

基于微分控制策略的快速粒子群优化算法

樊吕彬, 刘亚红, 张 玮

(太原理工大学 化学化工学院, 太原 030024)

摘 要: 标准粒子群优化算法的速度更新机制为比例-积分(PI)控制策略, 而由于其中固有积分项的存在, 系统容易产生振荡, 导致搜索速度慢。为此, 根据比例-积分-微分(PID)控制特性, 提出一种快速粒子群优化算法。在标准粒子群及其改进算法中加入微分控制来克服振荡, 提高收敛速度, 增加搜索过程的稳定性。仿真结果表明, 与标准粒子群算法和全信息粒子群算法相比, 该算法在保证寻优精度和可靠性的同时, 大幅提高了寻优速度, 具有较高的运算效率。

关键词: 粒子群优化; 控制策略; 微分控制; 偏差; 收敛速度

中文引用格式: 樊吕彬, 刘亚红, 张 玮. 基于微分控制策略的快速粒子群优化算法[J]. 计算机工程, 2018, 44(2): 244-250, 281.

英文引用格式: FAN Lübin, LIU Yahong, ZHANG Wei. Fast Particle Swarm Optimization Algorithm Based on Differential Control Strategy[J]. Computer Engineering, 2018, 44(2): 244-250, 281.

Fast Particle Swarm Optimization Algorithm Based on Differential Control Strategy

FAN Lübin, LIU Yahong, ZHANG Wei

(College of Chemistry and Chemical Engineering, Taiyuan University of Technology, Taiyuan 030024, China)

[Abstract] The velocity updating mechanism of Particle Swarm Optimization (PSO) algorithm is Proportion-Integral (PI) control strategy. On account of the inherent integral term, the system is prone to oscillation, which leads to the low search speed. Therefore, according to the characteristics of Proportional-Integral-Derivative (PID) control, this paper proposes a fast PSO algorithm. The differential control is added in the standard PSO algorithm and its improved algorithm to overcome the oscillation, improve the convergence speed, and increase the stability of the search process in the meantime. Simulation results show that, compared with the standard PSO algorithm and the full information PSO algorithm, the proposed algorithm improves the search speed and has high computing efficiency while ensuring the optimization accuracy and reliability.

[Key words] Particle Swarm Optimization (PSO); control strategy; differential control; deviation; convergence speed

DOI: 10.3969/j.issn.1000-3428.2018.02.042

0 概述

粒子群优化(Particle Swarm Optimization, PSO)算法是一种基于种群全局随机搜索策略的智能进化算法^[1]。相比于其他智能算法, 粒子群优化算法在解决多目标优化、动态寻优等问题上, 具有结构简单、易编程实现等特点, 被广泛应用于函数优化、资源配置、神经网络训练和路径选择等多个领域。

为提高粒子群优化算法种群在进化后期的多样性, 避免寻优过程陷入局部最优, 克服早熟收敛, 相关学者提出了多种改进方法, 按照改进机制的不同主要是在参数设置^[2]、学习机制^[3]和拓扑结构^[4]等 3 个方面。

相应的改进算法在避免陷入局部最优上取得了较好的效果, 但对于加快算法的寻优速度、提高优化效率, 则需要考虑研究算法本身的特性进而找到改进策略。比例-积分-微分(Proportional-Integral-Derivative, PID)控制是目前最常用的控制策略, 已大量应用于各种控制问题。受 PID 控制的启发, 文献[5]认为 PSO 是一种有 2 个输入和 1 个输出的反馈控制系统, 通过在粒子群优化算法中引入加速度项, 使算法进化行为呈现三阶系统特性, 增加种群产生的多样性; 文献[6]针对 PIDPSO 引入加速度项后可选参数多的问题, 为提高算法的计算效率, 使参数随着进化代数而变化, 提出了一种自适应 PID 粒子群算法; 文献[7]提出了一种增量型 PID 控制器 PSO

基金项目: 山西省自然科学基金(2015011019)。

作者简介: 樊吕彬(1992—), 男, 硕士研究生, 主研方向为智能算法; 刘亚红, 硕士研究生; 张 玮(通信作者), 副教授、博士。

收稿日期: 2017-01-11 **修回日期:** 2017-02-18 **E-mail:** zhangwei01@tyut.edu.cn

(IPID-PSO),通过对粒子轨迹特性的分析,经过变换后将粒子群速度更新机制解释为一个增量型PID控制过程。

本文受前人工作的启发,研究种群寻优过程的运行机制,推导出PSO近似于一种比例-积分(Proportional-Integral,PI)控制,而由于其固有积分属性的存在,使算法收敛速度变慢。为使种群在寻优过程中快速向最优位置搜索,减少与全局最优值之间的偏差,避免寻优过程振荡,通过添加微分项,提出微分控制的改进策略,利用该策略加快标准粒子群及其改进算法的收敛速度,改善搜索特性,提高寻优效率。

1 粒子群算法速度轨迹分析

在标准PSO中,整个种群随机初始化 m 个粒子,每个粒子根据其速度和位置的不同,随机分配一个速度向量 $\mathbf{v}_i(t)$ 和位置向量 $\mathbf{x}_i(t)$ 。粒子在多维空间的飞行过程中逐代搜索最优解,其中每个粒子都是潜在解。粒子根据其自身和整个种群所经过的最好适应度值的位置确定下一次飞行的速度和方向,个体所经过的适应度最好位置称为个体最优点 $pbest$,种群经过的适应度最好的位置称为全局最优点 $gbest$ 。种群根据个体最优点和全局最优点进行逐代进化,其速度更新公式为:

$$\mathbf{v}_{i,d}(t+1) = w \mathbf{v}_{i,d}(t) + c_1 r_1 (pbest_{i,d} - \mathbf{x}_{i,d}(t)) + c_2 r_2 (gbest_d - \mathbf{x}_{i,d}(t)) \quad (1)$$

种群位置更新公式为:

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

其中, w 称为惯性权重,表示对上一代速度的惯性继承。 c_1 和 c_2 为常数,分别称为个体和社会加速因子,用以调节迭代步长。 r_1 和 r_2 为取值在 $[0,1]$ 之间服从均匀分布的随机数。

2 粒子群优化算法的控制策略分析及改进

2.1 粒子群算法控制策略分析

在标准粒子群算法中,加速因子 c_1 和 c_2 可以平衡种群在进化过程中的全局勘探和局部探索能力,通常取 $c_1 = c_2 = c$,则由式(1)得:

$$\begin{aligned} \mathbf{v}(t+1) &= w\mathbf{v}(t) + c\{r_1(\mathbf{p}_i(t) - \mathbf{x}(t)) + r_2(\mathbf{p}_g(t) - \mathbf{x}(t))\} = \\ &= w\mathbf{v}(t) + c(r_1 + r_2)\left(\frac{r_1}{r_1 + r_2}\mathbf{p}_i(t) + \frac{r_2}{r_1 + r_2}\mathbf{p}_g(t) - \mathbf{x}(t)\right) = \\ &= w\mathbf{v}(t) + ca\{a_1\mathbf{p}_i(t) + (1 - a_1)\mathbf{p}_g(t) - \mathbf{x}(t)\} \end{aligned} \quad (3)$$

其中, a_1 为取值在 $[0,1]$ 之间的随机数, a 为取值在 $[0,2]$ 之间的随机数。

令 $\mathbf{B}(t) = a_1\mathbf{p}_i(t) + (1 - a_1)\mathbf{p}_g(t)$,则 $\mathbf{B}(t)$ 为个体最优值位置与全局最优值位置的矢量和,将其代入式(3)可得:

$$\mathbf{v}(t+1) = w\mathbf{v}(t) + ca\{\mathbf{B}(t) - \mathbf{x}(t)\} \quad (4)$$

根据控制策略理论,将粒子群优化算法视为反馈控制系统,把种群个体作为控制对象, $\mathbf{B}(t)$ 位置为给定值,个体位置 $\mathbf{x}(t)$ 为测量值,则其反馈控制的系统偏差为 $e(t) = \mathbf{B}(t) - \mathbf{x}(t)$,通过负反馈调节,使粒子的当前位置向最优粒子靠近。

首先假设每次迭代时最优粒子位置 $\mathbf{B}(t)$ 固定,将式(4)带入式(2)中得:

$$\mathbf{x}(t+1) = \mathbf{x}(t) + w\mathbf{v}(t) + ca\{\mathbf{B}(t) - \mathbf{x}(t)\} \quad (5)$$

对上式两侧进行期望操作,由于 a_1 和 a 为随机数,则其期望值分别为 $\frac{1}{2}$ 和1,式(5)经整理得:

$$E\mathbf{x}(t+1) - E\mathbf{x}(t) = wE\mathbf{v}(t) + c(\mathbf{B}' - E\mathbf{x}(t)) \quad (6)$$

其中, $\mathbf{B}' = \frac{1}{2}(\mathbf{p}_i(t) + \mathbf{p}_g(t))$ 。

利用式(6)的递推关系可得:

$$E\mathbf{x}(t+1) - E\mathbf{x}(1) = w(E\mathbf{x}(t) - E\mathbf{x}(0)) + c \sum_{j=0}^k (\mathbf{B}' - E\mathbf{x}(j)) \quad (7)$$

根据偏差 $e(t)$ 的定义将上式整理为:

$$E\mathbf{x}(t+1) = E\mathbf{x}(t) + (1 - w)Ee(t) + c \sum_{j=0}^k Ee(j) + \mathbf{x}(1) - w\mathbf{x}(0) - (1 - w)\mathbf{B}' \quad (8)$$

则其速度期望为:

$$E\mathbf{v}(t+1) = (1 - w)Ee(t) + c \sum_{j=0}^k Ee(j) + \mathbf{x}(1) - w\mathbf{x}(0) - (1 - w)\mathbf{B}' \quad (9)$$

式(9)通过与PID控制策略的位置型表达式(10)^[8]相类比,可得粒子群优化算法的速度更新机制为比例-积分(Proportional-Integral,PI)控制。

$$\mathbf{u}(t) = K_p e(t) + K_i \sum_{k=0}^t e(k) + K_d (e(t) - e(t-1)) \quad (10)$$

其中,比例项为 $K_p = 1 - w$,积分项 $K_i = c$,速度初值为 $\mathbf{x}(1) - w\mathbf{x}(0) - (1 - w)\mathbf{B}'$ 。因此可以将标准粒子群算法视为一种比例-积分(PI)控制策略。

2.2 微分项粒子群算法控制策略

文献[9]认为不同的系统性能参数会使粒子群算法的寻优过程呈现出不同性能:超调量越大则种群将会在更大搜索空间寻优,算法具有更强的探测能力;衰减比越大,则种群将以更快的速度靠近最优值位置,但会减弱粒子群的搜索能力,不利于跳出局部最优。

根据偏差的比例(P)、积分(I)、微分(D)进行控制,是控制系统中应用最为广泛的一种控制规律。经过上节分析可知,粒子群优化算法的寻优过程为比例-积分(PI)控制。若在粒子群优化算法中加入微分控制项,则:

1)微分控制项会减小超调量和衰减比,克服系统振荡,使响应过程更加柔和,提高系统的稳定性。

2) 微分控制能够加快系统的响应速度,减小调整时间,使得系统响应快速趋于稳态。

离散化的微分项位置型控制算式为:

$$\mathbf{D}(t) = K_d(\mathbf{e}(t) - \mathbf{e}(t-1)) \quad (11)$$

离散控制算式有位置型和增量型 2 种。位置型控制算式会使偏差逐代累积,同时占用较多的数据储存单元,不利于计算机运算。因此,将微分项的位置型控制算式相减可得增量型控制算式:

$$\begin{aligned} \Delta \mathbf{D}(t) &= \mathbf{D}(t) - \mathbf{D}(t-1) = \\ &K_d(\mathbf{e}(t) - \mathbf{e}(t-1)) - K_d(\mathbf{e}(t-1) - \mathbf{e}(t-2)) = \\ &K_d\{(\mathbf{B}(t) - \mathbf{x}(t)) - 2(\mathbf{B}(t-1) - \mathbf{x}(t-1)) + \\ &(\mathbf{B}(t-2) - \mathbf{x}(t-2))\} \end{aligned} \quad (12)$$

将上式带入标准粒子群算法的速度迭代方程,可得带微分控制项的速度迭代方程:

$$\begin{aligned} \mathbf{v}_{i,d}(t+1) &= w \mathbf{v}_{i,d}(t) + c_1 r_1 (pbest_{i,d} - \mathbf{x}_{i,d}(t)) + \\ &c_2 r_2 (gbest_d - \mathbf{x}_{i,d}(t)) + \Delta \mathbf{D}_i \end{aligned} \quad (13)$$

由于在进化过程中通过对标准粒子群算法引入微分项,利用微分控制策略对最优位置的偏差进行反馈调节,因此同样可将微分环节引入到改进粒子群算法中。

通过将微分控制项引入标准粒子群优化 (Standard SPO, SPSO) 算法及全信息粒子群 (Full Information Particle Swarm, FIPS) 算法^[10],分别可得其改进算法 D-SPSO 和 D-FIPS 的速度迭代方程。

D-SPSO 算法速度迭代方程为:

$$\begin{aligned} \mathbf{v}_{i,d}(t+1) &= w \mathbf{v}_{i,d}(t) + c_1 r_1 (pbest_{i,d} - \mathbf{x}_{i,d}(t)) + \\ &c_2 r_2 (gbest_d - \mathbf{x}_{i,d}(t)) + \Delta \mathbf{D}_i \end{aligned} \quad (14)$$

其中, $\mathbf{B} = \frac{1}{2}(\mathbf{p}_i(t) + \mathbf{p}_g(t))$ 。

D-FIPS 算法速度迭代方程为:

$$\mathbf{v}_{i,d}(t+1) = w \mathbf{v}_{i,d}(t) + c_k (\mathbf{FIPS}(t) - \mathbf{x}_{i,d}(t)) + \Delta \mathbf{D}_i \quad (15)$$

其中, $\mathbf{B} = \mathbf{FIPS}(t)$ 。

2.3 轨迹特性分析

由于粒子群算法在解空间中进行随机搜索,因此搜索轨迹的稳定性是衡量其性能的重要指标^[11],以改进算法 D-SPSO 为例对其搜索轨迹的收敛性进行证明。

定理 参数 w 、 $c_1 + c_2$ 和 K_d 满足条件式 (16) 时, D-SPSO 算法是稳定收敛的。

$$\begin{cases} c_1 + c_2 > 0 \\ 4 - 4w + (c_1 + c_2) > 0 \\ 8K_d - (c_1 + c_2) + 8 > 0 \\ 4 + 4w - (c_1 + c_2) - 8K_d > 0 \\ 2(1-w)(1+K_d) + (c_1 + c_2)K_d > 0 \end{cases} \quad (16)$$

证明:通过进化方程可以看出,粒子在每一维的更新都是相互独立的,且由于各粒子是随机生成,因此可以将算法设为一维情况进行处理,其结果同样适用于其他粒子。根据降维简化,将式 (14) 带入式 (2),得到如下离散型迭代公式:

$$\begin{aligned} \mathbf{x}_{t+3} + (c_1 r_1 + c_2 r_2 - 1 - w + K_d) \mathbf{x}_{t+2} + (w - 2K_d) \mathbf{x}_{t+1} + \\ K_d \mathbf{x}_t - c_1 r_1 pbest(t) - c_2 r_2 gbest(t) = 0 \end{aligned} \quad (17)$$

假设在很小的迭代次数内,种群的全局最优值 p_{gbest} 保持不变,对式 (17) 两边求其数学期望^[12],由于 r_1 和 r_2 为 $[0,1]$ 之间均匀分布的随机数,因此以期望值为 $\frac{1}{2}$,则得到下式:

$$\begin{aligned} Ex_{t+3} + \left(\frac{c_1 + c_2}{2} - 1 - w + K_d \right) Ex_{t+2} + (w - 2K_d) Ex_{t+1} + \\ K_d Ex_t = \frac{c_1 pbest + c_2 gbest}{2} \end{aligned} \quad (18)$$

根据 Z 变换平移定理^[13]对式 (18) 进行变换操作,经整理后如下:

$$(z^3 + T_1 z^2 + T_2 z + T_3) Ex(z) = S \quad (19)$$

其中, $T_1 = \frac{c_1 + c_2}{2} - 1 - w + K_d$, $T_2 = w - 2K_d$, $T_3 = K_d$ 。

$$\begin{aligned} S &= \frac{c_1 pbest + c_2 gbest}{2} + z^3 x(0) + \\ &z^2 [x(1) + T_1 x(0)] + z [x(2) + T_1 x(1) + T_2 x(0)] \end{aligned}$$

令 $z = \frac{u+1}{u-1}$,对式 (19) 进行双线性变换:

$$Au^3 + Bu^2 + Cu + D = S \quad (20)$$

其中, $A = 1 + T_1 + T_2 + T_3$, $B = 3 + T_1 - T_2 - 3T_3$, $C = 3 - T_1 - T_2 + 3T_3$, $D = 1 - T_1 + T_2 - T_3$ 。

由式 (20) 可以得出,其特征方程为:

$$Au^3 + Bu^2 + Cu + D = 0 \quad (21)$$

离散系统稳定性判据:在劳斯阵列表第 1 列元素均大于 0,且特征方程各系数均取正值时,系统趋于稳定。

根据离散系统稳定性判据可得,若要使系统趋于稳定,则特征方程式 (21) 应满足下式:

$$\begin{cases} A > 0 \\ B > 0 \\ C > 0 \\ D > 0 \\ \frac{BC - AD}{B} > 0 \end{cases} \Rightarrow \begin{cases} c_1 + c_2 > 0 \\ 4 - 4w + (c_1 + c_2) > 0 \\ 8K_d - (c_1 + c_2) + 8 > 0 \\ 4 + 4w - (c_1 + c_2) - 8K_d > 0 \\ 2(1-w)(1+K_d) + (c_1 + c_2)K_d > 0 \end{cases} \quad (22)$$

即当参数 w 、 $c_1 + c_2$ 和 K_d 满足条件式 (22) 时,每个粒子的寻优轨迹收敛,种群寻优过程趋于稳定。证毕。

2.4 算法流程

微分控制策略标准粒子群算法 D-SPSO 的执行流程如图 1 所示。

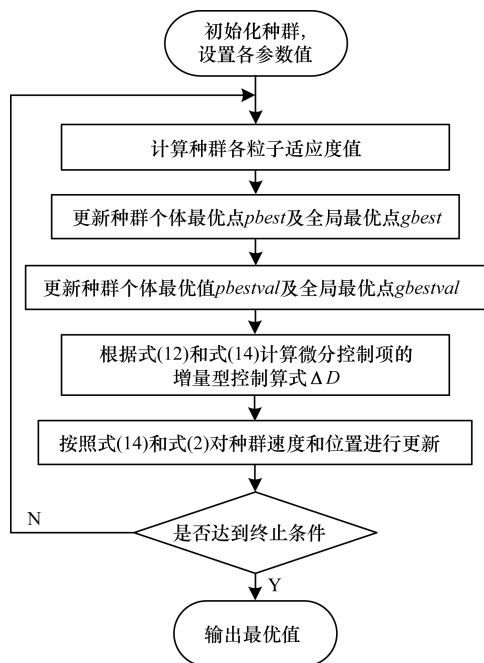


图1 D-SPSO 算法流程

微分控制策略全信息粒子群算法 D-FIPS 的执行步骤如下:

步骤1 初始化种群,随机生成种群中各粒子位置和速度,并设置各参数值。

步骤2 计算种群中每个粒子适应度值。

步骤3 更新种群个体最优值 $pbest$ 及个体最优值 $pbestval$ 。

步骤4 更新种群各邻域中的全信息系数 c_k 及全信息粒子 $FIPS^{[10]}$ 。

步骤5 根据式(12)和式(15)计算微分控制项的增量型控制算式 ΔD 。

步骤6 按照式(15)和式(2)对种群速度和位置进行更新。

步骤7 判断是否达到终止条件,若满足,则停止迭代,输出最优值;否则,迭代次数 $t = t + 1$,并转至步骤2。

3 实验仿真及性能分析

3.1 仿真实验结果

为了验证微分控制策略的有效性,分别利用标准粒子群算法 SPSO 和改进算法 FIPS 进行实验,并选取7个经典测试函数,通过仿真实验分析其性能。测试函数的寻优范围、极值和目标精度如表1所示。

表1 经典测试函数

序号	函数名	寻优范围	极值	精度
F1	Sphere	$[-100, 100]^D$	0	$1.0e-6$
F2	Rosenbrock	$[-2.048, 2.048]^D$	0	100
F3	Griewank	$[-600, 600]^D$	0	$1.0e-2$
F4	Ackley	$[-32, 32]^D$	0	$1.0e-2$
F5	Rotated Griewank	$[-600, 600]^D$	0	$1.0e-2$
F6	Shifted Rosenbrock	$[-2.048, 2.048]^D$	390	100
F7	Shifted Griewank	$[-600, 600]^D$	-180	$1.0e-2$

Sphere 为单模态二次函数,在定义域中只有全局极小值点没有局部极小值点,是评价算法收敛精度的经典函数;Rosenbrock 为非凸、病态单峰函数,最优点不在空间中心,算法很难辨别搜索方向;Griewank 为剧烈震荡的多峰函数,其局部极小值随维数不断增加,在低维空间更难寻优;Ackley 为具有大量局部最优点的多峰函数,全局最优解位于一个狭窄区域,多峰函数常用以评价算法跳出局部最优的性能。

在实验过程中,每组实验独立运行30次,种群经过5000次迭代后停止运行,即达到 $2.5E+05$ 次函数计算次数(FE)时终止运算。种群规模设为50,在30维空间内进行寻优,粒子的最大寻优速度 V_{\max} 通常设置为寻优范围的0.1倍。惯性权重 w 采用文献[14]提出的线性递减惯性权重策略,以平衡进化过程中的局部和全局搜索能力。

根据轨迹稳定性分析可得:当参数 $w, c_1 + c_2$ 取值为表2中时, K_d 取值为 $[0, 0.2]$ 。分别对微分控制策略的 K_d 采用阶跃值和固定值,采用阶跃值时,前期较大的 K_d 有助于增加微分控制的作用,减小系统的超调量,加快收敛速度;后期采用较小的 K_d ,突出积分项的影响,增加种群的多样性,同时进行精细搜索提高精度,从而在整个过程中提高寻优效率。各算法的参数设置如表2所示。

表2 各算法及参数设置

算法	参数设置
SPSO	$w = 0.4 \sim 0.9, c_1 = c_2 = 2.0$ $w = 0.4 \sim 0.9, c_1 = c_2 = 2.0$
PID-SPSO1	$K_d = \begin{cases} 0.18, & i < 4000 \\ 0.05, & \text{其他} \end{cases}$
PID-SPSO2	$w = 0.4 \sim 0.9, c_1 = c_2 = 2.0, K_d = 0.05$
FIPS	$w = 0.4 \sim 0.9, \sum c = 4.1$ $w = 0.4 \sim 0.9, \sum c = 4.1$
PID-FIPS1	$K_d = \begin{cases} 0.18, & i < 4000 \\ 0.05, & \text{其他} \end{cases}$
PID-FIPS2	$w = 0.4 \sim 0.9, \sum c = 4.1, K_d = 0.05$

3.2 算法性能分析

3.2.1 优化精度分析

从表 3 中的寻优结果来看,相比于标准 SPSO, D-SPSO1 和 D-SPSO2 算法对各个函数的适应度均值都有提高,表明在相同的计算次数下,D-SPSO 具有更高的计算效率。其中,*Mean* 表示在经过 $2.5E + 05$ 次函数计算后,所得到的平均适应度值,*Std* 表示 $2.5E + 05$ 次函数计算后的标准差,*Min* 表示迭代达到终止条件时,函数所得到的最小适应度值,*Imp* 表示改进算法的平均适应度值提升程度。在方差方面,D-SPSO 对函数 F1、F2、F4、F7 取得的性能上的提高,有助于提高算法的稳定性。同样对于改进算法 FIPS,如表 4 所示,D-FIPS1 和 D-FIPS2 算法对函数 F1、F3、F5、F7 的适应度均值都有较大提高,且方差优于 FIPS,可见微分控制策略不但提高了计算效率也增强了算法的稳定性。

表 3 测试函数仿真结果 1

函数	性能指标	SPSO 算法	D-SPSO1 算法	D-SPSO2 算法
F1	<i>Mean</i>	7.33E-51	2.90E-105	5.02E-63
	<i>Std</i>	1.57E-50	1.36E-104	2.75E-62
	<i>Min</i>	2.22E-54	2.54E-111	8.97E-82
	<i>Imp/%</i>	—	100	100
F2	<i>Mean</i>	3.04E+01	2.14E+01	2.70E+01
	<i>Std</i>	2.33E+01	1.51E+01	2.03E+01
	<i>Min</i>	2.83E+00	1.47E-02	3.57E-02
	<i>Imp/%</i>	—	29.6	11.8
F3	<i>Mean</i>	1.93E-02	1.41E-02	1.47E-02
	<i>Std</i>	2.49E-02	1.57E-02	1.89E-02
	<i>Min</i>	0.00E+00	0.00E+00	0.00E+00
	<i>Imp/%</i>	—	26.94	23.83
F4	<i>Mean</i>	2.47E-14	1.79E-14	2.35E-14
	<i>Std</i>	5.39E-15	5.04E-15	7.39E-15
	<i>Min</i>	1.33E-14	6.22E-15	1.33E-14
	<i>Imp/%</i>	—	27.53	4.86
F5	<i>Mean</i>	1.81E-02	8.69E-03	1.49E-02
	<i>Std</i>	1.83E-02	1.14E-02	1.53E-02
	<i>Min</i>	0.00E+00	0.00E+00	0.00E+00
	<i>Imp</i>	—	51.99	17.68
F6	<i>Mean</i>	2.51E+01	2.44E+01	2.48E+01
	<i>Std</i>	1.79E+01	1.91E+01	1.47E+01
	<i>Min</i>	3.99E+00	8.61E+00	7.65E+00
	<i>Imp/%</i>	—	2.78	1.2
F7	<i>Mean</i>	1.23E-02	8.78E-03	1.14E-02
	<i>Std</i>	1.56E-02	1.03E-02	1.11E-02
	<i>Min</i>	0.00E+00	0.00E+00	0.00E+00
	<i>Imp/%</i>	—	28.62	7.32
平均	<i>Imp/%</i>	—	38.21	23.81

表 4 测试函数仿真结果 2

函数	性能指标	FIPS 算法	D-FIPS1 算法	D-FIPS2 算法
F1	<i>Mean</i>	4.20E-32	4.47E-65	9.16E-47
	<i>Std</i>	2.52E-32	6.76E-65	5.99E-47
	<i>Min</i>	1.04E-32	1.72E-66	1.07E-47
	<i>Imp/%</i>	—	100	100
F2	<i>Mean</i>	2.42E+01	2.16E+01	2.31E+01
	<i>Std</i>	3.73E-01	7.37E-01	3.89E-01
	<i>Min</i>	2.37E+01	1.96E+01	2.23E+01
	<i>Imp/%</i>	—	10.74	4.55
F3	<i>Mean</i>	1.36E-05	3.97E-09	2.14E-08
	<i>Std</i>	6.30E-05	1.71E-08	6.91E-08
	<i>Min</i>	1.11E-16	0.00E+00	0.00E+00
	<i>Imp/%</i>	—	99.97	99.84
F4	<i>Mean</i>	2.66E-15	2.66E-15	2.66E-15
	<i>Std</i>	0.00E+00	0.00E+00	0.00E+00
	<i>Min</i>	2.66E-15	2.66E-15	2.66E-15
	<i>Imp/%</i>	—	0	0
F5	<i>Mean</i>	1.11E-05	5.46E-09	1.03E-05
	<i>Std</i>	5.88E-05	1.77E-08	4.02E-05
	<i>Min</i>	0.00E+00	0.00E+00	0.00E+00
	<i>Imp/%</i>	—	99.97	7.25
F6	<i>Mean</i>	2.41E+01	2.16E+01	2.32E+01
	<i>Std</i>	3.12E-01	7.89E-01	3.29E-01
	<i>Min</i>	2.34E+01	1.90E+01	2.21E+01
	<i>Imp/%</i>	—	10.37	3.72
F7	<i>Mean</i>	1.56E-05	9.59E-09	1.67E-07
	<i>Std</i>	6.43E-05	3.97E-08	6.72E-07
	<i>Min</i>	6.66E-15	0.00E+00	0.00E+00
	<i>Imp/%</i>	—	99.94	98.93
平均	<i>Imp/%</i>	—	60.14	35.70

为便于比较微分策略对粒子群改进算法的性能提高程度,通过适应度均值^[15]研究其性能的提升:

$$Imp = \frac{Mean(PSO) - Mean(D-SPSO)}{|Mean(PSO)|} \times 100\% \quad (23)$$

若微分控制改进粒子群算法得到的适应度均值更小,则 *Imp* 为正值,否则其为负值。

由表 3、表 4 可以看出,D-SPSO 和 D-FIPS 对原算法的平均适应度均有提高,其中,D-SPSO 对函数 F1、F2、F4、F5 和 F7 的提高均在 30% 左右;D-FIPS 对函数 F1、F3、F5 和 F7 取得了近似于 100% 的改进效果。对于改进算法较难优化的多峰函数和旋转与偏移函数,D-SPSO 和 D-FIPS 取得了较好的平均适应度提升度。同时根据平均提升度可以看出,在 K_d 采用阶跃值时,对于优化性能的提升度要高于采用固定值,说明在不同时期微分控制作用的强弱,对粒子群算法性能的改进具有明显的影响。

3.2.2 算法复杂度及平均计算次数分析

算法复杂度^[16]是评估算法性能的一个重要指标,决定了算法的寻优速度和优化效率。以标准粒子群算法 SPSO 为例,设求解问题的维数为 D ,种群规模为 N ,最大迭代次数为 T 。在每次进化过程中,算法主要的计算量包括粒子个体最优位置更新与全局最优位置更新两部分,在引入微分控制策略后,由于位置 $\mathbf{B}(t)$ 为 $pbest$ 和 $gbest$ 的矢量和,其更新由后者位置更新来确定。最复杂情况下最大计算时间复杂度为 $T \times O(N \times D)$,保留最优个体的存储空间复杂度为 $O(N \times D)$,所以该算法的最大复杂度:

$$O(T, N, D) = T \times O(N \times D) + O(N \times D) \quad (24)$$

由式(24)可以得到,相对于基本粒子群算法,在引入微分项后,微分策略的粒子群算法并没有明显地增加算法的复杂度。

图 2 和图 3 为各算法在经过 30 次实验仿真后所需的平均计算次数,其中函数计算次数(FE)反映了各算法在寻优过程中调用测试函数的次数。从图中可以看出:对各测试函数,SPSO 和 FIPS 算法的时间复杂度均最大,在引入微分策略后,使 D-SPSO 和 D-FIPS 算法达到寻优要求时的函数计算次数最少,加快了寻优速度。

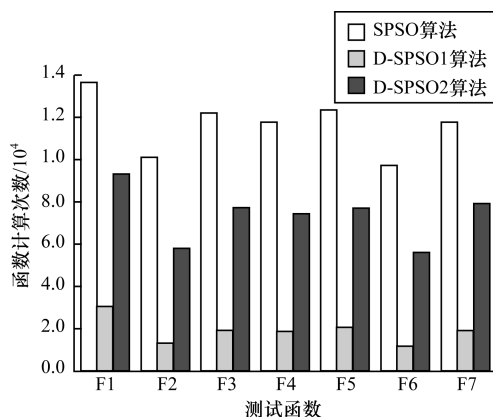


图 2 SPSO 与 D-SPSO 测试函数平均计算次数

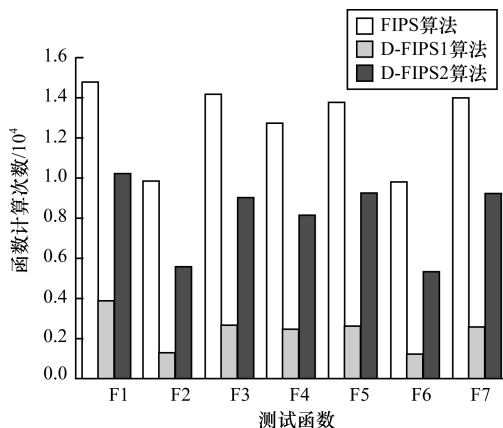


图 3 FIPS 与 D-FIPS 测试函数平均计算次数

3.2.3 优化成功率分析

表 5、表 6 记录了各算法的函数计算次数和收敛成功率。其中, Fe 表示算法在固定函数迭代次数内满足阈值所需的平均函数计算次数,反映了收敛速度的快慢, SR 表示算法成功收敛试验次数占总试验次数的比例。算法 SPSO、D-SPSO1 和 D-SPSO2 的平均函数计算次数分别为 116 745、19 466 和 73 826, 平均收敛成功率分别为 76.19%、84.76% 和 80.48%。算法 FIPS、D-FIPS1 和 D-FIPS2 的平均函数计算次数分别为 127 772、24 210 和 81 170, 平均收敛成功率均为 100%。证明 D-SPSO 和 D-FIPS 不仅能以很少的函数计算次数达到寻优精度要求,同时也能取得较高的成功率,说明微分控制策略不仅提高了优化速度也能够避免陷入局部最优,具有良好的优化效率。

表 5 仿真所需函数计算次数及成功率 1

函数	性能指标	SPSO	D-SPSO1	D-SPSO2
F1	FE	136 611	30 740	93 405
	SR/%	100	100	100
F2	FE	101 205	13 553	58 135
	SR/%	100	100	100
F3	FE	122 211	19 873	77 612
	SR/%	40	60	56.67
F4	FE	117 831	19 340	74 496
	SR/%	100	100	100
F5	FE	123 853	20 961	77 379
	SR/%	40	70	50
F6	FE	97 393	11 975	56 260
	SR/%	100	100	100
F7	FE	118 112	19 823	79 494
	SR/%	53.33	63.33	56.67

表 6 仿真所需函数计算次数及成功率 2

函数	性能指标	FIPS	D-FIPS1	D-FIPS2
F1	FE	148 735	39 075	102 531
	SR/%	100	100	100
F2	FE	98 860	13 126	54 393
	SR/%	100	100	100
F3	FE	142 423	26 990	90 408
	SR/%	100	100	100
F4	FE	127 511	25 236	81 856
	SR/%	100	100	100
F5	FE	138 455	26 695	92 610
	SR/%	100	100	100
F6	FE	97 986	12 101	53 481
	SR/%	100	100	100
F7	FE	140 436	26 250	92 915
	SR/%	100	100	100

3.2.4 优化过程曲线

为了更加直观地反映各算法的搜索特性,对SPSO和FIPS分别选择不同的测试函数,与相应的微分控制策略改进算法进行比较。从图4~图7可以看出,微分策略在保证足够的寻优精度条件下,能够加快种群寻优速度,提高优化效率,同时对于采用不同 K_d 的微分策略,在不同时期微分作用的强弱对取得不同的进化速度有明显差异。

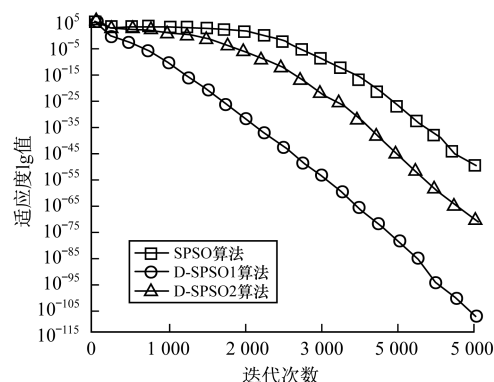


图4 Sphere函数收敛曲线

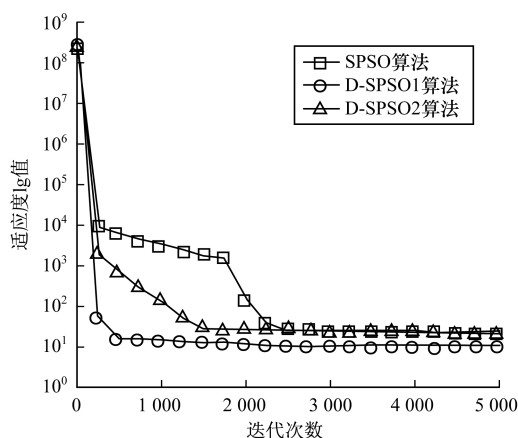


图5 ShiftedRosenbrock函数收敛曲线

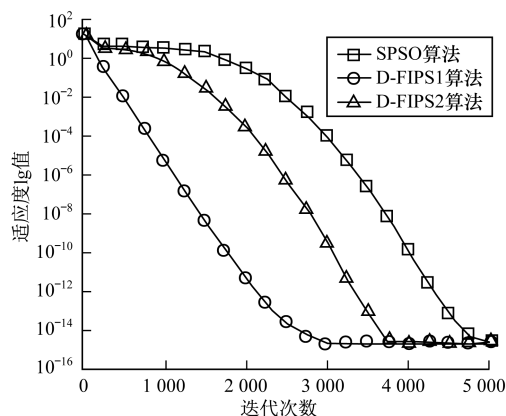


图6 Ackley函数收敛曲线

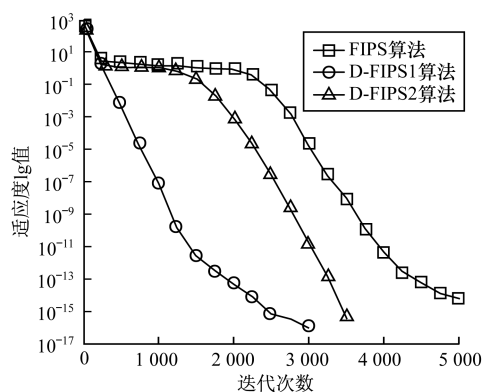


图7 RotatedGriewank函数收敛曲线

4 结束语

本文通过对粒子群优化算法进行轨迹分析,认为其搜索过程为比例-积分(PI)控制策略,通过在标准粒子群及其改进算法中引入微分项,提出微分控制策略的快速粒子群算法。通过对D-SPSO和D-FIPS进行仿真实验对比,认为微分控制策略的改进算法在解决大多数测试函数时能达到预设寻优精度,并且其函数计算次数少、寻优速度快。证明了引入该策略的改进粒子群优化算法在解决进化过程中出现的寻优效率低和寻优速度慢的问题上,取得了良好的效果,而且该策略不仅适用标准粒子群算法,对于改进优化算法也有较好的效果,对大多数粒子群算法都具有适用性,具有良好的应用拓展性。

参考文献

- [1] KENNEDY J, EBERHART R. Particle Swarm Optimization [C]//Proceedings of the 4th IEEE International Conference on Neural Networks. Washington D. C., USA: IEEE Press, 1995: 1942-1948.
- [2] RATNAWEERA A, HALGAMUGE S K, WATSON H C. Self-organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration coefficients [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240-255.
- [3] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.
- [4] KENNEDY J, MENDES R. Population Structure and Particle Swarm Performance [C]//Proceedings of the Evolutionary Computation Congress. Washington D. C., USA: IEEE Press, 2002: 1671-1676.
- [5] CUI Zhihua, CAI Xingjuan. Integral Particle Swarm Optimization with Dispersed Accelerator Information [J]. Fundamenta Informaticae, 2009, 95(4): 427-447.

(下转第281页)

- [4] HU Ruizhen, FAN Lubin, LIU Ligang. Co-segmentation of 3D Shapes via Subspace Clustering [J]. Computer Graphics Forum, 2012, 31(5): 1703-1713.
- [5] BIASOTTI S, CERRI A, BRONSTEIN A, et al. Recent Trends, Applications, and Perspectives in 3D Shape Similarity Assessment [J]. Computer Graphics Forum, 2016, 35(6): 87-119.
- [6] HUANG Qixing, KOLTUN V, GUIBAS L. Joint Shape Segmentation with Linear Programming [J]. ACM Transactions on Graphics, 2011, 30(6).
- [7] van KAICK O, XU Kai, ZHANG Hao, et al. Co-hierarchical Analysis of Shape Structures [J]. ACM Transactions on Graphics, 2013, 32(4).
- [8] MENG Min, XIA Jiazhi, LUO Jun, et al. Unsupervised Co-segmentation for 3D Shapes Using Iterative Multi-label Optimization [J]. CAD Computer Aided Design, 2013, 45(2): 312-320.
- [9] SHU Zhenyu, QI Chengwu, XIN Shiqing, et al. Unsupervised 3D Shape Segmentation and Co-segmentation via Deep Learning [J]. Computer Aided Geometric Design, 2016, 43(3): 39-52.
- [10] TSUCHIE S, HOSINO T, HIGASHI M. High-quality Vertex Clustering for Surface Mesh Segmentation Using Student-t Mixture Model [J]. CAD Computer Aided Design, 2014, 46(1): 69-78.
- [11] van KAICK O, TAGLIASACCHI A, SIDI O, et al. Prior Knowledge for Part Correspondence [J]. Computer Graphics Forum, 2011, 30(2): 553-562.
- [12] DORIGO M, GAMBARELLA L. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.
- [13] 魏东, 吴良杰, 佐丹, 等. 基于混合蚁群算法的网格任务调度 [J]. 计算机工程, 2010, 36(3): 215-217.
- [14] 张耀楠, 杨乐, 康雁. 蚁群优化在超声图像运动矢量估计中的应用 [J]. 东北大学学报(自然科学版), 2012, 33(3): 327-331.
- [15] SHAPIRA L, SHAMIR A, COHEN-OR D. Consistent Mesh Partitioning and Skeletonisation Using the Shape Diameter Function [J]. Visual Computer, 2008, 24(4): 249-259.
- [16] UGAIL H, LIU Lei, SHENG Yun, et al. Graph Cut Based Mesh Segmentation Using Feature Points and Geodesic Distance [C]//Proceedings of 2015 International Conference on Cyberworlds. Visby, Sweden: [s. n.], 2015: 115-120.
- [17] LI Guodong, CHEN Xinjian, SHI Fei, et al. Automatic Liver Segmentation Based on Shape Constraints and Deformable Graph Cut in CT Images [J]. IEEE Transactions on Image Processing, 2015, 24(12): 5315-5329.
- [18] SHI Zhenfeng, LE Dan, YU Liyang, et al. 3D Mesh Segmentation Based on Markov Random Fields and Graph Cuts [J]. IEICE Transactions on Information and Systems, 2012, E95-D(2): 703-706.

编辑 金胡考

(上接第250页)

- [6] 蔡星娟, 崔志华, 曾建潮, 等. 自适应PID控制微粒群算法 [C]//第二十六届中国控制会议论文集. 北京: [出版者不详], 2007.
- [7] ZHANG J, YANG S. A Novel PSO Algorithm Based on an Incremental PID Controlled Search Strategy [J]. Soft Computing, 2016, 20(3): 991-1005.
- [8] 于海生. 计算机控制技术 [M]. 北京: 机械工业出版社, 2007.
- [9] ZHANG Wei, LIU Jiao, FAN Lübin, et al. Control Strategy PSO [J]. Applied Soft Computing, 2016, 38(3): 75-86.
- [10] MENDES R, KENNEDY J, NEVES J. The Fully Informed Particle Swarm: Simpler, Maybe Better [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210.
- [11] 韩璞, 孟丽, 王彪, 等. 粒子群算法中粒子轨迹特性研究 [J]. 计算机仿真, 2015, 32(12): 235-240.
- [12] 张玮, 王化奎. 粒子群算法稳定性的参数选择策略分析 [J]. 系统仿真学报, 2009, 21(14): 4339-4344, 4350.
- [13] 李宁, 孙德宝, 邹彤, 等. 基于差分方程的 PSO 算法粒子运动轨迹分析 [J]. 计算机学报, 2006, 29(11): 2052-2060.
- [14] SHI Yuhui, EBERHART R. Parameter Selection in Particle Swarm Optimization [C]//Proceedings of International Conference on Evolutionary Programming. Berlin, Germany: Springer, 1998: 591-600.
- [15] LIM W H, ISA M N A. Adaptive Division of Labor Particle Swarm Optimization [J]. Expert Systems with Applications, 2015, 42(14): 5887-5903.
- [16] 王永贵, 林琳, 刘宪国. 基于 CGA 和 PSO 的双种群混合算法 [J]. 计算机工程, 2014, 40(7): 148-153.

编辑 顾逸斐