

基于最优加权图匹配的服务功能链部署方法

李丹, 兰巨龙, 王鹏, 胡宇翔

(国家数字交换系统工程技术研究中心, 河南 郑州 450000)

摘 要: 服务功能链技术通过对虚拟网络功能的编排来支持灵活的网络服务请求。针对资源有限网络中的服务功能链部署问题, 提出了一种基于最优加权图匹配的服务功能链部署方法, 把服务功能链组合为功能拓扑图, 利用邻接矩阵特征向量分解算法获取功能拓扑与物理拓扑的加权图匹配方式, 并通过爬山算法对匹配结果进一步优化。仿真结果表明, 所提方法在降低服务功能链部署所需带宽的同时, 优化了节点负载和链路带宽的均衡度, 可以支持更多的服务请求, 且复杂度低, 具有较高的时效性。

关键词: 服务功能链; 加权图匹配; 特征向量分解; 爬山算法

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2019059

Service function chain deployment algorithm based on optimal weighted graph matching

LI Dan, LAN Julong, WANG Peng, HU Yuxiang

National Digital Switching System Engineering & Technology Research Center, Zhengzhou 450000, China

Abstract: Service function chain can support flexible network service requirement by linking virtual network functions. Aiming at the problem of service function chain deployment in a resource-constrained network, an algorithm for service function chain deployment based on optimal weighted graph matching was proposed. The service function chains was composed into graphs of functional topography, and the optimal matching results between graphs of functional topology and physical topology was obtained using eigendecomposition approach, and furtherly the matching results by hill-climbing method was optimized. Simulation results show that, the proposed algorithm can reduce the required bandwidth to deploy service function chains, balance the load of nodes and bandwidth of links, and support more service requests. What is more, the algorithm has a lower computation complexity and higher time efficiency.

Key words: service function chain, weighted graph matching, eigendecomposition approach, hill-climbing method

1 引言

互联网的高速发展和广泛应用, 对网络中的新功能和新服务提出了更高的要求。网络功能虚拟化技术(NFV, network function virtualization)^[1]通过分离逻辑功能与物理资源, 将多样化的网络功能和服务从底层物理资源中抽象出来, 改变了传统网络

功能的设计和部署方式, 可以更加灵活、高效地利用网络资源。NFV 技术将传统网络中部署在特定硬件设备上的功能虚拟化为虚拟网络功能, 可以根据需要灵活部署在任意一个通用功能平台。在 NFV 的基础上, 服务功能链技术(SFC, service function chaining)^[2]将不同的网络功能按照一定的顺序连接起来, 支持某种具体的服务请求, 实现网络服务的

收稿日期: 2018-05-31; 修回日期: 2019-02-21

基金项目: 国家高技术研究发展计划(“863 计划”)基金资助项目(No.2015AA016102); 国家自然科学基金资助项目(No.61521003, No.61702547)

Foundation Items: The National High Technology Research and Development Program of China (863 Program) (No.2015AA016102), The National Natural Science Foundation of China (No.61521003, No.61702547)

个性化定制。要将虚拟的服务功能链与实际的网络拓扑结合起来, 需要将这些功能部署在网络中的节点上, 并选择路径使数据依次经过提供这些功能的节点。软件定义网络 (SDN, software defined networking) [3] 将控制平面和数据平面分离, 在控制平面集中计算最优的功能部署位置和数据传输路径, 使服务功能链的动态配置和部署成为可能。

为了在网络中提供多样化的服务, 需要部署大量功能顺序不同的服务功能链, 但是由于网络中的节点处理能力和链路带宽有限, 如何在资源有限的条件下尽可能多地部署服务功能链是本文研究的主要问题。目前, 针对服务功能链的部署问题已经有了一些相关研究, 文献[4]建立了基于 SDN 架构的服务功能链部署模型, 通过统一控制机制来确保功能部署和数据传输的准确性, 并举例说明了该架构可以支持的优化目标, 但是没有给出具体的部署算法。文献[5-6]以优化数据传输和处理时延为目标, 给出了服务功能链部署的问题模型以及优化求解算法。文献[7-8]从时延、分组丢失率、中继节点数等方面建立了多参数混合优化模型, 来提高服务功能链的综合服务性能。文献[9-10]通过遗传算法来适应服务功能链对服务质量的动态需求, 在网络环境变化时仍能提供较好的服务质量。

以上算法主要针对服务性能进行优化, 而没有考虑网络资源的限制。文献[11]以整个网络的带宽资源需求为优化目标, 研究了虚拟网络功能的优化部署问题。文献[12]针对同时优化带宽需求和数据传输处理时延的问题, 设计了一种启发式搜索算法。文献[13-14]研究了在数据中心网络中部署服务功能链的网络流量均衡算法。文献[15]研究了一种功能合并部署算法, 该算法统计各个功能之间的带宽需求, 将带宽需求较大的功能合并部署在同一节点上来降低路径带宽。但是这些文献侧重于对带宽资源的优化利用, 而缺少对节点处理能力和链路带宽的综合考虑。针对节点处理能力和带宽资源有限的情况, 文献[16-17]通过启发式算法来实现降低链路带宽需求和减少功能部署开销之间的平衡。文献[18]将网络中所有节点的剩余处理能力、存储空间和带宽加权组合为节点剩余资源, 把服务功能链部署在满足约束条件且剩余资源总和最大的路径上, 但是该算法需要遍历所有部署方式, 计算复杂度较高。文献[19]将服务功能链分解为多层有向图的组合, 通过维特比算法求解每一层的节点选择, 但该算法

在选择节点的过程中会优先将功能部署在相同节点中, 容易导致部分节点负载过重而无法支持新的服务请求。文献[20]将服务功能链的部署问题转化为功能拓扑和物理拓扑的加权图匹配问题, 将节点处理能力和链路带宽作为参数来选择符合要求的节点匹配方式, 该算法可以快速生成符合传输需求的部署方式, 但是会将带宽需求较大的虚拟链路映射为跳数较多的物理传输路径, 造成大量带宽损耗, 且带宽和负载均衡效果较差。文献[21]提出了一种最优加权图匹配算法, 该算法利用特征向量分解算法实现 2 幅同构加权图的快速匹配, 可以较大概率实现最优匹配, 即使得到的不是最优解, 所得结果仍然可以通过爬山算法修正为最优解, 但是该方法对待匹配的加权图有一定的要求, 无法直接解决服务功能链部署问题。

基于以上分析, 本文提出了一种基于最优加权图匹配的服务功能链部署方法, 把相邻的服务功能链组合为功能拓扑图, 将功能链的部署问题转化为功能拓扑与物理拓扑的加权图匹配问题, 通过对加权图邻接矩阵的扩充和元素替换来满足文献[21]的同构条件, 采用邻接矩阵特征向量分解算法和爬山优化算法得到功能和链路的最优匹配。与其他算法相比, 本文方法的创新点体现在以下 3 个方面: 1) 可以实现多条服务功能链的同时部署, 降低计算复杂度, 具有较高的时效性; 2) 通过合理的元素替换, 均衡链路匹配过程中路径带宽和传输跳数的关系, 避免带宽浪费, 节省网络资源; 3) 计算节点和链路资源的最优匹配, 能够同时均衡全网的节点负载和链路带宽资源, 可以支持更多的服务请求。

2 服务功能链部署与加权图匹配问题

2.1 服务功能链部署问题

网络的物理拓扑可以建模为一个无向加权图 $G=(V, E)$, C_e 表示链路 e ($e \in E$) 的带宽, C_v 表示节点 v ($v \in V$) 的处理能力。网络中需要部署一组服务请求 $R=\{S_1, S_2, \dots\}$, 其中 $S_i=\{f_{i1}, f_{i2}, \dots\}$, 表示第 i 条服务功能链, 该服务功能链所需带宽用 b_i 表示, f_{ij} ($j=1, 2, \dots$) 表示 S_i 中的第 j 项功能, 该功能所需的处理能力用 d_{ij} 表示。对于每一条服务功能链, 都要为其中的功能找到部署节点, 并选择一条路径依次连接这些节点, 在部署功能和选择路径的过程中应该减少并均衡网络资源的消耗, 尽可能将

处理能力需求较高的功能和带宽需求较高的链路匹配到剩余资源较多的节点和路径上, 以支持更多的服务请求。

本文将服务功能链的部署问题建模为一个多约束优化问题, 如式(1)~式(3)所示。

$$\max |R| \tag{1}$$

$$\text{s.t. } \sum_{i,j:p_{ij}=v_k} d_{ij} \leq C_v, \forall v_k \in V \tag{2}$$

$$\sum_{i:e_k \in l_i} b_i \leq C_e, \forall e_k \in E \tag{3}$$

其中, $|R|$ 表示网络能支持的服务请求数, $p_{ij} = v_k$ 表示功能 f_{ij} 部署在节点 v_k 上, $e_k \in l_i$ 表示链路 e_k 包含在第 i 条服务功能链的传输路径之中。式(2)和式(3)分别表示每个节点的处理能力限制和每条链路的带宽限制。该多约束优化问题的求解是一个 NP-hard 问题, 为此, 本文提出一种最优加权图匹配算法, 在减少和均衡网络资源消耗的同时, 快速生成符合传输需求的部署方式。

2.2 加权图匹配问题

定义 1 最优加权图匹配^[21] (OWGM, optimal weighted graph matching)。对于 2 个节点数为 n 的加权图 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$, 要找到节点集合 V_1 和 V_2 之间的映射函数 \mathbf{P} , 使 G_1 在该函数下新生成的加权图与 G_2 的差最小。加权图的差是指 2 个加权图的邻接矩阵之差的 F-范数。

根据定义 1, 最优加权图匹配问题可表示为式(4)。

$$\min J(\mathbf{P}) = \|\mathbf{P}\mathbf{A}_1\mathbf{P}^T - \mathbf{A}_2\| \tag{4}$$

其中, \mathbf{P} 是节点映射函数, 即 \mathbf{P} 中的每一行和每一列都只有一个元素为 1, 其他元素均为 0; $J(\mathbf{P})$ 表示 G_1 经过函数 \mathbf{P} 作用后和 G_2 的差; \mathbf{A}_1 和 \mathbf{A}_2 分别表示 2 个图的邻接矩阵。

文献[21]给出了一种可以快速计算近似最优匹配的特征向量分解算法。如果 2 个加权图是近似同构的(同构是指一个图的边经过有限次交换后与另一张图相同), 则可以将式(4)进行等价转化, 把加权图匹配问题转化为以 $\bar{\mathbf{U}}_2\bar{\mathbf{U}}_1^T$ 为效率矩阵的指派问题, 其中 $\bar{\mathbf{U}}_1$ 和 $\bar{\mathbf{U}}_2$ 分别是 \mathbf{A}_1 和 \mathbf{A}_2 的特征向量的绝对值矩阵, 最后通过匈牙利算法^[22]对 $\bar{\mathbf{U}}_2\bar{\mathbf{U}}_1^T$ 进行处理可以得到最优映射函数, 表示为

$$\max \text{tr}(\mathbf{P}^T\bar{\mathbf{U}}_2\bar{\mathbf{U}}_1^T) \tag{5}$$

其中, $H(x)$ 表示匈牙利算法。

$$\mathbf{P} = H(\bar{\mathbf{U}}_2\bar{\mathbf{U}}_1^T) \tag{6}$$

利用特征向量分解算法进行加权图匹配需要一定的前提条件, 为了利用文献[21]的算法对服务功能链和物理拓扑进行匹配, 需要在算法中解决以下 3 个问题。

1) 在加权图匹配问题中, 2 幅图的顶点和边的数目相同, 但是通常情况下服务功能链中的功能数少于物理拓扑中的节点数。

2) 在加权图匹配问题中, 2 幅图的链路是一一对应进行匹配的, 而在服务功能链的部署过程中, 2 个相邻功能间的虚拟链路既可以映射为一条物理链路, 也可以映射为多条物理链路。

3) 特征向量分解算法要求待匹配的 2 个图近似同构, 但是服务功能链与物理拓扑顶点数不同, 链路权值不相关, 不满足同构条件。

3 方法设计

3.1 服务功能链组合

本文方法利用最优加权图匹配算法计算功能和链路的映射关系, 支持多条服务功能链的同时部署, 首先需要将服务功能链组合为功能拓扑图。以图 1 所示的 2 条服务功能链为例, 将 2 条功能链中的相同功能合并, 图中的功能权值表示该功能需要的节点处理能力, 链路权值表示其连接的功能间传输路径所需带宽, 需要注意的是, 2 条功能链重叠部分的权值应为二者之和。

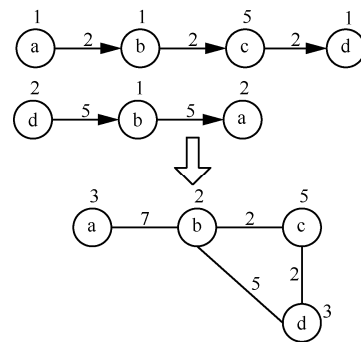


图 1 服务功能链组合为功能拓扑

3.2 邻接矩阵扩充

令 \mathbf{A}_1 和 \mathbf{A}_2 分别表示功能拓扑和物理拓扑对应的邻接矩阵, 其中, \mathbf{A}_1 中对角线元素表示各功能所需的节点处理能力, 其他元素表示对应功能间所需的传输带宽; \mathbf{A}_2 中对角线元素表示各节点的剩余处

理能力，其他元素表示对应节点间的链路剩余带宽。为了符合图匹配问题中顶点数相同的要求，需要在 A_1 中增加参数为 0 的元素，将其扩充到与 A_2 相同的维度。图 2 给出了一组待匹配的功能拓扑图与物理拓扑图。由于功能拓扑图的顶点数为 4 而物理拓扑图的顶点数为 6，因此需要在功能拓扑的邻接矩阵 A_1 中增加 2 行 0 元素和 2 列 0 元素，使其与物理邻接矩阵维度相同，如式(7)所示。

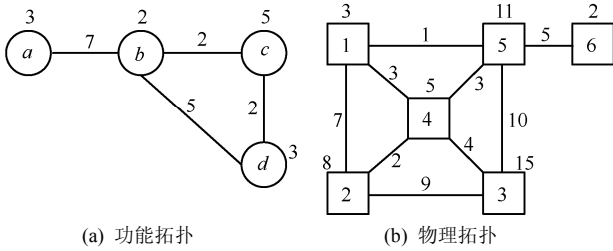


图 2 待匹配的功能拓扑与物理拓扑

$$A_1 = \begin{bmatrix} 3 & 7 & 0 & 0 \\ 7 & 2 & 2 & 5 \\ 0 & 2 & 5 & 2 \\ 0 & 5 & 2 & 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 7 & 0 & 0 & 0 & 0 \\ 7 & 2 & 2 & 5 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 5 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

3.3 矩阵元素替换

在服务功能链的部署过程中，相邻功能既可以匹配到相邻节点，也可以匹配到非相邻节点。为了在链路映射的过程中满足以上要求，需要对物理拓扑邻接矩阵中表示链路带宽的矩阵元素进行替换，为每一对节点选择一条最优路径并更新匹配权值。在文献[20]中，按照最小带宽最大原则为非相邻节点计算路径并以该带宽作为新的链路匹配权值，虽然该方法支持相邻功能与非相邻节点的匹配，但是存在以下 2 个问题：1) 没有更新相邻节点间的路径带宽，导致相邻节点间直连带宽较小时无法进行链路匹配；2) 最小带宽最大路径很可能需要经过多次转发，路径较长，匹配带宽需求较高的虚拟链路后会造成大量的带宽损耗。为此，算法 1 为任意一对节点计算一条路径，使该路径的最小带宽与传输跳数的比值最大，并把该比值更新为链路匹配权值，从而均衡路径带宽和传输跳数的关系，避免带宽浪费。

算法 1 物理拓扑链路权值更新算法

1) for any nodes o and t in G do // o : 源节点，

t : 目的节点

2) $s(1)=o, u=o, \text{hop}(o)=0, \text{label}(i)=0$ if $i \neq o$, $\text{label}(o)=10\ 000$ // s : 已确定节点集合, u : 转接点, hop : 各节点到源节点跳数, label : 各节点权值

3) while t is not in s do

4) for $i=1$ to n and i is not in s do

5) $a = \frac{\min(\text{label}(u) \times \text{hop}(u), A_2(u,i))}{\text{hop}(u)+1}$

6) if $\text{label}(i) < a$ then

7) $\text{label}(i)=a, \text{hop}(i)=\text{hop}(u)+1,$

$f(i)=u$

8) end if

9) end for

10) $u = \text{node with maximum label}$

11) put u into s

12) end

13) $\text{path}(1)=t, v=t$ // path : 节点 o 和 t 之间的传输路径, v : 前序节点

14) while o is not in path do

15) $v=f(v)$

16) put v into path

17) end

18) end for

算法 1 给出了任意节点间链路权值更新的伪代码，权值计算的方法和搜索最短路径的方法类似。第 2) 行将已确定权值集合 s 和转接点 u 初始化为源节点 o ，并对各节点的跳数和权值进行初始化；4) 行~9) 行计算各节点经过转接点达到源节点的最小带宽与跳数比值，若超过原始权值则对该节点的权值、跳数和前序节点进行更新；10) 行~11) 行将更新后权值最大的未确定权值节点设置为转接点并加入到已确定集合中；在目的节点权值确定后，13) 行~17) 行根据各节点的前序节点，通过回溯算法生成由目的节点到源节点的传输路径。图 2 中物理拓扑的邻接矩阵 A_2 元素替换过程如下，其中括号内的值为经过更新的权值。

$$A_2 = \begin{bmatrix} 3 & 7 & 0 & 3 & 1 & 0 \\ 7 & 8 & 9 & 2 & 0 & 0 \\ 0 & 9 & 15 & 4 & 10 & 0 \\ 3 & 2 & 4 & 5 & 3 & 0 \\ 1 & 0 & 10 & 3 & 11 & 5 \\ 0 & 0 & 0 & 0 & 5 & 2 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix}
 3 & 7 & \left(\frac{7}{2}\right) & 3 & \left(\frac{7}{3}\right) & \left(\frac{5}{4}\right) \\
 7 & 8 & 9 & 2 & \left(\frac{9}{2}\right) & \left(\frac{5}{3}\right) \\
 \left(\frac{7}{2}\right) & 9 & 15 & 4 & 10 & \left(\frac{5}{2}\right) \\
 3 & 2 & 4 & 5 & 3 & \frac{3}{2} \\
 \left(\frac{7}{3}\right) & \left(\frac{9}{2}\right) & 10 & 3 & 11 & 5 \\
 \left(\frac{7}{4}\right) & \left(\frac{5}{3}\right) & \left(\frac{5}{2}\right) & \left(\frac{3}{2}\right) & 5 & 2
 \end{bmatrix} \quad (8)$$

特征向量分解算法要求待匹配的 2 个加权图近似同构，因此需要进一步对功能拓扑的邻接矩阵元

素进行替换，使其与更新后的物理拓扑图同构。替换方法如下。首先，将各功能和各节点分别按照处理能力由大到小的顺序排列，把功能拓扑邻接矩阵中表示处理能力需求的元素依序替换为各节点的剩余处理能力，“0”元素则随机替换为较小的节点处理能力，实现 2 个加权图的顶点同构。然后，用同样的办法将功能拓扑邻接矩阵中表示带宽需求的元素进行替换，实现 2 个加权图的边同构，满足文献[21]的算法要求。这种替换方法的目的是，在构建同构加权图的过程中，尽可能将处理能力需求较高的功能和带宽需求较高的链路匹配到剩余资源较多的节点和路径上。图 2 中功能拓扑的邻接矩阵元素替换过程如式(9)所示，加号两边分别是非“0”元素和“0”元素的替换矩阵。

$$A_1 = \begin{bmatrix}
 3 & 7 & 0 & 0 & 0 & 0 \\
 7 & 2 & 2 & 5 & 0 & 0 \\
 0 & 2 & 5 & 2 & 0 & 0 \\
 0 & 5 & 2 & 3 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix} \Rightarrow \begin{bmatrix}
 11 & 10 & 0 & 0 & 0 & 0 \\
 10 & 5 & 7 & 9 & 0 & 0 \\
 0 & 7 & 15 & 5 & 0 & 0 \\
 0 & 9 & 5 & 8 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix} + \begin{bmatrix}
 0 & 0 & \frac{3}{2} & \frac{7}{2} & \frac{7}{3} & \frac{5}{2} \\
 0 & 0 & 0 & 0 & \frac{5}{3} & 2 \\
 \frac{3}{2} & 0 & 0 & 0 & 3 & 4 \\
 \frac{7}{2} & 0 & 0 & 0 & \frac{9}{2} & \frac{5}{4} \\
 \frac{7}{3} & \frac{5}{3} & 3 & \frac{9}{2} & 2 & 3 \\
 \frac{5}{2} & 2 & 4 & \frac{5}{4} & 3 & 3
 \end{bmatrix} \quad (9)$$

3.4 特征值分解与匈牙利算法

得到近似同构的 2 个邻接矩阵后，可以按照文献 [21] 的算法流程计算映射函数 P ，使

$J(P) = \|PA_1P^T - A_2\|$ 最小，令 \bar{U}_1 和 \bar{U}_2 分别表示式(8)和式(9)中 A_1 和 A_2 的特征向量的绝对值矩阵，则效率矩阵 $\bar{U}_2\bar{U}_1^T$ 为

$$\bar{U}_2\bar{U}_1^T = \begin{bmatrix}
 0.8546 & 0.8611 & 0.6602 & 0.5905 & 0.7135 & 0.4339 \\
 0.8971 & 0.8934 & 0.7013 & 0.4860 & 0.5570 & 0.3605 \\
 0.6925 & 0.7455 & 0.6461 & 0.4992 & 0.8442 & 0.5555 \\
 0.7123 & 0.5750 & 0.8980 & 0.8680 & 0.5624 & 0.6497 \\
 0.4704 & 0.4809 & 0.7146 & 0.7383 & 0.5599 & 0.6204 \\
 0.5274 & 0.3798 & 0.5750 & 0.3432 & 0.7324 & 0.9253
 \end{bmatrix} \quad (10)$$

利用匈牙利算法对该效率矩阵进行求解，得到最优映射函数如式(11)所示。

$$P = \begin{bmatrix}
 0 & 1 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix} \quad (11)$$

P 中数值为 1 的元素表示其所在行对应的功能部署在其所在列对应的节点上，即 $a \rightarrow 2$ ， $b \rightarrow 1$ ， $c \rightarrow 5$ ， $d \rightarrow 3$ 。第 5、6 行对应的是将功能拓扑扩充后增加的虚拟功能，不需要进行处理。

3.5 爬山优化算法

由于图匹配的过程中并没有判断匹配后对应元素的大小关系，可能出现节点或链路超载的情况，如按照式(11)对图 2 进行匹配，会导致节点 1

和节点 2 之间链路超载，另外，考虑到特征向量分解算法得到的不一定总是最优解，需要对匹配结果进一步优化修正。文献[21]表明，即使特征向量分解算法得到的不是最优解，该结果仍然可以通过爬山算法修正为最优解。因此在算法 2 中，以 3.4 节得到的映射函数 P 为初始值，利用爬山算法对其进行优化，每次迭代过程中尝试交换一对顶点，在满足式(2)和式(3)的限制条件前提下寻找使得 $J(P)$ 更小的交换方式，直至没有更好的交换方式。

算法 2 映射函数修正爬山算法

- 1) flag=1 //flag: 判断修正过程是否结束标志
- 2) while flag=1 do
- 3) flag=0, $J=\|PA_1P^T-A_2\|$ //J: 映射函数作用后 2 个图差值
- 4) for $i=1$ to n do
- 5) for $j=1$ to n do
- 6) $Q=P$
- 7) exchange column i and column j of Q //Q: 交换一对顶点后的映射函数
- 8) $K=\|QA_1Q^T-A_2\|$ //K: 新的映射函数作用后 2 个图差值
- 9) if there is no negative number in $QA_1Q^T-A_2$ and $K<J$ then
- 10) $J=K, P=Q, \text{flag}=1$
- 11) end if

- 12) end for
- 13) end for
- 14) end

算法 2 给出了映射函数修正爬山算法的伪代码,3)行计算当前映射函数作用后 2 张图差值;6)行~7)行尝试交换映射函数的一对顶点;8)行~11)行将新的映射函数作用后的差值和当前差值比较,如果在符合限制条件的情况下新的差值更小,则修正映射函数。式(11)中的映射函数经爬山算法的优化过程如式(12)所示,更新后的匹配结果为 $a \rightarrow 1, b \rightarrow 2, c \rightarrow 5, d \rightarrow 3$ 。

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

3.6 方法流程说明与复杂度分析

综上所述，图 3 给出了基于最优加权图匹配的服务功能链部署方法的计算流程。从图 3 中可以看出，本文方法的复杂度主要由路径权值更新算法（算法 1）、特征向量分解、匈牙利算法和映射函数修正爬山算法（算法 2）构成。文献[20]和文献[22]表明，特征向量分解和匈牙利算法的复杂度均为 $O(n^3)$ ，而通过对伪代码的分析可知，算法 1 和算法 2 的复杂度分别为 $O(n^2)$ 和

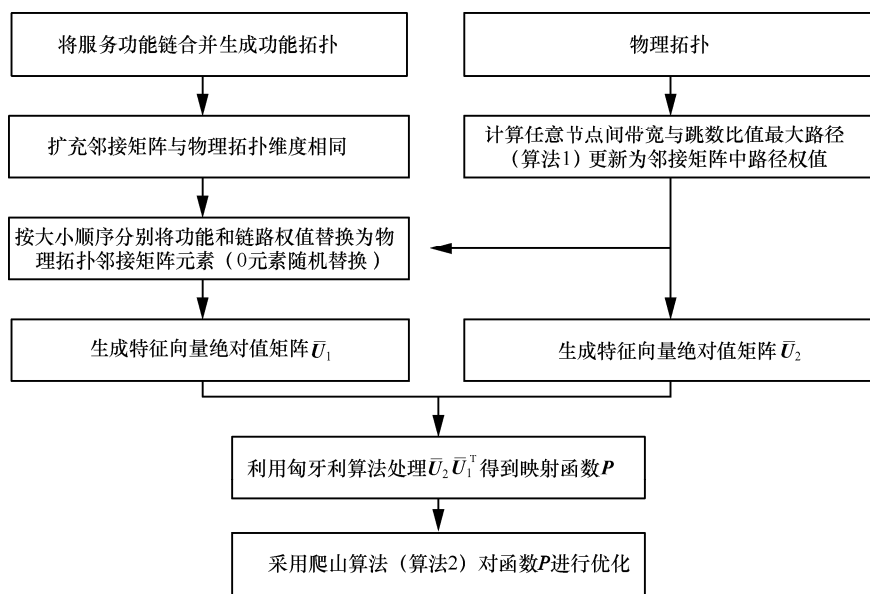


图 3 本文方法流程示意

$O(mn^2)$ ，其中 m 和 n ($m \ll n$) 分别表示爬山算法执行次数和节点个数。

4 性能仿真与分析

为了对本文提出的服务功能链部署方法性能进行验证，在 2.80 GHz CPU，4 GB 内存的 PC 机上通过 Matlab 软件进行了仿真实验并对实验结果处理分析。仿真场景设计为，由 20 个节点构成随机网络拓扑按照 Waxman-Salam 模型^[23]生成，每个节点能够提供的处理能力为 50 MHz，每条链路的总带宽为 50 Mbit/s。网络可以提供 8 种功能，每个需要部署的服务功能链由任意顺序排列的 2~8 个互不相同的功能组成，每个功能需要的处理能力在 0~2 MHz 之间，每条服务功能链所需带宽在 0~5 Mbit/s 之间，每个功能拓扑图由 2 条服务功能链合并生成。为了提高算法评估的精确性，采用蒙特卡洛方法，在每个测试场景下进行了 100 组仿真测试，取平均值作为测试结果。

本文将平均服务请求处理时间、节点负载均衡度、链路带宽均衡度、平均链路剩余带宽和网络能够支持的最大服务请求数作为方法性能的评价指标，其中节点负载均衡度是指节点剩余处理能力的均方差，链路带宽均衡度是指链路剩余带宽的均方差，并将本文方法与文献[20]中同样采用图匹配策略的特征值分解算法和文献[19]中的多层图算法进行比较。特征值分解算法将非相邻节点间的最大路径带宽更新为链路权值，根据邻接矩阵的特征向量计算效率矩阵，最后采用启发式算法依次为每个功能选择符合限制条件的匹配方式。多层图算法将服务功能链分解为多层有向图，通过维特比算法计算每一层的节点选择代价，最后回溯搜索最优的节点和链路匹配方式。

图 4 显示了各算法对服务请求的平均处理时间和服务功能链长度的关系。从图中可以看出，多层图算法的平均处理时间和服务功能链的长度近似成正比关系，这是因为该算法的分层数由服务功能链中的功能个数决定，分层数越多需要的计算时间越长。由于本文方法和特征值分解算法的计算复杂度仅与物理拓扑的节点数相关，因此平均处理时间不会随着服务功能链长度的增加而变化，特征值分解算法的处理时间略小于本文方法，是因为本文方法增加了映射函数的修正过程。

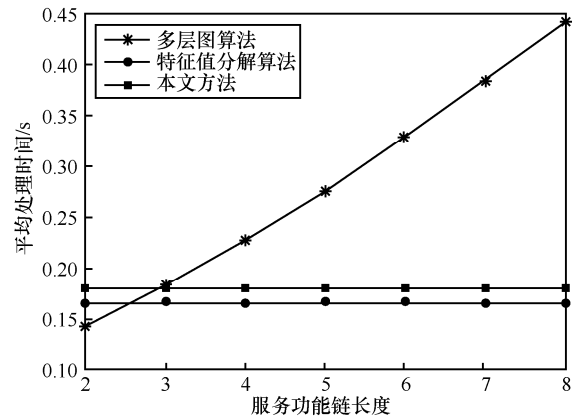


图 4 平均服务请求处理时间

图 5 和图 6 分别给出了服务功能链长度为 5 时，各算法对节点负载和链路带宽的均衡性能。由于多层图算法优先将功能部署在相同节点上，没有考虑对节点负载和链路带宽的均衡，因此节点剩余处理能力和链路剩余带宽的均方差最高。特征值分解算法根据节点和链路的剩余网络资源对服务功能链进行匹配，但是该算法得到的不是最优匹配，因此带宽和负载的均衡性能介于多层图算法和本文方法之间。由于本文方法在匹配过程中对邻接矩阵进行了同构化处理，可以较大概率实现最优匹配，负载均衡度和带宽均衡度都明显优于其他算法，且本文算法通过爬山算法修正映射函数，进一步优化了对负载和带宽的均衡效果。

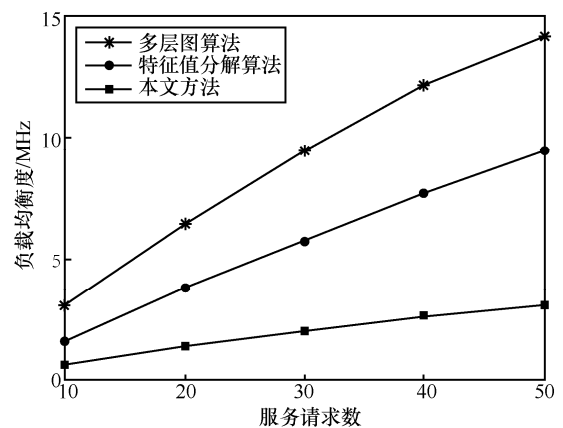


图 5 节点负载均衡性能

图 7 给出了服务功能链长度为 5 时，平均链路剩余带宽随着服务请求增加的变化情况。虽然多层图算法的剩余带宽明显高于其他 2 种算法，但是由于该算法将功能集中部署在相同节点，使得部分节点因剩余处理能力不足而无法支持新功能的部署，当服务请求数大于 40 后，新的服务请求难以找到

符合限制条件的部署方式。与特征值分解算法相比，本文方法采用带宽与跳数的比值来更新物理拓扑邻接矩阵元素，避免为跳数较多的路径匹配带宽需求较高的虚拟链路，因此部署服务功能链需要的带宽更小。图 8 给出了 3 种算法能够支持的最大服务请求数与服务功能链长度的关系。可以看出，本文方法由于降低了带宽损耗，且负载均衡度和带宽均衡度高，因此能够提供的服务请求数明显多于特征值分解和多层图算法，且服务功能链越长，本文方法的优势越明显，这是因为较长的服务功能链对负载和带宽的均衡性能要求更高。

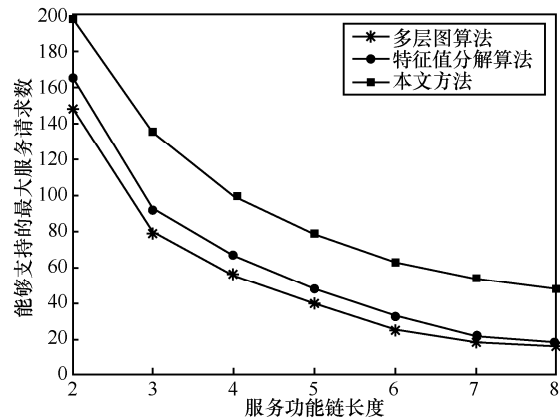


图 8 最大服务请求数

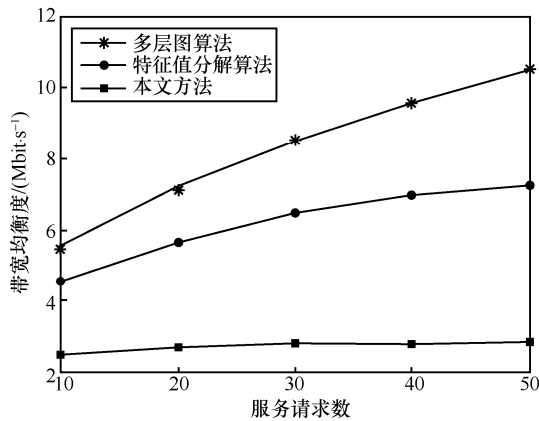


图 6 链路带宽均衡性能

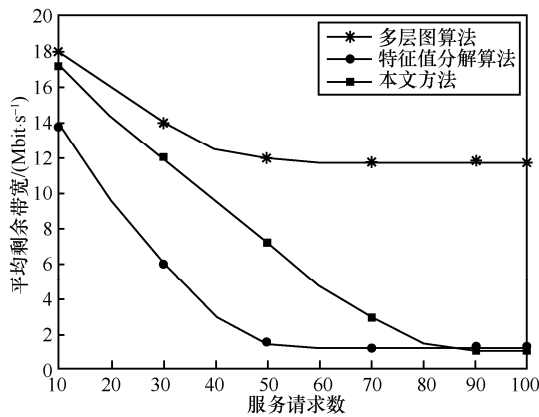


图 7 平均链路剩余带宽

5 结束语

针对资源有限的网络中的服务功能链部署问题，本文提出了一种基于最优加权图匹配的服务功能链部署方法，把多条服务功能链组合为功能拓扑图，通过邻接矩阵的扩充和元素替换实现功能拓扑图与物理拓扑图的同构，采用邻接矩阵特征向量分解算法快速获取最优加权图匹配方式，并通过爬山

算法对匹配结果进一步优化。该方法可以支持多条服务功能链的同时部署，计算复杂度低，具有较高的时效性。与其他算法相比，该方法降低了服务功能链部署所需的链路带宽，对节点负载和链路带宽资源的均衡性能更好，可以在相同的网络资源条件下支持更多的服务请求。下一步研究考虑搭建服务功能链部署原型系统，将本方法在真实环境中进行验证。

参考文献：

- [1] MIJUMBI R, SERRAT J, GORRICO J L, et al. Network function virtualization: state-of-the-art and research challenges[J]. IEEE Communications Surveys & Tutorials, 2017, 18(1):236-262.
- [2] QUINN P, GUICHARD J. Service function chaining: creating a service plane via network service headers[J]. Computer, 2014, 47(11):38-44.
- [3] NUNES B A A, MENDONCA M, NGUYEN X N, et al. A survey of software-defined networking: past, present, and future of programmable networks[J]. IEEE Communications Surveys & Tutorials, 2014, 16(3):1617-1634.
- [4] LI Y, ZHENG F, CHEN M, et al. A unified control and optimization framework for dynamical service chaining in software-defined NFV system[J]. Wireless Communications IEEE, 2015, 22(6):15-23.
- [5] LEE G, KIM M, CHOO S, et al. Optimal flow distribution in service function chaining[C]// The International Conference on Future Internet. ACM, 2015:17-20.
- [6] MARTINI B, PAGANELLI F, CAPPANERA P, et al. Latency-aware composition of virtual functions in 5G[C]// Network Softwarization. IEEE, 2015:1-6.
- [7] MEHRAGHDAM S, KELLER M, KARL H. Specifying and placing chains of virtual network functions[C]// IEEE International Conference on Cloud NETWORKING. IEEE, 2014:7-13.
- [8] LEIVADEAS A, FALKNER M, LAMBADARIS I, et al. Resource management and orchestration for a dynamic service chain steering model[C]// Global Communications Conference. IEEE, 2017:1-6.
- [9] RANKOTHGE W, MA J, LE F, et al. Towards making network function virtualization a cloud computing service[C]// IFIP/IEEE Interna-

- tional Symposium on Integrated Network Management. IEEE, 2015:89-97.
- [10] OTOKURA M, LEIBNITZ K, KOIZUMI Y, et al. Application of evolutionary mechanism to dynamic Virtual Network Function Placement[C]// IEEE International Conference on Network Protocols. IEEE, 2016:1-6.
- [11] SAHHAF S, TAVERNIER W, ROST M, et al. Network service chaining with optimized network function embedding supporting service decompositions[J]. Computer Networks the International Journal of Computer & Telecommunications Networking, 2015, 93(P3):492-505.
- [12] CHUA F C, WARD J, ZHANG Y, et al. Stringer: balancing latency and resource usage in service function chain provisioning[J]. IEEE Internet Computing, 2016, 20(6):22-31.
- [13] KLINKOWSKI M, WALKOWIAK K. On advantages of elastic optical networks for provisioning of cloud computing Traffic[J]. IEEE Network, 2013, 27(27):44-51.
- [14] ZHANG L, ZHU Z. Spectrum-efficient anycast in elastic optical inter-datacenter networks [J]. Optical Switching & Networking, 2014, 14(4):250-259.
- [15] YANG K, ZHANG H, HONG P. Energy-aware service function placement for service function chaining in data centers[C]// Global Communications Conference. IEEE, 2017:1-6.
- [16] FANG W, ZENG M, LIU X, et al. Joint Spectrum and it resource allocation for efficient VNF service chaining in inter-datacenter elastic optical networks[J]. IEEE Communications Letters, 2016, 20(8):1539-1542.
- [17] KUO T W, LIOU B H, LIN C J, et al. Deploying chains of virtual network functions: On the relation between link and server usage[C]// IEEE International Conference on Computer Communications. IEEE, 2016:1-9.
- [18] XIE L, JIANG Y, WANG B, et al. An approach for network function combination based on least busy placement algorithm[J]. China Communications, 2016, 13(S1):167-176.
- [19] BARI F, CHOWDHURY S R, AHMED R, et al. Orchestrating virtualized network functions[J]. IEEE Transactions on Network & Service Management, 2016, PP(99):1-1.
- [20] MECHTRI M, GHRIBI C, ZEGHLACHE D. A scalable algorithm for the placement of service function chains[J]. IEEE Transactions on Network & Service Management, 2016, 13 (3) :533-546
- [21] UMEYAMA S. An eigendecomposition approach to weighted graph matching problems[J]. Pattern Analysis & Machine Intelligence IEEE Transactions on, 1988, 10(5):695-703.
- [22] PAPADIMITRIOU C H, STEIGLITZ K. Combinatorial optimization: algorithms and complexity[J]. IEEE Transactions on Acoustics Speech & Signal Processing, 1982, 32(6):1258-1259.
- [23] SALAMA H F. Multicast routing for real-time communication of high-speed networks[D]. Raleigh: North Carolina State University, 1996.

[作者简介]



李丹 (1989-), 男, 辽宁沈阳人, 博士, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为新型网络体系结构、路由交换技术。



兰巨龙 (1962-), 男, 河南郑州人, 博士, 国家数字交换系统工程技术研究中心教授、博士生导师, 主要研究方向为网络体系结构、信息安全。



王鹏 (1985-), 男, 河南周口人, 博士, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为新型网络体系结构、路由技术。



胡宇翔 (1982-), 男, 河南周口人, 博士, 国家数字交换系统工程技术研究中心副研究员、硕士生导师, 主要研究方向为新型网络体系结构、网络安全。