

分类号	<u>TN929.5</u>	密级	<u>公开</u>
UDC	<u>621.39</u>	学位论文编号	<u>D-10617-30852-(2017)-01063</u>

重庆邮电大学硕士学位论文

中文题目	<u>大规模 MIMO 上行链路信号检测</u>
	<u>算法研究</u>
英文题目	<u>Research on Detection Algorithm for</u>
	<u>Uplink Massive MIMO Systems</u>
学 号	<u>S140131067</u>
姓 名	<u>李楠</u>
学位类别	<u>工程硕士</u>
学科专业	<u>电子与通信工程</u>
指导教师	<u>王茜竹 高级工程师</u>
完成日期	<u>2017 年 4 月 7 日</u>

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得重庆邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：李楠

签字日期：2017 年 5 月 26 日

学位论文版权使用授权书

本学位论文作者完全了解 重庆邮电大学 有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 重庆邮电大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名：

李楠

导师签名：

王磊竹

签字日期：2017 年 5 月 26 日

签字日期：2017 年 5 月 28 日

摘要

随着天线数量的增加,大规模多输入多输出(Multiple-Input Multiple-Output, MIMO)系统上行链路信号检测算法的复杂度也大幅增加,原有检测算法无法实际应用在工程项目中。因此,本文研究的内容是大规模 MIMO 系统下复杂度低并且性能好的信号检测算法。

首先,本文重点分析两种适用于大规模 MIMO 系统的信号检测算法:似然上升搜索(Likelihood Ascend Search, LAS)和主动禁忌搜索(Reactive Tabu Search, RTS)算法。RTS 算法凭借着“逃脱”策略,求得全局最优解向量,性能优于 LAS 算法,但存在复杂度过高的问题。随着收发天线数量增加,初始值计算复杂度占整体算法复杂度比例越来越大,因此初始值计算复杂度的降低对于 RTS 算法整体复杂度的降低也是至关重要的。

然后,针对 RTS 算法初始值计算复杂度过高这一问题,本文给出一种基于高斯赛德尔(Gauss-Seidel, GS)迭代法的改进 RTS 检测算法,即通过 GS 迭代法替换原来的 MMSE 算法求解初始解向量。仿真结果验证表明:在收发天线数为 128×16 ,信噪比为 16dB 时,4QAM 调制方式下改进的算法和原有的 RTS 算法相比,误码率性能损失 1.6×10^{-5} 的前提下,初始解向量算法复杂度降低 78.5%。但是随着调制方式阶数的增加,改进的算法和原有 RTS 算法的误码率性能差距越来越大,还需要对 GS-RTS 算法改进,提高其高阶调制方式下误码率性能。

最后,对于 GS-RTS 算法高阶调制方式下误码率性能较差的问题,本文引入块分星座图(Block Constellation, BC)方法求解 GS 迭代法初始解向量,从而加快 GS 迭代法收敛速率,提升 GS 算法性能,进一步提高 GS-RTS 算法整体性能。仿真结果验证表明:在收发天线数为 128×16 ,信噪比为 16dB 时,改进的 BC-GS-RTS 算法和 GS-RTS 算法相比,复杂度增加 12.5%的前提下,收敛速率明显提升,16QAM 调制方式下误码率性能提高 3 倍,得到近似于原有 RTS 算法性能。因此,用户采用高阶调制方式的情况下,改进的 BC-GS-RTS 检测算法可以很好地替代原有的 RTS 信号检测算法。

关键词: 大规模 MIMO, 信号检测, 主动禁忌搜索, 迭代法, 块分星座图

Abstract

With the increase of the number of antennas, the complexity of the uplink signal detection algorithm for Massive MIMO systems increases significantly, the original detection algorithm cannot be applied in reality. Therefore, the focus of this paper is the signal detection algorithm with low complexity and good performance under Massive MIMO systems.

Firstly, this paper focuses on two kinds of Massive MIMO signal detection algorithms: Likelihood Ascend Search (LAS) algorithm and Reactive Tabu Search (RTS) algorithm. The RTS algorithm can obtain the global optimal solution vector based on the "escape" strategy, therefore the performance is better than that of LAS algorithm, but the complexity is higher than that of LAS algorithm. With the increase of the number of transceivers, the initial calculation complexity occupies an increasing proportion of the overall algorithm complexity, therefore the reduction of the initial calculation complexity is also important for the overall complexity of the RTS algorithm.

Secondly, an improved RTS detection algorithm based on Gauss-Seidel (GS) iteration method is proposed, which is to replace the original MMSE algorithm by GS iterative method. The simulation results show that when the number of transmit and receive antennas is 128×16 and the SNR is 16dB, compared with the original RTS algorithm, the complexity of the initial solution vector calculation of the improved algorithm is reduced by 78.5% with 1.6×10^{-5} bit error rate performance loss in 4QAM modulation. However, with the increase of modulation order, the gap of two algorithms BER performance is enlarged, so the GS-RTS algorithm is proposed to improve the BER performance in the high order modulation.

Finally, this thesis introduces the block constellation (BC) method in GS-RTS algorithm for high order modulation. The block constellation method is used to compute the initial solution vector of GS iterative method, which can accelerate the convergence of GS iterative method and improve the performance of GS algorithm, to further improve the overall performance of GS-RTS algorithm. The simulation results show that when the number of transceivers is 128×16 and the SNR is 16dB, compared with the GS-RTS algorithm, the complexity of the improved BC-GS-RTS algorithm is

increased by 12.5%, but the BER performance of the BC-GS-RTS algorithm is improved by 3 times and can be approximated to the original RTS algorithm in 16QAM modulation. Therefore, the improved BC-GS-RTS detection algorithm can well replace the original RTS signal detection algorithm in the high order modulation.

Keywords: Massive MIMO, Signal detection, Reactive tabu search algorithm, Iterative method, Block constellation

<http://www.ixueshu.com>

目录

图录	VI
表录	VII
注释表	VIII
第 1 章 引言	1
1.1 研究背景及意义	1
1.2 大规模 MIMO 检测算法的研究现状	3
1.3 主要研究内容	6
1.4 论文结构安排	6
第 2 章 大规模 MIMO 系统检测算法	8
2.1 大规模 MIMO 系统上行链路信道模型	8
2.2 信道硬化	10
2.3 常规信号检测算法	12
2.3.1 最优检测算法	12
2.3.2 线性检测算法	14
2.3.3 常规算法复杂度和性能分析	16
2.4 基于本地邻域搜索算法	18
2.4.1 LAS 算法原理	19
2.4.2 LAS 算法缺陷分析	21
2.4.3 LAS 算法复杂度分析	21
2.4.4 LAS 算法性能仿真	21
2.5 本章小结	23
第 3 章 基于 GS 迭代法的改进 RTS 检测算法	24
3.1 RTS 算法	24
3.1.1 RTS 算法原理	24
3.1.2 RTS 算法和 LAS 算法对比	30
3.2 GS-RTS 算法	32

3.2.1 GS 算法	32
3.2.2 GS-RTS 算法流程	35
3.3 仿真与分析	36
3.3.1 GS-RTS 算法性能仿真	36
3.3.2 GS-RTS 算法复杂度分析	40
3.3.3 GS-RTS 算法收敛性分析	41
3.4 本章小结	42
第 4 章 基于块分星座的改进 GS-RTS 检测算法	43
4.1 块分星座图方法	43
4.1.1 BC 方法原理	43
4.2 BC-GS-RTS 算法	46
4.2.1 BC-GS 算法	46
4.2.2 BC-GS-RTS 算法流程	48
4.3 仿真与分析	49
4.3.1 BC-GS-RTS 性能仿真	49
4.3.2 BC-GS-RTS 复杂度分析	51
4.3.3 BC-GS-RTS 收敛性分析	53
4.4 本章小结	54
第 5 章 总结与展望	55
5.1 本文工作总结	55
5.2 下一步研究工作展望	56
参考文献	58
致谢	62
攻读硕士学位期间从事的科研工作及取得的成果	63

图录

图 1.1 全球移动数据平均月流量预测(2015-2020)	1
图 1.2 大规模 MIMO 信号检测算法分类	4
图 2.1 大规模 MIMO 系统上行链路模型图	9
图 2.2 天线数为 32×32 的矩阵 $H^T H$ 信道硬化表现	11
图 2.3 天线数为 64×64 的矩阵 $H^T H$ 信道硬化表现	11
图 2.4 天线数为 128×128 的矩阵 $H^T H$ 信道硬化表现	12
图 2.5 线性检测算法性能对比	18
图 2.6 不同天线数量下 LAS 算法性能对比	22
图 3.1 RTS 算法流程图	25
图 3.2 4QAM 调制下星座图邻域定义	26
图 3.3 RTS 和 LAS 算法 BER 对比	31
图 3.4 RTS 和 LAS 算法复杂度对比	32
图 3.5 几种主流迭代算法 BER 对比	35
图 3.6 RTS 和 GS-RTS 算法流程对比	36
图 3.7 4QAM 下 RTS 算法和改进的 GS-RTS 算法 BER 对比	38
图 3.8 16QAM 下 RTS 算法和改进的 GS-RTS 算法 BER 对比	39
图 3.9 GS 和 MMSE 算法复杂度对比	41
图 3.10 不同迭代次数下改进的 GS-RTS 算法 BER 对比	42
图 4.1 块分星座图与常规方法对比	45
图 4.2 三种算法 BER 对比	48
图 4.3 BC-GS-RTS 和 GS-RTS 算法流程对比	48
图 4.4 16QAM 下三种算法 BER 对比	51
图 4.5 三种初始解算法复杂度对比	53
图 4.6 不同迭代次数下 BC-GS-RTS 算法 BER 对比	54

表录

表 2.1 常规检测算法性能和复杂度对比	17
表 2.2 线性算法仿真参数设置	17
表 2.3 LAS 算法流程	19
表 2.4 LAS 算法仿真参数设置	22
表 3.1 4QAM 下 RTS 算法禁忌表	28
表 3.2 GS 迭代算法仿真参数配置	35
表 3.3 GS-RTS 算法仿真参数	37
表 3.4 GS 和 MMSE 算法计算复杂度对比	40
表 4.1 块分星座图方法实现过程	44
表 4.2 BC-GS 算法实现流程	46
表 4.3 BC-GS 算法仿真参数	47
表 4.4 BC-GS-RTS 算法仿真参数	50
表 4.5 三种初始解算法计算复杂度对比	52

注释表

AMP	Approximate Message Passing, 近似信息传递
AWGN	Additive White Gaussian Noise, 加性高斯白噪声
BC	Block Constellation, 块分星座图
BER	Bit Error Rate, 误码率
BP	Belief Propagation, 置信度传播
CDMA	Code Division Multiple Access, 码分多址
CSI	Channel State Information, 信道状态信息
FDD	Frequency Division Duplexing, 频分双工
GS	Gauss Seidel, 高斯赛德尔
LAS	Likelihood Ascend Search, 似然上升搜索
LLR	Log Likelihood Ratio, 对数似然比
LS	Local Search, 本地邻域搜索
LTE-A	Long Term Evolution-Advanced, 长期演进技术升级版
MAP	Maximum A Posteriori, 最大后验概率
MF	Matched Filter, 匹配滤波
MIMO	Multiple-Input Multiple-Output, 多入多出技术
ML	Maximum Likelihood, 最大似然
MMSE	Minimum Mean Square Error, 最小均方误差
MRC	Maximum Ratio Combination, 最大比合并
MUSA	Multi User Shared Access, 多用户共享接入
NSA	Neumann Series Approximation, 纽曼级数扩展
PDMA	Pattern Division Multiple Access, 图样分割多址接入
QAM	Quadrature Amplitude Modulation, 正交幅度调制
QPSK	QuadriPhase Shift Keying, 四相移键控
RTS	Reactive Tabu Search, 主动禁忌搜索
SCMA	Sparse Code Multiple Access, 稀疏码多址接入

SD	Sphere Decoding, 球形译码
SISO	Single Input Single Output, 单输入单输出
SNR	Signal Noise Ratio, 信噪比
TDD	Time Division Duplexing, 时分双工
TS	Tabu Search, 禁忌搜索
VR	Virtual Reality, 虚拟现实
ZF	Zero Forcing, 迫零检测

<http://www.ixueshu.com>

第1章 引言

1.1 研究背景及意义

随着智能穿戴设备、无人机、虚拟现实(Virtual Reality, VR)智能体感设备、无人驾驶汽车等新型多样化的智能终端设备迅速普及与研发,近些年来移动数据流量呈现出爆炸式的增长。思科《全球移动数据流量(2015-2020)白皮书》中预测(见图 1.1),自 2015 到 2020 年全球移动数据平均月流量以每年 53%的速度递增,预计 2020 年全球移动数据平均月流量将达到 30.6EB (30.6×10^{18} 字节),这对现有第四代移动通信系统(4G)的无线数据业务需求提出了巨大的挑战^[1]。

面对物联网、虚拟现实、人工智能等新技术的蓬勃发展,现有的无线接入网络暴露出频谱资源短缺、频谱利用率低、能耗效率低等严重问题。为了更好地解决上述问题,第五代移动通信系统(5G)呼之欲出。按照 IMT-2020(5G)推进组《5G 概念白皮书》中定义,5G 将会解决连续广域覆盖、热点大容量、低功耗大连接和低时延高可靠四个主要技术场景带来的挑战^[2]。和 4G 相比,5G 的提升是全方位的,其中用户体验速率达到 1Gbps,峰值速率更是达到数十 Gbps,空口时延 1ms,可靠性接近 100%,移动数据容量是现有的 1000 倍以及能耗效率提高 100 倍^[3]。

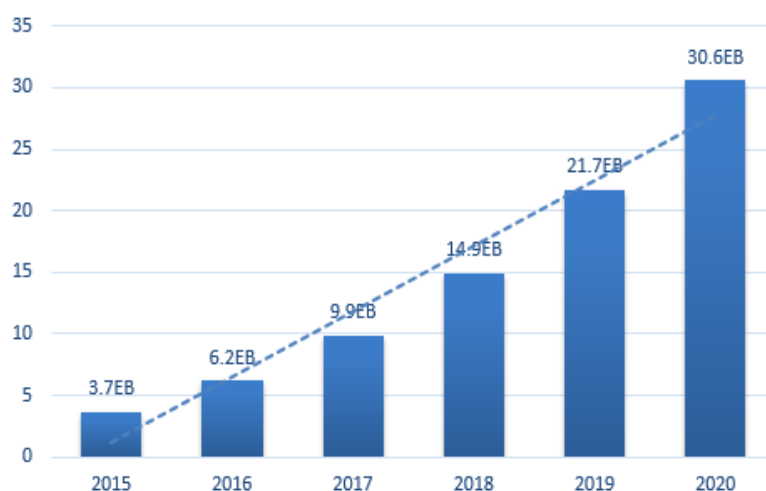


图 1.1 全球移动数据平均月流量预测(2015-2020)

从上述描述可以看出, 5G 具备高性能、低时延与高容量的特性, 而这些优点主要体现在毫米波、小基站、全双工、新型多址以及 Massive MIMO(大规模 MIMO)这五大关键技术上^[4]:

(1) 毫米波

无线传输速率提高一般有两种方法: 一是增加频谱利用率; 二是增加频谱带宽。毫米波技术就是通过第二种方法增加速率的, 以 60GHz 为例, 每个信道可用信号带宽为 2GHz。然而毫米波技术最大的缺点就是就是穿透力差、衰减快, 致密化小基站正好弥补这一问题。

(2) 小基站

5G 网络将不再只依赖大型基站, 大量体积更小的基站入网服务成为趋势, 它们可以覆盖大基站无法触及的通信盲区。基于这种异构网络可以使小区网络更加致密化, 在提高网络服务的同时也大大降低了功耗。

(3) 全双工

全双工技术是在现有的频分双工(Frequency Division Duplexing, FDD)和时分双工(Time Division Duplexing, TDD)模式基础上的一次重大革新, 通信设备的发射机和接收机允许在上下行相同的时间使用相同的频率进行工作, 从而使通信节点实现真正的双向通信, 提高系统的吞吐量, 降低延迟时间。

(4) 新型多址

由中兴提出的多用户共享接入(Multi User Shared Access, MUSA)、华为提出的稀疏码多址接入(Sparse Code Multiple Access, SCMA)和大唐提出的图样分割多址(Pattern Division Multiple Access, PDMA)等新型多址接入技术通过叠加传输多用户在相同资源上信息, 而接收端凭借算法分离出每一个用户信息, 极大地提高了频谱利用率和系统容量^[5]。

(5) Massive MIMO (大规模 MIMO)

多输入多输出(Multiple-Input Multiple-Output, MIMO)技术已经实际应用在 4G 网络中, 现有的基站根据 LTE-A 最新标准可以配置 8 根天线。然而面对新的无线网络需求, 这显然是不够的^[6]。通常来说, 发送端和接收端配置的天线数量越多, 数据的传输速率、可靠性以及系统的容量都会得到显著提升。正是基于这一点, 2010 年美国贝尔实验室的 Marzetta 教授提出 Massive MIMO 概念^[7], 和 4G 的 MIMO

技术相比 5G 基站可以配置上百根天线,这些天线通过 Massive MIMO 技术形成大规模天线阵列取代目前的多天线技术。

斯隆大学教授 Ove Edfors 指出:“Massive MIMO 开启无线通讯新方向,Massive MIMO 技术通过导入空间域的途径,其方式是在基站端采用大量天线以及为其进行同步处理,如此在频谱和能源效率方面取得几十倍增益。”显而易见基站端配置上百根天线带来的优势是巨大的:首先信道空间自由度大幅增加,当基站端天线数远大于小区用户天线总数时,基站和用户之间信道趋向于正交^[8],用户之间相互干扰趋于消失;其次大规模天线阵列增益可以提高用户信噪比,基站可以同时为更多的小区用户发送接收数据,从而将小区的无线网络容量提高数十倍或者更大;然后文献[9]中指出发射功率和基站天线数量总和的平方成反比,所以基站端布置大量天线还可以大幅降低基站发射功率,从而提高能耗效率,更加绿色环保;最后文献[10]中证明当基站天线数逐渐增多时,借助于最大比合并(Maximum Ratio Combination, MRC)的接收器,大规模 MIMO 系统能够显著提高上行链路频谱效率。

毋庸置疑,大规模 MIMO 技术是 5G 能否实现商用的关键技术之一,凭借着系统容量大、能源消耗少和频谱效率高等优势成为近些年业内的研究热点。可是基站端布置大规模天线阵列带来大量的机遇,同时也带来了巨大的挑战,尤其是上行链路信号检测算法。随着基站端天线数量从几根跃升到几百根,小区系统容量提升,用户数随之增加,所以上行链路无论是发送端还是接收端天线数量都大幅增加,原有的 MIMO 信号检测算法因为天线数增加复杂度过高,无法实际应用在大规模 MIMO 系统中。因此在保证性能的前提下,研究出适用于上行链路的低复杂度信号检测算法对于大规模 MIMO 技术的实际工程应用意义重大。本文研究的重点就是大规模 MIMO 上行链路信号检测算法研究。

1.2 大规模 MIMO 检测算法的研究现状

大规模 MIMO 系统信号检测算法实际就是在接收端将大规模天线阵列发送的信号恢复出来。随着基站端天线数量和小区用户数大幅增加,原有的 MIMO 信号检测算法存在复杂度过高,存在难以在硬件平台实现的问题。所以近些年业内提出了许多适用于大规模 MIMO 系统的低复杂度检测算法,比如基于本地邻域搜索(Local Search, LS)算法的似然上升搜索(Likelihood Ascend Search, LAS)算法^[1]和主

动禁忌搜索(Reactive Tabu Search, RTS)算法^[12]等。传统的 MIMO 非线性检测算法因为天线增加,复杂度大幅增加,无法适用在大规模 MIMO 系统中^[13],所以本文不在叙述,具体信号检测算法分类见图 1.2。

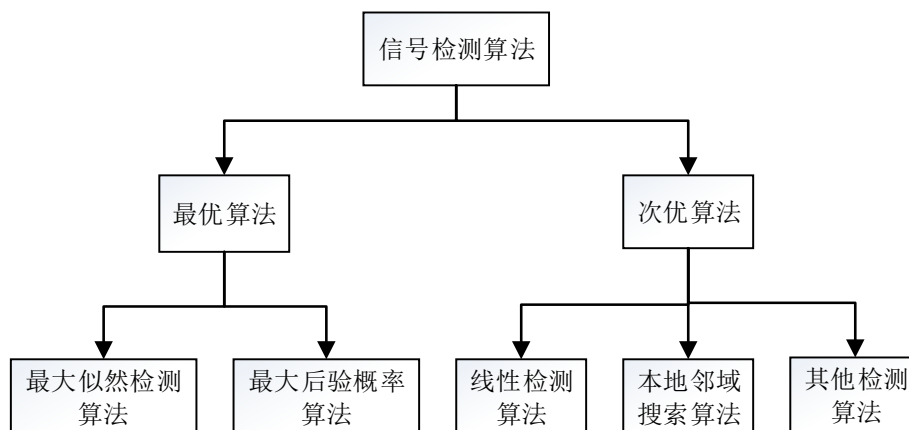


图 1.2 大规模 MIMO 信号检测算法分类

1. 最优检测算法

大规模 MIMO 信号检测算法是在 MIMO 信号检测算法的基础上发展研究的,在 MIMO 检测算法中性能最优的当属最大似然(Maximum Likelihood, ML)检测算法和最大后验概率(Maximum A Posteriori, MAP)检测算法。ML 检测算法原理是基于最大似然准则,也就是从所有的用户信息中选取似然概率最大的一组作为检测结果^[14]。这样遍历所有用户信息寻找最优解的做法固然可以得到很好的性能,可是随着小区用户的天线数增加,复杂度随之呈指数的增加。MAP 是软解调算法,输出为比特的软信息,通过对软信息进行软判决完成检测,具备更好的检测性能^[15]。尽管两种算法性能最优,但是都因为复杂度过高,无法进行实际工程应用。

2. 线性检测算法

相比于最优检测算法,次优检测算法中的线性检测算法虽然性能不好,但是复杂度大幅降低,更加适用于实际工程中。线性检测算法原理是通过预先设计的变换矩阵对接收信号进行线性变换,并根据硬判决从而得到用户的发送信息比特,典型算法有匹配滤波(Matched Filter, MF)检测算法、迫零(Zero Forcing, ZF)检测算法和最小均方误差(Minimum Mean Square Error, MMSE)检测算法。MF 算法是最为简单的线性检测算法,将来自其他数据流的干扰统一认为是噪声处理^[16];相比于 MF 算法, ZF 算法可以通过预先设计转换矩阵彻底去除来自其他数据流的干扰,

但是由于信道噪声与转换矩阵相乘,使得噪声被放大^[17];MMSE 算法在前两种算法的基础上充分考虑了噪声因素,既保证噪声不会放大,也可以彻底去除来自其他数据流的干扰,所以 MMSE 算法性能都优于 ZF 算法和 MF 算法^[18]。

3. 基于本地邻域搜索算法

随着基站布置天线数增多,可以负载的小区用户数也相应增多,发射端和接收端之间的信道矩阵维数也不断增大,大规模 MIMO 系统信道出现特有的“信道硬化”的性质^[19]。根据这一特性,一大批基于机器学习和人工智能思想的信号检测算法凭借着复杂度低、性能接近最优算法的优势进入人们的视野,这其中就包括 LAS 算法和 RTS 算法。

LAS 算法和 RTS 算法都是基于本地邻域搜索的启发式算法,其算法原理为初始阶段需要运用 MIMO 系统中的线性检测算法求解初始值解向量,接着以一定参考值作为评判标准,选取参考值最小的向量,作为最优解向量输出。主动禁忌搜索(RTS)算法相比于同样基于本地邻域搜索的似然上升搜索(LAS)算法,他们共同点都是从一个初始解向量开始,接着不断尝试从当前解向量的邻域内寻找更优的解向量。不同点是 LAS 算法规定只可以向更小的参考值向量转变,这就可能陷入局部最小值陷阱,从而无法得到全局最优解。而 RTS 算法通过利用禁忌表这一反馈机制避免陷入局部死循环,在 LAS 算法基础上允许向较大的参考值向量转变,从而跳出局部最小值陷阱,求得全局最优解向量^[20]。

根据性能仿真结果分析,两种算法在复杂度不高的前提下,性能明显优于传统线性检测算法,更加适用于大规模 MIMO 系统中,而 RTS 算法凭借着“逃脱”策略,性能进一步得到提升,优于 LAS 算法,所以成为本文的研究重点。而根据 RTS 算法复杂度仿真结果分析,RTS 算法复杂度主要由初始解向量和迭代搜索过程两部分组成,并且随着天线数量增加,初始解向量部分复杂度逐渐成为整体复杂度的最主要部分,所以针对 RTS 算法初始解向量求解复杂度降低的研究也显得十分重要。

4. 其他检测算法

除了上述提及的信号检测算法外,近些年还出现了一大批适用于大规模 MIMO 系统的检测算法。比如,置信度传播(Belief Propagation, BP)算法,其算法利用建立图模型来求解发射信号概率问题^[21]。近似信息传递(Approximate Message

Passing, AMP)算法, 运用因子图的方法来计算节点间收发信息^[22]。线性检测算法 ZF 和 MMSE 涉及矩阵求逆过程, 随着天线数量增加, 大规模矩阵共轭转置求逆计算带来的复杂度过高问题不容忽视, 于是提出了一些基于迭代算法的低复杂度检测算法, 比如纽曼级数扩展(Neumann Series Approximation, NSA)迭代法^[23]、Richardson 迭代法^[24]和高斯赛德尔(Gauss-Seidel, GS)迭代法^[25]等。

1.3 主要研究内容

本文的主要研究方向是大规模 MIMO 上行链路信号检测算法研究。本文首先介绍了大规模 MIMO 系统信道新特性及其上行链路系统信道模型; 然后重点研究了现有的大规模 MIMO 系统上行链路信号检测算法, 对基于启发式思想的主动禁忌搜索(RTS)信号检测算法原理进行了详细的阐述, 并对其存在的缺陷进行了分析; 最后, 分别针对 RTS 算法初始值计算算法复杂度过高这一问题给出一种基于 GS 迭代法求解初始值的改进 RTS 信号检测算法, 针对改进的 GS-RTS 算法高阶调制性能较差这一问题给出一种基于块分星座图方法的改进检测算法, 并对两种改进检测算法分别进行了性能、复杂度和收敛性的仿真实验和分析。

1.4 论文结构安排

本文共分为五章, 各章的内容安排如下:

第 1 章: 引言。首先分析了新技术对于现有无线网络提出的新需求, 并对大规模 MIMO 技术的优越性和存在的问题进行了简要的叙述, 从而说明论文的研究背景和意义, 最后介绍了大规模 MIMO 系统上行链路信号检测算法的研究现状。

第 2 章: 大规模 MIMO 系统检测算法。首先主要是介绍了大规模 MIMO 系统信道特性和上行链路系统模型, 然后介绍了最优算法和线性检测算法原理, 并分析其缺陷, 接着重点介绍了基于本地邻域搜索的 LAS 算法, 通过实验仿真对其性能验证。

第 3 章: 基于 GS 迭代法的改进 RTS 检测算法。首先介绍了 RTS 算法和 GS 迭代法的基本原理, 然后针对链路用户信号采用低阶调制的情况给出一种基于 GS

迭代法求解初始值的改进 RTS 检测算法，从而降低原有的 RTS 算法整体复杂度，并对其改进算法的性能、复杂度和收敛性进行了仿真分析。

第4章：基于块分星座图的改进 GS-RTS 检测算法。针对 GS-RTS 算法链路用户信号采用高阶调制的情况性能较差这一问题，给出基于块分星座图思想的改进算法，从而提高 GS-RTS 算法检测性能，并对改进的算法进行了性能、复杂度和收敛性三方面的仿真分析验证。

第5章：总结和展望。对本文所做的研究工作进行总结，并且指出下一步要研究的内容和方向。

<http://www.ixueshu.com>

第2章 大规模 MIMO 系统检测算法

大规模 MIMO 技术是在 4G 中大放异彩的 MIMO 技术基础上而提出的一种新型多天线技术，其通过在基站端布置大规模天线阵列这一举措，极大地提高系统吞吐量、能量效率以及设备接入数量，可以应对未来热点高容量等场景中多用户接入时所带来的网络挑战，是 5G 的一种关键技术。本章节主要介绍大规模 MIMO 系统上行链路信道模型、“信道硬化”新特性以及常规信号检测算法，重点介绍基于本地邻域搜索的似然上升搜索算法原理，并通过实验仿真对比这些算法的复杂度和性能。

2.1 大规模 MIMO 系统上行链路信道模型

传统 MIMO 是以 V-BLAST 模型为基础建立信道模型的，大规模 MIMO 是从 MIMO 演进而来的，所以大规模 MIMO 系统信道模型同样基于 V-BLAST 模型。大规模 MIMO 系统中基站端和用户端之间通信传输算法研究分为基站端到用户端（下行链路）预编码算法和用户端到基站端（上行链路）信号检测算法两个方向^[26]，本文主要研究小区内多用户端发送信号到基站端的检测算法，并基于此建立上行链路信道模型。

本文考虑的场景是单个蜂窝小区下的大规模 MIMO 系统上行链路（见图 2.1），在基站一侧配置 M 根天线，为天线总数为 K 的单天线用户服务（默认均为单天线用户，所以本文提及小区用户数等同于小区用户天线数）。根据大规模 MIMO 系统上行链路特性，基站端布置大规模天线阵列，所以通常 $M \geq K$ 。

在空间中无线传输会受到时延、频移等参数影响，根据上行链路系统模型我们得知发射天线个数为 K ，接收天线个数为 M ， τ 为时延变量， t 为时间变量，第 j 个发射天线和第 i 个接收天线之间的等效信道冲击响应为 $h_{ij}(\tau; t)$ ， $i=1, 2, 3 \dots M$ ， $j=1, 2, 3 \dots K$ ，本系统信道可以表示为 $M \times K$ 矩阵 $\mathbf{H}(\tau; t)$ ：

$$\mathbf{H}(\tau; t) = \begin{bmatrix} h_{11}(\tau; t) & h_{12}(\tau; t) & \cdots & h_{1K}(\tau; t) \\ h_{21}(\tau; t) & h_{22}(\tau; t) & \cdots & h_{2K}(\tau; t) \\ \vdots & \vdots & \ddots & \vdots \\ h_{M1}(\tau; t) & h_{M2}(\tau; t) & \cdots & h_{MK}(\tau; t) \end{bmatrix} \quad (2.1)$$

第 j 个发射天线发送的信号为 $s_j(t)$ ，如果考虑无噪声的情况下，第 i 个接收天线接收的信号表示为：

$$r_i(t) = \sum_{j=1}^{N_K} \int_{-\infty}^{+\infty} h_{ij}(\tau; t) s_j(t - \tau) d\tau = \sum_{j=1}^{N_K} h_{ij}(\tau; t) \otimes s_j(\tau) \quad (2.2)$$

上式(2.2)中 \otimes 表示卷积，将式(2.1)代入可以得到：

$$\mathbf{r}(t) = \mathbf{H}(\tau; t) \otimes \mathbf{s}(\tau) \quad (2.3)$$

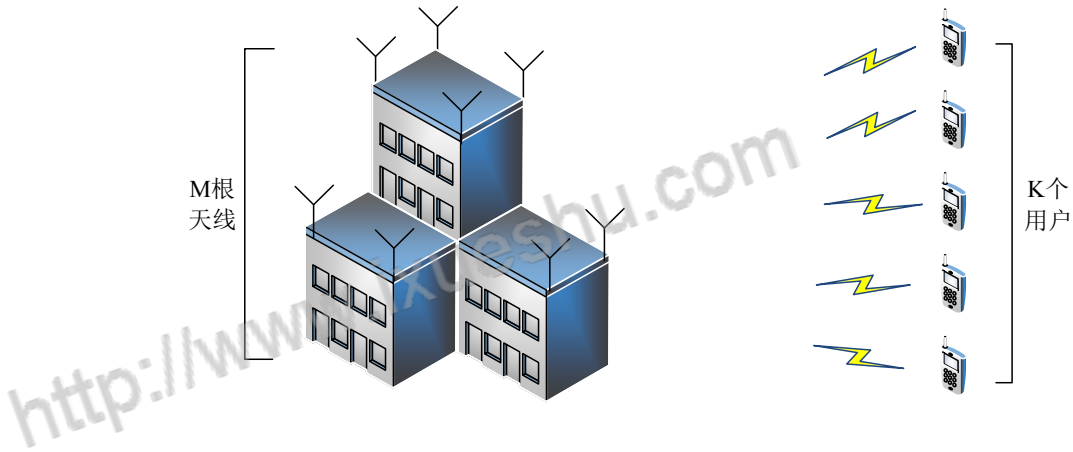


图 2.1 大规模 MIMO 系统上行链路模型图

如果为频率非选择性信道，信道矩阵 \mathbf{H} 还可以表示为：

$$\mathbf{H}(t) = \begin{bmatrix} h_{11}(t) & h_{12}(t) & \cdots & h_{1K}(t) \\ h_{21}(t) & h_{22}(t) & \cdots & h_{2K}(t) \\ \vdots & \vdots & \ddots & \vdots \\ h_{M1}(t) & h_{M2}(t) & \cdots & h_{MK}(t) \end{bmatrix} \quad (2.4)$$

本文研究的重点是上行链路信号检测算法的性能，所以选用的信道模型为平坦衰落信道下的基带时域离散信号模型，该系统复数模型为：

$$\mathbf{y}_c = \mathbf{H}_c \mathbf{x}_c + \mathbf{n}_c \quad (2.5)$$

其中发射信号向量 $\mathbf{x}_c \in \mathbb{C}^K$ ， $\mathbf{x}_c = [x_1, x_2, \dots, x_K]^T$ ；接收信号向量 $\mathbf{y}_c \in \mathbb{C}^M$ ， $\mathbf{y}_c = [y_1, y_2, \dots, y_M]^T$ ； \mathbb{C} 表示原始信号通过调制后的复数信号集；信道增益矩阵

$\mathbf{H}_c \in \mathbb{C}^{K \times M}$, \mathbf{H}_c 中的所有元素服从独立同分布的复高斯分布, 均值为 0, 方差为 1; 噪声向量 $\mathbf{n}_c \in \mathbb{C}^M$, \mathbf{n}_c 中所有元素服从独立同分布的复高斯分布, 均值为 0, 方差为 $N_k \bar{E} / \gamma$, \bar{E} 表示发送端发送每个符号的平均能量, γ 为接收天线平均信噪比。

为了便于后续检测算法的分析, 我们将式(2.5)化为实数型大规模 MIMO 系统模型:

$$\mathbf{y}_r = \mathbf{H}_r \mathbf{x}_r + \mathbf{n}_r \quad (2.6)$$

$$\mathbf{y}_r = \begin{bmatrix} \Re(\mathbf{y}_c) \\ \Im(\mathbf{y}_c) \end{bmatrix}, \quad \mathbf{H}_r = \begin{bmatrix} \Re(\mathbf{H}_c), -\Im(\mathbf{H}_c) \\ \Im(\mathbf{H}_c), \Re(\mathbf{H}_c) \end{bmatrix} \quad (2.7)$$

$$\mathbf{x}_r = \begin{bmatrix} \Re(\mathbf{x}_c) \\ \Im(\mathbf{x}_c) \end{bmatrix}, \quad \mathbf{n}_r = \begin{bmatrix} \Re(\mathbf{n}_c) \\ \Im(\mathbf{n}_c) \end{bmatrix} \quad (2.8)$$

因此上式(2.7) \mathbf{y}_r 为一个 $2M \times 1$ 维实数接收信号矩阵, \mathbf{H}_r 为一个 $2M \times 2K$ 维实数信道增益矩阵。式(2.8) \mathbf{x}_r 为一个 $2K \times 1$ 维实数发送信号矩阵, \mathbf{n}_r 为一个 $2M \times 1$ 维实数噪声信号向量矩阵。上行链路信号检测任务就是从接收到的信号向量 \mathbf{y}_r 中通过算法估计出发送的信号向量 \mathbf{x}_r , 本文应用场景假设基站对接收端信道状态信息 (Channel State Information, CSI) 完全已知, 也就是信道增益矩阵 \mathbf{H}_r 基站接收端已知。

2.2 信道硬化

大规模 MIMO 系统的基本思想是在上行链路现有的接收端 (基站一侧) 安装大量的额外天线来提高系统容量, 从而也增加小区内发送端 (终端用户) 数量, 提高传输速率。除了上述大规模 MIMO 系统带来的变化外, 随着天线数量增加, 信道也出现了新特征, 即信道硬化^[27]。

利用信道硬化这一特性可以降低一些适用于大规模 MIMO 系统的检测算法复杂度, 比如第三章中的 RTS 算法中的初始值向量求解算法, 所以这里简单叙述一下其现象与原理。根据随机矩阵理论 Marchenko-Pasture 定理^[28]: 如果矩阵 \mathbf{H} 中的全部元素独立同分布于均值为零, 方差为 $1/N$ 的任意形式分布, 随着矩阵 \mathbf{H} 的列数 M 和行数 K 不断增加至无穷大 (即 $M, K \rightarrow +\infty$), 并且行数和列数比例趋于常

数（即 $M/K \rightarrow c$ ， c 表示常数），矩阵 $\mathbf{H}^T \mathbf{H}$ 的特征值趋近于确定分布，即 Marchenko-Pasture 分布。

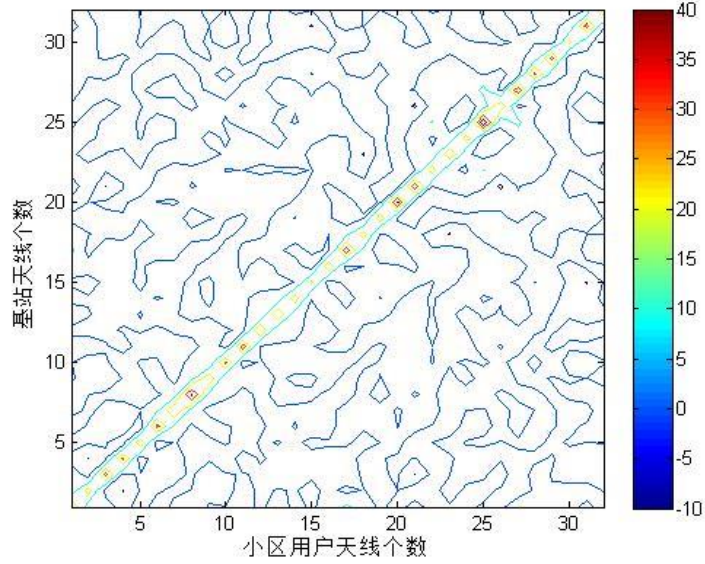


图 2.2 天线数为 32×32 的矩阵 $\mathbf{H}^T \mathbf{H}$ 信道硬化表现

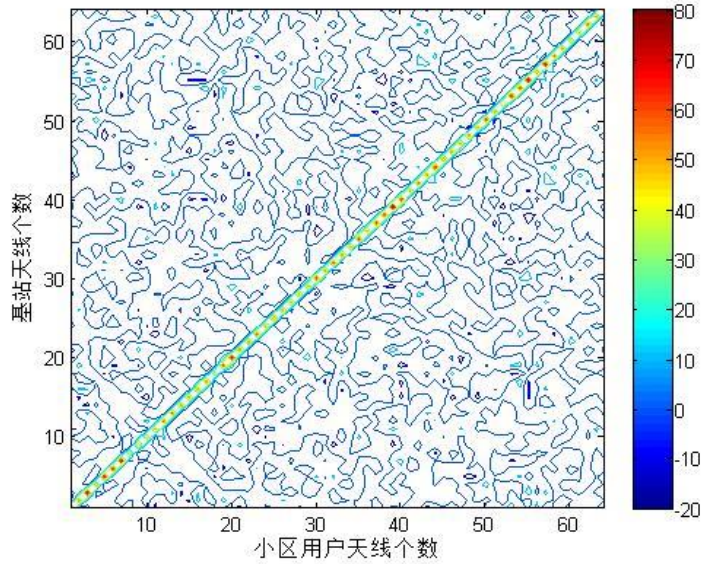


图 2.3 天线数为 64×64 的矩阵 $\mathbf{H}^T \mathbf{H}$ 信道硬化表现

大规模 MIMO 系统随着基站端天线增加，可以负载的小区用户数量也相应增加，又根据上述对信道增益矩阵 \mathbf{H} 的定义，矩阵 \mathbf{H} 满足 Marchenko-Pasture 定理的要求条件，我们将其运用到大规模 MIMO 系统上行链路信道中来。随着信道增益矩阵 \mathbf{H} 规模不断增大，矩阵 $\mathbf{H}^T \mathbf{H}$ 的特征值趋近于确定分布，也就是出现信道硬化

现象。我们通过三个场景来验证这一现象的存在,如图 2.2-2.4 所示,通过观察 $\mathbf{H}^T \mathbf{H}$ 矩阵所有元素取绝对值后生成的等高线图,我们发现对角上的元素明显比非对角元素大很多,并且随着发射端和接收端天线从 32 根增加到 128 根,这一现象越发显著。

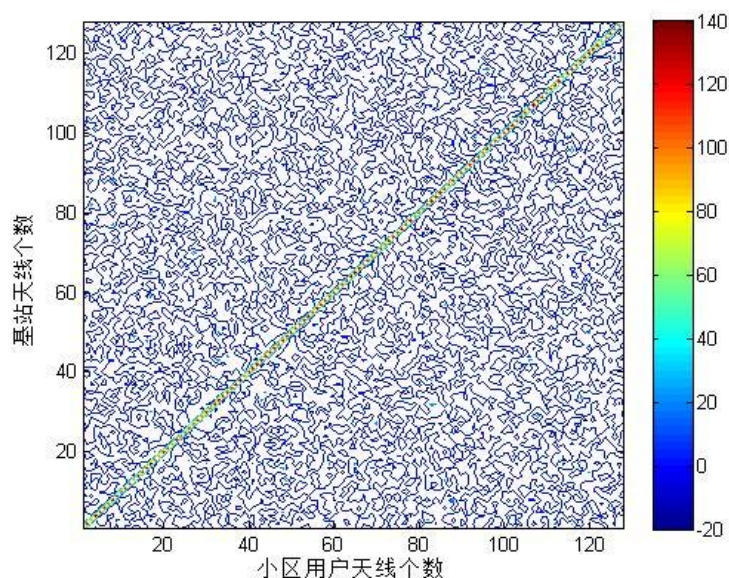


图 2.4 天线数为 128×128 的矩阵 $\mathbf{H}^T \mathbf{H}$ 信道硬化表现

2.3 常规信号检测算法

大规模 MIMO 系统常规信号检测算法主要是指应用于 MIMO 的经典检测算法,一般用于天线数量较小的情况下,分为最优检测算法、线性检测算法。正如第一章介绍,最优检测算法主要包括最大似然检测算法(ML)和最大后验概率检测算法(MAP),线性算法有匹配滤波检测算法(MF)、迫零检测算法(ZF)和最小均方误差检测算法(MMSE)。下面就结合大规模 MIMO 系统上行链路信道模型,具体针对上述检测方法进行分析介绍。

2.3.1 最优检测算法

1. 最大后验概率检测算法

最大后验概率检测是基于软判决的检测算法，输出为信息比特的软信息，即对数似然比(Log likelihood ratio, LLR)，LLR 的定义^[29]如下：

$$LLR(b_i) = \log \left(\frac{Pr\{b_i = 1 | y, h\}}{Pr\{b_i = 0 | y, h\}} \right) \quad (2.9)$$

上式中 $Pr\{b_i = 1 | y, h\}$ 和 $Pr\{b_i = 0 | y, h\}$ 分别是 $b_i = 1$ 和 $b_i = 0$ 的后验概率；当 $LLR(b_i) > 0$ 时，则 $b_i = 1$ 的可能性较大；而当 $LLR(b_i) < 0$ 时，则 $b_i = 0$ 的可能性大。

根据式(2.5)中调制后的复数发射信号 \mathbf{x}_c 可以表示为：

$$\mathbf{x}_c = \mathbf{x}_I + j\mathbf{x}_Q \quad (2.10)$$

而复数发射信号 \mathbf{x}_c 与其相对应的信息比特映射关系为：

$$b_1, b_2, \dots, b_{2k-1}, b_{2k} = b_{1,x_I}, b_{1,x_Q}, b_{2,x_I}, b_{2,x_Q}, \dots, b_{k,x_I}, b_{k,x_Q} \quad (2.11)$$

式(2.5)中复数接收信号 \mathbf{y}_c 对应的软信息为 $LLR(b_{i,x_I}), LLR(b_{i,x_Q}) (i=1, 2, \dots, k)$ ；首先计算出 $LLR(b_{i,x_I}), LLR(b_{i,x_Q}) (i=1, 2, \dots, k)$ ，然后进行译码，最后输出实部和虚部的比特流数据。

根据式(2.11)可得接收信号 \mathbf{y}_c 软信息表达式为：

$$\begin{cases} LLR(b_{i,x_I}) = \log \left(\frac{Pr\{b_{i,x_I} = 1 | \mathbf{Y}, \mathbf{h}\}}{Pr\{b_{i,x_I} = 0 | \mathbf{Y}, \mathbf{h}\}} \right) \\ LLR(b_{i,x_Q}) = \log \left(\frac{Pr\{b_{i,x_Q} = 1 | \mathbf{Y}, \mathbf{h}\}}{Pr\{b_{i,x_Q} = 0 | \mathbf{Y}, \mathbf{h}\}} \right) \end{cases} \quad (2.12)$$

运用贝叶斯准则，则得：

$$LLR(b_{i,x_I}) = \log \left(\frac{\sum_{x_c \in b_i=1} f_{Y|h,x} \{ \mathbf{Y} | \mathbf{h}, \mathbf{x}_c \}}{\sum_{x_c \in b_i=0} f_{Y|h,x} \{ \mathbf{Y} | \mathbf{h}, \mathbf{x}_c \}} \right) \quad (2.13)$$

$$f_{Y|h,x} \{ \mathbf{Y} | \mathbf{h}, \mathbf{x}_c \} = \frac{1}{\sigma \sqrt{\pi}} \exp \left(\frac{-1}{\sigma^2} \|\mathbf{Y} - \mathbf{h}\mathbf{x}_c\|^2 \right) \quad (2.14)$$

所以式(2.14)可以化为：

$$LLR(b_{i,x_I}) = \log \left(\frac{\sum_{x_c \in b_i=1} \exp \left(\frac{-1}{\sigma^2} \|\mathbf{Y} - \mathbf{h}\mathbf{x}_c\|^2 \right)}{\sum_{x_c \in b_i=0} \exp \left(\frac{-1}{\sigma^2} \|\mathbf{Y} - \mathbf{h}\mathbf{x}_c\|^2 \right)} \right) \quad (2.15)$$

又根据 max 近似函数：

$$\log\left(\sum_j \exp(-X_j)\right) \approx -\min_j(X_j) \quad (2.16)$$

式(2.15)可以化为：

$$LLR(b_{i,x_i}) = \frac{1}{\sigma^2} \left\{ \min_{x_c \in b_i=0} \|Y - Hx_c\|^2 - \min_{x_c \in b_i=1} \|Y - Hx_c\|^2 \right\} \quad (2.17)$$

最大后验概率检测算法是基于软输出判决算法中性能最优的^[30]，可是根据式(2.17)可以得到计算每一个比特信息需要对 $2^{Q \cdot K}$ 个发送比特向量进行检测（ Q 表示选取的调制阶数），其复杂度随着选取的调制阶数和发射天线数呈指数增长，在 MIMO 系统中都无法使用，所以无法实际使用在大规模 MIMO 系统中。

2. 最大似然检测算法

最大似然算法(ML)是硬判决算法中性能最优的，通过遍历性的计算接收信号 y_c 和发射信号 x_c 经过信道增益矩阵 H 后所有可能值的欧氏距离，找出最小值即为最优解向量^[31]。其公式为：

$$x_{ML} = \arg \min_{x_c \in Q^{N_K}} \|y_c - H_c x_c\|^2 = \arg \min_{x_c \in Q^{N_K}} \sum_{i=1}^{N_M} \left\| y_i - \sum_{j=1}^{N_K} H_{ij} x_j \right\|^2 \quad (2.18)$$

当发送信号服从独立分布且概率相等时，最大似然检测算法等同于最大后验概率检测算法，所以同样的随着调制阶数和发送端天线数量增加，ML 算法复杂度也成指数增加。比如当发射天线数为 5，选取 16QAM 进行调制，调制阶数 $Q = 4$ ，则 ML 需要遍历的信号向量个数为 $2^{Q \cdot K} = 1048576$ ，所以 ML 算法在天线数量不多，调制阶数不算高的情况下复杂度都如此之高，很难应用在实际工程中，一般是用作算法性能对比的理论科研中。

2.3.2 线性检测算法

1. 匹配滤波算法

匹配滤波算法(MF)是线性检测算法中最为简单的算法。我们定义 h_i 为信道增益矩阵 H 的第 i 列向量， $i = 1, 2, \dots, K$ 。将公式(2.5)化为：

$$y_c = H_c x_c + n_c = \sum_{i=1}^{N_K} h_i x_i + n_c$$

$$= \mathbf{h}_k \mathbf{x}_k + \sum_{i=1, i \neq k}^{N_K} \mathbf{h}_i \mathbf{x}_i + \mathbf{n}_c \quad (2.19)$$

根据式(2.19), MF 算法特点在于只检测第 k 个发射天线发送的信号 \mathbf{x}_k , 而将接收到的来自于其他天线的信号当作噪声处理。其信号检测算法过程只将信号共轭转置矩阵 \mathbf{H}^H 同接收到的信号 \mathbf{y}_c 相乘:

$$\hat{\mathbf{x}} = \mathbf{H}_c^H \mathbf{y}_c \quad (2.20)$$

MF 算法变换矩阵为 $\mathbf{G}_{MF} = \mathbf{H}_c^H$, 其复杂度仅为 $M \times K$, 凭这一点非常适用于大规模 MIMO 系统, 可是由于发射天线增加以及使用高阶调制模式, 来自于其他天线的干扰增强, MF 算法性能表现差强人意。

2. 迫零检测算法

迫零检测算法(ZF)是一种引入迫零向量的线性算法, 首先简单叙述迫零向量矩阵推导过程。根据式(2.18)得:

$$\|\mathbf{y}_c - \mathbf{H}_c \mathbf{x}_c\|^2 = \mathbf{y}_c^H \mathbf{y}_c - \mathbf{y}_c^H \mathbf{H}_c \mathbf{x}_c - \mathbf{x}_c^H \mathbf{H}_c^H \mathbf{y}_c + \mathbf{x}_c^H \mathbf{H}_c^H \mathbf{H}_c \mathbf{x}_c \quad (2.21)$$

式(2.19)针对 \mathbf{x}_c 求导得:

$$\frac{d\|\mathbf{y}_c - \mathbf{H}_c \mathbf{x}_c\|^2}{d\mathbf{x}_c} = -2\mathbf{H}_c^H \mathbf{y}_c + 2\mathbf{H}_c^H \mathbf{H}_c \mathbf{x}_c \quad (2.22)$$

为了寻找最优解向量, 将上式右端等于 0, 推导出 ZF 算法公式^[32]为:

$$\hat{\mathbf{x}} = (\mathbf{H}_c^H \mathbf{H}_c)^{-1} \mathbf{H}_c^H \mathbf{y}_c \quad (2.23)$$

所以 ZF 算法迫零向量矩阵为 $\mathbf{G}_{ZF} = (\mathbf{H}_c^H \mathbf{H}_c)^{-1} \mathbf{H}_c^H$, 我们定义 \mathbf{g}_k 为矩阵 \mathbf{G}_{ZF} 的第 k 列向量, 所以根据 ZF 算法估计第 k 个天线发送的信号为:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \mathbf{g}_k \mathbf{y}_c = \mathbf{g}_k \mathbf{H}_c \mathbf{x}_c + \mathbf{g}_k \mathbf{n}_c \\ &= \mathbf{x}_k + \mathbf{g}_k \mathbf{n}_c \end{aligned} \quad (2.24)$$

3. 最小均方误差算法

最小均方误差算法(MMSE)是基于检测算法估计出来的发射信号 \mathbf{x}_{MMSE} 和实际发送的信号 \mathbf{x}_c 均方误差值最小思想实现的。我们定义 MMSE 算法转换矩阵为 \mathbf{G}_{MMSE} , 所以 MMSE 算法公式^[33]为:

$$\hat{\mathbf{x}} = \mathbf{G}_{MMSE} \mathbf{y}_c \quad (2.25)$$

估计发射信号 $\hat{\mathbf{x}}$ 和实际发射的信号 \mathbf{x}_c 均方误差值表示为:

$$\begin{aligned} \|\hat{\mathbf{x}} - \mathbf{x}_c\|^2 &= (\mathbf{G}_{MMSE} \mathbf{y}_c - \mathbf{x}_c)(\mathbf{G}_{MMSE} \mathbf{y}_c - \mathbf{x}_c)^H \\ &= \mathbf{G}_{MMSE} \mathbf{y}_c \mathbf{y}_c^H \mathbf{G}_{MMSE}^H - \mathbf{G}_{MMSE}^H \mathbf{y}_c \mathbf{x}_c^H - \mathbf{x}_c \mathbf{y}_c^H \mathbf{G}_{MMSE}^H + \mathbf{x}_c \mathbf{x}_c^H \end{aligned} \quad (2.26)$$

我们定义矩阵 \mathbf{y}_c 的协方差矩阵为 $\mathbf{R}_{y_c y_c} = \mathbf{E}(\mathbf{y}_c \mathbf{y}_c^H)$, 矩阵 \mathbf{x}_c 的协方差矩阵为 $\mathbf{R}_{x_c x_c} = \mathbf{E}(\mathbf{x}_c \mathbf{x}_c^H)$, 噪声矩阵协方差 $\mathbf{R}_{n_c n_c} = \mathbf{E}(\mathbf{n}_c \mathbf{n}_c^H)$, 矩阵 \mathbf{x}_c 和矩阵 \mathbf{y}_c 互协方差矩阵为 $\mathbf{R}_{x_c y_c} = \mathbf{E}(\mathbf{x}_c \mathbf{y}_c^H)$ 和 $\mathbf{R}_{y_c x_c} = \mathbf{E}(\mathbf{y}_c \mathbf{x}_c^H)$, 所以上式期望均值可以写成:

$$\begin{aligned} &\mathbf{E}\left\{(\mathbf{G}_{MMSE} \mathbf{y}_c - \mathbf{x}_c)(\mathbf{G}_{MMSE} \mathbf{y}_c - \mathbf{x}_c)^H\right\} \\ &= \mathbf{G}_{MMSE} \mathbf{R}_{y_c y_c} \mathbf{G}_{MMSE}^H - \mathbf{R}_{x_c y_c} \mathbf{G}_{MMSE}^H - \mathbf{G}_{MMSE} \mathbf{R}_{y_c x_c} + \mathbf{R}_{x_c x_c} \end{aligned} \quad (2.27)$$

可以将式(2.25)化为关于 \mathbf{G}_{MMSE} 的函数:

$$f(\mathbf{G}_{MMSE}) = \mathbf{G}_{MMSE} \mathbf{R}_{y_c y_c} \mathbf{G}_{MMSE}^H - 2\mathbf{G}_{MMSE} \mathbf{R}_{y_c x_c} + \mathbf{R}_{x_c x_c} \quad (2.28)$$

为了求得均方差值最小只需上式对转换矩阵求向量梯度并等于零, 即:

$$\frac{\partial f}{\partial \mathbf{G}_{MMSE}} = 2\mathbf{R}_{y_c y_c} \mathbf{G}_{MMSE}^H - 2\mathbf{R}_{y_c x_c} = 0 \quad (2.29)$$

所以得到: $\hat{\mathbf{x}} = \mathbf{R}_{x_c y_c} \mathbf{R}_{y_c y_c}^{-1} \mathbf{y}_c$

(2.30)

根据计算又得到: $\mathbf{R}_{n_c n_c} = \mathbf{E}(\mathbf{n}_c \mathbf{n}_c^H) = \sigma^2 \mathbf{I}$ (2.31)

$$\mathbf{R}_{y_c y_c} = \mathbf{E}(\mathbf{y}_c \mathbf{y}_c^H) = \mathbf{H}_c^H \mathbf{H}_c + \sigma^2 \mathbf{I} \quad (2.32)$$

$$\mathbf{R}_{x_c y_c} = \mathbf{E}(\mathbf{x}_c \mathbf{y}_c^H) = \mathbf{H}^H \quad (2.33)$$

将(2.30)和(2.31)代入(2.28)得到 MMSE 算法公式为:

$$\hat{\mathbf{x}} = (\mathbf{H}_c^H \mathbf{H}_c + \sigma^2 \mathbf{I})^{-1} \mathbf{H}^H \mathbf{y}_c \quad (2.34)$$

所以 MMSE 算法转换矩阵: $\mathbf{G}_{MMSE} = (\mathbf{H}_c^H \mathbf{H}_c + \sigma^2 \mathbf{I})^{-1} \mathbf{H}^H$ (2.35)

2.3.3 常规算法复杂度和性能分析

根据前面对大规模 MIMO 系统常规检测算法的描述, 整理出各种算法性能和复杂度对比, 如下表 2.1。首先从算法复杂度分析, 尽管 MAP/ML 算法是所有算

法中性能最出色的，可是牺牲的却是极高的复杂度 $O(2^{Q \cdot K})$ ，这决定这两种算法只能作为理论研究的对比算法；而线性算法整体复杂度低优势明显，复杂度最低的是 MF 算法仅为 $O(MK)$ ；而因为涉及对矩阵 $\mathbf{H}^H \mathbf{H}$ 求逆计算，ZF 和 MMSE 算法复杂度均升高到 $O(K^3)$ ，不过 MMSE 算法相对于 ZF 算法增加了一次矩阵加法和两次矩阵乘法运算，所以 MMSE 算法是线性检测算法中复杂度最高的。

表 2.1 常规检测算法性能和复杂度对比

检测算法	性能	复杂度
MAP/ML	最优	$2^{Q \cdot K}$
MF	差	MK
ZF	较差(低信噪比条件下，性能不如 MF)	K^3 (略低于 MMSE 算法)
MMSE	一般(信噪比任何情况下，性能都优于 MF 和 ZF 算法)	K^3

下面对三种线性检测算法在大规模 MIMO 系统场景下进行性能对比，首先配置系统参数，具体参数列表见表 2.2。

表 2.2 线性算法仿真参数设置

参数名称	配置说明
小区用户数	16
基站天线数	M
调制方式	16QAM
信噪比	8dB
用户位置	均匀分布
信号检测算法	MF/ZF/MMSE
信道类型	平坦慢衰落

再从性能的角度来说，MAP/ML 算法凭借着遍历性搜索检测出发射信号的方法，性能是所有算法中最优的，而线性算法性能明显差于 MAP/ML 算法。在线性算法中 MF 算法无法去除来自其他天线的信号干扰，所以性能最差；而 ZF 算法性

能居中，相比于 MF 算法，ZF 算法可以在接收端将来自于其他天线的干扰去除，但是噪声被迫零向量矩阵放大了，所以在低信噪比条件下，性能还不如 MF 算法；相比于 MF 和 ZF 算法，MMSE 算法不仅可以将来自其他天线的干扰信号去除，而且还避免了迫零算法因为噪声增强对信号检测产生的影响，所以算法性能明显优于 ZF 和 MF 两种算法。

图 2.5 为三种线性检测算法 BER 仿真结果，从图中得以验证三种算法性能优劣，同时随着基站天线数增加，三种算法性能都有所提高，不过仍无法满足大规模 MIMO 系统的性能要求。比如当基站天线增加到 100 时，MMSE 算法 BER 达到 10^{-4} ，而基站天线达到 120 时，ZF 算法 BER 也将接近 10^{-4} 。所以凭借着性能相对损失不大和复杂度低的优势，MMSE 算法通常作为本地邻域搜索算法的初始值算法，比如后续的 LAS，RTS 算法。

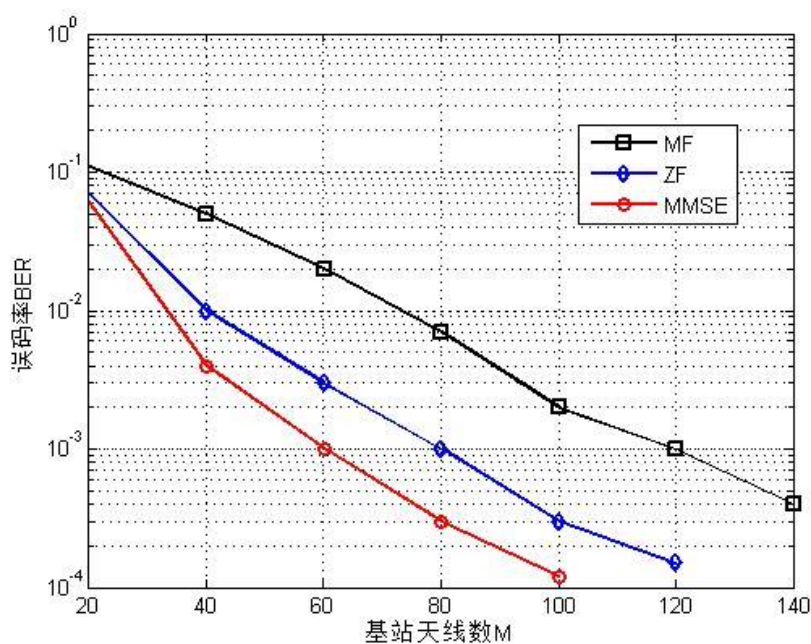


图 2.5 线性检测算法性能对比

2.4 基于本地邻域搜索算法

虽然传统 MIMO 线性检测算法复杂度低，可是三种线性检测算法都存在随着收发天线数增加，天线间干扰信号增强，算法性能无法满足大规模 MIMO 系统 BER

性能要求。相比而言, ML 算法性能最优, 可是复杂度随天线指数级别增加, 所以算法复杂度过高决定其无法应用在大规模 MIMO 系统中。因此, 不仅具备线性检测算法复杂度低的特性, 而且性能又接近最优检测算法, 同时适用于大规模 MIMO 系统的检测算法成为研究热点, 在这其中就包括基于本地邻域搜索的检测算法^[34]。

本地邻域搜索是一种启发式算法, 可以用于解决大规模 MIMO 系统检测算法中最小均方求解问题, 并且随着天线数量增加, 算法性能趋近于 ML 算法^[35]。不同于深度优先搜索和广度优先搜索这类遍历性穷举算法, 它的一个重要特征在于邻域的定义, 避免全局盲目搜索同时确保进行的每一次搜索在邻域内找到最优解, 然后依次进行这一步骤, 直到满足算法限定的要求, 输出检测结果。下面介绍 LAS 算法原理以及不同条件下算法性能和复杂度分析, RTS 算法和其改进算法将在第三章重点叙述。

2.4.1 LAS 算法原理

似然上升搜索算法(LAS)最早应用在神经网络的研究, 凭借着复杂度低的优势主要解决大规模像素位图恢复问题。最初由 Sun 将这一算法引入通信领域, 在 CDMA 系统中进行多用户信号检测^[36]。而首次将该算法用于大规模 MIMO 系统信号检测是由 K.Vishnu Vardhan, Saif K.Mohammed 和 A.Chockalingam 提出的^[37]。

表 2.3 LAS 算法流程

步骤 1: 基于线性算法计算初始解向量;
步骤 2: 定义当前解向量的邻域;
步骤 3: 根据 ML 代价函数确定最优邻居向量;
步骤 4: 根据 ML 代价函数判断该邻居向量是否优于当前解向量(如果判断“是”进行步骤 5; 如果判断“不是”直接进行步骤 6);
步骤 5: 将该邻居向量替换为当前解向量并回到步骤 2;
步骤 6: 终止并将当前解向量作为最优解输出。

LAS 算法适用于大规模 MIMO 系统中, 所以其算法模型为实数型大规模 MIMO 系统模型见式(2.6)。将公式(2.6)和(2.18)结合我们可以得到:

$$\mathbf{x}_{ML} = \arg \min \|\mathbf{y}_r - \mathbf{H}_r \mathbf{x}_r\|^2 = \arg \min \phi(\mathbf{x}) \quad (2.36)$$

我们定义 ML 代价函数为：

$$\phi(\mathbf{x}_r) \triangleq \mathbf{x}_r^H \mathbf{H}_r^H \mathbf{H}_r \mathbf{x}_r - 2\mathbf{y}_r^H \mathbf{H}_r \mathbf{x}_r \quad (2.37)$$

LAS 算法整体流程从初始解向量计算开始的，接着根据算法定义当前解向量的邻域，并在该邻域内根据 ML 代价函数选出最优解向量作为最优邻居向量，继续将这个最优邻居向量作为初始解向量重复进行上述迭代过程，直到该邻域内找不到更优的解向量，算法结束输出解向量，具体过程见上表 2.3。

下面针对上述流程作简要解释，首先初始解向量算法一般是基于线性检测算法，比如 ZF，MMSE。其次 LAS 算法邻域定义和球形译码(Sphere Decoding, SD)算法不同（根据欧氏距离定义邻域大小），是通过比较两个信号向量中不同的符号数而定义的^[38]。而 LAS 算法是按照逐个邻域进行迭代搜索过程的，相邻的两个迭代向量 $\mathbf{x}_r^{(n+1)}$ 和 $\mathbf{x}_r^{(n)}$ 的 ML 代价函数差值可以表示为：

$$\Delta\phi(\mathbf{x}_r^{(n)}) = \phi(\mathbf{x}_r^{(n+1)}) - \phi(\mathbf{x}_r^{(n)}) \quad (2.38)$$

为了搜索到更优的解向量，使得 $\Delta\phi(\mathbf{x}_r^{(n)}) < 0$ 。为了寻找到最优解，式(2.35)对 $\mathbf{x}_r^{(n)}$ 求向量梯度：

$$\mathbf{d}^{(n)} = \frac{\partial\phi(\mathbf{x}_r^{(n)})}{\partial(\mathbf{x}_r^{(n)})} = 2\mathbf{H}_r^H \mathbf{y}_r - 2\mathbf{H}_r^H \mathbf{H}_r \mathbf{x}_r^{(n)} \quad (2.39)$$

我们定义两个迭代向量差值为： $\mathbf{a}^{(n)} \triangleq \mathbf{x}_r^{(n+1)} - \mathbf{x}_r^{(n)}$ 以及 $\mathbf{b}^{(n)} \triangleq -\mathbf{H}_r^H \mathbf{H}_r \mathbf{a}^{(n)}$ ，将 $\mathbf{a}^{(n)}$ ， $\mathbf{b}^{(n)}$ 和式(2.37)代入式(2.36)可以得到：

$$\Delta\phi(\mathbf{x}_r^{(n)}) = \sum_{j \in T(n)} \mathbf{a}_j^{(n)} (\mathbf{d}_j^{(n)} + \mathbf{b}_j^{(n)}) \quad (2.40)$$

这里需要说明 $\mathbf{a}_j^{(n)}$ ， $\mathbf{b}_j^{(n)}$ 和 $\mathbf{d}_j^{(n)}$ 分别代表其向量的第 j 个元素，而 $T(n)$ 表示发射天线数量的集合($T(n) \in \{1, 2, \dots, N_K\}$)。如果迭代向量 $\mathbf{x}_r^{(n+1)}$ 和 $\mathbf{x}_r^{(n)}$ 每次仅考虑一个符号不同，称为 1-LAS 算法，仅在选中的一根天线上做数据更新，计算量为 N_K 。如果 $T(n) = L$ ，那么需要从 $C_{N_K}^M$ 个邻域中计算 ML 代价函数，也就是需要对 L 根天线数据更新，我们将这种情况称为 L-LAS 算法^[39]。

2.4.2 LAS 算法缺陷分析

似然上升算法是一种基于本地邻域搜索算法，也是一种启发式算法。随着收发天线大幅增加，LAS 算法凭借着低复杂度和性能好的优势适用于大规模 MIMO 系统，可是我们通过对算法流程分析发现，在邻域搜索过程中 LAS 算法容易陷入局部死循环，终止算法进程，这样的得到的解向量只是局部最优解向量，而非全局最优解向量，从而降低算法整体性能。

2.4.3 LAS 算法复杂度分析

在发射天线为 N_K ，基站接收天线为 N_M 的大规模 MIMO 系统上行链路中，L-LAS 算法复杂度主要来自两方面。第一：初始解向量计算；第二：迭代搜索过程。首先因为使用 ZF 和 MMSE 算法最为初始解向量算法，所以根据表 2.1 得知复杂度为 $O(N_K^3)$ ；其次第二点迭代搜索过程，如果 $L=1$ ，这一过程复杂度为 $O(N_K)$ ，如果 $L>2$ ，复杂度为 $O(N_K^L)$ ，所以这一过程复杂度和 M 取值息息相关。综上所述，L-LAS 算法整体复杂度为 $O(N_K^3 + N_K^L)$ ，通过下面实验仿真得到 $L=2$ 性能已经可以满足大规模 MIMO 系统需求，因此 LAS 算法复杂度主要来自于第一点初始向量计算，不过平均每个符号复杂度为 $O(N_K^2)$ ，还是适合应用在大规模 MIMO 系统信号检测中的。

2.4.4 LAS 算法性能仿真

本节基于大规模 MIMO 系统对 LAS 算法进行 BER 性能仿真验证。首先根据单小区条件下上行链路特征配置仿真参数，设定的小区用户和基站天线数量分别为 32, 64 和 128，调制方式为 4QAM，而初始解向量算法是 MMSE 算法，LAS 算法复杂度随着 L 值增加而增大，所以选定 L 值为 1 和 2，具体参数配置见表 2.4。然后根据表 2.4 中配置的仿真参数，进行不同天线数量条件下 LAS 算法的 BER 性能对比（见图 2.6）。

表 2.4 LAS 算法仿真参数设置

参数名称	参数大小
小区个数	1
基站天线数	32; 64; 128
小区用户数	32; 64; 128
调制方式	4QAM
卷积码速率	1/2
接收端译码算法	软输入 Viterbi
信道模型	瑞利衰落
信号检测算法	1-LAS/2-LAS
初始解向量算法	MMSE

从图 2.6 中可以看出，随着天线数量增加（从 32×32 增加到 128×128 ），1-LAS 和 2-LAS 算法性能都不断提高。比如，在 2-LAS 算法天线数为 64×64 ，信噪比为 14dB 时，误码率 BER 为 1×10^{-4} ，而天线数为 128×128 ，在误码率 BER 同样为 1×10^{-4} 条件下，信噪比下降了 2dB；随着 L 值增加，LAS 算法性能也有所提高，但是没有天线增加对算法的产生影响大。

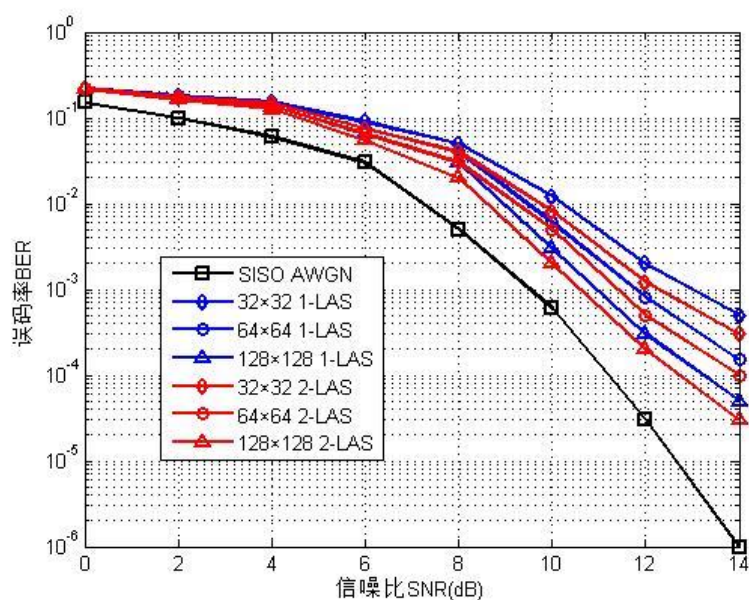


图 2.6 不同天线数量下 LAS 算法性能对比

2.5 本章小结

本章首先介绍了大规模 MIMO 系统上行链路模型和随着收发天线增加其信道出现的新特性“信道硬化”，其次介绍了大规模 MIMO 系统的常规信号检测算法原理，包括 MAP/ML 最优检测算法和传统 MIMO 中的线性检测算法，并比较分析了这些算法性能和复杂度，然后进一步介绍了基于本地邻域搜索的似然上升搜索检测算法原理和整体流程，通过实验仿真对其算法复杂度和性能做出了分析，并简单说明其算法存在缺陷的地方，为后两章 RTS 检测算法的改进研究打下理论基础。

第3章 基于GS迭代法的改进RTS检测算法

从上一章我们通过实验仿真发现LAS信号检测算法面对收发天线数目达到上百根的大规模MIMO系统仍然表现出检测性能好和复杂度低的优势,然而LAS算法也存在容易陷入局部死循环的问题,从而找到局部最优解而不是全局最优解。为了解决这一问题,本章将分析另一种基于邻域搜索的主动禁忌搜索算法(RTS),同样适用于大规模MIMO系统,并且在保持LAS算法性能好、复杂度低的前提下,利用禁忌表这一反馈机制避免陷入局部死循环,从而找到全局最优解向量。但是通过实验仿真,发现RTS算法初始解向量算法存在复杂度过高的问题,于是在RTS算法的基础上引入高斯赛德尔(GS)迭代算法,给出基于GS迭代法求解初始向量的GS-RTS改进算法,从而进一步降低算法复杂度。

3.1 RTS 算法

禁忌搜索(Tabu Search, TS)算法最初由Glover提出的^[40],主要用来解决数学里的优化组合问题,后来2009年N.Srindhi在TS算法基础上加入动态特性,故而形成主动禁忌(RTS)算法^[41]。RTS算法是一种基于机器学习和人工智能思想的检测算法,具备反馈机制,同时也是一种启发式算法。之所以称之为具有人工智能思想的算法,是因为RTS算法是对人类思维过程的一种模拟,建立“记忆列表”来接纳一些表现不是最优的解向量从而跳出局部最优解向量“陷阱”。面对上百根天线数量的大规模MIMO系统,RTS算法凭借着高检测性能和复杂度低的优势成为业内研究的热点^[42]。为了更好地说明RTS算法的优越性和存在的不足之处,本节首先介绍RTS算法原理以及实现步骤。

3.1.1 RTS 算法原理

在叙述RTS算法原理之前,首先说明其算法系统模型。RTS算法和LAS算法一样,都适用于第2章建立的实数型大规模MIMO系统信道模型(见公式2.6)。

RTS算法同样首先进行初始向量求解过程,通常还是基于传统MIMO系统的线性检测算法,比如ZF和MMSE等;然后根据邻域定义准则围绕初始解向量(当

前解向量)定义邻域向量;接着在邻域解向量中选取最优解向量,并将初始解向量(当前解向量)移动到邻域中最优解向量。不过在这一步骤中,不同于LAS算法,RTS算法即使邻域中最优解向量ML代价函数值不如初始解向量(当前解向量),也允许初始解向量(当前解向量)移动到邻域中的最优解向量,这一举措可以避免LAS算法陷入局部最优解向量困境,从而寻找到全局最优解向量;最终将上述步骤迭代一定次数,满足算法终止条件后,将所有迭代过程中寻找到的最优解向量作为最终解向量输出。(具体RTS流程图见3.1)

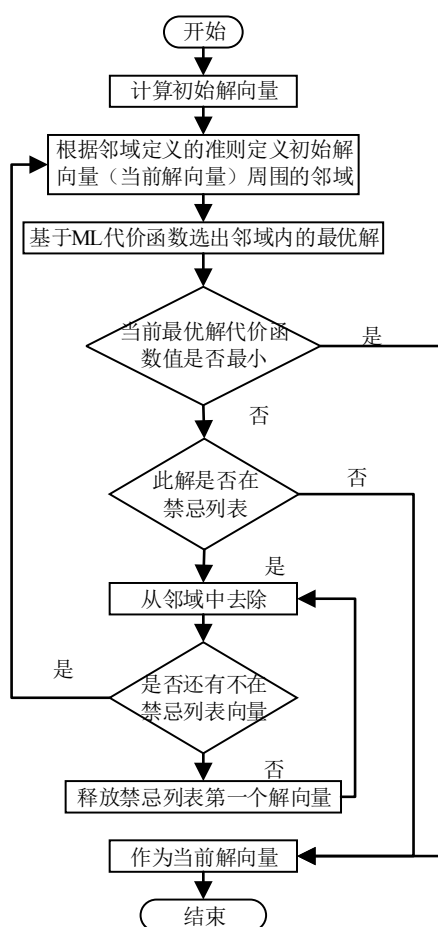


图 3.1 RTS 算法流程图

上述过程中解向量邻域的定义是基于一定数量的迭代实现的,RTS算法通过标记过去几次迭代中已经移动过的解向量生成“禁忌表”(禁止再次移动到这些解向量),以确保在搜索解向量空间中避免陷入局部死循环。这些迭代的数值被定义为“禁忌周期”,其数值根据在搜索路径中观察到的解向量的重复次数动态地改变。而禁忌周期系数是为了提高算法收敛速度,提高算法性能。

这里具体说明一下 RTS 算法不同于 LAS 算法的邻域定义准则、禁忌表生成过程、禁忌周期、终止条件和搜索迭代过程。

1. 邻域定义

首先我们定义星座图上的邻域, 假设调制符号集 $\mathbb{C}\{c_1, c_2 \dots c_Q\}$ 星座图上星座点的个数为 Q , 并定义 $L(c_Q)$ 为星座点 c_Q 的邻域, $|L(c_Q)| = L$, 所以 L 的取值为 1 到 $Q-1$, 选取最优邻居向量基于欧式距离大小评判的, 并定义 $w_v(c_Q)$ 为邻域 $L(c_Q)$ 的第 v 个元素 ($v=1, 2 \dots L$), 也就是星座点 c_Q 在星座图上的第 v 个邻居。对于一个选取的星座点 c_Q , 其他 L 个星座点构成它的邻域, 而和它欧式距离最小的向量成为最优邻居向量。下面用具体事例加以说明, 结合图 3.2 在 4QAM 调制下星座图上的星座点数为 4 (即 $Q=4$), 所以 L 的取值范围为 1 到 3。如果 $L=2$, 每个星座点的邻域分别为: $L(c_1) = \{c_2, c_3\}$, $L(c_2) = \{c_3, c_4\}$, $L(c_3) = \{c_2, c_4\}$, $L(c_4) = \{c_3, c_2\}$; 如果 $L=3$, 每个星座点邻域为: $L(c_1) = \{c_2, c_3, c_4\}$, $L(c_2) = \{c_1, c_4, c_3\}$, $L(c_3) = \{c_1, c_4, c_2\}$, $L(c_4) = \{c_2, c_3, c_1\}$ 。

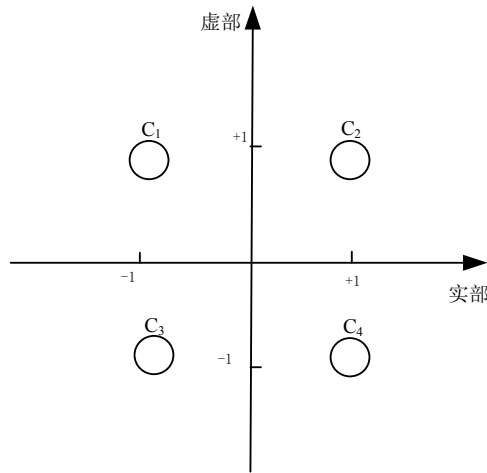


图 3.2 4QAM 调制下星座图邻域定义

在这基础上我们定义发射信号向量的邻域, 假设经过 i 次迭代得到的输出向量为: $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)} \dots x_{N_k}^{(i)}]$, 并且 $\mathbf{x}_k^{(i)} \in \mathbb{C}$ 。于是我们定义 $\mathbf{s}^{(i)}(k, v)$ 为 $\mathbf{x}^{(i)}$ 的第 (k, v) 个邻居向量, $\mathbf{s}^{(i)}(k, v)$ 定义如下:

$$\mathbf{s}^{(i)}(k, v) = [s_1^{(i)}(k, v), s_2^{(i)}(k, v) \cdots s_{N_k}^{(i)}(k, v)] \quad (3.1)$$

这里 k 表示发射天线数 ($k = 1, 2 \cdots K$), $v = 1, 2 \cdots L$ 。并且 $\mathbf{s}^{(i)}(k, v)$ 中任何一个元素满足下式:

$$\mathbf{s}_n^{(i)}(k, v) = \begin{cases} \mathbf{x}_n^{(i)} & n \neq k \\ \mathbf{w}_v(\mathbf{x}_k^{(i)}) & n = k \end{cases} \quad (3.2)$$

所以我们发现 $\mathbf{x}^{(i)}$ 和 $\mathbf{s}^{(i)}(k, v)$ 只有第 k 个元素不同, 而第 k 个元素正好是 $\mathbf{x}_k^{(i)}$ 的星座图上的第 v 个邻居, 解向量 $\mathbf{x}^{(i)}$ 邻域向量包含 $K \times L$ 个邻居向量。这里我们再根据上面 4QAM 调制下对发射向量邻域定义举例说明, 假设发射天线数为 $K = 3, L = 2$, 发射向量为 $\mathbf{x} = [c_1, c_2, c_3]^T$, 邻域内的邻居向量有: $\{\mathbf{x}_1 = [c_2, c_2, c_3]^T, \mathbf{x}_2 = [c_3, c_2, c_3]^T, \mathbf{x}_3 = [c_1, c_3, c_3]^T, \mathbf{x}_4 = [c_1, c_4, c_3]^T, \mathbf{x}_5 = [c_1, c_2, c_1]^T, \mathbf{x}_6 = [c_1, c_2, c_4]^T\}$, 共有 $K \times L = 6$ 个。

2. 禁忌表

首先简述一下向量“移动”过程, $\mathbf{x}^{(i)}$ 的第 (k, v) 个邻居向量移动到 $\mathbf{x}^{(i+1)}$ 中, 表达式为:

$$\mathbf{x}^{(i+1)} = \mathbf{s}^{(i)}(k, v) \quad (3.3)$$

这就称之为一次移动 ($move(k, v)$)。通过上面例子分析, 向量每次移动操作有 $K \times L$ 种可能, 所以根据调制阶数 Q 不同, 每次星座图上的点移动操作可能性有 $K \times Q \times L$ 种可能。

禁忌表实质是一个行数为 $K \times Q$, 列数为 L 的矩阵, 我们用 T 表示, 矩阵中每个元素表示对应向量移动的禁忌值。这里禁忌值是非负整数, 在每一次迭代过程中更新数值, 禁忌值决定搜索过程中的移动方向。下面结合上面例子做出 RTS 算法在发射天线数为 N_k , 调制方式为 4QAM 情况下的禁忌表。(见表 3.1) 这里因为选用 4QAM, 所以 $Q = 4, L = 3$ 。

3. 禁忌周期

RTS 算法禁忌周期我们用 P 表示, 这是一个动态的非负数整数, 具体定义如下: 如果某次迭代一个向量的移动被标记在禁忌表中, 那么之后的 P 次迭代该向量仍不作为待移动的对象, 继续保持禁忌状态, 直到移动得到的向量比当前解向量更优为止。综上所述, 如果将禁忌表看成一个队列的话, 那么队列的长度就是禁忌周期, 而 RTS 算法的主动适应性就是通过禁忌周期的动态改变体现的。

表 3.1 4QAM 下 RTS 算法禁忌表

第 L 个邻居向量 发射向量 \mathbf{x}_{N_K}		$L = 1$	$L = 2$	$L = 3$
$x_1 \in \mathbb{C}$	c_1	$t_{1,1}$	$t_{1,2}$	$t_{1,3}$
	c_2	$t_{2,1}$	$t_{2,2}$	$t_{2,3}$
	c_3	$t_{3,1}$	$t_{3,2}$	$t_{3,3}$
	c_4	$t_{4,1}$	$t_{4,2}$	$t_{4,3}$
$x_2 \in \mathbb{C}$	c_1	$t_{5,1}$	$t_{5,2}$	$t_{5,3}$
	c_2	$t_{6,1}$	$t_{6,2}$	$t_{6,3}$
	c_3	$t_{7,1}$	$t_{7,2}$	$t_{7,3}$
	c_4	$t_{8,1}$	$t_{8,2}$	$t_{8,3}$
\vdots	\vdots	\vdots	\vdots	\vdots
$x_{N_K} \in \mathbb{C}$	c_1	$t_{4(N_K-1)+1,1}$	$t_{4(N_K-1)+1,2}$	$t_{4(N_K-1)+1,3}$
	c_2	$t_{4(N_K-1)+2,1}$	$t_{4(N_K-1)+2,2}$	$t_{4(N_K-1)+2,3}$
	c_3	$t_{4(N_K-1)+3,1}$	$t_{4(N_K-1)+3,2}$	$t_{4(N_K-1)+3,3}$
	c_4	$t_{4N_K,1}$	$t_{4N_K,2}$	$t_{4N_K,3}$

4. 终止条件

为了将算法的复杂度控制在一定范围内，我们定义最大迭代次数 \max_iter 和最大重复次数 \max_rep 这两个变量。如果搜索过程迭代次数达到最大迭代次数，则算法终止，将当前最优解向量作为最终解向量输出；如果当前的解向量是局部最小值（通过定义 $flag \in \{0,1\}$ 作为局部最小标识符），不是全局最小值，但是最大重复次数大于最大迭代次数，也将当前过程中得到的最优解向量作为最终解向量输出。

5. 搜索迭代过程

在定义的邻域内搜索全局最优解向量的迭代过程可以分为三步：

步骤一：初始化定义局部最小标识符 $flag = 0$ （ $flag = 0$ 表示当前解向量为非局部最小值），禁忌表矩阵 $T = 0$ ，重复次数 $N_{rep} = 0$ ，两个重复解向量之间迭代次

数 $n_{rep} = 0$ ，禁忌周期为一正整数 ($P = P_0$)，经过线性检测算法求得初始解为 $\mathbf{x}^{(0)}$ ，而 $\mathbf{g}^{(0)} = \mathbf{x}^{(0)}$ ($\mathbf{g}^{(i)}$ 表示经过 i 次迭代后最优解向量)。根据式(2.35)我们可以求得 $\mathbf{x}^{(i)}$ 的 $N_K \times L$ 个邻居向量的 ML 代价函数为 $\phi(\mathbf{s}^{(i)}(k, v))$ ，这里 $k = 1, 2, \dots, N_K$ ， $v = 1, 2, \dots, L$ 。于是定义：

$$(k_1, v_1) = \arg \min \phi(\mathbf{s}^{(i)}(k, v)) \quad (3.4)$$

如果满足下面两式(3.5)和(3.6)中的任意一个，就将当前解向量 $\mathbf{x}^{(i)}$ 向第 (k_1, v_1) 个邻居向量 $\mathbf{s}^{(i)}(k_1, v_1)$ 移动。

$$\phi(\mathbf{s}^{(i)}(k_1, v_1)) < \phi(\mathbf{g}^{(i)}) \quad (3.5)$$

$$T((k_1 - 1)Q + q, v_1) = 0 \quad (3.6)$$

式(3.6)中 $q \in \{1, 2, \dots, Q\}$ ，如果不满足任何一个条件则依次对 (k_u, v_u) 进行判定， $u = 2, 3, \dots, N_K L$ 。如果所有邻居向量都被禁忌，找出禁忌表中最小禁忌值，并将表中所有禁忌值减去这个最小禁忌值，重复进行上述过程，直到找到移动向量为止，并作为下一次迭代解向量：

$$\mathbf{x}^{(i+1)} = \mathbf{s}^{(i)}(k', v') \quad (3.7)$$

这里 $\mathbf{s}^{(i)}(k', v')$ 为 (k', v') 对应的 ML 代价函数值最小的邻居向量。

步骤二：对上一步最终获得解向量进行重复次数验证。如果发现当前获得解向量和以前迭代的解向量重复，则 $N_{rep} = N_{rep} + 1$ ， $P = P + 1$ ，并同时更新 n_{rep} 。如果禁忌周期 P 大于 βn_{rep} ，则 P 取值如下：

$$P = \begin{cases} P + 1 & P \leq \beta n_{rep} \\ \max(1, P - 1) & P > \beta n_{rep} \end{cases} \quad (3.8)$$

式(3.7)体现了禁忌周期 P 的动态特性，并根据 $\phi(\mathbf{g}^{(i)})$ 和 $\phi(\mathbf{x}^{(i+1)})$ 对比，对最优解、禁忌表中禁忌值和局部最小值标识符进行更新：

$$\begin{cases} T((k' - 1)Q + q', v') = 0, \mathbf{g}^{(i+1)} = \mathbf{x}^{(i+1)}, flag = 0 & \phi(\mathbf{x}^{(i+1)}) < \phi(\mathbf{g}^{(i)}) \\ T((k' - 1)Q + q', v') = P + 1, \mathbf{g}^{(i+1)} = \mathbf{g}^{(i)}, flag = 1 & \phi(\mathbf{x}^{(i+1)}) \geq \phi(\mathbf{g}^{(i)}) \end{cases} \quad (3.9)$$

步骤三：按照下式更新禁忌表的禁忌值：

$$T(m, n) = \max\{T(m, n) - 1, 0\} \quad (3.10)$$

这里 m 表示禁忌表矩阵行数($m=1, 2 \dots N_k Q$), n 表示列数($n=1, 2 \dots L$)。此时如果满足上面算法终止条件, 则将输出当前解向量结束; 否则, 继续回到步骤一进行迭代。

3.1.2 RTS 算法和 LAS 算法对比

LAS 算法和 RTS 算法虽然都是基于本地邻域搜索的启发式算法, 但是通过前面对两种算法的分析, 我们可以发现以下几点不同:

(1) LAS 算法定义的邻域是静态的, 而 RTS 算法通过引入禁忌表, 使得存在于禁忌表中的邻居向量无法被移动, 每次迭代后禁忌值、禁忌周期又重新更新, 所以 RTS 算法定义的邻域是动态的。

(2) LAS 算法搜索迭代过程致命的缺陷是只找到局部最小值, 而非全局最小值; 而 RTS 算法通过引入“逃脱”策略(即使最优邻居向量不如当前解向量, 也允许当前解向量移动到最优邻居向量), 跳出局部最小值陷阱, 找到全局最优解向量。

上述从理论上分析了两种算法的不同, 接着从复杂度和性能对两种算法进行对比。

1. LAS 和 RTS 性能对比

图 3.3 为 4QAM 调制方式下 RTS 算法和 LAS 算法不同天线数的 BER 性能仿真结果对比。我们发现凭借着“逃脱”策略, RTS 性能明显优于 LAS, 更适用于大规模 MIMO 系统中。比如在误码率同为 4×10^{-5} 和天线数同为 128×128 的条件下, LAS 算法信噪比为 14dB, 而 RTS 算法信噪比为 12.5dB, 两种算法信噪比差距达到 1.5dB; 而且两种算法随着收发天线数的增加, BER 性能都相应的提升。

2. LAS 和 RTS 复杂度对比

和 LAS 算法相同, RTS 算法复杂度也主要有两方面组成: 初始解向量计算和搜索迭代过程。两种算法在初始解向量计算复杂度是完全相同的, 都是线性算法 ZF 和 MMSE 算法复杂度, 不同的是 RTS 算法在搜索迭代过程中引入“逃脱”策略, 从而走出局部死循环, 提高算法性能, 以此同时复杂度也相应提升。

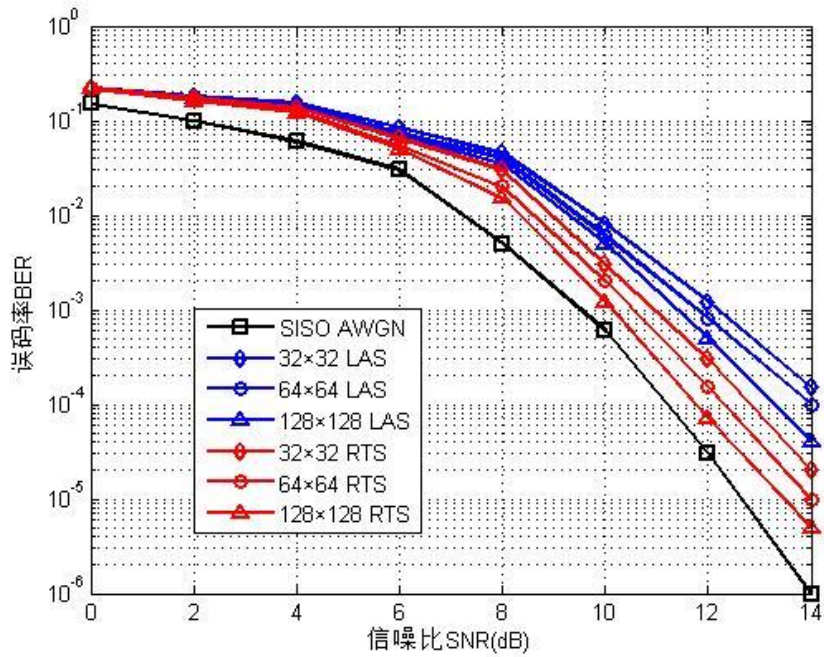


图 3.3 RTS 和 LAS 算法 BER 对比

图 3.4 为在 $BER=1.5 \times 10^{-2}$, 4QAM 调制方式下 RTS 和 LAS 算法计算复杂度对比, 并且加入两种算法迭代搜索过程复杂度作为对比。从图中我们验证了 RTS 算法相比 LAS 算法, 在提高 BER 性能的同时牺牲了一定的复杂度, RTS 整体复杂度比 LAS 算法高, 不过随着天线数增加, 两者复杂度相差逐渐减小。

同时我们发现初始解向量算法复杂度始终占据 RTS 算法整体复杂度最大的比例, 而且随着天线数量增加, 搜索迭代过程复杂度占整体复杂度比例越来越小, 即初始解向量复杂度占整体复杂度比例越来越大。比如, 在发射天线个数为 16, 也就是图中对应的横坐标为 4 时, RTS 算法整体复杂度为 9.2×10^4 , 而迭代搜索部分复杂度为 3.2×10^4 , 所以初始解向量复杂度占整体复杂的 65% 左右, 而迭代搜索部分的比例仅为 35% 左右; 当发射天线个数为 128, 图中横坐标对应的为 7 时, RTS 算法整体复杂度为 4.2×10^7 , 而迭代搜索部分复杂度为 5.2×10^6 , 初始解向量算法复杂度占整体达到 85% 左右, 迭代搜索部分的比例降为 15% 左右。因此, 研究如何降低初始解向量计算复杂度的工作, 对于 RTS 算法整体复杂度的降低也是至关重要的。

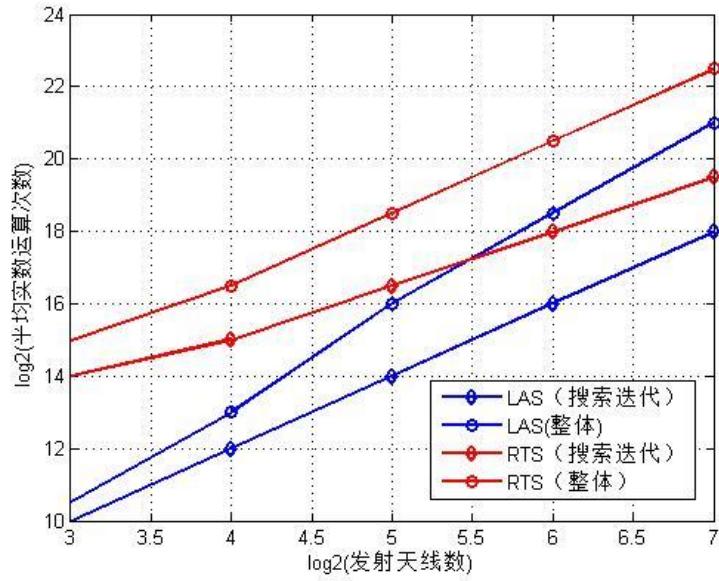


图 3.4 RTS 和 LAS 算法复杂度对比

3.2 GS-RTS 算法

通过 LAS 和 RTS 算法性能和复杂度的对比,我们发现 RTS 算法凭借着“逃脱”策略在牺牲一定复杂度的前提下,性能明显提高,更加适合大规模 MIMO 系统,所以成为本文重点研究算法。在 A.Chockalingam 的专著^[43]中指出 RTS 算法性能主要取决于邻域函数定义和初始值算法的选择。通常初始解向量算法使用的是最小均方误差 (MMSE) 算法,由于天线数量上升到上百根,MMSE 算法不可避免的涉及到矩阵的共轭转置求逆过程,所以基于 MMSE 求得初始值的 RTS 算法还是存在初始值计算复杂度过高的问题。基于此,为了避免大规模矩阵求逆过程,本文基于 GS 迭代法求解初始值的改进 RTS 算法,从近似值替代精确解的角度出发,经过一定的迭代次数,GS 算法可以高效地得到近似于 MMSE 算法精确解的值,同时计算复杂度大幅降低。本节首先介绍 GS 算法原理和优势特性,然后说明 GS-RTS 算法整体流程,并在下一节给出仿真结果和分析。

3.2.1 GS 算法

1. 算法原理

目前 RTS 算法初始解向量主要还是应用传统线性信号检测算法，比如最小均方差(MMSE)算法。根据第二章 MMSE 算法估计的发射信号向量 $\hat{\mathbf{x}}$ 表示为：

$$\hat{\mathbf{x}} = (\mathbf{H}^H \mathbf{H} + \sigma^2 \mathbf{I}_K)^{-1} \mathbf{H}^H \mathbf{y} \quad (3.11)$$

$$\hat{\mathbf{y}} = \mathbf{H}^H \mathbf{y} \quad (3.12)$$

$$\mathbf{W} = \mathbf{H}^H \mathbf{H} + \sigma^2 \mathbf{I}_K \quad (3.13)$$

$$\hat{\mathbf{x}} = \mathbf{W}^{-1} \hat{\mathbf{y}} \quad (3.14)$$

$$\mathbf{W} \hat{\mathbf{x}} = \hat{\mathbf{y}} \quad (3.15)$$

发射天线数为 K ，接收天线数为 M ，任意一个 $K \times 1$ 的非零发射向量 \mathbf{x} ，根据式(3.13)得到：

$$(\mathbf{H}\mathbf{x})^H \mathbf{H}\mathbf{x} = \mathbf{x}^H (\mathbf{H}^H \mathbf{H}) \mathbf{x} > 0 \quad (3.16)$$

这表明矩阵 $\mathbf{H}^H \mathbf{H}$ 为正定矩阵，噪声向量协方差矩阵 $\sigma^2 \mathbf{I}_K$ 也是正定的，所以矩阵 \mathbf{W} 也是正定矩阵。又因为：

$$(\mathbf{H}^H \mathbf{H})^H = \mathbf{H}^H \mathbf{H} \quad (3.17)$$

所以矩阵 $\mathbf{H}^H \mathbf{H}$ 也为对称矩阵，噪声向量协方差矩阵也是对称矩阵，所以矩阵 \mathbf{W} 也是对称矩阵，即矩阵 \mathbf{W} 为对称正定矩阵。

选取 MMSE 作为 RTS 算法初始解向量算法复杂度为 $O(K^3)$ ，随着大规模 MIMO 系统中小区用户数量增加，应用在实际工程中还是存在复杂度过高的问题。

在大规模 MIMO 系统中，随着发射和接收两端天线数量增加，天线间信道趋向正交，MMSE 算法的矩阵 \mathbf{W} 为对称正定性矩阵。基于这一特点，我们考虑使用 GS 迭代算法去求 MMSE 算法近似发射向量 \mathbf{x} ，从而降低初始值向量求解过程算法复杂度。

通常 GS 算法应用在求解 N 维线性方程：

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (3.18)$$

其中 \mathbf{A} 是 $N \times N$ 维 Hermitian 正定矩阵， \mathbf{x} 是 $N \times 1$ 解向量， \mathbf{b} 是 $N \times 1$ 向量。不同于传统方法直接计算 $\mathbf{A}^{-1}\mathbf{b}$ 得到 \mathbf{x} ，GS 方法可以通过迭代方法求解方程 $\mathbf{A}\mathbf{x} = \mathbf{b}$ ，从而降低复杂度。基于实数型大规模 MIMO 系统模型，MMSE 转换矩阵 \mathbf{W} 是 $2K \times 2K$ 维 Hermitian 正定矩阵，公式(3.15)和公式(3.18)也是一样类型，并且基于大

规模 MIMO 系统信道硬化的特性(矩阵 $\mathbf{H}^H \mathbf{H}$ 对角上的元素值比其他元素大很多), 我们可以分解 \mathbf{W} 为:

$$\mathbf{W} = \mathbf{D} + \mathbf{L} + \mathbf{U} \quad (3.19)$$

这里 \mathbf{D} 代表矩阵 \mathbf{W} 的对角元素组成的矩阵, \mathbf{L} 代表矩阵 \mathbf{W} 的严格下三角元素组成的矩阵, \mathbf{U} 代表矩阵 \mathbf{W} 的严格上三角元素组成的矩阵。将公式 (3.19) 代入公式 (3.15) 得到:

$$(\mathbf{D} + \mathbf{L} + \mathbf{U})\hat{\mathbf{x}} = \hat{\mathbf{y}} \quad (3.20)$$

$$(\mathbf{D} + \mathbf{L})\hat{\mathbf{x}} = \hat{\mathbf{y}} - \mathbf{U}\hat{\mathbf{x}} \quad (3.21)$$

从而发射向量估计值 \mathbf{x} 可以表示为:

$$\hat{\mathbf{x}} = (\mathbf{D} + \mathbf{L})^{-1} (\hat{\mathbf{y}} - \mathbf{U}\hat{\mathbf{x}}) \quad (3.22)$$

定义经过迭代 i 次的发射向量为 $\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)} \dots \mathbf{x}_{N_k}^{(i)})^T$, (当 $i=0$ 时, $\mathbf{x}^{(0)} = (0, 0 \dots 0)^T$, GS 迭代法默认初始值为零向量) 代入上式可以得到 GS 算法的迭代公式为:

$$\mathbf{x}^{(i+1)} = (\mathbf{D} + \mathbf{L})^{-1} (\hat{\mathbf{y}} - \mathbf{U}\mathbf{x}^{(i)}), i=0, 1, 2, \dots \quad (3.23)$$

当然这里为了加快 GS 算法的收敛, 可以乘以松弛因子 w ($0 < w < 2$)。

2. 性能仿真

本节通过几种迭代算法的实验仿真结果对比, 说明 GS 算法 BER 性能优劣。首先在 MATLAB 平台上根据大规模 MIMO 系统配置仿真参数: 我们考虑单小区情况下, 选取的基站天线数为 128, 小区用户天线数为 16, 调制方式为 4QAM, GS 迭代算法迭代次数为 4, 接收端译码算法为软输入 Viterbi, 卷积码速率为 1/2, 信道模型为上行链路系统模型(具体参数列表见表 3.2)。

然后根据表 3.2 的相应的参数配置, 对几种主流的迭代算法进行 BER 性能对比。从图 3.5 中可以看出 MMSE 性能最好, 当迭代次数达到一定的时候, Neumann、Richardson 和 GS 三种算法性能逐渐接近 MMSE。而在三种算法中, 不论迭代次数为多少, 性能都是 GS 算法最好, Richardson 居中, 而 Neumann 性能最差。比如迭代次数为 4, SNR(信噪比)为 14dB 时, MMSE 误码率为 1×10^{-6} , Neumann 为 5×10^{-6} , Richardson 为 2.1×10^{-6} , 而 GS 为 1.5×10^{-6} , GS 算法仅比 MMSE 高出 0.5×10^{-6} , 从而用实验仿真结果验证了 GS 算法性能优越。

表 3.2 GS 迭代算法仿真参数配置

参数名称	参数大小
小区个数	1
基站天线数	128
小区用户数	16
调制方式	4 QAM
卷积码速率	1/2
接收端译码算法	软输入 Viterbi
信道模型	瑞利衰落
最大迭代次数	4

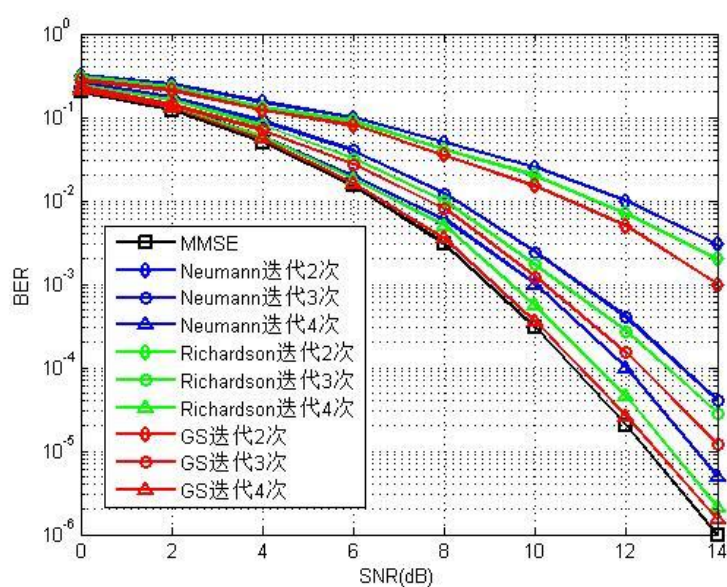
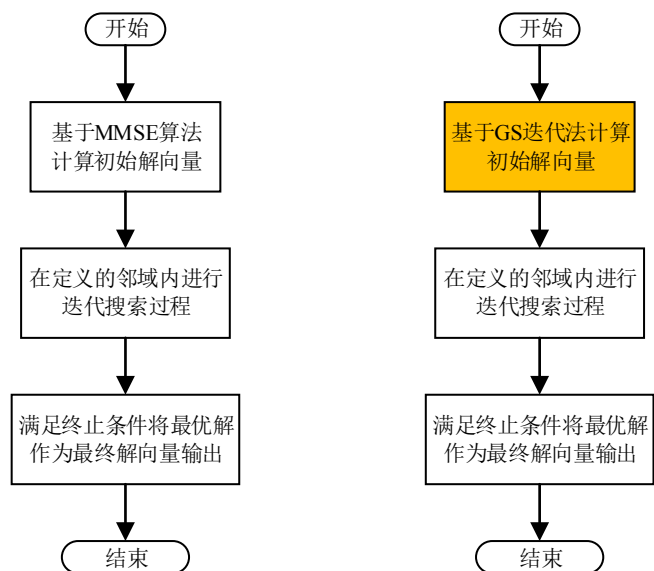


图 3.5 几种主流迭代算法 BER 对比

3.2.2 GS-RTS 算法流程

我们从前面 RTS 算法整体复杂度分析知道，RTS 算法复杂度主要来自于两方面（初始解求解过程和搜索迭代过程），可是随着天线增加，复杂度的第一部分初始解向量算法复杂度增速明显大于第二部分搜索迭代过程（见图 3.4），降低初始解向量计算复杂度成为不容忽视的问题。通过上一小节对 GS 算法的性能分析，GS 算法利用大规模 MIMO 出现的系统特性，通过迭代法可以得到近似于 MMSE 算法的性能，因此考虑引入 GS 迭代算法代替 MMSE 求解初始向量，从而降低 RTS 算法整体复杂度。如图 3.6 为 RTS 算法和改进的 GS-RTS 算法整体流程对比图，

从图中可以清楚地看出改进的 GS-RTS 算法相比于 RTS 算法，区别在于第一步骤替换为 GS 迭代法计算初始解向量。



(a) RTS 算法整体流程图 (b) GS-RTS 算法整体流程图
图 3.6 RTS 和 GS-RTS 算法流程对比

3.3 仿真与分析

根据 3.2 小节中的分析，为了验证基于 GS 迭代法求解初始解向量的改进 RTS 检测算法在性能达到大规模 MIMO 系统要求的前提下，复杂度进一步降低这个目标，本文借助 MATLAB 平台分别对 GS-RTS 算法 BER 性能、收敛性和复杂度进行了仿真，并对仿真结果进行了对比分析。

3.3.1 GS-RTS 算法性能仿真

首先依据大规模 MIMO 上行链路配置仿真参数：本文考虑单小区情况下，小区内用户数为 16，基站天线数最多 128 根，调制方式为 4QAM 和 16QAM，卷积码速率为 1/2，接收端译码算法为软输入 Viterbi，选取的是瑞利衰落信道模型。然后根据 GS 迭代法和 RTS 检测算法配置相关算法初始参数：禁忌周期 P 为 2，最大重复次数 \max_rep 为 75，RTS 算法最大迭代次数 \max_iter 为 80，GS 算法最大迭代次数为 4（由图 3.5 看出 GS 算法在迭代次数为 4 时最接近 MMSE 算法性能）。

接着在 MATLAB 平台上仿真运行，通过分析基站接收端的 BER（误码率）来检验 GS-RTS 算法的性能。在仿真结果中，x 轴为基站接收端的信噪比（Signal-to-Noise Ratio, SNR），y 轴基站接收端的 BER。详细的仿真条件如表 3.3 所示：

表 3.3 GS-RTS 算法仿真参数

参数类型	参数大小
小区个数	1
基站天线数	32; 64; 128
小区用户数	16
调制方式	4QAM; 16QAM
卷积码速率	1/2
接收端译码算法	软输入 Viterbi
信道模型	瑞利衰落
信号检测算法	RTS
禁忌周期	2
禁忌周期系数	0.1
最大重复数	75
最大迭代次数	80
初始解向量算法	GS
GS 迭代次数	4

这里针对上表小区用户天线数加以说明，大规模 MIMO 系统在基站端布置大规模天线阵列特性，带来的系统性能提升也主要来自于基站端，所以本文为了便于分析改进算法性能将初始仿真参数小区用户天线数固定为 16 根。

图 3.7 为 4QAM 调制方式下的仿真结果，为了分析算法在基站端不同天线数下的性能表现，基站端分别配置三种不同的天线数目：32，64，128。从图中，我们可以看出随着基站天线数量的增加，两种算法性能得到改善。比如 RTS 算法在 BER 同为 2×10^{-4} 时， 32×16 的信噪比为 16dB， 64×16 的信噪比为 14.5dB， 128×16 的信噪比为 13.2dB， 128×16 的信噪比相比 64×16 降低 1.3dB，相比 32×16 的信噪比降低 2.8dB；而 GS-RTS 算法在 BER 同为 1×10^{-3} 时， 32×16 的信噪比为 14dB， 64×16 的信噪比为 12.8dB， 128×16 的信噪比为 11.3dB， 128×16 的信噪

比相比 64×16 降低 1.5dB, 相比 32×16 的信噪比降低 2.7dB。虽然改进的 RTS 算法复杂度降低很多（见 3.3.2 小节分析），可是可以看出 BER 性能却没有受到很大影响，不论基站天线数的改变，始终 BER 性能都是很接近原来的 RTS 算法，并且随着天线数量增加，GS-RTS 算法 BER 性能越来越接近原来算法。比如在 SNR 为 16dB 时 32×16 的改进 RTS 的 BER 只比原来的 RTS 算法高 1.9×10^{-4} ， 64×16 的改进 RTS 的 BER 比原来的 RTS 算法高 5×10^{-5} ，而天线数 128×16 的改进 RTS 算法的 BER 只比原来的 RTS 算法高 1.6×10^{-5} 。

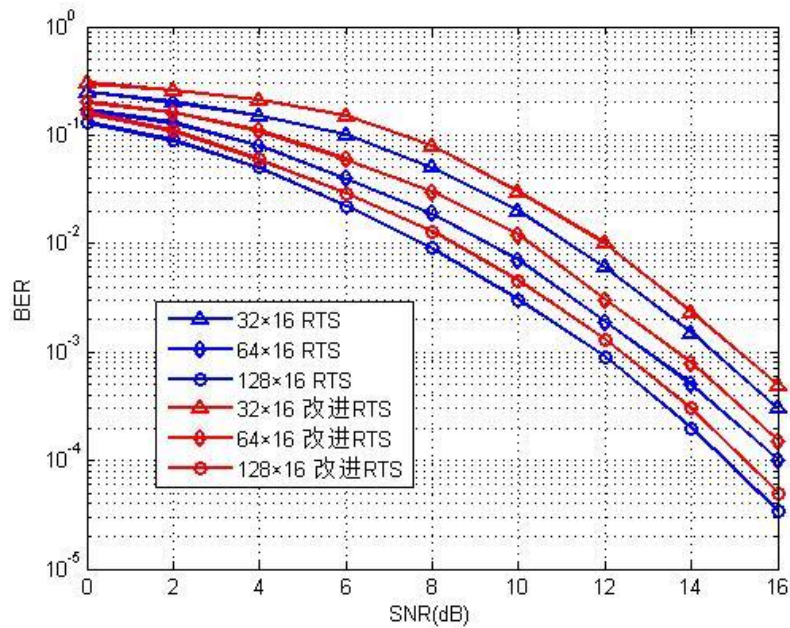


图 3.7 4QAM 下 RTS 算法和改进的 GS-RTS 算法 BER 对比

为了更好地验证在更高阶调制方式下 GS-RTS 算法检测性能，本文在 16QAM 调制方式情况下对不同基站天线数目检测算法的性能进行了仿真，同样基站天线数目分别为 32, 64, 128。图 3.8 为 16QAM 调制方式下的 BER 仿真结果，从图中我们同样可以看出基站天线数量的增加对两种算法性能都有提升作用，比如 RTS 算法在 BER 同为 1.5×10^{-3} 时， 32×16 的信噪比为 16dB， 64×16 的信噪比为 15dB， 128×16 的信噪比为 13.8dB， 128×16 的信噪比相比 64×16 降低 1dB，相比 32×16 的信噪比降低 2.2dB；而 GS-RTS 算法在 BER 同为 4×10^{-3} 时， 32×16 的信噪比为 16dB， 64×16 的信噪比为 15dB， 128×16 的信噪比为 14.1dB， 128×16 的信噪比相比 64×16 降低 0.9dB，相比 32×16 的信噪比降低 1.9dB。

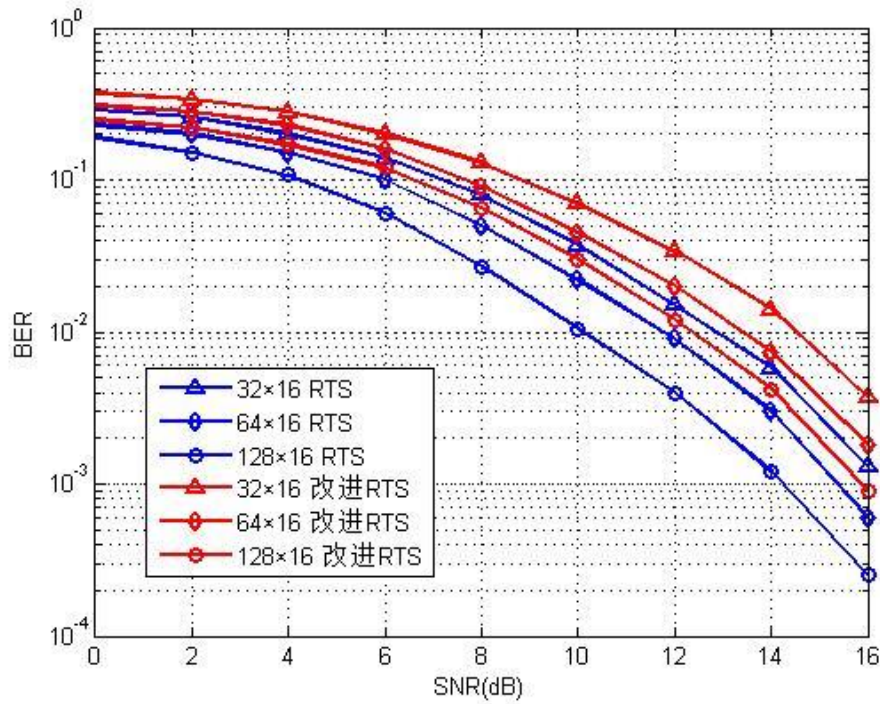


图 3.8 16QAM 下 RTS 算法和改进的 GS-RTS 算法 BER 对比

结合图 3.7-3.8 中不同调制方式下 RTS 算法和改进算法检测性能,可以看出相同的基站天线数下 4QAM 的两种算法性能明显优于 16QAM 调制方式。比如,在 SNR 为 16dB,收发天线都为 128×16 ,调制方式为 4QAM 的 RTS 算法 BER 为 2.7×10^{-5} ,16QAM 情况下 BER 为 6.1×10^{-4} ;4QAM 的 GS-RTS 算法 BER 为 9×10^{-4} ,16QAM 的 BER 为 4.2×10^{-5} 。还可以看出随着调制方式从 4QAM 变为 16QAM,GS-RTS 算法性能和原来的 RTS 算法性能差距越来越大。比如在 SNR 为 16dB,天线数都为 128×16 条件下,调制方式为 4QAM 时,两种算法性能差距为 1.6×10^{-5} ,而当调制方式为 16QAM 时,两种算法性能差距为 6.5×10^{-5} ,扩大了三倍多。

综合以上仿真结果分析可知,随着天线数量的增加,基于高斯赛德尔迭代法的改进 RTS 信号检测算法性能逐渐接近原来 RTS 算法;不论 RTS 算法还是 GS-RTS 改进算法其检测性能受调制方式影响都比较大,在高阶调制的模式下,性能都会有明显的下降,而 GS-RTS 受到的影响更加明显。

3.3.2 GS-RTS 算法复杂度分析

为验证 GS-RTS 的复杂度优势，本节将分析对比 GS-RTS 与传统 RTS 的计算复杂度，也即 GS 与 MMSE 算法复杂度的对比。根据公式(3.23)，可以将 GS 算法迭代方程变化为：

$$(\mathbf{D} + \mathbf{L})\mathbf{x}^{(i+1)} = \hat{\mathbf{y}} - \mathbf{U}\mathbf{x}^{(i)} \quad (3.24)$$

上式中左式 $2K \times 2K$ 维矩阵 $(\mathbf{D} + \mathbf{L})$ 和 $2K$ 维矩阵 $\mathbf{x}^{(i+1)}$ 相乘次数为 $2K^2 + K$ ，等式右边 $2K \times 2K$ 维矩阵 \mathbf{U} 和 $2K \times 1$ 维矩阵 $\mathbf{x}^{(i)}$ 相乘次数为 $2K^2 - K$ ，所以迭代一次 GS 算法需要 $4K^2$ 次相乘，而 MMSE 算法复杂度为 K^3 ，具体两种算法对比如下表 3.4 所示。

表 3.4 GS 和 MMSE 算法计算复杂度对比

算法名称	计算复杂度
MMSE 算法	K^3
迭代 2 次 GS 算法	$8K^2$
迭代 3 次 GS 算法	$12K^2$
迭代 4 次 GS 算法	$16K^2$

引入 GS 迭代法求解初始值的改进 RTS 算法，在初始解向量方面复杂度从原来的 $O(K^3)$ 下降一个级别到 $O(K^2)$ 。比如下图 3.9 中，经过两次迭代 GS 算法复杂度为复杂度为 $8K^2$ 。从图 3 中，我们可以发现随着小区用户数（这里默认小区用户都为单天线用户，小区用户数等同于小区用户天线数）增加，MMSE 和 GS 算法复杂度都有所增加，而 GS 算法随着迭代次数增加复杂度也会相应的增加，但是不论 GS 算法迭代次数增加到多少($2 < i < 4$)还是小区用户数增加多少，MMSE 算法的复杂度都是最高的。比如在小区用户数为 35 的条件下，MMSE 复杂度为 4.25×10^4 ，而迭代次数为 4 的 GS 算法复杂度为 1.4×10^4 。

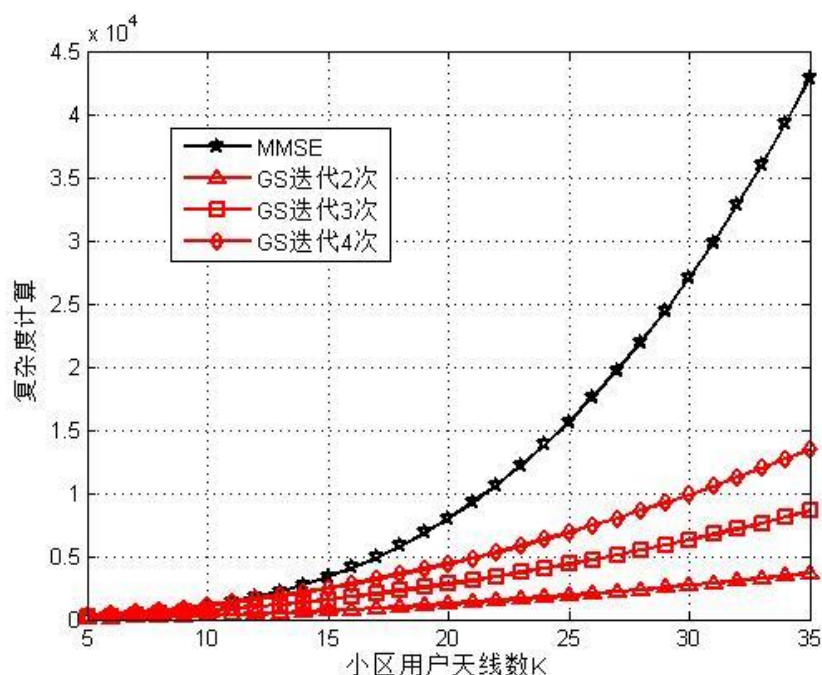


图 3.9 GS 和 MMSE 算法复杂度对比

3.3.3 GS-RTS 算法收敛性分析

下面分析改进的 GS-RTS 整体算法的收敛性，我们通过不同迭代次数（这里迭代次数是指 RTS 算法迭代搜索部分迭代次数，而非 GS 迭代法的迭代次数，表 3.4 已经限定 GS 迭代法迭代次数为 4 次）下算法性能的变化来验证这一性质。在表 3.4 配置的基础上，我们又增加小区用户天线数为 32 这一情况。从图 3.10 中，我们可以看出随着迭代次数的增加，不论天线数为 32×16 ， 64×16 还是 128×16 性能都呈现出收敛性，特别是迭代次数超过 70 以后，这一趋势更加明显。比如，当天线数为 128×16 ，迭代次数为 70 时，改进算法 BER 为 1.1×10^{-4} ，迭代次数增加到 100 时，BER 为 0.7×10^{-4} ，只比迭代 70 时低 4×10^{-5} 。所以随着迭代次数的递增，改进算法性能表现出稳定和优良的趋势。这里我们选取单输入单输出(Single Input Single Output, SISO)系统的加性高斯白噪声(Additive White Gaussian Noise, AWGN)信道性能作为参考标准。

最后针对小区用户天线数对算法性能的影响做简单描述。从图 3.10 中，通过对比天线数 128×16 和 128×32 的 BER，可以看出随着小区用户天线数从 16 增加到 32 性能有所提升。比如在迭代次数为 70 时， 128×32 的 BER 比 128×16 低 3

$\times 10^{-5}$ 。所以小区用户天线数量的增加也可带来性能的改善,不过牺牲的是复杂度的相应提升(见图3.8)。

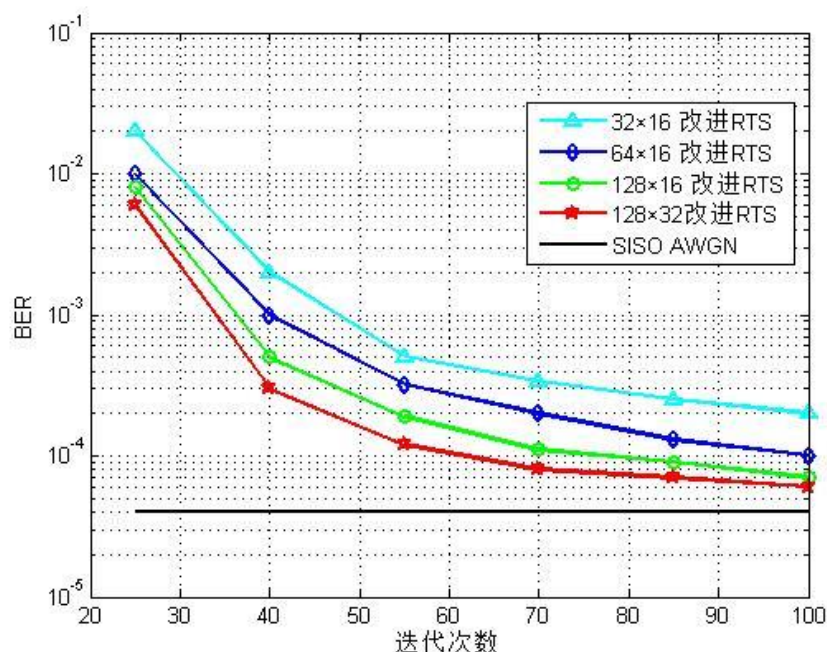


图 3.10 不同迭代次数下改进的 GS-RTS 算法 BER 对比

3.4 本章小结

本章主要研究了一种基于GS迭代法的改进RTS信号检测算法,即在RTS算法的第一步骤,通过高斯赛德尔迭代算法替换原来的MMSE算法,作为初始解向量算法。相比于原来的RTS检测算法,改进的RTS算法可以在性能只收到很小损失的前提下,大幅降低初始解算法的计算复杂度,从而降低算法整体复杂度。本章首先描述了RTS检测算法的基本原理,然后根据LAS算法和RTS算法对比仿真分析,突出RTS算法性能优越特性。接着根据对RTS算法整体复杂度的分析,发现初始解向量算法复杂度过高的问题,于是给出GS迭代法最为初始解向量的GS-RTS优化算法。最后通过实验仿真从性能、复杂度和收敛性三方面对改进算法进行验证,根据仿真结果分析,GS-RTS算法在低阶调制模式下性能达到下一代通信系统要求,而在高阶调制方式下性能还有待提高。

第4章 基于块分星座图的改进 GS-RTS 检测算法

上一章中基于 GS 迭代法求解初始解向量的改进 RTS 检测算法具有复杂度低、检测性能好的优点，但是对于链路采用高阶调制方式的情况，改进算法的检测性能还有待提高，因此 GS-RTS 检测算法适用于链路用户采用低阶调制方式的情况下。为了解决高阶调制方式下 GS-RTS 算法性能差的问题，本章将在 GS 迭代法基础上引入块分星座图(Block Constellation, BC)方法，即基于块分星座图的 GS 迭代法作为 RTS 的初始解向量(BC-GS-RTS)算法，从而使 GS-RTS 算法更好地应用在高阶调制模式下。

4.1 块分星座图方法

为了满足下一代移动通信系统的实际需求，大规模 MIMO 系统上行链路必须具备承载更多用户以及处理更加海量的数据的能力。使用高阶调制方式是解决这一问题的有效方式之一^[44]，可是随着小区用户数和基站天线数增加，大规模天线阵列下的信号检测算法在高阶调制方式下又出现性能较差的问题^[45]。

通过上一章对 GS 迭代算法的研究，我们发现 GS 算法的默认初始值为零向量矩阵，于是我们运用块分星座图方法求解 GS 算法的初始解向量，增加 GS 算法检测速率，缩短 GS 算法收敛次数，提高 GS 算法 BER 性能，从而进一步提高 GS-RTS 算法整体性能，使之更加适合大规模 MIMO 系统上行链路的高阶调制方式。下面具体说明块分星座图方法原理以及实现流程。

4.1.1 BC 方法原理

从 3.2.1 小节可知，小区用户数为 K 的条件下，通常 GS 迭代算法的初始解向量 $\mathbf{x}^{(0)} = (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)} \cdots \mathbf{x}_K^{(0)})^T$ 为 $2K \times 1$ 的零向量矩阵。定义 \mathbf{x}_i 和 $\hat{\mathbf{y}}_i$ 为发射向量 \mathbf{x} 和接收向量 $\hat{\mathbf{y}}$ 的第 i 个元素，而 \mathbf{w}_{ij}^{-1} 表示矩阵 \mathbf{W} 的第 i 行和第 j 列元素。根据第二章提到的，大规模 MIMO 系统随着收发天线数目增多，出现信道硬化的趋势，即矩阵 $\mathbf{H}^H \mathbf{H}$ 为对角元素占优矩阵（对角元素相对于非对角元素数值大很多）。由第三章又知

道 MMSE 算法的矩阵 \mathbf{W}^{-1} 既是对角占优矩阵，又是对称正定矩阵，对角元素都是正数。所以我们得出下式：

$$\hat{\mathbf{x}}_i \hat{\mathbf{y}}_i = \left(\sum_{j=1}^{2K} \mathbf{W}_{ij}^{-1} \hat{\mathbf{y}}_i \right) \hat{\mathbf{y}}_i > 0 \quad (4.1)$$

其中 $\hat{\mathbf{x}}_i$ 为发射向量 \mathbf{x}_i 的算法估计值， $\mathbf{y} = \mathbf{H}^H \mathbf{y}$ ，根据上式我们可以通过接收向量 $\hat{\mathbf{y}}_i$ 判断发射向量 \mathbf{x}_i 的正负，进一步根据 $\hat{\mathbf{y}}$ 判断初始解向量 $\mathbf{x}^{(0)}$ 每一个元素的范围：

$$\mathbf{z} = \hat{\mathbf{y}} - \mathbf{W} (a_1, a_2 \cdots a_{2K})^T \quad (4.2)$$

上式 a 代表一个常数定值，我们可以通过 z_i 知道 $\hat{\mathbf{x}}_i$ 是否大于常数 a ，从而确定 $\hat{\mathbf{x}}_i$ 所在星座图的范围。

正是基于这一点我们产生块分星座图的思想，首先将映射的星座图划分为若干块，然后界定每块区域的边界值，接着根据公式(4.1)和(4.2)确定发射向量 \mathbf{x}_i 属于哪个区域，并选取该区域的中心点最为初始解向量，替代原来的零向量最为初始解向量。具体算法实现过程如表 4.1。

表 4.1 块分星座图方法实现过程

步骤	操作
1	初始化：定义划分的星座图个数 N ，并根据选取的调制方式确定整数 C （后续会对 C 的取值举例说明），根据公式（3.12）计算 $\hat{\mathbf{y}}$ ；
2	根据下式定义划分区域的边界范围（即常数定值 a ）： $ a = \frac{2mC}{N}, m = 1, 2, \dots, \frac{N}{2} - 1; \quad (4.3)$
3	通过 $\hat{\mathbf{y}}_i$ 的正负，根据公式（4.1）和（4.2）确定 $\hat{\mathbf{x}}_i$ 所在区域： 如果 $\hat{\mathbf{y}}_i > 0$ ， $\mathbf{x}_i^{(0)} \in [2mC/N, 2(m+1)C/N]$ ； 如果 $\hat{\mathbf{y}}_i < 0$ ， $\mathbf{x}_i^{(0)} \in [-2(m+1)C/N, -2mC/N]$ ； 如果 $\hat{\mathbf{y}}_i = 0$ ， $\mathbf{x}_i^{(0)} = 0$ ；
4	选取所在区域的中心点作为 $\mathbf{x}_i^{(0)}$ ；
5	结束。

下面通过具体实例对上表说明块分星座图思想定义初始值过程。假设我们选取 64QAM 调制方式，首先确定 \mathbf{x}_i 映射星座图的实数轴数值。 \mathbf{x}_i 在实数轴的取值可能是 $\{-7, -5, -3, -1, +1, +3, +5, +7\}$ ，即所有可能取值数为 $C=8$ 。定义 $N=4$ ，也就是将 64QAM 映射的星座图块分为四个小区域，所以根据步骤 2 得到区域边界值 $a = -4, 4$ 。然后根据 $\hat{\mathbf{y}}_i$ 的值确定 $\hat{\mathbf{x}}_i$ 所在区域，比如假设 $\hat{\mathbf{y}}_i > 0$ ，并且 $\hat{\mathbf{y}}_i - \mathbf{W}a > 0$ ， $\hat{\mathbf{x}}_i$ 范围在 $[0, 4]$ 之间，我们就选取 $\mathbf{x}_i^{(0)} = 2$ ；如果 $\hat{\mathbf{y}}_i - \mathbf{W}a < 0$ ， $\hat{\mathbf{x}}_i$ 范围就在 $[4, 8]$ 之间，我们选取 $\mathbf{x}_i^{(0)} = 6$ ；当 $\hat{\mathbf{y}}_i < 0$ 则反之。而 \mathbf{x}_i 映射星座图的虚数轴取值和实数轴取值一样的过程，这里就不在叙述。通过下图 4.1 我们可以清楚的发现通过对 GS 算法引入块分星座图思想，初始值向量到发射向量距离 d_1 明显比零向量到发射向量距离 d_2 小，从而提高精确度，减少迭代次数，提高算法性能。

从上述具体介绍块分星座图方法的实现过程，我们可以看出如果选用 4QAM 调制方式，块分星座图和常规方法取值并无区别。而随着调制阶数增加，从 16QAM 增加到 64QAM 或者更高，块分星座图方法性能优势越来越明显。

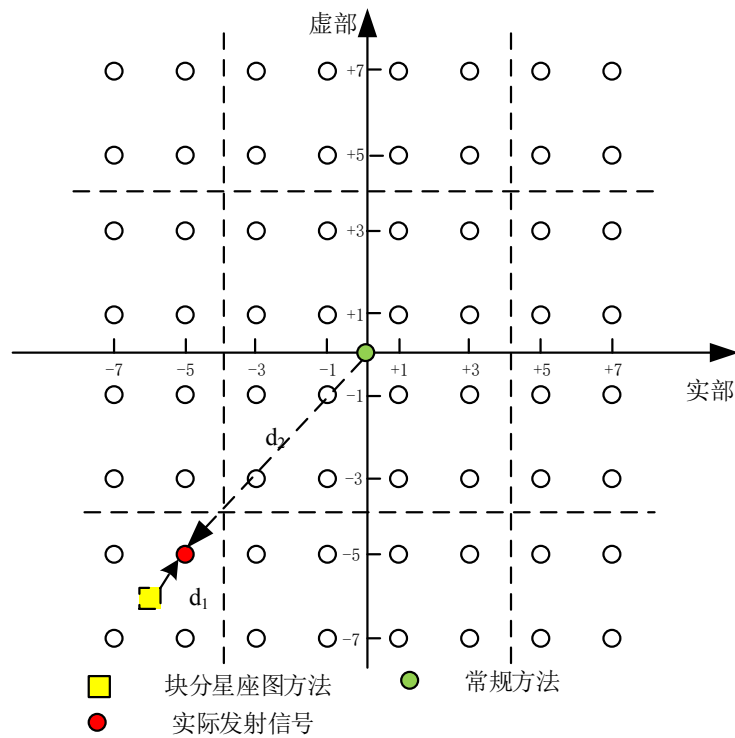


图 4.1 块分星座图与常规方法对比

4.2 BC-GS-RTS 算法

通过上面块分星座图方法和常规取零向量方法的对比，我们发现块分星座图方法凭借着划分映射星座小区域的做法，使得初始解向量在星座图上的星座点更加接近实际发射信号，从而加快 GS 迭代算法收敛速率。这种做法特别适合高阶调制方式下，基于这一点，本文考虑在 GS-RTS 算法基础上使用块分星座图方法，从而使得 GS-RTS 可以更好地应用在更高的调制方式中。本节首先介绍基于块分星座图的 GS 初始解向量算法，并通过仿真对比验证块分星座图方法对于 GS 迭代法在高阶模式下性能的提升，然后阐述 BC-GS-RTS 整体算法流程。

4.2.1 BC-GS 算法

1. BC-GS 算法流程

根据上一章对于 GS 算法原理的了解，以及前面对于块分星座图(BC)方法实现的叙述，我们可以得到基于块分星座图方法的 BC-GS 算法流程如下表 4.2：

表 4.2 BC-GS 算法实现流程

步骤	操作
1	将 MMSE 矩阵 \mathbf{W} 分解为 $\mathbf{W} = \mathbf{D} + \mathbf{L} + \mathbf{U}$ （ \mathbf{D} 代表矩阵 \mathbf{W} 的对角元素组成的矩阵， \mathbf{L} 代表矩阵 \mathbf{W} 的严格下三角元素组成的矩阵， \mathbf{U} 代表矩阵 \mathbf{W} 的严格上三角元素组成的矩阵）；
2	运用块分星座图方法求解 GS 迭代法初始解向量 $\mathbf{x}^{(0)}$ （本步骤具体实现见表 4.1）；
3	根据下式计算经过 i 次迭代发射信号向量 $\mathbf{x}^{(i)}$ ： $\mathbf{x}^{(i)} = (\mathbf{D} + \mathbf{L})^{-1} (\hat{\mathbf{y}} - \mathbf{U}\mathbf{x}^{(i-1)}), i = 0, 1, 2, \dots \quad (4.4)$
4	迭代次数递增，即 $i = i + 1$ ；
5	重复上述步骤一至步骤四，得到向量序列 $\mathbf{x}^{(i)}$ ；
6	结束。

2. 性能仿真

首先在 MATLAB 平台上根据大规模 MIMO 系统和 GS 算法原理配置仿真参数：我们选取基站天线数为 128，小区用户天线数为 16，调制方式为 16QAM，信道模型为上行链路系统模型，最大迭代次数为 4。（具体参数见表 4.3）

表 4.3 BC-GS 算法仿真参数

参数名称	参数大小
小区个数	1
基站天线数	128
小区用户数	16
调制方式	16QAM
卷积码速率	1/2
接收端译码算法	软输入 Viterbi
信道模型	瑞利衰落
最大迭代次数	4

本节通过对比 BC-GS 和 GS 算法的实验仿真结果，验证块分星座图方法对于 GS 算法性能的优化以及 BC-GS 算法更加适用于高阶调制方式。下面分别对 MMSE，GS 和 BC-GS 三种初始解向量算法在 16QAM 调制方式下进行 BER 对比，从而验证优化的 GS 算法性能进一步提高。在图 4.2 中可以看出 MMSE 的 BER 性能最好，随着迭代次数增加，GS 和 BC-GS 算法的 BER 性能逐渐提高。而凭借着块分星座图思想求解初始值的 BC-GS 算法在同样迭代次数条件下，BER 性能明显好于 GS 算法。例如当 SNR 为 12dB 时，迭代 4 次的 BC-GS 算法的 BER 仅比 MMSE 算法高 2×10^{-6} 。同时结合上一章图 3.5，我们发现随着调制方式从 4QAM 换成 16QAM，GS 算法性能与 MMSE 算法性能差距越来越大，而基于块分星座图方法的 BC-GS 算法随着迭代次数增加，仍然可以得到十分接近 MMSE 算法的性能。

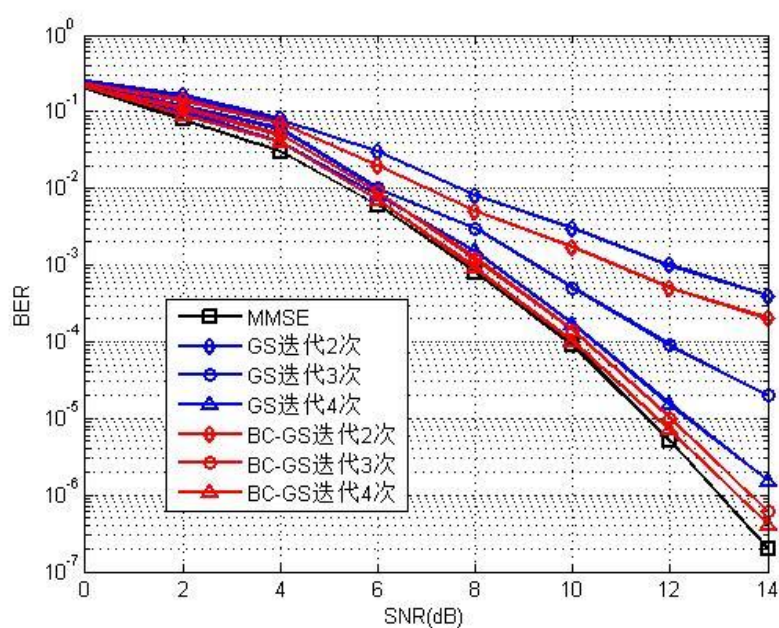


图 4.2 三种算法 BER 对比

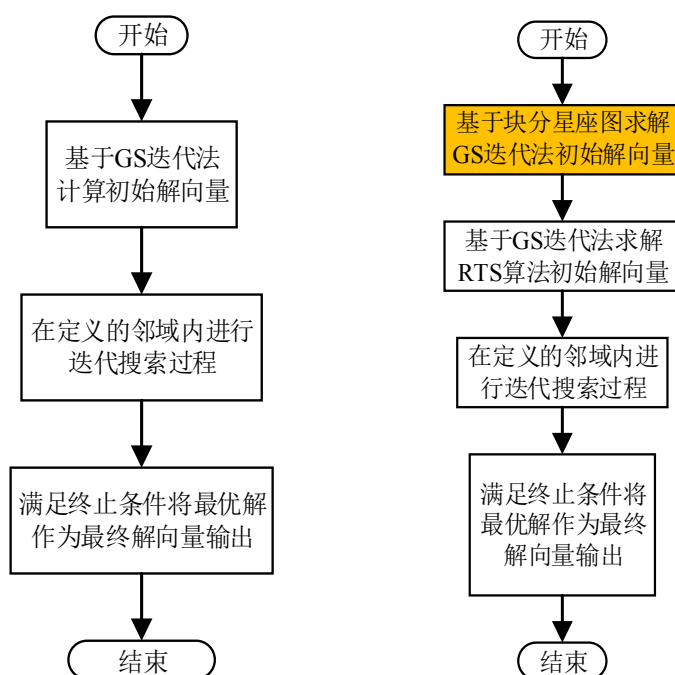


图 4.3 BC-GS-RTS 和 GS-RTS 算法流程对比

4.2.2 BC-GS-RTS 算法流程

图 4.3 为 GS-RTS 算法和改进的 BC-GS-RTS 算法整体流程对比图，从图中可以明显地看出改进的 BC-GS-RTS 算法相比于 GS-RTS 算法，区别在于增加了基于

块分星座图方法求解 GS 迭代法初始解向量的步骤。通过上一小节对 BC-GS 算法的性能分析，利用块分星座图的方法，BC-GS 算法在 16QAM 调制方式下可以得到近似于 MMSE 算法的性能，相比于 GS 算法，BER 性能得到明显的提升。因此考虑引入块分星座图方法，从而使 GS-RTS 算法更好地应用在高阶调制中。

4.3 仿真与分析

基于 4.2 节中的分析，为了验证高阶调制方式下基于块分星座图方法的 GS-RTS 检测算法性能，本文借助 MATLAB 平台对 BC-GS-RTS 检测算法的检测性能、复杂度和收敛性进行仿真，并对实验结果参照传统 RTS 算法和 GS-RTS 算法进行了对比分析。

4.3.1 BC-GS-RTS 性能仿真

首先依据大规模 MIMO 系统上行链路，配置仿真链路参数：本文考虑的是单小区情况下，和上一章相同小区用户数为 16 个，基站天线数为 32 根、64 根和 128 根，调制方式根据 LTE-A 上行链路配置为 16QAM，卷积码速率为 1/2，接收端译码算法为软输入 Viterbi，仿真选取的是瑞利衰落信道模型。然后根据块分星座图方法和 GS-RTS 检测算法配置相关算法初始参数：禁忌周期 P 为 2，最大重复次数 \max_rep 为 75，最大迭代次数 \max_iter 为 80，GS 算法最大迭代次数为 4，GS 迭代法初始解向量选取块分星座图方法，并根据块分星座图定义划分区域个数为 4。接着在 MATLAB 平台上仿真运行，通过分析基站接收端的 BER 误码率来检验 BC-GS-RTS 算法的性能。在仿真结果中，x 轴为基站接收端的 SNR，y 轴基站接收端的 BER。详细的仿真条件如表 4.4 所示：

图 4.4 为 16QAM 调制方式下 RTS、GS-RTS 和 BC-GS-RTS 三种算法的仿真结果，为了分析算法在基站端不同天线数下的性能表现，基站端分别配置三种不同的天线数目：32，64，128。从图中，我们可以看出随着基站天线数量的增加，三种算法性能得到改善，比如 RTS 算法在 BER 同为 1.5×10^{-3} 时， 32×16 的信噪比为 16dB， 64×16 的信噪比为 14.8dB， 128×16 的信噪比为 13.5dB， 128×16 的信噪比相比 64×16 降低 1.3dB，相比 32×16 的信噪比降低 2.5dB；GS-RTS 算法在 BER 同为 4×10^{-3} 时， 32×16 的信噪比为 16dB， 64×16 的信噪比为 14.9dB，

128×16 的信噪比为 14.1dB，128×16 的信噪比相比 64×16 降低 0.8dB，相比 32×16 的信噪比降低 1.9dB；而 BC-GS-RTS 算法在 BER 同为 3×10^{-3} 时，32×16 的信噪比为 16dB，64×16 的信噪比为 14.5dB，128×16 的信噪比为 13.2dB，128×16 的信噪比相比 64×16 降低 1.3dB，相比 32×16 的信噪比降低 2.8dB

表 4.4 BC-GS-RTS 算法仿真参数

参数类型	参数大小
小区个数	1
基站天线数	32; 64; 128
小区用户数	16
调制方式	16QAM
卷积码速率	1/2
接收端译码算法	软输入 Viterbi
信道模型	瑞利衰落
信号检测算法	RTS
禁忌周期	2
禁忌周期系数	0.1
最大重复数	75
最大迭代次数	80
初始解向量算法	GS
GS 迭代法次数	4
GS 初始解算法	BC
划分区域个数	4

结合上一章图 3.8 和图 4.4，我们发现同为 16QAM 的调制方式，三种算法 RTS 的 BER 性能最优，BC-GS-RTS 算法性能居中，而 GS-RTS 算法相对最差。更重要的是，我们发现通过对于 GS-RTS 算法的优化（引入块分星座方法），BC-GS-RTS 算法在 16QAM 高阶调制方式的 BER 性能显著提升，随着天线增加不断缩小与原来的 RTS 算法性能差距。比如在 SNR 为 16dB，天线数目同为 128×16 时，RTS 算法的 BER 为 2.5×10^{-4} ，BC-GS-RTS 算法的 BER 为 4.2×10^{-4} ，而 GS-RTS 算法的 BER 为 9×10^{-4} 。BC-GS-RTS 算法和原来的 RTS 算法性能差距为 1.7×10^{-4} ，而

GS-RTS 算法和原来的 RTS 算法性能差距为 6.5×10^{-4} ，所以 BC-GS-RTS 算法和 RTS 算法的性能差距仅为 GS-RTS 算法和 RTS 算法性能差距的四分之一。

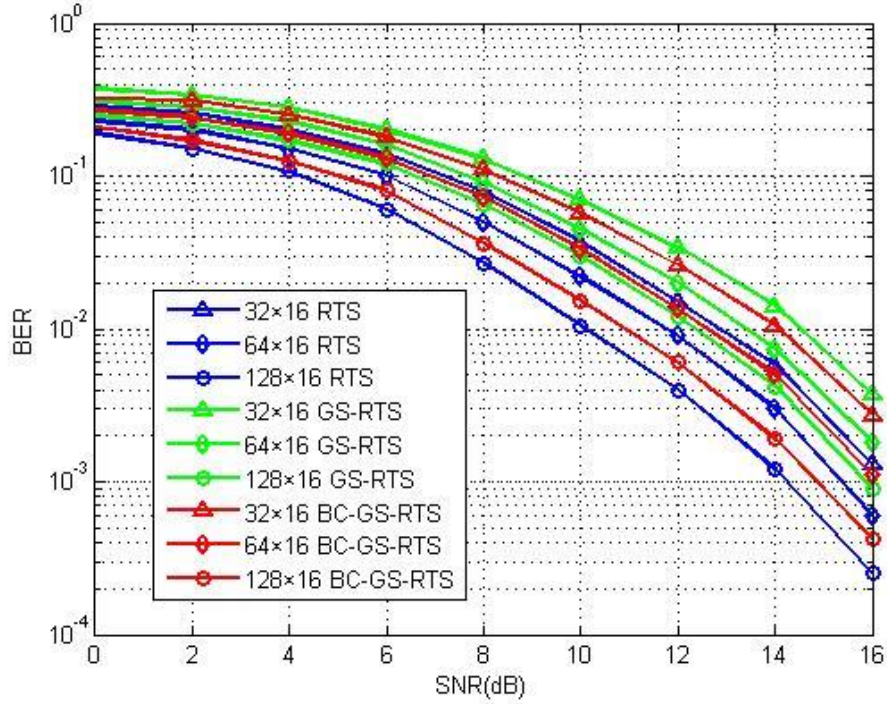


图 4.4 16QAM 下三种算法 BER 对比

综上所述，三种算法的 BER 性能对比，我们可以总结出通过引入块分星座方法，即使在更高阶的调制方式（相对于第三章的 4QAM）下，BC-GS-RTS 算法仍可以得到近似于原来 RTS 算法的 BER，从而验证了前面理论方面分析块分星座图方法对于 GS-RTS 性能的提升。

4.3.2 BC-GS-RTS 复杂度分析

对 GS-RTS 算法的改进主要是对初始解向量算法进行优化，即通过块分星座方法来替换常规零向量方法作为初始解向量算法，上一节通过仿真验证了块分星座图方法对于 GS-RTS 算法在高阶调制模式下性能的优化，本节将从复杂度层面对 BC-GS-RTS 算法进行分析。根据上一节复杂度分析可以推测 BC-GS-RTS 和 GS-RTS 算法复杂度的对比其实也就是 BC-GS 算法和 GS 算法的对比，而 BC-GS 算法和 GS 算法复杂度的差别就是计算块分星座图方法带来多少的复杂度。

BC-GS 算法复杂度主要由两部分组成：（1）GS 迭代算法的复杂度；（2）块分星座图方法的复杂度。第一部分复杂度上一章已经计算得出为 $i \times 4K^2$ （ i 表示 GS 算法迭代次数），而第二部分算法复杂度主要集中在 $\sum_{i=0}^{2K} \hat{\mathbf{y}}_i - \mathbf{W}\mathbf{a}_i$ 这一步骤，因为 MMSE 的矩阵 \mathbf{W} 为一个 $2K \times 2K$ 维矩阵，而 \mathbf{a} 为一个 $2K \times 1$ 维常数矩阵，所以相乘次数为 $2K^2$ 。

综上所述进行一次 BC-GS 算法所需 $4K^2 + 2K^2$ 次计算，GS 算法迭代一次复杂度为 $4K^2$ ，而 MMSE 算法复杂度为 K^3 。具体三种算法对比如下表 4.5 所示。

表 4.5 三种初始解算法计算复杂度对比

算法名称	计算复杂度
MMSE 算法	K^3
迭代 2 次 GS 算法	$8K^2$
迭代 3 次 GS 算法	$12K^2$
迭代 4 次 GS 算法	$16K^2$
迭代 2 次 BC-GS 算法	$10K^2$
迭代 3 次 BC-GS 算法	$14K^2$
迭代 4 次 BC-GS 算法	$18K^2$

引入块分星座图方法求解初始值的 BC-GS-RTS 算法相比 GS-RTS 算法在初始解向量方面复杂度有所提升，但是仍然从原来 RTS 算法的 $O(K^3)$ 下降一个级别到 $O(K^2)$ 。比如上表 4.5 中，经过两次迭代 BC-GS 算法复杂度为复杂度为 $10K^2$ 。从图 4.5 中，我们可以发现随着小区用户数增加，MMSE、GS 和 BC-GS 算法复杂度都有所增加，而 GS 和 BC-GS 算法随着迭代次数增加复杂度也会相应的增加，但是不论两种算法迭代次数增加到多少 ($2 < i < 4$) 还是小区用户数增加多少，MMSE 算法的复杂度都是最高的。比如在小区用户数为 35 的条件下，MMSE 复杂度为 4.25×10^4 次，迭代次数为 4 的 GS 算法复杂度为 1.4×10^4 次，而迭代 4 次的 BC-GS 算法复杂度为 1.65×10^4 次。具体 MMSE、GS 和 BC-GS 算法复杂度对比如下图 4.5 所示。

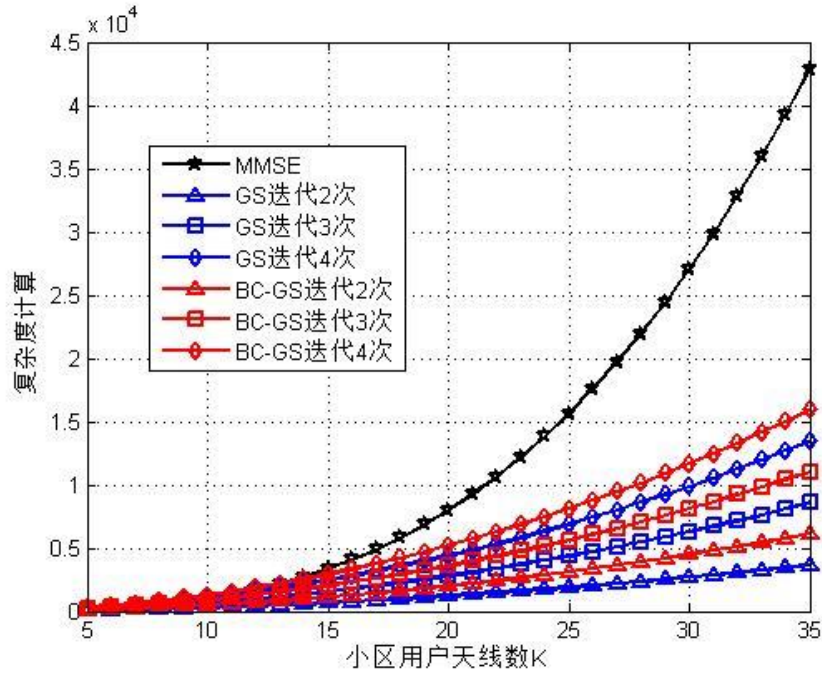


图 4.5 三种初始解算法复杂度对比

4.3.3 BC-GS-RTS 收敛性分析

接着分析 BC-GS-RTS 算法的收敛性，我们通过不同迭代次数下算法性能的变化来验证这一性质，这里我们选取单输入单输出系统的高斯白噪声信道性能作为参考标准。从图 4.6 中，我们可以看出随着迭代次数（这里迭代次数为搜索过程迭代次数，而非初始解向量迭代次数）的增加，不论天线数为 32×16 ， 64×16 还是 128×16 性能都呈现出收敛性，特别是迭代次数超过 60 以后，这一趋势更加明显。比如，当天线数为 128×16 ，迭代次数为 60 时，改进算法 BER 为 8×10^{-5} ，迭代次数增加到 100 时，BER 为 6×10^{-5} ，只比迭代 60 时低 2×10^{-5} 。所以随着迭代次数的递增，改进算法性能表现出稳定和优良的趋势。

结合上一章图 3.10 和图 4.6，我们还可以发现凭借着引入块分星座图方法，BC-GS-RTS 算法收敛速率加快，由原来的 70 次出现收敛趋势减少到 60 次左右，并且算法性能也得到相应提升。比如收发天线数同为 128×16 ，迭代次数同为 60 次，BC-GS-RTS 算法的 BER 为 1.1×10^{-4} ，而 GS-RTS 算法的 BER 为 8×10^{-5} 。

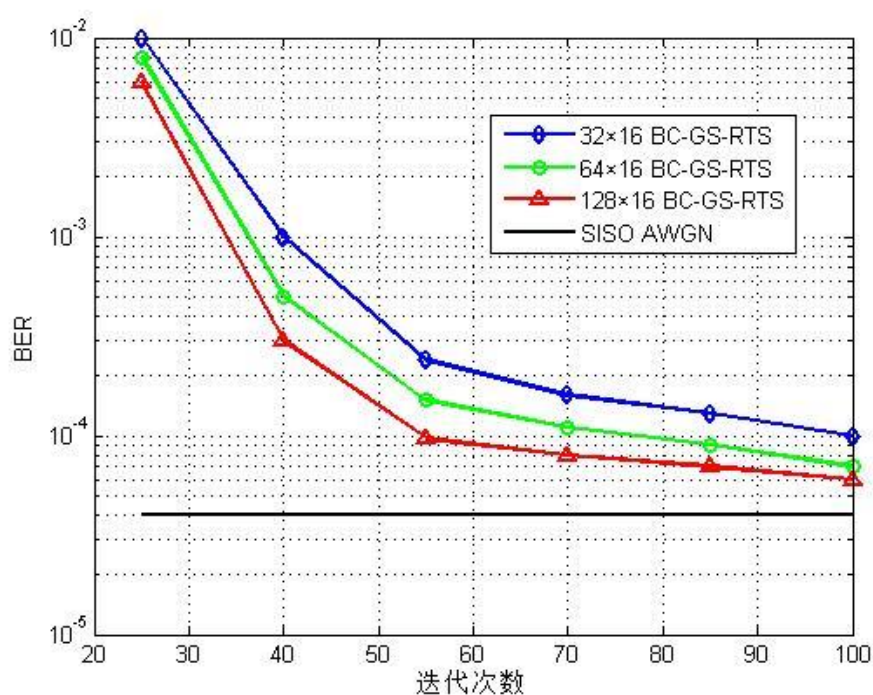


图 4.6 不同迭代次数下 BC-GS-RTS 算法 BER 对比

4.4 本章小结

通过上一章仿真结果发现 GS-RTS 算法虽然复杂度比原有 RTS 算法大幅降低，但是出现随着收发天线数增加，高阶调制下 BER 性能和 RTS 算法差距越来越大的问题。因此本章针对大规模 MIMO 系统高阶调制方式下的情况给出一种改进 GS-RTS 检测方法，即 BC-GS-RTS 信号检测算法，该检测算法在复杂度仅有较少损失的前提下，运用块分星座图方法促使 GS 算法加快收敛速率，提高检测性能，从而使 GS-RTS 算法更好地应用在高阶的调制方式中。

第5章 总结与展望

5.1 本文工作总结

大规模 MIMO 技术作为第五代移动通信系统关键技术之一，凭借着频谱利用率高、系统吞吐量大等特点成为业内的研究热点。但是由于发送端和接收端天线的大幅增加，原来适用于 MIMO 系统的信号检测算法因为复杂度过高无法实际应用在大规模 MIMO 系统中，性能好、低复杂度的大规模 MIMO 系统信号检测算法成为大规模 MIMO 技术实际工程应用的急需解决的问题之一。本文主要研究的是大规模 MIMO 系统上行链路，目前大规模 MIMO 系统上行链路的信号检测算法存在复杂度较高、高阶调制方式下性能差等问题。本文主要针对适用于大规模 MIMO 系统上行链路的 RTS 信号检测算法进行研究，并在此基础上给出改进算法。以下是本文的工作总结：

1. 研究了大规模 MIMO 系统上行链路系统模型、信道出现的新特性“信道硬化”、MIMO 系统中的常规检测算法以及适用于大规模 MIMO 上行链路的基于本地邻域搜索检测算法，并对各种检测算法的优缺点进行了比较分析，通过对比分析得出结论：目前在大规模 MIMO 系统上行链路的信号检测技术中，基于本地邻域搜索的检测算法性能明显优于常规检测算法；同样基于本地邻域搜索算法，RTS 算法凭借着“逃脱”策略，避免 LAS 算法陷入局部最优解陷阱，求得全局最优解向量，从而性能优于 LAS 算法；但是随着收发天线数量增加，RTS 算法初始解向量算法存在复杂度过高的问题，需要在原 RTS 检测算法的基础上进行改进，降低其整体复杂度。

2. 现有的 RTS 检测算法复杂度主要由初始解向量算法和迭代搜索过程两部分组成，而随着天线数量增加，初始解向量算法复杂度所占 RTS 算法整体复杂度比例越来越大。基于 RTS 信号检测算法存在初始解向量算法复杂度过高的问题，针对大规模 MIMO 系统上行链路，本文给出一种基于高斯赛德尔迭代法的改进 RTS 检测算法，即通过 GS 迭代法替换原来的 MMSE 算法求解初始解向量。GS 迭代算法在低阶调制方式下，经过一定次数的迭代过程后，可以达到近似于 MMSE 算法的 BER 性能，但是和 MMSE 算法相比复杂度大幅降低，从而降低 RTS 算法整体

复杂度。并且通过仿真结果验证,低阶调制方式下改进的算法和原有的 RTS 算法相比,在较少的性能损失前提下,算法整体复杂度降低明显。因此,在大规模 MIMO 系统上行链路用户信号采用低阶调制方式的情况下,改进的 GS-RTS 检测算法可以作为原有的 RTS 信号检测算法的替代方案。但是随着调制方式阶数的增加,改进的算法和原有 RTS 算法的 BER 性能差距越来越大,还需要对 GS-RTS 算法改进,提高其高阶调制方式下 BER 性能。

3. 针对 GS-RTS 算法用户信号采用高阶调制方式下 BER 性能较差的问题,本文给出一种的改进 GS-RTS 信号检测算法,即 BC-GS-RTS 算法。本算法引入块分星座图方法求解 GS 迭代法初始解向量,从而加快 GS 迭代法收敛速率,提升 GS 算法性能,进一步提高 GS-RTS 算法整体性能。通过仿真分析验证,改进的 BC-GS-RTS 在复杂度较小的牺牲的前提下(相比于 GS-RTS 算法复杂度略有增加,但是还是比原有的 RTS 算法复杂度低一个量级),收敛速率提升,高阶调制方式下 BER 性能明显提高,可以得到近似于原有 RTS 算法性能。因此,在大规模 MIMO 系统上行链路用户信号采用高阶调制方式的情况下,改进的 BC-GS-RTS 检测算法可以很好地替代原有的 RTS 信号检测算法。

5.2 下一步研究工作展望

为了将大规模 MIMO 技术更好地应用在实际工程中,低复杂度信号检测算法成为业内研究热点。由于个人的能力有限以及时间的仓促,对大规模 MIMO 系统上行链路信号检测算法的研究还不是特别深入,针对 RTS 检测算法的改进工作还有待完善,很多地方需要在以后做进一步的研究分析,主要包括以下几个方面:

1. 本文的 GS-RTS 算法只是降低 RTS 算法的初始解向量算法复杂度,对于 RTS 算法的搜索迭代过程复杂度没有改进,因此对于如何降低 RTS 算法在搜索迭代过程的复杂度问题需要进行下一步的研究,比如考虑联合置信度(BP)算法,从而提高 RTS 算法检测性能,减少迭代次数,加快收敛速率。

2. 本文针对用户信号采用低阶调制的情况,GS-RTS 算法降低 RTS 算法复杂度;针对用户信号采用高阶调制的情况,基于块分星座方法的 BC-GS-RTS 算法。两种基于 RTS 算法的改进检测算法各具优势,应用的场景也不同。如果两种改进算法应用在同一场景中,即相同的调制方式,GS-RTS 算法复杂度比 BC-GS-RTS

算法低，而 BC-GS-RTS 算法性能却优于 GS-RTS 算法，两种算法复杂度由小区用户数决定的，因此考虑在同一调制方式中随着小区用户数变化决定使用哪一种改进算法，从而达到对 RTS 算法的改进最大化，这里两种算法切换的门限值，即小区用户个数确定也值得进一步研究。

3. 本文考虑基于 GS 迭代法作为 RTS 算法初始解向量算法，近些年逐次超松弛迭代(Successive Over Relaxation, SOR)算法凭借性能优于 GS 迭代法优势，成为业内研究热点。可是 SOR 迭代法的最优松弛因子 w 很难选取，复杂度高于 GS 迭代法，下一步可以研究 SOR 最优松弛因子选取过程，提高选取速率，降低 SOR 迭代算法复杂度，从而选用 SOR 算法替换 GS 迭代法作为 RTS 算法初始解向量算法。

4. 本文第四章节只是在 16QAM 调制方式下对 BC-GS-RTS 算法进行实验仿真，为了更充分的验证算法对于高阶调制性能的优化改进，下一步工作将研究 64QAM 以及更高阶调制方式下的算法性能。

5. 本文算法研究还是基于 MIMO 的 V-BLAST 系统模型，为了更好地验证算法性能，搭建更加贴近实际的信道模型也值得研究，比如考虑基站端大规模天线阵列的排列方式对信道的影响。

参考文献

- [1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020[R]. 2014.
- [2] IMT-2020(5G)推进组. 5G 概念白皮书[R]. 2014.
- [3] 王胡成, 徐晖, 程志密, 等. 5G 网络技术研究现状和发展趋势[J]. 电信科学, 2015, 31(9): 156-162.
- [4] NGMN Alliance. NGMN 5G White Paper[R]. 2015
- [5] 唐超. 非正交多址接入下行链路信号检测技术研究[D]. 重庆: 重庆邮电大学, 2016.
- [6] Hoydis J, Brink S T, Debbah M. Massive MIMO in the UL/DL of Cellular Networks: How Many Antennas Do We Need?[J]. IEEE Journal on Selected Areas in Communications, 2013, 31(2):160-171.
- [7] Marzetta T L. Noncooperative cellular wireless with unlimited numbers of base station antennas[J]. IEEE Transactions on Wireless Communications, 2010, 9(11):3590-3600.
- [8] Ngo H Q, Larsson E G, Marzetta T L. Energy and Spectral Efficiency of Very Large Multiuser MIMO Systems[J]. Communications IEEE Transactions on, 2011, 61(4):1436-1449.
- [9] Wu M, Yin Bei, Wang Guohui, et al. Large-Scale MIMO Detection for 3GPP LTE: Algorithms and FPGA Implementations[J]. IEEE Journal of Selected Topics in Signal Processing, 2014, 8(5):916-929.
- [10] 张中山, 王兴, 等. 大规模MIMO关键技术研究及应用[J]. 中国科学:信息科学, 2015, 45(9):1095-1110.
- [11] Li Peng, Murch R D. Multiple output selection-LAS algorithm in large MIMO systems[J]. IEEE Communications Letters, 2010, 14(5):399-401.
- [12] Datta T, Srinidhi N, Chockalingam A, et al. Random-Restart Reactive Tabu Search Algorithm for Detection in Large-MIMO Systems[J]. IEEE Communications Letters, 2011, 14(12):1107-1109.
- [13] Rusek F, Persson D, Lau B K, et al. Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays[J]. Signal Processing Magazine IEEE, 2013,

- 30(1):40-60.
- [14] Gao Xinyu, Dai Linglong, Ma Yongkui, et al. Low-complexity near-optimal signal detection for uplink large-scale MIMO systems[J]. Electronics Letters, 2015, 50(18):1326-1328.
- [15] Larsson E G, Edfors O, Tufvesson F, et al. Massive MIMO for next generation wireless systems[J]. IEEE Communications Magazine, 2014, 52(2):186-195.
- [16] Lu Lu, Li Ye, Swindlehurst A L, et al. An Overview of Massive MIMO: Benefits and Challenges[J]. IEEE Journal of Selected Topics in Signal Processing, 2014, 8(5):742-758.
- [17] Qian Manli, Wang Yuanyuan, Zhou Yiqing, et al. A super base station based centralized network architecture for 5G mobile communication systems[J]. Digital Communications & Networks, 2015, 1(2):152-159.
- [18] Li Caifang, Lu Xu, Huang D D. Low Complexity Iterative MMSE-PIC Detection for Medium-Size Massive MIMO[J]. IEEE Wireless Communication Letters, 2016, 5(1):108-111.
- [19] Jose J, Ashikhmin A, Marzetta T L, et al. Pilot Contamination and Precoding in Multi-Cell TDD Systems[J]. IEEE Transactions on Wireless Communications, 2011, 10(8):2640-2651.
- [20] Srinidhi N, Datta T, Chockalingam A, et al. Layered Tabu Search Algorithm for Large-MIMO Detection and a Lower Bound on ML Performance[J]. IEEE Transactions on Communications, 2010, 59(11):2955-2963.
- [21] Forney G D. Codes on Graphs: Fundamentals[J]. IEEE Transactions on Information Theory, 2013, 60(10):5809-5826.
- [22] 秦闯, 郑紫微, 娄达平. 基于近似信息传递算法的大规模 MIMO 信号检测[J]. 电信科学, 2016, 32(9):16-20
- [23] Alshamary H A J, Xu W. Efficient Optimal Joint Channel Estimation and Data Detection for Massive MIMO Systems[J]. IEEE International Symposium on Information Theory, 2016, 31(16): 875-879.
- [24] Kong Dejin, Xia Xianggen, Jiang Tao. A Differential QAM Detection in Uplink Massive MIMO Systems[J]. IEEE Transactions on Wireless Communications, 2016, 15(9):6371-6383.
- [25] Wu Zhizhen, Zhang Chuan, Xue Ye, et al. Efficient architecture for soft-output massive MIMO detection with Gauss-Seidel method[C]// IEEE International

- Symposium on Circuits and Systems. QC, Canada, IEEE, 2016:1886-1889.
- [26] Mohammed S K, Larsson E G. Per-Antenna Constant Envelope Precoding for Large Multi-User MIMO Systems[J]. Communications IEEE Transactions on, 2012, 61(3):1059-1071.
- [27] 周将运. Massive MIMO 系统的检测算法研究[D]. 成都: 电子科技大学, 2015.
- [28] Tulino M A, Verdu S. Random matrix theory and wireless communication[J]. Communications and Information theory, 2014, 1(1):1-182.
- [29] Studer, Christoph, Icskei, Helmut. Soft-Input Soft-Output Single Tree-Search Sphere Decoding[J]. IEEE Transactions on Information Theory, 2010, 56(10):4827-4842.
- [30] 尤明厚, 陶小峰, 崔琪楣, 等. Turbo-MIMO 系统中一种基于部分后验概率的软检测算法[J]. 电子与信息学报, 2010, 32(7): 1531-1527.
- [31] Yin Haifan, Gesbert D, Filippou M, et al. A Coordinated Approach to Channel Estimation in Large-Scale Multiple-Antenna Systems[J]. IEEE Journal on Selected Areas in Communications, 2012, 31(2):264-273.
- [32] Song Jian, Huang Su, Wang Jun, et al. The Noise Transfer Analysis in Frequency Domain Zero-Forcing Equalization[J]. IEEE Transactions on Communications, 2013, 61(6):1-12.
- [33] Poor H V, Verdu S. Probability of error in MMSE multiuser detection[J]. IEEE Transactions on Information Theory, 1997, 43(3):858-871.
- [34] Kolen A, Pesch E. Genetic local search in combinatorial optimization[M]// Local search in combinatorial optimization. Princeton University Press, 2003:587--596.
- [35] Mohammed S K, Zaki A, Chockalingam A, et al. High-Rate Space-Time Coded Large-MIMO Systems: Low-Complexity Detection and Channel Estimation[J]. Selected Topics in Signal Processing IEEE Journal of, 2009, 3(6):958-974.
- [36] Sun Y. Eliminating-highest-error and fastest-metric-descent criteria and iterative algorithms for bit-synchronous CDMA multiuser detection[C]// IEEE International Conference on Communications. IEEE Xplore, 1998:1576-1580.
- [37] Vishnu Vardhan K, Mohammed S K, Chockalingam A, et al. A Low-Complexity Detector for Large MIMO Systems and Multicarrier CDMA Systems[J]. IEEE Journal on Selected Areas in Communications, 2008, 26(3):473-485.

-
- [38] 杨海洋. Massive MIMO 中低复杂度接收算法的研究[D]. 成都: 电子科技大学, 2013.
- [39] 邓恺. 大规模 MIMO 中低复杂度检测算法设计[D]. 南京: 南京邮电大学, 2016.
- [40] Glover F. Future paths for integer programming and links to artificial intelligence[J]. Computers & Operations Research, 1986, 13(5):533-549.
- [41] Srinidhi N, Mohammed S K, Chockalingam A, et al. Neal-ML Signal Detection in Large-Dimension Linear Vector Channels Using Reactive Tabu Search[J]. IEEE Global com, 2009, 19(2):148-156.
- [42] 皮秀敏. Massive MIMO 中基于 BP 的检测算法研究[D]. 成都: 电子科技大学, 2015.
- [43] Chockalingam A, Rajan B S. Large MIMO Systems[M]. Cambridge University Press, 2014.
- [44] 杨大江, 宋常健, 钟子发. 高阶多输入多输出系统下基于禁忌搜索的信号检测[J]. 计算机工程, 2015, 41(12):140-143.
- [45] 商阳. 256QAM 高阶调制下 MIMO 信号检测算法研究[D]. 重庆: 重庆邮电大学, 2016.

致谢

时光荏苒，岁月如梭，三年的研究生生活即将步入尾声。回首这三年，不论是欢声和笑语，还是辛劳和汗水，都将成为我这辈子最宝贵的回忆。在这毕业论文即将完成之际，谨向所有关心过我的和给过我无私帮助的老师、同学、朋友以及家人致以最诚挚的感谢。

首先，我要特别感谢我的导师王茜竹老师，本论文是在王老师悉心指导下完成的。在这三年的求学生涯中，王老师扎实广泛的专业知识，严谨认真的治学态度，追求完美的工作作风，诲人不倦的为师品德，平易近人的人格魅力对我产生了深远的影响。本论文从开题到最终完成，每一步王老师都给予我大力的帮助及宝贵的意见，特别是在论文研究方向的确立以及最终毕业论文完成的过程中，王老师悉心指导，倾注了大量的时间和心血。此外，王老师还言传身教地教会我很多做人的道理，这些都使我终生受益。

然后，我要感谢 LTE-V 无线传输技术标准化及样机研发验证项目组各位成员：邱聪聪、高亚龙、徐林、王付龙等。LTE-V 项目组给我提供了宝贵的学习与实践平台，在项目中遇到问题大家一起想办法解决，出现 bug 大家一起调试的经历我永远不会忘记。项目进展过程中，在项目组的各位成员的帮助下，我的专业技能不断提升，这给我的毕业求职过程也带来巨大的帮助。

同时我要感谢实验室团队的刘艳林、赖香武、杨彩敏、王文静、周小宇、罗林等同学，感谢这三年来无论是学习上还是生活上大家给予我的关怀和帮助，正是大家在实验室营造互相督促、互相鼓励、互相关怀的良好学习氛围，才使我按时顺利地完成毕业论文。

我还要感谢我的父母和朋友们，正是他们在背后默默地支持我，为我的生活和学习提供了强大的精神后盾，在自我怀疑和犹豫的关键时刻给予我不断前进的力量，我才能顺利地完成三年的研究生学业。

最后，我要感谢辛苦审阅我的论文以及参与我毕业答辩的各位专家和教授，感谢你们在百忙之中抽出时间来参加我论文的评审和答辩，并提出宝贵的意见。

攻读硕士学位期间从事的科研工作及取得的成果

参与科研项目：

- [1] TD-LTE-Advanced 终端基带芯片工程样片研发（2012ZX03001012），国家重大科技专项项目，2015.03-2016.01.
- [2] LTE-V 无线传输技术标准化及样机研发验证（2016ZX03002019-007），国家重大科技专项项目，2016.03-2017.01.

发表及完成论文：

- [1] **李楠**，王茜竹，黄德玲. 低复杂度大规模MIMO信号检测算法研究[J]. 信息通信, 2017, 169(1): 32-34.
- [2] 王茜竹，**李楠**，黄德玲. 大规模MIMO系统的改进RTS信号检测算法[J]. 计算机工程与应用. （北大中文核心，CSCD核心，已录用）

word版下载: <http://www.ixueshu.com>

免费论文查重: <http://www.paperyy.com>

3亿免费文献下载: <http://www.ixueshu.com>

超值论文自动降重: http://www.paperyy.com/reduce_repetition

PPT免费模版下载: <http://ppt.ixueshu.com>

阅读此文的还阅读了:

1. [基于线性迫零V-BLAST算法的MIMO MC-CDMA系统上行链路性能分析](#)
2. [LTE上行MIMO检测算法研究](#)
3. [大规模MIMO上行链路的信号检测方法](#)
4. [一种用于大规模MIMO信号检测的改进算法](#)
5. [应答器上行链路信号解调及译码的实现](#)
6. [基于临近信息传递算法的大规模MIMO信号检测](#)
7. [用于MIMO空分复用系统上行链路接收机的球形检测算法](#)
8. [采用迫零接收机的大规模MIMO上行链路支持用户数研究](#)
9. [大规模MIMO系统的改进RTS信号检测算法](#)
10. [基于V-BLAST的MIMO信号检测算法](#)
11. [大规模MIMO信号检测算法研究](#)
12. [大规模MIMO系统上行链路性能分析](#)
13. [大规模MIMO信号检测算法研究](#)
14. [采用迫零接收机的大规模MIMO上行链路支持用户数研究](#)
15. [基于Gibbs采样的MIMO信号检测改进算法](#)
16. [MIMO雷达信号检测前跟踪算法及性能](#)
17. [采用迫零接收机的大规模MIMO上行链路支持用户数研究](#)
18. [大规模MIMO系统中低复杂度的上行信号检测算法的综述](#)
19. [LTE FDD上行MIMO检测算法的应用](#)
20. [MIMO系统的信号检测算法分析](#)
21. [CDMA系统的上行链路仿真研究](#)
22. [一种上行链路大规模MIMO系统的低复杂度检测算法](#)
23. [大规模MIMO系统中低复杂度的上行信号检测算法的研究](#)
24. [基于解调参考信号的LTE上行信号检测算法](#)
25. [EPON上行链路动态带宽分配算法\(DBA\)分析](#)

- [26. 大规模MIMO上行链路导频污染问题的研究](#)
- [27. 基于LTE下行链路的MIMO优化算法研究](#)
- [28. LTE-A上行链路中的串行干扰消除检测算法研究](#)
- [29. LTE FDD上行MIMO检测算法研究](#)
- [30. 大规模MIMO信号检测算法研究](#)
- [31. LTE上行链路虚拟MIMO系统的用户配对算法研究](#)
- [32. 大规模MIMO系统中的导频调度算法研究](#)
- [33. 大规模MIMO系统中上行链路导频设计](#)
- [34. 一种用于大规模MIMO的改进信号检测算法](#)
- [35. 一种用于大规模MIMO信号检测的改进算法](#)
- [36. 支持QoS的LTE MIMO下行链路资源调度算法研究](#)
- [37. 低复杂度大规模MIMO信号检测算法研究](#)
- [38. 基于线性迫零V-BLAST算法的MIMO MC-CDMA系统上行链路性能分析](#)
- [39. 应答器上行链路信号MTIE测试方法](#)
- [40. 双上行链路MIMO技术让移动WiMAX网络如虎添翼\(上\)](#)
- [41. 应答器上行链路信号实验系统研究](#)
- [42. LTE上行链路虚拟MIMO关键技术研究](#)
- [43. LTE-A上行链路均衡技术研究](#)
- [44. LTE上行链路MIMO技术研究](#)
- [45. 大规模MIMO系统中上行链路导频设计](#)
- [46. 基于混沌理论的应答器上行链路信号检测方法研究](#)
- [47. 双上行链路MIMO技术让移动WiMAX网络如虎添翼\(下\)](#)
- [48. 大规模MIMO上行链路信号检测算法研究](#)
- [49. 一种用于大规模MIMO信号检测的改进算法](#)
- [50. MIMO-OFDM系统传统信号检测算法](#)