

基于多种群粒子群算法和布谷鸟搜索的联合寻优算法

高云龙, 闫鹏

(厦门大学 信息技术与科学学院, 福建 厦门 361005)

摘要: 为了提高动态多种群粒子群(DMS-PSO)算法的全局搜索能力, 将布谷鸟搜索算法(CS)引入DMS-PSO算法中, 提出DMS-PSO-CS算法. 采用中位数聚类算法将整个种群动态划分为若干小种群, 各个小种群作为底层种群通过PSO算法进行寻优, 再将每个小种群中的最优粒子作为高层种群的粒子通过CS算法进行深度优化. 将所提出算法应用于CEC 2014测试函数, 并与CS算法和其他改进的PSO算法进行比较. 实验结果表明, 所提出算法能够显著提高全局搜索能力和算法效率.

关键词: 粒子群算法; 动态多种群; 布谷鸟搜索; 中位数聚类

中图分类号: TP273

文献标志码: A

Unified optimization based on multi-swarm PSO algorithm and cuckoo search algorithm

GAO Yun-long, YAN Peng

(School of Information Science and Engineering, Xiamen University, Xiamen 361005, China. Correspondent: GAO Yun-long, E-mail: gaoyl@xmu.edu.cn)

Abstract: In order to improve the global search ability of the dynamic multi-swarm PSO(DMS-PSO) algorithm, the cuckoo search(CS) algorithm is introduced into DMS-PSO algorithm, the algorithm named DMS-PSO-CS is proposed, which is a bi-layer optimal algorithm. The whole swarm is dynamically divided into several small populations by using the median clustering algorithm, and each small population is optimized by using PSO algorithm in the bottom layer. The best particle from each small population is selected as a member of the swarm in the top layer, then the top swarm is deeply optimized by using CS algorithm. The DMS-PSO-CS algorithm is employed to solve the CEC 2014 test functions. In comparison with the CS algorithm and other improved PSO algorithms, the experimental results show that the proposed algorithm can obviously improve the global search ability and computation efficiency.

Keywords: particle swarm optimization; dynamic multi-swarm; cuckoo search; median clustering

0 引言

自Kennedy等^[1-2]提出PSO算法以来, 该算法得到了广泛的发展和应^[3-6]. 针对标准PSO算法容易陷入局部最优和精度较低的问题, Kennedy等^[7]提出了局部版本的PSO算法, 将邻域的最优值作为粒子的学习对象间接地将最优值在整个粒子群中传播, 增加了整个种群的多样性; Mendes等^[8]提出了基于fully-informed策略的FIPSO算法, 算法根据环形、金字塔形、方形等不同的拓扑结构来选取不同的邻域, 每个粒子利用全部邻域粒子的加权信息进行寻优.

多种群PSO算法也是一种特殊结构的局部PSO算法, 根据一定的组合方式将粒子群划分为若干个

小种群, 每个小种群构成一个邻域, 各小种群在各自所在邻域并行寻优. 因为PSO算法是由社会行为演化而来, 在社会中每个团体都有自己的思想和行为, 每个人跟随团体的意志来调整自己的思想和行为, 在算法中每个小种群跟随核心粒子进行寻优, 每个小种群之间通过整个种群的最优值进行联系和交流. 多种群PSO算法是一种分布式进化算法, 通过多个子种群的各自进化执行协同搜索, 并通过种群迁徙实现搜索信息的共享. 多种群PSO算法基于个体的相似度对种群进行划分或动态处理, 不同子种群负责搜索不同的区域, 多种群通过协同搜索维持了种群的多样性^[9].

文献[10]提出了DMS-PSO算法, 算法将整个种

收稿日期: 2015-03-23; 修回日期: 2015-06-23.

基金项目: 国家自然科学基金项目(61203176); 福建省自然科学基金项目(2013J05098).

作者简介: 高云龙(1979—), 男, 副教授, 博士, 从事机器学习和电力调度等研究; 闫鹏(1990—), 男, 硕士生, 从事智能算法和电力调度的研究.

群划分为若干个小种群, 周期性地随机划分和重组小种群, DMS-PSO 比其他的一些局部粒子群效果要好, 在多峰问题上表现出良好的搜索能力, 但随机重组小种群具有盲目性; 文献 [11] 在 DMS-PSO 算法的基础上引入了准牛顿法, 周期性地提炼最优值在前 25% 的粒子并进行局部搜索, 从实验结果可以看到, 算法在复杂多维问题上有较好的表现; 文献 [12] 提出了一种将 DMS-PSO 算法与和声算法 (SHS) 相结合的 DMS-PSO-SHS 算法, 周期性地将 HS 运用到小种群中所有的历史最优解进行优化; 文献 [13] 提出了一种 K -均值聚类的 KDMSPSO 算法, 该算法通过 K -均值聚类算法将种群划分为多子群, 并对子群进行动态重组以增强子群间的信息交流; 文献 [14] 提出了一种基于多种群分层的 HSPPSO 算法, 在每个小种群中用 PSO 算法优化之后再每个小种群的最优粒子重新组成一个新的粒子群, 再次用 PSO 算法进行优化, 将得到的最优值传递给每个小种群, 通过两次 PSO 算法优化提高了算法的搜索能力. 多种群粒子群算法在求解多峰问题上比其他一些 PSO 的变形算法表现更好, 但是全局搜索能力仍然有待改进.

CS 算法是由 Yang 等^[15]在 2009 年根据模拟布谷鸟寻窝产卵的行为提出的一种新型群智能算法. CS 算法因其参数少、易实现等特点已被成功应用于求解各类最优化问题^[16-17]. CS 算法一个很显著的特点就是采用 Lévy 飞行产生新解, Lévy 飞行的高度随机性能使搜索过程遍布整个搜索空间, 使得算法的全局搜索能力较强. 但是 Lévy 飞行高度的随机性造成 CS 算法在局部进行精细搜索的能力较差, 算法收敛较慢.

目前, 已有一些学者将 CS 算法引入到 PSO 算法中, 将两种算法相结合用于求解实际应用问题. 文献 [18] 利用 PSO 算法的快速收敛特性为布谷鸟搜索寻找更好的宿主鸟巢提出了一种基于粒子群的布谷鸟聚类算法, 该算法用于解决 Web 文本聚类; 文献 [19] 提出了 PSO-CS 算法, 该算法调整了布谷鸟算法对待最差鸟巢的做法, 将最差鸟巢作为 PSO 算法的一个粒子, 最好的鸟巢作为最优解进行寻优.

结合以上研究成果, 本文在 DMS-PSO 算法中引入 CS 算法, 提出一种基于多种群粒子群和布谷鸟搜索的联合寻优算法 (DMS-PSO-CS). 该算法结合 DMS-PSO 算法和 CS 算法各自的优势来平衡局部搜索与全局搜索, 并采用中位数聚类算法将整个粒子群动态划分为若干小种群, 以有效避免随机重组小种群的盲目性, 进而加快收敛速度, 提高算法精度.

1 多种群 PSO 算法和 CS 算法

1.1 PSO 算法

PSO 算法是基于群智能算法的一种启发式算法. 根据自然界中群鸟捕食的特点, 鸟群中个体在飞行过

程中根据自己的经验和对其他个体的学习来不断更新和调整自己的位置, 并最终找到食物的位置. 在实际优化问题中, PSO 算法中每个粒子都代表优化问题的一个解, 每个粒子对应一个由适应度函数决定的适应度值. 粒子的速度决定了粒子移动的方向和距离, 并随自身及其他粒子的移动经验进行动态调整, 从而实现粒子群在可行解空间的寻优过程.

假设在 D 维的搜索空间中, 有 N 个粒子, 每个粒子的位置表示为 $[x_1, x_2, \dots, x_d]$, 相应的速度表示为 $[V_1, V_2, \dots, V_d]$. 粒子的速度和位置的迭代公式为

$$V_{id}^{k+1} = wV_{id}^k + c_1r_1(\text{pbest}_{id} - x_{id}^k) + c_2r_2(\text{gbest}_d - x_{id}^k), \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + V_{id}^{k+1}. \quad (2)$$

其中: w 为惯性权重, c_1 为自我学习因子, c_2 为社会学习因子, r_1 和 r_2 为 $[0, 1]$ 区间的随机数, pbest_{id} 为第 i 个粒子在第 d 维空间的个体历史最优值, gbest_d 为种群在第 d 维的历史最优值, x_{id}^k 为第 i 个粒子在第 d 维空间的第 k 次迭代位置, V_{id}^k 为第 i 个粒子在第 d 维空间的第 k 次迭代速度. 速度迭代式 (1) 的等式右侧包括 3 项: 第 1 项为粒子对当前速度的继承, 第 2 项为粒子的自我学习能力, 第 3 项为粒子的社会学习能力; 位置迭代式 (2) 的等式右侧为当前位移和改变的速度矢量和.

1.2 多种群 PSO 算法

多种群 PSO 算法来源于局部 PSO 算法, 在实际优化问题中, 大多数要解决的都是多峰问题, 除了最优点外还有多个极值点. 而 PSO 算法的优点是善于求解单峰问题, 并能迅速收敛到最优解. 但是对于多峰问题, PSO 算法会更容易陷入局部最优解并使粒子群停滞收敛. 为了增加种群的多样性, Kennedy 提出了局部 PSO 算法. 多种群 PSO 算法是一种基于邻域的特殊拓扑结构的局部 PSO 算法, 多种群 PSO 算法将整个粒子群划分为多个小种群, 相应的解空间也被划分为对应的子空间, 每个小种群在其所在的子空间寻找最优点. 通过并行计算增加计算效率, 同时也增强了 PSO 算法的全局搜索能力, 每个小种群之间通过整个种群最优值进行间接交流, 加强了粒子群的多样性.

1.3 中位数动态聚类

种群的划分目前采用固定种群和动态种群. 在采取固定种群的多种群 PSO 算法^[14]中, 小种群之间的交流过于局限, 整个种群的多样性比较单一. 每个小种群的粒子都限定在初始分配的种群中, 这样限制了粒子的自由, 在加强局部搜索能力的同时减弱了全局搜索能力. 文献 [13] 采用了 K -均值聚类算法对小种群进行动态划分, 并周期性地对小种群进行更新, 算法比固定种群采取随机重组小种群算法的多样性好, 但是对于每次调整, 各个聚类的粒子数量都是一样的.

实际上,各小种群的自我学习和社会学习能力的不同会导致各个种群的凝聚力不同.优秀的种群会吸引更多的粒子加入;相反,表现较差的种群其粒子数量可能越来越少.针对算法后期粒子分布可能不均匀的情况,本文采用中位数聚类划分小种群,周期性地对小种群进行调整,粒子根据与各聚类中心的相似度自由确定自己的所属种群,从而使得每个小种群可以自由竞争,加快收敛速度和搜索效率.算法的思想是:每次迭代过程中采用中位数聚类算法计算各粒子到每个聚类中心的距离,从而确定以每个聚类中心为核心的小种群,包括每个小种群所包含粒子的数量以及各粒子的位置和速度信息.

中位数动态聚类算法流程.

Step 1: 初始化. 随机选取 K 个点作为聚类中心.

Step 2: 更新聚类. 计算各点到各个聚类中心的距离,将这个点和与它最近的聚类中心划分为一个聚类.

Step 3: 计算每个聚类中的点在每一维的中位数,并将中位数作为下一次聚类的聚类中心.

Step 4: 判断是否达到调整周期,若到达,则随机更新聚类中心,否则进行 Step 5.

Step 5: 重复 Step 2 和 Step 3,直至达到终止条件.

1.4 CS 算法

CS 算法是受自然启发的一种新型群智能算法,CS 算法是模拟布谷鸟寻找适合产卵的鸟巢位置的过程.布谷鸟有种特殊能力,他们挑选其他宿主的鸟巢来下蛋,并且这些鸟巢中通常有宿主刚产下的蛋.布谷鸟将蛋下在这些蛋附近,或者扔掉宿主产的蛋来增加他们自己的蛋的成功孵化概率,但是宿主可能发现这些鸟蛋并把他们扔掉或者重新再建一个鸟巢.

布谷鸟孵蛋过程是一个非常复杂的过程,CS 算法是根据简化的布谷鸟孵蛋过程演化而来,在实际优化问题中,每个寻优问题的解代表布谷鸟产的一个蛋,寻优问题的适应度代表布谷鸟产蛋的质量.

CS 算法主要遵循以下 3 个原则: 1) 每只布谷鸟一次只能产一只蛋,并随机选择鸟巢来孵化; 2) 在随机挑选的一组鸟巢中,最好的鸟巢将会保留到下一代; 3) 可利用的鸟巢数量 n 是固定的,一个鸟巢的主人能发现外来鸟蛋的概率 P 属于 $[0, 1]$.

当宿主发现这些蛋,他们可能将其扔掉,也可能在一个新的地方重新建立鸟巢,如果没有被发现,这些鸟蛋将会成功孵化并通过 Lévy 飞行选择新的孵化地点.

在布谷鸟算法中,假设在 D 维搜索空间中有 N 个鸟蛋,第 k 次迭代第 i 个鸟蛋的位置为 e_i^k ,布谷鸟的路径和位置更新公式如下:

$$e_i^{k+1} = e_i^k + \delta_i. \quad (3)$$

其中: δ_i 为需要进行的位置变化量,且有

$$\delta_i = 0.01 \times s_i \oplus (e_i^k - e^{\text{best}}). \quad (4)$$

其中: \oplus 为点乘, e^{best} 为目前找到的最好鸟巢.

随机步长由对称 Lévy 分布产生,通过 Mantegna 算法产生 s_i , 即

$$s_i = \frac{u}{|v|^{1/\beta}}. \quad (5)$$

其中: $u(u_1, u_2, \dots, u_d), v(v_1, v_2, \dots, v_d)$ 为 D 维空间的向量,且有 $\beta = 3/2$. u 和 v 的每个分量服从如下正态分布:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2); \quad (6)$$

$$\sigma_u \sim \left(\frac{\Gamma(1 + \beta) \cdot \sin(\pi \cdot \beta/2)}{\Gamma((1 + \beta)/2) \cdot \beta \cdot 2^{(\beta-1)/2}} \right)^{1/\beta}, \sigma_v = 1. \quad (7)$$

Lévy 飞行包括随机定向且无特征尺度的直线运动序列,每段序列的步长满足重尾分布,重尾分布的均值和方差均为无限大^[16],经常发生的相对较短的直线运动会间歇性地由更偶尔出现的步长较长的运动代替,Lévy 飞行可以确保搜索整个空间,因此布谷鸟能够在搜索空间进行比标准高斯随机过程更有效的探索.文献[20]证明了 CS 算法满足全局收敛条件,可以确保 CS 算法能够得到全局最优解.

2 基于双层框架的联合优化算法

2.1 双层优化框架

PSO 算法具有较好的局部搜索能力和收敛性强的特性,CS 算法由于 Lévy 飞行具有高度的随机性而具有较强的跳出局部最优解的能力和全局搜索能力.为了将这两种算法的优点相结合,本文提出基于 PSO 算法和 CS 算法的双层优化框架.首先,将种群划分为几个小种群,底层由 PSO 算法优化各个小种群,在各自所在的区域进行局部搜索;然后,将每个小种群中最优的粒子送到高层作为 CS 算法的解进行深度优化,得到的最优解返回到每个小种群中.这种双层优化框架能加快算法的收敛性,提高算法的全局搜索能力.图 1 展示了双层优化框架的示意图.

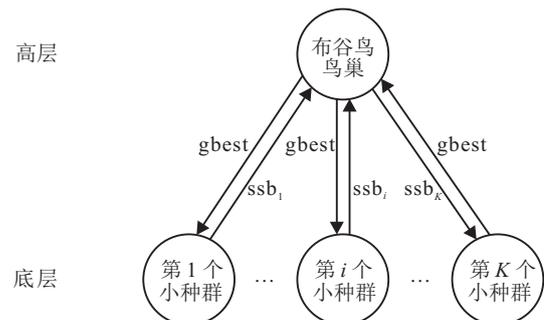


图 1 双层优化框架

在图 1 中: 底层由 K 个小种群构成, ssb_i 为第 i 个小种群的最优解,将 K 个种群的最优解送到高层用 CS 算法进行深度寻优,得到整个种群的最优解 $gbest$,并将 $gbest$ 传入到每个小种群中.

2.2 多种群粒子群算法和布谷鸟双层寻优算法

2.2.1 算法思想

文献[21]表明,在粒子群中社会影响因素相对于自我认知能力是压倒性的,每个粒子应该更多地关注群体中其他个体的信息;文献[8]提出了基于 fully-informed 策略的 FIPSO 算法,主要思想是在社会影响中粒子关注的不仅仅是最优点,在下次飞行中,当前最优点不一定找到最优的搜索区域,这样可能忽略一些重要信息,在社会影响中还要关注粒子邻域中的其他粒子。

聚类中心作为每个种群的核心粒子,在小种群中的作用是举足轻重的,本文的策略是粒子在飞行过程中综合小种群中核心粒子和最优粒子的信息进行寻优.文献[22]也将对标准 PSO 算法中自我学习的部分替换成了对近邻的中心粒子的学习,算法效果更好.底层用改进的 PSO 算法对每个小种群进行优化,每个粒子的位置和速度更新如下:

$$V_{id}^{k+1} = wV_{id}^k + c_1r_1(ssf_i - x_{id}^k) + c_2r_2(gbest - x_{id}^k), \quad (8)$$

$$x_{id}^{k+1} = x_{id}^k + V_{id}^{k+1}. \quad (9)$$

其中: ssf_i 为各小种群的聚类中心, $gbest$ 为整个种群的最优解。

2.2.2 算法步骤

算法底层.

Step 1: 初始化粒子群. 设置粒子群总数 N 、小种群个数 K 、学习因子等常量。

Step 2: 在粒子群中随机选取 K 个粒子作为聚类中心,并计算每个粒子到这 K 个聚类中心的欧氏距离,将每个粒子归为属于与它最近的聚类中心所在的小种群中。

Step 3: 在每个小种群中,计算每个粒子的适应度作为粒子的历史最优解,将小种群的最优适应度作为小种群的当前最优解,将整个粒子群的最优适应度作为整个种群的历史最优解。

Step 4: 根据式(8)和(9)更新每个小种群中各个粒子的速度和位置,并计算每个粒子的适应度。

Step 5: 将粒子的当前适应度值和历史最优值进行比较,如果当前适应度更优,则将当前适应度作为粒子的历史最优值,将粒子所在位置作为粒子的历史最优解.计算小种群的最优值,将小种群最优值对应的粒子位置作为小种群的当前最优解;计算整个种群的最优值,并与整个种群的历史最优值进行比较,如果当前最优值比历史最优值大,则将当前最优值作为整个种群的历史最优值,将最优值对应粒子的位置作为历史最优解。

算法高层.

Step 6: 将各个小种群的最优解送到高层作为布谷鸟的初始位置,将小种群数量作为鸟巢的个数,初始化各种 CS 算法参数。

Step 7: 保留上一代最优鸟巢的位置,对于其他鸟巢,计算随机步长,对鸟巢位置进行更新,并计算每个鸟巢的适应度,如果当前适应度优于历史适应度,则将当前适应度作为历史适应度。

Step 8: 如果随机数大于被主人发现的可能性 P ,则更新鸟巢的位置,如果新的适应度优于之前的解,则将其替换。

Step 9: 找出最后得到的最优解,如果得到的最优解大于在底层中求得的历史最优解,则用布谷鸟优化的最优解作为历史最优解。

Step 10: 判断迭代次数是否超过 R 代,随机选取种群中的 K 个粒子作为新的聚类中心,采用中位数聚类算法确定新的聚类中心。

Step 11: 判断是否达到终止条件,如果没有,则返回 Step 4。

2.2.3 算法流程

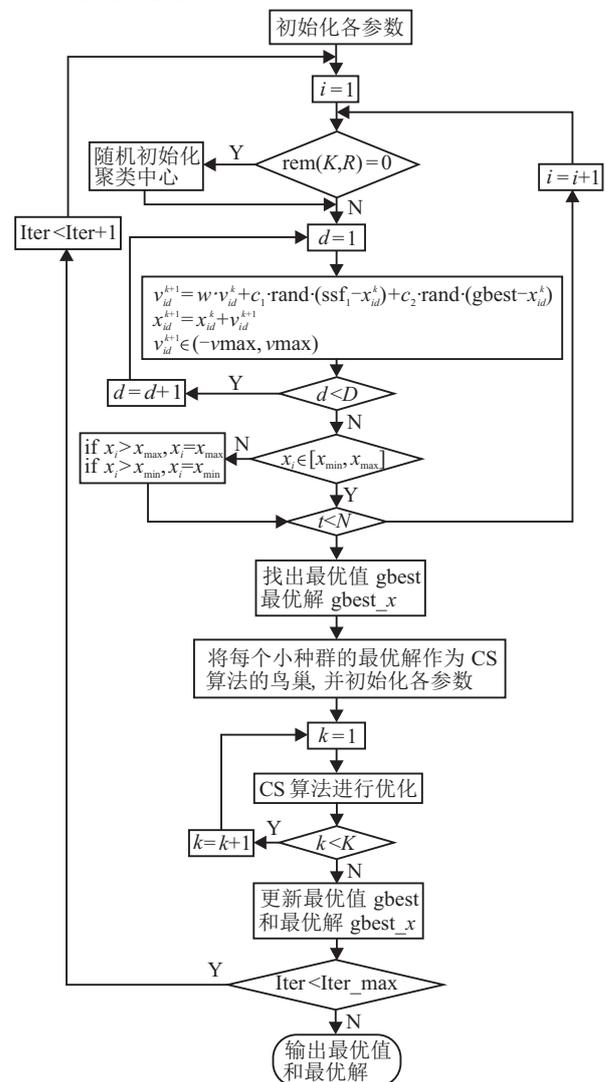


图 2 DMS-PSO-CS 算法

3 实验仿真及分析

3.1 测试函数的选取

为了验证DMS-PSO-CS算法的寻优能力和收敛效率, 选取CEC 2014测试函数集中的2个单峰函数和2个多峰函数作为测试函数并对不同算法进行测试和比较. 为了增加测试难度并检验不同环境下算法的性能, 对4个测试函数进行了平移和旋转处理. 平移处理是将测试函数的全局最优点以向量 \mathbf{o} 进行平移, $\mathbf{o} = [o_1, o_2, \dots, o_D]^D$, 在 $[-80, 80]^D$ 中随机产生; 旋转处理是通过施密特正交化产生一个正交矩阵 \mathbf{M} 并对函数进行旋转.

测试函数中两个单峰函数表示如下:

1) Rotated Discus Function (F_1).

$$F_1 = f_1(\mathbf{M}_1(\mathbf{x} - \mathbf{o}_1)) + F_1^*, F_1^* = 200,$$

$$f_1(\mathbf{x}) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2.$$

2) Rotated Discus Function (F_2).

$$F_2 = f_2(\mathbf{M}_2(\mathbf{x} - \mathbf{o}_2)) + F_2^*, F_2^* = 300,$$

$$f_2(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2.$$

两个多峰函数表示如下:

3) Shifted and Rotated Ackley's Function (F_3).

$$F_3 = f_3(\mathbf{M}_3(\mathbf{x} - \mathbf{o}_3)) + F_3^*, F_3^* = 500,$$

$$f_3(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e.$$

4) Shifted and Rotated Griewank's Function (F_4).

$$F_4 = f_4(\mathbf{M}_4(\mathbf{x} - \mathbf{o}_4)) + F_4^*, F_4^* = 700,$$

$$f_4(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_i \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1.$$

本文对100维的测试函数进行实验, 所有测试函数的可行搜索区域定义在 $[-100, 100]^D$ 内. 每次实验迭代次数为3000次, 粒子群总数设置为30, 最大速度为可行搜索区域的20%. 为了消除算法随机性对实验的影响, 每次实验独立运行20次, 分别记录实验的最小值、最大值、平均值和方差.

3.2 小种群数量和调整周期

在动态多种群PSO算法中, 小种群数量 m 和调整周期 R 的选择很关键. 当小种群的中心粒子陷入局部最优解时, 如果不及时调整和更新小种群, 则小种群难以跳出局部最优; 如果调整周期太短, 则会使计算复杂度增加, 可能使种群还没来得及做精细搜索就要重新分配到新的种群中进行搜索, 使搜索效率变低.

本文选取Rotated Discus Function函数作为检测函数对不同的种群数量和调整周期进行测试, 实验中粒子群总数取30, 每次实验独立运行30次, 每次迭代3000次, 记录每次实验函数值与最优值的差值, 取30次实验的均值作为选择最优的 m 和 R 的依据. Rotated Discus Function在不同参数下的函数值如表1所示.

在表1中: 第1行各数值为调整周期的大小, 第1列各数值为不同的小种群数量. 从表1可以看出, 当小种群数量为5, 调整周期为20时, 函数值达到最优.

表1 Rotated Discus Function在不同参数下的函数值

数量	大小				
	5	10	20	50	100
2	1.09e-12	6.96e-14	5.68e-14	8.67e-14	4.26e-14
3	2.03e-12	5.40e-14	4.55e-14	9.09e-14	4.55e-14
5	2.25e-13	7.25e-14	1.83e-14	5.97e-14	4.83e-14
10	7.77e-13	3.13e-14	7.11e-14	2.54e-14	4.83e-14
15	6.45e-13	4.12e-14	3.41e-14	4.97e-14	5.83e-14

3.3 比较算法说明

为了说明本文算法的特点, 选取以下几种经典的具有代表性的算法进行比较.

PSO-w^[3]: 惯性权重动态调整的粒子群算法;

PSO-cf^[4]: 带收缩因子的粒子群算法;

PSO-local^[7]: 局部粒子群算法, 采取环形结构组建邻域;

FIPSO^[8]: 全联通信 PSO 算法, 采取环形结构组建邻域;

DMS-PSO^[10]: 动态多种群粒子群算法;

VDMP SO^[6]: 基于速度差分变异的粒子群算法;

HSPPSO^[14]: 双层粒子群算法.

双层算法框架下可以衍生出以下几种算法.

HSPPSO-m: 在HSPPSO算法的基础上, 用中位数聚类算法将种群划分为 K 个小种群, 底层小种群用PSO算法寻优之后将每个小种群中最优粒子组建为高层种群, 再用PSO算法对高层种群进行优化, $c_1 = c_2 = 1.4962$, $w = 0.7298$, v_{\max} 取搜索区域长度

的20%;

HSPPSO-CS: 在HSPPSO算法的基础上进行改进, 底层种群由PSO算法优化之后将每个小种群中最优粒子组建为高层种群, 对高层种群用CS算法进行优化, $c_1 = c_2 = 1.4962$, $w = 0.7298$, v_{max} 取搜索区域长度的20%.

3.4 实验结果和分析

图3~图6分别展示了11种算法在2个单峰函数和2个多峰函数上的收敛曲线, 收敛曲线表示为函数值与最优值之差的均值随迭代次数的变化情况.

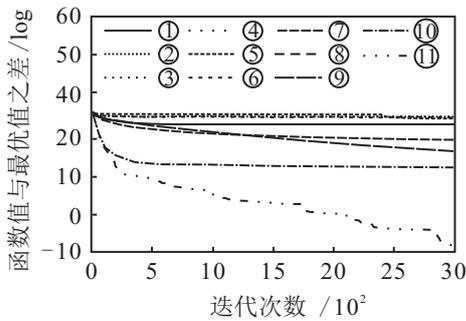


图3 Rotated Bent Cigar Function

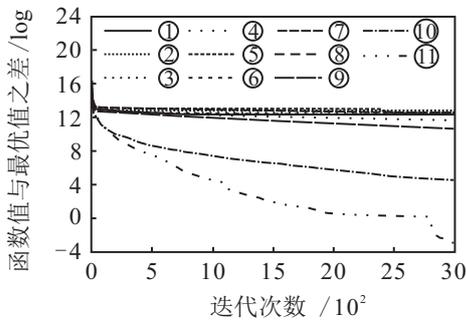


图4 Rotated Discus Function

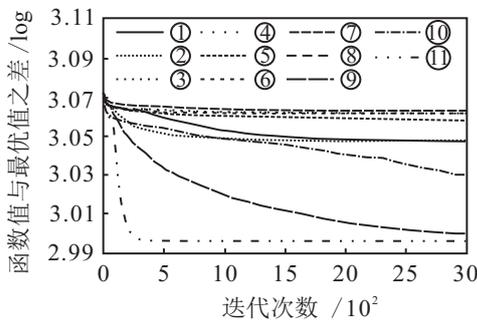


图5 Shifted and Rotated Ackley's Function

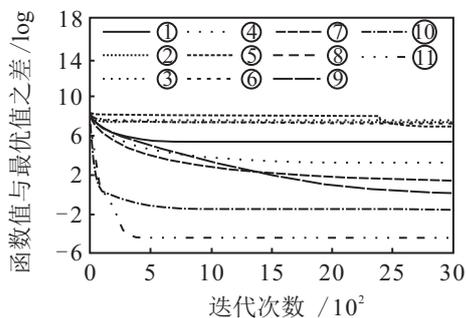


图6 Shifted and Rotated Griewank's Function

在图3~图6中: 曲线①表示PSO-w算法, 曲线②表示PSO-cf算法, 曲线③表示PSO-local算法, 曲线④表示FIPSO算法, 曲线⑤表示DMS-PSO算法, 曲线⑥表示VDMPSO算法, 曲线⑦表示HSPPSO算法, 曲线⑧表示CS算法, 曲线⑨表示HSPPSO-m算法, 曲线⑩表示HSPPSO-CS算法, 曲线⑪表示DMS-PSO-CS算法.

表2展示了11种算法分别在4个测试函数为100维时的测试结果, 测试结果记录了在20次独立运行下的函数值与最优值之差的最小值、最大值、均值和方差, 其中每个测试函数中均值的最好值用黑体显示.

对表2的实验结果作如下分析: PSO-w和PSO-cf为全局搜索算法, 因此在前期表现出较好的收敛性, 但是容易陷入局部最优解, 由于PSO-w算法是一种惯性权值逐渐变小的算法, 在全局搜索进行之后, 对搜索区域展开局部搜索, 因此PSO-w算法的精度比PSO-cf算法好; PSO-local算法、FIPSO算法和DMS-PSO算法为局部搜索算法, 由于算法在保持局部搜索能力的同时减弱了全局搜索能力, 这两种算法在算法初期收敛速度相对于全局搜索算法慢, 由于算法在后期的局部搜索能力较强, FIPSO算法在后期不管是在单峰函数还是多峰函数上都表现较好; VDMPSO算法虽然在算法中引入了差分进化的思想, 但是在单峰函数和多峰函数上表现一般; HSPPSO算法采用双层算法框架, 在多数测试函数中取得了较好的收敛特性, 由于在高层中对算法进行进一步优化, HSPPSO算法的精度也较高.

本文提出的其他双层算法框架下的算法较之前描述的算法表现出更好的结果. HSPPSO-CS算法结合了PSO算法和CS算法的优点, 在算法后期表现出较好的收敛性和精度, 但由于小种群采取固定的分组方法, 种群的多样性小, 精度没有DMS-PSO-CS算法好; HSPPSO-m算法弥补了HSPPSO算法多样性小的不足, 采取中位数聚类算法对小种群进行动态组建, 使得各小种群保持较好的活性和自由度, 特别对于多峰问题, 可以看出, HSPPSO-m算法在算法后期仍然表现出较好的收敛特性, 并达到较好的精度.

可以看出, 对于函数 F_3 , 所有的算法都陷入局部最优, 但DMS-PSO-CS算法仍然比其他算法表现好.

由于DMS-PSO-CS算法采取了一种双层优化框架, 在底层各小种群用PSO算法进行优化之后送入高层CS算法进行深度寻优. 算法中采取中位数聚类算法周期性地对小种群进行动态地组建, 促进了小种群之间的交流, 使得粒子群的多样性变大. 可以看出, 不管是在单峰函数还是多峰函数中, DMS-PSO-CS算法都表现出了良好的收敛性和搜索精度.

表2 实验仿真

算法类型	统计特性	F_1	F_2	F_3	F_4
PSO-w	Best	1.45E+10	1.69e+05	2.10e+01	1.33e+02
	Worst	3.16e+10	3.40e+05	2.11e+01	3.63e+02
	Mean	2.30e+10	2.28e+05	2.11e+01	2.19e+02
	Std.	4.52e+09	3.97e+04	4.77e-02	5.24e+01
PSO-cf	Best	1.17e+11	2.93e+05	2.09e+01	1.30e+03
	Worst	2.06e+11	4.57e+05	2.12e+01	2.02e+03
	Mean	1.71e+11	3.66e+05	2.10e+01	1.61e+03
	Std.	1.94e+10	4.13e+04	7.37e-02	2.07e+02
PSO-local	Best	1.23e+11	1.48e+05	2.13e+01	9.73e+02
	Worst	3.13e+11	3.62e+05	2.14e+01	2.99e+03
	Mean	2.02e+11	2.83e+05	2.14e+01	1.96e+03
	Std.	4.95e+10	5.57e+04	2.20e-02	4.78e+02
FIPSO	Best	2.01e+07	9.96e+03	2.09e+01	1.18e+00
	Worst	1.40e+08	4.10e+02	2.10e+01	1.79e+00
	Mean	2.67e+09	1.14e+05	2.14e+01	2.55e+01
	Std.	5.18e+08	1.72e+04	2.44e-02	5.36e+00
DMS-PSO	Best	1.25e+09	1.41e+04	2.07e+01	2.33e+01
	Worst	8.46e+09	5.44e+04	2.10e+01	6.49e+01
	Mean	4.16e+09	2.78e+04	2.09e+01	4.01e+01
	Std.	2.28e+09	1.02e+04	6.97e-02	1.09e+01
VDMPSO	Best	3.81e+10	7.62e+04	2.12e+01	3.51e+02
	Worst	4.71e+10	9.03e+04	2.12e+01	4.39e+02
	Mean	4.36e+10	1.48e+05	2.07e+01	4.05e+02
	Std.	3.07e+09	5.15e+03	1.55e-02	2.92e+01
HSPPSO	Best	2.56e+08	1.97e+05	2.13e+01	2.78e+00
	Worst	5.03e+08	3.68e+05	2.14e+01	6.98e+00
	Mean	3.76e+08	2.64e+05	2.14e+01	4.21e+00
	Std.	5.88e+07	3.92e+04	3.06e-02	1.05e+00
CS	Best	1.01e+05	8.71e+04	2.12e+01	3.03e-01
	Worst	2.85e+06	1.55e+05	2.14e+01	9.30e-01
	Mean	1.30e+06	1.10e+05	2.13e+01	6.04e-01
	Std.	6.84e+05	1.66e+04	3.57e-02	1.70e-01
HSPPSO-m	Best	1.33e+07	2.62e+04	2.00e+01	1.08e+00
	Worst	3.22e+07	6.55e+04	2.02e+01	1.43e+00
	Mean	1.92e+07	4.18e+04	2.01e+01	1.19e+00
	Std.	5.72e+06	8.75e+03	3.23e-02	8.63e-02
HSPPSO-CS	Best	1.27e+04	1.28e+01	2.00e+01	4.03e-02
	Worst	1.11e+06	3.72e+02	2.11e+01	5.29e-01
	Mean	2.62e+05	9.44e+01	2.07e+01	2.18e-01
	Std.	2.92e+05	9.91e+01	4.25e-01	1.42e-01
DMS-PSO-CS	Best	2.59e-09	4.09e-07	2.00e+01	9.09e-13
	Worst	2.47e-03	3.83e-01	2.00e+01	9.96e-02
	Mean	2.97e-04	5.43e-02	2.00e+01	1.27e-02
	Std.	7.70e-04	1.12e-01	2.57e-06	3.12e-02

4 结 论

本文提出了一种将动态多种群粒子群算法与布谷鸟算法结合的DMS-PSO-CS算法,在双层种群框架下综合了粒子群算法和布谷鸟算法的优点,并且通过中位数聚类算法对底层粒子群进行重组,改善了

随机重组的盲目性,提高了动态多种群粒子群算法的全局搜索能力.通过本文的算法框架下衍生出的几种PSO算法的变形,并经过实验对比可知,所提出的改进PSO算法都有明显改善,其中DMS-PSO-CS算法不管对于单峰函数还是多峰函数都表现出了卓越的性能.

参考文献(References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Perth: IEEE Press, 1995: 1942-1948.
- [2] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of the Sixth Int Symposium on Micromachine and Human Science. Nagoya: IEEE Press, 1995: 39-43.
- [3] Shi Y, Eberhart R. A modified partical swarm optimizer[C]. Proc of IEEE Int Conf on Evolutionary Computation. Anchorage: IEEE Press, 1998: 69-73.
- [4] Clerc M, Kennedy J. The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [5] James Kennedy. Stereotyping: Improving particle swarm performance with cluster analysis[C]. Proc of the 2002 Congress on Evolutionary Computation. La Jolla: IEEE Press, 2000: 1507-1512.
- [6] Jiang S, Wang Q, Jiang J. Particle swarm optimization algorithm based on velocity differential mutation[C]. Proc of the 21st Chinese Control and Decision Conf. Guilin: IEEE Press, 2009: 1860-1865.
- [7] Kennedy J, Mendes R. Population structure and particle swarm performance[C]. Proc of the 4th Congr Evolutionary Computation. Honolulu: IEEE Press, 2002: 1671-1676.
- [8] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8(2): 204-210.
- [9] 王凌, 沈婧楠, 王圣尧, 等. 协同进化算法研究进展[J]. 控制与决策, 2015, 30(2): 193-202.
(Wang L, Shen J N, Wang S Y, et al. Advances in co-evolutionary algorithms[J]. Control and Decision, 2015, 30(2): 193-202.)
- [10] Liang J J, Suganthan P N. Dynamic multi-swarm particle swarm optimizer[C]. Proc of IEEE Int Swarm Intelligence Symposium. Pasadena: IEEE Press, 2005: 124-129.
- [11] Liang J J, Suganthan P N. Dynamic multi-swarm particle swarm optimizer with local search[C]. Proc of Congress on Evolutionary Computation. Edinburgh: IEEE Press, 2005: 522-528.
- [12] Zhao S Z, Suganthan P N, Pan Q K, et al. Dynamic multi-swarm particle swarm optimizer with harmony search[J]. Expert Systems with Applications, 2011, 38(4): 3735-3742.
- [13] 刘衍民, 隋常玲, 赵庆祯. 基于 K -均值聚类的动态多种群粒子群算法及其应用[J]. 控制与决策, 2011, 26(7): 1019-1025.
(Liu Y M, Sui C L, Zhao Q Z. Dynamic multi-swarm particle swarm optimizer based on K -means clustering and its application[J]. Control and Decision, 2011, 26(7): 1019-1025.)
- [14] 吕林, 罗绮, 刘俊勇, 等. 一种基于多种群分层的粒子群优化算法[J]. 四川大学学报: 工程科学版, 2008, 40(5): 171-176.
(Lü L, Luo Q, Liu J Y, et al. A hierarchical structure poly-particle swarm optimization algorithm[J]. J of Sichuan University: Engineering Science Edition, 2008, 40(5): 171-176.)
- [15] Yang X S, Deb S. Cuckoo search via Lévy flights[C]. Proc of the World Congress on Nature & Biologically Inspired Computing. Coimbatore: IEEE Press, 2009: 210-214.
- [16] Yang X S, Deb S. Cuckoo search: Recent advances and applications[J]. Neural Comput & Applic, 2014, 24(1): 169-174
- [17] Erik Cuevas, Adolfo Reyna-Orta. A cuckoo search algorithm for multimodal optimization[J]. Scientific World J, 2014(2014): 497-514.
- [18] Moe Moe Zaw, Ei Ei Mon. Web document clustering by using pso-based cuckoo search clustering algorithm[J]. Recent Advances in Swarm Intelligence and Evolutionary Computation Studies in Computational Intelligence, 2015, 585(1): 263-281.
- [19] Karthikeyan M, Venkatalakshmi K. Energy conscious clustering of Wireless Sensor Network using PSO incorporated cuckoo search[C]. The 3rd Int Conf on Computing Communication Networking Technologies. Coimbatore: IEEE Press, 2012: 1-7.
- [20] Wang F, He X S, Wang Y, et al. Markov model and convergence analysis based on cuckoo search algorithm[J]. Computer Engineering, 2012, 38(11): 180-185.
- [21] Kennedy J, Mendes R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms[C]. IEEE Int Workshop on Soft Computing in Industrial Applications. Binghamton: IEEE Press, 2003: 515-519.
- [22] Frans Van Den Bergh, Andries P Engelbrecht. A cooperative approach to particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 225-239.

(责任编辑: 闫妍)