2008－MPS－68 (13)
2008／3／5

線形変換回路最適化における計算時間削減アルゴリズム

鈴木 麻衣† 佐々木孝雄† 豊嶋 久道†

† 神奈川大学工学部

あらまし 線形変換回路は，DCT，DFT など様々な変換で使われている回路であり，ハードウェアで実現する場合，遅延時間，回路規模をできるだけ小さくすることが要求される．そのために定数乗算器をシフトと加減算器に置き換え，同一入力における演算部を共有化する方法が提案されているが，それらの組合せ最適化問題では，係数行列のサイズの増加に伴い計算時間や最適性に問題が生じる．本研究では，入力の順序によって遅延時間，回路規模が変化するため，入力順序を組合せ最適化問題として扱う．その際，線形変換回路の係数行列を分割し解空間を絞り，合成においてシフト加減算を決定する際の探索に制限をかけ，効率的に探索を行うことにより，計算時間削減を行う方法を提案する．

Computation Reduction Algorithm of Linear Transform Ciruit Optimization

Mai SUZUKI†, Takao SASAKI†, and Hisamichi TOYOSHIMA†

† Faculty of Engineering, Kanagawa University

Abstract A linear transform circuit is often used in various transform such as DCT and DFT. By replacing each constant multiplier with shifts and adders/subtractors, and sharing the partial circuit whose input is the same, the circuit area can be reduced. However, in the combinatorial optimization regarding circuit sharing, the problem occurs that the computation time is lengthened and the optimality deteriorates asthe size of the coefficient matrix grows. In this study, the above optimization is treated as a combinatorial optimization regarding the input order. Furthermore, to shorten the computation time, we propose the method of partitioning the coefficient matrix and reducing the search space in synthesizing coefficients.

1. はじめに

線形変換回路は，DFT，DCT など様々な変換において使われている回路である．この線形変換回路をハードウェアで実現する場合，遅延時間，回路規模をできるだけ小さくすることが要求される．

線形変換回路合成の従来法として，同一入力に対して複数の定数乗算（multiple constant multiplication,MCM）回路として合成し，各々の出力を互いに加算し回路規模を削減する方法が知られている．この MCM 回路は，回路中の乗算部をシフトと加減算器に置き換え，同一入力における演算部を共有化させるアルゴリズムで合成できる [1]．また，線形変換回路において複数の入力をまとめて扱い，MCM 回路の入力を複数に拡張した回路と見なし，MCM 回路の一係数を線形変換回路の係数行列の係数列とすることで線形変換回路を合成する方法もある [2]．

しかし，このシフト加算の組み合わせの最適化問題では，係数行列のサイズ，係数の bit 幅が大きくなるに従い解空間が膨大になり，計算時間や最適性に問題が生じる．

本研究では，入力の順序によって遅延時間・回路規模が変化するため，入力順序を組合せ最適化問題として扱う．最適化手法には，シフト加減算の組み合わせ決定に時間がかかることを考慮し，単点探索の SA に複数の初期解を与え，多点探索として用いる．その際，線形変換回路の係数行列を分割し解空間を絞り，合成においてシフト加減算をの探索に制限をかけ，効率的に探索を行うことにより，計算時間削減を行う方法を提案する．

2. 線形変換回路

2.1 複数の定数乗算（MCM）回路

複数の定数乗算（MCM）回路とは，信号処理，画像処理，数値計算等においてしばしば現れる複数の定数（N

個）と変数間の乗算処理を行う回路で，次式で表される．ここで，x は入力値，a_i は係数（乗数）を表す．

$$y_i = a_i x \quad (i = 0, 1, 2, \cdots, N-1) \tag{1}$$

乗算は，加減算とシフトの組み合わせで置き換えることで同等の回路が得られ，演算を共有することで加算器数の削減が可能である．

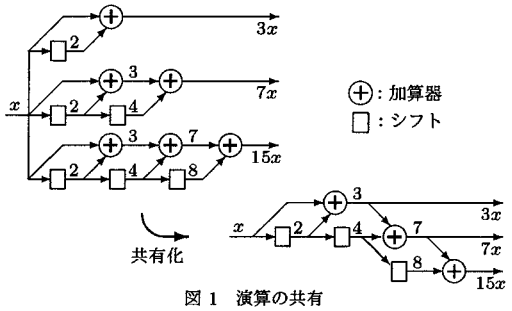

図 1　演算の共有

2.2　線形変換回路

線形変換回路は，n 個の入力 $x_j (j = 0, 1, \cdots, n-1)$ と m 行 n 列の係数行列 $\mathbf{T}$（式 (2)）との乗算により，m 個の結果 $y_i (i = 0, 1, \cdots, m-1)$ を出力する変換回路である．出力を $\mathbf{Y} = [\, y_0 \; y_1 \; \cdots \; y_{m-1} \,]^{\mathrm{T}}$，入力を $\mathbf{X} = [\, x_0 \; x_1 \; \cdots \; x_{n-1} \,]^{\mathrm{T}}$ とすると，一般式は式 (3) で表される．

$$\mathbf{T} = \begin{bmatrix} t_{0\,0} & \cdots & t_{0\,j} & \cdots & t_{0\,n-1} \\ \vdots & & \vdots & & \vdots \\ t_{i\,0} & \cdots & t_{i\,j} & \cdots & t_{i\,n-1} \\ \vdots & & \vdots & & \vdots \\ t_{m-1\,0} & \cdots & t_{m-1\,j} & \cdots & t_{m-1\,n-1} \end{bmatrix} \tag{2}$$

$$\mathbf{Y} = \mathbf{T}\mathbf{X} \tag{3}$$

線形変換回路合成の従来法として，同一入力 x_j に対し係数行列 $\mathbf{T}$ の j 列目の係数列 $\mathbf{T}_j = [\, t_{0j} \;\; t_{1j} \;\; \cdots \;\; t_{m-1\,j} \,]$ を MCM 回路として合成し，それらを互いに加算する方法が知られている [1]．この方法では，複数の MCM 回路の出力を互いに加算する必要があるため，加算器数の増加が問題となる．

これに対し，式 (3) の線形変換の，i 行目の係数列 $t_{ij} (j = 0, 1, \cdots, n-1)$ を $\mathbf{T}_i$ とすると，

$$y_i = \mathbf{T}_i \mathbf{X} \tag{4}$$

となり，これを MCM 回路の入力を複数に拡張した回路とみなすことができ，MCM 回路の一係数を係数行列の係数列とすることで線形変換回路を合成できる．

3.　最　適　化

線形変換回路では，一度に複数の係数を合成するため，シフト加減算，共有化の組み合わせは膨大であり，決定的手法では回路規模，遅延時間の削減は期待できない．

そこで本研究では，入力 x_j の順序を入れ替えることにより係数行列 $\mathbf{T}$・シフト加減算の組み合わせが変化し，遅延時間，回路規模が変化することに着目する．入力の順序を解表現とし，組合せ最適化問題として扱い，入力順序を最適化することにより加算器数を削減する．

また，シフト加減算の組み合わせ決定に時間がかかることを考慮し，単点探索の SA に複数の初期解を与え，多点探索として用いる．

3.1　SA(Simulated Annealing)

SA は，金属などの結晶組織を高温から徐々に冷却していき，エネルギー準位の低い，ひずみのないきれいな結晶を得る方法（Annealing）を模倣した手法である．

現在の解よりも新しい解の評価値がよければ新しい解が採択され，悪くなる場合でも，現在の解と新しい解の評価値の差分 $\Delta Fitness$，現在の温度 T，乱数 R に対し，

$$R < e^{-\Delta Fitness/T} \quad (0 \leq R \leq 1) \tag{5}$$

となる場合は，新しい解を採択する．

3.2　最適化手法

初めに初期集団となる解を生成し，評価計算を行う．その初期集団の中から評価値の高い解を複数選び，SA の初期解とする．終了条件を満たすまで近傍解の生成，評価計算，採択を繰り返して準最適解を求める．近傍解は，選ばれた 1 点をランダムに選んだ解に置き換えることにより生成し，保存された最良解が 30 世代変化しなければ終了する．図 2 に最適化の流れを示す．

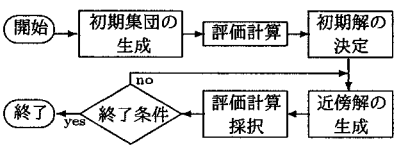

図 2　最適化手法のフローチャート

3.3　解　表　現

線形変換の入力の順序を解とする．入力の順序は，図 4 のように入力 x_j の順番を順序表現 [3] を用いて表現する．この方法では，まず入力のリストを作成し，次に来る入力が残りの入力のリストの中で何番目に相当するかの番号を並べる．

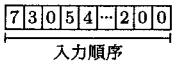

図 3　解表現の例

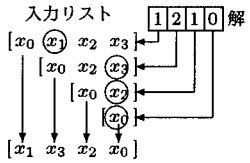

図 4　順序表現の例

3.4 評価計算

評価値 *Fitness* を次式で定義する．

$$Fitness = \frac{1}{adder} \tag{6}$$

ここで *adder* は，以下の線形変換回路合成によって計算される総加算数とする．

線形変換回路は，各入力 x_j をシフト加算することにより合成される．合成は，i 行目の係数列 $\mathbf{T}_i$ 単位に行い，シフト加減算を決定するために各 x_j に対応する中間係数列と評価値を用いる．合成における中間値を $s_0x_0 + s_1x_1 + \cdots + s_{n-1}x_{n-1}$ とすると，その中間係数列は $\mathbf{S} = [\, s_0\ s_1\ \cdots\ s_{n-1}\,]$ と表現でき，以下の手順で係数列を合成する．

(a) 初期値として，K（初期値 n）個の中間係数列 $\mathbf{S}_k (k = 0, \cdots, K-1)$ を用意する．

$$\mathbf{S}_k\text{の各要素 } s_{ki} = \begin{cases} 1 & (i = k) \\ 0 & (i \neq k) \end{cases} \tag{7}$$

(b) 2 つの中間係数列を式 (8) のようにシフト加減算し $\mathbf{S}'$ とし，式 (9) によって計算される評価値 F が最小となる $\mathbf{S}'$ を $\mathbf{S}'_{min}$ として求める．評価値は係数列 $\mathbf{T}_i$ と $\mathbf{S}'$ の差の総和とする．

$$\mathbf{S}' = 2^a\mathbf{S}_{i0} \pm 2^b\mathbf{S}_{i1} \tag{8}$$

$$\text{評価式：} F = \sum_{j=0}^{n-1} \left| t_{ij} - s'_j \right| \tag{9}$$

(c) $F \neq 0$ の場合，係数列 $\mathbf{T}_i$ を $\mathbf{T}_i = \mathbf{T}_i - \mathbf{S}'_{min}$ として更新する．

(d) 中間係数列に $\mathbf{S}_K = \mathbf{S}'_{min}$ を追加，$K = K + 1$ と更新し，$F = 0$ となるまで (b)～(d) を繰り返す．

(e) 追加した中間係数列を互いに加算し係数列とする．

以上の 5 ステップを全ての入力に対し行うことによって線形変換回路を合成する．

4. 計算時間削減

線形変換回路合成では，係数行列のサイズ・係数の bit 幅が大きくなるに従い，シフト加減算の組み合わせが増えるため，計算時間や最適性に問題が生じる．

そこで本研究では，合成する線形変換回路の行列式を分割し，分割されたそれぞれの線形変換回路を最適合成することにより，計算時間削減を行う方法を提案する．また，分割された線形変換を合成する際，シフト加減算決定の探索に制限をかけ，組み合わせを調べる回数を効率的に減らすことによっても計算時間の削減を行う．

4.1 分割手法

線形変換の行列式を分割することにより，計算時間の削減を行う．分割数を d とし，式 (3) を 2 分割 ($d = 2$) した場合の例を式 (10) に示す．

$$\begin{bmatrix} y_0 \\ \vdots \\ y_i \\ \vdots \\ y_{m-1} \end{bmatrix} = \begin{bmatrix} t_{00} & \cdots & t_{0j} \\ \vdots & & \vdots \\ t_{i0} & \cdots & t_{ij} \\ \vdots & & \vdots \\ t_{m-10} & \cdots & t_{m-1j} \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_j \end{bmatrix} + \begin{bmatrix} t_{0j+1} & \cdots & t_{0n-1} \\ \vdots & & \vdots \\ t_{ij+1} & \cdots & t_{in-1} \\ \vdots & & \vdots \\ t_{m-1j+1} & \cdots & t_{m-1n-1} \end{bmatrix} \begin{bmatrix} x_{j+1} \\ \vdots \\ x_{n-1} \end{bmatrix} \tag{10}$$

分割された行列を互いに加算するため，分割しない場合に比べ加算器数は $m(d-1)$ 個増加するが，分割することによりそれぞれの係数行列が小さくなり，シフト加減算の組み合わせが少なくなるため，計算時間の削減が可能となる．

4.2 探索回数の削減

線形変換回路の合成において，シフト加減算の組み合わせが膨大なため，評価計算に時間がかかる．そこで，シフト加減算の組み合わせを効率よく探索するために，以下の 2 つの手法で探索に制限をかけることにより，計算時間削減を行う．

I. 3.4(b) において $\mathbf{S}'_{min}$ を決定する際，評価値計算の回数を減らすことにより計算時間の削減を行う．
評価値 F が最小となる $\mathbf{S}'_{min}$ を決定する際，それまでの最小評価値 F を F'_{min} とし，評価式の k 項までの部分和 f_k を以下の式で定義する．

$$f_k = \sum_{j=0}^{k} \left| t_{ij} - s'_j \right| \quad (0 \leq k \leq n-1) \tag{11}$$

このとき，

$$f_k > F'_{min} \tag{12}$$

となった時点で評価値計算を終了することにより，計算時間削減が可能である．この場合，$\mathbf{S}'_{min}$ にならない $\mathbf{S}'$ の場合のみ評価値 F を計算しないことになるので，全探索と同等の解が得られる．

II. 3.4(b) において $\mathbf{S}'_{min}$ を決定する際，$\mathbf{S}_{i0}$ に選ばれる対象を制限し探索を行うことによって，計算時間の削減を行う．

ここで，係数行列 $\mathbf{T}_i = [\,t_{i0}\ t_{i1}\ \cdots\ t_{i\,n-1}\,]$ の最大要素の位置 $p(0 \leqq p \leqq n-1)$ を以下の式で定義する．

$$p = MaxElement\{\mathbf{T}_i\} \tag{13}$$

このとき $\mathbf{T}_i$ の最大要素と同じ位置の要素がゼロである $\mathbf{S}$ は，$\mathbf{S}_{i0}$ に選ぶ対象から除外する．

$\mathbf{T}_i$ の最大要素と同じ位置の要素がゼロである中間係数列は，$\mathbf{S}_{i0}$ に選ばれる可能性がとても低いため，組み合わせを調べる回数を効率的に減らすことが可能である．そのため，全探索とほぼ同等の適応度の解が得られ，計算時間の削減も可能となる．

5. シミュレーション

5.1 分割手法における比較

入出力数 16，係数の bit 幅 6bit の場合の，分割手法における評価計算 1 回の時間の比較を表 1 に示す．計算時間は，分割しない場合と比較し，2 分割では 86%，4 分割では 96%程度削減されていることが分かる．

表 1 分割手法における計算時間の比較

分割数	計算時間 [s]
0	1676
2	239.9
4	32.17

5.2 探索の制限方法による比較

入出力数 16，係数の bit 幅 6bit の場合の，探索の制限方法による評価計算 1 回の時間の比較を表 2 に示す．

表 2 制限方法による計算時間の比較

制限方法	計算時間 [s]
全探索	1676
制限 I	593.0
制限 II	404.0
制限 I・II	86.32

全探索の場合と比較して，制限 I の場合は 65%，制限 II の場合は 76%，両方の場合は 95%程度計算時間が削減されていることが分かる．

5.3 最 適 化

SA のパラメータ，線形変換回路のパラメータを以下のように設定し，シミュレーションを行った．

表 3 SA パラメータ

初期集団数	初期解数	初期温度	冷却率
50	10	50	0.9

表 4 線形変換回路パラメータ

入出力数	係数の bit 幅	分割数	制限
16	6	0, 2, 4	I・II

加算器数，プログラムの実行時間の結果を表 5 に示す．

表 5 シミュレーション結果

分割数	実行時間 [min]	加算器数
0	555.9	308.2
2	140.5	315.6
4	32.52	334.8

分割しない場合と比較して，加算器数は 10%以下の増加なのに対し，計算時間は 75%～96%程度に削減されていることが分かる．

6. む す び

本研究では，線形変換回路最適化における問題点を指摘し，それらを解決するための手法を提案した．線形変換回路の係数行列を分割し，合成においてシフト加算減を決定する際の中間係数列の探索に制限をかけ，それぞれを最適合成することで線形変換回路最適化における計算時間削減を行った．

その結果，最適化が困難となるサイズの線形変換回路を分割し，解空間を絞ることで最適化を可能にすることができた．また，シフト加減算の組み合わせを効率良く探索するために，制限をかけた探索を行い，計算時間の削減が可能となった．制限をかけなかった場合と比較し，多少の加算器数の増加が見られたが，計算時間の大幅な削減が可能になった．そのため，提案法は線形変換回路最適化の計算時間の削減に有効であるといえる．

文 献

[1] M.Potkonjak, et al.,"Multiple constant Multiplications: Efficient and Versatile Framework and Algorithms for Exploring Common Subexpression Elimination", IEEE Trans. on CADCAS, vol.15, No.2, pp.151–165, Feb. 1996.

[2] 佐藤 圭介，佐々木 孝雄，豊嶋 久道，"線形変換回路の係数列合成順序を考慮した演算コスト削減アルゴリズム"，電子情報通信学会，技術研究報告，CAS2004–46, pp.25–28, Nov.2004.

[3] 坂和 正敏，田中 雅博，"遺伝的アルゴリズム" 朝倉書店，1995.