文章编号: 1006-3080(2006)04-0462-04

分段式微粒群优化算法

滕居特[1,2], 陈国初[1], 顾幸生[1]
(1. 华东理工大学自动化研究所, 上海 200237; 2. 华东理工大学研究生院, 上海 200237)

摘要: 提出一种分段式微粒群优化算法。该算法将所要搜索的区域分成若干段, 首先在每一区段内搜索出区段的最优位置, 然后将各区段的最优位置组成一微粒群, 继续搜索全局最优位置。通过对5 个常用标准测试函数进行优化计算, 仿真结果表明: 分段式微粒群优化算法能有效地搜索到全局最优解, 具有比基本微粒群优化算法更快的搜索速度和更好的优化性能。

关键词: 分段; 微粒群优化算法; 优化; 仿真
中图分类号: TP273 文献标识码: A

Multi-sections Particle Swarm Optimization Algorithm

TENG Ju-te[1,2], *CHEN Guo-chu*[1], *GU Xing-sheng*[1]
(1. *Research Institute of Automation, East China University of Science and Technology*, *Shanghai* 200237, *China*; 2. *Graduate School, East China University of Science and Technology*, *Shanghai* 200237, *China*)

Abstract: A multi-sections particle swarm optimization algorithm (MSPSO) is proposed. The new algorithm assumes that the search area is divided into several sections. The best position of each section is found using particle swarm optimization algorithm and a new swarm that consists of the best positions of sections searches for the global best position by particle swarm optimization algorithm again. Then, both MSPSO and PSO are used to resolve five widely used test functions´ optimization problems. Results show that MSPSO has quicker convergence velocity and better optimization performance than PSO.

Key words: multi-sections; particle swarm optimization algorithm; optimization; simulation

微粒群优化算法PSO(Particle Swarm Optimization Algorithm)是由 Kennedy 和 Eberhart 于 1995 年提出的一种新的进化计算算法[1~2], 它来源于鸟类或鱼类觅食过程中迁徙和群集的模拟。由于 PSO 概念和参数调整都很简单而且容易编程实现, 它既保持传统进化算法深刻的群体智慧背景, 同时又有自己许多良好的优化性能。因此, PSO 一经提出, 立刻引起进化计算领域学者们的广泛关注[3~4], 短短几年便获得快速发展, 在诸多领域得到应用而且范围越来越广泛[5~6], 已形成学术界一个新的研究热点。

但是, 基本微粒群优化算法存在搜索后期由于搜索策略效率低下导致的后期搜索速度慢、搜索效率低、优化性能不够好的缺陷[7~8]。本文针对基本微粒群优化算法的这一缺陷, 提出分段式微粒群优化算法 MSPSO(Multi-sections Particle Swarm Optimization Algorithm), 即将所要搜索的区域分成若干段, 首先在每一区段内搜索出区段的最优位置, 然后将各区段的最优位置组成一微粒群, 继续搜索全局最优位置。并用 5 种常用优化测试函数进行测试并进行比较。

收稿日期: 2005-04-15
基金项目: 国家自然科学基金项目(60274043); 国家高技术研究发展计划项目(2002AA412610)
作者简介: 滕居特(1967-), 女, 山东威海人, 博士生, 研究方向为智能控制技术。E-mail: jtteng@ecust.edu.cn

1　基本微粒群优化算法

假设在一个 D 维搜索空间中, 有 m 个微粒组成一微粒群, 其中第 i 个微粒的空间位置为 $x_i=(x_{i1}, x_{i2}, x_{i3}, \cdots, x_{iD})$, $i=1, 2, \cdots, m$, 它是优化问题的一个潜在解, 将它代入优化目标函数可以计算出相应的适应值, 根据适应值的大小可衡量 x_i 的优劣。第 i 个微粒所经历的最好位置称为其个体历史最好位置, 记为 $P_i=(p_{i1}, p_{i2}, p_{i3}, \cdots, p_{iD})$, 相应的适应值为其个体历史最好适应值 F_i; 同时, 每个微粒还具有各自的飞行速度 $V_i=(v_{i1}, v_{i2}, v_{i3}, \cdots, v_{iD})$。所有微粒经历过的位置中的最好位置称为全局历史最好位置, 记为 $P_g=(p_{g1}, p_{g2}, p_{g3}, \cdots, p_{gD})$, 相应的适应值为全局历史最好适应值 F_g。对每一代微粒, 其第 d 维($1\leq d\leq D$) 根据如下方程迭代[7]:

$$v_{id}=\chi(\omega v_{id}+c_1r_1(p_{id}-x_{id})+c_2r_2(p_{gd}-x_{id})) \tag{1}$$

$$x_{id}=x_{id}+v_{id} \tag{2}$$

其中: χ 为收缩因子; ω 为惯性权值; c_1 和 c_2 都为正常数, 称为加速系数; r_1 和 r_2 是两个在[0, 1] 范围内变化的随机数。式(1) 的第一部分由微粒先前速度的惯性引起, 为 ‘惯性 ’部分; 第二部分为 ‘认知 ’部分, 表示微粒本身的思考, 即微粒本身的最好信息对自己下一步行为的影响; 第三部分为 ‘社会 ’部分, 表示微粒间的信息共享和相互合作, 即群体最好信息对微粒下一步行为的影响。在整个搜索过程中, 微粒下一步的行动均是由先前速度的惯性、微粒本身的最好经验和整个群体的最好经验共同影响的结果。

搜索时, 微粒的位置被最大位置和最小位置限制, 如果某微粒在某维的位置超出该维的最大位置或最小位置, 则该微粒的位置被限制为该维的最大位置或最小位置。同样, 微粒的速度也被最大速度和最小速度所限制, 如果当前对微粒的加速度导致它在某维的速度超过该维的最大速度或最小速度, 则该微粒在该维的速度被限制为该维的最大速度或最小速度。

PSO 算法的搜索性能在很大程度上取决于算法参数的选择和控制, 包括微粒群规模的确定、收缩因子与惯性权值的控制、加速系数、最大允许迭代次数及其他参数的选择, 这是一项很难解决的工作。此外, 由于实际问题的多样性和复杂性, 尽管已出现了许多改进的 PSO 算法, 但远不能满足实际需要。研究新的改进的 PSO 算法以便能更好地用于实际问题的求解是很有意义的研究工作, 也是 PSO 算法当前的研究热点之一。

2　分段式微粒群优化算法

2.1　分段式微粒群优化算法原理

基本微粒群优化算法无需提供任何初始化信息就能在所求问题的解空间中求得最优解, 是一种新兴的、很有发展前景的全局优化方法。虽然基本微粒群优化算法简洁有效、对许多优化问题具有良好的优化性能。但是, 基本微粒群优化算法存在搜索后期由于搜索策略效率低下导致的后期搜索速度慢和搜索效率低的缺陷。通过基本微粒群优化算法的可视化工具, 可以发现微粒在接近全局历史最优位置时会快速地失去它的 “横向 ”运动能力, 导致微粒在全局历史最优位置周围徘徊而无法接近最优点。这种现象对优化算法来说是极其低效的, 因此必须寻求通过改进算法的寻优策略来加快微粒的寻优速度。尤其是在基本微粒群优化算法搜索的后期, 众多微粒都拥挤在历史最好位置周围, 很多微粒还在进行重复性的无效搜索。本文将搜索分成不同区段并进行分阶段搜索。将所要搜索的区域分成数个区段, 首先在每一区段内搜索出区段的最优位置, 然后将各区段的最优位置组成一微粒群, 继续搜索全局最优位置。

2.2　分段式微粒群优化算法流程

分段式微粒群优化算法流程如下:

(1) 初始化设置群体的规模、参数维数、惯性权值、加速系数、最大允许迭代次数或适应值误差限、所分区段数、各微粒的初始位置和初始速度等。

(2) 按目标函数评价各微粒的初始适应值。保留各微粒的初始适应值为其初始个体历史最好适应值, 找出各区段的初始区段最好适应值和初始区段最好位置。

(3) 根据式(1) 计算各微粒新的速度, 并对其进行限幅处理; 根据式(2) 计算各微粒新的位置, 并对其进行限幅处理。

(4) 更新各微粒个体历史最好适应值和个体历史最好位置; 更新各区段的区段历史最好适应值和区段历史最好位置。

(5) 如果各区段的区段历史最好适应值连续若干次(本文取 15 次) 没能得到改善, 则认为该区段的区段历史最好位置和最好适应值找到, 该区段历史最好微粒搜索结束。

(6) 如果所有区段的区段最好微粒均找到或目前迭代次数超过设定的迭代次数(本文为设定的最

大允许迭代次数的一半),转至步骤(7),否则转至步骤(3)。

(7) 将所有区段的区段最好微粒重新组合成一新的微粒群。

(8) 根据式(1)计算各微粒新的速度,并对微粒新的速度进行限幅处理;根据式(2)计算各微粒新的位置,并对微粒新的位置进行限幅处理。

(9) 更新新微粒群中各微粒个体历史最好适应值和个体历史最好位置;更新新微粒群中全局历史最好适应值和全局历史最好位置。

(10) 若满足停止条件(适应值误差小于设定的适应值误差限或迭代次数超过设定的最大允许迭代次数),搜索停止,输出全局历史最优位置和全局历史最优适应值为所求结果。否则,返回步骤(8)继续搜索。

3 测试函数的优化与结果讨论

3.1 测试函数

本文用的标准测试函数为:

(1) Rosenbrock 函数

$$\min f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

其中: $-2.048 \leq x_1, x_2 \leq 2.048$, 该函数有一个全局最小(1, 1),最小值为0。

(2) Levy F5 函数

$$\min f(x, y) = \left\{\sum_{i=1}^{5} i \cos[(i-1)x + i]\right\} \times \left\{\sum_{i=1}^{5} i \cos[(i+1)y + i]\right\} + [(x + 1.425\,13)^2 + (y + 0.800\,32)^2]$$

其中: $-10 \leq x, y \leq 10$, 此函数有760个局部极小点,其中只有一个(-1.306 8, -1.424 8)为全局最小,最小值为-176.137 5。

(3) Ackley 函数

$$\min f(x) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n} x_j^2}} - e^{\frac{1}{n}\sum_{j=1}^{n}\cos(2\pi x_j)} + 22.712\,82$$

其中: $-5 \leq x_j \leq 5$, 此函数有一个(0, 0)为全局最小,最小值为0。这个函数的搜索十分困难,求解Ackley函数的最小值是优化算法应用的一个有力的例证。

(4) Rastrigin 函数

$$\max f(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$$

其中: $-5.12 \leq x_i \leq 5.12$, 当 $n = 2$ 时,该函数有4个全局最大值点,全局最大值为80.706 6。

(5) Alpine 函数

$$\max f(x) = \left[\prod_{i=1}^{n} \sin(x_i)\right]\left[\sqrt{\prod_{i=1}^{n}(x_i)}\right]$$

其中: $0 \leq x \leq 10$, 此函数有一个(7.917, 7.917)为全局最大,最大值为2.808($n = 20$),(当 $n = 2$ 时,全局最大值为7.885 6)。

3.2 实验结果与讨论

本文同时用分段式微粒群优化算法和基本微粒群优化算法对以上函数进行优化测试实验。共有两组测试实验,第一组测试实验主要考察两种算法寻优时的达优率,第二组实验主要考察两种算法寻优时的搜索速度。

达优率是优化算法在寻优时找到全局最优解的概率。比如,某一优化算法对某一优化测试函数进行独立100次寻优测试,结果有80次测试找到了全局最好解,那么就说这一优化算法在这一测试条件下对这一测试函数的达优率为80%。用优化函数寻优,首先追求的是寻优率,如果优化时全局最优值都极难找到,那优化性能也就无从谈起了。所以,达优率是优化函数寻优时最重要的一项指标。达优率越高,说明优化算法越容易找到全局最优解,算法的优化性能越好;反之,说明优化算法越不容易找到全局最优解,算法的优化性能就越不尽如人意。为了比较两种算法对以上测试函数优化时的达优率,第一组测试实验设计如下:寻优时,最大允许迭代次数为3 000,惯性权值从1.8衰变到0.06,加速系数 c_1、c_2 为2,收缩因子为0.8;基本微粒群的总微粒个数为100;在分段微粒群优化算法中,将所搜索区域分为10区段,每区段的微粒个数为10。由于两种算法都是随机搜索算法,每次搜索的结果有可能不同,单凭一两次搜索的结果难以说明搜索的有效性。所以,本文从概率的角度对比算法的有效性,优化时用两算法对每一测试函数都进行100次独立测试,并进行统计汇总,对比结果见表1。

由表1可知,对于Rosenbrock、Levy F5、Ackley、Rastrigin和Alpine函数,两优化算法在实验中都能很好地逃脱局部极值点,并都能以1的概率找到全局最好解。

收敛时间也是优化算法寻优时所要考虑的另一项重要指标。收敛时间越短,则算法的搜索速度越快,算法更容易收敛于全局最优解;反之,则算法的搜索速度越慢,算法更不容易收敛于全局最优解。为了比较MSPSO和PSO对以上测试函数优化时的收敛时间,第二组测试实验和第一组测试实验大体

表 1　两种优化算法寻优时的达优率、适应值平均值与适应值最优值的对比

Table 1　Success rate, mean fitness and optimal fitness of functions found for the test functions with MSPSO and PSO

Test functions	PSO			MSPSO		
	Success rate (%)	Mean fitness	Optimal fitness	Success rate (%)	Mean fitness	Optimal fitness
Rosenbrock	100	0	0	100	0	0
Levy F5	100	− 176.14	− 176.14	100	− 176.14	− 176.14
Ackley	100	0	0	100	0	0
Rastrigin	100	80.706 6	80.706 6	100	80.706 6	80.706 6
Alpine	100	78.885 6	78.885 6	100	78.885 6	78.885 6

上相似, 只是在第二组实验中, 为了确保每一优化算法对每一测试函数都能找到全局最好解, 最大允许迭代次数设为 4 000, 适应值误差限为 10^{-5}, 也就是倘若对某测试函数运行了一段时间后, 该时刻的全局历史最好适应值和该函数最优点值的误差绝对值小于 10^{-5}, 就认为该算法在此时刻找到了这一测试函数的全局最优解, 那么运行到该时刻所花的时间就为该算法对这一测试函数本次的收敛时间。同样, 为了减弱随机性的影响, 本文从平均值的角度对比两算法的收敛时间, 优化时用两算法对每一测试函数都进行 100 次独立测试, 并进行平均值计算, 两算法对每一测试函数的平均收敛时间对比结果见表 2。

表 2　两种优化算法寻优时平均收敛时间的对比

Table 2　Mean convergence time for the test functions with MSPSO and PSO

Test functions	Mean convergence time/s	
	PSO	MSPSO
Rosenbrock	1.438	0.726
Levy F5	2.266	1.094
Ackley	1.971	0.998
Rastrigin	1.475	0.821
Alpine	1.783	0.852

由表 2 可知, 对于 Rosenbrock、Levy F5、Ackley、Rastrigin 和 Alpine 函数, 分段式微粒群优化算法的收敛时间明显小于基本微粒群优化算法的收敛时间。分段式微粒群优化算法能比基本微粒群优化算法更快地找到全局最好解, 搜索速度明显提高, 优化性能明显得到改善。

4　结束语

针对基本微粒群优化算法存在搜索后期由于搜索策略效率低下导致的后期搜索速度慢、搜索效率低、优化性能不够好的缺陷, 提出一种改进的微粒群优化算法—— 分段式微粒群优化算法, 它能有效地搜索到全局最优解, 搜索速度和优化性能比基本微粒群优化算法有明显提高, 是一种快速有效、很有应用前景的优化算法。

参考文献:

[1]　Kennedy J, Eberhart R C. Particle swarm optimization [A]. Proc IEEE Int Conference on Neural Networks [C]. Piscataway, NJ: IEEE Service Center, 1995. 1942–1948.

[2]　Eberhart R C, Kennedy J. A new optimizer using particle swarm theory [A]. Proc the Sixth Int Symposium on Micro Machine and Human Science[C]. Nagoya, Japan: IEEE Service Center, 1995. 39–43.

[3]　Eberhart R C, Shi Y. Particle swarm optimization: Developments, applications and resources [A]. Proc 2001 Congress on Evolutionary Computation [C]. Seoul, South Korea: IEEE Service Center, 2001. 81–86.

[4]　Lu W Z, Fan H Y, Lo S M. Application of evolutionary neural network method in predicting pollutant levels in downtown area of Hong Kong [J]. Neurocomputing, 2003, **51**(1): 387–400.

[5]　陈国初, 俞金寿. 微粒群优化算法[J]. 信息与控制, 2005, **34**(3): 318–324.

[6]　陈国初, 俞金寿. 增强型微粒群优化算法及其在软测量中的应用[J]. 控制与决策, 2005, **20**(4): 377–381.

[7]　Parsopoulos K E, Vrahatis M N. Recent approaches to global optimization problems through particle swarm optimization [J]. Natural Computing, 2002, **1**(2–3): 235–306.

[8]　Ioan Cristian Trelea. The particle swarm optimization algorithm: Convergence analysis and parameter selection [J]. Information Processing Letters, 2003, **85**(1): 317–325.