

# 基于搜索空间划分与 Canopy $K$ -means 聚类的 种群初始化方法

李 钊<sup>†</sup>, 袁文浩, 任崇广

(山东理工大学 计算机科学与技术学院, 山东 淄博 255000)

**摘要:** 为了提高差分进化算法对搜索空间的探索与开发能力, 提高差分进化算法的收敛性与算法的进化效率, 提出一种基于搜索空间均匀划分与局部搜索和聚类相结合的种群初始化方法. 该方法首先对决策变量空间进行均匀划分, 并从各个子空间中随机选择一个个体, 得到的个体能够覆盖整个搜索空间; 然后, 利用 Hooke-Jeeves 算法对各子空间进行局部搜索得到局部最优的个体, 并结合改进的 Canopy 算法与  $K$ -means 聚类算法, 辨识搜索空间中的前景区域, 以此为基础对局部搜索产生的局部最优个体进行筛选, 最终生成初始种群中的个体. 通过与其他种群初始化方法对 CEC2017 中 5 个测试函数进行实验对比, 所提出的方法的运行时间可缩减为已有方法的 0.75 倍, 适应度函数可减少为已有方法的 0.03 倍, 且具有最小的标准差以及最优的收敛特性.

**关键词:** 种群初始化; 搜索空间划分; Hooke-Jeeves 算法; 局部寻优;  $K$ -means 聚类

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0358

开放科学(资源服务)标识码(OSID):



**引用格式:** 李钊, 袁文浩, 任崇广. 基于搜索空间划分与 Canopy  $K$ -means 聚类的种群初始化方法[J]. 控制与决策, 2020, 35(11): 2767-2772.

## Population initialization based on search space partition and Canopy $K$ -means clustering

LI Zhao<sup>†</sup>, YUAN Wen-hao, REN Chong-guang

(College of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China)

**Abstract:** In order to improve the ability of exploration and exploitation and improve the convergence and evolutionary efficiency for differential evolution algorithms, a population initialization method based on uniform partition of search space, local search and clustering is proposed. Firstly, the decision variable space is partitioned uniformly, and an individual is randomly selected from each subspace, and the selected individuals can cover the whole search space. Then Hooke-Jeeves algorithm is used to search each subspace locally, and the local optimal individuals are got. Combined with the improved Canopy algorithm and  $K$ -means clustering algorithm, the promising region in the search space is identified. Based on this, the local optimal individuals generated by local search are screened, and the individuals for the initial population are finally generated. Compared with other population initialization methods for five CEC2017 test functions, the running time of the proposed method can be reduced to 0.75 times, and the fitness function can be reduced to 0.03 times that of the existing methods. And the proposed method has the minimum standard deviation and the optimal convergence characteristics.

**Keywords:** population initialization; search space partition; Hooke-Jeeves algorithm; local optimization;  $K$ -means clustering

## 0 引言

种群的初始化对差分进化算法的收敛性至关重要, 也会影响搜索高质量解的效率. 高质量的初始种群可有效解决差分进化算法搜索能力差的问题. 大

部分的种群初始化都是采用随机产生的方法, 例如基于伪随机数发生器(pseudo-random number generator, PRNG)的方法和基于混沌数发生器(chaotic number generator, CNG)的方法等<sup>[1-2]</sup>. 随机方法必然导致初

收稿日期: 2019-03-26; 修回日期: 2019-07-09.

基金项目: 国家自然科学基金项目(61701286); 山东省自然科学基金项目(ZR2018LF002, ZR2017LF004); 淄博市校城融合项目(2018ZBXC021).

<sup>†</sup>通讯作者. E-mail: lizhao\_buaa@126.com.

始种群中存在低质量的个体,并且很难保证覆盖整个搜索空间,从而需要更多的时间收敛到前景区域。

在随机法的基础上,结合反向学习、局部搜索算法和基于中心点的采样方法等,人们提出了一些新的种群初始化方法<sup>[3-7]</sup>。Bajer等<sup>[5]</sup>提出一种基于聚类与柯西突变的种群初始化方法,通过识别搜索空间的前景区域,从中选择高质量的初始个体,并对个体进行柯西突变,以保持个体的多样性。Poikolainen等<sup>[6]</sup>提出一种基于聚类的种群初始化方法,利用Rosenbrock等局部搜索算法对局部空间进行探索,选择局部最优解,然后利用聚类算法探索整个搜索空间,选择前景区域。Elsayed等<sup>[7]</sup>提出一种基于序列的启发式空间填充方法,使得初始个体能够均匀覆盖整个搜索空间。目前提出的方法可有效提高搜索最优解的概率,但是仍存在以下问题:

1) 需要大量的计算,例如基于反向学习的种群初始化方法,需要额外地对两个种群中的个体进行适应度函数的计算,从而选择最优的个体作为初始种群个体;

2) 采用随机法产生的原始种群个体的质量决定着整个算法的性能;

3) 基于序列的种群初始化中的个体虽然能够覆盖整个搜索空间,但是并未对前景空间中的个体和孤立个体进行区分,导致种群中存在低质量的孤立个体,从而影响搜索效率。

针对上述问题,本文提出一种新的种群初始化方法,该方法将基于决策变量划分的搜索空间划分方法与局部搜索和聚类相结合,通过决策变量均匀划分得到能够均匀覆盖整个搜索空间的个体,利用局部搜索方法对各子空间进行局部搜索得到局部最优的初始个体,利用改进的Canopy算法结合K-means算法辨识搜索空间中前景区域,以此为基础对局部搜索产生的局部最优个体进行筛选,最终生成初始种群中的个体。采用该方法产生的初始种群可以覆盖整个搜索空间,种群初始化过程中在追求个体多样性的同时,又确保从前景区域中取得个体,可有效提高差分进化算法的搜索能力,提高差分进化算法的收敛性。

## 1 搜索空间的划分

图1为种群初始化方法的总体框图,主要包括搜索空间划分、前景区域选择和初始种群的构建等,本节主要对搜索空间的划分进行分析。

### 1.1 多目标优化概述

差分进化算法的本质是一种多目标优化算法,用于求解多维空间中的最优解。不失一般性,多目标优

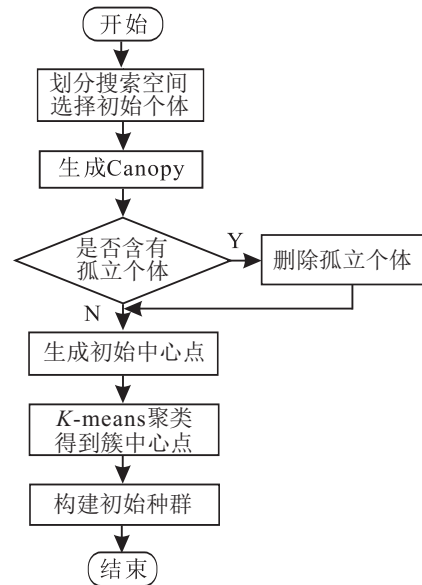


图1 种群初始化方法总体框架图

化可用下式<sup>[8-9]</sup>表示:

$$f(x^*) \leq f(x);$$

$$\text{s.t. } \forall x \in D, x_{i \min} \leq x_i \leq x_{i \max}. \quad (1)$$

其中:  $f(x)$  为目标函数,  $x = [x_1, x_2, \dots, x_n]$  为候选解, 包含  $n$  个决策变量, 每个决策变量  $x_i$  搜索域的取值范围为  $[x_{i \min}, x_{i \max}]$ ;  $D$  为搜索空间。

### 1.2 决策变量搜索域的划分

设  $[x_{\min}, x_{\max}]$  为  $n$  个决策变量搜索域的取值范围,  $x_{\min} = [x_{1 \min}, x_{2 \min}, \dots, x_{n \min}]$  为  $n$  个决策变量最小值的集合,  $x_{\max} = [x_{1 \max}, x_{2 \max}, \dots, x_{n \max}]$  为  $n$  个决策变量最大值的集合。为确保候选解均匀覆盖整个搜索空间,首先需要对每个决策变量的搜索域进行均匀划分。

若将搜索域划分为  $q$  个子空间,则需要  $q+1$  个子向量对  $[x_{\min}, x_{\max}]$  进行划分。  $q+1$  个子向量  $S_k$  可通过下式得到:

$$S_k = \begin{cases} x_{\min}, & k = 1; \\ S_{k-1} + l, & 2 \leq k \leq q; \\ x_{\max}, & k = q + 1. \end{cases} \quad (2)$$

其中  $l$  为两个子向量之间的间隔,有

$$l = \frac{x_{\max} - x_{\min}}{q}. \quad (3)$$

### 1.3 初始个体的选择

搜索空间划分完成后,从各个子空间选择一个随机个体,然后以该随机个体为初始点,利用Hooke-Jeeves算法<sup>[10]</sup>对搜索空间进行探索,选择局部最优的个体作为初始个体。

以一个初始点的探索过程为例,具体的探索过程

如下.

1) 给定初始点  $x^1 \in D$ , 初始步长向量  $\delta$ , 加速因子  $\alpha \geq 1$ , 缩减率  $\beta \in (0, 1)$ , 精度  $\varepsilon > 0$ ,  $e_j = (0, \dots, 0, 1, 0, \dots, 0)^T$ ,  $j = 1, 2, \dots, n$ . 令  $y^1 = x^1$ ,  $k = 1$ ,  $j = 1$ .

2) 轴向搜索. 从参考点  $y^j$  出发, 依次沿  $x$  各分量 (即坐标轴) 的方向以一定的步长寻找函数的下降点, 令

$$y^{j+1} = \begin{cases} y^j + \delta e_j, & f(y^j + \delta e_j) < f(y^j); \\ y^j - \delta e_j, & f(y^j - \delta e_j) < f(y^j); \\ y^j, & \min[f(y^j + \delta e_j), f(y^j - \delta e_j)] \geq f(y^j). \end{cases} \quad (4)$$

当  $x$  的各个坐标轴遍历完成后, 若  $f(y^{n+1}) < f(x^k)$ , 则进行模式搜索, 否则需重新调整步长进行轴向搜索.

3) 模式搜索. 令  $x^{k+1} = y^{n+1}$ , 将  $y^1 = x^{k+1} + \alpha(x^{k+1} - x^k)$  作为新的参考点, 令  $k := k+1$ ,  $j = 1$ . 新的参考点确定完成后转到 2) 进行轴向搜索.

4) 调整步长. 如果  $\delta \leq \varepsilon$ , 则停止搜索,  $x^k$  即为初始个体; 否则, 令  $\delta := \beta\delta$ ,  $y^1 = x^k$ ,  $x^{k+1} = x^k$ , 同时令  $k := k+1$ ,  $j = 1$ , 转 2) 继续新的轴向搜索.

从每个子空间中选择随机个体作为初始点, 可保证初始点能够均匀覆盖整个搜索空间. 采用 Hooke-Jeeves 算法对搜索空间进行轴向和模式搜索, 可对最优解的数量和位置进行一个粗略的评估, 并将搜索得到的局部最优解作为下一步聚类的候选个体, 以提高参与聚类的初始个体的质量, 使得前景区域的选择更有指向性, 为前景区域的选择奠定基础.

## 2 前景区域的选择

当候选个体确定后, 采用  $K$ -means 聚类算法对候选个体进行聚类,  $K$ -means 聚类算法是一种简单的迭代型聚类算法, 可将多个个体构成的个体集划分为  $k$  个簇, 使得同簇内个体间比较相似, 而不同簇个体间差异比较大, 每个簇代表一个前景区域.  $K$ -means 聚类算法的运行速度比较快, 同时易于修改, 所以在实际中使用非常广泛<sup>[11-13]</sup>.

在实际应用中  $K$ -means 聚类算法也存在一些问题, 例如需要提前确定  $k$  值, 以及初始的  $k$  个聚类中心是随机选取的, 而候选个体中可能存在孤立个体, 如果将孤立个体作为初始聚类中心, 则会影响到聚类结果. 针对上述问题, 本文结合 Canopy 算法<sup>[14-15]</sup>对聚类过程进行修改, 改进后的算法聚类效果更好, 聚类准确性更高.

### 2.1 Canopy 算法个体选择机制

常规的 Canopy 算法首先给定候选个体集合与距离阈值  $T_1$  和  $T_2$ ,  $T_1$  为 Canopy 集合的半径,  $T_2$  为非 Canopy 候选个体集合的半径, 且  $T_1 > T_2$ ,  $T_1$  和  $T_2$  的设定可通过交叉验证方式确定<sup>[16]</sup>. 然后从候选个体中随机选择个体  $x_i$  作为 Canopy, 同时将个体  $x_i$  从候选个体中移除, 并利用下式计算候选个体中所有个体到  $x_i$  的距离:

$$d = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}. \quad (5)$$

其中  $x_{ik}$  表示个体  $x_i$  的第  $k$  个决策变量, 且  $i \neq j$ <sup>[14]</sup>. 将  $d$  小于  $T_1$  的个体划到 Canopy 中, 同时将  $d$  小于  $T_2$  的个体从候选个体中移除. 重复该过程, 直到没有候选个体时结束.

为了保持算法的随机性, 需要随机选择个体作为 Canopy, 不可避免地会选择孤立个体或者性能较差的个体, 从而降低算法的效率. 为了提高将局部最优个体作为 Canopy 的概率, 同时避免孤立个体作为 Canopy, 本文对 Canopy 算法的候选个体选择机制进行了修改.

针对选择局部最优个体的问题, 设计局部最优个体选择机制. 首先, 计算各候选个体的适应度函数  $F(x_i)$ , 其中  $x_i$  为第  $i$  个候选个体; 然后, 利用

$$P(x_i) = \frac{F(x_i)}{\sum_{j=1}^N F(x_j)} \quad (6)$$

计算各候选个体的选择概率, 利用轮盘赌选择法<sup>[5]</sup>对候选个体进行选择. 利用该方法可有效提高选择局部最优的个体作为 Canopy 的概率, 同时又保证了个体选择的随机性.

针对孤立个体问题, 设计孤立个体判别机制. 当 Canopy 划分完成后, 假设生成  $m$  个 Canopy, 每个 Canopy 即为一个簇, 每个簇中个体的数量为  $\text{num}_i$ ,  $i = 1, 2, \dots, m$ . 若  $\text{num}_i < \text{numPoint}$ , 则将第  $i$  个簇中个体判定为孤立个体, 将该簇及簇中的个体删除.  $\text{numPoint}$  为判定簇中个体是否为独立个体的阈值, 其值可根据候选个体的数量及实际的应用背景确定. 当所有孤立个体及相应的簇都删除后, 剩余簇的数量即为  $K$ -means 聚类算法的  $k$  值, 然后选择每个簇的中心点作为  $K$ -means 聚类算法的初始中心点  $c_i$ ,  $i = 1, 2, \dots, k$ .

### 2.2 $K$ -means 聚类算法

通过 Canopy 算法得到  $K$ -means 聚类算法的  $k$  值和初始中心点后, 利用  $K$ -means 算法进行进一步的聚

类划分. 根据 Canopy 算法生产的  $k$  个中心点  $c_i$ , 形成  $k$  个簇  $S_i$ , 根据下式计算候选个体到中心点的距离:

$$S_i = \{x : \|x - c_i\|^2 \leq \|x - c_p\|^2 : \forall p = 1 : k, p \neq i\}, \quad (7)$$

并将候选个体分配到距离最小的簇中<sup>[11]</sup>.

当候选个体划分完成后, 利用下式重新计算各个簇的中心点:

$$c_i = \frac{1}{|S_i|} \sum_{x \in S_i} x, \quad i = 1, 2, \dots, k. \quad (8)$$

其中:  $|S_i|$  为第  $i$  个簇中包含候选个体的数量,  $\sum_{x \in S_i} x$  为第  $i$  个簇中候选个体各分量之和.

当新的中心点计算完成后, 再次利用式 (7) 对簇进行重新划分, 循环此过程直到中心点不再发生变化或者达到设定的迭代次数, 此时  $K$ -means 聚类过程结束.

### 3 初始种群的构建

经过聚类后, 各个簇的中心点可能会集中在某个局部最优区域, 当对搜索空间进行探索时, 若该区域包含全局最优点, 则进化算法最终收敛于全局最优点; 若该区域不包含全局最优点, 则进化算法最终会收敛于该区域的某个局部最优点, 从而导致提前收敛. 为了避免进化算法的提前收敛, 需保证初始种群中个体的多样性以及个体可覆盖整个搜索空间, 因此除了将  $K$ -means 聚类后各个簇的中心点作为初始种群个体外, 还将采用以下方法对初始种群进行填充.

因为经过 Hooke-Jeeves 算法产生的候选个体是在搜索空间均匀划分的前提下得到的, 是对搜索空间的初步探索, 得到的候选个体可覆盖整个搜索空间, 并保持了个体间的多样性. 首先, 将候选个体中已作为 Canopy 的个体剔除, 以避免初始种群个体的重复选择; 然后, 利用式 (6) 重新计算各候选个体的选择概率, 利用轮盘赌选择法对候选个体进行选择, 将得到的个体作为初始种群的个体, 直到达到初始种群所需的个体数量, 初始种群构建完成.

### 4 实验对比分析

为了验证本文提出的种群初始化方法的有效性, 将提出的种群初始化方法应用于差分进化算法, 并与现有的种群初始化方法分别从算法运行时间、适应度函数和收敛性等方面进行对比分析. 为了便于对比分析, 差分进化算法采用相同的控制参数, 设变异策略为 DE/best/1, 变异算子  $F = 0.5$ , 交叉概率  $C_r = 0.3$ , 最大迭代次数为 200, 种群大小为 30, 个体的维度为 10, 每种算法独立运行 100 次. 本实验的测

试平台采用 jMetal, jMetal 是 Java 实现的一套多目标优化框架, 可较方便地实现各种多目标优化方法, 并在 3.20 GHz Intel Core(TM) i5-3470 CPU 的计算机上运行.

本实验选用测试函数集 CEC2017 进行测试, CEC2017 包含 30 个测试函数, 其中包括 3 个单峰函数, 7 个基本的多峰函数和 20 个混合函数, 混合函数由多个子函数构成. 为了提高实验的测试效率, 本实验从中选择了前 5 个测试函数进行测试.

实验选择 3 个主流的种群初始化方法作对比分析, 包括基于聚类和柯西变异的种群初始化方法 (population initialization based on clustering and cauchy mutation, PICCM)<sup>[5]</sup>, 基于聚类的种群初始化方法 (cluster-based population initialization, CBPI)<sup>[6]</sup> 和基于序列的种群初始化方法 (sequence-based deterministic initialization, SBDI)<sup>[7]</sup>.

#### 4.1 运行时间对比

表 1 为基于不同种群初始化方法的差分进化算法运行时间的均值与标准差, 运行时间单位为 ms. 综合分析表 1 可知, 本文提出的基于搜索空间划分和局部寻优方法的种群初始化方法, 可有效减少差分进化算法的运行时间, 并有效减小了运行时间的标准差. 例如函数  $f_5$ , 本文提出的方法的运行时间为 SBDI 方法的 3/4, 其标准差只有 SBDI 的 1/4 左右. 主要原因在于, 虽然 SBDI 也对搜索空间进行了划分, 但是其搜索空间中个体的选择是随机的, 从而降低了算法的搜索效率. 另外, CBPI 和 PICCM 方法中, 均采用  $K$ -means 聚类算法进行局部寻优, 而且 CBPI 也采用 Rosenbrock 等局部搜索算法对搜索空间进行搜索, 但是其局部搜索的初始个体是随机选择的, 并且不能保证覆盖搜索空间. 而本文提出的种群初始化方法, 不仅对搜索空间进行了划分, 而且利用 Hooke-Jeeves 算法选择个子空间内局部最优的个体作为初始个体, 保证初始个体覆盖搜索空间, 同时具有局部最优特性. 因此, 上述现有方法的运行时间要高于本文提出的方法, 并且标准差要大于本文提出的方法.

#### 4.2 适应度函数的均值

表 2 为基于不同种群初始化方法的差分进化算法的适应度函数的均值与标准差. 与运行时间的结果类似, 基于本文提出的种群初始化方法的差分进化算法, 具有更优的适应度均值和标准差. 例如函数  $f_4$  和  $f_5$  的均值分别为 15.8 和 16.2, 基本收敛于该函数的真实最小值, 而函数  $f_1$ 、 $f_2$  和  $f_3$  的均值与方差也优于其他方法. 针对函数  $f_2$ , 其均值为 SBDI 方法的 3%, 而

表 1 运行时间的均值与标准差

函数	Proposed		CBPI		PICCM		SBDI	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
$f_1$	5.09e+03	36.34	5.25e+03	65.38	5.63e+03	85.30	5.85e+03	93.18
$f_2$	4.53e+03	19.65	4.98e+03	25.57	5.47e+03	39.55	5.61e+03	46.74
$f_3$	4.34e+03	15.42	4.75e+03	23.78	5.25e+03	35.78	5.47e+03	36.38
$f_4$	4.77e+03	39.64	5.08e+03	55.68	5.69e+03	100.38	5.93e+03	120.68
$f_5$	4.13e+03	16.34	4.53e+03	28.41	5.19e+03	40.65	5.53e+03	58.67

表 2 适应度函数的均值与标准差

函数	Proposed		CBPI		PICCM		SBDI	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
$f_1$	3.33e+03	4.68e+04	4.01e+03	3.14e+05	4.85e+05	3.36e+05	6.78e+05	4.53e+05
$f_2$	3.54e+04	3.25e+04	4.68e+04	3.21e+04	5.38e+05	4.87e+05	1.18e+06	4.77e+06
$f_3$	4.83e+03	4.02e+02	6.68e+03	4.36e+02	7.08e+03	5.46e+02	1.52e+04	5.04e+03
$f_4$	1.58e+01	0.00e+00	3.16e+01	0.00e+00	3.82e+02	3.83e+00	4.07e+02	4.65e+00
$f_5$	1.62e+01	0.00e+00	3.25e+01	0.00e+00	4.36e+02	4.02e+00	5.27e+02	4.47e+00

标准差约为 1%。更小的适应度函数均值说明进化算法更有效地收敛于函数的最小值,而更小的方差则说明进化算法的稳定性更优。主要原因在于利用 Hooke-Jeeves 算法选择局部最优的个体提高了空间探索的效率,有利于得到局部最优的候选个体,改进的 Canopy 算法结合  $K$ -means 算法,可进一步提高  $K$ -means 算法初始个体的质量,利于寻找各个簇内的局部最优个体,从而使得进化算法收敛于全局最优解。

### 4.3 收敛曲线分析

图 2~图 6 为采用不同种群初始化方法的差分进化算法分别对函数  $f_1 \sim f_5$  计算的收敛曲线对比,由于适应度函数的取值范围较大,为了便于对比,采用对数函数对适应度函数值进行压缩。

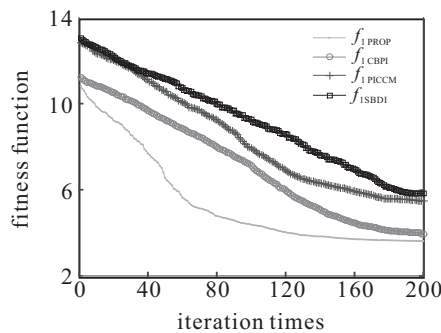


图 2 函数  $f_1$  的收敛曲线

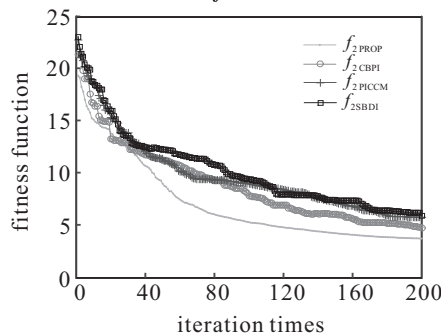


图 3 函数  $f_2$  的收敛曲线

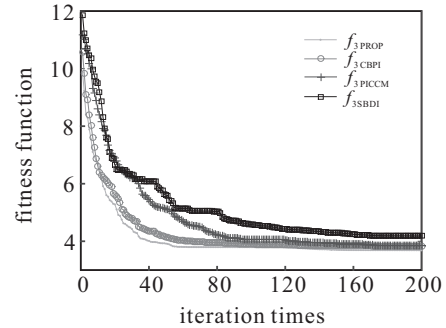


图 4 函数  $f_3$  的收敛曲线

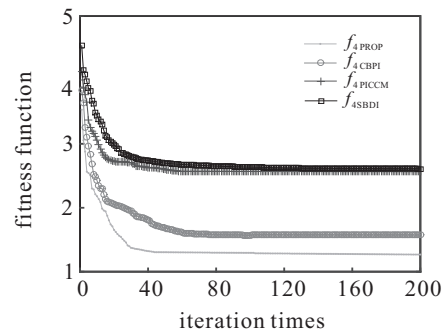


图 5 函数  $f_4$  的收敛曲线

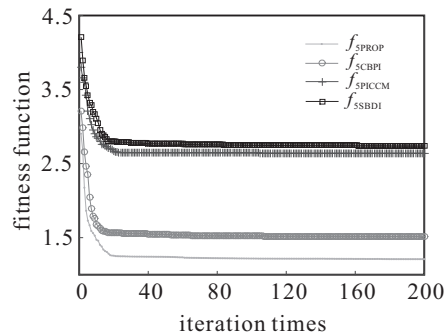


图 6 函数  $f_5$  的收敛曲线

由图 2~图 6 可知,采用本文提出的种群初始化方法的差分进化算法,在迭代初始时具有更优的适应度函数值。例如,与 SBDI 方法相比较,函数  $f_2$  最大可

达到2.3个数量级,与CBPI相比较,最大可达到2个数量级,说明经过空间划分以及利用Hooke-Jeeves算法局部寻优后,初始种群中具有更优的个体。同时,本文提出的方法具有更快的收敛速度并收敛于更小的适应度函数值,主要原因在于改进后的Canopy算法进一步提高了 $K$ -means算法中初始中心的质量,提高了聚类算法的效率。另外,将 $K$ -means聚类的中心点与轮盘赌选择法得到的个体作为初始种群个体,保证初始种群中个体的多样性以及个体可覆盖整个搜索空间,从而提高差分进化算法收敛效率。

## 5 结论

本文针对差分进化算法的种群初始化问题进行了研究,将基于决策变量划分的搜索空间划分方法与Hooke-Jeeves算法局部搜索和改进的 $K$ -means聚类算法相结合,提出了一种新的种群初始化方法。并将该种群初始化方法应用于差分进化算法中,并与现有的具有代表性的种群初始化方法对CEC2017测试函数进行了实验对比。实验结果表明,本文提出的种群初始化方法具有更短的运行时间、更小的适应度函数和更优的收敛特性,有效地提高了差分进化算法的进化效率。

## 参考文献(References)

- [1] Borhan K, Li X D, Qin A K. A review of population initialization techniques for evolutionary algorithm[C]. Proceedings of the 2014 IEEE Congress on Evolutionary Computation. Beijing: Institute of Electrical and Electronics Engineers Inc, 2014: 2585-2592.
- [2] 田红军, 汪镭, 吴启迪. 一种求解多目标优化问题的进化算法混合框架[J]. 控制与决策, 2017, 32(10): 1729-1738.  
(Tian H J, Wang L, Wu Q D. A hybrid framework of evolutionary algorithm for solving multi-objective optimization problems[J]. Control and Decision, 2017, 32(10): 1729-1738.)
- [3] Mustafi D, Sahoo G. A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the  $k$ -means algorithm with applications in text clustering[J]. Soft Computing, 2019, 23(15): 6361-6378.
- [4] Yang Y H, Xu X B, He S B, et al. Cluster-based niching differential evolution algorithm for optimizing the stable structure of metallic clusters[J]. Computational Materials Science, 2018, 149: 416-423.
- [5] Bajer D, Martinović G, Brest J. A population initialization method for evolutionary algorithms based on clustering and cauchy deviates[J]. Expert Systems with Applications, 2016, 60(1): 294-310.
- [6] Poikolainen I, Neri F, Caraffini F. Cluster-based population initialization for differential evolution frameworks[J]. Information Sciences, 2015, 297: 216-235.
- [7] Elsayed S, Sarker R, Coello C A. Sequence-based deterministic initialization for evolutionary algorithms[J]. IEEE Transactions on Cybernetics, 2017, 47(9): 2911-2923.
- [8] 吴秀丽, 袁琦. 差分进化算法求解带批处理机的机器人制造单元调度问题研究[J]. 控制与决策, 2020, 35(1): 74-82.  
(Wu X L, Yuan Q. Differential evolution algorithm for solving robotic cell scheduling problem with batch-processing machines[J]. Control and Decision, 2020, 35(1): 74-82.)
- [9] 侯莹, 韩红桂, 乔俊飞. 基于参数动态调整的多目标差分进化算法[J]. 控制与决策, 2017, 32(11): 1985-1990.  
(Hou Y, Han H G, Qiao J F. Adaptive multi-objective differential evolution algorithm based on the dynamic parameters adjustment[J]. Control and Decision, 2017, 32(11): 1985-1990.)
- [10] Tolga A, Egemen Y. Multiobjective Hooke-Jeeves algorithm with a stochastic Newton-Raphson-like step-size method[J]. Expert System with Application, 2019, 117(3): 166-175.
- [11] Marco C, Aritz P, Jose A L. An efficient approximation to the  $K$ -means clustering for massive data[J]. Knowledge-Based Systems, 2017, 117(2): 56-69.
- [12] Surya K, Tripti M, Vinay K J, et al. LeaderRank based  $k$ -means clustering initialization method for collaborative filtering[J]. Computers and Electrical Engineering, 2018, 69(6): 598-609.
- [13] Jocelyn Z G, Ado A A, Abdelhak M G. Efficient  $K$ -means based clustering scheme for mobile networks cell sites management[J]. Computer and Information Sciences, 2018, 11: 1-8.
- [14] Zhang G, Zhang C C, Zhang H Y. Improved  $K$ -means algorithm based on density Canopy[J]. Knowledge-Based Systems, 2018, 145(4): 289-297.
- [15] Li J J, Zhang K, Yang X L, et al. Category preferred Canopy- $K$ -means based collaborative filtering algorithm[J]. Future Generation Computer Systems, 2019, 93: 1046-1054.
- [16] 胡小建, 韦超豪. 基于Canopy和 $k$ -means算法的订单分批优化[J]. 合肥工业大学学报: 自然科学版, 2017, 40(3): 414-419.  
(Hu X J, Wei C H. Order batching optimization based on Canopy and  $k$ -means algorithm[J]. Journal of Hefei University of Technology: Natural Science, 2017, 40(3): 414-419.)

## 作者简介

李钊(1983—),男,讲师,博士,从事机器学习、多目标优化等研究, E-mail: lizhao\_buaa@126.com;

袁文浩(1985—),男,讲师,博士,从事语音信号处理和语音增强等研究, E-mail: why\_sdut@126.com;

任崇广(1982—),男,副教授,博士,从事智能装备、云计算和大数据处理等研究, E-mail: renchg@sdut.edu.cn.