Average-Case Competitive Analyses for Ski-Rental Problems

Hiroshi Fujiwara Kazuo Iwama

Abstract

Let s be the ratio of the cost for purchasing skis over the cost for renting them. Then the famous result for the ski-rental problem shows that skiers should buy their skis after renting them $(s-1)$ times, which gives us an optimal competitive ratio of $2-\frac{1}{s}$. In practice, however, it appears that many skiers buy their skis before this optimal point of time and also many skiers keep renting them forever. In this paper we show that these behaviors of skiers are quite reasonable by using an *average-case competitive ratio*. For an exponential input distribution $f(t) = \lambda e^{-\lambda t}$, optimal strategies are (i) if $\frac{1}{\lambda} \leq s$, then skiers should rent their skis forever and (ii) otherwise should purchase them after renting approximately $s^2\lambda \ (< s)$ times. Thus average-case competitive analyses give us the result which differs from the worst-case competitive analysis and also differs from the traditional average cost analysis. Other distributions and related problems are also discussed.

1 Introduction

Suppose that the costs of renting and purchasing skis are \$100 (per ski-tour) and \$1,000, respectively. Then the well-known result for the ski-rental problem says that skiers should rent skis for the first nine $(= (1000-100)/100)$ times and should purchase them when going to the tenth. This strategy gives us an optimal competitive ratio (= 1.9 for this price setting) [Kar92]. In practice, however, many skiers seem to buy their skis *earlier* than this optimal point of time or to *keep renting* them forever. In this paper, we show that this common behavior of skiers is quite reasonable from a theoretical point of view, using *average-case competitive analyses*.

Average-case analyses for the competitive ratio of online problems have rarely appeared in the literature, although they are quite popular in other areas of algorithms. The reason is probably as follows: An online problem can be regarded as a game between an online player and an adversary. The adversary selects the input which attacks weakest points of the online player's strategy. To make this possible, the adversary must have the freedom of selecting inputs without any restriction. Note that average-case analyses assume some input distribution, which is public to the online player. The adversary of course has to follow the distribution and this can be a significant restriction against the adversary's freedom, or even destroy the essence of online games. This seems to be a common perception of most researchers who have had a negative attitude against such a model.

1.1 Our Contributions

In this paper, we reconsider this common perception, and claim that average-case competitive analyses occasionally provide us with even more interesting results compared to the conventional worst-case analysis. Our problem here is the ski-rental problem already mentioned. Let X be a

[0]We are grateful to Yasuo Okabe and Gerhard Woeginger for their useful comments. This work was supported by KAKENHI (04J00740). The final publication is available at http://dx.doi.org/10.1007/s00453-004-1142-x.

random variable for the total number of times the skier goes skiing. We assume an exponential distribution $f(t) = \lambda e^{-\lambda t}$, $\lambda > 0$, for $\Pr(X = t)$. Note that this distribution is equivalent to the following natural model: At each occasion, the skier goes skiing with probability $p = 1 - \lambda$ and quits skiing with probability $1 - p$.

Our results are summarized as follows: Let s be the ratio of the cost for purchasing skis over the cost for renting them.

(A) Both strategies, namely (i) to buy skis after renting them some constant times and (ii) to keep renting them forever, can be optimal depending on the value of λ. Note that case (ii) can never be optimal in the worst-case analysis and case (i) can never be optimal in the average *cost* analysis as described in Section 5.

(B) Case (ii) becomes optimal if $s \geq \frac{1}{\lambda}$ (= the average number of ski tours the skier makes).

(C) Otherwise, i.e. if $s < \frac{1}{\lambda}$, the optimal point of time for buying skis is approximately after renting them $s^2\lambda$ times. (For example, suppose that $s\lambda = \frac{1}{3}$. Then the skier should buy skis after renting them $\frac{1}{3}s$ times.) Thus the optimal point of time for buying skis is shifted to front compared to the worst-case analysis where the skier should buy skis after renting them $s - 1$ times.

Possible merits and demerits of our approach are as follows:

(1) Although it is true that (worst-case) competitive analyses have provided us with a lot of beautiful results which would have been impossible without this new measure, they do have limitations. For example, different algorithms such as LRU, CLOCK, FWF and FIFO for paging have all the same competitive ratio, although people have experienced a lot of difference among their empirical performances [BE98]. Average-case competitive analyses, as well as other attempts including [CN99], might help to overcome these difficulties.

(2) Unfortunately it is usually hard to estimate the input distribution. This is not so serious for the ski-rental problem since its input structure is very simple, but apparently many combinatorial problems have more complicated input structures. Nevertheless, there do exist a good number of interesting problems whose input structure might be a bit more complicated than ski-rental but is still tractable. They include the TCP acknowledgment problem [DGS98], the Bahncard problem [Fle01], and the currency conversion problem [EFKT92]. In this paper, we shall give a brief discussion on how to extend our approach to these problems.

(3) Competitive ratio sometimes seems peculiar. In ski-rental, for example, the always-rent algorithm pays \$100 and the immediately-buy algorithm \$1,000 if the skier goes skiing only once. If the skier goes 100 times, the former pays \$10,000 while the latter \$1,000. In both cases the ratio is the same 10 but there is a huge difference in the absolute cost. This might lead us to consider that the absolute cost is a better measure than the competitive ratio and the former should be used whenever possible. But there is also an opposite viewpoint: Just compare the costs of going to ski tours once and nine times. The always-rent algorithm pays \$100 and \$900; the latter is nine times as much as the former or the latter has nine times as much weight as the former if we use the absolute cost. However, both are equal to the offline optimal cost and hence to give the same competitive ratio (= 1) seems more reasonable. Thus both competitive ratio and absolute cost have their own characteristics. It should also be noted that interesting phenomena appear when the number of ski tours is relatively small, which might make the first argument going to ski tours 100 times less realistic.

1.2 Related Research

First of all, one should not confuse the average competitive ratio and the *average cost*. (Recall that the latter had been a popular measure before competitive analyses were introduced.) Although details are given later, let us take a look at the following simple example. For the same prices as before ($100 for rental and $1,000 for purchase), consider the following two algorithms: (a) Buy skis at the very beginning, and (b) buy them after renting them six times. Our input distribution is somewhat artificial: People go skiing six times in total with probability 0.5 and ten times with the same probability. Then the average competitive ratio is $(1000/600) \times 0.5 + 1.0 \times 0.5 = 1.33$ for algorithm (a) and $1.0 \times 0.5 + (1600/1000) \times 0.5 = 1.3$ for (b). However, the average cost is $1,000 for (a) and $1,100 for (b). Thus we get completely opposite results depending on which measure is used.

Karlin et al. made a great contribution to online analyses for what they call "the ski-rental-family of problems." [FW98, KMMO94] shows that there exists a deterministic online algorithm such that its average cost is $e/(e-1) \approx 1.58$ times the average offline cost without assuming any specific input distribution. They also give a randomized online algorithm for the ski-rental problem whose competitive ratio is optimal $e/(e-1)$ [KMMO94]. In their recent paper [KKR01], the randomized, optimal $e/(e-1)$ competitive ratio is extended to other problems of the family such as TCP acknowledgment [DGS98] and Bahncard [Fle01] problems.

Koutsoupias and Papadimitriou [KP94] considered a "partial knowledge" of the input distribution for online players. Raghavan [Rag92] also restricted the power of adversaries, whose input has to satisfy some statistical properties. Also, there are models which allow online players to use several powerful tools or relatively restrict the power of adversaries: Randomization used by online players is a most popular one ([BLS92] and many others). Also included in this category are to allow online players to make "forecasts" [aB97, IY99] and to allow them to maintain several different solutions and to select (the best) one of them at the end of the game [HIMT00]. The access graph model for paging also restricts the power of adversaries (e.g., [BIRS95]).

2 Average-Case Competitive Analyses

The costs of renting and purchasing skis are denoted by 1 and s, respectively. An online algorithm for this problem is determined completely by deciding how many ($= k$) times the skier should rent skis before buying them, and therefore such an algorithm is denoted by $A(k)$. In this paper, we use a continuous model just because of the ease of calculation. It should be noted that similar results are obtained by using an equivalent discrete model and we often use terminology of the discrete model, such as "in each occasion of going to a ski tour." Let $\textsc{alg}(k, t)$ and $\textsc{opt}(t)$ denote the cost of the online algorithm and the cost of the optimal offline algorithm, respectively, where t is the total number of times the skier goes skiing. Apparently,

$$\textsc{alg}(k,t) = \begin{cases} t & : \quad 0 \le t \le k, \\ k+s & : \quad k < t, \end{cases} \tag{1}$$

$$\textsc{opt}(t) = \min(s,t). \tag{2}$$

The worst-case competitive ratio for the algorithm $A(k)$ is denoted by $\max_t \frac{\text{ALG}(k,t)}{\text{OPT}(t)}$.

Lemma 1. *The following strategy provides an optimal worst-case competitive ratio of 2: The skier should purchase skis after renting s times.*

Now let $f(t)$ be a probability density function for the input distribution. Then we can define the average-case competitive ratio as:

$$c(k) = \boldsymbol{E}\left[\frac{\textsc{alg}(k,t)}{\textsc{opt}(t)}\right] = \int_0^\infty \frac{\textsc{alg}(k,t)}{\textsc{opt}(t)} \cdot f(t)dt. \tag{3}$$

As the function $f(t)$, we use a so-called exponential distribution defined by

$$f(t) = \lambda e^{-\lambda t} \quad (\lambda > 0). \tag{4}$$

As mentioned before, this distribution means that in each occasion, the skier continues to go skiing with probability $1 - \lambda$ and quits skiing with λ. Note that its mean value $\int_0^\infty t\lambda e^{-\lambda t} dt$ is equal to $\frac{1}{\lambda}$, which shows how many times in total the skier goes skiing on average.

We calculate (3) for two regions $0 < k \leq s$ and $s < k$, for which the value of $c(k)$ is denoted by $c_1(k)$ and $c_2(k)$, respectively. For $0 < k \leq s$, one can obtain

$$\begin{aligned} c_1(k) &= 1 - e^{-\lambda k} + (k+s)\int_k^s \frac{1}{t}\lambda e^{-\lambda t} dt + \frac{k+s}{s} e^{-\lambda s} \\ &= 1 - e^{-\lambda k} + \lambda(k+s)(Ei(-\lambda s) - Ei(-\lambda k)) + \frac{k+s}{s} e^{-\lambda s}, \end{aligned} \tag{5}$$

where $Ei(-x) = -\int_x^\infty \frac{e^{-t}}{t} dt$, called the exponential integral, cannot be expressed as an elementary function [MUH56]. On the other hand, for $s < k$, we can obtain

$$c_2(k) = 1 + \frac{1}{\lambda s} e^{-\lambda s} - \left(\frac{1}{\lambda s} - 1\right) e^{-\lambda k}. \tag{6}$$

Differentiating $c_1(k)$, we can get its first derived function:

$$\begin{aligned} \frac{dc_1(k)}{dk} &= -\frac{s}{k}\lambda e^{-\lambda k} + \int_k^s \frac{1}{t}\lambda e^{-\lambda t} dt + \frac{1}{s} e^{-\lambda s} \\ &= -\frac{s}{k}\lambda e^{-\lambda k} + \lambda(Ei(-\lambda s) - Ei(-\lambda k)) + \frac{1}{s} e^{-\lambda s}. \end{aligned} \tag{7}$$

Also, for $s < k$,

$$\frac{dc_2(k)}{dk} = \left(\frac{1}{s} - \lambda\right) e^{-\lambda k}. \tag{8}$$

It is also possible to obtain the second derived functions of $c(k)$, which are the following:

$$\frac{d^2 c_1(k)}{dk^2} = \lambda e^{-\lambda k} \frac{1}{k^2} (\lambda s k + s - k), \tag{9}$$

$$\frac{d^2 c_2(k)}{dk^2} = \left(\lambda - \frac{1}{s}\right) \lambda e^{-\lambda k}. \tag{10}$$

3 Optimal Online Strategies

Since we are using the continuous model, $c(k)$ diverges to $+\infty$ as k approaches zero. This is not too important because we are interested in the value of $c(k)$ for $k =$ positive integers. Optimal strategies of the online player differ depending on whether $\frac{1}{\lambda} < s$ or $\frac{1}{\lambda} \geq s$. Note that $\frac{1}{\lambda} < s$ means that the average number of total ski-tours is less than the cost of purchase.

Case 1. $\frac{1}{\lambda} < s$ (see Table 1). In this case the skier should rent skis forever since $c(k)$ decreases monotonically: One can see that $\frac{d^2 c_1(k)}{dk^2}$ is always positive for $0 < k < s$ from (9). In addition,

$$\lim_{k \to s-0} \frac{dc_1(k)}{dk} = \left(s - \frac{1}{\lambda}\right) e^{-\lambda s} < 0, \tag{11}$$

Table 1: *Case 1.* $\frac{1}{\lambda} < s$

k	0	$\cdots$	s	$\cdots$	$+\infty$
$c'(k)$		$-$		$-$	
$c(k)$	$+\infty$	$\searrow$	$1+e^{-\lambda s}$	$\searrow$	$1+\frac{1}{\lambda s}e^{-\lambda s}$

Table 2: *Case 2.* $\frac{1}{\lambda} = s$

k	0	$\cdots$	s	$\cdots$	$+\infty$
$c'(k)$		$-$		0	
$c(k)$	$+\infty$	$\searrow$	$1+e^{-1}$	$\rightarrow$	$1+e^{-1}$

which implies that $\frac{dc_1(k)}{dk}$ is negative and $c_1(k)$ decreases monotonically for $0 < k < s$. It is not hard to see that $c_2(k)$ also decreases monotonically since (8) is negative for $s < k$. Note that

$$\lim_{k\to\infty} c_2(k) = 1 + \frac{1}{\lambda s}e^{-\lambda s}, \tag{12}$$

which is the average-case competitive ratio when the skier does not buy skis. Figure 1 shows the value of $c(k)$ for $s = 10$. The curve for $\lambda = 0.2$ illustrates the current case. As one can see $c(k)$ decreases monotonically and converges to 1.0676676 when $k \to \infty$. Thus the skier following the optimal strategy suffers from only 7% more than the offline cost.

Case 2. $\frac{1}{\lambda} = s$ (see Table 2). In this case the skier should rent skis for the first s times since $c(k)$ decreases monotonically for $0 < k < s$ and is flat for $s < k$ (calculation is just the same as *Case 1*). Note that the skier can buy skis at any moment after renting s times and it is also equivalently nice to keep renting skis forever.

Case 3. $\frac{1}{\lambda} > s$ (see Table 3). In this case the skier should buy skis after renting them a certain number of times. As is shown below, $c(k)$ has a minimum peak. Namely, such a unique k, which satisfies $\frac{dc_1(k)}{dk} = 0$, exists between 0 and s. Note that $c_2(k)$ increases monotonically for $s < k$. As mentioned previously, $\frac{d^2c_1(k)}{dk^2}$ is always positive for $0 < k < s$. Then one can see that the minimum peak exists between 0 and s by the following two facts, i.e.,

$$\lim_{k\to s-0} \frac{dc_1(k)}{dk} = \left(s - \frac{1}{\lambda}\right) e^{-\lambda s} > 0, \tag{13}$$

and

$$\lim_{k\to +0} \frac{dc_1(k)}{dk} = -\infty. \tag{14}$$

The reason why (14) holds is the following: By substituting $\frac{1}{t} \leq -\frac{1}{sk}(t-s) + \frac{1}{s} \quad (k \leq t \leq s)$ to the equation of $\frac{dc_1(k)}{dk}$, we can derive

$$\frac{dc_1(k)}{dk} < \frac{1}{\lambda sk}\left\{\lambda s(1-\lambda s)e^{-\lambda k} + (1-\lambda k)(e^{-\lambda s} - e^{-\lambda k})\right\}. \tag{15}$$

Setting $k = 0$ in the term surrounded by the brackets in the right-hand side yields

$$e^{-\lambda s} - 1 + \lambda s - (\lambda s)^2. \tag{16}$$

Regarding this equation as a function of λs, it is easy to see that (16) is always finite and negative. Furthermore the right-hand side of (15) is continuous with respect to k and therefore

Table 3: *Case 3.* $\frac{1}{\lambda} > s$

k	0	$\cdots$	k_0	$\cdots$	s	$\cdots$	$+\infty$
$c'(k)$		$-$		$+$		$+$	
$c(k)$	$+\infty$	$\searrow$	$c_1(k_0)$	$\nearrow$	$1+e^{-\lambda s}$	$\nearrow$	$+\infty$

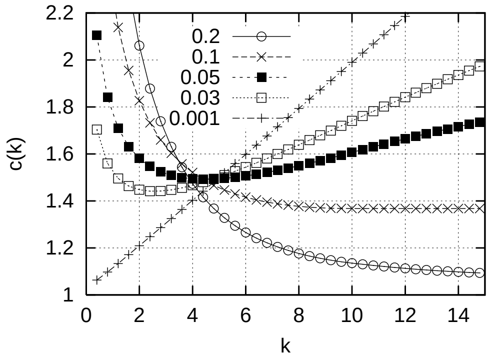

Figure 1: Average-case CR $c(k)$ ($s = 10$, $\lambda = 0.2, 0.1, 0.05, 0.03, 0.001$)

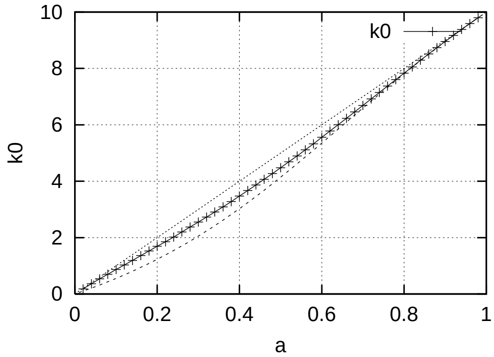

Figure 2: Value of k_0 and its upper and lower bounds

diverges to $-\infty$ as $k \to 0$. Thus, (14) holds (see Table 3). On the other hand, for $s < k$, one can easily see that $c_2(k)$ increases monotonically since (8)> 0. This *Case 3* is illustrated in Fig. 1 as the curves for $\lambda = 0.05$ and $\lambda = 0.03$. One can find a minimum peak for each curve the value of which is much smaller than 1.9 of the worst-case analysis. The value of $c(k)$ for $k = k_0$, namely an optimal competitive ratio, is written as:

$$c(k_0) = 1 - \left(1 - \lambda s - \frac{\lambda s^2}{k_0}\right) e^{-\lambda k_0}. \tag{17}$$

Case 4. $\frac{1}{\lambda} \to \infty$. In this case the skier should buy skis at the beginning since $c(k)$ increases monotonically: It turns out that $c(k) \longrightarrow 1+\frac{k}{s}$ as $\lambda \to 0$ for all range of k. This is obviously true for $0 < k \le s$. For $s < k$, applying the Mean Value Theorem to $c_2(k) = 1 + \frac{1}{\lambda s}\left(e^{-\lambda s} - e^{-\lambda k}\right) + e^{-\lambda k}$, we can conclude that $c_2(k) \longrightarrow 1 + \frac{k}{s}$, namely converging to a linear function of k.

4 Optimal Timing for Purchasing Skis

We consider *Case 3* (i.e. $\frac{1}{\lambda} > s$) in more detail to see the value of k_0 at which the average competitive ratio $c(k)$ becomes minimum. The inequality $k_0 < s$ means that the optimal point of time for purchasing skis is shifted to the front compared to the worst-case. Recall that k_0 is the solution of the equation (7)$= 0$, which seems to be hard to solve analytically. Let a be defined as $a = s\lambda$ $(0 < a < 1)$ which is the ratio of λ over $\frac{1}{s}$. Figure 2 is the result of numerical calculation using the Newton Method, which suggests the relation between a and k_0.

As for upper and lower bounds for k_0, they satisfy the following inequalities (see Fig. 2 where these lower and upper bounds are shown by dotted lines):

$$s\left\{1 - \frac{1}{a} + \sqrt{\left(1 - \frac{1}{a}\right)^2 + 1}\right\} < k_0 < sa. \tag{18}$$

We prove the upper bound first. Applying $\frac{1}{t} \geq \frac{1}{s}$ $(s \leq t \leq k)$ to the definite integral term of $\frac{dc_1(k)}{dk}$, we have

$$\frac{dc_1(k)}{dk} > -\frac{s}{k}\lambda e^{-\lambda k} + \int_k^s \frac{1}{s}\lambda e^{-\lambda t}dt + \frac{1}{s}e^{-\lambda s} = \left(\frac{1}{s} - \frac{\lambda s}{k}\right) e^{-\lambda k}. \tag{19}$$

The last expression becomes zero when $k = s^2\lambda = sa$, which means that $k_0 < sa$ holds for the solution k_0 satisfying $\frac{dc_1(k)}{dk} = 0$, since $\frac{dc_1(k)}{dk}$ increases monotonically as mentioned earlier.

We next prove the lower bound: By using $\frac{1}{t} \leq -\frac{1}{sk}(t-s) + \frac{1}{s}$ $(k \leq t \leq s)$, we have

$$\frac{dc_1(k)}{dk} < \left(\frac{1}{s} - \frac{\lambda s}{k}\right) e^{-\lambda k} + \frac{1}{sk}\int_k^s (s-t)\lambda e^{-\lambda t}dt. \tag{20}$$

We furthermore use $e^{-\lambda t} < e^{-\lambda k}$ for the definite integral term, which implies

$$\frac{dc_1(k)}{dk} < \frac{\lambda}{2sk}e^{-\lambda k}\left\{k^2 - 2\left(s - \frac{1}{\lambda}\right)k - s^2\right\}. \tag{21}$$

The last expression yields zero when $k = s - \frac{1}{\lambda} + \sqrt{\left(s - \frac{1}{\lambda}\right)^2 + s^2}$. Thus we can conclude $s\left\{1 - \frac{1}{a} + \sqrt{\left(1 - \frac{1}{a}\right)^2 + 1}\right\} < k_0$ for the same reason as before.

Let us calculate the difference of the upper and lower bounds. Let

$$g(a) = sa - s\left\{1 - \frac{1}{a} + \sqrt{\left(1 - \frac{1}{a}\right)^2 + 1}\right\}. \tag{22}$$

Then $\frac{dg(a)}{da}$ becomes zero when $a \approx 0.366$, which means the difference becomes maximum there. Since $g(0.366) < 0.099s$, we finally obtain the following theorem.

Theorem 1. *The following strategy provides an optimal average-case competitive ratio for the exponential input distribution $f(t) = \lambda e^{-\lambda t}$: (i) If $\frac{1}{\lambda} \leq s$, then the skier should rent skis forever. (ii) Otherwise, the skier should purchase skis after renting k_0 times, where k_0 satisfies $s^2\lambda - \frac{s}{10} < k_0 < s^2\lambda$. Optimal competitive ratios are given by (12) and (17) for cases (i) and (ii), respectively.*

5 Average Cost for Ski-Rental Problem

As mentioned in the first section, the average-case competitive ratio is different from the *average cost*. For the same problem and the same input distribution, the average cost $m(k)$ of the algorithm $A(k)$ can be written as:

$$m(k) = \int_0^\infty \text{ALG}(k,t) \cdot f(t)dt = \left(s - \frac{1}{\lambda}\right)e^{-\lambda k} + \frac{1}{\lambda}. \tag{23}$$

This average cost again shows different properties depending on λ. (i) If $\frac{1}{\lambda} < s$, then $m(k)$ decreases monotonically, (ii) if $\frac{1}{\lambda} = s$, then $m(k)$ is constant, and (iii) if $\frac{1}{\lambda} > s$, then $m(k)$ increases monotonically (see Fig. 3 for $s = 10$ and $\lambda = 0.2, 0.1, 0.05$). Namely, the best strategy of skiers is either to rent their skis forever or to buy them at the very beginning. Thus the strategy of buying skis after a certain times of renting them, which is a heart of competitive analysis and is quite interesting in practice also, can no longer be optimal.

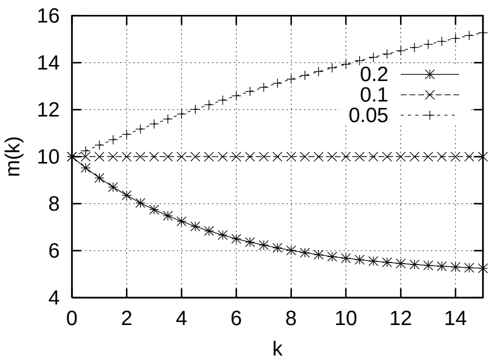

Figure 3: Average cost $m(k)$

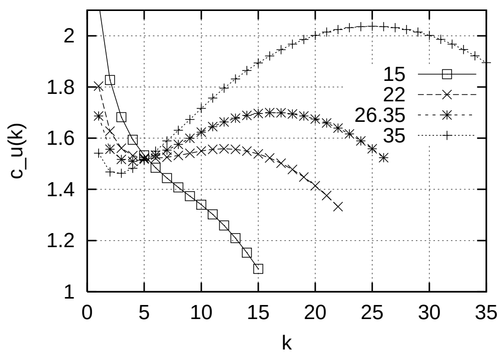

Figure 4: The average-case CR $c_u(k)$ for the uniform distribution

6 Other Input Distributions

Recall that the exponential distribution means the skier continues to go skiing with probability $1-\lambda$ (quits skiing with probability λ) which is the same at each occasion. A natural modification is to select a distribution such that this probability for continuing skiing increases or decreases as time goes. As an example of the increasing case, let us consider a *uniform distribution*

$$f_M(t) = \begin{cases} \frac{1}{M} & : \quad 0 \le t \le M, \\ 0 & : \quad t > M. \end{cases} \tag{24}$$

Namely the maximum number of ski tours is fixed ($= M$) and the probability that the skier goes skiing i ($\le M$) times in total is the same ($= \frac{1}{M}$) for all $i \le M$. (As one can see, the probability of continuing skiing at each occasion increases as time goes on.)

Average-case competitive ratio $c_u(k)$ is calculated for $c_{u1}(k)$ ($0 < k \le s$) and for $c_{u2}(k)$ ($s < k$) as before. Although details are omitted, we have

$$c_{u1}(k) = 1 - \frac{s}{M} + \frac{k+s}{M}(\ln s - \ln k) + \frac{k}{s}, \tag{25}$$

$$c_{u2}(k) = 1 + \frac{s}{2M} + \frac{M-s}{Ms}k - \frac{1}{2Ms}k^2. \tag{26}$$

Figure 4 shows the behavior of $c_u(k)$ for $s = 10$ and $M = 15$, $M = 22$, $M = 26.35$ (in this case, $c_u(k)$ becomes minimum for two different values of k), and $M = 35$. Note that if $M \le s$ then it is obvious that the skier should not purchase skis. One can see that the general tendency is similar to the case of the exponential distribution: If M is relatively small, then to continue renting skis is better. If M is large, then the skier should purchase skis at some moment, again earlier than the sth ski-tour.

7 Related Problems

7.1 TCP Acknowledgment

Suppose that n packets $P_1, P_2, \cdots, P_n$ arrive at time $a_1, a_2, \cdots, a_n$, respectively, each of which should be acknowledged. However, we do not have to send an acknowledgment for each P_i exactly at time a_i but we can postpone it. If the next packet P_{i+1} arrives while postponing the acknowledgment for P_i, it is enough to send only one acknowledgment packet to acknowledge both P_i and P_{i+1} simultaneously (similarly for three or more packets). As for the cost, we incur (1) a unit cost, called the *acknowledgment cost*, per acknowledging packet and (2) a unit cost, called the *latency cost*, per outstanding packet per unit time. If packet P_i waits 0.7 time units and P_{i+1} 0.4 time units until the acknowledgment packet is sent, for example, then the total cost for these two packets is $0.7 + 0.4 + 1 = 2.1$. (The last 1 is the acknowledgment cost and all the others are the latency cost.) More formally, suppose that k acknowledgments are sent at time $t_1, t_2, \cdots, t_k$. Then the total cost is

$$k + \sum_{1 \le j \le k} \text{latency}(j), \tag{27}$$

where

$$\text{latency}(j) = \sum_{i \text{ s.t. } t_{j-1} < a_i \le t_j} (t_j - a_i). \tag{28}$$

[DGS98] proved that the following natural online algorithm has a competitive ratio of two: The algorithm waits until the latency cost for outstanding packets becomes one, i.e., the same as a unit acknowledgment cost. [KKR01] gave the randomized algorithm which achieves the best possible competitive ratio of $e/(e-1)$. It should be noted that if we know that the number of packets is two and ignore the acknowledgment cost for the second packet (since to acknowledge it immediately is obviously optimal), then the problem becomes exactly the same as the ski-rental problem [DGS98]. Namely, the acknowledgment cost for the first packet corresponds to the ski-purchase cost and its latency cost to the ski-rental cost.

In what follows, we consider the case that the number of packets is three, under the following assumption: (i) The first packet P_1 arrives at time zero. (ii) The second packet P_2 arrives at time t_2 under the distribution that $f(t_2) = \lambda e^{-\lambda t_2}$. (iii) The third packet P_3 arrives at time $t_2 + t_3$ under the same distribution, i.e., $f(t_3) = \lambda e^{-\lambda t_3}$. The final acknowledgment cost for P_3 (and possibly for outstanding P_2 and P_1) is ignored for the same reason as before. (Recall that the two-packet TCP is equivalent to the ski-rental. One can see that the three-packet TCP is equivalent to the *two-person ski-rental* where if the two persons purchase two sets of skis at the same time then they can buy the two sets for the price of one.) Our online algorithm is denoted by $A(k)$, which does not acknowledge until the sum of latency costs for outstanding packets becomes k. (If we set $k = 1$, then, it is the same as [DGS98].) More formally, $A(k)$ operates as follows for our three-packet model: (1) If $t_2 + 2t_3 \le k$, then $A(k)$ responds for P_1, P_2 and P_3 with a single acknowledgment immediately after P_3 comes and its acknowledgment cost is ignored as mentioned before. (2) If $t_2 + 2t_3 > k$ and $t_2 \le k$, then $A(k)$ sends two acknowledgments when the latency cost becomes k and when P_3 comes. (3) If $t_2 > k$ and $t_3 \le k$, then $A(k)$ sends two acknowledgments, one for P_1 and the other for P_2 and P_3. (4) Otherwise (i.e. $t_2 > k$ and $t_3 > k$), $A(k)$ sends three acknowledgments for each packet.

Therefore the cost of $A(k)$, denoted by $\text{ALG}(k, t_2, t_3)$, can be written as follows

$$\text{ALG}(k, t_2, t_3) = \begin{cases} t_2 + 2t_3 & : \quad t_2 + 2t_3 \le k, \\ k + 1 & : \quad t_2 + 2t_3 > k,\ t_2 \le k, \\ k + 1 + t_3 & : \quad t_2 > k,\ t_3 \le k, \\ 2k + 2 & : \quad t_2 > k,\ t_3 > k. \end{cases} \tag{29}$$

The offline cost $\text{OPT}(t_2, t_3)$ is obtained as

$$\text{OPT}(t_2, t_3) = \min(t_2 + 2t_3, t_2 + 1, t_3 + 1, 2). \tag{30}$$

Now one can see that the average-case competitive ratio $c(k)$ of $A(k)$ can be written as

$$c(k) = \int_0^\infty \int_0^\infty \frac{\text{ALG}(k, t_2, t_3)}{\text{OPT}(t_2, t_3)} f(t_2) f(t_3) dt_2 dt_3. \tag{31}$$

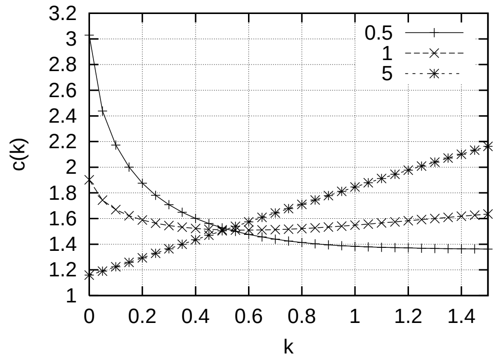

Figure 5: Average-case CR for TCP acknowledgment

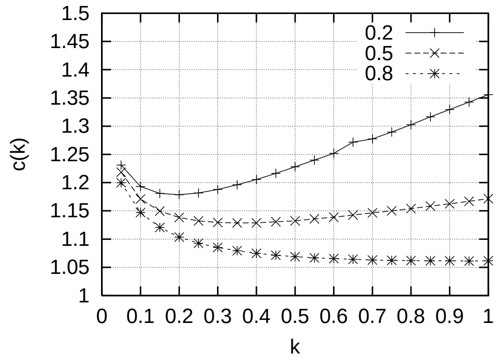

Figure 6: Average-case CR for Bahncard problem

Figure 5 shows the relation between $c(k)$ and k for $\frac{1}{\lambda} = 0.5$, 1 and 5. (Recall that $\frac{1}{\lambda}$ is equal to the average interval of packet arrivals.) If $\frac{1}{\lambda}$ is small, i.e., each packet arrives with short interval, then k should be large, or it is the best to send only one acknowledgment when P_3 comes. If $\frac{1}{\lambda}$ is large, i.e., each packet arrives with long interval, then it becomes better to acknowledge each immediately. In between (e.g., when $\frac{1}{\lambda} = 1.0$ in the figure), there is an optimal value k_0 for k and $c(k_0)$ is much smaller than the worst-case competitive ratio ($= 2.0$).

7.2 Bahncard Problem

If the online player buys a Bahncard with cost C, then subsequently, he/she can buy railway tickets for reduced prices, i.e., β (< 1) times regular prices. The algorithm is determined by the parameter k such that the player buys a Bahncard after he/she has spent a cost of $kC/(1-\beta)$ for purchasing tickets of regular prices. (Here we assume that Bahncards never expire.) It is known [Fle01] that the optimal algorithm for the worst case is to set $k = 1.0$, which achieves a CR of $2 - \beta$. Average-case analysis uses $f(t) = \lambda e^{-\lambda t}$ for the distribution of the total regular prices of tickets to be purchased. The result is very similar to ski-rental, which is illustrated in Fig. 6. When $\beta = 0.5$, $C = 10$ and $\lambda = 0.01$, the average-case competitive ratio becomes minimum when $k = 0.35$, i.e., Bahncards should be purchased much earlier than the worst-case strategy.

7.3 One-Way Currency Trading

The trader (online player) initially has one dollar and gradually exchanges it to another currency, say yen. In the one-way model, we assume that the exchange rate (= how much yen can be bought for one dollar) is monotonically increasing. However, the rate suddenly drops to the minimum at some unknown point and the game ends at that moment. The trader has to change all the remaining dollar at this minimum rate. The goal is to obtain as much yen as possible when the game ends. A well-known strategy for this game is called *threat-based* [EFKT92], which guarantees the same competitive ratio c, whenever the rate drops. If the distribution $f(t)$ for the time t when the rate drops is known, then we can obviously have more benefit by adopting our trading strategy to the distribution. Details are omitted in this paper, but the improvement of competitive ratio seems moderate compared to the other problems discussed so far. (For example, the ratio 1.92 for threat-based strategy is reduced to 1.86 for the average-case analysis.)

8 Concluding Remarks

Recall that the ski-rental problem has the following structure: (i) Its input is given as a sequence of opportunities for ski tours (as a sequence of 1's formally) whose total number is unknown. (ii) The competitive ratio is relatively independent of the absolute cost (i.e., the CR can be large when the absolute cost is small). If an online problem has such a structure, then the average-case competitive analyses appear to give us some interesting knowledge about optimal strategies. In other words, if (i) is not met, then the analysis becomes hard, and if (ii) is not met, then the result will be similar to the average-cost analysis which is usually easier. The difficulty due to (i) might be bypassed by using, for instance, numerical analyses and/or simulation. Therefore it might be more important, for the future research, to examine online problems from the second point of view, i.e., to study the independency between the competitive ratio and absolute costs.

References

[aB97] S. al Binali. The competitive analysis of risk taking with applications to online trading. In *Proc. FOCS '97*, pages 336–344, 1997.

[BE98] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[BIRS95] A. Borodin, S. Irani, P. Raghavan, and B. Schieber. Competitive paging with locality of reference. *J. Comput. Sys. Sci.*, 50:244–258, 1995.

[BLS92] A. Borodin, N. Linial, and M. Saks. An optimal online algorithm for metrical task systems. *J. ACM*, 39:745–763, 1992.

[CN99] M. Chrobak and J. Noga. LRU is better than FIFO. *Algorithmica*, 23(2):180–185, 1999.

[DGS98] D. R. Dooly, S. A. Goldman, and S. D. Scott. TCP dynamic acknowledgment delay: Theory and practice (extended abstract). In *Proc. STOC '98*, pages 389–398, 1998.

[EFKT92] R. El-Yaniv, A. Fiat, R. M. Karp, and G. Turpin. Competitive analysis of financial games. In *Proc. FOCS '92*, pages 327–333, 1992.

[Fle01] R. Fleischer. On the Bahncard problem. *Theoret. Comput. Sci.*, 268(1):161–174, 2001.

[FW98] A. Fiat and G. J. Woeginger, editors. *Online Algorithms: The State of the Art.* Springer-Verlag, London, UK, 1998.

[HIMT00] M. Halldorsson, K. Iwama, S. Miyazaki, and S. Taketomi. Online independent sets. In *Proc. COCOON '00*, volume 1858 of *LNCS*, pages 202–209. Springer, 2000.

[IY99] K. Iwama and K. Yonezawa. Using generalized forecasts for online currency conversion. In *Proc. COCOON '99*, volume 1627 of *LNCS*, pages 409–421. Springer, 1999.

[Kar92] R. M. Karp. On-line algorithms versus off-line algorithms: How much is it worth to know the future? In *IFIP 12th World Computer Congress*, pages 416–429, 1992.

[KKR01] A. R. Karlin, C. Kenyon, and D. Randall. Dynamic TCP acknowledgement and other stories about $e/(e-1)$. In *Proc. STOC '01*, pages 502–509, 2001.

[KMMO94] A. R. Karlin, M. S. Manasse, L. McGeogh, and S. Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994.

[KP94] E. Koutsoupias and C. Papadimitriou. Beyond competitive analysis. In *Proc. FOCS '94*, pages 394–400, 1994.

[MUH56] S. Moriguchi, K. Udagawa, and S. Hitotsumatsu. *Mathematics Formulas I.* Iwanami Shoten, 1956.

[Rag92] P. Raghavan. A statistical adversary for on-line algorithms. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 7, pages 79–83, 1992.