

F\_024

組み合わせ最適化アルゴリズムを用いた有限語長多項式近似におけるハードウェアコスト最小化

Hardware cost minimization in finite word length polynomial approximation using combinational optimization algorithms

小谷 光史†  
Koji Kotani

佐々木 孝雄†  
Takao Sasaki

豊嶋 久道†  
Hisamichi Toyoshima

1. はじめに

ハードウェア記述言語による回路設計では、初等関数やユーザー独自の関数など、ハードウェア記述言語では直接表現できない関数がある。そのような場合、有限語長多項式近似を用いることによって表現することが可能となるが、近似式を用いることによる演算誤差と回路規模との間にトレードオフの関係がある。そこで本研究では、組み合わせ最適化アルゴリズムを用いることによって、演算誤差を許容範囲内に抑えた上で、加算器、乗算器などのハードウェアコストを最小にする方法を提案する。

2. 多項式近似

ハードウェア記述言語では直接表現できない関数を実現する方法として多項式近似法が知られている。[1]これは、任意の関数  $F(X)$  を、次式のように  $X$  に関する  $N$  次の多項式として近似する方法である。

$$F'(X) = C_0 + C_1X + \dots + C_NX^N \quad (1)$$

各係数が無限語長の場合、近似誤差を数値的に最適化する手法が様々知られているが、ハードウェア実現する場合、各係数は有限語長で量子化されるため、近似誤差やハードウェアコストの最適化は、組み合わせ最適化問題として考える必要がある。

3. ハードウェアモデル

ハードウェアモデルは、多項式近似式を基にして実現する。入力値を各々係数と乗算し、その結果を全て加算する。さらに、このままだと理想値との誤差が出てしまうので、誤差の補正にテーブルを用いる。テーブルは式(2)で表し、ビット幅により数が増加する。

$$Table(X) = F(X) - F'(X) \quad (2)$$

ハードウェアモデルを図1に示す。

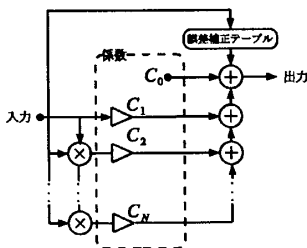


図1: ハードウェアモデル

3.1 ハードウェアコスト

ハードウェアコストとは、実際に回路を構成する上で必要となる回路規模のことをいう。本研究では、加算器、乗算器といった演算回路とテーブルが必要となる。これら全てについてのハードウェアコストを考えると、全体の回路規模が加算器何個分に相当するかと換算し、ハードウェアコストとして定義する。

4. 最適化手法

多項式近似を用いて演算を表現するためには、係数を決定しなければならない。ここで、演算  $F(X)$  が解析的な関数であるならば、テーラー展開などの展開法を使うことによって、係数の決定をすることが可能である。しかし、他の任意の関数(解析的でない関数)では係数の決定をすることが困難な場合がある。よって、係数の値を決定するには組合せ最適化が必要となる。

この問題は、そのままでは探索範囲が膨大になっているので、ハイブリッドアルゴリズムを採用する。まず、広域的な探索を得意とする遺伝的アルゴリズム(GA)では、演算の理想値との誤差が小さくなるような係数を初期値とし求め、局所的な探索を得意とするタブーサーチ(TS)では、テーブル参照を用いることで誤差が無くなるようにし、その時のハードウェアコストが最小となる係数を探索する。

探索法の流れ図を図2に示す。

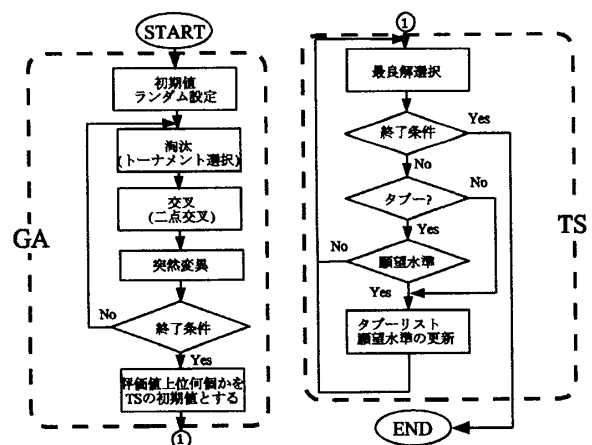


図2: 探索法のアルゴリズム

4.1 遺伝的アルゴリズム(GA)

GAでは、トーナメント選択、2点交叉、突然変異、評価値計算といったように単純GAによって探索を行う。

†神奈川大学大学院工学研究科電気電子情報工学専攻

4.2 遺伝的アルゴリズムでの遺伝子表現

それぞれの係数値を  $Q_n$  フォーマットの固定小数点数で表現し、図3のように  $M$  個の係数を並べて遺伝子とする。

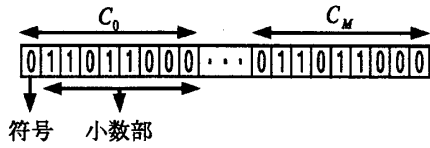


図3: GA の遺伝子表現

ただし、ここでは、 $M < N$  とする。つまり、GA では解空間を制限し探索を行う。これは、GA の目的が最適解を求めるのではなく、タブーサーチの適切な初期値を求めることであるからである。

4.3 遺伝的アルゴリズムでの評価値計算

理想値  $F(X)$  と測定値  $F'(X)$  による誤差の絶対値の最大値の逆数を評価値として式 (3) で表す。

$$fitness = \frac{1}{\max\{|Table(X)|\}} \quad (3)$$

$X_R$ : 評価範囲,  $X \in X_R$

4.4 タブーサーチ (TS)

TS は、短期メモリ、長期メモリ、願望水準を用いて探索を行う。短期メモリは、既にチェックした解に戻ることを避ける。長期メモリは、解の選択回数を保存し選択回数でペナルティを与え他の解への探索を続ける。願望水準は、短期メモリの働きを弱めるために用いる。

近傍の生成方法として確率的係数選択による近傍の生成を行う。これは、一つの係数を微小に変化させた係数セットを近傍解とするが、近傍解は全ての係数に対して生成するのではなく、前述の短期メモリ、長期メモリを用いて確率的に選択されたいくつかの係数に対してのみ生成する。 [2]

4.5 タブーサーチでの解表現

TS ではハードウェアコストを考慮した探索を行うため、GA とは違った表現方法が必要となる。

そこで、TS では CSD 表現を用いる。これは、 $\{0,1\}$  からなる 2 進数表現の代わりに、 $\{-1,0,1\}$  を用いた数で表現する方法の冗長 2 進表現 (Signed Digit 表現, SD 表現) というものを用い、最小の重みを持つ冗長 2 進表現の中で非零ビットが連続して現れないようにしたものを用いる。この方法を用いてそれぞれの係数値を求め、図4のように係数を並べることで解表現とする。

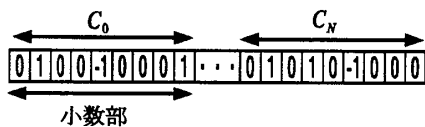


図4: TS の解表現

4.6 タブーサーチでの評価値計算

TS では、演算回路と誤差補正のテーブルのハードウェアコストを評価し、最適化を行う。但し、演算回路のうち、係数乗算器は CSD によるシフト加算の加算器数で表し、入力演算回路は適当な重み付けをした加算器数に換算して評価する。また、テーブルのハードウェアコストも入力ビット数の最大値に応じて加算器数に換算して評価する。

演算器のハードウェアコストを  $S_c$ 、テーブルのハードウェアコストを  $T_c$  とすると、これらの和の逆数を評価関数とする。

$$fitness = \frac{1}{S_c + T_c} \quad (4)$$

5. シミュレーション結果

本手法をいくつかの関数に対して評価シミュレーションを行い、適用した関数はいずれも 1 入力 1 出力のものとし、入出力ビット数はいずれも 9 ビットとした。GA, TS のパラメータは以下のように設定する。

- GA: 個体数:200, 世代数:200  
 交叉率:70%, 突然変異率:10%, M:4
- TS: タブーリストの数:7, 近傍解の数:4, N:6

それぞれ 100 回試行した結果、平均のハードウェアコストとハードウェアコストの最小値を表1に示す。

表1: ハードウェアコスト

演算対象	平均コスト	最小コスト
$(\cos(X))\log(X+1)$	71.1	67
$(\sin(X))\log(X+1)$	78.1	60
$\frac{x^2}{\log(X+1)}$	70.1	45

テーブルのみを用いた場合のハードウェアコストは 81 であり、本手法により、ハードウェアコストを削減することができた。

6. むすび

本研究では組み合わせ最適化アルゴリズムを用い、有限語長多項式近似とテーブル参照を併用することで、係数の誤差をある範囲までに制限して求め、ハードウェアコストを削減することに成功した。

今後の課題として、ハードウェアコストの換算方法を、ゲートレベルで行うことによって評価し、コストの削減を計るといったことが必要となる。

参考文献

- [1] 高橋, 豊嶋: デジタル回路設計入門, 培風館, pp112-113, 2000.
- [2] 豊嶋, 松本, 佐々木: 確率的タブーサーチによる係数乗算ブロックで構成された FIR デジタルフィルタの設計, 電子情報通信学会論文誌, VOL. J88-A, 1, pp99-103, January 2005.