A Reinforcement Learning Method to Scheduling Problem of Steel Production Process

Fang Guo\textsuperscript{1*}, Yongqiang Li\textsuperscript{1}, Ao Liu\textsuperscript{1} and Zhan Liu\textsuperscript{1}

\textsuperscript{1}Graduate Management Team, Naval Aviation University, Yantai, Shandong, 264001, China

\textsuperscript{*}Corresponding author’s e-mail: guofang575856@163.com

Abstract. In this paper, a reinforcement learning method is utilized to solve the steel production scheduling problem. Based on characteristics of steel production processing, the model of hybrid flow-shop scheduling problem is constructed. Then the model is attributed to a Markov Decision Process, and corresponding states, actions, reward function are put forward. When trading off the exploration and exploitation, an improved $\varepsilon$-greedy policy is designed. Finally, this hybrid flow-shop scheduling model based on reinforcement learning is applied to the scheduling example of steel production processing. Compared to genetic algorithm, the reinforcement learning method gets the better result.

1. Introduction

The core processes of steel production include four main stages: steelmaking, refining, casting and rolling. The relationship between stages is sequential, continuous and closely connected. The interruption of the stages will lead to unnecessary losses [1]. From the beginning of steelmaking to the storage of rolled products, the overall optimization of all processes is emphasized in steel production. To achieve the goal, the actual production needs not only the optimal production scheduling results, but also the dynamic, random and real-time requirements. Usually, the scheduling problem in steel production process is attributed to hybrid flow-shop scheduling problem.

The hybrid flow-shop scheduling problem has been researched widely. The research algorithms of hybrid flow-shop is abundant, which, generally speaking, can be divided into three categories: exact algorithm [2,3], heuristic algorithm [4] and intelligent algorithm [5,6]. The exact algorithm can obtain the optimal solution theoretically, its computation time, however, is usually unacceptable, so the algorithm is generally only suitable for solving small-scale problems. Heuristic algorithms, which is usually based on specific heuristic rules, can quickly get the solution of the problem, but it is difficult to guarantee the quality of the solution. With the development of computational intelligence, lots of intelligent optimization methods have been proposed and effectively solved hybrid flow-shop scheduling problem, which make them widely applied.

Since put forward, the reinforcement learning, being considered as an important approach to general artificial intelligence, is utilized in an array of fields, such as games [7], robot control [8,9], parameter optimization [10], machine vision [11].

Rare literatures, however, are found in hybrid flow-shop scheduling problem based on reinforcement learning. A reinforcement method, in this paper, is applied to solve the steel production processes scheduling problem.
2. Description of steel production process

In this section, the scheduling problem of steel production process is described as a hybrid flow-shop scheduling problem, whose model is established correspondingly.

The processes of steel production mainly include four stages: steelmaking, refining, casting and rolling. Supposed there are $m_1, m_2, m_3, m_4$ machines in each stage respectively. During the steel production process, each work piece needs to be machined at four stage in the deterministic sequence, the work piece can be worked in any one machine in each stage. The flow chart of the scheduling problem of steel production process is illustrated in figure 1. Based on the above analysis, the steel production process can be abstracted as a hybrid flow-shop scheduling problem with four stages.

Suppose that $J_i$ ($i=1, 2, ..., n$) is the work pieces to be machined, where $n$ is the total counts of work pieces. $m_j$ ($j=1, 2, 3, 4$) is the count of machines in each stage. $t_{i,j,l}$ is the working time of the work piece $J_i$ at stage $j$, machine $l$. ($j=1, 2, 3$). $AT_{i,j,l}$ is the arrival time. Correspondingly, $ST_{i,j,l}$ and $ET_{i,j,l}$ are the starting and ending time of work piece $J_i$ machined at stage $j$, machine $l$. $BT_{j,l}$ and $FT_{j,l}$ are the time when the machine $l$ at stage $j$ begins to work and finishes to. Then the model of the scheduling problem of steel production process is.

$$\min \max \{ ET_{i,j,l} \} \quad i=1, 2, ..., n; \quad j=1, 2, ..., 4; \quad l=1, 2, ..., m_k \quad \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^{m_j} y_{i,j,l} = 1 \quad i=1, 2, ..., n; \quad j=1, 2, 3, 4 \quad \quad (2)$$

$y_{i,j,l} = 1$ if work piece $J_i$ is worked at stage $j$, machine $l$, else $y_{i,j,l} = 0$

$$ST_{i,j,l}=\max \{ AT_{i,j,l}, FT_{j,l} \} \quad i=1, 2, ..., n; \quad j=1, 2, 3, 4; \quad l=1, 2, ..., m_j \quad \quad (3)$$

$$ET_{i,j,l}=ST_{i,j,l}+t_{i,j,l} \quad i=1, 2, ..., n; \quad j=1, 2, 3, 4; \quad l=1, 2, ..., m_j \quad \quad (4)$$

$$BT_{j,l}=ST_{i,j,l} \quad j=1, 2, 3, 4; \quad l=1, 2, ..., m_j \quad \quad (5)$$

if work piece $J_i$ is the first one machined at stage $j$, machine $l$ and without interruption.

$$FT_{j,l}=BT_{j,l}+t_{i,j,l} \quad \quad (6)$$

if work piece $J_i$ is worked at stage $j$, machine $l$, together with equation (5), equation (6) reveals the time when the machine $l$ at stage $j$ starts to work and stop to.

3. The reinforcement learning model of steel production process scheduling

In this section, the model built in section 2 is attributed to Markov Decision Process, and corresponding parameters are designed.

A Markov Decision Process is often expressed as a tuple $\left(S, A, T, R\right)$, where $S$ and $A$ denote a set of states and actions, $T: S \times A \rightarrow \{0,1\}$ is the state transition probability distribution after the agent
taking action $a$ in state $s$, and $R: S \times A \times S' \rightarrow R$ is the reward function on taking action $a$ and transferring to state $s'$ in state $s$.

In the scheduling problem of steel production process, the state is donated as a tuple $(\text{stage}, \text{work piece})$, the action is the machines in next stage. When choosing actions, an improved $\varepsilon$-greedy policy is adopted, which is shown as equation (7), where $\beta$ is a small decimal, $\varepsilon$ is the times of iterations, $\varepsilon_0$ is the initial value, which makes the agent easy to explore the action with non-optimal $Q$-value sufficiently at the beginning of the study and select the action with optimal $Q$-value with larger and larger probability in the later stage of learning. The reward function is defined as equation (8), in which $c_{-t_{i,j,l}}=FT_{j,l}-BT_{j,l}$ is the time of work piece $l$ before finishing being machined on machine $l$ at stage $j$, $\omega$ and $b$ are positive constants to make sure the reward function negatively correlated to $c_{-t_{i,j,l}}$.

$$\varepsilon = \varepsilon_0 - \beta \varepsilon$$  \hspace{1cm} (7)

$$r(s,a) = -\omega \times c_{-t_{i,j,l}} + b$$  \hspace{1cm} (8)

The iteration equation of $Q$-value is represented as

$$Q(s,a) = Q(s,a) + \alpha (r + \gamma \max_a Q(s',a') - Q(s,a))$$  \hspace{1cm} (9)

where $\alpha$ is the learning rate.

4. Case validation and result

In this section, an example of scheduling problem of steel production process is used to validate the quality of the reinforcement learning method.

It is assumed that there are 12 work pieces in a steel production process, labelled with 1 to 12, and there are 3,3,2,2 parallel machines in each stage. The machining time of each work piece in each machine in the four stage is shown in table 1.

| Table 1. Machining time of each work piece in each machine |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | J1   | J2   | J3   | J4   | J5   | J6   | J7   | J8   | J9   | J10  | J11  | J12  |
| steelmaking     |      |      |      |      |      |      |      |      |      |      |      |      |
| M1              | 45   | 45   | 50   | 50   | 45   | 45   | 47   | 50   | 48   | 45   | 46   | 48   |
| M2              | 48   | 50   | 45   | 48   | 46   | 45   | 50   | 45   | 46   | 47   | 50   | 50   |
| M3              | 50   | 45   | 46   | 48   | 45   | 47   | 48   | 46   | 47   | 45   | 47   |      |
| refining        |      |      |      |      |      |      |      |      |      |      |      |      |
| M1              | 35   | 35   | 35   | 34   | 30   | 31   | 32   | 33   | 33   | 34   | 35   |      |
| M2              | 35   | 36   | 36   | 35   | 35   | 35   | 30   | 30   | 33   | 33   | 30   | 31   |
| M3              | 30   | 35   | 36   | 35   | 50   | 50   | 34   | 30   | 30   | 30   | 35   | 35   |
| casting         |      |      |      |      |      |      |      |      |      |      |      |      |
| M1              | 30   | 35   | 31   | 32   | 34   | 33   | 35   | 34   | 35   | 30   | 32   |      |
| M2              | 35   | 34   | 34   | 33   | 32   | 32   | 31   | 30   | 30   | 34   | 35   | 30   |
| rolling         |      |      |      |      |      |      |      |      |      |      |      |      |
| M1              | 25   | 25   | 30   | 27   | 28   | 30   | 29   | 24   | 25   | 32   | 31   | 25   |
| M2              | 26   | 30   | 31   | 31   | 31   | 26   | 25   | 27   | 25   | 26   | 25   | 30   |

During the steel production process scheduling, the transferring time of the work piece between two stages is ignored. The parameters, in the reward function, are set as $\omega=4$ and $b=200$, and the parameters in the improved $\varepsilon$-greedy are set as $\varepsilon_0=0.1$, $\beta=0.01$, which is adopted when the iteration is less than 100. To select the action with large reward, the greedy policy is utilized when iteration is beyond 100.

On the basis of section 2 and 3, in each time, 100 stochastic work piece sequences are chosen randomly and for any sequence, 200 episodes are conducted. The minimum of 100 stochastic sequences is set as the optimal value of this time. 10 times are executed in this paper, whose results are compared with one result calculated by genetic algorithm [12]. Details are shown in table 2. And one of the optimal
scheduling time is 307 minutes, and the corresponding initial sequence is [6, 4, 11, 7, 1, 2, 8, 3, 5, 12, 10, 9], whose Gantt chart is illustrated in figure 2.

Table 2. The comparation of optimal results of 10 times

|        | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| genetic algorithm | 347 |     |     |     |     |     |     |     |     |     |
| reinforcement learning | 307 | 311 | 310 | 307 | 309 | 309 | 311 | 307 | 308 | 307 |

Figure 2. Gantt chart of one of the optimal scheduling.

In figure 2, the y-label is the location where the work piece is machined, for instance, ‘S1_2’ reveals the second machine in stage 1. The number in the chart means the stage where the work piece lies and its tag, such as ‘2-12’ donates the 12th work piece is worked in the second stage.

As table 2 shows, the reinforcement learning method is superior to the genetic algorithm in solving the scheduling problem of steel production process.

5. Conclusion

The scheduling problem of steel production process is addressed in this paper. The literature investigation reveals the universality of hybrid flow-shop scheduling problem and the reinforcement learning.

Firstly, the scheduling model of steel production process is constructed, and is attributed to the Markov Decision Process, for which the special states, actions and reward function are designed and the improved \( \varepsilon \)-greedy is designed to trade-off the exploration and exploitation.

The contribution of this paper mainly lies in utilizing the reinforcement learning to solve the scheduling problem in steel production process. The result achieved by this method is not the optimal one, but is better than the result obtained by the genetic algorithm.

In the future, other reinforcement learning method will be considered to solve the scheduling problem of steel production process. Besides, improved intelligent algorithms may be researched to solve the relative problems.

References

[1] Lixin Tang, Zihou Yang. (1996) Research on Framework of Steelmaking-Continuous Casting Production Planning and Scheduling. Journal of Northeastern (Natural Science). 17(6), 664-667.

[2] HAOUAR IM, HIDR IL, GHARB IA. (2006) Optimal Scheduling of a Two-stage Hybrid Flow Shop. Mathematical Methods of Operations Research. 64(1) : 107 - 124.
[3] XIE J X, WANG X J. (2005) Complexity and Algorithms for Two-stage Flexible Flow-shop Scheduling with Availability Constraints. Computers & Mathematics with Applications. 50(10 - 12): 1629 - 1638.

[4] RUIZ R, VAZQUEZ RODRIQUEZ J A. (2010) The Hybrid Flow Shop Scheduling Problem. European J of Operational Research. 205 (1): 1 - 18.

[5] BELKAD IK, GOURGAND M, BENYETTOU M, et al. (2006) Sequential and Parallel Genetic algorithm for the Hybrid Flow Shop Scheduling Problem. J of Applied Science. 6(4): 775 - 8

[6] NADER IB, ZAND IEH M, KHALEGI GHOSHE BALAGH A, et al. An Improved Simulated Annealing for Hybrid Flow-shop with Sequence-dependent Setup and Transportation Times to Minimize Total Completion Time and Total Tardiness. Expert Systems with Applications. 2009, 36 (6): 9625 - 9633

[7] Mnih V, Kavukcuoglu K, Silver D, et al. (2013) Playing atari with deep reinforcement learning. Proceedings of Workshops at the 26th Neural Information Processing System. Lake Tahoe, US. 2013:201-220

[8] Lillicrap T P, Hunt J J, Pritzel A, et al. (2016) Continuous control with deep reinforcement learning. Computer Science, 8(6): A187

[9] Duan Y, Chen X, Houthooft R, et al. (2016) Benchmarking deep reinforcement learning for continuous control. Proceedings of the 32nd International Conference on Machine Learning. New York, USA, 2016: 1329-1338

[10] Hansen S. (2016) Using deep q-learning to control optimization hyperparameters. arXiv preprint arXiv:1602.04062, 2016

[11] Oh J, Guo X, Lee H, et al. Action-conditional video prediction using deep networks in atari games. Proceedings of the Neural Information Processing Systems. Montreal, Canada, 2015: 2863-2871

[12] JianShuang Cui, TieKe Li, WenXin Zhang. (2005) Hybrid flow-shop scheduling model and its genetic algorithm. Journal of University of Science and Technology Beijing, 27(5):623–626