

# AgletBlackboard: Aglets 系统协同机制的扩充<sup>\*</sup>

周嵩 周宇 陶先平 吕建

(南京大学软件新技术国家重点实验室 计算机科学与技术系 南京210093)

**摘要** 随着 Internet 的不断发展,移动 Agent 技术的应用越来越广泛。实际的应用通常需要由多个移动 Agent 共同完成任务,如何组织与协调移动 Agent 之间的行动带来了分布式环境中移动 Agent 之间协同的研究。本文介绍了移动 Agent 的四种协同模式,并针对 Aglets 系统中 Aglet 之间通过直接协同存在的不足,设计实现了基于黑板协同模式的 AgletBlackboard,进一步提高了 Aglets 系统的功能。

**关键词** 协同,黑板,Aglets

## AgletBlackboard: an Extension of the Coordination Mechanism in Aglets System

ZHOU Song ZHOU Yu TAO Xian-Ping LU Jian

(State Key Lab. for Novel Software Technology, Department of Science and Technology, Nanjing University, Nanjing 210093)

**Abstract** With the development of Internet, the mobile agent technology is used widely. An application usually needs several mobile agents do a task together. How to coordinate these mobile agents's activities is more and more important in such a distributing environment. In this paper, we introduce four kinds of the mobile agent's coordination model and facing the shortcoming of the direct coordination model used in Aglets system, we design and implement a stationary aglet named AgletBlackboard based on blackboard coordination model to improve the ability of the Aglets system to coordinate the aglets executing in the Aglets system.

**Keywords** Coordination, Blackboard, Aglets

### 1 前言

分布式网络尤其是 Internet 的快速发展要求新的技术的出现,移动 Agent 技术作为一种能够很好适应分布式环境应用的计算模式被越来越多的网络应用所使用<sup>[1,2]</sup>。Internet 所具有的异构性和不可预测性使得在其中的单一实体很难单独来完成相应的任务,需要通过和其他实体或环境的协同来共同完成任务。因此,对于身处异构分布式环境中的移动 Agent 而言,协同是必不可少的。

移动 Agent 在其生命周期中,需要和其他实体共同协作完成任务<sup>[3]</sup>。交互的对象可能是其他的移动 Agent,或是其运行环境中的相应资源与服务。通常,一个具体的应用往往需要由多个移动 Agent 共同来完成。这样,移动 Agent 之间就需要进行必需的通讯与协同来进行资源与信息的传递与共享,使得移动 Agent 之间可以协作地进行工作。再者,移动 Agent 在完成过程中,在网络中进行移动的同时,必然需要对不同网络节点上的资源与服务进行访问与调用,这就必然需要移动 Agent 与它们进行交互与协同。

移动 Agent 在其生命周期中不可避免地要与其他实体进行交互与协同,采用何种方式进行协同,如何协同更能适应移动 Agent 在不同应用中的需要便成了需要进一步讨论的问题。

本文的第2节分析了现有的移动 Agent 协同模式,并针对原有 Aglets 系统中对于 Aglet 之间的协同只有直接协同模式所存在的不足,提出了将黑板模式引入该系统;一个基于黑板模式的 AgletBlackboard 的设计与实现在第3节中进行了详细的阐述,它是作为一个静止的 Aglet 来实现的,是担当了

Aglet 之间协同的一个中介的角色,它实现了 Aglets 系统中交互的异步,并提供了更为灵活的交互方法和安全控制的机制;第4节则给出了一个运用 AgletBlackboard 的示例说明;AgletBlackboard 对于原有 Aglets 系统的贡献以及我们进一步的工作在最后部分进行了小结。

### 2 问题的提出

移动 Agent 的协同模式是指协同过程中移动 Agent 之间交互的模式。Giacomo Cabri<sup>[3]</sup>从交互的时间耦合度(交互是否同步)和空间耦合度(交互是否需要共有的知识考虑)考虑,将移动 Agent 的协同模式分为四类:

- 直接模式:时间紧耦合、空间紧耦合;代表系统 Aglets<sup>[4~6]</sup>、Agent-TCL<sup>[7]</sup>;优点:响应快且准确性高;缺点:对网络连接要求高,对移动 Agent 不可预测性的处理能力差,安全难以保障;

- 会议模式:时间紧耦合、空间松耦合;代表系统 Tele-script<sup>[8]</sup>、Mole<sup>[9]</sup>;优点:对交互行为的完全控制;缺点:3W (who、where、when)的严格限定,使得移动 Agent 自治性差;

- 黑板模式:时间松耦合、空间紧耦合;代表系统 Ambient<sup>[10]</sup>、ffMain<sup>[11]</sup>;优点:实现了交互的异步,保证了移动 Agent 的自治性,有效地进行安全控制;缺点:需要统一的消息标识和复杂的交互协议;

- Space 模式:时间松耦合、空间松耦合;代表系统 MARS<sup>[12]</sup>、PageSpaces<sup>[13]</sup>;优点:通信原语简单,有效地实现安全控制,对不确定环境的适应;缺点:匹配算法复杂,系统开销大,调用的不确定性。

Aglets 系统<sup>[4~6]</sup>是 IBM 开发的基于 Java 的移动 Agent

<sup>\*</sup> 本项目受973项目(No. 2002CB312002)、863项目(No. 2002AA1160101)、国家自然科学基金项目(No. 60273034, No. 60233010)、江苏省基础研究计划(No. BK2002203)资助。周嵩 硕士研究生,研究领域:移动 agent 技术,软件协同技术。吕建 教授,博士性导师,研究领域:分布对象技术,移动 agent 技术,软件协同技术。

系统,是迄今为止最为流行的移动 Agent 系统之一。Aglets 系统中 Aglet(即 Aglets 系统中的 Agent)之间的交互采用直接协同模式,是通过传递消息来进行的。交互的 Aglet 之间必须确切地知道对方的 proxy,且同步地进行消息的收发。由于直接协同模式本身所固有的一些缺点,使得 Aglets 系统在 Aglet 之间的交互方面存在着一些问题:

- 由于 Aglets 系统中 Aglet 之间的交互都是通过收发消息来完成的,交互的双方必须确切地知道对方,因此 Aglets 系统中的交互无法适应环境的不确定性,它要求交互的 Aglet 是唯一确定的,是必须先于交互而存在的。

- 当一个 Aglet 需要向另一个 Aglet 发送消息时,必须先确切地知道对方的 proxy,如果不能得到对方的 proxy,则无法发送消息,从而无法进行交互。Aglet 的移动使得其位置是很难预测的,因此 Aglet 的 proxy 也是很难确切知道的。Aglets 系统虽然提供了方法可以对 Aglet 的 proxy 进行查找,但前提是该 Aglet 必须首先将自己的 proxy 进行注册。这种注册先于查找的方法显然能力有限。

- 由于消息的接收是同步的,因此 Aglet 之间的交互也是同步的。交互的双方必须在同一时间建立连接进行消息的传递,实现彼此的交互。同步给交互带来了很大的局限性。

在实际应用中由于系统规模的不断扩大,移动 Agent 位置的定位、跟踪将很难实现;由于环境的不确定性导致交互的对象有时也无法事先确定,甚至在某些情况下,交互的一方可能在另一方已经消亡后才被创建。例如,在电子商务中,商家在商场中发布货物信息时,并不知道最终购买者是谁,而买家在发布求购信息时也不知道谁最后会成为自己的交易对象。在这种场景下,如果仍然需要交互的 Aglet 之间事先确切地知道对方的 proxy 是不现实的,Aglet 系统现有的直接协同方式显然无法满足这种交互的需求。为此,我们考虑将黑板协同模式引入 Aglets 系统。

黑板协同模式最早是在 AI 领域中由 Erman<sup>[14]</sup> 在 Hearsay-II 系统中提出的。黑板主要由三部分组成:接口、黑板和控制部分。外部实体通过接口对黑板进行访问;黑板是一个共享数据空间,用来存储所有的交互信息;控制部分则对访问请求进行统一处理并负责安全控制。黑板为交互实体提供了一套复杂的交互协议来进行有效的协同控制;实现了对协同时间上的解耦合,使得交互可以异步的进行;对交互信息以及交互行为的统一管理保证了系统和交互实体的安全;但交互实体需要对消息标识预先进行统一的规定。与直接模式和会议模式相比较,黑板模式在时间上的解耦合,支持异步交互的优势是无法比拟的;而 space 模式虽然实现了对时间和空间的双重解耦合,因而更具灵活性,但随之带来了巨大的系统开销和复杂的匹配算法,这在一些情况下是不值得的。

针对 Aglets 系统的不足以及黑板模式的特点,我们选择引入黑板模式来提高 Aglets 系统处理上述场景时的能力:

第一,黑板模式可以很好地解决交互 Aglet 之间时间上的紧耦合,实现交互的异步。黑板作为中介,负责存储与管理 Aglet 之间交互所需传递的信息。Aglet 之间原来通过彼此之间收发消息来完成的交互,变成了分别对黑板上的信息进行读写的操作。原来 Aglet—Aglet 的消息传递途径变成了 Aglet—黑板—Aglet 的形式。经过黑板这个中转站,使得本来 Aglet 之间通过建立一次直接连接完成的消息传递被分解成可以异步完成的两部分,即消息发送方向黑板上写入信息和消息接收方从黑板中读取信息。接收方的 Aglet 可以在发送方的 Aglet 在黑板上写入信息后的某个时刻对信息进行读取,从而实现了交互的异步。

第二,Aglet 系统提供了一套完整的命名和通信机制,命名机制保证了 Aglet 的命名唯一性,通信机制提供了丰富的通信方法。Aglet 系统的命名机制正好和黑板模式下要求的统一消息标识相吻合,且系统的通信机制中的消息处理机制也为黑板的交互协议处理提供了现成的方法,黑板可以很方便地对此加以利用。

第三,Aglets 系统借助事件机制中的触发回调功能来进行安全控制。相比较,黑板模式通过对交互消息的集中管理,能够实现对系统交互的统一控制,可以更为有效地进行安全保障。

### 3 AgletBlackboard 的设计

AgletBlackboard 是我们设计的一个黑板模式用于 Aglets 系统的实现机制。考虑到黑板是作为 Aglet 之间交互的中介存在的,为了保持系统的一致性,我们将 AgletBlackboard 设计成一个静止的 Aglet。用户 Aglet 对它的访问同样是通过收发消息来完成。

由于 AgletBlackboard 是静止的,因此它的 proxy 是很容易就能获得的。AgletBlackboard 在创建之后,就会在其所在的 context 中发布它的 proxy 等相关信息,用户 Aglet 可以很方便地从 context 中得到它的 proxy,并对其进行访问和操作。

通过 AgletBlackboard,用户 Aglet 在进行交互时,并不直接和交互对方进行消息传递,而是通过对 AgletBlackboard 的访问进行间接交互。因此用户 Aglet 不再需要确切地知道交互对方 Aglet 的 proxy,只需将交互信息作为发送给 AgletBlackboard 的消息参数,遵循相应的协议来完成交互。它不必考虑接收消息的 Aglet 现在在哪里、什么时候来取走消息。而接收消息的 Aglet 可以在之后的某个时刻从 AgletBlackboard 中读取信息以完成交互,它同样不需要知道这时候信息发送方的情况,即使发送信息的 Aglet 可能已经消亡。通过将直接的交互变成间接的交互,AgletBlackboard 实现了 Aglet 之间交互的异步,并使得交互双方不再需要显式地知道对方的 proxy,而只要知道 AgletBlackboard 的 proxy 就可以了,而作为一个静止的 Aglet,AgletBlackboard 的 proxy 显然是很容易获得的。

AgletBlackboard 作为交互的中介,其实是一个交互信息的中转站,负责对用户 Aglet 之间交互的信息进行存储与管理。它需要提供用户 Aglet 一套完善的信息调用机制供用户 Aglet 对 AgletBlackboard 中的信息进行访问与操作。

#### 3.1 需求分析

AgletBlackboard 作为 Aglet 之间交互的中介,以下几部分功能是必须提供的:

- 接口:AgletBlackboard 作为用户 Aglet 之间交互的中介,通过对交互 Aglet 之间的信息的存储和管理来进行协同,因此 AgletBlackboard 必须为用户 Aglet 提供一套完善的接口,便于用户 Aglet 对 AgletBlackboard 进行访问,对交互信息进行操作以实现交互。

- 交互信息存储机制:用户 Aglet 之间的交互信息是作为消息参数在 Aglet 和 AgletBlackboard 之间传递的。AgletBlackboard 作为交互中介就是通过对这些交互信息的存储和管理来实现的。需要交互的 Aglet 将交互信息作为消息参数发送给 AgletBlackboard 后,AgletBlackboard 负责保存和管理这些交互信息,在其他 Aglet 需要这些信息时提供给它们。因此必须有一个管理这些交互消息的存储机制。

- 消息请求处理机制:AgletBlackboard 是作为一个静止

的 Aglet 存在的,用户 Aglet 对 AgletBlackboard 的访问请求是通过收发消息来完成的。AgletBlackboard 是通过用户对用户 Aglet 的消息处理来提供服务,从而实现交互中介功能的。因此,需要一个对用户 Aglet 的访问请求进行处理的请求处理机制。

· 安全认证机制:为了保证系统、Aglet 和交互信息的安全,使得系统能够正常运行,安全控制机制是必不可少的。基于黑板模式的 AgletBlackboard 通过对交互信息的统一管理,可以很好地对安全进行全面的控制。

### 3.2 AgletBlackboard 的结构与功能

AgletBlackboard 由如下几部分组成(见图1)。

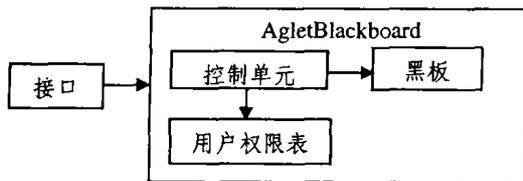


图1 AgletBlackboard 结构图

· 接口:AgletBlackboard 共提供了九条请求消息原语供用户 Aglet 对其进行访问时使用,并规定消息的类型大写。其中前三条原语是用于对安全的控制,用户 Aglet 在进行访问前需要进行身份认证并获得相应的安全权限;AgletBlackboard 在用户 Aglet 的合法身份被确认之后才允许其对信息进行访问。后六条则是满足用户 Aglet 对 AgletBlackboard 中的信息进行访问时的不同需求设计的:

- REGISTER(id,password); //注册;
- LOGIN(id,password); //登录;
- LOGOUT(id); //注销本次登录;
- POST(String subject, String poster, Access-List, String content); //发布信息,并设定对该信息进行操作所需的安全权限;
- GET(String subject); //读取一条满足条件的信息;
- DGET(String subject); //读取并擦去一条满足条件的信息;
- GETALL(String subject); //读取所有满足条件的信息;
- DGETALL(String subject); //读取并擦去所有满足条件的信息;
- MODIFY(String subject,String newContent); //对发布的信息进行修改;
- 黑板:存放用户 Aglet 需要交互的信息,Aglet 通过对信息的读写完成交互。信息存储的格式为:(String subject, String posterid, AccessList, String content);
- subject:信息的名,用于进行信息匹配。为了适应应用的复杂性,subject 是可以分段匹配的,形如 subject1:subject2:subject3,每个分段之间用冒号(:)分隔开,同时支持通配符(\*)的使用。分段的 subject 给具体应用中复杂的协议提供了支持;
- posterid:记录了发布信息的 Aglet,为信用值中的 self 权限提供支持;
- AccessList:它记录了对该信息进行操作所需的权限,其格式形如(GET:所需权限;DGET:所需权限;MODIFY:所需权限);

· content:信息的内容,以字符串的形式表示;

· 用户权限表:AgletBlackboard 为了保证信息的安全,防止怀有恶意的 Aglet 进行破坏,需要用户 Aglet 通过身份认证才可以对其进行访问。用户 Aglet 首先需要进行注册,向 AgletBlackboard 提交自己的用户名和密码,注册成功后,就可以使用该用户名登录 AgletBlackboard 进行访问。用户名作为访问 AgletBlackboard 的唯一标识需要妥善保存,AgletBlackboard 还为每个用户名设定了安全权限,可以有效地管理和控制该用户对信息的操作权限。AgletBlackboard 使用一张用户权限表存放用户的注册信息与权限信息,进行统一的安全控制和管理。用户权限表的数据存储格式为(String id, String password, Int credit);

· id:用户 Aglet 访问 AgletBlackboard 所使用的用户名,需要用户 Aglet 首先进行注册,之后通过登录,作为用户 Aglet 在 AgletBlackboard 中的身份标识。

· password:用户名密码,在用户登录时进行身份验证,AgletBlackboard 提供用户 Aglet 对密码的修改。

· credit:AgletBlackboard 赋予每个用户名一个信用值,每条信息的相应操作均需要一定的信用值等级,用户只有达到了所要求的信用值等级才能对信息进行需要的操作。通过信用值实现了对用户访问信息的权限控制,信用值的高低体现了该用户在 AgletBlackboard 中访问权限的高低。AgletBlackboard 还提供了对用户名的信用值进行管理的方法,可以对信用值进行动态的设定与调整。

· 控制单元:这是 AgletBlackboard 的核心组成部分。它主要负责以下几部分功能的实现和控制。

· 初始化:AgletBlackboard 在创建之后,控制单元负责将 proxy 在所在的 context 中进行发布,供用户 Aglet 查询;

· 消息处理:用户 Aglet 对 AgletBlackboard 的访问是通过收发消息来完成的,控制单元负责将用户 Aglet 所发送的消息进行消息处理,这是 AgletBlackboard 最重要的部分。控制单元通过对消息的分析,获得用户 Aglet 的具体访问请求,经过相应的处理后将结果消息发送回用户 Aglet;

· 安全控制:控制单元负责对用户 Aglet 的注册、登录进行管理和身份认证,并实现对用户 Aglet 访问信息的请求进行操作权限的控制和管理,确保信息和 AgletBlackboard 的安全;

· 黑板管理:黑板的管理,对信息的查找、添加、删减以及修改等工作由控制单元负责完成。对于用户 Aglet 对信息的操作请求,控制单元经安全认证之后,帮助用户定位所需的信息,并完成对信息请求的操作。

### 3.3 AgletBlackboard 的实现

AgletBlackboard 作为一个静止的 Aglet,它继承了 Aglet 类。它作为 Aglet 之间交互的中介,对交互信息进行统一的存储和管理,实现了 Aglet 之间的异步交互,并提供强大的安全控制机制。用户 Aglet 对 AgletBlackboard 的访问仍然是通过 Aglet 之间的消息传递来实现的。

AgletBlackboard 主要有 ItemForm、UserInfo 和 AgletBlackboard 三个类:

ItemForm 类是 AgletBlackboard 中用于存储和管理用户 Aglet 之间交互信息的。它提供了对存储信息的查找、添加、删除和修改等操作。

UserInfo 类完成对用户权限表的管理。为了保证 AgletBlackboard 和其中信息的安全,用户 Aglet 在对 AgletBlack-

board 的访问前需要身份认证,对于信息的操作同样需要权限的检查。UserInfo 类提供了对用户信息与权限的查找、增减和修改等管理。

AgletBlackboard 类是 AgletBlackboard 的核心部分。它完成 AgletBlackboard 的初始化工作,对信息的管理,对用户 Aglet 的消息进行处理并实现对 AgletBlackboard 的安全控制。AgletBlackboard 类最关键的部分就是其消息处理单元,它是对 Aglet 类中 handleMessage 方法的复用。AgletBlackboard 提供了九条接口原语供用户 Aglet 使用,handleMessage 方法通过对消息类型的匹配,执行相应的操作。

#### 4 应用示例

现以电子商务为例,给出一个简单的应用示例:商家将出售商品的信息在 AgletBlackboard 上发布,买家通过查询获得相应的信息;同样,买家的求购信息也可以发布供商家查询。在这个示例中,规定 subject 的格式为(信息类型,出售 or 求购:商品名称:信息发布者),对信息的操作权限按信用值高低

控制。

```
class AgletBlackboard extends Aglet {
    //在 context 中发布 AgletBlackboard 的 proxy.
    getAgletContext().setProperty("AgletBlackboard", getProxy());
    //AgletBlackboard 的核心部分,处理用户请求.
    public boolean handleMessage(Message msg){};
}
class AgletSeller extends Aglet {
    //获取 AgletBlackboard 的 proxy.
    blackboardProxy = (AgletProxy) (getAgletContext().getProxy("AgletBlackboard"));
    //登录 AgletBlackboard.
    reply = (String) blackboardProxy.sendMessage("REGISTER";:id,password);
    //向 AgletBlackboard 发布商品信息.
    userAglet.blackboardProxy.sendMessage("POST";:subject,posterid,accessList,content);
}
class AgletBuyer extends Aglet {
    blackboardProxy = (AgletProxy) (getAgletContext().getProxy("AgletBlackboard"));
    reply = (String) blackboardProxy.sendMessage("REGISTER";:id,password);
    //从 AgletBlackboard 中读取商品信息.
    Vector replis = (Vector) userAglet.blackboardProxy.sendMessage("GETALL";:subject);
}
```

系统运行结果如图2所示。

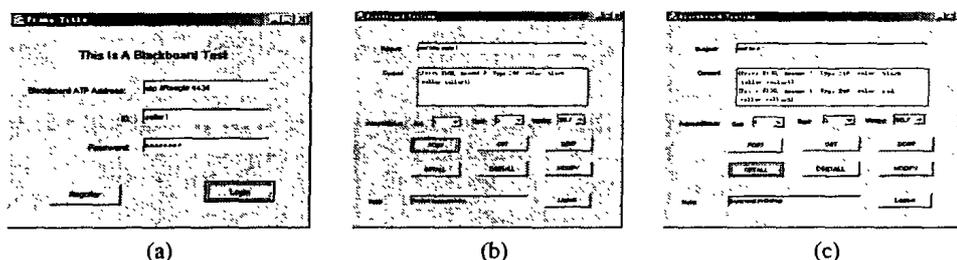


图2

图 a 为用户 Aglet 登录已知节点上 AgletBlackboard 的界面,需要输入用户 id 和 password。

图 b 为用户 Aglet 发布商品信息界面:Subject 一栏中的 sell:bike:seller1 表示这是一条出售商品的信息,出售商品是 bike,出售者是 seller1;Content 中描述了该条出售信息的详细资料,包括商品的价格、数量、款式、颜色和出售者等信息;AccessSetup 设定了对该信息的访问权限值,该信息允许信用值高于1的用户读,允许信用值高于3的用户读并擦去信息,只允许自己对信息进行修改。

图 c 为用户 Aglet 查询商品信息界面:Subject 一栏中 sell:bike:\* 表示用户需要查询出售 bike 的信息,对于出售者没有特殊限定.getall 命令共获得两条符合要求的商品出售信息,分别为 seller1 和 seller3 提供的。

**小结** Aglets 系统原有的通信和交互机制对于异步协同存在着不足,Aglet 由于自身的移动导致 proxy 的难以获得性同样制约着原有系统的交互。AgletBlackboard 的开发作为对 Aglets 系统交互模式的一个有益的补充,使得 Aglet 可以在不知道对方 proxy 的情况下依然可以完成相应的交互。由于 AgletBlackboard 位置是静止的,因此它的 proxy 很容易获得,从而使得交互变得方便可行。AgletBlackboard 作为一个 Aglet 之间交互的中介,实现了 Aglet 间交互的异步。Aglet 之间的直接交互被分割为交互双方分别对 AgletBlackboard 的访问。交互的信息被存储在 AgletBlackboard 中,交互双方对信息的写与读可以在不同的时间进行,原来的一次性交互被划分为交互双方与 AgletBlackboard 的两次独立的交互行为。AgletBlackboard 对交互信息的统一管理,使得在安全控制方面为 Aglet 和系统的正常运行提供了可靠的保证。用户 Aglet 对 AgletBlackboard 的访问以及对信息的操作需要身份认证

和安全权限控制,这使 Aglet 和交互信息的安全都得到了提高。

本文中 AgletBlackboard 只是作为本地的一个信息中介存在。对于存在于不同节点上的多个 AgletBlackboard 之间的协同还没有考虑,即如何创建一个分布式的 AgletBlackboard。另外,对于采用何种垃圾收集机制也是我们下一步将继续研究的问题。

#### 参考文献

- 1 Proc. of the 2nd Intl. Workshop on Mobile Agents. Stuttgart (D), Lecture Notes in Computer Science, No. 1477, Springer-Verlag (D), Sep. 1998
- 2 Mobile Object Systems. Lecture Notes in Computer Science, No. 1222, Springer-Verlag (D), Feb. 1997
- 3 Cabri G, Leonardi L, Zambonelli F. How to Coordinate Internet Applications based on Mobile Agents. IEEE Seventh Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), IEEE CS Press, Stanford (USA), June 1998
- 4 Lange D B, Oshima M. Programming and Deploying Java (tm) Mobile Agents with Aglets (tm). Addison-Wesley, Reading (MA), Aug. 1998
- 5 Oshima M, Karjoth G. Aglets Specification (1.0). <http://www.mypcera.com/softxue/www.trl.ibm.co.jp/aglets/spec-version10.html>, May. 1997
- 6 Lange D B, Chang D T. IBM Aglets Workbench White Paper. <http://www.mypcera.com/softxue/aglets.trl.ibm.co.jp/documentation.html>, Sep. 1996
- 7 Gray R. Agent-Tcl: A flexible and secure mobile-agent system. In: Mark Diekhans Mark Roseman ed. Proc. of the fourth Annual Tcl/Tk Workshop (TCL '96), Monterey, California, July 1996

(下转第171页)

表2 聚类重组结果

$\alpha \backslash w$	1.0	1.2	1.4	1.6	1.8	2.0
1.0	20	10	6	6	6	5
1.2	18	10	6	6	6	5
1.4	15	10	6	6	6	5
1.6	10	10	6	6	6	5
1.8	10	10	6	6	6	5
2.0	9	8	6	6	6	5

(a) 数据集 1

$\alpha \backslash w$	1.0	1.2	1.4	1.6	1.8	2.0
1.0	35	10	5	5	4	1
1.2	28	9	5	5	4	1
1.4	20	9	5	5	4	1
1.6	12	8	5	5	4	1
1.8	8	8	5	5	4	1
2.0	7	7	5	5	4	1

(b) 数据集 2

$\alpha \backslash w$	1.0	1.2	1.4	1.6	1.8	2.0
1.0	5	3	3	3	3	1
1.2	5	3	3	3	3	1
1.4	5	3	3	3	3	1
1.6	5	3	3	3	3	1
1.8	4	3	3	3	3	1
2.0	3	3	3	3	3	1

(c) 数据集 3

$\alpha \backslash w$	1.0	1.2	1.4	1.6	1.8	2.0
1.0	2	2	2	2	2	2
1.2	2	2	2	2	2	2
1.4	2	2	2	2	2	2
1.6	2	2	2	2	2	2
1.8	2	2	2	2	2	2
2.0	2	2	2	2	2	2

(d) 数据集 4

表3 禁忌搜索优化结果

$\alpha \backslash w$	1.0	1.2	1.4	1.6	1.8	2.0
1.0	6	6	6	6	6	4
1.2	6	6	6	6	6	4
1.4	6	6	6	6	6	4
1.6	6	6	6	6	6	4
1.8	6	6	6	6	6	4
2.0	6	6	6	6	6	4

(a) 数据集 1

$\alpha \backslash w$	1.0	1.2	1.4	1.6	1.8	2.0
1.0	35	8	4	4	4	1
1.2	28	8	4	4	4	1
1.4	20	8	4	4	4	1
1.6	12	8	4	4	4	1
1.8	8	8	4	4	4	1
2.0	7	7	4	4	4	1

(b) 数据集 2

$\alpha \backslash w$	1.0	1.2	1.4	1.6	1.8	2.0
1.0	3	3	3	3	3	1
1.2	3	3	3	3	3	1
1.4	3	3	3	3	3	1
1.6	3	3	3	3	3	1
1.8	3	3	3	3	3	1
2.0	3	3	3	3	3	1

(c) 数据集 3

$\alpha \backslash w$	1.0	1.2	1.4	1.6	1.8	2.0
1.0	2	2	2	2	2	2
1.2	2	2	2	2	2	2
1.4	2	2	2	2	2	2
1.6	2	2	2	2	2	2
1.8	2	2	2	2	2	2
2.0	2	2	2	2	2	2

(d) 数据集 4

表3所示为禁忌优化阶段得到的聚类簇数目估算结果,可见禁忌优化对估算结果有了较大改善,除了数据集4保持正确估算结果外,数据集1和3的估算结果基本满足  $w$  和  $\alpha$  大范围变化的要求,同时,对于数据集2的复杂分布特征,TABU-

Clustering 算法有了进一步改善,正确估算区间  $\alpha$  扩展到[1.4, 1.8]的范围内。

综上所述,TABU-Clustering 算法分阶段优化计算结果,最终得到正确的聚类簇数目的估算结果。由于数据集2和4,特别是数据集2的无规则分布,聚类中心不能代表聚类簇特征,因此对于文[6~8]中提出的以聚类簇中心代表聚类簇进行运算是不可行的。通过仿真实验表明,TABU-Clustering 算法对紧凑球型分布和非紧凑球型分布的对象集合均具有较好的适应能力,能够估算出正确的聚类数目。

**结论** 聚类分析面临的主要挑战之一是如何准确确定模式的数目,由于在多数情况下,算法设计人员不能预知聚类簇数目,导致算法陷入局部最优解或者实验人员必须逐一实验来确定聚类簇数目。本文提出基于禁忌搜索的聚类簇数目估算算法—Tabu-Clustering,算法能同时处理具备紧凑球型分布或非紧凑球型对象分布的对象集合。算法通过三个阶段逐步优化结果,最终估算出合适的聚类簇数目,实验表明 Tabu-Clustering 算法用于聚类簇数目估算是可行的。

参考文献

- Jain A K, Dubes R. Algorithms for clustering data. New Jersey: Prentice-Hall, 1988
- Selim S Z, Ismail M A. K-means-type algorithm; generalized convergence theorem and characterization of local optimality. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1984, 6 (1): 81~87
- Bandyopadhyay S, Maulik U, Pakhira M K. Clustering Using Simulated Annealing With Probabilistic Redistribution. International Journal of Pattern Recognition and Artificial Intelligence, 2001, 15 (2): 269~285
- Koontz W L, Narendra P M, Fukunaga K. A branch and bound clustering algorithm. IEEE Transaction on Computers, 1975, 24 (9): 908~915
- Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique. Pattern Recognition, 2000, 33 (9): 1455~1465
- Boudraa A O. Dynamic estimation of number of clusters in data sets. Electronics Letters, 1999, 35 (19): 1606~1608
- Bandyopadhyay S, Maulik U. Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recognition, 2002, 35 (6): 1197~1208
- Kothari R, Pitts D. On finding the number of clusters. Pattern Recognition Letters, 1999, 20 (4): 405~416
- Tseng L Y, Yang S B. A genetic clustering algorithm for data with non-spherical-shape clusters. Pattern Recognition, 2000, 33 (7): 1251~1259
- Tseng L Y, Yang S B. A genetic approach to the automatic clustering problem. Pattern Recognition, 2001, 34 (2): 415~424
- Glover F, Laguna M. Tabu Search. Boston: Kluwer Academic Publishers, 1997

(上接第141页)

- White J. Mobile Agents. In: J. Bradshaw ed.: Software Agents, AAAI Press, Menlo Park (CA), 1997. 437~472
- Baumann J, Hohl F, Straßer M, Rothermel K. Mole-Concepts of a Mobile Agent System. WWW Journal, Special issue on Applications and Techniques of Web Agents, 1998
- Cardelli L, Gordon D. Mobile Ambients. Foundations of Software Science and Computational Structures. Lecture Notes in Computer Science, No. 1378, Springer-Verlag (D), 1998. 140~155
- Domel P, Lingnau A, Drobniak O. Mobile Agent Interaction in Heterogeneous Environment. 1st International Workshop on Mobile Agents, Lecture Notes in Computer Science, Springer-Verlag

- (D), 1997, 1219: 136~148
- Cabri G, Leonardi L, Zambonelli F. MARS: A Programmable Coordination Architecture for Mobile Agents. IEEE Internet Computing, 2000, 4(4): 26~35
- Ciancarini P, Tolksdorf R, Vitali F, Rossi D, Knoche A. Coordinating Multi-Agents Applications on the WWW: a Reference Architecture. IEEE Trans. on Software Engineering, 1998, 24(8): 362~375
- Erman L D, Hayes-Roth E, Lesser V R, Reddy D R. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. Computing Surveys, 1980, 12(2): 213~253