

5G 网络下资源感知的服务功能链 协同构建和映射算法

孙士清, 彭建华, 游伟, 李英乐

(中国人民解放军战略支援部队信息工程大学信息技术研究所, 450002, 郑州)

摘要: 为了充分发挥网络功能虚拟化带来的灵活性优势,实现 5G 场景下定制化服务的高效部署,提出了一种资源感知的服务功能链协同构建和映射算法。该算法首先根据服务请求构建出服务功能链集合,然后对集合中的服务链进行映射,最终求解出资源优化的服务链部署方案。在服务功能链构建阶段,采用基于广度优先搜索的服务功能链构建算法,得到服务功能链所有的构建方案;在服务链映射阶段,采用双层编码方法,将服务功能链构建方案和映射方案进行混合编码,然后利用改进的遗传粒子群算法对问题进行求解,计算出符合当前底层网络状态的服务功能链构建方案和映射方案。仿真结果表明,与 TASAR 算法相比,所提出的协同构建和映射算法服务请求接受率提高了 9%,带宽资源开销降低了 13%,节点资源开销提高了 9%。

关键词: 5G 网络;服务功能链;网络功能虚拟化;虚拟网络功能;粒子群算法

中图分类号: TN915.81 **文献标志码:** A

DOI: 10.7652/xjtuxb202008018 **文章编号:** 0253-987X(2020)08-0140-09



A Coordinating Composition and Mapping Algorithm for a Service Function Chain With Resource-Aware

SUN Shiqing, PENG Jianhua, YOU Wei, LI Yingle

(Institute of Information Technology, PLA Strategic Support Force Information Engineering University, Zhengzhou 450002, China)

Abstract: A coordinating composition and mapping algorithm for a service function chain with resource-aware is proposed to give full play to the advantages of flexibility brought by the virtualization of network functions and to realize the efficient deployment of customized services in 5G scenarios. The algorithm firstly constructs a service function chain set according to service requests, then maps the service chains in the set, and finally the service chain deployment scheme with resource optimization is solved. A service function chain generation algorithm based on breadth-first search is used to obtain all possible composition schemes of the service function chain in the composition stage. The service function composition schemes and the mapping schemes are coded together by a double-layer coding method in the mapping stage, and then an improved genetic particle swarm optimization algorithm is used to calculate the optimal service function composition scheme and mapping scheme in line with the current substrate network. Simulation results and a comparison with the TASAR algorithm show that the service request acceptance rate of the proposed coordinating composition and mapping algorithm increases by 9%, the bandwidth resource overhead is reduced by 13%, and the node resource overhead is

increased by 9%.

Keywords: 5G network; service function chain; network function virtualization; virtualized network function; particle swarm algorithm

国际电信联盟(ITU)定义了5G的三大应用场景,分别为增强移动宽带(eMMB)、低时延高可靠通信(uRLLC)、海量机器类连接(mMTC)^[1]。5G将赋能各垂直行业,为其提供定制化的网络服务^[2]。然而,由于传统电信核心网网元中网络功能和专有硬件紧耦合的特性,导致硬件升级和网络服务的生命周期管理过程非常复杂^[3],这不仅造成运营商利润的损失,而且阻碍了电信行业的创新^[4],难以满足万物互联时代的服务需求。

为解决上述问题,运营商致力于将虚拟化技术应用到电信行业,提出了网络功能虚拟化(NFV)^[5]的概念。网络功能虚拟化是指将网络功能与专有硬件进行分离,并采用软件实现虚拟网络功能(VNF)。NFV可以根据网络需要在不同的位置上进行实例化,如数据中心、网络节点和用户终端设备(CPE)等^[6]。不同的NFV按照一定策略连接起来,构成了服务功能链(SFC)^[7]。不同的业务流量具有差异化的服务需求,因此,运营商需要对不同的业务流量进行分类,制定不同的路由策略,通过不同的服务功能链以满足相应的服务请求。

5G将实现按需灵活的网络服务部署方式,因此,服务功能逻辑顺序未知场景下的服务功能链的构建和映射将是5G管理和编排平面需要解决的重点问题^[8]。服务功能链构建是指编排不同的网络功能形成NFV转发图或服务功能链,以提供网络服务^[9],解决的是服务功能之间的逻辑关系问题。服务功能链的构建受3个方面的影响:①NFV之间的依赖关系,例如,互联网安全协议(IPSec)解密器通常部署在NAT网关之前^[10],但虚拟专用网络(VPN)代理在防火墙前后部署均可^[11];②NFV对流量速率的改变,例如Citrix Cloud Bridge广域网加速器在进行流量转发之前会进行流量压缩,可以使流量降低80%,而BCH(63,48)编码器由于校验和开销则会使流量增加30%^[12];③NFV部署位置,受到网路节点功能和性能的限制。服务功能链映射是指将服务功能链映射至底层物理节点,选择NFV承载服务链中对应的网络功能,以满足网络服务需求^[13]。如何根据服务请求构建服务功能链并进行高效映射已经成为学术界关注的热点问题。大量文献专注于解决其中一个问题,例如文献[14]将服务

功能链的构建问题建模为整数线性规划问题,以求得带宽需求最小的服务功能链构建方案。文献[15-16]提出了基于强化学习的服务功能链映射方法。少量文献将服务链构建问题和映射问题同时进行研究^[17]。文献[9]提出了在线服务功能链联合构建和映射算法,首先以构建出服务功能之间带宽需求最小的服务链,然后对其进行映射。该方法一方面将服务链构建和映射问题独立进行求解,服务链构建过程中忽略了底层网络中VNF实例的分布以及负载状态等信息,生成的服务功能链可能需要较大的部署开销;另一方面在映射过程中未考虑底层链路剩余带宽,所构建的服务功能链可能会部署失败。文献[18]提出了一种基于回溯法的服务功能链联合构建和映射方法,该方法从当前部署节点出发,采用贪婪策略部署下一服务功能,在部署过程中未考虑整个底层网络的状态,容易陷入局部最优。文献[19]提出一种带宽资源最小化的服务功能链部署方法,该方法针对服务功能之间无依赖关系、部分依赖关系和全依赖关系这3种情况设计了不同的服务功能链部署方法,但该文献没有考虑不同的服务功能链构建方案造成链路中流量变化对节点资源消耗的影响。

当前研究方法存在两个方面问题:一方面,在当前两阶段的部署方法中单一服务功能链构建方案对部署问题造成了约束,缩小了问题求解的解空间;另一方面,启发式的方式容易陷入局部最优。为此,本文提出了一种资源感知的服务功能链协同构建和映射算法(RACCM)。在服务功能链构建阶段,采用广度优先搜索算法生成符合服务请求中服务功能之间依赖关系的所有服务功能链构建方案;在服务功能链映射阶段,设计了基于双层编码的服务功能链构建和映射编码方式,充分利用粒子群算法(PSO)全局寻优的能力,实现服务功能链部署方案的快速求解。

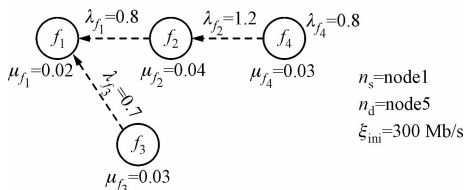
1 问题描述和模型建立

1.1 问题描述

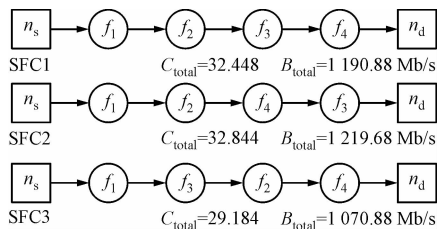
当租户需要向客户提供数据服务时,运营商需要根据其传输的数据流量的业务类型制定不同的网络功能路由策略,以提供网络服务。ETSI指出,

VNF 转发图由服务提供商或者其系统集成合作伙伴进行开发,并且可以由行业指定模板进行定制^[20]。根据文献[9,17-18]的研究,一个服务请求的部署过程可以分为服务链构建和映射两个过程。

如图 1a 所示:一个服务请求包含了源节点 n_s 、目的节点 n_d 、初始流量速率 ξ_{ini} 、服务功能需求 4 个主要组成部分;服务功能需求中除了所需的服务功能类型外,还规定了不同类型的服务功能之间的依赖关系,图中服务功能 f_2 指向 f_1 的虚线代表 f_2 依赖 f_1 ,在构建服务链时 f_2 位于 f_1 之后; λ_f 为服务功能输出的流量与输入流量比值, μ_f 为处理每 1 Mb/s 流量所占用的处理单元数量。图 1b 为根据该服务请求中服务功能之间的依赖关系所生成的服务功能链, C_{total} 为该服务功能链所需的处理单元数量, B_{total} 为该服务链中服务功能之间带宽总需求。若不考虑底层网络中 VNF 实例的分布情况,服务链 SFC3 为资源需求最小的服务功能链构建方案。



(a) 服务请求



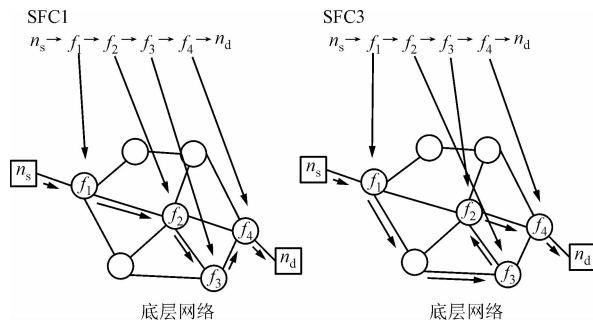
(b) 由图 1a 构建出的服务链

图 1 服务请求及其对应的服务功能链

图 2 为不同的服务功能链构建方案实际映射情况, SFC1 实际映射带宽开销为 1 190.88 Mb/s, SFC3 实际映射带宽开销为 1 310.88 Mb/s, 虽然 SFC3 服务功能之间带宽总需求小于 SFC1, 但其实际映射带宽开销大于 SFC1。因此, 在进行服务功能链部署时, 构建和映射过程是紧密关联的, 应当同时考虑底层网络中 VNF 实例的分布情况、带宽资源等底层网络状态, 才能获得资源优化的部署方案。

1.2 模型建立

1.2.1 网络模型 底层网络采用有向图 $G=(V, E)$ 进行表示, 其中 V 为物理节点集合, E 为物理链路集合。每个物理节点上附着的一个或者多个服务



(a) SFC1 的映射情况

(b) SFC3 的映射情况

图 2 不同的服务功能链构建方案实际映射情况

器, 同时具有流量转发和运行多个不同类型的 VNF 的能力, 且能够虚拟化出多个虚拟处理器。对每个物理节点 $v \in V$, 其剩余的虚拟处理器数量记为 C_v , 节点数据处理时延为 t_v 。设 F 为网络中 VNF 类型集合, 对于每种类型的服务功能 $f \in F$, 其流量改变率为 λ_f , 为 VNF 输出流量与输入流量之比, 每 1 Mb/s 流量所需的虚拟处理器数量开销为 μ_f 。 F_v 为节点 v 能够承载的网络功能集合, $h_{f,v}=1$ 表示类型为 f 的 VNF 可以运行在节点 v 上, 同时定义 V_f 为网络功能 f 可以部署的节点集合。节点 v 到节点 w 的链路 $e_{v,w} \in E$ 为有向边, 其剩余带宽容量为 $B_{v,w}$, 传输时延为 $t_{v,w}$ 。底层视图中两个节点之间的最短路由路径集合记为 $P_{shortest}$ 。

每个服务请求可以由一个五元组 $\langle v_s, v_d, F_{req}, \xi_{ini}, T_{req} \rangle$ 进行表示, 其中 v_s 和 v_d 分别为业务数据流量的源节点和目的节点, F_{req} 为所需的服务功能类型集合, ξ_{req} 为初始流量速率, T_{req} 为服务请求时延阈值。服务请求中规定了 F_{req} 中不同类型的 VNF 之间的约束关系, 其具有传递性和非对称性^[19]。以图 1a 为例, 传递性是指若 f_2 依赖 f_1 、 f_4 依赖 f_2 , 则 f_4 依赖 f_1 ; 非对称性是指若 f_2 依赖 f_1 , 则 f_1 不依赖 f_2 。定义依赖矩阵 D , 当 f_2 依赖 f_1 时, $D(f_1, f_2)=1$, 则图 1a 中服务请求的依赖矩阵可以表示为

$$D = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

当服务功能 f 位于服务链中第 i 位时, 设 $p_{f,i}=1$, 否则为 0, 特别地, $p_{v_s,0}=1$ 。本文仅考虑单播通信。

1.2.2 问题建模 定义网络功能部署变量 $q_{f,v}$, 其值为 1 时表示所需的服务功能 f 映射在节点 v , 由其上对应类型的 VNF 实例承载所需的服务功能,

否则为0;定义虚拟链路映射变量 $l_{f,g,v,w}$,当相邻的两个服务功能 f, g 之间的路由路径经过底层链路 $e_{v,w}$,且流量由链路的端点 v 到 w 时,其值为1,否则为0。本文仅对节点所能承载的 VNF 类型进行约束,有关 VNF 处理能力中的自适应调整方面,文献[21]已经进行了较充分的研究,本文不再对其进行探讨。对于任何成功部署的服务功能链,均要满足以下约束条件

$$\sum_{v \in V} q_{f,v} = 1, \quad \forall f \in F_{\text{req}} \quad (2)$$

$$\sum_{w \in V} l_{f,g,v,w} = 1, \quad \forall f, g \in F_{\text{req}}, v \in V \quad (3)$$

式(2)和式(3)为节点部署约束,式(2)表示服务请求中所有的服务功能均能部署在底层节点上,且仅能部署在一个节点上。为了防止流量分割带来的性能下降,式(3)表示对从节点 v 流出的流量,仅能从一条物理链路流出。

服务功能链构建约束如下

$$\sum_{f \in F_r} p_{f,i} = 1, \quad \forall i \in (1, 2, \dots, |F_{\text{req}}|) \quad (4)$$

$$\sum_{i=1}^{|F_r|} p_{f,i} = 1, \quad \forall f \in F_{\text{req}} \quad (5)$$

$$i < j, \quad p_{f,i} = 1 \wedge p_{g,j} = 1 \wedge D(f, g) = 1 \quad (6)$$

式(4)表示服务链中任一位置仅能有一种网络功能;式(5)表示服务链中没有重复的网络功能;式(6)表示若服务功能 g 依赖于 f ,则流量需先经过 f 。

容量约束如下

$$\sum_{i=1}^{|F_{\text{req}}|} p_{f,i} \mu_f q_{f,v} h_{f,v} \xi_{\text{ini}} \prod_{k=0}^{i-1} \left(\sum_{g \in F_{\text{req}}} p_{g,k} \lambda_g \right) \leq C_v \quad (7)$$

$$\forall v \in V; \forall f, g \in F_{\text{req}}$$

$$\sum_{i=0}^{|F_r|} \xi_{\text{ini}} p_{f,i} l_{f,g,v,w} \prod_{k=0}^i \left(\sum_{x \in F_{\text{req}}} p_{x,k} \lambda_x \right) \leq B_{v,w} \quad (8)$$

$$\forall v, w \in V; \forall f, g, x \in F_{\text{req}}; p_w^k = 1$$

式(7)表示节点需要有足够计算资源处理需要承载的数据流量;式(8)表示对于路由路径中的每条链路要有足够的剩余带宽。

路由约束如下

$$\sum_{w \in V} l_{f,g,v,w} - \sum_{w \in V} l_{f,g,w,v} = \begin{cases} 1, & q_{f,v} = 1 \wedge q_{g,v} \neq 1 \\ -1, & q_{g,v} = 1 \wedge q_{f,v} \neq 1 \\ 0, & \text{其他} \end{cases}$$

$$\forall v \in V; \forall f, g \in F_{\text{req}} \quad (9)$$

式(9)表示在 f 到 g 的路由路径中,除了路径端点外,中间节点的流出链路数量与流入链路数量相等。

时延约束如下

$$\sum_{f \in F_{\text{req}}} q_{f,v} t_v + \sum_{f \in F_{\text{req}}} l_{f,g,v,w} t_{v,w} \leq T_{\text{req}} \quad (10)$$

$$\forall v \in V, e_{v,w} \in E$$

式(10)表示服务链的部署方案应当满足服务请求的时延阈值。

本文以最小化服务请求的部署开销为优化目标。对于任一服务请求 o_{req} ,其部署开销如下

$$\text{cost}(o_{\text{req}}) = \alpha \sum_{v \in V} \sum_{i=1}^{|F_{\text{req}}|} p_{f,i} \mu_f \left(\xi_{\text{ini}} \prod_{k=0}^{i-1} p_{g,k} \lambda_g \right) q_{f,v} h_{f,v} + \beta \sum_{s \in V} \sum_{i=0}^{|F_{\text{req}}|} \left(\xi_{\text{ini}} \prod_{k=0}^i p_{g,k} \lambda_g \right) p_{f,i} l_{f,g,v,w} \quad (11)$$

式中:第一项为计算资源部署开销;第二项为带宽资源部署开销; α 和 β 分别为对应的归一化权重因子。

2 算法描述

2.1 基于广度优先搜索的服务功能链构建算法

算法1利用广度优先搜索的思想,在每一次的循环中,以每一个当前构建方案扩展为基础,扩展多个新的服务功能分支,最终完成所有构建方案的搜索。以 G_{SFC} 表示构建方案集合, $g_{\text{SFC},k}$ 为其中第 k 个构建方案,每个构建方案以向量进行表示,向量中的每一位数字代表服务功能类型。算法1步骤如下:①当 G_{SFC} 为空集时,则将集合 F_{req} 中的服务功能赋值给集合 G_{SFC} ,作为初始构建方案,然后分别以每个服务功能开始,进行服务链的构建和扩展;②当 G_{SFC} 为非空集时,对于其中每一个构建方案 $g_{\text{SFC},k}$,计算 F_{req} 中除去 $g_{\text{SFC},k}$ 中已有的服务功能后剩余的服务功能集合 F_{rem} ,并根据依赖矩阵 D 判断 F_{rem} 中每一个服务功能是否均能放置在 $g_{\text{SFC},k}$ 最后一个服务功能 f_{last} 后;③若 F_{rem} 中所有服务功能均能放置在 f_{last} 之后,则分别将 F_{rem} 中的服务功能 $f_{\text{rem},n}$ 与 $g_{\text{SFC},k}$ 一起构建该次循环产生的服务链构建方案,即以 $g_{\text{SFC},k}$ 为基础扩展多个构建分支。进行依赖条件判断的原因,在由 $g_{\text{SFC},k}$ 扩展的每一个完整服务链构建方案中,其对应的 F_{rem} 中的服务功能均会出现一次,若 F_{rem} 中一个服务功能不满足与 f_{last} 之间的依赖条件,则由 $g_{\text{SFC},k}$ 生成的所有完整构建方案都将不满足依赖条件,因此提前进行依赖条件判断,以降低算法搜索空间。

算法1 基于广度优先搜索的服务功能链构建算法
输入 F_{req}, D
输出 服务功能链构建方案集合 G_{SFC}

1 初始化:服务链构建过程集合 $G_P = \emptyset, j_c = 1,$

```

1   $G_{\text{SFC}} = \emptyset$ ;
2  while  $j_c \leq |F_{\text{req}}|$ 
3    if  $G_{\text{SFC}} = \emptyset$ 
4       $G_{\text{SFC}} = F_{\text{req}}$ 
5       $j_c = j_c + 1$ 
6    else
7      for  $k = 1 : |G_{\text{SFC}}|$ 
8        计算  $g_{\text{SFC},k}$  对应的剩余服务功能集合  $F_{\text{rem}}$ 
           $= \mathbb{I}_{F_{\text{req}}} g_{\text{SFC},k}$ , 获取  $g_{\text{SFC},k}$  中最后一个服务
          功能  $f_{\text{last}}, m_c = 0$ ;
9        for  $l_c = 1 : |F_{\text{rem}}|$ 
10         if  $\mathbf{D}(f_{\text{last}}, f_{\text{rem},l_c}) = 1$  or  $\mathbf{D}(f_{\text{last}},$ 
               $f_{\text{rem},l_c}) = 0 \& \& \mathbf{D}(f_{\text{rem},l_c}, f_{\text{last}}) =$ 
               $= 0$ 
11            $m_c = m_c + 1$ ;
12         end if
13       end for
14       if  $m_c = |F_{\text{rem}}|$ 
15         for  $n_c = 1 : |F_{\text{rem}}|$ 
16            $[g_{\text{SFC},k}, f_{\text{rem},l}]$  进入集合  $G_p$ 
17         end for
18       end if
19     end for
20      $j_c = j_c + 1$ 
21      $G_{\text{SFC}} = G_p; G_p = \emptyset$ ;
22   end if
23 end while

```

2.2 基于改进遗传粒子群算法的服务功能链映射算法

服务功能链的映射问题已经被证明是 NP 难问题^[19], 本文需要求解最优的服务链构建方案及其映射方案, 问题复杂度更高。粒子群算法作为一种基于群智能理论的优化算法, 具有运算速度快、全局搜索能力好等特点, 非常适合在动态、多目标场景下寻优。但是, 传统的粒子群算法一般应用于连续问题, 难以直接应用于离散问题。为此本文提出基于改进遗传粒子群算法 (PSO-GA) 的服务功能链映射算法, 即采用遗传算法 (GA) 改进粒子群算法, 用遗传算法的交叉和变异操作代替传统粒子群算法的位置更新和速度更新; 同时引入双层编码方法, 实现服务链构建方案与部署方案同时求解。

2.2.1 个体编码与解码 采用十进制整数编码的方式对服务功能链的构建和映射方案进行编码, 长度为 $2|F_{\text{req}}|$ 。图 3 为个体编码示意图, 以图 1 中的

服务功能链 SFC3 为例, 前 4 位为服务链构建编码, 不同的序号代表不同的服务功能类型, 由左至右的顺序代表服务功能之间的逻辑先后顺序, 后 4 位为服务链映射编码, 分别对应前 4 位不同服务功能所映射的节点位置, 如 VNF1 映射在底层网络 6 号节点。在解码过程中, 首先根据服务链构建编码计算每个服务功能所需要的计算资源数量以及服务功能之间所需链路带宽, 然后根据映射编码在路径集合 P_{shortest} 中查找服务功能映射节点之间流量转发路径, 判断路径上链路是否满足带宽需求, 若不满足, 则利用最短路径算法重新计算转发路径。这种服务功能路径的计算方式能够减少节点之间路径的计算次数, 提高算法效率。

1 3 2 4 | 6 1 5 3

图 3 个体编码示意图

2.2.2 适应度值计算 将粒子 i 对应的部署开销作为其适应度值, 表达式如下

$$\text{fit}(i) = \text{cost}(i) \quad (12)$$

式(12)与式(11)计算相同。

2.2.3 粒子交叉 粒子交叉分为个体最优交叉和群体最优交叉两部分, 个体最优交叉是指个体和个体最优粒子进行交叉得到新粒子, 群体最优交叉是指个体与群体最优粒子进行交叉得到新粒子。由于服务功能数量及其相互之间的顺序约束, 编码中构建编码组合数量远小于映射编码组合数量, 因此为了减少非可行粒子的出现并降低算法运算的复杂度, 对两者采用不同的交叉策略。定义 p_{cs} 为服务链编码交叉选择概率, 即当某一个体与最优个体交叉时, 以概率 p_{cs} 进行构建编码部分交叉, 以 $1 - p_{\text{cs}}$ 进行映射编码部分交叉。在服务链构建编码部分交叉时, 交换个体前 $|F_{\text{req}}|$ 位; 在进行映射部分交叉时, 在后 $|F_{\text{req}}|$ 位编码中随机选择交叉位置进行单点交叉。

2.2.4 个体变异 与个体交叉类似, 定义 p_{ms} 为服务链编码变异选择概率。当某一个体进行变异时, 若进行服务链构建编码部分变异, 则在构建方案集合 G_{SFC} 中随机选择一个服务功能链构建方案进行替换; 若映射编码部分进行变异, 则在前 $|F_{\text{req}}|$ 位编码中随机选择一位, 在该位服务功能 f 对应的承载节点集合 V_f 中随机选择节点编号进行替换。

算法 2 步骤如下: ①进行初始化, 根据 G_{SFC} 和 V_f 生成初始粒子群 G_{ini} 和个体最优集合 G_{rib} , 并计算对应适应度值集合 $\Gamma_{\text{ini}}, \Gamma_{\text{rib}}$, 从而得到群体最优粒子 R_G 和其适应度值 Γ_G ; ②进行个体最优交叉操作,

根据随机生成的交叉概率 p_r 进行对应的交叉操作,并以类似的方法进行群体最优交叉和个体变异操作;③对新的粒子群中的个体进行可行性检查,即对映射节点的剩余带宽容量进行检查,若不满足服务链映射要求,则从集合 V_f 随机选择其他映射节点进行替换;④按照公式(11)计算粒子适应度值,根据适应度值进行个体最优粒子与群体最优粒子更新,在满足迭代次数时终止算法循环。算法 2 中用到的符号和含义如表 1 所示。

表 1 算法 2 中的符号和物理意义

符号	含义
N_{ini}	粒子群个体数
M_l	算法迭代次数
p_{cs}	编码交叉选择概率
p_{ms}	编码变异选择概率
G_{ini}	粒子群集合
$g_{ini,i}$	G_{ini} 中第 i 个个体
G_{rib}	G_{ini} 个体最优集合
$g_{rib,i}$	G_{rib} 中第 i 个个体
Γ_{ini}	G_{ini} 中个体的适应度值集合
$\varphi_{ini,i}$	$g_{ini,i}$ 对应的适应度值
Γ_{rib}	G_{rib} 中个体对应的适应度值集合
$\phi_{rib,i}$	$g_{rib,i}$ 对应的适应度值
R_{PG}	当前群体最优粒子
Γ_{PG}	R_{PG} 对应适应度值
R_G	群体最优粒子
Γ_G	适应度值

算法 2 基于改进遗传粒子群算法的服务功能链部署算法

输入 底层网络参数: $G=(V,E),C_v,B_{v,w},V_f,\lambda_f,\mu_f$, 服务请求参数: $v_s,v_d,\xi_{req},G_{SFC},\alpha,\beta$, 算法参数: 群体个体数 N_{ini} , 迭代次数 M_l,p_{cs},p_{ms} ;
输出 群体最优粒子 R_G 及其开销 Γ_G ;

```
1 /* 初始化 */
2 for  $i=1:N_{ini}$ 
3     随机选择  $g_{SFC} \in G_{SFC}$ ;
4     对  $g_{SFC}$  中每一个 VNF, 从集合  $V_f$  中随机选择映射节点, 构成映射部分编码  $C_{emb}$ ;
5      $g_{ini,i}=[g_{SFC}, C_{emb}]$ ;  $g_{ini,i}$  进入粒子群集合  $G_{ini}$ 
6 end for
7 按照步骤 2~步骤 5 生成个体最优粒子  $g_{rib,i}$  以及个体最优集合  $G_{rib}$ ;
8 按照公式(11)计算  $g_{rib,i}$  与对应的适应度值  $\varphi_{ini,i}$ 、
```

```
 $\varphi_{rib,i}$  并生成  $G_{ini}$  与  $G_{rib}$  对应的适应度集合  $\Gamma_{ini}, \Gamma_{rib}$ ;
9 计算当前群体最优  $R_{PG}$  和适应度值  $\Gamma_{PG}$ ;
10 群体最优  $R_G=R_{PG}$ , 对应的适应度值  $\Gamma_G=\Gamma_{PG}, n_c=0$ ;
11 while  $n_c \leq M_l$ 
12     /* 个体最优交叉 */
13     for  $j=1:N_{ini}$ 
14         随机生成交叉概率  $p_r$ ;
15         if  $p_r < p_{cs}$ 
16              $g_{ini,i}$  与  $g_{rib,i}$  进行服务链构建编码交叉;
17         else
18              $g_{ini,i}$  与  $g_{rib,i}$  进行服务链映射编码交叉;
19         end if
20     end for
21     /* 群体最优交叉 */
22     对每个  $g_{ini,i}$  与  $R_G$  进行交叉操作;
23     /* 粒子变异 */
24     对每个  $g_{ini,i}$  进行变异操作;
25     对每个  $g_{ini,i}$  进行可行性检查;
26     /* 个体最优粒子与群体最优粒子更新 */
27     计算个体适应度集合  $\Gamma_{ini}$ ;
28     for  $k=1:N_{ini}$ 
29         if  $\varphi_{ini,k} < \varphi_{rib,k}$ 
30              $\varphi_{rib,k}=\varphi_{ini,k}, g_{ini,k}=g_{rib,k}$ ;
31         end if
32     end for
33     计算当前群体最优粒子  $R_{PG}$  和适应度值  $\Gamma_{PG}$ ;
34     if  $\Gamma_{PG} < \Gamma_G$ 
35          $\Gamma_G=\Gamma_{PG}, R_G=R_{PG}$ ;
36     end if
37      $n_c=n_c+1$ ;
38 end while
```

3 仿真实验

3.1 实验设置

为了验证算法性能,使用 MATLAB 2018b 软件在配置为 Intel Core i7-6700 CPU、16 GB RAM 台式计算机上进行仿真。借鉴文献[15,19]的研究,仿真拓扑采用 SNDlib 上 pdh 拓扑^[22],包含 11 个点和 34 条边,每个节点初始处理资源为 256 个处理单元,实验中该拓扑每条无向边设置为两条有向边,每个方向链路带宽为 700 Mb/s。实验中 VNF 类型如表 2 所示,每个节点从中随机选取 3 种作为承载类

型。借鉴文献[9]中表 1,本文在表 2 中规定了不同服务功能流量改变率和每 1 Mb/s 带宽所需处理单元数量。实验中不同服务请求类型如图 4 所示,每个服务请求的源节点和目的节点随机选择。权重因子 α 和 β 分别为 1 和 0.05。

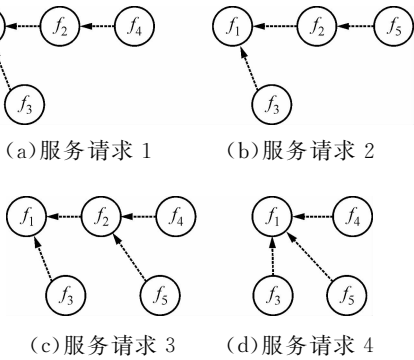


图 4 4 种不同的服务请求类型

表 2 服务功能类型及其参数

服务功能	f_1	f_2	f_3	f_4	f_5
λ_f	0.8	1.2	0.7	0.9	1.1
$u_f/(\text{Mb} \cdot \text{s}^{-1})$	0.02	0.04	0.03	0.05	0.03

3.2 实验分析

首先,以图 4a 中的服务请求为例,验证本文提出的改进遗传粒子群算法的收敛性和有效性。设置服务请求带宽为 30 Mb/s,粒子群个体数 $N_{\text{ini}}=30$,迭代次数 $N_i=200$,并且实验 200 次得到的数据取平均值,结果如图 5 所示。

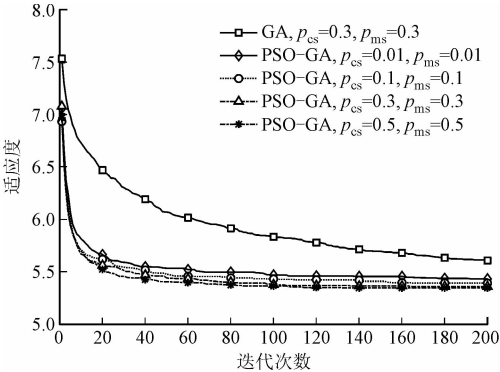


图 5 p_{cs} 和 p_{ms} 取值对算法性能影响

图 5 为不同的 p_{cs} 和 p_{ms} 取值对算法性能的影响。从图中可以看出,本文所提出的 PSO-GA 算法能够实现有效收敛,增大服务链构建编码部分的交叉和变异概率能够改善算法性能。主要原因为,当 p_{cs} 和 p_{ms} 增加时,服务链构建编码部分的交叉和变异更够扩大粒子群算法搜索空间,相比于映射部分的交叉和变异,能够为该粒子引入更多的有效信息,

从而更快地寻找最优解。在 p_{cs} 和 p_{ms} 均为 0.3 时可以看出,改进遗传粒子群算法比传统的遗传算法收敛速度更快,这主要是因为 PSO-GA 算法是一种更为高效的并行搜索算法,在算法运行过程中,每个粒子均保留其个体极值,通过个体最优交叉,每个粒子都能动态跟踪当前搜索状况,动态调整搜索策略,同时通过群体最优交叉,群体最优粒子能够引导整个粒子群趋近最优解;而遗传算法通过交叉变异和选择复制进行种群迭代,每个染色体无法保留其迭代过程中的个体最优信息,其搜索能力有限。

然后,将本文提出的 RACCM 算法与文献[19]中的 TASAR 算法、随机选取服务链构建策略的 RANDOM 算法进行对比,验证在多个服务请求到达的情况下算法求解出的服务功能链部署方案的资源利用有效性。实验中每个服务请求的请求类型从图 4 中随机进行选择,所有服务请求带宽服从[20, 30] Mb/s 之间均匀分布,每种对比算法运行 5 次,取平均值,实验结果如图 6~图 9 所示。图 6 为不同服务请求数量下的服务接受率对比,可以看出,当服务请求数量超过 700 时,TASAR 算法开始出现接受率下降的情况,在服务请求数量超过 1 000 时,曲线呈直线下降趋势,没有新的请求可以部署成功。RACCM 算法的服务请求接受率比 TASAR 算法提高了 9%左右,比 RANDOM 算法提高了 10%左右,主要原因为 RACCM 算法部署服务请求所消耗的资源最低,在相同的资源条件下能够容纳更多的服务功能链部署,如图 7 所示。

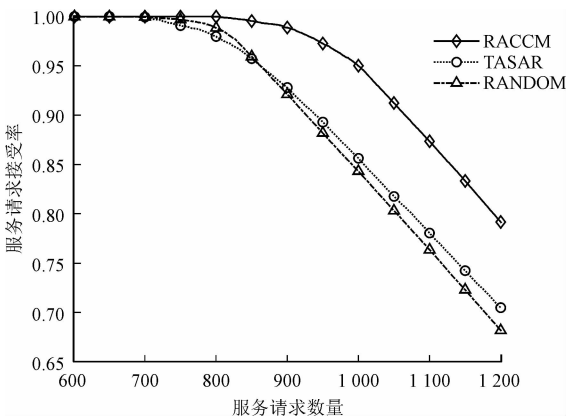


图 6 不同服务请求数量下的服务请求接受率

图 7 为不同服务请求数量下成功部署的服务请求的归一化平均部署开销,在资源充足情况下 RACCM 算法对应的平均部署开销最低,为 4.6 左右,TASAR 算法居中,为 4.75 左右,RANDOM 算法最高,为 5.5 左右,主要原因为:RACCM 能够根

据底层节点所能承载的 VNF 类型以及资源使用情况选择最优的服务功能链构建和映射方案,而 TASAR 在进行服务功能链构建在服务功能依赖约束下将流量改变率最小的服务功能放置在服务链最靠前的位置,以降低服务功能之间的带宽需求,但是该种方式并未考虑底层节点所能承载的服务功能类型约束,可能导致服务功能之间路由路径的延长,增加带宽资源开销;RANDOM 算法既没有考虑服务功能类型约束,也没考虑服务功能对流量的改变,其平均部署开销最高。从图 7 中还可以看出,后期平均部署开销增大,原因为部分节点计算资源耗尽,服务链的路由跳数增加,带来了带宽开销增加,进而平均部署开销增加。

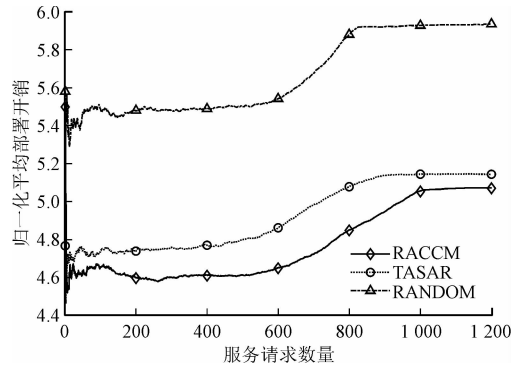


图 7 不同服务请求数量下的归一化平均部署开销

图 8 和图 9 分别为不同服务请求数量下计算资源和带宽资源利用率比较。可以看出,在服务请求数量小于 900 时,RACCM 算法节点资源比 TASAR 增加了约 10% 左右,带宽资源降低了约 13%。这是由于 TASAR 算法在确定服务链构建方案时仅关注不同服务功能对流量速率的改变,寻求构建时带宽需求最优的方案,虽然其在计算资源开销上是最低的,但是由于忽视了网络中 VNF 实例

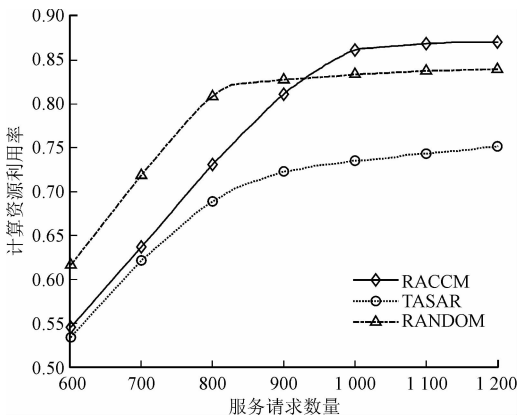


图 8 3 种算法在不同服务请求数量下的计算资源利用率比较

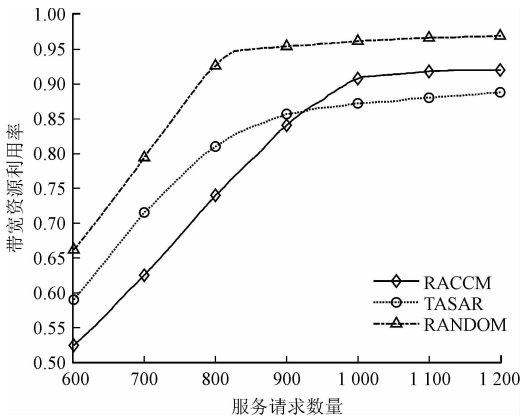


图 9 3 种算法在不同服务请求数量下的带宽资源利用率比较

分布以及节点资源使用情况,其带宽资源将会被更快耗尽,这也印证了 TASAR 算法在图 6 中的请求接受率比 RACCM 算法要低,在图 7 中其平均部署开销比 RACCM 算法高。当服务请求数量超过 1 000 时,RACCM 算法的计算资源和带宽资源利用率均高于 TASAR 算法,主要是因为 RACCM 算法比 TASAR 算法成功部署的服务请求数量多。

4 结束语

本文主要研究了虚拟化环境下服务功能链的构建和映射问题,提出了一种资源感知的服务功能链协同构建和映射算法,实现服务功能链资源优化部署。该方法基于单个服务请求,具有实时、快速、高效特点,能够满足动态场景下服务链的灵活部署。未来将结合人工智能技术,对虚拟资源动态扩展方法进行研究,以满足智慧化网络运营需求。

参考文献:

[1] ZHANG Shunliang. An overview of network slicing for 5G [J]. IEEE Wireless Communications, 2019, 26 (3): 111-117.

[2] YE Qiang, LI Junling, QU Kaige, et al. End-to-End quality of service in 5G networks: examining the effectiveness of a network slicing framework [J]. IEEE Vehicular Technology Magazine, 2018, 13(2): 65-74.

[3] GUPTA A, HABIB M F, MANDAL U, et al. On service-chaining strategies using virtual network functions in operator networks [J]. Computer Networks, 2018, 133: 1-16.

[4] HAWILO H, JAMMAL M, SHAMI A. Network function virtualization-aware orchestrator for service function chaining placement in the cloud [J]. IEEE Journal on Selected Areas in Communications, 2019,

- 37(3): 643-655.
- [5] SARAIVA D S N F, LACHOS P D A, ROSA R V, et al. Network service orchestration: a survey [J]. *Computer Communications*, 2019, 142/143(6): 69-94.
 - [6] GREENE N, PARKER R, PERRY R. Is your network ready for digital transformation? IDC white paper [R/OL]. (2017-01-18) [2019-08-05]. <https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/digital-network-architecture/network-ready-digital-transformation.pdf>.
 - [7] HALPERN J, PIGNATARO C. Service function chaining (SFC) architecture: internet engineering task force [R/OL]. (2015-10-13) [2019-08-05]. <https://tools.ietf.org/html/rfc7665>.
 - [8] The 5G Architecture Working Group. View on 5G architecture: version 3.0 [R/OL]. (2019-06-19) [2019-08-05]. <https://5g-ppp.eu/5g-ppp-architecture-public-consultation/>.
 - [9] LI Junling, SHI Weisen, YE Qiang, et al. Online joint VNF chain composition and embedding for 5G networks [C] // 2018 IEEE Global Communications Conference. Piscataway, NJ, USA: IEEE, 2018: 8647700.
 - [10] CISIO. NAT order of operation [EB/OL]. (2008-09-30) [2019-08-05]. <https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/6209-5.html>.
 - [11] Microsoft Technet. VPNs and firewalls [EB/OL]. (2008-11-09) [2019-08-05]. [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc958037\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc958037(v=technet.10)).
 - [12] MILLER M, BRANKA V, LES B. Satellite communications: mobile and fixed services [M]. Berlin, Germany: Springer Science & Business Media, 1993: 247-249.
 - [13] BU Chao, WANG Xingwei, CHENG Hui, et al. Enabling adaptive routing service customization via the integration of SDN and NFV [J]. *Journal of Network and Computer Applications*, 2017, 93: 123-136.
 - [14] OCAMPO A F, GIL-HERRERA J, ISOLANI P H, et al. Lecture Notes in Computer Science, Vol 10356: Optimal service function chain composition in network functions virtualization [M]. Berlin, Germany: Springer Verlag, 2017: 62-76.
 - [15] 袁泉, 汤红波, 黄开枝, 等. 基于 Q-learning 算法的 vEPC 虚拟网络功能部署方法 [J]. *通信学报*, 2017, 38(8): 172-182.
YUAN Quan, TANG Hongbo, HUANG Kaizhi, et al. Deployment method for vEPC virtualized network function via Q-learning [J]. *Journal on Communications*, 2017, 38(8): 172-182.
 - [16] 魏亮, 黄韬, 张娇, 等. 基于强化学习的服务链映射算法 [J]. *通信学报*, 2018, 39(1): 90-100.
WEI Liang, HUANG Tao, ZHANG Jiao, et al. Service chain mapping algorithm based on reinforcement learning [J]. *Journal on Communications*, 2018, 39(1): 90-100.
 - [17] JALALITABAR M, GULER E, ZHENG Danyang, et al. Embedding dependence-aware service function chains [J]. *Journal of Optical Communications and Networking*, 2018, 10(8): C64-C74.
 - [18] BECK M T, BOTERO J F. Coordinated allocation of service function chains [C] // 58th IEEE Global Communications Conference. Piscataway, NJ, USA: IEEE, 2015: 7417401.
 - [19] 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 Hongbo, QIU Hang, YOU Wei, et al. A reliability-guarantee method for service function Chain deployment based on joint backup [J]. *Journal of Electronics and Information Technology*, 2019, 41(12): 3006-3013.
 - [20] European Telecommunications Standards Institute. Network functions virtualisation (NFV), management and orchestration; NFV-MAN 001 V1.1.1 [S/OL]. (2014-12) [2019-08-05]. https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf
 - [21] YU Hui, YANG Jiahai, FUNG C. Elastic network service chain with fine-grained vertical scaling [C] // 2018 IEEE Global Communications Conference. Piscataway, NJ, USA: IEEE, 2018: 8648096.
 - [22] ORLOWSKI S, WESSÄLY R, PIÓRO M, et al. SNDlib 1.0: survivable network design library [J]. *Networks*, 2010, 55(3): 276-286.

(编辑 刘杨)