

分类号 _____
学校代码 10487

学号 M201271797
密级 _____

华中科技大学

硕士学位论文

基于量子群智能优化算法的
频谱感知技术研究

学位申请人 李 凌 云

学 科 专 业 : 通信与信息系统

指 导 教 师 : 魏蛟龙 教授

答 辩 日 期 : 2015.05.28

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Engineering**

**Spectrum Sensing Technology Research Based on
Quantum Swarm Intelligence Optimization Algorithm**

Candidate : Li Lingyun

Major : Communication and Information System

Supervisor : Prof. Wei Jiaolong

Huazhong University of Science & Technology

Wuhan 430074, P.R.China

May, 2015

独创性声明

本人声明所提交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文授权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本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和汇编本学位论文。

本论文属于 保密，在_____年解密后适用本授权书。

不保密。

(请在以上方框内打“√”)

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

摘要

无线电频谱是全人类的公共资源，其具有有限性和不可再生的特点。通信技术的迅猛发展导致频谱资源日益短缺，如何克服这种短缺导致的无线通信发展瓶颈，已成为该领域研究的焦点。认知无线电被誉为是能够提高无线频谱资源利用率的一项新技术，它是一个能够连续不断感知外部环境，并且通过学习适应外部无线环境的智能通信系统。

频谱感知是认知无线电中的重要组成部分，能够准确检测出授权用户的信号，避免影响授权用户通信效果，以实现提高频谱利用率。本课题研究过程中采用了一种线性协作频谱感知模型。

本文提出了基于量子计算原理的量子群智能算法，以提高传统群智能算法性能。仿真试验结果表明：量子群智能算法比传统算法具有更高的收敛精度和更快的收敛速度。同时本文将量子群智能算法应用于频谱感知这一工程问题。

本文基于混合蛙跳算法和量子理论，对混合蛙跳算法进行改进，提出了一种全新的量子蛙跳算法，并将其应用于认知协作频谱感知。同时，本文基于传统细菌觅食算法和量子理论，改进细菌觅食算法，提出了一种全新的量子细菌觅食算法，并提出基于该算法的认知协作频谱感知方法。

关键词：认知无线电；频谱感知；量子群智能；量子蛙跳；量子细菌觅食

ABSTRACT

Radio spectrum may be one of the most tightly regulated resources of all time, which has the characteristic of scarceness and can not be reproduced. The scarceness of wireless spectrum hampers the rapid development of wireless communication service. How to use wireless spectrum is the hot issue of wireless communication field. Cognitive radio is viewed as a novel approach for improving the utilization of a precious natural resource: the radio spectrum, it is defined as an intelligent wireless communication system that is aware of its environment and uses the methodology of understanding-by-building to learn from the environment.

Spectrum sensing, as a key technology in cognitive radio, needs to reliably detect signals from licensed primary radios to avoid harmful interference. In our paper, we adopt an optimal linear cooperation framework for spectrum sensing model.

In order to improve the performance of swarm intelligence algorithm, two novel algorithms based on the theory of quantum computing are proposed. The simulation results show that the newly proposed algorithms have a better searching speed and an obvious accuracy. In addition, we applied our newly designed algorithm in spectrum sensing.

Based on shuffled frog leaping algorithm and quantum theory, quantum frog leaping algorithm is proposed. Then, we applied our novel algorithm in the spectrum sensing issue and proposed the optimal linear cooperation model for spectrum sensing based on quantum frog leaping algorithm. Based on bacterial foraging algorithm and quantum theory, quantum bacterial foraging algorithm is proposed. Then, we applied our novel algorithm in the spectrum sensing issue and proposed the optimal linear cooperation model for spectrum sensing based on quantum bacterial foraging algorithm.

Key words: Cognitive radio; Spectrum sensing; Quantum swarm intelligence; Quantum frog leaping algorithm; Quantum bacterial foraging algorithm

目 录

摘 要.....	I
ABSTRACT.....	II
第 1 章 绪论.....	1
1.1 课题研究背景	1
1.2 认知无线电发展现状	2
1.3 本文主要研究工作	3
第 2 章 认知协作频谱感知模型	5
2.1 单节点感知方法	5
2.2 协作频谱感知方法	6
2.3 基于修正偏差因子的频谱感知方法	8
2.4 基于粒子群算法的频谱感知方法	9
2.5 仿真试验及结果分析	10
2.6 本章小结.....	11
第 3 章 基于量子蛙跳算法的认知无线电频谱感知	13
3.1 量子进化概述	13
3.2 混合蛙跳算法介绍	15
3.3 基于量子蛙跳算法的认知无线电频谱感知	16
3.4 本章小结.....	28
第 4 章 基于量子细菌觅食算法的认知无线电频谱感知	29
4.1 细菌觅食算法介绍	29
4.2 基于量子细菌觅食算法的认知无线电频谱感知	31
4.3 本章小结.....	43
第 5 章 论文总结.....	45
致 谢.....	47
参考文献.....	48

第1章 绪论

1.1 课题研究背景

无线通信业务的日益增长与无线频谱资源的有限性和不可再生的矛盾越来越突出，因此，如何提高频谱资源的利用率以实现与用户需求相适应十分关键。纵观无线通信领域的发展，不难看出，每一次无线通信领域取得重大突破的同时都离不开对频谱资源的有效利用问题的进步和发展。而目前无线通信的发展突飞猛进，现有无线频谱资源难以满足用户需求，如何克服这种短缺所带来的限制已成为得到该领域研究学者的广泛关注^[1]。

一份来自美国联邦通信委员会的研究表明，当前业界使用的频谱分配策略对于频谱资源的使用是非常不平衡的^[2]。通常会出现下列场景：有时候，对于某些应用在非授权频段的业务会非常繁忙；而某些时候，某些授权频段的业务非常空闲，这常常表现在信号传播特性比较好的低频频段，这会导致频谱利用率降低，频谱资源存在着巨大浪费，很多频段都未能得到充分利用^[3-5]。由于传统频谱分配是静态的，无线业务所对应的频谱范围固定分配，而当前的现状是大多数国家和地区可用的频谱资源几乎都已经分配给现有的业务，而随着技术不断发展，新涌现的系统 and 新兴业务所能利用的频谱资源非常有限，难以满足业务的开展^[6]。认知无线电(Cognitive Radio)技术就是针对这些困难产生的技术，它主要针对如何缓解不断增长的无线业务需求和当前紧缺的频谱资源之间的矛盾，如何在现有的无线资源条件有限下充分利用频谱。认知无线电能够有效地缓解缺少频谱资源的现状，因此该技术有着广阔的发展空间^[7]。

认知无线电^[8-9]最早由瑞典皇家技术学院的 Joseph Mitola 博士 Joseph Mitola 博士 1999 年提出。认知无线电，顾名思义就是可以对周围的无线电通信环境进行感知，并可以对感知到的环境信息进行理解和分析，运用学习、理解等方式，自适应的调整内部的通信机理、实时改变特定的操作参数，以实现无线环境的适应，进而实现对空闲频谱的主动寻找和充分使用^[10]。认知无线电技术的关键是利用频谱感知(Spectrum Sensing)系统，最终可以做到动态频谱分配(DSA: dynamic spectrum allocation)和频谱共享(Spectrum Sharing)^[11]。

不难发现,要想实现认知无线电通信,一个关键技术就是频谱感知^[12-14],频谱感知可以准确获知某一时刻、某一地点是否有空闲频段可以供认知用户利用;与此同时,频谱感知技术可以随时检测是否存在准备接入认知频段的新授权用户,当检测到授权用户时,认知用户应及时退出该频段以避免对授权用户的干扰。认知用户具备认知能力,通过对周围频谱环境的智能感知,感知是否存在已经分配给授权用户但没有得到充分利用的“频谱空洞”,通过这种自动搜寻和充分利用实现与当前授权用户共享频谱。作为认知能力的重要前提,频谱感知性能的优劣对整个认知无线电系统的质量至关重要。

1.2 认知无线电发展现状

目前,国内外的多家高校院所、研究机构已经开展了对认知无线电的研究,并且在标准化工作的推动中,已经取得了一定的研究成果。

其中 IEEE 组织积极推动开展认知无线电的国际标准化工作,并主要制定出下述标准: IEEE 802.22、IEEE 802.16h 和 IEEE 802.11h 等^[15]。

北美和欧洲率先研究认知无线电的理论技术,其中,美国开展的下一代通信计划 XG 项目,着力于对认知无线电动态频谱管理标准的研究和开发。在这一项目中采用了的完全自由的随机频谱接入的对等结构网络通信。欧盟第六框架目的端到端配置(2R)项目,该项目针对多种无线接入系统共存的环境下的认知无线电技术。在随后的 2008 年,欧盟组织启动了名为 E3(End to End Efficiency)的第七框架重点大项目,该项目主要涵盖认知无线电及网络的技术、商业、发展规划和国际标准化等方面,旨在实现异构网中的应用和业务的无缝接入。

另外,国外的某些高校和科研机构也加入了对认知无线电技术的研究,针对体系结构、理论模型和实现平台等方面重点关注,并主要提出了基于 OFDM 的动态频谱接入频谱池 Spectrum Pooling 系统^[16]、CORVUS(Cognitive Radio approach for usage of Virtual Unlicensed Spectrum)结构^[17]、基于 OFDM 的动态频谱网络架构 OCRA、UWAN 系统^[18]、由 UCSB 的研究组合作的 Nautilus 项目^[19]、基于遗传算法的生物启发认知模型 Bio-CR、泛在网动态和智能化频谱管理网以及基于干扰温度的多址动态频谱分配等。

相比于国外研究,国内的对于认知无线电及相关技术的研究开展相对较晚。但

最近几年，国家高度重视并大力支持认知无线电的研究和发展。国内最早的研究起步于 2005 年 7 月的国家 863 计划，该计划首次成立“认知无线电关键技术”课题，随后的 2006 年、2007 年，在自然科学基金的资助扶持下，多个认知无线电项目相关课题相继成立，2008 年则是认知无线电发展关键的一年，随着信息产业部将认知无线电项目作为的重点项目予以资助，重点关注并着力研究频谱感知、频谱资源动态管理以及抗干扰等一系列问题。随后，国家 973 计划重点资助并启动“认知无线网络基础理论与关键技术研究”项目，同时把认知无线网络的体系结构、认知无线电协议、动态网络资源的智能化管理和控制机制、无线网络的认知基础理论、认知网络传输机制、端到端重构机制等作为重点领域重点研究。与此同时，在国家重大专项中也强调“频谱资源共享、感知并灵活使用的相关技术研究及验证”，而“基于频谱资源共享的无线通信系统”作为国家 863 计划重点项目，重点关注频谱资源共享，并验证了 698-806MHz 频段^[15]，可以看出认知无线电技术逐渐从理论走向实用。

综上所述，目前国内外的研究机构对于认知无线电技术的研究工作正在从基本理论以及针对需求分析的研究，逐步迈向对于认知网络的具体实现乃至商业化标准的关键技术研究。上述研究涉及物理层到无线资源管理层甚至到网络体系架构层乃至无线通信领域，并将逐步从理论走向应用。

本文重点研究认知无线电中的频谱感知问题。传统的单点频谱感知技术容易受到外界恶劣环境的影响，使得检测结果不可靠。本文基于认知无线电协作频谱感知模型^[20-21]，并且将智能算法用于该模型以提高认知用户的检测概率。

智能计算迅猛发展，智能算法也逐步在各个工程领域得到应用。传统的以及一些新近提出的智能算法诸如粒子群算法、混合蛙跳算法以及细菌觅食算法，在解决协作频谱感知问题中都存在着全局收敛性差，容易陷入局部收敛的缺点。因此，急需在原有基础上对这些经典算法进行改进，以提高它们在工程问题中的性能。本文就是立足于以上观点，将量子计算、文化算法与原有智能计算方法有机结合，获得更好的全局收敛性能，以拓宽智能计算在工程背景下尤其是认知频谱感知问题中的应用范围。

1.3 本文主要研究工作

本文的主要研究工作包括：

1、在充分研究认知无线电的理论概念和当前现状的基础上，本文分析并研究了目前广泛使用的认知无线电频谱感知方法，并重点研究了认知无线电协作频谱感知模型。研究基于通用修正偏差因子(modified deflection coefficient, MDC)的频谱感知方法。然而需要关注的是，对于通用修正偏差因子的求解方法是一种次优方法，它的收敛精度低，难以保证理论最优。

2、依据混合蛙跳算法、细菌觅食算法以及粒子群算法等一些典型智能算法的基本原理，研究智能算法在认知无线电频谱感知中的应用并进行仿真。仿真试验结果表明：群智能计算收敛精度可以显著提高，而且群智能算法是一种全局最优的求解算法。

3、基于量子计算的方法，设计出全新的量子蛙跳算法和量子细菌觅食算法。将量子计算与智能算法有机结合，并通过对通用基准函数的测试验证新设计算法的优越性，并把它用在认知协作频谱感知模型中。仿真结果表明，基于量子蛙跳算法和量子细菌觅食算法的协作感知方法，相对于粒子群算法等经典演化计算方法具有更优越的性能，工程应用领域发展潜力巨大。

第2章 认知协作频谱感知模型

自从认知无线电的概念提出以来就受到广泛关注，而频谱感知作为认知无线电的关键技术，也成为领域内研究热点。目前广泛使用的频谱感知方法根据检测节点的不同，可以分为单节点感知及协作频谱感知两大类，如下图 2.1 所示：

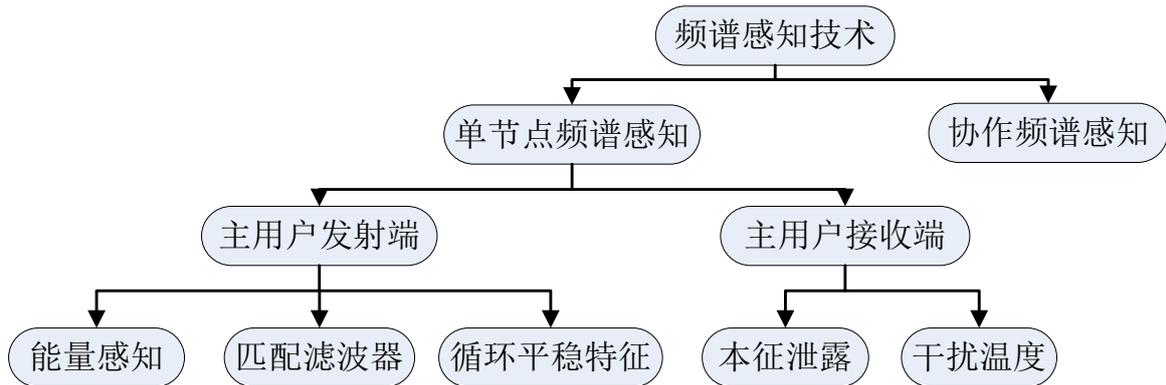


图 2.1 频谱感知技术分类

2.1 单节点感知方法

频谱感知的关键目标是判断感知频段上是否存在授权用户的信号，由此可知，频谱感知的实时性和准确性对于系统性能至关重要。频谱感知准确性主要通过检测概率和虚警概率两个参数来表征。检测概率是指当感知频段上存在授权用户时，认知用户能够正确检测出授权用户的概率；而虚警概率表示频段上不存在授权用户，认知用户错误判断为授权用户的概率。频谱感知系统应该降低虚警概率并增大检测概率，以实现正确感知并提高频谱利用率和感知系统的吞吐量。单节点感知方法很容易受到信道恶劣环境的干扰而影响系统检测的准确性，但是由于单节点感知方法是构造合作感知方法的核心和基础，本节简要描述单节点感知方法。

2.1.1 匹配滤波器检测

当高斯白噪声信道时，最佳检测器是匹配滤波器，它可以使瞬时输出信噪比（signal to noise ratio, SNR）达到最大^[25]。在匹配滤波器检测中，接收端需要利用提前知晓的包括调制、数据包等授权用户的信号信息。

2.1.2 能量检测

能量感知方法算法简单且计算量小，是目前研究的最充分的频谱感知算法，并逐渐为一些国际标准采用^[26]。能量检测算法的核心思想是积累接收信号，该操作可在时域也可在频域内进行，并将得到的信号能量与提前设定的阈值比较，进而得到感知结果。

由于噪声无时无刻不在变化，因此能量检测方法的判决门限的选择对于系统检测性能有较大影响。针对这一问题双门限检测方法^[27]应运而生，主要目的借助双门限消除噪声的不确定性因素。

2.1.3 基于干扰温度的检测

干扰温度模型是测量干扰的新模型，所谓干扰温度是指单位频带内接收天线位置上发射功率的等效温度^[28]。

干扰温度模型的主要思想是：接收机设置了一个表征其能承受的最大干扰值的干扰温度极限。只要认知用户信号的传输能够保证主接收机的干扰温度在上述最大干扰范围以内，认知用户就可以使用授权用户频带。

2.2 协作频谱感知方法

基于单节点感知方法，研究学者提出了协作感知方法，该方法的目的是克服通信过程中阴影和衰落的影响。所谓协作是指利用多个认知用户，协作频谱感知就是对他们的感知结果、检验统计量或检验数据融合并进一步处理，根据融合结果估计判断频谱的使用情况^[29]。

协作频谱感知主要包括认知用户域和融合中心^[30-31]。为了实现各独立认知用户的相互合作，认知用户域是指认知用户的参与数量、通信方式、合作方式以及上述认知用户的感知结果的上报策略。而融合中心则重点关注集中频谱感知算法，融合中心可以根据认知用户域上报的频谱结果，采用合适的融合算法操作，获得频谱利用率。

2.2.1 线性协作模型介绍

前面介绍的单节点检测方法容易受阴影和深度衰落^[32]影响，而使检测结果不可

靠。本文采用了文献[20]中提出的线性协作频谱感知模型以解决上述问题，如此便在认知用户之间构建了一种的分布式合作检测模式^[33-34]。通过融合中心，将不同信道传输的信号融合处理，从而提高认知用户的检测概率^[35]。

2.2.2 频谱感知线性协作模型

参照图2.2可以得到线性协作频谱感知模型。假设协作认知网络中有 M 个认知用户，在 k 时刻各用户的本地感知二元假设检验模型如下所示：

$$\begin{cases} H_0 : x_j(k) = v_j(k) \\ H_1 : x_j(k) = h_j s(k) + v_j(k) \end{cases} \quad (2.1)$$

上述表达式中， $s(k)$ 为主用户信号，认知用户均能获知该信号。 h_j 为传输过程中的信道衰减，假设信道衰减恒定不变。 $x_j(k)$ 为第 j 个认知用户接收到的信号， $v_j(k)$ 为该信号的加性高斯白噪声，白噪声的方差 $\sigma = [\sigma_1^2, \sigma_2^2, \dots, \sigma_M^2]^T$ 。

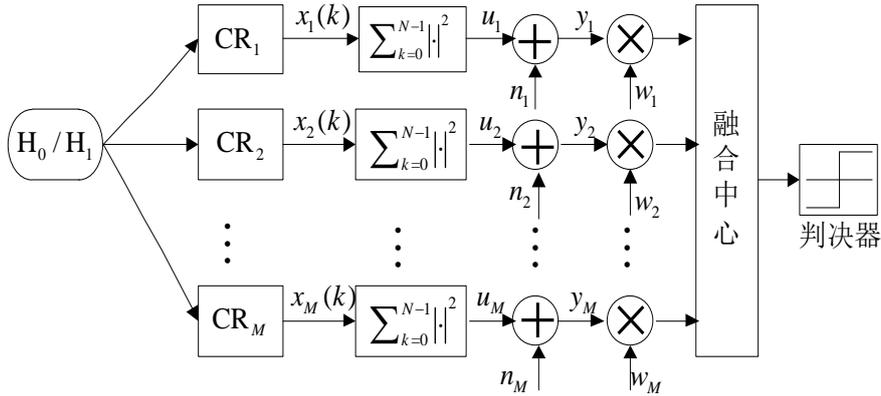


图 2.2 线性协作频谱感知示意图

每个用户的本地感知采用能量检测方法，经过 N 点采样，判决统计量为：

$$u_j = \sum_{k=0}^{N-1} |x_j(k)|^2 \quad (2.2)$$

上述判决统计量经过控制信道最终到达融合中心：

$$y_j = u_j + n_j \quad (2.3)$$

上式中，在控制信道传输过程中，受到噪声 n_j 影响，控制信道噪声为高斯噪声，均方差为 0，方差为 $\delta = [\delta_1^2, \delta_2^2, \dots, \delta_M^2]^T$ 。各认知用户的信号表示为 y_j ，则融合中心的全局判决统计量表示如下：

$$y_c = \sum_{j=1}^M w_j y_j \quad (2.4)$$

上式中, $\mathbf{w}=[w_1, w_2, \dots, w_M]$ 表示控制判决的权重, 该权重值表示融合中心全局检测过程中每个认知用户的贡献情况。

融合中心通过比较 y_c 和门限 γ_c : 如果 $y_c \geq \gamma_c$, 则认为授权用户信号存在; 相反, 则认为没有授权用户信号。

虚警概率可表示为:

$$P_f = Q\left[\frac{\gamma_c - N\boldsymbol{\sigma}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T}}\right] \quad (2.5)$$

检测概率的表达式为

$$P_d = Q\left[\frac{\gamma_c - (N\boldsymbol{\sigma} + E_s \mathbf{h})^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}}\right] \quad (2.6)$$

其中, $Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$, $E_s = \sum_{k=0}^{N-1} |s(k)|^2$, $\mathbf{A} = 2N \text{diag}^2(\boldsymbol{\sigma}) + \text{diag}(\boldsymbol{\delta})$, $\mathbf{B} = 2N \text{diag}^2(\boldsymbol{\sigma}) + \text{diag}(\boldsymbol{\delta}) + 4E_s \text{diag}(\mathbf{h}) \text{diag}(\boldsymbol{\sigma})$, $\mathbf{h} = [|h_1|^2, |h_2|^2, \dots, |h_M|^2]^T$ 。

在虚警概率 P_f 一定时, 使检测概率 P_d 最大化即为认知无线的最终目标。

其中,

$$\gamma_c = N\boldsymbol{\sigma}^T \mathbf{w}^T + Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} \quad (2.7)$$

将(2.7)式代入(2.6)式, 得到

$$P_d = Q\left[\frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}}\right] \quad (2.8)$$

Q 函数单调递增, 因此公式 (2.8) 要取得最大值, 也就相当于下式取最小值, 即:

$$f(\mathbf{w}) = \frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}} \quad (2.9)$$

2.3 基于修正偏差因子的频谱感知方法

修正偏差因子 (modified deflection coefficient, MDC) 方法由文献[20]中提出, 该方法可以解决协作感知问题。由于直接获得(2.9)式的解析解比较困难, 修正偏差因子方法并不是直接优化(2.9)式来解决该优化问题, 而是将上述表达式中的最小化的优化为MDC的最大取值优化, MDC的表达式如下^[20]

$$d_m^2(\mathbf{w}) = \frac{[\mathbf{E}(y_c | H_1) - \mathbf{E}(y_c | H_0)]^2}{\text{Var}(y_c | H_1)} = \frac{(E_s \mathbf{h}^T \mathbf{w}^T)^2}{\mathbf{w} \mathbf{B} \mathbf{w}^T} \quad (2.10)$$

上式中 $\mathbf{E}(\square)$ 为求均值操作, $\text{Var}(\square)$ 为求方差操作。

在上述方法中， $d_m^2(w)$ 表示协作频谱感知中的检测概率。那么，上述频谱感知问题可以转变成了下述函数 $d_m^2(w)$ 的最大化优化：

$$\max_w d_m^2(w) \quad (2.11)$$

由此求得的一个最优权重向量为

$$w^* = B^{-1} B^{-T} h \quad (2.12)$$

2.4 基于粒子群算法的频谱感知方法

1995年，美国的 Kenney 和 Eberhart 两位学者率先提出粒子群算法（PSO），该算法基于群体智能理论，它是一种对鸟群觅食过程的近似，即通过个体的相互合作和竞争所得到的群集智能指导搜索。粒子群算法参数非常简洁，不需要复杂的调整，而且操作简单便于实现，因此，该在函数优化领域，模糊系统控制领域，神经网络训练领域，多用户检测领域以及其它遗传算法领域得到充分利用。

在粒子群算法中，将每一个最优候选解称为一个“粒子”，若干个候选解就构成了鸟的群体。该算法中的粒子没有重量和也没有体积，而是通过目标函数决定其适应度值。这些粒子在各自的解的空间运动，粒子的运动方向和距离由其运动速度确定，这些粒子随时动态调整各自的位置信息，以追随自身的最优位置以及全局的最优位置。

考虑最小化优化问题，假设搜索空间为 N 维，粒子群规模为 S ，则在第 t 次迭代中，第 i 个粒子的当前位置是 $\mathbf{X}_i(t)=[X_{i1}(t), X_{i2}(t), \dots, X_{iM}(t)]$ ，它的当前速度是 $\mathbf{V}_i(t)=[V_{i1}(t), V_{i2}(t), \dots, V_{iM}(t)]$ ，用下式表示其此刻个体的最优位置 $\mathbf{P}_i(t)=[P_{i1}(t), P_{i2}(t), \dots, P_{iM}(t)]$ ，其更新公式为

$$\mathbf{P}_i(t) = \begin{cases} \mathbf{X}_i(t), & \text{if } f[\mathbf{X}_i(t)] < f[\mathbf{P}_i(t)] \\ \mathbf{P}_i(t-1), & \text{if } f[\mathbf{X}_i(t)] \geq f[\mathbf{P}_i(t)] \end{cases} \quad (2.13)$$

全局最优位置则是粒子群整体搜索到的当前最优位置，用下述公式表征： $\mathbf{G}(t)=[P_{g1}(t), P_{g2}(t), \dots, P_{gM}(t)]$, $1 \leq g \leq M$ 。粒子群算法的速度和位置更新通过下式计算得到：

$$V_{ij}(t+1) = V_{ij}(t) + c_1 \cdot \text{rand} \cdot (P_{ij}(t) - X_{ij}(t)) + c_2 \cdot \text{rand} \cdot (G_j(t) - X_{ij}(t)) \quad (2.14)$$

$$X_{ij}(t+1) = V_{ij}(t+1) + X_{ij}(t) \quad (2.15)$$

上式中， $1 \leq i \leq S, 1 \leq j \leq M$ ， t 表示迭代次数， c_1 和 c_2 表示学习因子，其中， c_1 用于

调整各个粒子到达自身最优位置的步长， c_2 则用于调整各个粒子到达全局最优位置的步长，而上式中的 rand 表示随机数，并在 0-1 之间均匀分布；将粒子的速度变化范围严格的控制在下述区间内 $V_{ij}(t) \in [-V_j^{\max}, V_j^{\max}]$ ，同时将粒子的位置控制在下式表示的搜索空间 $X_{ij}(t) \in [-X_j^{\max}, X_j^{\max}]$ 内，以使得粒子在搜索过程中尽量降低其飞离上述搜索空间的概率。

通过以上对于粒子群算法的介绍，将基于粒子群算法的协作频谱感知方法的流程归纳如下：

Step 1: 参数初始化。设置 $t=0$ ，设定粒子群体规模 S 。对任意指定的 i 和 j ，初始位置和初始速度 $X_{ij}(0) \in [-X_j^{\max}, X_j^{\max}]$ ， $V_{ij}(0) \in [-V_j^{\max}, V_j^{\max}]$ 随机产生。初始化 $P_{ij}(0) = X_{ij}(0)$ 。归一化处理各个粒子的初始位置。

Step 2: 针对各个粒子，得到其当前位置 $X_i(t)$ 的适应度值 $f[X_i(t)]$ 。

Step 3: 针对各个粒子，比较上述位置 $X_i(t)$ 的适应度值 $f[X_i(t)]$ 和粒子的个体最优位置 $P_i(t)$ 的适应度值 $f[P_i(t)]$ ，如果上述适应度值比 $f[P_i(t)]$ 好，则替代 $P_i(t) = X_i(t)$ 。

Step 4: 针对各个粒子，比较上述位置 $X_i(t)$ 的适应度值 $f[X_i(t)]$ 和粒子群整体的全局最优位置 $G(t)$ 的适应度值 $f[G(t)]$ ，同样的，如果上述适应度值比 $f[G(t)]$ 好，则替代 $G(t) = X_i(t)$ 。

Step 5: 根据上述公式 2.14) 和公式(2.15)，更新各个粒子的速度以及位置。对新的速度和位置进行钳位处理，并对经过钳位处理的粒子位置进行归一化处理。

Step 6: 如果没有达到结束条件，这一结束条件可以是预先设定的一个最大迭代次数阈值，那么令 $t=t+1$ ，重新执行 Step 2。

2.5 仿真试验及结果分析

本节将基于修正偏差因子的协作感知方法与基于经典粒子群算法的协作感知方法进行比较，仿真过程中，虚警概率参数取值如下： $P_f=0.1$ ， $P_f=0.2$ 和 $P_f=0.3$ ，并将得到的实验结果填入表格 2.1-2.2 内。假设授权用户发出的信号 $s(k)=1$ ，采样次数 $N=20$ ，仿真过程进行 200 次独立试验。本文中将仿真参数分布设置为如下向量所示： $\sigma = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]^T$ ； $\delta = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]^T$ ； $h = [0.6, 0.5, 0.7, 0.3, 0.4, 0.5, 0.3, 0.4]^T$ 。粒子群算法的参数设置为：学习因子 c_1 与 c_2 相等，

均设置为 2，将粒子总数取值 50，并将所有粒子的速度变化范围严格限制在所述区间的十分之一，仿真过程中迭代次数设置为 100 次。从仿真结果可看出：基于粒子群算法的感知方法优于基于修正偏差因子的感知方法，从而也证实了修正偏差因子方法确实不是一种全局最优的求解方法。

表 2.1 本地认知用户为 8 个时的目标函数仿真结果

虚警概率	$P_f=0.1$	$P_f=0.2$	$P_f=0.3$
MDC	-2.08374	-2.39648	-2.62200
PSO	-2.08581	-2.39720	-2.62202

表 2.2 本地认知用户为 8 个时的检测概率仿真结果

虚警概率	$P_f=0.1$	$P_f=0.2$	$P_f=0.3$
MDC	0.98140	0.99172	0.99562
PSO	0.98150	0.99173	0.99563

表 2.3 和表 2.4 给出了另一种仿真参数条件下的仿真结果，仿真参数设置： $\sigma=[0.9,1.3,1.0,2.0,0.8,1.8]^T$ ； $\delta=[0.4,0.6,0.4,0.2,0.5,0.4]^T$ ； $h=[0.6,0.5,0.7,0.3,0.4,0.5]^T$ 。其余参数设置与上一仿真试验相同。从仿真结果可以进一步证实修正偏差因子方法不是一种全局最优的求解方法。

表 2.3 本地认知用户为 6 个时的目标函数仿真

虚警概率	$P_f=0.1$	$P_f=0.2$	$P_f=0.3$
MDC	-1.69131	-1.99700	-2.21742
PSO	-1.69539	-1.99867	-2.21797

表 2.4 本地认知用户为 6 个时的检测概率仿真

虚警概率	$P_f=0.1$	$P_f=0.2$	$P_f=0.3$
MDC	0.95461	0.97708	0.98670
PSO	0.95499	0.97717	0.98672

2.6 本章小结

本章中研究了频谱感知的采用的典型检测方法，本章涵盖了从简单到复杂，由理想的假设条件到实际中的非理想的条件的研究，以实现与实际通信中认知无线电

的应用场景更加贴近和实用。由于单节点检测方法在检测性能上存在缺陷，故本文采用一种线性协作频谱检测方法，该检测方法是本文后续内容的基础。另外本章还介绍了基于修正偏差因子的协作频谱感知方法和基于经典粒子群算法的协作频谱感知方法。本文还仿真验证了现有的通用偏差因子的协作频谱感知方法，通过仿真结果得出结论，这一算法是次优的，难以取得理论最优解。

第3章 基于量子蛙跳算法的认知无线电频谱感知

前面介绍了常用的认知无线电协作频谱感知模型，可以将上述模型看作是比较典型的连续优化问题。目前基于达尔文进化论观点的群智能算法^[36]是解决优化问题的一个很好的办法。但是传统的人工智能算法对于协作频谱感知问题的处理，都存在着收敛精度低亦或是收敛速度慢的缺陷。因此，本章创新性地引入量子计算的概念，设计了全新的量子蛙跳算法。本文还对 Benchmark 函数的测试，并进一步证明所提出的算法是有效且通用的。同时，将量子蛙跳算法应用于解决认知协作频谱感知问题，并给出了相应的仿真结果。仿真结果表明，相比经典群智能算法频谱感知方法言，基于量子蛙跳算法的频谱感知算法有更高的收敛精度、更优的收敛速度，在工程应用领域进一步证实了所设计的新算法具有优越的性能和巨大的潜力。

3.1 量子进化概述

对于量子计算(Quantum Computation)的研究最早开始于 1982 年，在当时诺贝尔物理学奖获得者 Richard Feynman 创新性地将计算看作是一个物理过程^[37]。科学计算发展迅猛，量子计算以其独特的特质和优势作为前沿学科，受到世界各国的研究和紧密跟踪。量子计算的运算能力非常强大，尤其是其还具有的指数存储容量、量子并行以及指数加速等诸多优势^[38-39]。有一些计算机专家甚至预言，随着量子计算研究中一个个难题的突破与解决，在本世纪肯定会出现比现在的通用电子计算机运算器速度快 1000 倍以上的量子计算机。

与传统的计算方法不同，量子计算是由量子力学直接进入算法领域。与其他传统的计算方法最根本的不同是，量子计算具有特有的能够加速计算速度的量子并行性^[40-41]。量子态具有叠加性和相干性，同时量子比特之间还具有纠缠性，正是利用这种特性，量子计算显示出不可比拟的巨大优势。

基于上述核心思想，量子计算理论通过融合进化算法可以得到量子进化算法，该算法是与传统算法不同的全新的理论框架，该算法是在量子计算的理论的基础上，并且通过使用量子比特对状态编码。

量子进化算法中用量子位表示最小的信息单元，并将其称作是量子比特。量子比特的状态有 0 或 1 两种取值，其叠加状态由下述公式表示为：

$$|\Phi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (3.1)$$

上式中， α 和 β 是代表各自对应的状态出现概率， α 和 β 均为复数，并且满足 $|\alpha|^2 + |\beta|^2 = 1$ ， $|\alpha|^2$ 和 $|\beta|^2$ 则分别表示量子比特在状态 0 和状态 1 时的概率值。当 $|\alpha|^2 = 1$ 时，表示状态 $|0\rangle$ 为概率 1，而状态 $|1\rangle$ 不出现；反之，当 $|\beta|^2 = 1$ 时，表示状态 $|1\rangle$ 为概率 1，而状态 $|0\rangle$ 则不出现。通过外界的观测测量，使量子位状态从 $|0\rangle$ 和 $|1\rangle$ 的叠加状态，坍塌到与观测相应的特定状态结果。

在量子计算中，最重要也最根本的是可以利用量子态对信息编码。如前文所述，量子比特是量子计算中最小的信息存储单元。量子计算过程是对量子态编码的幺正变换，这样才能保证量子态的正交归一性。因此，在量子计算时，所述的基本逻辑门操作是上述提及的的幺正变换。这些幺正变换在量子进化算法中被称作量子门。量子门的类型很多，下面介绍几种较常用的量子门幺正矩阵：

(1) 量子非门

$$U_{not} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.2)$$

U_{not} 可以使 $|\Phi\rangle$ 中的 $|0\rangle$ 变成 $|1\rangle$ ，并将 $|1\rangle$ 变成 $|0\rangle$ ，具体参见下述公式所示：

$$U_{not} |\Phi\rangle = U_{not} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \alpha|1\rangle + \beta|0\rangle \quad (3.3)$$

(2) 量子 Hadamard 门

$$U_H = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \quad (3.4)$$

Hadamard 门的作用是对一个量子位进行变换，成为两个量子位的相干叠加

$$U_H |0\rangle = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad (3.5)$$

$$U_H |1\rangle = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (3.6)$$

(3) 量子旋转门

$$U_R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (3.7)$$

其中 θ 表示量子旋转角。

3.2 混合蛙跳算法介绍

混合蛙跳算法(SFLA)^[42-43]是由 Eusuff 和 Lansey 两位学者在 2000 年率先提出的,该算法来自于青蛙觅食行为的研究。该算法模拟青蛙种群的跳跃行为追寻最优解,特别是连续优化问题的求解具有较快的收敛速度。其与文化基因传承思想一致,通过在族群内采用局部搜索与全局信息相混合的协同搜索方法以得到最优解^[44]。大量实验、仿真、测试都一定程度的表明,该算法在对高维、多局部极值等类似的函数优化问题求解时,效果明显并有独特的优势。混合蛙跳算法概念非常简单易于理解,其参数少易实现而且计算速度快,目前已广泛成功应用于函数优化、组合优化、多用户检测、资源网络分配及管网优化等方面。最重要的是混合蛙跳算法相比其他智能算法而言,具有比较少的控制参数。

本文简要介绍 SFLA 的基本步骤,并以 M 维函数的最小化作为实施例。假设青蛙种群的规模为 P 。在 M 维空间内,假设第 i 只青蛙的坐标是 $\mathbf{X}_i = [X_{i1}, X_{i2}, \dots, X_{iM}]$ 。每只青蛙个体的适应度用其坐标相关的函数 $f(\mathbf{X}_i)$ 表示,根据上述适应度,种群中的青蛙个体按降序排列。然后将所有青蛙个体划分为 m 个族群,每个族群中青蛙个体数目为 n 。按如下策略进行分配:序号为 1 的青蛙放入第 1 个族群,序号为 2 的青蛙放入第 2 个族群, ..., 序号为 m 的青蛙放入第 m 个族群。然后,序号为 $m+1$ 的青蛙放入第 1 个族群,并依次类推,直至所有青蛙分配完毕。分别用 $\mathbf{X}_b = [X_{b1}, X_{b2}, \dots, X_{bM}]$ 和 $\mathbf{X}_w = [X_{w1}, X_{w2}, \dots, X_{wM}]$ 分别表示 m 个族群中,适应度最优和最劣青蛙的位置, $\mathbf{X}_g = [X_{g1}, X_{g2}, \dots, X_{gM}]$ 表示全局最优适应度青蛙的位置。对一次族群范围内迭代过程,具有最差适应度青蛙位置 \mathbf{X}_w 的第 j 维分量按下式更新:

$$D_j = \text{rand} \cdot (X_{bj} - X_{wj}) \quad (3.8)$$

$$X_{wj}^{\text{new}} = X_{wj} + D_j \quad (3.9)$$

上式中, D_j 是在更新时分量 j 移动的距离,初始随机产生 $D_j \in [d_j^{\min}, d_j^{\max}]$, d_j^{\min} 为最小移动步长, d_j^{\max} 为最大的移动步长。若 $\mathbf{X}_w^{\text{new}}$ 适应度结果没有得到优化,则将公式(3.8)中的 X_{bj} 替换为 X_{gj} ,使青蛙更有效往全局最优位置移动,同时按下式更新

$$D_j = \text{rand} \cdot (X_{gj} - X_{wj}) \quad (3.10)$$

$$X_{wj}^{\text{new}} = X_{wj} + D_j \quad (3.11)$$

如果 $\mathbf{X}_w^{\text{new}}$ 的适应度仍然没有得到改进,则继续在定义域区间内继续随机产生新

的青蛙位置 \mathbf{X}_w^{new} 。并通过上述步骤实现了族群内的一次迭代，并不断重复以上更新操作直至迭代完成。直到所有族群完成上述相同的更新步骤，便完成一个进化过程。在完成该进化过程后再次混合所有青蛙，计算适应度并按其结果降序排列青蛙，继续进行下一个进化过程。

综上所述，基于混合蛙跳算法解决优化问题的流程如下

(1) 参数初始化。 p 表示青蛙种群总数， m 表示青蛙的族群数， n 表示每个族群中青蛙的数目， gen 表示族群内的最大迭代次数， Gen 表示整个群体的最大进化代数。依据上述定义，随机生成青蛙初始位置。

(2) 计算每只青蛙的适应度值。并根据该适应度值的降序，对青蛙进行排列。 \mathbf{X}_g 表示适应度全局最优青蛙的位置。

(3) 划分全部群体变为 m 个族群。分别用参数 \mathbf{X}_b 和 \mathbf{X}_w 表示在该族群内的最优的适应度值和最劣的适应度值的青蛙位置。

(4) 设置 $i_m=0$ ，其中 i_m 为当前族群数。

(5) $i_m = i_m + 1$ ，设置 $i_{gen}=0$ ，其中 i_{gen} 为该族群的迭代次数。

(6) $i_{gen} = i_{gen} + 1$

(7) 在族群 i_m 内，对最差适应度青蛙的位置 \mathbf{X}_w 进行更新：

$$D_j = \text{rand} \cdot (\mathbf{X}_{bj} - \mathbf{X}_{wj}), \quad \mathbf{X}_{wj}^{new} = \mathbf{X}_{wj} + D_j。$$

若是适应度值没有得到改进，则用 \mathbf{X}_g 代替 \mathbf{X}_b ，使其更有效趋向全局最优解：

$$D_j = \text{rand} \cdot (\mathbf{X}_{gj} - \mathbf{X}_{wj}), \quad \mathbf{X}_{wj}^{new} = \mathbf{X}_{wj} + D_j。$$

如果上述表示的适应度值仍然没有得到改善，则在定义域区间内产生一只新的青蛙位置 \mathbf{X}_w^{new} 。

如此完成族群 i_m 的依次循环，对族群内的 \mathbf{X}_b ， \mathbf{X}_w 和 \mathbf{X}_g 进行更新。

(8) 如果 $i_{gen} < gen$ ，重复步骤 (6)。

(9) 如果 $i_m < m$ ，重复步骤 (5)。

(10) 如果达到最大进化迭代次数，输出 \mathbf{X}_g 及其对应的最优值，计算完成；

否则进入步骤 (2)。

3.3 基于量子蛙跳算法的认知无线电频谱感知

传统的混合蛙跳算法虽然是一种优秀的群智能算法，但是它具有容易陷入局部

收敛这一瓶颈。本文引入了量子计算，通过量子化传统的蛙跳算法，可以得到一种新型量子蛙跳算法(QSFLA)，便可很好的克服这一瓶颈，并且可以很好地解决认知无线电频谱感知这一连续优化问题。

3.3.1 量子蛙跳算法介绍

混合蛙跳算法是基于特定青蛙个体以及全体青蛙个体的信息交换过程来进化的一种群体智能方法^[45]。但是混合蛙跳算法并不改变青蛙个体的信息，也不会综合全部个体的主流思想，这样会限制算法的寻优能力，为此本文提出量子蛙跳算法(QSFLA)。QSFLA算法将量子计算创新性地引入到青蛙跳跃方程中，设计出全新的跳跃方程。

在量子蛙跳算法QSFLA中，用下述公式表示第*i*只青蛙的量子位置：

$$\mathbf{d}_i = [d_{i1} \quad d_{i2} \quad \cdots \quad d_{iM}] = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iM} \\ \beta_{i1} & \beta_{i2} & \cdots & \beta_{iM} \end{bmatrix} \quad (3.12)$$

其中 $|x_{ij}|^2 + |\beta_{ij}|^2 = 1$, ($j=1,2,\dots,M$)。定义 x_{ij} 和 β_{ij} 为实数并且 $0 \leq x_{ij} \leq 1$, $0 \leq \beta_{ij} \leq 1$ ^[46]。 θ_{ij} 表示量子旋转角，用 \mathbf{d}_{ij} 来表示第*i*只青蛙的量子位置，该位置值通过量子旋转门 $U(\theta_{ij})$ 进行更新。上述第*i*只青蛙的量子位置中的第*j*个量子位置的更新方式下述公式表示：

$$\mathbf{d}_{ij}^{new} = \text{abs}(U(\theta_{ij})\mathbf{d}_{ij}) = \text{abs}\left(\begin{bmatrix} \cos \theta_{ij} & -\sin \theta_{ij} \\ \sin \theta_{ij} & \cos \theta_{ij} \end{bmatrix} \mathbf{d}_{ij}\right) \quad (3.13)$$

根据文献[47]的方式，第*i*只青蛙的量子位置可以表示为 $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$ ，第*i*只青蛙的量子位置的第*j*个量子位置按照如下方式更新

$$x_{ij}^{new} = |x_{ij} \cos \theta_{ij} - \sqrt{1 - (x_{ij})^2} \sin \theta_{ij}| \quad (3.14)$$

第*i*只青蛙的位置为 $\bar{\mathbf{x}}_i = [\bar{x}_{i1}, \bar{x}_{i2}, \dots, \bar{x}_{iM}]$ ，也就是其 $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$ 量子位置的映射位置。其中 $0 \leq x_{ij} \leq 1$, $\bar{x}_{ij} = l_j + x_{ij}(u_j - l_j)$, $l_j \leq \bar{x}_{ij} \leq u_j$ 。 l_j 是第*j*维变量的下限， u_j 是第*j*维变量的上限。第*i*只青蛙的适应度值为 $f(\bar{\mathbf{x}}_i)$ 。

在求解 M 维优化问题中，量子蛙跳算法在定义域内随机产生 p 个青蛙。 $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$ 是指第 i 只青蛙的量子位置，对于第一代的进化，根据各个青蛙的适应度值将其降序排列，随后，将这些青蛙均等的分成 m 个族群，并且保证每个族群中都具有 n 只青蛙，不难看出， $p = m \times n$ 。上述划分操作具体如下：

分别把序号为 1 的青蛙放入第 1 个族群，序号为 2 的青蛙放入第 2 个族群，...，序号为 m 的青蛙放入第 m 个族群。然后，序号为 $m + 1$ 的青蛙放入第 1 个族群，并依次类推，直至所有青蛙分配完毕。在每次迭代中，在根据适应度值表示，即用 $\mathbf{x}_b = [x_{b1}, x_{b2}, \dots, x_{bM}]$ 和 $\mathbf{x}_w = [x_{w1}, x_{w2}, \dots, x_{wM}]$ 表示每个族群中最优和最差适应度值的青蛙量子位置。并且用 $\mathbf{x}_g = [x_{g1}, x_{g2}, \dots, x_{gM}]$ 来表示在所有青蛙中具有全局最优适应度值的青蛙量子位置。

上述操作可以把青蛙种群划分成为一个个族群，在划分得到的每个族群中，还可以继续包括子群。每个青蛙的跳跃方程由下述策略决定：

并根据下述公式对具有最差适应度值的青蛙的量子位置 \mathbf{x}_w 实施更新操作：

$$\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{bj}) + \text{randn} \cdot (x_{wj} - z_j) \quad (3.15)$$

$$x_{wj}^{\text{new}} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}| \quad (3.16)$$

上式中，用 rand 表示 0-1 区间内的均匀随机数，用 randn 表示高斯分布的随机变量，其均值为 0，方差为 1。用下式 \mathbf{z} 表示族群中所有青蛙的量子位置的平均：

$$\mathbf{z} = [z_1, z_2, \dots, z_M] \quad (3.17)$$

其中 $z_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$ ， $j = 1, 2, \dots, M$ 。

量子位置 $\mathbf{x}_w^{\text{new}}$ 的适应度结果如果没有得到优化，则按照下式更新：

$$\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{gj}) + \text{randn} \cdot (x_{wj} - z_j) \quad (3.18)$$

$$x_{wj}^{\text{new}} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}| \quad (3.19)$$

如果上述最差适应度值仍没有得到改善，则在上述定义域内生成用 $\mathbf{x}_w^{\text{new}}$ 表示的新的随机的青蛙量子位置。

不断重复上述更新操作，直至完成所有迭代次数。当全部族群完成了内部进化之后，将青蛙种群重新进行混合，进入下一代进化，直到达到预先设定的最大进化代数。

综上所述，量子蛙跳算法的流程如下：

(1) 参数初始化。 p 表示青蛙群体总数， m 表示青蛙族群数， n 表示每个族群内的青蛙数，其中 $p = m \times n$ 。 gen 表示族群内的最大迭代次数， Gen 表示整个群体的最大进化代数。依据上述定义，随机生成青蛙初始量子位置。

(2) 计算每只青蛙的适应度值。并根据该适应度值的降序，对青蛙进行排列。

\mathbf{x}_g 表示适应度全局最优青蛙的量子位置。

(3) 将全部群体分为 m 个族群。划分规则：分别把第 1 只青蛙放入第 1 个族群中，再将第 2 只青蛙放入第 2 个族群中，...，类似地，将第 m 只青蛙放到第 m 个族群中，所有族群放完后，在从头操作，即，将第 $m+1$ 只青蛙放入到第 1 个族群中，根据这种方式放置青蛙，直到所有的青蛙都放置完成。并分别用 \mathbf{x}_b 和 \mathbf{x}_w 表示在族群内具有最好适应度值和最差适应度值的青蛙量子位置。

(4) 设置 $i_m=0$ ，其中 i_m 为当前族群数。

(5) $i_m = i_m + 1$ ，设置 $i_{gen}=0$ ，其中 i_{gen} 为该族群的迭代次数。

(6) $i_{gen} = i_{gen} + 1$

(7) 在族群 i_m 内，对具有最差适应度值的青蛙量子位置 \mathbf{x}_w 依据上述量子蛙跳算法的策略进行更新。首先，按照如下方程更新 \mathbf{x}_w

$$\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{bj}) + \text{randn} \cdot (x_{wj} - z_j), \quad x_{wj}^{new} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}|。$$

如果新的青蛙量子位置的适应度值没有改进，则在更新策略中使用 \mathbf{x}_g 代替 \mathbf{x}_b ，使其更快地向全局最优解收敛

$$\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{gj}) + \text{randn} \cdot (x_{wj} - z_j), \quad x_{wj}^{new} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}|。$$

如果上述适应度值仍然尚未得到有效改善，则还需产生用 \mathbf{x}_w^{new} 表示的青蛙量子位置，该新位置是在量子域范围内随机产生的。

这样便完成了族群 i_m 的依次循环，更新该族群内的 \mathbf{x}_b ， \mathbf{x}_w 和 \mathbf{x}_g 。

(8) 如果 $i_{gen} < gen$ ，进入步骤 (6)。

(9) 如果 $i_m < m$ ，进入步骤 (5)。

(10) 如果达到最大进化迭代次数，输出 \mathbf{X}_g 及其对应的最优值，计算完成；否则进入步骤 (2)。

3.3.2 量子蛙跳算法的测试

本文选取四个通用测试函数^[49-50]对量子蛙跳算法进行测试，并且将测试结果与传统的混合蛙跳算法^[42-43]、粒子群算法^[51]以及细菌觅食算法^[52]进行了比较。

选取的测试函数分别为 Sphere 函数、Sum Squares 函数、Griewank 函数和 Ackley 函数，其公式分别为

$$f_1(\mathbf{x}) = \sum_{j=1}^M x_j^2 \quad (3.20)$$

其中, $x_j \in [-5.12, 5.12], (j=1, 2, \dots, M)$ 。

$$f_2(\mathbf{x}) = \sum_{j=1}^M jx_j^2 \quad (3.21)$$

其中, $x_j \in [-10, 10], (j=1, 2, \dots, M)$ 。

$$f_3(\mathbf{x}) = \frac{1}{4000} \sum_{j=1}^M x_j^2 - \prod_{j=1}^M \cos\left(\frac{x_j}{\sqrt{j}}\right) + 1 \quad (3.22)$$

其中, $x_j \in [-10, 10], (j=1, 2, \dots, M)$ 。

$$f_4(\mathbf{x}) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{M} \sum_{j=1}^M x_j^2}\right) - \exp\left(\frac{1}{M} \sum_{j=1}^M \cos 2\pi x_j\right) \quad (3.23)$$

其中,。

量子蛙跳算法的参数设置如下: 将青蛙群体总数 p 设置为 50, 将青蛙族群数 m 设置为 10, 同时将每个族群青蛙数 n 设置为 5, 每个族群内的最大迭代次数 gen 设置为 5, 而总的最大进化代数 Gen 设置为 1000。

对于混合蛙跳算法, 选取与量子蛙跳算法一致的参数 p, m, n, gen, Gen , 另外 d_j^{\max} 和 d_j^{\min} 设置为: $d_j^{\max} = 0.5, d_j^{\min} = -0.5$ 。

细菌觅食算法的参数设置为: $S=50, N_c=100, N_s=4, N_{re}=5, N_{ed}=2, P_{ed}=0.1, c(i)=0.01$ 。

粒子群算法选取如下参数: 学习因子 $c_1=c_2=2$, 粒子总数 50, 所有粒子速度变化的范围为定义区间的 10%。

在整个测试过程中测试函数的维数取 $M=35$, 试验的迭代次数为 1000, 运行 200 次独立试验。

仿真后得到的收敛曲线如图 3.1-3.4 所示, 由仿真图可知, 尽管混合蛙跳算法和细菌觅食算法的收敛速度很快, 但是收敛精度不高; 而粒子群算法的收敛精度相对来说比较高, 但缺点是它的收敛速度很慢。在所有 4 种智能算法中, 量子蛙跳算法无论收敛速度还是收敛性能上都是最优的。

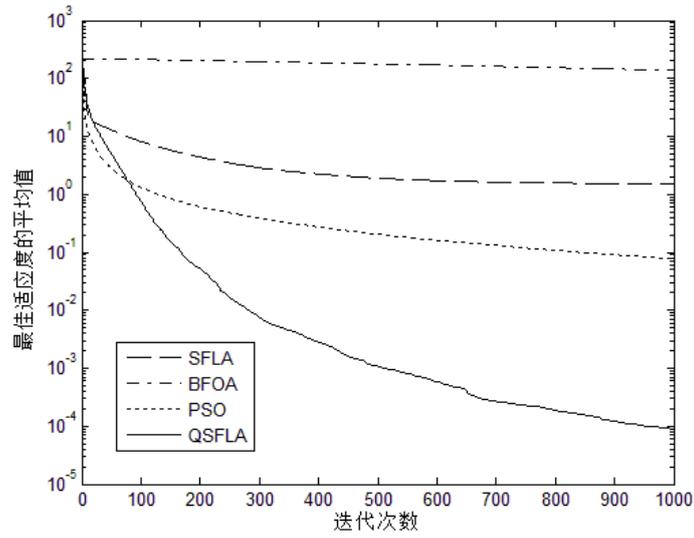


图 3.1 Sphere 函数的收敛曲线

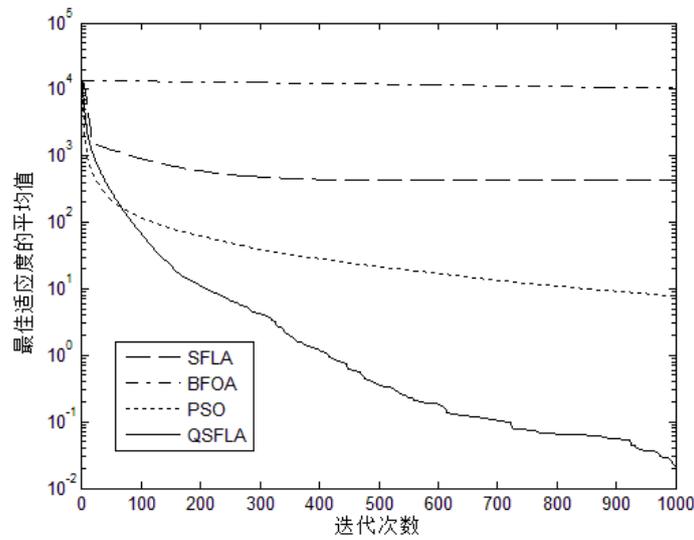


图 3.2 Sum Squares 函数的收敛曲线

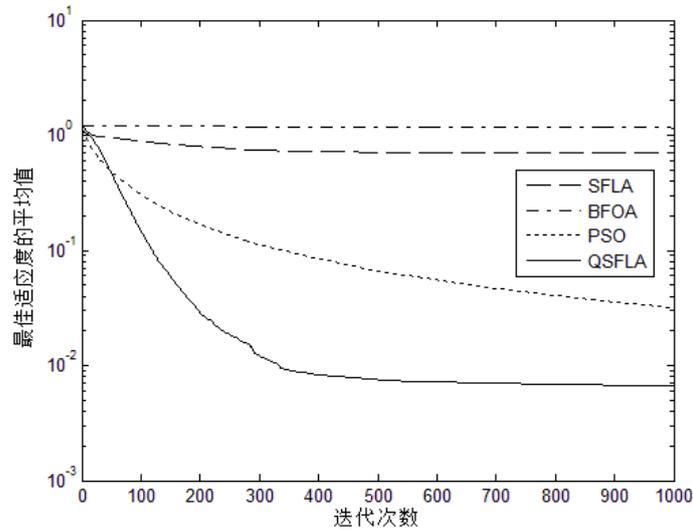


图 3.3 Griewank 函数的收敛曲线

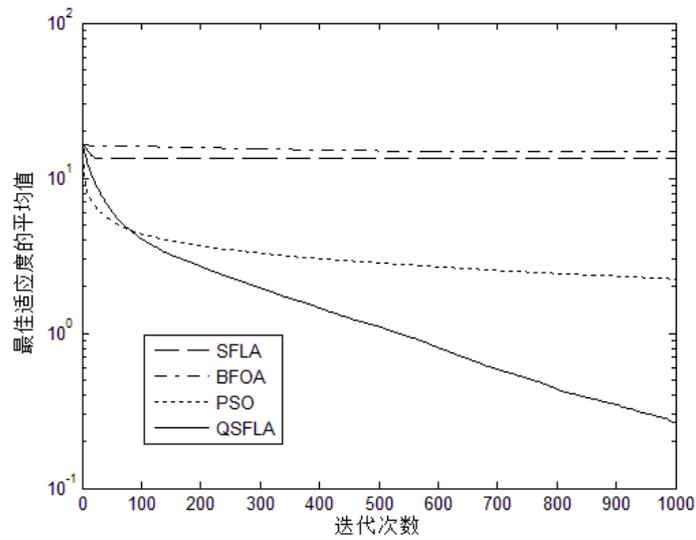


图 3.4 Ackley 函数的收敛曲线

3.3.3 基于量子蛙跳算法的认知无线电频谱感知

综上所述，利用量子蛙跳算法实现对频谱感知问题的求解，可以转化成对公式(2.9)进行优化。从而求解协作频谱感知检测概率最大化的技术问题，便可转化成求解：

$$\mathbf{w}^\square = \arg \min_{\mathbf{w}} \frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}} \quad (3.24)$$

可以看出该式最优解由权重向量 \mathbf{w}^\square 决定， \mathbf{w}^\square 同样使得在预定虚警概率值 P_f ，能

够获得最大的检测概率。如果 \mathbf{w}^\square 是使(3.24)式最小化的最优解，则 $\lambda \mathbf{w}^\square$ 也是使(3.24)改式最小化的最优解，明显就会有无穷个最优解。对 \mathbf{w}^\square 进行归一化操作的约束条件为：

$$\min_{\mathbf{w}} f(\mathbf{w}), \text{s.t. } 0 \leq w_j \leq 1, j=1, 2, \dots, M, \sum_{j=1}^M w_j = 1 \quad (3.25)$$

由此设计适应度函数为

$$\begin{cases} f(\mathbf{w}) = \frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}}, \\ \text{s.t. } 0 \leq w_j \leq 1, j=1, 2, \dots, M, \sum_{j=1}^M w_j = 1. \end{cases} \quad (3.26)$$

此优化问题的最优解就是使得(3.24)式最小化的最优权重向量。

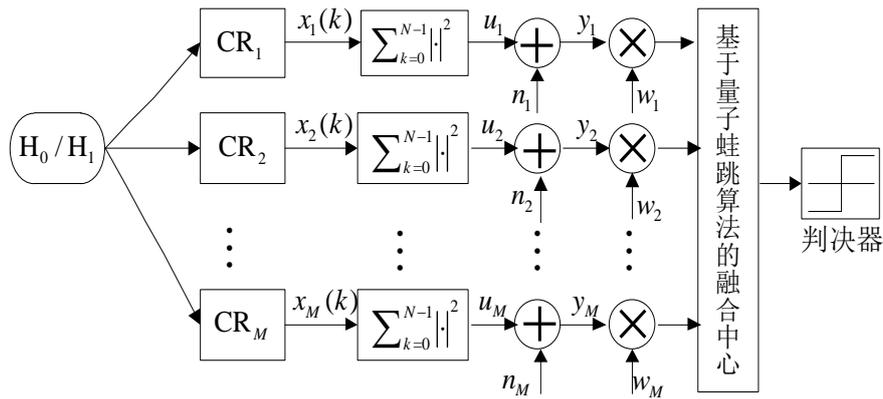


图 3.5 基于量子蛙跳算法的协作频谱感知模型框图

可以利用量子蛙跳算法来实现优化。在基于量子蛙跳算法的协作感知方法中创新性的设计了基于量子蛙跳算法的融合中心，其原理框图如图 3.5 所示。对于协作频谱感知模型，青蛙的位置与量子位置是等价的，在量子域[0,1]内随机产生量子位。因此，对于协作感知模型的目标函数的优化问题可以等价于量子蛙跳算法的适应度函数的优化问题，第 i 个青蛙的量子位置 $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$ 的适应度函数可以表示为

$$f(\mathbf{x}_i) = \frac{Q^{-1}(P_f) \sqrt{\mathbf{x}_i \mathbf{A}(\mathbf{x}_i)^T} - E_s \mathbf{h}^T(\mathbf{x}_i)^T}{\sqrt{\mathbf{x}_i \mathbf{B}(\mathbf{x}_i)^T}}, \text{s.t. } 0 \leq x_{ij} \leq 1, j=1, 2, \dots, M, \sum_{j=1}^M x_{ij} = 1. \quad (3.27)$$

由此可知，本文中的基于量子蛙跳算法的计算协作频谱感知权重的操作过程具体如下步骤：

(1) 参数初始化。青蛙群体总数用 p 来表示，青蛙族群数 m 来表示，每个族群内的青蛙数 n 来表示。因此可以得到 $p = m \times n$ 。同样还需要设置族群内的最大迭代次数 gen 以及整个群体的最大进化代数 Gen 。在定义域内随机产生青蛙的初始量子位置，并且对在定义域内随机分布的青蛙的初始位置进行归一化处理。

(2) 计算每只青蛙的适应度值，并根据计算得到适应度值的降序对青蛙排列。用 \mathbf{x}_g 表示青蛙量子位置为全局最优适应度的青蛙量子位置。

(3) 将全部群体分为 m 个族群。分别用 \mathbf{x}_b 和 \mathbf{x}_w 表示在族群内具有最优适应度值和最差适应度值的青蛙量子位置。

(4) 设置 $i_m = 0$ ，其中 i_m 为当前族群数。

(5) $i_m = i_m + 1$ ，设置 $i_{gen} = 0$ ，其中 i_{gen} 为该族群的迭代次数。

(6) $i_{gen} = i_{gen} + 1$

(7) 在族群 i_m 内，对具有最差适应度值的用 \mathbf{x}_w 表示的青蛙量子位置继续依据前述量子蛙跳算法的进行更新操作。首先，按照如下方程更新 \mathbf{x}_w

$$\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{bj}) + \text{randn} \cdot (x_{wj} - z_j), \quad x_{wj}^{new} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}|。$$

归一化处理青蛙新的量子位置 \mathbf{x}_w^{new} 。如果新量子位置的适应度值没有得到改进，使用 \mathbf{x}_g 代替 \mathbf{x}_b ，使其更快地向全局最优解收敛

$$\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{gj}) + \text{randn} \cdot (x_{wj} - z_j), \quad x_{wj}^{new} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}|。$$

继续归一化处理青蛙新的量子位置 \mathbf{x}_w^{new} 。如果结果还未得到改进，则在量子域内随机产生全新青蛙量子位置 \mathbf{x}_w^{new} ，继续对 \mathbf{x}_w^{new} 进行归一化处理。

如此完成族群 i_m 的循环，更新 \mathbf{x}_b ， \mathbf{x}_w 和 \mathbf{x}_g 。

(8) 如果 $i_{gen} < gen$ ，进入步骤 (6)。

(9) 如果 $i_m < m$ ，进入步骤 (5)。

(10) 如果达到最大进化迭代次数，输出 \mathbf{X}_g 及其对应的最优值，计算完成；否则进入步骤 (2)。

3.3.4 仿真实现与结果分析

仿真实验中，针对协作频谱感知应用背景，对量子蛙跳算法与传统蛙跳算法、细菌觅食算法和粒子群算法进行了比较，其参数与 3.3.2 节所选取参数相同。假设该认知网络中的认知用户的个数为 M ，而在这些认知用户中，每一个用户都可以各自

独立的检测自身目标频谱。认知网络中的权重向量 $\mathbf{w} = [w_1, w_2, \dots, w_M]$ 为一个 M 维向量，在所提出的基于量子蛙跳算法的协作感知模型中，第 i 只青蛙维数与权重向量维数一致，也等于认知用户的数量。从而达到青蛙与上述需要优化的权重向量之间映射的目的。文中还做出如下假定：授权用户信号 $s(k) = 1$ ，采样次数 $N = 20$ 。

图 3.6 和图 3.7 分别为虚警概率 $P_f = 0.1$ ，18 个认知用户条件下，选取不同迭代次数时的最优目标函数值和检测概率的变化曲线。此时进行 200 次独立试验，智能计算算法的终止迭代次数设为 200。除上述参数外，上述协作频谱感知模型所需要的其他参数表示为：

$$\sigma = [2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5, 0.9, 2.7, 2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5]^T$$

$$\delta = [1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8, 2.0, 3.1, 1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8]^T$$

$$h = [0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5, 0.2, 0.3, 0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5]^T$$

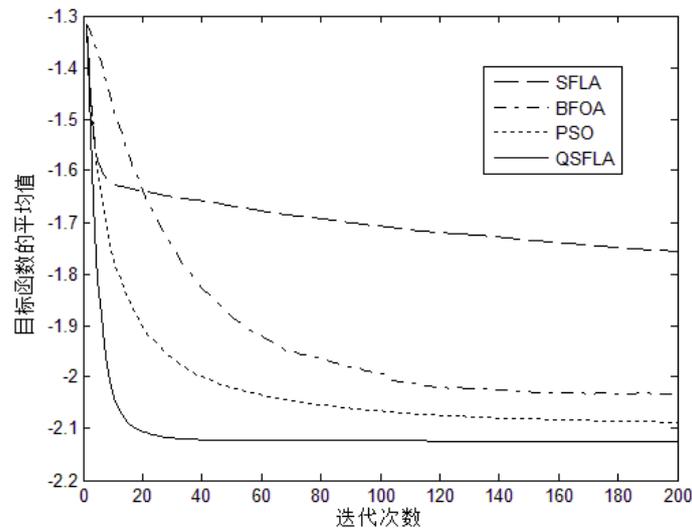


图 3.6 有 18 个本地认知用户时的目标函数仿真结果

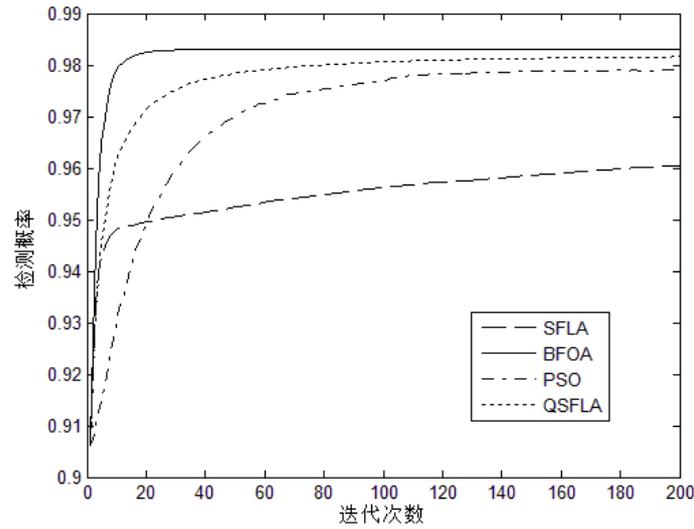


图 3.7 有 18 个本地认知用户时的检测概率仿真结果

图 3.8 和图 3.9 则给出假设网络中有 30 个认知用户时, 给定虚警概率 $P_f=0.1$ 时, 当迭代次数取值不同时, 最优目标函数值和检测概率的变化曲线。此时进行 200 次独立试验, 智能计算算法终止迭代次数设为 400。参数设置:

$$\sigma = [2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5, 0.9, 2.7, 2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5, 0.9, 2.7, 2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5, 0.9, 2.7]^T$$

$$\delta = [1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8, 2.0, 3.1, 1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8, 2.0, 3.1, 1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8, 2.0, 3.1]^T$$

$$h = [0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5, 0.2, 0.3, 0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5, 0.2, 0.3, 0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5, 0.2, 0.3]^T$$

其余参数与上一仿真实验相同。

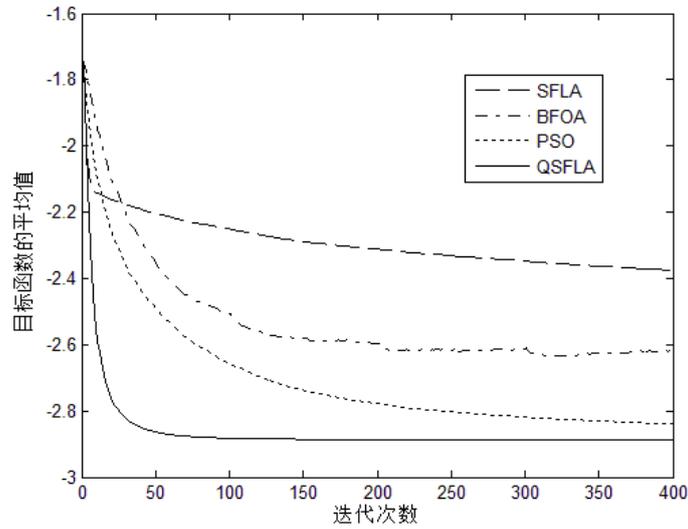


图 3.8 有 30 个本地认知用户时的目标函数仿真结果

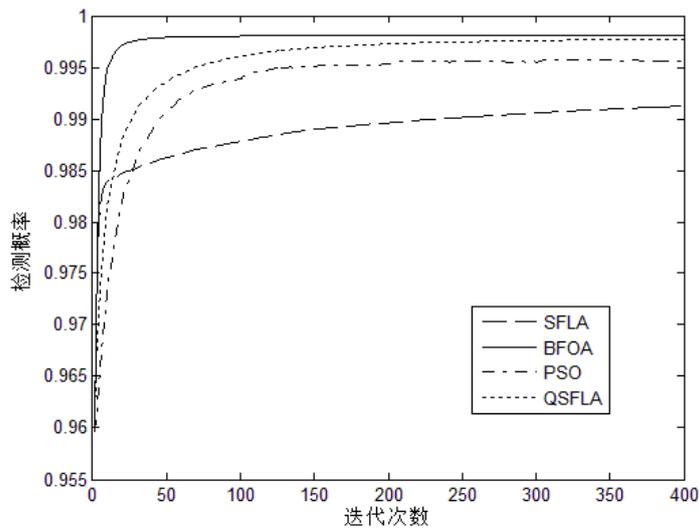


图 3.9 有 30 个本地认知用户时的检测概率仿真结果

本文还对另外一种低维情况进行了仿真，这时认知网络中有 10 个本地认知用户。这种情况中，分别使得 $P_f=0.01$, $P_f=0.05$, $P_f=0.1$ 和 $P_f=0.2$ 。参数为 $\sigma = [2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5, 0.9, 2.7]^T$; $\delta = [1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8, 2.0, 3.1]^T$; $\mathbf{h} = [0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5, 0.2, 0.3]^T$ 。仿真结果列在表 3.1 和表 3.2 之中。从中可以看出：在所有 4 种算法之中，量子蛙跳算法无论在收敛精度和收敛速度方面都是最优的。

表 3.1 本地认知用户为 10 个时的目标函数仿真结果

算法	$P_f=0.01$		$P_f=0.05$	
	平均值	标准差	平均值	标准差
BFOA	-0.3448	0.0046	-0.8997	0.0044
SFLA	-0.5110	0.0469	-0.9956	0.0323
PSO	-0.5195	0.0073	-1.0040	0.0082
QSFLA	-0.5282	5.289e-10	-1.0123	1.201e-9

算法	$P_f=0.1$		$P_f=0.2$	
	平均值	标准差	平均值	标准差
BFOA	-1.1792	0.0047	-1.5043	0.0045
SFLA	-1.2575	0.0297	-1.5765	0.0293
PSO	-1.2609	0.0084	-1.5852	0.0091
QSFLA	-1.2735	4.900e-10	-1.5927	6.4902e-10

表 3.2 本地认知用户为 10 个时的检测概率仿真结果

算法	$P_f=0.01$		$P_f=0.05$	
	平均值	标准差	平均值	标准差
BFOA	0.6349	0.0176	0.8159	0.0086
SFLA	0.6953	0.0016	0.8403	0.0011
PSO	0.6983	0.0026	0.8423	0.0020
QSFLA	0.7013	0.0000	0.8443	0.0000

算法	$P_f=0.1$		$P_f=0.2$	
	平均值	标准差	平均值	标准差
BFOA	0.8808	0.0060	0.9337	0.0038
SFLA	0.8957	0.0009	0.9425	0.0005
PSO	0.8963	0.0015	0.9435	0.0010
QSFLA	0.8986	0.0000	0.9444	0.0000

3.4 本章小结

本章主要介绍了创新性设计的量子群智能算法—量子蛙跳算法，并且与传统蛙跳算法、细菌觅食算法和粒子群算法等典型演化算法进行了比较和仿真。由 Benchmark 函数测试的仿真结果可知，量子蛙跳算法相对经典群智能算法，具有更快的收敛速度和更高的收敛精度，从而验证了本文所提出的算法的有效性和优越性。本文还针对认知无线电频谱感知问题深入研究，充分利用量子蛙跳算法，并产生了基于上述量子蛙跳算法的新型协作频谱感知算法。并且将其与经典的解决认知无线电频谱感知的算法做比较，通过仿真试验的结果说明了量子蛙跳算法应用于工程问题时也具有较大的优势和发展潜力。

第4章 基于量子细菌觅食算法的认知无线电频谱感知

在第 3 章中，本文成功地将量子计算与混合蛙跳算法结合，构造出量子蛙跳算法，并且经过仿真试验验证了算法的优越性。与混合蛙跳算法一样，细菌觅食算法也存在着收敛精度低和收敛速度慢的缺陷。因此，本章同样将量子计算的原理引入到细菌觅食的趋向性操作之中，设计出全新的量子细菌觅食算法。通过对测试函数的测试，可以验证所提出的算法的有效性和通用性。同时，将量子细菌觅食算法应用于解决认知协作感知问题，并进行计算机仿真。根据仿真结果不难看出，相对于传统频谱感知方法，本文提出的基于量子细菌觅食算法的所述频谱感知算法优势非常明显。

4.1 细菌觅食算法介绍

随着群体智能算法的蓬勃发展，Passino 于 2002 年提出了模拟人类大肠杆菌觅食行为的细菌觅食优化算法(BFOA)^[53-54]。细菌觅食算法以其直观和易理解性得到国内外学者的广泛关注。细菌觅食算法主要实现机理是通过趋向性操作、复制操作以及迁徙操作实现迭代计算^[55]。

假设想寻求函数 $J(\Theta_i)$ 的最小值(其中 $\Theta_i \in R_M$)，本文使用细菌觅食算法来解决此优化问题，进而同时说明细菌觅食算法的基本原理。

(1) 趋向性操作

细菌种群总数用 S 来表示，各细菌位置用 $\Theta_i = [\Theta_{i1}, \Theta_{i2}, \dots, \Theta_{iM}]$ 来表征，上式中 i 分别取值 $1, 2, \dots, S$ 。令 $J^i(j, k, l)$ 表示细菌 i 的适应度值，而括号中的参数 j, k, l 分别表示第 j 次趋向性操作，第 k 次复制操作以及第 l 次迁徙操作。其中，对于第 i 个细菌的 d 维分量经过的第 j 次趋向性操作表示如下：

$$\Theta_{id}^{j+1} = \Theta_{id}^j + c(i) \frac{\Delta_d(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (4.1)$$

其中 $c(i)$ 表示细菌 i 每次游弋的单位步长， $\Delta(i) = [\Delta_1(i), \dots, \Delta_M(i)]^T$ ，其中每个元素 $\Delta_d(i)$ 是 $[-1, 1]$ 之间的随机数， $d = 1, 2, \dots, M$ 。经过上述趋向性操作，细菌 i 就顺着旋转方向行走用 $c(i)$ 表示长度的距离。如果移动后的细菌位置 Θ_i^{j+1} 对应的适应度 $J^i(j+1, k, l)$ 优于 $J^i(j, k, l)$ ，则在该方向上继续前进 $c(i)$ 距离，直到达到事先设定的

华中科技大学硕士学位论文

最大游弋次数 N_s ；亦或如果 $J^i(j+1,k,l)$ 不优于 $J^i(j,k,l)$ ，结束本次游弋操作，细菌随机旋转寻找下一个移动方向。

(2) 复制操作

细菌完成 N_c 次趋向操作，实现一次复制。复制操作过程中，设置 $S_r = \frac{S}{2}$ ，且用下式代表第 i 个细菌的健康值 $J_{health}^i = \sum_{j=1}^{N_c+1} J^i(j,k,l)$ 。将细菌种群中的所有细菌按照健康值的降序排列。具有较大健康值的 S_r 个细菌消亡，具有较小健康值的 S_r 个细菌分裂（该过程意味着新复制的子代细菌代替消亡的细菌位于与它们父代细菌相同的位置）。

(3) 迁徙操作

细菌完成 N_{re} 次复制操作，实现一次迁徙。对于每一细菌，以概率 P_{ed} 进行变异。如果对某一细菌进行变异，在定义域内随机产生一个新的细菌代替变异细菌。

当所有细菌的迁徙 N_{ed} 次后，那么该细菌种群进化完成一次。

总结上述过程，基于量子细菌觅食算法的优化问题的由下述步骤表明：

Step 1: 初始化参数 $S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, c(i)$ 和 Θ_i ，其中 $i=1,2,\dots,S$ 。在定义域内随机产生初始细菌位置。

Step 2: 迁徙操作循环： $l = l+1$

Step 3: 复制操作循环： $k = k+1$

Step 4: 趋向性操作循环： $j = j+1$

[a] 对于每个细菌，按照如下方式进行一次趋向性操作。

[b] 计算适应度函数 $J^i(j,k,l)$ 。

[c] 令 $J_{last}^i = J^i(j,k,l)$ 。

[d] 旋转：产生由公式 $\Delta(i) \in R^M$ 表征的随机向量，该公式中 $\Delta_d(i)$ 是取值-1到1范围内的随机数， $d=1,2,\dots,M$ 。

[e] 移动：根据下述公式，细菌 i 的 d 维分量完成第 j 次趋向操作：

$$\Theta_{id}^{j+1} = \Theta_{id}^j + c(i) \frac{\Delta_d(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

其中， $d=1,2,\dots,M$ ，这样细菌 i 便在该旋转方向上移动 $c(i)$ 距离。

[f] 计算 $J^i(j+1,k,l)$ 。

[g] 游弋：

- 1) 令 $m=0$
- 2) 当 $m < N_s$ 时
 - 令 $m=m+1$
 - 如果 $J^i(j+1, k, l) < J_{last}^i$, 保存 $J_{last}^i = J^i(j+1, k, l)$, 然后按照如下方式进行

$$\Theta_{id}^{j+1} = \Theta_{id}^{j+1} + c(i) \frac{\Delta_d(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

其中, $d=1, 2, \dots, M$ 。按照[f]步骤, 用 Θ_i^{j+1} 计算 $J^i(j+1, k, l)$ 。

- 否则, 如果 $m = N_s$, 结束while循环。

[h]如果上述 $i \neq S$, 则调回步骤[b], 对下一个细菌进行处理。

Step 5: 如果上述 $j < N_c$, 则跳转回 Step 4 继续完成趋向操作。

Step 6: 复制操作:

- [a] 对于 k 和 l 值给出时, 用公式 $J_{health}^i = \sum_{j=1}^{N_c+1} J^i(j, k, l)$ 表征细菌 i 的健康情况,

细菌 i 从取值至 S 。同时, 根据降序排列所有细菌。

- [b] 具有较低健康值的 S_r 个细菌消亡, 保留的 S_r 个细菌分裂 (该过程意味着新复制的子代细菌代替消亡的细菌位于与它们父代细菌相同的位置)。

Step 7: 如果 $k < N_{re}$, 进入 Step 3。继续实现复制下一代操作, 与此同时, 着手新的趋向性操作。

Step 8: 迁徙操作: 对于每一细菌, 以概率 P_{ed} 进行变异。如果对某一细菌进行变异, 在定义域内随机产生一个新的细菌。如果 $l < N_{ed}$, 进入 Step 2; 否则终止算法。

4.2 基于量子细菌觅食算法的认知无线电频谱感知

研究者发细菌觅食算法同样具有一些典型智能算法的全局收敛性差的缺陷。趋向性操作是细菌觅食算法的最为核心的操作^[56], 因而本文将量子计算引入到趋向性操作之中, 设计出全新的量子机制的趋向性操作。本文还通过仿真, 进一步验证了所提出的量子细菌觅食算法性能优异。同时将量子细菌觅食算法应用于线性协作感知模型而得到基于量子细菌觅食算法的协作感知方法, 通过对该方法的测试, 进一步验证了在频谱感知问题背景下该算法同样具有更加优异的全局收敛性和更快的收

敛速度。

4.2.1 量子细菌觅食算法介绍

细菌觅食算法在科学和工程领域已经得到了广泛的应用，为了提高其性能，本文设计了全新的量子细菌觅食算法(QBFOA)。趋向性操作是细菌觅食算法的核心操作，在QBFOA中，用量子旋转门来完成趋向性操作。将趋向性操作构建成量子化的随机巡游行为，以便寻求最优解。

在QBFOA中，量子细菌的位置表示为 $\bar{\theta}_i = [\bar{\theta}_{i1}, \bar{\theta}_{i2}, \dots, \bar{\theta}_{iM}]$ ，其中 $\bar{\theta}_{id} \in [l_d, u_d]$ ， $d=1, 2, \dots, M$ 。 l_d 为第 d 维变量的下限， u_d 为第 d 维变量的上限。假定要寻求函数 $J(\bar{\theta}_i)$ 的最小值(其中 $\bar{\theta}_i \in R_M$)，适应度函数 $J(\bar{\theta}_i)$ 代表外界环境对细菌各种影响的综合度量^[52-53]。根据文献[47]的量子进化的简化方法，第 i 个量子细菌的量子位置可以表示为 $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iM}]$ ，第 i 个量子细菌的第 d 量子位在第 j 次趋向性操作时按照如下方式进行

$$\theta_{id}^{j+1} = |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}| \quad (4.2)$$

量子位置 $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iM}]$ 的映射位置由公式 $\bar{\theta}_i = [\bar{\theta}_{i1}, \bar{\theta}_{i2}, \dots, \bar{\theta}_{iM}]$ 表示，同时，量子位置的每个元素即量子位取值范围为 $0 \leq \theta_{id} \leq 1$ ， $d=1, 2, \dots, M$ 。用 $J(\bar{\theta}_i)$ 表示量子细菌 i 的适应度值。

在一个 M 维的优化问题中，在量子细菌觅食算法的开始阶段，在定义区间内随机产生初始量子细菌，初始量子细菌总数为 S 。 $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iM}]$ 是量子细菌 i 的量子位置，当寻找到最优适应度值时，保存此时量子细菌的量子位置 $\theta_{m_i n} = [\theta_{m_i n 1}, \theta_{m_i n 2}, \dots, \theta_{m_i n M}]$ 。量子位置的更新依据下述趋向性操作实现：

$$\phi_{id}^{j+1} = r_1 \cdot (\theta_{id}^j - \theta_{\min d}^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j) \quad (4.3)$$

$$\theta_{id}^{j+1} = |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}| \quad (4.4)$$

上式中， r_1 和 r_2 分别表示在0-1取值区间内的均匀随机数，而 r 则是在-1-1取值区间内的均匀随机数。 $\text{sign}(r)$ 表示函数，其值在 $\{-1, 0, 1\}$ 之中。 β 是权重因子，所有 S 个量子细菌的量子位置的平均值用 z 表示如下：

$$z = [z_1, z_2, \dots, z_M] \quad (4.5)$$

基于上述的分析与介绍，量子细菌觅食算法的流程按照如下步骤：

Step 1: 初始化参数 S , N_c , N_s , N_{re} , N_{ed} , P_{ed} 和 θ_i ，其中 $i=1, 2, \dots, S$ 。初始化

的量子位位置在量子域内随机产生。

Step 2: 迁徙操作循环: $l = l+1$

Step 3: 复制操作循环: $k = k+1$

Step 4: 趋向性操作循环: $j = j+1$

[a] 保存当前量子细菌的用 θ_i 表示的量子位置, 且 i 取值范围是1到S之间的正整数。

[b] 寻找当前的最优适应度值, 保存相关量子细菌的量子位置

$$\theta_{\min}^j = [\theta_{\min 1}^j, \theta_{\min 2}^j, \dots, \theta_{\min M}^j]。$$

[c] 针对每一个量子细菌, 依据下述步骤执行趋向性操作。

[d] 计算适应度函数 $J^i(j, k, l)$ 。

[e] 令 $J_{last}^i = J^i(j, k, l)$ 。

[f] 旋转: 计算全部S个量子细菌的量子位置的平均值 $z^j = [z_1^j, z_2^j, \dots, z_M^j]$ 。

[g] 移动: 根据公式(4.3)和公式(4.4)计算

$$\begin{aligned} \phi_{id}^{j+1} &= r_1 \cdot (\theta_{id}^j - \theta_{\min d}^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j) \\ \theta_{id}^{j+1} &= |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}| \end{aligned}$$

其中 $d = 1, 2, \dots, M$ 。

[h] 用映射位置 $\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 计算适应度值 $J^i(j+1, k, l)$ 。

[i] 游弋:

1) 令 $m=0$

2) 当 $m < N_s$ 时

- 令 $m=m+1$

- 如果 $J^i(j+1, k, l) < J_{last}^i$, 保存 $J_{last}^i = J^i(j+1, k, l)$, $\theta_i^j = \theta_i^{j+1}$, 根据公式(4.3)和公式(4.4)计算

$$\begin{aligned} \phi_{id}^{j+1} &= r_1 \cdot (\theta_{id}^j - \theta_{\min d}^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j) \\ \theta_{id}^{j+1} &= |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}| \end{aligned}$$

其中 $d = 1, \dots, M$ 。按照 [g] 步骤, 用新的映射位置

$\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 计算 $J^i(j+1, k, l)$ 。

- 否则, 如果 $m = N_s$, 结束while循环。

[j]如果 $i \neq S$ ，进入步骤[b]处理下一个量子细菌。

Step 5: 如果 $j < N_c$ ，则跳转回步骤4继续完成趋向性操作。

Step 6: 复制操作:

[a] 在给定 k 和 l 的情况下，对于每一个量子细菌 $i = 1, 2, \dots, S$ ，使得健康值 $J_{health}^i = J^i(N_c, k, l)$ 表示第 i 个量子细菌的健康状况。按照健康值的降序对所有量子细菌进行排序。

[b] 具有较低健康值的 S_r 个量子细菌消亡，保留的 S_r 个量子细菌分裂（该过程意味着新复制的子代量子细菌代替消亡的量子细菌位于与它们父代量子细菌相同的位置）。

Step 7: 如果 $k < N_{re}$ ，进入 Step 3。继续上述复制操作，同时，着手新的趋向性操作。

Step 8: 迁徙操作：对于每一量子细菌，以概率 P_{ed} 进行变异。如果对某一量子细菌进行变异，在量子域内随机产生一个新的量子细菌。如果 $l < N_{ed}$ ，进入 Step 2；否则终止算法。

为了计算与比较的需要，令 ite 表示BFOA和QBFOA算法的当前迭代次数， $ite = N_c \cdot (k-1) + N_c \cdot N_{re} \cdot (l-1) + j$ 。

4.2.2 量子细菌觅食算法的测试

为了验证量子细菌觅食算法的优越性与高效性，选择四个通用测试函数对其进行了测试，并且将测试结果与混合蛙跳算法^[42-43]、粒子群算法^[51]以及细菌觅食算法^[52]进行了比较。所选取的测试函数为 Ackley 函数、Griewank 函数、Sphere 函数和 Rosenbrock 函数，其公式分别为

$$f_1(\bar{\theta}) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{M} \sum_{d=1}^M \bar{\theta}_d^2}) - \exp(\frac{1}{M} \sum_{d=1}^M \cos 2\pi \bar{\theta}_d) \quad (4.6)$$

其中， $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_M]$ 并且 $\bar{\theta}_d \in [-15, 30], d = 1, \dots, M$ 。

$$f_2(\bar{\theta}) = \frac{1}{4000} \sum_{d=1}^M \bar{\theta}_d^2 - \prod_{d=1}^M \cos(\frac{\bar{\theta}_d}{\sqrt{d}}) + 1 \quad (4.7)$$

其中， $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_M]$ 并且 $\bar{\theta}_d \in [-10, 10], d = 1, \dots, M$ 。

$$f_3(\bar{\theta}) = \sum_{d=1}^M \bar{\theta}_d^2 \quad (4.8)$$

华中科技大学硕士学位论文

其中, $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_M]$ 并且 $\bar{\theta}_d \in [-5.12, 5.12], d=1, \dots, M$ 。

$$f_4(\bar{\theta}) = \sum_{d=1}^{M-1} (100(\bar{\theta}_{d+1} - \bar{\theta}_d^2)^2 + (\bar{\theta}_d - 1)^2) \quad (4.9)$$

其中, $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_M]$ 并且 $\bar{\theta}_d \in [-2, 2], d=1, \dots, M$ 。

将上述量子细菌觅食算法中涉及各个参数定义为:

$$S=50, N_c=100, N_s=2, N_{re}=5, N_{ed}=2, P_{ed}=0.1, \beta=1。$$

对于细菌觅食算法, 参数与量子细菌觅食算法保持一致, 另外单位游弋步长设置为 $c(i)=0.01$ 。

混合蛙跳算法的参数设置为: 将青蛙群体总数 p 设置为 50, 将青蛙族群数 m 设置为 10, 同时将每个族群青蛙数 n 设置为 5, 每个族群内的最大迭代次数用 gen 表示, 并设置为 5, 而总的最大进化代数 Gen 表示, 并将其设置为 1000。另外 d_j^{\max} 和 d_j^{\min} 设置为: $d_j^{\max} = 0.5, d_j^{\min} = -0.5$ 。

细菌觅食算法的参数设置为: $S=50, N_c=100, N_s=4, N_{re}=5, N_{ed}=2, P_{ed}=0.1, c(i)=0.01$ 。

粒子群算法选取如下参数: 学习因子 $c_1 = c_2 = 2$, 粒子总数 50, 所有粒子速度变化的范围为定义区间的 10%。

测试函数的 M 维取值为 40, 迭代 1000 次, 同时运行的独立试验的次数为 200。在上述限定中, 得到的仿真收敛曲线图, 参照图 4.1-4.4。通过仿真图的结果可知, 在所有算法中, 量子细菌觅食算法具有最优的收敛速度及收敛性能。

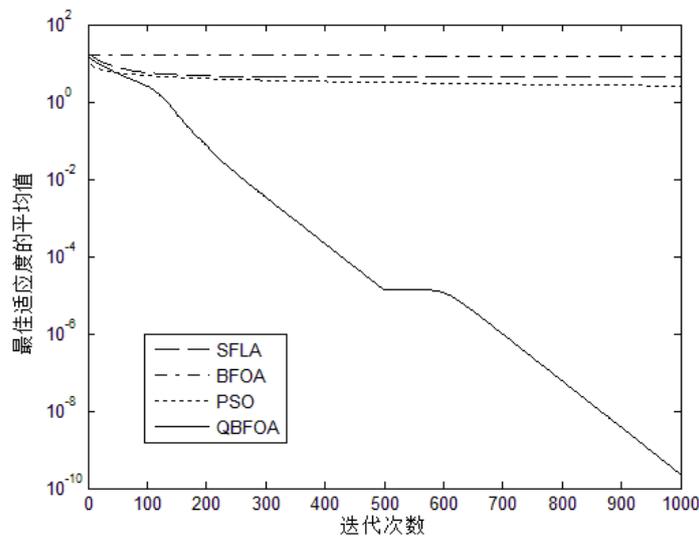


图 4.1 Ackley 函数的收敛曲线

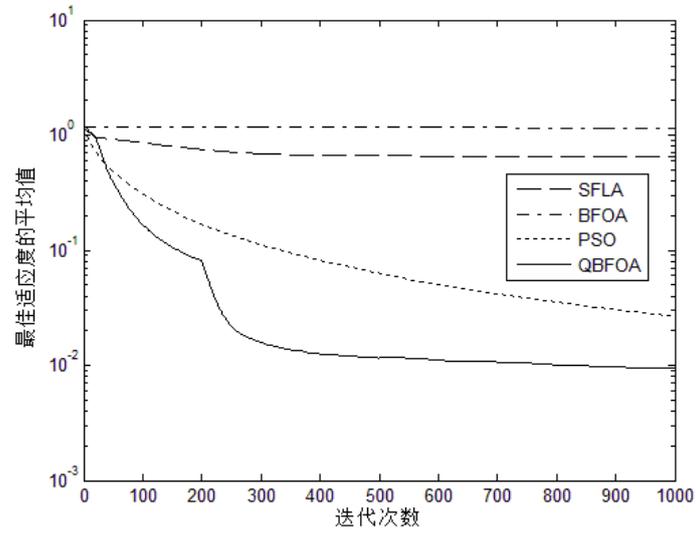


图 4.2 Griewank 函数的收敛曲线

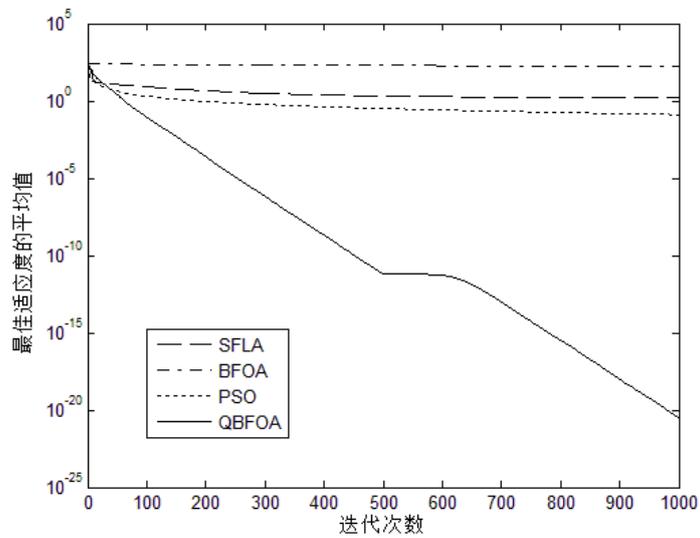


图 4.3 Sphere 函数的收敛曲线

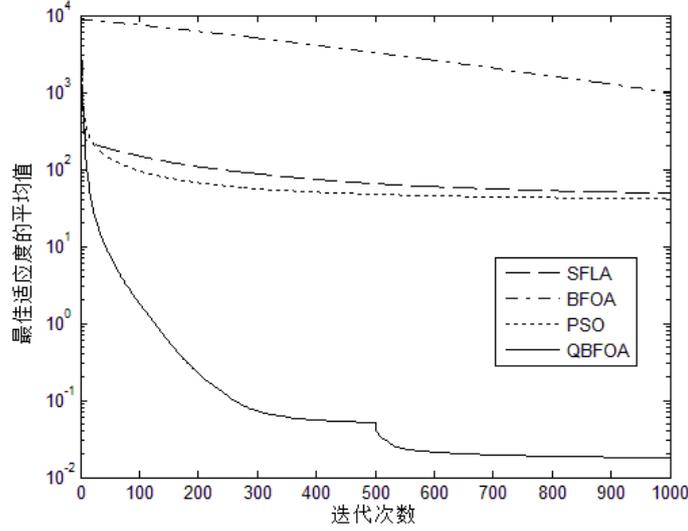


图 4.4 Rosenbrock 函数的收敛曲线

4.2.3 基于量子细菌觅食算法的认知无线电频谱感知

采用量子细菌觅食算法求解频谱感知问题直接优化(2.9)式。将前述协作频谱感知检测概率最大化求解，转变成求解以下函数：

$$\mathbf{w}^\square = \arg \min_{\mathbf{w}} \frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}} \quad (4.10)$$

可以看出该式最优解由权重向量 \mathbf{w}^\square 决定， \mathbf{w}^\square 同样使得在预定虚警概率值 P_f ，能够获得最大的检测概率。如果 \mathbf{w}^\square 是使(4.10)式最小化的最优解，则 $\lambda \mathbf{w}^\square$ 也是使(4.10)式最小化的最优解，明显就会有无穷个最优解。对 \mathbf{w}^\square 进行归一化操作的约束条件为：

$$\min_{\mathbf{w}} f(\mathbf{w}), \text{ s.t. } 0 \leq w_j \leq 1, j = 1, 2, \dots, M, \sum_{j=1}^M w_j = 1 \quad (4.11)$$

由此设计适应度函数为

$$\begin{cases} f(\mathbf{w}) = \frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}}, \\ \text{s.t. } 0 \leq w_j \leq 1, j = 1, 2, \dots, M, \sum_{j=1}^M w_j = 1. \end{cases} \quad (4.12)$$

此优化问题的最优解就是使得(4.10)式最小化的最优权重向量。

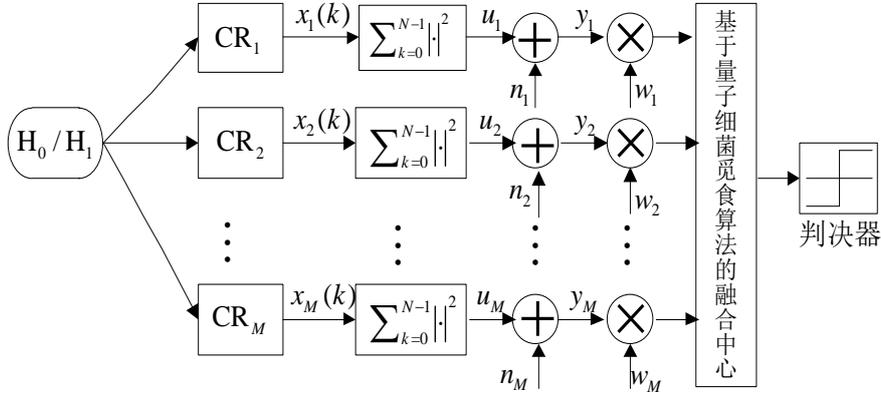


图 4.5 量子细菌觅食算法的协作频谱感知模型示意图

本文采用量子细菌觅食算法来解决前文提及的优化。在基于量子细菌觅食算法的协作感知方法中，本文创新性的设计了基于量子细菌觅食算法的融合中心，其原理框图如图 4.5 所示。对于协作频谱感知模型，量子细菌的位置与量子位置是等价的，在量子域 $[0,1]$ 内随机产生量子位。由此可知，可以将上述协作感知模型中的优化目标函数转化成针对量子细菌觅食算法的适应度函数优化，用公式 $\bar{\theta}_i = [\bar{\theta}_{i1}, \bar{\theta}_{i2}, \dots, \bar{\theta}_{iM}]$ 表示量子细菌 i 的位置，则上述公式与其量子位置 $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iM}]$ 等同，那么适应度函数计算公式如下：

$$\begin{cases} J(\bar{\theta}_i) = \frac{Q^{-1}(P_f) \sqrt{\bar{\theta}_i A(\bar{\theta}_i)^T} - E_s h^T(\bar{\theta}_i)^T}{\sqrt{\bar{\theta}_i B(\bar{\theta}_i)^T}}, \\ \text{s.t. } 0 \leq \bar{\theta}_{id} \leq 1, d = 1, 2, \dots, M, \sum_{d=1}^M \bar{\theta}_{id} = 1. \end{cases} \quad (4.13)$$

根据下述步骤可以计算得到基于量子细菌觅食算法的协作频谱感知权重向量：

Step 1: 初始化参数 $S, N_c, N_s, N_{re}, N_{ed}, P_{ed}$ 和 θ_i ，其中 $i=1,2,\dots,S$ 。初始化的量子位位置在量子域内随机产生，对初始化的量子位置进行归一化处理。

Step 2: 迁徙操作循环： $l = l+1$

Step 3: 复制操作循环： $k = k+1$

Step 4: 趋向性操作循环： $j = j+1$

[a] 保存用 θ_i 表示的量子细菌量子位置， i 为1到 S 之间的正整数。

[b] 寻找当前的最优适应度值，保存相关量子细菌的量子位置

$$\theta_{\min}^j = [\theta_{\min 1}^j, \theta_{\min 2}^j, \dots, \theta_{\min M}^j]。$$

[c] 针对量子细菌，执行下述趋向性操作。

[d] 计算适应度函数 $J^i(j, k, l)$ 。

[e] 令 $J_{last}^i = J^i(j, k, l)$ 。

[f] 旋转：计算全部 S 个量子细菌的量子位置的平均值 $z^j = [z_1^j, z_2^j, \dots, z_M^j]$ 。

[g] 移动：根据公式(4.3)和公式(4.4)计算

$$\begin{aligned}\phi_{id}^{j+1} &= r_1 \cdot (\theta_{id}^j - \theta_{mnd}^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j) \\ \theta_{id}^{j+1} &= |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}| \end{aligned}$$

其中 $d=1, 2, \dots, M$ 。对 $j+1$ 代趋向性操作的量子细菌的量子位置

$\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 进行归一化处理。

[h] 利用 $\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 映射位置计算用 $J^i(j+1, k, l)$ 表示的适应度值。

[i] 游弋：

1) 令 $m=0$

2) 当 $m < N_s$ 时

- 令 $m=m+1$

- 如果 $J^i(j+1, k, l) < J_{last}^i$ ，保存 $J_{last}^i = J^i(j+1, k, l)$ ， $\theta_i^j = \theta_i^{j+1}$ ，根据公式(4.3)和公式(4.4)计算

$$\begin{aligned}\phi_{id}^{j+1} &= r_1 \cdot (\theta_{id}^j - \theta_{mnd}^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j) \\ \theta_{id}^{j+1} &= |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}| \end{aligned}$$

其中 $d=1, 2, \dots, M$ 。对 $j+1$ 代趋向性操作的量子细菌的量子位置

$\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 进行归一化处理。按照[g]步骤，用新的映射位置

$\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 计算 $J^i(j+1, k, l)$ 。

- 否则，如果 $m = N_s$ ，结束while循环。

[j] 如果 $i \neq S$ ，进入步骤[b]处理下一个量子细菌。

Step 5: 在 $j < N_c$ 时，跳转回步骤4 继续完成所述趋向性操作。

Step 6: 复制操作：

[a] 在给定 k 和 l 的情况下，对于每一个量子细菌 $i = 1, 2, \dots, S$ ，使得健康值

$J_{health}^i = J^i(N_c, k, l)$ 表示第 i 个量子细菌的健康状况。按照健康值的降序对

所有量子细菌进行排序。

[b] 具有较低健康值的 S_r 个量子细菌消亡，保留的 S_r 个量子细菌分裂（该过程意味着新复制的子代量子细菌代替消亡的量子细菌位于与它们父代量子细菌相同的位置）。

Step 7: 如果 $k < N_{re}$ ，进入 Step 3。继续实现复制操作，同时着手新的趋向性操作。

Step 8: 迁徙操作：对于每一量子细菌，以概率 P_{ed} 进行变异。如果对某一量子细菌进行变异，在量子域内随机产生一个新的量子细菌。如果 $l < N_{ed}$ ，进入 Step 2；否则终止算法。

4.2.4 仿真实现与结果分析

仿真过程中，在协作频谱感知这个工程问题的背景下，将量子细菌觅食算法与传统蛙跳算法、细菌觅食算法和粒子群算法比较，其参数与 4.2.2 节所选取参数相同。假设认知网络中的认知用户数目为 M ，则这些用户中每一个都可互相独立检测各自目标频谱。认知网络中的权重向量 $\mathbf{w} = [w_1, w_2, \dots, w_M]$ 为一个 M 维向量，在所提出的基于量子细菌觅食算法的协作感知模型中，权重向量与量子细菌 i 的维数相等，这也就意味着与认知用户数相同。因此可以实现量子细菌和待优化权重向量二者相互映射。本文将授权用户的信号用 $s(k)$ 表示，且值等于 1，同时假定采样次数 N 为 20 次。

图 4.6 和图 4.7 分别给出了虚警概率 $P_f = 0.1$ ，10 个认知用户条件下，选取不同迭代次数时的最优目标函数值和检测概率的变化曲线。此时进行 200 次独立试验，智能计算算法的终止迭代次数设为 200。本文还设定认知无线电协作频谱感知模型中其他参数：

$$\boldsymbol{\sigma} = [2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5, 0.9, 2.7]^T;$$

$$\boldsymbol{\delta} = [1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8, 2.0, 3.1]^T;$$

$$\mathbf{h} = [0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5, 0.2, 0.3]^T$$

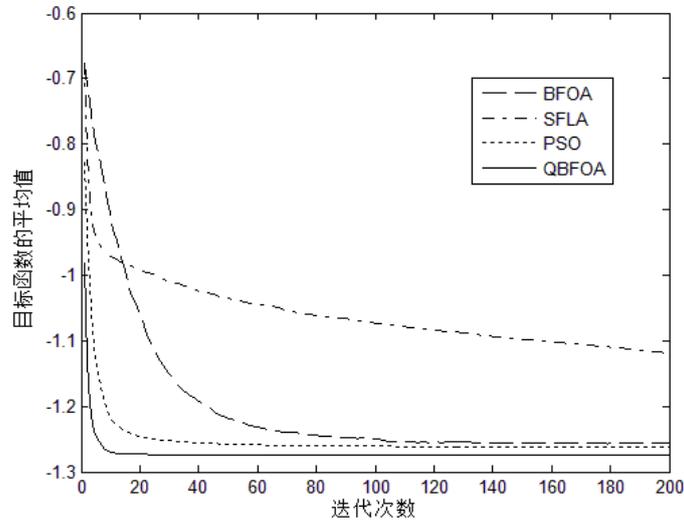


图 4.6 有 10 个本地认知用户时的目标函数仿真结果

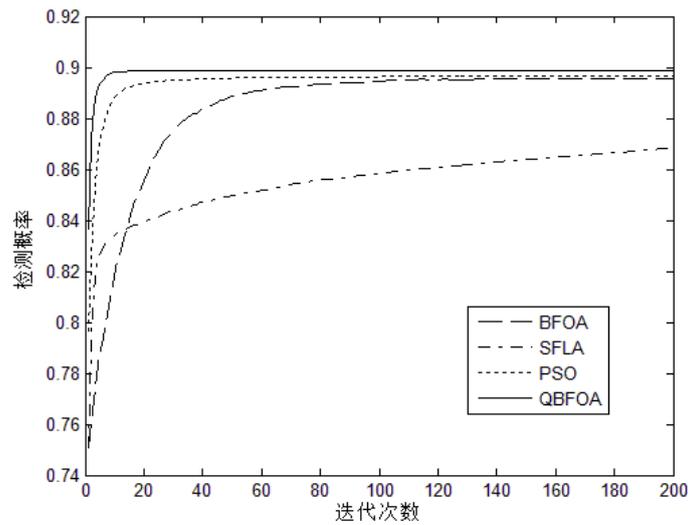


图 4.7 有 10 个本地认知用户时的检测概率仿真结果

图 4.8 和图 4.9 分别给出了虚警概率 $P_f = 0.1$ ，25 个认知用户条件下，当迭代次数取值不同时，最优目标函数值和检测概率的曲线图。进行 200 次独立试验，所有智能计算算法终止迭代次数设为 400。参数如下：

$$\begin{aligned} \sigma &= [2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5, 0.9, 2.7, 2.1, 2.3, 0.7, 2.8, 1.1, 3.6, 2.1, 2.3, 1.9, \\ & 2.2, 2.0, 2.7, 1.1, 3.2, 1.5]^T \\ \delta &= [1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8, 2.0, 3.1, 1.1, 0.6, 2.1, 3.5, 2.5, 0.3, 1.5, 0.7, 2.2, \\ & 3.3, 1.4, 0.6, 2.2, 3.6, 2.2]^T \end{aligned}$$

$h = [0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5, 0.2, 0.3, 0.3, 0.4, 0.4, 0.5, 0.3, 0.4, 0.5, 0.6, 0.1, 0.5, 0.3, 0.7, 0.6, 0.5, 0.2]^T$

其余参数与上一仿真实验相同。

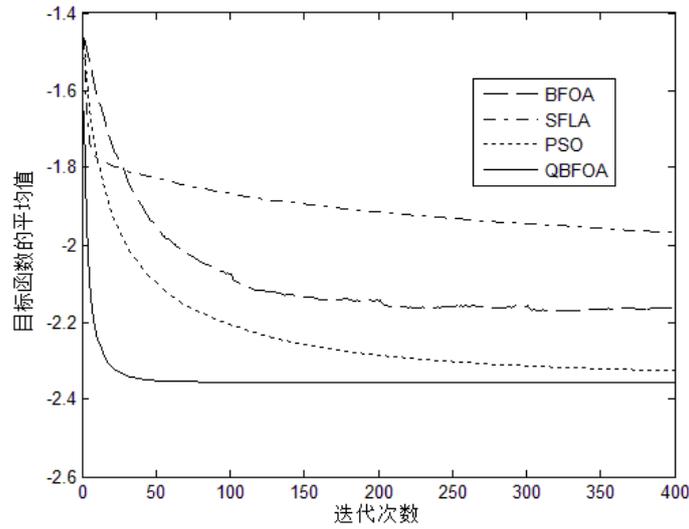


图 4.8 有 25 个本地认知用户时的目标函数仿真结果

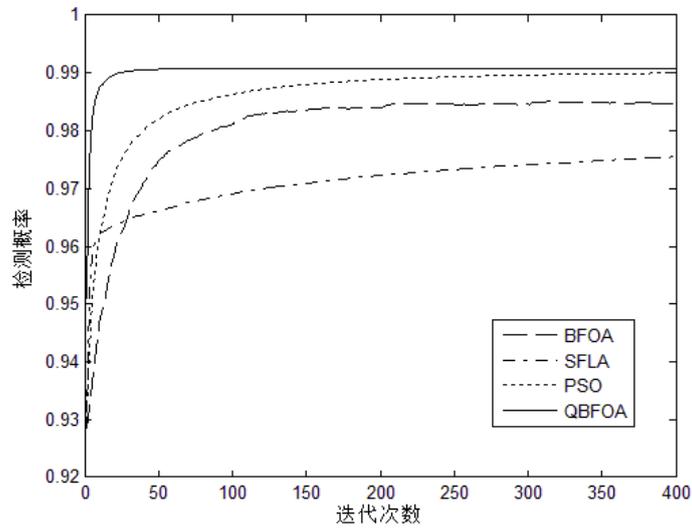


图 4.9 有 25 个本地认知用户时的检测概率仿真结果

同时，本文还对另外一种低维情况进行了仿真，这时认知网络中只有 8 个本地认知用户。这种情况中，分别使得 $P_f=0.05$, $P_f=0.1$, $P_f=0.15$ 和 $P_f=0.2$ 。参数 $\sigma = [2.0, 2.5, 0.9, 2.7, 1.3, 3.3, 2.0, 2.5]^T$; $\delta = [1.3, 0.8, 2.0, 3.8, 2.3, 0.4, 1.3, 0.8]^T$; $h = [0.4, 0.5, 0.7, 0.3, 0.4, 0.3, 0.6, 0.5]^T$ 。仿真结果列在表 4.1 和表 4.2 之中。从中可以看出：

在所有 4 种算法之中，量子细菌觅食算法无论在收敛精度和收敛速度方面都是最优的。

表 4.1 本地认知用户为 8 个时的目标函数仿真结果

算法	$P_f=0.05$		$P_f=0.1$	
	平均值	标准差	平均值	标准差
BFOA	-0.9240	0.0026	-1.1813	0.0025
SFLA	-0.8393	0.0355	-1.1167	0.0260
PSO	-0.9298	0.0094	-1.1879	0.0070
QBFOA	-0.9319	0.0000	-1.1891	0.0000

算法	$P_f=0.15$		$P_f=0.2$	
	平均值	标准差	平均值	标准差
BFOA	-1.3558	0.0026	-1.4958	0.0028
SFLA	-1.2845	0.0331	-1.4769	0.0138
PSO	-1.3635	0.0002	-1.5031	0.0002
QBFOA	-1.3639	0.0000	-1.5034	0.0000

表 4.2 本地认知用户为 8 个时的检测概率仿真结果

算法	$P_f=0.05$		$P_f=0.1$	
	平均值	标准差	平均值	标准差
BFOA	0.8223	0.0007	0.8813	0.0005
SFLA	0.7993	0.0100	0.8679	0.0056
PSO	0.8238	0.0025	0.8826	0.0014
QBFOA	0.8243	0.0000	0.8828	0.0000

算法	$P_f=0.15$		$P_f=0.2$	
	平均值	标准差	平均值	标准差
BFOA	0.9124	0.0004	0.9326	0.0004
SFLA	0.9005	0.0058	0.9302	0.0019
PSO	0.9136	0.0000	0.9336	0.0000
QBFOA	0.9137	0.0000	0.9336	0.0000

4.3 本章小结

借鉴上章的成功经验，本章亦引入量子计算的基本观点，并且创新性地设计了另外一种量子群智能算法—量子细菌觅食算法，并且也将它与传统蛙跳算法、细菌觅食算法和粒子群算法进行了比较和仿真。对基准测试函数的测试表明，量子细菌觅食算法与经典群智能算法相比，可以提高收敛精度和收敛速度，同样验证了细菌觅食算法的通用性。本文还针对认知无线电频谱感知问题，创造性的构建出提出的基于量子细菌的协作感知算法，并且将其与经典的解决认知无线电频谱感知的算法

华中科技大学硕士学位论文

做比较，仿真试验的结果证实了基于量子细菌觅食算法的频谱感知方法的优越性。

第5章 论文总结

无线通信发展迅猛，不断缺少的无线频谱资源逐渐成为限制并制约无线通信持续发展的阻碍。美国联邦委员会的最新调查报告表明，当前世界范围内广泛使用的固定频谱分配政策使得频谱利用率极为低下，很多频段都没有能够利用充分，而其他某些非授权频段却由于业务繁多而导致非常拥挤。认知无线电技术就是针对这种问题应运而生。认知无线电的提出，可以有效地缓解频谱资源紧缺的问题，提供了解决频谱利用率问题的一种可能。因此，认知无线电技术也正成为国内外专家学者关注和研究的热点。

本文主要是针对频谱感知这一认知关键技术进行研究的。频谱感知直接影响到频谱资源的利用率，它已成为认知无线电领域中的非常重要的核心技术。而对于频谱感知的研究主要有下述三个方面：本地感知、协作感知和感知机制优化。本文也与此契合，从这三个角度对频谱感知技术深入探讨和研究，并以上述研究作为基础确定了下一步的继续研究的方向。

本文的主要的研究工作有：

(1) 对现阶段认知无线电频谱感知方法进行分析研究，并将重点集中在认知无线电协作频谱感知模型上。依据混合蛙跳算法、细菌觅食算法以及粒子群算法等经典智能算法的基本原理，研究智能算法在认知无线电频谱感知中的应用并进行仿真。

(2) 针对量子计算，本文提出量子蛙跳算法，同时，本文还通过仿真利用通用测试函数对提出算法进行仿真验证，并证明其优越性。在此基础上，本文还将其用在认知协作频谱感知领域中。仿真结果表明，基于量子蛙跳算法的协作感知方法，相对于粒子群算法等经典人工智能算法具有更优越的性能，在工程应用领域具有很大的发展潜力。

(3) 借鉴量子细菌觅食算法的设计过程，将量子计算的基本原理应用到细菌觅食算法的趋向性操作中，完成对细菌觅食优化算法的量子化。本文还利用通用测试函数进行仿真，并验证了上述改进算法的有效性和优越性。另外，亦将量子细菌觅食算法用于解决认知协作感知问题。仿真实验的结果表明，在该工程背景下细菌觅食算法的优势和效果仍然十分地明显。

由于时间和水平有限，本文中所提出的频谱感知算法还都有待进一步的完善和

优化，具体可以从以下几个方面进行改进：

(1) 参数性能优化。本文提出的量子群智能算法（量子蛙跳算法，量子细菌觅食算法）的相关参数设置尚有改进的余地，算法中的重点参数的设置的研究非常有意义，以达到更加满意的效果。

(2) 协作感知的系统性能与参与协作的用户数量、每个用户门限值以及各个用户的位置分布情况等都有很大的关系，因此，为了实现最佳的检测性能，如何有效的设置协作感知参数是认知无线电研究工作中的重要内容。

(3) 对于现阶段的频谱感知模型，无论是单节点感知还是协作感知，都是采用高斯噪声模型，暂未研究突发噪声的特殊情况。因此，对于其他噪声背景条件下的频谱感知方法研究，也是一项重要内容。

(4) 目前现有的研究主要针对单个感知用户网络参与协作的情况，而多用户协作往往性能更优，因此基于网络层的多感知用户协作也是未来研究的一个方向。

总之，基于量子群智能算法的协作频谱感知方法能够解决经典频谱感知方法所不能解决的一些问题，因而具有较高的研究和应用价值。

致 谢

三年光阴转瞬即逝，在论文即将完成之际，向所有关心和帮助我的人表示衷心的感谢。

首先要特别感谢我的导师魏蛟龙教授，本论文能顺利完成离不开魏老师的悉心指导。魏老师学识渊博、治学严谨，在学术上给予了我很多指导性的意见；平易近人、和蔼可亲的为人风范更会让我终身难忘。

其次，向葛海龙主任表示诚挚的谢意，感谢他对我工作、学习和生活提供的支持和帮助。同时感谢实验室万俊超、赵学阳、付鼎、石建峰等同学，在我求学道路上给予的帮助。

感谢父母妻儿对我求学生涯的支持、鼓励和帮助。

参考文献

- [1] Haykin Simon. Cognitive radio: brain-empowered wireless communications[J]. IEEE Journal on Selected Areas in Communications, 2005, 23: 201-220.
- [2] P. Kolodzy. Spectrum Policy Task Force: Findings and Recommendations. International Symposium on Advanced Radio Technologies (ISART), March 2003.
- [3] W. Tuttlebee . Software Defined Radio : Origins , Drivers and International Perspectives[J]. New York: Wiley, 2002, PE698—703.
- [4] G. Staple and K. Werbach. The end of spectrum scarcity[J]. IEEE Spectrum, 2004, 41(3):48-52.
- [5] M. McHenry. Report on Spectrum Occupancy Measurements. Shared Spectrum Company, Available: http://www.sharedspectrum.com/?section=nsf_summary.
- [6] Yang J. Spatial channel characterization for cognitive radios[D]. Berkeley: University of California, 2004.
- [7] J. Mitola. Cognitive radio for flexible mobile multimedia communications[J]. IEEE International Workshop on Mobile Multimedia Communications, 1999, pp.3-10.
- [8] J. Mitola et al. Cognitive radios: Making software radios more personal[J]. IEEE Personal Communications, vol.6, no.4, August 1999.
- [9] J. Mitola. Cognitive radio: An integrated agent architecture for software defined radio[J]. PhD Dissertation Royal Inst. Technol. (KTH), Stockholm, Sweden, 2000.
- [10] FCC-03-322. Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies[R]. FCC Technological Document.
- [11] 张平,冯志勇. 认知无线网络[M]. 北京:科学出版社, 2010.
- [12] D. Cabric, S. M. Mishra, D. Willkomm, et al. A cognitive radio approach for usage of virtual unlicensed spectrum[A]. 14th IST Mobile and Wireless Communications Summit, 2005.
- [13] Danijela Cabric, Artem Tkachenko, Robert. W., et al. Spectrum sensing measurements of pilot, Energy, and collaborative detection[A]. Military Communications Conference, 2006: 1- 7.

- [14] Danijela Cabric, Robert W. Brodersen. Physical layer design issues unique to cognitive radio systems[C]. PIMRC, 2005: 759- 763.
- [15] 郭彩丽, 冯春燕, 曾志民. 认知无线电网络技术及应用[M]. 北京: 电子工业出版社, 2010.
- [16] T.A.Weiss and F.K.Jondral. Spectrum pooling: an innovative strategy for the enhancement of spectrum efficiency[J]. IEEE Radio Communications, 2004, 42: S8-S14.
- [17] D.Cabrie, I.D.O'Donnell, M.S.Chen and R.W.Brodersen. Spectrum sharing radios[J]. IEEE Circuit and Systems Magazine, Second Quarter 2006, pp, 30-45.
- [18] I.F. Akyildiz, W.Lee, M.C.Vuran and S.Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey[J]. Computer Networks, 2006, 50: 2127-2159.
- [19] H. Zheng and C. Peng. Collaboration and fairness in spectrum assignment in opportunistic spectrum access[C]. in: Proc.40th annual IEEE International Conference on Communications(ICC), 2005: 3132 - 3136.
- [20] Z. Quan, S.G. Cui and Ali H.Sayed. Optimal Linear cooperation for spectrum sensing in cognitive radio networks[J]. IEEE journal of selected topics in signal processing, vol. 2, no. 1,pp.28-40, 2008.
- [21] Z. Quan, S.G. Cui and Ali H. Sayed. An optimal strategy for cooperative spectrum sensing in cognitive radio networks[J]. In Proc. IEEE GLOBECOM, Washington, DC, Nov. 2007.
- [22] Gu Junrong, Jang Sung Jeon and Kim Jae Moug. A proactive dynamic spectrum access method against both erroneous spectrum sensing and asynchronous inter-channel spectrum sensing[J]. KSII Transactions on Internet and Information Systems, 2012, 6(1):361-378.
- [23] D. Cabric, S. M. Mishra, and R. Brodersen. Implementation issues in spectrum sensing for cognitive radios[C]. In Proc. 38th Asilomar Conf. Signals, Systems and Computers, Pacific Grove, CA, Nov. 2004: 772-776.
- [24] 张中兆, 高玉龙. 认知无线电中的频谱感知研究进展[J]. 黑龙江大学工程学报,

2011, Vol 2, No 3.

[25] Sahai A, Hoven N, Tandra R. Some fundamental limits in cognitive radio[C]. Allerton Conf. on Commun., Control and Computing, Virginia, USA:[s.n.], 2004.

[26] Digham F, Alounini M, Simon M. On the energy detection of unknown signals over fading channels[C]. Proc. IEEE ICC 2003, Alaska, USA: [s.n.], 2003, 5:3575-3579.

[27] 胡晓宁, 仵国锋, 胡捍英. 已知噪声不确定性的双门限协作频谱感知[EB/OL]. 2010, <http://www.cnki.net/kcms/detail/11.2127.tp.20110315.1622.016.html>.

[28] Kolodzy P J. Interference temperature: a metric for dynamic spectrum utilization[J]. International Journal of Network Management, 2006, 16:103-113.

[29] Yücek, Tefvik; Arslan, Hüseyin. A survey of spectrum sensing algorithms for cognitive radio applications[J]. IEEE Communications Surveys and Tutorials. 2009, 11(1):116-130.

[30] Chien, Wen-Bin. Yang, Chih-Kai. Huang, Yuan-Hao. Energy-saving cooperative spectrum sensing processor for cognitive radio system[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2011, 58(4):711-723.

[31] Wang, Xiao Yu, Wong, Alexander, Ho, Pin-Han. Dynamically optimized spatiotemporal prioritization for spectrum sensing in cooperative cognitive radio[J]. Wireless Networks, 2010, 16(4): 889-901.

[32] A. Ghasemi and E. Sousa. Collaborative spectrum sensing for opportunistic access in fading environments[C]. In Proc. IEEE Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN), Baltimore, MD, pp.131-136, 2005.

[33] R.L. Chen, J.M. Park, Y.T. Hou and J.H. Reed. Toward secure distributed spectrum sensing in cognitive radio networks[J]. IEEE Communications Magazine, 2008, vol.46, no.4, pp.50-55.

[34] R. S. Blum, S. A. Kassam and H. V. Poor. Distributed detection with multiple sensors: Part II—Advanced topics[J]. In Proc. IEEE, 1997, vol. 85, no.1, pp.64-79.

[35] G. Ghurumuruhan and Y. Li. Agility improvement through cooperative diversity in cognitive radio[C]. In Proc. IEEE GLOBECOM, St.Louis, MO, 2005: 2505-2509.

- [36] J. Culberson. On the futility of blind search: an algorithmic view of no free lunch[J]. *Evolutionary Computation Journal*, 1998, vol. 6, no. 2, pp. 109-128.
- [37] A. Narayanan. Quantum computing for beginners[J]. *Proceedings of the 1999 Congress on Evolutionary Computation*, 1999, IEEE Press, Piscataway, NJ, PP 2231-2238.
- [38] A. Barenco, D. Deutsch, A. Ekert and R. Jozsa. Conditional quantum dynamics and logic gates[J]. 1995, *Phys. Rev. Lett* 74(20), PP.4083-4088.
- [39] Deutsch D. Quantum computational networks[J]. *Proceedings of the Royal Society, London A*, 1989, 425:73-90.
- [40] A. O. Pittenger. *An introduction to Quantum Computing Algorithms*[J]. 2000, Birkhauser.
- [41] H. Bhrman, R. Cleve and A. Wgderson. Quantum vs: classical communication and computation[J]. *Proc. 30th Syrup.on Theory of Comp*, 1998, 63-68.
- [42] M.M. Eusuff and K.E. Lansey. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. *Journal of Water Resources Planning and Management*, 2003, vol. 129, no. 3, pp. 210-225.
- [43] M. Eusuff, K. Lansey, F. Pasha. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization[J]. *Engineering Optimization*, 2006, vol. 38, No. 2, pp. 129-154.
- [44] T. Niknam and E. Azad Farsani. A hybrid self-adaptive particle swarm optimization and modified shuffled frog leaping algorithm for distribution feeder reconfiguration[J]. *Engineering Applications of Artificial Intelligence*, 2010, vol.23, no.8, pp.1340-1349.
- [45] E. Elbeltagi, T. Hegazy, and D. Grierson. Comparison among five evolutionary-based optimization algorithms[J]. *Advanced Engineering Informatics*, 2005, vol. 19, no. 1, pp.43-53.
- [46] H.Y. Gao, Y.Q. Liu, M.Diao. Robust multi-user detection based on quantum bee colony optimization[J]. *International Journal of Innovative Computing and Applications*, 2011, vol. 3, no.3, pp. 160 - 168.
- [47] H.Y. Gao, J.L. Cao and M. Diao. A simple quantum-inspired particle swarm

optimization and its application[J]. Information Technology Journal, 2011, vol.10, no.12, pp.2315-2321.

[48]E. Elbeltagi, T. Hegazy and D. Grierson. A modified shuffled frog leaping optimization algorithm: applications to project management[J]. Structure and Infrastructure Engineering, 2007, vol. 3, no. 1, pp. 53-60.

[49] D. Whitley, R. Beveridge, C. Graves and K. Mathias. Test driving three 1995 genetic algorithms: new test functions and geometric matching[J]. Journal of Heuristics,1995, vol.1, no. 1, pp. 77-104.

[50] H.H. Rosenbrock. An automate method for finding the greatest or least value of a function[J]. The Computer Journal, 1960, vol. 3 no.3, pp. 175-184.

[51]J. Kennedy and R. Eberhart. Particle swarm optimization[J]. In Proc. IEEE Int. Conf. Neural Netw., 1995, pp. 1942-1948.

[52] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control[J]. IEEE Control Systems Magazine, 2002, vol.22, no.3, pp.52-67.

[53] Liu Y. and Passino K.M. Biomimicry of social foraging bacteria for distributed optimization models, principles, and emergent behaviors[J]. Journal of Optimization Theory and Applications, 2002, Vol. 115, No. 3, pp. 603-628.

[54] DeRosier, D. The turn of the screw: The bacterial flagellar motor[J]. Cell, 1998, Vol. 93, pp. 17-20.

[55] Das S., Biswas A., Dasgupta S. and Abraham A. Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications[J]. Foundations of Comput Intel,2009, Vol.3, pp.23-55.

[56] Das, S., Dasgupta, S., et.al. On stability of the chemotactic dynamics in bacterial foraging optimization algorithm[J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2009, Vol.39, No.3, pp. 670-679.