求解非线性约束问题的混合粒子群优化算法

张利凤 胡小兵

(重庆大学数学与统计学院 重庆 401331)

摘 要 将处理约束问题的乘子法与改进的粒子群算法相结合,提出了一种求解非线性约束问题的混合粒子群算法。此算法兼顾了粒子群优化算法和乘子法的优点,对迭代过程中出现的不可行粒子,利用乘子法处理后产生可行粒子,然后用改进的粒子群算法来搜索其最优解,这样不仅减小了粒子群算法在寻优过程中陷入局部极小的概率,而且提高了搜索精度。数值试验结果表明提出的新算法具有搜索精度更高、稳定性更强、鲁棒性更好等特点。

关键词 非线性约束优化,粒子群算法,乘子法

中图法分类号 TP301.6 **文献标识码** A

Hybrid Particle Swarm Algorithm of Solving Nonlinear Constraint Optimization Problems

ZHANG Li-feng HU Xiao-bing

(Department of Mathematics and Statistics, Chongqing University, Chongqing 401331, China)

Abstract Combining a multiplier method which deals with constraint problems with improved particle swarm optimization algorithm, a new hybrid particle swarm optimization algorithm was proposed for solving non-linear constraint problems. The new algorithm takes advantage of the particle swarm optimization algorithm and the multiplier method, for the non-available particle appearing in the iterative process, using the multiplier method to produce feasible particle, and then search its optimal value by improved particle swarm optimization. Thus it can not only reduce the probability of falling into local minimum, but also can improve the search accuracy. And the numerical tests show that the proposed new algorithm has the characteristics of validity and searching for more precise particle and better robustness.

Keywords Non-linear constraint optimization, Particle swarm optimization algorithm, Multiplier method

1 引言

粒子群优化(particle swarm optimization,简称PSO)算法由Kennedy和Eberhart[1]于1995年提出,Yuhui Shi和Eberhart[2]于1998年提出了一种改进的粒子群算法,也就是在基本粒子群算法上添加了一个惯性因子ω来控制PSO算法的收敛趋势。作为一种新的智能算法,粒子群算法由于其参数少、结构简单、收敛性快等优点,迅速在智能算法中的应用中占有一席之地。近年来,一些随机的方法,例如遗传算法、差分进化算法、粒子群优化算法等都已经成功应用到约束优化问题中。而粒子群优化算法由于其简单易行,并且收敛速度快,受到国内外学者的广泛关注。利用随机性优化方法来求解约束问题时,处理好约束条件是取得优化效果的关键。文献[3]利用可行基规则来处理约束条件,提出了带有可行基规则的混合PSO算法来求解约束优化问题;文献[4]中把违背函数称为一个目标函数,利用双目标来处理约束条件,提出了自适应的PSO算法来求解约束优化问题;文献[5]中提出了基于外点法的混合粒子群优化算法来求解约束优化问题。本文利用乘子法处理约束条件,把粒子群优化与乘子法相结合,提出了一种新的处理约束优化问题的方法。数值试验表明,新算法具有有效性、通用性、稳健性,是一种有潜力的算法。

2 乘子法求解混合约束优化问题

通常一个约束优化问题可以表示成一个带约束集的线性和非线性规划问题。一般地,一个混合优化问题可以表示如下:

$$\begin{cases}\min f(x)\\ s.t.\quad g_i(x)\leqslant 0, 1\leqslant i\leqslant m\\ \qquad h_j(x)=0, 1\leqslant j\leqslant l\end{cases}$$

其中,$x=(x_1,x_2,\cdots,x_n)$,于是目标函数$f(x):R^n\to R$;$g_i(x):R^n\to R, i=1,2,\cdots,m$;$h_j(x):R^n\to R, j=1,2,\cdots,l$。

对于不等式约束,通过引入松弛变量$z=(z_1,z_2,\cdots,z_m)^T$,将上述模型转化为等式约束的问题:

$$\begin{cases}\min f(x)\\ s.t.\quad g_i(x)+z_i^2=0\quad 1\leqslant i\leqslant m\\ \qquad h_j(x)=0, 1\leqslant j\leqslant l\end{cases}$$

利用等式约束的乘子法,将上述问题转化为无约束的优化问题,于是得到一个增广目标函数:

$$\min_{x,z} F(x,u,c,z)=f(x)+\sum_{i=1}^{l}uh_{ii}(x)+\frac{c}{2}\sum_{i=1}^{l}h_i^2(x)+\frac{1}{2c}\sum_{i=1}^{m}\{[\max(0,z_i+cg_i(x))]^2-z_i^2\}$$

式中,u与c为罚因子,且$c_k\to+\infty$,对于上述极小化问题,我

张利凤(1985—),女,硕士生,主要研究方向为智能计算,E-mail:zhanglifeng0819@yahoo.cn;胡小兵(1975—),男,副教授,硕士生导师,主要研究方向为现代优化技术、机器人控制技术、计算机软件设计。

们先考虑问题$\min_{z} F(x,u,z,c)$的求解来消去 z。实际上，从 z_i 相互独立可知，上式的关于 z 的极小化问题也就是求解问题：

$$\min_{z_i}\left\{u_i(g_i(x)+z_i^2)+\frac{c}{2}(g_i(x)+z_i^2)^2\right\}$$

$$i=1,2,\cdots,m$$

令 $v_i=g_i(x)+z_i^2$，则上式的无约束优化问题可化为二次凸规划求极小的问题：

$$\min_{v_i}\omega_i=u_iv_i+\frac{c}{2}v_i^2$$

$$s.t.\ \ v_i\geqslant g_i(x)$$

令$\frac{d\omega_i}{dv_i}=0$，解得 $v_i=-\frac{u_i}{c}$可得问题的约束最优解 v_i^* 及最优目标值 ω_i^*，即：

$$v_i^*=\max\left\{g_i(x),-\frac{u_i}{c}\right\}$$

$$\omega_i^*=u_iv_i^*+\frac{c}{2}(v_i^*)^2$$

$$=\begin{cases}u_ig_i(x)+\frac{c}{2}(g_i(x))^2, & 若\ v_i^*=g_i(x)\geqslant-\frac{u_i}{c}\\ -\frac{1}{2c}u_i^2, & 若\ v_i^*=-\frac{u_i}{c}>g_i(x)\end{cases}$$

由于：

$$u_ig_i(x)+\frac{c}{2}g_i^2(x)=\frac{1}{2c}[2cu_ig_i(x)+c^2g_i^2(x)]$$

$$=\frac{1}{2c}[(cg_i(x)+u_i)^2-u_i^2]$$

因而，上式的两种情况可以合写成统一的表达式：

$$\omega_i^*=\frac{1}{2c}\{[\max(0,u_i+cg_i(x))]^2-u_i^2\}$$

从而得到新的增广目标函数为：

$$F(x,u,c)=f(x)+\sum_{i=1}^{l}u_ih_i(x)+\frac{c}{2}\sum_{i=1}^{l}h_i^2(x)+\frac{1}{2c}\sum_{i=1}^{m}$$

$$\{[\max(0,u_i+cg_i(x))]^2-u_i^2\}$$

其中，u_i^k的迭代公式为：

$$u_i^{k+1}=\max(0,u_i^{(k)}+c_kg_i(x^{(k)})),c_k\to+\infty$$

(本文中取 $u^0=0,c_k=0.1\times2^k$)，且计算的停止条件可以用约束条件：$v_i^{(k)}=g_i(x^{(k)})+(z_i^{(k)})^2\approx0$ 表示。

即：

$$v^{(k)}=\{\sum_{i=1}^{m}(v_i^*)^2\}^{\frac{1}{2}}$$

$$=\left\{\sum_{i=1}^{m}\left[\max\left(g_i(x^{(k)}),-\frac{u_i^{(k)}}{c_k}\right)\right]^2\right\}^{\frac{1}{2}}<\varepsilon$$

时停止计算。

3 基本粒子群优化算法

1995 年，Kennedy 和 Eberhart[1] 提出基本的 PSO 算法，1998 年 Yuhui Shi 和 Eberhart[2] 提出了一种改进的粒子群算法，后来被称为标准粒子群算法。即在 D 维搜索空间中，第 i 个粒子的位置和速度可分别表示为 $X_i=(x_{i,1},x_{i,2},\cdots,x_{i,d})$ 和 $V_i=[v_{i,1},v_{i,2},\cdots,v_{i,d}]$，其中 $i=1,2,\cdots,N$。通过评价各微粒的目标函数，确定 t 时刻每个微粒所经过的最佳位置(pbest)$P_i=(p_{i,1},p_{i,2},\cdots,p_{i,d})$ 以及群体所发现的最佳位置(gbest)P_g，并按如下公式来更新自己的位置和速度：

$$v_{i,j}(t+1)=\omega v_{i,j}(t)+c_1r_1(p_{i,j}(t)-x_{i,j}(t))+$$

$$c_2r_2(p_{g,j}(t)-x_{i,j}(t)) \quad (1)$$

$$x_{i,j}(t+1)=x_{i,j}(t)+v_{i,j}(t+1),j=1,2,\cdots,d \quad (2)$$

式中，ω 为权惯性因子，c_1 和 c_2 为正的加速常数，r_1 和 r_2 为在 0 和 1 这之间均匀分布的随机数。另外，通过设置微粒的速度区间$[v_{\min},v_{\max}]$和位置区间$[x_{\min},x_{\max}]$，则可以对微粒的移动进行适当的限制。

其中式(1)的右边由 3 部分组成，第一部分反映当前速度的影响，联系粒子当前的状态，起到了平衡全局和局部搜索的能力；第二部分反映认知模式(cognition modal)的影响，即粒子本身记忆的影响，使粒子具有全局搜索能力，避免陷入局部极小；第三部分反映社会模式(social modal)的影响，即群体信息的影响，体现粒子间的信息共享，由于加速因子 c_1,c_2 是认知模式和社会模式的影响因子，因此也可以通过调整 c_1 和 c_2 的值来平衡种群的多样性与收敛性。

4 混合粒子群优化算法

近年来，一些随机的方法，例如遗传算法、差分进化算法、粒子群优化算法等都已经成功应用到约束优化问题中。在用粒子群算法解决约束优化问题时，处理好约束条件是取得优化效果的关键，但由于粒子群算法在寻优过程中容易陷入局部极小，因此我们将粒子群算法和乘子法结合起来解约束优化问题，即在粒子群算法随机搜索的粒子中，如果所搜索的粒子不满足约束条件，则用乘子法计算得到的结果来替换，如果所得结果较优，则更新最优位置，否则继续搜索。由此，基于乘子法的混合粒子群优化算法求解约束优化问题的基本流程如下：

Step1 (初始化)设置当前迭代代数 $t=1$，确定种群的大小 N，搜索空间的维数 D，在整个搜索空间产生 N 个粒子的初始位置 x_i^0 和速度 v_i^0，如果某个粒子 $x_i^0(i=1,2,\cdots,N)$不满足约束条件，则以 x_i^0 为初始点，用乘子法计算得到结果$(x_i^0)^*$替换 x_i^0，并计算适应值。

Step2 将(替换后)第 i 个粒子的最好位置 p_i 设置为该粒子的当前位置，p_g 设置为种群中最佳粒子的位置。

Step3 对粒子群中所有粒子执行如下操作：

Step3.1 根据式(1)和式(2)，更新每个粒子的速度与当前位置。

Step3.2 如果每个粒子 $x_i(t)$不满足约束条件，则以 $x_i(t)$初始点，用乘子法计算得到结果$(x_i(t))^*$替代 $x_i(t)$，并计算每个粒子的适应度。

Step3.3 如果当前粒子的适应度优于自身的 p_i 适应度，则设置 p_i 等于此粒子的当前适应值。

Step3.4 如果当前粒子 i 的适应度优于 p_g 的适应度，则设置 p_g 等于此粒子的当前适应值。

Step4 让 $t=t+1$，返回 Step3，直到获得一个预期的适应值或 t 达到设定的最大迭代次数时停止。

5 数值例子及分析

我们选取常见的测试函数来检验新算法，以下这些例子都是常用的带约束条件的函数优化问题，约束条件有等式约束、不等式约束以及两者的混合，测试函数如下：

例 1

$$\begin{cases}\min f(x)=0.5x_1x_2^{-1}-x_1-5x_2^{-1}\\ s.t.\ g_1(x)=0.01x_2x_3^{-1}+0.01x_2+0.0005x_1x_3\leqslant 1\\ 70\leqslant x_1\leqslant 150;1\leqslant x_2\leqslant 30;0.5\leqslant x_3\leqslant 21\end{cases}$$

例 2

$$\begin{cases}\min f(x)=168x_1x_2+3651.2x_1x_2x_3^{-1}+40000x_4^{-1}\\ s.t.\ g_1(x)=1.0425x_1x_2^{-1}\leqslant 1\\ g_2(x)=0.00035x_1x_2\leqslant 1\\ g_3(x)=1.25x_1^{-1}x_4+41.63x_1^{-1}\leqslant 1\\ 40\leqslant x_1\leqslant 44;40\leqslant x_2\leqslant 45;60\leqslant x_3\leqslant 70;0.1\leqslant x_4\leqslant 1.4\end{cases}$$

例 3

$$\begin{cases}\min f(x)=x_1\\ s.t.\ g_1(x)=-\dfrac{1}{4}x_1+\dfrac{1}{2}x_2-\dfrac{1}{16}8x_1^2\leqslant 1\\ g_2(x)=\dfrac{1}{14}x_1^2+\dfrac{1}{14}x_2^2-\dfrac{3}{7}x_2\leqslant 1\\ 1\leqslant x_1\leqslant 5.5;1\leqslant x_2\leqslant 5.5\end{cases}$$

例 4

$$\begin{cases}\min f(x)=169\times(x_1+x_2+1)/(37x_1+73x_2+13)\times\\ (x_1+2x_2+1)/(63x_1-18x_2+39)\\ s.t.\ g_1(x)=5x_1-3x_2\leqslant 3\\ 1.5\leqslant x_1\leqslant 3;2\leqslant x_2\leqslant 3.5\end{cases}$$

例 5

$$\begin{cases}\min f(x)=(x_1+x_2+x_3)(2x_1+x_2+x_3)\\ \qquad (x_1+2x_2+2x_3)\\ s.t.\ g_1(x)=(x_1+2x_2+x_3)^{1.1}(2x_1+2x_2+x_3)^{1.3}\leqslant 100\\ 1\leqslant x_1\leqslant 3;1\leqslant x_2\leqslant 3\end{cases}$$

例 6

$$\begin{cases}\min f(x)=x_1^2+x_2^2+x_1x_2-14x_1-16x_2+(x_3-10)^2+\\ \qquad 4(x_4-5)^2+(x_5-3)+2(x_6-1)^2+5x_7^2+\\ \qquad 7(x_8-11)^2+2(x_9-10)^2+(x_{10}-7)^2+45\\ s.t.\quad g_1(x)=105-4x_1-5x_2+3x_7-9x_8\geqslant 0\\ g_2(x)=-3(x_1-2)^2-4(x_2-3)^2-2x_3^2+7x_4+120\geqslant 0\\ g_3(x)=-10x_1+8x_2+17x_7-2x_8\geqslant 0\\ g_4(x)=-x_1^2-2(x_2-2)^2+2x_1x_2-14x_5+6x_6\geqslant 0\\ g_5(x)=8x_1-2x_2-5x_9+2x_{10}+12\geqslant 0\\ g_6(x)=-5x_1^2-8x_2-(x_3-6)^2+2x_4+40\geqslant 0\\ g_7(x)=3x_1-6x_2-12(x_9-8)^2+7x_{10}\geqslant 0\\ g_8(x)=-0.5(x_1-8)^2-2(x_2-4)^2-3x_5^2+x_6+30\geqslant 0\\ -10.0\leqslant x_i\leqslant 10.0(i=1,2,\cdots,10)\end{cases}$$

例 7

$$\begin{cases}\min f(x)=-0.5(x_1x_4-x_2x_3+x_3x_9-x_5x_9+\\ \qquad x_5x_8-x_6x_7)\\ s.t.\ g_1(x)=x_3^2+x_4^2-1\leqslant 0\\ g_2(x)=x_9^2-1\leqslant 0\\ g_3(x)=x_5^2+x_6^2-1\leqslant 0\\ g_4(x)=x_1^2+(x_2-x_9)^2-1\leqslant 0\\ g_5(x)=(x_1-x_5)^2+(x_2-x_6)^2-1\leqslant 0\\ g_6(x)=(x_1-x_7)^2+(x_2-x_8)^2-1\leqslant 0\\ g_7(x)=(x_3-x_5)^2+(x_4-x_6)^2-1\leqslant 0\\ g_8(x)=(x_3-x_7)^2+(x_4-x_8)^2-1\leqslant 0\\ g_9(x)=x_7^2+(x_8-x_9)^2-1\leqslant 0\\ g_{10}(x)=x_2x_3-x_1x_4\leqslant 0\\ g_{11}(x)=-x_3x_9\leqslant 0\\ g_{12}(x)=x_5x_9\leqslant 0\\ g_{13}(x)=x_6x_7-x_5x_8\leqslant 0\\ -10\leqslant x_i\leqslant 10(i=1,2,\cdots,8),0\leqslant x_9\leqslant 20\end{cases}$$

例 8

$$\begin{cases}\min f(x)=f_1(x_1)+f_2(x_2)\\ s.t.\ h_1(x)=-x_1+300-\dfrac{x_3x_4}{131.078}\cos(1.48477-x_6)+\\ \qquad \dfrac{0.90798x_3^2}{131.078}\cos(1.47588)=0\\ h_2(x)=-x_2-\dfrac{x_3x_4}{131.078}\cos(1.48477+x_6)+\\ \qquad \dfrac{0.90798x_4^2}{131.078}\cos(1.47588)=0\\ h_3(x)=-x_5-\dfrac{x_3x_4}{131.078}\sin(1.48477+x_6)+\\ \qquad \dfrac{0.90798x_4^2}{131.078}\sin(1.47588)=0\\ h_4(x)=200-\dfrac{x_3x_4}{131.078}\sin(1.48477-x_6)+\\ \qquad \dfrac{0.90798x_3^2}{131.078}\sin(1.47588)=0\\ 0\leqslant x_1\leqslant 400,0\leqslant x_2\leqslant 1000,340\leqslant x_3\leqslant 420,\\ 340\leqslant x_4\leqslant 420,-1000\leqslant x_5\leqslant 1000,0\leqslant x_6\leqslant 0.5236\end{cases}$$

其中

$$f_1(x_1)=\begin{cases}30x_1, & 0\leqslant x_1<300\\ 31x_1, & 300\leqslant x_1<400\end{cases}$$

$$f_2(x_2)=\begin{cases}28x_2, & 0\leqslant x_2<100\\ 29x_2, & 100\leqslant x_2<200\\ 30x_2, & 200\leqslant x_2<1000\end{cases}$$

表 1 列出测试函数运行结果的比较。

表 1 测试函数运行结果的比较

example	Best	Worst	Mean	Std	Time(s)
例 1	−147.666666666667	−147.666666666667	−147.666666666667	0	1.049
例 2	469739.033786623	10000000	1209907.36845103	12887013.0925515	12.487
例 3	1	1	1	0	4.6200
例 4	0.2100624439623	0.212546219980094	0.210931668050469	0.003840593915947	10.7820
例 5	60	60	60	0	2.4500
例 6	25.486030875	30.689930876	27.925715643	248.484545369	138.8478
例 7	−0.57484653390	−0.236353789	−0.46377389271	0.98247542	479.375110
例 8	9178.21534156	9262.9800035	9216.371635224	8.363514	527.8373

为了对比，我们将新算法与文献[5-9]进行比较，文献[5]利用外点法来处理约束条件，把粒子群算法与外点法相结合来解决约束优化问题，文献[6-9]分别采用确定性的全局优化方法求解约束优化问题，本文中的新算法及其比较见表 2。

（下转第 188 页）

2010/07/15/5921.html
[2] Kamvar S. Extrapolation Methods for Accelerating PageRank computations[D]. USA:Stanford University,2003
[3] Haveliwala T H. Efficient Computation of PageRank[R]. Stanford. 1999
[4] Soon I Y,Koh S N. Speech enhancement using 2-D fourier transform[J]. IEEE Transactions on Speech and Audio Processing,2003,11(6):717-724
[5] Haveliwala T H. Topic-sensitive PageRank[C]//Proceedings of the Eleventh International World Wide Web Conference. Hoho Lulu Hawaii,2002
[6] Richardson M,Domingos P. The intelligent surfer:Probabilistic combination of link and content information[J]. PageRank Advances in Neural Information Processing Systems,2002,14:673-680
[7] 黄德才,戚华春. PageRank算法研究[J]. 计算机工程,2006(2)
[8] 白似雪,刘华斌. 基于页面分块模型的PageRank算法研究[J]. 南昌大学学报,2008,6
[9] Diligenti M,Gori M,Maggini M. Web page scoring systems for horizontal and vertical search[C]//The 11th Int'l World Wide Web Conference. Honolulu,Hawaii,USA,2002
[10] 黄德才,戚华春,钱能. 基于主题相似度模型的TS-PageRank算法[J]. 小型微型计算机系统,2007(03)
[11] 王冬,雷景生. 一种基于PageRank的页面排序改进算法[J]. 微电子学与计算机,2009(4)
[12] 张岭,马范援. 加速评估算法:一种提高Web结构挖掘质量的新方法[J]. 计算机研究与发展,2004,41(1):98-103
[13] 王德广,周志刚,梁旭. PageRank算法的分析及其改进[J]. 计算机工程,2010(11)
[14] 戚华春,黄德才,等. 具有时间反馈的PageRank改进算法[J]. 浙江工业大学学报,2005,33(3):272-275
[15] 王小玲,胡平. 基于个人兴趣和反馈技术的PageRank算法研究[M]. 合肥:合肥工业大学出版社,2006(3)
[16] 李凯,赫枫龄,左万利. PageRank-Pro一种改进的网页排序算法[J]. 吉林大学学报,2003(4)

(上接第180页)

表2 新算法与文献[5-9]运行结果的比较表

example	Reference	Time(s)	Optimal value	ε_f
例1	[8]	2	−83.249728406	0
	[6]	<1	−83.249728406	10^{-9}
	This paper	1.049	−147.666666666667	0
例2	[6]	2	623249.87529475	10^{-9}
	[8]	3	623249.876118100	0
	This paper	12.487	469739.033786623	0
例3	[6]	1	1.177124327	0
	[5]	28.6160	1.177643	0
	This paper	4.6200	1	0
例4	[6]	<1	0.2276.9428	0
	[5]	14.9040	0.2107658815	0
	This paper	10.7820	0.2100624439623	10^{-9}
例5	[6]	0	60	0
	[5]	3.3047	60	0
	This paper	2.4500	60	0
例6	[4]	—	24.319	0
	[5]	173.2500	24.8368215	0
	This paper	138.8478	25.486030875	0
例7	[9]		−0.8660254037	0
	[5]	481.9150	−0.841353568	0
	This paper	479.375110	−0.57484653390	10^{-9}
例8	[9]	—	8853.967480648	—
	[5]	481.9150	8931.5135060	—
	This paper	527.8373	9178.21534156	10^{-9}

从表2可以看出,新算法在50次的搜索过程中,基本上找到了最优解,同时对最优解的寻找也提高了一定的搜索精度,特别对于例1,例2,例3,例4,新算法找到的最优解和最优值都比文献[5-9]要好,显示了新的算法具有更高的稳定性和更好的鲁棒性;例6、例7、,例8是高维复杂的约束优化问题,通常被认为是用进化算法很难优化的复杂函数,从表1可以看出新算法基本上找到了最优解,但是我们可以看出新算法在求解高维的约束优化问题时容易陷入局部极值,并且运行的时间也比较长,如何克服这一缺点提高算法的寻优性能也是我们今后进一步研究的课题。

结束语 对解决混合约束优化问题,本文提出了基于乘子法的混合粒子群算法来求解约束优化问题,该方法主要是利用乘子法来处理在算法迭代中出现的不可行粒子,设计出新的混合的粒子群优化算法。通过乘子法的引入来处理约束条件,可以减少适应值函数的复杂性,提高计算结果的可靠性。数值实验显示了新算法的有效性、通用性和稳健性,是一种有潜力的全局优化算法,但对高维函数的优化,效果仍然不是很理想,如何能克服这一缺点,提高算法的寻优性能是我们今后进一步研究的课题。

参 考 文 献

[1] Kennedy J,Eberhart R C. Particle swarm optimization[C]// Proc. IEEE International Conference on Neural Networks. Perth,Australia,IEEE ServiceCenter,Piscataway,NJ,IV:1942-1948
[2] Shi Y,Eberhart R C. A Modified Particle Swarm Optimizer[C]// Proc. of the IEEE CEC. 1998:69-73
[3] He Qie,Wang Ling. A hybrid particle swarm optimization with a feasible-based rule for constrained optimization[J]. Applied Mathematics and Computation,2007,186:1407-1422
[4] Lu Hai-yan,Chen Wei-qi. Self-adaptive velocity particle swarm optimization for solving constrained optimization problems[J]. Journal of global optimization,2008,41(3):427-445
[5] 高岳林,李会荣. 非线性约束优化问题的混合粒子群算法[J]. 计算数学,2010,32(2):135-146
[6] Shen Pei-ping,Jiao Hong-wei. A new rectangle branch-and-pruning approach for generalized geometric programming[J]. Applied Mathematics and Computation,2006,183:1207-1038
[7] Shen Pei-ping,Jiao Hong-wei. Linearization method for a class of multiplicative programming with exponent[J]. Applied Mathematics and Computation,2006,183:328-336
[8] Shen Pei-ping,Zhang Ke-cun. Global optimization of signomial geometric programming using linear relaxation[J]. Applied Mathematics and Computation,2004,150:99-114
[9] Zhang Min, Luo Wei-jian, Wang Xu-fa. Differential evolution with dynamic stochastic selection for constrained optimization [J]. Information Sciences,2008,178:3043-3074
[10] 王凌,刘波. 微粒群优化与调度算法[M]. 北京:清华大学出版社,2008:39-40