

# 未知环境中多 Agent 自主协作规划策略

唐贤伦, 李亚楠, 樊 峥

(重庆邮电大学工业物联网与网络化控制教育部重点实验室, 重庆 400065)

**摘要:** 针对多智能体(Agent)系统在未知环境中自主协作规划存在任务死锁及协作效率不高的问题, 提出一种基于改进蚁群算法的多 Agent 协作策略, 并用于多 Agent 协作搬运中。该方法将 Agent 所处位置和目标任务之间的距离以及信息素控制因子引入蚁群算法。实验结果表明, 该方法相比没有引入距离因子的协作方法, 协作效率更高; 相比没有引入控制因子的协作方法, 可有效防止任务死锁发生。

**关键词:** 多智能体; 未知环境; 协作; 蚁群算法; 任务死锁

**中图分类号:** TP 242

**文献标志码:** A

**DOI:** 10.3969/j.issn.1001-506X.2013.02.18

## Multi-Agent autonomous cooperation planning strategy in unknown environment

TANG Xian-lun, LI Ya-nan, FAN Zheng

(Key Laboratory of Industrial Internet of Things & Networked Control, Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**Abstract:** A multi-Agent cooperation strategy based on the improved ant algorithm is proposed for solving the problems of task deadlock and low cooperation efficiency when the multi-Agent system works in unknown environment, then the method is applied to multi-Agent cooperative transportation. The distance between the Agent's position and the task's position and the pheromone control factor are introduced in the ant colony algorithm. Experimental results show that the cooperation efficiency of this method is higher than those ones without considering the distance factor, and can avoid task deadlock more effectively than those ones without the task pheromone control factor.

**Keywords:** multi-Agent; unknown environment; cooperation; ant algorithm; task deadlock

## 0 引言

智能体(Agent)可以很好地解决一些简单的问题。但是随着问题复杂性的提高,任务会在时间和空间上超出单个 Agent 的能力范围,于是多智能体系统(multi-Agent system, MAS)被提出来了。然而在 MAS 中只依靠个体 Agent 在解决问题时是不完整、不科学、不经济的,因此协作则成为 MAS 的必不可少的行为,协作同样也是 MAS 的需要解决的关键问题<sup>[1-5]</sup>。

多 Agent 协作的研究平台很多,如机器人足球、编队、搬运问题等<sup>[6-8]</sup>。其中,机器人足球涉及合作和对抗关系,由于本身的规则较多,增加了问题的复杂性;编队侧重团队合作,不涉及对抗关系;搬运问题涉及 Agent 之间的合作,而且本身规则约束少,能够使许多复杂的概念具体化。在较复杂的场合下,如机器人协作搬运、运输和救援等<sup>[9-10]</sup>,一般涉及多 Agent 系统的多项技术,如多 Agent 之间的合作与对抗、多 Agent 编队控制<sup>[11]</sup>等,因此受到很多学者的

青睐。搬运问题的这些特点使其对多 Agent 系统的研究直接方便,在问题的说明上很具代表性,且具有较大的研究空间。

MAS 中的 Agent 在未知环境中的协作规划具有重要的意义。自主协作规划也就是由系统自主地对人物做出分配和制定出完成任务的 Agent 的数量。在很多情况下,这些协作规划的功能是很有必要的,例如搬运、月球探测等。同时这也是 MAS 对环境适应性的评判标准之一。

本文针对未知环境中多 Agent 协作搬运中的自主协作规划进行研究。搬运问题具体实现中存在两个突出问题,即如何让 Agent 放弃那些搬不动的箱子即任务死锁以及如何提高 Agent 搬箱子效率。文献[12]提出的仿生算法即蚁群算法为解决上述问题提供了理论支持。文献[13]将蚁群算法引入多机器人协作问题中,并对可能出现的任务死锁问题提出了规避算法。文献[14]提出了将信息素“置零”的更快地检测到死锁并采取规避策略的协作方法。以上方法虽然能够解决任务死锁问题,但离目标任务较近的处于空

闲状态的 Agent 很可能由于信息素不够选择概率低而放弃执行此目标任务,所以协作效率不高。针对该问题,本文将 Agent 所处位置和目标任务之间的距离以及信息素控制因子引入蚁群算法并用于多 Agent 协作搬运中。实验结果表明,该方法在多 Agent 协作搬运中既能有效解决任务死锁问题又能显著提高搬运效率。

## 1 基本蚁群算法

蚁群算法是仿真自然界中蚁群寻找从蚂蚁窝到食物源目的地之间最短路径的一种随机的搜索的算法。文献[12]在提出该算法的时候,利用了蚁群寻找食物的优化过程与著名的旅行商问题(traveling salesman problem, TSP)<sup>[15]</sup>之间的一定的相似性,通过人工地模拟蚂蚁找寻食物的行为过程来求解 TSP。

令  $m$  为蚁群中的蚂蚁的数量,  $b_i(t)$  表示在  $t$  时刻位于城市  $i$  的蚂蚁的数量,  $\tau_{ij}(t)$  表示在  $t$  时刻从  $i$  到  $j$  即  $ij$  连线上的信息素的量。在初始时刻,将  $m$  只蚂蚁随机地放到  $n$  座城市上,此时设置各路径上的初始信息量的值都为  $\tau_{ij}(0) = C$  ( $C$  为常数)。然后每只蚂蚁根据路径上残留的信息素的量独立自主地选择所要转移的下一个城市。在时刻  $t$ , 蚂蚁  $k$  从城市  $i$  转移到下一个城市  $j$  的概率定义为

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{l \in allowed_k} \tau_{il}^\alpha(t) \eta_{il}^\beta(t)}, & j \in allowed_k \\ 0, & \text{其他} \end{cases} \quad (1)$$

式中,  $allowed_k = \{0, 1, 2, \dots, n-1\} - tabu_k$  表示蚂蚁  $k$  下一步可以选择转移的所有城市集,列表  $tabu_k$  则是记录下了此蚂蚁  $k$  所走过的所有城市集,当所有  $n$  个城市都加入到了  $tabu_k$  中时候,则可认为蚂蚁  $k$  完成了一次城市循环,这个时候蚂蚁  $k$  所访问所有城市的过程中所走过的路径就是问题的一个解;  $\eta_{ij}$  作为启发式因子,用来表示蚂蚁从城市  $i$  转移到城市  $j$  的期望的高低,在蚁群算法中,  $\eta_{ij}$  通常情况下可以取值为城市  $ij$  之间距离的倒数;  $\alpha$  和  $\beta$  分别表示为路径  $ij$  上的信息素的量和启发因子的重要程度。

当蚂蚁完成了一次循环后,相应的此路径的信息素浓度根据式(2)进行更新。

$$\begin{cases} \tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij}, & \rho \in (0, 1) \\ \Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \end{cases} \quad (2)$$

式中,  $\rho$  表示为残留信息素的保留部分的程度,相应的  $1-\rho$  可以表示为信息素的挥发程度;  $\Delta \tau_{ij}$  表示为在此次循环路径上的信息素的增量值;  $\Delta \tau_{ij}^k$  可以表示为第  $k$  只蚂蚁在本次路径循环中存留在路径  $ij$  上的信息量的值。假如蚂蚁  $k$  并没有经过路径  $ij$ , 那么  $\Delta \tau_{ij}^k$  的值可认为为零, 于是  $\Delta \tau_{ij}^k$  表示为

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{第 } k \text{ 只蚂蚁经过路径 } ij \\ 0, & \text{其他} \end{cases} \quad (3)$$

式中,  $Q$  为常数;  $L_k$  表示第  $k$  只蚂蚁在此次路径循环的过程中所走过的的路径的长度的值。

## 2 基于蚁群算法的多 Agent 协作策略

### 2.1 多 Agent 通信方式

黑板模型是为了解决分布在不同环境下多个主体协作完成目标任务的分布和并行计算模型。该模型能够实现异构知识源的集成,是 Agent 请求或者信息共享的主要通信模式。它管理一个 Agent 通信讨论区,其他 Agent 将需要共享的信息写入到讨论区中以实现信息的共享。黑板模型可以为 Agent 通信模型提供如下条件: Agent 的并发控制、Agent 的交互、Agent 的实时控制等。即在 MAS 中的每一个 Agent 都会按照一定的时间间隔从“黑板”中获取所感兴趣、于自己有利的信息,同样, Agent 也会把自己的一些信息简单地放到“黑板”中去。这样 MAS 中的 Agent 就可以通过“黑板”系统达到信息共享的目的。

本实验中采用的是“黑板原理”的通信方式,这一方法具有高灵活、强鲁棒性等特点,并且不会由于个体或者是局部的失误影响到系统的全局性能。

### 2.2 多 Agent 协作策略

设  $n$  个 Agent 进入到一个包含着  $m$  件未完成的任务的未知区域中。这些 Agent 首先要进行独立的任务搜寻,即所有 Agent 都在这个未知区域中随机游荡着搜寻未完成任务。在开始时刻对应于所有未完成任务  $j$  ( $j=1, 2, \dots, m$ ) 的“激素” $\tau_j$  为零。

若 Agent  $i$  ( $i=1, 2, \dots, n$ ) 发现未完成任务  $j$  时, Agent 试着自己完成这一任务  $j$ 。如成功完成了任务则 Agent  $i$  将任务  $j$  已完成情况写到“黑板”上;如不成功,它则将任务  $j$  的相关信息(如位置)存到“黑板”系统上,来“告诉”其他空闲的 Agent,并将任务  $j$  相对应的信息素的量  $\tau_j$  赋予一个大于零的值,于是

$$\tau_j = \Delta \tau > 0 \quad (4)$$

同时, Agent  $i$  在任务  $j$  处等待一段时间。

若 Agent  $i$  没有发现未完成任务,即它处于没有执行任何任务的“空闲”状态的时候,它则每当间隔一段时间就在“黑板”上的任务相关区域进行任务搜索。如果没有发现任何一件任务的信息素的值大于零,那么它将继续在此未知区域中随机游荡搜索任务。如果发现了有  $k$  件任务的信息素的  $\tau_s$  ( $s=1, 2, \dots, k$ ) 值大于零,则计算与任务  $s$  相关的概率为

$$p_s = \frac{(\tau_s)^\alpha}{\sum_{s=1}^k (\tau_s)^\alpha}, \quad s = 1, 2, \dots, k \quad (5)$$

Agent  $i$  在下一步以概率  $p_s$  选择任务  $s$ , 表示为

$$p(s = 0 \rightarrow s = 1) = p_s \quad (6)$$

若 Agent  $i$  发现任务  $s$  并执行,且 Agent 能够独自完成此任务  $s$ , 或可以与在  $s$  处等待的其他 Agent 一起完成任务,则 Agent 将黑板上与此任务  $s$  对应的信息素  $\tau_s$  的值,置

成零。这样做的目的是避免其他 Agent 受到错误任务信息的指导,仍然继续向  $s$  所在的区域靠拢,造成资源的浪费,并且太多的 Agent 聚集在同一个区域很有可能会造成一定的运动阻塞。若 Agent  $i$  未能完成或与在  $s$  处等待的其他 Agent 一起仍然没有完成此任务,则此任务的信息素的  $\tau_s$  值继续保持大于零的数。如果在多个 Agent 参与协作的情况下仍不能完成,则将  $\tau_s$  更新为

$$\tau_s = \tau_s + h\tau \quad (7)$$

式中,  $h\tau$  是增加的“激素”浓度,  $h$  是参与协作的 Agent 数量。以此来吸引更多的 Agent 参与协作。

### 3 引入距离因子和控制因子的改进蚁群算法

当 Agent 执行与一件超出自己能力的任务,同时不再被其他任务所吸引,则定义这个 Agent 陷入任务死锁。

基本蚁群算法可以较好地完成多 Agent 的协作,但是,通过仿真发现基本蚁群算法用于多智能体的协作会出现任务死锁现象。一个任务的完成难度很大,以致所有的智能体 Agent 协作也无法完成,此时如果不跳出死锁,则多智能体的协作会陷入一种死循环状态,即上面所说的任务死锁。为了防止任务死锁的发生,本文改进蚁群算法,引入 Agent 所处位置和目标任务之间的距离和自适应控制因子,使得个体 Agent 具备摆脱任务死锁继续完成其他任务的能力。

算法实现的思路是:若处于空闲状态的个体 Agent  $i$  发现仿真环境中存在着  $k$  件任务,且这些任务的信息素浓度都为大于零的值,分别为  $\tau_s (s=1, 2, \dots, k)$ , 于是计算与这些任务相关的概率为

$$p_{is} = \frac{(\tau_{is})^\alpha (\eta_{is})^\beta}{\sum_{s=1}^k (\tau_{is})^\alpha (\eta_{is})^\beta} \quad (8)$$

式中,  $\tau_{is}$  是 Agent  $i$  所接触到的“激素”浓度,初始化  $\tau_{is} = \tau_s$ ;

$\eta_{is} = \frac{1}{d_{is}}$  为距离因子,  $d_{is}$  为 Agent  $i$  和任务  $S$  之间的欧氏距离。Agent  $i$  在下一步以概率  $p_{is}$  选择任务  $S$ , 即

$$p(s=0 \rightarrow s=1) = p_{is} \quad (9)$$

设 Agent 完成任务所需要的时间为  $T$ , 经过时间  $T$  后, Agent  $i$  若未能完成任务  $s$ , 则放弃  $s$ , 重新进行选择, 引入控制因子  $\lambda$ , 修改  $\tau_{is}$  为

$$\tau_{is} = \lambda\tau_{is} \quad (10)$$

$$\lambda = 1 - \frac{n_t - 1}{n} \quad (11)$$

式中,  $n_t$  为参与该任务的 Agent 数量;  $n$  为 Agent 的总数量。可以看出, 当 Agent 发生任务死锁时,  $n_t = n + 1, \lambda = 0, \tau_{is} = 0, p_{is} = 0$ 。这就使得 Agent  $i$  放弃任务  $s$  而执行其他任务变成可能。

由式(8)可知, 处于空闲状态的 Agent 离目标任务较近信息素高选择概率大使执行此目标任务的可能性增大, 提高了协作效率。

## 4 实验及结果分析

### 4.1 搬运问题

搬运问题即一群 Agent 协作搬运未知环境中的若干箱子, 该问题描述的是在一未知的二维网格世界中,  $N$  个 Agent 协作搬运  $M$  个箱子, 每个 Agent 在一个时间单位内只能横向或者纵向移动一个网格, 箱子随机摆放, 且不允许两个 Agent 或箱子占据相同位置, 最终搬运成功是指多个 Agent 通过协作将全部箱子运完(搬不动则放弃)。

在搬运问题中可以设置的参数如下: 成功搬运箱子、搬运区域大小和形状、Agent 的移动规则、所能探测到的范围、Agent 之间是否能够通信以及通信的方式。

### 4.2 搬运问题的参数及任务的描述

根据以上所述, 现将参数设定如下: ①成功搬运箱子。在本文中成功搬运箱子的定义为把区域中的所有箱子全部搬完(死锁例外)。②搬运区域的大小和形状。本文中定义的区域是一个封闭的二维栅格结构的工作区域, 该区域的大小为  $800 \text{ cm} \times 600 \text{ cm}$ , 形状为  $1 \text{ cm} \times 1 \text{ cm}$  的方格, 在该工作区域中有 3 个 Agent, 4 个箱子, 每个 Agent 占据 20 个方格, 每个箱子占据 20 个方格。③Agent 的移动规则。搬运者 Agent 和箱子的初始位置是随机的。搬运者 Agent 的移动方式是可以横纵向移动也可延对角线移动, 且所有 Agent 每次移动的距离都只有 20 格。④所能探测到的目标及范围。对于搬运者 Agent 来说, 本文设定它所能探测到的范围是 50 个格。⑤搬运者之间是否能够通信以及通信的方式。搬运者 Agent 最初的运动方式是随机的, 在搬运者 Agent 探测到箱子时, 搬运者 Agent 之间开始通过协作来搬运箱子。

### 4.3 协作搬运问题仿真

仿真环境为  $800 \text{ cm} \times 600 \text{ cm}$  的世界, 如图 1 所示。世界中分布着 4 个箱子, 质量分别为 1 kg、3 kg、5 kg 和 7 kg。有 3 个 Agent 参与搬运任务, 在世界中找到这些箱子并搬运回家。家的区域如图 1 左上角 Home 区域。

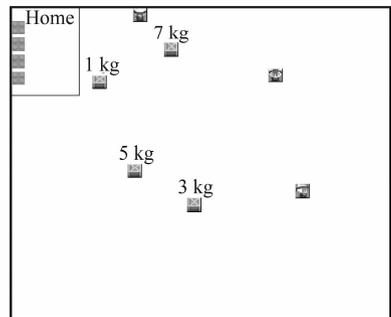


图1 协作搬运的初始界面

Agent 搬运能力、移动速度等参数的设置界面以及结果输出如图 2 所示。由于每个 Agent 的搬运能力设置为

2 kg,所以 1 kg 的箱子只需 1 个 Agent 就可以完成,3 kg 的箱子需 2 个 Agent 协作搬运,5 kg 的箱子需要 3 个 Agent 参与,而 7 kg 箱子即使全部 Agent 都参与协作也搬不动,即出现了任务死锁。

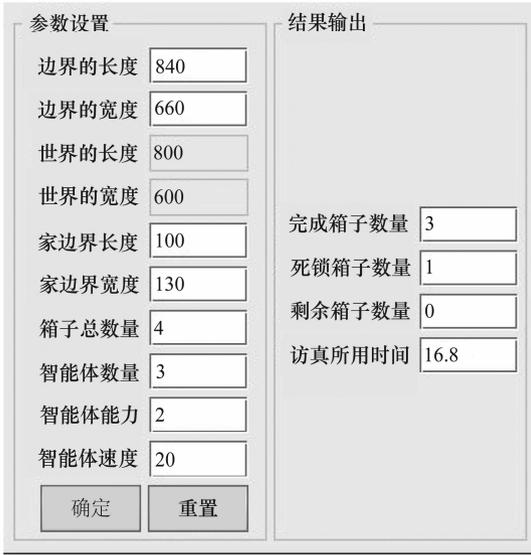


图 2 实验系统参数设置及结果输出

如图 3 所示,搬运者 Agent 放弃了 7 kg 箱子,已经完成了对 1 kg 箱子的搬运,防止了任务死锁的发生。

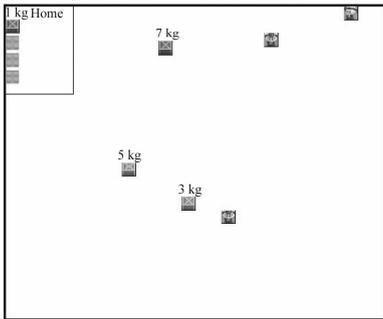


图 3 7 kg 箱子已被放弃,已经完成了对 1 kg 箱子的搬运

如图 4 所示,搬运者正在搬运 3 kg 的箱子。

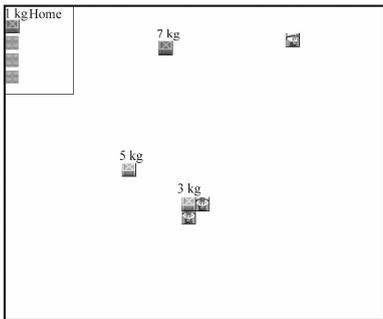


图 4 7 kg 箱子已被放弃,正在搬运 3 kg 的箱子

如图 5 所示,搬运者 Agent 已经完成了对 1 kg 和 3 kg 箱子的搬运,正在搬运 5 kg 的箱子。

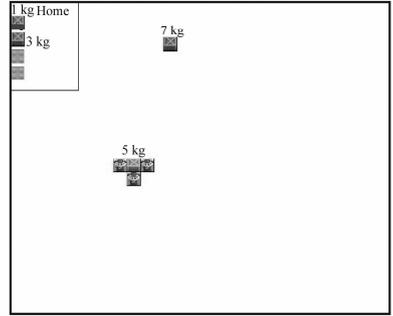


图 5 1 kg、3 kg 箱子搬运完成,正在搬运 5 kg 的箱子

如图 6 所示,搬运者 Agent 已经完成了对 1 kg、3 kg、5 kg 和 7 kg 箱子的搬运,仿真结束。

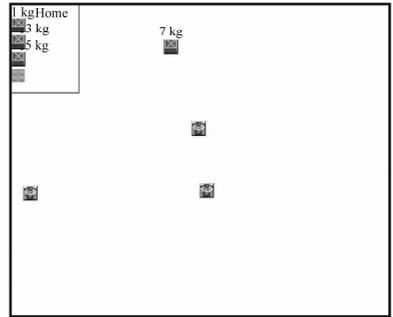


图 6 箱子搬运完成,仿真结束

为验证本文提出的引入距离因子和控制因子的协作算法的有效性,需做两类不同的实验。首先,将基本协作算法与引入距离因子和控制因子的协作算法进行任务死锁实验。其次,将规避算法、信息素“置零”算法与本文提出的引入距离因子和控制因子的协作算法进行搬运效率对比实验。

分别用基本蚁群协作算法、规避算法、“置零”算法以及引入距离因子和控制因子协作算法进行 Agent 搬运仿真实验,实验结果见表 1。随着实验次数的增加,基本蚁群算法出现任务死锁的次数增加,而其余 3 种算法都没出现任务死锁。说明:规避算法、“置零”算法以及引入距离因子和控制因子的协作算法都能防止任务死锁的发生。

表 1 不同协作算法任务死锁实验结果

算法	100 次	1 000 次	1 000 次	3 组实验死锁次数占总次数百分比/%
	实验死锁次数	实验死锁次数	实验死锁次数	
基本蚁群算法	270	322	249	28.03
规避算法	0	0	0	0
信息素“置零”	0	0	0	0
本文算法	0	0	0	0

表 2 为文献[13]中的规避算法、文献[14]中的“置零”算法以及引入距离因子和控制因子的协作算法在多次实验中 Agent 移动平均步数。规避算法和“置零”算法平均每次完成实验 Agent 移动的次数分别为 486 次和 382 次。而引

入距离因子和控制因子的协作算法平均完成实验 Agent 移动的次数为 296。较之规避算法和“置零”算法有大幅提高。说明本文提出的引入距离因子控制因子的协作策略能减少 Agent 移动次数,节省了搬运成本,提高了协作性能,提高了协作效率。较之“置零”算法提高了 22.513% 的性能,而较之规避算法提高了 39.095% 的性能。

表 2 不同协作算法性能试验结果

算法	1 000 次实验机器人移动平均步数			3 组实验机器人 平均移动步数
	第 1 组	第 2 组	第 3 组	
文献[13]算法	442	520	497	486
文献[14]算法	369	401	376	382
本文算法	284	311	293	296

仿真结果表明,引入距离因子和控制因子能有效解决任务死锁问题且提高了协作效率。

## 5 结 论

本文针对 MAS 于未知环境下的自主协作规划中存在的任务死锁以及协作效率不高的问题,将 Agent 所处位置和目标任务之间的距离以及信息素控制因子引入蚁群算法并应用到多 Agent 协作搬运问题中,使得多 Agent 系统具有在未知环境中自主协作和规划的能力,并防止了任务死锁的发生,优化了协作算法,提高了协作效率。实验结果表明,本文提出的将距离因子和控制因子引入蚁群算法的多 Agent 协作策略,为解决 MAS 在协作中可能出现的任务死锁及协作效率不高的问题提供了有效的解决途径。

## 参考文献:

[1] Riedmiller M, Gabel T. On experiences in a complex and competitive gaming domain: reinforcement learning meets robot cup[C]//*Proc. of the 3rd IEEE Symposium on Computational Intelligence and Games*, 2007:17-23.

[2] Jiao W P. Multi-Agent cooperation via reasoning about the behavior of others[J]. *Computational Intelligence*, 2010, 26(1):57-83.

[3] Andrew L, Fan W. Multi-depots vehicle muting problem: a one-stage approach[J]. *IEEE Trans. on Automation Science and Engineering*, 2005, 2(4):397-402.

[4] Semsar-Kazerooni E, Khorasani K. A game theory approach to multi-Agent team cooperation[C]//*Proc. of the American Control Conference on Hyatt Regency Riverfront*, 2009:10-12.

[5] Liu D Y, Li J. Advances in multi-Agent coordination[C]//*Proc. of the International Conference on Machine Learning and Cybernetics*, 2006:125-130.

[6] Alhashel E. A conceptual Agent cooperation model for multi-Agent systems' team formation process[C]//*Proc. of the Third International Conference on Convergence and Hybrid Information Technology*, 2008:12-20.

[7] Li T J, Peng Y Q, Wu J G. Study on multi-Agent task allocation based on the contract net[C]//*Proc. of the IEEE International Conference on Control and Automation*, 2007:2317-2321.

[8] Hou X P, Wang Y W, Wang H. Research on coordination decision of CSCM based on multi-Agent system[J]. *Journal of Systems & Management*, 2008, 17(5):548-552.

[9] Bellifemine F, Caire G, Greenwood D. Developing multi-Agent systems with JADE[M]. West Sussex, England: Wiley, 2007.

[10] Vieira R, Moreira A, Wooldridge M, et al. On the formal semantics of speech-act based communication in an Agent-oriented programming language[J]. *Artificial Intelligence Research*, 2007, 29(1):221-267.

[11] Bulka B, Gaston M, Desjardins M. Local strategy learning in networked multi-Agent team formation [J]. *Autonomous Agents and multi-Agent Systems*, 2007, 15(1):318-324.

[12] Han Y F, Shi P F. An improved ant colony algorithm for fuzzy clustering in image segmentation [J]. *Neuron Computing*, 2007, 70(27):665-671.

[13] 丁滢颖, 何衍, 蒋静坪. 基于蚁群算法的多机器人协作策略[J]. 机器人, 2003, 25(5):414-418. (Ding Y Y, He Y, Jiang J P. Multi-robot cooperation method based on the ant algorithm[J]. *Robot*, 2003, 25 (5):414-418.)

[14] 李玉坚, 刘国栋. 一种基于蚁群算法的多机器人协作方法[J]. 江南大学学报(自然科学版), 2007, 6(2):132-135. (Li Y J, Liu G D. A multi-robotics cooperation method based on the ant algorithm[J]. *Journal of Southern Yangtze University (Natural Science Edition)*, 2007, 6(2):132-135.)

[15] 俞庆生, 林冬梅, 王东. 多旅行商问题研究综述[J]. 价值工程, 2012, 31(2):166-168. (Yu Q S, Lin D M, Wang D. An overview of multiple traveling salesman problem[J]. *Value Engineering*, 2012, 31(2):166-168.)

## 作者简介:

唐贤伦(1977-),男,副教授,博士,主要研究方向为多 Agent 系统、群体智能。

E-mail: tangxun@hotmail.com

李亚楠(1985-),男,硕士研究生,主要研究方向为多智能体协作。

E-mail: lynlkx@126.com

樊 峥(1985-),男,硕士研究生,主要研究方向为群体智能、智能系统。

E-mail: ertouyizhong@163.com