情報処理学会論文誌　数理モデル化と応用　Vol.6　No.3　11–22 (Dec. 2013)

ロックアップ期間による制約を考慮した確率的バンディット問題

小宮山 純平[1,a)]　佐藤 一誠[1,b)]　中川 裕志[1,c)]

受付日 2013年1月30日，再受付日 2013年3月21日，
採録日 2013年4月14日

概要：バンディット問題は，複数のアーム（選択肢）から最も報酬の高いものを探す問題であり，探索と活用のトレードオフの代表的なモデルの 1 つである．近年において，情報推薦，最適経路探索，最適化，モデル選択などの分野への応用を動機として，バンディット問題は機械学習やオペレーション・リサーチの分野において注目を浴びている．本研究はロックアップ期間（選択するアームを変更できない期間）の制約を考慮したバンディット問題を提案し，どのような方策をとればよいかを調べる．既存の多くの有益なアルゴリズムがロックアップ期間を含めた場合に自然に拡張可能であることを示し，その regret（性能）を評価する．この regret がロックアップ期間の最大の大きさに依存することを示す．さらに，ロックアップ期間が大きい場合に regret を減らすことができる Balancing and Recommendation（BaR）メタアルゴリズムを提案する．また，計算機実験の結果を示し，理論的な結果と比較し考察する．

キーワード：バンディット問題，多腕バンディット，逐次最適化，確率的最適化

Multi-armed Bandit Problem with Lock-up Periods

Junpei Komiyama[1,a)]　Issei Sato[1,b)]　Hiroshi Nakagawa[1,c)]

Received: January 30, 2013, Revised: March 21, 2013,
Accepted: April 14, 2013

Abstract: The multi-armed bandit problem, which is widely used to model sequential decision making, has recently attracted much attention. Motivated by actual applications, we propose a version of the multi-armed bandit problem in which the forecaster's choice is restricted. In this problem, rounds are divided into lock-up periods and the forecaster must select the same arm throughout a period. While there has been much works on finding optimal algorithms for the stochastic multi-armed bandit problem, their use under restricted conditions is not obvious. We extend the application ranges of these algorithms by proposing their natural conversion from ones for the stochastic bandit problem to ones for the multi-armed bandit problem with lock-up periods. We prove that the regret of the converted algorithms is $O(\log T + L_{max})$, where T is the total number of rounds and L_{max} is the maximum size of the lock-up periods. The regret is preferable, except for the case when the maximum size of the lock-up periods is large. For this cases, we propose a meta-algorithm that results in a smaller regret by using a empirical best arm for large periods. We empirically compare and discuss these algorithms.

Keywords: multi-armed bandit problem, sequential optimization, stochastic optimization

1. はじめに

多腕バンディット問題（バンディット問題）は，逐次最適化問題の最も代表的なモデルの 1 つである．この問題が最初に提唱されたのは 1950 年代まで遡り [1]，現在考えら

[1] 東京大学
The University of Tokyo, Bunkyo, Tokyo 113–0033, Japan
[a)] junpei_komiyama@mist.i.u-tokyo.ac.jp
[b)] sato@r.dl.itc.u-tokyo.ac.jp
[c)] nakagawa@dl.itc.u-tokyo.ac.jp

表 1　バンディット問題とその応用例の対応関係
Table 1　Applications of bandit problem.

	アーム	報酬	制約
A/B テスト	デザイン	ユーザの反応	設定の更新タイミング
診療	医療オプション	患者の経過	設備やオペレーション上の理由
工場	生産方式	生産量，歩留まり	設備投資上の理由
コグニティブ無線 [2]	チャンネル（周波数帯）	通信情報量	上位レイヤ（パケットなど）のサイズ

れている機械学習の問題としては最古参といってもよいであろう．近年において，その単純さと応用範囲の広さから再び注目が集まり，研究が加速している．

多腕バンディット問題という奇妙な単語の由来は，英語のバンディットマシン（アームを持つスロットマシン）である．いずれかのアームを選択すると，対応するマシンに応じた確率的な報酬が得られる．複数のアームの中から最良のものを選ぶのが問題の目的である．プレイヤがすべてのアームに関する完全情報を持っているなら，つねに期待報酬の最も高いものを選べばよい．しかし，アームの報酬がどうなっているかは最初には分からないので，逐次得られる報酬情報を通じて動的に最適なアームを探ってゆく必要が出てくる．バンディット問題におけるテーマは探索と活用のトレードオフ（*Exploration-Exploitation* tradeoff）である．現在の時点で最も良さそうなアームを活用することは短期的に大きい報酬をもたらす．しかし，一見最適でないアームを探索してゆくと，実はそちらのほうが最適であったという低確率で起こりうる事象を見逃さないようにでき，長期的な期待報酬の増加につながる．このような単純な問題設定であるが，近年における機械学習の隆盛に相まってバンディット問題の多くの応用・拡張が提案されている．例としてはモデル比較 [3], [4]，凸最適化 [5]，多クラス分類問題 [6], [7]，モンテカルロ木検索 [8] などがあげられる．

基本的なバンディット問題では，アームを毎ラウンド自由に選択できることを仮定している．しかし，現実におけるシステムでは，自由にアームを選択できない期間が発生することがしばしば起こりうる．たとえば，次のような例を考えてみるとよいであろう．

例 1.（A/B テスト）A/B テストは，ウェブサイトの新機能のテスト手法として一般的に使われている手法である．複数のウェブページの新デザイン差分を用意し，どれが最もユーザの反応が良いかを一部のユーザでサンプリング比較してサイトの更新の効果測定を行いたい．1 人のユーザに対して複数のデザインのうち 1 つだけしか見せることはできないという点が，バンディット問題におけるフィードバック設定に対応している．A/B テストの対象は，トップページのデザイン全体更新のような大きいものから，バナー広告の配置変更，画像の更新など多岐にわたる．大規模ウェブページにとって，訪問者数の増減は最も大きな関心の 1 つであり，その最適化は大きな意味を持つ．しかし，ウェブコンテンツの切替えには多くの制約が発生する．たとえば，バナー広告は広告契約者との都合から一定期間表示し続ける必要があるし，分散したサーバの設定を切り替えるときには，複数のサーバからの情報を集積し，データを更新する都合から，ある程度以上の頻度では切替えができないことが考えられる．

例 2.（診療）診療の問題は，バンディット問題の動機付けとして初期に提案され，近年に至るまで多くの応用がある [9], [10], [11]．考慮される複数の医療オプションの中から，患者の受ける利益（報酬）を最大化させるものを選ぶ問題は，バンディットの設定そのものである．多くの場合，医師のオペレーションや医薬品の在庫量などの都合から，同じ医療オプションを一定サイズ選択し，結果をもとに次のスケジュールを考えるということが考えられるが，このような制約下における最適なオプション選択がしたい．

これらの，バンディット問題の適用可能な例を**表 1** に示す．基本的なバンディット問題は逐次最適化問題であり，アームを毎ラウンド選択し，その中で選択を適応的に最適化することを目指す．しかし，現実における多くの問題では，アームの選択を変更することに制約がかかる．制約のかかったバンディット問題は，本質的には逐次選択問題とバッチ選択問題の中間と考えることができる．これらの問題における制限を，ロックアップという形でモデル化し，調べることが本研究の目的である．ロックアップという単語は，もとは金融で使われる単語で，株式や証券をその利害関係者が売却できない期間のことを指す．本稿では，バンディットアームが外部的な制約で変更できなくなることを表すためにこの単語を利用した．

本稿の構成は以下のようになる．

- 2，3 章において問題の定式化を行う．2 章で確率的バンディット問題について説明した後，3 章でロックアップ期間による制約を導入する．
- 確率的バンディットにおけるアルゴリズムは，アームの制約が入った場合にどのように振る舞えばよいかが自明ではない．4 章では，多くのアルゴリズムをロックアップによる制約があった場合に自然に拡張する方法を示す．拡張されたアルゴリズムの regret が $O(\log(T) + L_{max})$（T はラウンド数，L_{max} は最も長いロックアップ期間）になることを示す．

- 4 章で示した regret は L_{max} が小さい場合は最適だが，$L_{max} > \log T$ では前者の項が問題となる．5 章では，L_{max} が大きい場合に有効なメタアルゴリズムを提案する．
- 6 章では，これらの成果を数値的に検証するために行った計算機実験の結果について報告する．
- 最後に，7 章で本稿の内容についてまとめる．

2. バンディット問題

本研究では確率的バンディット問題を扱う．確率的バンディット問題では，報酬が一定の確率分布から選ばれるものと仮定する．アームの数を K とする．各アームは，正規化された区間 $[0,1]$ に値をとる定常の報酬確率分布 $\nu_1, ..., \nu_K$ を持つと仮定する．

バンディット問題の枠組みを**図 1** に示す．各ラウンド t において，予測者はアーム I_t を選択し，そのアームの確率分布から独立に引かれる報酬 $X(t)$ を受け取る．次のラウンドでは，これまでの選択と報酬を利用して，また新しいアームを選択していく．あるラウンド T に達したところで，この問題は終了する．予測者の目的はラウンド $1, ..., T$ を通した累積報酬の最大化であり，これがどの程度達成できているかは，以下で定義される regret（累積 regret）の期待値によって評価される．

$$\mathbf{R}[T] = \mu_* T - \sum_{i=1}^{K} \mu_i T_i(T). \tag{1}$$

ここで，μ_i はアーム i の報酬の期待値（$= \mathbb{E}[\nu_i]$）であり，$T_i(T)$ はラウンド T までにアーム i が選択された回数である．i^* を最も報酬の期待値が高いアーム，そして $\mu_* = \mu_{i^*}$ と定義する．また，i^* を最適なアーム，それ以外のアームを非最適なアームと呼ぶことにする．regret の意味は，最も確率の高いアームを選択し続けたときの期待報酬と，アルゴリズムが実際に選んだアームの期待報酬の差である．最適なアームの期待報酬と，アーム i の期待報酬の差を $\Delta_i = \mu_* - \mu_i$ と表記すると，regret はアルゴリズムが非最適なアーム i を選んだときに Δ_i だけ上昇する．この regret を最小化することがアルゴリズムにとっての目標となる．

2.1 バンディット問題のアルゴリズム

確率的バンディット問題において最も知られているアルゴリズムは，おそらく UCB [12], [13] であろう．UCB は各アームの信頼上界（Upper Confidence Bound）を推定し，それが最大となるアームを毎ラウンド選択する．より正確には，UCB アルゴリズムに従う予測者は，以下のようにアームを選択する．各ラウンド $t = 1, ..., T$ において，

1. これまでに一度も選択されていないアームが 1 つ以上あった場合，そのいずれかを選択する

初期条件：アームの数 K
各ラウンド $t = 1, ..., T$ において，
(1) 予測者はアーム I_t を選択する
(2) 予測者は報酬 $X(t) \sim \nu_{I_t}$ を受け取る

図 1　確率的バンディット問題
Fig. 1　Stochastic bandit problem.

2. すべてのアームが一度以上選択されている場合，次の UCB Index を最大化するアームを選択する

$$\hat{X}_i(t-1) + \sqrt{\frac{a \log t}{T_i(t-1)}} \tag{2}$$

ここで，$\hat{X}_i(t-1)$ はラウンド $t-1$ の終了時点での経験期待値（アーム i を引いたときに得た報酬の平均）であり，a はパラメータ定数である．

探索と活用の言葉で述べると，最初の項（経験期待値）は活用，2 番目の項は未知の真の期待値の経験期待値からのずれの上界を見積もるという意味で探索に対応している．パラメータ a がこの探索と活用のバランスを決めており，a が大きくなればなるほど探索の割合が大きくなる．活用を大きくとると非最適なアームを探索するラウンド数を減らすことができる．しかし，低確率で最適なアームの経験期待値が低い事象が起きることがあり，その場合に非最適なアームを最適であると誤認してしまうリスクがある．一方，探索を大きくとると非最適なアームを多く探索しなければならないが，多くのアームを均等に探索するため，真の最適なアームが実際に経験期待値が最大である可能性を高めることができる．これが，バンディット問題における探索と活用のトレードオフである．初期に提唱された UCB1 [12] は，上記の UCB において $a = 2$ としたものであり，これは保守的な値（探索を多めに見積もったもの）である．確率分布の分散に応じてこの値は小さくすることができる．実用的な問題にバンディットのアルゴリズムを応用する場合には，トレーニングセットなどで最適な a の値を探し，その値を利用するということがしばしば行われている．

UCB は，その単純な定式（信頼上界）と理論的な扱いやすさから，多くの派生アルゴリズムがある．本研究では UCB-E [14]，UCB-V [15]，KL-UCB [16]，MOSS [17]，UCB-Tuned [12] などのアルゴリズムを実験的に比較したが，これらは UCB と同じく信頼上界を利用している．

UCB は決定的アルゴリズム（次に選択するアームがこれまでの報酬から一意に定まる）であるが，一方で，確率的な選択を入れた ϵ_n-greedy アルゴリズムについても紹介する．ϵ_n-greedy アルゴリズムに従う予測者は，各ラウンド $t = 1, ..., T$ において，

1. 乱択確率を $\epsilon_t = \min\{1, cK/d^2 t\}$ とおく．ここで，c と d はアルゴリズムのパラメータである．

2a. 確率 $1 - \epsilon_t$ で，これまで選択されたアームの中で最

も経験期待値の高いアーム $\arg\max_i \hat{X}_i(t-1)$ を選択する．

2b. 確率 ϵ_t で，ランダムにすべてのアームを等確率で選択する．

これらのアルゴリズムはすべて regret の期待値が $O(\log T)$ であり，これは定数倍のオーダまで最適であることが知られている．$\sum 1/t = \log(t)$ であるので，1 ラウンドあたりの非最適アームの探索は最低でも $1/t$ オーダで必要であるということである．探索をこれ以上少なくしてしまうと，経験期待値の低いアームが実は最適なアームであった確率が大きくなってしまう．定数部まで最適なアルゴリズムについては文献 [13], [16], [18], [19] などを参照されたい．ロックアップ制約の導入によってアームが自由に選択できなくなったとき，この探索と活用のバランスをどの程度保てるかが本研究のテーマである．

2.2　先行研究

本稿の提案するロックアップ制約のある確率的バンディット問題を説明する前に，確率的バンディット問題における制約を扱った先行研究（**表 2**）を概説する．最も多く研究されてきたモデルは，スイッチコストを持つバンディット問題である．これはアームの切替えに一定のコストがかかる設定で，アーム選択の変更回数を抑えた最適なアルゴリズムを探すものである．より詳細には，文献 [20], [21], [22] などを参照されたい．また，広告などへの応用を動機として，休眠のあるバンディット問題 [23] が提案されている．これは，ラウンドごとに選択肢の幅が変わる問題であり，掲載可能な広告のリストが動的に変わっていくことをモデル化したものである．コミットのあるバンディット問題 [24] は，ある期間まで自由にアームを選択をすることができ，探索が十分だと予測者が判断したら，その時点であるアームにコミットするというものである．予測者の regret は探索の期間の長さと，コミットしたアームが最適なアームかどうかに依存する．

スイッチコストやコミットのあるバンディット問題のモデルと，本稿で提案するロックアップ制約のあるバンディット問題の一番の違いは，制約を含めた最適化か，もしくは制約が与えられた下での最適化かであろう．スイッチコストの問題の場合はスイッチコストを払うことによってアームを変更できるし，コミットのあるバンディット問題の場合は探索期間を予測者の判断で伸ばすことができる．つまり，制約期間を含めた最適化の問題であると考えることができる．一方，ロックアップ制約では，外部からアームを変更できない期間という制約が与えられ，ロックアップ中はアームを変更できないというものになっている．そういう意味で，制約が与えられた条件での最適化の問題であると考えることができる．

表 2　バンディット問題の関連研究
Table 2　Related problems.

問題	制約
スイッチコスト [20], [21], [22]	アームの切替えにコスト
休眠 [23]	選択可能なアームの集合に制限
コミット [24]	コミット後にアームの切替え不可
ロックアップ（本稿）	ロックアップ期間中にアームの切替え不可

3.　ロックアップ期間による制約のあるバンディット問題

2 章で定式化した確率的バンディット問題に，ロックアップ期間による制約を導入する（**図 2**）．ラウンド $1,...,T$ が，N 個のロックアップ期間に分解されるとする．それぞれのロックアップ期間の長さを $L_1,...,L_N$ とする．各期間の初めと終わりのラウンドを $(s_1,f_1),...,(s_N,f_N)$ と表記する．ラウンドを分割しているため，$s_1=1$，$f_n+1=s_{n+1}$，$f_N=T$ である．また，括弧を使ってサイズ順にソートされたロックアップ期間の番号を表記する．つまり，$L_{(1)}$ は $\max_n L_n$，$L_{(2)}$ は 2 番目に大きなロックアップ期間の長さ，以下 3 以降も同様とする．また，$L_{(1)}$ を L_{max} とも表記する．ロックアップ期間による制約を考慮したバンディット問題は**図 3** のように進行する．基本となる確率的バンディット問題では各ラウンドに予測者が自由にアームを選

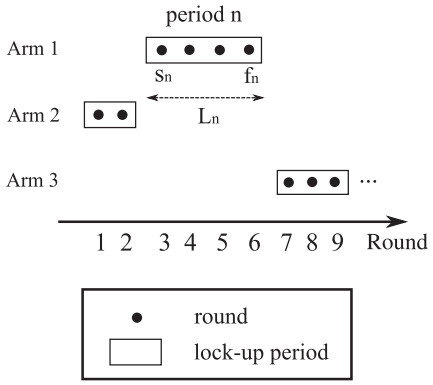


図 2　ロックアップ制約．黒い点がラウンド，四角がロックアップ期間を表す
Fig. 2　Lock-up bandit. A black dots represent rounds and rectangles represent lock-up periods.

初期条件：アームの数 K，ラウンド数 T，ロックアップ期間の長さ $L_1,...,L_N$
各ラウンド $t=1,...,T$ において，
(1) そのラウンドがいずれかのロックアップ期間の開始ラウンドなら（$\exists n s_n = t$），予測者はアーム I_t を選択する．そうでないなら，予測者は直前のラウンドに選択したものと同じアーム I_t を選択する．
(2) 予測者は報酬 $X(t) \sim \nu_{I_t}$ を受け取る

図 3　ロックアップ期間による制約のあるバンディット問題
Fig. 3　Bandit problem with lock-up periods.

択できていたのに対して，ロックアップのあるバンディット問題では各ロックアップ期間の最初のみアームを選択することができ，ロックアップ期間中は前のラウンドと同じアームを選択しなければならない．アルゴリズムの評価手法である regret については，通常のバンディット問題と同様である．

3.1　ラウンドとロックアップ期間の表記について

本稿では，ラウンド数を表す添字として $t \in \{1, ..., T\}$，ロックアップ期間を表す添字として $n \in \{1, ..., N\}$ を使用する．また，添字 $i \in \{1, ..., K\}$ をアームの番号を表すために使用する．例として，アーム i がラウンド T までに選択された回数 $T_i(T)$ は以下のように表される．

$$T_i(T) = \sum_{t=1}^{T} \mathbb{I}_{I_t=i}. \tag{3}$$

ここで，$\mathbb{I}_{\mathcal{E}}$ は事象 $\mathcal{E}$ が真なら 1，偽なら 0 である指示関数とする．

予測者は各ロックアップ期間 n の最初にアームを選択し，期間中はアームを変更できず単一のアームを選び続ける．そのため，このアームを I_n と定義することができる．また，T_i は以下のように表すこともできる．

$$T_i(L_1, ..., L_N) = \sum_{n=1}^{N} L_n \mathbb{I}_{I_n=i}. \tag{4}$$

このことと regret の定義から，各ロックアップ期間の最初に非最適なアーム i を選択すると，そのロックアップ期間に比例した大きさの regret（$= L_n\Delta_i$）を被ることが分かる．

3.2　ラウンド数 T およびロックアップ期間の長さ $L_1, ..., L_N$ がアルゴリズムに伝えられるかどうか

通常のバンディット問題（図 1）では，アルゴリズムがラウンド数 T を知らないことを仮定している．本研究では，ロックアップ期間という制約を導入してその上のアルゴリズムを考えるが，

1. 4 章で提案する既存のアルゴリズムのロックアップへの拡張は，ラウンド数 T および各ロックアップ期間の長さが与えられない（毎ラウンド終了時に，現在のロックアップ期間および全ラウンドが終了したかどうかが与えられる）状況でも動作する．一方で，
2. 5 章で提案する BaR メタアルゴリズムはラウンド数 T および各ロックアップ期間の長さがアルゴリズムの開始時に既知であることを前提としている．

また，T が既知の場合，終盤で探索と活用のトレードオフを崩し，主に活用をとることより，T ラウンドまでの regret の期待値を下げるための最適化をすることができるが，本研究ではこのような最適化については考慮していない．

3.3　ロックアップ期間の順番

ロックアップのあるバンディット問題にとって，ロックアップ期間の順番は本質的に重要である．例として，2 つの問題を考えてみよう．ロックアップ期間の数はどちらも $N = 10$ とする．1 つ目の問題では $L_1, ..., L_9 = 1, .., 1$，$L_{10} = 10$ であり，2 つ目の問題では $L_1 = 10$，$L_2, ..., L_{10} = 1, ..., 1$ であるとする．このとき，1 つ目の問題のほうが 2 つ目の問題より regret を小さくできる．これは，1 つ目の問題では 10 番目のロックアップ期間に選択するアームをそれ以前の報酬情報に基づいて選択できるのに対し，2 つ目の問題では，最初のロックアップ期間に選択するアームを無情報で選択しなければならないため，最低でも $(K-1)/K$ の確率で大きな（10 ラウンド分の）regret を被るためである．

4.　既存のアルゴリズムのロックアップ制約を考慮した拡張

2.1 節において，確率的バンディット問題とそのアルゴリズムを説明した．これらのアルゴリズムは毎ラウンド自由にアームを選択できることを前提としており，アームの選択に制限が加えられたときにどのように振る舞えばいいかが自明ではない．本章では，UCB や ϵ_n-greedy をはじめとした多くのアルゴリズムの，ロックアップがあるバンディット問題への自然な拡張を紹介する．これらのアルゴリズムは，

1. 毎ラウンド，アームごとの目的関数を最大化するアームを選択する．ここで，
2. 目的関数はそれぞれのアーム i を選択した数 T_i，経験期待値 $\hat{X}_i$，経験分散 $\hat{V}_i$ など，アームごとに独立した量の関数である，

という特徴を持つ．たとえば，UCB の目的関数（UCB index）は $T_i(t-1)$ と $\hat{X}_i(t-1)$ の関数である．また，ϵ_n-greedy は確率 1-ϵ で経験期待値 $\hat{X}_i(t-1)$ を最大化するアームを選び，確率 ϵ でランダムにアームを選択する．ロックアップを考慮した場合は，アームを選択できないラウンドが存在するが，これらのアルゴリズムは以下のようにして自然に拡張できる．各ラウンドにおいて，もしアームを選択できる場合，これまでと同様にアームごとの目的関数を最大化するアームを選択し，アルゴリズムの内部状態（i.e., アーム選択数，経験期待値など）を更新する．そうでない場合も，同様にアルゴリズム内部状態の更新を行う．この方法によってロックアップ制約のあるバンディット問題に拡張されたアルゴリズムを，′ を末尾につけて記述する（例：UCB → UCB′）．注意すべき点として，この方法では自然に拡張できないアルゴリズムも存在する．たとえば，アームの候補リストを管理していてその中から順番に選ぶようなアルゴリズムはこの方法では拡張できないため，制約に対応した別の自明でない拡張方法が必要である．

元となるアルゴリズムは，確率的バンディットの枠組みにおいて最適な探索と活用のバランス（i.e., $O(\log T)$ の regret）を実現していた．ロックアップ制約のあるバンディットにアルゴリズムを拡張した場合の UCB′ と ϵ_n-greedy′ の regret について，以下の定理が成立する．

定理 4.1.（UCB′ の regret 上界）ロックアップ制約ありのバンディット問題において，探索の大きさを決めるパラメータを $a=2$ に設定した *UCB′* に従う予測者の *regret* の期待値について，以下が成立する．

$$\mathbb{E}[\mathbf{R}(L_1,...,L_N)] \leq \sum_{i \neq i^*} \left\{ \frac{8\log T}{\Delta_i} + L_{max}\Delta_i \left(1+\frac{\pi^2}{3}\right) \right\}. \tag{5}$$

証明概略：この証明は文献 [12] の定理 1 をロックアップ制約のある場合に拡張したものである．$L_{max}=1$ のとき，文献 [12] の定理 1 と一致する．ベースとなる証明は，非最適なアーム i が選ばれる回数が $T_i(t) \geq \lceil (8\log T)/\Delta_i^2 \rceil$ を超えると，非常に低確率でしかアームが選ばれないことに依存している．ロックアップ制約を含めた拡張にした場合の変更は，

(1) $(8\log T)/\Delta_i^2$ の代わりに $(8\log T)/\Delta_i^2 + (L_{max}-1)$ を上界としている．これは，アームの選択数が $T_i(t) \geq (8\log T)/\Delta_i^2$ を超えたときの $T_i(t)$ がたかだかこの値だからである．

(2) $T_i(t) \geq \lceil (8\log T)/\Delta_i^2 \rceil$ を超えた場合のアーム選択数の上界である $\pi^2/3$ を，L_{max} 倍で抑えている．

定理 4.2.（ϵ_n-greedy′ の regret 上界）*ϵ_n-greedy′* をパラメータ $0 < d \leq \Delta$ で実行した場合，十分大きな c と t に対して，非最適なアーム i がラウンド t に選択される確率は，以下のように上から抑えられる．

$$\frac{cK}{d^2 t} + o(1/t). \tag{6}$$

十分大きなラウンド t において，非最適なアームを選択する確率が $O(1/t)$ で抑えられるので，漸近的な regret は $O(\log T)$ となる．

証明概略：ϵ_n-greedy′ は，確率 ϵ でランダムにアームを選択するポリシである（2.1 節参照）．このランダムな選択においてアーム i が選ばれる回数を T_i^R とおく．今回の証明において重点となるのは，T_i^R によって得られる報酬情報から，非最適なアームが選ばれる回数を十分うまくバウンドできることを示すことである．バーンスタインの不等式より，T_i^R の平均と分散から，高確率でのバウンドを得ることができる．ベースとなる証明（文献 [12] の定理 3）と同様にこの平均と分散を得るのが鍵となるが，ロックアップ制限がある場合には分散がたかだか L_{max}^2 倍で抑えられる．

定理 4.1 によって，任意のロックアップ期間 $L_1,...,L_N, \sum_n L_n = T$ に対して，UCB′ の regret は $O(\log T + L_{max})$ で抑えられることが分かる．直観的には，元のアルゴリズムの探索と活用のバランスが，L_{max} に比例したぶんだけ最適な値からずれていると解釈が可能である．L_{max} が T と比べて十分小さいとき，UCB′ と ϵ_n-greedy′ は $O(\log T)$ の regret を達成することができる．確率的バンディット問題において，$O(\log T)$ のバウンドは定数倍まで最適であるので，L_{max} が十分小さいロックアップ問題に対しては，ロックアップの制約が入ってもほぼ同じ量でバウンドできると考えることができるであろう．しかし，$\log T$ と比較して大きいサイズのロックアップ期間があった場合，この regret は L_{max} の項が支配的となる．次章では，大きいサイズのロックアップがあった場合に regret を減らすための方策について説明する．

5. BaR メタアルゴリズム

大きなロックアップ期間がある場合，その期間の最初で非最適なアームを選んでしまうと期間のサイズに比例する regret を被ることになる．この問題に対処するため，本章では BaR（Balance and Recommendation）メタアルゴリズムを提案する．5.2 節では，純探索問題というバンディット問題の派生問題を考えるにあたって定義された単純 regret の概念について説明する．5.3 節では，ロックアップ制約のあるバンディット問題における BaR アルゴリズムを提案する．また，5.4 節では，単純 regret が実際に知られているアルゴリズムである UCB-E について説明する．本章全体での目的は，ロックアップ制約のあるバンディット問題における regret を減少させることである．

5.1 ロックアップ制約と regret

ロックアップ期間中，アルゴリズムはアームの選択を変えることができない．そのため，n 番目のロックアップ期間の開始時に非最適なアーム i を選択してしまったとすると regret は $\Delta_i L_n$ 増加することになる．ベースとなるアルゴリズムが探索と活用のバランスを最適にしていても，ロックアップ期間で急激に探索を増やすことになってしまい，バランスが崩れてしまうことなる．この理由から，非最適なアームを長いロックアップ期間の初めに選択するのを防止したい．

5.2 純探索問題と単純 regret の最小化

バンディット問題はラウンド $1,...,T$ において，全体として最適なアームを高確率で選ぶことを目標とする問題である．一方，ある特定のラウンドにおいてどのくらいの確率で最適なアームを選ぶことができるのかということは，バンディット問題そのものでは考慮されてこなかった経緯がある．この後者の問題について専門に研究した論文として文献 [25] がある．この論文において，Bubeck らは純探索バンディット問題（純探索問題，**図 4**）という派生問題を提案した．この問題の設定は確率的バンディット問題と

初期条件：アームの数 K
各ラウンド $t \in \{1, ...\}$ において，
(1) 予測者はアーム I_t を選択する
(2) 予測者は報酬 $X(t) \sim \nu_{I_t}$ を受け取る
(3) 予測者は推薦アームを選択する
ゲームの終了ラウンドにおいて，推薦アームの性能は以下の単純 regret によって評価される

$$r(t) = \mu_* - \mu_{\psi_t}. \tag{9}$$

図 4　純探索バンディット問題 [25]
Fig. 4　Pure-exploration bandit problem [25].

同様であるが，各ラウンドの終了時に，選択したアームとは別に推薦アームを出力することを要求される．ゲームが終了したラウンドにおいて，推薦したアームの 1 ラウンドだけの regret（単純 regret）を最小化するのが純探索問題の目的である．2 章において定義した通常の regret は，単純 regret との区別のためにこれ以降累積 regret と呼ぶことにする．累積 regret と単純 regret の間には，以下の漸近的なトレードオフがある．

結果 5.1.（累積 regret と単純 regret のトレードオフ [25]）あるアルゴリズムに従ってアームを選択することを考える．このとき，このアルゴリズムの累積 *regret* と，このアルゴリズムに付随して推薦アームを選ぶ任意の方法の単純 *regret* の間に以下の関係が成立する．ある関数 ξ に対して，定数 C が存在し，すべてのベルヌーイ分布であるアームのリスト $\{\nu_1, ..., \nu_K\}$ に対して

$$\mathbb{E}[\mathbf{R}(T)] \leq C\xi(T), \tag{7}$$

を満たすとする．このとき，ラウンド T での単純 *regret* は以下の下界を持つ．すべての $K \geq 3$ 以上のそれぞれ異なるベルヌーイ分布に対して，定数 D とアーム列の順序 $\{\nu_1, ..., \nu_K\}$ が存在し，

$$\mathbb{E}[\mathbf{r}(T)] \geq \frac{\Delta}{2} \exp\left(-D\xi(T)\right), \tag{8}$$

を満たす．ここで，$\Delta = \min_{i \neq i^*} \Delta_i$ である．

上の定理は直観的に以下の意味を持つ．つまり，あるラウンド T での単純 regret を下げるためには，そこまでのラウンド $\{1, .., T\}$ で一定量の探索をする（累積 regret を持つ）必要がある．ラウンド T での r（単純 regret）とそれまでの R（累積 regret）には $r \sim \exp(-DR)$（係数 D は分布に依存）の関係が存在するため，通常のバンディットアルゴリズムでは，単純 regret が多項式オーダ，累積 regret が log オーダになる．

実際に単純 regret を最小化する方法として，経験期待値最大のアーム（Empirical Best Arm），最も頻繁に選択したアーム（Most Played Arm），そしてこれまでの選択に基づいてアームを確率的に選択する方法（Empirical Distributions of Plays）の 3 つが提案されている．Bubeck ら [25] による実験では経験期待値最大のアームが最も高性能であった．本稿では，経験期待値最大のアームを推薦アームとして使用する．これ以降で推薦アームについて言及した場合，つねに経験期待値最大のアームを選ぶ方法を指すものとする．探索と活用の言葉では，推薦アームは活用のみをとるための方法である．

初期条件：アームの数 K，ロックアップ期間の長さ $L_1, ..., L_N$，ロックアップ期間のしきい値 L_t，ベースアルゴリズム $\mathcal{A}$
各ロックアップ期間 $n = 1, ..., N$ の開始時に，
(1) ロックアップ期間 n の大きさ L_n が L_t 以上の（$L_n \geq L_t$）場合，$\mathcal{A}$ から推薦アームを受け取り，それをこの期間の選択アーム I_n とする．この期間における報酬は $\mathcal{A}$ にフィードバックしない．
(2) そうでない場合，$\mathcal{A}$ にアームを選択させ，それをこの期間の選択アーム I_n とする．この期間における報酬を毎ラウンド $\mathcal{A}$ にフィードバックする．

図 5　BaR メタアルゴリズム
Fig. 5　BaR metaalgorithm.

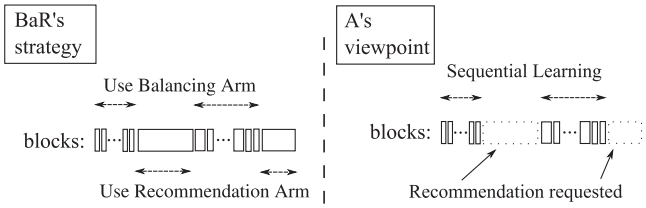


図 6　BaR とベースアルゴリズム $\mathcal{A}$ の関係．この図では 2 つの大きいロックアップ期間において推薦アームが使われている
Fig. 6　Relation between BaR and Base algorithm $\mathcal{A}$.

5.3　BaR メタアルゴリズム

5.1 節で述べたように，大きいロックアップ期間で非最適なアームを選んでしまうことによって regret が増加するため，これを避けるのがロックアップ期間のあるバンディット問題において重要となる．今回提案する BaR（**図 5**）は，大きいロックアップ期間において推薦アームを利用し，それ以外の期間についてはベースとなるバンディットアルゴリズムを実行する（**図 6**）．ベースとなる（ロックアップ制約のあるバンディット問題の）アルゴリズムを $\mathcal{A}$ と表記する．また，ベースアルゴリズムが $\mathcal{A}$ の BaR を $[\mathrm{BaR}, \mathcal{A}]$ と表記する．BaR の開始時に，推薦アームを利用するかどうかのしきい値 L_t を決める．各ロックアップ期間の開始時に，そのロックアップ期間の長さ L_n が L_t 以上の場合，アルゴリズムは $\mathcal{A}$ に推薦アームを問い合わせ，そのアームを選択とする．ロックアップ期間の間に得られた報酬の情報について，$\mathcal{A}$ へのフィードバックは行わない．そうでない場合は，BaR は $\mathcal{A}$ に透過的に振る舞う（つまり $\mathcal{A}$ に選択するアームを問い合わせ，結果として得られた報酬の情報を $\mathcal{A}$ にフィードバックする）．推薦アームを利用する期間の集合を，推薦セットと定義する．推薦セットに含まれる期間は L_t より大きいすべての期間なので，その数を N_r

とすると $(1),...,(N_r)$（ロックアップ期間を大きさの降順にソートした最初の N_r 個の期間の番号のリスト）である．

[BaR, $\mathcal{A}$] の regret は，$\mathcal{A}$ の単純 regret と累積 regret を用いて次のように表せる．

$$\mathbf{R}(L_1,...,L_N) = \mathbf{R}_{base}(L_1,...,L_N \setminus L_{(1)},...,L_{(N_r)}) + \sum_{n=1}^{N_r} L_{(n)}\mathbf{r}_{base}([L_{n'}|n' < (n), L_{n'} \notin \{L_{(1)},...,L_{(N_r)}\}]), \tag{10}$$

ここで，右辺の第 1 項（累積 regret 部分，$\mathbf{R}_{base}$）を，ベースアルゴリズム $\mathcal{A}$ を現在のロックアップ制約の問題から推薦アームを利用するラウンドを除いたバンディット問題で走らせた問題の regret として定義する．また，右辺の第 2 項（単純 regret 部分，$\mathbf{r}_{base}$）は，推薦セットのそれぞれに対してのベースアルゴリズムの単純 regret の和として定義する．ただし，ここでの単純 regret は累積 regret と同様に推薦セットを除いた環境での和である．例として以下の設定を考える．ロックアップ期間の数を $N = 100$，推薦セットを 50 番目と 100 番目のロックアップ期間とする．この場合は累積 regret 部分はベースアルゴリズム $\mathcal{A}$ を，50 番目と 100 番目のロックアップ期間を除いた残りのロックアップ期間で走らせたときの regret であり，単純 regret 部分は，ベースアルゴリズムの，1 から 49 までのロックアップ期間の後での 50 番目のロックアップ期間での単純 regret と 1 から 49，51 から 99 番目のロックアップ期間の後での 100 番目のロックアップ期間での単純 regret の和である．

このように，BaR メタアルゴリズムは regret をベースアルゴリズムの累積 regret と単純 regret の和に分解する．累積 regret 部分から任意のロックアップ期間を取り除くことができ（推薦アームを取り除いたバンディット問題では，L_{max} を減らすことができる），またそのロックアップ期間では最も最適なアームを高確率で選択するため，大きいロックアップ期間で regret が増加するリスクを大きく減らすことができる．

次の関心は，単純 regret と累積 regret をどのように見積もればよいかである．4 章において，我々は UCB′ と ϵ_n-greedy′ の累積 regret の上界を導出したが，単純 regret についてはまだ示していない．実際，単純 regret が注目されたのは比較的最近なので，アルゴリズムの単純 regret に関してはまだ知られていないことが多いのが現状である．次節では，累積 regret と単純 regret の両方が導出可能な例である UCB-E について紹介し，それらの regret を考察する．

5.4 UCB-E

UCB-E は文献 [14] によって最初に導入された UCB の派生アルゴリズムである．文献 [14] で考えられていた問題は，その目標がバンディット問題における目標（総報酬の最大化問題）と異なるものであったが，探索のパラメータを調整することによって，このアルゴリズムはバンディット問題にも使用することができる．UCB がアームの探索部分の上界を $\sqrt{a\log t/T_i(t-1)}$ とおくのに対し，UCB-E は $\sqrt{a/T_i(t-1)}$ とおく（つまり，ラウンドごとに分子を変更せず固定している）．実際，ラウンド数 T が既知の場合，パラメータ $a = 2\log T$ とおけば累積 regret について UCB の上界をまったく同じように適用できる．また，経験的にも，UCB とパラメータ $a = 2\log T$ の UCB-E はほぼ同じように振る舞う．加えて，UCB-E は単純 regret の上界が文献 [14] によって証明されている．この証明はロックアップ期間がある場合に拡張可能であり，以下にその結果を述べる．

定理 5.1.（UCB-E′ の累積 regret の上界）*UCB-E′* がパラメータ $a \geq 2\log T$ で実行された場合，

$$\mathbb{E}[\mathbf{R}(L_1,...,L_N)] \leq \sum_{i\neq i^*}\left\{\frac{4a}{\Delta_i} + \Delta_i L_{max}\left(1+\frac{\pi^2}{3}\right)\right\}. \tag{11}$$

が成立する．

証明概略：証明は定理 4.1 と同様である．

定理 5.2.（UCB-E′ の単純 regret の上界）*UCB-E′* がパラメータ $0 < a \leq 25(T - KL_{max})/(36H_1)$ で実行された場合，その単純 *regret* は以下のように抑えられる．

$$\mathbb{E}[\mathbf{r}(L_1,...,L_N)] \leq 2TK\exp\left(-\frac{2a}{25}\right), \tag{12}$$

ここで，$\Delta = min_{i\neq i^*}\Delta_i$ と定義すると，$H_1 = \sum_{i\neq i^*} 1/\Delta_i^2 + 1/\Delta^2$ である．

証明概略：証明は，ゲーム全体を通して，高確率で信頼区間 $1/5\sqrt{a/T_i(t-1)}$ から外れないという事実がベースとなる．この証明方法は $a \leq 25(T - KL_{max})/(36H_1)$ を満たせばロックアップ制約の存在下でも成立する．

5.5 推薦アームの情報を捨てる理由

BaR では，推薦アーム使用時の報酬情報をベースアルゴリズム $\mathcal{A}$ にフィードバックしていない．一見，これは不合理に見える．なぜなら，報酬はアルゴリズムが推薦アームを使用するにせよしないにせよ同じ確率分布から引かれるため，報酬情報を教えることは基本的にプラスになると考えられるからである．BaR によって報酬情報をベースアルゴリズムにフィードバックしなかった理由は，そうすることによってアルゴリズムの regret を累積 regret と単純 regret に分割でき，解析をしやすくするためである．もう 1 つの理由としては，実際にフィードバックを与えないほうが良くなる自明なケースを作ることが可能である点がある．

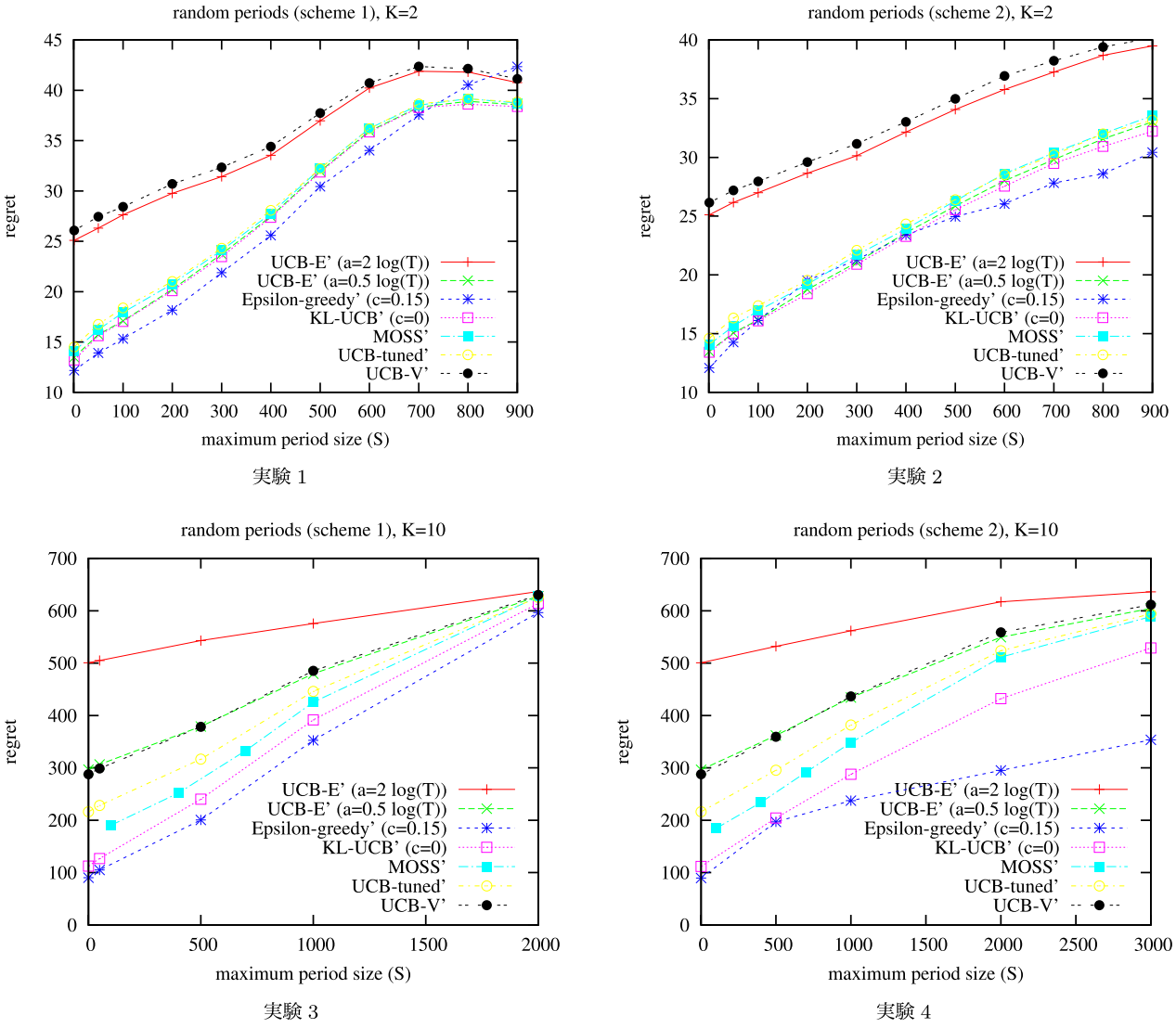


図 7　1 つ目の実験セット．ランダムに生成されるロックアップ期間の最大サイズ S と regret の関係を示す

Fig. 7　First set of experiments. Cumulative regret as a function of maximum lock-up size S.

6. 計算機実験

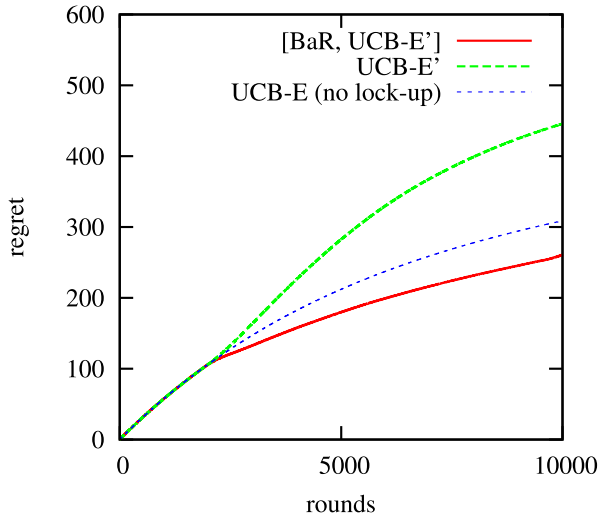
提案したモデルおよびアルゴリズムについて，その性質を見るため，計算機による実験を行った．実験は目的ごとに大きく 2 つのセットに分かれる．

- 1 つ目の実験セット（**図 7**（実験 1-4））は，4 章で提案された従来のアルゴリズムの拡張が，ロックアップ制約を導入したバンディット問題に対してどのようなパフォーマンスを見せるかということについて調べた．
- 2 つ目の実験セット（**図 8**（実験 5-8））は，5 章で提案されたメタアルゴリズム（BaR）についての事前・事後分析を行った．

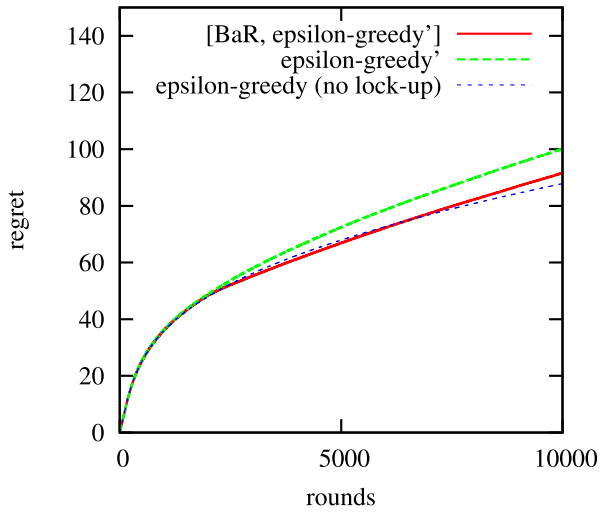
最初に実験設定について説明した後，実験の結果について考察する．注意すべき点として，複数のアルゴリズムの間の優劣を比較することは今回の目的ではない．いくつかのアルゴリズムはパラメータを持っており，それらのアルゴリズムのパフォーマンスはパラメータに大きく依存する．ϵ_n-greedy は今回の実験において最も regret が少なかったが，これは文献 [12] の 4 章において報告された，経験的に良い値を使用しているためである．また，この論文において ϵ_n-greedy のパフォーマンスは最適なパラメータから外れると急激に落ちることが指摘されている．UCB などの信頼上界を利用するアルゴリズムと異なり，ϵ_n-greedy は regret の理論的な保障の範囲が限定されていることや，パラメータの推定が困難である理由などから実用的には使いにくい．事実，バンディットの研究において，ϵ_n-greedy は UCB より利用されているケースが少ない．

6.1 実験設定

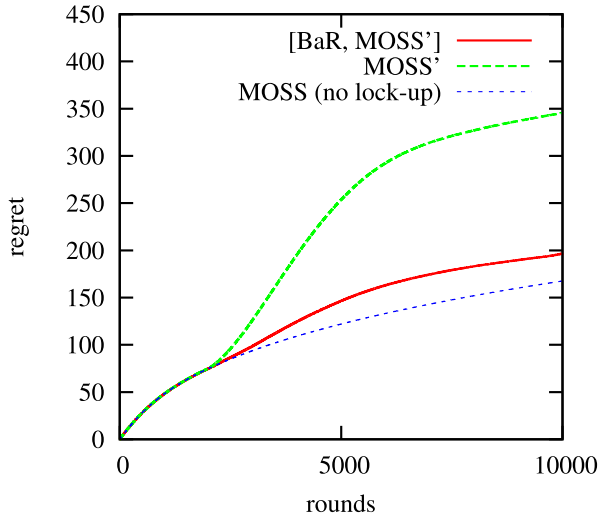
実験のすべてのアームの報酬分布はすべてベルヌーイ分



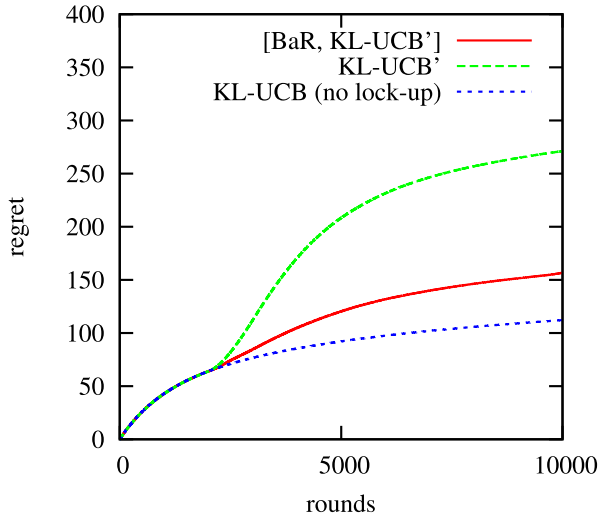
実験 5



実験 6



実験 7



実験 8

図 8　2 つ目の実験セット．ラウンド $T > 2000$ でランダムにロックアップ期間を生成した場合のラウンド数と regret の関係を示す．それぞれのグラフは同じアルゴリズム（BaR 有・無）とさらにロックアップのない問題での regret を比較している．regret は 10,000 回の計算機実験の平均であるため，滑らかな曲線を描いている

Fig. 8　Second set of experiments. Regret as a function of rounds. Lock-up periods are randomly generated after $T = 2000$.

布とした（実験の性質そのものはベルヌーイ分布に依存しないため，任意の $[0,1]$ に値をとる確率分布で同様の実験を行うことは可能である）．

実験 1，2 は 2 アームのバンディット問題であり，それぞれのアームの期待値は $(\mu_1, \mu_2) = (0.55, 0.45)$ とした．また，ラウンド数 $T = 1000$ とした．それ以外の実験（実験 3-8）は 10 アームのバンディット問題であり，それぞれのアームの期待値は $(\mu_1, ..., \mu_{10}) = (0.1, 0.05, 0.05, 0.05, 0.02, 0.02, 0.02, 0.01, 0.01, 0.01)$ とした．また，ラウンド数 $T = 10000$ とした．これらの実験の設定は，文献 [12] と文献 [16] を参考にした．

ベースとなるアルゴリズムは UCB-E [12]（パラメータ $a = 2\log T, 1/2\log T$），ϵ_n-greedy [12]（パラメータ $(c, d) = (0.15, 0.1)$），UCB-V [15]（パラメータ $(b, c, \zeta) = (1, 1, 1)$），KL-UCB [16]（パラメータ $c = 0$），MOSS [17]（パラメータなし）and UCB-Tuned [12]（パラメータは参考論文と同様）である．

最初の実験セット（図 7）では，ロックアップ期間の最大サイズ S と regret の関係を見た．実験 1 と 2，実験 3 と 4 がほぼ同じ設定でロックアップ期間の割合が異なるものとなっている．それぞれの試行において，ロックアップ期間は以下のようにランダムに生成した．すべてのロックアップ期間の合計ラウンド数が T に到達するまで，新しいロックアップ期間を追加していく．新しいロックアップ期間は

サイズ $\{1, ..., S\}$ の中から，(1) 等確率で（実験 1，3）(2) サイズに反比例した確率で（実験 2，4）生成される．つまり，実験 1，3 では $\{1, ..., S\}$ の大きさのロックアップ期間が同じ割合で生成されるのに対して，実験 2，4 では小さいロックアップ期間がより多く生成されることになる．全体の期間のサイズが T となるように，最後のロックアップ期間を切り落とすようにした．ここで，$T = 1000$（実験 1，2），$T = 10000$（実験 3，4）である．つまり，横軸の S はここに述べる手続きで生成されうるロックアップ期間の最大サイズであり，実際の試行ごとのロックアップ期間の最大サイズは $L_{max} \leq S$ である．すべての実験の各 S の値において，regret の値は 10,000 回の試行の平均値を表示した．これまで述べてきたように，バンディット問題において探索が足りない場合には，経験期待値最大のアームが実際に最適なアームでないというリスクが低確率で存在する．このリスクを計算機で評価するためには，試行回数をラウンド数と同じオーダでとる必要がある．そのため，本実験ではすべての値を 10,000 回の試行の平均としている．

2 つ目の実験セット（図 8）では，ラウンドと regret の関係を調べた．設定は同一であり，実験ごとにそれぞれ異なるアルゴリズムを調べた．以下が実験設定である．ロックアップ期間は以下の手続きでランダムに生成された．最初の 2,000 ラウンドは，ロックアップを設定しない (i.e., $L_1, ..., L_{2000} = 1$)．ラウンド 2,001 から 10,000 までは，ロックアップ期間は 1 つ目の実験セットと同様にランダムに生成した．具体的には，全ラウンド数が 10,000 に達するまで，サイズ $\{1, ..., 1000\}$ のいずれかのロックアップ期間をサイズに反比例する確率で生成する．実験 5-8 はそれぞれのベースアルゴリズム（UCB-E′，ϵ_n-greedy′，MOSS′ と KL-UCB′）ごとに，ベースアルゴリズムと BaR を適用した後のアルゴリズムを比較したものである．BaR のハイパーパラメータは $L_t = 400$ とした．さらに，ロックアップ期間がどの程度探索と活用のバランスを妨げているかを見るために，ロックアップ期間のない通常の確率的バンディット問題の regret も比較した．

6.2　実験結果と考察

1 つ目の実験セット（図 7（実験 1-4））の結果を見ると，すべてのアルゴリズムにおいて，最大ロックアップ期間サイズ S と regret が比例しているのが分かる．ここで，実験 1 と 2（または実験 3 と 4）では，サイズの大きいロックアップ期間の割合が大きく異なる（実験 1（実験 3）では，$\{1, ..., S\}$ のすべての大きさのロックアップ期間が均等に生成されるが，実験 2（実験 4）ではサイズに反比例して大きいロックアップ期間が生成されにくくなる）．それにもかかわらず，実験 1 と 2（実験 3 と 4）の図はよい類似を見せている．これは，ロックアップ期間による制約がその最大サイズに強く依存することを示している．つまり，割合が少ない場合でも，大きいロックアップ期間の存在がアルゴリズムに影響を与えることになる．2 つ目の実験セット（図 8（実験 5-8））では BaR メタアルゴリズムの効果を検証した．すべての元アルゴリズムにおいて，BaR を利用することによって有意に regret を減少させられたことが分かる．また，ベースとなるアルゴリズムの regret が大きいほど，BaR を利用する前と利用した後の regret の減少量が大きい．とくに，ベースアルゴリズムが UCB-E′ の場合（実験 5），[BaR, UCB-E′] の regret はロックアップの制限のないバンディットでの結果より小さくなった．ロックアップ期間の制約があるほうがはるかに難しい問題であるため，これは驚くべき結果に見えるが，以下のように解釈が可能である．regret の大きかった UCB-E では，他のアルゴリズムと比較して探索がやや過剰である．そのため，BaR によって大きいロックアップ期間の部分での探索をカットすることによって，探索と活用のバランスがより最適に近い状態になったと考えられる．

7.　おわりに

本稿では，バンディット問題において，アームの変更に外部からの制限がある場合のモデルであるロックアップ期間を導入した拡張を提案した．確率的バンディット問題は複数のアームの中で最も期待報酬を高いものを探す問題であり．良いアルゴリズムは探索と活用をバランスしている．多くのアルゴリズムは自然にロックアップ制約のある場合への拡張が可能だが，ロックアップ期間中に同じアームを選び続けないとならないため，このバランスが崩れる可能性がある．探索は $O(1/t)$ 程度が最適であり，ラウンドが十分大きければ探索より活用の部分が大きくなるため，バランスが探索側に崩れると立て直すのが難しい．この問題に対処するために，大きいロックアップ期間では活用のみをとる方向にアルゴリズムを傾ける手法（BaR）を提案した．結果として，アルゴリズムの regret を最適トレードオフ状態に近づけることができた．バンディット問題におけるアームの最適な割当てのアルゴリズムは多く研究されているが，完全に自由にアームを選べないような制約が入った場合でも，制約が小さければこれらのアルゴリズムは十分使えるといえるであろう．

参考文献

[1] Robbins, H.: Some aspects of the sequential design of experiments, *Bulletin of the AMS*, Vol.58, pp.527–535 (1952).

[2] Zhao, Q., Tong, L., Swami, A. and Chen, Y.: Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework, *IEEE Journal on Selected Areas in Communications*, Vol.25, pp.589–600 (online), DOI: 10.1109/JSAC.2007.070409 (2007).

[3] Maron, O. and Moore, A.W.: Hoeffding Races: Accelerating Model Selection Search for Classification and Func-

tion Approximation, *NIPS*, pp.59–66 (1993).
[4] Mnih, V., Szepesvári, C. and yves Audibert, J.: Empirical Bernstein stopping, *International Conference on Machine Learning*, pp.672–679 (online), DOI: 10.1145/1390156.1390241 (2008).
[5] Agarwal, A., Dekel, O. and Xiao, L.: Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback, *Computational Learning Theory*, pp.28–40 (2010).
[6] Kakade, S.M., Shalev-shwartz, S. and Tewari, A.: Efficient bandit algorithms for online multiclass prediction, *ICML*, pp.440–447 (online), DOI: 10.1145/1390156.1390212 (2008).
[7] Crammer, K. and Gentile, C.: Multiclass Classification with Bandit Feedback using Adaptive Regularization, *ICML*, pp.273–280 (2011).
[8] Kocsis, L. and Szepesvári, C.: Bandit Based Monte-Carlo Planning, *ECML*, pp.282–293 (2006).
[9] Gittins, J.: *Multi-armed bandit allocation indices*, Wiley-Interscience series in systems and optimization, Wiley (1989).
[10] Berry, D.A.: Modified Two-Armed Bandit Strategies for Certain Clinical Trials, *Journal of The American Statistical Association*, Vol.73, pp.339–345 (online), DOI: 10.1080/01621459.1978.10481579 (1978).
[11] Press, W.H.: Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research, *Proc. National Academy of Sciences*, Vol.106, pp.22387–22392 (online), DOI: 10.1073/pnas.0912378106 (2009).
[12] Auer, P., Cesa-bianchi, N. and Fischer, P.: Finite-time Analysis of the Multiarmed Bandit Problem, *Machine Learning*, Vol.47, pp.235–256 (2002).
[13] Lai, T.L. and Robbins, H.: Asymptotically Efficient Adaptive Allocation Rules, *Advances in Applied Mathematics*, Vol.6, No.1, pp.4–22 (1985).
[14] Audibert, J.-Y., Bubeck, S. and Munos, R.: Best Arm Identification in Multi-Armed Bandits, *COLT* (2010).
[15] Audibert, J., Munos, R. and Szepesvári, C.: Exploration-exploitation trade-off using variance estimates in multi-armed bandits, *Theoretical Computer Science* (2008).
[16] Garivier, A. and Cappé, O.: The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond, *Journal of Machine Learning Research — Proceedings Track*, Vol.19, pp.359–376 (2011).
[17] Audibert, J.-Y. and Bubeck, S.: Minimax policies for adversarial and stochastic bandits, *COLT* (2009).
[18] Burnetas, A.N. and Katehakis, M.N.: Optimal Adaptive Policies for Markov Decision Processes, *Mathematics of Operations Research*, Vol.22, pp.222–255 (online), DOI: 10.1287/moor.22.1.222 (1997).
[19] Honda, J. and Takemura, A.: An Asymptotically Optimal Bandit Algorithm for Bounded Support Models, *Computational Learning Theory*, pp.67–79 (2010).
[20] Jun, T.: A survey on the bandit problem with switching costs, *De Economist*, Vol.152, No.4, pp.513–541 (2004).
[21] Mahajan, A. and Teneketzis, D.: Multi-armed bandit problems, *Foundations and Applications of Sensor Management*, pp.121–151 (2008).
[22] Guha, S. and Munagala, K.: Multi-armed Bandits with Metric Switching Costs, *ICALP* (*2*), pp.496–507 (2009).
[23] Kleinberg, R.D., Niculescu-mizil, A. and Sharma, Y.: Regret Bounds for Sleeping Experts and Bandits, *Machine Learning*, Vol.80, pp.425–436 (2008).
[24] Bui, L., Johari, R. and Mannor, S.: Committing Bandits, *NIPS*, pp.1557–1565 (2011).
[25] Bubeck, S., Munos, R. and Stoltz, G.: Pure Exploration in Multi-armed Bandits Problems, *ALT*, pp.23–37 (2009).

小宮山 純平

1985 年生．2007 年東京大学工学部卒業．2009 年東京大学大学院工学系研究科修士課程修了．2012 年より東京大学大学院情報理工学系研究科博士課程在籍．

佐藤 一誠　（正会員）

2011 年東京大学大学院情報理工学系研究科博士課程修了．2011 年より東京大学情報基盤センター助教．2013 年より科学技術振興機構さきがけ研究員を兼務．統計的機械学習およびデータマイニングの研究に従事．

中川 裕志　（正会員）

1953 年生．1975 年東京大学工学部卒業．1980 年東京大学大学院工学系研究科博士課程修了．工学博士．1980 年より横浜国立大学工学部勤務．1999 年より東京大学情報基盤センター教授．統計的機械学習の研究に従事．