

基于禁忌遗传优化的离线静态虚拟网映射算法

余建军¹ 吴春明²

(衢州职业技术学院 浙江 衢州 324000)¹ (浙江大学计算机科学与技术学院 杭州 310027)²

摘 要 离线静态虚拟网映射问题是 NP 难问题,其任务是以物理网提供商收益最大化为目标,在物理网上完成虚拟网子集的映射。文中对离线静态虚拟网映射问题及其研究现状进行介绍,指出当前离线静态虚拟网映射算法仅适用于小规模问题或特殊问题的求解,进而提出了一种适用于中大规模的一般离线静态虚拟网映射问题的求解算法。首先,基于收益优先的虚拟网映射顺序策略、节点等级匹配的虚拟节点映射策略以及最小化资源消耗量的虚拟链路映射策略,提出离线静态虚拟网映射问题的贪婪算法;然后,基于遗传算法和禁忌搜索混合的优化策略,提出离线静态虚拟网映射问题的禁忌遗传算法。实验表明,所提出的禁忌遗传算法具有较高的虚拟网构建完成率和物理网提供商收益,虚拟网构建完成率和物理网提供商收益分别比基线算法提高了 34% 和 42%。

关键词 离线虚拟网映射,贪婪算法,禁忌遗传算法,NP 难问题

中图分类号 TP393 文献标识码 A DOI 10.11896/jsjcx.181001981

Offline Static Virtual Network Mapping Algorithm Based on Tabu Search Genetic Optimization

YU Jian-jun¹ WU Chun-ming²

(Quzhou College of Technology, Quzhou, Zhejiang 324000, China)¹

(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)²

Abstract The offline static virtual network mapping problem is an NP-hard problem. It is to map the subset of virtual networks onto the physical network aimed to maximize the profit of physical network provider. This paper reviewed the offline static virtual network mapping problem and the current research progress for this problem and pointed out that the current offline static virtual network mapping algorithm is only suitable for solving small-scale problems or special problems. Therefore, this paper proposed a general offline static virtual network mapping algorithm suitable for solving medium and large scale. The virtual network mapping order strategy based on revenue priority, the virtual node mapping strategy by node rank matching and the virtual link mapping strategy that minimizes resource consumption are used to complete the greedy algorithm design for offline static virtual network mapping problem. Then, the optimization strategy based on the hybrid algorithm of genetic algorithm and tabu search are used to complete the tabu search genetic algorithm design for offline static virtual network mapping problem. Experiments show that the proposed algorithm has higher virtual network construction completion rate and gets better physical network provider revenue, and the virtual network construction completion rate and physical network provider revenue are 34% and 42% higher than the baseline algorithm, respectively.

Keywords Offline virtual network mapping, Greedy algorithm, Tabu search genetic algorithm, NP-hard problem

1 概述

网络虚拟化作为构建未来网络的关键技术,旨在通过抽象、分配和隔离机制,在由一个或多个基础设施提供商提供的底层物理网上构建出多个独立且可以是异构的虚拟网,从而为用户提供多样的网络服务^[1-2]。虚拟网映射是网络虚拟化中的一项挑战性任务^[3],即在底层物理网上为虚拟网分配满足相应约束条件的物理资源。

根据虚拟网的请求是动态到达还是预先已知的,将虚拟网映射问题分为在线虚拟网映射问题和离线虚拟网映射问题^[4]。离线虚拟网映射问题是组合优化问题,其优化目标主要是使物理网提供商的收益最大化。同时,离线虚拟网映射问题也是 NP 难问题^[5-6],且由于单个虚拟网映射可行问题是 NP 完全问题^[7],因此离线虚拟网映射问题是不可近似问题。

根据虚拟网请求是否动态变化,可将离线虚拟网映射问题分为离线静态虚拟网映射问题和离线动态虚拟网映射问

收到日期:2018-10-25 返修日期:2019-04-15 本文受浙江省自然科学基金资助项目(LY14F020010),国家 863 高技术研究发展计划项目(2015AA015602,2015AA016013)资助。

余建军(1969—),男,硕士,教授,主要研究方向为网络虚拟化技术以及算法设计与分析,E-mail:yjj691121@126.com(通信作者);吴春明(1967—),男,博士,教授,主要研究方向为互联网体系结构、柔性可重构网络、网络资源弹性管控与虚拟化、网络试验床、网络安全主动防御等。

题。虚拟节点 CPU 容量需求和虚拟链路带宽需求不是常量,而是变量^[8-9],这是虚拟网请求动态性的最常见表现。当前针对离线虚拟网映射问题的求解算法较少^[7-11],已有算法可分为离线动态虚拟网映射算法^[8-9]和离线静态虚拟网映射算法^[7,10-11]两类。

针对离线动态虚拟网映射问题,文献[8]基于高斯分布和昼夜模式组合的虚拟网动态模型,提出了以能耗最小化为目标的离线动态虚拟网映射问题的启发式算法;文献[9]以物理网提供商收益最大化为目标,在构建离线动态虚拟网映射问题的机会约束混合整数线性规划模型之后,提出了相应的启发式算法。

针对离线静态虚拟网映射问题,目前提出的求解算法分为精确算法^[10]和近似算法^[7,11]两类。文献[10]首先构建了内含物理资源批量出租模型的离线静态虚拟网映射问题的混合整数线性规划模型,然后用优化软件 CPLEX12.6 进行求解;文献[7]针对虚拟网拓扑结构是仙人掌图的情景,提出了一种基于线性规划和随机舍入技术的离线静态虚拟网映射问题的近似算法;文献[11]首先提出了内含虚拟网结构的离线静态虚拟网映射问题的线性规划模型,然后提出了一种基于随机舍入技术的离线静态虚拟网映射问题的固定参数近似算法。

由于离线静态虚拟网映射问题是 NP 难问题^[6],因此求解该问题的精确算法具有非多项式时间(指数时间)复杂度,也就是说精确算法仅适用于小规模离线静态虚拟网映射问题的求解。同时如前文所述,一般离线静态虚拟网映射问题是不可近似问题,故所谓的近似算法^[7,11]都是针对某些特殊情况的离线静态虚拟网映射问题。

针对中大规模的虚拟网映射场景,本文基于贪婪策略和混合优化策略,提出了求解一般离线静态虚拟网映射问题的 G_OSVM (Greedy Offline Static Virtual Network Mapping) 算法和 GT_OSVM (Genetic Tabu Offline Static Virtual Network Mapping) 算法。所提出的算法虽不能保证求解的最优性,但一般能在可接受的计算时间和空间内给出问题的有效解。

2 网络模型和问题的定义

2.1 物理网

用无向图 $G^0 = (N^0, E^0)$ 表示物理网, N^0 和 E^0 是其节点集和链路集。第 i ($1 \leq i \leq |N^0|$) 个物理节点 n_i^0 有 CPU 容量和地理位置属性,分别记为 $c(n_i^0)$ 和 $loc(n_i^0)$;第 j ($1 \leq j \leq |E^0|$) 条物理链路 e_j^0 有带宽属性,记为 $b(e_j^0)$ ^[10]。

2.2 虚拟网集合和虚拟网

虚拟网集合记为 $G = \{G^j | j \in R, R = [1, n]\}$,其中无向图 $G^j = (N^j, E^j)$ 表示第 j 个虚拟网, N^j 和 E^j 分别是第 j 个虚拟网的节点集和链路集;第 j 个虚拟网的第 i ($1 \leq i \leq |N^j|$) 个虚拟节点 n_i^j 有 CPU 容量需求和地理位置属性,分别记为 $c(n_i^j)$ 和 $loc(n_i^j)$;第 j 个虚拟网的第 i ($1 \leq i \leq |E^j|$) 条虚拟链路 e_i^j 有带宽属性,记为 $b(e_i^j)$ 。另外,第 j 个虚拟网的虚拟网映射收益属性表示物理网提供商在完成该虚拟网映射后所获得的收益;第 j 个虚拟网的属性 d_j 表示虚拟节点与所映射的物理

节点间的距离必须小于或等于 d_j ,该约束条件称为虚拟节点映射的位置约束条件。

2.3 离线静态虚拟网映射问题的定义

离线静态虚拟网映射问题指给定一个物理网 $G^0 = (N^0, E^0)$ 和一个包含 n 个静态虚拟网(指属性值不会动态变化的虚拟网)的虚拟网集合 $G = \{G^j | j \in R, R = [1, n]\}$,求解 R 的子集 R' ,以使 R' 中所包含的虚拟网的映射收益之和最大化,即最大化物理网提供商的收益。子集 R' 满足条件 1) 和条件 2)。1) 对于任意 $r \in R'$,完成对应的虚拟网 $G^r = (N^r, E^r)$ 的映射,具体包括虚拟节点映射和虚拟链路映射:①将虚拟节点 n_i^r ($1 \leq i \leq |N^r|$) 映射到底层物理网中满足位置约束的物理节点上,同时在映射的物理节点上为 n_i^r 分配 CPU 容量资源;②将 e_i^r ($1 \leq i \leq |E^r|$) 映射到以 e_i^r 的两个节点所映射的物理节点为端点的唯一一条物理路径上,同时在该路径上为 e_i^r 分配带宽资源;③每个虚拟节点必须映射到不同且唯一的物理节点上。2) 完成子集 R' 所对应的虚拟网子集 $\{G^r | r \in R'\}$ 映射后,各物理节点所分配的 CPU 总容量和各物理链路所分配的总带宽分别不能超出各物理节点的 CPU 容量和各物理链路的带宽。

3 离线静态虚拟网映射问题的贪婪算法 G_OSVM

3.1 G_OSVM 算法的基本思想

G_OSVM 算法的核心是虚拟网映射顺序策略和虚拟网映射策略。首先,将 n 个虚拟网按照映射收益进行降序排序,并按照排序依次进行映射,即给高收益的虚拟网更高的优先级。该步骤包含了收益优先的虚拟网映射顺序策略。然后,在映射某个虚拟网时,采用两阶段映射方法:在虚拟节点映射阶段,各虚拟节点按照所需资源量的大小进行排序,并按顺序将虚拟节点映射到符合位置和容量约束的可用资源量最大的物理节点上,这就是虚拟节点映射时所采用的贪婪策略;在虚拟链路映射阶段,各虚拟链路按照带宽需求量的大小进行排序,并采用 K 短路径算法^[12]依次将虚拟链路映射到符合容量约束的相对资源消耗量最小的物理路径或最短的物理路径上,这就是虚拟链路映射时所采用的贪婪策略。

物理节点的可用资源量和虚拟节点的所需资源量都采用节点级别(noderank)表示^[13]。节点级别是基于马尔可夫随机游走模型,根据节点(包括物理节点和虚拟节点)自身的资源可及性以及与其相连节点的资源可及性来计算的,具体计算方法参见文献[13]。

采用 K 短路径算法进行虚拟链路映射的关键是确定物理链路的权重,具体是通过评估物理链路总带宽、物理网最短路径平均长度和所有虚拟网的链路总带宽等确定物理链路的权重取 1 还是取虚拟链路带宽与物理链路带宽之比,具体见 G_OSVM 算法流程。

3.2 G_OSVM 算法流程

G_OSVM 算法的具体流程如算法 1 所示。

算法 1 G_OSVM 算法

输入:物理网 $G^0(N^0, E^0)$,虚拟网集合 $G = \{G^j | j \in R, R = [1, n]\}$, k, ϵ 和 h

输出:完成映射的虚拟网子集 R'

1. a = 物理链路总带宽/所有虚拟网的链路总带宽, b = 物理网最短路径的平均长度;
 // 若 flag 等于 1, 则物理链路权重取相对值

2. if ($a > b \times h$) { flag = 1; } else { flag = 0; }

3. 将 n 个虚拟网按照映射收益排序, 它们将构成按照映射收益降序排序的新的虚拟网集合 G // 满足 $\rho_i \geq \rho_j (i < j)$

4. $i = 1$

5. while ($i \leq n$) // 外循环, 每次处理一个虚拟网映射

{

5.1. 采用文献[13]中的算法 1, 利用指定的 ϵ 计算所有物理节点 n_s^0 ($s \in [1, |N^0|]$) 和第 i 个虚拟网的所有虚拟节点 n_v^i ($v \in [1, |N^i|]$) 的节点级别 $r(n_s^0)$ 和 $r(n_v^i)$ 。

5.2. 将第 i 个虚拟网络 G^i 的虚拟节点序号按照节点级别降序规则重新设定, 即重新设定后能够保证 $r(n_a^i) \geq r(n_b^i) (a < b)$, 同时修改相应虚拟链路的两个端点的序号。

5.3. $m = 1$ 。

5.4. while ($m \leq |N^i|$) // 内循环, 完成每个虚拟节点的映射

{

5.4.1. 求满足 n_i^m 的位置和容量约束且没有被第 i 个虚拟网的其他虚拟节点所映射的物理节点的集合 V 。

5.4.2. if (V 不空) { 将 n_i^m 映射到 V 中节点级别最大的物理节点上; 并且在该物理节点上为 n_i^m 分配 CPU 容量资源。}

5.4.3. if (V 空) { break; } // 跳出内循环

5.4.4. $m = m + 1$ 。
 } // 内循环结束

5.5. if ($m \leq |N^i|$) { 释放分配给 G^i 的所有物理网资源; 转步骤 5.8; } // 当有虚拟节点不能完成映射时, 终止当前虚拟网映射, 转下一个虚拟网映射。

5.6. if (flag == 1)

{

5.6.1. $m = 1$ 。

5.6.2. while ($m \leq |E^i|$) // 内循环, 完成每条虚拟链路的映射

{

5.6.2.1. 将物理网 G^0 中可用带宽小于 $b(e_i^m)$ 的物理链路删除, 并将物理链路的权重设为 $b(e_i^m)$ 与其带宽之比, 然后构建物理网 G^{0b} 。

5.6.2.2. 在物理网 G^{0b} 中用 K 短路径算法[12]求解虚拟链路 e_i^m 所能映射的 k 条物理路径。

5.6.2.3. 在 k 条路径中选择跳数最少的路径 VP。

5.6.2.4. if (VP 不空) { 将 e_i^m 映射到 G^0 的物理路径 VP; 并在该物理路径上为 e_i^m 分配带宽资源。}

5.6.2.5. if (VP 为空) { break; } // 跳出内循环

5.6.2.6. $m = m + 1$ 。
 } // 内循环结束

5.6.3. if ($m \leq |E^i|$) { 释放分配给 G^i 的所有物理网资源; 转步骤 5.8; } // 当有虚拟链路不能完成映射时, 终止当前虚拟网映射, 转下一个虚拟网映射。

5.6.4. 将第 i 个虚拟网 G^i 添加进完成映射的虚拟网集 R' 。
 } // end if

5.7. if (flag == 0)

{

5.7.1. $m = 1$ 。

5.7.2. while ($m \leq |E^i|$) // 内循环, 完成每条虚拟链路的映射

{

5.7.2.1. 物理网 G^0 中可用带宽小于 $b(e_i^m)$ 的物理链路删除, 并将物理链路的权重都设置为 1, 然后构建物理网 G^{0b} 。

5.7.2.2. 在物理网 G^{0b} 中用 K 短路径算法[12]求解虚拟链路 e_i^m 所能映射的 k 条物理路径。

5.7.2.3. 在 k 条物理路径中选择相对资源消耗最少的物理路径 VP。// 物理路径的相对资源消耗量指虚拟链路 e_i^m 的带宽与该路径上的物理链路带宽的比值之和。

5.7.2.4. if (VP 不空) { 将 e_i^m 映射到 G^0 的物理路径 VP; 并在该物理路径上为 e_i^m 分配带宽资源。}

5.7.2.5. if (VP 为空) { break; } // 跳出内循环

5.7.2.6. $m = m + 1$ 。
 } // 内循环结束

5.7.3. if ($m \leq |E^i|$) { 释放分配给 G^i 的所有物理网资源; 转步骤 5.8; } // 当有虚拟链路不能完成映射时, 终止当前虚拟网映射, 转下一个虚拟网映射。

5.7.4. 将第 i 个虚拟网络 G^i 添加进完成映射的虚拟网集 R' 。

} // end if

5.8. $i = i + 1$ 。

} // 外循环

6. 输出完成映射的虚拟网集 R' , 结束。

4 离线静态虚拟网映射问题的禁忌遗传算法 GT_OSVMN

4.1 GT_OSVMN 算法的基本思想

GT_OSVMN 算法是求解离线静态虚拟网映射问题的禁忌遗传算法。禁忌遗传算法[14]基于禁忌搜索思想, 将遗传算法的交叉算子和变异算子改进为具有记忆功能的禁忌交叉算子和禁忌变异算子, 从而使算法能够跳出局部最优解, 提升算法获得更优解的概率。

(1) 染色体编码和初始种群的生成

针对包含 n 个虚拟网的虚拟网集合 $G \{G^j | j \in R, R = [1, n]\}$, 用长度为 $\sum_{a \in R} |N^a|$ 的串表示虚拟网集合的虚拟节点映射方案 (即染色体或个体)。

染色体的第 $\sum_{a \in [1, j-1]} |N^a| + i$ 个元素 (表示第 j 个虚拟网的第 i 个虚拟节点 n_i^j) 的值表示 n_i^j 所映射的物理节点, 其值要么为空, 要么属于集合 $\Omega(n_i^j)$ (满足 n_i^j 映射的位置约束条件的物理节点集合)。

染色体的子串 $[\sum_{c \in [1, a-1]} |N^c| + 1, \sum_{b \in [1, a]} |N^b|]$, 表示第 a 个虚拟网中所有虚拟节点所映射的物理节点。如果子串中元素取值出现空值的情况, 则表明相应虚拟节点没有完成映射。子串中各元素的取值必须不同 (空值除外), 以保证该虚拟网的每个虚拟节点映射到不同的物理节点或没有完成映射。

初始种群包含 N 个个体, 其中 $N-2$ 个个体通过随机方法生成, 另外采用 G_OSVMN 算法中的虚拟节点映射方法 (即使该算法仅完成部分虚拟节点的映射, 其结果也放入初始种群) 生成 1 个个体, 最后 1 个个体是用全空染色体表示的。

(2) 适应度函数

每个染色体 $i (1 \leq i \leq N)$ 包含虚拟网集合 G 中每个虚拟网 $G^j (j \in [1, n])$ 的虚拟节点映射方案, 采用 G_OSVMN 算

法的虚拟链路映射方法,可求得虚拟网集合 G 中每个虚拟网 G^j 的映射方案,然后计算映射收益 $\sum_{j \in R} (\rho_j \times y_j)$ (若完成第 j 个虚拟网的构建,则 y_j 取 1,否则 y_j 取 0),将其作为该染色体的适应度 F_i 。若染色体中对应第 a 个虚拟网的子串中出现空值,则 y_a 直接取 0;若染色体 i 所确定的虚拟节点映射方案导致某个物理节点所分配的 CPU 总容量超出该物理节点的 CPU 容量,则 F_i 取 0。

(3) 选择算子

采取无放回余数随机选择策略^[15]选择算子,具体步骤如下:1)计算第 $i(1 \leq i \leq N)$ 个个体的生存数目 $[N_i]$,其中 $N_i = N \times F_i / \sum_{j=1}^N F_j$;2)用赌盘选择法确定其余 $N - \sum_{j=1}^N [N_j]$ 个个体。

此方法可确保适应度值比平均适应度值大的个体一定能够被遗传到下一代群体中,因此选择误差较小。

(4) 禁忌交叉算子

在一个长度为 L 的禁忌表中记录染色体的适应度值,该值的渴望水平取父代群体适应度值的平均值。

首先,使用双点交叉法^[15]产生新的染色体,即随机产生两个父代染色体的两个交叉点,然后交换两个染色体的两个交叉点间的部分染色体。例如,两个父代染色体为 $A = 17, 4, | 5, 6, 1 |, 2, 20, 23$ (包含两个虚拟网,每个虚拟网都含 4 个虚拟节点,两个虚拟网的 8 个虚拟节点分别被映射到第 17, 4, 5, 6, 1, 2, 20, 23 号物理节点上)和 $B = 3, 36, | 4, 7, 12 |, 35, 9, 13$,其中交叉点用符号“|”表示,则对 A 和 B 执行双点交叉后将得到 $A_1 = 17, 4, | 4, 7, 12 |, 2, 20, 23$ 和 $B_1 = 3, 36, | 5, 6, 1 |, 35, 9, 13$ 。若双点交叉操作产生的新染色体串中与同一虚拟网对应的子串中存在相同序号(空值除外)的情况,则用随机方法对不属于交换部分的元素进行有效、不重复的再映射;若在 A_1 串所表示的虚拟节点映射方案中,将虚拟网 G^1 的 n_1^2 和 n_1^3 都映射到物理节点 n_0^3 ,则用随机方法将 n_1^2 重新映射到属于 $\Omega(n_0^3)$ 的物理节点 n_0^6 (G^1 中必须没有其他节点映射到 n_0^6),从而得到 $A_2 = 17, 2, | 4, 7, 12 |, 2, 20, 23$ 。

然后,计算新染色体(如上文的 A_2 和 B_1)的适应度值,适应度值大于或等于渴望水平的新染色体直接进入下一代;适应度值小于渴望水平但不属于禁忌的新染色体直接进入下一代;适应度值小于渴望水平但属于禁忌的新染色体则选择父代的最好的个体进行入下一代。

最后,用所接受的新染色体或父代中最好的个体更新禁忌表。

(5) 禁忌变异算子

首先,用随机方式对父代个体中的基因进行突变。例如,在某父代染色体中随机选择代表虚拟节点 n_j^i 的元素进行突变,使其取值由原来的 x 突变为不等于 x 且属于 $\Omega(n_j^i)$ 的 y ,从而形成新的染色体。

染色体串中与某一虚拟网对应的子串元素不能出现相同取值(空值除外)。因此,上例中还需要把新染色体串中与虚拟网 G^j 对应的子串中取值为 y 的其他元素(如有的话)再次进行变异,直到与虚拟网 G^j 对应的子串中元素的取值都不相同(空值除外)或到达指定变异深度。如到达指定变异深度,与虚拟网 G^j 对应的子串中元素的取值还存在相同,则取消前

面所有的变异操作;并重新随机选择元素开始变异操作,直到在指定的随机选择次数之内完成变异;如在指定的随机选择次数内没有完成变异,则取消本次变异操作。

然后,计算新染色体(新个体)的适应度值,适应度值大于或等于渴望水平的新染色体直接进入下一代;适应度值小于渴望水平但不属于禁忌的新染色体直接进入下一代;适应度值小于渴望水平但属于禁忌的新染色体则选择父代的最好个体进行入下一代。

最后,用所接受的新染色体或父代中最好的个体更新禁忌表。

4.2 GT_OSVMN 算法流程

GT_OSVMN 算法的具体流程如算法 2 所示。

算法 2 GT_OSVMN 算法

输入:物理网络 $G^0(N^0, E^0)$, 虚拟网集合 $G = \{G^j | j \in R, R = [1, n]\}$, 交叉概率 P_c , 变异概率 P_m , 种群大小 N , 遗传代数 T , 禁忌表长度 L , 变异深度 D , 随机选择次数 C

输出:完成映射的虚拟网子集 R'

1. 随机生成初始种群,大小为 $N-2$;用 G_OSVMN 算法中的虚拟节点映射方法(即使该算法仅完成部分虚拟节点的映射,其结果也放入初始种群)生成 1 个染色体加入种群,使种群大小为 $N-1$;最后增加 1 个取值全空的染色体加入种群,使种群大小为 N 。
2. 用个体适应度函数计算每个个体的适应度 F_i 。具体步骤是:首先基于个体染色体内含的所有虚拟网的虚拟节点映射方案,用 G_OSVMN 算法中的虚拟链路映射方法完成虚拟链路映射(即完成虚拟网的构建),然后计算 F_i ,计算完成后取消链路映射。
3. $count = 0$ 。
4. if($count < T$) {转步骤 12.}
5. 计算当代的渴望水平。
6. 用无放回余数随机选择方式执行选择操作。
7. 按交叉概率 P_c 执行禁忌交叉操作(禁忌表长度为 L)。
8. 按变异概率 P_m 执行禁忌变异操作(变异深度为 D ,禁忌表长度为 L ,随机选择次数为 C)。
9. 用个体适应度函数计算当代每个个体的适应度 F_i ,具体步骤同步步骤 2。
10. $count = count + 1$ 。
11. 转步骤 4。
12. 首先,用个体适应度函数计算当代每个个体的适应度 F_i ;然后,按照适应度 F_i 最大的个体进行相关虚拟网的虚拟节点映射;最后,用 G_OSVMN 算法中的虚拟链路映射方法对已经完成虚拟节点映射的虚拟网进行虚拟链路映射,并构建映射的虚拟网的集合 R' 。

5 算法分析

5.1 时间复杂性分析

求解 K 短路径的算法^[12]的复杂度是 $O(k(m+n \log n))$,其中 m 是边数, n 是顶点数。而 G_OSVMN 算法的时间复杂度由算法 1 的步骤 5.6.2.2 决定,故其时间复杂度为 $(\sum_{j=1}^n E^j) \times k \times (E^0 + N^0 \times \log_2^N)$ 。

GT_OSVMN 算法的时间复杂度由算法 2 的步骤 9 决定,故其时间复杂度为 $T \times N \times (\sum_{j=1}^n E^j) \times k \times (E^0 + N^0 \times \log_2^N)$ 。

5.2 算法平均性能实验分析

一般地,离线静态虚拟网映射问题是不可近似问题,这决定了G_OSVNM算法和GT_OSVNM算法都不是近似算法,故下面通过实验对G_OSVNM算法和GT_OSVNM算法的平均性能进行评估。

5.2.1 对比算法

由于目前对一般离线静态虚拟网映射问题的研究还较少,因此本文将G_OSVNM算法、GT_OSVNM算法与基线算法B_OSVNM(Baseline Offline Static Virtual Network Mapping)进行对比分析。

基线算法B_OSVNM将 n 个虚拟网依序进行映射,在映射某个虚拟网时,采用两阶段映射方法。在虚拟节点映射阶段,各虚拟节点依序映射到符合位置和容量约束的可用CPU资源量最大的物理节点上;在虚拟链路映射阶段,各虚拟链路依序映射到符合容量约束的最短物理路径上。

5.2.2 仿真环境及性能评估指标

实验使用Matlab进行模拟仿真,利用物理节点平均利用率、物理链路平均利用率、物理网提供商平均收益和虚拟网构建成功率(虚拟网构建成功的个数占构建请求数的百分比)这4个指标对G_OSVNM算法和GT_OSVNM算法的平均性能进行评估。

G_OSVNM算法中, ϵ 取 $0.0001^{[13]}$, k 取4, h 取2。GT_OSVNM算法中,交叉概率 P_c 取0.7,变异概率 P_m 取0.1,种群大小 N 取30,遗传代数 T 取30,禁忌表长度 L 取5,变异深度 D 取3,随机选择次数 C 取5。

5.2.3 仿真实验数据的设定

目前,物理网和虚拟网的具体特征还不清晰^[16],本文使用GT-ITM工具,通过常用的随机方法生成10个物理网和5类共10个虚拟网组的实验数据^[17-18]。

每个物理网有50个节点和612条链路,节点CPU容量和链路带宽都在整数区间 $[50,100]$ 内均匀分布,节点位置属性用 x 坐标和 y 坐标表示, x 和 y 都在整数区间 $[1,100]$ 内均匀分布。

第1类的2个虚拟网组都包含60个虚拟网,第2类的2个虚拟网组都包含75个虚拟网,第3类的2个虚拟网组都包含90个虚拟网,第4类的2个虚拟网组都包含105个虚拟网,第5类的2个虚拟网组都包含120个虚拟网。

每个虚拟网的节点数在整数区间 $[2,10]$ 内均匀分布,连通度是50%;节点CPU容量在整数区间 $[1,9]$ 内均匀分布,链路带宽在整数区间 $[1,25]$ 内均匀分布;节点位置属性用 x 坐标和 y 坐标表示, x 和 y 都在 $[1,100]$ 内均匀分布;虚拟网的映射收益等于所有虚拟节点的CPU容量和所有虚拟链路的带宽容量之和;所有虚拟网的虚拟节点位置与所映射的物理节点位置间的距离必须小于或等于10。按照上述设定,形成了60个数据实例,下面针对这60个数据实例的实验结果(按虚拟网组的类别进行算术平均)进行分析。

5.2.4 实验结果及分析

(1)虚拟网构建成功率和物理网提供商映射收益的分析

从图1和图2可以看出,在虚拟网构建平均成功率和物

理网提供商平均收益性能方面,G_OSVNM算法分别比B_OSVNM算法约高出21%和31%。通过分析具体实验数据,发现其原因为:1)B_OSVNM算法将 n 个虚拟网依序进行映射时,出现了在完成低收益的虚拟网构建时占用了部分物理网瓶颈资源的情况,从而导致后续的高收益虚拟网不能完成构建,并且物理网提供商平均收益的增长率高于虚拟网构建平均成功率的增长率,这说明该情况出现的概率较大;2)在单个虚拟网映射时,B_OSVNM算法采用了将虚拟节点依序映射到符合位置和容量约束的可用CPU资源量最大的物理节点上的策略,因而存在将CPU容量需求小的虚拟节点映射到瓶颈物理节点(位置约束导致只能映射到瓶颈物理节点的情况)以及将虚拟节点映射到剩余CPU容量较大但资源可及性较低^[10]的物理节点上的情况,从而导致虚拟网不能完成构建,或因占用较多瓶颈资源导致后续虚拟网构建失败概率增加;3)B_OSVNM算法在单个虚拟网映射时采用的将虚拟链路依序映射到最短物理路径上的策略,存在带宽需求小的虚拟链路占用了瓶颈物理链路并增加了瓶颈物理链路数量的情况,从而导致虚拟网不能完成构建,或因占用较多瓶颈资源导致后续虚拟网构建失败概率增加。这些原因从反面证明了G_OSVNM算法的虚拟网映射顺序策略和虚拟网映射策略的有效性。

从图1和图2中还可以看出,GT_OSVNM算法的虚拟网构建平均成功率和物理网提供商平均收益分别比G_OSVNM算法约高出11%和9%,其主要原因是GT_OSVNM算法将G_OSVNM算法的解作为一个染色体放入初始种群,且通过迭代优化求解离线静态虚拟网映射问题的映射方案。物理网提供商平均收益的增长率低于虚拟网构建平均成功率的增长率,这说明拒绝部分将占用较多物理网瓶颈资源的高收益虚拟网,有助于提高物理网提供商的收益。

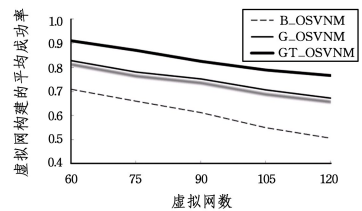


图1 虚拟网构建的平均成功率

Fig. 1 Average success rate of virtual network construction

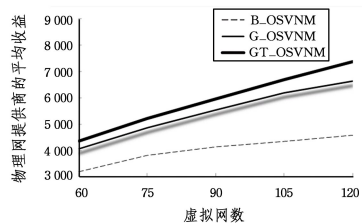


图2 物理网提供商的平均收益

Fig. 2 Average revenue of physical network provider

(2)物理网资源利用情况的分析

从图3和图4可以看出,G_OSVNM算法的物理节点和物理链路的平均利用率分别比B_OSVNM算法约高出19%

和 12%。结合图 1 和图 2 可知,G_OSVMN 算法平均利用率高的原因是其有更高的虚拟网构建成功率和物理网提供商平均收益。

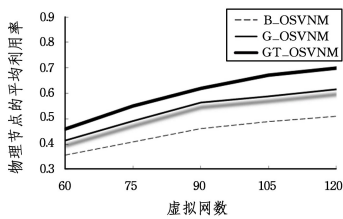


图 3 物理节点的平均利用率

Fig. 3 Average utilization of physical nodes

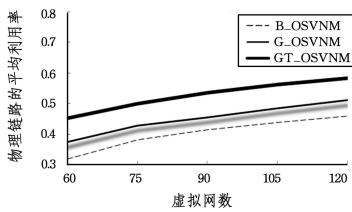


图 4 物理链路的平均利用率

Fig. 4 Average utilization of physical links

同样,从图 3 和图 4 可以看出,GT_OSVMN 算法的物理节点和物理链路的平均利用率分别比 G_OSVMN 算法约高出 12%和 16%。结合图 1 和图 2 可知,GT_OSVMN 算法平均利用率高的原因是其有更高的虚拟网构建成功率和物理网提供商平均收益。

结束语 针对已有离线静态虚拟网映射问题的求解算法不适用于中大规模的一般离线静态虚拟网映射问题的现状,本文首先基于贪婪策略完成 G_OSVMN 算法的设计;然后,在 G_OSVMN 算法的基础上,基于混合优化策略完成禁忌遗传 GT_OSVMN 算法的设计;最后,基于中大规模的一般离线静态虚拟网映射问题的实例,对 G_OSVMN 算法和 GT_OSVMN 算法进行了仿真实验,结果表明所提出的算法具有较好的平均性能和较强的实用性。当然,所提算法还有许多问题有待进一步研究,如算法平均性能的理论分析,如何根据虚拟网建立请求模式自适应调整算法,如何修正算法使之适用于离线动态虚拟网映射问题等。

参考文献

[1] CHOWDHURY N M M K, BOUTABA R. A survey of network virtualization[J]. *Computer Networks*, 2010, 54(5): 862-876.

[2] YE Z L, ZHU Y Q, JI P N, et al. Virtual infrastructure mapping in software-defined elastic optical networks[J]. *Photonic Network Communications*, 2017, 34(1): 34-44.

[3] FISCHER A, BOTERO J F, BECK M T, et al. Virtual Network Embedding: A Survey[J]. *IEEE Communications Surveys and Tutorials*, 2013, 15(4): 1888-1906.

[4] 余建军. 虚拟网映射问题及算法研究[M]. 杭州: 浙江大学出版社, 2018.

[5] YU J J, WU C M. Computational Complexity Analysis of Virtual Network Mapping Problem[J]. *Computer Science*, 2018, 45(11): 87-91. (in Chinese)

余建军, 吴春明. 虚拟网映射问题的计算复杂性分析[J]. *计算机科学*, 2018, 45(11): 87-91.

[6] AMALDI E, CONIGLIO S, KOSTER A M C A, et al. On the computational complexity of the virtual network embedding problem[J]. *Electronic Notes in Discrete Mathematics*, 2016, 52(3): 213-220.

[7] ROST M, SCHMID S. Charting the Complexity Landscape of Virtual Network Embeddings[C]// *Proceedings IFIP Networking*. Zurich, Switzerland, 2018: 1-9.

[8] ZHANG Z, SU S, ZHANG J, et al. Energy aware virtual network embedding with dynamic demands: Online and offline [J]. *Computer Networks*, 2015, 93(P3): 448-459.

[9] CONIGLIO S, KOSTER A, TIEVES M. Data Uncertainty in Virtual Network Embedding: Robust Optimization and Protection Levels[J]. *Journal of Network & Systems Management*, 2016, 24(3): 681-710.

[10] CONIGLIO S, GRIMM B, KOSTER A M C A, et al. Optimal offline virtual network embedding with rent-at-bulk aspects[J]. *Computer Science*, 2015, 14(2): 1-9.

[11] ROST M, SCHMID S. (FPT-) Approximation Algorithms for the Virtual Network Embedding Problem[EB/OL]. (2018-03-12)[2018-10-25]. <https://arxiv.org/abs/1803.04452>.

[12] KATO H, IBARAKI T, MINE H. An efficient algorithm for K shortest simple paths[J]. *Networks*, 1982, 12(4): 411-427.

[13] CHENG X, SU S, ZHANG Z, et al. Virtual network embedding through topology-aware node ranking [J]. *ACM SIGCOMM Computer Communication Review*, 2011, 41(2): 38-47.

[14] GLOVER F, KELLY J P, LAGUNA M. Genetic algorithms and tabu search: Hybrids for optimization[J]. *Computers & Operations Research*, 1995, 22(1): 111-134.

[15] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1999.

[16] CHOWDHURY M, RAHMAN M R, BOUTABA R. ViNE-Yard: virtual network embedding algorithms with coordinated node and link mapping[J]. *IEEE/ACM Transactions on Networking*, 2012, 20(1): 206-219.

[17] SU Y Z, MENG X R, MENG Q W, et al. Environment Adaptive and Joint Topology Aware Virtual Network Embedding Algorithm[J]. *Journal of Electronics & Information Technology*, 2018, 40(1): 79-86. (in Chinese)

苏玉泽, 孟相如, 孟庆微, 等. 环境自适应的拓扑联合感知虚拟网映射算法[J]. *电子与信息学报*, 2018, 40(1): 79-86.

[18] YU J J, WU C M. Virtual Network Mapping Strategy and Competitive Analysis Based on Cost Constraint[J]. *Telecommunications Science*, 2016, 32(2): 47-54. (in Chinese)

余建军, 吴春明. 基于成本约束的虚拟网映射策略及竞争分析[J]. *电信科学*, 2016, 32(2): 47-54.