

文章编号: 1001-0920(2013)07-1087-07

基于混合策略自适应学习的并行粒子群优化算法

伍大清^{1,2,3}, 郑建国¹

(1. 东华大学 旭日工商管理学院, 上海 200051; 2. 南华大学 计算机科学与技术学院, 湖南 衡阳 421001; 3. 人工智能四川省重点实验室, 四川 自贡 643000)

摘要: 针对当前各种粒子群优化算法解决问题时存在的局限性, 提出一种基于混合策略自适应学习的粒子群优化算法(HLPSO). 该算法从收敛速度、跳出局部极值、探索、开发几个不同角度融合了4种具有不同优势的变异策略, 当面对不同形态的复杂问题时通过自适应学习机制选择出合适的策略来完成全局寻优. 通过对7个标准测试函数的仿真实验并与其他算法相比较, 所得结果表明了所提出的算法具有较快的收敛速度、较高的精度以及很强的跳出局部极值的能力.

关键词: 粒子群优化; 自适应学习; 变异策略; 函数优化

中图分类号: TP301

文献标志码: A

Improved parallel particle swarm optimization algorithm with hybrid strategy and self-adaptive learning

WU Da-qing^{1,2,3}, ZHENG Jian-guo¹

(1. Glorious Sun School of Business and Management, Donghua University, Shanghai 200051, China; 2. Department of Computer Science and Technology, University of South China, Hengyang 421001, China; 3. Artificial Intelligence Key Laboratory of Sichuan Province, Zigong 643000, China. Correspondent: WU Da-qing, E-mail: DQW_1982@126.com)

Abstract: In view of the limitation of the current kinds of particle swarm optimization(PSO) algorithm, a self-adaptive learning of hybrid strategy algorithm based on parallel particle swarm optimization(HLPSO) is proposed. The algorithm combines four strategies reasonably in the different point of view: Convergence, jump out, exploration and exploitation, which chooses an appropriate strategies to solve the different forms of problems through adjusting the probability of the strategies gradually in the process of optimizing. Moreover, simulation experiment on a suite of 7 benchmark functions is given, and the comparisons with other algorithms are provided. The results show that the proposed approach has better convergence rate and great capability of preventing premature convergence.

Key words: particle swarm optimization; self-adaptive learning; mutation strategy; function optimization

0 引言

粒子群优化算法(PSO)^[1]是 Kennedy 和 Eberhart 于 1995 年提出的一种全局优化进化算法, 它源于对鸟群和鱼群群体觅食运动行为的模拟. 粒子群优化算法具有收敛速度快、设置参数少、算法简单、容易实现等特点, 已被广泛应用于函数优化计算、管理决策^[2-3]和工程领域^[4]. 随着问题复杂性的增加, 现有的粒子群优化模型已越来越难以满足实际问题的需要, 很多研究者相继从不同角度对标准粒子群算法进行了改进, 以提高算法的性能. 迟玉红等^[5]提出了一种

基于空间缩放和吸引子的粒子群优化算法, 该算法能够保证算法全局探测能力和局部开发能力; 张顶学等^[6]提出了一种基于种群速度的自适应粒子群算法; Liang 等^[7]提出的 CLPSO 算法是为了解决复杂的多峰问题而提出的, 该算法拥有非常好的多样性保持机制, 但在单峰优化问题上的寻优效率较低; 为改进 CLPSO 算法存在的不足, Li 等^[8]提出了一种自适应的学习策略, 根据种群中粒子的运行状况动态地为每个粒子指派学习样本, 以增强粒子间的信息交流; 胡旺等^[9]提出了与速度无关的简化粒子群优化算法,

收稿日期: 2012-03-06; 修回日期: 2012-07-08.

基金项目: 国家自然科学基金项目(70971020/G010301); 人工智能四川省重点实验室项目(2012RYJ03).

作者简介: 伍大清(1982-), 女, 讲师, 博士生, 从事智能计算的研究; 郑建国(1962-), 男, 教授, 博士生导师, 博士后, 从事数据挖掘等研究.

用以提高收敛速度和精度; Mendes 等^[10]提出了牵制粒子飞行趋势的不应该只是该粒子的邻接粒子的观点, 并据此提出了一种广泛利用搜索信息的 FIPS 算法; 贾树晋等^[11]为了提高算法的收敛性和非支配解集的多样性, 提出了一种基于局部搜索与混合多样性策略的多目标粒子群算法 (LH-MOPSO). 目前, 绝大多数关于 PSO 算法的研究是从改进原始标准 PSO 算法的变异算子入手, 以达到提升 PSO 在某一方面性能的目的, 比如收敛速度、搜索精度和求解规模较大问题的能力等, 往往只能解决某一类复杂问题, 而 PSO 算法在处理多种问题上均具有很强的普适性能力仍有待提高. 目前, 自适应学习方法已经成为不少学者研究的热点问题^[12-13], 许多工作已经证实了自适应学习框架的引入有助于增强算法同时应对多种不同复杂问题的能力.

本文提出一种基于混合策略自适应学习机制的并行粒子群优化 (HLPSO) 算法, 从不同的角度融合了 4 种具有不同优势的变异策略, 通过自适应学习机制选出恰当的策略来解决不同形态的复杂问题, 以期达到因地制宜的效果.

1 基本粒子群优化算法概述

在标准 PSO 算法中, 假设在一个 D 维的空间中包含 m 个粒子, 每个粒子作为搜索空间中待优化问题的一个可行解, 通过粒子之间的协作与竞争来寻找问题的最优解. 在第 t 次迭代中, 第 i 个粒子的当前位置表示为 $x_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{id}(t))$, 当前速度表示为 $v_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{id}(t))$. 在每次迭代中, 粒子个体搜索到的最好位置用 $pb_i(t) = (pb_{i1}(t), pb_{i2}(t), \dots, pb_{id}(t))$ 表示, 记作 pb_{best_i} , 称为个体最优; 群体中所有粒子搜索到的最好位置用 $gb(t) = (gb_1(t), gb_2(t), \dots, gb_d(t))$ 表示, 记作 gb_{best} , 称为全局最优. 优化问题的过程可看作粒子不断更新的过程, 每个粒子根据式 (1) 和 (2), 以当前速度、个体最优位置和全局最优位置来调整自身的飞行速度和方向, 有

$$v_{id}(t+1) = wv_{id}(t) + c_1r_1(pb_{id}(t) - x_{id}(t)) + c_2r_2(gb_d(t) - x_{id}(t)), \quad (1)$$

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

其中: $i = 1, 2, \dots, m$; $d = 1, 2, \dots, D$; r_1 和 r_2 为服从 $U(0, 1)$ 分布的随机数; 学习因子 c_1 和 c_2 为非负常数, 通常取 $c_1 = c_2 = 2$; w 为惯性系数.

2 混合策略自适应学习的粒子群优化算法

2.1 不同角度的并行混合策略

针对不同形态问题的复杂程度, 所选用的 PSO 模型也应该不同, 即更新策略不同, 如何将不同策略通过自适应机制融合在一起是目前亟需解决的问题.

本文先从以下几种不同类型的函数优化展开讨论: 首先, 对于单模问题或极值点之间距离比较远的多模问题, 粒子最优的学习策略是向它们邻近的最佳粒子学习, 以便有效地找到极值点; 其次, 对于类似于 Rosenbrock 病态优化问题, 极值点处在一个非常平滑的斜坡上, 最好的策略是学习粒子自身的最优位置, 这样比从一个远距离区域去搜索一个更好的位置更容易得到全局最优解; 再次, 对于存在多个局部极值并均匀分布的问题, 广泛学习邻居信息更有利于粒子探索更广阔的搜索空间, 从而实现全局寻优; 最后, 对于已经停滞在局部极值的粒子, 唯一选择就是应用扰动算子让它跳出局部极值来寻找更优的位置, 显然该操作也能帮助粒子探索更广阔的搜索空间. 基于以上思想, 为了提高粒子群算法的普适性和实用性, 本文从快速收敛到全局最优值、开发个体最佳位置、探索新的前景区域、跳离局部最优解 4 个目的出发, 融合以下 4 种策略.

策略 1 (收敛策略) 文献 [7,14] 已经详细证明了标准 PSO 算法可以没有粒子速度的概念. 显然, 仅由粒子位置控制进化过程, 可以简化分析和控制粒子的进化过程, 能够极大地提高收敛速度和精度, 避免由粒子速度项引起的粒子发散而导致后期收敛变慢和精度低的问题. 因此, 为了提高搜索速度应引入该收敛策略. 不含速度项的粒子群优化方程如下:

$$x_i^d(t+1) = wx_i^d(t) + c_1r_1^d(pb_i^d(t) - x_i^d(t)) + c_2r_2^d(gb^d(t) - x_i^d(t)). \quad (3)$$

策略 2 (扰动策略) 当算法处于进化停滞时, 粒子群中的粒子都会出现早熟, 从而“聚集”到某一位置, 直至突破进化停滞局面粒子才会“飞散”开去. 文献 [14] 通过严格的数学推导得到

$$\lim_{t \rightarrow \infty} x(t) = p^* = \frac{c_1r_1}{c_1r_1 + c_2r_2}pb + \frac{c_1r_1}{c_1r_1 + c_2r_2}gb. \quad (4)$$

由式 (4) 可得出粒子将聚集到由自身极值 pb 和群体全局极值 gb 决定的极值之上, 如果所有粒子在向 p^* 靠近的过程中没有找到优于 gb 的位置, 则进化过程将处于停滞状态, 粒子逐渐聚集到 p^* . 如果调整个体极值 pb' 和全局极值为 gb' , 使所有粒子飞向新的位置 p^* , 从而经历新的搜索路径和领域, 则发现更优解的概率会比较大. 该策略引进进化停滞代数, 增加极值扰动算子对个体极值 pb 和全局极值 gb 同时进行随机扰动, 从而能够让粒子快速跳出局部极值点, 此时有

$$v_i^d(t+1) = wx_i^d(t) + c_1r_1^d(r_3pb_i^d(t) - x_i^d(t)) + c_2r_2^d(r_4gb^d(t) - x_i^d(t)). \quad (5)$$

令 s_p 和 t_p 分别表示个体极值进化停滞代数和需要扰动的停滞代数阈值, s_g 和 t_g 分别表示全局极值进化停

滞代数和需要扰动的停滞代数阈值, 则

$$r_3(r_4) = \begin{cases} 1, & \text{else;} \\ U(0, 1), & s_p > t_p (s_g > t_g). \end{cases}$$

策略3 (探索策略) 文献[7]已经证明, 让粒子的每个维度随机地向自身或其他粒子学习, 让粒子在更大的潜在空间飞行有利于全局搜索. 该学习策略能够极大地增强种群的多样性.

$$v_i^d(t+1) = wv_i^d(t) + c_1 r_i^d(\text{pb}_{\text{rand}(i)}^d(t) - x_i^d(t)). \quad (6)$$

策略4 (开发策略) 对于一些难以求解的单峰问题, 让粒子从自身最优位置进行学习, 有利于粒子进一步探索更优的极值点.

$$v_i^d(t+1) = wv_i^d(t) + c_1 r_i^d(\text{pb}_i^d(t) - x_i^d(t)). \quad (7)$$

2.2 自适应学习机制

HLPSO算法采用自适应学习机制动态调整每一种策略的执行概率, 以增强算法的灵活性, 从而有效地拓展PSO的适用范围. 具体操作如下: 首先给每一种策略分配一个统一的初始执行概率 P_k , 初始化第 k 种策略的累加器 $C_k = 0$, 在每次迭代中, 新生成的粒子将按照它们的适应度进行排序; 然后, 按照式(8)顺序给每个粒子分配权重; 最后, 每一个粒子的权重将被累加到对应策略的累加器中. 经过一定数量的代数 G_s 之后, 下面的计算规则将被用于调整第 k 种策略的执行概率:

$$w_i = \frac{\log(N - i + 1)}{\log(1) + \log(2) + \dots + \log(n)}, \quad (8)$$

$$P_k = \frac{C_k}{G} \beta + (1 - \beta)P_k + \varepsilon, \quad (9)$$

$$P_K = P_K / \sum_{k=1}^K P_K. \quad (10)$$

其中: N 为粒子数; p_k 为第 k 种策略每次更新后的概率; β 为学习系数; C_k 与 G_s 的比值保证学习样本小于1; ε 为每种策略的最小选择概率, 以保证每种策略都能被随机选中; 式(10)是对每种策略的执行概率进行归一化.

2.3 基于自适应多策略融合的粒子群优化算法

HLPSO算法描述如下.

Step 1: 初始化粒子群. 随机初始化粒子数为 N 的群体以及对应的适应度, 各个粒子所对应的速度, 初始化迭代计数器MaxCycle, 粒子的局部最优位置pb和群体最佳位置gb.

Step 2: 初始化自适应策略. 初始化策略的执行概率 $P_k = 1/k, k = 1, 2, 3, 4$, 设置自适应学习周期为 $L_s = 10$, 学习强度 $\alpha = 1/5, \varepsilon = 0.01$, 按式(10)设置各个粒子权重以及各个策略的累加器 $C_k = 0$, 设置个体极值和全局极值需扰动的停滞代数阈值 $t_p = t_g = 5$.

Step 3: 迭代更新. 对种群中所有的粒子执行如下

操作.

Step 3.1: 通过随机轮盘赌策略选择出第 k 种策略来更新第 i 个粒子. 如果选择策略1, 则用式(3)更新粒子速度和位置; 如果选择策略2, 则用式(5)和(2)更新粒子速度和位置; 如果选择策略3, 则用式(6)和(2)更新粒子的位置; 如果选择策略4, 则用式(7)和(2)更新粒子的速度和位置.

Step 3.2: 计算粒子的适应度值, 分别更新粒子的局部最优位置pb和群体最佳位置gb. 注意, 这里对速度越界的粒子适应度值不进行更新.

Step 3.3: 计算粒子的适应度值, 用式(8)计算各个粒子的权重, 并累加到对应的策略累加器 C_k 中.

Step 3.4: 判断迭代次数是否能整除自适应学习周期 L_s . 若是, 则用式(9)和(10)更新4种策略选择概率, 否则转Step 3.5.

Step 3.5: 判断是否满足算法终止条件. 若满足, 则执行Step 4; 否则迭代次数 $t = t + 1$, 自适应评价代数 $FES = FES + 1$, 转Step 3.1.

Step 4: 输出最优解位置及对应的适应度值, 算法结束.

2.4 HLPSO进化方程的收敛性能分析

HLPSO算法在自适应学习机制的微调下, 每个粒子在不同阶段根据需要选择出适合自身的策略来完成寻优, 因此对HLPSO的收敛性分析必须对4种更新策略分别进行分析才能得到该混合算法的收敛性.

定理1 若收敛策略的平均行为根据其期望值进行观察, 则当 $0 < w < \frac{c_1 + c_2}{2}$ 时, 该策略收敛于 $\frac{c_1 p_0 + c_2 p_g}{c_1 + c_2 - 2w}$.

证明 由式(1)和(2)组成的联合进化方程, 去掉PSO进化方程的粒子速度项便可使原来的二阶微分方程简化为一阶微分方程. 令 $\varphi_1 = r_1 c_1, \varphi_2 = r_2 c_2, \varphi = \varphi_1 + \varphi_2, \rho = \frac{\varphi_1 p_b + \varphi_2 p_g}{\varphi_1 + \varphi_2}$, 则可以将式(1)和(2)转变成

$$v(t+1) = \omega v(t) + \varphi(\rho - x(t)), \quad (11)$$

$$x(t+1) = x(t) + v(t+1). \quad (12)$$

将式(11)与(12)迭代, 可以得到

$$x(t+2) + (\phi - w - 1)x(t+1) + wx(t) = \varphi\rho. \quad (13)$$

根据以上符号定义, 式(3)可以变形为如下二阶微分方程:

$$x(t+1) + (\varphi - w)x(t) = \varphi\rho. \quad (14)$$

式(14)的微分方程解为 $x(t) = C e^{(w-\rho)t} - \frac{\varphi\rho}{w-\rho}$. 要使 $\lim_{t \rightarrow \infty} x(t)$ 收敛, 则 $\lim_{t \rightarrow \infty} \left| \frac{x(t+1)}{x(t)} \right| < 1$, 解得 $0 < w <$

φ , 所以式 (14) 收敛于 $\lim_{t \rightarrow \infty} x(t) = \frac{\varphi\rho}{\varphi - w}$.

因为 r_1 和 r_2 服从均匀分布, 所以收敛策略的平均行为可通过其期望值进行观察, 即

$$E(c_1r_1) = c_1 \int_0^1 \frac{x}{1-0} dx = \frac{c_1}{2},$$

$$E(c_2r_2) = c_2 \int_0^1 \frac{x}{1-0} dx = \frac{c_2}{2},$$

所以

$$\lim_{t \rightarrow \infty} x(t) = p^* = \frac{\varphi\rho}{\varphi - w} = \frac{c_1r_1p^* + c_2r_2gb}{c_1r_1 + c_2r_2 - w} = \frac{c_1p^* + c_2gb}{c_1 + c_2 - 2w}.$$

因此, 当

$$0 < w < \varphi = c_1r_1 + c_2r_2 = \frac{c_1 + c_2}{2}$$

时, 收敛策略 1 收敛于 $\frac{c_1p^* + c_2gb}{c_1r_1 + c_2r_2 - 2w}$. \square

Berghf 博士^[15]对随机优化算法给出了其全局收敛必须满足的条件. 首先给出以下两个假设:

假设 1 若 $f(D(z, \varepsilon)) \leq f(z), \varepsilon \in S$, 则

$$f(D(z, \varepsilon)) \leq f(\varepsilon).$$

其中: D 为产生问题解的函数; ε 为从概率空间 (R_n, B, U_k) 产生的随机向量; f 为目标函数; S 为 R_n 的子集, 表示问题的约束空间; U_k 为 B 上的概率度量, B 为子集的 σ 域.

假设 2 若对 S 的 Borel 子集 A , 有 $V(A) > 0$, 则

$$\prod_0^{\infty} (1 - u_k(A)) = 0.$$

其中: $V(A)$ 为子集 A 的 n 维闭包, $U_k(A)$ 为测度 U_k 产生的概率.

定理 2 设 f 为一个可测函数, S 为 R^n 的一个可测集, $\{Z_k\}_{k=1}^{+\infty}$ 为随机算法动态方程产生的解序列, 则当满足假设 1 和假设 2 时, 该策略保证以概率 1 收敛于全局最优解, 即

$$\lim_{t \rightarrow \infty} p[Z_k \in R_\varepsilon] = 1.$$

其中: R_ε 为全局最优点集合, $p[Z_k \in R_\varepsilon]$ 为第 k 代生成解 $Z_k \in R_\varepsilon$ 的概率.

下面通过验证扰动策略是否满足假设 1 和假设 2 来证明该策略的收敛性. 根据式 (5) 和 (2) 可得到如下位置更新公式:

$$x(t+1) = x(t) + wv(t) - x(t)(c_3\varphi_1 + c_4\varphi_2) + \varphi_1pb + \varphi_2gb, \quad (15)$$

其中: $\varphi_1 = r_1c_1, \varphi_2 = r_2c_2$. 整理得

$$x(t+1) = x(t) + (1 + w - c_3\varphi_1 - c_4\varphi_2) - wx(t-1) + \varphi_1pb + \varphi_2gb. \quad (16)$$

算法运行到第 k 代时, 第 i 个粒子的样本空间支撑集为

$$C_{i,k} = (1 + w - c_3\varphi_1 - c_4\varphi_2)x_{i,j,k-1} - wx_{i,j,k-2} + \varphi_1pb + \varphi_2gb. \quad (17)$$

化简得

$$C_{i,k} = x_{i,j,k-1} + w(c_3x_{i,j,k-1} - c_4x_{i,j,k-1}) - \varphi_1(pb - c_3x_{i,j,k-1}) + \varphi_2(gb - c_4x_{i,j,k-1}). \quad (18)$$

其中: $0 \leq \varphi_1 \leq c_1, 0 \leq \varphi_2 \leq c_2; 0 \leq c_3 \leq 1, 0 \leq c_4 \leq 1; x_{i,j,k}$ 为第 i 个粒子进化到第 k 代时第 j 维分量的长度, 当 $\max(c_1|pb - c_3x_{i,j,k}|, c_2|gb - c_4x_{i,j,k}|) < 0.5d_j(S)$ 时, 有 $V[C_{i,j} \cap S] < V(S), d_j(S)$ 为子集的长度. 由此可得: 随着每次迭代的进行, 搜索空间在逐渐减少, 满足假设 1; 该策略在搜索停滞时通过停滞干扰系数、扰动局部极值和全局极值、分散粒子, 使粒子搜索空间非零, 满足假设 2.

综上所述, 扰动策略满足 2 个假设, 能够有效地以概率 1 收敛于全局最优解. 同理可以证明探索策略以及开发策略的收敛性, 限于篇幅, 这里不再赘述.

2.5 算法平均计算时间复杂度理论分析

下面针对本文提出的 HLPSO 算法和传统 PSO 算法的平均计算时间复杂度进行理论分析和比较. 根据以上对基本 PSO 算法以及 HLPSO 算法的描述, 对于基本的 PSO 算法, 假设粒子的数量为 N , 迭代总次数为 \maxgen , 每个粒子每一次迭代需要的运算时间为 $T_1 = T_2 = \dots = T_n = T$, 则可以得出基本 PSO 算法进行优化所需要的总的运行时间为 $N \times \maxgen \times T$. 对于 HLPSO 算法, 随着自适应策略的运行, HLPSO 算法进行优化后所需要的总的运行时间为 $\sum_{i=1}^{\maxgen} \sum_{j=1}^N T_i$.

由上述分析可知, 两种算法复杂度的差别主要体现在每一次迭代中粒子选用更新策略和每个粒子一次迭代所需要的运行时间上, 与基本 PSO 在同一个数量级, 是比较理想的状态. HLPSO 算法以较少的算法复杂度为代价换取了精度的大幅提高是值得的, 符合算法改进的基本原则.

3 实验与分析

为了说明所提出算法的有效性, 本实验选取 7 个 Benchmark 函数, 其中 f_1 和 f_2 为单峰函数, $f_3 \sim f_7$ 为多峰函数, 7 个测试函数的特性值如表 1 所示.

将 HLPSO 算法与文献 [7] 中的 CLPSO、文献 [10] 中的 FIPS、文献 [9] 中的 TSPSO 算法进行实验比较, 各算法的参数设置见表 2. 仿真中, 算法用 Matlab 7.0 编程工具实现, 所有算法针对每个函数独立运算 30 次, 每个函数测试维数为 30, 种群大小为 40, 最大迭

表 1 测试函数的名称、最优值、搜索空间和初始值设置

func	function name	X	$f(X)$	search range	initialization range
f_1	Sphere	$[0, 0, \dots, 0]$	0	$[-100, 100]$	$[-100, 50]$
f_2	Rosenbrock	$[1, 1, \dots, 1]$	0	$[-2.048, 2.048]$	$[-2.048, 2.048]$
f_3	Ackley	$[0, 0, \dots, 0]$	0	$[-32.768, 32.768]$	$[-32.768, 16]$
f_4	Griewank	$[0, 0, \dots, 0]$	0	$[-600, 600]$	$[-600, 200]$
f_5	Rastrigrin	$[0, 0, \dots, 0]$	0	$[-5.12, 5.12]$	$[-5.12, 2]$
f_6	Noncontiguous Restraining	$[0, 0, \dots, 0]$	0	$[-5.12, 5.12]$	$[-5.12, 2]$
f_7	Weierstrass	$[0, 0, \dots, 0]$	0	$[-0.5, 0.5]$	$[-0.5, 0.2]$

代次数为 5 000, 得到优化结果平均值、方差作为算法的衡量标准, 测试结果见表 3. 图 1~图 7 是以上几种算法优化 7 个 Benchmark 测试函数的性能比较.

表 2 各个算法参数设置

算法	参数设置
FIPS	$w = 0.7298, c_1 = c_2 = 2$
CLPSO	$w = [0.4, 0.9], c = 2.0$
TSPSO	$w = 0.8, c_1 = c_2 = 2$
HLPSO	$w = 0.9 - T_{\text{current}} \times 0.7 / \text{maxgen}$

由表 3 数据对比可以看出, 在大部分标准测试函数中, 无论是解的质量还是算法的收敛精度和稳定性, 在这 4 种算法中 HLPSO 算法在单峰函数和多峰函数上都取得了最好的优化结果. 从图 1~图 7 函数收敛图可以看出, 在单峰问题上, HLPSO 和 TSPSO 在精度方面都具有良好的表现, 这两种算法都引入了扰动策略, 但 HLPSO 在收敛速度上优于 TSPSO. 这是因为 HLPSO 经过短暂的迭代自适应调整可以同时获得深度和广度信息. CLPSO 对复杂的多峰问题非常有效,

由于其多样性保持机制过于强壮, 导致 CLPSO 在单峰优化问题上的寻优效率较低. 值得注意的是, 在公认的非常难解决的函数 Rosenbrock 上, HLPSO 可以经过一段时间的进化后顺利找到位于非常狭长峰谷的全局最优, 而其余 PSO 优化算法则在离全局最优较远的地方停止搜索. 这有力地验证了深度搜索速度更新机制在 HLPSO 所发挥的作用能够满足其设计期望. 对于难以优化的高维函数, 尤其是有多个局部极小点的多峰值函数, 其他算法寻优时很容易陷入局部极值. 粒子一旦陷入局部最优, 则很难从局部最优跳出, 因此难以取得理想的效果. HLPSO 算法通过自适应调整策略增强了种群多样性, 使得粒子更容易跳出局部极值, 在更广泛的搜索范围内进一步寻找更好的解, 从而获得比其他算法更好的优化效果. 结果表明, HLPSO 算法具有很好的跳出局部最优和快速收敛到最优解的能力, 无论在单峰还是多峰函数上都表现得非常优秀, 因此可以得出 HLPSO 具有很好的自适应性的结论.

表 3 HLPSO 和其他 PSO 测试 30 维 Benchmark 函数效率实验结果比较

func	MAPSO mean(Std Dev)	MAPSO mean(Std Dev)	MAPSO mean(Std Dev)	MAPSO mean(Std Dev)
f_1	6.03 e-307 (0.00 e+000)	5.15 e-019 (2.77 e-010)	2.78 e-012 (5.87 e-013)	3.22 e-012 (1.79 e-012)
f_2	2.46 e-002 (1.35 e-001)	1.46 e+001 (3.62 e+000)	2.46 e+001 (2.19 e-001)	2.46 e+001 (3.22 e-001)
f_3	7.85 e-016 (0.00 e+000)	3.55 e-010 (1.00 e-010)	4.81 e-007 (9.17 e-008)	4.66 e-004 (1.43 e-007)
f_4	0.00 e+000 (0.00 e+000)	7.16 e-012 (1.59 e-011)	1.16 e-006 (1.87 e-006)	1.87 e-005 (6.39 e-005)
f_5	0.00 e+000 (1.00 e+000)	3.26 e-009 (2.84 e-009)	7.43 e+001 (1.24 e+001)	7.48 e+000 (1.37 e+000)
f_6	0.00 e+000 (1.00 e+000)	5.98 e-009 (6.78 e-009)	6.08 e+001 (8.35 e+000)	6.89 e+001 (1.26 e+001)
f_7	0.00 e+000 (1.00 e+000)	4.36 e-012 (2.02 e-012)	1.14 e-001 (1.48 e-001)	2.18 e-001 (2.95 e-001)

在这组对比实验中产生了大量实验数据, 为分析对比算法的性能带来许多困难, 所以采用双侧 T-检验方法对所得实验数据进行统计分析和量化, 以便较客观地评价 HLPSO 算法与对比算法之间的性能差异程度. 采用 T-检验方法对 HLPSO 与几种变异 PSO 算法的差异显著性检验结果如表 4 所示. 设显著性水平 $\alpha = 0.05$, 因为每个测试重复执行 30 次, 所以自由度 $d_f = 29$, 查表可得 $t_{0.05}(29) = 2.045$. 由 $t = \bar{d} / S_{\bar{d}}$ 求 t 值, 其中 \bar{d} 为配对检验数据差数平均值, $S_{\bar{d}}$ 为差数平均的标准差, t 值计算结果如表 4 所示. 当 $|t| < t_{0.05}(29)$ 时, 认为 HLPSO 算法与相应变异 PSO 算法无显著差异, 记为 0; 否则认为差异显著, 记为 1. 最后一行数值是 7

个标准测试函数中优化结果有显著差异的函数个数. 由表 4 可见, 与 CLPSO、TSPSO、FIPSO 相比, HLPSO 算法对大多数测试函数的优化性能差异显著.

表 4 HLPSO 与几种 PSO 算法的差异显著性检验

func	TSPSO	CLPSO	FIPSO
f_1	-1.559	10.903	-1.484
f_2	-3.231	0.737	-12.135
f_3	-6.704	-9.445	-16.48
f_4	-14.046	0.728	-1.517
f_5	-23.530	5.956	-8.826
f_6	-7.903	0.620	-7.912
f_7	-1.406	-1.935	-1.071
显著差异个数	5	3	4

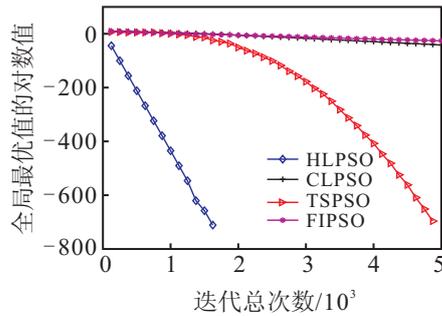


图 1 4种不同算法的 Sphere 优化函数收敛曲线

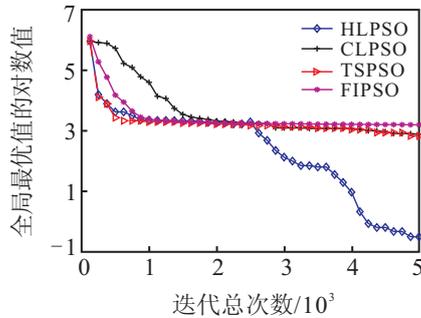


图 2 4种不同算法的 Rosenbrock 优化函数收敛曲线

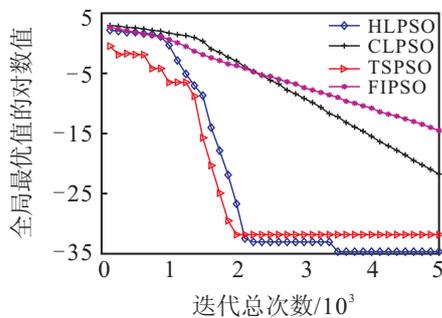


图 3 4种不同算法的 Ackley 优化函数收敛曲线

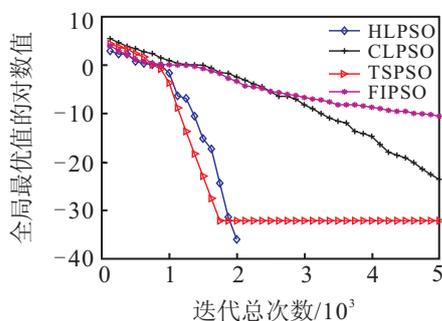


图 4 4种不同算法的 Griewank 优化函数收敛曲线

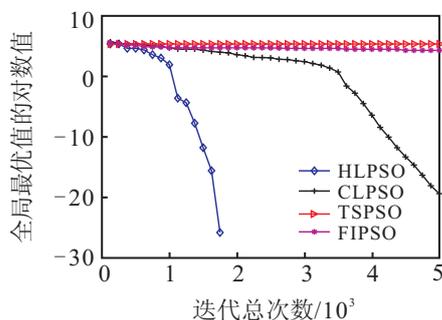


图 5 4种不同算法的 Rastrigrin 优化函数收敛曲线

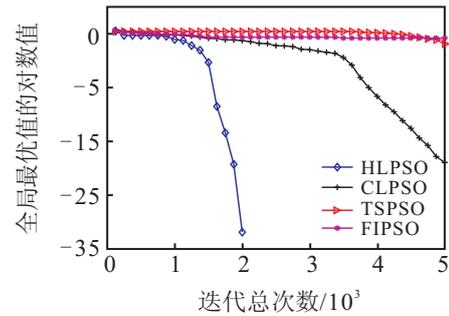


图 6 4种不同算法的 Noncontiguous Restrining 优化函数收敛曲线

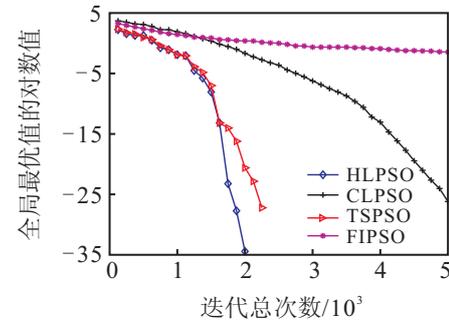


图 7 4种不同算法的 Weierstrass 优化函数收敛曲线

4 结 论

本文针对目前很多 PSO 算法是面向特定问题而设计的,其鲁棒性和普适性均不强等问题,提出了一种基于混合策略自适应学习的并行粒子群优化算法 (HLPSO).为了平衡算法的探索和开发能力,改进的算法引入了 4 种并行变异策略,从而既克服了早熟现象,又提高了算法的收敛速度.仿真实验结果表明,所提出的 HLPSO 算法显著提高了粒子群优化算法的性能,在优化效率、优化性能和鲁棒性方面均有较大的改善,并具有较强的普适性.

参考文献(References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization[C]. Int Conf on Neural Networks. Perth: IEEE, 1995: 1942-1948.
- [2] Scriven I, Junwei L, Lewis A. Electromagnetic noise source approximation for finite-difference time-domain modeling using near-field scanning and particle swarm optimization[J]. IEEE Trans on Electro-magnetic Compatibility, 2010, 52(1): 89-97.
- [3] Wei-Neng Chen, Jun Zhang, Chung H S-H. A novel set-based particle swarm optimization method for discrete optimization problems[J]. IEEE Trans on Evolutionary Computation, 2010, 14(2): 278-300.
- [4] Rong-Jong Wai, Jeng-Dao Lee. Real-time PID control strategy for maglev transportation system via particle swarm optimization[J]. IEEE Trans on Industrial Electronics, 2010, 58(2): 629-646.
- [5] 迟玉红, 孙富春, 王维军, 等. 基于空间缩放和吸引子的粒子群优化算法[J]. 计算机学报, 2011, 34(1): 115-130.

- (Chi Y H, Sun F C, Wang W J, et al. An improved particle swarm optimization algorithm with search space zoomed factor and attractor[J]. Chinese J of Computers, 2011, 34(1): 115-130.)
- [6] 张顶学, 廖锐全. 一种基于种群速度的自适应粒子群算法[J]. 控制与决策, 2009, 23(7): 756-761.
(Zhang D X, Liao R Q. Adaptive particle swarm optimization algorithm based on population velocity[J]. Control and Decision, 2009, 23(7): 756-761.)
- [7] 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.
- [8] Changhe Li, Shengxiang Yang. An adaptive learning particle swarm optimizer for function optimization[C]. IEEE Congress on Evolutionary Computation. Trondheim, 2009: 381-388.
- [9] 胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报, 2007, 18(4): 861-868.
(Hu W, Li Z S. A simpler and more effective particle swarm optimization algorithm[J]. J of Software, 2007, 18(4): 861-868.)
- [10] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 204-210.
- [11] 贾树晋, 杜斌, 岳恒. 基于局部搜索与混合多样性策略的多目标粒子群算法[J]. 控制与决策, 2012, 27(6): 813-818
(Jia S J, Du B, Yue H. Local search and hybrid diversity strategy based multi-objective particle swarm optimization algorithm[J]. Control and Decision, 2012, 27(6): 813-818.)
- [12] Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. IEEE Trans on Evolutionary Computation, 2009, 13(2): 398-417.
- [13] Jasper A, Vrugt Bruce A Robinson, James M Hyman. Self-adaptive multimethod search for global optimization in real-parameterspaces[J]. IEEE Trans on Evolutionary Computation, 2009, 13(2): 243-259.
- [14] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization[C]. Proc of the ICEC. Washington, 1999: 1951-1957.
- [15] Bergh van den F. An analysis of particle swarm optimisers[D]. Pretoria: Department of Computer Science, University of Pretoria, 2002.

(上接第1086页)

- [4] Meng Q, Huang Y, Cheu R L. Competitive facility location on decentralized supply chains[J]. European J of Operational Research, 2009, 196(2): 487-499.
- [5] Redondo J L, Fernandez J, Garcia I, et al. Parallel algorithms for continuous multifacility competitive location problems[J]. J of Global Optimization, 2011, 50(4): 557-573.
- [6] Kucukaydin H, Aras N, Altinel I k. Competitive facility location problem with attractiveness adjustment of the follower: A bilevel programming model and its solution[J]. European J of Operational Research, 2011, 208 (3): 206-220.
- [7] 范建华. 基于阶梯型衰退效用函数的竞争选址问题[J]. 管理学报, 2009, 6(12): 1638-1642.
(Fan J H. The choice of competitive location for the chain-firm on the basis of step decay utility function[J]. Chinese J of Management, 2009, 6(12): 1638-1642.)
- [8] 张曦, 杨超, 胡丹丹. 考虑市场扩张和吞并的新设施竞争选址模型[J]. 工业工程与管理, 2009, 14(2): 43-47.
(Zhang X, Yang C, Hu D D. The competitive location problem with market expansion and cannibalization[J]. Industrial Engineering and Management, 2009, 14(2): 43-47.)
- [9] Huff D L. Defining and estimating a trading area[J]. J of Marketing, 1964, 28(3): 34-38.
- [10] Benati S. Categorical data fuzzy clustering: an analysis of local search heuristics[J]. Computer & Operation Research, 2008, 135(6): 766-775.
- [11] Nakanishi M, Cooper L G. Parameter estimation for a multiplicative competitive interaction model: Least squares approach[J]. J of Marketing Research, 1974, 11(3): 303-311.
- [12] Zou D X, Gao L Q, Wu J H, et al. A novel global harmony search algorithm for reliability problems[J]. Computers & Industrial Engineering, 2010, 58(2): 307-316.
- [13] 李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法[J]. 系统工程理论与实践, 2002, 22(11): 32-38.
(Li X L, Shao Z J, Qian J X. An optimizing method based on autonomous animats: Fish-swarm algorithm[J]. Systems Engineering-Theory & Practice, 2002, 22(11): 32-38.)
- [14] Zou D X, Gao L Q, Li S, et al. Solving 0-1 knapsack problem by a novel global harmony search algorithm[J]. Applied Soft Computing, 2011, 11(2): 1556-1564.