面向应用的快速多峰寻优算法

吴 江, 胡捍英, 吴 瑛
(解放军信息工程大学, 郑州 450002)

摘 要: 工程应用中的多峰寻优问题要求搜索目标函数的多个极值点,现有的多峰优化方法难以直接利用应用问题的先验知识引导算法过程,多峰寻优效率较低。基于粒子群优化算法设计一种面向应用的多峰寻优算法,能有效利用易于获得的先验参数,如峰间分辨率、峰位置精度、峰值个数等实现快速多峰搜索。该算法保持了粒子群算法的简单性并改善了搜索多样性,使其可控地收敛到多个峰值上。将该算法与几种典型的多峰寻优方法进行了对比测试和分析,结果表明,对复杂多峰函数,该算法能以最快的收敛速度实现多峰搜索。

关键词: 面向应用; 多峰寻优; 粒子群优化算法
中图分类号: TP399 **文献标志码**: A **文章编号**: 1001-3695(2008)12-3617-04

Application-oriented fast optimizer for multi-peak searching

WU Jiang, HU Han-ying, WU Ying
(PLA Information Engineering University, Zhengzhou 450002, China)

Abstract: Multi-peak optimizations in engineering applications are needed to search multiple extrema of objective functions. Previous multi-peak searching methods usually cannot make use of prior parameters to guide the algorithm directly. This paper proposed an application-oriented multi-peak optimizer based on the particle swarm optimization(PSO). It took advantage of prior parameters such as peak resolution, solution accuracy, and peak number demanded, which could usually be ascertained in real-world problems. The algorithm kept the simplicity of basic PSO and expanded its searching diversity. This paper compared the new algorithm with some typical multimodal optimization algorithms on the basis of which the tests and analyses of them were conducted. Results show that the new algorithm can successfully locate the multiple extrema that outhors need at highest speed.

Key words: application-oriented; multi-peak searching; PSO

0 引言

在许多工程应用问题中,常需要寻找多个较优解,如工程设计中获得次优的备用方案、军事应用中多目标探测等。这些问题可以归结为非线性函数多峰寻优问题,即要求优化算法在搜索全局最优解的同时输出前 N 个局部最优解。现有优化方法大多讨论如何获取全局最优解,而对多局部极值的寻优问题研究较少[1]。受实际需求推动,近年来求多个局部最优解的算法引起了较大的研究兴趣,这种算法通常称为多峰搜索(multi-peak searching)或多模式优化(multimodal optimization)。目前实现多峰寻优的方法主要有三类: a)使用大量随机初始化解的局部优化算法,这种方法的优点是通用性较强,但其局部优化算法往往对目标函数的光滑性或连续性有要求,而且在目标函数峰值数量较多时运算量很大; b)利用遗传算法中小生境(niche)的思想创建子种群(subpopulation or sub-swarm)保持搜索的多样性[1,2],其优点是在峰值情况不可预测时能自动根据目标函数划分种群,但该方法缺乏全局信息引导,计算量分散故而对需要的较优峰收敛较慢; c)结合序列生境技术(sequential niche technique, SNT)[3]的全局优化算法,即重复进行多次全局寻优并使用下降函数(derating function)修改适应值函数,这种方法可以系统地遍历峰值,但其固有的串行性限制了寻优的效率。

在实际的应用问题中,往往容易确定目标函数的某些先验参数,研究如何充分利用这些参数提高优化算法的效率是很有现实意义的。本文利用粒子群优化算法(particle swarm optimization, PSO)形式简单、全局收敛速度快的特点,将其全局极值点($g$best)改为可由工程参数控制的多个 $l$best(local best),各粒子以最近的 $l$best 作为社会经验更新速度,以较少的计算量实现可控的多峰搜索,得到一种改进的 PSO 多峰寻优算法。文中将该算法与三种具有代表性的多峰优化算法进行了对比分析,验证了算法的优势。

1 问题描述

工程优化问题中根据应用目的,往往可事先明确对解的某些要求,由此确定的参数在优化算法设计中应充分利用。首先,易于确定的是峰间分辨率 $\Delta_{dist}=(\delta_1,\cdots,\delta_D)$。其中: δ_d $(d=1,\cdots,D)$ 是第 d 维上的峰间分辨率; D 是解空间的维数; Δ_{dist} 表示系统可区分的两个峰间的最小距离,它往往取决于实际系统的目标分辨能力。其次,可由系统参数测量精度和应用需求确定问题的求解精度,在本文中称之为峰位置精度,并记为 A_{peak},它以 D 维上的精度值为分量构成,代表了用户允许的近似解偏离真实解的最大误差。另外,系统用户对峰值个数的

收稿日期: 2007-12-26; 修回日期: 2008-05-05

作者简介: 吴江(1976-),男,河南泌阳人,博士,主要研究方向为信号处理、人工智能(wjs_mail@126.com); 胡捍英(1961-),男,教授,博士,主要研究方向为通信信号处理、移动通信; 吴瑛(1960-),女,教授,硕士,主要研究方向为阵列信号处理.

需求总是有限的,即可指定较优峰值的个数,这要求算法找到前 N_{peak} 个局部最优解。

按照前面的讨论,多峰寻优问题可描述为寻找可行解 X_i, (i = 1, 2, ⋯, N_{peak}) , 满足

$$\begin{cases} X_i \in \{ X \mid f(X) \geq f(X') ,\ |X' - X| \leq \Delta_{peak} \} \\ |X_i - X_i^*| \leq A_{peak} \\ f(X_1) \geq f(X_2) \geq \cdots \geq f(X_{N_{peak}}) \geq \cdots \end{cases} \quad (1)$$

其中: X、X'、X_i、$X_i^* \in \Omega$ 是 D 维决策变量; $\Omega \subset R^D$ 为可行域; X_i^* 表示第 i 个峰处的精确解或真实解; f 为目标函数; $\Delta_{peak} = \Delta_{dist} / 2$ 定义为峰分辨半径。对于常用的边界约束问题,设 Ω由搜索空间的边界 B 所围成, B 是 $D \times 2$ 的矩阵,由每一维的区间上、下界构成。对于目标函数求极小峰值问题,可转换为对 [- $f(X)$] 求极大峰值问题。

2 多峰寻优 PSO 算法

2.1 基本 PSO 及其改进型

1995 年 Kennedy 等人[4] 提出粒子群优化算法。在 PSO 中,用种群表示问题潜在解的集合,集合中的每个粒子表示一个可行解。初始种群通常是随机产生,然后在解空间中运动(演化)。在种群演化过程中,所有粒子间进行全局通信,每个粒子能共享其他粒子当前的搜索结果。记每个粒子经历的最好位置为 $p$best,整个种群经历的最好位置为 $g$best。在第 k 次迭代中,第 s 个粒子按下式计算其下一代第 d(d = 1, ⋯, D) 维的飞行速度,并更新位置

$$v_{s,d}^{k+1} = v_{s,d}^k + c_1 \, rand_1^k (pbest_{s,d}^k - x_{s,d}^k) + c_2 \, rand_2^k (gbest_d^k - x_{s,d}^k) \quad (2)$$

$$x_{s,d}^{k+1} = x_{s,d}^k + v_{s,d}^{k+1} \quad (3)$$

其中: c_1、c_2 是加速系数,分别调节个体极值点($p$best) 和全局极值点($g$best) 对粒子速度的影响, $rand_1$、$rand_2$ 是[0, 1] 间的随机数,即粒子依据个体记忆和社会经验确定自己的运动速度,这使算法具有了群体智能。

M. Clerc[5] 认为收缩因子(constriction factor) 有助于确保 PSO 算法收敛。这种方法的速度更新公式为

$$v_{s,d}^{k+1} = \chi \, v_{s,d}^k + c_1 rand_1^k (pbest_{s,d}^k - x_{s,d}^k) + c_2 rand_2^k (gbest_d^k - x_{s,d}^k)) \quad (4)$$

其中,收缩因子为

$$\chi = 2 / |2 - \varphi - \sqrt{\varphi^2 - 4\varphi}| ;\ \varphi = c_1 + c_2 ,\ \varphi > 4 \quad (5)$$

Carlisle 等人[6] 优选了一组较为合理的参数, $c_1 = 2.8$, $c_2 = 1.3$[6] 。

此外还有很多有针对性的改进型[7] ,使 PSO 在解决复杂的优化问题和工程应用上显示出很大的优势,能与遗传算法等其他优越的优化算法相匹敌,而且因其结构简单、参数少、全局收敛快的特点备受关注,是一种很有前途的优化算法。然而,利用 PSO 进行多峰寻优的尝试较少,特别是在具体工程应用中,需要研究能有效利用峰值先验知识的针对性算法。

2.2 面向应用的多峰寻优 PSO 算法

2.2.1 由峰值分辨半径和峰值个数控制的局部最优解列表

标准 PSO 中的 $g$best 是单一的,它使整个种群中所有粒子均有向全局最优解靠拢的趋势,为得到局部极值,本文将其改为由多个较优的局部最优解组成的局部最优解列表($l$best 列表) 。笔者希望算法结束时在列表中得到与要求的多个峰对应的局部最优解,并且这些解依其峰值降序排列。为保证一个峰值在列表中仅对应惟一的 $l$best,规定任意两个 $l$best 间的距离满足

$$|lbest_i - lbest_j| > \Delta_{peak} ;\ i \neq j \quad (6)$$

列表长度设置为 N_{peak},这样,局部最优解列表是受应用参数直接控制的。列表对所有粒子公开,各粒子在每一代演化时从表中选择位置最近的 $l$best 代替式(4) 中的 $g$best。这样,表中多个峰值将整个种群划分为多个分组,并对组内粒子产生吸引力,而组内的粒子以其 $p$best 竞争该峰值对应的 $l$best。这里应该指出的是,粒子的归组是根据粒子的 $p$best 和目标函数的具体情况动态变化的,这种变化可能会导致组的分裂与合并,即多个 $l$best 在列表中是动态竞争的。算法演化过程中较优的 $l$best 将被插入列表或向前移位;较差的 $l$best 可能被近距的其它 $l$best 合并,或向后移位甚至被淘汰出列表。

2.2.2 由峰位置精度控制的收敛精度

考虑到实际问题中对解的精度要求总是有限的,如果一个峰附近的粒子聚集程度(离散度) 可以满足参数 A_{peak},则该峰没有必要再搜索下去,应将该峰内的所有粒子重新初始化以保持粒子群的整体活性,而将其对应的 Lbest 保留在表中(标记为停止峰) 参与后续的表项竞争。算法设计中,对列表的每个位子记一个平均半径 m_i,表示第 i 个峰($lbest_i$) 与其内部所有粒子个体极值点的平均距离,设 C_i 表示第 i 个峰内的粒子数,则平均半径第 k 代的值定义为

$$m_i^k = \left(\sum_{c=1}^{C_i} |pbest_c^k - lbest_i^k| \right) / C_i \quad (7)$$

关于搜索的停止准则,目前普遍使用的是迭代次数限制或根据输出峰值稳定性判定,但根据式(4) ,即使粒子不在峰值附近也最终将失去活性,输出峰值稳定性准则在这里是不合适的。考虑到多个粒子在非峰值处高密度聚集的可能性极小,笔者认为当某个 $l$best 周围聚集的 $p$best 个数超过某个指定的数值 C_{peak} 且平均半径达到 A_{peak} 时,该峰的寻优完成。即搜索达到精度的条件是

$$m_i^k \leq A_{peak} ;\ C_i \geq C_{peak} \quad (8)$$

这里 C_{peak} 可取经验值,如 $C_{peak} = 10$。

2.2.3 算法流程

a) 粒子群初始化。根据应用问题设置粒子总数 $N_{particle}$,各粒子在 Ω中随机选择初始位置 X_s^1,并赋予随机的初始速度 V_s^1,记个体最优解 $pbest_s^1 = X_s^1$($s = 1, \cdots, N_{particle}$) 。

b) $l$best 列表初始化。创建一个长度为 N_{peak} 的空表,记表中每一项的初始适应度为 - ∞。

c) 第 k 代中第 s 个粒子按下述规则运动:

(a) 计算该粒子的 $p$best$_s^k$ 与表中每个 $l$best 的距离 $L_i = |pbest_s^k - lbest_i|$,依 $L_i \leq \Delta_{peak}$ 判断同峰($i = 1, 2, \cdots, N_{peak}$) 。若有同峰的 $l$best 则进一步判断其是否被标记为停止峰,是停止峰则该粒子重新初始化, $s = s + 1$ 转(c) 计算第 $s + 1$ 个粒子,不是停止峰则转(b) ;若 $p$best$_s^k$ 在表中无同峰的 $l$best 则此粒子有可能新建一个峰值,将 $p$best$_s^k$ 按其适应度顺序插入列表,插入成功则 $s = s + 1$ 重新进入(c) 计算第 $s + 1$ 个粒子,插入不成功转(d) 。

(b) 记列表中与 $p$best$_s^k$ 距离最近的局部最优解为 $lbest_{near}$,从 $p$best$_s^k$ 与 $lbest_{near}$ 中选择较优的保留在列表中,若 $p$best$_s^k$ 未能进入列表则转 d) 。

(c) 检查发生过替换的 $l$best 与列表中其他 $l$best 是否有同峰现象,若没有则转(d) ;若存在同峰的 $l$best,则保留最优者,

其他较差的同峰表项清空, 被清空的位子由后面表项前移一位补上; 发生变动的 *l*best 可能会优于表中居于前面的峰值, 为保证降序排列, 根据其适应度向前调整次序。

(d) 寻找与 $p\text{best}_s^k$ 最近的代表活动峰的 $l\text{best}_{\text{near}}$, 以 $l\text{best}_{\text{near}}$ 为该粒子的 *g*best, 按照式(4) 更新速度。其中, 收缩因子按式(5) 计算, 按式(3) 更新位置。

(e) 算法终止条件。当 *l*best 列表中每一个解的精度均满足式(8) , 即所有找到的较优峰均被标记为停止峰时算法终止。

3　优化实例对比分析

本文选择三种典型的多峰优化算法与本文算法进行对比测试, 这三种算法分别代表本文引言中提到的三类方法。算法 1, 结合 Monte Carlo 初值设定的单纯形优化算法(simplex algorithm, SM) [8], 在本文中简称 MC-SM。其中, SM 由 MATLAB 中的 fminsearch 函数实现; 算法 2, 利用动态生成子种群的小生境 PSO, 即 NichePSO, 按文献[9] 实现; 算法 3, 引入序列生境技术的 PSO 算法, 在本文中称为 SNT-PSO, 仿照文献[3] 中的方法实现。测试环境为 Pentium 4 CPU 2. 8 GHz, 512 MB RAM, 编程环境为 MATLAB R2007a。

测试函数 1: Rastrigin

$$\min f_1(x_1, x_2) = \sum_{i=1}^{2} [x_i^2 - 10\cos(2\pi x_i) + 10];$$
$$\text{subject to } -5 \le x_1, x_2 \le 5 \tag{9}$$

测试函数 2: Shubert

$$\max f_2(x_1, x_2) = \{\sum_{i=1}^{5} i\cos[(i+1)\times x_1 + i]\} \times \{\sum_{i=1}^{5} i\cos[(i+1)\times x_2 + i]\};$$
$$\text{subject to } -5 \le x_1 \le 10,\ -4 \le x_2 \le 8 \tag{10}$$

这两个基准测试函数在本文选定的解空间(见式(9) (10)) 内均具有大量的局部极值点。其中: f_1 只有 1 个全局最小值, 测试时要求算法搜索到较优的 9 个最小点(如图 1 中小方框所示) , 其取值见表 1; f_2 有 4 个全局极大值, 测试时要求算法搜索到较优的 8 个极大点(如图 2 中小方框所示) , 其取值见表 2。为在相同条件下进行对比, 各算法中相似的参数设置一致。NichePSO 和 SNT-PSO 中的生境半径(niche radius) 与本文算法中的峰分辨半径相同, 对 f_1 取 $\Delta_{\text{peak}} = [0.5\quad 0.5]$, 对 f_2 取 $\Delta_{\text{peak}} = [0.3\quad 0.3]$。三种 PSO 算法中对每个局部极值设置相同的粒子数, 如 NichePSO 的子群规模为 30, SNT-PSO 每次全局寻优使用 30 个粒子, 本文算法对 N_{peak} 个局部极值并行搜索所用的总粒子数为 $30 \times N_{\text{peak}}$ 个。各算法中的个性参数分别对两个测试函数调节到最佳值。所有算法的停止条件为输出 N_{peak} 个局部最优解, 且满足

$$|X_i - X_i^*| \le A_{\text{peak}};\quad i = 1, \cdots, N_{\text{peak}} \tag{11}$$

即输出的每个近似解与精确解(见表 1 和 2) 相比均达到一定的求解精度, 这里令解的精度 $A_{\text{peak}} = [0.01\quad 0.01]$。若允许充足的运算时间和估值次数, 四种算法在理论上均可以达到式(11) 的要求。

由于考察的四种算法迭代方式差异较大, 且迭代次数与算法中解的规模相关, 不宜依迭代次数对比收敛性能。考虑到在很多实际应用中对目标函数估值需占用较多的运算资源, 更合理的做法是将各算法输出解的精度随估值次数变化的曲线作为算法收敛性能的对比依据。其中, 输出解的精度定义为与真实解的平均误差, 按下式计算

$$\text{AES} = \left(\sum_{i=1}^{N_{\text{peak}}}\sum_{d=1}^{Dms} |x_{id} - x_{id}^*|\right) / (N_{\text{peak}} \times Dms) \tag{12}$$

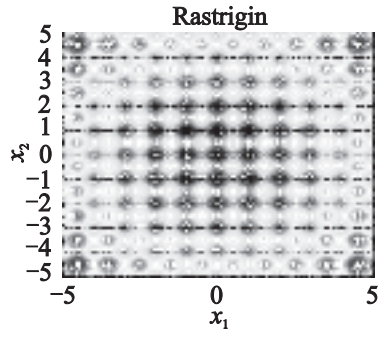

图1　测试函数1中的9个较优极值点

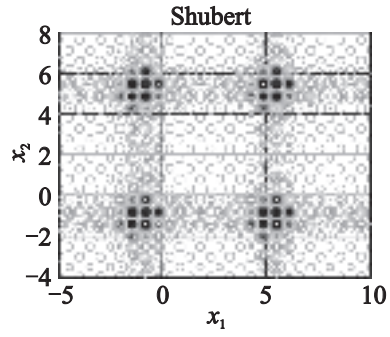

图2　测试函数2中的8个较优极值点

表1　测试函数1的9个较优极小解

i	X_i^*		$f(X_i^*)$
1	0	0	0
2	0	-0.995 0	0.995 0
3	-0.995 0	0	0.995 0
4	0	0.995 0	0.995 0
5	0.995 0	0	0.995 0
6	-0.995 0	0.995 0	1.989 9
7	0.995 0	-0.995 0	1.989 9
8	0.995 0	0.995 0	1.989 9
9	-0.995 0	-0.995 0	1.989 9

表2　测试函数2的8个较优极小解

i	X_i^*		$f(X_i^*)$
1	5.482 9	5.482 9	210.482 3
2	-0.800 3	-0.800 3	210.482 3
3	-0.800 3	5.482 9	210.482 3
4	5.482 9	-0.800 3	210.482 3
5	4.858 1	4.858 1	165.659 7
6	-1.425 1	-1.425 1	165.659 7
7	-1.425 1	4.858 1	165.659 7
8	4.858 1	-1.425 1	165.659 7

每种算法各运行 50 次取平均, 它们对两测试函数的收敛曲线分别见图 3 和 4, 算法结束时所用的平均估值次数和平均时间见表 3 和 4。因为 MC-SM 中的多边形和 NichePSO 中的子群均具有均匀分布的初始解, 且在演化时各搜索单元仅寻找最近的局部极值, 目标函数的各个峰值被找到的机会是均等的。由于 Rastrigin 函数的峰是光滑的, 单纯形对单峰的收敛性能优于局部 PSO 算法, MC-SM 的收敛曲线优于 NichePSO。Rastrigin 函数的最优峰在(0, 0) 处, 而大量次优峰有向最优峰集中的趋势(图 1) 。SNT-PSO 和本文算法都是利用所有粒子对全局较优的峰值进行搜索, 与 MC-SM 和 NichePSO 相比能够利用目标函数峰值集中的特性较快地收敛。在本文提出的算法中各粒子向最近的较优峰飞行, 与 SNT-PSO 中各粒子向剩余最优峰飞行的机制相比可节省一部分试探步数, 故其收敛性能略优于 SNT-PSO。表 3 中所示本文算法收敛耗时比 MC-SM 和 SNT-PSO 稍多, 但它所需的估值次数是最少的, 当目标函数估值运算较复杂时有理由相信本文算法具有相对较快的收敛速度。

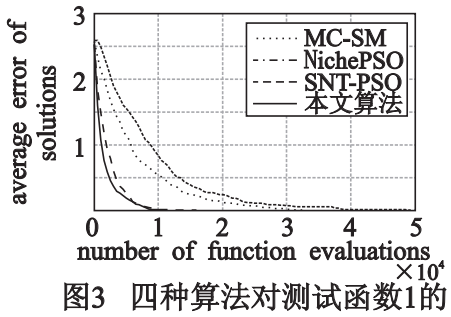

图3　四种算法对测试函数1的多峰寻优收敛曲线

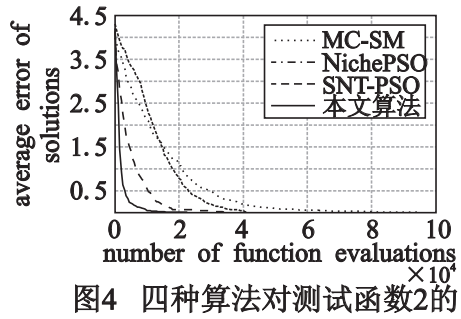

图4　四种算法对测试函数2的多峰寻优收敛曲线

表 3　四种算法对测试函数 1 的寻优代价

算法	MC-SM	NichePSO	SNT-PSO	本文算法
平均估值次数	18 357.20	24 403.28	8 911.08	6 555.60
平均时间/s	1.885 3	4.991 4	2.009 3	2.632 2

表 4　四种算法对测试函数 2 的寻优代价

算法	MC-SM	NichePSO	SNT-PSO	本文算法
平均估值次数	37 221.72	25 491.14	17 725.40	7 719.74
平均时间/s	4.011 5	4.643 4	3.633 5	2.890 4

Shubert 函数的特点是所要寻找的峰位置比较分散, 各全局极值相距较远, 且每个最高峰被一些次高峰所围。由于局部

极值过多, MC-SM 和 NichePSO 需要大量的局部搜索才能收敛到较大的 8 个峰, 收敛速度明显较慢; 而 SNT-PSO 和本文算法都具有很强的全局寻优能力, 能以较快速度收敛到分散的极值上(图 4) 。SNT-PSO 中粒子在分别向各极值聚集的过程中, 那些与目的极值距离较大的粒子需要较长时间达到稳定, 引入了多余的运算量, 而本文算法中各粒子仅向距离最近的较大峰值飞行, 所有粒子同时对多个较大峰进行搜索, 能以最快的速度到达目的极值附近。图 4 和表 4 的结果表明, 在对较优峰分散的多峰目标函数优化时, 本文算法对多峰的收敛性能明显优于其他三种算法。值得注意的是, 很多应用问题中目标函数的多个较优峰呈无规则分布, 且在主峰周围有副峰(或旁瓣) 存在, 与 Shubert 函数接近, 故本组测试结果具有一定的现实意义。

总体而言, MC-SM 和 NichePSO 算法在演化过程中, 因其运算量分散在整个解空间, 难以利用求解问题的先验知识加快向目的峰值的收敛, 适合用于峰值个数较少并需要找到所有局部最优解的场合, 而在对超多峰函数进行较优峰搜索时, 表现的收敛性能明显比本文算法差。SNT-PSO 虽然在收敛性能上与本文算法接近, 但其固有的串行运算结构限制了寻优效率。本文提出的算法自始至终受应用参数的引导, 能最有效地利用峰值个数、峰间分辨率和峰位置精度先验参数, 使粒子运动直接面向目的峰值, 从而获得最高的多峰寻优效率。此外, 从图 3 和 4 中的收敛曲线可以看出, 本文算法在演化的初期收敛速度非常快, 即用很少的试探次数就能识别出要求的多个峰。如果根据目标函数的具体情况在演化后期结合具有强局部收敛性的算法。可进一步形成各种实用的高性能多峰寻优算法, 本文提出的算法思想具有较大的开发潜力。

4 结束语

本文提出的改进型粒子群算法是一种基于实际系统参数面向应用的多峰优化算法, 与已有的三种代表性多峰寻优算法相比具有以下优势: a) 能以较小的运算代价快速地对目标函数的多个较优峰搜索极值; b) 能直接利用由实际系统性能和应用需求所确定的先验参数, 引导优化过程; c) 算法中的 *l*best 列表竞争、停止峰标记和粒子重新初始化的机制有效地将运算量调整到新峰值的发现上, 使算法首先以很快的速度收敛到多个峰, 然后利用冗余的时间对 *l*best 列表进行可能的修正, 平衡了高效与可靠的实用需求; d) 该算法在演化早期对多峰的收敛速度非常快, 为后期收敛的改进提供了较大余地, 可衍生出多种实用多峰寻优算法。

参考文献:

[1] RITS R, ENGELBRECHT A P, Van DEN BERGH F. Locating multiple optima using particle swarm optimization[J]. Applied Mathematics and Computation, 2007, 189: 1859-1883.

[2] BRITS R, ENGELBRECHT A P, Van DEN BERGH F. A niching particle swarm optimizer[C]//Proc of the 4th Asia-Pacific Conference on Simulated Evolution and Learning. 2002: 692-696.

[3] BEASLEY D, BULL D, MARTIN R. A sequential niche technique for multimodal function optimization[J]. Evolutionary Computation, 1993, 1(2): 101-125.

[4] KENNEDY J, EBERHART R C. Particle swarm optimization[C]// Proc of IEEE International Conference on Neural Networks. Perth: [s. n.], 1995: 1942-1948.

[5] CLERC M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization[C]//Proc of Congress on Evolutionary Computation. Piscataway: IEEE Service Center, 1999: 1951-1957.

[6] CARLISLE A, DOZIER G. An off-the-shelf PSO[C]//Proc of Workshop on Particle Swarm Optimization. Indianapolis: [s. n.], 2001: 1-6.

[7] Van DEN BERGH F. An analysis of particle swarm optimizers[D]. South Africa: Department of Computer Science, University of Pretoria, 2002.

[8] LAGARIAS J C, REEDS J A, WRIGHT M H, *et al*. Convergence properties of the nelder-mead simplex method in low dimensions[J]. SIAM Journal of Optimization, 1998, 9(1): 112-147.

[9] ZCAN E, YILMAZ M. Particle swarms for multimodal optimization [C]//BELICZYNSKI B, *et al*., Lecture Notes in Computer Science. [S. l.]: Springer-Verlag, 2007: 366-375.

(上接第 3616 页)

[7] RUBER T R. Towards principles for the design of ontologies used for knowledge sharing[J]. International Journal of Humarr Computer Studies, 1994, 43(5-6): 907-928.

[8] SU Xiao-meng, HAKKARAINEN S, BRASETHVIK T. Semantic enrichment for improving systems interoperability[C]//Proc of ACM Symposium on Applied Computing. 2004: 14-17.

[9] 田磊, 覃征, 衡星辰, 等. 基于本体的多源异构 XML 数据近似查询方法[J]. 西安交通大学学报, 2007, 41(6): 702-706.

[10] 马骞, 杨以涵, 郭金智. 基于本体的调度决策支持软件系统的设计[J]. 电力系统自动化, 2004, 28(20): 57-62.

[11] 王建州, 李廉, 汪映海. 面向 ontology 的电力信息系统的研究[J]. 北京邮电大学学报, 2005, 28(z1): 116-120.

[12] GRUBER T. A translation approach to portable ontology specifications [J]. Knowledge Acquisition, 1993, 5: 199-220.

[13] IEC TC57 wS13. Draft IEC 61970, Energy management system application program interface(EMS-API)[S]. 2001.

[14] McCARTHY J. Programs with common sense[C]//MINSKY M. Proc of Semantic Information. [S. l.]: MIT Press, 1968: 403-418.

[15] FINK L H, CARLSEN K. Operating under stress and strain[J]. IEEE Spectrum, 1978, 15: 48-53.

[16] WILLEM N B. Construction of engineering ontologies for knowledge sharing and reuse[D]. Netherlands: University of Twente, 1997.

[17] CHANDRASEKARAN B, JOSEPHESON J, BENJAMINS V R. Ontologies: what are they? why do we need them? [J]. IEEE Intelligent Systems, 1999, 14(1): 20-26.

[18] USCHOLD M, GRUNINGER M. Ontologies: principles, methods and applications[J]. Knowledge Engineering Review, 1996, 11(2): 93-155.

[19] IEC TC57. IEC 61850-7, Communication networks and systems in substations[S]. 2003.

[20] 王萍, 王利军, 王险峰. 面向对象电网知识库系统的研究与实践[J]. 电力系统自动化, 2002, 26(11): 62-65.

[21] 刘昕鹏. Ontology 理论研究与应用建模[EB/OL]. [2003]. http://gis.pku.edu.cn/Resources/TR.

[22] 徐振宁, 黄凯歌, 张维明, 等. Ontology 建模方法研究[J]. 计算机科学, 2002, 29(1): 68-71.