

文章编号:1001-9081(2010)06-1516-03

改进的 PSO 混合算法

杨恢先¹, 刘子文², 汪俊², 王绪四¹, 谢鹏鹤¹

(1. 湘潭大学 材料与光电物理学院, 湖南 湘潭 411105; 2. 湘潭大学 信息工程学院, 湖南 湘潭 411105)

(yanghx@xtu.edu.cn)

摘要:为了提高粒子群算法的寻优速度和寻优精度,提出一种改进的 PSO 混合算法。在差分进化(DE)算法中引入了动态比例因子,在 PSO 算法中引入 DE 算法的变异、交叉操作,重新构造 PSO 算法的粒子位置更新公式。选取了 4 个基准函数进行测试,并与其它 PSO 混合算法作了比较。仿真结果表明该方法是有效的。

关键词:粒子群算法;差分进化算法;变异;交叉

中图分类号: TP183 **文献标志码:**A

Modified PSO hybrid algorithm

YANG Hui-xian¹, LIU Zi-wen², WANG Jun², WANG Xu-si¹, XIE Peng-he¹

(1. Faculty of Materials, Optoelectronics and Physics, Xiangtan University, Xiangtan Hunan 411105, China;

2. College of Information Engineering, Xiangtan University, Xiangtan Hunan 411105, China)

Abstract: This paper proposed a novel Particle Swarm Optimization (PSO) hybrid algorithm to improve the optimum speed and performance of the PSO algorithm. This new algorithm introduced a dynamic proportion operator into differential evolution algorithm and also introduced mutation, crossover operator from DE algorithm into PSO algorithm. Then the position updating formula of PSO was reconstructed. At last, this paper chose four reference functions to have a test and compared the results with other PSO hybrid algorithms. The simulation results verify the effectiveness of this approach.

Key words: Particle Swarm Optimization (PSO) algorithm; Differential Evolution (DE) algorithm; mutation; crossover

0 引言

粒子群(Particle Swarm Optimization, PSO)^[1]算法是一种随机搜索策略的优化算法,该算法模拟鸟群的飞行行为而获得多维寻优能力,同传统的优化算法如遗传算法相比,该算法具有实现简单、收敛速度快的优点。文献[2]提出的差分进化(Differential Evolution, DE)算法是一种基于群体进化的算法,具有记忆个体最优解和种群内信息共享的特点。近年来,为了解决粒子群算法在收敛后期容易陷入局部最优,从而导致“早熟”^[3]问题,许多学者结合其他的算法与粒子群算法进行融合,试图在粒子群算法原有的收敛速度快、易于实现的基础上,更加精确地收敛到全局最优值。如文献[4]提出的一种高效的遗传粒子混合算法(GA-PSO),使得算法的操作既简单,又效率高。文献[5]提出的一种广义蚁群与粒子群结合算法(GACO-PSO),使得算法可实现非凸空间上的高效搜索。

本文在 DE 中引入了动态比例因子,在 PSO 算法中引入 DE 算法的变异、交叉操作,重新构造 PSO 算法的粒子位置更新公式,提出一种改进的粒子群混合算法——DE-PSO 算法。

1 粒子群算法

粒子群算法数学描述如下:

设搜索空间为 m 维,总粒子数为 n ,第 i 个粒子的位置表示为向量 $\mathbf{X}_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{ij}, \dots, \mathbf{x}_{in})$, 第 i 个粒子的速度表示为向量 $\mathbf{V}_i = (\mathbf{v}_{i1}, \mathbf{v}_{i2}, \dots, \mathbf{v}_{ij}, \dots, \mathbf{v}_{in})$, p_i 为第 i 个粒子迄今为止搜索到的最优位置, p_{ij} 是 p_i 的第 j 维元素; p_j^* 为整个粒子

群迄今为止搜索到的最优位置。每个粒子的速度和位置更新公式为:

$$\mathbf{v}_{ij}(t+1) = w \times \mathbf{v}_{ij}(t) + c_1 \times \zeta \times (p_{ij}(t) - \mathbf{x}_{ij}(t)) + c_2 \times \eta \times (p_j^*(t) - \mathbf{x}_{ij}(t)) \quad (1)$$

$$\mathbf{x}_{ij}(t+1) = \mathbf{x}_{ij}(t) + \mathbf{v}_{ij}(t+1) \quad (2)$$

$$w = w_{\max} - iter \times \frac{w_{\max} - w_{\min}}{iter_{\max}} \quad (3)$$

其中: $1 \leq i \leq n, 1 \leq j \leq m$; $\mathbf{v}_{ij}(t)$ 、 $\mathbf{v}_{ij}(t+1)$ 分别表示第 i 个粒子在 j 维方向上的当前速度和更新后的速度; $\mathbf{x}_{ij}(t)$ 、 $\mathbf{x}_{ij}(t+1)$ 分别表示第 i 个粒子在 j 维方向上的当前位置和更新后的位置; c_1 、 c_2 为常量, 称为加速因子; ζ 、 η 为 $[0, 1]$ 的随机数; w 为惯性权重, w_{\max} 、 w_{\min} 分别是 w 的最大值和最小值; $iter$ 、 $iter_{\max}$ 分别为当前迭代次数和最大迭代次数。

2 混合 DE-PSO 算法

2.1 差分算法

变异算子 对每个在 t 时刻的个体 $\mathbf{x}_i(t)$ 实施变异操作得到与其相对应的变异个体 $\mathbf{v}_i(t)$, 即:

$$\mathbf{v}_i(t) = \mathbf{x}_{i_3}(t) + F_i(t) \times (\mathbf{x}_{i_1}(t) - \mathbf{x}_{i_2}(t)) \quad (4)$$

$$F_i(t) = F_{i_4}(t) + N(0, 1) \times (F_{i_5}(t) - F_{i_6}(t)); \\ i_4 \neq i_5 \neq i_6 \quad (5)$$

其中: $\mathbf{x}_i(t)$ 为父代基向量, $\mathbf{x}_{i_2}(t) - \mathbf{x}_{i_1}(t)$ 称为父代差分向量, $F_i(t)$ 为动态比例因子。

交叉算子 对 $\mathbf{x}_i(t)$ 和由式(4)生成的变异个体 $\mathbf{v}_i(t)$ 实施交叉操作,生成实验个体 $\mathbf{u}_{ij}(t)$, 即:

收稿日期:2009-12-23;修回日期:2010-03-04。 基金项目:海南省自然科学基金资助项目(60897)。

作者简介:杨恢先(1963-),男,湖南益阳人,教授,主要研究方向:智能控制、智能信息处理、数字控制; 刘子文(1983-),男,湖南新化人,硕士研究生,主要研究方向:控制系统、CAD。

$$\mathbf{u}_{ij}(t) = \begin{cases} \mathbf{v}_{ij}(t), & \text{rand}(j) < C_r \text{ 或 } j = r \\ \mathbf{x}_{ij}(t), & \text{其他} \end{cases} \quad (6)$$

其中:rand(j) 为(0,1) 的均匀分布随机数,C_r 为范围在(0,1) 的交叉概率,r 为[1,m] 的随机数。

选择算子 对实验个体 $\mathbf{u}_{ij}(t)$ 和 $\mathbf{x}_{ij}(t)$ 的目标函数进行比较,对于最小化问题则选择目标函数低的个体作为新种群的个体 $\mathbf{x}_{ij}(t+1)$,即:

$$\mathbf{x}_{ij}(t+1) = \begin{cases} \mathbf{u}_{ij}(t), & f(\mathbf{u}_{ij}(t)) \leq f(\mathbf{x}_{ij}(t)) \\ \mathbf{x}_{ij}(t), & \text{其他} \end{cases} \quad (7)$$

其中f为目标函数。

2.2 混合 DE-PSO 算法

根据 DE 算法的策略,重新构造 PSO 算法的粒子位置更新公式为:

$$\mathbf{x}_{ij}(t) = \begin{cases} r_{ij}(t) + F_1(t) \times (\mathbf{x}_{i,j}(t) - \mathbf{x}_{i,j}(t)), & U(0,1) > p_r \\ p_{i,j}(t), & \text{其他} \end{cases} \quad (8)$$

$$r_{ij}(t) = r_2 \times p_{ij}(t) + (1 - r_2) \times p_{ij}^*(t)$$

其中: $i_1, i_2, i_3 \in (0, n)$; $F_1(t)$ 由式(5) 进行计算; $r_2 \sim U(0, 1)$; p_r 是重组概率; $p_{i,j}(t)$ 从 p_{ij} 中随机选择。

算法步骤如下。

第1步 给定粒子的群体规模n,随机初始化全体粒子的速度和位置。

第2步 对于种群中的每一个粒子i,用式(8) 更新位置 $\mathbf{x}_i(t)$;并且计算适应度值:设 f_i 为 $\mathbf{x}_i(t)$ 的适应度值。

第3步 更新 $p_i(t)$ 。如果 f_i 比 $p_i(t)$ 的适应度值更优,将 $p_i(t)$ 用 $\mathbf{x}_i(t)$ 替换。

第4步 更新 $p_i^*(t)$ 。设 $Xbest(t)$ 为当前群里最优位置,如果 $Xbest(t)$ 比 $p_i^*(t)$ 更优,将 $p_i^*(t)$ 用 $Xbest(t)$ 替换。

第5步 对于种群中的每一个粒子i,用式(1)~(2) 分别更新速度 $\mathbf{v}_i(t+1)$ 和位置 $\mathbf{x}_i(t+1)$ 。

第6步 当 $iter < iter_{max}$ 时,跳转回第2步;否则,算法结束。

3 实验和分析

通过4个典型的测试函数对 DE-PSO 进行测试,并选用已经提出的 GA-PSO 和 GACO-PSO 算法作为对比算法与 DE-PSO 进行比较。4个优化函数如表1所示。其中函数 f_1 为简单的单峰函数,在原点达到极小值; f_2 为非凸的病态函数,在 $\mathbf{x}_i = 1$ 时达到极小值; f_3 在 $\mathbf{x}_i = 0$ 时达到全局极小值,当 $\mathbf{x}_i = \pm k\pi\sqrt{i}$ ($i = 1, 2, \dots, n; k = 1, 2, \dots, n$) 时,达到局部极小值; f_4 为多峰函数,存在多个局部最优点。

3种算法的优化目标是函数值最小化,针对每个函数运行30次求平均值,仿真结果与统计数据见表2。计算结果显示 DE-PSO 算法无论在最小值还是平均值上都比 GA-PSO、GACO-PSO 算法的性能有所提高。图1给出了4个测试函数维数为30时,分别采用3种算法寻优得到的函数曲线。实验中,3种算法的种群大小均为30,最大迭代次数为2000。DE-PSO 算法中, $p_r = 0.6$, $w_{max} = 0.9$, $w_{min} = 0.4$, $c_1 = c_2 = 1.49$; GA-PSO、GACO-PSO 算法中的参数均按原文献选取。由图1可以看出,3种算法在所有函数优化问题中,都具有较快的全局收敛速度、较高的收敛精度,GA-PSO、GACO-PSO 两种算法在规定代数内均能收敛,但 DE-PSO 算法的收敛值更优。虽然在 Griewank、Ackley 函数方阵中,GACO-PSO 算法的收敛速度较 DE-PSO 算法快,但是 DE-PSO 算法的收敛精度要高于 GA-PSO、GACO-PSO 算法。

表1 测试函数的维数、初值范围和目标值

函数表达式	函数名	维数	初值范围	目标值
$f_1 = \sum_{i=1}^n (\mathbf{x}_i^2)$	Sphere	30	$[-10, 10]^{30}$	0
$f_2 = \frac{1}{4000} \sum_{i=1}^n \mathbf{x}_i^2 - \prod_{i=1}^n \cos\left(\frac{\mathbf{x}_i}{\sqrt{i}}\right) + 1$	Griewank	30	$[-60, 60]^{30}$	0
$f_3 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi \mathbf{x}_i)\right) + 20 + e$	Ackley	30	$[-32, 32]^{30}$	0
$f_4 = \sum_{i=1}^n \left[100(\mathbf{x}_{i+1} - \mathbf{x}_i)^2 + (\mathbf{x}_i - 1)^2 \right]$	Rosenbrock	30	$[-10, 10]^{30}$	0

表2 3种算法的性能比较

算法	f_1			f_2			f_3			f_4		
	最小值	平均值	成功率/%									
DE-PSO	0.012	0.018	100	0.006	0.008	100	0.010	0.013	100	1.129	1.228	100
GACO-PSO	0.107	0.164	98	0.015	0.021	100	0.031	0.036	100	6.781	6.981	100
CA-PSO	0.098	0.106	100	0.009	0.011	100	0.017	0.023	100	4.291	4.612	100

4 结语

为了提高粒子群优化算法的搜索精度和搜索速度,提出了一种新型的 PSO 与 DE 的混合算法。一方面,本文将在基

本差分进化算法中引入了自适应变异算子,另一方面,在 PSO 算法中引入 DE 算法的变异、交叉操作,重新构造 PSO 算法的粒子位置更新公式。通过4个基准函数的测试,仿真结果表明了该算法具有很好的收敛速度和收敛精度。

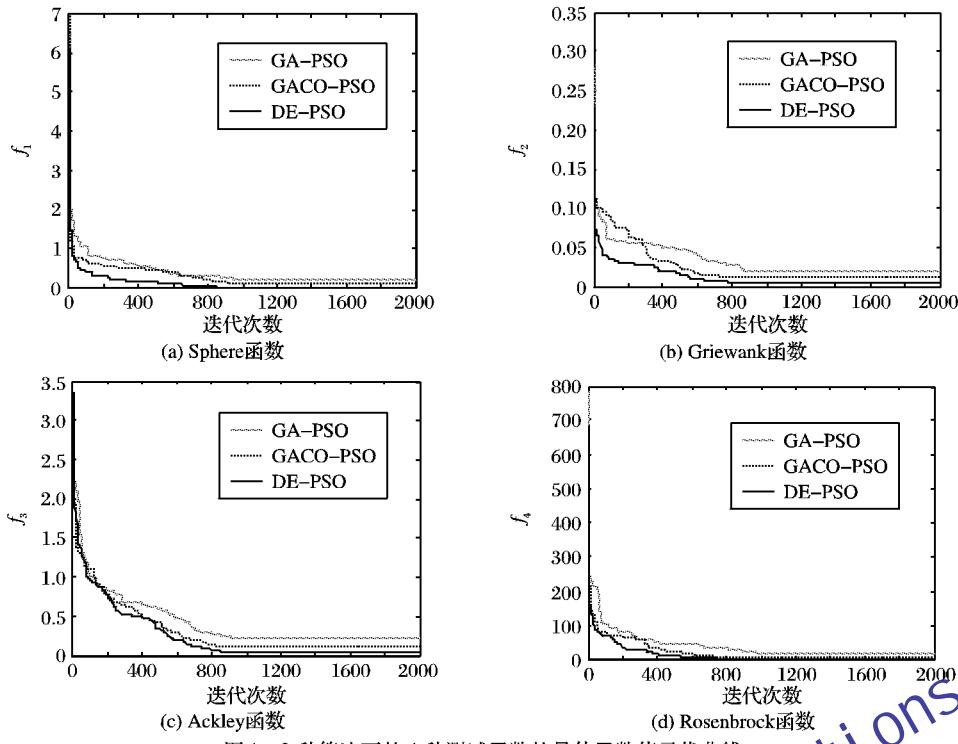


图 1 3 种算法下的 4 种测试函数的最佳函数值寻优曲线

参考文献:

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// Proceedings of the IEEE International Conference on Neural Networks. Washington, DC: IEEE, 1995: 1942–1948.
- [2] STORN R, PRICE K. Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces [EB/OL]. [2009-10-10]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.9696>.

- [3] ARUMUGAM M S, RAO M V C, CHANDRAMOHAN A. A new and improved version of particle swarm optimization algorithm with global-local best parameters [J]. Knowledge and Information Systems, 2008, 16(3): 324–350.
- [4] 彭晓波, 桂华卫, 黄志武, 等. GAPSO: 一种高效的遗传粒子混合算法及其应用[J]. 系统仿真学报, 2008, 20(18): 5025–5031.
- [5] 戈剑武, 祁荣宾, 钱锋, 等. 一种改进的自适应差分进化算法[J]. 华东理工大学学报: 自然科学版, 2009, 35(4): 600–605.

(上接第 1515 页)

2) 该算法从 M 组 BP 网络的初始权值种群出发, 利用蚂蚁算法的正反馈机制和分布式并行计算能力, 可使 BP 算法能在全局多点同时进行解的搜索, 避免了 BP 算法易陷入局部极小点的局限, 改善了 BP 算法的性能。

3) 该算法保留了 BP 算法概念清晰、易于理解, 计算简单的特点。

4) BP 网络的性能还要受到网络结构(主要是指隐节点数)、函数的复杂性和样本质量等多种因素的影响, 本文未考虑这些因素, 这些问题还有待进一步深入研究。

参考文献:

- [1] NIELSON H R. Theory of the back propagation neural networks [C]// Proceeding of the International Joint Conference on Neural Networks. Washington, DC: IEEE, 1989(1): 593–606.
- [2] CHERKASSAY V, SHEPHERD R. Regularization effect of weight initialization in back propagation networks [C]// The 1998 IEEE International Joint Conference on Neural Networks. Washington, DC: IEEE, 1998: 2258–2261.
- [3] 师彪, 李郁侠, 于新花, 等. 改进粒子群-BP 神经网络模型的短期电力负荷预测[J]. 计算机应用, 2009, 29(4): 1036–1039.
- [4] 陆琼瑜, 童学锋. BP 算法改进的研究[J]. 计算机工程与设计, 2007, 28(3): 648–650.
- [5] 洪露, 穆志纯. 基于免疫遗传算法的前向神经网络设计[J]. 计

算机工程, 2006, 32(15): 179–180, 183.

- [6] 潘昊, 侯清兰. 基于粒子群优化算法的 BP 网络学习研究[J]. 计算机工程与应用, 2006, 42(16): 41–43, 66.
- [7] 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.
- [8] 李祚泳, 汪嘉杨, 郭淳. PSO 算法优化 BP 网络的新方法及仿真实验[J]. 电子学报, 2008, 36(11): 2224–2228.
- [9] 王晶. 蚁群算法优化前向神经网络的一种方法[J]. 计算机工程与应用, 2006, 42(25): 53–55.
- [10] DORIGO M, MANIEZZO V, COLONI A. The ant system optimization by a colony of cooperating Agents [J]. IEEE Transactions on Systems, Man and Cybernetics, Part B, 1996, 26(1): 29–41.
- [11] DORIGO M, BONABEAU E, THERAULAZ G. Ant algorithms and stigmergy [J]. Future Generation Computer Systems, 2000, 16(8): 851–871.
- [12] WU QING-HONG, ZHANG JI-HUI, XU XIN-HE. An ant colony algorithm with mutation features[J]. Journal of Computer Research and Development, 1999, 36(10): 1240–1245.
- [13] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53–66.