

基于分组简化粒子群算法的盲源分离

季 策, 单长芳, 沙 毅, 周荣坤

(东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

摘 要: 传统盲源分离算法普遍存在收敛精度低和易陷入局部最优的缺点, 针对上述问题, 提出将蛙跳算法的分组思想应用到盲源分离算法中. 该分组思想是将整个粒子群分为多组子群体, 每组粒子在进行组内寻优的同时进行全局寻优, 从而增加了粒子之间的差异性, 可以有效避免早熟收敛. 该算法以负熵为目标函数, 通过对分离矩阵进行调整, 使各个信号分量之间相互独立, 从而完成对瞬时混合信号的盲源分离. 实验仿真结果表明, 提出的算法与基本的粒子群盲源分离算法相比, 能有效避免早熟收敛并进一步提高收敛精度和算法的稳定性.

关 键 词: 盲源分离; 简化粒子群算法; 分组; 蛙跳算法; 负熵

中图分类号: TH 701; TN 911.7 文献标志码: A 文章编号: 1005-3026(2018)06-0787-05

Blind Source Separation Based on Grouping Simplified Particle Swarm Optimization Algorithm

Ji Ce, SHAN Chang-fang, SHA Yi, ZHOU Rong-kun

(School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China. Corresponding author: SHAN Chang-fang, E-mail: 1373755561@qq.com)

Abstract: Traditional algorithm of blind source separation (BSS) is easy to fall into partial optimum value, and the convergence precision is low. In view of these disadvantages, the BSS method based on improved simplified particle swarm optimization algorithm was proposed, by which the whole particle swarm could be divided into several groups. Each particle was optimized while optimizing the whole area, and the difference between particles was increased. What's more, premature convergence was avoided effectively. The negative entropy was taken as the objective function in the proposed algorithm, and the separation matrix was adjusted to separate each signal component from each other, so as to accomplish the blind source separation of instantaneous mixed signals. The simulation results show that the proposed algorithm is effective in avoiding premature convergence, and further improving convergence accuracy and algorithm stability compared with the basic particle swarm algorithm.

Key words: blind source separation (BSS); simplified particle swarm optimization (SPSO) algorithm; grouping; leapfrog algorithm; negentropy

盲源分离 (blind source separation, BSS) 产生于人们对“鸡尾酒会效应”问题的研究^[1], 其实质是仅根据观测信号实现对源信号的分离. 目前盲源分离技术已在语音信号^[2]和生物医学信号^[3]等方面得到了广泛的应用.

盲源分离算法大多采用梯度优化算法^[4]来进行寻优, 但梯度算法全局收敛性能不理想. 针对这一缺点, 一些学者提出将群智能算法, 如粒子群

算法^[5]、遗传算法及蚁群算法等应用到盲源分离的研究中. 粒子群算法由于参数少、易于实现等优点, 在盲源分离的过程中更具优势. 文献[6]提出了带有梯度加速粒子群算法的盲源分离, 文献[7]提出了一种基于动态学习因子的改进粒子群的盲源分离算法, 不足之处是该算法计算速度相对较慢. 文献[8]提出了一种自适应改变惯性权重的粒子群算法的盲分离, 利用粒子的适应度值

自适应地调整粒子的惯性权重,加快了算法的收敛速度,避免了早熟收敛. 胡旺等^[9]提出了一种更为简洁且高效的粒子群优化算法,该算法的进化方程中没有粒子的速度参数,而且方程由二阶降到了一阶,简化了粒子的进化过程. 但由于简化粒子群算法中的每个粒子在寻优过程中迭代公式差异性不大,导致粒子间的差异性不强,算法在进化后期容易出现早熟收敛的情况.

本文在文献[9]的基础上将分组思想应用在简化粒子群算法中. 仿真结果证明该算法能有效避免早熟收敛,提高收敛精度和算法稳定性.

1 简化粒子群算法

线性瞬时混合盲分离问题的数学描述为

$$\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t) + \mathbf{n}(t). \quad (1)$$

其中: $\mathbf{S}(t) = [s_1(t), s_2(t), \dots, s_n(t)]^T$ 为相互统计独立的源信号,源信号 $\mathbf{S}(t)$ 通过一个未知的线性系统后,在系统末端得到 m 个观测信号,即 $\mathbf{X}(t) = [x_1(t), x_2(t), \dots, x_m(t)]^T$; $\mathbf{n}(t)$ 为加性观测噪声. 在忽略噪声的情况下, $\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t)$. 线性瞬时混合盲分离的目标就是计算出分离矩阵 \mathbf{W} ,使得分离后的信号 $\mathbf{Y}(t)$ 为

$$\mathbf{Y}(t) = \mathbf{W}\mathbf{X}(t) = \mathbf{W}\mathbf{A}\mathbf{S}(t) = \hat{\mathbf{S}}(t). \quad (2)$$

其中 $\mathbf{Y}(t) = [y_1(t), y_2(t), \dots, y_n(t)]^T$, 如果能使 $\mathbf{W}\mathbf{A} = \mathbf{I}$, 则 $\mathbf{Y}(t) = \mathbf{S}(t)$, 那么源信号就从混合信号中分离了出来,从而解决了盲源分离问题.

粒子群算法是基于种群的随机优化算法. 种群中粒子的适应度值、飞行的速度和方向都由优化函数来决定. 每个粒子根据记忆去追踪最优粒子从而不断进行优化. 具有惯性权重的标准粒子群算法速度与位置更新公式如下:

$$v_i(t+1) = wv_i(t) + c_1r_1(p_{\text{best}} - x_i(t)) + c_2r_2(g_{\text{best}} - x_i(t)), \quad (3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (4)$$

其中: t 为迭代次数; w 为惯性权重; c_1 和 c_2 为学习因子; r_1 和 r_2 为分布于 $[0, 1]$ 间的随机数; p_{best} 为粒子自身最优位置; g_{best} 为群体最优位置.

文献[9]对基本的粒子群算法中的速度项进行分析,证明了粒子的速度项在粒子的进化过程中不是必要的,从而提出了一种简化粒子群算法 (SPSO), 描述如下:

$$x_i(t+1) = wx_i(t) + c_1r_1(p_{\text{best}} - x_i(t)) + c_2r_2(g_{\text{best}} - x_i(t)). \quad (5)$$

由式(5)和式(3), 式(4)对比可知, 式(5)中没有粒子的速度项. 文献[9]证明粒子速度与进

化过程无关. 该算法避免了人为确定参数 $[-v_{\text{max}}, v_{\text{max}}]$ 影响粒子收敛速度和收敛精度的问题, 而且方程也由二阶降到一阶, 有利于分析和控制粒子的进化过程.

2 基于改进的 SPSO 的盲源分离

2.1 分组简化粒子群算法

在上述简化粒子群算法中, 每个粒子在进化过程中都采用式(5)进行进化, 导致粒子间的差异性不强, 粒子就会出现早熟. 针对上述问题, 引入了蛙跳算法中的分组思想.

蛙跳算法分组思想: 在湿地中生活着一群青蛙, 青蛙群体被分为不同的子群体, 每个子群体首先执行局部搜索. 子群体中个体之间的文化都不同, 它们之间相互影响, 不断进化, 当达到一定阶段后, 再进行全局信息交换实现子群体之间的信息交流.

利用这个分组思想将整个粒子群分为多组子群体, 每组粒子在进行组内寻优的同时进行全局寻优, 充分利用了组内的最优位置信息, 这样不同组粒子的迭代公式各不相同, 增加了粒子之间的差异性, 可以有效避免早熟收敛.

分组简化粒子群算法可描述如下:

$$x_i(t+1) = wx_i(t) + c_1r_1(p_{\text{best}} - x_i(t)) + c_2r_2(g_{\text{best}} - x_i(t)) + c_3r_3(g'_{\text{best}} - x_i(t)). \quad (6)$$

式中: c_1 为自身学习因子; c_2 为组内学习因子; c_3 为全局学习因子; p_{best} 为粒子自身最优位置; g_{best} 为粒子组内最优位置; g'_{best} 为群体最优位置. 粒子根据 g_{best} 和 g'_{best} 来更新自己的位置, 充分利用了局部和全局信息.

2.2 基于分组 SPSO 的盲源分离

本文以负熵作为优化的目标函数, 在分离信号过程中, 当分离信号的非高斯性最大时, 表示已完成对混合信号的分离. 而非高斯性的大小由负熵来表示, 负熵的定义为

$$J(y) = H(y_g) - H(y). \quad (7)$$

其中 $H(y)$ 表示随机变量 y 的微分熵, 定义为

$$H(y) = - \int p(y) \lg p(y) dy. \quad (8)$$

其中, y_g 是高斯变量, 它与 y 的均值和方差相同. 高斯变量的非高斯性越强, 微分熵越小, 负熵越大. 所以 y 的非高斯性越强, 负熵越大, 即 $J(y)$ 越大; 负熵的计算比较复杂, 难以直接应用, 因此常利用式(9)逼近负熵:

$$J(y) = c\{E[G(y)] - E[G(y_g)]\}^2. \quad (9)$$

其中: c 为大于零的常数; $G(\cdot)$ 为非二次型函数, 常取

$$G_1(y) = \frac{1}{a} \lg \cosh(ay) \quad , \quad (10)$$

$$G_2(y) = -\exp(-\frac{y^2}{2}) \quad , \quad (11)$$

$$G_3(y) = \frac{1}{4}y^4 \quad . \quad (12)$$

式中, $1 \leq a \leq 2$. 式(11) 只能用于源信号为超高斯的情况, 式(10) 可用于超高斯和亚高斯混合情况. 本文仿真时采用式(10) 作为负熵的非二次型函数. 本文将利用上述的分组简化粒子群算法, 对分离矩阵进行优化, 从而使负熵达到最大, 最终完成对混合信号的分离. 基于分组简化粒子群的盲源分离算法分为如下几步:

- 1) 读取观测信号, 对其进行中心化和预白化;
- 2) 初始化粒子群, 并初始化各个参数;
- 3) 利用目标函数计算每个粒子的适应度, 并将粒子的适应度按大小进行排序并分组, 得到全局最优 g'_{best} ;
- 4) 更新每个粒子的个体最优位置和组内最优位置;
- 5) 对于每个小组中的粒子按照式(6) 更新自身位置, 迭代完成后对每个粒子按适应度大小进行排序, 排序后的粒子进入下一次组内迭代, 并转到4);
- 6) 达到组内迭代次数后, 各组粒子进行下一次分组, 转到3);
- 7) 达到分组次数后, 输出适应度最好的粒子, 即分离矩阵 W ;
- 8) 输出分离信号, 算法结束.
- 算法流程如图 1 所示.

3 仿真结果及分析

3.1 仿真环境

本文分别选用两路和四路实地采集的语音信号进行实验. 各项参数设置如下: 取种群规模为 50 (分为 5 组, 每组 10 个粒子), 惯性权重 w 从 0.9 到 0.4 线性递减, 学习因子 $c_1 = c_2 = c_3 = 1$, 算法迭代 50 次.

3.2 评价指标

本文采用算法的相似系数、信干比及粒子的全局平均适应度作为评价指标.

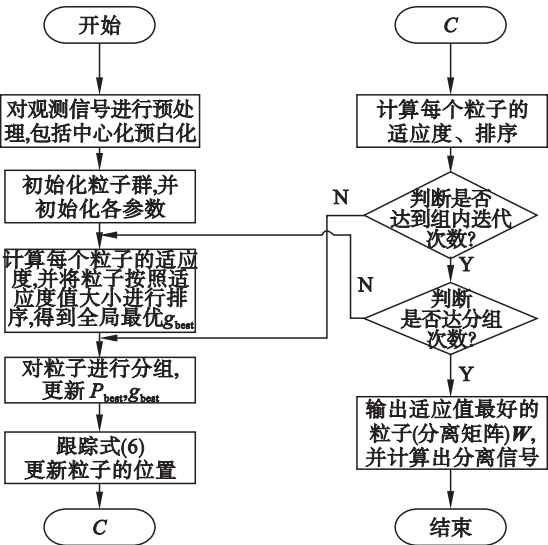


图 1 算法流程图
Fig. 1 Algorithm flowchart

相似系数 ρ_{ij} 定义如下:

$$\rho_{ij} = \frac{|\sum_{t=1}^N s_i(t)y_j(t)|}{\sqrt{\sum_{i=1}^N s_i^2(t) \sum_{j=1}^T y_j^2(t)}} \quad . \quad (13)$$

相似系数 ρ_{ij} 是用来衡量两个信号的相似程度, ρ_{ij} 越接近于 1 表示分离效果越好.

信干比定义如下:

$$SIR_i = \frac{\max_l |G_{il}|^2}{\sum_{j \neq l} |G_{ij}|^2} \quad . \quad (14)$$

式中: G 为系统矩阵, $G = WA$; $\max_l |G_{il}|^2$ 表示系统矩阵的第 i 行中绝对值最大的元素的平方; $\sum_{j \neq l} |G_{ij}|^2$ 表示系统矩阵其他元素的平方和, SIR 越大, 说明算法的分离性能越好.

3.3 仿真结果及分析

实验 1: 选用两路语音信号, 一路为男生信号, 另一路为女生信号. 采样频率为 10 kHz, 采样点数为 3×10^4 . 混合矩阵为二维随机矩阵:

$$A = \begin{bmatrix} 0.15124 & 0.6694 \\ -1.7400 & 0.9813 \end{bmatrix} \quad .$$

其中, 算法 1 表示基本粒子群算法的盲源分离, 算法 2 为本文提出的基于分组简化粒子群算法的盲源分离. 算法分离结果如图 2 ~ 图 5 所示.

由图 4 和图 5 可以看出, 基本粒子群算法和改进的粒子群算法都能完成混合声音信号的盲分离, 为了更加清楚、直观看到本文提出算法的有效性, 通过计算信号的相似系数 ρ_{ij} 、信干比 SIR 和平均适应度 F_{avg} 来衡量.

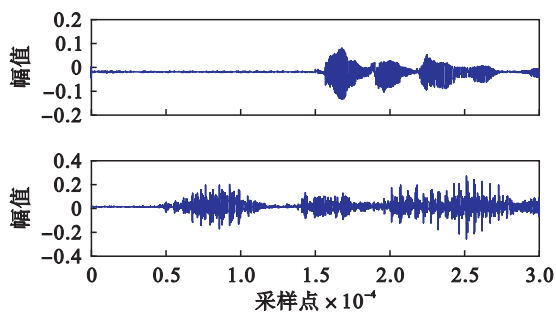


图 2 源信号
Fig. 2 Source signals

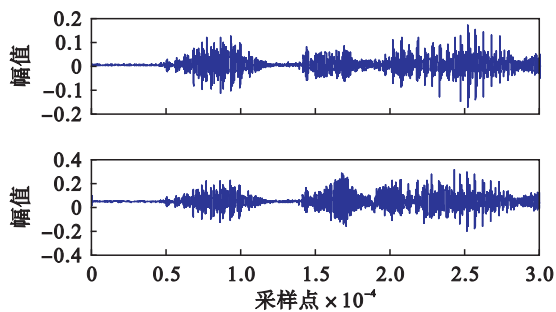


图 3 混合信号
Fig. 3 Mixed signals

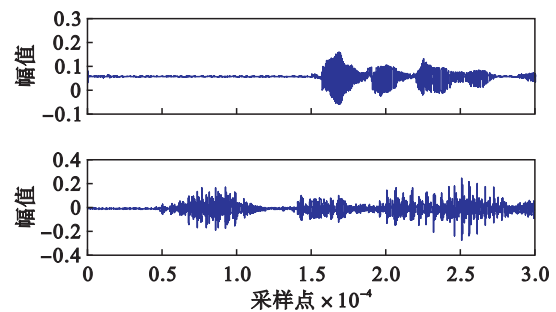


图 4 算法 1 分离的信号
Fig. 4 Separating signals by algorithm 1

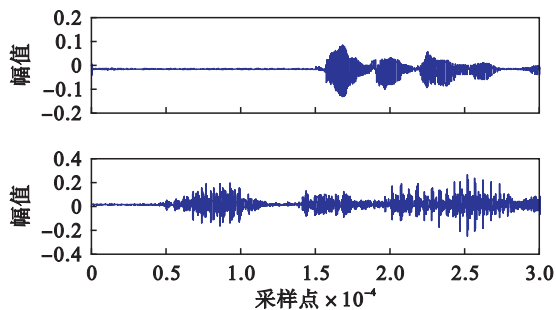


图 5 算法 2 分离的信号
Fig. 5 Separating signals by algorithm 2

表 1 算法性能对比

Table 1 Comparison of algorithm performance

参数	算法 1	算法 2
相似系数 ρ_{ij}	0.889 2	0.976 8
	0.899 0	0.993 5
信干比 SIR	4.376 5	5.554 7
平均适应度 F_{avg}	0.006 5	0.006 7

实验 2: 选用四路语音信号, 采样频率为 10 kHz, 采样点数为 3×10^4 . 混合矩阵为四维随机矩阵:

$$A = \begin{bmatrix} 0.153\ 74 & 0.832\ 5 & 0.223\ 4 & 0.885\ 6 \\ -1.741\ 3 & 0.682\ 3 & 0.114\ 8 & 0.667\ 5 \\ 0.774\ 3 & 0.036\ 9 & 0.371\ 5 & 0.537\ 4 \\ 0.583\ 2 & 0.114\ 3 & 0.064\ 7 & 0.775\ 1 \end{bmatrix}.$$

其中, 算法 1 表示基本粒子群算法的盲源分离, 算法 2 为本文提出的基于分组简化粒子群算法的盲源分离. 算法分离结果如图 6 ~ 图 9 所示.

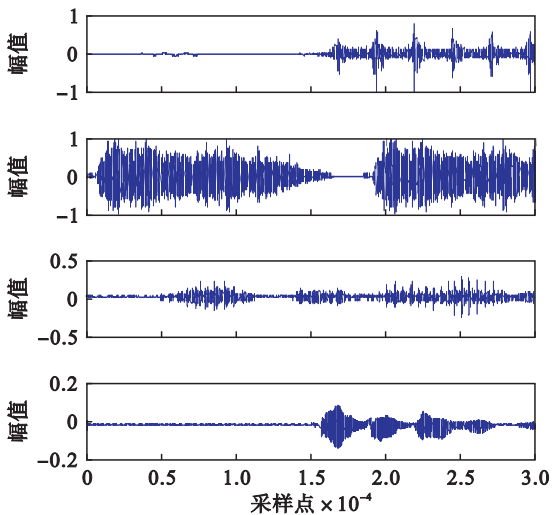


图 6 源信号
Fig. 6 Source signals

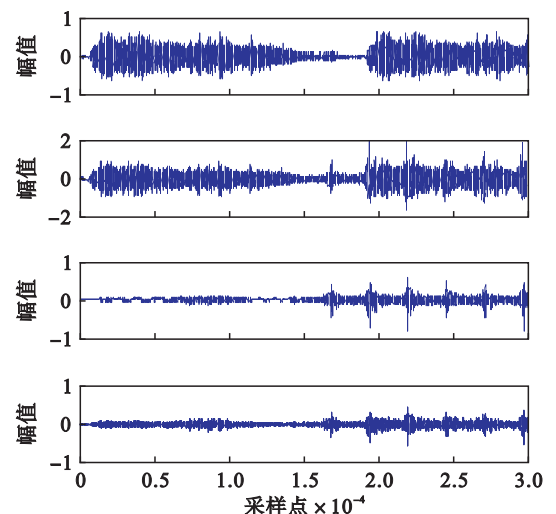


图 7 混合信号
Fig. 7 Mixed signals

从表 1 和表 2 可以看出, 本文提出的改进算法在相似系数和信干比上均要优于基本粒子群算法的盲源分离, 在粒子群的平均适应度上改进算法略好于基本算法. 在仿真分析的基础上, 本文又从算法复杂度方面进行分析, 从式(3)和式(4)可以看出基本粒子群算法粒子每次迭代需要计算 5

次加法和 5 次乘法运算,从式(6)可以看出本文改进算法粒子每次迭代需要计算 6 次加法和 7 次乘法运算,从而得出算法 1 和算法 2 的计算复杂度分别为 $N \times m \times 10$, $N \times m \times 13$. 其中, N 表示种群规模, m 表示迭代次数. 可以看出,本文提出的基于分组思想粒子群算法的算法复杂度要高于基本粒子群算法,但在收敛性能方面要好于基本粒子群算法.

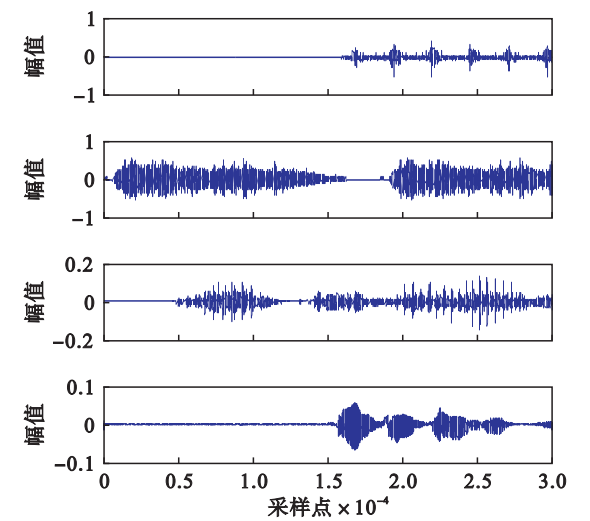


图 8 算法 1 分离的信号

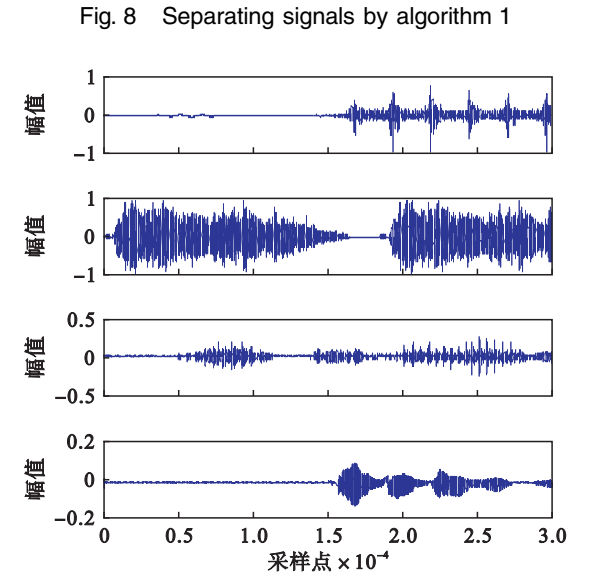


图 9 算法 2 分离的信号

Fig. 9 Separating signals by algorithm 2

表 2 算法性能对比

Table 2 Comparison of algorithm performance

参数	算法 1	算法 2
相似系数 ρ_{ij}	0.863 7	0.965 7
	0.898 7	0.994 2
	0.900 2	0.996 4
	0.879 3	0.987 7
信干比 SIR	6.745 3	8.239 8
平均适应度 F_{avg}	0.007 4	0.008 2

4 结 语

本文将蛙跳算法的分组思想应用到盲源分离中,提出了一种基于分组简化粒子群的盲源分离算法,并对其进行仿真和性能分析. 实验结果证明,本文提出的改进算法在相似系数和信干比上均要优于基本粒子群算法的盲源分离,在粒子群的平均适应度上改进算法略好于基本算法. 综上,本文的改进算法具有较高的搜索精度和稳定性,能够有效避免早熟收敛问题.

参考文献:

[1] Nie W, Li Y B, Liu D D. A blind source separation algorithm of non-stationary signals based on local polynomial Fourier transform [C]//Progress in Electromagnetic Research Symposium (PIERS). Shanghai, 2016:4996 – 5000.

[2] 季策, 刘梦蝶, 陶奕名. 基于皮尔逊系统的语音信号盲源分离[J]. 东北大学学报(自然科学版), 2015, 36(1):6 – 9.
(Ji Ce, Liu Meng-die, Tao Yi-ming. The blind source separation for speech signal based on person system [J]. Journal of Northeastern University (Natural Science), 2015, 36(1):6 – 9.)

[3] Subasi A, Gursoy M I. EEG signal classification using PCA, ICA, LDA and support vector machines [J]. Expert Systems with Applications, 2010, 37(12):8659 – 8666.

[4] Gao L, Zhang T Q, He D N, et al. A variable step-size EASI algorithm based on PI for DS-CDMA system blind estimation [C]//IEEE 5th International Congress on Image and Signal Processing. Chongqing, 2012:1730 – 1734.

[5] Liu J H, Mei Y, Li X D. An analysis of the inertia weight parameter for binary particle swarm optimization [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5):666 – 681.

[6] Luo T H, Zhang C. Blind source separation using the particle swarm optimization algorithm with gradient acceleration [J]. Computer Simulation, 2010, 27(10):112 – 115.

[7] Xi Z H, Bian L J, Jin Y. A novel blind source separation method based on improved particle swarm optimization [J]. Applied Science and Technology, 2010, 37(1):12 – 14 + 22.

[8] 张朝柱, 张健沛, 孙晓东. 基于自适应粒子群优化的盲源分离[J]. 系统工程与电子技术, 2009, 31(6):1275 – 1278.
(Zhang Chao-zhu, Zhang Jian-pei, Sun Xiao-dong. Blind source separation based on adaptive particle swarm optimization [J]. Systems Engineering and Electronics, 2009, 31(6):1275 – 1278.)

[9] 胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法 [J]. 软件学报, 2007, 18(4):861 – 868.
(Hu Wang, Li Zhi-shu. A simpler and more effective particle swarm optimization algorithm [J]. Journal of Software, 2007, 18(4):861 – 868.)