

文章编号: 1001-0920(2012)04-0557-05

基于 SQP 局部搜索的混沌粒子群优化算法

徐文星¹, 耿志强¹, 朱群雄¹, 顾祥柏^{1,2}

(1. 北京化工大学 信息科学与技术学院, 北京 100029; 2. 中国石化集团 炼化工程公司, 北京 100029)

摘要: 提出一种基于序贯二次规划(SQP)法的混沌粒子群优化方法(CPSO-SQP). 将混沌 PSO 作为全局搜索器, 并用 SQP 加速局部搜索, 使得粒子能够在快速局部寻优的基础上对整个空间进行搜索, 既保证了算法的收敛性, 又大大增加了获得全局最优的几率. 仿真结果表明, 算法精度高、成功率高、全局收敛速度快, 明显优于现有算法. 将所提出的算法用于高密度聚乙烯(HDPE)装置串级反应过程的乙烯单耗优化, 根据工业反应机理以及现场操作经验分析可知, 所提出的算法是可行的.

关键词: 粒子群优化; 序贯二次规划; 混沌映射; 非线性约束优化

中图分类号: TP18; TQ015.9; TP391.9

文献标识码: A

Chaos particle swarm optimization algorithm integrated with sequential quadratic programming local search

XU Wen-xing¹, GENG Zhi-qiang¹, ZHU Qun-xiong¹, GU Xiang-bai^{1,2}

(1. School of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China; 2. Sinopec Engineering, Beijing 100029, China. Correspondent: ZHU Qun-xiong, E-mail: zhuqx@mail.buct.edu.cn)

Abstract: A sequential quadratic programming integrated particle swarm optimization algorithm (CPSO-SQP) is proposed. This new algorithm uses CPSO, which makes the best of ergodicity of chaos mapping, as the global optimizer while the SQP is employed for accelerating the local search. Thus, the particles are able to search the whole space while finding local optima fast, which increases the possibility of exploring a global optimum in problems with more local optima while ensuring the convergence of algorithm. The simulation results for benchmark functions show that CPSO-SQP has better accuracy, more probability of finding global optimum and faster speed of convergence than those reported in the literature. The feasibility of the method is illustrated with the challenging ethylene piece yardage optimization problem of a cascade HDPE reaction course.

Key words: particle swarm optimization; sequential quadratic programming; chaos mapping; constrained nonlinear programming

1 引言

粒子群优化算法(PSO)是一种受鸟类觅食行为研究的启发而提出的智能群体搜索算法^[1]. 算法无需领域知识, 实现简单、收敛速度快, 而且在全局和局部搜索能力上具有很大的灵活性, 并能在二者之间保持一种平衡, 已经成为目前随机搜索策略的研究热点之一. 但是, 从文献的调研中可以看出, 现行的 PSO 算法存在 2 个主要的搜索性能问题: 1) 在搜索开始阶段收敛得较快, 但在趋近于极值点时的局部搜索调整却比较慢, 而且不能保证收敛到局部最优点; 2) 种群多样性损失过快, 导致算法早熟. 针对问题 1), 文献 [2-5] 尝

试直接将爬山法, Nelder-Mead 单纯形法以及序贯二次规划(SQP)法等作为局部搜索算法与 PSO 混合, 或利用梯度信息动态改变惯性权重, 提高了收敛速度, 但对高维多极值问题很难获得全局最优. 针对问题 2), 文献 [6-12] 利用混沌、扰动、多种群竞争等策略保持种群多样性, 搜索到全局最优的几率明显增大, 但所需迭代次数较多, 且精度不高.

本文提出将 SQP 作为局部搜索策略, 以加速 PSO 算法的局部收敛, 并运用混沌映射^[13]保持种群多样性, 以加强 PSO 算法的全局搜索能力, 同时解决了上述两大问题.

收稿日期: 2010-05-18; 修回日期: 2010-09-12.

基金项目: 国家 863 计划项目(2006AA04Z184); 中央高校科研业务费项目(ZZ1136).

作者简介: 徐文星(1982-), 女, 博士生, 从事工业过程优化与控制的研究; 朱群雄(1960-), 男, 教授, 博士生导师, 从事过程建模与优化控制的研究.

2 基于SQP局部搜索的混沌粒子群优化算法(CPSO-SQP)

2.1 标准PSO

PSO最早由Kennedy和Eberhart^[1]于1995年提出. 在PSO中, 每个解都是一个“粒子”. 首先初始化一群随机粒子; 然后通过跟踪个体极值和全局极值, 按下式更新其速度和位置:

$$v_{id}^{k+1} = \omega \cdot v_{id}^k + c_1 \cdot \text{rand}() \cdot (P_{id} - x_{id}^k) + c_2 \cdot \text{rand}() \cdot (P_{gd} - x_{id}^k), \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}, \quad (2)$$

直至找到最优解.

2.2 基于SQP的局部搜索

SQP方法最早由Wilson^[14]于1963年在其博士论文中首次提出, 特别适用于非线性约束规划问题. 其算法思想^[15]是将原问题的Lagrange方程进行二次近似, 形成QP子问题, 即

$$\begin{aligned} \min_{d \in \mathbf{R}^n} & \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d, \\ \text{s.t.} & \nabla g(x_k)^T d + g(x_k) \leq 0. \end{aligned} \quad (3)$$

其中: 下标 k 是当前迭代次数; H 是Hessian矩阵, 可由拟牛顿法, 如BFGS法, 近似求出. QP子问题的解作为下一步迭代的线性搜索方向.

2.3 混沌映射

因其遍历性及随机性, 混沌策略已多次被融合进粒子群算法, 帮助粒子逃离局部最优^[6-10]. 本文尝试在粒子群进化收敛, 即全局最优值的变化小于某个给定的值 ε 时, 将种群按给定概率 P_m 进行混沌映射. 具体方法为, 对于所有非全局最优粒子配置分布于 $[0, 1]$ 之间的随机数 r , 如果 $r \leq P_m$, 则对该粒子中的每一维变量进行Logistic混沌映射^[13].

$$x_{k+1} = \mu x_k (1 - x_k), \quad 0 \leq x_0 \leq 1, \mu = 4. \quad (4)$$

记录该粒子当前最优解, 然后进入新一轮的寻优搜索. 混沌映射子程序(CMS)的步骤如下:

Step 1: 初始化 N 维变量 X 中需要进行混沌映射的变量下标 i .

Step 2: 对于变量 x_i , 进行如下操作:

Step 2.1: 按下式将其映射为 $[0, 1]$ 之间的混沌变量 cx_i :

$$cx_i = \frac{x_i - x_{i \min}}{x_{i \max} - x_{i \min}}, \quad (5)$$

其中 $x_{i \max}$ 和 $x_{i \min}$ 分别为第 i 维变量的搜索上下界.

Step 2.2: 按式(4)更新混沌变量 cx_i .

Step 2.3: 将混沌变量 cx_i 按下式转换成决策变量 x_i :

$$x_i = x_{i \min} + cx_i \cdot (x_{i \max} - x_{i \min}). \quad (6)$$

2.4 CPSO-SQP算法框架

CPSO-SQP算法的思想是: 将混沌PSO算法作为全局收敛器, 而SQP算法作为局部搜索器, 算法根据收敛情况自动在全局与局部搜索之间切换. 具体过程可归结为: 在PSO算法收敛, 即当前全局最优值的变化小于某个给定值 ε 时, 以当前全局最优位置为初始点, 调用SQP进行局部搜索. 此时, 为了避免早熟, 需按一定概率进行混沌映射, 促使粒子逃离局部极值范围, 再次进入PSO寻优过程, 如此迭代直至终止. 综合考虑PSO和SQP算法的复杂度, 为了便于公平地与其他算法进行比较, 文中使用函数评价次数来反映算法的效率. 算法步骤如下:

Step 1: 初始化.

Step 1.1: 在各变量取值范围内随机初始化粒子群. 设定群体规模为 M , 最大函数评价次数为 $\max\text{FEs}$, 期望最优值为 VTR , 有效函数评价次数为 FES , 混沌变换概率 P_m .

Step 1.2: 在搜索范围内均匀随机初始化所有粒子的位置为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})^T$, 速度为 $V_i = (v_{i1}, v_{i2}, \dots, v_{iN})^T$, 向量长度 N 为问题维数.

Step 1.3: 计算 $f(X_i)$. 令种群中具有最佳函数值的微粒为初始CPSO搜索的全局极值位置 P_g 以及整个算法的全局极值位置 P_{best} , 各微粒本身位置为其初始个体极值位置 P_i , 累积函数评价次数 $k = M$.

Step 2: 如果 $k \geq \max\text{FEs}$, 则保存优化结果 $\min\{f(P_g), f(P_l)\}$, 结束; 否则, 执行以下步骤:

Step 2.1: 按式(1)和(2)更新粒子的速度 V_i 和位置 X_i , 并判断 $f(P_g)$ 是否第1次小于 VTR . 如果是, 则 $\text{FES} = k$;

Step 2.2: 根据群体中各微粒的信息, 更新 P_i 和 P_g , 并记录全局最优的粒子下标 g_{best} .

Step 3: 判断PSO是否收敛. 如果 $P_g^{k-M} - P_g^k \leq \varepsilon$, 则继续; 否则, 返回Step 2.

Step 4: 以全局最优粒子的位置 $X_{g_{\text{best}}}$ 为初始点, 运行SQP算法, 更新 $X_{l_{\text{best}}}$, P_l 和 k .

Step 5: 如果 $k \geq \max\text{FEs}$, 则保存优化结果 $\min\{f(P_g), f(P_l)\}$, 结束; 否则, 继续.

Step 6: 对每个粒子, 生成 $[0, 1]$ 之间的随机数 r , 如果 $r \leq P_m$ 且下标 $i \neq g_{\text{best}}$, 则在混沌搜索空间运行CMS进行混沌映射, 并计算其新位置所对应的目标值 f_i^{k+1} , 更新 k, P_g , 判断是否第1次小于 VTR . 如果是, 则 $\text{FES} = k$.

Step 7: 返回Step 2.

3 实验及结果讨论

3.1 实验设置

本文选取表1所示的5个全局最优值为0的最小化函数问题, 维数 $N=10, 20, 30$; 群体规模 $M=30$; 惯性因子 w 采用文献[6]中的线性惯性权重 (简称LDW), $w_{\max}=0.9, w_{\min}=0.4, c_1=2, c_2=2$, 速度上下界 $v_{\max}=(x_{\max}-x_{\min})/2, v_{\min}=-v_{\max}$, 收敛阈值 $\varepsilon=0.01$, 混沌变换概率 $P_m=0.2$, 映射维数为 N , 在迭代过程中混沌搜索区间为 $[(1-\eta) \cdot P_g, (1+\eta) \cdot P_g], \eta=1.5$; 对Rosenbrock函数设置最大函数评价次数maxFES为60000次, 其他函数为4500次. 对每个测试函数寻优50次, 并将小于1.00E-100的值记为0.00E+00.

表1 用于测试的Benchmark函数

name	function	range
Sphere	$f_1 = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$
Rosenbrock	$f_2 = \sum_{i=1}^{n-1} [100(x_{i+1}-x_i^2)^2 + (x_i-1)^2]$	$[-30, 30]^n$
Rastrigin	$f_3 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$
Griewank	$f_4 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$
Ackley	$f_5 = -20 \exp\left(-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right] + 20 + e$	$[-32, 32]^n$

3.2 仿真结果及分析

1) 算法寻优精度测试. 按照3.1节中的设置, 将本文提出的CPSO-SQP算法与相同环境下的SQP和CPSO^[6]算法进行比较. 从表2中的比较结果可以看出, 对于所测试的所有Benchmark函数, 由CPSO-SQP算法优化得到的平均、最优、最差结果都是最小的, 这表明CPSO-SQP算法寻优的精确度优于其他3种算法.

2) 算法收敛速度测试. 在算法收敛速度测试中, Rosenbrock的维数Dim=10, 优化目标为1.00E-2, 其他函数的维数Dim=30, 优化目标为1.00E-5. 记录CPSO-SQP算法每次运行时达到优化目标所需要的函数计算次数, AvgFES表示算法达到优化目标的平均函数计算次数, BestFES表示最少函数计算次数, SR表示达到优化目标的寻优次数占实验次数的比例, 将其结果与相同环境下的SQP, 权重线性下降PSO(LDWPSO)^[6]和CPSO^[6]算法测试结果进行比较, 结果如表3所示.

因为SQP算法达到收敛即停止迭代, 不会继续运行至最大函数评价次数, 因此对SQP算法考察AvgFES和BestFES没有意义. 对于所测的5个函数, CPSO-SQP算法的成功率都达到了100%, CPSO算法对Rosenbrock函数的寻优成功率只有60%, LDWPSO算法对多个函数的寻优成功率都不超过60%, 对

表2 CPSO-SQP和CPSO, SQP的收敛精度性能测试对比

Fun	Dim	CPSO-SQP			CPSO ^[6]			SQP		
		MEAN	BEST	WORST	MEAN	BEST	WORST	MEAN	BEST	WORST
Sphere	10	0.00E+00	0.00E+00	0.00E+00	3.42E-12	1.44E-81	1.71E-10	2.57E-27	3.57E-28	8.82E-27
	20	0.00E+00	0.00E+00	0.00E+00	7.59E-06	1.63E-78	1.52E-04	7.82E-27	1.80E-27	1.69E-26
	30	0.00E+00	0.00E+00	0.00E+00	3.92E-05	9.36E-73	1.32E-02	1.66E-26	6.16E-27	2.89E-26
Rosenbrock	10	3.55E-07	1.57E-08	2.59E-06	9.39E-03	1.19E-08	9.01E-02	1.44E+00	7.56E-12	3.99E+00
	20	2.38E-08	3.55E-07	2.03E-06	2.28E-02	2.44E-06	3.63E-01	9.57E-01	5.63E-11	3.99E+00
	30	6.31E-07	2.49E-08	3.13E-06	4.82E-02	1.35E-05	9.28E-01	1.28E+00	1.56E-11	3.99E+00
Rastrigin	10	0.00E+00	0.00E+00	0.00E+00	1.80E-07	0.00E+00	7.80E-06	6.92E+01	2.39E+01	1.54E+02
	20	0.00E+00	0.00E+00	0.00E+00	7.44E-05	0.00E+00	2.01E-03	1.35E+02	3.98E+01	2.55E+02
	30	0.00E+00	0.00E+00	0.00E+00	2.52E-03	0.00E+00	1.26E-01	1.97E+02	9.35E+01	3.38E+02
Griewank	10	0.00E+00	0.00E+00	0.00E+00	2.13E-10	0.00E+00	6.42E-09	3.54E-01	2.89E-09	3.63E+00
	20	0.00E+00	0.00E+00	0.00E+00	2.65E-07	0.00E+00	1.32E-05	1.48E-04	4.38E-12	7.40E-03
	30	0.00E+00	0.00E+00	0.00E+00	2.24E-04	0.00E+00	1.12E-02	5.39E-09	1.01E-13	2.15E-08
Ackly	10	2.13E-16	0.00E+00	3.55E-15	1.60E-08	8.82E-16	6.33E-07	1.91E+01	1.52E+01	2.00E+01
	20	1.63E-15	0.00E+00	3.55E-15	3.27E-06	8.82E-16	1.39E-04	1.94E+01	1.67E+01	2.00E+01
	30	2.77E-15	0.00E+00	3.55E-15	1.24E-04	8.82E-16	5.89E-03	1.94E+01	1.44E+01	2.17E+01

表3 CPSO-SQP和CPSO, LDWPSO, SQP的收敛速度性能测试对比

Fun	Dim	CPSO-SQP			CPSO ^[6]			LDWPSO ^[6]			SQP
		AvgFES	BestFES	SR	AvgFES	BestFES	SR	AvgFES	BestFES	SR	SR
Sphere	30	1195.3	737	50/50	1696	990	50/50	34128	3153	50/50	50/50
Rosenbrock	10	2971.2	421	50/50	41544	3750	31/50			0/50	32/50
Rastrigin	30	1220.6	573	50/50	2480	1170	50/50			0/50	0/50
Griewank	30	1188.3	174	50/50	2154	1020	50/50	34161	31920	23/50	50/50
Ackly	30	1952.2	1314	50/50	2201	1320	50/50	42546	38130	49/50	0/50

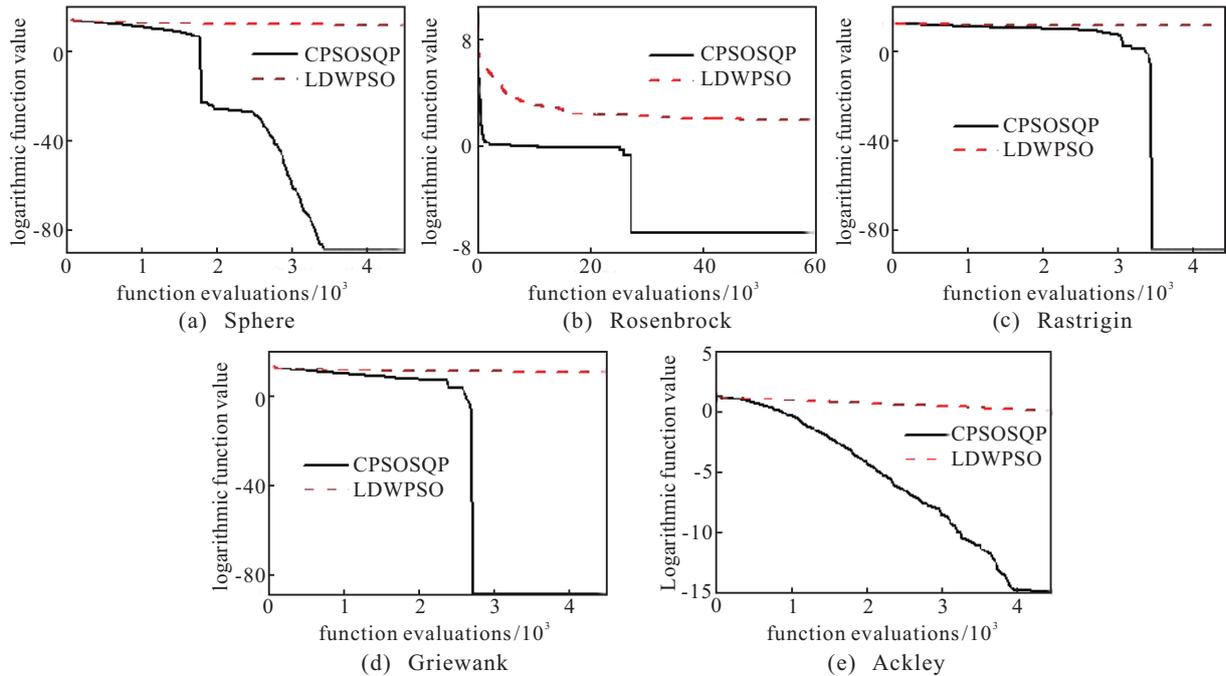


图 1 CPSOSQP 与 LDWPSO 在测试函数上的比较

Rastrigin 函数甚至无法寻找到最优解, SQP 算法则对于不同问题其成功率差异明显. CPSO-SQP 算法的 AvgFES 和 BestFES 值明显低于 LDWPSO 和 CPSO 算法, 可见使用 CPSO-SQP 算法进行优化所需的平均相对时间较少, 这说明其算法效率较高. 总之, CPSO-SQP 算法不论成功率还是执行效率都是 3 种算法中最好的, 鲁棒性较好, 更能满足优化效果稳定性的需求.

图 1 为 CPSO-SQP 和 LDWPSO 对 5 个标准测试函数进行优化, 对数平均适应度值随函数评价次数变化情况的比较. 图中: 虚线为 LDWPSO 法, 直线为本文方法结果. 不难发现, 迭代开始阶段, CPSO-SQP 算法的收敛速度大于 LDWPSO 算法, 而且经过一定的迭代次数之后, LDWPSO 算法基本收敛, 陷于局部极小值, 这时 CPSO-SQP 算法仍能继续寻优过程, 最后寻优精度更是远高于 LDWPSO 算法. 主要原因是 CPSO-SQP 算法中混沌映射可以帮助 PSO 不断地跳出局部最优, 而每当找到更优的全局最优时, SQP 算法便会以最优粒子所在位置为初始点进行局部搜索. 由此可见, 所提出的 CPSO-SQP 算法确实加快了局部收敛速度, 并能有效地逃离局部最优, 增加了获得全局最优的可能性, 全局搜索的性能和效果都很好.

4 应用实例

使用 RBF 降耗模型^[16]作为适应度函数, 将 CPSO-SQP 算法应用于某企业高密度聚乙烯装置串联反应过程总单体单耗的优化.

1) 目标函数. 设实际单耗为 $f(x)$, 则目标函数为 $\min_x f(x)$, 其中 $x = \{x_1, x_2, \dots, x_n\}$ 为输入向量. 各输入量分别为: 一反应器温度 x_1 , 一反应器压力 x_2 , 二反应器温度 x_3 , 二反应器压力 x_4 , 闪蒸罐压力 x_5 , 一反催化剂进料量 x_6 , 二反催化剂进料量 x_7 , 一反乙烯分压 x_8 , 一反氢气乙烯分压比 x_9 , 二反乙烯分压 x_{10} , 二反氢气乙烯分压比 x_{11} .

2) 约束条件. $84 \leq x_1 \leq 85, x_2 = 0.85, 76 \leq x_3 \leq 78, x_4 = 0.22, 0.22 \leq x_5 \leq 0.3, 60 \leq x_6 \leq 80, 45 \leq x_7 \leq 55, x_8 = 0.85, x_9 = 5.7, x_{10} = 0.95, x_{11} = 0.09$.

3) 优化结果. 对比优化前后的操作条件, 企业乙烯生产的总单体单耗如表 4 所示. 由表 4 数据可以看出, 使用本文提出的 CPSO-SQP 算法, 乙烯单耗比优化前降低了 1.06%, 比使用 PSO 算法优化的操作降低了 0.41%, 显然本文提出的 CPSO-SQP 算法优化效果明显优于 PSO 算法. 仿真实验结果与现场操作结果一致, 而且符合工业反应机理. 算法能够在一定程度上指导生产操作, 降低乙烯单耗, 节约生产成本, 应用效果良好.

表 4 乙烯单耗优化操作对比(单耗单位: 吨/吨聚乙烯)

	变 量	优化前	PSO 优化后	CPSO-SQP 优化后
操作变量	一反应器温度/℃	84.46	84.93	84.60
	二反应器温度/℃	77.09	77.98	77.70
	闪蒸罐压力/Mpa	0.27	0.25	0.23
	一反催化剂流量/(kg/h)	70.14	64.09	60
	二反助催化剂流量/(kg/h)	47.02	51.49	47.93
输出	乙烯单耗/(吨/吨)	1.0214	1.0148	1.0106

5 结 论

本文在利用SQP加快PSO的局部搜索速度的同时,应用混沌映射避免早熟,提出了CPSO-SQP算法,使得粒子能够在快速局部寻优的基础上对整个空间进行搜索.对高维Benchmark函数的仿真结果表明,所提出的算法收敛速度快、精度高,且全局寻优能力强.应用CPSO-SQP算法对某企业高密度聚乙烯装置串级反应过程的乙烯单耗进行优化,根据工业反应机理以及现场操作经验分析可知,本文算法是可行的.

参考文献(References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]. IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.
- [2] Zhao S Z, Liang J J, Suganthan P N. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization[C]. Proc of 2008 IEEE Congress on Evolutionary Computation. Hong Kong: IEEE Press, 2008: 3845-3852.
- [3] Fan S S, Zahara E. A hybrid simplex search and particle swarm optimization for unconstrained optimization[J]. European J of Operational Research, 2007, 181(2): 527-548.
- [4] Chang Y P. Integration of SQP and PSO for optimal planning of harmonic filters[J]. Expert Systems with Applications, 2010, 37(3): 2522-2530.
- [5] 肖健梅, 李军军, 王锡淮. 梯度微粒群优化算法及其收敛性分析[J]. 控制与决策, 2009, 24(4): 560-564.
(Xiao J M, Li J J, Wang X H. Convergence analysis of particle swarm optimization and its improved algorithm based on gradient[J]. Control and Decision, 2009, 24(4): 560-564.)
- [6] 刘玲, 钟伟民, 钱锋. 改进的混沌粒子群优化算法[J]. 华东理工大学学报: 自然科学版, 2010, 36(2): 267-272.
(Liu L, Zhong W M, Qian F. An improved chaos-particle swarm optimization algorithm[J]. J of East China University of Science and Technology: Natural Science Edition, 2010, 36(2): 267-272.)
- [7] He Y Y, Zhou J Z, Xiang X Q, et al. Comparison of different chaotic maps in particle swarm optimization algorithm for long-term cascaded hydroelectric system scheduling[J]. Chaos, Solitons and Fractals, 2009, 42(5): 510-518.
- [8] Xiang T, Liao X F, Wong K W. An improved particle swarm optimization algorithm combined with piecewise linear chaotic map[J]. Applied Mathematics and Computation, 2007, 190(2): 1637-1645.
- [9] 孙艳霞, 王增会, 陈增强, 等. 混沌粒子群优化及其分析[J]. 系统仿真学报, 2008, 20(21): 5920-5923.
(Sun Y X, Wang Z H, Chen Z Q, et al. Chaotic particle swarm optimization and analysis[J]. J of System Simulation, 2008, 20(21): 5920-5923.)
- [10] 高雷阜, 刘旭旺. 一种基于混沌的自适应粒子群全局优化方法[J]. 计算机工程与应用, 2010, 26(3): 51-53.
(Gao L F, Liu X W. Adaptive particle swarm global optimization algorithm based on chaos[J]. Computer Engineering and Applications, 2010, 26(3): 51-53.)
- [11] Zhao X C. A perturbed particle swarm algorithm for numerical optimization[J]. Applied Soft Computing, 2010, 10(1): 119 - 124.
- [12] 耿志强, 朱群雄, 顾祥柏, 等. 基于多群竞争 PSO-RBFNN 的乙烯裂解深度智能优化控制[J]. 化工学报, 2010, 61(8): 1942-1948.
(Geng Z Q, Zhu Q X, Gu X B, et al. Optimal control of cracking depth based on multi-swarm competitive PSO-RBFNN for ethylene cracking furnace[J]. CIESC J, 2010, 61(8): 1942-1948.)
- [13] May R. Simple mathematical models with very complicated dynamics[J]. Nature, 1976, 261(5560): 459-467.
- [14] Wilson R B. A simplicial algorithm for concave programming[D]. Boston: Graduate School of Business Administration, Harvard University, 1963.
- [15] Fletcher R. Practical methods of optimization[M]. New York: Wiley, 1987: 304-316.
- [16] 吴丽芳, 徐圆, 朱群雄. 基于组件的石化过程智能建模与优化系统的设计与开发[J]. 计算机与应用化学, 2009, 26(8): 979-984.
(Wu L F, Xu Y, Zhu Q X. Design and development of intelligent modeling and optimization for petrochemical process based on component[J]. Computers and Applied Chemistry, 2009, 26(8): 979-984.)