

自适应双层粒子群优化算法

周志勇, 李荣雨

(南京工业大学 计算机科学与技术学院, 江苏 南京 211816)

摘 要: 针对粒子群优化算法(PSO)在解决复杂问题时存在着早熟收敛和进化后期收敛速度较慢的问题, 提出一种自适应双层粒子群优化算法(ADLPSO). 在每次种群迭代时, 利用双层粒子群中的记忆群体向全局最优位置靠近的特性, 通过改进的粒子群更新公式更新记忆群体. 同时为提高群体的多样性, 对惯性权重进行自适应调整, 并令自适应过程和双层粒子群的更新同步进行. 仿真结果表明 ADLPSO 算法能够快速得到更优解.

关键词: 粒子群优化; 双层粒子群; 自适应; 惯性权重

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2016)11-0010-04

Adaptive Double Layers Particle Swarm Optimization Algorithm

ZHOU Zhi-yong, LI Rong-yu

(College of computer science and technology, Nanjing Tech University, Nanjing 211816, China)

Abstract: The particle swarm optimization (PSO) has some shortcomings, such as premature convergence and low convergence speed in the late evolutionary. An improved algorithm is proposed which is Adaptive Double Layer Particle Swarm Optimization algorithm(ADLPSO). In each population iteration, the algorithm takes advantage of the characteristic of memory swarm to tent to the global optimal position on double layer particle swarm, memory swarm being updated by using an improved the memory particle swarm update formula. At the same time, in order to improving the diversity of the population, it uses an improved adaptive adjustment strategy to update inertia weight and makes the adaptive process and the update of the double layer particle swarm synchronization. The experiment results show that that the new algorithm is more fast to find better solution.

Key words: particle swarm optimization; double layers; adaptive adjustment; inertia weight

1 引言

粒子群优化算法(PSO, Particle Swarm Optimization)是一种基于对鸟群、鱼群捕食行为模拟的群体智能优化算法^[1], 因其具有简单、容易实现并且有效的特点, 它已经被成功应用到许多优化领域^[2]. 但是在解决复杂问题时 PSO 容易发生早熟收敛, 并且在进化后期收敛速度较慢^[3].

为提高算法的收敛精度, 文献^[4]指出惯性权重的大小会影响算法的收敛性能, 随后提出了随着迭代的进行线性地减小惯性权重的策略. 惯性权重从线性的改变到非线性的下降以及各种非线性

性下降策略^[4-6]确实改善了算法的寻优能力, 但无论哪种递减策略都会使群体在搜索后期丢失多样性, 这对于求解复杂问题时群体很容易发生早熟收敛. 为解决多样性丢失问题, 又提出了一种对惯性权重的自适应操作^[7-8]. 各种自适应都不同程度地解决了全局搜索和局部寻优之间的转化问题. 文献^[7]提出当发生早熟收敛时, 根据群体早熟收敛程度和个体适应度值自适应地调整粒子的惯性权重, 使得粒子跳出局部最优以及强化优秀粒子的局部寻优能力. 但该算法并没有充分发挥出粒子群的多样性特点以避免早熟收敛的发生并且收敛速度较慢. 在算法的收敛速度研究中, 粒子群体

被分为当前和记忆两个群体,不同群体具有各自不同的特性,其中记忆群体有着向全局最优位置靠近的趋势^[9-11],并且记忆群体的更新对当前群体多样性影响很小,利用记忆群体的特性能够提高算法的收敛速度.然而使用记忆粒子群加速收敛必然会使算法在一定程度上过早收敛.

为了平衡好算法的收敛速度与全局收敛性,本文提出一种自适应双层粒子群优化算法(ADLPSO).

2 基本 PSO 算法

算法首先随机初始化粒子群体,粒子群中的每个粒子 P_i 都代表着待优化问题的潜在解.每个粒子都有自己的状态:速度 $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ 和位置 $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$.其中 d 表示维度(待解问题的未知数个数). X_i 对应的目标函数值即可作为该粒子的适应度,算法通过适应度来衡量粒子的优劣.算法在每一次迭代中每个粒子都记忆着 2 个位置,一个是 P_i 自身目前找到的最好位置 P_{besti} ,另一个是整个群体目前找到的最好位置 G_{best} .每个粒子通过跟踪这 2 个位置来更新自己的位置和速度.在下次迭代($t+1$)中,粒子 P_i 的速度和位置更新公式如式(1)、(2)所示.

$$V_{i,d}(t+1) = \omega * V_{i,d}(t) + c_1 * r_1 * (p_{besti,d}(t) - X_{i,d}(t)) + c_2 * r_2 * (G_{best,d}(t) - X_{i,d}(t)) \quad (1)$$

$$V_{i,d}(t+1) = X_{i,d}(t) + V_{i,d}(t+1) \quad (2)$$

式中, c_1, c_2 是加速因子. r_1, r_2 是两个在 $(0, 1)$ 之间的随机数.

PSO 算法迭代终止条件为预先确定的最大迭代次数或者是对优化结果的精度要求.算法终止时的全局极值(G_{best})即为最后的最优解.

3 自适应双层粒子群优化算法

3.1 双层粒子群

对于粒子群算法的优化大多都是一种针对当前群体进化的改进,很少有人通过研究粒子群的记忆群体来改善粒子群算法优化的性能.然而最新研究显示当前群体和记忆群体在搜索过程中展现了不同的特性^[8-11].具体来说,当前粒子群趋向于探索搜索空间,找到最优解大致位置,记忆粒子群趋向于开发最优解出来,利用好双层粒子群的特性可以使算法得到更好的优化效果.

对于当前群体的更新,每个粒子通过跟踪 2 个极值来更新自己的位置.式(1)、(2)可以等价式(3).

$$X_{i,d}(t+1) = X_{i,d}(t) + \omega * V_{i,d}(t) + c_1 * r_1 * (p_{besti,d}(t) - X_{i,d}(t)) + c_2 * r_2 * (G_{best,d}(t) - X_{i,d}(t)) \quad (3)$$

其实,PSO 可以被看成是基于一种特殊的拓扑结构的算法,而这种拓扑结构中的粒子包括个体极值位置的粒子和全局极值位置的粒子.按照上述所说,可以把式(3)简化为式(4).

$$X_{i,d}(t+1) = X_{i,d}(t) + \omega * V_{i,d}(t) + \sum_{P_k \in N_i} c_k r_k (P_{k,d}(t) - X_{i,d}(t)) \quad (4)$$

上式中, P_k 代表着要更新粒子的拓扑结构内的领域粒子. N_i 表示领域粒子的数量.

记忆粒子是当期粒子到目前为止找到的最好位置处的粒子.当前粒子的更新是通过追踪自身的记忆粒子和全局的记忆粒子来更新下一步的位置.而记忆粒子本身没有属于它的记忆粒子,它的更新只能通过全局的记忆粒子也就是全局极值来更新自己的位置.同时,记忆粒子的产生是当前粒子在更新过程中的精英策略的选择,记忆本身是没有速度的.所以,式(4)可以写成式(5).

$$X_{i,d}(t+1) = X_{i,d}(t) + c_i r_3 (G_{best,d}(t) - X_{i,d}(t)) \quad (5)$$

按照式(5)可以对记忆群体进行更新,但是更新后的位置需要进行优劣性评价.更新后位置的好坏直接影响着我们是否确认这一次的更新,所以这里需要一个中间变量存储记忆粒子更新的位置,用 $R_{i,d}^{temp}$ 替代 $X_{i,d}(t+1)$,式(5)变为式(6):

$$R_{i,d}^{temp} = X_{i,d}(t) + c_i r_3 (G_{best,d}(t) - X_{i,d}(t)) \quad (6)$$

如果 $f(R_{i,d}^{temp}) < f(R_i)$,更新操作将得到确认,此次进化成功, X_i 的位置将更新为 $R_{i,d}^{temp}$.其中 R_i 表示第 i 个粒子的记忆粒子位置(个体极值).否则,此次将不进行对该记忆粒子的更新.

和当前粒子群更新公式相比,式(6)更简单,而且记忆粒子群的更新不会影响到当前群体参数变化,当前群体的多样性会得到很好的保持.但因为缺少了个体极值经验的分享,仅仅通过全局极值更新记忆粒子群的位置,使得粒子群更新的牵制因子变少,记忆粒子群更容易快速的收敛到局部或者全局最优位置.

3.2 ADLPSO

上述提出利用记忆粒子群的特性可以加速算法收敛速度,但是该算法更容易陷入局部最优.为了平衡好算法的收敛速度与全局收敛性,本文提出一种自适应双层粒子群优化算法(ADLPSO),通过在双

层粒子群中使用一种自适应惯性权重的算法充分提高群体的多样性。

ADLPSO 算法使用一种基于粒子群分工的自适应惯性权重的方法,通过评价每个粒子的好坏程度,给予不同的粒子适合的惯性权重,各自完成相应的任务。这种方法增强了粒子群的多样性特点,平衡了 PSO 算法的全局搜索能力和局部寻优能力。图 1 显示了粒子群分类的方法。

图 1 中, f_i 是粒子 P_i 的适应度值, 粒子群的平均适应度值表示为 f_{avg} , f'_{avg} 是优于平均适应度值那些粒子的平均适应度值, f_m 表示当前群体最优粒子的适应度值。根据粒子的适应度值将粒子群分为 3 类: 较差粒子群、一般粒子群、优秀粒子群。同时采用文献[7]中提出的计算方法设置每种粒子的惯性权重。

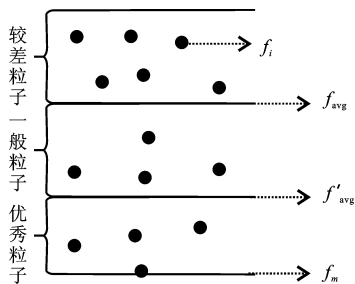


图 1 粒子分类图

因为记忆粒子群趋向于开发出最优解,可以利用记忆粒子群的特性提高算法的收敛速度,算法在群体更新时对记忆粒子群通过式(6)进行更新。因为记忆粒子群的更新不会影响当前粒子群的参数变化,这样就可以在提高收敛速度的前提下最小程度地影响当前群体的多样性。所以将上述提到的惯性权重自适应策略和记忆粒子群更新的操作结合起来,可以在尽量保证自适应策略增加群体多样性的情况下加速算法的收敛。

通过式(6)可以对记忆粒子群进行更新,但是对于其中参数 c_i 的选择还没有确定。粒子群在更新时,刚开始粒子比较发散时,大值 c_i 使得步长比较大,有利于全局搜索;迭代到后期小值 c_i 使得步长变小,更利于局部的寻优,所以群体越收敛时应该赋予记忆群体更小的 c_i 。其进化公式可以采取式(7)。其中, $c_{\min}$ 表示 c_i 取到的最小值。

$$c_i = c_i - (c_i - c_{\min}) * \left| \frac{f_i - f'_{\text{avg}}}{f_m - f'_{\text{avg}}} \right| \quad (7)$$

但是当前这种自适应操作是处于算法进入早熟收敛阶段进行的,仅当在这种阶段下进行自适应操

作使得种群的多样性不能得到充分的发挥。本文所提出的算法取消了发生自适应过程的条件,使得种群进化和自适应操作同步进行。通过在双层粒子群算法下应用改进的自适应过程得到最终的算法 ADLPSO,它能够有效地避免早熟收敛的发生,进一步提高收敛精度。

3.3 算法描述

如 3.2 节所述,记忆粒子群可以在最小程度地影响当前粒子群多样性的情况下加速算法收敛速度。自适应策略与双层粒子群的结合不仅可以保证自适应策略的优势又可以加速算法收敛速度。这种结合称之为初步的自适应双层粒子群优化算法 (ADLPSO1)。记忆群体的更新确实加强了算法的收敛速度。但是收敛速度的提高必然在一定程度上牺牲了算法的全局搜索能力。为了平衡好算法的收敛速度与全局收敛性,3.2 节中改进了算法的自适应过程,在 ADLPSO1 中去除发生自适应的条件,让自适应和群体更新同步进行,得到 ADLPSO2 算法 (即最终的 ADLPSO 算法),这进一步提高了粒子群的种群多样性。双层粒子群的结合加上改进的自适应策略使得 ADLPSO 算法既能使得算法避免早熟收敛又能加速算法收敛速度。

综上所述,本文所提出的 ADLPSO 算法,其算法流程描述如下:

- (1) 初始化粒子群,初始参数、粒子速度、位置,计算各个粒子的适应度值和个体极值和全局极值。
- (2) 对每个粒子 P_i 执行以下的操作:
 - ① 根据粒子的适应度值的不同为粒子 P_i 采取相应的惯性权重。
 - ② 根据式(1)和式(2)对 P_i 速度和位置进行更新。
 - ③ 根据式(6)更新记忆粒子 R_i 的位置,如果 $f(R_i^{\text{temp}}) < f(R_i)$,确认此次更新,记忆粒子将被新的位置 R_i^{temp} 取代。
 - ④ 计算 P_i 的适应度值 f_i 。
 - ⑤ 若 $f_i < f(P_{\text{best}i})$,则更新 $P_{\text{best}i}$ 为 P_i 当前位置。
 - ⑥ 若 $f_i < f(G_{\text{best}})$,则更新 G_{best} 为 P_i 当前位置。
- (3) 判断算法是否满足终止条件(迭代次数满或者达到最优解),若满足转向(4),否则(2)。
- (4) 输出全局最优解,算法结束。

4 仿真分析

本文采用四个经典的函数测试本文算法的性

能,即表 1.

表 1 测试函数描述				
函数名	维度	搜索区间	最优位置	最优值
Rosenbrock	30	$[-10,10]$	$(0,0,\cdots,0)$	0
Griewank	10	$[-600,600]$	$(0,0,\cdots,0)$	0
Rastrigin	30	$[-5.12,5.12]$	$(0,0,\cdots,0)$	0
Ackley	30	$[-32,32]$	$(0,0,\cdots,0)$	0

这 4 个基准函数中,Rosenbrock 是单峰函数,可以用来测试算法克服和防止早熟收敛的性能.Griewank、Rastrigin 和 Ackley 是多峰函数,具有大量的局部最优点.由于 Griewank 维度超过 15 维后函数特性趋于单峰,所以这里选取 10 维.

实验测试的参数列于表 2.

表 2 实验参数设置表	
参数名称	大小或区间
种群大小	30
c_1,c_2	2
c_{min}	0.2
初始 c_i	1
惯性权重区间	$[0.4,0.9]$
最大迭代次数	1 000
算法运行次数	30

本文提出的算法将与标准 PSO 算法(BPSO)和自适应粒子群算法(APSO)进行比较.设定如下几个评价标准:(1)平均最优值(MeanBst),运行算法 30 次后得到的最优值的期望,用来衡量粒子寻优的平均质量.(2)最优值(OptV),运行 30 次后,其中算法找到的最好一次的优化结果.(3)平均迭代次数(MeanC),算法优化结果成功时(即达到精度要求),需要平均的迭代次数,用来显示算法的收敛速度的性能.

通过实验得出表 3 所示的评价标准的数据.

从表中可以看出,ADLPSO 算法收敛速度明显快于 APSO.表 3 中的 MeanC 指标显示,对于 Griewank 和 Rastrigin 函数,APSO 和 ADLPSO 算法都得到了或者接近理论最优值,但是 ADLPSO 算法的收敛速度收提高了 25%左右.这证实了由于加入双层的概念,利用记忆粒子群更新的特性确实加快了算法的收敛.对于 Rosenbrock 和 Ackley 函数,本文算法和其他算法均没有达到理论最优值,但是可以看出本文所提出的 ADLPSO 算法得到的优化效果更优.具体来说,对于 Rosenbrock 函数,本文算法较 APSO 算法而言,收敛精度提高了 20%.对于 Ack-

表 3 评价标准对比				
函数名	算法	MeanBst	OptV	MeanC
Rosenbrock	BPSO	350.684	8.031 5	—
	APSO	26.751 8	25.344 6	—
	ADLPSO	20.885 0	18.301 1	—
Griewank	BPSO	0.013 8	1.756e—010	—
	APSO	0.781 9	0.436 8	580
	ADLPSO	0.262 3	0.320 8	450
Rastrigin	BPSO	48.824 3	22.884 0	—
	APSO	0	0	—
	ADLPSO	0	0	210.5
Ackley	BPSO	0.892 3	5.107e—006	—
	APSO	8.98e—01	4.441e—015	—
	ADLPSO	4.204e—015	8.881e—016	—

ley 函数,本文算法得到的的平均最优值较 APSO 算法提高了 14 个数量级.可以看出本文算法对于收敛精度的提高效果明显.这证实了改进的自适应操作充分提高了种群的多样性,有效的避免了算法早熟收敛.综述,ADLPSO 算法不仅保证了较好的优化效果,而且极大的提高了算法的收敛速度.

5 结束语

本文针对粒子群优化算法的早熟收敛和优化速度慢的问题,提出了一种自适应双层粒子群优化算法(ADLPSO).算法利用双层粒子群中记忆群体的更新操作对自适应粒子群算法进行优化,加快了算法的收敛速度.在此基础上通过取消发生自适应过程的条件增强了种群的多样性,有效避免了早熟收敛.实验仿真表明 ADLPSO 算法不仅具有较好的全局寻优能力,而且具有较快的收敛速度.

参考文献:

[1] Kennedy J, Eberhart R C. Particle swarm optimization [C]//IEEE International Conference on Neural Networks, Piscataway, NJ: IEEE Press, 1995: 1942-1948.

[2] Khare Anula, Rangnekar Saroj. A review of particle swarm optimization and its applications in solar photovoltaic system[J]. Applied SoftComputing, 2013, 12 (5): 2997-3006.

[3] Alec Banks, Jonathan Vincent, Chukwudi Anyakoha. A review of particle swarm optimization. Part I: background and development [J]. Nat Comput, 2007, 35 (6): 467-484.

其他优秀算法相比,边缘保持效果和图像清晰度较好.同时,该算法还可以被应用到解决曝光图像多于两张的情况,且效果较好.

参考文献:

- [1] Goshtasby A A. Fusion of multi-exposure images [J]. Image and Vision Computing, 2005, 23(6): 611-618.
- [2] Mertens T, Kautz J, Van Reeth F. Exposure fusion: a simple and practical alternative to high dynamic range photography [J]. Computer Graphics Forum, Blackwell Publishing Ltd, 2009, 28(1): 161-171.
- [3] Li S, Kang X. Fast multi-exposure image fusion with median filter and recursive filter [J]. Consumer Electronics, IEEE Transactions on, 2012, 58 (2): 626-632.
- [4] He K, Sun J, Tang X. Guided image filtering [J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35(6): 1397-1409.
- [5] Itti L, Koch C, Niebur E. A model of saliency-based visual attention for rapid scene analysis [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1998, 20 (11): 1254-1259.
- [6] Hou X, Zhang L. Saliency detection: A spectral residual approach[C]//Computer Vision and Pattern Rec-

ognition, CVPR'07. IEEE Conference on. USA: Minneapolis, IEEE, 2007: 1-8.

- [7] Lu S, Tan C S, Lim J H. Robust and efficient saliency modeling from image co-occurrence histograms [J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2014, 36(1): 195-201.
- [8] Hossny M, Nahavandi S, Creighton D. Comments on 'Information measure for performance of image fusion' [J]. Electronics letters, 2008, 44(18): 1066-1067.
- [9] Xydeas C S, Petrović V. Objective image fusion performance measure [J]. Electronics letters, 2000, 36 (4): 308-309.
- [10] 张少辉, 崔仲远, 于来行. 基于 DWT 结合 NSCT 的快速图像融合算法[J]. 微电子学与计算机, 2015, 32 (9): 40-44.

作者简介:

郭剑桥 男, (1994-), 硕士研究生. 研究方向为多曝光图像融合算法、硬件实现.

白雪飞(通信作者) 男, (1977-), 博士, 讲师. 研究方向为集成电路设计. E-mail: cbguo@mail.ustc.edu.cn.

杨灿美 男, (1965-), 博士, 研究员. 研究方向为信号处理、低功耗与高密度 SoC 架构与设计.

(上接第 13 页)

- [4] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]//Proceedings of the IEEE Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1998: 69-73.
- [5] Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J]. Computers & Operations Research, 2006, 33 (3): 859-871.
- [6] 郇振华, 梅莉, 祝远鉴. 复合策略惯性权重的粒子群优化算法[J]. 计算机应用, 2012, 32(8): 2116-2119.
- [7] 韩江洪, 李正荣, 魏振春. 一种自适应粒子群优化算法及其仿真研究[J]. 系统仿真学报, 2006, 18(10): 2969-2971.
- [8] 敖永才, 师奕兵, 张伟, 等. 自适应惯性权重的改进粒子群算法[J]. 电子科技大学学报, 2014, 55(6): 874-881.
- [9] Wei Hong Lim, Nor Ashidi Mat Isa. Two-layer particle swarm optimization with intelligent division of labor

[J]. Engineering Applications of Artificial Intelligence, 2013, 26(1): 2327-2348.

- [10] Epitropakis M C, Plagianakos V P, Vrahatis M N. Evolving cognitive and social experience in particle swarm optimization through differential evolution: a hybrid approach[J]. Information Sciences, 2010, 216 (24): 50-92.
- [11] Wei Hong Lim, Nor Ashidi, Mat Isa. An adaptive two-layer particle swarm optimization with elitist learning strategy [J]. Information Sciences, 2014 (273): 49-72.

作者简介:

周志勇 男, (1990-), 硕士研究生. 研究方向为群体智能优化算法的研究及其应用. E-mail: 15189828818@163.com.

李荣雨 男, (1977-), 教授. 研究方向为工业系统的监控与优化控制.