基于状态聚类的多站点 CSPS 系统的协同控制方法

唐 昊[1,2] 裴 荣[2] 周 雷[2] 谭 琦[1]

摘 要 单站点传送带给料加工站 (Conveyor-serviced production station, CSPS) 系统中, 可运用强化学习对状态–行动空间进行有效探索, 以搜索近似最优的前视距离控制策略. 但是多站点 CSPS 系统的协同控制问题中, 系统状态空间的大小会随着站点个数的增加和缓存库容量的增加而成指数形式 (或几何级数) 增长, 从而导致维数灾, 影响学习算法的收敛速度和优化效果. 为此, 本文在站点局域信息交互机制的基础上引入状态聚类的方法, 以减小每个站点学习空间的大小和复杂性. 首先, 将多个站点看作相对独立的学习主体, 且各自仅考虑邻近下游站点的缓存库的状态并纳入其性能值学习过程; 其次, 将原状态空间划分成多个不相交的子集, 每个子集用一个抽象状态表示, 然后, 建立基于状态聚类的多站点反馈式 Q 学习算法. 通过该方法, 可在抽象状态空间上对各站点的前视距离策略进行优化学习, 以寻求整个系统的生产率最大. 仿真实验结果说明, 与一般的多站点反馈式 Q 学习方法相比, 基于状态聚类的多站点反馈式 Q 学习方法不仅具有收敛速度快的优点, 而且还在一定程度上提高了系统生产率.

关键词 多站点 CSPS 系统, 局域信息交互, 状态聚类, 反馈式 Q 学习

引用格式 唐昊, 裴荣, 周雷, 谭琦. 基于状态聚类的多站点 CSPS 系统的协同控制方法. 自动化学报, 2014, **40**(5): 901−908

DOI 10.3724/SP.J.1004.2014.00901

Coordinate Control of Multiple CSPS System Based on State Aggregation Method

TANG Hao[1,2] PEI Rong[2] ZHOU Lei[2] TAN Qi[1]

Abstract In a single conveyor-serviced production station (CSPS) system, we can learn an approximate optimal look-ahead policy by reinforcement learning (RL) through exploring the state-action space. However, for the coordinate control problem in a multiple CSPS system, the state space will grow exponentially or geometrically as the number of stations and the capacity of buffer increase. As a result, the learning process will suffer from the curse of dimensionality, which may have a negative influence on convergence speed and optimized value. Therefore, by combining a local information interaction mechanism among stations, we introduce a state aggregation method to reduce the size and complexity of each station′s leaning space. Firstly, each station is regarded as an independent learning agent that incorporates only the buffer state of its nearest downstream station into its own learning process. Secondly, the original state space is divided into several disjoint sets and each set is represented by an abstract state, and a multiple-agent state aggregation feedback Q-learning (SAFQL) algorithm is proposed afterwards. Through our proposed approach, the agent can learn an optimized look-ahead policy over the abstract state space to improve the entire system′s processing rate. Finally, we demonstrate by a numerical example that, in comparison to general feedback Q-learning algorithm, SAFQL algorithm can not only fasten the convergence speed, but also improve the processing rate in some degree.

Key words Multiple conveyor-serviced production station (CSPS), local information interaction, state aggregation, feedback Q-learning (SAFQL)

Citation Tang Hao, Pei Rong, Zhou Lei, Tan Qi. Coordinate control of multiple CSPS system based on state aggregation method. *Acta Automatica Sinica*, 2014, **40**(5): 901−908

现实世界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, 称为传送带给料加工站 (Conveyor-serviced production station, CSPS), 它源自以福特 (Ford) 生产线为代表的工业生产自动化过程, 是现代生产中广泛存在的一类生产自动化系统的抽象模型[1−7]. 作为一类具有智能性的柔性生产加工模型, 它已被部分先进制造企业所采用, 如深圳富士通的手机液晶生产线和无锡东芝的硬盘装配线上, 故研究该类系统的最优控制问题, 具有重要的现实意义. 日本电气通信大学的松

收稿日期 2013-01-28 录用日期 2013-05-11
Manuscript received January 28, 2013; accepted May 11, 2013
国家自然科学基金 (61174186, 71231004), 国家国际科技合作项目 (2011FA10440), 教育部新世纪优秀人才计划项目 (NCET-11-0626), 高等学校博士学科点专项科研基金 (20130111110007) 资助
Supported by National Natural Science Foundation of China (61174186, 71231004), the International Science and Technology Cooperation Program of China (2011FA10440), Program for New Century Excellent Talents in University (NCET-11-0626), and Specialized Research Fund for the Doctoral Program of Higher Education (20130111110007)
本文责任编委 宋士吉
Recommended by Associate Editor SONG Shi-Ji
1. 合肥工业大学电气与与自动化工程学院 合肥 230009 2. 合肥工业大学计算机与信息学院 合肥 230009
1. School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009 2. School of Computer and Information, Hefei University of Technology, Hefei 230009

井正之教授是最早从事 CSPS 系统研究的先驱之一, 已建立了单站点 CSPS 的半 Markov 决策过程 (Semi-Markov decision process, SMDP) 模型和相关理论计算方程[1]. 结合此项工作, 文献 [3] 提出了一种基于摄动思想的模型无关 (Model-free) 在线策略迭代算法, 以解决 CSPS 系统的最优控制问题.

随着现代生产制造业的发展, 很多生产线往往配置有多个加工站点, 于是诞生了多站点 CSPS 系统[4−7]. Nakase 等[5] 和 Feyzbakhsh 等[6] 多站点 CSPS 系统看作一个整体, 并运用遗传算法及其改进的算法对多站点 CSPS 系统的最优前视距离控制策略进行求解. 但是对于多站点 CSPS 系统, 常规的数值求解方法存在 "建模难" 的问题. 因此, 在前期工作基础上, 本课题组针对多站点 CSPS 的协同前视距离控制问题, 把每个 CSPS 站点视为一个自主的智能体处理, 并运用性能势理论, 研究了一种适用于平均和折扣两种性能准则下的与模型无关的多智能体强化学习算法[7], 从而解决了系统的异步决策和学习控制问题.

最近研究结果揭示, 与单站点 CSPS 相比, 若进一步提高多站点 CSPS 系统的缓存库容量, 将会提高系统总的生产率. 但是缓存库容量的增加, 将导致系统状态空间增大, 且状态 – 行动空间将呈几何级数增, 从而影响强化学习算法的学习速度和学习效果、降低学习效率, 阻碍学习算法在实际系统中的应用. 另外, 我们最近在多站点 CSPS 系统协同控制的研究过程中还发现, 随着缓存库容量的增加, 有很多状态 – 行动对的性能值学习对于优化过程来说是冗余的或影响甚微. 目前克服这种 "维数灾" 问题的常用方法有状态聚类和分层强化学习等. 其中, 分层强化学习通过抽象技术 (如状态抽象、时态抽象等) 降低状态空间和行动空间的维数或规模[8−9]; 状态聚类将多个相似的状态聚为一个抽象状态, 以达到减少状态空间的目的. 通常, 聚类状态之间的变化可看作事件, 决策者仅当该类事件发生时才选择行动. 因此, 本文将借鉴事件驱动优化思想[10], 考虑采用状态聚类思想对系统的原始状态空间进行划分聚类, 以减少站点的学习空间大小和复杂性, 然后, 在局域信息交互的基础上, 引入状态聚类反馈式 Q 学习算法, 从而提高算法的学习速度和效率, 提高算法的实时性, 并寻求整个系统的生产加工率最大.

1 多站点 CSPS 系统

多站点 CSPS 的物理模型见文献 [7], 它概括了现代生产企业中常用的一类自动化生产线, 如机器人装配线等. 这类生产线以加工站为主体, 工件由传送带运到加工站进行加工[3]. 如果工件周期性到达且工件加工时间固定, 只要流水线的节拍设计合理, 就会实现最优生产控制. 而在工件的到达或其加工过程具有随机性的情况下, 通常需要为每个加工站配置一个存放未加工工件的缓存库, 以提高生产加工过程的柔性与可靠性, 并减少加工站的随机空闲等待时间, 从而提高系统生产率. 多站点 CSPS 的物理模型中, 每个站点可被看作一个 Agent, 每个 Agent 通过前视传感器, 沿着传送带向前看一段距离, 即前视距离. 若前视距离内至少有一个工件, 则 Agent 等待第一个工件到达并卸载放入缓存库 (Buffer) 中; 否则, Agent 从缓存库中取出一个工件进行加工. 其中, 工件的加工过程分 L 步进行, 每步加工时间一般假设服从指数分布, 其中, Agent 从 Buffer 中取出工件可看作加工的第一步, 而 Agent 加工完成后的产品放入成品库 Bank 中. 这里的前视距离即为需要考虑的决策变量 (或称为控制变量、行动), 一般取决于缓存库的当前空余量. 系统的协同控制优化目标是, 每个 Agent 各自选择一个最优前视控制策略, 相互协同共同使系统总的加工率达到最大.

在多站点 CSPS 系统中, 由于站点的串行分布特点, 上游站点的决策对下游所有站点的运行都将产生影响, 但是如果没有信息交互, 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负载平衡, 进而影响系统生产加工率的提高. 故本文在引入状态聚类方法的同时, 与文献 [7] 类似, 依然借鉴文献 [11] 的局域信息交互机制, 把下游站点 $i+1$ 的缓存库状态信息反馈到上游站点 i 的代价函数学习中, 通过反应扩散机制寻求实现站点间的负载平衡, 使系统总的加工率最大.

不失一般性, 本文研究基于以下假设: 1) 站点间间距相等; 2) 传送带匀速运行, 故前视距离可等效成前视时间; 3) Bank 的容量无限大, 且不考虑 Bank 的库存代价; 4) 工件到达服从参数为 λ 的泊松分布; 5) 加工时间服从 L 阶 Erlang 分布; 6) 工件的卸载时间忽略不计, 加工完放置 Bank 的时间也忽略不计 (可看作处理环节一部分).

2 基于状态聚类的多站点反馈式 Q 学习方法

状态聚类方法是通过类化来移除冗余信息或隐藏不相关信息, 达到简化原始状态空间的目的[12]. 其基本思想是在原来的状态空间上, 通过一个映射函数 (即抽象函数), 将原状态空间映射成多个类, 每类表示一个抽象状态, 由多个原状态组成, 显然聚类后的抽象状态空间小于原状态空间, 强化学习算法在抽象的状态空间上学习[13]. 针对大规模的问题, 聚类技术是一种直观且可用的技术, 而聚类的关键在于如何将原状态空间进行抽象, 因此, 研究不同的

抽象机制, 对解决各种实际问题产生深远影响. Li 等给出了形式化的在线状态聚类定义, 并分析了 MDP 模型下的 5 种在线抽象机制, 最后, 将其应用于位翻转 (Bit-Flip) 实例中, 以分析每种聚类机制对减小状态空间的贡献[13]. 在这些抽象机制下, 状态聚类的依据是状态–行动对的值. 而 Singh 等提出了软聚类思想, 即定义一个聚类概率矩阵 P, 每个状态 s 以概率 $P(x|s)$ 属于类 x (每个状态可属多个类), 并给出了固定的软聚类方法和自适应软聚类方法[14]. Gunady 等提出把获取相同报酬的邻近状态聚成一类以减小状态空间, 并将此状态聚类技术应用于常规的 Q 学习当中[15]. 在多站点 CSPS 系统中, 随着站点缓存库容量的增加, 状态空间将对应变大, 特别是状态–行动空间将成几何级数增长. 对于 Q 学习算法来说, 为了获得近似最优策略而在相对较大的状态空间上进行充分的行动探索会影响算法的实时性, 甚至是不必要的. 另外, 由于各站点的状态形式简单, 故本文采用邻近状态划分的聚类方法, 以此建立多站点系统的反馈式 Q 学习算法, 在该算法中其状态聚类采取的是固定划分方式, 学习过程不需要进行聚类计算, 因此, 几乎没有额外计算代价. 若采取在线聚类和软聚类方法, 将会带来明显的额外计算代价.

2.1　基于状态聚类的数学模型

在多站点 CSPS 系统的物理模型[7] 中, 假设工件按照参数为 λ 的泊松流到达第一个加工站, 工件加工时间服从 Erlang 分布, 如果站点个数仅为 1, 则系统的最优前视距离控制问题可建模成 SMDP 模型来研究[16]. 当站点个数不为 1 时, 文献 [7] 将缓存库的剩余容量作为状态, 运用性能势理论, 构建了一种适用于平均和折扣两种性能准则的反馈式多 Agent Q 学习算法, 以求解异步决策模式下的多站点协同控制策略. 与此类似, 在状态聚类方法中, 论文仍将系统中的每个站点看作一个自主学习的 Agent, 下面以一个典型的 Agent i (i 表示站点的序号, $1 \le i \le N$) 为例, 叙述具体的数学模型和算法.

假设每个站点的缓存库容量为 M, 记 Agent i 的缓存库空余容量为状态 s_i, 则有 $0 \le s_i \le M$, 且其原始状态空间为 $S_i = \{0, 1, 2, \cdots, M\}$. 记聚类后的状态空间为 E_i, 并定义抽象函数 $\phi : S_i \to E_i$, 则对任意的 s_i, $\phi(s_i) \in E_i$ 为聚类后的一个抽象状态, 显然 ϕ 非一一映射. 反之, 在抽象反函数的作用下, 对应抽象状态 e_i, $\phi^{-1}(e_i)$ 是一集合, 由一些相似的原状态组成. 这意味着 $\{\phi^{-1}(e_i)|e_i \in E_i\}$ 将原始状态空间 S_i 划分成多个类, 即聚类后的抽象状态空间. 本文将原状态空间划分成 4 个抽象状态, 分别称作全满状态 e_{ff}、半满状态 e_{sf}、半空状态 e_{sv}, 以及全空状态 e_{fv}. 具体的映射过程如下: $0 \to e_{ff}$, $[1, M/2] \to e_{sf}$, $(M/2, M) \to e_{sv}$, $M \to e_{fv}$. 因此, 经过状态聚类后, 本文的抽象状态空间为 $E_i = \{e_{ff}, e_{sf}, e_{sv}, e_{fv}\}$. 设 Agent i 的前视距离为控制变量 (即行动), 只与聚类状态 e_i 有关, 记为 $a_i(e_i)$, 且 $a_i(e_i) \in A_i = [0, l] \cup \{\infty\}$, 其中, A_i 为 Agent i 的可用行动集, l 为 Agent i 的最大前视距离. 因为传送带是匀速的, 故传送带上的距离可等效成时间来表示. 显然, 系统存在如下两种特殊工作状态, 其行动具有唯一性.

1) 全空状态 e_{fv} : Agent i 将一直等待, 直到有工件到达, 等效为 $a_i(e_{fv}) \equiv \infty$;

2) 全满状态 e_{ff} : Agent i 无需前视, 直接从缓存库中取出工件进行加工, 等效为 $a_i(e_{ff}) \equiv 0$.

于是, Agent i 的聚类前视策略 v_i 定义为 $v_i(a_i(e_{ff}), a_i(e_{sf}), a_i(e_{sv}), a_i(e_{fv}))$. 另外, 聚类状态和行动的组合数将远小于原状态–行动对的总数, 即待学习的策略空间呈几何级数减小.

假设 Agent i 在当前决策时刻 T_n ($T_0 = 0$) 时的聚类状态为 $e_i(T_n)$, 根据策略 v_i 选择行动 $a_i(e_i(T_n))$. 如果在 $a_i(e_i(T_n))$ 内至少有一个工件, 并且第一个工件离 Agent i 的位置为 $\theta_i(T_n)$, 则 Agent i 等待 $\theta_i(T_n)$ 后, 将工件从传送带上卸载下来并放入缓存库中 (不失一般性, 可假设卸载是瞬时完成的, 没有时间消耗), 然后, 进入下一决策时刻 $T_{n+1} = T_n + \theta_i(T_n)$, 且其自身状态转移到聚类状态 $e_i(T_{n+1})$. 在此转移过程中, 站点消耗的单位时间代价如下:

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

其中, K_1 表示单位时间等待代价, K_2 表示本站点与下一站点的缓存库空余量差值的单位时间反馈代价.

如果在 $a_i(e_i(T_n))$ 内没有工件, 则 Agent i 从缓存库中取出一个工件进行加工, 设加工时间为 $\mu_i(T_n)$. 工件加工完成后, 放入 Bank 中 (假设是瞬时完成的, 没有时间消耗), 并转入下一决策时刻 $T_{n+1} = T_n + \max(a_i(e_i(T_n)), \mu_i(T_n))$, 同时其自身状态转向聚类状态 $e_i(T_{n+1})$. 在此转移过程中, 站点消耗的单位时间代价如下:

当 $T_n \le t < T_n + \mu_i(T_n)$ 时,

$$f^i(e_i(T_n), a_i(e_i(T_n)), e_i(T_{n+1})) = K_2(s_{i+1}(T_n) - s_i(T_n)) \tag{2}$$

当 $T_n + \mu_i(T_n) \le t < T_{n+1}$ 时,

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

其中, t 表示状态转移过程中的任意时刻, $T_{n+1}-(T_n+\mu_i(T_n))$ 为等待时间. 另外, 式 (1)∼(3) 中的反馈项 $K_2(s_{i+1}(T_n)-s_i(T_n))$ 可选择多种定义形式, 例如换成 $K_2\cdot \mathrm{e}^{(s_{i+1}(T_n)-s_i(T_n))}$, 但是针对不同形式的反馈项, K_2 的大小应有所不同, 以保证合适的反馈作用.

在代价式 (1)∼(3) 中, 既有反映自身的等待代价项 K_1, 又有反映邻近站点与自身站点状态差值信息的反馈代价项 $K_2(s_{i+1}-s_i)$, 其设计源自反应扩散思想[11]. 从公式形式上, 只有下游站点向上游站点反馈, 是单向的, 实际上, 上游站点的决策结果直接影响下游站点的操作, 故这种交互仍然是双向的.

在多站点系统的优化控制中, 每个 Agent 相对独立, 其控制策略仅依赖于自身的聚类状态, 与其它站点无关, 站点间的交互只是体现在代价函数的计算中, 并不会因为多 Agent 之间的交互而改变学习空间的大小. 系统学习优化目标是为每个 Agent 寻找一个最优的从聚类状态到行动的映射, 即聚类状态前视策略, 使平均代价性能准则 η^{v_i} 和折扣代价性能准则 $\eta_\alpha^{v_i}(e_i)$ ($\alpha>0$, 为折扣因子) 最小. 其中 $\eta_\alpha^{v_i}(e_i)$, 表示系统在聚类状态 v_i 策略下从初始聚类状态 e_i 出发的无穷时段折扣累积代价, 而 e_i 一般包含了多个原始状态. 根据文献 [16] 可知, 任意给定一个控制策略, $\eta_\alpha^{v_i}(s_i)$ 与原始状态 s_i 有关, 因此 $\eta_\alpha^{v_i}(e_i)$ 针对不同的原始状态取值可能不同.

2.2 状态聚类反馈式 Q 学习算法

Q 学习算法是强化学习中应用最广泛的学习算法之一, 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, 不断学习并逼近状态–行动对的函数值而对问题进行求解. 尽管该算法在适当的假设下能够保证收敛到最优的结果, 但为了保证收敛, 需要无穷次访问所有的状态–行动对, 故在实际系统运用时, 往往面临收敛速度慢和学习不充分的缺点. 在引入状态聚类方法以后, 每个 Agent 的抽象状态空间明显减小, 一方面收敛速度有了显著提高, 学习时间明显缩短; 另一方面又能保证每个聚类状态都能获得充分学习, 进而优化效果有所提升.

在基于状态聚类的反馈式 Q 学习过程中, Agent i 的一个典型聚类状态转移记 $(e_i(T_n),a_i(e_i(T_n)),e_i(T_{n+1}))$, 对应一个观测样本 $\langle s_i(T_n),a_i(e_i(T_n)),s_i(T_{n+1}),w_i(T_n),\mu_i(T_n)\rangle$, 其中, $w_i(T_n)=T_{n+1}-T_n$ 表示相邻两次决策的间隔时间. 根据此观测样本, 可计算 Agent i 从 T_n 转移到 T_{n+1} 这一过程中累积的总代价 $f_\alpha^i(e_i(T_n),a_i(e_i(T_n)),e_i(T_{n+1})),\alpha\geq 0$.

当 Agent i 卸载工件时,

$$\begin{aligned} f_\alpha^i(e_i(T_n),a_i(e_i(T_n)),e_i(T_{n+1}))=\\ K_1\cdot T_\alpha(w_i(T_n))+K_2\cdot(s_{i+1}(T_n)-\\ s_i(T_n))\cdot T_\alpha(w_i(T_n)) \end{aligned} \tag{4}$$

当 Agent i 加工工件时,

$$\begin{aligned} f_\alpha^i(e_i(T_n),a_i(e_i(T_n)),e_i(T_{n+1}))=\\ K_1\cdot(T_\alpha(w_i(T_n))-T_\alpha(\mu_i(T_n)))+\\ K_2\cdot(s_{i+1}(T_n)-s_i(T_n))\cdot T_\alpha(w_i(T_n)) \end{aligned} \tag{5}$$

其中, 对 $\forall w>0$, 有 $T_\alpha(w)=\int_0^w \mathrm{e}^{-\alpha t}\mathrm{d}t=(1-\mathrm{e}^{-\alpha w})/\alpha$, 且当 $\alpha\to 0$, 有 $T_0(w)=w$.

由式 (4) 和式 (5), 以及基于性能势的 Q 学习理论[4, 16], 可得 Agent i 在折扣准则和平均准则下统一的即时差分公式为

$$\begin{aligned} d_i(T_n)=f_\alpha^i(e_i(T_n),a_i(e_i(T_n)),e_i(T_{n+1}))-\\ T_\alpha(w_i(T_n))\bar{\eta}^i-Q_i(e_i(T_n),a_i(e_i(T_n)))+\\ \mathrm{e}^{-\alpha w_i(T_n)}\min_{a\in A_i}Q_i(e_i(T_{n+1}),a_i(e_i(T_{n+1}))) \end{aligned} \tag{6}$$

其中, $\bar{\eta}^i$ 表示 Agent i 的平均代价的学习值. 于是, Agent i 基于状态聚类的 Q 值更新公式如下:

$$\begin{aligned} Q_i(e_i(T_n),a_i(e_i(T_n)))=Q_i(e_i(T_n),a_i(e_i(T_n)))+\\ \gamma_i(e_i(T_n),a_i(e_i(T_n)))d_i(T_n) \end{aligned} \tag{7}$$

其中, $\gamma(e_i(T_n),a_i(e_i(T_n)))$ 表示 Agent i 的 Q 值学习步长, 一般比平均代价的学习步长衰减慢, 与文献 [7] 和 [16] 类似, 可取 $\gamma_i(e_i(T_n),a_i(e_i(T_n)))=1/c_i(e_i(T_n),a_i(e_i(T_n)))^\beta$, $0<\beta<1$. $c_i(e_i(T_n),a_i(e_i(T_n)))$ 为 Agent i 访问聚类状态–行动对 $(e_i(T_n),a_i(e_i(T_n)))$ 的次数.

于是, Agent i 的状态聚类反馈式 Q 学习算法 (State aggregation feedback Q-learning, SAFQL) 的具体步骤如下:

步骤 1 (状态划分). 对原始状态空间 S_i 进行划分, 形成抽象状态空间 E_i;

步骤 2 (初始化). 令 $n=0$, 并对所有的聚类状态–行动对, 令 $Q_i(e_i(T_n),a_i(e_i(T_n)))=0$;

步骤 3 (行动选择). 在 T_n 决策时刻, 在抽象状态 $e_i(T_n)$ 下, 根据 ε-greedy 策略, 选择行动 $a_i(e_i(T_n))$;

步骤 4 (执行行动并观测计算代价). 执行行动 $a_i(e_i(T_n))$ 后, Agent i 进行一次聚类状态转移 $(e_i(T_n),a_i(e_i(T_n)),e_i(T_{n+1}))$, 相应获得一个观测样本 $\langle s_i(T_n),a_i(e_i(T_n)),s_i(T_{n+1}),w_i(T_n),\mu_i(T_n)\rangle$, 然后根据式 (4) 和式 (5) 计算一步累积代价 $f_\alpha^i(e_i(T_n),a_i(e_i(T_n)),e_i(T_{n+1}))$;

步骤 5 (Q 值更新). 计算平均代价 $\overline{\eta}^i$, 然后根据式 (6) 计算即时差分 $d_i(T_n)$, 最后根据式 (7), 更新 Q 值;

步骤 6 (终止条件判断). 若算法满足终止条件, 则算法结束; 否则, 令 $n := n + 1$, 转向步骤 3.

3　实验结果

仿真实验中, 系统物理参数和部分学习算法参数的设置如表 1 所示. 其中, 每个工件加工时间服从 L 阶 Erlang 分布.

表 1　参数设置表

Table 1　Parameters setting table

参数名称	参数值
工件到达率 λ	0.8
工件服务 Erlang 分布 $(L, \bar{\mu})$	$(4, 1)$
站点个数 N	4
最大前视距离 l	4
缓存库容量 M	9
聚类个数 C	4
单位时间等待代价系数 K_1	2.0
缓存库空余量差值单位时间反馈代价系数 K_2	1.5
学习步长的指数因子 β	0.5

在多站点 CSPS 系统中, 缓存库用来存放未加工的工件, 为每个 Agent 配置一个缓存库可以减少每个工件生产周期的平均等待时间, 增加平均加工时间, 进而提高整个系统的加工率. 但是缓存库的容量配置并不是越多越好, 随着缓存库容量的增加系统所能达到的最优值最终会出现饱和现象. 特别地, 由于系统状态空间大小取决于缓存库容量, 而强化学习方法是通过数据样本对所有可能的状态–行动对进行学习, 并且在实际应用中, 学习样本的获取能力一般受限, 因此, 缓存库容量设计过大, 反而由于学习不够充分影响常规算法的学习优化效果. 而运用状态聚类方法则能有效提高算法的学习效率, 克服原始状态空间比较大时学习不充分的缺点. 图 1 给出了在不同缓存库容量 M 下分别运用 FQL 和 SAFQL 两种算法获得的系统加工率变化曲线, 其中, SAFQL 算法中采用固定的聚类状态个数 4 (故其曲线可从 $M = 5$ 的聚类开始). 从 FQL 对应的曲线可见, $M \le 9$ 时, 系统的加工率随着缓存库容量的增加而变大, 而当 $M > 9$ 时, 系统加工率随着缓存库容量的增加反而变小或基本不变. 从 SAFQL 对应曲线可见, 随着 M 的增大, 学习值一直是上升的 (后期有饱和趋势). 因此, 在仿真实验中, 为了便于比较, 我们主要考察 $M = 9$ 时的情况 (此时 FQL 算法能获得相对比较好的优化结果), 此时, 聚类状态按图 2 所示的形式进行划分.

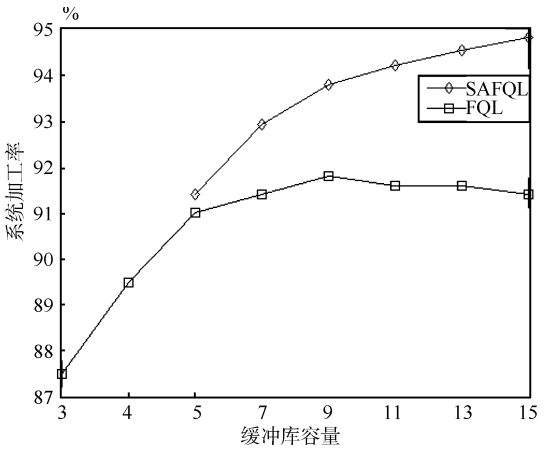


图 1　两种学习算法下的系统加工率

Fig. 1　The processing rates of the two algorithms under different capacities of the buffer

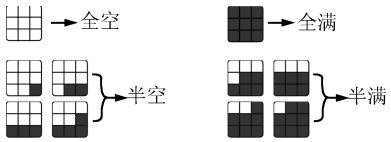


图 2　状态划分聚类示意图

Fig. 2　The aggregating diagram of state partition

根据表 1 的参数设置, 整个系统的仿真学习过程如下: 算法从初始策略 v_0 开始, 每学习 2 000 步得到一个新的 Greedy 策略 v, 然后, 用 Monte-Carlo 方法对该策略进行评估, 即在策略 v 下, 系统多次独立运行 200 000 步, 并对代价结果进行统计平均作为该策略的性能评估值. 另外, 后面所给的一些实验结果数值是算法多次运行再统计平均的结果.

首先考虑平均准则优化问题. 图 3 为 FQL 和 SAFQL 两种算法的系统总加工率优化曲线, 可见两种算法都能有效提升系统的加工率. 但是 SAFQL 算法是在减小的聚类状态空间上学习, 学习探索更集中, 因此, 在相同学习步数下, 学习更加充分, 其学习速度比普通的 FQL 明显快得多, 而且学习效果也更好. 图 4 是两种学习算法下系统总的平均代价曲线, 其变化规律与系统加工率正好反向对应. 图 5 分别统计了两种算法获得的最终 Greedy 策略作用下各站点的负载率. 由于引入了邻近站点间的缓库存空余量差值作为多 Agent 系统学习的交互扩散项, SAFQL 与 FQL 两种算法都具有较好的负载均衡性 (系统运行模式决定了上游站点出力肯定多于下游站点, 不会达到绝对平均). 其中, T 时段内 Agent i 的负载率计算公式为 $P_L^i(T) = P_i(T)/P(T)$. $P_i(T)$ 为

T 时段内 Agent i 加工工件的个数, $P(T)$ 为 T 时段内所有站点加工工件的总数.

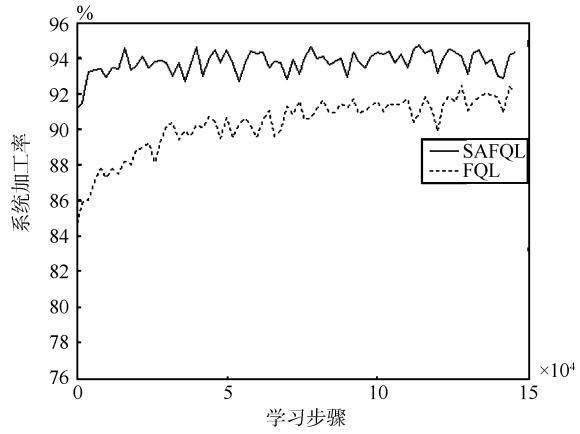


图 3 两种学习算法下的系统加工率

Fig. 3 The system processing rates of the two algorithms

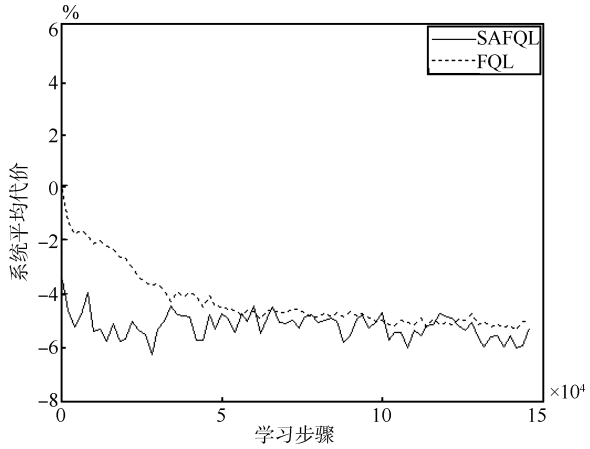


图 4 两种学习算法下的系统总平均代价

Fig. 4 The system total average costs of the two algorithms

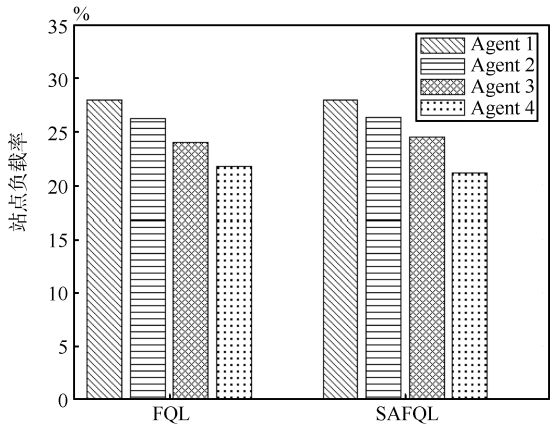


图 5 各站点的负载率

Fig. 5 Load rates of each station

图 6 是两种算法在折扣因子 $\alpha = 0.1$ 时, 分别从初始状态 $E = \{e_{sv}, e_{sv}, e_{sf}, e_{sf}\}$ 出发获得的系统加工率优化曲线, 图 7 是相应的折扣代价优化曲线. 可见折扣准则下, 较之于 FQL 算法, SAFQL 算法在学习速度和优化效果上仍然具有明显优势. 最后, 考虑本文提出的聚类状态学习机制对状态空间减小的贡献. 图 8 给出了不同 Buffer 容量设计时的系统加工率曲线, 表 2 总结了系统在不同缓存库容量配置下的原始状态空间大小和抽象状态空间大小, 以及两种学习算法所求最终策略对应的生产加工率情况. 显然, SAFQL 算法中, 每个站点的抽象状态空间大小为 $C = 4$, 整个系统的抽象状态空间大小为 $C^N = 4^4$. 而一般的 FQL 算法中, 每个站点的抽象状态空间大小为 $M + 1$, 整个系统的状态空间计算公式为 $(M + 1)^4$. 通过表 2 数据和图 8 综合可见, 通过本文给出的状态聚类机制, 缓存库容量越大, 状态空间减少得越多, 学习结果也有一定改进, 并能保持较快的收敛速度.

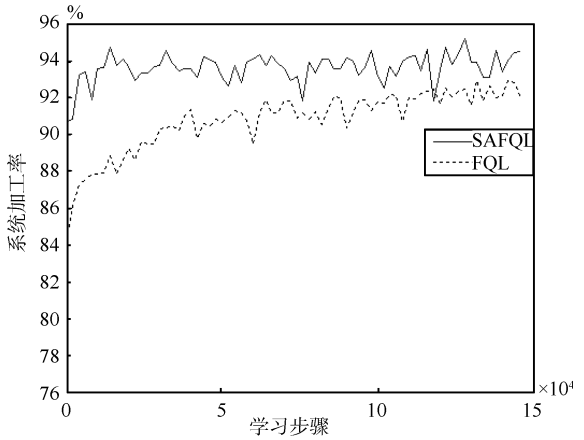


图 6 两种学习算法在折扣准则下系统的加工率

Fig. 6 The system processing rates of the two algorithms with discounted criterion

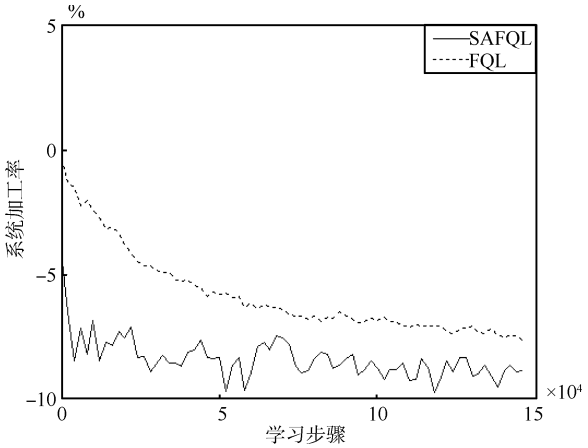


图 7 两种学习算法下系统的折扣代价

Fig. 7 The system discount costs of the two algorithms

表 2　两种算法在不同缓存库容量下的系统状态空间和学习收敛步数

Table 2　The system state space and convergence steps of the two algorithms under different capacities of the buffer

性能指标		$M=5$	$M=7$	$M=9$	$M=11$	$M=13$
站点、整个系统状态空间大小	FQL	(6, 1 296)	(8, 4 096)	(10, 10 000)	(12, 20 736)	(14, 38 416)
	SAFQL	(4, 256)	(4, 256)	(4, 256)	(4, 256)	(4, 256)
状态空间减少百分比 (%)		(33.3, 80.2)	(50, 93.8)	(60, 97.4)	(66.6, 98.7)	(71.4, 99.3)
系统最终策略对应的生产率	FQL	0.9103	0.916	0.9179	0.9140	0.9132
	SAFQL	0.9141	0.9293	0.9379	0.9422	0.9454
生产率提高百分比 (%)		40	15	22	31	35

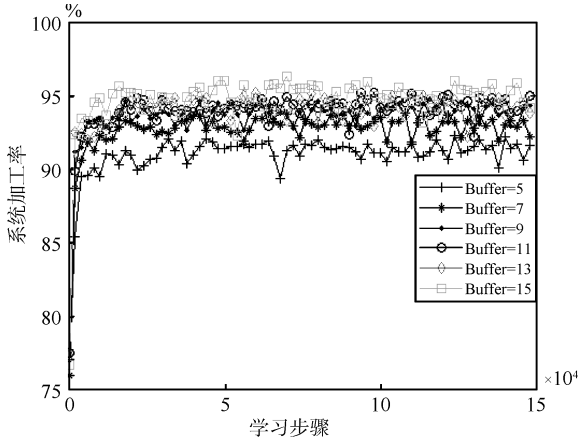


图 8　SAFQL 算法在不同 Buffer 容量下的加工率

Fig. 8　The processing rates of SAFQL algorithm under different capacity of the buffer

4　结论

本文首先采用聚类方法, 将相似的状态映射成一个抽象状态, 以达到减少状态空间的目的; 然后, 采用反应扩散思想, 将下游站点的缓存库剩余容量信息通过代价函数反馈到上游站点的学习过程中. 结合上述两种思想, 本文提出了状态聚类反馈式 Q 学习算法, 以优化多站点 CSPS 系统前视距离的控制问题. 最后的实验结果显示: 反馈项的引入, 实现了上下游站点的相互协作, 不仅改善了站点间的负载平衡, 也一定程度上提高了生产处理率; 而状态聚类技术的引入, 在收敛速度和优化效果两方面均起到了明显的改进作用. 特别地, 状态聚类方法减少了计算复杂度与空间复杂度, 更适合解决一类实际离散事件动态系统的在线优化控制或实时控制. 本文中, 状态聚类是一种硬聚类机制, 即每个原始状态都以概率 1 映射到某一个类, 且在 Q 值学习之前就已确定并保持不变, 因而可离线完成. 然而此种聚类机制在动态变化的复杂环境中不能正确反映系统的随机动态特性. 因此, 可考虑选择软聚类机制, 建立基于在线聚类的学习优化算法, 即每学习一段时间, 便根据学习情况对状态空间的概率划分进行重新调整. 此方法如何应用于 CSPS 系统的优化控制, 将有待进一步研究.

References

1 Matsui M. A generalized model of convey-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 Hao T, Tamio A. Look-ahead control of conveyor-serviced production station by using potential-based online policy iteration. *International Journal of Control*, 2009, **82**(10): 1917−1928

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

5 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

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

7 Tang Hao, Wan Hai-Feng, Han Jiang-Hong, Zhou Lei. Coordinated look-ahead control of multiple CSPS system by multi-agent reinforcement learning. *Acta Automatica Sinica*, 2010, **36**(2): 289−296
(唐昊, 万海峰, 韩江洪, 周雷. 基于多 Agent 强化学习的多站点 CSPS 系统的协作 Look-ahead 控制. 自动化学报, 2010, **36**(2): 289−296)

8 Yan Q C, Liu Q, Hu D J. A hierarchical reinforcement learning algorithm based on heuristic reward function. In: Proceedings of the 2nd IEEE International Conference on Advanced Computer Control. Shenyang, China: IEEE, 2010. 371−376

9 Botvinick M M. Hierarchical reinforcement learning and decision making. *Current Opinion in Neurobiology*, 2012, **22**(6): 956−962

10 Jia Q S. Event-based optimization with lagged state information. In: Proceedings of the 31st Chinese Control Conference. Hefei, China: IEEE, 2012. 2055−2060

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

12 Wright R, Lin S. Evolutionary tile coding: an automated state abstraction algorithm for reinforcement learning. In: Proceedings of the the 2010 Abstraction, Reformulation, and Approximation. Atlanta, Georgia, USA: the Association for the Advancement of Artificial Intelligence Workshops, 2010

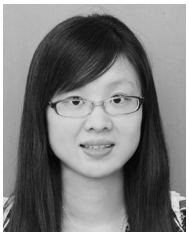
13 Li L H, Walsh T J, Littman M L. Towards a unified theory of state abstraction for MDPs. In: Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics. Fort Lauderdale, Florida, USA: Kluwer AcademicPublishers, 2006. 531−539

14 Singh S P, Jaakkola T, Jordan M I. Reinforcement learning with soft state aggregation. In: Proceedings of the 1995 Conference on Neural Information Processing Systems. Denver, CO, USA: MIT, 1995. 361−368

15 Gunady M K, Gomaa W. Reinforcement learning generalization using state aggregation with a maze-solving problem. In: Proceedings of the 2012 Japan-Egypt Conference on Electronics, Communication and Computers. Alexandria, Egypt: IEEE, 2012. 157−162

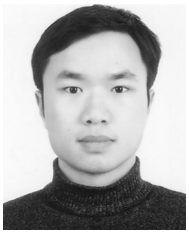
16 Cao X R. Semi-Markov decision problems and performance sensitivity analysis. *IEEE Transaction on Automatic Control*, 2003, **48**(5): 758−769

唐 昊 合肥工业大学电气与自动化工程学院教授. 2002 年获得中国科学技术大学博士学位. 主要研究方向为离散事件动态系统, 强化学习, 神经元动态规划及智能优化. 本文通信作者.
E-mail: htang@hfut.edu.cn
(**TANG Hao** Professor at the College of Electrical Engineering and Automation, Hefei University of Technology. He received his Ph. D. degree from University of Science and Technology of China in 2002. His research interest covers discrete event dynamic system, reinforcement learning, neural dynamic programming and intelligent optimization. Corresponding author of this paper.)



裴 荣 合肥工业大学计算机与信息学院硕士研究生. 2010 年获得合肥工业大学计算机与信息学院学士学位. 主要研究方向为强化学习, 生产线优化.
E-mail: peirong_1987@163.com
(**PEI Rong** Master student at the College of Computer and Information, Hefei University of Technology. She received her bachelor degree from Hefei University of Technology in 2010. Her research interest covers reinforcement learning and the production line optimization.)

周 雷 合肥工业大学计算机与信息学院博士研究生. 2006 年获得合肥工业大学计算机与信息学院硕士学位. 主要研究方向为离散事件动态系统, 强化学习, 智能优化方法.
E-mail: zhouleizhl@163.com
(**ZHOU Lei** Ph. D. candidate at the College of Computer and Information, Hefei University of Technology. He received his master degree from Hefei University of Technology in 2006. His research interest covers discrete event dynamic system, reinforcement learning, and intelligent optimization methods.)



谭 琦 合肥工业大学电气与自动化工程学院讲师, 博士. 主要研究方向为生产优化调度, 智能计算方法.
E-mail: tanqi@hfut.edu.cn
(**TAN Qi** Ph. D., lecturer at the College of Electrical Engineering and Automation, Hefei University of Technology. His research interest covers production scheduling optimization and intelligent calculation methods.)