

## 混沌免疫多目标算法求解认知引擎参数优化问题\*

柴争义<sup>1)2)†</sup> 陈亮<sup>1)</sup> 朱思峰<sup>2)</sup>

1) (河南工业大学信息科学与工程学院, 郑州 450001)

2) (西安电子科技大学计算机学院, 西安 710071)

(2011年3月15日收到; 2011年7月4日收到修改稿)

合理的认知引擎参数设置可以提高频谱的使用性能. 通过分析认知无线网络中的认知引擎参数配置, 给出了其数学模型, 并将其转化为一个多目标优化问题, 进而提出一种基于混沌免疫多目标优化的求解方法. 算法使用 Logistic 混沌映射初始化种群, 并在每一代将混沌特性用于最优解集的搜索; 设计了适合此问题的免疫克隆算子和抗体群更新算子, 保证了 Pateto 最优解集分布的多样性和均匀性. 最后, 在多载波环境下对算法进行了仿真实验. 结果表明, 算法可以根据信道条件和用户服务的动态变化, 自适应调整各个子载波的发射功率和调制方式, 可以求出更多满足偏好需求的解, 满足认知引擎参数优化要求.

**关键词:** 混沌, 多目标免疫算法, 认知引擎, 参数配置

**PACS:** 88.80.ht, 64.60.aq

## 1 引言

认知无线网络是一种具有智能的无线通信网络, 其智能主要来自认知引擎<sup>[1]</sup>. 认知引擎的根本目的是根据信道条件的变化和用户需求, 自适应调整其内部通信参数配置, 优化通信系统性能, 从而适应环境和用户需求的变化. 如何利用认知引擎得到最优决策引起了研究者的普遍关注<sup>[2-6]</sup>. 从本质上看, 认知引擎参数决策是一个多目标优化问题, 适合用智能方法求解. 文献[2]提出了认知引擎决策的数学模型, 并通过遗传算法求解; 文献[3]和文献[4]分别通过量子遗传算法和混合遗传算法进行求解; 而文献[5]和文献[6]分别使用粒子群优化和蚁群优化进行实现. 这些算法<sup>[2-6]</sup>均取得了较好的求解效果, 但在求解时认知引擎参数决策中, 均采用线性加权的方法, 实际上是将多目标问题转化为单目标问题求解. 由于难以确定合适的权值, 并且加权法处理多目标优化问题时, 每次只能得到一种权值情况下的最优解并容易漏掉一些最优解<sup>[7]</sup>, 因此, 求解效果还有待提高. 免疫优化算法具有较强的寻优能力, 已经在工程领域得到广泛应用<sup>[8,9]</sup>.

基于此, 本文利用免疫算法寻优能力较强的特性, 提出一种基于免疫多目标优化的认知引擎参数选择和决策方法, 求出算法的 Pateto 最优解集, 提高优化效果. 通过多载波环境对算法进行了仿真. 结果表明, 本算法可以根据信道条件, 给出理想的参数配置, 实现认知引擎决策优化.

## 2 认知引擎参数决策的分析与建模

认知无线网络中, 认知用户(次用户)可以在不影响授权用户(主用户)的情况下, 使用授权用户的空闲频谱<sup>[10-12]</sup>, 并根据信道环境和用户服务需求的变化自适应地调整传输参数(如: 传输功率、调制方式等)以提高空闲频谱的使用性能(如更大化传输速率、更小化传输功率等), 从而达到最佳工作状态.

由此可见, 认知引擎参数决策需要动态地满足多个目标, 如: 必须适应具体的信道条件; 必须满足用户的服务需求; 必须遵守特定频段的频谱特性等, 因此, 其是一个多目标优化问题. 本文根据多载波频谱环境、用户需求以及频谱限制定义出以下 3

\* 国家高技术研究发展计划(863计划)(批准号: 2009AA12Z210)、国家自然科学基金(批准号: 61001202, 61003199 和 61072139)、教育部博士点基金(批准号: 20090203120016 和 20100203120008)、中央高校基本科研业务费专项资金(批准号: JY10000902001)和博士后面上基金(批准号: 20090461283)资助的课题.

† E-mail: super\_chai@tom.com

个目标函数并进行归一化<sup>[2-6]</sup>:

1) 最小化传输功率

$$f_{\text{min-power}} = 1 - \frac{p_l}{L \times P_{\text{max}}},$$

其中,  $p_l$  为子载波  $l$  的传输功率 ( $1 < l < L$ ),  $P_{\text{max}}$  为子载波的最大传输功率,  $L$  为子载波的数目.

2) 最小化误码率 BER(比特错误率)

$$f_{\text{min-BER}} = 1 - \frac{\lg(0.5)}{\lg(p_{\text{be}})},$$

其中,  $p_{\text{be}}$  为  $L$  个子信道的平均误码率. 具体计算公式根据所采用的调制方式不同而不同, 见文献 [2].

3) 最大化数据率(吞吐量)

$$f_{\text{max-throughput}} = \frac{\frac{1}{L} \sum_{l=1}^L \log_2 M_l - \log_2 M_{\text{min}}}{\log_2 M_{\text{max}} - \log_2 M_{\text{min}}},$$

其中,  $L$  为子载波的数目,  $M_l$  为第  $l$  个子载波对应的调制进制数,  $M_{\text{max}}$  为最大调制进制数,  $M_{\text{min}}$  为最小调制进制数. 为了与前两个目标函数表述一致, 将其转换为求最小值问题:

$$f'_{\text{max-throughput}} = \frac{1}{f_{\text{max-throughput}}}.$$

因此, 本文所要优化的目标模型为

$$\min y = (f_{\text{min-power}}, f_{\text{min-BER}}, f'_{\text{max-throughput}}). \quad (1)$$

从上面的优化目标来看, 它们之间相互制约, 比如: 同时实现最小化传输功率和最小化 BER 对传输功率的需求上就存在冲突, 即对单个目标的优化往往导致其他目标性能的恶化. 可见, 此问题是一个多目标优化问题. 本文将问题转化为: 调整各个子载波的发射功率和调制方式, 寻求多目标优化的 Pateto 最优解集(非支配解集), 进而根据用户需求, 选择最满意解, 并通知认知引擎决策进行参数调整, 优化系统性能.

### 3 算法关键技术与具体实现

#### 3.1 关键技术

##### 3.1.1 编码方式

由于决策引擎主要是对参数进行调整, 本文使用二进制对每个子载波的调制方式和发射功率进行编码. 调制方式包括 BPSK, QPSK, 16QAM 和 64QAM 四种, 发射功率共有 64 种可能取值, 范围设置为 0—25.2 dBm, 间隔为 0.4 dBm<sup>[2-6]</sup>. 假设用  $c_1$  表示对四种调制方式的编码, 则需要两位二进制进行编码, 取值为 0, 1, 2, 3, 依次对应 BPSK,

QPSK, 16QAM, 64QAM; 用  $c_2$  表示对发射功率的编码, 由于有 64 种可能取值, 故编码位数为 6, 编码与发射功率的大小依次对应. 因此, 抗体长度由  $c_1$  和  $c_2$  的编码串联而成, 共 8 位. 例如, 调制方式为 16QAM, 发射功率为 24.4 dBm, 则对应的抗体编码为 10111100.

##### 3.1.2 亲和度函数

由于本文的目的是要得到满足优化目标所需的参数配置, 因此, 直接将 (1) 式所示目标函数作为衡量个体性能的亲合度函数.

#### 3.2 求解本问题的多目标免疫优化算法

本算法由初始化、免疫克隆、克隆变异、克隆选择、抗体群更新等操作组成, 基本步骤如下.

##### 步骤 1 初始化

给定抗体种群规模  $N$ , 克隆系数  $q$ , 最大迭代次数  $g_{\text{max}}$ ; 抗体编码长度  $c$ ; 初始化迭代次数  $it = 0$ .

为了确保抗体产生的随机性并遍历所有抗体空间, 本文初始抗体种群的产生使用 Logistic 映射:  $x_{n+1} = \mu x_n(1 - x_n)$ , 其中,  $n = 1, 2, \dots, N$ ,  $\mu = 4$  (此时系统处于完全混沌状态, 其状态空间为 (0,1)<sup>[8,13]</sup>). 随机产生第一个抗体  $x_1$  ( $c$  个具有微小差异的初值), 然后按照 Logistic 映射依次生成规模为  $N$  的抗体种群, 记为:  $A(it) = \{a_1(it), a_2(it), \dots, a_N(it)\}$ .

##### 步骤 2 对抗体群 $A(it)$ 进行克隆操作

$$A'(it) = R_C^p(A(it))$$

本算法中采用的是整体克隆的方式, 克隆系数为  $q$ , 表示如下:

$$\begin{aligned} A'(it) &= R_C^p(A(it)) \\ &= R_C^p(a_1(it) + \dots + R_C^p(a_N(it))) \\ &= \{a_1^1(it), a_1^2(it), \dots, a_1^q(it)\} + \dots \\ &\quad + \{a_N^1(it), a_N^2(it), \dots, a_N^q(it)\}. \end{aligned}$$

##### 步骤 3 对抗体群 $A'(it)$ 进行变异

$$\begin{aligned} A''(it) &= R_m^c(A'(it)) \\ &= R_m^c(\{a_1^1(it), a_1^2(it), \dots, a_1^q(it)\} + \dots \\ &\quad + R_m^c\{a_N^1(it), a_N^2(it), \dots, a_N^q(it)\}) \\ &= \{a_1^1(it), a_1^2(it), \dots, a_1^q(it)\} + \dots \\ &\quad + \{a_N^1(it), a_N^2(it), \dots, a_N^q(it)\}, \end{aligned}$$

其中,

$$R_m^c(a_n^t(it)) = a_n^t(it) (n = 1, 2, \dots, N,$$

$$t=1, 2, \dots, q).$$

本算法中的变异  $R_m^c$  采用超变异<sup>[14]</sup>, 即对某些基因位依照概率  $p_m$  取反.

**步骤 4** 克隆选择  $A'''(it) = R_C^S(A''(it))$

克隆选择操作选出非支配抗体, 针对多目标优化解集的特点, 本文设计的克隆选择操作具体如下.

**步骤 4-1** 对抗体群  $A''(it)$  中的每一个抗体, 计算其对应的  $m$  个目标函数值 (本文中  $m = 3$ ), 得到  $N(it)$  个  $m$  维的矢量组成的目标值矩阵  $F(A''(it))$ .

**步骤 4-2** 将抗体群  $A''(it)$  划分为两个抗体群: 支配抗体群  $A_{\text{dom}}(it)$  (抗体个数为  $N_{\text{dom}}(it)$ ) 和非支配抗体群  $A_{\text{non}}(it)$  (抗体个数为  $N_{\text{non}}(it)$ ), 并且  $N_{\text{dom}}(it) + N_{\text{non}}(it) = q \cdot N(it)$ .

**步骤 4-3** 选出非支配抗体, 并从中随机选择若干个体 (10%), 以它们的拷贝作为初始解, 进行混沌搜索, 以得到更多非支配解; 克隆选择后得到  $A'''(it) = A_{\text{non}}(it)$ , 更新计算非支配抗体相应的目标函数值矩阵  $F(A'''(it))$ .

个体的混沌搜索过程如下:

$$a'_i = b \times a_i + (1 - b) \times (1 - a_i) \pm v \times \text{logisitic}(i)$$

其中,  $a_i$  为抗体基因位,  $a'_i$  为混沌更新后的抗体基因位,  $b$  为抗体的影响因子, 文中设置取值范围为  $0.1 \leq b \leq 0.4$ ,  $v$  为混沌收缩因子, 设置取值范围为  $0.1 \leq v \leq 0.3$ <sup>[15,16]</sup>. 这样保证了更新后的抗体变量仍位于  $[0, 1]$  之间.

**步骤 5** 抗体群更新操作  $A''''(it) = R_C^{RS}(A'''(it))$

对抗体群  $A'''(it)$  更新得到新的抗体群  $A''''(it)$  和新的目标函数值矩阵  $F(A''''(it))$ .

$$\begin{aligned} A''''(it) &= R_C^{RS}(A'''(it)) \\ &= R_C^{RS}(\{a'_1(it), a'_2(it), \dots, a'_{N_{\text{non}}(it)}(it)\}) \\ &= \{a''_1(it), a''_2(it), \dots, a''_{N_n}(it)\}. \end{aligned}$$

抗体群更新操作过程具体如下.

**步骤 5-1** 给出抗体种群规模  $N_{\text{non}}$ , 期望保留的抗体种群规模  $Nn$ ; 初始化  $i = 1, j = 1$ ; 并且满足  $1 < i < m, 1 < j < N_{\text{non}}$ ; 本文中,  $m = 3$ .

**步骤 5-2** 根据第  $i$  个目标亲和度值将种群按升序排列:  $[F(A'''(it))](:, i) = [f_i(A'''(it))]$ .

其中, 亲和度值的计算如下.

对边界解上的抗体分配一个无穷大的亲和度值, 即  $k_{i1} = Nn, k_{im} = Nn$ ; 对其他抗体, 分配如下亲和度值:

$$k_{ij} = \{ (F(A'''(it)))(j + 1, i) - (F(A'''(it)))(j - 1, i) \}$$

$$\begin{aligned} &\times \{ \delta + \max(F(A'''(it)))(:, i) \\ &- \min(F(A'''(it)))(:, i) \}^{-1} \end{aligned}$$

其中,  $\max(*), \min(*)$  分别表示在所有抗体的亲和度值中, 第  $i$  个目标的最大值和最小值,  $\delta$  是一个很小的正数, 主要是保证任何时候分母都不为 0. 而  $F(A'''(it))(j + 1, i)$  表示抗体  $a'_{j+1}(it)$  的第  $i$  个目标亲和度值.

**步骤 5-3** 如果  $i = m$ , 转步骤 5-4; 否则  $i = i + 1$ , 转步骤 5-2.

**步骤 5-4** 如果  $j = N_{\text{non}}(it)$ , 转步骤 5-5; 否则,  $j = j + 1; i = 1$ , 转步骤 5-2.

**步骤 5-5** 计算第  $j$  个抗体的亲和度值:  $f(k_j) = k_{1j} + k_{2j} + \dots + k_{mj}$ , 即为该抗体的亲和度函数值.

**步骤 5-6** 如果  $N_{\text{non}}(it) = Nn$ , 则停止, 否则, 转步骤 5-7.

**步骤 5-7** 删除亲和度函数值最小的抗体及其对应的目标值矩阵中的值, 得到新的抗体群  $A'''_1(it)$  及目标函数矩阵  $F_1(A'''_1(it))$ ; 令  $N_{\text{non}}(it) = N_{\text{non}}(it) - 1, A'''(it) = A'''_1(it)$ ;  $F(A'''(it)) = F_1(A'''_1(it)), i = 1, j = 1$ , 转步骤 5-2.

**步骤 6** 如果  $it > g_{\text{max}}$ , 则输出抗体群  $A''''(it)$  及其目标函数矩阵  $F(A''''(it))$ ; 否则, 令  $A(it + 1) = A''''(it), F(A(it + 1)) = F(A''''(it)), it = it + 1$ , 转到步骤 2.

### 3.3 算法特点和优势分析

1) 由于非支配抗体的优劣无法比较, 所以, 克隆操作采用整体克隆的方式, 即对每一个非支配抗体采用相同的克隆系数. 克隆实现了空间的扩张, 有利于得到分布较广的 Pateto 前端.

2) 克隆选择操作

本算法中, 克隆选择之前, 先将抗体群中的抗体划分为支配抗体和非支配抗体, 保证了只有非支配抗体才能进入到下一代.

3) 混沌映射用于初始化抗体种群, 增强了抗体的遍历性和多样性; 在 Pateto 最优解集的附近进行混沌搜索, 提高了搜索的广度, 可以产生更多的非支配解, 提高解集分布的均匀性.

4) 抗体群更新操作

按照上面的策略, 非支配抗体的个数可能会非常多, 将使得运算速度变慢. 因此, 本文设计了抗体群更新操作, 即当克隆选择出来的非支配抗体超过

一定数目  $Nn$  的时候, 将 Pateto 前端上比较密集的地方对应的抗体删除, 保证运算速度的同时, 很好地保证了所得解分步的均匀性.

## 4 仿真实验及结果分析

### 4.1 实验环境及参数设置

为了验证本文算法的性能, 在 Windows 环境下, 使用 Matlab-Simulink 中的 IEEE802.11a 模型对算法进行模拟实现, 在多载波系统中对算法性能进行了仿真分析<sup>[2-6,17,18]</sup>. 算法参数设置如下: 子载波数  $L = 32$ , 每个子载波信道可独立选择不同的发射功率和调制方式; 通过给每个子载波分配一个 0—1 之间的随机数表示该载波对应的动态信道衰落因子; 噪声功率初始为 0.01 mW(用于计算  $p_{be}$ )<sup>[6]</sup>; 发射功率共有 64 种可能取值, 范围设置为 0—25.2 dBm, 间隔为 0.4 dBm(即  $P_{\max} = 25.2$ ); 可选调制方式包括 BPSK, QPSK, 16QAM 和 64QAM 四种(即  $M_{\max} = 64$ ,  $M_{\min} = 2$ ). 其他更多的方式只影响 BER 计算公式, 并不影响模拟结果<sup>[2,6]</sup>. 子载波的数目  $L = 32$ , 编码总长度  $c = 256$ .

通过反复实验调整, 免疫多目标优化算法的参数设置如下: 最大进化代数  $g_{\max} = 200$ ; 种群规模  $N = 50$ , 克隆系数  $q = 4$ , 变异概率  $p_m = 0.3$ , 希望保留的非支配抗体种群的规模  $Nn = 50$ ,  $\delta = 0.001$ .

实验验证了在不同无线信道条件下认知引擎的工作性能. 由于 IEEE802.11a 模型使用可编程的无线信道, 能够在 no fading 信道、flat fading 信道、AWGN 信道等之间相互切换, 所以在实验过程中通过手动切换信道就可以仿真无线信道的动态变化<sup>[17,18]</sup>. 用户服务类型和需求分为四种模式: 模式 1 适用于低发射功率情况, 如文件传输; 模式 2 适用于可靠性要求高的应用(要求误码率较低), 如保密通信; 模式 3 适用于高数据速率要求的应用, 如宽带视频通信; 模式 4 则对各个目标函数的偏好相同, 寻求一种平衡. 由于不同服务类型对发射功率、数据率和 BER 的要求不同, 所以把这些要求转换成相应的决策值, 以便于算法从得到的 Pateto 最优解集中选出一个最满意解.

### 4.2 实验步骤

1) 发送端和接收端使用初始传输参数在一个无线信道上传输; 感知频谱环境, 包括信道条件、

用户的服务需求. 如果有变化, 转向步骤 2).

2) 如果信道条件和类型改变, 则根据 (1) 式重新计算亲和度函数, 转向步骤 3); 否则, 从以前计算保存的结果中查找出当前信道条件对应的 Pateto 最优解集, 并转向步骤 4).

3) 执行本文上面设计的混沌免疫多目标优化算法, 求解得到一个 Pateto 最优解集.

4) 根据用户服务需求, 运用层次分析法和模糊评判集成的策略<sup>[19,20]</sup>从 Pateto 最优解集选择一个最令人满意的解, 并通知认知引擎更新其传输参数; 然后转向步骤 1).

### 4.3 实验结果

运行本算法得到 Pateto 最优解集, 经过解码(编码映射)得到认知引擎的参数优化结果. 根据服务模式类型, 选取部分有代表性解. 表 1 列出了在不同信道条件下用户所需的不同服务类型的最满意解.

表 1 不同信道条件下的最满意解

信道类型	服务类型	发射功率 /dBm	数据率 /Mbps	误码率/%
no fading	模式 1	0.11	5.23	0.13
no fading	模式 2	10.0	2.20	0.02
no fading	模式 3	2.35	5.23	0.10
no fading	模式 4	0.32	2.20	0.10
flat fading	模式 1	0.12	5.08	0.14
flat fading	模式 2	11.5	2.03	0.03
flat fading	模式 3	2.27	5.52	0.11
flat fading	模式 4	0.20	5.83	0.10
AWGN	模式 1	0.15	5.23	0.12
AWGN	模式 2	10.2	2.20	0.02
AWGN	模式 3	2.25	6.00	0.10
AWGN	模式 4	0.19	5.62	0.11

从表 1 中的结果可以看出, 算法能够根据信道条件和用户服务类型的变化自适应地优化传输参数. 以 AWGN 信道类型为列.

1) 首先服务类型是模式 1. 算法优化结果为发射功率平均值为 0.15 dBm, 明显小于其他模式, 说明本算法可以很好地实现模式 1 下对最小化发射功率的偏好, 同时, 算法兼顾了最小化误码率和最大化数据率的要求(误码率为 0.12%, 数据率为 5.23 Mbps).

2) 然后服务类型变为模式 2. 算法优化结果为最小化误码率为 0.02%, 小于其他模式的误码率, 说明本算法实现了模式 2 下要求误码率最小的目标要求(调制方式基本上为 BPSK). 同时, 也兼顾了发射功率较小和数据率较大的目标(发射功率为 10.2 dBm, 数据率为 2.20 Mbps).

3) 接下来, 服务类型变为模式 3. 算法优化结果为: 平均数据率为 6 Mbps(调制方式均为 64QAM), 说明本算法达到了模式 3 下对最大化数据率的目标要求.

4) 最后, 服务类型变为模式 4. 从优化结果来看, 算法更倾向于实现发射功率最小化和数据率最

大化. 这是因为: 误码率最小化与发射功率最小化和数据率最大化存在冲突, 同时保证发射功率最小化和数据率最大化的抗体亲和力高于要求误码率最小的抗体亲和力.

图 1 显示了本文算法在 AWGN 信道下的参数优化结果, 进一步验证了算法的有效性.

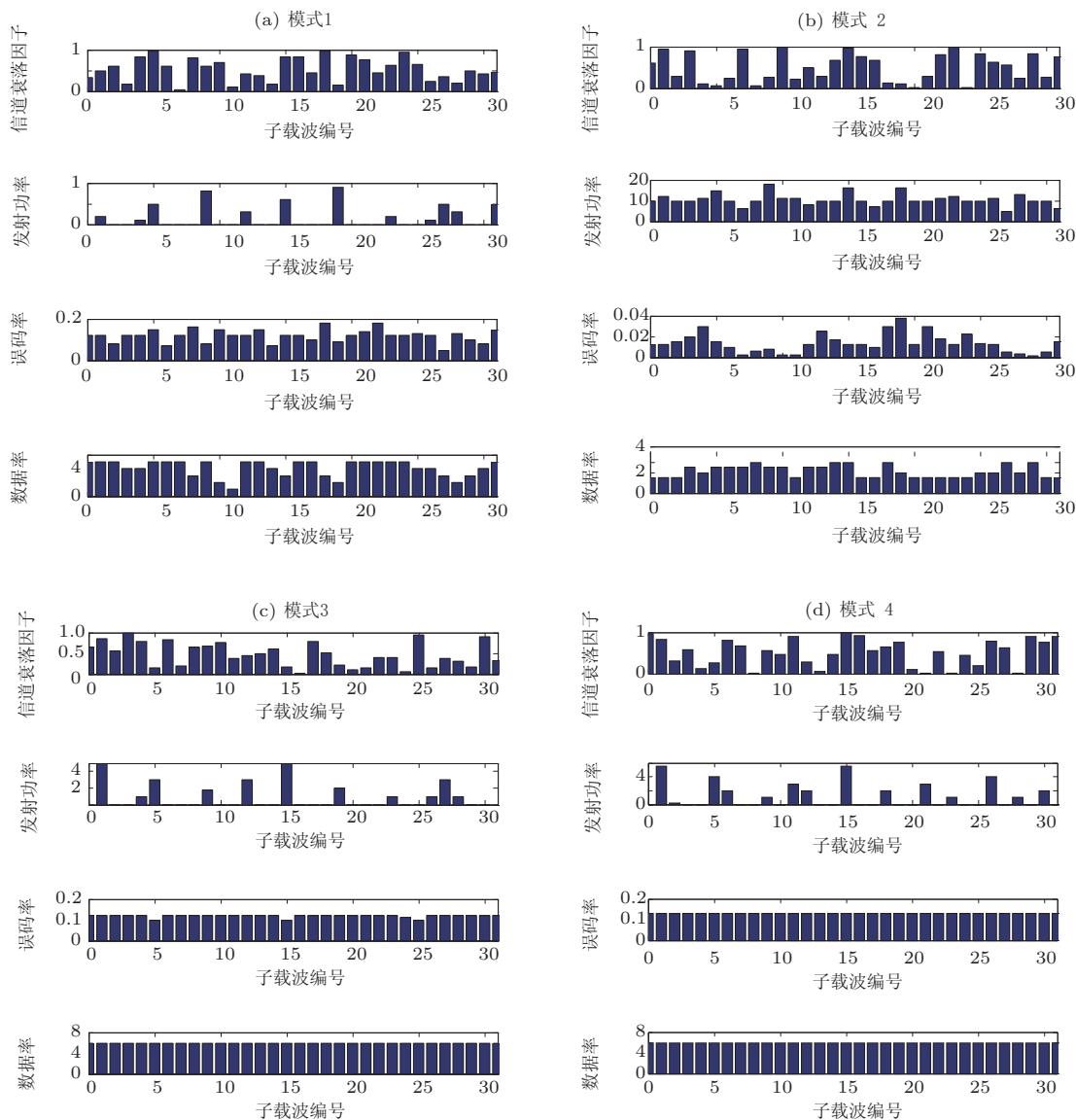


图 1 本文算法调整结果

表 2 相关算法性能比较

服务类型	发射功率/dBm		数据率/Mbps		误码率/%	
	本算法	文献 [6]	本算法	文献 [6]	本算法	文献 [6]
模式 1	0.15	0.18	5.23	5.20	0.13	0.14
模式 2	10.0	11.0	2.20	2.12	0.02	0.03
模式 3	2.35	2.60	5.23	5.18	0.10	0.12
模式 4	0.32	0.41	2.20	2.04	0.10	0.13

#### 4.4 相关算法比较

将本文算法与求解认知引擎的最新代表性文献 [6] 进行比较 (文献 [6] 比文献 [2—5] 有更好的性能). 在 AWGN 信道类型下, 算法各运行 10 次, 取平均值. 对比实验采用与文献 [6] 相同的权重, 对得到的 Pateto 解集进行选优, 通过计算得到相同权重下的最优方案. 结果如表 2 所示.

从表 2 中可以看出, 本算法得到的解更优. 因为本文设计的混沌多目标免疫算法的各种策略, 有力保证了可以得到分布范围较广且均匀的 Pateto 解集, 避免了文献 [6] 中对多目标加权处理可能漏掉的最优解, 有利于得到符合用户决策需求的最优解. 此外, 与已有的算法 [2-6] 相比, 本算法的运行

次数减少. 这是因为: 假设信道条件相同而用户所需的服务类型不同, 此时求出的 Pateto 最优解集是一样的, 所以无需重新运行算法, 只需从中选出一个最满意解即可, 进而减少了算法的运行次数.

#### 5 结论

文章提出了一种混沌免疫多目标优化算法求解认知引擎的参数优化问题, 并在多载波仿真环境下进行实验验证. 结果表明, 本算法可以根据信道条件和用户需求的变化, 自适应调整子载波的发射功率和调整方式, 实现引擎参数的优化, 达到最佳工作状态.

- 
- [1] Haykin S 2008 *IEEE J. Sel. Area Comm.* **23** 201
  - [2] Tim R N, Brett A B, Alexander M 2008 *Wirel. Commun. Mob. Com.* **7** 1129
  - [3] Zhao Z J, Zheng S L, Shang J N, Kong X Z 2007 *Acta Phys. Sin.* **56** 6760 (in Chinese) [赵知劲, 郑仕链, 尚俊娜, 孔宪正 2007 物理学报 **56** 6760]
  - [4] Ayman E H 2009 *The Ninth IEEE Malaysia International Conference on Communications*, Chicago, July 21–23, 2009 p1
  - [5] Zhao Z J, Xu S Y, Zheng S L, Yang X N 2009 *Acta Phys. Sin.* **58** 5118 (in Chinese) [赵知劲, 徐世宇, 郑仕链, 杨小牛 2009 物理学报 **58** 5118]
  - [6] Zhao N, Li S Y, Wu Z L 2011 *Wire. Per. Comm.* **58** 1
  - [7] Yang D D, Jiao L C, Gong M G, Feng J 2010 *Comput. Intell.* **26** 359
  - [8] Chai Z Y, Liu F 2011 *J. Commun.* **31** 591 (in Chinese) [柴争义, 刘芳 2011 通信学报 **31** 591]
  - [9] Chai Z Y, Liu F, Zhu S F 2011 *Acta Phys. Sin.* **60** 068803 (in Chinese) [柴争义, 刘芳, 朱思峰 2011 物理学报 **60** 068803]
  - [10] Zu Y X, Zhou J, Zeng C C 2010 *Chin. Phys. B* **19** 119501
  - [11] Zheng R R, Liu Y A 2010 *Acta Elec. Sin.* **38** 632 (in Chinese) [张然然, 刘元安 2010 电子学报 **38** 632]
  - [12] Ycek T, Arslan H 2009 *IEEE Comm.* **1** 116
  - [13] Zhou J, Zu Y X 2010 *Acta Phys. Sin.* **59** 7508 (in Chinese) [周杰, 俎云霄 2010 物理学报 **59** 7508]
  - [14] Gong M G, Jiao L C, Liu F, Ma W P 2010 *Knowl. Inf. Syst.* **25** 523
  - [15] Lei D M, Yan X P, Wu Z M 2006 *Acta Elec. Sin.* **34** 1142
  - [16] Newman T R, Evans J B 2008 *The 3rd IE-EE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Chicago, March 10–12, 2008 p1
  - [17] Rieser C J 2006 (*Ph. D. Dissertation*) (Virginia: Virginia Tech)
  - [18] Mackenzie A B, Reed J H 2009 *IEEE Comm. Lett.* **97** 660
  - [19] Zhang C Y, Dong X, Wang X J 2010 *J. Mech. Eng.* **46** 156 (in Chinese) [张超勇, 董星, 王晓娟 2010 机械工程学报 **46** 156]
  - [20] Zhou X S, Jiong S, Jian X 2010 *J. S. E. Univ.* **26** 563

# Parameter optimization of cognitive engine based on chaos multi-objective immune algorithm\*

Chai Zheng-Yi<sup>1)2)†</sup>    Chen Liang<sup>1)</sup>    Zhu Si-Feng<sup>2)</sup>

1) (*School of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China*)

2) (*School of Computer Science and Technology, Xidian University, Xi'an 710071, China*)

(Received 15 March 2011; revised manuscript received 4 July 2011)

## Abstract

Reasonable setting of engine parameters can improve the performance of the spectrum use. By analyzing engine parameter adjustment of cognitive wireless network, the mathematical model is given, and then it is converted into a multi-objective optimization problem. A chaos multi-objective immune algorithm is proposed to solve the problem. Logistic mapping is used to initialize population and search for the best solutions in every generation. The operators of cloning and antibodies updating are designed, which ensures that the distribution of Pareto optimal solutions is more diverse and uniform. Finally, the simulation experiments are done to test the algorithm under multi-carrier system. The results show that the algorithm can adjust transmission power and modulation mode according to the change of channel and user demands. More solutions with preferences are obtained, so it meets the demands for parameter optimization of cognitive engine.

**Keywords:** chaos, multi-objective immune algorithm, cognitive engine, parameters adjustments

**PACS:** 88.80.ht, 64.60.aq

---

\* Project supported the National High Technology Research and Development Program of China (Grant No. 2009AA12Z210), the National Natural Science Foundation of China (Grant Nos. 61001202, 61003199, 61072139), the National Research Foundation for the Doctoral Program of Higher Education of China (Grant Nos. 20090203120016 and 20100203120008), the Fundamental Research Funds for the Central Universities (Grant No. JY10000902001), and the Postdoctoral Science Foundation Funded Project of China (Grant No. 20090461283).

† E-mail: super\_chai@tom.com