文章编号:1000-5862(2015)01-0059-05

# 求解约束优化问题的改进粒子群优化算法

米永强[1],高岳林[1,2*]

(1. 宁夏大学数学计算机学院,宁夏 银川 750021;
2. 北方民族大学信息与系统科学研究所,宁夏 银川 750021)

**摘要**:针对约束优化问题,提出了一种改进的粒子群优化算法. 该算法利用罚函数法将约束优化问题处理为无约束优化问题,并利用可行基规则来更新个体极值和全局极值,使不可行的粒子尽快飞向可行域,显著提高了算法的全局搜索能力. 在标准粒子群算法研究基础上,为了提高粒子群算法求解非线性复杂优化问题的性能,对速度方程和惯性权重做了改进. 数值算例表明,该算法是求解约束优化问题的一种较为有效的全局优化算法.

**关键词**:约束优化问题;粒子群优化;全局优化;罚函数;可行基规则

**中图分类号**:TP 18　　**文献标志码**:A　　**DOI**:10.16357/j.cnki.issn1000-5862.2015.01.11

## 0 引言

约束优化问题(constrained optimization problems,简称 COPs)是日常生活和工程应用等领域经常出现的一类非线性规划问题. 解决该问题通常有确定性和随机性 2 种方法. 确定性算法求解这类问题时受到对目标函数或约束条件可导等要求的限制,故其在处理具有约束性等特点的复杂问题时,易陷入局部最优而很难得到全局最优. 20 世纪50 年代起,人们从生物进化过程中得到启发,提出了遗传算法、粒子群算法、蚁群算法、鱼群算法等来解决复杂的优化问题. 而这些算法在求解约束优化问题中都不受函数连续、可微等条件的限制,因此众多学者已将这些算法应用于约束优化问题中.

J. Kennedy 等[1]提出了 1 种随机搜索算法——粒子群优化算法(Particle Swarm Optimization Algorithm,PSO). 其具有原理简单、易于实现、设置参数少等优点. 因此,它已被广泛地应用于科学和工程等领域. 但是,该算法也有一定的缺陷,如搜索过程中易陷入局部最优解、自身也没有处理约束条件的机制. Hu Xiaohui 等[2]将 PSO 用于求解有约束条件的优化问题,从而,证实了该算法求解约束优化问题的可行性. 受其影响,国内外学者已成功地利用 PSO 来求解约束优化问题[3-8]. 然而,利用智能优化方法求解约束问题的关键在于约束条件的处理. 魏静萱等通过定义不可行度阈值来比较粒子优劣,让性能较优的不可行解微粒保留下来,提出了求解约束优化问题的模糊粒子群算法;高岳林等利用外点法对约束条件进行了处理,提出了混合粒子群算法求解非线性约束优化问题;张利凤等提出基于乘子法的混合粒子群算法求解非线性约束优化问题.

本文在上述研究的基础上,基于罚函数法采用分层次惩罚来处理约束条件,为提高粒子群算法求解非线性约束优化问题的搜索性能,改进了标准粒子群算法的速度方程和惯性权重,从而提出了 1 种改进粒子群优化算法并用于求解约束优化问题. 通过对典型约束函数的测试,数值算例验证该方法是一种较为有效可行的算法.

## 1 问题描述

约束优化问题的一般形式:

$$\min f(x)$$
$$\text{s.t.}\begin{cases}\max\{0,-g_i(x)\},\\ i=1,2,\cdots,l,\\ \max\{|h_i(x)|-\varepsilon,0\},\\ i=l+1,l+2,\cdots,n,\end{cases}\tag{1}$$

**收稿日期**:2014-11-15

**基金项目**:国家自然科学基金(60962006,11161001)和国家民委科研课题(12BFZ005)资助项目.

**通信作者**:高岳林(1963-),男,陕西榆林人,教授,博士生导师,主要从事最优化理论方法及应用、智能计算与智能信息处理研究.

其中 $x=(x_1,x_2,\cdots,x_n)\in F\subseteq S\subseteq \mathbf{R}^n$ 表示决策变量,$g_i(x)$ 和 $h_i(x)$ 分别表示不等式约束和等式约束. 一般来说,等式约束 $h_i(x)=0$ 通常可转化为 $h_i(x)-\varepsilon\leqslant 0$ 和 $-h_i(x)-\varepsilon\leqslant 0$. $\varepsilon$ 是约束条件中等式约束的容忍度,通常取很小的正数.

## 2 标准粒子群优化算法

Shi Yuhui 等[9]在文献[1]给出基本 PSO 算法的基础上,提出了标准粒子群算法,即在 $D$ 维搜索空间中,由 $n$ 个粒子组成 1 个种群,第 $i$ 个粒子在 $D$ 维搜索空间中的位置表示为 $X_i=(x_{i1},x_{i2},\cdots,x_{iD})$,第 $i$ 个粒子的速度可表示为 $V_i=(v_{i1},v_{i2},\cdots,v_{iD})$,第 $i$ 个粒子 $t$ 时刻搜索到的最好位置记为 $P_i=(p_{i1},p_{i2},\cdots,p_{iD})$,整个粒子群搜索到的最好位置记为 $P_g=(p_{g1},p_{g2},\cdots,p_{gD})$,每次迭代的过程中,粒子在第 $d$ 维上按如下公式来更新自己的位置和速度:

$$v_{id}(t+1)=wv_{id}(t)+c_1r_1(p_{id}-x_{id}(t))+c_2r_2(p_{gd}-x_{id}(t)),\tag{2}$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1),\tag{3}$$

其中 $w$ 为惯性权重,其作用是平衡算法全局搜索和局部搜索能力;$c_1$ 和 $c_2$ 为学习因子,用其调整向 $P_i$ 和 $P_g$ 方向飞行的最大步长,通常取2;$r_1$ 和 $r_2$ 为区间[0,1]上均匀分布的随机数.(2)式中第1部分表明粒子当前状态,第 2 部分是粒子的自我学习部分,为避免陷入局部最小值,粒子通过思考自身的位置来调整下一步的位置和速度,第 3 部分为社会学习部分,即粒子相互之间通过信息交流来更新自己的下一步. 防止粒子盲目搜索,一般建议将其速度限制于区间 $[-v_{\max},v_{\max}]$,其中 $v_{\max}$ 是预先给定的最大速度.

## 3 改进的粒子群优化算法求解约束优化问题

### 3.1 罚函数法处理约束条件

约束条件的处理是粒子群优化算法解决约束优化问题的关键环节. 作为处理该问题较为广泛的一种方法——罚函数法,其求解的基本思想是把该约束优化问题中的等式和不等式约束函数通过加权处理再与原目标函数相结合,从而得到新的目标函数,把带有约束的优化问题转化为无约束的优化问题. 本文将(1)式转化为

$$F(x,\sigma_t)=f(x)+\sigma(t)P(x),$$

其中$f(x)$ 为原约束优化问题中的目标函数,$\sigma(t)$ 为惩罚因子,取 $\sigma(t)=t\sqrt{t}$,$P(x)$ 为惩罚项. 其选取参照文献[10],具体形式如下:

$$P(x)=\sum_{i=1}^{m}\theta(e_i(x))e_i(x)^{\alpha},$$

$$e_i(x)=\begin{cases}\max\{0,-g_i(x)\},\\ i=1,2,\cdots,l,\\ \max\{|h_i(x)|-\varepsilon,0\},\\ i=l+1,l+2,\cdots,n,\end{cases}$$

其中 $\theta(e_i(x))$ 是分层次惩罚函数;$\alpha$ 是惩罚强度;$e_i(x)$ 是种群中粒子 $x$ 违反第 $i$ 个约束条件的程度函数;违反约束优化问题(1)中全体约束条件的违反度函数表示为

$$V_{io}(x_j)=\sum_{i=1}^{l}[\max\{0,g_i(x_j)\}]+\sum_{i=l+1}^{n}[|h_i(x_j)|-\varepsilon,0].\tag{4}$$

约束优化问题(1)中粒子 $x_j$ 的适应度函数可表示为

$$f_{itness}(x_j)=f(x_j)+t\sqrt{t}P(x_j).\tag{5}$$

数值实验中将参数 $\alpha$ 和 $\theta(e_i(x))$ 设置如下:

$$\alpha=\begin{cases}1,e_i(x)<1,\\ 2,e_i(x)\geqslant 1,\end{cases}$$

$$\theta(e_i(x))=\begin{cases}10,\ 0<e_i(x)<0.001,\\ 20,\ 0.001\leqslant e_i(x)\leqslant 0.1,\\ 100,0.1<e_i(x)<1,\\ 300,e_i(x)\geqslant 1.\end{cases}$$

### 3.2 速度方程的改进

在标准粒子群算法的速度方程(2)中,学习因子和随机数共同控制着粒子的飞行轨迹,可见,该算法的性能受其加速因子的影响. 为此,本文对速度方程(2)进行了如下改进:

$$v_{id}(t+1)=wv_{id}(t)+\lambda(p_{id}-x_{id}(t))+(1-\lambda)(p_{gd}-x_{id}(t)),\tag{6}$$

其中 $\lambda$ 为调整因子,通过数值实验验证了 $\lambda\in[0,1]$,且 $\lambda$ 从 1 按线性变化递减到 0 效果更佳.

显然,改进的速度方程(6)减少了参数,这样就避免了依靠经验对学习因子的取值进行调整. 改进后使粒子个体的自我学习和社会学习有了紧密联系,即通过线性递减粒子群的局部最优和递增全局最优的影响来控制粒子的速度,从而进一步增强粒子群的全局搜索能力.

### 3.3 非线性递减的惯性权重

在标准的粒子群算法中,粒子当前速度受其先

前速度的影响程度是由惯性权重来决定，文献 [9] 将惯性权重 $w$ 引入 PSO 算法中，通过大量数值实验分析得出较大的惯性权值利于全局搜索，而较小的惯性权值利于局部搜索. 为使算法在全局搜索和局部搜索能力方面达到良好的平衡，文献 [9] 提出了线性递减策略，即

$$w(t) = (w_{\max} - w_{\min})(t_{\max} - t)/t_{\max} + w_{\min},$$

然而，该策略在算法运行过程中处理一些复杂、非线性约束优化问题时效果并不明显. 为更好地求解非线性约束优化问题，作如下调整:

$$w(t) = w_{\max} - (w_{\max} - w_{\min})(t/t_{\max})^3, \quad (7)$$

其中 $t$ 为当前迭代的次数，$t_{\max}$ 为最大的迭代次数，$w_{\max}$ 和 $w_{\min}$ 分别是设置的最大和最小惯性权重，本文取 $w_{\max} = 0.9$，$w_{\min} = 0.4$.

### 3.4　个体极值和全局极值的更新准则

为使 PSO 算法更好地向其个体极值和全局极值调整飞行方向，本文在可行基规则[11]（针对极小化问题）基础上作如下调整，从而更新个体最优解和全局最优解. 预先给定 1 个常数 $\varepsilon$（$\varepsilon$ 为违反约束的容忍度），其随迭代次数增加而减小. 只要满足下列条件之一:

（i）粒子 $i$ 和粒子 $j$ 的约束违反度都不大于 $\varepsilon$，且粒子 $i$ 的适应度值较小;

（ii）粒子 $i$ 的约束违反度不大于 $\varepsilon$，但粒子 $j$ 的约束违反度函数值大于 $\varepsilon$;

（iii）粒子 $i$ 和粒子 $j$ 的约束违反度都大于 $\varepsilon$，且粒子 $i$ 违反约束程度较小;

则粒子 $i$ 优于粒子 $j$，即粒子 $i$ 获胜.

### 3.5　基于约束优化问题的改进粒子群优化算法（IPSO）的步骤

Step 1　设定种群的最大规模 $N$，维数 $D$，调整因子 $\lambda$ 及惯性权重 $w$，并初始化粒子的位置和速度;

Step 2　根据(4)和(5)式依次计算出第 $i$ 个粒子的约束违反度和适应度函数值;

Step 3　令第 $i$ 个粒子的初始位置 $p_i = x_i$，依据可行基规则计算出当前全局最优值 $p_g$;

Step 4　根据(3)，(6)和(7)式，更新各个粒子的当前位置，速度和惯性权重;

Step 5　利用调整的可行基规则对粒子的个体极值 $p_{best}$ 和全局极值 $g_{best}$ 进行更新.

Step 6　让 $t = t + 1$，若满足终止条件，转向 Step7; 否则返回 Step2.

Step 7　输出结果，算法停止.

## 4　数值实验与结果分析

为测试本文提出的 IPSO 求解约束优化问题的性能，选用文献[12]中的 9 个具有代表性的约束函数来检验 IPSO，这些约束函数中的约束条件有等式约束和不等式约束. 其中约束条件中的线性不等式、非线性不等式、线性等式、非线性等式的个数分别用 $L_I$，$N_I$，$L_E$，$N_E$ 表示. 测试函数主要特征如表 1 所示.

表 1　测试函数的主要性质

| 问题 | $n$ | 函数 | $\rho$/% | $L_I$ | $N_I$ | $L_E$ | $N_E$ |
|---|---|---|---|---|---|---|---|
| $g$01 | 13 | 2 次 | 0.000 3 | 9 | 0 | 0 | 0 |
| $g$03 | 10 | 非线性 | 0.002 6 | 0 | 0 | 0 | 1 |
| $g$04 | 5 | 2 次 | 27.007 9 | 0 | 6 | 0 | 0 |
| $g$05 | 4 | 非线性 | 0 | 2 | 0 | 0 | 3 |
| $g$06 | 2 | 非线性 | 0.005 7 | 0 | 2 | 0 | 0 |
| $g$08 | 2 | 非线性 | 0.858 1 | 0 | 2 | 0 | 0 |
| $g$09 | 7 | 非线性 | 0.519 9 | 0 | 4 | 0 | 0 |
| $g$10 | 6 | 线性 | 0.000 5 | 3 | 3 | 0 | 0 |
| $g$11 | 2 | 2 次 | 0.097 3 | 0 | 0 | 0 | 1 |

数值实验在 Matlab 7.0 条件下运行，最大迭代次数 1 000，种群最大规模 130，等式约束的容忍度 $\varepsilon = 1 \times 10^{-5}$. 在同一条件下独立运行 10 次，10 次实验的最好结果（记为 Best）、最优平均值（记为 Mean）及最差结果（记为 Worst）见表 2，其中 NA 表示未公布的数据.

从表 2 可以看出，本文提出的 IPSO 算法对于上述测试函数基本上都找到了最优解. 与 HM 算法[13]相比，除 $g$03 和 $g$11 外，IPSO 算法无论是从最好值、最优平均值，还是最差值，均优于 HM 算法. 与 SAVPSO 算法[14]和 MicroPSO 算法[15]相比，IPSO 算法的求解结果基本上优于这 2 种算法. 与 HDPSO 算法[16]相比，IPSO 算法求解结果大多数也都好于 HDPSO 算法. 因此，通过数值仿真结果表明，IPSO 算法是求解混合约束优化问题的一种较为有效算法.

为证实本文 3.3 节提出的非线性递减惯性权重的显著性，在其他参数设置相同条件下测试了部分有代表性函数，所得结果如表 3 所示.

从表 3 可以看出，在求解非线性约束优化问题时，非线性递减策略的惯性权重优于文献[9]提出的线性递减策略.

表 2 不同算法对标准测试函数的实验结果对比

| 函数及最优值 | 结果 | HM | SAVPSO | MicroPSO | HDPSO | IPSO |
|---|---|---|---|---|---|---|
| $g01$ | Best | −14.788 6 | −15.000 0 | −15.000 1 | −15.000 0 | −15.000 0 |
| (−15.000 0) | Mean | −14.708 2 | −14.715 1 | −13.273 4 | −15.000 0 | −15.000 0 |
| | Worst | −14.615 4 | −12.450 0 | −9.701 2 | −15.000 0 | −14.999 9 |
| $g03$ | Best | 0.999 70 | 1.004 80 | 1.000 40 | 1.000 0 | 0.998 68 |
| (1.000 0) | Mean | 0.998 90 | 1.003 40 | 0.993 60 | 1.000 0 | 0.987 52 |
| | Worst | 0.997 80 | 0.997 60 | 0.667 40 | 1.000 0 | 0.965 41 |
| $g04$ | Best | −30 664.500 0 | −30 665.539 0 | −30 665.539 8 | −30665.5390 | −30 666.074 4 |
| (−30 665.539 0) | Mean | −30 655.300 0 | −30 665.530 0 | −30 665.539 7 | −30665.5300 | −30 666.074 4 |
| | Worst | −30 645.900 0 | −30 665.539 0 | −30 665.533 8 | −30665.5390 | −30 666.074 4 |
| $g05$ | Best | NA | 5 126.484 2 | 5 126.646 7 | 5 131.932 0 | 5 126.567 2 |
| (5 126.498 1) | Mean | NA | 5 202.362 7 | 5 495.238 9 | 5 158.390 0 | 5 129.329 4 |
| | Worst | NA | 5 520.146 7 | 6 272.742 3 | 5 202.173 0 | 5139.625 8 |
| $g06$ | Best | −6 952.100 0 | −6 961.813 9 | −6961.8371 | −6 961.814 0 | −6 961.813 8 |
| (−6 961.813 9) | Mean | −6 342.600 0 | −6 961.813 8 | −6391.8370 | −6 961.814 0 | −6 961.813 9 |
| | Worst | −5 473.900 0 | −6 961.813 8 | −6961.8355 | −6 961.814 0 | −6 961.813 9 |
| $g08$ | Best | −0.095 825 | −0.095 825 | −0.095 825 | −0.095 825 | −0.095 825 |
| (−0.095 825) | Mean | −0.089 157 | −0.095 825 | −0.095 825 | −0.095 825 | −0.095 825 |
| | Worst | −0.029 144 | −0.095 825 | −0.095 825 | −0.095 825 | −0.095 825 |
| $g09$ | Best | 680.910 0 | 680.632 0 | 680.630 7 | 682.563 0 | 680.656 1 |
| (680.630 1) | Mean | 681.160 0 | 680.653 0 | 680.669 1 | 684.337 0 | 680.656 9 |
| | Worst | 683.180 0 | 680.699 0 | 680.637 1 | 683.193 0 | 680.656 7 |
| $g10$ | Best | 7 174.900 0 | 7 054.125 6 | 7 090.452 4 | 7 055.493 0 | 7 077.074 4 |
| (7 049.25) | Mean | 8 163.600 0 | 7 173.266 1 | 7 747.629 8 | 7 317.549 0 | 7 084.584 2 |
| | Worst | 9 659.300 0 | 7 335.240 0 | 10 533.665 8 | 7 583.923 0 | 7 120.979 0 |
| $g11$ | Best | 0.750 0 | 0.749 0 | 0.749 9 | 0.750 9 | 0.750 0 |
| (0.750 0) | Mean | 0.750 0 | 0.749 0 | 0.767 3 | 0.772 1 | 0.750 3 |
| | Worst | 0.750 0 | 0.749 0 | 0.992 5 | 0.832 8 | 0.750 7 |

表 3 $w$ 对算法的影响

| 函数 | 结果 | 线性变化 | 非线性变化 |
|---|---|---|---|
| $g01$ | Best | −14.991 7 | −15.000 0 |
| | Mean | −14.962 6 | −15.000 0 |
| | Worst | −14.935 5 | −14.999 9 |
| $g06$ | Best | −6 961.441 1 | −6 961.813 8 |
| | Mean | −6 960.694 2 | −6 961.813 9 |
| | Worst | −6 960.036 9 | −6 961.813 9 |
| $g08$ | Best | −0.095 825 | −0.095 825 |
| | Mean | −0.095 799 | −0.095 825 |
| | Worst | −0.095 756 | −0.095 825 |
| $g11$ | Best | 0.750 0 | 0.750 0 |
| | Mean | 0.750 6 | 0.750 3 |
| | Worst | 0.752 5 | 0.750 7 |

## 5 结束语

针对混合约束优化问题,本文提出了一种改进的粒子群优化算法.该算法引入罚函数法来处理约束条件,考虑到标准粒子群算法中粒子的自我认知和社会认知之间的联系,对速度方程进行了改进,提高了粒子群算法求解非线性约束优化问题的性能.同时,调整了惯性权重,使其克服了原有的惯性权重难以适应算法运行中的复杂、非线性变化,为了使算法的全局搜索能力显著提高,在可行基规则基础上,依据粒子违反约束条件的程度来更新个体极值和全局极值,使不可行的粒子尽快飞向可行域.数值实验

结果表明，本文提出的算法是求解约束优化问题的一种较为有效的算法.

## 6 参考文献


[1] Kennedy J，Eberhart R. Particle swarm optimization [C] // In Proceedings of the IEEE Intentional Joint Conference on Neural Networks. Perth WA: IEEE Press，1995: 1942-1948.

[2] Hu Xiaohui，Eberhart R. Solving constrained nonlinear optimization problems with particle swarm optimization [C] // Proc of the 6th World Multi conference on Systemics. Orlando: Cybernetics and Informatics，2002: 203-206.

[3] 董颖，唐加福，许宝栋，等. 一种求解非线性规划问题的混合粒子群优化算法 [J]. 东北大学学报: 自然科学版，2003，24(12): 1141-1144.

[4] Mazhoud I，Hadj-Hamou K，Bigeon J，et al. Particle swarm optimization for engineering problems: A new constraint-handling mechanism [J]. Engineering Application of Artificial Intelligence，2013，26(4): 1263-1273.

[5] Elsayed S M，Sarker R A，Mezura-Montes E. Self-adaptive mix of particle swarm methodologies for constrained optimization [J]. Information Science，2014，277: 216-233.

[6] 魏静萱，王宇平. 一种解决约束优化问题的模糊粒子群算法 [J]. 电子与信息学报，2008，30(5): 1218-1221.

[7] 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.

[8] 张利凤，胡小兵. 求解非线性约束问题的混合粒子群优化算法 [J]. 计算机科学，2011，38(10): 178-180.

[9] Shi Yuhui，Eberhart R. A modified particle swarm optimizer [C] // In Proceedings of the IEEE Congress on Evolutionary Computation. Anchorage，AK: IEEE Press. 1998: 69-73.

[10] Yang Jinn Moon，Chen Yingping，Horng J T，et al. Applying family competition to evolution strategies for constrained optimization [J]. Lecture Notes in Computer Science，1997，1213: 201-211.

[11] Deb K. An efficient constraint handing method for genetic algorithms [J]. Methods for Applied Mechanical Engineering，2000，186(2): 311-388.

[12] Runarsson T P，Yao Xin. Stochastic ranking for constrained evolutionary optimization [J]. IEEE Transactions on Evolutionary Computation，2000，4(3): 248-254.

[13] Koziel S，Michalewicz Z. Evolutionary algorithms，Hmomorphous mapping，and constrained parameter optimization [J]. Evolutionary Computation，1999，7(1): 19-44.

[14] Caponetto R，Fortuna L，Fazzino S，et al. Chaotic sequences to improve the performance of evolutionary algorithms [J]. IEEE Transactions on Evolutionary Computation，2003，7(3): 289-304.

[15] Cabrera J C F. Handling constraints in particle swarm optimization using a small population size [C] // Proc 6th MICAI Aguascalientes. Mexico: IEEE Press，2007: 41-51.

[16] 刘衍民. 一种求解约束优化问题的混合粒子群算法 [J]. 清华大学学报: 自然科学版，2013，53(2): 242-246.


# The Improved Particle Swarm Optimization Algorithm for Solving Constrained Optimization Problems


MI Yongqiang[1]，GAO Yuelin[1,2*]

(1. School of Mathematics and Computer Science，Ningxia University，Yinchuan Ningxia 750021，China;

2. Research Institute of Information and System Science，Beifang University of Nationalities，Yinchuan Ningxia 750021，China)



**Abstract:** For constrained optimization problems，an improved particle swarm optimization algorithm is proposed. The algorithm uses the penalty function method to handle the problem of constrained optimization to unconstrained optimization problems，and the feasibility based rule is used to update individual optimal and global extremum，it makes the infeasible particles to fly to the feasible region as soon as possible，the global search ability of the algorithm is improved significantly. On the basis of the particle swarm algorithm research，the velocity equation and inertia weight is improved so that to improve the performance of particle swarm optimization algorithm to solve the complicated and nonlinear optimization problem. Numerical experiments show that the proposed algorithm is a global optimization algorithm with higher efficiency for unconstrained optimization.



**Key words:** constrained optimization problems; particle swarm optimization; global optimization; penalty function; feasibility based rule



(责任编辑: 曾剑锋)