

基于离散 PSO 算法的 RSC 码识别

吴昭军¹, 张立民², 钟兆根¹, 刘 杰²

(1. 海军航空大学电子信息工程系, 山东烟台 264001; 2. 海军航空大学信息融合所, 山东烟台 264001)

摘 要: 针对目前 RSC 码盲识别算法容错性不好、计算量大的缺点, 提出了基于离散 PSO 算法的 RSC 码识别方法. 首先根据 RSC 码特殊的编码结构, 构建出基于离散 PSO 算法的识别模型, 并推导出了算法的适应度函数以及算法终止的判决门限; 其次, 按照构建出的识别模型, 提出了详细的 RSC 码编码参数的识别步骤和流程框图, 克服了目前 RSC 码识别算法的缺陷. 仿真实验表明: 该算法在寄存器个数较小时, 具有较高的识别性能; 同时在误码率高达 0.05 条件下, 各种寄存器个数下的 RSC 码参数识别率均能达到 90% 以上, 部分甚至达到 100%, 且算法的计算量仅仅与寄存器个数以及码元路数呈线性增加, 与 Walsh-Hadamard 算法相比, 计算量大大减少.

关键词: RSC 码; 离散 PSO 算法; 适应度函数; 终止门限; 识别

中图分类号: TN911.7

文献标识码: A

文章编号: 0372-2112(2018)07-1644-08

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2018.07.015

Blind Recognition of RSC Based on Discrete PSO

WU Zhao-jun¹, ZHANG Li-min², ZHONG Zhao-gen¹, LIU Jie²

(1. Department of Electronics and Information Engineering, Naval Aviation University, Yantai, Shandong 264001, China;

2. Department of Information Fusion, Naval Aviation University, Yantai, Shandong 264001, China)

Abstract: In order to address the defects of poor error tolerance and large amount of calculation in current RSC encoder recognition algorithms, a new algorithm based on discrete PSO is proposed. Firstly, because of the special structure of RSC codes, the recognized model based on the discrete PSO algorithm is built and the fitness of algorithm as well as terminal threshold are deduced. Then, the detailed steps and process to recognize the RSC codes are put forward according to the identification model to overcome the defects in current algorithms. The simulation results show that the algorithm has perfect performance when the number of memories is small. The correct ratio of recognition can reach 90% at different numbers of memories when the rate of bit error is 0.05, some of which can even reach 100%. Besides, the calculation of algorithm increases linearly with the number of memories and branches of codes, which is reduced greatly compared with the Walsh-Hadamard.

Key words: recursive systematic convolutional codes; the discrete particle swarm optimization; the function of fitness; terminal threshold; recognition

1 引言

为对抗恶劣的信道噪声, 信道编码方式被广泛应用于现代数字通信中. 作为信道编码中极具代表性之一的 Turbo 码^[1], 以其接近于香农极限的性能, 被广泛应用于卫星通信、深空探测等领域^[2,3]. 在 Turbo 码中, 递归系统卷积码 (Recursive Systematic Convolutional, RSC) 是主要的分量编码器, 所以 RSC 码的识别是整个 Turbo 码识别的重要内容, 同时 RSC 码的正确识别对于

后续交织器的识别, 同样具有重要意义^[4,5].

针对 RSC 码盲识别的传统算法主要有四种^[6], 即解线性方程组方法、欧几里得算法、快速双合冲算法、Walsh-Hadamard 变换算法. 解线性方程组方法^[7], 利用了二元域中, 码元之间线性约束关系来构建方程组, 但是稍一存在误码, 码元间的线性关系不复存在, 故算法容错性能差; 欧几里得算法^[8]和快速双合冲算法^[9]能够实现卷积码快速盲识别, 但同样是基于无误码或是误码极少的理想情况, 而 Walsh-Hadamard 变化算

法^[10]的容错性能极强,但该算法实质上是一个穷举算法,计算量随着移位寄存器个数以及码元路数而成指数倍增加;文献[11]首次提出了基于 EM 算法的识别方法,该方法容错性能较好,但算法不稳定,极易陷入局部极小值且计算量较大,与文献[11]一样,文献[12]同样具有这样的缺点.此外,还有许多算法基于软判决信息来识别,如文献[13],其容错性能较好但是计算量与 Walsh-Hadamard 变换相当.综合前人所做工作来看,目前 RSC 码识别算法性能离高效、可靠还有一定的差距.

针对上述 RSC 码生成多项式参数识别中存在的问题,本文提出了基于离散粒子群算法(Particle Swarm Optimization, PSO)的盲识别方法.该方法具有较好的容错性,仿真结果表明:在误码率为 10^{-2} 数量级时,不同编码记忆长度下的 RSC 码编码器的识别能够达到 90% 以上,且计算量相对于 Walsh-Hadamard 变换而言,大大减小.

2 RSC 码的基本原理

递归系统卷积码的编码结构存在着线性反馈结构,整个编码器等效于一个无限冲激响应滤波器,当输入序列的码重为 1 时,其输出序列码重将会达到无穷大,而不能像一般的卷积码一样足够的零码元能够使编码寄存器的状态能够达到零状态,正因为 RSC 码具有这样的特点,常常作为 Turbo 码编码器的分量编码器,图 1 所示为 RSC 码的编码结构示意图.

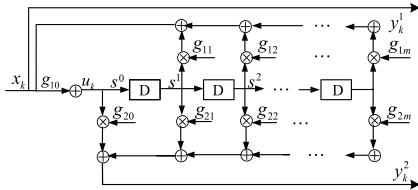


图1 RSC码编码结构示意图

图 1 中 x_k 为 k 时刻的信息码元, y_k^1, y_k^2 分别为 k 时刻第一路与第二路编码生成码元, RSC 码的前向生成 $P(D)$ 多项式系数为 $\{g_{2,0}, g_{2,1}, g_{2,m}\}$, 反馈部分 $Q(D)$ 系数为 $\{g_{1,0}, g_{1,1}, g_{1,m}\}$. 将图 1 以多项式的形式表示^[14], 得到:

$$Y^1(D) = X(D) \quad (1)$$

$$Y^2(D) = X(D) \frac{g_{2,0} + g_{2,1}D + \dots + g_{2,m}D^m}{g_{1,0} + g_{1,1}D + \dots + g_{1,m}D^m} \quad (2)$$

式(1)、式(2)中, $X(D), Y^1(D), Y^2(D)$ 为信息序列与编码序列多项式, 可表示为:

$$X(D) = x_0 + x_1D + \dots + x_kD^k + \dots \quad (3)$$

$$Y^1(D) = y_0^1 + y_1^1D + \dots + y_k^1D^k + \dots \quad (4)$$

$$Y^2(D) = y_0^2 + y_1^2D + \dots + y_k^2D^k + \dots \quad (5)$$

$Q(D)$ 为反馈多项式, $P(D)$ 为前项多项式.

联立式(1)、式(2), 进一步得到:

$$Y^2(D)Q(D) \oplus Y^1(D)P(D) = 0 \quad (6)$$

在一个编码约束长度下, 将式(6)进行展开, 得到:

$$\begin{bmatrix} y_{1,0}^2 & y_{1,1}^2 & \dots & y_{1,m}^2 \\ y_{2,0}^2 & y_{2,1}^2 & \dots & y_{2,m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ y_{t,0}^2 & y_{t,1}^2 & \dots & y_{t,m}^2 \end{bmatrix} \cdot \begin{bmatrix} g_{1,m} \\ g_{1,m-1} \\ \vdots \\ g_{1,0} \end{bmatrix} \oplus \begin{bmatrix} y_{1,0}^1 & y_{1,1}^1 & \dots & y_{1,m}^1 \\ y_{2,0}^1 & y_{2,1}^1 & \dots & y_{2,m}^1 \\ \vdots & \vdots & \ddots & \vdots \\ y_{t,0}^1 & y_{t,1}^1 & \dots & y_{t,m}^1 \end{bmatrix} \cdot \begin{bmatrix} g_{2,m} \\ g_{2,m-1} \\ \vdots \\ g_{2,0} \end{bmatrix} = \mathbf{r}_{t \times 1} \quad (7)$$

式(7)中存在着 t 个编码约束方程, 当不存在误码时, 每一个编码约束方程都能够成立, 即得到的结果向量 $\mathbf{r}_{t \times 1}$ 为一个全零向量, 而当存在误码时, 只要多项式参数 $P(D), Q(D)$ 是正确的, 那么结果向量 $\mathbf{r}_{t \times 1}$ 中 0 元素个数一定是比较多的, 而多项式系数不正确时, 向量 $\mathbf{r}_{t \times 1}$ 中元素 0、1 出现的概率应为 0.5^[15], 元素 0 的个数一定少于当多项式参数正确的情况, 所以结果向量 $\mathbf{r}_{t \times 1}$ 中元素 0 的个数能够表征衡量多项式 $P(D), Q(D)$ 为实际编码多项式的可靠性大小, 即结果向量中元素 0 个数越多, 估计的多项式 $P(D), Q(D)$ 越有可能为实际的编码多项式.

本文假定 RSC 码的码率以及码长已经完成了识别, 针对该问题, 文献[16~18]已经提出基于矩阵分析原理识别算法, 该算法具有较强的工程实用价值, 所以本文将识别的重点放在了 RSC 码生成多项式的识别上, 即在 RSC 码率、码长已知的条件下, 仅仅利用截获的 Y^1, Y^2 两路码元序列实现 RSC 码前项多项式与反馈多项式参数的识别.

3 基于离散 PSO 的识别算法

3.1 识别模型的建立

从第二节中 RSC 码编码的特殊结构来看, 正确的生成多项式系数能够使得式(7)中结果向量中 0 元素的个数最多, 而非正确估计的生成多项式系数将导致结果向量中 0、1 元素随机出现. 这两种情况下, 二者所得到的结果向量 $\mathbf{r}_{t \times 1}$ 中元素 0 个数之间的差距是存在的, 由此给 RSC 码参数的识别提供了可能. 由此, 可以将式(7)中结果向量中 0 元素的个数作为衡量所估计的 RSC 多项式参数接近于实际的编码多项式参数程度的标准, 即结果向量中 0 元素个数越多, 其对应的多项式系数就越有可能为实际的编码多项式. 为了便于处理, 这里将式(7)中结果向量 $\mathbf{r}_{t \times 1}$ 中元素 0 映射为 1, 而 1 映射为 -1, 映射表达式为:

$$\mathbf{V}_{t \times 1} = \text{ones}(t, 1) - 2 \cdot \mathbf{r}_{t \times 1} \quad (8)$$

其中, $\text{ones}(t, 1)$ 为向量元素都为 1 的 $t \times 1$ 的列向量.

在式(8)的基础上,定义出统计变量 $weight(\mathbf{V}_{t \times 1})$,即:

$$weight(\mathbf{V}_{t \times 1}) = \sum_{i=1}^t v_i \quad (9)$$

其中, $v_i \in \{-1, 1\}$, 代表列向量 $\mathbf{V}_{t \times 1}$ 第 i 行上的元素. $weight(\mathbf{V}_{t \times 1})$ 同样能够表征所估计的多项式满足编码方程可能性大小, 即 $weight(\mathbf{V}_{t \times 1})$ 越大, 则对应多项式越有可能为实际的编码多项式, 通过以上分析可知: RSC 码多项式的估计问题, 实质上为二元域中的数值优化问题, 即:

$$\mathbf{g}^* = \underset{\mathbf{g}}{\operatorname{argmax}}(weight(\mathbf{V}_{t \times 1}), \mathbf{g}^*) \quad (10)$$

其中, $\mathbf{g} = \{g_{1,0}, g_{1,1}, \dots, g_{1,m}, g_{2,0}, g_{2,1}, \dots, g_{2,m}\}$, 为估计的多项式系数值, 其系数值为 0 或 1.

由于是在二元域中的参数优化问题, 所以针对该优化问题, 本文采用性能较为优越的离散粒子群算法.

3.2 算法原理

在粒子群算法^[19]中, 每一个个体称之为一个“粒子”, 代表着一个潜在的解. 例如在一个 D 维的目标搜索空间中, 每一个粒子可以看作是空间中的一个点, 设群体的规模为 M , 其中一个粒子 $\mathbf{Z}_i$ 表示为 $(Z_{i,1}, Z_{i,2}, \dots, Z_{i,D})$, $1 \leq i \leq M$ 的位置矢量, 设预先设定的适应性函数为 $p(\mathbf{Z}_i)$, 能够衡量粒子当前位置的优劣程度, $\mathbf{V}_i = (V_{i,1}, V_{i,2}, \dots, V_{i,D})$ 代表粒子 $\mathbf{Z}_i$ 飞行速度, 假定 $\mathbf{P}_i$ 为粒子 $\mathbf{Z}_i$ 迄今为止找到的最优位置, $\mathbf{P}_g$ 为整个粒子群迄今为止所搜索到的最优的位置, 在每次迭代过程中, 粒子根据以下式子进行速度更新以及位置更新^[19].

$$\mathbf{V}_i^{k+1} = w \cdot \mathbf{V}_i^k + c_1 \cdot r_1 (\mathbf{P}_i - \mathbf{Z}_i^k) + c_2 \cdot r_2 (\mathbf{P}_g - \mathbf{Z}_i^k) \quad (11)$$

$$\mathbf{Z}_i^{k+1} = \mathbf{Z}_i^k + \mathbf{V}_i^{k+1} \quad (12)$$

其中, c_1, c_2 为学习因子, 使粒子具有自我总结与向群体中优秀个体学习的能力, r_1, r_2 都为在 $0, 1$ 之间随机取值的参数, 而 w 为惯性系数, 起着权重局部最优和全局最优的作用, 为了算法的更加容易收敛到全局最优点, 一般设定 w 值为一个随着时间线性减小的函数^[19], 即:

$$w(k) = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \cdot k \quad (13)$$

式中, $w_{\max}, w_{\min}$ 分别为初始设定的最大最小惯性系数, $iter_{\max}$ 为最大迭代次数.

上述为基本连续离子群算法的基本原理, 由于本文解决的优化问题是在离散的二元域空间中, 所以在连续粒子群算法的基础上引出离散粒子群算法^[20].

为了将粒子群算法离散化, 算法有当前状态变量决定粒子将被判为 0 或 1 的概率, 既有:

$$P(Z_{i,j}^{k+1} = 1) = f(Z_{i,j}^k, V_{i,j}^k, \mathbf{P}_i, \mathbf{P}_g) \quad (14)$$

其中, $1 \leq j \leq D$. 离散化函数 $f(\cdot)$ 需要在离散的二元空间内使得粒子趋于判决选择为 0 或是 1, 即由粒子速度决定出一个在区间 $[0, 1]$ 之间的概率参数 s_j , 一般使用 sigmoid 函数, 即:

$$s_j = \operatorname{sigmoid}(V_{i,j}^k) = \frac{1}{1 + e^{-V_{i,j}^k}} \quad (15)$$

此时, 为了避免粒子的速度太大, 取定速度最大值为 6, 最小值为 -6, 阈值函数 s_j 的取值范围为 $[0.0025, 0.9975]$. 修改后的离散粒子群算法与连续离子群算法流程类似, 但速度与位置更新公式变换为^[19,20]:

$$\mathbf{V}_i^{k+1} = w \cdot \mathbf{V}_i^k + c_1 \cdot r_1 (\mathbf{P}_i - \mathbf{Z}_i^k) + c_2 \cdot r_2 (\mathbf{P}_g - \mathbf{Z}_i^k) \quad (16)$$

$$Z_{i,j}^k = \begin{cases} 1, & \rho < s_j \\ 0, & \text{其他} \end{cases} \quad (17)$$

其中, ρ 为区间 $[0, 1]$ 之间的随机数, 算法中其他参数与基本粒子算法一致.

离散粒子群算法的终止条件为当 $k = inter_{\max}$ 或是当适应度函数值大于设定的门限 Λ 值.

3.3 适应度函数与终止条件的确定

分析离散 PSO 算法的原理可知, 其中较为重要的部分在于算法中适应度函数以及终止条件的确定. 针对 RSC 的识别, 适应度函数必须要反映出粒子群中个体即潜在解与实际的编码多项式的接近程度, 而终止条件的选择合适能够在一定条件下减小算法的运算量.

同样从 RSC 码的编码结构出发, RSC 编码多项式的识别等效于二元域中函数的优化问题, 所以这里将待优化的目标函数作为粒子群个体的适应度判定标准, 故将式(8)中向量码重作为适应度函数, 即:

$$fitness = weight(\mathbf{V}_{t \times 1}, \mathbf{Z}_i) \quad (18)$$

适应度函数表征了群体中个体粒子位置优劣程度, 适应度函数越大, 其个体粒子位置就越接近于实际的编码多项式参数.

当群体中个体粒子位置参数不为实际的编码多项式参数时, 式(7)中每一个编码约束校验方程式成立概率为 0.5, 从而得到统计量 $weight(\mathbf{V}_{t \times 1})$ 服从二项式分布, 其均值为 0, 方差为 t . 当 $t \rightarrow \infty$ 时, 由中心极限定理, 可知: 该分布服从正态分布, 即得到: $weight(\mathbf{V}_{t \times 1}) \sim N(0, t)$.

由此, 设定漏警概率 α 为一定值, 求取判决门 Λ 限, 即得到:

$$\int_{\Lambda}^{\infty} \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}} dx \quad (19)$$

令 $\alpha = 0.001$, 通过查询正态分布表, 得到门限 Λ 的计算方法为:

$$\Lambda = 3.1\sqrt{t} \quad (20)$$

所以算法的终止的条件为当适应度函数值 $fitness > \Lambda$, 或迭代次数 $k = inter_{\max}$.

3.4 识别算法的实现

由以上适应度函数以及算法终止条件的确定, 可

以得到基于离散 PSO 算法的 RSC 识别方法,得到具体的 RSC 码识别算法步骤如下:

步骤 1 初始化截获数据码元,构建数据矩阵,设定粒子群规模 $M = (2 \cdot 2^{2(m+1)})/3$ (综合实时性与可靠性)、最大迭代次数 $inter_{\max} = 20$ 、粒子个体最大寻优速度 $V_{i,\max}$ 以及最小寻优速度 $V_{i,\min}$ 等算法初始参数;

步骤 2 随机产生并筛选个体码重大于 3,并且种群规模为 M 的粒子群 $Z_i^0 = (Z_{i,1}^0, Z_{i,2}^0, \dots, Z_{i,D}^0)$, $1 \leq i \leq M$,并随机产生个体的初始化速度 $V_i^0 = (V_{i,1}^0, V_{i,2}^0, \dots, V_{i,D}^0)$, $1 \leq i \leq M$;

步骤 3 第 k 次迭代时 ($k = 1, 2, \dots, inter_{\max}$),剔除码重小于 3 的个体粒子,然后计算剩余每个粒子的适应度函数值 $fitness_i^k$,通过比较存储每个个体迭代过程中的最优值位置 $gbest^k$ 以及整个群体的最优值位置 $zbest^k$;

步骤 4 按照式 (16) 与式 (17) 更新粒子个体的速度与位置,若 $V_{i,j}^{k+1} (j = 1, \dots, D) > V_{\max}$ 则 $V_{i,j}^{k+1} (j = 1, \dots, D) = V_{\max}$,若 $V_{i,j}^{k+1} (j = 1, \dots, D) < V_{\min}$,则 $V_{i,j}^{k+1} (j = 1, \dots, D) = V_{\min}$;

步骤 5 重复步骤 3 与步骤 4,直到群体最优粒子所对应的 $fitness > \Lambda$ 或是 $k = inter_{\max}$,算法结束,输出 $zbest$.

上述 RSC 码识别流程虽然针对码率为 $1/2$ 的情况,但是同样适用于码率为 $1/n$ 的情况,此时只需要将第一路系统信息码元作为公共码元,与后续的第 2 路,第 3 路...第 n 路码元结合,构建出 $n - 1$ 个码率为 $1/2$ 的递归系统卷积码,然后应用上述识别算法实现识别,得到每一路的生成多项式系数,进而完成识别。

3.5 计算复杂度分析

分析算法流程,算法的运算量主要来源于对粒子个体的适应度函数值计算,其中涉及到了二元域中矩阵计算,假定截获的码元构建出的数据矩阵行数为 t ,则完成一个粒子个体的适应度计算需要进行 $2t$ 次的向量乘与 1 次向量加。假定 RSC 码码率为 $1/n$,则可得到完成一次参数识别,所需要最大向量乘次数 mul 与向量加 sum 为:

$$mul = 2(n - 1) \cdot m \cdot inter_{\max} \cdot t \quad (21)$$

$$sum = (n - 1) \cdot m \cdot inter_{\max} \quad (22)$$

其中, t 为截获数据所构建的矩阵 Y^1, Y^2 行数,与截获数据长度的关系为:

$$t = \lfloor L/2(m + 1) \rfloor \quad (23)$$

$\lfloor \cdot \rfloor$ 表示向下取整, m 为编码器的寄存器个数。

通过对所提出 RSC 码识别算法计算复杂度的分析可知,该算法的运算量较小,当初始种群规模与迭代次数一定时,算法的计算量仅仅与截获码元长度 L 以及码元路数 n 成线性增加。

4 仿真实验

4.1 算法的可行性验证

仿真设定截获的码元数目为 2000 个,选用编码多项式为 $(37, 25)$,表示方式为八进制。最大迭代次数 $inter_{\max} = 100$ 次, $V_{\max} = 6$, $V_{\min} = -6$,惯性系数 $w_{\max} = 2$, $w_{\min} = 0.5$,学习因子 $c_1 = 2, c_2 = 2.1$,设定二进制对称信道误码率为 0.01。仿真时,以实际工程中最大的编码寄存器个数作为估计值,即 $m' = 6$,由算法的流程,此时选定种群规模 $M = 2^{14} \cdot 2/3$,而仿真的编码寄存器个数为 5,得到每次迭代时的适应度函数值以及判决门限结果如下,此时的门限值 Λ 为 67.53:

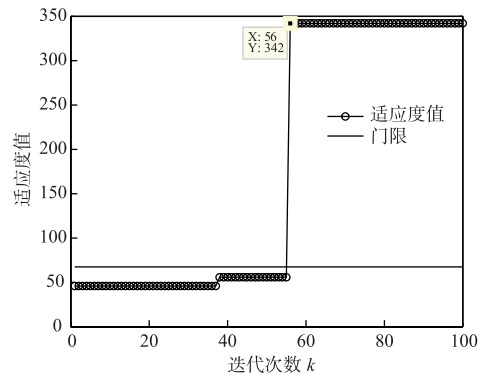


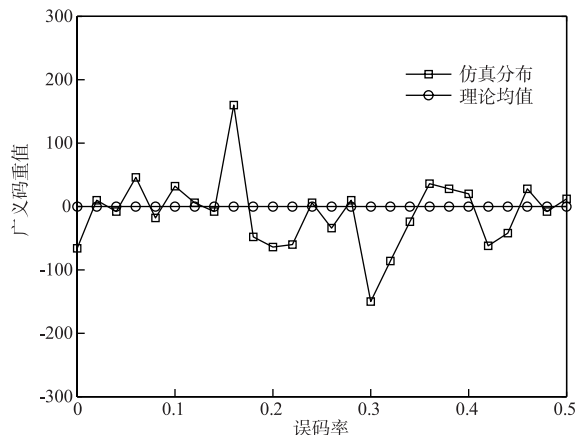
图2 不同迭代次数下的适应度值

在第 56 次迭代时,出现适应度值 $fitness = 342 > 67.53$,此时识别已经完成,得到反馈识别参数为:1,1,0,1,0,1,1,前项识别参数为:1,0,1,1,1,0,1,将该多项式因式分解,消去公约数,得到估计的编码多项式即为设定的多项式系数,说明该算法对于 RSC 码识别是有效的。

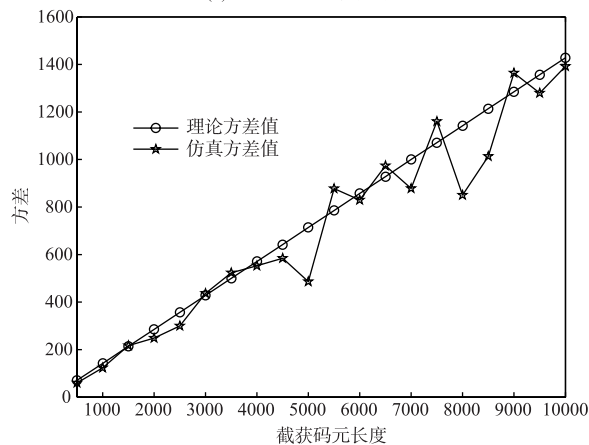
4.2 适应度值统计特性验证

进一步验证适应度函数值的概率分布,首先在不同误码率条件下,对比适应度函数的理论均值与仿真值,设定截获的码元长度 $L = 10000$,选择 $(15, 13)$ 为实际编码多项式, $(15, 17)$ 为遍历时的多项式;其次,固定误码率为 0.05,对比适应度函数值的理论方差与仿真方差,截获的码元长度从 500 到 10000,间隔 500 取值,同时在某一码元长度下,对适应度值重复仿真 100 次再求取方差,得到图 3 所示的结果。

从图 3 实际的适应度值概率分布与理论推导的概率分布情况是比较吻合的,仿真均值有所波动,但是还是在理论值 0 附近徘徊,说明了在均值这个统计量上,理论推导出的分布是正确的;从方差来看,仿真结果都在理论线附近徘徊,证明方差的理论推导结果也是正确的。综合上述分析,可得出,理论推导的适应度值概率分布能够较好的反映出实际的情况。



(a) 适应度函数均值分布验证



(b) 适应度函数方差分布验证

图3 适应度函数统计分布验证

4.3 算法容错性分析

4.3.1 编码寄存器个数对算法容错性影响

选取 5 种 RSC 编码器,其多项式分别为 (7,5)、(13,17)、(25,37)、(45,67) 以及 (113,133),编码器所对应的寄存器个数变换从 2 到 6,误码率变化范围从 0 到 0.3,间隔 0.01 取值,其它参数设定与 4.1 中一致,对每一种编码器进行 1000 次蒙特卡洛重复试验,统计正确的识别率,得到算法针对不同编码寄存器个数下的识别性能如图 4 所示。

从结果来看,编码寄存器个数越大,算法越不容易搜寻到全局最优解,从整体上来看,算法的识别性能是比较好,在误码率在 10^{-2} 量级时,各种类型的 RSC 码识别率能够达到 90% 以上。

4.3.2 截获码元长度对算法容错性影响

设定仿真时 RSC 码编码器为 RSC(13,17),其寄存器个数为 3。假定截获的码元长度为 500、1000、1500、2000、2500,误码率变化范围从 0 到 0.2 间隔 0.01 取值,其他参数设定与 4.1 中一致,对每一种码元长度,在不同误码率下进行 1000 次的蒙特卡洛仿真,统计正确的识别率,得到结果如图 5 所示。

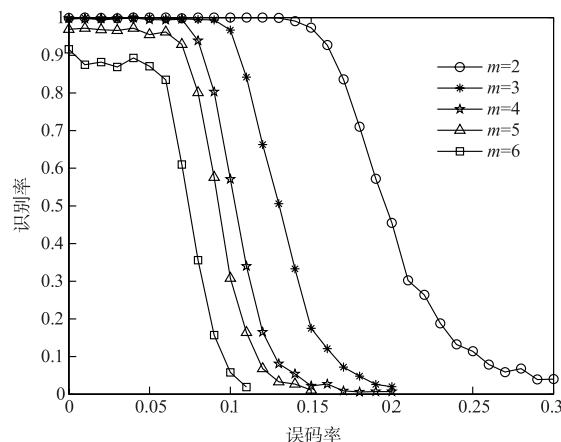


图4 不同寄存器个数下的识别性能

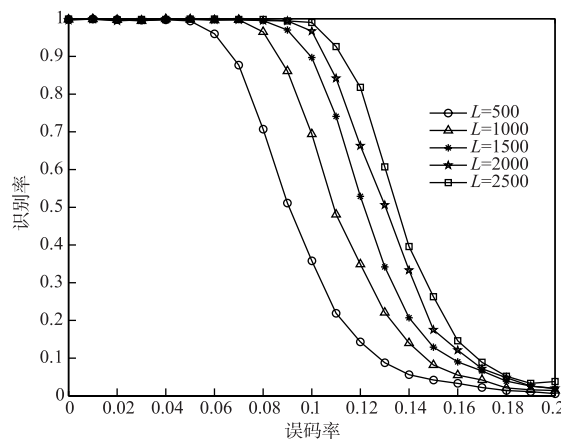


图5 截获码元长度对算法性能的影响

从结果来看,截获码元长度能够有效的提高算法对于参数的识别性能,在误码率较大的环境中,可以通过增加截获码元个数来克服寄存器个数过大所带来的算法性能下降的缺点。

4.3.3 估计的寄存器个数对算法的影响

在实际工程中,编码器的寄存器个数 m 多数条件下是不能获得的,研究在估计寄存器个数 m' 与实际的寄存器个数 m 不同差值条件下,算法的容错性能具有较大的现实意义。仿真是同样选用 (13,17),其编码寄存器个数为 3,工程实践中,编码器的寄存器个数取值范围为 [2,6],与实际的差值为 0、1、2、3,此时由于估计的寄存器个数增加了,等价于解空间的维度增加了,在多维空间中一定会使得解的个数增多,这 4 种情况所对应的解的个数分别为 1 种、3 种、7 种、15 种,每种情况下都除去了全零解,这些解都是正确的,同样通过 1000 次的蒙特卡洛试验,统计在不同误码率条件下的识别率,记录结果如图 6 所示。

从图 6 可以看出,算法的性能并不是单纯的随着 m' 与 m 的差值越大而逐渐变差,比如仿真结果在差值

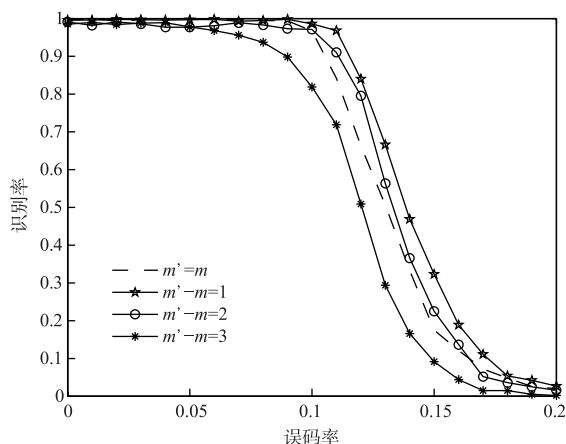


图6 不同估计编码寄存器个数下的算法性能

为 1 与 2 的两种情况下要好于 $m' = m$ 的情况,而在差值为 3 时,算法的性能要明显差于以上三种情况,出现这种结果的主要原因为:一方面,当估计的 m' 越大,会造成解空间的维度越大,优质解的个数与维度成 2 的次方倍的增加,对于算法而言就越能找到全局最优解;另一方面,当维度增加了同样会造成算法搜寻全局最优解的难度加大;从总体上来看, m' 与 m 差值变换对于算法的识别性能变化不大,好于或接近 $m' = m$ 的情况。

4.3.4 初始种群选取以及迭代次数对算法的影响

选取 (13,17) 为实际编码多项式,设定截获的码元长度为 1000,在固定的最大遗传代数为 20 的情况下,种群规模为 5,10,20,50;同时,在固定的种群规模为 50 时,最大遗传代数为 1,2,5,10;误码率变化范围为 0 到 0.2,间隔 0.01 取值,蒙特卡洛试验次数为 1000,得到如图 7 所示的结果。

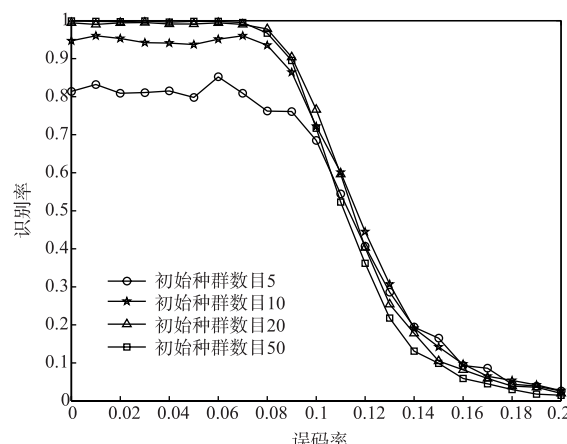
对于离散 PSO 算法而言,在误码率较小时,初始种群数目以及最大迭代次数越大,算法越能够实现参数的可靠识别,但当误码率逐渐增大后,这两种因素对于算法的影响并不明显,所以可以按照实际信道噪声恶劣程度来选择初始种群与最大迭代次数,从而减少算法的计算量。

4.4 离散 PSO 算法与经典算法的比较

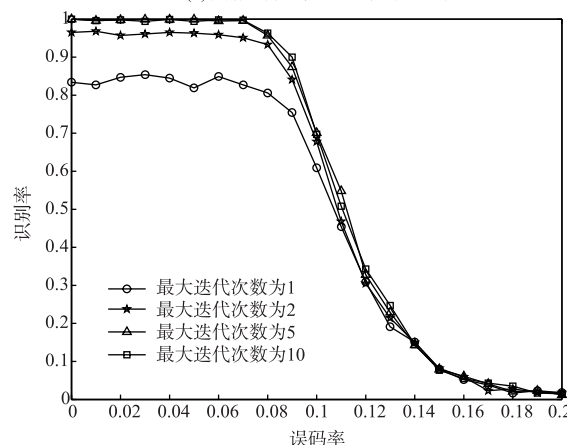
将 Walsh-Hadamard 变换算法与本文提出的识别算法进行比较,首先针对不同编码寄存器个数下,两种算法的识别性能对比,其结果如图 8 所示,其中实线为提出的算法的性能,虚线为 Walsh-Hadamard 变换算法。

分析所得到的结果可知,当编码寄存器个数较小时,所提出的算法性能要好于 Walsh-Hadamard 变化;当增加时,新算法的识别性能要差于 Walsh-Hadamard 变换,在低误码率的条件下,二者差异较为明显,高误码率下,性能较为接近。

进一步比较两种识别算法的计算复杂度,选取寄



(a) 初始种群规模对识别性能影响



(b) 迭代次数对于识别性能影响

图7 初始种群规模和迭代次数对于算法的影响

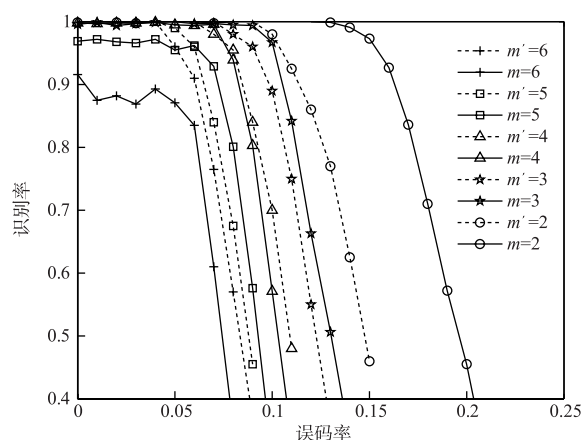


图8 提出算法与Walsh-Hadamard算法的比较

寄存器个数分别为 2 到 7,将 Walsh-Hadamard 变换算法以及离散 PSO 算法在同一条件下,对参数进行识别,得到完成一次识别,所需要的时间,见表 1 所示。

从表 1 两种算法完成一次参数识别的时间来看,当寄存器个数较少时,两者的识别时间相近,但当寄存器个数增加时,两者的差距就非常明显了.究其主要原因

是经典的 Walsh-Hadamard 变换算法计算量与 RSC 码寄存器个数呈指数倍增加,而本文提出的算法计算量仅与寄存器个数呈线性增加,因此本文算法的计算量要远远小于 Walsh-Hadamard 变化算法。

表 1 离散 PSO 算法与经典算法的识别时间对比

方法	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$
离散 PSO 算法	0.0473s	0.0611s	0.1422s	0.5736s	1.5873s	5.0877s
经典算法	0.0182s	0.3516s	1.1859s	6.4706s	52.998s	235.763s

5 结论

本文通过构建 RSC 码的编码模型,根据 RSC 码特殊的编码结构,提出了基于离散 PSO 算法下的 RSC 码编码多项式识别方法,推导出了适应度函数以及判决门限值.通过仿真实验验证了所提出算法对参数识别的正确性,并且研究了编码寄存器个数 m 、截获码元长度 l 对算法容错性能的影响.与 Walsh-Hadamard 变换对比,当编码寄存器个数不大时,新算法性能要远远优于 Walsh-Hadamard 变换,而当编码寄存器个数较大时,提出的识别算法容错性稍差于 Walsh-Hadamard 变换,但计算量与 Walsh-Hadamard 变换相比,大大减少.综上所述可知:所提出的算法容错性能较好,计算量较小,该方法在信息截获识别中具有一定的工程应用价值。

参考文献

- [1] Jaeyong S, Kyungwhoon C, Kyeongcheol Y. Low-complexity decoding of block Turbo codes based on the chase algorithm[J]. IEEE Communication Letters, 2017, 21(4): 706 – 709.
- [2] Thomas J, Balazs M. Turbo code design for short blocks [A]. 8th Advanced Satellite Multimedia Systems Conference and the 14th Signal Processing for Space Communications Workshop [C]. Palma de Mallorca, 2016. 1 – 6.
- [3] LI H, GAO Z, ZHAO M, et al. Partial iterative decode of turbo codes for on-board processing satellite platform[J]. IEEE Journals & Magazines, 2015, 12(11): 1 – 8.
- [4] 任亚博, 张健, 刘以农. 高误码率下 Turbo 码交织器的恢复方法[J]. 电子与信息学报, 2015, 37(8): 1927 – 1930.
REN Y B, ZHANG J, LIU Y N. Reconstruction of Turbo-code interleaver at high bit error rate[J]. Journal of Electronics & Information Technology, 2015, 37(8): 1927 – 1930. (in Chinese)
- [5] 刘俊, 李静, 彭华. 基于校验方程平均符合度的 Turbo 码交织器估计[J]. 电子学报, 2016, 44(5): 1213 – 1217.
LIU J, LI J, PENG H. Estimation of Turbo-code inverage conformity of parity-check equation[J]. Acta Electronica Sinica, 2016, 44(5): 1213 – 1217. (in Chinese)
- [6] 谢辉, 黄知涛, 王峰华. 信道编码盲识别技术研究进展[J]. 电子学报, 2013, 41(6): 1166 – 1176.
XIE H, HUANG Z T, WANG F H. Research progress of blind recognition of channel coding[J]. Acta Electronica Sinica, 2013, 41(6): 1166 – 1176. (in Chinese)
- [7] BARBIER J. Reconstruction of Turbo-code encoders[J]. Defense & Security, 2005, 5819: 463 – 473.
- [8] 解辉, 王峰华, 黄知涛, 张锡祥. 基于改进欧几里得算法的卷积码快速盲识别算法[J]. 国防科技大学学报, 2012, 34(6): 159 – 162.
XIE H, WANG F H, HUANG Z T, et al. A fast method for blind recognition of convolutional codes based on improved Euclidean algorithm[J]. Journal of National University of Defense Technology, 2012, 34(6): 159 – 162. (in Chinese)
- [9] 邹艳, 陆佩忠. 关键方程的新推广[J]. 计算机学报, 2006, 29(5): 711 – 718.
ZOU Y, LU P Z. A new generalization of key equation[J]. Chinese Journal of Computers, 2006, 29(5): 711 – 718. (in Chinese)
- [10] 刘健, 王晓军, 周希元. 基于 Walsh-Hadamard 变换的卷积码盲识别[J]. 电子与信息学报, 2010, 32(4): 884 – 888.
LIU J, WANG X J, ZHOU X Y. Blind recognition of convolutional coding based on Walsh-Hadamard transform[J]. Journal of Electronics & Information Technology, 2010, 32(4): 884 – 888. (in Chinese)
- [11] DEBESSU Y G, WU H C, JIANG H. Novel blind encoder parameter estimation for Turbo codes[J]. IEEE Communications Letters, 2012, 16(16): 1917 – 1920.
- [12] YU P D, LI J, PENG H. A least square method for parameter estimation of RSC sub-codes of Turbo codes[J]. IEEE Communications Letters, 2014, 18(4): 644 – 647.
- [13] 于沛东, 李静, 彭华. 一种利用软判决的信道编码识别新算法[J]. 电子学报, 2013, 41(2): 302 – 305.
YU P D, LI J, PENG H. A novel algorithm for channel coding recognition using soft-decision[J]. Acta Electronica Sinica, 2013, 41(5): 302 – 305. (in Chinese)
- [14] 武恒洲, 罗霄斌, 刘杰. Turbo 码盲识别技术研究[J]. 无线电工程, 2015, 45(5): 24 – 27.
WU H Z, LUO X B, LIU J. Research on blind recognition of Turbo codes[J]. Journal of Radio Engineering, 2015, 45(5): 24 – 27. (in Chinese)
- [15] 解辉, 王峰华, 黄知涛. 基于最大似然检测的 $(n, 1, m)$ 卷积码盲识别方法[J]. 电子与信息学报, 2013, 35(7): 1671 – 1676.
XIE H, WANG F H, HUANG Z T. Blind recognition of $(n, 1, m)$ convolutional code based on maximum likelihood detection[J]. Journal of Electronics & Information Technology, 2013, 35(7): 1671 – 1676. (in Chinese)

- [16] 张旻,陆凯,李歆昊,吕全通. 归零 Turbo 码的盲识别方法[J]. 系统工程与电子技术,2016,38(6):1424-1427.
ZHANG M, LU K, LI X H, et al. Blind recognition method for the Turbo codes on trellis termination[J]. Journal of Systems Engineering and Electronics, 2016, 38(6): 1424-1427. (in Chinese)
- [17] ALI N, OMID A, SAMAD F. Blind recognition algorithm of Turbo codes for communication intelligence systems [J]. International Journal of Computer Science Issues, 2011, 8(6): 68-72.
- [18] MARAZIN M, GAUTIER R, BUREL G. Blind recovery of k/n rate convolutional encoders in a noisy environment [J]. EURASIP Journal on Wireless Communications and Networking, 2011, 2011(1): 168-177.
- [19] 纪震,廖惠连,吴青华. 粒子群算法及应用[M]. 北京: 北京科学出版社, 2009: 17-50.
JI Z, LIAO H L, WU Q H. The Particle Swarm Optimization Algorithm and Its Application[M]. Beijing: Science Press. 2009. 17-50. (in Chinese)
- [20] 谢承旺,邹秀芬,夏学文,王志杰. 一种多策略融合的多目标粒子群优化算法[J]. 电子学报, 2015, 43(8): 1538-1544.

XIE C W, ZOU X F, XIA X W, et al. A multi-objective particle swarm optimization algorithm integrating multiply strategies[J]. Acta Electronica Sinica, 2015, 43(8): 1538-1544. (in Chinese)

作者简介



吴昭军 男, 1992 年生于四川蓬溪. 海军航空大学博士研究生, 主要研究方向为信道编码盲识别.



张立民 男, 1966 年出生于辽宁开原, 博士, 教授、博士生导师, 主要研究方向为卫星信号处理及应用.

E-mail: iamzlm@163.com