基于遗传算子的粒子群优化算法的比较分析

雷秀娟[1,2],史忠科[2],孙瑰琪[1]
LEI Xiu-juan[1,2],SHI Zhong-ke[2],SUN Gui-qi[1]

1.陕西师范大学 计算机科学学院,西安 710062
2.西北工业大学 自动化学院,西安 710072
1.College of Computer Science,Shaanxi Normal University,Xi'an 710062,China
2.College of Automation,Northwestern Poly-technical University,Xi'an 710072,China
E-mail:xjlei168@163.com

LEI Xiu-juan,SHI Zhong-ke,SUN Gui-qi. Comparison and analysis of particle swarm optimization method based on genetic operator. Computer Engineering and Applications,2008,44(14):65-66.

Abstract: To analyse the performance of improved particle swarm optimization method deeply,three strategies are designed to experiment several standard test functions optimization problem. One of the strategies is linear inertia weight reduction,the second is the PSO with genetic operator,and the third is rejoining the constriction factor based on the second strategy. Through the optimization and simulation of the test functions in MATLAB 7.0 software,the performance of the PSO mixed the genetic operator and constriction factor is best of all.
Key words: Particle Swarm Optimization(PSO);Linearly Decreasing Weight(LDW);genetic operator;constriction factor

摘 要:为了深入分析探讨改进的粒子群优化算法的性能,针对典型的函数优化问题,设计了3种方案:(1)采用线性递减惯性权重的PSO;(2)基于遗传算子的PSO;(3)在方案(2)基础上,加入收缩因子χ。在MATLAB 7.0中对常用的测试函数进行优化仿真,发现当融合遗传算子和收缩因子时,算法性能最优。
关键词:粒子群优化;线性递减惯性权重;遗传算子;收缩因子
DOI:10.3778/j.issn.1002-8331.2008.14.018 **文章编号**:1002-8331(2008)14-0065-02 **文献标识码**:A **中图分类号**:TP18

1 前言

粒子群优化算法通过粒子(潜在的解)在解空间追随最优粒子进行迭代搜寻最优值,为了改善粒子群优化算法摆脱局部极值点的能力,提高算法的收敛速度和精度,目前已提出了很多改进策略。为了比较几种改进策略的优劣,针对典型的函数优化问题,设计了3种方案:(1)采用线性递减惯性权重的PSO;(2)基于遗传算子的PSO;(3)在方案(2)基础上,加入收缩因子χ。通过对常用的测试函数的优化仿真,来比较各种改进方案的效果。

2 算法描述及函数表达式

假设在一个n维的目标搜索空间中,有N个粒子组成一个群体,其中第i个粒子表示一个n维的向量$\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{in})$,$i=1,2,\cdots,N$,每个粒子的位置就是一个潜在的解。第$i$个粒子的"飞行"速度也是一个$n$维的向量,记为$\boldsymbol{v}_i=(v_{i1},v_{i2},\cdots,v_{in})$。记第$i$个粒子迄今为止搜索到的最优位置为$p_i=(p_{i1},p_{i2},\cdots,p_{in})$,整个粒子群迄今为止搜索到的最优位置为$p_g=(p_{g1},p_{g2},\cdots,p_{gn})$。

为了使粒子保持运动惯性,使其有扩展搜索空间的趋势和探索新的区域,Eberhart和Shi[1,2]提出了对速度更新方程加惯性权重w。方程(1)和(2)通常称为标准方程。

$$v_i^{(t+1)}=wv_i^{(t)}+c_1r_1(p_i^{(t)}-x_i^{(t)})+c_2r_2(p_g^{(t)}-x_i^{(t)}) \tag{1}$$

$$x_i^{(t+1)}=x_i^{(t)}+v_i^{(t+1)} \tag{2}$$

其中c_1和c_2是学习因子,为非负常数;w是惯性权重;r_1和r_2是介于[0,1]之间的随机数。迭代中止条件根据具体问题一般选为最大迭代次数等。

几个常用的优化函数表达式如下,其中$x\in[-10,10]$。

$$f(x)=\sum_{d=1}^{D}x_d^{\,2} \tag{3}$$

$$f(x)=\sum_{d=1}^{D}(100(x_d-x_d^{\,2})^2+(x_d-1)^2) \tag{4}$$

$$f(x)=\frac{\sum_{d=1}^{D}(\max(\sum_{d=1}^{D}x_d^{\,2})}{100} \tag{5}$$

$$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{d=1}^{D}x_d^{\,2}}\right)-\exp\left(\frac{1}{D}\sum_{d=1}^{D}\cos(2\pi x_d)\right)+20+e \tag{6}$$

基金项目:教育部科学技术研究重点项目(No.107106);教育部高等学校科技创新工程重大项目培育基金项目。
作者简介:雷秀娟(1975-),女,陕西师范大学副教授,硕士生导师,西北工业大学博士后,研究方向为智能计算、粒子群优化。
收稿日期:2007-08-31 **修回日期**:2007-11-12

3　方案设计及仿真

(1)采用线性递减惯性权重 PSO。

对惯性权重 w 已有诸多改进,对于 w 的取值仍处探讨之中,目前还未形成定论。线性递减权重[3](Linearly Decreasing Weight,LDW)策略实现简单,使用广泛,对大多数问题效果较理想。LDW 表示为:

$$w^k=(w_{ini}-w_{end})\times\frac{(T_{\max}-t)}{T_{\max}}+w_{end} \tag{7}$$

其中,$T_{\max}$ 为最大进化代数,w_{ini} 为初始惯性权重,w_{end} 为进化至最大代数时的惯性权重。典型的取值 $w_{ini}=0.9$,$w_{end}=0.4$。

为便于图示说明,下面的仿真图中横轴表示进化次数,纵轴表示适应值。其适应值变化曲线如图 1 所示。图 1(a)、(b)、(c)和(d)分别代表函数(3)、(4)、(5)和(6)采用线性递减惯性权重 PSO 算法进行优化的适应值变化图。

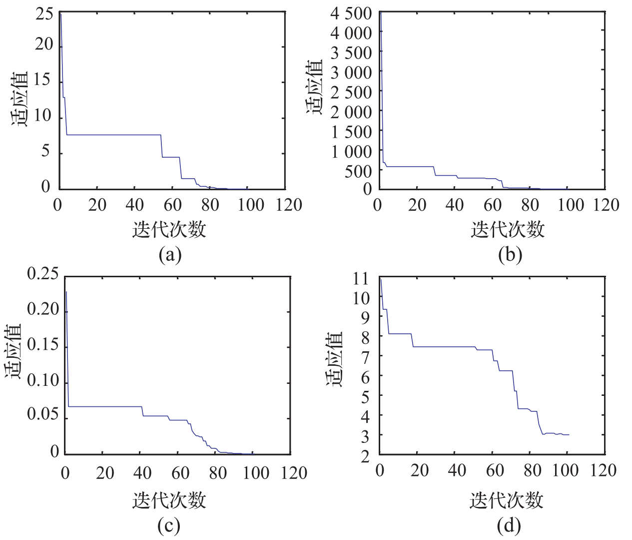

图 1　线性递减惯性权重 PSO 算法适应值变化图

(2)采用基于遗传算子的 PSO。

引入基于简单遗传操作的 PSO 算法[4],该算法的基本思想是利用 PSO 算法每次迭代的最优粒子位置 p_g 及最优粒子速度 v_g 为基础进行变异,然后对变异前后的粒子的分量进行随机交叉操作,从而产生新一代粒子群。相当于在种群更新时进行了再一次筛选,从而减少迭代次数,加快收敛速度。该算法实现简单,大致步骤如下,详见文献[4]:

①确定参数:学习因子 $c_1,c_2,\mu\in[0,2]$ 是实常数,交叉参数 $C\in[0,1]$,种群规模 N 等。

②随机初始化粒子群中粒子的位置和速度。

③按式(1)和式(2)对粒子进行操作。

④随机选择部分粒子按下式进行变异操作,其中 $i1,i2\in\{1,2,\cdots,N\}$ 互不相同亦不同于 i:

$$x_i'\leftarrow p_g+\mu(x_{i1}-x_{i2})$$

$$v_i'\leftarrow v_g+\mu(x_{i1}-x_{i2})$$

⑤交叉操作:

结合 $x_i=(x_{i1},x_{i2},\cdots,x_{in})$ 和 $x_i'=(x_{i1}',x_{i2}',\cdots,x_{in}')$ 产生 $x_i''=(x_{i1}'',x_{i2}'',\cdots,x_{in}'')$;

结合 $v_i=(v_{i1},v_{i2},\cdots,v_{in})$ 和 $v_i'=(v_{i1}',v_{i2}',\cdots,v_{in}')$ 产生 $v_i''=(v_{i1}'',v_{i2}'',\cdots,v_{in}'')$;

⑥形成新一代粒子群,其中 $f(\cdot)$ 是适应度函数,也称适应值:

$$\begin{cases}x_i\leftarrow x_i'' & \text{if}(f(x'')<f(x_i))\\ x_i\leftarrow x_i & \text{if}(f(x_i'')\geqslant f(x_i))\end{cases}$$

$$\begin{cases}v_i\leftarrow v_i'' & \text{if}(f(x_i'')<f(x_i))\\ v_i\leftarrow v_i & \text{if}(f(x_i'')\geqslant f(x_i))\end{cases}$$

⑦若达到最大代数或得到满意解,则优化过程结束,否则返回步骤③。

仿真中算法的初始化参数如下:粒子群规模 20,学习因子 $c_1=c_2=1$,惯性因子 $w=0.6$,交叉概率 $P_c=0.5$,变异概率 $P_m=0.05$,为评价算法的收敛性能,连续运行 20 次所得函数全局最小值的平均值作为算法的衡量指标。其适应值变化曲线如图 2。

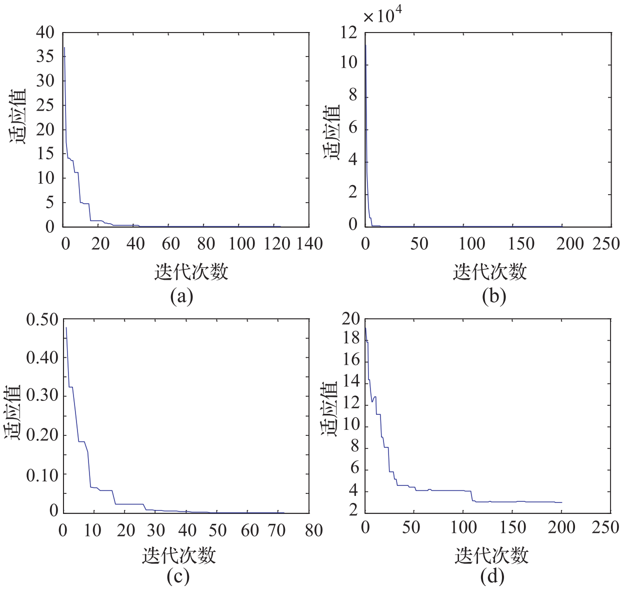

图 2　基于遗传操作的 PSO 算法适应值变化图

图 2(a)、(b)、(c)和(d)则分别代表函数(3)、(4)、(5)和(6)基于遗传操作 PSO 优化后的适应值变化图。比较发现,图 2 较图 1 的优化效果有所提高,对相同的函数进行优化,图 2 的迭代次数有所减少,计算时间也相应地减少了。

(3)在方案(2)基础上,加入收缩因子 χ。

Clerc 和 Eberhart[2,5] 的研究发现压缩因子有助于确保 PSO 算法更快地收敛。它将速度更新方程描述如下:

$$v_i^{(t+1)}=\chi(wv_i^{(t)}+c_1r_1(p_i^{(t)}-x_i^{(t)})+c_2r_2(p_g^{(t)}-x_i^{(t)})) \tag{8}$$

$$\chi=\frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|},\varphi=c_1+c_2,\varphi>4 \tag{9}$$

加入交叉变异操作后,如图 1、2 的对比可见,算法的性能已经有所提高,接着在混合算法的基础上再加入收缩因子 χ,4 个函数优化后的适应值变化图如图 3。

图 3 依次是对上述 4 个函数进行优化后的适应值变化图。从图中可以看出,在遗传 PSO 算法的基础上,再加入收敛因子,会使效果更加明显,更能体现出此混合算法的优越性。

4　结论及分析

设计了三种方案,在 MATLAB7.0 中对常用的测试函数进行优化仿真,发现当融合遗传算子和收缩因子时,算法性能最

(下转 79 页)