Teoria da Complexidade Aditiva para Transformadas

G. Jerônimo da Silva Jr. e R. M. Campello de Souza

***Resumo*— Este artigo introduz a teoria da complexidade aditiva para transformadas, a qual é utilizada para implementar uma transformada com o menor número possível de adições. Aspectos relativos à implementação, tais como o número de passos e o número mínimo de somadores são abordados. Uma aplicação para minimizar a complexidade aditiva da transformada rápida de Fourier é apresentada.**

***Palavras-Chave*— Complexidade aditiva, matrizes bielementares, quantidade de somadores, transformada rápida de Fourier.**

***Abstract*— This article introduces the additive complexity theory over transforms, which is used to implement a transform with the minimum number of additions. Aspects related to the implementation, number of steps and minimum number of adders are discussed. An application aiming at minimizing the additive complexity of the fast Fourier transform is presented.**

***Keywords*— Additive complexity, bielemental matrix, number of adders, fast Fourier transform.**

I. INTRODUÇÃO

QUANDO J. W. Cooley e J. W. Tukey apresentaram, em 1965, a transformada rápida de Fourier (FFT) [1], houve um grande avanço na área de processamento digital de sinais, seguido de uma busca cada vez maior por algoritmos mais eficientes [2]. Em 1978, S. Winograd apresentou uma nova FFT baseada num algoritmo de convolução cíclica [3] e, posteriormente, publicou um livro sobre complexidade aritmética [4]. Esses trabalhos estabeleceram uma maneira de contar o número de multiplicações e adições necessárias para se implementar um determinado algoritmo; grande parte dos esforços, entretanto, se concentrou em minimizar a complexidade multiplicativa. Teoremas sobre a complexidade multiplicativa foram propostos enquanto a complexidade aditiva era obtida a partir do algoritmo final de forma heurística [5]-[8].

A motivação que levou à investigação relatada neste artigo foi a perspectiva de se minimizar a complexidade aditiva de um determinado algoritmo que requer um número estabelecido de multiplicações. Para isso, uma nova forma de se contabilizar a complexidade é introduzida, a qual considera multiplicações triviais independentemente das adições. A originalidade deste artigo, contudo, se defronta com o problema de se obter referências sobre o assunto.

A complexidade aditiva de um algoritmo ou transformada é o número de adições utilizadas pelo mesmo para se obter o resultado desejado. Por exemplo, considere um algoritmo somador dado por

$$X = x_0 + x_1 + x_2 + x_3, \tag{1}$$

em que $x_n \in \mathbb{R}$, $n = 0, 1, 2, 3$, são variáveis de entrada. Em forma matricial

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{2}$$

Neste caso, observa-se que a computação pode ser feita via somador de duas entradas de diversas maneiras, como por exemplo

$$X = \{[(x_0 + x_1) + x_2] + x_3\}, \tag{3}$$

ou

$$X = [(x_0 + x_1) + (x_2 + x_3)]. \tag{4}$$

Observa-se que, não importando a ordem ou arranjo das adições, o número de adições para se computar X não muda e é dado por

$$A_r = \text{Numeros de elementos} - 1. \tag{5}$$

Neste caso, $A_r = 3$ é a complexidade aditiva, entretanto, existe uma diferença sutil entre as duas implementações. A implementação decorrente de (3) apresenta três passos, mas possibilita a implementação com apenas um somador, enquanto que a implementação decorrente de (4) possibilita a computação em dois passos, considerando que $(x_0 + x_1)$ e $(x_2 + x_3)$ podem ser computados em paralelo, contudo, isto demanda dois somadores no processador. Considere agora o seguinte algoritmo com somadores

$$\begin{aligned} y_0 &= x_0 + x_1 + x_2 + x_3 && (6)\\ y_1 &= x_0 + x_3, && (7) \end{aligned}$$

ou, em forma matricial,

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{8}$$

Neste caso, utilizando a Equação (5) para cada variável, obtém-se três adições devido a y_0 e uma adição para y_1, totalizando $A_r = 4$. Entretanto, o algoritmo pode ser implementado por

$$\begin{aligned} a_1 &= x_0 + x_3 && (9)\\ a_2 &= x_1 + x_2 && (10)\\ y_0 &= a_1 + a_2 && (11)\\ y_1 &= a_1, && (12) \end{aligned}$$

com $A_r = 3$. Logo, observa-se que existem maneiras eficientes de se implementar algoritmos de forma a minimizar a complexidade aditiva. Além disso considere, por exemplo, a computação de $y = x + x$, que pode ser efetuada com uma

G. J. da Silva Jr. e R. M. Campello de Souza, Grupo de Processamento de Sinais, Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife, PE, E-mails: gilsonjr@gmail.com, ricardo@ufpe.br. Este trabalho foi parcialmente financiado pelo CNPq (140304/2009-6).

adição ou com uma multiplicação por meio de $y = 2x$. No caso da complexidade multiplicativa, existe uma teoria bem estabelecida que define o que é uma multiplicação [4], mas a literatura não se preocupou em estabelecer uma definição para a adição. A próxima seção introduz os conceitos básicos para a teoria da complexidade aditiva. Na Seção III o problema de implementar transformações racionais é apresentado e aspectos sobre a complexidade aditiva envolvida no processo são discutidos. A Seção IV apresenta a primeira solução para minimizar a complexidade aditiva sobre transformações de uma forma geral, bem como alguns exemplos. As conclusões são apresentadas na Seção V.

II. Conceitos Básicos

As definições apresentadas a seguir representam um breve resumo de alguns conceitos da teoria de complexidade multiplicativa [4], [6].

Definição 1: Todo e qualquer elemento de uma função aritmética que seja uma entrada ou função de uma entrada é chamado de *variável* ou *indeterminado*.

Por exemplo, se um algoritmo computa $y = 2(x_1 + x_2)$, x_1 e $(x_1 + x_2)$ são considerados indeterminados.

Definição 2: Considera-se uma multiplicação trivial quando, entre os elementos envolvidos, existe pelo menos um que pertence ao corpo dos números racionais.

Assim sendo, $y = x + x = 2x$ não é considerado multiplicação ou adição, mas uma multiplicação trivial.

Definição 3: Considera-se uma multiplicação quando os elementos envolvidos são dois indeterminados ou um indeterminado e um elemento que não pertence ao corpo dos números racionais.

Como exemplo, $x_1 x_2$ e $\sqrt{2}x_1$ são consideradas multiplicações. Estas são as principais definições da teoria da complexidade multiplicativa, que agora deve ser complementada para formar as bases de uma teoria de complexidade aditiva.

Definição 4: Os indeterminados x_1 e x_2 são ditos dependentes, se x_1 pode ser escrito na forma $x_1 = qx_2$, $q \in \mathbb{Q}$ e são ditos independentes, em caso contrário.

Definição 5: Considera-se uma adição quando os dois indeterminados envolvidos na operação são independentes.

Essas definições podem ser consideradas como o postulado da teoria da complexidade aritmética, que envolve a complexidade aditiva e multiplicativa. Subtrações são consideradas adições, desde que $x_1 - x_2 = x_1 + (-x_2)$. Estabelecida a definição de adição, segue-se o estudo da complexidade aditiva para transformadas.

III. Complexidade Aditiva em Transformadas

Uma transformada pode ser representada de forma matricial por

$$V = \Psi v, \tag{13}$$

em que

$$V \triangleq \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_{M-1} \end{bmatrix}, \tag{14}$$

$$\Psi \triangleq \begin{bmatrix} \varphi_{0,0} & \varphi_{0,1} & \cdots & \varphi_{0,N-1} \\ \varphi_{1,0} & \varphi_{1,1} & \cdots & \varphi_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{M-1,0} & \varphi_{M-1,1} & \cdots & \varphi_{M-1,N-1} \end{bmatrix} \tag{15}$$

e

$$v \triangleq \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{bmatrix}. \tag{16}$$

Na condição particular em que $\varphi_{i,j} \in \mathbb{Q}$, isto é, a matriz de transformação Ψ contém apenas elementos racionais, a transformada não contém multiplicações. Neste caso, tem-se apenas multiplicações triviais e adições. Sobre essa condição é estudada a complexidade aditiva de transformadas.

Teorema 1: Seja Ψ uma matriz racional, sem linhas nulas, com M linhas, N colunas e com Z elementos nulos. Se $\mathrm{C_A}(\Psi)$ denota a complexidade aditiva de se computar

$$V = \Psi v, \tag{17}$$

através do método direto, então,

$$\mathrm{C_A}(\Psi) = M(N-1) - Z. \tag{18}$$

Demonstração: Utilizando a Equação (5) para a linha i, tem-se

$$A_i = N - 1 - z_i, \tag{19}$$

em que z_i é o número de elementos nulos na linha i. Assim

$$\mathrm{C_A}(\Psi) = \sum_{i=1}^{M}(N - 1 - z_i), \tag{20}$$

$$\mathrm{C_A}(\Psi) = M(N-1) - \sum_{i=1}^{M} z_i, \tag{21}$$

e, desde que $\sum_{i=1}^{M} z_i = Z$, o teorema esta demonstrado. ∎

Teorema 2: Sejam Ψ_a e Ψ_b matrizes racionais. Se $\mathrm{C_A}(\Psi_a\Psi_b)$ denota a complexidade aditiva de se computar a transformação racional $V = \Psi v$, $\Psi = \Psi_a\Psi_b$, através de $v_b = \Psi_b v$ e $V = \Psi_a v_b$, então

$$\mathrm{C_A}(\Psi_a\Psi_b) = \mathrm{C_A}(\Psi_a) + \mathrm{C_A}(\Psi_b). \tag{22}$$

Demonstração: Por definição, $\mathrm{C_A}(\Psi_a\Psi_b)$ é a complexidade aditiva de se computar

$$V = \Psi_a\Psi_b v. \tag{23}$$

Denotando

$$V_b \triangleq \Psi_b v, \tag{24}$$

o qual é computado com $\mathrm{C_A}(\Psi_b)$ adições, então pode-se computar V por

$$V = \Psi_a V_b, \tag{25}$$

com $\mathrm{C_A}(\Psi_a)$ adições, totalizando $\mathrm{C_A}(\Psi_a) + \mathrm{C_A}(\Psi_b)$ adições. ∎

De forma geral, deseja-se obter a menor complexidade aditiva para implementar o produto Ψv. Pode-se então utilizar (18) diretamente ou fatorar a matriz na forma

$$\Psi = \prod_{r=1}^{R} \Psi_r, \tag{26}$$

e então cascatear as implementações dos Ψ_r e obter a complexidade através de

$$\mathrm{C_A}(\prod_{r=1}^{R}\Psi_r) = \sum_{r=1}^{R}\mathrm{C_A}(\Psi_r). \quad (27)$$

Na seção a seguir, essas relações são utilizadas para derivar um método de minimização da complexidade aditiva em transformações racionais e transformadas não racionais.

IV. Implementações com Complexidade Aditiva Mínima

Para implementar transformações racionais de maneira eficiente, no que diz respeito à complexidade aditiva, vamos definir um tipo simples de matriz, na qual estruturas e transformadas mais complexas podem ser obtidas por cascateamento das mesmas.

Definição 6: Define-se como *matriz bielementar* uma matriz, com elementos racionais, em que cada uma das suas linhas deve ter no máximo dois elementos não nulos; além disso, as linhas com dois elementos não nulos, consideradas como vetores no R^N, devem apresentar direções distintas.

Toda matriz racional Ψ pode ser fatorada na forma

$$\Psi = \Psi_1\Phi, \quad (28)$$

em que a matriz Φ é bielementar. Essa fatoração obviamente não é única. Uma maneira possível de se obter uma fatoração dessa forma é escolher pares de colunas de Ψ e, para cada par, escolher vetores em direções distintas que completem todas as direções contidas no par de colunas de Ψ. Com este procedimento, a matriz Φ é bielementar. Escolhendo-se a matriz bielementar Φ, a matriz Ψ_1 pode ser obtida, desde que cada linha i de Ψ é dada por

$$v_i^T = \sum_{j=1}^{m} c_{i,j}u_j^T, \quad (29)$$

em que $c_{i,j}$ são os coeficientes da linha i e coluna j de Ψ_1 e u_j^T é a linha j de Φ. Considerando essa decomposição das linhas de Ψ num determinado par de colunas específico, têm-se que

$$\hat{v}_i^T = \sum_{j=j_1}^{j_2} c_{i,j}u_j^T, \quad (30)$$

em que $\hat{v}_i^T$ é o vetor v_i^T com zeros nas posições não referentes ao par de colunas específico. Desde que todas as direções estão contidas em Φ, o elemento $c_{i,j}$ pode ser obtido por

$$c_{i,j} = \begin{cases} q, & \text{se } \hat{v}_i^T = qu_j^T, \\ 0, & \text{caso contrário,} \end{cases} \quad (31)$$

para $j = j_1, \ldots, j_2$. Fazendo-se isso para todos os pares, é possível obter a matriz Ψ_1.

Exemplo 1: Fatorar a matriz de Hadamard de ordem quatro na forma $\Psi_1\Phi$. Essa matriz é dada por

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Escolhendo como pares de colunas, $(1,2)$ e $(3,4)$, pode-se escolher a matriz Φ por

$$\Phi = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix},$$

e com isso, a matriz Ψ_1 é computada por (31) ou, por simples observação,

$$\Psi_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Observa-se que a implementação da transformação $V = H_4v$, pela forma direta (18), tem complexidade aditiva dada por

$$\mathrm{C_A}(H_4) = 12,$$

enquanto que esta implementação da forma $V = \Psi_1\Phi v$ apresenta

$$\mathrm{C_A}(\Psi_1\Phi) = 8,$$

de modo que a última implementação é mais eficiente que a primeira.

Outra vantagem de se utilizar a fatoração em matrizes bielementares é que a implementação, o número de passos e a quantidade mínima de somadores podem ser derivadas a partir da mesma. O número de passos é dado pelo número de matrizes da fatoração, neste caso dois. A quantidade mínima de somadores é dada pelo número de adições da matriz da fatoração que apresenta maior número de adições, neste caso quatro. A implementação pode ser facilmente obtida das matrizes Φ e Ψ_1; a Figura 1 mostra a implementação da transformada de Hadamard de ordem 4 utilizando a fatoração em matrizes bielementares.

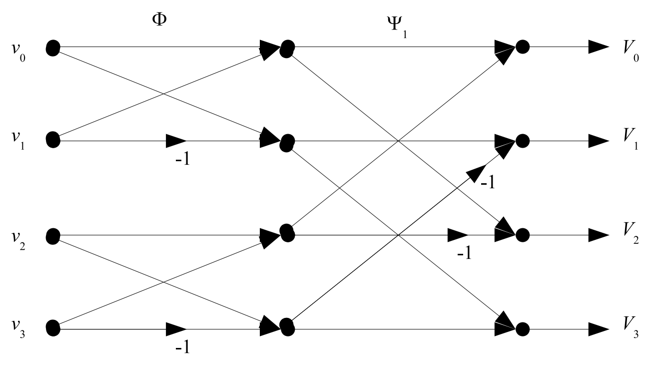

Fig. 1. Implementação da transformada de Hadamard de ordem 4 utilizando a fatoração em matrizes bielementares

Proposição 1: Um método para minimizar a complexidade aditiva da transformação racional $V = \Psi v$, fatorando Ψ em matrizes bielementares:

- Escolher a fatoração $\Psi = \Psi_1\Phi$ que maximize

$$G_0 = \mathrm{C_A}(\Psi) - \mathrm{C_A}(\Psi_1\Phi), \quad (32)$$

para $G_0 > 0$;

- Repetir o item anterior para fatorar $\Psi_i = \Psi_{i+1}\Phi_i$ até que o máximo valor de G_i, obtido por

$$G_i = \mathrm{C_A}(\Psi_i) - \mathrm{C_A}(\Psi_{i+1}\Phi_i), \tag{33}$$

seja igual a zero;
- Implementar Ψ através de

$$\Psi = \Psi_m \Phi_{m-1} \ldots \Phi_2 \Phi_1 \Phi. \tag{34}$$

Sobre esta fatoração, se não existe $G_0 > 0$ que satisfaça o primeiro passo, diz-se então que Ψ é irredutível. A variável G_i pode ser vista como o ganho em complexidade aditiva (o número de adições economizadas) por se implementar Ψ_i através de $\Psi_{i+1}\Phi_i$. Se Ψ_m é bielementar, pode-se dizer que o número de passos do algoritmo é dado por

$$\text{Números de passos} = m + 1, \tag{35}$$

e o número mínimo de somadores necessários para implementar o algoritmo em $m+1$ passos, definindo $\Phi_0 \triangleq \Phi$ e $\Phi_m \triangleq \Psi_m$, é dado por

$$\text{Número de somadores} = \max_i(\mathrm{C_A}(\Phi_i)). \tag{36}$$

Teorema 3: Seja Ψ uma matriz racional a qual foi fatorada em matrizes bielementares, pela Equação (34), utilizando a Proposição 1. Então, a complexidade aditiva da fatoração é dada por

$$\mathrm{C_A}(\Psi_m \Phi_{m-1} \ldots \Phi_1 \Phi) = \mathrm{C_A}(\Psi) - \sum_{i=0}^{m-1} G_i. \tag{37}$$

Demonstração: Pode-se escrever, utilizando (22) em (33), que

$$\begin{aligned}
\mathrm{C_A}(\Psi_m\Phi_{m-1}) &= \mathrm{C_A}(\Psi_{m-1}) - G_{m-1} && (38)\\
\mathrm{C_A}(\Phi_{m-2}) &= \mathrm{C_A}(\Psi_{m-2}) - \mathrm{C_A}(\Psi_{m-1}) - G_{m-2} && (39)\\
\mathrm{C_A}(\Phi_{m-3}) &= \mathrm{C_A}(\Psi_{m-3}) - \mathrm{C_A}(\Psi_{m-2}) - G_{m-3} && (40)\\
&\vdots \\
\mathrm{C_A}(\Phi_1) &= \mathrm{C_A}(\Psi_1) - \mathrm{C_A}(\Psi_2) - G_1 && (41)\\
\mathrm{C_A}(\Phi) &= \mathrm{C_A}(\Psi) - \mathrm{C_A}(\Psi_1) - G_0. && (42)
\end{aligned}$$

Somando-se membro a membro estas expressões resulta, no primeiro membro, a complexidade aditiva da fatoração. No segundo membro, todas os termos $\mathrm{C_A}(\Psi_i)$ se cancelam, restando apenas $\mathrm{C_A}(\Psi)$ menos o somatório de G_i. ■

Isso justifica que a melhor implementação para $V = \Psi v$, dado que Ψ contém elementos racionais, é obtida através da Proposição 1. Entretanto, quando a matriz Ψ torna-se suficientemente grande, o problema de maximizar G_i pode-se tornar inviável. Esse problema se resume em escolher os melhores pares de colunas para Ψ.

Exemplo 2: Fatorar a matriz

$$\Psi = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 1 & -1 & -1 \end{bmatrix},$$

como produto de matrizes bielementares. O primeiro passo seria escolher os pares de colunas para efetuar a primeira fatoração, $\Psi = \Psi_1\Phi$. Observa-se que os zeros das colunas 2 e 5 são bem correlacionados, assim como os das colunas 3 e 4. Uma boa correlação de zeros é um indicativo de que estas colunas são uma boa escolha para a fatoração. Assim, os pares de colunas escolhidos são $(2,5)$ e $(3,4)$, a coluna 1 fica isolada. Escolhe-se então

$$\Phi = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

e computa-se Ψ_1 através de (31) ou por simples observação, o que resulta em

$$\Psi_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Desde que Ψ_1 é bielementar a fatoração está concluída.

Exemplo 3: Fatorar a matriz

$$\Psi = \begin{bmatrix} 1 & 1 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 & 1 \\ 0 & -\frac{1}{2} & 1 & -1 \end{bmatrix},$$

como produto de matrizes bielementares. Neste caso, independente dos pares de colunas escolhidos, obtém-se sempre 3 vetores bidimensionais com direções diferentes. Escolhendo-se

$$\Phi = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

existem varias soluções para Ψ_1; escolhe-se a solução (31), a mais apropriada, ou seja,

$$\Psi_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -\frac{1}{2} & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 1 & 0 \end{bmatrix}.$$

Observa-se que $G_0 = 0$, entretanto a matriz ainda pode ser fatorada em matrizes bielementares.

Na maioria das transformações práticas, as matrizes de transformações não são racionais; neste caso, uma implementação ideal preocupa-se em minimizar o número de multiplicações da transformação. Neste contexto, existem algoritmos que implementam uma transformação qualquer Ψ através da fatoração

$$\Psi = CBA, \tag{43}$$

em que C e A são matrizes com elementos racionais e a matriz B é uma matriz diagonal [5]. Essa fatoração é feita de modo a minimizar o número de multiplicações, entretanto, A e C são transformações racionais. Portanto, para minimizar o número de adições sobre esses algoritmos, utiliza-se a Proposição 1 para fatorar as matrizes A e C em matrizes bielementares.

Exemplo 4: Considere a matriz $W = CBA$ da transformada rápida de Fourier (FFT) para $N = 3$, fatorada pelo algoritmo de decomposição do núcleo em bases ciclotômicas [8],

$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & j \\ 0 & 1 & -j \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \gamma_2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 1 & -1 \end{bmatrix},$$

em que $\gamma_2 = -\text{sen}(2\pi/3)$. Iniciando pela matriz A, a qual pode ser fatorada, e escolhendo-se o par de colunas $(2, 3)$, tem-se

$$A = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & 0 \end{bmatrix},$$

isto é, $A = A_1\Phi$. A matriz C, após separada em parte real e parte imaginária, não contém adições. Portanto, a implementação da FFT através de

$$V = CBA_1\Phi v,$$

contém 1 multiplicação, 1 multiplicação trivial e 4 adições. Para comparação, a FFT de Winograd implementa o mesmo algoritmo com 2 multiplicações e 4 adições.

Exemplo 5: Existe uma FFT otimizada, em relação à complexidade multiplicativa, para $N = 5$, que faz uso da fatoração $W = CBA$, com apenas quatro multiplicações. Para determinar a complexidade aditiva desse algoritmo, dado que A é racional, B é diagonal e C é racional complexa, basta avaliar a complexidade aditiva de A e das partes real e imaginária de C. A matriz A é dada por

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 1 & -1 & -1 \end{bmatrix},$$

a qual já foi fatorada no Exemplo 2 e contém 6 adições. A parte real da matriz C é dada por

$$C_r = \begin{bmatrix} 1 & 1 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 & 1 \\ 0 & -\frac{1}{2} & 1 & -1 \\ 0 & -\frac{1}{2} & 1 & -1 \\ -\frac{1}{2} & 0 & 1 & 1 \end{bmatrix},$$

a qual pode ser expressa por

$$C_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \Psi,$$

em que a matriz Ψ é a matriz do Exemplo 3 e, portanto, apresenta 6 adições. A parte imaginária da matriz C é dada por

$$C_i = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & -1 & -1 \\ -1 & 0 & -1 \end{bmatrix},$$

a qual pode ser decomposta em

$$C_i = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix},$$

com complexidade aditiva igual a 2. Portanto, a complexidade aditiva da FFT otimizada de comprimento cinco é 14. Para comparação, a FFT de Winograd implementa o algoritmo com 5 multiplicações e 13 adições.

V. CONCLUSÕES

Uma nova teoria sobre a complexidade aditiva foi introduzida com o objetivo de implementar transformadas com o menor número de multiplicações e adições possível. Um método para minimizar a complexidade aditiva, baseado na fatoração em matrizes bielementares, foi introduzido e aplicado em transformações lineares. Exemplos da aplicação deste método foram apresentados, em que obtêm-se ganhos de complexidade aditiva quando comparado com o método direto. Além disso, para transformações que contêm apenas os elementos 0, 1 e -1, com N colunas, a complexidade aditiva se reduz para $N \log_2 N$, como no caso das transformadas de Hadamard. Exemplos da implementação de algoritmos FFT com complexidade aditiva mínima também foram apresentados.

AGRADECIMENTOS

Os autores agradecem ao Prof. Dr. Hélio M. de Oliveira por suas valorosas contribuições ao presente trabalho. O primeiro autor agradece ao CNPq pelo apoio recebido durante a realização deste trabalho.

REFERÊNCIAS

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965. [Online]. Available: http://www.jstor.org/stable/2003354

[2] M. Heideman, D. Johnson, and C. Burrus, "Gauss and the history of the fast Fourier transform," *ASSP Magazine, IEEE*, vol. 1, no. 4, pp. 14–21, Oct 1984.

[3] S. Winograd, "On computing the discrete Fourier transform," *Mathematics of Computation*, vol. 32, pp. 175–199, Jan. 1978.

[4] ——, *Arithmetic Complexity of Computations*. SIAM Publications, 1980.

[5] R. E. Blahut, *Fast Algorithms for Signal Processing*, 2nd ed. Cambridge University Press, 2010.

[6] M. T. Heideman, *Multiplicative Complexity, Convolution, and the DFT*, 2nd ed. Springer-Verlag, 1988.

[7] G. J. da Silva Jr., R. M. Campello de Souza, and H. M. de Oliveira, "New algorithms for computing a single component of the discrete Fourier transform," *International Symposium on Communication Theory and Applications, ISCTA*, vol. 10, pp. 151–154, Jul. 2009.

[8] G. J. da Silva Jr. and R. M. Campello de Souza, "Cyclotomic basis for computing the discrete Fourier transform," *International Telecommunications Symposium, ITS*, vol. 7, pp. 1–5, September 2010.