

利用带感知能力的粒子群算法求解约束优化问题

顾宏杰, 许力

(浙江大学 电气工程学院, 杭州 310027)

(guhongjie@zju.edu.cn)

摘要:提出一种求解约束优化问题的改进粒子群优化算法。它利用可行性判断规则处理约束条件,更新个体最优解和全局最优解。通过为粒子赋予自适应感知能力,算法能较好地平衡全局和局部搜索,且有能力跳出局部极值,防止早熟。边界附近粒子的感知结果被用来修正其飞行速度以加强算法对约束边界的搜索。实验结果表明,新算法收敛速度快,寻优能力强,能很好地求解约束优化问题。

关键词:约束优化问题;粒子群优化;自适应感知能力;约束边界

中图分类号: TP18 **文献标志码:** A

Perceptive particle swarm optimization algorithm for constrained optimization problems

GU Hong-jie, XU Li

(College of Electrical Engineering, Zhejiang University, Hangzhou Zhejiang 310027, China)

Abstract: A new Particle Swarm Optimization (PSO) algorithm was proposed to solve the constrained optimization problems. A feasibility-based rule handled constraints to update the individual optimal solutions and the global optimal solutions. Adaptive perceptive ability of particles can balance the global search and the local search. This ability can also help to avoid prematurity. The velocity of particle around the boundary was revised by the results of perceiving to enhance the search around the boundary. The simulation results show that the proposed approach has fast convergence and better optimization ability, and is suitable for solving constrained optimization problems.

Key words: constrained optimization problem; Particle Swarm Optimization (PSO); adaptive perceptive ability; constrained boundary

0 引言

约束优化问题广泛存在于工业生产和日常生活中,是一类满足等式或不等式约束情况下对目标函数进行优化的问题。约束优化问题具有多变量和非线性等特点。

粒子群优化(Particle Swarm Optimization, PSO)^[1]算法简单高效、鲁棒性强,最初用于解决无约束优化问题。为了处理约束优化问题,引入罚函数,将约束优化转化为无约束问题^[2]。但惩罚因子通常动态变化,而且数值难定。为避免使用罚函数,可将约束条件转化为一个或多个目标函数^[3]。Zahara等人将Chootinan等人提出的梯度修正方法^[4]嵌入到NM-PSO中以求解约束优化问题^[5],但若约束条件不可微,该法便不可行。

Deb提出了一种基于遗传算法(Genetic Algorithm, GA)的可行性判断规则^[6]。该规则已被引入PSO中^[7-8],规定可行解优于不可行解,以引导粒子进入可行域。但粒子进入可行域的速度过快,导致对约束边界附近区域的搜索不足。另外,可行解对不可行解的绝对优势将很容易使算法早熟^[7-8]。

为了避免使用罚函数带来的复杂性,本文提出的带自适应感知能力的PSO算法(Perceptive PSO, PPSO)使用可行性判断规则处理约束条件,该规则不引入任何附加参数,简单有效。通过模拟自然界中飞鸟等生物的感知能力,PPSO为粒子赋予自适应感知能力。粒子利用感知到的信息更新局部最优

$PBest$ 与全局最优 $GBest$,平衡全局和局部搜索。为了降低常规PSO具有的早熟风险,PPSO中低速粒子的大范围感知能力将为算法提供跳出局部极值的机制。由于可行性判断规则会降低约束边界附近的粒子数量,削弱边界附近的搜索能力,算法通过加强边界附近粒子的感知能力来弥补不足。同时,将部分欲飞出可行域的粒子约束在可行域内,保证算法对可行域内部的搜索力度,加快收敛速度。实验表明,该算法能兼顾全局搜索与局部搜索,具有较强的寻优能力,并能较好地防止早熟。

1 约束优化问题与粒子群优化算法

1.1 约束优化问题

一般情况下,约束优化问题可描述如下:

$$\text{Min } f(\mathbf{x}) \quad (1)$$

$$\text{s. t. } g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, p$$

$$h_j(\mathbf{x}) = 0, j = 1, 2, \dots, q$$

式中: $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ 为决策向量,维数为 n ; $X = \{\mathbf{x} | x_i \in \mathbf{R}, x_i^L \leq x_i \leq x_i^U\}$ 为搜索空间(x_i^L, x_i^U 为变量 x_i 的下边界和上边界); p 为不等式约束的个数; q 为等式约束的个数。通常将等式约束 $h_j(\mathbf{x}) = 0$,转化为不等式约束 $|h_j(\mathbf{x})| - \varepsilon \leq 0$,其中 ε 为一个略大于零的正数。

1.2 粒子群优化算法

粒子群优化算法是一种模拟鸟群或鱼群在觅食过程中所

收稿日期:2010-07-15;修回日期:2010-08-31。

作者简介:顾宏杰(1986-),男,江苏海门人,硕士研究生,主要研究方向:人工智能、优化算法;许力(1964-),男,江苏无锡人,教授,博士生导师,主要研究方向:智能控制与智能系统、机器人控制、工业自动化。

表现的群体智能的仿生优化算法。群体中的个体(粒子)通过自身的学习和相互之间的信息共享不断调整其飞行速度,最终汇集到最优点。粒子的速度和位置更新如下:

$$\begin{cases} v_i^{t+1} = \omega v_i^t + c_1 r_1 (x_{iPBest}^t - x_i^t) + c_2 r_2 (x_{iGBest}^t - x_i^t) \\ x_i^{t+1} = x_i^t + v_i^{t+1} \end{cases} \quad (2)$$

其中: $i = 1, 2, \dots, M$ (M 为粒子总数); 相关系数包括惯性权重 ω , 可使 ω 随 t 增加而线性递减; 学习因子 c_1 和 c_2 , r_1 和 r_2 是 $[0, 1]$ 内均匀分布的随机数; x_{iPBest}^t 和 x_{iGBest}^t 表示粒子 i 到时间 t 时的个体最优解和种群到时间 t 时的最优解。

2 PPSO 算法

2.1 可行性判断规则

个体最优解和全局最优解的更新对 PSO 非常重要。可将 Deb^[6] 提出的可行性判断规则引入到 PSO 中。

规则 1 如果两个粒子都在可行域内, 那么适应值较高的粒子获胜。

规则 2 如果一个粒子在可行域内, 而另一个在可行域外, 那么可行域内的粒子获胜。

规则 3 如果两个粒子都在可行域外, 那么违反约束程度较小的粒子获胜。

本文将非可行解 $x = \{x_1, x_2, \dots, x_n\}$ 对约束的违反程度定义如下:

$$v = \frac{t}{p+q} \left(\sum_{i=1}^p \max[g_i(x), 0] + \sum_{j=1}^q \max[|h_j(x)| - \varepsilon, 0] \right) \quad (3)$$

式中 t 表示该非可行解违反约束条件的个数。

2.2 为 PSO 中粒子赋予自适应感知能力

常规 PSO 中的粒子在任何时刻只能获取当前位置的适应值, 而自然界的飞鸟等生物都具有一定感知能力, 它们通过视觉和嗅觉等方式感知环境并引导自身行为。受此启发, 本文通过为粒子赋予一定的感知能力而提出具有感知能力的粒子群优化算法 (PPSO)。具体地, 粒子在新位置附近可随机生成多个感知点, 通过与当前位置和个体最优解 x_{iPBest}^t 的比较来更新 x_{iPBest}^t 。通过合理配置粒子感知区域和感知点, 使粒子同时具备较强的全局和局部搜索能力, 并能跳出局部极值, 防止早熟。

本文采用方形感知区域, 针对可行域内粒子、可行域外粒子及欲飞出可行域的边界粒子, 分别采取不同的感知策略, 粒子的感知区域及感知点随粒子的位置及速度而不断变化, 以适应不同的“环境”。

由规则 2 可知, 可行解优于不可行解。因此, 种群最优位置 x_{GBest}^t 通常在可行域内。可行域外粒子与 x_{GBest}^t 之间的距离相对较大, x_{GBest}^t 与 x_{iPBest}^t 相对差亦较大。因此, 这些粒子飞行速度通常过快, 易快速越过某局部区域而未对其搜索。可行域外粒子具有感知能力后, 感知点使得单个粒子搜索范围从一个点扩大到一个小区, 增强了局部搜索能力, 大大增加了发现新可行域的概率。粒子的感知区域与其速度及当前位置有关。当粒子以速度 $v_{out} = \{v_{out}^1, v_{out}^2, \dots, v_{out}^n\}$ 到达可行域外某个位置 $x_{out} = \{x_{out}^1, x_{out}^2, \dots, x_{out}^n\}$ 时, 感知区域 Ω_{out} 定义如下:

$$\begin{cases} l = \max(v_{out}^1, v_{out}^2, \dots, v_{out}^n) \\ \Omega_1 = \{x \mid x_{out}^i - l/2 \leq x_i \leq x_{out}^i + l/2\} \\ \Omega_{out} = \Omega_1 \cap X \end{cases} \quad (4)$$

感知点的数量 k 与方格边长 l 及维数 n 关系如下:

$$k = \min[\text{fix}(5 \times l \times n) + 1, 50] \quad (5)$$

式中: $\text{fix}()$ 表示截尾取整函数, k 满足 $1 \leq k \leq 50$ 。显然, 粒子速度越快, 感知区域就越大, 感知点越多, 感知能力也越强。

可行域内的粒子负责搜索全局最优解, 它们违反约束的程度为零。由可行性判断规则可知, 违反约束的程度越小, 粒子越容易聚拢于 x_{GBest} 而无法跳出 x_{GBest} 附近区域^[11], 算法易陷入局部极值。为可行域内粒子赋予感知能力后, 用感知点的信息更新 x_{iPBest} 将为粒子群带来新的有价值信息, 优化粒子的飞行轨迹。可行域内粒子 $x_{in} = \{x_{in}^1, x_{in}^2, \dots, x_{in}^n\}$ 的感知区域 Ω_{in} 定义如下:

$$\begin{cases} l_i = \frac{x_{in}^U - x_{in}^L}{1 + 5 \times |v_{in}| \times n}; i = 1, 2, \dots, n \\ l = \max(l_1, l_2, \dots, l_n) \\ \Omega_1 = \{x \mid x_{in}^i - l_i/2 \leq x_i \leq x_{in}^i + l_i/2\} \\ \Omega_{in} = \Omega_1 \cap \Omega_2 \end{cases} \quad (6)$$

其中: $|v_{in}|$ 为速度向量的模; n, x_{in}^L, x_{in}^U 如式 (1) 中所述; Ω_2 表示包围已知可行域的最小方形区域。感知点数量 k 仍由式 (5) 确定。显然, 粒子速度大, 感知区域小, 有利于加强局部搜索; 粒子速度小, 感知区域大, 有利于加强全局搜索, 算法将同时具备较强的全局、局部搜索能力。当算法陷入局部极值时, 绝大部分粒子的速度 $|v_{in}|$ 接近于零, $\Omega_1 \approx \Omega_2$, 粒子将感知整个已知可行域, 若感知点的适应值优于局部极值, 则 x_{GBest} 将被更新。在新的 x_{GBest} 的吸引下, 粒子便能跳出局部极值。

粒子快速进入可行域, 导致约束边界附近区域粒子数过少, 不利于对该区域的搜索。为改善对约束边界附近的搜索, 本文利用感知结果修正粒子的飞行速度, 将部分欲飞出可行域的粒子约束在可行域内, 保证可行域内的粒子数, 提高寻优速度。具体的, 对于试图飞出可行域的粒子, 本文将以一定概率 $\rho = 0.8$ 感知其附近区域并将其约束在可行域内, 粒子飞出可行域的概率为 $1 - \rho = 0.2$ 。飞出的粒子可重新搜索新的可行域。

若粒子试图以速度 $v^{t+1} = \{v_1^{t+1}, v_2^{t+1}, \dots, v_n^{t+1}\}$ 从可行域内位置 x^t 飞出到 x^{t+1} , 则感知区域为以 x^t, x^{t+1} 的中点为中心, $d = \max[v_1^{t+1}, v_2^{t+1}, \dots, v_n^{t+1}]$ 为边长的方格, 增加感知点数量至:

$$k = \min[\text{fix}(10 \times d \times n) + 1, 100] \quad (7)$$

感知点的密度大于非边界区域以加强感知效果。若最优感知点 x_b^{t+1} 的适应值优于 x^t , 则修改粒子速度 $v_b^{t+1} = x_b^{t+1} - x^t$, 粒子飞向 x_b^{t+1} 而非可行域外的 x^{t+1} ; 否则, 令 $x^{t+1} = x^t$, 粒子停在 x^t 处, 等待下次迭代。

2.3 PPSO 算法步骤

PPSO 算法以具有自适应感知能力的粒子群优化算法为主干, 利用可行性判断规则比较粒子所在位置及感知点处的适应度, 以更新粒子的个体最佳位置 x_{iPBest} 和全局最佳位置 x_{GBest} , 对飞出可行域的粒子以概率 ρ 执行边界感知策略, 并允许小部分粒子飞出可行域探索新区域。算法流程如图 1。

3 实验与分析

3.1 测试对象的描述

$$\text{测试对象 1: } \text{Min } f(x) = -\frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$$

$$\begin{aligned} \text{s. t. } g_1(\mathbf{x}) &= x_1^2 - x_2 + 1 \leq 0 \\ g_2(\mathbf{x}) &= 1 - x_1 + (x_2 - 4)^2 \leq 0 \\ 0 &\leq x_1 \leq 10, 0 \leq x_2 \leq 10 \\ \text{最优解 } \mathbf{x}^* &= \{1.227\ 971\ 3, 4.245\ 373\ 3\}, f(\mathbf{x}^*) = \\ &-0.095\ 825. \end{aligned}$$

测试对象2: $\text{Min } f(\mathbf{x}) = x_1 + x_2 + x_3$

$$\begin{aligned} \text{s. t. } g_1(\mathbf{x}) &= -1 + 0.0025(x_4 + x_6) \leq 0 \\ g_2(\mathbf{x}) &= -1 + 0.0025(x_5 + x_7 - x_4) \leq 0 \\ g_3(\mathbf{x}) &= -1 + 0.01(x_8 - x_5) \leq 0 \\ g_4(\mathbf{x}) &= -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0 \\ g_5(\mathbf{x}) &= -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0 \\ g_6(\mathbf{x}) &= -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0 \\ 100 &\leq x_1 \leq 10000, 1000 \leq x_i \leq 10000; i = 2, 3 \\ 10 &\leq x_i \leq 1000; i = 4, \dots, 8 \\ \text{最优解 } \mathbf{x}^* &= \{579.316\ 7, 1\ 359.943\ 5\ 110.071, \\ &182.0174, 295.598\ 5, 217.979\ 9, 286.416\ 2, 395.597\ 9\}, \\ f(\mathbf{x}^*) &= 7049.3307. \end{aligned}$$

测试对象3: Keane's Bump^[9]。

Bump 函数是一个非线性多峰函数,其复杂程度随着维数增加而急剧增加,很难优化,而且其最优解在约束边界附近,不易获得。本文测试时, $n = 20$, 目前已知最优解 $f_{\text{Best}} = -0.803\ 619$ 。由于维数较高,有必要增加粒子数及迭代次数: $M = 500, \text{Iter} = 600$ 。Keane's Bump 描述如下:

$$\text{Min } f(\mathbf{x}) = \frac{-\left|\sum_{i=1}^n \cos^4(x_i) - 2\prod_{i=1}^n \cos^2(x_i)\right|}{\sqrt{\sum_{i=1}^n ix_i^2}}$$

$$\text{s. t. } g_1(\mathbf{x}) = 0.75 - \prod_{i=1}^n x_i \leq 0$$

$$g_2(\mathbf{x}) = \sum_{i=1}^n x_i - 7.5n \leq 0$$

$$0 \leq x_i \leq 10, 1 \leq i \leq n$$

测试对象4: 压力罐设计问题^[10]。

压力罐罐体是长度为 L 的圆柱体,内半径为 R ,壁厚为 T_s ; 两头均是壁厚为 T_h 半球体。设计目标是在满足耐压为 3000 psi, 最小容积为 750 ft³ (1 ft = 30.48 cm) 的要求下,使材料、成形及焊接成本最低。令 $\mathbf{x} = [T_s, T_h, R, L]$, 可建立模型如下:

$$\text{Min } f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

$$\text{s. t. } g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(\mathbf{x}) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(\mathbf{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$$

$$g_4(\mathbf{x}) = x_4 - 240 \leq 0$$

$$0 \leq x_1, x_2 \leq 99, 10 \leq x_3, x_4 \leq 200$$

测试对象5: 横梁焊接问题^[11]。

横梁焊接示意图见文献[11], 横梁宽度为 t , 厚度为 b , 总长为 $l + L$, 焊缝长度为 l , 横梁露出基板长度为 $L = 14\text{ in}$ (1 in = 2.54 cm), 焊缝宽度为 h 。应满足焊缝剪应力、横梁弯曲应力、压曲临界负载、横梁顶端变位等约束。令 $\mathbf{x} = [h, l, t, b]$, 可建立模型如下:

$$\text{Min } f(\mathbf{x}) = 1.10471x_2^2 + 0.04811x_3x_4(14.0 + x_2)$$

$$\text{s. t. } g_1(\mathbf{x}) = \tau(\mathbf{x}) - 13600 \leq 0$$

$$g_2(\mathbf{x}) = \sigma(\mathbf{x}) - 30000 \leq 0$$

$$g_3(\mathbf{x}) = x_1 - x_4 \leq 0$$

$$g_4(\mathbf{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$$

$$g_5(\mathbf{x}) = 0.125 - x_1 \leq 0$$

$$g_6(\mathbf{x}) = \delta(\mathbf{x}) - 0.25 \leq 0$$

$$g_7(\mathbf{x}) = P - P_c(\mathbf{x}) \leq 0$$

$$0.1 \leq x_1, x_4 \leq 2, 0.1 \leq x_2, x_3 \leq 10$$

$$\text{其中 } \tau(\mathbf{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' =$$

$$\frac{QR}{J}, Q = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, J =$$

$$2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \sigma(\mathbf{x}) = \frac{6PL}{x_4x_3^2}, \delta(\mathbf{x}) = \frac{4PL^3}{Ex_3^3x_4},$$

$$P_c(\mathbf{x}) = \frac{4.013E\sqrt{\frac{x_3^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), P = 6000, L = 14,$$

$$E = 30 \times 10^6, G = 12 \times 10^6.$$

测试对象6: 压力弹簧设计问题^[12]。

压力弹簧钢丝直径为 d , 簧圈平均直径为 D , 弹簧有效匝数为 N 。约束条件与弹簧形变、剪应力、自振频率及外直径有关^[12], 设计目标是在满足这些约束的前提下, 使材料成本最低。令 $\mathbf{x} = [d, D, N]$, 建立模型如下:

$$\text{Min } f(\mathbf{x}) = (x_3 + 2)x_2x_1^2$$

$$\text{s. t. } g_1(\mathbf{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$$

$$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0$$

$$g_3(\mathbf{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

$$g_4(\mathbf{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

$$0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15$$

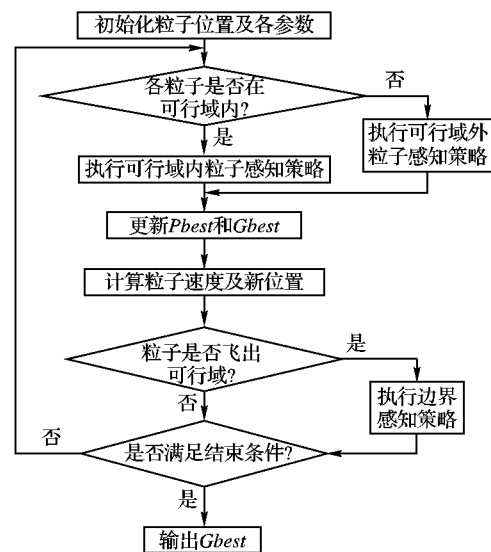


图1 算法流程

3.2 实验参数设置

本文对6个经典对象进行了测试。在标准 PSO 中加入边界感知策略后形成算法 PPSO1, 此时只有部分欲飞出可行域的粒子具有感知能力; 为所有粒子赋予感知能力后形成本文最终提出的算法 PPSO。分别利用标准 PSO、PPSO1、PPSO

优化测试对象 1~3,以证明本文算法的有效性。通过仿真对象 4~6,比较 PPSO 与其他算法,以体现本文算法的性能优势。除非特别说明,参数统一选取如下:粒子数 $M = 100$,最大惯性权重 $\omega_{\max} = 0.9$,最小惯性权重 $\omega_{\min} = 0.4$,学习因子 $c_1 = c_2 = 2$,采用 Matlab 进行编程。

3.3 PSO、PPSO1 与 PPSO 的比较

分别利用标准 PSO、PPSO1、PPSO 优化测试对象 1~3,以检验加入感知能力后的效果,各运行 20 次。为了形象直观地比较三种算法的收敛及优化性能,本文给出了各算法对测试对象 1~3 的最佳收敛曲线,如图 2~4 所示。由图可知,PPSO1 的收敛速度及优化精度均优于标准 PSO,当最优解位于约束边界附近时(测试对象 2,3),这种优势更加明显;而当最优解离约束边界较远时(测试对象 1),试图飞出可行域的粒子数较少,PPSO1 效果并不明显。

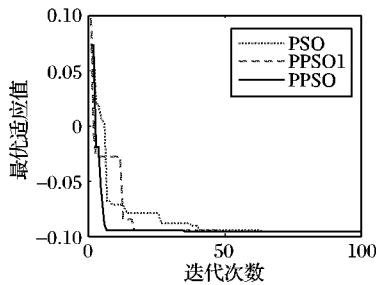


图 2 测试对象 1 收敛曲线

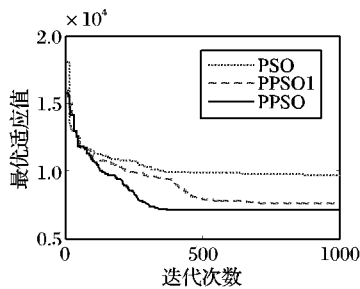


图 3 测试对象 2 收敛曲线

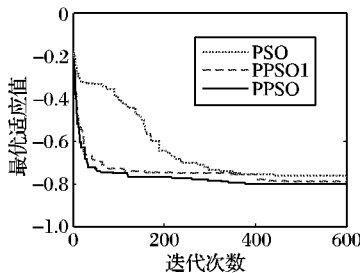


图 4 测试对象 3 收敛曲线

PPSO 利用粒子感知到的信息更新 x_{GBest} 使算法能及时跳出局部极值,减少无效迭代次数。PPSO 测试对象 1~3 时,均未陷于局部极值,仿真结果见表 1。表 1 列出了 20 次运行后的最优值、最差值、平均值及标准误差。PPSO 中粒子的感知点可看做“卫星粒子”,但不需计算飞行速度,不需更新位置,其计算复杂度大大低于标准 PSO 中的粒子。有了这些粒子后,可适量减少粒子总数(PPSO 中粒子总数为 PSO、PPSO1 的一半)。从时间复杂度分析,利用简单的感知点“替代”标准 PSO 中粒子,减少无效迭代次数均能有效降低时间复杂度。仿真结果表明,PPSO 的收敛速度、搜索精度均大大优于标准 PSO,也优于 PPSO1。

3.4 PPSO 与其他算法的比较

对象 4~6 是三个工程优化问题,被广泛用于测试约束优

化算法的性能^[5,7-8,13-14]。本文将每个对象都运行 20 次,记录每一次运行结果,并计算标准误差。各对象最优结果及对应决策变量列于表 2 中。表 3 分别列出各对象的最优、最差结果及标准误差并与其他算法结果进行比较。仿真结果表明,PPSO 对三个工程对象都取得了较好的结果,标准误差较小,稳定性较好。尤其是测得对象 4 的最优值 $f_{Best}(x) = 5885.3328$, 优于相比较的其他算法所获结果。算法迭代次数少,收敛速度快,具有很强的全局、局部搜索能力,有较高的实际应用价值。

表 1 对象 1~3 仿真结果

对象	最优值	最差值	平均值	标准差
1	-0.95825	-0.95824	-0.95825	2.1385e-6
2	7049.3845	7056.1654	7052.2437	1.8654
3	-0.803619	-0.802774	-0.803179	2.4046e-4

表 2 对象 4~6 最优仿真结果

对象	$f_{Best}(x)$	x_1	x_2	x_3	x_4
4	5885.3328	0.778169	0.384649	40.319619	200.0000
5	1.724852	0.205730	3.470489	9.036624	0.205730
6	0.0126652	0.0516838	0.356592	11.296367	—

表 3 对象 4~6 仿真结果及比较

对象	算法	最优值	最差值	平均值	标准差
4	NM-PSO ^[5]	5930.3137	5960.0557	5946.7901	9.1614
	CPSO ^[13]	6061.0777	6363.8041	6147.1332	86.4545
	HPSO ^[8]	6059.7143	6288.6770	6099.9323	86.2022
	CDE ^[14]	6059.7340	6371.0455	6085.2303	43
	PSO-DE ^[7]	6059.7143	6059.7143	6059.7143	1.0e-10
	PPSO	5885.3328	5906.0036	5898.2604	4.0583
5	NM-PSO ^[5]	1.724717	1.733393	1.726373	0.003497
	CPSO ^[13]	1.728024	1.782143	1.748831	0.012926
	HPSO ^[8]	1.724852	1.814295	1.749040	0.040049
	CDE ^[14]	1.733461	1.824105	1.768158	2.2e-02
	PSO-DE ^[7]	1.7248531	1.7248811	1.7248579	4.1e-06
	PPSO	1.724852	1.724855	1.724853	8.3e-07
6	NM-PSO ^[8]	0.0126302	0.012633	0.0126314	8.7375e-07
	CPSO ^[13]	0.0126747	0.012924	0.0127300	5.1985e-05
	HPSO ^[8]	0.0126652	0.0127191	0.0127072	1.5824e-05
	CDE ^[14]	0.0126702	0.012790	0.012703	2.7e-05
	PSO-DE ^[7]	0.0126652	0.0126653	0.0126652	1.2e-08
	PPSO	0.0126652	0.0126679	0.0126661	8.1e-07

4 结语

本文提出一种处理约束优化问题的新型算法 PPSO。算法利用可行性判断规则处理约束条件;为 PPSO 中粒子赋予自适应感知能力并合理设置感知区域,使可行域外粒子能更好地搜索可行域,可行域内粒子具备较强的全局和局部搜索能力并能及时跳出局部极值,防止早熟。同时,边界感知策略加强了对边界附近区域搜索。实验结果表明,算法具有较强的寻优能力,高效稳定,并显著降低了陷入局部最小的概率。进一步的研究将着重于粒子感知能力的改善和感知效率的提高。

参考文献:

[1] KENNEDY J, EBERHART R C. Particle swarm optimization[C]// Proceedings of the IEEE International Conference on Neural Networks. New York: IEEE, 1995: 1942-1948.

本文使用基于 VSM 的传统话题追踪方法与前面所提出的方法进行对比,结果如表 1 所示。

表 1 本文和 BaseLine 实验结果

话题	相关 帖子数	不相关 帖子数	基于 VSM 的 TTT 方法			本文的方法		
			召回率/%	准确率/%	F1-Measure/%	召回率/%	准确率/%	F1-Measure/%
范跑跑事件	90	600	84.44	73.79	78.76	88.89	98.77	93.57
周久耕事件	95	600	93.68	90.82	92.23	89.47	95.51	92.39
三鹿奶粉事件	95	600	96.84	32.86	49.07	86.32	87.23	86.77
周老虎事件	99	600	96.84	85.19	90.64	90.91	90.91	90.91

本实验中,由于部分话题所包含的帖子中描述话题的特征与新闻报道比较一致,因此造成对于部分话题,基于 VSM 的 TTT 方法召回率比较高,但是与本文的方法相比仍稍差。尤其在准确率上,基于 VSM 的 TTT 方法性能不是很稳定,对数据的依赖性比较强。经过分析实验结果发现,使用基于 VSM 的 TTT 方法追踪到的无关贴确实是因为话题模型中的大量无关特征造成的。另外,对比两者的综合评价指标 *F1-Measure*,本文的方法也优于基于 VSM 的 TTT 方法。

5 结语

本文利用论坛帖子的特点研究了论坛话题追踪,较好地解决了帖子短文本特性对话题追踪的影响。本文方法对关键词表的准确度要求较高,当话题重心发生漂移时须更新相关报道并重新构建关键词表,否则可能造成后续帖子的漏检。此外,本文对分词效果要求较高,需进行必要的未登录词识别。未来的工作主要集中在以下几点:首先,在现有方法的基础上加入对话题重心漂移的研究;其次,通过获取话题关键词,实现对话题在论坛上的发展演变趋势的分析。

参考文献:

- [1] 陈映. BBS 与主流报纸的议题互动[D]. 广州:暨南大学, 2005.
- [2] MAKKONEN J. Semantic classes in topic detection and tracking [D]. Helsinki: Department of Computer Science, University of Helsinki, 2009.
- [3] 焦健, 翟有利. 知网的话题更新与跟踪算法研究[J]. 北京交通大学学报, 2009, 30(5): 132-136.
- [4] ZHENG W, ZHANG Y, HONG Y, *et al.* Topic tracking based on keywords dependency profile[C]// Proceedings of the 4th Asia Information Retrieval Symposium. Berlin: Springer-Verlag, 2008: 129-140.
- [5] 任晓东, 张永奎, 薛晓飞. 基于 K-Modes 聚类的自适应话题追踪技术[J]. 计算机工程, 2009, 35(9): 222-224.
- [6] 林鸿飞, 宋丹, 杨志豪. 基于语义框架的话题追踪方法[C]//中国中文信息学会二十五周年学术会议. 北京: 清华大学出版社, 2006: 383-384.
- [7] ZHU MINGLIANG, HU WEIMING, WU OU. Topic detection and tracking for threaded discussion communities[C]// Proceedings of the 2008 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. Washington: IEEE Computer Society, 2008: 77-83.
- [8] 杨洁, 季铎, 蔡东风, 等. 基于联合权重的多文档关键词抽取技术[J]. 中文信息学报, 2008, 22(6): 75-79.
- [9] 邱立坤, 程巍, 龙志祯, 等. 面向 BBS 的话题挖掘初探[C]// 全国第八届计算语言学联合学术会议. 北京: 清华大学出版社, 2005: 401-407.
- [10] 刘群, 李素建. 基于知网的词汇语义相似度的计算[C]// 第三届汉语词汇语义学研讨会. 台北: [出版者不详], 2002: 59-76.
- [1] PARSPOULOS K E, VRAHATIS M N. Particle swarm optimization method for constrained optimization problems[C]// Proceedings of the 2nd Euro-International Symposium on Computational Intelligence. Slovakia: IOS, 2002: 214-220.
- [2] VENTER G, HAFTKA R T. Constrained particle swarm optimization using a bi-objective formulation[J]. Structural and Multidisciplinary Optimization, 2010, 40(1/6): 65-76.
- [3] CHOOTINAN P, CHEN A. Constraint handling in genetic algorithms using a gradient-based repair method[J]. Computers and Operations Research, 2006, 33(8): 2263-2281.
- [4] ZAHARA E, KAO Y T. Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems[J]. Expert Systems with Applications, 2009, 36(2): 3880-3886.
- [5] DEB K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2/4): 311-318.
- [6] LIU HUI, CAI ZIXING, WANG YONG. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization[J]. Applied Soft Computing, 2010, 10(2): 629-640.
- [7] HE QIE, WANG LING. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization[J]. Applied Mathematics and Computation, 2007, 186(2): 1407-1422.
- [8] KEANE A J. Experiences with optimizers in structural design[EB/OL]. [2010-02-10]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.7516>.
- [9] KANNAN B K, KRAMER S N. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design[J]. Journal of Mechanical Design, 1994, 116(2): 405-411.
- [10] RAO S S. Engineering optimization[M]. 4th ed. Hoboken: John Wiley and Sons, 2009.
- [11] ARORA J S. Introduction to Optimum design[M]. 2nd ed. San Diego: Elsevier Academic Press, 2004.
- [12] HE QIE, WANG LING. An effective co-evolutionary particle swarm optimization for constrained engineering design problems[J]. Engineering Applications of Artificial Intelligence, 2007, 20(1): 89-99.
- [13] HUANG FUZHUO, WANG LING, HE QIE. An effective co-evolutionary differential evolution for constrained optimization[J]. Applied Mathematics and Computation, 2007, 186(1): 340-356.

(上接第 88 页)