

遺伝的アルゴリズムにおけるスキーマ解析の一手法

高島 一哉 宮内 秀和 岡田 三郎

takahata@cniri.go.jp

工業技術院 中国工業技術研究所

〒737-01 呉市広末広2-2-2

遺伝的アルゴリズムの理論的解析においては、遺伝子の一部分が評価関数にどのような影響を与えるかを定量的に評価することが重要である。既存手法のWST(Walsh-Schema Transform)はこのための有効な手法であるが、遺伝子が2値の場合にしか適用できない。本報告では非2値遺伝子にも適用できるOSD(Orthogonal Schema Decomposition)を紹介する。OSDは2値遺伝子に適用した場合WSTと同等であることからWSTの拡張である。WSTに高速変換アルゴリズムがあるようにOSDにも高速分解アルゴリズムがある。OSDは両親から継承した評価値の定量的評価、評価関数のスペクトル解析等に有用である。

Mathematical tool for schema analysis in genetic algorithms

Kazuya Takabatake, Hidekazu Miyauchi and Saburo Okada

takahata@cniri.go.jp

Chugoku National Industrial Research Institute, AIST, MITI

2-2-2 Hiro-Suehiro, Kure, 737-01 Japan

In theoretical analysis of genetic algorithms, it is important to evaluate how parts of a gene affect the value of the objective function. WST (Walsh Schema Transform) is an useful method for this purpose. However, we can use WST only when all genes are binary. In this report, we introduce a method called OSD (Orthogonal Schema Decomposition) which can be used also for non-binary genes. As WST has a fast transform algorithms, OSD also has a fast decomposition algorithm. OSD is useful for evaluating the inheritance from parents and the spectral analysis of objective functions.

1. はじめに

遺伝的アルゴリズムでは文字列(string)⁴⁾に対して評価値が与えられる。文字列は対立遺伝子(allele)⁴⁾と呼ばれる文字の列である(図1)。文字列に対して評価値を与える関数を評価関数(objective function)と呼ぶ。

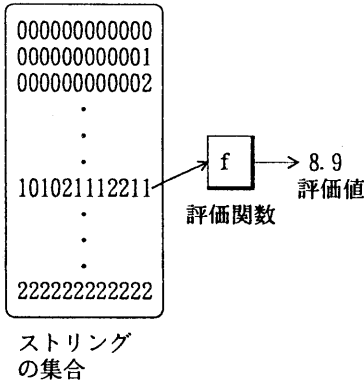


図1 文字列と評価関数

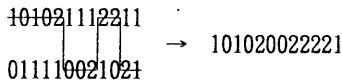


図2 交差

通常の交差(crossover)⁴⁾に基づいた遺伝的アルゴリズムでは両親の文字列の断片をつなぎ合わせて子の文字列が作られる(図2)。したがって文字列の断片は何らかの両親の形質を子に伝えなければならない。言い換えるならば文字列の断片は何らかの意味を持たなければならない。

以上のことより文字列の断片が評価値に与える影響を定量的に評価すること、すなわちスキーマ⁴⁾解析は遺伝的アルゴリズムの解析に重要な意味を持つ。スキーマ解析で用いられる数学的手法としてWalsh関数系を用いた評価関数の直交展開、WST(Walsh-Schema Transform^{2) 3) 6)}, Partition Coefficient Transform⁴⁾とも呼ばれる)がよく知られている。しかしながらWSTはストリ

ングが2値(例えば0と1)からのみなる遺伝的アルゴリズムの場合にしか適用できない。実際の遺伝的アルゴリズムでは非2値の文字列が使われる場合が多く⁵⁾、場合によっては遺伝子座により対立遺伝子が異なる複雑な文字列が用いられることもある¹⁾。

本報告では文字列が非2値の場合にも適用できるOSD(Orthogonal Schema Decomposition)⁷⁾について紹介する。OSDは2値文字列に適用した場合WSTと同義のものとなる。したがってOSDはWSTを非2値文字列の場合に適用できるよう拡張したものと考えられる。OSDは単に3値、4値...の遺伝的アルゴリズムに適用できるだけでなく、遺伝子座によって対立遺伝子が異なる複雑な場合にも適用できる。

本報告では2節でOSDの定義、意味について述べる。3節でOSDとWSTの関係について述べる。4節でOSDとWSTの使い道について述べる。

2. Orthogonal Schema Decomposition

以下の議論に用いる記号、記法、用語について定義しておく。

Ω_i : 遺伝子座*i*における対立遺伝子の集合。

$0 < |\Omega_i| < \infty$. 例: $\Omega_i = \{0, 1, 2\}$.

Ω : 文字列の集合. $\Omega = \Omega_0 \times \dots \times \Omega_{N-1}$.

例: $\Omega = \{0, 1\} \times \{0, 1, 2, 3\} \times \{0, 1, 2\}$.

* : 不定値を表す記号。

S : スキーマの集合。

$S = (\Omega_0 + \{*\}) \times \dots \times (\Omega_{N-1} \times \{*\})$. スキーマは集合として扱うこともある。例: $021 \in *21$.

% : 確定値を表す記号。

T : スキーマテンプレートの集合. $T = \{*, \%\}^N$.

\leq : SまたはTにおける順序関係。

$S_0 \dots S_{N-1} \leq u_0 \dots u_{N-1} \Leftrightarrow \forall i [u_i = * \Rightarrow s_i = *]$.

例: $*%* \leq *%*$, $*9* \leq *96$. $s \leq u$ かつ $s \neq u$ のとき $s < u$ と記す。(T, \leq)は束順序集合であり \wedge, \vee が以下のように定義される。

$$(t \wedge u)_i = \begin{cases} \% & t_i = u_i = \% \\ * & \text{その他} \end{cases}$$

$$(t \vee u)_i = \begin{cases} * & t_i = u_i = * \\ \% & \text{その他} \end{cases}$$

例: $\%* \% \wedge \%* \% = \%* \%$, $\%* \% \vee \%* \% = \%* \%$.

$x\$t(x \in \Omega, t \in T)$: ストリング x をスキーマテンプレート t によりマスクしたスキーマ.

$$(x\$t)_i = \begin{cases} x_i & t_i = \% \\ * & t_i = * \end{cases}$$

例: $012\$ \%* \% = 0*2$.

$0(s)(s \in S)$, $0(t)(t \in T)$: スキーマ s , スキーマテンプレート t の次数. s, t 中の $*$ 以外の遺伝子の数. 例: $0(*11) = 0(\%* \%) = 2$.

$L(s)(s \in S)$, $L(t)(t \in T)$: スキーマ s , スキーマテンプレート t の長さ. s, t 中の最も左に位置する $*$ 以外の遺伝子と最も右に位置する $*$ 以外の遺伝子間の距離. ただし $L(* \dots *) = 0$ と定義する. 例: $L(*1*2) = L(\% \%* \%) = 3$.

f : 評価関数. $f: \Omega \rightarrow \mathbf{R}$

V : 関数空間. $V = \{f | f: \Omega \rightarrow \mathbf{R}\}$

$\dim(A)$: 部分空間 A の次元.

Σ : 部分空間の和. $\Sigma_i A_i = \{ \Sigma_i a_i | a_i \in A_i \}$

\cdot : V での内積. $f \cdot g = \sum_{x \in \Omega} f(x)g(x)$

$\| \cdot \|$: V でのユークリッドノルム.

$$\| f \| = (f \cdot f)^{1/2}$$

\perp : 直交. $a, b \in V$, $A, B \subseteq V$ のとき

$$a \perp b \Leftrightarrow a \cdot b = 0$$

$$a \perp B \Leftrightarrow \forall b \in B [a \perp b]$$

$$A \perp B \Leftrightarrow \forall a \in A, b \in B [a \perp b]$$

$$A^\perp (A \subseteq V): A \text{ の直交補空間. } A^\perp = \{v \in V | v \perp A\}$$

2. 1. OSD の定義

$G_t, H_t (t \in T)$ を

$$G_t = \{f | \forall x, y \in \Omega [x\$t = y\$t \Rightarrow f(x) = f(y)]\} \quad (1)$$

$$H_t = \begin{cases} G_{* \dots *} & t = * \dots * \\ G_t \cap (\sum_{u < t} G_u)^\perp & \text{その他} \end{cases} \quad (2)$$

と定義する. G_t は要するに t が $\%$ を持つ遺伝子座の遺伝子にのみ依存して関数値が決まる関数の集合である. 以下の定理が成り立つ⁷⁾.

$$\text{定理1)} t \neq u \Rightarrow H_t \perp H_u \quad (3)$$

$$\text{定理2)} G_t = \sum_{u \leq t} H_u \quad (4)$$

$$\text{系)} V = \sum_{t \in T} H_t \quad (5)$$

これらの定理と系は部分空間族 $\{H_t\} (t \in T)$ の直交性(すべての H_t は互いに直交する)と完全性(すべての H_t は全空間を張る)を示している. したがって任意の関数 $f \in V$ は

$$f = \sum_{t \in T} f_t \quad (f_t \in H_t) \quad (6)$$

と互いに直交する関数に一意に展開される. これが OSD (Orthogonal Schema Decomposition) である.

2. 2. OSD の意味

関数の次数を

$$0(f) = \min\{0(t) | f \in G_t\} \quad (7)$$

で定義する. OSD は与えられた関数を次数の低い関数から順に用いておこなう逐次近似の意味を持つ. 例えば $\Omega = \Omega_0 \times \Omega_1$ のとき f_t は以下の手順で求められる.

1. $f_{**} \in G_{**}$ を $\|f - f_{**}\|$ が最小となるように定める.

2. $f_{*x} \in G_{*x}$, $f_{x*} \in G_{x*}$ を $\|f - f_{**} - f_{*x}\|$, $\|f - f_{**} - f_{x*}\|$ が最小となるように定める.

3. $f_{**x} = f - f_{**} - f_{*x} - f_{x*}$ とする.

v を与えられたベクトル, A を部分空間, P を A への射影作用素とするとき $\|v - a\| (v \in A)$ を最小化する a は $a = Pv$ である. したがって手順 1, 2, 3 は以下の手順と同じこととなる.

$$1'. f_{**} = P_{**} f$$

$$2'. f_{*x} = P_{*x}(f - f_{**}), f_{x*} = P_{x*}(f - f_{**})$$

$$3'. f_{**x} = f - f_{**} - f_{*x} - f_{x*}$$

ただし P_t は G_t への射影作用素であり

$$P_t f(x) = |x\$t|^{-1} \sum_{y \in x\$t} f(y) \quad (8)$$

で定義される. $\Omega = \{0, 1, 2\}^2$ のときの OSD の例を図 3 に示す.

ここに示した OSD の計算手順はあくまで関数の逐次近似としての OSD の側面を説明するためのものである. ちょうど WST が高速 Walsh 変換のような高速計算アルゴリズムを持つように, OSD も高速分解アルゴリズム⁷⁾を持つので, 実際の計算にはそちらを用いた方がよい.

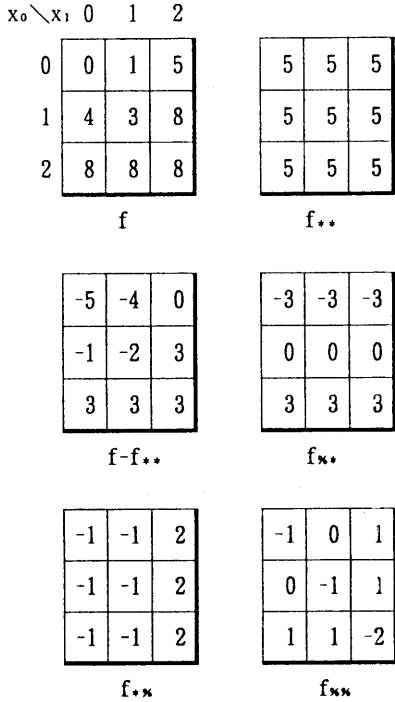


図3 OSDの例

OSDの持つもう一つの意味はスキーマの持つ正味の価値を表すということである。スキーマsの持つ価値を表す代表的な量に平均評価値

$$A(s) = |s|^{-1} \sum_{x \in s} f(x) \quad (9)$$

がある。平均評価値を f_t を用いて表すと、

$$\begin{aligned} A(x\$t) &= |x\$t|^{-1} \sum_{y \in x\$t} f(y) \\ &= P_t f(x) && \text{(式(8)より)} \\ &= P_t \sum_{u \in T} f_u(x) && \text{(式(13)より)} \\ &= P_t \sum_{u \leq t} f_u(x) && (\because \neg u \leq t \Rightarrow H_u \perp G_t) \\ &= \sum_{u \leq t} f_u(x) && (\because u \leq t \Rightarrow H_u \subseteq G_t) \end{aligned} \quad (10)$$

となる。ここで $F: S \rightarrow R$ なる関数を

$$F(x\$t) = f_t(x) \quad (11)$$

で定義すると

$$\begin{aligned} A(s) &= \sum_{v \leq t} f_v(x) && (s = x\$t \text{ とおいた}) \\ &= \sum_{v \leq t} F(x\$v) \\ &= \sum_{u \leq t} F(u) \end{aligned} \quad (12)$$

となる。式(12)はスキーマの関数AとFの関係式となっている。反転公式により

$$F(s) = \sum_{u \leq s} (-1)^{o(s)-o(u)} A(u) \quad (13)$$

となる。

さて平均評価値には以下のような欠点がある。今スキーマ*1120が高い平均評価値を持っていたとしよう。しかし同時に*1120に含まれる低次のスキーマ、例えば*112*、が同程度の高い平均評価値を持っていたならば、実は遺伝子0はあってもなくてもよい付け足しでスキーマ*112*こそが真に価値のあるスキーマだったということになる。そこでスキーマの正味価値というものを考え、スキーマsの平均評価値はsに含まれる($u \leq s$ なる)すべてのスキーマuの正味価値の総和であると考えると、式(12)よりスキーマuの正味価値というのは正に $F(u)$ であることがわかる。

3. OSDとWSTの関係

3. 1. WSTの定義

この節では2値ストリング、 $\Omega = \{0, 1\}^N$ の場合を扱う。No1(s)をスキーマsに含まれる1の個数とする。 $\phi_t: \Omega \rightarrow R$ を

$$\phi_t(x) = (-1)^{N \circ 1(x\$t)} \quad (14)$$

と定義する。 $\{\phi_t\} (t \in T)$ はWSTの基と呼ばれる。

以下の定理が成り立つ。

$$\text{定理1'} \quad t \neq u \Rightarrow \phi_t \perp \phi_u \quad (15)$$

定理2' $\{\phi_u\} (u \leq t)$ により張られる部分空間は G_t である。

これらの定理はOSDの定理1, 2に相当する。任意の関数 $f \in V$ は一意に決まる係数 $\{w_t\} (t \in T)$ を用いて次式のように $\{\phi_t\} (t \in T)$ の線型和として表される。

$$f = \sum_{t \in T} w_t \phi_t \quad (16)$$

これがWSTである。

3. 2. OSDとWSTの関係

H_t の次元について考える。便宜上、積記号 Π について命題P(i)を満たすiが1つも存在しないとき $\Pi_{P(i)} x_i = 1$ と定義する。

$$\dim(G_t) = \Pi_{t_i = *k} |\Omega_i| \quad (17)$$

は自明である。

$$\sum_{u \leq t} \dim(H_u) = \dim(G_t) \quad (18)$$

に反転公式を用いれば

$$\begin{aligned} \dim(H_t) &= \sum_{u_i} (-1)^{o(u_i)-o(u)} \prod_{u_i} |\Omega_i| \\ &= \prod_{t_i} (|\Omega_i| - 1) \end{aligned} \quad (19)$$

を得る.

2値ストリングの場合はすべての*i*について $\Omega_i = 2$ であるからすべての*t*について $\dim(H_t) = 1$ となる. したがって各*t*について H_t から1個ずつ0でないベクトルを選べばそれらは*V*の直交基となる. 実はこの直交基の1つがWSTの基である. OSDの*f_t*とは

$$f_t = w_t \phi_t \quad (20)$$

の関係がある. したがって*f_t*をそのまま表すか, 基の係数倍で表すかの違いを除いて2値ストリングに対するOSDとWSTに本質的な違いはない. 非2値ストリングに対するOSDでは, $\dim(H_t) = 1$ とはならないため, WSTのときのように基の線型和で関数を表すことはしない. WSTは変換であるのにOSDは分解である理由である.

4. OSDの使い道

ここではOSDの使い道の例をあげる. これらの使い道はまたWSTの使い道でもあるが, 非2値ストリングへの拡張によりOSDの方が適用範囲が広がっている.

4. 1. 相続分の評価

OSDによって*f(x)*は $s \leq x$ なる*F(s)*に分解される. *s*がどこから来たかということを考えれば両親からの評価値の相続分を評価できる. いま2つのストリングdadとmomから交差によってストリングkidができたとして. kidのスキーマで $s \leq \text{dad}$ となるものはdadから, $s \leq \text{mom}$ となるものはmomから受け継いだスキーマである. 例としてdad=2685, mom=1646からkid=2646ができたとして.

$$\begin{aligned} f(2646) &= F(2***)+F(26**) \\ &\quad +F(****)+F(*6**) \\ &\quad +F(**4*)+F(**6) \\ &\quad +F(*64*)+F(*6*6) \\ &\quad +F(**46)+F(*646) \\ &\quad +F(2*4*)+F(2**6) \\ &\quad +F(264*)+F(26*6) \\ &\quad +F(2*46)+F(2646) \end{aligned} \quad \left. \begin{array}{l} \text{dadから} \\ \text{momから} \\ \text{新スキーマ} \end{array} \right\} \quad (21)$$

式(21)第1, 2行がdadからの相続分, 第2~5行がmomからの相続分, 第6行以降がkidが交差によって偶然得た評価値である. もしある評価関数*f*について新スキーマ分の評価値の絶対値が両親からの相続分の絶対値に比べて大きいことが頻繁に起こるようならば, 評価値の良い両親からでもでたらめに評価値の悪い子供の産まれる可能性が高くなるので, 遺伝的アルゴリズムで効率的に最大値探索が行えないことになる.

4. 2. 評価関数のスペクトル解析

OSDは評価関数*f*を直交する関数に分解する. これは直交する正弦波の系列でフーリエ解析をするのと似ている. OSDでもパワースペクトルを図示することができる. 次数*i*の関数をひとまとめにして

$$f_{0i} = \sum_{o(t)=i} f_t \quad (22)$$

と定義し, これを*f*の次数*i*成分と呼ぶことにする. また長さ*i*のスキーマテンプレートについての関数をひとまとめにして

$$f_{1i} = \sum_{L(t)=i} f_t \quad (23)$$

と定義し, これを*f*の長さ*i*成分と呼ぶことにする.

$$f = \sum_{i \leq N} f_{0i} = \sum_{i \leq N} f_{1i} \quad (24)$$

であるが直交性より

$$\|f\|^2 = \sum_{i \leq N} \|f_{0i}\|^2 = \sum_{i \leq N} \|f_{1i}\|^2 \quad (25)$$

$$\|f_{0i}\|^2 = \sum_{o(t)=i} \|f_t\|^2 \quad (26)$$

$$\|f_{1i}\|^2 = \sum_{L(t)=i} \|f_t\|^2 \quad (27)$$

である.

図4に示す9都市の巡回セールスマン問題において起点の都市を1つ固定し, 残りの8都市を廻る順序をGref fenstetterらによる順列の符号化法¹⁾を用いてストリングとし, その巡回路長を評価値とした評価関数のパワースペクトルを図5に示す.

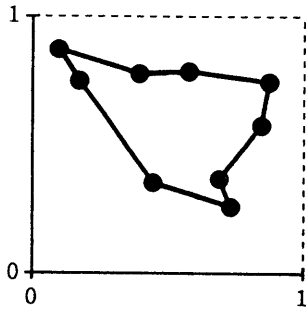


図4 巡回セールスマン問題

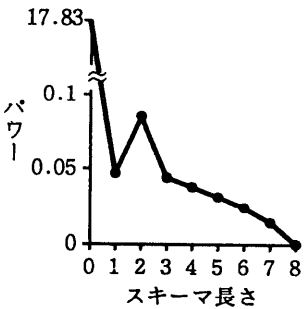
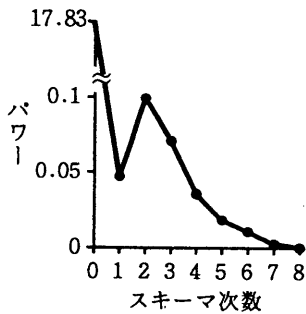


図5 評価関数のパワースペクトル

縦軸のパワーは $|\Omega|^{-1} \|f_{0i}\|^2$, $|\Omega|^{-1} \|f_{1i}\|^2$ を示している。パワーが比較的短い低次のスキーマの成分に集中していることが見てとれる。

遺伝的アルゴリズムではビルディングブロック仮説(building block hypothesis)^{22) 4)}「短い、低次の良い評価値を持つスキーマをつなぎ合わせるにより評価値の高いストリングができる」を満たさない評価関数は最大値探索しにくいと言

われている。ビルディングブロック仮説は「パワースペクトルが短い、低次のスキーマに集中していること」と言い換えられる。

5. まとめ

OSDは遺伝的アルゴリズムにおけるスキーマ解析に役立つ数学的手法である。OSDはWSTを非2値ストリングに使用できるように拡張したものと考えられる。OSDやWSTによって両親からの評価値の相続分の評価や評価関数のパワースペクトル解析が行える。

参考文献

- 1) J. Grefenstette, R. Gopal, B. Rosmatia and V. Gucht: Genetic algorithms for the traveling salesman problem; Proc. of the 1st. ICGA (1985)
- 2) D. E. Goldberg: Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction; Complex Systems, vol. 3, pp. 129~152 (1989)
- 3) D. E. Goldberg: Genetic Algorithms and Walsh Functions: Part II, Deception and Its Analysis; Complex Systems, vol. 3, pp. 153~171 (1989)
- 4) D. E. Goldberg: Genetic Algorithms in Search, Optimization & Machine Learning; Addison-Wesley (1989)
- 5) L. Davis: Handbook of Genetic Algorithms; Van Nostrand Reinhold (1990)
- 6) C. L. Bridges and D. E. Goldberg: The Nonuniform Walsh-Schema Transform; Foundations of Genetic Algorithms, pp. 13~22, Morgan-Kaufmann (1991)
- 7) 高畠一哉, 宮内秀和, 岡田三郎: 非2値遺伝子のためのスキーマ分解; システム制御情報学会論文誌, vol. 8, no. 10 (1995)