Average-Case Competitive Analysis for Ski-rental Problem with Uniform Distribution

Ali Menati(CUHK), Minghua Chen(CUHK)

1 Introduction

Suppose a skier wants to go skiing, and he has to pay 1 dollar for renting ski for one day or pay s dollars to buy his own. The skier does not know the exact number of days that he would like to ski and every morning when he goes skiing he needs to decide in an online fashion whether to rent or buy. The goal is to minimize the total cost of skiing. This problem is called ski-rental problem in the literature.
If the skier knows in advance how many days he will go skiing, it is easy to make the optimal decision. Specifically if the total number of days he will go skiing is less than s, then it is better to keep renting the ski; otherwise, he should buy the ski at the beginning. This strategy with the perfect knowledge of the future input is called the optimal **offline** algorithm, denoted as **OPT**. Meanwhile, when the number of days that skier goes skiing is not known in advance, the skier needs to make decision in an **online** fashion. In this paper, we assume that we have a distributional knowledge of the number of days that the skier goes skiing and our goal is to design distribution-aware online algorithms for the ski-rental problem.

2 Problem Formulation

Consider t to be the total number of days the skier goes skiing. We define an online algortihm $A(k)$ such that it rents ski for k days and then buys it on the $k+1$ day. Let $Cost(A(k),t)$ and $Cost(OPT,t)$ denote the cost of the online algorithm $A(k)$ and the cost of the optimal offline algorithm OPT, respectively. By the description of algorithms we have

$$Cost(A(k),t) = \begin{cases} t, & if\ 0 \leq t \leq k; \\ k+s, & otherwise. \end{cases} \tag{1}$$

and

$$Cost(OPT,t) = \min(s,t) \tag{2}$$

It should be noted that both t and k are integer numbers but we use a continuous model to make the calculation easier, and we can use this result in an equivalent discrete model.

To evaluate an online algorithm, we compare its performance to that of the optimal offline algorithm. This notion of evaluation is called competitive analysis, and the **competitive**

ratio of the online algorithm is defined as the maximum ratio between the costs of the online and offline algorithms across all possible values of t. Namely

$$CR(A(k)) \triangleq \max_t \frac{Cost(A(k),t)}{Cost(OPT,t)} \tag{3}$$

This description indicates the worst-case performance of the online algorithm among all possible inputs and guarantees the performance of the algorithm in all situations. But it provides a pessimistic estimation of the practical performance of the algorithm. This is because the worst-case competitive analysis concentrates on a single worst-case input rather than possible inputs that might appear in practice. These drawbacks can be addressed by average-case analysis where we have statical statical knowledge of the input. In this section we assume that we know that t is uniformly distributed between m and M $(0 \le m < M)$. Its PDF function is given by

$$f(t) = \begin{cases} \frac{1}{M-m}, & m \le t \le M; \\ 0, & t < m, M < t. \end{cases} \tag{4}$$

The **average-case competitive ratio**, denoted as $\widetilde{CR}$, is defined as the expectation of competitive ratios of algorithm $A(k)$ over the input distribution [1]:

$$\widetilde{CR}(A(k)) \triangleq E\left[\frac{Cost(A(k),t)}{Cost(OPT,t)}\right] = \int_0^\infty \frac{Cost(A(k),t)}{Cost(OPT,t)} f(t)\, dt\,. \tag{5}$$

This metric interprets the performance ratio as a random variable and computes its expectation. In the literature, an alternative metric for the average-case analysis is **average-cost ratio**, which is defined as the ratio of the expected cost of the online algorithm over the expected cost of the optimal offline algorithm:

$$\overline{CR}(A(k)) \triangleq \frac{E[Cost(A(k),t)]}{E[Cost(OPT,t)]} = \frac{\int_0^\infty Cost(A(k),t) f(t)\, dt}{\int_0^\infty Cost(OPT,t) f(t)\, dt}\,. \tag{6}$$

This metric is related to, but different from the average-case competitive ratio. The differences between these two metrics are as follows:

1) Average-cost ratio favors algorithms that perform well on inputs for which the optimal offline cost is large, due to the fact that inputs with small offline cost contribute little to the expected cost of the algorithm. Meanwhile, average-case competitive ratio compares the cost of the online algorithm to the cost of the optimal offline algorithm on each input separately and favors algorithms that perform well on many inputs.

2) Defining average-case competitive ratio as the expectation of performance ratios and using Markov's inequality enables us to derive upper bound on the probability that for a random t the ratio $\frac{Cost(A(k),t)}{Cost(OPT,t)}$ is more than a certain factor away from the expected competitive ratio.

$$Pr\left[\frac{Cost(A(k),t)}{Cost(OPT,t)} \ge aE\left[\frac{Cost(A(k),t)}{Cost(OPT,t)}\right]\right] \le \frac{1}{a},\ \forall\ a > 0, \tag{7}$$

which means that $\frac{Cost(A(k),t)}{Cost(OPT,t)}$ exceeds $E\left[\frac{Cost(A(k),t)}{Cost(OPT,t)}\right]$ by more than a factor of a with probability at most $\frac{1}{a}$. In other words, when an algorithm has a small average-case competitive ratio, then the probability that the performance ratio for a random t takes a large value has to also be small and for this algorithm, a very bad performance happens with a very low probability. 3) Average-cost ratio enables us to compare the average cost of the online algorithm with the average cost of the optimal offline algorithm. Meanwhile, since $E[Cost(OPT, t)]$ is constant, to compare the performance of two online algorithms, we only need to compare their average cost $E[Cost(A(k), t)]$ and choose the one with smaller average cost. Sometimes this feature makes the average-cost ratio easier to compute as compared to the average-case competitive ratio.

Both of these two metrics are very common in the literature and for different applications one may be more preferred than the other. Based on the discussion about the differences between these two metrics, we choose the average-case competitive ratio defined in (5) as our metric to evaluate the performance of our online algorithm. Because we want an algorithm that performs well on many inputs rather than favoring inputs for which the optimal offline cost is large. Our goal is to find the best deterministic online algorithm $A(k)$ for the ski-rental problem such that the average-case competitive ratio is minimized. We formulate this minimization problem as .

$$k^* = \arg\min_{k \geq 0} \widetilde{CR}(A(k)) \,. \tag{8}$$

To solve the minimization problem in (8), first, we show that we only need to consider $0 \leq k \leq M$, where M is the predicted maximum number of days that skier goes skiing. Then we divide the feasible region into two parts, first $0 \leq k < m$, where m is the predicted minimum number of days that skier goes skiing, and second $m \leq k \leq M$. For each part, we solve the problem and we find k_1, and k_2 such that

$$k_1 = \arg\min_{0 \leq k < m} \widetilde{CR}(A(k)), \quad and \quad k_2 = \arg\min_{m \leq k \leq M} \widetilde{CR}(A(k)) \,. \tag{9}$$

Now to compute k^*, we only need to compare $\widetilde{CR}(A(k_1))$ with $\widetilde{CR}(A(k_2))$, and choose k^* as follows:

$$k^* = \arg\min_{k} \{\widetilde{CR}(A(k_1)), \widetilde{CR}(A(k_2))\} \tag{10}$$

where this k^* gives us the best deterministic online algorithm $A(k^*)$ for the ski-rental problem. Here we define some notations in Table 1, which we use in the next section.

notation	definition
$c(k)$	$\widetilde{CR}(A(k))$
$\dot{c}(k)$	$\frac{d\widetilde{CR}(A(k))}{dk}$
$\ddot{c}(k)$	$\frac{d^2\widetilde{CR}(A(k))}{dk^2}$

3 Optimal Online Strategy

Without loss of generality, we assume that $m < s < M$. Because otherwise if s is less than or equal to m, the skier knows that he will go to skiing at least m days and renting m days cost him more than buying at the beginning, since he should buy ski at the beginning and if s is larger than or equal to M, the skier knows that he will go to skiing at most M days and renting M days cost him less than buying at the beginning, since he should keep renting ski until the end.

We know that skier goes skiing at most M days. Hence, $k \geq M$ means that skier never buys ski, and keeps renting the ski until the end. Lemma 1 states that for $k \geq M$ the average-case competitive ratio is constant and it is equal to $c(M)$. Therefore instead of considering $k \geq 0$, we just need to consider $0 \leq k \leq M$ as the feasible region and solve the minimization problem for this region.

Lemma 1. $c(k) = c(M), \forall\ k \geq M.$

Proof. The online algorithm $A(k)$ rents ski for k days and then buys it on the $k+1$ day. Hence $k \geq M$ means that the skier never buys the ski and keeps renting it until the end. We calculate the average-case competitive ratio for $k \geq M$ as follows:

$$\begin{aligned} c(k) &= \int_0^\infty \frac{Cost(A(k),t)}{Cost(OPT,t)} f(t)\,dt = \int_m^M \frac{Cost(A(k),t)}{Cost(OPT,t)} \frac{1}{M-m}\,dt \\ &= \int_m^s \frac{t}{t} \frac{1}{M-m}\,dt + \int_s^M \frac{t}{s} \frac{1}{M-m}\,dt = \frac{1}{M-m}\Big[\frac{M^2}{2s} + \frac{s}{2} - m\Big] = c(M) \end{aligned}$$

□

To solve the minimization problem for $0 \leq k \leq M$, we divide this region into two parts as follows:

Part 1: $0 \leq k < m$

Here m is the minimum number of days that skier goes skiing and the algorithm will rent the ski for k days and then buys it. When $0 \leq k < m$, renting the ski only increases the cost, since the skier knows that he will go skiing more than k days and then it is better to buy the ski at the beginning. We calculate the average-case competitive ratio for this part as follows:

$$\begin{aligned} c(k) &= \int_0^\infty \frac{Cost(A(k),t)}{Cost(OPT,t)} f(t)\,dt = \int_m^M \frac{Cost(A(k),t)}{Cost(OPT,t)} \frac{1}{M-m}\,dt \\ &= \int_m^s \frac{k+s}{t} \frac{1}{M-m}\,dt + \int_s^M \frac{k+s}{s} \frac{1}{M-m}\,dt \\ &= \frac{1}{M-m}\left[s\ln(\frac{s}{m}) + M - s + k[\ln(\frac{s}{m}) + \frac{M}{s} - 1]\right] \geq c(k) = c(0),\ \forall\ 0 \leq k < m. \end{aligned} \quad (11)$$

Here $c(k)$ is always increasing and $c(0)$ is the minimum average-case competitive ratio. Hence $k_1 = \arg\min_{0 \leq k < m} c(k) = 0$, where $k = 0$ means the skier buys ski at the beginning.

Part 2: $m \leq k \leq M$

Meanwhile in this part we also have $m \leq k \leq M$, therefore we consider two regions $m \leq k \leq s$, and $s < k \leq M$ for k and the value of $c(k)$ for these two region is denoted by

$c_1(k)$ and $c_2(k)$, respectively. We can obtain $c_1(k)$ for $m \leq k \leq s$ as follows:

$$\begin{aligned}
c_1(k) &= \int_0^{\infty} \frac{Cost(A(k),t)}{Cost(OPT,t)} f(t)\,dt = \int_m^M \frac{Cost(A(k),t)}{Cost(OPT,t)} \frac{1}{M-m}\,dt \\
&= \int_m^k \frac{t}{t}\,\frac{1}{M-m}\,dt + \int_k^s \frac{k+s}{t}\,\frac{1}{M-m}\,dt + \int_s^M \frac{k+s}{s}\,\frac{1}{M-m}\,dt \\
&= \frac{1}{(M-m)}\left[M-m-s+(k+s)\ln\frac{s}{k}+\frac{kM}{s}\right] \qquad (12)
\end{aligned}$$

and we can obtain $c_2(k)$ for $s < k \leq M$ as follows:

$$\begin{aligned}
c_2(k) &= \int_0^{\infty} \frac{Cost(A(k),t)}{Cost(OPT,t)} f(t)\,dt = \int_m^M \frac{Cost(A(k),t)}{Cost(OPT,t)} \frac{1}{M-m}\,dt \\
&= \int_m^s \frac{t}{t}\,\frac{1}{M-m}\,dt + \int_s^k \frac{t}{s}\,\frac{1}{M-m}\,dt + \int_k^M \frac{k+s}{s}\,\frac{1}{M-m}\,dt \\
&= \frac{1}{(M-m)}\left[M-m+\frac{s}{2}+\frac{(M-s)k}{s}-\frac{k^2}{2s}\right] \qquad (13)
\end{aligned}$$

Therefore for $c(k)$ when $m \leq k \leq M$ we have:

$$c(k) = \begin{cases} c_1(k) = \frac{1}{(M-m)}\left[M-m-s+(k+s)\ln\frac{s}{k}+\frac{kM}{s}\right], & m \leq k \leq s; \\ c_2(k) = \frac{1}{(M-m)}\left[M-m+\frac{s}{2}+\frac{(M-s)k}{s}-\frac{k^2}{2s}\right], & s < k \leq M. \end{cases} \qquad (14)$$

We can also obtain first and second derivative for this two functions as follows:

$$\dot{c_1}(k) = \frac{1}{(M-m)}[\ln s - \ln k + \frac{M}{s} - \frac{(k+s)}{k}] \qquad (15)$$

$$\dot{c_2}(k) = \frac{(M-s-k)}{(M-m)k} \qquad (16)$$

$$\ddot{c_1}(k) = \frac{s-k}{k^2(M-m)} \qquad (17)$$

$$\ddot{c_2}(k) = -\frac{1}{s(M-m)} \qquad (18)$$

Here we describe optimal online strategies for two cases

Case 1 : $M \leq 2s$

$c_1(k)$ $[m \leq k \leq s]$: Here $\ddot{c_1}(k) = \frac{s-k}{k^2(M-m)}$ is always non-negative. In addition, $\dot{c_1}(k)_{k=s} = \frac{[M-2s]}{s(M-m)}$ is non-positive. Therefore $c_1(k)$ is monotonically decreasing.

$c_2(k)$ $[s < k \leq M\ ]$: Here $\ddot{c_2}(k) = -\frac{1}{s(M-m)}$ is always negative. In addition, $\dot{c_2}(k)_{k=s} = \frac{[M-2s]}{s(M-m)}$ is non-positive. Therefore $c_2(k)$ is monotonically decreasing. Fig 1 illustrates this case. It shows the value of $c(k)$ for $m = 2$, $M = 15$, and $s = 10$.

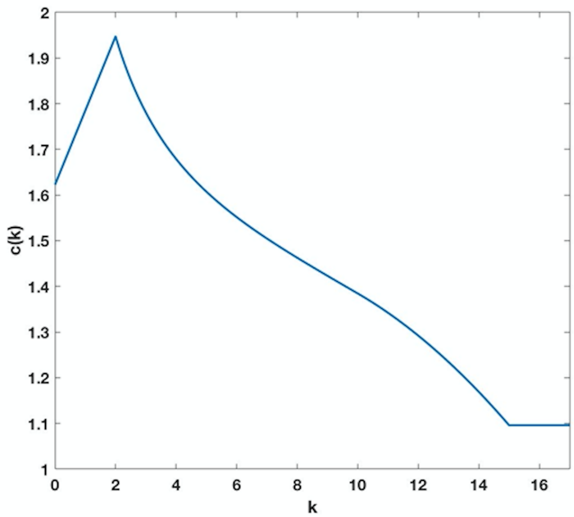


Figure 1: Case 1:($M \leq 2s$)(m=2 ,M=15 ,s=10)

As one can see in figure 1, both $c_1(k)$ and $c_2(k)$ are monotonically decreasing. We can also see that $c(k)$ is a monotonically increasing function for $0 \leq k < m$ and it a constant function for $k \geq M$.

To summarize the results for the case when $M \leq 2s$: $c(k)$ is a continuous function of k and it is always decreasing for $m \leq k \leq M$. Hence $k_2 = \underset{m \leq k \leq M}{\arg\min}\, c(k) = M$.
Therefore in this case we have :

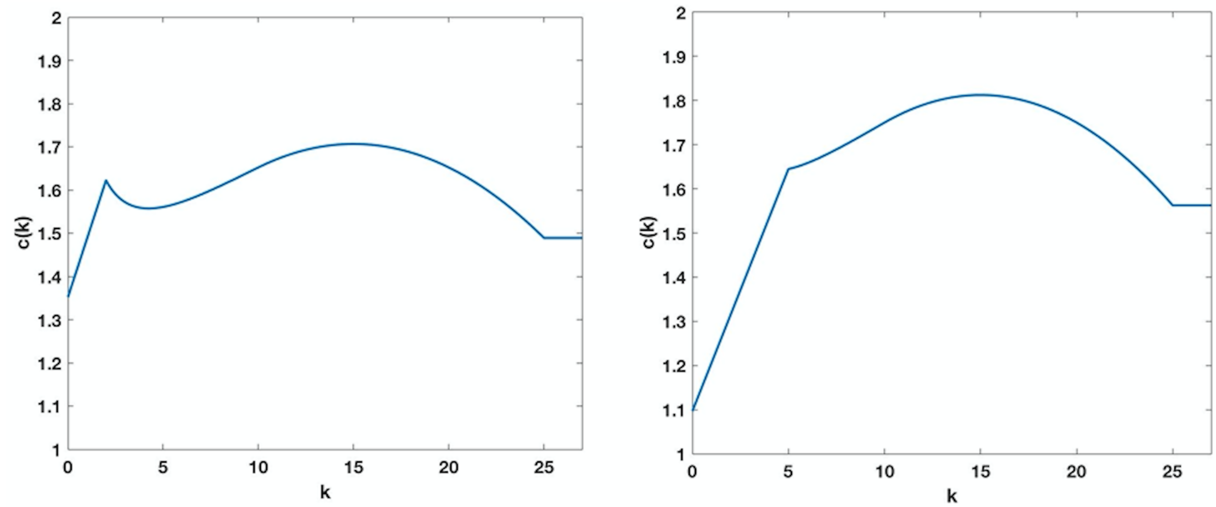
$$k^* = \underset{0 \leq k \leq M}{\arg\min}\, c(k) = \underset{0 \leq k \leq M}{\arg\min}\, \{c(0), c(M)\} \tag{19}$$

Which means that in the case when $M \leq 2s$ we will rent untill the end or we buy at the beginning.

Case 2 : $M > 2s$
$c_1(k)$ $[m \leq k \leq s\ ]$: Here $\ddot{c_1}(k) = \frac{s-k}{k^2(M-m)}$ is always non-negative. Hence, $c_1(k)$ is a convex function and since $\dot{c_1}(k)_{k=s} = \frac{[M-2s]}{s(M-m)}$ is positive, there might be a minimum peak in this region. To find the minimum peak we should solve The equation $\dot{c_1}(k) = 0$. This equation may or may not have a solution for $m \leq k < s$. If it has a solution, then there is a minimum peak in this region and if it does not have a solution in this region then $c_1(k)$ is monotonically increasing for $m \leq k \leq s$.
To compute $\tilde{k}$ such that $\dot{c_1}(k)_{k=\tilde{k}} = 0$ we need to solve the following equation

$$\ln(\frac{s}{\tilde{k}}) - \frac{s}{\tilde{k}} = 1 - \frac{M}{s} \tag{20}$$
$$s.t. \quad m \leq \tilde{k} < s$$

One can see that the value of $\tilde{k}$ does not depends on m but after computing we need to compare it with m to see if $\tilde{k} < m$ or not. Figure 2 illustrates this case. In this figure $s = 10$,



(a) m=2, M=25,s=10

(b) m=5, M=25,s=10

Figure 2: Case2($M > 2s$)

and $M = 25$, Here we want to find the minimum peak for $c_1(k)$. Therefore we need to solve the following equation.

$$\ln(\frac{10}{\tilde{k}}) - \frac{10}{\tilde{k}} = 1 - \frac{25}{10} \tag{21}$$

By solving this equation we have $\tilde{k} = 4.25$. If $m \leq 4.25$, we have a minimum peak at $\tilde{k} = 4.25$ (figure 2(a) illustrates this case). If $m > 4.25$ then $c_1(k)$ is monotonically increasing (figure 2(b) illustrates this case).

$c_2(k)$ $[s < k \leq M]$: Here $\ddot{c_2}(k) = -\frac{1}{s(M-m)}$ is always negative. Hence, $c_2(k)$ is a concave function and since $\dot{c_2}(k)_{k=s} = \frac{[M-2s]}{s(M-m)}$ is positive, and $\dot{c_2}(k) = 0$ has a unique solution in $k = M - s$, there is a maximum peak in $k = M - s$. As one can see in figure 2, there is always a maximum peak in $k = 15$.

$\dot{c_2}(k)_{k=s} = \frac{[M-2s]}{s(M-m)}$

To summarize the results for the case when $M > 2s$: in this case $c_1(k)$ is a convex function that might have a minimum peak at $\tilde{k}$ (figure 2(a)), and if it does not have a minimum, then it is monotonically increasing (figure 2(b)). But $c_2(k)$ is a concave function which has a maximum peak at $k = M - s$. Therefore for these two situation we have :

a) $c_1(k)$ has a minimum peak at $\tilde{k}$:
$k_2 = \underset{m \leq k \leq M}{\arg\min}\, c(k) = \underset{m \leq k \leq M}{\arg\min}\, \{c(\tilde{k}), c(M)\}$
and

$$k^* = \underset{k}{\arg\min} \quad \{c(0), c(\tilde{k}), c(M)\} \tag{22}$$

b) $c_1(k)$ does not have a minimum peak :

$$k_2 = \arg\min_{m \le k \le M} c(k) = \arg\min_{m \le k \le M} \{c(m), c(M)\}$$

and

$$k^* = \arg\min_k \quad \{c(0), c(m), c(M)\} = \arg\min_k \quad \{c(0), c(M)\} \tag{23}$$

The following table summarizes the results for different cases.

case	$k_1 = \arg\min_{0 \le k < m} c(k)$	$k_2 = \arg\min_{m \le k \le M} c(k)$	$\arg\min_k \{c(k_1), c(k_2)\}$
case 1 ($M < 2s$)	0	M	$\arg\min_k \{c(0), c(M)\}$
case 2 ($M \ge 2s$) without $\tilde{k}$	0	$\arg\min_k \{c(m), c(M)\}$	$\arg\min_k \{c(0), c(M)\}$
case 2 ($M \ge 2s$) with $\tilde{k}$	0	$\arg\min_k \{c(\tilde{k}), c(M)\}$	$\arg\min_k \{c(0), c(\tilde{k}), c(M)\}$

4 Average-cost Ratio

In this section we will do the same analysis for the average-cost ratio which is defined as the ratio of the expected cost of the online algorithm over the expected cost of the optimal offline algorithm:

$$c(k) = \overline{CR}(A(k)) \triangleq \frac{E[Cost(A(k), t)]}{E[Cost(OPT, t)]} = \frac{\int_0^\infty Cost(A(k), t) f(t)\, dt}{\int_0^\infty Cost(OPT, t) f(t)\, dt}\,. \tag{24}$$

Our goal is to find the best algorithm for the ski rental problem with the minimum average-cost ratio. The average cost of the optimal offline algorithm is as follows:

$$E[Cost(OPT, t)] = \frac{1}{M-m}[-\frac{1}{2}(s^2 + m^2) + Ms]. \tag{25}$$

Which has a positive value because $\frac{1}{M-m}$ is always positive and

$$-\frac{1}{2}(s^2 + m^2) + Ms = -\frac{1}{2}(s^2 + m^2) + Ms + s^2 - s^2 = \frac{1}{2}(s^2 - m^2) + s(M - s) \tag{26}$$

is also always positive.

The average cost of the online algorithm is as follows:

$$E[Cost(A(k), t)] = \frac{1}{M-m}[-\frac{1}{2}(m^2 + k^2) + Mk + Ms - ks]. \tag{27}$$

Since for the average-cost ratio we have:

$$c(k) = \frac{E[Cost(A(k), t)]}{E[Cost(OPT, t)]} = \frac{-\frac{1}{2}(m^2 + k^2) + Mk + Ms - ks}{-\frac{1}{2}(s^2 + m^2) + Ms} \tag{28}$$

$$\dot{c}(k) = \frac{M - s - k}{-\frac{1}{2}(s^2 + m^2) + Ms} \tag{29}$$

$$\ddot{c}(k) = \frac{-1}{-\frac{1}{2}(s^2 + m^2) + Ms} \tag{30}$$

As one can see $\ddot{c}(k)$ is always negative and $\dot{c}(k) = 0$ has an answer at $k = M - s$ which means that we have a maximum peak at this point. Therefore to make a decision we should compare The competitive ration of the following three cases and choose the one with the minimum average-case competitive ratio.
1) buy at the beginning with the average-case competitive ratio equal to $c(0)$.
2) buy at the day number m with the average-case competitive ratio equal to $c(m)$.
3) buy at the end with the average-case competitive ratio equal to $c(M)$.

$$c(0) = \frac{s(M - m)}{-\frac{1}{2}(s^2 + m^2) + Ms} \tag{31}$$

$$c(m) = \frac{Mm + Ms - ms - m^2}{-\frac{1}{2}(s^2 + m^2) + Ms} \tag{32}$$

$$c(M) = \frac{\frac{1}{2}(M^2 - m^2)}{-\frac{1}{2}(s^2 + m^2) + Ms} \tag{33}$$

Here we want to prove that $c(m)$ is always bigger than or equal to $c(0)$.

$$c(m) - c(0) = \frac{m(M - m)}{-\frac{1}{2}(s^2 + m^2) + Ms} \geq 0 \tag{34}$$

As one can see $c(m)$ is always bigger than or equal to $c(0)$ which means that to make a decision we just need to compare $c(M)$ with $c(0)$ and choose the one with the smaller value.

$$c(M) - c(0) = \frac{\frac{1}{2}(M - m)(M + m - 2s)}{-\frac{1}{2}(s^2 + m^2) + Ms} \tag{35}$$

$\frac{1}{2}(M - m)$ is always positive therefore $c(M) - c(0)$ is positive if $M + m - 2s \geq 0$ or $s \leq \frac{M+m}{2}$. Therefore if $s \leq \frac{M+m}{2}$ we should buy at the beginning and if $s > \frac{M+m}{2}$ we should keep renting until the end. This is a reasonable result because the average of the total number of days we go to ski is $\int_0^\infty \tau . f(t) dt = \frac{M+m}{2}$. Then if this is bigger than s we should buy at the beginning and if it is less than s we should keep renting.

Therefore for the first and third intervals of k, we know the minimum and we only need to find the minimum average-case competitive ratio for $m \leq k \leq M$.

References

[1] Fujiwara, H. and Iwama, K. Average-Case Competitive Analyses for Ski-Rental Problems . Algorithmica (2005)