

分布式环境中基于社会网络的资源发现机制

徐群叁, 徐邦海, 孙玉娟

XU Qun-san, XU Bang-hai, SUN Yu-juan

鲁东大学 计算机科学与技术学院, 山东 烟台 264025

School of Computer Science and Technology, Ludong University, Yantai, Shandong 264025, China

E-mail: xqsyt@163.com

XU Qun-san, XU Bang-hai, SUN Yu-juan. Mechanism of resource discovery based on social network in distributed environments. Computer Engineering and Applications, 2009, 45(34): 110-113.

Abstract: In this paper, a method of resource discovery based on social network is designed which is non-center, survivable, and self-adaptive. Through this method, the nodes can cooperate with each other to find the resources they want. And then, the model and algorithm of the method is presented. Finally, the simulation results have shown that the method can be used to improve the discovery performance, be good at adapting the different distributed resources and mode of user's requesting, and survive in the dynamic environment of network.

Key words: distributed environment; resource discovery; social network

摘要: 基于社会网络设计了一个无中心的、可生存的、自适应的资源发现方法, 使得网络节点相互协作, 以发现所需资源。接着, 详细地给出了方法的模型、算法。仿真结果表明该方法能够改善发现的性能、很好地适应不同的资源分布和用户请求模式以及能够从动态的网络环境中生存。

关键词: 分布式环境; 资源发现; 社会网络

DOI: 10.3778/j.issn.1002-8331.2009.34.034 **文章编号:** 1002-8331(2009)34-0110-04 **文献标识码:** A **中图分类号:** TP393

1 引言

今天的绝大多数分布式网络应用(如网格系统), 一般是基于持久性的服务架构, 通常采用一种有中心或者具有层次型协作的分布式结构。因为有大量的节点频繁地加入或离开, 分布式网络是一个不稳定的环境, 因此为了避免瓶颈的出现而必须分散其功能, 这样新的服务才能够生存并自适应于变化的环境^[1]。也就是说, 分布式网络中的资源发现机制需要在一个广域的环境中, 实施自治的、灵活的行为来生存、自适应于不确定的因素(如, 动态的资源状态、不同的资源分布、变化的用户资源请求)。然而, 目前的发现技术如目录服务不能应付动态资源发现的挑战。因此, 有必要探讨新的发现方法。

人类社会作为一个系统, 具有很多独特的特点, 例如人们之间有着天然的方式去查找信息或资源^[2]。有证据表明, 这种在大型社会网络中依靠熟识关系的搜索方式是非常有效的。社会学家 S. Milgram 在“小世界”方面的先驱性工作就是这方面的有力佐证^[3]。因为人们不可能掌握网络的全局信息, 所以能够通过熟识关系网络及很少的连接数将信息直接传送给一个特定的、远方的人是令人惊奇的。这不仅启示了在一个大型社会网络中个体之间短路径的存在^[4-5], 而且表明普通人可以找到这种短路径^[6]。此外, 一些研究工作也解释了社会网络的可搜索性^[7]。这些研究表明社会网络提供了一个完全无中心的、天生自适应的、

可生存的、个人对个人的信息资源发现、收集与发布的通信方法。显然这种便捷有效且不需具备全局信息的搜索方法可以应用到分布式网络的资源发现研究中。

主要应用社会网络的主要机制和特点来设计分布式网络应用中的资源发现方法。在网络实体中设计了简单智能, 使用户能基于类似于社会网络的协作定位所需要的资源, 如个人通信行为和基于信任的聚类。这里, Agent 技术、P2P 计算、社会网络机制的三者结合看起来非常完美。Agent 是面向协作和通信的, 而 P2P 计算支持典型的所有节点同等角色和责任的分布式系统。这些特点又特别适合于该方法中的社会实体。

2 基于社会网络的资源发现方法

2.1 社会网络的特点

在现实社会中, 人们经常通过询问他们的熟人来获得好的建议, 这也可以看作是人与人之间的一种协作。这种协作有以下一些主要特点:

(1) 人与人之间的通信: 不可否认的是, 它在一个无中心的社会环境中传递和寻找有价值信息的过程中扮演了一个很重要的角色。

(2) 局部的视野: 每个人只有相对有限的熟人。没有人知道关系链上所有的个体。

作者简介: 徐群叁(1974-), 男, 讲师, 主要研究领域为计算机网络; 徐邦海(1974-), 男, 博士, 副教授, 主要研究领域为网络安全; 孙玉娟(1977-), 女, 讲师, 主要研究领域为信息工程及控制。

收稿日期: 2008-06-30 **修回日期:** 2008-09-19

(3)个体的自治特点:社会活动没有全局的控制中心,如朋友圈就不是由一个最高的管理机构指定的,而是各个个体自己作为决策主体形成的。

(4)信任:信任是一个很重要的社会概念,能够帮助人们处理他们的社会环境,并且在所有的人类的相互作用中得到体现。

(5)聚类:在现实社会中,个体会通过长期的协作形成聚类。

2.2 基于社会网络的发现机制

基于对社会网络的观察,提出一个三阶段的发现机制:信息服务 Agent 的关系构成、请求处理策略和基于信任的关系重构。

(1)信息服务 Agent 的关系构成主要是负责收集和更新目前的网络中所参与 Peer 的信息,而形成一个关系网络。一个 Peer 通过联系一个网络成员 Peer 加入到网络中来。一个网络成员地址的获得可以通过生态网络平台的信息侦听机制。对其他 Peer 的主动搜索包括用网络节点级的连接广播一个查询消息来获取周围的 Peer 信息。一旦一个邻近的信息服务 Agent 被发现,就可以建立一个关系。此外,通过其他信息服务 Agent 的介绍和发现相关的相互作用,一个信息服务 Agent 也可以与其他 Agent 建立一个关系。要加入网络的 Peer 所联系的成员 Peer 将返回自己的 agentAddress 属性。这样,待加入的 Peer 就可以访问所联系 Peer 的公共信息,如它提供了什么样的服务。这些信息有助于待加入的 Peer 决定是否加入一个已经存在的生境。这里,相互认识的 Peer,称之为“熟人”。关系信息可以随着时间被加强:一接到从一个以前未知的 Peer 传来的消息,一个 Peer 会将这个新的信息加入到它 relationshipDescription 属性列表中。这样,一套关系就被建立和调整来形成一个所有信息服务 Agent 的社会网络。

这种关系网络可以被看成是一个覆盖网络^[8],它是一个建立在一个或者多个已经存在的网络之上的独立的虚拟网络。当信息服务 Agent 之间更多的关系被获得,社会网络将实现出好的性质。例如,社会网络包括很多到达目标信息服务 Agent 的关系路径,这样当通信失败或者信息服务 Agent 死亡都不能影响消息的可达性。并且,如果关于信息 Agent 的一个信息改变了(如在网络中的位置改变),关系网络仍然保持,这是由于信息服务 Agent 能注意它们的熟人,并更新变化的信息。

(2)请求处理策略是执行搜索本身。一个请求消息包括消息 ID(全局唯一 ID)、消息类型(显示消息是一个请求)、请求始发者的信息(如 agentID 和 agentAddress)、请求传递者信息(传递请求消息的信息服务 Agent 的 agentID 和 agentAddress),一套参数来指定目标资源的属性、一套权值来描述每个资源属性的重要程度(在一个高度动态和异构的下一代 Internet 环境中,采用一个全局的命名机制是不可行的)。

当一个信息服务 Agent 收到一个请求的消息,它首先检查消息 ID 来判断它是否以前看到过这个消息。如果看到过,它会丢弃这个消息。如果没有看到过,它会在消息处理列表中创建一个新的条目。一个消息处理列表包括三个项目:接收到的消息 ID、消息类型、到达时间。“接收到的消息 ID”记录了所接收到的请求消息的 ID,这是为了辨别同一消息从多个路径到达。“消息类型”项目表示了消息的类型(如,是一个 inform 消息)。“到达时间”项目记录了消息第一次到达的时间,它被用来决定什么时候删除一个条目,以防列表无限制地增长。

在完成对新接收到请求消息入表处理后,信息服务 Agent

采用请求评价模型来评价请求。然后信息服务 Agent 决定是以本地资源来响应请求、传递搜索请求给它的熟人、或者拒绝请求。注意,一个信息服务 Agent 可以既响应又传递搜索的请求。下面将介绍请求的评价模型,它包括一个资源匹配模型和请求传递模型。

(1)资源匹配模型:一个请求消息包括描述目标资源属性的参数。以向量空间的形式来表述这些参数。在一个 Peer 上提供的网络资源被描述为一个向量。同样,请求的资源也被描述成一个向量。下面,给出两个定义,建立资源匹配模型。

定义 1 给定一个单请求属性值 r 和一个单资源属性值 g , 则 r 和 g 的匹配强度被定义为:

$$MS_{single}(r, g) = \begin{cases} 1, & g \text{ satisfies } r \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

例如,一个请求属性值 r 表示为“超过 128 MB 可用内存”,一个资源属性值 g 表示“256 MB 可用内存”。这样根据式(1),得到 $MS_{single}(r, g)=1$ 。

定义 2 给定一个请求向量 $R = \langle r_1, r_2, \dots, r_n \rangle$, 一个权值向量 $W = \langle w_1, w_2, \dots, w_n \rangle$ (其中 $\sum_{i=1}^n w_i = 1$) 表示 R 中各个请求属性的重要程度,一个资源向量 $G = \langle g_1, g_2, \dots, g_n \rangle$, 则 R 和 G 之间的匹配强度被定义为:

$$MS_{vector}(R, W, G) = \sum_{i=1}^n MS_{single}(r_i, g_i) \cdot w_i \quad (2)$$

因此,给定一个阈值 $\omega_i (0 \leq \omega_i \leq 1)$, 如果 $MS_{vector}(R, W, G) \geq \omega_i$, 则 R 和 G 匹配。例如,对于一个三元组向量 $\{MS_{single}(r_1, g_1), MS_{single}(r_2, g_2), MS_{single}(r_3, g_3)\} = \{1, 1, 0\}$, 且有 $W = \langle 0.5, 0.3, 0.2 \rangle$ 。用定义 2, 可以获得

$$MS_{vector}(R, W, G) = \sum_{i=1}^3 MS_{single}(r_i, g_i) \cdot w_i = 1 \times 0.5 + 1 \times 0.3 + 0 \times 0.2 = 0.8$$

如果 $\omega_i = 0.8$, 则资源与请求匹配。如果 $\omega_i = 0.9$, 不匹配。用户可以在请求消息中设定不同的权值向量来着重描述它们所需要的资源属性。

(2)请求传递模型:当一个信息服务 Agent 处理一个请求消息时,它不仅检查本地资源是否满足要求,还通过计算最佳匹配强度来决定将该请求传递给哪些熟人。一个信息服务 Agent 可以存储它熟人的一些信息。其他的信息(也就是 trustCredit 和 collaborationRecord) 可以总体上形成一个关系网络。通过标注它的熟人的 trustCredit 值, 信息服务 Agent 信赖某些熟人,不信任另一些来完成资源发现活动。一个以前表现很好的信息服务 Agent 在未来的发现中也可能表现很好。此外,在 collaborationRecord 中, 用户以前评价的相似请求的信息可能帮助用户获得需要的资源。基于 trustCredit 和 collaborationRecord, 提出一个请求传递模型。

假定信息服务 Agent A_i 是信息服务 Agent A_j 熟人之一。 A_i 的熟人列表如下所示:

$$C_i = \{(I_i^{sel}, trust_{i,i}, I_i^{col}), (I_2^{sel}, trust_{i,i_2}, I_2^{col}), \dots, (I_n^{sel}, trust_{i,i_n}, I_n^{col})\} \quad (3)$$

其中, I_i^{sel} 是 A_i 的熟人 A_i 的自描述信息(包含 agentID、agentAddress、serviceType、serviceDescription), $trust_{i,i_j}$ 是在 $[0, 1]$ 之间的一个数值, 表示 A_i 对 A_j 的 trustCredit 值。 I_i^{col} 是 A_i 的 collaborationRecord 属性中关于 A_i 的信息, 表示如下:

$$I_i^{col} = \{(R_i^1, W_i^1, \delta_i^1), (R_i^2, W_i^2, \delta_i^2), \dots, (R_i^m, W_i^m, \delta_i^m)\} \quad (4)$$

其中, $R_i^k = \langle r_{i_1}^{k_1}, r_{i_2}^{k_2}, \dots, r_{i_j}^{k_j} \rangle$ 是 A_i 以前回答或传递的请求向量。 $W_i^k = \langle w_{i_1}^{k_1}, w_{i_2}^{k_2}, \dots, w_{i_j}^{k_j} \rangle$ 是一个表示 R_i^k 每个属性的权值向量。 δ_i^k 是在 $[-\frac{1}{2}, 1]$ 之间的一个参数, 表示请求向量 R_i^k 的用户评价。这里, 假定对于信息服务 Agent 和发现记录没有限制。 A_i 通过信息服务 Agent 关系重构, 定时地更新这些记录。根据上述表达式, 给出了传递强度的定义。

定义 3 给定一个请求向量 R , 一个表征 R 的每一个属性的权值向量 W , 在 collaborationRecord 中最佳的匹配强度被定义为

$$MS_{opt}(R, W, R_i^k) = \max\{MS_{vector}(R, W, R_i^1) \cdot \delta_i^1, MS_{vector}(R, W, R_i^2) \cdot \delta_i^2, \dots, MS_{vector}(R, W, R_i^m) \cdot \delta_i^m\} \quad (5)$$

进一步地, 信息服务 Agent A_i 对于它的熟人 A_{i_1} 的传递强度被定义为:

$$FS_{vector}(R, W, A_i, A_{i_1}) = \eta \cdot trust_{i_1, i} + (1 - \eta) \cdot MS_{opt}(R, W, R_i^k) \quad (6)$$

其中, η 是给定 trustCredit 和 collaborationRecord 的权值。

如果最佳传递强度

$$FS_{opt}(R, W, A_i, A_{i_1}) = \max\{FS_{vector}(R, W, A_i, A_{i_1}), FS_{vector}(R, W, A_i, A_{i_2}), \dots, FS_{vector}(R, W, A_i, A_{i_n})\}$$

被获得, A_i 将做出一个决策: 哪一个熟人最可能导向想要的资源。在 A_i 将请求信息传递给它的熟人之前, 它将自己的信息注册在消息上。并且, 在传递过程中, A_i 将会指定一个阈值 θ_i 。需要注意的是, 通常有 $\theta_i \leq \omega_i$ (ω_i 是匹配阈值)。如果 $FS_{opt}(R, W, A_i, A_{i_1}) \geq \theta_i$, A_i 将传递请求消息。并且, 如果 A_i 既不以本地资源响应, 又不传递消息, 它将丢弃这消息。

算法 1 处理一个 Peer 所接收到请求信息。

假定一个信息服务 Agent A_i (本地资源向量为 G_i , A_i 为其任意一个熟人, 它匹配的阈值和传递阈值分别为 ω_i 和 θ_i) 接收到一个具有资源请求向量 R 和一个权值向量 W 的请求消息。

根据下列规则, 该 Agent 处理这条消息:

检查消息 ID (每个消息都包括一个唯一的 ID)

if 相同的请求消息被获得过

丢弃这条消息

Else

在请求消息处理列表中创建一个新的条目

评价资源请求

if 匹配强度 $MS_{vector}(R, W, G_i) \geq \omega_i$

以本地资源响应请求

end if

if 最佳传递强度 $FS_{opt}(R, W, A_i, A_{i_1}) \geq \theta_i$

在请求消息上注册信息

传递查询请求

else

拒绝请求

end if

end if

在所提出的方法中, 用户只需要确定三个参数: η, ω_i, θ_i 。 η

的影响是帮助用户选择一个适合的参数。另外两个参数 ω_i 和 θ_i , 是为了带给信息服务 Agent 一种“责任”机制而被引入该方法的。如果一个信息服务 Agent 设置了一个不合适的 ω_i 或者 θ_i , 它将会因为不负责任的行为导致不满意的发现而受到惩罚。

一个提供了高质量服务的信息服务 Agent, 将被奖励—它的 trustCredit 获得了提高。而一个提供低质量服务的信息服务 Agent 将被惩罚—它的 trustCredit 将被降低。

使用 TTL (Time To Live) 值来描述传递请求的时限。此外, 当返回搜索结果时, 通常有两种方法。匹配给定请求的信息服务 Agent 或者直接将搜索的结果返回给发现起始者, 或者沿着发现传递的路线返回。采用前者的方案, 因为分布式网络要承担起社会信息基本架构的责任, 需要有负责性、性能调节、复制决策的特点, 而后者方案所带来的一系列匿名性的问题显然不合需要。

(3) 基于信任的关系重构是为下一次更加有效的搜索做准备。这里, 讨论一种关系重构的动态机制, 它将有助于更有效的资源发现。这种机制通过建立和改变信息服务 Agent 之间的 trustCredit 值来更新和加强关系网络。通过 trustCredit 值, 信息服务 Agent 的可靠性被表达。

一接到一个请求命中 (request hit), 请求起始者将返回一个支付 (defray) 消息, 该消息包括一个协作记录、一个表示用户对请求命中的评价的信任值。如果一个信息服务 Agent 的关系不包括相应的协作记录, 则这个包括初始用户评价的协作记录被加入到 relationshipDescription 属性中。信任值可以是一个奖励, 也可以是一个惩罚, 它表示用户对接收到的请求命中的偏爱程度。这个消息沿着开始传递的路径被传播, 当在这一条路径上的中间信息服务 Agent 接收到消息, 它将调节被用来传递初始发现请求的关系的 trustCredit 值。

作为奖励 (请求起始者偏爱程度高) 时, TrustCredit 值增加; 作为惩罚 (请求起始者偏爱程度低) 时, TrustCredit 值减少。

定义 4 给定包含在一个支付消息中的, 代表一个奖励或者惩罚一个信任值 γ ($-\frac{1}{2} \leq \gamma \leq 1$), 信息服务 Agent A_i 以下列公式更新它的熟人的 trustCredit 值:

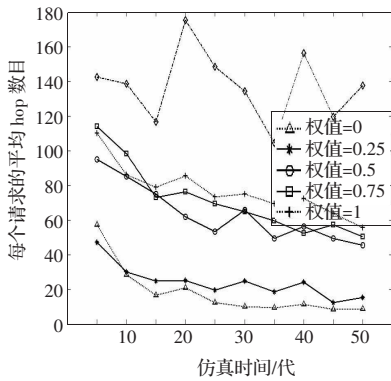
$$trust_{i_1, i}^+ = \begin{cases} trust_{i_1, i}^- (1 - \gamma^2) + \gamma^2, & \gamma \geq 0 \\ trust_{i_1, i}^- (2 - \frac{1}{1 + \gamma}), & \gamma < 0 \end{cases} \quad (7)$$

其中, $trust_{i_1, i}^-$ 是 $[0, 1]$ 间的一个数值, 表示更新前 A_i 对于 A_{i_1} 的 trustCredit 的值, 相应地, $trust_{i_1, i}^+$ 是更新后 trustCredit 的值。选择上述公式的目的在于使这些评价下降得很快, 而上升得很慢。下面给出了一个 Peer 收到的处理支付消息的算法 2。

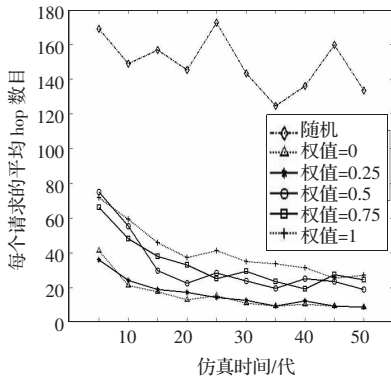
算法 2 处理一个 Peer 所接收到的支付消息。

假定一个信息服务 Agent A_i 通是支付消息传播路线中的一个中间 Peer, A_{i_1} 是 A_i 的下一个 Peer (也就是 A_i 传递请求消息给 A_{i_1})。如式 (3) 定义, I_i^{col} 是 A_i 知道的有关于 A_{i_1} 的信息, δ_i^k (δ_i^{k+}) 表示更新前 (后) 用户的评价。 $trust_{i_1, i}^-$ ($trust_{i_1, i}^+$) 表示更新前 (后) A_i 对 A_{i_1} 的 trustCredit 值。

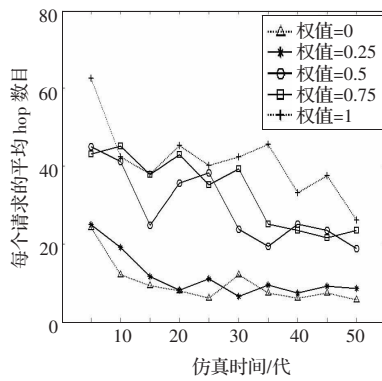
当 A_i 接收到一个带有信任值 γ 和协作记录 I_i^k 的支付消息, 它根据下列原则处理消息:



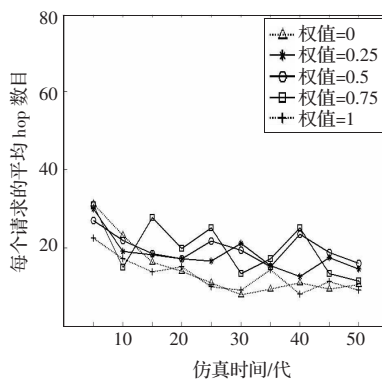
(a)平衡的资源分布和无偏向的请求方案



(b)不平衡的资源分布和无偏向的请求方案



(a)平衡的资源分布和有偏向的请求方案



(b)不平衡的资源分布和有偏向的请求方案

图1 无偏向的用户请求方案在两种不同资源分布情况下每个请求发现的平均 hop 数

图2 有偏向的用户请求方案在两种不同资源分布情况下每个请求发现的平均 hop 数

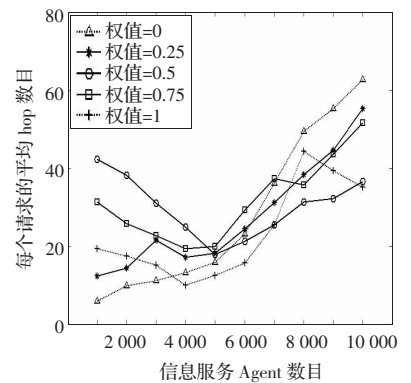


图3 在非平衡的资源分布和有偏向的请求方案下,变化的用户请求对发现性能的影响

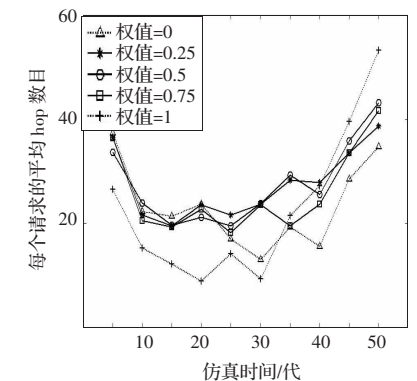


图4 在非平衡的资源分布和有偏向的请求方案下,不同信息服务 Agent 数目对发现性能的影响

检查 relationshipDescription 属性

if $I^k \notin I_j^{col}$

将 I^k 加入 I_j^{col}

Else

根据公式 $\delta_j^{k+} = (\delta_j^{k-} + \gamma) / 2$ 更新用户评价

end if

if $\gamma \geq 0$

$$trust_{i,j}^+ = trust_{i,j}^- (1 - \gamma^2) + \gamma^2$$

Else

$$trust_{i,j}^+ = trust_{i,j}^- (2 - \frac{1}{1 + \gamma})$$

end if

3 仿真分析

为了评价该方法,在网络平台上开发了一个离散事件仿真器。网络平台支持可“即插即用”的功能,提供了一套易用的编程的 API。这有助于该方法仅做很小的改动便可以用于实际的部署。与资源发现相关的仿真设置描述如下:分布式网络环境中很多因素都会影响资源发现方法的性能。在初始的关系网络的拓扑结构之上,通过仿真在以下几个方面评价该方法的性能:(1)资源分布;(2)用户请求模式;(3)信息服务 Agent 的数目和可靠性。

考虑到仿真实验和发现方法中的随机性问题,多次重复实验,因此本章所给出的结果为多次运行的测量平均值。

首先考虑在一定的信息服务 Agent 数目(5 000)、无偏向的

用户请求方案下,在平衡和非平衡两种资源分布环境中,不同的 η 值对发现性能的影响(用 hop 数来衡量,它是一个衡量节点数量的单位),如图 1 所示。Agent 数目为 5 000, TTL 值定为 200,作为比较,在图 1 中也给出一个随机传递方法的性能测试结果。

然后,检查了采用有偏向的用户请求方案,资源平衡分布和不平衡分布两种情况下, η 取不同值时的发现性能,如图 2 所示。其中 Agent 数目为 5 000, TTL 值定为 100。

进一步评价了动态用户请求对发现的影响。按照以下方式逐步改变用户请求(有偏向的用户请求集):每 10 代为一个时间间隔,在每个间隔的开始分别以概率 $\xi=0$ (用户请求无变化)、0.002、0.02、0.2、1 (用户请求完全变化)改变用户请求集,结果如图 3 所示。信息服务 Agent 的数目为 5 000。TTL 的值定位 100。当仿真时间为 10、20、30、40、50 代时,用户请求变化的概率分别为 0、0.002、0.02、0.2、1。在 50 代仿真时间内,不同资源分布和用户请求方案下,仿真网络大小从 1 000 更改到 10 000 来评价关系网络大小(也就是信息服务 Agent 的数目)对发现方法的影响。图 4 显示了非平衡环境下,有偏向的用户请求方案的关系网络大小对发现方法影响的结果。TTL 值设为 100。

4 结束语

社会网络是一个自然的、非常有效的帮助人们获得信息的方法。应用社会网络中的部分原理和概念来设计动态的资源发现方法。在仿真实验中,首先将所提出的发现方法与随机传递方法作比较,比较结果表明该方法能够形成关系聚类,并能明