強化学習を用いたジョブショップ・スケジューリング問題の一解法

荒井　誠　宮澤　武　関谷　昌平

A Solution of Job-Shop Scheduling Problem Using Reinforcement Learning

Makoto ARAI　Takeshi MIYASAWA　Shohei SEKIYA

Abstract- This paper describes a solution of Job-Shop scheduling problem developed based on new algorithm. We used Reinforcement Learning to the pursuit of this good effect. That is, it expresses as multi-agents which the jobs are the set of agents that are moved on Gant chart. To obtain effective move as the agents, we give the reward according to evaluation of movement and we tune the parameters of the Learning using Q-learning technique. And scheduling that shows a moreover good effect by this processing is searched. In the paper, we mention about some experimental results and examine the applicability of the proposed methodology.

Keyword : Job-Shop Scheduling Problem, Reinforcement Learning

１．緒　言

ジョブショップスケジューリング問題(Job-Shop Scheduling Problem :以下JSP)[1]は、ある製品の製造工程において複数の仕事を複数の機械に割り当てる組合せ最適化問題の一つである。本研究は、JSPの解法に関して、新しい手法を提案し、その適用可能性を議論する。一般にJSPは、加工時の滞留時間や総所要時間の最小化を目指すものであるが、対象の仕事や機械が増えるにしたがい、列挙法による対応や解析的に最適化を行うことが困難となる。そこで、提案するアプローチ手法では、各々の仕事を１つのエージェントとするマルチエージェントとして問題空間を設定する。各々のエージェントは、ガントチャートを想定した２次元空間内の移動位置によって与えられる報酬（評価値）から、最も高い報酬を得る位置を探索する強化学習[2]を行う。これによって、問題の規模や仕事、機械の種類に依存しない解探索が可能となる。本論文は問題の解決方法を論じ、アルゴリズムの有効性を示すために行ったいくつかの数値実験の結果を示す。

２．ジョブショップスケジューリング問題

2.1　JSP の概要

生産現場では、所定の期間に対象となる製品の種類と数量が定められると、これらを加工するにあたって必要な生産設備（工作機械・加工装置・工具…etc.）、生産に必要な作業員の確保、作業員の行う仕事の分担を行う。さらに、生産工程が実施されるにあたり、それらを詳細に取り扱った時間日程すなわちスケジュールが必要となる。

スケジューリング（Scheduling）問題とは、このように計画に基づいて一連の生産設備に割り当てた一定期間内の仕事に対して、作業者または設備別に作業の開始・終了時刻、あるいは作業を行う順序を決定することである。これは順序付け問題とも呼ばれ、サービスを持っている対象群にサービスを与える順序を定める問題で、一般に製造工場内をモデルとして取り扱う。

2.2　仕事及び機械の順序付け

ここで、n 個の仕事と m 台の機械が与えられ、各仕事を加工する機械の順序が決められているときに、各機械上での仕事の処理順序を定める問題を想定する。この場合、n 個の仕事 J があるとすると、

$$J_q = J_1, J_2, J_3, \Lambda\ J_n \tag{1}$$

となる。また、仕事を処理する m 台の機械 M を

$$M_r = M_1, M_2, M_3, \mathrm{K}\ , M_m \tag{2}$$

と表現する。この場合での各仕事 J_q を加工する機械の順序 M_{ri}（i=1,2,…m）を「技術的順序」といい、

*　釧路高専機械工学科
**　(株)　ジェーシー

$$T_q = (qr_1, qr_2, qr_3, \Lambda\ qr_m) \tag{3}$$

と表され、T_q を第 q 行とする $n \times m$ 行列 T を技術的順序行列という。

また、加工時間は各仕事 J_q の各機械 M_r 上での処理の時間 p_{qr} で表し、加工時間列は

$$P_q = (p_{q1}, p_{q2}, \Lambda\ , p_{qm}) \tag{4}$$

となる。このときの P_q を第 q 行とする $n \times m$ 行列 P を加工時間行列という。

各機械 M_r 上での仕事 J_q の処理を「作業」といい、O_{qr} で表す。そして、各機械 M_r 上での仕事の処理順序 J_{qj}（j=1,2,…,n）を機械　Mr 上の「仕事順」といい、

$$S_r = (rq_1, rq_2, rq_3, \Lambda\ rq_n) \tag{5}$$

と表す。

2.3　JSP の分類

JSP はジョブ（仕事）の到着の仕方によって「静的」と「動的」に分けられる。ジョブの到着時刻に関する情報が既知である場合は「静的」であり、ジョブの到着の仕方に確率的要素が含まれる場合など、情報が前もってわかっていないことを「動的」であるという。

また、取り扱うスケジューリングの性格によって「確定的」と「確率的」に分類される。前者はジョブの到着時刻のほかに、納期、技術的順序、加工時間、機械の処理能力などに関する情報が全て既知である場合は、確定的に分類される。一方、後者はこれらの事柄のうち、一つでも確率的要素を含む場合である。これらの関係を図１に示す。

本研究では、静的で確定的なジョブショップスケジューリング問題を想定している。

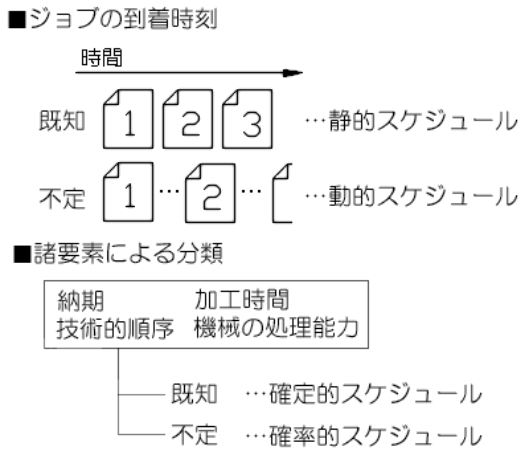

図１．スケジュールの分類

2.4　評価と制約条件

スケジュールを評価するためには、目的関数として「評価基準」を設定する必要があり、よく使用されるものとして、最大および平均の滞留時間、総所要時間、最大および平均の納期遅れなどがある。通常これらの評価基準は時間を単位としており、この最小化が図られる。その他の評価基準として、総加工費用の最小化や各種の機械稼働率などが用いられる。

また、処理の条件として、

・どの仕事も m 台の機械で一度だけ加工される

・各機械は同時に二つ以上の仕事は処理できない

・各仕事は一度に二つ以上の機械では処理できない

とするとき、n 個の仕事と m 台の機械に対する技術的順序行列と加工時間行列が既知であり、目的関数として全ての仕事を完了させるまでの時間（総所要時間）を設定すると、これを最小にするように各機械上での仕事の処理順序を定め、各作業の開始時刻を決定（スケジュール）するのが $n \times m$ の JSP である。

３．強化学習

強化学習とは、エージェントという制御対象が、与えられた環境を観測して行動を出力し、環境がどの状態にある時にどのような行動を出力するのが最適であるのかを学習するアルゴリズムである。エージェントは出力した行動が良い行動だったかを、環境から報酬という情報を得ることで学習する。

強化学習において代表的なものが *Q-Learning* である。エージェントは学習のたびに期待値 $Q(s,a)$ を更新する。s は環境の状態を、a は行動を表す。すなわち、どの環境がどの状態にある時に、どの行動をとるのが最善であるのかを記憶するものである。*Q-Learning* の流れを以下に示す。

1. **環境の状態 s_t を観測する。**
2. **任意の探査戦略に従って、行動 a_t を出力する。**
3. **環境から報酬 r_t を受け取る。**
4. **状態遷移後の環境の状態 s_{t+1} を観測する。**
5. **式 (6) の更新式に従って $Q(s,a)$ を更新する。**
6. **時間 t を $t+1$ へ進め、1. へ戻る。**

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot Q(s_t, a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1}, a)] \tag{6}$$
$$(0 < \alpha < 1,\ \ 0 < \gamma < 1)$$

ここで、α は学習率といい、行動の結果をどれだけ期待値に反映するかを表す。また、γ は割引率であり、過去の報酬を割引く割合を表す。過去の報酬を割引く理由は、実環境では時間の経過とともに状態が変化することや、エージェントの故障などが想定され、このような状況において、全て同じ重みで評価することは

妥当ではないからである。

行動選択（探査戦略）にはいくつかの方法が提唱されているが、ここでは ε-*greedy* 戦略を適用する。これは乱率 ε の確率でランダムに行動し、それ以外は最大の期待値を持つ行動を出力するものである。

4. JSP への適用

4.1　アプローチ手法

従来の計算的手法によるスケジューリング問題の解法は、スケジューリング問題に付随してくる様々な要素を計算パラメータとして捉え、最終的な目標を目的関数として作成し、その関数を最小化あるいは最大化として最適化するものであった。そうして最適化されたデータを基にガントチャートを作成し、出来上がったスケジュールを評価する。

しかし、このような手法は、求めたいスケジュールは設計者が決定し、その都度目的関数を書き直さなければならない。大規模な問題になれば、目的関数の設定の難しさや煩雑さが生じ、最終的には最適解を求めることが不可能となる。一方、強化学習を用いれば、問題の規模にかかわらず、一つのアルゴリズムで解くことが可能である。また、評価基準によって項目分けして値を任意に設定できるようにしておけば、その評価基準を満たすスケジュールが求められたときに、報酬を与えることで、解となるスケジュールに近づいていく学習が行われる。

これらを実現するために、本研究ではガントチャート上でスケジューリングを行う方法を提案する。すなわち、適当に配置した各作業を移動や並べ替えを行いながら最適なスケジュールへと導いていく方法で、この過程に強化学習を導入する。本研究では、図２に示す縦軸に仕事、横軸に時間をとり、各作業が処理される時間区間を表現するガントチャートを用い、これを強化学習エージェントに自動生成させるシステムを構築する。その際、単一のエージェントに最適なシステムを追求させるのではなく、各作業をそれぞれ個々のエージェントとして周囲の状態を観測しながら、協調

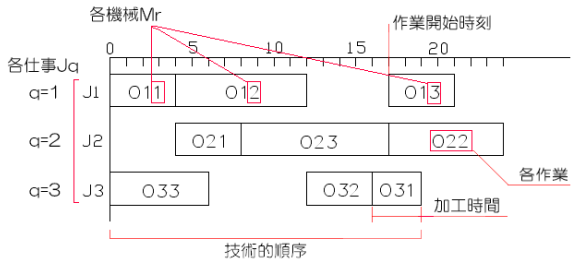

図２．ガントチャートの一例

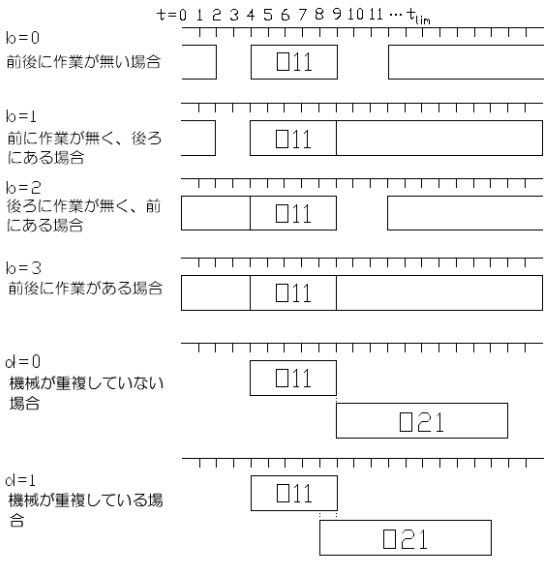

図３．環境の定義

的に行動と学習をするマルチエージェントとした。

4.2　環境

各エージェントはそれぞれ自分の環境を持つ。図 3 に示すように、環境は前後に他のエージェントが隣接しているか否かを b とし 4 パターン、これに同一番号の機械が重複しているか否かを d として加えた 8 パターンに、開始時間 t を考慮した状態 $s(b,d,t)$ と定義する。

4.3　行動

次に、各エージェントはどのような行動が可能なのかを設定する。本研究では、各エージェントが可能な行動を「前進」、「後退」と「停止」の３種類とし、１回の計算処理中にいずれか１つの行動をするものとした。すなわち、「前進」は時間を 1 単位進めること、「後退」はその逆であり、「停止」はその場に留まることを意味している。当然、状態 b によっては、不可能な行動も存在するが、ここで注意しなければならないのは、それによって行動の選択肢を狭めてはならないことである。例えば b=1 では「前進」することはできないが、ここで「前進」の選択肢を排除せず、「前進」を行うことを想定する。この場合、エージェント

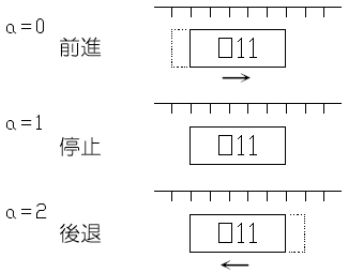

図４．エージェントの行動

がこの状態で「前進」を出力しても、環境は何も状態を遷移させなかったように見える。しかし、この時は「遷移しなかった状態」に遷移しているとし、その「遷移しなかった状態」を考慮して期待値（Q 値）を更新する。

4.3　報酬

「学習」を開始すると、各エージェントは「前進」「停止」「後退」を繰り返しながら期待値を更新していく。この過程で実行可能解が現れれば、その解が設定した評価基準を満たしているかどうかを判断する。評価基準を満たしていれば、これを第一の解として、報酬基準量の報酬を与え、初期配置に戻る。また、評価基準を満たさなくとも初期配置には戻るが、報酬は与えられない。

エージェントはその後も学習を繰り返すが、さらに良質な解を発見した場合は、過去の報酬量＋報酬基準量の報酬を与える。それを次の解とする。これにより、エージェントはより高い報酬が与えられるところに行き着く性質を持つので、これを繰り返すことで最適解に至る。

5　数値実験

上記までのアルゴリズムの有効性検証のために、JSP解法のための強化学習シミュレータを開発した。その概観を図５に示す。システムは、通常のパーソナルコンピュータを用い、開発用プログラミング言語は Borland 社 C++Builder Ver 5.0 である。

初期条件として、仕事数 n、機械台数 m、技術的順序、加工時間、制限時間の順に記述したテキストファイルを事前に作成しておき、これを対象問題の入力データとして用いる。開発したシミュレータでは、入力データを基に、エージェントを初期配置する。初期配置は左詰めで技術的順序に従うものである。

次に、評価基準設定や強化学習パラメータを項目毎に入力する。評価基準設定は、総所要時間、最大滞留時間、平均滞留時間、総合計滞留時間の各パラメータを選択入力でき、強化学習パラメータでは初期期待値、学習率、割引率、乱率、報酬基準量を直接あるいはスライドバーによって入力できるようにした。エージェントはこれらの入力された評価基準を満たすように学習を進める。

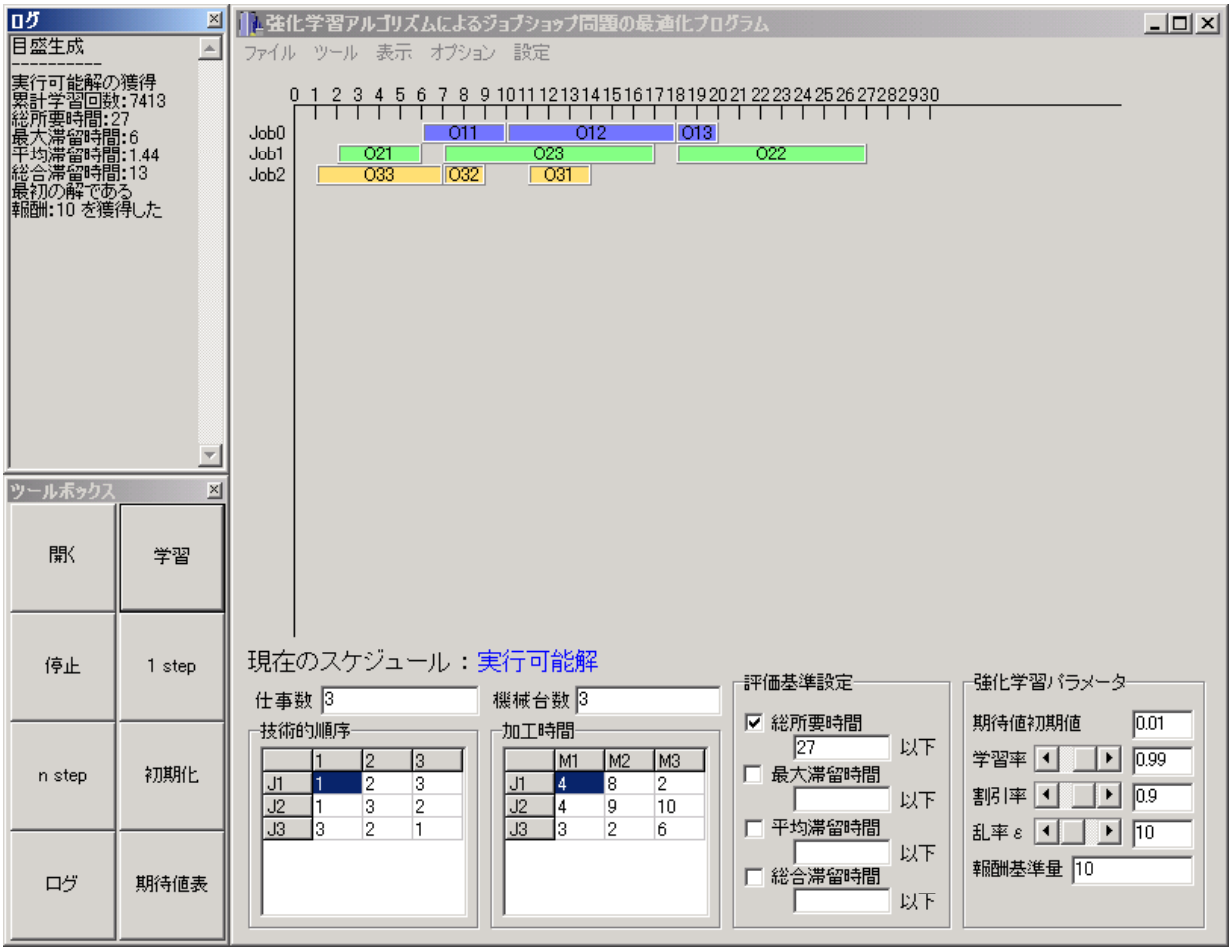

図５．強化学習シミュレータの概観

６　実　験

6.1　実験条件

実験に用いる入力データは

仕事数 n　3

機械台数 m　3

技術的順序　仕事 1：01,02,03

仕事 2：01,03,02

仕事 3：03,02,01

加工時間　仕事 1：04,08,02

仕事 2：04,09,10

仕事 3：03,02,06

制限時間　30

である。

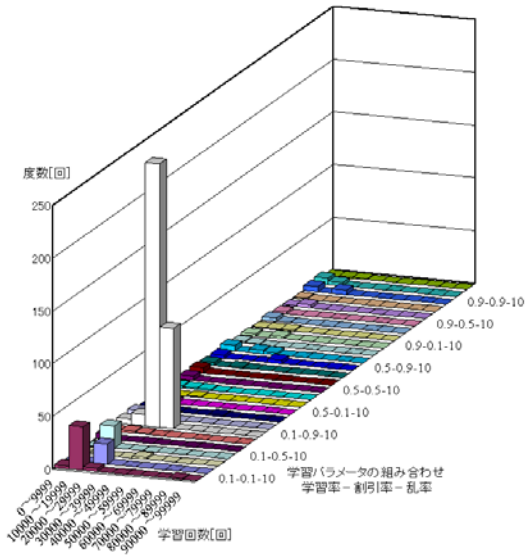

図６.学習パラメータの影響（報酬獲得解）

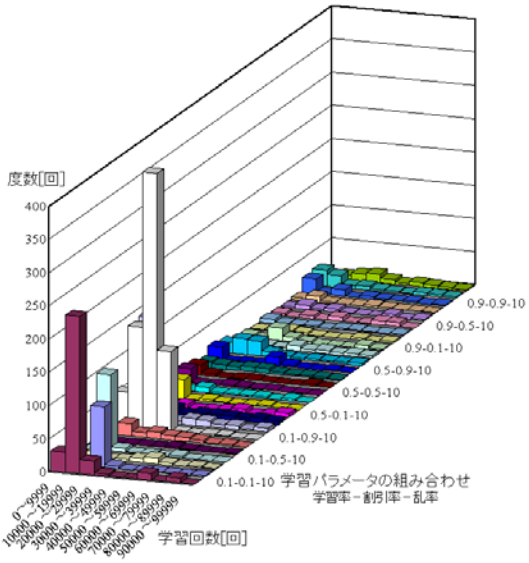

図７．学習パラメターの影響（実行可能解数）

評価基準は

総所要時間　：30 時間以下

最大滞留時間：10 時間以下

平均滞留時間：5 時間以下

とした。

6.2　実験１

最初に、強化学習パラメータの影響を調べるため、100000 回の学習までに、どのように解の出現の状態が変化するかを観測した。強化学習パラメータは

期待値初期値：1.0

学習率 α　：0.1 , 0.5 , 0.9

割引率 γ　：0.1 , 0.5 , 0.9

乱率 ε　：10、50、90

報酬基準量　：100

とした。0～100000 回まで、10000 回毎の報酬を獲得した解の数を図６に、実行可能解の出現数を図７に示す。結果から、共に α=0.1, γ=0.9, ε=10 が最も効率よく解を獲得していることを示している。

6.3　実験２

上記実験で良好な結果を示した学習パラメータを用いて、期待値初期値を 0.01,0.1,1、報酬基準量を 10,100,1000 と推移させた場合の獲得状況も観察した。その結果を図８、図９に示す。図から期待値初期値は 0.1 で報酬基準量は 10 が最適であると考えられた。これは、この設定が 100000 回という長いスパンでも期待値が発散せずに報酬を獲得し続けたことと、実行可能解の中に含まれる報酬獲得解の数が多かったことによる。また、図５の状態での評価基準値（総所要時間、滞留時間）がどのように推移したかを図１０に、求められた最適スケジュールを図１１に示す。

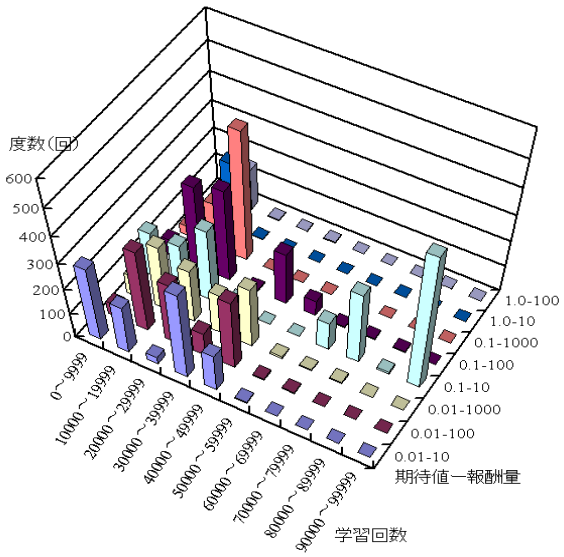

図８．期待値初期値と報酬基準量の影響（報酬獲得数）

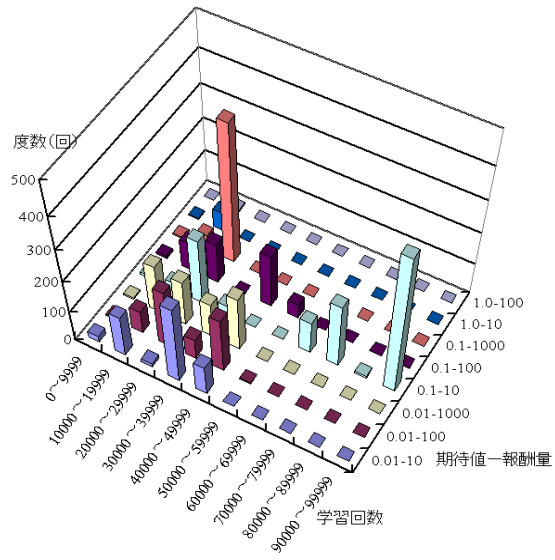

図９．期待値初期値と報酬基準量の影響（実行可能解）

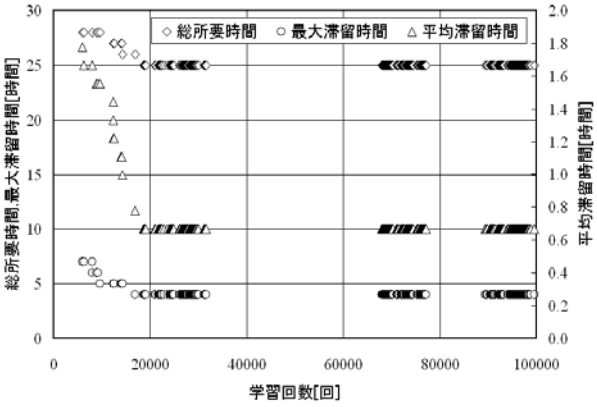

図 10．評価基準の推移

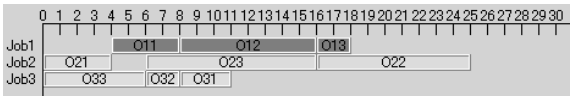

図 11．獲得した最適解

7 結　言

本研究では、生産システムでの自動加工を実施するうえで、大きな問題となるジョブショップ問題の解法について、新しい手法を提案した。提案した手法は強化学習を用い、単一のエージェントが問題を最適化する手法とは別に、各作業をエージェントとする協調的なマルチエージェントとして問題の枠組みを設定し、ガントチャート上での最適化を図ることで、問題の解を得るものである。考案したアルゴリズムの有効性を検証のために、シュミュレータシステムを構築し、数値実験を行った。その結果から次のことが結言できる。

1. 規模の小さな問題に対しては、パラメータを適切に設定することで十分な解が得られた。
2. 規模の大きな問題に対しても、パラメータを適切に設定することで、対応していくことが期待できる。その際は、報酬の与え方や、環境の設定などに再考の余地があると考えられる。

以上のことから、提案するアプローチの JSP への適用可能性を確認出来た。

最後に、本研究で取り扱った実験例は、ジョブの到着時刻や仕事に関する情報が全て既知である静的で確定的な問題であったが、実際の生産現場では、仕事の割り込みなど動的な要素が多く、これに対応するためにアルゴリズムの検討とともに、その有効性を確認するための検証実験を継続的に行う予定である。

【参考文献】

(1)古川正志・荒井　誠・吉村　斎・浜　克己共著：システム工学,コロナ社

(2) Richard S.Sutton and Andrew G.Barto: Reinforcement Laerning, MIT Press