

文章编号: 1001-0920(2008)07-0756-06

一种精英退火微粒群算法

李军军, 肖健梅, 王锡淮

(上海海事大学 电气自动化系, 上海 200135)

摘要: 提出一种针对部分较优微粒进行退火操作的精英退火微粒群算法. 在退火操作中, 结合 Logistic 方程的特点设计了一种新的错位调整方式, 对当前已知最优区域重点搜索. 该算法能增强算法的探索和开发能力, 避免计算量过度增加. 典型测试函数结果显示, 该方法可同时提高算法的搜索速度和搜索精度. 将基于该方法的 PID 控制器应用于发电机电压调节系统(AVR), 计算结果表明, 该 PID 控制器可以获得更为满意的控制性能指标.

关键词: 微粒群优化; 精英退火; 函数优化; AVR 系统

中图分类号: TP18 **文献标识码:** A

Elitist annealing particle swarm optimization algorithm

LI Jun-jun, XIAO Jian-mei, WANG Xi-huai

(Department of Electrical and Automation, Shanghai Maritime University, Shanghai 200135, China. Correspondent: WANG Xi-huai, E-mail: wxh@shmtu.edu.cn)

Abstract: A sort of elitist annealing particle swarm optimization algorithm is proposed, in which part of the particle swarm is annealed. A new kind of annealing method based on the character of the Logistic function is designed. Then the current best space is searched more detailedly. This algorithm can enhance the exploration and exploitation ability of the algorithm. At the same time, the computation time is well controlled. Typical function optimization problem results show that this method possesses good convergent performance with faster convergent rate. And practical automatic voltage regulator (AVR) system test results verify that the proposed method can obtain more satisfied performance criterion.

Key words: Particle swarm optimization; Elitist annealing; Function optimization; AVR system

1 引言

微粒群优化(PSO)算法最初是为了图形化地模拟鸟群优美而不可预测的运动. Eberhart 和 Kennedy^[1]于 1995 年提出了 PSO 的最初版本. 之后, Shi 和 Eberhart^[2]引入惯性权重, 形成了当前的标准版本. 标准微粒群优化算法具有较强的全局搜索能力, 同时也存在易陷入局部极值的缺点^[3].

模拟退火算法(SA)由 Kirkpatrick 于 20 世纪 80 年代初提出, 是基于金属退火的机理建立起的一种全局最优方法, 在理想状态下可以得到全局最优解.

文献[4-6]将微粒群算法和模拟退火算法相结合, 提出了新的混合优化算法. 这些混合算法虽能提高算法性能, 但却不可避免地增加了算法的复杂度,

计算效率不高. 本文针对标准微粒群算法的特点, 以部分较优微粒作为精英微粒, 结合 Logistic 方程的特点设计了一种新的位置生成方式, 对精英微粒进行一种新的模拟退火操作. 该方法能提高算法的探索和开发能力, 避免算法陷入局部极值, 且所增加的计算量有限.

2 标准微粒群优化算法

标准微粒群优化算法首先在解空间内随机初始化一组微粒, 这些微粒经过若干次迭代后找到最优解. 迭代公式如下:

$$V_{id}^{t+1} = V_{id}^t + c_1 R(X_{id}^{pb} - X_{id}^t) + c_2 R(X_d^{gb} - X_{id}^t), \quad (1)$$

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

其中: $d = 1, 2, \dots, D$; $t (t = 1, 2, \dots)$ 为迭代次数;

收稿日期: 2007-04-26; 修回日期: 2007-10-19.

基金项目: 上海市科技攻关项目(06DZ11202); 上海市教委重点学科建设项目(J50602); 上海市教委科研项目(08ZZ78).

作者简介: 李军军(1981—), 男, 江苏南通人, 博士生, 从事电力系统智能优化等研究; 王锡淮(1961—), 男, 江苏淮安人, 教授, 博士生导师, 从事复杂系统建模、优化控制等研究.

为惯性权重; c_1 和 c_2 分别为认知参数和社会参数; R 为 $[0, 1]$ 之间的随机数; X_{id}^{pb} 是微粒在其搜索过程中到达过的最优位置; X_d^{gb} 是整个微粒群目前找到的最优位置. 此外, 微粒的速度 V_{id} 被一个最大速度 $V_{max, d}$ 所限制.

3 精英退火微粒群算法

标准微粒群算法具有较强的全局搜索能力, 但同时也容易陷入局部极值. 而模拟退火算法具有较强的局部搜索能力, 并能使搜索过程避免陷入局部最优解, 与 PSO 具有良好的互补性. 但如果在微粒群算法中, 对所有微粒都采用模拟退火操作, 虽能提高算法的搜索精度, 但计算量却大大增加.

研究发现, 在微粒群迭代过程中, 与 X^{gb} 较为接近的部分微粒的调整, 对于提高 PSO 的计算效率能起到重要作用. 若这部分微粒出现调整, 则可能找到比 X_i^{pb} 或 X^{gb} 更优的位置, 更新 X_i^{pb} 或 X^{gb} , 从而提高算法的搜索速度; 同时也可能使算法跳出局部极值. 当这部分微粒的数量不大时, 它们的调整并不会给计算增加过多的复杂度.

本文提出一种精英退火微粒群算法 (EAPSO). 该算法在微粒群每次迭代结束之前, 增加了与 X^{gb} 较为接近的部分微粒的模拟退火操作, 并将这部分微粒称为精英微粒.

3.1 精英微粒的选取

本文综合考虑微粒的位置和适应度, 取与 X^{gb} 的距离和适应度都较为接近的微粒作为精英微粒.

以求解 $\min f(X)$ 问题为例, 将目标函数作为微粒适应度函数. 每一代结束后, 将所有微粒分别按适应度从低到高、与 X^{gb} 的距离从小到大排序, 获得各微粒的两个排名 P_{fi}^t 和 P_{di}^t ($i = 1, 2, \dots, N$, N 为微粒个数). 这里的各微粒与 X^{gb} 的距离可采用 L_p 范数距离, 即

$$L_p(X_i^t, X^{gb}) = \left(\sum_{d=1}^D |X_{id}^t - X_d^{gb}|^p \right)^{1/p}.$$

为减少计算量, 本文中 $P = 1$, 即采用 Manhattan 距离.

再依据 $P_i^t = P_{fi}^t + P_{di}^t$, 从低到高将所有微粒排序, 获得各微粒的综合排名 P_i^t . 将综合排名在前 $\cdot N$ 位的微粒 X_{ip}^t ($i = 1, 2, \dots, N$) 取出, 作为精英微粒. 其中 \cdot 为精英系数, 表示在种群中精英微粒的比例.

3.2 精英微粒的退火操作

在基本模拟退火算法中, 新位置是在原位置的周围随机产生的, 这种方法比较方便, 但存在很大的盲目性.

文献 [7-10] 将 Logistic 映射引入随机优化算

法, 产生一定长度的 Logistic 混沌序列, 可利用其遍历性和随机性提高搜索能力. 本文通过对 Logistic 方程 [11] 进行分析, 发现了其中变量 x_n, x_{n+1} 的单步变化特点, 并由此设计了一种新的新位置生成方式, 将其应用于精英微粒的模拟退火操作中, 而非产生混沌序列. 该方式可使精英微粒得以进行较为积极的探索开发, 提高算法的搜索速度与精度.

3.2.1 Logistic 方程

Logistic 方程是 Verhulst 于 1838 年提出的, 用于刻画生物群体中个体总量增长的情况. 该方程为

$$x_{n+1} = \mu x_n (1 - x_n). \tag{3}$$

当 $\mu = 4$, $x_n \in [0, 0.5]$ 时, $x_{n+1} \in [0, 1]$. 研究发现, x_n 和 x_{n+1} 与 $[0, 1]$ 区间中点 0.5 的距离大致呈现如下特点: 若 x_n 距离 0.5 较远, 则 x_{n+1} 与 0.5 的距离比 x_n 近; 若 x_n 距离 0.5 较近, 则 x_{n+1} 与 0.5 的距离比 x_n 远. 例如图 1 中, $d_1 > d_2, d_3 > d_4$; 又如如图 2 中, 当 $x_n \in (0, 0.25), (0.25, 0.5]$ 时, $|x_{n+1} - 0.5| / |x_n - 0.5|$ 分别小于或大于 1. 由图 1 和图 2 可知, 若 x_n 距离 0.5 越近, 则 x_{n+1} 与 0.5 的距离比 x_n 越远; 且上述这种较近与较远都是在有限范围内的.

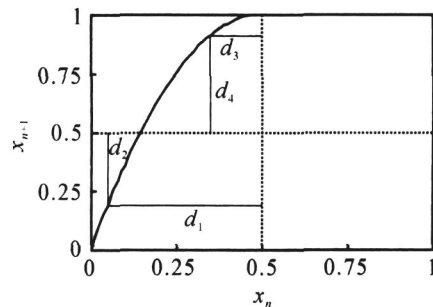


图 1 $x_n \in [0, 0.5]$ 时 x_{n+1} 的曲线图

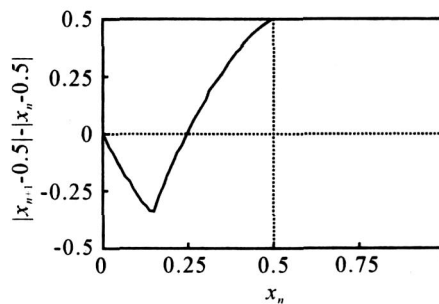


图 2 $x_n \in [0, 0.5]$ 时 $|x_{n+1} - 0.5| / |x_n - 0.5|$ 的曲线图

如果将 X_d^{gb} 映射至 0.5, 精英微粒位置映射到 $[0, 1]$ 区间内, 并作类似于 x_n, x_{n+1} 的尝试性运动, 则精英微粒位置会在一定范围作较大幅度的调整. 即: 距离 X^{gb} 较近的精英微粒作远离 X^{gb} 的尝试性运动, 而距离 X^{gb} 较远的精英微粒作接近 X^{gb} 的尝试性运动. 这样, 精英微粒更有可能发现较 X^{gb} 更优的位置, 也更能增强算法跳出局部极值的能力.

然而 x_{n+1} 是由 x_n 和式 (3) 决定的唯一确定值,

而 x_{n+1} 可根据下式得出:

$$x_{n+1} = \begin{cases} 0.5 + \text{rand}(x_{n+1} - 0.5), \\ x_n \quad [0, 0.25]; \\ 1 - \text{rand}(1 - x_{n+1}), \\ x_n \quad (0.25, 0.5]. \end{cases} \quad (4)$$

x_{n+1} 在图3左半部阴影部分取随机值. x_{n+1} 与 x_n 具有上述同样的特点,且更适合于算法的优化.

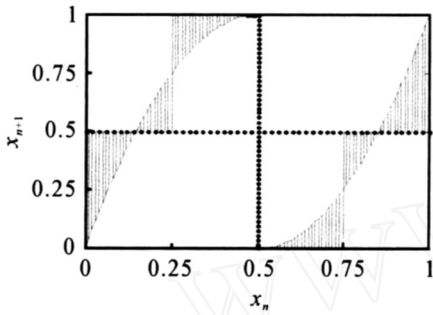


图3 x_{n+1} 位置示意图

为在区间 $[0.5, 1)$ 上同样拥有上述特点,可将式(3)调整为

$$x_{n+1} = \begin{cases} \mu x_n (1 - x_n), & x_n < 0.5; \\ 1 - \mu x_n (1 - x_n), & x_n > 0.5. \end{cases} \quad (5)$$

x_{n+1} 仍由式(4)决定,即 x_{n+1} 在图3中的阴影部分取随机值.

3.2.2 精英微粒新位置的生成

将精英微粒某维映射到 Logistic 方程所在区间时,将该维上与 Logistic 方程所在区间对应的区间称为该维的退火区间;将退火区间的中心和半径分别称为退火中心和退火半径;退火区间可视为当前已知最优区域.

将精英微粒第 d 维 X_{pid}^t ($i = 1, 2, \dots, N$) 与 X_d^{gb} 的最大距离值作为该维的退火半径 R_d^{sa} ,即

$$R_d^{sa} = \max_{i=1,2,\dots,N} \{ |X_{pid}^t - X_d^{gb}| \}, \quad d = 1, 2, \dots, D. \quad (6)$$

对于微粒的每一维,将 X_d^{gb} 作为退火中心,区间 $[X_d^{gb} - R_d^{sa}, X_d^{gb} + R_d^{sa}]$ ($d = 1, 2, \dots, D$) 即为退火区间.在微粒群迭代过程中,该区间的位置及大小是不断变化的.

将该退火区间映射到式(5)和(4)所在的 $[0, 1]$ 区间,即精英微粒 X_{pi}^t 的第 d 维 X_{pid}^t 通过下式映射为 X_{pid0}^t :

$$X_{pid0}^t = 0.5 + \frac{X_{pid}^t - X_d^{gb}}{2R_d^{sa}}. \quad (7)$$

然后,将 X_{pid0}^t 作为式(5)中的 x_n ,并将由式(5)和(4)求出的 x_{n+1} 作为 X_{pid1}^t ,即

$$X_{pid1}^t = \begin{cases} 0.5 + \text{rand}[\mu X_{pid0}^t (1 - X_{pid0}^t) - 0.5], \\ x_n \quad 0.25; \\ 1 - \text{rand}(1 - \mu X_{pid0}^t (1 - X_{pid0}^t)), \\ 0.25 < x_n < 0.5; \\ \text{rand}(1 - \mu X_{pid0}^t (1 - X_{pid0}^t)), \\ 0.5 < x_n < 0.75; \\ 0.5 + \text{rand}[0.5 - \mu X_{pid0}^t (1 - X_{pid0}^t)], \\ x_n \quad 0.25. \end{cases} \quad (8)$$

再按下式将 X_{pid1}^t 映射到微粒的搜索空间:

$$X_{pid}^t = X_d^{gb} + (X_{pid1}^t - 0.5) \cdot 2R_d^{sa}. \quad (9)$$

X_{pi}^t 的所有维都进行上述操作之后,即得到它的新位置 $X_{pi}^t = [X_{pi1}^t, X_{pi2}^t, \dots, X_{piD}^t]$.

3.2.3 精英微粒新位置的接受规则

微粒由原位置 X_{pi}^t 变为新位置 X_{pi}^t 的接受概率由模拟退火中的 Metropolis 规则确定,即

$$p = \begin{cases} 1, & E_{\text{new}} < E_{\text{old}}; \\ \exp\left(-\frac{E_{\text{new}} - E_{\text{old}}}{T(t)}\right), & E_{\text{new}} > E_{\text{old}}. \end{cases} \quad (10)$$

其中: E_{new} 和 E_{old} 分别为微粒新位置和原位置的适应度; $T(t)$ 为 PSO 第 t 代时的温度,采用经典模拟退火算法的降温方式时,则

$$T(t) = \frac{T_0}{\lg(1+t)}. \quad (11)$$

若新位置优于旧位置,则将其适应度值与 X_{id}^{pb} 和 X_d^{gb} 进行比较,如果较优,则更新 X_{id}^{pb} 和 X_d^{gb} .

3.3 EAPSO 计算复杂度分析

EAPSO 主要包含两部分:1) 微粒群按标准 PSO 迭代生成下一代微粒;2) 精英微粒的模拟退火操作.前者为标准 PSO 的时间,后者为增加的时间.微粒群规模为 N ,精英微粒数量为 N ,设最大迭代次数为 M .对于单个微粒,设每次按式(1)和(2)生成下代微粒的时间为 T_g ,计算适应度的时间为 T_f ,比较更新 X_i^{pb} 和 X^{gb} 的时间为 T_u ,精英微粒选择时间为 T_w ,模拟退火时间为 T_{sa} .则标准 PSO 的计算时间 T_1 约为 $NM(T_g + T_f + T_u)$.设精英微粒退火之后,新位置优于旧位置的有 $B\%$,则增加的计算时间 T_2 约为 $[T_w + N(T_{sa} + T_f + T_u \cdot B\%)]M$,其中 $B\% < 1$.总计算时间为 $T_1 + T_2$.EAPSO 与标准微粒群算法相比,增加的时间量为

$$\frac{T_2}{T_1} = \frac{[T_w + N(T_{sa} + T_f + T_u \cdot B\%)]M}{NM(T_g + T_f + T_u)} \times 100\% = \left[\frac{T_w}{N(T_g + T_f + T_u)} + \frac{T_{sa} + T_f + T_u \cdot B\%}{T_g + T_f + T_u} \right] \times 100\%.$$

其中： T_g, T_u, T_{sa} 都不大。 T_{se} 与问题的维数有关，维数大时 T_{se} 相对大些；维数小时 T_{se} 相对小些。而对于一般问题，维数大时 T_f 也会较大，因此在一般情况下，当 T_2 较小时， T_2/T_1 不会太大，即 EAPSO 相对于标准 PSO 算法增加的计算量并不大。

4 算法性能分析

为验证本文方法的有效性，首先应用 EAPSO 对文献[6]中测试函数 $F_1 \sim F_4$ ，以及文献[9]中测试函数 $f_6 \sim f_9$ 进行仿真分析。种群规模为 20， c_1, c_2 采用文献[12]的参数设置方案，即 $\omega = 0.729$,

表 1 标准测试函数优化结果比较

函数	算法	误差	收敛率	平均收敛代数	平均收敛时间/s
F_1	PSO	0.001	33/50	877.60	0.1455
	SA	0.001	22/50	798.54	0.1107
	SAPSO	0.001	50/50	395.00	0.1176
	EAPSO	0.001	47/50	127.63	0.0322
F_2	PSO	0.001	43/50	725.68	0.1157
	SA	0.001	24/50	199.90	0.0315
	SAPSO	0.001	50/50	342.00	0.0875
	EAPSO	0.001	47/50	191.57	0.0483
F_3	PSO	0.001	19/50	929.34	0.4240
	SA	0.001	28/50	2 000.00	0.7264
	SAPSO	0.001	39/50	501.00	0.3194
	EAPSO	0.001	40/50	213.64	0.1408
F_4	PSO	0.001	39/50	856.10	0.2621
	SA	0.001	50/50	853.60	0.1775
	SAPSO	0.001	50/50	408.00	0.2008
	EAPSO	0.001	50/50	115.19	0.0482
f_6	PSO	0.001	50/50	140.20	0.0495
	SA	0.001	44/50	109.56	0.0378
	SAPSO	0.001	50/50	83.22	0.0477
	EAPSO	0.001	50/50	48.16	0.0264
f_7	PSO	0.001	38/50	156.88	0.0745
	SA	0.001	24/50	137.54	0.0529
	SAPSO	0.001	45/50	75.42	0.0707
	EAPSO	0.001	46/50	39.48	0.0339
f_8	PSO	0.001	26/50	1 389.14	0.3763
	SA	0.001	16/50	1 106.49	0.2941
	SAPSO	0.001	38/50	835.91	0.3259
	EAPSO	0.001	36/50	461.85	0.1553
f_9	PSO	0.001	44/50	180.78	0.0746
	SA	0.001	21/50	130.66	0.0394
	SAPSO	0.001	50/50	112.49	0.0582
	EAPSO	0.001	50/50	60.72	0.0303

$c_1 = 1.49445, c_2 = 1.49445. V_{max,d}$ 设为微粒动态范围的 30%， $\omega = 0.1, \omega_{min} = 1, T_0 = 0.001$. PSO, SA, SAPSO 的参数设置见文献[6]. 采用 Matlab6.5 编程工具，计算机的配置为 P 2.40 GHz, 256 MB, Windows XP. 表 1 给出了利用 PSO, SA, SAPSO, EAPSO 各重复 50 次计算的收敛率、平均收敛代数及平均收敛时间。

由表 1 数据对比可以看出，在 8 个标准测试函数中，EAPSO 的收敛率均优于 PSO 和 SA，与 SAPSO 相当。可见 EAPSO 与 SAPSO 同样具有较好的全局搜索能力和较高的精度，有效避免了早熟现象。而 EAPSO 的平均收敛代数、平均收敛时间较 PSO 和 SAPSO 大为减少。可见 EAPSO 在增强 PSO 搜索精度的同时，能明显提高收敛速度。

这是由于在 EAPSO 中，除了模拟退火的 Metropolis 规则能以一定概率接受劣解，有利于算法跳出局部极值，提高算法的搜索精度之外，精英微粒也进行了特有的错位调整方式：以全局最优位置为基准，将较近的微粒向较远处调整，较远的微粒向较近处调整。这种运动方式使得精英微粒能够对全局最优位置附近的当前已知最优区域进行较为全面均衡地搜索。较近的微粒易陷入局部极值，将其向较远处调整，有利于其跳出局部极值，从而提高了算法的搜索精度。而较远的微粒向较近处调整，则易于发现较其目前个体最优更优的位置，从而提高其搜索速度；若发现较全局最优更优的位置，则能提高整个微粒群的搜索速度。另外， ω 仅为 0.1，根据 3.3 节的分析，算法计算复杂度的增加是有限的。

5 EAPSO 在 AVR 系统中的应用

同步发电机通常装有自动电压调节器 (AVR)。AVR 可连续调节励磁以保证发电机端电压在给定水平上，这是发电机电压控制的主要方式，其稳定性对于电力系统的安全性具有重要的作用。为提高 AVR 系统的性能，目前研究人员已做了大量的研究工作^[13-16]。

本文利用 EAPSO 对一实际发电机 AVR 系统进行 PID 参数优化。该 AVR 励磁系统由放大器、励磁机、发电机和传感器等 4 部分组成。AVR 系统框图及相关参数参见文献[14]。

性能评价函数 $W(K)$ ^[14] 定义为

$$\min_{K: \text{stabilizing}} W(K) = (1 - e^{-t_s}) (M_p + E_{ss}) + e^{-t_s} (t_s - t_r) \quad (12)$$

其中： M_p 为最大超调量， t_r 为上升时间， t_s 为调节时间， E_{ss} 为稳态偏差， K 为 $[k_p, k_i, k_d]$ ， ω 为权系数。以 $W(K)$ 作为算法的适应度函数。

输入为阶跃信号. k_p, k_i, k_d 的搜索空间分别为 $[0, 1.5], [0, 1], [0, 1]$. 以 K 为微粒的实数编码串, 采用实数编码 GA, PSO, IPSO^[17], EAPSO 优化 PID 控制器的 3 个参数 k_p, k_i, k_d , 并分别称为 GA-PID 控制器, PSO-PID 控制器, IPSO-PID 控制器和 EAPSO-PID 控制器. GA 的种群规模为 50, 交叉率为 0.6, 变异率为 0.01; IPSO 的独立群簇数为 4; PSO, IPSO, EAPSO 的微粒数为 50, 迭代次数为 150, 其他参数同第 4 节. 分别设置为 1.0, 1.5. 获得的 E_w 均为 0, 其余优化结果及适应度比较见表 2 和表 3. GA-PID, PSO-PID, EAPSO-PID 控制器阶跃响应的比较见图 4 和图 5.

由表 2 和表 3, 图 4 和图 5 可见, EAPSO-PID 控制器获得了较为满意的参数优化结果. 与其他 3 种控制器相比, AVR 系统的最大超调量、调整时间和上升时间都得到了改善. 说明本文 EAPSO 算法更为有效.

表 2 $\omega = 1.0$ 时的优化结果比较

参数	控 制 器			
	GA-PID	PSO-PID	IPSO-PID	EAPSO-PID
k_p	0.866 3	0.657 0	0.661 8	0.664 0
k_i	0.753 1	0.539 0	0.542 4	0.538 0
k_d	0.336 5	0.245 8	0.251 1	0.253 1
$M_p / \%$	7.340 0	1.160 0	1.144 5	1.072 4
t_s	0.851 9	0.402 5	0.397 4	0.395 3
t_r	0.195 9	0.276 7	0.267 6	0.266 1
$W(K)$	4.881 1	0.779 5	0.771 2	0.725 4

表 3 $\omega = 1.5$ 时的优化结果比较

参数	控 制 器			
	GA-PID	PSO-PID	IPSO-PID	EAPSO-PID
k_p	0.893 5	0.647 6	0.647 0	0.655 5
k_i	0.645 8	0.521 6	0.508 6	0.512 6
k_d	0.401 4	0.237 5	0.241 5	0.245 1
$M_p / \%$	7.480 0	0.910 0	0.871 0	0.826 0
t_s	0.982 2	0.416 8	0.415 7	0.407 7
t_r	0.175 0	0.286 1	0.277 8	0.273 5
$W(K)$	5.991 1	0.736 1	0.707 4	0.671 6

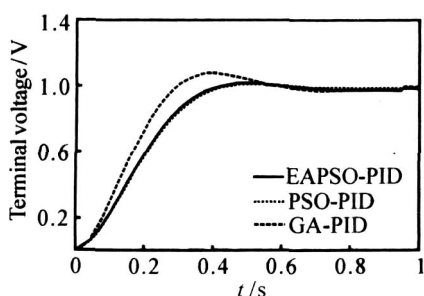


图 4 $\omega = 1.0$ 时的阶跃响应

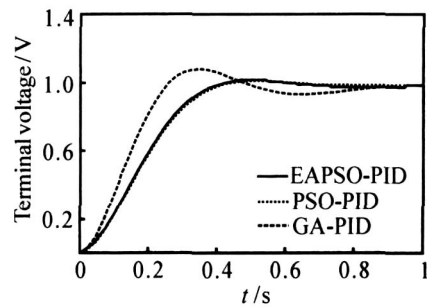


图 5 $\omega = 1.5$ 时的阶跃响应

6 结 论

本文结合 Logistic 方程的特点, 针对标准微粒群算法易陷入局部极值的缺点, 提出一种精英退火微粒群算法. 该方法仅对与全局最优微粒接近的部分微粒进行退火操作, 在提高算法搜索效果的同时, 控制了计算量的增加.

通过函数测试表明, 精英退火微粒群算法在增强算法搜索精度的同时, 能明显提高收敛速度. 将该方法应用于实际 AVR 系统的 PID 控制器参数整定, 获得了良好的控制效果.

参考文献(References)

- [1] Eberhart R C, Kennedy J. A new optimizer using particles swarm theory[C]. Proc of 6th Int Symposium on Micro Machine and Human Science. Nagoya, 1995: 39-43.
- [2] Shi Yu-hui, Eberhart R. A modified particle swarm optimizer[C]. Proc of IEEE Int Conf on Evolutionary Computation. Anchorage, 1998: 69-73.
- [3] 谢晓锋, 张文俊, 杨之廉. 微粒群算法综述[J]. 控制与决策, 2003, 18(2): 129-134.
(Xie X F, Zhang W J, Yang Z L. Overview of particle swarm optimization[J]. Control and Decision, 2003, 18(2): 129-134.)
- [4] 高鹰, 谢胜利. 基于模拟退火的粒子群优化算法[J]. 计算机工程与应用, 2004, 40(1): 47-50.
(Gao Y, Xie S L. Particle swarm optimization algorithms based on simulated annealing[J]. Computer Engineering and Applications, 2004, 40(1): 47-50.)
- [5] Wang Xi-huai, Li Jun-jun. Hybrid particle swarm optimization with simulated annealing[C]. The 2004 Int Conf on Machine Learning and Cybernetics. Shanghai, 2004: 2402-2405.
- [6] 王丽芳, 曾建潮. 基于微粒群算法与模拟退火算法的协同进化方法[J]. 自动化学报, 2006, 32(4): 630-635.
(Wang L F, Zeng J C. A Cooperative evolutionary algorithm based on particle swarm optimization and simulated annealing algorithm [J]. Acta Automatica Sinica, 2006, 32(4): 630-635.)
- [7] 李亚东, 李少远. 一种新的遗传混沌优化组合方法[J].

- 控制理论与应用, 2002, 19(1): 143-145.
(Li Y D, Li S Y. A new genetic chaos optimization combination method [J]. Control Theory and Applications, 2002, 19(1): 143-145.)
- [8] 王焱, 刘景录, 孙一康. 基于变尺度混沌优化策略的混合遗传算法[J]. 控制与决策, 2002, 17(6): 958-960.
(Wang Y, Liu J L, Sun Y K. Hybrid genetic algorithm based on mutative scale chaos optimization strategy[J]. Control and Decision, 2002, 17(6): 958-960.)
- [9] 杜海峰, 公茂果, 刘若辰, 等. 自适应混沌克隆进化规划算法[J]. 中国科学 E: 信息科学, 2005, 35(8): 817-829.
(Du H F, Gong M G, Liu R C, et al. Adaptive parallel chaos immune evolutionary programming[J]. Science in China Series E: Information Sciences, 2005: 35(8): 817-829.)
- [10] 雷德明, 严新平, 吴智铭. 多目标混沌进化算法[J]. 电子学报, 2006, 34(6): 1142-1145.
(Lei D M, Yan X P, Wu Z M. Multi-objective chaotic evolutionary algorithm [J]. Acta Electronica Sinica, 2006, 34(6): 1142-1145.)
- [11] 唐巍, 李殿璞, 陈学允. 混沌理论及其应用研究[J]. 电力系统自动化, 2000, 24(7): 67-70.
(Tang W, Li D P, Chen X Y. Chaos theory and research on its applications[J]. Automation of Electric Power Systems, 2000, 24(7): 67-70.)
- [12] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization[C]. Proc of the IEEE Conf on Evolutionary Computation. California: IEEE Service Center, 2000: 84-88.
- [13] 胡凌燕, 辛勇, 项安. 基于变参数控制的谐波励磁发电机电压调节器[J]. 电力系统自动化, 2003, 27(22): 89-91.
(Hu L Y, Xin Y, Xiang A. Automatic voltage regulator of harmonic excited generator based on variable parameter PID control [J]. Automation of Electric Power Systems, 2003, 27(22): 89-91.)
- [14] Zwe-lee Gaing. A particle swarm optimization approach for optimum design of PID controller in AVR system [J]. IEEE Trans on Energy Conversion, 2004, 19(2): 384-391.
- [15] Dong Hwa Kim, Jae Hoon Cho. Intelligent control of AVR system using GA-BF [C]. Lecture Notes in Computer Science Proc of Springer. Melbourne, 2005: 854-859.
- [16] Dong Hwa Kim, Jin Ill Park. Intelligent PID controller tuning of AVR system using GA and PSO [C]. Lecture Notes in Computer Science Proc of Springer. Hefei, 2005: 366-375.
- [17] 房启超, 徐林, 王建辉, 等. 改进的 PSO 及其在结晶器液位控制中的应用[J]. 仪器仪表学报, 2006, 27(11): 1399-1402.
(Fang Q C, Xu L, Wang J H, et al. Improved PSO and its application in crystallizer liquid level control system[J]. Chinese J of Scientific Instrument, 2006, 27(11): 1399-1402.)

(上接第 755 页)

- [12] Zhou K, Low K, Wang D, et al. Zero-phase odd-harmonic repetitive controller for a single-phase PWM inverter[J]. IEEE Trans on Power Electronics, 2006, 21(1): 193-202.
- [13] Shi J, Gao F, Wu T J. Robust design of integrated feedback and iterative learning control of a batch process based on a 2D Roesser system[J]. J of Process Control, 2005, 15(1): 907-924.
- [14] Xie L H, Du C L. H_∞ control and filter of two-dimensional system [M]. Berlin: Springer, 2002: 5-25.
- [15] Paszke W, Krzysztof G, Eric R. Guaranteed cost control of uncertain differential linear repetitive processes[J]. IEEE Trans on Circuits and Systems-II: Express Briefs, 2004, 51(11): 629-634.
- [16] Khargonek P P, Petersen I R, Zhou K. Robust stabilization of uncertain linear systems: Quadratic stabilizability and H_∞ control theory[J]. IEEE Trans on Automatic Control, 1990, 35(3): 356-361.
- [17] Balas G, Chiang R, Packard A, et al. Robust control toolbox user's guide [M]. Natick: The Math Works, 2005: 134-157.