コスト付き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

檀浦 詠介　　櫻井 幸一

九州大学大学院システム情報科学研究科情報工学専攻
〒812-81 福岡市東区箱崎 6-10-1
Phone.092-641-1101 Fax.092-632-5204
sakurai@csce.kyushu-u.ac.jp

あらまし 文献 [R.El-Yaniv, A.Fiat, R.Karp, and G.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，区間毎に最適な取引を行なうことで双方向の取引を実現している．また [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なわれている．

しかしこれらのアルゴリズムは取引の際のコストを無視しており，コスト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．コスト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．

本稿ではコスト付きモデルに対するオフライン最適アルゴリズムの動作を解析し，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なった．

キーワード：経済ゲーム，為替交換，オンラインアルゴリズム，オフラインアルゴリズム，競合比

On designing on-line algorithms for money making with costs of exchanging

Eisuke DANNOURA *and* Kouichi SAKURAI

Department of Computer Science and Communication Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-01, Japan
Phone.092-642-4050 Fax.092-632-5204
Corresponding to sakurai@csce.kyushu-u.ac.jp

Abstract: On-line algorithms for money-making trading problem are investigated in [R.El-Yaniv, A.Fiat, R.Karp, and G.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, we invest the improved bidirectional algorithm in [Dannoura and Sakurai. IPSJ Transaction Vol.37, No.12 (1996)].

However, in these papers the costs of exchanging are ignored, and on-line algorithms for the model with costs 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 costs, design and improve the on-line algorithms for the model.

Keywords: financial game, money-trading, on-line algorithm, off-line algorithm, competitive ratio

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とり，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，あらかじめ全ての要求を受けとった上で，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 [Kar92]．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 [BLS92]，ロボット操縦 [BRS91，FFKRRV91，PY91] 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とり，各々に対応した処理，及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毎に新たな障害が見つかっていく，というモデルの下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．文献 [Cov91] で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，文献 [Ragh91] では統計的な敵 (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．このようなオンラインアルゴリズムの評価方法の一つとして競合比 (competitive ratio) と呼ばれるものがある [KMRS88]．これは，期間 T において得られた情報を基に何らかのコストがかかる行動をとる時に，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わすというものである．

文献 [EFKT92] では，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二つの通貨 (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なう単一方向取引アルゴリズムを各々の区間毎に適用し，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，為替交換に伴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コスト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コストの導入を行な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．つまり従来はコストが一切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，コスト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，コスト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なった．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なった．

2 従来の二通貨間為替交換アルゴリズム

文献 [EFKT92] 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」という二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，文献 [EFKT92] における為替交換問題のモデル及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

取引を行なう期間 $[0, T]$ において，円相場 x(円 / ドル，$m \le x \le M$，かつ m, M は知られているとする) は時刻 $t(t \in [0, T])$ の関数として表わされ，$[0, T]$ 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得るものとする．つまり，不連続点の発生に反応して取引を行なう場合，不連続点の発生した後の相場で取引は行なわれる．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得るので，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

文献 [EFKT92] では，単一方向アルゴリズムと双方向アルゴリズムという 2 つのアルゴリズムが提案されている．

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) のみを行ない，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な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な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なう．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 の時のドルの額を $D(x)$，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全て円に交換する
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なう
3. 規則 2. の条件で取引を行なう場合は，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 及び R は以下のように定義される．

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

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

定理 2.1 [EFKT92] 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 アルゴリズムの改良

本章では，[DS96] において我々が行なった，[EFKT92] の双方向アルゴリズムの改良について説明する．

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，二つの方向に交互に適用することで，双方向の取引を行な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，[EFKT92] で示された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) がわかる．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．

今回はこの点に注目し，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出来る単一方向取引アルゴリズムを構築したので以下に示す．

3.2 改良した単一方向取引アルゴリズム

今回改良し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(証明は後述)．

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全て円に交換する
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なう
3. 規則 2. の条件で取引を行なう場合は，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)

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

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

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

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

ただし，$\tilde{r}$ 及び $\tilde{R}$ 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)．

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

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

定理 3.1 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

証明 アルゴリズムに示された $D(x)$ の式から，$a \in [m, \tilde{r}^2 m]$ の場合の $Y(x)$ を求める．$x \in [a, \tilde{r}^2 m]$ では $D(x) = 1$ より $Y(x) = 0$ である．$x \in [\tilde{r}^2 m, M]$ の場合，

$$D'(x) = -\frac{1}{\tilde{r}(x - \tilde{r}m)}$$

が導かれる．$xD'(x) = -Y'(x)$(ドルの増加量と円の減少量の比は円相場の値となる)から，

$$Y'(x) = \frac{x}{\tilde{r}(x - \tilde{r}m)}$$

となり，

$$\begin{aligned} Y(x) &= 0 + \int_{\tilde{r}^2 m}^{x} \frac{t}{\tilde{r}(t - \tilde{r}m)} dt \\ &= \frac{x}{\tilde{r}} - \tilde{r}m(1 - \frac{1}{\tilde{r}} \ln \frac{x - \tilde{r}m}{\tilde{r}^2 m - \tilde{r}m}) \end{aligned}$$

である．ここで，$\tilde{r}mD(x) + Y(x) \geq \frac{x}{\tilde{r}}$ ($x \in [\tilde{r}^2 m, M]$)のとき等号成立)である．ここで $\tilde{r}mD(x) + Y(x)$ は，相場の最大値が x であり，相場が次の瞬間 $\tilde{r}m$ になり，そのまま終了した場合にこのアルゴリズムによって得られる円の額である．この時のオフラインアルゴリズムで得られる円は x であるので，$a \in [m, \tilde{r}^2 m]$ の場合の競合比は $\tilde{r}$ である．

同様に $a \in [\tilde{r}^2 m, M]$ では $\tilde{r}mD(x) + Y(x) \geq x/\tilde{R}$ が得られ，競合比は $\tilde{R}(\leq \tilde{r})$ である．

ゆえに，$a \in [m, M]$ かつ終了時の相場が $\tilde{r}m$ 以上の場合の競合比は $\tilde{r}$ 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

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

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，破線は従来の値(r)，実線は今回の値($\tilde{r}$)である．

定理 3.2 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．

証明 今回のアルゴリズムは $[\tilde{r}m, M]$ においては El-Yaniv らのアルゴリズムと同じ動作を行ない，$\tilde{r}$ という競合比が得られる．しかし実際の変動範囲は $[m, M]$ である．定理 3.1 の証明に示したように，今回のアルゴリズムでは

$$\tilde{r}mD(x) + Y(x) \geq \frac{x}{\tilde{r}}$$

($x \geq \tilde{r}^2 m$ の時に等号成立)が成り立つが，実際には，残りのドルを $x = m$ で円に変えなければならない可能性がある．ここで，$D(x), Y(x) \geq 0$ より

$$mD(x) + Y(x) \geq \frac{\tilde{r}mD(x) + Y(x)}{\tilde{r}}$$

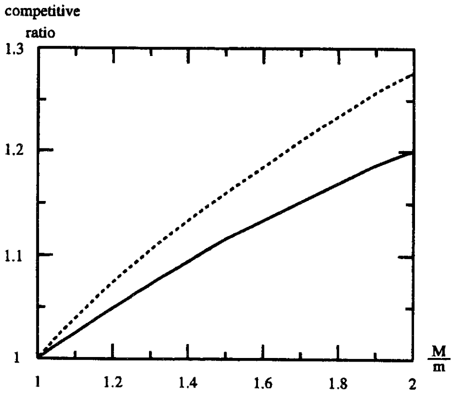

図 1: 競合比の比較

$(Y(x) = 0,$ すなわち $x \le \tilde{r}^2 m$ の時に等号成立) である．ゆえに，

$$mD(x) + Y(x) \ge \frac{x}{\tilde{r}^2}$$

$(x = \tilde{r}^2 m$ の時に等号成立) が成り立つ． □

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時，以下の定理が成り立つ．

定理 3.3 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．

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

この定理 **3.3** の証明は省略するので，必要ならば [DS96] を参照してほしい．

4 コスト付き為替交換問題

これまでに扱ってきた為替交換問題のモデルでは，通貨交換の際のコストは一切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上で，従来のモデルに対してコストを導入する．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なう．また，競合比の下限の解析も合わせて行なっていく．

4.1 相場の上下限が知られている連続的モデル

取引を行なう期間 $[0, T]$ において，ドル売円買の相場 x ($m \le x \le M$，かつ m, M は知られているものとする) は時刻 $t(t \in [0, T])$ の関数として表わされ，$[0, T]$ 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得るものとする．

4.2 コスト付き問題に対する最適なオフラインアルゴリズム

本節ではコスト付き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．

従来のモデルでの最適アルゴリズムは，円相場 x が極値をとるとき，かつそのときのみ取引を行なうという，単純かつ自明なものであった．しかしながらコスト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なったが，コスト付きモデルでは，この場合に利益よりもコストの方が大きくなり，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，コスト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，最適アルゴリズムを考える．

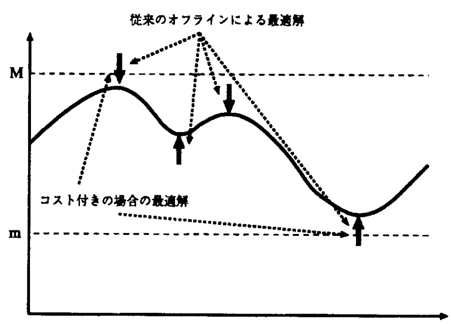

図 2: オフラインでの最適解の比較

ここで考察する最適オフラインアルゴリズムでは，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取った場合にのみ取引を行なうことから，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 $(s_1, b_1), (s_2, b_2), \cdots, (s_{n/2}, b_{n/2})$ という形で表現する．ここで i は最適なオフラインアルゴリズムによる取引回数であり，(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，幾つめの入力で行なったかを示す．(例えば，e_1 でドル売円買，e_2 で円売ドル買を行なうならば，$(s_1, b_1) = (1, 2)$ と表す．) また，最適なオフラインアルゴリズムによる解を一意に定めるため，同じ金額が得られる最適な出力が複数存在する場合は以下の制約に従うものとする．

制約

1. 取引回数の少ない方を解とする
2. 来るだけ早く取引を行なうものを解とする

最適なオフラインアルゴリズムを以下に示す．

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．

以下にこのオフラインアルゴリズムの最適性の証明を行なう．

まずは，この (空集合でない) 出力が最適なオフラインアルゴリズムの解であるための必要十分条件を示す．

条件

1. 任意の $J(0 \le J < s_1)$ において $e_J < e_{s_1}$
2. 任意の J_1, J_2 $(0 \le J_1 \le J_2 \le s_1)$ において $e_{J_1} \le ce_{J_2}$
3. 任意の $j(1 \le j \le n/2)$ において $e_{s_j} > ce_{b_j}$
4. 任意の $J(s_j \le J < b_j,\ 1 \le j \le n/2)$ において $e_{b_j} < e_J \le e_{s_j}$
5. 任意の $J_1, J_2(s_j \le J_1 \le J_2 \le b_j, 1 \le j \le n/2)$ において $ce_{J_1} \ge e_{J_2}$
6. 任意の $j(1 \le j \le n/2)$ において $e_{s_{j+1}} > ce_{b_j}$
7. 任意の $j, J(1 \le j < n/2,\ b_j \le J < s_{j+1})$ において $e_{b_j} \le e_J < e_{s_{j+1}}$
8. 任意の j, J_1, J_2 $(b_j \le J_1 \le J_2 \le s_{j+1},\ 1 \le j \le n/2)$ において $e_{J_1} \le ce_{J_2}$
9. 任意の $J(b_{n/2} \le J \le k)$ において $e_{b_{n/2}} \le e_J$
10. 任意の j, J_1, J_2 $(b_{n/2} \le J_1 \le J_2 \le k)$ において $e_{J_1} \le ce_{J_2}$

証明. まずは上に示した条件が必要条件であることを示すために，解が上の条件が満たさない場合は，より大きな金額を得る解が存在することを示す．

1. $e_J > e_{s_1}$ の場合，(s_1, b_1) と取引を行なうよりも (J, b_1) と行なった方が大きな金額が得られる．等号成立の場合は制約 1. に反する．
2. $e_{J_1} > ce_{J_2}$ の場合，(J_1, J_2) という取引を加えることでより大きな金額が得られる．
3. $e_{s_j} < ce_{b_j}$ の場合，(s_j, b_j) の取引により損失となり，$e_{s_j} = ce_{b_j}$ の場合は制約 1. に反する．
4. $e_J > e_{s_j}$ の場合，(s_j, b_j) と取引を行なうよりも (J, b_j) の方が大きな金額が得られる．$e_{b_j} > e_J$ の場合，(s_j, b_j) と取引を行なうよりも (s_j, J) の方が大きな金額が得られる．$e_{b_j} = e_J$ の場合は制約 1. に反する．
5. $ce_{J_1} < e_{J_2}$ の場合，(s_j, b_j) と取引を行なうよりも $(s_j, J_1), (J_2, b_j)$ の方が大きな金額が得られる．$ce_{J_1} = e_{J_2}$ の場合は制約 1. に反する．
6. $e_{s_{j+1}} < ce_{b_j}$ の場合，$(s_j, b_j), (s_{j+1}, b_{j+1})$ と取引を行なうよりも (s_j, b_{j+1}) の方が大きな金額が得られる．等号成立の場合は制約 1. に反する．

7. $e_{b_j} > e_J$ の場合 (s_j, b_j) よりも (s_j, J), $e_J > e_{s_{j+1}}$ の場合は (s_{j+1}, b_{j+1}) よりも (J, b_{j+1}) と行なった方が金額が得られる．$e_J = e_{s_{j+1}}$ の場合は制約 1. に反する．

8. $e_{J_1} > ce_{J_2}$ の場合，(J_1, J_2) という取引を加えることでより大きな金額が得られる．

9. $e_{b_{n/2}} > e_J$ の場合，$(s_{n/2}, b_{n/2})$ と取引を行なうよりも $(s_{n/2}, J)$ と行なった方が大きな金額が得られる．

10. $e_{J_1} > ce_{J_2}$ の場合，(J_1, J_2) という取引を加えることでり大きな金額が得られる．

次に，この必要条件を満たす解が 1 つしか存在し得ないことを示す．

まずは (s_1, b_1) が 1 つに定まるか，解が空集合であるかのいずれかであることを示す．条件 1. のみを満たす値を，s_1 の値の候補の集合として考える．この集合を $S_1 = \{x_1, x_2, \cdots, x_l\}(x_1 < x_2 < \cdots < x_l, e_{x_1} < e_{x_2} < \cdots < e_{x_l})$ とする．このとき，$x_1 = 1$ である．ゆえに $l \geq 1$ である．

ここで，$l = 1$ の場合と $l \geq 2$ の場合に分けて考える．

1. $l = 1$ の場合

ここで，$e_J < e_1/c$ を満たす J が $J \geq 2$ に存在するか否かで場合分けを行なう．

(a) 存在しない場合

$l = 1$ より e_1 が最大値である．定理より，解は空集合となる．

(b) 存在する場合

$s_1 = 1$ となることを示す．$s_1 = 0$ とすると条件 4. に反し，$s_1 \geq 2$ とすると条件 1. に反するため，$s_1 = 1$ または，解が空集合のいずれかとなる．そこで，$s_1 = 1$ として対応する b_1 が存在するかを考える．

条件 3.4. を満たす b_1 の候補を $B_1 = \{y_1, y_2, \cdots, y_i\}$ $(y_1 < y_2 < \cdots < y_i,\ e_{y_1} > e_{y_2} > \cdots > e_{y_i})$ とする．

このとき，$ce_{y_j} < e_y$ $(y_j \leq y \leq y_{j+1}, j < i)$ を満たす j, y が存在するか否かを考える．

i. (j, y) の組が 1 つ以上存在する場合

各組の中で j が最小のものを j とすると，$b_1 = y_j$ とすることで条件 5. が満たされる．

ii. (j, y) の組が存在しない場合

y_i より小さな値が以降に存在しないことから条件 5. が満たされ，$b_1 = y_i$ となる．

2. $l \geq 2$ の場合

s_1 の候補の集合 $S_1 = \{x_1, x_2, \cdots, x_l\}$ を考え，これに対して $e_{x_j} > ce_x$ $(x_j \leq x \leq x_{j+1}, j < l)$ を満たす (j, x) の組が存在するか否かで場合分けを行なう．

(a) (j, x) の組が 1 つ以上存在する場合

最小の j に対する x_j を s_1 とみなし，**1.** と同様に (s_1, b_1) が一意に定まることが証明される．

(b) (j, x) の組が存在しない場合

定理より解が空集合となる．

以上より，(s_1, b_1) は一意に定められる．

次に，(s_j, b_j) が一意に定められたとき，(s_{j+1}, b_{j+1}) も一意に定められる，または存在しないことを証明する．

まずは $ce_{b_j} < e_J$ なる $J(J \leq b_j)$ が存在するか否かで場合分けを行なう．

1. 存在する場合

条件 6. のみを満たす s_{j+1} の値の候補を S_{j+1} とする．これに対し，S_1 のときと同様に，(s_{j+1}, b_{j+1}) が一意に定まる，もしくは存在しないことが証明される．

2. 存在しない場合

b_j が一意に定まっているので，条件 6. か条件 9. が成り立つ．$ce_{b_j} < e_J$ なる $J(J \leq b_j)$ が存在しないので，b_j は条件 6. を満たすことが出来ない．ゆえに条件 9. より b_j 以降の解は存在しない．またこの時条件 10. も成り立つ．

以上より，(s_{j+1}, b_{j+1}) も一意に定められる．

よって，最適なオフラインアルゴリズムの解であるための必要条件を満たす解は，このオフラインアルゴリズムの出力解以外に存在しない．ゆえに，このオフラインアルゴリズムは最適である．□

これにより，コスト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わかる．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 コスト付き問題に対するオンラインアルゴリズム

本節では，コスト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毎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 $cr_c m$ 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既に交換が終っていることがわからない).

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なわない．それ以外は，全て円 (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なう．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 の時のドルの額を $D(x)$，円の額を $Y(x)$ とし，初期値を $x = a,\ D = 1, Y = 0$ とする.

$a \in [m, cr_cm]$

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

$a \in (cr_cm, M]$

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

ただし，r_c 及び 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．)

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

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 4.1 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. まずは各区間毎に 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な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なう値) 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なう値は最小でも cm であることから，この二区間で，最適オフラインアルゴリズムは高々 r_c 倍に増やすことしかできない．よってこの場合，何もしなくとも二区間での競合比が 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，[EFKT92] 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，各区間毎に 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 回の交換を行なう場合，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 $\frac{x_*D_*/r_c}{cx_*} = D/(c{r_c}^{i+1})$ である．よって，競合比は $c{r_c}^{i+1}$ である. □

ここで，$c = 1$(コストなし) のとき，$r_c = r$ であり，このアルゴリズムは [EFKT92] のオンライン双方向アルゴリズムと一致する.

4.4 コスト付き双方向アルゴリズムの改良

本節では，コスト付き双方向アルゴリズムの改良を行なう.

本節で示す双方向アルゴリズムは，[DS96] で我々が行なった改良 (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，[DS96] 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 $c{\tilde{r_c}}^2m$ 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なわない．それ以外は，全て円(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．

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

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なう．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 の時のドルの額を $D(x)$，円の額を $Y(x)$ とし，初期値を $x = a$, $D = 1$, $Y = 0$ とする．

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

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

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

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

このとき，以下の定理が成り立つ．

定理 4.2 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なう時刻を $t_i (t_i < t_{i+1}, t_0 = 0, t_k = T)$ 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なうことがプレイヤーにわかった時刻(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，区間毎に定義するために，以下のように x の関数として拡張する．

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

$$\tilde{R}_n(x) = \begin{cases} \tilde{R}_c(x) & n \text{ が偶数のとき} \\ \tilde{R}_c(\frac{mM}{x}) & n \text{ が奇数のとき} \end{cases}$$

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．

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

(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．

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

上の三つの式より以下の式が導かれる．

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

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全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 $c = 1$(コストなし) のとき，このアルゴリズムは我々が改良したオンライン双方向アルゴリズムと一致する．

4.5 コスト付き問題に対する競合比の下限

El-Yaniv らは為替交換問題に対する競合比の下限を，制限付きの入力に対する最適アルゴリズムを考察することで示したが，同じ入力を考察することでコスト付き問題に対する下限も得ることが出来る．この入力を図3に示す．

この入力に対する最適なオンラインアルゴリズムは，従来の単一方向アルゴリズムを，$x \in [cm, M]$ として適用したものであり，これにより下限として $\tilde{r_c}^{2/n}$ ($\tilde{r_c} = \ln \frac{\frac{M}{cm}-1}{\tilde{r_c}-1}$) が得られる．

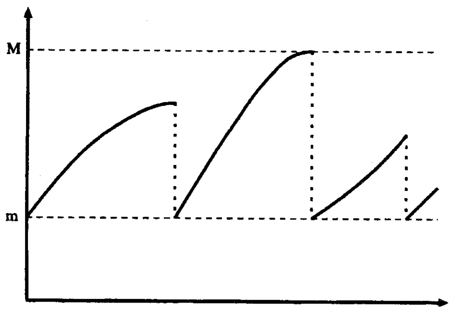

図 3: 従来の下限を示す制限

5 今後の課題

今回，コスト付き問題に対するオンラインアルゴリズムを提案し，また競合比の下限についても解析を行なったが，上下限の間には大きな開きが存在する．今後は上下限両面から，競合比の改善を計りたい．

参考文献

[BLS92] A.Borodin, N.Linial, and M.Saks. An optimal online algorithm for metrical task system. Journal of the ACM, 39:745-763, 1992.

[BRS91] A.Blum, P.Raghavan, and B.Schieber. Navigating in unfamiliar geometric terrain. In Proceeding of the 23rd Annual ACM Symposium on Theory of Computing, pages 494-504, 1991.

[Cov91] T.M.Cover. UNIVERSAL PORTFORIOS, In Journal of Mathematical Finance, Vol.1, No.1, pages1-29, January 1991.

[CCEKL95] Andrew Chou, Jeremy Cooperstock, Ran El-Yaniv, Michael Klugerman and Tom Leighton. The Statistical Adversary Allows Optimal Money-Making Trading Strategies. In Proceeding of SODA'95, 1995.

[DS96] 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, Dec. 1996.

[EFKT92] R.El-Yaniv, A.Fiat, R.Karp and G.Turpin. Competitive Analysis of Financial Games. In Proceeding of the 33rd Annual Symposium on Foundations of Computer Science, pp.327-333, 1992.

[EK93] Ran El-Yaniv and Richard M.Karp. The Mortgage Problem. Proceedings of the 2nd Israel Symposium on Theory and Computing Systems, 1993.

[FFKRRV91] A.Fiat, D.P.Foster, H.J.Karloff, Y.Rabani, Y.Ravid and S.Vishwanathan. Competitive algorithms for layered graph traversal. In Proceeding of the 32nd Annual IEEE Symposium on Foundations of Computer Science, pages 288-297, 1991.

[Kar92] Richard M.Karp. On-Line Algorithms Versus Off-Line Algorithms: How Much is it Worth to know the Future?, In Proceeding of IFIP, 1992.

[KMRS88] A.R.Karlin, M.S.Manasse, L.Rudolph, and D.D.Sleator. Competitive snoopy caching. Algorithmica,3(1):70-119,1988.

[PY91] C.H.Papadimitriou and M.Yanakakis. Shortest paths without a map. Theoretical Computer Science, 84:127-150, 1991.

[Ragh91] P.Raghavan. A statistical adversary for on-line algorithms. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 7;79-83, 1991.

[RS94] P.Raghavan and M.Snir. Memory versus randomization in on-line algorithms. IBM Journal of Research and Development, 38:683-707, 1994.