# スキーレンタル問題について

藤原 洋志*

## 概 要

スキーレンタル問題は、オンライン最適化問題を解説するための代表例として用いられるほど至って単純な問題である。オリジナルの「レンタル」と「購入」の 2 状態を拡張し状態数を複数にした問題は、理論面だけでなく、実用面においても省電力制御等の幅広い応用を持つ興味深い研究対象である。本稿では多状態スキーレンタル問題とその既存研究を紹介し、さらに拡張した無限状態スキーレンタル問題に対する我々の研究について述べる。

## 1 はじめに

最近建物の廊下やトイレなどで、人がやってくると点灯して、人がいなくなると**しばらく待ってから消灯**するように設定された蛍光灯をよく見かける。そのような蛍光灯の消費電力量をオンライン最適化してみよう。具体的には人がいなくなってからどのくらい待って消灯するのが最適か考えてみる。そもそも、なぜ即座に消灯しないかといえば、蛍光灯は点灯に比較的多くの電力量を必要とするからである。(実は最近の器具では点灯のための電力量より、器具の寿命短縮の方がはるかに重要らしいのだが、ここでは目をつぶって「点灯に多くの電力量が必要である」ことを仮定してほしい。) 人がいなくなってから次に誰かがやってくるまでの、いわゆる 1 回のアイドル期間のみに注目しよう。すると実はこの問題は古典的スキーレンタル問題 [KMRS88] を連続量にした問題と等価であることが分かる。既存結果から、競合比 [BE98] を基準に最適化すると、(点灯に要する電力量)/(単位時間当たり消費電力) の時間だけ待って消灯するのが最適であると分かる。

| | オン (状態 0) | スタンバイ (状態 1) | 休止状態 (状態 2) | オフ (状態 3) |
|---|---|---|---|---|
| $r_i$ | 1 | 0.5 | 0.2 | 0 |
| $b_i$ | 0 | 0.5 | 0.6 | 1 |

表 1: 多状態スキーレンタル問題の状態例。$r_i$ は状態 $i$ の単位時間当たり消費電力、$b_i$ は再びオンにするのに要する電力量。各数値は説明のための設定。

古典的スキーレンタル問題では、プレーヤは「レンタルし続ける」または「購入してそれを使う」という 2 つのオプション (**状態**と呼ぶ) を取り得たが、その間の状態をとることを許した**多状態スキーレンタル問題**により PC の省電力制御問題等を記述することが出来る [ISG02]。Windows には低消費電力モードである「スタンバイ」や「休止状態」等が実装されており (表 1 参照)、例えば、最後の操作から *30* **分操作がなければスタンバイ、さらに** *1* **時間操作がなければ休止状態に遷移**するといったようなアルゴリズムが動いていたりする。この「スタンバイ」等が多状態スキーレンタル問題において「レンタル」と「購入」の間の状態に対応し、スキーレンタルの文脈では、購入するよりも小さな初期費用を払いかつ使用に応じた費用も払う「リース」を意味する。多状態スキーレンタル問題についても競合比を基準に最適化することが可能で、最適アルゴリズムを状態数の多項式時間で計算可能である [AIS08] (会議版は [AIS04])。しかしながらその最適アルゴリズムを陽に表す方法は現在知られていない。

我々はさらに拡張した**無限状態スキーレンタル問題**を考える。状態は無限個存在し、従って状態遷移

*豊橋技術科学大学

は連続的に行われる。我々はまず、多状態スキーレンタル問題に対し競合比 2 を保証するアルゴリズムの無限状態版を与える。さらにそれをもとに、関数をパラメータとして一般のアルゴリズムを表す。我々の研究の目標は、**状態集合を入力として、最適アルゴリズムを閉じた形で出力**する枠組みを構築することである。繰り返しになるが、多状態の問題に対しても最適アルゴリズムは求まるものの、状態集合を用いて陽に表せないのであった (詳しく言えば、それは 2 分探索と動的計画法によるものである)。もし無限状態の問題で解決できれば、多状態の問題の解決への足がかりとなろう。

無限状態スキーレンタル問題についての最適アルゴリズムを考察する際、我々は関数解析を道具として利用する。既に我々は通貨交換問題を取り上げ [FIS09]、この種のオンライン最適化問題に対し関数解析が非常に有効な手段となることを実証した。上で述べたように、無限状態の問題に対する一般のアルゴリズムは関数をパラメータとして表せるので、その関数を最適化することにより最適アルゴリズムを見つけたい。

**関連研究**　古典的スキーレンタル問題やその周辺については [徳山 07] にやさしく解説してある。多状態スキーレンタル問題については様々な捉え方をされており、その結果 Power-Down Strategies や Multislope Ski Rental 等、多様な呼ばれ方をしている。むしろ、この問題が現実問題に幅広く応用可能であることを物語っている。次章以降で議論するのは、状態遷移する際に差分 $(b_i - b_{i-1})$ のコストを払うという加法的モデルである。一方、状態遷移する際に一般の $d_{i,j} > (b_i - b_{i-1})$ のコストを払う非加法的モデルも研究されている [AIS08]。後で紹介する定理 3 は非加法的モデルに対しても成立する。

また本稿では決定性の問題のみ扱うことにするが、プレーヤが乱数を利用できる乱択モデルに関しては次のような結果が知られている。古典的スキーレンタル問題においては競合比 $e/(e-1) \approx 1.58$ が達成でき、これが最適である [KKR03]。論文 [LPSR08] では、多状態スキーレンタル問題で加法的である場合に最適な乱択アルゴリズムを計算するアルゴリズム、及び非加法的多状態スキーレンタル問題についての競合比 $e \approx 2.73$ のアルゴリズムが与えられている。尚これらの結果には全て、敵対者は乱数の分布は知っているがその値を知ることの出来ないという oblivious model[BE98] が仮定されている。

## 2　多状態スキーレンタル問題

古典的スキーレンタル問題 [KMRS88] を拡張した多状態スキーレンタル問題は次のような問題である。記法は [LPSR08] に従う。ただしスキーレンタルの用語ではなく、PC の省電力制御の文脈において説明する。ユーザが最後に PC を操作した時刻を時刻 0 とする。そこから再び操作し始めるまでの時間を**アイドル期間**と呼び $t$ と書く。**状態**とは前述したようなモードであると考えてもらいたい。PC は $(k+1)$ 個の状態のうちいずれかの状態である。状態 0 とは電源オン、状態 $k$ とは電源オフを表し、その間に $(k-1)$ 個の低消費電力モードが存在する。PC が消費する電力量 (以降コストと呼ぶ) のルールは以下の通りである。

1. 状態 $i$ にある時、単位時間ごとに $r_i$ だけコストが必要。
2. 状態 $(i-1)$ から状態 $i$ に遷移する時は、その瞬間、$(b_i - b_{i-1})$ のコストが必要。

ただし、$0 = b_0 < b_1 < \cdots < b_k = 1$、また $1 = r_0 > r_1 > \cdots > r_k = 0$ とする。つまり電力の大きい状態から順に並んでいるとする。先に載せた表 1 は $k = 3$ の場合の状態集合の具体例である。

2 番目のルールについては次のように解釈してもらいたい。$b_i$ とは、状態 $i$ から状態 0 に戻るためのコストを表す。今、再び操作し始めるまでのアイドル時間を考えるので、最終的には必ず状態 0 (電源オン) に戻る必要がある。そのため、1 つ先の状態に進むとそこから状態 0 に戻るには、その差分だけ多くコストがかかる。2 番目のルールは、そのことが確

定した時点でその分を先払いするということを意味している。

尚、これらをスキーレンタルの文脈で説明すると、アイドル期間とはスキーに行く回数を意味する。また状態 0 は「レンタル」、状態 $k$ は「購入」、また一般に、状態 $i$ は初期費用 $b_i$ を払いかつ単位時間毎に $r_i$ のコストを払うという「リース」のようなものを表している。特に $k=1$ とすれば古典的スキーレンタル問題の連続量版となっている。

この問題に対する (決定性) アルゴリズム ALG は、

$$ALG = (T_0, T_1, \cdots, T_k) \tag{1}$$

で表現される ($0 = T_0 \leq T_1 \leq \cdots \leq T_k$)。ここで各 $T_i$ は状態 $(i-1)$ から状態 $i$ に遷移する時刻を表している。時刻 $t$ ($T_i \leq t \leq T_{i+1}$) までにアルゴリズム ALG が払うコストは、

$$\begin{aligned} ALG(t) =& r_i \cdot (t - T_i) + \sum_{j=0}^{i-1} \{r_j \cdot (T_{j+1} - T_j)\} + b_0 \\ & + \sum_{j=0}^{i-1} (b_{j+1} - b_j) \\ =& r_i \cdot (t - T_i) + \sum_{j=0}^{i-1} \{r_j \cdot (T_{j+1} - T_j)\} + b_i \end{aligned} \tag{2}$$

と書ける。アイドル期間が前もって分かっている最適オフラインアルゴリズム OPT は、明らかに、時刻 0 でどの状態が最適か判断し即座にそれに遷移する。従ってそのコストは

$$OPT(t) = \min_{0 \leq i \leq k} (b_i + r_i \cdot t) \tag{3}$$

と表せる。図 1 に表 1 の状態集合に対する $OPT(t)$ を図示する。以降では状態の集合として、どの状態 $i$ ($0 \leq i \leq k$) に対しても $b_i + r_i \cdot t = OPT(t)$ を満たす $t$ が存在するものと仮定する。

アルゴリズムの評価には競合比 (competive ratio)[BE98] を用いる。本問題に対するアルゴリズム ALG の競合比を

$$\max_{t>0} \frac{ALG(t)}{OPT(t)} \tag{4}$$

と定義する。

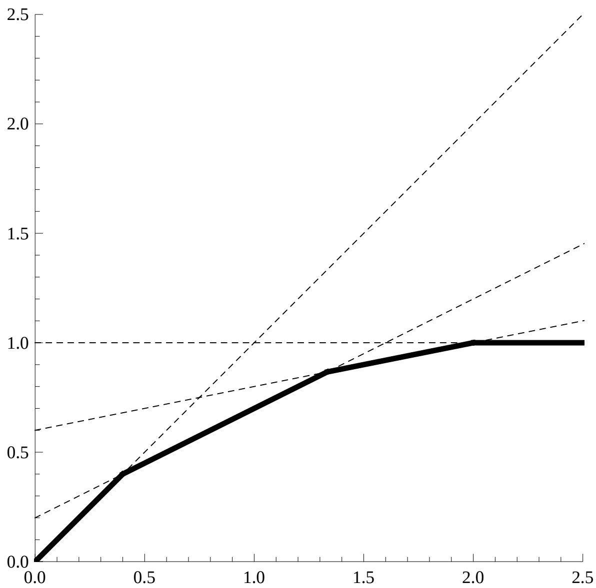


図 1: 横軸はアイドル期間 $t$、縦軸はアルゴリズムが払うコスト。細点線は時刻 0 でそれぞれの状態に遷移した場合。太実線は最適オフラインアルゴリズム OPT。状態集合は表 1 のもの。

## 2.1 アルゴリズム LE

最適オフラインアルゴリズム OPT を別の角度から見ると、アイドル期間 $t$ に対して最適な状態が一意に定まることが分かる。今から紹介するアルゴリズム LE (Lower Envelope)[ISG02] は、直観的には、最適オフラインアルゴリズムを模倣し、常にそれと同じ状態であり続けるというものである。最適オフラインアルゴリズムに従うと状態 $i$ にいるようなアイドル期間とは、$b_{i-1} + r_{i-1} \cdot t$ と $b_i + r_i \cdot t$ との交点、及び $b_i + r_i \cdot t$ と $b_{i+1} + r_{i+1} \cdot t$ との交点の間である (図 1 参照)。従ってアルゴリズム LE は、

$$LE = (0, -\frac{b_1 - b_0}{r_1 - r_0}, -\frac{b_2 - b_1}{r_2 - r_1}, \cdots, -\frac{b_k - b_{k-1}}{r_k - r_{k-1}}) \tag{5}$$

と書ける。各成分にマイナスがついているが値は全て正である。定理 1 にあるようにアルゴリズム LE は競合比 2 を達成する。具体例を示そう。例えば表 1 の状態集合に対するアルゴリズム LE は、

$$LE = (0, \frac{2}{5}, \frac{4}{3}, 2) \tag{6}$$

である。のコストは図 2 に太点線で図示してある。

尚、アルゴリズム LE は古典的スキーレンタル問題において「時刻 1 までレンタルして、時刻 1 で購入」というアルゴリズムに対応している。このケースについてはこれが最適であることが知られている。$k \geq 2$ の場合には必ずしも最適ではない。

**定理 1** ([ISG02]). 一般の $k$ に対し、アルゴリズム $LE$ の競合比 $= 2$。また競合比 $= 2$ となるのは、状態 $k$ に遷移した直後。

**定理 2** ([KMRS88]). $k = 1$ の場合（古典的スキーレンタル問題）、アルゴリズム $LE$ は競合比の評価で最適アルゴリズムである。競合比 $= 2$。

## 2.2 最適アルゴリズム

図 2 からも、アルゴリズム LE が最適ではないことが分かる。実際、(0, 3/10, 1, 2) としたアルゴリズムのコスト (太実線) は、$2 \cdot OPT(t)$ を表す細点線より常に下にあることが分かる。即ちこのアルゴリズムは競合比が 2 を切っている。

論文 [AIS08] は、最適アルゴリズムを求めるための、動的計画法と 2 分探索に基づいたアルゴリズムを与えている。しかし最適アルゴリズムやその競合比は陽に求まらない。次章の無限状態スキーレンタル問題の研究によりその最適アルゴリズムとその競合比が閉じた形で与えられれば、再度ここで検討することにしたい。

次の補題は最適アルゴリズムを求めるための補題であるが、状態数が無限の場合の考察の際にも重要な示唆を与える。

**補題 1** ([AIS08]). 競合比$=\rho$ を達成するアルゴリズムが存在すると仮定する。すると、以下の性質を満たすアルゴリズム $A(T_0, T_1, \cdots, T_k)$ が存在する: 全ての $t \in \{T_0, T_1, \cdots, T_k\}$ に対し、$A(t)/OPT(t) = \rho$。

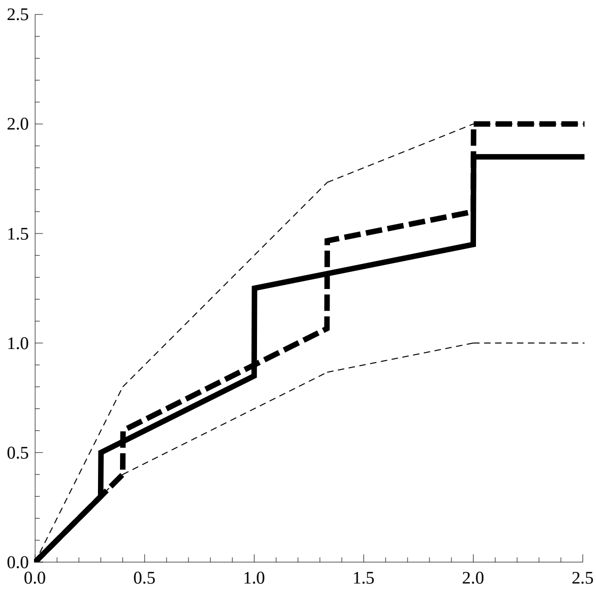


図 2: 横軸はアイドル期間 $t$、縦軸はアルゴリズムが払うコスト。アルゴリズム LE (太点線) と、それを改良したアルゴリズムの例 (太実線)。細点線は $OPT(t)$ と $2 \cdot OPT(t)$。状態集合は表 1 のもの。

**定理 3** ([AIS08]). 最適な競合比が $\rho$ であるとき、競合比 $(\rho + \epsilon)$ のアルゴリズムを $O(k^2 \log k \log(1/\epsilon))$ 時間で計算可能である。

# 3 無限状態スキーレンタル問題

これまでの議論と逆の発想で、状態数が無限のスキーレンタル問題を定式化する。議論は最適オフラインアルゴリズムから始まる。要するに、最適オフラインアルゴリズムのコストをまず与え、それを包絡線として持つような関数族を状態集合とする。定数 $T > 0$ を定める。関数 $w \in C^2[0, T]$ で $w''(t) < 0$ $(t \in [0, T])$、$w'(0) > 0$、及び $w'(T) = 0$ を満たす

ものを考える。曲線 $y = w(t)$ は、接線族

$$\{y = w'(\tau)(t-\tau) + w(\tau) : \tau \in [0,T]\} \quad (7)$$

を持ち、逆に、この接線族は包絡線として $y = w(t)$ を持つ。無限状態スキーレンタル問題ではこの接線族を、$\tau$ をパラメータとする状態集合とみなす。即ち、状態 $\tau$ について、単位時間当たりコスト $= w'(\tau)$、状態 0 からの状態遷移コスト $= -\tau \cdot w'(\tau) + w(\tau)$ である。明らかに $OPT(t) = w(t)$ である。

## 3.1 再びアルゴリズム LE

多状態スキーレンタル問題と同様にアルゴリズム LE を構成しよう。時刻が $t = x$ のとき、アルゴリズム LE は、アイドル期間が $x$ であるとあらかじめ知っているオフライン最適アルゴリズムと同じ状態、つまり状態 $x$ にあると考える。微小時間 $\Delta x$ が経過したときに、アルゴリズム LE が払うコストを考える。ルールに従い、コストの増分は以下の 2 つの和である。状態数が無限なので、常に状態遷移をしていることに注意する。

1. 単位時間毎のコスト: $w'(x) \cdot \Delta x$
2. 状態遷移のコスト: $\eta$

$\eta$ は以下のようにして求められる。$\eta$ は接線 $l : y = w'(x)(t-x) + w(x)$ と接線 $l' : y = w'(x+\Delta x)(t - x + \Delta x) + w(x+\Delta x)$ の $y$ 切片の差であるから (多状態の問題における $b_i - b_{i-1}$ のことである)、

$$\begin{aligned}\eta &= \{w(x) - xw'(x)\} \\ &\quad - \{w(x+\Delta x) - (x+\Delta x)w'(x+\Delta x)\} \\ &\approx -xw''(x)\Delta x \end{aligned} \quad (8)$$

よって、

$$LE(x+\Delta x) - LE(x) = (w'(x) - xw''(x))\,\Delta x \quad (9)$$

$\Delta x \to 0$ として、

$$\frac{d}{dx}LE(x) = w'(x) - xw''(x) \quad (10)$$

これを積分する。

$$\begin{aligned} LE(t) &= \int_0^t (w'(x) - xw''(x))\,dx \\ &= w(t) - \left\{[xw'(x)]_0^t - \int_0^t w'(x)dx\right\} \\ &= w(t) - \{tw'(t) - w(t)\} \\ &= 2w(t) - tw'(t) \end{aligned} \quad (11)$$

例えば、$OPT(t) = w(t) = -(t-1)^2 + 1$ に対しては、$LE(t) = 2t$ となる (図 3 参照)。

競合比について考える。

$$\begin{aligned} \frac{LE(t)}{OPT(t)} &= \frac{2w(t) - tw'(t)}{w(t)} \\ &= 2 - \frac{tw'(t)}{w(t)} \end{aligned} \quad (12)$$

は増加関数である。$t = T$ で値は 2。

**定理 4.** 状態数が無限の場合に対し、アルゴリズム $LE$ の競合比 $= 2$。また競合比 $= 2$ となるのは、時刻 $T$。

## 3.2 一般のアルゴリズム

これまでのアルゴリズム LE についての議論を基に、一般のアルゴリズムを表すことを試みる。繰り返しになるが、 アルゴリズム LE は時刻 $t$ で、状態 $t$ にあるのであった。ここに注目して、その対応をずらしてみる。即ち、関数 $s : [0,T] \to [0,T]$ を導入して、時刻 $t$ で、状態 $s(t)$ にあるようなアルゴリズム ALG を考える。ただし時刻 0 で状態 0 にあるため $s(0) = 0$、前の状態には戻らないため $s'(t) \geq 0$ とする。

アルゴリズム LE と同様にしてコストの関数を求める。時刻が $x$ から $x + \Delta x$ になったときに、アルゴリズム ALG が払うコストを考える。

1. 単位時間毎のコスト: $w'(s(x)) \cdot \Delta x$
2. 状態遷移のコスト: $-s(x)s'(x)w''(s(x)) \cdot \Delta x$

よって、$\Delta x \to 0$ として、

$$\frac{d}{dx}ALG(x) = w'(s(x)) - s(x)s'(x)w''(s(x)) \quad (13)$$

これを積分する。

$$ALG(t) = w(s(t)) - s(t)w'(s(t)) + \int_0^t w'(s(x))dx \quad (14)$$

アルゴリズム LE は状態数が多状態の場合に必ずしも最適アルゴリズムではなかったが、無限状態の場合も同様である。実際、先程と同じ $w(t) = -(t-1)^2+1$ に対しては例えば $s(t) = -(t-1)^2+1$ を選ぶと、競合比は 1.6 になる (図 3 参照)。

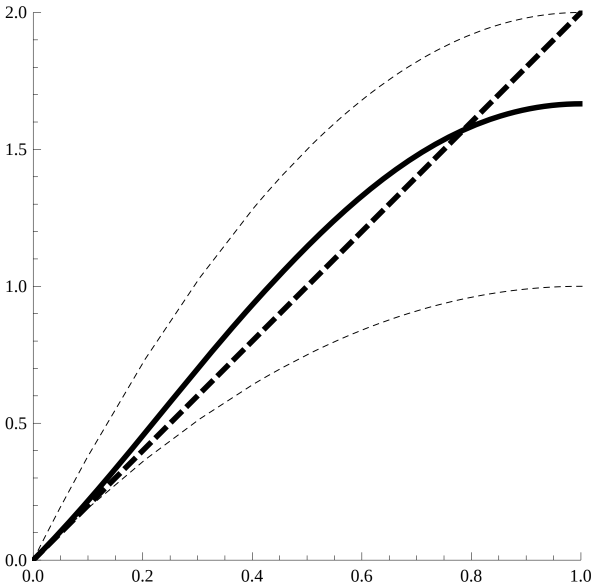


図 3: 横軸はアイドル期間 $t$、縦軸はアルゴリズムが払うコスト。アルゴリズム LE (太点線) と、それを改良したアルゴリズムの例 ($s(t) = -(t-1)^2+1$ とした場合)(太実線)。細点線は $OPT(t) = w(t)$ と $2 \cdot OPT(t)$。ここで $w(t) = -(t-1)^2+1$。

さて本研究の目標は競合比を最小化するアルゴリズムを求めることである。そのための取り組みとしてまず考えられるのが (1) 全ての $t \geq 0$ について $ALG(t) \leq c \cdot OPT(t)$ を満たすような最小の $c$ とそのときの $s$ を求めることである。また、多状態の問題については補題 1 が成立することを思い出してほしい。これと同等なことが無限状態の問題についても成立すると仮定すると、(2) 全ての $t \geq 0$ について $ALG(t) = c \cdot OPT(t)$ を解くことが考えられる。

### 3.3 $ALG(t) \leq c \cdot OPT(t)$ を解く

関数 $s : [0,T] \to [0,T]$ を用いて表されるアルゴリズム ALG のうちで、$ALG(t) \leq c \cdot OPT(t)$ を満たすような最小の $c$ を探す。以下のように定式化する。

$$
\begin{aligned}
(\mathcal{P})\ \ \text{minimize}\ f :=& c \\
\text{subject to}\ g_1 :=& OPT(t) - c\dot{A}LG(t) \\
=& w(s(t)) - s(t)w'(s(t)) \\
&+ \int_0^t w'(s(x))dx - c \cdot w(t) \leq 0, \\
& t \in [0,T] \quad (15) \\
g_2 :=& -s'(t) \leq 0,\ t \in [0,T] \quad (16) \\
h :=& s(0) = 0 \quad (17)
\end{aligned}
$$

ガトー微分 [Lue69] を計算すると、

$$Df(c,s)(\delta,u)(t) = (\delta, 0) \quad (18)$$

$$Dg_1(c,s)(\delta,u)(t) = (-\delta \cdot w(t), -s(t)w''(s(t))u(t) + \int_0^t w''(s(x))u(x)dx) \quad (19)$$

$$Dg_2(c,s)(\delta,u)(t) = (0, -u'(t)) \quad (20)$$

$$Dh(c,s)(\delta,u)(t) = (0, u(0)) \quad (21)$$

KKT 条件 [Lue69] を考える。$Df + \langle \lambda_1, Dg_1 \rangle + \langle \lambda_2, Dg_2 \rangle + \langle \lambda_3, Dh \rangle$ の $c$ 成分は、

$$1 - \int_0^T w(t)\lambda_1(t)dt = 0 \quad (22)$$

$Df + \langle \lambda_1, Dg_1 \rangle + \langle \lambda_2, Dg_2 \rangle + \langle \lambda_3, Dh \rangle$ の $s$ 成分は、

$$\int_0^T \left\{ -s(t)w''(s(t))u(t) + \int_0^t w''(s(x))u(x)dx \right\} \cdot \lambda_1(t)dt + \int_0^T (-u'(t))\, \lambda_2(t)dt + \lambda_3 u(0) \tag{23}$$

また、

$$\langle \lambda_1, g_1 \rangle = \int_0^T \{w(s(t)) - s(t)w'(s(t))u(t) + \int_0^t w'(s(x))u(x)dx - c \cdot w(t)\} \lambda_1(t)dt \tag{24}$$

$$\langle \lambda_2, g_2 \rangle = \int_0^T (-s'(t))\, \lambda_2(t)dt \tag{25}$$

$c$ と $s$ が問題 $(\mathcal{P})$ の局所最適解となる必要条件は、(22),(23),(24),(25) が全てゼロ、かつ、

$$\lambda_1(t) \geq 0,\ \lambda_2(t) \geq 0,\ t \in [0, T] \tag{26}$$

が成立すること。さらに $\lambda_1$ と $\lambda_2$ はそれぞれ、$g_1$ と $g_2$ 属する関数空間の共役空間の元であること。

### 3.4 $ALG(t) = c \cdot OPT(t)$ を解く

常にある競合比を達成するアルゴリズム、つまりある $0 < c < 2$ に対し、

$$ALG(t) = c \cdot OPT(t),\ t \in [0, T] \tag{27}$$

を満たす $s$ が存在するのか考察する。ヒントは、状態が有限の場合に関する補題 1 である。補題 1 を無限状態版に焼き直すと、「競合比 $\rho$ を達成するアルゴリズムが存在すれば、全ての $t \in [0, T]$ において $ALG(t) = \rho \cdot OPT(t)$ を満たすアルゴリズムが存在する」ということである (これが正しいかは不明)。(27) は

$$w(s(t)) - s(t)w'(s(t)) + \int_0^t w'(s(x))dx = c \cdot w(t) \tag{28}$$

であるが、$t$ で微分して、微分方程式

$$w'(s(t)) - s(t)s'(t)w''(s(t)) = c \cdot w'(t) \tag{29}$$

が得られる。

## 4 今後の課題

多状態スキーレンタル問題については、状態集合を絞った場合に最適アルゴリズムや最適競合比が陽に表せないか検討する。また最適競合比を近似計算することにも興味がある。

無限状態スキーレンタル問題についてはほとんど未解決といっていい。3.3 章の問題に関して、$s$ の関数空間を狭めるとどうなるか考える。また数値シミュレーションで解の見当を立てることも検討する。

## 参考文献


[AIS04] J. Augustine, S. Irani, and C. Swamy. Optimal power-down strategies. In *Proc. FOCS '04*, pp. 530–539, 2004.

[AIS08] J. Augustine, S. Irani, and C. Swamy. Optimal power-down strategies. *SIAM J. Comput.*, Vol. 37, No. 5, pp. 1499–1516, 2008.

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

[FIS09] H. Fujiwara, K. Iwama, and Y. Sekiguchi. Average-case competitive analyses for one-way trading. *Journal of Combinatorial Optimization*, 2009.

[ISG02] S. Irani, S. Shukla, and R. Gupta. Competitive analysis of dynamic power management strategies for systems with multiple power saving states. In *Proceedings of the Design Automation* *and Test Europe Conference*, pp. 117–123, 2002.

[KKR03] A. R. Karlin, C. Kenyon, and D. Randall. Dynamic tcp acknowledgment and other stories about e/(e-1). *Algorithmica*, Vol. 36, No. 3, pp. 209–224, 2003.

[KMRS88] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator. Competitive snoopy caching. *Algorithmica*, Vol. 3, pp. 77–119, 1988.

[LPSR08] Z. Lotker, B. Patt-Shamir, and D. Rawitz. Rent, lease or buy: Randomized algorithms for multislope ski rental. In *Proc. STACS '05*, pp. 503–514, Dagstuhl, Germany, 2008. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl.

[Lue69] D. G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons Inc., New York, 1969.

[徳山 07] 徳山豪. オンラインアルゴリズムとストリームアルゴリズム. 共立出版, 2007.