一种面向运营成本优化的虚拟网络功能部署和路由分配策略

史久根 张 径* 徐 皓 王 继 孙 立

(合肥工业大学计算机与信息学院 合肥 230009)

摘 要：随着网络功能虚拟化(NFV)技术的发展，虚拟网络功能(VNF)可以通过服务功能链(SFC)的形式部署在如虚拟机的通用平台中，为管理带来灵活性。但是对于服务提供商来说，由于网络基础设施的复杂性和日益增长的服务需求，给VNF的部署带来了高昂的运营成本(OPEX)。针对此问题，该文提出一种面向OPEX优化的策略，旨在最小化OPEX中的激活、能耗和传输成本，得到VNF部署和路由分配优化方案。为此建立一种全新的混合整数线性规划(MILP)模型，并设计包括遗传算法(GA)在内的3种OPEX优化算法。仿真实验评估在不同资源配给下MILP和3种算法的OPEX及其性能，其中GA算法在节点资源配比60%以上时可以得到近似于MILP模型的解决方案。

关键词：网络功能虚拟化；虚拟网络功能部署；运营成本；服务功能链

中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2019)04-0973-07

DOI: 10.11999/JEIT180522

Joint Optimization of Virtualized Network Function Placement and Routing Allocation for Operational Expenditure

SHI Jiugen ZHANG Jing XU Hao WANG Ji SUN Li

(*School of Computer and Information, Hefei University of Technology, Hefei* 230009, *China*)

Abstract: With the development of Network Function Virtualization (NFV), Virtual Network Functions (VNFs) can be deployed in a common platform such as virtual machines in the form of Service Function Chaining (SFC), providing flexibility for management. However for service providers, these come with high OPerational EXpenditure (OPEX), due to the complexity of the network infrastructure and the growing demand for services. To solve this problem, a strategy for OPEX optimization is proposed, which aims to minimize the startup cost, energy consumption, transmission cost and obtain VNF deployment and routing allocation optimization scheme. The VNF deployment problem as a new Mixed Integer Linear Programming (MILP) model is formulated, and three OPEX optimization algorithms are designed including Genetic Algorithm (GA). The OPEX of MILP model and optimization algorithms are compared under different resource allocation constraints. The calculation result shows that the GA can obtain the near-optimal solutions when node resource ratio is more than 60%.

Key words: Network Function Virtualization (NFV); Virtual Network Function (VNF) placement; OPerational EXpenditure (OPEX); Service Function Chaining (SFC)

1 引言

近些年来，网络中的数据、应用越来越多，关于网络的性能、安全、管理等问题就逐渐成为人们关注的焦点。网络服务提供商(Internet Service Provider, ISP)为用户提供各种不同的虚拟网络功能(Virtualized Network Function, VNF)，如防火墙、NAT、负载均衡机制等，来解决上述问题以满足日益增长的服务需求[1]。研究表明现在VNF的数量与网络中转发设备的数量相当[2]。

由于网络功能的重要性，如何优化VNF部署中的运营成本(OPerational EXpenditure, OPEX)也受到了关注。然而，解决VNF部署问题仍然是一个具有挑战性的任务[2,3]。一方面，VNF部署的位置和数量会影响OPEX成本以及网络性能，以下几点在优化OPEX时需要考虑的：(1)实例化VNF到虚拟机的激活成本；(2)运行所有VNF的能耗成

收稿日期：2018-05-28；改回日期：2018-11-30；网络出版：2018-12-10
*通信作者： 张径 zj0910@mail.hfut.edu.cn
基金项目：国家重大科学仪器设备开发专项(2013YQ030595)
Foundation Item: The National Major Scientifc Instruments Development Project (2013YQ030595)

本；(3)请求路由的传输成本；(4)部署还需要满足一系列的约束条件，比如每个请求的路由必须按照指定顺序经过服务功能链。另一方面，当前研究往往忽略了网络功能存在资源共享能力有限的问题。随着网络请求和流量的日益增加，单个节点的处理能力满足不了现有的需求，需要激活更多的节点共同处理，增加了部署的难度。

最早基于虚拟机架构的VNF部署可以追溯到Flowstream方案[4]。当前研究主要集中在网络功能服务功能链上。文献[5,6]从部署成本角度出发，前者通过建立ILP模型并使用贪心算法最小化流量负载成本和虚拟深度包检测服务的放置成本，但是局限于单服务功能链场景；后者进一步优化多种VNF实例的部署成本，并提出非零和博弈算法，然而忽略了VNF的有序问题。在优化资源利用率上，Luizelli等人[7]对服务功能链拼接后通过2路搜索动态调整模型，但是模型的服务功能链不易扩展组合。Bhamare等人[8]从网络时延的角度考虑VNF的部署，提出在多云场景中最小化云间流量和响应时间，保证网络服务质量。

文献[9–11]在VNF部署中都使用了遗传算法。Cao等人[9]提出混合NFV环境的概念，并设计了4种遗传算法用于最小化带宽开销和最大化链路利用率。Carpio等人[10]利用复制的方式部署VNF保证网络的负载均衡，并为大型网络设计了遗传算法。Rankothge等人[11]通过遗传算法在云数据中心动态部署VNF，最大限度地利用网络资源。

本文的研究与VNF-OP[12]相似，从优化运营成本的角度考虑VNF部署问题，但是后者没有深入分析不同资源配给下的OPEX情况。

2 VNF部署问题

2.1 部署架构

基于虚拟机平台实例化VNF是NFV架构的一种部署方式[2,3]，具有良好的隔离性和灵活性。如图1所示，每个虚拟机(Virtualized Machine, VM)只能实例化一种类型的VNF，并且新虚拟机的开辟需要物理节点(Physical Machine, PM)分配一定数量的资源，如CPU和存储资源，所以尽量将请求的VNF分配到已有的虚拟机中可以节约成本。

然而，每个虚拟机所能实例化的VNF数量取决于虚拟机所能承受的吞吐量[3]。同时由于资源有限，每个物理节点只能部署一定数量的虚拟机，但是每个虚拟机可以根据需要激活，休眠或迁移[13]。我们把虚拟机与吞吐容量的限制作为VNF部署模型的一个基本特征。

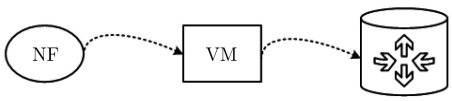


图 1 网络功能部署在物理节点的虚拟机中

2.2 OPEX成本

部署VNF的OPEX有以下3种主要成本[12]：

(1)激活成本：在部署VNF之前，每个物理节点都需要完成前期准备工作，比如不同类型VM的预配置。部署VNF到某个VM，意味着需要激活相应的物理节点。根据拓扑中各节点度的大小将物理节点分为核心节点和边缘节点两类。一般情况下激活核心节点的成本要小于边缘节点。

(2)能耗成本：根据文献[14]的结论，服务器的能耗开销与所消耗的资源成正比。为此规定VNF能耗开销主要由活跃PM和活跃VM产生。如果PM或VM没有部署任何的VNF，则处于休眠状态不考虑能耗开销。

(3)传输成本：主要考虑请求的流量和路由从源点到终点所经过的链路数，链路的转发也需要消耗一定的传输资源。实际中能耗成本与传输成本呈负相关关系。当减少活跃节点以节约成本时会造成网络的平均传输成本增加。

综上，部署VNF时需要考虑在不同节点资源配比下最小化OPEX成本，通过发现(a)最少的VNF部署数量，(b)最佳的部署位置，以及(c)最优的请求路由，得到VNF部署和路由分配方案。

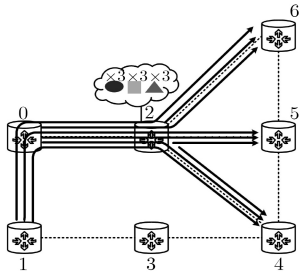
2.3 服务功能链(SFC)

由于某些应用的要求，请求流需要经过一组指定顺序的网络功能，我们把组成有序的网络功能集称之为服务功能链(Service Function Chaining, SFC)[3]。比如，http应用由于考虑安全性，需要请求流经过的SFC：firewall → IDS → proxy。因此，在部署VNF时需要考虑SFC的要求[2]：(1)有序性：请求流必须经过指定顺序的SFC; (2)有效性：请求流必须经过SFC中所有的VNF。

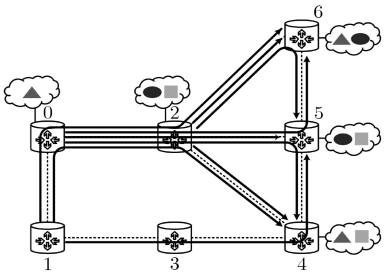
通过下面的例子，可以看到不同的VNF部署方案对OPEX的影响。如图2(a)所示，有9个端到端的请求，每个请求都规定好了流量大小，并随机生成需要经过的SFC次序。在拓扑中设置节点2和节点5为核心节点，假设每个VNF类型的虚拟机吞吐容量都为5 Gbps。分别考虑在节点虚拟机资源充足和受限两种情况下最小化OPEX。通过使用第3节描述的模型得到最优的VNF部署和路由方案。我们发现：在节点资源充足的方案1中，如图2(b)，可以得到最少的激活成本和能耗成本，但是传输成本会上升。所以为避免过长的路径，可以在模型中

请求	流量(Gbps)	源点　SFC次序　终点
1	1	⓪ ▲ ● ■ ④
2	1	⓪ ▲ ● ■ ⑤
3	2	⓪ ▲ ■ ● ⑥
4	1	① ● ■ ▲ ④
5	3	① ■ ▲ ● ⑤
6	3	① ■ ● ▲ ⑥
7	1	② ● ▲ ■ ④
8	1	② ● ■ ▲ ⑤
9	1	② ● ■ ▲ ⑥

(a)请求和SFC次序



(b)方案1



(c)方案2

图 2 节点虚拟机资源充足和受限2种情况的解决方案

加入延迟约束。节点资源受限的方案2(节点VM容量为2，其小于SFC长度)，如图2(c)，虽然可以通过最短路径转发降低传输成本，但是活跃节点的增加提高了能耗成本，甚至可能会出现违反约束条件的情况。

3　MILP模型

针对上述问题，通过控制请求路由来实现VNF部署的优化。本文结合多商品流模型[15]，提出了VNF部署问题中面向OPEX优化的MILP模型。

首先，定义如下：(1)将网络拓扑抽象成无向图$G(E,V)$，假设图中每个节点V都可以部署一定数量的虚拟机，每个虚拟机只能实例化一种类型的VNF。(2)D为一组端到端的请求集合，每个请求$k\in D$的流从起点s_k到目的点t_k, $s_k\neq t_k$，流量$r_k>0$；并且指明了请求的SFC次序M_k，其中每个VNF的次序$m_n^k\geq 0$, $n\in N$, N为VNF的种类集合。(3)为了避免网络回路，每条流在图中是沿简单无环路径转发。

接着，定义模型的约束条件。式(1)是流守恒约束。变量$f_{u,v}^k\in\{0,1\}$表示流$k\in D$是否经过链路$(u,v)\in E$，除源点和终点外，图中每个节点的入流和出流数量需要一致。

$$\sum_{(u,v)\in E} f_{u,v}^k-\sum_{(v,u)\in E} f_{v,u}^k=\begin{cases}1, & u=s_k,\\ -1, & u=t_k,\\ 0, & \text{其它}\end{cases}\quad \forall k\in D,\ u\in V \tag{1}$$

式(2)是链路的带宽约束。每个链路上的流量之和不能超过该链路的带宽容量$B_{u,v}$。

$$\sum_{k\in D} f_{u,v}^k r_k\leq B_{u,v},\ \forall (u,v)\in E \tag{2}$$

式(3)保证了每个请求的SFC都必须被部署到网络中$x_{i,n}^k\in\{0,1\}$，并且每个VNF只需要一个节点的虚拟机来实例化中。

$$\sum_{i\in V} x_{i,n}^k=1,\ \forall k\in D, n\in N: m_n^k>0 \tag{3}$$

同时，为了保证每个请求的SFC只能沿着路由上的节点部署，需要式(4)和式(5)的约束条件。

$$\sum_{(u,v)\in E} f_{u,v}^k\geq x_{i,n}^k,\forall k\in D, i\in V, n\in N \tag{4}$$

$$\sum_{i\in V} x_{i,n}^k<m_n^k,\forall k\in D, n\in N \tag{5}$$

式(6)—式(8)用于规定流依次经过节点的序号δ_i^k，对于流未经过的节点，其序号设为0。同时式(7)的约束避免流转发过程中出现回路。

$$\delta_v^k>\delta_u^k+f_{u,v}^k-|V|(1-f_{u,v}^k),\quad \forall k\in D,(u,v)\in E \tag{6}$$

$$\sum_{(u,v)\in E}|V|f_{u,v}^k\leq\delta_u^k,\forall k\in D,u\in V \tag{7}$$

$$\delta_i^k=1,\forall k\in D, i\in V\colon i=s_k \tag{8}$$

式(9)规定了每个SFC中VNF部署的功能序号λ_n^k, 0表示未部署。

$$\lambda_n^k=x_{i,n}^k\delta_i^k,\forall k\in D,i\in V,n\in N \tag{9}$$

由于式(9)是非线性约束，需要转化成线性约束。

最终SFC部署的功能序号必须满足请求的功能次序要求。即式(10)的功能次序一致约束。

$$\lambda_i^k\leq\lambda_j^k,\forall k\in D,i\in N: m_i^k+1=m_j^k \tag{10}$$

其次，每个节点还要满足各类型虚拟机吞吐容量T_i^n和虚拟机数量C_i的限制，分别满足式(11)和式(12)。

$$\sum_{k\in D} x_{i,n}^k r_k/T_i^n\leq\eta_i^n,\forall i\in V,n\in N \tag{11}$$

$$\sum_{n\in N}\eta_i^n\leq z_iC_i,\forall i\in V, z_i\in\{0,1\} \tag{12}$$

为了真实反映VNF部署中有限资源对OPEX的影响，本文设置资源配比参数p_i来调节节点中的虚拟机数量，$p_i=C_i/C_{\max}$, $0<p_i\leq 1$。

此外，每条请求流也需要满足式(13)的时延要求l_k。

$$\sum_{(u,v)\in E} f_{u,v}^{k} \le l_k, \forall k \in D \tag{13}$$

本文的优化目标是最小化OPEX。具体的成本阐述如下：

(1)激活成本：a_i表示各节点的激活代价。

$$\mathbb{Z} = \sum_{i\in V} a_i z_i \tag{14}$$

(2)能耗成本：其中e^{PM}和e^{VM}分别是活跃节点和活跃虚拟机所需的能耗开销。

$$\mathbb{N} = \sum_{i\in V}\left(e^{\mathrm{PM}} p_i z_i + \sum_{n\in N} e^{\mathrm{VM}} \eta_i^n\right) \tag{15}$$

(3)传输成本：σ表示链路传输1 Mbit流量所需要的费用。

$$\mathbb{R} = \sum_{k\in D}\sum_{(u,v)\in E} f_{u,v}^{k} r_k \sigma \tag{16}$$

综上，我们建立起优化OPEX的VNF部署模型：

$$\min\ \alpha\mathbb{Z}+\beta\mathbb{N}+\mu\mathbb{R},\ \text{s.t. 式(1)} - \text{式(13)}$$

其中，α, β, μ是用于调节目标函数的影响权重。比如，调整α使传输成本对其他目标函数的影响控制在1%的范围内。保证在最小化激活和能耗成本的同时降低传输成本。

4 OPEX优化算法

由于VNF部署问题被证明是NP-hard问题[3]，能够找到精确的解决方案具有极大的挑战性。一方面，随着问题的规模，比如请求数量、拓扑大小的增加，求解时间会呈指数型增长。另一方面，一旦请求或节点发生变化，就需要再次求解，这无疑增加了运算成本。为此，本文设计3种面向大型网络的OPEX优化算法。

4.1 启发式算法

首先，本文提出2种基于贪心思想的启发式算法(HEU和HEU$^+$)。

(1)HEU：如表1的算法1和表2的算法2所示，在保证每个请求$k\in D$的SFC次序要求下，计算出可行的SFC部署方案并规划路由。其中，SFC中每个VNF的位置节点保存在SFC_k(preSFC_k, $\mathrm{finalSFC}_k$)，路由信息保存在R_k，每个节点$i\in V$实例化的虚拟机种类和吞吐量记录在VM_i。部署的过程如下：(1)算法1依次寻找请求的SFC部署方案，对每个请求的SFC检查在需要新开辟t个虚拟机的情况下是否可以部署到该请求的路径集中。(2)首先从$t=0$开始迭代，直到SFC中每个VNF都需要新建虚拟机为止。(3)每次迭代都以最短路径的顺序判断此路径是否满足部署SFC的条件，如算法2所示。

表 1 算法1：发现可行的SFC部署方案

输入：$G(V,E)$, N, D, M_k, l_k
输出：VM_i, $\mathrm{finalSFC}_k$ by a tuple $\langle \mathrm{order}\in N^*, n\in N, i\in V\rangle$, R_k

(1) **while** 未分配的请求$d_k\in D$ **do**
(2) 获取从s_k到t_k时延l_k内的简单路径集合P_k，以最短路径排序;
(3) t=0; //t为SFC中需要初始化的虚拟机个数
(4) **while** $t\le |M_k|$ **and** $\mathrm{Path}_k\in P_k$ **do**
(5) 从N中选出所有t-子集的组合序列initVMs;
(6) 调用算法2得到请求k的预部署方案preSFC_k;
(7) **if** $|\mathrm{preSFC}_k| == |M_k|$ **the**
(8) $\mathrm{finalSFC}_k \leftarrow \mathrm{preSFC}_k$, $R_k \leftarrow \mathrm{Path}_k$; **break**;
(9) **end if**
(10) **end while**
(11) **end while**
(12) **return** VM_i, $\mathrm{finalSFC}_k$, R_k;

表 2 算法2：请求SFC的预部署

输入：initVMs, VM_i, C_i, T_i^n
输出：preSFC_k by a tuple $\langle \mathrm{order}\in N^*, n\in N, i\in V\rangle$

(1) $\mathrm{preSFC}\leftarrow\varnothing$;\; order $\leftarrow$ 1;//order为SFC的次序索引
(2) **while** $i\in V$ in Path_k **and** order $\le |M_k|$ **do**
(3) $n=\{n|n\in N: M_k=\mathrm{order}\}$//找到对应索引的VNF
(4) **if** $n\in \mathrm{VM}_i$ **and** $r_k+\mathrm{VM}_i$的吞吐容量 $\le T_i^n$ **then**
(5) $\mathrm{preSFC}_k\cup\{\langle \mathrm{order}, n, i\rangle\}$; order++; 返回步骤(2);
(6) **end if**
(7) **if** $n\notin \mathrm{VM}_i$ **and** $n\in$ initVMs **and** $|\mathrm{VM}_i| < C_i$ **then**
(8) $\mathrm{VM}_i\cup\{n\}$; $\mathrm{preSFC}_k\cup\{\langle \mathrm{order}, n, i\rangle\}$; order++; 返回步骤(2);
(9) **end if**
(10) **end while**
(11) **return** preSFC_k;

(2)HEU$^+$：为进一步减少OPEX成本，通过HEU算法得到可行的SFC部署方案后，表3的算法3尝试关闭那些虚拟机数量较少的活跃节点，将VNF实例迁移到其他虚拟机中。减少活跃节点的数量以节约成本。由于算法采用贪心思想得到部署方案后停止迭代，所以算法的平均时间远小于最坏情况。

表 3 算法3：休眠活跃节点

(1) **while** 活跃节点$i\in V$能被关闭 **do**
(2) 关闭虚拟机数量最少且在图G中的度最小的节点i;
(3) 调用**算法1**重新计算SFC部署方案;
(4) 如果没有可行的方案，则回溯上次方案;
(5) **end while**

4.2 遗传算法

由于上述的启发式算法容易陷入局部最优，为了最大限度的优化OPEX，本文为VNF部署问题设计了一种遗传算法(GA)。其关键部分如下：

(1)编码与种群初始化：采用二进制编码，对拓扑中的每个物理节点$i \in V$进行编码，以表示节点是否处于休眠状态。在初始化种群时，假设所有节点都处于正常状态可以被激活，对于那些边缘节点，在下次迭代中有更高的几率被休眠。

(2)适应度函数与选择：交叉或变异染色体可能会使部署方案不能满足MILP的约束条件。为此对于每代的染色体，根据HEU算法找到可行的部署方案。并且使用MILP的优化目标作为适应度函数，通过轮盘赌方法选择出子代。

(3)迭代次数与变异概率：在每次进化中使用单点交叉。设置GA算法的迭代次数为100，变异概率为30%。

5 仿真实验

为了保证模型的可行性和算法的有效性，仿真实验建立在真实的网络拓扑中，数据来自SNDlib[16]。其中，MILP模型采用OPL语言建模，通过CPLEX 12.7求解。OPEX优化算法使用JUNG框架(Java Universal Network/Graph framework)搭建。实验运行在2个Intel Xeon E5-2624 v4处理器和128 GB内存的机器上。

在展示结果之前，首先简要地介绍实验的参数设置和评估指标。实验源码：http://dwz.cn/ouGd6P。

5.1 实验设置

(1)拓扑设置：主要关注网络服务提供商的VNF部署，为此设置两种类型的网络拓扑，(a)小型网络pdh(V=11, E=34, $|D|$=18)，用于比较MILP和3 种OPEX优化算法；(b)较大规模和流量的城域网络newyork(V=16, E=49, $|D|$=56)，用于检测GA算法的性能。为简化实验，所有请求都来自网络中的边缘节点。

(2)VM设置：根据现有数据推断出PM和VM分别在消耗相同资源(1核CPU为单位)情况下的能耗开销[12](PM: 80.5 W, VM: 165.9 W)，以及各类VNF在单位VM中的吞吐容量[17](Firewall: 600 Mbps, IDS: 400 Mbps, Proxy: 300 Mbps)。并且假设网络中的每个节点可以部署虚拟机的最大数量C_i都相同。

(3)SFC设置：实验中为了避免资源分配不均对实验指标的干扰，统一SFC的长度和次序，设置长度为3的SFC: firewall → IDS → proxy。

5.2 评估指标

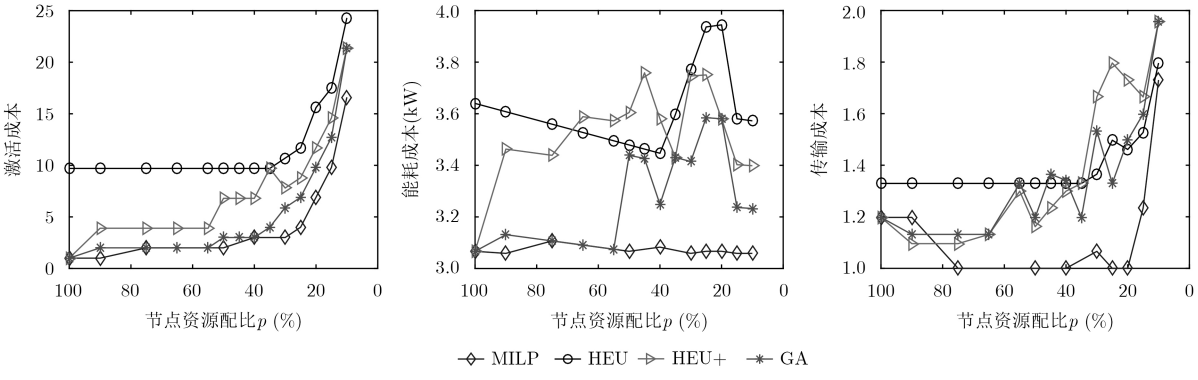
为了统一调整不同网络中节点的虚拟机资源数量，设置$C_i = p \times |\mathrm{VM}|, 0 < p \leq 1$，其中$|\mathrm{VM}|$为资源不受限方案中活跃VM的最小数量。为量化OPEX，激活成本用单位活跃PM数量表示，并假设边缘节点的激活成本是核心节点的2倍；能耗成本用真实能耗开销表示；传输成本用当前路径与最短路径跳数的时延比表示。

5.3 性能分析

(1)MILP最优解与3种OPEX优化算法的关系：首先，比较MILP和各类算法在部署VNF时的OPEX差异和变化趋势。为此选择pdh拓扑，保持请求集流量不变，并设置最大时延为5跳。图3分别展示了MILP, HEU, HEU$^+$和GA算法的OPEX。从图3可以看出不管何种成本，GA算法都优于另外2种算法，并且资源配比在60%以上时，接近MILP最优解；但是随着节点资源的限制，求解难度的增加导致部署方案开始出现波动，与MILP相比存在一定的差距。另外，由于HEU算法只采用最短路径策略部署VNF，导致其成本在总体上效果最差。

接着，我们具体分析在不同资源配给下OPEX各成本的变化：

(a)激活成本：随着节点资源的限制，单个节点可以部署的虚拟机数量随之减少，网络需要开辟



(a)激活成本　(b)能耗成本　(c)传输成本

图 3 pdh网络在不同资源配给下的OPEX

更多的活跃节点来保证所有VNF均被实例化，导致其激活成本不断增加，直到资源减少到p=10%时无法在约束条件内得到部署方案(图3(a))。

(b)能耗成本：由于能耗开销与所消耗的服务器资源成正比。MILP可以得到最少资源消耗的部署方案，所以能耗成本保持在稳定的最小开销。然而其他3种OPEX算法在资源限制下，实例化的虚拟机数量存在波动，导致需要消耗额外的能耗，但是相比之下GA算法的能耗成本波动最小(图3(b))。

(c)传输成本：不同资源配给下的传输成本也存在差异(图3(c))。首先，随着节点资源配比的减少，传输成本会随之下降，因为会开辟更多的虚拟机使VNF可以就近实例化在较短路径中；但是资源减少到一定程度，由于有些请求的SFC在最短路径上无法部署，不得不延长传输路径导致成本上升。值得注意的是当p=10%时，由于节点虚拟机容量为2小于SFC链长度，部署难度增加会使传输成本上升更严重。

虽然MILP在小规模网络中可以得到更优秀的OPEX，但是求解所花费的平均时间要远大于GA算法需要的时间。如图4所示，其求解时间会随节点资源配比的限制而呈指数型增长，同时受网络规模和请求数量的限制。

(2)GA算法的性能分析：为进一步分析GA算法在动态部署中的性能，本文在Newyork网络中随机生成1个月的请求流量(其中为基准流量分配p=100%的节点资源)，如图5所示。对于每天的流量请求本文分别使用GA和HEU算法得到VNF部署方案，图6分别展示了1个月的OPEX结果。GA算法相比于HEU算法可以降低4倍的激活成本，并且可以得到更低的能耗成本，却是以牺牲传输成本为代价的。但是对于服务提供商来说激活节点所需的成本更高。

对于GA算法1个月的OPEX变化趋势来说，激

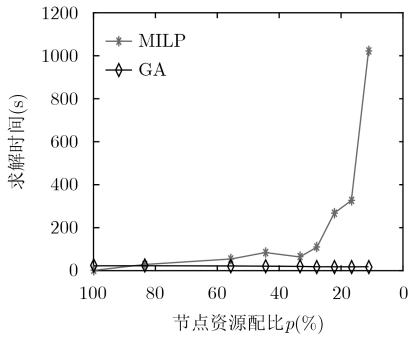


图 4 pdh网络在不同资源配比下的求解时间

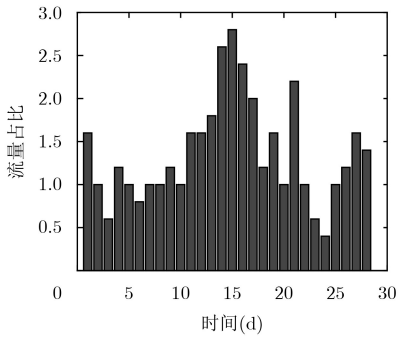
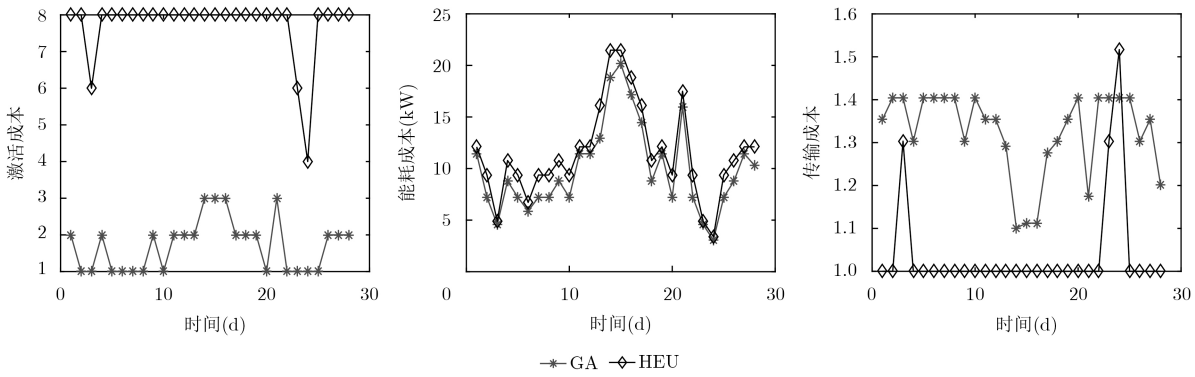


图 5 Newyork网络在1个月中的流量分布

活和能耗成本随着流量的变化而变化，但是传输成本却呈相反的变化趋势，这是由于两者在模型中呈现负相关关系。本文的策略是在保证最小化激活和能耗成本的同时尽量降低传输成本。另外，当请求流量小于基准流量时(流量占比小于1)，由于节点有充足的资源可供VNF部署，能耗和传输成本趋于稳定，但是能耗开销依然随之变化，这是因为能耗成本与流量对虚拟机资源的消耗相关。

6 结束语

降低OPEX对VNF部署问题具有重要的影响，可以节约成本，避免不必要的资源浪费。本文提出一种在NFV网络中面向OPEX优化的VNF部署和路由分配策略，设计可组合服务功能链的MILP数



(a)激活成本 (b)能耗成本 (c)传输成本

图 6 Newyork网络在1个月中的OPEX

学模型，并提出3种OPEX优化算法。其中GA算法更接近MILP最优解。最后，仿真实验验证了其有效性。未来进一步提高解决方案的可伸缩性以应对突发流量，研究实时复杂条件下的VNF部署问题。

参 考 文 献

[1] HAN B, GOPALAKRISHNAN V, JI L, *et al*. Network function virtualization: Challenges and opportunities for innovations[J]. *IEEE Communications Magazine*, 2015, 53(2): 90–97. doi: 10.1109/mcom.2015.7045396.

[2] XIN Li and CHEN Qian. A survey of network function placement[C]. Consumer Communications & Networking Conference, Las Vegas, USA, 2016: 948–953. doi: 10.1109/ccnc.2016.7444915.

[3] HERRERA J G and BOTERO J F. Resource allocation in NFV: A comprehensive survey[J]. *IEEE Transactions on Network and Service Management*, 2016, 13(3): 518–532. doi: 10.1109/TNSM.2016.2598420.

[4] GREENHALGH A, HUICI F, HOERDT M, *et al*. Flow processing and the rise of commodity network hardware[J]. *ACM SIGCOMM Computer Communication Review*, 2009, 39(2): 20–26. doi: 10.1145/1517480.1517484.

[5] 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. doi: 10.1109/netsoft.2015.7116121.

[6] LIN Tachun, ZHOU Zhili, TORNATORE M, *et al*. Demand-aware network function placement[J]. *Journal of Lightwave Technology*, 2016, 34(11): 2590–2600. doi: 10.1109/JLT.2016.2535401.

[7] 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]. IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, Canada, 2015: 98–106. doi: 10.1109/INM.2015.7140281.

[8] BHAMARE D, SAMAKA M, ERBAD A, *et al*. Optimal virtual network function placement in multi-cloud service function chaining architecture[J]. *Computer Communications*, 2017, 102: 1–16. doi: 10.1016/j.comcom.2017.02.011.

[9] CAO Jiuyue, ZHANG Yan, AN Wei, *et al*. VNF placement in hybrid NFV environment: Modeling and genetic algorithms[C]. IEEE International Conference on Parallel and Distributed Systems, Wuhan, China, 2017: 769–777. doi: 10.1109/ICPADS.2016.0105.

[10] CARPIO F, DHAHRI S, and JUKAN A. VNF Placement with replication for load balancing in NFV networks[C]. IEEE International Conference on Communications, Paris, France, 2017: 1–6. doi: 10.1109/ICC.2017.7996515.

[11] RANKOTHGE W, MA Jiefei, LE Franck, *et al*. Towards making network function virtualization a cloud computing service[C]. IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, Canada, 2015: 89–97. doi: 10.1109/INM.2015.7140280.

[12] BARI F, CHOWDHURY S R, AHMED R, *et al*. Orchestrating Virtualized Network Functions[J]. *IEEE Transactions on Network & Service Management*, 2016, 13(4): 725–739. doi: 10.1109/TNSM.2016.2569020.

[13] 史久根, 许辉亮, 陆立鹏. 软件定义网络中数据中心虚拟机迁移序列问题的研究[J]. 电子与信息学报, 2017, 39(5): 1193–1199. doi: 10.11999/JEIT160792.
SHI Jiugen, XU Huiliang, and LU Lipeng. Research on the migration queue of data center's virtual machine in software defined networks[J]. *Journal of Electronics & Information Technology*, 2017, 39(5): 1193–1199. doi: 10.11999/JEIT160792.

[14] MOUSTAFA N, MASHALY M, and ASHOUR M. Modeling and simulation of energy-efficient cloud data centers[C]. International Conference on Engineering and Technology, Cairo, Egypt, 2014: 1–5. doi: 10.1109/ICENGTECHNOL.2014.7016745.

[15] ASSAD A A. Multicommodity network flows—A survey[J]. *Networks*, 1978, 8(1): 37–91. doi: 10.1002/net.3230080107.

[16] ORLOWSKI S, WESSÄLY R, PIÓRO M, *et al*. SNDlib 1.0—Survivable network design library[J]. *Networks*, 2010, 55(3): 276–286. doi: 10.1002/net.20371.

[17] MARTINS J, AHMED M, RAICIU C, *et al*. ClickOS and the art of network function virtualization[C]. 11th USENIX Symposium on Networked Systems Design and Implementation, Seattle, USA, 2014: 459–473.

史久根：男，1963年生，副教授，研究方向为嵌入式系统、计算机网络和软件定义网络.

张　径：男，1993年生，硕士生，研究方向为软件定义网络、网络功能虚拟化.

徐　皓：男，1994年生，硕士生，研究方向为软件定义网络、控制器部署.

王　继：男，1993年生，硕士生，研究方向为软件定义网络、网络功能虚拟化.

孙　立：男，1993年生，硕士生，研究方向为软件定义网络、网络功能虚拟化.