

## ビルディングブロック重複のある問題に対する D<sup>5</sup>-GA の適用

辻 美和子<sup>†</sup> 棟朝 雅晴<sup>††</sup> 赤間 清<sup>††</sup>

D<sup>5</sup>-GA [4] は互いに依存する遺伝を検出し、その情報を利用して探索を行う GA である。従来の D<sup>5</sup>-GA では、得られた依存関係情報をもとに、問題変数を同一のビルディングブロックを構成する変数の集合へと分割し、交叉において同一の集合に属する遺伝子座を同時に交換することで部分解の効果的な交換を促進した。しかしながら、実際の問題における部分解はしばしば互いに要素を共有すると考えられる。本論文では、D<sup>5</sup>-GA をビルディングブロック重複のある問題に適用するために、リンケージ集合の構造を拡張する。また、得られた問題構造情報を用いて、重複するビルディングブロックをできる限り (1) 少ない破壊で (2) 多く交換するため Yu ら [6] による交叉手法を改良する。

### Applying the D<sup>5</sup>-GA for problems with overlapping building blocks

MIWAKO TSUJI<sup>†</sup>, MASAHARU MUNETOMO<sup>††</sup> and KIYOSHI AKAMA<sup>††</sup>

The D<sup>5</sup>-GA [4] detects interdependent loci and use the information as a guidance on exploration. The existing D<sup>5</sup>-GA divides all loci into exclusive sets — called linkage sets — of loci which are linked tightly to construct a building block and exchanges loci in a same set simultaneously to enhance efficient building block mixing. However, some real-world problems have overlapping building blocks. In this paper, we extend the structure of the linkage sets to apply the D<sup>5</sup>-GA for problems with overlapping building blocks. In addition, we modify a crossover method proposed by Yu et al. [6] to achieve (1) smaller BB disruptions and (2) larger BB exchanges.

#### 1. はじめに

ビルディングブロック仮説によれば、遺伝的アルゴリズム (GA) は問題を部分問題に分割し、各部分問題における適応度の高い解を組み合わせるアルゴリズムである。このような部分解はビルディングブロック (BB) と呼ばれる。交叉における BB 交換の効率探索全体の効率の鍵となる。効率的な交叉は各ビルディングブロックをその名の通りブロックとして組み合わせ、BB 破壊を避けることで達成されると考えられる。

しかし、どの遺伝子座が同一の部分問題を構成するかを事前に知ることはしばしば困難である。この問題を克服するためには、遺伝子座間の依存関係を検出し、同一の部分問題を構成する遺伝子座を自動的に同定することが必要である。そのため戦略のひとつは、なんらかの手法により問題を陽に分割し、これに基づいて個体を再組み合わせることである [2, 4, 5, 7]。これらの手法では、全遺伝子座はリンケージ集合と呼ばれるいくつかの集合へと直和分割され、交叉の際には同一のリンケージ集合

に属する遺伝子座は同一の子個体へと受け継がれる。

本研究で取り上げる D<sup>5</sup>-GA [4] も互いに依存する遺伝を検出し、その情報を利用して探索を行う GA である。D<sup>5</sup>-GA における既存の BB 組み合わせでは、部分問題どうしが互いに重複せず、BBs どうしが平易に組み合わせられる場合が主に考えられてきた [7]。しかし、実際の問題はしばしば BB の重複を持つと考えられる。そのような問題の例として図 1 を考える。図では、遺伝子 4 が BB<sub>1</sub> と BB<sub>2</sub> の、遺伝子 7 が BB<sub>2</sub> と BB<sub>3</sub> の、要素となっている。仮に一点交叉を用いる場合、図 2 上の点線で示すように部分列を交換すれば BB<sub>2</sub> が破壊される。一方、図 2 下の位置で部分列を交換すれば、BB 破壊は起こらないが、子個体として新たな探索点を得ることができない。

本論文では D<sup>5</sup>-GA の BBs 重複のある問題への適用を行う。そのために、リンケージ集合の構造を変数の直和集合から拡張する。また、得られたリンケージ集合を用いて重複する BBs に対して、Yu ら [6] による手法を改良できる限り (1) 少ない破壊で (2) 多くの BBs を交換するための交叉戦略を提案する。

#### 2. 問題構造の同定

一般的に、問題を陽に部分問題に分割する手法は以下のような手順で実行される。まず、遺伝子座の間の依存関係を検出する。このためにさまざまな手法が提案されている。例えば、D<sup>5</sup> (Dependency Detection for Distri-

<sup>†</sup> 北海道大学大学院情報科学研究科

Division of Systems and Information Engineering, Graduate School of Information Science and Technology, Hokkaido University.

<sup>††</sup> 北海道大学情報基盤センター 大規模計算システム研究部門

Division of Large-Scale Computational Systems, Information Initiative Center, Hokkaido University.

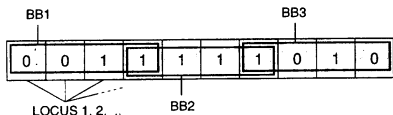


図1 重複するビルディングブロック

### D<sup>5</sup> のアルゴリズム

- (1) 個体群を初期化する
- (2) 各遺伝子座  $i$  に対して
  - (a) 個体群中の各個体で  $i$  の値を反転させたときの適応度差分を計算する
  - (b) 適応度差分を基準に個体を部分個体群に分類する
  - (c) 各部分個体群中の個体上で最小のエントロピー  $E$  を与えるような遺伝子座の集合  $w$  を探索する。こうして得られたリンケージ集合候補のうち、さらに最小の  $E$  を与えた  $w$  を  $i$  に関するリンケージ集合  $v_i$  とする。

図3 D<sup>5</sup> (Dependency Detection for Distribution Derived from  $df$ ) のアルゴリズム

bution Derived from  $df$ ) [4] は各遺伝子座に対して、その遺伝子座と依存している遺伝子座をリストする。また、LINC [1] は可能な 2 遺伝子座組の間に依存関係があるかどうか調べる。次に、これらの情報を用いて同一の部分問題を構成するであろう遺伝子座をひとつの集合にまとめる。従来の研究では、BB 重複が存在する場合には弱い依存関係は無視する [2] か、進化戦略などをを用いて最適な分割を探索 [5] することにより、全遺伝子座を直和分割してリンケージ集合を構成した。これらの手法は、ある BB に属する遺伝子が別の BB の適応度寄与に対して与える影響が小さい場合など、問題全体から見て重複が本質的でない場合には有効に機能する。しかし、重複部分がどの BB に属しているとも言えないような場合には、BB 破壊を引き起こすおそれがある。

本章では、BB 重複のある問題に D<sup>5</sup>-GA を適用する。D<sup>5</sup> による依存関係の検出について簡単に述べた後、重複を許したリンケージ集合を構成する手法について述べる。

#### 2.1 D<sup>5</sup> による依存関係の検出

D<sup>5</sup> は、ある遺伝子の値を摂動 (遺伝子の値の  $0 \rightarrow 1, 1 \rightarrow 0$  の変化) したときの適応度と元の適応度との差により分類された個体の分布に基づき変数間の依存関係を検出する手法である。各遺伝子座に対してその変数と依存する遺伝子座のリストを与える。図3にこのアルゴリ

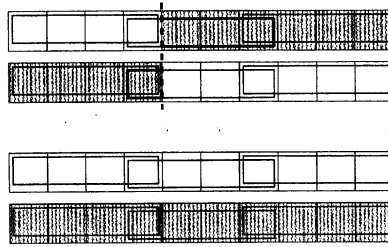


図2 BB 破壊の起こる交叉 (上), BB 交換の起こらない交叉 (下)

ズムを示す。図におけるリンケージ集合  $w$  に関するエントロピー  $E$  は次式で与えられる：

$$E(w) = -\sum_{x=1}^{2^{|w|}} p_x \log p_x$$

ただし、 $x$  は  $w$  上に定義されるスキーマ、 $p_x$  は  $x$  の部分個体群中の出現率、 $|w|$  は集合  $w$  の要素数である。

この手法の基礎となるのは以下のような考察である：問題が次式のように分解可能であると仮定する。

$$f(s) = \sum_{v \in V} f_v(s) \quad (1)$$

$v$  は部分問題  $f_v$  を定義する遺伝子座の集合、 $V$  は集合  $v$  の集合である。遺伝子座  $i$  を含む集合  $v$  の集合を  $V_i$ 、 $i$  を含まない集合  $v$  の集合を  $\bar{V}_i$  とする。BB 重複が存在しないとき、 $V_i$  はただひとつの集合を要素として持つ。BB 重複が存在し、 $i$  が複数の部分問題の構成要素となっているとき、 $V_i$  は 2 つ以上の集合を要素として持つ。

遺伝子座  $i$  に対する摂動による適応度の変化は、 $v \in \bar{V}_i$  なる  $f_v$  においては明らかに 0 である。よって、 $s_i$  を摂動後の個体として全体的変化量は、

$$\begin{aligned} df_i(s) &= f(s_i) - f(s) \\ &= \sum_{v \in V_i} (f_v(s_i) - f_v(s)) \end{aligned} \quad (2)$$

である。ゆえに  $df_i(s)$  は、 $v \in V_i$  に含まれる遺伝子座のみから決定される。

逆に、さまざまな個体に対して摂動を行って適応度の変化量を調査し、同一のもしくは近い  $df_i(s)$  を与えた個体を集めたとき、 $i$  および  $i$  に関連する遺伝子座の値は偏りを示すはずである。そのような特定の値の組み合わせを持つ遺伝子座の集合は、無作為な値の組み合わせを持つ場合と比較して、低いエントロピーを与える。よって、同一の  $df_i(s)$  を与えた個体上で低いエントロピーを与える遺伝子座の集合を探索することで、 $i$  と依存する遺伝子座を検出することができる。

#### 2.2 リンケージ集合の構造

前述のように D<sup>5</sup> は、各遺伝子座に関してその遺伝子と依存関係のある遺伝子座のリストを与える。仮に BB 重複が存在せず依存関係の検出が過不足なく実行できた場合、同一の部分問題を構成する遺伝子座はすべて同じリストを持つはずである。よって、そこから重複するリストを取り除けば、適切なリンケージ集合が得られる。

本研究では、BB 重複がある場合も同じ考え方でリストをとり除き、得られた集合をすべて交叉の際に用いるリンケージ集合とする。具体的には、以下のようにして

リンケージ集合を得る：

- (1) すべてのリストをリンケージ集合  $v_i$  ( $i = 1, 2, \dots, l$ ) とする。
- (2)  $v_i \subsetneq v_j$  ( $j = 1, 2, \dots, i-1, i+1, \dots, l$ ) なる  $v_i$  を除く
- (3)  $v_i = v_j$  なる  $v_j$  ( $j = 1, 2, \dots, i-1, i+1, \dots, l$ ) を除く

以上の操作で得られた集合を利用した交叉手法については次章で述べる。

### 3. 重複を考慮した交叉手法

BB 破壊と BB 交換による影響の解析から、Yu ら [6] は以下のような交叉手法を提案した。

- (1) なんらかの手法で BBs を同定し、重複する問題構造を把握する。
- (2) 頂点  $V$  が BB、枝  $E$  が BB ペアの重複を示すグラフ  $G = (V, E)$  を構成する。
- (3)  $G$  を  $|E| - |E_1| - |E_2|$  が最小となるような部分グラフ  $G_1 = (V_1, E_1)$ ,  $G_2 = (V_2, E_2)$  に分割する。

問題が部分問題をいくつ含むと、一度の交叉は遺伝子座をただ 2 つに分割するだけである。すなわち交換する集合と、そのまま受け継がれる集合である。以下ではこの集合を交叉集合と呼ぶ。 $|E| - |E_1| - |E_2|$  は 2 つの交叉集合の間にまたがる枝の数である。上の処理は最小の BB 破壊を与える交叉集合を構成するように設計されている。

本研究では、より小さい BB 破壊とより大きい BB 交換を実現するために、上述の手法の (2) を以下のように改良する：

- (2A) 頂点  $V$  が BB、枝  $E$  が BB ペアの重複を示すグラフ  $G = (V, E)$  を構成する。
- (2B) すべての枝  $(i, j) \in E$  に対して、もし交叉しようとする親個体  $p_1, p_2$  において重複部分  $BB_1 \cap BB_j$  が同一ならば、 $E := E - \{(i, j)\}$  とする。

提案手法では (2B) において、親個体組ごとに BB 破壊の可能性をさらに詳細に調査する。

図 1 のように BBs が重複する例の場合、既存手法では  $E = \{(1, 2), (2, 3)\}$  となる。 $\{BB_1, BB_2\}$  もしくは  $\{BB_1\}$  を交換するのが最良の選択となる。このとき  $|E| - |E_1| - |E_2| = 1$  である。

提案手法は具体的な親個体の値を含めて BB 破壊の可能性を検討する。仮に親個体と各部分解が

p1 0110011111 {0110}, {0011}, {1111}  
 p2 0010010001 {0010}, {0010}, {0001}

であった場合に、 $BB_1$  と  $BB_2$  の重複部分 (4 ビット目) の要素は同じであるから、(2B) により枝 (1, 2) は  $E$  から除かれ、 $E = \{(2, 3)\}$  となる。この新しい  $E$  に対して、最良の交叉は  $\{BB_1\}$  を交換することであると判断される。このとき、 $|E| - |E_1| - |E_2| = 0$  である。実際、 $BB_1$  を交換した後に、得られる子個体は

c1 0010011111 {0010}, {0011}, {1111}  
 c2 0110010001 {0110}, {0010}, {0001}

であり BB 破壊なしに新たな子個体を構成できる。一方既存手法では、確率  $1/2$  で  $\{BB_1, BB_2\}$  を交換することが選べれ、そのときの子個体は

c1 0010010111 {0010}, {0010}, {0111}  
 c2 0110011001 {0110}, {0011}, {1001}

となり、それぞれの子個体において  $BB_3$  が破壊される。

さらに、提案手法においては、親個体の具体的な値にあわせて交叉を設計することで、従来手法では交叉点とならなかった点でも交叉が行えるため、結果的により多くの組み合わせパターンを試みる事が可能となる。

### 4. 実験

5 ビットの二進数からなる  $\text{trap}_5$  関数が環状に重複する関数で実験を行った。この関数は次式で定義される：

$$f(x) = \text{trap}_5(x_1x_2x_3x_4x_5) + \text{trap}_5(x_4x_5x_6x_7x_8) + \dots + \text{trap}_5(x_{l-2}x_{l-1}x_lx_1x_2)$$

$l$  は個体長である。現実の問題では部分解どうしを密に符号化できるとは限らないため、 $x$  は実際の個体  $s$  の遺伝子座のランダムな並べ換えとする。

構成要素となる  $\text{trap}_5$  関数は以下のように定義される：

$$\text{trap}_5(x_1x_2x_3x_4x_5) = \begin{cases} \frac{4-u}{5}, & u = 0, 1, 2, 3, 4 \\ 1, & u = 5 \end{cases} \quad (3)$$

$u$  は  $x_i = 1$  の数である。各  $\text{trap}_5$  関数は騙し性を持つため、山登り法やより小さい部分列の組み合わせにより部分最適解を発見することがほとんど期待できない。のみならず、そのような局所的な探索はしばしば最適解 (11111) とは離れた局所最適解 (00000) を与える。そのため、この問題では初期集団で異なる個体上に得られた各最適解を破壊なしに組み合わせ、最終的に 1 つの個体に集めることが要求される。

まず、 $D^5$  による依存関係検出の後に、各遺伝子座に対する依存関係リストの同時発生確率からより強い依存関係を判断する手法である tightness detection [2] を用いて重複のないリンケージ集合を構成する手法 (以下便宜上  $D^5$ -GA1 とする)、3 節で述べた重複を考慮した既存手法 (以下  $D^5$ -GA2 とする) と提案手法 (以下  $D^5$ -GA3 とする) の 3 手法を比較する。構成要素である  $\text{trap}_5$  関数の数を 20, 30, 40 個と変化させて各 100 回の試行を行い最適解が得られた回数と、その場合に最初の最適解が得られた世代数、個体群がすべて最適解に収束した世代数を調査した。それぞれの個体長は 60, 90, 120 となる。重複を考慮した交叉においては、最小の BB 破壊の分割が複数得られた場合は、そのうちで交換できる BB 数が最も多い集合を選択した。個体数はすべての場合で 1200 とし、BB 組み合わせのみによる性能を調べるために突然変異は用いない。

結果を表 1 に示す。まず、BB 重複を考慮しない  $D^5$ -

| $l$ | 手法                  | 最適解が得られた回数 | 最適解が得られた世代数 | 最適解に収束した世代数 |
|-----|---------------------|------------|-------------|-------------|
| 60  | D <sup>5</sup> -GA1 | 0          | -           | -           |
|     | D <sup>5</sup> -GA2 | 87         | 26.0        | 35.3        |
|     | D <sup>5</sup> -GA3 | 100        | 14.7        | 27.8        |
| 90  | D <sup>5</sup> -GA2 | 53         | 33.7        | 42.5        |
|     | D <sup>5</sup> -GA3 | 99         | 19.8        | 27.8        |
| 120 | D <sup>5</sup> -GA2 | 41         | 40.3        | 49.0        |
|     | D <sup>5</sup> -GA3 | 99         | 24.8        | 33.1        |

表 1 実験結果

|                  |
|------------------|
| 0 : 0 1 57 58 59 |
| 1 : 2            |
| 2 : 3 4 5        |
| 3 : 6 7 8        |
| .....            |
| 19 : 54 55 56    |

図 4 重複しない集合の例

|                     |
|---------------------|
| 0 : 0 1 2 3 4       |
| 1 : 3 4 5 6 7       |
| 2 : 6 7 8 9 10      |
| .....               |
| 18 : 54 55 56 57 58 |
| 19 : 0 1 57 58 59   |

図 5 重複する集合の例

GA1 では、最も小さい問題である  $l = 60$  の場合で最適解を発見していない。そのため、より大きな問題での試行は省略した。図 4 に、交叉に用いられたリンケージ集合を示す。この場合、リンケージ集合は重複を持たないので、交叉においては無作為に選択したいくつかのリンケージ集合に属する遺伝子を交換する。しかし、リンケージ集合が部分問題を被覆していないため、頻繁な BB 破壊が起り、最適解を得ることができない。

対して、重複を許したリンケージ集合を図 5 に示す。図から、各部分問題が正しく検出されている。この情報は、既存手法である D<sup>5</sup>-GA2 と提案手法である D<sup>5</sup>-GA3 で共通に用いられる。にも関わらず、提案手法がすべての問題で高い割合で最適解を得ているのに対して、既存手法では問題が大きくなるにつれ最適解を得る割合が下がっている。また最適解を得るまでの世代数も、提案手法のほうが既存手法よりも短い。この原因は以下のように考えられる：既存手法では BBs が重複する位置ではできる限り交叉を避けるが、提案手法では BBs が重複していても BBs 破壊につながらない場合は交叉を行っている。結果として提案手法のほうが多くの交叉パターンを試すことができ、新たな優れた解が発見されるまでの時間（イノベーションタイム）が短縮される。ゆえに、高速に最適解に到達できる。既存手法では、交叉による組み合わせパターンが少ないため、これまでよりも優れた新たな解が発見されるのが遅い。このために、しばしば最適解を得るよりも速く個体群が局所解に収束する。

次に、提案手法について、それぞれの問題サイズで必要とされた適応度評価回数を同様の問題を解いたときの BOA [3] と比較する。BOA は有望な解候補をベイジアンネットワークを用いて確率モデル化し、モデルに従って新たな解を得る手法である。いずれの手法も各問題に対して 100 回の独立な試行を行い、98%以上の割合で最適解に収束したときの適応度評価回数の平均を調査した。BOA のパラメータである入力辺数は予備実験から決定

| $l$ | D <sup>5</sup> -GA3 | BOA     |
|-----|---------------------|---------|
| 60  | 100.032             | 96.464  |
| 90  | 142.509             | 194.265 |
| 120 | 184.970             | 300.709 |

表 2 収束までに必要とされた適応度評価回数

し、個体群サイズは最も低コストで要求される最適解の水準が得られるよう実験的に定めた。

実験の結果を表 2 に示す。表からどちらの手法も問題サイズの増加に対して、計算コストの爆発的な増加を起こすことなく適当な評価回数で最適解を得ている。特に問題サイズが大きいとき、提案手法のほうがより少ないコストで最適解を得ている。これは BB 重複を含まない分解可能関数への適用結果と同じである。

## 5. おわりに

BB 重複を持つ問題に対して D<sup>5</sup>-GA を適用した。そのために、リンケージ集合の構造を拡張し、BB 重複を持つ問題に対する交叉戦略を提案した。提案された交叉戦略は Yu らの既存手法の拡張であり、より少ない BB 破壊でより多くの BB 交換を推進することができる。

実験を行い、提案した D<sup>5</sup>-GA の性能を確認した。提案手法は既存手法よりもより高い割合で最適解を得た。さらに、最適解が得られた場合より少ない世代数しか必要としなかった。

## 参考文献

- 1) Munetomo, M. and Goldberg, D. E.: Identifying Linkage Groups by Nonlinearity/Non-monotonicity Detection, *Proc. GECCO 99*, pp. 433-440 (1999).
- 2) Munetomo, M. and Goldberg, D. E.: Linkage Identification by Non-monotonicity detection for Overlapping Functions, Technical Report IlliGAL Report No.99005, University of Illinois at Urbana-Champaign (1999).
- 3) Pelikan, M., Goldberg, D. E. and Cantú-Paz, E.: BOA: The Bayesian optimization algorithm, *Proc. GECCO 99* (1999).
- 4) Tsuji, M., Munetomo, M. and Akama, K.: Modeling Dependencies of Loci with String Classification According to Fitness Differences, *Proc. GECCO2004, LNCS 3103*, pp. 246-257 (2004).
- 5) Yu, T.-L., Goldberg, D. E., Yassine, A. and Chen, Y.-P.: Genetic algorithm design inspired by organizational theory: Pilot study of a dependency structure matrix driven genetic algorithm, *Proc. ANNIE 2003*, pp. 327-332 (2003).
- 6) Yu, T.-L., Sastry, K. and Goldberg, D. E.: Linkage learning, overlapping building blocks, and systematic strategy for scalable recombination, *Proc. GECCO 2005*, pp. 1217-1224 (2005).
- 7) 辻, 棟朝, 赤間: 適応度差分により分類された個体の分布推定を用いた遺伝的アルゴリズムの構築, 複雑系科学シンポジウム 2004, pp. 157-162 (2004).