情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2010-AL-129 No.3
2010/3/5

確率的アルゴリズムの場合の No Free Lunch 定理の厳密な証明

橋 谷 祐 司[†1]　櫻 川 貴 司[†2]

本稿では，確率的アルゴリズムを用いた場合における No Free Lunch 定理（NFL 定理）を厳密に証明する．NFL 定理は「どのような最適化問題に対しても効率良く解を導き出す万能の探索アルゴリズムは存在しない」ということが帰結される定理で，多くの最適化問題や探索アルゴリズムについての研究論文に引用されている．本稿では確率的アルゴリズムを形式的に定義し，いくつかの補題を示した上で確率的アルゴリズムの場合の NFL 定理を確率論によって厳密に証明する．

A Rigorous Proof of No Free Lunch Theorem for Stochastic Algorithms

Yuji Hashiya [†1] and Takashi Sakuragawa[†2]

In this paper, a rigorous proof of No Free Lunch Theorem (NFLT) for stochastic algorithms is presented. NFLT shows that there is no algolithm which can solve every optimization problem efficiently. A lot of research papers on optimization problems and search algorithms refer to it. We give a formal definition of stochastic algorithm, prove some lemmas and then complete a regourous proof of NFLT for stochastic algorithms using probability theory.

1. はじめに

本稿では，最適化問題を解く探索アルゴリズムに関する定理である No Free Lunch 定理（以下，NFL 定理）をテーマとして取り上げる．主な目的は，確率的アルゴリズムの場合での NFL 定理の厳密な証明を与えることである．NFL 定理とは「最適化問題について探索空間と評価関数の取りうる値が有限集合である場合，どのような探索アルゴリズムであっても平均的な探索効率は同じになる」ということを意味する定理で，Wolpert と Macready によって提唱された[1)]．最適化問題を解く際，どのような問題に対しても効率よく解を見つけ出す汎用の探索アルゴリズムがあれば，解くべき問題に対して個々にアルゴリズムを構築しなくとも探索することができるので都合が良いといえる．しかし NFL 定理から，そのような都合の良い万能探索アルゴリズムが存在しないことが帰結される．

1) では決定的アルゴリズムにおける NFL 定理の証明が与えられているが，その証明は式変形に頼っており，NFL 定理が成立する理由を直感的に理解しづらい面がある．柳井・伊庭はこの点を指摘し，Wolpert らの定義を整備した上で決定的アルゴリズムの場合での NFL 定理の厳密な別証明を与えた[2)]．また，1) では確率的アルゴリズムにおいても NFL 定理が成り立つことが示唆されているが，具体的な証明は与えられていない．以上を踏まえ，2) で与えられた証明を確率論的にさらに厳密にした形で紹介することと，確率的アルゴリズムにおける NFL 定理の厳密な証明を具体的に与えることを本稿の目的とする．

まず，第 2 節において本稿で扱う基本的な用語の説明を行う．一般に用いられるよりも用語の意味を限定して用いることがあるので注意する．

第 3 節では NFL 定理の紹介，証明，考察を行う．NFL 定理の説明のため，小節 3.1 において大まかな定義を述べ，NFL 定理の背景と目的について説明する．続いて小節 3.2 では決定的アルゴリズムを用いた場合の NFL 定理について述べる．ここでは柳井・伊庭[2)] の与えた証明に基づき，いくつかの箇所を変更して NFL 定理の証明を行う．小節 3.3 では確率的アルゴリズムを用いた場合の NFL 定理について述べる．ここでは 1) に従って 2) に対応した形で確率的アルゴリズムを定義し，いくつかの補題を用いて確率的アルゴリズムにおける NFL 定理に厳密な証明を与える．そして第 4 節で考察を行う．

2. 準　　備

本稿で扱う基本的な用語を説明する．一部の用語についての詳しい定義は第 3 節を参照．

最適化問題　ある集合 X について，X に含まれるそれぞれの要素に対しある関数によって評価値が与えられるとき，評価値が最大となる要素を見つける問題を本稿において最適化問題と呼ぶ．ここで，X に含まれるそれぞれの要素を**解候補**，各解候補に評価値を与える関数を**評価関数**，評価値が最大となる解候補を**最適解**と呼ぶ．評価値の集合は実

†1 京都大学総合人間学部, hashiya@sojin06.mbox.media.kyoto-u.ac.jp
Faculty of Integrated Human Studies, Kyoto University
†2 京都大学大学院人間・環境学研究科, sakura@i.h.kyoto-u.ac.jp
Graduate School of Human and Environmental Studies, Kyoto University

数全体の集合などの全順序集合を想定しているが，評価値の集合が一般の有限集合であっても NFL 定理は成立する．

インスタンス 一般的な問題に対し，その具体的な個々の問題例をインスタンスという．インスタンスに評価関数が対応する．例えば巡回セールスマン問題に対し，京都市内のすべての寺社を参拝する最短経路を求める問題は一つのインスタンスである．本来の NFL 定理ではあらゆるインスタンスに対応した汎用の探索アルゴリズムについて考えるため，すべての評価関数を考える．

探索 最適化問題において最適解を見つける動作を探索と呼ぶ．

探索履歴 既に探索した解候補とそれらの評価値から成る列．

探索アルゴリズム 探索を行うアルゴリズム．本稿でいう探索アルゴリズムとは次のように抽象化したものである．[1] 探索アルゴリズムはまず初めに一つの解候補を調べ，その後はこれまでに探索した解候補とそれらの評価値のみによって次に探索する解候補を一つずつ選び調べていく．ただし一度調べた解候補を再び調べることはないものとする．探索アルゴリズムが k 個の解候補を調べた状態を **$\boldsymbol{k}$ ステップまで探索した**という．本稿では**決定的アルゴリズム**と**確率的アルゴリズム**を考える．

決定的探索アルゴリズム 過去の探索履歴から次に探索する解候補を一意的に決定する探索アルゴリズム．2 つの決定的アルゴリズム a, a' について，同じ探索履歴に対し次に探索する解候補が同じであれば a, a' を同じアルゴリズムであるとみなす．

確率的探索アルゴリズム 過去の探索履歴から解候補全体の集合上の確率分布が決まり，その確率分布により次に探索する解候補が確率的に決まる探索アルゴリズム．ただし，過去に探索した解候補が選ばれる確率は 0 であるとし，一度調べた解候補を再び調べることはないとする．2 つの確率的アルゴリズム σ,σ' について，同じ探索履歴に対し次に探索する解候補の確率分布が同じであれば σ,σ' を同じアルゴリズムであるとみなす．

3. NFL 定理

本節において，NFL 定理の背景を説明し，NFL 定理で用いる記号を定義する．そして，いくつかの補題を示した上で決定的アルゴリズムと確率的アルゴリズムそれぞれの場合における NFL 定理の証明を行う．

3.1 NFL 定理とは

NFL 定理は，「最適化問題について探索空間と評価関数の取りうる値が有限集合である場合，どのような探索アルゴリズムであっても平均的な探索効率は同じになる」ことを帰結する定理である．以下，NFL 定理の証明に先立ち，その背景と概要を示す．なお，記号の用い方については柳井・伊庭の論文[2] を参考にさせていただいた．

3.1.1 大まかな定義

説明のため，本稿で用いる記号の大まかな定義を以下に示す．

X すべての解候補の成す有限集合．

Y すべての解の評価値の成す有限集合．

F 評価関数全体の集合．$F = \{f : X \to Y\}$．

D_k^X $1 \le k \le |X|$ として，k ステップまで探索を行ったときにあり得る解候補列全体の集合．$D_k^X \subseteq X^k$ であり，一つの解候補列に同じ解候補が 2 回以上現れることはない．

D_k^Y $1 \le k \le |X|$ として，k ステップまで探索を行ったときにあり得る解の評価値列全体の集合．$D_k^Y = Y^k$

D_k $1 \le k \le |X|$ として，k ステップまで探索を行ったときにあり得る探索履歴全体の集合．$D_k = D_k^X \times D_k^Y$

A 決定的探索アルゴリズム全体の集合．有限集合となる．

Σ 確率的探索アルゴリズム全体の集合．これは通常のプログラミング言語を一つ固定した場合のプログラム全体の集合よりも濃度が高くなり，非可算無限集合となる．

$F(d_k)$ 探索履歴 $d_k \in D_k$ を満たす評価関数の集合．$F(d_k) \subseteq F$

$S_k(a, f)$ アルゴリズム $a \in A$ と評価関数 $f \in F$ から決まる探索履歴．$S_k(a, f) \in D_k$

$R_k(a, d_k^Y)$ アルゴリズム $a \in A$ と評価値列 $d_k^Y \in D_k^Y$ から決まる探索履歴．$R_k(a, d_k^Y) \in D_k$

ここで S_k と R_k は決定的アルゴリズムの場合にのみ定義される．

$S_k(a, f)$ については，アルゴリズムによって最初に探索する解候補が決まっており，その評価値は f によって決まる．よって次に探索する解候補が決まり，以下同様にして k ステップの探索履歴が一意的に決まる．

$R_k(a, d_k^Y)$ については，アルゴリズムによって最初に探索する解候補が決まっており，その評価値として d_k^Y の最初の要素をアルゴリズムに返す．すると次に探索する解候補が決まり，以下同様にして解候補列が一意的に決まる．すなわち探索履歴が一意的に決まる．さらに，任意の $f \in F(R_k(a, d_k^Y))$ について $R_k(a, d_k^Y) = S_k(a, f)$ が成り立ち，これを S_k と R_k の**整合性**と呼ぶ[2]．このことは後に補題 3 として与える．

要素の数については，以下のようになる．

$$|F| = |Y|^{|X|} \qquad |D_k^Y| = |Y|^k \qquad |F(d_k)| = |Y|^{|X|-k}$$

解候補集合 X と評価値集合 Y はともに有限なので，評価関数の集合 F も有限となる．

Vol.2010-AL-129 No.3
2010/3/5

$F(d_k)$ については，X のうち k 個の解候補は d_k によって f で写った先が決まっているため，残り $|X|-k$ 個の解候補の写り先の組合せだけ d_k を満たす評価関数が存在する．つまりその個数は $|Y|^{|X|-k}$ 個となる．

Ω を根元事象全体の集合とし，Ω から決まる確率変数を次のように定める．P_a と P_σ はそれぞれ決定的アルゴリズム a と確率的アルゴリズム σ を用いたときの確率測度を表す．

$\mathcal{D}_k^X$ ：k ステップまで探索したときの解候補列を表す確率変数．
$\mathcal{D}_k^Y$ ：k ステップまで探索したときの評価値列を表す確率変数．
$\mathcal{D}_k$ ：k ステップまで探索したときの探索履歴を表す確率変数．
$\mathcal{X}_k$ ：k ステップ目に探索する解候補を表す確率変数．
$\mathcal{F}$ ：評価関数を表す確率変数．

通常通り確率測度の式を略記し，$P_a(\{\omega \in \Omega \mid \mathcal{D}_k^X(\omega) = d_k^X\})$ のことを $P_a(\mathcal{D}_k^X = d_k^X)$ と書くなどする．例えば，アルゴリズム a を用いて評価関数 f である問題の探索を k ステップ行った結果，評価値列が d_k^Y となる条件付き確率は $P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f)$ と表される．

正整数 $m \leq |X|$ を探索の最大ステップ数とし，$1 \leq k \leq m$ とすると，

$$\mathcal{D}_k^X \overset{\text{a.s.}}{=} (\pi_1(\mathcal{D}_m^X), \pi_2(\mathcal{D}_m^X), \ldots, \pi_k(\mathcal{D}_m^X)) \qquad \mathcal{D}_k \overset{\text{a.s.}}{=} (\mathcal{D}_k^X, \mathcal{D}_k^Y)$$
$$\mathcal{D}_k^Y \overset{\text{a.s.}}{=} \mathcal{F}(\mathcal{D}_k^X)(= (\mathcal{F} \circ \pi_1(\mathcal{D}_k^X), \ldots, \mathcal{F} \circ \pi_k(\mathcal{D}_k^X))) \qquad \mathcal{X}_k \overset{\text{a.s.}}{=} \pi_k(\mathcal{D}_k^X)$$

が成り立つ．ここで，$\overset{\text{a.s.}}{=}$ は確率 1 で等号が成り立つことを表し，π_i は i 番目の要素を取り出す射影関数 $\pi_i : (x_1, x_2, \ldots, x_i, \ldots, x_k) \mapsto x_i$ である．

これらから，例えば $\{\omega \in \Omega \mid \mathcal{D}_k^X(\omega) = d_k^X \wedge \mathcal{F}(\omega) = f\}$ を略記して $[\mathcal{D}_k^X = d_k^X \wedge \mathcal{F} = f]$ と書くなどすると，

$$\begin{aligned}
[\mathcal{D}_k^X = d_k^X \wedge \mathcal{F} = f] &= [\mathcal{D}_k^X = d_k^X \wedge \mathcal{F} = f \wedge \mathcal{F}(\mathcal{D}_k^X) = f(d_k^X)] \\
&\cong [\mathcal{D}_k^X = d_k^X \wedge \mathcal{F} = f \wedge \mathcal{D}_k^Y = f(d_k^X)] \quad (\mathcal{D}_k^Y \overset{\text{a.s.}}{=} \mathcal{F}(\mathcal{D}_k^X) \text{ より}) \\
&\cong [\mathcal{D}_k = (d_k^X, f(d_k^X)) \wedge \mathcal{F} = f] \quad (\mathcal{D}_k \overset{\text{a.s.}}{=} (\mathcal{D}_k^X, \mathcal{D}_k^Y) \text{ より})
\end{aligned}$$

が成り立つので，$[\mathcal{D}_k^X = d_k^X \wedge \mathcal{F} = f]$ と $[\mathcal{D}_k^X = d_k^X \wedge \mathcal{F} = f \wedge \mathcal{D}_k^Y = f(d_k^X)]$ と $[\mathcal{D}_k = (d_k^X, f(d_k^X)) \wedge \mathcal{F} = f]$ はどれも事象として同じであるといえる．ここで，$\cong$ は対称差の確率が 0 であることを表す．

$$A \cong B \Leftrightarrow P((A \setminus B) \cup (B \setminus A)) = 0$$

以下の補題・定理の証明では，事象として同じであることの議論は省略する．

3.1.2 NFL 定理の背景と意味

最適化問題を解くためのアルゴリズムには，ランダム探索，山登り探索，焼きなまし法，遺伝的アルゴリズムなど様々な種類がある．では，ある最適化問題が与えられたとき，どの探索アルゴリズムを用いて探索を行うのが最も良いのか．

それぞれのアルゴリズムの良し悪しを評価するためには，ある種の効率を定義する必要がある．考えられる効率の与え方としては,「必要となる計算時間，記憶容量に対してどの程度良い解が得られるか」というものがあるが，NFL 定理では探索アルゴリズムの出力結果のみから与えられる効率を考える[1)]．すなわち，k ステップまで探索したとき評価値列 $d_k^Y \in D_k^Y$ が出力されたとすると，D_k^Y から実数全体の集合 $\mathbf{R}$ への関数 $\Phi : D_k^Y \to \mathbf{R}$ を用いた $\Phi(d_k^Y)$ がアルゴリズムの（k ステップまで探索したときの）効率であり，値が大きい方が良いものとする．ここで，探索アルゴリズム a の平均効率は次のように表される．

$$\sum_{d_k^Y \in D_k^Y} \Phi(d_k^Y) P_a(\mathcal{D}_k^Y = d_k^Y) \tag{1}$$

Φ を固定した場合，式 (1) の値がより大きい探索アルゴリズムの方が効率が良いといえる．

さらに，式 (1) は任意の $f \in F$ について $P_a(\mathcal{F} = f) > 0$ であるとき

$$\sum_{d_k^Y \in D_k^Y} \Phi(d_k^Y) P_a(\mathcal{D}_k^Y = d_k^Y) = \sum_{d_k^Y \in D_k^Y} \Phi(d_k^Y) \sum_{f \in F} P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f) P_a(\mathcal{F} = f) \tag{2}$$

と変形できるが，NFL 定理が主張するのは式 (2) 右辺の $\sum_{f \in F} P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f)$ がアルゴリズムによらず一定値をとるということである．$P_a(\mathcal{F} = f)$ が一定，すなわち $P_a(\mathcal{F} = f) = 1/|F|$ のとき NFL 定理より式 (2) 右辺がアルゴリズムによらず一定値をとるので，結局これまでの仮定の上においてはどのような探索アルゴリズムであっても平均的探索効率が同じになるといえる．

3.2 決定的アルゴリズムでの NFL 定理

本小節において，決定的アルゴリズムにおける NFL 定理の証明を行う．なお，本小節で与える定義・補題・定理と証明は，柳井・伊庭[2)] が与えたものに基づいている．2) と本稿とで異なる点を以下に挙げる．

- 確率空間においてアルゴリズムは確率変数としては扱わず，パラメータとして扱うようにした．アルゴリズムをパラメータとして扱っても証明方法はほぼ同じであり，モデルとしても妥当である．（さらに，後に確率的アルゴリズムを用いた場合について考える際にも，確率空間を有限なものとして構成することができる．）
- 決定的アルゴリズムの動作を確率を用いて定式化することによって，補題・定理が成立する前提条件を明確にし，定義と補題の追加，一部証明の変更を行った．
- 条件付確率を用いた議論において，条件部分の確率が 0 である場合を考慮して一部命題の変更を行った．

- 一部表記法を変更した．

3.2.1 厳密な定義

3.1.1 で大まかにしか定義を述べていなかったものについて，厳密な定義を以下に示す．

X，Y を要素が 2 個以上ある有限集合とする．正整数 $m \le |X|$ を探索の最大ステップ数とする．また，k は $k \le m$ を満たす正整数とする．これ以降本稿では特に記さない場合，m を一つ固定し，k はこの条件を満たす任意の正整数とする．

⟦定義 1⟧ (評価関数) F を X から Y への写像全体の集合とする．

$$F = \{f : X \to Y\}$$

また，f の定義を $f : X^k \to Y^k$ に拡張する．

$$f : (x_1, x_2, \ldots, x_k) \mapsto (f(x_1), f(x_2), \ldots, f(x_k))$$

⟦定義 2⟧ (解候補列，評価値列，探索履歴)

$$D_k^X = \{(x_1, x_2, \ldots, x_k) \in X^k \mid 1 \le \forall i \le k, 1 \le \forall j \le k \quad i \ne j \Rightarrow x_i \ne x_j\}$$
$$D_k^Y = Y^k$$
$$D_k = D_k^X \times D_k^Y$$

⟦定義 3⟧ (列の追加) $1 \le k < m$ として $\|_k^X : D_k^X \times X \to X^{k+1}$ を次のように定義する．

$$(x_1, x_2, \ldots, x_k)\|_k^X x = (x_1, x_2, \ldots, x_k, x)$$

同様に，$\|_k^Y : D_k^Y \times Y \to D_{k+1}^Y$ を次のように定義する．

$$(y_1, y_2, \ldots, y_k)\|_k^Y y = (y_1, y_2, \ldots, y_k, y)$$

また，$\|_k : D_k \times (X \times Y) \to X^{k+1} \times D_{k+1}^Y$ を次のように定義する．

$$(d_k^X, d_k^Y)\|_k(x, y) = (d_k^X \|_k^X x, d_k^Y \|_k^Y y)$$

これらは列に新しい要素を追加する関数である．以後，添え字を省き単に $\|$ と書く．

⟦定義 4⟧ (決定的アルゴリズム)

$$A_k = \{a_k : D_k \to X \mid \forall d_k \in D_k, 1 \le \forall i \le k \quad a_k(d_k) \ne \pi_i \circ \pi_1(d_k)\} \quad (1 \le k < m)$$
$$A = X \times A_1 \times A_2 \times \ldots A_{m-1}$$

ここで a_k についての条件は既に探索した X の元を再び探索しないということである．$a = (a_0, a_1, a_2, \ldots, a_{m-1}) \in A$ を用いると，$f \in F, x \in X, 1 \le i < m, d_i \in D_i$ として，

$$\begin{cases} P_a(\mathcal{F} = f) > 0 \Rightarrow P_a(\mathcal{X}_1 = x \mid \mathcal{F} = f) = \delta(x, a_0) \\ P_a(\mathcal{D}_i = d_i, \mathcal{F} = f) > 0 \Rightarrow P_a(\mathcal{X}_{i+1} = x \mid \mathcal{D}_i = d_i, \mathcal{F} = f) = \delta(x, a_i(d_i)) \end{cases}$$

が成り立つ[*1]．最初の式は必ず a_0 から探索を開始することを表し，2 番目の式は探索履歴が d_i のときに次に必ず $a_i(d_i)$ を探索することを表している．2) ではアルゴリズムの動作を，P_a がこのような式を満たすものとして定式化して与えてはいない．3.2 の補題・定理は，3.1.1 の確率変数についての仮定と，この定式化したアルゴリズムの動作から証明される．

⟨補題 1⟩ $1 \le k < m$ のとき，$a_k \in A_k, d_k = (d_k^X, d_k^Y) \in D_k$ とすると次が成り立つ．

$$d_k^X \| a_k(d_k) \in D_{k+1}^X$$

本補題は 2) で証明された補題と同じであり，証明は 2) を参照してほしい．

この補題は，アルゴリズムによって選ばれた解候補をこれまでに探索した解候補列に新しく付け加えても，依然として解候補列の体裁を保つことを保証する．

⟦定義 5⟧ (探索履歴を満たす評価関数)

$d_k \in D_k$ に対し以下のように定義する．

$$F(d_k) = \{f \in F \mid 1 \le \forall i \le k \quad f(\pi_i \circ \pi_1(d_k)) = \pi_i \circ \pi_2(d_k)\}$$

⟦定義 6⟧ (アルゴリズムと評価関数から決まる探索履歴)

$a = (a_0, a_1, \ldots, a_{m-1}) \in A$，$f \in F$ に対して，$S_k : A \times F \to D_k$ を

$$S_1(a, f) = (a_0, f(a_0))$$

$1 \le i < m$ のとき

$$x_{i+1} = a_i(S_i(a, f))$$
$$S_{i+1}(a, f) = S_i(a, f)\|(x_{i+1}, f(x_{i+1}))$$

と再帰的に定義する．また，$S_k^X : A \times F \to D_k^X$ と $S_k^Y : A \times F \to D_k^Y$ を

$$S_k^X(a, f) = \pi_1 \circ S_k(a, f)$$
$$S_k^Y(a, f) = \pi_2 \circ S_k(a, f)$$

と定義する．

補題 1 より $x_{i+1} = a_i(S_i(a, f))$ を解候補列 d_i^X に付け加えたものは必ず D_{i+1}^X の要素となる．即ち S_{i+1} の値は必ず D_{i+1}^X の要素となるので，この定義は well-defined である．

⟨補題 2⟩ 任意の $a \in A$，$f \in F$ について，$P_a(\mathcal{F} = f) > 0$ のとき次が成り立つ．

$$P_a(\mathcal{D}_k = S_k(a, f) \mid \mathcal{F} = f) = 1 \tag{3}$$

この補題は 2) の中で提示されていないものであり，a に f を与えて実行したときの探索履歴が確率 1 で $S_k(a, f)$ になることを保証している．証明は定義 4 の後の P_a の満たすべき性質と 3.1.1 の確率変数についての仮定を用いて k についての数学的帰納法による．証明の詳しい道筋は紙面の都合により省略する．参照したい場合は例えば 3) にある．

⟦定義 7⟧ (アルゴリズムと評価値列から決まる探索履歴)

$a = (a_0, a_1, \ldots, a_{m-1}) \in A$，$t = (y_1, y_2, \ldots, y_k) \in D_k^Y$ に対して，

---

*1 δ はクロネッカーのデルタで，$\delta(x, x) = 1$，$x \ne y$ のとき $\delta(x, y) = 0$ である．

Vol.2010-AL-129 No.3
2010/3/5

$$c_1 = (a_0, y_1)$$
$$c_{i+1} = c_i \| (a_i(c_i), y_{i+1}) \qquad (1 \leq i < m)$$
と再帰的に定義し，$R_k : A \times D_k^Y \to D_k$ を次のように定義する．
$$R_k(a,t) = c_k$$
また，$R_k^X : A \times D_k^Y \to D_k^X$ と $R_k^Y : A \times D_k^Y \to D_k^Y$ を次のように定義する．
$$R_k^X(a,t) = \pi_1 \circ R_k(a,t)$$
$$R_k^Y(a,t) = \pi_2 \circ R_k(a,t) \ (= t)$$
補題 1 より $a_i(c_i)$ を解候補列 d_i^X に付け加えたものは必ず D_{i+1}^X の要素となる．即ち R_{i+1} の値は必ず D_{i+1}^X の要素となるので，この定義は well-defined である．

R_k^Y は $t \in D_k^Y$ に対して恒等関数となっていることに注意する．

3.2.2 補　　題

NFL 定理の証明に用いるいくつかの補題を証明する．

〈補題 3〉 (S_k と R_k の整合性) 任意の $a \in A$，$t \in D_k^Y$，$f \in F(R_k(a,t))$ について次が成り立つ．
$$S_k(a,f) = R_k(a,t)$$
本補題は 2) で証明された補題と同じであり，証明は 2) を参照してほしい．

〈補題 4〉 任意の $a \in A$，$t \in D_k^Y$，$f \in F(R_k(a,t))$ について，$P_a(\mathcal{F} = f) > 0$ のとき次が成り立つ．
$$\begin{cases} P_a(\mathcal{D}_k^Y = t \mid \mathcal{F} = f) = 1 \\ P_a(\mathcal{D}_k^Y \neq t \mid \mathcal{F} = f) = 0 \end{cases}$$
[証明] 補題 3 より，
$$S_k^Y(a,f) = R_k^Y(a,t) = t$$
であり，
$$\begin{aligned} P_a(\mathcal{D}_k^Y = t \mid \mathcal{F} = f) =& P_a(\mathcal{D}_k^Y = S_k^Y(a,f) \mid \mathcal{F} = f) \\ \geq& P_a(\mathcal{D}_k^X = S_k^X(a,f), \mathcal{D}_k^Y = S_k^Y(a,f) \mid \mathcal{F} = f) \\ =& P_a(\mathcal{D}_k = S_k(a,f) \mid \mathcal{F} = f) \\ =& 1 \quad (\text{補題 2 より}) \end{aligned}$$
となる．したがって $P_a(\mathcal{D}_k^Y = t \mid \mathcal{F} = f) = 1$ が成り立つ．
また，$P_a(\mathcal{D}_k^Y \neq t \mid \mathcal{F} = f) = 1 - P_a(\mathcal{D}_k^Y = t \mid \mathcal{F} = f) = 0$ が成り立つ． ■

補題 4 は 2) でも示されているが，そこでは本稿の定義 4 の後にあるようなアルゴリズムの動作を定式化した式が明示されておらず，証明にも P_a が現れていない．確率変数や確率測度が満たす仮定から命題を導くのが確率論の定理や証明の典型的な形式の一つである[4]．本稿ではそういった確率論的な形式で命題を証明することを目指している．それゆえ定式化したアルゴリズムの動作，即ち各確率変数と P_a についての条件からこの補題や NFL 定理を証明すべきと考え，補題 2 を介してそのような証明を与えた．

〈補題 5〉 (F の分割) V を実数体上の任意のベクトル空間とする．任意の $\varphi : F \to V$，$a \in A$ について次が成り立つ．
$$\sum_{f \in F} \varphi(f) = \sum_{t \in D_k^Y} \sum_{f \in F(R_k(a,t))} \varphi(f)$$
証明は以下のように行う．右辺では評価関数の空間を分割し，分割された各部分集合ごとに和をとっている．そして各部分集合は $t \in D_k^Y$ によってパラメータ化されている．ここで示すべきなのは，右辺の和において重複なくすべての $f \in F$ が出てきている，ということである．すなわち，
- 各部分集合が互いに交わりを持たないこと．
- 各部分集合の要素の数の総和が $|F|$ であること．

を示せばよい．証明の残りの部分については 2) を参照してほしい．

3.2.3 NFL 定理の証明 (決定的アルゴリズムの場合)

決定的アルゴリズムを用いたときの NFL 定理を証明する．

【定理 1】 (NFL 定理) 任意の $a \in A$，$d_k^Y \in D_k^Y$ について，$\forall f \in F \quad P_a(F = f) > 0$ とすると次が成り立つ．
$$\sum_{f \in F} P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f) = |Y|^{|X|-k}$$
[証明] 補題 5 により a によって F を分割すると，
$$\sum_{f \in F} P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f) = \sum_{t \in D_k^Y} \sum_{f \in F(R_k(a,t))} P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f)$$
となる．ここで，補題 4 より，$f \in F(R_k(a,t))$ ならば
$$P_a(\mathcal{D}_k^Y = t \mid \mathcal{F} = f) = 1$$
$$P_a(\mathcal{D}_k^Y \neq t \mid \mathcal{F} = f) = 0$$
となる．よって，
$$\sum_{f \in F(R_k(a,t))} P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f) = \begin{cases} 0 & (t \neq d_k^Y) \\ |F(R_k(a,t))| = |Y|^{|X|-k} & (t = d_k^Y) \end{cases}$$

となるので,

$$\sum_{f\in F} P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f) = |Y|^{|X|-k}$$

と計算される.

以上により,アルゴリズム a によらず $\sum_{f\in F} P_a(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f)$ は一定値 $|Y|^{|X|-k}$ をとることがわかった.よって NFL 定理は成立する. ■

以上のような NFL 定理の証明は Wolpert らの与えた証明[1] よりも厳密なものである[2].さらにこの証明を考察することで,それぞれの決定的アルゴリズムの差異は関数空間の分割の仕方の違いとして現れると解釈できる[2].

3.3 確率的アルゴリズムでの NFL 定理

確率的アルゴリズムを用いた場合にも NFL 定理が成り立つことは Wolpert らの論文[1] において主張されているが,具体的な証明は与えられていない.柳井・伊庭らの論文[2] においては決定的アルゴリズムのみに限定して考えることの妥当性を一定の根拠つきで主張してはいるが,確率的アルゴリズムでの NFL 定理に関しては決定的アルゴリズムと同等の厳密さで定式化してはおらず,証明も与えられていない.Droste らの論文[5] において,確率的アルゴリズムでの NFL 定理の証明が与えられてはいるが,本稿で与えるように厳密な確率的アルゴリズムの定義の上で NFL 定理を証明したものではない.

ここでは,前小節において与えた定義に加え,いくつかの定義と補題を示した上で確率的アルゴリズムを用いた場合においても NFL 定理が成り立つことを証明する.

3.3.1 定義と補題

[[定義 8]] (確率的アルゴリズム) 確率的アルゴリズム全体の集合 Σ を次のように定義する.ここで $\boldsymbol{P}$ は X 上の確率測度全体の集合とする.

$$\Sigma_k = \{\sigma : D_k \to \boldsymbol{P} \mid \forall d_k \in D_k, 1 \le \forall i \le k \quad \sigma(d_k)(\pi_i \circ \pi_1(d_k)) = 0\} \quad (1 \le k < m)$$

$$\Sigma = \boldsymbol{P} \times \Sigma_1 \times \Sigma_2 \times \cdots \times \Sigma_{m-1}$$

ここで σ についての条件は,既に探索した X の元を再探索しないということである.また,この定義はほぼ 1) によるが,P_σ が満たすべき次の性質は 1),2) で与えられていない.

$\sigma = (\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_{m-1}) \in \Sigma$ を用いたとき,$f \in F, x \in X, 1 \le i < m, d_i \in D_i$ として

$$\begin{cases} P_\sigma(\mathcal{F} = f) > 0 \Rightarrow \sigma_0(\mathcal{X}_1 = x) = P_\sigma(\mathcal{X}_1 = x \mid \mathcal{F} = f) \\ P_\sigma(\mathcal{D}_i = d_i, \mathcal{F} = f) > 0 \Rightarrow \sigma_i(d_i)(\mathcal{X}_{i+1} = x) = P_\sigma(\mathcal{X}_{i+1} = x \mid \mathcal{D}_i = d_i, \mathcal{F} = f) \end{cases}$$

が成り立つ.

最初の式は最初に探索する X の元を選ぶ確率が,f に関わらず σ_0 で与えられることを表し,2 番目の式は探索履歴が d_i のときに次に探索する X の元を選ぶ確率がやはり f に関わらず確率分布 $\sigma_i(d_i)$ で与えられることを表す.小節 3.3 の補題や定理は,これらの性質と確率変数についての仮定から導かれる.

⟨補題 4′⟩ 任意の $\sigma \in \Sigma$,$d_k^X \in D_k^X$,$t \in D_k^Y$,$f \in F((d_k^X, t))$ について,$P_\sigma(\mathcal{F} = f) > 0$ のとき次が成り立つ.

$$\begin{cases} P_\sigma(\mathcal{D}_k^Y = t, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f) = P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f) \\ P_\sigma(\mathcal{D}_k^Y \ne t, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f) = 0 \end{cases}$$

[証明]

$$P_\sigma(\mathcal{D}_k^Y = t, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f) = \frac{P_\sigma(\mathcal{D}_k^Y = t, \mathcal{D}_k^X = d_k^X, \mathcal{F} = f)}{P_\sigma(\mathcal{F} = f)}$$

ここで,$f \in F((d_k^X, t))$ より $[\mathcal{D}_k^Y = t \wedge \mathcal{D}_k^X = d_k^X \wedge \mathcal{F} = f)]$ と $[\mathcal{D}_k^X = d_k^X \wedge \mathcal{F} = f)]$ は事象として同じものを表すので次のように変形される.

$$= \frac{P_\sigma(\mathcal{D}_k^X = d_k^X, \mathcal{F} = f)}{P_\sigma(\mathcal{F} = f)}$$

$$= P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$$

また,次が成り立つ.

$$P_\sigma(\mathcal{D}_k^Y \ne t, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$$

$$= P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f) - P_\sigma(\mathcal{D}_k^Y = t, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f) = 0$$ ■

この補題は d_k^X を t に写す f であれば,$\mathcal{F} = f \wedge \mathcal{D}_k^X = d_k^X$ に $\mathcal{D}_k^Y = t$ を条件として付け加えても確率が変わらないことを表している.この補題からただちに次の補題が導かれる.

⟨補題 4″⟩ 任意の $\sigma \in \Sigma$,$d_k^X \in D_k^X$,$t \in D_k^Y$,$f \in F((d_k^X, t))$ について,$P_\sigma(\mathcal{D}_k^X = d_k^X, \mathcal{F} = f) > 0$ のとき次が成り立つ.

$$\begin{cases} P_\sigma(\mathcal{D}_k^Y = t \mid \mathcal{D}_k^X = d_k^X, \mathcal{F} = f) = 1 \\ P_\sigma(\mathcal{D}_k^Y \ne t \mid \mathcal{D}_k^X = d_k^X, \mathcal{F} = f) = 0 \end{cases}$$

この補題は $\mathcal{F}$ が d_k^X を t に写す f で $\mathcal{D}_k^X$ が d_k^X の場合には確率 1 で $D_k^Y = t$ であることを表す.証明は容易であり省略する.

⟨補題 5′⟩ (F の分割) V を実数体上の任意のベクトル空間とする.任意の $\varphi : F \to V$,$d_k^X \in D_k^X$ について次が成り立つ.

$$\sum_{f\in F} \varphi(f) = \sum_{t\in D_k^Y} \sum_{f\in F((d_k^X, t))} \varphi(f)$$

補題 5′ は d_k^X に対し評価値列 $t \in D_k^Y$ を与える同値類に F を分割して $\varphi(f)$ の和を求めて

Vol.2010-AL-129 No.3
2010/3/5

よいということを意味する[*1]．補題 5 と類似の流れで示される．ここでは証明は省略する．

⟨補題 6⟩ 任意の $\sigma \in \Sigma$ に対し次が成り立つ．

$$\forall d_k^X \in D_k^X\ \forall d_k^Y \in D_k^Y\ \forall f_1, f_2 \in F((d_k^X, d_k^Y))$$
$$P_\sigma(\mathcal{F} = f_1) > 0 \wedge P_\sigma(\mathcal{F} = f_2) > 0 \tag{4}$$
$$\Rightarrow P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f_1) = P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f_2)$$

この補題により，$d_k^Y = f(d_k^X) \wedge P_\sigma(\mathcal{F} = f) > 0$ であるようなすべての $f \in F$ に対し $P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$ は同じ値をとることがわかる．証明は，確率変数についての仮定と P_σ の満たすべき性質を用いて k についての数学的帰納法による．詳細は省略する．

⟨補題 7⟩ 任意の $\sigma \in \Sigma$ について次が成り立つ．

$$\forall d_k^Y \in D_k^Y\ \forall \{f_{d_k^X}\}_{d_k^X \in D_k^X} \subseteq F$$
$$(\forall d_k^X \in D_k^X\ f_{d_k^X} \in F((d_k^X, d_k^Y)) \wedge P_\sigma(\mathcal{F} = f_{d_k^X}) > 0) \tag{5}$$
$$\Rightarrow \sum_{d_k^X \in D_k^X} P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f_{d_k^X}) = 1$$

ここで $\{f_{d_k^X}\}_{d_k^X \in D_k^X}$ は関数族であり，通常通り $f : D_k^X \to F$ であって $f(d_k^X)$ を $f_{d_k^X}$ と記している．証明は，確率変数についての仮定，P_σ の満たすべき性質，補題 6 を用いて k についての数学的帰納法による．詳細は省略する．

3.3.2 NFL 定理の証明 (確率的アルゴリズムの場合)

確率的アルゴリズムを用いたときの NFL 定理を証明する．

【定理 2】 (NFL 定理) 任意の $\sigma \in \Sigma$，$d_k^Y \in D_k^Y$ について，$\forall f \in F \quad P_\sigma(F = f) > 0$ とすると次が成り立つ．

$$\sum_{f \in F} P_\sigma(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f) = |Y|^{|X|-k}$$

[証明] k, σ, d_k^Y を固定する．$\sum_{d_k^X \in D_k^X} P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f) = 1$ より次が成り立つ．

$$\text{左辺} = \sum_{f \in F} \sum_{d_k^X \in D_k^X} P_\sigma(\mathcal{D}_k^Y = d_k^Y, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$$

和の順序を交換する．

$$= \sum_{d_k^X \in D_k^X} \sum_{f \in F} P_\sigma(\mathcal{D}_k^Y = d_k^Y, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$$

補題 5′ を用いて F を分割する．

$$= \sum_{d_k^X \in D_k^X} \sum_{t \in D_k^Y} \sum_{f \in F((d_k^X, t))} P_\sigma(\mathcal{D}_k^Y = d_k^Y, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$$

t についての和を計算する．補題 4′ より，$t = d_k^Y$ の場合だけ $P_\sigma(\mathcal{D}_k^Y = d_k^Y, \mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f) = P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$ となり，それ以外 $(t \neq d_k^Y)$ の場合では確率 0 となる．結局 $t = d_k^Y$ の場合だけが残る．

$$= \sum_{d_k^X \in D_k^X} \sum_{f \in F((d_k^X, d_k^Y))} P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$$

次に f についての和を計算する．補題 6 よりすべての $f \in F((d_k^X, d_k^Y))$ について $P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f)$ は同じ値をとる．$d_k^X \in D_k^X$ ごとに $F((d_k^X, d_k^Y))$ の要素を一つずつ選び $\{f_{d_k^X}\}_{d_k^X \in D_k^X}$ を作ると式は次のように変形される．

$$= \sum_{d_k^X \in D_k^X} |F((d_k^X, d_k^Y))| P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f_{d_k^X})$$
$$= \sum_{d_k^X \in D_k^X} |Y|^{|X|-k} P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f_{d_k^X})$$
$$= |Y|^{|X|-k} \sum_{d_k^X \in D_k^X} P_\sigma(\mathcal{D}_k^X = d_k^X \mid \mathcal{F} = f_{d_k^X}) = |Y|^{|X|-k}$$

最後の式変形は補題 7 による．以上により，アルゴリズム σ によらず $\sum_{f \in F} P_\sigma(\mathcal{D}_k^Y = d_k^Y \mid \mathcal{F} = f)$ は一定値 $|Y|^{|X|-k}$ をとることがわかった．よって NFL 定理は成立する． ■

4. 考察

決定的アルゴリズムと確率的アルゴリズムそれぞれの場合において NFL 定理の証明を示したが，どちらの場合も左辺を計算した結果，アルゴリズムによらず一定値 $|Y|^{|X|-k}$ をとるというものであった．つまり，探索アルゴリズムの平均効率は，そのアルゴリズムが決定的であるか確率的であるかにも関わらず同じになるということを厳密に示したことになる．

これは別の推論によっても導き出せる．すべての i に対し，ある特定の $x \in X$ についての $\sigma_i(d_i)(\{x\})$ の値が 1 であるような確率的アルゴリズムが決定的アルゴリズムであることを考えると，確率的アルゴリズムは決定的アルゴリズムを特別な場合として含んでいる．そして本稿では確率的アルゴリズムの場合の NFL 定理の証明として決定的アルゴリズムの場合の NFL 定理の証明から独立なものを与えているので，実は前者の証明のみで十分である．すると前者の場合に左辺が一定値であれば後者もそうであって同じ値であることがわかる．

本稿での確率的アルゴリズムは決定的アルゴリズムを各探索ステップで確率分布を与えるように拡張したものであるといえる．本稿での確率的アルゴリズムの取り扱い方法は，探索過程におけるアルゴリズムの確率的振る舞いを基本的な式により表現したものであり，実際

[*1] 補題 5′ を補題 5 の対応物と考えることもできるが，アルゴリズムの動作による F の分割にはなっていない．

の確率的アルゴリズムはすべて Σ の要素としてモデル化されると考えられる．他の確率的アルゴリズムの定義方法として,「決定的アルゴリズムの分布があり，どの決定的アルゴリズムを用いるかを探索開始時に確率的に選択する」，つまり A 上の確率測度として定義するという方法も考えられる[5)]．このように定義した確率的アルゴリズムが実際の確率的アルゴリズムをもれなくモデル化しているのかどうかという問題や，本稿で定義した確率的アルゴリズムとの関係を理解するためには更なる議論を要する．

5. 関 連 研 究

$\tau : X \to X$ を X の置換とし，評価関数 f の置換 τf を $\tau f(x) = f(\tau^{-1}(x))$ で定義する．Schumacher らは 6) において，NFL 定理が成立する必要十分条件は，評価関数の集合が X の任意の置換について閉じていることであることを示した．これにより，NFL 定理が成立する評価関数の集合を，置換について閉じている F の部分集合に狭めることができる．

Auger らは 7) において，解候補全体の集合が可算無限の場合と非可算無限の場合の NFL 定理について言及している．7) では条件を弱めた NFL 定理を考えており，解候補全体の集合が可算無限の場合には弱い NFL 定理が成り立ち，非可算無限の場合では成り立たないとしている．

6. ま と め

本稿では，決定的プログラムと確率的プログラムのそれぞれの場合について，アルゴリズムが実行されるときに確率変数と確率測度が満たすと考えられる論理式を明示的に与え，それらに基づいて NFL 定理を厳密に証明した．

NFL 定理とその帰結は，様々な仮定に基づいて示されることに注意しなければならない．本稿で与えた仮定をまとめると以下のようになる．

- 解候補全体の集合，解の評価値全体の集合は共に有限である．
- 探索アルゴリズムはこれまでの探索履歴しか参照しない[⋆1]．
- 探索アルゴリズムは一度探索した解候補を再び調べることはない．
- 探索アルゴリズムの効率とは，計算時間や記憶容量に関するものではなく，評価値列のみから与えられる特殊な意味での効率である．

謝辞　確率論について親切に教えていただきました京都大学大学院人間・環境学研究科の上木直昌先生に感謝致します．また，確率論について教えていただき，さらに研究について有益なディスカッションをさせていただいている京都の計算可能解析学の研究会の方々に感謝します．

⋆1 この仮定によって，対象とする探索アルゴリズムが制限されることはないと考えられる．

参 考 文 献

1) D. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, Vol.1, No.1, pp. 67–82, 1997.
2) 柳井孝介, 伊庭斉志. No free lunch theorem の別証明と解釈. 情報処理学会アルゴリズム研究会報告, No.26, pp. 1–8, 2005.
3) 橋谷祐司. No free lunch 定理の証明と考察, 2010. 京都大学総合人間学部卒業論文.
4) 伊藤清. 確率論の基礎［新版］. 岩波書店, 2004.
5) S.Droste, T.Jansen, and I.Wegener. Optimization with randomized search heuristics – the (A) NFL theorem, realistic scenarios, and difficult functions. *Theoret. Comput. Sci.*, Vol. 287, No.1, pp. 131–144, 2002.
6) C.Schumacher, M.Vose, and L.Whitley. The no free lunch and problem description length. In *Proc. of the 2001 Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 565–570. Morgan Kaufmann, 2001.
7) A. Auger and O. Teytaud. Continuous lunches are free! In *Proc. of the 9th annual conference on Genetic and evolutionary computation (GECCO-2007)*, p. 922. ACM, 2007.