计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(12):195−199 [doi: 10.15888/j.cnki.csa.007196]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

应用强化学习算法求解置换流水车间调度问题①

张东阳, 叶春明

(上海理工大学 管理学院, 上海 200093)
通讯作者: 张东阳, E-mail: 18721813835@163.com

摘　要: 面对日益增长的大规模调度问题, 新型算法的开发越显重要. 针对置换流水车间调度问题, 提出了一种基于强化学习 Q-Learning 调度算法. 通过引入状态变量和行为变量, 将组合优化的排序问题转换成序贯决策问题, 来解决置换流水车间调度问题. 采用所提算法对 OR-Library 提供 Flow-shop 国际标准算例进行测试, 并与已有的一些算法对比, 结果表明算法的有效性.

关键词: 置换流水车间调度; 强化学习; Q-Learning; 最大完工时间

引用格式: 张东阳,叶春明.应用强化学习算法求解置换流水车间调度问题.计算机系统应用,2019,28(12):195–199. http://www.c-s-a.org.cn/1003-3254/7196.html

Reinforcement Learning Algorithm for Permutation Flow Shop Scheduling to Minimize Makespan

ZHANG Dong-Yang, YE Chun-Ming

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In the face of increasing large-scale scheduling problems, the development of new algorithms becomes more and more important. A Q-Learning scheduling algorithm based on reinforcement learning is proposed for permutation flow shop scheduling problem. By introducing state variables and behavior variables, the scheduling problem of combinatorial optimization is transformed into sequential decision-making problem to solve the permutation flow shop scheduling problem. The proposed algorithm is used to test the Flow-shop international standard provided by OR-Library, and compared with some existing algorithms, the results show that the algorithm is effective.

Key words: permutation flow shop scheduling; reinforcement learning; Q-Learning; makespan

1　引言

排序问题是生产系统和服务系统中的一类典型问题, 它作为传统问题和热点问题, 至今一直有众多学者研究. 根据工件的不同加工特点, 排序问题可分为流水作业排序 (flow-shop scheduling)、自由作业排序 (open-shop scheduling) 和异序作业排序 (job-shop scheduling)[1], 其中 Flow-shop 排序又可分为同顺序和任意序两大类, 同顺序 Flow-shop 排序问题又称置换流水车间调度问题 (Permutation Flow Shop Problem, PFSP). 相关理论证明, 3 台机器以上的 PFSP 即为 NP 难题[2], 至今还未发现具有多项式时间的优化算法.

目前, 解决这类问题有效方法主要包括: 精确算法, 启发式算法, 智能优化算法. 如枚举法、分支定界法等精确算法只对小规模问题的求解有着很好的效果. 如 Gupta 法、RA 法和 NEH 法等启发式算法可以求解快速构造问题的解, 但是解的质量较差[3]. 近年来, 遗传算法[4]、人工神经网络[5]、蚁群算法[6]、粒子群算法[7], 烟花算法[8]、进化算法[9]等智能优化算法因能在合理

① 基金项目: 国家自然科学基金 (71840003); 上海理工大学科技发展项目 (2018KJFZ043)
Foundation item: National Natural Science Foundation of China (71840003); Science and Technology Development Program of University of Shanghai for Science and Technology (2018KJFZ043)
收稿时间: 2019-05-17; 修改时间: 2019-06-06; 采用时间: 2019-06-11; csa 在线出版时间: 2019-12-10

的时间内获得较优解, 受到了众多学者的广泛研究, 并已经成为解决该类问题的重要方法.

近年来, 随着技术的进步, 机器学习领域里一个古老又崭新的理论——强化学习[10,11], 又得到了科研人员的广泛重视. 但目前强化学习主要应用在游戏比赛、控制系统和机器人等领域[12−14], 应用在生产调度中的并不多. Wang 等学者将 Q-学习算法应用到单机作业排序问题上, 发现智能体经过学习能够从给定的规则中选出较好的规则, 证明了将强化学习应用到生产调度的可能性和有效性[15,16]. Zhang 等学者利用多智能协作机制解决了非置换流水车间调度问题[17], 国内的潘燕春等学者将 Q-学习算法与遗传算法有机的结合起来, 在作业车间调度取得较好的效果 [18,19]. 王超等学者提出了改进的 Q-学习算法运用于动态车间调度, 构建了一系列符合强化学习标准的规则[20]. 可以看到, 过往的文献里面强化学习应用在调度上大部分是多智能体协作, 或者与其它算法结合去解决调度问题. 因此, 本文用单智能体的强化学习来解决置换流水车间调度问题, 合理的将状态定义为作业序列, 将动作定义为机器前可选加工的工件, 来适应强化学习方法. 智能体通过与环境不断交互, 学习一个动作值函数, 该函数给出智能体在每个状态下执行不同动作带来的奖赏, 伴随函数值更新, 进而影响到行为选择策略, 从而找到最小化最大完工时的状态序列. 最后用该算法对 OR-Library 提供 Flow-shop 国际标准算例进行仿真实验分析, 最终的实验结果验证了算法的有效性.

2 置换流水车间调度问题描述

置换流水车间调度可以描述为[21,22]: n 个工件要在 m 个机器上加工, 每个工件的加工顺序相同, 每一台机器加工的工件的顺序也相同, 各工件在各机器上的加工时间已知, 要求找到一个加工方案使得调度的目标最优. 本文选取最小化最大加工时间为目标的调度问题. 对该问题常作出以下假设:

1) 一个工件在同一时刻只能在一台机器上工;

2) 一台机器在同一时刻智能加工一个工件;

3) 工件一旦在机器上加工就不能停止;

4) 每台机器上的工件加工顺序相同.

问题的数学描述如下, 假设各工件按机器 1 到 m 的顺序加工, 令$\pi=\{\pi_1,\pi_2,\cdots,\pi_n\}$为所有工件的一个排序. p_{ij}为工件 i 在机器上 j 的加工时间, 不考虑所有工件准备时间, $C(\pi_i,j)$为工件π_i在机器j上加工完成时间.

$$\begin{cases} C(\pi_1,1)=p_{\pi_1,1} \\ C(\pi_i,1)=C(\pi_{i-1},1)+p_{\pi_1,1} \\ C(\pi_i,j)=C(\pi_1,j-1)+p_{\pi_1,j} \\ C(\pi_i,j)=\max\{C(\pi_{i-1},j),C(\pi_i,j-1)\}+p_{\pi_i,j} \end{cases} \tag{1}$$

$$makespan=C_{\max}(\pi)=\mathrm{C}(\pi_n,m) \tag{2}$$

其中, 式 (1) 中的$i=2,\cdots,n;\ j=2,\cdots,n$, 式 (2) 为最大完工时间.

3 强化学习理论

强化学习是人工智能领域中机器学习的关键技术, 不同于监督学习和无监督学习, 主要特点在于与环境交互, 具有很强的自适应能力, 具有实时学习的能力. 强化学习的目标是在与环境的试探性交互中学习行为策略, 来获取最大的长期奖赏[23]. 图 1 描述了强化学习的过程, 强化学习最主要的是两个主体, 分别为智能体和智能体所处的环境, 环境意味着多样的复杂状态, 所有的状态可以看成一个集合S. 当智能体接受到t时刻的状态$s_t(s_t\in S)$以及从上一个状态s_{t-1}转变成s_t所得到的瞬时奖励r_t, 智能体从可选的行为集合 A 中选取一个动作a_t来执行, 这样环境状态就转移为s_{t+1}, 同时智能体接受来自于环境状态改变瞬时奖励r_{t+1}和$t+1$时刻的状态s_{t+1}, 根据从中学习到的经验, 来决策$t+1$时刻的动作a_{t+1}. 按此循环, 智能通过不断与环境交互, 根据学习到的策略, 不断尝试并调整自身的行为, 来获取最大的长期奖赏.

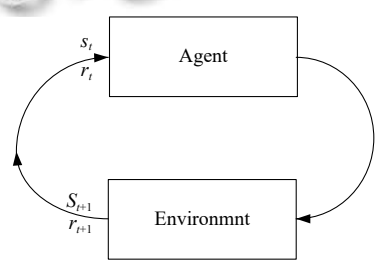

图 1 强化学习模型

强化学习的算法可以分为 2 类, 一类是基于模型已知的动态规划法, 一类是基于模型未知的算法如蒙特卡洛算法、Q-Learning 算法、TD 算法等. 本文采用的 Q-Learning 算法来解决问题. 算法的核心是一个简单的值迭代更新, 每个状态动作对都有一个 Q 值关联, 当智能体处在 t 时刻的状态s_t选择操作a_t时, 该状态动作对的 Q 值将根据选择该操作时收到的奖励和后续状

态的最佳 Q 值进行更新. 状态操作对的更新规则如下:

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha\left[r_t+\gamma\left(\max_{a_{t+1}}Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\right)\right] \tag{3}$$

式 (3) 中, α表示学习率, 学习率越大, 表明立即奖赏和未来奖赏对当前$Q(s_t,a_t)$的影响较大, 智能体越能看到未来的结果, 但收敛速度会比较慢, γ表示折扣因子, 代表决策者对得到的奖赏的偏好, 折扣因子越大, 智能体越有远见, 即考虑当前的选择对未来的结果造成的影响. 已经证明, 在马尔科夫决策过程中, 在某些限制条件下, QL 能够收敛到最优值[24]. QL 算法更新过程如算法 1 所示.

算法 1. QL 算法

初始化 Q 值
for each episode:
  初始化状态 s
  for each episode step:
    在 t 时刻下状态s_t的所有可能行为中选取一个行为a_t
    执行动作a_t, 得到下一状态s_{t+1}和奖赏值r_t
    更新 Q 值:

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha\left[r_t+\gamma\left(\max_{a_{t+1}}Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\right)\right]$$

    $s_t\leftarrow s_{t+1}$
    end for
end for

算法中的智能体有 2 种不同动作选择策略: 探索和利用, 而ε–贪心行动值法是智能体选择动作的常用的策略, 以较大概率 $(1-\varepsilon)$ 选择完全贪婪行动, 以较小概率 (贪婪率)ε随机选择行动[10]. ε越小越重于利用, 即利用现有的学习成果来选取行动, ε越大越重于探索, 即随机选取行动.

4 Q-Learning 在 PFSP 上的应用

利用强化学习解决 PFSP 问题, 最重要是如何将问题映射到强化学习模型中, 并利用相关算法来得到优化的策略结果. 本文构建了环境中的状态集, 动作和反馈函数, 并采用 QL 算法来进行优化.

定义 1. 状态集. 将 $n\times m$ 的 PFSP 问题中 m 个机器中的第一个机器当作智能体, 将第一台机器前的未加工的工件作为环境状态. 根据文献[25], 智能体的状态可以设定为$S_i=\left(A_i^r\right)$, 每个智能体 (i) 都有$|S_i|=2^n$(n 表示工件数) 状态. 在我们的案例中只有一个智能体, 因此, 所有状态的集合为$|S|=2^n$, 表示为$S=\{s_1,s_2,\cdots,s_2^n\}$.

定义 2. 动作集. 将智能体前可以选择加工的工件作为可选的动作. 因此, 在我们的案例中可选择的动作集为$A=\{a_1,a_2,\cdots,a_n\}$.

定义 3. 反馈函数. 我们选取最小化最大完工时间作为奖励信号, 最小化最大完工时间越小, 奖励值越大, 意味着选取的动作也好, 函数表示如下:

$$r=\frac{1}{makespan} \tag{4}$$

根据上述的定义, 将 PFSP 问题映射到 QL 算法中, 具体描述如算法 2 所示.

算法 2. QL 解决 PFSP 问题算法

初始化:
$Q(s, a)$={}
$Best$={}
for each episode step:
  初始化状态 s={}
  while not finished(所有工件):
   初始化所有动作值
   根据状态 s 利用贪心行动值法来选择动作
   执行动作, 得到下一个状态 s_{t+1} 和奖赏值 $r(1/makespan)$
   更新 $Q(s, a)$

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha\left[r_t+\gamma\left(\max_{a_{t+1}}Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\right)\right]$$

   $s\leftarrow s_{t+1}$
  end while
  if $makespan(s) < makespan(Best)$:
  $s\leftarrow Best$
end for

5 实验结果分析

为验证 QL 算法解决置换流水车间问题的性能, 选择 OR-Library 中 Carl 类例题, 21 个 Reeves 例题来进行测试, 将实验结果与一些算法进行比较. 算法程序用 Python3.0 编写, 运行环境为 Windows 7 64 位系统, 处理器为 2.60 GHz, 4 GB 内存. 经过初步实验, 分析了不同学习参数对学习的影响, 最后确定了相关参数, 迭代次数为 5000, 学习率为 0.1, 折扣因子为 0.8, 贪婪率为 0.2. 以相对误差率$RE=(C-C^*)/C^*\times 100\%$, 平均误差率$ARE=(C^a-C^*)/C^*\times 100\%$和最优误差率$BRE=(C^{\text{best}}-C^*)/C^*\times 100\%$为比较标准, 每个例题运行 20 次. 其中$C$为算法运行的结果, C^*为算列的最优值. C^a, C^{best}分别为所求解的平均值和最优值.

选择 Car 类例题与文献[25]提到的萤火虫算法 (FA)、粒子群算法 (PSO)、启发式算法 NEH 来进行比较. 从表 1 和图 2 中我们可以看出, QL 算法在 Car 类

算例中基本能找到最优值, 与 PSO, FA 等智能算法寻优效果上相差不大, 而启发式算法 NEH 求解算例最优的效果一般, 只适用于对精度要求不高的场合. 在算例的平均误差上, QL 算法求解质量优于 PSO 算法, 和 FA 算法不相上下, 展现了 QL 算法的良好稳定性. 说明 QL 算法对置换流水车间调度问题有较好的寻优能力.

表 1 Car 类问题测试结果

问题	n, m	C^*	QL		PSO		FA		NEH
			BRE	ARE	BRE	ARE	BRE	ARE	RE
Car1	11, 5	7038	0	0	0	0.05	0	0	0
Car2	13, 4	7166	0	0.71	0	2.58	0	1.29	2.93
Car3	12, 5	7312	1.19	1.91	1.19	2.34	0	1.86	3.12
Car4	14, 4	8003	0	1.12	0	1.12	0	0.33	0.39
Car5	10, 6	7720	0	0.61	0	1.33	0	0.59	1.49
Car6	8, 9	8505	0	0.94	0	1.17	0	0.57	5.43
Car7	7, 7	6590	0	0	0	0.36	0	0.04	0
Car8	8, 8	8366	0	0.31	0	0.74	0	0.28	2.37

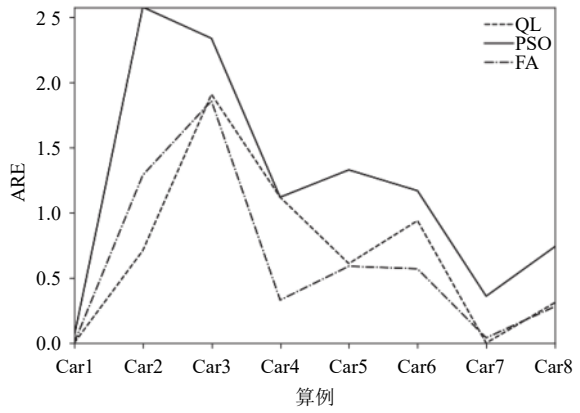

图 2 平均值比较

选取算例 Car4 算例来分析 QL 算法的收敛能力, 从图 3 中发现 QL 算法刚开始下降较快, 在中间一段时间内虽然两次陷入了局部最优值, 但最后都成功跳出了局部最优, 达到最优值.

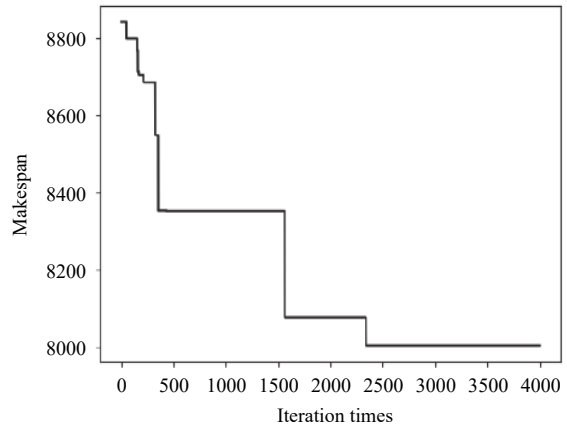

图 3 算例 Car4 最优值下降曲线图

表 2 给出了 QL 算法对 Reeves 例题 (Rec37、Rec39、Rec41 3 个算例的值不是最优值, 而是最优上界). 测试结果, 并与 PSO[25], GA[26]等知名算法比较, QL 算法的平均最优值最小, 表明整体上 QL 算法的寻优能力比 GA 和 PSO 算法都好.

表 2 Reeves 例题 BRE 测试结果

问题	n, m	C^*	QL	GA	PSO
Rec01	20, 5	1247	3.49	2.81	5.84
Rec03	20, 5	1109	2.07	1.89	2.43
Rec05	20, 5	1242	1.93	1.93	2.25
Rec07	20, 10	1566	1.6	1.15	2.87
Rec09	20, 10	1537	2.41	3.12	2.73
Rec11	20, 10	1431	4.82	3.91	7.75
Rec13	20, 15	1930	3.73	3.68	6.01
Rec15	20, 15	1950	2.87	2.21	5.53
Rec17	20, 15	1902	3.99	3.15	7.46
Rec19	30, 10	2093	3.92	4.01	8.36
Rec21	30, 10	2017	5.21	3.42	7.38
Rec23	30, 10	2011	5.22	3.83	8.25
Rec25	30, 15	2513	3.21	4.42	7.56
Rec27	30, 15	2373	4.26	4.93	8.51
Rec29	30, 15	2287	4.55	6.21	9.54
Rec31	50, 10	3045	4.2	6.17	11.32
Rec33	50, 10	3114	4.08	3.08	8.28
Rec35	50, 10	3277	1.1	1.46	5.76
Rec37	75, 20	4951	5.57	6.56	10.40
Rec39	75, 20	5087	4.34	6.39	9.53
Rec41	75, 20	4960	6.69	7.42	11.89
AVG			3.77	3.89	7.12

6 结束语

面对日益增长的大规模调度问题, 开发新型算法越来越重要. 本文提出了基于 QL 算法解决置换流水车间调度问题的算法, 通过对 Car 算例和 Reeves 算例等基准问题进行测试并与已有的算法进行比较, 评价了该算法的有效性. 结果表明, 该方法是解决置换流水车间调度问题的一种有效的方法.

参考文献

1 唐国春, 井彩霞. 现代排序论应用. 自然杂志, 2015, 37(2): 115–120.

2 Garey MR, Johnson DS, Sethi R. The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research, 1976, 1(2): 117–129. [doi: 10.1287/moor.1.2.117]

3 刘延风. 置换流水车间调度问题的几种智能算法[博士学位论文]. 西安: 西安电子科技大学, 2012.

4 Vallada E, Ruiz R. Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem. Omega, 2010, 38(1–2): 57–67. [doi: 10.1016/j.omega.2009.04.002]

5 El-Bouri A, Balakrishnan S, Popplewell N. A neural network to enhance local search in the permutation flowshop. Computers & Industrial Engineering, 2005, 49(1): 182–196.

6 Rajendran C, Ziegler H. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. European Journal of Operational Research, 2004, 155(2): 426–438. [doi: 10.1016/S0377-2217(02)00908-6]

7 张其亮, 陈永生. 一种新的混合粒子群算法求解置换流水车间调度问题. 计算机应用研究, 2012, 29(6): 2028–2030, 2034. [doi: 10.3969/j.issn.1001-3695.2012.06.006]

8 曹磊, 叶春明, 黄霞. 应用混沌烟花算法求解置换流水车间问题. 计算机应用与软件, 2016, 33(11): 188–192. [doi: 10.3969/j.issn.1000-386x.2016.11.044]

9 王大志, 刘士新, 郭希旺. 求解总拖期时间最小化流水车间调度问题的多智能体进化算法. 自动化学报, 2014, 40(3): 548–555.

10 Sutton RS, Barto AG. Reinforcement learning: An introduction. IEEE Transactions on Neural Networks, 1998, 9(5): 1054.

11 Mitchell TM, Carbonell JG, Michalski RS. Machine learning: A guide to current research. Boston, MA, USA: Springer, 1997.

12 杨文臣, 张轮, Zhu F. 多智能体强化学习在城市交通网络信号控制方法中的应用综述. 计算机应用研究, 2018, 35(6): 1613–1618. [doi: 10.3969/j.issn.1001-3695.2018.06.003]

13 孟伟, 洪炳熔, 韩学东. 强化学习在机器人足球比赛中的应用. 计算机应用研究, 2002, 19(6): 79–81. [doi: 10.3969/j.issn.1001-3695.2002.06.026]

14 王斐, 齐欢, 周星群, 等. 基于多源信息融合的协作机器人演示编程及优化方法. 机器人, 2018, 40(4): 551–559.

15 Wang YC, Usher JM. Learning policies for single machine job dispatching. Robotics and Computer-Integrated Manufacturing, 2004, 20(6): 553–562. [doi: 10.1016/j.rcim.2004.07.003]

16 Wang YC, Usher JM. Application of reinforcement learning for agent-based production scheduling. Engineering Applications of Artificial Intelligence, 2005, 18(1): 73–82. [doi: 10.1016/j.engappai.2004.08.018]

17 Zhang ZC, Wang WP, Zhong SY, *et al*. Flow shop scheduling with reinforcement learning. Asia-Pacific Journal of Operational Research, 2013, 30(5): 1350014. [doi: 10.1142/S0217595913500140]

18 潘燕春, 周泓, 冯允成, 等. 同顺序 Flow shop 问题的一种遗传强化学习算法. 系统工程理论与实践, 2007, 27(9): 115–122. [doi: 10.3321/j.issn:1000-6788.2007.09.015]

19 潘燕春, 周泓. Job-shop 排序问题的遗传强化学习算法. 计算机工程, 2009, 35(16): 25–28. [doi: 10.3969/j.issn.1000-3428.2009.16.009]

20 王超, 郭静, 包振强. 改进的 Q 学习算法在作业车间调度中的应用. 计算机应用, 2008, 28(12): 3268–3270.

21 王福才, 周鲁苹. 混合精英策略的元胞多目标遗传算法及其应用. 电子学报, 2016, 44(3): 709–717. [doi: 10.3969/j.issn.0372-2112.2016.03.032]

22 王凌. 车间调度及其遗传算法. 北京: 清华大学出版社, 2003.

23 胡文伟, 胡建强, 李湛, 等. 基于强化学习算法的自适应配对交易模型. 管理科学, 2017, 30(2): 148–160. [doi: 10.3969/j.issn.1672-0334.2017.02.012]

24 Tsitsiklis JN. Asynchronous stochastic approximation and Q-learning. Proceedings of the 32nd IEEE Conference on Decision and Control. San Antonio, TX, USA. 1994. [doi: 10.1109/CDC.1993.325119]

25 刘长平, 叶春明. 置换流水车间调度问题的萤火虫算法求解. 工业工程与管理, 2012, 17(3): 56–59, 65. [doi: 10.3969/j.issn.1007-5429.2012.03.010]

26 Yuan K, Hennequin S, Wang XJ, *et al*. A new heuristic-em for permutation flowshop scheduling. IFAC Proceedings Volumes, 2006, 39(3): 33–38.