文章编号: 1001-0920(2001)01-0123-04

基于线性搜索的混沌优化及其在非线性约束优化问题中的应用

张春慨, 李霄峰, 邵惠鹤
(上海交通大学 自动化研究所, 上海 200030)

摘 要: 提出基于线性搜索的混沌优化方法, 利用混沌变量的特定内在随机性和遍历性来跳出局部最优点, 而线性搜索可以提高局部空间的搜索速度和精度。结合精确不可微罚函数求解非线性约束优化问题。仿真结果表明, 该算法简单易行, 求解精度、收敛速度和可靠性较高, 是解决优化问题的一种有效方法。

关键词: 混沌优化; 精确罚函数; 线性搜索; 非线性约束
中图分类号: TP 18 **文献标识码**: A

Chaos Optimization Algorithm Based on Linear Search and Its Application to Nonlinear Constraint Optimization Problems

ZHANG Chun-kai, LI Xiao-feng, SHAO Hui-he
(Research Institute of Automation, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: The chaos optimization algorithm (COA) using linear search is proposed. It takes advantage of the intrinsic stochastic property and ergodicity of chaos movement to escape from the local minima. The local search ability of linear search can speed up the rate of convergence. To solve nonlinear constraint optimization problems, COA is combined with exact non-differentiable penalty function. The simulation results illustrate that this method is simple and easy to implement.

Key words: chaos optimization; exact penalty function; linear search; nonlinear constraint

1 引 言

对于大规模非线性约束优化问题, 通常描述为多模、非凸的。与传统的确定性优化算法相比, 高效率的直接搜索方法是一种很有潜力的方法, 例如模拟退火算法(SA)、进化算法(EA)、随机搜索方法等。它们的共同特点是具有通用性, 对计算资源的要求不高, 在解的允许空间内直接利用优化问题的一些点值, 而不管其解析性质如何, 且求出的极值点往往是全局最优点。

近期出现一种新型的直接搜索优化算法——混沌优化。它直接采用混沌变量在允许解空间进行搜索, 搜索过程按混沌运动自身规律进行。与按某种概率接受"劣解"跳出局部最优解的优化算法(如SA)相比, 混沌优化更易于跳出局部最优解, 且搜索效率

收稿日期: 1999-08-20; **修回日期**: 1999-10-22
基金项目: 国家973重点基础研究发展项目(G1998030415)
作者简介: 张春慨(1973—), 男, 山东烟台人, 博士生, 从事优化方法及控制、人工智能等的研究; 邵惠鹤(1936—), 男, 浙江宁波人, 教授, 博士生导师, 从事工业生产过程计算机控制与优化等方面的研究。

高,适合于完成全局优化搜索。文献[1]利用分叉理论通过神经网络进行寻优;文献[2]用混沌载波搜索的优化方法求解无约束的优化问题;文献[3]采用逐次减小混沌搜索空间以加快解的收敛速度的优化方法求解无约束的优化问题。上述几种方法仅仅利用混沌变量的特定内在随机性和遍历性来搜索最优解,并没有利用已找到的局部最优解,且当最优解位于小区间时,解的收敛速度仍然缓慢。

混沌优化作为一种直接搜索算法,其缺点是局部优化效果不理想。自然的想法是把混沌优化的全局搜索能力与一般的局部搜索算法相结合。鉴于混沌优化方法的一个重要特点是不要求优化问题具有连续性和可微性,若采用一般的局部优化算法,如最速下降法等梯度寻优算法,则会降低混沌优化的特点。为此,本文提出一种改进的混沌优化方法,利用在混沌搜索中得到的先验知识,通过线性搜索在局部搜索空间加速最优解的收敛速度,提高解的精度,并与精确不可微罚函数相结合,用于求解实际生产中普遍存在的非线性约束优化问题。

2　基于线性搜索的混沌优化算法

2.1　问题描述

非线性约束优化问题可描述为

$$\min \; f(X) \tag{1}$$

$$\text{s.t.}\quad g_i(X) = 0,\quad i = 1,2,\cdots,m_e \tag{2}$$

$$g_j(X) \ge 0,\quad j = 1,2,\cdots,m_b \tag{3}$$

其中,$X \in R^n$ 和 $f(X)$ 是目标函数,m_e 为等式约束个数,m_b 为不等式约束个数,优化问题(1)的可行域是有界的。对于约束优化问题,需要先对变量取值范围有一个合理估计 $[a_j, b_j]$, $j = 1,2,\cdots,n$。对于实际问题而言,通常是可以做到的。

$$a_i \le x_i \le b_i,\quad i = 1,2,\cdots,n \tag{4}$$

$[a_j, b_j]$ 的确定原则是使

$$\{x_i \mid a_i \le x_i \le b_i\} \supseteq \{x_i \mid g_j(x_i) \ge 0,\ j = 1,2,\cdots,m_b\}\quad i = 1,2,\cdots,n \tag{5}$$

对于约束处理,采用 l_1 不可微精确罚函数法[4],把原问题转化为以罚函数为目标函数的新的无约束优化问题。精确罚函数方法避免了计算上的序贯性,直接使约束优化的解与罚函数的某个极小点"精确"地一致。不可微精确罚函数是不可微函数,但构造相对较为简单,对于不需要梯度信息的优化方法很有吸引力。混沌优化不要求优化问题可微、连续,所以采用不可微精确罚函数是有效的。

$$P(\alpha, X) = f(X) + \alpha\Big[\sum_{i=1}^{m_e} |g_i(X)| + \sum_{i=1}^{m_b} (-\min(0, g_i(X)))\Big] \tag{6}$$

式中,$f(X)$ 是原问题目标函数,α 为罚因子,$P(\alpha, X)$ 是罚函数。

2.2　基于线性搜索的混沌优化算法

混沌变量的产生,常用如下的Logisic模型[5]:

$$x^{k+1} = \eta x^k (1 - x^k),\quad x^k \in [0,1] \tag{7}$$

其中 η 是控制参数,当 $\eta = 4.0$ 时,上式进入混沌状态,具有混沌的一般特性。对于 n 维优化问题,任意设定(0, 1)区间的 n 个相异的初值,但不能为0.25, 0.5和0.75。

在基于线性搜索的混沌优化算法中,利用前 N_1 次搜索找到的局部最优点 X^*_{old},与再做 N_1 次搜索找到的局部最优点 X^* 进行比较,若 $\|X^*_{\text{old}} - X^*\| > \epsilon$,则沿方向 $P = X^* - X^*_{\text{old}}$,从 X^* 出发做线性搜索求出 X^0,用 X^0 代替 X^*。由于利用了下降法,所以可加快算法的收敛速度,并可提高解的精度。具体算法的实现方案如下:

1) 初始化:由式(8)生成 n 个(0, 1)分布的混沌变量 λ_j, $j = 1,2,\cdots,n$。对于约束优化问题,需对变量取值范围先做一个合理估计 $[a_j, b_j]$ $(j = 1,2,\cdots,n)$,并取 N_1 和 N_2 为较大的整数。

2) 用混沌变量进行迭代搜索:

① 按式(8)将混沌变量变换到优化问题的允许解空间

$$x_j^k = a_j + \lambda_j (b_j - a_j),\quad j = 1,2,\cdots,n \tag{8}$$

② 如果当前点的函数值小于已有的最优值,即 $f(X^k) < f^*$,则保留已有最优点、当前点及其相应的最优函数值 $X^*_{\text{old}} = X^*$, $X^* = X^k$, $f^* = f(X^k)$;

③ 如果 f^* 经 N_1 步搜索后保持不变,且 $\|X^*_{\text{old}} - X^*\| > \epsilon$,则转步骤3);否则继续迭代搜索。

3) 求 $P = X^* - X^*_{\text{old}}$,记容许集合为 R,做线性搜索

$$f(X') = \min_a \{f(X^* + aP) \mid X^* + aP \in R\}$$

$$X^*_{\text{old}} = X',\quad X^* = X',\quad f^* = f(X')$$

4) $m = m + 1$,如果 $m > N_2$,则终止搜索,此时 X^* 即为全局最优点,f^* 为得到的最优解;否则转步骤2)。

表 1　问题 1 的求解结果

Trial	$f(x)$	x_1	x_2	x_3	x_4	x_5	$c_1(x)$	$c_2(x)$	$c_3(x)$
线性搜索混沌优化	-31 016.9	78.02	33.00	27.11	44.90	44.89	91.98	100.39	20.00
变尺度混沌优化	-30 901.8	79.22	33.29	27.49	44.50	43.29	91.74	100.18	20.04
Homaifar	-30 175.8	80.61	34.21	31.34	42.05	34.85	90.58	99.41	20.12
Fogel	-31 006.2	78.00	33.00	27.14	44.91	44.90	91.98	100.40	20.01

3　优化实例

以文献[5, 6]中给出的 2 个算例作为实例, 来验证新的混沌优化方法的有效性。

问题 1

$$\min f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$
$$\text{s.t.}\quad g_1(X) = x_1 - 2x_2 + 1 = 0$$
$$g_2(X) = x_1^2/4 - x_2^2 + 1 \geq 0$$

问题 2

$$\min f(x) = 5.357\,854\,7x_3^2 + 0.835\,689\,1x_1x_5 + 37.293\,239x_1 - 40\,792.141$$
$$\text{s.t.}\quad 0 \leq c_1(x) \leq 92$$
$$90 \leq c_2(x) \leq 110$$
$$20 \leq c_3(x) \leq 25$$

其中

$$c_1(x) = 85.334\,407 + 0.005\,686x_2x_5 + 0.000\,26x_1x_4 - 0.002\,205x_3x_5$$
$$c_2(x) = 80.512\,49 + 0.007\,132x_2x_5 + 0.002\,996x_1x_2 + 0.002\,181x_3^2$$
$$c_3(x) = 9.300\,961 + 0.004\,703x_3x_5 + 0.001\,255x_1x_3 + 0.001\,909x_3x_4$$
$$x_1 \in [78, 102],\quad x_2 \in [33, 45],\quad x_3 \in [27, 45]$$
$$x_4 \in [27, 45],\quad x_5 \in [27, 45]$$

采用上述混沌优化方法结合 l_1 不可微精确罚函数方法, 利用混沌的内在随机性和遍历性进行求解。在问题 1 中, 罚因子为 5.0×10^4, $N_1 = 200$, $N_2 = 50$, GA 的种群大小为 400, 染色体长度为 19, 交叉概率为 0.85, 突变概率为 0.02, 进化规划采用(0, 0.1)分布的高斯变异算子。在问题 2 中, 除了 GA 中的交叉概率为 0.8, 突变概率为 0.088 外, 其余参数均相同。对每种方法都独立运行 10 次, 平均最好结果分别见表 1 和表 2, 其中 Homaifar 和 Fogel 为文献[6](遗传算法) 和文献[7](进化规划) 的结果。图 1 和图 2 比较了基于线性搜索混沌算法和变尺度混沌算法的迭代过程。图中实线为变尺度混沌算法曲线, 虚线为新的混沌算法曲线。

表 2　问题 2 的求解结果

Trial	$f(x)$	x_1	x_2	$g_1(X)$	$g_2(X)$
线性搜索混沌优化	1.374 6	0.833 8	0.914 1	0.001	0.002
变尺度混沌优化	1.393 4	0.822 9	0.911 4	0.000	0.000
Homaifar	1.433 9	0.808 0	0.885 4	0.037	0.052
Fogel	1.378 8	0.833 3	0.912 4	0.008	-0.006 1

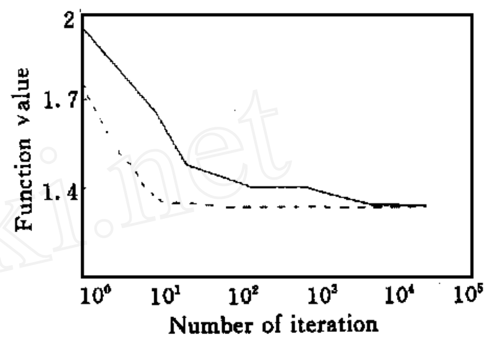

图 1　问题 1 的两种混沌算法迭代过程比较

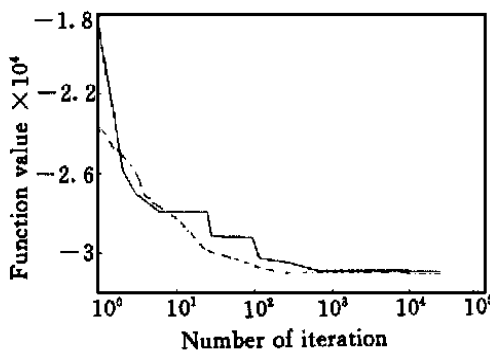

图 2　问题 2 的两种混沌算法迭代过程比较

从求解精度、收敛速度、可靠性及满足约束几方面看, 基于线性搜索的混沌算法的效果最好, Fogel 的结果次之, 而所有的混沌优化算法匀好于随机搜索和 Homaifar 给出的解。新的混沌算法与变尺度混沌算法相比, 收敛速度快, 解的精度高。混沌优化算法实现容易, 不需要特殊的编码、解码、选择、交叉等操作, 所以结果是令人满意的。

(下转第 128 页)

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{19}$$

将 $a = \lambda_{max}(H)$, $b = 2\lambda_M(A)\lambda_{max}(H)$, $c = -\lambda_{min}(Q)$ 代入式(19), 整理得

$$\|\Delta\bar{A}\| < -\lambda_M(\bar{A}) + \sqrt{\lambda_M^2(\bar{A}) + \frac{\lambda_{min}(Q)}{\lambda_{max}(H)}} \tag{20}$$

因为式(20) 成立, 保证式(18) 小于零成立, 所以预测控制系统(13) 具有鲁棒稳定性。(证毕)

如果选取 $Q = I$ 为单位矩阵, 则有

$$\|\Delta\bar{A}\| < -\lambda_M(\bar{A}) + \sqrt{\lambda_M^2(\bar{A}) + \frac{1}{\lambda_{max}(H)}} \tag{21}$$

状态反馈预测控制系统鲁棒稳定性设计的步骤如下:

1) 适当选取预测时域 P, 按标称系统 (A, B, C) 设计预测控制系统, 按式(6), $\bar{A} = A - BS^{-1}(P)K$ 使 $\bar{A}$ 稳定;

2) 检验是否满足定理 1 条件, 即当 $\Delta B = 0$ 时, $\Delta\bar{A} = \Delta A$; 当 $\Delta B \neq 0$ 时, $\Delta\bar{A} = \Delta A - \Delta BS^{-1}(P)K$;

3) 若满足定理条件, 则闭环预测控制系统具有鲁棒稳定性; 否则, 调整预测时域 P, 重复上述设计过程。

如果 $\Delta B = 0$, 则 $\Delta\bar{A} = \Delta A$, 此时定理 1 给出的结果是准确的; 如果 $\Delta B \neq 0$, 可用 $\|\Delta A\| + \beta\|\Delta B\|$ 代替 $\|\Delta\bar{A}\|$ 项, 其中 $\beta > 0$ 可由 $S^{-1}(P)K$ 确定。这时的判断结果是比较保守的。

3 结 论

本文分析了状态反馈预测控制系统的鲁棒稳定性, 给出了鲁棒稳定性的充分条件。基于状态空间模型, 使用实测状态变量反馈, 将参数不确定项引入控制作用中, 增加了状态变量的动态反馈。与 DMC, GPC 和 MC 算法相比, 改善了控制系统的鲁棒性, 为状态反馈预测控制系统的分析、设计及实际应用提供了理论依据。

参考文献:

[1] Culter C R, Ramaker B L. Dynamic matrix control——A computer control algorithm [M]. San Francisco: JACC, 1980.

[2] Clarke D W, Mohtadi C, Tuffs P S. Generalized predictive control[J]. Automatica, 1987, 23(2): 137-148.

[3] Clarke D W, Mohtadi C. Properties of generalized predictive control[J]. Automatica, 1989, 25(6): 859-875.

[4] 袁璞, 左信, 郑海涛. 状态反馈预估控制[J]. 自动化学报, 1993, 19(5): 569-577.

[5] 袁璞. 单值预估控制[J]. 石油大学学报(自然科学版), 1992, 16(5): 100-109.

[6] 胡品慧. 多变量状态反馈预测控制及应用[D]. 北京: 石油大学, 1999.

[7] 胡品慧, 袁璞. 状态反馈预测控制系统研究[J]. 石油大学学报(自然科学版), 2000, 24(2): 98-100.

[8] Garcia, C E, Prettt D M, Morari M. Model predictive control: Theory and practice——A survey[J]. Automatica, 1989, 25(3): 335-348.

[9] 袁璞. 生产过程动态数学模型及其在线应用[M]. 北京: 中国石化出版社, 1998.

[10] 舒迪前. 预测控制系统及其应用[M]. 北京: 机械工业出版社, 1998.

(上接第 125 页)

4 结 论

本文在混沌优化方法的基础上, 提出利用线性搜索来加快局部解空间的搜索速度, 并结合精确不可微罚函数求解非线性约束优化问题。算法简单实用, 对优化问题无需连续性、可微性的要求, 搜索效率较高, 是解决约束优化问题的一种有效方法。仿真实例表明, 在求解精度、收敛速度、满足约束及可靠性等方面, 该算法均能取得令人满意的结果。

参考文献:

[1] 张春慨, 邵惠鹤. 采用退化混沌突变的实数编码遗传算法及其应用[A]. WCICA'2000[C]. 合肥, 2000. 634-637.

[2] 李兵, 蒋慰孙. 混沌优化方法及其应用[J]. 控制理论与应用, 1997, 14(4): 613-615.

[3] 张彤, 王宏伟, 王子才. 变尺度混沌优化方法及应用[J]. 控制与决策, 1999, 14(3): 285-287.

[4] 袁亚湘. 非线性规划数值方法[M]. 上海: 上海科学技术出版社, 1993.

[5] Arun V Holden. Chaos[M]. Manchester University Press, 1986.

[6] Homaifar A, Qi C X, Lai S H. Constrained optimization via genetic algorithms[J]. Simulation, 1994, 62(4): 242-254.

[7] Fogel D B. An introduction to simulated evolutionary optimization[J]. IEEE Trans on Neural Network, 1994, 5(1): 3-14.