

doi:10.3969/j.issn.1673-9833.2024.05.008

# 异构云系统中成本和能耗联合优化的调度方法

艾明慧, 张龙信, 谭润提, 张艳芬

(湖南工业大学 计算机学院, 湖南 株洲 412007)

**摘要:** 为了解决异构云系统(HCS)中成本和能耗联合优化问题, 将群体智能优化算法应用于任务调度问题, 提出了一种基于反向解的白鲨优化算法(RS\_WSO)。RS\_WSO是一种元启发式算法, 包括种群初始化、计算反向解、追踪猎物、寻找猎物阶段。在表观基因组(EP)和高斯消元(GE)两个科学 workflows 进行实验, 结果表明RS\_WSO算法相比当前先进的元启发式算法, 在节约成本、减少能量消耗方面, 具有明显优势。

**关键词:** 异构云系统; 能耗; 成本; 元启发式算法; 白鲨优化算法

**中图分类号:** TP311

**文献标志码:** A

**文章编号:** 1673-9833(2024)05-0055-07

**引文格式:** 艾明慧, 张龙信, 谭润提, 等. 异构云系统中成本和能耗联合优化的调度方法[J]. 湖南工业大学学报, 2024, 38(5): 55-61.

## A Scheduling Method for Joint Optimization of Cost and Energy Consumption in Heterogeneous Cloud System

AI Minghui, ZHANG Longxin, TAN Runti, ZHANG Yanfen

(College of Computer Science, Hunan University of Technology, Zhuzhou Hunan 412007, China)

**Abstract:** In view of a joint optimization of cost and energy consumption in heterogeneous cloud systems (HCS), a swarm intelligence optimization algorithm is applied for task scheduling problems, with a reverse solution based white shark optimization algorithm (RS\_WSO) to be proposed. As a metaheuristic algorithm, RS\_WSO includes population initialization, calculation of reverse solutions, prey tracking and hunting. Experiments are carried out in two scientific workflows, involving epigenome (EP) and Gauss elimination (GE). The results show that RS\_WSO algorithm is characterized with clear advantages in terms of cost saving and energy consumption compared with the current advanced meta-heuristic algorithm.

**Keywords:** heterogeneous cloud system; energy consumption; cost; meta-heuristic algorithm; WSO

### 1 研究背景

随着人工智能和物联网的发展, 万物互联的时

代已经到来, 这些技术被广泛应用于智能家居、城市、交通等领域<sup>[1]</sup>, 为人们带来了更加便利和高效的生活方式。与此同时, 每日产生的数据量剧增,

收稿日期: 2023-06-24

基金项目: 国家重点研发计划基金资助专项子课题(2018YFB1003401); 湖南省自然科学基金资助项目(2023JJ50204, 2024JJ7154); 湖南省教育厅科研基金资助项目(23B0560)

作者简介: 艾明慧, 女, 湖南工业大学硕士生, 主要研究方向为高性能计算, E-mail: minghui\_ai@163.com

通信作者: 张龙信, 男, 湖南工业大学副教授, 博士, 硕士生导师, 主要研究方向为大数据处理, 高性能计算, E-mail: longxin.zhang@163.com

对海量数据的处理给高性能计算中心提出了更高的要求<sup>[2]</sup>。异构云系统（heterogeneous cloud system, HCS）中任务调度问题是 NP-Hard 问题<sup>[3]</sup>，解决方法通常分为启发式算法、元启发式算法和混合算法。其中，元启发式算法被证明是避免陷入局部最优解的最有效方法<sup>[4]</sup>。

元启发式算法中的群体智能优化算法，是一种基于群体智能的优化算法，它模拟了自然界中群体行为过程，通过群体协作和竞争来搜索最优解<sup>[5]</sup>。常见的群体智能优化算法包括：狮群算法<sup>[6]</sup>、蝙蝠算法<sup>[7]</sup>、人工蜂群算法<sup>[8]</sup>、鲸鱼优化算法（whale optimization algorithm, WOA）<sup>[9]</sup>和粒子群优化（particle swarm optimization, PSO）算法<sup>[10]</sup>等。

近年来，为解决 HCS 的任务调度问题，学者们围绕群体智能优化算法进行了大量的研究<sup>[11]</sup>。Zhang L. X. 等<sup>[12]</sup>提出能量最小化鲸鱼优化算法（energy minimization whale optimization algorithm, EM\_WOA）。EM\_WOA 是一种基于多元宇宙优化和鲸鱼优化算法相结合的优化算法，在满足成本约束的条件下，最小化 workflow 应用程序的能耗。EM\_WOA 能够有效避免局部最优，但是优化目标单一。P. Shukla 等<sup>[13]</sup>提出的多目标人工藻类算法，是一种基于藻类生长规律的多目标优化算法。在预处理阶段，根据子代的数量将任务分离到单独的列表中，以节约资源，同时优化调度长度、能耗和成本。但是，人工藻类算法还存在收敛速度慢、容易陷入局部最优的问题。M. Braik 等<sup>[14]</sup>提出了白鲨优化算法（white shark optimizer, WSO），该算法通过模拟白鲨狩猎行为来寻找最优解。WSO 具有很强的全局搜索能力，但是需要耗费大量的计算资源，时延较高。S. Mirjalili 等<sup>[15]</sup>提出的灰狼优化算法是一种基于自然灰狼族群行为的全局优化算法。该算法模仿了灰狼族群中的领袖、副领袖和普通狼的行为，通过不断地迭代来寻找最优解。由于灰狼种群等级众多，所以在迭代过程中，时延较高。

在资源受限的情况下，HCS 中的任务调度与优化是高性能领域研究的热门话题。5G 时代的到来，给人们的生产生活提供了便利，与此同时，海量的数据也给资源的分配带来了新的问题，如处理效率低、能耗大、成本高等。为了解决在预算成本约束下，同时降低能耗、节约成本的问题，本文提出一种异构云中成本和能耗联合优化的调度方法，名为基于反向解的白鲨优化算法（reverse solution-based white shark optimization, RS\_WSO）。模拟白鲨的捕食行为，并利用反向解扩大搜索空间，避免陷入局部最优。在满

足成本约束的前提下，实现能耗和成本的联合优化。

## 2 模型构建

### 2.1 系统和应用模型

HCS 模型是指由不同类型的计算资源（如 CPU、GPU、FPGA 等）组成的云计算系统。这种系统模型可以更好地满足不同应用场景下的计算需求，提高系统的计算性能和效率。HCS 中的任务调度模型则是在满足任务执行约束条件的前提下，使得任务的执行时间最短或者资源的利用率最高。

本文的目标计算资源由一组虚拟机  $U=\{u_1, u_2, \dots, u_U\}$  集合组成，其中  $U$  为虚拟机的数量。HCS 中的 workflow 应用模型通常由一组有向无环图（directed acyclic graph, DAG）来描述<sup>[15]</sup>，即  $G=\{T, E, W, C\}$ ，其中， $T=\{t_1, t_2, \dots, t_N\}$  表示有  $N$  个节点的工作流集合； $E$  表示 DAG 中具有数据依赖关系的边的集合； $C$  表示具有依赖关系的任务不在同一台虚拟机上执行时，数据传输所需通信时间的集合； $W$  是任务分配到一个虚拟机的计算开销的集合。一个简单的由 10 个任务节点组成的 DAG 示例图见图 1。

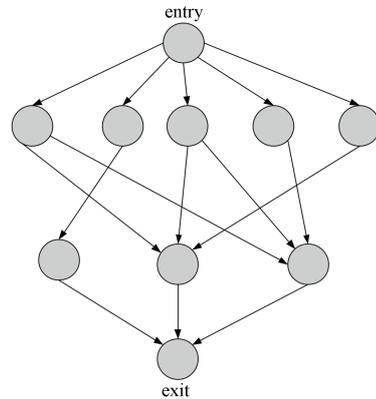


图 1 DAG 示例图

Fig. 1 DAG illustration diagram

### 2.2 能耗模型

动态电压频率调整技术（dynamic voltage and frequency scaling, DVFS），可以根据虚拟机的工作负载动态地调整电压和频率，以提高能效。DVFS 技术通过调整虚拟机的电压和频率，使虚拟机在不同负载下以最佳速度运行，从而提升其性能。当频率为  $f$  时，能耗的表达式为

$$P(f) = P_s + h(P_{\text{ind}} + C_{\text{ef}} f^m), \quad (1)$$

式中： $P_s$  为静态功率，本研究中忽略不计； $h$  为系统状态，表示当前系统是否消耗动态功率； $P_{\text{ind}}$  为频变动态功率； $C_{\text{ef}}$  为有效电容开关； $m$  为动态功率指标，其值为不小于 2 的常数。

虚拟机的节能频率  $f_{ee}$  表示为

$$f_{ee} = \sqrt[m]{\frac{P_{ind}}{(m-1)C_{ef}}}。 \quad (2)$$

虚拟机的最低频率  $f_{low}$  为最小频率  $f_{min}$  到节能频率  $f_{ee}$  之间的最大值, 表示为  $f_{low} = \max\{f_{min}, f_{ee}\}$ 。因此, 每个任务的实际有效频率  $f_h$  必须为  $f_{low} \leq f_h \leq f_{max}$ 。

任务  $t_i$  在虚拟机  $u_k$  上运行, 当频率为  $f_{k,h}$  时, 产生的能耗  $E(t_i, u_k, f_{k,h})$  的表达式为

$$E(t_i, u_k, f_{k,h}) = (P_{k,ind} + C_{k,ef} \times f_{k,h}) \times w_{i,k} \times \frac{f_{k,max}}{f_{k,h}}。 \quad (3)$$

式中  $w_{i,k}$  为任务  $t_i$  在虚拟机  $u_k$  上的执行时间。

### 2.3 成本模型

DAG 的成本模型与当前的商业云类似, 根据用户使用虚拟机的时间来收费。在 HCS 中, 每个虚拟机的单价不同。假设虚拟机  $u_k$  的单价为  $price_k$ , 不考虑通信成本, 因此任务  $t_i$  在频率为  $f_{k,h}$  的虚拟机  $u_k$  上的执行成本  $cost(t_i)$  记为

$$cost(t_i) = w_{i,k} \times price_k \times \frac{f_{k,max}}{f_{k,h}}。 \quad (4)$$

任务的最小执行成本计算如式 (5) 所示, 最大执行成本计算如式 (6) 所示。

$$cost_{min}(t_i) = \min_{u_k \in U} cost(t_i, u_k, f_{k,max}), \quad (5)$$

$$cost_{max}(t_i) = \max_{u_k \in U} cost(t_i, u_k, f_{k,low})。 \quad (6)$$

DAG 应用程序的总执行成本计算如式 (7) 所示, 最小执行总成本如式 (8) 所示, 最大执行总成本如式 (9) 所示。

$$cost(G) = \sum_{i=0}^N cost(t_i), \quad (7)$$

$$cost_{min}(G) = \sum_{i=0}^N cost_{min}(t_i), \quad (8)$$

$$cost_{max}(G) = \sum_{i=0}^N cost_{max}(t_i)。 \quad (9)$$

用户提交的 DAG 应用的成本预算  $cost_{bud}(G)$  必须大于或等于  $cost_{min}(G)$  且小于或等于  $cost_{max}(G)$ , 即  $cost_{min}(G) \leq cost_{bud}(G) \leq cost_{max}(G)$ 。

## 3 RS\_WSO 算法

### 3.1 适应度函数

本文将白鲨优化算法应用于 HCS 中的调度问题。为了在满足成本约束的情况下, 能够同时优化 workflow 应用程序的成本和能耗, 本文将整个算法的适应度函

数定义如下:

$$F = f_1 \times cost(G) + f_2 \times E(G)。 \quad (10)$$

式中:  $f_1$  和  $f_2$  分别为  $[0, 1]$  中的随机值, 且  $f_1 + f_2 = 1$ 。

### 3.2 编码

任务与虚拟机的编码过程如图 2 所示, 调度解由种群中每条白鲨的位置表示。Task-Ins 表示任务到实例的映射数组, 其中索引表示任务, 其值表示将执行该任务的实例。例如,  $Task-Ins[i]=j$  表示将  $t_i$  分配给索引为  $j$  的实例。白鲨的位置  $Position[i, j]$ , 是一个反映任务和虚拟机之间相关性 (也即实例到实例类型的映射) 的数组。这个数组的维度等于 workflow 中的任务总数。每个元素表示执行相应任务的虚拟机, 索引表示实例。例如,  $Position[0, 3]=0$  表示任务  $t_4$  将在编号为 0 的虚拟机上执行。数组中的最小值是 0, 而最大值是  $|U|-1$  (虚拟机数量)。

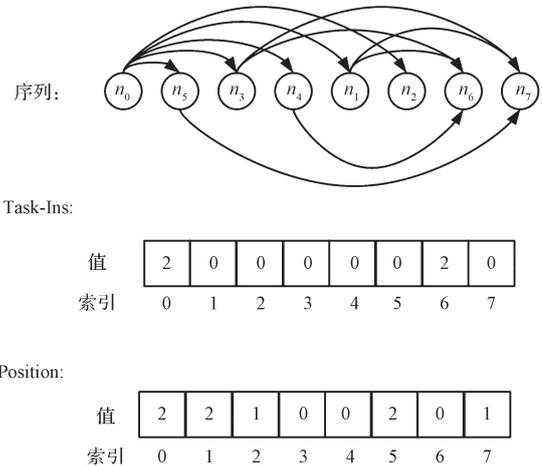


图 2 任务与虚拟机的编码过程

Fig. 2 Encoding process of tasks and VMs

### 3.3 WSO 算法

白鲨有着令人惊讶的听觉, 可利用它来侦查空间、寻找猎物。同时, 白鲨的嗅觉敏锐, 能够闻到猎物的气味。这些特性帮助它们探索整个空间, 并利用搜索域的每个可能区域来追捕猎物。

1) 种群初始化。WSO 算法通过随机生成初始化种群开始。种群大小为  $n$  的白鲨群, 在  $d$  维搜索空间中的位置可以描述为一个二维矩阵  $w$ , 如式 (11) 所示。

$$w = \begin{bmatrix} w_1^1 & w_2^1 & \cdots & \cdots & w_d^1 \\ w_1^2 & w_2^2 & \cdots & \cdots & w_d^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_1^n & w_2^n & \cdots & \cdots & w_d^n \end{bmatrix}, \quad (11)$$

式中:  $w$  为搜索空间中所有白鲨的位置;  $d$  为种群的维度;  $w_d^i$  为第  $d$  维中第  $i$  条白鲨的位置, 每一个位置对应一个候选解的方案。

初始化种群的位置创建如下:

$$\mathbf{w}_j^i = l_j + r \times (u_j - l_j). \quad (12)$$

式中： $\mathbf{w}_j^i$ 为第*j*维中第*i*条白鲨的初始位置向量； $u_j$ 和 $l_j$ 分别表示第*j*维搜索空间的上界和下界； $r$ 为区间[0, 1]中的随机数。

2) 接近猎物的移动速度。当一条白鲨感知到猎物的位置时，它会根据猎物移动时产生的波浪来判断猎物的位置，并通过波动运动向猎物靠近。这种波动运动定义如式(13)所示。

$$\mathbf{v}_{k+1}^i = \mu \left[ \mathbf{v}_k^i + p_1 (\mathbf{w}_{\text{gbest}_k} - \mathbf{w}_k^i) \times c_1 + p_2 (\mathbf{w}_{\text{best}}^{v_i} - \mathbf{w}_k^i) \times c_2 \right]. \quad (13)$$

式中： $\mathbf{v}_k^i$ 为第*i*条白鲨在*k*步骤中的当前速度向量； $\mathbf{w}_{\text{gbest}_k}$ 为白鲨第*k*次迭代获得的最佳位置向量； $\mathbf{w}_k^i$ 为第*i*条白鲨在*k*步骤中获得的当前位置向量； $\mathbf{w}_{\text{best}}^{v_i}$ 为第*i*条最佳位置向量鲨鱼群； $\mathbf{v}^i$ 为拥有最佳位置白鲨的第*i*个速度向量，计算公式由式(14)定义； $c_1$ 和 $c_2$ 为[0, 1]范围内均匀创建的两个随机数； $p_1$ 和 $p_2$ 分别表示控制 $\mathbf{w}_{\text{gbest}_k}$ 和 $\mathbf{w}_{\text{best}}^{v_i}$ 对 $\mathbf{w}_k^i$ 影响的因子，计算结果如式(15)和式(16)所示。

$$\mathbf{v} = \lfloor n \times \text{rand}(1, n) \rfloor + 1. \quad (14)$$

$$p_1 = p_{\max} + (p_{\max} - p_{\min}) \times e^{-(4k/K)^2}, \quad (15)$$

$$p_2 = p_{\min} + (p_{\max} - p_{\min}) \times e^{-(4k/K)^2}, \quad (16)$$

式(15)(16)中： $k$ 和 $K$ 分别为当前迭代次数和最大迭代次数； $p_{\min}$ 和 $p_{\max}$ 分别为白鲨获得良好运动的初始速度和从属速度。

经过严格分析， $p_{\min}$ 和 $p_{\max}$ 值分别为0.5和1.5。 $\mu$ 为WSO中建议的缩窄因子，用于分析白鲨的收敛行为，定义如式(17)所示， $\tau$ 为加速度系数，取值4.125。

$$\mu = \frac{2}{\left| 2 - \tau - \sqrt{\tau^2 - 4\tau} \right|}. \quad (17)$$

3) 寻找猎物。白鲨的大部分时间都在寻找潜在猎物，在潜在猎物中可以找到最佳或次优猎物。因此，白鲨的位置会不断变化。当听到猎物移动引起的波浪或闻到猎物的气味时，它们会向猎物靠近。猎物离开它的位置通常是因为白鲨移动到它身边，或是为了寻找食物。然而，猎物会把气味留在那个位置。在这种情况下，白鲨随机位置导航以寻找猎物，就像鱼群寻找食物源的行为一样。此时，用式(18)中定义的位置更新策略描述白鲨向猎物移动时的行为。

$$\mathbf{w}_{k+1}^i = \begin{cases} \mathbf{w}_k^i \cdot \neg \oplus \mathbf{w}_0 + u \cdot \mathbf{a} + l \cdot \mathbf{b}, & \text{rand} < mv; \\ \mathbf{w}_{k+1}^i = \mathbf{w}_k^i + \mathbf{v}_k^i / f, & \text{rand} \geq mv. \end{cases} \quad (18)$$

式中： $\mathbf{w}_{k+1}^i$ 为第*k+1*次迭代步骤中第*i*条白鲨的新位

置向量， $\neg$ 为否定运算符； $\text{rand}$ 为均匀分布在范围[0, 1]内的随机数； $\mathbf{a}$ 和 $\mathbf{b}$ 为一维二元向量，定义分别为式(19)和式(20)，

$$\mathbf{a} = \text{sgn}(\mathbf{w}_k^i - u) > 0, \quad (19)$$

$$\mathbf{b} = \text{sgn}(\mathbf{w}_k^i - l) < 0; \quad (20)$$

$l$ 和 $u$ 分别表示上下限制的搜索空间； $\mathbf{w}_0$ 为一个逻辑向量，其定义如式(21)所示，

$$\mathbf{w}_0 = \oplus(\mathbf{a}, \mathbf{b}); \quad (21)$$

“ $\oplus$ ”为逐位异或操作； $f$ 为白鲨的波浪运动频率，定义如式(22)所示，

$$f = f_{\min} + \frac{f_{\max} - f_{\min}}{f_{\max} + f_{\min}}. \quad (22)$$

其中 $f_{\min}$ 和 $f_{\max}$ 分别为波浪运动的最小频率和最大频率，本文所处理问题的 $f_{\min}$ 和 $f_{\max}$ 值分别为0.07和0.75； $mv$ 代表运动力随着迭代次数、白鲨接近猎物的距离增加，其表达式为

$$mv = \frac{1}{\left( a_0 + e^{(K/2-k)/a_1} \right)}. \quad (23)$$

其中 $a_0=6.25$ ， $a_1=100$ 。

4) 追踪猎物。白鲨可以在靠近猎物的最佳位置保持自己的位置，这种行为的表述如式(24)所示。

$$\mathbf{w}_{k+1}^i = \mathbf{w}_{\text{gbest}_k} + r_1 \overline{D_w} \text{sgn}(r_2 - 0.5), \quad r_3 < s_s. \quad (24)$$

式中： $\mathbf{w}_{k+1}^i$ 为第*i*条白鲨相对于猎物位置(当*i*≠1时，用 $\mathbf{w}_{k+1}^i$ 表示)； $\text{sgn}(r_2-0.5)$ 给出1或-1来改变搜索方向；变量 $r_1$ 、 $r_2$ 和 $r_3$ 为[0, 1]的随机数； $\overline{D_w}$ 为猎物(即食物源)与白鲨的距离，定义如式(25)所示； $s_s$ 为一个参数，用来表示白鲨跟随其他接近最佳猎物的白鲨时嗅觉和视觉的强度，定义如式(26)所示。

$$\overline{D_w} = \left| \text{rand} \times (\mathbf{w}_{\text{gbest}_k} - \mathbf{w}_k^i) \right|, \quad (25)$$

式中 $\mathbf{w}_k^i$ 为白鲨相对于 $\mathbf{w}_{\text{gbest}_k}$ 的当前位置。

$$s_s = \left| 1 - e^{(-a_2 \times k/K)} \right|, \quad (26)$$

式中， $a_2=0.0005$ 。

5) 鱼群行为。为了从数学上模拟白鲨群的捕猎行为，保留了前两个最优解，并根据这些最优位置更新其他白鲨的位置。提出的定义白鲨的鱼群行为公式如下：

$$\mathbf{w}_{k+1}^i = \frac{\mathbf{w}_k^i + \mathbf{w}_{k+1}^i}{2 \times \text{rand}}. \quad (27)$$

### 3.4 基于反向解的WSO优化算法

为了解决WSO算法收敛速度较低，提高寻优精度，避免局部最优，本文基于WSO算法提出了一种

基于反向解的 WSO 优化算法, 名为 RS\_WSO。下面给出反向解的定义。

反向解: 设  $W_i = \{w_{i,1}, w_{i,2}, \dots, w_{i,n}\}$  为白鲨群的随机选择生成的个体, 其反向解  $w_{i,j}^e$  的定义如下:

$$w_{i,j}^e = k(l_j + u_j) - w_{i,j} \quad (28)$$

式中:  $l_j$  和  $u_j$  为区间上的随机数, 分别为第  $j$  维搜索空间的上界和下界。由于精英个体位置更新后, 存在反向解脱离解空间的情况, 产生不可行解。因此, 可以用式 (29) 重置反向解。

$$w_{i,j}^e = rand(l_j, u_j) \quad (29)$$

RS\_WSO 算法的伪代码如算法 1 所示。

算法 1 WSO optimization algorithm based on reverse solution

输入:  $G = \{T, E, W, C\}, U = \{u_1, u_2, \dots, u_0\}$

输出: 最优解及相关参数

```

1: 初始化白鲨优化算法的相关参数;
2: 生成规模为  $N$  的种群并计算其适应度;
3: 选取距离猎物最近的种群并计算其反向解;
4: 初始化白鲨的移动速度并构建初始种群;
5: while  $k < K_{\max}$  do
6: 更新所有参数;
7: for  $i=1$  to  $n$  do
8: 根据式 (13) 计算白鲨速度;
9: end for
10: for  $i=1$  to  $n$  do
11: if  $rand < mv$  then
12:  $w_{k+1}^i = w_k^i \cdot \ominus \oplus w_0 + u \cdot a + l \cdot b$ ;
13: else
14:  $w_{k+1}^i = w_k^i + v_k^i / f$ ;
15: end if
16: end for
17: for  $i=1$  to  $n$  do
18: if  $rand \leq s_s$  then
19: 根据式 (25) 计算  $\overline{D}_w$ ;
20: if  $i=1$  then
21:  $w_{k+1}^i = w_{gbest_k} + r_1 \overline{D}_w \operatorname{sgn}(r_2 - 0.5)$ 
22: else
23:  $w_{k+1}^i = w_{gbest_k} + r_1 \overline{D}_w \operatorname{sgn}(r_2 - 0.5)$ 
24: 根据式 (27) 更新最优的白鲨位置
25: end if
26: end if
27: end for
28: 计算种群适应度并更新最佳白鲨位置;
29:  $k=k+1$ 
30: end while
31: return 最优解

```

首先初始化整个算法的相关参数。步骤 2~4 计

算适应度, 并利用反向解策略生成初始化种群。步骤 5~30 是算法的核心, 用于更新白鲨的位置并寻找最优解。步骤 7~9 计算白鲨的速度。步骤 10~16, 白鲨向着最佳猎物移动, 此时根据运动因子  $mv$  更新白鲨的位置。步骤 17~27, 种群中其他的白鲨向着位置最好的白鲨移动, 并出现鱼群行为, 以此更新白鲨的位置。步骤 28 计算位置更新后的种群适应度, 并更新白鲨的最佳捕食位置。步骤 29 迭代次数加 1。最后返回最优解。

### 3.5 RS\_WSO 算法时间复杂度分析

RS\_WSO 算法的种群规模为  $n$ , 初始解的空间维度为  $d$ , 最大迭代次数为  $K_{\max}$ 。基于反向解的初始化种群阶段的时间复杂度取决于白鲨群的种群规模和空间维度, 因此, 这个阶段的时间复杂度为  $O(n \times d)$ 。第 5~30 行的时间复杂度为  $O(K_{\max} \times n \times d)$ , 因此, 整个算法的时间复杂度为  $O(K_{\max} \times n \times d)$ 。

## 4 实验

### 4.1 实验设置及评价指标

本文选取著名的 WSO、WOA 和 PSO 算法作为对比。实验中模拟的 HCS 中具有 3 种单价和性能各异的虚拟机, 单价如表 1 所示, 其主要参数如下:  $10 \text{ ms} \leq w_{i,k} \leq 100 \text{ ms}$ ,  $0.03 \leq P_{k,\text{ind}} \leq 0.07$ ,  $2.5 \leq m_k \leq 3.0$ ,  $\varphi_{\{k,\text{max}\}} = 1.0 \text{ GHz}$ 。

表 1 不同 VM 的单价

Table 1 Unit prices of different VMs

$u_k$	$u_1$	$u_2$	$u_3$
price	3	5	4

实验采用两种 workflow 应用程序, 分别取两种算法在相同的预算成本下进行 50 次调度, 每次调度最大迭代次数  $K_{\max} = 100$ , 预算因子  $a = 3$ 。

为保证实验结果的真实性和准确性, 使用标准化成本 (normalized cost, NC) 和标准化能耗 (normalized energy consumption, NEC) 作为实验的评价指标。NC 的值越低, 算法的性能越好, NEC 同理。

### 4.2 epigenomics 并行应用

如图 3 所示, 这是一个典型的表观基因组学 (epigenomics, EP) workflow 应用程序<sup>[16]</sup>。它由多个不同的数据块组成。

表 2 显示了 RS\_WSO、WSO、WOA 和 PSO 算法在 5 种不同规模的 EP workflow 中的 NEC 值。随着节点数量的增加, RS\_WSO、WSO、WOA 和 PSO 算法的 NEC 值也增加。当  $n \leq 103$  时, 4 种算法的

NEC 值均不超过 1 000，然而，当  $n > 500$  时，NEC 值大于 3 000。当节点数为 523 时，RS\_WSO 在节省能耗方面比 WSO、WOA 和 PSO 分别提升了 4.67%、19.50% 和 27.90%。RS\_WSO 在节约能耗方面优于 WSO、WOA 和 PSO 算法。

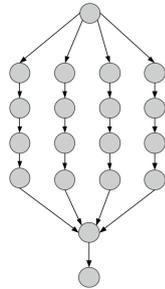


图 3 EP workflow 应用程序  
Fig. 3 EP workflow application

表 2 EP workflow 中不同规模下 4 种算法的标准化能耗  
Table 2 Standarized energy consumption of four algorithms at different scales in EP workflow

数据集大小 (EP)	RS_WSO (NEC)	WSO (NEC)	WOA (NEC)	PSO (NEC)
19	128.63	140.01	156.99	173.35
51	357.80	397.69	417.27	471.33
103	791.76	836.50	990.71	998.54
523	3 693.47	3 874.48	4 588.21	5 122.90
1 003	7 038.68	8 435.71	9 211.99	9 631.47

EP workflow 中不同规模下的 RS\_WSO、WSO、WOA 和 PSO 算法的 NC 值如表 3 所示。当  $n \leq 103$  时，4 种算法的 NC 值均不大于 1 000。以  $n=1 003$  为例，RS\_WSO 算法比 WSO、WOA 和 PSO 算法在标准化成本方面的优化率分别提高 11.90%、24.30% 和 61.10%，在降低成本方面表现出色。

表 3 EP workflow 中不同规模下 4 种算法的标准化成本  
Table 3 Standarization costs of four algorithms at different scales in EP workflow

数据集大小 (GP)	RS_WSO (NC)	WSO (NC)	WOA (NC)	PSO (NC)
19	61.82	69.16	75.30	162.60
51	176.08	190.01	207.70	441.33
103	402.20	420.45	502.07	934.81
523	1 916.17	2 011.72	2 401.32	4 814.42
1 003	3 522.12	3 997.93	4 652.63	9 054.29

### 4.3 高斯消元并行应用

高斯消元 (Gaussian elimination, GE) workflow 应用程序<sup>[16]</sup>如图 4 所示。GE 中任务节点的数量 ( $|N|$ ) 与任务节点层数满足： $|N| = S_\lambda = (\lambda^2 + \lambda - 2) / 2$ ，其中  $\lambda$  为高斯消元矩阵的维数。

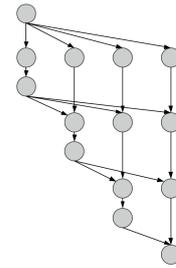


图 4 GE workflow 应用程序  
Fig. 4 GE workflow application

表 4 展示了 RS\_WSO、WSO、WOA 和 PSO 算法在不同规模下的 GE workflow 中的标准化能耗。当  $n=298$  时，4 种算法的 NEC 均超过了 2 000。随着数据集规模扩大，NEC 的值也不断增加。以节点数为 1 033 为例，此时 RS\_WSO 算法在 NEC 方面比 WSO、WOA 和 PSO 算法分别减少了 6.26%、26.86% 和 27.98%。不论哪种规模的数据集，RS\_WSO 算法在降低能耗方面都比 PSO 算法提升了 24.00% 以上，表现出明显的优势。

表 4 GE workflow 中不同规模下 4 种算法的标准化能耗  
Table 4 Standarized energy consumption of four algorithms at different scales in GE workflow

数据集大小 (GE)	RS_WSO (NEC)	WSO (NEC)	WOA (NEC)	PSO (NEC)
27	197.48	230.68	237.47	274.13
53	377.64	406.04	464.51	513.73
298	2 143.46	2 543.45	2 644.63	2 828.82
494	3 603.25	4 352.24	4 525.00	4 819.76
1 033	6 980.96	7 447.27	9 545.00	9 693.09

RS\_WSO、WSO、WOA 和 PSO 算法在 GE workflow 中不同规模下的标准化成本如表 5 所示。

表 5 GE workflow 中不同规模下 4 种算法的标准化成本  
Table 5 Standarization costs of four algorithms at different scales in GE workflow

数据集大小 (GE)	RS_WSO (NC)	WSO (NC)	WOA (NC)	PSO (NC)
27	100.83	121.45	113.77	258.10
53	193.94	201.44	235.72	482.22
298	1 078.87	1 315.38	1 389.45	2 649.53
494	1 811.92	1 937.86	2 227.83	4 530.93
1 033	3 581.34	4 095.22	4 707.79	8 996.07

以  $n=298$  为例，此时 RS\_WSO 算法的 NC 值为 1 078.87，而 WSO、WOA 和 PSO 算法的 NC 值分别为 1 315.38、1 389.45 和 2 649.53，RS\_WSO 算法比 WSO、WOA 和 PSO 算法的 NC 分别降低了 17.98%、22.35% 和 59.28%。在任务数量达到 1 033 时，RS\_WSO 算法比 WSO、WOA 和 PSO 算法的 NC 分别降低了 12.55%、23.93% 和 60.19%。实验结果表明，

RS\_WSO 算法在降低成本方面有明显的优势。

## 5 结语

针对 HCS 中预算成本受限, 联合优化成本和能耗的问题, 本文提出了一种基于反向解的白鲨优化算法。并将公共的 EP 和 GE workflow 应用程序作为测试数据集, 与著名的 WSO、WOA 和 PSO 算法进行对比, 所得实验结果表明, 在 NEC 和 NC 方面, RS\_WSO 算法比 WSO 算法分别提升了 4.67%~17.21% 和 3.72%~17.98%; 比 WOA 算法分别提升了 14.25%~26.86% 和 11.37%~24.30%; 比 PSO 算法分别提升了 20.71%~27.98% 和 56.98%~61.98%。RS\_WSO 算法在联合优化成本和能耗方面具有明显优势, 在节约成本的同时可以降低能耗。

### 参考文献:

- [1] LI K Q. Profit Maximization in a Federated Cloud by Optimal Workload Management and Server Speed Setting[J]. IEEE Transactions on Sustainable Computing, 2022, 7(3): 668-680.
- [2] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J]. 计算机研究与发展, 2019, 56(1): 69-89.  
SHI Weisong, ZHANG Xingzhou, WANG Yifan, et al. Edge Computing: State-of-the-Art and Future Directions[J]. Journal of Computer Research and Development, 2019, 56(1): 69-89.
- [3] KAVANAGH R, DJEMAME K, EJARQUE J, et al. Energy-Aware Self-Adaptation for Application Execution on Heterogeneous Parallel Architectures[J]. IEEE Transactions on Sustainable Computing, 2020, 5(1): 81-94.
- [4] GUPTA K, DEEP K, BANSAL J C. Spider Monkey Optimization Algorithm for Constrained Optimization Problems[J]. Soft Computing, 2017, 21(23): 6933-6962.
- [5] ZHENG Z D, YANG S X, GUO Y N, et al. Meta-Heuristic Techniques in Microgrid Management: A Survey[J]. Swarm and Evolutionary Computation, 2023, 78: 101256.
- [6] 刘生建, 杨 艳, 周永权. 一种群体智能算法: 狮群算法[J]. 模式识别与人工智能, 2018, 31(5): 431-441.  
LIU Shengjian, YANG Yan, ZHOU Yongquan, A Swarm Intelligence Algorithm-Lion Swarm Optimization[J]. Pattern Recognition and Artificial Intelligence, 2018, 31(5): 431-441.
- [7] SAGNIKA S, BILGAIYAN S, MISHRA B S P. Workflow Scheduling in Cloud Computing Environment Using Bat Algorithm[C]//Proceedings of First International Conference on Smart System, Innovations and Computing. Singapore: Springer, 2018: 149-163.
- [8] GOMATHI B, KRISHNASAMY K, BALAJI B S. Epsilon-Fuzzy Dominance Sort-Based Composite Discrete Artificial Bee Colony Optimisation for Multi-Objective Cloud Task Scheduling Problem[J]. International Journal of Business Intelligence and Data Mining, 2018, 13(1/2/3): 247-266.
- [9] MIRJALILI S, LEWIS A. The Whale Optimization Algorithm[J]. Advances in Engineering Software, 2016, 95(C): 51-67.
- [10] AJEENA BEEGOM A S, RAJASREE M S. Integer-PSO: A Discrete PSO Algorithm for Task Scheduling in Cloud Computing Systems[J]. Evolutionary Intelligence, 2019, 12(2): 227-239.
- [11] ZHANG L X, WANG L, XIAO M S, et al. EM\_WOA: A Budget-Constrained Energy Consumption Optimization Approach for Workflow Scheduling in Clouds[J]. Peer-to-Peer Networking and Applications, 2022, 15(2): 973-987.
- [12] SHUKLA P, PANDEY S. MAA: Multi-Objective Artificial Algae Algorithm for Workflow Scheduling in Heterogeneous Fog-Cloud Environment[J]. The Journal of Supercomputing, 2023, 79(10): 11218-11260.
- [13] BRAIK M, HAMMOURI A, ATWAN J, et al. White Shark Optimizer: A Novel Bio-Inspired Meta-Heuristic Algorithm for Global Optimization Problems[J]. Knowledge-Based Systems, 2022, 243: 108457.
- [14] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey Wolf Optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
- [15] SENAPATI D, SARKAR A, KARFA C. HMDS: A Makespan Minimizing DAG Scheduler for Heterogeneous Distributed Systems[J]. ACM Transactions on Embedded Computing Systems, 20(S5): 106.
- [16] BHARATHI S, CHERVENAK A, DEELMAN E, et al. Characterization of Scientific Workflows[C]//2008 Third Workshop on Workflows in Support of Large-Scale Science. Austin: IEEE, 2008(1): 10.

(责任编辑: 申 剑)