オンライン問題のクラス分け

1L－2

内田 敦[†] 宮崎 修一[‡] 岩間 一雄[‡]

[†] 京都大学工学部 [‡] 京都大学大学院情報学研究科

1 はじめに

オンライン問題は，次々に与えられる入力に対して，未来の知識なしに各時点で答を出力することを要求する問題であり，株取引などといった，かなり実用性の高い問題も知られている[2, 3]．オンライン問題に対するアルゴリズムをオンラインアルゴリズムと言う．オンラインアルゴリズムの良さは，同じ問題を解くオフラインアルゴリズム(入力を全て読み終ってから答を出すアルゴリズム)の出力する答との比(競合比)で測られる．

本研究ではオンライン問題のクラス分けを試みる．最適化問題に関しては，問題を解く近似アルゴリズムの性能によりクラス分けが行われており，それぞれのクラスの完全問題も知られている[1]．本稿ではそれを模倣し，問題の持つオンラインアルゴリズムの競合比によりクラス分けを行う．本稿ではまず，問題のクラスを定義し，それに則した変換及び完全性を定義する．さらに人工的な完全問題の存在も示す．

2 オンライン問題及びオンラインアルゴリズム

オンライン問題は，外部から連続した要求の列を受けとり，その要求それぞれに対して即座に行動を求める問題である．本稿では，オンライン問題を以下の様に定義する．

定義1．オンライン問題は以下の5つ組 (A, B, f, g, f) である．

A： $A = \{a_1, \cdots, a_n, \cdots\}$ は入力集合で高々可算な集合．各 a_i は各々一つの要求を表す．

B： $B = \{b_1, \cdots, b_n, \cdots\}$ は出力集合で高々可算な集合．各 b_i は各々一つの行動を表す．

f： A^n から $\{true, false\}$ への計算可能な関数．ただし，$f(a_{i_1} \cdots a_{i_n}) = true$ ならば，任意の $k \leq n$ に対して $f(a_{i_1} \cdots a_{i_k}) = true$ である．

g： $(A^n)' \times B^n$ から $\{true, false\}$ への計算可能な関数．$(A^n)' \subseteq A^n$ は $f(x) = true$ となる x の集合．ただし，$g(a_{i_1} \cdots a_{i_n}, b_{j_1} \cdots b_{j_n}) = true$ ならば，任意の $k \leq n$ に対して $g(a_{i_1} \cdots a_{i_k}, b_{j_1} \cdots b_{j_k}) = true$ である．また，任意の $a \in A$ に対して $g(a, b) = true$ となる $b \in B$ が存在する．$f(a_{i_1} \cdots a_{i_n}) = true$ かつ $g(a_{i_1} \cdots a_{i_{n-1}}, b_{j_1} \cdots b_{j_{n-1}}) = true$ ならば，$\{b_k \mid b_k \in B$ かつ $g(a_{i_1} \cdots a_{i_n}, b_{j_1} \cdots b_{j_{n-1}} b_k) = true\} \neq \emptyset$ である．

h： $A^n \times B^n$ の部分集合から正の整数への計算可能な関数．$g(a_{i_1} \cdots a_{i_n}, b_{j_1} \cdots b_{j_n}) = true$ なものの集合が h の定義域である．

f は入力列が適切な場合に $true$ を，不適切な場合に $false$ を返す関数である．g は入力列に対して出力列が適正であるかを規定する関数である．また，h はコスト関数である．

次に，決定性チューリング機械(DTM)を用いてオンラインアルゴリズムを定義する．このため，問題の入出力集合をアルファベットの有限集合で符合化する．すなわち，問題 $F = (A, B, f, g, h)$ に対して，$A, B \subseteq \{0, 1\}^*$ である．また，入力は入力テープに A の要素を # で区切って与えられるものとする．入力の最後には # が連続して2つ書かれているものとする．例えば，入力は 001#11#110## のようにテープ上に書かれる．

定義2．$F = (A, B, f, g, h)$ とする．このとき以下の動作をする DTM を問題 F に対するオンラインアルゴリズムと呼ぶ．

#'# と連続して読むまで以下を繰り返す．

i) 入力を(必要なら先頭から)# まで読み，今読んだ # を #' に書き換える．

ii) 入力を先読みせずに，$g(a_{i_1} \cdots a_{i_k}, b_{j_1} \cdots b_{j_k}) = true$ となる b_{j_k} を出力テープに書き，# を出力テープに書く．

#'# と連続して読むと入力は終了なので，出力列の後にコスト関数 h の値を出力する．

次にオンラインアルゴリズムの性能の尺度となる競合比を定義する．ここでは最小化問題を考える．

定義3．オンライン問題 $F = (A, B, f, g, h)$ の入力の系列 x を $x = a_{i_1} \cdots a_{i_n} (a_{i_j} \in A)$ とする．（ただし $f(x) = true$.）このとき $|x| = n$ と定義する．まず，x に対する最適解 $opt_F(x)$ を $opt_F(x) = \min_{y \in S(x)} \{h(x, y)\}$ と定義する．ただし，$S(x) = \{b_{j_1}, \cdots, b_{j_n} \mid b_{j_k} \in B$ かつ

Classification of on-line problems
Atsushi Uchida, Shuichi Miyazaki and Kazuo Iwama
Kyoto University

$g(x, b_{j_1}\cdots b_{j_n}) = true\}$ である．

次に，入力列 x，オンラインアルゴリズム T に対する競合比を $CR_F(x,T) = h(x,T(x))/opt_F(x)$ と定義する．さらに，入力列の長さ n と，オンラインアルゴリズム T に対する競合比を，$CR_F(n,T) = \sup_{|x|=n} CR_F(x,T)$ と定義する．$\forall n$ に対して $CR_F(n,T) \leq a(n)$ が成り立つ時，T を競合比が $a(n)$ のアルゴリズムと呼ぶ．

ここで，以下の2つのクラスを定義する．

定義4．全てのオンライン問題のクラスをOLP(On Line Problems) とし，競合比が定数のオンラインアルゴリズムを持つオンライン問題のクラスをCCR(Constant Competitive Ratio) とする．

3 OLP 完全性

本節では，オンライン問題間の競合比を保存する変換を定義する．

定義5．以下の性質を満たすDTM M_1，M_2 と関数 c が存在するとき，オンライン問題 $P = (A_P, B_P, f_P, g_P, h_P)$ からオンライン問題 $Q = (A_Q, B_Q, f_Q, g_Q, h_Q)$ へ変換可能であるという．

M_1 ： 関数 $t_1(A_P^n \to A_Q^n$ の関数) を計算するDTM.
M_2 ： 関数 $t_2(B_Q^n \to B_P^n$ の関数) を計算するDTM.
c ： 有理数から有理数への関数.

ただし，M_1 及び M_2 は，入出力の読み書きに関しては，定義2のDTMと同様の動作をする．また，t_1, t_2, c は以下の条件を満たす．ただし，(2)～(4) では，$x = a^P_{i_1}\cdots a^P_{i_n}$ $(a^P_{i_k} \in A_P)$，$y = b^Q_{j_1}\cdots b^Q_{j_n}$ $(b^Q_{j_k} \in B_Q)$ とする．(1) $t_1(a^P_{i_1}\cdots a^P_{i_n}) = a^Q_{j_1}\cdots a^Q_{j_n}$ ならば，任意の $k \leq n$ に対して $t_1(a^P_{i_1}\cdots a^P_{i_k}) = a^Q_{j_1}\cdots a^Q_{j_k}$，$t_2(b^P_{i_1}\cdots b^P_{i_n}) = b^Q_{j_1}\cdots b^Q_{j_n}$ ならば任意の $k \leq n$ に対して $t_2(b^P_{i_1}\cdots b^P_{i_k}) = b^Q_{j_1}\cdots b^Q_{j_k}$．(2) 任意の x に対して，$f_P(x) = true$ ならば $f_Q(t_1(x)) = true$．(3) 任意の x，y に対して，$g_Q(t_1(x), y) = true$ ならば，$g_P(x, t_2(y)) = true$．(4) 任意の x, y に対して，$h_Q(t_1(x), y)/opt_Q(t_1(x)) \leq \gamma$ ならば，$h_P(x, t_2(y))/opt_P(x) \leq c(\gamma)$．

定理1．オンライン問題 P がオンライン問題 Q に変換可能であり，かつ，オンライン問題 Q がクラスCCRに属するならば，オンライン問題 P もクラスCCRに属する．(証明略)

次に，先ほど定義した変換と2つのクラスを用いて，オンライン問題の完全性を定義する．

定義6．オンライン問題 Q が $Q \in$ OLPであり，かつ任意のオンライン問題 $P(P \in$ OLP$)$ が Q に変換可能ならば問題 Q はOLP-完全であるという．

性質1．任意のオンライン問題に，入出力テープを一つずつ持ち，以下の動作をする非決定性チューリング機械(NTM) を対応付けることができる．
入出力の読み書きに関しては，定義2のDTMと同様の動作をする．ただし出力は非決定的にゲスする．入出力の読み書きが終了したら，最終的な入出力系列に対して g を計算する．$false$ ならば終了．$true$ ならば h の値を出力する．

ここで，オンライン問題 U を定義する．前述のような動作をするNTM M には，逆にあるオンライン問題 $F = (A_F, B_F, f_F, g_F, h_F)$ が対応付けられることに注意されたい．U を以下の $(A_U, B_U, f_U, g_U, h_U)$ で定義される問題とする．

A_U ： $A_U = \{(a_i, M_P) \mid M_P$ は上述の動作をするNTMで，それに対応する問題を $P = (A_P, B_P, f_P, g_P, h_P)$ とする．また $a_i \in A_P\}$.
B_U ： $B_U = \{0,1\}^*$.
f_U ： $f_U((a_{i_1}, M_{i_1})\cdots(a_{i_n}, M_{i_n}))$ は，$i_1 = \cdots = i_n$ かつ $f_F(a_{i_1}\cdots a_{i_n}) = true$ の時及びその時に限り $true$.
g_U ： $g_U((a_{i_1}, M_l)\cdots(a_{i_n}, M_l), b_{j_1}\cdots b_{j_n})$ は，$g_F(a_{i_1}\cdots a_{i_n}, b_{j_1}\cdots b_{j_n}) = true$ の時及びその時に限り $true$.
h_U ： $h_U((a_{i_1}, M_l)\cdots(a_{i_n}, M_l), b_{j_1}\cdots b_{j_n}) = h_F(a_{i_1}\cdots a_{i_n}, b_{j_1}\cdots b_{j_n})$.

つまり，U は入力として，F とそれに対する入力列が与えられ，F と同じ出力列を出力する問題である．U の入力列の解集合は対応する入力列の F に対する解集合と一致しているため，ある意味 F を模倣していると考えられる．

定理2．U はOLP完全である．

証明．U はオンライン問題の定義を満たしており，$U \in$ OLPである．次に，任意のオンライン問題 F から U への変換を考える．M_F を F に対応付けられたNTMとする．ここで t_1 を $a_1\cdots a_n$ を $(a_1, M_F)\cdots(a_n, M_F)$ へ移す関数，t_2 を恒等写像する関数とし，c を恒等写像，$c(\gamma) = \gamma$ とする．U 及び t_1，t_2，c の定義から，t_1，t_2，c は定義5を満たす．従って U はOLP-完全である．□

参考文献

[1] P. Crescenzi and A. Panconesi, "Completeness in approximation classes," *Information and Computation*, vol.93, pp.241-262, 1991.

[2] R. El-Yaniv, A. Fiat, R. Karp and G. Turpin, "Competitive analysis of financial games," *proc. FOCS 92*, pp.327-333, 1992.

[3] R. M. Karp, "On-line algorithms versus off-line algorithms , How much is it worth to know the future?," *proc. IFIP 12th World Computer Congress*, Vol.1, pp.416-429, 1992.