

基于改进差分进化鲸鱼算法的经济负荷分配

马 健,李海明,李 鑫

(上海电力大学,上海 201306)

摘 要:针对电力系统经济负荷分配这一典型的非凸、非线性、组合优化问题,提出一种将改进差分进化算法和鲸鱼算法相结合的优化算法。该算法首先在鲸鱼优化算法中引入了非线性的收敛变化策略,加速寻优算法的迭代;再利用差分进化算法的交叉和选择,丰富算法种群个体信息,增强优化算法的全局收敛性;同时引入一种淘汰机制,将适应度较好的个体信息更快地保留用于下一次鲸鱼优化算法的迭代,提高了求最优解的速度和精度;最后,对多个经济负荷分配问题进行了测试,将该算法与标准鲸鱼算法、标准差分进化算法进行对比,验证了差分进化鲸鱼算法可以更合理地配置电力系统的经济负荷,能够有效找到可行解,避免陷入局部最优,能实现经济负荷的合理分配。

关键词:鲸鱼优化算法;差分进化算法;自适应策略;淘汰机制;经济负荷分配

中图分类号:TP301;TM73

文献标识码:A

文章编号:1673-629X(2022)03-0132-07

doi:10.3969/j.issn.1673-629X.2022.03.022

Economic Load Distribution Based on Improved Differential Evolution Whale Algorithm

MA Jian, LI Hai-ming, LI Xin

(Shanghai Electric Power University, Shanghai 201306, China)

Abstract: For the typical nonconvex, nonlinear, combinatorial optimization problem of economic load distribution in power system, an optimization algorithm combining improved differential evolutionary algorithm and whale algorithm is proposed. The algorithm first introduces a nonlinear convergence change strategy in the whale optimization algorithm to accelerate the iteration of the optimization algorithm, and then uses the crossover and selection of the differential evolution algorithm to enrich the individual information of the algorithm population and enhance the global convergence of the optimization algorithm. At the same time, it introduces an elimination mechanism to retain the information of individuals with better adaptability faster for the next iteration of the whale optimization algorithm, which improves the speed and accuracy of solving the optimal solution. Finally, the improved algorithm was tested on several economic load allocation problems, and the results were compared with the standard whale algorithm and the standard differential evolutionary algorithm to verify that the differential evolutionary whale algorithm can allocate the economic load of the power system more rationally and can effectively find feasible solutions to avoid falling into the local optimum, and can realize the reasonable distribution of economic load.

Key words: whale optimization algorithm; differential evolution algorithm; adaptive strategy; elimination mechanism; economic load distribution

1 概 述

电力经济负荷分配(economic load dispatch, ELD)指的是对某一区域配电网中的所有机组进行负荷的经济型分配,使得在成本最小的情况下利用现有资源可靠地满足用户的需求,以获得最大经济效益。

由于发电机组的阀点效应、配电网输电性能差异的限制以及功率平衡的和机组运行条件的约束,在分配机组负荷时往往使得最优解的求解区域无法拓宽。

早期通常选择拉格朗日法、动态规划法等数学常规优化方法处理该类负荷分配问题^[1]。除此之外,当配网系统中出现较多机组时,发电机组两端表现出的高维特性是基本数学优化方法所无法快速求解的。

近些年来,许多学者提出使用群智能算法来应对ELD问题中的难点。群智能算法对计算中目标函数的种类、计算数据的不确定性和搜索空间的结构具有很强的适应性,能够处理电网损耗、阀点效应、燃料差

收稿日期:2021-03-31

修回日期:2021-07-31

基金项目:国家自然科学基金项目(61772327)

作者简介:马 健(1996-),男,硕士研究生,研究方向为负荷分配、智能优化算法;李海明,教授,硕导,研究方向为智能信号处理的基础和应用研究。

异等因素带来的影响,使得求解结果更加符合工程实际。目前,粒子群算法(particle swarm optimization, PSO)^[2]、遗传算法(genetic algorithm, GA)^[3]、簇类进化算法(cluster evolutionary algorithm, CEA)^[4]、差分进化算法(differential evolution, DE)^[5]等在最优化领域均取得了不错的成果。

陈皓等在文章中提出了一种基于簇类搜索驱动的群体进化算法,该算法提出通过聚类对进化个体构建具有相互关联的簇类化组织,实现了对计算过程进行有目标的调整和优化^[6]。Rahmat N等提出利用差分进化免疫蚁群优化算法解决具有阀点效应的ELD问题。该算法将原始蚁群优化算法通过差分进化和人工免疫算法的突变、交叉、选择和克隆过程来改善原始算法的计算速度和防止计算进入停滞^[4]。Sen T等提出了三种元启发式算法的混合,蚁群优化(ant colony optimization, ACO)、人工蜂群算法(artificial bee colony, ABC)和谐波搜索(harmony search, HS)。寻找初始解集的任务由ACO算法处理。ABC算法检查并增强了由ACO算法生成的解,而HS算法从解集中删除了平庸的解,并用更高质量的解替换了它们^[7]。Chen X提出了一种通过双种群自适应DE算法(dual-population adaptive differential evolution, DPADDE)处理考虑阀点效应的大规模复合燃料ELD问题的方案^[8]。

鲸鱼优化算法(whale optimization algorithm, WOA)^[9]在近些年已成功应用于水库群供水^[10]、断路器的生产优化^[11]、鲁棒多用户检测^[12]、微网复合储能系统容量优化配置^[13]、电网无功优化调度^[14]等方面。WOA算法虽然结构简单,寻优方式新颖独特,但同其他传统的优化算法一样,在搜寻速度、搜寻最优解的精度方面仍不理想,也难以摆脱局部最优的困境,且在寻优的后期易出现搜索疲软的特征。DE算法是一种全局收敛性良好,运算操作步骤简洁的典型算法,但该算法初始种群的选取具有不稳定性,参数的选择不当将使算法在初期就陷入局部最优。

为此,考虑将鲸鱼优化和差分进化算法分别改进,然后选取两种算法各自的优点,合并成一种改进的差分进化鲸鱼优化算法(improved differential evolution whale optimization algorithm, IDEWOA)。首先使用WOA算法中的捕食和气泡网策略来代替DE中的变异的步骤,更新种群信息。然后将算法中的收敛因子调整为特殊的非线性更新策略以增强算法的探索能力。最后引入差分进化算法的交叉和选择环节增强种群的多样性,防止在求最优解的过程中收敛精度的过早下降。同时在交叉环节采用淘汰机制,提高了算法的寻优效率,从而证明了该算法在求解ELD问题中的优越性。

在上述研究的基础上,该文选取13机组和40机组的电力系统负荷,用改进的新算法对其进行优化分配的仿真测试,最后将该算法与标准鲸鱼优化算法、标准差分进化算法等进行对比分析,验证了该优化算法的优越性。

2 经济负荷分配模型

2.1 目标函数

电力系统经济负荷分配模型中,目标函数的解实际为整个电力系统机组发电费用的最小值:

$$\min F = \sum_{i=1}^m F_i(P_i) = \sum_{i=1}^m a_i P_i^2 + b_i P_i + c_i \quad (1)$$

式中, F 为机组发电所需的耗费; $F_i(P_i)$ 为机组 i 的费用特征函数; P_i 为机组 i 的运行功率; a_i , b_i , c_i 为耗量特征参数。

2.2 阀点效应

阀点效应将引起发电机组的耗量曲线上的不连续现象,其主要原因是机组气阀在运行过程中突然启动造成的拔丝现象。将阀点效应纳入计算中有利于提高负荷分配问题最优解的精度,故引入阀点效应之后的数学模型为:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + E_i \quad (2)$$

$$E_i = |d_i \sin(e_i(P_i^{\min} - P_i))| \quad (3)$$

式中, E_i 为第 i 台机组的阀点效应造成的费用影响; d_i 、 e_i 为机组特征常数; P_i^{\min} 为第 i 台发电机组的运行功率最小值。

2.3 约束条件

2.3.1 功率平衡约束

$$\sum_{i=1}^m P_i = P_D + P_L, i = 1, 2, \dots, m \quad (4)$$

其中, m 为系统中发电机组之和; P_D 为电力系统中的负荷目标值; P_L 为电网的损耗值。在一定配电网范围内,当系统内负荷集中分布且覆盖面较小时,其网络损失可忽略不计,故将式(4)调整为如下形式:

$$\sum_{i=1}^m P_i = P_D, i = 1, 2, \dots, m \quad (5)$$

2.3.2 机组运行约束

$$P_i^{\min} \leq P_i \leq P_i^{\max}, i = 1, 2, \dots, m \quad (6)$$

其中, P_i^{\max} 和 P_i^{\min} 分别表示第 i 台机组运行功率的最大值和最小值, P_i 为对应的第 i 台机组的有效功率。

综上所述,考虑了阀点效应、功率平衡约束和运行功率约束的ELD问题的数学模型为:

$$\min F = \sum_{i=1}^m F_i(P_i) = \sum_{i=1}^m a_i + b_i P_i + c_i P_i^2 + |d_i \sin(e_i(P_i^{\min} - P_i))| \quad (7)$$

$$\text{s. t. } \begin{cases} \sum_{i=1}^m P_i = P_D \\ P_i^{\min} \leq P_i \leq P_i^{\max} \end{cases} \quad (8)$$

由上式可知,ELD 问题的数学模型中含有非线性等式和不等式约束的限制,其函数变量不可微且不连续,所求的可行解范围非凸。为更有效地求解 ELD 问题,该文提出将原先的两种算法加以改进并组合成一种新型优化算法。

3 改进的差分进化鲸鱼优化算法

3.1 经典鲸鱼优化算法

WOA 算法的灵感来源于座头鲸的捕猎行为。算法的主要步骤为包围猎物、气泡网捕食、搜寻新猎物三步,算法的寻优过程如下:

(1) 包围猎物。

包围猎物阶段,当前种群的鲸鱼个体根据食物所在位置开展包围行动,以当前迭代中适应度最佳的鲸鱼坐标作为猎物所在的位置,而其他鲸鱼则通过不断更新自己坐标来包围猎物,其位置更新公式如下:

$$X_{G+1} = X^* - A \cdot |X_G - C \cdot X^*| \quad (9)$$

其中, X^* 为本次迭代最优解个体的坐标, X_G 为该鲸鱼种群的坐标, A 为收敛因子,由式(10)得出:

$$A = 2a \cdot r - a \quad (10)$$

A 采用线性递减策略从 2 变化至 0。 C 为一个随机数,表示鲸鱼捕食的摆动行为,由式(11)得出:

$$C = 2 \cdot r \quad (11)$$

在式(10)和式(11)中, r 表示为(0,1)的常数。

(2) 气泡网捕食。

WOA 中的气泡网捕食阶段,主要由螺旋吐气泡与包围捕食两种策略构成,其中螺旋运动按照公式(12)进行移动:

$$X_{G+1} = D \cdot e^{bl} \cdot \cos 2\pi l + X^* \quad (12)$$

式中, $D = |X^* - X_i|$ 表示该个体与目标猎物之间的距离, b 为螺旋线常量,一般取 1,用于定义鲸鱼运动方式, l 取值为 $[-1, 1]$ 。

在 WOA 中,鲸鱼螺旋线运动与包围捕食同时进行,故通常假设鲸鱼均有一半的可能性选择其中一种方式靠近猎物,其坐标迭代表达式如式(13)所示:

$$X_{G+1} = \begin{cases} X_G - A \cdot |X_G - C \cdot X^*| & \text{if } p < 0.5 \\ D \cdot e^{bl} \cdot \cos(2\pi l) + X^* & \text{if } p \geq 0.5 \end{cases} \quad (13)$$

式中, p 为 $[0, 1]$ 的随机数。

(3) 搜寻新猎物。

当 $|A| < 1$ 时,鲸鱼继续进行包围性的搜寻,并依照式(9) 改变其本身的坐标;当 $|A| \geq 1$ 时,鲸鱼进行随机的全局搜寻,防止出现局部最优。其坐标迭代表

达式如下:

$$D' = |X_G - C \cdot X_{\text{rand}}| \quad (14)$$

$$X_{G+1} = X_{\text{rand}} - A \cdot D' \quad (15)$$

其中, X_{rand} 表示该种群中的任意一头鲸鱼, D' 为此鲸鱼与其寻找到的更优猎物间的距离。

3.2 经典差分进化算法

(1) 变异环节。

在当前种群 G 的所有个体 $X_{i,G}$ ($i = 1, 2, \dots, N$) 中选取互异的三个作为变异个体:

$$V_{i,G+1} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \quad (16)$$

式中, r_1, r_2, r_3 表示 $[1, N]$ 中的随机自然数,同时满足 $r_1 \neq r_2 \neq r_3 \neq i$, F 为缩放因子,文中取 $F = 0.5$ 。

(2) 交叉环节。

交叉环节为使交叉后的种群 $U_{i,G+1}$ ($i = 1, 2, \dots, N$) 中能至少包含一个变异个体,采用以下策略保证变异个体 $V_{i,G}$ 的贡献率:

$$U_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{rand}(j) \leq \text{CR or } j = \text{rnb}(i) \\ x_{i,j,G+1} & \text{rand}(j) > \text{CR or } j \neq \text{rnb}(i) \end{cases} \quad (17)$$

式中, $\text{rand}(j)$ 的取值范围为 $[0, 1]$; $\text{rnb}(i)$ 表示 $[1, N]$ 中的随机整数; CR 为交叉概率因子,用以控制交叉的程度,文中取 $\text{CR} = 0.5$ 。

(3) 选择环节。

交叉环节后得到的个体 $U_{i,G+1}$ 与当前个体 $X_{i,G}$ 进行优劣对比,选择适应度更佳的个体放入下一代种群。

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \leq f(X_G) \\ X_G & \text{if } f(U_{i,G+1}) > f(X_G) \end{cases} \quad (18)$$

3.3 改进的差分进化鲸鱼优化算法

文中将具有较强探索能力的 DE 整合到 WOA 中,来完善 WOA 开发算法空间的能力。每轮迭代的初始阶段,WOA 利用气泡网和猎物搜寻机制对种群个体的位置进行预处理变异,生成初始化种群,再由 DE 对鲸鱼算法处理过的种群进行交叉和选择,来进行种群的更新迭代。具体改进方法如下:

3.3.1 约束条件处理

考虑到功率平衡约束在高维情况下复杂度过高的情况,文中提出一种将运行约束和功率平衡约束同时考虑在内的策略,具体步骤如下:

Step1: 设置种群序号为 i ($i = 1, 2, \dots, N$), N 为种群数量。

Step2: 计算当前种群 i 中所有机组的功率之和 P_{sum} 与系统需求的总负荷值 P_D , 并计算其差值。若差值不为 0,则继续下一步;若为 0,则跳转至 Step5。

Step3: 随机生成一个序号 r (r 为 $[1, m]$ 内的任意整数),将该序号下 r 机组的有功功率 P_i 设置为

$P_i = P_i - (P_{\text{sum}} - P_D)$ 。

Step4: 检查 P_i 的机组运行约束,若超出上下限,则将其设置为上下限。跳转至 Step2。重新验证是否满足差值为 0。

Step5: $i = i + 1$ 依次对所有种群的机组进行约束调整,直到所有种群的 P_{sum} 与系统需求的总负荷值 P_D 的差值均为 0。

3.3.2 鲸鱼优化算法取代差分进化算法的变异部分

$$W_{i,G+1} = \begin{cases} X_G - A \cdot |X_G - C \cdot X^*| & \text{if } p < 0.5 \text{ and } |A| < 1 \\ X_{\text{rand}} - A \cdot |X_G - C \cdot X_{\text{rand}}| & \text{if } p < 0.5 \text{ and } |A| \geq 1 \\ D \cdot e^{bl} \cdot \cos(2\pi l) + X^* & \text{if } p \geq 0.5 \end{cases} \quad (19)$$

其中, $W_{i,G+1}$ 为新一代变异种群。即用 $W_{i,G+1}$ 代替 $V_{i,G+1}$ 进行下一步的交叉操作。

同时改进鲸鱼优化算法的变异操作:WOA 的探索 and 开发能力的相互协调引导着整个算法的走向。其中,探索能力指的是搜索更广的区域范围,跳出局部最优的能力;开发能力指的是对已知解附近的区域进行局部搜索的能力,有利于加快收敛速度。

3.3.3 收敛因子的改进

通过文献[11,15]可知,WOA 算法中的收敛因子 a 对该算法的探索 and 开发能力有着至关重要的影响:收敛因子 a 越大,算法的全局搜索性能越强;若收敛因子 a 越小,则算法的局部寻优性能越强,收敛速率越高。而在标准 WOA 算法中,随着迭代的进行,收敛因子 a 这种线性递减的方式虽然让算法在前期拥有较好的全局搜索能力,但收敛速度过慢;而当算法即将达到最大迭代次数时,收敛因子 a 才逼近零点,其收敛速度才有所加快,但已经无法摆脱局部最优的困境,没有做到探索 and 开发的良好协调。因此原始的 WOA 算法中的收敛因子并不能有效利用算法的优越性。

现有的研究中,已有多重非线性更新策略。文献[12]中提出用三角函数代替原收敛因子,公式如下:

$$a = 2 - 2\sin\left(\frac{\pi}{2} \frac{t}{T}\right) \quad (20)$$

式中, t 和 T 分别表示当前迭代次数和最大迭代次数,下同。

文献[16]则提出采用分段的收敛因子更新策略,公式如下:

$$a = \begin{cases} 2 - \left(\frac{t}{T}\right)^2 & t \leq T/2 \\ 1 - \frac{2(t - \frac{T}{2})}{T} + \left(\frac{t - \frac{T}{2}}{T}\right)^2 & t > T/2 \end{cases} \quad (21)$$

文献[17]提出一种以 e 为底的正指数形式的收敛因子更新策略,公式如下:

$$a = 2 - \frac{2(e^{\frac{t}{T}} - 1)}{e - 1} \quad (22)$$

文中提出一种以常数为底的负指数形式的自适应收敛因子更新策略:

$$a = 2 * \mu^{(-\frac{t}{T})} \quad (23)$$

其中, μ 表示一常数,其取值根据算例的自身情况而调整,经过实验测试,取 $\mu = 1\,000$ 为最佳。

将上述四种策略函数与线性递减的曲线绘制在同一幅图中,如图 1 所示(取迭代次数 $T = 2\,000$)。

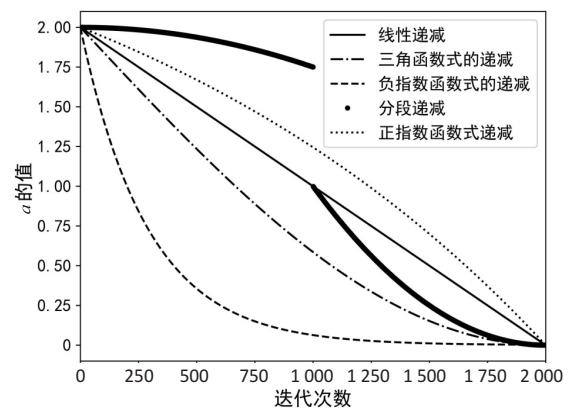


图1 收敛因子 a 的变化曲线

从图 1 中可知:负指数形式递减的更新策略相比于其他策略函数,其在算法的早期迭代阶段保持了较大的 a 值,使算法跳出局部极值的机会更多;在算法的中期,函数单调递减幅度高于其他四种策略,且能够将收敛因子 a 骤降到一个较小的值,从而有效保证收敛的高效性;而后期最优解的搜索范围基本确定,该策略中的收敛因子 a 又处于较小值且递减速度缓慢,算法的最终收敛精度得到了保障。

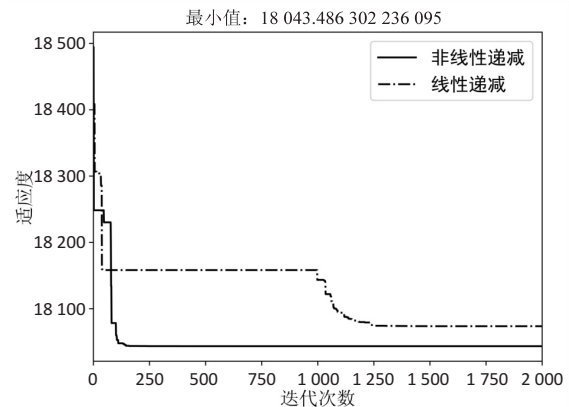


图2 非线性递减与线性递减的收敛对比

如图 2 所示,文中将仅采取非线性负指数形式递减策略的鲸鱼算法与经典鲸鱼算法进行检验(算例采用文章中的 13 台机组系统)对比。由图可知,采取非线性收敛因子的鲸鱼算法取得的全局最优极小值所需

要的迭代次数更少,极值更小,且在早期阶段就实现收敛,可有效改善传统鲸鱼算法中线性递减所带来的收敛性能不佳等负面效应。

3.3.4 引入交叉操作和淘汰机制

改进差分进化的交叉操作:交叉因子 CR 的大小很大程度上决定了差分进化算法的收敛性能。从等式(17)可知,CR 值越大, V_i 对 U_i 的贡献率越高,这意味着交叉操作后种群中将包含更多的变异个体;CR 的值越小, X_i 对 U_i 的贡献率越低,算法开拓其他区域解的能力越差,收敛效率不佳,但有利于保留原始个体特征(维持物种的多样性),从而使得算法在局部求解期间具有更高的成功率。

因此,若要在早期阶段更稳定地保持种群的多样性,则应从一个较小的 CR 值出发,以维持初期的多样性和成功率,而后逐渐增加 CR,提升后期的收敛速度,从而有效减少整个迭代中陷入局部最优的次数,这才是较为理想的 CR 变化规律。

此外,文中提出在交叉操作中提前筛选变异的个体,从而为种群中优秀的个体提供更多的机会;而劣势个体则被赋予较低的选择概率而被逐渐淘汰。因此,文中提出了一种基于个体适应度的淘汰机制(以求解最小值为例):

$$CR = \begin{cases} CR_{\min} + (CR_{\max} - CR_{\min}) \cdot \frac{f_i - f_{av}}{f_{\min} - f_{av}} & \text{if } f_i < f_{av} \\ CR_{\min} & \text{otherwise} \end{cases} \quad (24)$$

其中, CR_{\max} 和 CR_{\min} 分别为 CR 的最大值和最小值。同时,每次迭代都会计算所有种群个体适应度的平均值 f_{av} 并寻找最佳的适应度 f_{\min} 。以寻找全局最小值为例,如果种群适应度的平均值 f_{av} 低于个体当前的适应度 f_i ,可以将此个体视为优势个体。单独个体的交叉率 CR 将根据接近最佳适应度的程度而自适应调整。如果适应度的平均值 f_{av} 高于个体当前的适应度 f_i ,则可以将此个体视为劣势个体。该个体的 CR 将被重新设置为 CR_{\min} 。因此,优势个体将不断被保留至下一代,劣势个体由于被选择的概率较低而逐渐减少。

该方法下,适应度更好的个体将被保留,同时 CR 将逐步趋近于最大参数值 CR_{\max} ,从而契合 CR 的最佳变化规律。

3.3.5 正常的选择环节

图 3 为 IDEWOA 算法求解电力系统经济负荷分配模型的流程图。

4 算例仿真与分析

为测试 IDEWOA 算法对电力系统经济负荷的优化分配能力,分别采用标准 DE、标准 WOA 和

IDEWOA 算法对 13 台机组和 40 台机组的电力系统经济负荷进行最优化求解,引入阀点效应,暂不考虑电网损耗,系统的基本参数设置均来自文献[18]。

为避免算法随机性问题,分别对三种算法独立执行 40 次。图 4 和图 5 分别为三种算法对 2 个测试算例的收敛曲线图,表 1 和表 3 为多种算法对 2 个测试算例的统计结果。

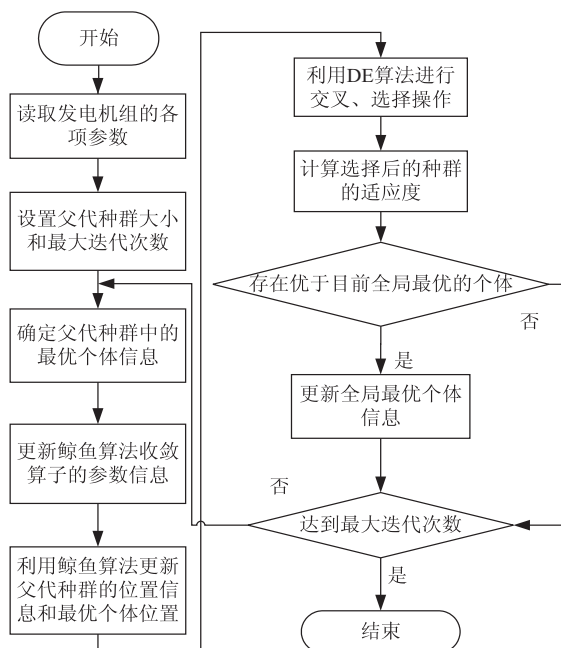


图 3 IDEWOA 算法流程

4.1 13 台机组的负荷分配

13 台机组系统要求的负荷值为 $D = 1\,800\text{ MW}$,维数为 13,种群规模为 65(一般设置为维数的 5~10 倍), CR_{\max} 和 CR_{\min} 分别取 1 和 0.5,迭代次数为 2 000 次。

表 1 13 台机组结果对比

算法	平均值	最大值	最小值	标准差
PSO ^[19]	18 256.23	18 489.98	18 095.90	114.00
SA ^[19]	18 160.31	18 337.76	18 050.25	76.00
CS ^[19]	18 058.90	18 108.19	17 997.51	25.10
DE	18 093.86	18 118.19	18 027.29	22.66
WOA	18 052.73	18 097.38	17 977.50	42.77
IDEWOA	17 972.88	17 972.96	17 972.81	0.04

IDEWOA 与其他算法仿真结果比较见表 1。其中,DE、WOA 和 IDEWOA 算法在 Python 3.7 上分别运行 40 次,统计并分析所得运算结果,PSO、SA 和 CS 的仿真结果参考自文献[19]。13 台机组的 IDEWOA 算法与基本 DE、基本 WOA 算法的运行结果对比如图 4 所示。图中所选取为 IDEWOA 算法中较优的一次运行结果。

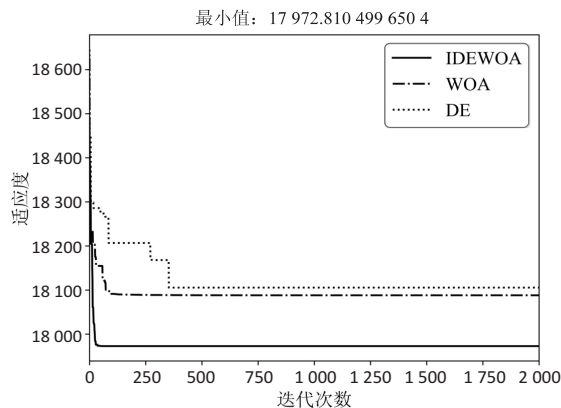


图4 13 机组运行对比

从图4中的曲线和表1中的数据可知,IDEWOA 算法相比于其他标准的 DE、WOA 以及其他类似的群智能算法,能够取得更好的效果,收敛速度提升显著,搜索精度较高。表1 数据显示:IDEWOA 算法运行平均值为 17 972.88,明显优于其他算法的最优解,且实验的标准差仅为 0.04,证明该算法所求解比较稳定,鲁棒性较强。从图4中可以看出,WOA 和 DE 算法在早期容易陷入早熟的特点较为明显,而 IDEWOA 经过改进后能够在短时间内多次跳出局部最优解区域,并在算法中期之前就取得了精确度较高的解。综上所述,IDEWOA 在 13 台机组的负荷分配问题上能够有效降低能耗,减少经济成本。

表2 为 IDEWOA 算法求解得到的仿真实验结果。该条件下求得的最优解即为 17 972.81。

表2 13 机组功率分配情况

机组	功率	机组	功率	机组	功率
P_1	628.32	P_6	60.0	P_{11}	40.0
P_2	224.40	P_7	60.0	P_{12}	40.0
P_3	297.55	P_8	60.0	P_{13}	55.0
P_4	109.87	P_9	60.0		
P_5	109.87	P_{10}	60.0		

4.2 40 台机组的负荷分配

为了测试 IDEWOA 算法在更高维环境中的稳定性,文中选择 IEEE 40 台机组 300 节点作为实验对象,该算例中机组数(维数)较多,增大了求解的难度,极其考验算法高维度的求解能力。其系统所需负荷值为 $D=10\,500\text{ MW}$,维数为 40,种群规模取维数的五倍,即 200, CR_{\max} 和 CR_{\min} 分别取 1 和 0.5,最大迭代次数 $t_{\max}=2\,000$ 。

IDEWOA 与其他 5 种算法仿真结果对比见表 3。其中同样对 DE、WOA 和 IDEWOA 算法分别在 Python 3.7 上独立随机地运行 40 次,PSO、SA 和 CS 的仿真结果参考自文献[19]。40 台机组的 IDEWOA 算法与基本 DE、基本 WOA 算法的运行结果对比如图 5 所

示。图中选取的为 IDEWOA 算法较优的一次运行结果。

表3 40 台机组结果对比

算法	平均值	最大值	最小值	标准差
PSO ^[19]	126 149.57	128 477.99	123 980.64	1 240.00
SA ^[19]	123 588.41	124 419.54	122 437.56	537.00
CS ^[19]	122 661.26	122 945.44	122 373.79	165.00
DE	124 335.19	125 051.93	123 577.44	433.51
WOA	120 883.26	120 988.60	120 858.82	38.37
IDEWOA	120 802.09	120 810.37	120 645.74	6.45

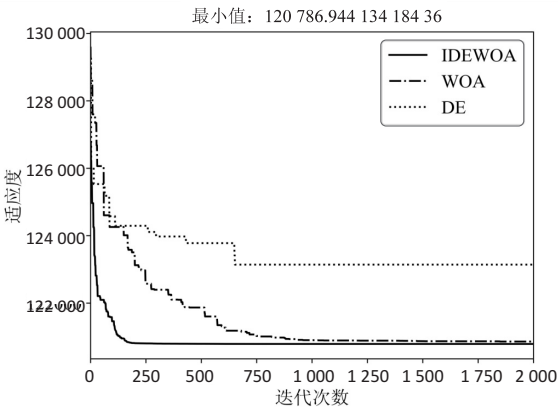


图5 40 机组运行对比

从图5中的曲线和表3中的数据可知,在高维度算例中,IDEWOA 算法相比于其他算法,优化配置能力明显更强。仿真实验的运行结果显示:IDEWOA 算法所得最优值的平均值、最大值和最小值均为所列算法中最低,其收敛性能优秀,其标准差为 6.45 也比其他算法低,说明在高维度下,IDEWOA 算法仍能保持较好的鲁棒性。

表4 40 机组功率分配情况

机组	功率	机组	功率	机组	功率
P_1	114.0	P_{15}	304.520 3	P_{29}	10.007 8
P_2	114.0	P_{16}	304.577 7	P_{30}	97.0
P_3	120.0	P_{17}	489.433 2	P_{31}	190.0
P_4	180.028 9	P_{18}	489.349 2	P_{32}	190.0
P_5	97.0	P_{19}	511.377 8	P_{33}	190.0
P_6	140.0	P_{20}	511.361 3	P_{34}	200.0
P_7	300.0	P_{21}	525.07	P_{35}	200.0
P_8	300.0	P_{22}	525.30	P_{36}	200.0
P_9	300.0	P_{23}	526.707 5	P_{37}	110.0
P_{10}	135.026 7	P_{24}	524.605 5	P_{38}	110.0
P_{11}	94.065 5	P_{25}	528.253 3	P_{39}	110.0
P_{12}	94.011 8	P_{26}	523.489 1	P_{40}	511.531 2
P_{13}	214.795 0	P_{27}	10.011 9		
P_{14}	394.367 0	P_{28}	10.101 6		

表 4 为 IDEWOA 算法求解得到的仿真实验结果。该条件下求得的最优解即为 120 645.74。

通过上述算例可以发现, IDEWOA 算法相比其他算法, 具有较好的全局收敛能力, 在寻优速度和收敛精度方面都有所提升, 且在高维条件下仍保持了良好的稳定性, 优越性明显。

5 结束语

文中提出一种改进的差分进化鲸鱼优化 (IDEWOA) 算法应用于电力系统经济负荷的寻优分配, 利用该算法可以解决非凸、非线性约束、高维度的优化问题。首先在鲸鱼算法中的搜寻包围猎物时期, 调整了原有的线性收敛因子, 使其更适应整个算法的寻优同时兼顾全局探索和局部开发能力。然后结合差分进化算法的交叉选择策略, 同时引入自适应淘汰机制, 加快整个算法的收敛, 推进了搜寻最优解的进程。最后将该算法应用于电力系统 13 机组和 40 机组负荷分配的优化测试, 测试结果验证了利用 IDEWOA 求解负荷分配问题的优越性和有效性。

下一步的研究是如何将 IDEWOA 算法应用于多目标优化领域以及如何改进算法中的控制参数。

参考文献:

- [1] 何佳欢. 基于差分进化算法的火电机组负荷分配问题研究 [D]. 沈阳: 沈阳工业大学, 2016.
- [2] GAING Z. Particle swarm optimization to solving the economic dispatch considering the generator constraints [J]. IEEE Transactions on Power Systems, 2003, 18 (3): 1187-1195.
- [3] CIORNEI I, KYRIAKIDES E. A GA-API solution for the economic dispatch of generation in power system operation [J]. IEEE Transactions on Power Systems, 2012, 27 (1): 233-242.
- [4] 陈皓, 潘晓英, 张洁. 一种基于簇类进化的电力经济负荷分配优化算法 [J]. 计算机研究与发展, 2016, 53 (7): 1561-1575.
- [5] STORN R, PRICE K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11 (4): 341-359.
- [6] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [7] RAHMAT N, MUSIRIN I. Differential evolution ant colony optimization (DEACO) technique in solving economic load dispatch problem [C]//2012 IEEE international power engineering and optimization conference. Melaka, Malaysia: IEEE, 2012: 263-268.
- [8] SEN T, MATHUR H. A new approach to solve economic dispatch problem using a hybrid ACO-ABC-HS optimization algorithm [J]. International Journal of Electrical Power and Energy Systems, 2016, 78 (Jun.): 735-744.
- [9] CHEN X. Novel dual-population adaptive differential evolution algorithm for large-scale multi-fuel economic dispatch with valve-point effects [J]. Energy, 2020, 203: 117874.
- [10] 银星黎. 基于改进多目标鲸鱼算法的水库群供水-发电-生态优化调度研究 [D]. 武汉: 华中科技大学, 2019.
- [11] 黄元春, 张凌波. 改进的鲸鱼优化算法及其应用 [J]. 计算机工程与应用, 2019, 55 (21): 219-225.
- [12] 孙希延, 范灼, 纪元法. 基于混合鲸鱼优化算法的鲁棒多用户检测 [J]. 科学技术与工程, 2019, 19 (13): 114-119.
- [13] 李玲玲, 王鑫, 郎永波, 等. 基于改进鲸鱼算法的微网复合储能系统容量优化配置 [J]. 电测与仪表, 2019, 56 (16): 104-110.
- [14] 滕德云, 滕欢, 潘晨, 等. 基于鲸鱼优化算法的无功优化调度 [J]. 电测与仪表, 2018, 55 (24): 51-58.
- [15] 龙文, 蔡绍洪, 焦建军, 等. 求解大规模优化问题的改进鲸鱼优化算法 [J]. 系统工程理论与实践, 2017, 37 (11): 2983-2994.
- [16] 郝晓弘, 宋吉祥, 周强, 等. 混合策略改进的鲸鱼优化算法 [J]. 计算机应用研究, 2020, 37 (12): 3622-3626.
- [17] 孙鹤旭, 张维, 雷兆明, 等. 基于改进鲸鱼优化算法的风电消纳调度研究 [J]. 高技术通讯, 2020, 30 (3): 298-304.
- [18] SINHA N, CHAKRABARTI R, CHATTOPADHYAY P. Evolutionary programming techniques for economic load dispatch [J]. International Journal of Emerging Electric Power Systems, 2003, 7 (1): 83-94.
- [19] 何湘竹, 黄继达. 基于约束布谷鸟算法的电力系统经济负荷分配 [J]. 计算机工程, 2016, 42 (3): 214-219.