 \cdot c + (m - m_1) \cdot pc$.
- The noshery prepares m shares of snacks, but there are m_2 buyers and $m_2 > m$. Because at this time the on-line player must quickly supply the snacks, so the on-line player need produce the $m_2 - m$ shares of snacks at the cost c_2 per snack. In this situation, the on-line player cost is $m \cdot c + (m_2 - m) \cdot c_2 = m_2 \cdot c + (m_2 - m) \cdot (c_2 - c)$. Obviously $c_2 > c$. If we let $c_2 = qc$, where $q \geqslant 1$, similarly, we can get $m_2 \cdot c + (m_2 - m) \cdot (q - 1) \cdot c$.

Considering the following two problems,

(1) If the player knows the exact number of the buyers every time over a period of time, the on-line player can do his best as long as the same number of shares of snacks can be prepared. In this way, the player always gets the optimum cost.

(2) However, if the number of the buyers is coming in over an on-line fashion, in other words, the on-line player does not know the number of the buyers when the he must make a decision for the number of the snacks for the coming day, how does he?

Given a sequence of numbers of buyers, the snack problem is to decide how many shares of snacks should be prepared beforehand. Obviously, problem (1) is an *off-line* problem and (2) is an *on-line* problem. For this number of snacks problem, the *off-line* problem can be solved easily. The optimal solution for the *off-line* problem can be achieved as long as the player prepares a sufficient number of snacks to meet the demand by these customers each day. However, problem (2) is very difficult for the decision-maker as he does not know the number of buyers in advance. Thus he must make his decision in an on-line fashion, deciding how many shares of snacks should be prepared without any knowledge of the future number of buyers.

Considering the general model of the number of snacks problem, we denote the actual sequence of number of buyers with $\sigma = (d_1, d_2, \dots, d_n)$, where d_i means the actual numbers of buyers on the ith day. Similarly, we denote by $\sigma' = (d'_1, d'_2, \dots, d'_n)$ the on-line number sequence of the snacks, namely the on-line decision-maker preparing d'_i shares of the snacks on the ith day. Let $C_{OPT}(\sigma)$ denote the off-line optimal cost to finish the service of σ; let $C_A(\sigma)$ denote the on-line cost for the same sequence, and let c, c_1, and c_2 denote the cost under the usual, inadequate, and superfluous situations respectively. Obviously, we can get the optimal solution to the off-line problem for a certain σ,

$$C_{OPT}(\sigma) = c \cdot \sum_{i=1}^{n} d_i,$$

and for any on-line algorithm A for this problem, we denote by

$$C_A(\sigma) = c \cdot \sum_{i=1}^{n} d_i + (c_2 - c) \cdot \sum_{\substack{i=1 \\ d_i > d_i'}}^{n} (d_i - d_i') + c_1 \cdot \sum_{\substack{i=1 \\ d_i' > d_i}}^{n} (d_i' - d_i),$$

the on-line cost. For any on-line algorithm A, the competitive ratio is defined as $\inf_\sigma C_A(\sigma)/C_{OPT}(\sigma)$. A small competitive ratio implies that A can do well in comparison with *OPT*. How can we design competitive algorithms with good competitive ratios for this on-line problem?

2.2. ASSUMPTIONS ABOUT THE FLUCTUATION OF THE NUMBER OF THE CUSTOMERS

The on-line player's priori information about the number of customers sequence or its function defines particular variants of the game. Let

- M = upper bound on the possible number of customers over the whole game, and m_i = upper bound on the possible number of customers on the ith day,
- m = lower bound on the possible number of customers over the whole game, and m_i = lower bound on the possible number of customers on the ith day,
- $\Phi = M/m$ (call this the *global fluctuation ratio*), and $\Phi_i = M_i/m_i$ (call this the *local fluctuation ratio*) on ith day,
- n = the number of days of over the game's period time.

2.3. RESULTS

In this paper, some optimal on-line algorithms were produced for four variants of the number of snacks problem.

- Variant 1. The general version, problem P, without any constraint for c_1 and c_2, and with M, m and of course Φ known to the on-line player;
- Variant 2. The degenerative version of variant 1, problem $P1$, with $c_1 = c_2 - c$, namely, $p = q - 1$, where $p \geqslant 1$, M, m and of course Φ known to the on-line player;
- Variant 3. The special version, problem P', with m_i and Φ_i, where $\Phi_i = \Phi_0$, and Φ_0 means a constant fluctuation, $i = 1, 2, \ldots, n$, and without any constraint for c_1 and c_2 known to the on-line player;

- Variant 4. The degenerative special version, problem P'', with m_i and Φ_i, where $\Phi_i = \Phi_0$, $i = 1, 2, \ldots, n$, known to the on-line player and with $c_1 = c_2 - c$, namely, $p = q - 1$, where $p \geqslant 1$.

For variants 1 and 2, the game proceeds in the following fashion: the on-line player chooses a strategy at the beginning of the game and his knowledge is just about *M*, *m* and of course Φ over the global period of the game. For problem $P1$, we derive an optimal competitive algorithm *ENHA*; for problem *P*, we also derive a similar optimal competitive algorithm *TENHA*.

However, for variants 3 and 4, the on-line player only knows some local information, e.g., m_i and Φ_i, of the *i*th day of the whole period of the game in the morning, and then chooses a strategy to solve the problem. In this paper, we only investigate the case of $\Phi_i = \Phi_0$, $i = 1, 2, \ldots, n$; the problem for some general cases is still open. Further, we give a competitive algorithm *LENHA* for P'' and a competitive algorithm *TLENHA* for P'.

We also discuss the fact that all competitive algorithms for the four variants give some lower bounds of relevant competitive ratios. In other words, no other on-line algorithms can do better.

3. The degenerative version of *P*: The problem *P*1

3.1. EXTREME NUMBERS HARMONIC ALGORITHM

EXTREME NUMBERS HARMONIC ALGORITHM. *For the on-line number of snacks problem* $P1$, *if the on-line player knows the lower and upper bounds of the numbers of the customers, namely, the two extreme numbers of the customers* M *and* m, *he can always prepare* $\overline{d} = 2Mm/(M + m)$ *snacks for each day over the period of time.*

3.2. COMPETITIVE RATIO

THEOREM 1. *For the on-line number of snacks problem* $P1$, *Extreme Numbers Harmonic Algorithm is a* $(1 + p \cdot (M - m)/(M + m))$*-competitive or* $(1 + p \cdot (\Phi - 1)/(\Phi + 1))$*-competitive algorithm.*

Proof. Obviously, if M and m were known and $c_1 = c_2 - c$, namely, $p = q - 1$, where $p \geqslant 1$, holds, the on-line cost of *ENHA* (denoted by *A*) satisfies the following formula,

$$
\begin{aligned}
C_A(\sigma) &= c \cdot \sum_{i=1}^{n} d_i + (c_2 - c) \cdot \sum_{\substack{i=1 \\ d_i > d_i'}}^{n} (d_i - d_i') + c_1 \cdot \sum_{\substack{i=1 \\ d_i' > d_i}}^{n} (d_i' - d_i) \\
&= c \cdot \sum_{i=1}^{n} d_i + pc \cdot \sum_{i=1}^{n} |d_i - d_i'|
\end{aligned}
\tag{1}
$$

For the *ENHA* strategy, the adversary can obviously choose $d_i = M$ or $d_i = m$ to be the worst-case sequence of numbers of customers in order to let the competitive ratio be as great as possible. First, let $d_i = M$ and let σ_M denote the sequence with all $d_i = M$. According to *ENHA*,

$$d_i' = \overline{d} = \frac{2Mm}{M+m} \leqslant d_i = M, i = 1, 2, \ldots, n,$$

then,

$$\begin{aligned} C_A(\sigma_M) &= c \cdot \sum_{i=1}^{n} d_i + pc \cdot \sum_{i=1}^{n} |d_i - \overline{d}| \\ &= c \cdot nM + pc \cdot n \left(M - \frac{2Mm}{M+m} \right) \\ &= \left(1 + p \cdot \frac{M-m}{M+m} \right) \cdot cnM \\ &= \left(1 + p \cdot \frac{M-m}{M+m} \right) \cdot C_{OPT}(\sigma_M) \end{aligned} \tag{2}$$

Similarly, we can get the following result if we let $d_i = m$ and denote the sequence of customers under this condition by σ_m. Considering $\overline{d} \geqslant m$ holds,

$$\begin{aligned} C_A(\sigma_m) &= c \cdot \sum_{i=1}^{n} d_i + pc \cdot \sum_{i=1}^{n} |d_i - \overline{d}| \\ &= c \cdot nm + pc \cdot n \left(\frac{2Mm}{M+m} - m \right) \\ &= \left(1 + p \cdot \frac{M-m}{M+m} \right) \cdot cnm \\ &= \left(1 + p \cdot \frac{M-m}{M+m} \right) \cdot C_{OPT}(\sigma_m) \end{aligned} \tag{3}$$

The extreme situations ($d_i = M$ or $d_i = m$) are the worst possible case; in other words, no other σ, which is different from σ_m and σ_M, can lead to some worse cases in order to enlarge the competitive ratio. It seems that, for any σ,

$$\begin{aligned} \frac{C_A(\sigma)}{C_{OPT}(\sigma)} &\leqslant 1 + p \cdot \frac{M-m}{M+m} \\ &= 1 + p \cdot \frac{\Phi - 1}{\Phi + 1} \end{aligned} \tag{4}$$

The proof is completed. □

3.3. A LOWER BOUND OF THE COMPETITIVE RATIO

Actually, the *ENHA* gives a lower bound of competitive ratio.

THEOREM 2. *For on-line number of snacks problem $P1$, if the off-line adversary chooses a strategy with which a fixed number of customers is chosen for all days, the competitive ratio $(1+p\cdot(M-m)/(M+m))$ or $(1+p\cdot(\Phi-1)/(\Phi+1))$, which is given by ENHA, is a lower bound; in other words, no other better competitive ratio can be achieved.*

Proof. We need to prove that if the on-line player chooses another fixed number as his decision (denoted by algorithm A'), e.g., $\overline{d}' \neq 2Mm/(M+m)$, the competitive ratio will change to worse. Without loss of generality, we assume that $\overline{d}' < 2Mm/(M+m)$. We will then prove that if the off-line adversary chooses σ_M the competitive ratio $1+p\cdot(M-m)/(M+m)$ cannot be achieved. Under the above statement, we have

$$\begin{aligned} C_{A'}(\sigma_M) &= c\cdot\sum_{i=1}^{n} d_i + pc\cdot\sum_{i=1}^{n}|d_i-\overline{d}'| \\ &= c\cdot nM + pc\cdot n(M-\overline{d}') \\ &= \left(1+p\cdot\left(1-\frac{\overline{d}'}{M}\right)\right)\cdot cnM \\ &= \left(1+p\cdot\left(1-\frac{\overline{d}'}{M}\right)\right)\cdot C_{OPT}(\sigma_M) \\ &> \left(1+p\cdot\frac{M-m}{M+m}\right)\cdot C_{OPT}(\sigma_M) \end{aligned} \tag{5}$$

The last inequality holds for $\overline{d}' < 2Mm/(M+m)$. This means that if the off-line adversary chooses σ_M, the performance of the on-line algorithm A' is worse than A. Similarly, under the condition with $\overline{d}' > 2Mm/(M+m)$, the same result can be obtained. □

3.4. ABOUT THE UPPER BOUND OF THE COMPETITIVE RATIO

In fact, if the on-line player chooses σ'_M but the off-line player chooses σ_m, we can get an upper bound of competitive ratio for the on-line number of snacks problem. It is

$$1+p\cdot\left(\frac{M}{m}-1\right) = 1+p\cdot(\Phi-1).$$

Thus, if the on-line player chooses any integer number from m to $2Mm/(M+m)$, then he will get a relevant competitive ratio, which is between $1+p\cdot(\Phi-1)/(\Phi+1)$ and $1+p\cdot(\Phi-1)$.

4. The general version: The problem *P*

4.1. TRANSFORMATIVE EXTREME NUMBERS HARMONIC ALGORITHM

Obviously, the results from Section 3 can be used to produce an on-line algorithm for *P*. In fact, we propose an on-line algorithm for *P* as follows:

TRANSFORMATIVE EXTREME NUMBERS HARMONIC ALGORITHM: *For on-line number of snacks problem P, if the on-line player knows the lower and upper bounds of the numbers of the customers, namely, the two extreme numbers of customers* M *and* m*, then he can choose a fixed number* $\bar{\bar{d}} = Mm \cdot (p + q - 1)/(M \cdot p + m \cdot (q - 1))$ *of snacks for all time.*

4.2. COMPETITIVE RATIO

With the above on-line algorithm, we can easily obtain the following theorem.

THEOREM 3. *For on-line number of snacks problem P, the Transformative Extreme Numbers Harmonic Algorithm is a* $(1 + p \cdot (q - 1)\,(M - m)/(M)) \cdot p + m \cdot (q - 1))$*-competitive or a* $(1 + p \cdot (q - 1)\,(\Phi - 1)/(\Phi \cdot p + (q - 1))$*-competitive algorithm.*

The proof for Theorem 3 is similar to that of Theorem 1, and is omitted here.

4.3. A LOWER BOUND OF COMPETITIVE RATIO

For on-line number of snacks problem *P*, we also have the following theorem.

THEOREM 4. *For on-line number of snacks problem* P*, if the off-line adversary chooses a strategy with which a fixed number of customers is chosen for all days, the competitive ratio* $(1 + p \cdot (q - 1) \cdot (M - m)/(M \cdot p + m(q - 1))$ *or* $(1 + p \cdot (q - 1)\,(\Phi - 1)/(\Phi \cdot p + (q - 1))$*, which is given by TENHA, is a lower bound; e.g., no other better competitive ratio can be achieved.*

We also omit its proof since it is similar with the proof of Theorem 2.

5. The degenerative special version of *P*′: Problem *P*″

5.1. LOCAL EXTREME NUMBERS HARMONIC ALGORITHM

For problem P'', we have the following algorithm,

LOCAL EXTREME NUMBERS HARMONIC ALGORITHM: *For problem* P''*, if the on-line player knows that the lower bounds and local fluctuation ratios of the number of customers of the* i*th day, namely,* m_i *and* $\Phi_i = \Phi_0$*,* $i = 1, 2, \ldots, n$*,* *and if* $c_1 = c_2 - c$*, namely,* $p = q - 1$*, where* $p \geqslant 1$*, holds, he can prepare* $\overline{d}_i = 2\Phi_0 \cdot m_i/(\Phi_0 + 1)$ *snacks on the* i*th day of the game.*

5.2. COMPETITIVE RATIO

With the above algorithm, we have the following theorem,

THEOREM 5. *For problem* P''*, the LENHA is a* $(1 + p \cdot (\Phi_0 - 1)/(\Phi_0 + 1))$*-competitive algorithm.*

Proof. If m_i and $\Phi_i = \Phi_0, i = 1, 2, \dots, n$ are known and $c_1 = c_2 - c$, namely, $p = q - 1$, where $p \geqslant 1$, holds, for the *ENHA* strategy, obviously the off-line adversary can choose $d_i = \Phi_0 \cdot m_i$ or $d_i = m_i$ to be the worst case sequence of numbers of customers of the ith day in order to make the competitive ratio as great as possible. First, let $d_i = m_i$ and let σ_{m_i} denote the relevant sequence. Then according to the *LENHA* and formula (1), the on-line cost of σ_{m_i} satisfies

$$\begin{aligned} C_A(\sigma_{m_i}) &= c \cdot \sum_{i=1}^{n} d_i + pc \cdot \sum_{i=1}^{n} |d_i - \overline{d}_i| \\ &= c \cdot \sum_{i=1}^{n} m_i + pc \cdot \sum_{i=1}^{n} \left(\frac{2\Phi_0 \cdot m_i}{\Phi_0 + 1} - m_i \right) \\ &= \left(1 + p \cdot \frac{\Phi_0 - 1}{\Phi_0 + 1} \right) \cdot c \cdot \sum_{i=1}^{n} m_i \\ &= \left(1 + p \cdot \frac{\Phi_0 - 1}{\Phi_0 + 1} \right) \cdot C_{OPT}(\sigma_{m_i}). \end{aligned}$$

Step 2 holds for $\overline{d}_i \geqslant d_i = m_i$.

Similarly, we can get the following result if we let $d_i = \Phi_0 \cdot m_i$ and denote the sequence of customers under this condition by σ_{M_i}. Considering $\overline{d}_i \leqslant \Phi_0 \cdot m_i$ holds, we have

$$\begin{aligned} C_A(\sigma_{M_i}) &= c \cdot \sum_{i=1}^{n} d_i + pc \cdot \sum_{i=1}^{n} |d_i - \overline{d}_i| \\ &= c \cdot \sum_{i=1}^{n} (\Phi_0 \cdot m_i) + pc \cdot \sum_{i=1}^{n} \left(\Phi_0 \cdot m_i - \frac{2\Phi_0 \cdot m_i}{\Phi_0 + 1} \right) \\ &= \left(1 + p \cdot \frac{\Phi_0 - 1}{\Phi_0 + 1} \right) \cdot c \cdot \sum_{i=1}^{n} (\Phi_0 \cdot m_i) \\ &= \left(1 + p \cdot \frac{\Phi_0 - 1}{\Phi_0 + 1} \right) \cdot C_{OPT}(\sigma_{M_i}). \end{aligned}$$

Because the extreme situations ($d_i = \Phi_0 \cdot m_i$ or $d_i = m_i$) are the possible worst case, namely, no other σ, which is different from σ_{m_i} and σ_{M_i}, can lead to any worse cases in order to enlarge the competitive ratio. It seems that, for any σ,

$$\frac{C_A(\sigma)}{C_{OPT}(\sigma)} \leqslant 1 + p \cdot \frac{M - m}{M + m} = 1 + p \cdot \frac{\Phi_0 - 1}{\Phi_0 + 1}.$$

The proof is completed. □

5.3. A LOWER BOUND OF COMPETITIVE RATIO

Actually, the *LENHA* gives an lower bound of competitive ratio.

THEOREM 6. *For problem P'', if the on-line player chooses a fixed number as his strategy on the ith day, then the competitive ratio $(1 + p \cdot (\Phi_0 - 1)/(\Phi_0 + 1))$ is a lower bound of competitive ratio. In other words, no other better competitive ratio can be achieved.*

The proof is omitted.

6. The special version: Problem *P′*

6.1. TRANSFORMATIVE LOCAL EXTREME NUMBERS HARMONIC ALGORITHM

For problem P', we give the on-line algorithm as follows

TRANSFORMATIVE LOCAL EXTREME NUMBERS HARMONIC ALGORITHM: *For problem P', if the on-line player knows the lower bound and local fluctuation ratio of the numbers of the customers of the ith day, namely, m_i and $\Phi_i = \Phi_0, i = 1, 2, \ldots, n$ and without any constraint for c_1 and c_2, the optimal on-line algorithm is to choose a fixed number $\overline{\overline{d}}_i = \Phi_0 \cdot m_i \cdot (p+q-1)/(\Phi_0 \cdot p + (q-1))$ of snacks on the ith day of the game.*

6.2. COMPETITIVE RATIO

THEOREM 7. *For problem P', the TLENHA is a $(1 + p \cdot (q - 1) \cdot (\Phi_0 - 1)/(\Phi_0 \cdot P + (q - 1)))$-competitive algorithm.*

The proof is omitted.

6.3. A LOWER BOUND OF COMPETITIVE RATIO

Similarly, we obtain the following result, THEOREM 8. *For problem P', if the on-line player chooses a fixed number as his strategy on the ith day, then the competitive ratio* $(1 + p \cdot (q - 1) \cdot (\Phi_0 - 1)/(\Phi_0 \cdot p + (q - 1))$ *is a lower bound of the competitive ratio; in other words, no other better competitive ratio can be achieved.*

Again, the proof is omitted.

7. Valuation of the results

7.1. VALUATION OF THE COMPETITIVE RATIOS

We give some competitive algorithms for the different versions of the *NSP* problem. All the competitive ratios of these algorithms are obviously the functions of the relevant fluctuation of the numbers of the customers. For example, for problem $P1$, the competitive ratio (denoted by α) is determined by the function, $\alpha = 1 + p \cdot (\Phi - 1)/(\Phi + 1)$. If we fix the value of the p, e.g., let p=1, then we get, $\alpha = 2 - 2/(\Phi + 1)$. It is easy to see that α increases with Φ, but it has an upper bound of 2. In fact, because $\lim_{\Phi\to\infty}(1 + p \cdot (\Phi - 1)/(\Phi + 1)) = p + 1$, we always have an upper bound for the competitive ratio. Figure 1 shows that competitive ratio varies with Φ.

7.2. COMPARISON OF PROBLEMS P1 AND P$''$

In this paper, we investigate four variants of the *SNP* and obtain some competitive algorithms. Obviously, for the same sequence of numbers of the customers, making some comparison of problems $P1$ and P'' can be quite interesting. The difference between problems $P1$ and P'' is that the on-line player knows the global fluctuation of the whole game or the fluctuation of the ith day of the game. Intuitively, because the problem P'' gives more knowledge to the on-line player, the relevant competitive algorithm should perform somewhat better. Figure 2 shows the difference between the two problems. Herein, for the same sequence it is easily to know that $M = m_{\max} \cdot \Phi_0$, $m = m_{\min}$ and then $\Phi = m_{\max} \cdot \Phi_0/m_{\min}$. And from the Figure 1, if let $\Phi = 16$ and $\Phi = 2$, we can get two competitive ratios, 1.88 and 1.33, for problems P'' and $P1$ on the same sequence, respectively.

8. Conclusions and future work

A striking feature of the number of snacks problem is the conceptual simplicity of the optimal strategy. To attain a given competitive ratio, the on-line player simply defends himself against the threat of the adversary's choosing the worst sequence for his on-line strategy.

If we know only the lower bound of the number of customers, e.g. m, then what will happen? And in this case, how should the on-line player choose his strategy? For $P1$, if we can think that the upper bound of the customers is $M \to \infty$, e.g.

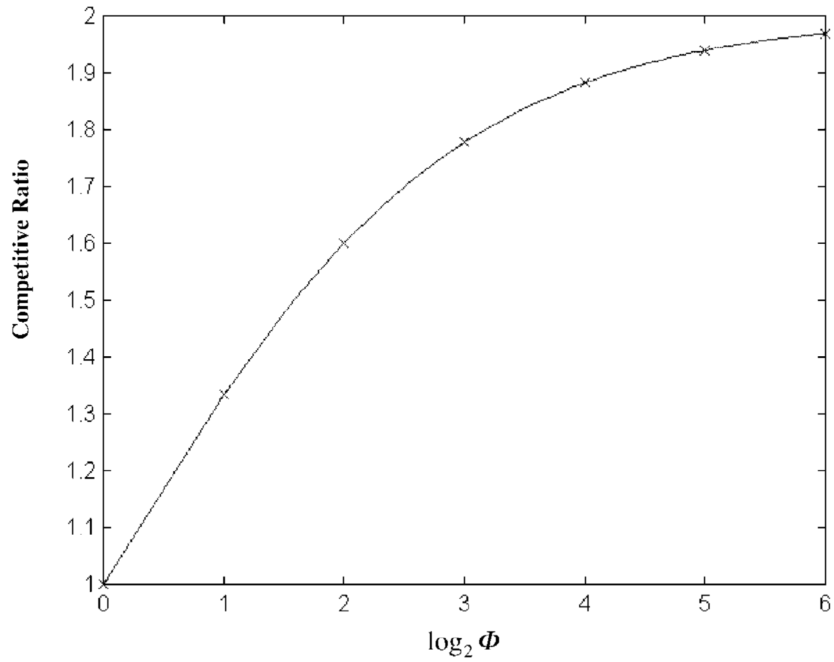

Figure 1. The curve of competitive ratio with the fluctuation Φ.

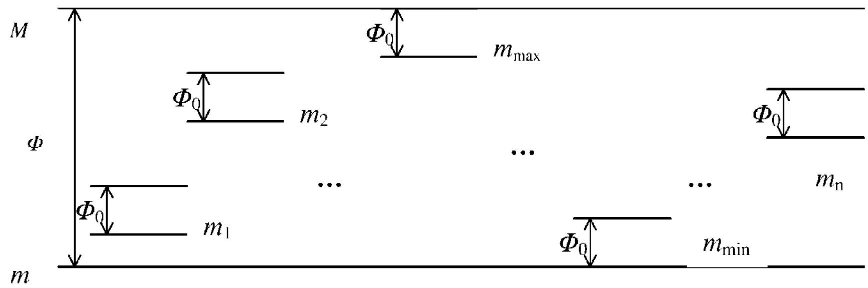

Figure 2. The differences of the problem $P1$ and problem P'' (n days game).

$\Phi \to \infty$, then the optimal on-line strategy is to prepare $2m$ snacks every day. In fact this strategy gives a $(p + 1)$-competitive algorithm for the *NSP P*1.

Some cases for the number of snacks problem are still open. For example, (1) if we don't know both M and m but just the fluctuation Φ, how can we design an on-line strategy for the number of snacks problem? (2) In this paper, we have just discussed a situation in which the off-line player chooses a fixed number to satisfy the whole sequence. For the number of snacks problem, can we design other competitive algorithms with which the on-line player can choose a varying number that is a function of a past part of customers' sequence? And with these competitive algorithms, can we get some better competitive ratio?

Above, we have derived some lower bounds for on-line snacks problems P and $P1$, given that the on-line player chooses a fixed number as his strategy. We would like to give a conjecture that these lower bounds also hold in any cases, irrespective of whether the on-line algorithm is a fixed number or not.

CONJECTURE *For any competitive algorithm of an on-line number of snacks problem,* $(1 + p \cdot (\Phi - 1)/(\Phi + 1))$ *and* $(1 + p \cdot (q - 1)\,(\Phi - 1)/(\Phi \cdot p + q - 1))$ *are the lower bounds of the competitive ratio for* $P1$ *and* P *respectively.*

Acknowledgement

The authors wish to thank Mr. Hui Chen for some helpful discussions and significant suggestions. We are also grateful for partial financial support from Central Research Grant GV-975 of the Hong Kong Polytechnic University. The authors would like to acknowledge the support of research grant from NSF of China, No. 19731001.

References

Sleator, D.D. and Tarjan, R.E. (1985), Amortized Efficiency of List Update and Paging Rules, *Communication of the ACM* 28, 202–208.

Karlin, M., Manasse, L.R. and Sleator, D.D. (1988), Competitive Snoopy Caching, *Algorithmica* 3, 79–119.

Manasse, M.S., McGeoch, L.A. and Sleator, D.D. (1988), Competitive Algorithms for On-line Problems. In *Proc. 20th Annual ACM Symp. on Theory of Computing* 322–333.

Borodin, A., Linial, N. and Sake, M. (1992), An Optimal On-line Algorithm for Metrical Task Systems. *Journal of ACM* 39, 745–763.

Manasse, M.S., McGeoch, L.A. and Sleator, D.D. (1990), Competitive Algorithms for Server Problems, *Journal of Algorithms* 11, 208–230.

Ben-david, S., Borodin, A., Karp, R.M., Tardos, G. and Wigderson, A. (1990), On the Power if Randomization in On-line Algorithms. In *Proc. 22nd Annual ACM Symp. on Theory of Computing*, 379–386.

Woodall, D.R. (1974), The Bay Restaurant: A Linear Storage Problem. *American Mathematical Monthly* 81, 240–246.

Kierstead, H.A. and Trotter, W.T. (1981), An Extremal Problem in Recursive Combinatorics, *Congressus Numerantium* 33, 143–153.

Yao, A.C.C. (1980), New Algorithm for Bin Packing, *J. Assoc. Comput. Mach.* 27, 207–227.

Albers, S. and Leonardi, S. (1999), Online Algorithms. *ACM Computing Surveys*.

El-Yaniv, R., Fiat, A., Karp, R. and Turpin, G. (1992), Competitive Analysis of Financial Games. *Foundations of Computer Science, 1992. Proceedings 33rd Annual Symposium*, pp. 327–333.

Ma, W. M., Xu, Y.F. and Wang, K.L. (2001), On-line k-Truck Problem and its Competitive Algorithm. *Journal of Global Optimization* 21(1), 15–25.