

基于拓扑结构与粒子变异改进的粒子群优化算法

许胜才[†], 蔡 军, 程 昀, 王海湘

(贺州学院 建筑工程学院, 广西 贺州 542899)

摘 要: 为使粒子群优化算法(PSO)优化过程的多样性与收敛性得到合理解决,以提高算法优化性能,基于种群拓扑结构与粒子变异提出两种粒子群改进算法 RSMPSO 和 RVMPSO. 改进算法将具有信息定向流动的闭环拓扑结构与星型拓扑结构或四边形拓扑结构相结合,促使粒子在前期寻优过程中具有较高的多样性,保证搜索的广度,而在后期满足粒子群的整体收敛性,保证寻优的精度. 同时,将布谷鸟搜索算法(CS)中的偏好随机游走变异策略引入改进算法中,增强粒子跳出局部最优的能力. 对标准测试函数的仿真实验表明,所改进的 PSO 算法与其他 6 个对比算法相比不仅操作简单,优化精度高,而且在算法收敛性及稳健性方面都有着更出色的表现.

关键词: 粒子群算法; 布谷鸟搜索; 拓扑结构; 变异

中图分类号: TP273

文献标志码: A

Modified particle swarm optimization algorithms based on topology and particle mutation

XU Sheng-cai[†], CAI Jun, CHENG Yun, WANG Hai-xiang

(College of Architecture Engineering, Hezhou University, Hezhou 542899, China)

Abstract: Two kinds of modified particle swarm optimization (PSO) algorithms called particle swarm optimization with ring-star topology and particle mutation (RSMPSO) and particle swarm optimization with ring-von neumann topology and particle mutation (RVMPSO) are presented based on topology and mutation of particle swarm, which enable the diversity and convergence of the particle swarm to be preserved in optimizing procedure to improve the optimization performance of the PSO algorithm. In the modified PSO algorithms, the ring topology with information directionally flowing is combined with star topology or square topology. This strategy enables the particle to have more diversity during the early-stage optimizing to guarantee the search extent. And in the later period, this strategy enables the particle swarm to be converged to make sure the accuracy of optimizing. Meanwhile, a mutation strategy with random walk in a biased way in cuckoo search (CS) algorithm is introduced to the modified PSO algorithm to improve the ability of particle of escaping the local optimum. The simulation experiment on the standard test function shows that, compared with other six algorithms, the modified PSO algorithm possesses the features of easy operation and high optimizing accuracy, and a better performance in diversity and robustness.

Keywords: particle swarm optimization; cuckoo search; topology; mutation

0 引 言

粒子群优化算法(PSO)^[1]在寻优过程中,粒子通过个体间相互协作和信息共享,以自身和种群的历史最优位置对粒子当前的运动方向和运动速度加以影响,引导整个群体向可能解的方向移动,粒子在运动过程中逐渐增加发现较优解的可能性. PSO 算法具有原理简单、参数较少、能解决全局优化问题等优点,现已广泛应用于数据处理、系统辨识、模式识别、信

号处理、机器人控制、神经网络及工程结构优化等领域^[2]. 然而,标准的 PSO 算法在实践过程中也暴露出一些缺点,如算法早熟、易陷入局部最优、对多模态优化问题收敛性差等. 因此,针对 PSO 算法改进的研究不断涌现,主要包括 3 个方面:参数动态调整的改进,种群拓扑结构的改进以及融合其他算法的改进.

PSO 算法中参数的改进主要针对算法速度更新公式中的惯性权重和学习因子进行的. 惯性权重最

收稿日期: 2017-08-11; 修回日期: 2018-02-03.

基金项目: 贺州学院博士科研基金项目(HZUBS201610).

责任编辑: 赵珺.

作者简介: 许胜才(1985—),男,讲师,博士,从事结构优化设计等研究; 蔡军(1988—),男,讲师,硕士,从事防灾减灾及防护工程等研究.

[†]通讯作者. E-mail: xushctzr@163.com.

早由 Shi 等^[3]引入,采用线性下降的调整策略,后来还有学者提出了非线性调整策略,包括对数下降策略^[4]、混沌调整策略^[5]、随机调整策略^[6]、自适应调整策略^[7]等.学习因子对粒子的探索行为和收敛行为具有重要影响,在标准 PSO 算法中通常取为常数.为了增加粒子群的寻优能力,一些学者对学习因子提出诸如自适应的调整策略^[8]、基于平均速度的调整策略^[9]等. PSO 算法中粒子合作方式即种群拓扑结构对 PSO 算法的寻优性能有着重要的影响. PSO 算法开发者 Kennedy 等^[10]最早从粒子群的拓扑结构对 PSO 算法进行了改进,提出了局部版的 PSO 算法,并深入研究了种群拓扑结构对算法优化性能的影响,结果表明局部版的 PSO 算法相比全局版的 PSO 算法有较大的改进.此外,许多学者基于静态的拓扑结构理论,提出了动态拓扑结构以增加种群多样性,用于提升粒子群的探索寻优能力^[11-14].若将 PSO 算法与其他进化算法混合,引入其他算法的优点,取长补短,往往能产生一些比较好的效果^[15]. Sun 等^[16]以标准 PSO 算法的认知学习和社会学习优化机制为基础,根据金属导体中自由电子的定向漂移运动和无规则随机热运动表现,提出了一种具有较强全局搜索能力的随机漂移粒子群算法(RDPSO).李蓉等^[17]提出了一种融合梯度搜索算法和遗传算法中繁殖操作技巧的改进 PSO 算法,结果显示该算法具有更好的收敛速度和收敛精度.高云龙等^[18]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,提出了 DMS-PSO-CS 算法,实验结果显示所提出的算法能够显著提高全局搜索能力和算法效率.还有一些学者在 PSO 算法中引入了粒子的变异机制以增加种群多样性,从而避免算法早熟收敛^[19-20].

本文从粒子群的拓扑结构改进出发,将具有信息定向流动的闭环拓扑结构与全局版的星型拓扑结构或局部版的四边形拓扑结构相结合,促使粒子在前期寻优过程中具有较高的多样性,保证搜索的广度;而在后期满足群体的整体收敛性,保证寻优的精度.同时,将 CS 算法中的偏好随机游走变异策略引入改进的 PSO 算法中,增强粒子跳出局部最优的能力.仿真实验表明,改进的 PSO 算法不仅操作简单,且在算法优化精度、收敛性以及稳健性等方面都有着更出色的表现.

1 标准粒子群优化算法

1.1 算法基本原理

标准 PSO 算法的数学描述为:在 D 维目标搜索空间中, n 个粒子组成一个群体,每个粒子的状态由

位置向量和速度向量描述,位置向量 $\mathbf{X}_i = (X_{i1}, X_{i2}, \dots, X_{id})$,速度向量 $\mathbf{V}_i = (V_{i1}, V_{i2}, \dots, V_{id})$.粒子在搜索最优解的飞行过程中记录两个向量,一个是粒子个体最优位置 $\mathbf{pbest}_i = (\text{pbest}_{i1}, \text{pbest}_{i2}, \dots, \text{pbest}_{id})$;另一个是种群最优位置 $\mathbf{gbest} = (\text{gbest}_1, \text{gbest}_2, \dots, \text{gbest}_d)$.这两个极值在粒子运动过程中会根据搜索到的更优位置而不断更新.算法开始运行时,首先在搜索空间中随机初始化粒子的两个状态向量,然后采用如下公式更新粒子状态:

$$V_{id}(t+1) = wV_{id}(t) + c_1 \text{rand}1_{id}(\text{pbest}_{id}(t) - X_{id}(t)) + c_2 \text{rand}2_{id}(\text{gbest}_d(t) - X_{id}(t)), \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1). \quad (2)$$

其中: w 为惯性权重,用于平衡算法的全局搜索能力和局部搜索能力; c_1 和 c_2 为学习因子,通常取 $c_1 = c_2 = 2$; $\text{rand}1$ 和 $\text{rand}2$ 是在 $[0, 1]$ 之间服从均匀分布的随机数.粒子搜索的范围一般要限制在 $[X_{\min}, X_{\max}]$ 以内,若粒子位置在更新过程中超出了限制范围,则将超出边界的粒子位置重新初始化.过大的粒子速度可能会造成数值发散,因此要对粒子的最大速度进行限制,一般取 $V_{\max} = -V_{\min} = (X_{\max} - X_{\min})/2$;若粒子的速度超出了限制范围,则重新初始化.

粒子的速度更新公式(1)分为3部分,第1部分为粒子的惯性运动部分,反映粒子对当前自身运动状态的信任;第2部分为“认知”项,反映粒子参考之前所经历过的最优位置,以此指导下一步行为决策;第3部分为“社会”项,反映粒子间的信息共享与相互合作.按照式(1)和(2)所表达的算法称为全局版的 PSO 算法.随后, Kennedy 等^[10]基于粒子群拓扑结构改进提出了局部版的 PSO 算法,常采用的5种拓扑结构如图1所示.在局部版中,粒子在“社会”项改为追随拓扑结构中粒子的邻域最优 \mathbf{lbest}_i .局部版的 PSO 算法粒子速度更新公式为

$$V_{id}(t+1) = wV_{id}(t) + c_1 \text{rand}1_{id}(\text{pbest}_{id}(t) - X_{id}(t)) + c_2 \text{rand}2_{id}(\text{lbest}_{id}(t) - X_{id}(t)). \quad (3)$$

由式(3)可见,局部版的 PSO 与全局版的 PSO 差别仅在于“社会”项,这也决定了它们在收敛速度和跳出局部最优的能力上有所差异.全局版的 PSO 算法通常收敛速度快,但更易陷入局部极值.局部版的 PSO 扩大了粒子的探索区域,增加了找到最优解的可能性,但收敛速度较慢.

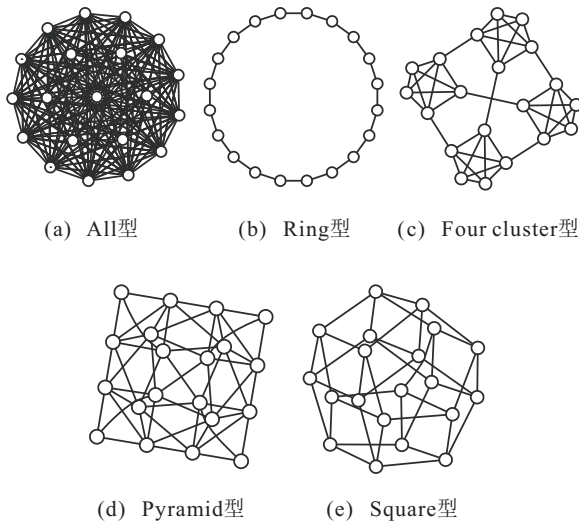


图1 PSO算法种群拓扑结构

1.2 算法存在问题

标准PSO算法存在的问题主要体现在寻优的精度和收敛性上. 在全局版的PSO算法中, 粒子趋向于追随当前种群最优粒子飞行, 如果当前种群最优粒子不是离全局最优解最近的粒子, 则有可能会失去对潜在最优解的搜索, 最终陷入局部最优无法逃离. 虽然局部版的PSO算法增加了粒子追寻最优解的多样性, 在一定程度上改善了寻优精度, 但实验表明, 某一种拓扑结构只针对某一类优化函数具有较好的优化效果, 不具有普适性. 而在收敛性方面, 由于标准PSO算法粒子速度更新公式中, “认知”项的设置使粒子向种群最优粒子飞行的过程中也同时向自身个体最优位置探索, 如果当前迭代步的粒子位置不是个体最优位置, 则往往引起粒子在搜索最优解的路上来回游荡^[12], 不仅降低了对最优位置的发掘能力, 同时也降低了算法的整体收敛性; 所以标准的PSO算法为了使寻优过程得以收敛, 往往要采用约束粒子飞行速度或对速度更新公式采用压缩因子的方式实现^[21].

针对标准PSO算法易陷入局部最优和整体收敛性差的问题, 已有大量的研究从参数的调整、优化种群拓扑结构以及融合其他算法3个方面对PSO算法加以改进. 参数的调整策略对PSO算法优化性能的提升作用往往依赖问题特性, 很大程度上还取决于算法使用者的经验, 所以这种改进方式不具有普遍适用性. 种群拓扑结构的改进是从粒子间合作学习策略方面所作的改进, 但过于简单的拓扑结构不利于搜寻全局最优点, 过于复杂的拓扑结构又不利于推广应用. 混合版的PSO算法吸收了其他方法的优点, 在一定程度上提升了优化性能, 但这些混合策略均引入了一些新的随机变异措施和参数. 新引入的改进措施增加了算法的复杂性, 甚至新引入的参数若设置不合理, 性能反而不如改进前的PSO算法, 所以应用范围

也受到了限制.

PSO算法理想的改进措施应是在粒子寻优过程中, 在寻优前期应有较高的多样性, 保证搜索的广度; 而在后期应有整体收敛措施, 保证寻优的精度.

2 粒子群优化算法改进策略

2.1 拓扑结构改进策略

标准的PSO算法速度更新公式中的“认知”项是使粒子过早失去多样性从而使算法陷入局部最优的重要因素. 因为在粒子搜索过程早期, 如果当前步粒子的位置就是个体最优位置, 则“认知”项为0, 相当于粒子的速度更新公式中只有“社会”项, 众多粒子一致地只向种群最优位置探索, 必然会使粒子陷入种群当前的局部最优而无法跳出. 事实上, 在未知全局最优位置时, 有理由认为全局最优位置可能位于任何一个粒子个体最优位置的附近, 因此, 在每一个粒子的个体最优位置附近执行搜索将会大大提高找到最优解的概率. 据此分析, 在PSO算法速度更新公式的“认知”项中, 应将粒子的搜索方向从自身个体最优位置转变为其他粒子的个体最优位置. 为保证这种搜索的收敛性, 将粒子按编号顺序串联成闭环结构, 并与星形拓扑结构联合建立起新的PSO算法拓扑结构, 本文将这种新的拓扑结构命名为环形-星形混合拓扑结构, 简称RS拓扑结构. 其速度更新公式为

$$V_{id}(t+1) = wV_{id}(t) + c_1 \text{rand}1_{id}(\text{pbest}_{i+1,d}(t) - X_{id}(t)) + c_2 \text{rand}2_{id}(\text{gbest}_d(t) - X_{id}(t)), \quad (4)$$

对应的种群拓扑结构如图2所示.

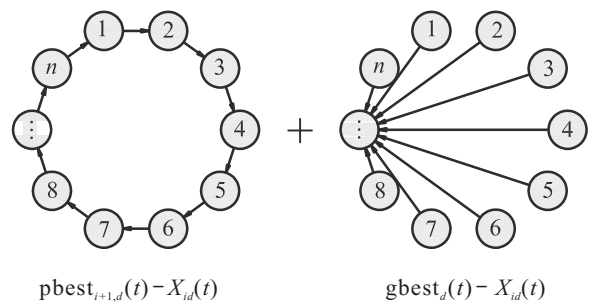


图2 环形-星形混合拓扑结构

按照RS拓扑结构的建立方法, 也可建立起新的局部版PSO算法. 将闭环结构与四边形结构(也称von neumann结构)进行组合, 则可建立起环形-四边形混合拓扑结构, 简称RV拓扑结构. 其速度更新公式为

$$V_{id}(t+1) = wV_{id}(t) + c_1 \text{rand}1_{id}(\text{pbest}_{i+1,d}(t) - X_{id}(t)) + c_2 \text{rand}2_{id}(\text{lbest}_{id}(t) - X_{id}(t)), \quad (5)$$

对应的种群拓扑结构见图3.

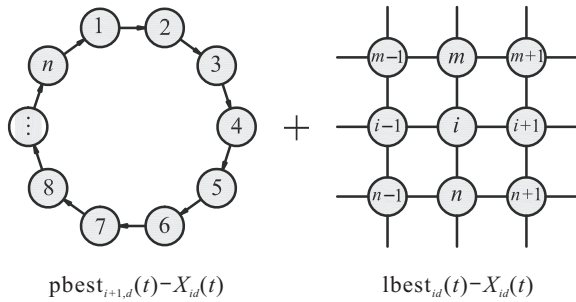


图3 环形-四边形混合拓扑结构

在RS拓扑结构中,粒子以序号排列形成环状结构,每个粒子*i*追随前面粒子*i+1*的个体最优位置和种群最优位置,与标准的PSO算法相比,在“认知”项有所不同.从该拓扑结构可看出,在每一迭代步,每个粒子的个体最优位置都有被其他粒子搜索的机会;同时,种群最优位置的周围,所有其他的粒子都对其进行搜索.粒子的这种搜索方式在解空间中形成错综复杂的搜索网,且覆盖面广,极大提高了找到最优解的概率.可见RS拓扑结构在寻优前期具有较高的多样性,能保证搜索的广度.

已有研究表明^[22],如果迭代时间足够长,标准PSO算法的粒子位置最终收敛结果为

$$\lim_{t \rightarrow +\infty} x(t) = P^* = \frac{c_1}{c_1 + c_2} \text{pbest} + \frac{c_2}{c_1 + c_2} \text{gbest}. \quad (6)$$

假设在寻优达到一定时间后,粒子*n*的个体最优位置为种群最优解*P**,则在RS拓扑结构中,粒子*n-1*所追随的粒子个体最优位置和种群最优位置为同一个对象,则有

$$\lim_{t \rightarrow +\infty} x_{n-1}(t) = \frac{c_1}{c_1 + c_2} \text{gbest} + \frac{c_2}{c_1 + c_2} \text{gbest} = P^*. \quad (7)$$

在RS拓扑结构中,粒子追随其他粒子的个体最优位置时按粒子编号串成闭环结构,搜索信息是按粒子编号定向流动,因此各粒子的收敛结果为

$$\lim_{t \rightarrow +\infty} x_{n-2}(t) = \lim_{t \rightarrow +\infty} x_{n-1}(t) = P^*, \quad (8)$$

$$\lim_{t \rightarrow +\infty} x_{n-3}(t) = \lim_{t \rightarrow +\infty} x_{n-2}(t) = P^*, \quad (9)$$

⋮

$$\lim_{t \rightarrow +\infty} x_1(t) = \lim_{t \rightarrow +\infty} x_2(t) = P^*, \quad (10)$$

$$\lim_{t \rightarrow +\infty} x_n(t) = \lim_{t \rightarrow +\infty} x_1(t) = P^*. \quad (11)$$

从以上分析可以证明,RS拓扑结构在后期具有整体收敛趋势,从而保证了寻优的精度.

在RV拓扑结构中,每个粒子*i*追随前面粒子*i+1*的个体最优位置和邻域最优位置,相比RS拓扑结构,

粒子在解空间中形成的搜索网覆盖面更广,可进一步提高找到最优解的概率.RV拓扑结构的粒子整体收敛趋势与RS拓扑结构的类似,不同之处在于缺少了向种群最优位置学习的机制,可能会降低算法收敛速度,但四边形拓扑结构的信息是连通共享的,依然可以保证算法整体的收敛性.

所改进的两种PSO算法拓扑结构,每个粒子都有两个学习对象,即便其中一个陷入局部最优,也可以通过追随另一个粒子而跳出.粒子在“认知”项信息交流的方式上属于定向流动的闭环结构,当其中一个粒子找到全局最优解,其他粒子会间接或直接向最优粒子有序地靠近.由于种群具有整体收敛趋势,从而无需再对粒子速度进行约束,只要参数设置合理,迭代时间充足,最终可确保整个粒子群收敛到同一最优解.

2.2 粒子变异改进策略

基于拓扑结构改进的PSO算法同其他随机优化算法一样仍存在陷入局部最优的风险.当粒子陷入局部最优而无法按照既定寻优方法跳出时,最简单的跳出方法就是对粒子实施变异^[19].合理的变异措施应在保持现有寻优结果的基础上增加发掘最优位置的概率.参照布谷鸟算法中布谷鸟由于宿主按一定概率发现外来鸟蛋而重新选择巢穴的变异策略^[23],本文对粒子的变异思想描述为:粒子在追随其他粒子的个体最优位置进行搜索时,对粒子*i*的个体最优位置*pbest_i*的每一维赋予一个随机信任度*P*,当*P*小于某一信任度阈值时,则对该维位置进行变异,若变异后的粒子位置*LX_i*优于变异前的*pbest_i*,则用变异后的位置*LX_i*替代变异前的*pbest_i*.变异方法参照布谷鸟算法中的偏好随机游走方法,即

$$LX_{id}(t) = \begin{cases} \text{pbest}_{id}(t), & P > P_a; \\ \text{pbest}_{id}(t) + \text{rand}(\text{pbest}_{jd}(t) - \text{pbest}_{kd}(t)), & P \leq P_a. \end{cases} \quad (12)$$

其中:*pbest*下标*j*和*k*表示同一种群中随机抽取的其他粒子的编号;*rand*是在[0,1]之间服从均匀分布的随机数;*P_a*是信任度阈值,为0~1之间的数.

式(12)的变异操作只针对粒子的个体最优位置,并不影响粒子当前有序的搜索行为,而只是作为一种辅助手段增加找到全局最优解的概率.同时,粒子最大的变异步长不会超过种群粒子间的最长欧氏距离.在算法前期,粒子分散,相距较大,则变异范围也较大,保证了算法搜索的广度;在算法后期,粒子位置

变异的范围将逐渐缩小,从而粒子个体最优位置的变异行为具有收敛性. 信任度阈值 P_a 反映粒子个体最优位置的变异程度, P_a 取值在 0.1 ~ 0.5 之间时可以使算法取得较好的优化效果.

2.3 基于拓扑结构和粒子变异改进的粒子群算法

根据拓扑结构改进策略和粒子变异改进策略,本文建立两种新的 PSO 算法: 环形-星形混合拓扑结构与粒子变异联合建立的 PSO 算法 (Particle swarm optimization with ring-star topology and particle mutation, RSMPSO), 环形-四边形混合拓扑结构与粒子变异联合建立的 PSO 算法 (Particle swarm optimization with ring-von neumann topology and particle mutation, RVMP SO). 两种改进的 PSO 算法以拓扑结构改进为核心,以粒子变异为辅助手段,主要程序结构如下.

Step 1: 程序初始化.

Step 1.1: 设置相关参数: 粒子数 n 、惯性权重 w 、学习因子 c_1 和 c_2 、信任度阈值 p_a 、最大迭代次数 t_{\max} 等.

Step 1.2: 种群初始化. 随机初始化粒子速度 \mathbf{V} 和位置 \mathbf{X} .

Step 1.3: 计算初始适应度值 fitness, 即要优化的目标函数值.

Step 1.4: 记录粒子个体最优位置 pbest、邻域最优位置 lbest 和种群最优位置 gbest.

Step 2: 循环迭代计算.

While(满足迭代停止条件)

1) 根据迭代过程计算惯性权重 w .

2) 按照相应的种群拓扑结构速度更新公式和位置更新公式更新粒子的速度 \mathbf{V} 和位置 \mathbf{X} .

3) 计算新的适应度值 fitness.

4) 根据计算得到的更优解,更新粒子个体最优位置 pbest.

5) 按照式 (12) 对粒子个体最优位置 pbest 进行变异,若变异后的位置 LX 更优,则将 LX 作为 pbest.

6) 根据算法拓扑结构类型更新邻域最优位置 lbest 和种群最优位置 gbest.

7) 存储迭代过程中要记录的信息.

End

Step 3: 结果输出和分析.

3 改进算法性能测试与分析

3.1 测试函数

为测试两种改进 PSO 算法,选取 9 个典型的 Benchmark 测试函数,如表 1 所示. 表中函数分为 3 类,

其中第 1 类函数 f_1 和 f_2 为单峰函数,函数 f_1 的曲面较规则,而函数 f_2 是一个典型病态香蕉型的单峰函数,算法能找到全局最优的机会很小. 函数 $f_2 \sim f_7$ 是第 2 类函数,属于非线性的多模态函数. 这几个测试函数都引入了三角函数项,使得函数曲面存在大量的波峰波谷,在通往全局最优的路径上优化算法很容易陷入一个局部最优点,导致算法难以获得全局最优,当维数增加时优化难度还会加大. 第 3 类函数 f_8 和 f_9 称为旋转多模态函数,函数结构形式与 f_4 和 f_5 相同,只是变量 \mathbf{y} 是由正交矩阵 \mathbf{M} 与 \mathbf{x} 相乘得到^[24],可见旋转多模态函数的变量互相影响,函数曲面极其病态,增加了寻优的复杂性.

3.2 算法参数设置

将 RSMPSO 算法和 RVMP SO 算法与标准的全局版 PSO 算法 (GPSO)、四边形拓扑结构 PSO 算法 (VPSO)^[10]、环形拓扑结构 PSO 算法 (LPSO)^[10]、全面学习的 PSO 算法 (CLPSO)^[12]、随机漂移 PSO 算法 (RDPSO)^[16] 以及布谷鸟搜索算法 (CS)^[23] 等进行比较,以验证所提出的改进 PSO 算法的有效性. 根据相应参考文献,各优化算法的参数设置如表 2 所示. 惯性权重 w 传统做法是随迭代步增加而线性递减. 为了增加算法优化收敛的平滑行,改进的 PSO 算法惯性权重采用非线性递减的策略,公式为

$$w = w_{\max} - (w_{\max} - w_{\min}) \sqrt{t/t_{\max}}. \quad (13)$$

其中: w_{\max} 为最大惯性权重, w_{\min} 为最小惯性权重, t 为当前迭代步, t_{\max} 为最大迭代步.

通常 PSO 算法迭代停止条件为最大迭代次数或找到的最优解与真实最优解的距离满足一定的误差. 显然当真实最优解未知时,该停止条件无法自动控制算法的终止时间. 如果优化算法具有整体收敛趋势,当种群粒子收敛于最优解时,种群粒子的平均适应度值与最小适应度值必然非常接近,反之理解亦然. 为此,本文采用如下迭代停止条件:

$$t = t_{\max} \text{ or } |\text{mean}(\text{fitness}) - \min(\text{fitness})| \leq 10^{-10}. \quad (14)$$

各算法的种群规模为 50,最大迭代次数为 1×10^5 ,以使各算法的优化能力得到较充分的检验. 测试函数的维数均为 30,每个测试函数的寻优计算独立运行 30 次,取寻优结果平均值 (Mean)、标准差 (Std)、平均迭代次数 (Mean FEs) 及计算时间 (Time) 来反映算法优化性能.

本次测试算法的运行环境为: 计算机 CPU 为 i7-4790,内存为 16 GB,软件环境为 Win7 操作系统和 Matlab 2010b.

表1 标准测试函数

函数名称	表达式	维度	搜索范围	最优解	最小值
Quadric	$f_1(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j\right)^2$	30	$[-65.536, 65.536]^D$	$[0, 0, \dots, 0]^D$	0
Rosenbrock	$f_2(\mathbf{x}) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	30	$[-2.048, 2.048]^D$	$[1, 1, \dots, 1]^D$	0
Ackley	$f_3(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{1/D \sum_{i=1}^D x_i^2}\right) - \exp\left(1/D \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	30	$[-32.768, 32.768]^D$	$[0, 0, \dots, 0]^D$	0
Griewank	$f_4(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^D$	$[0, 0, \dots, 0]^D$	0
Rastrigin	$f_5(\mathbf{x}) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30	$[-5.12, 5.12]^D$	$[0, 0, \dots, 0]^D$	0
Schwefel	$f_6(\mathbf{x}) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i }$	30	$[-500, 500]^D$	$[420.97, 420.97, \dots, 420.97]^D$	-12 569.49
Levy	$f_7(\mathbf{x}) = \sin^2(\pi y_1) + \sum_{i=1}^{D-1} \{(y_i - 1)^2 \times [1 + 10 \sin^2(\pi y_i + 1)]\} + (y_D - 1)^2 [1 + \sin^2(2\pi y_D)]$ $y_i = 1 + (x_i - 1)/4$	30	$[-10, 10]^D$	$[1, 1, \dots, 1]^D$	0
Rotated Griewank	$f_8(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D y_i^2 - \prod_{i=1}^D \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1$ $\mathbf{y} = \mathbf{M}\mathbf{x}^T$	30	$[-600, 600]^D$	$[0, 0, \dots, 0]^D$	0
Rotated Rastrigin	$f_9(\mathbf{x}) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10)$ $\mathbf{y} = \mathbf{M}\mathbf{x}^T$	30	$[-5.12, 5.12]^D$	$[0, 0, \dots, 0]^D$	0

表2 各算法参数设置

算法名称	GPSO	LPSO	VPSO	CLPSO	RDPSO	CS	RSMPSO	RVMPPO
	$n = 50$	$n = 50$	$n = 50$	$n = 50$	$n = 50$	$n = 50$	$n = 50$	$n = 50$
算法参数	$t_{\max} = 1 \times 10^5$	$t_{\max} = 1 \times 10^5$	$t_{\max} = 1 \times 10^5$	$t_{\max} = 1 \times 10^5$	$t_{\max} = 1 \times 10^5$	$t_{\max} = 1 \times 10^5$	$t_{\max} = 1 \times 10^5$	$t_{\max} = 1 \times 10^5$
	$w: 0.9 \rightarrow 0.4$	$w: 0.9 \rightarrow 0.4$	$w: 0.9 \rightarrow 0.4$	$w: 0.9 \rightarrow 0.4$	$\alpha: 0.9 \rightarrow 0.3$	$\alpha = 0.01$	$w: 0.9 \rightarrow 0.5$	$w: 0.9 \rightarrow 0.5$
	$c_1 = c_2 = 2$	$c_1 = c_2 = 2$	$c_1 = c_2 = 2$	$c = 1.49445$	$c_1 = c_2 = 2$	$\beta = 1.5$	$c_1 = c_2 = 2$	$c_1 = c_2 = 2$
	$V_{\max} = X_{\max}$	$V_{\max} = X_{\max}$	$V_{\max} = X_{\max}$	$m = 7$	$\beta = 1.5$	$P_a = 0.25$	$P_a = 0.25$	$P_a = 0.25$
				$V_{\max} = X_{\max}$	$V_{\max} = X_{\max}$			

注: 对于 Rosenbrock 函数和 Schwefel 函数, 改进的 PSO 算法采用速度约束可以取得更好的优化效果。

3.3 结果分析

3.3.1 优化精度及效率

表3和表4给出了各优化算法的优化结果和计算速度,其中字体加粗部分表示对某个测试函数最好的寻优结果。

由计算结果可以看出, GPSO、LPSO和VPSO只对部分函数有较好的优化效果,而对函数Rastrigin、Rosenbrock和Rotated Rastrigin等优化效果较差。CLPSO是标准PSO算法的较大改进,但仍然在一些函数如Rosenbrock上表现不尽人意。RDPSO在优化精度和结果标准差方面相比标准PSO改进效果不明显。CS算法是较新提出的优化算法,在优化性能

方面比标准的PSO算法有较大提升,而RSMPSO和RVMPPO相比CS算法有更大的进步。RSMPSO和RVMPPO相对于GPSO和VPSO,区别仅在于速度更新公式上的“认知”项和外加一个变异措施,但本文改进算法优化能力在各个函数上都得到了大幅度提升,优化精度较高,且优化结果标准差较小,优化性能稳定可靠,可见所提出的改进措施卓有成效。

从各算法的优化计算速度来看, GPSO、LPSO和VPSO停机迭代次数多,基本要达到最大迭代次数要求才结束计算,但所消耗时间并不是很高,主要得益于算法结构简单。CLPSO不仅迭代次数多,且算法结构复杂,以致计算时间在这些算法当中最长,虽然优

表3 各优化算法在测试函数上的优化结果比较

测试函数		GPSO	LPSO	VPSO	CLPSO	RDPSO	CS	RSMP SO	RVMP SO
Quadric	Mean	1.45e-36	2.76e-17	2.39e-21	4.85e-09	2.25e-11	5.44e-11	1.81e-13	2.49e-13
	Std	4.12e-36	5.07e-17	4.07e-21	3.25e-09	1.34e-11	5.42e-11	6.92e-14	1.07e-13
Rosenbrock	Mean	1.55e+01	5.74e+00	3.76e+00	2.08e+01	1.95e+01	1.84e-11	9.49e-14	3.07e-13
	Std	4.25e+00	2.87e+00	2.17e+00	3.14e+00	8.04e-01	3.86e-11	2.32e-13	4.29e-13
Ackley	Mean	9.05e-12	2.06e-11	2.98e-11	8.07e-11	7.05e-11	1.91e-10	1.38e-14	7.89e-13
	Std	2.82e-12	1.02e-11	6.02e-12	3.45e-11	1.34e-11	3.52e-11	3.02e-15	1.85e-15
Griewank	Mean	1.95e-02	0.00e+00	2.46e-03	0.00e+00	5.66e-03	0.00e+00	0.00e+00	0.00e+00
	Std	2.04e-02	0.00e+00	7.87e-03	0.00e+00	7.03e-03	0.00e+00	0.00e+00	0.00e+00
Rastrigin	Mean	7.82e+00	1.42e+01	9.95e+00	4.58e-14	9.82e+00	3.32e-02	4.74e-16	5.92e-17
	Std	3.73e+00	2.76e+00	3.45e+00	2.58e-13	2.95e+00	1.82e-01	1.23e-15	3.24e-16
Schwefel	Mean	-12 569.49	-12 569.49	-12 569.49	-12 569.49	-10 655.16	-12 569.49	-12 569.49	-12 569.49
	Std	2.25e-12	2.25e-12	2.36e-12	3.05e-12	1.12e+03	2.41e-12	5.41e-12	1.85e-12
Levy	Mean	4.79e-14	5.83e-13	1.53e-12	1.48e-12	1.17e-11	5.46e-11	1.94e-25	1.31e-25
	Std	3.58e-14	4.49e-13	8.83e-13	4.87e-12	4.45e-12	2.59e-11	4.41e-25	4.02e-25
Rotated Griewank	Mean	1.85e-02	1.60e-03	4.06e-03	2.12e-12	5.58e-03	0.00e+00	4.07e-17	7.40e-18
	Std	2.01e-02	4.02e-03	7.99e-03	3.48e-12	7.68e-03	0.00e+00	1.86e-16	2.82e-17
Rotated Rastrigin	Mean	4.28e+01	2.79e+01	2.86e+01	2.12e+01	1.91e+01	4.53e+01	1.43e+01	1.06e+01
	Std	1.21e+01	5.45e+00	6.58e+00	3.42e+00	6.06e+00	1.04e+01	7.61e+00	7.86e+00

表4 各优化算法在测试函数上的计算速度比较

测试函数		GPSO	LPSO	VPSO	CLPSO	RDPSO	CS	RSMP SO	RVMP SO
Quadric	Mean FEs	100 000	100 000	100 000	100 000	21 481	20 786	41 292	47 444
	Time/s	20.82	20.94	38.69	365.12	5.26	11.23	12.62	23.72
Rosenbrock	Mean FEs	100 000	100 000	100 000	100 000	100 000	50 130	47 299	52 033
	Time/s	16.43	16.89	33.79	224.98	19.52	21.85	12.31	22.36
Ackley	Mean FEs	47 625	56 489	55 659	22 896	2 022	4301	9 679	9 602
	Time/s	7.78	10.75	19.76	53.56	0.41	1.95	2.35	4.16
Griewank	Mean FEs	53 969	80 190	79 923	100 000	4 415	10 259	11 634	11 699
	Time/s	9.79	16.41	28.16	293.38	0.96	4.89	3.38	5.62
Rastrigin	Mean FEs	100 000	100 000	100 000	47 096	25 292	24 872	15 152	14 549
	Time/s	16.96	19.24	35.95	88.67	5.39	11.05	3.80	6.36
Schwefel	Mean FEs	82 202	85 518	100 000	22 378	97 559	13 576	27 275	29 696
	Time/s	14.21	17.25	34.47	35.45	21.99	6.27	7.57	13.84
Levy	Mean FEs	44 487	51 416	50 278	18 363	1 451	3 389	8 549	8 481
	Time/s	7.62	10.23	17.52	38.25	0.32	1.64	2.44	3.97
Rotated Griewank	Mean FEs	50 393	100 000	100 000	100 000	3 326	46 295	12 868	12 582
	Time/s	12.32	25.98	39.30	298.56	1.02	25.91	4.99	8.03
Rotated Rastrigin	Mean FEs	94 425	100 000	100 000	100 000	55 406	100 000	21 577	83 769
	Time/s	19.72	24.95	41.31	191.45	16.41	53.75	7.49	46.52

化精度表现较好,但优化效率没有优势. RDPSO 和 CS 算法在大多数函数上满足平均适应度值与最小适应度值差值小于某一极小值的迭代停止条件,加之算法结构简单,所以消耗计算时间较少,特别是 RDPSO 在个别函数(如 Ackley 函数和 Levy 函数)的优化上有极高的优化效率. 本文改进的两种 PSO 算法在所测试的函数中均能满足适应度值收敛的迭代停止条件. RSMP SO 不仅优化精度高,优化迭代次数也少,虽然算法结构较标准的 PSO 复杂,但由于较少的迭代次数使得算法计算时间消耗相对较少. RVMP SO 相比 RSMP SO 算法结构稍复杂,在计算

时间上稍有增加,但比起其他 PSO 算法仍表现出较好的计算效率.

3.3.2 优化过程收敛性

各 PSO 算法拓扑结构的不同,决定了它们的收敛性能也存在差异. 从表4中可以初步看出,凡是达到最大迭代次数才能终止程序的算法,种群的平均适应度值与最小适应度的差值未能满足给定精度要求,可见种群在最后未能收敛. 这里以各算法优化效果具有较大差异的函数 Rastrigin 为例,给出各优化算法的平均适应度值和最小适应度值的变化曲线,进一步分析各优化算法的收敛特性,如图4所示.

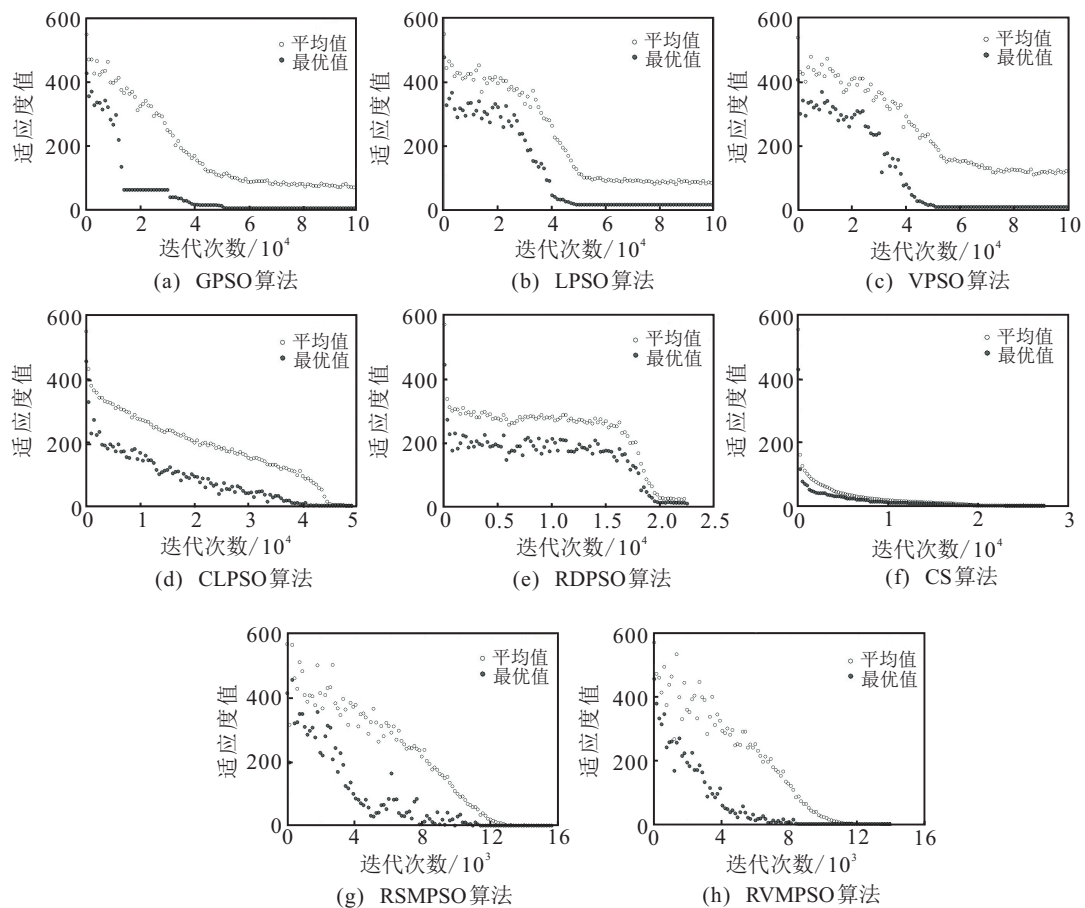


图4 函数Rastrigin优化收敛曲线

由图4可看出,GPSO、LPSO和VPSO三种算法的平均适应度值和最小适应度值变化曲线未能在最后收敛在一起.之所以会产生这种情况,主要是因为速度更新公式中的“认知”项造成粒子在搜索过程呈现振荡趋势以致不能收敛,而且粒子的这种发散效应并未能显著提高寻优精度. CLPSO算法采用全面向其他粒子学习的进化方式,具有引导粒子向种群最优粒子收敛的趋势,但由于引导过程增加了较多的随机措施,使得CLPSO算法的收敛性不够稳定,如对于函数Rosenbrock和函数Griewank就不能收敛. RDPSO算法对粒子实施定向漂移运动,从而使算法的收敛性较好,但从表3也可以看出,算法容易陷入局部最优而降低优化精度. CS算法每一步的搜索都是参考种群最优位置进行的,收敛速度快,且其特有的Levy飞行搜索机制使算法具有较高的寻优精度. RSMPSO和RVMPSTO的优化曲线表明,算法前期粒子群适应度值分散,多样性较高,而因其特有的环形封闭拓扑结构,搜索信息最终向种群最优处流动,保证了算法的整体收敛性,收敛迭代次数也较少.收敛性是算法获得较高寻优精度的重要保证,结合表4来看,两种改进PSO算法的收敛性不会因测试函数的不同而变化,因此对各种函数都具有较高的寻优精度.

3.3.3 优化算法稳健性

优化算法的稳健性体现于各种复杂优化函数的适应性和算法参数的普适性,如果算法只针对某一类优化问题或者针对不同的优化问题需要调整不同的算法参数才能取得较好的优化效果,则说明算法优化性能的稳健性较差.表3和表4的优化结果表明,相比其他6种算法,改进的RSMPSO和RVMPSTO对所有测试函数都获得了较高的优化精度和计算效率,且计算过程中不必做过多的参数调整,算法结构简单,体现了较稳健的优化性能.

为了检验算法参数对算法稳健性的影响情况,以PSO算法参数惯性权重 w 为对象进行分析.惯性权重 w 采用5种不同的调整策略进行对比,包括常数策略(w_1)、线性下降策略(w_2)、凹曲线下降策略(w_3)、指数曲线下降策略(w_4)和随机策略(w_5).以较难优化的单峰函数Rosenbrock为测试对象,以优化30次的平均值和标准差进行比较,结果如表5所示.由结果可知,GPSO、LPSO、VPSO和CLPSO的优化结果受惯性权重调整策略的影响较大,不同的调整策略会得到不同的优化结果,且各算法取得最优结果对应的最优调整策略也不同,优化精度也不高,可见这些算法

表5 基于不同惯性权重调整策略的各优化算法优化性能比较

惯性权重类型		GPSO	LPSO	VPSO	CLPSO	RSMP SO	RVMP SO
$w_1 = 0.7298$	Mean	2.43e+01	2.15e+01	2.54e+01	8.03e+00	6.22e-17	1.29e-16
	Std	3.67e-01	6.05e+00	4.13e-01	5.01e+00	4.57e-17	1.45e-16
$w_2 = w_{max} - (w_{max} - w_{min})t/t_{max}$	Mean	1.55e+01	5.74e+00	3.76e+00	2.08e+01	6.63e-14	6.86e-14
	Std	4.25e+00	2.87e+00	2.17e+00	3.14e+00	1.08e-13	1.17e-13
$w_3 = w_{max} - (w_{max} - w_{min})\sqrt{t/t_{max}}$	Mean	2.13e+01	5.79e+00	3.19e-02	1.08e+01	8.49e-14	4.07e-13
	Std	1.71e+01	1.58e+00	3.11e-02	1.96e+00	2.12e-13	4.24e-13
$w_4 = w_{min} + (w_{max} - w_{min})e^{-5(t/t_{max})^2}$	Mean	1.56e+01	4.06e+00	1.91e-02	2.01e+01	3.60e-13	7.74e-13
	Std	2.84e+00	6.67e-01	2.27e-02	5.54e-01	4.91e-13	1.73e-14
$w_5 = 0.5 + rand/2$	Mean	3.14e+01	2.65e+01	2.66e+01	2.97e+02	3.72e-18	5.75e-18
	Std	1.73e+01	1.04e+00	8.61e-01	2.69e+01	3.64e-18	6.08e-18

注: w_{max} 、 w_{min} 分别为0.9和0.4.

的稳健性不强. 而RSMP SO和RVMP SO算法的优化结果基本不受调整策略的影响, 在各种惯性权重的调整策略下都取得了非常好的优化效果.

测试函数的维数越高, 函数越复杂, 优化的难度也越大. 将函数Rosenbrock的维数由小到大逐步提高至100维, 得到各算法优化结果曲线对比如图5所示. 可见, GPSO、LPSO、VPSO、CLPSO、RDPSO随着维数的增加, 优化出的最小适应度值也越来越大, 对于高维函数存在优化困难. 而CS、RSMP SO和RVMP SO并没有因维数的增加而降低优化精度, 直至维数达到100维时, RSMP SO和RVMP SO的优化精度才有所降低. 事实上, RSMP SO和RVMP SO可以通过增加终止程序的最大迭代次数条件来提高高维

函数的优化精度. 如图6所示, 当最大迭代次数设为200 000代时, RSMP SO和RVMP SO算法与CS算法一样可以成功寻优, 而其他5种PSO算法却未能通过增加迭代次数来提高优化精度.

RSMP SO和RVMP SO的特点是在算法的速度更新公式的“认知”项采用了信息定向流动的闭环结构, 这是算法具有整体收敛性的重要保证, 将这种闭环结构与其他拓扑结构(如Pyramid型或Four cluster型)结合使用, 也可以构造出类似RSMP SO和RVMP SO的改进PSO算法, 这里不再详述.

4 结论

本文在分析现有PSO算法存在的弊端基础上, 提出了能使算法优化过程的多样性与收敛性得到合理解决的两种改进PSO算法RSMP SO和RVMP SO. 改进的PSO算法采用信息定向流动的闭环拓扑结构与全局版的星型拓扑结构或局部版的四边形拓扑结构相结合的方式, 能有效促使粒子在前期寻优过程中具有较高的多样性, 保证搜索的广度, 而在后期满足群体的整体收敛性, 保证寻优的精度; 同时, 在改进的PSO算法中增加了对粒子个体最优位置的变异操作, 使算法能有效跳出局部最优, 增大了找到全局最优解的概率. 通过对测试函数的仿真实验可知, RSMP SO和RVMP SO算法不仅具有较高的寻优精度, 在计算效率上也比其他PSO算法更高. 通过比较各算法优化过程的收敛性和算法稳健性可以看出, RSMP SO和RVMP SO算法可以保证优化过程最终收敛于同一最优解, 保证算法寻优的精度, 并且受算法参数的影响较小, 具有较强的算法稳健性. 所改进的两种PSO算法相比标准PSO算法在计算复杂度和计算时间上并未增加多少, 而优化性能却显著提高, 可作为标准PSO算法的替代.

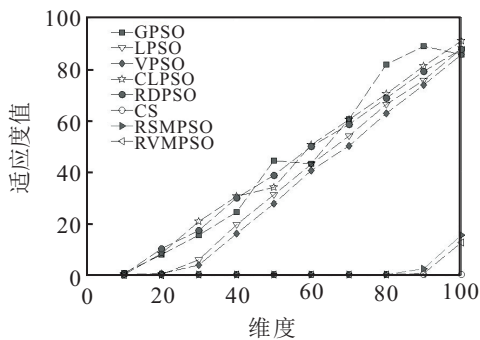


图5 函数Rosenbrock优化结果随维度变化曲线

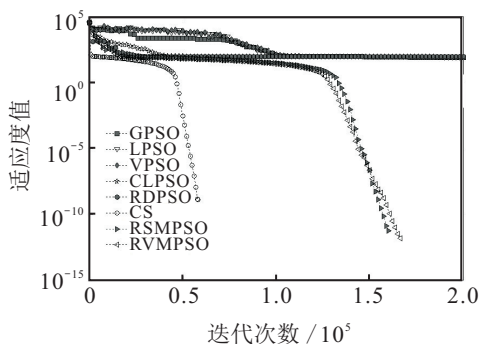


图6 函数Rosenbrock(100维)优化收敛曲线

参考文献(References)

- [1] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of the 6th Int Symposium on Micro Machine and Human Science. Nagoya: IEEE, 1995: 39-43.
- [2] Poli R. An analysis of publications on particle swarm optimization applications[R]. London: Department of Computer Science In University of Essex, 2007: 1-41.
- [3] Shi Y, Eberhart R. A modified particle swarm optimizer[C]. Proc of IEEE Int Conf on Evolutionary Computation. Piscataway: IEEE, 1998: 69-73.
- [4] 戴文智, 杨新乐. 基于惯性权重对数递减的粒子群优化算法[J]. 计算机工程与应用, 2015, 51(17): 14-19.
(Dai W Z, Yang X L. Particle swarm optimization algorithm based on inertia weight logarithmic decreasing[J]. Computer Engineering and Applications, 2015, 51(17): 14-19.)
- [5] 吴秋波, 王允诚, 赵秋亮, 等. 混沌惯性权值调整策略的粒子群优化算法[J]. 计算机工程与应用, 2009, 45(7): 49-51.
(Wu Q B, Wang Y C, Zhao Q L, et al. Particle swarm optimization with chaotic inertia weight adjusting strategy[J]. Computer Engineering and Applications, 2009, 45(7): 49-51.)
- [6] 赵志刚, 黄树运, 王伟倩. 基于随机惯性权重的简化粒子群优化算法[J]. 计算机应用研究, 2014, 31(2): 361-363.
(Zhao Z G, Huang S Y, Wang W Q. Simplified particle swarm optimization algorithm based on stochastic inertia weight[J]. Application Research of Computers, 2014, 31(2): 361-363.)
- [7] Zheng Y J, Ling H F, Guan Q. Adaptive parameters for a modified comprehensive learning particle swarm optimizer[J]. Mathematical Problems in Engineering, 2012, 68(3): 939-955.
- [8] Zhan Z H, Zhang J, Li Y, et al. Adaptive particle swarm optimization[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B, 2009, 39(6): 1362-1381.
- [9] 高哲, 廖晓钟. 基于平均速度的混合自适应粒子群算法[J]. 控制与决策, 2012, 27(1): 152-155.
(Gao Z, Liao X Z. Hybrid adaptive particle swarm optimization based on average velocity[J]. Control and Decision, 2012, 27(1): 152-155.)
- [10] Kennedy J, Mendes R. Population structure and particle swarm performance[C]. Proc of the 2002 Congress on Evolutionary Computation. Washington: IEEE, 2002, 2: 1671-1676.
- [11] Van den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 225-239.
- [12] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Trans on Evolutionary Computation, 2006, 10(3): 281-295.
- [13] Wang H, Sun H, Li C, et al. Diversity enhanced particle swarm optimization with neighborhood search[J]. Information Sciences, 2013, 223: 119-135.
- [14] 石松, 陈云. 层次环形拓扑结构的动态粒子群算法[J]. 计算机工程与应用, 2013, 49(8): 1-5.
(Shi S, Chen Y. Dynamic particle swarm optimization algorithm with hierarchical ring topology[J]. Computer Engineering and Applications, 2013, 49(8): 1-5.)
- [15] 王华秋, 曹长修. 基于模拟退火的并行粒子群优化研究[J]. 控制与决策, 2005, 20(5): 500-504.
(Wang H Q, Cao C X. Parallel particle swarm optimization based on simulated annealing[J]. Control and Decision, 2005, 20(5): 500-504.)
- [16] Sun J, Wu X, Palade V, et al. Random drift particle swarm optimization algorithm: Convergence analysis and parameter selection[J]. Machine Learning, 2015, 101(1/3): 345-376.
- [17] 李蓉, 沈云波, 刘坚. 改进的自适应粒子群优化算法[J]. 计算机工程与应用, 2015, 51(13): 31-36.
(Li R, Shen Y B, Liu J. Improved adaptive particle swarm optimization algorithm[J]. Computer Engineering and Applications, 2015, 51(13): 31-36.)
- [18] 高云龙, 闫鹏. 基于多种群粒子群算法和布谷鸟搜索的联合寻优算法[J]. 控制与决策, 2016, 31(4): 601-608.
(Gao Y L, Yan P. Unified optimization based on multi-swarm PSO algorithm and cuckoo search algorithm[J]. Control and Decision, 2016, 31(4): 601-608.)
- [19] 周利军, 彭卫, 邹芳, 等. 自适应变异粒子群算法[J]. 计算机工程与应用, 2016, 52(7): 50-55.
(Zhou L J, Peng W, Zou F, et al. Particle swarm optimization based on self-adaptive mutation[J]. Computer Engineering and Applications, 2016, 52(7): 50-55.)
- [20] 袁晗, 徐春梅, 杨平, 等. 一种基于子群变异的粒子群优化算法[J]. 计算机应用研究, 2017, 34(4): 1076-1079.
(Yuan H, Xu C M, Yang P, et al. New particle swarm optimization based on subswarm mutation[J]. Application Research of Computers, 2017, 34(4): 1076-1079.)
- [21] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [22] Trelea I C. The particle swarm optimization algorithm: convergence analysis and parameter selection[J]. Information Processing Letters, 2003, 85(6): 317-325.
- [23] Yang X S, Deb S. Engineering optimisation by cuckoo search[J]. Int J of Mathematical Modelling & Numerical Optimisation, 2010, 1(4): 330-343.
- [24] Salomon R. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms[J]. Bio Systems, 1996, 39(3): 263-278.