

文章编号:1001-9081(2009)09-2483-04

混合粒子群算法优化分数阶 PID 控制参数研究

胡海波,黄友锐

(安徽理工大学 电气与信息工程学院,安徽 淮南 232001)

(lengxing0351@yahoo.com.cn)

摘要:分数阶比例—积分—微分(PID)控制器是一种把 PID 控制器的整数阶次推广到分数的比例、积分、微分控制器,它比传统的 PID 控制器更能精确地控制复杂的被控系统。而参数的取值对控制效果的好坏起着决定性作用,为此提出了一种混合粒子群算法 BFA-PSO 优化参数值。该算法将具有趋化、繁殖和驱散特点的细菌觅食算法和参数少,易于优化的粒子群算法相结合来计算精确的分数阶 PID 控制器的参数值。通过对传统 PID 控制器和分数阶 PID 控制器参数优化的实验仿真,结果表明基于该算法的分数阶 PID 控制不仅无超调量、收敛速度快,而且鲁棒性强、收敛精度高,可用于控制不同的对象和过程。

关键词:分数阶 PID 控制器;粒子群算法;细菌觅食算法;参数优化;鲁棒性

中图分类号: TP18 **文献标志码:** A

Research of fractional order PID controller using hybrid particle swarm optimization

HU Hai-bo, HUANG You-rui

(College of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan Anhui 232001, China)

Abstract: Fractional order Proportional-Integral-Derivative (PID) controller is a PID controller to the fractional Proportional-Integral-Derivative controller; it can more precisely control the complicated system than the traditional PID controller. The parameter values play a decisive role in the effects of control. A new hybrid algorithm was proposed. The Bacterial Foraging Algorithm and Particle Swarm Optimization (BFA-PSO) were combined to calculate the precise fractional order controller parameter values, which used the chemotactic, reproduction and elimination-dispersal of the BFA and combined the merits of few parameter and being easy to optimize. The simulation results show that the algorithm based on the fractional order PID control is of not only non-overshoot and fast convergence but also robustness and high precision, so it can be used to control different kinds of objects and processes.

Key words: fractional order Proportional-Integral-Derivative (PID) controller; Particle Swarm Optimization (PSO); Bacterial Foraging Algorithm (BFA); parameter optimization; robustness

0 引言

I. Podlubny 教授于 1999 年提出的分数阶 PID 控制器^[1]($PI^\lambda D^\mu$) 精确地分析出分数阶控制系统的动态响应,也证明其用于分数阶对象可取得比传统 PID 控制器更好的性能。因为 $PI^\lambda D^\mu$ 控制器不仅有传统 PID 控制器的 K_p, K_i, K_d 这三个参数,还有积分阶次 λ 和微分阶次 μ 两个可调参数,使得整个控制器参数的调节范围扩大、调整难度增大,却能够更灵活、精确地控制受控对象。许多研究者都不同程度的研究了 $PI^\lambda D^\mu$ 控制器的鲁棒性能及其应用,如在给定相位裕量和幅值裕量下 $PI^\lambda D^\mu$ 控制器的设计^[2], PD^μ 控制器的设计^[3], 分数阶 PI 和 PID 控制器的稳定性的研究^[4], 以及一种运用在实际系统中更为有效的分数阶控制器^[5]。由于简单的分数阶控制器比整数阶控制器有更优的动态性能和鲁棒性,所以分数阶控制器仍然用在整数阶模型对象上,但这都是在整定模型给出的情况下通过实验的方法调节出最佳参数值,由于控制的复杂性,仍然没有系统的方法来设定积分阶次 λ 和微分阶次 μ 的值。

现有多种方法可求得各个参数值,如仿生算法中的 PSO 优化整数 PID 的参数^[6]。改进 PSO 设计 $PI^\lambda D^\mu$ 控制器^[7],但是

其 200 次的迭代次数使得优化时间长,收敛速度变慢。文献 [8] 也是基于 PSO 设计 $PI^\lambda D^\mu$ 控制器,通过整数 PID 和分数阶 PID 控制的对比,表明了 $PI^\lambda D^\mu$ 控制器的优越性。但因其采用基本 PSO,收敛速度和精度值都不是很好。

PID 控制要求精确地控制动态响应和实时监控,故本文将具有群体智能并行算法,易跳出局部极小值等优点的细菌觅食算法和模型简单,易于操作的粒子群算法相结合,既避免了寻优过程中过早的陷入局部最优,又保证了其具有较快的收敛速度和更高的收敛精度。通过优化 PID 控制器的 $[K_p, K_i, K_d]$ 3 个参数和 $PI^\lambda D^\mu$ 控制器的 $[K_p, K_i, K_d, \lambda, \mu]$ 的 5 个参数的实验仿真,结果表明 $PI^\lambda D^\mu$ 控制比 PID 控制更精准,且 BFA-PSO 采用较少的迭代次数便能解出精确参数值。通过 PSO 和 BFA-PSO 的比较得出用 BFA-PSO 更适用于求解 $PI^\lambda D^\mu$ 控制器参数精确值,基于 BFA-PSO 的 $PI^\lambda D^\mu$ 控制无超调量,并有很好的收敛精度和较快的响应速度,大大提高了参数寻优的效率。

1 混合粒子群优化算法

1.1 粒子群优化算法

粒子群优化算法是 (Particle Swarm Optimization, PSO)^[9]

收稿日期:2009-03-25;修回日期:2009-05-10。

作者简介:胡海波(1985-),女,山西太原人,硕士研究生,主要研究方向:控制理论、控制工程;黄友锐(1971-),男,安徽淮南人,教授,主要研究方向:计算机控制、智能信息处理。

是由美国的 Kennedy 和 Eberhart 于 1995 年提出的,源于对鸟群和鱼群群体觅食运动行为的模拟,它是一种全局优化进化算法。PSO 求解优化问题时,每个问题的解被看作是搜索空间中一个没有质量和体积的微粒。所有粒子都有一个被优化的函数决定其适应度值,且还有一个速度决定粒子飞行的方向和距离,该飞行速度根据个体的飞行经验和群体的飞行经验进行动态的调整。在每一次迭代中,粒子通过跟踪两个“极值”来更新自己:一个是个体极值 P_{id} ,即粒子自身目前所找到的最优解;另一个是全局极值 P_{gd} ,即整个种群目前找到的最优解。在找到这两个最优解时,粒子根据如下两个公式来更新自己的速度和位置:

$$V_{id}^t = \omega \cdot V_{id}^{t-1} + c_1 \cdot rand_1() \cdot (P_{id}^{t-1} - X_{id}^{t-1}) + c_2 \cdot rand_2() \cdot (P_{gd}^{t-1} - X_{id}^{t-1}) \quad (1)$$

$$X_{id}^t = X_{id}^{t-1} + V_{id}^t \quad (2)$$

其中: X 代表粒子当前位置; V 代表粒子的速度; ω 为惯性权重系数,它使粒子保持运动特性,使其有扩展搜索空间的趋势,有能力探索新的区域。研究发现较大惯性权重 ω 值有利于跳出局部极小点,较小的惯性权重 ω 值有利于算法收敛,因此提出自适应调整 ω 的策略,即随着迭代的进行,线性减小 ω 的值。对全局搜索,通常的好方法是在前期有较高的搜索能力,而在后期有较高的开发能力以加快收敛速度。通常采用如下公式来调整:

$$\omega = \omega - iter(\omega_{max} - \omega_{min})/iter_{max} \quad (3)$$

其中: $iter$ 指当前的代数, $iter_{max}$ 指运行的最大迭代次数, ω_{max} , ω_{min} 是惯性权重系数的最大值和最小值,可取由 1.4 到 0,或由 0.9 到 0.4,或由 0.95 到 0.2。加速常数 C_1 和 C_2 代表将每个粒子推向 P_{id} 和 P_{gd} 位置的统计加速项的权重,一般取(0,2)之间的随机数, $rand$ 是产生(0,1)之间随机数的函数。

1.2 细菌觅食算法

细菌觅食算法 (Bacterial Foraging Algorithm, BFA)^[10] 是由 K. M. Passino 于 2002 年提出的一种新型仿生类算法,该算法基于 E. coli 大肠杆菌在人体肠道内吞噬食物的行为。大肠杆菌在觅食过程中,通过身体的鞭毛顺时针或逆时针的转动来推动自身的运行或翻转,并以 E/T (E 是细菌群得到的能量, T 是搜寻的时间) 的速率一步步地寻找营养最丰富的地方。细菌群的觅食行为可以看作是不断优化的过程,优化问题的解对应搜索空间内细菌的状态,即优化函数适应度值。在整个过程中,细菌的生命周期会结束或者被杀死,也可能在不同的温度下繁殖。BFA 包括趋化 (chemotactic), 繁殖 (reproduction), 驱散 (elimination-dispersal) 三个步骤。

1) 趋化是细菌朝向含有营养的区域聚集的行为。在趋化过程中,细菌运动模式包括翻转 (tumble) 和前进 (swim)。翻转是细菌向任意方向移动单位步长,其更新公式为: $x(j+1, k, l) = x(j, k, l) + c(i)q(j)$ 。 $c(i)$ 是细菌在任意方向被指定的步长的大小,该值要大于 0。每当细菌完成一次翻转后,检查适应度值是否改变,若适应度值得到改善,细菌将沿同一方向继续移动若干步,如此循环直至适应度值不再改善,或达到设定的移动步数临界值,此过程定义为前进。

2) 当满足设定的趋化次数,细菌将进行分裂繁殖。细菌的分裂繁殖过程遵循自然界“优胜劣汰,适者生存”原则。适应度值好(即细菌终生觅食过程中消耗小)的细菌分裂成两个子细菌,子细菌继承了母细菌的特性,拥有和母细菌相同的位置和步长。而适应度值差(即细菌终生觅食过程中消耗大)或者生命周期结束的细菌将被驱散。为了确保细菌总数

的不变,将驱散相同数目的细菌,在每次驱散中,细菌或者死亡或者以一定概率被驱散到空间里的任意位置。

$$P\{j, k, l\} = \{x^i(j, k, l) \mid i = 1, 2, \dots, s\} \quad (4)$$

式(4)说明 s 个细菌在第 j 个趋化步,第 k 个复制步和第 l 个驱散步中第 i 个细菌的位置。

1.3 混合粒子群优化算法

在求解优化问题时,粒子群在解空间内不断迭代过程中,如果某粒子发现某个当前最优位置时,其他粒子将迅速向其靠拢,如果该最优位置为局部最优点,粒子群就无法在解空间内重新搜索,使粒子群聚集在局部最优处,过早的陷入局部收敛中。为了改进这种缺点,本文将 PSO 和具有趋化、繁殖和驱散特点的细菌觅食算法相结合,根据式(5)来混合优化。

$$J(i, j, R, ED) = F(P(\cdot, i, j, R, ED)) \quad (5)$$

其中: i, j 分别代表细菌(也是粒子)的数量和趋化变量, R 是细菌的繁殖变量, ED 是细菌的驱散变量, F 函数是适应度函数。

细菌觅食算法特有的趋化过程可以加强细菌的局部搜索能力,繁殖过程可以加快细菌的搜索速度,驱散过程是保证能搜索到全局最优位置。

在解空间内,细菌的更新是由趋化、繁殖和驱散决定,相应的粒子也会根据细菌的更新和适应度函数来更新自己的位置和速度,根据式(6)、(7)不断更新个体极值和全局极值,直到迭代结束。

$$plocal_position(\cdot, i, j) = current_position(\cdot, i, index(i, \cdot)) \quad (6)$$

$$global_position = current_position(\cdot, I, index(I, \cdot)) \quad (7)$$

基于以上可以看出,具有全局搜索能力的细菌觅食算法根据细菌的更新来带动粒子群的更新,原本因为根据适应度函数更新而易陷入局部收敛中的 PSO,现多了细菌的趋化、繁殖和驱散操作来更新自己的位置和速度,弥补了粒子群算法“早熟”的缺陷,从而求解出全局最优值。

2 BFA-PSO 优化分数阶 PID 控制器参数

2.1 分数阶 PID 控制器

分数阶微积分是任意阶微分和积分的理论,随着其在控制中的研究,出现了控制理论的新领域,即分数阶控制理论与分数阶控制器。分数阶控制系统是基于分数阶微积分控制理论建立起来的。以分数阶控制理论的角度,闭环控制系统有 4 类:整数阶对象模型与整数阶控制器,整数阶对象模型与分数阶控制器,分数阶对象模型与整数阶控制器,分数阶对象模型与分数阶控制器。分数阶 PID 控制器是整数 PID 控制器的一般化,将整数阶的积分项 λ 和微分项 μ 的阶次扩展到了分数领域,其中积分阶次 λ 和微分阶次 μ 可以是任意的实数。它的时域表达式为:

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) \quad (8)$$

其中 $D_t^{\alpha} \equiv {}_a^c D_t^{\alpha}$ 为 Caputo 定义, $\lambda > 0$ 为积分阶次, $\mu > 0$ 为微分阶次,可以为任意实数。 K_p, K_i, K_d 是控制器的控制参数。

实际上,在分数阶微积分的定义下,分数阶 $PI^{\lambda}D^{\mu}$ 控制器是一个无限维数的滤波器。其设计原理是根据系统的控制性能指标和闭环特征方程的零极点来设计和优化控制参数的。

对式(8)求拉普拉斯变化为:

$$L\{{}_a^c D_t^{\alpha} f(t)\} = S^{\alpha} F(s) - \sum_{k=0}^{N-1} S^{\alpha-k-1} f^{(k)}(0) \quad (9)$$

并可以得到分数阶 PID 控制器的传递函数为:

$$G(s) = K_p + K_i S^{-\lambda} + K_d S^{\mu} \quad (10)$$

当 $\lambda = 1$ 和 $\mu = 1$ 就是整数阶 PID 控制器。当 $\lambda = 0$ 时为 PD^μ 控制器。当 $\mu = 0$ 时为 PI^λ 控制器。当 $K_i = 0, \mu = 1$ 时为 PD 控制器。当 $K_d = 0, \lambda = 1$ 时为 PI 控制器。可见,以上的任一种控制器都是分数阶 PID 控制器的一种特殊情况。通过合理的整定参数,分数阶 PID 控制器能更好地、更精确地提高系统的控制效果。

2.2 BFA-PSO 优化 $PI^\lambda D^\mu$ 控制器参数

分数阶 PID 控制器的优越性显而易见,但是正是如此调节参数的难度会更大,基于这种情况考虑,本文用 BFA-PSO 来优化分数阶 $PI^\lambda D^\mu$ 控制器的 5 个参数,得到其精确值。 $PI^\lambda D^\mu$ 控制器的设计实际上是多维函数优化问题。对于 $PI^\lambda D^\mu$ 控制器 BFA-PSO 采用的是实数编码,参数寻优中的粒子可以直接编码为 $[K_p, K_i, K_d, \lambda, \mu]$,控制参数优化的目的在于使控制偏差趋于零,有较小的超调量和较快的响应速度。为得到满意的过渡过程动态特性,采用误差绝对时间积分性能指标为参数选择的最小目标函数。为了防止控制能量过大,在目标函数中加入控制输入的平方项,即:

$$J = \int_0^\infty (\omega_1 |e(t)| + \omega_2 u^2(t)) dt + \omega_3 t_s \quad (11)$$

其中 $e(t)$ 为控制误差, $u(t)$ 为控制器输出, $\omega_1, \omega_2, \omega_3$ 为权值, t_s 为上升时间。

为了避免系统超调,采用惩罚功能,即一旦产生超调,超调量将作为最优指标的一项,即:

$$J = \int_0^\infty (\omega_1 |e(t)| + \omega_2 u^2(t) + \omega_4 |ey(t)|) dt + \omega_3 t_s; \quad ey(t) < 0 \quad (12)$$

其中 $\omega_4 \gg \omega_1, ey(t) = y(t) - e(t-1), y(t)$ 为被控对象输出。

3 实验仿真

BFA-PSO 实现步骤如下。

第 1 步 初始化。设定粒子群数目大小,随机确定粒子的位置和速度,设置各个参数的值。

第 2 步 初始化。设定细菌群的数目大小位置,确定趋化、繁殖步长,驱散、繁殖概率等各个参数的值。

第 3 步 将每个粒子(细菌)的位置向量依次作为 $PI^\lambda D^\mu$ 控制器参数,并计算粒子群的适应度值,初始化 P_i, P_g 。

第 4 步 细菌翻转,搜寻最优位置,并同时更新粒子群的适应度值和 P_i, P_g 的值。

第 5 步 判断粒子群的适应度值是否得到改善,是则继续进行下一步;否则转到第 4 步。

第 6 步 细菌是否达到趋化步数,是则细菌繁殖和驱散,否则转到第 4 步。

第 7 步 终生消耗最小的细菌分裂,消耗多的细菌以一定概率被驱散,产生新的细菌群,转到第 4 步。

第 8 步 判断是否满足最大迭代次数,是则结束;否则转到第 4 步。

BFA-PSO 参数设置:粒子群(细菌)总数 $N = 60$,粒子大小 $D = 5$ ($[K_p, K_i, K_d, \lambda, \mu]$ 5 个参数), C_1 和 C_2 取值为 2,惯性系数 $\omega_{max} = 0.9, \omega_{min} = 0.4, \omega_1 = 0.999, \omega_2 = 0.001, \omega_3 = 2.0, \omega_4 = 100$,采样时间为 $0.001 s^{[11]}$,最大迭代次数为 60 次。

为了对比性,本文采用文献[8]给出的一个被控对象的例子,即 $\frac{400}{s^2 + 50s}$ 。

BFA-PSO 优化 $PI^\lambda D^\mu$ 控制器参数,解出目标函数值 $J = 24.0959$,求得最优解为:

$K_p = 15.9587, K_i = 0.1024, K_d = 11.0503, \mu = 0.59, \lambda = -1.5648$ 。即 $PI^\lambda D^\mu$ 控制器的传递函数为: $15.9587 + 0.1024s^{-1.5648} + 11.0503s^{0.59}$ 。

令 $\lambda = \mu = 1$, BFA-PSO 优化 PID 控制器参数,求得最优解为: $K_p = 3.885, K_i = 0.0013, K_d = 2.550$ 。即 PID 控制器的传递函数为: $3.885 + 0.0013s^{-1} + 2.550s$ 。

如图 1,被控对象的阶跃响应是在 0.6 s 趋于稳定,用 BFA-PSO 优化 $PI^\lambda D^\mu$ 控制器参数,图 2 显示快到 20 代时函数值就基本不变了。从图 3 可以明显看出,在迭代次数仅为 60 次的情况下,用 BFA-PSO 优化 $PI^\lambda D^\mu$ 控制器的 5 个参数后,被控对象的阶跃响应大约在 0.06 s 就收敛了,而且没有超调,并得到了精确的参数值,而在同样的参数下优化 PID 控制器的 3 个参数,在 0.2 s 才渐渐趋于平稳,还有超调量。可见用 BFA-PSO 优化 $PI^\lambda D^\mu$ 控制比整数 PID 有更优的收敛精度和更快的收敛速度。图 4 是用 PSO^[8] 和 BFA-PSO 都优化 $PI^\lambda D^\mu$ 控制器参数,通过比较得出 PSO 优化后不仅有超调量,还有微小的振荡,所以 BFA-PSO 比 PSO 更适合优化整定分数阶控制器,使被控对象有更好的动态响应。

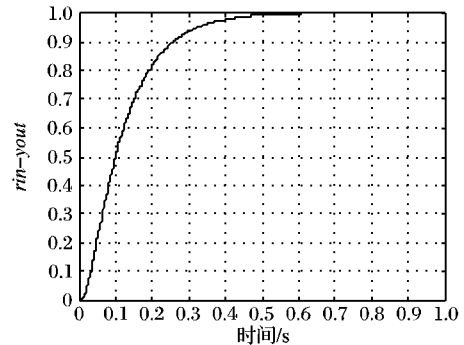


图 1 被控对象未受控制的阶跃响应

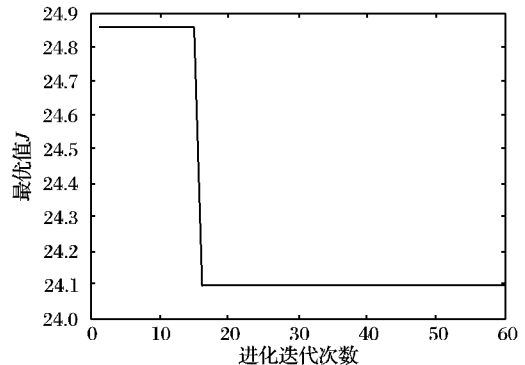


图 2 BFA-PSO 优化 $PI^\lambda D^\mu$ 后目标函数 J 的收敛曲线

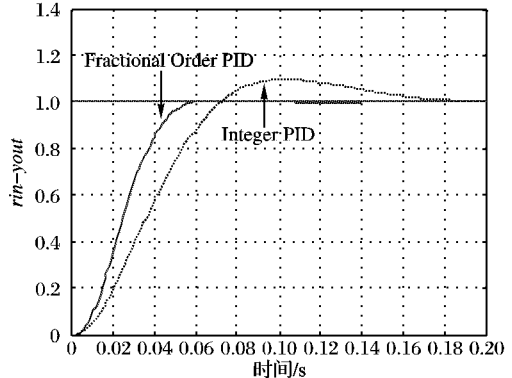


图 3 BFA-PSO 优化 $PI^\lambda D^\mu$ 和 PID 参数阶跃跟踪比较

4 结语

随着分数阶控制理论的研究和发展,分数阶控制已成为

近年来控制领域研究的一个新热点,对于许多复杂的实际系统,整数阶控制器模型已经不能很好地提高系统的控制性能,这就有必要采用分数阶控制系统来精确地描述系统的特性。基于此本文将基于细菌觅食算法的混合粒子群算法,即 BFA-PSO 来整定优化 $PI^\lambda D^\mu$ 控制器和 PID 控制器的参数,有效提高了 PSO 的计算效率。实验仿真结果表明分数阶 PID 控制器可以取得比传统 PID 控制器更好的动态性能和鲁棒性,而且通过 BFA-PSO 与 PSO 的比较得出,该算法更适合于分数阶控制器参数的求解,证明基于 BFA-PSO 的 $PI^\lambda D^\mu$ 控制有较好的收敛稳定性、更快的响应速度和更高的精度值。

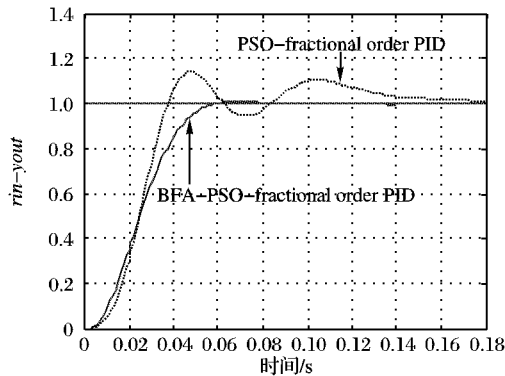


图 4 PSO 和 BFA-PSO 优化 $PI^\lambda D^\mu$ 参数动态响应

参考文献:

[1] PODLUBNY I. Fractional-order systems and $PI^\lambda D^\mu$ -controllers[J]. IEEE Transactions on Automatic Control, 1999, 44(1): 208 - 214.
 [2] ZHAO CHUN-NA, XUE DING-YU, CHEN YANG-QUAN. A fractional order PID tuning algorithm for a class of fractional order plants [C]// ICMA2005: Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation. Washington, DC: IEEE

Computer Society, 2005: 216 - 221.
 [3] LI HONG-SHENG, CHEN YANG-QUAN. A fractional order proportional and derivative (FOPD) controller tuning algorithm[C]// CCDC 2008: 2008 Chinese Control and Decision Conference. Yantai, Shandong, China: [s. n.], 2008: 4059 - 4063.
 [4] HAMAMCI S E. Stabilization using fractional-order PI and PID controllers [J]. Nonlinear Dynamics, 2008, 51(1/2): 329 - 343.
 [5] MONJE C A, VINAGRE B M, CHEN Y Q, et al. Optimal tunings for fractional $PI^\lambda D^\mu$ [C]// Proceedings of the 1st IFAC Symposium on Fractional Differentiation and its Applications. Bordeaux, France: [s. n.], 2004: 675 - 686.
 [6] CHEN S F. Particle swarm optimization for PID controllers with robust testing [C]// Proceedings of the 6th International Conference on Machine Learning and Cybernetics. Hong Kong, China: [s. n.], 2007: 956 - 961.
 [7] CAO JUN-YI, CAO BING-GANG. Design of fractional order controller based on particle swarm optimization [J]. International Journal of Control, Automation, and Systems, 2006, 4(6): 775 - 781.
 [8] MAITI D, BISWAS S, KONAR A. Design of a fractional order PID controller using particle swarm optimization technique [EB/OL]. [2009 - 01 - 05]. <http://arxiv.org/pdf/0810.3776>.
 [9] KENNEDY J, EBERHART R C. Particle swarm optimization[C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth, WA, Australia: IEEE Computer Society, 1995: 1942 - 1948.
 [10] LIU Y, PASSINO K, MPASSINO K M. Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviors [J]. Journal of Optimization Theory and Applications, 2002, 115(3): 603 - 628.
 [11] 刘金琨. 先进 PID 控制及其 Matlab 仿真[M]. 北京: 电子工业出版社, 2003.

(上接第 2482 页)

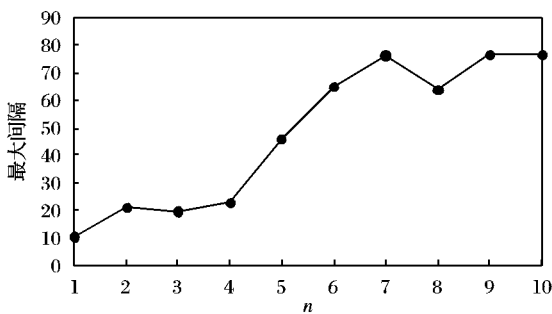


图 5 枚举算法中聚类个数对于最大间隔的影响

表 2 子类合并情况

聚类数目	间隔	枚举次数
100	0	1
80	23.6	1
20	56.2	2
7	60.2	4
5	76.5	16

表 2 描述了算法在执行过程中,子类的合并情况,最初把 100 个数据认为 100 个子类,通过任意划分得到间隔为 23.6,合并那些类间最小距离小于 57.2 的数据为一类,由于我们的枚举是任意的,所以有些时候不能很准确地直接找到一个间隔进行聚类,比如聚类数目为 7 的时候,我们枚举了 4 次才找到一个更大的间隔使得聚类数目变成 5。间隔随着聚类数目

的变少而逐步增加,因为距离小于 2 倍间隔的类已被合并。

4 结语

为了提出一种更有效的算法来解决支持向量机反问题,本文根据间隔和类间最短距离的关系提出了一种基于聚类合并的支持向量机反问题求解算法,试验证明本算法有效地解决了求解支持向量机反问题高时间复杂度的问题。

参考文献:

[1] WANG XI-ZHAO, HE QIANG, CHEN DE-GANG, et al. A genetic algorithm for solving the inverse problem of support vector machines [J]. Neurocomputing, 2005, 68: 225 - 238.
 [2] HAO PEI-YI, CHIANG J H, TU YI-KUN. Hierarchically SVM classification based on support vector clustering method and its application to document categorization [J]. Expert Systems with Applications, 2007, 33(3): 627 - 635.
 [3] BELSOLE E, SAUVAGEOT J L, PRATT G W, et al. Merging clusters of galaxies observed with XMM-Newton [J]. Advances in Space Research, 2005, 36(4): 630 - 635.
 [4] ZHU JIE. An improved algorithm for inverse problem of SVMs based on clustering[C]// Proceedings of the 5th International Conference on Machine Learning and Cybernetics. Washington, DC: IEEE Computer Society, 2006: 4694 - 4698.
 [5] CERVANTES J, LI XIAO-OU. Support vector machine classification for large data sets via minimum enclosing ball clustering [J]. Neurocomputing, 2008, 71(4/6): 611 - 619.