LOWER BOUNDS FOR THE MULTISLOPE SKI-RENTAL PROBLEM

Hiroshi Fujiwara[1]**,** ***Yasuhiro Konno***[1]**,** ***Toshihiro Fujito***[1]

[1] *Toyohashi University of Technology, 1-1 Tenpaku-cho, Toyohashi, 441-8580 Japan*

Keywords: online algorithm, competitive analysis, online optimization, ski-rental problems.

Abstract

The multislope ski-rental problem is an extension of the classical ski-rental problem, where the player has several options of paying both of a per-time fee and an initial fee, in addition to pure renting and buying options. Damaschke gave a lower bound of 3.62 on the competitive ratio for the case where arbitrary number of options can be offered. In this paper we propose a scheme that for the number of options given as an input, provides a lower bound on the competitive ratio, by extending the method of Damaschke. This is the first to establish a lower bound for each of the 5-or-more-option cases, for example, a lower bound of 2.95 for the 5-option case, 3.08 for the 6-option case, and 3.18 for the 7-option case. Moreover, it turns out that our lower bounds for the 3- and 4-option cases respectively coincide with the known upper bounds. We therefore conjecture that our scheme in general derives a matching lower and upper bound.

1 Introduction

In the *ski-rental problem* the player is offered two options for getting his/her ski gear: either to *rent* (all) ski gear by paying a fee each time of skiing, or to *buy* it. Once the player has bought ski gear, it is available for free forever. The objective is to minimize the total cost, under the setting that the player does not know how many times he/she is going skiing in the future. A strategy of such a player is to rent ski gear for the time being and then buy it. The performance of a strategy is measured by the *competitive ratio*. We say the competitive ratio of a strategy to be c if the player according to the strategy is charged at most c times the optimal offline cost, i.e., one with the number of times of skiing known in advance. A matching upper and lower bound on the competitive ratio is known to be 2 [7]. In other words, for any price setting, there exists a strategy with a competitive ratio of 2 and it cannot be improved.

The *multislope ski-rental problem* is an extension of the ski-rental problem [1, 8]. The player here has not only the pure rent and buy options, but also several options of paying *both of some per-time fee and some initial fee*, for example, to rent only a pair of skis and boots after having bought other gear like ski clothes. For this problem, an upper bound of 4 [2] and a lower bound of 3.62 [4] are known.

When it comes to application, however, the result for the multislope ski-rental problem seems less helpful, since each of these bounds becomes tight in the case where sufficiently many options are offered to the player. We should mention here that the multislope ski-rental problem can be seen as *Dynamic Power Management* [6] on a mobile electronic device equipped with multiple *energy-saving states*, such as Sleep, Stand By, and Hibernate states on a Windows laptop computer. The energy-saving states correspond to the options in the multislope ski-rental problem. The objective is to minimize the energy consumption during an idling time with its length unknown. In reality, it seems that the number of energy-saving states on an electronic device is at most ten or so. Our aim is thus to seek a lower bound for such a realistic case.

1.1 Our Contribution

In this paper we reveal lower bounds on the competitive ratio for the multiple ski-rental problem for the case where the number of options is specified. More precisely, we design a scheme that provides a lower bound with the number of options given as an input, extending the method of Damaschke for the general case [4]. See Table 1. This work is the first to show a lower bound for each of the 5-or-more-option cases. We establish a lower bound of 2.95 for the 5-option case, 3.08 for the 6-option case, 3.18 for the 7-option case, and so on.

Besides, it turns out that our lower bounds for the 3- and 4-option cases, 2.47 and 2.75, respectively, exactly coincide with the known upper bounds [5]. That is to say, our scheme achieves a matching lower and upper bound for these cases. We therefore conjecture that also for the 5-or-more-option cases, the output of our scheme is equal to or quite close to the matching lower and upper bound.

1.2 Related Work

The (classical) ski-rental problem was first introduced as an optimization model of snoopy caching by Karlin et al. [7], where a matching lower and upper bound of 2 was derived.

The start point of research on the multislope ski-rental problem can be found in the paper of Irani et al. in the context of Dynamic Power Management [6]. They studied online strategies for switching multiple

Table 1: Lower and upper bounds for the multislope ski-rental problem. Bold font indicates bounds established in this paper. "# states" means how many options are offered to the player.

# states	2	3	4	5	6	7	8	9	10	general
Lower bound	2 [7]	2.47 [5]	2.75 [5]	**2.95**	**3.08**	**3.18**	**3.25**	**3.31**	**3.36**	3.62 [4]
Upper bound	2 [7]	2.47 [5]	2.75 [5]	4 [2]						

energy-saving states. Augustine et al. [1] proposed an algorithm that for a given instance of the multislope ski-rental problem, outputs the best possible strategy and its competitive ratio. The best lower bound so far of 3.62 was established by Damaschke [4], on which our contribution is mainly based. On the other hand, the best upper bound so far of 4 was shown by Bejerano et al. [2]. These two works preceded introduction of the multislope ski-rental problem, since they are a corollary from the results on the *online investment problem* which can be regarded as a special case of the multislope ski-rental problem, in which the player is obliged to buy new gear every time changing options. Fujiwara et al. [5] gave a matching lower and upper bound for each of the 3- and 4-option cases: 2.47 and 2.75, respectively.

To establish a lower bound is, in other words, to reveal the hardest instance for the player. For the multislope ski-rental problem the easiest instance had also been non-trivial. Fujiwara et al. [5] showed the easiest instance and the best possible competitive ratio for it.

1.3 Note on Rounding

Throughout this paper numerical rounding is all done to the nearest value. It would be a conventional manner that a lower bound is rounded down while an upper bound is rounded up. The reason why we nevertheless insist on rounding to the nearest is because there appear some matching lower and upper bounds, such as those in Table 1. Refer to Table 2 for the values with more precision.

2 Methods

In this section we give a detailed formulation of the multislope ski-rental problem.

2.1 Instance

An instance of the $(k+1)$-*slope ski-rental problem* consists of $k+1$ *states*, each of which stands for a way to get the player's ski gear. We collectively refer to the $(k+1)$-slope ski-rental problem for all k as the *multislope ski-rental problem.* Although we have called a state an *option* in Section 1 for ease of understanding, we will hereafter use the word "state". This is referred to as a *slope* in some literature. State 0 and state k are *to rent* and *to buy*, respectively. States $1, \ldots, k-1$ correspond to the options that the player pays *both of some per-time fee and some initial fee*, for example, to rent only a pair of skis and boots after having bought other gear such as ski clothes. Let r_i and $b_{i,j} (\geq 0)$ denote the per-time fee of state i and the initial fee for transitioning from state i to j, respectively. In this paper we impose the following natural constraints:

$$r_0 = 1, r_k = 0, b_{0,k} = 1 \tag{1}$$

$$r_i > r_j \text{ for } 0 \leq i < j \leq k, \tag{2}$$

$$b_{l,j} - b_{l,i} \leq b_{i,j} \leq b_{l,j} \text{ for } 0 \leq l < i < j \leq k. \tag{3}$$

(1) normalizes so that per-time and initial fees are all scaled down to between zero and one. This normalization may look somewhat strange, but it makes sense; the number of times of skiing will also be scaled soon. (2) says that the states are ordered so that the per-time fee decreases. The left inequality in (3) is a constraint that a direct transition from state l to j is equal to or cheaper than one shortly stopping another state i. The right inequality in (3) says that a transition from state i to j is no cheaper than one from state $l < i$. An instance is thus represented by a pair of such vectors $(\boldsymbol{r}, \boldsymbol{b})$.

For example, a store may offer the following options: Rent everything for \$50 per day (state 0), buy everything for \$500 (state 2), or rent just skis and boots for \$30 per day with buying other gear for \$100 (state 1). Also, the store may allow you to change state 1 to 2 by charging \$450. Then, we may formulate as $(r_0, r_1, r_2) = (50/50, 30/50, 0) = (1, 0.6, 0)$ and $(b_{0,1}, b_{0,2}, b_{1,2}) = (100/500, 500/500, 450/500) = (0.2, 1, 0.9)$.

2.2 Strategy and Competitiveness

Without loss of generality, we assume the number of times of skiing is a real number $t \geq 0$. We sometimes identify t with the time during which the player repeatedly goes skiing. At each time instant, the player at state i, who is paying r_i per time unit, either (a) transitions to a different state j by paying $b_{i,j}$ or (b) keeps staying at state i. In this setting a *strategy* of the player is described by a vector $\boldsymbol{x}$ with $k+1$ entries. Each entry x_i indicates the time when the player transitions to state i from state $i-1$. The sequence of the entries is assumed to be non-decreasing, since we consider only such instances for which the player cannot save cost by backward transition, due to the constraint (2). Since state 0 incurs no initial fee, we can assume that $x_0 = 0$, i.e., the player always starts from state 0. The player may transition from state i directly to j by skipping the states between. For such a transition, we set $x_{i+1} = \cdots = x_{j-1} = x_j$ and define a relation of $i \prec j$. By time t such that $x_i \le t < x_{i+1}$, the player according to strategy $\boldsymbol{x}$ will have paid a cost of

$$ON(\boldsymbol{x}, t) := r_i\,(t - x_i) + \sum_{l=0}^{i-1} r_l\,(x_{l+1} - x_l) + \sum_{0 \le l \prec m \le i} b_{l,m}.$$

The first and second terms are the sum of per-time fees for the states chosen so far, and the third is the sum of initial fees. On the other hand, the *optimal offline player* behaves optimally with t known. Due to the constraint (3), the optimal offline player will choose the best state for him/her at the beginning and then keep staying there. Therefore, the cost incurred by time t is written as

$$OPT(t) := \min_{0 \le j \le k} (r_j t + b_{0,j}).$$

We measure the performance of a strategy using the *competitive ratio*, which is a standard measure in online optimization [3]. We say that strategy $\boldsymbol{x}$ has a competitive ratio of c, if

$$ON(\boldsymbol{x}, t) \le c \cdot OPT(t)$$

holds for all $t \ge 0$.

2.3 Investment Instance

In the rest of this paper we consider the subset of instances that satisfies an additional constraint

$$b_{i,j} = b_{0,j} \text{ for } 0 < i < j \le k,$$

which means in reality that when the player transitions to a state, his/her own gear cannot be reused and therefore he/she is obliged to buy new gear from scratch. In [5] such an instance is referred to as an *investment instance*, which corresponds to *online capital investment* [4]. Note that in the above numerical example, we can have an investment instance if we modify $b_{1,2}$ into one.

In the following two sections we will deal with only such instances and therefore write $b_{i,j}$ simply as b_j. It is demonstrated how to determine $\boldsymbol{b} = (b_1, \ldots, b_k)$ so that a lower bound is established. We add also that there will no longer appear any explicit strategy $\boldsymbol{x}$; our discussion is based on a known lemma which bounds the competitive ratio of arbitrary strategies.

3 General Lower Bound

Damaschke derived a lower bound of $\frac{5+\sqrt{5}}{2} \approx 3.62$ for the general case, that is to say, for the case where arbitrary number of states can be offered [4]. We here review the result while decomposing the derivation in order to extend it to a lower bound for the case where the number of options is specified. We first extract the construction of $\boldsymbol{b}$ with some modification. Although Damaschke constructed $\boldsymbol{r}$ and $\boldsymbol{b}$ through a single procedure, we here discuss them separately.

Procedure 1. *(for determining $\boldsymbol{b}$ [4]) Given $c \ge 2$ as a parameter, starting with*

$$q_1(c) = \frac{c}{c-1}, \tag{4}$$

determine sequence $\{q\}$ according to

$$q_{i+1}(c) = \frac{c^3 - c^2 + c q_i(c)}{c^3 - c^2 - (c-1)^2 q_i(c)}. \tag{5}$$

Let $m(c)$ be the largest integer such that for all $1 \le i \le m(c)$, $q_i(c) \le c$ holds. Next, starting with $b_{m(c)+1} = 1$, determine sequence $\{b\}$ in a reverse order by

$$b_i = \frac{b_{i+1}}{c - \frac{c-1}{c} q_i}$$

for $1 \le i \le m(c)$. Case (i): if $q_{m(c)}(c)$ is exactly equal to c, output $\boldsymbol{b} = (b_1, \ldots, b_{m(c)})$. (Note that due to (5), $b_{m(c)} = b_{m(c)+1}$ holds.) Case (ii): otherwise, i.e., if $q_{m(c)}(c) < c$, output $\boldsymbol{b} = (b_1, \ldots, b_{m(c)+1})$.

It is observed that $q_{i+1}(c)$ is defined for every $q_i(c)$ with $1 \le i \le m(c)$, since the denominator of the right hand side of (5) is, noting $q_i(c) \le c$,

$$\begin{aligned} c^3 - c^2 - (c-1)^2 q_i(c) &= (c-1)^2 (c + 1 + \frac{1}{c-1} - q_i(c)) \\ &\ge (c-1)^2 (1 + \frac{1}{c-1}) \\ &> 0. \end{aligned}$$

With some proper $\boldsymbol{r}$, an $(m(c)+1)$-state instance is generated from Case (i), and an $(m(c)+2)$-state instance from Case (ii). In the later discussion, an instance from Case (i) plays a significant role. We here give a numerical example of $c = 2.5$. We first have $q_1 = 1.66667$, $q_2 = 2.40741$. Since the next term $q_3 = 3.88889$ is bigger than 2.5, we know $m(c) = 2$. Then, corresponding to Case (ii), $\boldsymbol{b}$ is determined as $(b_1, b_2, b_3) = (0.631579, 0.947368, 1)$.

In the description of Procedure 1, differently from the original construction, values in $\boldsymbol{b}$ are scaled down between zero and one, which does not affect the argument of competitiveness. We have also clarified the way of stopping the generation of sequence $\{q\}$.

Lemma 1. *([4]) For $c \ge 2$, determine $\boldsymbol{b}$ according to Procedure 1. Then, for all $\varepsilon > 0$, there exists $\boldsymbol{r}$ such that any strategy for instance $(\boldsymbol{r}, \boldsymbol{b})$ has a competitive ratio of at least $\min(c - \varepsilon, q_{m(c)})$.*

The purpose of Procedure 1 is to obtain this lemma. According to the original description [4], state $i+1$ is inductively constructed from state i and state $i-1$ so that if a strategy does not state at state i, then the competitive ratio becomes above $c - \varepsilon$. Consequently, any strategy with one or more skips between states has a competitive ratio of at least $c - \varepsilon$, while any strategy with no skip has a competitive ratio of at least $q_{m(c)}$. $r_1, r_2, \ldots, r_{k-1}$ are chosen sufficiently small so that they form a decreasing sequence. Applying Lemma 1 to the above example, we know that any strategy for that instance has a competitive ratio larger than 2.40741.

The paper [4] then states that for any $c = \frac{5+\sqrt{5}}{2} - \delta$ with $\delta > 0$, there exists i such that $q_i(c) > c$, which leads to the conclusion below.

Theorem 1. *([4]) Any strategy for the multislope ski-rental problem has a competitive ratio of at least* $\frac{5+\sqrt{5}}{2}$.

4 Results

In this section we obtain a lower bound for the case where the number of states is explicitly specified. The result in Section 3 involves an instance with sufficiently many states; $m(c)$ grows as c approaches $\frac{5+\sqrt{5}}{2}$. Our aim is to construct a $(k+1)$-state instance so that $q_k(c)$ is equal to c, exploiting Procedure 1 and Lemma 1.

A $(k+1)$-state instance can simulate an instance with k or fewer states by making redundant states. Besides, it is known that a lower bound for 2-state (i.e., classical) ski-rental problem is 2 [7]. Therefore, the range of candidates for c is between 2 and $\frac{5+\sqrt{5}}{2}$.

One should take good care in handling sequence $\{q\}$ appearing in Procedure 1. If one does not terminate the generation as we do there, the sequence is in general not either increasing or decreasing, and may even include a negative term. Although the general term can be obtained in a closed form with the generating function method, we do not carry it out here.

Our scheme is simply to solve formally $c = q_k(c)$, given from (4) and (5), and to take its largest root which lie between 2 and $\frac{5+\sqrt{5}}{2}$. A series of arguments below will prove that the scheme in fact yields a lower bound for the $(k+1)$-slope ski-rental problem. We begin with analyzing the behavior of each $q_i(c)$ as a function of c.

Lemma 2. *Run Procedure 1 with c_0 such that $2 < c_0 < \frac{5+\sqrt{5}}{2}$. For $1 \le i \le m(c_0)+1$, see each $q_i(c)$ as a function $q_i : c \mapsto q_i(c)$. Then, q_i is continuous and monotonically decreasing on $[c_0, \frac{5+\sqrt{5}}{2})$.*

Proof. We prove the lemma by induction. In addition to the statement of the lemma, we show the differentiability on $[c_0, \frac{5+\sqrt{5}}{2})$ as well. It is easy to confirm that $q_1(c) = \frac{c}{c-1}$ is continuous, differentiable, and monotonically decreasing on $[c_0, \frac{5+\sqrt{5}}{2})$.

Suppose that for some i with $1 \le i \le m(c_0)$, q_i is continuous and differentiable, and $\frac{dq_i(c)}{dc} < 0$ holds on $[c_0, \frac{5+\sqrt{5}}{2})$. From (5), we immediately know that q_{i+1} is continuous and differentiable on $c \in [c_0, \frac{5+\sqrt{5}}{2})$. Denoting $q_i(c)$ simply as q, we have

$$\begin{aligned}\frac{dq_{i+1}(c)}{dc} =& \frac{1}{(c-1)^2(c^2+q-cq)^2} \cdot \\ & \{-q(c^4 - 2c^3 + 4c^2 - 2c - (c^2-1)q) + \\ & c^2(c-1)(c^2-c+1)\frac{dq}{dc}\}\end{aligned}$$

Since the denominator is

$$\begin{aligned}c^2 + q - cq &= (c-1)(c+1+\frac{1}{c-1} - q) \\ &\ge (c-1)(1+\frac{1}{c-1}) \\ &> 0,\end{aligned}$$

$\frac{dq_{i+1}(c)}{dc}$ is always defined. The proof is done if we show that $\frac{dq}{dc} < 0$ implies $\frac{dq_{i+1}(c)}{dc} < 0$. Let us see the numerator. We easily have for $c \ge 2$,

$$c^2(c-1)(c^2-c+1) > 0.$$

We finally look into the function

$$h(c,q) := c^4 - 2c^3 + 4c^2 - 2c - (c^2-1)q.$$

Apparently, this decreases as q grows. Thus, for $q \le c_0 < \frac{5+\sqrt{5}}{2} < 4$, $h(c,q) > h(c,4) = (c-2)(c^3-2) \ge 0$. Therefore, $\frac{dq_{i+1}(c)}{dc} < 0$. □

Intuitively, Lemma 4 below claims that the number of states can always be increased by choosing larger c. Before that, we give a helper lemma, which is a statement found in [4].

Lemma 3. *([4]) For all $i \ge 1$, $q_i(\frac{5+\sqrt{5}}{2}) < \sqrt{5}$.*

Lemma 4. *The following two statements hold true. (A) For all $2 < c \le c' < \frac{5+\sqrt{5}}{2}$, $m(c') \ge m(c)$. (B) For all $2 < c < \frac{5+\sqrt{5}}{2}$, there exists c' with $c < c' < \frac{5+\sqrt{5}}{2}$ such that $m(c') = m(c)+1$ and $q_{m(c')}(c') = c'$.*

Proof. (A) By definition of m, $q_i(c) \le c$ holds for all $1 \le i \le m(c)$. Lemma 2 implies that for any $c \le c' < \frac{5+\sqrt{5}}{2}$, $q_i(c') \le q_i(c)$ for each i. Again by definition of m, we have $m(c') \ge m(c)$.

(B) By definition of m, it holds that $q_i(c) \le c$ for all $1 \le i \le m(c)$, and that $q_{m(c)+1}(c) > c$. See the functions $x \mapsto q_{m(c)+1}(x)$ and $x \mapsto x$. Lemma 3 guarantees $q_{m(c)+1}(\frac{5+\sqrt{5}}{2}) < \sqrt{5}$ $(< \frac{5+\sqrt{5}}{2})$. Lemma 2 says that $x \mapsto q_{m(c)+1}(x)$ is continuous and monotonically decreasing on $x \in (c, \frac{5+\sqrt{5}}{2})$. Therefore, the two functions must have an intersection point $c' \in (c, \frac{5+\sqrt{5}}{2})$. The c' satisfies the desired properties. □

The next lemma is the heart of our scheme.

Table 2: Our lower bound and the equation that has a root of the value. Each equation is displayed in a reduced form by eliminating irrelevant factors.

# states	Lower bound	Equation
3	2.46557	$c^3 - 4c^2 + 5c - 3 = 0$
4	2.75488	$c^3 - 5c^2 + 8c - 5 = 0$
5	2.94789	$c^5 - 7c^4 + 19c^3 - 27c^2 + 21c - 8 = 0$
6	3.08302	$c^5 - 8c^4 + 25c^3 - 40c^2 + 34c - 13 = 0$
7	3.18123	$c^7 - 10c^6 + 42c^5 - 99c^4 + 145c^3 - 135c^2 + 76c - 21 = 0$
8	3.25479	$c^7 - 11c^6 + 51c^5 - 132c^4 + 210c^3 - 209c^2 + 123c - 34 = 0$
9	3.31128	$c^9 - 13c^8 + 74c^7 - 246c^6 + 534c^5 - 795c^4 + 822c^3 - 577c^2 + 254c - 55 = 0$
10	3.35558	$c^9 - 14c^8 + 86c^7 - 308c^6 + 717c^5 - 1137c^4 + 1241c^3 - 909c^2 + 411c - 89 = 0$

Table 3: $\boldsymbol{b}$ of the instance that achieves a lower bound.

# states	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9
3	0.682328	1							
4	0.398445	0.699223	1						
5	0.216735	0.422177	0.711088	1					
6	0.113682	0.236803	0.43869	0.719345	1				
7	0.0584374	0.127465	0.25124	0.450502	0.725251	1			
8	0.0296859	0.0669354	0.137759	0.261927	0.459209	0.729605	1		
9	0.0149719	0.0346041	0.0735019	0.145631	0.270046	0.465803	0.732901	1	
10	0.00751682	0.0177065	0.0385179	0.0786695	0.151777	0.276354	0.470914	0.735457	1

Lemma 5. *For given $k \geq 2$, solve equation $q_k(c) = c$ formally. Let $\bar{c}$ be the maximum of its real roots which lie between 2 and $\frac{5+\sqrt{5}}{2}$. Given $\bar{c}$, Procedure 1 outputs $\boldsymbol{b}$ with k entries.*

Proof. Let M be the set of real roots of equation $q_k(c) = c$ which are in $(2, \frac{5+\sqrt{5}}{2})$.

(I) First we prove that there exists $c \in M$ such that $m(c) = k$. Let us generate sequence $\{q\}$ with some small c, say $c_1 = 11/5 = 2.2$. We have $q_1(c_1) = 11/6 < 11/5$ and then $q_2(c_1) = 671/216 > 11/5$. Hence, $m(c_1) = 1$. Repeatedly applying (B) of Lemma 4 to this for $k-1$ times, we find c such that $m(c) = k$ and $q_{m(c)}(c) = q_k(c) = c$. The found c surely belongs to M.

(II) Next we claim that $m(c) \leq k$ for all $c \in M$. By definition of M, any $c \in M$ satisfies $q_k(c) = c$. Please note that some $q_k(c)$ may be beyond the last element of sequence $\{q\}$ defined by Procedure 1. Anyway, we derive $q_k(c) = \frac{c^3-c^2+c^2}{c^3-c^2-(c-1)^2c} = c + 1 + \frac{1}{c-1} > c$, which implies that $m(c) \leq k$ by definition of m.

(III) By (A) of Lemma 4 and the above (II), it follows that for all $c \in M$, $m(c) \leq m(\bar{c}) \leq k$. Together with (I), we conclude $m(\bar{c}) = k$. Since $q_k(\bar{c}) = \bar{c}$, Case (i) of Procedure 1 applies. Therefore, Procedure 1 outputs $\boldsymbol{b}$ with k entries. □

Our main theorem follows immediately from Lemmas 1 and 5.

Theorem 2. *Any strategy for the $(k+1)$-slope ski-rental problem has a competitive ratio of at least the maximum of real roots of equation $q_k(c) = c$ which lie between 2 and $\frac{5+\sqrt{5}}{2}$.*

Table 2 shows our lower bound for each $2 \leq k \leq 9$ and the equation that has a root of the value. The reason why only up to the 10-state case is given here is simply because of space limitation. For any k, if one derives $q_k(c)$ by (4) and (5), and solves $q_k(c) = c$ numerically by the Newton method for example, then one can have a lower bound with desired precision. See Table 3 for numerical values of $\boldsymbol{b}$.

For each of the 3- and 4-state cases, our lower bound exactly coincides with the matching lower and upper bounds established in [5], in the sense that for each case the bounds are derived from the same equation. We conjecture that our scheme derives a matching lower and upper bound for the case where the number of states is arbitrarily fixed.

5 Conclusion

For the case where arbitrary number of states are allowed, there remains a gap between the lower and upper bounds of 3.62 and 4. Damaschke conjectured that a matching lower and upper bound of 4 can be achieved. Our conjecture is, in contrast, that the matching bound is 3.62, namely, the upper bound can be improved. We have already observed that our scheme which is derived from Damaschke's method obtains a matching lower and upper bound at least for each of the 3- and 4-state cases.

Our scheme involves recursive derivation of $q_k(c)$. A future work is then to design a simpler scheme, or hopefully to express the lower bound explicitly using k. Another interesting work is to improve the upper bound of 4.

Acknowledgments

This work was supported by KAKENHI (23700014 and 23500014).

References

[1] J. Augustine, S. Irani, and C. Swamy. Optimal power-down strategies. *SIAM J. Comput.*, 37(5):1499–1516, 2008.

[2] Y. Bejerano, I. Cidon, and J. Naor. Dynamic session management for static and mobile users: a competitive on-line algorithmic approach. In *Proc. DIAL-M '00*, pages 65–74, 2000.

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

[4] P. Damaschke. Nearly optimal strategies for special cases of on-line capital investment. *Theor. Comput. Sci.*, 302(1-3):35–44, 2003.

[5] Hiroshi Fujiwara, Takuma Kitano, and Toshihiro Fujito. On the best possible competitive ratio for multislope ski rental. In *Proc. ISAAC '11*, volume 7074 of *LNCS*, pages 544–553. Springer, 2011.

[6] S. Irani, S. Shukla, and R. Gupta. Online strategies for dynamic power management in systems with multiple power-saving states. *ACM Trans. Embed. Comput. Syst.*, 2(3):325–346, 2003.

[7] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator. Competitive snoopy caching. *Algorithmica*, 3:77–119, 1988.

[8] Z. Lotker, B. Patt-Shamir, and D. Rawitz. Rent, lease or buy: Randomized algorithms for multislope ski rental. In *Proc. STACS '08*, pages 503–514, 2008.