

## 附件 1

# 中俄数学挑战基金 2024 年度项目申报指南

## 一、总体目标

中俄数学挑战基金旨在推动数学领域的创新与发展，并促进数学成果在产业中的实际应用。产业界与学术界协同，共同凝练，发布数学难题；同时欢迎不同领域的研究学者积极申报课题，共同解决产业中的数学难题。

## 二、难题方向

2024 年度数学挑战基金池共计 700 万人民币，拟支持 14 个项目，每个项目拟资助 50 万人民币（税前），项目实施周期一般为 1 年。

序号	难题方向
1	线性分组码的最优缩短
2	大规模光网络规划保护与路由优化
3	高效计算 - 大规模定制布局优化算法
4	高效计算 - 超大组合空间的高效搜索算法
5	高效计算 - 芯片算法实现功耗优化
6	矩阵向量运算的最优实现
7	光纤非线性薛定谔偏微分方程精确高效求解问题
8	有限观测下的无线信道重建
9	基于数学规划模型的多信道联合资源分配算法
10	AI 领域 - 基于 AI 的动态无线环境下的多种 RRM 决策任务
11	AI 领域-AI 使能 HBF 架构下全量实时信道信息 (CSI) 获取
12	Score-based 扩散模型采样加速
13	大规模复数稀疏矩阵的 AI 加速迭代求解算法
14	通信避免的 LU 分解 (CALU) 的数值稳定性分析

注：受资助者在项目研究过程中形成的与项目相关的成果的著作权及专利等，包括但不限于论文、著作、源代码等，其知识产权权利归属，由资助方与受资助者及其所在单位共享，详细条款由项目资助协议具体协定。

## 指南方向 1: 线性码的最优缩短及相关应用

### 1. 背景

研究具有好参数和低复杂度编译码算法的纠错码是编码研究的中心课题。利用具有较好译码算法的码来构造新的具有更好参数的码是编码研究的一个重要方向，通常有打孔 (puncture)、缩短 (shorten)、直和、张量积等等。对已有较好译码算法的码的信息位进行缩短来优化其参数是常规做法，这样通常仅需要对现有译码算法进行简单调整便可以继续使用。缩短时可能会得到一个极小距离更大的码，多数情况下需要处理较多的坐标位置，这会导致码率下降较多。应用中期望在保持一定码率情况下(即缩短可接受的坐标位置的个数)来寻找参数最优的码。

由于软译码算法的应用，使得重量达到码的极小距离的码字个数也成为衡量码的纠错能力的重要指标。应用中希望通过缩短方式得到码的极小距离尽可能大的新码；如果无法使新码的极小距离增大，期望能在极小距离和码率相同的情况下，得到一个重量达到极小距离的码字个数最少的新码。

目前已有关于二元 Hamming 码，二元 Extended Hamming 码等的一些相关结果。对二元 Hamming 码已有清楚的缩短方案；对二元 Extended Hamming 码的缩短码，已有极小距离码字个数计数的上下界的部分刻画。进一步深入研究 Extended Hamming 码及更一般的码的缩短码的极小距离码字个数的精确值及上下界是一个有挑战性和有应用前景的研究课题。

### 2. 问题描述

设  $C$  为有限域  $GF(q)$  上的  $[n, k, d]$  线性码。设  $C_T$  表示码  $C$  在坐标位置集合  $T \subseteq \{1, 2, \dots, n\}$  上的缩短码 (shortened code)。假设  $C$  的生成矩阵为  $G = [-P | I_k]$ 。

**问题：** 对于  $T = \{i_1, \dots, i_s\} \subseteq \{n - k + 1, \dots, n\}$  的基数为  $|T| = s$  的列的标号，通过适当策略寻找基数为  $s$  的特殊集合  $T'$ ，使得  $C_{T'}$  是所有集合  $\{C_T\}$  中重量为  $d$  的码字个数最少的线性码。

### 3. 目标

a) **理论目标：** 研究有限域上一般线性码或具有丰富代数结构的线性码 (如循环码, BCH 码, 张量积码等) 的缩短码的重量达到极小距离的码字个数的刻画。具体来说:

- 1) 给出相应缩短码中码字重量达到码的极小距离的码字个数的上下界;
- 2) 在一些特殊情况下给出 (1) 中码字个数的计数公式;
- 3) 给出达到最优的缩短算法，并将其与已有结果进行对比。

b) **应用目标：** 给出二元 Extended Hamming 码及相关码的最优缩短方案和算法，并与前述一般线性码获得的结果做对比；例如，在缩短位数介于 30-40 之间时给出参数为  $[128, 120, 4]$  二元 Extended Hamming 码的最优缩短算法。

## 指南方向 2：大规模光网络规划保护与路由优化

### 1. 背景

光网络中的 RWA 问题是对两个节点间的业务寻找路由并且分配波长。该路由应该是连接原宿节点的一条简单路由。为了应对网络断纤，在网络规划的过程中，不仅需要规划工作路径还需要规划保护路径。

### 2. 问题描述

已知：

**网络：**给定一张无向图  $G$ ，该图包括节点  $V$  和边  $E$ ，其中边  $E$  有具体长度。

**约束：**最大的 lightpath 长度  $L_{max}$ ；一组频率变量  $[M] = \{1, 2, \dots, M\}$ ， $M=80$ ；以及每个频率的容量  $C_{max} = 80$ 。

**业务需求  $D$ 。** 一个业务  $d \in D$  可用向量  $(v, u, c)$  表示，其中  $v, u \in V$  是原、宿节点， $c$  是一个业务的容量需求， $c \in C$ ， $C = \{1, 2, 8, 32, 80\}$ 。

定义：

**Lightpath** 是一个元组  $(P, f)$ ，其中  $P \subseteq E$  是一条简单路径且无环路，并且该路径的长度  $\leq L_{max}$ ， $f$  是一个频率  $f \in [M]$ 。

**Scenario  $S \subseteq E$**  是一个网络状态，当某条边故障时，该边会暂时被从网络中移除时。所有使用故障边的 lightpath 都会因为该边的故障而失效。本题中仅仅考虑 1 次故障场景。

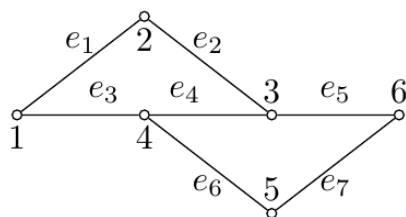
**Working path  $w(d)$**  是某个业务  $d=(v, u, c)$  的工作路径，该工作路径由一系列 lightpath 组成，这些 lightpaths 构成一个从  $v$  到  $u$  的一条简单路径。在没有任何边故障的情况下，该工作路径  $w(d)$  是激活状态。

**Recovery path  $r(d, S)$**  是某个业务  $d=(v, u, c)$  的工作路径。scenario  $S$  是边  $S$  故障场景，一系列 lightpath 构成一条从  $v$  到  $u$  的一条路径，并且该路径不包含故障边  $S$ 。仅仅在 scenario  $S$  下，路径  $r(d, S)$  是激活状态。

**Temporary configuration  $g(h, S)$**  是一个临时配置，可表示为一个向量  $(g_1, g_2, g_8, g_{32}, g_{80})$ ，该配置是 lightpath  $h$  在场景 scenario  $S$  下的配置。其中， $g_c$  是容量为  $c$  的业务的数量，且这些业务的激活路径在 scenario  $S$  下都使用了该 lightpath  $h$ 。

**Configuration  $G(h)$**  是一个 lightpath  $h$  的配置，它兼容了所有场景 scenario  $S$  的  $g(h, S)$ 。 $G(h) = (g_1, g_2, \dots, g_{80})$ ， $C(h) = \sum_{c \in C} c \cdot g_c$ 。

**Network**

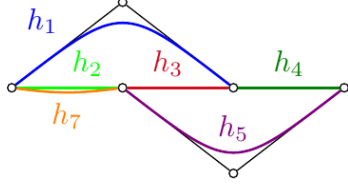


**Demands**

$$d_1 = (1, 6, 8), \quad d_2 = (1, 6, 8),$$

$$d_3 = (1, 3, 32), \quad d_4 = (1, 3, 32).$$

**Lightpaths**



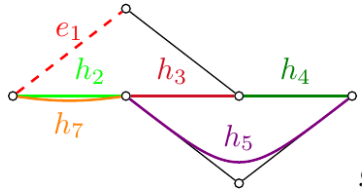
**Working paths**

$$w(d_1) = (h_2, h_3, h_4), \quad w(d_2) = (h_2, h_5),$$

$$w(d_3) = (h_1), \quad w(d_4) = (h_1),$$

$$g(h_3, \emptyset) = (0, 0, 1, 0, 0).$$

**Scenario{e<sub>1</sub>}**

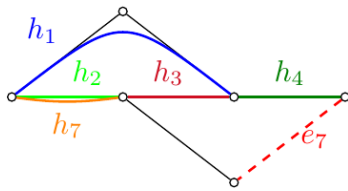


$$r(d_1, \{e_1\}) = (h_2, h_3, h_4), \quad r(d_2, \{e_1\}) = (h_2, h_5),$$

$$r(d_3, \{e_1\}) = (h_7, h_3), \quad r(d_4, \{e_1\}) = (h_7, h_3),$$

$$g(h_3, \{e_1\}) = (0, 0, 1, 2, 0).$$

**Scenario{e<sub>7</sub>}**



$$r(d_1, \{e_7\}) = (h_2, h_3, h_4), \quad r(d_2, \{e_7\}) = (h_2, h_3, h_4),$$

$$r(d_3, \{e_7\}) = (h_1), \quad r(d_4, \{e_7\}) = (h_1),$$

$$g(h_3, \{e_7\}) = (0, 0, 2, 0, 0).$$

**Configuration of h<sub>3</sub>**

$$G(h_3) = (0, 0, 2, 2, 0)$$

### 3. 目标

**寻找:**

Lightpath 集合  $H$ 。

每条业务  $d$  的工作路径 Working path  $w(d)$ 。

每条业务  $d$  的在每一个故障场景下的恢复路径 Recovery path  $r(d,S)$ , scenario  $S = \{e\}$ ,  $e \in E$ 。

最小化 Lightpath 的数量  $|H|$ 。

**约束:**

- 1) 任意一条边  $e \in E$  的任意一个频率  $f \in [M]$  不能同时被两个或者多个 lightpaths 使用。
- 2) 工作和恢复路径全部由 lightpath 构成,  $\text{lightpaths} \in H$ 。
- 3) 如果工作路径  $w(d)$  不包含故障边  $S$ , 则  $r(d,S) = w(d)$ 。
- 4) 任意 lightpath  $h \in H$ ,  $C(h) \leq C_{max}$ 。
- 5)  $|V| < 5000$ ,  $|E| < 10000$ ,  $|D| < 100000$ 。
- 6) 时间限制: 在普通 PC 上运行 1 小时。

- 参考文献

Reformulation and Decomposition Approaches for Traffic Routing in Optical Networks.

## 指南方向 3：大规模定制布局优化算法

### 1. 背景

传统芯片算法实现流程中，设计人员按照一定规模约束，将整个芯片算法系统划分成多个 HLB (High Level Block)，并完成 HLB 之间布局和 IO 接口规划，然后使用工具完成 HLB 内部自动布局布线。过程中，芯片设计工具识别出的布局布线瓶颈点由设计人员进行优化解决。当前芯片设计工具大多针对 GPU、NPU 等通用算力架构进行优化，针对 oDSP 算法架构进行布局布线优化，是未来降低连线功耗的有效途径。

### 2. 现状

通过分阶段优化、启发式初始放置、局部搜索等方法，可实现~300 个算法单元放置和优化。

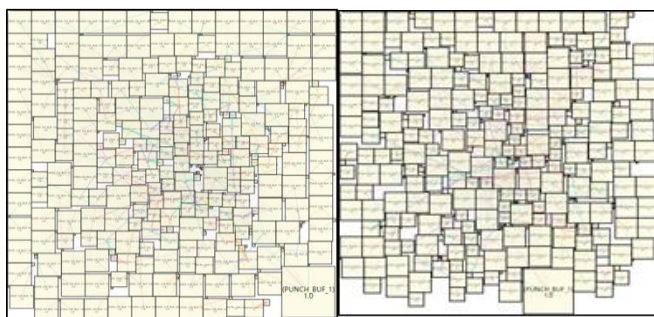


图 1 两类放置方式对比示意图

### 3. 问题描述

问题:解决 3K 规模单元宏布局问题,在满足面积利用率约束下使得总的连线代价最小。

本问题研究目标为在给定芯片算法单元集合下,通过拆分布局获得尽可能高的面积利用率和尽可能低的连线代价。

1、现有方案支持处理的单元颗粒度大,如果继续拆分,目标单元数预计增长 10 倍,单元布局合法放置空间大小呈指数速度上升。

2、为提升面积利用率,在放置基础上还需合理调整单元形状(面积利用率 = 单元总面积 / 布局矩形总面积 \* 100%)。

3、为降低连线代价,需将逻辑相连的单元靠近放置,而调整单元位置的同时可能影响面积利用率,实际设计中需要二者兼顾。

4、考虑基于给定单元的自动组合和拆分,从整体上进行面积利用率和连线代价的优化。

**输入：给定布局单元初始参数**

1、初始单元的集合 ( $|V| \leq 3000$ )，及初始单元面积  $a_i, \forall i \in V$ 。500 个元素的单元集初始输入参数示例如下：

NodeID	NodeArea	Parallel	NodeType
0	0	400	I0-I
1	62400	400	FuncUnit
2	84800	400	FuncUnit
3	16400	1	FuncUnit
4	20000	1	FuncUnit
...	...	...	...
500	0	400	I0-0

- a. NodeID: 初始单元编号
- b. NodeArea: 初始单元面积参数
- c. Parallel: 初始单元并行度，例如 Node2 为 400 个并行工作且相互独立的相同单元，总面积为 84800；可在布局时根据需要进行拆分，拆分出的小单元之间相互独立，无连接关系。并行度为 1 的单元表示该单元为一个整体，内部子模块有相互连接关系，不支持自动拆分。
- d. NodeType: 节点类型，其中 I0-0 代表数据输入，I0-I 代表输出，I0 接口分布于整体布局矩形的边缘，不占面积；FuncUnit 为功能单元。

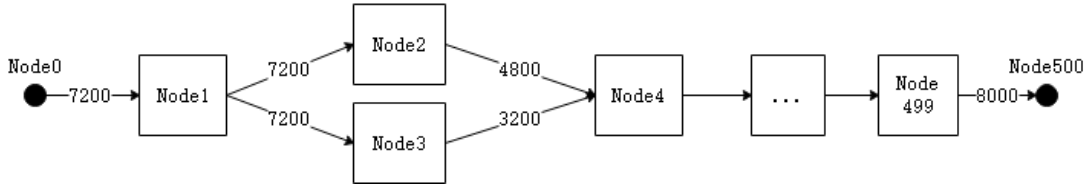
2、单元间的连接关系  $E$ ，每一条连接的边权  $w: E \rightarrow \mathbb{N}$  表示连线数量。输入参数示例：下表所示为 4 组连接关系

SrcNodeID	DstNodeID	ConnectionWeight
0	1	7200
1	2	7200
1	3	7200
2	4	4800
3	4	3200
...	...	...
499	500	8000

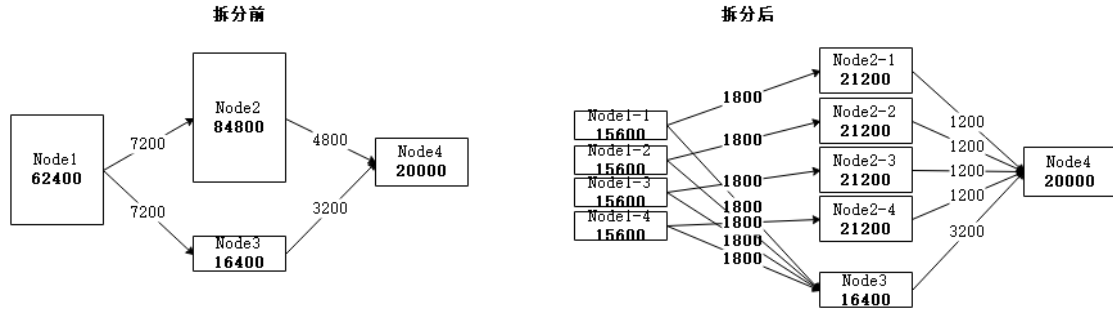
- a. SrcNodeID: 连接源单元 ID
- b. DstNodeID: 连接目标单元 ID
- c. Connection Weight: 连接线权重参数，连线权重支持根据源单元或目标单元按照并行度进行拆分。

3、非必要提示信息：初始单元宽高集合  $(w_{i0}, h_{i0})_{i \in V}$ ,  $w_{i0}h_{i0} = a_i$ ；初始布局区域总宽高  $W_0, H_0$

表格所示子系统逻辑图如下：

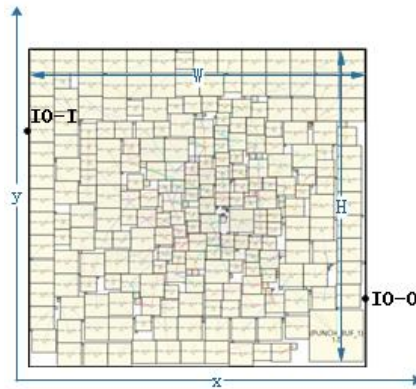


按并行度拆分示例：第 3 组连接关系源 Node1 和 Node2 均匀 1 拆 4，Node3 不支持拆分；拆分前后对比下图所示：



输出：

- 1、最终布局区域总宽高  $W, H$  及布局面积利用率
- 2、最终单元集每个单元的横/纵坐标  $x: V \rightarrow [0, W], y: V \rightarrow [0, H]$  (如进行了单元拆分, 则输出拆分后单元集坐标)
- 3、最终单元宽高集合  $(w_{i1}, h_{i1})_{i \in V}$
- 4、布局效果图, 示例如下:



#### 4. 目标

a) 理论目标:

面积利用率和总线长下界理论分析

- a. 理论分析单元颗粒度面积分布和面积利用率之间关系
- b. 给定输入下总连线代价下界分析  $\sum_{e \in E} w_e \cdot HPWL_e(x, y)$

b)、应用目标:

完成给定输入下单元布局，最小化总线长  $\sum_{e \in E} w_e \cdot HPWL_e(x, y)$

约束：

1、IO 节点分布在整個布局矩形框的左右边缘，方向为左进右出；具体位置不约束。

2、布局面积利用率  $\geq 90\%$ :  $\frac{\sum_{i \in V} a_i}{WH} \geq 0.9$

3、单元面积恒定约束:  $w_{i1}h_{i1} = a_i, \forall i \in V$

4、布局区域总宽高比约束:  $2:1 < W:H < 1:2$

5、所有模块位于布局区域内:  $\forall i \in V, x_i + \frac{w_i}{2} \leq W, y_i + \frac{h_i}{2} \leq H$

6、单元相互不重叠:  $\forall i \neq j \in V, R_i(x, y) \cap R_j(x, y) = \emptyset$

$R_i(x, y) = \{(a, b): x_i - \frac{w_i}{2} \leq a \leq x_i + \frac{w_i}{2}, y_i - \frac{h_i}{2} \leq b \leq y_i + \frac{h_i}{2}\}$  为单元  $i$  对应的矩形区域

#### • 参考文献

[1] Module Placement on BSG-Structure and IC Layout Applications. Shigetoshi NAKATAKey Department of Electrical and Electronic Engineering Tokyo Institute of Technology;

<https://dl.acm.org/doi/pdf/10.5555/244522.244865>

[2] TCG: A Transitive Closure Graph-Based Representation for General Floorplans. Jai-Ming Lin and Yao-Wen Chang, Member, IEEE <http://cc.ee.ntu.edu.tw/~ywachang/Papers/tvlsi-tcg.pdf>

## 指南方向 4：高效计算— 超大组合空间的高效搜索算法

### 1. 背景

多天线系统会向更大带宽和更多天线持续演进。未来对加速中等尺寸 (X100~X1000) 的矩阵运算需求强烈。矩阵伪逆的运算加速更是其中的核心诉求之一。本课题目标是探索在给定算子集合上矩阵伪逆集合运算的自动化加速。

### 2. 现状

基于深度学习，业界有针对矩阵乘[1]和排序算法[2]的自动化加速进行了研究。但针对矩阵的伪逆运算还没有很好的结果。如图 1 所示，对中等尺寸的矩阵集合的伪逆运算的优化目前还主要是依靠人工优化。



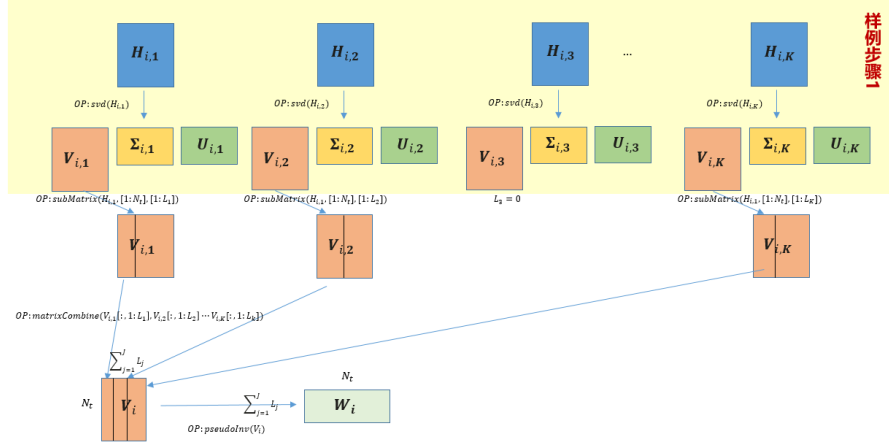


图 1 一个执行流程样例

### 3. 问题描述

#### a) 问题定义

给定一个矩阵集合  $\{H_{m,k}\}$ , 一个实对角矩阵  $D$ , 和一个非负整数集合  $\{L_{m,k}\}$ :

$$H_{m,k} \in \mathbb{C}^{N_t \times N_r}, \quad m \in \{0, \dots, M-1\}, \quad k \in \{0, \dots, K-1\},$$

$$D = \alpha \cdot I, \quad \alpha \in \mathbb{R}^+,$$

$$L_{m,k} \in \{0, 1, \dots, N_r\},$$

且满足

$$N_t \geq \sum_k L_{m,k}, \quad m \in \{0, \dots, M-1\}$$

针对在所有  $m$  上完成计算功能:

$$W_m = V_m (V_m^H V_m + D)^{-1}, \quad m \in \{0, \dots, M-1\}$$

寻找在给定的算子集合/action space 上, 结果正确, 且运算代价/cost 最小的计算图, 其中

$$\begin{aligned} V_m &= [\dots \bar{V}_{m,k} \dots] \\ \bar{V}_{m,k} &= V_{m,k}[:, \mathbf{0}:L_{m,k}-1] \\ \{U_{m,k}, \Sigma_{m,k}, V_{m,k}\} &= \text{SVD}(H_{m,k}) \end{aligned}$$

#### b) 参数说明

$$N_t \in \{64, 128, 256\},$$

$$N_r \in \{4, 8, 16\},$$

$$M = \{192, 384, 768, 1536\},$$

$$K = 100$$

算子名称	算子格式	Lapack函数	详细信息链接	算子输出	cost计算 (参考计算量)	cost计算 (特定硬件代价表)
矩阵乘	OP:matMul(A,B)	zgemm		C:计算结果	矩阵A的大小为m*n,矩阵B的大小为n*p,算子开销为m*n*p	查表计算
满阵乘下三角	OP:matMulTriangle(A,B)	dtrmm		C:计算结果	矩阵A的大小为m*n,矩阵B的大小为n*n,算子开销为0.5*m*n <sup>2</sup>	查表计算
下三角乘满阵	OP:triangleMulMat(A,B)	trmm		C:计算结果	矩阵A的大小为n*n,矩阵B的大小为n*p,算子开销为0.5*n <sup>2</sup> *p	查表计算
矩阵乘列	OP:matMulVec(A,B)	dgemv		C:计算结果	矩阵A的大小为n*m,向量B大小为m,算子开销为n*m	查表计算
共轭向量内积	OP:conjugateInnerProduct(A)	zdotc		B:计算结果	向量A大小为n,算子开销为n <sup>2</sup>	查表计算
向量乘	OP:vectorMul(A,B)	axpy		C:计算结果	向量A大小为n,向量B大小为n,算子开销为n <sup>2</sup>	查表计算
共轭向量外积	OP:conjugateOuterProduct(A)	Hadamard(自定义函数)		B:计算结果	向量A大小为n,算子开销为n <sup>2</sup>	查表计算
向量内积	OP:vectorInnerProduct(A,B)	dot		C:计算结果	向量A大小为n,向量B大小为n,算子开销为n <sup>2</sup>	查表计算
向量加法	OP:vectorAdd(A,B)	Vaddl(自定义函数)		C:计算结果	向量A大小为n,向量B大小为n,算子开销为n	查表计算
SVD分解	OP:svd(A)	dgesvd		V,Sigma,U:左奇异向量,奇异值矩阵,右奇异向量	查表计算	查表计算
子矩阵提取	OP:subMatrix(A,[x_start,x_end],[y_start,y_end])	Submatrix(自定义函数)		B:计算结果	算子开销为(x_end-x_start)*(y_end-y_start)	查表计算
矩阵合并	OP:matrixCombine(A,B)	Mergedmatrix(自定义函数)		res:计算结果	假设每个矩阵的尺寸为n*m,j,算子开销为n*(sum_i m_i)	查表计算
Cholesky分解	OP:chol(A)	dpctrf		L:计算结果	查表计算	查表计算

图 2 算子集合的基本信息示例

算子集合基本信息如图 2 所示，其中包含：

- 算子集合&典型规格：（取自部分）Lapack

矩阵乘：*matrixMul*  
 矩阵分解：*subMatrix*  
 矩阵合并：*matrixCombine*  
 共轭乘：*conjugateMul*  
 矩阵求逆：*inv*  
 矩阵伪逆：*pseudoInv*  
 ...

- 对应的算子开销函数

函数形式  
 表格形式

#### 4. 目标

##### b) 理论目标：

在给定基础算子集合的条件下，研究特定矩阵计算函数的自动优化方法（包括但不限于 AI 类方法）。

##### c) 应用目标：

设计搜索算法，在给定如下输入的条件下：

- 数据矩阵集合，包含  $\{H_{m,k}\}$ ， $D$ ，及非负整数集合  $\{L_{m,k}\}$
- 目标矩阵的计算公式： $W_m = V_m(V_m^H V_m + D)^{-1}$
- 可用的算子集合
- 与算子集合对应的算子开销
- 计算核心数量  $C$ ，（各核心同构，核心间数据无代价搬移）

输出：

- 使用给定算子集合中的算子，构造计算代价最小的 Top3 形式化计算流程（例如，计算图）：输入数据矩阵集合  $\{H_{m,k}\}$ ，通过形式化计算流程，输出计算结果集

合  $\{\widehat{W}_m\}$ , 满足

$$\max_m |\widehat{W}_m - W_m| < \sigma$$

其中,  $\sigma$  为计算精度需求。

• 参考文献

[1]. Discovering faster matrix multiplication algorithms with reinforcement learning, Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, ..., Nature volume 610, pages47 - 53 (2022)

[2]. Faster sorting algorithms discovered using deep reinforcement learning, Daniel J. M., Andrea M., Anton Z., ..., Nature volume 618, pages257 - 263 (2023)

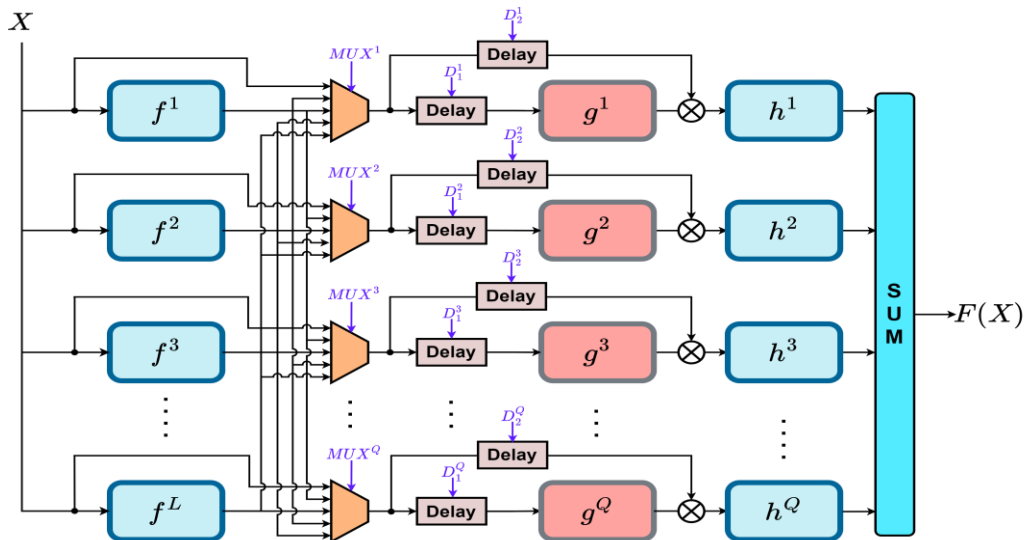
## 指南方向 5: 高效计算-芯片算法的低功耗实现-低比特位宽前向网络

### 1. 背景

通信算法需要支持的带宽和速率越来越大, 算法所需功耗/芯片面积也越来越大, 位宽设计是降低硬件开销的重要技术路径, 希望研究典型的位宽设计算法, 降低硬件开销。

### 2. 现状

如下图所示模型是一种常见的建模复杂非线性的级联结构, 建模能力通常较非级联结构强, 模型中的位宽一般选取为 12~16bit, 位宽的大小直接影响了芯片的资源



### 3. 问题描述

优化目标:

$$\begin{aligned} & \min_{B_X, B_\alpha, B_C, B_\beta} \text{Cost}(F, X) \\ & \text{st. } \|F(X) - F_\infty(X_\infty)\| \leq e \end{aligned}$$

**参数说明:**

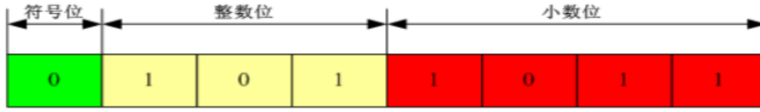
其中,  $X = [X(1), X(2), \dots] \in \mathbb{C}^N$  为输入信号序列,  $N$  为输入信号长度;  $X_\infty$  表示  $X$  全精度浮点版本;  $e \in \mathbb{R}^+$  为期望误差上限;  $\text{Cost}(\cdot)$  为代价函数

前向函数  $F(\cdot)$  由下列子函数按图所示拓扑组合而成;  $F_\infty(\cdot)$  表示  $F(\cdot)$  的全精度浮点版本:

$$\begin{aligned} f^i: y^i(n) &= \sum_{m=0}^M \alpha_m^i x^i(n-m), i = 1, 2, \dots, L \\ D_j^i: y^i(n) &= x^i(n - D_j^i), \quad i = 1, 2, \dots, Q, \quad j = 1, 2 \\ g^i: y^i(n) &= \sum_{p=0}^P c_p^i |x^i(n)|^{2p}, i = 1, 2, \dots, Q \\ h^i: y^i(n) &= \sum_{m=0}^M \beta_m^i x^i(n-m), i = 1, 2, \dots, Q \end{aligned}$$

其中  $D_j^i, p \in \mathbb{Z}^+$ ;  $\alpha_m^i, c_p^i, \beta_m^i \in \mathbb{C}$ ;  $B_X, B_\alpha, B_C, B_\beta$  分别表示信号  $X$ , 子函数系数  $\alpha_m^i, c_p^i, \beta_m^i$  的位宽。

案例  $B = 8$



#### 4. 目标

- 一种针对前向网络结构的位宽搜索和分析方法, 位宽相对基线优化40%
- 使能最小化如上网络中各个节点信号和系数的位宽, 同时保证位宽压缩后输出信号  $F(X)$ , 与无限精度 (浮点) 输出信号  $F_\infty(X_\infty)$  之间误差小于设定值  $e (e < 0.5\text{dB})$

• **参考文献**

[1] Dinis D C, Cordeiro R F, Barradas F M, et al. Agile single-and dual-band all-digital transmitter based on a precompensated tunable delta-sigma modulator. IEEE Transactions on Microwave Theory and Techniques, 2016, 64(12): 4720-4730.

[2] Gholami A, Kim S, Dong Z, et al. A survey of quantization methods for efficient neural network inference. Low-Power Computer Vision. Chapman and Hall, CRC, 2022: 291-326.

[3] Nagel M, Fournarakis M, Amjad R A, et al. A white paper on neural network quantization. arXiv preprint arXiv:2106.08295, 2021

[4] X. Liu, W. Chen and Z. Feng, "Broadband Digital Predistortion Utilizing Parallel Quasi-Wiener-Hammerstein Model with Extended Dynamic Range," 2021 IEEE MTT-S International Wireless Symposium (IWS), Nanjing, China, 2021, pp. 1-3.

## 指南方向 6：矩阵向量运算的最优实现

### 1. 背景

FIR 滤波器以及离散傅里叶变换 DFT 是现代通信芯片中的能耗大户，也是最基础的运算单元之一。他们被广泛的用于包括特征提取，信道均衡，调制解调等计算。掌握一种高效的 FIR/DFT 实现结构，对于改善通信芯片的能效比具有重大意义。

用数学的方式来描述，FIR 和 DFT 可以被写成

$$\mathbf{y} = \mathbf{H}\mathbf{x}$$

其中  $\mathbf{x}^T = \{x_0 \ x_1 \ \dots \ x_{N-1}\}$  作为系统的输入向量， $\mathbf{H}$  作为计算矩阵， $\mathbf{y}$  作为输出向量。

对于 DFT 运算来说， $\mathbf{H}$  是一个大小为  $N$  的正方矩阵且其每一元素值为  $(\mathbf{H}_N)_{mn} = e^{-j\frac{mn2\pi}{N}}$ 。对于长度为  $L$  的 FIR 滤波器而言，它的计算矩阵  $\mathbf{H}$  是一个大小为  $L \times N$  的 Toeplitz 矩阵。

由于 DFT/FIR 中的计算矩阵  $\mathbf{H}$  都具有一定的结构特征，如果我们能找到一种聪明的对矩阵  $\mathbf{H}$  的分解方式，我们便有机会能实现对 DFT/FIR 运算电路的大幅简化。

举例来，经典的 Toom-Cook 算法利用了拉格朗日多项式对矩阵  $\mathbf{H}$  进行分解以降低乘法的运算次数[1]。

$$\begin{aligned} \mathbf{y} = \mathbf{H}\mathbf{x} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} s(\beta_0) \\ 0.5s(\beta_1) \\ 0.5s(\beta_2) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} h_0 & 0 & 0 \\ 0 & 0.5(h_0 + h_1) & 0 \\ 0 & 0 & 0.5(h_0 - h_1) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \mathbf{x} \end{aligned}$$

其中  $\mathbf{H} = \begin{bmatrix} h_0 & 0 \\ h_1 & h_0 \\ 0 & h_1 \end{bmatrix}$ ， $\beta_0, \beta_1, \beta_2$  为一组不同的实数， $s(\beta)$  则为关于  $\mathbf{H}$  和  $\mathbf{x}$  的拉格朗日插值结果[1]。

可以看出，我们把矩阵向量的分解计算问题和 FIR/DFT 电路的微架构实现问题进行了映射。不过，高效的找到矩阵向量问题的高效分解方式并非易事。首先，搜索空间巨大。让我们考虑  $3 \times 3$  卷积的情况，并应用一个约束，即分解的矩阵元素取值  $\mathbf{F} = \{0, \pm 1\}$ ，这样芯片实现时就不会涉及乘法。在这种情况下，潜在分解结果的维数仍超过  $10^{23}$ 。此外，由于我们找到一个好的分解方法的目的是为了得到一个高能效的电路架构。我们在计算高效的分解方式时必须考虑来自实际实现的 ASIC 的影响和约束，例如线长、互连、驱动强度等等，这使得搜索更加困难。在许多情况下，电路设计人员发现，在理论上操作次数较少的分解并不一定能转化为硬件效率较高的电路架构。

### 2. 现状

当前的 FIR/FFT 设计领域，主要依赖算法和芯片设计人员从一些经典结构出发，例如 split Radix FFT 结构或者 Winograd FIR，针对特定的芯片指标进行手工设计。对于更广义的矩阵运算优化问题，有大量基于增强学习的方法，例如“Discovering novel algorithms with AlphaTernor”。

### 3. 问题描述

对于  $\mathbf{y} = \mathbf{H}\mathbf{x}$ ，假定  $\mathbf{H}$  的大小不高于  $100 \times 100$  且  $\mathbf{H}$  的翻转率是  $\mathbf{x}$  的 1%，找出满足实际需要的，拥有乘法次数从最小到最大的解的集合。其中，翻转率的定义可以被直观的理解为，对于矩阵  $\mathbf{H}$  中所含元素的计算操作的代价为向量  $\mathbf{x}$  中所含元素代价的 1%。

- 参考文献

[1] [www.ece.umn.edu/users/parhi/SLIDES/chap8.pdf](http://www.ece.umn.edu/users/parhi/SLIDES/chap8.pdf)

## 指南方向 7：光纤非线性薛定谔方程求解

### 1. 背景

在高速光纤链路中，光强度对光的折射率会产生影响，从而导致克尔 (Kerr) 非线性效应，这是限制 WDM 通信系统容量增长的主要制约因素之一。从麦克斯韦方程组出发，基于光纤的晶体特征，由克尔非线性可以推导得出表征该效应的光纤非线性薛定谔方程。在 WDM 系统中，若要实现信号的精准仿真，需要对描述光域信号的一类非线性薛定谔方程，即 Manakov-PMD 方程，进行有效求解。然而对该方程在真实通信场景下的精确求解通常需要非常大的计算量，例如，若使用单核处理器，往往需要几周的时间才能完成相应场景的计算。

### 2. 现状

由于超宽带光纤非线性信道模型非常复杂，导致数字仿真所需的计算量巨大，这已成为光纤光学领域高效、高精度求解算法发展的主要瓶颈。当前针对光纤非线性薛定谔方程，主要通过根据信号光纤传输特征并借助快速傅里叶变换优势而专门发展起来的所谓分步傅里叶方法 (Govind p. Agrawal, Nonlinear Fiber Optics, Academic Pres, 2013) 来进行数值求解。但即使如此，单波 2000 公里仿真仍需要 8 个小时，5 个波需要一周时间，而实际系统满波可达 80 个波。

### 3. 问题描述

寻求较已有经典方法更为高效的 Manakov-PMD 方程求解方法。该方程具有如下形式：

$$\frac{\partial \mathbf{E}(z, t)}{\partial z} + j \frac{\beta_2}{2} \frac{\partial^2 \mathbf{E}(z, t)}{\partial t^2} + \frac{\alpha}{2} \mathbf{E}(z, t) = j\gamma |\mathbf{E}(z, t)|^2 \mathbf{E}(z, t), \quad z > 0, \quad t > 0$$

其中  $\mathbf{E}(z, t) = [E_x(z, t), E_y(z, t)]^T \in L^2(\mathbb{R})$  为光信号包络信息函数，单位  $w$  (瓦)； $z$  代表传输距离，其最大值  $Z$  在千公里量级； $t$  代表传输时间，由输入数据序列决定，和符号速率有关，方程中视为变量，单位  $s$  (秒)； $\beta_2$  为正实数，表征色散特征，量级在  $10^{-27}$  到  $10^{-26}$  之间，单位  $s^2/m$ ； $\alpha$  为正实数，代表衰减系数，量级在  $10^{-5}$  左右； $\gamma$  为表征光纤非线性特征的正实数，量级在  $10^{-4}$  到  $10^{-3}$  之间，单位  $1/(m \cdot w)$ 。方程中参数  $\beta_2$ ,  $\alpha$ ,  $\gamma$ ,  $Z$ , 和发端信号  $E(0, t)$  已知。同时，信号  $\mathbf{E}(z, t)$  在传输过程中，通常会进行中继集总放大：假定链路为等跨，每跨光纤长度为  $L$ ，当传输距离为每跨的终点时 ( $z = k \cdot L$ ,  $k = 0, 1, \dots, Z/L$ )，信号幅值会被放大  $G = e^{\alpha L/2}$  倍。

### 4. 目标

#### a) 理论目标：

构建高效的非线性薛定谔方程求解新方法，在精度不变的情况下，实现相比分步傅里叶模型运算量减少 10 倍以上。并在理论上证明该解法的可靠性。

b)、应用目标:

区分场景，在传输距离>1200km 或<160km 情况下，设计针对特定场景的高效信道高效求解模式。

## 指南方向 8：有限观测下的无线信道重建 - 寻找更有效的模型

### 1. 背景

如图 1所示Massive MIMO是无线接入网络未来演进的重要方向，其性能提升严重依赖所能获取的信道信息的精度。随着Massive MIMO通道数和用户数的提升，传统方法获取高精度信道信息需要承受越来越高的空口资源开销和计算复杂度，严重制约了Massive MIMO的演进。本命题目标探索Massive MIMO信道信息获取的新方法和理论性能极限。

• Massive MIMO的下行传输示意

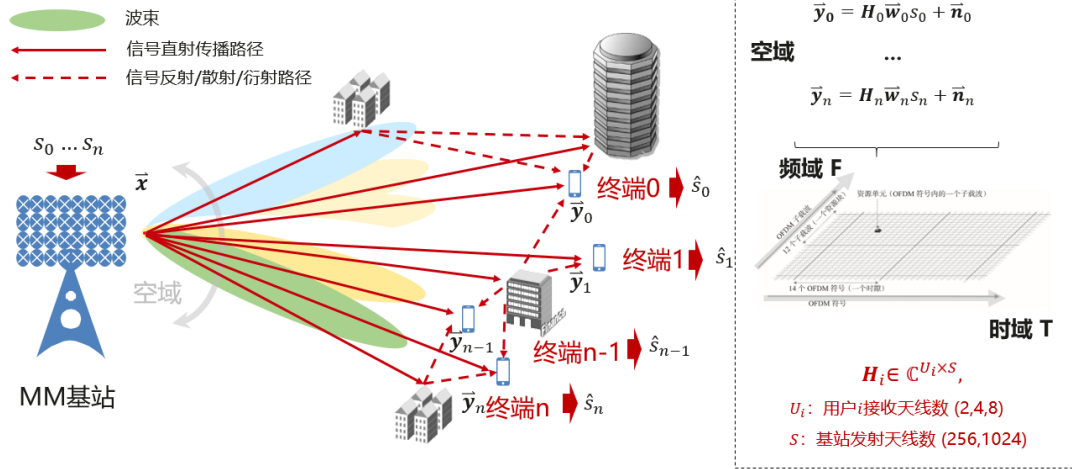


图 1 Massive MIMO的下行传输

### 2. 现状

除去 LS, MMSE 等经典线性方法，学术界提出了使用压缩感知[1]类方法利用稀疏性特征增强信道估计，并提出原子范数等方法解决 off-grid 问题[2][3]，也从优化的角度研究了稀疏性方法的性能界[4]。但由于无线信道并不是学术定义的严格稀疏，且以上算法的复杂度较高，压缩感知并没有很好的应用于现实。

### 3. 问题描述

a) 问题定义:

Massive MIMO 信道重建可以用如图 2 所示的反问题[5]模型描述，建模为求取最低维

度参数空间。

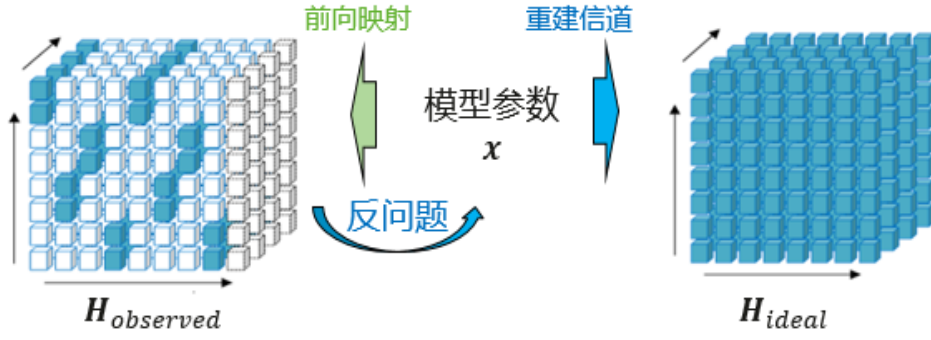


图 2 信道重建的反问题模型

$$\begin{aligned} & \min_{x, D} (N) \\ & s. t. \mathbf{H}_{est} = \mathbf{D}(x) \\ & \|\mathbf{H}_{est} - \mathbf{H}_{ideal}\| \leq e \\ & \mathbf{H}_{observed} = \mathcal{P}_{\Omega}(\mathbf{H}_{ideal} + \mathbf{n}) \end{aligned}$$

其中

$$\begin{aligned} & x \in \mathbb{C}^N \\ & \mathbf{D} \in \mathbb{C}^N \rightarrow \mathbb{C}^{U \times S \times F \times T} \\ & e \in \mathcal{R}^+ \\ & \mathbf{n} \in \mathcal{N}(0, \sigma^2 \mathbf{1}) \\ & [\mathcal{P}_{\Omega}(\mathbf{H})]_{u,s,f,t} = \begin{cases} \mathbf{H}_{u,s,f,t} & \text{if } (u, s, f, t) \in \Omega \\ \mathbf{0} & \text{if } (u, s, f, t) \notin \Omega \end{cases} \end{aligned}$$

$\Omega$  表征观测图样，有多种配置。

b) 参数说明：

$$\begin{aligned} & U \in [4, 8] \\ & S \in [128, 256, 1024] \\ & F \in [1024, 2048, 4096] \\ & T \in [10, 20, 50] \end{aligned}$$

#### 4. 研究目标

a) 理论目标

- 低复杂度的多域结构性信道重建模型：通过受到噪声污染的观测  $\mathbf{H}_{observed}$ ，寻找最低维模型的参数空间  $x$  和对应变换  $\mathbf{D}$  来重构目标矩阵  $\mathbf{H}_{ideal}$ ，且重构误差小于等于  $e$ ；
- 通过受到噪声污染的有限观测矩阵  $\mathbf{H}_{observed}$  估计模型参数  $x$  的算法；
- 重建精度界分析： $\|\mathbf{H}_{est} - \mathbf{H}_{ideal}\| \leq e$  满足所需的条件；重建精度  $e$  与维度  $N$  和噪声方差  $\sigma^2$  的关系；

b) 应用目标

- 推荐模型和重建算法能够工作于如下场景 [6] [7] 中工作：



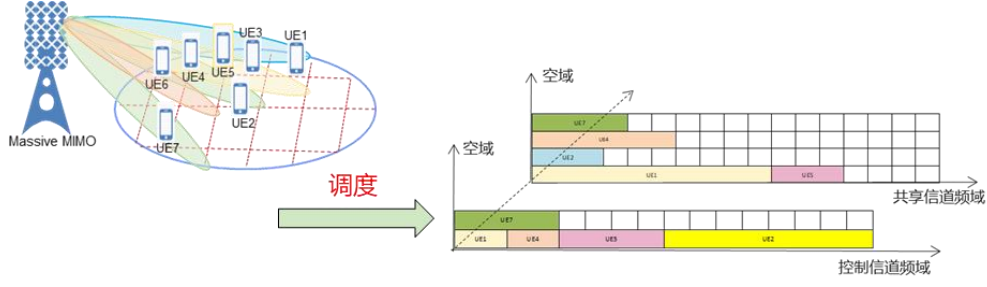
- ✓ 信道响应在多普勒域，延迟域，方位角，俯仰角，极化多域张成的超空间内分布；
  - ✓ 信道响应在每个单一维度上都不是理想稀疏，单个维度上甚至可能是稠密的，但在多域联合张成的空间中是近似稀疏或稀疏的；
  - 观测过程非理想
    - ✓  $H_{observed}$  被噪声污染，观测信噪比低（0dB ~ 10dB）；
    - ✓ 观测矩阵初始相位未知，且不连续
  - 基于推荐模型的信道重建相对 MMSE 算法有精度收益；
  - 基于推荐模型的信道重建具有较低复杂度：常数 × MMSE 估计 > 推荐模型 > LS
- 参考文献
- [1]. Christian R. Berger..., Application of Compressive Sensing to Sparse Channel Estimation, 2010;
  - [2]. Yingming Tsai,, Millimeter-Wave Beamformed Full-dimensional MIMO Channel Estimation Based on Atomic Norm Minimization, 2017
  - [3]. JunQuan Deng...MmWave Channel Estimation via Atomic Norm Minimization for Multi-User Hybrid Precoding, 2018 (WCNC);
  - [4]. E. J. Candès and T. Tao, "Decoding by Linear Programming," IEEE Trans. Info. Theory, vol. 51, no. 12, Dec. 2005
  - [5]. Tarantola, Albert (2005). "Front Matter" (PDF). Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM
  - [6]. 3GPP TR 36.873 V12.7.0, Study on 3D channel model for LTE
  - [7]. 3GPP 3D MIMO channel model: a holistic implementation guideline for open source simulation tools

## 指南方向 9：基于数学规划模型的多信道联合资源分配算法

### 1. 背景

在 5G-NR 通信系统中，gNodeB 使用 Massive MIMO 技术显著增强了网络容量。我们正聚焦于一个核心挑战：研发高效且高性能的空口信道调度算法，需精密平衡算法复杂度与性能。传统逐信道资源分配常陷于局部最优，无法达致全局最佳。

我们正探索一种创新低复杂度方案，联合优化控制与共享信道资源，旨在智能整合资源、最大化系统吞吐，同时确保算法简单高效，利于实际部署运维，不增加过多计算负担。这一策略力求在理论与实践开辟新径，高效解锁 5G-NR 的广阔容量。



## 2. 现状

传统联合优化算法倾向于使用启发式方法，虽简化了复杂度，但往往与最优解存在较大差距。相比之下，基于数学求解器的通用算法致力于多轮迭代逼近最优解，却因高复杂度而效率受限。

目前业界实践大多通过标量运算实现，未能充分利用现代计算机硬件的矢量计算优势。此外，现行算法多为串行设计，或采用并行策略但资源利用率不高，进一步制约了算法效能的提升。

## 3. 问题描述

问题定义：

$$\max_{A \in \{0,1\}^{U \times C}, B \in \{0,1\}^{U \times R}} \langle x, y \rangle_w$$

$$s. t. x_u = f(A_{u,:}), u \in \mathbb{U}$$

$$y_u = g(B, u), u \in \mathbb{U}$$

$$g(B, u) = \min\{d_u, \sum_{j=0}^{R-1} \gamma \cdot \log_2 \left( 1 + s_u \cdot \frac{b_{u,j}}{\|B_{:,j}\|_1} \cdot \prod_{v \in \mathbb{U}, b_{v,j}=1 \text{ and } u \neq v} (1 - r_{u,v}) \right)\}$$

$$f(A_{u,:})$$

$$= \begin{cases} 1, & \text{where } \|A_{u,:}\|_1 = \max\{j | a_{uj} = 1\} - \min\{j | a_{uj} = 1\} + 1 = l_u, \min\{j | a_{uj} = 1\} \in \mathbb{T}_u \\ 0, & \end{cases}$$

$$\|B_{:,j}\|_1 \leq \beta, \forall j < R$$

$$\|A_{:,j}\|_1 \leq \alpha, \forall j < C$$

$$\forall u, v < U, u \neq v \text{ if } \exists j < C \text{ s. t. } a_{uj} = a_{vj} = 1, \text{ then } \mathbb{D}_u \cap \mathbb{D}_v = \emptyset$$

$$A \in \{0,1\}^{U \times C}, \quad a_{i,j} \text{ 代表矩阵 } A \text{ 的 } i \text{ 行 } j \text{ 列元素}; \quad B \in \{0,1\}^{U \times R}, \quad b_{i,j}$$

代表矩阵  $B$  的  $i$  行  $j$  列元素

参数说明：

$$R \in \mathbb{N}, R \leq 64, \mathbb{S} = \{1, 2, \dots, R\};$$

$$C \in \mathbb{N}, C \leq 135, \mathbb{E} = \{1, 2, \dots, C\};$$

$$U \in \mathbb{N}, U \leq 20, U = \{1, 2, \dots, U\};$$

$$w \in \mathbb{N}^U, s \in \mathbb{N}^U, d \in \mathbb{N}^U$$

$$\alpha \in \mathbb{N}, \alpha \leq 4; \beta \in \mathbb{N}, \beta \leq 8; \gamma \in \mathbb{N}$$

$$F = \{0, 1, 2, \dots, 31\},$$

$$V^{U \times U} = \begin{bmatrix} r_{11} & \cdots & r_{1U} \\ \vdots & \ddots & \vdots \\ r_{U1} & \cdots & r_{UU} \end{bmatrix}, 0 \leq r_{i,j} = r_{j,i} \leq 1$$

$$l_u \in \{1, 2, 4, 8, 16\}, T_u \subset \mathbb{E}, D_u \subset F$$

#### 4. 目标

##### d) 理论目标:

研究离散优化问题的矩阵计算和并行计算实现理论:

- 1) **矩阵与图论方法的应用:** 利用矩阵理论和图论工具加速离散优化问题求解。
- 2) **并行算法的设计与分析:** 创造能用矩阵运算并行处理的算法, 适应 GPU 或分布式系统。
- 3) **并行计算效率与通信开销平衡:** 平衡计算效率和通信成本, 减少数据交换, 提高资源利用率。

##### e) 应用目标:

探索低复杂度的联合寻优算法, 侧重矢量化与并行化策略, 旨在达成以下优化目标:

- 1) **矢量化算法架构:** 通过精心设计算法流程, 确保其能直接映射为高效的矢量运算, 使得矢量计算部分占算法总执行过程的至少 80%, 从而大幅提升计算密集型任务的处理速度。
- 2) **并行化算法架构:** 在确保至少 80% 的系统资源有效利用基础上, 实现算法的并行化部署, 目标为达到 4 倍的加速效果, 即并行执行相较于串行执行时间缩短至四分之一, 同时保证寻优解不低于理论最优解的 80%, 以此在保证解质量的同时, 极大缩短算法执行时间。

#### • 参考文献

- 1) P. R. M., M. R., A. Kumar and K. Kuchi, "Downlink Resource Allocation for 5G-NR Massive MIMO Systems," 2021 National Conference on Communications (NCC), Kanpur, India, 2021, pp. 1-6, doi: 10.1109/NCC52529.2021.9530169.
- 2) Y. -H. Liu and K. C. -J. Lin, "Traffic-Aware Resource Allocation for Multi-User Beamforming," in IEEE Transactions on Mobile Computing, vol. 22, no. 6, pp. 3677-3690, 1 June 2023, doi: 10.1109/TMC.2022.3141787.
- 3) Kepner, J. & Gilbert, J. R. (eds.) (2011). Graph Algorithms in the Language of Linear Algebra (Vol. 22). SIAM. ISBN: 978-0-89871-990-1

- 4) Y. Chen, Y. Wu, Y. T. Hou and W. Lou, "mCore+: A Real-Time Design Achieving  $\sim 500 \mu\text{s}$  Scheduling for 5G MU-MIMO Systems," 2023,

## 指南方向 10: 基于 AI 的动态无线环境下的多种 RRM 决策任务

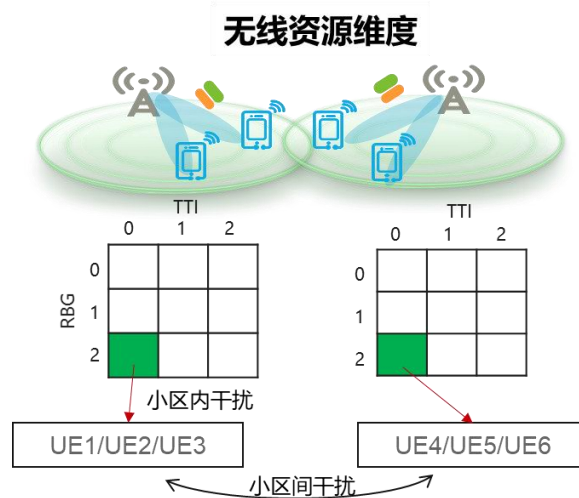
### 1. 背景

通过设计 RRM 智能化架构与算法, 对时频空功无线资源进行最优分配, 同时决策调度用户的 Mcs、Rank、预编码权重, 来实现最优的全维度资源决策, 在一定能耗下最大化整网体验。

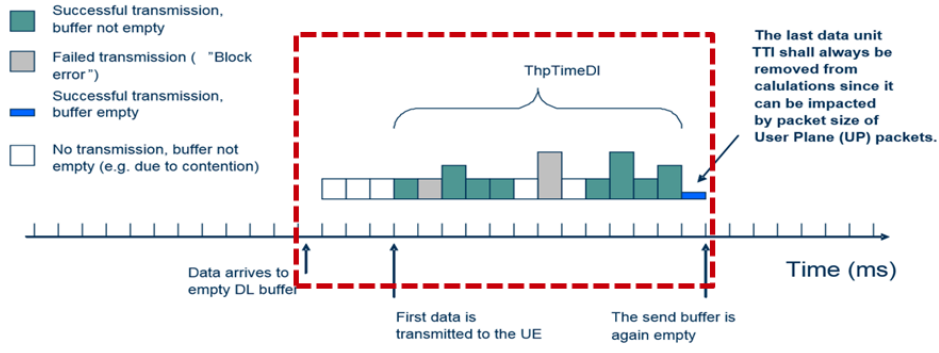
### 2. 现状

无线资源如图所示, 可以分为如下:

- 时频资源: 每个小区都有一定量的时频资源。即上图示中的一个小方格代表最小的时频调度单元。
- 空间资源: 每个小区存在多个波束, 不同波束的用户可使用相同的时频资源 (产生小区内干扰), 即MU配对。
- 功率资源: 功率可以分配给不同的时频资源, 也可以分配给同一时频资源中的不同用户。



用户体验定义如下



$$\text{ThpVolDI} = \sum \text{ (kbits)}$$

$$\text{Total DL transferred volume} = \sum \text{ (kbits)}$$

参考: 3GPP TS32.450

$$\text{Throughput in DL} = \text{ThpVolDI} / \text{ThpTimeDI} \text{ (kbits/s)}$$

目标：通过合理时频空功多维资源编排，从而提升用户体验速率。

基线求解思路：

- 1) 确定每个用户在 RBG 上的得分，基线算法采用 EPF 计算优先级，作为得分。
- 2) 如何调度？按照 RBG 选用户或者用户选 RBG 的策略选出得分最大的用户调度。
- 3) 若每个 RBG 可以承载多个用户，基线按照空分谱效率最大化进行调度

### 3. 问题描述

问题的数学形式表征如下：

$$\begin{aligned} & \text{感知时间最小} \quad \min_{P, X, V} \|Y\|_1 \\ & \text{s. t.} \\ & \text{感知时间} \quad Y = \text{sign}(Q - M) \quad Y \in \{0, 1\}^{N_2 \times N_4} \\ & \text{残留缓存} \quad Q_{:,t+1} = Q_{:,t} - M_{:,t} + \lambda_{:,t} \quad Q; M; \lambda \in R^{N_2 \times N_4} \\ & \text{调度数据} \quad M = g(R) \quad R; P \in R^{N_1 \times N_2 \times N_3 \times N_4 \times N_6} \\ & \text{信噪比tensor} \quad R = f(H, P, X, V) \quad X \in \{0, 1\}^{N_1 \times N_2 \times N_3 \times N_4 \times N_6} \\ & \text{能耗约束} \quad h(P, X, V) \leq \delta_1 \mathbf{1}_{N_1 \times N_4} \quad H; V \in \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4 \times N_5 \times N_6} \\ & \text{每RB功率约束} \quad P_{XV} \cdot_{2,6} \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6} \leq \delta_2 \mathbf{1}_{N_1 \times N_3 \times N_4} \\ & \text{每小区功率约束} \quad P_{XV} \cdot_{2,3,6} \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6} \leq \delta_3 \mathbf{1}_{N_1 \times N_4} \\ & \text{功率时域约束} \quad P_{XV} \cdot_{2,3,4,6} \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6} \leq \delta_3 \mathbf{1}_{N_1} \\ & \quad P_X = P \odot X \quad P_{XV} = P \odot X \odot \|V \cdot_5 V\| \end{aligned}$$

其中， $f(\cdot)$  表示如下：

$$\begin{aligned} & \text{信干噪比关系} \quad f(H, P, X, V) = A/C + \theta(Mcs_{\text{最新上报}}) \\ & \text{信干噪比分子} \quad A = P_X \odot \|H \cdot_5 V\|^2 \quad \theta \text{为不确定参数, 受测量H不准或接收机影响, 不表示加性影响} \\ & \quad H_{(i)} = H_{:,i,::,::} \times_2 \mathbf{1}_{1 \times N_2} \quad H_{:,i,::,::} \in \mathbb{C}^{N_1 \times 1 \times N_3 \times N_4 \times N_5 \times N_6} \\ & \quad \quad \quad \quad \quad \quad \quad \quad H_{(i)} \in \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4 \times N_5 \times N_6} \\ & \quad B = (P_X \odot \|H_{(i)} \cdot_5 V\|^2) \cdot_{12} \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6} \in \mathbb{C}^{N_3 \times N_4} \\ & \text{信干噪比分母} \quad C = \mathbf{1}_{N_1 \times N_2} \circ B - A + \sigma^2 \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6} \end{aligned}$$

$h(\cdot)$ 表示如下：

$$h(\mathbf{P}, \mathbf{X}, \mathbf{V}) = P_{static} \mathbf{D} + \varepsilon \mathbf{P}_{XV} \cdot_{2,3,6} \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6}$$

有效天线数  $d_{k,t} = \|\mathbf{V}_{k,:,:,t,:} \cdot_{2,3,6} \mathbf{V}_{k,:,:,t,:}\|_0$   $\mathbf{D} = \text{tensor}(d_{k,t}) \in R^{N_1 \times N_4}$

$g(\cdot)$ 表示如下:

1、计算  $10 \log_{10} \left( \frac{R \cdot_{3,6} \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6}}{X \cdot_{3,6} \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6}} \right)$  作为用户的 SINR

2、将 SINR 按照如下量化为 Mcs

SINR (db)	-6.55	-4.51	-2.8	-0.84	0.98	2.63	4.69	5.59	6.53	7.5
Mcs	0	1	2	3	4	5	6	7	8	9
SINR (db)	8.38	8.92	10.26	11.11	12.03	12.91	13.94	14.96	15.91	16.93
Mcs	10	11	12	13	14	15	16	17	18	19
SINR (db)	18.02	18.93	19.56	20.46	21.45	22.32	23.74	24.54	25.43	
Mcs	20	21	22	23	24	25	26	27	28	

3、将 Mcs 映射到数据量

$$\mathbf{M} = \{144 * 16 * \text{eff} * (\mathbf{X} \cdot_{3,6} \mathbf{1}_{N_1 \times N_2 \times N_3 \times N_4 \times N_6})\} \cdot \mathbf{1}_{N_1 \times N_2 \times N_4}$$

其中, Mcs 到 eff 的映射如下:

Mcs	0	1	2	3	4	5	6	7	8	9
eff	0.1523	0.2344	0.377	0.6016	0.877	1.1758	1.4766	1.6953	1.9141	2.1602
Mcs	10	11	12	13	14	15	16	17	18	19
eff	2.4063	2.5703	2.7305	3.0293	3.3223	3.6094	3.9023	4.2129	4.5234	4.8164
Mcs	20	21	22	23	24	25	26	27	28	
eff	5.1152	5.332	5.5547	5.8906	6.2266	6.5703	6.9141	7.1602	7.4063	

相关运算符定义如下:

$\mathbf{Q}_{:,t+1}$  中的  $:$ , 表示  $\mathbf{Q}$  的该维 (第 1 维) 所有元素不受影响。

$$\mathbf{A} \cdot_i \mathbf{B} \text{ 张量乘 } \mathbf{A} \cdot_i \mathbf{B} = \sum_{i=1}^{N_i} (\mathbf{A})_{:,i, :, :, :} \odot (\mathbf{B})_{:,i, :, :, :}$$

$\odot$  点乘  $\times_n$   $n$  模乘  $\circ$  外积  
各变量含义与维度说明如下:

参数维度说明:

$N_1 = 36$  ——表示小区数

$N_2 = 72$  ——表示待调度用户数

$N_3 = 17$  ——表示每小区 RBG 数

$N_4 = 100$  ——表示考虑的 slot 数

$N_5 = 32$  ——表示天线数

$N_6 = 2$  ——表示用户的流数

$\mathbf{Y} \in \{0,1\}^{N_2 \times N_4}$  —— $y_{i,t}$  表示第  $i$  个用户在第  $t$  个 TTI 上是否有数据

$\mathbf{Q} \in R^{N_2 \times N_4}$  —— $Q_{i,t}$  表示第  $i$  个用户在第  $t$  个 TTI 上的残留缓存

$\mathbf{M} \in R^{N_2 \times N_4}$  —— $M_{i,t}$  表示第  $i$  个用户在第  $t$  个 TTI 上的调度数据

$\lambda \in R^{N_2 \times N_4}$  —— $\lambda_{i,t}$  表示第  $i$  个用户在第  $t$  个 TTI 上的新来数据

$\mathbf{R} \in R^{N_1 \times N_2 \times N_3 \times N_4 \times N_6}$  ——其元素表示第  $k$  个小区, 第  $i$  个用户, 第  $b$  个 RBG, 第  $t$  个 TTI, 第  $l$  流的信干噪比

$\mathbf{P} \in R^{N_1 \times N_2 \times N_3 \times N_4 \times N_6}$  ——其元素表示第  $k$  个小区, 第  $i$  个用户, 第  $b$  个 RBG, 第  $t$  个 TTI, 第  $l$  流的发射功率

$\mathbf{X} \in \{0,1\}^{N_1 \times N_2 \times N_3 \times N_4 \times N_6}$  ——其元素表示第  $k$  个小区, 第  $i$  个用户, 第  $b$  个 RBG, 第  $t$  个 TTI, 第  $l$  流是否分配资源

$\mathbf{H} \in \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4 \times N_5 \times N_6}$  ——其元素表示第  $k$  个小区, 第  $i$  个用户, 第  $b$  个 RBG, 第  $t$  个 TTI, 第  $a$  根天线, 第  $l$  流的信道衰落因子

$\mathbf{V} \in \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4 \times N_5 \times N_6}$  ——其元素表示第  $k$  个小区, 第  $i$  个用户, 第  $b$  个 RBG, 第  $t$  个 TTI, 第  $a$  根天线, 第  $l$  流的加权因子

$\theta \in R^{N_1 \times N_2 \times N_3 \times N_4 \times N_6}$  ——其元素表示第  $k$  个小区, 第  $i$  个用户, 第  $b$  个 RBG, 第  $t$  个 TTI, 第  $l$  流的信噪比误差

$\mathbf{D} \in R^{N_1 \times N_4}$  —— $d_{i,t}$  表示第  $k$  个小区在第  $t$  个 TTI 上的有效天线数

关键挑战:

- 信息不全或者建模不准: AI 监督训练标签难以获取。信道信息不准确、终端接收机解调难以建模, 利用最优化难以求解获取标签。如何与环境交互, 改进建模提升最终的网络性能; 特别是  $f(\cdot)$  和  $g(\cdot)$  建模不准。

- a) 信道信息不准确等因素致使优化难以建模，求解“南辕北辙”。当权值精度或者干扰精度不准时，优化反而带来更多干扰，导致性能下降
- b) 引入MU配对、功率分配、MCS、RANK以及多小区协同后，对谱效的建模公式难以刻画真实的收益
- c) AMC策略：当前的OLLA合一策略节省内存，牺牲了部分性能
- d) 其它因素：基站估算的配对干扰与接收端的实际解调存在偏差
- 分布式计算：多小区信息交互有限且存在交互时延，如何设计低通信开销的分布式算法；
- 高维约束：AI如何处理高维约束，保证输出在可行域范围内；
- 高决策空间：对多小区的时频空功等无线资源进行最优分配，同时决策调度用户的Mcs、Rank、预编码权重等，决策空间维度爆炸，传统AI收敛慢，甚至难以收敛；
- 高算力诉求：求解复杂度或者算力要求，随着求解小区或者用户数增长，计算复杂度也随之增长，要求在产品算力下的时间复杂度在300us以内。

#### 4. 目标

- 目标：在满足能耗不超过门限的前提下，最大化整网用户体验。
- 输入：一段时间采样，每个用户在每个小区的每个RBG的估计信道H，以及Mcs，模块静态动态功耗、每个小区的带宽。
- 输出：每个用户在每个小区的每个RBG是否占用以及功率，以及用户在这个小区的Mcs，天线关断数，时域关断数、预编码权重。
- 约束：模块的功率约束、资源约束、数据量约束。
- 不确定性信息：按照决策Mcs调度，TB块会有不同概率发生错误。
- 诉求：时间复杂度在300us@6频，每频6小区，每频17个RBG，总共72用户/4核CPU@2.2G。

### 指南方向 11：AI 使能 HBF 架构下全量实时信道信息获取

#### 1. 背景

MIMO (Multi-Input Multi-Output) 技术通过空间波束成形和空分复用，能极大提升无线通信系统性能和能效。随着 MIMO 天线数量的增多，系统成本也将大幅抬升，更好的选择是采用模数混合波束成形架构 (HBF, Hybrid BeamForming)，具体如下图所示，

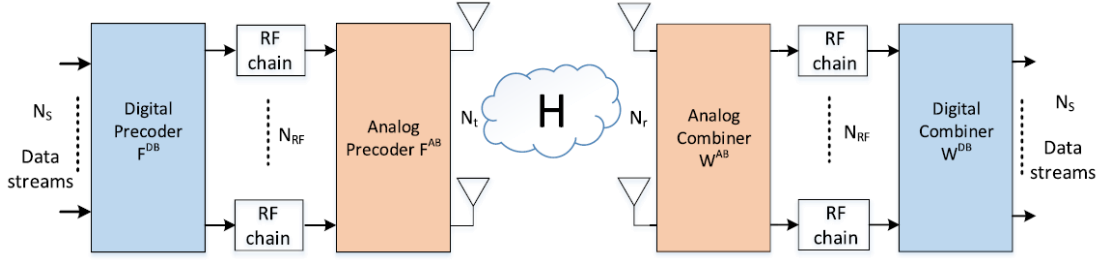


图1 HBF系统收发示意图[1]

其中 $N_t$ 和 $N_r$ 分别表示发端和收端的天线数， $N_{RF}$ 表示对应的数字通道或RF chain数，一般地， $N_t/N_r > N_{RF}$ 。 $F^{DB}/F^{AB}$ 和 $W^{DB}/W^{AB}$ 分别表示发端和收端的数字预编码权值和模拟预编码权值，通常模拟预编码权值仅具备相位调节能力，数字预编码权值可以同时进行幅度和相位调节。该课题研究在HBF系统架构下，如何高效获取全量的原始空口信道信息 $\mathbf{H}$ 。

## 2. 现状

如下图2所示，现有通信系统中一般采用两种方式来完成空口信道信息 $\mathbf{H}$ 的测量，其一是通过终端发送SRS测量信号，基站直接完成上行信道信息 $\mathbf{H}_{UL}$ 测量；其二是基站发送CSI-RS导频，终端首先完成下行信道信息 $\mathbf{H}_{DL}$ 测量，然后将 $\mathbf{H}_{DL}$ 量化处理后再反馈回基站。对于TDD系统而言，相同时刻下满足 $\mathbf{H}_{UL} = \mathbf{H}_{DL}^H$ 。为了完成基站覆盖范围内的全量测量，模拟预编码采用DFT（Discrete Fourier Transform）预编码权值对所覆盖范围进行依次扫描，称为Analog beam scan。数字预编采用单位矩阵，直接获得原始信道测量量。该方法测量开销巨大，对所有用户以及需要测量的位置（主要指时域和频域），均需要分配相应的测量资源。实际系统仅能实现局部的稀疏测量，缺省部分则通过邻近测量结果保持获得，但是精度大大降低，从而显著降低了系统性能。如何通过少量的稀疏测量来恢复全量高精信道信息一直是研究热点，有待突破。

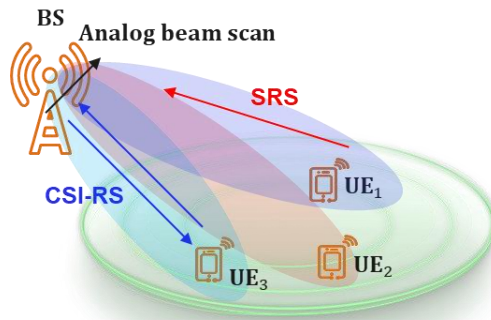


图2 HBF系统中信道测量示意图

## 3. 问题描述

上述问题，可以抽象提炼为如下的数学问题，已知 $\mathbf{X} \in \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4}$ 为4维的复数张量，对于任意的正整数 $n_1 \in [1, N_1]$ ， $n_2 \in [1, N_2]$ ，可获得对 $\mathbf{X}(n_1, n_2, :, :)$ 的观测结果如下，

$$\mathbf{Y}(n_1, n_2, :, :) = \mathbf{W}(n_1, :, :)\mathbf{X}(n_1, n_2, :, :)\mathbf{V}(n_1, :, :) + \mathbf{N}(n_1, n_2, :, :) \in \mathbb{C}^{M_3 \times M_4}$$

其中 $\mathbf{N} \in \mathbb{C}^{N_1 \times N_2 \times M_3 \times M_4}$ ，表示独立的复高斯白噪声，即对于任意正整数 $n_1 \in [1, N_1]$ ， $n_2 \in [1, N_2]$ ， $n_3 \in [1, M_3]$ ， $n_4 \in [1, M_4]$ ，满足 $\mathbf{N}(n_1, n_2, n_3, n_4) \sim \mathcal{CN}(0, \sigma_n^2)$ 。 $\mathbf{W}(n_1, :, :) \in \mathbb{C}^{M_3 \times N_3}$ 和 $\mathbf{V}(n_1, :, :) \in \mathbb{C}^{N_4 \times M_4}$ 分别表示两个独立的降维矩阵，这里有 $M_3 < N_3$ 和 $N_4 > M_4$ ，当 $n_1$ 给定时，



对于不同的 $n_2$ ,  $\mathbf{X}(n_1, n_2, :, :)$ 将共享相同的降维权值 $\mathbf{W}(n_1, :, :)$ 和 $\mathbf{V}(n_1, :, :)$ , 并且对于 $\mathbf{W}$ 和 $\mathbf{V}$ 中的任意元素, 满足 $|\mathbf{W}(i, j, k)| = 1$ 和 $|\mathbf{V}(i, j, k)| = 1$ 。此外, 对于 $(n_1, n_2) \in \Omega_1$ , 当 $n_1$ 给定时存在某个压缩量化函数 $g: \mathbb{C}^{|\Omega_1, n_2| \times M_3 \times M_4} \rightarrow \mathcal{Q}(n_1, :)$ ,  $\mathcal{Q}$ 表示长度为 $|\mathcal{Q}(n_1, :)|$ 的 0/1 比特序列。

研究需求描述如下:

1. 设计集合 $(n_1, n_2) \in \Omega_1$ , 相应的 $\mathbf{W}(n_1, :, :)$ 和 $\mathbf{V}(n_1, :, :)$ 降维矩阵集, 记为,

$$\theta_1 = \left\{ \Omega_1 = \{(n_1, n_2)\}, \{\mathbf{W}(n_1, :, :), \mathbf{V}(n_1, :, :)\}_{(n_1, :)\in\Omega_1} \right\}$$

以及压缩量化函数 $g: \mathbb{C}^{|\Omega_1, n_2| \times M_3 \times M_4} \rightarrow \mathcal{Q}(n_1, :)$ , 有,

$$\mathcal{Q} = \left\{ g(\{\mathbf{Y}(n_1, n_2, :, :)\}_{(n_1, n_2)\in\Omega_1}) \right\}_{(n_1, :)\in\Omega_1}$$

2. 设计集合 $(n_1, n_2) \in \Omega_2$ , 相应的 $\mathbf{W}(n_1, :, :)$ 和 $\mathbf{V}(n_1, :, :)$ 降维矩阵集, 记为,

$$\theta_2 = \left\{ \Omega_2 = \{(n_1, n_2)\}, \{\mathbf{W}(n_1, :, :), \mathbf{V}(n_1, :, :)\}_{(n_1, :)\in\Omega_2} \right\}$$

相应的观测结果为,

$$\mathcal{O} = \{\mathbf{Y}(n_1, n_2, :, :)\}_{(n_1, n_2)\in\Omega_2}$$

3. 设计映射函数 $f: \left\{ \mathbb{C}^{|\Omega_2, n_1| \times |\Omega_2, n_2| \times M_3 \times M_4}, \{0, 1\}^{|\mathcal{Q}|} \right\} \rightarrow \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4}$ , 记映射结果为,

$$\hat{\mathbf{X}} = f(\mathcal{O}, \mathcal{Q})$$

4. 基于1/2/3的设计, 求解如下目标问题,

$$\min_{\theta_1, \theta_2, g, f} \|\mathbf{X} - \hat{\mathbf{X}}\|^2 + (\alpha \cdot |\Omega_1| + \beta \cdot |\Omega_2|) + \gamma \cdot |\mathcal{Q}|$$

$$s. t. \hat{\mathbf{X}} = f(\mathcal{O}, \mathcal{Q}), f: \left\{ \mathbb{C}^{|\Omega_2, n_1| \times |\Omega_2, n_2| \times M_3 \times M_4}, \{0, 1\}^{|\mathcal{Q}|} \right\} \rightarrow \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4}$$

$$\mathcal{O} = \{\mathbf{Y}(n_1, n_2, :, :)\}_{(n_1, n_2)\in\Omega_2} \in \mathbb{C}^{|\Omega_2, n_1| \times |\Omega_2, n_2| \times M_3 \times M_4}$$

$$\mathcal{Q} = \left\{ g(\{\mathbf{Y}(n_1, n_2, :, :)\}_{(n_1, n_2)\in\Omega_1}) \right\}_{(n_1, :)\in\Omega_1}, g: \mathbb{C}^{|\Omega_1, n_2| \times M_3 \times M_4} \rightarrow \{0, 1\}^{|\mathcal{Q}(n_1, :)|}$$

$$\forall (n_1, n_2), \mathbf{Y}(n_1, n_2, :, :) = \mathbf{W}(n_1, :, :)\mathbf{X}(n_1, n_2, :, :)\mathbf{V}(n_1, :, :) + \mathbf{N}(n_1, n_2, :, :) \in \mathbb{C}^{M_3 \times M_4}$$

$$\theta_1 = \left\{ \Omega_1 = \{(n_1, n_2)\}, \{\mathbf{W}(n_1, :, :), \mathbf{V}(n_1, :, :)\}_{(n_1, :)\in\Omega_1} \right\}$$

$$\theta_2 = \left\{ \Omega_2 = \{(n_1, n_2)\}, \{\mathbf{W}(n_1, :, :), \mathbf{V}(n_1, :, :)\}_{(n_1, :)\in\Omega_2} \right\}$$

$$\forall (i, j, k), |\mathbf{W}(i, j, k)| = |\mathbf{V}(i, j, k)| = 1$$

其中 $\alpha, \beta, \gamma$ 为给定的标量权重系数。

5. 在问题4的基础上, 假设参数集合 $\theta_1$ 和 $\theta_2$ 给定, 得到如下子问题1,

$$\min_{g, f} \|\mathbf{X} - \hat{\mathbf{X}}\|^2 + \gamma \cdot |\mathcal{Q}|$$

$$s. t. \hat{\mathbf{X}} = f(\mathcal{O}, \mathcal{Q}), f: \left\{ \mathbb{C}^{|\Omega_2, n_1| \times |\Omega_2, n_2| \times M_3 \times M_4}, \{0, 1\}^{|\mathcal{Q}|} \right\} \rightarrow \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4}$$

$$\mathcal{O} = \{\mathbf{Y}(n_1, n_2, :, :)\}_{(n_1, n_2)\in\Omega_2} \in \mathbb{C}^{|\Omega_2, n_1| \times |\Omega_2, n_2| \times M_3 \times M_4}$$

$$\mathcal{Q} = \{g(\{\mathbf{Y}(n_1, n_2, :, :)\}_{(n_1, n_2) \in \Omega_1})\}_{(n_1, :)\in\Omega_1}, g: \mathbb{C}^{|\Omega_1, n_2| \times M_3 \times M_4} \rightarrow \{0, 1\}^{|\mathcal{Q}(n_1, :)|}$$

$$\forall (n_1, n_2), \mathbf{Y}(n_1, n_2, :, :) = \mathbf{W}(n_1, :, :)\mathbf{X}(n_1, n_2, :, :)\mathbf{V}(n_1, :, :) + \mathbf{N}(n_1, n_2, :, :) \in \mathbb{C}^{M_3 \times M_4}$$

6. 在问题5的基础上, 假设  $g: \mathbb{C}^{|\Omega_1, n_2| \times M_3 \times M_4} \rightarrow \{0, 1\}^{|\mathcal{Q}(n_1, :)|}$  给定, 得到如下子问题2,

$$\min_f \|\mathbf{X} - \hat{\mathbf{X}}\|^2$$

$$s.t. \hat{\mathbf{X}} = f(\mathcal{O}, \mathcal{Q}), f: \{\mathbb{C}^{|\Omega_2, n_1| \times |\Omega_2, n_2| \times M_3 \times M_4}, \{0, 1\}^{|\mathcal{Q}|}\} \rightarrow \mathbb{C}^{N_1 \times N_2 \times N_3 \times N_4}$$

$$\mathcal{O} = \{\mathbf{Y}(n_1, n_2, :, :)\}_{(n_1, n_2) \in \Omega_2} \in \mathbb{C}^{|\Omega_2, n_1| \times |\Omega_2, n_2| \times M_3 \times M_4}$$

$$\mathcal{Q} = \{g(\{\mathbf{Y}(n_1, n_2, :, :)\}_{(n_1, n_2) \in \Omega_1})\}_{(n_1, :)\in\Omega_1}, g: \mathbb{C}^{|\Omega_1, n_2| \times M_3 \times M_4} \rightarrow \{0, 1\}^{|\mathcal{Q}(n_1, :)|}$$

$$\forall (n_1, n_2), \mathbf{Y}(n_1, n_2, :, :) = \mathbf{W}(n_1, :, :)\mathbf{X}(n_1, n_2, :, :)\mathbf{V}(n_1, :, :) + \mathbf{N}(n_1, n_2, :, :) \in \mathbb{C}^{M_3 \times M_4}$$

附录: 各参数的典型取值范围如下表所示

表 1. 各参数典型取值范围

参数	取值范围
$N_1$	1~100
$N_2$	1000~5000
$N_3$	100~10000
$N_4$	1~100
$M_3$	10~1000
$M_4$	1~10

## f) 目标

### 理论目标

- (1) 从信息可恢复角度, 理论分析给定恢复精度下需要的最小测量开销。
- (2) 从 CSI-RS 和 SRS 融合测量角度, 分别对应测量  $\mathcal{Q}$  和  $\mathcal{O}$ , 当给定各自的代价权重  $\alpha$  和  $\beta$  时, 理论分析最优的组合测量策略。

### b)、应用目标:

- (1) 在稀疏测量  $\mathcal{Q}$  对应的测量比例约为 1%, 稀疏测量  $\mathcal{O}$  对应的测量比例约为 5%, 且每次测量反馈比特数  $< 100$  时, 全量信道信息恢复精度  $\text{NMSE} < -10\text{dB}$ 。

### 参考文献

- [1] Ahmed I, Khammari H, Shahid A, et al. A survey on hybrid beamforming techniques in 5G: Architecture and system model perspectives[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 3060–3097.

## 指南方向 12: Score-based 扩散模型采样加速

### 1. 背景

当前基于 score-based 的扩散模型在生成领域展现了惊人的潜力。然而，此类模型依赖于扩散生成，通常需要 >50 步的采样来达到满意的效果，这带来了较大的推理代价。因此针对扩散生成的采样加速算法是一个非常重要的研究课题。

### 2. 现状

基于扩散生成的过程，可以表达为<sup>[1]</sup>

$$dx = f(x; t) - \eta_t \nabla_x \log p_t(x) \approx f(x; t) - \eta_t G_\theta(x; t) \quad (1)$$

其中  $\nabla_x \log p_t(x)$  是真实扩散中间态中的似然函数梯度；由于真实数据分布不可获得，因此往往通过训练  $G_\theta(x; t)$  来拟合  $\nabla_x \log p_t(x)$  进行近似。扩散过程 (1) 式可以实现从高斯数据  $p_0(x)$  ( $t = 0$ ) 到真实数据分布  $p_1(x)$  ( $t = 1$ ) 之间的概率分布迁移，实现对真实数据分布的采样生成。

当前采样加速主要有几个思路。一个重要的方向是设计不同的采样器，如 iPNDM, DPM-Solver；另一个方向是优化采样间隔，如 AYS, AMED。

### 3. 问题描述

针对 (1)，假设  $f(x; t) = \alpha_t x + \beta_t t$ ， $G_\theta(x; t)$  是个深度神经网络，给定离散采样次数  $N$  (通常在  $2 \sim 100$  步)。输出优化后的采样策略，包括采样器、采样间隔。

### 4. 目标

1) 采样器优化：相比 DPM-Solver 系列，采样步数减少 25% 以上。生成质量 (参考 Clip Score, FID, Hpsv2, 人工评测等) 不劣化。

2) 采样间隔优化：输出优化后的采样间隔策略：

$$t_i = \text{schedule}(i)$$

使得最终的扩散误差： $D := E_{x_0 \sim p_0} \|x_1^{\text{discrete}} - x_1^{\text{continuous}}\|^2$  最优，其中  $x_1^{\text{discrete}}$  是使用优化采样间隔输出的最终结果， $x_1^{\text{continuous}}$  是 (1) 是连续采样获得的结果 (通常使用  $N=1000$  模拟得到)。

两个目标可选择任一达成。对于目标 2，需要限制对  $G_\theta(x; t)$  采样数控制在 10k 个样本以内。具体的网络选择可以参考 SD1.5-XL。

#### • 参考文献

- [1] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, Ben

Poole. Score-Based Generative Modeling through Stochastic Differential Equations

## 指南方向 13: 大规模复数稀疏矩阵的加速迭代求解算法

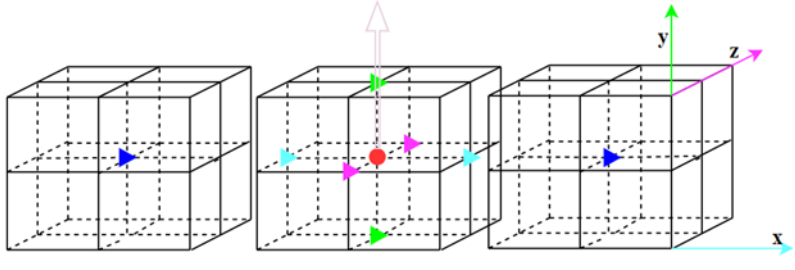
### 1. 背景

很多仿真问题中都需要求解大规模线性方程组，有一类线性方程组是通过 Wilson-

Dirac 离散方法得到的 4 维大规模复数稀疏线性方程组  $Ma = b$ ,  $M$  是大规模复数稀疏对称矩阵且  $M$  中的每个非零元不是一个浮点数, 而是一个小规模复数矩阵, 也就是对于一个  $M$  算例  $(x, y, z, t)$  中的每一个位置  $n$  处的  $a(n)$  都要求解方程  $M(n)a(n) = b(n)$ , 其中  $M(n)$  是由 4 个维度各 2 个方向的  $u_\mu$  和对应的  $\gamma_\mu$  做克罗内克积得到, 具体求解见下图计算公式所示;

这类线性方程组的求解在量子色动力学 (QCD) 中尤为常见, 对于一个  $M$  算例 (32, 32, 32, 8), 每一个格点的复数矩阵规模为  $12 \times 12$ , 求解这类大规模线性方程组是典型的内存受限型问题, 针对矩阵  $M$  的对称性、前沿数值算法和硬件平台特性都还有巨大的优化空间;

目前求解该问题常用迭代求解方法有 BiCGStab、Multigrid 等。对一个典型的算例, 其系数矩阵的维数一般在  $O(10^9)$  量级, 对计算资源需求很大, 如何加速该类型方程组的求解迫在眉睫。

$$u_\mu(n) = \begin{bmatrix} u_{\mu,1,1} & u_{\mu,1,2} & u_{\mu,1,3} \\ u_{\mu,2,1} & u_{\mu,2,2} & u_{\mu,2,3} \\ u_{\mu,3,1} & u_{\mu,3,2} & u_{\mu,3,3} \end{bmatrix} \in \mathbb{C}^{3 \times 3}, a(n) = \begin{bmatrix} a_{1,1} \\ \vdots \\ a_{12,1} \end{bmatrix}, b(n) = \begin{bmatrix} b_{1,1} \\ \vdots \\ b_{12,1} \end{bmatrix}$$


$$b(n) = M(n)a(n), M(n) \propto -\frac{1}{2} \sum_{\mu=\pm 1}^4 (I - \gamma_\mu) \otimes u_\mu a(n \pm \mu)$$

## 2. 现状

该场景的迭代求解算法有如下难点:

- 1) **方程结构特殊:** 矩阵  $M$  非显性, 源项  $b$  只有一个位置为  $1+0j$ , 其余位置为  $0+0j$ ;
- 2) **迭代耗时长:** 当  $M$  的特征值越小时候, BiCGStab 等单网格算法迭代不收敛会愈加显著 (临界降速问题), 导致了该类算法的迭代耗时长, 成为性能瓶颈;
- 3) **预处理选择:** 针对不同算例需要选择合适的预处理方式来加速迭代算法, 同时预处理本身会花费大量时间, 需针对本问题进行特殊设计。

当前有数值和 AI 两种方案:

- **使用预处理后的 BiCGStab:** 目前在 (32,32,32,8) 算例上使用预处理后的 BiCGStab 算法计算 12 组源需要迭代 240 步左右, 耗时 216s 左右 (单 Arm 节点);
- **使用 AI 方法作为迭代初值:** 使用卷积类 AI 模型对给定  $b$  和  $U$  预测模范围为 (1e-3, 1) 的  $a$ , 迹的相对误差  $< 1e-2$ ; 进一步将预测结果作为 BiCGStab 迭代初值, 迭代步数下降

9%（单Arm节点），离预期效果仍有差距。

### 3. 问题描述

构造高效且泛化性强的迭代方案：基于给定数据提供高效、泛化性强的迭代方案，包括但不限于数值算法、AI 技术等，性能超越现有方案

### 4. 目标

- a) **计算精度达标**：替代方案在给定数据上验证，计算结果的迹的相对误差 $<1e-2$ ；
- b) **减少迭代耗时**：与BiCGStable法迭代的数值方式比较，迭代耗时减少30%；
- c) **高能物理场景验证**：将构造方案应用于量子色动力学（QCD）验证效果，期望迭代耗时较原始减少30%。

#### • 参考文献

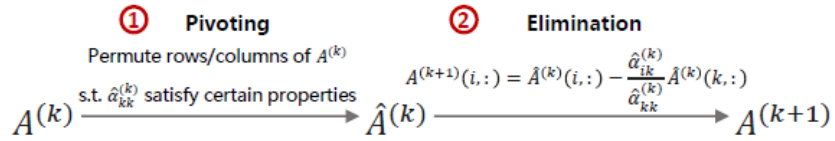
- [1] 孙玮, 毕玉江, 程耀东. ARM 处理器上的格点 QCD 计算与优化[J]. 计算机科学, 2023, 50(6): 52-57. <https://doi.org/10.11896/jsjcx.230200159>
- [2] 陈莹, 丁亨通, 冯旭, 等. 格点量子色动力学在中国[J]. 现代物理知识, 2020, 32(1): 36-44.

## 指南方向 14: 通信避免的 LU 分解 (CALU) 的数值稳定性分析

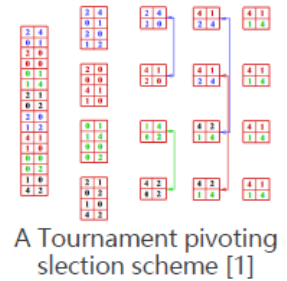
### 1. 背景

线性方程组的求解是数值线性代数的核心，稠密求解器在工业应用中得到了广泛的应用。选主元是数值稳定的稠密线性代数求解器的关键步骤之一。部分选主元高斯消去法(GEPP)是 LU 分解中最稳定、应用最广泛的算法之一。然而，对于使用集群求解器的大规模应用程序，与部分选主元相关的通信成本是一个日益严重的瓶颈。CALU (Communication Avoiding LU factorization, 通信避免的 LU 分解) 算法就是为了解决这一通信瓶颈问题而提出的。它主要使用了一个通信避免的选主元方案来减少通信开销。

At the k-th step, Gaussian Elimination with a pivoting scheme can be expressed as follows:



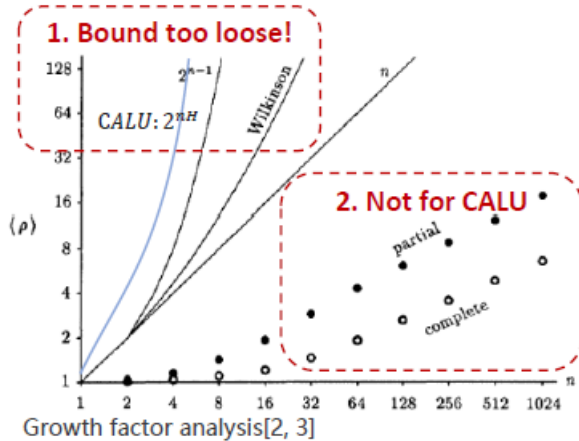
b steps of gaussian elimination (in a blocked LU algorithm)	
Partial Pivoting	CA/tournament pivoting
1. Permute a row in $A^{(k)}$ s.t. $\hat{\alpha}_{kk}^{(k)}$ is the largest element in its column 2. Perform elimination 3. Repeat 1-2 for b times	1. Select b rows using a selection scheme (details in [1]) 2. Permute these b rows to the top b rows to form $\hat{A}^{(k)}$ 3. Perform elimination w/o pivoting for b times



## 2. 现状

Wilkinson proved that the relative error of solving  $Ax=b$ , where  $A$  is factorized with LU, is related to the growth factor [2]:

$$\frac{\|\bar{x} - x\|_{\infty}}{\|x\|_{\infty}} \leq 4n^2 \kappa_{\infty}(A) \rho \epsilon \quad \text{where} \quad \rho = \frac{\max_{i,j,k} |\alpha_{ij}^{(k)}|}{\max_{i,j} |\alpha_{ij}|} \quad \begin{matrix} \epsilon \text{ is the machine precision} \\ \kappa_{\infty}(A) \text{ is the condition of } A \end{matrix}$$



1. Loose bound [3] exists for CALU and partial pivoting but it is too loose to justify the practical usage of CALU.
2. Average case/probabilistic analysis [2] for GEPP shows sublinear growth factor, but no such study exists for CALU.

## 3. 问题描述

CALU 的数值稳定性分析并不成熟。无法在所场景采用 CALU 安全地替代 GEPP, i. e. GETRF。

## 4. 目标

- 1) 通过理论稳定性分析证明 CALU 的增长因子 (growth factor) 是次线性的, 同时给出其与传统 GEPP 算法的对比分析。给出增长因子与算法参数, 例如 CALU 算法中的参数 H 等, 的参数化理论分析。
- 2) 自适应通信避免算法: 给出一个可以在 CALU 与部分选主元算法之间平滑过渡的自适应算法, 其稳定性及通信开销需要介于 CALU 与 GEPP 之间。

### • 参考文献

- [1] Grigori, Laura, James W. Demmel, and Hua Xiang. "Communication avoiding Gaussian elimination."

SC'08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing. IEEE, 2008.

- [2] Trefethen, Lloyd N., and Robert S. Schreiber. "Average case stability of Gaussian elimination." SIAM Journal on Matrix Analysis and Applications 11.3 (1990): 335-360.
- [3] Grigori, Laura, James W. Demmel, and Hua Xiang. "CALU: a communication optimal LU factorization algorithm." SIAM Journal on Matrix Analysis and Applications 32.4 (2011): 1317-1350.

## 致谢

衷心感谢在指南编制过程中给予指导和支持的专家学者和工作人员，包括：北京大学张继平、陈大岳、胡俊、章志飞、范辉军、董彬、周晓华、文再文、林亦波、范少锋、徐婷、于欣卉，西安电子科技大学郑晓静，山东大学陈增敬、王璐，吉林大学张然、贾继伟，四川大学张旭、杨凡意，南京大学秦厚荣、刘克勤，中国科技大学叶向东、陈仁杰，华中师范大学刘宏伟，兰州大学周又和、王记增，大湾区大学冯锦超、杨斯崑，武汉大学张继伟，复旦大学李颖州，俄罗斯莫斯科国立大学 Andrey Shafarevich、Alexander Ivanov、Alexander Zheglov、Vladimir Bogachev、Andrei Krylov，俄罗斯高等经济大学 Ivan Arzhantsev、Sergei Kuznetsov、Vasilii Gromov、Andrey Delitsyn，俄罗斯科学院索伯列夫数学研究所 Dmitry Tkachev、Adil Yerzin、Yuri Kochetov，俄罗斯国家人工智能发展中心 Evgeny Burnaev 等。