AI スケジューラの AI 品質評価

－ 強化学習のケーススタディ －

The AI Scheduler Evaluation by workshop sectional meeting of SQiP Workshop

- Fisibility studies for reinforced learning -

研究員：　平井　宣（株式会社ＩＨＩエスキューブ）
主査：　石川 冬樹（国立情報学研究所）
副主査：　栗田 太郎（ソニー株式会社）
　　徳本 晋（株式会社富士通研究所）

研究概要
本論文では， AI 活用による生産計画の探索手法を提案する．生産計画はデータボリューム，バリエーションが多く，人手での計画は難しくソフトウェアの活用が不可欠である．ところが，現状のソフトウェアは高額で，探索には，みなし計算が避けられない．また，性能や精度に問題を抱えることが多い．
本手法により，探索の効率化を図り，ローコードによる開発効率化に道筋をつけた．その結果，生産計画にまつわる構造的なコスト問題を解消し，変化への柔軟性を高めることに貢献できる．

1. はじめに

スマートファクトリーなど製造業でも AI の活用が始まっている．しかしながら，生産計画分野[1]では，計画変更や歩留り改善にまだ多くの課題を抱えている．その一つに生産計画システムの硬直化，肥大化の問題がある．
また，最も短い工期で，製造設備の最適な稼働率を確保するには，本来緻密に計算され，何度でも見直せることができればよいのであるが，いわゆる最適化問題[2]と呼ばれる総当たりや計算量の壁により，みなしで計算されることが多い．
総当たりの自動化や繰返し計算が柔軟な AI に着目し，先の課題解決につながるような AI スケジューラを考案した．
以下，本論文の構成を述べる．まず 2 章で現状分析と課題提起を行う．3 章， 4 章で解決策の提案と評価結果を示す．5 章で評価結果に対する考察を行い， 6 章で成果と将来の展望で結ぶ．

2. 解決すべき課題

2.1. スケジューリングとは

工場の生産管理において，製品を作るにあたり，生産量に対する工期と製造設備などのリソースがどのくらい必要なのかという生産計画が必要となる．この計画立案をスケジューリングと呼ぶ．計画法としては，古典的なジョンソン法[3]や，近年では JIT 計画法（ジャストインタイム）[4][5]が知られている．
スケジューリングは，まず製品の作る順序を決める．決まった順序に従って，工期やリソースの手配が定まっていくからである．
なお，スケジューリングを考えるうえで，必要な用語を以下に示す．（表 2-1）

2.2. 現状分析

スケジューリングソフトは，個別に開発されたものが多く，標準化されたアルゴリズムやライブラリは存在しない．このため，ソフトウェアは肥大化，ブラックボックス化され，運用保守に人が張り付く必要があるため，コストパフォーマンスに課題を抱えている．

また，スケジューリングとは，オーダ，タスクの総当たりであり，全ての組合せを計算するには膨大な時間とリソースが必要となる．例えば，表 2-2 の 6 つのタスクからなる生産計画の場合，タスクの順列は $_3P_3$=6 通りであるが，10 タスクになると $_{10}P_{10}$=3,628,800 通りにもなる．

さらに計画は見直しが生じることが多く，現実的には職人の勘と努力で立案され，人材育成にも課題がある．

表 2-1 スケジューリングの主な用語

用語	意味
オーダ	製品を作る単位．オーダで作る数量が定まる．
タスク	切削，加工，検査などオーダを構成する作業，WBS．タスクで作業時間が定まる．
リソース	タスクを遂行するための加工装置など．各リソースの設置台数で生産性が定まる．

表 2-2 6 タスクの生産計画

オーダ	生産数量	タスク 1	タスク 2
#1	10 個	5.0 h	5.0 h
#2	10 個	6.0 h	4.0 h
#3	10 個	4.0 h	3.0 h

2.3. 課題提起

先ほどのジョンソン法は，タスクの数に上限があり，JIT 計画法もアルゴリズムが確立されているわけでもない．AI を活用した興味深いソフトウェア製品や論文が多数存在する[6][7][8][9][10]が，利用者視点で調べると，学術的な内容にとどまっていたり，需要予測や学習データに基づいた計算支援であったりした．

そこで，本論文は以下の課題解決を示す．

・総当り計算におけるレスポンス改善
・ローコードアルゴリズムによるソフトウェア規模(ソースコードの量)の削減

3. 解決策の提案

3.1. 課題の解決方針

(1)強化学習の活用

総当たり計算の自動化や計算量の削減を実現するために強化学習を活用する．強化学習とは，エージェントのアクションに報酬をつけ，記録することで，最も高い報酬が得られる探索結果を求めることができる機械学習である[11] [12]．

また強化学習はニューラルネットワーク等でアクションの探索ルートを蓄積することができるので，同じ探索ルートであれば，都度計算する必要がない．

(2) スケジューリングモデルの考案

2 つの行列と演算式を使ってタスクの実行順序を算出する．1 つはローカル行列(A:$a_{i,j}$ (i,j)=(オーダ,タスク))．もう 1 つは，環境行列(E:$e_{r,i,j}$ (r,i,j)=(探索ルート,オーダ,タスク))である．

演算式 A⊗E は，オーダの順序を求める演算式である．この演算結果で探索ルート別の工期とリソース別の稼働率を算出することができる．工期と稼働率を算出する数式の説明は，本論文では割愛する．

3.2. ローカル行列

4 つのタスクからなるローカル行列の例を図 3-1 に示す.

$$\begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix} = a_{i,j} : [\text{作業時間}, \text{タスク順序}] = \begin{bmatrix} [1.0, null] & [0.5, null] \\ [0.5, null] & [0.1, null] \end{bmatrix}$$

図 3-1:4 タスクのローカル行列

3.3. 環境行列

環境行列は，エージェントがルート探索を行う座標となり，左下の(0,0)がスタート地点で，右上の(オーダ数,タスク数)がゴール地点となる(図 3-2)．この場合，探索されるルートは(0,0)→(1,0)→(0,1)→(1,1)と(0,0)→(0,1)→(1,0)→(1,1)の 2 つとなる．
強化学習のエージェントは，アクションに対する報酬を手掛かりに，先の 2 つのルートを推定できる．

この場合，環境行列は図 3-3 となる．要素の数値は，-1 が初期値，0～n の整数値はエージェントが通過した通過番号，-2 がタスクがない場合を示す値で，エージェントは通過できない座標を意味する．

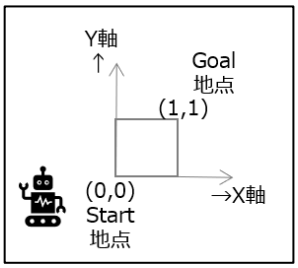

図 3-2:環境座標イメージ

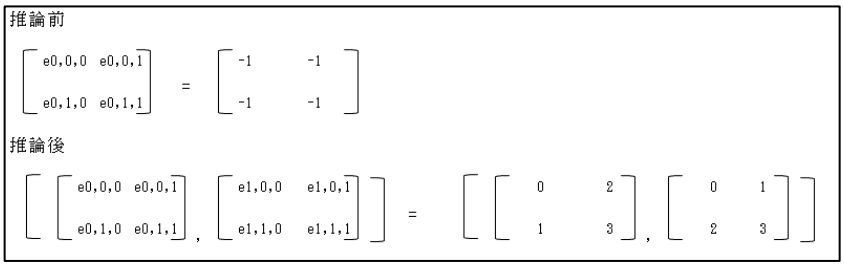

図 3-3:例に対する環境座標

3.4. 演算式によるタスク順序算出

行列演算式 A⊗E は，環境座標の $e_{r,i,j}$ をローカル座標の $a_{i,j}$ に合成する演算式である．

例に対する行列演算式 A⊗E の計算結果を図 3-4 に示す．赤字は演算式⊗により行列 A と行列 E の要素を合成した結果を示す．

$$A \otimes E = \begin{bmatrix} [1.0,null] & [0.5,null] \\ [0.5,null] & [0.1,null] \end{bmatrix} \otimes \left[\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \right]$$

$$= \left[\begin{bmatrix} [1.0,0] & [0.5,2] \\ [0.5,1] & [0.1,3] \end{bmatrix}, \begin{bmatrix} [1.0,0] & [0.5,1] \\ [0.5,2] & [0.1,3] \end{bmatrix} \right]$$

図 3-4:例に対する行列演算結果

4. 解決策の評価

4.1. 評価方針

まず，従来の計画法と計算結果を比較する．計算結果は，工期とリソース稼働率とした．計画の良し悪しを判断する KPI だからである．
算出方法は，以下の通り．
・ジョンソン法は公開されているアルゴリズムに基づいて手計算で算出した
・JIT 計画法は社内システムを参考にサンプルプログラムを作成[13][14]し算出した
・AI スケジューラは 3 章に基づいたサンプルプログラムを作成し算出した
・工期とリソース稼働率は，評価用にプログラムを開発し算出した．

次に，ローコードアルゴリズムかの評価は，サンプルプログラム総行数(以降 LOC)で比較する．

4.2. 評価データ

3 つのデータを準備した．データ 1 は，ジョンソン法が適用可能な評価データ（タスクが最大 2)．データ 2 は，表 4-1 に示す実業務を参考にして作成した評価データ．データ 3 は，表 4-2 に示す JIT 計画法の制約を示す評価データである．

なお，データ 2 でのオーダ別タスク別の作業時間は，各オーダの機種に一致するタスク一覧を取得し，数量×生産性÷多重度で算出する．

表 4-1　データ 2

オーダ		数量
機種	仕向け	
汎用	A発電	20
汎用	B工場	30
現地設置	C工場	30
半製品	Dプラント	50
半製品	Eプラント	50
汎用	F発電	20
汎用	G工場	30
現地設置	H工場	30
半製品	Iプラント	50
半製品	Jプラント	50
汎用	A発電	20
汎用	B工場	30
現地設置	C工場	30
半製品	Dプラント	50
半製品	Eプラント	50

機種	手番	タスク	多重度	生産性
汎用	0	切断・研磨	4	0.1
汎用	1	切削	2	0.1
汎用	2	組立・溶接	4	0.2
汎用	3	検査	1	0.1
現地設置	0	切断・研磨	4	0.1
現地設置	1	切削	2	0.2
現地設置	2	組立・溶接	4	0.1
現地設置	3	検査	1	0.1
半製品	0	切断・研磨	4	0.1
半製品	1	切削	2	0.1
半製品	2	組立・溶接	4	0.1
半製品	3	検査	1	0.1

表 4-2　データ 3

オーダ	タスク	作業時間
o1	t0	20.0
o1	t1	20.0
o2	t0	1.0
o2	t1	20.0

4.3. 評価方法

(1)ジョンソン法との比較

データ 1 による比較結果を表 4-3 に示す．

(2)JIT 計画法との比較

データ 2 による比較結果を表 4-4 に示す．

表 4-3:データ 1 による比較

	AI スケジューラ	ジョンソン法
工期(日)	12	12
稼働率(%)	58.3	58.3

表 4-4:データ 2 による比較

		AI スケジューラ	JIT 計画法
工期(日)		54.0	56.5
稼働率 (%)	切断・研磨	25.0	23.9
	切削	58.3	55.8
	組立・溶接	31.9	30.5
	検査	100.0	95.6
	平均	49.5	51.5

データ 3 による比較結果を表 4-5 に示す.

表 4-5:データ 3 による比較

		AI スケジューラ	JIT 計画法
工期(日)		41.0	60.0
稼働率(%)	t0	51.2	35.0
	t1	97.6	66.7
	平均	74.4	50.8

(3)JIT 計画法の LOC との比較

サンプルプログラムの LOC の比較結果を表 4-6 に示す.

表 4-6:LOC の内訳と比較

LOC の内訳	AI スケジューラ	JIT 計画法
AI 固有	201	-
JIT 固有	-	26
その他	149	149
合計	305	175

5. 考察

5.1. 得られた知見

(1)ジョンソン法との比較

工期，稼働状況とも同じ値で，ジョンソン法が示す最短工期，最も効率の良い稼働状況を示すことができた.

また，ジョンソン法はタスクが 2 つを超えると適用できない． AI スケジューラはオーダ，タスク数の制限がないので，この点において実用的であることを示している.

さらに，AI スケジューラの処理時間は数ミリ秒で，学習の収束性も示されており，精度，性能ともに問題は見られなかった(図 4-1).

なお，図 4-2 の座標中の番号は，エージェントが示した探索ルートである．オーダに対するタスクの順列が守られていることがわかる.

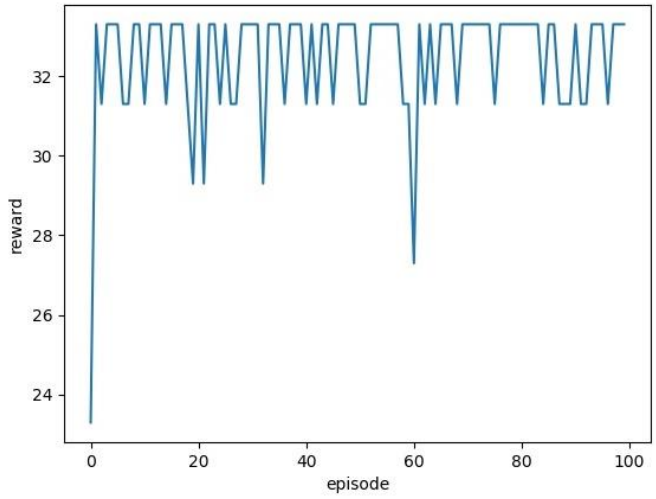

図 4-1:学習の収束性

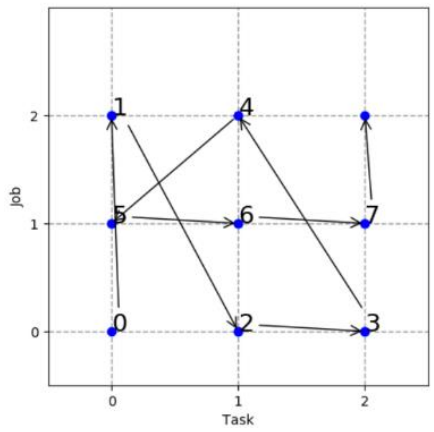

図 4-2:環境座標の座標リスト

(2)JIT 計画法との比較

AI スケジューラは，短い工期で，リソースの各稼働率も，高い．

また，タスク数が多い場合でも，数秒以内に算出でき，実用的に問題ないことが分かった．環境行列の行数，列数が同じ部分は計算結果が再利用できるので，高速化の効果が出ている．

(3)JIT 計画法の LOC との比較

表 4-6 より LOC は JIT 計画法が小さい．ところが，表 4-5 を見ると，JIT 計画法のサンプルプログラムの LOC はさらに増やす必要がある．

一方 AI スケジューラは様々な生産計画に対し，行列数と行列演算式は変わらないので，LOC は変化しない．

評価をさらに進める必要はあるが，AI スケジューラの LOC が増加せず収束していく可能性は高いといえる．

5.2. 上記考察による AI スケジューラの有効性について

強化学習から得られた生産計画は，工期，稼働率を見ると，他の方法と比較しても有効であることが分かった．また，実用に耐えうる性能であることと，複数の探索ルートに対し，工期や稼働率は評価できる値を示していた．

また，行列演算式は，並列処理が可能なため，さらなる高速化の実装余地がある．

行列と演算式を組み合わせることで，ローコードなアルゴリズムの可能性が期待できるといえる．

5.3. AI スケジューラの課題

実際のスケジューリングは様々な制約条件が追加される．例えば，リソースに割り当てる工員の休暇，残業，資格など，これらは総当り計算の過程で，考慮が必要となる．今回は分かりやすいケースに限定した．

また，２つの行列に分離して推論する方法は，スケジューリング問題に類似するもの（ナースのシフト表やシステム開発の開発計画など），他の最適化問題（配送計画，ルート検索など）にも応用できる可能性があるが，解決策の検討では除外している．

今回は強化学習を使って，組合せ計算の効率化，自動化を示した．DQN などの深層強化学習もしくは今後の新しい機械学習との比較も今後の課題である．

6. まとめ

6.1. 成果

スケジューリングが抱える課題の１つである総当たりの自動化，効率化には強化学習が有効であることが分かった．

また，ブラックボックス・肥大化の従来計算アルゴリズムに対し，環境座標とローカル座標を組合せるモデル化が有効であることが分かった．

結果，強化学習によりスケジューリングの解法アルゴリズムがオープンになり，工場の生産計画領域に対し，コストパフォーマンスを高めることができる．

6.2. 将来の展望

最適化問題のうち，今回はスケジューリングについての解法がパターン化できたが，他のケースもパターン化できる可能性は高い．今後，SE がお客様へ AI を提案する機会に，積極的にこの解法を活かし，発展させていきたい．

7. 謝辞

最後に，本研究会の指導員であられた，石川冬樹主査，栗田太郎副主査，徳本晋副主査には，多方面にわたり御指導いただけた．研究の難しさ，達成感など今後の業務に活かせる考え方や自信を得ることができ，皆様に厚く御礼申し上げたい．

8. 参考文献

[1] 秋庭雅夫/佐久間章行 経営工学シリーズ 生産管理 Mar. 1987
[2] Google OR Tools https://developers.google.com/optimization/
[3] Wikipedia : Job-shop scheduling https://en.wikipedia.org/wiki/Job-shop_scheduling
[4] 柳在圭,鳩野逸生,富山伸司,田村坦之 J-Stage システム制御情報学会論文誌 vol. 12 No. 9 JIT 生産におけるロット工 程のスケジューリング手法 1999
[5] 電気学会ＧＡ等組合せ最適化手法応用調査専門委員会／編 遺伝アルゴリズムとニューラルネット　スケジューリングと組合せ最適化 1998
[6] NEC 社「「IFS Cloud」」 https://jpn.nec.com/ifs/products/ifs-guide.html
[7] FLEXSCHE 製品のご紹介 https://www.flexsche.com/product/
[8] 清藤駿成/宗形聡 日立ソリューションズ東日本技報 第 25 号 深層強化学習を用いた生産計画技術の研究
[9] 荒井 誠/宮澤 武/関谷 昌平 釧路工業高等専門学校紀要 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
[10] 遺伝的アルゴリズムによるスケジューリング問題の解法 https://qiita.com/NoriakiOshita/items/188f3b7c6ea5bd5c7e1a,2019
[11] 伊藤多一,今津義充,須藤広大,仁ノ平将人,川﨑悠介,酒井裕企,魏崇哲 現場で使える！Python 深層強化学習入門 強化学習と深層学習による探索と制御 Aug. 2019
[12]【入門】Q 学習の解説と python での実装 ～シンプルな迷路問題を例に～ https://www.tcom242242.net/entry/ai-2/
[13] PROSCHEDES https://pr.fujitsu.com/jp/news/2000/03/22.html
[14] 鈴木努 ネットワーク分析（R で学ぶデータサイエンス） May. 2017