

## 面向软件定义网络的服务功能链优化部署算法研究

卢昱 刘益岑\* 李玺 陈兴凯 乔文欣 陈立云

(陆军工程大学信息工程系 石家庄 050003)

**摘要:** 针对网络功能虚拟化(NFV)环境下, 现有服务功能链部署方法无法在优化映射代价的同时保证服务路径时延的问题, 该文提出一种基于IQGA-Viterbi学习算法的服务功能链优化部署方法。在隐马尔可夫模型参数训练过程中, 针对传统Baum-Welch算法训练网络参数容易陷入局部最优的缺陷, 改进量子遗传算法对模型参数进行训练优化, 在每一迭代周期内通过等比例复制适应度最佳种群的方式, 保持可行解多样性和扩大空间搜索范围, 进一步提高模型参数的精确度。在隐马尔可夫链求解过程中, 针对隐含序列无法直接观测这一难点, 利用Viterbi算法能精确求解隐含序列的优势, 解决有向图网络中服务路径的优化选择问题。仿真实验结果表明, 与其它部署算法相比, 所提IQGA-Viterbi学习算法能有效降低网络时延和映射代价的同时, 提高了网络服务的请求接受率。

**关键词:** 网络功能虚拟化; 服务功能链部署; 隐马尔可夫模型; 量子遗传算法; Viterbi算法

中图分类号: TN915.81

文献标识码: A

文章编号: 1009-5896(2019)01-0074-09

DOI: 10.11999/JEIT180264

## Research on Placement Algorithm of Service Function Chaining Oriented to Software Defined Networking

LU Yu LIU Yicen LI Xi CHEN Xingkai QIAO Wenxin CHEN Liyun

(Information Engineering Department, Army Engineering University, Shijiazhuang 050003, China)

**Abstract:** For Network Function Virtualization (NFV) environment, the existing placement methods can not guarantee the mapping cost while optimizing the network delay, a service function chaining optimal placement algorithm is proposed based on the IQGA-Viterbi learning algorithm. In the training process of Hidden Markov Model (HMM) parameters, the traditional Baum-Welch algorithm is easy to fall into the local optimum, so the quantum genetic algorithm is proposed, which can better optimize the model parameters. In each iteration, the improved algorithm maintains the diversity of feasible solutions and expands the scope of the spatial search by replicating the best fitness population with equal proportion, thus improving the accuracy of the model parameters. In the process of solving Hidden Markov chain, to overcome the problem that can not be directly observed for hidden sequences, Viterbi algorithm can solve the implicit sequences exactly and solve the problem of optimal service paths in the directed graph. Experimental results show that the network delay and mapping costs are lower compared with the existing algorithms. In addition, the acceptance ratio of requests is raised.

**Key words:** Network Function Virtualization (NFV); Service Function Chaining (SFC) placement; Hidden Markov Model (HMM); Quantum Genetic Algorithm (QGA); Viterbi algorithm

### 1 引言

随着互联网服务需求不断扩大, 传统静态僵化的服务模式已无法满足租户的需求, 存在服务功能可拓展性差、网络维护成本高等弊端, 且静态僵化

的服务功能难以适应当前互联网多租户模式的动态需求。为此, 新型动态服务模式研究成为近来研究的热点<sup>[1,2]</sup>。特别是软件定义网络<sup>[3]</sup>(Software Defined Networking, SDN)和网络功能虚拟化<sup>[4]</sup>(Network Function Virtualization, NFV)技术的结合, 增强网络的灵活性和可拓展性, 达到基础设施资源共享的效果, 为动态服务功能链(Service Function Chaining, SFC)技术发展提供了支撑。

但是, 动态服务功能链技术研究还处于起步阶段, 在应用中存在很多突出问题, 其中服务链优化部署是亟待解决的问题之一。由于服务链部署为

收稿日期: 2018-03-21; 改回日期: 2018-07-27; 网络出版: 2018-08-24

\*通信作者: 刘益岑 18419764051@163.com

基金项目: 国家自然科学基金(51377170, 61271152), 国家青年科学基金(61602505)

Foundation Items: The National Natural Science Foundation of China (51377170, 61271152), The National Youth Science Fund Project (61602505)

NP难问题<sup>[5]</sup>，已展开的工作主要集中在设计启发式算法或精确算法以获取近似最优解。Bhamare等人<sup>[6]</sup>针对VNF的编排顺序和流量调度问题，提出了一种基于亲和度的启发式算法的服务功能部署策略。该方案是将VNF节点和逻辑链路映射过程同时完成，降低了部署实现的复杂度和服务处理时间，但算法不能保证解的最优性，搜索容易陷入局部最优。Bari等人<sup>[7]</sup>提出了一种基于Viterbi算法的虚拟网络功能自适应部署方法，该方法通过设置目标函数和相应的约束条件，可以得到全局的最优解，但Viterbi算法仅能对给定模型参数条件下的服务路径进行求解。Xiong等人<sup>[8]</sup>提出了一种网络虚拟化条件下的VNF编排方法，将服务功能路径以量子编码的形式进行迭代求解，该方法在求解优化路径时具有简单、鲁棒性等特点，但在搜索后期算法自身的盲目性和随机性导致出现早熟的问题。Lukovszki等人<sup>[9]</sup>提出了一种基于Greedy算法的服务功能链部署方法，该方法通过穷举所有满足连通性和策略需求的路径，并在其中选择部署开销最小的服务路径，但请求处理时间较长，时延优化精度不高。Moens等人<sup>[10]</sup>提出了一种虚拟网络功能部署模型，该模型采用决策树算法实现对功能模块的映射和编排，所提算法仅能得到映射收益的最优解，而无法兼顾时延优化的精确度。

上述部署方法从不同的角度进行了探索，其主要研究不足在于：上述部署方法的优化目标都是单一的最小化映射代价，或者单一的最小化时延。因此，本文更加侧重于解决服务链部署过程中的映射代价-时延多目标优化问题，从提高服务链部署综合性能的角度考虑，利用加权求和的方法将多个目标转化为单一的部署开销目标，提出一种基于IQGA-Viterbi算法的优化部署方法，以实现网络资源的优化分配。

## 2 网络模型与问题描述

### 2.1 基于SDN的服务功能链部署模型

如图1所示，该模型分为编排平面、控制平面和数据平面。编排平面作为虚拟网络功能(Virtual Network Function, VNF)的开发环境，其主要作用是对网络中VNF进行集中化管理，并根据不同的业务需求和适用场景构建出SFC策略来管控网络。控制平面根据策略构建结果组合VNF，进而形成逻辑功能链，SDN控制器根据服务功能资源需求和底层资源信息，并按照一定的部署策略将VNF和虚拟链路映射到底层，进而形成具体服务路径。数据平面主要包括通用型的硬件设备(如标准化转发设备和x86硬件资源设备等)，其主要作用

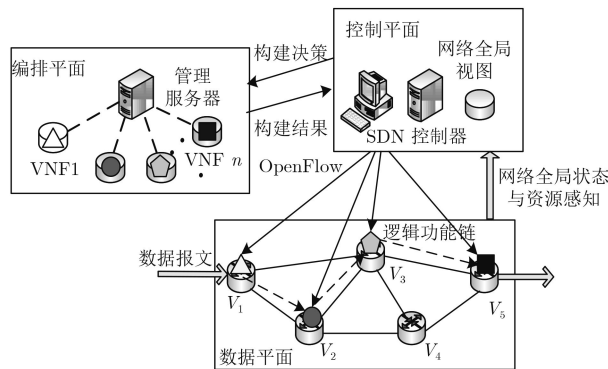


图1 基于SDN的服务功能链部署模型

是接收控制平面下发的规则，对数据报文执行具体的功能处理。

### 2.2 服务功能链部署问题描述

服务功能链部署过程可描述为在接收到用户的SFC服务请求后，SDN控制器中的资源管理、VNF编排模块根据服务请求和底层资源状态，按照指定的部署策略找到VNF和虚拟链路的优化部署位置，并生成一条满足特定功能、性能的服务路径。如图2所示，服务功能链部署过程实质为两级映射模型，即“SFC策略—逻辑功能链”和“逻辑功能链—具体服务路径”。“SFC策略—逻辑功能链”过程是应用程序通过北向接口实现VNF的动态编排组合；“逻辑功能链—具体服务路径”过程是通过南向接口实现对底层资源的合理分配。

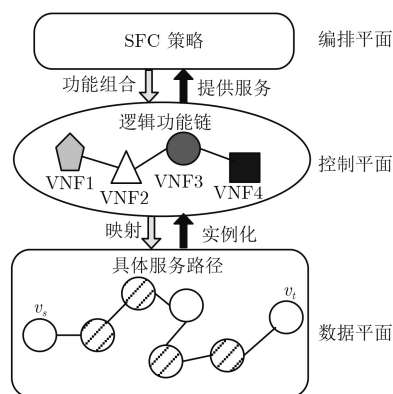


图2 二级映射模型

**定义1 SFC策略：**包含目标集合和服务功能集合，目标集合表示所需执行处理动作的目标，服务功能集合表示数据报文必须依次遍历的服务功能。SFC策略可以表示为集合  $P = \{v_s, v_t, (c_1, c_2, \dots, c_m)\}$ ，式中  $C = \{c_1, c_2, \dots, c_m\}$  表示从源端节点  $v_s$  到目的端节点  $v_t$  的数据报文所需经过的服务功能序列， $m$  表示请求的服务功能数量。

**定义2 逻辑功能链：**控制器根据SFC策略组

合VNF功能模块形成的逻辑功能链, 可用集合  $F = \{f_1, f_2, \dots, f_m\}$  表示VNF逻辑功能链。采用赋权的有向图  $G^v$  表示全部VNF可部署节点及其上下文连接关系, 记为  $G^v = (N^v, L^v)$ , 式中  $N^v$  表示VNF的逻辑节点集合, 即  $N^v = \{n_1^v, n_2^v, \dots, n_{i-1}^v, n_i^v, n_{i+1}^v, \dots, n_m^v | 1 \leq i \leq m\}$ ,  $L^v$  表示逻辑节点之间的虚拟链路集合, 即  $L^v = \{(n_i^v, n_j^v), \dots | 1 \leq i < j \leq m\}$ 。

**定义 3** 具体服务路径: 根据指定的目标函数为逻辑功能链中的VNF和虚拟链路找到对应的最优放置位置, 形成从源端节点到目的端节点的具体服务路径。底层网络由所有物理节点及连接链路组成, 可以用赋权无向图  $G^s = (N^s, L^s)$  表示, 其中  $N^s$  表示底层物理节点集合, 即  $N^s = \{n_1^s, n_2^s, \dots, n_{j-1}^s, n_j^s, n_{j+1}^s, \dots, n_k^s | 1 \leq j \leq k\}$ ,  $L^s$  表示物理节点之间的物理链路集合, 即  $L^s = \{(n_i^s, n_j^s), \dots | 1 \leq i < j \leq k\}$ , 其中  $k$  表示底层网络物理节点数量。

以图2“SFC策略—逻辑功能链—具体服务路径”过程为例, 来说明服务功能链部署流程。首先, 编排平面根据租户需求构建SFC策略; 其次, 控制平面根据SFC策略组合形成由4个VNF组成的逻辑功能链; 最后, 将形成的逻辑功能链按照一定的策略映射到物理底层网络, 若映射成功, 则分配相应的实例化资源, 进而形成从源端节点到目的端节点的具体服务路径。在具体服务路径中, 带有阴影的节点表示提供服务的节点, 未带有阴影的物理节点仅仅起到转发数据的作用。

### 2.3 服务功能链部署评价指标

与文献[7,8]类似, 为一个服务请求进行动态部署, 其部署质量可主要从时延、映射代价2个方面进行评价。设VNF节点  $n_i^v$  选择物理节点  $n_j^s$  的指示变量为  $\mathbf{X}_{m \times k}$ ,  $\mathbf{X}_{m \times k} = 1$  表示选择, 否则为0。据此, 评价指标的形式化表述如下

(1)时延: 服务提供商成功将服务请求映射到底层物理网络过程中, 为租户构建出的一条端到端服务路径, 时延表示数据流从源端节点发出到目的节点所耗费的时间。服务路径时延的降低有助于提高用户的使用率, 为服务提供商获取更多的收益。时延由VNF处理时延和传输链路时延组成, 其定义为

$$D = \sum_{i \in N^v} \sum_{j \in N^s} \mathbf{X}_{i,j} \mathbf{W}_i^p + \sum_{i,j \in N^v} \sum_{k,l \in N^s} \mathbf{X}_{i,k} \mathbf{X}_{j,l} \mathbf{W}_{k,l}^t \quad (1)$$

式中,  $\mathbf{W}_{1 \times m}^p$  表示VNF处理时延开销矩阵,  $\mathbf{W}_{n \times n}^t$  表示传输时延开销矩阵。

(2)映射代价: 服务提供商成功将请求映射到

底层物理网络过程中, 映射代价表示所使用节点计算资源和链路带宽资源支出的费用之和。在物理资源容量有限的条件下, 高效资源分配有助于服务提供商降低成本。映射代价由逻辑链路映射代价和VNF节点映射代价组成, 其定义为

$$M = \sum_{i \in N^v} \sum_{j \in N^s} \mathbf{X}_{i,j} \mathbf{E}_i^p + \sum_{i,j \in N^v} \sum_{k,l \in N^s} \mathbf{X}_{i,k} \mathbf{X}_{j,l} \mathbf{E}_{k,l}^t \quad (2)$$

式中,  $\mathbf{E}_{1 \times m}^p$  表示VNF节点映射代价矩阵,  $\mathbf{E}_{n \times n}^t$  表示逻辑链路映射代价矩阵。

(3)部署开销: 上述定义的时延和映射代价是衡量部署质量的重要指标, 为了同时对这2个指标进行优化, 利用加权求和的方法将多个目标转化为单一的部署开销目标, 定义部署的开销为

$$C = \alpha D + \beta M \quad (3)$$

式中,  $D$ 和 $M$ 已进行了无量纲处理, 缩放因子 $\alpha$ 和 $\beta$ 用于调整优化目标, 以适应不同场景对部署策略的影响,  $C$ 值越小, 服务提供商支出成本越低、收益获取越高。

## 3 算法描述

### 3.1 基于IQGA-Viterbi学习算法的服务链动态部署

在服务功能链部署问题中, VNF编排关系及其上下文组成是可观测的, 但观测不到VNF具体放置的物理节点, 即无法观测具体服务路径。由此可知, 具体服务路径是一个含有隐含未知参数的马尔科夫过程, 可看成一条隐马尔科夫链。因此, 服务链部署问题可视为隐马尔科夫模型的求解过程<sup>[11]</sup>。对于参数已知的隐马尔科夫模型, 可用基于动态规划的值迭代或策略迭代算法<sup>[7]</sup>来求解隐含序列。但是传统单一的动态规划算法难以准确获取底层网络参数, 无法在短时间内实现服务链优化部署并进行物理资源的精确分配。此时对每个服务请求时间内的网络参数训练成为一种有效的手段, 本文提出了一种基于IQGA-Viterbi学习算法的优化部署方法。该方法的主要分为以下几步: 首先, 在SFC请求达到之后, 根据初始网络状态描述服务链可部署的有向图 $\lambda^0$ ; 其次, 积累一段时间的运行数据, 采用改进量子遗传(Improved Quantum Genetic Algorithm, IQGA)学习算法对网络设施数据进行训练, 以准确地得到底层网络参数 $\lambda$ ; 最后通过Viterbi算法求得最小化开销的服务路径, 并输出优化部署方案 $\pi_s$ 。

在网络参数训练过程中, Baum-Welch算法<sup>[12]</sup>对于多目标优化问题往往只能到一个局部最优解。为了精确获取网络环境中的底层参数, 本文借鉴量子遗传算法并对其进行改进。量子遗传算法<sup>[13]</sup>是一

种基于量子计算和人工免疫的智能方法，适用于动态系统的多目标优化问题。IQGA算法的改进之处在于在每一迭代周期内将适应度最优的种群进行等比例复制，选择最佳的种群保存下来，选择后种群的规模与选择前保持一致，其主要特点是增加了可行解的多样性和扩大了空间搜索范围，进而避免了Baum-Welch算法早熟的问题。IQGA算法将初始网络状态参数 $\lambda^0=(\mathbf{A}^0, \mathbf{B}^0, \mathbf{\Pi}^0)$ 进行量子编码，如图3所示，每条染色体由 $\mathbf{\Pi}$ 编码区、 $\mathbf{A}$ 编码区和 $\mathbf{B}$ 编码区3个部分组成，即HMM的3个参数，其中 $\mathbf{\Pi}$ ， $\mathbf{A}$ ， $\mathbf{B}$ 分别称之为初始状态的参数分布矩阵、传输路径开销矩阵、VNF节点处理开销矩阵。在各编码操作之前需进行归一化处理，且各编码区不能进行跨区操作。给定初始网络参数 $\lambda^0$ 和训练所用的观测序列集合 $O$ ，以该模型下观测序列部署开销最小化为进化目标，对参数反复重估、迭代直至满足终止条件Max\_step，即模型参数收敛到一定范围不再明显改善为止，从而训练出一组最精确的网络底层参数 $\lambda=(\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ 。

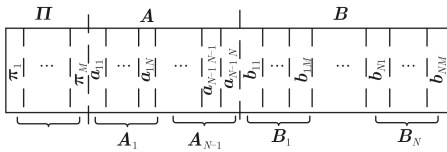


图3 HMM参数的染色体编码

在服务路径求解过程中，Viterbi算法<sup>[14]</sup>可用于在已知HMM参数 $\lambda=(\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ 条件下寻找观测事件的隐含序列 $S$ 。因此，本文借鉴Viterbi算法来搜索最小开销的服务路径。假设 $t$ 时刻的部分网络视图如图4所示， $f_{m-1}$ 和 $f_m$ 分别表示服务链中相邻的两种VNF。 $B_j(f_m)$ 表示时刻 $t$  VNF节点 $f_m$ 部署在服务节点 $s_j$ 的开销值， $A_{ij}$ 表示时刻 $t$  VNF节点 $f_{m-1}$ ， $f_m$ 之间逻辑链路映射到服务节点 $s_i$ ， $s_j$ 之间链路的开销值， $C_i^t$ 表示在时刻 $t$ 所选择节点 $s_i$ 部署VNF节点后服务路径的开销值。定义Viterbi变量 $\delta_t(j) =$

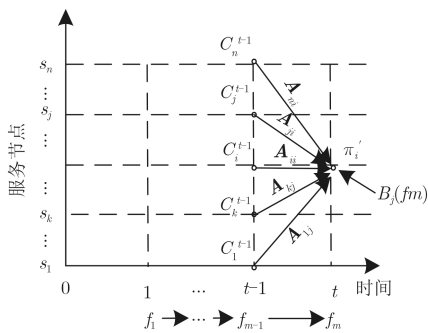


图4  $t$ 时刻网络视图

$\min [\mathbf{A}(s_i, s_j) + \mathbf{B}(s_j, f_m)]$ ,  $1 \leq i < j \leq n$  以求解最小开销路径问题的最优子结构，其表示在时刻 $t$ ，按照服务链 $l$ 的顺序约束 $\varphi_l$ 完成在服务节点 $s_j$ 部署VNF节点 $f_m$ 以及服务节点 $s_i$ 到 $s_j$ 之间逻辑链路映射的最小开销。定义标记函数 $\varphi_t(i) = \arg \min [\mathbf{A}(s_i, s_j) + \mathbf{B}(s_j, f_m)]$ ,  $1 \leq i < j \leq n$ 以回溯产生最小开销的具体服务路径，其表示在时刻 $t$ 取得最小开销值序列所选择的 $t-1$ 时刻服务节点。定义变量 $\pi_{f_m}$ 为顺序约束 $\varphi_l$ 中VNF节点 $f_m \in O$ 的部署位置。服务路径的构造主要分为递归和回溯过程：首先，从流入源端节点 $v_s$ 开始，在每层递归过程中计算 $t-1$ 时刻部署VNF节点 $f_{m-1} \in O$ 到 $t$ 时刻部署VNF节点 $f_m \in O$ 的最小开销，即Viterbi变量 $\delta_t(j)$ ，对于 $t$ 时刻完成VNF节点 $f_m \in O$ 在服务节点 $s_i$ 部署，需要考虑 $t-1$ 时刻中所有可能的服务路径开销值 $C_i^{t-1}$ ，可根据 $\delta_t(j)$ 计算出最小开销服务路径在时刻 $t$ 所选择服务节点 $s_i$ ，直到递推到目的端节点 $v_t$ ，即可得到具体路径的最小开销值 $\min \delta_t(f_m)$ ；其次，以服务链 $l$ 中最后一个VNF节点所选择的服务节点 $s_n$ 为起点，沿着标记函数 $\psi_t(i)$ 计算出上一时刻所选择的服务节点，依次回溯直到计算出 $s_1$ ，即可得到最小开销值的服务路径 $S$ ，并输出部署方案 $\pi_s$ 。

IQGA-Viterbi学习算法具体描述如表1所示。

### 3.2 算法复杂性理论分析

相比给定HMM模型参数采用Viterbi算法求解优化路径的情形，IQGA-Viterbi部署方法的请求处理时延稍微变长，原因是该算法以牺牲一定计算时间对模型参数进行训练，得到一组最精确的底层网络模型参数后，再利用Viterbi算法来实现服务链的优化部署。假设请求中有 $n$ 个VNF节点需要部署，底层拓扑结构中 $m$ 个物理节点。IQGA-Viterbi算法分为两个阶段，在HMM参数训练过程的计算量主要在于种群复制、变异和选择等，其时间复杂度为 $O(mn)$ ，而在路径求解过程中采用递归方法搜索的Viterbi算法，其时间复杂度为 $O(m^2n)$ 。综上所述，IQGA-Viterbi算法的时间复杂为 $O(m^2n+mn)$ 。

## 4 实验结果及性能分析

算法实验包括两部分，首先为了验证本文改进的量子遗传算法(IQGA)的优化性能，将其与Baum-Welch算法<sup>[12]</sup>(BW)和量子遗传算法<sup>[15]</sup>(QGA)作对比，并选取均方根误差作为评价指标，其反映训练算法的泛化能力。其次，为了验证本文所提出的部署算法(IQGA-Viterbi)的有效性，将其与贪婪算法<sup>[16]</sup>(Greedy)、量子遗传算法<sup>[17]</sup>(QGA)和维特比

表1 IQGA-Viterbi学习算法具体过程

输入：服务请求序列  $O$ ，底层网络初始参数  $\lambda^0$

输出：最小化开销部署策略  $\pi_s$

步骤1 IQGA算法初始化。设置训练所用的观测序列  $O$  数目为  $K$ ，将底层初始网络参数编码成种群大小为  $M$ 、量子比特编码长度为  $N$  的染色体。记迭代次数  $t$  的种群  $P(t) = \{C_1^{(t)}, C_2^{(t)}, \dots, C_M^{(t)}\}$ ，其中种群个体  $C(t)$   $M(m=1, 2, \dots, M)$  的量子比特表示，初始状态的网络参数可表示为如式(4)。

$$C_m^{(0)} = \begin{bmatrix} \alpha_{m1}^{(0)} & \alpha_{m2}^{(0)} & \dots & \alpha_{mN}^{(0)} \\ \beta_{m1}^{(0)} & \beta_{m2}^{(0)} & \dots & \beta_{mN}^{(0)} \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & \dots & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} & \dots & 1/\sqrt{2} \end{bmatrix}, m = 1, 2, \dots, M \quad (4)$$

步骤2 种群  $P(t)$  复制、变异和选择。对于种群  $P(t)$  进行等比例复制生成  $P'(t)$ ，利用高斯变异的操作对种群  $P'(t)$  进行更新，并生成种群  $P''(t)$ ，通过选择等比例压缩  $P''(t)$  生成  $P(t+1)$ 。

步骤3 评价种群  $P(t+1)$ 。对于经选择后种群  $P(t+1)$  中个体的量子位进行测量，随机生成  $[0, 1]$  的随机数，若该随机数大于或等于  $|\alpha|^2$  或  $|\beta|^2$ ，则测量的结果取值为1，否则取0。该过程是得到每个个体测量后的状态  $X_C = \{x_1, x_2, \dots, x_N\}$ ，并将其转化为十进制数，代入目标函数如式(5)。

$$\text{Fitness}(X_C) = \frac{1}{K} \sum_{k=0}^K \frac{1}{C^{(k)}} \quad (5)$$

步骤4 更新重估计开销值  $\hat{a}_{ij}$ ， $\hat{b}_{ik}$ 。记录并保存当前迭代次数  $t$  的最佳个体，以及其对应的开销数值矩阵  $\hat{A}_{n \times n}$  和  $\hat{B}_{m \times n}$ ，据此更新模型重估参数  $\hat{a}_{ij}$ ， $\hat{b}_{ik}$ ，在迭代次数  $t+1$  时更新网络底层视图，并继续对底层网络参数反复迭代、重估。采用最大进化迭代次数  $\text{Max\_step}$  作为算法的终止条件，判断是否到达最大进化迭代次数，若是则终止进化后得到一组最优的网络底层参数  $\lambda = (A, B, \Pi)$ ，否则返回至步骤2继续迭代。

步骤5 Viterbi算法参数初始化。将IQGA算法得到  $\lambda = (A, B, \Pi)$  和观测序列  $O$  作为Viterbi的输入。

步骤6 Viterbi变量  $\delta_t(j)$  递推。从源端节点  $v_s$  开始，根据服务链中VNF编排顺序  $\varphi_t$ ，在每个迭代周期内计算从服务节点  $s_i$  到候选节点  $s_j$  的最小开销，即Viterbi变量  $\delta_t(j) = \min [A(s_i, s_j) + B(s_j, f_m)]$  的计算，按照式(6)的递推方式，搜索整个观察序列  $O$  条件下的具体服务路径  $S$ ，直到最后流出目的节点  $v_t$ 。位于不同时刻的部署开销值，取其中最小值，进而得到最小开销服务路径的函数值  $\min \delta_t(f_m)$ 。

$$\left. \begin{aligned} d_{t_1}(f_1) &= d(s_1) \\ d_{t_2}(f_2) &= \min \{d_{t_1}(f_1) + A(s_1, s_k) + B(s_k, f_2)\} \\ &\vdots \\ d_{t_m}(f_m) &= \min \{d_{t_{m-1}}(f_{m-1}) + A(s_j, s_n) + B(s_n, f_m)\} \end{aligned} \right\} \quad (6)$$

步骤7 标记函数  $\varphi_t(i)$  回溯。记录位于当前时刻最小开销序列应选取的前一时刻部署服务节点，利用标记函数  $\varphi_t(i) = \arg \min [A(s_i, s_j) + B(s_j, f_m)]$  依次回溯，以服务链  $l$  中最后一个VNF节点选取开销最小时所选择的服务节点  $s_n$  为起点，沿着函数  $s_{i-1} = \varphi_t(s_i)$  给出上一时刻所选择的服务节点，直到计算出  $s_1$ ，得到开销最小状态序列路径，并输出服务路径的构造方案  $\pi_s = \{\pi_{f_1}, \pi_{f_2}, \dots, \pi_{f_m}\}$ 。

算法<sup>[18]</sup>(Viterbi)进行对比，选取请求接受率、请求处理时间、映射代价和时延作为性能指标进行仿真，其中请求处理时间反映算法计算性能，请求接受率反映算法解空间的搜索性能，映射代价和时延反映算法搜索最优解的性能。

#### 4.1 实验环境和参数设置

本实验在配置为Intel Core i7-3770 3.60 GHz，8 GB内存的Linux系统PC机上运行。网络拓扑结构是利用GT-ITM<sup>[19]</sup>工具生成，算法程序通过Matlab软件运行。本文借鉴SNDlib库中的Abilene骨干网测试例子<sup>[20]</sup>作为网络拓扑结构来进行实验，如图5所示，该拓扑由12个节点和15条链路组成，以节点12为数据流量出口，其余11个节点作为流量的入口。假设所有节点部署在云数据中心，所有节点能够承载服务功能。依据文献<sup>[21,22]</sup>，底层拓扑是具有一定动态性的网络环境，因此选择期望(单位：ms)取值范围为(0, 1]的平稳随机分布表示时延，底层节点的计算能力和链路的带宽服从[10000，

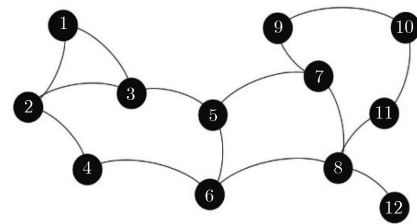


图5 Abilene骨干网拓扑结构

15000]均匀分布，链路和节点资源的单位成本均为10。SFC请求到达强度服从100~1000的泊松分布。每条SFC请求由不同类型的服务功能组成，其数量服从2~5的随机分布，每个服务所需的计算资源量在[1, 10]均匀分布，逻辑链路所需的带宽资源量在[1, 10]均匀分布。优化目标中的缩放因子  $\alpha$  和  $\beta$  均设为1，在IQGA算法中，种群规模  $M=20$ ，高斯变异概率  $r=0.1$ ，最大迭代次数  $\text{Max\_step}=100$ 。为了使结果更准确，仿真共进行了10次，取得的所有实验结果均为平均值。

为了验证IQGA学习算法在设定最大迭代次数  $Max\_step=100$  内的泛化能力，本节选取交叉验证法中的训练均方根误差作为泛化性能的评价指标，其反映学习算法最优解的稳定性。记迭代次数为  $t$  的种群中个体  $C_i^{(t)}$  的均方根误差  $\sigma(C_i^{(t)})$  计算式为

$$\sigma(C_i^{(t)}) = \sqrt{\frac{1}{S} \sum_{i=1}^S [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]} \quad (7)$$

式中， $S$  表示训练的样本数量，即模型状态数量； $x_i, y_i$  分别为样本链路和节点的实际开销值； $\hat{x}_i, \hat{y}_i$  分别为样本链路和节点的重估计开销值。本节所采用上述设置的网络拓扑进行训练，得到表2和图6所示的实验结果。

表 2 算法测试结果统计与比较

算法	误差最小值	平均迭代数
IQGA	0.214	20
QGA	0.223	48
BW	0.248	76

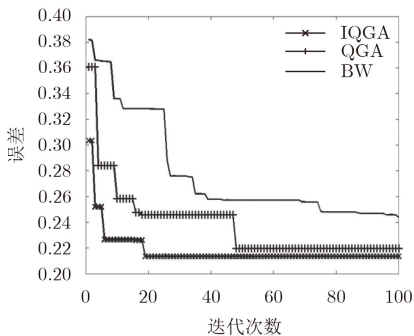


图 6 不同算法的寻优性能比较

表2给出了3种算法的均方根误差性能指标比较。在平均迭代次数方面，IQGA算法平均在20代以内找到最优解，QGA算法平均在48代以内，而BW算法在76代以上。在均方根误差值方面，IQGA算法搜索到的误差值最小，而BW算法最大，QGA算法适中，即IQGA算法最优解更稳定。原因在于BW算法通过迭代循环直至最大迭代次数为止，但容易陷入局部最优。QGA算法采用量子比特编码和量子旋转门等方式进行种群变异、选择，通过较小的规模能获得较优解。而IQGA算法在最优解和收敛速度方面表现出良好的性能，该算法对适应度值最优的父代进行复制，保持解的多样性和扩大空间搜索范围，进一步提高了寻优能力、收敛速度。因此IQGA算法在最大迭代次数  $Max\_step=100$  范围内能够更有效地训练优化HMM模型参数。

图6显示了各算法的寻优性能对比，分析结果与表2结果相一致。

#### 4.2 部署算法性能对比

(1)请求处理时间：表示从发出请求到完成部署过程的运行时间，运行时间的取值越小越好。实验中生成100~1000次请求强度，分别统计各算法的请求处理时延，如图7所示。

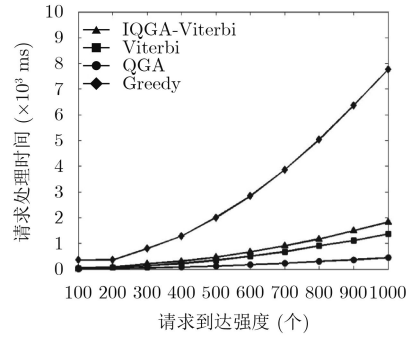


图 7 不同请求强度下的服务请求处理时间

图7表示不同请求到达强度下各算法的请求处理时间，QGA算法的请求处理时间最小，能够更快速地完成服务链的部署，反之Greedy算法的请求处理时延较大，在请求数目为1000时，Greedy算法的处理时间约为QGA算法的18~20倍，Viterbi算法和IQGA-Viterbi算法居中。分析其原因是Greedy实质是两步映射算法，即VNF节点映射和逻辑链路映射，在VNF映射阶段过程中穷举出底层满足约束条件的节点，时间复杂度为  $O(m^n)$ ，在逻辑链路映射过程中通过最短路径选择节点之间，带宽占用最少的路径，时间复杂度为  $O(m^2)$ ，完成映射的VNF节点需要等待队列中下一个VNF节点完成映射后才能建立服务路径，因此Greedy不仅搜索空间度大，而且虚拟链路映射容易受到VNF映射的影响，导致队列中VNF等待的时间较长，算法运行效率低，则总的复杂度则为  $O(m^n + m^2)$ 。Viterbi采用无向迭代的方式搜索开销最小的路径，算法的运行过程中需要遍历一次观测序列，时间复杂度为  $O(n)$ ，而每次需要计算当前各个状态开销最小的前一状态，时间复杂度为  $O(m^2)$ ，则总的复杂度为  $O(m^2n)$ 。IQGA-Viterbi相比单一的动态规划算法时间更长，原因是该算法以增加一定的计算时间为代价来获取准确网络参数，根据3.2节的算法复杂度的理论分析，本文算法的时间复杂度为  $O(mn + m^2n)$ 。QGA算法计算量主要在于初始化、适应度求解和变异阶段，能够在较小种群规模求得相对较优解，其时间复杂度为  $O(mn)$ ，但易止于局部最优解，因此请求处理时

间消耗最少。由图7可知,各算法变化趋势与上文理论分析相一致。

(2)请求接受率:表示成功部署的SFC条数与SFC请求总条数之比,取值越大越好。实验中生成100~1000次请求强度,分别统计各算法的请求接受率,如图8所示。图8表示不同请求到达强度下各算法的请求接受率,IQGA-Viterbi算法接受率约为92.3%,Viterbi算法接受率约为88.9%,QGA算法接受率约为82.1%,Greedy算法接受率约为67.3%。分析原因是Greedy以链路时延作为选择策略,造成底层链路频繁复用,增加了节点处理的排队时间和底层网络负载,进而降低了请求接受率。QGA容易在迭代后期陷入局部最优,大量的空闲VNF占用底层网络资源,容易形成资源瓶颈,而Viterbi采用递归的方法搜索最优路径,同时考虑节点、链路资源状态,对VNF进行细致化管理,提高了物理资源利用率,可接受更多的服务请求。相比Greedy算法、QGA算法和Viterbi算法,IQGA-Viterbi算法综合考虑了时延和映射代价两个因素,在避免资源过量的同时保证资源的高效利用,尽最大限度为队列中其它请求完成部署留出实例化资源,因此具有最高的请求接受率。

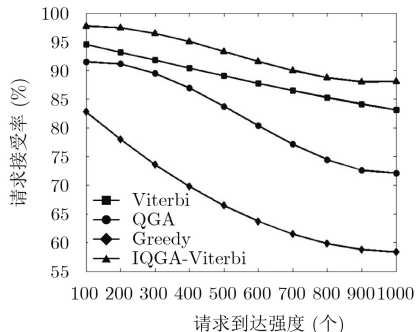


图8 不同请求强度下的服务请求接受率

(3)映射代价:表示逻辑功能链映射到底层物理网络过程中,使用节点计算资源和链路带宽资源支出的费用之和,映射代价的取值越小越好。实验中生成100~1000次请求强度,分别统计各算法的映射代价,如图9所示。

图9表示不同请求到达强度下各算法的映射代价,IQGA-Viterbi算法的映射代价最小,反之Greedy算法的映射代价最大,Viterbi算法和QGA算法居中。分析原因是Greedy和Viterbi在降低时延的同时仅能获得映射代价的次优解,但Viterbi综合考虑节点和链路的资源状态,增加了资源分配的灵活性,降低了一定的映射代价,而Greedy始终以时延作为转发抉择,因此该部署方

法造成额外的映射代价。QGA采用并行的计算方式,在多目标优化求解问题上能够表现出相对较好的性能,但容易陷入局部最优。IQGA-Viterbi能够搜索得到全局最优解,保证优化映射代价的同时保证服务路径时延,对每个时间周期内的物理资源精确分配,避免空闲VNF占用物理资源,降低了单位时间内的成本,因此具有最小映射代价。

(4)时延:表示数据流从源端到目的端之间所用的通信时间,其取值越小越好。实验中生成100~1000次请求强度,分别统计各算法的网络总时延,如图10所示。图10表示不同请求到达强度下各算法的网络总时延。随着请求到达强度的增加,可用的资源量减少,网络时延逐渐增加,曲线逐渐趋近平缓。随着资源瓶颈的出现,请求接受率逐渐降低,服务路径构建失败的次数增多。其中,IQGA-Viterbi算法的网络总时延最小,Greedy算法的网络总时延约为IQGA-Viterbi算法的3倍,Viterbi算法和QGA算法居中。原因在于Greedy在逻辑链路映射过程容易受到VNF节点映射的影响,造成承载VNF节点之间的路径较长,导致更大的网络时延。QGA将VNF节点和逻辑链路映射一步完成,通过迭代计算搜索时延最小的服务路径。Viterbi在给定HMM参数下,单位时间内时延的采样值近似瞬时时延,提高了时延优化的精确度。然而上述3种算法获取的底层网络参数不精确,因此网络总

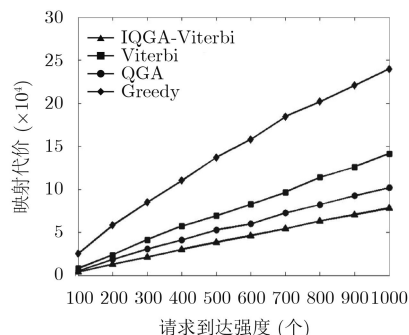


图9 不同请求强度下的服务链映射代价

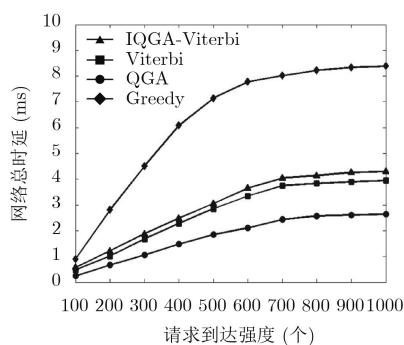


图10 不同请求强度下的时延均值比较

时延优化精确度相对较低。IQGA-Viterbi在每个事件周期内对网络参数进行训练，在参数训练过程中采用复制适应度最佳种群的机制，扩大空间搜索范围，快速地收敛于最优的网络底层参数。在求解服务路径过程中采用动态规划算法，该算法能够在满足资源约束条件下以递归的方式搜索时延最短路径，提高了时延优化精确度，因此具有更好的时延累积效果、最小网络时延。

## 5 结束语

本文提出基于IQGA-Viterbi学习算法的服务链优化部署方法能够优化映射代价的同时保证服务路径时延。首先利用改进的量子遗传算法优化训练HMM模型参数，其次借鉴动态规划方法求解隐含序列，从而在满足资源约束条件下为租户在线定制开销最小的服务路径。与其它部署算法对比，本文方法因为考虑了服务链部署的综合性能，在网络总时延、映射代价和请求接收率等性能指标上有良好的性能。目前，业界对智能VNF编排器的研究不断深入，下一步将针对可靠性感知下的服务功能链部署研究，以满足网络服务链可靠性需求和电信级的网络服务质量，建立以租户为中心的智能维护体系。

## 参考文献

- [1] BHAMARE D, JAIN R, SAMAKA M, *et al.* A survey on service function chaining[J]. *Journal of Network & Computer Applications*, 2016, 75(C): 138–155. doi: [10.1016/j.jnca.2016.09.001](https://doi.org/10.1016/j.jnca.2016.09.001).
- [2] MEDHAT A and TALEB T. Service function chaining in next generation networks: State of the art and research challenges[J]. *IEEE Communications Magazine*, 2017, 55(2): 216–223. doi: [10.1109/mcom.2016.1600219rp](https://doi.org/10.1109/mcom.2016.1600219rp).
- [3] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, *et al.* OpenFlow: Enabling innovation in campus networks[J]. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 69–75. doi: [10.1145/1355734](https://doi.org/10.1145/1355734).
- [4] HAN Bo, GOPALAKRISHNAN V, JI Lusheng, *et al.* Network function virtualization: Challenges and opportunities for innovations[J]. *IEEE Communications Magazine*, 2015, 53(2): 90–97. doi: [10.1109/mcom.2015.7045396](https://doi.org/10.1109/mcom.2015.7045396).
- [5] KIM S, PARK S, KIM Y, *et al.* VNF-EQ: Dynamic placement of virtual network functions for energy efficiency and QoS guarantee in NFV[J]. *Cluster Computing*, 2017, 20(3): 1–11. doi: [10.1007/s10586-017-1004-3](https://doi.org/10.1007/s10586-017-1004-3).
- [6] BHAMARE D, SAMAKA M, ERBAD A, *et al.* Optimal virtual network function placement in multi-cloud service chaining architecture[J]. *Computer Communications*, 2017, 102(C): 1–16. doi: [10.1016/j.comcom.2017.02.011](https://doi.org/10.1016/j.comcom.2017.02.011).
- [7] BARI M F, CHOWDHURY S R, AHMED R, *et al.* On orchestrating virtual network functions[C]. *International Conference on Network and Service Management*, Barcelona, Spain, 2015: 50–56. doi: [10.1109/cnsm.2015.7367338](https://doi.org/10.1109/cnsm.2015.7367338).
- [8] XIONG Gang, Hu Yuxiang, TIAN Le, *et al.* A virtual service placement approach based on improved quantum genetic algorithm[J]. *Information and Electronic Engineering Frontiers*, 2016, 17(7): 661–671. doi: [10.1631/fitee.1500494](https://doi.org/10.1631/fitee.1500494).
- [9] LUKOVSKI T, ROST M, and SCHMID S. It's a match!: Near-optimal and incremental middlebox deployment[J]. *ACM SIGCOMM Computer Communication Review*, 2016, 46(1): 30–36. doi: [10.1145/2875951.2875956](https://doi.org/10.1145/2875951.2875956).
- [10] MOENS H and TURCK F. VNF-P: A model for efficient placement of virtualized network functions[C]. *International Conference on Network and Service Management*, Beijing, China, 2014: 418–423. doi: [10.1109/cnsm.2014.701](https://doi.org/10.1109/cnsm.2014.701).
- [11] ZHANG Lijun, HERMANS H, and JANSEN D. Logic and model checking for Hidden Markov Models[C]. *International Conference on Formal Techniques for Networked and Distributed Systems*, Berlin, Germany, 2005: 98–112. doi: [10.1007/11562436\\_9](https://doi.org/10.1007/11562436_9).
- [12] ZHANG Zengyin, YUAN Changan, HU Jianjun, *et al.* HMM training method based on GEP and Baum-Welch algorithms[J]. *Computer Engineering & Design*, 2010, 31(9): 2027–2029.
- [13] XIONG Yan, CHEN Huanhuan, MIAO Fuyou, *et al.* A quantum genetic algorithm to solve combinatorial optimization problem[J]. *Acta Electronica Sinica*, 2004, 32(11): 1855–1858.
- [14] BOULOUTAS A, HART G W, and SCHWARTZ M. Two extensions of the Viterbi algorithm[J]. *IEEE Transactions on Information Theory*, 2002, 37(2): 430–436. doi: [10.1109/18.75270](https://doi.org/10.1109/18.75270).
- [15] ZHANG Lifang and ZHANG Xiping. Network traffic prediction based on BP neural networks optimized by quantum genetic algorithm[J]. *Computer Engineering & Science*, 2016, 10(3): 12–20.
- [16] ZHANG Ying, BEHESHTI N, BELIVEAU L, *et al.* StEERING: A software-defined networking for inline service chaining[C]. *IEEE International Conference on Network Protocols*, Raleigh, USA, 2014: 1–10. doi: [10.1109/icnp.2013.673](https://doi.org/10.1109/icnp.2013.673).
- [17] BASTA A, HOFFMANN K, HOFFMANN K, *et al.* Applying NFV and SDN to LTE mobile core gateways, the functions placement problem[C]. *The Workshop on All Things Cellular: Operations*, Chicago, USA, 2014: 33–38.



- doi: [10.1145/2627585.2627592](https://doi.org/10.1145/2627585.2627592).
- [18] 刘彩霞, 卢干强, 汤红波, 等. 一种基于Viterbi算法的虚拟网络功能自适应部署方法[J]. 电子与信息学报, 2016, 38(11): 2922–2930. doi: [10.11999/JEIT1650507](https://doi.org/10.11999/JEIT1650507).  
LIU Caixia, LU Ganqiang, TANG Hongbo, *et al.* Adaptive deployment method for virtualized network function based on Viterbi algorithm[J]. *Journal of Electronics & Information Technology*, 2016, 38(11): 2922–2930. doi: [10.11999/JEIT1650507](https://doi.org/10.11999/JEIT1650507).
- [19] ZEGURA E, CALVERT K, and BHATTACHARJEE S. How to model an Internet network[J]. *Proceedings of IEEE Infocom*, 1996, 2: 594–601. doi: [10.1109/infcom.1996.493353](https://doi.org/10.1109/infcom.1996.493353).
- [20] ORLOWSKI S, WESSALY R, PIORO M, *et al.* SNDlib 1.0-survivable network design library[J]. *Networks*, 2010, 55(3): 276–286. doi: [10.1002/net.20371](https://doi.org/10.1002/net.20371).
- [21] CLAYMAN S, MAINI E, GALIS A, *et al.* The dynamic placement of virtual network functions[C]. Network Operations and Management Symposium, Seoul, Korea, 2014: 1–9. doi: [10.1109/noms.2014.6838412](https://doi.org/10.1109/noms.2014.6838412).
- [22] 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. doi: [10.1016/j.comnet.2015.09.035](https://doi.org/10.1016/j.comnet.2015.09.035).
- 卢 昱: 男, 1960年生, 教授, 研究方向为下一代网络体系架构.  
刘益岑: 男, 1990年生, 硕士生, 研究方向为智能VNF编排技术、软件定义服务.  
李 玺: 男, 1982年生, 讲师, 研究方向为新型信息网络关键理论与技术.  
陈兴凯: 男, 1988年生, 博士生, 研究方向为未来网络体系架构关键技术.  
乔文欣: 女, 1992年生, 博士生, 研究方向为网络虚拟化技术可靠性.  
陈立云: 男, 1968年生, 教授, 研究方向为深度学习、人工智能.