

基于改进蚁群算法的可信服务发现

盛国军^{1,2}, 温涛^{1,2}, 郭权², 宋晓莹^{1,2}

(1. 东北大学 软件中心, 辽宁 沈阳 110004; 2. 大连东软信息学院 辽宁省网络安全与计算技术重点实验室, 辽宁 大连 116023)

摘要: 针对非结构化 P2P 网络中的服务发现问题, 提出了一种基于改进蚁群算法的可信服务发现方法。该方法在传统蚁群算法基础上应用若干新的策略控制蚁群的行为, 如动态蚁群策略、子蚂蚁策略、服务节点的信誉评估策略以及蚂蚁的恶意节点惩罚策略等, 高信誉服务节点在蚂蚁的推荐下进行邻居更换和自组织。这些策略和方法为蚁群算法在分布式环境下服务发现的应用研究提供了一种新思路。实验结果表明, 本算法在各种常见复杂网络环境下仍可保证较高的可信服务查准率和更高的综合服务发现效率, 通过对实验数据的分析和解释得到了若干有益的结论, 为进一步的研究工作奠定了基础。

关键词: 服务发现; 自组织网络; 可信服务; 动态蚁群策略; 子蚂蚁策略

中图分类号: TP311

文献标识码: A

文章编号: 1000-436X(2013)10-0037-12

Trustworthy service discovery based on a modified ant colony algorithm

SHENG Guo-jun^{1,2}, WEN Tao^{1,2}, GUO Quan², SONG Xiao-ying^{1,2}

(1. Software Center, Northeastern University, Shenyang 110004, China;

2. Liaoning Key Lab of Network Security and Computing Technology, Dalian Neusoft Information Institute, Dalian 116023, China)

Abstract: Aiming at the problem of service discovery in unstructured P2P network, a trustworthy service discovery method based on a modified ant colony algorithm was proposed. Some novel policies were introduced to control the behavior of the ant colony, such as the dynamic ant colony policy, the sub-ant policy, the reputation evaluation policy to service peers, and the punishment policy to malicious peers. Service peers with high reputation could change their neighbors and make self-organization under the recommendation of ants. These policies and methods provide new ideas for the application researches of ant colony algorithms in the field of distributed service discovery. The results of experiments show that the proposed algorithm can guarantee higher trust service precision and better integrated service discovery performance in complex network environments, and some useful conclusions were obtained through the analysis and explanation of the experimental data, laying a solid foundation for further researches.

Key words: service discovery; self-organizing network; trustworthy service; dynamic ant colony policy; sub-ant policy

1 引言

P2P 网络中的各节点是逻辑对等的, 各节点自治又相互合作。可以把 internet 看成一个 P2P 系统, 系统中的组件通过互相提供服务进行对等交互。网络中任何节点都可以发布自己的服务, 服务消费者在面临网络中存在的大量功能相同或相似、服务质量不同的服务, 由于缺乏交互经验而使其面临着

不确定的交易风险, 在正式调用目标服务之前, 服务消费者无法在可行的时间和成本范围内对这些第三方提供的 Web 服务进行完备的测试, 从而很难得知目标服务是否安全可靠, 这为 P2P 环境下的可信服务发现提出了更高的要求, 即 P2P 系统应具有一个能使节点间相互帮助和协作来发现高信誉服务节点的方法。在大规模开放的 P2P 系统中, 采用传统的服务等级协议或第三方中心节点来保证服

收稿日期: 2013-01-28; 修回日期: 2013-04-22

基金项目: 国家自然科学基金资助项目(61170168, 61170169)

Foundation Item: The National Natural Science Foundation of China (61170168, 61170169)

服务质量的方法耗时耗力,且运行成本很高,而信任和信誉^[1]机制更加适用于大规模开放 P2P 系统中可信资源的管理,近年来在学术界和工业界受到极大关注。EigenTrust^[2]是典型的全局信任模型,该模型通过节点间信任度的迭代为每个节点计算全局信任值,网络中节点的全局信誉由与其交互过的其他节点对该节点的评价聚合而成,该模型还假设 P2P 网络在创建时就具有全局高信任值的节点存在,其在整个 P2P 网络范围内的迭代将导致巨大开销,这是大部分基于 P2P 网络的全局信任模型所具有的共同问题。泛洪(flooding)是传统非结构化 P2P 网络资源查找的典型路由算法,其盲目的消息转发方式造成的通信开销十分巨大,对于大规模 P2P 网络,即使采用 TTL 限制的广播方式也会造成大量网络资源浪费。而蚁群算法^[3](ACS, ant colony system)由于具有良好的自适应性和动态自组织性,能较好地解决大规模 P2P 网络中的路由问题。

本文为解决上述非结构化 P2P 网络中的可信服务定位问题,将蚁群算法与 P2P 网络中的信誉机制相结合,提出了一种基于自组织 P2P 网络与改进蚁群算法的可信服务发现方法。该方法在传统蚁群算法基础上应用若干新的策略控制蚁群的行为,高信誉服务节点在蚂蚁的推荐下进行邻居更换和自组织。实验结果表明,该算法可在各种常见复杂网络环境下保证较高的可信服务查准率,且可以大幅度降低蚂蚁网络漫游的总跳数,从而使算法的单跳查询效率得到大幅提升。

2 相关工作

文献[4]提出一种基于 P2P 网络和语义技术的服务发现方法,该方法将 P2P 网络中语义相似的节点聚类成语义群,每个服务查询请求按照语义匹配度被路由到特定的语义群以提高查询效率。文献[5]提出一种基于结构化 P2P 网络的服务发现方法,该方法采用增强的 Chord 协议,用分层次的多环取代传统 Chord 协议中的单环,根据打分选择出服务提供能力强的节点组成超环来管理 P2P 网络。文献[6]提出了一种基于多 agent 系统的分布式服务发现框架,该系统中的 agent 可根据自己的服务查询偏好和效用进行邻居更新和自组织以提高服务查询的成功率。潘静等^[7]提出一种基于声誉的推荐者发现方法,该方法通过信任的传递计算具有高声誉值的服务推荐者,其研究重点是发现可信的服务推荐

者。谢晓芹等^[8]提出一种基于推荐网络的服务发现方法,该方法使用推荐有效地提高了服务发现的效率,引入蚁群算法并使用节点社交度和服务信息度引导蚂蚁的漫游过程,但该模型没有考虑 P2P 网络中恶意服务提供者对服务发现结果造成的影响。郑啸等^[9]提出一种基于移动 agent 的服务发现机制,该方法中,SearchAgent 从无结构化 P2P 网络中具有服务注册功能的中心服务节点出发与 GuideAgent 合作寻找目标服务,该方法仍然缺少对可信服务发现问题的考虑。FELIX G M 等^[10]提出了一种 P2P 网络环境下基于蚁群算法的可信服务发现模型 TACS,在该模型中,蚂蚁发现的高信誉服务节点对应路径上的信息素得到增强,恶意服务节点对应路径上的信息素值将会得到惩罚,该模型的主要问题在于:在每轮迭代后,服务请求节点要通过对蚂蚁发现的服务的实际调用结果来评价该节点的信誉类型,多次迭代将导致大量的服务调用成本,该设计不合理也不具有实践可操作性;另外 TACS 模型对应实验中的 P2P 网络只有几十个节点以内的数量级,节点数量太少导致蚂蚁的路由复杂度和难度过于简单从而缺乏对算法性能的有力支持。

在以上相关研究的基础上,本文提出了一种非结构化 P2P 网络中的可信服务发现方法,该方法将蚁群算法与 P2P 网络中的信誉机制相结合,蚂蚁利用其漫游路径上的服务消费者节点对相关服务的历史调用信息推导出服务提供者的局部信誉视图,并将其用于对相关服务提供者的信誉评估中。通过常见的各种类型大规模 P2P 网络的仿真实验表明,该方法能获得较高的可信服务查全率和查准率。

3 问题描述

定义 1 信誉: Josang 将信誉定义^[11]为“信誉是群体关于某人或某事的特征或立场的大众观点和看法。”

定义 2 服务的局部信誉: P2P 服务网络中,所有该服务消费者节点集合中的部分节点对该服务结果的总体评价和看法。

定义 3 服务交易日志是一个四元组 $STL=(sp, kws, time, sat)$, sp 是服务提供者节点标识, kws 是被调用服务的关键字标识, $time$ 是服务调用时间, $sat [-1, 1]$ 是服务消费者调用完服务后对该服务结果的满意度, 当一个节点调用另一个节点的服务后, 服务请求者节点中的 agent 就会生成一个新的

服务交易日志并保存在该节点的日志库中。

定义 4 蚂蚁是 P2P 网络中的移动 agent, 可以定义为一个八元组: $A=(id,tl,P,fv(P),et,grt, fr(et,grt), TTL)$, 其中, $id \in N^+$ 是蚂蚁的标识, tl 是蚂蚁路由禁忌表, 用于记录蚂蚁在一次迭代构造解的过程中已访问过的节点, $P=\{p_1, p_2, \dots, p_d, \dots, p_n\}$ 是蚂蚁完成一次完整迭代后构造的解, $fv(P)$ 是 P 的评价函数, 其返回值代表蚂蚁的路径质量, $et=(sp, kws, eRep)$ 是蚂蚁在当前迭代的漫游过程中收集到的各个服务消费者节点对多个服务的评价迭加表, 通过该表可以计算出多个消费者节点对某服务的综合看法, 蚂蚁将途经的各节点中关于服务节点 sp 中以 kws 标识的服务的交易日志 $STL.sat$ 进行时间加权迭加, 形成蚂蚁对服务的局部看法; $grt=(sp,kws,eRep)$ 是到目前迭代为止蚁群中所有蚂蚁在以前迭代中得到的各节点关于各服务的信誉表, $fr(et,grt)$ 函数用于评估服务节点的综合信誉值。局部信誉信息表 et 和 grt 具有相同的表结构, 其前 2 个字段为主键, 最后一个字段 $eRep$ 是服务的信誉值, 两者的区别为: et 存储的是单只蚂蚁在单次迭代中的服务信誉信息, 而 grt 存储的是所有蚂蚁在所有迭代中收集到的服务信誉信息, 在每次迭代后由每只蚂蚁的 et 按相同主键加权迭加合并而成, TTL 是蚂蚁的最大网络跳数限制。

定义 5 蚁群是一个六元组 $AC=(ASet, C, c, A_g^*, ACPS, Policies)$, 其中, $ASet=\{A_1, A_2, \dots, A_k, \dots, A_m\}$ 表示 m 个蚂蚁的集合, C 是整个蚁群的总迭代次数, $c \in [1, C]$ 是蚁群的当前迭代次数且 c 以步长 1 递增, $ASet$ 中各蚂蚁具有相同进化代数及相同的路径评价函数, $A_g^* \in ASet$ 为蚁群本次迭代中获得最优解的蚂蚁, $ACPS$ 是蚁群路径相似度指标, $Policies=\{pInfoCollect, pInfoExchange, pInfoCombine, pRepEvaluate, pRouteSelect, pDynAnts, pSubAnt, pPheUpdate, pPunishment\}$ 是蚁群的策略集合, 用于控制蚁群中所有蚂蚁的行为, 该集合包括信誉信息采集策略 $pInfoCollect$ 、信息交换策略 $pInfoExchange$ 、信息合并策略 $pInfoCombine$ 、对服务节点的信誉评估策略 $pRepEvaluate$ 、蚂蚁路由选择策略 $pRouteSelect$ 、动态蚁群策略 $pDynAnts$ 、子蚂蚁策略 $pSubAnt$ 、信息素更新策略 $pPheUpdate$ 以及蚂蚁的恶意节点惩罚策略 $pPunishment$, 同一个蚁群中每只蚂蚁都继承所在蚁群的所有策略。

定义 6 节点是一个七元组, $Peer=(id, stlSet, wsSet, kwsSet, neighbors, Policies, Agent)$, 其中, id 为节点的标识, 在真实的 P2P 网络中即为节点的 IP 地址, $stlSet=\{STL_1, STL_2, \dots, STL_n\}$ 是该节点的服务交易日志表, 记录该节点的历史服务调用经验, $wsSet=\{ws_1, ws_2, \dots, ws_m\}$ 是该节点提供的服务列表, 其中的每个服务以一个关键字向量来标识, 并将 $wsSet \neq \emptyset$ 的节点称为服务提供者节点(SPP, service provider peer), $kwsSet$ 是与 $wsSet$ 中每个服务相对应的服务关键字集合, $neighbors$ 是该节点的邻居节点列表, $Policies=\{pServicePub, pSTLStore, v pNeibUpdate\}$ 是该节点的策略集合, 包括该节点的服务发布策略 $pServicePub$ 、服务交易日志存储策略 $pSTLStore$ 以及邻居更新策略 $pNeibUpdate$, $agent$ 是运行在该节点上的智能体, 负责管理该节点的 $stlSet$ 、与经过该节点的移动 agent 交互以及执行该节点的策略集合中的策略。

定义 7 自组织网络 $SON=(Peers, R, AC)$, 其中, $Peers=\{Peer_1, Peer_2, \dots, Peer_n\}$ 是网络中 n 个节点的集合, $R=\{rNeighbor, rInvocation, rRecommend\}$ 是网络中关系的集合, 包括节点间邻居关系 $rNeighbour \subset (Peers \times Peers)$, 节点间调用关系 $rInvocation \subset (Peers \times Peers)$, 蚁群 AC 中蚂蚁与节点之间的推荐关系 $rRecommend \subset (AC.ASet \times Peers)$ 。网络中的各个节点是自治的, 可以指定并执行自己的邻居更新策略, 节点通过添加邻居节点加入到自组织网络, 服务提供者节点向其直接邻居发布自己提供的服务。

4 TSDMACS 算法设计

策略是一系列规则的集合, 用于约束相关网络实体的行为。TSDMACS(trustworthy service discovery with modified ant colony system)算法通过为网络中的节点和移动 agent 制定相关的策略来控制算法的行为, 各策略可以看作是 TSDMACS 算法的子算法。

4.1 节点的服务发布策略(pServicePub)

新加入自组织网络 SON 的服务提供者节点 SPP 必须执行其服务发布策略, 通过自己的直接邻居向整个网络发布其所提供的服务, 即在 $SPP.neighbors$ 集合的每个节点中记录一个三元组 $\langle SPPid, kws, m \rangle$ 作为一个服务的发布标记, 其中 kws 是服务节点所提供服务的关键词, 当节点间邻居关系解除时, 对应的服务发布信息被删除。 m 是蚂蚁路由过程中

的启发信息且在蚁群的整个迭代过程中其值保持不变。

4.2 节点的服务交易日志存储策略(pSTLStore)

当 SON 网络中的节点 Peer_i 调用节点 Peer_j 提供的服务后, 两者之间就形成了一个调用关系, 即 $\langle Peer_i, Peer_j \rangle \in SON.R.Invocation$, Peer_i. Agent 在 Peer_i. stlSet 中保存一个服务交易日志 STL 来记录这样一个调用关系, 当调用的是一个恶意服务节点提供的虚假或有害服务时, 根据其危害程度, Peer_i. Agent 将一个 [-1,0) 之间的值赋给 STL.sat, 对于提供正常功能的服务, STL.sat 的取值范围为 [0,1], 其大小表示用户对服务调用后的满意度大小, 从一定程度上反映了调用时刻该服务的质量水平。当有蚂蚁到达节点 Peer_i 时, 蚂蚁通过与 Peer_i. Agent 进行交互来收集 Peer_i. stlSet 中保存的历史服务交易记录。

4.3 蚂蚁信息采集策略(pInfoCollect)

在 TSDMACS 算法的迭代中, 当蚂蚁 A_k 路过一个节点 Peer_i 时, A_k 向 Peer_i. Agent 询问是否具有其所查询的关键字所对应的服务交易记录 STL 若存在, 则将该 STL 中的信息经过计算后添加到 A_k 的服务评价表 A_k. et 中, 添加的过程涉及到 STL 所创建时间的影响, 即时间越晚的 STL 对综合信誉的影响越大, 为此, 设计了时间加权评价值: $timeWeightedSat = (d^{\Delta t} \times STL.sat)$, 其中, $\Delta t = (tNow - STL.time)$ 即当前时间与 STL 创建时间之间的差值, 常量 $d \in (0,1)$ 用于控制时间差的影响程度, 其值越小则离现在时间越长的 STL 的权重越小。设 A_k 在沿途各节点收集到的关于服务节点 sp 中以 kws 标识的服务 ws 的交易日志集合为 $STLSet = \{STL_1, STL_2, \dots, STL_n\}$, 蚂蚁对指定服务的信誉评价计算公式为

$$(A_k.et.eRep)_{ws} = \frac{1}{n} \sum_{i=1}^n (d^{\Delta t} \times STL_i.sat)$$

若 A_k 在沿途节点中没有收集到某个服务 ws 的任何交易日志信息, 则 A_k 对服务 ws 不做信誉评估, 该服务的信誉类型对蚂蚁 A_k 为未知。

4.4 蚂蚁信息交换策略(pInfoExchange)

单只蚂蚁收集服务交易日志信息的能力有限, 在 TSDMACS 算法的同一轮迭代中, 蚁群中的所有蚂蚁彼此之间具有一定的概率到达相同的节点, 这为蚂蚁之间的直接信息交流提供了可能。在传统蚁群算法中, 蚂蚁之间仅利用信息素进行间接信息交

流, 前轮迭代中的蚂蚁将包含特定含义的信息“写”到其经过的路径上, 其他后来经过相同路径的蚂蚁从路径上“读”出该信息来指导自己的路由选择行为。观察真实世界中蚂蚁的行为可以发现, 蚂蚁彼此之间还存在着另外一种更直接的信息交流方式, 即当一只蚂蚁与另一只蚂蚁碰面时, 两者之间存在一定的概率彼此用 2 根触角触碰一下, 这样一个接一个的信息传递方式使得更多的蚂蚁受到刺激来共同协作完成一个任务。结合非结构化 P2P 网络中服务定位任务的实际特点, TSDMACS 算法中的人工蚂蚁除了用信息素间接交流外, 还引入类似真实世界中蚂蚁的直接信息交流方式以加强蚂蚁之间更加紧密的合作, 即让在同一轮迭代中到达同一个节点的蚂蚁彼此之间以概率 1 直接进行服务交易日志信息交换, 该设计可以通过蚂蚁的运动促使服务的信誉信息在 P2P 网络中快速传播, 从而尽快地以较低的网络跳数定位可信的服务提供者节点。设 et_i 是 Ant_i 沿途收集到的关于服务 ws 的评价记录, 则具体的信息交换规则为

$$\begin{aligned} & \text{if} (Ant_i.et_{i,ws} \text{ not Exists in } Ant_j.et) \\ & \quad \text{add } Ant_i.et.ev_{i,ws} \text{ to } Ant_j.et; \end{aligned}$$

其含义是 Ant_i 和 Ant_j 彼此将自己拥有的而对方不存在的服务评价信息复制给对方。

4.5 蚂蚁的信息合并策略(pInfoCombine)

在蚁群的一轮迭代后、下一轮迭代开始前, 各蚂蚁要将各自收集到的服务评价信息在发起查询的蚁巢节点进行汇总生成各服务的综合信誉信息表 grt, 并将 grt 赋值给每一只参加下一轮迭代的蚂蚁, 即

$$\text{for}(k=1 \text{ to } m)\{ A_k.grt = grt;\}$$

m 为蚁群中参加下一轮迭代的蚂蚁个数, grt 中保存了到当前轮为止所有迭代过程中整个蚁群所发现的服务的信誉信息, 若 grt 中已存在关于同一个服务 ws 的信誉信息 grt.eRep_{ws}, 则将后来的蚂蚁所携带的该服务的信誉信息进行加权后迭加到原来在 grt 中已经存在的记录上, 如下:

$$grt.eRep_{ws} = (a \times grt.eRep_{ws}) + (b \times A_k.et_{ws}.eRep)$$

$a, b \in (0,1) \wedge (a + b) = 1$, 本文取 $a = 0.4, b = 0.6$, 表明算法更加侧重于蚂蚁后收集到的服务信誉信息。

4.6 蚂蚁的服务节点信誉评估策略(pRepEvaluate)

当蚂蚁 A_k 在其漫游过程中到达一个与其查询

关键字匹配的服务提供者节点后，将启动其节点信誉评估策略以评估当前服务节点及其邻居中的服务节点的信誉类型，具体步骤如下：

Input : $A_k.et, A_k.grt$, 待评估的服务提供者节点 SPP

Output : 节点 SPP 的信誉类型 $repType, repType \in \{High, Common, Malicious, Unknown\}$

```

if(  $SPP$  exist in ( $A_k.et$  or  $A_k.grt$ ) ){
     $eRep_{spp} = A_k.fr(A_k.et, A_k.grt)$ 
    if( $eRep_{spp} < j$  and  $eRep_{spp} - 1$ ) return Malicious;
    if( $eRep_{spp} > j$  and  $eRep_{spp} < s$ ) return Common;
    if( $eRep_{spp} < s$  and  $eRep_{spp} - 1$ ) return High;
} else return Unknown ;
    
```

A_k 首先查询其 $A_k.et$ 和 $A_k.grt$ 中是否包括 SPP 的信誉信息，若在两者中都存在，则取两者加权迭加后的信誉值；若在 $A_k.et$ 和 $A_k.grt$ 中的一个存在，则直接取其存在的信誉值；若两者中都不存在，则对当前节点的信誉评估失败，节点的信誉类型为 Unknown，结束评估过程。常数 $j = 0, s = 0.1$ 。

4.7 蚂蚁路由选择策略(pRouteSelect)

TSDMACS 算法采取动态变化的伪随机比例选择参数 q_0 ， q_0 取值按照式(1)进行。

$$q_0 = 1 - \sqrt{\frac{cQ_{max}}{C(1+C-c)}} \quad (1)$$

其中， c 为蚂蚁的当前迭代次数， C 为蚂蚁迭代的总次数， $Q_{max} = 0.999$ 是一个预先给定的常量， q_0 的函数曲线如图 1 所示。

蚂蚁在选择下一个节点之前先生成一个 $[0, 1]$ 区间中的随机数 $rand$ ，然后将 $rand$ 与 q_0 的值进行比较，并且根据比较结果进行路由选择。TSDMACS 蚂蚁路由选择算法描述如下：

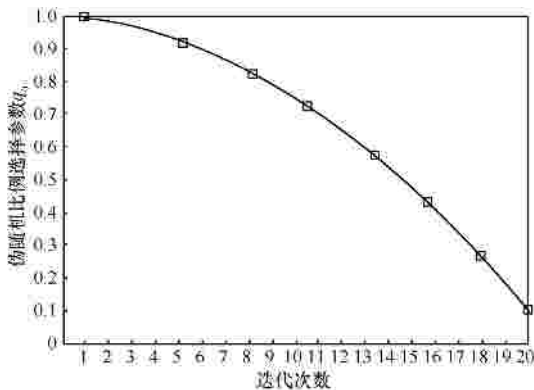


图 1 伪随机比例选择参数的函数曲线

Input：蚂蚁 A_k ，网络 SON，信息素与启发信息重要性参数 a 和 b 。

Output：蚂蚁 A_k 将要访问的下一个邻居节点 nextNode。

- 1) $neibSet = A_k.P.currentNode.neighbors$ 。
- 2) $nextNode = getHighRepNeighbor(neibSet)$; // A_k

用收集到的信誉信息对当前节点的邻居节点进行信誉评估，若存在未访问的高信誉节点则返回；否则，执行 3)。

3) 对当前节点的邻居集 $neibSet$ 中的节点进行信誉评估，若发现信誉低的恶意服务节点则排除。

- 4) 按照式(1)计算伪随机比例选择参数 q_0 。

5) $rand = makeRand()$ //生成一个 $[0, 1]$ 之间的随机数 $rand$ 。

6) if($rand > q_0$) { //蚂蚁 A_k 按照式(2)^[12]中的随机比例规则确定的转移概率 $P_{ij}^k(t)$ 选择其下一跳的邻居节点 nextNode。

$$P_{ij}^k(t) = \begin{cases} \frac{t_{ij}^a(t)m_{ij}^b(t)}{\sum_{s \in A_k.tl} t_{is}^a(t)m_{is}^b(t)}, & j \notin A_k.tl \\ 0, & \text{其他} \end{cases} \quad (2)$$

return nextNode;

} else { //蚂蚁 A_k 按照式(3)^[3]的伪随机比例规则选择其下一跳的邻居节点 nextNode。

$$s = \begin{cases} \operatorname{argmax}_{u \in A_k.tl} t_{ru}^a(t).m_{ru}^b(t), & \text{若}(rand < q_0) \\ S, & \text{其他} \end{cases} \quad (3)$$

return nextNode;

}

式(2)和式(3)中 $A_k.tl$ 是蚂蚁 A_k 的禁忌表。由上式可知，转移概率 P_{ij}^k 与 $t_{ij}^a(t).m_{ij}^b(t)$ 成正比，参数 a 和 b 分别反映了信息素信息 $t_{ij}(t)$ 和启发信息 $m_{ij}(t)$ 的相对重要程度。由该路由算法可见，随着蚁群迭代次数的增加及 q_0 值逐渐变小，蚂蚁路由选择的随机性变大，从而降低了蚂蚁在迭代后期路径重叠的概率。

4.8 动态蚁群策略(pDynAnts)

动态蚁群策略即蚁群的动态蚂蚁数量策略。与传统的蚁群算法不同，为节省网络带宽，提高单跳查询效率，TSDMACS 算法在迭代过程中采用动态蚂蚁数量策略和子蚂蚁策略，该设计是基于蚁群算法结合 P2P 服务发现业务的特点而考虑的，即蚁群

算法总是倾向于将最优解路径上的信息素加强,在迭代的中后期由于这种正反馈的作用,蚁群中的大多数蚂蚁将集中于最优解路径附近震荡或与最优解路径的边重叠,导致蚁群早熟而停止进化,发现新的可信服务节点的概率降低,这相当于浪费了大量的蚂蚁跳数资源。TSDMACS 算法在每一轮迭代开始之前,动态调整参与本轮网络漫游的蚂蚁数量,使蚂蚁数量随迭代次数的增加动态减少。动态蚂蚁数量计算方法为

$$m_c = \left\lceil 1 + (m - 1) \left(1.0 - \frac{AC.c}{AC.C} \right) \right\rceil \quad (4)$$

其中, m 为蚁群 AC 的初始蚂蚁数量, $AC.C$ 为蚁群总迭代次数, $AC.c$ 为蚁群的当前迭代次数。

4.9 节点的邻居更新策略(pNeibUpdate)

TSDMACS 算法采用基于信誉评估的邻居更新策略,即蚂蚁 A_k 到达一个服务提供者节点 SPP 后,根据自己收集到的其他服务节点的信誉信息向 SPP 进行邻居推荐,具体步骤如下。

1) A_k 首先采用服务节点信誉评估策略对当前服务节点进行信誉评估。

2) 若对 SPP 的评估结果为高信誉节点,则进入 3) 进行邻居推荐,否则,该策略结束。

3) 蚂蚁 A_k 根据 $A_k.et$ 和 $A_k.grt$ 中收集到的信誉信息计算各个服务提供者节点的信誉值并形成服务提供者列表 $spList$ 。

4) 蚂蚁 A_k 检索 $SPP.neighbors$ 中是否存在 $spList$ 中信誉低的恶意节点,若存在则向 $SPP.Agent$ 建议从邻居中删除这些恶意节点, $SPP.Agent$ 执行删除操作。

5) 蚂蚁 A_k 随机从 $spList$ 中选取 r_1 个高信誉节点(若存在)向 $SPP.Agent$ 推荐作为 SPP 的邻居节点。

6) 若 SPP 的邻居中不包含所推荐的节点,则 $SPP.Agent$ 接受推荐, $SPP.Agent$ 检查自己的邻居数量在添加新邻居后是否超过最大邻居数量限制,若超过限制,则删除一个普通的邻居节点,然后添加蚂蚁 A_k 所推荐的高信誉服务节点作为邻居。

4.10 蚁群的子蚂蚁策略(pSubAnt)

TSDMACS 算法中 SON 网络在蚂蚁的推荐下,进行基于信誉的邻居更新,从而实现了网络的自组织。在该机制的作用下,蚁群经过多轮迭代后, $P2P$ 网络中的大多数高信誉服务节点逐渐互联或者靠近,形成了一个由高信誉服务节点形成的“小世界

网络”,以共同对抗恶意服务节点群体。在蚁群迭代的中后期,高信誉服务提供者互联的概率逐渐增加,蚂蚁 A_k 在到达一个服务提供者节点后,不立即返回蚁巢,而是启动子蚂蚁(sub-ant)策略进行局部精细搜索,具体步骤如下。

1) A_k 先向蚁巢节点申请其可以生成的子蚂蚁数量 m_{sub} , 规定 m_{sub} 取值范围满足如下限制规则: $(m_{sub} - maxNeighbors / 2) \wedge (m_{sub} - l \cdot m - m_{other})$, 其中, $maxNeighbors$ 为节点的最大邻居数限制, $l \in N^+$ 为倍数因子, m 为初始蚁群中蚂蚁的数量, m_{other} 为本轮迭代中蚁群已经生成的子蚂蚁数量。

2) A_k 按照申请到的子蚂蚁数量创建 m_{sub} 只子蚂蚁,子蚂蚁继承了其母蚂蚁中除 $A_k.id$ 以外的所有属性,包括 $A_k.et$ 和 $A_k.grt$,并为子蚂蚁设置 TTL 属性,本文设置其 $TTL = 3$ 。

3) 各子蚂蚁按照式(2)的随机比例规则确定的转移概率选择其下一跳邻居节点 $nextNode$ 并且 TTL 值对应递减,当遇到服务提供者节点或者其 $TTL = 0$ 时结束漫游,子蚂蚁在漫游过程中不再创造下一代子蚂蚁。

4) 各子蚂蚁将得到的服务提供者节点及路径返回给其对应的母蚂蚁 A_k 后自动消亡。

子蚂蚁策略本质上是一种局部精细搜索策略,在可信服务节点小世界网络的高密度区域,有针对性地释放出子蚂蚁进行搜索,与从蚁巢节点出发相比,从当前可信服务节点出发的子蚂蚁消耗了更少的网络跳数,并且在可信服务的高密度区域搜索在很大程度上增加了可信服务的命中率,从而提高了整个蚁群的单跳服务发现效率。本文将从蚁巢节点出发并能释放出子蚂蚁的蚂蚁称作母蚂蚁。

4.11 蚂蚁信息素更新策略(pPheUpdate)

TSDMACS 算法中,蚂蚁路由过程中使用的启发信息 m 由加入 $P2P$ 网络的服务提供者节点向其邻居进行服务发布时生成,而蚂蚁路由中用到的另一个重要信息是信息素 t ,其值的大小是蚂蚁所要查询的目标服务的信誉度的函数 $t = f(eRep_{ws})$ 。与 MMAS 算法相似, TSDMACS 算法将蚂蚁路径上的信息素的变动范围设定为 $[t_{min}, t_{max}]$, 本文两者分别取值 $t_{max} = 0.999, t_{min} = 0.001$, 将各条有向边上的初始信息素设置为信息素最大值 t_{max} ; 蚂蚁在构造解的同时进行信息素的局部更新; 当蚁群 AC 的一轮

迭代 c 结束，第 $c+1$ 轮迭代开始前，蚁群要进行各蚂蚁的路径质量评估，并选出本轮迭代中最优蚂蚁进行全局信息素更新，没找到服务节点的蚂蚁不参加路径质量评估。蚂蚁 A_k 的路径质量计算公式为

$$A_k \cdot fv(A_k \cdot P) = \sqrt{\frac{Q(1 + eRep_{spp})}{length(A_k \cdot P)}} \quad (5)$$

其中， $Q=0.5$ 是一个预设置常量， $SPP = A_k \cdot P$ 。LastNode 是蚂蚁路径上最后一个节点， $eRep_{spp}$ 是该节点的信誉值，函数 $length(A_k \cdot P)$ 求蚂蚁 A_k 的路径长度。AC.A_g^{*} 全局信息素更新公式为

$$\begin{cases} t_{ij} = (1-r)t_{ij} + \Delta t_{ij} \\ \Delta t_{ij} = rSQ_g^* \end{cases} \quad (6)$$

其中， t_{ij} 表示蚂蚁 A_g^* 路径上边 $\langle i, j \rangle$ 上的信息素含量， $r \in (0,1]$ 是信息素挥发系数， $SQ_g^* = A_g^* \cdot fv(A_g^* \cdot P)$ 是最优蚂蚁 A_g^* 的路径质量。TSDMACS 算法的信息素局部更新公式为

$$t_{ij} = (1-j)t_{ij} + \frac{j}{Q_0} \quad (7)$$

其中， $j \in (0,1)$ 是信息素挥发系数， Q_0 是一个大实数常量。

4.12 蚂蚁的恶意节点惩罚策略(pPunishment)

TSDMACS 算法中，蚂蚁 A_k 对被发现和评估为恶意服务提供者节点的路径进行不对称惩罚，惩罚因子 pf 的计算公式为

$$pf = \left(\frac{ei}{ec(ec - ei + 1)} \right)^p \quad (8)$$

其中， ec 为 $A_k \cdot P$ 中边的数量， $ei \in \{1, 2, L, ec\}$ 为边的索引号， p 为幂形式调节参数，其值越大 pf 的函数曲线越陡峭，本文取 $p=1.2$ ， pf 的函数曲线如图 2 所示。对应的信息素惩罚公式为

$$t_{ij} = (1-j)t_{ij}(1-pf) + \frac{jSQ}{ec} \quad (9)$$

其中， $j \in (0,1)$ 是信息素挥发系数， ec 为 $A_k \cdot P$ 中边的数量。恶意节点路径信息素惩罚因子函数曲线如图 2 所示。由图 2 可见，TSDMACS 算法对 $A_k \cdot P$ 中越靠近恶意服务节点 ($A_k \cdot P.LastNode$) 的边的惩罚力度越大。

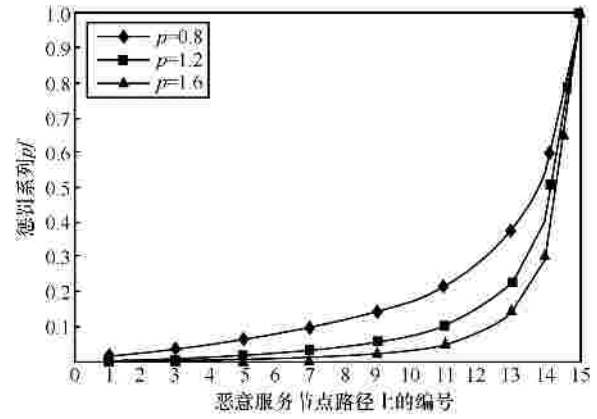


图 2 恶意节点路径信息素惩罚因子函数曲线 ($p=1.2, ec=15$)

4.13 TSDMACS 算法描述

算法 1 基于改进蚁群算法的可信服务发现算法 TSDMACS

Input: 发起服务查询的节点 queryClient (蚁巢), 被查询的目标服务的关键词 kws , 蚁群的初始蚂蚁数 m , 蚁群的总迭代次数 C , SON 网络

Output: 按照信誉排序的服务节点列表 $wsPeerList$

for $c=1$ to C

- 1) 应用动态蚁群策略，用式(4) 获得参与本轮迭代的母蚂蚁数量 m_c 。
- 2) $AC.ASet = createAnts(m_c)$: 创建 m_c 只母蚂蚁并且放置到 $queryClient$ 节点上，为每只蚂蚁装载上轮迭代汇总后的信誉表 grt ，将 $queryClient$ 添加到每只蚂蚁的路径和禁忌表中。

3) do:

for $k=1$ to m_c

蚂蚁 A_k 使用路由选择策略获得其下一跳的邻居节点 $nextNode$;

A_k 访问 $nextNode$ 并将 $nextNode$ 加入 $A_k.tl$;

if($currentNode$ is not $serviceProvider$) {

A_k 执行信息交换策略;

A_k 执行信息采集策略;

} else {

$currentNode.Agent$ 在蚂蚁 A_k 的推荐下执行邻居更新策略;

A_k 执行子蚂蚁策略;

}

蚂蚁 A_k 执行信息素局部更新;

endfor

until: ($each\ ant\ find\ a\ service$) or ($all\ neighbors$)

visited) or ($TTL = 0$)

4) 用式(5)评价每一只蚂蚁解的质量并且选择出本轮迭代的最优蚂蚁 A_g^* 以及发现恶意服务节点的蚂蚁集合 $maliciousAnts = \{A_1, A_2, \dots, A_p\}$ 。

5) A_g^* 执行全局信息素更新。

6) 对 $maliciousAnts$ 中的每只蚂蚁的路径进行惩罚。

7) 蚁巢 queryClient 执行信誉信息合并策略, 将各蚂蚁收集到的信誉信息进行汇总存储到各蚂蚁的 grt 表中

endfor

8) 将最后一轮迭代汇总后的信誉表 grt 中各个服务节点按照信誉的大小排序形成 $wsPeerList$ 并返回。

5 TSDMACS 算法的理论分析

5.1 TSDMACS 算法若干性质

定理 1 设 C 为蚁群总迭代次数, m 为蚁群初始 (最大) 蚂蚁数量, 则 TSDMACS 算法中所有轮迭代所有母蚂蚁消耗的总跳数 $totalHops$ 不超过

$$TTL \left(C + \frac{(m-1)(C-1)}{2} \right)$$

证明 设第 c 轮 ($c \in \{1, 2, \dots, C\}$) 迭代的母蚂蚁数量为 m_c , 令 $m_c = \lfloor R_c \rfloor$, 则 $m_c \leq R_c$; 在式(4)中,

令 $R_c = 1 + (m-1) \left(1 - \frac{c}{C} \right)$, 则所有迭代中的母蚂蚁

数量为

$$\begin{aligned} pAnts &= m_1 + m_2 + \dots + m_c = 1 + (m-1) \left(1 - \frac{1}{C} \right) + \\ & 1 + (m-1) \left(1 - \frac{2}{C} \right) + \dots + 1 + (m-1) \left(1 - \frac{C}{C} \right) \\ &= C + (m-1) \left(C - \frac{1+2+\dots+C}{C} \right) = C + (m-1) \left(\frac{C-1}{2} \right) \end{aligned}$$

即: $pAnts \leq C + \frac{(m-1)(C-1)}{2}$, 而每只母蚂蚁的最

大网络跳数限制为 TTL , 则有

$$totalHops \leq TTL \left(C + \frac{(m-1)(C-1)}{2} \right) \text{得证。}$$

定理 2 TSDMACS 算法中, 令 SPP^* 是 SON 网络中关于蚂蚁查询关键字 kws 的最佳服务提供者节点, SQ^* 为节点 SPP^* 到蚁巢节点的路径质量, 根据式(5), SPP^* 节点是所有相关服务提供者节点中拥

有最高信誉值以及到蚁巢节点的路径最短的服务节点; $c \in \{1, 2, \dots, AC, C\}$ 是蚁群的任一次迭代, r 是蚂蚁的全局信息素挥发系数, 则对于任一条边 $\langle r, s \rangle$ 有 $\lim_{c \rightarrow \infty} \lim_{rs} (c) (SQ^*/r)$ 。

证明 TSDMACS 算法中, 在任意一次迭代 c 之后, 本次迭代路径质量最优蚂蚁 A_g^* 要进行全局信息素更新, 由式(6)可知, A_g^* 在边 $\langle r, s \rangle$ 上新添加信息素量为 $\Delta t_{rs}^* = r A_g^* \cdot fv(A_g^*.P)$, 因为 $A_g^* \cdot fv(A_g^*.P)$

SQ^* 并且 $r \in (0, 1]$, 所以有 $\Delta t_{rs}^* \leq SQ^*$; 则在第 1 次迭代中, 边 $\langle r, s \rangle$ 上的最大信息素含量 $maxPhe_{rs}$ 为 $(1-r)t_0 + SQ^*$, 其中, t_0 为边 $\langle r, s \rangle$ 上的初始信息素含量, 在第 2 次迭代中, $maxPhe_{rs}$ 为

$$\begin{aligned} & (1-r)((1-r)t_0 + SQ^*) + SQ^* \\ &= (1-r)^2 t_0 + (1-r)SQ^* + SQ^* \end{aligned}$$

则在第 c 次迭代中 $maxPhe_{rs}$ 为

$$(1-r)^c t_0 + SQ^* \sum_{i=0}^{c-1} (1-r)^i$$

其中, $\sum_{i=0}^{c-1} (1-r)^i = \sum_{i=0}^{c-1} (1-r)^i$ 是一个公比为 $(1-r)$ 的等比数列的和, 因此在第 c 次迭代边 $\langle r, s \rangle$ 上的最大信息素含量为 $(1-r)^c t_0 + SQ^* \frac{1-(1-r)^c}{r}$; 又

因为 $(1-r) \in [0, 1)$ 使得 $\lim_{c \rightarrow \infty} (1-r)^c \rightarrow 0$, 则随着迭代次数 c 的持续增加该值将渐近收敛于 (SQ^*/r) , 即: $\lim_{c \rightarrow \infty} \lim_{rs} (c) (SQ^*/r)$, 得证。

5.2 TSDMACS 算法复杂度分析

设 C 为蚁群总迭代次数, m 为蚁群最大蚂蚁数量, 则各步骤的算法复杂度为: 获取动态蚂蚁数量 $O(1)$; 创建 m_c 只蚂蚁 $O(m_c)$; 执行单只蚂蚁路由选择策略 $O(\max(neighbours))$, 该值不超过网络节点所允许的最大邻居数量; 单只蚂蚁执行信息交换策略和信息采集策略 $O(p_1 \times (C_{ac}^2 |ic_1| + ic))$, 其中, $p_1 \in [0, 1]$ 为当前节点是目标服务提供者节点的概率, 具有不确定性, ac 为在同一轮迭代中同一个节点上碰面的蚂蚁的数量, 具有不确定性, $|ic_1|$ 为从对方蚂蚁获得的其不拥有的信誉记录的数量, ic 是蚂蚁从当前节点采集的信誉记录的数量; 当前节点 agent 执行邻居更新策略 $O(p_2 \times spc \times (r_1 + m_1))$, p_2 为当前节点是高信誉节点的概率, spc 为蚂蚁收集到的信誉记录的数量具有不确定性, r_1 为当前节点

添加的高信誉邻居节点的数量, m_i 为其删除的恶意邻居节点的数量, r_i 和 m_i 的取值均不超过网络允许的节点的最大邻居数量; 执行子蚂蚁策略 $O(m_{sub} \times pLen)$, m_{sub} 为申请到的子蚂蚁数量, 其取值范围为 $m_{sub} \in N^+ \wedge m_{sub} \in [0, maxNeighbours / 2]$, $pLen \in [1, TTL]$, 本文取 $TTL = 3$ 为子蚂蚁路径长度的上限, 蚂蚁执行信息素局部更新, 由于局部信息素更新是蚂蚁在路由的每一跳同时进行的, 因此其单只蚂蚁单跳的代价是 $O(1)$; 本轮迭代最优解蚂蚁执行全局信息素更新 $O(length(A_g^*.P))$, $A_g^*.P$ 为最优解蚂蚁的路径; 发现恶意节点的蚂蚁对其路径的信息素进行惩罚 $O(m_{ma} \times \overline{lengthP})$, m_{ma} 为进行路径信息素惩罚的蚂蚁的数量, $\overline{lengthP}$ 为这些蚂蚁的平均路径长度; 执行信誉信息合并策略 $O(spCounts_c)$, $spCounts_c$ 为本次迭代蚂蚁发现的服务的个数。忽略上述存在不确定性步骤的复杂度并略去低阶项对 TSDMACS 算法复杂度的级数进行估算

$$O\left(C \times pAnts \times \left(\overline{P_1} + \frac{maxNeighbours \times TTL_{sub}}{2}\right)\right) \quad (10)$$

其中, C 为总迭代次数, $pAnts$ 为所有迭代中所使用的母蚂蚁的总数量, $\overline{P_1}$ 是母蚂蚁的平均路径长度, 其大小是 P2P 网络规模、目标服务在网络中的密度以及可信服务节点组成的小世界网络节点间平均距离的函数; 由定理 1 可知

$$pAnts = C + \frac{(m-1)(C-1)}{2}$$

此处取所有子蚂蚁所可能消耗的最大代价, 即假设每只母蚂蚁都能创建 $maxNeighbours / 2$ 只子蚂蚁并且每只子蚂蚁都能完整地消耗掉其所拥有的 TTL_{sub} , 将 $pAnts$ 用其上限代入式(10)得

$$O\left(\frac{(2C^2 + (C^2 - C)(m-1)) \times (\overline{P_1} + N_0)}{4}\right) \quad (11)$$

其中, $N_0 = maxNeighbours \times TTL_{sub}$ 为常量, 令变量 $\overline{P_1} = N_1 C$, $(m-1) = N_2 C$, N_1 和 N_2 为任意实数, 则式(11)转换为

$$O\left(\frac{(2C^2 + (C^2 - C)N_2 C)(2N_1 C + N_0)}{4}\right) \Rightarrow O\left(\frac{2N_1 N_2 C^4 + (4N_1 + N_0 N_2 - 2N_1 N_2)C^3 + (2N_0 - N_0 N_2)C^2}{4}\right)$$

其中, 变量 C 为蚁群总的迭代次数, 其他项均为常量, 可见 TSDMACS 算法复杂度最高为 4 次幂级。

6 实验及结果分析

6.1 实验设计与评价方法

TSDMACS 算法利用蚂蚁在 P2P 网络中漫游并收集目标服务的历史信息, 因此在算法执行前必须在网络中生成相关服务的调用信息。本实验中, 采用多线程机制模拟服务请求者节点的服务调用。在 P2P 网络创建完成后、算法正式启动前, 实验平台创建并启动 5 个线程对象并发运行, 每个线程在其每次运行中都从网络中随机取得一个服务请求者节点 Peer, 对随机取得的一个服务提供者节点的相关服务进行调用, 并将调用结果以服务交易日志 STL 的形式保存在节点 Peer 中。本文引进最大最小蚁群(original MMAS)算法、TACS 算法及随机漫步(random walk)算法与 TSDMACS 算法进行对比。TSDMACS 是基于 Original MMAS 的改进, 与该算法进行纵向对比可以检测 TSDMACS 算法的改进效果, 而 TACS 算法是目前具有代表性的基于蚁群算法和 P2P 网络的可信服务发现方法, 与该算法进行横向对比可发现 TSDMACS 算法的性能提升效果, 而随机漫步算法则是 P2P 网络中资源发现最基本的方法, 本文将其作为其他 3 种算法的比较基础。相关指标定义如下。

定义 8 可信服务查全率(TSRecall, trustworthy service recall)为

$$TSRecall = \frac{|tsOutSet|}{|tsAllInP2P|}$$

其中, $|tsOutSet|$ 为到目前迭代为止算法检出的不同可信服务数量, $|tsAllInP2P|$ 为 P2P 网络中拥有的可信服务总数, 该值在真实的 P2P 网络中无法准确得知, 但在仿真实验中由于各种服务提供者节点的数量在网络初始化时已知, 因此可以用于评价算法的效率, 由定义可知, $TSRecall \in [0, 1]$ 。

定义 9 可信服务查准率(TSPrecision, trustworthy service precision)为

$$TSPrecision = \begin{cases} \frac{|tsOutSet|}{|wsAllOutSet|}, & |wsAllOutSet| > 0 \\ 0, & |wsAllOutSet| = 0 \end{cases}$$

其中, $|tsOutSet|$ 为到目前迭代为止算法检出的不同

的可信服务数量； $|wsAllOutSet|$ 为到目前迭代为止算法检出的不同服务总数，包括恶意服务数量和可信服务数量，当算法没有找到任何服务节点时， $TSPrecision = 0$ ，由定义可知， $TSPrecision \in [0,1]$ 。

6.2 实验环境

实验与开发环境为：HP Proliant DL380 G5, Intel(R) Xeon(R) CPU E5450 @ 3.00 GHz ,4 G RAM, Microsoft Windows Server 2003 Enterprise Edition Service pack 2 ,JDK1.6.0, Eclipse SDK Version:3.5.2, JVM 堆内存大小[- Xms128 M, - Xmx512 M] ,MATLAB 7.4.0(R2007a)。本实验在斯坦福大学开发的 P2P 模拟器 Query Cycle Simulator^[13]的基础上加入 TSDMACS 算法的代码进行仿真，算法部分采用 Java 语言实现，并在实验过程中将实验结果写入文本文件，然后用 MATLAB 从文件中读取数据做图。本实验中，P2P 网络的节点总数为 $peerCounts=10000$ ，善意服务提供者节点和恶意服务提供者节点的密度都为 0.4%，即各为 40 个节点，每个节点的初始邻居数为 4，节点的最大邻居数限制为 6。蚁群的总迭代次数 $C = 20$ ，蚁群初始蚂蚁数量 $m = 10$ ，信息素变动上限和下限分别取值 $t_{max} = 0.999, t_{min} = 0.001$ ，母蚂蚁的 $TTL = peerCounts \times 0.5\% = 50$ ，子蚂蚁的 $TTL = 3$ ，信息素 t_{ij} 的权重系数 $a = 1.0$ ，服务的启发信息 m_{ij} 的权重系数 $b = 1.0$ 。

6.3 实验 1 静态网络环境实验

静态 P2P 网络是指在蚁群整个迭代过程中网络节点数不发生变化。由于蚁群算法中涉及随机变量，使得单次实验结果具有一定的不确定性，因此重复执行本实验 50 次，每次实验迭代 20 次，取各实验中各指标相同迭代的平均值做图。可信服务查全率和可信服务查准率分别如图 3 和图 4 所示，由图可见，随着迭代次数的增加，TSDMACS 算法的 $TSRecall$ 和 $TSPrecision$ 指标始终高于其他对比算法，迭代结束时 $TSRecall$ 和 $TSPrecision$ 2 个指标值分别达到了 90.6% 和 96.65%，并且 TSDMACS 算法能在迭代早期达到较高的可信服务发现效率； $TSRecall$ 指标仅表明了算法的服务定位能力，而通过观察算法的 $TSPrecision$ 指标可以发现算法对好坏服务的区分能力。在图 4 中，TSDMACS 算法的可信服务查准率指标在整个迭代过程中的优势表现得更加明显，这是因为 TSDMACS 算法使用了基于服务调用日志的服务信誉评估机制，蚂蚁在当前节点进行路由选择时，会先根据服务调用日志对邻居节

点进行信誉评估，若发现邻居节点中有誉度较低的恶意节点则会主动避开，然后按照传统蚁群算法的路由选择算法进行路由选择，从而大幅度地提高了可信服务查准率；而 TACS 算法中使用了对恶意节点的信息素惩罚机制也具有一定的好坏服务区分能力，其可信服务查准率要明显高于 OriginalMMAS 和 RandomWalk 算法，OriginalMMAS 与 RandomWalk 算法中缺少可信服务的信誉评估机制，无法有效区分目标服务提供者节点的信誉类型，其可信服务查准率一直停留在 0.5 附近很难提高。

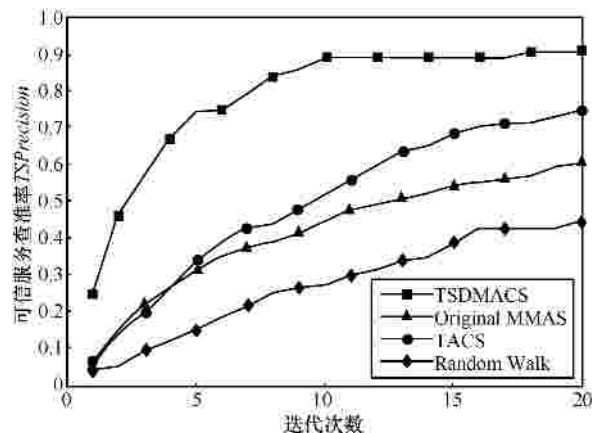


图 3 可信服务查全率 $TSRecall$

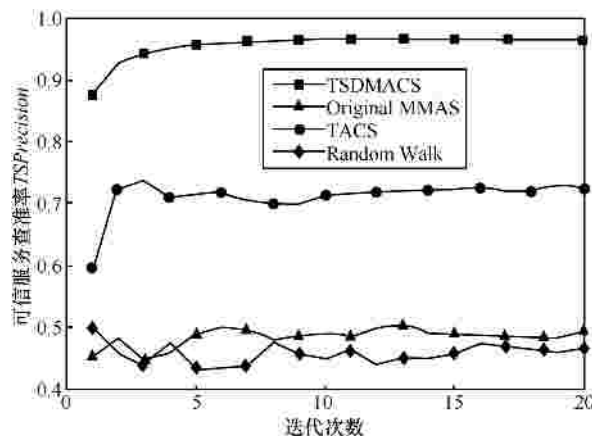


图 4 可信服务查准率 $TSPrecision$

6.4 实验 2 不同规模静态网络环境实验

为观察 TSDMACS 算法在不同规模的静态网络中的表现，进行了 10 次独立实验，实验中网络的规模从 10 000~28 000 以 2 000 为单位递增，保持网络中的服务数量以及好坏服务比例不变，每次实验迭代 20 次，取一次实验中相关指标在各次迭代中的平均值做图，实验结果如图 5 和图 6 所示。由图可见，随着网络规模逐渐扩大，各种算法的查全率都呈一定的下降趋势，这是因为网络

规模扩大而服务的绝对数量不变，相当于稀释了服务的密度，使蚁群发现新服务的难度加大。在 10 种不同规模的静态网络环境下，TSDMACS 算法的可信服务发现效率相对于其他几种对比算法仍具有明显优势，其可信服务查准率在 10 次实验中平均值为 92.32%，在网络规模迅速扩大的情况下仍可保持较高的查准率，该特性对于组合服务的执行成功率非常重要，因为在后续的服务选择过程中，组合服务抽象节点对应的候选服务集合中的服务数量变少对组合服务执行的影响相对不大，但是候选服务集合中质量差的恶意服务所占的比例的多少却对组合服务的执行成功率有着重要的影响。

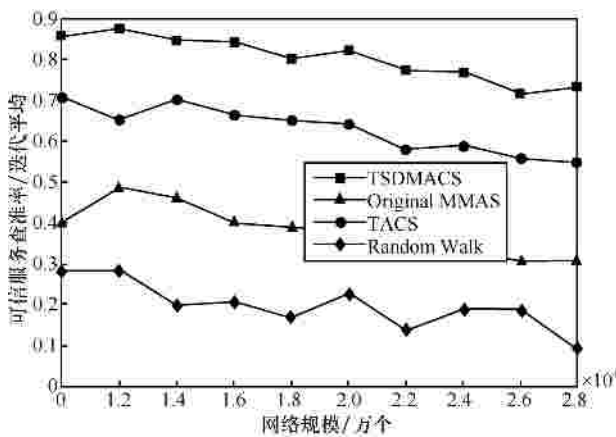


图 5 不同网络规模下的可信服务查全率迭代平均

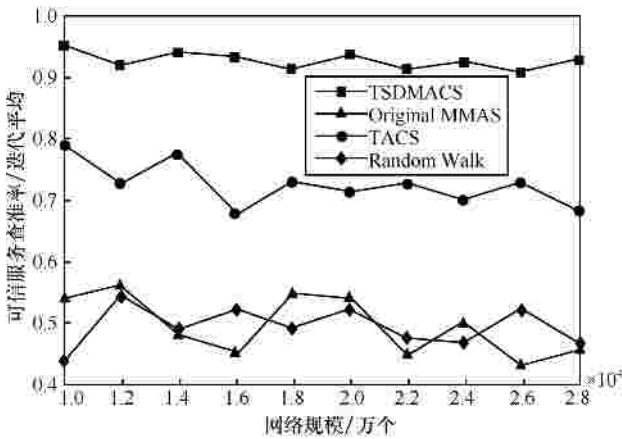


图 6 不同网络规模下的可信服务查准率迭代平均

6.5 实验 3 动态网络环境实验

在动态 P2P 网络中，任意节点可随时加入或离开网络，从而导致网络规模及拓扑结构的动态变化。本实验网络初始规模为 10 000，共迭代 20 次，保持网络中服务数量及好坏服务比例不变，每次迭

代随机选择 2 000 个非服务节点离开网络，同时又有 3 000 个新非服务节点加入网络以动态增加网络规模和破坏网络原有的拓扑结构，新加入网络的节点随机选择邻居连接，同时为网络中由于节点的频繁离开而产生的孤立节点随机选择邻居节点进行连接，实验结果如图 7 和图 8 所示。由图可见，与静态网络环境相比，在网络中有大规模节点频繁加入和离开的动态网络环境情况下，TSDMACS 算法仍可保持相对较高的可信服务查全率和查准率，迭代结束值分别达到 85.12%和 91.81%，并在整个迭代过程中其值始终高于其他对比算法，可见 TSDMACS 算法在动态网络环境下，仍然可以获得较高的可信服务查询效率，其若干策略的应用在动态网络环境中是有效的。

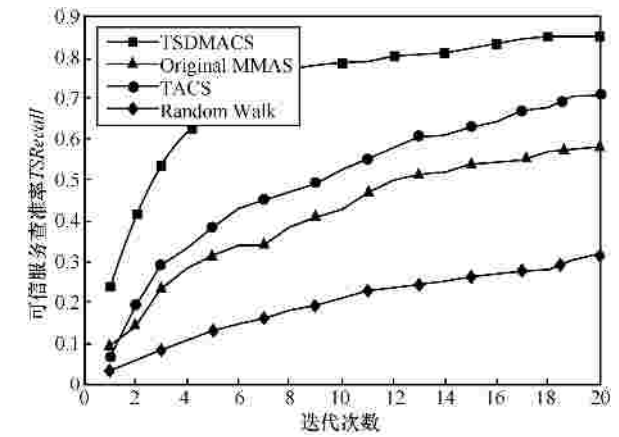


图 7 动态网络环境下的可信服务查全率 $TSRecall$

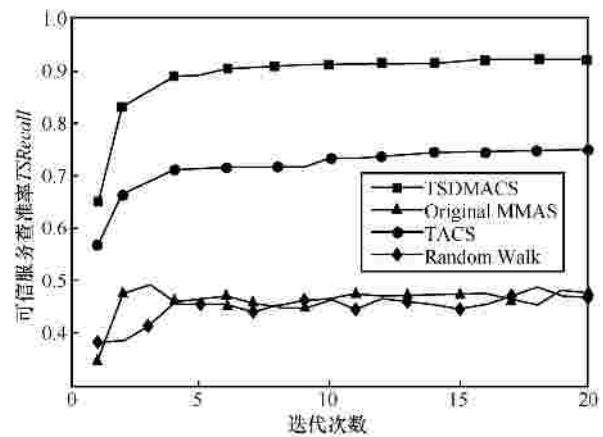


图 8 动态网络环境下的可信服务查准率 $TSPrecision$

7 结束语

本文针对非结构化 P2P 网络中的服务发现问题，提出了基于改进蚁群算法的可信服务发现方法

TSDMACS,该方法在传统 MMAS 算法基础上为蚁群添加若干新的策略,如动态蚂蚁数量策略、子蚂蚁策略、服务节点的信誉评估策略等,实验结果表明该算法能够很好适应动态 P2P 网络环境,相对于传统的蚁群算法和随机漫步算法具有更高的综合服务发现效率,该算法尤其对恶意服务节点的过滤效果明显,在各种常见的复杂网络环境下仍可保证较高的可信服务查准率,从而为组合服务的成功执行提供有效保障,实验结果表明 TSDMACS 算法在 MMAS 算法基础上的改进效果明显,算法中所提出的若干新的策略为蚁群算法在分布式环境中的应用研究提供了新思路。

参考文献:

- [1] TAVAKOLIFARD M, ALMEROOTH K C. A taxonomy to express open challenges in trust and reputation systems[J]. *Journal of Communications*, 2012, 7(7):538-551.
- [2] KAMVAR S D, SCHLOSSER M T, GARCIA-MOLINA H. The eigentrust algorithm for reputation management in P2P networks[A]. *Proceedings of the 12th International Conference on World Wide Web*[C]. New York, USA, 2003.640-651.
- [3] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach in the traveling salesman problem[J]. *IEEE Transactions on Evolutionary Computing*, 1997, 1(1):53-66.
- [4] MODICA G D, TOMARCHIO O, VITA L. A P2P based architecture for semantic Web service discovery[J]. *International Journal of Software Engineering and Knowledge Engineering*, 2011, 21(7):1013-1035.
- [5] ADALA A, TABBANE N. Discovery of semantic Web services with an enhanced-chord-based P2P network[J]. *International Journal of Communication Systems*, 2010, 23(11):1353-1365.
- [6] VAL E D, VASIRANI M, REBOLLO M, *et al.* Enhancing decentralized service discovery through structural self-organization[A]. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*[C]. Valencia, Spain, 2012. 1429-1430.
- [7] 潘静, 徐锋, 吕建. 面向可信服务选取的基于声誉的推荐者发现方法[J]. *软件学报*, 2010, 21(2):388-400.
PAN J, XU F, LV J. Reputation-based recommender discovery approach for service selection[J]. *Journal of Software*, 2010, 21(2):388-400.
- [8] 谢晓芹, 宋超臣, 张志强. 一种基于推荐网络和蚁群算法的服务发现方法[J]. *计算机学报*, 2010, 33(11):2094-2103.
- XIE X Q, SONG C C, ZHANG Z Q. A service discovery method based on referral network and ant-colony algorithm[J]. *Chinese Journal of Computers*, 2010, 33(11):2094-2103.
- [9] 郑啸, 罗军舟, 宋爱波. 基于 Agent 和蚁群算法的分布式服务发现[J]. *软件学报*, 2010, 21(8):1795-1809.
ZHENG X, LUO J Z, SONG A B. Distributed service discovery based on agent and ant colony algorithm[J]. *Journal of Software*, 2010, 21(8):1795-1809.
- [10] FELIX G M, GREGORIO M P, ANTONIO F G S. TACS, a trust model for P2P networks[J]. *Wireless Personal Communications*, 2009, 51:153-164.
- [11] JOSANG A, ISMAIL R, BOYD C. A survey of trust and reputation systems for online service provision[J]. *Decision Support Systems*, 2007, 43(2):618-644.
- [12] COLORNI A, DORIGO M, MANIEZZO V. Distributed optimization by ant colonies[A]. *Proceedings of ECAL91-European Conference on Artificial Life*[C]. Paris, France, 1991.134-142.
- [13] Stanford University. Stanford P2P sociology project[EB/OL]. <http://p2p.stanford.edu/www/publications.htm#simulator>.

作者简介:



盛国军(1974-),男,辽宁大连人,东北大学博士生,主要研究方向为服务计算。



温涛(1962-),男,陕西宝鸡人,博士,东北大学教授、博士生导师,主要研究方向为网络安全、服务计算。

郭权(1973-),男,辽宁大连人,博士,大连东软信息学院教授,主要研究方向为服务计算。

宋晓莹(1984-),女,辽宁朝阳人,东北大学博士生,主要研究方向为无线传感器网络。