

具有异构分簇的粒子群优化算法研究

李文锋, 梁晓磊, 张 煜

(武汉理工大学物流工程学院, 湖北武汉 430063)

摘 要: 粒子群优化(Particle Swarm Optimization, PSO)算法在复杂多峰函数可行域空间搜索时极易陷入局部极值点. 研究表明改变种群拓扑结构和调整算法参数有助于改善种群的多样性, 但是目前研究中少有同时考虑种群全局拓扑结构和局部粒子个体能力. 本文提出一种具有异构分簇特性的自适应 PSO 算法. 该算法采用 K -均值聚类算法对种群进行动态分簇, 形成多异构子群, 并采用 Ring 型拓扑结构进行子群间信息流通. 而后采用基于寻解水平评价的粒子自适应参数调整策略进行个体调整. 通过实验分析表明该算法能够提高粒子群优化的种群的多样性、粒子活性、搜索能力和收敛性能, 同时也降低了算法对参数初值的依赖性.

关键词: 粒子群算法; 自适应; 异构; 聚类; 函数优化

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2012)11-2194-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2012.11.009

Research on PSO with Clusters and Heterogeneity

LI Wen-feng, LIANG Xiao-lei, ZHANG Yu

(School of Logistics Engineering, Wuhan University of Technology, Wuhan, Hubei 430063, China)

Abstract: Particle Swarm Optimization (PSO) algorithm easily falls into local optimal solution when solving complex multi-modal function optimization problem. Researches show that dynamic topology and variable parameters can improve the diversity of swarm to improve the situation. However, the effect of topology and parameters is rarely considered simultaneously. In this paper, a new PSO algorithm based on clustering is proposed. It takes K -means clustering method to divide the swarm into different neighborhoods dynamically. These neighborhoods have different number of particles and are heterogeneous clusters. A Ring-structure is applied to exchange information among clusters. Furthermore, a novel discriminating method is proposed to detect the exploring stage of a cluster. Each particle adjusts its parameters automatically according to the exploring stage of its cluster. The results of experiments show that the operations above can improve diversity and energetic of the particles, increase exploring ability and convergence, and reduce the dependence of initial election of parameters.

Key words: particle swarm optimization; adaptability; heterogeneous; clustering; function optimization

1 引言

粒子群优化算法(Particle Swarm Optimization, PSO)是由学者 Kennedy 和 Eberhart^[1,2]受鸟群觅食行为启发提出的一种群智能优化技术,其主要思想是通过个体之间的合作和信息共享实现对目标问题的求解. PSO 算法具有结构简洁、并行强、搜索效率高等特点,因此在多目标优化、调度优化、神经网络训练^[3]、车辆路径问题等领域得到了广泛的应用.但是和其它智能算法类似,PSO 也存在早熟现象.针对这个问题很多学者进行了研究.文献[4]中的测试证明在基本的粒子群算法中,较大的惯性权重 ω 会使种群偏向于探索新区域,导致了搜索

分散;较小的 ω 则会使种群倾向于进行局部搜索,极易出现早熟.最大速度 V_{\max} 则对粒子寻解的步长具有一定影响,过大过小都不适合粒子移动.为此,很多学者针对这点进行了研究,提出了多种改进方式. Shi 和 Eberhart^[5]采用了一种根据算法迭代进程线性调节惯性因子的方法.还有一些采用模糊控制、随机处理等方式的改进措施,都获得了较优的结果. Clerc^[6]在分析迭代过程的基础上提出了添加收敛因子的改进方法,通过参数设置来防止种群的过分扩散.种群的邻域结构对粒子群算法的性能有着重要影响. Mendes 和 Kennedy^[7]对 5 种经典的邻域拓扑结构进行了研究.在其研究中发现, lbest 能保持一个较高的收敛成功率,但是收敛较慢. Kennedy^[8]在

邻域拓扑分析的基础上,引入聚类分析思想,用聚类后的簇的中心粒子和全局的簇中心对邻域中其他粒子进行更新,这种方式对于一些测试函数得到了较优的测试结果.倪庆剑等^[9]在分析几种典型的粒子群拓扑结构后,结合 Kennedy 提出的高斯动态 PSO 算法^[10],提出了一种基于可变多簇结构的动态概率 PSO 算法,通过在粒子群进化的几个不同阶段采用不同的簇结构,来协调算法的搜索能力和寻优性能.王雪飞^[11]提出采用小世界网络具有大的簇系数和小的平均距离特性来构建邻域结构.在高鹰^[12]的基于聚类的粒子群优化算法中,采用每个聚类簇中目前所搜索到的最优解指导其他粒子进行速度位置更新,但是,对于采用何种方式进行聚类、如何保证聚类的成功率以及参数选择对改进方法的影响等问题,文献没有进一步讨论和研究.

本文从拓扑结构和种群自适应性的角度出发,提出一种具有异构分簇特性的自适应粒子群优化算法(Adaptive Particle Swarm Optimization based Clustering, AP-SO-C),采用聚类方式对种群进行动态分割,形成的多簇根据各自搜寻到的解与邻域的最好解进行比较,指导簇内的粒子进行自适应的参数调整.动态的簇划分使得每个簇具有不同的粒子数目,不仅增加了簇的差异性和活性,而且使得粒子可以在簇间迁移.这种簇间粒子交换也增加了簇间的信息交流,从而提高了搜索效率.而通过自适应参数调整,不仅可以消除算法对参数设置的依赖,而且可以赋予每个簇不同的搜索能力,进一步增强簇的差异性和多样性.与此同时,这种参数的自适应性调整,可以进一步提高进化优良粒子簇的局部寻优能力,从而改善算法的搜索精度和效率.

2 算法的分簇和自适应策略

2.1 粒子群分簇策略

与一些社会群体结构类似,单个粒子的行为主要来自对邻域内最好粒子的学习. Kennedy 等人^[11]的研究指出粒子的邻域拓扑结构对粒子的寻解能力有很大影响.借鉴“物以类聚”的思想,本文采用聚类算法对粒子群体进行动态划分,将具有相似特征的粒子聚集作为一个子群.与文献^[8]定期变换子群数量以及一些文献中固定子群规模的方式不同,这种划分是动态的,每次群体更新后都会重新进行划分.这样,不仅每个簇的粒子数不同,形成若干个异构的簇.而且粒子可以在簇间迁移,增强了簇间信息的交换能力.

K-means 算法是一种应用较广的聚类方法,其核心思想是以元素间距离为测度,将均值作为中心,以最小化各元素到所属中心的距离总和为目标.假设选出 K 个粒子作为中心(p_1, p_2, \dots, p_K),将粒子间的距离定义

为 $dis(x_i, x_j) = \sum_{n=1}^D (x_{in} - x_{jn})^T \cdot (x_{in} - x_{jn})$, 则聚类的目标函数为 $F = \sum_{j=1}^M \sum_{k=1}^K dis(x_j, p_k)$. 粒子群的聚类过程如下:

Step1 从种群 M 中随机选取 K 个粒子 p_1, p_2, \dots, p_K , 每个粒子作为一个簇的中心.

Step2 计算其余粒子到各簇中心的距离 $dis(x_j, p_k)$, $j \in (1, 2, \dots, M)$, $k \in (1, 2, \dots, K)$, 根据距离最小原则,将各数据点划分于距离最近的簇.

Step3 对于每个簇,采用该簇内所有粒子位置的均值作为新的簇中心.

Step4 重复 Step2, 直至达到预设目标, 停止迭代, 输出结果.

在粒子群算法中,种群蕴含具有一种聚集的趋势.吕振肃等^[12]在文献^[13]基础上从数学角度进行了推导,证明整个种群最终将收敛于粒子个体极值和种群全局极值区间中的某位置.考虑粒子自身的这种聚集性,本文在进行聚类操作时对聚类目标进行了修正,将聚类操作设计为伴随着整个搜索进程的一种渐进式逐步聚类过程,不需每次操作都达到稳定态.这也继承了粒子群算法的聚集求精的特性.

在粒子聚类成簇以后,对粒子进行适应度选择,选其中最优粒子最为本簇的最优解 **Clusterbest**. 对于各簇之间的联系,在本文中借鉴 *lbest* 版本的经典 PSO 算法所采用 Ring 型结构对各簇进行信息交流.假设将第 i 簇当前自身搜索到的最优解记为 **Clusterbest_i**, 通过比较与其相邻两簇的 **Clusterbest_{i-1}** 和 **Clusterbest_{i+1}** 的优劣,选择最优的作为第 i 簇的 **Neibest_i**.

考虑全局极值和簇极值的影响,本文构建种群个体粒子速度的更新公式,如式(1).而个体粒子位置的更新仍采用基本式,如式(2).

$$v_i^{t+1} = \omega \cdot v_i^t + c_1 \cdot r_1 \cdot (pbest_i^t - x_i^t) + c_2 \cdot r_2 \cdot 0.5 \cdot ((gbest^t - x_i^t) + (Neibest_i^t - x_i^t)) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

2.2 粒子群异构自适应策略

在上述聚类过程完成以后,每个簇中粒子的个数不同,使得簇的结构上有所不同.实际寻解中,由于簇内粒子的差异,不同簇会处于搜索的不同水平,例如离最优粒子的平均距离不同.因此可以依据这种差异对所有的簇进行搜索水平上的等级划分.

定义 1 将第 i 簇标记为 $Cluster_i$, $\|Cluster_i\|$ 表示 $Cluster_i$ 中所包含的粒子个数,记 f_{ij} 表示第 i 个簇中第 j 个粒子的适应度,则第 i 簇的适应度均值 $average_i$

$$= \sum_{j=1}^{\|Cluster_i\|} f_{ij} / \|Cluster_i\| ;$$

记 f_j 表示第 j 个粒子的适应

度,则种群适应度均值 $Average_g = \sum_{i=1}^M f_i/M$, M 为种群规模.

定义 2 将 ω_{\max} 和 ω_{\min} 记为初始化时粒子的最大和最小惯性权重, ω_i^{t+1} 为第 $t+1$ 代时粒子 i 的惯性权重, ω_i^{t+1} 的值将根据粒子的搜索水平在 ω_{\max} , ω_{\min} 和 ω_i^t 基础上进行调整,调整方式见式(3).

定义 3 将 V_f 记为初始化时粒子的最大速度, $V_{\max i}^{t+1}$ 为第 $t+1$ 代内粒子 i 所允许的最大速度, $V_{\max i}^{t+1}$ 的值将根据粒子的搜索水平在 V_f 和 $V_{\max i}^t$ 基础上进行调整,如式(4).

由于每个粒子在搜索过程中表现不同,对于不同层次的粒子,在寻优过程中需要它加强该层次所需要的能力,譬如性能表现劣的粒子,需要它快速摆脱较差的位置,加速开拓其它区域;对于较优的粒子,则需要它加强对所在区域的开发,挖掘更加优良的解.在本算法中将在基于种群分簇的基础上对粒子能力进行调整.对于聚类后的每个簇,由于簇中粒子优劣程度不同,簇的平均适应度均值也表现出一定的差异.利用这个差异可以对簇进行评价,以指导粒子进行 ω 和 V_{\max} 参数调整.根据比较结果可分为两种情况:

以最小化目标函数为例:(1)若 $average_i \geq Average_g$,则该簇处于较劣位置,距离最优解较远,因此应当加大此部分粒子的搜索步长,提高其全局探索能力,避免搜索过缓.调整步长通过增加惯性权重和减少对速度的限制的方式实现;(2)若 $average_i < Average_g$,则该簇处于较优位置,与最优解比较接近,因此减少此部分粒子的搜索步长,提高其局部搜索能力,避免步长过大,错失最优解.调整步长通过减少惯性权重和增加对速度的限制的方式实现.

根据两种不同情况,粒子的 ω 和 V_{\max} 参数采用式(3),(4)进行自适应调整.

$$\omega_i^{t+1} = \begin{cases} \omega_i^t + (\omega_{\max} - \omega_i^t) \cdot average_i / Average_g, & \text{if}(average_i \geq Average_g) \\ \omega_i^t - (\omega_i^t - \omega_{\min}) \cdot average_i / Average_g, & \text{if}(average_i < Average_g) \end{cases} \quad (3)$$

$$V_{\max i}^{t+1} = \begin{cases} V_{\max i}^t \cdot (1 + 1/V_f), & \text{if}(average_i \geq Average_g) \\ V_{\max i}^t \cdot (1 - 1/V_f), & \text{if}(average_i < Average_g) \end{cases} \quad (4)$$

2.3 算法步骤

主要分为两部分:

(1)聚类分簇,即根据种群数量采用 K-means 算法将粒子聚合成为若干个簇(子种群),而簇间采用 Ring 拓扑结构建立联系.

(2)参数的自适应调整,以确定每个簇与之搜索水平相适应的搜索能力.

具体算法步骤如下:

Step1 初始化种群.在初始化区域内,随机产生 m 个粒子位置 x_i 及其飞行的速度 v_i , $i \in (1, 2, \dots, m)$,计算每个粒子的初始适应值.设定初始的粒子 i 最大速度 $V_{\max i}$,最大惯性权重 ω_{\max} ,粒子个体 i 的惯性权重 ω_i .

Step2 随机选取 K 个粒子作为中心,采用 2.1 中聚类策略对种群进行划分,确定子种群 $Cluster_j$, $j \in (1, 2, \dots, K)$, K 为子种群数目.

Step3 根据当前粒子的速度和位置,计算出目前为止,各簇中粒子所搜索到的最优位置 $pbest_i$, $i \in (1, 2, \dots, m)$,将其中最优粒子的值作为当前簇的最优位置 $Clusterbest_j$, $j \in (1, 2, \dots, K)$.

Step4 对于簇 $Cluster_j$,将其最优值与相邻两个簇进行比较,采用其中最优的作为当前簇的 $Neibest_i$.

Step5 计算粒子的适应值,记录种群目前所搜索到的最优值 $gbest$.

Step6 判断种群是否收敛或达到所设精度,如若满足条件则停止迭代,输出结果;否则,采用式(3),(4)对粒子进行参数的自适应调整,重复 Step(2)~(6).

3 实验分析

3.1 实验设计

为了分析和比较基于聚类的 PSO 算法(KPSO)及增加了自适应性的 APSO-C 算法的性能,笔者设计了 3 种实验方案:种群分布度比较、算法性能的统计学分析和算法对参数选择的敏感性测试.在测试中,参考相关文献[6,9]实验选取了 4 个标准测试函数:Sphere、Rastrigin、Griewank 和 Ackley 函数. Sphere 为单峰函数, Griewank、Rastrigin 和 Ackley 函数为多峰函数,这些函数理论极值均为 0,初始范围设置为 $[0, 30]$.采用文献[9]推荐值,将 PSO、KPSO 和 APSO-C 的种群规模统一设置为 $N = 30$,初始最大速度 $V_{\max} = X_{\max} = 30$. PSO 学习因子为 $c_1 = c_2 = 1.5$,惯性权重采用线性递减方式 $\omega_{\max} = 0.95$, $\omega_{\min} = 0.4$,根据文献[4,5]研究,此时 PSO 算法有比较好的表现. KPSO 中 $c_1 = c_2 = 0.5$,惯性权重 $\omega = 0.65$,聚类数 $K = 5$; APSO-C 中,取 $c_1 = c_2 = 0.5$, $\omega_{\max} = 0.65$, $\omega_{\min} = 0.4$, $K = 5$, $V_f = 30$.测试中所有算法均采用 Java 语言实现,在 Eclipse 平台上编译运行.

3.2 实验结果及分析

3.2.1 实验 1 种群分布度对比分析

本实验目的是分析本算法思想对粒子的分布度的影响,这种分布度一定程度上反映了种群的多样性.为了便于粒子空间位置的图像表示,将可行域空间设置为 2 维.对于选定的 PSO、KPSO 和 APSO-C 在相同的初

始数据下进行测试,最大迭代次数设为 100. 在测试中分别记录在代数 $Iteration = 10, 30, 70, 100$ 时所有粒子位置信息, 鉴于结果的类似性以及篇幅的限制, 选取多峰函数 Griewank 测试结果作为主要分析对象.

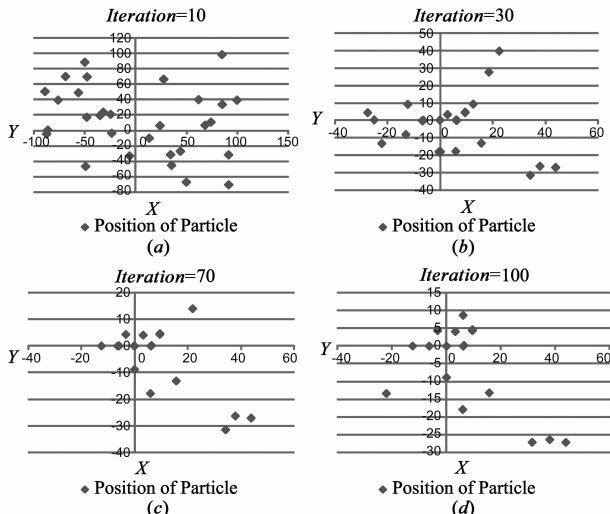


图1 PSO测试Geiewark函数时种群分布度

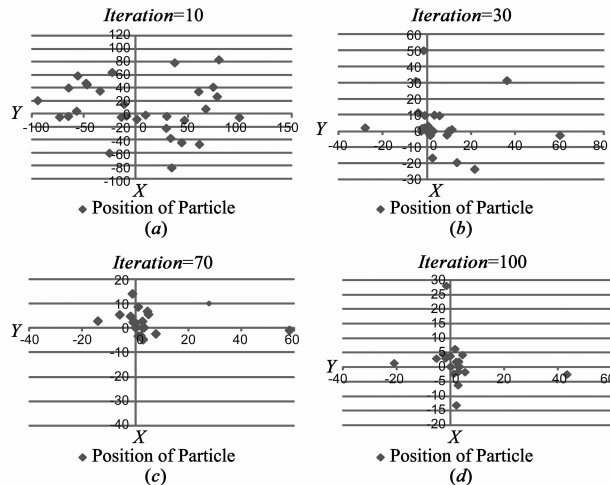


图2 KPSO测试Geiewark函数时种群分布度

Griewank 存在多个局部极值点, 具有很强的欺骗性, 不易优化. 从图 1 中可以看出, PSO 在测试中, 种群始终处于较分散状态, 从 $Iteration = 30$ 到 70 种群分布没有明显变化(图 1(b), (c)). 当迭代末期 $Iteration = 100$ 时, 种群的部分粒子陷入了几个局部极值点区域中, 无法跳出实施进一步的进化, 这也显示了基本 PSO 容易陷入局部极值的缺陷. 对于 KPSO 算法, 与 PSO 相比具有良好的全局收敛性, 迭代末期绝大部分粒子都能够进入全局最优值点的区域进行探索(图 2(c), (d)), 避免了陷入局部极值. 对于 APSO-C 算法, 当 $Iteration = 30$ 和 70 时, 比较图 3(b), (c) 与图 1(b), (c)、图 2(b), (c) 可以看出, APSO-C 中粒子聚集程度明显区别于 PSO 的松散状态, 种群具有全局收敛能力; 与 KPSO

相比, APSO-C 算法收敛较快. 这表明自适应调整策略下的 APSO-C 在多峰函数优化中, 不易陷入局部极值的区域; 收敛速度上也具有一定优势, 能快速收敛到全局最优值. 其余单峰函数和多峰函数的测试结果与上分析类似.

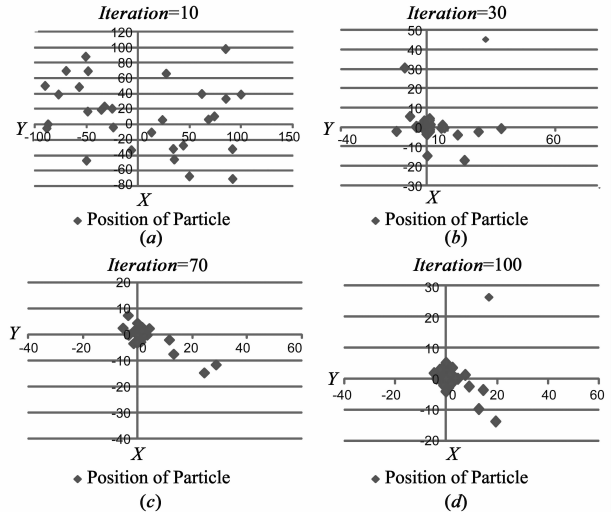


图3 APSO-C测试Geiewark函数时种群分布度

3.2.2 实验 2 算法性能的统计学分析

本实验的目的是采用统计方法比较 APSO-C、PSO 和 KPSO 的性能. 实验中针对每个函数每种算法独立运行 20 次, 每次固定运行 100 代, 统计参数包括: 最小值 (Min)、最大值 (Max)、平均最优值 (Mean Best)、标准差 (St. Dev). 实验结果见表 1.

表 1 函数测试数据

Fun	Algorithm	Min	Max	Mean Best	St. Dev
Sphere	PSO	1.59E-15	8.54E-13	1.49E-13	1.87E-13
	KPSO	8.35E-18	1.79E-15	3.62E-16	4.25E-16
	APSO-C	3.19E-31	4.41E-29	7.16E-30	1.07E-29
Rastrigin	PSO	1.77E-15	3.19E-14	9.15E-15	8.35E-15
	KPSO	0	1.42E-14	5.59E-15	4.62E-15
	APSO-C	0	0	0	0
Griewank	PSO	2.22E-16	2.51E-14	7.549E-15	6.87E-15
	KPSO	0	0	0	0
	APSO-C	0	0	0	0
Ackley	PSO	1.34E-08	3.24E-07	1.01E-07	7.51E-08
	KPSO	1.92E-09	1.19E-08	5.87E-09	2.78E-09
	APSO-C	8.88E-16	2.66E-15	7.11E-16	1.81E-15

单峰函数 Sphere 越接近极值其变化越为平缓, 因此常用于测试算法的寻优能力. 从表 1 可以看出, 在相同迭代次数的设定下, KPSO 和 APSO-C 解的寻优精度明显高于 PSO, 这说明采用聚类的处理方式, 有利于保持种群多样性和粒子活性, 促进种群持续进化. 而采用自适应处理的 APSO-C 算法其寻优的精度更高于 KPSO, 说明自适应处理增强了粒子能力, 通过综合种群最优值和簇最优值信息(式(3)), 粒子选择了更加适合自身

的方向进行寻解.

对于多峰函数,APSO-C 的性能强于 KPSO 和 PSO,在 Griewank 测试中,KPSO 和 APSO-C 运行的 20 次都收敛到了最优解,而且 APSO-C 显出了较强的优势.这说明动态聚类的处理保证了整个种群的多样性和粒子的活性,从而提高了种群跳出局部最优的能力.而粒子根据自身寻解所处的水平进行自适应参数调整,进一步提高了求解能力.

3.2.3 实验 3 APSO-C 算法对参数的敏感性分析

本实验的目的是测试 ω_{\max} 对 APSO-C 算法性能的影响.实验中将 0.05 作为间隔, ω_{\max} 取 $[0.55, 0.95]$ 之间的数值,算法最大运行次数设置为 200.由于在 Geiewark、Rastrigin 和 Ackley 测试中都收敛到了极小值,所以实验中抽取 $Iteration = 50$ 时的数据进行比较.每个函数测试分别运行 20 次.试验结果具体见表 2、3、4 和 5.在这里选取单峰函数 Sphere 和多峰函数 Geiewark 为例进行分析.

表 2 Sphere 函数 ω_{\max} 敏感性测试 ($Iteration = 200$)

ω_{\max} Type	0.55	0.6	0.65	0.7	0.75
Max	1.26E-59	1.74E-59	3.37E-54	2.07E-43	2.34E-55
Min	1.81E-68	5.70E-69	1.92E-68	2.19E-68	2.38E-68
Mean Best	6.33E-61	8.72E-61	1.69E-55	1.04E-44	1.17E-56
St. Dev	2.76E-60	3.79E-60	7.35E-55	4.52E-44	5.10E-56
Avg.	200	200	200	200	200

ω_{\max} Type	0.8	0.85	0.9	0.95
Max	1.45E-49	7.26E-46	5.35E-52	1.39E-48
Min	1.98E-68	3.96E-68	3.16E-69	4.22E-69
Mean Best	1.08E-50	3.63E-47	2.68E-53	6.97E-50
St. Dev	3.45E-50	1.58E-46	1.16E-52	3.03E-49
Avg.	200	200	200	200

表 3 Geiewark 函数 ω_{\max} 敏感性测试 ($Iteration = 50$)

ω_{\max} Type	0.55	0.6	0.65	0.7	0.75
Max	1.19E-09	5.82E-12	1.15E-09	4.21E-08	1.68E-08
Min	9.66E-15	1.23E-14	5.22E-15	1.38E-14	5.11E-15
Mean Best	6.78E-11	8.68E-13	1.26E-10	2.57E-09	1.32E-09
St. Dev	2.58E-10	1.54E-12	2.91E-10	9.27E-09	3.85E-09
Avg.	60.45	59.55	61.65	60.5	62.7

ω_{\max} Type	0.8	0.85	0.9	0.95
Max	2.98E-09	2.04E-08	4.75E-12	4.56E-09
Min	8.99E-15	1.55E-14	2.89E-15	6.00E-15
Mean Best	2.22E-10	1.40E-09	6.17E-13	4.35E-10
St. Dev	6.84E-10	4.54E-09	1.31E-12	1.24E-09
Avg.	60.85	61.5	58	62.15

对于单峰函数 Sphere(表 2),在最大迭代次数内,对

于不同 ω_{\max} 取值,种群所寻的最差解 Max 介于 10^{-43} 与 10^{-59} 数量级之间,其精度完全符合预设要求;最优解 Min 更达到了 10^{-68} 数量级.从平均解 Mean Best 以及标准差 St. Dev 的统计值也可以看出,算法所寻得的解处于一个较优的水平,波动较小.对于多峰函数(表 3),其求解精度也都达到了测试的要求,分析各统计值,可以得出与单峰函数类似的结论.其余函数分析结论一致.

从分析中可以看出设置不同的参数对算法的性能没有特别明显的影响,改进的 APSO-C 算法在参数 ω_{\max} 的设置上具有良好的健壮性,对初始数值的选择不敏感.

表 4 Rastrigin 函数 ω_{\max} 敏感性测试 ($Iteration = 50$)

ω_{\max} Type	0.55	0.6	0.65	0.7	0.75
Max	4.25E-10	1.27E-07	1.80E-09	9.34E-07	1.58E-09
Min	1.23E-11	8.60E-12	1.70E-11	1.76E-11	1.19E-11
Mean Best	1.21E-10	6.79E-09	3.16E-10	4.78E-08	3.49E-10
St. Dev	8.96E-11	2.76E-08	4.47E-10	2.03E-07	4.28E-10
Avg.	63.25	63.9	63.65	63.85	64.1

ω_{\max} Type	0.8	0.85	0.9	0.95
Max	2.734E-06	3.27E-06	6.12E-06	2.52E-08
Min	1.02E-11	3.39E-11	1.44E-11	5.35E-12
Mean Best	1.37E-07	1.64E-07	3.14E-07	1.40E-09
St. Dev	5.97E-07	7.13E-07	1.35E-06	5.46E-09
Avg.	63.2	64.64	65.1	63.65

表 5 Ackley 函数 ω_{\max} 敏感性测试 ($Iteration = 50$)

ω_{\max} Type	0.55	0.6	0.65	0.7	0.75
Max	5.68E-05	2.63E-06	1.88E-06	5.61E-06	8.49E-05
Min	2.85E-07	2.54E-07	2.77E-07	1.71E-07	1.68E-07
Mean Best	6.26E-06	8.81E-07	7.46E-07	1.25E-06	4.95E-06
St. Dev	1.60E-05	5.73E-07	3.57E-07	1.35E-06	1.84E-05
Avg.	115.2	119.15	111.55	111.8	111.25

ω_{\max} Type	0.8	0.85	0.9	0.95
Max	0.000206	0.000342	3.9E-05	1.58E-06
Min	2.44E-07	2.66E-07	2.94E-07	1.77E-07
Mean Best	3.35E-05	1.97E-05	2.77E-06	7.79E-07
St. Dev	6.79E-05	7.43E-05	8.33E-06	3.99E-07
Avg.	114.35	120.55	115.45	111.25

4 结束语

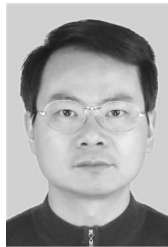
本文针对基本 PSO 算法易陷入早熟以及对参数设置敏感等问题,深入研究了算法的种群拓扑结构和分簇思想.论文提出了一种兼顾种群拓扑和个体能力的新型粒子群优化算法.通过算法的收敛性分析和种群分布度、统计学分析及对参数敏感性测试等多方面实

验表明:本文提出的 APSO-C 很好地解决了种群多样性、收敛速度以及参数敏感性等方面的问题,达到了预期效果.

参考文献

- [1] Kennedy J, Eberhart RC. Particle swarm optimization [A]. Proc. of the IEEE International conference on Neural Networks [C]. USA: IEEE Press, 1995. 1942 – 1948.
- [2] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory [A]. Proc. of the 6th Int'l Symp on Micro Machine and Human Science [C]. Nagoya, Japan: IEEE Press, 1995. 39 – 43.
- [3] 高海兵, 高亮, 周驰, 等. 基于粒子群优化的神经网络训练算法研究 [J]. 电子学报. 2004, 32(9): 1572 – 1574.
GAO Hai-bing, GAO Liang, ZHOU Chi. Particle Swarm Optimization Based Algorithm for Neural Network Learning [J]. Acta Electronica Sinica. 2004, 32(9): 1572 – 1574. (in Chinese)
- [4] Shi Y, Eberhart RC. A modified particle swarm optimizer [A]. Proc. of the IEEE Congress on Evolutionary Computation [C]. Anchorage: IEEE Press, 1998. 69 – 73.
- [5] Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization [A]. Proc. of the IEEE Congress on Evolutionary Computation 1999 [C]. Washington, DC: IEEE Press, 1999. 1955 – 1962.
- [6] Mendes R, Kennedy J, Neves J [A]. Watch why neighbor or how the swarm can learn from its environment [C]. Proc. of Swarm Intelligence Symposium. Indianapolis: IEEE Press, 2003. 88 – 94.
- [7] Kennedy J. Stereotyping: improving particle swarm performance with cluster analysis [A]. Proc. of the 2000 Congress on Evolutionary Computation [C]. La Jolla: 2000. 1507 – 1512.
- [8] 倪庆剑, 张志敏, 王蓁蓁, 邢汉承. 一种基于可变多簇结构的动态概率粒子群优化算法 [J]. 软件学报, 2009, 20(2): 339 – 349.
NI Qing-jian, ZHANG Zhi-min, WANG Zhen-zhen, XING Han-cheng. Dynamic probabilistic particle swarm optimization based on varying multi-cluster structure [J]. Journal of Software, 2009, 20(2): 339 – 349. (in Chinese)
- [9] Kennedy J. Dynamic-probabilistic particle swarms [A]. In: Proc. of the Conf. on Genetic and Evolutionary Computation [C]. Washington: ACM Press, 2005. 201 – 207.
- [10] 王雪飞. 一种具有动态拓扑结构的粒子群算法研究 [D]. 重庆: 西南大学博士学位论文. 2008: 205-207.
WANG Xue-fei. Research on dynamic topology of particle swarm algorithms [D]. Chongqing: Southwest University. 2008, 205 – 207. (in Chinese)
- [11] 高鹰, 谢胜利. 基于聚类的多子群粒子群优化算法 [J]. 计算机应用研究, 2006, 4: 40 – 41.
GAO Ying, XIE Sheng-li. A multi sub-population particle swarm optimizer based on clustering [J]. Application Research of Computers, 2006, 4: 40 – 41. (in Chinese)
- [12] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法 [J]. 电子学报. 2004, 32(2): 416 – 420.
LV Zhen-su, HOU Zhi-rong. Particle swarm optimization with adaptive mutation [J]. Acta Electronica Sinica. 2004, 32(3): 416 – 420. (in Chinese)
- [13] F van den Bergh. An analysis of particle swarm optimizers [D]. South Africa: Department of Computer Science, University of Pretoria, 2002. 81 – 83.

作者简介



李文锋 男, 1966 年出生, 湖南临湘人, 教授、博士生导师, IEEE 高级会员, IEEE SMC CSCWD 的 TC 成员, 现任武汉理工大学“物流与机器人技术实验室”主任. 主要研究方向: 环境感知与协同控制、物流自动化、复杂系统建模与仿真等.

E-mail: liwf@whut.edu.cn



梁晓磊 男, 1985 年出生, 山西省长治市人, 武汉理工大学物流工程学院博士研究生. 研究方向为: 群智能优化、物流系统仿真与建模、复杂系统分析.

E-mail: liangxiaolei.cn@gmail.com

张煜 男, 1974 年生, 天津人, 副教授, 武汉理工大学物流工程学院讲师, 主要研究方向: 系统仿真、优化; 虚拟现实; 流体传动及控制.