基于多 Agent 强化学习的多站点 CSPS 系统的协作 Look-ahead 控制

唐 昊 [1,2] 万海峰 [1] 韩江洪 [1,2] 周 雷 [1]

摘 要 研究多站点传送带给料生产加工站 (Conveyor-serviced production station, CSPS) 系统的最优控制问题, 其优化目标是通过合理选择每个 CSPS 的 Look-ahead 控制策略, 实现整个系统的工件处理率最大. 本文首先根据多 Agent 系统的反应扩散思想, 对每个 Agent 的原始性能函数进行改进, 引入了具有扩散功能的局域信息交互项 (原始项看作具有反应功能); 并运用性能势理论, 构建一种适用于平均和折扣两种性能准则的 Wolf-PHC 多 Agent 学习算法, 以求解决策时刻不同步的多站点的协作 Look-ahead 控制策略. 最后, 论文通过仿真实验验证了该算法的有效性, 学习结果表明, 通过性能函数的改进, 各工作站的负载平衡性得到改善, 整个系统的工件处理率也明显提高.

关键词 传送带给料生产加工站, Look-ahead 控制, 多 Agent 强化学习, 性能函数

DOI 10.3724/SP.J.1004.2010.00289

Coordinated Look-ahead Control of Multiple CSPS System by Multi-agent Reinforcement Learning

TANG Hao[1,2] WAN Hai-Feng[1] HAN Jiang-Hong[1,2] ZHOU Lei[1]

Abstract The optimal control problem of a multiple conveyor-serviced production station (CSPS) system is concerned. The objective is to maximize the part-processing rate of the entire system by choosing a suitable look-ahead control strategy for each CSPS. According to the reaction-diffusion mechanism of multi-agent systems, the original performance function of each agent is first modified by introducing an item with a diffusion function that denotes the interaction of local information (The original item is assumed to have a reaction function). Then, combined with the concept of performance potentials, a multi-agent algorithm, i.e., Wolf-PHC algorithm, is proposed to derive the coordinated look-ahead control strategy for systems with either discounted or average performance criteria, where the decision epoch of each agent is asynchronous. Finally, a simulation example is used to illustrate the effectiveness of the algorithm, and the simulation results show that due to the modification of the performance functions, the contributions of all the stations are well balanced, and the part-processing rate of the entire system is increased significantly.

Key words Conveyor-serviced production station (CSPS), look-ahead control, multi-agent reinforcement learning, performance function

在现实中的一些生产加工企业中, 存在一类由生产加工站作为加工主体的生产线, 例如先进制造业中的一些机器人装配线 (Robotic assembly line), 其中, 加工站由传送带输送工件进行加工, 这样的一类系统称为传送带给料生产加工站 (Conveyor-serviced production station, CSPS)[1−5]. 另外, 由于专业化、规模化和集约化生产的需要, 有些生产线往往配有多个 CSPS, 称为多站点 CSPS[3−5], 如图 1 所示. 这里, 若干站点依次串行分布在传送带一旁, 每个站点配有两个库, 一个用于存放从传送带上卸载下来的未加工工件, 称之为缓冲库 (Buffer). 另一个用于存放已加工工件, 称之为 Bank. 另外, 每个站点配有一个前视 (Look-ahead) 传感器, 如红外、雷达或摄像头等, 可感知或测定传送带上一定距离内是否有工件以及工件的位置信息.

这里, 前视距离为控制变量, 每个站点的工作过程是: 在当前决策时刻, 若前视距离内有工件, 则等待工件到达本站点并捡取放入缓冲库, 然后转入下一决策时刻; 否则, 直接从缓冲库中取出一个工件进行加工, 加工完毕后放入 Bank 中, 转入下一决策时刻. 系统中, 工件到达时间和加工时间是随机的, 其

收稿日期 2008-12-18 录用日期 2009-05-26
Manuscript received December 18, 2008; accepted May 26, 2009
国家自然科学基金项目 (60873003), 教育部留学回国人员科研启动基金, 安徽省自然科学基金 (090412046), 安徽高校省级自然科学研究重点项目 (KJ2008A058) 资助
Supported by National Natural Science Foundation of China (60873003), Scientific Research Foundation for Returned Scholars, State Ministry of Education of China, Nature Science Foundation of Anhui Province (090412046), and Nature Science Foundation of Education Department of Anhui Province (KJ2008A058)
1. 合肥工业大学计算机与信息学院 合肥 230009 2. 安全关键工业测控技术教育部工程研究中心 合肥 230009
1. School of Computer and Information, Hefei University of Technology, Hefei 230009 2. Engineering Research Center of Safety Critical Industry Measure and Control Technology, Ministry of Education, Hefei 230009

优化目标是求解每个站点的 Look-ahead 控制策略, 以协调各个 Agent 的工作, 使系统的代价或工件处理率达到最优.

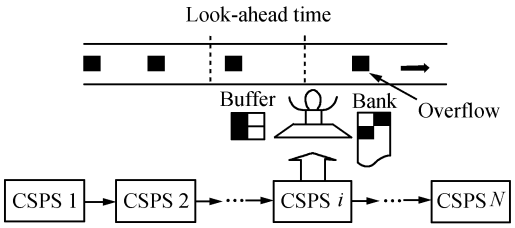


图 1 多站点 CSPS 物理模型

Fig. 1 Physical model of multiple CSPS system

日本电气通信大学的松井正之教授建立了单站点 CSPS 的半 Markov 决策过程 (Semi-Markov decision process, SMDP) 模型, 给出了求解一些性能值 (如单位生产周期的平均等待时间) 的理论计算方程[1], 并于最近研究了平均准则下的各种控制模式和系统的物理原理[2]. 文献 [4−5] 将多站点 CSPS 系统视为一个整体, 运用遗传算法及其改进算法求解了系统最优 Look-ahead 控制策略. 但是, 数值求解方法存在 "建模难" (Curse of modeling) 问题: 一方面, 模型参数的完全知识难以获取; 另一方面, 即便模型参数已知, 转移概率矩阵和性能函数矩阵的建立表达及等价处理也非常困难. 例如, 各种不同类型的代价 (如等待、库存和加工等代价), 其折扣累积的时间起点和终点可能不一样, 难以预先把决策点之间的累积代价的期望值等效为决策点的期望即时代价. 为克服这一难题, 可运用基于仿真技术的模型无关 (Model-free) 优化方法来解决 CSPS 的优化控制问题. 最近, 文献 [6] 针对单站点 CSPS 的 Look-ahead 控制, 基于性能势理论和 SMDP 数学模型, 建立了一种适用于平均或折扣准则的模型无关在线策略迭代学习优化算法. 针对 CSPS 的类似系统 — 多机器人自动化生产线的协调作业问题, 文献 [7] 运用 CMAC 网络逼近, 建立了统一的模型无关参数 Q 学习算法, 但这些文献并没有考虑多 Agent 学习算法. 因此, 在前述工作基础上, 本文将针对多站点 CSPS 的协作 Look-ahead 控制问题, 把每个加工站点视为一个自主的 Agent 学习体, 并运用性能势理论, 研究一种适用于平均和折扣两种性能准则的模型无关多 Agent 学习算法, 即 Wolf-PHC (Wolf policy hill-climbing) 算法, 解决系统的异步决策问题.

1 数学模型

假设系统中传送带匀速运行, 站点个数为 N, 且每个站点的配置和功能相同, 即每个站点的缓冲库容量均为 M, Bank 的容量均为无穷大. 记站点 i 的缓冲库空余量 s_i 作为 Agent i 自身的状态, 有 $0 \le s_i \le M$. 整个系统的状态 s 由各个 Agent 自身状态组合而成, 即 $s = \{s_1, s_2 \cdots, s_N\}$. 设每个站点的前视距离为控制变量 (或称行动), 只与自身状态 s_i 有关, 记为 $a_i(s_i)$. 由于传送带速度恒定, 故将前视距离 $a_i(s_i)$ 或其他距离都等效成时间来表示, 并假设 $a_i(s_i) \in [0, l]$, 这里 l 为最大前视距离 (时间). 于是, Agent i 的策略 v_i 定义为 $v_i = (a_i(0), a_i(1), \cdots, a_i(M))$.

现设 T_n 为第 n 个决策时刻 ($T_0 = 0$), 系统状态记为 $s(T_n)$, 且由 Agent i 进行决策, 其自身状态为 $s_i(T_n)$. 在策略 v_i 下, 设 Agent i 选择行动 $a_i(s_i(T_n))$. 如果在 $a_i(s_i(T_n))$ 范围内有工件且第一个工件离捡取点的位置为 $\theta_i(T_n)$, 则 Agent i 等待 $\theta_i(T_n)$ 后捡取工件并将其放入缓冲库, 且下一个决策时刻为 $T_{n+1} = T_n + \theta_i(T_n)$, Agent i 自身状态转移到 $s_i(T_n) - 1$; 如果在 $a_i(s_i(T_n))$ 范围内没有工件, 则 Agent i 直接从缓冲库中取出一个工件进行加工, 设加工过程不可中断, 加工时间 $\mu_i(T_n)$ 服从一般分布. 工件加工完后放入 Bank 中, 下一个决策时刻 $T_{n+1} = T_n + \max\{a_i(s_i(T_n)), \mu_i(T_n)\}$, 且 Agent i 的自身状态转移到 $s_i(T_n) + 1$. 当缓冲库已满时 ($s_i(T_n) = 0$), 无需前视, 令 $a_i(s_i(T_n)) = 0$; 当缓冲库为空时 ($s_i(T_n) = M$), 站点将一直等待直到有工件到达, 即相当于前视距离为无穷大, 即 $a_i(s_i(T_n)) = \infty$.

多站点系统的优化目标就是合理选择每个站点的 Look-ahead 控制策略, 使其协调工作, 并实现系统在无穷时段内的工件处理率最大. 故首先定义 T 时段内系统的工件处理率等于 T 时段内各个 Agent 加工工件总数 $P(T)$ 与 T 时段内工件到达总数 $Q(T)$ 的比值. 因此, 无穷时段内:

$$\text{系统工件处理率} = \lim_{T\to\infty} \frac{P(T)}{Q(T)}$$

各个站点的负载平衡性, 即各工作站的出力平衡性, 也是考察系统性能的重要指标, 且为影响系统工件处理率提高的一个关键要素. 故定义 T 时段内 Agent i 的负载率为 T 时段内 Agent i 的加工工件总数与 T 时段内各个 Agent 加工工件总数的比值. 因此, 无穷时段内:

$$\text{Agent } i \text{ 的负载率} = \lim_{T\to\infty} \frac{P_i(T)}{P(T)}$$

2 代价函数定义

一般假设工件按照参数为 λ 的 Poisson 流到达加工系统, 每个站点加工时间服从一般的随机分布. 对单站点 CSPS 系统, 其最优 Look-ahead 控制问

题可建模为 SMDP 来研究[6]; 对多站点 CSPS 系统, 若把所有站点当成一个 Agent, 采取集中控制, 则其最优 Look-ahead 控制问题也可建模为 SMDP, 但是这种处理方式将导致 "维数灾". 因此, 在多站点系统中, 我们将根据分布式自治系统的特点, 运用各 Agent 之间的信息交互, 来实现整个系统的学习优化控制. 此时, 每个站点都是一个自主的学习体.

首先考虑单站点 CSPS. 当系统处于稳态运行时, 在一段较长的时间范围内, 工件到达的期望数一定. 且每个工件加工的平均时间也趋于期望值. 因此, Agent 在单位时间内的等待时间越短, 意味单位时间内的加工时间就越长, 加工个数越多, 则系统的处理率便会越高. 为此, 定义 K_1 为单位等待时间的代价, 且记 $f(s(T_n), a(s(T_n)), s(T_{n+1}))$ 为 Agent 在状态 $s(T_n)$ 下, 采取行动 $a(s(T_n))$, 转移到下一状态 $s(T_{n+1})$ 前的单位时间代价. 于是, 当 $s(T_{n+1}) = s(T_n) - 1$ 时

$$f(s(T_n), a(s(T_n)), s(T_{n+1})) = K_1 \tag{1}$$

否则

$$f(s(T_n), a(s(T_n)), s(T_{n+1})) = \begin{cases} 0, & T_n \le t \le T_n + \mu(T_n) \\ K_1, & T_n + \mu(T_n) < t \le T_{n+1} \end{cases} \tag{2}$$

其中, $\mu(T_n)$ 为 T_n 到 T_{n+1} 间的实际加工时间.

根据文献 [8], 可定义 Agent 在策略 v 下的无穷时段折扣代价准则 $\eta_\alpha^v(s)$ 和平均代价准则 η^v, 其中 $\alpha > 0$ 是折扣因子. 学习优化的目标就是在策略空间中找到一个最优策略, 使得代价准则值最小. 按式 (1) 和 (2) 定义代价函数, 并按此折扣或平均准则进行优化, 就等同于优化系统的工件处理率.

类似于单站点系统, 在多站点中, 针对 Agent i, 把式 (1) 和式 (2) 中的 $s(T_n)$、$a(s(T_n))$ 和 $\mu(T_n)$ 分别换成 $s_i(T_n)$、$a_i(s(T_n))$ 和 $\mu_i(T_n)$, 则得到 Agent i 的单位时间折扣代价函数. 于是, Agent i 可根据此代价函数与自身状态信息构建独立的学习算法和决策机制. 由于站点的串行分布特点, 前面 (上游) 站点具有优先捡取工件的机会, 其决策对后面 (下游) 所有站点的运行都将产生影响. 但是, 在没有信息交互的情况下, 下游站点的决策对上游站点却不产生影响, 因而不利于站点间的协作. 因此, 上游站点总是比下游站点的学习收敛速度快, 且下游站点的学习变化无法反馈到上游. 这种协作的缺乏容易造成传送带上游站点的负荷过重, 而下游站点负荷较轻, 不利于站点间的负载平衡, 进而影响系统处理率的提高.

反应扩散方程是刻画多 Agent 系统动态特性的一个重要方法[9], 其主要思想是定义系统的能量函数与每个 Agent 的模式及其与所有相邻 Agent 的模式之差有关, 前者是反应项, 后者是扩散项, 系统的动力学过程就是能量函数的动态衰减过程. 在本模型中, 为体现各站点, 特别是相邻站点之间的协作关系, 可运用反应扩散思想, 将 Agent i ($i \ne N$) 与其下游相邻的 Agent 进行信息交互, 即将后者的状态信息通过代价函数反馈到前者作为影响其决策的参考因素. 具体地, 在站点 i 的代价函数定义中添加一个反馈项 (局部信息交互项), 将其下游紧邻站点的缓冲库库存空余量与本站点的库存空余量之差作为本站点的代价一部分. 于是, 对每个 Agent i ($i \ne N$) 的原始性能函数式 (1) 和 (2) 进行改进, 即当 $s_i(T_{n+1}) = s_i(T_n) - 1$ 时, 定义

$$f^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1})) = K_1 + K_2(s_{i+1} - s_i) \tag{3}$$

否则

$$f^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1})) = \begin{cases} 0, & T_n \le t \le T_n + \mu_i(T_n) \\ K_1 + K_2(s_{i+1} - s_i), & T_n + \mu_i(T_n) < t \le T_{n+1} \end{cases} \tag{4}$$

这里, K_2 表示相邻两 Agent 之间的单位缓冲库空余量差值的单位时间反馈代价. 这样, Agent $i+1$ 的库存信息可传递到 Agent i, 通过逐级信息反馈, 最终可提高系统中多个站点间的相互协作能力, 改善学习优化性能, 达到系统优化的目标. 上述两式中, 原始项 K_1 可看作具有反应功能, 而反馈项 $K_2(s_{i+1} - s_i)$ 可看作具有扩散功能的局域信息交互项. 对每个 Agent i, 其代价率与其下游紧邻的 Agent 有关, 故其性能函数与整个系统状态 s 有关, 如式 (3) 和 (4) 所示. 为区别起见, 利用式 (1) 和 (2) 进行的学习称为非反馈式学习, 按式 (3) 和 (4) 进行的学习称为反馈式学习.

3　基于性能势的多 Agent Q 学习算法

根据第 2 节, 各 Agent 的决策时刻为每次工件捡取之后或为一个工件加工完毕时, 在任意时刻, 两个或两个以上 Agent 需同时决策的概率为 0, 即多个 Agent 的决策时刻不同步, 为异步决策模式. 常用的多 Agent 算法, 如 Nash-Q 等, 一般采用同步决策模式, 难以应用到本文考虑的模型中. 因此, 在本文研究的多站点 CSPS 系统中, 需考虑异步决策学习算法. 首先, 根据多 Agent 分组学习思想[10], 在某个决策时刻, 可将系统中全部 N 个 Agent 视为两组, 其中, 不需要决策的 $N-1$ 个站点为第一组, 该组中所有 Agent 的策略保持不变; 另一个需要决

策的 Agent 为另一组, 其学习的原则是寻找一个与第一组所有 Agent 的策略互补的最优策略[10]. 具体地, 在某个决策时刻, 需要决策的 Agent 将更新其 Q 值, 其他 Agent 则不需要更新 Q 值.

Wolf-PHC 算法是一种较为常用的多 Agent 强化学习算法, 是 PHC 的一种改进算法, 其 Q 值更新过程不直接依赖其他 Agent 的信息[11], 因此可用来解决上述分组多 Agent 系统的异步决策问题, 并结合反应扩散思想, 通过站点间的逐级信息反馈, 来提高系统的协作能力, 改善学习优化性能. 学习中, 每个 Agent 采用混合策略 (Mixed policy) 且只保存自身的 Q 值表[12], 所以, 一方面, 它避免了一般 Q 学习中需要解决的探索和利用 (Explore vs. exploit) 这一矛盾问题[13]; 另一方面, 它可解决多 Agent 系统的异步决策问题. 另外, 与 Nash-Q 等算法相比, Wolf-PHC 只需保存较少的信息, 可降低多 Agent 问题求解的空间复杂度. 算法中, Agent i 的随机策略只与自身状态有关, 记为 π_i. 初始时每个状态下行动的选择概率都相同, 即对 $\forall s_i, a_i(s_i)$, $\pi_i(s_i, a_i(s_i)) = 1/\mid A_i \mid$, 这里, A_i 为 Agent i 的行动集, $\mid A_i \mid$ 表示 Agent i 的行动个数. 第 n 步学习中, 若是 Agent i 作决策, 其策略更新公式为

$$\pi_i(s_i(T_n), a_i) = \pi_i(s_i(T_n), a_i) + \begin{cases} \delta_i, & a_i = \arg\max\limits_{a \in A_i} Q_i(s_i(T_n), a) \\ \dfrac{-\delta_i}{\mid A_i \mid -1}, & \text{否则} \end{cases} \tag{5}$$

其中, δ_i 为 Agent i 的学习增量, 一般随学习推进呈下降趋势; $Q_i(s_i(T_n), a)$ 表示 Agent i 自身状态–行动对 $(s_i(T_n), a)$ 的 Q 值. 在 Wolf-PHC 算法中, 策略更新使用更理性的方法[12], 即满足 "输快赢慢" (Learn quickly while losing and slowly while winning) 原则. 结合 Q 值表, 可判断学习的 "输" 和 "赢", 并决定 δ_i 的取值, 即

$$\delta_i = \begin{cases} \delta_{\text{lose}}, & \text{若 "输"} \\ \delta_{\text{win}}, & \text{否则} \end{cases} \tag{6}$$

其中, 当 $\sum_{a_i \in A_i} \pi_i(s_i(T_n), a_i) Q_i(s_i(T_n), a_i) > \sum_{a_i \in A_i} \overline{\pi}_i(s_i(T_n), a_i) Q_i(s_i(T_n), a_i)$ 时为 "输" (Lose). 一般情况下, δ_{lose} 比 δ_{win} 大若干倍, 且 δ_{lose} 和 δ_{win} 随着学习的进行, 逐步衰减. 上式中, $\bar{\pi}_i(s, a)$ 称为平均策略 (Average policy), 其更新公式为

$$\bar{\pi}_i(s_i(T_n), a_i) = \bar{\pi}_i(s_i(T_n), a_i) + \frac{\pi_i(s_i(T_n), a_i) - \bar{\pi}_i(s_i(T_n), a_i)}{c_i(s_i(T_n))} \tag{7}$$

其中, $c_i(s_i(T_n))$ 为 Agent i 访问状态 $s_i(T_n)$ 次数.

Q 学习是一种模型无关 (Model-free) 的学习方法, 它通过仿真或观测系统的运行, 不断学习逼近状态–行动对 $(s_i, a_i(s_i))$ 的函数值进行问题的求解. 学习时, Agent i 的一个观测样本记为一 5 元组 $\langle s(T_n), a_i(s_i(T_n)), s(T_{n+1}), \omega_i(T_n), \mu_i(T_n) \rangle$, 其中, $\omega_i(T_n) = T_{n+1} - T_n$ 表示状态 $s(T_n)$ 的逗留时间. 据此可计算 T_n 到 T_{n+1} 间的累积代价 $f_\alpha^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1}))$. 当 $s_i(T_{n+1}) = s_i(T_n) - 1$ 时

$$f_\alpha^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1})) = K_1 T_\alpha(\omega_i(T_n)) + K_2(s_{i+1} - s_i) T_\alpha(\omega_i(T_n)) \tag{8}$$

否则

$$f_\alpha^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1})) = K_1(T_\alpha(\omega_i(T_n)) - T_\alpha(\mu_i(T_n))) + K_2(s_{i+1} - s_i) T_\alpha(\omega_i(T_n)) \tag{9}$$

其中, 对 $\forall \omega > 0$,

$$T_\alpha(\omega) = \int_0^\omega \mathrm{e}^{-\alpha t} \mathrm{d}t = \frac{1 - \mathrm{e}^{-\alpha\omega}}{\alpha}$$

$T_\alpha(\omega)$ 在 $\alpha \to 0$ 时的极限为 $T_0(\omega) = \omega$. 由式 (8) 和 (9), 以及性能势的 Q 学习理论[6], 可获得 Agent i 在平均和折扣性能准则下统一的即时差分公式

$$d_i(T_n) = f_\alpha^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1})) - T_\alpha(w_i(T_n))\bar{\eta}^i + \mathrm{e}^{-\alpha w_i(T_n)} \min_{a_i \in A_i} Q_i(s_i(T_{n+1}), a_i) - Q_i(s_i(T_n), a_i(s_i(T_n))) \tag{10}$$

其中, $\bar{\eta}^i$ 表示 Agent i 的平均代价的学习值. 根据文献 [6], 其值可通过下列估计 S_ω^i 和 S_f^i 来计算

$$S_f^i = S_f^i + \zeta^i(f_{\alpha=0}^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1})) - S_f^i) \tag{11}$$

$$S_\omega^i = S_\omega^i + \zeta^i(\omega_i(T_n) - S_\omega^i) \tag{12}$$

$$\bar{\eta}^i = \frac{S_f^i}{S_\omega^i} \tag{13}$$

式 (11) 中, $f_{\alpha=0}^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1}))$ 表示 $f_\alpha^i(s(T_n), a_i(s_i(T_n)), s(T_{n+1}))$ 在 $\alpha \to 0$ 时的极限, 即 T_n 到 T_{n+1} 时间段的实际累积代价, ζ^i 是一学习步长. 若令 m_i 是 Agent i 的学习步数, 且 $\zeta^i = 1/m_i$, 则 S_f^i 和 S_ω^i 分别为每步平均代价和每步平均逗留时间.

于是, 得到 Agent i 的 Q 值学习公式

$$\begin{aligned}&Q_i(s_i(T_n),a_i(s_i(T_n))) = \\&\quad Q_i(s_i(T_n),a_i(s_i(T_n))) + \\&\quad \gamma_i(s_i(T_n),a_i(s_i(T_n)))d_i(T_n)\end{aligned} \tag{14}$$

其中, $\gamma_i(s_i(T_n),a_i(s_i(T_n)))$ 为 Agent i 的 Q 值学习步长, 一般比 ζ^i 衰减慢, 可取 $\gamma_i(s_i(T_n),a_i(s_i(T_n))) = 1/c_i(s_i(T_n),a_i(s_i(T_n)))^{\beta}$. 这里, $c_i(s_i(T_n),a_i(s_i(T_n)))$ 为 Agent i 访问状态–行动对 $(s_i(T_n),a_i(s_i(T_n)))$ 次数, β 为常数, 且 $0<\beta<1$.

Agent i 的具体算法过程如下:

步骤 1. 令所有状态–行动对的 Q 值为零; 初始化随机策略 π_i 及其他控制条件; 令 $n=0$.

步骤 2. 在决策时刻 T_n 时, 根据混合策略 π_i 选择状态 $s_i(T_n)$ 下的行动 $a_i(s_i(T_n))$.

步骤 3. 执行行动 $a_i(s_i(T_n))$, 并记录样本观测数据 $\langle s(T_n),a_i(s_i(T_n)),s(T_{n+1}),\omega_i(T_n),\mu_i(T_n)\rangle$, 由式 (8) 或 (9) 计算 $f_\alpha^i(s(T_n),a_i(s_i(T_n)),s(T_{n+1}))$.

步骤 4. 根据式 (10)∼(13) 计算 $\bar{\eta}^i$ 和 $d_i(T_n)$, 再根据式 (14) 更新 $Q_i(s_i(T_n),a_i(s_i(T_n)))$.

步骤 5. 根据式 (7) 更新平均策略 $\bar{\pi}_i$ 在状态 $s_i(T_n)$ 时的行动选择概率; 根据式 (5) 更新随机策略 π_i 在状态 $s_i(T_n)$ 时的行动选择概率.

步骤 6. 若算法终止条件满足, 学习结束; 否则, $n=n+1$, 转步骤 2.

由式 (5) 和式 (7), 若令

$$\delta_{\text{win}} = \min\left\{\min_{a_i\in A_i}\frac{\pi_i(s_i(T_n),a_i)}{10},\ \frac{1}{c_i(s_i(T_n))}\right\}$$

$$\delta_{\text{lose}} = \min\left\{\min_{a_i\in A_i}\frac{\pi_i(s_i(T_n),a_i)}{5},\ \frac{2}{c_i(s_i(T_n))}\right\}$$

则 δ_{win} 和 δ_{lose} 将随着学习步数的增大逐步减小, 并且混合策略 π_i 的行动选择概率将不会出现负值.

4　实验仿真

设站点的个数 $N=4$, 并且等间距分布. 由于传送带匀速运行, 故将系统中的距离长度计量值转换为时间长度计量值. 具体仿真参数设定如下:

1) 相邻站点的间距: 5 s;

2) 最大前视距离: $l=4$ s;

3) 缓冲库容量: $M=4$;

4) 工件加工时间: 服从 $K=4$, $\bar{\mu}=4$, 总服务率 $\mu=\bar{\mu}/K=1.0$ 的 Erlang 分布, 其密度函数为 $f(t)=\bar{\mu}(\bar{\mu}t)^{K-1}\mathrm{e}^{-\bar{\mu}t}/(K-1)!$;

5) 单位时间等待代价系数: $K_1=2.0$;

6) 单位缓冲库空余量差值的单位时间反馈代价系数: $K_2=0.5$;

7) Agent 学习步长的指数因子: $\beta=0.6$.

仿真中, 将行动区间 [0 4] 等间距离散化为有限行动集 $\{0,0.8,1.6,2.4,3.2,4.0\}$, 分别用 $0,1,2,\cdots,5$ 整数表示其中 6 个行动, 因此每个 Agent 需要保存的状态–行动对的个数小于 $5\times6=30$. 学习初始时, 令每个 Agent 在任意状态下选择任意可行行动的概率都相等, 学习中每 2 000 步对学习效果进行一次测试, 即进行 Monte-Carlo 策略评估. 具体方法是, 根据各 Agent i 的当前随机策略 π_i 仿真系统运行 20 万步, 共进行 30 次独立实验, 然后估计当前策略作用下的相关性能值.

图 2 是 $\lambda=0.8$ 时, 反馈式学习与非反馈式学习的处理率优化曲线图, 可见两种学习方法都能有效提高整个系统的工件处理率, 但是反馈式学习由于引入了邻近站点间的库存余量差值作为多 Agent 系统学习的交互扩散项, 其学习优化效果要明显优于非反馈式学习. 图 3 为对应仿真中, 两种学习方式下每个 Agent i 在终止策略 π_i 下的负载率, 可见反馈式学习算法的各个 Agent 的出力平衡性也明显优于非反馈式学习. 这说明, Agent 之间通过局域信息交互, 不仅可以提高系统的总处理率, 还可以改善各个 Agent 的负载平衡性. 图 4 为对应图 2 仿真的系统总平均代价优化曲线, 其变化规律与图 2 的两条曲线正好反相对应; 另外, 图 5 是每个 Agent 在反馈式学习时的平均代价优化曲线.

图 6 和图 7 分别表示系统在对应图 2 的反馈式学习中, Agent 1 和 Agent 3 在状态 $\bar{s}=(3,2,2,1)$ 时的行动选择概率变化曲线. 两图说明系统状态在充分访问后, 其行动选择概率趋于收敛. 另外, 我们对两种学习方法所获得的最终策略分别进行策略评估, 得到两组不同的工件处理率估计值, 据此计算, 得到非反馈式学习的处理率样本方差为 0.6537, 而反馈式学习的处理率样本方差为 0.5575, 说明反馈式学习得到的终止策略的平稳性要好于非反馈式学习得到的终止策略.

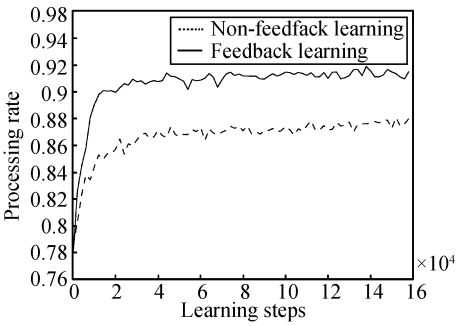


图 2　反馈式和非反馈式学习的总处理率

Fig. 2　Total processing rates for non-feedback learning and feedback learning

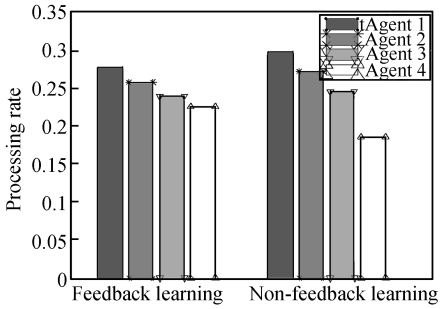


图 3 两种学习方法对应的各个 Agent 的负载率

Fig. 3 Contribution of each agent for the two learning methods

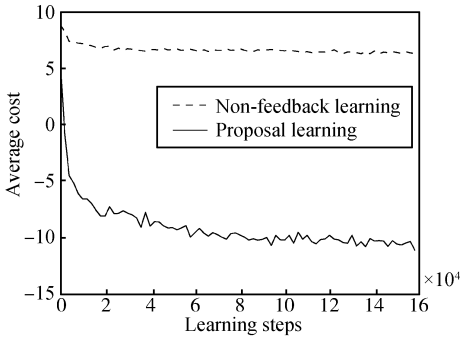


图 4 两种学习方法对应的总平均代价

Fig. 4 Total average costs for the two learning methods

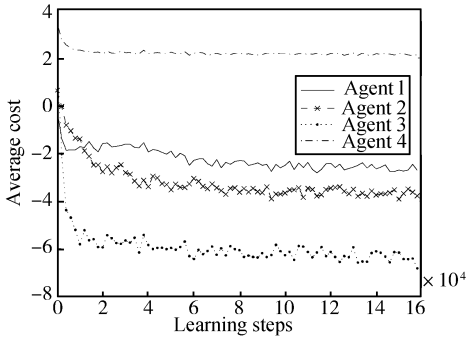


图 5 反馈式学习下各个 Agent 的平均代价

Fig. 5 Average cost of each agent via feedback learning

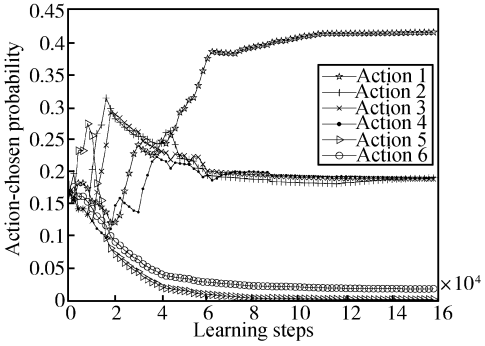


图 6 反馈式学习时 Agent 1 在状态 $\bar{s}$ 下的行动选择概率

Fig. 6 Action-chosen probabilities of Agent 1 at $\bar{s}$ via feedback learning

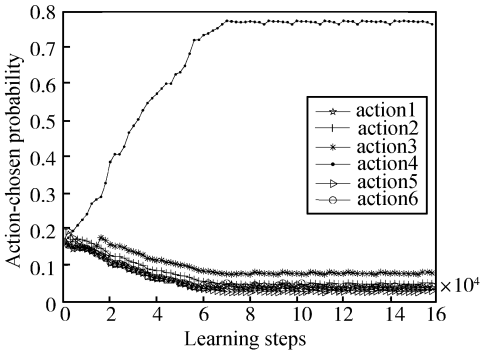


图 7 反馈式学习时 Agent 3 在状态 $\bar{s}$ 下的行动选择概率

Fig. 7 Action-chosen probabilities of Agent 3 at $\bar{s}$ via feedback learning

表 1 给出了系统在不同工件到达率下, 两种学习方法的工件处理率和各个 Agent 的负载率. 通过数据对比说明了反馈式学习方法的优越性. 当 λ 较大时, 每个 Agent 基本尽力满负荷工作, 反馈机制的引入并不能明显改变各个 Agent 的出力状况, 因此, 表中两种学习方法的对应数据基本接近, 并且对应每种学习方法的各个 Agent 的负载率也基本接近. 但是当 λ 较小时, 反馈式学习在处理率和出力平衡性方面的优势较为明显.

表 1 两种学习方法在不同 λ 下的总处理率和各 Agent 的负载率

Table 1 Total processing rates and the contribution of each agent for the two learning methods under different λ's

λ	学习方式	处理率	Agent 1	Agent 2	Agent 3	Agent 4
0.6	非反馈式学习	0.9429	0.3425	0.2936	0.2298	0.1341
	反馈式学习	0.9753	0.2989	0.2765	0.2154	0.1892
0.7	非反馈式学习	0.9201	0.3137	0.2882	0.2371	0.1610
	反馈式学习	0.9586	0.2909	0.2670	0.2314	0.2106
0.8	非反馈式学习	0.8725	0.2943	0.2767	0.2458	0.1825
	反馈式学习	0.9132	0.2774	0.2576	0.2394	0.2256
1.0	非反馈式学习	0.7612	0.2774	0.2667	0.2449	0.2110
	反馈式学习	0.8007	0.2626	0.2558	0.2465	0.2351
1.2	非反馈式学习	0.6783	0.2667	0.2609	0.2480	0.2244
	反馈式学习	0.6859	0.2614	0.2554	0.2470	0.2362

表 2 给出了系统站点数不同时, 两种学习方法的工件处理率, 其中工件到达率与站点数呈正比变化, 故处理率随 Agent 数目的变化不明显. 表中数据说明对具有不同站点数的 CSPS 系统, 文中研究的反馈式学习算法对非反馈式学习算法仍然具有较大优势.

表 2　站点数与 λ 正比变化时两种学习方法的总处理率

Table 2　Total processing rates for the two learning methods as the number of stations is proportional to λ

Agent 数目 N	λ	非反馈式学习	反馈式学习
3	0.6	0.8407	0.8946
4	0.8	0.8725	0.9132
5	1.0	0.8779	0.9202
7	1.4	0.8760	0.9049

另外, 表 3 还把多 Agent 学习算法与有关固定策略的控制效果进行比较, 其中, 零缓冲策略是指只要缓冲库中有一个工件, Agent 就直接对其进行加工, 即选择行动 0, 这样的策略相当于每个 Agent 每捡取一个工件就加工一个工件, 然后再等待新的工件到达; 经验策略是指每个 Agent 都采取相同的、与状态无关的固定行动, 根据我们的多次实验经验, 每个 Agent 按其在传送带上的前后分布顺序分别取较优行动 1, 2, 3 和 5. 从表 3 可以看出, 反馈式学习无论在处理率上还是在负载平衡性行上都明显好于前两种简单策略.

表 3　反馈式学习与固定策略比较

Table 3　Comparison between feedback learning and fixed policies

	零缓冲策略	经验策略	反馈式学习
处理率	0.7544	0.8474	0.9132
Agent 1 负载率	0.3090	0.2983	0.2774
Agent 2 负载率	0.2761	0.2749	0.2576
Agent 3 负载率	0.2338	0.2501	0.2394
Agent 4 负载率	0.1811	0.1767	0.2256

最后, 考虑折扣准则情况. 图 8 是 $\lambda = 0.8$ 时, 反馈式学习与非反馈式学习分别在折扣因子 $\alpha = 0.1$ 时的处理率优化曲线图, 可见折扣准则下反馈式学习仍然具有比较优势. 图 9 是对应的每个 Agent 在反馈式学习时的折扣代价优化曲线, 其中系统从前述状态 $\bar{s}$ 出发.

5　总结

本文根据多站点传送带给料生产加工站系统的特点, 把多 Agent 系统的反应扩散思想引入到 Wolf-PHC 学习算法中, 可有效解决系统的协作 Look-ahead 控制问题, 这种局域信息交互方法对存储空间的需求与 Agent 个数基本呈线性关系 (每个 Agent 对空间的最大需求保持不变), 避免了基于全域信息的多 Agent 学习算法的组合爆炸问题, 而且与一般的非交互独立学习算法相比, 可获得更好的工件处理率和负载平衡性. 另外, 在本文算法中, 可进一步将相邻 Agent 的缓冲库库存空余量与本站点的库存空余量之差作为 Agent 自身状态定义的一部分, 并定义新的 Look-ahead 控制策略, 构造相关学习算法, 这将是一个有待研究解决的问题. 此外, 考虑在线学习优化时, 如何运用适合小样本学习的机制, 如支持向量机, 来实现多站点 CSPS 系统在连续行动集下的小样本在线学习, 并构造相关神经元动态规划算法[14−16], 也将是一件有意义的工作.

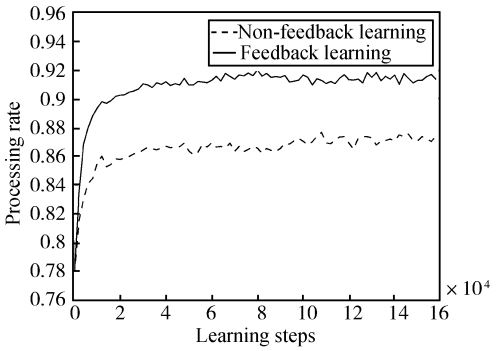


图 8　$\lambda = 0.8, \alpha = 0.1$ 时两种学习方法得到的总处理率

Fig. 8　Total processing rates for the two learning methods as $\lambda = 0.8$ and $\alpha = 0.1$

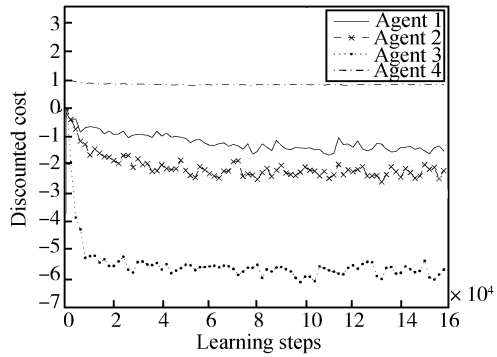


图 9　反馈式学习时每个 Agent 对应状态 $\bar{s}$ 的折扣代价

Fig. 9　Discounted cost of $\bar{s}$ for each agent via feedback learning

致谢

本文作者感谢日本经营工学会会长、日本电气通信大学松井正之教授向其赠送了有关 CSPS 的文献资料, 以及在 CSPS 研究中给予的指导和帮助.

References

1 Matsui M. A generalized model of conveyor-serviced production station (CSPS). *Journal of Japan Industrial Management Association*, 1993, **44**(1): 25−32

2 Matsui M. CSPS model: look-ahead controls and physics. *International Journal of Production Research*, 2005, **43**(10): 2001−2025

3 Yamada T, Satomi K, Matsui M. Strategic selection of assembly system under viable demands. *Assembly Automation*, 2006, **26**(4): 335−342

4 Nakase N, Yamada T, Matsui M. A management design approach to a simple flexible assembly system. *International Journal of Production Economics*, 2002, **76**(3): 281−292

5 Feyzbakhsh S A, Matsui M. Adam-eve-like genetic algorithm: a methodology for optimal design of a simple flexible assembly system. *Computers and Industrial Engineering*, 1999, **36**(2): 233−258

6 Tang H, Arai T. Look-ahead control of conveyor-serviced production station by using potential-based online policy iteration. *International Journal of Control*, 2009, **82**(10): 1917−1928

7 Tang Hao, Ding Li-Jie, Cheng Wen-Juan, Zhou Lei. The cerebellar-model-articulation-controller Q-learning for the task assignment of a handling system. *Journal of Control Theory and Application*, 2009, **26**(8): 884−888
(唐昊, 丁丽洁, 程文娟, 周雷. 搬运系统作业分配问题的小脑模型关节控制器 Q 学习算法. 控制理论与应用, 2009, **26**(8): 884−888)

8 Cao X R. Semi-Markov decision problems and performance sensitivity analysis. *IEEE Transactions on Automatic Control*, 2003, **48**(5): 758−769

9 Yuasa H, Ito M. Self-organizing system theory by use of reaction-diffusion equation on a graph with boundary. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. Tokyo, Japan: IEEE, 1999. 211−216

10 Scherrer B, Charpillet F. Cooperative co-learning: a model-based approach for solving multi agent reinforcement problems. In: Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence. Washington D. C., USA: IEEE, 2002. 463−468

11 Busoniu L, Schutter B D, Babuska R. Learning and Coordination in Dynamic Multiagent Systems, Technical Report 05-019, Delft Center for Systems and Control, Delft University of Technology, The Netherlands, 2005

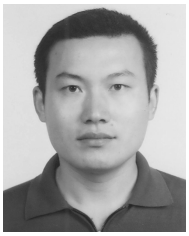
12 Bowling M, Veloso M. Rational and convergent learning in stochastic games. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence. Seattle, USA: Morgan Kaufmann, 2001. 1021−1026

13 Busoniu L, Babuska R, de Schutter B. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2008, **38**(2): 156−172

14 Tang Hao, Yuan Ji-Bin, Lu Yang, Cheng Wen-Juan. Performance potential-based neuro-dynamic programming for SMDPs. *Acta Automatica Sinica*, 2005, **31**(4): 642−645

15 Bertsekas D P, Tsitsiklis J N. *Neuro-Dynamic Programming*. Massachusetts: Athena Scientific, 1996

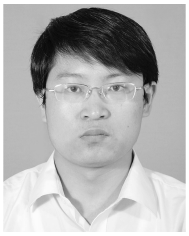
16 Gosavi A. *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Boston: Springer, 2003



唐 昊 合肥工业大学计算机与信息学院教授. 2002 年获得中国科学技术大学博士学位, 2005 年～2007 年于日本东京大学工学部从事博士后研究. 主要研究方向为离散事件动态系统、强化学习和神经元动态规划等智能学习优化、鲁棒决策、分层强化学习、多 Agent 学习. 本文通信作者.
E-mail: htang@hfut.edu.cn
(**TANG Hao** Professor at the School of Computer and Information, Hefei University of Technology. He received his Ph. D. degree in 2002 from University of Science and Technology of China. He was a postdoctoral researcher from 2005 to 2007 at the School of Engineering, University of Tokyo, Japan. His research interest covers discrete event dynamic system (DEDS), intelligent learning optimization by reinforcement learning and neuro-dynamic programming, robust decision, hierarchical reinforcement learning, and multi-agent learning. Corresponding author of this paper.)

万海峰 合肥工业大学硕士研究生. 主要研究方向为离散事件动态系统、多 Agent 强化学习.
E-mail: wanhaifeng1220@gmail.com
(**WAN Hai-Feng** Master student at Hefei University of Technology. His research interest covers discrete event dynamic system (DEDS) and multi-agent reinforcement learning.)

韩江洪 合肥工业大学计算机与信息学院教授. 主要研究方向为恶劣环境下安全关键工业控制、信息系统、分布式网络与分布式系统、先进制造技术、离散事件动态系统. E-mail: hanjh@hfut.edu.cn
(**HAN Jiang-Hong** Professor at the School of Computer and Information, Hefei University of Technology. His research interest covers safety-critical industrial control under bad conditions, information system, distributed network and distributed system, advanced manufacturing technology, and discrete event dynamic system (DEDS).)



周 雷 博士生, 合肥工业大学计算机与信息学院讲师. 主要研究方向为离散事件动态系统、强化学习、无线网络.
E-mail: zhouleizhl@163.com
(**ZHOU Lei** Ph. D. candidate, and lecturer at the School of Computer and Information, Hefei University of Technology. His research interest covers discrete event dynamic system (DEDS), reinforcement learning, and wireless network.)