2023 年中国研究生数学建模竞赛 B 题（华为题目）

DFT 类矩阵的整数分解逼近

一、问题背景

离散傅里叶变换（Discrete Fourier Transform，DFT）作为一种基本工具广泛应用于工程、科学以及数学领域。例如，通信信号处理中，常用 DFT 实现信号的正交频分复用（Orthogonal Frequency Division Multiplexing，OFDM）系统的时频域变换（见图 1）。另外在信道估计中，也需要用到逆 DFT（IDFT）和 DFT 以便对信道估计结果进行时域降噪（见图 2）。

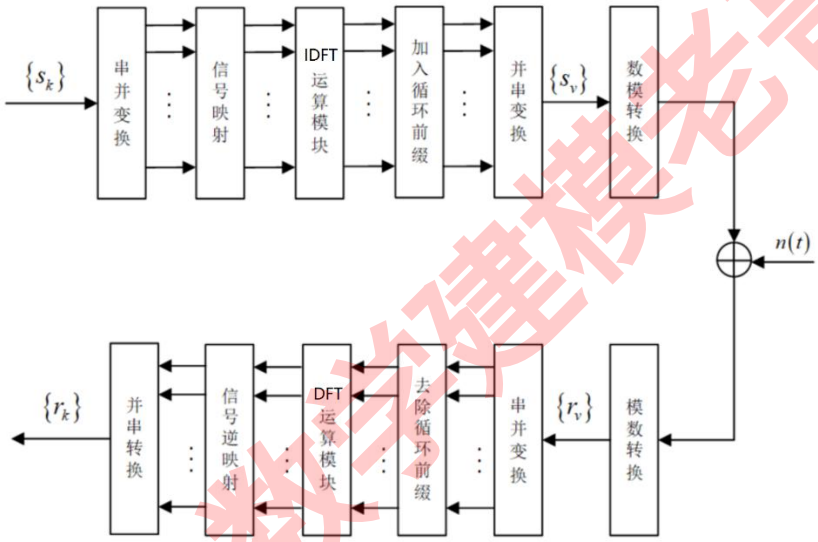

图 1 OFDM 系统流程

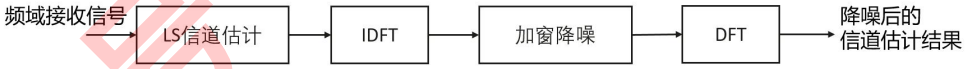

图 2 信道估计处理流程

在芯片设计中，DFT 计算的硬件复杂度与其算法复杂度和数据元素取值范围相关。算法复杂度越高、数据取值范围越大，其硬件复杂度就越大。目前在实际产品中，一般采用快速傅里叶变换（Fast Fourier Transform，FFT）算法来快速实现 DFT，其利用 DFT 变换的各种性质，可以大幅降低 DFT 的计算复杂度（参见[1][2]）。然而，随着无线通信技术的演进，天线阵面越来越大，通道数越来越多，通信带宽越来越大，对 FFT 的需求也越来越大，从而导致专用芯片上实现 FFT 的硬件开销也越大。为进一步降低芯片资源开销，一种可行的思路是将 DFT 矩阵分解成整数矩阵连乘的形式。

给定N点的时域一维复数信号$x_0, x_1, \ldots, x_{N-1}$，DFT 后得到的复数信号$\mathrm{X}_k$（$k = 0,1,\ldots,N-1$）由下式给出（其中 j 为虚数单位，下同）：

$$\mathrm{X}_k = \sum_{n=0}^{N-1} x_n * e^{-\frac{j2\pi nk}{N}} \quad ,k = 0,1,2,\ldots,N-1 \#(1)$$

写成矩阵形式为：

$$\mathbf{X} = \mathbf{F}_N \mathbf{x} \#(2)$$

其中 $\mathbf{x}= [x_0 \ x_1 \cdots \ x_{N-1}]^{\mathrm{T}}$为时域信号向量，$\mathbf{X} = [\mathrm{X}_0 \ \mathrm{X}_1 \cdots \ \mathrm{X}_{N-1}]^{\mathrm{T}}$为变换后的频域信号向量，$\mathbf{F}_N$为 DFT 矩阵，形式如下：

$$\mathbf{F}_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{N-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \cdots & w^{(N-1)(N-1)} \end{bmatrix}, w = \mathrm{e}^{-\frac{j2\pi}{N}} \#(3)$$

由于 DFT 矩阵的特殊结构，存在很多方法加速傅里叶变换的计算，其中，分治的策略以及蝶形计算单元的优化是 DFT 性能的关键。下面分别给出用 FFT 和矩阵连乘拟合近似计算 DFT 的具体思路。

FFT 思路：FFT 采用蝶形运算的思想，以 radix-3 蝶形计算为例，其计算过程可以表示为：

$$\begin{bmatrix} \mathrm{X}_0 \\ \mathrm{X}_1 \\ \mathrm{X}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -3 & -1 \\ 1 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & \sqrt{3}j/2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \#(4)$$

可以看到蝶形的设计相对于直接 DFT 矩阵乘积的形式大幅降低了复数乘法运算的次数。

矩阵连乘拟合思路：DFT 可以用传统的蝶形计算方法精确实现，也可以用一种矩阵乘法拟合近似获得，其核心思想是将 DFT 矩阵近似表达为一连串稀疏的、元素取值有限的矩阵连乘形式。以 radix-8 蝶形计算为例（参见[3]）：

$$\mathbf{F}_8 \approx \mathbf{P}\mathbf{A}_4\mathbf{D}\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1 \#(5)$$

其中$\mathbf{P} = [e_0 \ e_4 \ e_2 \ e_5 \ e_1 \ e_7 \ e_3 \ e_6]$为排列矩阵，$\mathbf{D} = \mathrm{diag}([1 \ 1 \ 1 \ j \ 1 \ j \ j \ 1])$为对角阵，$\mathbf{A}_1$~$\mathbf{A}_4$为稀疏矩阵，分别如下：

$$\mathbf{A}_1 = \begin{bmatrix} 1 & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & 1 & & & & 1 & \\ & & & 1 & & & & 1 \\ 1 & & & & -1 & & & \\ & 1 & & & & -1 & & \\ & & 1 & & & & -1 & \\ & & & 1 & & & & -1 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 1 & & 1 & & & & & \\ & 1 & & 1 & & & & \\ 1 & & -1 & & & & & \\ & 1 & & -1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & 1 \\ & & & & & & 1 & \\ & & & & & 1 & & -1 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & -1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & 1 \\ & & & & & 1 & 1 & \\ & & & & & 1 & -1 & \\ & & & & 1 & & & -1 \end{bmatrix}, \mathbf{A}_4 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & -1 & & & & \\ & & 1 & 1 & & & & \\ & & & & 1 & -1 & & \\ & & & & 1 & 1 & & \\ & & & & & & -1 & 1 \\ & & & & & & 1 & 1 \end{bmatrix}$$

可以看到在该方案中，分解后的矩阵元素均为整数，从而降低了每个乘法器的复杂度；另外$\mathbf{A}_1$~$\mathbf{A}_4$的稀疏特性可以减少乘法运算数量。可以看出，这其实是一种精度与硬件复杂度的折中方案，即损失了一定的计算精度，但是大幅降低了硬件复杂度。在对输出信噪比要求不高的情况下可以优先考虑此类方案。

二、建模描述

本题在不同约束条件下，研究 DFT 的低复杂度计算方案，目的是对目前芯片中利用 FFT 计算 DFT 的方法进行替代，以降低硬件复杂度。给定已知的 N 维 DFT 矩阵$\mathbf{F}_N$，设计 K 个矩阵$\mathcal{A}=\{\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_K\}$，使得矩阵$\beta\mathbf{F}_N$和$\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_K$在 Frobenius 范数意义下尽可能接近，即：

$$\min_{\mathcal{A},\beta}\mathrm{RMSE}(\mathcal{A},\beta)=\frac{1}{N}\sqrt{\|\beta\mathbf{F}_N-\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_K\|_F^{\,2}}\#(6)$$

其中β为实值矩阵缩放因子，可根据约束条件不同来设计。

相比于乘法，加法的硬件复杂度小得多，因此本题中只考虑乘法器的硬件复杂度：

$$C=q\times L$$

其中，q 指示分解后的矩阵$\mathbf{A}_k$中元素的取值范围。在以下的问题 2~5 中，我们限制$\mathbf{A}_k$中元素实部和虚部的取值范围为$\mathcal{P}=\{0,\pm1,\pm2,\ldots,\pm2^{q-1}\}$。以$\mathcal{P}=\{0,\pm1,\pm2,\pm4\}$为例，此时$q=3$。$L$ 表示复数乘法的次数，其中与 0、±1、$\pm j$或$(\pm1\pm j)$相乘时不计入复数乘法次数。例如：若$\mathcal{P}=\{0,\pm1,\pm2,\pm4\}$，则下列矩阵乘法的硬件复杂度$C=6$（$q=3$，$L=2$）：

$$\begin{bmatrix}1 & 2+4j\\ 1+2j & 0\end{bmatrix}\times\begin{bmatrix}0 & 1\\ 2+4j & 2-4j\end{bmatrix}$$

考虑以下两种约束条件：

约束 1：限定$\mathcal{A}$中每个矩阵$\mathbf{A}_k$的每行至多只有 2 个非零元素。

约束 2：限定$\mathcal{A}$中每个矩阵$\mathbf{A}_k$满足以下要求：

$$\mathbf{A}_k[l,m]\in\{x+jy|x,y\in\mathcal{P}\},\mathcal{P}=\{0,\pm1,\pm2,\ldots,\pm2^{q-1}\},k=1,2,\ldots,K;\ l,m=1,2,\ldots,N$$

其中，$\mathbf{A}_k[l,m]$表示矩阵$\mathbf{A}_k$第l行第m列的元素。在问题 2,3,4 中，固定$q=3$；在问题 5 中，需要寻找合适的q以满足精度要求，并且使得硬件复杂度 C 尽量低。

目前使用 FFT 进行 DFT 计算的方案硬件复杂度较高，因为我们希望研究一种替代方案来降低 DFT 计算的硬件复杂度，但同时我们对精度也有一定要求。请针对以下问题分别设计分解方法，既能最小化 RMSE，同时又使得乘法器的数量尽量少。

$\mathcal{A}$中矩阵的个数 K 的取值并没有限制，也是优化的变量之一。但需要注意，一般情况下，K 越小，硬件复杂度越低，但是如果增加矩阵的个数可以使得矩阵中包含更多的简单元素（0、±1、$\pm j$或$(\pm1\pm j)$），硬件复杂度也可能会降低，因此，需要根据硬件复杂度 C 的定义合理的设计 K。

问题 1：首先通过减少乘法器个数来降低硬件复杂度。由于仅在非零元素相乘时需要使用乘法器，若$\mathbf{A}_k$矩阵中大部分元素均为 0，则可减少乘法器的个数，因此希望$\mathbf{A}_k$为稀疏矩阵。对于$N=2^t,t=1,2,3,\ldots$的 DFT 矩阵$\mathbf{F}_N$，请在满足**约束 1** 的条件下，对最优化问题(6)中的变量$\mathcal{A}$和β进行优化，并计算最小误差（ 即(6)的目标函数，下同）和方案的硬件复杂度C（由于本题中没有限定$\mathbf{A}_k$元素的取值范围，因此在计算硬件复杂度时可默认$q=16$）。

问题 2：讨论通过限制$\mathbf{A}_k$中元素实部和虚部取值范围的方式来减少硬件复杂度的方案。对于$N=2^t, t=1,2,3,4,5$的 DFT 矩阵$\mathbf{F}_N$，请在满足**约束 2** 的条件下，对$\mathcal{A}$和β进行优化，并计算最小误差和方案的硬件复杂度C。

问题 3：同时限制$\mathbf{A}_k$的稀疏性和取值范围。对于$N=2^t, t=1,2,3,4,5$的 DFT 矩阵$\mathbf{F}_N$，请在**同时满足约束 1 和 2** 的条件下，对$\mathcal{A}$和β进行优化，并计算最小误差和方案的硬件复杂度C。

问题 4：进一步研究对其它矩阵的分解方案。考虑矩阵$\mathbf{F}_N=\mathbf{F}_{N1}\otimes\mathbf{F}_{N2}$，其中$\mathbf{F}_{N1}$和$\mathbf{F}_{N2}$分别是$N_1$和$N_2$维的 DFT 矩阵，$\otimes$表示 Kronecker 积（注意$\mathbf{F}_N$非 DFT 矩阵）。当$N_1=4,\ N_2=8$时，请在**同时满足约束 1 和 2** 的条件下，对$\mathcal{A}$和β进行优化，并计算最小误差和方案的硬件复杂度C。

问题 5：在问题 3 的基础上加上精度的限制来研究矩阵分解方案。要求将精度限制在 0.1 以内，即 RMSE≤0.1。对于$N=2^t, t=1,2,3...$的 DFT 矩阵$\mathbf{F}_N$，请在**同时满足约束 1 和 2** 的条件下，对$\mathcal{A}$和$\beta,\mathcal{P}$进行优化，并计算方案的硬件复杂度C。

参考文献

[1] James W. Cooley and John W. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series, Mathematics of Computation, vol. 19, no. 90, pp. 297-301, 1965. DOI:10.2307/2003354.

[2] K. R. Rao, D. N. Kim, and J. J. Hwang, Fast Fourier Transform: Algorithms and Applications, Springer, 2010.（中译本：快速傅里叶变换：算法与应用，万帅，杨付正译，机械工业出版社，2012.）

[3] Viduneth Ariyarathna, Arjuna Madanayake, Xinyao Tang, Diego Coelho, et al, Analog Approximate-FFT 8/16-Beam Algorithms, Architectures and CMOS Circuits for 5G Beamforming MIMO Transceivers, IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 8, no. 3, pp. 466-479, 2018. DOI: 10.1109/JETCAS.2018.2832177.

附录一：名词解释

- 复数乘法次数/复乘次数：进行复数乘法的次数，例如$(1+2j)\times(2+2j)$为一次复乘。
- 硬件复杂度：本题中，仅考虑乘法器带来的硬件复杂度，硬件复杂度仅与乘法器个数和每个乘法器的复杂度相关
- 乘法器个数：本题中，乘法器个数即为复乘次数

■ 单个乘法器的复杂度：单个乘法器的复杂度与乘法器的设计方法和输入数据的位宽等因素相关。在本题中，将乘法器的复杂度简化为仅与输入数据的取值范围相关。对于复数g∈ $\{x+jy|x,y\in\mathcal{P}\},\mathcal{P}=\{0,\pm1,\pm2,\ldots,\pm2^{q-1}\}$,其与任意复数z相乘的复杂度为q。