

# 基于多背包模型的分布式协同网络服务布局算法

**摘要:** 提出了一种基于多背包模型的分布式协同网络服务布局算法, 通过网络各节点协同获取全局网络状况, 将网络服务实例的布局问题建模为多背包问题, 求解得到网络服务实例的最优布局。仿真结果表明网络服务在网络中的合理布局可以最大化网络效益、提高节点效用比和响应率, 有助于物联网及其应用的良好运行。

**关键词:** 物联网; 服务布局; 多背包问题

**中图分类号:** TN929.5

**文献标识码:** A

## A Service Layout Algorithm for the Distributed Collaborative Network Based on Multi-Knapsack Model

**Abstract:** A distributed collaborative network service layout algorithm based on multi-knapsack model was proposed. The layout problem of network service instance is modeled as multi-knapsack problem, and the global network condition is obtained by network nodes to obtain the optimal layout of network service instance. The simulation results show that the reasonable layout of the network service in the network can maximize the network efficiency, improve the utility ratio and response rate of the node and help the Internet of things and its application well run.

**Key words:** Internet of things; service layout; multi-knapsack problem

### 1 引言

近些年来, 物联网快速发展, 机器与机器(machine-to-machine, M2M)通信是一种不在人为干扰下机器与机器间直接进行通信的一种方式[1], 物联网终端的数量也呈指数增长, 并且类型众多。各种业务服务在全局网络中存在有多个实例, 所以服务实例的布局优化对于提升服务质量和全局网络效益具有很高的现实意义, 然而海量的 IoT 终端设备将导致 SDN 交换机需要不断扩大流表, 设备持续进出网络时在交换机和控制器之间的通信开销增大; 并且物联网业务的类型具有多样性, 且服务实例的总体规模较大, 网络服务的最优布局也难以计算, 使得网络布局难以跟随用户需求动态变化。

服务布局问题是一个 NP 难问题, 其求解算法已经广泛的研究。随着数据中心(Data Center, DC)的发展, 数据中心内的虚拟机(Virtual Machine, VM)的合理布局能够有效的降低据中心的能源消耗, 降低成本, 所以很多学者针对这虚拟机布局方面展开了研究。文献[2] Y.Feng、B.Li 等设计了一种基于纳什讨价还价解决方案的算法, 作为管理这些 VM 迁移决策的实用方法, 能够随着时间的推移提高资源利用率。但是其缺点在于没有考虑网络服务类型的

多样性, 也就是多种服务类型实例的整体的布局问题。文献[4] Rochman、Levy 等考虑如何在网络环境中放置和有效利用资源的问题。运营商旨在将资源放在区域中, 以最大限度地减少提供需求的成本。文献[5]提出了一种基于博弈论的服务布局算法, 将不同类型的网络服务看作博弈的参与者, 博弈的商品则是网络节点资源。在网络资源约束和用户请求分布已知的情况下, 参与者不断的出价购买商品, 以布局其网络服务实例。文献[6]提出了选择 Open Flow 协议作为南接口协议。通过分离控制平面和数据平面模型, 使网络物联网的部署和管理更加方便, 在网络布局得到全局的网络知识。文献[7]-[9]提出了多背包问题的解决以及优化方法。

基于以上研究现状, 该文提出了基于多背包模型的分布式协同网络服务布局算法, 主要解决物联网不同服务类型业务的布局问题。该文首先根据业务的单位重量效益和时延定义了物联网业务请求的优先级, 将业务服务实例的布局问题建模为多背包模型; 其次算法先将业务请求在本地节点响应, 然后以遗传算法为核心, 利用混合遗传算法求解多背包问题; 最后通过仿真全局网络最大效益、节点响应率、节点效用比以及不同基本量下的节点响应率、节点效用比验证了算法有效性。

## 2 系统模型

### 2.1 网络模型及问题描述

本论文研究的系统模型如图 1 所示, 该文中网络由多个 SDN 控制器组成, 网络中节点是单个 SDN 控制器, 在这个单节点上可以布局多种类型的服务实例。由于某些服务类型的请求量大, 而某些服务的请求量小, 为了避免网络布局过程中过于偏向“受欢迎”服务, 而挤掉“不受欢迎”服务的资源, 导致无法响应这些服务的请求, 所以每类服务在网络中的布局既要网络角度出发提升网络资源利用率, 又要从服务角度出发提升用户体验。假设网络的所有节点掌握其它节点在某时刻的服务承载能力以及处理的服务请求分布等全局知识, 那么, 服务布局问题 (Placement problem) 的解可建模为网络效用最大化问题。

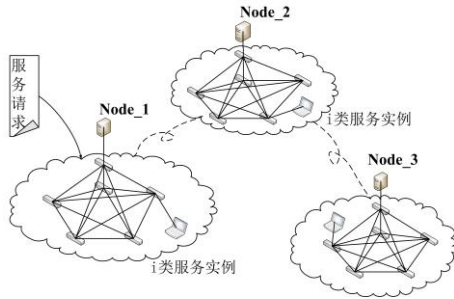


图 1 系统模型

### 2.2 数学模型

将网络抽象为无向图模型  $G=(N, \varepsilon)$ ,  $N=\{v_1, v_2, \dots, v_N\}$  为节点集合,  $\varepsilon$  为链路集合。该抽象网络模型中的节点可以是一个物理节点、一个网络、一个控制器控制的 SDN 网络或者网络功能等等。每个控制器节点可布局 类型的服务, 设不同类型服务的请求数为随机变量。标识  $k$  节点是否为业务请求  $d$  的本地节点  $y_{dk}$ ,  $d=1, 2, \dots, D, k=1, 2, \dots, N$ ,  $y_{dk}=1$  时表示  $k$  节点为请求  $d$  的本地节点, 反之则不是本地节点。在此网络模型中业务类型  $i$  在节点  $k$  上的分布记为  $D_i^k, i=1, \dots, M, k=1, \dots, N$ 。设  $D_{ik}$  为用户对位于节点  $k$  的服务类型为  $i$  的请求量, 网络整体的请求量为  $D$ 。

$$D = \sum_{i=1}^M \sum_{k=1}^N D_{ik} \quad (1)$$

$Z=\{z^1, z^2, \dots, z^N\}$  为节点  $N$  的布局资源约束, 即网络节点  $k$  的布局资源为  $z^k$ , 每类服务实例所占节点资源为  $w_i$ 。由于布局一类服务的实例达到一定的规模才会产生期望的效益, 所以本模型综合考虑布局的成本与效益。为了保证最小布局量, 本模型拟设每类服务  $i$  的最低布局规模为基本量  $B_i$ ,

由于所设定的基本资源  $B_i$ , 所以我们可以得到的  $i$  类型的服务实例实际的布局量要大于基本布局量  $B$ 。

$$\sum_{k=1}^N \sum_{d=1}^D x_{dk} * f_{id} \geq B_i \quad (2)$$

其中  $f_{id}=1$  表示请求  $d$  为  $i$  服务实例类型, 否则为 0。  $x_{dk}=1$  表示为  $k$  节点响应请求  $d$ , 反之,  $k$  节点未对请求响应。

那么  $k$  节点上响应的  $i$  类服务实例个数为  $n_{ik}$

$$n_{ik} = \sum_{d=1}^D x_{dk} * f_{id} \quad (3)$$

服务布局问题可表述为: 已知服务请求分布(4), 在可行域(5)内, 从布局策略集合(6)中选择最优策略, 基于多背包的分布式协同网络服务布局问题, 在网络资源约束和用户请求分布已知的情况下, 以此来优化网络服务实例的布局, 实现网络收益  $R$  最大化。综上所述, 即给定基本布局量  $B$  和服务请求分布  $D_{ik}$ , 求解布局在每个节点的各种类型的服务  $i$  的实例数  $n_{ik}$ 。该文提出的算法分配目标是在满足单个节点资源的基础上, 最大化全局网络服务实例的效益之和, 即最大化网络总效益。

$$D_i^k, i=1, \dots, M, k=1, \dots, N \quad (4)$$

$$Y = \sum_{i=1}^M n_{ik} * w_i \leq z^k \quad (5)$$

$$Y = \{n_{ik}\} \quad i=1, 2, \dots, M \quad k=1, 2, \dots, N \quad (6)$$

### 2.3 业务请求优先级

文献[10]重点介绍了 M2M 业务的 QoS 分类。根据流量特征提出了 M2M 业务类别根据业务请求类型的不同。因此我们根据其业务请求类型, 确定其时延敏感性、单位重量效益和 QoS 优先级。根据 M2M 业务是否时延敏感的性质, 我们将其分为三种类型: 实时性时延敏感业务、实时性业务和时延容忍类业务, 并且设定其时延分别为 100ms、200ms 和 NA; 所谓单位重量效益, 该文定义  $p_i / w_i$  为每种业务类型单位重量效益。根据业务类型单位重量效益以及时延特性, 该文规定其 QoS 等级分别为 1、2 和 3, 如表 1 所示。根据表 1 所制定的标准对不同的业务请求类型进行分类 (依据 QoS 优先级) 可以使得业务服务布局合理。

表 1 业务请求类型

业务类型	业务单位重量效益	时延	丢包率	QoS 等级
实时性且时延敏感业务如视频流报警	高	100ms	$10^{-3}$	1

实时性业务如 实时监控	中	200ms	10 <sup>-3</sup>	2
时延容忍类业 务如周期监控	低	NA	NA	3

### 3 业务服务布局算法

#### 3.1 业务服务实例产生的效益

当用户请求服务类  $i$  的实例时，若请求在本地节点上响应，则产生的效益为  $r_i^{loc}$ 。若该请求在本地节点以外的节点被响应，则产生的效益为  $r_i^{rem}$ 。显而易见  $r_i^{loc} > r_i^{rem}$ 。如果两者相差越大，业务请求远程节点响应的代价则越大，效益也相应地降低。另外，为了获得更大的效益，需要考虑各种服务类型的布局效益。

$$p_i = \begin{cases} r_i^{loc} & \text{local response} \\ r_i^{rem} & \text{rem response} \end{cases} \quad (7)$$

$$R_i = \sum_{k=1}^N \sum_{d=1}^D x_{dk} * f_{id} * p_i \quad (8)$$

在全局网络下布局  $i$  类型的服务实例产生的效益为(8)，其中  $p_i$  为单个  $i$  类型服务请求在节点上响应所产生的效益。

#### 3.2 业务服务实例布局

在节点的资源充足时，可以为每个业务请求分配充足的资源，则取得最大化效益；但是在资源不能够为所有的业务请求分配资源时，我们需要对请求业务进行取舍以达到最大化网络效益。

网络中节点视作资源为  $z^k$  ( $k = 1, 2, \dots, N$ ) 的  $N$  个背包，每个业务请求视作一个物品，总共有  $D$  个物品，并且每类业务请求的重量为  $w_i$  ( $i = 1, 2, \dots, M$ )，表示每类业务请求所占的资源；每类业务请求所产生的效益是  $p_i$ 。全局网络所产生的总效益是每个业务请求在节点上所产生的效益之和。

综上所述，优化方程(9)可以表示为：

$$\text{Maximize} \quad \sum_{i=1}^M \sum_{k=1}^N \sum_{d=1}^D x_{dk} * f_{id} * p_i \quad (9.1)$$

$$\text{Subject to} \quad \sum_{i=1}^M n_{ik} * w_i \leq z^k, k = 1, \dots, N \quad (9.2)$$

$$\sum_{k=1}^N \sum_{d=1}^D x_{dk} * f_{id} \geq B_i, i = 1, \dots, M \quad (9.3)$$

$$\sum_{k=1}^N x_{dk} \leq 1, d = 1, \dots, D \quad (9.4)$$

$$\sum_{i=1}^M f_{id} \leq 1, d = 1, \dots, D \quad (9.5)$$

$$x_{dk} \in \{0, 1\}, d = 1, \dots, D \quad k = 1, \dots, N \quad (9.6)$$

$$f_{id} \in \{0, 1\}, d = 1, \dots, D \quad i = 1, \dots, M \quad (9.7)$$

其中  $n_{ik} = \sum_{d=1}^D x_{dk} * f_{id}$ ， $R_i(t)$  为布局类型为  $i$  的服务所产生的效用。约束条件(9.2)保障了节点  $k$  的资源约束；约束条件(9.3)保障了网络中所有节点分布的  $i$  类型服务总和要高于类型服务的基本布局量  $B_i$ ；约束(9.5)中  $f_{id}=1$  表示业务请求  $d$  为  $i$  类型服务实例，否则为 0； $x_{dk}=1$  表示为  $k$  节点响应请求  $d$ ，反之  $k$  节点未对请求  $d$  响应；约束(9.6)表示每个服务请求  $d$  仅有一个节点上响应；约束(9.7)表示每个服务请求的类型。

在该文的业务实例布局方法，当节点的资源充足时，可以为每个业务请求分配资源，则取得最大化效益；在节点资源不足以布局下所有的业务请求响应时，优先将在本地节点的请求业务进行布局，而且根据业务类型的优先级优先布局单位重量效益较高的业务，如果完成本地节点业务请求的布局还有剩余资源，则可以接收来自外部节点的请求，反之如果本地资源不足以承受本地的所有业务请求，则向外部节点进行请求。该文方法在布局最初就要布局各种类型服务实例，并且其实例数量要高于基本布局量  $B_i$ ，所以可以去掉约束条件(9.3)；并且由于  $f_{id}$  是用来标识业务请求  $d$  为  $i$  服务实例类型，所以其在业务请求分布已知的条件下，其值是固定的，所以去掉约束(9.5)和约束(9.7)。假设其节点资源  $z^k$  ( $k = 1, 2, \dots, N$ ) 都为  $z$ 、业务请求  $d$  所得的效益为  $p_d$  和所占资源为  $w_d$ ，因此可以表示为：

$$p_d = \sum_{i=1}^M f_{id} * p_i \quad (10)$$

$$w_d = \sum_{i=1}^M f_{id} * w_i \quad (11)$$

将(9)简化后优化方程(12)为：

$$\text{Maximize} \quad \sum_{k=1}^N \sum_{d=1}^D x_{dk} * p_d \quad (12.1)$$

$$\text{Subject to} \quad \sum_{d=1}^D x_{dk} * w_d \leq z, k = 1, \dots, N \quad (12.2)$$

$$\sum_{k=1}^N x_{dk} \leq 1, d = 1, \dots, D \quad (12.3)$$

$$x_{dk} \in \{0, 1\}, d = 1, \dots, D \quad k = 1, \dots, N \quad (12.4)$$

可见优化方程(12)为一个典型的 0-1 多背包模型，其解是  $N * D$  的矩阵，每一位表示业务请求和节点的关系， $x_{dk}=1$  表示为业务请求  $d$  在  $k$  节点上响应，反之表示业务请求  $d$  没在  $k$  上响应；约束(12.2)表示每个节点的资源约束；约束(12.3)表示每

个服务请求  $d$  仅在一个节点上响应。如果使用遗传算法进行二进制编码，当节点数和业务请求数乘积比较大时，搜索空间规模会很大，为  $O(2^{N \times D})$ ，因此将模型(12)转化为如下模型(13)。

$$\text{maximize } \sum_{d=1}^D (x_d \sim= 0) * p_d \quad (13.1)$$

$$\text{subject to } \sum_{d=1}^D (x_d == k) * w_d \leq z, k=1, \dots, N \quad (13.2)$$

$$x_d \in \{0, 1, 2, \dots, N\} \quad (13.3)$$

其中  $x_d = k (k \neq 0)$  表示节点  $k$  响应了第  $d$  个业务请求， $x_d = 0$  表示业务请求  $d$  没有被响应； $(x_d == k)$  和  $(x_d \sim= 0)$  是逻辑表达式，表达式为真时其值为 1，否则为 0；约束(9.1)是每个节点的资源约束。求解多背包问题的混合遗传算法（HGA）流程图如图 2 所示

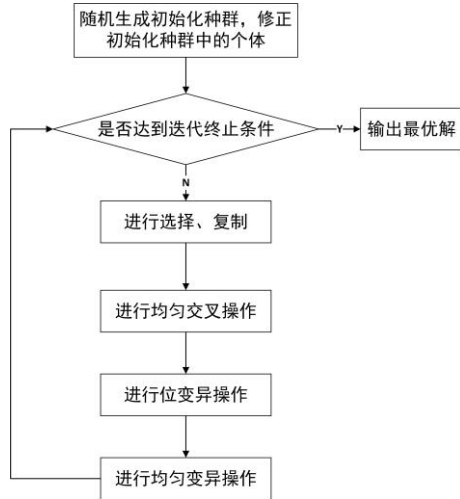


图 2 混合遗传算法流程图

$$f = \sum_{d=1}^D (x_d \sim= 0) * p_d \quad (14)$$

本算法中适应值函数定义为(14)，并且生成初始种群。HGA 交叉操作采用均匀交叉，能够有效地保持群体的多样性，使算法具有较强的搜索能力。均匀交叉操作方法是：

步骤 1 随机地产生与父个体等长的两个 0-1 掩码，利用掩码形成两个新个体。

步骤 2 用修复和修正程序将新个体进行修复和修正。

步骤 3 将两个修复和修正后的新个体与两个父个体放在一起选择，选择两个适应值较高的个体作为子代。

HGA 变异操作采用位变异和均匀变异操作。采用均匀变异能够有效地保持群体的多样性，使算法具有较强的搜索能力。均匀变异操作如下：

随机地产生与父个体等长的一个 0-1 掩码（掩码中的片段表明了父个体的位是否变异），利用掩

码形成一个新个体。用修复和修正程序将新个体修复和修正。

并且最大迭代次数  $n$  取为 50 代，达到则停止迭代并输出最优解。此模型(14)即可采用此服务布局算法求解。

---

算法 基于多背包的网络服务布局算法

---

输入：

节点资源： $z_k$ ；

业务请求矩阵： $D$ ；

业务请求效益： $p_d$ ；

业务请求所占资源： $w_d$ ；

业务标识符： $y_{dk}$ ；

迭代次数： $n=50$ ；

交叉率： $p_{jr}=0.8$ ；

位变异率： $p_{wr}=0.01$ ；

均匀变异率： $p_{ar}=0.1$ ；

输出：

服务选择矩阵： $x_d$

1：业务请求矩阵  $D$  基于  $p_d / w_d$  降序排列得到  $D_s$

2：For  $k \in [1, N]$  Do //请求首先在本地节点响应

3：For  $d \in [1, D]$  Do

4：If  $y_{dk} == 1$  And

$$\sum_{d=1}^D (x_d == k) * w_d \leq z_k \quad \text{Do}$$

5： $x_d = k$ ；//业务请求  $d$  在本地节点  $k$  响应

6： $z_k = z_k - w_d$ ；//更新节点  $k$  资源

7： $D_s = D_s - \{d\}$ ；//在业务请求集合

$D_s$  中删除请求  $d$

8：End If

9：End For

10：End For

11： $\{x_d\} = \text{HGA}(D_s, z_k, p_d, w_d, p_{jr}, p_{wr}, p_{ar}, n)$ ；

---

## 4 仿真分析

为了体现该文提出的基于多背包模型的分布式协同网络服务布局算法的优越性, 该文将提出的算法与传统服务布局算法进行比较。每类业务请求的时延、单位重量效益以及优先级如表 2 所示, 仿真参数如表 3 所示。

表 2 业务性能参数

业务单位重量效益	时延 (ms)	业务优先级
0 - 2	NA	5
2 - 4	NA	4
4 - 7	200	3
7 - 10	50	2
大于 10	10	1

表 3 仿真参数

业务类型个数 ( $M$ )	10
基本布局量 ( $B$ )	500
节点个数 ( $N$ )	50
节点资源 ( $z$ )	8000
业务所占资源 ( $w_d$ )	10-50
业务效益 ( $p_d$ )	20-260

### 4.1 网络总效益

首先给出度量服务布局性能最直观的参数——网络总效益。图 3 给出了在给出请求不断增加的情况下, 比较了该文提出的方法和传统方法。仿真结果表明, 在业务请求较少的情况下, 该文提出的方法与传统方法得到的网络总效益相同。当请求增加到一定程度, 会出现节点资源不足, 此时该文的方法得到的总效益明显多于传统的方法。

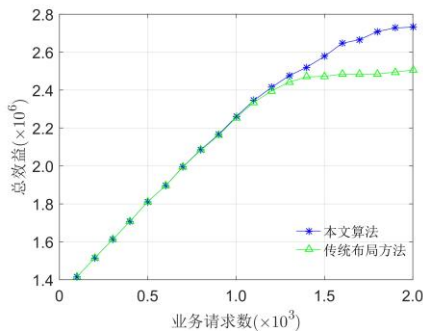


图3 网络总效益和业务请求数关系

### 4.2 节点响应率

节点响应率——用于度量业务请求是否被成功响应。其反映了方法对用户请求体验的影响, 其值高表明用户体验好, 反之则表明用户体验差, 并且可能会出现服务等级达不到要求。节点  $k$  的请求响应率为  $r(k)$ ——为节点  $k$  的实际响应的请求数与节点总的请求数比, 即

$$r(k) = \frac{\sum_{d=1}^D (x_d == k)}{\sum_{i=1}^M D_{ik}} \quad (15)$$

网络中平均的请求响应率为:

$$r = \frac{1}{N} \sum_{k=1}^N r(k) \quad (16)$$

图 4 仿真结果表明, 在业务请求较少的情况下, 该文提出的方法与传统方法都能响应所有的请求, 即节点响应率为 1。当请求增加到一定程度, 该文方法和传统方法都会出现节点响应率降低, 但是该文方法请求响应率明显高于传统方法。

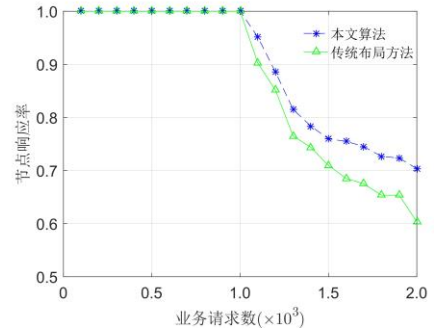


图4 节点响应率与业务请求数关系

### 4.3 节点效用比

节点效用比用来度量网络利用率, 反应了单元资源的收益情况。其值越高, 表示效益越大。节点  $k$  的节点效用比  $R(k)$  可以定义为节点  $k$  的实际布局业务所占资源与节点资源之比。

$$R(k) = \frac{\sum_{d=1}^D (x_d == k) * w_d}{z} \quad (17)$$

网络中平均的节点效用比为:

$$R = \frac{1}{N} \sum_{k=1}^N R(k) \quad (18)$$

图5仿真结果表明,在业务请求较少的情况下,该文提出的方法与传统方法节点效用比相同。当业务请求数量增加,该文方法和传统方法的节点效用比都增加,但是该文方法节点效用比明显高于传统方法。

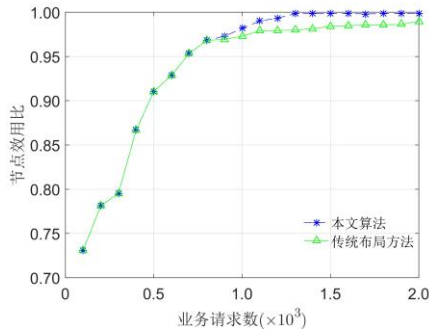


图5 节点效用比和业务请求数的关系

#### 4.4 基本布局量

调节基本布局量可以实现不同业务类型的服务布局的公平。本次仿真设置基本布局量分别为 500, 1000 和 2000。图 6a 是在基本布局量不同的情况下请求响应率的变化曲线,图 6b 给出了在基本布局量不同的情况下节点效用比的变化曲线。仿真结果表明,随着基本布局量的增加,请求响应率也随之上升,而节点资源的利用率呈现下降趋势,在基本布局量(2000)过多的时候下降的最为明显。由仿真结果表明,如果布局中设置合适的基本布局量( $B=1000$ ),能够降低业务服务类型数量变化带来的影响,使得请求响应率和节点效用比之间达到均衡。

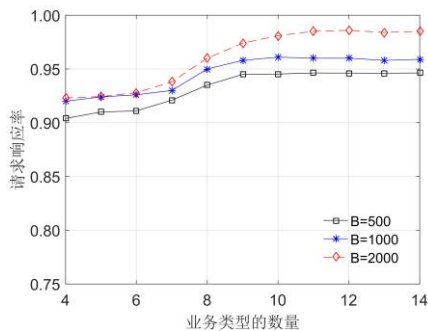


图 6a 不同基本布局量下的请求响应率

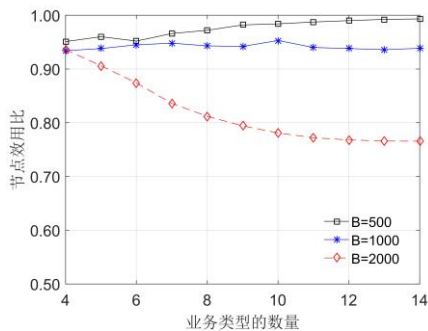


图 6b 不同基本布局量下的节点效用比

## 5 结语

针对全局网络的效益最大化,该文通过分析物联网业务的特性,提出了基于多背包模型的分布式协同网络服务布局算法,优先分配单位重量效益高的业务请求,保障了业务 QoS;并且业务请求优先在本地节点响应,减少业务请求远程节点响应的代价。仿真结果表明该算法可以提高全局网络的总效益,同时能够提高节点响应率和节点效用比,具有一定的优越性。

#### 参考文献:

- [1] Mehmood Y, Görg C, Muehleisen M, et al. Mobile M2M communication architectures, upcoming challenges, applications, and future directions[J]. EURASIP Journal on Wireless Communications and Networking, 2015, 2015(1):250.
- [2] Feng Y, Li B, Li B. Bargaining towards maximized resource utilization in video streaming datacenters[J]. 2012, 131(5):1134-1142.
- [3] Shen H, Li Z. New bandwidth sharing and pricing policies to achieve a win-win situation for cloud provider and tenants[C]// INFOCOM, 2014 Proceedings IEEE. IEEE, 2016:835-843.
- [4] Rochman Y, Levy H, Brosh E. Resource placement and assignment in distributed network topologies[C]// INFOCOM, 2013 Proceedings IEEE. IEEE, 2013:1914-1922.
- [5] 程国振. 基于元能力的网络功能组合 关键技术研究[D]. 解放军信息工程大学, 2015. CHEN Guozhen. Research on the Key Technologies of Function Composition Based on Atomic Capability[D]. PLA Information Engineering University, 2015
- [6] Tu L, Zhou X, Sheng Q. Analysis on Internet of Things Architecture Based on SDN[J]. Wireless Internet Technology, 2016.
- [7] Laalaoui Y, M'Hallah R. A binary multiple knapsack model for single machine scheduling with machine unavailability[M]. Elsevier Science Ltd. 2016.
- [8] 宋海生, 傅仁毅, 徐瑞松, 等. 求解多背包问题的混合遗传算法[J]. 计算机工程与应用, 2009, 45(20):45-48.



- SONG Haisheng, FU Renyi, XU Ruisong, et al. Hybrid genetic algorithm for multi -knapsack problem[J]. Computer Engineering and Applications, 2009, 45 (20) : 45-48.
- [9] Parada L, Herrera C, Sepúlveda M, et al. Evolution of new algorithms for the binary knapsack problem[J]. Natural Computing, 2016, 15(1):181-193.
- [10] Liu R, Wu W, Zhu H, et al. M2M-Oriented QoS Categorization in Cellular Network[C]// International Conference on Wireless Communications, NETWORKING and Mobile Computing. IEEE, 2011:1-5.