文章编号: 1001-0920(2006)04-0430-05

基于强化学习算法的多机器人系统的冲突消解策略

任 燚, 陈宗海
(中国科学技术大学 自动化系, 合肥 230027)

摘 要: 多机器人系统中, 随着机器人数目的增加, 系统中的冲突呈指数级增加, 甚至出现死锁. 本文提出了基于过程奖赏和优先扫除的强化学习算法作为多机器人系统的冲突消解策略. 针对典型的多机器人可识别群体觅食任务, 以计算机仿真为手段, 以收集的目标物数量为系统性能指标, 以算法收敛时学习次数为学习速度指标, 进行仿真研究, 并与基于全局奖赏和Q 学习算法等其他 9 种算法进行比较. 结果表明所提出的基于过程奖赏和优先扫除的强化学习算法能显著减少冲突, 避免死锁, 提高系统整体性能.

关键词: 多机器人; 过程奖赏; 优先扫除; 强化学习
中图分类号: TP242 **文献标识码**: A

Interference Solving Strategy in Multiple Robot System Based on Reinforcement Learning Algorithm

REN Yi, CHEN Zong-hai
(Department of Automation, University of Science and Technology of China, Hefei 230027, China. Correspondent: CHEN Zong-hai, E-mail: chenzh@ustc.edu.cn)

Abstract: In a multiple mobile robot system, interface increases exponentially with the increasing number of robots, even deadlock may occur. A reinforcement learning algorithm based on process reward and prioritized sweeping is presented as interference solving strategy. Simulation experiments for forage as task verify the system performance of collected attractors and the learning rate. Comparisons of other nine strategies such as the algorithm based on global reward and Q-learning, show that the presented algorithm based on process reward and prioritized sweeping can decrease interference, avoid deadlock and improve group performance.

Key words: Multiple mobile robot system; Process reward; Prioritized sweeping; Reinforcement learning

1 引 言

近年来, 基于行为的自主移动机器人系统由于具有突出的鲁棒性、灵活性和容错性等优点, 获得了越来越多的关注[1, 2]. 多机器人系统是典型的多智能体系统, 在非结构化的环境中, 如何有效地组织和协调多个智能体完成复杂任务, 已成为人工智能和机器人学的研究热点. 但随着机器人数目的增加, 受系统中有限资源的制约, 机器人间的冲突呈指数级增长, 甚至发生死锁, 致使整个系统瘫痪. 因此, 多机器人的冲突消解和死锁问题的解决引起了众多研究者的关注.

多机器人系统运行过程中会发生多种形式的冲突, 本质上都是对系统内有限资源的竞争, 这种冲突达到极端的情况, 就表现为一种死锁的形式——多个机器人之间完全僵持的局面. 为了解决这个问题, Balch[3] 提出了一种地域型异构觅食策略; Wang 等[4, 5] 也提出了基于局部感知的排队协调策略和广播式通讯策略.

这些策略虽然能在一定程度上提高多机器人系统的群体性能, 但都是基于对特定任务和环境下的特定冲突形式进行手工编程实现的. 多机器人系统中, 当任务和环境变得复杂时, 要完全依靠程序员的手工编程实现其基本行为的设计, 任务变得非常繁重, 甚至是不可能的, 同时对硬件的要求也很高. 在

收稿日期: 2005-03-28; **修回日期**: 2005-06-14
作者简介: 任燚(1977—), 女, 河南洛阳人, 博士生, 从事机器人行为学的研究; 陈宗海(1963—), 男, 安徽桐城人, 教授, 博士生导师, 从事复杂系统的建模、仿真与控制等研究.

这种背景下, 具有自学习能力的机器人成为一个新的研究热点, 这个研究方向的一个关键问题是利用学习技术增强机器人的智能, 即其自主解决问题的能力, 在诸多学习方法中, 强化学习是得到较广泛关注的一种学习方法[6].

本文以基于过程奖赏和优先扫除[7]的强化学习算法(PS-process)作为多机器人系统的冲突消解策略, 以收集的目标物数目为系统性能指标, 以算法收敛时学习次数为学习速度指标, 进行多机器人的可识别群体觅食仿真实验, 并与基于全局奖赏、局部奖赏(local)、子任务方法(subtask)、过程奖赏的Q 学习算法、PS 算法以及手工编程的同构策略(homo)和地域型异构策略(hetero)进行比较. 这样共对 10 种算法即Q-global, Q-local, Q-subtask, Q-process, PS-global, PS-local, PS-subtask, PS-process, Hand-homo 和 Hand-hetero 进行了仿真实验比较.

2　基于过程奖赏的冲突消解策略

目前多机器人系统应用的强化学习算法主要通过两种方式: 一种是直接应用单机器人的强化学习算法, 也有的通过与其他机器人适当交互来加快学习过程; 另一种是将整个多机器人系统看成一个单一的学习系统, 这种系统假设使得多机器人系统的实际应用特别困难[8]. 本文直接应用单机器人的Q 学习和优先扫除算法, 但考虑多机器人系统的特性, 改进其奖赏函数.

强化学习算法的一个重要特征是完成任务后获得奖赏. 对于多机器人间的基于强化学习算法的协调机制, Balch[3]提出了全局奖赏和局部奖赏函数. 全局奖赏是当任何一个机器人获得目标物并把它放入基地区时, 奖励多机器人系统中每个机器人, 即

$$r_{\mathrm{global}}(t)=\begin{cases}1, & \text{If any agent delivered an attractor}\\ & \text{to homezone at time } t-1;\\ -1, & \text{Otherwise.}\end{cases}\tag{1}$$

而局部奖赏是当任何一个机器人获得目标物并把它放入基地区时, 只是奖励这个机器人, 即

$$r_{\mathrm{local}}(t)=\begin{cases}1, & \text{If the agent delivered an attractor}\\ & \text{to homezone at time } t-1;\\ -1, & \text{Otherwise.}\end{cases}\tag{2}$$

这两种奖赏都是基于性能的一种奖赏, 即结果奖赏, 它只简单地给机器人表达任务, 而没有列举怎样执行任务. 比如机器人系统在完成觅食任务时, 仅当一个机器人成功收集目标物并将之放入基地区时才对一个或多个机器人给予奖赏. 这种奖赏函数的最大问题是奖赏被延迟了, 机器人必须成功完成一系列动作后才能得到奖赏(一般来说, 一个任务都由一系列的动作组成), 这会给动作间的奖赏分配带来困难. 同时全局奖赏的每个机器人都必须考虑其他机器人的状态, 选择动作时也必须考虑集体的利益, 所以具有状态空间和动作空间庞大的特点, 学习速度很慢, 甚至并不收敛. 而局部奖赏可能由于奖赏延迟导致错误奖赏, 使得偶然体现出较高性能的机器人不断获得较大的强化信号, 也使得对它的性能评价不断增长, 而总体性能更好的机器人却因此得不到足够大的强化信号.

为了解决这个问题, Mahadevan[9] 提出了子任务的方法, 子任务方法是将整体任务的学习分解成多个不同子任务的学习的方法, 利用不同的行为状态定义各个子任务, 同时定义了从一个状态转移到另一个状态的条件. 显然这种方法减少了学习空间的大小, 从而加快了学习收敛速度. 根据这种方法, 机器人觅食任务可分解成两个子任务: 捡拾到和递送好(目标物)

$$r_{\mathrm{subtask}}(t)=r_{\mathrm{acquire}}(t)+r_{\mathrm{deliver}}(t),\tag{3}$$

子任务方法的奖赏由两部分组成, $r_{\mathrm{acquire}}(t)$ 和 $r_{\mathrm{deliver}}(t)$ 分别是捡拾到和递送好(目标物) 的奖赏

$$r_{\mathrm{acquire}}(t)=\begin{cases}1, & \text{If the agent picked up}\\ & \text{attractor at time } t-1;\\ -1, & \text{Otherwise.}\end{cases}\tag{4}$$

$$r_{\mathrm{deliver}}(t)=\begin{cases}1, & \text{If the agent delivered attractor}\\ & \text{to homezone at time } t-1;\\ -1, & \text{Otherwise.}\end{cases}\tag{5}$$

但是, 子任务方法的奖赏本质上还是一种结果奖赏, 只是把一个整体的结果奖赏按照子任务分解为几个不同的子结果奖赏, 因此没有解决局部奖赏的根本问题.

针对这种情况, 本文提出过程奖赏的方法. 所谓过程奖赏是指关注完成任务过程中每个动作同时关注完成任务的趋势, 即同时考虑机器人在某时刻的状态是接近或远离完成任务. 过程奖赏函数能实时对机器人完成任务的每个动作和趋势进行奖赏

$$r_{\mathrm{process}}(t)=r_{\mathrm{action}}(t)+r_{\mathrm{trend}}(t).\tag{6}$$

过程奖赏由两部分组成, $r_{\mathrm{action}}(t)$ 和 $r_{\mathrm{trend}}(t)$ 分别是对动作和趋势进行奖赏. 针对机器人觅食任务, 其动作包括放置目标物到基地区、机器人拾取目标物和遗弃目标物于非基地区, 而完成觅食任务的趋势包括机器人拾取目标物并朝基地区移动和机器人拾取目标物背离基地区移动. 因此

$$r_{\text{action}}(t)=\begin{cases}2, \text{ If delivered attractor to}\\ \quad \text{homezone at time } t\text{-}1;\\ 2, \text{ If picked up attractor}\\ \quad \text{at time } t\text{-}1;\\ -\ 2, \text{ If dropped attractor outside}\\ \quad \text{homezone at time } t\text{-}1;\\ -\ 1, \text{ Otherwise.}\end{cases} \tag{7}$$

$$r_{\text{trend}}(t)=\begin{cases}0.5, \text{ If holding attractor and moving}\\ \quad \text{towards homezone at } t\text{-}1;\\ -\ 0.5, \text{ If holding attractor}\\ \quad \text{and moving away}\\ \quad \text{from homezone at } t\text{-}1;\\ 0, \text{ Otherwise.}\end{cases} \tag{8}$$

相比于上述的 3 种奖赏, 本文提出的过程奖赏可以从以下 4 个方面增强学习算法的鲁棒性, 从而提高算法收敛速度和机器人的系统性能: 1) 过程奖赏对机器人的每个动作都能提供实时奖赏. 当机器人完成一项任务时, 强化学习算法通过试错而获得大量的经验, 结果奖赏获得的经验仅被一次性用于调整 Q 函数, 而过程奖赏关注机器人的每个动作和趋势, 充分利用强化学习自身产生的经验, 并能实时提供奖赏. 2) 过程奖赏可以终止某些行为, 鼓励尝试新行为带来奖赏. 完成复杂的任务一般都由一系列的行为组成, 完成任务最终都会产生一个奖赏信号, 因此结果奖赏在获得奖赏之前无法停止正在进行的行为, 如机器人拾取目标物时, 它的下一步行为是朝基地区移动并将目标物放入基地区才能获得奖赏, 而过程奖赏可以由于系统冲突终止朝基地区移动, 继续拾取别的目标物照样可以获得一定的奖赏. 因此过程奖赏函数给行为停止提供了"非单一"的方法. 3) 过程奖赏降低了在特定条件下由于错误的行为而获得的偶然奖赏. 过程奖赏是一种增量式的奖赏, 而不是结果奖赏的那种"一步到位"式的奖赏, 因而能减少间断和偶然的成功所带来的奖赏. 4) 过程奖赏还可以通过加强条件 —— 行为关系降低强化学习算法对噪声的敏感度. 过程奖赏不像结果奖赏那样, 对可能有噪声影响的结果(如成功收集目标物并将之放入基地区) 进行惟一的奖赏, 因而对噪声影响不大. 过程奖赏函数为由噪声造成的间断和潜在的错误奖赏提供了去噪效果.

3 仿真实验研究

为了比较分析 PS-process 在提高多机器人系统性能和学习速度方面的优越性, 本文针对多机器人的可识别群体觅食任务(机器人需要识别目标物的颜色并将之递送到对应的基地区), 采用不同的仿真环境和学习算法, 进行几类实验. 实验中系统性能的体现是机器人收集到基地区的总的目标物数量, 显然数量越多代表系统性能越好, 但学习速度(算法收敛速度) 体现在算法收敛时的学习次数, 学习次数越少, 学习速度越快, 这里的算法收敛是指一定学习次数后每次学习机器人的策略改变次数(Q 值表示改变次数) 的平均值基本稳定.

3.1 仿真环境

仿真实验平台是以 Nomad 150 为模拟对象, 平台的控制系统是基于运动模式编制的, 由 4 类基本行为构成: avoid 躲避行为、move- to 移动行为、swirl- to 绕行行为和 noise 噪声行为[2]. 本文的手工编程策略是按照有限状态机(FSA)[3] 描述的那样按部就班地执行 6 个行为: 漫游、捡拾红(绿) 目标物、递送红(绿) 目标物、搜索基地区. 而基于学习的机器人, 则是通过强化学习算法选择 6 个行为中的某个行为, 进而由奖赏函数提供奖赏信号反馈给机器人.

本文觅食任务的环境有两类: 静态环境和动态环境, 图 1 所示的是 8 个机器人的静态环境, 环境中有 7 个圆形和 8 个菱形固定障碍物, 1 个红色和 1 个绿色固定基地区以及 60 个红色和 60 个绿色固定目标物. 动态环境与静态环境的区别在于其绿色目标物是随机移动的, 机器人的任务是搜寻目标物并把它放入正确的基地区中. 为了比较算法的系统性能和学习速度, 所采用的学习算法包括 4 种奖赏函数的 Q 学习算法和优先扫除算法, 手工编程算法包括同构策略和地域型异构策略. 这 10 种算法分别在每类环境上进行 300 次学习, 每次学习花费 10 min 仿真时间. 强化学习算法的折扣因子 $\gamma=0.8$, 学习因子 $\alpha=0.2$.

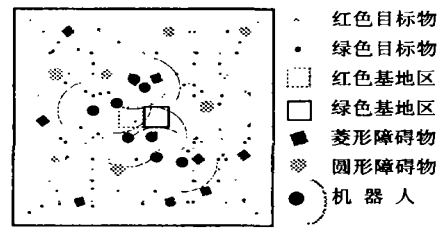


图 1 8 个机器人的静态仿真环境

3.2 仿真结果与讨论

表 1 和表 2 分别是多机器人系统的可识别群体觅食任务的群体规模从 2 到 16 个机器人的系统性能和学习速度. 其中表 1 的数值是算法收敛时的学习次数(表 2 中的相应数值) 到 300(每种算法总的学习次数) 这个区间的平均每次学习机器人所收集

表 1　群体规模系统性能表

	机器人数	Hand-homo	Hand-hetero	Q-global	PS-global	Q-local	PS-local	Q-subtask	PS-subtask	Q-process	PS-process
静态环境	2	39.02	39.18	9.78	15.27	35.77	37.42	36.57	39.12	40.54	41.67
	4	57.49	57.56	3.94	10.07	53.38	59.83	53.88	60.02	59.89	60.12
	6	67.85	67.79	7.09	12.00	62.85	62.88	62.78	63.17	62.95	68.49
	8	68.12	69.66	5.80	10.66	64.60	70.32	65.07	70.33	19.91	71.48
	10	69.69	70.77	5.75	13.73	58.29	58.39	59.12	60.38	14.15	75.25
	12	55.33	63.41	5.37	16.76	43.76	47.66	48.55	49.79	14.06	75.58
	14	40.02	49.44	4.04	6.65	32.52	39.84	33.13	42.51	13.98	72.96
	16	32.54	32.57	4.68	16.90	25.45	29.03	25.55	29.78	13.65	70.24
动态环境	2	34.68	37.25	8.34	8.93	8.93	9.24	9.38	12.11	12.16	39.67
	4	44.46	46.15	11.91	12.86	15.95	25.02	18.12	28.79	12.18	54.53
	6	60.26	65.26	16.54	16.81	21.79	43.91	22.33	48.15	11.76	66.86
	8	66.76	68.32	15.93	16.71	23.31	40.44	24.28	50.12	12.18	70.47
	10	64.42	66.32	11.65	12.06	26.26	44.65	27.11	52.15	9.49	75.01
	12	52.83	54.86	16.64	14.51	25.26	31.01	25.55	34.79	17.00	74.10
	14	44.07	50.06	13.74	11.39	22.33	27.84	23.17	28.12	10.32	72.67
	16	36.64	37.44	13.77	10.11	19.31	21.60	21.05	23.90	16.48	69.64

表 2　群体规模学习速度表

	机器人数	Q-global	PS-global	Q-local	PS-local	Q-subtask	PS-subtask	Q-process	PS-process
静态环境	2	180	126	174	123	152	108	100	72
	4	184	128	178	128	168	113	102	76
	6	190	130	182	130	173	136	105	92
	8	200	125	220	124	199	128	105	96
	10	220	108	128	118	110	118	98	98
	12	244	102	130	120	122	121	108	100
	14	250	110	135	121	138	127	110	108
	16	278	112	138	128	144	132	112	110
动态环境	2	96	148	94	146	92	117	26	60
	4	100	170	100	150	98	110	30	75
	6	120	175	110	152	107	120	50	78
	8	178	202	154	104	114	112	54	80
	10	180	210	156	108	133	121	70	85
	12	220	225	160	125	140	132	50	100
	14	266	126	178	175	142	165	50	110
	16	270	128	196	144	150	125	52	112

的总的目标物数量, 如Q-global 的目标物数量 9.78 是学习次数从 180 到 300 这个区间的平均每次学习 2 个机器人所收集的总的目标物数量

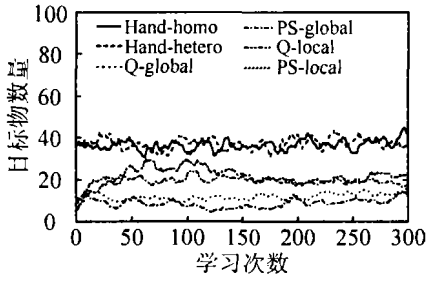
从表 1 中可以看出, PS-process 在所有的工况(不管工况中机器人数量的多少, 还是静态环境或动态环境) 所收集到的目标物总是最多, 即其系统性能最好. 这表明了本文应用基于过程奖赏和优先扫除 PS-process 作为多机器人系统的冲突消解策略是有效的, 它能显著减少冲突, 提高系统整体性能

观察两种手工编程策略——同构策略和地域型异构策略, Hand-hetero 的系统性能总比 Hand-homo 好, 这与 Balch[3] 的结果一致. 观察学习算法, 仅有 PS-process 的系统性能好于手工编程, 基于全局奖赏算法(Q-global 和 PS-global) 的系统性能与手工编程相差太多, 因此这两种算法并不适合多机器人系统的可识别群体觅食任务. PS-local 比 Q-local 的系统性能要好, 但在静态环境中当机器人数量超过 8 个, 在动态环境中当机器人数量超过 10 个时,

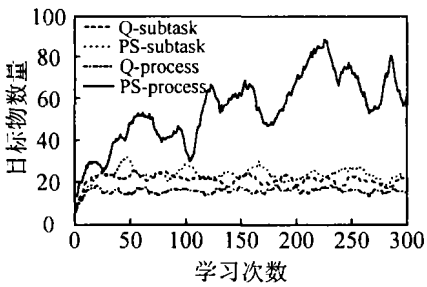
系统性能均开始下降. 这种规律也适合于 PS-subtask 和 Q-subtask, 同时基于子任务方法的系统性能都比相应的基于局部奖赏函数的策略要好. Q-process 在静态环境中机器人数量较少时, 系统性能与手工编程差不多, 但当机器人数量增多或系统处于动态环境中时, 系统性能急剧下降. 虽然 PS-process 在众多的策略中系统性能最好, 但当机器人数量增多时, 其系统性能也有一定的下降, 在本文特定的任务和环境中, PS-process 在静态环境中当机器人数量超过 12 个, 在动态环境中当机器人数量超过 10 个时, 系统性能也有一定的下降. 然而即使这样, PS-process 收集到的总的目标物数量仍远多于其他策略.

比较这 8 种算法的学习速度, 如表 2 的群体规模学习速度表所示, 静态环境中 PS-process 的学习速度最快, 而在动态环境中 Q-process 的学习速度最快, 这表明相对于全局奖赏、局部奖赏和子任务方法, 基于过程奖赏的学习速度最快, 基于全局奖赏的学习速度最慢.

图 2 和图 3 是 16 个机器人的动态环境的仿真实验结果, 其中图 2 是 10 种算法(包括手工编程策略和强化学习算法)的系统性能比较, 图 3 是其中 8 种强化学习算法的学习速度比较. 从图中可以看出, PS-process 所收集的目标物数量是其他 9 种算法的两倍多; 而在学习速度方面, 基于过程奖赏的学习算法最好. 图 2 和图 3 中各算法所收集的目标物数量

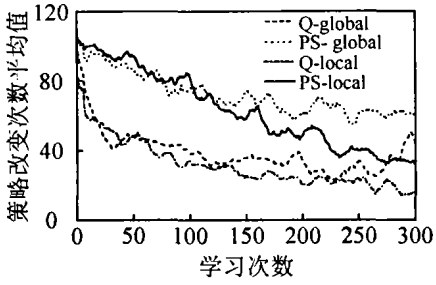


(a) 前 6 种算法

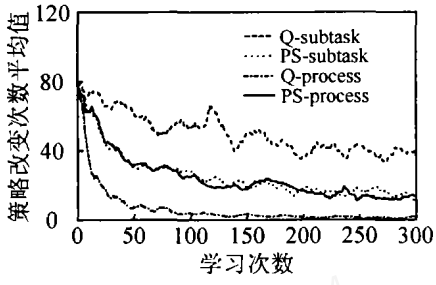


(b) 后 4 种算法

图 2 系统性能比较



(a) 前 4 种算法



(b) 后 4 种算法

图 3 学习速度比较

并不稳定, 这是由于每次实验中动态目标物移动的随机性造成的.

4 结 论

本文提出了一种基于过程奖赏和优先扫除的强化学习算法作为机器人之间的冲突消解策略, 利用两类仿真环境(静态环境和动态环境)的多机器人可识别群体觅食任务为比较手段, 以收集的目标物数量为系统性能指标, 以算法收敛时学习次数为学习速度指标, 进行实验研究, 并将所提出的 PS-process 算法与 Q-global, PS-global, Q-local, PS-local, Q-subtask, PS-subtask, Q-process, Hand-homo 和 Hand-hetero 等 9 种算法进行了比较, 结果表明:

1) 本文所提出的 PS-process 的系统性能最优. 在静态环境中当机器人数量超过 12 个, 在动态环境中当机器人数量超过 10 个时, 系统性能虽然也有一定的下降, 但其下降速度很慢并且系统性能仍远高于其他算法. 这说明 PS-process 能有效地降低机器人之间的冲突, 提高系统性能.

2) 相对于全局奖赏、局部奖赏和子任务方法, 基于过程奖赏的算法学习速度最快, 基于全局奖赏的学习速度最慢. 这也说明全局奖赏并不适合于多机器人可识别群体觅食任务.

参考文献(References)

[1] Asama H, Arai T, Fukuda T, et al. Distributed Autonomous Robotic Systems[M]. Tokyo: Springer-Verlag, 1994: 50-61.

(下转第 439 页)

Based Supercritical Power Plant Controller [A]. *Proc of 35th IEEE Conf on Decision and Control* [C]. Kobe, 1996: 4486-4491.

[4] 吕志来, 张保会. 基于ANN 和模糊控制相结合的电力负荷短期预测方法[J]. *电力系统自动化*, 1999, 23(22): 37-39.
(Lv Z L, Zhang B H. Short Term Lord Forecasting Method Based on Combination of ANN and Fuzzy Control[J]. *Automation of Electric Power System*, 1999, 23(22): 37-39.)

[5] Chen Y Q, Liu J Z, Zeng D L, et al. Design of a Self-learning Fuzzy-neural Networks and Application to a Boiler-turbing Coordinated Control System [A]. *Int Conf on Electrical Engineering* [C]. Jeju Island, 2002: 283-287.

[6] Wen T. $H\infty$Control for a Boiler-turbine Unit[A]. *Proc of the* 1999 *IEEE, Int Conf on Control Applications* [C]. Hawaii, 1999: 910-914.

[7] 柳洪义, 马现刚, 朱树森. 微波催化连续反应实验系统的温度控制[J]. *东北大学学报*, 2003, 24(3): 256-259.
(Liu H Y, Ma X G, Zhu S S. Temperature Control of Microwave Catalysis Continuous Reaction Experiment System [J]. *J of Northeastern University*, 2003, 24(3): 256-259.)

[8] Wu Z Q, Mizumoto M. PID-type Fuzzy Controller and Parameters Adaptive Method [J]. *Fuzzy Set and System*, 1996, 78(1): 23-25.

[9] 魏毅新, 王新春. 燃煤锅炉蒸汽压力的模糊控制[J]. *包头钢铁学院学报*, 2002, 21(2): 162-164.
(Wei Y X, Wang X C. The Fuzzy Control on Vapor Pressure of Coal-burning Boiler [J]. *J of Baotou University of Iron and Steel Technology*, 2002, 21(2): 162-164.)

[10] 施海平, 吴征, 吴永海. 模糊控制技术在国产 200 MW 机组协调控制系统上的应用[J]. *中国电力*, 1999, 32(3): 47-53.
(Shi H P, Wu Z, Wu Y H. Application of Fuzzy Control Technology in Coordinated Control System of Indigenous 200 MW Generating Units [J]. *Electric Power of China*, 1999, 32(3): 47-53.)

[11] 陈彦桥, 王印松, 刘吉臻, 等. 基于 PID 型模糊神经网络的火电站单元机组协调控制[J]. *动力工程*, 2003, 23(1): 2219-2223.
(Chen Y Q, Wang Y S, Liu J Z, et al. The Boiler-turbine Coordinated Control Based on the PID-type Fuzzy Neural Network in the Fossil-fired Power Station [J]. *Power Engineering*, 2003, 23(1): 2219-2223.)

(上接第 434 页)

[2] 任淼, 陈宗海. 环境因素对多自主移动机器人系统的影响研究[J]. *计算机工程与应用*, 2005, 41(22): 61-63.
(Ren Y, Chen Z H. Study on Effect of Environmental Factors on Multiple Autonomous Robot Systems [J]. *Computer Engineering and Applications*, 2005, 41(22): 61-63.)

[3] Balch T R. *Behavior Diversity in Learning Robot Teams* [D]. Atlanta: Georgia Institute of Technology, 1998.

[4] 王坤, 陈卫东. 分布式多移动机器人系统中基于局部感知的排队协调策略研究[J]. *机器人*, 2002, 24(6): 540-544.
(Wang K, Chen W D. Queue Coordination Strategy Based on Local Sensing in Distributed Multiple Mobile Robot Systems[J]. *Robot*, 2002, 24(6): 540-544.)

[5] 陈卫东, 李振海, 席裕庚. 分布式多移动机器人系统冲突及其消解策略的实例研究[J]. *系统仿真学报*, 2002, 14(10): 1288-1301.
(Chen W D, Li Z H, Xi Y G. Interference and Its Solving Strategy in Distributed Multiple Autonomous Robot System: A Case Study [J]. *J of System Simulation*, 2002, 14(10): 1288-1301.)

[6] 张汝波, 顾国昌, 刘兆德等. 强化学习理论算法及应用[J]. *控制理论与应用*, 2000, 17(5): 637-642.
(Zhang R B, Gu G C, Liu Z D, et al. Reinforcement Learning Theory, Algorithms and Its Application [J]. *Control Theory and Applications*, 2000, 17(5): 637-642.)

[7] Moore A W, Atkeson C G. Prioritized Sweeping: Reinforcement Learning with Less Data and Less Real Time[J]. *Machine Learning*, 1993, 13(1): 103-130.

[8] Logan B, Theodoropoulos G. The Distributed Simulation of Multiagent Systems [J]. *Proc of the IEEE*, 2001, 89(2): 174-185.

[9] Mahadevan S, Connel J. Automatic Programming of Behavior-based Robots Using Reinforcement Learning [J]. *Artificial Intelligence*, 1992, 55(2-3): 311-365.