

文章编号: 1001-0920(2011)03-0381-06

简化的分类微粒群算法及其在风电场建模中的应用

陈国初¹, 杨 维^{1,2}, 张延迟¹, 徐余法¹, 俞金寿²

(1. 上海电机学院 电气学院, 上海 200240; 2. 华东理工大学 信息科学与工程学院, 上海 200237)

摘要: 提出一种简化的分类微粒群算法. 首先将微粒按适应值的差异划分成较好、普通和较差 3 类; 然后对这 3 类微粒分别采用 3 种对应的没有速度项的简化模型进行动态调整, 有效地增加了种群的多样性. 通过对 4 种典型测试函数的仿真实验, 并与经典 PSO 和 2 个目前较为流行的改进 PSO 进行比较, 实验结果表明了所提出的改进算法具有更好的优化性能. 将改进算法用于风电场风速概率模型优化的实验结果表明, 与传统最小二乘法相比, 该方法拟合的 Weibull 参数精度更高, 更具实际参考价值.

关键词: 微粒群优化算法; 简化微粒群优化算法; 微粒分类; 动态模型; Weibull 模型

中图分类号: TP273

文献标识码: A

Simplified classification PSO and its application in wind farm modeling

CHEN Guo-chu¹, YANG Wei^{1,2}, ZHANG Yan-chi¹, XU Yu-fa¹, YU Jin-shou²

(1. Electric Engineering School, Shanghai Dianji University, Shanghai 200240, China; 2. College of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China. Correspondent: CHEN Guo-chu, E-mail: chgsh@yahoo.com.cn)

Abstract: A simplified classification particle swarm optimization algorithm(PSO) is proposed. At first, particles are divided into three categories, such as the better, ordinary and the worse according to their fitness. Then, three types of simplified models without velocity part in classical particle swarm optimization algorithm are used to adjust these three kinds of classified particles respectively. The diversity of algorithm is enhanced effectively. Through the simulation experiments with four test functions, compared with the basic PSO and another improved PSO currently, the improved algorithm proposed has better optimization performance. Finally, the improved algorithm is applied to optimize wind probability modeling, and the results show that this method has more accuracy and more practical reference than least-squares method.

Key words: PSO; simplified PSO; particle classification; dynamic model; Weibull model

1 引 言

微粒群优化算法(PSO)是 Kennedy 和 Eberhart 等人^[1-2]于 1995 年提出的一种新的进化计算算法, 其核心思想来源于群体中个体之间信息的社会共享和协同进化. PSO 具有参数少、结构简单、高效、收敛快等优点, 在解决高维、多峰、非线性等问题中得到了广泛的应用^[3-5].

尽管微粒群算法具有简单、高效的优点, 但在求解复杂的高维、多峰函数时存在虽然全局模型收敛速度快但易陷入局部最优, 局部模型进化后期收敛速度慢等缺陷. 为了克服这种缺陷, 近几年出现了许多关于这些方面的改进方法. 比如, 惯性权值

的线性递减 PSO^[6], 这种方法的进一步发展是模糊自适应 PSO^[7], 带收缩因子的 PSO^[8]以及加速系数时变 PSO^[9]等. 这些经典方法在低维、单峰函数中取得了比较好的效果, 但在高维、多峰函数上依然难以有效避免陷入局部最优. 文献[10]引入突跳因子来改变微粒的既定速度, 在一定程度上提高了收敛精度, 但其重大缺陷是发生突跳的条件是置微粒速度为零, 这样会使微粒完全丧失活性, 随着迭代的进行, 种群的多样性急剧减弱, 使微粒不能快速、有效地逃出局部最优. 文献[11]提出一种新型多步式位置可选择更新的粒子群算法, 将标准粒子群中速度的单步更新分解成 3 步更新, 取所生成的 3 个位置中的最好位置作为

收稿日期: 2009-12-08; 修回日期: 2010-04-10.

基金项目: 国家自然科学基金项目(70773041); 上海市教委重点学科项目(J51901); 上海市教委科研创新重点项目(09ZZ211); 闵行区-上海电机学院区校合作项目(08Q07).

作者简介: 陈国初(1971—), 男, 副教授, 博士, 从事复杂系统建模、仿真与智能算法等研究; 俞金寿(1939—), 男, 教授, 博士生导师, 从事工业过程建模、仿真、优化与控制等研究.

最终结果,细化了粒子的搜索轨迹,在不增加算法复杂度的条件下提高了个体极值以及全局极值的更新速率,改善了算法的收敛速度和精度,但存在3个模型的更新步长太单一,收敛精度受局限的问题。

尽管这些方法在一定程度上提高了算法的收敛速度和精度,但这些改进措施并未从本质上改善微粒陷入局部极值的缺陷,其根本原因在于对所有微粒均采用统一的迭代模型,迭代方式过于单一,从而混淆了不同微粒对全局搜索能力和局部开发能力的需求差异。因此,本文首先根据微粒适应值的差异将其划分成较好、普通和较差3种类型,然后用3种对应的没有速度项的PSO简化模型(认知模型、完全模型和社会模型)分别进行动态优化,即适应值好的微粒放慢速度在其邻域内细致搜索,适应值差的微粒加快速度全局搜索,适应值一般的微粒进行动态平衡搜索。这种改进的算法记为PSO-SPW。

针对传统方法,如最小二乘法、矩估计法、极大似然法等,在求解用于拟合风电场风能资源特性的Weibull分布模型的双参数时,存在效率低、精度差等缺陷,本文尝试将上述改进型智能算法PSO-SPW用于Weibull双参数的建模与优化。

2 基本微粒群算法

假设有 m 个微粒组成的一微粒群,在 D 维搜索空间中每个个体都是优化问题的一个潜在解,其中第 i 个微粒的空间位置为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD}), i = 1, 2, \dots, m$ 。将它代入优化目标函数可计算出相应的适应值,根据适应值可衡量 x_i 的优劣。第 i 个微粒所经历的最好位置称为其个体历史最好位置,记为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$,对应的为个体历史最好适应值 F_i ,具有各自的飞行速度 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ 。所有微粒经历过的位置中的最好位置称为全局历史最好位置,记为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$,相应的适应值为全局历史最好适应值 F_g 。对于第 k 代的第 i 个微粒,其第 d 维速度、位置($1 \leq d \leq D$)根据如下方程迭代:

$$v_{id}^{(k+1)} = \omega v_{id}^{(k)} + c_1 r_1 (p_{id}^{(k)} - x_{id}^{(k)}) + c_2 r_2 (p_{gd}^{(k)} - x_{id}^{(k)}), \quad (1)$$

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

其中: ω 为当前惯性权值,本文约定 $\omega = \omega_{\max} - \text{iter}(\omega_{\max} - \omega_{\min})/\text{MaxIt}$, iter 为当前迭代次数, ω_{\max} 和 ω_{\min} 为 ω 的上下限, MaxIt 为最大允许迭代次数; c_1 和 c_2 均为正常数,称为学习因子; r_1 和 r_2 为2个在 $[0, 1]$ 范围内变化的随机数。通常这种惯性权值递减的PSO也被称为基本PSO(记为PSO-LDW)。

由速度迭代公式可以看出:在基本PSO中,第1部分是惯性部分,表示粒子对上一次迭代速度的继承;

第2部分为认知部分,是粒子对自身的思考;第3部分为社会部分,表示粒子间的信息共享和相互合作。当 $c_1 \neq 0, c_2 = 0$ 时,对应认知模型;当 $c_1 = 0, c_2 \neq 0$ 时,对应社会模型;当 $c_1 \neq 0$ 且 $c_2 \neq 0$ 时,对应完全模型。认知模型只考虑粒子本身的信息,缺少社会信息的交流和共享,所以收敛速度慢;社会模型只考虑群体因素,倾向于向群体学习,收敛速度比较快,但容易早熟。为了平衡群体因素和个体因素的影响,文献[12]提出一种局部和全局混合的动态调整PSO,实例表明,该算法在高维、多峰函数中取得了较好效果,但这种调节极其有限,稳定性不够好。

通过以上分析可知,绝大多数改进算法难以有效避免局部最优和进化后期收敛速度慢等缺陷,根本原因在于混淆了不同微粒对全局搜索和局部开发的需求能力的需求差异,所以有必要对微粒进行分类,不同的微粒使用不同的模型调整,这样每个微粒的搜索目标和方向更具明确性,减少了搜索的盲目性,从而不仅加快了收敛速度,同时提高了收敛精度。

3 改进的微粒群算法

3.1 简化的微粒群算法

通过分析基本微粒群算法的进化迭代方程(1)和(2)可以发现,在微粒群算法中粒子速度的概念不是必需的。从基本PSO的2个迭代方程看,微粒群最终的目的是让粒子的位置向量 x 逼近解空间的最优解。因此,微粒在整个搜索过程中,其轨迹变化最本质的体现是其位置向量 x 的变化过程,所以可直接考察 x 的变化轨迹。另外,基本PSO模型中的速度不是人们常说的物理意义上的速度,即位移对时间的一阶导数,而仅仅只有数学上的意义,其方向表示微粒下一步的迭代方向,其大小表示微粒下一步的迭代步长,但这个方向有可能偏离微粒的正确进化方向,从而使得微粒难以收敛。另外,经典PSO中的式(2)直接将位置和速度相加,不符合客观运动规律。由以上分析可知,微粒进化方程的速度项完全可以去掉,这样有以下2个好处:1)原来的进化方程由二阶变成一阶,从而为对微粒的收敛性运动轨迹分析提供较大的便利;2)避免对于不同问题,由于人的主观判断差异,使设置的最大速度阈值有可能影响微粒的收敛速度和精度。综上所述,式(1)和(2)可合并为一个单一的进化方程,该算法模型记为PSO-NV,其方程如下:

$$x_{id}^{(k+1)} = \omega x_{id}^{(k)} + c_1 r_1 (p_{id}^{(k)} - x_{id}^{(k)}) + c_2 r_2 (p_{gd}^{(k)} - x_{id}^{(k)}). \quad (3)$$

由位置迭代公式可以看出,在去掉速度项的PSO中,第1部分是惯性部分,表示粒子对上一次迭代速度的继承;第2部分为认知部分,是粒子对自身的思

考; 第 3 部分为社会部分, 表示粒子间的信息共享和相互合作. 当 $c_1 \neq 0, c_2 = 0$ 时, 对应认知模型; 当 $c_1 = 0, c_2 \neq 0$ 时, 对应社会模型; 当 $c_1 \neq 0$ 且 $c_2 \neq 0$ 时, 对应完全模型. 可见, 与原基本 PSO 相比, 简化的 PSO 不仅迭代原理未发生变化, 而且结构更加简单.

3.2 对微粒进行分类

由前述分析可知, 在微粒群的进化过程中, 非常有必要根据各个微粒的适应值的差异将其进行合理分类, 这样才能有效保证不同微粒对全局搜索和局部搜索的需求能力差异的平衡. 为此, 本文提出根据微粒适应值的均值将微粒划分为 3 类, 其位置进化原理如图 1 所示.

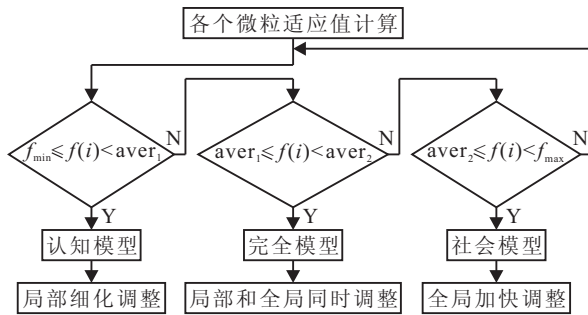


图 1 改进算法的位置更新模型框图

这里以极小化为例, 在第 k 代进化中求出所有微粒的均值 $aver$, 最好适应值 f_{min} , 最差适应值 f_{max} , 分别将具有这两种适应值的微粒称为最好微粒和最差微粒; 然后继续求出位于 f_{min} 与 $aver$ 之间的所有微粒的均值 $aver_1$, 位于 $aver$ 与 f_{max} 之间的所有微粒的均值 $aver_2$. 将位于 f_{min} 与 $aver_1$ 之间的微粒称为较好微粒, 将位于 $aver_1$ 与 $aver_2$ 之间的微粒称为普通微粒, 将位于 $aver_2$ 与 f_{max} 之间的微粒称为较差微粒. 具体的进化方程如下:

1) 较好微粒的进化模型. 若微粒的适应度满足

$$f_{min} \leq f(i) < aver_1,$$

则该部分微粒是群体中较为优秀的个体, 它们已经接近全局最优, 所以采用认知模型对其进行调整. 这样可以让这些微粒在其邻域附近作更为细致的搜索, 而且速度稍微变慢, 可保证以更好的收敛精度向全局最

优值靠近, 其位置进化方程为

$$x_{id}^{(k+1)} = \omega x_{id}^{(k)} + c_1 r_1 (p_{id}^{(k)} - x_{id}^{(k)}). \quad (4)$$

2) 普通微粒的进化模型. 若微粒的适应度满足

$$aver_1 \leq f(i) < aver_2,$$

则该部分微粒是群体中较为普通的个体, 即中等微粒. 这些微粒与全局最优的距离适中, 所以采用完全模型动态调整, 其位置进化方程为

$$x_{id}^{(k+1)} = \omega x_{id}^{(k)} + c_1 r_1 (p_{id}^{(k)} - x_{id}^{(k)}) + c_2 r_2 (p_{gd}^{(k)} - x_{id}^{(k)}). \quad (5)$$

3) 较差微粒的进化模型. 若微粒的适应度满足

$$aver_2 \leq f(i) \leq f_{max},$$

则该部分微粒是群体中较差的个体, 这些微粒距离全局最优值较远, 应采用社会模型进行调整, 使这些微粒加快靠近全局最优解的收敛速度, 其位置进化方程为

$$x_{id}^{(k+1)} = x_{id}^{(k)} + c_2 r_2 (p_{gd}^{(k)} - x_{id}^{(k)}). \quad (6)$$

在完全模型的进化过程中, 迭代初期, c_1 取较大值, c_2 取较小值有利于加强微粒的自我学习能力; 在迭代后期, c_1 取较小值, c_2 取较大值有利于加强微粒间的信息共享和群体之间的学习能力, 这样可继承变加速系数 PSO 的优秀部分. 在认知模型和社会模型的进化过程中, 为加重认知模型的自身认知能力和社会模型的社会学习能力, 将完全模型中的 2 个学习因子相加, 即 $c = c_1 + c_2$. 因为惯性权重 ω 具有平衡全局探索和局部开发的能力, 所以这 3 个进化模型 ω 均采用基本 PSO 中的线性递减方式.

在算法复杂度方面, PSO-SPW 并没有增加选择、交叉和变异等之类的操作, 主要是将微粒按其适应值的好坏分成 3 类, 每一类按相应的模型动态调整, 这些步骤都是并行操作. 而本文算法去掉了速度项, 使迭代过程更加简化, 所以比基本 PSO 算法更为简洁, 同时具有更高的搜索效率和精度.

4 仿真实验

4.1 测试函数

测试函数及其参数设置如表 1 所示.

表 1 各个测试函数及其参数设置

函数名	表达式	维数	变量范围	最优解	误差限
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	40	[-100, 100]	0	1e-6
Rosenbrock	$f_2(x) = \sum_{i=1}^n 100(x_{i+1}^2 - x_i)^2 + (x_i - 1)^2$	30	[-600, 600]	0	10
Griewangk	$f_3(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-100, 100]	0	1e-6
Schaffer F 6	$f_4(x) = 0.5 + \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1 + 0.001(x^2 + y^2))^2}$	2	[-100, 100]	0	1e-6

表1所示的4个测试函数中: Sphere函数 $f_1(x)$ 是较为简单的单峰函数, 只有一个全局最小点 $(0, 0, \dots, 0)$, 最小值为0, 可以测试算法的寻优精度, 考察算法的执行性能; Rosenbrock函数 $f_2(x)$ 为单峰函数, 各变量之间有很强的耦合性, 难以进行全局极小化, 在 $x_i = 1 (i = 1, 2, \dots, n)$ 时取得唯一的全局最小值0, 在 $x_i = 1 (i = 1, 2, \dots, n)$ 时取得唯一的全局最小值0, 可用于评价算法的执行效率; Griewangk函数 $f_3(x)$ 为多峰值函数, 全局最小值为0, 在 $x_i = 0 (i = 1, 2, \dots, n)$ 时达到全局最小值; Schaffer F6函数 $f_4(x)$ 有无数个极小点, 其中只有一个 $(0, 0)$ 为全局最小, 最小值为0. Schaffer F6和Griewangk这2个多峰值函数都有很多局部极小值点, 一般算法较难找到全局最优值, 因此可用来检验算法跳出局部最优的能力.

4.2 测试结果与分析

为分析比较改进算法PSO-SPW的性能, 本文同时采用上述经典PSO-LDW, 简化算法PSO-NV和目前较好的改进PSO算法(如文献[11]中的PSO-MP和文献[12]中的PSO-DGLC)对上述4个优化函数进行测试. 在参与比较的几个优化算法的文献中, 由于各个测试函数的测试范围、变量维数以及所选精度不尽相同, 为更客观地比较各种算法的优化性能, 本文在所有算法中都设置统一的优化参数. 寻优时, $f_1(x) \sim f_4(x)$ 的维数分别取40, 30, 30, 2, 微粒的数目为30, 最大迭代代数500, 除了Rosenbrock函数 $f_2(x)$ 的适应值误差限设置为10, 其他函数均设置为 $1e-6$. ω 从0.9线性递减到0.4, c_1 从2.05线性递减到0.5, c_2 从

0.5线性递增到2.05, $c = 4.0$. 现设计如下2个实验:

实验1 给定最大迭代次数, 比较收敛精度;

实验2 同时给定迭代次数和适应值误差限, 比较收敛速度和达优率.

本文实验的收敛精度用平均适应值度量, 收敛速度用平均迭代次数度量. 在实验1中, 为更直观地比较各个函数的寻优效果, 对各个函数的适应度采用以10为底的对数, 同时为避免真数为0, 在每个适应值后面加上 $1e-7$ 作为截止值, 各个函数的寻优趋势如图2所示. 在实验2中, 规定在此条件下的函数最终适应值满足 $|f - f_{\min}| < \text{Err}$ 时(Err为给定的适应值误差限), 表示此次实验成功, 否则失败, 即达优率 $S = \text{成功次数} / \text{实验总次数} \times 100\%$. 为减少优化过程中随机性的影响, 每个算法对上述4个测试函数分别进行100次独立实验, 然后进行统计分析. 函数适应度收敛曲线和测试结果统计如图2所示.

由图2所示4个进化曲线可明显看出: PSO-SPW, PSO-MP和PSO-DGLC的收敛精度比经典PSO-LDW和PSO-NV有显著提高, 而PSO-SPW的收敛精度实际上比PSO-MP和PSO-DGLC高, 但本文为更好地观察各个算法的进化曲线, 并防止纵坐标的范围过大, 在函数 $f_1(x) \sim f_4(x)$ 的适应值后面加上了 $10e-6$ 作为误差限, 影响了对3个较优的改进PSO算法的实际收敛精度的观察. 收敛精度的提高主要得益于: 在PSO-SPW的每次迭代中将微粒进行了合理的分类, 对不同类型的微粒采用不同的简化的进化模型, 较好

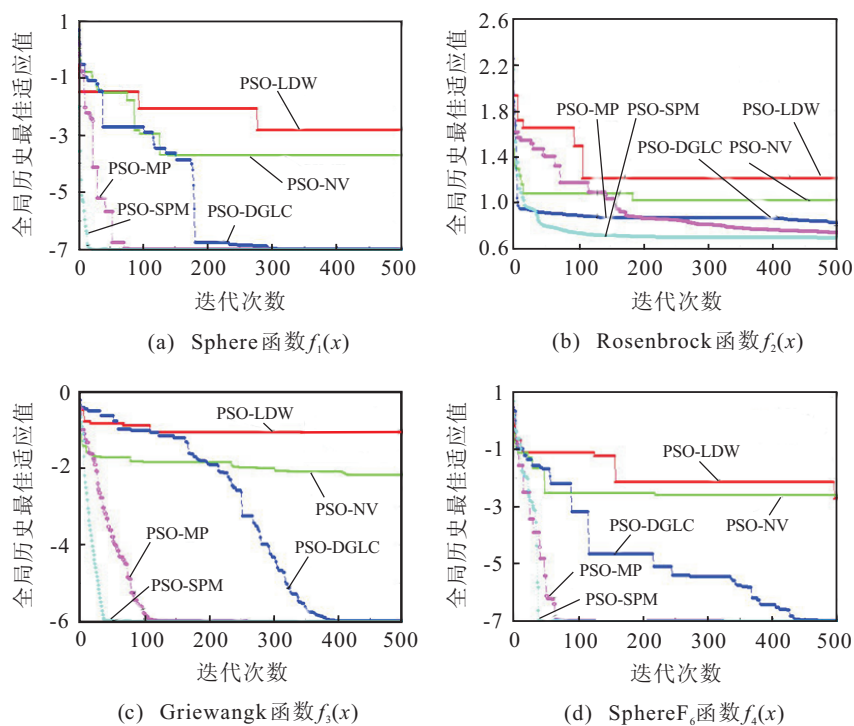


图2 函数 $f_1(x) \sim f_4(x)$ 在5个算法中的适应值进化曲线

表2 给定最大迭代次数下不同函数的仿真结果

算法类型	测试函数 $f_1(x)$		测试函数 $f_2(x)$		测试函数 $f_3(x)$		测试函数 $f_4(x)$	
	最优适应值	平均适应值	最优适应值	平均适应值	最优适应值	平均适应值	最优适应值	平均适应值
PSO-LDW	0.016 2	0.023 5	21.385	27.83	0.017 7	0.024 7	0.018 1	0.022 8
PSO-NV	0.011 5	0.021 3	20.092	26.97	0.011 5	0.022 6	0.016 9	0.020 4
PSO-DGLC	0	5.6e-9	0.410 7	9.943	0	7.8e-8	0	5.1e-8
PSO-MP	0	1.74e-11	0.196 2	5.184	0	2.1e-10	0	4.8e-10
PSO-SPW	0	5.8e-14	0.002 8	3.071	0	3.7e-12	0	2.6e-12

表3 同时给定迭代次数和误差限不同函数的仿真结果

算法类型	测试函数 $f_1(x)$		测试函数 $f_2(x)$		测试函数 $f_3(x)$		测试函数 $f_4(x)$	
	达优率	平均迭代步数	达优率	平均迭代步数	达优率	平均迭代步数	达优率	平均迭代步数
PSO-LDW	2	496.7	0	500	0	500	0	500
PSO-NV	9	487.3	0	500	0	500	0	500
PSO-DGLC	96	289.4	54.7	473.4	75	384.6	71	437.5
PSO-MP	100	67.8	78	421.6	84	113.7	81	74.8
PSO-SPW	100	23.5	94	235.2	97	47.3	95	46.2

地提高了算法的优化效率和优化精度;而 PSO-NV 又比经典 PSO-LDW 的收敛精度高,这在于去掉速度项后,微粒的进化过程更加简单,从而收敛速度相应提高,这也表明了去掉速度项的合理性。

从表2可以看出: PSO-SPW, PSO-MP 和 PSO-DGLC 无论是对单峰函数还是对多峰函数的优化,其收敛精度都能大大提高,相同条件下函数的平均适应值都比经典 PSO 更优。当经典 PSO 已经陷入局部最优时,这3个改进算法都能朝全局最优解的方向收敛,与 PSO-MP 和 PSO-DGLC 相比,在相同的迭代次数下, PSO-SPW 的收敛精度更优。

从表3可以看出:在达优率方面,后3者依然比前2者的优化效率有明显提高,而且收敛速度也有显著提升;从迭代次数可以看出,后3者有明显改善;从收敛速度看, PSO-SPW 较 PSO-MP 和 PSO-DGLC 都有很大改善。这些都表明了该算法的有效性和合理性。

5 风电场风速概率分布模型

在各种可再生能源的利用中,风能作为一种分布广泛、取之不尽、无污染的可再生能源,受到了广泛关注。20世纪70年代末,风电场的概念首先在美国提出,经过近30年的发展,截止到2008年底,我国除台湾省外累计安装风电机组11600多台,装机容量约1215.3万kW。

风电场的风能资源状况评估是开发风电场建设最基础的工作,一个地区的风能资源状况依赖于该地区风的统计特性,而体现风的统计特性的一个重要形式是风速的概率。对于由于气候地理因素的复杂性而形成各地风速分布多变的现实,目前还没有从成因和理论上给出某地风速概率分布的形式。为较精确地描述风速的随机性、间歇性及其变化特性,目前对风况只能进行长期观测,并进行统计分析才能掌握风速分布及其变化规律,在此基础上建立数学

统计模型表示实际风速分布。目前有不少统计模型用于拟合风速的分布,如瑞利(Rayleigh)分布^[13], β 分布^[14],韦布尔(Weibull)分布^[15]等,但应用最多的是两参数 Weibull 分布模型,因为它能更好地拟合实际风速曲线。

5.1 Weibull 分布模型

设由测风仪获得某风电场的风速序列 (V_1, V_2, \dots, V_n) 服从两参数 Weibull 分布,则分布函数为

$$F(V) = P(v \leq V) = 1 - \exp \left[- \left(\frac{V}{c} \right)^k \right], \quad (7)$$

其概率密度函数为

$$f(V) = \left(\frac{k}{c} \right) \left(\frac{V}{c} \right)^{k-1} \exp \left[- \left(\frac{V}{c} \right)^k \right]. \quad (8)$$

式中: c 和 k 分别为 Weibull 分布的尺度参数和形状参数,这2个参数控制 Weibull 分布曲线的形状; V 为给定风速,单位为 m/s。尺度参数 c 反映该风电场的平均风速,其量纲与速度相同; k 表示分布曲线的峰值情况,无量纲,若风速趋向于非常接近的一些值,则 k 取很高的值,分布曲线形状比较尖。

5.2 Weibull 参数计算

由 Weibull 模型可知,只要确定了参数 c 和 k ,则体现风资源性能的指标(如年平均风速、有效风能密度和年风能可利用时间)都可以求得。在具体使用过程中,参数的确定应根据当地可供使用的风速统计资料的不同情况作出不同的选择,通常可采用如下方法:最小二乘法(LSM)^[16],极大似然估计法^[17]和矩估计法^[18]等。这些传统的统计方法比较简单,难以满足实际精度要求。为此,本文尝试采用上述改进的智能算法 PSO-SPW 对 Weibull 的2个参数进行寻优。首先建立如下目标优化函数:

$$E(k, c) = \sum_{i=1}^N |F_{v_i} - f_{v_i}|^2. \quad (9)$$

表4 东北某风电场60m高处风速的Weibull参数计算结果

测风时间	年平均风速 \bar{V} /(m/s)	LSM		误差/%		PSO-SPW		误差/%		WASP	
		k	c	k	c	k	c	k	c	k	c
2007-04-07~2008-04-06	6.24	2.21	6.91	9.41	7.80	2.11	6.69	4.46	4.37	2.02	6.41
2008-04-07~2009-04-06	6.35	2.24	7.02	9.27	7.67	2.13	6.81	3.90	4.45	2.05	6.52

式中: E 为误差平方和函数, 其极小值对应的 k 和 c 即为所要求的最优解 (k, c); F_{v_i} 为实际第 i 个风速值 V_i 的概率; f_{v_i} 为将 V_i 代入式 (8) 得到的韦布尔概率值; N 为根据原始风速数据所分的风速段数.

下面的实例是根据某风电公司提供的东北某风电场近两年的测风数据, 从中选取 60 m 高的测风塔 2007-04-07~2008-04-06 和 2008-04-07~2009-04-06 这 2 年的测风数据. 这些数据是每隔 10 min 测 1 个风速点, 其中 2007-04-07~2008-04-06 共有 $6 \times 24 \times 366 = 52704$ 个数据, 在 2008-04-07~2009-04-06 共有 $6 \times 24 \times 365 = 52560$ 个数据. 计算结果如表 4 所示.

从表 4 可以看出, 与标准风速 Weibull 参数计算软件 WASP 算出的参数相比, PSO-SPW 算法的形状参数 k 和尺度参数 c 都比 LSM 的计算精度高, 其中 k 的最小误差仅为 3.90%, 尺度参数 c 最小误差为 4.37%, 均不高于 5%, 完全符合实际风电场建设要求. 因此, 该方法对于实际风电场选址具有较好的参考价值. 图 3 是分别采用 PSO-SPW 和 LSM 对 2007-04-07~2008-04-06 这一年的 60 m 高测风数据进行优化的 Weibull 参数所拟合的风速概率曲线与实际风速概率曲线的对比图.

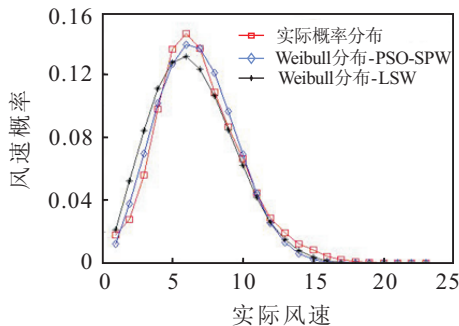


图3 实际风速概率分布与 Weibull 分布对比

从图 3 可以看出, 采用 PSO-SPW 优化的 k 和 c 拟合的曲线比采用 LSM 拟合的曲线精度高, 然而与实际分布曲线仍存在一定误差, 但这些误差在实际工程上可以满足要求. 总体而言, 与传统的统计方法相比, 采用智能算法所拟合的风速概率曲线的精度更高, 更具参考性.

6 结 论

针对传统微粒群算法在求解复杂的高维、多峰函数时, 虽然全局模型收敛速度快但存在易陷入局部最优, 局部模型进化后期收敛速度慢等缺陷, 本文提出了一种简化的分类微粒群算法. 该算法首先根据

微粒的适应值大小将各个微粒分成较好、普通和较差 3 类; 然后通过对基本 PSO 方程的速度项进行合理分析并将其去掉, 简化了微粒的进化方程. 在此基础上, 对 3 类微粒采用对应的 3 种模型 (认知模型、完全模型和社会模型) 分别进行动态调整. 在 3 种模型中, 惯性权值同时线性调整; 在完全模型中, c_1 和 c_2 线性调整. 这样不但继承了经典 PSO 的有利部分, 同时也是为了满足不同类型的微粒对全局搜索与局部开发所需要差异的一种合理的平衡. 通过对 4 种典型测试函数的仿真实验, 并与经典 PSO 和 2 个目前较好的改进 PSO 比较, 仿真实例表明了该改进算法的合理性和有效性. 此外, 从对算法复杂度的分析看, 改进算法的复杂度有所降低. 将 PSO-SPW 算法用于风电场风速概率模型优化的实例表明, 与常规 LSM 相比, 该算法的拟合精度更高, 更具参考价值.

参考文献(References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of the IEEE Int Conf on Neural Networks. Piscataway: IEEE Service Center, 1995: 1942-1948.
- [2] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of the 6th Int Symposium on Micro Machine and Human Science. Nagoya: IEEE Service Center, 1995: 39-43.
- [3] Eberhart R C, Shi Y. Particle swarm optimization: developments applications and resources[C]. Proc of the IEEE Congress on Evolutionary Computation. Piscataway: IEEE Service Center, 2001: 81-86.
- [4] Hu Xiaohui, Shi Yuhui, Eberhart R C. Recent advances in particle swarm[C]. Proc of IEEE Congress on Evolutionary Computation. Piscataway: IEEE Press, 2004: 90-97.
- [5] 陈国初, 俞金寿. 增强型微粒群优化算法及其在软测量中的应用[J]. 控制与决策, 2005, 20(4): 377-381. (Chen G C, Yu J S. Enhanced particle swarm optimization algorithm and its application in soft sensor[J]. Control and Decision, 2005, 20(4): 377-381.)
- [6] Shi Y H, Eberhart R C. Empirical study of particle swarm optimization[C]. Proc of IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 1999: 1945-1950.
- [7] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization[C]. Proc IEEE Int Conf on Evolutionary Computation. Seoul: IEEE Service Center, 2001: 101-106.

(下转第392页)