Vol.2015-ICS-179 No.15
2015/3/20

動的報酬予算制限多腕バンディット問題のアルゴリズム D-KUBEとSW-KUBEの提案

新美 真[1,a)]　伊藤 孝行[1,b)]

概要：本研究では，多腕バンディット問題を拡張した予算制限多腕バンディット問題を取り扱う．多腕バンディット問題とは，複数台あるスロットマシンをプレイするギャンブラーを模した問題である．予算制限多腕バンディット問題は多腕バンディット問題の拡張の一つで，コストと予算による制約が存在する．既存の予算制限多腕バンディット問題では静的な報酬確率分布のみを仮定しており，動的な報酬確率分布については想定していない．実世界の応用では，動的な報酬確率分布を想定したほうがより現実的である．例えば，予算制限多腕バンディット問題の実世界の応用の一つであるオンライン広告の効果は，トレンドや時間による影響により動的であると想定されるためである．本研究では予算制限多腕バンディット問題及び予算制限バンディットアルゴリズムを拡張し，動的な報酬確率分布を想定する．予算制限多腕バンディット問題の拡張に伴い，既存の予算制限バンディットアルゴリズムを拡張した D-KUBE 及び SW-KUBE を提案する．動的な報酬確率分布による人工的な問題空間を設定し，既存手法である KUBE と提案手法である D-KUBE 及び SW-KUBE との比較実験を行う．実験結果から動的な報酬確率分布において，提案手法である D-KUBE 及び SW-KUBE は既存手法である KUBE と比較して改善されることを確認する．

1. はじめに

多腕バンディット問題とは，複数台あるスロットマシン (以降アームと呼ぶ) をプレイするギャンブラーを模した問題である．アームから得られる報酬は，それぞれ独立で適当な確率分布に従うと仮定する．ギャンブラーの役割をするエージェントは，決められたプレイ回数の中で得られる報酬を最大化することを目的とする．得られる報酬を最大化するために，エージェントはアームの探索と活用をどのように行うかを求められる．探索とは既知でない，他の複数のアームを試行することである．探索を行うことで，より良いアームを選択するための情報を獲得する．活用とは，既知の情報をもとに良いアームを選ぶことである．活用を行うことで，探索して得られた情報を有効利用することが可能になる．しかし，探索に焦点を置くと正確な情報を得られるが，損失を生み出してしまう恐れがある．一方で，活用に焦点を置くとより良いアームを発見できない．探索と活用の間にあるトレードオフのことを探索と活用のジレンマと呼ぶ．バンディットアルゴリズムとはアームの探索と活用の均衡を取り，利益を報酬を最大化するためのアルゴリズムである．

本研究では，予算制限多腕バンディット問題及び予算制限バンディットアルゴリズムを拡張する．予算制限多腕バンディット問題とは，多腕バンディット問題に制約を持たせて拡張した問題の一つである．予算制限多腕バンディット問題は予算による制約が存在し，アームを選択する際に予算を消費しなければならない．エージェントは限られた予算の中で得られる報酬を最大化することを目的とする．

予算制限多腕バンディット問題の応用例として，オンライン広告，クラウドソーシング，及びワイヤレスセンサネットワークなどが挙げられる [1][2][3][4]．拡張する理由として，既存の予算制限多腕バンディット問題における設定では報酬の確率分布が変化せず，静的であるという仮定を立てていることが挙げられる．つまり，アームから得られる報酬の確率分布が一度決定されると以降は変化しない．しかし，現実世界の問題では，報酬の確率分布が変化し動的であることが想定される．

例として，ある企業が自社の商品やキャンペーンを宣伝するためにオンライン広告を用いる場合を考える．これを予算制限多腕バンディット問題と捉えると，企業の資金，Web ページや Web サイト，オンライン広告を打ち出す為に必要な金額，及びオンライン広告のクリック数，インプレッション数及びコンバージョン数などがそれぞれ，予算，

[1] 名古屋工業大学
Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555, Japan
[a)] niimi.makoto@itolab.nitech.ac.jp
[b)] ito.takayuki@nitech.ac.jp

アーム，コスト，及び報酬に当たる．オンライン広告は，時間やイベントの影響及び個人の嗜好の変化により得られる報酬が変化することが想定される．例えば，本橋ら [5] はオンライン広告の一つであるバナー広告の効果について，トレンド及び日付による影響により変動するとしている．従って，予算制限多腕バンディット問題を拡張し，報酬の確率分布を動的にする拡張は必要である．また，既存のアルゴリズムでは動的な報酬確率分布を想定していない．従って，提案する問題空間に適応した予算制限バンディットアルゴリズムを提案する．

以下に本論文の構成を示す．まず，2 章では本論文の関連研究として，確率的多腕バンディット問題，予算制限多腕バンディット問題及び動的報酬多腕バンディット問題について説明する．さらに，予算制限多腕バンディット問題及び動的報酬多腕バンディット問題のバンディットアルゴリズムについて紹介する．3 章では本論文で提案する動的報酬による予算制限多腕バンディット問題，及び動的報酬対応予算制限バンディットアルゴリズムである D-KUBE 及び SW-KUBE について述べる．4 章では実験設定及び結果について述べる．最後に，本論文のまとめを示し，今後の課題を述べる．

2. 既存の多腕バンディット問題とアルゴリズム

2.1 多腕バンディット問題

多腕バンディット問題の起源は，治験計画に関するものが最初である [6]．その後，多腕バンディット問題は Robbins[7] により定式化された．近年は，制約や設定を加え，多腕バンディット問題を拡張した研究がなされている．

確率的多腕バンディット問題

確率的多腕バンディット問題とは，多腕バンディット問題の中で標準的な多腕バンディット問題である．確率的多腕バンディット問題には，プレイ可能な K 本のアームが存在する．エージェントは，タイムステップごとにこれらのアームの中から一つをプレイする．エージェントは，アームをプレイすることにより報酬を獲得する．アームから得られる報酬はそれぞれ独立した，異なる確率分布に従う．エージェントの目的は，限られたプレイ回数 T の中で，受け取る報酬の合計を最大化することである．受け取る報酬の合計を最大化するために，エージェントはアームから受け取る報酬の期待値を最大化しなければならない．従って，可能な限り少ない探索で最も期待報酬の高いアームを見つけ繰り返しプレイすることが目的となる．

予算制限多腕バンディット問題

予算制限多腕バンディット問題とは，確率的多腕バンディット問題を拡張した多腕バンディット問題の一つである．確率的多腕バンディット問題と予算制限多腕バンディット問題との違いは，アームのプレイ回数が異なる点である．確率的多腕バンディット問題では，限られたプレイ回数の中で報酬の合計を最大化している．しかし，予算制限多腕バンディット問題には二つの制約があるためにプレイ回数が一意でない．二つの制約とは，予算とコストである．予算制限多腕バンディット問題では，それぞれのアームに対してコストが設定される．エージェントは予算を所持しており，アームを選択する際に予算を消費しなくてはならない．エージェントは，アームにコストが設定された中で与えられた予算に従って探索及び活用を行う．限られた予算の中で，利益を最大化することがエージェントの目的となる．

動的報酬多腕バンディット問題

動的報酬多腕バンディット問題とは，確率的多腕バンディット問題を拡張した多腕バンディット問題の一つである．確率的多腕バンディット問題と動的報酬多腕バンディット問題の違いは，アームの報酬確率分布の時間経過による変化の有無である．確率的多腕バンディット問題では，報酬確率分布が時間経過により変化しない中で報酬の合計を最大化している．動的報酬多腕バンディット問題では，アームの報酬確率分布が時間経過により変化する設定がある．報酬が動的であるため，エージェントは報酬分布の変化に対応し，報酬の合計を最大化することを目的とする．

2.2 予算制限バンディットアルゴリズム

KUBE

knapsack based upper confidence bound exploration and exploitation（以降 KUBE と呼ぶ）は，活用と同時に探索も行う予算制限バンディットアルゴリズムである [8]．KUBE を Algorithm1 に記述する．KUBE はそれ

Algorithm 1 The KUBE Algorithm

$t = 1; B_t = B$;
while pulling is feasible **do**
  if $B_t < \min_i c_i$ **then**
    STOP! { pulling is not feasible}
  end if
  if $t \leq K$ **then**
    Initial phase: play arm $i(t) = t$;
  else
    use density-ordered greedy to calculate $M^*(B_t) = \{m^*_{i,t}\}$, the solution of Equation 2.3;
    randomly pull $i(t)$ with $P(i(t) = i) = \frac{m^*_{i,t}}{\sum_{k=1}^{K} m^*_{k,t}}$;
  end if
  update the estimated upper bound of arm $i(t)$;
  $B_{t+1} = B_t - c_{i(t)}; t = t + 1$;
end while

ぞれのタイムステップ t で，アームがプレイ可能かどうかを確認する (steps 3-4)．もしアームがプレイ可能である場合，KUBE は初期フェイズとしてそれぞれの腕を一度だけプレイする (steps 6-7)．その後タイムステップ $t > K$ で最も良いと思われるマシンの組み合わせを推定しアームをプレイする (steps 9-10)．推定には貪欲法を式 (1) に適用して求める．

$$\max \sum_{i=1}^{K} m_{i,t} \left(\hat{\mu}_{i,n_{i,t}} + \sqrt{\frac{2 \ln t}{n_{i,t}}} \right)$$
$$\text{s.t.} \quad \sum_{i=1}^{K} m_{i,t} c_i \leq B_t, \forall i, t : m_{i,t} \text{integer} \qquad (1)$$

ここで $m_{i,t}$，$\hat{\mu}_{i,n_i}$，及び n_i はそれぞれ式 (1) を満たすアームのプレイ回数，アーム i がプレイして得られた報酬の平均から求められた推定報酬，及びタイムステップ t までにアーム i をプレイした回数を表す．$\sqrt{\frac{2 \ln t}{n_{i,t}}}$ は，タイムステップ t でのアーム i の探索手当を意味する．特に，それぞれのタイムステップ s で選択されたアームを $i(s)$，得られた報酬を $r(s)$ とすると，$\hat{\mu}_{i,n_i}$ は式 (2) を計算することで求められる．

$$\hat{\mu}_{i,n_i} = \frac{1}{n_{i,t}} \sum_{s=1}^{t} \mathbf{I}_{\{i(s)=i\}} r(s) \qquad (2)$$

エージェントの目標は残りの予算 B_t に対応した KUBE の式 (1) を満たす定数 $\{m_{i,t}\}_{i \in K}$ を見つけることである．ここで $\mathbf{I}_{\{i(s)=i\}}$ は，$i(s) = i$ となる時 1 を返す指示関数である．この問題は NP-hard であるため，貪欲法を用いてアームの組み合わせを求めている．アーム i の期待報酬密度に基づく評価値は

$$\frac{\hat{\mu}_{i,n_{i,t}}}{c_i} + \frac{\sqrt{\frac{2 \ln t}{n_{i,t}}}}{c_i} \qquad (3)$$

より求められる．ここで KUBE の式 (1) を満たす最大となるアームの組み合わせの解を $M^*(B_t) = \{m^*_{i,t}\}$ とする．$\{m^*_{i,t}\}$ を用いて KUBE はランダムにプレイするアームを選択する．プレイされるアームの確率は式 (4) に従う．

$$P(i(t) = i) = \frac{m^*_{i,t}}{\sum_{k=1}^{K} m^*_{k,t}} \qquad (4)$$

プレイされた後，選択されたアームの推定上限と残りの予算 B_t を更新する（steps 12-13）．

2.3 動的報酬バンディットアルゴリズム

D-UCB

Discounted UCB（以降 D-UCB と呼ぶ）は，減衰率 γ を用いて動的な報酬確率分布に適応させたものである [9]．アルゴリズムを Algorithm2 に記述し，詳細を説明する．

ここで t はタイムステップを表す．D-UCB はまずそれぞれのアームを一度だけプレイする．その後タイムステップごとに，最も良いと推定されたアームをプレイする．即時的な期待報酬 $\bar{\mu}_t(\gamma, i)$ は，式 (5) 及び式 (6) から求められる．

Algorithm 2 Discounted UCB
1: **for** t from 1 to K **do**
2: play arm $i(t) = t$;
3: **end for**
4: **for** t from $K+1$ to T **do**
5: play arm $i(t) = \arg\max_{1 \leq i \leq K} \bar{\mu}_t(\gamma, i) + b_t(\gamma, i)$.
6: **end for**

$$\bar{\mu}_t(\gamma, i) = \frac{1}{n_t(\gamma, i)} \sum_{s=1}^{t} \gamma^{t-s} r_s(i) \mathbf{I}_{\{i(s)=i\}} \qquad (5)$$

$$n_t(\gamma, i) = \sum_{s=1}^{t} \gamma^{t-s} \mathbf{I}_{\{i(s)=i\}} \qquad (6)$$

この時，$n_t(\gamma, i)$，$r_s(i)$，及び $\mathbf{I}_{\{i(s)=i\}}$ はそれぞれ，減衰率の和，タイムステップ s でアーム i から得られた報酬，及び $i(s) = i$ となる時 1 を返す指示関数である．$i(t)$ はタイムステップ t で選択されたアームである．つまり，最近得られた報酬に重みをつけて報酬の推定を行うことによって動的な報酬に適応している．$b_t(\gamma, i)$ は，減衰探索手当であり，式 (7) から求められる．

$$b_t(\gamma, i) = 2 \sqrt{\frac{\xi \log n_t(\gamma)}{n_t(\gamma, i)}} \qquad (7)$$

この時 $n_t(\gamma)$ は式 (8) より求められる．

$$n_t(\gamma) = \sum_{i=1}^{K} n_t(\gamma, i) \qquad (8)$$

ここで ξ は定数を設定する．ξ の値の範囲は，$\frac{1}{2} < \xi \leq 1$ である．

SW-UCB

Sliding-Window UCB（以降 SW-UCB と呼ぶ）は，参照数 τ を用いて動的な報酬確率分布に適応させたものである [10]．SW-UCB を Algorithm3 に記述し，詳細を説明する．

Algorithm 3 Sliding-Window UCB
1: **for** t from 1 to K **do**
2: play arm $i(t) = t$;
3: **end for**
4: **for** t from $K+1$ to T **do**
5: play arm $i(t) = \arg\max_{1 \leq i \leq K} \bar{\mu}_t(\tau, i) + b_t(\tau, i)$.
6: **end for**

t はタイムステップを表す．SW-UCB は，まずそれぞれのアームを一度だけプレイする．探索後タイムステップごとに，最も良いと推定されたアームをプレイする．即時的

な期待報酬 $\bar{\mu}_t(\tau,i)$ は，式 (9) 及び式 (10) から求められる．

$$\bar{\mu}_t(\tau,i) = \frac{1}{n_t(\tau,i)} \sum_{s=t-\tau+1}^{t} r_s(i)\mathbf{I}_{\{i(s)=i\}} \tag{9}$$

$$n_t(\tau,i) = \sum_{s=t-\tau+1}^{t} \mathbf{I}_{\{i(s)=i\}} \tag{10}$$

この時，$n_t(\tau,i)$，$r_s(i)$，及び $\mathbf{I}_{i(s)=i}$ はそれぞれ現在のタイムステップ t から $t-\tau+1$ までの選択回数，タイムステップ s でアーム i から得られた報酬，及び $i(s)=i$ となる時 1 を返す指示関数を表す．$i(t)$ はタイムステップ t で選択されたアームである．$b_t(\tau,i)$ は，減衰探索手当であり式 (11) によって表される．

$$b_t(\tau,i) = \sqrt{\frac{\xi \log(t \wedge \tau)}{n_t(\tau,i)}} \tag{11}$$

この時 $t \wedge \tau$ は，t 及び τ の最小値を意味する．ここで ξ は，定数を設定する．ξ の値の範囲は，$\frac{1}{2} < \xi \leq 1$ である．

3. 提案する多腕バンディット問題とアルゴリズム

3.1 動的報酬予算制限多腕バンディット問題

動的報酬予算制限多腕バンディット問題は，既存研究の予算制限多腕バンディット問題を拡張した多腕バンディット問題である．動的報酬予算制限多腕バンディット問題は，それぞれのアームにコストが設定されている，エージェントは，予算を所持しておりアームをプレイする際に，予算を消費しなくてはならない．さらにアームの報酬確率分布が，時間経過により変動する．動的報酬予算制限多腕バンディット問題では，限られた予算の中で報酬確率分布の変動に適応し報酬の合計を最大化することを目的とする．

3.2 動的報酬予算制限バンディットアルゴリズム D-KUBE

decreasing knapsack based upper confidence bound exploration and exploitation（以降 D-KUBE と呼ぶ）は，KUBE に D-UCB の推定報酬の算出方法を組み合わせたアルゴリズムである．D-KUBE は，D-UCB で用いる減衰率 γ を用いて，KUBE を動的な報酬確率分布に適応させる．D-KUBE アルゴリズムを Algorithm4 に記述し，詳細に説明する．

D-KUBE はそれぞれのタイムステップ t で，アームがプレイ可能かどうかを確認する（steps 3-4）．もし，アームがプレイ可能である場合，D-KUBE はまずそれぞれの腕を一度だけプレイする（steps 6-7）．その後，タイムステップ $t > K$ で，最も良いと思われるマシンの組み合わせを推定する．推定には，ナップザック問題を式 (12) に適用し，問題を解くことで求める．

Algorithm 4 The D-KUBE Algorithm

```
1: t = 1; B_t = B;
2: while pulling is feasible do
3:    if B_t < min_i c_i then
4:       STOP! { pulling is not feasible}
5:    end if
6:    if t ≤ K then
7:       Initial phase: play arm i(t) = t;
8:    else
9:       use density-ordered greedy to calculate M*(B_t) = {m*_{i,t}}, the solution of Equation 12;
10:      randomly pull i(t) with P(i(t) = i) = m*_{i,t} / Σ_{k=1}^{K} m*_{k,t};
11:   end if
12:   update the estimated evaluation value of arm i(t) by Equation 17;
13:   B_{t+1} = B_t − c_{i(t)}; t = t + 1;
14: end while
```

$$\max \sum_{i=1}^{K} m_{i,t} \left(\bar{\mu}_t(\gamma,i) + b_t(\gamma,i)\right)$$
$$\text{s.t.} \quad \sum_{i=1}^{K} m_{i,t} c_i \leq B_t, \forall i,t : m_{i,t}\text{integer} \tag{12}$$

ここで，$m_{i,t}$，$\bar{\mu}_t(\gamma,i)$，及び $b_t(\gamma,i)$ はそれぞれ，式 (12) を満たすタイムステップ t でのアーム i のプレイ回数，タイムステップ t でのアーム i の即時的な期待報酬，及びタイムステップ t でのアーム i の減衰探索手当を表す．即時的な期待報酬 $\bar{\mu}_t(\gamma,i)$ は，式 (13) 及び式 (14) より求められる．

$$\bar{\mu}_t(\gamma,i) = \frac{1}{n_t(\gamma,i)} \sum_{s=1}^{t} \gamma^{t-s} r_s(i)\mathbf{I}_{\{i(s)=i\}} \tag{13}$$

$$n_t(\gamma,i) = \sum_{s=1}^{t} \gamma^{t-s} \mathbf{I}_{\{i(s)=i\}} \tag{14}$$

この時，$n_t(\gamma,i)$，$r_s(i)$，及び $\mathbf{I}_{\{i(s)=i\}}$ はそれぞれ，減衰率の和，タイムステップ s でアーム i から得られた報酬，及び $i(s)=i$ となる時 1 を返す指示関数を表す．$i(t)$ はタイムステップ t で選択されたアームである．また減衰探索手当 $b_t(\gamma,i)$ は式 (15) より求められる．

$$b_t(\gamma,i) = 2\sqrt{\frac{\xi \log n_t(\gamma)}{n_t(\gamma,i)}} \tag{15}$$

この時

$$n_t(\gamma) = \sum_{i=1}^{K} n_t(\gamma,i) \tag{16}$$

ここで ξ は $0.5 < \xi \leq 1$ となる定数を設定する．

目標は，余りの予算 B_t に対応した D-KUBE の式 (3.2) を満たす定数 $\{m_{i,t}\}_{i \in K}$ を見つけることである．この問題は，NP 困難であるため貪欲法を用いてアームの準最適組み合わせを求めている．アーム i の期待報酬密度に基づく評価値は，式 (17) より求められる．

$$\frac{\bar{\mu}_t(\gamma,i)}{c_i}+\frac{b_t(\gamma,i)}{c_i} \tag{17}$$

ここで，D-KUBE の式 () を満たす最大となるアームの組み合わせの解を $M^*(B_t)=\{m_i^*,t\}$ とする．$\{m_{i,t}^*\}$ を用いて，D-KUBE はランダムにプレイする腕を選択する．プレイされるアームの確率は式 (18) に従う．

$$P(i(t)=i)=\frac{m_{i,t}^*}{\sum_{k=1}^{K}m_{k,t}^*} \tag{18}$$

プレイされた後選択されたアームの評価値と残りの予算 B_t を更新する（steps 12-13）．残り予算が無くなり，アームをプレイすることができなくなるまで繰り返される．

SW-KUBE

sliding-window knapsack based upper confidence bound exploration and exploitation（以降 SW-KUBE と呼ぶ）は，KUBE に SW-UCB の推定報酬の算出方法を組み合わせたアルゴリズムである．SW-KUBE は SW-UCB で用いる参照数 τ を用いて，KUBE を動的な報酬確率分布に適応させる．SW-KUBE アルゴリズムを Algorithm5 に記述し，詳細に説明する．

Algorithm 5 The SW-KUBE Algorithm

1: $t=1; B_t=B$;
2: **while** pulling is feasible **do**
3: **if** $B_t<\min_i c_i$ **then**
4: STOP! { pulling is not feasible}
5: **end if**
6: **if** $t\le K$ **then**
7: Initial phase: play arm $i(t)=t$;
8: **else**
9: use density-ordered greedy to calculate $M^*(B_t)=\{m_{i,t}^*\}$, the solution of Equation 19;
10: randomly pull $i(t)$ with $P(i(t)=i)=\frac{m_{i,t}^*}{\sum_{k=1}^{K}m_{k,t}^*}$;
11: **end if**
12: update the estimated evaluation value of arm $i(t)$ by Equation 23;
13: $B_{t+1}=B_t-c_{i(t)}; t=t+1$;
14: **end while**

SW-KUBE はそれぞれのタイムステップ t で，アームがプレイ可能かどうかを確認する（steps 3-4）．もしアームがプレイ可能である場合，SW-KUBE はまずそれぞれの腕を一度だけプレイする（steps 6-7）．その後タイムステップ $t>K$ で，最も良いと思われるマシンの組み合わせを推定する．推定には貪欲法を式 (19) に適用して求める．

$$\max\sum_{i=1}^{K}m_{i,t}\left(\bar{\mu}_t(\tau,i)+b_t(\tau,i)\right)$$
$$\text{s.t.}\quad\sum_{i=1}^{K}m_{i,t}c_i\le B_t,\forall i,t: m_{i,t}\text{integer} \tag{19}$$

ここで，$m_{i,t}$ は式 (19) を満たすアームのプレイ回数，$\bar{\mu}_t(\tau,i)$ は即時的な期待報酬，$b_t(\tau,i)$ は減衰探索手当を表す．即時的な期待報酬 $\bar{\mu}_t(\tau,i)$ は，式 (20) 及び式 (21) より求められる．

$$\bar{\mu}_t(\tau,i)=\frac{1}{n_t(\tau,i)}\sum_{s=t-\tau+1}^{t}r_s(i)\mathbf{I}_{\{i(s)=i\}} \tag{20}$$

$$n_t(\tau,i)=\sum_{s=t-\tau+1}^{t}\mathbf{I}_{\{i(s)=i\}} \tag{21}$$

この時，$n_t(\tau,i)$，$r_s(i)$，及び $\mathbf{I}_{i(s)=i}$ はそれぞれ，現在のタイムステップ t から $t-\tau+1$ までの選択回数，タイムステップ s でアーム i から得られた報酬，及び $i(s)=i$ となる時 1 を返す指示関数である．I_t はタイムステップ t で選択されたアームである．減衰探索手当 $b_t(\tau,i)$ は，式 (22) より求められる．

$$b_t(\tau,i)=\sqrt{\frac{\xi\log(t\wedge\tau)}{n_t(\tau,i)}} \tag{22}$$

この時，$t\wedge\tau$ は，t 及び τ の最小値により表される．ここで ξ は $0.5<\xi\le 1$ となる定数を設定する．目標は，余りの予算 B_t に対応した SW-KUBE の式 (3.1) を満たす定数 $\{m_{i,t}\}_{i\in K}$ を見つけることである．この問題は NP-hard であるため貪欲法を用いて，アームの準最適組み合わせを求めている．アーム i の期待報酬密度に基づく評価値は，

$$\frac{\bar{\mu}_t(\tau,i)}{c_i}+\frac{b_t(\tau,i)}{c_i} \tag{23}$$

より求められる．ここで SW-KUBE の式 (19) を満たす最大となるアームの組み合わせの解を $M^*(B_t)=\{m_i^*,t\}$ とする．$\{m_{i,t}^*\}$ を用いて SW-KUBE はランダムにプレイする腕を選択する．プレイされるアームの確率は式 (24) に従う．

$$P(i(t)=i)=\frac{m_{i,t}^*}{\sum_{k=1}^{K}m_{k,t}^*} \tag{24}$$

プレイされた後，選択されたアームの評価値と残りの予算 B_t を更新する（steps 12-13）．

4. 評価実験

4.1 実験設定

提案手法である D-KUBE 及び SW-KUBE の評価実験の設定について述べる．本論文の実験設定は，Tran らの実験設定に従う [8]．アームの数 K は 100 とし，アームの報酬の確率分布は，切断正規分布を用いる．切断正規分布の平均，分散，及び定義域の設定について述べる．

- 平均 $\mu_i=[10,20]$
- 分散 $\sigma_i=\frac{\mu_i}{2}$
- 定義域 $[0,\ 2\mu_i]$

平均 μ は，与えられた [10，20] の範囲からランダムに選ばれる．分散及び定義域は平均値が与えられることで求められる．アームのコストについては，[1，10] の範囲からそれぞれのアームに対してランダムにコストを設定する．さらに本研究では，既存の問題設定である静的な報酬確率分布を含め以下のように Case 0，Case 1，及び Case 2 を設定する．

- Case 0：静的な報酬確率分布
- Case 1：アームごとに設定された間隔で再設定される動的な報酬確率分布
- Case 2：sin 関数に従う周期性のある動的な報酬確率分布

Case 0 は既存の問題設定と同様の設定を用いる．Case 1 はそれぞれのアームごとにランダムに変動期間を設定する．本評価実験において変動期間は [100，200] とした．アームはタイムステップが設定された変動期間を経過するごとに，切断正規分布の平均値をランダムに設定し直す．Case 2 はそれぞれのアームごとに角度 θ を設定する．θ は初期値として [0，359] からランダムに設定される．θ はタイムステップが経過すると同時に 1 ずつ増加する．$\theta \geq 360$ の時 θ は 0 に設定される．切断正規分布の平均値は，$15+5\sin\theta$ に従って変化する．

Case 0，Case 1 及び Case 2 を設定する理由を述べる．既存の問題設定である Case 0 を設定した理由として，既存手法をそのまま動的な報酬確率分布に適用した時の損失を確認するためである．また提案手法が動的な報酬確率分布にのみ最適化されてしまい，静的な報酬確率分布で損失を生まないか確認するためである．動的な報酬確率分布として，Case 1 及び Case 2 を設定した理由として，どのように報酬確率分布が変化するかによって損失率が変化するかを確認するためである．Case 1 では一定間隔でアームが再設定され変化する急な変動を想定している．Case 2 では徐々に報酬の確率分布が変化する緩やかな変動を想定している．

D-KUBE 及び SW-KUBE の比較対象として，既存の予算制限バンディットアルゴリズム KUBE を用いる．ここで各種バンディットアルゴリズムのパラメータについて述べる．D-KUBE の減衰率 γ 及び ξ の設定について述べる．D-KUBE の減衰率 γ は，既存の動的報酬バンディットアルゴリズムである D-UCB の問題設定をもとに設定する．D-UCB では，減衰率 γ を式 (25) に基づき設定していた．

$$\gamma = 1 - \frac{1}{4\sqrt{T}} \tag{25}$$

ここで T は総プレイ数を表す．動的報酬多腕バンディット問題ではタイムステップ t が T に到達した時に終了する．しかし予算制限多腕バンディット問題では，ラウンド数が予算及びアームのコストに依存する．従って，T を変更し，式 (4.2) 及び式 (4.3) に従って設定する．

$$\gamma = 1 - \frac{1}{4\sqrt{\frac{B}{c}}} \tag{26}$$

$$c = \frac{1}{K}\sum_{i=1}^{K} c_i \tag{27}$$

c はアームのコストの平均である．予算 B をコストの平均 c で割ることで T に近似させる．D-KUBE の ξ について，D-UCB と同様の値である 0.6 を設定した．

SW-KUBE の参照数 τ 及び ξ の設定について述べる．SW-KUBE の参照数 τ は，既存の動的報酬バンディットアルゴリズムである SW-UCB の問題設定をもとに設定する．

$$\tau = 4\sqrt{T\log T} \tag{28}$$

しかし D-KUBE と同様にラウンド数 T が予算及びアームのコストに依存する．従って T を変更し式 (29) 及び式 (30) に従って設定する．

$$\tau = 4\sqrt{\frac{B}{c}\log\frac{B}{c}} \tag{29}$$

$$c = \frac{1}{K}\sum_{i=1}^{K} c_i \tag{30}$$

SW-KUBE の ξ は，D-KUBE と同様に，0.6 に設定した．

4.2 実験結果

KUBE と D-KUBE 及び SW-KUBE を比較した結果について述べる．Case0，Case1 及び Case2 について KUBE と D-KUBE を比較した図が図 1，図 2 及び図 3，KUBE と SW-KUBE を比較した図が図 4，図 5 及び図 6 になる．図 1 及び図 2 から分かるように静的な報酬確率分布において，KUBE と提案手法の間に損失率の差は小さい．一方で，図 3，図 4，図 5 及び図 6 から分かるように動的な報酬確率分布において，提案手法の方が KUBE と比較して損失率が小さいことが分かる．また，Case 1 と Case 2 の比較により緩やかな変動のほうが損失率が大きいことが確認された．本実験結果の比較により，提案手法は静的な報酬確率分布においては損失率を大きくすることなく，動的な報酬確率分布にうまく適応できていることが分かった．

5. おわりに

本研究で提案した D-KUBE 及び SW-KUBE は既存手法である KUBE と比較し，静的な報酬確率分布では損失率をあまり増加せず，動的な報酬確率分布では損失率が小さくなり改善された．また，急な変化と緩やかな変化では，緩やかな変化の方が損失率が大きくなることが分かった．今後の課題として，損失率を小さくするためにアルゴリズムを改善することが挙げられる．また，本研究では人工的

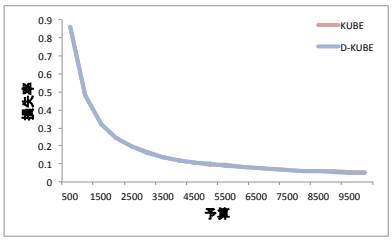

図 1 KUBE と D-KUBE の比較：Case 0

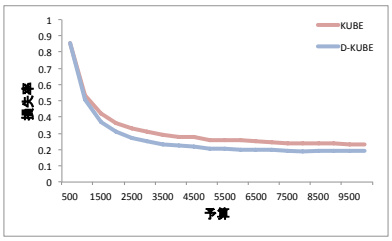

図 3 KUBE と D-KUBE の比較：Case 1

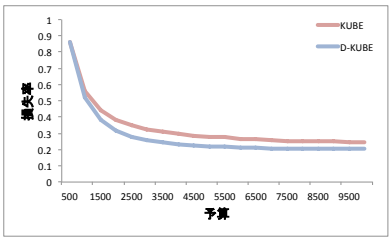

図 5 KUBE と D-KUBE の比較：Case 2

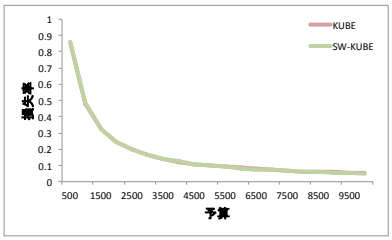

図 2 KUBE と SW-KUBE の比較：Case 0

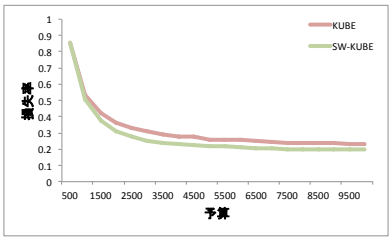

図 4 KUBE と SW-KUBE の比較：Case 1

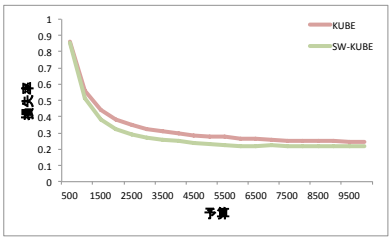

図 6 KUBE と SW-KUBE の比較：Case 2

図 7 既存手法 KUBE と提案手法 D-KUBE 及び SW-KUBE の比較

な実験設定を用いて，提案手法と既存手法の比較及び評価を行った．現実のデータなどを用いていないため，妥当性に欠ける点がある．従って，実験設定の妥当性の確保も課題としてあげられる．

参考文献

[1] Tran-Thanh, Long, et al. ”Efficient regret bounds for online bid optimisation in budget-limited sponsored search auctions. ” In, 30th Conference on Uncertainty in Artificial Intelligence, 2014.

[2] Tran-Thanh, Long, et al. ”Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks.” Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[3] Tran-Thanh, Long, et al. ”BudgetFix: budget limited crowdsourcing for interdependent task allocation with quality guarantees.” Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[4] Tran-Thanh, Long, Alex Rogers, and Nicholas R. Jennings. ” Long-term in- formation collection with energy harvesting wireless sensors: a multi-armed bandit based approach.” Autonomous Agents and Multi-Agent Systems 25.2 (2012): 352-394.

[5] 本橋永至, et al. ”状態空間モデルによるインターネット広告のクリック率予測.” オペレーションズ・リサーチ: 経営の科学=[O] perations research as a management science [r] esearch 57.10 (2012): 574-583.

[6] Thompson, W. R. ”On the likelihood that one unknown probability exceeds another in view of the evidence of two samples”, Biometrika, Vol. 25, No. 3/4, pp. 285294 (1933)

[7] Robbins, Herbert. ”Some aspects of the sequential design of experiments.” Bulletin of the American Mathematical Society 58 (1952), no. 5, 527–535.

[8] Tran-Thanh, Long, Chapman, Archie, Rogers, Alex and Jennings, Nicholas R. (2012) ”Knapsack based optimal policies for budget-limited multi-armed bandits.” In, Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12), Toronto, CA, 22 Jul 2012. , 1134-1140.

[9] Kocsis, Levente, and Csaba Szepesvári. ”Discounted ucb.” 2nd PASCAL Challenges Workshop. 2006.

[10] Aurélien Garivier and Eric Moulines. 2011. ”On upper-confidence bound policies for switching bandit problems.” In Proceedings of the 22nd international conference on Algorithmic learning theory (ALT’11), Jyrki Kivinen, Csaba Szepesvri, Esko Ukkonen, and Thomas Zeugmann (Eds.). Springer-Verlag, Berlin, Heidelberg, 174-188.