

doi:10.3969/j.issn.1001-893x.2015.08.016

引用格式:张晓红,黄继海.网络功能虚拟化中的性能感知服务组合优化[J].电讯技术,2015,55(8):919-925. [ZHANG Xiaohong, HUANG Jihai. Performance-Aware Service Composition Optimization in Network Function Virtualization[J]. Telecommunication Engineering, 2015, 55 (8): 919-925.]

网络功能虚拟化中的性能感知服务组合优化*

张晓红**,黄继海

(中州大学 信息工程学院,郑州 450005)

摘 要:服务功能链是网络功能虚拟化的重要支撑,为了构建满足功能和性能需求的服务功能链,需要建立服务的性能模型,从而产生基于性能的服务组合优化问题。一种基于性能模型的服务组合优化问题被建模,并针对复杂约束情况下的无效解干扰,提出了改进的模拟退火算法,该算法包含基于层次属性的产生函数和基于偏离度的目标函数。仿真结果表明,该算法提高了 21% 的服务组合成功率,同时降低了组合成本和时间消耗。仿真结果验证了所提算法的有效性。

关键词:网络功能虚拟化;服务功能链;服务组合;性能模型;模拟退火

中图分类号:TN915.8 **文献标志码:**A **文章编号:**1001-893X(2015)08-0919-07

Performance-Aware Service Composition Optimization
in Network Function Virtualization

ZHANG Xiaohong, HUANG Jihai

(Information Engineering College, Zhongzhou University, Zhengzhou 450005, China)

Abstract: Service function chain is a great support to Network Function Virtualization (NFV). To construct a service function chain satisfying function and performance demands, a performance model is required for the service entities, and then the performance-aware service composition optimization problem (SCOP) is coming. A performance model is proposed to analyze the SCOP, and an improved Service Composition Mode search algorithm based on simulated annealing (SCM-SA) is designed to lighten the impact of invalid solutions which do not satisfy the complex constraints. The SCM-SA algorithm contains a generation function based on level attribute and a cost function based on offset degree. Simulation results show that the SCM-SA algorithm increases the service composition success rate (CSR) by 21% while decreasing the composition cost and time cost. Simulation results confirm the validity of the proposed algorithm.

Key words: network function virtualization; service function chain; service composition; performance model; simulated annealing

1 引言

对于运营商而言,提供网络服务的成本越来越高。考虑到当前网络的复杂性,即使是针对突发需求而实施的简单升级,也可能需要数月甚至数年的时间才能完成。然而,在计算技术的进步和 IT 环境中虚拟技术成功的鼓舞下,网络运营商正努力推动

类似的技术,希望以此作为解决扩展性问题的一种办法。网络功能虚拟化(Network Function Virtualization, NFV)^[1]便是此类选择中的一种。它利用标准的 IT 虚拟化技术,将多种类型的网络设备合并到业界标准的高容量服务器、交换机和存储设备中,并将其置于数据中心、网络节点等地。

* 收稿日期:2015-01-27;修回日期:2015-07-07 Received date:2015-01-27;Revised date:2015-07-07

** 通讯作者:zhongzhouzhx@126.com Corresponding author:zhongzhouzhx@126.com

• 919 •

NFV 的重要技术支撑是服务功能链 (Service Function Chain, SFC) 的构建。通过将特定的网络服务抽象为一系列有序的报文处理功能实体, 可以实现新网络服务的灵活部署与服务隔离。然而, 现有研究对 NFV 中的 SFC 构建技术依然停留在资源分配^[2-3]和标准建立^[4-5]阶段, 无法实现 SFC 的服务最优组合。

在计算机网络中, 针对服务组合优化问题的研究主要集中在基于构件的软件工程 (Component-Based Software Engineering, CBSE) 和面向服务体系结构 (Service-Oriented Architecture, SOA) 等领域。虽然一个抽象的服务功能对应的所有具体服务实体在功能上是等价的, 但在其他属性上是存在差异的, 所以可以根据非功能属性进行服务实体选择和组合优化。在不同领域, 非功能属性具有不同含义, 基于这些属性约束建立服务组合优化模型并求解, 已成为 CBSE 和 SOA 中的热点问题。比如 QoS 感知^[6]的 Web Services 组合问题的研究, 文献^[7-8]针对顺序、并发、分支、循环等 4 种控制结构的 Web Services 组合模型, 建立了包括费用、响应时间、信誉、可用性等参数的评价体系来评价组合的效果, 并利用整数规划方法从全局角度进行最优化解。文献^[9-10]的研究思路与文献^[7-8]类似, 提出了遗传算法来解决 QoS 感知的 Web Services 组合问题。此外, 文献^[11-13]以云计算系统为背景, 建立基于 QoS 的服务组合模型, 并提出各种启发算法。文献^[14]将信任概念和信任度引入到 Web Services 组合中, 分别基于信任感知和信任增强建立数学模型, 并采用蚁群算法求解之。

上述研究主要针对 Web Services 的组合优化问题, 其考虑的 QoS 属性并不适用于 NFV 体系, 且忽略了属性的匹配约束。为此, 本文提出一种适合于 NFV 架构的服务实体性能模型, 并根据服务实体间的性能匹配和资源总量约束, 形式化分析服务组合优化问题; 在求解过程中, 为减轻大量无效解的干扰, 提出了改进的模拟退火算法 SCM-SA, 该算法包含基于层次属性的产生函数和基于偏离度的目标函数。

本文的后续部分组织如下: 第 2 节提出一种针对服务实体的性能模型, 并在此基础上分析基于性能的服务组合优化问题; 第 3 节提出改进的模拟退火算法 SCM-SA 及其相关函数设定; 第 4 节对提出的 SCM-SA 算法进行实验评估; 第 5 节为结束语。

2 问题描述

2.1 服务功能链模型

定义 1 服务拓扑。在 NFV 体系中, 服务功能链可以抽象为服务拓扑 $G^c = (V, E, s, t)$ 。其中, V 为服务功能集合, E 为服务间的连接关系, s 为该服务功能链的源节点, t 为该服务功能链末节点。

服务功能链的构建过程分为两步: 服务拓扑映射和服务组合优化。对于第一步, 我们通过虚拟网映射技术^[2-3], 将服务拓扑嵌入到底层网络中, 并为其分配节点资源和带宽资源。对于第二步, 我们需要描述服务组合的实体和约束条件。

定义 2 服务实体。提供具体服务的功能实体称之为服务实体。每个服务对应大量的服务实体, 虽然它们的服务属性相同, 但是具有多样化的性能参数, 所以我们将服务 i 的第 j 个服务实体抽象为

$$e_{ij} = (\text{Config}_{\text{in}}, \text{Res}, \text{Config}_{\text{out}}),$$

其中:

$$\text{Config}_{\text{in}} = [c_1^{\text{in}}, c_2^{\text{in}}, \dots, c_m^{\text{in}}],$$

$$\text{Config}_{\text{out}} = [c_1^{\text{out}}, c_2^{\text{out}}, \dots, c_m^{\text{out}}],$$

$$\text{Res} = [r_1, r_2, \dots, r_k].$$

式中, $\text{Config}_{\text{in}}$ 为服务实体的输入性能矢量, $\text{Config}_{\text{out}}$ 为服务实体的输出性能矢量, Res 为服务实体运行时所占用的系统资源矢量, k 为所考虑的资源类型数, c_i^{in} 和 c_i^{out} 分别为第 i 个输入性能属性和输出性能属性, m 为所考虑的性能属性个数。特别地, 若 c_i^{in} 为 ϕ , 则说明服务实体不受该性能属性的约束; 若 c_i^{out} 为 ϕ , 则说明服务实体不对外提供该性能属性的信息。

性能属性是指服务实体在运行时所需要的或表现出的某一性能方面的特性。例如某一报文分类实体在运行时输入报文的到达速率需小于 10 Gb/s, 而输出的分类结果速率为 125 Mb/s。

两个同种类型的性能属性 c_1 和 c_2 进行叠加, 记为 $c_1 \oplus c_2$ 。根据性能属性类型不同, 叠加符号 $\oplus$ 对应的数学运算可以是交集运算、并集运算、求和运算或最小 (大) 值运算等。例如, 当性能属性为传输速率时, $c_1 \oplus c_2 = c_1 + c_2$; 当性能属性为最大时延时, $c_1 \oplus c_2 = \max(c_1, c_2)$; 当性能属性为执行时间时, c_1 和 c_2 均为取值区间, $c_1 \oplus c_2 = c_1 \cup c_2$ 。特别地, 若 $c_2 = \phi$, 则 $c_1 \oplus c_2 = c_1$; 若 $c_1 = \phi$, 则 $c_1 \oplus c_2 = c_2$ 。

定义 3 性能矢量叠加。两个性能矢量 Config_1 和 Config_2 进行叠加, 记为 $\text{Config}_1 \oplus \text{Config}_2 = [c_{11},$

$c_{12}, \dots, c_{1m}] \oplus [c_{21}, c_{22}, \dots, c_{2m}]$ 。具体表达式为
 $\text{Config}_1 \oplus \text{Config}_2 = [c_{11} \oplus c_{21}, c_{12} \oplus c_{22}, \dots, c_{1m} \oplus c_{2m}]$ 。

2.2 服务组合优化

服务组合优化问题就是从每个服务 i 对应的候选服务实体集合 S_i 中挑选出合适的实体 e_{ij} , 从而得到成本最优的服务组合模式 SCM_{best} , 且满足性能约束和资源约束。图 1 中的 $\text{SCM}_1 = \{e_{11}, e_{21}, e_{31}, e_{41}, e_{51}\}$ 和 $\text{SCM}_2 = \{e_{13}, e_{22}, e_{32}, e_{42}, e_{52}\}$ 即为两种可能的服务组合模式。

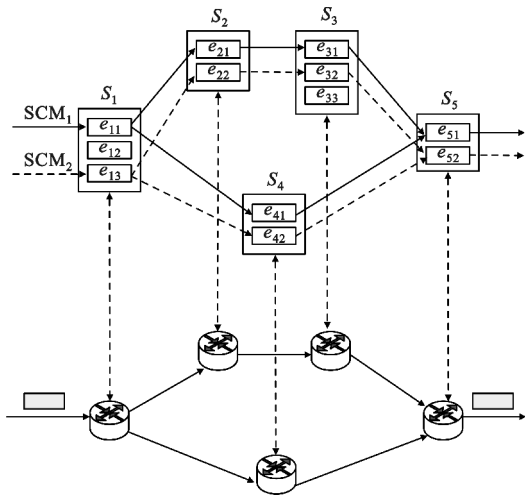


图 1 服务组合优化问题

Fig. 1 Service composition optimization problem

服务组合优化问题的关键在于使所有服务实体的性能需求得到满足。

定义 4 性能满足。若服务实体 e_{ij} 的输入性能矢量 $\text{Config}_{\text{in}}$ 被所依赖的服务实体 $\{e_1, e_2, \dots, e_l\}$ 所满足, 则下式成立:

$$\begin{cases} e_{ij} \cdot \text{Config}_{\text{in}} \cdot c_k \subseteq \bar{C} \cdot c_k \\ c_k \neq \emptyset, k = 1, 2, \dots, m \end{cases} \quad (1)$$

式中, $\bar{C} = e_1 \cdot \text{Config}_{\text{out}} \oplus e_2 \cdot \text{Config}_{\text{out}} \oplus \dots \oplus e_l \cdot \text{Config}_{\text{out}}$ 。式(1)可以简单地用符号表示为 $e_{ij} \cdot \text{Config}_{\text{in}} \simeq \bar{C}$, 符号 $\simeq$ 表示性能矢量间的满足关系。令 $\text{Satisfied}(\cdot)$ 为判别函数, 若服务实体 e_{ij} 的性能需求得到满足, 则 $\text{Satisfied}(e_{ij}) = 1$, 反之为零。

此外, 在虚拟网构建过程中, $\text{Resource} = [R_1, R_2, \dots, R_k]$ 是服务组合的资源约束条件; 服务组合后的服务功能链也需要满足虚拟网的整体性能需求 $\text{Config}_{\text{req}} = [c_1, c_2, \dots, c_m]$, 所以末节点候选服务实体的输出性能矢量需要与 $\text{Config}_{\text{req}}$ 相匹配。

综合上述设定, 服务组合优化问题可以描述为:

寻找一个服务组合模式 $\text{SCM}_{\text{best}} = \{e_{1a_1}, e_{2a_2}, \dots, e_{na_n}\}$, n 表示候选实体集合个数, a_i 表示第 i 个候选服务实体集合中选择的实体序号, $1 \leq i \leq n, 1 \leq a_i \leq M$, 使其满足

- (1) 除源节点外, 其他服务实体的性能需求均得到满足;
- (2) 末节点的输出性能矢量满足服务功能链的整体性能需求 $\text{Config}_{\text{req}}$;
- (3) 组合模式中所有实体占用的资源不超过虚拟网限定的资源阈值 Resource ;
- (4) 在满足上述 3 个条件下, 选取成本最低的服务组合模式。

形式化描述如下:

$$\begin{aligned} \min_{a_1, a_2, \dots, a_n} \sum_{i=1}^n \left(\sum_{j=1}^k (w_j \cdot e_{ia_i} \cdot \text{Res. } r_j) + p_i \right), \quad (2) \\ \begin{cases} \text{Satisfied}(e_{ia_i}) = 1, i \neq 1 \\ e_{na_n} \cdot \text{Config}_{\text{out}} \simeq \text{Config}_{\text{req}} \\ \sum_{i=1}^n e_{ia_i} \cdot \text{Res. } r_j \leq \text{Resource. } R_j \\ i = 1, 2, \dots, n \\ j = 1, 2, \dots, k \\ a_i = 1, 2, \dots, M \end{cases} \quad (3) \end{aligned}$$

式(2)表示组合模式中所有服务实体的总成本, 而每个服务实体的成本 = 所占各种资源的成本 + 实体的固有成本, 其中 w_j 为资源属性 r_j 的单位成本; p_i 为候选服务实体 e_{ia_i} 的固有成本, 为了便于描述, 式(3)中将 e_{1a_1} 作为源节点、 e_{na_n} 作为末节点, 分别描述了上文中(1)、(2)和(3)三种约束条件; M 为候选服务实体集合大小的上界。

服务组合优化问题是一个具有复杂约束的整数规划问题, 其复杂性为 NP-hard。如果采用遍历方式, 则最坏情况下搜索的模式数为 $O(M^n)$, 模式的有效性判别时间复杂度为 $O(n^2)$, 所以总的时间复杂度为 $O(M^n n^2)$, 仅适用于小规模服务组合。为适应不同规模的服务组合, 需要采用启发算法。

现有启发式算法如模拟退火算法、遗传算法、粒子群算法等主要采用罚函数方法或者松弛方法消除约束条件, 但考虑到本文提出的服务组合优化问题的约束条件属于整数约束而且较为复杂, 上述方法均难以保证能以较大概率收敛到有效解。因此, 本文提出一种改进的启发式算法, 能够有目的地产生新解, 从而提高算法收敛到有效解的概率。然而, 遗传算法和粒子群算法的新解产生过程相对受限,

因此后文对模拟退火算法进行改进,利用解的性能需求约束的分层特性,设计适合服务组合优化问题的新解产生函数和目标函数。

3 服务组合优化算法

模拟退火算法是一种适合解决 NP-hard 问题的通用有效的全局优化算法,其基本思想是从任意解出发,每次产生一个邻居解,直接接受使目标函数更优的邻居解,或以一定概率接受使目标函数变差的邻居解,不断迭代上述过程,最终得到使目标函数全局最优的近似解。采用改进的模拟退火算法解决服务组合优化问题,关键在于确定新解的产生函数(Generation Function)和目标函数(Cost Function)的计算公式,以克服无效解的干扰。

3.1 基于层次属性的产生函数

首先,需要对问题的解进行编码。服务组合优化问题的解(即服务组合模式)使用了一个整数数组来表示,数组的长度为组成服务功能链的服务实体数 n ,数组中的每个元素为该服务实体所对应的索引号,比如服务组合模式 $SCM_i = \{e_{13}, e_{21}, e_{32}, e_{42}, e_{54}\}$ 可以被编码为 $[3, 1, 2, 2, 4]$ 的数组。然后,算法的初始化过程就是对数组中的元素赋予随机的正整数值,并且取值范围不超过相应候选服务实体集合的大小。

为了更好地逼近有效解,模拟退火算法设计了具有记忆性的新解产生函数。所谓有记忆性是指新解产生函数能够记录当前解的层次属性(见定义 5),并且根据具体情况产生三种类型的新解:具有相同层次属性的新解;具有更高层次属性的新解;具有更低层次属性的新解。下面给出解的层次属性定义。

定义 5 解的层次属性。根据图论中的反向拓扑排序算法,解数组中的每个元素所代表的候选服务实体都存在唯一的层次编号。新解是通过随机改变旧解数组中的若干元素得到的,并且这些元素所代表的候选服务实体具有相同的层次编号,所以每个解的产生都与一个层次编号相关联,该编号称为解的层次属性。

在对候选服务实体的层次编号进行预计算后,新解产生函数的处理流程如下所示:

PROCEDURE GenFunc(M_{old}):

Input:

M_{old} : 当前解数组;

Output:

M_{new} : 新产生的解数组;

if level > 最大层次数 then

$M_{new} \leftarrow$ 随机改变 M_{old} 中一个数组元素 $M_{old}[i]$;

level $\leftarrow M_{old}[i]$ 对应服务实体的层次编号;

return M_{new} ;

end if

CurEntities $\leftarrow M_{old}$ 中层次为 level 的候选服务实体集合;

preEntities $\leftarrow M_{old}$ 中层次小于 level 的候选服务实体集合;

Conflict $\leftarrow$ CurEntities 中与 preEntities 的服务实体存在性能依赖关系冲突的服务实体集合;

if Conflict is empty then

level $\leftarrow$ level + 1;

return GenFunc(M_{old});

else

for each e in Conflict do

$i \leftarrow e$ 所对应服务实体的索引;

随机改变 $M_{old}[i]$;

end for

$M_{new} \leftarrow M_{old}$;

return M_{new} ;

end if

GenFunc 函数每次只针对首先出现性能约束不匹配的一层进行随机改变生成新解;如遇有效解,则随机从某一层开始新解的生成。经过逐层推进,解的性能约束条件逐渐得到满足,从而趋近有效解。这种方式有利于避免普通随机生成方式的盲目性,提高有效解的搜索效率。

3.2 基于偏离度的目标函数

对于当前解 M_{old} 和新产生的解 M_{new} ,若目标函数值 $f(M_{new}) < f(M_{old})$,则直接接受 M_{new} 为当前解;若 $f(M_{new}) \geq f(M_{old})$,则依接受概率接受 M_{new} 。接受概率 $P(M_{old}, M_{new}, t)$ 的计算公式如下:

$$P(M_{old}, M_{new}, t) = e^{-[f(M_{new}) - f(M_{old})] / t} \quad (4)$$

根据公式(4),定义如下的目标函数计算公式:

$$f(M) = \begin{cases} \text{Cost}(M), & M \text{ 为有效解} \\ (1 + \text{Conflict.size}()) \cdot \text{Cost}(M), & M \text{ 为无效解} \end{cases} \quad (5)$$

式(5)中的 Cost 函数用来计算当前解(服务组合模式)所消耗的成本值,具体见公式(2);Conflict.size()表示当前层次的服务实体集合中违反与上层服务实体性能依赖关系的个数,1 + Conflict.size()即为解的偏离度。

3.3 基于模拟退火的服务组合模式搜索

基于上述函数设计,可以得到基于模拟退火的服务组合模式搜索算法 (Service Composition Mode search algorithm based on Simulated Annealing, SCM-SA) 的完整处理流程如下:

```
生成初始解  $M_0$ ;  
当前解  $M \leftarrow M_0$ , 并计算其目标函数值  $f(M)$ ;  
初始化参数: 初始温度  $T_s$ , 终止温度  $T_e$ , 降温比例  $a$ , 每个温度下迭代次数  $L$ , 收敛条件  $R$ ;  
当前解的层次属性 level = 1, 当前温度  $t = T_s$ ;  
对服务拓扑进行反向排序, 得到层次编号;  
while  $t > T_e$  do  
     $j \leftarrow 0$ ;  
    while  $j < L$  do  
         $M_{new} \leftarrow \text{GenFunc}(M)$ ;  
        if  $f(M_{new}) < f(M)$  then  
             $M \leftarrow M_{new}$ ;  
        else  
            以概率  $P(M, M_{new}, t)$  接受  $M_{new}$  为当前解;  
        end if  
         $j \leftarrow j + 1$ ;  
    end while  
     $t \leftarrow t * a$ ;  
    if  $M$  连续  $R$  次不变 then  
        return  $M$ ;  
    end if  
end while
```

3.4 复杂度分析

由于 SCM-SA 算法是基于模拟退火算法, 因此新解的迭代产生次数是受参数影响的, 如果将算法参数视作常数, 则迭代次数的复杂度是 $O(1)$ 。对于新解产生函数 GenFunc, 其计算复杂性主要体现在第 8 步“while $j < L$ do”中, 其复杂度为 $O(n^2)$, 而该步的执行次数由服务拓扑的最大节点度 D 和最大层次数 H 决定, 最坏情况下次数为 $O(D+H)$, 因此函数 GenFunc 的最坏情况复杂度为 $O(n^2(D+H))$ 。

4 实验评估

4.1 实验设置

通过 C++ 编程模拟服务组合优化问题并实现模拟退火算法。实验方案主要为了评估 SCM-SA 算法相比于其他启发式算法在服务组合成功率和组合代价上的优势。服务组合成功率 CSR 是指所有

服务组合请求中成功的比例。

在实验数据上, 利用 TGFF^[15] 工具随机生成具有指定节点数的服务拓扑, 且只考虑有向无环图 (DAG)。利用 C++ 语言生成候选服务实体集合, 其中每个候选服务实体集合大小在 $[3, 10]$ 内均匀分布。每个候选服务实体的性能矢量长度为 3, 且性能属性的取值为 0 或 1, 叠加符号 $\oplus$ 对应最小值运算, 其中 1 的概率为 90%。候选服务实体的资源属性个数为 3, 且取值在 $[10, 100]$ 内均匀分布, 各资源属性的单位成本 w_1, w_2 和 w_3 分别设为 1.0、1.0 和 1.0, 总量均设为 4000。此外, 候选服务实体的固有成本取值在 $[10, 15]$ 内均匀分布。

4.2 性能评估

实验中, SCM-SA 算法的主要参数取值如下: 初始温度 $T_s = 1000$, 终止温度 $T_e = 10$, 降温比例 $a = 0.9$, 每个温度下迭代的次数 $L = 5 \times n$ (n 为服务拓节点数), 收敛条件 $R = 50$ 。在不同服务组合规模下 ($n = 10, 15, 20, 25, 30, 35, 40$), 分别进行 60 次服务组合问题求解, 计算不同算法的服务组合成功率 CSR, 结果如图 2 所示。可以看出, 随着服务组合规模的扩大, 服务间的性能依赖关系更加复杂化, 从而使各种算法的 CSR 值逐渐降低。其中, SCM-SA 算法的 CSR 值保持在 46.7% 以上, 高于其他算法; 单纯考虑有效解的模拟退火算法 Feasible-SA 的 CSR 值始终低于 SCM-SA 算法, 且随组合规模 n 的扩大而下降至 16.7%; 不考虑解有效性的基本遗传算法 GA 的 CSR 值最低, 不超过 10%。上述结果表明, 单纯考虑有效解方式会使部分搜索陷入僵局, 降低 CSR 值, 基本启发式算法 (GA) 则很难在复杂约束条件下收敛到有效解, 所以 CSR 值最低。

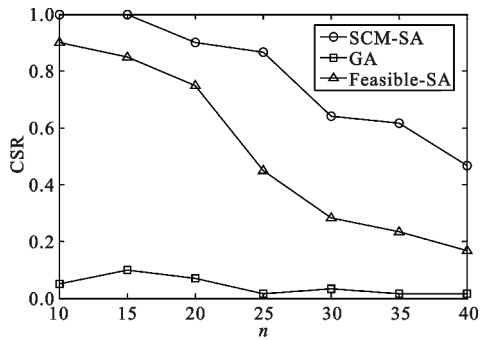


图 2 不同算法的服务组合成功率 (CSR) 比较
Fig.2 CSR comparison among different algorithms

相同实验环境下, 比较不同算法的最优服务组

合模式成本值,如图 3 所示。从实验结果可以看出,不同算法的服务组合模式成本值的大小关系为 $C_{SCM-SA} < C_{Feasible-SA} < C_{GA}$,其中 Feasible-SA 和 GA 算法的成本值比较接近。随着服务组合规模 n 的扩大,不同算法的成本值差距逐渐增大。上述结果表明,单纯考虑有效解方式会使结果陷入局部最优,而基本启发式算法又易受无效解的干扰,从而使算法得到的成本值偏大。

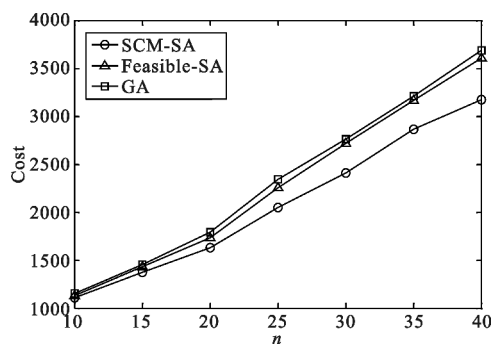


图 3 不同算法得到的最优组合代价

Fig. 3 Composition cost comparison among different algorithms

比较不同算法的时间消耗,如图 4 所示。从实验结果可以看出,基本遗传算法 GA 的时间消耗最小且波动不明显,单纯考虑有效解的模拟退火算法 Feasible-SA 的时间消耗最大,且随问题规模呈非线性增长,而 SCM-SA 算法的时间消耗低于 Feasible-SA,原因在于,GA 的新解产生函数跟问题规模无关,而 Feasible-SA 和 SCM-SA 的新解产生函数跟问题规模有关,并且 Feasible-SA 只考虑有效解,导致调用新解产生函数的次数多于 SCM-SA,因此时间消耗大于 SCM-SA。

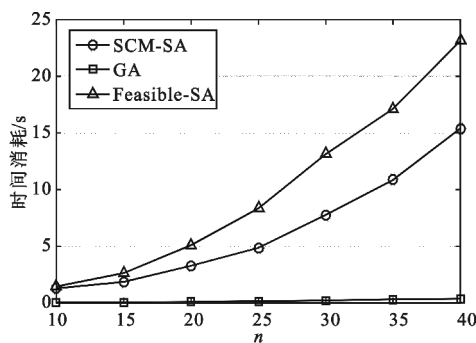


图 4 不同算法的时间消耗

Fig. 4 Time cost comparison among different algorithms

上述实验在不同组合规模和仿真数据下重复进行,结果表明 SCM-SA 算法在服务组合成功率和服

5 结束语

本文对网络功能虚拟化中的服务功能链构建和服务组合优化问题进行了论述。首先,针对服务实体的性能描述需求和匹配约束特性,给出了一种服务实体性能模型,并在此基础上形式化分析服务组合优化问题。然后,为减轻大量不满足性能约束的无效解对搜索过程的干扰,提出了改进的模拟退火算法 SCM-SA。最后,通过仿真实验验证了 SCM-SA 算法在服务组合成功率和组合成本上具有明确的优势。

本文对单一服务组合请求下的优化问题进行了研究,而实际的服务组合请求可以是在线到达的,所以下一步需对在线模式下的服务组合优化问题进行研究。

参考文献:

- [1] Network Functions Virtualisation(NFV). Network Operator Perspectives on Industry Progress, ETSI [EB/OL]. 2014-04-01 [2015-01-05]. http://portal.etsi.org/NFV/NFV_White_Paper2.pdf.
- [2] Yu M, Yi Y, Rexford J, et al. Rethinking virtual network embedding: substrate support for path splitting and migration[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 17-29.
- [3] Chowdhury M, Rahman M 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.
- [4] Boucadair M, Jacquenet C, Parker R, et al. Service Function Chaining: Framework & Architecture[EB/OL]. 2014-08-16 [2015-01-05]. <http://tools.ietf.org/html/draft-boucadair-sfc-framework-02>.
- [5] Quinn P, Beliveau A. Service Function Chaining(SFC) Architecture[EB/OL]. 2014-02-04 [2015-01-05]. <http://tools.ietf.org/html/draft-merged-sfc-architecture-01>.
- [6] Menasce D. QoS issues in Web service[J]. Internet Computing, 2002, 6(6): 72-75.
- [7] Zeng L, Benatallah B, Ngu A H H, et al. Qos-aware middleware for web services composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327.
- [8] Zeng L, Benatallah B, Dumas M, et al. Quality driven web services composition[C]//Proceedings of the 12th International Conference on World Wide Web. New York: ACM, 2003: 411-421.
- [9] 张成文, 苏森, 陈俊亮. 基于遗传算法的 QoS 感知的 Web 服务选择[J]. 计算机学报, 2006, 29(7): 1029-1037.

ZHANG Chengwen, SU Sen, CHEN Junliang. Genetic algorithm on web services selection supporting QoS[J].

- Chinese Journal of Computers, 2006, 29 (7) : 1029 – 1037. (in Chinese)
- [10] 蒋哲远,韩江洪,王钊. 动态的 QoS 感知 Web 服务选择和组合优化模型[J]. 计算机学报,2009(5):1014–1025. JIANG Zheyuan, HAN Jianghong, WANG Zhao. An Optimization Model for Dynamic QoS-Aware Web Services Selection and Composition[J]. Chinese Journal of Computers, 2009(5): 1014–1025. (in Chinese)
- [11] Ma H, Bastani F, Yen I L, et al. QoS-driven service composition with reconfigurable services[J]. IEEE Transactions on Services Computing, 2013, 6(1): 20–34.
- [12] Xiang F, Hu Y, Yu Y, et al. QoS and energy consumption aware service composition and optimal – selection based on Pareto group leader algorithm in cloud manufacturing system[J]. Central European Journal of Operations Research, 2014, 22(4): 663–685.
- [13] Tao F, Lai Li Y, Xu L, et al. FC-PACO-RM: a parallel method for service composition optimal – selection in cloud manufacturing system[J]. IEEE Transactions on Industrial Informatics, 2013, 9(4): 2023–2033.
- [14] 王勇,代桂平,侯亚荣. 信任感知的组合服务动态选择方法[J]. 计算机学报,2009,32(8): 1668–1675.
- WANG Yong, DAI Guiping, HOU Yarong. Dynamic methods of trust-aware composite service selection[J]. Chinese Journal of Computers, 2009, 32 (8) : 1668 – 1675. (in Chinese)
- [15] Dick R P, Rhodes D L, Wolf W. TGFF: task graphs for free [C]//Proceedings of the 6th International Workshop on Hardware/Software Codesign. Seattle: IEEE, 1998: 97–101.

作者简介:



张晓红(1979—),女,河南舞钢人,2008年获硕士学位,现为讲师,主要从事计算机网络方面的研究;

ZHANG Xiaohong was born in Wugang, Henan Province, in 1979. She received the M. S. degree in 2008. She is now a lecturer. Her research concerns computer network.

Email: zhongzhouzxxh@126.com

黄继海(1977—),男,河南濮阳人,2008年获硕士学位,现为副教授,主要从事计算机网络方面的研究。

HUANG Jihai was born in Puyang, Henan Province, in 1977. He received the M. S. degree in 2008. He is now an associate professor. His research concerns computer network.