

# 基于 IWOA 算法求解并联冷机负荷分配问题

于军琪<sup>1</sup>, 康智桓<sup>1</sup>, 荆 竞<sup>2</sup>, 赵安军<sup>1</sup>, 陈时羽<sup>2</sup>

(1. 西安建筑科技大学, 西安 710055;

2. 中国建筑西北设计研究院有限公司, 西安 710018)

**摘要:** 中央空调系统并联冷水机组系统能耗非常大, 如果操作不当, 能耗会大大增加; 针对 OCL 问题提出了一种求解连续非线性优化问题的改进鲸群优化算法, 首先, 为使后续迭代寻优的搜索空间更精确, 运用混沌映射初始化种群, 使初始解均匀遍布解空间; 其次引入变异指数对收敛因子进行改进, 平衡了局部勘探和全局勘探的关系; 之后引入正弦和余弦使算法收敛到全局最优解, 防止了算法过早收敛, 提高了算法的收敛精度; 最后, 通过两个典型的案例来评估 IWOA 算法的性能, 并将其与应用于 OCL 问题的其他优化算法进行了比较; 结果表明, IWOA 算法是解决 OCL 问题的有效方法; 此外, 算法性能的比较显示, IWOA 算法在收敛速度和电能消耗方面相比于其他应用于 OCL 问题的优化方法提供了更好的解决方案。

**关键词:** 负荷分配; 并联冷机; 鲸鱼优化算法; 混沌映射

## Solving the Load Distribution Problem of Parallel Chillers Based on IWOA Algorithm

YU Junqi<sup>1</sup>, KANG Zhihuan<sup>1</sup>, JING Jing<sup>2</sup>, ZHAO Anjun<sup>1</sup>, CHEN Shiyu<sup>2</sup>

(1. Xian University of Architecture and Technology, Xian 710055, China;

2. Northwest Architectural Design and Research Institute, Xian 710018, China)

**Abstract:** The parallel chiller energy consumption of central air-conditioning systems is very large, improper operation will greatly increase energy consumption. Aimed at the optimal chiller load (OCL) problem, an improved whale population optimization algorithm is proposed to solve the continuous nonlinear optimization problem. Firstly, in order to make the search space of the subsequent iteration more accurate, the chaotic mapping is used to initialize the population, so that the initial solution evenly spread across the solution space. Secondly, the variation index is introduced to improve the convergence factor and balance the relationship between local exploration and global exploration. After that, sine and cosine are introduced to make the algorithm converge to the global optimal solution, and preventing the premature convergence of the algorithm and improving the convergence accuracy of the algorithm. Finally, the performance of IWOA algorithm is evaluated by two typical cases, and compared with other optimization algorithms that applied to the OCL problem, the results show that the IWOA algorithm is an effective method by solving the OCL problem. In addition, the comparison of the algorithm performance shows that the IWOA algorithm provides a better solution than other optimization methods applied to the OCL problem in terms of convergence speed and power consumption.

**Keywords:** load distribution; parallel cold machine; whale optimization algorithm; chaotic mapping

## 0 引言

近年来在中国, 建筑消耗了大约 20% 的一次能源。城市建筑可以消耗高达 70% 的城市一次能源<sup>[1]</sup>。所以提高建筑能源效率对于可持续发展至关重要。冷水机组通常服务于商业建筑。作为中央空调系统的主要部件, 冷水机组约占商业建筑制冷能耗的 35%。由于冷水机组能耗巨大, 因此, 如何在不同冷负荷下运行制冷机系统, 使能耗最小化成为一个重要的问题<sup>[2]</sup>。通过仔细设计和运行冷水机组, 发现多台冷水机组系统由不同性能特性和容量的冷水机组成, 因此各冷水机的最优负荷比组合成为一个有价值的研究课题。

近年来, 冷水机组负荷优化分配 (OCL, optimal chiller load) 问题受到越来越多的关注, 研究人员使用各种优化算法不断对优化结果进行改进。文献 [3] 在最优冷水机组负荷研究中采用拉格朗日法 (LM, Lagrange method) 最小化不同冷负荷下的能耗, 结果表明拉格朗日法能使能耗最小化, 但在低需求时不能收敛; 紧接着文献 [4] 提出利用遗传算法求解, 遗传算法克服了低需求下的收敛性, 但相比于拉格朗日法平均能耗提高了 0.4% 左右; 文献 [5-6] 又提出利用模拟退火法 (SA, simulated annealing) 和梯度法 (GM, gradient method) 改进拉格朗日法的收敛性, 发现该方法能在短时间内产生精度较高的结果。文献

收稿日期: 2022-10-18; 修回日期: 2022-11-30。

基金项目: 国家重点研发计划项目 (2017YFC0704100)。

作者简介: 于军琪 (1969-), 男, 陕西宝鸡人, 博士, 教授, 主要从事智能建筑、暖通空调方向的研究。

引用格式: 于军琪, 康智桓, 荆 竞, 等. 基于 IWOA 算法求解并联冷机负荷分配问题[J]. 计算机测量与控制, 2023, 31(7): 169-176, 183.

[7] 采用粒子群算法 (PSO, particle swarm optimization) 对问题进行求解, 发现 PSO 相比于 GA 算法能够获得更好的优化结果。因为 OCL 问题的变量是连续的, 所以文献 [8] 采用连续遗传算法 (CGA, continuous genetic algorithm) 和粒子群算法来解决, 而研究也表明粒子群算法能够改善遗传算法和拉格朗日法的性能。文献 [9] 提出了一种差分进化算法, 该算法在寻优方面优于遗传算法, 并克服了遗传算法存在的问题。文献 [10] 提出了一种基于高斯分布函数的改进萤火虫算法 (FA, firefly algorithm), 优化了多冷却器负载问题中的能耗。文献 [11] 提出了一种新的差分布谷鸟搜索算法 (DCSA, differential cuckoo search approach), 该算法基于差分算子, 提高了算法的整体寻优能力。文献 [12] 使用非线性波纹权重因子和自适应排斥因子的改进波纹蜜蜂群优化算法 (IRBSO, improved ripple bee swarm algorithm) 算法, 并将其用于解决冷水机组的经济调度问题。文献 [13-4] 提出的改进人工鱼群算法 (IAFSA, improved artificial fish swarm algorithm) 和改进入侵杂草算法 (EIWO, improved invasive weed optimization) 也为解决此问题提供了更好的解决方案。文献 [15] 提出运用改进分数阶达尔文粒子群算法 (IFODPSO, improved fractional order Darwinian particle swarm optimization) 求解, 也是解决 OCL 问题的有力工具。结果表明, 与现有优化算法相比, 该算法在收敛精度、收敛速度和稳定性等方面的综合性能有了显著提高。以上研究表明, 元启发式算法和数学规划方法都是解决 OCL 问题强有力的工具且均能从中央空调系统实际用能情况出发解决问题。但是元启发式算法需要较大种群规模、对种群初始化要求较高等问题使得一种优化算法在特定优化问题上的优化性能较好, 但是并不能推广到大多数优化问题上从而算法适用性大打折扣。鲸鱼优化算法<sup>[16]</sup> (WOA, whale optimization algorithm) 是 2016 年提出的一种基于鲸鱼捕食习惯而研究出的一种新的智能算法, 本文以该算法为基础解决冷水机组负荷优化分配问题。

WOA 算法的推广引起了研究者们的关注, 并在文献 [17-18] 中用于求解单目标热电厂经济调度问题。基本的 WOA 为中小型测试系统提供了更高质量的解决方案, 并且收敛速度更快。但是, 对于大规模复杂问题, 基本 WOA 算法由于早熟收敛、收敛速度慢等问题往往不能收敛到最优解。这是因为在基本的 WOA 算法中, 局部勘探和全局开发之间存在差异。针对基本 WOA 出现的问题, 许多研究者都做了不同的修改, 通过适当平衡勘探阶段和开发阶段来提高 WOA 性能。文献 [19] 引入了改进的混沌鲸优化算法 (CWOA, chaos whale optimization algorithm) 来求解太阳能光伏电池的参数估计, 该方法利用混沌映射进行计算, 并自动选择方法内部参数。文献 [20] 引入了一种带局部搜索策略的混合鲸优化方法来求解置换流水车间调度问题。文献 [21] 采用了一种基于最新开发的 WOA 的新的 MLP 训练方法来优化神经网络中的连接权值。文献 [22]

为了准确提取不同太阳能光伏模型问题的参数, 提出了一种改进的鲸鱼优化算法。该方法采用了两种猎物搜索策略, 有效的保持了勘探与利用的平衡, 提高了 WOA 算法的性能。从以上研究可以看出, 不同学者对于基本鲸群算法出现的收敛速度慢、寻优精度低、早熟收敛等问题提出的各种改进鲸群算法应用于各领域的优化问题均获得了良好的实验结果。

综上所述, 由于 OCL 问题优化变量的维度及最终结果由冷机的数量及各冷机的性能参数决定, 基本鲸群算法已经无法满足针对 OCL 问题中出现的高维问题, 快速收敛到全局最优解。因此, 提出一种改进鲸群算法 (IWOA, Improved whale optimization algorithm), 并通过仿真实验对其性能进行检验。

## 1 问题描述

冷水机组由于运行灵活、备用容量大、检修中断少等特点, 常采用并联的多套冷水机组与共用配电系统相连接<sup>[23]</sup>。图 1 展示了一个典型的解耦冷冻水系统。冷冻水系统的多台泵通过旁通管连接, 旁通管连接回水总管和供水总管。冷水机组与泵机组一体化, 相互独立运行<sup>[24]</sup>。考虑到需求增量的发生, 需要比冷水泵提供的流量更大的流量。总管由回流水通过旁路提供, 旁路定义了额外冷水机组容量的要求。因此, 将启动另一台冷水机泵。相反方向的水流决定了产能过剩的条件, 这就关闭了冷水泵。在全电制冷的多台冷水机组系统中, 在满足负荷需求的情况下, 各机组能耗之和最小时, 机组性能最佳。PLR (partial load ratio) 定义为冷水机的冷负荷与冷水机功耗的比值。

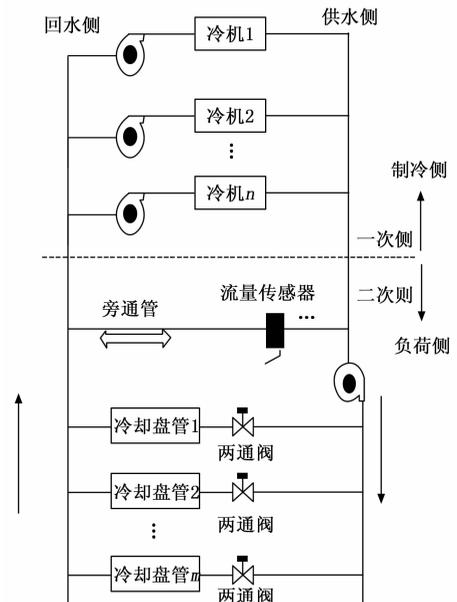


图 1 多冷机系统结构图

每台冷机的性能参数都有所不同, 冷机负荷的变化取决于末端负荷需求的变化。为了将系统冷负荷分配给制冷机组, 多冷机系统允许可变的水流通过冷却盘管, 并保持

恒定的水流通过冷却器。此外, 允许供水和回水通过旁通管道彼此流动, 并且可以根据系统冷负荷需求来控制水流。

典型 OCL 问题的优化目标, 是使多台冷水机组系统的总功耗最小化。因此, 它主要是指在能满足整个系统末端负荷需求情况前提下, 追求达到一种至少不应超过整个系统的运行功率极限条件下的系统最大的冷机部分负荷比 PLR 使末端冷机部分的负荷功率总和达到最小, 从而获得最佳的系统性能, 实现节能减排的目的<sup>[15]</sup>。因此本文将离心式冷水机组的功耗描述为:

$$P_i = a_i + b_i * R_i + c_i * R_i^2 + d_i * R_i^3 \quad (1)$$

式中,  $P_i$  为第  $i$  台冷水机组功率,  $a_i$ 、 $b_i$ 、 $c_i$ 、 $d_i$  为第  $i$  台冷机本身的功耗系数,  $R_i$  为第  $i$  台冷机的部分负荷比。

OCL 问题是指在满足给定的目标函数最小的情况下, 求一个不大于中央空调系统最低运行功率限制条件下的冷水机组部分负荷率 PLR 值, 使之能在满足整个中央空调系统末端负荷要求条件的情况条件下仍以系统最低限度的最低运行总功率来工作, 考虑到系统的冷负荷平衡, 目标函数如 (2) 式所示应最小化:

$$OF = \sum_{i=1}^m P_i \quad (2)$$

式中,  $OF$  为并联冷机系统总功率,  $m$  为并联冷机台数, 根据制造厂商的建议, 当冷却器打开时, PLR 值必须大于 0.3。当冷却器关闭时, PLR 值为 0。解决 OCL 问题的目的是找到冷水机组 PLR 的最佳值, 该值可以获得最小的运行功率并满足运行约束。

因此, 并联冷机负荷分配优化模型的目标函数和约束条件的数学式如下:

$$\min(P) \quad (3)$$

$$\text{s. t. } 0.3 \leq R_i \leq 1 \text{ 或 } R_i = 0 \quad (4)$$

$$\sum_{i=1}^m R_i \times Q_i^0 = CL, i = 1, 2, \dots, N \quad (5)$$

式中,  $Q_i^0$  为第  $i$  台冷水机组的额定制冷量;  $CL$  为系统末端冷却负荷需求。

## 2 鲸群优化算法

### 2.1 标准鲸群优化算法

标准鲸群优化算法 (WOA, whale optimization algorithm) 包括几个步骤:

#### 2.1.1 包围猎物

WOA 算法将目前群体的最佳结果作为精英鲸鱼的位置, 其余个体将精英鲸鱼的位置作为猎物的位置。通过改变位置进而包围猎物, 鲸群的行为公式如下:

$$\vec{D} = |\vec{CX} * (t) - \vec{X}(t)| \quad (6)$$

$$\vec{X}(t+1) = X * (t) - \vec{D} \cdot \vec{A} \quad (7)$$

其中:  $t$  代表当前迭代次数,  $\vec{X}(t+1)$  是  $t+1$  代更新后的个体,  $\vec{X} * (t)$  是第  $t$  代领头鲸的位置,  $\vec{X}(t)$  是更新  $t$  代后个体的位置, 如果有更好的解决方案,  $\vec{X} * (t)$  会在每次迭代中更新。向量  $\vec{A}$  和  $\vec{C}$  的计算公式如下:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (8)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (9)$$

$$\vec{a} = 2 - \frac{2t}{T_{\max}} \quad (10)$$

其中:  $\vec{r}$  是  $[0, 1]$  之间的随机数,  $\vec{a}$  的值在迭代过程从 2 线性减小到 0,  $t$  表示当前迭代次数,  $T_{\max}$  表示最大迭代次数。

#### 2.1.2 气泡网方式狩猎

气泡网觅食法是座头鲸独特的捕猎方式, 为了对座头鲸独特的捕猎方式进行数学建模, 设计了以下两种方法。

##### 1) 收缩包围机制:

这种行为利用式 (7) 更新自己的位置, 并通过降低式 (8) 中  $\vec{a}$  值来实现。其中  $\vec{A}$  值随着迭代次数增加, 逐渐从 2 线性下降到 0,  $\vec{A}$  值为  $[-2, 2]$  之间的随机数。图 2 显示了二维空间假设的猎物位置为  $(X^*, Y^*)$ , 当  $0 < A < 1$  时, 鲸群会根据公式 (7) 从  $(X, Y)$  逐渐移动到  $(X^*, Y^*)$ 。

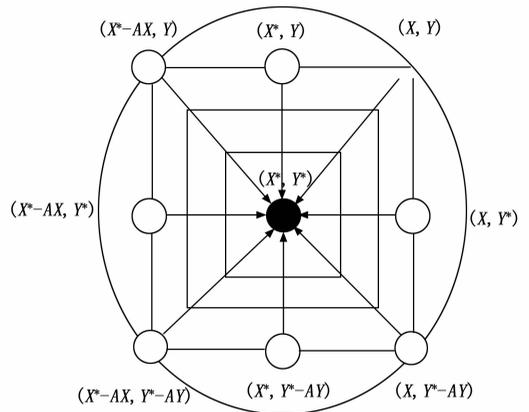


图 2 鲸群算法收缩包围策略示意图

##### 2) 螺旋包围机制:

螺旋包围机制中, 通过计算座头鲸与猎物之间的距离建立座头鲸与猎物之间的螺旋方程。螺旋方程如下:

$$\vec{X}^l(t+1) = X^*(t) + \cos(2\pi t) \cdot e^{bl} \cdot \vec{D}^l \quad (11)$$

$$\vec{D}^l = |X^*(t) - X(t)| \quad (12)$$

其中:  $\vec{X}^*(t)$  表示目前最好的位置向量,  $b$  为常数用来定义螺旋线的形状, 一般设置为 1,  $l$  是  $[-1, 1]$  中的随机数,  $\vec{D}^l$  表示鲸鱼和猎物的距离。如图 3 所示, 展示了位于  $(X, Y)$  的鲸鱼和位于  $(X^*, Y^*)$  的猎物之间的距离通过螺旋更新策略进行位置更新。

当座头鲸群不断朝猎物周围方位移动觅食时, 首先会绕着一个逐渐缩小的圆环包围猎物, 与此同时不断螺旋上升对猎物进行捕食, 为了模拟这种捕捉方式, 假设有百分之五十的可能性在收缩包围机制或螺旋包围机制中选择并更新座头鲸的位置。数学模型如下:

$$\vec{X}(t+1) = \begin{cases} \vec{X} * (t) - \vec{A} \cdot \vec{D}, & p < 0.5 \\ \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi t) + \vec{X} * (t), & p \geq 0.5 \end{cases} \quad (13)$$

其中:  $p$  为  $[0, 1]$  之间的随机数。

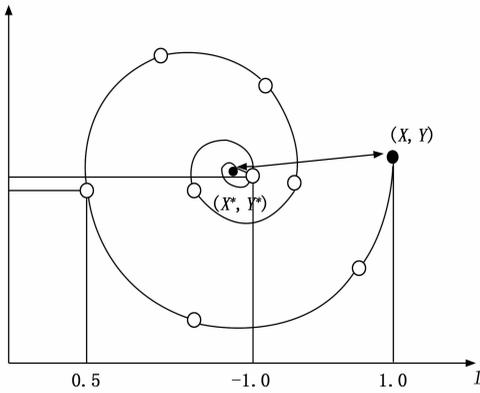


图 3 鲸群算法螺旋包围策略示意图

### 2.1.3 搜索猎物

在捕猎过程中，鲸群会参照其他族群的位置来改变自己的当前位置，以这种方式扩大搜索范围进而搜寻更优的猎物位置。WOA 算法中，每个个体根据收敛因子  $|A|$  的改变确定下一步的搜索范围。当  $|A| > 1$  时，个体鲸鱼将扩大随机捕食的搜索范围，并执行全局搜索。数学模型如下。

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (14)$$

$$\vec{X}(t+1) = \vec{X}_{rand}(t) - \vec{D} \cdot \vec{A} \quad (15)$$

其中： $\vec{D}$  是迭代过程中求解的某条鲸鱼与随机选择的单个鲸鱼之间的距离， $\vec{X}_{rand}$  是从当前种群中选择的随机位置向量。

综上所述，鲸群算法的综合搜索策略如下所示：

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D}, & p > 0.5 \\ \vec{X}_{rand}(t) - \vec{A} \cdot \vec{D}, & p < 0.5 \cup A < 1 \\ \vec{D}' \cdot e^{pt} \cdot \cos(2\pi t) + \vec{X}^*(t), & p < 0.5 \cup A \geq 1 \end{cases} \quad (16)$$

## 2.2 改进鲸群优化算法

### 2.2.1 混沌映射初始化种群

鲸鱼的初始位置是随机生成的，通常情况下初始值会随机遍布整个解空间。针对不同的优化问题，初始值对算法的收敛精度和速度有不同程度的影响<sup>[24]</sup>，基本鲸群算法在解决函数优化的问题时，往往使用了随机初始化种群的方式，而无法确保初始参数在解空间内均匀分布，这就导致当初始值远离了全局最优解，就会大大增加搜索时间，延缓算法收敛速度，同时也有陷入局部最优解的可能性。

混沌运动因为其遍历性、随机性和规律性等特点，能够使初始化后的种群更均匀地在整个解空间分布。目前，它已被广泛应用于蜂群、灰狼和粒子群优化等智能算法中。本文首先引入了 Skew tent 混沌映射，Skew tent 混沌映射在其参数区域内是一种混沌映射，同时拥有均匀的概率分布函数以及良好的相似性。与更广泛使用的 logistic 映射相比，它减少了解空间边缘区域的高分布率问题，并且具有良好的均匀分布性能。Skew tent 算法公式为：

$$\begin{cases} X_{k+1} = \frac{x}{\varphi}, & 0 < x_k < \varphi \\ X_{k+1} = \frac{1-X_k}{1-\varphi}, & \varphi < x_k < 1 \end{cases} \quad (17)$$

其中： $\varphi \in [0, 1]$ 。当  $\varphi = 0.5$  时是 tent 映射最经典的形态。此时，得到的序列是均匀分布的，并且针对各个参数有着几乎相同的分布密度<sup>[25]</sup>。所以本文引用的 tent 混沌映射如下所示：

$$X_{k+1} = \begin{cases} 2X_k, & 0 \leq X_k \leq \frac{1}{2} \\ 2(1-X_k), & \frac{1}{2} \leq X_k \leq 1 \end{cases} \quad (18)$$

### 2.2.2 收敛因子改进

在基本鲸群算法的迭代计算中，前期收敛因子  $\vec{A}$  的值越大，将增强其全局搜索能力，提高收敛速度，在算法迭代的后期，当  $\vec{A}$  取较小的值时，收敛速度会变慢，以增强其局部搜索能力，提高收敛精度。但如果  $\vec{A}$  线性下降，它将在整个迭代过程中以相同的速度下降。这种相对固定的减速速率不能维持鲸鱼种群的多样性，不利于算法的全局收敛性。故引入变异指数  $u$  对式 (10) 进行改进。

$$\vec{a} = 2 - \left( \frac{2t}{T_{max}} \right)^\mu \quad (19)$$

其中： $t$  表示当前迭代次数， $T_{max}$  为最大迭代次数， $\mu$  取 2, 1, 0.5。

如图 4 所示，当变异指数  $\mu$  为 1 时，随着迭代时间的增加，搜索间隔呈现线性缩小的趋势。当变异指数  $\mu$  为 0.5 时，算法在初始搜索阶段搜索区间迅速缩小，这可能会导致算法陷入局部最优。在后期，随着迭代次数的增加，收敛区域的缩减速度逐渐减慢。当变异指数  $\mu$  为 2 时，搜索区间在初始搜索阶段收缩非常缓慢。收敛因子在较长时间内保持较大值，增强了算法的全局搜索能力。随着迭代时间的增加，收缩速度逐渐增大， $\vec{A}$  长时间保持较小值，提高了算法的搜索精度，降低了算法陷入局部最优解的可能性。在整个迭代周期内的明显速度差异也可以保护种群的多样性。因此，本文在原有收敛因子计算公式的基础上增加了变异指数  $\mu$ ，取  $\mu = 2$ 。

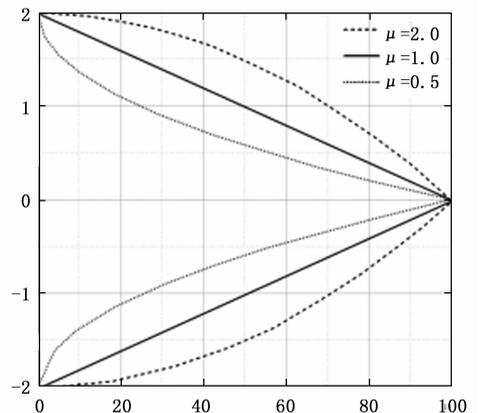


图 4 变异指数迭代变化图

### 2.2.3 改进螺旋更新位置

从式 (11) 中我们能够看到，WOA 算法的螺旋更新策略模型可以帮助种群在后期阶段快速地聚集。虽然它能够

提高算法后期的收敛速度, 但同时也会损失种群的多样性, 从而增大算法陷入局部最优解的风险<sup>[26]</sup>。因此, 本文对螺旋位置更新模型做出如下改进:

$$\vec{X}'(t+1) = \vec{X}^*(t) + \vec{D}' \cdot e^{bl} \cdot \sigma \cdot \cos(2\pi t) + \sin \vec{X}(t) \quad (20)$$

$$\sigma = a - a \cdot \left(\frac{t}{t_{\max}}\right) \quad (21)$$

其中:  $\vec{X}^*(t)$  表示目前最好的位置向量,  $b$  为常数用来定义螺旋线的形状, 一般设置为 1,  $l$  是  $[-1, 1]$  中的随机数,  $\vec{D}'$  表示鲸鱼和猎物的距离, 正弦  $\sin \vec{X}(t)$  在  $[-1, 1]$  的范围进行辅助和干扰, 扩大算法的搜索范围, 使算法更容易搜索全局最优解。 $\sigma$  控制余弦局部开发的搜索区域范围,  $a$  为常数,  $t$  为当前迭代次数,  $t_{\max}$  为最大迭代次数, 随着迭代次数的增加余弦搜索范围逐渐减小, 并收敛到一个最优解。正弦与余弦相互作用最后使整个计算的结果收敛至一个全局最优解, 提高了整个算法的收敛精度, 防止了算法的早熟。

综上所述, 改进鲸群算法流程图如图 5 所示。

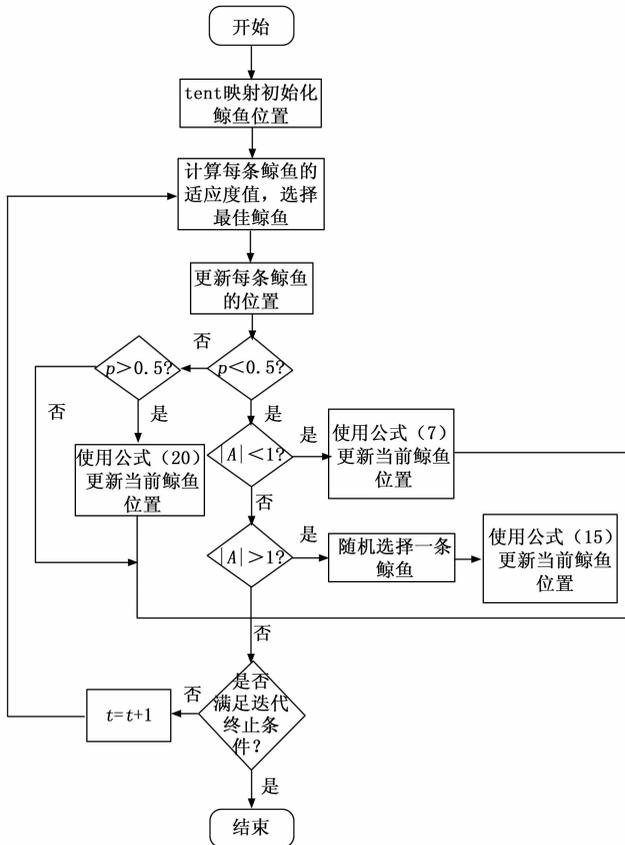


图 5 改进鲸群算法流程图

### 3 测试案例与结果分析

#### 3.1 测试案例

为了证明 IWOA 算法在解决 OCL 问题上的优势, 选择了两个典型的并联冷机系统验证算法性能。在这项研究中, 为了说明 IWOA 算法解决 OCL 问题的最佳能力, 对两个案

例进行了测试。第一个多冷水机组系统由 3 个 800 RT 机组组成。第二个多冷水机组系统由 4 个 1 280 RT 机组和两个 1 250 RT 机组组成。两组冷机工况数据均选取来自台湾新竹科技园内的一家半导体厂的冷机数据进行实验。表 1 展示了两个案例冷机的性能参数。

表 1 测试案例冷机性能参数

案例	冷机编号	a	b	c	d	额定制冷量/RT
案例 1	1	100.950	818.61	-973.430	788.55	800
	2	66.598	606.34	-380.580	275.95	800
	3	130.090	304.50	14.377	99.80	800
案例 2	1	399.345	-122.12	770.46	0	1 280
	2	287.116	80.04	700.48	0	1 280
	3	-120.505	1525.99	-502.14	0	1 280
	4	-19.121	898.76	-98.15	0	1 280
	5	-95.029	1202.39	-352.16	0	1 250
	6	191.750	224.86	524.04	0	1 250

#### 3.2 结果分析

为消除偶然性, 本研究所有实验数据均在对每个实验工况分别进行了 30 次独立实验基础上得到的, 首先将改进鲸群算法用于 OCL 问题的求解。在案例 1 实验中将 IWOA 算法与 GA<sup>[4]</sup>、PSO<sup>[7]</sup>、IFODPSO<sup>[15]</sup> 等其他曾用于求解 OCL 问题的元启发式算法进行对比, 结果如表 2 所示。从表 2 可看出 IWOA 算法在各工况下运行的结果均优于 GA 算法, 在不同负荷需求下可节能 2.84~149.93 kW 不等, 特别是当负荷需求低于 1 440 RT (60%) 时, 均节能超过 100 kW。当冷机在高负荷需求条件下运行时, 节能效率有所下降, 但依然可节能 2.84~7.16 kW。与 PSO、IFODPSO 相比在各种负荷条件下均获得了相似的优化结果。在案例 2 的实验中, 将 IWOA 算法与 GA、PSO、DS 算法进行了对比, 如表 3 所示: 表 3 显示了冷水机组的总功耗和每台冷水机组的部分负荷比, IWOA 算法在不同需求下与 GA、PSO、IFODPSO 算法相比可节能 0.95~159.79 kW。当部分负荷比小于 80% 时, IWOA 算法计算的最小能耗比 GA 算法高 1~4.3%, 比 PSO 算法高 1.6~2.1%, 比 IFODPSO 算法高 1.59~2.19%。在部分负荷比高于 80% 时, IWOA 算法的优化结果与 IFODPSO 算法相同。与其他优化算法相比在统计平均运行时间方面, IWOA 算法优于 PSO 算法和 GA 算法。这是因为提高算法收敛精度和减少运行时间是两个相互矛盾的目标, 但随着计算工具性能的不断提高, 这一矛盾得到了很好的解决<sup>[25]</sup>。图 6 展示了两个不同案例下 IWOA 算法的收敛曲线, 对于案例 1 对应的三台冷机低维问题, IWOA 算法可以保证在 10 代之内快速收敛到最优值, 这是因为通过混沌映射, 将初始化粒子均匀遍布解空间, 以及通过变异指数对收敛因子的改进, 既可以加快算法的收敛速度又可以保证算法不易陷入局部最优解。而面对案例 2 的 6 台冷机对应的高维问题, 算法相应的收敛速度有所减慢, 但在 30 代内算法就能保证收敛到最优值,

表 2 案例 1 中 GA、PSO、IFODPSO、IWOA 算法结果对比

负荷需求	冷机序号	GA		PSO		IFODPSO		IWOA	
		R	总功率 P/kW						
2 160(90%)	1	0.805 1	1 590.96	0.730 1	1 583.81	0.725 2	1 583.80	0.728 0	1 583.80
	2	0.932 3		0.970 0		0.974 8		0.972 1	
	3	0.963 2		1.000		1.000		1.000 0	
1 920(80%)	1	0.701 7	1 406.03	0.660 1	1 403.20	0.659 2	1 403.19	0.666 9	1 403.19
	2	0.795 3		0.860 0		0.858 5		0.853 9	
	3	0.903 4		0.880 0		0.882 3		0.879 3	
1 680(70%)	1	0.690 0	1 250.06	0.600 0	1 244.32	0.600 0	1 244.32	0.595 4	1 244.32
	2	0.678 3		0.740 0		0.740 0		0.744 5	
	3	0.731 7		0.760 0		0.760 0		0.760 1	
1 440(60%)	1	0.521 7	1 107.76	0.000 0	993.60	0.000 0	993.60	0.000 0	993.60
	2	0.740 7		0.890 0		0.896 3		0.884 2	
	3	0.538 1		0.910 0		0.903 7		0.915 8	
1 200(50%)	1	0.488 2	971.21	0.000 0	832.33	0.000 0	832.32	0.000 0	832.32
	2	0.443 7		0.740 0		0.743 0		0.754 4	
	3	0.568 3		0.760 0		0.757 0		0.745 7	
960(40%)	1	0.305 5	842.18	0.000 0	692.25	0.000 0	692.25	0.000 0	692.25
	2	0.318 5		0.570 0		0.536 8		0.578 1	
	3	0.576 4		0.630 0		0.663 2		0.621 9	

表 3 案例 2 中 GA、PSO、IFODPSO、IWOA 算法结果对比

负荷需求	冷机序号	GA		PSO		IFODPSO		IWOA	
		R	总功率 P/kW						
6 858(90%)	1	0.705 2	4 766.33	0.882 6	4 739.53	0.812 7	4 738.58	0.806 3	4 739.02
	2	0.769 2		0.779 9		0.749 6		0.769 5	
	3	0.986 8		0.999 6		1.000 0		1.000 0	
	4	0.986 8		0.999 6		1.000 0		1.000 0	
	5	0.979 3		0.999 8		1.000 0		1.000 0	
	6	0.884 2		0.818 3		0.838 6		0.824 9	
6 477(85%)	1	0.620 7	4 459.16	0.760 6	4 423.04	0.722 6	4 422.65	0.675 7	4 421.63
	2	0.774 2		0.655 5		0.656 3		0.684 7	
	3	0.992 7		1.000 0		1.000 0		1.000 0	
	4	0.958 9		1.000 0		1.000 0		1.000 0	
	5	0.995 6		1.000 0		1.000 0		1.000 0	
	6	0.759 5		0.683 6		0.716 5		0.741 0	
6 096(80%)	1	0.809 9	4 185.87	0.659 1	4 147.69	0.642 7	4 143.71	0.642 3	4 143.71
	2	0.547 4		0.579 8		0.562 5		0.562 0	
	3	0.987 8		0.999 1		1.000 0		1.000 0	
	4	0.962 4		0.997 9		1.000 0		1.000 0	
	5	0.989 7		0.992 1		1.000 0		1.000 0	
	6	0.502 9		0.571 0		0.716 5		0.595 7	
5 717(75%)	1	0.579 7	3 940.60	0.771 3	3 921.07	0.557 7	3 904.75	0.000 0	3 842.55
	2	0.562 1		0.771 7		0.469 2		0.714 8	
	3	0.942 8		0.300 0		1.000 0		1.000 0	
	4	0.790 8		0.999 1		1.000 0		1.000 0	
	5	0.995 1		1.000 0		1.000 0		1.000 0	
	6	0.633 9		0.718 7		0.472 5		0.793 6	
5 335(70%)	1	0.583 1	3 706.22	0.641 8	3 642.55	0.672 6	3 625.77	0.000 0	3 546.43
	2	0.576 7		0.662 1		0.595 5		0.583 1	
	3	0.523 0		0.330 1		0.595 5		1.000 0	
	4	0.949 7		0.990 6		1.000 0		1.000 0	
	5	0.952 1		0.999 0		1.000 0		1.000 0	
	6	0.620 7		0.580 6		0.637 4		0.622 1	

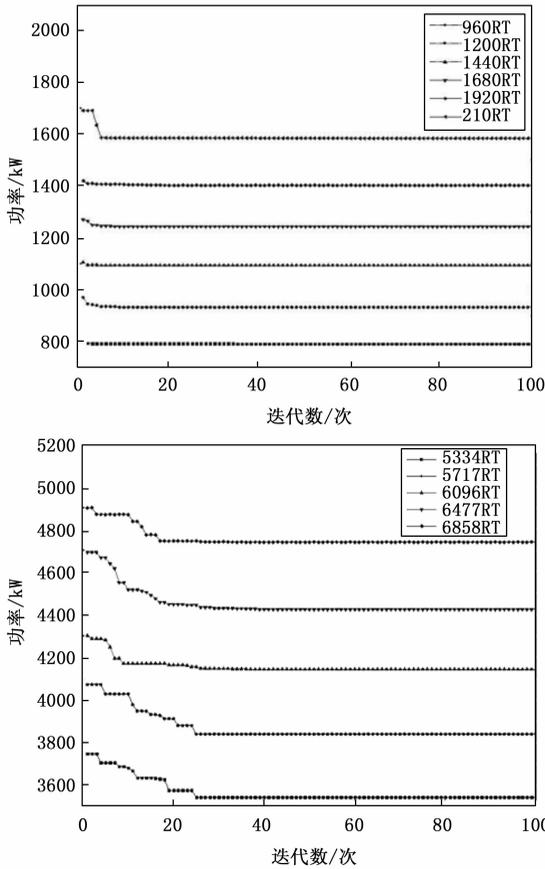


图 6 不同情况下 IWOA 算法收敛曲线

通过实验可以看出算法具有良好的收敛性。从表 4 可以看出, 将案例 1 和案例 2 中的 WOA 算法与 IWOA 算法在每个负荷需求下 30 次独立运行后的最大、最小和平均能耗进行了比较, 并通过标准差验证了能耗的离散程度, 以此来验证算法的稳定性。案例 1 中, 当符合需求低于 60% 时, IWOA 算法标准差为 0, 说明了在低负荷条件下, IWOA 算法运行非常稳定。当符合需求高于 60% 时, 两种算法的标准差均维持在较低水平。说明两种算法在针对只有 3 台冷机的低维优化问题时, 均展现出了算法的稳定性, 而 IWOA 算法相比于 WOA 算法稳定性更佳。案例 2 中, 当负荷需求低于 75% 时, IWOA 算法非常稳定, 标准差为 0, 并在高负荷需求下, 标准差维持在 0.004 3~0.007 3 这一低水平下。而 WOA 算法在案例 2 中的标准差差距较大分别达到了 28.358 7、0.979 4、2.369 1、6.833 6、9.468 7 远远高于相同工况下 IWOA 算法所得结果标准差。由此可见 WOA 算法在面对 6 台冷机对应的高维优化问题时, 容易陷入局部最优解, 使得数据差异明显, 从而导致标准差差距较大。此外, 2 个案例中 IWOA 算法相比于 WOA 算法在不同负荷需求条件下总功率的最小值、最大值、平均值等都要更低, 更加证明了 IWOA 算法的优势。

另外图 7 显示了 WOA 算法和 IWOA 算法在两个案例中 30 次独立实验的平均计算结果与最优预测值的相对误差。

表 4 WOA 算法与 IWOA 算法优化结果对比

案例	负荷需求	优化算法	总功率 P/kW			标准差
			最小值	平均值	最大值	
案例 1	90%	WOA	1 583.80	1 583.82	1 593.90	0.027 5
		IWOA	1 583.80	1 583.81	1 583.85	0.006 9
	80%	WOA	1 403.21	1 403.23	1 403.27	0.017 9
		IWOA	1 403.19	1 403.20	1 403.22	0.006 9
	70%	WOA	1 244.32	1 244.35	1 244.40	0.023 7
		IWOA	1 244.32	1 244.34	1 244.36	0.008 6
	60%	WOA	993.60	993.60	993.62	0.011 4
		IWOA	993.60	993.60	993.60	0.000 0
	50%	WOA	832.32	832.32	832.32	4.625 3e-04
		IWOA	832.32	832.32	832.32	3.584 3e-12
40%	WOA	692.25	692.25	692.25	0.000 0	
	IWOA	692.25	692.25	692.25	0.000 0	
案例 2	90%	WOA	4 738.60	4 743.56	4 769.66	9.468 7
		IWOA	4 738.57	4 738.57	4 738.59	0.004 3
	85%	WOA	4 421.70	4 428.39	4 439.58	6.833 6
		IWOA	4 421.64	4 421.65	4 421.66	0.007 3
	80%	WOA	4 143.80	4 146.32	4 151.19	2.369 1
		IWOA	4 143.70	4 143.71	4 143.72	0.006 6
	75%	WOA	3 842.66	3 843.38	3 845.53	0.979 4
		IWOA	3 842.66	3 842.66	3 842.66	0.000 0
	70%	WOA	3 628.13	3 658.74	3 748.69	28.358 7
		IWOA	3 546.43	3 546.43	3 546.43	0.000 0

首先针对图 (a) 和图 (c) 分别对应的案例 1 的运行结果可看出, WOA 算法相对误差均低于 0.015%, 负荷要求为 40% 和 50% 时, 算法曲线平坦, 负荷要求为 80% 和 90% 时相对误差出现了一定的震荡。而 IWOA 算法在负荷需求为 40%、50%、60% 时曲线均比较平坦, 没有出现明显的误差。负荷需求为 70%、80%、90% 时最大相对误差也不超过 0.003 5%。针对图 (b) 和图 (d) 分别对应的案例 2 的运行结果, WOA 算法的相对误差均低于 5%, 负荷需求为 70% 和 75% 时曲线较平坦。IWOA 算法相对误差均低于 0.000 6%, 负荷需求为 70% 和 75% 时曲线较平坦。图中 IWOA 算法在 30 次独立运行的实验结果相对误差虽然在某些工况下有所变动, 但相比于 WOA 算法误差更小、曲线更加平坦从而验证了算法可靠的稳定性。

通过实验表明, IWOA 算法是解决 OCL 问题的一种行之有效的方法。

#### 4 结束语

通过增加冷却需求和冷却器数量, 每个机组必须以最高效率运行, 并将功耗降至最低。因此, 为了降低建筑物或公司的运营成本, 减少污染物的排放, OCL 问题变得越来越重要。在这项研究工作中, IWOA 算法作为一种有效的优化方法被引入到 OCL 问题中。本文中用于解决两个不同的测试系统, 以评估该方法的有效性和性能。将所提供的最优解与其他优化方法进行了比较, 结果表明, 这些优化方法在功耗方面有了显著的提高。IWOA 算法的收敛

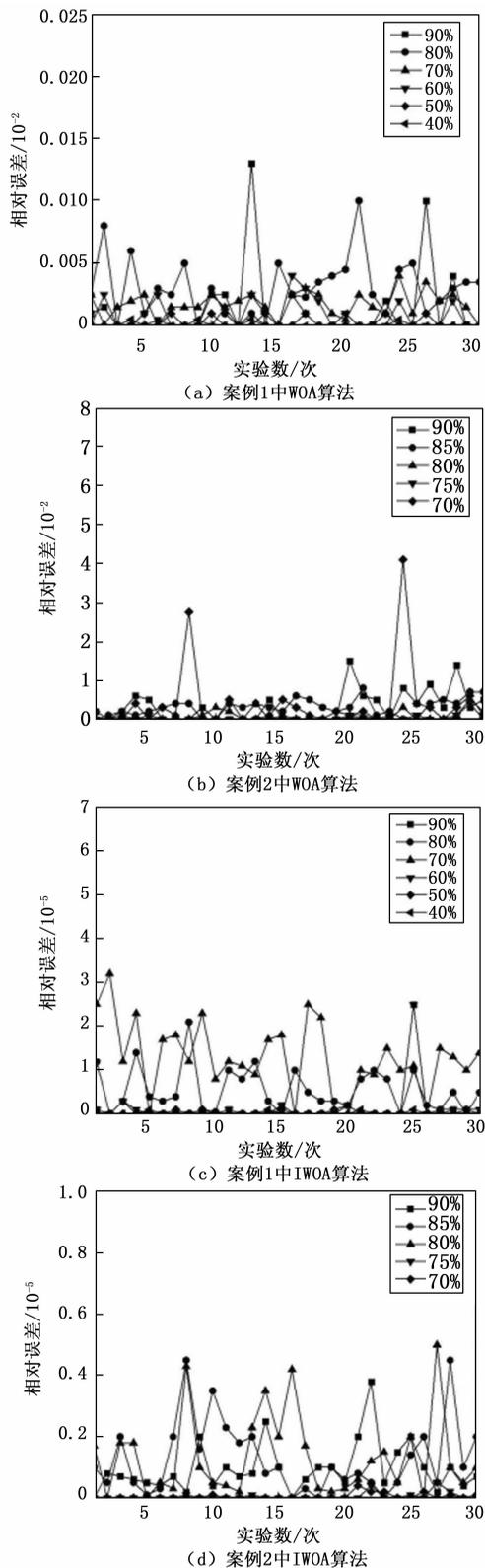


图 7 不同案例下 WOA 算法和 IWOA 算法相对误差

特性表明它能够在较短的时间间隔内获得最优解。该方法可方便地推广应用于大型多冷水机组系统的 OCL 问题。未来的趋势可以集中在研究大规模系统中的 OCL 问题和冷水

机组的每日优化调度上。此外，制冷机满足制冷需求过程中的废热可以回收利用，以满足系统的热需求。

参考文献:

[1] 于军琪, 王 福, 赵安军, 等. 基于改进烟花算法的并联冷机负荷分配优化 [J]. 控制与决策, 2021, 36 (11): 2618 - 2626.

[2] YU F W, CHAN K T. Energy signatures for assessing the energy performance of chillers [J]. Energy and Buildings, 2005, 37 (7): 739 - 746.

[3] CHANG Y C. A novel energy conservation method—optimal chiller loading [J]. Electric Power Systems Research, 2004, 69 (2): 221 - 226.

[4] CHANG Y C. Genetic algorithm based optimal chiller loading for energy conservation [J]. Applied Thermal Engineering, 2005, 25 (17): 2800 - 2815.

[5] CHANG Y. An innovative approach for demand side management—optimal chiller loading by simulated annealing [J]. Energy, 2006, 31 (12): 1883 - 1896.

[6] CHANG Y C, CHAN T S, LEE W S. Economic dispatch of chiller plant by gradient method for saving energy [J]. Applied Energy, 2010, 87 (4): 1096 - 1101.

[7] LEE W S, LIN L C. Optimal chiller loading by particle swarm algorithm for reducing energy consumption [J]. Applied Thermal Engineering, 2009, 29 (8): 1730 - 1734.

[8] JAHANBANI A, Ardakani, F. Fattahi Ardakani, S. H. Hosseini. A novel approach for optimal chiller loading using particle swarm optimization [J]. Energy and Buildings, 2008, 40 (12): 2177 - 2187.

[9] LEE W S, LIN L C, KAO Y C. Optimal chiller loading by differential evolution algorithm for reducing energy consumption [J]. Energy and Buildings, 2011, 43 (2/3): 599 - 604.

[10] L. DOS Santos Coelho, MARIANI V C. Improved firefly algorithm approach applied to chiller loading for energy conservation [J]. Energy Build, 2013, 59: 273 - 278.

[11] L. DOS SANTOS COELHO, KLEIN C E, SABAT S L, et al. Optimal chiller loading for energy conservation using a new differential cuckoo search approach [J]. Energy, 2014, 75: 237 - 243.

[12] LO C C, TSAI S H, LIN B S. Economic dispatch of chiller plant by improved ripple bee swarm optimization algorithm for saving energy [J]. Appl. Therm. Eng, 2016, 100: 1140 - 1148.

[13] ZHENG Z X, LI J P. Optimal chiller loading by improved invasive weed optimization algorithm for reducing energy consumption [J]. Energy and Buildings, 2018, 161: 80 - 88.

[14] ZHENG Z X, LI J Q, DUAN P Y. Optimal chiller loading by improved artificial fish swarm algorithm for energy saving [J]. Mathematics and Computers in Simulation, 2019, 155 (C): 227 - 243.

(下转第 183 页)