社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－AL－100（1）
2005／3／17

No Free Lunch Theorem の別証明と解釈

柳井 孝介　　伊庭 斉志

東京大学大学院新領域創成科学研究科基盤情報学専攻

本稿では No Free Lunch Treorem （NFL）の別証明を与える．NFL は「どんな問題に対しても平均的に効率良く解けるような探索アルゴリズムは存在しない」ということを主張する定理であり，探索アルゴリズムあるいは最適化法の研究に大きな影響を与えた．本稿では，より簡潔でかつ直観的な証明を与える．我々は評価関数の空間を部分集合に分割し，それぞれの部分集合ごとにパフォーマンスが得られる確率を合計する．関数空間の分割により，定理のより深い理解が可能となり，またアルゴリズムと問題の関係が明確となる．

Simple Proof and Interpretation of No Free Lunch Theorem

Kohsuke Yanai　　Hitoshi Iba

Dept. of Frontier Infomatics, Graduate School of Frontier Sciences,
The University of Tokyo.

In this paper, a simple proof of No Free Lunch Theorem (NFL) is presented. NFL asserts that no search algorithm exists which can solve every problem efficiently. This theorem has serious implications for the study of search algorithms and optimization. The proof given in this paper is simpler and more intuitive than the original one. In the proof, we divide the fitness function space into a set of subsets, and calculate the sum of probabilities in each subset. The division of the function space helps us to understand the theorem more clearly and highlights the relationship between algorithms and problems.

1. はじめに

本稿ではメタヒューリスティクスの探索性能について議論する．代表的なメタヒューリスティクスとして，遺伝的アルゴリズム，分布推定アルゴリズム [佐久間 03] [倉橋 03]，タブーサーチ，ランダム探索，シミュレーティッドアニーリング法，山登り法，Evolutionary Strategy，Memetic Algorithm などが挙げられる．これらの手法は以下の 2 つの共通の性質を持つ．

（1）汎用的であり，問題を限定しない．
（2）問題を限定しないゆえ，（一見）問題に対する事前知識を用いない．

本稿ではこれら 2 つの性質を持つメタヒューリスティックを単に探索アルゴリズムと呼ぶことにする．

問題を限定せずに用いることができるのであれば，アルゴリズムの平均的な探索性能を議論することができる．すなわち，あらゆる問題を想定し，すべての問題に対する探索性能を平均する．この探索性能の平均値により，探索アルゴリズムの性能を評価することが可能となる．そこで，「ある 2 つのアルゴリズムの間の探索性能の平均値の差はどれほどか」や「平均的に最もよい探索性能を持つアルゴリズムはどれか」が我々の興味の対象となる．

No Free Lunch Theorem（以後 NFL と略す）はこの問いにおおまかな解答を与えた．NFL とは「平均すればどの探索アルゴリズムの性能も同じである」と主張する定理であり，Wolpert と Macready によって 1995 年に証明された [Wolpert 95]．NFL よりただちに，「どんな問題に対しても平均的に効率良く解けるような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．NFL が成立する枠組みは限定されているが，探索アルゴリズムの性能に関する基本原理と考えられ [吉澤 01]，探索アルゴリズムあるいは最適化法の汎用性という概念に大きな影響を与えた．特に進化計算のコミュニティでこの定理は重要視されており，進化計算に関する国際会議 GECCO（Genetic and Evolutionary Computation Conference）では，2001 年，2002 年，2004 年の開催時に NFL のチュートリアルを開いて議論がなされている．また複雑系科学においても，共進化の考え方に重大な影響を与えた [Kauffman 00]．

本稿では NFL の別証明を与える．本稿で用いた NFL の前提は Wolpert らのものと全く同じである

が，我々の証明ははるかに簡潔でなおかつ厳密である．また証明の過程を図にイメージ化することができ，これは定理に対する深い理解と洞察をもたらす．Wolpert らによる証明は，確率論の公式を用いた式変形によって構成されている．確率論による解析は探索アルゴリズムを扱うフレームワークを提供するとの指摘もあるが [Wolpert 97]，Wolpert らの証明は式変形に頼っているため，証明自体が複雑で NFL が成立する理由を直観的に理解しづらい．

我々の証明のキーとなるアイディアは評価関数の空間の分割である．これは探索アルゴリズムの性能問題に対する新しい視点をもたらす．本論文で示す証明に基づいて，どのように探索手法と問題の関係を考えねばならないかの一つの指針が示される．関数空間の分割という考え方は，ただ証明を簡単にするだけではなく，探索アルゴリズムの関数空間に対する役割を明らかにし，アルゴリズムの性能を議論する上での方法論を提供する．

以下 2 章で [Wolpert 95] に従って記号の定義と NFL を説明し，3 章で NFL の証明を与える．4 章で本論文で示す証明法に基づいて考察を行い，5 章で本論文をまとめる．また付録では形式的な定義と補題の証明を与える．

2. No Free Lunch Theorem

2·1 前　　提

本節では NFL が前提とするフレームワークについて説明する．すなわち，対象となる探索アルゴリズムを厳密に形式化する．ここで述べる前提は [Wolpert 95] のものと全く同一である．これらの前提が実際の探索アルゴリズムに合致しているか否かについては 4·2 節で考察する．

まず 1 章で述べたように，汎用的で問題を限定せず，問題に対する事前知識を用いないアルゴリズムが考察の対象となる．またコンピュータでアルゴリズムを実装することを想定して，扱う集合は数も含みすべて有限とする．

これらの性質を満たす探索アルゴリズムの一般的な枠組みを形式化するために，具体的に，遺伝的アルゴリズム（Genetic Algorithm：以下 GA）と巡回セールスマン問題（Travelling Salesman Problem：以下 TSP）を例に挙げて考える．TSP の解は 2 進数の配列にコーディングされているとする．GA はまず，ランダムに 2 進数の配列を生成する．次に，ランダムに生成された 2 進数配列をデコードして，TSP の解候補を評価関数により評価し，それぞれの解候補に評価値を割り当てる．評価値は，解候補がどれぐらい良いかを示す指標である．GA はそれぞれの解候補の評価値を基に，遺伝的オペレータを用いて新しい 2 進数配列を合成する．GA は解候補の評価と新しい解候補の生成を繰り返して探索を行っていく．

上の GA と TSP の例を一般化した枠組みが，NFL の対象となる．図 1 に，NFL の適用対象となる探索アルゴリズムの動作を図示する．まず，アルゴリズムにより最初の解 x_1 が与えられる．多くの探索アルゴリズムでは，x_1 はランダムに与えられる．次に，解 x_1 が評価関数によって評価されて，評価値 y_1 が与えられる．探索アルゴリズムは，これまでの解候補の履歴と評価値の履歴から，次の解候補 x_{i+1} を提示する．多くの汎用的な探索アルゴリズムは，以上の形式で探索を行う．例えば，1 章でヒューリスティクスの例として紹介した GA，分布推定アルゴリズム，タブーサーチ，ランダム探索，シミュレーティッドアニーリング法，山登り法，Evolutionary Strategy，Memetic Algorithm などが上で述べた枠組みに収まることは容易に分かる．

以上を形式化して，記号を導入する．解候補の空間を X，解の評価値の空間を Y とし，X，Y は有限と仮定する．例えば，TSP の解空間は有限である．解空間が有限であれば，解の優劣を決めるには，評価値の空間も有限で十分である．Y の元は数あるいはベクトルと考えれば理解しやすいが，以下では Y の元は数である必要はなく，任意の有限集合としても議論は成り立つ．$F = \{f : X \to Y\}$ とする．$f \in F$ を評価関数とし，F を評価関数全体の空間と見なす．$f \in F$ は，解 $x \in X$ に対し，その解がどれだけよいかの評価値 $y \in Y$ を与える関数である．ここで探索アルゴリズム a をこれまでの探索履歴，すなわちこれまで挙げた解の候補 $(x_1, x_2, \cdots, x_i) \in X^i$ とその評価値の履歴 $(y_1, y_2, \cdots, y_i) \in Y^i$ から，次の解候補 $x_{i+1} \in X$ を一つ出力する関数と定義する．

X から Y への写像のとり方は，高々$|Y|^{|X|}$ であるので，$|F| = |Y|^{|X|}$ である．ゆえに F は有限となる．NFL は，有限集合 X から Y へのあらゆる写像すべてを考慮する（表 1 参照）．探索問題は評価関数を一意に定める．同じ評価関数を持つ探索問題を同一の問題と見なせば，探索問題と評価関数を同一視することができる．ゆえに NFL は，あらゆる問題すべてを考慮した上で，アルゴリズムの性能について言及する．

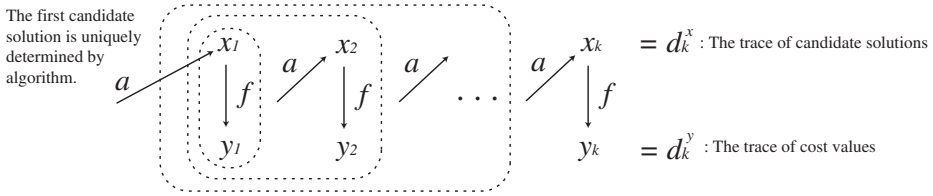

図 1 探索アルゴリズム．最初の解候補はアルゴリズム a により一意的に決められる．その後はこれまで解候補の履歴と評価値の履歴から新しい解候補 x_{i+1} がアルゴリズム a により出力される．これにより a と f から一意的に $d_k \in D_k$ が決まるのでこれを $S(a,f)$ と書く．

表 1 写像の例．NFL では有限集合 X から Y へのあらゆる写像すべて $f_1, f_2 \cdots$ を考慮する．解は 2 進数 3bit の配列にコーディングされており，評価値は 1 から 2^3 までの整数値とする．

解	f_1	f_2	$\cdots$
000	1	2	$\cdots$
001	2	1	$\cdots$
$\vdots$			
111	8	7	$\cdots$

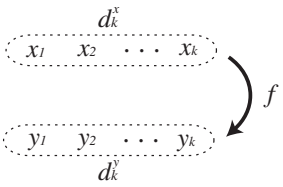

図 2 $F(d_k)$．履歴 d_k をすべて満たすような f の集合．すなわち $f \in F(d_k)$ なら $\forall i\ [y_i = f(x_i)]$.

2·2 定　　義

厳密な定義は付録 A にある．k を $k < |X|$ である自然数とする．

X　解候補の空間．有限集合．

Y　解の評価値の空間．有限集合．

F　$F = \{f : X \to Y\}$．評価関数の集合と見なす．

D_k^X　探索を k ステップ進めた時の解候補全体の集合．ただし $D_k^X \subset X^k$ であり重複する解候補を含まない．

D_k^Y　探索を k ステップ進めた時の解の評価値全体の集合．$D_k^Y = Y^k$

D_k　探索を k ステップ進めた時の探索履歴全体の集合．$D_k = D_k^X \times D_k^Y$

A　探索アルゴリズム全体の集合．

$F(d_k)$　探索履歴 $d_k \in D_k$ を完全に満たすような F の部分集合．

ここで D_k^X は重複する解候補を含まないことに注意する．すなわち $(x_1, x_2, \cdots, x_k) \in D_k^X$ とすると

$$\forall i\ \forall j\ [i \neq j \to x_i \neq x_j] \tag{1}$$

となる．よって探索アルゴリズム $a \in A$ は解候補の履歴を参照して，これまで解候補として挙げられていない X の元を一つ出力する．また $F(d_k)$ は履歴 d_k をすべて満たすような f の表す（図 2 参照）．すなわち $f \in F(d_k)$ なら $\forall i\ [y_i = f(x_i)]$ である．

要素の数については以下のようになる．

$$|F| = |Y|^{|X|} \tag{2}$$

$$|D_k^Y| = |Y|^k \tag{3}$$

$$|F(d_k)| = |Y|^{|X|-k} \tag{4}$$

特に $F(d_k)$ の要素の数は k のみに依存し，d_k には依らないことに注意されたい．d_k が定められているため，X の中の k 個の要素は対応付けが決まっている．残りの $|X| - k$ 個の要素の対応付けを考えればよいので全部で $|Y|^{|X|-k}$ 個の $f \in F$ が d_k を満たすことが分かる．

またアルゴリズム a と評価関数 f を用いて探索を k ステップ進めて得られた探索履歴を $S_k(a,f) \in D_k$ で表す（図 1 参照）．逆にアルゴリズム a と解の評価値の履歴 $d_k^y \in D_k^Y$ が得られたとき，図 3 に示すようにして a と d_k^y に矛盾しないような $d_k^x \in D_k^X$ を一つ定めることができる．まず，探索アルゴリズム a により，最初の解 x_1 が与えられる．y_1 は与えられているので，x_1 と y_1 からアルゴリズム a により x_2 が与えられる．以下同様にして，順に x_i だけ決めていく．このようにして，a と d_k^y から一意的に D_k の元を一つ決めることができるのでこれを $R_k(a, d_k^y) \in D_k$ と書く．R_k の評価値の履歴の成分は恒等写像になっている．$R_k(a, d_k^y)$ を満たすような評価関数の集合を $F(R_k(a, d_k^y))$ と書く．

ここで $f \in F(R_k(a, d_k^y))$ を一つ定めれば，定義より，f は探索履歴 $R_k(a, d_k^y)$ を完全に満たすので，

$$R_k(a, d_k^y) = S_k(a, f) \tag{5}$$

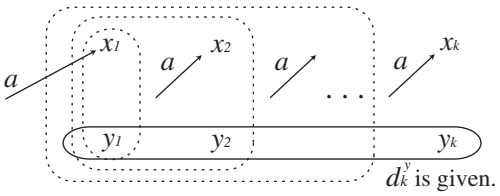

図 3 $R_k(a, d_k^y)$．y_i は与えられているので，順に x_i だけ決めていく．これにより a と d_k^y から一意的に $d_k \in D_k$ が決まるのでこれを $R_k(a, d_k^y)$ と書く．

となることはほぼ自明である（図 1，図 3 参照）．これを S_k と R_k の**整合性**と呼ぶ．厳密に整合性がとれることは付録 B の補題 3 で示している．

以後，k ステップまでの解候補の履歴を表す確率変数を $\mathcal{D}_k^x$，k ステップまでの解の評価値の履歴を表す確率変数を $\mathcal{D}_k^y$，探索アルゴリズムを表す確率変数を $\mathcal{A}$，評価関数を表す確率変数を $\mathcal{F}$ とする．例えば，

$$\mathrm{P}(\mathcal{D}_3^y = (3,7,4) \mid \mathcal{A} = a) \tag{6}$$

は探索アルゴリズムが a のとき，評価値の履歴が $(3,7,4)$ となる条件付確率を表している．

2·3 Wolpert らによる NFL

Wolpert らによる NFL は以下のように書ける．

[定理 1]（NFL） $a_1, a_2 \in A$, $d_k^y \in D_k^Y$ とすると，

$$\forall a_1\ \forall a_2\ \forall d_k^y\ [\sum_{f \in F} \mathrm{P}(\mathcal{D}_k^y = d_k^y \mid \mathcal{A} = a_1, \mathcal{F} = f) = \sum_{f \in F} \mathrm{P}(\mathcal{D}_k^y = d_k^y \mid \mathcal{A} = a_2, \mathcal{F} = f)] \tag{7}$$

つまり，探索アルゴリズムと評価関数が決まったときに，ある評価値の履歴が得られる確率は，あらゆる $f \in F$ に対して和をとるとアルゴリズムに依らない．言い換えると，平均するとどのアルゴリズムも同じ評価値の履歴を出力するということである．

探索アルゴリズムと評価関数を決めて解の探索を行うとき，そのアルゴリズムの探索性能は解の評価値の履歴によって決まる．評価値の集合 Y が順序付けられている場合，しばしば評価値の履歴の最大値によって性能が評価される．すなわち，探索の最大ステップを m とし，解の評価値の履歴として $d_m^y = (y_1, y_2, \cdots, y_m) \in D_m^Y$ が得られたとすると，

$$\max_{k \le m}\ y_k \tag{8}$$

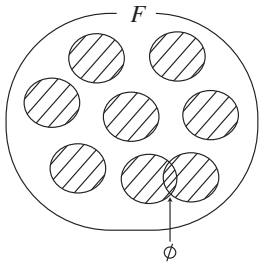

図 4 F の分割．各部分集合ごとに和をとる．各部分集合が交わりを持たずに F を覆うことが出来ていれば良い．

によりアルゴリズムの探索性能が評価される．他にも，評価値の平均や評価値の統計量を計算して探索性能が評価されるが，いずれも評価値の履歴のみに依存する．ゆえに，評価値の履歴に基づくいかなる探索性能の測り方をしても，探索性能はアルゴリズムに依らないことがわかる．

上で述べた NFL は評価値の履歴の分布がアルゴリズムに依らないことを述べており．これは非常に強い主張である．

3. 証　　明

NFL の証明を行う．まずは評価関数の空間 F を分割する補題から証明する．この補題の基本的なアイディアは [Sharpe 00] にも示されているが，証明は厳密になされておらず，特に f が重複しないことが示されていない．f が重複しないことは決して自明ではない．以下では V を任意のベクトル空間とする．

[補題 1]（F の分割の補題） $\varphi : F \to V$, $a \in A$ とすると，

$$\forall a\ \forall \varphi\ \forall k\ [\sum_{f \in F} \varphi(f) = \sum_{t \in D_k^Y} \sum_{f \in F(R_k(a,t))} \varphi(f)] \tag{9}$$

左辺はあらゆる評価関数で和をとっているが，右辺では評価関数の空間を分割してそれぞれの部分集合ごとに和をとっている（図 4 参照）．それぞれの部分集合は解の評価値の履歴 $t \in D_k^Y$ でパラメータ化されている．ここで証明すべきことは右辺の和においてすべての $f \in F$ が重複なく出てきていることである．すなわち，すべての部分集合が互いに交わりを持たないことと，すべての部分集合の要素を合計すると $|F|$ になることを示せばよい．

《証明》（補題の証明） まず，$t_1 \neq t_2\ (t_1, t_2 \in D_k^Y)$ に対し，$F(R_k(a,t_1)) \cap F(R_k(a,t_2)) \neq \phi$ と仮定する．すると，

$$\exists f\ [f \in F(R_k(a,t_1)) \wedge f \in F(R_k(a,t_2))] \quad (10)$$

であるから，この f を用いると S_k と R_k の**整合性**より

$$R_k(a,t_1) = S_k(a,f) = R_k(a,t_2) \quad (11)$$

となり，履歴は完全に一致するが，これは $t_1 \neq t_2$（評価値の履歴は異なる）に矛盾．よって，

$$F(R_k(a,t_1)) \cap F(R_k(a,t_2)) = \phi \quad (12)$$

が成り立ち，異なる $t \in D_k^Y$ に対し，$f \in F(R_k(a,t))$ は重複しない．また以上を用いれば，

$$\begin{aligned} |\bigcup_{t \in D_k^Y} F(R_k(a,t))| &= \sum_{t \in D_k^Y} |F(R_k(a,t))| && (13)\\ &= \sum_{t \in D_k^Y} |Y|^{|X|-k} && (14)\\ &= |D_k^Y| \times |Y|^{|X|-k} && (15)\\ &= |Y|^k \times |Y|^{|X|-k} && (16)\\ &= |Y|^{|X|} = |F| && (17) \end{aligned}$$

であるから，結局すべての $f \in F$ に対して和をとっていることがわかる．ゆえに命題は示された． □

上で示した補題を用いて NFL を証明する．

《証明》（NFL の証明） 補題は任意の $a \in A$ に対して成立するので，a_1 によって F を分割する．

$$\begin{aligned} &\sum_{f \in F} \mathrm{P}(\mathcal{D}_k^y = d_k^y\ |\mathcal{A} = a_1, \mathcal{F} = f)\\ &= \sum_{t \in D_k^Y} \sum_{f \in F(R_k(a_1,t))} \mathrm{P}(\mathcal{D}_k^y = d_k^y\ |\mathcal{A} = a_1, \mathcal{F} = f) \end{aligned} \quad (18)$$

ここで，$f \in F(R_k(a_1,t))$ と a_1 を用いて探索を行ったとき，S_k と R_k の**整合性**より，評価値の履歴は t になるので，$f \in F(R_k(a_1,t))$ ならば，

$$\mathrm{P}(\mathcal{D}_k^y = t\ |\mathcal{A} = a_1, \mathcal{F} = f) = 1 \quad (19)$$

$$\mathrm{P}(\mathcal{D}_k^y \neq t\ |\mathcal{A} = a_1, \mathcal{F} = f) = 0 \quad (20)$$

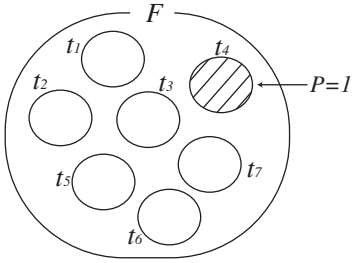

図 5 証明のイメージ．各部分集合の大きさは等しく，評価値の履歴が $t_4 = d_k^y$ となる確率は，t_4 が定める F の部分集合上では常に 1 となる．

となる（付録 B の補題 4 参照）．よって，

$$\begin{aligned} &\sum_{f \in F(R_k(a_1,t))} \mathrm{P}(\mathcal{D}_k^y = d_k^y\ |\mathcal{A} = a_1, \mathcal{F} = f)\\ &= \begin{cases} 0 & d_k^y \neq t \\ |F(R_k(a_1,t))| = |Y|^{|X|-k} & d_k^y = t \end{cases} \end{aligned} \quad (21)$$

であるので，

$$\sum_{f \in F} \mathrm{P}(\mathcal{D}_k^y = d_k^y\ |\mathcal{A} = a_1, \mathcal{F} = f) = |Y|^{|X|-k} \quad (22)$$

以上により $\mathrm{P}(\mathcal{D}_k^y = d_k^y\ |\mathcal{A} = a_1, \mathcal{F} = f)$ をあらゆる評価関数 f に対して和をとったものはアルゴリズム a_1 に依らないことが示せた．ゆえに NFL は成立する． □

4. 考　　察

4·1 証明の直観的理解

3 章の証明のイメージを図 5 に示す．3 章では関数空間 F を分割して，$t \in D_k^Y$ でパラメータ化されるそれぞれの F の部分集合 $F(R(a,t))$ ごとに和をとった．$d_k^y \in D_k^Y$ が生じる確率は，$F(R_k(a,d_k^y))$ を台（support）とし，$F(R_k(a,d_k^y))$ 上で常に 1 をとる．よって，関数空間全体で和をとるとき，$F(R_k(a,t))$ の中の f の個数がある評価値の履歴が起こる頻度を表す．ところが，アルゴリズムごとに F の分割のされ方は異なるが，k が一定なら，分割された集合の大きさ $|F(R_k(a,t))|$ はアルゴリズム a および評価値の履歴 t に依らず等しくなる．すわなち，どのアルゴリズムを用いても，F は均等に分割され，分割数も一定となる．ゆえにどの $d_k^y \in D_k^Y$ の頻度を考えても，アルゴリズムに依らず一定となる．

次に $F(R_k(a,t)) = |Y|^{|X|-k}$ となることの意味について考察する．これはあるアルゴリズムを用いた

ときに，ある評価値の履歴 t が得られるような問題が全部で $|Y|^{|X|-k}$ 個あることを意味する．アルゴリズムおよび履歴に依存せずに一定である．すなわち，GA で非常によい評価値の履歴を出す問題がいくつか存在すれば，ランダム探索で同じ評価値の履歴を出す問題が同数存在することになる．これも図 5 から直観的に理解できる．アルゴリズムにより異なるのは関数空間の分割のされ方だけであり，分割された部分集合の大きさは一定である．それぞれの部分集合は評価値の履歴 t でパラメタ化されており，t_1 でパラメタ化される部分集合の評価関数においては，必ず評価値の履歴が t_1 になることが保障されていた．ゆえに，ある評価値の履歴が得られるような問題は，アルゴリズムに依らず同数存在することになる．

4·2 NFL の仮定

NFL は，X と Y が有限という仮定の下に成立する．しかし，コンピュータで扱える数は離散数であるため，コンピュータで探索問題を解くという前提の下では，X と Y が有限という仮定は自然なものであると考えられる．

また，NFL の枠組みでは，$F = \{f : X \to Y\}$ を評価関数の集合と見なしている．$|F| = |Y|^{|X|}$ であるため，評価関数全体の集合が有限となる．離散的な解空間を持つ組み合わせ最適化問題を想定した場合は，この仮定に問題はない．しかし，一般的な関数を考えた場合は，この仮定は正しくないように思われる．ここで，Y を値域とする一般的な関数の集合 F' に対し，同値関係 $\sim$ を，

$$f \sim g \Leftrightarrow \forall x \in X \ [f(x) = g(x)] \tag{23}$$

により定める．Y を値域とする一般的な関数全体の集合 F' をこの同値関係 $\sim$ で同値類別した商集合 $F'/\sim$ は，明らかに F と一対一に対応付けることができる．ゆえに，Y を値域とする一般的な関数全体を考えた場合でも，X と Y が有限ならば，F を $F'/\sim$ に置き換えて同様に証明をすることができる．F を $\sim$ で同値類別して，F を $F'/\sim$ に置き換えることは，X 上で入出力が一致する f と g を同じ問題と見なしていることに等しい．X 上で入出力が一致する f と g を同じ問題と見なすことは，コンピュータで探索問題を解く上では自然なことであると思われる．よって，Y を値域とする一般的な関数の集合 F' に対しても，NFL は成立すると考えられる．

探索アルゴリズムに関しては次の 2 つが仮定されている．

(1) アルゴリズムが決定的である．

(2) これまで解候補として挙げられていない解を出力する．

乱数の seed を含めて考えれば，確率的なアルゴリズムも決定的であると考えられるため，1 の仮定は妥当であると思われる．また 2 の仮定は，実際の探索アルゴリズムでは当てはまらないが，探索問題の解空間が十分に大きいことを考えると，近似的には成立すると考えられる．

4·3 定理の解釈

NFL は，解空間 X と評価値の空間 Y を有限と仮定し，X 上で入出力が一致する評価関数を持つ問題を同一の問題と見なすとき，すべての評価関数において均等に和をとれば，評価値の履歴の分布はアルゴリズムに依らずに一定であることを意味している．これを数式で表現すると，以下のようになる．評価関数の確率分布を一様分布と仮定する．すなわち

$$\mathrm{P}(\mathcal{F} = f) = \frac{1}{|F|} \tag{24}$$

とすると，NFL の両辺に上式を掛ければ，

$$\begin{aligned} \forall a_1 \ \forall a_2 \ \forall d_k^y \ [\mathrm{P}(\mathcal{D}_k^y = d_k^y \ | \mathcal{A} = a_1) \\ = \mathrm{P}(\mathcal{D}_k^y = d_k^y \ | \mathcal{A} = a_2)] \end{aligned} \tag{25}$$

となる．

ここで，すべての評価関数において均等に和をとること，あるいは評価関数の確率分布を一様分布と仮定することが，定理の解釈に重大な影響を与えることが分かる．評価関数を限定する場合には，アルゴリズムによる差異が生じることもある．例えば，評価関数として，定義域に対して単調増加する関数のクラスだけを考えるときは，定義域の最大値を返すアルゴリズムが最も良いアルゴリズムである．また，単峰性の関数だけからなるクラスであれば，山登り法は遺伝的アルゴリズムよりも平均して良い成績を上げると予想できる．

一方，任意の入出力を満たす問題は，人工的にいくらでも作れるため，評価関数の分布に偏りがあるかどうかは主観的なものにならざるを得ない．

4·4 探索アルゴリズムの意味

前述したように，関数空間で均等に和をとらなければ，NFL は成立しない．本節では，評価関数の空間を限定したときに，なぜある 2 つのアルゴリズムに差が生じるかについて考察する．

探索のパフォーマンスは評価値の履歴によって与えられる．つまり，D_k^y には順序関係が定められてい

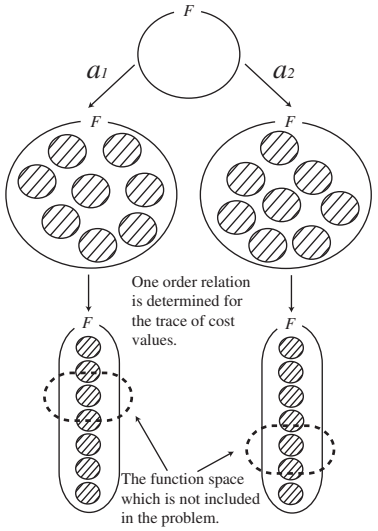

図 6 アルゴリズムによる差異．アルゴリズムは F の分割の仕方を決める．D_k^y に順序関係を一つ定めれば，F の部分集合に順序が定まる．ここで実際に問題がどのような関数空間を作るかでアルゴリズムの良さは決まる．

る．補題によれば，アルゴリズムが異なると F の分割の仕方も異なってくる．F をアルゴリズムによって分割したときのそれぞれの部分集合は，評価値の履歴 $t \in D_k^Y$ でパラメータ化されている．よって，t によって F の部分集合は順序付けられる．関数空間を狭めたときに，F の分割のどの部分が取り去られるかがそのアルゴリズムが限定された問題空間で有効か否かを決定する．つまり，順序が小さい部分集合の要素が取り除かれるアルゴリズムほど良いいアルゴリズムと考えられる（図 6 参照）．逆に取り除かれる部分が似ていれば，2 つのアルゴリズムはほとんど性能差がないと考えられる．

以上のような解釈によればアルゴリズムの役割は，評価関数の空間 F の分割の仕方を決定することにあると考えられる．対象としている問題のクラスで関数空間がどのような形をしているかと，アルゴリズムがどのように関数空間を分割するかを調べれば，ある問題空間におけるアルゴリズムの優劣を理論的に定めることができると思われる．

4·5 関 連 研 究

X の置換を σ とし，f の置換 σf を $\sigma f(x) = f(\sigma^{-1}(x))$ で定義する．[Schumacher 01] には，評価関数の集合が置換に対して閉じていることが，NFL が成立するための必要十分条件であることが示されている．ただし証明には，本稿で証明した NFL から導かれる結果を用いている．また，[Igel 03] には評価関数の分布が一様分布でない場合に，NFL が成立するための条件が示されている．

一方，[吉澤 01][山田 98] などでは，NFL を前提として，探索空間の地形構造の解析がなされている．また，探索空間上での距離と評価値に相関がある場合に，GA での探索が有効であることが示唆されている [Jones 95] [Kauffman 00].

5. お わ り に

本論文では NFL のより簡単で直観的な証明を与え，この新しい証明に基づいて，NFL が意味するところを論じた．NFL によれば，アルゴリズムの性能は問題を限定せずには語ることはできない．ゆえにアルゴリズムがどのように問題の構造を有効利用できるかを見なければならないが，これを可能とする一般的な方法を確立するのは困難である．4 章では，アルゴリズムによる関数空間の分割の仕方と問題のクラスが作る関数空間の形を調べることにより，この困難を解決できる可能性があることを示唆した．本論文で示した証明と解釈が，アルゴリズムを理論的に評価する際の一つの視点を提供できることを期待する．

謝　　辞

証明のチェックをしていただいた岩野孝之氏，柴田剛志氏，野沢康文氏に心から謝意を表したい。

◇ 付　　録 ◇

NFL の厳密な数学的モデルを示す．形式的な定義と補題の証明を与える．

A. 定　　義

X と Y を有限集合とする．k を $k < |X|$ である自然数とする．

【定義 1】 X から Y への写像**全体**の集合を F とする．すなわち，

$$F = \{f : X \to Y\} \tag{A.1}$$

【定義 2】

$$D_k^X = \{(x_1, x_2, \cdots, x_k) \in X^k \mid \forall i \, \forall j \, [i \neq j \to x_i \neq x_j] \} \tag{A.2}$$

$$D_k^Y = Y^k \tag{A.3}$$

$$D_k = D_k^X \times D_k^Y \tag{A.4}$$

【定義 3】 $||_k^x : D_k^X \times X \to X^{k+1}$ を

$$(x_1, x_2, \cdots, x_k)||_k^x x = (x_1, x_2, \cdots, x_k, x) \tag{A.5}$$

で定義する．同様に，$||_k^y : D_k^Y \times Y \to D_{k+1}^Y$ を

$$(y_1, y_2, \cdots, y_k)||_k^y y = (y_1, y_2, \cdots, y_k, y) \tag{A.6}$$

で定義する．また $||_k : D_k \times (X \times Y) \to X^{k+1} \times D_{k+1}^Y$ を

$$(d_k^x, d_k^y)||_k(x,y) = (d_k^x||_k^x x, d_k^y||_k^y y) \tag{A.7}$$

で定義する．これはただ単に履歴を加える関数である．以後，簡単のため添え字は省いて単に || と書く．

【定義 4】

$$A_k = \{a_k : D_k \to X \mid \forall d_k\ \forall i\ [a_k(d_k) \neq x_i]\ \} \tag{A.8}$$

$$A = X \times A_1 \times A_2 \times \cdots \times A_{m-1} \tag{A.9}$$

ここで $d_k = ((x_1, x_2, \cdots, x_k), (y_1, y_2, \cdots, y_k))$ である．A の最初の X は一番最初の解候補を表す．

【補題 2】 $a_k \in A_k$，$d_k = (d_k^x, d_k^y) \in D_k$ とすると，

$$\forall a_k \forall d_k\ [d_k^x||a_k(d_k) \in D_{k+1}^X] \tag{A.10}$$

《証明》 $d_k^x = (x_1, x_2, \cdots, x_k)$ とおくと，

$$d_k^x||a_k(d_k) = (x_1, x_2, \cdots, x_k, a_k(d_k)) \tag{A.11}$$

ここで $d_k^x \in D_k^X$ だから，$\forall i\ \forall j\ [i \neq j \to x_i \neq x_j]$ であり，また A_k の定義より，$\forall i\ [a_k(d_k) \neq x_i]$ であるから，重複は全くなく，$d_k^x||a_k(d_k) \in D_{k+1}^X$. □

【定義 5】

$$F(d_k) = \{f \in F \mid \forall i\ [f(x_i) = y_i]\ \} \tag{A.12}$$

ただし，$d_k = ((x_1, x_2, \cdots, x_k), (y_1, y_2, \cdots, y_k))$ である．

【定義 6】 $a = (a_0, a_1, \cdots, a_{m-1}) \in A$, $f \in F$ に対して，$S_k : A \times F \to D_k$ を

$$S_1(a,f) = (a_0, f(a_0)) \tag{A.13}$$

$$x = a_i(S_i(a,f)) \tag{A.14}$$

$$S_{i+1}(a,f) = S_i(a,f)||(x, f(x)) \tag{A.15}$$

により再帰的に定義する．補題 2 より，これは明らかに well-defined である．$S_k(a,f) = (d_k^x, d_k^y)$ として，

$$S_k^x(a,f) = d_k^x \tag{A.16}$$

$$S_k^y(a,f) = d_k^y \tag{A.17}$$

と定義する．

【定義 7】 $a = (a_0, a_1, \cdots, a_{m-1}) \in A$, $t = (y_1, y_2, \cdots, y_k) \in D_k$ に対して，

$$c_1 = (a_0, y_1) \tag{A.18}$$

$$c_{i+1} = c_i||(a_i(c_i), y_{i+1}) \tag{A.19}$$

と再帰的に計算し，$R_k : A \times D_k^y \to D_k$ を

$$R_k(a,t) = c_k \tag{A.20}$$

により定義する．補題 2 より，これは明らかに well-defined である．$R_k(a,t) = (d_k^x, t)$ として，

$$R_k^x(a,t) = d_k^x \tag{A.21}$$

$$R_k^y(a,t) = t \tag{A.22}$$

と定義する．R_k^y が $t \in D_k^y$ に対し，恒等関数となることに注意する．

B. 補　　題

【補題 3】 $a \in A$，$t \in D_k^Y$ とすると，

$$\forall a\ \forall t\ \forall f \in F(R_k(a,t))\ [S_k(a,f) = R_k(a,t)] \tag{B.23}$$

《証明》

$$a = (a_0, a_1, \cdots, a_{m-1}) \tag{B.24}$$

$$t = (\eta_1, \eta_2, \cdots, \eta_k) \tag{B.25}$$

$$S_k(a,f) = ((x_1, x_2, \cdots, x_k), (y_1, y_2, \cdots, y_k)) \tag{B.26}$$

$$R_k(a,t) = ((\xi_1, \xi_2, \cdots, \xi_k), t) \tag{B.27}$$

とおくと，$f \in F(R_k(a,t))$ より $\forall i\ [f(\xi_i) = \eta_i]$.

$$(\xi_1, \eta_1) = (\xi_1, f(\xi_1)) = (a_0, f(a_0)) = (x_1, y_1) \tag{B.28}$$

により履歴の最初は一致する．i 番目までの履歴 $d_i \in D_i$ が一致したと仮定すると，使っている探索アルゴリズムが同じなので，$\xi_{i+1} = a(d_i) = x_{i+1}$．ここで，

$$\eta_{i+1} = f(\xi_{i+1}) = f(x_{i+1}) = y_{i+1} \tag{B.29}$$

であるから，新たに加わる履歴も一致する．ゆえに数学的帰納法により，すべての探索履歴は一致する． □

【補題 4】 $a \in A$，$t \in D_k^Y$ とすると，

$$\forall a\ \forall t\ \forall f \in F(R_k(a,t))\ [\mathrm{P}(\mathcal{D}_k^y = t\ |\mathcal{A} = a, \mathcal{F} = f) = 1] \tag{B.30}$$

《証明》 補題 3 より，

$$S_k^y(a,f) = R_k^y(a,t) = t \tag{B.31}$$

ゆえに $f \in F(R_k(a,t))$ を用いれば必ず $\mathcal{D}_k^y = t$ となる．よって命題は証明された． □

◇ 参　考　文　献 ◇

[Igel 03] Igel, C. and Toussaint, M.: On classes of functions for which No Free Lunch results hold, *Information Processing Letters* (2003)

[Jones 95] Jones, T. and Forrest, S.: Fitness Distance Correlation as a Measure of Problem Difficulty for GAs, in *Proceedings of 6th International Conference on Genetic Algorithms*, pp. 184–192 (1995)

[Kauffman 00] Kauffman, S. A.: *Investigations*, Oxford University Press (2000)

[Schumacher 01] Schumacher, C., Vose, M. D., and Whitley, L. D.: The No Free Lunch and Problem Description Length, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 565–570, Morgan Kaufmann (2001)

[Sharpe 00] Sharpe, O. J.: *Towards a Rational Methodology for Using Evolutionary Search Algorithms*, PhD thesis, University of Sussex (2000)

[Wolpert 95] Wolpert, D. H. and Macready, W. G.: No Free Lunch Theorems for Search, Technical Report SFI-TR-95-02-010, Santa Fe, NM (1995)

[Wolpert 97] Wolpert, D. H. and Macready, W. G.: No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67–82 (1997)

[吉澤 01] 吉澤 大樹, 橋本 周司：巡回セールスマン問題における地形構造の解析, 人工知能学会論文誌, Vol. 16, No. 3, pp. 309–315 (2001)

[佐久間 03] 佐久間 淳, 小林 重信：確率分布推定に基づく実数値 GA の新展開, 人工知能学会誌, Vol. 18, No. 5, pp. 479–485 (2003)

[山田 98] 山田武士, Colin, R. R.：フローショップスケジュール問題の地形解析と遺伝的局所探索による解法, 情報処理学会論文誌, Vol. 39, No. 7, pp. 2112–2123 (1998)

[倉橋 03] 倉橋 節也, 勝又 勇治, 寺野 隆雄：ベイジアン最適化手法と分布推定アルゴリズムの動向, 人工知能学会誌, Vol. 18, No. 5, pp. 487–494 (2003)