Journal of Computational Mathematics, Vol.13, No.4, 1995, 351–356.

THE MULTIPLICATIVE COMPLEXITY AND ALGORITHM OF THE GENERALIZED DISCRETE FOURIER TRANSFORM(GFT)*

Y.H. Zeng
(*7th Department, National University of Defence Technology, Changsha, China*)

Abstract

In this paper, we have proved that the lower bound of the number of real multiplications for computing a length 2^t real GFT(a,b) ($a=\pm 1/2, b=0$ *or* $b=\pm 1/2, a=0$) is $2^{t+1}-2t-2$ and that for computing a length 2^t real GFT(a,b)($a=\pm 1/2, b=\pm 1/2$) is $2^{t+1}-2$. Practical algorithms which meet the lower bounds of multiplications are given.

1. Introduction

Since the fast Fourier transform was proposed, great interests for fast algorithms have been aroused. In this area, there have been many achievements, which have greatly stimulated the development of digital signal processing and other fields. The computational complexity is to study what the best algorithm will be for a given problem. There are many standards for appraising whether an algorithm is good or bad. In numerical computation, a common standard is the number of multiplications, that is, we say an algorithm is good or bad if the number of multiplications is large or small. The famous mathematican S. Winograd and L. Auslander have done some pioneering works in this area. They found the lower bound of the number of multiplications for multiplying two polynomials, and also gave an algorithm which met the lower bound[1]. Some later,they found the lower bound of the number of mulitplications for computing the discrete Fourier transform (DFT)[2–3]. After that, Heidemann-Burrus and Duhamel et al. also studied the multiplicative complexity of DFT. Heidemann-Burrus pointed out that $2^{t+1}-t^2-t-2$ multiplications is necessary for computing a length-2^t DFT, and also gave a practical algorithm which met the bound[4]. Some later, Duhamel et al. also proved the assertion and gave a new algorithm[5].

Generalized discrete Fourier transform (GFT) is a generalization of DFT (In ref.[10] the transform is called DET). It is shown that GFT is better than DFT in some applications. Let $x(n)(n=0,1,\cdots,N-1)$ be a real number sequence, we call

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{(n+a)(k+b)}, k=0,1,\cdots,N-1. \tag{1}$$

* Received April 26, 1994.

the generalized discrete Fourier transform of $\{x(n)\}$, where $W_N = e^{-i\frac{2\pi}{N}}$ and $i = \sqrt{-1}$. In (1) a is called the time parameter and b the frequency parameter. A GFT with time parameter a and frequency parameter b is denoted by GFT(a,b). Especially, if $a = b = 0$, (1) is the DFT. It is very interesting to determine the multiplicative complexity of GFT. Since in practical applications a,b can either be 0 or $\pm 1/2$, so when we discuss the multiplicative complexity, we confine our research on these cases.

2. The Computation of GFT(a,b)(a,b are integers)

If a,b are integers, the compution of GFT(a,b) is almost the same as that of DFT. In fact, if we set

$$x^{'}(n) = \begin{cases} x(N+n-a), & n = 0, 1, \cdots, a-1 \\ x(n-a), & n = a, \cdots, N-1 \end{cases}$$

and denote the DFT of $\{x^{'}(n)\}$ by $\{X^{'}(k)\}$, then it is easy to prove that

$$X(k-b) = X^{'}(k), \quad k \in Z$$

where $\{X(k)\}$ means the GFT(a,b) of $\{x(n)\}$. Therefore, the multiplicative complexity of GFT(a,b)(a,b are integers) is the same as that of DFT.

3. The Multiplicative Complexity and Algorithm of GFT(0,1/2) and GFT(1/2,0)

1. The relationship between DFT and GFT(1/2,0)

Let $\{Y(k)\}$ be the DFT of $\{x(n)\}(n = 0, 1, \cdots, N-1; N = 2^t)$, that is

$$Y(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, k = 0, 1, \cdots, N-1.$$

In the following, $\{Y(k)\}$ is turned to a series of GFT(1/2,0).

$$Y(k) = \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{nk} + \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{(n+1/2)k} \ .$$

If we set $\{U(k)\}$ and $\{V(k)\}$ to be the DFT of $\{x(2n)\}(n = 0, 1, \cdots, N/2-1)$ and the GFT(1/2,0) of $\{x(2n+1)\}(n = 0, 1, \cdots, N/2-1)$ respectively, then

$$Y(k) = U(k) + V(k), k = 0, 1, \cdots, N/2-1,$$

$$Y(k+N/2) = U(k) - V(k), k = 0, 1, \cdots, N/2-1.$$

Therefore, a DFT of length N is decomposed to a DFT of length N/2 and a GFT(1/2,0) of length N/2 plus N additions. If $N/2 \geq 2$, the decomposition can continue. In general, a DFT of length 2^t can be decomposed to one GFT(1/2,0) of length 2^{t-1}, one GFT(1/2,0) of length 2^{t-2}, $\cdots$, one GFT(1/2,0) of length 2 plus one DFT of length 2.

2. The multiplicative complexity and algorithm of GFT(1/2,0)

In [9] we have proved that a GFT(1/2,0) of length N can be turned to a DCT-II of length N/2 and a DST-II of length N/2 plus some additions. We will not give the details here. In [11] we proved that a DCT-II or a DST-II of length 2^{t-1} can be computed using $2^t - t - 1$ multiplications and also gave an algorithm which met this bound. Therefore, we get an algorithm which uses $2(2^t - t - 1) = 2^{t+1} - 2t - 2$ real multiplications to compute a GFT(1/2,0) of length 2^t.

Theorem 1. *At least $2^{t+1} - 2t - 2$ multiplications must be used for computing a GFT(1/2,0) of length 2^t, and there exists an algorithm whose number of multiplications meets this bound.*

Proof. The algorithm discussd above meets the bound. So, we need only to show that $2^{t+1} - 2t - 2$ multiplications is necessary for any algorithm which computes a GFT(1/2,0) of length 2^t.

If t=0 or 1, the conclusion is obviously right.

Now, assume $t \geq 2$. We have shown that a DFT of length 2^{t+1} can be decomposed to a DFT of 2^t and a GFT(1/2,0) of length 2^t. Assume that we have an algorithm to compute a GFT(1/2,0) of length 2^t which costs $M_1(2^t)$ multiplications. Using this algorithm combined with the algorithm which computes a DFT of length 2^t with $2^{t+1} - t^2 - t - 2$ multiplications, we get an algorithm to compute the DFT of length 2^{t+1} with $M_1(2^t) + 2^{t+1} - t^2 - t - 2$ multiplications. But it has been proved that $2^{t+2} - (t+1)^2 - (t+1) - 2$ multiplications is necessary for computing a DFT of length 2^{t+1}, we see that

$$M_1(2^t) + 2^{t+1} - t^2 - t - 2 \geq 2^{t+2} - (t+1)^2 - (t+1) - 2$$

This leads to

$$M_1(2^t) \geq 2^{t+1} - 2t - 2.$$

3. The relationship between DFT and GFT(0,1/2)

Let $\{Y(k)\}(k = 0, 1, \cdots, N-1; N = 2^t)$ be the DFT of a real number sequence $\{x(n)\}(n = 0, 1, \cdots, N-1)$. Then

$$Y(2k) = \sum_{n=0}^{N/2-1} (x(n) + x(n+N/2))W_{N/2}^{nk},$$

$$Y(2k+1) = \sum_{n=0}^{N/2-1} (x(n) - x(n+N/2)W_{N/2}^{n(k+1/2)},$$

$$k = 0, 1, \cdots, N/2 - 1.$$

This tells us that a DFT of length 2^t can be computed through computing a DFT of length 2^{t-1} and a GFT(0,1/2) of length 2^{t-1} plus some additions. The DFT of length 2^{t-1} can be computed in the same way. Therefore, a DFT of length 2^t can be decomposed to a GFT(0,1/2) of length 2^{t-1}, a GFT(0,1/2) of length 2^{t-2}, $\cdots$, a GFT(0,1/2) of length 2 and a DFT of length 2. In other words, a DFT can be computed by a series of GFT(0,1/2).

4. The multiplicative complexity and algorithm of GFT(0,1/2)

First, in [9] an algorithm was given for computing GFT(0,1/2) by IDCT-II and IDST-II. The algorithm uses a IDCT-II of length 2^{t-1} to compute a GFT(0,1/2) of length 2^t plus some additions. In [11] we gave an algorithm for IDCT-II or IDST-II of length 2^{t-1} which costs $2^t - t - 1$ multiplications. So, we get an algorithm that computes a GFT(0,1/2) with $2(2^t - t - 1) = 2^{t+1} - 2t - 2$ real multiplications.

Theorem 2. *At least $2^{t+1} - 2t - 2$ multiplications must be used for computing a real GFT(0,1/2) of length 2^t, and there exists an algorithm which meets the bound.*

Proof. The proof for this theorem is the same as theorem 1, so we simply omit the detail.

4. The Multiplicative Complexity and Algorithm of GFT(1/2,1/2)

Let $\{X(k)\}$ denote the GFT(1/2,1/2) of a real number sequence $\{x(n)\}$ $(n = 0, 1, \cdots, N-1; N = 2^t)$, that is

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{(n+1/2)(k+1/2)}, k = 0, 1, \cdots, N-1.$$

In the following, DCT-II is used for computing the GFT(1/2,1/2). Let U(k) and V(k) be defined by

$$U(k) = \sum_{n=0}^{N-1} x(n) \cos \frac{\pi(2n+1)(2k+1)}{2N}, k = 0, 1, \cdots, N-1.$$

$$V(k) = \sum_{n=0}^{N-1} x(n) \sin \frac{\pi(2n+1)(2k+1)}{2N}, k = 0, 1, \cdots, N-1.$$

Then we see that

$$X(k) = U(k) - iV(k).$$

Noticing that $U(N-1-k) = -U(k)$, we know that it is sufficient to get $\{U(k)\}$ from $U(0), \cdots, U(N/2-1)$. If $N \geq 4$, then the method in [6] can be used to compute {U(k)} which needs the computation of a skew-cyclic convolution (SCC) of length N/2 plus some additions. $\{V(k)\}$ can be turned to a similar forms as $\{U(k)\}$, in fact

$$V(k+N/2) = \sum_{n=0}^{N-1} (-1)^n x(n) \cos \frac{\pi(2n+1)(2k+1)}{2N}, k = 0, 1, \cdots, N/2-1.$$

Futhermore, $V(N-1-k) = V(k)$. Hence, if $N \geq 4$, $\{V(k)\}$ can also be computed by a SCC of length N/2.

In general, a GFT(1/2,1/2) of length N ($N \geq 4$) can be computed by two SCC of length N/2 plus some additions. If the Winograd algorithm for multiplying two polynomials is used, then the computation of SCC of length $N/2$ needs $N-1$ real multiplications. Therefore, only $2N-2$ real multiplications is necessary for computing the GFT(1/2,1/2). It is obvious that 2 real multiplications is necessary and sufficient for the GFT(1/2,1/2) of length 2.

Theorem 3. *At least $2^{t+1}-2$ real multiplications must be used for computing a GFT(1/2,1/2) of length 2^t, and there exists an algorithm which meets the bound.*

Proof. First, the algorithm discussed above meets the bound.

Second, we want to show that $2^{t+1}-2$ multiplications is necessary.

Let $\{X(k)\}$ be the GFT(0,1/2) of $\{x(n)\}(n=0,1,\cdots,M-1; M=2^{t+1})$. $\{X(k)\}$ can be decomposed to a GFT(0,1/2) of length $N=2^t$ and a GFT(1/2,1/2) of length 2^t. In fact,

$$X(k)=\sum_{n=0}^{N-1} x(2n)W_N^{n(k+1/2)}+\sum_{n=0}^{N-1} x(2n+1)W_N^{(n+1/2)(k+1/2)}, \quad k=0,1,\cdots,N-1. \quad (2)$$

and $X(k+N)=X^*(k)$(* means the conjugate). We see that the first sum of (2) is a GFT(0,1/2) of length N and the second sum is a GFT(1/2,1/2) of length N. This tells us that a GFT(0,1/2) of length 2^{t+1} can be computed by a GFT(0,1/2) of length 2^t and a GFT(1/2,1/2) of length 2^t. If we have an algorithm to compute a GFT(1/2,1/2) of length 2^t with $M_2(2^t)$ multiplications, then combined with the algorithm in section 3.4 for computing the GFT(0,1/2) of length 2^t we get an algorithm which computes the GFT(0,1/2) of length 2^{t+1} with $M_2(2^t)+2^{t+1}-2t-2$ multiplications. But it is shown in section 3.4 that $2^{t+2}-2(t+1)-2$ multiplications is necessary for the GFT(0,1/2) of length 2^{t+1}, hence we get

$$M_2(2^t)+2^{t+1}-2t-2 \geq 2^{t+2}-2(t+1)-2.$$

This leads to

$$M_2(2^t) \geq 2^{t+1}-2.$$

5. Other Kinds of GFT

Any GFT(a,b) with $a,b \in \{0,-1/2,1/2\}$ can be turned to a GFT(a,b) with $a,b \in \{0,1/2\}$. Reference [8-9] for the details. More accurately, a GFT(-1/2,b) can be turned to a GFT(1/2,b) and a GFT(a,-1/2) can be turned to GFT(a,1/2) without any multiplications. Therefore, we have the following theorem.

Theorem 4. *For a real GFT(a,b) of length 2^t*

(i) If $a=b=0$, then $2^{t+1}-t^2-t-2$ real multiplications is necessary;

(ii) If $a=\pm 1/2, b=0$, or $b=\pm 1/2, a=0$, then $2^{t+1}-2t-2$ real multiplications is necessary;

(iii) If $a = \pm 1/2, b = \pm 1/2$, then $2^{t+1} - 2$ real multiplications is necessary.

There are algorithms which meet the bounds in all the cases respectively.

The above results can be extended to complex GFT. In [4-5] it was shown that a complex DFT needs at least $2^{t+2} - 2t^2 - 2t - 4$ real multiplications, exactly two times that of a real DFT. Also, we can show that the lower bound of the multiplicative complexity of a complex GFT is exactly two times that of a real GFT. We will not give the details here. Algorithms for complex GFT$'s$ which meet the lower bounds can be construsted in a similar way.

Recently a new kind of discrete orthogonal transform called DWT[12] is used in signal processing. From the relationship between DWT and GFT[13,14] it is easy to show that theorem 4 is correct if GFT(a,b) is replaced by DWT(a,b).

References

[1] S. Winograd, Some bilinear forms whose multiplitative complexity depends on the field of constants, *Math. Syst. Theorem*, 10 (1977), 169–180.

[2] S. Winograd, On the multiplicative complexity of the discrete Fourier transform, *Adv. Math.*, 32 (1979), 83–117.

[3] L. Auslander and S. Winograd, The multiplicative complexity of the discrete Fourier transform, *Adv. Applied Math.*, 5 (1984), 87–109.

[4] M.T. Heidemann and C.S. Burrus, On the number of multiplications necessary to compute a length-2^n DFT, *IEEE Trans. ASSP*, 34 (1986), 91–95.

[5] P. Duhamel, Algorithms meeting the lower bounds on the multiplicative complexity of length-2^n DFT's and their connection with practical algorithm, *IEEE Trans. SP*, 38 (1990),1504–1511.

[6] W.P. Li, A new algorithm to compute the DCT and its inverse, *IEEE Trans. SP*, 39 (1991), 1305–1313.

[7] M.S. Nian and Y.H. Mao, Using the generalized discrete Fourier transform to compute linear convolution and inverse convolution, *Acta Electronica Sinica (China)*, (1988).

[8] Y.H. Zeng et al., New algorithm for GFT, skew-cyclic convolution and cyclic convolution, *Proceedings of ICSP'90*, Beijing, 151–152.

[9] Y.H. Zeng and Z.R. Jiang, Parallel algorithm for GFT, skew-cyclic convolution and cyclic convolution, *J. Numer. Math. Comput. Appl.*, (China), 14 (1993), 28-37.

[10] G.L. Zhang and A.L. Pan, Theory and Applications of Digital Spectral Techniques, National Defense Technology Press, China, 1992.

[11] Y.H. Zeng, The multiplicative complexity and algorithm for DCT, DST, DHT and DFT, *J. National University of Defense Technology*, (to appear)

[12] Z. Wang and B.R. Hunt, The discrete W transform, *Appl. Math. Comput.*, 16 (1985), 19–48.

[13] Y.H. Zeng, A parallel algorithm for DWT and GFT, *Proceedings of the IEEE TENCON'93*, Beijing, Vol 3, 361–364.

[14] Z.R. Jiang, Y.H. Zeng and P.N. Yu, Fast Algorithms, National University of Defense Technology Press, Changsha, China,1994.