

# 基于多目标飞蛾算法的电力系统无功优化研究

李伟琨<sup>1</sup> 阙波<sup>2</sup> 王万良<sup>1</sup> 倪立洲<sup>3</sup>

(浙江工业大学计算机科学与技术学院 杭州 310023)<sup>1</sup>

(国网浙江省电力公司电力控制研究所 杭州 310000)<sup>2</sup>

(杭州天丽科技有限公司自动化控制研究所 杭州 310000)<sup>3</sup>

**摘要** 鉴于电力需求的日益增长与传统无功优化方法的桎梏,如何更加合理有效地解决电力系统的无功优化问题逐渐成为了研究的热点。提出一种多目标飞蛾扑火算法来解决电力系统多目标无功优化的问题,算法引入固定大小的外部储存机制、自适应的网格和筛选机制来有效存储和提升无功优化问题的帕累托最优解集,算法采用 CEC2009 标准多目标测试函数来进行仿真实验,并与两种经典算法进行性能的对比分析。此外,在电力系统 IEEE 30 节点上将该算法与 MOPSO,NGSGA-II 算法的求解结果进行比较分析的结果表明,多目标飞蛾算法具有良好的性能,并在解决电力系统多目标无功优化问题上具有良好的潜力。

**关键词** 多目标优化,进化算法,无功优化,飞蛾扑火,电力系统

**中图分类号** TP182 **文献标识码** A

## Multi-objective Moth-flame Optimization Algorithm Based Optimal Reactive Power Dispatch for Power System

LI Wei-kun<sup>1</sup> QUE Bo<sup>2</sup> WANG Wan-liang<sup>1</sup> NI Li-zhou<sup>3</sup>

(College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China)<sup>1</sup>

(Institute of Power Control, State Grid Zhejiang Electric Power Company, Hangzhou 310000, China)<sup>2</sup>

(Institute of Automation Control, Hangzhou Telek Technology Co., Ltd., Hangzhou 310000, China)<sup>3</sup>

**Abstract** In view of the increasing power energy demand and the drawback of conventional reactive power optimization methods, how to effectively solve the reactive power optimization has become a hot spot in power research. This paper proposed a multi-objective model of reactive power optimization problems in power system and a multi-objective moth-flame optimization algorithm (MOMFA) to optimize problems with multiple objectives for the first time. A fixed-sized external archive, grid and select mechanism are integrated to the MOMFA for maintaining and improving the pareto optimal solutions. The proposed algorithm is compared with two well-known algorithms on CEC multi-objective optimization test problems. Moreover, the proposed algorithm was simulated in real power system data and compared with two well-known algorithms: multi-objective particle swarm optimization (MOPSO) and non-dominated sorting genetic algorithm version 2 (NSGA-II). The results demonstrate that the proposed algorithm is outperforms other algorithms in reactive power optimization.

**Keywords** Multi-objective optimization, Evolutionary algorithm, Reactive power optimization, Moth-flame, Power system

## 1 引言

随着电力需求的日益增加,传统解决优化问题的方法已无法满足需求,而随着研究的不断深入,许多新兴技术被应用到电力系统优化问题的求解中,尽管如此,当解决实际的工程问题时,仍会面临许多挑战。其中,最为突出的就是电力系统的多目标无功优化问题(Optimal Reactive Power Dispatch, ORPD)。电力系统无功优化调度问题是指在电力系统无功电源较为充裕的情况下,通过调节发电机机端电压、变压器抽头变比、无功补偿装置的出力等操作来调整无功潮流,使得系

统电压值达到合格值,同时全网有功损耗最小<sup>[1]</sup>。一方面,电力系统的无功优化问题属于典型的多目标非线性复杂优化问题,传统的线性、非线性规划解决方法<sup>[2-4]</sup>对数学模型的要求较高,不利于实际操作。另一方面,电力系统无功优化作为多目标优化问题,与单目标优化所不同的是,其无法像单目标优化那样找到一个单独的最优解来满足所有的优化目标<sup>[5]</sup>,多目标优化最终得到的往往是一组解集,这组解集是多个相互冲突的目标间相互权衡取舍后的结果集合<sup>[6]</sup>。而进化算法的提出与应用较好地弥补了传统方法在解决无功优化问题上的不足。近年来,越来越多的进化算法(如遗传算法(Genetic al-

本文受国家电网公司科技项目(SGZJ0000BGJS1500460)资助。

李伟琨(1990—),男,博士,CCF 会员,主要研究方向为进化计算;阙波(1961—),男,高级工程师,主要研究方向为电力生产、调度控制;王万良(1957—),男,博士,教授,主要研究方向为人工智能,E-mail: zjutwwl@zjut.edu.cn;倪立洲(1971—),男,工程师,主要研究方向为智能自动化控制。

gorithm, GA)<sup>[7-11]</sup>、粒子群算法(Particle Swarm Optimization, PSO)<sup>[12-13]</sup>、进化规划(Evolutionary Programming, EP)、进化策略(Evolutionary Strategy, ES)<sup>[14-16]</sup>、模拟退火算法<sup>[17]</sup>等)被用来解决电力系统的无功优化问题。尽管这些算法都可以在优化问题中取得一定的优势,但仍难以避免陷入局部最优,此外,临界阈值、温度条件等参数的设定给实际的操作带来了困难,使得算法难以实际应用<sup>[18]</sup>,就像 Wolpert 等人<sup>[19]</sup>所证明的理论中阐述的一样,并不存在一种算法可以解决所有的优化问题,而这也是电力系统多目标无功优化研究领域不断创新提升的动力所在。近年来新算法不断地涌现,如水波算法(Water Wave Optimization, WWO)<sup>[20]</sup>、动态自适应差分进化算法(Dynamic Adaptive Differential Evolution Algorithm, DADE)<sup>[21]</sup>。鉴于此,本文提出一种多目标飞蛾算法(MOMFA)来解决电力系统无功优化问题,算法以单目标飞蛾算法(Moth-Flame Optimization algorithm, MFO)<sup>[22]</sup>为基础,加入了固定的存储、自适应的网格和筛选机制来有效地储存和提升电力系统无功优化问题的非支配解。

本文第2节简要介绍电力系统多目标无功优化问题模型;第3节详细阐述应用多目标飞蛾算法解决无功优化问题的方法与步骤;第4节通过与两种经典算法在标准测试函数的对比仿真,对算法的性能进行了深入分析;第5节通过IEEE 30节点真实电力系统数据对算法进行了仿真分析,并与MOPSO, NSGAI算法的结果进行对比分析;最后总结全文。

## 2 电力系统多目标无功优化模型

本节主要介绍电力系统多目标无功优化问题模型,多目标无功优化问题主要涉及有功网损、电压偏移、静态电压稳定裕3个方面,其可用如下公式表示<sup>[1,19]</sup>。

### 2.1 以有功网损耗最小为目标

$$F = \min \sum_{i,j \in N_k} G_{ij} (V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij}) \quad (1)$$

其中,  $G_{ij}$  为节点  $i$  与节点  $j$  之间的支路电导,  $V_i$  和  $V_j$  分别表示节点  $i$  和节点  $j$  的电压模值,  $\theta_{ij}$  表示节点  $i$  与节点  $j$  之间的电压相角差,  $N_k$  表示系统参与损耗计算的支路数目。

### 2.2 以电压偏移量最小为目标

$$F = \min \sum_{i=1}^{N_B} \frac{V_i - \bar{V}_i}{\Delta V_i^{\max}} \quad (2)$$

其中,  $V_i$  表示节点  $i$  的实际电压,  $N_B$  表示系统节点的总数,  $\bar{V}_i$  表示节点  $i$  的期望电压,  $\Delta V_i^{\max}$  表示电压偏差最大允许值, 其可由如下公式得到:

$$\Delta V_i^{\max} = V_i^{\max} - V_i^{\min}$$

### 2.3 以静态电压稳定裕度最大为目标

$$F = \max(\delta_{\min}) \quad (3)$$

其中,  $\delta_{\min}$  表示系统雅可比矩阵最小奇异值, 为方便计算, 将其统一处理为最小化目标, 处理后的公式如下:

$$F = \min\left(\frac{1}{\delta_{\min}}\right) \quad (4)$$

### 2.4 模型约束条件

$$P_{Gi} - P_{Di} = V_i \sum_{j=1}^{N_G} V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad (5)$$

其中,  $P_{Gi}$ ,  $P_{Di}$  分别表示节点  $i$  对应发电机的有功功率与负荷

有功功率,  $G_{ij}$ ,  $B_{ij}$  分别表示节点导纳矩阵的实部和虚部。

$$Q_{Gi} - Q_{Di} + Q_{Ci} = V_i \sum_{j=1}^{N_G} V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \quad (6)$$

其中,  $Q_{Gi}$ ,  $Q_{Di}$  分别表示节点  $i$  对应发电机的无功功率与负荷无功功率,  $Q_{Ci}$  表示无功补偿量。

$$V_G^{\min} \leq V_{Gi} \leq V_G^{\max} \quad (7)$$

其中,  $V_{Gi}$  表示发电机的端电压,  $V_G^{\min}$ ,  $V_G^{\max}$  分别表示端电压的最小值与最大值, 此外,  $i \in N_G$ ,  $N_G$  表示系统发电机节点数。

$$T_k^{\min} \leq T_k \leq T_k^{\max} \quad (8)$$

其中,  $T_k$  表示可调变压器变比,  $T_k^{\min}$ ,  $T_k^{\max}$  表示其最小值与最大值, 此外,  $k \in N_T$ ,  $N_T$  表示有载变压器数。

$$Q_{Ci}^{\min} \leq Q_{Ci} \leq Q_{Ci}^{\max} \quad (9)$$

其中,  $i \in N_C$ ,  $N_C$  表示补偿电容器数目,  $Q_{Ci}^{\min}$ ,  $Q_{Ci}^{\max}$  分别表示无功补偿量的最小值与最大值。

$$Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \quad (10)$$

其中,  $i \in N_G$ ,  $N_G$  表示系统发电机节点数,  $Q_{Gi}^{\min}$ ,  $Q_{Gi}^{\max}$  分别表示无功功率的最小值与最大值。

$$V_i^{\min} \leq V_i \leq V_i^{\max} \quad (11)$$

其中,  $i \in N_B$ ,  $V_i^{\min}$  和  $V_i^{\max}$  分别表示电压的最小值与最大值。

## 3 多目标飞蛾算法对模型求解

### 3.1 多目标飞蛾算法

飞蛾算法(MFO)是近年来新提出的一种进化算法, 其主要灵感来自于飞蛾在趋向光源时的飞行方式, 即横定向移动方式, 其特点是生物在移动时身体与光源方向总是保持一个固定角度。在该算法中, 将飞蛾的飞行过程抽象化, 使其总是保持一个固定的角度进行探索, 以便数学建模, 而多目标飞蛾算法也是在这基础之上延伸而来的。为了方便描述飞蛾与光源间的关系, 本文用下述的数学公式对其进行模拟, 其表达式如下<sup>[22]</sup>:

$$M_i = S(M_i, F_j) \quad (12)$$

其中,  $M_i$  表示第  $i$  个飞蛾,  $F_j$  表示第  $j$  团光源,  $S$  为螺旋函数。鉴于整个螺旋的起点应为飞蛾, 终点应为光源的位置, 下面给出针对算法的对数螺旋函数定义(注: 这里给定的螺旋范围不会超过搜索空间):

$$S(M_i, F_j) = D_i \cdot e^{ht} \cdot \cos(2\pi t) + F_j \quad (13)$$

其中,  $D_i$  表示第  $i$  个飞蛾与第  $j$  团光源间的距离,  $h$  是用来描述对数螺旋的常数,  $t$  是一个随机数, 其取值范围为  $[-1, 1]$  (见图1)。距离  $D$  的定义如下:

$$D = |F_j - M_i| \quad (14)$$

其中,  $M_i$  表示第  $i$  个飞蛾,  $F_j$  表示第  $j$  团光源。

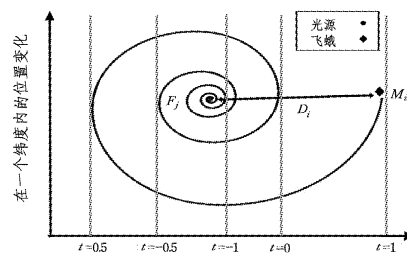


图1 MOMFA算法对数螺旋简图

由式(14)可知, 飞蛾的下一个位置取决于光源的位置, 而式(13)中参数  $t$  描述了飞蛾的下一个位置与光源的远近期

度。除了上述基础外,为了更好地解决多目标优化问题,MOMFA 加入了两个新的机制。新加入的两部分的机理与 MOPSO 非常相似。第一部分为储存机制,通过它可以有效地储存已获得的非支配解;第二部分为自适应网格和筛选机制,其主要作用为辅助算法更好地将飞蛾和光源从储存机制中筛选出来,从而将其作为横定向运动中的元素进行优化计算。

### 3.2 存储机制

算法中加入的存储机制是一个简单便捷的单元,其主要服务于算法存储和提取非支配解(帕累托最优解)。在存储机制中,控制器是其中最为重要的部分,通过控制器可以控制储存何时接受新的非支配解,以及存储空间何时达到饱和状态。在为存储设定最大成员数后,所有非支配解将在不断的迭代中与存储中的非支配解进行比较,具体的情况可分为以下 3 类:

1) 存储中至少存在一个解支配着即将进入存储的新解,若满足上述情况,则这个新解将不会被加入到存储中。

2) 存储中至少存在一个解被即将加入存储的新解所支配,若满足上述情况,则这个新解将被加入到存储中。

3) 存储中的解与即将加入存储的新解之间并不存在任何支配关系,若满足上述情况,则这个新解将被加入到存储中。

### 3.3 自适应网格与筛选机制

为了获得均匀分布的帕累托前端,MOMFA 算法采用了文献[23]中的自适应网格机制。自适应网格在真实空间中是由超立方体所组成的立体空间,由于存储设定了最大成员数,当存储达到饱和时,需要剔除部分解来为即将加入的新解释放空间,存储中多样性较差的解将被剔除。如果新加入的解超出了原有超立方体范围,那么网格会自动延伸将新解包含在范围之内,并更新所有机制。在 MFO 算法中,已获得的最优解将不断引领搜索因子向可能区域搜索,以获得全局最优解。而在 MOMFA 中,算法无法像单目标算法那样轻易比较优劣。但由于算法的储存机制中存储了已获得的非支配解,筛选机制只要从中选出多样性最好的解并把它作为飞蛾算子带入计算即可。在 MOMFA 中,采用轮赌盘的形式来进行筛选,每一个超立方体的筛选几率由如下公式定义:

$$P_i = \frac{\alpha}{N_i} \quad (15)$$

其中, $\alpha$  是一个比 1 大的常数, $N_i$  表示第  $i$  部分中已获得非支配解的数目。

采用自适应网格机制的最大优势便是计算复杂度的降低,MOMFA 算法的计算复杂度为  $O(MN^2)$ ,其中  $N$  表示个体的数量, $M$  表示优化目标的个数。

总体来说,MOMFA 算法的复杂度与 NSGA-II, MOPSO, SPEA2, PAES 算法相持平,相较 NSGA 算法与 PSEA 算法的复杂度  $O(MN^3)$  更胜一筹。MOMFA 算法集成了 MFO 算法的优秀特质,并加入了新的机制使得其在处理多目优化问题时依然具有良好的性能,算法的基础框架如算法 1 所示。

#### 算法 1 The MOMFA Algorithm

1. Initialize the population and the parameters
2. Calculate the objective values of power model for each search agent
3. Find the non-dominated solutions and initialized the archive

4. Initialize the Moth with the selection mechanism(Equ. 15)
5. While stop criterion is not satisfied do
6. for each search agent
7. Update the position by Equ. 12, Equ. 13, Equ. 14
8. end for
9. Update the parameters
10. Calculate the objective values of power model for each search agent
11. Find the non-dominated solutions
12. Update the archive with the non-dominated solutions obtained so far
13. if archive is full
14. Run the grid and select mechanism to omit
15. Add a new solutions to the archive
16. end if
17. if any new solution outside the hypercube
18. Update the grids to cover the new solutions
19. end if
20. Update the Moth with the selection mechanism
21. end While
22. return archive

## 4 算法性能分析

### 4.1 实验设置

由于 CEC 测试函数提供了可靠的测试平台并包含多种搜索空间,如凸与非凸、连续与非连续等,本节采用 CEC2009<sup>[24]</sup>作为算法的测试函数。此外,为直观地评估算法的性能,选取两种经典算法(MOPSO 与 MOEA/D)作为对比算法。其中 MOPSO 与 MOEA/D 算法的版本分别来自原作者个人主页直接下载版本,MOMFA 算法为作者自行编写。在评估指标方面,本节采用 3 种指标来进行评估,分别为反转世代距离(Inverted Generational Distance, IGD)<sup>[25]</sup>、空间评价指标(Spacing, SP)<sup>[26]</sup>、最大传播距离(Maximum spread, MS)<sup>[27]</sup>。3 种指标的定义如下:

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (16)$$

其中, $n$  表示真实帕累托最优解的个数, $d_i$  表示真实帕累托最优解与其最邻近的帕累托最优解间的欧氏距离。

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (17)$$

其中, $\bar{d}$  为  $d_i$  的平均值, $n$  表示所获的帕累托最优解的个数,对于所有  $i, j = 1, 2, 3, \dots, n, d_i$  满足以下公式: $d_i = \min_j (|f_i(\vec{x}) - f_i^*(\vec{x})| + |f_j^*(\vec{x}) - f_j(\vec{x})|)$ 。

$$MS = \sqrt{\sum_{i=1}^o \max(d(a_i, b_i))} \quad (18)$$

其中, $a_i$  为第  $i$  个目标中的最大值, $b_i$  为第  $i$  个目标中的最小值, $d$  为计算欧氏距离的函数, $o$  为算法中优化目标的个数。对于 3 种指标,其中 IGD 指标的值越小,算法优化后的解越趋近真实帕累托前端。SP 指标则反映了解的分布情况,SP 的值越小,算法优化后的解在帕累托真实前端的分布越均匀。而 MS 指标的值越大,算法的延展性越好。

4.2 实验结果

将每一个测试函数运行 30 次并对结果进行统计分析,其分析结果如表 1—表 3 所列。算法设定种群为 500,最大迭代次数为 300,为增加对比,算法加入了各个测试函数的真实帕累托前端,其对比结果如图 2—图 8 所示。

表 1 测试函数在 IGD 指标下的统计结果

IGD	标准	MOMFA	MOPSO	MOEA/D
UF1	best	0.04011	0.14595	0.10808
	worst	0.0685	0.29348	0.15622
	median	0.05404	0.26499	0.12837
	std	<b>0.013662163</b>	0.056225513	0.020723113
UF2	best	0.03395	0.07101	0.04996
	worst	0.04749	0.0835	0.14233
	median	0.03904	0.07522	0.08775
	std	0.004529552	<b>0.00408366</b>	0.20500075
UF3	best	0.10241	0.30308	0.26487
	worst	0.39376	0.71023	<b>0.2876</b>
	median	0.16538	0.28752	0.26964
	std	0.035849064	0.172163477	<b>0.008524501</b>
UF4	best	0.04933	0.07906	0.07802
	worst	0.06194	0.10158	0.08178
	median	0.05216	0.08352	0.08144
	std	0.004287307	0.008108421	<b>0.001596466</b>
UF5	best	0.27011	0.061645	0.39884
	worst	0.47521	2.66103	1.25441
	median	0.29501	2.23917	1.06831
	std	<b>0.074829469</b>	0.837220491	0.293602212
UF6	best	0.36323	0.46641	0.34138
	worst	0.5586	1.20378	0.54647
	median	0.41573	0.79299	0.46691
	std	0.083411598	0.290239507	<b>0.070833556</b>
UF7	best	0.03133	0.08571	0.07859
	worst	0.07568	0.17373	0.42122
	median	0.06031	0.10957	0.39249
	std	<b>0.016995574</b>	0.032301106	0.162555152

表 2 测试函数在 MS 指标下的统计结果

IGD	标准	MOMFA	MOPSO	MOEA/D
UF1	best	1	0.9867	0.9861
	worst	0.9823	0.9374	0.6572
	median	0.99873	0.9672	0.7859
	std	0.006947775	0.016051991	0.105840797
UF2	best	1	0.99	0.9686
	worst	0.84332	<b>0.9765</b>	0.7905
	median	0.9998	0.9808	0.8888
	std	0.064002387	<b>0.005087789</b>	0.061781324
UF3	best	1	1	0.6165
	worst	0.87723	0.78843	0.4688
	median	0.94982	0.86653	0.5756
	std	0.05790837	0.087180749	<b>0.054695397</b>
UF4	best	0.998	0.9832	0.9763
	worst	0.9873	0.9664	0.9638
	median	0.9918	0.9723	0.9706
	std	<b>0.003576031</b>	0.005852179	0.00444
UF5	best	1	0.645	0.6591
	worst	0.76612	0.5014	0.2942
	median	0.992673	0.56	0.3942
	std	0.159325104	<b>0.054179051</b>	0.113155811
UF6	best	0.9994	0.8352	0.7823
	worst	0.9712	0.3758	0.3676
	median	0.9788	0.5774	0.6173
	std	<b>0.011081588</b>	0.172607446	0.148279115
UF7	best	1	0.9918	0.9761
	worst	0.9998	0.9531	0.3229
	median	0.99873	0.9615	0.4193
	std	0.006947775	0.015440855	0.268859833

表 3 测试函数在 SP 指标下的统计结果

IGD	标准	MOMFA	MOPSO	MOEA/D
UF1	best	0.014896	0.013513	<b>0.000183</b>
	worst	0.023027	0.052491	<b>0.014615</b>
	median	0.018632	0.025096	<b>0.011035</b>
	std	<b>0.0027666</b>	0.0142941	0.0061849
UF2	best	0.017587	0.007405	<b>0.006615</b>
	worst	<b>0.028537</b>	0.079123	0.042391
	median	0.025439	0.016851	0.013062
	std	<b>0.0040856</b>	0.0269323	0.0137057
UF3	best	0.045976	0.010710	<b>0.002040</b>
	worst	0.090849	0.071515	<b>0.017521</b>
	median	0.051634	0.013583	<b>0.010055</b>
	std	0.0194194	0.023919801	<b>0.0066568</b>
UF4	best	0.016744	0.006903	<b>0.006765</b>
	worst	0.028997	0.023706	<b>0.025734</b>
	median	0.020745	0.010221	<b>0.010906</b>
	std	<b>0.0044944</b>	0.0064592	0.0068380
UF5	best	0.026636	0.014198	<b>0.007295</b>
	worst	<b>0.042418</b>	0.169397	0.070885
	median	0.027523	<b>0.017918</b>	0.034851
	std	<b>0.0064346</b>	0.0610232	0.0284408
UF6	best	0.028052	0.005583	<b>0.005087</b>
	worst	<b>0.034789</b>	0.108273	0.102779
	median	0.031169	0.077375	0.018143
	std	<b>0.0025537</b>	0.0457718	0.0366772
UF7	best	0.019042	0.014155	<b>0.0001897</b>
	worst	<b>0.032185</b>	0.044382	0.1027794
	median	0.023194	0.035119	0.0107172
	std	<b>0.0050454</b>	0.01159870	0.03813855

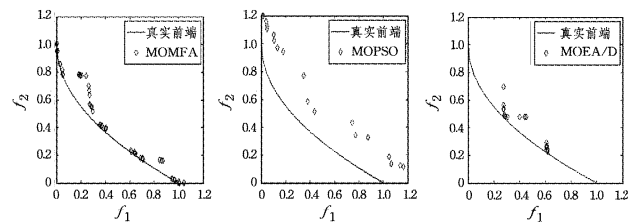


图 2 MOMFA, MOPSO, MOEA/D 在 UF1 下与真实前端的对比图

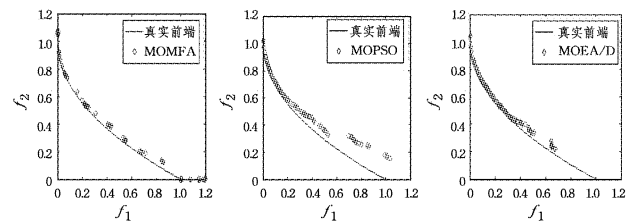


图 3 MOMFA, MOPSO, MOEA/D 在 UF2 下与真实前端的对比图

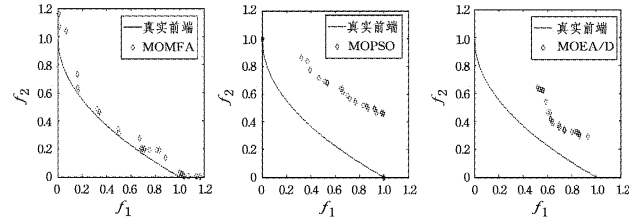


图 4 MOMFA, MOPSO, MOEA/D 在 UF3 下与真实前端的对比图

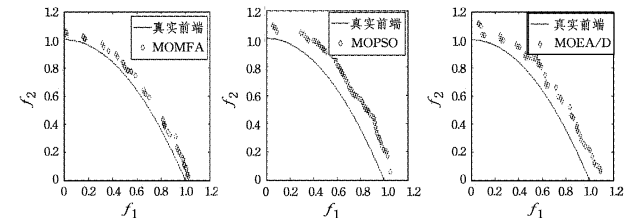


图 5 MOMFA, MOPSO, MOEA/D 在 UF4 下与真实前端的对比图

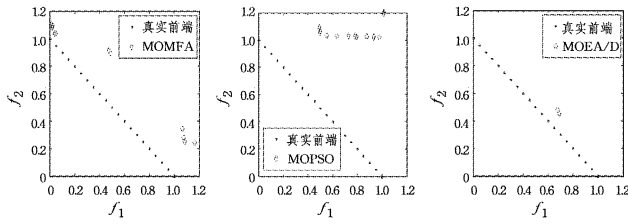


图 6 MOMFA, MOPSO, MOEA/D 在 UF5 下与真实前端的对比图

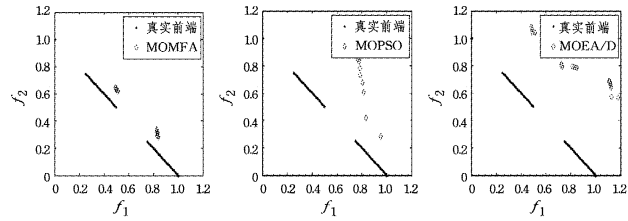


图 7 MOMFA, MOPSO, MOEA/D 在 UF6 下与真实前端的对比图

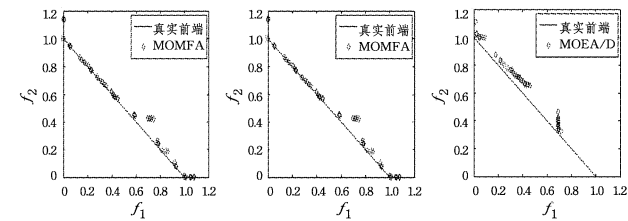


图 8 MOMFA, MOPSO, MOEA/D 在 UF7 下与真实前端的对比图

由于 CEC 测试函数提供了可靠的测试平台并包含多种搜索空间,如凸与非凸、连续与非连续等,本节采用 CEC2009 作为算法的测试函数。此外,为直观地评估算法的性能,本节选取了两种经典算法(MOPSO 与 MOEA/D)作为对比算法。

表 1 为 3 种算法运行测试函数后在 IGD 指标下的统计结果,从表 1 中可以清晰地看到 MOMFA 在 UF1 测试函数下的性能良好。此外,对比 MOPSO 和 MOEA/D 算法, MOMFA 的非支配解在真实帕累托前端的覆盖程度上也较为优秀,尽管 MOEA/D 在 SP 指标上占优,但 MOMFA 算法分布的稳定性更胜一筹。3 种算法在 UF1 测试函数下与真实帕累托前端的对比图如图 2 所示,图 2 中也显示出 MOMFA 更加靠近真实帕累托前端。

如图 3 所示, MOMFA 在 UF2 测试函数下的性能依然良好,其中 IGD 指标下的最优、最差、中值都为 MOMFA 所获的。尽管在表 1 中, IGD 指标的标准值最优为 MOPSO 所获得,但 MOMFA 在真实前沿的覆盖率以及非支配解的分布均匀度上比其他两种算法略胜一筹(见表 2、表 3)。此外,虽然 UF2 测试函数在 SP 指标下的最优值由 MOEA/D 所获得,但其波动性较大,而 MOMFA 在 SP 指标下的稳定性优于其他两种算法。

从 MOMFA, MOPSO, MOEA/D 在 3 种指标下的统计结果可以看到, MOMFA 在 UF3 测试函数下的性能表现较为良好。尽管其在 IGD 指标下的稳定性比 MOEA/D 略逊一筹,但 MOMFA 在真实帕累托前端的覆盖程度远胜于 MOEA/D。此外,图 4 同样显示出 MOMFA 在 UF3 测试函数上的优良性能,但相比于 MOEA/D,其在分布均匀度上仍有不足。

而对于 UF4 测试函数, MOMFA 的表现远超 MOPSO 和 MOEA/D,从表 1 可以看到, IGD 指标下的最优、最差、中值最优都为 MOMFA 所取得,这证明了其在 UF4 上具有良好

的收敛性,其所获的非支配解更加贴近帕累托真实前端。而图 5 中 3 幅对比图也验证了表 1 中的统计结果, MOMFA 在真实前沿的覆盖程度上较 MOPSO 与 MOEA/D 更为良好,尽管在分布均匀度上的最优值为 MOEA/D 取得,但 MOMFA 的分布稳定性更好,波动较小(见表 2、表 3)。

与上述段落描述的测试函数有所不同的是, UF5 测试函数的真实帕累托前端是不连续的,图 6 中的 3 幅算法对比图也清晰地显示了这一点。但 MOMFA 仍然表现出良好的收敛性,在 IGD 指标上,所有最优值都由 MOMFA 获得,其非支配解与真实帕累托前端的距离比 MOPSO 与 MOEA/D 更为接近。另一方面, MOMFA 在帕累托真实前沿的覆盖程度上依然表现出色。尽管在 SP 指标的最小值为 MOEA/D 所取得,但 MOMFA 的 SP 指标数据波动性更小,从图 6 可以看出其在分布均匀程度上依然具有良好的表现。

对于 UF6 测试函数,由图 7 可以看到,其真实帕累托前端仍然包含不连续的部分。MOMFA 在 UF6 测试函数中依然保持了 UF5 中良好的收敛性,由表 1 可以看到, IGD 指标下的最优、最差、中值的最佳数据都由 MOMFA 取得,尽管在数据波动性上略输于 MOEA/D,但 MOMFA 算法的真实前沿覆盖程度远胜 MOPSO 与 MOEA/D,其中 MS 指标表现的所有最优数据都为 MOMFA 所得。此外,其分布均匀程度虽在最优值上略逊,但其波动较小,数据分布均匀,而从图 7 也可以看出 MOMFA 在 UF6 测试函数上的表现依然良好。

由图 8 可以看到, UF7 测试函数具有线性帕累托真实前端,相比于具有凸与非凸帕累托真实前端的测试函数,算法在 UF7 测试问题上更容易趋向真实帕累托前端。从图 8 可以看出,3 种算法在 UF7 测试函数上表现优良,但从 3 种指标的统计结果来看, MOMFA 更加靠近帕累托真实前端,其具有更好的收敛性。而在真实前沿覆盖程度上, MOMFA 依然比其他两种算法表现出众,其中 MS 指标下的 4 种标准最优值都由 MOMFA 获得。而在 SP 指标下,尽管最优值并未在 MOMFA 中,但其最大值比其他两种算法都小,且值的稳定性较好,可见其分布均匀程度依然很好。

通过实验的结果分析可以看到, MOMFA 总体上具有良好的性能。从测试函数在 3 个指标下的统计结果来看,相比于 MOPSO, MOMFA 整体上较优。这得益于 MOMFA 算法集成的固定存储机制与自适应的网格及筛选机制,尽管其与 MOPSO 类似,但是 MOPSO 在每一次迭代时更新  $gBest$ ,而 MOMFA 则在每次算子更新位置时更新,这使得 MOMFA 的性能得到提升。而对于 MOEA/D,从实验的数据与图像结果可以看出,其在线性真实帕累托前端的情况下具有良好的性能,而相比于 MOMFA, MOEA/D 的真实前沿覆盖情况与收敛情况都稍逊。这主要是由于 MOEA/D 自身是基于分解的进化算法,其针对线性前端具有良好的性能。

## 5 算例验证

### 5.1 算例数据

本文采用 IEEE 30 节点作为无功优化仿真测试的平台,各并联电容器无功补偿容量的上限为 50 Mvar,下限为 0 Mvar,调节步长为 1 Mvar;变压器变比的上限为 1.1,下限为 0.9,调节步长为 0.0125<sup>[28-29]</sup>。初始条件下,各节点无功功率

补偿容量设置为0。各发电机端电压为1.0p.u.,变压器变比为1.0。系统总的负荷量为 $P_{load}=284.3\text{MW}$ , $Q_{load}=126.2\text{Mvar}$ ,系统总损耗 $P_{loss}=20.88\text{MW}$ , $Q_{loss}=81.02\text{Mvar}$ <sup>[30]</sup>。算法程序采用MATLAB编制,其具体参数设置如下:种群规模为300,最大迭代次数为500。

## 5.2 优化结果

本节以第2节中建立的模型为基础,采用多目标飞蛾算法来处理电力系统多目标无功优化问题,此外还加入了目前已应用在电力系统的MOPSO算法与NSGAI算法的处理结果作为对比,3种算法处理无功优化问题所获得的非支配解的情况如图9—图11所示。

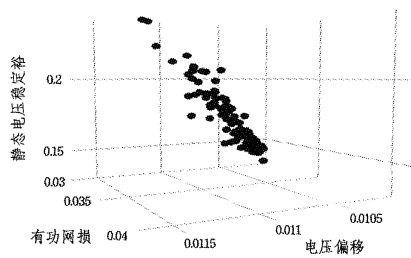


图9 MOMFA处理无功优化问题的结果图

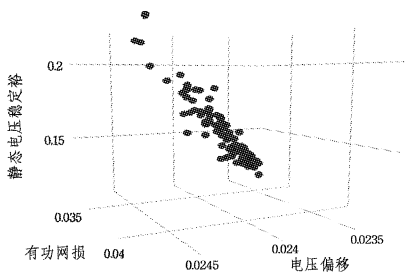


图10 NSGAI算法处理无功优化问题的结果图

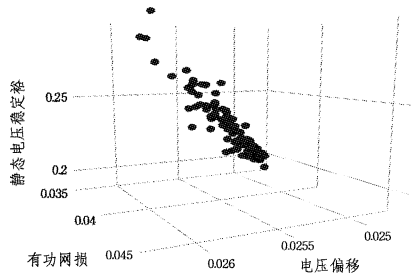


图11 MOPSO算法处理无功优化问题的结果图

对于多目标优化,其最大特征在于无法像单目标问题那样找到唯一最优解。在多目标优化中,由于多个目标间的相互冲突与制约,算法最终将得到一组权衡各个目标后的解集来表征多目标优化处理结果。在图9—图11中,圆点代表3个算法在多目标无功优化问题下所得的非支配解。通过对比可以看出,MOMFA在无功优化问题上表现优异,特别是在有功网损、电压偏移两个目标上所得的优化结果明显比NSGAI,MOPSO优秀。而从3个算法所得结果的统计情况来看(见表4),尽管NSGAI与MOPSO在有功网损与电压偏移两个目标上表现良好,且MOPSO在有功网损目标上波动较小,但MOMFA的统计结果仍然略胜一筹,其中最优、最差、中值的最优都由MOMFA获得。而在静态电压稳定裕目标上,统计结果采用 $\delta_{\min}$ 值作为评价指标,其值越大,结果越优

(同理, $\frac{1}{\delta_{\min}}$ 越小,结果越优)。从表4可以看到,MOPSO较MOMFA,NSGAI表现较优,其最优值、最差值、中值的最佳结果都为MOPSO所获得,但MOMFA结果的波动性较小,且统计结果较NSGAI仍然较优。此外,从不同层面来看,采用MOMFA的无功优化求解方式可以提供较其他两种算法更优的结果,如在考虑同电压偏移时,MOMFA所提供的优化方案中网损更小,而如果从经济层面考虑,在所需的静态电压稳定裕度值的基础上,MOMFA的优化方案同样可提供较低的网损。综上所述,从电力系统无功优化的整体情况来看,MOMFA在电力系统无功优化问题上的表现良好并具有更好的潜力。

表4 3种算法处理多目标无功优化问题的统计结果/p.u.

目标	标准	MOMFA	NSGAI	MOPSO
有功网损	最好	0.03196	0.03522	0.03525
	最差	0.05160	0.06565	0.04565
	中值	0.03928	0.04565	0.03825
	标准差	0.003600	0.007548	0.006432
电压偏移	最好	0.01123	0.02306	0.02506
	最差	0.02169	0.03459	0.03059
	中值	0.01446	0.02906	0.02705
	标准值	0.008866	0.007201	0.006629
静态电压稳定裕	最好	0.26602	0.02332	0.02932
	最差	0.1302	0.01196	0.01206
	中值	0.19068	0.01664	0.02364
	标准值	0.012111	0.008808	0.0199242

**结束语** 随着研究的深入,电力系统的多目标无功优化逐渐成为了电力系统研究领域的热点,本文以有功网损最小化、电压偏移最小化与静态电压稳定裕最大化为目标,提出一种全新的多目标飞蛾算法来解决电力系统多目标无功优化问题,算法以飞蛾趋向光源过程中的路径探索与规划为基础,其简捷却不失高效。在本文的工作中,固定的存储机制与自适应的网格和筛选机制被集成在算法中,使得算法可以高效地存储和筛选非支配解。此外,算法采用了CEC2009的测试函数进行仿真测试,并与MOPSO,MOEA/D两种经典算法进行了对比分析,结果显示MOMFA算法具有良好的性能。此外,通过与两种应用于无功优化问题的算法MOPSO,NSGAI在IEEE33节点上进行对比分析,证明了多目标飞蛾算法在处理电力系统多目标无功优化问题上具有良好的性能与潜力。尽管如此,MOMFA算法仍有不足,目前该算法只能解决低纬度的多目标问题,如果问题纬度上升或目标数较大,其固定存储会因容易获得非支配解而快速趋于饱和,最终导致算法运行后无法获得理想的结果。接下来将针对无功优化的负荷模型等问题进行进一步研究与分析。

## 参考文献

- [1] 张勇军,任震,李邦峰. 电力系统无功优化调度研究综述[J]. 电网技术,2005,29(2):50-56.
- [2] SASSON A M. Combined Use of the Powell and Fletcher-Powell Nonlinear Programming Methods for Optimal Load Flows[J]. IEEE Transactions on Power Apparatus & Systems,1969,PAS-88(10):1530-1537.
- [3] 罗铸. 电力系统无功和电压的优化调节[J]. 电力系统自动化,1988,12(1):11-17.

- [4] 程莹,刘明波. 求解离散无功优化的非线性原-对偶内点算法[J]. 电力系统自动化,2001,25(9):23-27.
- [5] COELLO C A C. Evolutionary multi-objective optimization: a historical view of the field[J]. Computational Intelligence Magazine, IEEE, 2006, 1(1): 28-36.
- [6] COELLO C A C, VAN VELDHUIZEN D A, LAMONT G B. Evolutionary algorithms for solving multi-objective problems [M]//New York: Kluwer Academic. 2002.
- [7] IBA K. Reactive power optimization by genetic algorithm m[J]. IEEE Transactions on Power Systems, 1994, 9(2): 685-692.
- [8] 张勇军,任震,钟红梅,等. 实时无功优化调度中的邻域搜索改进遗传算法[J]. 电网技术,2003,27(1):22-25.
- [9] 张勇军,任震,钟红梅,等. 基于突变遗传算法的无功规划优化[J]. 电力系统自动化,2002,26(23):29-32.
- [10] 赵登福,周文华,张伏生,等. 遗传算法在无功优化应用中的改进[J]. 电网技术,1998,22(10):34-36.
- [11] 王志华,尹项根,李光熹. 伪并行遗传算法在无功优化中的应用[J]. 电网技术,2003,27(8):33-35.
- [12] 周晖,周任军,谈顺涛,等. 用于无功电压综合控制的改进粒子群优化算法[J]. 电网技术,2004,28(13):45-49.
- [13] 吴秀华,朴在林,徐静,等. 基于改进粒子群优化算法的电力系统无功电压综合控制[J]. 电力系统保护与控制,2007,35(21):28-33.
- [14] WU Q H, MA J T. Power system optimal reactive power dispatch using evolutionary programming[J]. IEEE Transactions on Power Systems, 1995, 10(3): 1243-1249.
- [15] GOMES J R, SAAVEDRA O R. Optimal reactive power dispatch using evolutionary computation: extended algorithms[J]. IET Proceedings-Generation Transmission and Distribution, 1999, 146(6): 586-592.
- [16] PRADHAN P M, PANDA G. Solving multi-objective problems using cat swarm optimization [J]. Expert Systems with Applications, 2012, 39(3): 2956-2964.
- [17] 顾丹珍,徐瑞德. 一种地区电网多目标无功优化的新方法—改进模拟退火算法[J]. 电网技术,1998,22(1):71-74.
- [18] 王克文,张东岳. 电力系统无功优化算法综述[J]. 电测与仪表,2016,53(10):73-76.
- [19] WOLPERT D H, MACREARY W G. No free lunch theorems for optimization[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82.
- [20] 张蓓,郑宇军. 水波优化算法收敛性分析[J]. 计算机科学,2016,43(4):41-44.
- [21] 李章维,周晓根,张贵军. 一种动态自适应差分进化算法[J]. 计算机科学,2015,42(6A):52-56,74
- [22] MIRJALILI S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, 2015, 89: 228-249.
- [23] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256-279.
- [24] ZHANG Q, ZHOU A, ZHAO S, et al. Multi-objective optimization test instances for the CEC 2009 special session and competition[OL]. <http://dces.essex.ac.uk/staff/qzhang/MOEAcompetition/cec09testproblem0904.pdf>.
- [25] SIERRA M R, COELLO C A C. Improving PSO-based multiobjective optimization using crowding, mutation and  $\epsilon$ -dominance [M]//Evolutionary multi-criterion optimization. Springer Berlin Heidelberg, 2005: 505-519.
- [26] VAN VELDHUIZEN D A, LAMONT G B. On measuring multi-objective evolutionary algorithm performance[C]//Proceedings of the 2000 Congress on Evolutionary Computation. IEEE, 2000, 1: 204-211.
- [27] ZITZLER E, DEB K, THIELE L. Comparison of multiobjective evolutionary algorithms: Empirical results [J]. Evolutionary computation, 2000, 8(2): 173-195.
- [28] KACHORE P, PALANDURKAR M V. TTC and CBM Calculation of IEEE-30 Bus System[C]//International Conference on Emerging Trends in Engineering & Technology, Ictet 2009. Nagpur, Maharashtra, India, 2009: 539-542.
- [29] EISSA M M, ABDEL-HAMEED T S, GABBAR H. A novel approach for optimum number and location of FACTS devices on IEEE-30 bus grid using meta-heuristic based Harmony Search [C]//IEEE International Conference on Smart Energy Grid Engineering. 2013: 1-10.
- [30] 陈前宇,陈维荣,戴朝华. 电力系统无功优化多目标处理与算法改进[J]. 电力系统保护与控制,2014,42(5):129-135.

(上接第 479 页)

- [3] LIU L Y, WANG J F, LEE, et al. Design and hardware architectures for dynamic Huffman coding [J]. IEE Proceedings: Computers and Digital Techniques, 1995, 142(6): 411-418.
- [4] 张新超. 基于 FPGA 的 Huffman 编码并行实现及高速存储系统设计[D]. 西安:长安大学,2015.
- [5] 郭诚欣,陈红,孙辉,等. 基于现代硬件的并行内存排序方法综述[J]. 计算机学报,2017,40(9):2070-2092.
- [6] BATCHER K E. Sorting networks and their applications [C]//AFIPS '68. 1968: 307-314.
- [7] 高劲松,孟令奎. 硬件排序器研究进展[J]. 电子计算机与外部设备,1997(5):38-40.
- [8] KOCH D, TORRESEN J. FPGASort: A high performance sorting architecture exploiting run-time reconfiguration on FPGAs for large problem sorting [C]//Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays. 2011: 45-54.
- [9] 倪泽峰,王振华,谭毅华,等. 并行哈夫曼编码器的硬件设计与实现[J]. 微电子学与计算机,2002,19(10):66-68.
- [10] HE T, XLI T, LI Y J, et al. The Analysis and Improvement of Huffman Algorithm[C]//Proceedings of 2011 4th IEEE International Conference on Computer Science and Information Technology (ICCSIT 2011). 2011: 529-532.
- [11] TAEYEON L, JAEHONG P. Design and implementation for static Huffman encoding hardware with parallel shifting algorithm[C]//IEEE Nuclear Science Symposium Conference Record. 2003: 1315-1318.