# Optimization and Distributed Execution of MAS Cooperative Strategy Based on MDP in Wireless Sensor Networks *

*WANG Xiao-ling, MU De-jun*, LIU Zhe-yuan*

*(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)*

**Abstract:** In order to reduce the complexity created by MDP model and the cooperation traffic, the method of creating strategy tree by the model is improved. Using the context-specific and conditional independence existing among the agent states in wireless sensor networks, the tree created by SPI algorithm is decomposed and optimized. This makes the independent agents in MAS running independently, and only communicating with each other when cooperation is needed. During operation Q-decomposition approach is proposed for resource allocating. Simulation experiment was developed by STARLOGO. Simulation indicates that MAS applying the strategy not only accomplishes the task and gains the reward, but effectively reduces traffic simultaneously.

**Key words:** MAS; MDP; wireless sensor network; context-specific independence; conditional independence; Q-decomposition

**EEACC:** 6150P

# 无线传感器网络中基于 MDP 的 MAS 协作策略的优化及分布执行 *

王晓伶,慕德俊*,刘哲元

(西北工业大学自动化学院,西安 710072)

**摘 要**:为降低马尔可夫决策模型生成 MAS 协作策略的复杂度,减少协作通信量,在无线传感器网络中利用 agent 状态之间存在的条件独立性与上下文独立性关系提出了一种新的优化方法。方法通过分解并优化 SPI 算法生成的策略树,使得 MAS 中处于独立状态的 agent 可以分布独立运行,只有在需要同其他 agent 协商时才进行通信。并在协作中采用 Q 分解机制实现共享资源的分配,减少资源使用冲突,获取更大奖励。使用 STATLOGO 软件对方法进行验证,实验结果表明该方法在 MAS 完成协作任务获取目标奖励的同时,具有产生通信量较小的优点。

**关键词**:多智能体系统;马尔可夫决策过程;无线传感器网络;上下文独立;条件独立;Q 分解

**中图分类号**:TP393.15 **文献标识码**:A **文章编号**:1004-1699(2009)04-0520-06

近年来利用 agent 技术构建无线传感器网络已成为的一个重要的研究方向[1-2,10-11]。agent 如何在无线传感器网络中进行协作是研究的一个难点,这是因为 agent 在协作过程中不仅要观察自身的状态,同时也要推断合作者的状态及动作。针对协作环境的不确定性,马尔可夫决策过程(MDP,Markov Decision Process)广泛应用于 MAS(Multi-Agent System)建模研究中。但是 MDP 生成协作策略比较复杂,即使只有两个 agent 进行协作,其时间复杂度为 NEXP-complete[3]。一种降低 MDP 计算复杂度的方法是将 MDP 观察域分类,其中一类 MDP 称为传输独立 MDP,它的系统状态可以被分割为独立 agent 的状态,这些 agent 状态仅和自身的活动相关,独立于系统联合活动,传输独立 MDP 模型生成策略的复杂度为 NP-complete[4]。虽然这种方法大大降低了复杂度,但并不是所有的 MAS

**基金项目**:国防基础科研项目资助(C2720061361)
**收稿日期**:2008-11-12 **修改日期**:2009-02-18

系统观察域是传输独立的。另外一种方法是使用可分解 MDP 模型,这种方法充分利用系统状态变量中存在的条件独立关系、上下文独立关系简化生成协作策略的复杂度,复杂度可以减少为 P-complete[6]。但是这种方法在执行过程中不考虑通信的代价,即执行过程中协商双方需要随时进行协商通信。文献[5]提出了一种优化 MDP 模型生成策略的方法,该方法分解、优化生成的策略树,使得 MAS 可以分布执行策略树,在一定程度上减少了通信的花销。但是在生成 agent 策略时,这种方法没有优化相应的策略子树,策略树在执行过程中会重复访问部分状态节点,从而产生冗余通信。

为减少冗余的通信量,进一步优化协作策略,本文在无线传感器网络中利用 d-分离、上下文分离对分解后的策略子树进行优化,进一步提取子树中存在的条件独立性和上下文独立性关系,去除子树中冗余的节点状态特征,从而减少协作过程的通信量。

在 MAS 协作过程中状态集是 agent 各种可能的动作状态和可用资源的合集,每个可能动作包含对该 agent 资源的分配。当 agent 之间竞争共享资源时,一个 agent 的动作可能会影响到其它 agent 完成协作任务时所获取的报酬。本文采用 Q 分解机制[7-8]解决共享资源在协作 agent 之间的分配,以保证 MAS 任务收益的最大化。实验表明经过优化后的策略在获取目标奖励同时可以分布执行并且能有效地减少通信代价。

## 1　MDP 模型描述

在 MAS 中任务协作、资源分配策略是一组随机过程,马尔可夫决策过程用来处理协作 agent 的联合动作、联合观察结果、协作过程中资源的分配以及最终生成任务协作策略。

本文 MDP 模型结构表示为

$$< \mathrm{Res}, \mathrm{Ta}, S, X, A, P, R >$$

其中:* Res = {$\mathrm{res}_1, \cdots, \mathrm{res}_n$}表示可用资源的有限集合,每类资源在分配时有一个局部限制 $L_{\mathrm{res}}$ 及全局总体限制 $G_{\mathrm{res}}$,即一类资源每次分配不超过 $L_{\mathrm{res}}$ 而总数不超过 $G_{\mathrm{res}}$;

* Ta 表示需要完成的任务集合;

* $S$ 表示状态的有限集合;

* $X = \{X_1, \cdots, X_n\}$,表示 agent 状态的向量空间,状态特征 $X_i$ 对应于状态集合 $S$ 中状态 $S_i$,特征可以是布尔型或者其他数据类型;

* $A$ 表示可能的联合动作集合;

* $P$ 定义状态变迁概率函数,表示当给定一个动作时由一个状态变迁到另一个状态的可能性。且对于 $\forall z \in Z, Z \subset X$,变迁概率函数为 $z$ 中各状态属性变迁概率函数的连乘积

$$P(z \mid X_i, a) = \prod_{i=1}^{|Z|} P(Z_i \mid X_i, a);$$

* $R$　$X \times A \rightarrow R$,表示奖励函数;

* $\lambda$ 表示折扣因子,其值介于 0 和 1 之间。

在无线传感器网络中,MDP 模型的状态特征和动作之间的关系可以用 Bayesian 网进行描述:$t+1$ 时刻模型的状态特征依赖于其父状态 $t$ 时刻的特征值,条件独立于其他变量特征值,Bayesian 网中变量之间的这种条件依赖关系用条件概率表表示。但是在多数情况下,变量只是依赖一些特定的父变量特征值,而不是全部的父特征值,称为上下文独立关系[5-6],使用条件概率树描述。如图 1 树型结构所示,状态 Y 在 X 取值为 0 时,上下文独立于 Z。

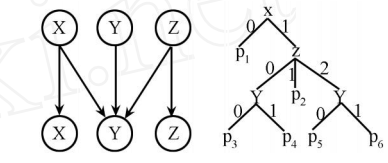


图 1　无线传感器网络中的 Bayesian 状态条件概率树

## 2　策略树的分解与执行

在 MDP 模型中 SPI(Structured Policy Iteration)算法[6]用来生成 MAS 协作策略树,策略树的叶子节点表示联合动作,非叶子节点表示 agent 状态。SPI 算法生成的策略树在 MAS 中是集中联合执行的,即使处于可独立执行状态,agent 也需等待其他 agent 状态。为使策略树在 MAS 中分布执行,单个 agent 需一个可以匹配当前状态特征的执行策略。在文献[5]的基础上本文提出一种优化方法,首先将 SPI 生成的策略树转换为使各 agent 可分布执行策略树[5];然后提取并删除冗余的状态特征,优化生成的策略子树;最后在执行过程中实现资源的优化配置。

### 2.1　联合执行策略树的分解

定义 1　动作独立,如果在策略树叶子节点动作集合中存在一个动作,该动作同其他动作联合执行,且是多个联合动作的交集,在遍历该子树节点时会被同一个 agent 执行。例如联合动作集{ax,ay,bz}中,动作{a}独立;而在动作集{ax,by,cz}中没有独立的动作。

定义 2　交叉简化操作[5],用来合并策略树分支中动作集合交集非空的叶子节点或者子树。在策略树非叶子节点中递归调用该操作,如果节点中所

有的孩子都是叶子节点,并且节点动作集合的交集非空,则该节点是冗余的,节点被一个包含交集的新叶子节点所代替;如果孩子中仅有一个是非叶子节点,该节点冗余,且如果该节点子树中包含于叶子节点动作集合的交集,则该节点被子树代替。

将 SPI 生成的策略树转化为 agent 可分布执行的策略树的方法如表 1。

表 1 策略子树分解方法

| DecomposePolicy(joint-policy) |
|---|
| Input :joint-policy / / 联合执行策略树 |
| Output :policy / / 分解后策略树 |
| 1. 对每个 agent 重组策略树,使 agent 状态处于策略树根节点; |
| 2. 对于策略树中叶子节点,找出所有独立动作; |
| 3. 使用预定义动作序列解除联合动作之间的关系; |
| 4. 将联合动作分解为个体动作; |
| 5. 使用交叉简化操作简化策略树 |

## 2.2 策略子树的优化

联合策略树经过转换后,对每个 agent 生成一个单独的策略树,在策略树中存在着大量的冗余状态节点,在执行前需要对子树进行简化及优化。

定义 3 d-分离[9]:$X$, $Y$, $Z$ 是 Bayesian 网中有向非循环图 T 中三个互不相交的节点集,$Z$ d-分离 $X$ 和 $Y$,记作 $I(X, Y|Z)_T$,当且仅当 $X$ 中任一节点和 $Y$ 中任一节点间的任一路径被节点集 $Z$ 阻塞。

定义 4 给定 Bayesian 网 T 和上下文 c,在上下文 c 条件下,删除 Bayesian 网 T 中的无意义弧得到的结果网络定义为 T(c)。在 T(c)中,如果 $X$ 和 $Y$ 被 $Z \cup C$ d-分离,则称上下文 c 及变量表 $Z$, $X$ 和 $Y$ 被上下文分离[9],记作 $I_c(X, Y|Z, c)$。

定义 5 简化操作,用来去除策略树中冗余的内部节点,在策略树中递归调用简化操作。策略树 T 的一个分支 b 中如果有两个或两个以上的内部节点使用同一个值标记(即节点状态特征值相同),则保留处于分支最顶部的节点,删除其他节点,并且对于每个被删除的节点,只保留其一个子树,该子树边的标签值同于从保留的节点遍历到该节点时的标签值(见图 2)。另外,如果一个内部节点将一个棵树分为两个或两个以上的相同子树,则将该节点删除,并将其上的父连接弧直接定向到其中的一个子树上,删除其余子树。

首先利用定义 3d-分离操作提取策略树中的条件独立性关系,然后在条件独立性关系中使用定义 4 操作提取可上下文分离节点,分离这些节点,简化策略树,最后在树中中递归调用简化操作,简化树。

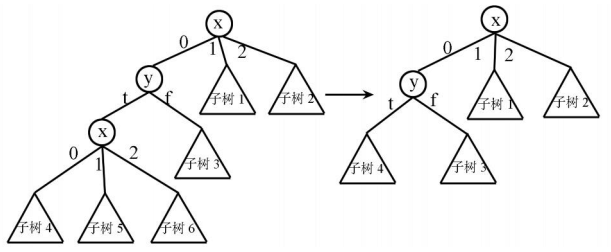


图 2 简化操作示例

最终经过优化后形成的子策略树如图 3 所示,树中根节点表示属于 agent i 的当前状态特征,非叶子状态表示协作 agent 的特征值。

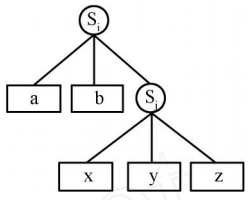


图 3 agent i 的分布执行策略树

## 2.3 资源分配

经过分解、优化后的子策略树在执行过程中需要配置各类资源。资源分配是一个随机过程,并且一个 agent 的资源分配可能会影响其它协作者的收益,为使得 MAS 协作任务获取的报酬最大化、合理分配资源,本节将 Q-分解机制[7-8]应用在资源分配中。

无线传感器网络中资源是一个有限集合,资源按使用类型分为两类,一类称为消耗性资源,该类资源经使用后数量减少;另一类称为非消耗性资源,又称临界资源,该类资源经使用后数量不变,但不能为多个 agent 同时使用。每类资源在分配时有一个局部限制 $L_{res}$ 及全局总体限制 $G_{res}$,即一类资源每次分配不能超过 $L_{res}$ 而总数不能超过 $G_{res}$。Q-分解应用于当资源已在 agent 之间共享,但是一个 agent 的动作可能影响到其它 agent 获取的报酬,并且协作任务 ta 整体报酬 $R$ 可以分解为各个 $R_i$ 之和(即 $R = \sum_{i \in Ag} R_i, i \in Ag$,表示参与协作任务的 agent)的条件下。

对于经过分解优化后的子策略 $\pi$ 而言使用资源时有

$\mathrm{res}_i(\pi(s)) \leq L_{res}(s)$,其中 $s \in S$ 表示子策略 $\pi$ 中 agent 处于状态 $s$ 时,分配资源 $i$ 所受的局部限制;

$\mathrm{res}_i(tra) \leq G_{res_i}(tra)$,其中 $tra$ 为 agent 完成目标任务的整个执行过程,表示 agent 执行中分配资源 $i$ 所受的全局限制。

$V^{\pi}(s)$ 表示从状态 $s$ 出发,并且以后遵循此策略

而获取的回报,最优值表示为

$$V^{\pi}(s) = R(s) + \max_{a \in A(s)} \gamma \sum_{i \in Ag} P(S' \mid s, a) V^{\pi}(S') \quad (1)$$

$Q(a,s)$ 表示处于状态 $s$ 下,采用动作 $a$ 取得的回报值。

$$Q(a,s) = R(s) + \gamma \sum_{a \in A(s)} P(S' \mid s, a) Q(a', S') \quad (2)$$

表 2 是利用 Q-分解进行资源分配的算法。对特定协作策略 $\pi$ 进行资源分配是一个迭代学习的过程,首先每个 agent $i$ 从各自的角度计算 $Q_i(a_i, s_i)$ 值,然后 agent $i$ 将 $Q_i(a_i, s_i)$ 发送到资源分配仲裁者,仲裁者选取一个使 $V^{\pi}(s)$ 值最大的动作 $a_i$;在下一次迭代中,状态 s 被更新,重新计算 $Q_i(a_i, s_i)$,选取下一个动作。经过一次迭代后,资源状态更新为 $Res_i / res_i(a_i)$。

表 2　资源分配算法

```
ResAllocate(s)
Input: s//状态特征
1. V^π(s) = 0;
2. 每个 agent i 使用公式 2 计算 Q_i(a_i, s_i),其中 a_i ∈ A_i(s);
3. //选择一个使得 V^π(s) 最大的动作 a_i
4. For all a ∈ A(s) do
5. {
6.   Q(a,s) = 0;
7.   For all i ∈ Ag do
8.   {
9.     Q(a,s) = Q(a,s) + Q_i(a_i, s_i);
10.  }
11.  If Q(a,s) > V^π(s)
12.  {
13.    V^π(s) = Q(a,s);//选取最大的 Q 值
14.    For all i ∈ Ag do
15.    {
16.      //策略 π 选取动作 a_i
17.      π_i(s) = a_i;
18.      Q_i*(a_i, s_i) = Q_i(a_i, s_i);
19.    }
20.  }
21. }
```

### 2.4　策略的分布执行

对于单个 agent i 而言经过分解、优化后的策略树是一个以自身状态为根节点的策略树,如图 3 所示,$S_i$、$S_j$ 分别为 agent i、j 的观察状态。agent i 遍历策略树,根据自身状态选择分支,如果选择的分支是叶子节点,则执行叶子节点中的动作 a;当遍历到与其协作的 agent j 状态节点时,agent i 需要同 agent j 进行协商通信以了解其状态,并据此选择要执行的动作。分布执行的递归算法如表 3 所示。

表 3　agent 分布执行策略的递归算法

```
Individual Execute(policy, state, tm_state)
Input: policy;//图 3 中经过简化后的策路树
       state;//agent 当前状态集合
       tm_state;//已知的协作 agent 状态值集合
Output: action;//返回一个执行动作
define: index;//表示节点的第几个子树
①如果 policy 是叶子节点,则直接返回 policy 指向的动作;
②如果 policy 遍历的当前节点是非叶子节点,且 policy.value 包含于 state,则:
  a) index = state[policy.value];
  b) 返回
     Individual Execute(policy.child_index, state, tm_state);
③如果 tm_state 为空,则:
  a) 找出所有 state 中需要同其他 agent 协商的状态特征 value;
  b) 同其他 agent 协商,并取得这些值;
  c) 创建 tm_state 值集合;
④index = tm_state.value[policy.value];
⑤返回 Individual Execute(policy.child_index, state, tm_state)。
```

## 3　实验与评估

(1)实验环境描述

本文实验是在 starlogo 中模拟无线传感器网络中节点 agent 向目标坐标 T 移动的过程,当到达目标 T 时 agent 向其他 agent 发送到达信号 Signal。

无线传感网络部署在一个 15 ×15 的矩阵空间中,各 agent 节点处于空间中的随机位置(图 4 表示了一次实验的初始环境:标注中 Ai 表示 agent i 初始位置,T 表示目标位置),各节点 agent 可以自由的在矩阵空间中活动,只有在到达目标坐标时才相互通信,之前则是各自独立的运行。

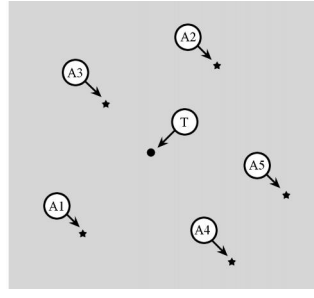


图 4　试验环境示例

单个节点 agent 动作集为{North, South, West, East, Stop, Signal};agent 动作成功的可能性 $P$ 为 90 %;当双方 agent 成功发送 Signal 得到奖励 $R$ 为 +20,反之如果双方协作失败则予以 - 20 的

惩罚;agent 每次移动消耗 1 的代价;在该实验中 agent 观察自己坐标($x$, $y$),相互之间不知道对方的位置,只有在到达指定目标时才发送 Signal,之前 agent 则在上下文独立的环境中独立运行。

在资源分配的模拟中,定义两类资源:消耗性资源($R_{c1}$, $R_{c2}$),临界资源 $R_{NC}$。实验中使用消耗性资源的设置如表 4;对于临界资源 $R_{NC}$ 的访问设置临界区,采用 FCFS 方式处理,agent 到达目标坐标 T 之后,互斥使用 $R_{NC}$ 发送信号 signal。

表 4 资源分配限制设定

| 资源 | 资源总数 | 局部限制($L_{res}$) | 全局限制($G_{res}$) |
|---|---|---|---|
| $R_{c1}$ | 16 | 3 | 8 |
| $R_{c2}$ | 10 | 2 | 5 |

(2) 实验结果分析

试验中分别采用可自由通信的联合执行策略(表 5 中简称 JOINT-POLICY)、文献[5]中采用的 Dec-MDP 生成的策略、本文策略在上述试验环境中运行 200 次,在每次运行中 Ai 初始位置分别处于不同的随机位置。比较结果见表 5 和图 5:

表 5 三种不同策略执行结果比较

| 策略项目 | 平均奖励($\sigma^2$) | 平均通信量($\sigma^2$) |
|---|---|---|
| JOINT-POLICY | 17.384(1.062) | 7.032(1.059) |
| Dec-MDP | 16.378(1.142) | 3.332(1.042) |
| 本文模型 | 17.384(1.062) | 2.553(1.053) |

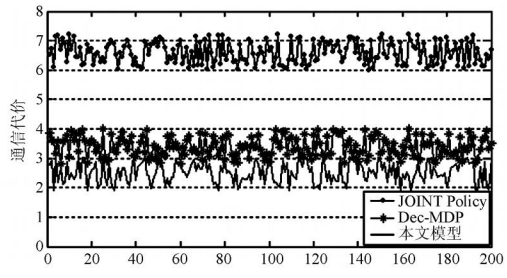


图 5 三种策略消耗的通信代价

本文策略利用 Bayesian 网路中存在独立性关系,提取策略树节点中状态特征,使得处于独立状态的 agent 可以独立运行,只有在需要协作时,才同协作 agent 进行通信交互,这样大大减少了系统整体的通信代价,比联合执行策略 JOINT-POLICY 减少了近 67 %的通信量;在分布执行前本文策略对分解后的子策略进行优化,除去了子策略树中冗余的节点(或子树),减少了对这些状态的访问通信,通信量比文献[5]模型生成的策略降低了近 22 %。

本文实验环境中 agent 是在 15 ×15 的矩阵空间中活动,agent 只有在到达目标坐标时才相互通信,之前则是各自独立运行,系统中存在大量的独立状态。这样当 agent 在更大的矩阵空间中运行时,agent 运行过程中存在的独立状态将会更多,系统也将会节省更多的通信量。

图 6 是在资源使用有限制的情况下,本文策略同文献[5]策略取得的奖励比较。在实际的应用中,资源使用一方面受数量的限制,另一方面受共享冲突的限制。文献[5]模型生成的策略是在假设资源可以任意使用的前提下 agent 进行协作。在资源的使用限制设定如表 4 时,本文策略利用 Q 分解机制解决资源共享冲突对协作报酬获取的影响,系统运行期间获取的平均奖励比文献[5]Dec-MDP 增加了近 6 %。

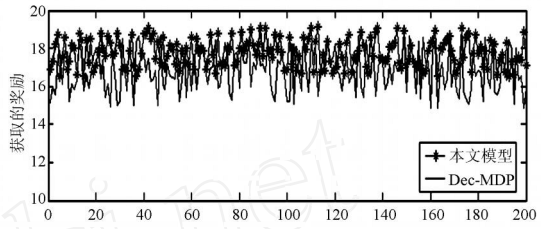


图 6 两种策略获得的奖励比较

## 4 结论

本文利用无线传感器网络中 agent 状态之间存在的条件独立性、上下文独立性关系,分解、优化 SPI 生成的策略树,使得 MAS 中处于独立状态的 agent 可以分布独立运行,只有在需要同其他 agent 协商时才进行通信。在资源分配过程中,采用 Q 分解机制优化资源配置,减少资源共享冲突。实验表明采用该方法生成的协作策略在完成协作任务获得目标奖励的同时可以有效降低通信量。

本文中 MAS 协作策略是在全部可观察的马尔可夫决策过程模型中生成的,下一步的工作就如何将本文中的方法应用到 POMDP(部分可观察)模型中展开研究。

**参考文献**:

[1] Qi H, Xu Y. Mobile Agent Based Collaborative Signal and Information Processing in Sensor Networks. Proceeding of IEEE, 2003,91(8):1172-1183.

[2] IL YAS M, MAHGOUB I. Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems [M]. USA: CRC Press, 2005.

[3] Bernstein D S, Givan R, Immerman N, and Zilberstein S. The Complexity of Centralized Control of Markov Decision Processes[J]. Mathematics of Operations Research, 2002.

[4] Goldman C V, Zilberstein S. Decentralized Control of Cooperative Systems: Categorization and Complexity Analysis [J]. Journal of AI Research, 2004.

[5] Maayan Roth, Reid Simmons and Manuela Veloso. Exploiting

Factored Representations for Decentralized[J]. AAMAS07, 2007:469-475.

[6] Boutilier C, Dearden R, and Goldszmidt M. Stochastic Dynamic Programming with Factored Representations[J]. Artificial Intelligence, 2000.

[7] Pierrick Plamondon, Brahim Chaib-draa and Abder Rezak Benaskeur. A Q-Decomposition and Bounded RTDP Approach to Resource Allocation[J]. AAMAS07, 2007:1212-1219.

[8] Stuart Russell, Andrew L. Zimdars. Q-Decomposition for Reinforcementlearning Agents[J]. Proceedings of the Twentieth International Conference on Machine Learning. 2003.

[9] 王飞,刘大有. Bayesian 网中的独立关系[J]. 计算机科学. 2001. 12:33-36.

[10] 陈志,王汝传,孙力娟. 一种无线传感器网络的多 agent 系统模型[J]. 电子学报. 2007. 2:240-243.

[11] 林华杰,史浩山. 基于无线传感器网络移动代理变种在 TinyOS 中的实现[J]. 传感技术学报. 2007. 10:2324-2327.

[12] 张新良,石纯一. M_POMDP 模型及其划分求解算法[J]. 清华大学学报(自然科学版). 2005. 10:30-36.

[13] 杨善林,胡小建. 复杂决策任务的建模与求解方法[M]. 科学出版社. 2007.

**王晓伶**(1980-),男,博士研究生,主要研究方向为网络安全、多智能体系统,wang2004mail@gmail.com

**慕德俊**(1966-),男,教授,博士生导师,主要研究方向为网络信息安全、并行计算.

**刘哲元**(1982-),男,博士研究生,主要研究方向为网络信息安全.