文章编号: 1001-0920(2007)09-1027-05

基于竞争-协作式信息交互的并行混沌优化算法研究

袁小芳, 王耀南, 吴亮红
(湖南大学 电气与信息工程学院, 长沙 410082)

摘 要: 针对混沌优化对初始值敏感、搜索精确解效率低等不足,研究一种基于竞争-协作式信息交互的并行混沌优化(ICPCO)算法. ICPCO 算法采取并行混沌迭代机制,在每一次迭代搜索之后,根据并行优化解分布状况不同,分别采取竞争或协作式信息交互再次寻优. 描述 ICPCO 算法思想和实现步骤,分析其收敛性和优化性能. 仿真实验表明, ICPCO 算法不仅具有全局搜索能力,而且以信息交互方式提高了优化效率和搜索精度,算法收敛,稳定性增强.

关键词: 计算智能; 混沌; 并行混沌优化; 信息交互; 全局搜索
中图分类号: TP18; TP301 **文献标识码**: A

Parallel chaotic optimization algorithm based on competitive-cooperative inter-communication

YUAN Xiao-fang, WANG Yao-nan, WU Liang-hong
(College of Electrical and Information Engineering, Hu'nan University, Changsha 410082, China. Correspondent: YUAN Xiao-fang, E-mail: yuanxiaof@21cn.com)

Abstract: For the problem that chaotic optimization is sensitive to initial values and suffers from slow convergence velocity around optimum and poor efficiency for accurate optimum, a parallel chaotic optimization algorithm based on competitive/cooperative inter-communication (ICPCO) is proposed. In the proposed algorithm, each variable is mapped to several chaotic variables called parallel chaotic system, and competitive/cooperative inter-communication between parallel variables is employed for research in accordnce with distributing-status of parallel variables. Both framework and algorithmic implementation of the proposed algorithm are introduced. Optimization performance and convergence of ICPCO are analyzed. Simulations demonstrate that this algorithm has better performance over other parallel chaotic optimization.

Key words: Computation intelligence; Chaos; Parallel chaotic optimization; Inter-communication; Global search

1 引 言

混沌现象存在于绝大多数非线性系统中,表现出介于规则与随机之间的一种随机行为. 根据混沌运动的独特性能,人们研究了以混沌运动为基础的多种类型混沌优化算法[1-8]. 目前,混沌优化算法可分为以下 4 类: 1) 直接利用混沌变量搜索[1,2]; 2) 与其他算法组合成混合优化算法,如混沌结合禁忌搜索[3]、并行混沌结合单纯形法[4]等; 3) 将混沌机制引入其他优化算法,如混沌进化算法[5]、混沌模拟退火[6]等; 4) 由外部机制产生的混沌噪声构成混沌神经网络优化[7,8].

混沌优化算法具有较强的全局搜索能力,然而在最优解邻域收敛慢,搜索精确解的效率不高. 由于混沌随机性强,初始值不同时,算法优化效率和搜索精度差别很大,收敛稳定性不强. 考虑到混沌优化的这些缺点与其串行机制相关,学者们提出了并行混沌优化算法(PCO)[2,4]. 文献[2]研究了同时采用 2 种不同混沌机制的混沌优化算法,文献[4]研究了在并行混沌优化粗搜索之后由单纯形法搜索精确解. PCO 算法不仅全局搜索能力强,而且以并行混沌机制提高了搜索速度和精度. 然而,这些 PCO 算法没有考虑并行变量之间的信息交互,因而整体优化效率还不够高.

本文依据"独立迭代、信息交互"思想,充分考虑并行变量之间的信息交互,提出了一种基于竞争-协作式信息交互的并行混沌优化算法(ICPCO). 与其

收稿日期: 2006-03-01; **修回日期**: 2006-04-28.
基金项目: 国家自然科学基金项目(60375001); 高校博士点基金项目(20030532004).
作者简介: 袁小芳(1979—),男,湖南安仁人,博士生,从事智能控制、机器学习等研究; 王耀南(1957—),男,昆明人,教授,博士生导师,从事智能控制理论与应用、智能信息处理等研究.

他 PCO 算法不同,ICPCO 算法着重研究了并行变量之间的信息交互,以信息交互来搜索适应度更优的解. 根据并行优化解分布状况不同,ICPCO 算法分别采取竞争或协作式信息交互:并行解分布分散时,采取竞争式信息交互,由整体最优解引导并行变量在其较大的邻域寻优;并行解分布集中时,采取协作式信息交互,由单纯形法进行局部寻优. 研究表明,ICPCO 算法全局搜索能力强、收敛稳定性强,信息交互方式提高了整体优化效率和搜索精度.

2 并行混沌优化算法

文献[9]提出了一种折叠次数无限混沌映射(ICMIC),即

$$x_{n+1} = f(a, x_n) = \sin(a/x_n),$$
$$-1 \leq x_n \leq +1,\ x_0 \neq 0,$$
$$a \in (0, +\infty),\ n = 0, 1, \cdots. \quad (1)$$

计算 Lyapunov 指数也表明该映射模型比常用的 Logistic 映射、Chebyshev 映射具有更优的混沌特征. 本文采用 ICMIC 映射作为并行混沌优化(PCO)算法的混沌载波. 这里取 $a = 2$. 在 PCO 算法中,对于 n 个优化变量,并行地给定 $P \times n$ 个混沌变量,即每一个优化变量由 P 个混沌变量来独立并行映射,优化结果取 P 个并行混沌变量的映射最优值.

考虑如下一类优化问题:

$$\min f(x),\ x = \{x_1, x_2, \cdots, x_n\},\ a_i \leq x_i \leq b_i. \quad (2)$$

这里:x 为 n 维优化变量,$[a_i, b_i]$ 为优化变量的定义域. 定义以下变量:k 为迭代次数;P 为并行数;j 表示每一个并行变量,$j = 1, 2, \cdots, P$;x_{ij}^k 表示第 k 次迭代时,第 i 个变量对应的并行第 j 个变量值;mx 表示混沌变量值;mx_{ij}^k 表示第 k 次迭代时,第 i 个变量对应的并行第 j 个混沌变量值;fp_j^* 为并行最优值,即每一组并行变量的当前最优值;$x_{ij}^\wedge$ 为对应的并行最优解;f^* 为整体最优值,即 $f^* = \min(fp_j^*)$;x_i^* 为对应的整体最优解.

为提高搜索效率,PCO 算法自适应收缩搜索范围,算法描述如下(其中 $i = 1, 2, \cdots, n, j = 1, 2, \cdots, P$):

Step1: 初始化. $k = 1$,随机产生 $P \times n$ 个混沌轨迹,得到 mx_{ij}^k,对于 $x_{ij}^k, x_{ij}^\wedge, x_i^*, fp_j^*, f^*$ 取初始值,取搜索范围为定义域 $a_{ij}^c = a_i, b_{ij}^c = b_i$.

Step2: 并行迭代混沌变量. 采用混沌映射(1)计算混沌变量

$$mx_{ij}^{k+1} = \sin(2/mx_{ij}^k), \quad (3)$$

将混沌变量线性映射到优化变量的搜索区间,即

$$x_{ij}^k = a_{ij}^c + |mx_{ij}^k| (b_{ij}^c - a_{ij}^c). \quad (4)$$

Step3: 用并行优化变量独立迭代搜索.

If $f(x_{ij}^k) < fp_j^*$, $fp_j^* = f(x_{ij}^k)$, $x_{ij}^\wedge = x_{ij}^k$,

Else fp_j^*, $x_{ij}^\wedge$ 不变;

If $fp_j^* < f^*$, $f^* = fp_j^*$, $x_i^* = x_{ij}^\wedge$,

Else f^*, x_i^* 不变.

Step4: 自适应收缩搜索范围,即

$$a_{ij}^c = x_{ij}^\wedge - q \cdot (b_{ij}^c - a_{ij}^c),$$
$$b_{ij}^c = x_{ij}^\wedge + q \cdot (b_{ij}^c - a_{ij}^c). \quad (5)$$

其中 q 为自适应收缩系数

$$q = 1 - \left(\frac{k-1}{k}\right)^d, \quad (6)$$

d 为设定的参数. 为确保变化后的搜索区间不超过定义域,作以下处理:

$$\text{If } a_{ij}^c < a_i,\ a_{ij}^c = a_i;$$
$$\text{If } b_{ij}^c > b_i,\ b_{ij}^c = b_i. \quad (7)$$

Step5: 判断是否满足终止条件,若满足,则结束;否则 $k = k + 1$,转 Step2 继续迭代.

3 ICPCO 算法思想与设计

3.1 算法思想与流程

PCO 算法采取独立迭代搜索方式,整体搜索效率不高,在复杂问题优化中,迭代次数往往随着变量维数的增大而急剧增加. 分析 PCO 算法迭代过程发现,在一定迭代时期内,某一组并行优化解 $x_{ij}^\wedge$ 的适应度没有改善,而且这种情况比较普遍,从而影响到整体优化效率. 考虑到当前最优解 x_i^* 接近全局最优解的概率显然更大,这时采用 x_i^* 去引导 $x_{ij}^\wedge$ 寻优. 依据"独立迭代、信息交互"思想,ICPCO 算法根据并行优化解分布状况采取相应的信息交互方式:$x_{ij}^\wedge$ 分布比较分散时,并行变量采取竞争式信息交互,由 x_i^* 引导 $x_{ij}^\wedge$ 在其邻域寻优;$x_{ij}^\wedge$ 分布集中时,采取协作式信息交互,由局部搜索能力强的单纯形法对 $x_{ij}^\wedge$ 进

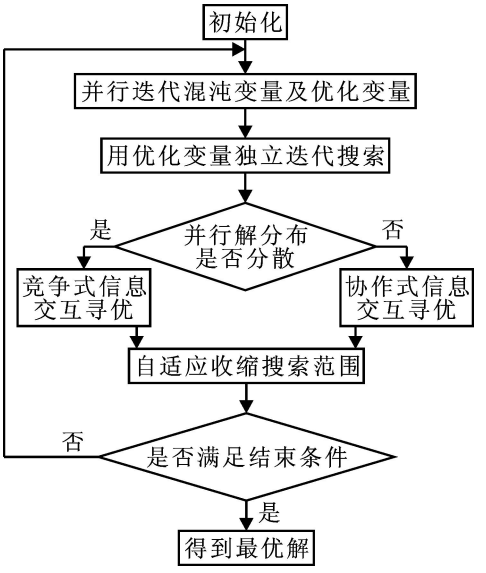


图 1 ICPCO 算法流程框图

行精确搜索. ICPCO 算法利用整体优化解 x_i^* 信息去引导 x_{ij} 寻找适应度更优的解,完善了寻优途径,提高了优化性能.

ICPCO 算法流程如图 1 所示.

3.2　竞争式信息交互

并行优化解 x_{ij} 分布分散时,采取竞争式信息交互:整体优化解 x_i^* 在竞争中取得胜利, x_{ij} 以邻域迁移的方式在 x_i^* 较大的邻域进行寻优,以产生适应度更优的解. 竞争式信息交互描述如下:

Step1: 确定竞争中的胜者. 每一次混沌优化迭代之后,依据各个并行解的适应度,确定整体最优值 f^* 所对应的整体最优解 x_i^* 在竞争中取得胜利.

Step2: 确定邻域. 邻域是以 x_i^* 为中心的一个 n 维球面,即

$$D = \{ D \mid D \in R^n, D = x_i^* + q_d \mid b_i - a_i \mid \}. \tag{8}$$

其中: q_d 为自适应收缩系数, $q_s = 1 - ((k-1)/k)^m$, m 为设定的一个参数. 搜索初期 q_d 接近于 1,有利于全局搜索. 随着迭代次数增加, q_d 逐渐减小,有利于局部精确搜索.

Step3: 邻域迁移. 邻域迁移就是将 x_{ij} 引导到以 x_i^* 为中心的 D 域,且迁移后的 x_{ij}^d 在 D 域内以一种遍历性的方式分布,即

$$x_{ij}^d = x_i^* + D_i \sin(2/m x_{ij}^k). \tag{9}$$

Step4: 邻域寻优. If $f(x_{ij}^d) < f(x_{ij})$, $fp_j^* = f(x_{ij}^d)$, $x_{ij} = x_{ij}^d$, Else, fp_j^* 和 x_{ij} 保持原值; If $fp_j^* < f^*$, $f^* = fp_j^*$, $x_i^* = x_{ij}$, Else, f^* 和 x_i^* 保持原值.

算法中若邻域迁移可以提高适应度,则进行迁移;否则, x_{ij} 保持原值. 寻优后的 x_{ij}, x_j^* 将作为下一次并行混沌迭代的起点. 这里邻域范围不宜过小,否则并行优化解分布过早地集中在一个较小的邻域,算法将过早地进入协作式信息交互,不利于全局搜索.

3.3　协作式信息交互

并行优化解分布集中时,算法将考虑寻找精确最优解. 此时采取协作式信息交互,以 Nelder- Mead 单纯形法[10] 增强 ICPCO 算法的局部搜索能力. 协作式信息交互是对每一个优化变量所对应的 P 个并行优化解 x_{ij} 分别进行一次单纯形处理. 协作式信息交互描述如下(其中 $i = 1, 2, \cdots, n$):

首先形成初始单纯形. 对于 n 维优化问题,采用 n 个独立的单纯形,每一个优化变量 x_i 的 P 个并行优化解 x_{ij} 就是每一个单纯形的 P 个初始顶点. 用 xw_i, xb_i 分别表示优化变量 x_i 的最差解和最优解,且有

$$xm_i = (x_{i1}^* + x_{i2}^* + \cdots + x_{ip}^* - xw_i)/(P-1). \tag{10}$$

单纯形法操作的主要步骤简述如下:

Step1: 反射

$$xr_i = xm_i + (xm_i - xw_i). \tag{11}$$

Step2: 延伸. 当反射成功时,延伸

$$xe_i = xm_i + \alpha(xm_i - xw_i), \tag{12}$$

这里 α 为延伸系数, $\alpha > 1$.

Step3: 收缩. 当反射失败时,收缩

$$xc_i = xm_i + \beta(xm_i - xw_i), \tag{13}$$

这里 β 为收缩系数, $0 < \beta < 1$.

Step4: 缩小边长. 当反射、收缩均失败时,缩小边长

$$xw_i = (xw_i + xb_i)/2. \tag{14}$$

由单纯形更新的 x_{ij} 将作为下一次混沌迭代的起点. 单纯形法可以对 x_{ij} 进行局部搜索,淘汰最差解 xw_i,并产生适应度更优的解,算法更快逼近精确最优解.

3.4　ICPCO 的算法实现

与其他 PCO 算法不同,ICPCO 算法以信息交互寻优来提高整体优化效率. ICPCO 算法根据 x_{ij} 分布状况来确定信息交互方式,因而合理评估 x_{ij} 的分布状况是算法的一个重要内容. 这里以 x_{ij} 与 x_i^* 的距离来评估其分布状况,距离定义为

$$d = \sqrt{\sum_j^n \sum^P (x_i^* - x_{ij})^2}. \tag{15}$$

依据 d 值确定信息交互方式:若 $d \geq d_m$,采取竞争式信息交互;否则,采取协作式信息交互. d_m 为根据具体优化问题设定的距离参数.

ICPCO 算法步骤归纳如下:

Step1: 确定算法参数:并行数 P 和距离判据 d_m 等.

Step2: 初始化. 随机产生初始变量值.

Step3: 并行迭代混沌变量和优化变量.

Step4: 用优化变量独立迭代搜索.

Step5: 判断. If $d \geq d_m$, 转 Step6; 否则转 Step7.

Step6: 竞争式信息交互寻优. 进行一次邻域迁移寻优,迭代并行优化解及其适应度,转 Step8.

Step7: 协作式信息交互寻优. 进行一次单纯形操作,迭代并行优化解及其适应度,转 Step8.

Step8: 自适应收缩搜索范围.

Step9: 判断是否满足终止准则. 若满足,则终止搜索;否则,转 Step3 继续搜索.

ICPCO 算法既具有混沌优化算法的全局搜索能力,又以信息交互方式完善了寻优途径,提高了整

体优化效率. 竞争式信息交互在当前最优解较大的邻域寻优;协作式信息交互可以进行局部搜索,快速逼近精确最优解.

4 算法性能分析

对于 ICPCO 算法,并行数 P 取值对于算法优化性能影响较大. 这里以函数 F 为例,研究并行数 P 对于算法性能的影响.

$$F = x_1^1 + x_2^2 + x_3^2 + x_4^2,$$
$$-100 \leq x_i \leq 100. \quad (16)$$

这里终止代数 K_s 是指优化最优值与实际最优值偏差小于 ε 时的迭代次数 k 值;平均终止代数 K_m 是指 m 次独立运行时 K_s 的算术平均值;终止代数方差 S_d 定义为 $\sqrt{\frac{1}{m}\sum_{i}^{m}(K_{s_i} - K_m)^2}$,反映了 m 次独立运行时的收敛稳定性. 表 1 描述了并行数 P 取值不同时的仿真结果.

表 1 并行数 P 不同时的算法性能

P 值	最小终止代数	最大终止代数	平均终止代数	终止代数方差	平均计算时间
3	251	310	289	26.6	0.39
4	245	301	275	24.2	0.42
5	164	301	238	40.6	0.42
6	167	295	210	28.2	0.39
7	175	269	204	23.9	0.38
8	187	262	216	16.7	0.39
10	212	268	235	13.8	0.43
15	239	295	268	13.5	0.48

表 1 中数据为独立运行的 30 组仿真数据,即 $m = 30$,且 $\varepsilon = 10^{-6}$. 表 1 的仿真表明,并行数 P 取值对于算法性能的影响比较明显. 在平均终止代数 K_m 上,开始阶段 K_m 随着 P 值增加而减小,当 P 值超过一定值后,K_m 随着 P 值增加而增大. 在终止代数方差 S_d 上,随着并行数 P 的增加,S_d 呈减小趋势,表明 P 值越大,收敛稳定性越强. 在计算时间上,随着并行数 P 的增加,计算时间呈增大趋势,这是因为 P 值越大,每一次迭代中的计算量增加. 分析表明,P 值对算法性能影响较大,取值不宜过大或过小,针对优化问题 F 可取 $P = 6 \sim 8$ 之间,并可以作为其他优化问题的参考.

5 仿真研究

实验 1 取文献[2]中的 4 个函数进行优化计算,将 ICPCO 算法与文献[2,4-6]中的双混沌机制优化(算法 1),并行混沌结合单纯形法(算法 2),混沌进化算法(算法 3) 和混沌模拟退火(算法 4) 4 种优化算法进行对比. 取并行数 $P = 6$,$\varepsilon = 10^{-6}$,$m = 30$,即各个算法均独立运行 30 次. 优化函数为

$$F_1 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2,$$
$$-2.048 \leq x_1, x_2 \leq 2.048; \quad (17)$$

$$F_2 = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)][30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)],$$
$$-2 \leq x_1, x_2 \leq 2; \quad (18)$$

$$F_3 = \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} - 0.5,$$
$$-100 < x_i < 100; \quad (19)$$

$$F_4 = (x_1^2 + x_2^2)^{0.25}[\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1], \quad -100 < x_i < 100. \quad (20)$$

ICPCO 算法与其他 4 种算法的对比仿真结果如表 2 和图 2 所示,数据为 30 次独立运行的平均结果. 图 2 还描述了 ICPCO 算法与其他 4 种算法在求解 $F_1 \sim F_4$ 时的收敛速度曲线. 表 2 和图 2 的仿真数据表明,ICPCO 算法比其他 4 种并行混沌优化算法的性能更优:用最少的迭代次数便搜索到全局优化解,表明 ICPCO 算法的搜索效率高;终止代数方差 S_d 也最小,表明 ICPCO 算法收敛稳定性强,受初始值的影响极小;在计算时间上,ICPCO 算法也比其他方法少.

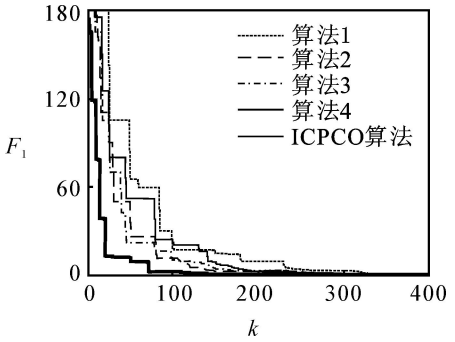
实验 2 优化算法辨识 ARMAX 模型参数. 设描述系统的数学模型为

$$y(k) - 1.5y(k-1) + 0.7y(k-2) = u(k-1) + 0.5u(k-2) + g(k). \quad (21)$$

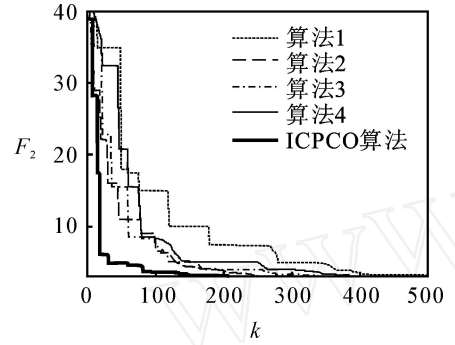
式中:$g(k)$ 为随机白噪声序列,其均值为 0,方差 $\sigma^2 = 0.02$. 输入实验信号分别用长度为 127 的伪随机二进制序列(PRBS) 和正弦序列. 设参数搜索区间为[-5,5],仍将 ICPCO 算法与双混沌机制优化(算

表 2 ICPCO 算法与其他算法的对比仿真

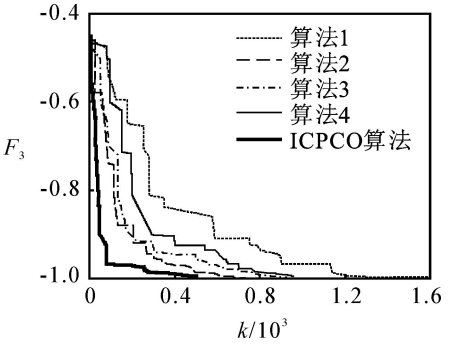
函数	算法 1			算法 2			算法 3			算法 4			ICPCO 算法		
	K_m	S_d	t	K_m	S_d	t	K_m	S_d	t	K_m	S_d	t	K_m	S_d	t
F_1	363	39.8	0.35	229	14.3	0.35	268	38.7	0.58	313	41.0	0.31	152	13.2	0.28
F_2	429	49.2	0.62	246	18.9	0.48	367	41.2	0.79	392	43.5	0.39	167	17.1	0.36
F_3	1 644	158.0	0.95	758	35.1	0.87	974	52.9	1.10	951	61.4	0.73	538	29.6	0.71
F_4	1 712	185.1	1.46	789	34.4	1.04	926	70.1	1.61	980	62.3	0.84	548	33.5	0.80



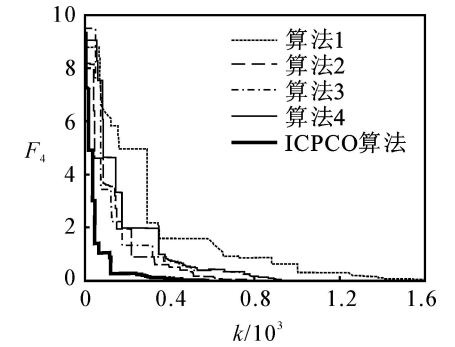
(a)　优化函数 F_1



(b)　优化函数 F_2



(c)　优化函数 F_3



(d)　优化函数 F_4

图 2　求解函数 F_1 ～ F_4 时的收敛速度比较

法 1) ,并行混沌结合单纯形法(算法 2) ,混沌进化算法(算法 3) 和混沌模拟退火(算法 4) 4 种算法对比. 取并行数 $P = 6$,终止代数为 1 500,辨识结果如表 3 所示.

仿真数据表明,ICPCO 算法辨识的精度最高,比其他 4 种优化算法更接近实际精确值,而且是无偏估计,验证了算法的优化性能.

表 3　优化算法辨识系统参数

输入	算法	a_1	a_2	b_1	b_2
PRBS	算法 1	1.485 0	- 0.662 1	0.938 2	0.516 2
	算法 2	1.490 7	- 0.719 7	0.962 4	0.490 6
	算法 3	1.510 9	- 0.695 5	1.013 9	0.496 5
	算法 4	1.491 5	- 0.709 4	0.975 2	0.509 8
	ICPCO	1.500 3	- 0.699 6	1.000 1	0.500 2
sin t	算法 1	1.456 8	- 0.721 6	1.056 0	0.529 1
	算法 2	1.486 8	- 0.708 9	1.071 6	0.493 3
	算法 3	1.489 0	- 0.709 1	1.040 8	0.501 9
	算法 4	1.490 5	- 0.709 8	0.968 2	0.510 6
	ICPCO	1.499 8	- 0.700 2	0.999 9	0.500 1
精确值		1.5	- 0.7	1.0	0.5

6　结　　论

ICPCO 算法采取并行混沌优化机制,以信息交互来寻找适应度更优的解. 并行优化解分布分散时,采取竞争式信息交互,由整体最优解引导并行优化解在其较大的邻域进行寻优;并行优化解分布集中时,采取协作式信息交互,由单纯形法进行精确最优解搜索. ICPCO 算法全局搜索能力强,由于采取了并行混沌机制,收敛稳定性强. 信息交互方式完善了寻优途径,提高了整体优化效率.

参考文献(References)

[1] 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.
(Li Bing, Jiang Wei-sun. Chaos optimization method and its application [J]. Control Theory and Applications, 1997, 14(4) :613-615.)

[2] 修春波, 刘向东, 张宇河. 双混沌机制优化方法及其应用[J]. 控制与决策, 2003 ,18(6) : 724-726.
(Xiu Chun-bo, Liu Xiang-dong, Zhang Yu-he. Optimization algorithm using two kinds of chaos and its application[J]. Control and Decision, 2003 ,18(6) : 724-726.)

[3] Sun Changzhi, Chen Zhifei, Li Hongmei. Chaotic optimization and Taboo search algorithms for design of underwater thruster motor [C]. 6th Int Conf on Electrical Machines and Systems. Beijing, 2003: 149-152.

[4] 张志新, 张明廉. 基于并行混沌和单纯形法的混合全局优化算法[J]. 系统仿真学报, 2004, 16(1) : 35-37.
(Zhang Zhi-xin, Zhang Ming-lian. A hybrid global optimization algorithm based on parallel chaos optimization and simplex method [J]. J of System Simulation, 2004, 16(1) : 35-37.)

(下转第 1043 页)

$$\begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix} d(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f(k) + p_1(k) \begin{bmatrix} -0.1 & 0 \\ 0 & -0.2 \end{bmatrix} x(k),$$

$$y(k) = [1 \quad 0]\, x(k),$$

其中 $p_1(k)$ 在 $[-0.5, 0.5]$ 上服从均匀分布,即均值为 0,方差为 0.288 7.

应用定理 1 设计如式(7) 所示的残差产生器可得

$$H = [-1.025\,1 \quad 0.206\,2 \quad 0.305\,9]^{\mathrm{T}},$$

进一步,取 $N = 100$,在 $\|d\|_{2,K} = 1$ 的假设条件下可计算对应于 100 个 $p_1(k)$ 样本的残差评价函数值 $J_s^{(j)}(r)$, $j = 1, 2, \cdots, 100$. 分别应用算法 1 和算法 2 计算得:如果取 $J_{th} = 0.465\,1$,则按照式(17) 计算的 FAR_e 为 10 %;如果 FAR_a 设置为 15 %,则 J_{th} 可选取为 0.462 1.

5　结　　语

本文提出了一种解决受乘性随机噪声影响的一类线性不确定离散时间系统鲁棒故障检测问题的 LMI 方法. 选用基于观测器的 FDF 作为残差产生系统,把故障估计用作残差信号,将 FDF 设计归结为 H_∞-滤波问题. 设计均方稳定的 FDF,使残差与故障之间的误差在均方意义下能量最小. 应用 LMI 方法推导证明了问题可解的充分条件,通过求解 LMI 得到观测器增益矩阵的解. 另外,对于残差评价问题也进行了研究. 研究表明,给定残差评价函数和阈值,通过对两者的比较可判断故障的发生,并可进一步估计系统的实际故障误报率;通过选择合适的阈值,可确保系统的实际故障误报率低于指定的容许误报率 FAR_a.

参考文献(References)

[1] Chen J, Patton R J. Robust model-based fault diagnosis for dynamic systems [M]. Boston: Kluwer Academic Publishers, 1999.

[2] Frank P M, Ding S X, Koppen-Seliger B. Current developments in the theory of FDI [C]. Proc of the SAFEPROCESS 2000. Budapest: Elsevier Science, 2000: 16-27.

[3] Isermann R. Model-based fault-detection and diagnosis-status and applications[J]. Annual Reviews in Control, 2005, 29(1): 71-85.

[4] Chen J, Patton R J. Standard H_∞ filtering formulation of robust fault detection [C]. Proc of the SAFEPROCESS 2000. Budapest: Elsevier Science, 2000: 256-261.

[5] Zhong M, Ye H, Shi P, et al. Fault detection for Markovian jump systems[J]. IEE Proc Control Theory and Application, 2005, 152(4): 397-402.

[6] Zhong M, Ding S X, Lam J, et al. LMI approach to design robust fault detection filter for uncertain LTI systems[J]. Automatica, 2003, 39(3): 543-550.

[7] Ding S X, Zhang P, Frank P M. Application of probabilistic robustness technique to the fault detection system design [C]. Proc of the 42th IEEE CDC. Hawaii: IEEE Press, 2003: 972-977.

[8] Boyd S, Ghaoyi L El, Feron E, et al. Linear matrix inequalities in systems and control theory [M]. Philadelphia: SIAM, 1994.

[9] Kailath T, Sayed A H, Hassibi B. Linear estimation [M]. Englewood: Prentice-Hall, 2000.

(上接第 1031 页)

[5] Zou Xiufen, Wang Minting, Zhou Anmin, et al. Evolutionary optimization based on chaotic sequence in dynamic environments [C]. Proc of IEEE Int Conf on Networking, Sensing &Control. Taiwan, 2004: 1364-1369.

[6] Wang L, Smith K. On chaotic simulated annealing[J]. IEEE Trans on Neural Networks, 1998, 9(4): 716-718.

[7] Kwok T, Smith K A. An unified framework for chaotic neural-network approaches to combinatorial optimization [J]. IEEE Trans on Neural Networks, 1999, 10(4): 978-981.

[8] Chen Luonan, Aihara Kazuyuki. Global searching ability of chaotic neural networks [J]. IEEE Trans on Circuits and Systems-I, 1999, 46(8): 974-993.

[9] Di He, Chen He, Ling-ge Jiang, et al. Chaotic characteristics of a one-dimensional iterative map with infinite collapses [J]. IEEE Trans on Circuits and Systems- I, 2001, 48(7): 900-906.

[10] Guo Guanqi, Yu Shouyi. Evolutionary parallel local search for function optimization [J]. IEEE Trans on Systems, Man & Cybernetics, 2003, 33(6): 864-876.