取引手数料を考慮したオンライン為替交換アルゴリズムの効率解析

檀 浦 詠 介† 櫻 井 幸 一†

文献 [El-Yaniv, Fiat, Karp, and Turpin: *Proc. of FOCS* (1992)] においてオンライン為替交換アルゴリズムが提案されている．これは円相場（円/ドル）の増加区間と減少区間に分割し，区間ごとに最適な取引を行うことで双方向の取引を実現している．また文献 [檀浦，櫻井：情報処理学会論文誌，Vol.37, No.12 (1996)] では双方向アルゴリズムの改良が行われている．しかしこれらのアルゴリズムは取引の際の手数料・税金などのコストを無視しており，手数料を考慮したオンラインアルゴリズムは提案されていない．手数料を考慮したモデルでは，従来は自明であったオフライン最適アルゴリズムが非自明となり，これを基準とした評価方法（競合比）によるオンラインアルゴリズムの設計も困難となる．本稿では手数料付きモデルに対するオフライン最適アルゴリズムを提案し，これに基づいたオンラインアルゴリズムの設計および改善を行った．また，競合比の下限に関しても合わせて解析を行った．

The Performances of On-line Algorithms for Money-trading with Brokerage

EISUKE DANNOURA† and KOUICHI SAKURAI†

On-line algorithms for money-making trading problem are investigated in Ref. [El-Yaniv, Fiat, Karp, and Turpin: *Proc. of FOCS* (1992)]. Their bidirectional algorithms divide exchange rate (yen/dollar) run into upward runs and downward runs, then repeat the optimal unidirectional algorithms in each run. Moreover, the improved bidirectional algorithm is invested in Ref. [Dannoura and Sakurai: *Trans. IPSJ Japan*, Vol.37, No.12 (1996)]. However, in these papers the costs of exchanging, e.g. brokerage and taxes, are ignored, and on-line algorithms for the model with brokerage hasn't been invested yet. The optimal off-line algorithm for the model isn't obvious, though the algorithm for the usual model is obvious, so it is difficult to design the on-line algorithm based on the competitive ratio, comparing the on-line algorithm to the optimal off-line algorithm. We analyze the performance of the optimal off-line algorithm for the model with brokerage, design and improve the on-line algorithms for the model. And further, we analyze the lower bound of competitive ratio for the model.

1. はじめに

オンラインアルゴリズムとは外部から連続した要求を受け取り，各々に対して瞬時に反応するアルゴリズムである．これに対してオフラインアルゴリズムとは，あらかじめすべての要求を受け取ったうえで，その要求全体に基づいて反応を起こすアルゴリズムである[9]．これまでにオンラインアルゴリズムは，未来の情報がない状態で，即決断しなければならないような状況が生じる様々な問題に対して応用されている．タスクシステム[1]，ロボット操縦[2),8),11]がこうした例である．タスクシステムでは，外部から連続した要求を受け取り，各々に対応した処理，およびシステムの状態遷移にコストがかかるというモデルにおいて，いかにコストを小さくするかということについて考えられている．また，ロボット操縦においては，ロボットが移動するごとに新たな障害が見つかっていくというモデルのもとでのオンラインアルゴリズムについて考察がなされている．また，最近ではこのような問題以外に，経済問題に対しての応用も研究されている．Cover[3]は株式投資におけるポートフォリオ選択問題を取り扱っている．ここでは株式市場における複数の証券に対してどのように投資を分散させればよいかを示す簡単なオンラインアルゴリズムを示している．また，Raghavan は統計的な敵（価格の変動）に対するオンラインの投資アルゴリズムを解析している[12]．このようなオンラインアルゴリズムの評価方法の1つとして競合比（competitive ratio）と呼ばれるものがある[10]．これは，期間 T において得られた情報を基に何らかのコ

† 九州大学大学院システム情報科学研究科情報工学専攻
Department of Computer Science and Communication Engineering, Kyushu University

ストがかかる行動をとるときに，オフラインアルゴリズムによる最適な行動の結果生じるコストを C_{OPT}，X というオンラインアルゴリズムを用いた結果を C_X とする．このとき $sup_T \frac{C_X(T)}{C_{OPT}(T)}$ を X の競合比とし，それをどれだけ1に近付けることができるかでそのアルゴリズムの良さを表すというものである．

El-Yaniv ら[6]は，二通貨間為替交換問題において，円相場変動に関する仮定のもとで，利得に関する競合比を有界にするオンラインアルゴリズムを設計している．二通貨間為替交換問題とは，変動相場制のもとで2つの通貨（ドルと円）の間で交換を繰り返すことにより利益を得ることを目的とする問題である．ここで提案されているアルゴリズムは，円相場（円/ドル）の値が増加している区間と減少している区間に分割し，増加している区間ではドルから円，減少している区間では円からドルという方向に取引を行う単一方向取引アルゴリズムを各々の区間ごとに適用し，それを繰り返すという操作により実現されている．

しかしながら，ここで採用されているモデルでは，為替交換にともなうコストが考慮されていないという問題点がある．実際の取引では通貨交換の際に手数料などのコストが不可避であることから，運用の際にはこのような手数料を考慮に入れた為替交換問題に対する双方向取引のアルゴリズムが必要となる．

為替交換問題に対して取引手数料の導入を行う際にまず問題となるのは，従来自明であった最適なオフラインアルゴリズムの動作が非自明となる点である．つまり従来は手数料がいっさい考慮されていなかったため，わずかな相場変動に対しても反応していたが，手数料を考慮したモデルではわずかな変動では利益が得られないため，ある程度大きな変動にのみ反応するようにする必要がある．

本研究では，手数料付き為替交換問題に対するオフライン最適アルゴリズムの動作を解析するとともに，このオフライン最適アルゴリズムによって得られる利得を基準とした評価方法（競合比）に基づいたオンラインアルゴリズムの設計を行った．さらに，この問題に対する競合比の下限についても解析を行った．

2. El-Yaniv らの二通貨間為替交換アルゴリズム

El-Yaniv ら[6]による二通貨間為替交換アルゴリズムは，円とドルの交換による取引を，「連続的モデル」と「離散的モデル」という2つのモデルのもとで実現している．本研究では離散的モデルを採用する．以下本章では，El-Yaniv らによる為替交換問題のモデルおよびアルゴリズムを説明する．

2.1 二通貨間為替交換問題の連続的モデル

取引を行う期間 $[0,T]$ において，円相場 x（円/ドル，$m \leq x \leq M$，かつ m，M は知られているとする）は時刻 t ($t \in [0,T]$) の関数として表され，$[0,T]$ 内の任意の時刻において取引可能である．ただし，関数 $x(t)$ には不連続点が存在しうるものとする．つまり，不連続点の発生に反応して取引を行う場合，不連続点の発生した後の相場で取引は行われる．これは，現実世界においては急激な相場の変動に対して反応するのが間に合わないことが十分ありうるので，それを考えに入れたものである．

このときの最適オフラインアルゴリズムは，$x(t)$ が極大値をとなったときにドルを円に，極小値のときに円をドルに交換するというものである．

2.2 為替交換アルゴリズム

El-Yaniv らは，為替交換アルゴリズムとして，単一方向アルゴリズムと双方向アルゴリズムという2つのアルゴリズムを提案している．

単一方向アルゴリズムとは以下のようなものである．ある一定期間 T において，単一方向への取引（円売ドル買か円買ドル売のいずれか）のみを行い，一方の通貨から別の通貨へ交換する状況を考える．ある入力（相場の変動）に対して最適な取引を行った結果得られる金額と，A というオンラインアルゴリズムを用いて取引した結果得られる金額の比の最大値を A の競合比（competitive ratio）と定義し，これを最小とする単一方向アルゴリズムが設計されている．

双方向アルゴリズムは，アルゴリズムを実行する期間を，相場が上昇している区間と降下している区間に分割し，単一方向アルゴリズムを，上昇区間ではドルから円の方向に，降下区間では円からドルの方向に交互に適用することで交換を行うというものである．

このアルゴリズムは以下に示す規則に従って取引を行う．以下の記述はドルから円への交換の場合であり，相場が x のときのドルの額を $D(x)$，円の額を $Y(x)$ とし，初期値を $x=a$，$D=1$，$Y=0$ とする．

規則

(1) 終了時には残っているドルをすべて円に交換する．

(2) 規則(1)のケース以外では，それまでよりも相場が上昇した場合に取引を行う．

(3) 規則(2)の条件で取引を行う場合は，相場の変化に応じて以下の金額を所持するように円を買う（$D(x)$ は減少関数であり，$D(M)=0$ であることに注意）．

$a \in [m, rm]$

$$\begin{cases} x \in [a, rm] & D(x) = 1 \\ x \in [rm, M] & D(x) = 1 - \dfrac{1}{r} \ln \dfrac{x - m}{rm - m} \end{cases}$$

$a \in [rm, M]$

$$\begin{cases} x = a & D(a) = \dfrac{a(1 - \frac{1}{R})}{a - m} \\ x \in [a, M] & D(x) = \dfrac{a(1 - \frac{1}{R})}{a - m} - \dfrac{1}{R} \ln \dfrac{x - m}{a - m} \end{cases}$$

ただし，r および R は以下のように定義される．

$$r = \ln \frac{\frac{M}{m} - 1}{r - 1}$$

$$R = \begin{cases} r & a \in [m, rm] \\ 1 + \dfrac{a - m}{a} \ln \dfrac{M - m}{a - m} & a \in [rm, M] \end{cases}$$

定理 2.1 [6]相場の変動が，アルゴリズム内で仮定された範囲 $[m, M]$ を越えなかった場合，この双方向アルゴリズムは k 区間で r^k という競合比を満たす．

3. 双方向アルゴリズムの改良

本章では，檀浦らによる，El-Yaniv らの双方向アルゴリズム[6]の改良[5]について説明する．

3.1 El-Yaniv らのアルゴリズムの問題点

前章で述べたように，El-Yaniv らの双方向アルゴリズムは，アルゴリズムを実行する期間を，相場が上昇している区間と降下している区間に分割し，それぞれの区間に対して r という競合比を満たす単一方向取引アルゴリズムを，2 つの方向に交互に適用することで，双方向の取引を行うというものである．この双方向アルゴリズムを適用すると，k 区間に対する競合比は r^k となる．

El-Yaniv らは単一方向アルゴリズムを最適に（r が最小になるように）設計したが，各区間は独立ではない（前の区間の最終値 = 次の区間の初期値という関係がある）ので，部分的に最適なものを繰り返しても全体は最適にはならない．

つまり，El-Yaniv らの単一方向アルゴリズム（ドルから円への交換の場合）は，円相場 x の初期値 a が M に近づくほど，より小さな競合比が実現可能となっている．つまりこれを双方向問題に適用した場合，前の区間の終了時の x の値から，次の区間での競合比（つまり，オフラインとオンラインの比の上限）が分かる．したがって双方向の場合，ある区間の終了時の値が次の区間の競合比を小さくするような値ならば，その区間で悪い結果になっても，全体としては悪くはならないということになる．

檀浦らはこの点に注目し，前後の区間との関係を考慮に入れ，ある区間で r よりも悪い結果になっても別の区間で取り戻せばよい，という考えを用いることで，双方向取引においては r^k より小さい競合比を得ることができる単一方向取引アルゴリズムを構築した．このアルゴリズムを以下に示す．

3.2 改良した単一方向取引アルゴリズム

檀浦らにより改良された単一方向取引アルゴリズム（上昇区間の場合）は k 区間（k は任意の自然数，ただし k 番目の区間の終了時において円相場 $x(t)$ は連続である）に適用したときに競合比が $\tilde{r}^k$ になるように設計されている．また，1 区間での競合比は $\tilde{r}^2$ である．

単一方向取引アルゴリズムを以下に示す．

規則

(1) 終了時には残っているドルをすべて円に交換する．

(2) 規則 (1) のケース以外では，それまでよりも相場が上昇した場合に取引を行う．

(3) 規則 (2) の条件で取引を行う場合は，相場の変化に応じて以下の金額を所持するように円を買う（$D(x)$ は減少関数であり，$D(M) = 0$ であることに注意）．

$a \in [m, \tilde{r}^2 m]$

$$\begin{cases} x \in [a, \tilde{r}^2 m] & D(x) = 1 \\ x \in [\tilde{r}^2 m, M] & D(x) = 1 - \dfrac{1}{\tilde{r}} \ln \dfrac{x - \tilde{r}m}{\tilde{r}^2 m - \tilde{r}m} \end{cases}$$

$a \in [\tilde{r}^2 m, M]$

$$\begin{cases} x = a & D(a) = \dfrac{a(1 - \frac{1}{\tilde{R}})}{a - \tilde{r}m} \\ x \in [a, M] & D(x) = \dfrac{a(1 - \frac{1}{\tilde{R}})}{a - \tilde{r}m} \\ & \qquad - \frac{1}{\tilde{R}} \ln \frac{x - \tilde{r}m}{a - \tilde{r}m} \end{cases}$$

ただし，$\tilde{r}$ および $\tilde{R}$ は以下のように定義される（$D(M) = 0$ を満たすように定義している）．

$$\tilde{r} = \ln \frac{\frac{M}{\tilde{r}m} - 1}{\tilde{r} - 1}$$

$$\tilde{R} = \begin{cases} \tilde{r} & a \in [m, \tilde{r}^2 m] \\ 1 + \dfrac{a - \tilde{r}m}{a} \ln \dfrac{M - \tilde{r}m}{a - \tilde{r}m} & a \in [\tilde{r}^2 m, M] \end{cases}$$

定理 3.1 [5]終了時点の相場が $\tilde{r}m$ 以上であれば，上記の単一方向アルゴリズムの競合比は $\tilde{r}$ である．

El-Yaniv らのアルゴリズムにおいて，

$$r = \ln \frac{\frac{M}{m} - 1}{r - 1}$$

という式が成り立ち，今回のアルゴリズムにおいて

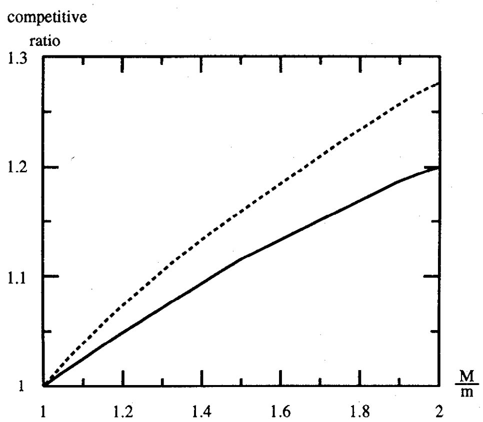


図 1 競合比の比較

Fig. 1 The graph of competitive ratios of two algorithm.

$$\tilde{r} = \ln \frac{\frac{M}{\tilde{r}m} - 1}{\tilde{r} - 1}$$

である．このとき，$r > \tilde{r}$ である．これらが具体的にどのような値となるかを図 **1** に示す．ここで，破線は El-Yaniv らによる値（r），実線は檀浦らによる値（$\tilde{r}$）である．

定理 3.2[5)] この単一方向アルゴリズムの 1 区間での競合比は $\tilde{r}^2$ である．

3.3 改良した単一方向取引アルゴリズムの双方向への適用

上に示した単一方向アルゴリズムを，各区間に対して繰り返し適用するとき，以下の定理が成り立つ．

定理 3.3[5)] 連続する k 区間において，この双方向アルゴリズムは以下のような競合比を保証する．

$$\begin{cases} \tilde{r}^k & k \text{ 番目の区間の終了時に } x(t) \text{ が連続} \\ \tilde{r}^{k+1} & k \text{ 番目の区間の終了時に } x(t) \text{ が不連続} \end{cases}$$

4. 手数料付き為替交換問題

これまでに扱ってきた為替交換問題のモデルでは，通貨交換の際の手数料はいっさい考慮されていなかった．しかしながら実際の取引では，交換の際に何らかのコストが必要となるため，これまでに提案されたアルゴリズムの実用性には問題が出てくる．

今回我々は，従来のアルゴリズムの実用上の問題点を考察したうえで，従来のモデルに対して手数料を導入する．このことにより，従来のモデルでは自明であった最適オフラインアルゴリズムの動作が非自明となる．本章では，この最適オフラインアルゴリズムの動作を解析し，これを基準とした評価方法（競合比）を用いたオンラインアルゴリズムの設計を行う．また，

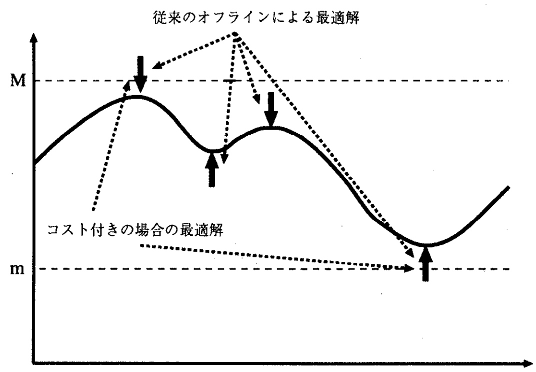


図 2 オフラインでの最適解の比較

Fig. 2 Comparing the two optimal off-line algorithms for each model.

競合比の下限の解析も合わせて行っていく．

4.1 相場の上下限が知られている連続的モデル

取引を行う期間 $[0, T]$ において，ドル売円買の相場 x（$m \leq x \leq M$，かつ m，M は知られているものとする）は時刻 t ($t \in [0, T]$) の関数として表され，$[0, T]$ 内の任意の時刻において取引可能である．また，円売ドル買の相場は cx ($c > 1$) とする．また，関数 $x(t)$ には，従来のモデルと同様に不連続点が存在しうるものとする．

4.2 手数料付き双方向問題に対する最適なオフラインアルゴリズム

本節では手数料付き双方向問題に対する最適オフラインアルゴリズムについて考察する．

最適オフラインアルゴリズムとは，いかなる入力（相場の変動）に対しても，それより大きな金額を得ることができないというアルゴリズムである．従来のモデルでの双方向問題に対する最適オフラインアルゴリズムは，円相場 x が極値をとるとき，かつそのときのみ取引を行うという，単純かつ自明なものであった．しかしながら手数料付きのモデルで同様のアルゴリズムを適用した場合最適とはならない．これは，従来の最適アルゴリズムでは，極大値と極小値がごくわずかな差しかない場合でも交換を行ったが，手数料付きモデルでは，この場合に利益よりも手数料の方が大きくなり，結果として不利益となる可能性があるためである．そのため，手数料付きの場合の最適オフラインアルゴリズムは，従来の最適アルゴリズムよりも少ない回数の取引になる（図 **2** 参照）．このことを考慮に入れて，双方向最適オフラインアルゴリズムを考える．

以下では，最初ドルのみを持っているものとし，このドルを増やすことを考えることにする．

ここで，最適なオフラインアルゴリズムは入力関数

が極値をとった場合にのみ取引を行うことから，入力を $(e_0, e_1, \cdots, e_k)$ $(e_0 < e_1 > \cdots e_{k-1} > e_k)$ という極値の列であると考える．

また，これに対する出力（最適なオフラインアルゴリズムの解）は $(s_1, b_1), (s_2, b_2), \cdots$ という形で表現する．ここで (s_j, b_j) は j 回目のドル売円買と j 回目の円売ドル買を，いくつめの入力で行ったかを示す（たとえば，e_1 でドル売円買，e_2 で円売ドル買を行うならば，$(s_1, b_1) = (1, 2)$ と表す）．

ただし，ある入力に対して最適なオフラインアルゴリズムの動作が一意に定まるとは限らない．これは，最終的に同じ金額が得られるような解（取引の仕方）が複数存在する場合があるからである．以下に例を示す．

(1) $e_1 = e_3$ の場合，$(1, 5)$ と交換しても $(3, 5)$ と交換しても同じ金額が得られる．

(2) $e_1 = ce_4$ の場合，$(1, 4)$ という交換を行っても行わなくても同じ金額が得られる．

(3) $ce_2 = e_3$ の場合，$(1, 2)$，$(3, 4)$ と交換しても $(1, 4)$ と交換しても同じ金額が得られる．

上記の 3 通りの例の組合せにより，最適解が複数存在することがありうる．そこで，本節で示す最適オフラインアルゴリズムは，(1) の場合は先に取引を行い（$(3, 5)$ ではなく $(1, 5)$ で取引を行い），(2)，(3) の場合は，取引回数が少なくなるように取引を行うアルゴリズムである．取引回数が少ないようにオフラインアルゴリズムを定めるのは，次節以降でオンラインアルゴリズムを議論する際に，取引回数の少ない（オフラインアルゴリズムの）最適解に基づて議論した方が，より小さな競合比が得られるからである．

まずこのオフラインアルゴリズムを示し，次にこのアルゴリズムが最適であることを示す．

アルゴリズム：

(1) $I = 1$，$i = 1$ とする．

(2) $e_I < e_J$ または $e_I > ce_J$ を満たす最小の J $(J > I)$ を探す．

 (a) $e_I < e_J$ の場合
 $I = J$ として (2) へ．

 (b) $e_I > ce_J$ の場合
 $s_i = I$ を解とする．$I = J$ として (3) へ．

 (c) 条件を満たす値が存在しない場合
 終了する．

(3) $e_I > e_J$ または $ce_I < e_J$ を満たす最小の J $(J > I)$ を探す．

 (a) $e_I > e_J$ の場合
 $I = J$ として (3) へ．

 (b) $ce_I < e_J$ の場合
 $b_i = I$ を解とする．$i{+}{+}$，$I = J$ として (2) へ．

 (c) 条件を満たす値が存在しない場合
 $b_i = I$ を解とする．終了する．

定理 4.1 上記の手数料付きオフラインアルゴリズムは最適である．

証明. まずは上記のアルゴリズムが空集合を出力する（いっさい取引を行わない）場合について考える．

上記のアルゴリズムが空集合を出力するのは，入力（円相場変動の極値の列）$(e_0, e_1, \cdots, e_k)$ において，$e_{J_1} > ce_{J_2}$ かつ $0 \leq J_1 < J_2 \leq k$ を満たす J_1，J_2 が存在しない場合，かつそのときのみである．これは，「いっさいの取引を行わない」のが最適となる（どのような取引を行っても利益を得ることが不可能である）ための必要十分条件である．よって，上記のアルゴリズムが空集合を出力する場合は，最適オフラインアルゴリズム（の 1 つ）である．

次に，上記のアルゴリズムが空でない解を出力する場合について考える．上記のアルゴリズムによる出力を

$$(s_1, b_1), (s_2, b_2), \cdots, (s_{n/2}, b_{n/2}) \qquad (1)$$

とする．このとき，この出力解は以下の条件を満たしている．

条件：

(1) 任意の J $(0 \leq J < s_1)$ において $e_J < e_{s_1}$

(2) 任意の J_1, J_2 $(0 \leq J_1 \leq J_2 \leq s_1)$ において $e_{J_1} \leq ce_{J_2}$

(3) 任意の j $(1 \leq j \leq n/2)$ において $e_{s_j} > ce_{b_j}$

(4) 任意の J $(s_j \leq J < b_j,\ 1 \leq j \leq n/2)$ において $e_{b_j} < e_J \leq e_{s_j}$

(5) 任意の J_1, J_2 $(s_j \leq J_1 \leq J_2 \leq b_j,\ 1 \leq j \leq n/2)$ において $ce_{J_1} \geq e_{J_2}$

(6) 任意の j $(1 \leq j \leq n/2)$ において $e_{s_{j+1}} > ce_{b_j}$

(7) 任意の j, J $(1 \leq j < n/2,\ b_j \leq J < s_{j+1})$ において $e_{b_j} \leq e_J < e_{s_{j+1}}$

(8) 任意の j, J_1, J_2 $(b_j \leq J_1 \leq J_2 \leq s_{j+1},\ 1 \leq j \leq n/2)$ において $e_{J_1} \leq ce_{J_2}$

(9) 任意の J $(b_{n/2} \leq J \leq k)$ において $e_{b_{n/2}} \leq e_J$

(10) 任意の J_1, J_2 $(b_{n/2} \leq J_1 \leq J_2 \leq k)$ において $e_{J_1} \leq ce_{J_2}$

また，ある最適オフラインアルゴリズムの出力解を

$$(s^*_1, b^*_1), (s^*_2, b^*_2), \cdots, (s^*_{t/2}, b^*_{t/2}) \qquad (2)$$

（ただし，$s^*_i \neq cb^*_i$，かつ $cb^*_i \neq s^*_{i+1}$ である．つまり，複数存在する最適解のうち，取引回数が最小の解（の 1 つ）である）とする．

以下では，$n=t$，かつ任意の $1 \leq i \leq$ において $e_{s_i}=e_{s^*_i}$，$e_{b_i}=e_{b^*_i}$（つまり，2つのアルゴリズムは，同じ回数の取引を，相場が同じ値のときに行う．ただし同じ時刻であるとは限らない）であることを証明する．

まずは，$e_{s_1}=e_{s^*_1}$ を示す．s_1，b_1，s^*_1，b^*_1 の大小関係により場合分けを行う．

(1) $s^*_1 < s_1$ の場合
 (a) $b^*_1 \leq s_1$ の場合
 条件2より $s^*_1 \leq cb^*_1$ であるので，(s^*_1, b^*_1) という取引では利益を得られない．よって，(2) が最適解であることに矛盾する．
 (b) $b^*_1 > s_1$ の場合
 条件1より $s^*_1 < s_1$ であるので，(s^*_1, b^*_1) よりも (s_1, b^*_1) の方が利益が大きい．よって，(2) が最適解であることに矛盾する．

(2) $s_1 \leq s^*_1 \leq b_1$ の場合
 条件4より $e_{s_1} \geq e_{s^*_1}$ である．
 (a) $e_{s_1} > e_{s^*_1}$ の場合
 (s^*_1, b^*_1) よりも (s_1, b^*_1) の方が利益が大きい．よって，(2) が最適解であることに矛盾する．
 (b) $e_{s_1} = e_{s^*_1}$ の場合
 (2) が最適解であることに矛盾しない．

(3) $b_1 < s^*_1$ の場合
 (s^*_1, b^*_1) よりも $(s_1, b_1), (s^*_1, b^*_1)$ と取引を行った方が利益が大きい．よって，(2) が最適解であることに矛盾する．

ゆえに $e_{s_1}=e_{s^*_1}$ かつ $s_1 \leq s^*_1 \leq b_1$ である．

次に，$e_{b_1}=e_{b^*_1}$ を示す．上と同様に場合分けを行う．

(1) $s^*_1 < b^*_1 < b_1$ の場合
 (a) $s^*_2 \leq b_1$ の場合
 条件5より，$cb^*_1 \geq s^*_2$ であるので，(s^*_1, b^*_1)，(s^*_2, b^*_2) という取引よりも，(s^*_1, b^*_2) という取引の方が，利益が大きい，もしくは等しい．よって，(2) が取引回数の最も少ない最適解であることに矛盾する．
 (b) $s^*_2 > b_1$ の場合
 条件4より，$b^*_1 > b_1$ であるので，(s^*_1, b^*_1) よりも (s^*_1, b_1) の方が利益が大きい．よって，(2) が最適解であることに矛盾する．

(2) $b_1 \leq b^*_1 \leq s_2$ の場合
 条件7より，$e_{b_1} \leq e_{b^*_1}$ である．
 (a) $e_{b_1} < e_{b^*_1}$ の場合
 (s^*_1, b^*_1) よりも (s^*_1, b_1) の方が利益が大きい．よって，(2) が最適解であることに矛盾する．
 (b) $e_{b_1} = e_{b^*_1}$ の場合
 (2) が最適解であることに矛盾しない．

(3) $b^*_1 > s_2$ の場合
 (s^*_1, b^*_1) よりも $(s^*_1, b_1), (s_2, b^*_1)$ の方が利益が大きい．よって，(2) が最適解であることに矛盾する．

ゆえに $e_{b_1}=e_{b^*_1}$ かつ $b_1 \leq b^*_1 \leq s_2$ である．

次に，n と t の大小関係により場合分けを行う．

(1) $t<n$ の場合
 $e_{s_1}=e_{s^*_1}$，$e_{b_1}=e_{b^*_1}$ の証明と同様にして $e_{s_j}=e_{s^*_j}$，$e_{b_j}=e_{b^*_j}$ $(j=1,2,\cdots t-1)$ が証明できる．このとき，(s^*_t, b^*_t) をいかなる値にとっても，(1) よりも大きな利益を得ることはできない．ゆえに (2) が最適解であることに矛盾する．

(2) $t>n$ の場合
 $e_{s_1}=e_{s^*_1}$，$e_{b_1}=e_{b^*_1}$ の証明と同様にして $e_{s_j}=e_{s^*_j}$，$e_{b_j}=e_{b^*_j}$ $(j=1,2,\cdots n)$ が証明できる．このとき，(s^*_{n+1}, b^*_{n+1}) をいかなる値にとっても，条件10より $s^*_{n+1} \leq cb^*_{n+1}$ である．よって，(2) が取引回数の最も少ない最適解であることに矛盾する．

(3) $t=n$ の場合
 $e_{s_1}=e_{s^*_1}$，$e_{b_1}=e_{b^*_1}$ の証明と同様にして $e_{s_j}=e_{s^*_j}$，$e_{b_j}=e_{b^*_j}$ $(j=1,2,\cdots n)$ が証明できる．このとき，(2) が最適解であることに矛盾しない．

ゆえに，$n=t$，かつ任意の $1 \leq i \leq$ において $e_{s_i}=e_{s^*_i}$，$e_{b_i}=e_{b^*_i}$ である．

よって，上記のオフラインアルゴリズムは，最適オフラインアルゴリズムである． □

これにより，手数料付き問題に対する最適オフラインアルゴリズムの動作が分かる．次節ではこれに基づいたオンラインアルゴリズムを考察する．

4.3 手数料付き問題に対するオンラインアルゴリズム

本節では，手数料付き問題に対するオンラインアルゴリズムを提案する．このアルゴリズムは，El-Yaniv らの単一方向アルゴリズムを，前節で示した最適オフラインアルゴリズムに基づいた区間ごとに適用するこ

とで実現しており，El-Yaniv らの双方向アルゴリズムの拡張になっている．この双方向アルゴリズムは，以下の規則に従って単一方向アルゴリズムを繰り返すことで実現している．

規則

(1) 円からドル（ドルから円）への単一方向アルゴリズムは，円相場が cr_cm より大きく（$\frac{M}{cr_c}$ より小さく）なった時点から取引を始める（つまり，円相場が cr_cm より大きくならない限りは，何もしなくても r_c という競合比が満たされる）．

(2) 円からドル（ドルから円）への単一方向アルゴリズムは，円相場が，単一方向アルゴリズム適用期間中の最大値（最小値）の $1/c$ より小さく（c 倍より大きく）なった時点で終了する（つまり，$1/c$ 未満になるまで最適オフラインアルゴリズムではすでに交換が終わっていることが分からない）．

(3) 終了時点でドル（円）のみしか持っていない場合はそのまま終了し，次の区間では取引は行わない．それ以外は，すべて円（ドル）に交換する．

このとき，単一方向アルゴリズムでは以下の式に従って取引を行う．以下の記述はドルから円への交換の場合であり，相場が x のときのドルの額を $D(x)$，円の額を $Y(x)$ とし，初期値を $x=a$，$D=1$，$Y=0$ とする．

$a \in [m, cr_cm]$

$$\begin{cases} x \in [a, cr_cm] \\ \quad D(x) = 1 \\ x \in (cr_cm, M] \\ \quad D(x) = \dfrac{c(r_c-1)}{cr_c-1} - \dfrac{1}{r_c}\ln\dfrac{x/m-1}{cr_c-1} \end{cases}$$

$a \in (cr_cm, M]$

$$\begin{cases} x = a \\ \quad D(a) = \dfrac{a(1-\frac{1}{R_c})}{a-m} \\ x \in [a, M] \\ \quad D(x) = \dfrac{a(1-\frac{1}{R_c})}{a-m} - \dfrac{1}{R_c}\ln\dfrac{x-m}{a-m} \end{cases}$$

ただし，r_c および R_c は以下のように定義される．

$$r_c = 1 + \frac{cr_c-1}{cr_c}\ln\frac{M/m-1}{cr_c-1}$$

（$c=1$ のとき，$r_c=r$ である．）

$$R_c = \begin{cases} r_c & a \in [m, cr_cm] \\ 1+\dfrac{a-m}{a}\ln\dfrac{M-m}{a-m} & a \in [cr_cm, M] \end{cases}$$

この双方向アルゴリズムによって得られる競合比は以下のようになる．

定理 4.2 この双方向アルゴリズムによって得られる競合比は ${cr_c}^{n+1}$ である．ただし，n は最適オフラインアルゴリズムによる取引の回数である．

証明. まずは各区間ごとに r_c という競合比が満たされることを証明する（ここではドルから円への交換を行う区間について説明する）．

ある区間での x の最大値（オフラインアルゴリズムドルから円への交換を行う値）を $x_{\max}$ とする．

1. $x_{\max} \le cr_cm$ の場合：

最適オフラインアルゴリズムが次の区間で，オフラインアルゴリズムが円からドルへの交換を行う値は最小でも cm であることから，この 2 区間で，最適オフラインアルゴリズムはたかだか r_c 倍に増やすことしかできない．よってこの場合，何もしなくとも 2 区間での競合比が r_c となる．

2. $x_{\max} > cr_cm$ の場合：

この単一方向アルゴリズムは，El-Yaniv らの単一方向アルゴリズムの $D(x)$ に $a=cr_cm$ を代入し，このとき $r_c=R$ となるように設計されている．よって，各区間ごとに r_c という競合比が実現する．

ただし，オフラインアルゴリズムが i 回の交換を行う場合，このオンラインアルゴリズムは $i+1$ 区間に分割してしまう．ここで，最適オフラインアルゴリズムで得られるドルを D，オンラインアルゴリズムで i 番目の区間が終了した時点のドルの値を $D_*=\frac{D}{{r_c}^i}$，$i+1$ 番目の区間の x の最大値を x_* とする．このときオンラインアルゴリズムが，$i+1$ 番目の区間の終了時に得られる円は x_*D_*/r_c である．しかし，この問題はドルを得ることを目的としているため，これをドルに戻す必要がある．このとき得られるドルは最小で $\frac{x_*D_*/r_c}{cx_*} = D/({cr_c}^{i+1})$ である．よって，競合比は ${cr_c}^{i+1}$ である． □

ここで，$c=1$（手数料なし）のとき，$r_c=r$ であり，このアルゴリズムは El-Yaniv らのオンライン双方向アルゴリズムと一致する．

4.4 手数料付き双方向アルゴリズムの改良

本節では，手数料付き双方向アルゴリズムの改良を行う．

本節で示す双方向アルゴリズムは，手数料なしのモデルにおいて，El-Yaniv らのアルゴリズムに対して我々が行った改良[5)]（3 章参照）と同じ手法によって，前節のアルゴリズムを改良したものである．このアルゴリズムは $c=1$ のとき，手数料なしのモデルに対する著者らの双方向アルゴリズムと一致する．

このアルゴリズムは以下の規則に従って単一方向ア

ルゴリズムを繰り返すことで実現している．

規則

(1) 円からドル（ドルから円）への単一方向アルゴリズムは，円相場が $c\tilde{r_c}^2 m$ より大きく（$\frac{M}{c\tilde{r_c}^2}$ より小さく）なった時点から取引を始める．

(2) 円からドル（ドルから円）への単一方向アルゴリズムは，円相場がその適用期間中の最大値（最小値）の $\frac{1}{c}$ より小さく（c 倍より大きく）なった時点で終了する．

(3) 終了時点でドル（円）のみしか持っていない場合はそのまま終了し，次の区間では取引は行わない．それ以外は，すべて円（ドル）に交換する．

ただし，$\tilde{r_c}$ は以下のように定義される．

$$\tilde{r_c} = \begin{cases} \tilde{r} & c = 1 \text{ のとき} \\ 1 + \dfrac{c\tilde{r_c} - 1}{c\tilde{r_c}} \ln \dfrac{M/\tilde{r_c}m - 1}{c\tilde{r_c} - 1} & c > 1 \text{ のとき} \end{cases}$$

このとき，単一方向アルゴリズムでは以下の式に従って取引を行う．以下の記述はドルから円への交換の場合であり，相場が x のときのドルの額を $D(x)$，円の額を $Y(x)$ とし，初期値を $x = a$，$D = 1$，$Y = 0$ とする．

$a \in [m, c\tilde{r_c}^2 m]$

$$\begin{cases} x \in [a, c\tilde{r_c}^2 m] \\ \quad D(x) = 1 \\ x \in [c\tilde{r_c}^2 m, M] \\ \quad D(x) = 1 - \dfrac{1}{\tilde{r_c}^2} \ln \dfrac{x - \tilde{r_c}m}{c\tilde{r_c}^2 m - \tilde{r_c}m} \end{cases}$$

$a \in [c\tilde{r_c}^2 m, M]$

$$\begin{cases} x = a \\ \quad D(a) = \dfrac{a(1 - \frac{1}{\tilde{R_c}})}{a - \tilde{r_c}m} \\ x \in [a, M] \\ \quad D(x) = \dfrac{a(1 - \frac{1}{\tilde{R_c}})}{a - \tilde{r_c}m} - \dfrac{1}{\tilde{R_c}} \ln \dfrac{x - \tilde{r_c}m}{a - \tilde{r_c}m} \end{cases}$$

このとき，以下の定理が成り立つ．

定理 4.3 このアルゴリズムによって得られる競合比は $c\tilde{r_c}^{n+1}$ である．ただし，n はオフラインアルゴリズムによる取引の回数である．

証明 アルゴリズムを適用する期間を $[0, T]$，最適オフラインアルゴリズムが取引を行う時刻を t_i ($t_i < t_{i+1}, t_0 = 0, t_k = T$) とする．以下の証明は，プレイヤーは最初ドルのみを持っているものとし，このドルを増やすことを目的としているものとする．

$x(t_i)$ で最適なオフラインアルゴリズムが取引を行うことがプレイヤーに分かった時刻（つまり，オンラインアルゴリズムによる区間の終了時）の x の値を x_i とする．

単一方向アルゴリズムの中で定義されていた $\tilde{R_c}$ を，区間ごとに定義するために，以下のように x の関数として拡張する．

$$\tilde{R_c}(x) = \begin{cases} \tilde{r_c} & x \in [m, c\tilde{r_c}^2 m] \\ 1 + \dfrac{x - \tilde{r_c}m}{x} \ln \dfrac{M - \tilde{r_c}m}{x - \tilde{r_c}m} & x \in [c\tilde{r_c}^2 m, M] \end{cases}$$

$$\tilde{R_n}(x) = \begin{cases} \tilde{R_c}(x) & n \text{ が偶数のとき} \\ \tilde{R_c}\left(\dfrac{mM}{x}\right) & n \text{ が奇数のとき} \end{cases}$$

また，n 番目の区間における最適な取引と実際の取引の利得の比を $R_n{}^*$ と定義すると，以下の式が成り立つ．

$$\tilde{R_n}^* \le \tilde{r_c} \alpha_n^*(x_n) \beta_{n-1}^*(x_{n-1}) \quad (n = 1, 2, \cdots, i)$$

ただし，$\alpha_n^*(x)$ と $\beta_n^*(x)$ は以下のように定義される．

$$\alpha_n^*(x) = \begin{cases} \max\left[1, \dfrac{\tilde{r_c}x}{M}\right] & n \text{ が偶数} \\ \max\left[1, \dfrac{\tilde{r_c}m}{x}\right] & n \text{ が奇数} \end{cases}$$

（以上の定義より $\alpha_n^*(x) \ge 1$ である）

$$\beta_n^*(x) = \frac{\tilde{R_n}(x)}{\tilde{r_c}} \ (\le 1)$$

また，$\alpha_n^*(x)$，$\beta_n^*(x)$ の間には以下のような関係がある．

$$\alpha_n^*(x)\beta_n^*(x) \le 1 \quad (n = 1, 2, \cdots, i-1)$$

上の3つの式より以下の式が導かれる．

$$\prod_{n=1}^{i} \tilde{R_n}^* \le \tilde{r_c}^i \beta_0^*(x_0) \alpha_i^*(x_i) \prod_{n=1}^{i-1} \{\alpha_n^*(x_n)\beta_n^*(x_n)\} \le \tilde{r_c}^{i+1}$$

ただし，$x_i = M$ の場合，このオンラインアルゴリズムはこの時点で得られたドルをすべて円に交換してしまうが，もし $x = M$ のままアルゴリズムの適用期間を終了した場合，M/c（円/ドル）でドルに交換する必要があるため，得られたドルが $1/c$ になってしまう．よって競合比は以下のようになる．

$$sup\ c \prod_{n=1}^{k} \tilde{R_n}^* = c\tilde{r_c}^{k+1}$$

□

ここで $c = 1$（手数料なし）のとき，このアルゴリズムの動作は，手数料を考慮しないモデルにおいて我々が改良したオンライン双方向アルゴリズムと一致

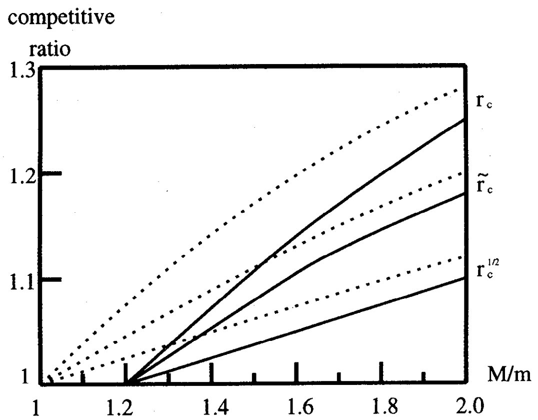


図 3 手数料を考慮したモデルにおける競合比の比較
Fig. 3 The graph of competitive ratios for the model with brokerage.

する．

ここで，r_c および $\tilde{r_c}$ の値を**図 3** に示す．破線で示しているのが $c = 1$ のときの r_c および $\tilde{r_c}$ の値であり，実線で示しているのが $c = 1.2$ のときの値である．

4.5 手数料付き問題に対する競合比の下限

El-Yaniv らは単一方向問題において，競合比の上限と下限を一致させ，最適な単一方向アルゴリズムを示している[6]．この単一方向アルゴリズムの競合比（競合比の上限）と下限は，以下の理由により一致している．

El-Yaniv らの単一方向アルゴリズムより小さな競合比を実現する別のアルゴリズムが存在するとし，2 人のプレイヤーが各々 2 つのアルゴリズムに従い，ドルから円へ交換を行うとする（El-Yaniv らの単一方向アルゴリズムを用いるプレイヤーを A，もう 1 人を B とする）．

円相場が rm の時点で A は $D(x) = 1$，$Y(x) = 0$ を満たしている．次の瞬間 $x = m$ になった場合，得られるドルは $1/m$ である．ここで B が，r より小さな競合比を実現するには $1/m$ より多くの円を得なければならないので，すでにドルを円にある程度交換しておく必要がある．

同様に x の値が上昇するたびに，B は A よりも多くのドルを円に交換する必要がある．しかし，El-Yaniv らの単一方向アルゴリズムでは，$x = M$ のときにちょうど $D(x) = 0$ となるように設計しているため，B は当然 $x = M$ となる以前にドルがなくなり，それ以降「A よりも多くのドルを円に交換する」ことができなくなる．このことから，El-Yaniv らの単一方向アルゴリズムよりも優れた競合比のアルゴリズムは存在しない．

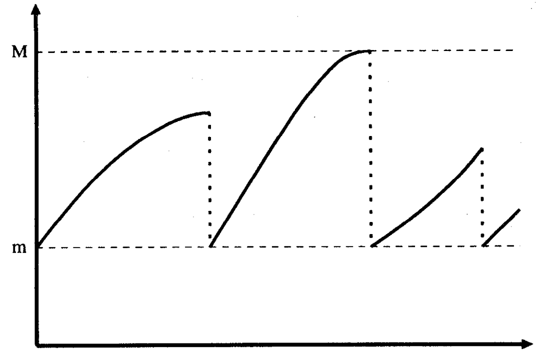


図 4 下限を示す制限
Fig. 4 The input which brings the lower bound.

El-Yaniv らは，双方向問題に関しても $r^{k/2}$ という競合比の下限を示している．これは以下のような手法により証明されている．

双方向問題では，相場 x が $x \in [m, M]$ を満たすという制限が用いられている．この制限に対する双方向最適アルゴリズムは未解決であるが，より強い制限を加えることで，El-Yaniv らの最適単一方向アルゴリズムを利用して全体として最適なアルゴリズムを設計できる場合がある（**図 4** 参照）．これは，円相場 x が，最初 m から連続的に上昇し，ある（未知の）値になった次の瞬間に不連続に降下して m になり，以降同様の上昇・降下を繰り返すというものである．

図 4 の入力に対しては，不連続に m に落ちることが分かっていることから，円からドルへの交換は（オフラインアルゴリズムと同じ）円相場 m で交換可能であるため，上昇区間でドルから円への交換を最適単一方向アルゴリズムに従い交換し，$x = m$ の時点ですべての円をドルに交換する，という操作の繰返しにより，全体として最適なオンラインアルゴリズムとなり，競合比は $r^{k/2}$ となる．

ここで，双方向問題でのモデル（$x \in [m, M]$ という制限しかないモデル）について考えると，このモデルは図 4 のような入力を含んでいるため，$r^{k/2}$ よりも小さな競合比を実現することは不可能である．よって，上下限を知られたモデルに対する双方向問題において $r^{k/2}$ という下限を示すことができる．

今回は，手数料付き双方向問題において，El-Yaniv らと同様に，図 4 のような入力に対する最適なオンラインアルゴリズムを考慮することで，手数料付き問題における競合比の下限が得られたのでこれを示す．

手数料付き問題における，図 4 の入力に対する最適なオンラインアルゴリズムは以下の動作を行う．相

場の上昇区間では，$x \in [m, cm]$ では取引を行わず，$x \in [cm, M]$ では，従来の単一方向アルゴリズムを，円相場の変動範囲を $[m, M]$ ではなく $[cm, M]$ として適用し，ドルから円への交換を行い，$x = m$ の時点ですべての円をドルに交換する．以下これを繰り返すことにより，最適なオンラインアルゴリズムとなる．

これにより下限として $\bar{r}_c{}^{n/2}$ ($\bar{r}_c = \ln \frac{\frac{M}{cm}-1}{\bar{r}_c-1}$) が得られる．

5. 今後の課題

本稿で考察した手数料とは，取引量に応じて一定比の手数料がかかるというものであった．しかしながら，取引の際の手数料には，取引量に応じたもの以外に取引回数に応じたものが存在する．ローン組替問題で考察されている手数料はこのようなものである[7]．このような，取引回数に応じた手数料を考察したモデルでは，1回の取引ごとに一定額の手数料が必要となり，取引回数を多くすると手数料が大きくなってしまうため，できるだけ取引回数を少なくするアルゴリズムが必要となる．

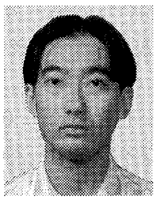
El-Yaniv らや我々の提案した単一方向アルゴリズムは，取引回数が無限大であるため，取引回数に応じた手数料を考慮したモデルでは使用できない．

今後は，本稿で考察したモデルに対して，上下限両面から競合比の改善を計るとともに，取引回数に応じた手数料を考慮したモデルでのアルゴリズムの設計を考察したい．

参考文献

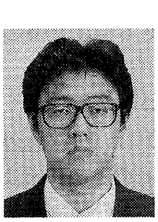
1) Borodin, A., Linial, N. and Saks, M.: An optimal online algorithm for metrical task system, *J. ACM*, Vol.39, pp.745–763 (1992).
2) Blum, A., Raghavan, P. and Schieber, B.: Navigating in unfamiliar geometric terrain, *Proc. 23rd Annual ACM Symposium on Theory of Computing*, pp.494–504 (1991).
3) Cover, T.M.: UNIVERSAL PORTFORIOS, *Journal of Mathematical Finance*, Vol.1, No.1, pp.1–29 (1991).
4) Chou, A., Cooperstock, J., El-Yaniv, R., Klugerman, M. and Leighton, T.: The Statistical Adversary Allows Optimal Money-Making Trading Strategies, *Proc. SODA '95* (1995).
5) 檀浦詠介，櫻井幸一：二通貨間為替交換問題に対するオンラインアルゴリズムの設計と解析，情報処理学会論文誌，Vol.37, No.12 (1996).
6) El-Yaniv, R., Fiat, A., Karp, R. and Turpin, G.: Competitive Analysis of Financial Games, *Proc. 33rd Annual Symposium on Foundations of Computer Science*, pp.327–333 (1992).
7) El-Yaniv, R. and Karp, R.M.: The Mortgage Problem, *Proc. 2nd Israel Symposium on Theory and Computing Systems* (1993).
8) Fiat, A., Foster, D.P., Karloff, H.J., Rabani, Y., Ravid, Y. and Vishwanathan, S.: Competitive algorithms for layered graph traversal, *Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science*, pp.288–297 (1991).
9) Karp, R.M.: On-Line Algorithms Versus Off-Line Algorithms: How Much is it Worth to know the Future?, *Proc. IFIP* (1992).
10) Karlin, A.R., Manasse, M.S., Rudolph, L. and Sleator, D.D.: *Competitive snoopy caching. Algorithmica*, 3 (1), pp.70–119 (1988).
11) Papadimitriou, C.H. and Yanakakis, M.: Shortest paths without a map, *Theoretical Computer Science*, Vol.84, pp.127–150 (1991).
12) Raghavan, P.: A statistical adversary for on-line algorithms, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol.7, pp.79–83 (1991).
13) Raghavan, P. and Snir, M.: Memory versus randomization in on-line algorithms, *IBM Journal of Research and Development*, Vol.38, pp.683–707 (1994).

(平成9年 4 月7日受付)
(平成9年11月5日採録)



檀浦　詠介（学生会員）

平成8年，九州大学工学部情報工学科卒業．現在，同大大学院システム情報科学研究科情報工学専攻修士2年．経済ゲーム，オンラインアルゴリズムに関する研究に従事．



櫻井　幸一（正会員）

昭和61年九州大学理学部数学科卒業．昭和63年同大大学院修士課程修了．同年三菱電機（株）入社．現在，九州大学大学院システム情報科学研究科情報工学専攻助教授．計算複雑性理論，暗号理論，情報セキュリティの研究に従事．工学博士．日本数学会会員．