

## 面向 SDN 的虚拟网络映射算法研究

王健<sup>1</sup>, 赵国生<sup>2</sup>, 李志新<sup>1</sup>

(1. 哈尔滨理工大学计算机科学与技术学院, 黑龙江 哈尔滨 150080;

2. 哈尔滨师范大学计算机科学与信息工程学院, 黑龙江 哈尔滨 150025)

**摘要:** 针对虚拟网络的映射问题, 提出一种在 SDN 环境下基于蚁群混合遗传算法的虚拟网络映射方法。通过建立线性规划模型, 将映射过程分为节点映射和链路映射, 首先基于该融合算法将虚拟节点映射到物理节点上, 再利用最短路径算法将虚拟链路映射到物理链路上, 以此提高虚拟网络请求的接受率。仿真实验结果表明, 与 D-ViNE、R-ViNE 以及 RW-BFS 相比, 蚁群混合遗传算法能有效地将虚拟网络请求的接受率提高 10%, 并较好地改善了节点和链路的平均利用率以及映射的收益成本比值。

**关键词:** SDN; 网络虚拟化; 蚁群混合遗传算法; 虚拟网络映射

**中图分类号:** TP393

**文献标识码:** A

## Research on mapping algorithm of virtual network oriented to SDN

WANG Jian<sup>1</sup>, ZHAO Guo-sheng<sup>2</sup>, LI Zhi-xin<sup>1</sup>

(1. School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China;

2. School of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China)

**Abstract:** In order to solve the problem of virtual network mapping, a mapping method based on ant colony hybrid genetic algorithm was put forward under SDN environment, which established a linear programming model for virtual network mapping, and divided the mapping process into node mapping and link mapping. Firstly, the fusion algorithm was adopted, in which virtual nodes were mapped to physical nodes. Then the shortest path algorithm was used to map the virtual link to a physical link. On this basis, the acceptance ratio of virtual network requests can be improved. Simulation experiment results show that acceptance rate of virtual network requests can be increased by 10% efficiently using the ant colony hybrid genetic algorithm, compared with existing mapping algorithms D-ViNE, RW-BFS and R-ViNE. Further more, proposed method can greatly improve the average utilization rate of nodes and links and the ratio of the mapping income to cost.

**Key words:** SDN, network virtualization, ant colony hybrid genetic algorithm, virtual network mapping

### 1 引言

众所周知, 互联网已经成为当今社会通信领域、信息交流、经济和商业运营以及多媒体业务等

的基础设施之一。当前 Internet 之所以能够如此迅速且深入地融入人们的日常生活和工作中, 现有的网络架构功不可没。但是, 随着业务规模的不断延伸和扩展以及业务类型的不断丰富, 各种新型应用

收稿日期: 2017-01-06; 修回日期: 2017-07-02

基金项目: 国家自然科学基金资助项目 (No.61403109, No.61202458); 高等学校博士学科点专项科研基金资助项目 (No.20112303120007); 黑龙江省自然科学基金资助项目 (No.F2017021); 哈尔滨市科技创新人才研究专项资金资助项目 (No.2016RAQXJ036)

**Foundation Items:** The National Natural Science Foundation of China (No.61403109, No.61202458), The Specialized Research Fund for the Doctoral Program of Higher Education of China (No.20112303120007), The Natural Science Foundation of Heilongjiang Province (No.F2017021), The Specialized Research Fund for Scientific and Technological Innovation Talents of Harbin (No.2016RAQXJ036)

层出不穷，不同应用在服务质量、安全性以及可扩展性等方面对网络环境提出了不同的需求，而现有的网络架构和协议很难满足新型应用的发展需求，在一定程度上出现了僵化现象<sup>[1]</sup>。网络虚拟化作为构建新型互联网架构的新一代技术，它的出现可以有效地解决当前 Internet 的这一瓶颈<sup>[2~4]</sup>，但是，由于虚拟网络环境存在多层次、不确定等特性，使在现有环境下实现网络虚拟化存在很大的局限性，于是，一种新兴的网络体系架构——SDN<sup>[5]</sup>应运而生。它将控制与转发分离，并且实现了可编程地集中控制，支持网络虚拟化<sup>[6]</sup>，二者相辅相成，是 2 种具有高度相关性的技术的有效结合，可以很好地满足未来网络对于组网的灵活性、网络的集中管理等需求<sup>[7,8]</sup>。

虚拟网络的映射问题作为网络虚拟化技术实现过程的一项重要挑战，成为众多学者研究的热点，目前，这一问题已被证明为 NP 难问题<sup>[9]</sup>。为了实现网络虚拟化，就必须解决虚拟网络映射这一难点。针对这种情况，已有研究主要基于以下 4 种思路进行求解。1) 将虚拟网络映射建立数学模型进行求解。代表性研究成果主要由 Cheng 等<sup>[10]</sup>提出了一种基于拓扑感知的节点排序的虚拟网络映射算法，该算法基于物理网络的资源和拓扑结构，应用马尔可夫随机游走模型随机安排网络节点，这种利用拓扑感知来评估节点资源的机制，虽然能够使资源得到很好的评估和表征，但是却没有考虑底层物理网络中的节点资源和链路资源的区别。Chowdhury 等<sup>[11]</sup>使用确定性和随机舍入技术分别设计了 D-ViNE 和 R-ViNE 2 种虚拟网络映射算法，将映射问题用混合整数规划模型表示，然后通过整数规划松弛算法对节点映射和链路映射 2 步进行协调，并且添加了对负载均衡因子的表述，在规模较小的虚拟网络映射中获得最优解，但是该算法的时间复杂度过高，导致算法的求解效率不高，在较大规模的虚拟网络动态映射中难以应用。2) 将虚拟网络映射中的约束条件进行简化进而求解。Liu 等<sup>[12]</sup>提出了一种基于临近原则的虚拟网络映射算法，该算法在原有的节点映射这一步中只考虑性能因素的基础上引入距离因素，但是算法没有充分考虑实际的虚拟网络映射中的节点映射过程中对于节点的约束及限制，使映射结果过于理想化，脱离了实际的虚拟网络请求。3) 将规模相对较大的虚拟网络分解为多个小规模的网络进行映射。Dietrich 等<sup>[13]</sup>提出了一种基于底层网络信息的多个基础设施提

供商 (InP) 场景下的映射算法，该算法通过对流量矩阵的分割，将虚拟网络划分为若干子虚拟网络，进而由不同 InP 对各子虚拟网络进行映射，但该算法忽略任务处理的公平性，而且通常无法取得最优解。Peng 等<sup>[14]</sup>为了提高虚拟网络映射算法的效率，提出一种基于图的邻接分割算法，将大规模虚拟网络分解为多个邻接的星型结构，使底层物理网络能更好地支持虚拟网络映射，但是，这种方法没有充分考虑底层物理节点的负载均衡，导致底层的物理资源没有得到高效的利用。4) 用启发式算法<sup>[15]</sup>进行求解，快速得到虚拟网络映射的近似最优解或可行解。Lezama 等<sup>[16]</sup>提出了一种基于差分进化的算法，对光互联网环境下的可生存性的虚拟拓扑映射问题进行求解，该算法充分考虑底层物理拓扑出现节点或链路故障导致严重损失的情况下的路由可行性，而且只需要很少的控制参数，改良了系统性能。Hong 等<sup>[17]</sup>考虑到以最小化映射嵌入在多域光网络上的给定虚拟业务需求来实现降低总的网络链路开销，由此提出了一种基于分层软件定义网络——H-SDN 作为控制平面进行域间信息交换的通道，然后利用启发式方法通过虚拟拓扑的分区和收缩机制将虚拟链路映射到多域光链路上，通过该方法可以大幅度降低问题的时间复杂度。但上述映射算法会造成底层网络资源的浪费，从而降低虚拟网络请求接受率和系统收益。

综上，将网络中的实体资源规约为可操纵层面上的资源过程中，现有的映射算法没有充分考虑到资源约束、负载均衡以及算法性能等问题，导致映射结果不真实，可信度不高。在上述研究基础上，为提高现有虚拟网络算法的性能，提出了一种在 SDN 网络环境下基于蚁群混合遗传算法的虚拟网络映射方法，将底层物理层网络中的实体资源抽象化为可管理的虚拟资源，即将系统资源规约为运行情况下的可操纵资源。将提高虚拟网络请求的接受率作为求解目标，针对目标函数构造虚拟网络映射问题的线性规划模型，同时，将整个映射过程划分为节点映射阶段和链路映射阶段，从 2 个阶段分别进行映射求解。最后进行了模拟仿真实验与分析，与传统的映射算法相比较，利用蚁群混合遗传算法能够有效地提高虚拟网络请求的接受率。

## 2 虚拟网络映射的层次化结构

面对用户对于现有网络的需求，研究人员不

断地提出旨在提高互联网性能、可靠性以及可扩展性的服务和协议。当下互联网的网络服务供应商主要担任着 2 种角色，一是管理网络中的基础设施，二是提供服务给终端用户，然而，这 2 种角色的耦合却阻碍了新型协议和服务的部署。网络虚拟化技术<sup>[18,19]</sup>的出现，可以从根本上更有效地解决当前 Internet 存在的问题，为未来互联网的发展提供了方向，有效地支持了网络技术的更新。网络虚拟化层次结构如图 1 所示。

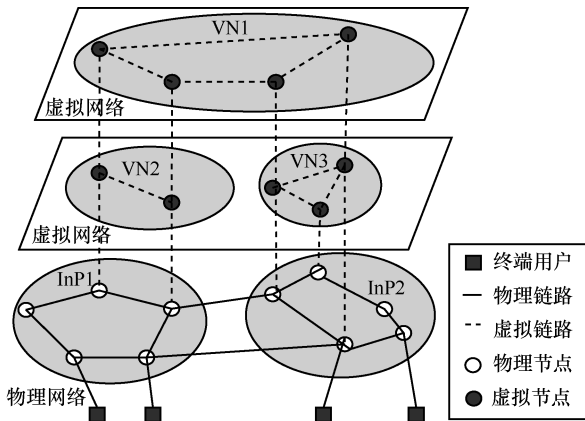


图 1 网络虚拟化层次结构

虚拟网络作为网络虚拟化环境中最基本的组成单元，主要由一组多个虚拟节点和多条连接虚拟节点的虚拟链路组合而成，每一个虚拟节点对应一个特定的物理节点，每一条虚拟链路也相应地对应一条物理链路，并且包含该路径上的部分物理信息。网络虚拟化技术通过将底层的物理网络及其组件（如交换机、路由器等）等物理资源进行抽象、映射、隔离，实现在同一底层物理网络上部署多个相互隔离、互不影响的虚拟网络，而且虚拟网络之

间可以共享底层网络资源，然后根据用户需求提供服务。面向 SDN 的架构中，不仅可以通过集中控制器获得链路信息、网络拓扑信息及交换机状态等，同时可以屏蔽底层物理设备差异，并根据设备状态适时地灵活分配与自主管理底层的物理资源。

### 3 虚拟网络映射的建模与形式化

根据虚拟网络组成，虚拟网络的映射过程通常包括两方面，一是节点的映射，也就是把根据用户的需求所构建的虚拟网络中的虚拟节点映射到底层物理网络上；另外是链路的映射，即在完成节点的映射选择之后，将虚拟网络的链路映射到实际的底层物理网络上。与此同时，被选择的 2 个节点之间的链路必须满足服务对于虚拟网络构建需求中对于虚拟链路的要求。其中，节点一般考虑 CPU、内存和位置因素，而链路则考虑带宽等因素。与传统的虚拟网络映射不同的是，在 SDN 环境下，需要引入 SDN 环境中物理资源的差异，设计高效的虚拟网络映射算法，具体表现为在虚拟网络映射过程中对 SDN 中的底层物理网络和虚拟网络分别引入转发资源 *forward()* 和控制资源 *control()* 并加以表示和描述。图 2 所示为虚拟网络向底层物理网络映射的一种方案。

#### 3.1 底层物理网络的建模及描述

在 SDN 环境下，底层物理网络具有一般图的特性，由此可以构建基于图论模型的底层物理网络的数学模型。在本文中，将底层物理网络用加权无向图表示如下： $G_s = (N_s, L_s, S_s^N, S_s^L)$ 。其中， $N_s$  表示底层物理网络中的所有物理节点的集合，包括路由器节点和服务器节点； $L_s$  表示底层物理网络中的

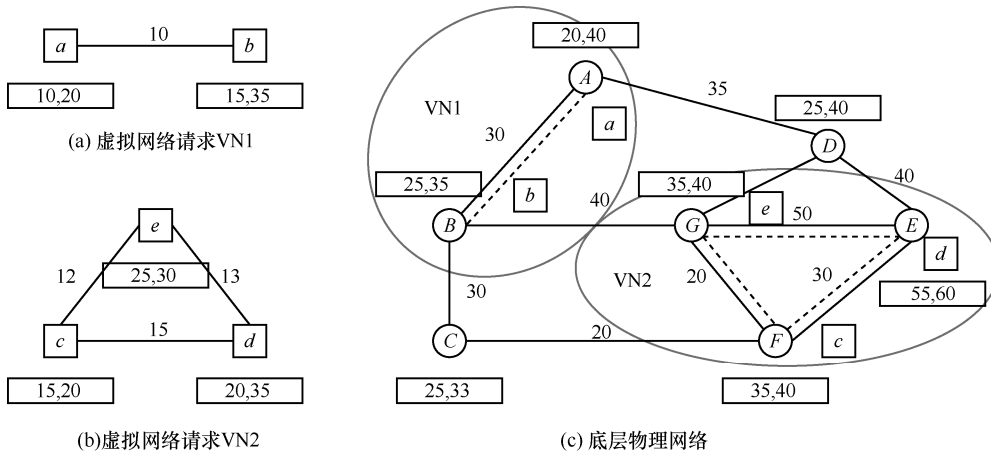


图 2 虚拟网络映射方案

所有物理链路的集合； $S_s^N$  表示底层物理网络中节点  $n_s$  ( $n_s \in N_s$ ) 的属性，包括该节点的转发资源  $forward(n_s)$ ，以及该节点的控制器资源  $control(n_s)$ ； $S_s^L$  表示底层物理网络所有可用的链路  $l_s$  ( $l_s \in L_s$ ) 所具有的属性集合， $l_s^{ij}$  表示节点  $i$  和节点  $j$  之间的链路，该链路当前可用的剩余带宽资源用  $band(l_s^{ij})$  表示； $P_s$  代表底层物理网络中的所有无环链路集合， $P_s(i, j)$  代表节点  $i$  和  $j$  之间的无环链路集合。由于同一虚拟请求中的 2 个虚拟节点不能被同时映射到同一个底层物理网络的物理节点上，所以虚拟链路只能被映射到  $P_s$  上，而不是所有的物理链路集合。

如图 2 所示，物理网络节点周围的矩形框内的 2 个数字分别表示该节点可用的控制资源和转发资源，物理链路上标注的数字则代表该链路可用的带宽资源。

### 3.2 虚拟网络的建模及描述

根据对底层物理网络的拓扑结构的描述，虚拟网络也可用带权无向图  $G_v = (N_v, L_v, C_v^N, C_v^L)$  的形式来表示。其中， $N_v$  代表虚拟网络请求的节点集合； $L_v$  代表虚拟链路的集合；虚拟网络中对虚拟节点  $n_v$  ( $n_v \in N_v$ ) 的约束即资源需求用  $C_v^N$  表示，如该虚拟节点的转发资源需求  $forward(n_v)$  及其控制资源需求  $control(n_v)$ ；虚拟链路  $L_v$  ( $L_v \in L_v$ ) 的资源约束用  $C_v^L$  表示； $band(l_v)$  代表带宽需求。

同样地，如图 2 所示，虚拟网络节点周围的矩形框内的 2 个数字分别表示该节点所需要的控制资源和转发资源，虚拟网络的链路上的数字代表该链路所需要的带宽资源。

### 3.3 虚拟网络映射描述

虚拟网络的映射主要包括虚拟节点的映射和虚拟链路的映射，它可以被表示为  $G_v$  到  $G_s$  的一个子集的一个映射

$$f: G_v \rightarrow (N'_s, L'_s, S_s^n, S_s^l)$$

其中， $N'_s \subset N_s$ ， $L'_s \subset L_s$ ， $S_s^n$  和  $S_s^l$  分别用来表示映射过程中实际分配给虚拟网络请求的节点资源和链路资源。

节点映射和链路映射分别表示为  $f_N$  和  $f_L$

$$f_N: (N_v, C_v^N) \rightarrow N'_s$$

$$f_L: (L_v, C_v^L) \rightarrow P'_s$$

其中， $P'_s \subset P_s$ 。

图 2 所示的虚拟网络请求 VN1 和 VN2 的映射

方案表示为  $\{a \rightarrow A, b \rightarrow B\}$  和  $\{c \rightarrow F, d \rightarrow E, e \rightarrow G\}$ ，链路映射方案分别表示为  $\{(a, b) \rightarrow (A, B)\}$  和  $\{(c, d) \rightarrow (F, E), (c, e) \rightarrow (F, G), (e, d) \rightarrow (G, E)\}$ 。同时，任何映射方案必须满足虚拟网络请求映射的约束条件。

## 4 虚拟网络映射约束条件及优化目标

对于虚拟网络映射中的节点映射，任意一个虚拟网络请求中的虚拟节点  $n_v$  ( $n_v \in N_v$ )，都必须通过某种映射关系逐一向底层物理网络的物理节点映射。其中，对于存在于同一个虚拟网络请求中的所有虚拟节点，每一个虚拟节点只能被映射到底层物理网络中的一个物理节点上，位于不同的虚拟请求中的虚拟节点可以被映射到同一个底层节点上。在本文中，用  $f_N(i, j)$  表示虚拟节点  $n_v^i$  向物理节点  $n_s^j$  的映射是否成功，若节点映射成功，则  $f_N(i, j) = 1$ ；否则， $f_N(i, j) = 0$ 。节点映射需要满足的约束关系为

$$\begin{cases} control(n_s^j) \geq control'(n_v^i), f_N(i, j) = 1 \\ forward(n_s^j) \geq forward'(n_v^i), f_N(i, j) = 1 \\ \sum_{n_v^i \in N_v} f_N(i, j) \leq 1, \forall n_s^j \in N_s \\ \sum_{n_s^j \in N_s} f_N(i, j) = 1, \forall n_v^i \in N_v \\ f_N(i, j) \in \{0, 1\} \end{cases} \quad (1)$$

其中， $control'(n_s^j)$  代表物理节点  $n_s^j$  的剩余控制资源， $forward'(n_s^j)$  代表物理节点  $n_s^j$  的剩余转发资源，如式(2)所示。

$$\begin{cases} control'(n_s^j) = control(n_s) - \sum_{f_N(i, j)=1} control(n_v^i) \\ forward'(n_s^j) = forward(n_s) - \sum_{f_N(i, j)=1} forward(n_v^i) \end{cases} \quad (2)$$

只有虚拟网络请求中的所有虚拟节点都根据某种映射关系向物理网络节点映射成功，才说明节点映射成功，否则，节点映射失败。

对于虚拟网络映射中的链路映射，为了降低映射的复杂度并提高映射算法的效率，假设底层物理网络不支持路径分割，在此前提下进行链路映射。首先选取能够成功映射的物理节点，然后利用最短路径算法，将虚拟网络请求中的每一条虚拟链路映射到相应的物理链路上。同理，该物理链路上的带宽资源必须能够满足被映射的虚拟链路对于带宽

的需求，其约束条件用数学表达式为

$$band'(l_s^i) \geq band(l_v^j), \forall l_s^i \in P_s(i, j), P_s(i, j) = f_L(i, j) \quad (3)$$

其中， $band'(l_s^i)$  表示物理链路  $l_s^i$  的剩余带宽资源， $f_L(i, j)$  表示节点  $i$  和  $j$  之间的链路映射，如式(4)所示。

$$\begin{cases} band'(l_s^i) = band(l_s^i) - \sum_{l_v^j \in f_L(i, j)} band(l_v^j) \\ f_L(i, j) \in \{0, 1\} \end{cases} \quad (4)$$

现有对于虚拟网络映射问题的研究中，对于映射结果的评价标准也因目标和需求的不同而各不相同。其中，常用的主要评估指标主要有算法构建的成本与效益、虚拟网络请求的接受率、底层物理网络的负载强度、节点和链路的平均利用率、构建算法的时间复杂度以及底层网络资源的利用率等。虚拟网络映射最初的设计目标，只需要将虚拟网络请求中的节点和链路能够在满足约束条件的前提下映射到底层物理网络的节点和链路上。之后，考虑到实际运营问题，提出了构建虚拟网络映射的成本与效益问题<sup>[20]</sup>。基础设施供应商希望在长期运营的过程中能够将虚拟网络映射成本最小化以及效益最大化，虚拟网络映射的成本与效益表示如下

$$\begin{cases} C(i) = \sum_{n_v \in N_v^i} f(n_v) + \sum_{l_v \in L_v} \sum_{l_s \in L_s} b(l_v, l_s) \\ C = \frac{\sum_{i \in I} C(i)}{T + \delta} \\ R_G(i) = \sum_{n_v \in N_v^i} f(n_v) + \sum_{l_v \in L_v} c(l_v) \\ R = \frac{\sum_{i \in I} R_G(i)}{T + \delta} \\ R_r = \frac{R_G(i)}{C(i)} \end{cases} \quad (5)$$

其中， $I$  表示在时间  $0 \sim T$  内底层网络所接受的虚拟网络的请求；虚拟网络映射构建的成本用  $C(i)$  表示，在时间  $T$  内的底层物理网络长期运营的平均资源开销为  $C$ ； $\sum_{n_v \in N_v^i} f(n_v)$  指节点映射过程中所消耗的资源； $\sum_{l_v \in L_v} \sum_{l_s \in L_s} b(l_v, l_s)$  指虚拟链路  $l_v$  映射到  $l_s$  上的过程中所消耗的资源；基础设施供应商接受第  $i(i \in I)$

个虚拟网络所获得收益用  $R_G(i)$  表示； $\sum_{n_v \in N_v^i} f(n_v)$  表

示在满足请求资源的约束下完成一次映射的节点收益；同样， $\sum_{l_v \in L_v} c(l_v)$  表示在完成链路映射后得到

的收益； $R_r$  表示虚拟网络映射的收益与成本之比，在时间  $T$  内的底层物理网络长期运营的平均效益为  $R$ ， $\delta$  为一个无限趋于 0 的正数。

虚拟网络请求的接受率为

$$P = \frac{\sum_{i=0}^T VN_S}{\sum_{i=0}^T VN} \quad (6)$$

其中， $VN_S$  表示成功被映射到底层物理网络的虚拟网络请求个数， $VN$  表示虚拟网络请求的总个数。

本文的研究是基于底层物理网络不支持路径分割的前提下进行的，所以符合节点和链路映射关系  $f(i, j) \in \{0, 1\}$ ，这种规划属于整数线性规划问题。本文以提高虚拟网络请求的接受率作为主要的评估指标，即以虚拟网络请求的接受率作为目标函数，对映射算法进行优化，目标函数为

$$\max \left( f(i, j) \frac{\sum_{i=0}^T VN_S}{\sum_{i=0}^T VN} \right) \quad (7)$$

其中，当节点和链路都映射成功时  $f(i, j) = 1$ ，若节点或者链路映射不成功，则  $f(i, j) = 0$ 。

## 5 基于蚁群混合遗传算法的虚拟网络映射

为了解决网络虚拟化的资源映射问题，基于蚁群算法和遗传算法的思想，提出了基于蚁群混合遗传算法的虚拟网络映射算法。这种算法是一种 2 阶段映射算法，也就是虚拟节点和虚拟链路的映射过程不在同一阶段进行。首先根据资源的约束条件对节点进行预选取，然后利用该融合算法对映射中的节点进行映射，最后使用最短路径算法对虚拟链路进行映射。

### 5.1 节点预选取

首先，为每个虚拟节点找一个可以映射的节点集合  $\Phi(i) = \theta(n_v^i) \cap \{n_s \in N_s | f_N(i, j) = 1\}$ 。其中， $\theta(n_v^i) = \{n_s \in N_s\}$  为所有虚拟节点的集合。若  $\Phi(i)$  为空集，表示在底层物理网络中不存在符合虚拟请求中虚拟节点的映射，虚拟网络请求失败；反之，通

过最短路径算法选择虚拟节点  $n_v^i$  和  $n_v^j$  分别映射到物理节点  $n_s^i$  和  $n_s^j$  之间的无环路径集合中满足约束条件的路径作为映射的链路。

## 5.2 虚拟网络映射算法步骤

基于蚁群混合遗传算法的虚拟网络映射算法, 首先需要确定遗传算法和蚁群算法的初始化参数, 如遗传算法的编码规则、初始种群  $X$ 、种群中个体数目  $N$ 、适应度函数  $fit(X_i)$  以及交叉算子  $P_c$ 、变异算子  $P_m$  等遗传操作的设计, 以下蚁群算法的蚂蚁数量  $M$ 、迭代次数  $N$ 、信息素分布的  $Q$  值等。算法的输入及输出如下。

**输入** 虚拟网络请求的拓扑图  $G_v$ , 底层物理网络拓扑图  $G_s$ , 确定好编码方式以及初始化的各个控制参数。

**输出** 映射节点对  $(n_v^i, n_s^i)$ , 其中,  $n_v^i \in N_v$ ,  $n_s^i \in N_s$ 。

基于对混合算法的认知, 将其应用到虚拟网络映射问题中, 包括以下 3 个模块。

1) 初始化模块: 利用遗传算法得到蚁群算法的信息素初始分布。

**Step1** 利用预选取的节点集合, 随机产生一组作为初始种群  $X = (X_1, X_2, \dots, X_i)$ ;

**Step2** 分别计算  $fit(X_i)$ ;

**Step3** 判断此时是否已达到收敛要求, 如果已经达到, 则输出结果, 反之, 继续运行;

**Step4** 根据  $fit(X_i)$  的大小采取适值比例方法计算每个个体被选择的概率;

**Step5** 采取轮盘赌机制进行个体的选择操作;

**Step6** 将被选择到的个体执行交叉操作;

**Step7** 将蚁群算法的基于信息素的正反馈机制引入到变异操作中;

**Step8** 进化代数增加 1, 返回 Step3;

**Step9** 将取得的结果转化为蚁群算法的信息素初始分布矩阵。

2) 求最优解模块: 利用遗传算法得信息素初始分布以及最初设定的算法控制参数执行蚁群算法, 解出最优解  $X_i$ 。

**Step1** 将禁忌表 Table 设置为空;

**Step2** 设置蚂蚁路径的起点, 更新允许访问的节点列表 NodeList、未访问节点列表 UnNodeList 和禁忌表 Table;

**Step3** 根据选择策略选择下一节点;

**Step4** 更新允许访问节点列表 NodeList、未访问节点列表 UnNodeList、禁忌表 Table 及信息素浓度;

**Step5** 如果未访问节点列表 UnNodeList 为空, 则返回 Step2, 反之, 继续执行;

**Step6** 计算每只蚂蚁经过的路径的长度  $L$  和等待时间  $T$ , 记录当前的最优解  $X_i$ 。

3) 输出最优解模块: 保存蚂蚁觅食过程中经过的路径, 输出最优解  $X_i$ , 进而通过最短路径算法对链路进行映射。

## 5.3 虚拟网络映射算法求解

蚁群算法是 Dorigo 等<sup>[21]</sup>提出的一种新型的仿生优化算法, 该算法能够很好地解决求解问题的解的优化问题。但是蚁群进化到一定的代数后容易导致较优解所在的路径上的信息素浓度过高, 使过多的蚂蚁聚集于个别的路径上, 出现早熟停滞现象, 导致得到的结果是局部最优解。另外, 刚初始化的蚁群初期信息素匮乏, 导致求解速度很慢, 计算时间过长。遗传算法提出了一个求解问题的通用框架, 可以用来表示现实复杂的现象, 快速地解决难以解决的问题。然而, 在应用遗传算法解决问题的过程中, 也暴露了一些问题, 如早熟收敛现象的出现, 导致算法的求解结果容易陷入局部最优解, 而且算法后期的收敛速度明显下降; 另外, 在算法中往往只考虑种群内个体的竞争关系, 忽略了种群内个体之间的协同关系。

针对上述 2 种算法的局限, 本文提出了一种蚁群混合遗传算法, 来解决虚拟网络映射问题。基于蚁群混合遗传算法的虚拟网络映射算法, 首先利用遗传算法对蚁群算法中的初始信息素的分布进行优化, 通过这种做法就可以克服由于蚁群算法初始化过程中由于规模过大而导致的收敛速度过慢的不足。然后, 再利用蚁群算法的并行性和正反馈机制, 针对要求解的网络虚拟化的映射问题求出优化解, 从而避免了遗传算法的求解过程中陷入局部最优的问题。

本文所提出的虚拟网络映射算法是以蚁群算法为基础, 主要通过蚁群算法的初始信息素环节引入遗传算法, 利用遗传算法的全局收敛性对蚁群算法的初始信息素的分布做出指引, 优化对蚁群算法产生重大影响的启发因子  $\alpha$ 、期望因子  $\beta$  和信息素挥发度  $\rho$  这 3 个参数, 得到最优的初始信息素分布, 然后再通过蚁群算法对问题的解进行优化。其

中, 将蚁群算法中的转移概率矩阵转换为当前虚拟网络中的虚拟节点映射到底层物理网络中的每个物理节点的概率, 如果当前的物理节点资源满足不了虚拟请求中的虚拟节点的资源约束, 则设置转移概率为 0, 将其表示为

$$P_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{n'_s \in N_s} [\tau_{in'_s}(t)]^\alpha [\eta_{in'_s}]^\beta}, \forall n'_s \in N_s \\ 0, \text{其他} \end{cases} \quad (8)$$

其中,  $h_{ij} = \frac{1}{\sum_{l^j \in L_v} Resource(Map(l^j))}$ ,  $l^j$  表示虚拟节点  $n_v^i$  映射到物理节点  $n_s^j$  上的链路映射集合, 而  $Resource(Map(l^j))$  则表示虚拟链路映射到物理链路的资源消耗大小;  $\tau_{ij}(t)$  表示  $t$  时刻, 虚拟节点  $n_v^i$  映射到底层物理网络中的物理节点  $n_s^j$  上的信息素含量。另外,  $\alpha$  和  $\beta$  用于调节信息素和启发式因子的权重。该算法会应用轮盘赌算法, 根据节点映射概率选择被映射的底层物理节点。然后, 采取以下方法对节点进行预选取, 进而通过最短路径算法对链路进行映射。

对于算法信息素设置与更新规则, 本文采取最大最小蚁群算法<sup>[22]</sup> (MMAS) 的规则, 它最显著的特点就是将各边的信息素的取值控制在  $(\tau_{min}, \tau_{max})$  范围之内, 而且对于信息素的更新并不是对每一条路径进行更新, 而是只对每次迭代中产生的最短路径上的信息素进行更新, 采取这

种更新规则, 使搜索有效地控制在最短路径附近, 这样有利于更简便地找到最优解, 有效解决早熟和停止问题。根据文献[22]中提出的 MMAS 算法中对信息素的设置和更新策略, 由此得到信息素  $\tau_{ij}(p)$  的更新方程为

$$\tau_{ij}(p) = \begin{cases} (1 - \rho)\tau_{ij}(p) + \Delta\tau_{ij}(p)_{add} \\ \Delta\tau_{ij}(p)_{max}, \tau_{ij}(p) > \Delta\tau_{ij}(p)_{max} \\ \Delta\tau_{ij}(p)_{min}, \tau_{ij}(p) < \Delta\tau_{ij}(p)_{min} \end{cases} \quad (9)$$

其中,  $\rho$  是信息素的挥发率,  $\Delta\tau_{ij}(p)_{max}$  和  $\Delta\tau_{ij}(p)_{min}$  分别代表边  $l_s(i, j)$  上信息素的最大值和最小值 (本文中分别取值  $\infty$  和 60),  $\Delta\tau_{ij}(p)_{add}$  代表边  $l_s(i, j)$  上信息素的增加量, 它的取值如式(10)所示, 其中,  $Q$  是常数。

$$\Delta\tau_{ij}(p)_{add} = \begin{cases} \frac{Q}{t}, (i, j) \text{属于迭代最短路径} \\ 0, \text{其他} \end{cases} \quad (10)$$

为了求得虚拟网络映射问题的解, 在蚁群混合遗传算法中, 采用实数编码的方式, 将单个蚂蚁的状态对应问题的解, 表示如下: 单只蚂蚁的当前状态  $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ , 其中,  $d$  表示虚拟网络请求中虚拟节点的个数,  $x_{ij}$  代表虚拟节点映射到物理节点的编号。

## 6 仿真实验

该实验在 SDN 环境下, 采用 3 层验证平台进行仿真实验, 如图 3 所示。其中, 物理层利用 Mininet

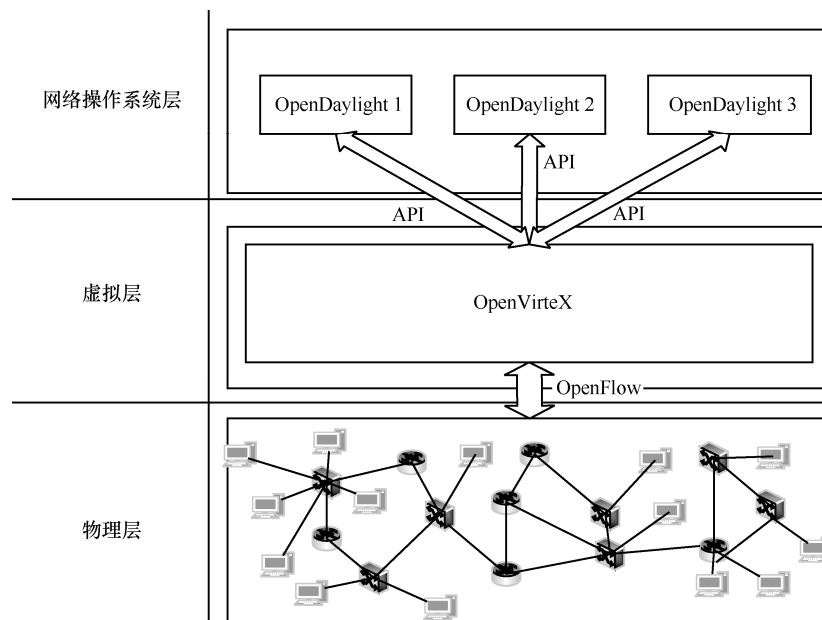


图 3 SDN 网络下的实验验证平台架构

软件定义一个中等规模的 SDN；虚拟化层利用 OpenVirteX 软件创建多个虚拟网络，并对这些虚拟网络进行管理；而网络操作系统层则利用 OpenDaylight 控制器下发流表对 SDN 按需求进行控制。

参照文献[11]，利用 GT-ITM 网络拓扑生成器生成底层物理网络，将其设计为含有 50 个节点和大约 25 条链路的拓扑结构，每一对物理节点之间的连接概率设置为 0.5，并且每个节点的转发资源和控制资源以及物理链路的带宽资源分别服从[50,100]的均匀分布。虚拟网络请求的生存时间指的是虚拟网络请求从到达开始直到请求失效的时间差，在仿真实验中，采用时间窗对虚拟网络请求的到达与离开进行表示，每个时间窗为 100 个时间单位，在一个时间窗内，设置虚拟网络请求的到达数目为均值是 4 的泊松分布，并且设置虚拟网络请求的生存时间为均值是 10 个的时间窗的指数分布。每个虚拟网络请求中包含服从[2,10]的整数均匀分布的虚拟节点数，每一对虚拟节点之间的连接概率设置为 0.5，设置虚拟节点的转发资源量和控制资源量以及虚拟链路所需要的带宽资源量都服从[0,20]的整数均匀分布。

在实验中，对于蚁群混合遗传算法中的参数设定如下：设置蚂蚁数量  $M$  为 50，迭代次数  $N$  为 50，交叉算子  $P_c$  为 0.75、变异算子  $P_m$  为 0.05，信息素分布的  $Q$  值为 1 000。为了取得较为准确并且可信度高的实验结果，重复进行 5 次实验，

然后取平均值进行分析。本文以虚拟网络请求的接受率作为主要评价目标，同时也对节点和链路的平均利用率、虚拟网络映射的收益成本比值以及虚拟网络请求的平均等待时间进行分析，将仿真实验结果与 RW-BFS<sup>[10]</sup>算法、D-ViNE<sup>[11]</sup>算法和 R-ViNE<sup>[11]</sup>算法进行比较，实验结果与分析如下。

虚拟网络请求的接受率作为本文所提出算法的主要评价指标，从图 4 中可以看出，在相同的实验环境下，4 种映射算法的虚拟网络请求接受率随着时间的增加逐渐降低但是却趋于稳定。而且从实验结果图中显示，请求率最差的是 R-ViNE 算法，而该融合算法的接受率始终高于其他算法，最终稳定在 88%左右。主要原因就是该融合算法对于初始种群进行了预选取，这在一定程度上优化了实验结果，而且该算法能够充分利用遗传算法的全局收敛性和蚁群算法的并行性以及信息素的正反馈机制，很大程度上提高了虚拟网络映射的效率。

对 4 种映射算法的节点平均利用率的仿真结果进行了比较，实验结果如图 5 所示。节点平均利用率指的是底层物理网络节点已利用的转发资源和控制资源与物理节点上的转发资源和控制资源的总和的比值。通过比较发现，在仿真实验开始的最开始阶段，融合算法的节点平均利用率相比于其他算法并没有特别明显的优势，但是随着时间的增加，其优势也愈发明显，最终结果趋于 45%，而其他 3 种映射算法的结果趋于 35%，远远小于融合算法的节点平均利用率。

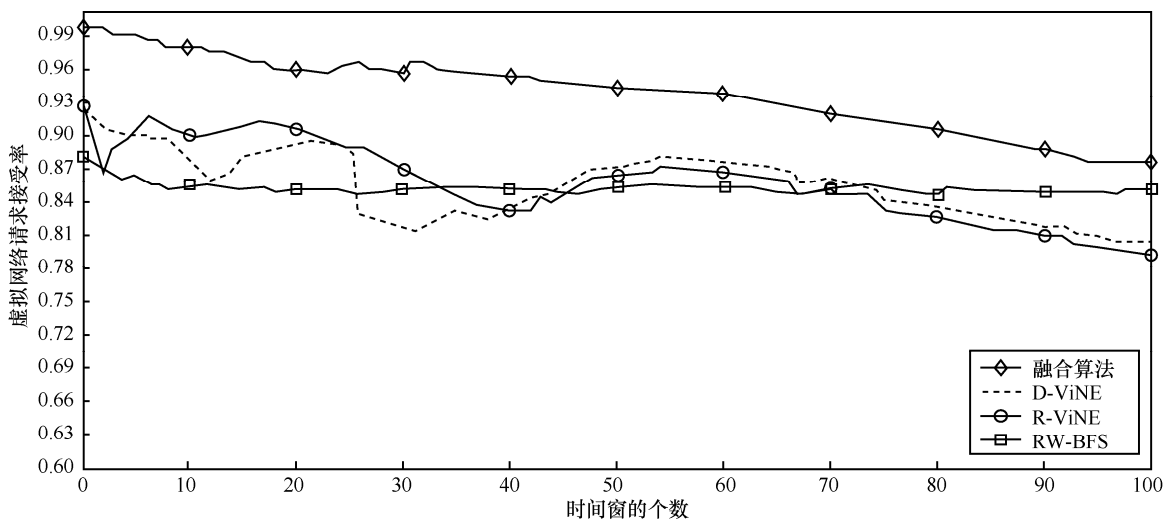


图 4 虚拟网络请求接受率

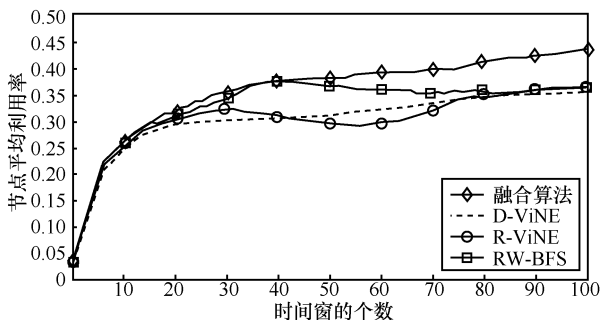


图 5 节点平均利用率

图 6 比较了 4 种映射算法的链路平均利用率的仿真结果。链路平均利用率指的是底层物理网络中已利用的带宽资源量除以总的带宽资源量。在链路平均利用率的比较中，发现融合算法在实验最开始阶段有着明显的优势，后来反而被 RW-BFS 算法反超，最终结果趋于 27%，相比于其他 3 种算法的 24%，相差并不太多，优势也并不太明显，这主要是因为在该融合算法的目标函数寻优的过程中并未充分考虑到链路的利用率。另外，融合算法采用最短路径对链路进行映射，相比于其他 3 种算法，得到的只是相近的结果。为了取得更好的链路利用率，在后续研究中考虑利用回溯算法进行链路映射的优化，在无向图中，以深度优先搜索的方式选择出符合条件的路径，然后按照链路的利用率大小进行合理映射。

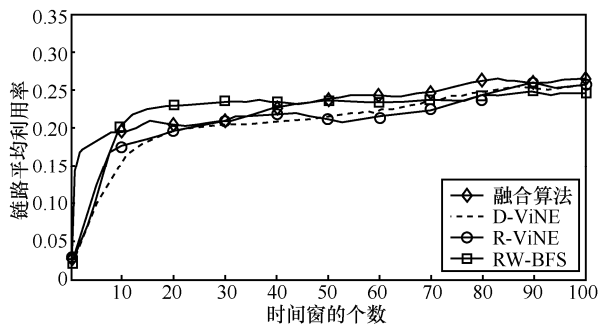


图 6 链路平均利用率

映射算法的虚拟网络映射的收益成本比值仿真结果，如图 7 所示。从图中可以看出，4 种映射算法的收益成本比值结果变化幅度都不太大，相对比较平缓，而融合算法的收益成本比值最终趋于最高，稳定在 82% 左右，结果较为理想。实验结果表明融合算法由于采用了最短路径算法进行链路映射可以在一定程度上降低虚拟网络映射的成本，同时有效地提高了映射的收益。

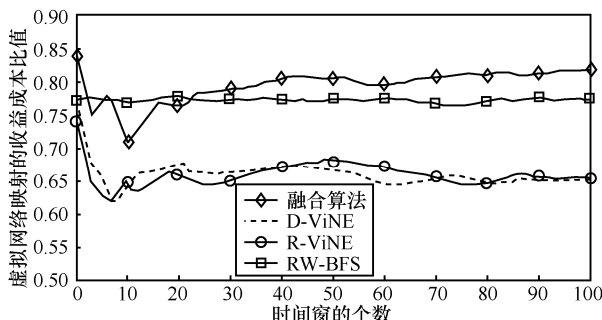


图 7 虚拟网络映射的收益成本比值

图 8 对比了 4 种算法的用户服务所对应的虚拟网络请求的平均等待时间，从图中可以看出，在实验刚开始的时候，4 种算法的平均等待时间相差无几，但是随着时间的推进，虚拟网络请求的等待时间快速延长，但是融合算法的时间增长速度明显低于其他 3 种算法，可见算法在时间指标上的优势。

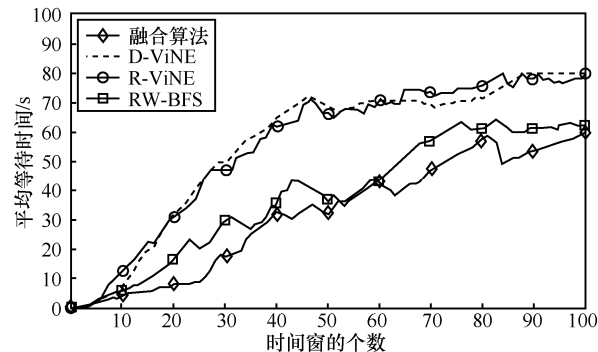


图 8 平均等待时间

综上所述，本文提出的基于蚁群混合遗传算法的虚拟网络映射算法在虚拟网络请求接受率、节点和链路平均利用率、虚拟网络映射的收益成本比值以及平均等待时间等方面都有比较明显的优势，这充分证明了该融合算法能够有效地解决网络虚拟化中的虚拟网络的映射问题。

### 7 结束语

本文主要针对网络虚拟化技术中最重要的虚拟网络映射问题进行探讨和研究，在坚持底层物理资源是有限的且不支持链路分割的前提下，改善了传统的虚拟网络映射问题中效率低、开销大以及过于限制底层物理网络的资源等缺陷。在综合了遗传算法和蚁群算法各自的优点后，提出一种新型的基于蚁群混合遗传算法的虚拟网络映射算法。该算法以提高虚拟网络请求的接受率为目标，通过建立虚拟网络映射的线性规划模型来对虚拟网络的映射

进行求解。仿真实验结果表明，该融合算法的虚拟网络请求接受率比传统的 D-ViNE 算法和 R-ViNE 算法以及 RW-BFS 算法有较为明显的提高。在上述研究基础上，下一步工作将集中在对基于支持链路分割的前提下，虚拟网络映射问题的研究。

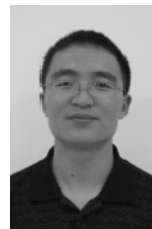
### 参考文献：

- [1] 李小玲, 王怀民, 丁博, 等. 虚拟网络映射问题研究及其进展[J]. 软件学报, 2012, 23(11): 3009-3028.  
LI X L, WANG H M, DING B, et al. Research and development of virtual network mapping problem[J]. Journal of Software, 2012, 23(11): 3009-3028.
- [2] HAN B, GOPALAKRISHNAN V, JI L S, et al. Network function virtualization: challenges and opportunities for innovations[J]. IEEE Communications Magazine, 2015, 53(2): 90-97.
- [3] LIANG C C, RICHARD Y. Wireless network virtualization: a survey, some research issues and challenges[J]. IEEE Communications Surveys & Tutorials, 2015, 17(1): 358-380.
- [4] LEFERIS M, STUART C, ALEX G. Information exchange management as a service for network function virtualization environments[J]. IEEE Transactions on Network and Service Management, 2016, 13(3): 564-577.
- [5] 张朝昆, 崔勇, 唐嵩祯, 等. 软件定义网络 (SDN) 研究进展[J]. 软件学报, 2015, 26(1): 62-81.  
ZHANG C K, CUI Y, TANG H Y, et al. State-of-the-art survey on software-defined networking(SDN)[J]. Journal of Software, 2015, 26(1): 62-81.
- [6] WANG Z M, WU J X, WANG Y, et al. Survivable virtual network mapping using optimal backup topology in virtualized SDN[J]. China Communications, 2014, 11(2): 26-37.
- [7] FRAJ J, SUBHARTHI P. Network virtualization and software defined networking for cloud computing: a survey [J]. IEEE Communications Magazine, 2013, 51(11): 24-31.
- [8] SHEETAL C, RADHA S, MANI P R. SDN network virtualization survey[C]//International Conference on Wireless Communications Signal Processing and Networking. 2016: 650-655.
- [9] ABDALLAH J, AHMED K. Cost-efficient mapping for fault-tolerant virtual networks[J]. IEEE Transactions on Computers, 2015, 64(3): 668-681.
- [10] CHENG X, SU S, ZHANG Z B, et al. Virtual network embedding through topology-aware node ranking[J]. ACM SIGCOMM Computer Communication Review, 2011, 41(2): 38-47.
- [11] MOSHARAF C, MUNTASIR R R, BOUTABA R. ViNEYard: virtual network embedding algorithms with coordinated node and link mapping[J]. IEEE/ACM Transactions on Networking, 2012, 20(1): 206-219.
- [12] LIU J, HUANG T, CHEN J Y, et al. A new algorithm based on the proximity principle for the virtual network embedding problem[J]. Journal of Zhejiang University-Science C (Computers & Electronics), 2011, 12(11): 910-918.
- [13] DIETRICH D, RIZK A, PAPADIMITRIOU P. Multi-provider virtual network embedding with limited information disclosure[J]. IEEE Transactions on Network and Service Management, 2015, 12(2): 188-201.
- [14] 彭利民. 基于图的邻接分割的虚拟网络映射算法[J]. 华南理工大学学报, 2015, 43(1): 66-71.  
PENG L M. Virtual network mapping algorithm based on group adjacency segmentation[J]. Journal of South China University of Technology, 2015, 43(1): 66-71.
- [15] FESTA P, BRIEF A. Introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems[C]//16th International Conference on Transparent Optical Networks. 2014: 1-20.
- [16] LEZAMA F, GERARDO C, SARMIENTO A M, et al. Survivable virtual topology mapping in IP-over-WDM networks using differential evolution optimization [J]. Photonic Network Communications, 2014, 28(3): 306-319.
- [17] HONG S, JUE J P, PARK P, et al. Survivable virtual topology design in ip over WDM multi-domain networks [J]. IEEE/OSA Journal of Optical Communications & Networking, 2016, 8(6): 408-416.
- [18] WOJCIECH B. Role of network virtualization in designing future Internet[C]// 15th International Telecommunications Network Strategy and Planning Symposium. 2012: 1-3.
- [19] JULIVER DE J G H, JUAN F B V. Network functions virtualization: a survey[J]. IEEE Latin America Transactions, 2016, 14(2): 983-997.
- [20] MARCO S, MASSIMO T, GIACOMO V. Impact of processing costs on service chain placement in network functions virtualization[C]// IEEE Conference on Network Function Virtualization and Software Defined Network. 2015: 191-197.
- [21] DORIGO M, MANIEZZE V, COLORNI A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996, 26(1): 29-41.
- [22] THOMAS S, HOLGER H. Improvements on the ant-system: introducing MAX-MIN ant system[C]//International Conference on Artificial Neural Networks and Genetic Algorithm. 1997: 245-249.

### 作者简介：



王健 (1979-), 女, 黑龙江哈尔滨人, 博士, 哈尔滨理工大学副教授、硕士生导师, 主要研究方向为认知网络、可生存网络、可信计算。



赵国生 (1977-), 男, 黑龙江七台河人, 博士, 哈尔滨师范大学教授、硕士生导师, 主要研究方向为认知网络、可生存网络、SDN。

李志新 (1992-), 女, 山东青岛人, 哈尔滨理工大学硕士生, 主要研究方向为网络与信息安全。