

基于强化学习与鲸鱼优化算法的低延迟网络功能调度方法

刘智伟¹ 舒兆港¹

LIU Zhiwei SHU Zhaogang

摘要

针对在虚拟网络中如何对延迟敏感的服务功能链（SFC）上的虚拟网络功能（VNF）在调度过程时将 VNF 映射、调度联合解决并最小化 VNF 处理延迟的问题，提出了一种基于强化学习自调节鲸鱼优化算法，通过不同环境自动调节鲸鱼优化算法中的参数，同时使用交叉、变异以及领域算子整体来提高鲸鱼优化算法性能，在保证满足 SFC 的约束需求的情况下，以最小化完成时间为目标，建立 VNF 调度模型。模型通过改进鲸鱼优化算法和强化学习自动调节参数来加快收敛速度，找寻最优解。仿真结果表明，所提出的算法与 HGWOA 算法和 GABL 算法相比在性能上分别提升 10% 与 16%。

关键词

服务功能链；网络功能虚拟化；VNF 调度；鲸鱼优化算法

doi: 10.3969/j.issn.1672-9528.2024.05.005

0 引言

随着未来 5G 时代的发展，对网络的性能和灵活性的要求更为迫切。从扩展现实到自动驾驶，再到远程机器人手术，这些低延迟服务和应用对网络的要求变得更加严格。而网络功能虚拟化（network function virtualization, NFV）在推动移动网络发展方面将扮演着至关重要的角色。与传统网络相比，它引入了一种更为灵活和可编程的范式。在传统网络中，通常依赖专有硬件，每次需要引入新的网络功能时，都需要大规模部署昂贵且专门定制的硬件设备。这不仅增加了资本和运营成本^[1-2]，而且限制了网络的灵活性和适应性。而网络功能虚拟化的核心思想是将网络功能抽象为软件实例，即虚拟网络功能（virtualization network function, VNF）^[3]。这一转变使得网络服务的部署变得更为灵活，不再受制于特定硬件的限制。通过将网络功能虚拟化，人们不再需要大规模部署专用硬件，而是可以利用通用硬件上的虚拟化实例来提供各种网络服务。这不仅减少了硬件成本，还提高了网络的扩展性和适应性，使得网络能够更加迅速地适应不断变化的服务需求。

当然，在构建支持网络功能虚拟化的基础设施上配置网络服务（network service, NS）并非一项简单的任务，这类服务要求其流量必须按照特定的顺序经过一系列虚拟网络功能的引导，同时必须在极为严格的服务期限内完成^[4-6]。这一挑战被称为 NFV 资源分配（NFV-RA）^[7-8]。VNF-RA 主要由

三部分组成，分别是 VNF 组成、VNF 映射和 VNF 调度。为了满足具有严格服务期限的要求并提高网络整体服务质量（quality of service, QoS），就必须优化每个服务中 VNF 的节点选择和任务顺序，而这也是虚拟网络功能调度的主要作用。

在文献 [9] 中首次将 VNF 调度定义为车间调度模型，但未提出相应的多项式时间方案解决。文献 [10] 着眼于云计算中的 VNF 调度问题，考虑了多种用户需求，例如延迟和成本，提出了一种基于公平加权亲和的启发式算法 FWS，设计了最大完成时间最小化和传输总流量最小化的双目标。文献 [11] 中提出了一种基于 VNF 实例共享的服务功能链部署算法，通过建立 VNF 和虚拟链路映射模型，在保证 SFC 时延需求的前提下，最大化 VNF 实例共享，以平衡链路转发成本和 VNF 部署成本。相比于已有算法，该方法降低了总运营成本，并在增加 SFC 数量时保持较高的服务接收率。文献 [12] 提出的先进启发式算法，用于将虚拟网络功能迁移到另一个可用节点，从而有效提高服务功能链的利用率和接收率，该算法的目标是更有效地管理和优化网络资源，以提升整体性能。文献 [13] 研究了 MEC 环境下的 VNF 调度问题，提出了一个在线的 VNF 调度算法，可将网络中的 VNF 根据实际流量进行实例化、缩放、迁移和销毁。文献 [14] 提出将该问题抽象为作业车间调度问题，优化的目标为最小化调度完成时间，并提出使用禁忌搜索算法解决 VNF 中的资源分配与映射的问题。文献 [15] 在功能链路调度研究中，将虚拟网络功能动态映射至虚拟机并进行调度，并设计了一种混合蜂群算法。该算法先使用了随机方法进行初始化种群，然后使用蜂群算法进行求解，使用随机方法初始化种群，无法保证初始解的全局性，影响解的效果。文献 [16-17] 研究了 VNF 放置和调度

1. 福建农林大学计算机与信息学院 福建福州 350002

[基金项目] 福建省自然科学基金资助项目（2020J01574）；中国高校产学研创新基金资助项目（2021FNA05003）

的联合问题,然而文献[16]主要关注放置新的 VNF,而不考虑网络中已部署的 VNF 用于调度 NS 的可重用性。这可能不是一个经济高效的解决方案,因为放置新实例通常会产生额外成本。此外,文献[17]在调度网络服务时没有考虑通信延迟,这可能导致违反给定的服务期限。

可以看出,现有的一些研究中仅单独考虑 VNF 的放置或者 VNF 的调度并没有联合在一起^[18-20],同时有些研究也没考虑到流量的传输问题,这些可能并不能很好地满足服务要求,其次用传统的进化算法来解决问题又会陷入局部最优解。针对这些问题,本文提出了一个基于强化学习的改进鲸鱼算法,本文主要贡献是:(1)同时采用了 POX 交叉和负载均衡变异的方式提高可行解的多样性,避免陷入局部最优;(2)通过强化学习,根据环境实时调节鲸鱼参数,加快求解速度;(3)实验证明该算法与其他启发式算法相比可以获得更有效的 VNF 调度方案。

1 问题描述与模型

1.1 问题描述

虽然在现有的 VNF 资源分配问题中,主要是 VNF 组成、VNF 映射和 VNF 调度这三个问题,但对服务质量 QOS 影响最大的还是 VNF 映射和 VNF 调度问题。VNF 映射是寻求在考虑一组请求的服务功能链(service function chain, SFC)^[21]的情况下,根据 VNF 和网络节点中的约束条件,以合适的方式在网络基础设施中分配 SFC 中的 VNF,将其映射到网络节点中。VNF 调度则是在 VNF 映射阶段完成基础上,其目标是在满足 SFC 的约束条件的基础上,最小化整体 SFC 的完成时间。为了更好地理解,本文首先假设在虚拟网络中有 3 个支持 NFV 的节点和由 3 条虚拟链路组成,如图 1 所示。

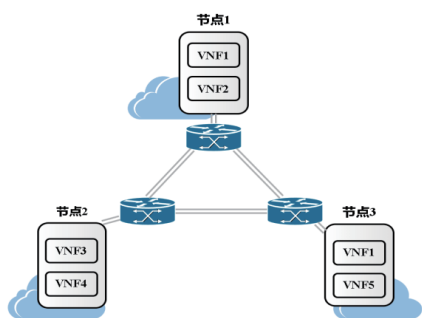


图 1 网络拓扑

假设有 3 条 SFC 进入网络进行调度,分别是 SFC1: VNF1 → VNF3, SFC2: VNF2 → VNF3, SFC3: VNF1 → VNF4,同时 VNF_i 必须映射在具有 VNF_i 功能的节点上才能正常运行,其中 SFC1 与 SFC3 的每个 VNF 处理时间为 1, SFC1 的传输时间为 1, SFC3 的传输时间为 2, SFC2 的每个 VNF 处理时间为 2, 传输时间为 3。假设这 3 条 SFC 已经完成了映射阶段,如图 2 所示。

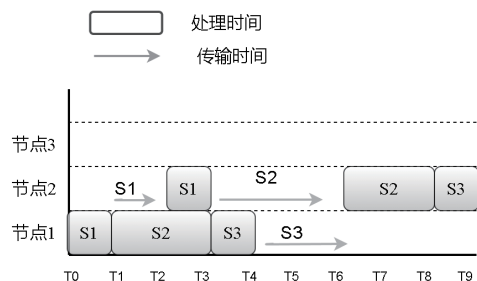


图 2 VNF 映射

当前完成时间为 $T=9$, 此时需要对这 3 条 SFC 进行调度来减少整体的完成时间,如图 3 所示。

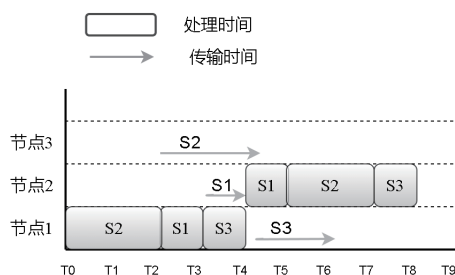


图 3 基于映射的 VNF 调度

可见,经过调度后确实能有效减少整体完成时间,当然这是在结合 VNF 映射阶段基础上进行的,如果将 VNF 映射与 VNF 调度进行结合,那么整体完成时间将减少至 $T=7$ 时效果最好,如图 4 所示。

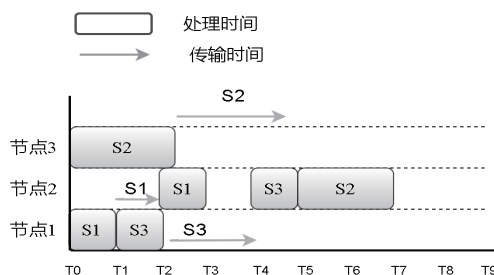


图 4 VNF 映射与调度结合

由此可知,单纯地在原来 VNF 映射后的基础上进行调度无法得到较好的效果,只有当 VNF 映射与调度相结合,即在 VNF 映射阶段就开始考虑调度问题才能最大化效益。为此,本节将描述网络模型及其约束条件。

1.2 网络模型

在物理网络图 $G(N,L)$ 中, N 表示为虚拟节点用来托管和运行不同类型的 VNF, L 表示为连接每两个节点之间的虚拟链路。现有一组 SFC, 并且每个 SFC 都由 j 个不同功能的 VNF 组成,其中 SFC 上的 VNF 都要按照一定顺序进行调度,同时每个 VNF 只能在一个节点上进行实例化操作,而每个节点在同一时刻也只能对一个 VNF 进行处理,即在节点处理 VNF 过程中不能中断。模型参数说明如表 1 所示。

表 1 参数及意义

参数	意义
S	SFC 的数量
V_s	SFC 上 VNF 的个数
n	第 n 个节点
s_j^i	第 i 个 SFC 的第 j 个 VNF
f	VNF 的类型
δ	时间戳
$f_{s_j^i}$	虚拟网络功能 s_j^i 的类型
$p_{s_j^i}^n$	虚拟网络功能 s_j^i 在节点 N 上的处理时延
$T_{s_j^i}^{s_{j+1}^i}$	虚拟网络功能 s_j^i 与 s_{j+1}^i 的传输时延
$K_{s_j^i}^{n\delta} \in (0,1)$	在第 δ 时间戳时第 i 个 SFC 的第 j 个 VNF 是否在节点 N 上运行, 是为 1, 否则为 0

接下来, 考虑优化目标及其所需要的约束。

(1) 确保 SFC 上的 VNF 类型与映射对象节点上的 VNF 实例的类型相同:

$$\sum_{n \in N, s \in S} K_{s_j^i}^{n\delta} f = f_{s_j^i} \quad (1)$$

(2) 表示当 SFC 的第 j 个 VNF 还在处理时第 $j+1$ 个 VNF 不能处理:

$$\sum_{n \in N, s \in S} K_{s_j^i}^{n\delta} \leq 1 - \sum_{n' \in N, s \in S} K_{s_{j+1}^i}^{n'\delta} \quad (2)$$

(3) 表示当节点的 VNF 实例正在处理当前 SFC 的 VNF 时不能再处理其他 SFC 的 VNF:

$$K_{s_j^i}^{n\delta} = 1 - K_{s_{j'}^i}^{n\delta}, n \in N \quad (3)$$

(4) 表示 SFC 的 VNF 同一时刻只能映射在一个节点上:

$$\sum_{s \in S} K_{s_j^i}^{n\delta} = 1 \quad (4)$$

最大完成时间 T 为:

$$T = \sum_{i=1}^S \sum_{j=1}^{V_s} K_{s_j^i}^{n\delta} (P_{s_j^i}^n + T_{s_j^i}^{s_{j+1}^i}) \quad (5)$$

为此, 在满足以上约束的同时, 本文的目的是最小化完成时间 $\min(T)$ 。

2 基于强化学习的鲸鱼算法调度模型

2.1 鲸鱼优化算法

鲸鱼优化算法 (whale optimization algorithm, WOA) 是一种启发式算法^[22], 灵感来源于鲸鱼捕食, 其主要由包围捕食、气泡攻击、搜寻猎物等三个过程组成。

2.1.1 包围捕食

在鲸鱼捕食的过程中, 由于在搜索空间中最优设计的位置并非事先已知, WOA 算法假设当前的最佳候选解即为目标猎物或接近最优解。在确定了最佳搜索代理后, 其他搜索

代理将努力更新它们的位置并逐渐靠近最佳搜索代理。这一行为可以通过以下方程表示为:

$$X_{(t+1)} = X_{(t)}^* - A \times |C \times X_{(t)}^* - X_{(t)}| \quad (6)$$

式中: A 和 C 为系数向量, $X_{(t)}^*$ 为当前最优解, $X_{(t)}$ 为当前解, A 和 C 的计算公式为:

$$A = 2a \times r - a, C = 2 \times r \quad (7)$$

式中: a 为收敛因子在迭代过程中从 2 线性下降到 0, r 为 $[0,1]$ 中的随机数。

2.1.2 气泡攻击

对于气泡攻击来说, 主要有两种方式, 分别是收缩包围机制 (其本质上与包围捕食相同) 和螺旋更新方式, 其方式是通过计算鲸鱼与最优位置的距离创建的一个螺旋等式, 其公式为:

$$X_{(t+1)} = X_{(t)}^* + e^{bl} \times \cos(2\pi l) \times |X_{(t)}^* - X_{(t)}| \quad (8)$$

鲸鱼在一个不断缩小的圆圈围绕着猎物游动, 同时沿着螺旋形路径游动。为了对这种同时发生的行为进行模拟, 假设有 50% 的可能性在收缩包围机制和螺旋模型之间进行选择, 以便在优化过程中更新鲸鱼的位置, 则当概率 p 小于 50% 时对应公式 (6), 当概率大于 50% 时对应公式 (8)。

2.1.3 搜索猎物

与包围机制不同的是, 对于搜索机制来说主要是由系数 A 来确定。当 $|A| > 1$ 时, 鲸鱼以一种随机的方式通过彼此的位置搜索猎物, 不再选择猎物来更新自身位置。当 $|A| \leq 1$ 时, 它们采用了一种群体策略, 随机选取群体中的个体, 用其替代原先的猎物。这一策略的效果是迫使鲸鱼们远离猎物所在的位置, 从而增强 WOA 算法的整体全局寻优能力, 其公式为:

$$X_{(t+1)} = X_{rand} - A \times |C \times X_{rand} - X| \quad (9)$$

式中: X_{rand} 是随机鲸鱼位置向量。接下来具体描述鲸鱼优化算法的改进。

2.2 改进鲸鱼优化算法

2.2.1 编码与解码

在网络系统中, 数量巨大的 SFC 使得在 VNF 调度变得更加复杂。为此, 在用 WOA 解决调度问题时, 编码和解码是非常关键, 一个好的编码可以减少不可行解的产生, 来缩短代码的运行时间, 提高整体的效率。本文采用的是分段式编码过程, 由 MS 和 OS 两个部分组成一个完整的编码, OS 存放所需处理 VNF 的排序向量, MS 是每个 VNF 的节点选择的向量。如图 5 所示, 图中有 3 条 SFC, 每条 SFC 都由两个 VNF 组成, 其中 O(3,1) 表示第 3 个 SFC 中的第 1 个 VNF 最先处理, 选择的节点则对应着 SFC3 区域段的第 1 块, 即为 1 号节点。

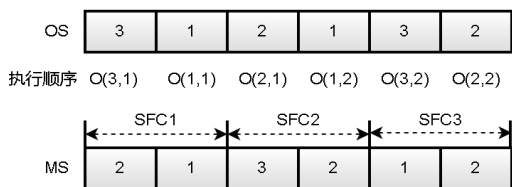


图5 编码图

传统的解码并非每个解向量都对应于不同的处理时间。解空间中的 VNF 与节点处理时间之间依然存在着多对一的映射关系。改进的解码方法可以将结果映射到更好、更少的点，显著缩小搜索空间，减少不可行解的产生。为此，本文引用一种插入式的贪婪解码算法^[23]，其主要是根据 OS 编码部分通过从左向右依次进行解码，同时在不移动已经排布好的 VNF 的前提下，插入到对应节点上最佳时刻安排处理，以此方式直至所有的 VNF 排布完成。如图 6 所示，正常 S3 的第二个 VNF 是在 S2 之后完成的，而进行贪婪的解码将 S3 调度到 S1 之前，这样可以减少整体的完成时间。

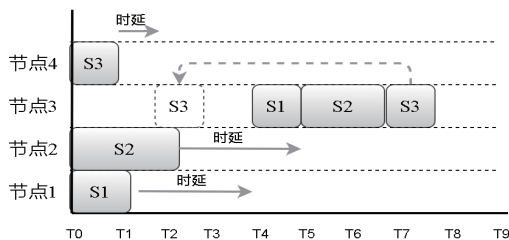


图6 贪婪解码

2.2.2 种群初始化

种群的初始化质量对算法的求解速度与质量有极大的影响。大部分文献采取的随机初始化方法使得初始解的分布随机，机器负载不均匀，种群质量欠佳，从而导致需要更多次迭代与较大的种群规模才能达到较为理想的目标值。为避免浪费大量的计算资源，本文使用了一种 GLR 初始化方法。该方法的 MS 编码包括如下 3 个部分：全局选择（global selection, GS）用于保证加工机器上的负载平衡；局部选择（local selection, LS）保证单个工序加工时间最短或者优先选择机器负荷最小的机器进行加工；随机选择（random selection, RS）保证种群的多样性。OS 编码全部由随机编码构成。

2.2.3 交叉与变异

在改进搜索的过程中，本文引用了遗传算法中的交叉与变异来保持可行解的多样性，同时本文应用了领域来提高局部搜索能力。

交叉：本文对不同的编码层采用不同的交叉方法。对于 OS 编码层，使用 POX 交叉的方式进行编码；对于 MS 编码层，使用多点交叉的方式进行编码。通过以上的 OS 和 MS 的交叉操作，可以提高算法的全局搜索能力，并且保证交叉后的解仍然是可行解，避免产生不合法的解。

变异：为解决变异的问题同时使节点负载均衡化来减少完成时间，本文在变异操作阶段设计了针对网络中节点的负载平衡变异策略，其中节点负载率定义为：

$$L_k = \frac{\sum_k \sum_i P_{ik} X_{ik}}{T} \quad (10)$$

式中： L_k 为负载率， X_{ik} 为在节点 K 上所有的处理 VNF 数， P_{ik} 表示在 K 上每个 VNF 的处理时间， T 为整体的最大完成时间。操作过程如下所示。

步骤 1：对鲸鱼个体进行解码，获得其目标函数 T 及各节点负载率。

步骤 2：对最高负载率节点上的 VNF 进行时间上的排序，并选择处理时间最长 VNF。

步骤 3：查看当前工序可选节点集合，对可选节点负载排序。

步骤 4：OS 编码不变，改变 MS 中当前 VNF 对应的节点，解码求得 T 。

步骤 5：若 T 降低，则保存新的 MS 编码，否则不改变原本 MS 编码。

步骤 6：选择下一高负载 VNF，重复步骤 3 ~ 5，直至该节点上的 VNF 全部遍历完成。

步骤 7：输出保存新的 MS 编码。该过程不仅可以降低负载高节点的负载率，还可以填补低负载率节点空白，实现负载均衡的同时，还能降低整体的完成时间。

2.2.4 领域结构

为了更好地探索局部搜索的准确性和有效性，本文采用变邻域搜索策略来提高当前最优个体的质量，本文同时使用了三种领域结构。N1：在 OS 排序部分中任选两个元素，所选元素需对应不同 SFC 的 VNF，然后对所选元素进行交换操作；N2：在 OS 排序部分中任选两个元素，然后将后一元素插入到前一元素之前的位置；N3：在 MS 分配部分任选 1 至 3 个元素，该元素对应工序的可选节点数应多于 1，然后将该 VNF 分配至不同节点上。在每次邻域搜索操作之后评估新的调度解决方案。如果新的调度方案比原来的调度方案更好，则将新的调度方案设置为原来的调度方案。具体过程如下：（1）设置参数 λ 、 $\lambda_{\max}$ 、 i 、 $i_{\max}$ 和 X ，其中 $\lambda_{\max}$ 为最大迭代次数， λ 为当前迭代次数， i 和 $i_{\max}$ 是参数分别为 1 和 3， X 为当前最优解；（2）对 i 进行循环，当 i 为 1 时使用 N1 领域搜索，当 i 为 2 时使用 N2 领域搜索，当 i 为 3 时使用 N3 领域搜索，如果新的调度解优于 X 则替代，否则 $i+1$ ；（3）如果 $i > i_{\max}$ 则 $t+1$ 并回到步骤 2 继续进行循环；4. 如果 $\lambda > \lambda_{\max}$ 则结束。

2.3 强化学习

强化学习技术在人工智能、机器学习等领域中得到了广

泛的研究与关注^[24]。强化学习是一种特殊的机器学习方法，它主要的工作原理是通过当前环境作为输入给智能体，再通过输出动作反馈给环境，在这个智能体与环境不断交互的过程中将会产生一些奖励，一个好的动作将会产生一些不错的奖励来驱使智能体在后续选择中加强这个动作的趋势，反之则会减弱，以此来达到最佳决策的目的。

Q-learning 学习是强化学习中的一种，其目标是通过迭代更新估计的 Q 值来学习最优策略，以在每个状态下选择最优的动作。 Q 值保存在 Q 值表中，用于记录智能体的学习经历。初始 Q 值表是通过算法生成的零值矩阵，其行数等于状态数量，列数等于动作数量。 Q 值的计算是综合考虑智能体经历的状态、选择的动作和获得的奖励，具体公式为：

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (11)$$

式中： $Q(s_t, a_t)$ 是在 s_t 状态下执行动作 a_t 的估值； α 是学习速率（0 到 1 之间的值），表示新信息相对于旧信息的权重； r_{t+1} 表示执行动作 a_t 后的及时奖励； γ 为折扣因子（0 到 1 之间的值），表示对未来奖励的重视程度； $\max_a Q(s_{t+1}, a_{t+1})$ 表示在执行动作 a_{t+1} 后 Q 表中状态 s_{t+1} 的最高期望 Q 值。

从上述小节可知，在传统的鲸鱼优化算法中首先有两个很重要的参数，分别是 P 和 A ，这两个参数均涉及鲸鱼优化算法的全局搜索和局部搜索能力之间的平衡，但在传统鲸鱼优化算法中，参数 P 采用的是一种等概率阈值法，即阈值为 0.5 并通过与随机生成 0 到 1 之间的值比较来进行策略选择。而这种等概率的策略选择方式可能会使鲸鱼无法选择适合当前群体的捕食策略，使得算法出现收敛速度慢、陷入局部最优等问题。其次就是参数 A ，它与收敛因子 a 线性相关，而传统鲸鱼优化算法的收敛因子 a 随迭代次数的增加而线性递减，这种线性的变化方式不利于算法有效协调全局和局部搜索的能力。为了解决上述两个问题，本文采用了基于强化学习的鲸鱼改进算法，通过环境的改进和智能体的判断，自适应地改变 p 值和 A 值来更好地协调全局搜索和局部搜索能力，同时提高收敛速度。下一小节中首先给出了强化学习的状态特征设置，然后依次介绍了动作和奖励设置，最后介绍了运行流程。

2.3.1 状态设置

在 VNF 调度过程中，本文的目标是最小化完成时间，本文将以种群的适应度值和负载率为重点进行 4 个状态的设置。

(1) 种群的平均适应度值：

$$f = \frac{\sum_{i=1}^N f(x_i^t)}{N} \quad (12)$$

(2) 种群的最佳适应度值：

$$f_{\max} = \max f(x_i) \quad (13)$$

(3) 种群负载标准差：

$$L_{std} = \sqrt{\frac{\sum_{k=1}^m (L_k - L_{ave})^2}{m}} \quad (14)$$

(4) 种群的多样性：

$$D = \frac{\sum_{i=1}^N \left| f(x_i^t) - \frac{\sum_{j=1}^N f(x_j^t)}{N} \right|}{N} \quad (15)$$

式中： $f(x_i^t)$ 表示第 t 代的种群中第 i 个个体的适应度值， $\max f(x_i)$ 表示第 t 代种群中最优适应度的个体。为了提高网络的稳定性和收敛性，本文将数据进行归一化处理，具体公式为 $x_{\text{new}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$ ，其中 x_{new} 为归一化后的新值， x 、 $x_{\min}$ 、 $x_{\max}$ 分别为所需归一化值、对应 x 中的最小值、对应 x 中的最大适应度值。种群的平均适应度值和负载标准差能更好地反映整体的种群状态，提高整体的种群质量，从而获得最佳的个体，因此 W_1 与 W_4 之间的比重应比 W_2 与 W_3 略大一点， $W_1=W_4=0.3$ ， $W_2=W_3=0.2$ ， $W_1+W_2+W_3+W_4=1$ 。

2.3.2 动作设置

本文设置一组动作由两部分组成，由于鲸鱼算法参数 P 是随机由 0 到 1 之间生成的一个值。对于 p 而言，本文设置动作为在 0 到 1 范围内生成 10 个动作，每个动作间隔为 0.1。对于 A 的动作来说，虽然在鲸鱼优化算法中 A 为线性收敛的，但其本身还是与固定值 1 相比来进行判断，为此本文将其设置为一个大小为 2 的动作集合，分别是 [-1,1]。

2.3.3 奖励设置

目标是最小化完成时间，因此需要考虑当前状态 $t+1$ 和上一状态 t 之间变化，通过变化设置奖励告知智能体哪些行动能够获得更高的奖励，其中主要是通过最佳种群适应度 $f_{\max}$ 、平均适应度值 f 来定义。奖励计算过程如下。

算法 1 奖励设置

输入： $f_{\max}$ ， f 。

输出：Reward。

```

1  if  $f_{\max(t+1)} < f_{\max(t)}$  then
2      Reward += 10
3  else if  $f_{\max(t+1)} = f_{\max(t)}$  then
4      if  $f_{(t+1)} < f_{(t)}$  then
5          Reward += 5
6      else if  $f_{(t+1)} = f_{(t)}$  then
7          Reward += 0
8      else
9          Reward -= 5
10     end if
11 else
12     Reward -= 10
13 end if
    
```

基于上述改进的鲸鱼和强化学习结合整体的运行流程如

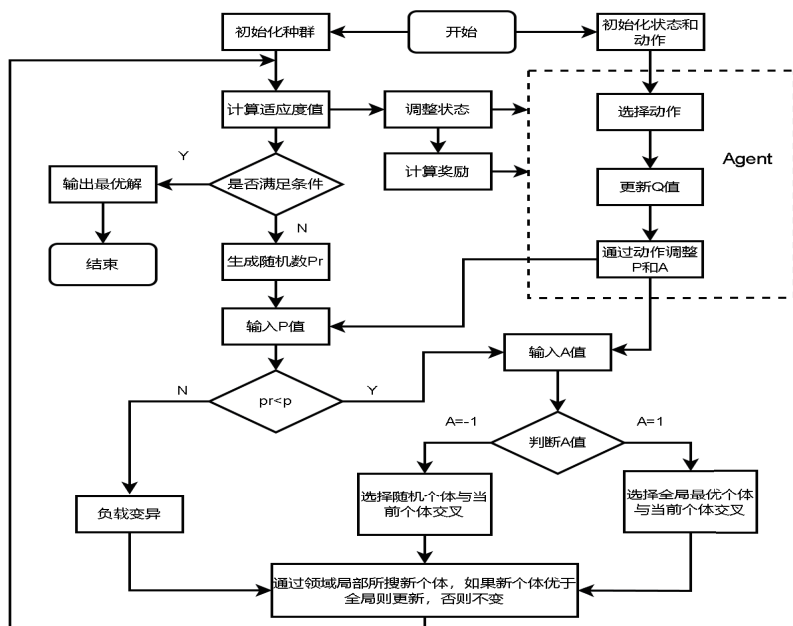


图7 流程图

图7和算法2所示。

算法2 RLWOA 流程

- 1 初始化鲸鱼算法参数, 初始化 RL 中的状态和动作
- 2 初始化种群并根据公式计算种群适应度, 负载标准差, 形成状态 State
- 3 While $t \leq t_{\max}$ do
- 4 通过 State 状态选择动作
- 5 通过动作生成 P 和 A 值
- 6 生成 0 至 1 之间的随机数 P_r
- 7 if $P_r < P$ then
- 8 if $A = -1$ then
- 9 随机个体与新个体进行交叉
- 10 end if
- 11 if $A = 1$ then
- 12 选取全局最优个体与新个体交叉
- 13 end if
- 14 else
- 15 对当前个体进行负载变异
- 16 end if
- 17 对全局最优个体进行局部搜索, 若适应度值减小, 则替换
- 18 更新状态 State+1
- 19 计算奖励 Reword
- 20 更新 Q 值表
- 21 $t = t + 1$
- 22 end

3 仿真与性能分析

为了测试该算法能够有效求解虚拟网络功能调度问

题, 本文将与文献 [25] 中的 GABL 算法和文献 [26] 中的 HGWOA 算法进行比较, 种群数量设置为 100, 迭代次数设置为 50。本实验在 Windows10 系统, i5 处理器上, 使用 MATLAB 编写仿真程序实验运行, 折扣率 γ 设置为 0.2, 学习率 α 设置为 0.75, 贪婪率设置为 0.8。

3.1 实验分析

首先, 本文设置了 6 个全连接的虚拟网络节点, 并且生成 10 条服务功能链进行调度, 每条服务功能链由 4~6 条 VNF 组成, 节点的处理时延为 1~6 ms。

链路的传输时延为 1~5 ms, 先将该 10 条服务功能链放入网络中进行虚拟网络功能。图 8 至图 10 分别是 RLWOA、GABL 和 HGWOA 中 SFC 的整体时延。在图 8 至图 10 中, x 轴表示整体 SFC 的最大完成时间, y 轴表示虚拟节点, 上述方块中不同颜色表示不同的 SFC。其中, 每个方块上的数字表示第几个 SFC 中的第几个 VNF, 例如左上方为 1, 左下方为 1, 则表示为第 1 条 SFC 的第 1 个 VNF。从图中可以看出使用 RLWOA 时最低的完成时延是 42.5 ms, GABL 时最低的完成时延是 45 ms, HGWOA 的最低完成时延是 43.8 ms, 这表明使用 RLWOA 进行虚拟网络功能调度时效果将会更好。

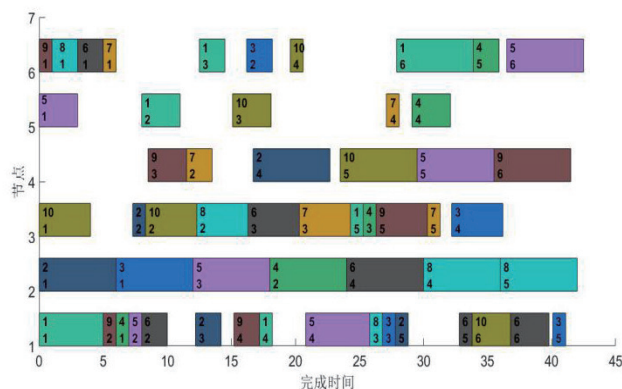


图8 RLWOA 的完成时间

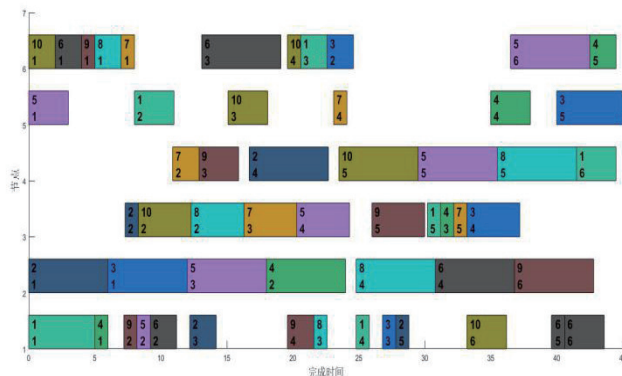


图9 GABL 的完成时间

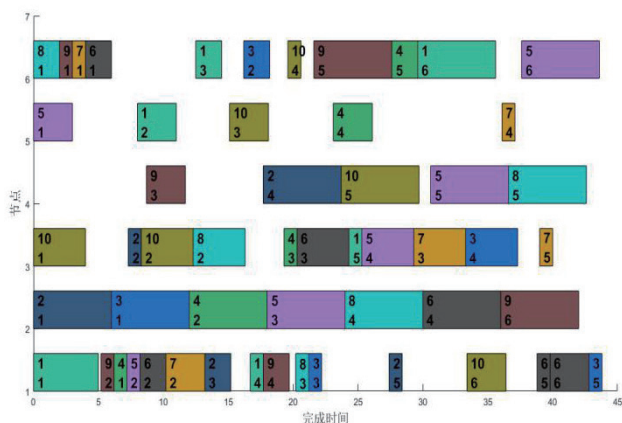


图 10 HGWOA 的完成时间

以下将显示三种算法在实验过程中的迭代图,如图 11 所示。可以看出, x 轴值为 0 时,本文提出的 RLWOA 算法与 GABL 算法和 HGWOA 算法相比值最低,说明 RLWOA 算法具有较好的初始解,也表明该种群初始化过程较好。此外,不难看出,使用 RLWOA 能够有更好的收敛效率,说明使用强化学习进行训练能够加快收敛速度,同时避免陷入局部最优,找到更优秀的解。

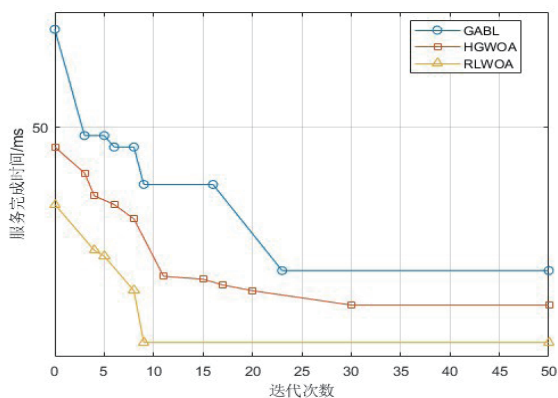


图 11 3 种算法迭代对比

除上述实验以外,本文将采用文献 [27] 的网络结构进行模拟实验。首先,本文将网络由 20 个节点全连接组成,服务功能链数量设置为 10、20、30 等至 100 个,同时由 10 个不同的虚拟网络功能组成上述服务功能链,每个服务功能链最多由 10 个虚拟网络功能组成,且每个虚拟网络功能可以在 1~4 个节点上进行实例化。其次,其节点上的虚拟网络功能的处理时间为 2~10 ms,节点与节点之前的传输时间具有随机行为 1~10 ms,结果如图 12 所示。从图中可以看出,随着服务功能链的不断增加,RLWOA 算法依旧保持着较好的调度效果,使得服务时间有所减少,整体来看比 HGWOA 算法提升了 10% 的性能,与 RLWOA 算法相比大约提升了 16%。由此可见,RLWOA 算法能够更好地找到更优的调度方案,在更短的时间里完成服务功能链。

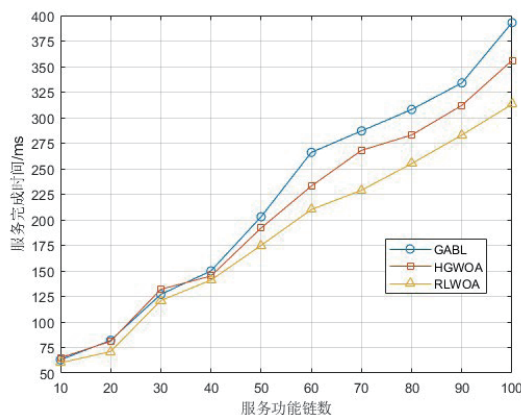


图 12 不同服务功能链下算法比较

4 结语

本文研究了虚拟网络功能调度问题,以最小化完时间为优化目标建立了相应的数学模型,并提出强化学习与鲸鱼优化算法相结合,进行了求解。在所提的基于强化学习的鲸鱼优化算法中所做的工作有对鲸鱼算法的改进,例如种群初始化、编码与贪婪的解码、交叉与变异等,同时加入强化学习自适应调节参数,在整体上提升了算法的调度能力。最后,实验结果表明,本文的算法无论在收敛速度上还是在调度的结果上,均有较好的表现。

在后续的工作中,将研究更多在调度中的实际因素,例如节点资源、链路资源限制、服务功能链上时延约束等方面,同时将研究更新的智能算法,例如 DQN、Dueling Double DQN 等,与虚拟网络功能调度相结合来获得更好的调度效果,以此能够在 5G 的低延迟情况下更好地满足客服的服务需求,提高运营商的收益。

参考文献:

- [1]MIJUMBI R, SERRAT J, GORRICHIO J L, et al.Network function virtualization:state-of-the-art and research challenges[J].IEEE communications surveys &tutorials, 2015, 18(1): 236-262.
- [2]MIJUMBI R, SERRAT J, GORRICHIO J L, et al.Management and orchestration challenges in network functions virtualization[J].IEEE communications magazine, 2016, 54(1): 98-105.
- [3]HAN B, GOPALAKRISHNAN V, JI L, et al.Network function virtualization:challenges and opportunities for innovations[J]. IEEE communications magazine,2015,53(2):90-97.
- [4]ALAMEDDINE H A, SEBBAH S, ASSI C.On the interplay between network function mapping and scheduling in VNF-based networks: a column generation approach[J]. IEEE transactions on network and service management, 2017, 14(4): 860-874.

- [5]ALAMEDDINE H A, QU L, ASSI C. Scheduling service function chains for ultra-low latency network services[C]//2017 13th International Conference on Network and Service Management. Piscataway:IEEE,2017:1-9.
- [6]ALAMEDDINE H A, TUSHAR M H K, ASSI C. Scheduling of low latency services in softwarized networks[J].IEEE transactions on cloud computing, 2019,9(3):1220-1235.
- [7]VIRTUALIZATION N F.NETWORK FUNCTION VIRTUALIZATION[J].NFV_White_Paper3. pdf, 2020.
- [8]HERRERA J G, BOTERO J F.Resource allocation in NFV:a comprehensive survey[J].IEEE transactions on network and service management, 2016,13(3):518-532.
- [9]RIERA J F, ESCALONA E, BATALLE J, et al. Virtual network function scheduling:concept and challenges[C]//2014 International Conference on Smart Communications in Network Technologies. Piscataway:IEEE,2014:1-5.
- [10]BHAMARE D, SAMAKA M, ERBAD A, et al.Multi-objective scheduling of micro-services for optimal service function chains[C]//2017 IEEE International Conference on Communications,[v.4].Piscataway:IEEE,2017:1-6.
- [11]王媛滔,舒兆港,钟一文,等.基于 VNF 实例共享的服务功能链部署算法[J].计算机应用研究,2023,40(6):1806-1811.
- [12]FENG H, SHU Z, TALEB T, et al.An aggressive migration strategy for service function chaining in the core cloud[J]. IEEE transactions on network and service management, 2022, 20(2): 2025-2039.
- [13]SARRIGIANNIS I, RAMANTAS K, KARTSAKLI E, et al.Online VNF lifecycle management in an MEC-enabled 5G IoT architecture[J].IEEE internet of things journal, 2019, 7(5): 4183-4194.
- [14]RIERA J F, HESSELBACH X, ESCALONA E, et al.On the complex scheduling formulation of virtual network functions over optical networks[C]//16th International Conference on Transparent Optical Networks.Piscataway:IEEE,2014:1-5.
- [15]黄睿,张红旗.安全服务链中虚拟网络功能分配与调度算法研究[J].计算机应用研究,2019,36(3):890-895.
- [16]GHOLIPOOR N, SAEEDI H, MOKARI N, et al. E2E QoS guarantee for the tactile internet via joint NFV and radio resource allocation[J].IEEE transactions on network and service management, 2020,17(3):1788-1804.
- [17]LI J, SHI W, YANG P, et al.On dynamic mapping and scheduling of service function chains in SDN/NFV-enabled networks[C]//2019 IEEE Global Communications Conference,[v.1]. Piscataway: IEEE, 2019:1-6.
- [18]HYODO N, SATO T, SHINKUMA R, et al. Virtual network function placement for service chaining by relaxing visit order and non-loop constraints[J].IEEE access, 2019,7:165399-165410.
- [19]ALAHMAD Y, AGARWAL A, DARADKEH T.Cost and availability-aware VNF selection and placement for network services in NFV[C]//2020 International Symposium on Networks, Computers and Communications,[v.2].Piscataway: IEEE, 2020: 630-635.
- [20]FENG H, SHU Z, TALEB T, et al.An aggressive migration strategy for service function chaining in the core cloud[J]. IEEE transactions on network and service management, 2022, 20(2): 2025-2039.
- [21]FU X, YU F R, WANG J, et al.Service function chain embedding for NFV-enabled IoT based on deep reinforcement learning[J].IEEE communications magazine, 2019, 57(11): 102-108.
- [22]NASIRI J, KHIYABANI F M. A whale optimization algorithm (WOA) approach for clustering[J].Cogent mathematics, 2018, 5(1): 1483565.
- [23]张超勇,管在林,刘琼,等.一种新调度类型及其在作业车间调度中的应用[J].机械工程学报,2008(10):24-31.
- [24]WANG H, GU M, YU Q, et al.Adaptive and large-scale service composition based on deep reinforcement learning[J]. Knowledge-based systems, 2019,180(15):75-90.
- [25]刘光远,曹晶仪,庞紫园,等.一种低时延虚拟网络功能映射及调度优化算法[J].西安交通大学学报,2023, 57(2): 121-130.
- [26]李西兴,杨道明,李鑫,等.基于混合遗传鲸鱼优化算法的柔性作业车间自动导引车融合调度方法[J].中国机械工程,2021, 32(8):938-950+986.
- [27]王琛,汤红波,游伟,等.一种 5G 网络低时延资源调度算法[J].西安交通大学学报,2018,52(4):117-124.

【作者简介】

刘智伟(1998—),男,江西萍乡人,硕士研究生,研究方向:虚拟网络功能调度。

舒兆港(1981—),通信作者(email: zgshu@Fafu.Edu.cn),男,江西鄱阳人,博士,副教授,硕士生导师,研究方向:软件定义网络与网络功能虚拟化、机器学习与智能化网络体系结构、网络切片、云计算与边缘计算等。

(收稿日期:2024-03-19)