

文章编号: 1001-0920(2005)04-0474-04

旋转曲面变换 PSO 算法解非线性最优控制问题

熊 勇^{1,2}, 路文初¹, 刘继忠¹, 胡上序¹

(1. 浙江大学 智能信息工程研究所, 浙江 杭州 310027; 2 武汉大学 自动化系, 武汉 430081)

摘 要: 针对利用粒子群优化算法进行多极值点函数优化时, 存在陷入局部极小点和搜寻效率低的问题, 提出旋转曲面变换方法, 将被优化函数映射到一个同胚曲面上, 它将当前局部极小点变换为全局最大点, 并保持被优化函数值在当前局部极小点以下部分的形状不变, 从而克服陷入局部极小点的问题, 最后将其用于解一个非线性系统的最优控制问题, 实验结果证明了该方法的可行性和有效性。

关键词: 粒子群; 旋转曲面变换; 局部极小; 非线性; 最优控制

中图分类号: TP183

文献标识码: A

Rotate surface transformation PSO algorithm for solving nonlinear optimal control problem

XIONG Yong^{1,2}, LU Wen-chu¹, LIU Ji-zhong¹, HU Shang-xu¹

(1. Institute of Intelligent Information Engineering, Zhejiang University, Hangzhou 310027, China; Department of Automation, Wuhan University, Wuhan 430072, China. Correspondent: XIONG Yong, E-mail: bear-brave@sohu.com)

Abstract: Particle swarm optimization (PSO) algorithm is easy to be trapped into local minima in optimizing multimodal function. Rotate surface transformation (RST) method is proposed to overcome the defect. RST method transforms local minimum to global maximum and keeps the values of the optimized function unchanged under the local minimum. A nonlinear optimal control problem regarding ferry ship is employed for examining the validity of the proposed method. Experimental results show that the method is effective.

Key words: particle swarm; rotate surface transformation; local minimum; nonlinearity; optimal control

1 引 言

粒子群优化(PSO)算法是近年来出现的一种全局优化算法。Kennedy 和 Eberhart^[1,2]从动物群体的觅食行为受到启示,发现当整个群体在搜寻某个目标时,对于其中的某个个体,它通常参照群体中目前处于最优位置的个体和自身曾经达到的最优位置调整下一步的搜寻,从而设计了粒子群优化算法。

PSO 是一种随机、并行的优化算法。它不要求被优化函数具有可微、可导、连续等性质,算法简单。但对于有多个局部极值点的函数,容易陷入到局部极值点中,对此本文提出了旋转曲面变换方法,将当前局部极小点变换为全局最大点,并保持被优化函

数值在当前局部极小点以下部分的形状不变。

2 基本 PSO 算法

基本 PSO 算法迭代公式如下:

$$v_{id}(t+1) = v_{id}(t) + c_1 \phi(p_{id}(t) - x_{id}(t)) + c_2 \phi(p_{gd}(t) - x_{id}(t)), \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1). \quad (2)$$

其中: i 和 d 分别为粒子和维数的序号, v 和 x 分别为每个粒子的速度和位置, t 为迭代计算的步数, c_1 和 c_2 为系统的控制参数, ϕ 和 ϕ_2 均为 $(0, 1)$ 区间上的随机数, p_{id} 为粒子 i 到目前得到的最好优化结果的位

收稿日期: 2004-07-05; 修回日期: 2004-09-16

作者简介: 熊勇(1976—),男,湖北黄冈人,博士生,从事智能信息处理的研究; 胡上序(1934—),男,浙江杭州人,教授,博士生导师,从事过程控制、智能信息处理的研究。

置, p_{gd} 为到目前整个粒子群体得到的最好优化结果所对应粒子的位置

3 基于旋转曲面变换的粒子群优化算法

3.1 旋转曲面变换

本文方法的基本原理是: 监测每个粒子对应的当前被优化函数值, 当粒子陷入局部极值点时, 利用一个变换改变局部极小值邻近的函数形状, 使得变换后的函数相对于原函数更容易优化, 同时又避免陷入局部极值点, 从而改善基本 PSO 算法的效果

在寻找被优化函数 $f(x)$ 的全局极小时, 每当粒子达到某个极小点 $f(x_1)$, 可假设它为一个局部极小点, 为跳出该点且不改变原函数的全局极值点, 变换应满足如下条件:

- 1) 此变换必须使所有的粒子迅速离开 $f(x)$ 在 $f(x_1)$ 的区域;
- 2) 此变换必须使所有 $f(x) < f(x_1)$ 的区域函数值不变

符合上述要求的变换有多种, 为了使该变换尽可能简单, 以减少计算时间, 本文采用一种旋转曲面变换, (即 h 变换), 并采用基于此变换的 hPSO 方法. 这个变换将原局部极小点 x_1 设为一个旋转曲面 $g(x)$ 的顶点, 将原函数中大于 $f(x_1)$ 的点映射到旋转曲面上, 并保持所有小于 $f(x_1)$ 点的函数值不变, 其变换式为

$$g(x) = f(x_1) + (\text{sign}(f(x) - f(x_1)) + 1)H(m_1, m_2, x - x_1) + (f(x) - f(x_1))(\text{sign}(f(x) - f(x_1)) - 1) / -2, f(x) > f(x_1); \quad (3)$$

$$g_1(x) = \lim_{x \rightarrow x_1^+} g(x) = f(x_1) + 2H(m_1, m_2, 0), f(x) < f(x_1). \quad (4)$$

其中 H 表示一个旋转曲面函数, 它可以是任意一个平面曲线绕一轴线旋转而成, 例如 H 可取锥面

$$H(x) = d(x_1, m_1) - k(x_1, m_2) |x - x_1|.$$

其中: $d(x_1, m_1)$ 和 $k(x_1, m_2)$ 表示由 x_1, m_1 和 m_2 决定的两个正值常数; m_1 和 m_2 为参数值, 用于确定此旋转曲面距 $f(x_1)$ 的高度及其张角; $|x - x_1|$ 表示向量 $x - x_1$ 的模, 即 x 到 x_1 的欧氏距离; sign 为符号函数, 即

$$\text{sign}(x) = \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ -1, & x < 0 \end{cases} \quad (5)$$

旋转曲面变换方法与 TA 在 (Tunneling Algorithm) 原理上有相似之处, 当寻找全局最小点时, 如果出现陷入局部极值问题, 二者均通过对被优化函数施行某种变换, 使局部极值点完全消失或减

弱. 但两种方法的变换形式和实施原则不同, 本文方法仅对被优化函数大于局部极小点的部分进行变换, TA 则整个函数进行变换

3.2 参数 m_1 和 m_2 的确定

为了使变换满足上述要求, 变换后的旋转曲面必须能够覆盖整个搜索空间, 因此高度和张角要满足一定的要求. 以锥面为例, m_1 表示锥面顶点到 x_1 点的距离, a 为锥面的半顶角, 锥体底面半径 R 为搜索区域中相距最远两点间的长度. 令 $m_2 = \cos a$, $m_{\min} = R \text{ctg} a$, 则

$$m_{\min} = R m_2 / \sqrt{1 - m_2^2} \quad (6)$$

其中 m_2 表示旋转曲面的倾斜程度, 可根据具体问题适当选择. m_2 一旦确定, 则 m_1 不得小于由式 (6) 确定的 m_{\min} 值

3.3 粒子到达边界的处理

在实行变换后, 除了原来小于 $f(x_1)$ 的区域可能陷入局部极小外, 还有可能陷入局部极小点的是搜寻区域的边界附近, 因为那里的梯度很小. 为了避免出现这种情况, 当粒子超过边界或正好到达边界时, 启动一个随机的调整步骤, 使之返回到搜索区域. 令

$$x = h_1 + \text{rand}(h_2 - h_1).$$

其中: rand 为 $(0, 1)$ 之间的随机数, h_2 和 h_1 为搜寻区域的上界和下界

3.4 hPSO 的收敛性分析

文献 [3] 证明了当式 (1) 和 (2) 的系数满足一定条件时, 算法将收敛到局部极小点. 假如函数在所搜寻的范围内只有一个极小点, 则必定全局收敛. 对于在搜寻范围内只有有限个局部极值点的函数, 通过不断进行 h 变换, 最后只得到一个最小的极值点, 即为全局最小点

4 轮船渡河的最优控制问题

设有一条船横渡强水流的河, 河两岸的距离为 h , 船相对于水流的速度为常数 v_0 . 以船出发点为原点建立直角坐标系, x 和 y 分别表示船在水平和垂直方向的位移, M 表示船的航向与 x 方向的夹角, x , y 和 M 均为时间 t 的函数. 设水流在水平和垂直方向上的流速分别为 $\mu(x, y)$ 和 $\omega(x, y)$, μ 和 ω 均为某种形式的函数. 驾驶员希望以最短的航程到达河对岸, 因此需要寻找一种优化的驾驶方法, 即找出优化的航向角 M 随时间变化的函数 $M(t)$.

该控制系统的方程为

$$\dot{x}(t) = v_0 \cos M + \mu(x, y), \quad (7)$$

$$\dot{y}(t) = v_0 \sin M + \omega(x, y). \quad (8)$$

其中: x, y 为状态变量, M 为控制变量. 本文取

$$\mu(x, y) = a_1 \exp(a_2 x - a_3 (y - (h/2))^2), \quad (9)$$

$$\omega(x, y) = \text{sign}(y - (h/2)) a_4 \exp(a_5 x - a_6 (y - (h/2))^2), \quad (10)$$

其中 $a_1 \sim a_6$ 为参数

为达到使总航程最短的目的, 根据曲线长度的积分公式, 必须使如下指标最小:

$$J[M(t)] = \int_{t_0}^{t_f} (x^2 + y^2)^{1/2} dt \quad (11)$$

其中: t_0 为起始时间, t_f 为到达对岸的时间

将式(7)和式(8)代入式(11)得

$$J(M(t)) = \int_{t_0}^{t_f} (v_0^2 + 2v_0(\mu \cos M + \omega \sin M) + \mu^2 + \omega^2)^{1/2} dt \quad (12)$$

此外, 根据问题的实际意义, 初始时刻和末端时刻应分别满足如下约束:

$$x(t_0) = x_0, y(t_0) = y_0, \quad (13)$$

$$y(t_f) = h \quad (14)$$

其中 x_0 和 y_0 为初始时刻的位置, 整个问题的完整描述如下:

受控系统(7)和(8)在满足始态和终态约束(13)和(14)的条件下, 求控制函数 $M(t)$, 使得关于 $M(t)$ 的泛函指标 $J[M(t)]$ 最小. 因为系统是非线性的, 如果采用通常的变分方法, 则必须要解非线性的偏微分方程, 计算量很大^[4,5]. 本文考虑利用 hPSO 方法得到数值解

5 用 hPSO 方法解非线性最优控制问题

为了采用 hPSO 方法求解上述问题, 需要将目标泛函离散化, 从而使其转化为普通的多元函数, 即

$$J(M(t)) = \int_{t_0}^{t_f} (v_0^2 + 2v_0(\mu \cos M(t) + \omega \sin M(t)) + \mu^2 + \omega^2)^{1/2} \Delta t \quad (15)$$

计算 $J(M)$ 时, 事先不能确定式(15)右端的结束时间 t_f , 若用 t 作自变量不便于计算. 故将河岸之间的距离 h 均分成 n 段, i 表示第 i 次调整航向, M 和对应的相邻两次调整之间的时间差 Δt 均为 i 的函数, 分别用 $M(i)$ 和 $\Delta t(i)$ 表示. 利用欧拉法将式(7)和式(8)离散化得到

$$\Delta y = (v_0 \sin M(i) + \omega(x(i), y(i))) \Delta t(i), \quad (16)$$

$$x(i+1) = x(i) + (v_0 \cos M(i) + \mu(x(i), y(i))) \Delta t(i), \quad (17)$$

$$y(i+1) = y(i) + \Delta y = y(i) + h/n. \quad (18)$$

因此

$$J(M(t)) = \sum_{i=1}^n (v_0^2 + 2v_0(\mu \cos M(i) + \omega \sin M(i)) + \mu^2 + \omega^2)^{1/2} \Delta t(i)$$

$$\omega \sin M(i) + \mu^2 + \omega^2)^{1/2} \Delta t(i). \quad (19)$$

$J(M)$ 的计算步骤如下:

1) 在给出初始位置 x_0, y_0 和其他参数后, 根据式(16)计算 $\Delta t(i)$;

2) 将 $\Delta t(i)$ 代入式(17)计算 $x(i+1)$;

3) 根据式(18)计算 $y(i+1)$;

4) 将 $x(i+1)$ 和 $y(i+1)$ 代入式(16)计算 $\Delta t(i+1)$, 依次递推, 直至得到所有的 $x(i), y(i), \Delta t(i), i = 1, 2, \dots, n$, 将它们代入式(19)相加即得 $J(M)$;

5) 利用 hPSO 方法对该函数进行优化, 即可得到问题的离散数值解 $M(i), i = 1, 2, \dots, n$.

6 仿真计算实验

从实际情况考虑, 必须保持船在 y 方向上的合速度 $\dot{y} > 0$, 否则船不可能到达对岸. 根据式(8)可知 M 存在如下约束: $\sin M > -(\omega/v_0)$. 根据问题的实际意义, 将 M 的搜寻区间限制在 $[0, \pi]$, 则有

$$\pi - \arcsin(-\omega/v_0) > M > \arcsin(-\omega/v_0). \quad (20)$$

此外, 还必须满足

$$-1 < \omega/v_0 < 0 \quad (21)$$

如果所有 $M(i)$ 满足式(20)和式(21), 而且满足 $\dot{x} = 0$, 则船所走的航线必然是一条直线, 即为最短距离的航线. 与其相对应的情况是: 船在垂直于河岸方向的合速度始终大于零, 而与河岸平行的方向合速度始终等于零. 下面的仿真模拟仅考虑除此以外的其他情况, 实验参数设置和初始值满足式(21), 搜寻范围 $M(i) (i = 1, 2, \dots, n)$ 满足式(20), 具体如表1所示

表1 参数设置

a_1	a_2	a_3	a_4	a_5	a_6	n	h	v_0	x_0	y_0
2	200	1.2	1	2000	1	1000	100	1.5	200	0

因为 hPSO 本质上是一种随机方法, 所以分别用基本 PSO 和 hPSO 进行 100 次实验 $k (k = 1 \sim 100)$ 为实验次数; $s (s = 1 \sim 1000)$ 为每次实验迭代步数; z, e, j 分别表示基本 PSO 和 hPSO 仿真计算 100 次得到的平均全局最小值、方差、平均 CPU 时间, 结果如图 1 和图 2 及表 2 所示. 其中

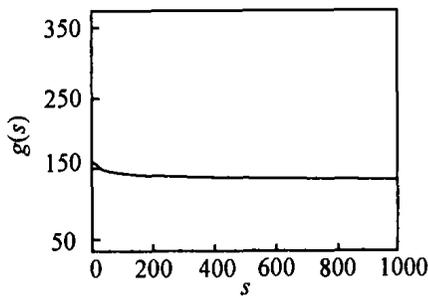
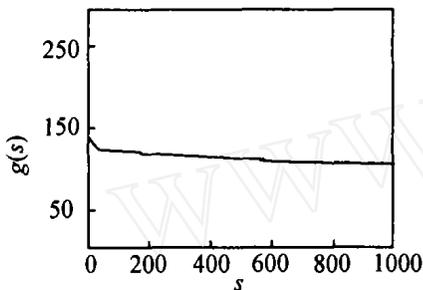
$$g(s) = \frac{1}{100} \sum_{k=1}^{100} J(M(s))$$

表示 100 次实验平均的 J 对 s 的函数

表2 仿真计算结果

基本 PSO			hPSO		
z_1	e_1	j_1/s	z_2	e_2	j_2/s
132.4	2.56	14	105.3	2.33	16

7 结 论

图 1 采用 PSO 优化的 $g(s)$ 曲线图 2 采用 hPSO 优化的 $g(s)$ 曲线

从图 1、图 2 和表 1、表 2 的结果可以看出,在方差和平均计算时间基本相同的前提下,hPSO 方法

比 PSO 方法得到的平均全局值小,在优化 $J(M(s))$ 时比 PSO 下降快,PSO 迭代 750 次后基本保持稳定 由此可见,通过 h 变换在很大程度上克服了陷入局部极小点的问题,提高了 PSO 方法对于多极值点函数的全局优化效率 该方法可推广到其他非线性系统最优控制问题的求解,为解决非线性系统最优控制问题提供了一种有效途径

参考文献(References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization [A]. *Proc of IEEE Int Conf on Neural Networks*[C]. Piscataway, 1995: 1942-1948
- [2] Kennedy J. The particle swarm: Social adaptation of knowledge[A]. *Proc of IEEE Int Conf on Evolutionary Computation*[C]. Indianapolis, 1997: 303-308
- [3] Clerc M, Kennedy J. The particle swarm-explosion, stability and convergence in a multidimensional complex space[J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(1): 58-73
- [4] Rubio J E. *Control and optimization*[M]. Manchester: Manchester University Press, 1985
- [5] 胡跃明. *非线性控制系统理论与应用*[M]. 北京: 国防工业出版社, 2002

(上接第 473 页)

- [3] Shanma J S, Athans M. Analysis of nonlinear gain scheduled control system [J]. *IEEE Trans on Automatic Control*, 1990, 35(8): 898-907.
- [4] Shanma J S, Athans M. Gain scheduling: Potential hazards and possible remedies[J]. *IEEE Control System Magazine*, 1992, 12(3): 101-107.
- [5] Jiang J. Optimal gain scheduling controller for a diesel engine [J]. *IEEE Control System Magazine*, 1994, 14(4): 42-48
- [6] Apkarian P, Gahinet P. A convex characterization of gain scheduled H controller [J]. *IEEE Trans on Automatic Control*, 1995, 40(5): 853-863
- [7] Kajiwara H, Apkarian P, Gahinet P. LPV techniques for control of an inverted pendulum [J]. *IEEE Control System Magazine*, 1999, 19(1): 44-54
- [8] 李士勇. *模糊控制 神经控制和智能控制*[M]. 哈尔滨: 哈尔滨工业大学出版社, 1998: 365-368
- [9] Mandani E H. A misconception of theory and application[J]. *IEEE Expert*, 1994, 9(4): 27-28