基于 sharing 函数具有死区初始化功能的粒子群算法

李白雅[1],姜柏庄[1],段晓磊[2],黄强[1]

(1. 湖南科技大学 信息与电气工程学院,湖南 湘潭 411201;2. 湖南科技大学 人文学院,湖南 湘潭 411201)

摘 要:主要是研究粒子群优化原理,针对粒子群算法的中局部最优问题,提出一种具有死区初始化粒子群算法. 首先通过观察 MATLAB 可视化下粒子的运行轨迹,分析粒子陷入局部最优时的特征,并针对运行过程中出现停滞现象的粒子群,以当前局部最优粒子为中心画定"死区",并对"死区"内的粒子重新初始化. 利用标准测试函数进行测试,仿真结果表明,改进后算法不仅具有良好的稳定性,而且提高了粒子突破局部收敛限制的能力,从而提高了粒子群搜索最优解的能力.

关键词:PSO 算法;局部最优;死区初始化;MATLAB

中图分类号:TP301.6 **文献标识码**:A **文章编号**:1672-9102(2013)04-0064-05

在科学技术研究中,很多计算问题都可以归结为具有非线性和多峰特性目标函数的全局优化问题,粒子群优化是解决此类问题的一种有效方法,它具有程序实现简单、需要调整参数少、无需任何梯度信息等特点. 因此,在函数优化、组合优化以及许多工程领域均得到了广泛应用.

自 1995 年 Kennedy 和 Eberhart 提出粒子群优化(Particle Swarm Optimization, PSO)算法以来[1],得到了众多学者的关注,各种关于 PSO 算法应用的研究成果不断涌现. 该算法是受鸟类觅食行为的启发而提出的一种群体智能进化优化算法. 在此基础上 Shi 与 Eberhart 于 1988 年引入了惯性权重因子[2] ω 以协调算法的全局探测与局部搜索. 这种改进算法被称为标准粒子群算法(BPSO). 然而标准粒子群算法存在着收敛速度慢、容易陷入局部极值等缺点,许多学者都在 BPSO 的基础上进行过改进. 如算法位置速度公式本身的改进[3-4]以及与其他进化算法的结合[5-7]. 文献[8]提出了利用粒子群适应值的统计规律对粒子进行分类,对属于不同类别的粒子采用不同的进化模型,以此大大提高算法的优化效率和优化精度. 文献[9]提出针对运行过程中出现停滞现象的粒子群,围绕其局部最优位置重新初始化,引导粒子突破局部极值的限制. 总结以往算法改进的优势并结合 sharing 函数提出了一种"死区"初始化的方法,并针对运行过程中出现停滞现象的粒子群,以当前局部最优粒子为中心画定"死区",并对"死区"内的粒子重新初始化.

1 标准粒子群优化算法

PSO 算法的基本思想是将每个粒子视为优化问题的一个可行的潜在解,解的优劣程度由一个事先确定好的适应度函数(Fitness function)来确定[10]. 每个粒子在搜索空间中运动,并由一个速度变量来决定其飞行方向和距离,粒子通过对上一代速度的继承、自身飞行经验以及社会同伴的飞行经验来改变本代粒子的位置. 在每一次迭代中粒子将记忆 2 个极值:一个是粒子本身迄今为止获得的最优值 pbest,另一个是整个粒子群体至今为止获得的最优

收稿日期:2013-04-02

基金项目:湖南省科技计划项目(2011FJ6028)

通信作者:李白雅(1962-),女,湖南湘潭人,博士,教授,主要从事电机控制及设计研究. E-mail:380311779@qq.com

值 gbest.

假设一个由 M 个粒子形成的 D 维空间中,粒子以一定的速度在搜寻空间中飞行,粒子 i 在第 t 次迭代的状态如下.

位置:

$$x_i^t = (x_{i1}(t), x_{i2}(t), \cdots, x_{iD}(t))^{\mathrm{T}}.$$

$x_{id}(t) \in [ld, ud]$ 其中 ld, ud 分别为搜索空间的下限和上限.

速度: $v_i^t = (v_{i1}(t), v_{i2}(t), \cdots, v_{iD}(t))^{\mathrm{T}}$.

$v_{id}(t) \in [v_{\min}, v_{\max}]$ 其中 $v_{\min}, v_{\max}$ 分别为速度的最小和最大值.

粒子个体最优位置:

$$p\mathrm{best}_{iD}^t = (p\mathrm{best}_{i1}(t), p\mathrm{best}_{i2}(t), \cdots, p\mathrm{best}_{iD}(t))^{\mathrm{T}}.$$

种群全体历史最优位置:

$$g\mathrm{best}_{iD}^t = (g\mathrm{best}_{i1}(t), g\mathrm{best}_{i2}(t), \cdots, g\mathrm{best}_{iD}(t)).$$

式中 $1 \leqslant i \leqslant M, 1 \leqslant d \leqslant D$.

在 $t+1$ 次迭代的过程中通过以下公式更新粒子速度与位置:

$$v_{id}(t+1) = v_{id}(t) + R_1 C_1 (p\mathrm{best}_{id}(t) - x_{id}(t)) + R_2 C_2 (g\mathrm{best}_{id}(t) - x_{id}(t)). \quad (1)$$

$$x_{id}(t+1) = v_{id}(t) + x_{id}(t). \quad (2)$$

式中, R_1, R_2 为(0,1)区间均匀分布的随机数, c_1, c_2 为学习因子,(1)式中由 3 部分组成:一部分为对上一次速度的继承;一部分为自身飞行经验;一部分为同伴飞行经验.

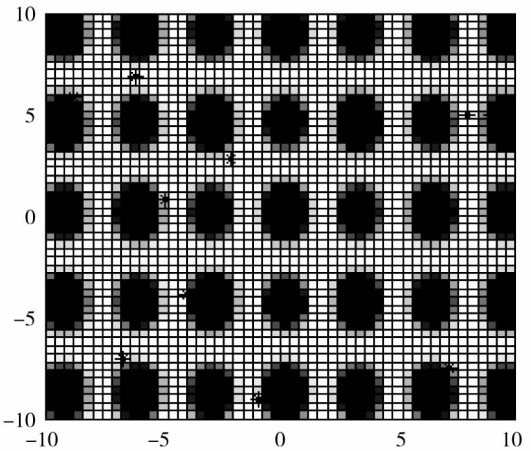
Y. Shi 在 1998 年对原始 PSO 算法进行了调整,增加了一个惯性权重 ω 来调整算法的全局与局部搜索能力,经试验研究发现惯性权重 ω 对算法性能形成主要影响,较大的 ω 值有利于跳出局部最优,利于进行全局寻优;较小的 ω 值有利于局部寻优,加速算法收敛. 修改后称为标准粒子群算法(SPSO). 其位置改变公式和原始粒子群算法相同,速度改变公式如下:

$$v_{id}(t+1) = \omega v_{id}(t) + R_1 C_1 (p\mathrm{best}(t) - x_{id}(t)) + R_2 C_2 (g\mathrm{best}_{id}(t) - x_{id}(t)). \quad (3)$$

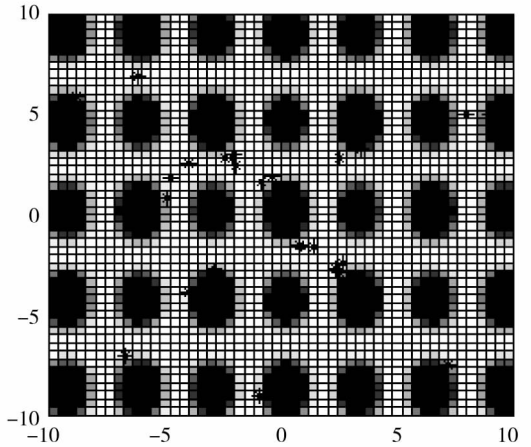
2 在 MATLAB 可视化下分析粒子搜索过程

为了能够更清晰直观的观测和分析粒子在陷入局部最优时的运行情况,本文在 MATLAB 可视化的前提下[11],使用 Griewank 函数作为演示函数,绘制了粒子运行轨迹图,如图 1 所示.

为了对粒子群算法的运行过程观察的更为仔细,演示过程中,粒子的种群数量设定为 15,最大迭代次数为 20 次. 从图 1 中对比分析可知,当粒子发现最优位置时会影响种群中的所有粒子,使粒子在当前这个最优位置的约束下向该点聚集,假设这一点为局部最优点,如果其他粒子在段时间内无法发现更好的适应值,则会出现局部最优而无法自行突破. 所以本文的改进方向为提出一种死区初始化的粒子群算法,在突破局部最优方面进行研究.



(Ⅰ)第一次运行图



(Ⅱ)第10次运行图

图 1 俯视三维演示图

Fig. 1 Overlooking the 3D demonstration graph

3 DZIA − PSO(dead − zone initiation adaptive PSO)算法

由粒子群的速度更新公式可以看出,当前的最优极值对粒子的进化方向和速度都有极大的约束作用,本文在 MATLAB 可视化的情况下对陷入局部最优的粒子进行分析可知,当粒子陷入局部最优后,经过 N 次迭代,大部分粒子将会聚集在局部极值的周围. 由于单个粒子受某个局部极值的约束,所以若大部分粒子均被同一个局部极值所限制时,粒子将很难通过自身的进化来突破这种限制,此时只能依靠其他粒子的成功发现才能突破局部区域的限制.

3.1 死区划分及死区内粒子初始化

以往的改进方法大部分注重单个粒子自身的进

化,当粒子陷入局部最优或停滞现象时大多也只是依靠粒子之间的合作和竞争[12-13],虽然很多改进方法可以在一定的程度上增大粒子群的多样性,但是对突破局部最优或停滞限制的能力仍然有限.

为了解决上述问题本文提出了一种"死区"初始化的方法. 主要思想是当判断粒子出现局部最优或停滞现象时,将以局部极值为中心、d_r 为粒子领域半径的区域划定死区,对死区范围内的粒子进行重新初始化,并利用 sharing 函数对进入死区的粒子进行排斥.

由粒子进化分析可知,当粒子陷入局部最优时粒子的个体最优和全局最优会出现更新缓慢或不更新现象,所以我们可以通过粒子最优解更新情况来判断粒子是否陷入局部最优[14],方法如下.

下面以第 k 步迭代为例,

令: $F_k^{\max} = \max\{p_i^k\}$, (4)

$$F_k^{\text{mean}} = \frac{1}{N}\sum_{i=1}^{N} p_i^k. \tag{5}$$

式中, p_i 表示第 i 个粒子曾经达到的最大适应值, $F_k^{\max}$ 分别表示第 k 步的全局最优适应值及平均个体最优适应值.

令: $H_1 = F_{k+1}^{\max} > F_k^{\max}$, (6)

$H_2 = F_{k+1}^{\text{mean}} > F_{k_1}^{\text{mean}}$. (7)

对于第 $k+1$ 步迭代而言,若条件 H_1 或 H_2 成立说明粒子进化正常,若条件连续 N 步迭代不成立或条件 H_2 连续 M 步迭代不成立($N > M$),则认为粒子群处于暂时的停滞状态. 在 M,N 的选取原则上要注意,条件 H_1 体现了全局搜索是某个粒子的成功寻优,而条件 H_2 更能体现出粒子群的整体进化趋势,尤其是在搜索后期,如果 M 的取值过小会大大降低粒子的收敛速度,如果 M 值过大则起不到应有的作用. 所以为了兼顾收敛速度和精度,M 的取值一般控制在最大迭代次数的百分之一,而 N 的值可以视 M 而定,一般是 M 的 5 ~ 8 倍.

粒子的初始化:

$$X_{i,k}^{\text{reset}} = P\text{best} + (\sigma_{i,1} - \sigma_{i,2})X_{\kappa}^{\max}. \tag{8}$$

初始化范围计算:

$$X_k^{\max} = \frac{x^{\max}}{\max\{1,(\kappa^{\max}-1)\}}, \tag{9}$$

$$x^{\max} = [x_1^{\max}, x_2^{\max}, \cdots, x_R^{\max}]. \tag{10}$$

式中,Pbest 为当前的局部极值位置; κ 为死区个数; $X_{i,k}^{\text{reset}}$ 为第 i 个粒子在第 κ 次重新初始化后的位置; $\sigma_{i,1}$,$\sigma_{i,2}$ 为相互独立的在[0,1]之间产生的随机数; $X_k^{\max}$ 为第 κ 次重新初始化的动态范围; $\kappa^{\max}$ 为设定的最大死区个数,要满足 $\kappa^{\max} > 1$; $x^{\max}$ 为粒子在 R 维空间的动态范围; $x_r^{\max}$ 为粒子在($1 < r < R$) 空间中的最大位置.

Sharing 函数由 Goldberg 于 1987 年提出[15],它通过共项函数调整群体中各个个体的适应度. 在部分粒子的重新分布过程中,算法能够根据调整后的新适应度适当的将粒子重新分布,本文利用 sharing 函数的这种特性对进入死区的粒子进行排斥.

方法如下:

$$\text{sharing}(d(i,j)) = \begin{cases} 1 - \dfrac{d(i,j)}{d_r}, \cdots d(i,j) < d_r; \\ 0, \cdots d(i,j) \geqslant d_r. \end{cases} \tag{11}$$

式中, d_r 为死区半径,大小由使用者给定; $d(i,j)$ 为粒子 i 与局部最优点 j 之间的距离. 对于进入死区的粒子基于 sharing 函数进行排斥操作

$$X_{i-\text{new}}^n = \frac{X_i^n}{1 - \text{sharing}(x)}. \tag{12}$$

式中, $X_i^n = (X_i^1, X_i^2, \cdots X_i^n)$ 为粒子 i 第 k 步所在的位置坐标. $X_{i-\text{new}}^n$ 为通过 sharing 函数排斥作用为粒子 i 产生的新坐标.

式(3) ~ 式(12)为 DZIA - PSO 算法的完整过程.

3.2 完整的 DZIA - PSO 算法流程

下面是完整的 DZIA - PSO 算法的流程:

Step1 初始化粒子群. 利用混沌随机数设定粒子的初始位置和速度.

Step2 评价每一个粒子. 计算粒子的适应值,如果好于该粒子当前的个体极值,则将 P_i 设置为该粒子的位置,且更新个体极值. 如果所有粒子的个体极值中最好的好于当前的全局极值则将 P_g 设置为该粒子的位置,且更新全局极值.

Step3 根据式 6 ~ 式 7,判断粒子进化是否正常,若正常进入 Step4,若不正常按下述步骤进行.

Step3.1 根据局部极值划定死区,并给死区编号 $j(j = 1,2,3\cdots\cdots n)$

Step3.2 根据 sharing 函数对死区范围的粒子进行重新初始化.

Step3.3 初始化完毕,进入 Step4

Step4 更新粒子的速度和位置.

Step5 检测粒子位置,判断是否进入死区. 若不存在死区或粒子位置不在死区范围内,进入 Step6 若进入死区范围内,根据死区编号和式(12)产生新的粒子位置并赋值给进入的粒子.

Step6 如果达到结束条件(达到设定最优位置或最大迭代次数)则结束,否则返回 Step2

4 实验及结果分析

4.1 测试函数及参数设定

为了检测 DZIA - PSO 算法对突破局部最优的

有效性,考虑如下 3 种标准测试函数[16]:

Rastrigin 函数:

$$F_1(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]. \quad (13)$$

式中, $x_i \in [-10,10]$.

Griewank 函数:

$$F_2(x) = \frac{1}{4\,000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1. \quad (14)$$

式中, $x_i \in [-600,600]$.

Ackley 函数:

$$F_3(x) = -20\exp\left(-0.2 \times \sqrt{\frac{1}{30}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e. \quad (15)$$

式中, $x_i \in [-10,10]$.

上述的测试函数都为复杂的多峰值函数,均拥有一个全局极小点,当 $x = \{0,0,\cdots,0\}$ 时取得全局最优值 $f(x) = 0$. 针对不同测试函数的特点,测试条件设置如下: $F_1(x)$, $F_2(x)$ 函数设置为 10 维, $F_3(x)$ 函数设置为 2 维,最大迭代次数都为 2 000;种群大小 popsize = 40;学习因子 $c1 = c2 = 2$;惯性权重 $w = 0.9$; $N = 60$; $M = 10$; $\kappa^{\max} = 100$.

4.2 结果分析

根据测试函数的特点,将测试函数本身作为适应度函数,将位置向量带入测试函数后,得到的函数值越小说明算法的优化效果越好. 由于函数的初始化位置是随机的,所以相同的算法和测试函数条件下每次得到的优化结果会出现略微差异. 本文将运用 LDW - PSO, DZIA - PSO 算法分别对 3 种测试函数进行优化,并各自独立运行 15 次,然后取平均值. 3 种函数的优化效果如图 2 ~ 图 4 所示.

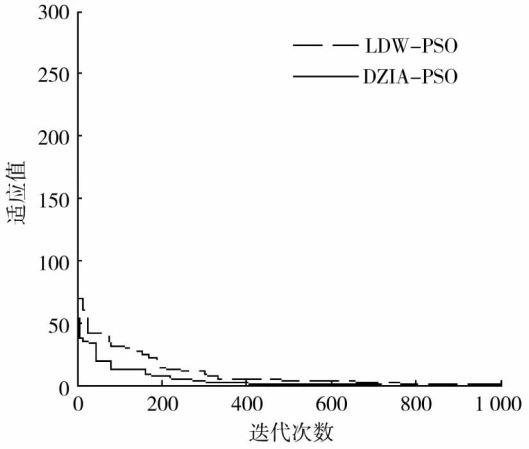


图 2 Rastrigin 函数的优化曲线

Fig. 2 Optimal curve for Rastrigin function

从图中的曲线可以看出对于收敛速度来说,虽然 LDW - PSO 在局部的收敛性能更好一些,但是当局部最优出现后,由于 ZDIA - PSO 算法对部分粒子进行了初始化是粒子获得了更大的速度,加强了搜索效率,所以对于多峰函数中当粒子的最大速度一定时,ZDIA - PSO 算法具有更好的全局搜索速度. 由图 4 可以看出当局部最优出现后 ZDIA - PSO 算法在突破限制的能力上有了明显的提高,在 10 次的进化过程中,改进算法没有失效现象,表明该改进算法具有良好的稳定性.

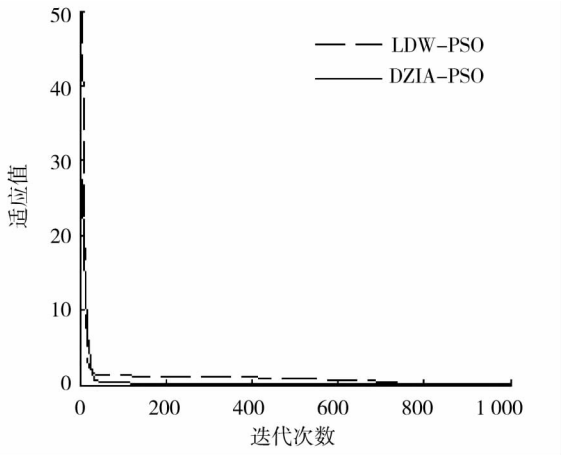


图 3 Griewank 函数的优化曲线

Fig. 3 Optimal curve for Griewank function

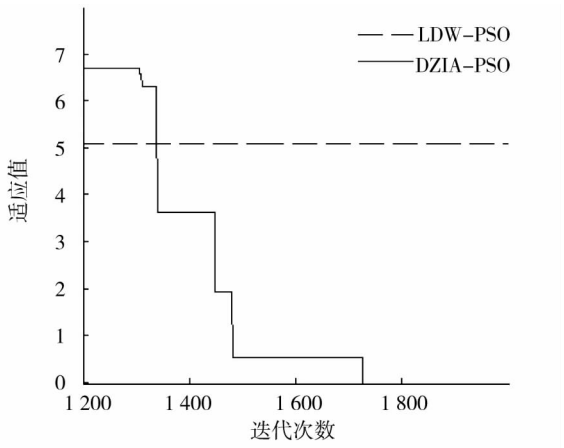


图 4 Ackley 函数的局部放大优化曲线

Fig. 4 Local optimal curve for Ackley function

5 结论

提出的基于 sharing 函数具有死区初始化功能的粒子群算法,通过研究在 MATLAB 可视化下粒子的局部最优现象,确定了通过死区初始化的方法来加强粒子突破局部最优的能力. 为了展现粒子突破局部最优的能力,对 3 种典型的多峰值测试函数进行了仿真实验,并与 LDW - PSO 进行了比较. 结果表明改进后的算法不仅具有良好的稳定性,而且提

高了粒子突破局部收敛限制的能力,使粒子群在搜索最优解的能力上有了进一步的提高.

参考文献:

[1] Kennedy J, Eberhart R C. Particle swarm optimization[C]//IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Press, 1995, 1942 – 1948.

[2] Shi Y, Eberhart R C. A modified particle swarm opitmzier[C]// IEEE World Congress on Computational Intelligence, 1998, 69 – 73.

[3] Kennedy J, Eberhart R C. Particle swarm optimization[C]//Proc IEEE. Int'l Conf Neural Networks. Piscataway, NJ: IEEE Service Center, 1995: 1942 – 1948.

[4] 耶刚强,孙世宇,梁彦,等. 基于动态粒子数的微粒群优化算法[J]. 信息与控制,2008,37(1):18 – 27.
Ye G Q, Sun S Y, Liang Y, et al. Particle swarm optimization based on dynamic particle number [J]. Information and Control, 2008, 37(1): 18 – 27.

[5] Chen G M, Huang X B, Jia J Y, et al. Natural exponential inertiaweight strategy in particle swarm optimization [C]// Proceedingsof the Six World Congress on Intelligent Control and Automation. Piscataway, NJ, USA: IEEE, 2006: 3672 – 3675.

[6] 窦全胜, 周胜光, 马铭. 粒子群优化的两种改进策略[J]. 计算机研究与发展, 2005, 42(5): 897 – 904.
Dou Q S, Zhou S G, Ma M. Two improvement strategies for particle swarm optimization[J]. Journal of Integrative Plant Biology, 2005, 42(5): 897 – 904.

[7] 黄芳, 樊晓平. 基于岛屿群体模型的并行粒子群优化算[J]. 控制与决策, 2006, 21(2): 175 – 179.
Huang F, Fan X P, Parallel particle swarm optimization based on island group model[J]. Control and Decision, 2006, 21(2): 175 – 179.

[8] 陈民铀,张聪誉,罗辞勇. 自适应进化多目标粒子群优化算法[J]. 控制与决策,2009,24(12):1851 – 1855.
Chen M Y, Zhang C Y, Luo C Y. Adaptive evolutionary multi – objective particle swarm optimization[J]. Control and Decision, 2009, 24(12): 1851 – 1855.

[9] 陶新民,徐晶,杨立标,等. 改进的多种群协同进化微粒子群优化算法[J]. 控制与决策,2009,24(9):1406 – 1411.
Tao X M, Xu J, Yang L B, et al. Multi – species cooperative particle swarm optimization algorithm[J]. Control and Decision, 2009, 24(9): 1406 – 1411.

[10] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.
Wang L. Intelligent optimization algorithm and application[M]. Beijing: Tsinghua University Publishing House, 2001.

[11] 苏辉. 基于搜索空间划分和 sharing 函数的粒子群优化算法[J]. 四川大学学报:自然科学版,2007,44(5):985 – 989.
Su H. An improved particle swarm optimization based on search space division and sharing function [J]. Journal of Sichuan University: Natural Science Edition, 2007, 44(5): 985 – 989.

[12] 徐鸣,沈希,马龙华,等. 一种多目标粒子群改进算法的研究[J]. 控制与决策,2009,24(11):1713 – 1718.
Xu M, Shen X, Ma L H, et al. Research on modified multi – objective particle swarm optimization[J]. Control and Decision, 2009, 24(11): 1713 – 1718.

[13] 陶新民,徐晶,杨立标,等. 改进的多种群协同进化微粒子群优化算法[J]. 控制与决策,2009,24(9):1406 – 1411.
Tao X M, Xu J, Yang L B, et al. Multi – species cooperative particle swarm optimization algorithm [J]. Control and Decision, 2009, 24(9): 1406 – 1411.

[14] 杜继永,张凤鸣. 一种具有初始化功能的自适应惯性权重粒子群算法[J]. 信息与控制,2012,41(2):165 – 169.
Du J Y, Zhang F M. A particle swarm optimization algorithm with initialized adaptive inertia weights[J]. Information and Control, 2012, 41(2): 165 – 169.

[15] Goldberg D E, Richardson J. Genetic algorithm with sharing for multimodal function optimizationg[C]// Proceedings of the Second Internationgal Conference on Genetic Algorithms, 1987: 42 – 49.

[16] Nedjah N, Mourelle L M. Swarm intelligent systems[M]. Berlin, Germany: Springer – Verlag, 2006.

Based on sharing function with dead – zone initialization of particle swarm optimization algorithm

LI Bai – ya[1], JIANG Bai – zhuang[1], DUAN Xiao – lei[2], HUANG Qiang[1]

(1. School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China;
2. Institute of Humanities, Hunan University of Science and Technology, Xiangtan 411201, China)

Abstract: An algorithm of particle swarm optimization (PSO) provided with a dead zone initialization was put forward, according to the problem of PSO in the local optimum, research on particle swarm optimization. The aim was to breakthrough the limitation of local convergence, according to the observation on the MATLAB visual particle trajectories. It analysed the characteristics of particles trapped in local optimum, and run appear stagnation phenomenon in the process of particle swarm, painting for the center of the current local optimal particle "dead – zones", and initializing "dead – zones" in the particle. The results of simulation show that the optimization precision of PSO algorithm are improved, making use of the optimal standard of test functions of the algorithm improved.

Key words: PSO algorithm; local optimum; dead zone initialization; MATLAB