

DOI:10.11992/tis.201610004

网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.tp.20170407.1744.008.html>

# 依概率收敛的改进粒子群优化算法

钱伟懿, 李明

(渤海大学 数理学院, 辽宁 锦州 121013)

**摘要:** 粒子群优化算法是一种随机优化算法,但它不依概率 1 收敛到全局最优解。因此提出一种新的依概率收敛的粒子群优化算法。在该算法中,首先引入了具有探索和开发能力的两个变异算子,并依一定概率对粒子当前最好位置应用这两个算子,然后证明了该算法是依概率 1 收敛到  $\varepsilon$ -最优解。最后,将该算法应用到 13 个典型的测试函数中,并与其他粒子群优化算法比较,数值结果表明所给出的算法能够提高求解精度和收敛速度。

**关键词:** 粒子群优化算法;随机优化算法;变异算子;依概率收敛;全局优化;进化计算;启发式算法;高斯分布

**中图分类号:** TP301.6 **文献标志码:** A **文章编号:** 1673-4785(2017)04-0511-08

中文引用格式:钱伟懿,李明.依概率收敛的改进粒子群优化算法[J].智能系统学报,2017,12(4):511-518.

英文引用格式:QIAN Weiyi, LI Ming. Improved particle swarm optimization algorithm with probability convergence[J]. CAAI transactions on intelligent systems, 2017, 12(4): 511-518.

## Improved particle swarm optimization algorithm with probability convergence

QIAN Weiyi, LI Ming

(College of Mathematics and Physics, Bohai University, Jinzhou 121013, China)

**Abstract:** The particle swarm optimization (PSO) algorithm is a stochastic optimization algorithm that does not converge to a global optimal solution on the basis of probability 1. In this paper, we present a new probability-based convergent PSO algorithm that introduces two mutation operators with exploration and exploitation abilities, which are applied to the previous best position of a particle with a certain probability. This algorithm converges to the optimum solution on the basis of probability 1. We applied the proposed algorithm in 13 typical test functions and compared its performance with that of other PSO algorithms. Our numerical results show that the proposed algorithm can improve solution precision and convergence speed.

**Keywords:** particle swarm optimization; stochastic optimization algorithm; mutation operator; probability convergence; global optimization; evolutionary computation; heuristic algorithm; Gaussian distribution

粒子群优化算法是 Kennedy 等<sup>[1]</sup>在 1995 年提出的一种群体搜索的随机优化算法。由于 PSO 算法的参数少而且易操作,所以在实际问题中得到了广泛的应用。对 PSO 算法的研究主要有以下几个方面:1)对 PSO 算法自身参数的改进,这方面的工作主要关于惯性权重的自适应改进<sup>[2-9]</sup>和对学习因子的改进<sup>[9-11]</sup>;2)将其他进化算法与 PSO 相结合,比如,遗传算法与 PSO 算法结合<sup>[12-13]</sup>,差分进化算法与 PSO 算法结合<sup>[14]</sup>,模拟退火算法与 PSO 算法

结合<sup>[15]</sup>;3)在 PSO 算法引入一些改进 PSO 算法性能的其他算子,比如,把高斯扰动策略加入粒子群优化算法中<sup>[16]</sup>,把均匀设计方法引入到粒子群优化算法中<sup>[17]</sup>,把变异策略<sup>[18]</sup>、精英策略<sup>[19]</sup>、局部搜索策略<sup>[20]</sup>及邻域搜索策略<sup>[21]</sup>引入到粒子群优化算法中;4)PSO 算法的理论分析,比如,基于线性系统理论研究 PSO 收敛性<sup>[22-25]</sup>,基于随机过程研究 PSO 收敛性<sup>[26]</sup>,但是粒子群算法不依概率收敛<sup>[26]</sup>。本文给出了一种依概率 1 收敛的 PSO 算法,该算法在标准粒子群优化算法实施位置更新后按一定概率对较好的粒子实施具有开发能力的变异操作,对较差的粒子实施具有探索能力的变异操作,从而平衡了算法的全局搜索能力和局部搜索能力,提高了算

收稿日期:2016-10-05. 网络出版日期:2017-04-07.

基金项目:国家自然科学基金项目(11371071);辽宁省教育厅科学研究项目(L2013426).

通信作者:钱伟懿. E-mail:qianweiyi2008@163.com.

法的效率,另外,证明了该算法依概率1收敛到 $\varepsilon$ -最优解。实验结果表明,提出的算法能够提高全局搜索能力,算法的收敛速度明显加快。

## 1 标准 PSO 算法

标准粒子群算法中,粒子群由 $N_p$ 个粒子组成,在时刻 $k$ ,第 $i$ 个粒子在 $d$ 维空间中速度和位置的更新公式为

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (2)$$

式中: $i=1,2,\dots,N_p, d=1,2,\dots,D, D$ 为决策变量的维数, $\omega$ 为惯性权重, $\mathbf{x}_i^k = (x_{i1}^k, x_{i2}^k, \dots, x_{id}^k)$ 表示粒子 $i$ 的当前的位置; $\mathbf{v}_i^k = (v_{i1}^k, v_{i2}^k, \dots, v_{id}^k)$ 是粒子 $i$ 的当前的速度; $\mathbf{P}_i^k = (p_{i1}^k, p_{i2}^k, \dots, p_{id}^k)$ 是粒子 $i$ 当前最优位置, $\mathbf{P}_g^k = (p_{g1}^k, p_{g2}^k, \dots, p_{gd}^k)$ 是全局最优位置; $c_1, c_2$ 是加速度因子, $r_1, r_2$ 是0~1之间的随机数。第 $k$ 次迭代的惯性权重表示为

$$\omega_k = (\omega_{\text{start}} - \omega_{\text{end}}) \left( \frac{K_{\text{max}} - k}{K_{\text{max}}} \right) + \omega_{\text{end}} \quad (3)$$

式中: $\omega_{\text{start}}, \omega_{\text{end}}$ 分别为最初和最终的惯性权重, $K_{\text{max}}$ 是最大迭代步。

## 2 改进的 PSO 算法

提高 PSO 算法性能的有效方法之一就是平衡算法的探索能力和开发能力。我们的目的是对粒子所经历的最好位置进行操作,一方面能够使其跳出局部最优;另一方面使其具有一定的局部搜索能力。为此,给出两个变异算子:

$$\bar{P}_{id}^k = P_{id}^k \cdot \rho_1 N(0, \sigma^2) \quad (4)$$

$$\bar{P}_{id}^k = P_{id}^k + \rho_2 N(0, \sigma^2) \quad (5)$$

式中: $\rho_1 > 1$ 和 $\rho_2 > 1$ 是常数, $N(0, \sigma^2)$ 是具有时变标准差 $\sigma$ 的高斯分布的随机变量, $\sigma$ 表示为

$$\sigma = \sigma_{\text{max}} - (\sigma_{\text{max}} - \sigma_{\text{min}}) \frac{k}{K_{\text{max}}} \quad (6)$$

式中: $\sigma_{\text{max}}$ 和 $\sigma_{\text{min}}$ 是标准差 $\sigma$ 的最终和初始值, $k$ 是当前迭代步, $K_{\text{max}}$ 是最大迭代步。因为 $\rho_1 N(0, \sigma^2)$ 的比值比1大或比1小,又因为 $P_{id}^k \cdot \rho_1 N(0, \sigma^2)$ 是一个随机值,所以它比 $P_{id}^k$ 大或小,这样 $\bar{P}_{id}^k$ 可能远离 $P_{id}^k$ ,因此说式(4)具有一定的全局搜索能力。另一方面,式(5)表明在 $P_{id}^k$ 附近搜索,因此式(5)具有一定的局部搜索能力。

如果每次都对 $\mathbf{P}_i^k$ 实施式(4)或式(5)的变异操作,那么导致算法运行时间长,因此按一定概率 $\mu$ 对

$\mathbf{P}_i^k$ 实施式(4)或式(5)的变异操作,即当 $f(\mathbf{P}_i^k) \leq f_{\text{ave}}$ 时,对 $\mathbf{P}_i^k$ 按概率 $\mu$ 实施式(5)的变异操作,当 $f(\mathbf{P}_i^k) > f_{\text{ave}}$ 时,对 $\mathbf{P}_i^k$ 按概率 $\mu$ 实施式(4)的变异操作,其中

$$f_{\text{ave}} = \frac{1}{N_p} \sum_{j=1}^{N_p} f(\mathbf{P}_j^k), \quad (7)$$

$$\mu = \begin{cases} \frac{\alpha(f_{\text{ave}} - f(\mathbf{P}_i^k))}{f_{\text{ave}} - f(\mathbf{P}_g^k)}, & f(\mathbf{P}_i^k) \leq f_{\text{ave}} \\ \frac{\alpha(f(\mathbf{P}_i^k) - f_{\text{ave}})}{f_{\text{max}} - f_{\text{ave}}}, & f(\mathbf{P}_i^k) > f_{\text{ave}} \end{cases} \quad (8)$$

式中: $f_{\text{max}} = \max_{1 \leq i \leq N_p} \{f(\mathbf{P}_i^k)\}$ ,  $\alpha$ 是一个自适应参数,它被定义为

$$\alpha = 1 - 0.8 \frac{k}{K_{\text{max}}} \quad (9)$$

由式(8)知,对适应值较好的粒子实施局部搜索的概率较大,而对适应值较差的粒子实施全局搜索的概率较大。由于当算法迭代到后期,粒子比较集中,从而导致 $\frac{f_{\text{ave}} - f(\mathbf{P}_i^k)}{f_{\text{ave}} - f(\mathbf{P}_g^k)}$ 或 $\frac{f(\mathbf{P}_i^k) - f_{\text{ave}}}{f_{\text{max}} - f_{\text{ave}}}$ 接近1,若 $\alpha$ 大,那么实施式(4)或式(5)的变异操作的粒子较多,从而导致算法运算时间长,因此按式(9)实行单调递减策略。

位置与速度越界处理:

$$x_{id}^{k+1} = \begin{cases} x_{id}^{k+1}, & l_d \leq x_{id}^{k+1} \leq u_d \\ l_d + \text{rand}() \cdot (u_d - l_d), & \text{其他} \end{cases} \quad (10)$$

$$v_{id}^{k+1} = \begin{cases} v_{id}^{k+1}, & v_{\text{min}} \leq v_{id}^{k+1} \leq v_{\text{max}} \\ v_{\text{min}}, & v_{id}^{k+1} < v_{\text{min}} \\ v_{\text{max}}, & v_{id}^{k+1} > v_{\text{max}} \end{cases} \quad (11)$$

式中: $v_d^{\text{min}} = 0.5l_d, v_d^{\text{max}} = 0.5u_d$ 。

改进 PSO 算法的流程如下:

1) 设置参数,随机初始化粒子的位置和速度,计算各粒子的适应度,确定每个粒子的经历过的最优位置及全局最优位置,令 $k=1$ ;

2) 按式(3)计算惯性权重,并按式(1)和式(2)进行速度与位置更新;

3) 按式(10)和式(11)进行越界处理;

4) 对粒子 $i(i=1,2,\dots,N_p)$ 计算其适应值,更新其经历过的最优位置 $\mathbf{P}_i^{k+1}$ 和全局最优位置 $\mathbf{P}_g^{k+1}$ ,并按式(7)计算 $f_{\text{ave}}$ ;

5) 若 $f(\mathbf{P}_i^{k+1}) \leq f_{\text{ave}}$ ,按式(9)计算 $\alpha$ ,按式(8)计算 $\mu$ ,在 $[0,1]$ 选取一个随机数 $\text{rand}$ ,若 $\text{rand} < \mu$ ,则按式(6)计算 $\sigma$ ,按式(5)更新粒子经历过的最优位置,得到 $\bar{\mathbf{P}}_i^{k+1}$ ,转7);

6) 若 $f(\mathbf{P}_i^{k+1}) > f_{\text{ave}}$ ,按式(9)计算 $\alpha$ ,按式(8)计

算 $\mu$ ,在 $[0,1]$ 选取一个随机数 rand,若  $\text{rand} < \mu$ ,则按(6)式计算 $\sigma$ ,按(4)式更新粒子经历过的最优位置,得到 $\bar{P}_i^{k+1}$ ;

7)令

$$P_i^{k+1}(\text{new}) = \begin{cases} \bar{P}_i^{k+1}, & f(\bar{P}_i^{k+1}) < f(P_i^{k+1}) \\ P_i^{k+1}, & f(\bar{P}_i^{k+1}) \geq f(P_i^{k+1}) \end{cases};$$

8)令 $P_i^{k+1} = P_i^{k+1}(\text{new})$ ,更新 $P_g^{k+1}$ ;

9)判断是否满足终止条件(本文采用最大迭代步),若满足则输出结果,否则令 $k=k+1$ ,转2)。

### 3 收敛性分析

本文考虑如下全局优化问题:

$$\text{minimize } f(x)$$

$$\text{subject to } x \in S$$

式中: $f: S \rightarrow R$ 是实值函数,  $S = \{x_{\min} \leq x \leq x_{\max}\}$ ,

$x_{\min} = (l_1, l_2, \dots, l_D)$ ,  $x_{\max} = (u_1, u_2, \dots, u_D)$ 。

**定义1** 设 $f(x)$ 是实值函数, $x^* \in S$ ,若对任意 $x \in S$ ,

$$f(x^*) \leq f(x)$$

则称 $x^*$ 为 $S$ 的全局最优解。

**定义2** 对任意 $\varepsilon > 0$ ,设

$$B_\varepsilon = \{x \in S \mid |f(x) - f(x^*)| < \varepsilon\}$$

则称 $B_\varepsilon$ 为 $\varepsilon$ 最优解集, $x \in B_\varepsilon$ 被称为 $\varepsilon$ -最优解。

本文假设 $\mu(B_\varepsilon) > 0$ ,其中 $\mu$ 为勒贝格测度。

由定义1和2得:

$$x_{id}^{k+1} = x_{id}^k + \omega_k v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \quad (12)$$

由式(12)可看出,在第 $k+1$ 代粒子群中第 $i$ 个粒子的状态由 $x_i^k, v_i^k, P_i^k, P_g^k$ 决定,所以令 $\varphi_i = (x_i, v_i, P_i, P_g) \in \Gamma$ ,其中 $\Gamma = S \times V \times S \times S \in R^{4D}$ 。

**定义3** 设 $\Phi_{N_p} = \{\Phi \mid \Phi = (\varphi_1, \dots, \varphi_{N_p}), \varphi_i \in \Gamma, i = 1, \dots, N_p\}$ ,则称 $\Phi_{N_p}$ 为状态空间, $\Phi \in \Phi_{N_p}$ 称为状态。

**定义4** 设 $\Omega = \{\omega \mid \omega = (\Phi_1, \Phi_2, \Phi_3, \dots), \Phi_k \in \Phi_{N_p}, \forall k\}$ ,则称 $\Omega$ 为样本空间, $\omega \in \Omega$ 称为状态序列,其中 $\Phi_k = (\varphi_1^k, \varphi_2^k, \dots, \varphi_{N_p}^k), \varphi_i^k = (x_i^k, v_i^k, P_i^k, P_g^k)$ 。

由改进算法可知, $f(P_g^{k+1}) \leq f(P_g^k)$ ,所以有如下引理。

**引理1** 对 $\forall k \geq 1$ ,若 $P_g^k \in B_\varepsilon$ ,则 $P_g^{k+1} \in B_\varepsilon$ 。

**引理2** 对 $\forall i, k$ ,若 $\bar{P}_i^k \in B_\varepsilon$ ,那么 $P_i^{k+1}(\text{new}) \in B_\varepsilon$ 。

**证明** 如果 $P_i^k(\text{new}) = P_i^k$ ,由算法7)知 $f(\bar{P}_i^k) \geq f(P_i^k)$ ,再由 $\bar{P}_i^k \in B_\varepsilon$ 有 $P_i^k \in B_\varepsilon$ ,从而 $P_i^{k+1}(\text{new}) \in$

$B_\varepsilon$ ;如果 $P_i^k(\text{new}) = \bar{P}_i^k$ ,则结论是显然的。

**引理3**  $\forall \Phi \in \Phi_{N_p}, P_g^{k+1}$ 由算法生成的,则存在 $\rho > 0$ ,使得

$$\text{pr}(P_g^{k+1} \in B_\varepsilon \mid \Phi_k = \Phi) \geq \rho$$

式中 pr 表示概率测度。

**证明** 如果 $P_i^k$ 实施变异算子(4)或(5)操作,则令 $\chi(\bar{P}_i^k) = 1$ ,否则令 $\chi(\bar{P}_i^k) = 0$ ,由算法步骤5)和步骤6)及式(8)有

$$\text{pr}(\chi(\bar{P}_g^{k+1}) = 1) = \alpha, \text{pr}(\chi(\bar{P}_g^{k+1}) = 0) = 1 - \alpha$$

于是

$$\text{pr}(P_g^{k+1} \in B_\varepsilon \mid \Phi_k = \Phi) \geq \text{pr}(\chi(\bar{P}_g^{k+1}) = 1) \cdot$$

$$\text{pr}((P_g^{k+1} \in B_\varepsilon \mid \Phi_k = \Phi) \mid \chi(\bar{P}_g^{k+1}) = 1) =$$

$$\alpha \cdot \text{pr}((P_g^{k+1} \in B_\varepsilon \mid \Phi_k = \Phi) \mid \chi(\bar{P}_g^{k+1}) = 1) =$$

$$\alpha \cdot \text{pr}(P_g^{k+1}(\text{new}) \in B_\varepsilon \mid \Phi_k = \Phi)$$

由引理2有

$$\text{pr}(P_g^{k+1} \in B_\varepsilon \mid \Phi_k = \Phi) \geq \alpha \cdot \text{pr}(\bar{P}_g^{k+1} \in B_\varepsilon \mid \Phi_k = \Phi)$$

由于 $\bar{P}_g^{k+1}$ 的每个分量是独立的,再由式(5)知,随机变量 $\bar{P}_g^{k+1}$ 的密度函数为

$$g(x_1, x_2, \dots, x_D) = \left( \frac{1}{\sqrt{2\pi}\sigma\rho_2} \right)^D \exp\left(-\sum_{d=1}^D \frac{(x_d - p_{gd}^{k+1})^2}{2\sigma^2\rho_2^2}\right)$$

于是

$$\text{pr}(\bar{P}_g^{k+1} \in B_\varepsilon \mid \Phi_k = \Phi) =$$

$$\int_{B_\varepsilon} g(x_1, x_2, \dots, x_D) dx_1 dx_2 \dots dx_D \geq$$

$$\left( \frac{1}{\sqrt{2\pi}\sigma_{\max}\rho_2} \right)^D \exp\left(-D \frac{2m^2}{\sigma_{\min}^2\rho_2^2}\right) \mu(B_\varepsilon)$$

式中  $m = \max_{1 \leq i \leq D} \{|l_i|, |u_i|\}$ 。

$$\text{令 } \rho = \left( \frac{1}{\sqrt{2\pi}\sigma_{\max}\rho_2} \right)^D \exp\left(-D \frac{2m^2}{\sigma_{\min}^2\rho_2^2}\right) \mu(B_\varepsilon), \text{ 则}$$

$$\text{pr}(P_g^{k+1} \in B_\varepsilon \mid \Phi_k = \Phi) \geq \rho > 0.$$

**定理1** 设 $(\Phi_1, \Phi_2, \Phi_3, \dots) \in \Omega$ 是由算法任意生成的状态序列,则序列 $\{P_g^k\}$ 依概率1收敛到 $\varepsilon$ -最优解,即

$$\lim_{k \rightarrow \infty} \text{pr}\{P_g^k \in B_\varepsilon\} = 1$$

**证明** 由算法可知, $\Phi_{k+1}$ 只与 $\Phi_k$ 有关,所以 $\Phi_1, \Phi_2, \Phi_3, \dots$ 是马尔可夫过程,由马尔可夫过程性质及引理1可知

$$\text{pr}(P_g^{k+1} \notin B_\varepsilon) = \text{pr}(P_g^1 \notin B_\varepsilon) \cdot$$

$$\prod_{t=1}^k \text{pr}(P_g^{t+1} \notin B_\varepsilon \mid P_g^t \notin B_\varepsilon) \leq$$

$$\prod_{t=1}^k \text{pr}(P_g^{t+1} \notin B_\varepsilon \mid P_g^t \notin B_\varepsilon)$$

令

$$\Delta=\left\{\boldsymbol{\Phi} \mid \boldsymbol{P}_g \notin \boldsymbol{B}_\varepsilon\right\} \subset \Phi_{N_p}$$

则  $\forall \boldsymbol{\Phi} \in \Phi_{N_p}$ , 由引理 3 有

$$\begin{aligned} \operatorname{pr}\left(\boldsymbol{P}_g^{t+1} \notin \boldsymbol{B}_\varepsilon \mid \boldsymbol{P}_g^t \notin \boldsymbol{B}_\varepsilon\right) & \leq \\ \operatorname{pr}\left(\boldsymbol{P}_g^{t+1} \notin \boldsymbol{B}_\varepsilon \mid \boldsymbol{\Phi}_t=\boldsymbol{\Phi}\right) & =1-\rho \end{aligned}$$

于是  $\operatorname{pr}\left(\boldsymbol{P}_g^{k+1} \in \boldsymbol{B}_\varepsilon\right) \leq 1-(1-\rho)^k$ , 故

$$\lim _{k \rightarrow \infty} \operatorname{pr}\left\{p_g^k \in \boldsymbol{B}_\varepsilon\right\}=1$$

4 数值实验

为了评价改进算法(简称,IPSO)的性能,我们

选取文献[24]中的 13 个测试函数进行实验研究。前 7 个函数为高维单峰函数,后 6 个函数是高维多峰函数。在本文中 13 个问题的维数都取 30。IPSO 算法的实验结果与 LDIWPSO<sup>[2]</sup>、CDIWPSO<sup>[3]</sup>、DAPSO<sup>[4]</sup>、和 SSRDIWPSO<sup>[5]</sup> 实验结果进行比较。所有算法的共同参数设置如下: $\omega_{\text {start }}=0.9, \omega_{\text {end }}=0.4, N p=30, c_1=c_2=2, v_{\text {min }}=0.5 x_{\text {min }}, v_{\text {max }}=0.5 x_{\text {max }}, K_{\text {max }}=3000$ , IPSO 算法的其他参数设置如下: $\rho_1=2, \rho_2=0.1, \sigma_{\text {max }}=1, \sigma_{\text {min }}=0.2$ 。所有算法的程序都是由 MATLAB2007 实现,且每个实验独立运行 30 次,实验结果见表 1 和表 2。

表 1 高维单峰函数的实验结果

Table 1 Experimental results for unimodal functions

测试函数		LDIWPSO	CDIWPSO	DAPSO	SSRDIWPSO	IPSO
$F_1$	最好收敛值	1.802 4×10 <sup>-18</sup>	8.055 3×10 <sup>-39</sup>	1.490 6×10 <sup>-20</sup>	6.891 5×10 <sup>-53</sup>	2.997 8×10 <sup>-43</sup>
	最差收敛值	1.998 0×10 <sup>-15</sup>	7.131 7×10 <sup>-22</sup>	4.750 8×10 <sup>-14</sup>	6.357 7×10 <sup>-43</sup>	4.949 0×10 <sup>-37</sup>
	平均收敛值	2.575 9×10 <sup>-16</sup>	2.381 7×10 <sup>-23</sup>	2.149 4×10 <sup>-15</sup>	3.353 2×10 <sup>-44</sup>	1.026 2×10 <sup>-38</sup>
	方差	1.759 6×10 <sup>-31</sup>	1.695 2×10 <sup>-44</sup>	7.566 9×10 <sup>-26</sup>	2.013 7×10 <sup>-86</sup>	2.384 3×10 <sup>-75</sup>
$F_2$	最好收敛值	1.188 5×10 <sup>-12</sup>	6.515 1×10 <sup>-19</sup>	7.289 3×10 <sup>-14</sup>	1.754 5×10 <sup>-24</sup>	4.928 1×10 <sup>-32</sup>
	最差收敛值	7.217 0×10 <sup>-9</sup>	1.190 2×10 <sup>-10</sup>	7.899 7×10 <sup>-8</sup>	3.325 0×10 <sup>-18</sup>	1.149 3×10 <sup>-27</sup>
	平均收敛值	3.129 5×10 <sup>-10</sup>	6.892 0×10 <sup>-12</sup>	4.696 6×10 <sup>-9</sup>	2.378 8×10 <sup>-19</sup>	1.271 9×10 <sup>-29</sup>
	方差	1.721 0×10 <sup>-18</sup>	6.055 7×10 <sup>-22</sup>	2.480 3×10 <sup>-16</sup>	5.764 2×10 <sup>-37</sup>	1.099 7×10 <sup>-56</sup>
$F_3$	最好收敛值	68.453 9	1.184 3	58.269 4	0.092 9	2.352 6×10 <sup>-25</sup>
	最差收敛值	668.476 2	55.922 7	577.009 0	4.094 8	7.447 0×10 <sup>-21</sup>
	平均收敛值	255.741 5	13.927 6	230.084 3	0.858 9	7.865 4×10 <sup>-22</sup>
	方差	2.405 3×10 <sup>4</sup>	0.624 4	1.493 7×10 <sup>4</sup>	0.877 1	2.058 4×10 <sup>-42</sup>
$F_4$	最好收敛值	2.036 5	0.539 2	1.824 1	0.127 3	1.810 1×10 <sup>-21</sup>
	最差收敛值	8.298 2	3.883 0	8.747 8	0.977 9	1.009 6×10 <sup>-19</sup>
	平均收敛值	4.807 3	2.256 1	5.184 8	0.465 9	2.770 1×10 <sup>-20</sup>
	方差	3.355 2	0.624 4	2.676 7	0.070 2	3.693 2×10 <sup>-40</sup>
$F_5$	最好收敛值	0.330 8	0.006 9	6.655 2	0.008 3	23.567 9
	最差收敛值	147.1466	96.355 4	119.877 5	91.975 0	24.636 9
	平均收敛值	50.775 2	35.523 1	44.303 6	39.886 9	24.121 4
	方差	1.496 7×10 <sup>3</sup>	877.047 8	1.116 4×10 <sup>3</sup>	906.351 6	0.026 9
$F_6$	最好收敛值	0	0	0	0	0
	最差收敛值	0	0	0	0	0
	平均收敛值	0	0	0	0	0
	方差	0	0	0	0	0
$F_7$	最好收敛值	0.006 4	0.007 7	0.013 1	0.004 7	1.927 1×10 <sup>-6</sup>
	最差收敛值	0.049 6	0.035 5	0.044 1	0.021 8	5.276 0×10 <sup>-4</sup>
	平均收敛值	0.024 8	0.017 1	0.027 7	0.011 5	1.402 0×10 <sup>-3</sup>
	方差	1.081 1×10 <sup>-4</sup>	4.312 5×10 <sup>-5</sup>	8.598 9×10 <sup>-5</sup>	1.507 9×10 <sup>-5</sup>	1.112 7×10 <sup>-8</sup>

表 2 高维多峰函数的实验结果  
Table 2 Experimental results for multimodal functions

测试函数		LDIWPSO	CDIWPSO	DAPSO	SSRDIWPSO	IPSO
$F_8$	最好收敛值	$-1.031\ 9\times10^4$	$-1.053\ 6\times10^4$	$-1.049\ 7\times10^4$	$-9.312\ 4\times10^3$	$-1.162\ 2\times10^4$
	最差收敛值	$-8.582\ 0\times10^3$	$-8.917\ 5\times10^3$	$-8.167\ 4\times10^3$	$-7.338\ 3\times10^3$	$-9.450\ 6\times10^3$
	平均收敛值	$-9.297\ 0\times10^3$	$-9.692\ 0\times10^3$	$-9.378\ 8\times10^3$	$-8.084\ 5\times10^3$	$-1.030\ 1\times10^4$
	方差	$2.223\ 8\times10^5$	$1.741\ 4\times10^5$	$2.281\ 8\times10^5$	$1.932\ 3\times10^5$	$1.589\ 2\times10^5$
$F_9$	最好收敛值	16.914 3	16.914 3	21.889 1	20.894 1	0
	最差收敛值	45.768 1	45.768 1	50.742 8	65.667 2	0
	平均收敛值	30.470 3	28.621 6	34.197 5	41.987 2	0
	方差	50.486 5	49.748 4	55.308 4	135.840 4	0
$F_{10}$	最好收敛值	$4.471\ 4\times10^{-10}$	$7.993\ 6\times10^{-15}$	$5.399\ 5\times10^{-11}$	$7.993\ 6\times10^{-15}$	$8.881\ 6\times10^{-16}$
	最差收敛值	$5.302\ 0\times10^{-8}$	$2.436\ 9\times10^{-11}$	1.899 7	1.340 4	$4.440\ 9\times10^{-15}$
	平均收敛值	$8.536\ 4\times10^{-9}$	$8.305\ 7\times10^{-13}$	0.330 1	0.067 0	$9.177\ 8\times10^{-16}$
	方差	$104\ 575\times10^{-16}$	$1.976\ 4\times10^{-23}$	0.432 3	0.089 8	$1.051\ 8\times10^{-31}$
$F_{11}$	最好收敛值	0	0	0	0	0
	最差收敛值	0.029 6	0.044 2	0.073 1	0.056 6	0
	平均收敛值	0.010 3	0.013 5	0.014 2	0.013 9	0
	方差	$7.749\times10^{-5}$	$1.864\times10^{-4}$	$2.981\times10^{-4}$	$2.413\times10^{-4}$	0
$F_{12}$	最好收敛值	$4.188\times10^{-7}$	$1.925\times10^{-32}$	$1.812\times10^{-8}$	$1.780\times10^{-32}$	$5.124\times10^{-24}$
	最差收敛值	0.414 7	0.311 0	0.830 0	0.621 9	$3.365\ 8\times10^{-20}$
	平均收敛值	0.058 7	0.027 6	0.086 4	0.072 6	$2.412\ 9\times10^{-21}$
	方差	0.008 7	0.005 1	0.034 2	0.025 0	$3.029\ 0\times10^{-41}$
$F_{13}$	最好收敛值	$1.484\ 0\times10^{-16}$	$1.367\ 6\times10^{-31}$	$2.408\ 6\times10^{-17}$	$3.445\ 2\times10^{-32}$	$8.076\ 7\times10^{-19}$
	最差收敛值	0.011 0	0.011 0	0.043 9	0.043 9	0.021 0
	平均收敛值	0.003 3	0.003 7	0.006 0	0.002 7	0.002 7
	方差	$2.622\ 6\times10^{-5}$	$2.775\ 2\times10^{-5}$	$1.076\ 9\times10^{-4}$	$1.000\ 7\times10^{-4}$	$3.465\ 5\times10^{-5}$

从表 1 可以看出,除了测试函数  $F_1$ 、 $F_5$  和  $F_6$  外,IPSO 算法与其他 4 种算法相比,在寻优能力和稳定性方面明显优于其他 4 种算法,特别是测试函数  $F_3$  和  $F_4$ ,其他 4 种算法不能获得到最优解,而 IPSO 算法能够得到较理想的最优解,且稳定性也非常好。对于测试函数  $F_1$  来说,IPSO 算法不如 SSRDIWPSO 算法,但是优于其他 3 种算法。对于测试函数  $F_5$  来说,IPSO 算法的最好收敛值不如其他 4 种算法,但是对于平均收敛值和方差来说,IPSO 算

法优于其他 4 种算法。对于测试函数  $F_6$ ,IPSO 算法与其他 4 种算法获得相同的结果。总之,对于高维单峰函数来说 IPSO 算法有一定优势,其原因如下:函数  $F_1\sim F_7$  都单峰函数,由于 IPSO 算法有较好的局部搜索能力,所以 IPSO 算法对于单峰函数具有一定优势。函数  $F_1$  是非线性简单的单峰函数,所以大多数算法都能够找到最优解;函数  $F_5$  是很难极小化的典型病态二次函数,由于其全局最优解与可达到的局部最优之间有一道狭窄的山谷,所以算法很



难辨别搜索方向,找到全局最优解的机会很小,因此 5 种算法都没有找到全局最优解,而函数  $F_7$  是含有随机变量的函数,由于目标函数不确定,找到非常理想的全局最优解也是较困难的。

从表 2 可以看出,除了测试函数  $F_{12}$  和  $F_{13}$  外 IPSO 算法与其他 4 种算法相比,在寻优能力和稳定性方面明显优于其他 4 种算法,特别是测试函数  $F_9$  和  $F_{11}$ ,其他 4 种算法不能获得较好最优解,而 IPSO 算法能够得到非常理想的最优解,且稳定性也非常好。对于测试函数  $F_{12}$  来说,关于最好收敛值,IPSO 算法不如 SSRDIWPSO 和 CDIWPSO 算法,但是优于其他两种算法,而关于最差收敛值、平均收敛值和方差,IPSO 算法远好于其他 4 种算法。对于测试函数  $F_{13}$  来说,对于最好收敛值 IPSO 算法不如 SSRDIWPSO 和 CDIWPSO 算法,对于平均收敛值,IPSO 算法与 SSRDIWPSO 算法一样优于其他 3 种算法。总之,除了函数  $F_8 \sim F_{13}$  外,对于其他函数 IPSO 算法都能够得到满意结果,其原因如下:函数  $F_8 \sim F_{13}$  都是多峰函数且有较多局部最优解,由于 IPSO 算法按一定概率对适应值较差的粒子实行全局搜索,对于适应值较好的粒子实行局部搜索,所以 IPSO 算法对于解决多峰函数有一定优势。但是由于函数  $F_8$  的全局最优解与局部最优相距较远,且有一定的欺骗性,导致算法朝着错误方向搜索,因此不易找到满意的全局最优解;函数  $F_{13}$  含有大量深度不同的“坑”,导致算法陷入第一个坑后不易跳出来,因此也不易找到满意的全局最优解。

为了更清楚且直观地比较各种算法的收敛性,我们分别从高维单峰函数和高维多峰函数中选取 3 个函数进行收敛曲线分析,图 1~图 3 分别表示测试函数  $F_1$ 、 $F_4$ 、和  $F_6$  的收敛曲线,其中用  $100^{-100}$  代替 0。图 4~图 6 分别表示测试函数  $F_8$ 、 $F_{10}$  和  $F_{12}$  的收敛曲线。

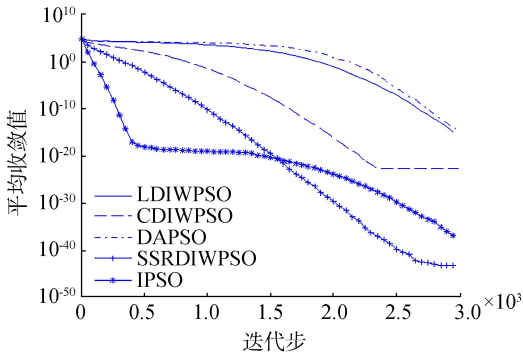


图 1 函数  $F_1$  的收敛曲线

Fig.1 Convergence curve of function  $F_1$

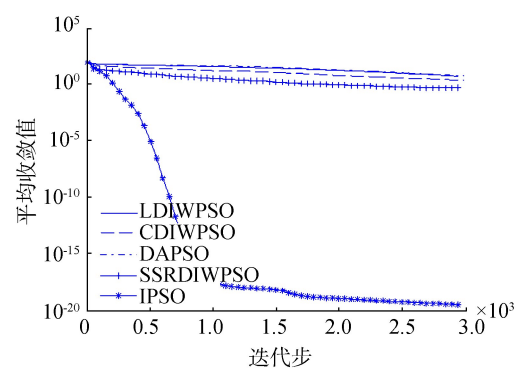


图 2 函数  $F_4$  的收敛曲线

Fig.2 Convergence curve of function  $F_4$

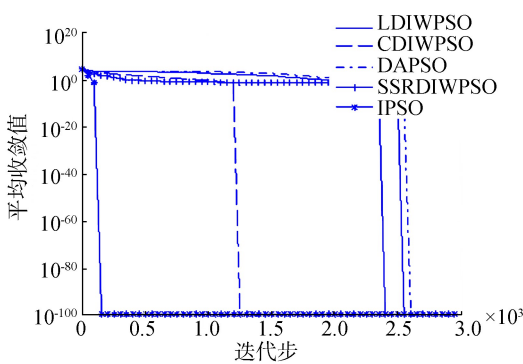


图 3 函数  $F_6$  的收敛曲线

Fig.3 Convergence curve of function  $F_6$

从图 1~6 中可以看出,对于函数  $F_4$ 、 $F_{10}$  和  $F_{12}$ , IPSO 算法的收敛速度及求解精度明显优于其他 4 种算法,对于函数  $F_6$ ,虽然求解精度一样,但是 IPSO 算法的收敛速度明显比其他 4 种算法收敛快,对于函数  $F_1$ ,虽然收敛精度 IPSO 算法不如 SSRDIWPSO 算法,但是他较快收敛到人们比较满意精度,对于函数  $F_8$ ,IPSO 算法的求解精度略优于其他 4 种算法。总之,IPSO 算法有较好的收敛速度和求解精度。

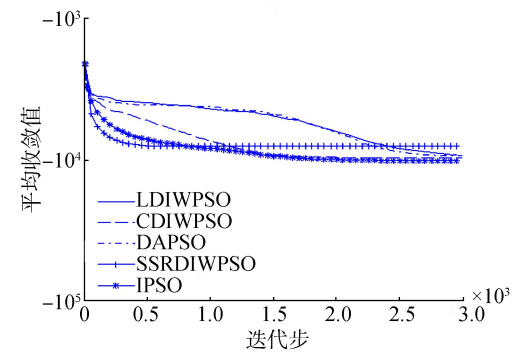


图 4 函数  $F_8$  的收敛曲线

Fig.4 Convergence curve of function  $F_8$

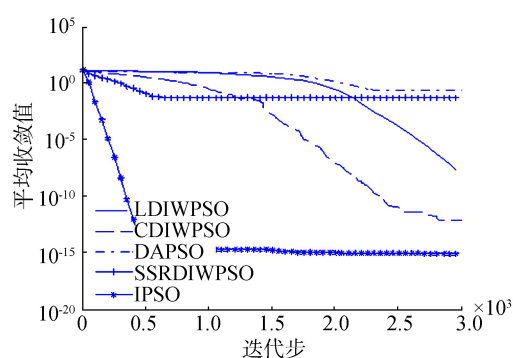


图 5 函数  $F_{10}$  的收敛曲线

Fig.5 Convergence curve of function  $F_{10}$

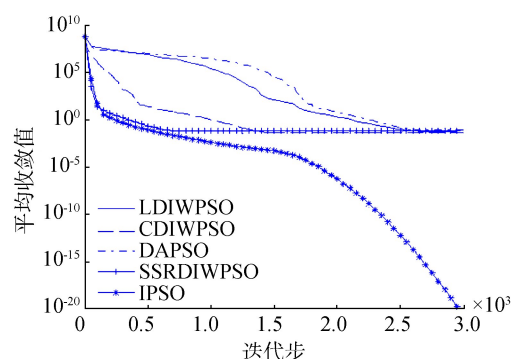


图 6 函数  $F_{12}$  的收敛曲线

Fig.6 Convergence curve of function  $F_{12}$

## 5 结束语

针对标准 PSO 算法不依概率收敛和易出现早熟收敛等缺点,提出了标准粒子群优化算法中增加了具有开发能力和探索能力两个变异算子,其目的是平衡算法的全局探索能力和局部搜索能力。并且从理论上证明所提出算法依概率 1 收敛到  $\varepsilon$ -最优解。数值结果表明,所提出的算法较大提高了全局寻优能力且具有较好的鲁棒性。在未来的工作中,将对变异算子中的参数设置进一步研究,以便提高算法收敛精度与收敛速度。

## 参考文献:

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization [C]//IEEE International Conference on Neural Networks. Perth, Australia, 1995: 1942-1948.
- [2] SHI Y H, EBERHART R C. Empirical study of particle swarm optimization[C]//IEEE International Conference on Evolutionary Computation. Washington, USA, 1999: 1945-1950.
- [3] FENG Y, TENG G F A X, WANG Y, et al. Chaotic inertia weight in particle swarm optimization[C]//Proceedings of the 2nd international conference on innovative computing,

- information and control. Kumamoto, Japan, 2007: 475.
- [4] SHEN X, CHI Z Y, CHEN J C. et al. Particle swarm optimization with dynamic adaptive inertia weight [C]//International conference on challenges in environmental science and computer engineering. Wuhan, China, 2010: 287-290.
- [5] ADEWUMI A O, ARASOMWAN A M. An improved particle swarm optimizer based on swarm success rate for global optimization problems [J]. Journal of experimental and theoretical artificial intelligence, 2016, 28 (3): 441-483.
- [6] PLUHACEK M, SENKERIK R, DAVENDRA D, et al. On the behavior and performance of chaos driven PSO algorithm with inertia weight[J]. Computers and mathematics with applications, 2013, 66: 122-134.
- [7] YANG C, GAO W, LIU N, et al. Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight [J]. Applied soft computing, 2015, 29: 386-394.
- [8] 郭建涛,刘瑞杰,陈新武. 用于跳频分量选取的修正适应度距离比粒子群算法[J]. 重庆邮电大学学报:自然科学版, 2015, 27(1): 26-30.  
GUO Jiantao, LIU Ruijie, CHEN Xinwu. Modified fitness-distance ratio based particle swarm optimizer for selection of frequency hopping components [J]. Journal of chongqing university of posts and telecommunications: natural science edition, 2015, 27(1): 26-30.
- [9] ARDIZZON G, CAVAZZINI G, PAVESI G. Adaptive acceleration coefficients for a new search diversification strategy in particle swarm optimization algorithms[J]. Information sciences, 2015, 299: 337-378.
- [10] 姜建国,田旻,王向前,等. 采用扰动加速因子的自适应粒子群优化算法[J]. 西安电子科技大学学报, 2012, 39(4): 74-79.  
JIANG Jianguo, TIAN Min, WANG Xiangqian, et al. Adaptive particle swarm optimization via disturbing acceleration coefficients[J]. Journal of xidian university, 2012, 39(4): 74-79.
- [11] 任凤鸣,李丽娟. 改进的 PSO 算法中学习因子( $c_1, c_2$ )取值的实验与分析[J]. 广东工业大学学报, 2008, 25(1): 86-89.  
REN Fengming, LI Lijuan. Experiment and analysis of the value selection of acceleration coefficients ( $c_1, c_2$ ) in IPSO method [J]. Journal of Guangdong university of technology, 2008, 25(1): 86-89.
- [12] GRIMACCIA F, MUSSETTA M, ZICH R. Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for electromagnetics [J]. IEEE transactions on antennas and propagation, 2007, 55(3): 781-785.
- [13] 熊伟丽,徐保国,吴晓鹏,等. 带变异算子的改进粒子群

- 算法研究[J]. 计算机工程与应用, 2006, 42(26): 1-3.
- XIONG Weili, XU Baoguo, WU Xiaopeng, et al. Study on the particle swarm optimization with mutation operator [J]. Computer engineering and applications, 2007, 55(3): 781-785.
- [14] 段玉红, 高岳林. 基于差分演化的粒子群算法 [J]. 计算机仿真, 2009, 26(6): 212-215.
- DUAN Yuhong, GAO Yuelin. A particle swarm optimization algorithm based on differential evolution [J]. Computer simulation, 2009, 26(6): 212-215.
- [15] 王丽芳, 曾建潮. 基于微粒群算法与模拟退火算法的协同进化方法 [J]. 自动化学报, 2006, 32(4): 630-635.
- WANG Lifang, ZENG JianChao. A cooperative evolutionary algorithm based on particle swarm optimization and simulated annealing algorithm [J]. Acta automatica sinica, 2006, 32(4): 630-635.
- [16] 艾兵, 董明刚. 基于高斯扰动和自然选择的改进粒子群优化算法 [J]. 计算机应用, 2016, 36(3): 687-691.
- AI Bing, DONG Minggang. Improved particle swarm optimization algorithm based on Gaussian disturbance and natural selection [J]. Journal of computer applications, 2016, 36(3): 687-691.
- [17] 刘宏达, 马忠丽. 均匀粒子群算法 [J]. 智能系统学报, 2010, 5(4): 336-341.
- LIU Hongda, MA Zhongli. A particle swarm optimization algorithm based on uniform design [J]. CAAI transactions on intelligent systems, 2010, 5(4): 336-341.
- [18] PEHLIVANOGLU Y V. A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks [J]. IEEE transactions on evolutionary computation, 2013, 17: 436-452.
- [19] LIM W H, MATISA N A. An adaptive two-layer particle swarm optimization with elitist learning strategy [J]. Information sciences, 2014, 273: 49-72.
- [20] WU G, QIU D, YU Y, et al. Superior solution guided particle swarm optimization combined with local search techniques [J]. Expert systems with applications, 2014, 41: 7536-7548.
- [21] WANG H, SUN H, LI C, et al. Diversity enhanced particle swarm optimization with neighborhood search [J]. Information sciences, 2013, 223: 119-135.
- [22] TIAN D P. A review of convergence analysis of particle swarm optimization [J]. international journal of grid and distributed computing, 2013, 6: 117-128.
- [23] LIU Q. Order-2 stability analysis of particle swarm optimization [J]. Evolutionary computation, 2014, 23: 187-216.
- [24] PAN F, LI X T, ZHOU Q, et al. Analysis of standard particle swarm optimization algorithm based on Markov chain [J]. Acta automatica sinica, 2013, 39(4): 81-289.
- [25] 戴月明, 朱达祥, 吴定会. 核矩阵协同进化的震荡搜索粒子群优化算法 [J]. 重庆邮电大学学报: 自然科学版, 2016, 28(2): 247-253.
- DAI Yueming, ZHU Daxiang, WU Dinghui. Shock search particle swarm optimization algorithm based on kernel matrix synergistic evolution [J]. Journal of chongqing university of posts and telecommunications: natural science edition, 2016, 28(2): 247-253.
- [26] YAO X, LIU Y, LIN G. Evolutionary programming made faster [J]. IEEE transactions on evolutionary computation, 1999, 3(2): 81-102.

#### 作者简介:



钱伟懿, 男, 1963年生, 教授, 博士, 主要研究方向为智能计算、优化理论与方法, 主持国家自然科学基金项目1项。发表学术论文60余篇, 出版专著3部。



李明, 男, 1991年生, 硕士研究生, 主要研究方向为智能计算。