

進化的グラフ生成システム EGG

– VLSI 演算回路合成への応用 –

Evolutionary Graph Generation System

– Its Application to VLSI Arithmetic Circuit Synthesis –

○本間尚文*, 青木孝文*, 樋口龍雄*

○ Naofumi Homma*, Takafumi Aoki*, and Tatsuo Higuchi*

*東北大学 大学院情報科学研究科

*Graduate School of Information Sciences, Tohoku University

キーワード : 回路設計 (circuit design), 進化論的計算手法 (evolutionary computation), データパス (datapath), 算術演算回路 (arithmetic circuits), 組み合わせ論理回路 (combinational logic circuits)

連絡先 : 〒 980-8579 仙台市青葉区荒巻字青葉 05 東北大学 大学院情報科学研究科 樋口研究室
本間尚文, Tel.: (022)217-7169, Fax.: (022)263-9406, E-mail: homma@higuchi.ecei.tohoku.ac.jp

1. はじめに

近年, 科学技術計算やデジタル信号処理などの分野において要求される演算能力は増加の一途をたどっている. 多くの DSP (Digital Signal Processor) や ASIC (Application Specific Integrated Circuit) が乗算器や積和演算器を搭載しており, 今後は三角関数演算や行列演算などの専用回路の開発も盛んになるものと考えられる. 一方, 近年, 従来の 2 進数系にとらわれず, 冗長数系や多進数系などの特殊数系を積極的に活用した演算アルゴリズム (Beyond-Binary Arithmetic) の有効性が示されており^{1)–3)}, 2 進数 VLSI の性能限界が顕在化するにつれて, その必要性はますます高まると予想される. このような応用範囲の拡大や設計手法の多様化に伴い, 用途ごとに最適な算術演算回路を設計することは, 設計者の経験と知識が性能を左右

する困難な作業となっている.

現在開発が盛んに行われている計算機による設計自動化 (EDA: Electronic Design Automation) ツールを用いても, 算術演算回路の合成は難しい. これは, 論理式の簡単化を基本とする論理合成では, 算術演算回路特有の構造 (ハードウェアアルゴリズム) に関する最適化が考慮されていないためである. このため, 高水準合成と呼ばれる EDA の設計フローでは, あらかじめ種々の算術演算回路構造が機能ブロックの形で提供され, それらを用途に応じて割り当てることで回路設計を行う. 結果として, EDA ツールにおいても算術演算回路は人手により経験的に設計されたライブラリの形で提供されることが多い. 汎用性を重視して設計された機能ブロックは, 全ての用途に関して最適とは限らず, 最悪の場合, 重大なオーバーヘッドに

なる場合もある。

著者らは、以上のような設計問題を本質的に解決するためには、人間の創造的活動をも代替するような新しい設計パラダイムが必須であると考え、そこで、本稿では、算術演算回路構造をハードウェアアルゴリズムの知識を用いずに創発的に生成する設計手法を提案している^{4)–7)}。その中心となる着想が、進化的計算手法に基づいて回路構造を合成する進化的グラフ生成手法 (EGG: Evolutionary Graph Generation) の提案である。生物の進化戦略を模倣する EGG は、グラフによって表現された回路構造を 1 つの個体としてモデル化し、多数の個体の集団を解の候補として保持する。さらに、各個体に対して「交叉」や「突然変異」といった進化的構造操作を施すとともに、得られた個体の「評価と選択」によって集団を繰り返し更新していくことによって解の多点探索を実現する。

以下では、まず、提案する EGG の基本概念について述べ、回路構造を表現する回路グラフの定義を与える。次に、EGG のシステム構成を示し、回路グラフへの進化的構造操作を定義する。その上で、EGG による算術演算回路合成の例として、定係数乗算器の合成実験を示す。

2. 進化的グラフ生成手法 EGG

2.1 EGG の基本概念

進化的計算手法を回路合成に適用する場合、いかに回路を個体として表現し、いかに進化的構造操作を実現するかが、探索効率を高める上で重要な鍵となる。Holland らにより形式化された古典的な理論^{8),9)}によると、進化的計算手法は高度に並列化されたやり方で良い解の「積木」を探索することにより機能すると言われている。これは、積木仮説とも呼ばれ、良好な解はいくつもの良い積木の組み合わせにより作り出されることを意味している。そこで、著者らは、以下のような回路の構

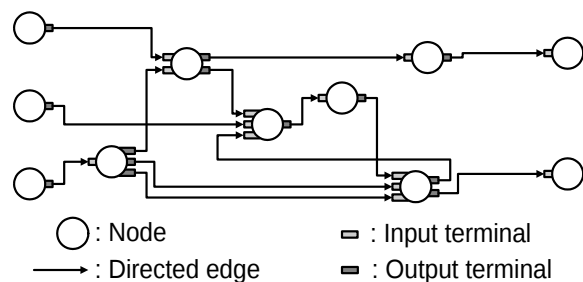


Fig. 1 回路グラフ

成における積木仮説から、進化的グラフ生成手法 (EGG: Evolutionary Graph Generation) を提案している。

回路の構成における積木仮説：意味のある機能をもつ回路は、その機能の一部を実現する部分回路の組み合わせによって構成される。

一般に、積木仮説が成立するには、実際の機能が近い個体は個体表現も類似している必要があると言われている。また、進化的計算手法では、積木を効率よく組み替える構造操作が必要不可欠となる。これに対して、従来の進化的計算手法である GA (Genetic Algorithms) や GP (Genetic Programming) は、それぞれ個体構造として「文字列」や「ツリー (木)」を採用し、構造操作の対象を「部分文字列」や「部分木」とする。そのため、回路構造に問題を限定した場合、表現のオーバーヘッドが大きく、ほとんど実用的ではなかった¹⁰⁾。一方、提案する EGG では、回路構造を直接表現可能な「回路グラフ」を個体表現として定義する。また、回路グラフを部分回路グラフの集合と解釈し、部分回路グラフに基づく構造操作を定義する。定義された構造操作は、部分回路グラフの交換や置換を行い、回路グラフに限定した効率的な探索を可能とする。

2.2 個体モデル

EGG では、個体を図 1 に示す回路グラフで表現する。回路グラフは、ノード、端子および有向辺の 3

つの集合からなる．ノードには，機能ノード，入力ノード，出力ノードがあり，ノード集合には1個以上のノードが含まれる．機能ノードは，1個以上の入力端子および出力端子を有する．一方，入力ノードは出力端子を，出力ノードは入力端子を各々1個だけ有する．ノードへの接続はすべて入出力端子を介して行われ，各端子は2本以上の有向辺とは同時に接続しないものとする．また，有向辺は必ずノードの出力端子から入力端子へ向かうものとする．ここで，出力端子と入力端子が常に1対1で対応し，未接続の端子が存在しない回路グラフを，特に完全回路グラフと呼ぶ．EGGでの個体は完全回路グラフを対象としており，それ以外の回路グラフは扱わない．

回路グラフ G は以下のように定義される．

$$G = (N^G, T_O^G, T_I^G, \nu_O^G, \nu_I^G, \epsilon^G) \quad (1)$$

ここで， N^G はノードと呼ばれる元からなる非空有限集合であり， T_O^G と T_I^G はそれぞれ出力端子と入力端子と呼ばれる元からなる有限集合である． ν_O^G は T_O^G から N^G への関数であり， ν_I^G は T_I^G から N^G への関数である． ν_O^G と ν_I^G は N^G に含まれるノードの種類によって一意に決定される．一方，有向辺を表す ϵ^G は， $S_O^G \subseteq T_O^G$ および $S_I^G \subseteq T_I^G$ とするとき， S_O^G から S_I^G への全単射である．ここで， $u \in S_O^G$ および $v \in S_I^G$ について，以下の関係が成り立つ．

$$v = \epsilon^G(u) \Leftrightarrow u \text{ から } v \text{ への有向辺が存在する.} \quad (2)$$

$S_O^G = T_O^G$ かつ $S_I^G = T_I^G$ のとき， G は完全回路グラフとなる．

回路グラフ G の部分回路グラフ G' は， $N^{G'} \subseteq N^G$ となる $N^{G'}$ から誘導された部分グラフとして定義される．同様に， G' に対する補部分回路グラフ $\overline{G'}$ は， $N^{\overline{G'}} (= \overline{N^{G'}} = N^G - N^{G'})$ から誘導された部分グラフとして定義される．また， G' と $\overline{G'}$ のいずれにも含まれない有向辺の集合をカットセットと呼ぶ．特に， G' から $\overline{G'}$ への有向辺を正方向と

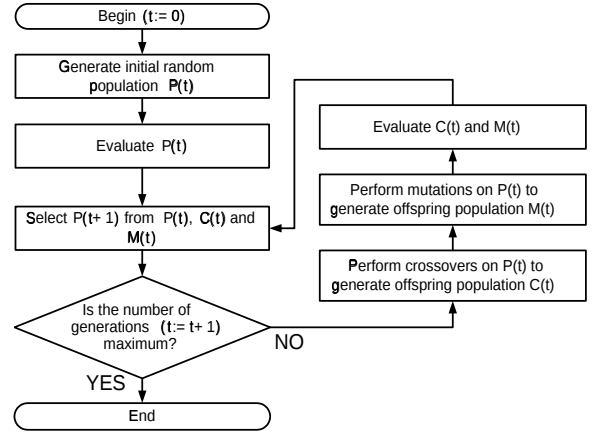


Fig. 2 EGG のシステムフロー

し， $\overline{G'}$ から G' への有向辺を負方向とする．任意の回路グラフ G_{p1} ， G_{p2} の任意の部分回路グラフ G'_{p1} ， G'_{p2} において，そのカットセットが整合しているとは，要素となる有向辺の数が正負ともに等しい場合を指す．

回路グラフの構造を特徴づける用語を以下に示す．回路グラフにおける半歩道とは，ノード n_{i-1} とノード n_i ($i = 1, 2, \dots, l$)の間に有向辺が存在するような，連続したノードの列 $n_0, n_1, \dots, n_l$ ($l \geq 1$)である．ここで，任意の2ノード間に半歩道が存在する回路グラフを連結な回路グラフと呼ぶ．全ての有向辺が n_{i-1} から n_i へ向う半歩道を歩道と呼ぶ．全てのノードが異なる歩道を道と呼ぶ． $n_0 = n_l$ の場合も道とし，これを閉路と呼ぶ．

2.3 システムの概要

図2に提案するEGGのシステムフローを示す．またEGGシステムの動作を制御する主要なパラメータを表1に示す．EGGシステムは，まず，初期世代の個体となる回路グラフをランダムに生成する．個体の生成は，あらかじめ設定した個体数 (Population size) 分だけ行われる．このとき，目的とする回路構造のグラフ論的な特徴を満たす個体のみを生成することで，探索空間を見通しよく限定することが可能である．例えば，目的とする回

路が組み合わせ回路であれば、解となる回路構造にフィードバックループ等が含まれないため、個体を閉路のない連結な回路グラフとすることで探索空間を大幅に削減できる。

次に、新たに生成された回路グラフに対して評価を与える。EGG では、回路グラフを回路としての機能と性能の両面から評価する。機能評価とは、個体の機能が目的とする機能とどの程度合致しているかを数値化したものである。一方、性能評価とは、遅延時間、面積および消費電力などに基づいて、個体の回路としての性能を概算したものである。生成された回路グラフは、まず、その回路としての機能を検証される。その後、機能評価値と性能評価値の線形加算から評価値を決定される。

全ての個体の評価値を決定した後、進化的構造操作の対象となる個体を選択する。進化的計算手法における選択は、これまでいくつかの方法が提案されており、個体の進化効率に大きな影響を与えることが知られている。本稿における EGG システムでは、選択方法の基本モデルとして知られる評価値比例戦略を採用する。評価値比例戦略は、重み付きルーレットモデルとも呼ばれ、各個体の評価値に比例した確率で子孫を残す。

選択された個体に対して、EGG の進化的構造操作を行い、新たな回路グラフを生成する。進化的構造操作には、「交叉」と「突然変異」があり、次節で詳しく述べる。本稿における EGG システムでは、各世代における個体数は一定なため、選択は個体数分行われる。次世代の個体群は、進化的構造操作により新たに得られた個体と、選択されたが進化的構造操作の対象とならなかった個体から形成される。EGG では、以上のような世代交代を、あらかじめ設定された世代 (Max. num. of generations) に到達するまで繰り返し行う。

Table 1 EGG の主要なパラメータ

パラメータ名	意味
Population size	1 世代の個体数
Max. num. of generations	終了条件となる最大世代数
Max. num. of nodes	一個体のノード数の上限
Min. num. of nodes	一個体のノード数の下限
Num. of inputs	個体の持つ入力ノード数
Num. of outputs	個体の持つ出力ノード数
Crossover rate	集団における交叉率
Mutation rate	集団における突然変異率

2.4 進化的構造操作

EGG における進化的構造操作は、回路構造における積木仮説に基づき、部分回路グラフの交換や置換として実現される。また、回路グラフの完全性を保持するため、操作対象となる部分回路グラフのカットセットに着目し、カットセットの整合条件を考慮した操作が行われる。以下では、EGG における交叉と突然変異について述べる。

交叉は、2 つの回路グラフの部分回路グラフを組み合わせて新たな回路グラフを作り出す操作である。図 3 (a) に交叉例を示す。(i) まず、親として選択された回路グラフ対 G_{p1} , G_{p2} のうち、一方の回路グラフから任意の部分回路グラフ G'_{p1} をランダムに選択する。(ii) 次に G'_{p1} のカットセットに整合する部分回路グラフ G'_{p2} を G_{p2} からランダムに選択する。(iii) その後、選択された G'_{p1} と G'_{p2} を交換し、未接続の端子をランダムに接続することで、新たな回路グラフ G_{c1} と G_{c2} を生成する。図 3 (a) の交叉例では、正方向のカットセット要素数が 2、負方向のカットセット要素数が 3 で整合のとれている部分回路グラフを交換している。回路グラフ G_{c1} は、以下のように定義される。

$$G_{c1} = (N^{G_{c1}}, T_O^{G_{c1}}, T_I^{G_{c1}}, \nu_O^{G_{c1}}, \nu_I^{G_{c1}}, \epsilon^{G_{c1}}) \quad (3)$$

ここで、 $\overline{G'_{p1}}$ および $\overline{G'_{p2}}$ をそれぞれ G'_{p1} , G'_{p2} の補部分回路グラフとすると、集合 $N^{G_{c1}}$, $T_O^{G_{c1}}$ および $T_I^{G_{c1}}$ は、

$$N^{G_{c1}} = N^{\overline{G'_{p1}}} \cup N^{G'_{p2}} \quad (4)$$

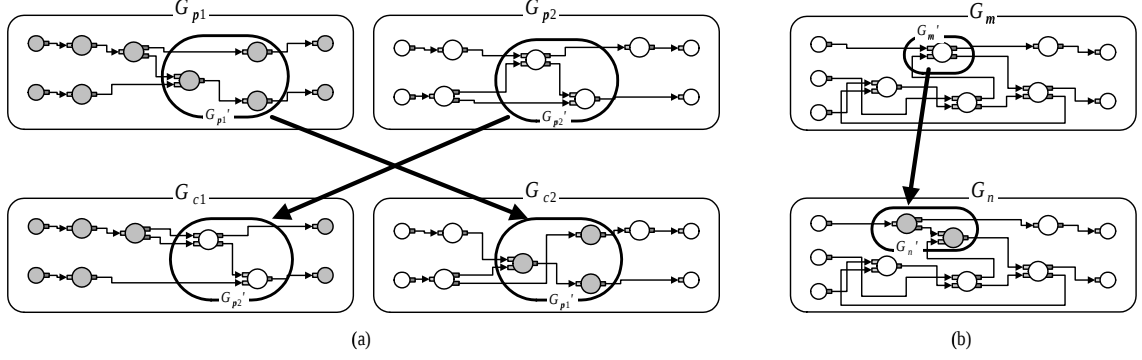


Fig. 3 EGGにおける構造操作：(a) 交叉，(b) 突然変異。

$$T_{O^{G_{c1}}} = T_{O^{G'_{p1}}} \cup T_{O^{G'_{p2}}} \quad (5)$$

$$T_{I^{G_{c1}}} = T_{I^{G'_{p1}}} \cup T_{I^{G'_{p2}}} \quad (6)$$

と定義される。また、関数 $\nu_{O^{G_{c1}}}$, $\nu_{I^{G_{c1}}}$ および $\epsilon^{G_{c1}}$ は、

$$\nu_{O^{G_{c1}}} = (\nu_{O^{G'_{p1}}} \setminus \overline{T_{O^{G'_{p1}}}}) \cup (\nu_{O^{G'_{p2}}} \setminus \overline{T_{O^{G'_{p2}}}}) \quad (7)$$

$$\nu_{I^{G_{c1}}} = (\nu_{I^{G'_{p1}}} \setminus \overline{T_{I^{G'_{p1}}}}) \cup (\nu_{I^{G'_{p2}}} \setminus \overline{T_{I^{G'_{p2}}}}) \quad (8)$$

$$\epsilon^{G_{c1}} = \epsilon^{\overline{G'_{p1}}} \cup \epsilon^{G'_{p2}} \cup \epsilon^{p^c} \cup \epsilon^{n^c} \quad (9)$$

と定義される。ここで、関数 “ $f \setminus A$ ” は、定義域を A に限定した関数 f とする。また、 ϵ^{p^c} は、定義域が $C_{O^{G'_{p2}}}$ で値域が $C_{I^{G'_{p1}}}$ となる任意の全単射であり、 ϵ^{n^c} は、定義域が $C_{O^{G'_{p1}}}$ で値域が $C_{I^{G'_{p2}}}$ となる任意の全単射である。カットセットの整合条件から、 ϵ^{p^c} と ϵ^{n^c} が存在することは明らかであり、 $\epsilon^{G_{c1}}$ もまた全単射となる。よって、式 (9) から $\epsilon^{G_{c1}}$ の定義域と値域はそれぞれ $T_{O^{G_{c1}}}$, $T_{I^{G_{c1}}}$ となり、 G_{c1} は完全回路グラフとなる。同様に、 G_{c2} も式 (4)–(9) における G'_{p1} と G'_{p2} および $\overline{G'_{p1}}$ と $\overline{G'_{p2}}$ を交換することで定義される。

一方、突然変異は、回路グラフの任意の部分回路グラフをランダムに生成した部分回路グラフで置き換える操作である。図 3 (b) に突然変異例を示す。(i) まず、適当に選択された回路グラフ G_m から、任意の部分回路グラフ G'_m をランダムに選択する。(ii) 次に、 G'_m とカットセットの整合する部分回路グラフ G'_n をランダムに生成する。(iii) そ

の後、 G'_m を G'_n に置換し、未接続の端子をランダムに接続することで、新たな回路グラフ G_n を生成する。図 3 (b) の突然変異例では、正方向のカットセット要素数が 2、負方向のカットセット要素数が 2 で整合のとれている部分回路グラフを置換している。回路グラフ G_n は、以下のように定義される。

$$G_n = (N^{G_n}, T_{O^{G_n}}, T_{I^{G_n}}, \nu_{O^{G_n}}, \nu_{I^{G_n}}, \epsilon^{G_n}) \quad (10)$$

ここで、 G'_n をランダムに生成された部分回路グラフ、 $\overline{G'_m}$ を G'_m の補部分回路グラフとすると、集合 N^{G_n} , $T_{O^{G_n}}$ および $T_{I^{G_n}}$ は、

$$N^{G_n} = N^{\overline{G'_m}} \cup N^{G'_n} \quad (11)$$

$$T_{O^{G_n}} = T_{O^{\overline{G'_m}}} \cup T_{O^{G'_n}} \quad (12)$$

$$T_{I^{G_n}} = T_{I^{\overline{G'_m}}} \cup T_{I^{G'_n}} \quad (13)$$

と定義される。また、関数 $\nu_{O^{G_n}}$, $\nu_{I^{G_n}}$ および ϵ^{G_n} は、

$$\nu_{O^{G_n}} = (\nu_{O^{\overline{G'_m}}} \setminus \overline{T_{O^{\overline{G'_m}}}}) \cup \nu_{O^{G'_n}} \quad (14)$$

$$\nu_{I^{G_n}} = (\nu_{I^{\overline{G'_m}}} \setminus \overline{T_{I^{\overline{G'_m}}}}) \cup \nu_{I^{G'_n}} \quad (15)$$

$$\epsilon^{G_n} = \epsilon^{\overline{G'_m}} \cup \epsilon^{G'_n} \cup \epsilon^{p^c} \cup \epsilon^{n^c} \quad (16)$$

と定義される。 G'_n における未接続な出力端子の集合を $C_{O^{G'_n}}$ 、未接続な入力端子の集合を $C_{I^{G'_n}}$ とすると、 ϵ^{p^c} は、定義域が $C_{O^{G'_n}}$ で値域が $C_{I^{\overline{G'_m}}}$ となる任意の全単射であり、 ϵ^{n^c} は、定義域が $C_{O^{\overline{G'_m}}}$ で値域が

Table 2 2進SW数系に基づくノード

記号	機能	方程式表現
3-2	3入力2出力桁上げ保存加算 (出力符号の反転, 出力分岐あり)	$\lambda_C C + \lambda_S S = X_0 + X_1 + X_2$ $\lambda_C, \lambda_S \in \{-1, 1\}$
FSA	バイアス補正演算と 2入力1出力桁上げ伝搬加算	$Y_0 = X_0 + X_1$
N-S	出力を上位へN桁ゼロつめ シフト (Nは1, 2, 4)	$Y_0 = 2^N X_0$
IN	回路の入力	$\lambda_I X_0, \lambda_I \in \{-1, 1\}$
OUT	回路の出力	Y_0

$C_I^{G'_n}$ となる任意の全単射である．カットセットの整合条件から, ϵ^{pc} と ϵ^{nc} が存在することは明らかであり, ϵ^{G_n} もまた全単射となる．よって, 式(16)から, ϵ^{G_n} の定義域と値域はそれぞれ $T_O^{G_n}$, $T_I^{G_n}$ となり, G_n は完全回路グラフとなる．

以上のように, EGGにおける進化的構造操作は, カットセットの整合性に注目することで, 未接続な入出力端子が生じないように回路グラフの構造を操作する．そのため, 選択された個体が完全回路グラフであれば, 操作後に生成される回路も完全回路グラフとなる．

3. 定係数乗算器の合成実験

本章では, 提案するEGGを用いた算術演算回路の合成実験の例として, 定係数乗算器の合成実験を示す． R を定数とした $Y = RX$ の演算を行う定係数乗算は, デジタル信号処理の分野において出現頻度の高い演算である．また, マイクロプロセッサやDSP (DSP: Digital Signal Processor) に搭載される定係数乗算器は, 主にカウンタツリーによる組み合わせ回路として実現され¹¹⁾, 次のような特徴を示す．(i) 最適構造が定係数値により大きく異なる．(ii) 同一の定係数乗算を実現する構造が多数存在する．(iii) 系統的に最適構造を導出する方法がない．以下では, 実験条件として, 回路グラフを構成するノードの機能やその回路グラ

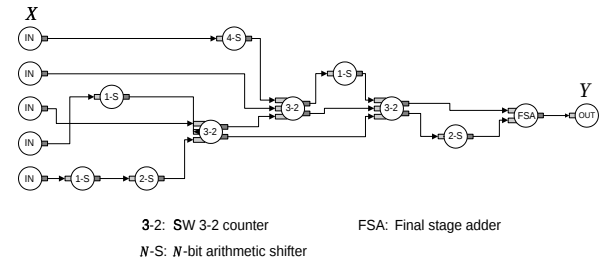


Fig. 4 個体となる回路グラフの例

フの評価方法を示す．その後, 異なる係数値の定係数乗算器を50個合成した結果を示す．

3.1 実験条件

まず, 回路グラフを構成するノードの数系および機能を示す．EGGによる算術演算回路合成では, 回路グラフを算術演算アルゴリズムを表現するデータフローグラフと解釈し, より抽象度の高いモデリングを行う．回路グラフのノードは, 算術演算アルゴリズムを記述するための抽象的な基本演算を表す．一方, 有向辺は, オペランドを規定する数系の型および数値データを表すために用いられる桁情報を有する．EGGの特徴の一つは, ノードの設定により任意の数系に基づくハードウェアアルゴリズムを合成可能な点である．本実験では, 高性能な定係数乗算アルゴリズムを合成可能な2進SW数系¹²⁾に基づくハードウェアアルゴリズムを合成する．表2に本実験で使用した2進SW数

Table 3 主要なパラメータの設定値

Population size	500
Crossover rate	0.7
Mutation rate	0.2
Max. num. of generations	500
Max. num. of nodes	50
Min. num. of nodes	2

系に基づくノードの種類を示す。

本実験では、組み合わせ回路によって実現される定係数乗算器を合成対象とするため、閉路のない連結な完全回路グラフを個体として扱う。ここで、閉路のない連結な完全回路グラフを完全回路木 (CCT: Complete Circuit Tree) と呼ぶ。実験で用いた EGG は、任意の CCT を生成するアルゴリズムと任意の回路グラフを CCT かどうか判定するアルゴリズムを、それぞれ初期世代生成の時と構造操作直後に使用することで、CCT のみに個体構造を限定している。個体となる CCT の例を図 4 に示す。

次に、本実験における個体の機能評価値と性能評価値の計算方法について示す。まず、機能評価値の導出方法を示す。CCT である個体から得られる機能は、記号的な機能検証⁵⁾により、 $Y = \hat{R}X + f(X)$ という数式で表される。ここで、 X および Y は整数の入出力を表す変数であり、 $\hat{R}$ は定係数値である。また、 $f(X)$ は機能検証の過程で生じる中間変数により表される非線形関数であり、 $f(X) = 0$ のとき、その回路グラフは任意の入力に対してある定係数乗算を行う。 $f(X)$ の項は中間変数の個数 $|f(X)|$ に比例して、個体の評価値を減少させる。本実験の機能評価値は、以下のように、得られた定係数値 $\hat{R}$ と目的とする係数値 R との相関を取ることで導出される。

まず、 R および $\hat{R}$ を符号無し 2 進数系の系列に

展開し、

$$R = r_0 2^0 + r_1 2^1 + r_2 2^2 \cdots = \sum_{j=0}^{\|R\|-1} r_j 2^j \quad (17)$$

$$\hat{R} = \hat{r}_0 2^0 + \hat{r}_1 2^1 + \hat{r}_2 2^2 + \cdots = \sum_{j=0}^{\|\hat{R}\|-1} \hat{r}_j 2^j \quad (18)$$

と表す。ここで $\|R\|$, $\|\hat{R}\|$ はそれぞれ、 R , $\hat{R}$ の系列語長である。また、 $r_j, \hat{r}_j \in \{-1, 0, 1\}$ である。二つの系列の相関値は、系列語長に差がある場合、語長がそろうまで短い系列をシフトさせ、一桁シフトするごとに計算する。系列語長の差を n ($= \|\hat{R}\| - \|R\|$) とすると、シフト量 k ($0 \leq k \leq n$) における相関値 M_k は、

$$M_k = \begin{cases} \frac{1}{\|\hat{R}\|} \sum_{j=0}^{\|\hat{R}\|-1} \delta(\hat{r}_j - r_{j-k}) & n \geq 0 \\ \frac{1}{\|R\|} \sum_{j=0}^{\|R\|-1} \delta(\hat{r}_{j-k} - r_j) & n < 0 \end{cases} \quad (19)$$

と与えられる。 $\delta(x)$ はデルタ関数であり、

$$\delta(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases} \quad (20)$$

と定義される。なお、上記の計算において、定義されていない桁は常に 0 とする。式 (19) から得られた相関値から、機能評価値 F は、

$$F = \max_{0 \leq k \leq |n|} [100M_k - C_1 k - C_2 |f(X)|] \quad (21)$$

と定義される。ここで、 C_1 および C_2 は定数であり、本稿では $C_1 = 2$, $C_2 = 5$ とする。

一方、本実験における性能評価値 P は、個体の回路としての面積と遅延時間から、

$$P = \frac{C_3}{DA} \quad (22)$$

と定義される。ここで、 A はグラフ表現を実際の回路にマッピングした際のモジュール間配線数であり、 D は入力から出力までの最大カウンタ段数である。以下では、 A を面積、 D を遅延時間と呼ぶ。面積と遅延時間の積 DA は、乗算器の性能を示

Table 4 定係数乗算器の DA : (a) EGG により合成された構造, (b) Wallace 木構造.

Index	Coefficient	DA		Index	Coefficient	DA		Index	Coefficient	DA	
		(a)	(b)			(a)	(b)			(a)	(b)
1	-2077	810	822	18	10075	1600	1824	35	17012	1324	1324
2	13492	2040	2250	19	2609	1280	1296	36	52	358	370
3	-20844	1548	1600	20	-17127	1564	1596	37	-29824	771	792
4	27155	2335	2375	21	5755	1556	1612	38	30321	1616	1616
5	-17614	1556	1572	22	-1749	1528	1528	39	19878	2290	2290
6	-1353	1276	1276	23	6674	1336	1336	40	-32424	1320	1320
7	10304	406	408	24	-24570	882	912	41	15315	1580	1620
8	-14338	456	458	25	-26881	1368	1376	42	30248	1336	1336
9	18639	1604	1604	26	4134	1072	1092	43	11452	1512	1576
10	-27400	1312	1320	27	14577	1372	1372	44	-15697	1572	1588
11	-4444	1256	1276	28	-1257	1260	1276	45	26204	2080	2275
12	-28961	1360	1392	29	3461	1332	1332	46	-28097	1384	1384
13	28959	1360	1392	30	-8390	1288	1312	47	22732	2245	2285
14	3548	819	819	31	14993	1560	1596	48	26605	1668	1668
15	-9566	1524	1572	32	-18597	1596	1628	49	-24213	2290	2375
16	-28565	1664	1664	33	9959	1560	1612	50	-27804	1576	1584
17	4833	1316	1316	34	-14886	1356	1356				

す尺度として頻繁に用いられる^{13),14)}. C_3 は P/F を 0.1 程度におさえるように設定する定数である. このため, 進化は主に機能評価を基に進み, 機能評価が一定のとき性能評価が有効となる.

3.2 実験結果

表3に実験で用いた EGG の主要なパラメータ値を示す. 本実験では定係数値を -32767 から 32767 の範囲でランダムに 50 個選択し, 16 ビット精度の定係数乗算器を係数値ごとに各 1 回合成した. その結果, 全ての係数値において, 完全に機能を満たす ($F = 100$) 解を合成することに成功した. 各試行における最良個体の評価値の推移を図5に示す. ここで, 横軸は世代数を表し, 縦軸は各世代における最良個体の評価値を表す. また, 合成した解の性能を表4に示す. ここで, 性能の指標となる DA 積 (遅延時間 D と面積 A の積) は, 値が小さいほど高性能であることを示す. EGG に

より合成された構造の DA を (a) とし, 人手により設計した Wallace 木構造の DA を (b) として示す. Wallace 木構造は, 定係数乗算器を最小のカウンタ段数で実現する構造の一つであり, 優れた構造として広く知られている. 表4の (a) と (b) の比較から, EGG により合成された全ての構造が, Wallace 木構造と同等以上の性能を有していることを確認できる.

進化の様子を示す具体的な例として, 図6に係数値 26204 を合成した際の進化過程を示す. ここで, 図の縦軸は世代数であり, 横軸は機能評価値 F および性能の指標となる DA 積である. 図中の点は各個体の評価値を表す. 図7に初期世代において最も評価値の高い個体を示す. 図6から, 世代が進むにつれ個体の機能評価が上昇していく様子が確認できる. 機能評価がある一定の値になると, 性能評価が有効に作用し, 性能評価が向上していく. 例では, 第 289 世代において完全に機能の合

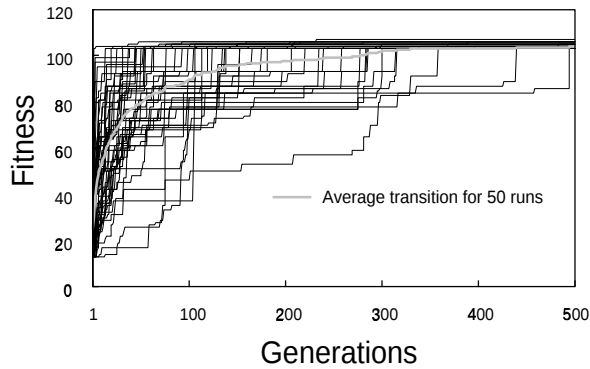


Fig. 5 最良個体の評価値の推移

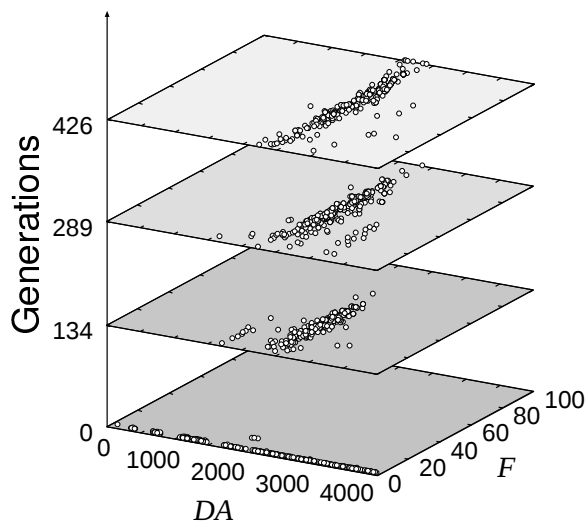


Fig. 6 定係数値 26204 の合成における進化過程

致した個体が生成される (図8)。しかし、その回路構造の DA は 2345 であり、人手により設計した Wallace 木構造 (DA は 2275) と比べ、冗長な部分が含まれる。しかし、さらに進化を重ねることで性能が向上し、第 426 世代には、 DA が 2070 となる構造が得られている (図9)。なお、最大世代数までの計算時間の平均は約 3.6 時間である。

4. おわりに

本稿では、回路構造を進化的計算手法に基づいて創発的に合成する手法として進化的グラフ生成手法 (EGG) を提案し、その有効性と動作を定係数乗算器の合成実験により示した。EGG の特徴は、

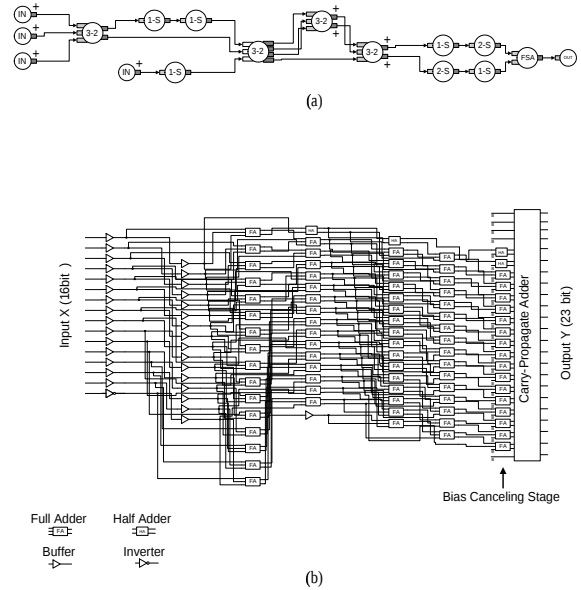


Fig. 7 初期世代における最良個体 ($F = 17$): (a) 回路グラフ, (b) 回路グラフに対応する回路構造

個体となる「回路グラフ」の「グラフ」としての側面と「回路」としての側面の両方に着目することで、求める回路構造を効率よく合成することにある。今後は、グラフ的な性質を利用した探索空間の削減方法や部分回路グラフの機能に着目した評価関数の設定方法を検討することで、EGG の適用可能な範囲を明確にしていくことが重要である。

References

- 1) 青木孝文, 常世大史, 樋口龍雄: 商選択テーブルを用いない高基数除算器の構成, 電子情報通信学会論文誌, J79-D-I-7, 416/424 (1996)
- 2) 青木孝文, 天田博章, 樋口龍雄: 冗長複素数系に基づく実数/複素数再構成型算術演算回路の構成, 電子情報通信学会論文誌, J80-D-I-8, 674/682 (1997)
- 3) T. Aoki, H. Nogi and T. Higuchi: High-radix CORDIC algorithms for VLSI signal processing, Proc. of the 1997 IEEE Workshop on Signal Processing Systems, 183/192 (1997)
- 4) T. Aoki, N. Homma and T. Higuchi: Evolutionary design of arithmetic circuits, IEICE Trans. Fundamentals, E82-A-5, 798/806 (1999)
- 5) N. Homma, T. Aoki and T. Higuchi: Evolutionary graph generation system with symbolic verification for arithmetic circuit design, IEE Electronics Letters, 36-11, 937/939 (2000)
- 6) N. Homma, T. Aoki and T. Higuchi: Evolutionary synthesis of fast constant-coefficient multipliers,

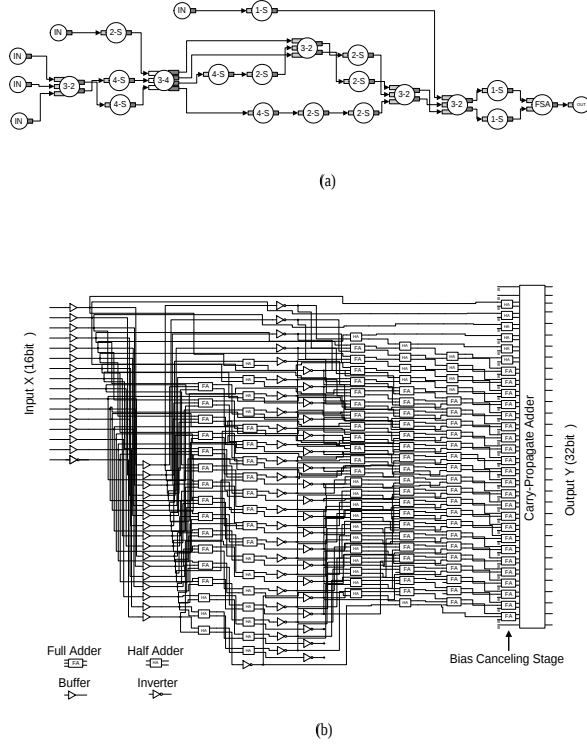


Fig. 8 第289世代における最良個体 ($F = 100$, $DA = 2345$): (a) 回路グラフ, (b) 回路グラフに対応する回路構造

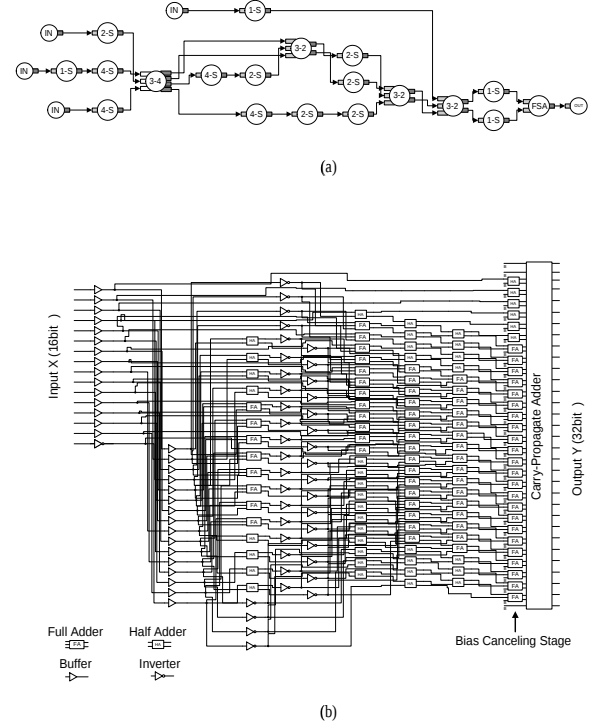


Fig. 9 第426世代における最良個体 ($F = 100$, $DA = 2070$): (a) 回路グラフ, (b) 回路グラフに対応する回路構造

- IEICE Trans. Fundamentals, E83-A-9, 1767/1777 (2000)
- 7) N. Homma, T. Aoki and T. Higuchi: Evolutionary graph generation system with transmigration capability for arithmetic circuit design, Proc. of The 2001 IEEE International Symposium on Circuits and Systems, 171/174 (2001)
 - 8) J. H. Holland: Adaptation in Natural and Artificial Systems, University of Michigan Press (1975)
 - 9) E. D. Goldberg: Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley (1989)
 - 10) F. J. Miller, P. Thomson and T. Fogarty: Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, 105/131, John Wiley & Sons (1997)
 - 11) R. I. Hartley: Subexpression sharing in filters using canonic signed digit multipliers, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., 43-10, 677/688 (1996)
 - 12) T. Aoki, Y. Sawada and T. Higuchi: Signed-weight arithmetic and its application to a field-programmable digital filter architecture, IEICE Trans. Electronics, E82-C-9, 1687/1698 (1999)
 - 13) P. R. Brent and T. H. Kung: The area-time complexity of binary multiplication, Journal of the Association for Computing Machinery, 28-3, 521/534 (1981)

- 14) R. H. Srinivas and K. Parhi: A fast radix-4 division algorithm and its architecture, IEEE Trans. Computers, 44-6, 826/831 (1995)