面向边缘端工业微服务部署的多目标白鲨优化算法

李孝斌，苟堃耀，尹超
（重庆大学绿色智能制造研究所，400044，重庆）

摘要：针对在有限资源的边缘节点上部署工业微服务时难以兼顾节点的资源利用和负载均衡状况的问题，提出了一种改进型多目标白鲨优化算法。考虑工业微服务部署时边缘节点资源利用和负载状况的协同优化问题，建立了一个以边缘节点计算资源余量、服务通信能耗、负载均衡状况和存储资源余量为指标的多目标优化模型；针对该模型的求解，从初始种群质量、收敛速度、跳出局部最优能力以及优质解的保留4个方面，分别引入了混沌映射初始化、自适应权重因子、差分进化算子以及精英保留策略对多目标白鲨优化算法进行改进。模型求解实验结果表明，相较于原始多目标白鲨优化算法，改进型多目标白鲨优化算法在上述4个指标上分别优化了13.8%、37.1%、63.9%、47.4%，相较于二代遗传算法则分别优化了63.2%、53.2%、39.1%、63.6%，且所提算法收敛速度更快。

关键词：工业微服务部署；边缘节点；多目标优化；白鲨优化算法
中图分类号：TP277 **文献标志码**：A
DOI：10.7652/xjtuxb202508001 **文章编号**：0253-987X(2025)08-0001-10

Multi-Objective White Shark Optimizer for Edge-Side Industrial Microservice Deployment

LI Xiaobin, GOU Kunyao, YIN Chao
(Green Intelligent Manufacturing Research Institute, Chongqing University, Chongqing 400044, China)

Abstract: To address the challenge of balancing resource utilization and load balancing in edge nodes with limited resources during industrial microservice deployment, an improved multi-objective white shark optimizer (IMOWSO) is proposed. Considering the co-optimization of resource utilization and load conditions in edge nodes during industrial microservice deployment, a multi-objective optimization model is established with four key metrics: computational resource margin of edge nodes, service communication energy consumption, load balancing status, and storage resource margin. To solve this model, the improved algorithm enhances the original multi-objective white shark optimizer (MOWSO) in four aspects: initial population quality, convergence speed, ability to escape local optima, and preservation of high-quality solutions. Specifically, chaotic mapping initialization, adaptive weight factors, differential evolution operators, and an elite retention strategy are introduced. Experimental results demonstrate that compared to the original MOWSO, the proposed IMOWSO achieves optimizations of 13.8%, 37.1%, 63.9%, and 47.4% in the four metrics, respectively. Furthermore, when compared to the second-generation genetic

收稿日期：2025-01-06。 作者简介：李孝斌(1987—)，男，副教授，博士生导师。 基金项目：国家重点研发计划资助项目(2022YFB3305703)；国家自然科学基金资助项目(52075060)；重庆市科技创新重大研发项目(CSTB2024TIAD-STX0029)。
网络出版时间：2025-04-15 网络出版地址：https://link.cnki.net/urlid/61.1069.T.20250414.1813.007

algorithm, the improvements reach 63.2%, 53.2%, 39.1%, and 63.6%, respectively, while also exhibiting faster convergence speed.

Keywords: industrial microservices deployment; edge nodes; multi-objective optimization; white shark optimization algorithm

在工业互联网环境下,伴随着工业设备的大量接入以及高并发的用户需求,微服务架构作为一种轻量级、模块化的软件架构风格,因其灵活性、可扩展性以及高可用性的特点[1-3],越来越受到制造企业的青睐。例如:通用电气的 Predix 平台[4]、西门子的 MindSphere 平台[5]、博世的 Bosch IoT Suite[6]平台等都采用了微服务架构。跟呆板、臃肿的传统单体架构相比[7],微服务架构将应用拆分为多个小型、自治的服务单元,每个服务单元都围绕着某个特定的业务功能进行构建和部署,以实现系统的高内聚、松耦合和快速迭代[8-10]。基于上述优点,在工业互联网环境下,企业可以灵活调整微服务部署策略以构建满足实际生产需求的制造应用系统。

目前,在部分领域已经有较多关于微服务的组合部署问题的研究。Filip 等[11]总结了与在异构系统中使用微服务相关的问题,并提出了一种使用此类系统调度任务的体系结构;Ma 等[12]针对不同资源中心上微服务实例的启动问题,提出了一种基于进化多目标理论的优化问题模型,并针对模型的求解提出了一种知识驱动的进化算法。在工业领域,针对微服务的部署也有部分研究。Fan 等[13]针对微服务部署时的延迟、可靠性和负载平衡感知的问题,提出了一种基于粒子群优化的调度算法对微服务调度的多目标优化问题进行求解;Lü 等[14]针对微服务在边缘节点上部署时的通信开销和负载均衡问题,提出了一种基于学习的算法并获得最优部署策略;Chen 等[15]为减少混合环境下微服务部署时物联网设备的平均等待时间,提出了一种基于边缘云混合环境异构和动态特征的基于微服务的部署问题,并提出了一个多缓冲区深度确定性策略梯度来提供更可取的微服务部署解决方案;孙梦宇等[16]提出了一种基于日志可观测的云-边协同微服务部署优化方法,构建了基于微服务的时序过程模型,采用多目标优化算法,最小化微服务配置时延和能耗。

现有的研究充分考虑了微服务在边缘节点上部署时的负载均衡和延迟层面的问题。但是,在考虑这些问题的同时,缺乏对于资源利用的协同优化,尤其是在当前工业互联网技术的飞速发展和应用的背景下,工业微服务的规模和复杂程度迅速增加。再加上企业对于工业微服务的请求呈现高并发的特点,导致企业在边缘端部署工业微服务时难以兼顾高的资源利用(包括计算资源、存储资源和网络资源)和负载均衡。因此,本文综合考虑工业微服务在边缘节点上部署时的资源利用和负载状况,对微服务的部署策略优化问题进行了研究。首先,以计算资源剩余、微服务通信能耗、负载均衡率、存储资源剩余为优化目标,构建了一种考虑资源利用和负载状况的工业微服务边缘化部署多目标优化模型。然后,针对模型的求解,通过引入混沌映射初始化、自适应权重因子、差分进化算子和精英保留策略,提出了一种改进型多目标白鲨优化(improved multi-objective white shark optimizer, IMOWSO)算法。最后,通过与原始白鲨优化(multi-objective white shark optimizer, MOWSO)算法以及二代遗传算法(non-dominated sorting genetic algorithm, NSGA-Ⅱ)的对比实验,验证了所提算法在解决工业微服务部署问题中的有效性和可行性。

1 考虑资源利用和负载状况的工业微服务部署模型

工业微服务的部署过程如图 1 所示,在工业现场存在多个拥有有限计算资源和存储资源的边缘节点,每个边缘节点的位置信息确定。现由企业筛选出若干个满足需求的微服务,每个微服务基于容器或者虚拟机部署在边缘节点上并启动,每个微服务的启动会消耗一定的计算资源和存储资源。微服务的优化部署问题实际上就是在给定的边缘节点、微

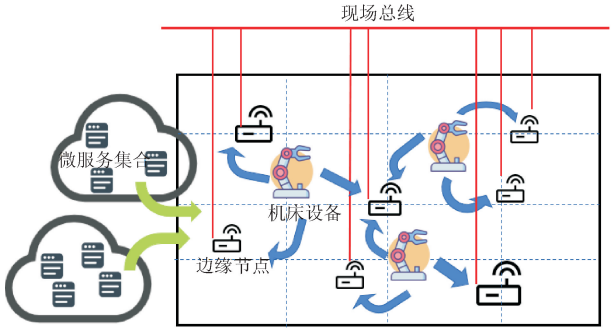


图 1 工业现场的微服务部署描述

Fig. 1 Description of microservice deployment in an industrial field

服务集合的前提下，寻求微服务实例在各边缘节点的启动和运行策略，使得在同时考虑节省资源和边缘节点负载均衡的同时，能够为制造企业提供高效快捷的服务。

1.1　微服务部署问题描述

在一个包含 m 个微服务和 n 个边缘节点的系统中，需要将微服务部署到边缘节点上，以优化 4 个目标：①最小化计算资源余量，尽可能充分利用边缘节点的计算资源；②最小化通信能耗，减少微服务之间的通信消耗的网络资源；③最小化负载均衡，实现边缘节点之间的负载均衡；④最小化存储资源余量，尽可能充分利用边缘节点的存储资源。

1.2　微服务部署相关变量定义

(1)工业微服务集。制造企业在对大量的微服务按照其需求完成筛选后，获得工业微服务集合。假设一个工业微服务集合为 $M=\{m_1,m_2,\cdots,m_m\}$，其中 m_i 表示该微服务集中索引为 i 的微服务，$m_i=(c_i,s_i,D_i)(1\leqslant i\leqslant m)$，$c_i$ 表示启动一个微服务 i 所需要的计算资源，s_i 表示启动一个微服务 i 所需要的存储资源，D_i 表示微服务 i 与该微服务集中其他微服务的通信数据量。

(2)边缘节点集群。边缘节点作为基于容器的工业微服务部署的实体，包含分布在不同位置的计算资源和存储资源。假设工业现场中一个可用于部署微服务的边缘节点集群为 $D=\{d_1,d_2,\cdots,d_n\}$，其中 d_j 表示该集群中索引为 j 的边缘节点，$d_j=(C_j,S_j,x_j^{(1)},x_j^{(2)})(1\leqslant j\leqslant n)$，$C_j$ 为边缘节点 j 的计算资源总量，S_j 为边缘节点 j 的存储资源总量，$x_j^{(1)}$、$x_j^{(2)}$ 分别表示边缘节点 j 在工业现场位置的横、纵坐标信息。

(3)工业微服务部署策略。对于工业现场存在的 n 个边缘节点 $D=\{d_1,d_2,\cdots,d_n\}$ 以及待部署的 m 个微服务 $M=\{m_1,m_2,\cdots,m_m\}$ 而言，工业微服务部署策略定义为 $\boldsymbol{x}=[x_1,x_2,\cdots,x_m]$，$x_i\in\{1,2,\cdots,n\}$，其中 x_i 表示第 i 个微服务被部署到的边缘节点编号。

1.3　考虑资源利用和负载状况的优化目标函数

考虑降低企业的运营成本、提高生产力、增强企业竞争力的需求，以计算资源余量、通信能耗、负载均衡、存储资源余量作为目标函数进行优化，下面对每个目标函数进行分析并建立数学模型。

(1)计算资源余量。在工业现场，计算资源可能会用于数据采集、状态监控、质量检测分析等场景。如果计算资源分配不当，会导致资源闲置，甚至还可能因资源的抢占导致生产任务的停滞。因此，第一个目标函数定义为计算资源余量，为各边缘节点剩余计算资源的平方和，记该目标函数为 f_1，表达式为

$$f_1=\sum_{j=1}^{n}\left(C_j-\sum_{i=1}^{m}c_i\delta_{ij}\right)^2 \tag{1}$$

式中：δ_{ij} 表示微服务 i 是否部署在边缘设备 j 上，若是则为 1，否则为 0，即

$$\delta_{ij}=\begin{cases}1, & x_i=j\\ 0, & x_i\neq j\end{cases} \tag{2}$$

(2)通信能耗。某些微服务之间往往存在通信依赖，例如状态监测微服务的状态数据来源于数据采集微服务，由于用户对工业微服务的请求呈现高并发的特性，微服务间的频繁通信可能导致网络拥塞和能耗增加，影响系统实时性。因此，第二个目标函数定义为微服务间的通信总能耗。对于每一对微服务 (i_1,i_2)，如果它们部署在不同的边缘节点上，则根据它们之间的通信数据量和设备间距离计算能耗，记该目标函数为 f_2，表达式为

$$f_2=\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}Q_{i_1,i_2}E_{\text{comm}}L(x_{i_1},x_{i_2}) \tag{3}$$

式中：Q_{i_1,i_2} 表示微服务 i_1 和 i_2 之间的关联系数；E_{comm} 表示每单位距离的数据传输能耗；$L(x_{i_1},x_{i_2})$ 表示微服务 i_1 和 i_2 所部署的边缘节点之间的物理距离，表达式为

$$L(x_{i_1},x_{i_2})=\sqrt{(x_{i_1}^{(1)}-x_{i_2}^{(1)})^2+(x_{i_1}^{(2)}-x_{i_2}^{(2)})^2} \tag{4}$$

(3)负载均衡。由于工业现场制造资源有限，用户往往希望能有较高的资源利用率而尽可能把较多的微服务部署到同一个边缘节点，这就导致如果有某一时刻某一类微服务的请求突增时，微服务由于资源抢占而导致系统崩溃。因此，第 3 个目标函数定义为负载均衡，计算各边缘节点的资源利用率，与平均利用率的偏差平方和，衡量负载均衡程度，记该目标函数为 f_3，表达式为

$$f_3=\frac{1}{n}\sum_{j=1}^{n}\left(\frac{\sum_{i=1}^{m}c_i\delta_{ij}}{C_j}-\overline{U}\right)^2 \tag{5}$$

式中：$\overline{U}$ 为平均资源利用率，定义式为

$$\overline{U}=\frac{1}{n}\sum_{j=1}^{n}\frac{\sum_{i=1}^{m}c_i\delta_{ij}}{C_j} \tag{6}$$

(4)存储资源余量。由于大规模机床设备产生大量数据、用户对微服务的高并发请求等原因，不当

的微服务部署策略必然导致存储资源的浪费，进而影响企业的运营成本。因此，第 4 个目标函数定义为存储资源余量，为各边缘节点剩余计算资源的平方和，记该目标函数为 f_4，表达式为

$$f_4 = \sum_{j=1}^{n} \left(S_j - \sum_{i=1}^{m} s_i \delta_{ij}\right)^2 \tag{7}$$

1.4 约束条件

基于上述待优化的目标函数，结合工业微服务的部署特点，提出微服务完备性约束、计算资源总量约束和存储资源总量约束共 3 个约束条件，具体描述如下。

(1)微服务完备性约束。微服务必须部署在某个边缘节点上

$$x_i \in \{1,2,\cdots,n\}, \forall i \in \{1,2,\cdots,m\} \tag{8}$$

(2)资源总量约束。边缘节点的资源不能超载，即

$$\sum_{i=1}^{m} c_i \delta_{ij} \leqslant C_j, \forall j \in \{1,2,\cdots,n\} \tag{9}$$

$$\sum_{i=1}^{m} s_i \delta_{ij} \leqslant S_j, \forall j \in \{1,2,\cdots,n\} \tag{10}$$

1.5 工业微服务部署优化模型

综合上述目标函数和约束条件，基于多目标优化的工业微服务部署模型描述为

$$\begin{cases} \min\limits_{x \in \boldsymbol{\Omega}} F(\boldsymbol{x}) = \{f_1, f_2, f_3, f_4\} \\ \text{s. t.} \\ g_1(\boldsymbol{x}): x_i \in \{1,2,\cdots,n\}, \forall i \in \{1,2,\cdots,m\} \\ g_2(\boldsymbol{x}): \sum\limits_{i=1}^{m} c_i \delta_{ij} \leqslant C_j, \forall j \in \{1,2,\cdots,n\} \\ g_3(\boldsymbol{x}): \sum\limits_{i=1}^{m} s_i \delta_{ij} \leqslant S_j, \forall j \in \{1,2,\cdots,n\} \end{cases} \tag{11}$$

式中：$\boldsymbol{\Omega}$ 为搜索空间。

从建立的优化模型可知，该问题属于典型的 NP-hard 问题，采用传统的多项式解决方法难以对问题进行求解。因此，本文寻求元启发式算法对模型进行求解。同时，在边缘节点上存在的计算资源、存储资源总和分别大于微服务启动所需求资源的情况下，该模型一定存在解。

2 改进型多目标白鲨优化算法

MOWSO 算法由 Braik 等[17]于 2022 年提出，该算法受白鲨导航和觅食时具有的非凡听觉和嗅觉启发。针对上述模型的求解，本文在 MOWSO 算法基础上通过引入混沌映射初始化、自适应权重因子、差分进化算子和精英保留策略提出了 IMOWSO 算法。

2.1 MOWSO 算法原理

白鲨的听觉和嗅觉器官十分发达，这使得它的觅食和捕食能力大大提高，白鲨优化算法主要包含以下阶段。

(1)快速向猎物移动。白鲨向猎物移动可以定义为

$$v_{k+1}^q = \mu\left[v_k^q + p_1\left(w_{\mathrm{gbest}_k} - w_k^q\right) r_1 + p_2\left(w_{\mathrm{best}}^{v_k^q} - w_k^q\right) r_2\right] \tag{12}$$

式中：$q = 1,2,\cdots,n$ 为白鲨的种群数量；v_{k+1}^q 为在第 $k+1$ 次迭代中第 q 个白鲨种群的新速度；v_k^q 为第 k 次迭代中第 q 个白鲨种群的当前速度；w_{gbest_k} 为第 k 次迭代时，白鲨种群到过的全局最优位置；w_k^q 表示在第 k 次迭代中第 q 条白鲨的当前位置；$w_{\mathrm{best}}^{v_k^q}$ 为种群中已知的第 q 个最佳位置；r_1 和 r_2 是在[0,1]范围内均匀生成的两个随机数；μ 为白鲨优化算法中用于分析白鲨收敛行为的收缩因子，定义为 $\mu = 2/\left|2 - \tau - \sqrt{\tau^2 - 4\tau}\right|$，其中 $\tau = 4.125$ 表示加速度系数；p_1 和 p_2 为固定权重因子，分别控制 w_{gbest_k} 和 $w_{\mathrm{best}}^{v_k^q}$ 对于 w_k^q 的影响，计算式为

$$p_1 = p_{\max} + (p_{\max} - p_{\min}) \mathrm{e}^{-\left(\frac{4k}{K}\right)^2} \tag{13}$$

$$p_2 = p_{\min} + (p_{\max} - p_{\min}) \mathrm{e}^{-\left(\frac{4k}{K}\right)^2} \tag{14}$$

其中，k 和 K 分别代表当前迭代次数和最大迭代次数，$p_{\min}$ 和 $p_{\max}$ 分别代表白鲨实现良好运动的初始速度和次速度。

(2)包围最佳猎物。白鲨包围最佳猎物的行为可表示为

$$w_{k+1}^q = \begin{cases} w_k^q(\neg \oplus w_0) + ua + lb, & c_{\mathrm{rand}} < m_v \\ w_k^q + \dfrac{v_k^q}{f}, & c_{\mathrm{rand}} \geqslant m_v \end{cases} \tag{15}$$

式中：w_{k+1}^q 表示在第 $k+1$ 次迭代时，第 q 条白鲨的新位置；$\neg$ 是负算子；$a = [\mathrm{sgn}(w_k^q - u) > 0]$ 和 $b = [\mathrm{sgn}(w_k^q - l) < 0]$ 表示两个布尔值二进制数；u 和 l 分别表示搜索空间的上下限；$w_0 = \oplus(a,b)$ 表示一个逻辑数，其中 $\oplus$ 为位异或运算；$f = f_{\min} + (f_{\max} - f_{\min})/(f_{\max} + f_{\min})$ 代表白鲨的波浪上起伏频率，其中 $f_{\min}$ 和 $f_{\max}$ 分别表示波动运动的最小频率和最大频率；c_{rand} 表示一个范围在[0,1]的随机数；$m_v = (a_0 + \mathrm{e}^{(K/2-k)/a_1})^{-1}$ 定义了白鲨的运动能力，迭代数越高，运动能力越强，其中 a_0 和 a_1 是两个正常数，用于管理探测和开发行为。

(3)向最佳位置靠近。当白鲨锁定猎物的位置后开始进行围攻,快速移动至最优进攻位置开始捕杀猎物,这个过程可表示为

$$\acute{w}_{k+1}^{q}=w_{\mathrm{gbest}_k}+r_1 D_{\mathrm{w}}\mathrm{sgn}(r_2-0.5),\ r_3<s_{\mathrm{s}} \tag{16}$$

式中:$\acute{w}_{k+1}^{q}$ 表示第 q 条白鲨相对于猎物位置的更新位置;$\mathrm{sgn}(r_2-0.5)$ 表示用 1 或 −1 来调整白鲨进行正向或逆向搜索;变量 $r_1\sim r_3$ 均为[0,1]内的随机数;$D_{\mathrm{w}}=\left|c_{\mathrm{rand}}(w_{\mathrm{gbest}_k}-w_k^q)\right|$ 表示猎物与白鲨之间当前的距离;$s_{\mathrm{s}}=\left|1-\mathrm{e}^{-a_2(k/K)}\right|$ 表示白鲨跟随其他接近最佳位置的白鲨时的嗅觉强度参数,其中 a_2 是一个正常数,用于管理探测和开发行为。

(4)鱼群行为。为了在数学上模拟白鲨群的行为,保留前两个最佳解,并根据两个最佳位置更新其他白鲨的位置,可以表示为

$$w_{k+1}^{q}=\frac{w_k^q+\acute{w}_{k+1}^{q}}{2c_{\mathrm{rand}}} \tag{17}$$

2.2　IMOWSO 算法改进过程

白鲨优化算法通过模拟白鲨的捕食行为,能有效探索解空间,寻找全局最优解,具有强大的全局搜索能力,能被应用于许多领域的优化问题。但是,工业场景下的用户高并发请求使得大规模的微服务需要同时部署,此外边缘节点也随着机床设备的大规模接入而急剧增加,这导致部署策略更加难以确定,原始白鲨优化算法在解决这类复杂问题时,仍然存在种群多样不足、收敛速度较慢、容易陷入局部最优、最优解易丢失的问题。

针对多目标白鲨优化算法的上述局限性进行改进,通过引入混沌映射、自适应权重因子、差分变异算子以及精英保留策略对算法性能进行调优,提出一种改进型白鲨优化算法。

(1)引入混沌映射策略,对种群初始化进行改进。文献[18]已经证实,混沌系统具备隐藏周期性以及非重复行为的特点。通过基于混沌映射的初始化策略,种群可以在一定范围内跨越几乎所有状态且不重复。混沌序列由 Logistic 映射产生

$$z_{n+1}=rz_n(1-z_n) \tag{18}$$

式中:z_n 是第 n 次迭代混沌变量的值;r 是 Logistic 映射的控制参数,用于控制混沌搜索速度,当 $r\in[3.57,4]$ 时,系统处于混沌状态[19],为确保生成的序列具有遍历性和伪随机性,设置 $r=4$。

将混沌映射应用于算法种群初始化的过程,则基于混沌映射的种群初始化可以表示为

$$w=l+z(u-l) \tag{19}$$

式中:w 表示初始解位置。使用混沌映射生成的序列 z 具有遍历性和伪随机性,能够更均匀地覆盖整个搜索空间。

(2)引入自适应权重因子。根据迭代次数或算法状态,自适应调整随机权重和收缩因子。文献[20]已经证明,自适应权重因子的大小会对算法的搜索能力产生不同影响:在算法运行初期,较大的自适应权重因子有助于提升算法的全局搜索能力;当算法进入后期阶段时,较小的自适应权重因子可使算法获得较好的局部搜索能力。因此,将式(12)中的固定权重和收缩因子改进为自适应权重 p_1'、p_2' 和收缩因子 μ',按照迭代次数动态调整参数,具体公式如下

$$p_1'=p_{1,\max}-\left((p_{1,\max}-p_{1,\min})\frac{g}{g_{\mathrm{m}}}\right) \tag{20}$$

$$p_2'=p_{2,\max}-\left((p_{2,\max}-p_{2,\min})\frac{g}{g_{\mathrm{m}}}\right) \tag{21}$$

$$\mu'=w_{\max}-\left((w_{\max}-w_{\min})\frac{g}{g_{\mathrm{m}}}\right) \tag{22}$$

式中:根据文献[21]选取 $p_{1,\max}=2.5$、$p_{1,\min}=0.5$、$p_{2,\max}=2.5$、$p_{2,\min}=0.5$、$w_{\max}=0.9$、$w_{\min}=0.4$;g 表示当前迭代次数;g_{m} 表示最大迭代数。

改进后的随机权重和收缩因子随着迭代数的增大从较大的值逐渐减小,初始阶段有助于全局搜索,后期促进局部收敛。

(3)引入差分进化算子。差分进化是一种有效的全局优化算法。将差分进化算子引入多目标白鲨优化算法中,可以利用种群个体的变异、交叉和选择机制,生成新的候选解,增加种群的多样性[22]。在每一代中,对部分个体应用差分进化操作。变异操作为

$$V_q=x_{r_1}+F(x_{r_2}-x_{r_3}) \tag{23}$$

式中:V_q 表示第 q 条白鲨变异后的个体;$x_{r_1}\sim x_{r_3}$ 是随机选择的 3 个个体,且互不相同;F 是缩放因子,一般取值为[0.5,1][23],为增加种群多样性,本文选择较高的 $F=1$。

交叉操作如下

$$U_{q_1,q_2}=\begin{cases}V_{q_1,q_2}, & c_{\mathrm{rand}}<C_{\mathrm{R}}\\ w^q, & c_{\mathrm{rand}}\geqslant C_{\mathrm{R}}\end{cases} \tag{24}$$

式中:U_{q_1,q_2} 表示个体 q_1 和 q_2 交叉操作后的个体;C_{R} 是交叉概率,一般取值为[0.8,0.95][23],本文选取 $C_{\mathrm{R}}=0.9$。

选择操作如下：如果 U_{q_1,q_2} 优于 w^q，则在下一代种群中接受 U_{q_1,q_2}，否则接受 w^q。

这种机制能够有效避免算法陷入局部最优，提高全局搜索能力。在工业微服务部署问题中，复杂的解空间和多个局部最优点使得算法容易陷入局部最优。差分进化算子能够帮助算法跳出局部最优，找到更优的解。

(4)引入精英保留策略。在 MOWSO 算法中，最优解可能在迭代过程中被新的解替代，存在丢失的风险。精英保留策略通过在每一代中保留当前找到的最优解，确保这些优质解不会被遗忘[24]。这有助于保持解的质量，防止算法退化，确保解的质量逐步提升或至少不下降以及引导种群演化。

3 采用 IMOWSO 算法的模型求解步骤

步骤 1 初始化，具体步骤如下。

(1)设置模型参数。初始化模型相关参数，包括 m、k、c_i、s_i、C_j、S_j、Q_{i_1,i_2}、$x^{(1)}$、$x^{(2)}$、E_{comm}。

(2)设置算法参数。初始化算法相关参数，包括 N_p、g_m、r、$p_{1,\max}$、$p_{1,\min}$、$p_{2,\max}$、$p_{2,\min}$、$w_{\max}$、$w_{\min}$、F、C_R、算法外部存档大小 N_r、网格划分数量 n_{grid}。

步骤 2 初始种群生成。使用混沌映射生成初始种群，其中初始化混沌序列 z_1 是随机生成的，采用式(18)生成混沌序列，将混沌序列映射到变量空间，生成初始解，确保变量在给定范围内，并满足整数约束。

步骤 3 初始评估。计算初始种群中每个个体的目标函数值，初始化每个个体的历史最优位置及其适应度，初始化全局最优存档，存储非支配解。

步骤 4 主循环。在迭代数从 1 到 g_m 的过程中，执行以下步骤。

(1)自适应参数更新。分别根据式(20)～(22)更新权重因子 p'_1、p'_2和收缩因子 μ'。

(2)白鲨速度和位置更新。对于种群中随机个体 i，从存档中随机选择全局最优解 w_{gbestk}，分别根据式(12)、(15)更新白鲨的速度和位置。

(3)考虑最佳鲨鱼个体的影响。对于种群中的每个个体，以一定概率 s_s 进行位置更新，从存档中随机选择全局最优解 w_{gbestk}，根据式(16)继续更新白鲨的位置，然后根据式(17)进行鱼群行为。

(4)差分进化算子。对于种群中的每个个体，随机选择 3 个不同的个体 x_{r_1} ～ x_{r_3}，按照式(23)生成变异个体，按照式(24)执行交叉操作，执行选择操作，修正种群个体以满足变量范围和整数约束，计算试验个体的适应度。

(5)个体最优和全局最优更新。如果存档大小不超过 N_r，更新每个个体的历史最优位置及其适应度，更新存档，将新的非支配解加入存档；如果存档大小超过 N_r，则根据拥挤度删除多余的解。

(6)精英保留策略。从存档中选取当前最优的个体，将该精英个体直接复制到下一代种群的第一个位置。

步骤 5 终止。当迭代数达到 g_m 时，终止算法，输出存档中的非支配解，作为最终的 Pareto 最优解集。

图 2 为采用 IMOWSO 算法求解工业微服务部署模型的流程图。

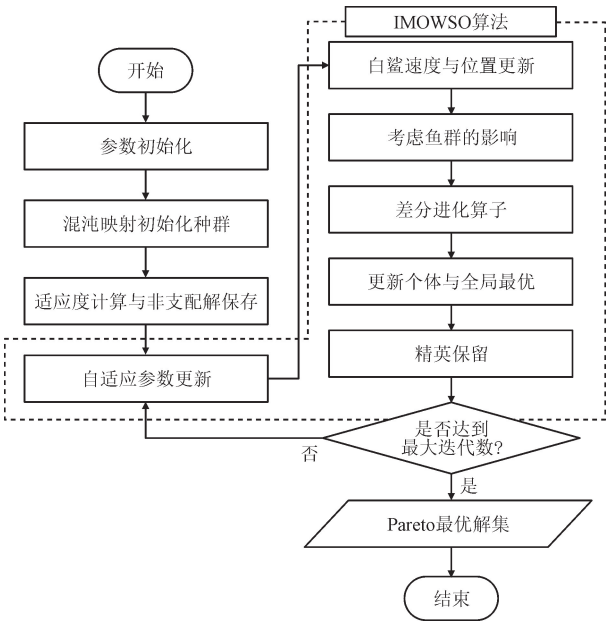


图 2 采用 IMOWSO 算法的模型求解流程

Fig. 2 Model solving process based on IMOWSO algorithm

4 实验验证

4.1 模型与算法参数配置

为了评估 IMOWSO 算法的性能及有效性，通过实验求解第 2 节中式(8)～(10)约束下的工业微服务部署多目标优化模型。本文实验采用阿里云提供的跟踪 V2018 数据集[25]，包括 18 个微服务和 10 个资源中心(替代边缘节点)。

表 1 展示了实验中的各算法参数与模型参数，具体包括 NSGA-Ⅱ、MOWSO 和 IMOWSO 这 3 种算法的共用参数及各自参数、微服务部署模型参数。其中：MOWSO 算法、NSGA-Ⅱ的参数来源于文献[17,24]，IMOWSO 算法的参数已在 2.2 小节中介绍。微服务和边缘节点的参数如表 2～3 所示。

表 1　各算法参数与模型参数

Table 1　Algorithm parameters and model parameter

设置对象	参数设置
NSGA-Ⅱ	交叉概率 0.9，变异概率 1/4，交叉分布指数 20，变异分布指数 20
MOWSO 算法	$n_{\text{grid}}=30$，$p_{\max}=1.5$，$p_{\min}=0.5$，$\mu=0.5$，$\tau=4.125$，$a_0=6.25$，$a_1=100$，$a_2=0.0005$
IMOWSO 算法	$n_{\text{grid}}=30$，$r=4$，$q_{1,\max}=q_{2,\max}=2.5$，$q_{1,\min}=q_{2,\min}=0.5$，$w_{\max}=0.9$，$w_{\min}=0.4$，$F=1$，$C_R=0.9$
算法共同部分	$N_p=100$，$g_m=1\,000$，$N_r=200$
本文模型	$E_{\text{comm}}=50$

表 2　微服务资源需求量

Table 2　Microservice resource requirements

i	c_i	s_i
1	1.6	1.3
2	4.1	13.8
3	3.0	5.2
4	5.1	6.6
5	3.1	6.9
6	5.7	4.2
7	7.4	2.5
8	4.6	4.3
9	3.1	3.2
10	2.9	2.4
11	6.2	5.2
12	6.9	5.3
13	6.6	7.1
14	12.9	6.8
15	9.4	8.1
16	6.4	6.3
17	6.3	3.8
18	1.6	3.2

表 3　边缘节点资源总量及位置信息

Table 3　Microservice resource requirements

j	C_j	S_j	$(x^{(1)}, x^{(2)})$
1	50	20	(28.603, 85.738)
2	50	35	(39.513, 93.088)
3	40	20	(55.246, 65.816)
4	50	30	(45.762, 78.908)
5	60	35	(15.671, 34.139)
6	40	20	(65.723, 38.077)
7	30	15	(71.404, 43.728)
8	45	25	(73.957, 40.677)
9	30	25	(65.723, 38.077)
10	20	40	(89.881, 34.247)

图 3 展示了微服务间的关联系数 Q_{i_1,i_2}。图中：0 表示微服务 i_1 和 i_2 之间无直接通信；1 表示微服务 i_1 和 i_2 之间有低频、小规模的数据交互；2 表示微服务 i_1 和 i_2 之间有中等强度的数据交互；3 表示微服务 i_1 和 i_2 之间有大量的数据交互。

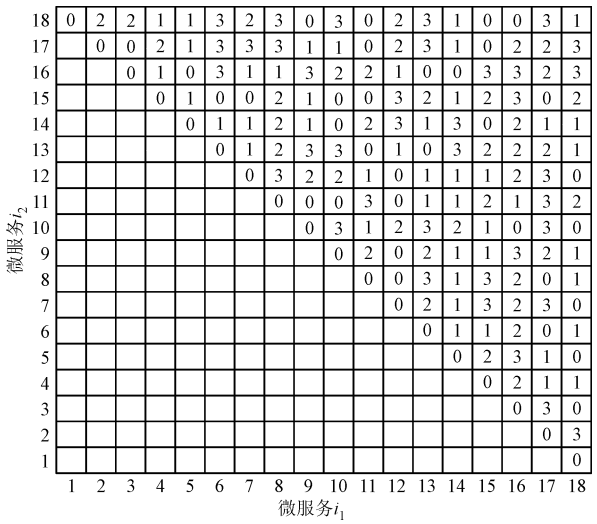


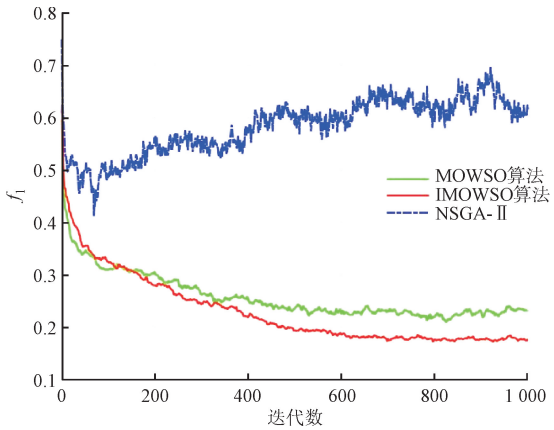
图 3　微服务间关联系数 Q_{i_1,i_2}

Fig. 3　The correlation coefficient between microservices Q_{i_1,i_2}

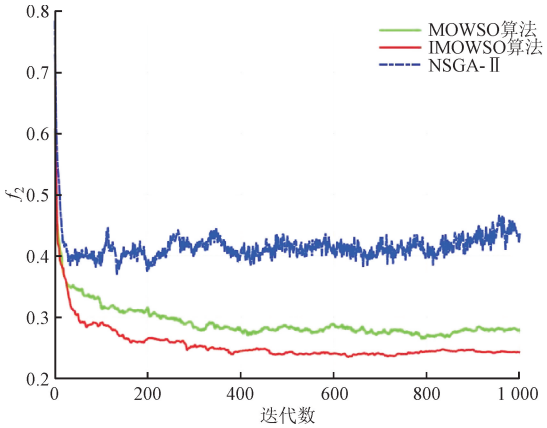
4.2　模型求解结果对比与分析

为验证 IMOWSO 算法相较于 MOWSO 算法以及 NSGA-Ⅱ在工业微服务边缘化优化部署问题中的性能表现，通过模型求解实验对比了 3 种算法在计算资源余量、通信能耗、负载均衡状况和存储资源余量 4 个优化目标函数上的表现。

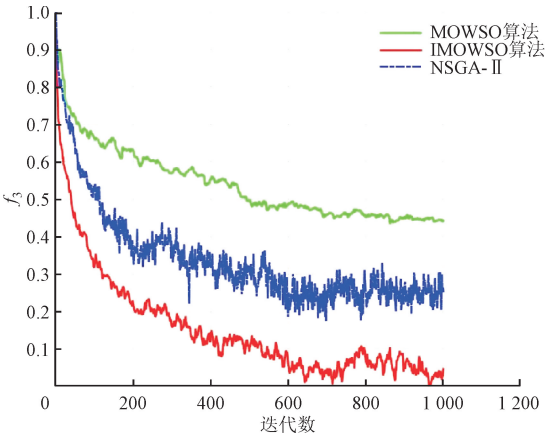
图 4 展示了 IMOWSO 算法、MOWSO 算法和 NSGA-Ⅱ在 4 个目标函数上的收敛曲线。可以看出，在求解本文提出的边缘端工业微服务部署模型时，IMOWSO 算法在收敛速度上快于 MOWSO 算法和 NSGA-Ⅱ，且在所有目标函数上都表现出了良好的稳定性和收敛性。



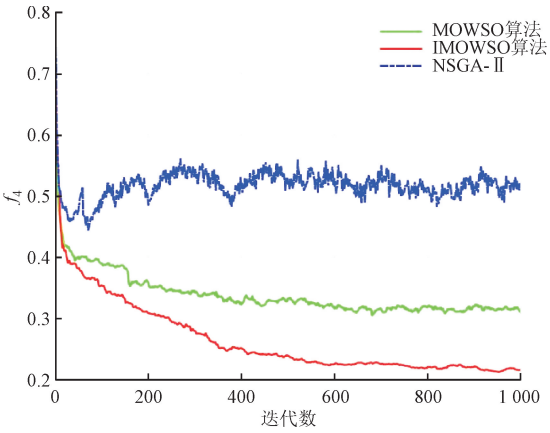
(a)计算资源余量



(b)服务通信能耗



(c)负载均衡状况

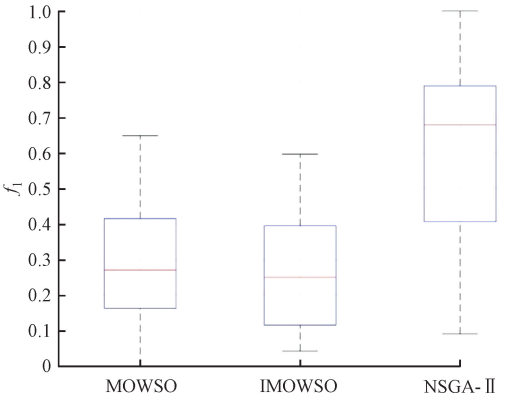


(d)存储资源余量

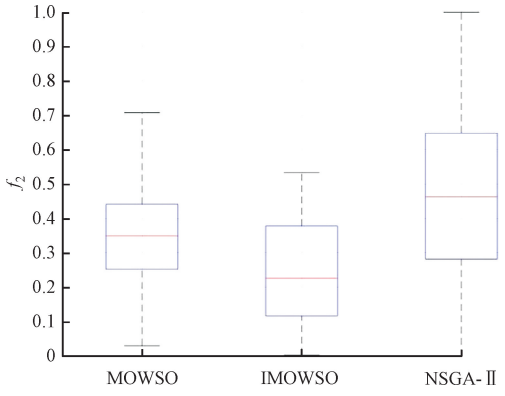
图 4 3 种算法在 4 个目标函数上的迭代曲线

Fig. 4 Iterative curves of the 3 algorithms on 4 objective functions

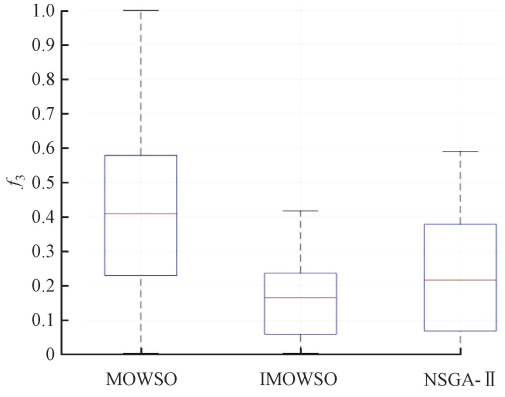
图 5 展示了 3 个算法在 4 个目标函数上的归一化目标函数值箱线图。可以看出，与 MOWSO、NSGA-Ⅱ状况相比，IMOWSO 在计算资源余量、服务通信能耗、负载均衡状况以及存储资源余量指标上均有明显的优化。具体而言，相较于 MOWSO，本文 IMOWSO 在 4 个目标函数上分别优化了 13.8%、37.1%、63.9%、47.4%，相较于 NSGA-Ⅱ分别优化了 63.2%、53.2%、39.1%、63.6%。



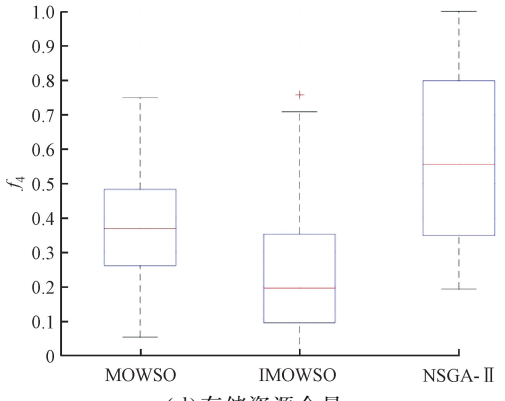
(a)计算资源余量



(b)服务通信能耗



(c)负载均衡状况



(d)存储资源余量

图 5 3 种算法在 4 个目标函数上的归一化目标函数值箱线图

Fig. 5 Box plots of normalized target values of 3 algorithms on 4 objective functions

综上所述可知，本文提出的改进多目标白鲨优化算法在工业微服务的优化部署问题中具有一定的优势，尤其在资源余量和负载均衡方面具有良好的优化效果。

5　结　论

(1)针对工业微服务在有限资源边缘节点上的优化部署问题，建立了一个以计算资源余量、服务通信能耗、负载均衡状况和存储资源余量为目标的优化模型。

(2)针对模型的求解，通过引入混沌映射初始化、自适应权重因子、差分进化算子和精英保留策略，提出了一种改进多目标白鲨优化算法。

(3)实验结果表明，本文提出的 IMOWSO 在模型求解收敛速度、稳定性方面优于 MOWSO 和 NSGA-Ⅱ。相较于 MOWSO，IMOWSO 在计算资源余量、服务通信能耗、负载均衡状况和存储资源余量 4 个优化目标上分别优化了 13.8%、37.1%、63.9%、47.4%，而相较于 NSGA-Ⅱ则分别优化了 63.2%、53.2%、39.1%、63.6%。

参考文献：

[1] 杨鹏. 工业互联网平台微服务参考框架标准解读 [J]. 信息技术与标准化，2023(10)：14-18.
YANG Peng. Interpretation of industrial internet platforms mircoservices reference framework [J]. Information Technology & Standardization，2023(10)：14-18.

[2] 国家市场监督管理总局，国家标准化管理委员会. 工业互联网平台 微服务参考框架：GB/T 42568—2023 [S]. 北京：中国标准出版社，2023.

[3] BIGHETI J A，FERNANDES M M，GODOY E P. Control as a service：a microservice approach to industry 4.0 [C]//2019 Ⅱ Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 & IoT). Piscataway，NJ，USA：IEEE，2019：438-443.

[4] STEIBER A，ALÄNGE S，GHOSH S，et al. Digital transformation of industrial firms：an innovation diffusion perspective [J]. European Journal of Innovation Management，2021，24(3)：799-819.

[5] KULAWIAK K E. Manufacturing the platform economy. An exploratory case study of MindSphere，the industrial digital platform from Siemens [D]. Oslo，Norway：University of Oslo，2021.

[6] JUNG S，FERBER S，CRAMER I，et al. Bosch IoT suite：exploiting the potential of smart connected products [M]//GASSMANN O，FERRANDINA F. Connected Business：Create Value in a Networked Economy. Cham：Springer International Publishing，2021：267-282.

[7] KOREN Y，HU S J，WEBER T W. Impact of manufacturing system configuration on performance [J]. CIRP Annals，1998，47(1)：369-372.

[8] BLINOWSKI G，OJDOWSKA A，PRZYBYŁEK A. Monolithic vs. Microservice architecture：a performance and scalability evaluation [J]. IEEE Access，2022，10：20357-20374.

[9] ZHANG He，LI Shanshan，JIA Zijia，et al. Microservice architecture in reality：an industrial inquiry [C]//2019 IEEE International Conference on Software Architecture (ICSA). Piscataway，NJ，USA：IEEE，2019：51-60.

[10] 李亚杰，李昭楠. 面向微服务的制造执行系统关键技术研究 [J]. 制造技术与机床，2023(9)：95-101.
LI Yajie，LI Zhaonan. Research on key technologies of microservice based manufacturing execution system [J]. Manufacturing Technology & Machine Tool，2023(9)：95-101.

[11] FILIP I D，POP F，SERBANESCU C，et al. Microservices scheduling model over heterogeneous cloud-edge environments as support for IoT applications [J]. IEEE Internet of Things Journal，2018，5(4)：2672-2681.

[12] MA Wubin，WANG Rui，GU Yuanlin，et al. Multi-objective microservice deployment optimization via a knowledge-driven evolutionary algorithm [J]. Complex & Intelligent Systems，2021，7(3)：1153-1171.

[13] FAN Guisheng，CHEN Liang，YU Huiqun，et al. Multi-objective optimization of container-based microservice scheduling in edge computing [J]. Computer Science and Information Systems，2021，18(1)：23-42.

[14] LÜ Wenkai，WANG Quan，YANG Pengfei，et al. Microservice deployment in edge computing based on deep Q learning [J]. IEEE Transactions on Parallel and Distributed Systems，2022，33(11)：2968-2978.

[15] CHEN Lulu，XU Yangchuan，LU Zhihui，et al. IoT microservice deployment in edge-cloud hybrid environment using reinforcement learning [J]. IEEE Internet of Things Journal，2021，8(16)：12610-12622.

[16] 孙梦宇，林睿. 基于日志可观测的云：边协同微服务部署优化 [J]. 电信科学，2023，39(12)：122-132.
SUN Mengyu，LIN Rui. Cloud-edge collaboration service deployment optimization based on log observability [J]. Telecommunications Science，2023，39

(12): 122-132.

[17] 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.

[18] CAPONETTO R, FORTUNA L, FAZZINO S, et al. Chaotic sequences to improve the performance of evolutionary algorithms [J]. IEEE Transactions on Evolutionary Computation, 2003, 7(3): 289-304.

[19] WU Guocheng, BALEANU D. Discrete fractional logistic map and its chaos [J]. Nonlinear Dynamics, 2014, 75(1): 283-287.

[20] YOUSRI D, ABDELATY A M, SAID L A, et al. Chaotic flower pollination and grey wolf algorithms for parameter extraction of bio-impedance models [J]. Applied Soft Computing, 2019, 75: 750-774.

[21] YOU Zhiyu, CHEN Weirong, HE Guojun, et al. Adaptive weight particle swarm optimization algorithm with constriction factor [C]//2010 International Conference of Information Science and Management Engineering. Piscataway, NJ, USA: IEEE, 2010: 245-248.

[22] MA Haiping, ZHANG Yajing, SUN Shengyi, et al. A comprehensive survey on NSGA-Ⅱ for multi-objective optimization and applications [J]. Artificial Intelligence Review, 2023, 56(12): 15217-15270.

[23] CHIN V J, SALAM Z, ISHAQUE K. An accurate and fast computational algorithm for the two-diode model of PV module based on a hybrid method [J]. IEEE Transactions on Industrial Electronics, 2017, 64(8): 6212-6222.

[24] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-Ⅱ [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[25] ZHU Jianyong, LU Bin, YU Xiaoqiang, et al. An approach to workload generation for cloud benchmarking: a view from Alibaba trace [C]//2023 IEEE 15th International Symposium on Autonomous Decentralized System (ISADS). Piscataway, NJ, USA: IEEE, 2023: 1-8.

(编辑 陶晴)