

文章编号: 2095-2163(2024)02-0009-09

中图分类号: TP301.6

文献标志码: A

种群熵启动反向学习的动态多种群粒子群算法

梁晓磊¹, 张孟镛¹, 周文峰², 武建国¹

(1 武汉科技大学 汽车与交通工程学院, 武汉 430065; 2 武汉理工大学 交通与物流工程学院, 武汉 430065)

摘要: 针对传统粒子群优化算法在求解复杂优化问题时容易陷入局部最优和停滞的问题, 提出采用种群熵启动反向学习的动态多种群粒子群算法。借鉴狮群算法划分狮群的思想, 采用动态多种群划分策略, 将粒子划分成3个不同行为子群, 对其实施不同的位置更新公式, 保持粒子在搜索过程中的多样性; 在迭代阶段, 为避免算法早熟, 构建了各维重心反向变异策略丰富变异备选个体, 并结合种群熵指标进行种群状态评价适时启动变异策略, 帮助粒子跳出局部最优。最后, 通过8个基准测试函数与同种类6种经典和新型改进算法, 在不同维度下进行测试对比。数值实验结果表明, 改进策略显著提升了粒子群算法搜索能力, 在搜索精度和搜索速度方面均优于其他对比算法。

关键词: 粒子群算法; 狮群算法; 种群熵; 反向学习; 动态多种群划分

Dynamic multi-swarm particle swarm optimization using population entropy to start reverse learning

LIANG Xiaolei¹, ZHANG Mengdi¹, ZHOU Wenfeng², WU Jianguo¹

(1 School of Automobile and Traffic Engineering, Wuhan University of Science and Technology, Wuhan 430065, China;

2 School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430065, China)

Abstract: Due to particle swarm optimization (PSO) easily being trapped in local optimum, a dynamic multi-type particle swarm optimization based on population entropy to start each dimension mutation adaptively is proposed. Inspired from the lion group searching behaviors, a dynamic multi-type particle division strategy is provided to divide the particles into three different classes, and implements different information updating models on them, which can maintain the diversity of particles in the search process. In the iterative stage, the algorithm introduces a population entropy value as the judgment condition to start the reverse learning strategy for the global optimal particles, thereby helping the particles to jump out of local optimal. Numerical experiments show that the proposed algorithm has stronger search ability than other selected algorithms with excellent performance in terms of accuracy and search speed.

Key words: particle swarm optimization; lions algorithm; population entropy; reverse learning; dynamic multi-type division.

0 引言

群智能优化技术是个体通过简单行为和协作机制, 使群体涌现出复杂智能的一种技术, 已经在众多领域得到了有效应用, 表现出了优秀的优化能力。粒子群算法 (Particle Swarm Optimization, PSO) 作为群智能技术中的一种, 由 Kennedy 和 Eberhart^[1-2] 受鸟群寻觅食物行为的启发而提出。粒子群算法具有结构简洁、参数少和鲁棒性强等特点, 在工程科学领域得到广泛应用。但其在求解复杂高维度函数时, 容易出现群体早熟, 导致陷入局部最优的不足, 影响

对复杂问题的求解。针对该问题, 众多学者从多方面提出了改进方案。例如, 在增加参数指标精英选择等不同策略方面: Zhao 等^[3] 引入参数指标来判断粒子是否进入早熟状态; 孙辉等^[4] 提出混合均值中心的概念, 并对其反向学习去增加粒子的开发能力; 刘天宇等^[5] 引入基于权重向量的多样性指标来检测粒子的多样性, 并采用自适应变异策略对外部种群进行更新。另外, 通过改变粒子种群间的拓扑结构来选择学习榜样也得到较好的结果。如: 李文锋等^[6] 采用聚类算法对种群拓扑结构进行动态划分, 以促进子群间信息流通; Jiang 等^[7] 利用双层粒子群结构增强两个层次粒子群的相互协作, 以提

作者简介: 梁晓磊 (1985-), 男, 博士, 副教授, 主要研究方向: 智能优化算法、复杂系统建模与仿真; 张孟镛 (1998-), 男, 硕士研究生, 主要研究方向: 智能优化算法、闸室调度; 周文峰 (1996-), 男, 博士研究生, 主要研究方向: 港口物流; 武建国 (2002-), 男, 本科生, 主要研究方向: 物流系统优化。

收稿日期: 2023-02-09

哈尔滨工业大学主办 ◆ 学术研究与应用

高粒子搜索效率;Chen等^[8]提出的新型双层种群结构中采用两种不同控制参数的推理突变操作产生精英粒子,以保证种群多样性。

为进一步提高粒子的寻优能力,帮助粒子更好地跳出局部最优,近年反向学习策略受到学者的青睐。周凌云等^[9]提出邻域重心反向学习策略,以邻域重心作为参考点计算全局最优位置的反向解,充分吸收群体的搜索经验,保持了种群的多样性。夏学文等^[10]利用适应度值较差的粒子指导部分粒子进行反向学习,从而保持粒子种群的多样性。唐可心等^[11]在算法迭代后期,引入基于时变概率实施反向学习策略和邻域扰动操作,对全局最优个体施加混合变异,帮助粒子跳出局部最优。Chen等^[12]对随机初始化的粒子采用反向学习策略,选取最优粒子做初始种群。虽然上述研究均在一定程度上提升了个体搜索能力,说明丰富种群多样性的反向学习是有效的,但对于何时启动反向学习从而避免计算资源浪费,现有研究还缺少深入分析。另外,在群体搜索过程中,如何有效评价群体状态以赋予不同搜索行为,也是提升算法性能重要问题。曹玉莲等^[13]基于拟熵物理量来判断何时启动传统的局部搜索策略。孙骞等^[14]提出一种基于熵模型的粒子群优化算法,通过粒子的熵模型这一概念比较粒子的聚集程度,便于后期进行局部搜索。熵模型研究为进行群体状态评价提供了有效的路径。本文基于种群状态赋予不同搜索能力的角度出发,提出了一种采用种群熵启动反向学习的动态多种群粒子群算法(Dynamic Multi-swarm Particle Swarm Optimization using Population Entropy to Start Reverse Learning, DMP SO-PERL),算法基于狮群狩猎划分狮群种类的思想,将粒子群动态划分为3个种群,根据粒子在搜索空间的不同位置,采用适合的更新策略,让各个子群充分地探索解空间,加强算法的开发和探索能力。同时,采用种群熵指标来反映粒子群的聚集程度和多样性,作为启动各维重心反向学习策略的判断条件,避免反向学习算子频繁启动,节省计算资源,帮助粒子跳出局部最优。

1 算法基本原理

1.1 标准粒子群优化算法

设种群大小为 N 的粒子群,在维度为 D 的空间中搜索,第 i 个粒子在解空间的位置为 $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$, $i = 1, 2, 3, \dots, N$,粒子的速度可表示为 $v_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$ 。同样的,粒子的个体

极值可表示为 $pbest_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD})$,种群的全局极值为 $gbest = (g_1, g_2, g_3, \dots, g_D)$ 。迭代到第 t 代时,粒子的速度和位置依据公式(1)、公式(2)更新:

$$v_{id}^{t+1} = wv_{id}^t + c_1r_1(p_{id}^t - x_{id}^t) + c_2r_2(g_d^t - x_{id}^t) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

其中, w 为惯性权重; c_1, c_2 为学习因子; r_1, r_2 为区间 $[0, 1]$ 上均匀分布的随机数。

1.2 狮群算法基本原理

狮群算法^[15]在求解优化问题时,将狮群分为3类:狮王、母狮和幼狮。狮王是狮群中最强壮的个体,保护领地,并分配食物,还要面对挑战;母狮主要负责捕猎,先大范围勘探,当靠近食物时会进行围猎;幼狮在饥饿时靠近狮王,等待分配食物,进食后,跟随母狮深度捕猎,长大后可以挑战狮王。不同种群的位置更新方式不同,狮王在最佳位置处小范围移动确保自己的特权,按公式(3)更新自身位置:

$$x_i^{t+1} = g^t(1 + \gamma \|x_i^t - g^t\|) \quad (3)$$

母狮与另外一母狮协作进行全局搜索,按公式(4)更新位置:

$$x_i^{t+1} = \frac{x_i^t + x_c^t}{2}(1 + \alpha_f\gamma) \quad (4)$$

幼狮根据公式(5)的3种行为更新自身位置:

$$x_i^{t+1} = \begin{cases} \frac{g^t + x_i^t}{2}(1 + \alpha_c\gamma), & q < \frac{1}{3} \\ \frac{x_m^t + x_i^t}{2}(1 + \alpha_c\gamma), & \frac{1}{3} \leq q \leq \frac{2}{3} \\ \frac{\bar{g}^t + x_i^t}{2}(1 + \alpha_c\gamma), & \frac{2}{3} \leq q < 1 \end{cases} \quad (5)$$

其中, g^t 为第 t 代群体最优位置; γ 为服从正态分布 $N(0, 1)$ 的随机数; x_i^t 为第 t 代第 i 头狮子的历史最佳位置; x_c^t 为在第 t 代母狮群中被随机挑选协作伙伴的历史最佳位置; α_f 为母狮移动扰动因子; α_c 为幼狮移动扰动因子; x_m^t 为被幼狮跟随的母狮的历史最佳位置; \bar{g}^t 为第 i 个幼狮在捕猎范围内被驱赶的位置; q 为 $U(0, 1)$ 均匀分布随机数。

2 种群熵启动反向学习的动态多种群粒子群算法

2.1 动态多种群的划分策略

2.1.1 划分思路

借鉴狮群算法的种群划分思想,在每一代进化

时,依据粒子适应度值进行子群动态划分,适应度值最好的粒子称为“top 粒子”,适应度值较好的部分粒子称为“次 top 粒子”,“top 粒子”和“次 top 粒子”总个数占种群数的比例为 β ,适应度值最差的部分粒子称为“弱 top 粒子”。“次 top 粒子”进行全局搜索,引领“弱 top 粒子”勘探更广的解空间区域进行深度挖掘,寻找最优位置。

相比于固定划分粒子的多种群,动态划分粒子种群方式可以赋予个体更多不同的学习方式,避免过度限制粒子的自由,促进粒子间的信息交流,以提高算法的搜索效率。

2.1.2 不同种群粒子的更新机制

(1) “top 粒子”更新机制

借鉴狮群算法中雄狮位置的更新方式,定义“top 粒子”主要留守当前最优适应度位置,游走在其周围进行深度搜索。本文提出公式(6)对其进行位置更新:

$$x_i^{t+1} = dis \cdot g^t (1 + \gamma \|x_i^t - g^t\|) + v_i^{t+1} \quad (6)$$

其中,极值扰动因子 dis ^[16] 采用非线性变化,以平衡“top 粒子”迭代过程中全局搜索和局部搜索能力。

中心极限定理指出,不论总体的分布情况如何,大多统计量的分布近似正态分布^[17]。因此,本文多维解空间粒子的位置服从多元正态分布。根据正态分布 $N(\mu, \sigma^2)$ 随机变量的概率规律, μ 描述了正态分布的集中趋势,其对应了正态分布的峰值。在多维解空间中, g^t 类似峰值 μ ，“top 粒子”在其周围游走,“次 top 粒子”和“弱 top 粒子”分布在其两侧,这样有利于搜索的粒子向峰值 g^t 靠拢,加快算法的收敛速度。

(2) “次 top 粒子”更新机制

结合母狮觅食过程中迭代机制,从“次 top 粒子”中随机挑选一个粒子协助当前粒子进行全局搜索,增加与其他粒子的信息交流,勘探更大范围的位置。“次 top 粒子”采用公式(7)进行位置更新:

$$x_i^{t+1} = \frac{x_i^t + x_c^t}{2} (1 + \alpha_f \gamma) + v_i^{t+1} \quad (7)$$

其中, α_f 为平衡扰动因子,计算见公式(8);

$$\alpha_f = step \cdot \exp\left(-\frac{10t}{T}\right)^{10} \quad (8)$$

其中, $step = 0.1 \cdot (X_{\max} - X_{\min})$, 为“次 top 粒子”的移动步长; X_{\max} 为解空间最大边界; X_{\min} 为解空间最小边界; T 为最大迭代次数。

当 $step = 1$ 时,“次 top 粒子”的移动扰动因子

α_f 的曲线如图 1 所示。从图中可见,扰动因子前期接近于 1,有利于“次 top 粒子”进行大范围的勘探;后期逐渐趋近于 0,让其进行深度的开采。

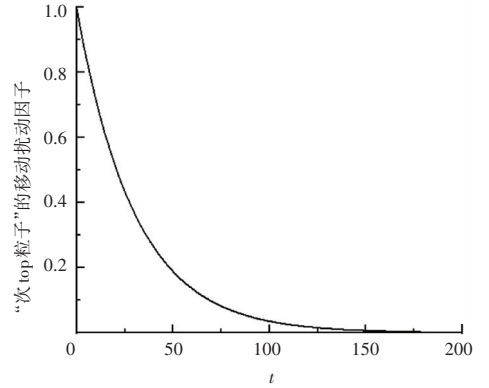


图 1 “次 top 粒子”移动扰动因子 α_f 变化图

Fig. 1 The curve of the movement disturbance factor α_f of the sub-top particle

(3) “弱 top 粒子”更新机制

结合“弱 top 粒子”的搜索特征,本文提出公式(9)对其进行位置更新:

$$x_i^{t+1} = \begin{cases} \frac{g^t + x_c^t}{2} (1 + \alpha_c \gamma) + v_i^{t+1}, & 0 \leq q \leq \frac{1}{2} \\ x_i^t + \sin(r \cdot \frac{x_i}{2}) \cdot v_i^{t+1}, & \frac{1}{2} < q \leq 1 \end{cases} \quad (9)$$

其中, x_c^t 为随机挑选的“次 top 粒子”的第 t 代历史最优位置; α_c 是“弱 top 粒子”的移动扰动因子, $\alpha_c = (T - t)/T$ 。分析式(9)可得:当 $0 \leq q \leq 1/2$ 时,“弱 top 粒子”位置更新公式的前面部分是由“次 top 粒子”的历史最佳位置和全局最优位置共同决定,加强“弱 top 粒子”与“次 top 粒子”和“top 粒子”的信息交流,可引导粒子对解空间进行更精细的搜索,加快了粒子向全局最优收敛的速度;当 $1/2 < q \leq 1$ 时,在位置更新的速度部分添加正弦扰动因子,帮助“弱 top 粒子”在进行搜索时跳出局部最优。

2.2 采用种群熵启动反向学习策略

2.2.1 各维重心反向变异策略

优化高维复杂问题时,粒子在迭代后期往往易陷入局部最优。为了跳出局部极值点,一些研究采用了反向学习策略,如邻域重心反向学习策略^[4]、混合均值中心反向学习策略^[5]等。目前,大多数学者对最优粒子的变异都采用所有维度或随机挑选部分维度进行变异的策略,对于高维复杂函数,计算其适应度值时会受其他维度之间的干扰,导致某些维度的更优值被其他维度的差值所掩盖,造成算法求解效率的低下。因此,本文对最优粒子采用各维重

心反向变异策略 (Dimensional Center of Gravity Reverse Mutation Strategy, DCRMS), 避免了跨维干扰, 取得更优的反向变异解。

定义 1 (重心) 种群大小是 N , 维度是 D 的种群为 $(X_1, X_2, X_3, \dots, X_N)$, 则整个种群的重心定义为

$$M_{cen} = \left(\frac{X_1 + X_2 + X_3 + \dots + X_N}{N} \right) \quad (10)$$

定义 2 (重心反向学习) 假设 $X_i = (x_1, x_2, x_3, \dots, x_D)$ ($X_{\max} \leq x_i \leq X_{\min}$) 是 D 维解空间的一个可行解, 其反向解由公式 (11) 确定:

$$\bar{X}_i = 2 \cdot r \cdot M_{cen} - X_i \quad (11)$$

其中, r 是 $U(0,1)$ 随机数。

假设当前迭代过程中全局最优粒子为 g_{best} , 则根据公式 (12), g_{best} 的反向解 \bar{g}_{best} 为

$$\bar{g}_{best} = 2 \cdot r \cdot M_{cen} - g_{best} \quad (12)$$

该策略利用公式 (10) 计算种群重心, 根据公式 (11) 计算各个维度上粒子的反向解, 通过比较各个维度上反向解和全局最优解的适应度值, 得到全局最优位置。最后根据公式 (12) 计算出全局最优位置的反向解 \bar{g}_{best}_i 。根据以上步骤, 每次迭代生成的反向位置最多可达到 2^D 个, 每生成一个反向解就更新全局最优位置, 这样充分利用种群信息找到精度更高的解, 获得优化问题的最优解的概率更大。

2.2.2 种群熵的定义

对于反向学习, 常采用随机确定间隔代数或每次迭代启动, 这样未在合适的时机或频繁启动反向学习策略, 造成计算资源的浪费, 增加算法在搜索过程陷入局部最优的风险。本文引入种群熵指标表征粒子的聚集程度和多样性, 从而根据熵值判断种群是否陷入局部最优, 从而启动反向学习。

基于粒子群算法特征, 本文提出根据粒子适应度值计算种群熵的公式 (Population Entropy, PE):

$$PE = \frac{1}{\exp(F_b \cdot N - F_s) + 1} \quad (13)$$

其中, F_b 是当前最优适应度值; F_s 是当前所有粒子的适应度值之和; PE 可以反映种群的聚集程度和多样性。

以 30 维 Quadric 函数的种群熵变化为例 (见图 2), 分析启动反向学习策略前后的种群熵 PE 值变化情况。算法搜索前期, 粒子进行全局探索, 分布均匀且多样性好, PE 值保持在 1; 随着迭代的进行,

PE 值出现下降趋势, 粒子快速向局部聚集, 多样性降低; 当迭代到 120 代时, PE 值下降到一定程度, 算法启动各维重心反向变异策略以跳出局部最优, 改善种群多样性, 迭代后期种群收敛, PE 维持在 0.5。

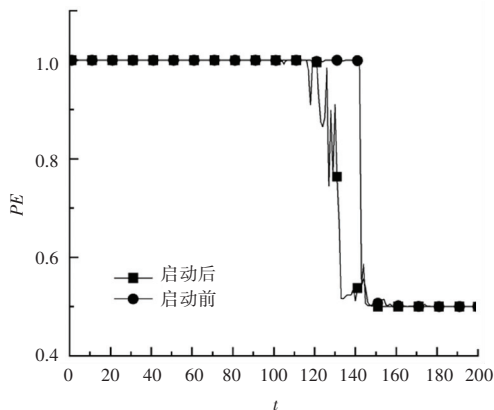


图 2 种群熵变化曲线

Fig. 2 Population entropy variation

从种群熵 PE 变化曲线可以看出, 在迭代前期种群熵的值比较平稳, 到某点急速下降, 此时全局搜索完成, 种群多样性开始降低。当种群熵下降到初始种群熵的一定比例 λ 时, 大部分粒子开始进入局部搜索状态, 本文中 λ 取 0.95^[12]。当种群陷入局部最优无法进一步探索, 启动 DCRMS 操作。对于种群陷入局部最优状态的判断采用公式 (14), 条件连续满足次数达到 Count。

$$|PE(t+1) - PE(t)| \leq 0.01 \quad (14)$$

2.3 算法步骤

步骤 1 初始化种群。在解空间, 随机产生 N 个粒子的位置 x_i 和速度 v_i , 计算粒子的适应度值。设定粒子速度的最大值 V_{\max} 和最小值 V_{\min} , 位置的最大值 X_{\max} 和最小值 X_{\min} , 学习因子 c_1, c_2 , 维度 D , 最大迭代次数 T 。

步骤 2 采用动态多种群划分策略, 将粒子群划分为“top 粒子”、“次 top 粒子”和“弱 top 粒子”, 并计算出初始种群熵。

步骤 3 根据公式 (1) 和公式 (6) 更新“top 粒子”的速度和位置; 公式 (1) 和公式 (7) 更新“次 top 粒子”的速度和位置; 公式 (1) 和公式 (9) 更新“弱 top 粒子”的速度和位置。

步骤 4 重新计算粒子的适应度值, 更新种群的全局最优位置 g_{best} 和粒子的个体最优位置 p_{best} 。

步骤 5 计算当次迭代的种群熵, 判断算法是否启动各维重心反向变异策略, 若是, 则按照公式

(12)更新全局最优位置,否则转至步骤 6。

步骤 6 判断算法是否满足迭代的终止条件,若满足,则转至下一步,否则转至步骤 2 进行下一步迭代寻优。

步骤 7 输出全局最优值,算法结束。

2.4 算法时间复杂度分析

从算法步骤来分析 DMP SO-Perl 算法的时间复杂度。由单次种群搜索行为分析可知,种群初始化的时间复杂度为 $O(N)$;计算所有粒子适应度值以及更新 $pbest$ 和 $gbest$,其时间复杂度为 $O(N)$;接下来对粒子群进行动态划分种群,形成 3 种种群的粒子,以此便于粒子间信息流通,其时间复杂度为 $O(K)$, K 是常数且 $K \geq 1$ 。在个体行为更新中,与基本 PSO 算法稍有不同,“top 粒子”、“次 top 粒子”和“弱 top 粒子”的位置速度更新的复杂度分别为 $O(D)$ 、 $O((N \cdot \beta - 1) \cdot D)$ 和 $O((N \cdot (1 - \beta)) \cdot D)$;更新粒子适应度的时间复杂度为 $O(N \cdot D)$ 。按算法的流程依次进行计算: DMP SO-Perl 算法的时间复杂度为 $\max\{O(N), O((N \cdot \beta - 1) \cdot D), O((N \cdot (1 - \beta)) \cdot D), O(N \cdot D)\}$ 。由于 $\beta < 1$,可知 DMP SO-Perl 算法的时间复杂度为 $O(N \cdot D)$ 。

可见,在固定迭代次数 T 下, DMP SO-Perl 算法的时间复杂度为 $O(T \cdot N \cdot D)$,因此本文算法的时间复杂度与基本 PSO 算法的时间复杂度 $O(T \cdot$

$N \cdot D)$ 相同。

3 实验分析

3.1 测试函数与实验设计

为检验算法有效性,本文选取 8 个标准测试函数(见表 1),其中包括单峰函数($f_1 - f_2$)和多峰函数($f_3 - f_8$)。

表 1 测试函数

Table 1 Test functions

特征	测试函数	函数名称	搜索空间	极值 F^*
单峰函数	f_1	Sphere	$[-100, 100]$	0
	f_2	Quadric	$[-100, 100]$	0
多峰函数	f_3	Rosenbrock	$[-2.084, 2.084]$	0
	f_4	Griewank	$[-600, 600]$	0
	f_5	Ackley	$[-32, 32]$	0
	f_6	Rastrigin	$[-5, 5]$	0
	f_7	Schwefel	$[-500, 500]$	0
	f_8	Weierstrass	$[-0.5, 0.5]$	0

本文将进行 3 组实验测试:参数 λ 的敏感性分析;算法构造策略的有效性实验;与其它改进的 PSO 算法在 30 维和 50 维的性能对比。所有实验基于 MATLAB2015b,在平台采用 win10 (64b), Intel (R) Core(TM) i7-9750H,主频 2.60 GHz; RAM 16.00 GB 环境下统一测试。实验中算法参数设置见表 2,对各测试函数独立运行 30 次进行统计分析。

表 2 算法参数设置

Table 2 Algorithm parameter settings

算法	参数设置
GA	$P_c = 0.7, P_m = 0.3$
SPSO	$c = 1, \omega_{max} = 0.9, \omega_{min} = 0.4$
SAPSO	$c = 1, \omega_{max} = 0.9, \omega_{min} = 0.4$
CLPSO	$c = 1, \omega_{max} = 0.9, \omega_{min} = 0.4, m = 5$
DMP SORH	$c_1 = 1, c_2 = 0.2, c_3 = 0.3, w = 0.8 * \exp(-0.5)^{(t/T)}$
HPSO-TS	$c = 1, \omega_{max} = 0.9, \omega_{min} = 0.4, k = 5$
DMP SO-Perl	$c = 1, \omega_{max} = 0.9, \omega_{min} = 0.4, \lambda = 0.95, \text{Count} = 20$

3.2 参数敏感性分析

在 DMP SO-Perl 算法中 Count 是决定启动 DCRMS 策略时机的重要参数,也影响算法的收敛速度。为了定量分析 Count 取值对算法的影响,本文采用 8 个标准测试函数,实验测试其敏感性。其中,Count 从 60 次到 20 次间隔 10 取值,30 次独立运行

实验的部分实验结果见表 3。

从实验数据可见,Count 取不同值时,函数 f_3 的结果表现出最优值和最差值相差较大,函数 f_7 得出的实验结果相差不大。由于函数 f_1, f_2, f_4, f_5, f_6 和 f_8 的结果相同且全为 0,因此未在表格中展示。综合对比,当 Count = 20 时,算法的整体效果最佳。

表3 不同 Count 值测试函数结果对比 ($D=30$ 维)Table 3 Comparison of test function results for different count values ($D=30$ dimension)

Func	Court	Max	Min	Mean
f_3	60	2.87E+01	3.94E-07	1.90E+01
	50	2.87E+01	6.44E-07	9.43E+00
	40	2.87E+01	1.12E-06	1.79E+01
	30	2.87E+01	1.72E-07	1.65E+01
	20	2.87E-04	1.22E-06	4.12E-05
f_7	60	7.71E-04	3.82E-04	4.38E-04
	50	6.18E-04	3.82E-04	4.23E-04
	40	9.40E-04	3.82E-04	4.67E-04
	30	8.52E-04	3.82E-04	4.38E-04
	20	1.22E-03	3.82E-04	4.72E-04

3.3 算法构造策略有效性实验

本节将在标准 PSO 算法的基础上,分别添加算法策略,验证其有效性和策略叠加对算法搜索性能的影响。其中,SPSO 是标准的粒子群算法;DMP SO 是基于标准粒子群算法添加动态划分多种群策略;PSO-PERL 是基于标准粒子群算法添加种群熵启动反向学习策略;DMP SO-PERL 是基于标准粒子群算法的框架加入以上两种改进策略。各算法在 8 个测试函数下搜索结果见表 4。

表4 算法构造策略的有效性实验数据 ($D=30$ 维)Table 4 Effectiveness data of algorithm construction strategy ($D=30$ dimension)

	SPSO	DMP SO	PSO-PERL	DMP SO-PERL
f_1	6.02E-03	0	8.17E-106	0
f_2	2.53E+00	1.61E-200	1.13E-37	0
f_3	2.54E+01	1.91E+01	2.88 E+01	4.12E-05
f_4	3.99E-02	0	0	0
f_5	3.01E+00	0	1.78E-15	0
f_6	4.19E+01	0	1.35 E+01	0
f_7	6.22E+03	5.22E-05	6.88E+03	4.72E-04
f_8	1.07E+01	0	0	0

在相同迭代次数下,本文提出的算法 DMP SO-PERL 在 8 个测试函数中取得了 8 个最优结果,从数据上可以看出 DMP SO-PERL 算法表现最优。DMP SO 算法和 PSO-PERL 算法分别取得 5 个和 2 个最优结果,SPSO 算法未取得最优结果。

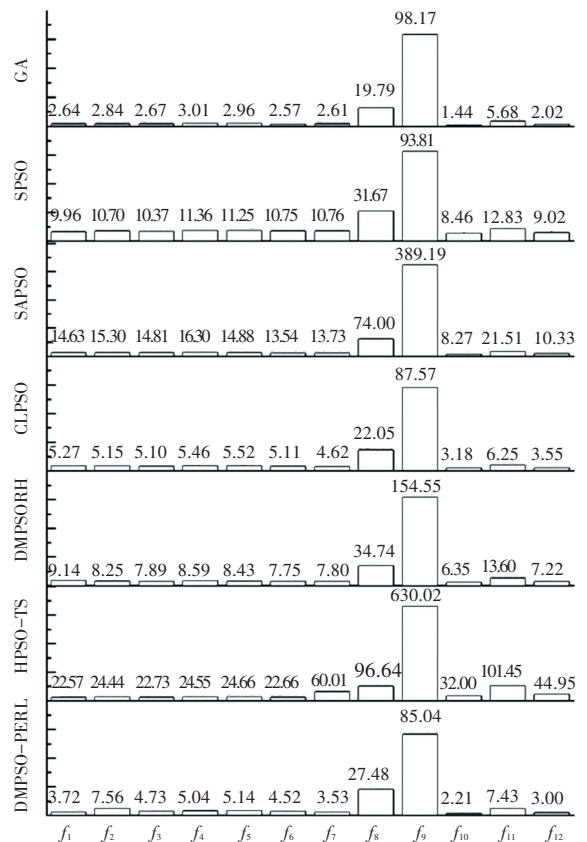
实验结果表明, DMP SO-PERL 算法在添加动态多种群划分策略后,在搜索前期能帮助粒子勘探较大的解空间,在搜索后期采用的基于种群熵启动重心反向变异策略有效地帮助粒子跳出局部最优,

使得算法搜索到更优精度的解。仅添加动态多种群划分策略的 DMP SO 算法,在搜索前期进行大范围的勘探,锁定较好解的区域,但是局部搜索挖掘能力不足,导致算法搜索精度不高。添加种群熵启动反向学习策略能帮助粒子有效地挖掘局部解的信息,但是 PSO-PERL 算法前期勘探能力较弱,无法搜索到较好解的区域易陷入局部最优。

综上所述,本文提出的两种策略都有积极作用,并且将其融合能够促进算法在性能上大幅度提升,取得较为满意的计算结果。

3.4 与其他算法对比分析

为了进一步验证本文算法的有效性,选择 GA、SPSO^[18]、SAPSO^[19]、CLPSO^[20]、DMP SO RH^[11]、HPSO-TS^[16]与本文算法进行对比。所有算法的种群规模 $N=30$,最大迭代次数 $T=3000$ 。根据相关文献[18-20],算法参数设置见表 2,本次实验分别对 $D=30$ 和 $D=50$ 维数进行对比。其中, $D=30$ 的算法运行时间如图 3 所示,实验结果见表 5; $D=50$ 的收敛性曲线见图 4。由于两个维度下所得结论相同,本文选择一个维度来展示。

图3 算法运行时间对比 ($D=30$ 维)Fig. 3 Comparison of running time ($D=30$ dimension)

3.4.1 优化性能分析

表 5 中, 单峰函数 f_1, f_2 在测试中, DMPSO-PERL 算法和 HPSO-TS 算法均取得理论上的最优结果。在多峰函数测试中, DMPSO-PERL 算法取得了 4 个 (f_4, f_5, f_6, f_8) 理论上的最优结果, 2 个 (f_3, f_7) 对比算法中的最优值。虽然 HPSO-TS 算法也取得 3 个 (f_4, f_6, f_8) 最优值, 但在函数 f_3, f_5, f_7 中, DMPSO-PERL 算法表现更优。特别在函数 f_5 上, 本文算法每次独立实验中都收敛到最优, 在求解质量和稳定性上表现最好; 在函数 f_7 上, CLPSO

算法、DMPSORH 算法和本文算法所得到的最优解表现最好, 但稳定性不如本文算法。

DMPSO-PERL 算法在 8 个标准测试函数上均能取得较好的求解结果, 单峰函数表明了算法搜索能力, 多峰函数同时存在多个局部最优解, 测试了算法的全局探索开发能力。与其他算法相比, 本文算法有着良好的收敛精度和稳定性, 在迭代后期均启动了变异策略帮助粒子跳出局部最优, 有较强的全局搜索能力和勘探能力。

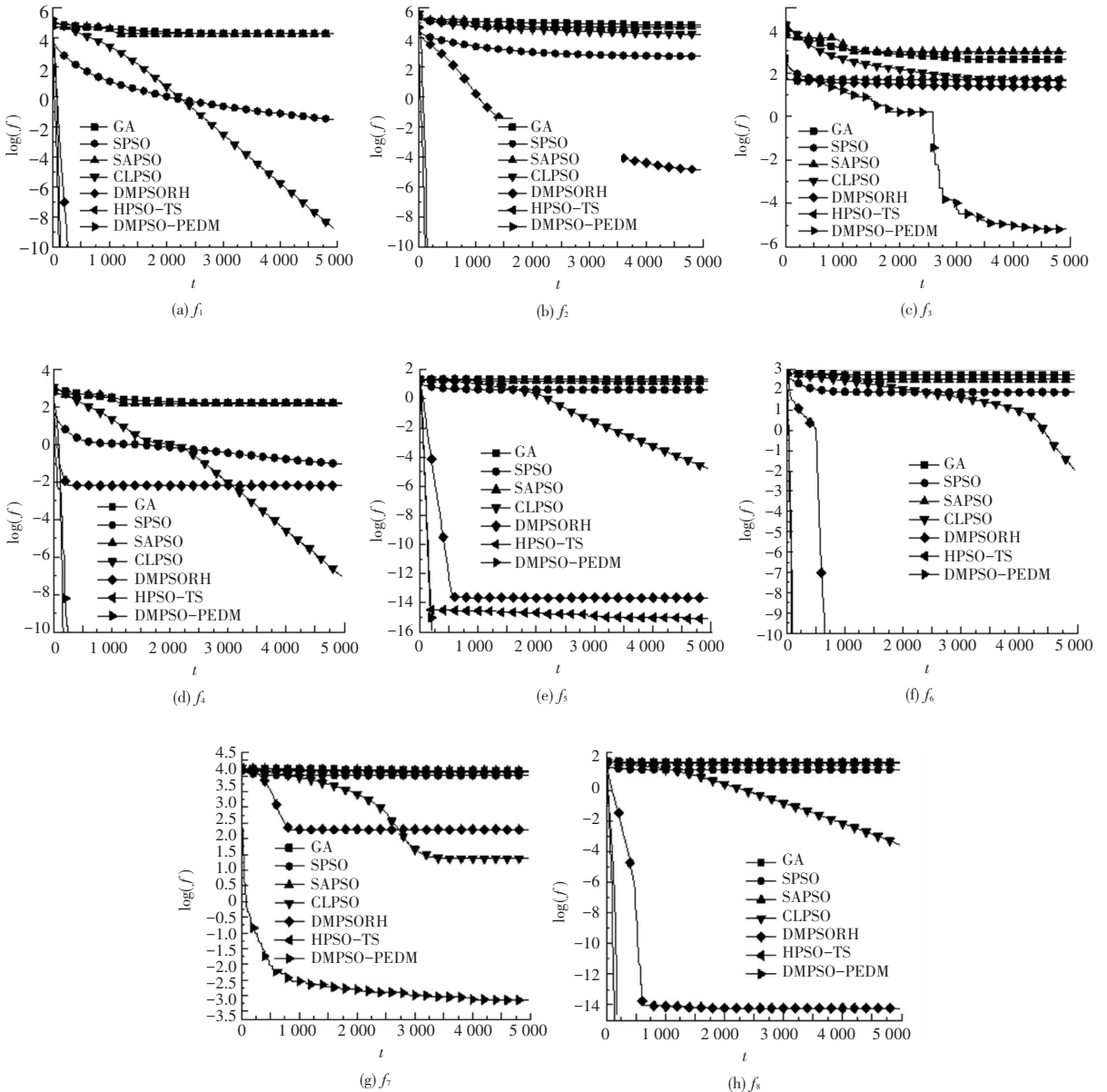


图 4 8 个函数测试曲线图 ($D=50$ 维)

Fig. 4 Curves of eight test functions ($D=50$ dimension)

表5 函数测试实验结果($D=30$ 维)Table 5 Experimental results of function test ($D=30$ dimension)

		GA	SPSO	SAPSO	CLPSO	DMPORH	HPSO-TS	DMPPO-PERL
f_1	Max	8.63E+01	1.71E-02	1.61E+04	1.84E-09	8.90E-108	0	0
	Min	1.12E+00	1.50E-03	4.22E+03	2.35E-10	5.00E-126	0	0
	Mean	1.20E+01	6.02E-03	9.41E+03	8.53E-10	3.10E-109	0	0
	SD	1.70E+01	4.04E-03	3.10E+03	4.01E-10	1.60E-108	0	0
f_2	Max	4.41E+04	5.78E+00	5.03E+04	4.55E+03	1.50E-06	0	0
	Min	1.79E+04	5.87E-01	5.67E+03	1.56E+03	1.04E-08	0	0
	Mean	2.85E+04	2.53E+00	1.69E+04	3.02E+03	3.54E-07	0	0
	SD	6.02E+03	1.37E+00	9.66E+03	7.15E+02	3.89E-07	0	0
f_3	Max	2.93E+02	2.94E+01	1.19E+03	3.86E+01	2.27E-02	2.90E+01	2.75E-04
	Min	1.46E+01	2.22E+01	1.71E+02	1.73E+01	3.57E-04	2.89E+01	1.22E-06
	Mean	1.52E+02	2.54E+01	4.31E+02	2.36E+01	7.46E-03	2.89E+01	4.12E-05
	SD	7.05E+01	1.45E+00	1.87E+02	3.86E+00	5.96E-03	3.05E-02	6.13E-05
f_4	Max	2.89E+00	1.19E-01	1.25E+02	2.76E-06	7.06E-02	0	0
	Min	1.43E+00	1.40E-02	3.84E+01	3.98E-08	0	0	0
	Mean	1.84E+00	3.99E-02	7.14E+01	7.08E-07	7.52E-03	0	0
	SD	3.33E-01	2.14E-02	1.93E+01	7.14E-07	1.43E-02	0	0
f_5	Max	2.07E+01	4.17E+00	2.00E+01	2.72E-05	4.26E-14	3.55E-15	0
	Min	1.93E+01	2.01E+00	1.22E+01	9.75E-06	3.55E-15	0	0
	Mean	2.01E+01	3.01E+00	1.67E+01	1.62E-05	1.84E-14	9.47E-16	0
	SD	3.26E-01	5.35E-01	2.56E+00	3.51E-06	8.66E-14	1.57E-15	0
f_6	Max	2.86E+02	7.26E+01	1.96E+02	1.04E-02	0	0	0
	Min	1.58E+02	2.59E+01	1.18E+02	2.49E-04	0	0	0
	Mean	2.33E+02	4.19E+01	1.51E+02	2.33E-03	0	0	0
	SD	3.13E+01	1.18E+01	1.88E+01	2.12E-03	0	0	0
f_7	Max	7.60E+03	8.15E+03	9.06E+03	1.18E+02	3.55E+03	9.34E+03	1.22E-03
	Min	4.72E+03	4.56E+03	6.41E+03	3.82E-04	3.82E-04	4.05E+03	3.82E-04
	Mean	6.15E+03	6.22E+03	8.14E+03	3.95E+00	2.37E+02	8.08E+03	4.72E-04
	SD	7.54E+02	1.04E+03	5.58E+02	2.13E+01	9.01E+02	1.15E+03	1.56E-04
f_8	Max	4.10E+01	1.68E+01	4.18E+01	2.66E-04	2.84E-14	0	0
	Min	3.16E+01	7.46E+00	3.57E+01	1.12E-04	0	0	0
	Mean	3.67E+01	1.07E+01	3.92E+01	2.00E-04	1.89E-15	0	0
	SD	2.54E+00	2.03E+00	1.63E+00	4.44E-05	6.17E-15	0	0

3.4.2 算法运行时间分析

如图3所示,本文提出的DMPPO-PERL算法在所有基于粒子群改进的对比算法中,取得6个(f_1 、 f_3 - f_7)最快运算速度,2个(f_2 和 f_8)次快运行速度。GA在所有算法中运行时间最短,但是其搜索精度远远不足。

综合而言,本文提出的采用种群熵启动反向学习策略,能够利用种群熵这一指标衡量粒子的聚散

程度,当陷入局部最优时启动反向学习策略更快跳出局部最优,在保证算法搜索精度的同时,节省了计算资源,在一定程度上提升算法的运行效率。

3.4.3 收敛性能分析

图4给出7种算法在50维度下的测试函数收敛曲线。对于大多数测试函数,DMPPO-PERL算法的收敛曲线在迭代一开始迅速下降,表明基于狮群算法的动态多种群划分策略有效提高了算法的搜索

范围和收敛速度。在迭代过程中,DMP SO-PERL 算法陷入停滞的次数远少于其他优化算法,仅在 f_3 出现了较为明显的停滞。当函数 f_3 迭代至2 600代左右时,本文算法快速收敛,表明当启动反向变异策略后,能快速找到精度更高的解。在函数 f_7 上,本文算法在收敛速度和精度上表现最佳,而其余算法收敛精度不理想,说明本文算法所采用的策略能提高算法收敛效率。

4 结束语

针对基本 PSO 算法容易陷入局部最优和求解精度不高等问题,本文研究了不同种群粒子的迭代机制和迭代后期的搜索需求,提出了一种采用种群熵启动反向学习动态多种群粒子群算法。算法借鉴狮群算法的思想,对粒子进行动态多种群划分以实施不同的位置更新机制,而通过定义了种群熵作为启动各维重心反向变异策略的判断条件,提升搜索后期各子群多样性。复杂测试函数实验的结果表明,本文算法比所选同类算法具有更高的求解精度和更好的稳定性和收敛速度。下一步将针对算法在复杂工程模型求解展开应用研究。

参考文献

- [1] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]// Proceedings of the Sixth International Symposium on Micro Machine and Human Science.1995:39-42.
- [2] KENNEDY J, EBERHART R. Particle swarm optimization[C]// Proceedings of IEEE International Conference on Neural Networks. 1995:1942-1948.
- [3] ZHAO J, LV L, WANG H, et al. Particle swarm optimization based on vector Gaussian learning[J]. Ksii Transactions on Internet and Information Systems, 2017, 11(4):2038-2057.
- [4] 孙辉,邓志诚,赵嘉. 混合均值中心反向学习粒子群优化算法

- [J]. 电子学报, 2019, 47(9):1809-1818.
- [5] 刘天宇,王翥. 一种多样性控制的多目标粒子群算法[J]. 西安电子科技大学学报, 2021, 48(3):106-114.
- [6] 李文锋,梁晓磊,张煜. 具有异构分簇的粒子群优化算法研究[J]. 电子学报, 2012, 40(11):2194-2199.
- [7] JIANG J J, WEI W X, SHAO W L, et al. Research on large-scale Bi-Level particle swarm optimization algorithm[J]. IEEE Access, 2021, 9:56364-56375
- [8] CHEN Y G, LI L X, PENG H P, et al. Particle swarm optimizer with two differential mutation[J]. Applied Soft Computing, 2017, 61:314-329.
- [9] 周凌云,丁立新,彭虎. 一种邻域重心反向学习的粒子群优化算法[J]. 电子学报, 2017, 45(11):2815-2824.
- [10] 夏学文,刘经南,高柯夫. 具备反向学习和局部学习能力的粒子群算法[J]. 计算机学报, 2015, 38(7):1397-1407.
- [11] 唐可心,梁晓磊,周文峰. 具有重组学习和混合变异的动态多种群粒子群优化算法[J]. 控制与决策, 2021, 36(12):2871-2880.
- [12] CHEN K, ZHOU F, YIN L, et al. A hybrid particle swarm optimizer with sine cosine acceleration coefficients[J]. Information Sciences, 2018, 422:218-241.
- [13] 曹玉莲,李文锋,张煜. 基于拟熵自适应启动局部搜索策略的混合粒子群算法[J]. 电子学报, 2018, 46(1):110-117.
- [14] 孙骞,高岭,刘涛. 基于熵模型的粒子群优化算法[J]. 东南大学学报:自然科学版, 2019, 49(6):1088-1093.
- [15] 刘生建,杨艳,周永权. 一种群体智能算法—狮群算法[J]. 模式识别与人工智能, 2018, 31(5):431-441.
- [16] 周文峰,梁晓磊,唐可心. 具有拓扑时变和搜索扰动的混合粒子群优化算法[J]. 计算机应用, 2020, 40(7):1913-1918.
- [17] 朱蓉,靳雁霞,范卫华. 融合优质粒子分布的粒子群优化算法[J]. 小型微型计算机系统, 2015, 36(3):576-580.
- [18] 徐利锋,黄祖胜,杨中柱. 引入多级扰动的混合型粒子群优化算法[J]. 软件学报, 2019, 30(6):1835-1852.
- [19] 方一鸣,郑贺军,刘乐. 基于模拟退火-粒子群算法优化支持向量机参数的连铸漏钢预报[J]. 中国机械工程, 2017, 28(12):1462-1467.
- [20] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive Learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3):281-295.