F-035

n-BDD の遺伝的操作の提案とロボットの運動生成への適用による評価

Genetic Manipulations for *n*-BDDs and Applying them to Controlling Robots

加納 政芳 † **鈴木 悠司** ‡ **伊藤 英則** ‡
Masayoshi Kanoh Yuji Suzuki Hidenori Itoh

1 はじめに

遺伝的プログラミングは，遺伝的アルゴリズムの遺伝子型を構造的な表現（木構造，グラフ構造）が扱えるように拡張したものである．文献 [2] では，多出力二分決定グラフ (*n*-output Binary Dicision Diagram; *n*-BDD) を遺伝的プログラミングによって進化させる手法を提案している．これを食物連鎖のシミュレーションに応用し，シミュレーション環境内で植物，草食動物，肉食動物の疑似生態系を実現している．また，文献 [4] では *n*-BDD を用いたロボットの動作獲得を遺伝的プログラミングによって実現している．これらの文献では，*n*-BDD の遺伝的操作として，insertion，deletion，mutation が利用されている．しかし，これらの遺伝的操作だけでは進化の終盤において，最適解を求めるのに必要な節点が十分に存在するにもかかわらず，節点のつながりを変更する操作が乏しいために，節点数が増え続け，その結果，無駄な進化を必要とすることになる．さらには最適解そのものが獲得できない可能性もある．

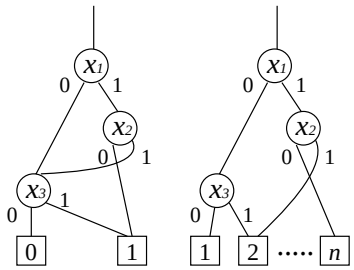
そこで本稿では，*n*-BDD の遺伝的操作として，replacement，inversion，abstraction を提案する．replacement と inversion は節点のつながりを変更する操作である．abstraction は deletion よりも強力な節点削除操作であり，*n*-BDD の簡略化と局所解へ収束を防ぐことを目的とする．本稿では，これらの遺伝的操作を用いてヒューマノイドロボットの椅子からの立ち上がり動作の獲得を行う．ロボットの動作獲得のような複雑な問題に対して，それぞれの遺伝的操作が進化に及ぼす影響を調べ，より有効な遺伝的操作の組み合わせを見出す．

2 二分決定グラフ

二分決定グラフ（BDD）は，論理関数の表現方法の 1 つであり [5, 6]，LSI の設計や，CAD システムのモデル検査などに利用されている．図 1 に BDD の例を示す．BDD は，2 種類の節点（変数節点：nonterminal vertex，定数節点：terminal vertex）と枝で構成される．枝には，変数節点の左下から出る 0 枝と，右下から出る 1 枝の 2 種類がある．図 1 では，変数節点は丸で表されており，各変数節点は割り当てられた変数を処理する．すなわち，変数節点に書かれた変数の値が 0 のときは 0 枝をたどり，1 のときは 1 枝をたどる．変数の処理は深さ 0 の位置，すなわちグラフの根に配置された変数節点から順に行われ，最終的に 1 つの四角で表される節点に到達する．この節点が定数節点であり，論理関数の出力となる．

図 1 の BDD は 3 つの変数 (x_1, x_2, x_3) を処理することができる．例えば，同図 (a) に $(x_1, x_2, x_3) = (0, 1, 1)$ を入力したときは，まず深さ 0 の位置に配置された x_1 の変数節点において 0 枝をたどり，つぎに深さ 2 に配置された x_3 の変数節点で 1 枝をたどり，出力 1 を得ることができる．同様にして出力 1 を得る組合せをすべて考えれば，この BDD は $f = x_1\overline{x_2} + x_3$ を表現していることがわかる．

多出力二分決定グラフ（Multi-terminal BDD: MTBDD）は，BDD を拡張したものであり，多出力を持つところに特徴がある．例えば，図 1(b) の MTBDD は *n* 個の出力を持つ．文献 [2] では，MTBDD のことを *n*-BDD と呼んでいるが，本稿では，一般的な MTBDD と，遺伝的プログラミングによって MTBDD を進化させる手法とを区別するために，後者を *n*-BDD と呼ぶことにする．次節では，*n*-BDD について概説する．



(a) BDD (b) MTBDD
図 1 BDD と MTBDD の例

3 *n*-BDD

図 2 に *n*-BDD の遺伝的操作を示す．*n*-BDD の遺伝的操作は 3 つからなる．

insertion は，新たな変数節点をランダムに選んだ枝の上に追加する．追加した変数節点の 0 枝か 1 枝のどちらかは，追加前に接続されていた節点に接続する．もう 1 つの枝は，任意の定数節点または，追加された変数節点よりも下位の任意の変数節点に接続する．

deletion はランダムに選んだ変数節点を削除する．削除した変数節点に接続していた枝は，削除した変数節点のいずれかの枝が接続していた節点を指すように設定する．この処理は根に対しても適用される．

mutation は，ランダムに選ばれた変数節点の持つ 0 枝または 1 枝の指す向きを変更する．ただし，新たに指す節点は，任意の定数節点または，選択された変数節点よりも下位の任意の変数節点とする．

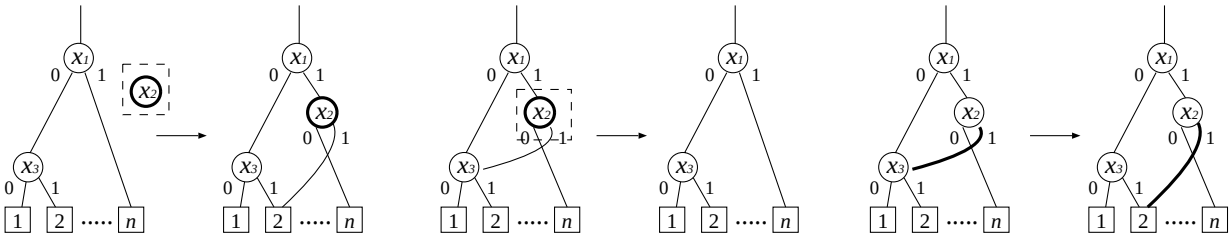
4 *n*-BDD のための新しい遺伝的操作

insertion, deletion, mutation のうち，節点から出る枝を直接変更できる遺伝的操作は mutation だけであり，またその mutation も一つの枝しか操作することができない．このため，十分な節点が存在する進化の終盤においては，適切な *n*-BDD を獲得することが困難になると考えられる．そこで本稿では，*n*-BDD の進化効率を高めるために，replacement，inversion，abstraction を提案する（図 3 参照）．以下にそれぞれの遺伝的操作について説明する．

4.1 replacement

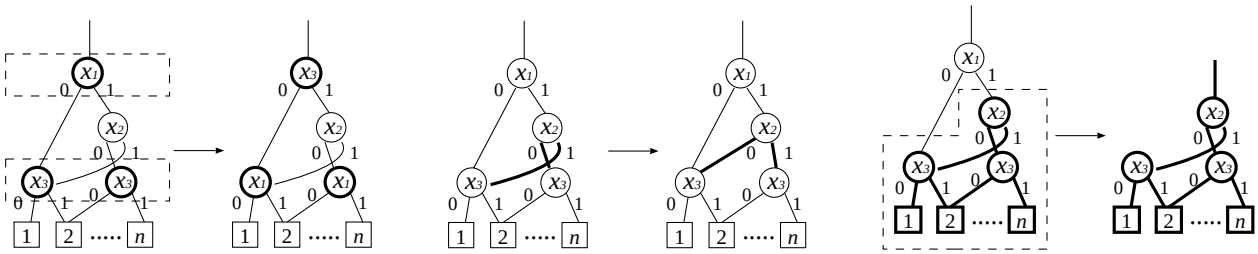
replacement（図 3(a)）は，*n*-BDD の処理する変数からランダムに 2 つ変数 (x_i, x_j) を選び，その変数を入れ替える．すなわち，x_i を処理していた変数節点は x_j を，x_j を処理していた変数節点は x_i を処理するように変更する．なお，*n*-BDD 内には変数節点として存在しない変数が選ばれた場合にも，この処理は適用される．たとえば，図 3(a) において，変数 x_4 も処理対象であるとき，replacement によってランダムに変数 x_1，x_4 が選ばれると，*n*-BDD 内の x_1 を処理する変数節点は，x_4 を処理するように変更さ

† 中京大学，Chukyo University
‡ 名古屋工業大学，Nagoya Institute of Technology



(a) insertion　(b) deletion　(c) mutation

図 2　*n*-BDD の遺伝的操作



(a) replacement　(b) inversion　(c) abstraction

図 3　*n*-BDD の新しい遺伝的操作

れる．

replacement によって，*n*-BDD の出力は大幅に変更されるので，進化の終盤など，変数節点が十分に存在し insertion や deletion では適合度が変化しにくい場面で効果を発揮できると考える．

4.2　inversion

inversion（図 3(b)）は，ランダムに選ばれた変数節点の 0 枝と 1 枝の指す向きを交換する．mutation が 1 つの枝の向きを変更するのに対して，inversion は 2 つの枝の出力を入れ替えるものである．

inversion は，mutation では適合度が変化しにくい場面で効果を発揮できると考える．

4.3　abstraction

abstraction（図 3(c)）は，ある変数節点を根とする部分木を切り出す処理である．したがって，根が選択された場合には適用されない．部分木の根となる変数節点として，進化の効率を高めるような節点を選ぶために，本稿では，1 つ前の遺伝的操作で操作された変数節点，または，操作された枝を持つ変数節点を部分木と根とすることとした．

abstraction は，変数節点の増加し複雑化したため適合度が変化しにくくなった場面で，大幅に *n*-BDD の節点を削除することにより，集団の多様性を高めることができると期待できる．

5　進化シミュレーション実験

提案する *n*-BDD の遺伝的操作の有効性を確認するために，ヒューマノイドロボットの行動則獲得についてシミュレーションする．ロボットの行動としては，椅子からの立ち上がり動作を考える (図 4). ヒューマノイドロボットとしては，図 5 に示す富士通オートメーション社製ロボット HOAP-1 を用いる．進化シミュレーションには，Open Dynamics Engine を用いた．ロボットは以下の情報をセンサにより感知する．

$$\boldsymbol{x}(t) = (x_0^W, x_1^W, x_0^K, x_1^K, x_0^A, x_1^A, x_0^B, x_1^B). \qquad (1)$$

ここで，x_i^W，x_i^K，x_i^A はそれぞれ，腰，膝，足首の角度を表現する変数であり，x_i^B は体の傾きを表す変数である（図 4）．これらの変数値は，現在の関節の角度値から表 1 のように決定した．同表において，$\theta_{\min}$，$\theta_{\max}$ はモータの可動範囲の限界値であり，それぞれ $\theta_{\min}^W = -80.0$[deg]，

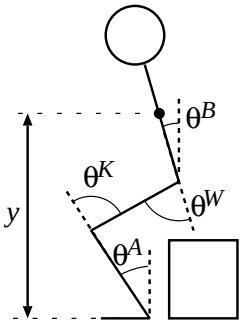


図 4　獲得動作

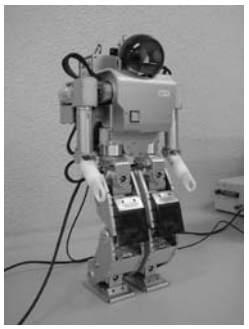


図 5　HOAP-1

表 1　角度情報と変数値の対応付け

Sensed degree	(x_0, x_1)
$\frac{3}{4}(\theta_{\max} - \theta_{\min}) \le \theta < \theta_{\max}$	(1, 1)
$\frac{1}{2}(\theta_{\max} - \theta_{\min}) \le \theta < \frac{3}{4}(\theta_{\max} - \theta_{\min})$	(1, 0)
$\frac{1}{4}(\theta_{\max} - \theta_{\min}) \le \theta < \frac{1}{2}(\theta_{\max} - \theta_{\min})$	(0, 1)
$\theta_{\min} \le \theta < \frac{1}{4}(\theta_{\max} - \theta_{\min})$	(0, 0)

表 2　行動の種類

Terminal vertex	Motor output
a_0	-20.0 [deg/sec](backward)
a_1	0
a_2	20.0 [deg/sec](forward)

$\theta_{\max}^W = 70.0$[deg]，$\theta_{\min}^K = 0.0$[deg]，$\theta_{\max}^K = 120.0$[deg]，$\theta_{\min}^A = -60.0$[deg]，$\theta_{\max}^A = 60.0$[deg]，$\theta_{\min}^B = -60.0$[deg]，$\theta_{\max}^B = 60.0$[deg] である．ロボットの行動出力は，表 2 とした．ロボットの制御時間間隔は，$\Delta t = 0.01$[s] とした．

1 試行は，ロボットが倒れるか 10 秒経過したときに終了するものとした．適合度は，ロボットの胸の位置 y[cm] とした（4 参照）．個体数は一世代 30 個体とした．遺伝的操作の組み合せは表 3 とした．なお，事前実験から，replacement の効果が高いことがわかったため，表 3 の組み合せについてのみ実験した．

図 6，図 7 にそれぞれ進化序盤，進化終盤のロボットの動作例を示す．進化序盤では，全く立ち上がることができ

表 3　ロボットの動作獲得のための遺伝的操作の組み合せと遺伝的操作の割合

実験	insertion	deletion	mutation	replacement	inversion	abstraction
(1)	0.7	0.15	0.15	-	-	-
(2)	0.7	-	-	0.15	0.15	-
(3)	0.7	-	-	0.15	-	0.15

図 6　進化序盤の動作例

図 7　進化終盤の動作例

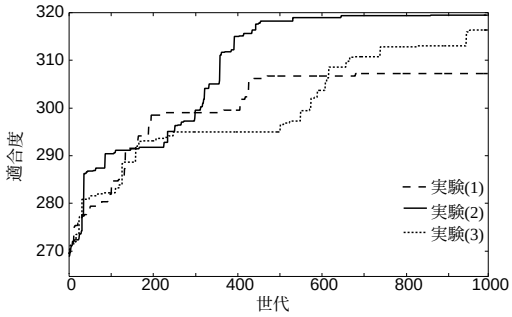


図 8　各世代の最大適合度（10 回平均）

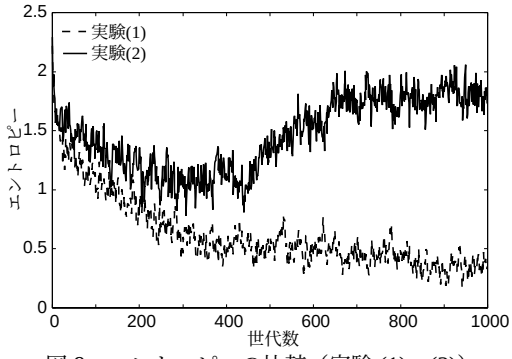


図 9　エントロピーの比較（実験 (1)，(2)）

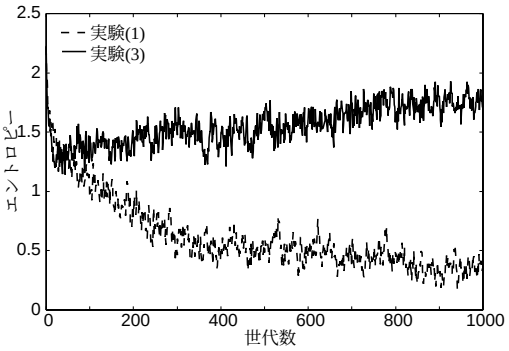


図 10　エントロピーの比較（実験 (1)，(3)）

ていないが，進化終盤では，ロボットがバランスをとりながら立ち上がっている様子が見てとれる．図 8 に実験 (1)〜(3) の適合度の推移（10 回の実験の平均）を示す．同図から実験 (2) の進化効率が最も高いことがわかる．また，実験 (1)，すなわち従来の遺伝的操作のみの場合が最も進化が遅いことがわかる．この原因は，集団の多様性にあると考える．図 9，図 10 にエントロピーの推移を示す．これらの図から，実験 (1) では進化が進むにつれて個体群の多様性が失われていくが，実験 (2)，(3) では，高い多様性を維持つつ進化が進行していることがわかる．したがって，実験 (1) では，進化序盤において良個体が発見できないと，その後，進化は収束へと向かってしまう．他方，(2)，(3) では，継続的に多様な個体が生成されるので，進化終盤においても，より良い個体を発見することができると考える．

以上より，replacement，inversion，abstraction を用いることで，効率的な進化が行えることがわかった．

6　おわりに

本稿では，*n*-BDD の遺伝的操作として，replacement，inversion，abstraction を提案した．replacement と inversion は節点のつながりを変更する操作であり，abstraction は deletion よりも強力な節点削除操作である．実験では，提案する遺伝的操作を用いてヒューマノイドロボットの椅子からの立ち上がり動作の獲得を行い，その有効性を確認した．

参考文献

[1] Holland, J. H.: *Adaption in natural and artificial systems*, University of Michigan Press (1975).

[2] 森脇康介，横井大祐，犬塚信博，伊藤英則：遺伝的プログラミング技法を用いた多出力二分決定グラフの進化 – 食物連鎖におけるマルチエージェントの進化シミュレーション，人工知能学会誌，Vol. 14, No. 3, pp. 477–484 (1999).

[3] 武藤敦子，大野　典，森脇康介，犬塚信博，伊藤英則：多出力二分決定グラフの APPLY 交叉を用いた食物連鎖モデル，電気学会論文誌 C，Vol. 121-C, No. 2, pp. 423–429 (2001).

[4] 加納政芳，伊藤英則：二分決定グラフの進化によるロボットの行動則の獲得，第 21 回東海ファジィ研究会，pp. 61–69 (2006).

[5] Akers, S. B.: Binary Decision Diagrams, *IEEE Transactions on Computers*, Vol. C-27, No. 6, pp. 509–516 (1968).

[6] Bryant, R. E.: Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Trans. on Comp.*, Vol. C-35, No. 8, pp. 677–691 (1986).