# 解多目标约束问题的改进 MaxMin-PSO 算法

黄圣杰,罗 琦,佟金颖
HUANG Sheng-jie,LUO Qi,TONG Jin-ying

南京信息工程大学 信息与控制学院,南京 210044
Department of Information and Communication,Nanjing University of Science and Technology,Nanjing 210044,China
E-mail:b1126@hi2000.com

**HUANG Sheng-jie,LUO Qi,TONG Jin-ying.Improved MaxMin Particle Swarm Optimization for solving multiobjective constrained optimization problems.Computer Engineering and Applications,2008,44(15):48-50.**

**Abstract:** In this paper,Max-Min fitness function and penalty function are combined together,and a practical and effective particle swarm optimization algorithm is proposed to solve multi-objectives constrained optimization problems.Non-inferior solutions are replaced according to the idea of cluster and compare.The method of selecting globally optimal solution from non-inferior solutions in turn is adopted instead of the ancient method.The experimental results show that the modified MaxMin-PSO algorithm converges more quickly and efficiently to Pareto solutions and achieve a well distribution.It also restrains the radiation of low dimension multi-objectives constrained functions.

**Key words:** particle swarm algorithm;MaxMin function;turn list;punish function

**摘 要:**将最大最小化适应度函数与罚函数相结合,提出了一种实用有效求解多目标约束优化问题的粒子群算法。采用归类和比较的思想进行替换非劣解;改变以往全局最优值的选取方法,而采用轮序方式从非劣解中获取。实验证明改进的 MaxMin-PSO 算法能更加有效的逼近 Pareto 解,收敛速度更快,分布更均匀,且能很好的抑制低维多目标约束问题的发散现象。

**关键词:**粒子群算法;最大最小适应函数;轮序;罚函数
**DOI:**10.3778/j.issn.1002-8331.2008.15.014 **文章编号:**1002-8331(2008)15-0048-03 **文献标识码:**A **中图分类号:**TP301.6

## 1 引言

现实世界经常遇到一些优化问题,而这些优化问题往往存在着两个或者更多的优化目标。多个优化目标之间存在着相互的联系,但又不能同时达到最优,这个时候就需要尽可能的使多个目标都向希望的目标值逼近。如果对于整个过程加上某种限制,那么就把这种优化问题称为多目标约束优化问题(MOP:Multiobjective Restriction Optimization Problem)。1896 年Pareto 首次从数学角度提出了多目标最优决策问题,引入 Pareto 最优解的概念,从此多目标约束优化问题的求解出现了一个新的阶段。此时问题表述为求解一组可能出现的解,将这组解叫做 Pareto 解。但是由于当时科技的限制,对于 Pareto 的求解比较困难,因此人们寻求新的方法。从 20 世纪 50 年代末到 20 世纪 90 年代初,多目标优化方法和理论的研究先后出现了加权和法、目标规划法、$\varepsilon$ 约束法等基于权重的多目标优化方法,但均有其局限性。

演化算法的产生为上述问题的解决提供了新的思路。1956 年 Rosenberg 在其博士论文中第一次提出了可以用基于遗传的搜索算法求解多目标约束优化问题,1985 年 Schaffer 的 VEGA 算法第一次将其实现[1]。到 20 世纪 90 年代,演化算法成为众多学者研究的热点。先后涌现出了大批优秀的论文和学术成果,如 Pure Pareto Ranking、MOGA、NSGA、NPGA 以及出现的其改进方法 PAES、SPEA、NSGA2。粒子群优化算法(Particle Swarm Optimization,PSO)是由 Kennedy 和 Eberhart 于 1995 年提出的一种优化算法[2],由于它实现简单并且得到的效果较好,受到很多学者的关注。其在多目标优化问题上,也涌现了一批优秀成果。如:K.E.Parsopoulos 和 M.N.Vrahatis 的 MOPSO 算法[3];MaxMin 适应度函数与粒子群算法相结合用于求解多目标优化问题[4],该方法在非劣最优解的求取上不同于以往的其他求解方法,通过最大最小的思想选择非劣最优解,在某些问题的求解上取得了较好的效果。

本文将最大最小化思想与罚函数相结合,提出了一种改进的 MaxMin-PSO 算法。并对最大最小适应度函数在非劣解处理上进行了改进,使其能更有效,更快的收敛到 Pareto 解。

## 2 多目标约束优化问题

在科学实践和社会生产中经常遇到的多目标约束优化问题,一般可表述为如下形式:

$$\begin{aligned}&\text{Min}\ \ Y=f(x)=(f_1(x),f_2(x),\cdots,f_n(x))\\&\text{s.t}\ \ \ g_i(x)\leqslant 0\\&\qquad x=(x_1,x_2,\cdots,x_n)\in R^n\end{aligned}\tag{1}$$

其中 $\boldsymbol{Y}$ 是目标向量,而 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in R^n$ 是状态向量,$g_i(x)\leqslant 0$ 是约束条件。

**作者简介:**黄圣杰(1982-),男,硕士,主要研究方向:智能算法与高等控制;罗琦(1958-),男,博导,教授,主要研究方向:随机分析;佟金颖(1983-),女,硕士,主要研究方向:智能算法与高等控制。
**收稿日期:**2007-09-04 **修回日期:**2007-11-28

多目标优化问题在现实中的应用研究非常广泛[5]。传统求解的方法往往是通过加权的方法将多目标转化为单目标,然后进行求解。但是这样存在着局限性:首先由于目标向量性质不一样,权值的选取没有可比较性;其次通过加权得到的目标值往往是片面的,不能全局反映情况;第三,通过加权的方法的到的值,带有主观性,不同的人权值选择不同。

因此在大多数情况下,多目标约束优化问题往往没有唯一最优解,而是有一组非劣最优解—Pareto 解。用如下定义:

**定义 1** 一个向量 $\boldsymbol{u}=(u_1,u_2,\cdots,u_m)$ 称为非劣于(Pareto 优于)向量 $\boldsymbol{v}=(v_1,v_2,\cdots,v_m)$,当且仅当对于 $\forall i\in\{1,2,\cdots,m\},u_i\leqslant v_i\wedge(\exists i\in\{1,2,\cdots,m\},u_i\leqslant v_i)$。

**定义 2** 一个解 $x_u\in R^n$ 称为问题(1)的最优解,当且仅当不存在 $x_v\in R^n$ 使得 $V=f(x_v)$ 非劣于 $U=f(x_u)$[6]。

Pareto 最优概念是建立在集合论基础上对多目标解的一种向量评估方式。对于科学实践和社会生产具有较大的指导作用。

## 3 粒子群算法

粒子群优化(PSO)算法是一种新兴的优化技术,其思想来源于人工生命和演化计算理论。PSO 通过粒子追随自已找到最好的解和整个群体找到的最好解来完成优化[7]。最初是 Eberhart 和 Kennedy 对于鸟群捕食行为的研究,建立了其模型,随后从这个模型中得到启发,将其用到了最优控制中来。PSO 算法虽然也是基于群体的,但是与传统的遗传算法不同,它通过个体之间的协作来寻找最优解。在优化问题的搜索空间中,每一个粒子就相当于一只鸟,而目标最优解就是食物。换句话说就是鸟被抽象成了没有质量和体积的微粒。每个粒子存在一个当前位置 $X$,而这个 $X$ 值对于目标函数有一个目标值。每个粒子记录了经过的位置的最好的目标值,而它相对应的位置称为 pbest(自身的飞行经验),而所有粒子的 pbest 中目标值最大的那个称为 gbest(全体粒子的经验)。每个粒子根据这两个值不断变化着自身的位置,飞向目标最优解。具体过程用模型(标准粒子群算法)表示如下:

$$v_{ij}(t+1)=wv_{ij}(t)+c_1r_{1j}(t)(p_{gi}(t)-x_{ij}(t)) \tag{2}$$

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

其中:下标"$j$"表示微粒的第 $j$ 维,"$i$"表示微粒 $i$,$t$ 表示第 $t$ 代,$c_1,c_2$ 为加速常数,通常在 0~2 间取值,$r_1,r_2$ 是两个相互独立的随机函数。$p_{ij}(t)$ 是个体最优位置,而 $p_{gi}(t)$ 称为全局最好位置,对应于全局最好粒子所处的位置。

标准粒子群算法的执行过程如下:

(1)初始化一群粒子,随机产生每个粒子的位置和速度;

(2)评价没个粒子的适应度(约束优化问题的处理中,适应度是目标函数值和罚函数共同作用得到的);

(3)对于每个粒子,将其适应度值与自身 *pbest* 作比较,如果当前值好于 *pbest*,则将当前值设为 *pbest*;

(4)如果在所有 *pbest* 中有优于当前 *gbest* 的,则将该 *pbest* 取代原来 *gbest*;

(5)根据式(2)、(3)改变每个粒子的速度和位置。

(6)如果未达到终止条件(最优目标值或者预先设定的最大代数),则返回(2)。

## 4 改进的 MaxMin-PSO 算法求解多目标约束优化问题

将最大最小化思想用于粒子群求解多目标优化问题[4],对于某些问题,其效果非常明显。也有人对其进行改进,如文献[8]。本文从其他角度进行改进,对约束条件采用最大最小适应值与罚函数相结合的方法获取非劣解;通过归类和轮序的方法从非劣解中选取全局最优解。改进后的 MaxMin-PSO 算法求解多目标约束优化问题,能很好的逼近 Pareto 解,防止局部发散,粒子的分部也比较均匀。

**定义 3** MaxMin 函数:假设有 $m$ 个目标值,$n$ 个粒子。首先每个粒子与自身以外的所有粒子进行比较,比较项目就是 $m$ 个目标值。在得到的 $m$ 个差值间选取其中最小的一个。然后,这个粒子与其他 $n$-1 个粒子的比较值中选取一个最大的作为该粒子的适应值。如果适应值小于 0(求最小值为例),那么就是说明了该粒子与其他粒子的比较中,均有某一目标值优于其他粒子。那么对于这 $n$ 个粒子,该粒子可以认为是非劣最优解。

$$f_{\max\min}=\max_{j=1,\cdots,N;u\neq v}\{\min_{i=1,\cdots,m}\{f_i(u)-f_i(v)\}\} \tag{4}$$

**定义 4** 罚函数:罚函数基本思想是通过序列无约束最小化技术(Sequential Unconstrained Minimization Technique),将约束优化问题转化为一系列无约束优化问题进行求解。具体做法是,根据约束条件的特点,构造特殊的惩罚函数。然后,将惩罚函数加到目标函数中去,将约束优化问题转化为无约束优化问题的求解。这种惩罚策略在无约束优化问题求解过程中,对于那些违反约束条件的迭代点给予很大的目标函数值(对于求极小值问题),迫使一系列的无约束问题的极小点或者无限制的靠近可行域,或者一直保持在可行域内移动。直到迭代点收敛到满足条件的最优目标值。通常构造的广义目标函数可表示为:

$$F(X)=\begin{cases}f(x),x\in\Omega\\ f(x)+P(x),x\notin\Omega\end{cases} \tag{5}$$

其中 $f(x)$ 代表原来的目标函数;$P(x)=h(k)H(x)$ 称为惩罚项,$h(k)$ 表示惩罚力度,$H(x)$ 为惩罚因子。

**定义 5** MaxMin 与罚函数结合:这部分主要处理约束条件。当粒子违法约束条件的时候,用罚函数给以惩罚,在这里将惩罚和 MaxMin 相结合,提出了一种 MaxMin 惩罚:

$$F_{\max\min}=f_{\max\min}+h(a)\exp(f_{\max\min}),h(a)=k*(\frac{i}{a})^2 \tag{6}$$

其中 $a$ 为约束个数,$i$ 为违反个数,$k$ 为常量。

**定义 6** 轮序:以往用 MaxMin 算法求解多目标问题时往往用适应值最小的那个点作为全局最优值。但是如果初始粒子中存在着局部最优点,而这个点在一开始时往往会适应值最小。这个时候粒子就会像局部最优解收敛,得到的效果不理想。这里选用轮序的方式来选取全局最优值,很好的抑制了这种现象。当选取某一个粒子作为全局最优解,粒子进行更新后,如果适应值最小仍是最初那个粒子,那么全局最优粒子选择为其他粒子中适应值最小的那个。

**定义 7** 归类:归类与聚类有所区别,聚类通过粒子与粒子间的距离来确定是否将粒子进行合并。而这里的归类,将靠近的一些粒子看作一个群体,局部分离非劣解。当某个局部的密度大于设定值时,对于这个局部,选择部分解继续保留,其它的舍去。这样做对于全局影响不大,而又很好的防止了粒子的不均匀分布。密度阀值 $\delta$ 公式如下:

$$\delta=\frac{sum_x}{sum} \tag{7}$$

其中 $sum_x$ 是指某一部分的粒子个数,而 $sum$ 是非劣解集中所有粒子个数。

具体算法如下:

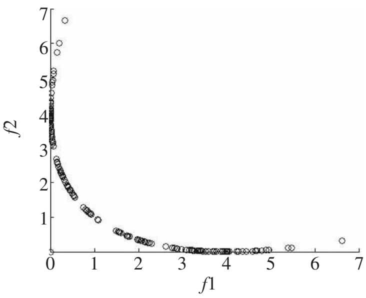

图 1 没有改进的算法实现测试函数 1

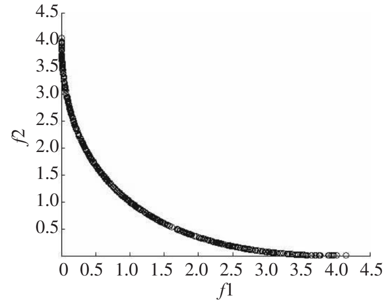

图 2 改进后的算法实现测试函数 1

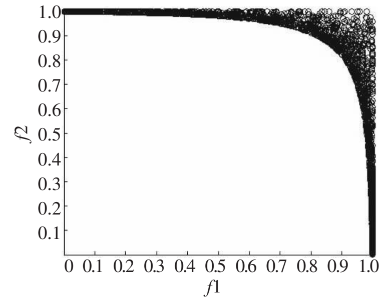

图 3 没有改进的算法实现测试函数 2

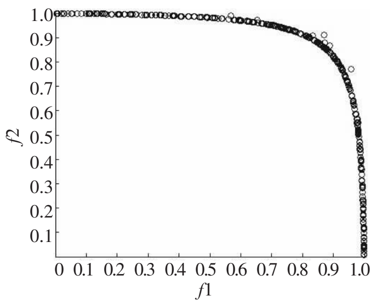

图 4 改进后的算法实现测试函数 2

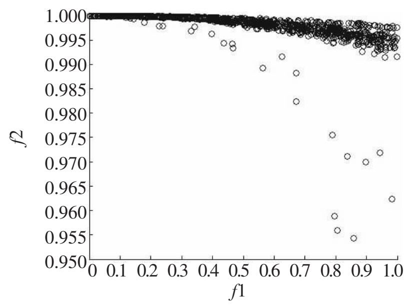

图 5 没有改进的算法实现测试函数 3

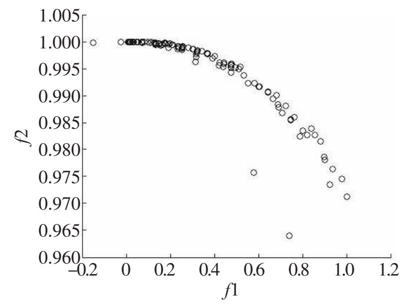

图 6 改进后算法实现测试函数 3

(1)初始化粒子群:随机产生 $N$ 个粒子,对于每一个粒子给以初始速度和位置。根据 MaxMin 与罚函数结合方法计算每个粒子的适应值,将每个适应值作为该粒子的最优个体值,将适应值小于 0 的个体存储于非劣解集。并在非劣解集中按照轮序的方式选取全局最优解;

(2)非劣解集优化:用 MaxMin 方法对非劣解集群体进行一次搜索,防止出现非支配解。用归类的方法将非劣解集划分 $m$ 份,设定密度最大值。当密度大于设定值时,随机选取其中一部分粒子保留;

(3)粒子的更新:按照粒子更新公式(2)、(3)对粒子进行更新;

(4)适应值计算:根据 MaxMin 与罚函数结合方法计算每个粒子的适应值;

(5)个体最优值与全局最优值更新:将每个粒子的适应值与对应的最优个体值进行比较,选取适应值小的作为更新后的个体最优值;将适应值小于 0 的个体存储于非劣解集,并在非劣解集中按照轮序的方式选取全局最优解;

(6)判断是否结束要求:是,则结束;否则,转到(2)。

## 5 数值实验

(1)实验环境:

CPU:PentiumIV 1.7 GHz;内存:256 MB;操作系统:Windows XP;程序编码:Matlab 编译实现。

(2)实验数据:

①测试函数 1

SCH 函数:

$$\text{Min } f(x)=(f_1(x),f_2(x))$$

$$\text{s.t. } f_1(x)=x^2, f_2=(x-2)^2, x\in[-5,7] \tag{8}$$

粒子群参数选取如下:$c1=c2=1.2$,$w$ 由 1.0 线性递减至 0.2,粒子数 20,迭代次数 100。

②测试函数 2

二维凸函数:

$$\text{Min } f(x)=(f_1(x),f_2(x))$$

$$\begin{aligned}\text{s.t. } & f_1(x)=1-\exp(-(x_1-1)^2-(x_2+1)^2)\\ & f_2(x)=1-\exp(-(x_1+1)^2-(x_2-1)^2)\\ & x_1,x_2\in[-10,10]\end{aligned} \tag{9}$$

粒子群参数选取如下:$c1=c2=1.2$,$w$ 由 1.0 线性递减至 0.2,粒子数 50,迭代次数 200。

③测试函数 3

高维 ZDT 函数:

$$\text{Min } f(x)=(f_1(x),f_2(x))$$

$$\begin{aligned}\text{s.t. } & f_1(x)=x_1, f_2(x)=1-\left[\frac{f_1}{g(x)}\right]^2\\ & g(x)=1+\left[\frac{9}{n\sum_{i=2}^{n}x_i}\right],\\ & x_i\in[0,1], n=30\end{aligned} \tag{10}$$

粒子群参数选取如下:$c1=c2=1.2$,$w$ 由 1.0 线性递减至 0.2,粒子数 50,迭代次数 200。

(3)实验结果与评价

测试函数 1:用原始算法实现如图 1,改进后的算法实现如图 2。图中可以看出,改进后的方法粒子的分布更均匀,并且粒子的数量也产生了质的变化。

测试函数 2:用原始算法实现如图 3,改进后的算法实现如图 4。从图上可以明显地看出改进后的方法收敛性更强,而原始算法的实现则容易局部发散。从均匀分布角度来说,改进后的算法分部也比较均匀。

测试函数 3:图 5 是原始算法实现,图 6 是改进后算法实现。可以看出,两种方法相比较,改进方法逼近效果更明显,但是两种方法均不能完全逼近。可见该方法在高维函数的多目标问题上还有待于进一步的研究。

## 6 结束语

粒子群算法用于求解多目标约束优化问题,关键仍然在于如何确定全局最优和局部最优。本文通过将 MaxMin 与罚函数

(下转 60 页)