Vol.2015-AL-153 No.8
2015/6/12

オンライン二分探索木の遅延更新に対するリグレット解析

松川 理拓[1,a)]　山内 由紀子[1,b)]　来嶋 秀治[1,c)]　山下 雅史[1,d)]

概要：本研究では，オンライン二分探索木のリグレット解析を行う．Kalai と Vempala は，オンライン意思決定問題の応用として，T 回の探索におけるリグレットの期待値が $o(T)$ となる遅延更新型のアルゴリズムを与えた．このリグレットの達成には，事前に T の値を知る必要がある．本稿では，T の値に関する事前情報を必要としないアルゴリズムを提案する．提案手法は Hannan のアルゴリズムに遅延更新を適用したもので，Kalai と Vempala の解析手法を応用し，提案手法のリグレットの期待値が $o(T)$ であることを示す．

Tadahiro Matsukawa[1,a)]　Yukiko Yamauchi[1,b)]　Shuji Kijima[1,c)]　Masafumi Yamashita[1,d)]

1. はじめに

大規模な計算機ネットワークの普及に伴い，大量データの効率的な管理，運用に注目が集まっている．個人の活動が膨大な量の情報を生み出し，流行する情報が絶え間なく変化する現代においては，必要なデータを高速に探索できることが重要である．

二分探索木を用いることで高速な探索が可能だが，過去の探索クエリ列の傾向をもとに木を再構築することで探索時間をさらに短縮できる可能性がある．そこで本研究では，未知の探索クエリ列に対し，節点を探索するのに必要なコストと木の再構築に必要なコストの和を最小化する問題を考える．Kalai と Vempala は，この問題をオンライン意思決定問題として捉え，T 回の探索におけるリグレットの期待値が $o(T)$ となる遅延更新手法を用いたアルゴリズムを与えた [3]．本稿では，Hannan のアルゴリズム [4] に遅延更新手法を用いることで T の値に関する事前情報を必要としないアルゴリズムを提案する．また，Kalai と Vempala の解析手法を応用することで，提案アルゴリズムのリグレットの期待値が $o(T)$ であることを示す．

本研究で紹介する既存アルゴリズムと提案アルゴリズムの位置付けは，表 1 の通りである．

	T に関する事前情報 必要	T に関する事前情報 不要
遅延更新 なし	FPL Tree	Hannan's Tree
	探索コスト：$o(T)$	探索コスト：$o(T)$
	移動コスト：$O(T)$	移動コスト：$O(T)$
遅延更新 あり	FLL Tree	本研究
	探索コスト：$o(T)$	探索コスト：$o(T)$
	移動コスト：$o(T)$	移動コスト：$o(T)$

表 1　各アルゴリズムの探索コストと移動コスト

2. 準備

2.1 オンライン二分探索木

n 個の節点 $v_1, v_2, \ldots, v_n$ を持つ二分探索木に対し，未知の探索クエリ列を与える．この探索クエリ列の長さを T とし，各時刻 $t (1 \leq t \leq T)$ 毎に節点が一つ探索されるものとする．

ここで，時刻 t において探索された節点を，n 次元ベクトル $\boldsymbol{s}^{(t)} \in \mathcal{S} \subset \mathbb{R}^n$ を用いて表す．ベクトルの各成分は，時刻 t で各節点が探索されたかどうかを $\{0, 1\}$ によって表す．例えば，時刻 t で節点 v_1 が探索されたならば，$\boldsymbol{s}^{(t)} = (1, 0, \ldots, 0)$ である．

また，時刻 t でのオンライン二分探索木の各節点の深さを，n 次元ベクトル $\boldsymbol{d}^{(t)} \in \mathcal{D} \subset \mathbb{R}^n$ を用いて表す．ベクトルの各成分は，時刻 t での各節点の深さを $[1, n]$ で表す．ここで節点の深さとは，(根からその節点までの枝数)+1 である．図 2.1 の各節点の深さを $\boldsymbol{d}$ によって表現すると，$\boldsymbol{d} = (3, 2, 3, 1, 2, 3)$ となる．

---

[1] 九州大学大学院 システム情報科学府
福岡県福岡市西区元岡 744
[a)] tadahiro.matsukawa@inf.kyushu-u.ac.jp
[b)] yamauchi@inf.kyushu-u.ac.jp
[c)] kijima@inf.kyushu-u.ac.jp
[d)] mak@inf.kyushu-u.ac.jp

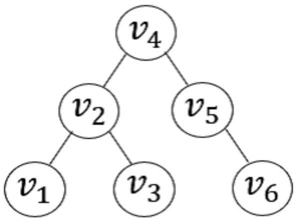

図 1　節点数 6 の二分探索木

2.2　探索コスト

時刻 t における探索コスト (Search Cost) とは，節点を探索するまでに要した比較回数，つまり探索した節点の深さである．時刻 t での探索コストはベクトル $\boldsymbol{s}^{(t)}$ と $\boldsymbol{d}^{(t)}$ の内積で表すことができる．

オンライン二分探索木は，探索を T 回行った時の探索コストの総和を小さくするために，自動的に木構造を更新する．

2.3　移動コスト

オンライン二分探索木が時刻 t での探索の後に木構造を更新するために要するコストを移動コスト (Switching Cost) と定義する．

2.4　最適二分探索木

最適二分探索木（Optimal Binary Search Tree）とは，

$$\sum_{i=1}^{n}(\text{節点 } v_i \text{ の探索回数})\times(\text{節点 } v_i \text{ の深さ})$$

を最小化する二分探索木である [6].

動的計画法を用いることで，節点数が n 個の最適二分探索木を $O(n^3)$ 時間で求めることができる．

ここで，M を最適二分探索木の木構造を計算する関数とし，

$$\begin{aligned} &M(\boldsymbol{s}^{(1)} + \boldsymbol{s}^{(2)} + \cdots + \boldsymbol{s}^{(t)}) \\ &= \arg\min_{\boldsymbol{d}\in\mathcal{D}} \boldsymbol{d}\cdot(\boldsymbol{s}^{(1)} + \boldsymbol{s}^{(2)} + \cdots + \boldsymbol{s}^{(t)}) \end{aligned}$$

と定義する．($M : \mathbb{R}^n \to \mathcal{D}$)

また，T 回の探索を行った後に得られる最適二分探索木の探索コストは，

$$\begin{aligned} &\min_{\boldsymbol{d}\in\mathcal{D}} \sum_{t=1}^{T} \boldsymbol{d}\cdot\boldsymbol{s}^{(t)} \\ &= M(\boldsymbol{s}^{(1)} + \boldsymbol{s}^{(2)} + \cdots + \boldsymbol{s}^{(T)})\cdot(\boldsymbol{s}^{(1)} + \boldsymbol{s}^{(2)} + \cdots + \boldsymbol{s}^{(T)}) \end{aligned}$$

で表される．

今後，簡単のため，

$$\boldsymbol{s}^{(1:t)} = \boldsymbol{s}^{(1)} + \boldsymbol{s}^{(2)} + \cdots + \boldsymbol{s}^{(t)}$$

と表記する．

2.5　評価手法

オンライン二分探索木問題は，未知の検索クエリ列に対して探索コストと移動コストの和を最小化することを目標とする．本稿では，**リグレット解析**によってオンライン二分探索木の性能を評価する．T 回の探索におけるあるオンライン二分探索木 A のリグレットを，

$$\begin{aligned} &\mathrm{Regret}_T(\mathrm{A}) \\ &= E[T \text{ 回における A の探索コストと移動コストの和}] \\ &\quad -M(\boldsymbol{s}^{(1:T)})\cdot(\boldsymbol{s}^{(1:T)}) \end{aligned}$$

と定義する [3].

第一項は T 回における A の探索コストと移動コストの総和の期待値であり，第二項は，T 回の探索を行った後に判明する最適二分探索木の探索コストである．

$\mathrm{Regret}_T(\mathrm{A}) = o(T)$ であれば，探索回数が増加するにつれて最適二分探索木とオンライン二分探索木 A とのコストの差が縮まる．

3.　既存アルゴリズム

3.1　Follow the Perturbed Leader

オンライン二分探索木問題は，エキスパート問題の応用であると捉えることができる．エキスパート問題の解法の一つとして，WMA(Weighted Majority Algorithm) がある．それに対し，Kalai と Vempala は FPL(Follow the Perturbed Leader Tree) アルゴリズムを提案した．FPL アルゴリズムの利点は，リグレットが WMA と殆ど変らない性能であるにもかかわらず，エキスパート問題以外の問題に対しても一般化がしやすい点にある [3].

FPL Tree アルゴリズムでは，各節点の探索回数にランダムに割り振った値を加え，その合計をもとに最適二分探索木を構成する．

```
FPL Tree
  パラメータ : N ∈ ℤ
  for t = 1 to T do
      for i = 1 to n do
          set p^(t)(v_i) ∈ {1, 2, ..., N} : 一様ランダムに選ぶ.
      set p^(t) = (p^(t)(v_1), p^(t)(v_2), ..., p^(t)(v_n)).
      if t=1 then
          最適二分探索木 M(p^(1)) を構築する.
      else
          最適二分探索木 M(s^(1:t-1) + p^(t)) を構築する.
      s^(t) を観測し，節点を探索する.
```

定理 1. FPL Tree のパラメータを $N \in \mathbb{Z}$ とすると，探索コストのリグレットは $O(\frac{nT}{N}) + O(n^2N)$，移動コストのリグレットは $O(n^3T)$ となる．

FPL Tree では各時刻毎に最適二分探索木を求める必要があるため，移動コストのリグレットは $o(T)$ にはならない．

3.2 Follow the Lazy Leader

Kalai と Vempala は，遅延更新型の FPL アルゴリズムである FLL(Follow the Lazy Leader) アルゴリズムを提案している．オンライン二分探索木問題のように，移動コストが発生するような場合には FPL アルゴリズムのリグレットは $o(T)$ にはならない．そこで，FLL アルゴリズムでは遅延更新手法を用いることで移動コストが生じる操作の回数を少なくしている．

FLL Tree

パラメータ：$N \in \mathbb{Z}$
for $i = 1$ to n **do**
　　set $f(v_i) = 0$.
　　set $p(v_i) \in \{1, 2, ..., N\}$：一様ランダムに選ぶ.
　　set $\boldsymbol{p} = (p(v_1), p(v_2), \dots, p(v_n))$.
最適二分探索木 $M(\boldsymbol{p})$ を構築する.
for $t = 1$ to T **do**
　　$\boldsymbol{s}^{(t)}$ を観測し，節点を探索する.
　　時刻 t で探索された節点を v_j とする.
　　$f(v_j) + +$
　　if $f(v_j) \geq p(v_j)$ **then**
　　　　$p(v_j) = p(v_j) + N$
　　　　set $\boldsymbol{p} = (p(v_1), p(v_2), \dots, p(v_n))$.
　　　　最適二分探索木 $M(\boldsymbol{p})$ を構築する.

定理 2. FLL Tree のパラメータを $N \in \mathbb{Z}$ とすると，探索コストのリグレットは $O(\frac{nT}{N}) + O(n^2 N)$，移動コストのリグレットは $O(\frac{n^3 T}{N})$ となる．

事前に T の値を知っていて，パラメータを $N = \sqrt{\frac{T}{n}}$ とすれば，上記の探索コストのリグレットは $O(n\sqrt{nT})$，移動コストのリグレットは $O(n^3\sqrt{nT})$ となり，$o(T)$ を達成する．

3.3 Hannan's Algorithms

Kalai と Vempala の手法は，事前に T の値を知っておかなければ探索コストのリグレットを $o(T)$ にすることができない．しかし，Hannan のアルゴリズム [4] をオンライン二分探索木に適用することで，T の事前情報を必要とせずとも探索コストのリグレットを $o(T)$ にすることが可能である．

Hannan's Tree

パラメータ：$N \in \mathbb{Z}$
for $t = 1$ to T **do**
　　for $i = 1$ to n **do**
　　　　set $p^{(t)}(v_i) \in \{1, 2, ..., N\lceil\sqrt{t}\rceil\}$：一様ランダムに選ぶ.
　　set $\boldsymbol{p}^{(t)} = (p^{(t)}(v_1), p^{(t)}(v_2), \dots, p^{(t)}(v_n))$.
　　if t=1 **then**
　　　　最適二分探索木 $M(\boldsymbol{p}^{(1)})$ を構築する.
　　else
　　　　最適二分探索木 $M(\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p}^{(t)})$ を構築する.
　　$\boldsymbol{s}^{(t)}$ を観測し，節点を探索する.

定理 3. Hannan's Tree のパラメータを $N \in \mathbb{Z}$ とすると，探索コストのリグレットは $O(\frac{2n\lceil\sqrt{T}\rceil}{N}) + O(n^2 N\sqrt{T})$，移動コストのリグレットは $O(n^3 T)$ となる．特に，$N = 1$ に対しても探索コストのリグレットは $O(\lceil\sqrt{T}\rceil)$ を達成することに注意を要する．

Hannan's Tree では各時刻毎に最適二分探索木を求める必要があるため，移動コストのリグレットは $o(T)$ にならない．

4. 提案アルゴリズム

Hannan のアルゴリズムに Kalai と Vempala の遅延更新手法を適用したアルゴリズムを提案し，リグレット解析を行う．本稿の目的は $O(\lceil\sqrt{T}\rceil)$ のリグレットの達成であり，簡単のため Hannan'Tree に現れるパラメータを $N = 1$ とした場合で記述する．

4.1 アルゴリズム

アルゴリズム 1

---
set $a = 0$.
for $i = 1$ to n **do**
　　set $f(v_i) = 0$
　　set $p^{(1)}(v_i) \in \{1, 2\}$：一様ランダムに選ぶ.
set $\boldsymbol{g}^{(1)} = (p^{(1)}(v_1), p^{(1)}(v_2), \dots, p^{(1)}(v_n))$.
最適二分探索木 $M(\boldsymbol{g}^{(1)})$ を構築する.
for $t = 1$ to T **do**
　　$\boldsymbol{s}^{(t)}$ を観測し，節点を探索する.
　　時刻 t で探索された節点を v_j とする.
　　$f(v_j) + +$
　　if $\lceil\sqrt{t}\rceil < \lceil\sqrt{t+1}\rceil$ (すなわち，$t = a^2$) **then**
　　　　a++
　　　　for $i = 1$ to n **do**
　　　　　　set $p^{(a)}(v_i) \in \{1, 2, ..., 2a\}$：一様ランダムに選ぶ.
　　　　　　set $g^{(a)}(v_i) = p^{(a)}(v_i)$
　　　　　　while $f(v_i) \geq g^{(a)}(v_i)$ **do**
　　　　　　　　set $g^{(a)}(v_i) = g^{(a)}(v_i) + 2a$
　　　　set $\boldsymbol{g}^{(a)} = (g^{(a)}(v_1), g^{(a)}(v_2), \dots, g^{(a)}(v_n))$.
　　　　最適二分探索木 $M(\boldsymbol{g}^{(a)})$ を構築する.
　　else
　　　　if $f(v_j) \geq g^{(a)}(v_j)$ **then**
　　　　　　set $g^{(a)}(v_j) = g^{(a)}(v_j) + 2a$.
　　　　　　set $\boldsymbol{g}^{(a)} = (g^{(a)}(v_1), g^{(a)}(v_2), \dots, g^{(a)}(v_n))$.
　　　　　　最適二分探索木 $M(\boldsymbol{g}^{(a)})$ を構築する.

定理 4. アルゴリズム *1* の探索コストのリグレットは $O(\frac{3n^2\lceil\sqrt{T}\rceil}{2}) + O(n\sqrt{T})$，移動コストのリグレットは $O(n^4\sqrt{T})$ となる．

4.2 準備

解析を行う前に，記法の定義と，Kalai と Vempala が示した補題の紹介を行う．

正成数 D，A，R を，

$$D = \max\{\|\boldsymbol{d} - \boldsymbol{d}'\|_1 \mid \boldsymbol{d}, \boldsymbol{d}' \in \mathcal{D}\}$$
$$A = \max\{\|\boldsymbol{s}\|_1 \mid \boldsymbol{s} \in \mathcal{S}\}$$
$$R = \max\{|\boldsymbol{d} \cdot \boldsymbol{s}'| \mid \boldsymbol{d} \in \mathcal{D}, \boldsymbol{s} \in \mathcal{S}\}$$

とする．ここで，$\|\boldsymbol{x}\|_1$ はベクトル $\boldsymbol{x} \in \mathbb{R}^n$ の L_1 ノルムを表し，

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

である．

オンライン二分探索木の場合，D，A，R はそれぞれ，

$$D \le n^2$$
$$A = 1$$
$$R \le n$$

を満たす．

また，$[0, N]^n$ を一辺の大きさが N である n 次元立方体とする．任意のベクトル $\boldsymbol{v} = (v_1, v_2, \ldots, v_n) \in \mathbb{R}^n$ に対し，$\boldsymbol{v} + [0, N]^n$ は原点を $(v_1, v_2, \ldots, v_n)$ とする一辺の大きさが N の n 次元立方体を表す．

補題 5. 任意の $T > 0$，任意の入力列 $\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(T)}$ および任意のベクトル列 $\boldsymbol{p}^{(0)} = 0$，$\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}, \ldots, \boldsymbol{p}^{(T)} \in \mathbb{R}^n$ に対して，

$$\sum_{t=1}^{T} M(\boldsymbol{s}^{(1:t)} + \boldsymbol{p}^{(t)}) \cdot \boldsymbol{s}^{(t)} \le M(\boldsymbol{s}^{(1:T)}) \cdot \boldsymbol{s}^{(1:T)} + D \sum_{t=1}^{T} \left\| \boldsymbol{p}^{(t)} - \boldsymbol{p}^{(t-1)} \right\|_\infty$$

が成り立つ．

補題 6. 任意のベクトル $\boldsymbol{v} \in \mathbb{R}^n$ に対して，n 次元立方体 $[0, N]^n$ と $\boldsymbol{v} + [0, N]^n$ は少なくとも $(1 - \frac{\|\boldsymbol{v}\|_1}{N})$ の割合を共有する．

4.3 解析

定理 4 の証明． $E\left[\sum_{t=1}^{T} M(\boldsymbol{g}^{(a)}) \cdot \boldsymbol{s}^{(t)}\right]$ が T 回における探索コストの期待値を表す．よって，

$$E\left[\sum_{t=1}^{T} M(\boldsymbol{g}^{(a)}) \cdot \boldsymbol{s}^{(t)}\right] - M(\boldsymbol{s}^{(1:T)}) \cdot \boldsymbol{s}^{(1:T)}$$

がアルゴリズムの探索コストのリグレットである．

アルゴリズム 1 では，時刻 t の時，$a = \lceil \sqrt{t} \rceil$ となる．その時の木構造は $M(\boldsymbol{g}^{(a)})$ であり，これは $\boldsymbol{p}^{(a)}$ には依存し，$\boldsymbol{g}^{(a')} (a' < a)$ には依存しない．ここで，各成分の値が $\{1, 2, \ldots, 2a\}$ をとる n 次元ベクトル $\boldsymbol{p'}^{(t)} \in \mathbb{R}^n$ を用いて，

$$\boldsymbol{g}^{(a)} = \boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)}$$

と書く．$\boldsymbol{p}^{(a)}$ の各成分の値は $\{1, 2, \ldots, 2a\}$ から一様ランダムに選択されているので，期待値に注目すると任意の i $(1 \le i \le n)$ に対して $E\left[p^{(a)}(v_i)\right] = E\left[p'^{(t)}(v_i)\right]$ が成り立つ．よって，

$$E\left[\boldsymbol{p}^{(a)}\right] = E\left[\boldsymbol{p'}^{(t)}\right] \tag{1}$$

となる．さらに，任意の i $(1 \le i \le n)$ に対して $E\left[p^{(a)}(v_i)\right] = E\left[ap^{(1)}(v_i)\right]$ が成り立つので，(1) 式より，

$$E\left[\boldsymbol{p}^{(a)}\right] = E\left[\boldsymbol{p'}^{(t)}\right] = E\left[a\boldsymbol{p}^{(1)}\right] \tag{2}$$

である．

また，補題 5 より，

$$\sum_{t=1}^{T} M(\boldsymbol{s}^{(1:t)} + E\left[\boldsymbol{p'}^{(t)}\right]) \cdot \boldsymbol{s}^{(t)} \le M(\boldsymbol{s}^{(1:T)}) \cdot \boldsymbol{s}^{(1:T)} + D \sum_{t=1}^{T} \left\| E\left[\boldsymbol{p'}^{(t)} - \boldsymbol{p'}^{(t-1)}\right] \right\|_\infty$$

が成り立つ．

(2) 式より，

$$\begin{aligned}
&\sum_{t=1}^{T} \left\| E\left[\boldsymbol{p'}^{(t)}\right] - E\left[\boldsymbol{p'}^{(t-1)}\right] \right\|_\infty \\
&= \left\| E[\boldsymbol{p'}^{(1)}] - E[\boldsymbol{p'}^{(0)}] \right\|_\infty + \left\| E[\boldsymbol{p'}^{(2)}] - E[\boldsymbol{p'}^{(1)}] \right\|_\infty \\
&\quad + \cdots + \left\| E[\boldsymbol{p'}^{(T)}] - E[\boldsymbol{p'}^{(T-1)}] \right\|_\infty \\
&= \left\| E[\boldsymbol{p}^{(1)}] \right\|_\infty + \left\| \left\lceil \sqrt{2} \right\rceil E[\boldsymbol{p}^{(1)}] - E[\boldsymbol{p}^{(1)}] \right\|_\infty \\
&\quad + \cdots + \left\| \left\lceil \sqrt{T} \right\rceil E[\boldsymbol{p}^{(1)}] - \left\lceil \sqrt{T-1} \right\rceil E[\boldsymbol{p}^{(1)}] \right\|_\infty \\
&= \left\| E[\boldsymbol{p}^{(1)}] \right\|_\infty \sum_{t=1}^{T} (\left\lceil \sqrt{t} \right\rceil - \lceil \sqrt{t-1} \rceil) \\
&= \left\| E[\boldsymbol{p}^{(1)}] \right\|_\infty \left\lceil \sqrt{T} \right\rceil \\
&\le \frac{3 \left\lceil \sqrt{T} \right\rceil}{2}
\end{aligned}$$

である．

従って，

$$\begin{aligned}
&\sum_{t=1}^{T} E\left[M(\boldsymbol{s}^{(1:t)} + \boldsymbol{p'}^{(t)}) \cdot \boldsymbol{s}^{(t)}\right] \\
&= \sum_{t=1}^{T} M(\boldsymbol{s}^{(1:t)} + E\left[\boldsymbol{p'}^{(t)}\right]) \cdot \boldsymbol{s}^{(t)} \\
&\le M(\boldsymbol{s}^{(1:T)}) \cdot \boldsymbol{s}^{(1:T)} + \frac{3D \left\lceil \sqrt{T} \right\rceil}{2}
\end{aligned} \tag{3}$$

となる．

次に，時刻 t での探索に最適二分探索木 $M(\boldsymbol{s}^{(1:t-1)} + E\left[\boldsymbol{p'}^{(t)}\right])$ と $M(\boldsymbol{s}^{(1:t)} + E\left[\boldsymbol{p'}^{(t)}\right])$ を用いた場合のコストの差を求める．

$\boldsymbol{s}^{(1:t-1)} + [0, 2a]^n$ と $\boldsymbol{s}^{(1:t)} + [0, 2a]^n$ の共通部分に $\boldsymbol{p}^{(a)} = \boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)}$ が存在するとき，

$$\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)} = \boldsymbol{s}^{(1:t)} + \boldsymbol{p''}^{(t)}$$

を満たすように，各成分の値が $[0, 2a]$ をとる n 次元ベクトル $\boldsymbol{p''}^{(t)} \in \mathbb{R}^n$ を選ぶ．このとき，時刻 t での探索に必要なコストは，$M(\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)})$ と $M(\boldsymbol{s}^{(1:t)} + \boldsymbol{p''}^{(t)})$ のどちらを使っても等しくなる．

一方，$\boldsymbol{s}^{(1:t-1)} + [0, 2a]^n$ と $\boldsymbol{s}^{(1:t)} + [0, 2a]^n$ の共通部分に $\boldsymbol{p}^{(a)} = \boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)}$ が存在しないとき，

$$\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)} \neq \boldsymbol{s}^{(1:t)} + \boldsymbol{p''}^{(t)}$$

である．この場合は，2 つの最適二分探索木 $M(\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)})$ と $M(\boldsymbol{s}^{(1:t)} + \boldsymbol{p''}^{(t)})$ の構造が異なるので，

$$\begin{aligned} &M(\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)}) \cdot \boldsymbol{s}^{(t)} \leq M(\boldsymbol{s}^{(1:t)} + \boldsymbol{p''}^{(t)}) \cdot \boldsymbol{s}^{(t)} + R \\ &\quad (\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)} \neq \boldsymbol{s}^{(1:t)} + \boldsymbol{p''}^{(t)}) \end{aligned} \tag{4}$$

が成り立つ．

補題 6 より，$\boldsymbol{s}^{(1:t-1)} + [0, 2a]^n$ と $\boldsymbol{s}^{(1:t)} + [0, 2a]^n$ は少なくとも $(1 - \left\|\boldsymbol{s}^{(t)}\right\|_1 / 2a)$ の割合を共有している．従って，$\boldsymbol{s}^{(1:t-1)} + [0, 2a]^n$ と $\boldsymbol{s}^{(1:t)} + [0, 2a]^n$ の共通部分に $\boldsymbol{p}^{(a)} = \boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)}$ が存在しない確率は，高々 $\left\|\boldsymbol{s}^{(t)}\right\|_1 / 2a$ である．

従って，時刻 t での探索コストの期待値について (4) より，

$$\begin{aligned} &E\left[M(\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)}) \cdot \boldsymbol{s}^{(t)}\right] \\ &\quad \leq E\left[M(\boldsymbol{s}^{(1:t)} + \boldsymbol{p''}^{(t)}) \cdot \boldsymbol{s}^{(t)}\right] + \frac{R\left\|\boldsymbol{s}^{(t)}\right\|_1}{2a} \\ &\quad \leq E\left[M(\boldsymbol{s}^{(1:t)} + \boldsymbol{p''}^{(t)}) \cdot \boldsymbol{s}^{(t)}\right] + \frac{RA}{2\left\lceil\sqrt{t}\right\rceil} \end{aligned} \tag{5}$$

が成り立つ．

また，期待値に注目すると，

$$E\left[\boldsymbol{p'}^{(t)}\right] = E\left[\boldsymbol{p''}^{(t)}\right]$$

である．

よって，(5) 式より，

$$\begin{aligned} &E\left[M(\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)}) \cdot \boldsymbol{s}^{(t)}\right] \\ &\quad \leq E\left[M(\boldsymbol{s}^{(1:t)} + \boldsymbol{p'}^{(t)}) \cdot \boldsymbol{s}^{(t)}\right] + \frac{RA}{2\left\lceil\sqrt{t}\right\rceil} \\ &\quad = M(\boldsymbol{s}^{(1:t)} + E\left[\boldsymbol{p'}^{(t)}\right]) \cdot \boldsymbol{s}^{(t)} + \frac{RA}{2\left\lceil\sqrt{t}\right\rceil} \end{aligned} \tag{6}$$

最後に (3)，(6) 式より，

$$\begin{aligned} &\sum_{t=1}^{T} E\left[M(\boldsymbol{g}^{(a)}) \cdot \boldsymbol{s}^{(t)}\right] \\ &\quad = \sum_{t=1}^{T} E\left[M(\boldsymbol{s}^{(1:t-1)} + \boldsymbol{p'}^{(t)}) \cdot \boldsymbol{s}^{(t)}\right] \\ &\quad \leq \sum_{t=1}^{T} \left(M(\boldsymbol{s}^{(1:t)} + E\left[\boldsymbol{p'}^{(t)}\right]) \cdot \boldsymbol{s}^{(t)} + \frac{RA}{2\left\lceil\sqrt{t}\right\rceil}\right) \\ &\quad \leq \sum_{t=1}^{T} M(\boldsymbol{s}^{(1:t)} + E\left[\boldsymbol{p'}^{(t)}\right]) \cdot \boldsymbol{s}^{(t)} + \sum_{t=1}^{T} \frac{RA}{2\left\lceil\sqrt{t}\right\rceil} \\ &\quad \leq M(\boldsymbol{s}^{(1:T)}) \cdot \boldsymbol{s}^{(1:T)} + \frac{3D\left\lceil\sqrt{T}\right\rceil}{2} + RA\sqrt{T} \\ &\quad \leq M(\boldsymbol{s}^{(1:T)}) \cdot \boldsymbol{s}^{(1:T)} + \frac{3n^2\left\lceil\sqrt{T}\right\rceil}{2} + n\sqrt{T} \end{aligned} \tag{7}$$

が得られる．

$E\left[\sum_{t=1}^{T} M(\boldsymbol{g}^{(a)}) \cdot \boldsymbol{s}^{(t)}\right]$ がアルゴリズムの探索コストの期待値であるので，題意を得る．

次に，アルゴリズム 1 の移動コストのリグレットを示す．アルゴリズム 1 では，$\left\lceil\sqrt{t}\right\rceil < \left\lceil\sqrt{t+1}\right\rceil$ を満たすときに木構造の更新を行う．この回数は，T 回の探索につき高々 $\sqrt{T}$ 回である．

また，$t = (a-1)^2$ が成り立った後 $t' = a^2$ $(t < t')$ が成り立つまでに節点の探索は $a^2 - (a-1)^2 = 2a - 1$ 回行われ，探索した節点 v_j について $f(v_j) \geq p^{(a)}(v_j)$ であれば木構造の更新を行う．一度節点 v_j について $f(v_j) \geq p^{(a)}(v_j)$ が成り立つと，$p^{(a)}(v_j)$ に $2a$ を加算するので，次に $f(v_j) \geq p^{(a)}(v_j)$ が成立するのは少なくとも $2a$ 回探索が行われた後である．従って，$2a - 1$ 回の探索の間に，各節点について一度ずつしかこの操作は発生しない．

以上より，アルゴリズム 1 では T 回の探索につき高々 $n\sqrt{T}$ 回しか木構造の更新を行わない．最適二分探索木を計算し，構築するためには $O(n^3)$ 時間が必要であるので，移動コストのリグレットは $O(n^4\sqrt{T})$ である．

□

5. まとめ

エキスパート問題及びエキスパート問題に類する問題を解くアルゴリズムとして，Kalai と Vempala は FPL アルゴリズム，FLL アルゴリズムを与えた．FPL アルゴリズムをオンライン二分探索木問題に用いることで，探索コストのリグレットを $o(T)$ にすることができる．しかし，各探索の後に最適二分探索木を計算する必要があるので，移動コストのリグレット $O(T)$ である．

FLL アルゴリズムは，遅延更新手法を導入することで上記の FPL アルゴリズムの移動コストに関しても $o(T)$ を達

成している．FLL アルゴリズムの課題は，探索コストと移動コストを $o(T)$ にするために事前に T の値を知っておかなければならないことである．

FLL アルゴリズムに対し，Hannan のアルゴリズムを用いた Hannan's Tree では，T に関する事前情報なしで探索コストのリグレットを $o(T)$ にすることができる．しかし，移動コストに関しては FPL アルゴリズムと同等の性能である．

以上の各アルゴリズムの特徴を踏まえ，本稿では遅延更新型の Hannan's Tree であるアルゴリズムを提案した．このアルゴリズムは Hannan のアルゴリズムをベースにしており，T に関する事前情報なしで探索コストのリグレットを $o(T)$ にすることができる．また，FLL アルゴリズムで用いられている遅延更新手法を適用することで，移動コストのリグレットについても $o(T)$ を達成した．

参考文献

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer, Finite-time analysis of the multiarmed bandit problem, Machine Learning, 47 (2002), 235–256.

[2] N. Cesa-Bianchi, O. Dekel, and O.Shamir Online learning with switching costs and other adaptive adversaries, arXiv:1302.4387v2, 2013.

[3] A. Kalai, S. Vempala, Efficient algorithms for online decision problem, Journal of Computer and System Sciences, 71 (2005), 291–307.

[4] J. Hannan, Approximation to Bayes risk in repeated plays, Contributions to the Theory of Games, vol. 3, Princeton University Press, Princeton, (1957), 97–139.

[5] E. Hazan, The convex optimization approach to regret minimization, http://ie.technion.ac.il/~ehazan/papers/shalom.pdf

[6] T.H. コルメン，C.E. ライザーソン，R.L. リベスト，C. シュタイン (共著), 浅野哲夫，岩野和生，梅尾博司，山下雅史，和田幸一 (共訳), アルゴリズムイントロダクション 第 3 版, 近代科学社，2013，pp.330–337.