

# 基于改进粒子群优化算法的约束多目标优化

阳春华, 莫志勋, 李勇刚

(中南大学信息科学与工程学院, 长沙 410083)

**摘要:** 针对约束多目标优化问题, 提出一种改进的粒子群优化算法, 采用距离量度和自适应惩罚函数相结合的约束处理技术, 通过可行解比例有效均衡目标函数和约束条件, 提高算法的边界搜索能力。定义新的 k 最近邻聚集密度, 保持解集分布性, 并将聚集密度和轮盘赌选择相结合选取全局最优粒子。仿真结果表明, 该算法在 Pareto 解集均匀性及逼近性方面均具有优势。

**关键词:** 约束多目标优化; 距离量度; 自适应惩罚; 粒子群优化算法

## Constrained Multi-objective Optimization Based on Improved Particle Swarm Optimization Algorithm

YANG Chun-hua, MO Zhi-xun, LI Yong-gang

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

**【Abstract】** An improved constrained multi-objective Particle Swarm Optimization(PSO) algorithm which adopts constraint handling technology based on distance measures and adaptive penalty functions is proposed. The proportion of feasible solutions is used to keep the balance between objective function and constraints and improve the boundary searching capability of the algorithm. A new k nearest neighbors crowding density is defined to maintain the diversity of solutions. It selects the global optimal particle through the combination of crowding density and roulette selection. Experimental results demonstrate that the algorithm is efficient for solving constrained multi-objective optimization problems.

**【Key words】** constrained multi-objective optimization; distance measure; adaptive penalty; Particle Swarm Optimization(PSO) algorithm

### 1 概述

近年来,用进化算法求解约束优化问题已成为研究热点,但是进化算法本身是一种无约束的搜索技术,因此,设计有效的约束处理机制是求解约束多目标优化问题的关键。目前广泛使用的约束处理方法是惩罚函数法,此外,将约束条件作为一个或多个目标处理的多目标法越来越受到重视<sup>[1]</sup>,但该方法往往使用区分解是否可行的比较准则<sup>[2]</sup>,不利于边界搜索。上述大多数求解约束优化问题的进化算法都是用遗传算法(GA)实现的,但一般 GA 编码比较复杂、效率较低。文献[3]提出的粒子群优化(Particle Swarm Optimization, PSO)算法具有很强的全局搜索能力、更快的收敛速度,在很多情况下比遗传算法表现更好,获得了广泛而成功的应用。

本文在文献[4]约束处理技术基础上,结合 PSO 算法的优点,提出一种新的基于 k 最近邻的个体聚集密度估计指标,并据此进行外部档案集的维护和全局领导粒子选取,充分利用不可行域的有用信息,进一步提高对可行域进行精确偏向搜索的能力。

### 2 约束多目标优化问题及标准粒子群优化算法

不失一般性,以最小化问题为例,约束多目标优化问题可描述如下:

$$\begin{aligned} & \text{Minimize: } f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ & \text{s. t.: } g_j(x) \leq 0, j = 1, 2, \dots, q \\ & h_j(x) = 0, j = q + 1, 2, \dots, m \\ & x = (x_1, x_2, \dots, x_n), \\ & x_i^{\min} \leq x_i \leq x_i^{\max}, i = 1, 2, \dots, n \end{aligned} \quad (1)$$

其中,  $x \in \Omega$  为决策向量,  $\Omega$  为可行域;  $S \subseteq R^n$  为决策空间;  $x_i^{\min}$  和  $x_i^{\max}$  为每维决策变量的上下限,  $i=1, 2, \dots, n$ ,  $g_j(x)$ 、 $h_j(x)$  均为  $R^n$  上的  $n$  元函数;  $f(x)$  为目标函数;  $g_j(x) \leq 0$  为第  $j$  个不等式约束条件;  $h_j(x) = 0$  为第  $j$  个等式约束条件。

PSO 算法是近年来被广泛关注和研究的一种全局智能优化算法。设在  $D$  维空间中,种群由  $N$  个粒子组成,则其第  $i$  个粒子的位置和速度为:

$$\begin{aligned} v_{id}(t+1) &= \omega v_{id}(t) + c_1 r_1 (pBest_{id}(t) - x_{id}(t)) + \\ & c_2 r_2 (gBest_{id}(t) - x_{id}(t)) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \quad (2)$$

其中,  $t$  为进化代数;  $x_{id}, v_{id}$  分别为粒子  $i$  的  $d$  维位置和速度分量;  $c_1$  和  $c_2$  为学习因子;  $r_1$  和  $r_2$  是  $[0,1]$  之间的随机数;  $pBest$  为个体在搜索过程中找到的历史最优位置;  $gBest$  为种群在搜索过程中找到的全局最优位置;  $\omega$  为惯性权重,用于控制当前速度对更新后速度的影响程度。

### 3 约束多目标粒子群优化算法

用 PSO 算法求解约束多目标优化问题,关键是设计合理

**基金项目:** 国家自然科学基金资助项目“数据驱动的多相交互冶金过程能耗优化方法研究及应用”(60874069);教育部高校博士点专项科研基金资助项目“基于数据驱动的锌电解过程能耗优化方法研究”(200805331104)

**作者简介:** 阳春华(1965 -),女,教授、博士、博士生导师,主研方向:复杂工业过程建模、仿真与优化,智能自动化控制;莫志勋,硕士研究生;李勇刚,副教授

**收稿日期:** 2010-04-13 **E-mail:** mzx165@126.com

的约束处理技术,维护外部档案,选取合适的全局和局部极值,使算法既能快速靠近可行域,又能在可行域中找到分布良好的 Pareto 前沿。本文将目标函数与个体的约束违反程度值相结合,定义距离量度和自适应惩罚函数值,构造新的目标函数,并据此采用非支配排序<sup>[3]</sup>构造中间集,作为全局最优粒子的备选集合;同时将当前种群的可行非支配个体更新到外部档案集,并采用 k 近邻个体聚集密度和轮盘赌选择相结合的方法选取全局最优粒子。

### 3.1 约束处理

一般来说,约束处理时,将等式约束条件转换为不等式约束的形式来处理,即

$$|h(x)|-\delta \leq 0 \quad (3)$$

其中,  $\delta$  为等式约束条件的容忍值,取较小的正数。

#### (1) 距离量度

通过在各维目标函数值上引入个体的约束违反程度,距离量度在一定程度上描述了个体与可行域边界的接近程度。其计算过程为首先归一化个体各维的目标函数值:

$$\tilde{f}_i(x) = \frac{f_i(x) - f_{\min}^i}{f_{\max}^i - f_{\min}^i} \quad (4)$$

因此,必须先找出当前群体中每一维目标函数的最大值和最小值:

$$f_{\min}^i = \min_x f_i(x) \text{ and } f_{\max}^i = \max_x f_i(x) \quad (5)$$

个体的约束违反程度定义如下:

$$v(x) = \frac{1}{m} \sum_{j=1}^m \frac{c_j(x)}{c_j^{\max}} \quad (6)$$

其中,

$$c_j(x) = \begin{cases} \max(0, g_j(x)) & \text{if } j = 1, 2, \dots, k \\ \max(0, |h_j(x)| - \delta) & \text{if } j = k+1, 2, \dots, m \end{cases} \quad (7)$$

则个体  $x$  在各目标维度的距离量度可定义为:

$$d_j(x) = \begin{cases} v(x) & \text{if } r_j = 0 \\ \sqrt{\tilde{f}_i(x)^2 + v(x)^2} & \text{otherwise} \end{cases} \quad (8)$$

其中,  $r_j$  为当前种群的可行解比例:

$$r_j = \frac{\text{当前种群的可行解}}{\text{种群规模}} \quad (9)$$

从式(8)可以看出,当  $r_j=0$ 、 $d=v(x)$  时,约束违反程度低者占优;当种群中有可行解存在,则距离量度将具有以下 3 个特点:1)对可行解,在各目标维度,目标函数值小的  $d_i$  也小。2)对不可行解,靠近  $\tilde{f}_i(x)-v(x)$  坐标原点的个体  $d_i$  较小。3)当比较可行解和不可行解时,若两者目标值接近,则可行解具有较小的  $d_i$ 。

#### (2) 自适应惩罚

为了对不可行个体进行额外的惩罚,根据不可行个体的原目标函数值和约束违反程度,定义 2 个惩罚函数,并通过可行解比例自适应调整 2 个惩罚的平衡度,以进一步降低不可行个体适应值,找出不可行解中较好的个体。2 个惩罚函数形式如下:

$$p_i(x) = (1-r_f)X(x) + r_f Y_i(x) \quad (10)$$

其中,  $X(x) = \begin{cases} 0 & \text{if } r_f = 0 \\ v(x) & \text{otherwise} \end{cases}$ ;  $Y_i(x) = \begin{cases} 0 & \text{if } x \text{ is feasible} \\ \tilde{f}_i(x) & \text{otherwise} \end{cases}$ 。

从式(10)可以看出,若可行解比例很低,则  $X$  惩罚项起主要作用,约束违反程度高的个体受惩罚力度大;如果群体

中可行解的比例高,则  $Y$  惩罚项起主要作用,目标函数值大的不可行个体受惩罚力度大;若  $r_f=0$ ,则  $p=0$ 。引入自适应惩罚函数之后,算法能够更好地在寻找可行解和在可行域中搜索更多的 Pareto 最优解之间取得平衡。当可行域比较狭窄、不连续时,由于初始种群中可行解比例较低,因此算法能够偏向于快速找到较多的可行解。

#### (3) 变换后的目标函数

根据距离量度和自适应惩罚函数项,最终的第  $i$  维目标函数形式为:

$$F_i(x) = d_i(x) + p_i(x) \quad (11)$$

新的目标函数有效利用不可行解的信息,具有 4 个明显的特点:1)若种群中可行解比例为 0,个体比较仅依据个体的约束违反程度,这将有助于算法快速找到可行解;2)当可行解比例不为 0,个体目标函数值小,约束违反程度低的个体占优;3)当 2 个个体有相等或相近的目标函数值,将依据自适应惩罚函数值决定个体之间的支配关系;4)若种群全部可行,则将依据个体原目标函数值进行比较。经过约束处理后的目标函数值可直接用于多目标粒子群的优化操作。

### 3.2 基于 k 近邻聚集密度的外部种群维护

外部种群维护是多目标 PSO 算法的关键算子,许多 MOEA 定义各种密度值进行种群维护,根据密度值决定解的加入和移出。典型的有文献[5]提出的密集距离、文献[6]采用的 k 最近邻的个体密度估算方法。结合文献[5-6]方法的优点,本文定义了一种新的基于 k 近邻距离的个体聚集密度估计指标,进行外部种群维护。

设多目标优化问题的目标空间为  $k$  维,为了量化粒子群中任意 2 个个体之间相互联系和相互影响的程度,定义个体  $i$  对  $y$  的影响函数为:

$$\psi(l_{i \rightarrow y}) = (1/(\sigma\sqrt{2\pi}))e^{-l_{i \rightarrow y}^2/2\sigma^2} \quad (12)$$

其中,  $l_{i \rightarrow y}$  表示个体  $i$  与  $y$  之间的欧几里德距离; $\psi$  高斯函数; $\sigma$  为标准偏差。设群体中距离  $y$  最近的  $k$  个个体集合为  $\delta$ ,则个体  $y$  的聚集密度定义为其周围  $k$  个近邻个体对它的影响因子的聚集:

$$D(y) = \sum_{i=1}^k \psi(l(i, y)), i \in \delta \quad (13)$$

这种计算个体密度的方法引入 k 最近邻和影响函数,既能反映个体在种群中的分布信息,也充分考虑了不同距离的个体对该个体的影响,距离越近影响越大,距离越远影响越小,直至没有影响。基于 k 近邻聚集密度的外部种群维护过程如下:

首先,设外部种群为  $NDset$ ,初始规模为  $N$ ,种群最大规模为  $M(N>M)$ 。然后,

- (1) 计算群体中个体之间的欧几里德距离  $l(i, j)$ ;
- (2) 根据距离确定个体  $y$  的 k 最近邻个体集合  $\delta_y$ ;
- (3) 用式(13)计算个体  $y$  的 k 近邻聚集密度  $D(y)$ ;
- (4) 计算当前群体中的中心个体:  
 $O: D(o) = \max\{D(y) | y \in NDset\}$ ;
- (5) 淘汰中心点,构成新的  $NDset$ ;
- (6) 若  $|NDset| > M$ ,则转(1)继续,否则,结束。

基于 k 近邻聚集密度更新外部种群,首先将当前种群中的非支配粒子加入外部集,并将外部集中被支配的粒子淘汰,当所搜索到的 Pareto 解集大于规定大小时,根据上述算法对外部集进行修剪。

### 3.3 全局极值的选取

在多目标 PSO 算法中,全局极值的选取十分重要,将直接影响算法的收敛速度和搜索能力。本文提出一种基于聚集密度的全局极值选取策略。

(1)用约束处理后的目标函数值构造当前群体的最大非支配集,依算法修剪后并入中间集,中间集是历代非支配集的并集。

(2)计算中间集各成员的个体聚集密度,根据该密度值进行轮盘赌选择,从中间集为当前群体每个粒子选取一个全局最优位置。

上述选取过程简单高效,给予位于稀疏区域的个体以较高的选择概率成为全局极值,有利于算法收敛到最优 Pareto 前沿。

### 3.4 算法流程

CMOPSO 算法具体流程如下:

Step1 初始化粒子群,设置外部档案集为空;

Step2  $t=t+1$ ,迭代计算开始循环;

Step3 按式(2)更新粒子的位置和速度,选取部分粒子执行非均匀分层变异操作,并进行边界检查;

Step4 约束处理,利用式(10)修改目标函数值;

Step5 基于修改的目标函数值,进行快速非支配排序,构造修改群体的非支配集合  $P1$ ;

Step6 基于聚集密度的个体密度估计算法,将  $P1$  更新进入中间集合;

Step7 将聚集密度和轮盘赌选择相结合,从中间集合为当前群体每个个体选取一个全局最优位置;

Step8 计算原目标函数值,选取可行非支配个体集合  $P2$ ,基于  $k$  近邻聚集密度,更新外部档案集;

Step9 基于约束支配准则<sup>[2]</sup>,选取更新个体极值;

Step10 判断是否到达最大进化代数,如不满足,则返回到 Step2;

Step11 否则,结束循环,输出外部档案集合。

## 4 数值实验

### 4.1 参数设置

实验中使用了 4 个标准测试函数<sup>[4]</sup>:CONSTR, SRN, TNK 和 BNH,并将本文提出的算法与 2 个经典的约束多目标进化算法进行比较:CNSGA-<sup>[2]</sup>和 CMOEA<sup>[4]</sup>。

实验中各参数设置如下: $c1=c2=0.5$ ,线性递减惯性权值  $\omega$   $[0.4,0.95]$ ,位置调整因子  $\theta=0.8$ ,变异概率 0.2。此外,种群的规模为 150,外部档案集和中间集大小为 200,每个算法独立运行 20 次,最大迭代代数为 300 代。

### 4.2 性能指标

实验采用文献[2]提出的性能指标比较解的优劣,收敛性能指标  $r$  用于评价所求 Pareto 前沿与真实 Pareto 前沿的趋近程度,其值越小,算法逼近真实 Pareto 最优前沿的程度越好;分布性能指标  $\Delta$  用于评价所求解在目标空间分布是否均匀,对于大多数分布广泛的点来说,该指标一般为 0。

### 4.3 实验结果分析

实验将 CMOPSO 分别与进化算法 CNSGA-<sup>[2]</sup>和 CMOEA<sup>[4]</sup>进行了比较,CNSGA- 基于约束支配进行约束处理,CMOEA 采用与本文类似的约束处理方法,但是用遗传算法实现。计算结果如表 1 所示。从表中可以看出,对于 4 个测试函数,3 种算法都能够较好地收敛到真实的 Pareto

前沿。与 CMOEA 相比,除了在 CONSTR 和 SRN 问题上 2 种算法收敛性相近外,CMOPSO 收敛性能和算法的稳定性均明显优于 CMOEA;与 CNSGA- 相比,CMOPSO 的优势不明显,但在求解 Pareto 前沿不连续的 TNK 函数时,仍保持了较大的优势。

表 1 收敛( $r$ )和分布( $\Delta$ )性能指标比较

| 算法     | 指标       | CONSTR  | SRN     | BNH     | TNK     |
|--------|----------|---------|---------|---------|---------|
| CMOEA  | $r$      | 0.019 1 | 0.016 8 | 0.016 1 | 0.009 9 |
|        | $\Delta$ | 0.321 0 | 0.385 0 | 0.421 0 | 0.466 0 |
| CNSGA- | $r$      | 0.022 1 | 0.009 7 | 0.015 0 | 0.011 3 |
|        | $\Delta$ | 0.403 0 | 0.378 0 | 0.520 0 | 0.483 0 |
| CMOPSO | $r$      | 0.020 1 | 0.017 2 | 0.014 9 | 0.006 2 |
|        | $\Delta$ | 0.302 0 | 0.351 0 | 0.298 0 | 0.322 0 |

表 1 和图 1 所示计算结果验证了根据  $k$  近邻聚集密度维护外部种群的有效性,与 CNSGA- 和 CMOEA 相比,CMOPSO 获得的分布性能指标普遍占优,Pareto 解集在整个均衡面上分布更广、更均匀;而 CMOEA 除了在 SRN 上多样性性能稍逊于 CNSGA- 外,在其他函数上 CMOEA 性能更优。

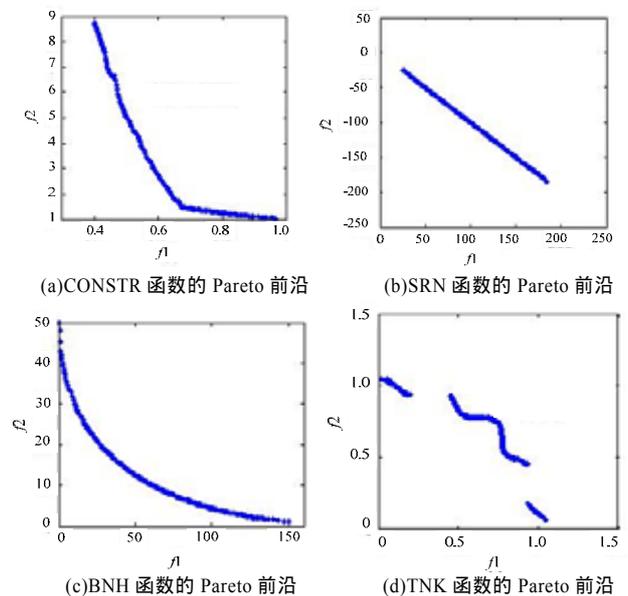


图 1 CMOPSO 算法获得的不同测试函数的 Pareto 前沿

这说明基于距离量度和自适应惩罚的约束处理机制能更好地利用不可行域信息,偏向边界搜索,有效增强搜索的广度;同时,采用  $k$  近邻聚集密度的分布度保持策略,能更精确地描述个体之间的相互影响程度,因此能更好地维持外部集合的均匀性。

## 5 结束语

本文引入一种新的约束处理技术,通过计算距离量度和自适应惩罚函数值,构造新的目标函数,据此进行粒子群操作,有效利用了不可行解,进一步提高了对可行解空间进行精确偏向搜索的能力。本文提出的新的基于  $k$  近邻聚集密度的解集分布性保持策略,既精确描述了不同距离个体之间的影响,又充分反映了个体在种群中的分布。对于 4 个测试函数,算法都能很好地收敛到真实 Pareto 前沿,解的分布也相当均匀。特别是在求解可行域狭窄以及全局最优解位于可行域边界上的约束多目标优化问题时,本文算法十分有效。

(下转第 213 页)