



研究与开发

基于改进灰狼优化算法的服务功能链映射算法

张岳, 张俊楠, 吴晓春, 洪晨, 周静静

(浙江工商大学信息与电子工程学院(萨塞克斯人工智能学院), 浙江 杭州 310018)

摘要: 随着工业互联网、车联网、元宇宙等新型互联网应用的兴起, 网络的低时延、可靠性、安全性、确定性等方面的需求正面临严峻挑战。采用网络功能虚拟化技术在虚拟网络部署过程中, 存在服务功能链映射效率低与部署资源开销大等问题, 联合考虑节点激活成本、实例化开销, 以最小化平均部署网络成本为优化目标建立了整数线性规划模型, 提出基于改进灰狼优化算法的服务功能链映射(improved grey wolf optimization based service function chain mapping, IMGWO-SFCM)算法。该算法在标准灰狼优化算法基础上添加了基于无环 K 最短路径(K shortest path, KSP)问题算法的映射方案搜索、映射方案编码以及基于反向学习与非线性收敛改进三大策略, 较好地平衡了其全局搜索及局部搜索能力, 实现服务功能链映射方案的快速确定。仿真结果显示, 该算法在保证更高的服务功能链请求接受率下, 相较于对比算法降低了11.86%的平均部署网络成本。

关键词: 网络功能虚拟化; 服务功能链; 资源优化

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2022275

Improved grey wolf optimization algorithm based service function chain mapping algorithm

ZHANG Yue, ZHANG Junnan, WU Xiaochun, HONG Chen, ZHOU Jingjing

School of Information and Electronic Engineering (Sussex Artificial Intelligence Institute),
Zhejiang Gongshang University, Hangzhou 310018, China

Abstract: With the rise of new Internet applications such as the industrial Internet, the Internet of vehicles, and the metaverse, the network's requirements for low latency, reliability, security, and certainty are facing severe challenges. In the process of virtual network deployment, when using network function virtualization technology, there were problems such as low service function chain mapping efficiency and high deployment resource overhead. The node activation cost and instantiation cost was jointly considered, an integer linear programming model with the optimization goal of minimizing the average deployment network cost was established, and an improved grey wolf optimization algorithm based service function chain mapping algorithm (IMGWO-SFCM) was proposed. The algorithm added three major strategies based on acyclic K shortest path (K shortest path, KSP) problem algorithm mapping scheme search, mapping scheme encoding and backward learning and nonlinear convergence improvement on the basis of standard grey wolf optimization algorithm, which better balanced its global search and local search ability, and realized the rapid determination of service function chain mapping scheme. Simulation results show that under the guarantee of higher service function chain request acceptance rate, compared with the comparison algorithm, the average deployment network cost was reduced by 11.86%.

收稿日期: 2022-04-19; 修回日期: 2022-10-20

通信作者: 吴晓春, spring-403@zjgsu.edu.cn

基金项目: 浙江省自然科学基金资助项目(NO.LY19F020002, No.LY19F020006); 浙江省新型网络标准与应用技术重点实验室(No.2013E10012)

Foundation Items: The Natural Science Foundation of Zhejiang Province (No.LY19F020002, No.LY19F020006), Zhejiang Key Laboratory of New Network Standards and Application Technology (No.2013E10012)



tion service function chain mapping (IMGWO-SFCM) algorithm was proposed. Three strategies: mapping scheme search based on acyclic KSP algorithm, mapping scheme coding and improvement based on reverse learning and nonlinear convergence were added to the standard grey wolf optimization algorithm to form this algorithm. The global search and local search capabilities were well balanced and the service function chain mapping scheme was quickly determined by IMGWO-SFCM. Compared with the comparison algorithm, IMGWO-SFCM reduces the average deployment network cost by 11.86% while ensuring a higher service function chain request acceptance rate.

Key words: network function virtualization, service function chain, resource optimization

0 引言

在传统运营商网络中,来自用户的海量服务需要大量专用的硬件设备部署实现。随着工业互联网、车联网^[1]、元宇宙等新型互联网应用的兴起,人们对网络的需求呈指数级增长,各式各样的服务请求日新月异。为了应对大量复杂的服务请求,服务提供商通常需要对一些已经部署的设备进行更新,传统网络功能与专用物理设备的高耦合,给服务提供商带来了较高的硬件成本、人工成本及维护成本。

网络功能虚拟化(network function virtualization, NFV)技术有效解决了上述问题。通过传统的网络功能解耦于传统专用物理设备,并以软件的形式部署在通用设备之上,显著减少了网络服务提供商的运营成本及资本支出。在NFV架构下,服务功能链(service function chain, SFC)由若干有序的虚拟网络功能(virtual network function, VNF)组成,为用户的网络服务功能链请求(service function chain request, SFCR)提供服务,通过编排形成的SFC按照方案映射后即可部署。优秀的部署方案可以有效地降低平均部署网络成本,提高资源利用率,为网络服务提供商更丰厚的利益。

NFV是使能网络重构的关键技术之一。通过NFV管控系统按需编排虚实网络资源,优化部署和迁移虚拟SFC可以实现网络切片和灵活部署,满足端到端的业务体验和高效的网络运营需求。近年虽然出现了大量关于虚拟服务功能链映射部

署问题的研究,但大部分只专注于解决其中一个问题且存在一些弊端。比如文献[2-5]都是使用整数线性规划的方法建模求解SFC映射问题,分别实现了降低端到端时延、减少物理服务器数量和最小化带宽使用率的优化目标,基本思路都是求解装箱问题,但由于其资源配额是静态的,不能反映VNF弹性特性,所以求解SFC映射时存在一定局限性。文献[6-7]提出的算法分别在映射和部署过程中做到带宽资源最小化,但都由于在SFC构建时对底层网络的考虑不足分别造成了部署开销和节点资源的更大损耗。文献[8]提出一种资源感知的服务功能链协同构建和映射算法,考虑了底层网络状态,但对映射顺序的忽视,增加了时延开销。文献[9-11]提到了基于贪心策略的启发式算法的映射部署方法,虽说启发式算法求解速度很快,但是会出现易陷入局部最优的问题。文献[12]提出对资源需求较少的SFC优先映射的方法,做到了减少资源开销,但是同样出现了易陷入局部最优的问题。由此分析,目前映射部署方法主要存在两方面问题:一方面单一的构建方案造成资源开销较高,另一方面启发式的方式易陷入局部最优,因此需要在保障资源优化和时延保障的同时,重视陷入局部最优问题,做到快速进行SFC的映射与部署。

针对上述考虑,本文提出一种基于改进灰狼优化(improved grey wolf optimization, IMGWO)算法的服务功能链映射方法。相关研究已证明,SFC的映射及部署问题是NP难(NP-hard)^[13-15]问题。在算法设计中,本文首先考虑部署过程的资源优

化和时延保障, 针对计算资源和带宽资源建立相关的资源约束, 构建虚拟网络功能实例化部署所需的总网络成本模型, 并将优化目标公式化为一个优化函数。选择利用元启发式的相关群智能算法并改进使其匹配 SFC 映射优化问题就是在多项式时间内找到该问题的较优解。灰狼优化 (grey wolf optimization, GWO) 算法是 Mirjalili 等^[16]于 2014 年提出的一种元启发式的全局搜索算法。迄今为止, 灰狼优化算法已被广泛应用且在不断的改进当中, 主要应用在约束函数优化、流水车间调度问题、多级阈值图像分割、高维优化问题、路径优化等领域。相对其他进化算法, 标准灰狼优化算法能更好地提高优化效率, 并控制搜索方向, 具有较强的全局搜索能力。但是由于标准灰狼优化算法确实无法直接应用于离散问题 (即不能直接应用于 SFC 映射问题中) 并发挥出其优势, 且存在收敛速度较慢及易陷入局部最优解的情况。首先为了实现改进灰狼优化算法在 SFC 映射问题中的适用, 添加了基于无环 K 最短路径 (K shortest path, KSP) 问题算法的映射方案搜索、映射方案编码策略, 其次针对收敛速度和局部最优问题分别添加了基于反向学习与非线性收敛改进的策略。通过添加三大策略改进后的算法得到的映射方案既可以做到平均部署网络成本低于同类型的算法, 同时加强了前期全局寻优能力和后期局部寻优能力, 提高了物理拓扑的服务功能链请求接受率。

1 相关研究现状

1.1 服务功能链映射研究现状

目前服务功能链的优化映射及部署问题通常被建模成为整数线性规划模型或混合整数线性规划模型, 现有的工作根据求解所采用的优化算法类型可以划分为求解器、启发式算法、元启发式算法及强化学习算法四大类。优化算法特点总结见表 1。

常见的求解器有 CPLEX^[17]、CVX^[18] 及 Gurobi^[19]。文献[4,20]将服务功能链的优化映射问题建模成为整数线性规划模型后, 利用 CPLEX 求解器进行了问题的求解。文献[21]利用 Convex (CVX) 求解器求解了所建立的整数线性规划模型。文献[22]则利用 Gurobi 求解了根据最小化服务功能链端到端时延原则而建立的混合整数线性规划模型。此类利用求解器的方法均表现出小规模网络拓扑求解速度快、大规模时速度就会变慢的特点, 无法满足用户在大规模网络拓扑中快速获得映射方案的需求。

近年来, 强化学习算法凭借其解决决策问题的先天优势逐渐在各个领域中得到了应用^[23]。文献[24]提出基于双重深度 Q 网络的 VNF 部署算法, 根据基于阈值的策略部署和释放 VNF 实例。仿真显示, 该算法在负载均衡及吞吐量方面均拥有较好的性能。强化学习算法针对优化目标的通用性主要由设置不同的反馈算法实现, 这对算法设计者的要求颇高。

表 1 优化算法特点总结

优化算法类型	特点
求解器	小规模的网络拓扑时求解速度快, 较为复杂的网络模型时求解速度较慢
强化学习	针对优化目标的通用性主要由设置不同的反馈算法来实现, 对算法设计者要求高
启发式算法	求解速度较快, 但无法保障得到的是全局最优解, 且单种启发式算法适用场景和目标有限制
元启发式算法	启发式算法的改进, 问题在于面对大规模的物理拓扑时易陷入局部最优解



面对大规模网络拓扑中快速求解服务功能链的优化映射及部署问题, 各类启发式算法相继被提出。文献[25]评估最适合当前服务功能链请求的节点及链路资源利用率, 并设计了启发式部署算法, 以最大化服务功能链请求接受率。文献[26]提出了基于特征分解的启发式算法, 通过对服务功能链请求及物理拓扑进行特征分解, 将 VNF 部署后的链路集成到邻接矩阵中, 并在物理拓扑中找到最接近的映射图, 确定最终的服务功能链映射方案, 实现请求接受率的提高及资源开销的优化。启发式算法虽说表现出求解速度快的特点, 但无法保障所得的解为全局最优解, 且单种启发式算法仅适用于特定的问题场景及相关优化目标。

元启发式算法结合了随机算法与局部搜索的策略^[27], 对启发式算法进行了改进。例如, 文献[28]使用了禁忌搜索算法, 随机生成一个初始解, 随后将其与邻居解进行比较, 若表现得更优, 则更新最优解并通过不断迭代得到最终的映射方案, 实现服务功能链部署开销的最小化。文献[29]利用基于改进麻雀搜索算法的服务功能链优化映射算法进行求解, 针对同一时刻到来的多个请求, 进行映射权重的排序, 优先映射权重高的请求。仿真结果显示, 该算法降低了部署开销。但元启发式算法的弊端是面对大规模的物理拓扑时易陷入局部最优解。而本文提出的改进灰狼优化算法的服务功能链映射方法可以通过非线性收敛策略加强算法的前期全局寻优及后期局部寻优能力。

1.2 灰狼优化算法

灰狼优化算法是通过模拟自然界灰狼等级制度和狼群搜索猎物的过程, 即利用灰狼种群内部等级制度, 实现对目标猎物的攻击, 是一种模仿灰狼的社会等级制度和猎食行为的新型群智能优化算法。该算法完成随机种群初始化后按照等级关系划将灰狼分为 4 组, 前 3 组 α 狼、 β 狼、 δ 狼为精英狼, α 狼作为领导者负责捕猎、食物分配等事务决策, β 狼协助 α 狼进行决策, 当 α 狼出

现空缺时接替其位置。 δ 狼听从 α 狼及 β 狼的指令, α 狼和 β 狼可能被降级成为 δ 狼。最底层的 ω 狼负责平衡种群内部的关系。3 组精英狼是适应度最好的 (当前最优解), 并且指导着第 4 组的狼向目标搜索。在优化过程中, 狼群不断包围猎物并更新自己的位置, 式 (1) 为灰狼与猎物的距离, 式 (2) 是灰狼个体的位置更新计算式。

$$C = |C \cdot X_p(t) - X(t)| \quad (1)$$

$$X(t+1) = X_p(t) - A \cdot D \quad (2)$$

其中, D 是灰狼与猎物的距离, t 表示当前的迭代次数, X_p 与 X 分别表示猎物和灰狼的位置向量, A 与 C 是系数向量, 求解如下:

$$A = 2a \cdot r_1 - a \quad (3)$$

$$C = 2r_2 \quad (4)$$

其中, r_1 和 r_2 是 $[0,1]$ 的随机数, a 是平衡 GWO 算法搜索能力的关键参数, 其构成如式 (5) 所示, 在算法迭代的时候由 2 到 0 线性变化, 其决定的 A 直接影响了灰狼个体是否对于猎物发动攻击。

$$a(t) = 2(1 - t/t_{\max}) \quad (5)$$

其中, $t_{\max}$ 表示最大迭代次数, 由此可知 a 的值会随着迭代次数的增加而逐渐减少, 而 a 的减少会导致 $|A|$ 减小的概率变大。 $|A|$ 的大小与灰狼攻击或寻找猎物的关系为, 当 $|A| \leq 1$ 时, 灰狼准备对猎物发起攻击, 表示算法正在局部挖掘, 有利于算法在局部极值的搜索; 当 $|A| > 1$ 时, 灰狼转向搜索其他猎物, 这种发散行为对应全局搜索, 有利于防止算法陷入局部最优。

灰狼能够识别猎物的位置并包围它们, 当灰狼识别得到猎物的位置后, α 狼带领 β 狼及 δ 狼指导狼群包围猎物。在使用灰狼优化算法时, 由于对最优方案也就是猎物的位置并不了解, 需要假设 3 组狼了解猎物的潜在位置。该算法保留迄今为止取得的 3 个最优解, 并根据这三者的位置

判断猎物所处的位置。

灰狼判断得到猎物的位置后, 头狼 α 狼带领 β 狼及 δ 狼指导狼群对猎物进行包围, 因为 α 、 β 、 δ 最靠近猎物位置, 所以利用这 3 匹狼的位置判断猎物大致的方向, 逐渐逼近猎物, 对应的位置更新如式 (6) 所示。

$$\begin{cases} D_\alpha = |C_1 \cdot X_\alpha - X| \\ D_\beta = |C_2 \cdot X_\beta - X| \\ D_\delta = |C_3 \cdot X_\delta - X| \end{cases} \quad (6)$$

其中, X_α 、 X_β 、 X_δ 分别对应狼群所在的位置。根据这 3 组灰狼位置的更新, 可以确定 ω 狼的移动方向和步长, 具体表达式如式 (7) 所示。此外, 可根据相对位置得 ω 狼如式 (8) 所示的最终位置。 ω 狼完成一次位置更新后, 整个种群会根据适应度函数, 对种群中个体的位置进行重新评估, 形成新的 α 、 β 、 δ 、 ω 4 组狼群。直至达到最大迭代次数, 灰狼完成捕猎任务, 此时即完成了对全局最优解的寻优。

$$\begin{cases} X_1 = X_\alpha - A_1 \cdot D_\alpha \\ X_2 = X_\beta - A_2 \cdot D_\beta \\ X_3 = X_\delta - A_3 \cdot D_\delta \end{cases} \quad (7)$$

$$X(t+1) = (X_1 + X_2 + X_3) / 3 \quad (8)$$

2 问题描述和数学建模

本节首先通过一个典型案例说明根据不同的映射方案进行 SFC 的部署将会带来不同的网络成本, 随后介绍网络模型并对问题进行数学建模, 最后通过分析确定映射的相关约束及优化目标。

2.1 问题描述

针对用户在应用程序层发起的完全相同的服务功能链请求 (service function chain request, SFCR), 控制层在收到后需将其编排为服务功能链。根据不同的映射方案进行部署的 SFC 会消耗不同的计算资源、内存资源、带宽资源、节点激活成本及部署成本, 进而使得不同的部署方案在相同的网络拓扑中产生各有差异的网络成本, 此外也会带来不一样的端到端时延。不同映射方案部署对比如图 1 所示。

因此根据服务功能链的映射及部署的 4 个基本原则, 图 1 (a) 包含 3 个 VNF 的简单服务功能链, 图 1 (b) 描述了该服务功能链的两种不同的映射及部署方案, 分别为 SFC^a 与 SFC^b , 其中实线代表映射方案 A, 虚线代表映射方案 B。数据由交换机 A 处流入并由交换机 F 处流出。在映射方案 A 中, 占用网络拓扑中 3 条链路的带宽资源,

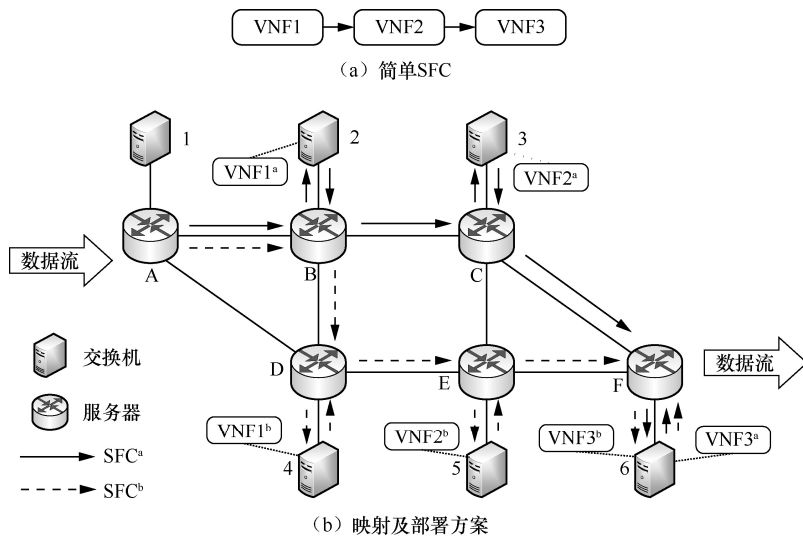


图 1 不同映射方案部署对比



而映射方案 B 占用了网络拓扑中 4 条链路的带宽资源, 并多进行了一次数据交换, 占用了不必要的带宽资源, 因此映射方案 A 的网络成本更佳。

上述两种映射方案说明了, 针对用户的 SFCR 进行不同的服务功能链映射方案的确定及部署, 可使得网络整体性能表现不同, 所带来的网络成本及资源利用率各有差异。具体来说, SFC 中 VNF 的不同节点部署会带来不一样的资源开销及网络成本, 接下来将会对 SFC 的映射方案确定问题进行数学建模。

2.2 网络模型

在物理网络方面, 可用无向图 $G=(N, E)$ 表示底层网络拓扑, 其中 $N=\{n_1, n_2, \dots, n_k\}$ 表示拓扑中的物理节点的集合, $E=\{(n_i, n_j) | n_i, n_j \in N, B_{ij} > 0\}$ 表示拓扑中的物理链路集合, Ei 为与之对应的虚拟链路, 其中 B_{ij} 代表节点 n_i 与 n_j 间的链路带宽容量, 容量大于 0 则代表该链路存在。

在用户网络服务请求方面, 按序到达的 SFCR 表示为 $S=\{1, 2, \dots, f, \dots\}$, 编排形成的 SFC 可以定义为四元组 $f=(V, T_{\max}, R_C, R_B)$, 其中 $V=\{v_1, v_2, \dots, v_n, \dots\}$ 为构成该服务功能链的 VNF 有序集合, 对应的 VNF 实例化于节点之上的部署成本集合为 $P=\{p_1, p_2, \dots, p_n, \dots\}$ 。 $T_{\max}$ 为该服务请求的最大可接受时延, R_C 为该服务功能链所包含的 VNF 对于计算资源的原始需求集合, R_B 为对于链路带宽资源的需求集合。

2.3 约束分析及优化目标

综合考虑网络拓扑中各节点计算资源、链路带宽资源以及 SFC 端到端时延的约束, 根据相关映射方案部署时产生的平均网络成本最小化, VNF 的映射问题可以建模成一个整数规划模型。

首先, 需确保所有服务功能链中的 VNF 均被部署在网络拓扑的一个物理节点中。因此本文定义一个二维决策变量 $D_i^{f, v_m} \in \{0, 1\}$, 当服务功能链 f 中的第 m 个 VNF 部署在物理节点 i 时, $D_i^{f, v_m}=1$, 否则 $D_i^{f, v_m}=0$ 。根据服务功能链顺序执行的特性, 某个 VNF 在被部署到物理节点之前,

需要确保前一个 VNF 已经完成物理节点上的部署, 即 $\sum_{i \in N} D_i^{f, v_m} \geq \sum_{j \in N} D_j^{f, v_{m+1}}, \forall v_m \in V, \sum_{i \in N} D_i^{f, v_{m-1}} \geq \sum_{j \in N} D_j^{f, v_m}, \forall v_m \in V$ 。

在节点资源中, 每个物理节点部署的 VNF 计算资源总和不超过其可用计算资源总量。计算资源也就是确保某个节点上所部署的全部 VNF 的 CPU 开销及内存开销不超过该物理节点的可提供资源。因此 CPU 资源约束如式 (9) 所示, 其中, cpu^{f, v_m} 表示处理服务功能链 f 中的某个 VNF 所带来的 CPU 开销, R_{cpu}^i 代表节点 i 所能提供的最大 CPU 资源。内存资源的约束同理:

$$\sum_{f \in S} \sum_{v_m \in V} \text{cpu}^{f, v_m} \cdot D_i^{f, v_m} < R_{\text{cpu}}^i, i \in N \quad (9)$$

完成对服务功能链的部署后, 还需要考虑网络拓扑中的各条物理链路的带宽资源有限, 服务功能链产生的需求所占用的链路带宽资源需要小于带宽总容量, 相关约束如式 (10), 其中, $\text{map}_{n, n+1}^{f, v, v+1} \in \{0, 1\}$ 是一个二维决策变量, 表示虚拟链路 $(v, v+1)$ 是否被映射到物理链路 $(n, n+1)$ 上, $B_{v, v+1}^f$ 表示服务功能链 f 中不同的 VNF 映射至两个物理节点后之间的带宽开销, $B_{n, n+1}$ 是两个节点之间的链路带宽容量。

$$\sum_{f \in S} B_{v, v+1}^f \cdot \text{map}_{n, n+1}^{f, v, v+1} < B_{n, n+1}, \quad (n, n+1) \in E, (v, v+1) \in Ei \quad (10)$$

由于不同的用户请求对服务功能链的端到端时延有不同的需求, 因此要进行延迟约束的定义。时延类型可分为处理时延、传播时延、传输时延以及排队时延^[30]。可将服务功能链部署后在物理链路上的时延表示为式 (11), 再结合用户的最大可接受时延 $T_{\max}$, 单条服务功能链的综合时延约束条件如式 (12) 所示。

$$d_f = d_f^i + d_{\text{prop}}^{(x, y)} + d_{\text{trans}}^{(x, y)} + d_{\text{que}}^{(x, y)}, f \in S \quad (11)$$

$$\sum_{\substack{(x, y) \in E, \\ (v, v+1) \in Ei}} \text{map}_{(x, y)}^{f, v, v+1} \cdot (d_{\text{que}}^{(x, y)} + d_{\text{trans}}^{(x, y)}) + \sum_{\substack{i \in N, \\ v_m \in f}} D_i^{f, v_m} \cdot d_f^i < T_{\max} \quad (12)$$

基于优化部署网络成本的目标,除了上述几点外还有节点激活开销和 VNF 部署成本。前者来源于对非活跃物理节点的激活,那么产生的总激活开销如式(13)所示, c_i 表示节点 i 的激活成本, $a_i \in \{0,1\}$ 表示节点 i 是否被激活; VNF 部署成本来源于在特定物理节点中一个新的 VNF 的实例化,所有节点产生的总 VNF 部署成本如式(14)所示,其中, $c_p^{v_m}$ 表示属于 p 类 VNF 的 v_m 部署于节点上所带来的部署成本。

$$C_{\text{active}} = \sum_{i \in N} c_i \cdot a_i \quad (13)$$

$$C_{\text{depoly}} = \sum_{i \in N} \sum_{f \in S} \sum_{v_m \in V'} c_p^{v_m} \cdot D_i^{f, v_m} \quad (14)$$

同理,基于式(9)、式(10)可得拓扑中总 CPU、内存和带宽资源开销计算式如式(15)~式(17)所示,在此基础上可以得到式(18)的优化目标函数,为协调各部分成本在资源开销中影响的比重引入权重系数 α 、 β 、 γ 、 δ 、 ε , 区间为 $[0,1]$ 。

$$C_{\text{cpu}} = \sum_{i \in N} \sum_{f \in S} \sum_{v_m \in V'} \text{cpu}^{f, v_m} \cdot D_i^{f, v_m} \quad (15)$$

$$C_{\text{mem}} = \sum_{i \in N} \sum_{f \in S} \sum_{v_m \in V'} \text{mem}^{f, v_m} \cdot D_i^{f, v_m} \quad (16)$$

$$C_{\text{bw}} = \sum_{(n, n+1) \in E} \sum_{(v, v+1) \in E} \sum_{f \in S} B_{v, v+1}^f \cdot \text{map}^{f, v, v+1} \quad (17)$$

$$\min \quad \alpha \cdot C_{\text{active}} + \beta \cdot C_{\text{depoly}} + \gamma \cdot C_{\text{cpu}} + \delta \cdot C_{\text{mem}} + \varepsilon \cdot C_{\text{bw}} \quad (18)$$

3 基于改进灰狼算法的服务功能链映射算法

引言中提到, SFC 部署被证实为 NP 困难问题。通常采用元启发式算法进行求解, 元启发式算法中包括大量群智能算法^[31], 如粒子群算法、蚁群算法、灰狼优化算法等。本文提出 IMGWO-SFCM, 并利用基于无环即 K 最短路径问题 (K shortest path, KSP) 算法的搜索方案提供初始种群的生成范围, 基于映射节点的

方案编码提供灰狼个体与映射方案对应的编码方案。随后通过改进灰狼优化算法, 迭代确定最终方案。

3.1 基于无环 KSP 算法的映射方案搜索策略

为了实现改进灰狼优化算法在服务功能链映射问题中的适用, 需要搜索网络拓扑中可能存在的映射方案, 使改进灰狼优化算法能基于反向学习策略生成初始种群。主要分为两步, 首先需要查找网络拓扑中源节点到目的节点中的前 K SP, 然后需要根据第一步所得的 K 条路径及所需部署 SFC 中 VNF 的具体个数进行映射方案的搜索并得到相关集合。

KSP 问题可以划分为限定无环 KSP 和一般 KSP^[32], 前者要求得到的路径都必须是简单路径, 后者则对路径没有任何限制, SFC 的映射方案搜索问题即限定无环 KSP 问题。目前限定无环 KSP 算法主要有偏离路径算法与改进 Dijkstra 算法, 本文采用文献[33]所提的偏离路径算法进行前 K 条最短路径的搜索。

在第一步得到包含节点数不一的各条路径后, 为确定可能存在的映射方案, 需要对各条路径进一步搜索。搜索的依据是所需部署 SFC 中 VNF 及物理节点的数量, 多个 VNF 可映射在同一物理节点中, 但映射顺序不得异于数据流方向。图 2 所示映射方案搜索过程为例, 包含两个 VNF 的 SFC 在有两个节点的物理链路中有 3 种映射方案, 分别是均映射于节点 1、均映射于节点 2 和按次序分别映射于节点 1 和节点 2。

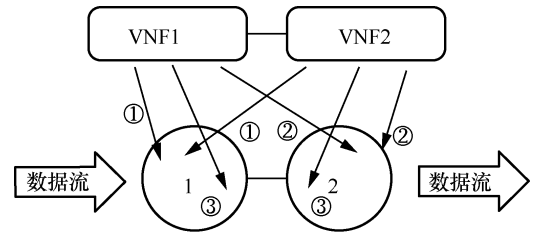


图 2 映射方案搜索过程

算法 1 展示了单链路映射方案搜索算法的具体工作流程, 伪代码如下。算法 1 以递归为核心



思想,如算法1第12行所示,该算法从每条链路的第1个节点开始,进行所有首个VNF映射位置为该节点的映射方案的搜索,同时将相关方案写入映射方案集合。该节点搜索完毕后循环至链路中下一节点。

算法1 单链路映射方案搜索算法

输入: 前 K 条最短链路集合, 服务功能链请求

输出: 映射方案集合 M

FnFunction:Select (链路长度, SFC 长度) 映射方案 = 映射方案+ 当前节点;

if SFC 长度 == 0 then

$M[i][n++].append$ (映射方案);

return

end

while $j < \text{链路长度} + 1$ do

Select (链路长度, SFC 长度-1)

$j = j + 1$

end

for $i \leftarrow 1$ to 链路数量 do

for $j \leftarrow 1$ to 链路长度 do

Select (链路长度, SFC 长度-1)

end

end

return M

3.2 映射方案编码策略

由于GWO一般用于求解连续型问题,不能直接应用于离散型的服务功能链映射^[34],因此要对服务功能链的映射方案进行编码,使灰狼个体与映射方案进行对应。IMGWO-SFCM算法中灰狼个体编码采用与物理节点对应的编码策略,映射解的长度与服务功能链中所需映射的虚拟网络功能个数相等。映射方案编码策略如图3所示,将需要部署的VNF与网络拓扑中的物理节点编号进行对应。可将其对应映射方案编码为(1,1,2,2,3)。

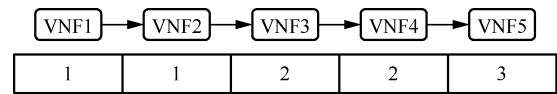


图3 映射方案编码策略

图4描述了该编码策略在多节点映射方案中与灰狼个体的对应关系。一个灰狼个体代表一种服务功能链映射方案,不同的映射方案构成了狼群。图4展示了两种可能的映射方案,5个VNF被映射在6个物理节点构成的网络拓扑中。可以看出方案1的映射方案为(1,2,2,2,3),方案2的为(1,1,4,4,4)。类似方案1、方案2的灰狼个体在不断的迭代中更新优势狼群,最后得到最优个体并输出相应的映射方案。

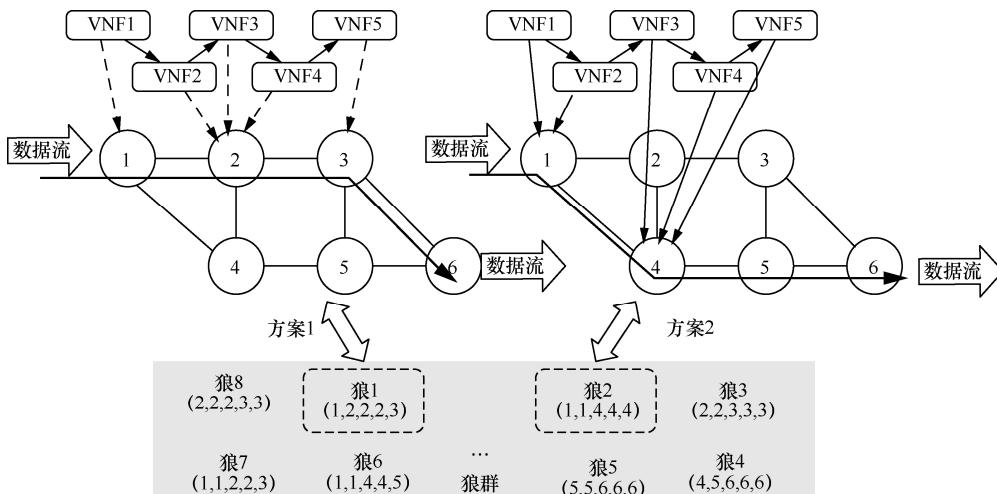


图4 灰狼个体与编码方案的对应

3.3 基于反向学习和非线性收敛的改进灰狼优化算法

标准灰狼优化算法相较于传统元启发式算法具有调节参数少、结构简单、易于实现的优点,已经被广泛运用在多个领域的优化问题中。但仍然存在后期收敛速度慢,可能会陷入与真实最优解相差很大的局部最优中。针对这两方面的问题,通过基于反向学习的种群初始化策略及收敛因子非线性优化策略对标准灰狼优化算法进行了改进,提出了改进灰狼优化(IMGWO)算法。

3.3.1 基于反向学习的种群初始化策略

对于群体智能优化算法,种群在搜索空间内的初始位置分布直接决定了初代种群的环境适应能力,即初代精英狼会直接影响狼群的狩猎效率,若初始精英狼正好在猎物附近,那么狼群可以快速逼近猎物并投入更多的精力用于精准定位猎物位置。在标准 GWO 算法中,初始种群随机初始化生成,无法保证较好的种群多样性^[35],一定程度上限制了算法的寻优性能。为了提高标准 GWO 算法的初始解质量,IMGWO 算法采用基于反向学习的种群初始化策略来生成狼群集合,得到质量较好的初始解,有利于提高算法的收敛性能。

反向学习的主要思想是通过寻找当前解集的反向解集^[36],让当前解集与反向解集竞争后根据结果选取有利于算法迭代的解,反向学习是提高随机搜索算法的有效策略,由反向数、反向点、反向学习 3 部分构成。对于在区间 $[a_i, b_i]$ 的任意实数 x_i ,式(19)定义了反向数 $\bar{x}_i$,式(20)为在 N 维空间中的反向点。

$$\bar{x}_i = a_i + b_i - x_i \quad (19)$$

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N), x_i \in [a_i, b_i] \quad (20)$$

设 x 为解集中的一个候选集, $\bar{x}$ 为反向解, f 为目标函数,在迭代过程中,同时计算候选集与其反向点的适应度值,并根据适应度比较的结果去认定适应度较优的作为候选解,判定过程如

式(21)所示。

$$x = \begin{cases} x, f(x) > f(\bar{x}) \\ \bar{x}, f(x) \leq f(\bar{x}) \end{cases} \quad (21)$$

在映射方案确定的问题中,反向数的获取区间为搜索后得到的映射方案集合,初始灰狼与反向灰狼对应关系如图 5 所示。 a 代表集合中第一个映射方案, b 代表集合中最后一个映射方案, X 为随机生成的其中一个初始灰狼个体, $\bar{X}$ 为其对应的反向灰狼个体。

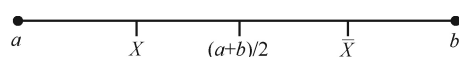


图5 初始灰狼与反向灰狼对应关系

最终基于反向学习策略的灰狼优化种群初始化流程为首先随机生成初始狼群,并根据初始狼群生成反向狼群。随后根据适应度函数确定两个种群中适应度最高的前3个个体并使其成为初始 α 狼、 β 狼、 δ 狼。

3.3.2 收敛因子非线性优化策略

易陷入局部最优解的问题主要来源于全局搜索能力与局部搜索能力之间的平衡。由第3.3节可知,标准 GWO 算法的全局及局部搜索能力取决于参数 $|A|$ 。而 $|A|$ 的值取决于收敛因子 a , a 是随着迭代次数线性降低的。线性降低由于忽略了实际优化时的具体情况,面对复杂、非线性的问题时易陷入局部最优解。为了在迭代前期避免陷入局部最优解,后期在最优解周围尽可能精确的搜索,需保证参数 a 的变化幅度呈现前缓后陡峭,对控制参数 a 执行非线性优化策略。基于该考量,提出了式(22)所示的非线性优化策略。

$$a(t) = 2 \times \left[1 - \frac{e^{t/t_{\max}}}{(e-1)^2} \right] \quad (22)$$

收敛因子对比如图 6 所示。为了直观地展示收敛因子 a 的迭代变化情况,图 6 展示了其与标准线性降低策略相比较的数值曲线。由迭代变化情况可以观察到收敛因子 a 在非线性的优化策略



下,在迭代前期的数值变化较为平缓,可以使参数 $|A|$ 在前期保持较大的值,有助于全局寻优,快速找到全局最优区域。随着迭代次数的增加, a 的变化趋势逐渐加快,使得灰狼更加专注于在全局最优区域内挖掘最优解,实现猎物的围捕。改进后的收敛因子能够更好地平衡算法的全局搜索与局部搜索能力。

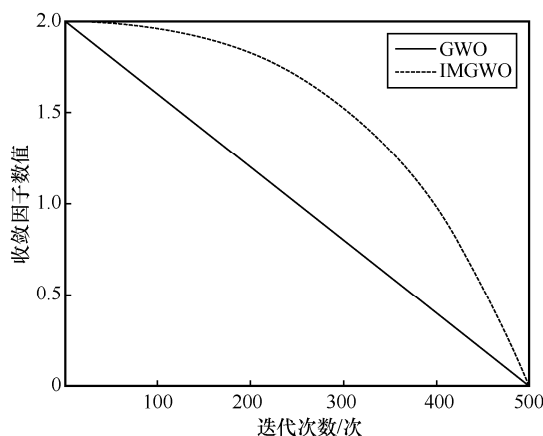


图6 收敛因子对比

3.3.3 IMGWO 算法工作流程

根据前述基于反向学习的种群初始化策略及收敛因子非线性优化策略的改进灰狼优化算法很好地兼顾了全局搜索及局部搜索的能力,加快了算法的整体收敛速度,IMGWO 具体可以分为以下7步。

步骤 1 设定灰狼种群初始化规模,最大迭代次数,初始化各参数。

步骤 2 根据随机初始化的结果利用反向学习生成初始种群。

步骤 3 计算狼群中每个灰狼个体的适应度值,根据适应度值排序选取前三的优势狼。

步骤 4 根据收敛因子非线性优化策略更新收敛因子 a 的值,并根据式(15)~式(18)更新参数数值。

步骤 4 更新灰狼个体位置。

步骤 5 计算灰狼个体位置更新后的种群适应度值,并更新优势狼群和它们的位置。

步骤 7 判断是否达到算法的约束条件或达到最大迭代次数,若满足,则算法结束,输出最优解;若不满足,则返回并重新执行步骤3~步骤6。

3.3.4 IMGWO 算法的测试

为了验证本节所提的改进灰狼优化算法的性能,本节首先采用6个经典单峰测试函数分别为: Sphere、Schwefel 2.22、Schwefel 1.2、Schwefel 2.21、Rosenbrock 和 Quartic,在 MATLAB R2020a 中进行仿真验证,比较 IMGWO 的对象为标准 GWO 算法、PSO 算法^[37]。执行仿真验证时,GWO 算法及 IMGWO 算法的迭代次数设置为 500,初始化种群规模设置为 30,测试空间的维度以给定的基准函数为准。每种算法的运行次数都为 30 次并取运行结果的均值、最优值、最差值作为最终数据进行分析。测试函数结果见表3。数据加下划线代表此项在3种算法中表现最佳,由表3的单峰函数测试结果可知,IMGWO 算法在测试函数 $f_1(x)$ 、 $f_2(x)$ 、 $f_3(x)$ 、 $f_4(x)$ 、 $f_6(x)$ 的寻优解更接近于最优解 0,寻优精度及算法稳定度均优于 GWO 算法。

3.4 映射算法设计

基于改进灰狼优化算法的服务功能链映射算法首先根据网络拓扑及服务功能链服务图进行物理节点资源、SFC 所需计算资源及带宽资源的计算,并根据相应资源状况进行单节点部署可能性的判定。在服务功能链实际部署的过程中会带来相关的部署成本及开启物理节点的激活成本,服务链中的网络功能部署在多个物理节点之上可以带来可用性上的保证,但也会使得服务功能链的端到端时延及部署网络成本显著增加。因此,当网络拓扑中存在物理节点能够支持单节点的服务功能链完全部署时,执行单节点部署操作可以减少物理节点的占用、降低物理节点激活成本、服务功能链端到端时延及部署成本。

表 3 测试函数结果

算法		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$
IMGWO	最优	1.80×10^{-50}	7.27×10^{-31}	5.40×10^{-14}	3.22×10^{-14}	25.449 1	<u>0.000 2</u>
	平均	1.35×10^{-48}	7.55×10^{-30}	3.71×10^{-10}	1.92×10^{-12}	<u>26.479 0</u>	<u>0.000 6</u>
	最差	1.21×10^{-47}	2.68×10^{-29}	5.13×10^{-9}	6.72×10^{-12}	<u>27.186 5</u>	<u>0.001 4</u>
GWO	最优	2.20×10^{-35}	1.31×10^{-20}	4.30×10^{-11}	3.52×10^{-9}	24.581 7	0.000 2
	平均	1.68×10^{-33}	8.68×10^{-20}	5.68×10^{-8}	2.16×10^{-8}	26.782 9	0.001 1
	最差	1.82×10^{-32}	3.95×10^{-19}	1.20×10^{-6}	9.06×10^{-8}	28.758 5	0.002 3
PSO	最优	1.44×10^{-6}	0.000 3	477.774 9	3.101 21	<u>16.661 8</u>	0.013 79
	平均	0.000 2	0.341 5	1 674.914	5.391 991	6 084.7	0.036 41
	最差	0.001 9	10.001 2	8 203.233	7.874 278	90 063.95	0.058 7

算法 2 单节点映射算法**输入:** 网络拓扑 $G=(N,E)$, 服务功能链请求 S **输出:** 映射方案

```

for  $i \leftarrow 1$  to  $N$  do
    if 该节点计算资源 > 当前 SFC 所需计算资源 then
        if 端到端时延及链路带宽满足用户需求 then
            该节点加入备选节点;
        end
    end
end
if 备选节点数 > 1 then
    选取映射后端到端时延最小的物理节点
end
return 单节点映射方案

```

如算法 2 伪代码中的第 1~5 行所示, 判断服务功能链是否能够进行单节点部署首先需对网络环境中各物理节点的资源状况进行遍历, 寻找是否存在拥有足够的资源供服务功能链部署及运营的物理节点, 若存在相关节点, 则进一步判断部署后的端到端时延是否满足用户的最低时延需求。当出现存在多个支持单独部署且均满足最低时延需求的节点情况时, 则根据最小化服务功能

链端到端时延的原则进行映射物理节点的确定。完成单节点映射可能性的判断后, 算法开始执行多节点映射方案确定的相关步骤。首先根据 SFC 的源节点与目的节点, 运用限定无环 KSP 算法及进行前 K 条最短路径的筛选, 随后通过算法 1 进行可能映射方案的进一步搜索, 使得灰狼种群初始化范围得以确定。随后, 通过改进灰狼优化算法相关流程输出最佳映射方案。基于改进灰狼优化的服务功能链映射算法伪代码如算法 3 所示。

算法 3 基于改进灰狼优化的服务功能链映射算法**输入:** 网络拓扑 $G=(N,E)$, 服务功能链请求 S **输出:** 映射方案

```

遍历网络拓扑中各节点的资源状况
if 能够进行单节点映射 then
    确定单节点映射方案  $\leftarrow$  算法 2
    break
else
    基于无环 KSP 算法的路径确定
    单链路映射方案的搜索  $\leftarrow$  算法 1
    基于反向学习的灰狼种群初始化
    计算初始狼群的适应度 (式 (10)) 并确定优势狼群
    while  $t <$  迭代次数 do
        for 每一个灰狼个体 do

```



```

        根据位置更新策略更新自己的位置
    end
    更新收敛因子,A,C
    计算灰狼种群的适应度
    更新优势狼群
     $t = t + 1$ 
end
end
return 最优灰狼个体(多节点映射方案)

```

4 仿真与结果分析

本节研究的优化映射方案主要是针对于服务功能链请求不断到达的情景,在不撤销已部署的服务功能链的情况下对比仿真结果的分析。使用 MATLAB 2020b 软件在配置为 AMD Ryzen 7 5 800H CPU、16.0 GB RAM 的计算机上完成,对 IMGWO-SFCM 算法在服务功能链的请求接受情况、平均部署网络成本等方面的性能进行了评估,并与 Random 随机算法、DP-COA^[38]算法、First-fit 算法、RACCM 算法^[7]、ProvisionTraffic 算法^[18]进行了比较。Random 算法的部署是随机选择拥有足够计算资源和链路带宽资源的网络节点进行节点映射和链路映射,First-Fit 算法的部署是选择第一个碰到的具有足够资源的网络节点进行节点映射和链路映射,DP-COA 算法采用动态规划的思想,将映射问题看作多阶段决策过程进行求解。RACCM 算法采用服务链构建方案与映射方案不断匹配的方式进行求解。ProvisionTraffic 算法为每一条 SFC 请求建立一个多阶段图,将部署 VNF 时所有可能的服务器位置添加到图中。

4.1 仿真设计

为了便于进行仿真分析,算法采用的网络拓扑为典型物理网络拓扑 NSFNET^[39],网络拓扑中某个时间段到达的服务功能链请求数量从 0 到 1 200 依次递增。拓扑参数及仿真过程中所需参数的设置基于研究^[40],见表 4。每条链路的带宽为

1 000 Mbit/s,各节点的 CPU 容量为 100 MIPS,内存为 1 000 Mbit/s,每个节点的假定总激活开销、VNF 部署成本、CPU、内存和带宽的重要性相同,因此目标函数中的权重系数 α 、 β 、 γ 均设置为 1, δ 、 ε 设置为 0.1。在实际情况中,也可以根据实际需求调整这些权重参数,从而达到所需的优化目标。拓扑拥有 14 个节点和 21 条链路,对于 NSFNET,能够承载 8 种类型的 VNF,每个 SFCR 最多包含 3 类 VNF,每个 SFCR 的带宽资源需求量的数值大小满足(0,10]的随机分布,每个 VNF 的具体 CPU 资源开销及内存资源开销各不相同,需进行前期的设定。

表 4 仿真参数

仿真对象	仿真参数	仿真值
NSFNET 拓扑	物理节点数	14
	物理链路数	21
	链路带宽资源	1 000 Mbit/s
	节点 CPU 资源	100 MIPS
	节点内存资源	1 000 MB
	节点激活成本	50
服务功能链请求	单服务功能链 VNF 数	4,5,6
	入口节点及出口节点	1,13
	带宽需求	(0,10] Mbit/s
虚拟网络功能	类型数	8
	CPU 开销及内存开销	每种类型前期确定
	单 VNF 部署成本	1

4.2 性能指标

为了验证网络功能虚拟化环境中 IMGWO-SFCM 的可用性,使用服务功能链请求接受率、网络节点计算资源利用率、链路带宽资源利用率、平均部署网络成本 4 个性能指标作为仿真分析对象。

4.3 仿真结果分析

首先,将服务功能链请求数目作为变量,服务功能链请求接受率如图 7 所示,显示了不同算法在相同服务功能链请求数目下的请求接受率。由图 7 变化趋势可以得到,当服务功能链请求数量不断增加时,会出现部分请求无法得到满足的

情况, 这个问题来源于网络资源总量的限制。当服务功能链请求数量较少时, 网络拓扑中的资源余量充足, 不会出现负载过高的情况。服务请求数量在 600 左右时, 几种算法均能 100% 的实现请求的接受, 但是随着请求数目的增加, 各算法的请求接受率均不同程度地下降。其中, Random 算法下降趋势最明显, 当请求数量达到 1 200 时, 接受率只有 65%。这是因为服务功能链请求数量的增加导致部分节点的资源被大量消耗, 同时产生了大量的资源碎片, 剩余的资源量无法继续容纳新的虚拟网络功能, 越来越多的请求无法被满足, 降低了服务功能链映射的成功率。在相同的下降趋势中, IMGWO-SFCM 算法通过灰狼位置更新策略, 不断为到达的服务功能链请求搜索当前最佳的映射方案, 使请求接受率相较于其他算法始终保持相对较高水平, 该算法的运用可以使得网络拓扑接受更多的服务请求。

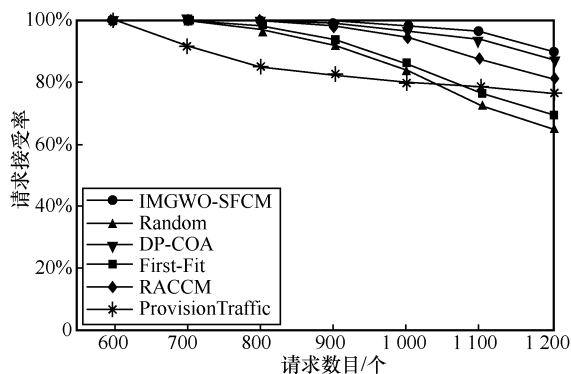


图7 服务功能链请求接受率

更高的请求接受率代表了网络拓扑中依据映射方案部署了更多的服务功能链, 同时也带来了更高的总部署网络成本。为了更好地对IMGWO-SFCM算法在部署成本方面的优化进行体现, 单条服务功能链的平均部署网络成本如图8所示, 显示了单条服务功能链的平均部署网络成本随服务功能链请求数目变化的情况。首先观察变化趋势, 随着请求数目的不断增加, 5种算法的平均部署网络成本在初期达到最高, 其原因在于

前期到达的服务请求在节点激活方面产生的开销较多, 同时较少的请求数目无法均分部署带来的带宽资源开销。在4种算法平均部署网络成本的比较中, IMGWO-SFCM算法的成本均处于最低水平, 在请求数量较大时, 仍然保持了较好的部署网络成本控制。Random算法带来的部署网络成本较高, 资源浪费的情况较为严重, 当请求数目达到1 000后无法接受新的请求, 因此平均部署网络成本保持不变。First-Fit算法带来的部署网络成本呈现持续上涨的趋势, 并在请求数目达到1 000时超过Random和RACCM算法。

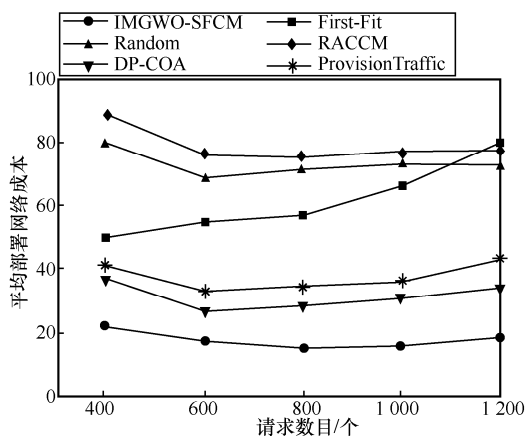


图8 单条服务功能链的平均部署网络成本

整个过程表明了IMGWO-SFCM算法对于网络拓扑中资源使用情况的优化要优于另外几种算法, 可以为用户节省服务功能链的部署成本。

网络节点计算资源利用率如图9所示, 给出了4种算法的网络节点计算资源利用率随服务功能链请求数目变化的折线图。随着请求的不断到达, IMGOW-SFCM算法拥有明显的性能优势, 这是由于其相较于另外几种算法而言拥有较强的全局搜索能力, 充分挖掘满足部署条件的映射方案, 有效缓解了计算资源的碎片化或链路带宽资源的瓶颈造成的影响, 在提高请求接受率的同时提高了网络节点的计算资源利用率。IMGOW-SFCM算法的计算资源利用率相比表现最差的Random算法提高了8%。

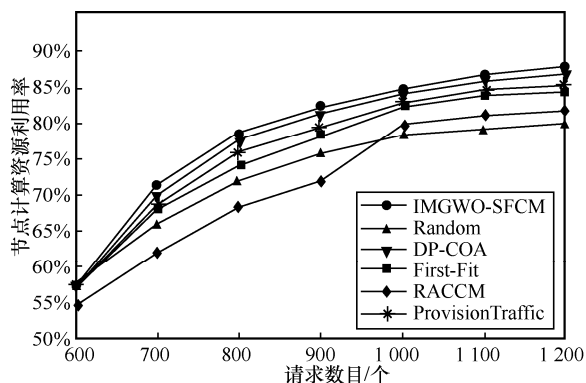


图9 网络节点计算资源利用率

链路带宽资源利用率如图 10 所示, 描述了带宽资源利用率随服务功能链请求数量的变化趋势。Random 算法的随机搜索机制导致了其较高的端到端路由跳数, 在前期占用了较多的链路带宽资源, 后期由于较低请求接受率, 其带宽资源利用率反而表现不佳。随着请求数目的增加, IMGOW-SFCM 算法的带宽资源利用率逐渐达到最高水平, 相较于其他几种算法均有所提升, 尤其比 Random 提升了 5.2%。由于所提算法更多地考虑了单节点中多 VNF 的部署, 减少了不必要的链路带宽消耗, 提高了底层网络的请求接受率, 其链路带宽资源利用率高于另外几种算法。

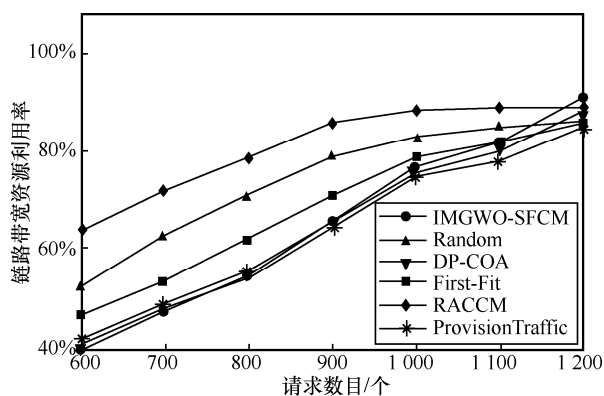


图10 链路带宽资源利用率

5 结束语

本文主要解决了网络功能虚拟化环境中, 在

满足服务功能链时延要求的情况下提供平均部署网络成本最优化的映射方案并进行部署的问题。首先描述了服务功能链部署的具体场景, 并分析了影响平均部署网络成本的因素, 其次建立了在满足用户时延需求及各项网络资源约束下的服务功能链部署总网络成本最小化模型, 最后提出一种基于改进灰狼优化的粒度可变 SFC 部署算法求解该组合优化问题。仿真结果表明, 该算法可以获得比对照算法更高的服务功能链请求接受率及更低的平均网络成本。

本文研究集中在水平方向于对服务功能链进行优化, 考虑服务功能链中不同的虚拟网络功能之间存在不产生相互逻辑影响的情况, 未来可以通过数据流的复制及兼并实现虚拟网络功能的并行执行, 从垂直的角度进行优化, 进一步压缩服务功能链端到端的长度, 降低端到端时延。

参考文献:

- [1] 彭新玉, 周扬, 董振江. 基于车联网远程驾驶的虚拟资源智能协同管理技术[J]. 电信科学, 2020, 36(4): 61-68.
PENG X Y, ZHOU Y, DONG Z J. Intelligent collaborative management technology of virtual resources based on internet of vehicles remote driving[J]. Telecommunications Science, 2020, 36(4): 61-68.
- [2] 李卓峰. 低能耗服务功能链的映射研究[D]. 成都: 电子科技大学, 2018.
LI Z F. Energy-efficient research of service function chain mapping[D]. Chengdu: University of Electronic Science and Technology of China, 2018.
- [3] QU L, ASSI C, SHABAN K. Delay-aware scheduling and resource optimization with network function virtualization[J]. IEEE Transactions on Communications, 2016, 64(9): 3746-3758.
- [4] LUIZELLI M C, BAYS L R, BURIOL L S, et al. Piecing together the NFV provisioning puzzle: efficient placement and chaining of virtual network functions[C]//Proceedings of 2015 IFIP/IEEE International Symposium on Integrated Network Management. Piscataway: IEEE Press, 2015: 98-106.
- [5] MOENS H, DE TURCK F. VNF-P: a model for efficient placement of virtualized network functions[C]//Proceedings of 10th International Conference on Network and Service Management (CNSM) and Workshop. Piscataway: IEEE Press, 2014: 418-423.
- [6] 汤红波, 邱航, 游伟, 等. 基于联合备份的服务功能链可靠

- 性保障的部署方法[J]. 电子与信息学报, 2019, 41(12): 3006-3013.
- TANG H B, QIU H, YOU W, et al. A reliability-guarantee method for service function chain deployment based on joint backup[J]. Journal of Electronics & Information Technology, 2019, 41(12): 3006-3013.
- [7] LI J L, SHI W S, YE Q, et al. Online joint VNF chain composition and embedding for 5G networks[C]//Proceedings of 2018 IEEE Global Communications Conference. Piscataway: IEEE Press, 2018: 1-6.
- [8] 孙士清, 彭建华, 游伟, 等. 5G 网络下资源感知的服务功能链协同构建和映射算法[J]. 西安交通大学学报, 2020, 54(8): 140-148.
- SUN S Q, PENG J H, YOU W, et al. A coordinating composition and mapping algorithm for a service function chain with resource-aware[J]. Journal of Xi'an Jiaotong University, 2020, 54(8): 140-148.
- [9] BOUET M, LEGUAY J, COMBE T, et al. Cost-based placement of vDPI functions in NFV infrastructures[J]. International Journal of Network Management, 2015, 25(6): 490-506.
- [10] SUN Q Y, LU P, LU W, et al. Forecast-assisted NFV service chain deployment based on affiliation-aware vNF placement[C]//Proceedings of 2016 IEEE Global Communications Conference. Piscataway: IEEE Press, 2016: 1-6.
- [11] BECK M T, BOTERO J F. Coordinated allocation of service function chains[C]//Proceedings of 2015 IEEE Global Communications Conference. Piscataway: IEEE Press, 2015: 1-6.
- [12] 程洪闪, 孟欢, 张晓辉. 服务功能链的优化映射策略[J]. 计算机与网络, 2021, 47(8): 54-56.
- CHENG H S, MENG H, ZHANG X H. Optimized mapping strategy of service function chain[J]. Computer & Network, 2021, 47(8): 54-56.
- [13] COHEN R, LEWIN-EYTAN L, NAOR J S, et al. Near optimal placement of virtual network functions[C]//Proceedings of 2015 IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2015: 1346-1354.
- [14] TAJIKI M M, SALSANO S, CHIARAVIGLIO L, et al. Joint energy efficient and QoS-aware path allocation and VNF placement for service function chaining[J]. IEEE Transactions on Network and Service Management, 2018, 16(1): 374-388.
- [15] YUAN B, REN B B. Embedding the minimum cost SFC with end-to-end delay constraint[C]//Proceedings of 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE). Piscataway: IEEE Press, 2020: 2299-2303.
- [16] MIRJALILI S, MIRJALILI S S M, LEWIS A. Grey wolf optimizer[J]. Advances in engineering software, 2014(69): 46-61.
- [17] BLIEKLU C, BONAMI P, LODI A. Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report[C]//Proceedings of the 26th RAMP Symposium. [S.l.:s.n.], 2014: 16-17.
- [18] GRANT M, BOYD S. CVX: MATLAB software for disciplined convex programming, version 2.1[Z]. 2014.
- [19] OPTIMIZATION G. Gurobi optimizer reference manual[Z]. 2020.
- [20] BARI F, CHOWDHURY S R, AHMED R, et al. Orchestrating virtualized network functions[J]. IEEE Transactions on Network and Service Management, 2016, 13(4): 725-739.
- [21] TAJIKI M M, SALSANO S, SHOJAFAR M, et al. Energy-efficient path allocation heuristic for service function chaining[C]//Proceedings of 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). Piscataway: IEEE Press, 2018: 1-8.
- [22] CZIVA R, PEZAROS D P. On the latency benefits of edge NFV[C]//Proceedings of 2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems. Piscataway: IEEE Press, 2017: 105-106.
- [23] 陈学松, 杨宜民. 强化学习研究综述[J]. 计算机应用研究, 2010, 27(8): 2834-2838, 2844.
- CHEN X S, YANG Y M. Reinforcement learning: survey of recent work[J]. Application Research of Computers, 2010, 27(8): 2834-2838, 2844.
- [24] PEI J N, HONG P L, PAN M, et al. Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks[J]. IEEE Journal on Selected Areas in Communications, 2019, 38(2): 263-278.
- [25] KUO T W, LIOU B H, LIN K C J, et al. Deploying chains of virtual network functions: on the relation between link and server usage[J]. IEEE/ACM Transactions on Networking, 2018, 26(4): 1562-1576.
- [26] MECHTRI M, GHRIBI C, ZEGHLACHE D. A scalable algorithm for the placement of service function chains[J]. IEEE Transactions on Network and Service Management, 2016, 13(3): 533-546.
- [27] ABDEL-BASSET M, ABDEL-FATAH L, SANGAIAH A K. Metaheuristic algorithms: a comprehensive review[M]//Computational Intelligence for multimedia big data on the cloud with engineering applications. Amsterdam: Elsevier, 2018: 185-231.
- [28] MIJUMBI R, SERRAT J, GORRICO J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions[C]//Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft). Piscataway: IEEE Press, 2015: 1-9.
- [29] 朱国晖, 景文焕, 李世昌. 基于改进麻雀搜索算法的服务功能链优化映射算法[J]. 计算机应用研究, 2022, 39(7): 2120-2123, 2131.
- ZHU G H, JING W H, LI S C. Optimized mapping algorithm of service function chain based on improved sparrow search algorithm[J]. Application Research of Computers, 2022, 39(7): 2120-2123, 2131.



- [30] DWARAKI A, WOLF T. Adaptive service-chain routing for virtual network functions in software-defined networks[C]// Proceedings of the 2016 Workshop on Hot Topics in Middleboxes and Network Function Virtualization. [S.l.:s.n.], 2016: 32-37.
- [31] 刘雪, 田云娜, 田园. 群智能算法研究综述[J]. 信息与电脑(理论版), 2021, 33(24): 63-69.
- LIU X, TIAN Y N, TIAN Y. A survey of swarm intelligence methods[J]. China Computer & Communication, 2021, 33(24): 63-69.
- [32] 徐涛, 丁晓璐, 李建伏. K 最短路径算法综述[J]. 计算机工程与设计, 2013, 34(11): 3900-3906, 3911.
- XU T, DING X L, LI J F. Review on K shortest paths algorithms[J]. Computer Engineering and Design, 2013, 34(11): 3900-3906, 3911.
- [33] HERSHBERGER J, MAXEL M, SURI S. Finding the K shortest simple paths: a new algorithm and its implementation[J]. ACM Transactions on Algorithms (TALG), 2007, 3(4): 45.
- [34] GUPTA S, DEEP K. Cauchy grey wolf optimiser for continuous optimisation problems[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2018, 30(6): 1051-1075.
- [35] GAIDHANE P J, NIGAM M J. A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems[J]. Journal of Computational Science, 2018(27): 284-302.
- [36] XIA X W, LIU J N, LI Y X. Particle swarm optimization algorithm with reverse-learning and local-learning behavior[J]. Journal of Software, 2014, 9(2): 350-357.
- [37] MARINI F, WALCZAK B. Particle swarm optimization (PSO). A tutorial[J]. Chemometrics and Intelligent Laboratory Systems, 2015, 149: 153-165.
- [38] 刘昀. 虚拟网络功能资源分配与服务功能链路由研究[D]. 合肥: 中国科学技术大学, 2020.
- LIU Y. Virtual network function resource allocation and service function chain routing[D]. Hefei: University of Science and Technology of China, 2020.
- [39] MILLS D L, BRAUN H. The NSFNET backbone network[C]//Proceedings of the ACM Workshop on Frontiers in Computer Communications Technology - SIGCOMM '87. New York: ACM Press, 1988: 191-196.
- [40] PEI J N, HONG P L, XUE K P, et al. Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(10): 2179-2192.

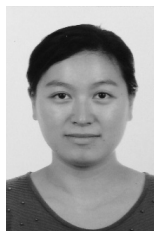
[作者简介]



张岳(1998-), 女, 浙江工商大学硕士生, 主要研究方向为知识图谱、新一代网络技术架构。



张俊楠(1997-), 男, 浙江工商大学硕士生, 主要研究方向为新一代网络技术架构。



吴晓春(1983-), 女, 博士, 浙江工商大学高级实验师、硕士生导师, 主要研究方向为新一代网络技术架构、软件定义网络、网络功能虚拟化、人工智能与网络安全的结合等。

洪晨(1998-), 女, 浙江工商大学硕士生, 主要研究方向为知识图谱、新一代网络技术架构。

周静静(1980-), 女, 博士, 浙江工商大学副教授、硕士生导师, 主要研究方向为新一代网络技术架构、软件定义网络、网络流量建模与分析、大数据处理、深度学习等。