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-

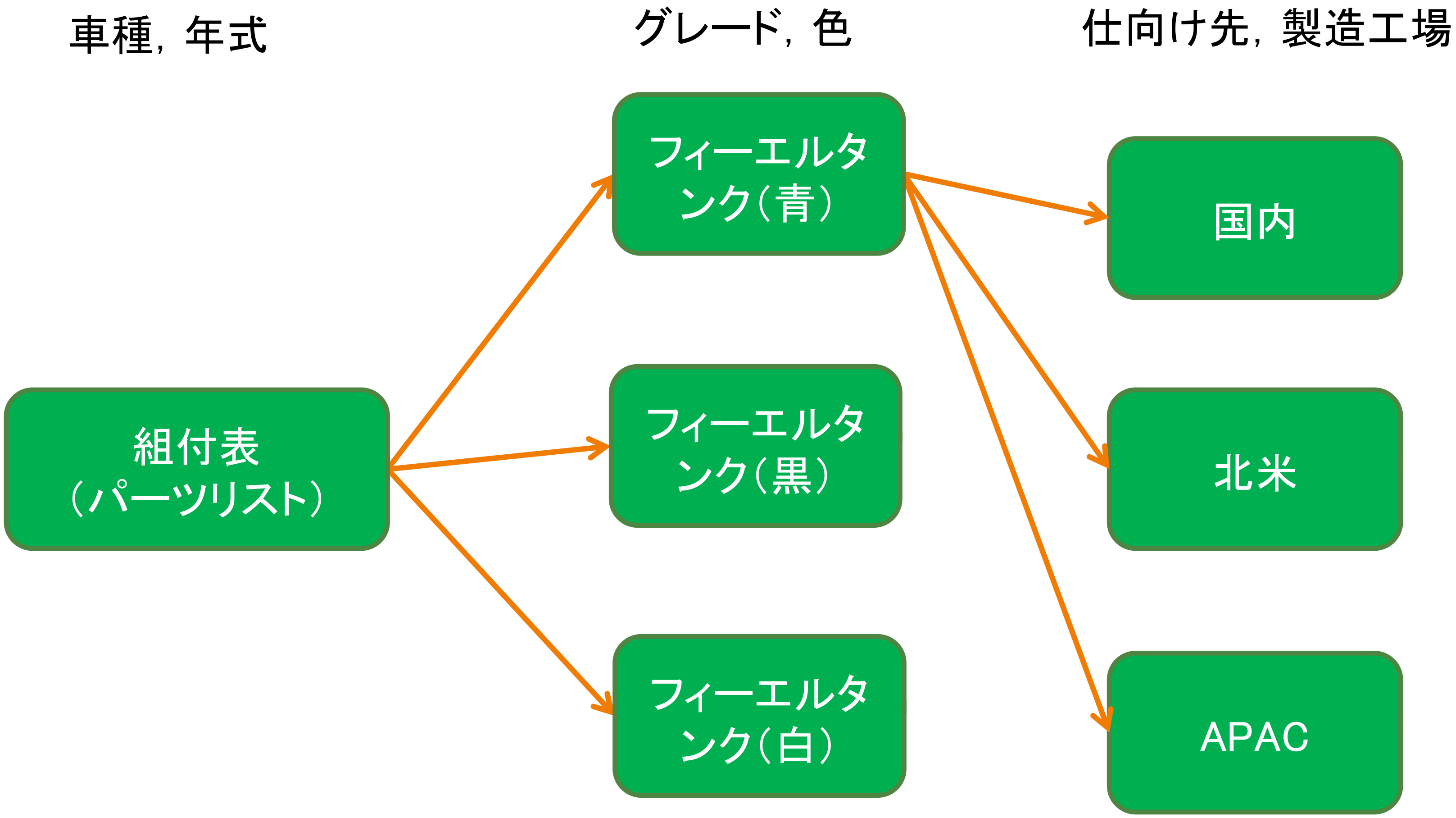
研究コース5 AI Scheduler チーム
研究員：平井 宣 株式会社ＩＨＩエスキューブ

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

2023/2/24

オーダとは

オーダで，製品，製造工場，販売先，数量が決まる．



Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

タスクとは

タスクで作業時間が定まる.

板金，溶接
8.0 h

段取り
1.0 h

研磨，下塗り
2.0 h

搬送
2.0 h

塗装・焼入
4.0 h

検査，組立
2.0 h

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

生産計画とは

２つの生産設備M1、M2 が直列に連結されたフローショップ工程で、
５つのジョブの総処理時間を最小にする生産スケジュールについて考える。
すなわち、各ジョブは、まず、生産設備M1 で処理され、次にM2 で処理される。
ただし、各生産設備は、1度に1つのジョブしか処理できないものとする。
各ジョブの各生産設備における処理時間が下表に示されるとき、
最小の総処理時間(すべてのジョブの処理を完了するまでの時間)を下記の解答群から選べ。

表　処理時間データ

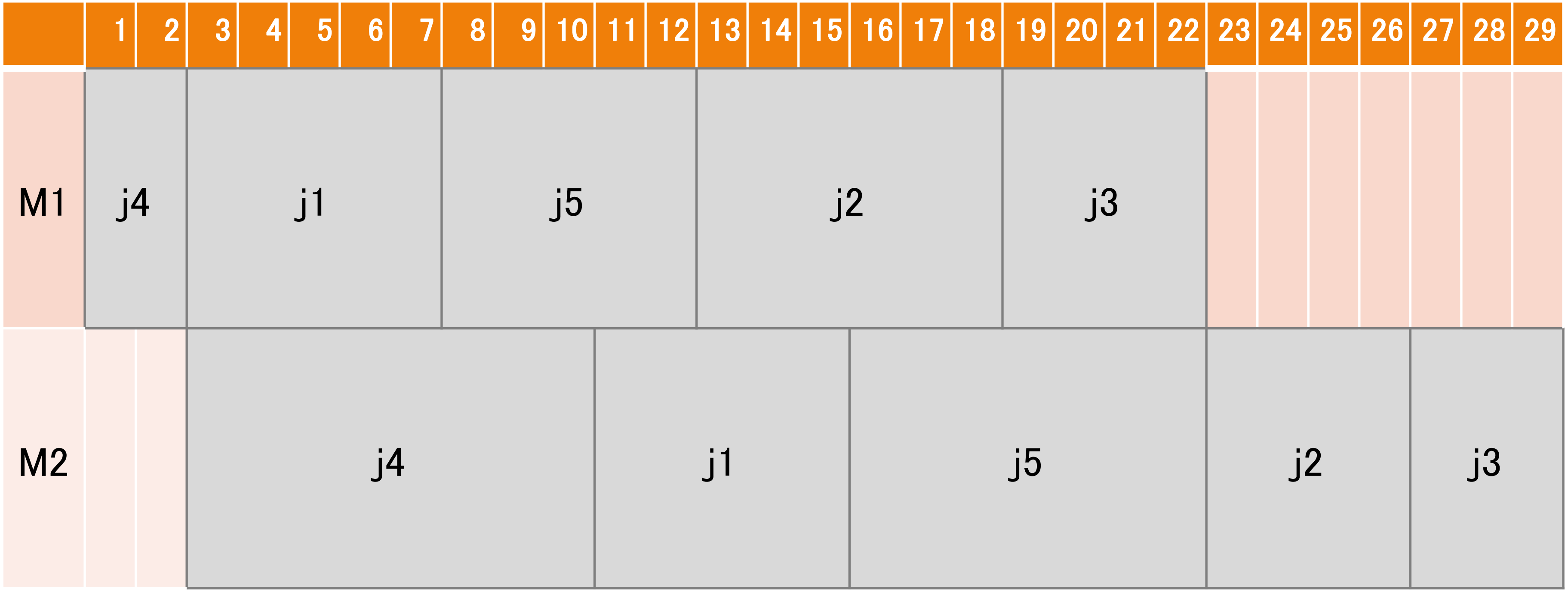
	M1	M2
ジョブ1	5	5
ジョブ2	6	4
ジョブ3	4	3
ジョブ4	2	8
ジョブ5	5	7
合計	22	27

〔解答群〕
ア　27
イ　29　（正解）
ウ　31
エ　33

引用：平成20年 中小企業診断士試験の問題より引用

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

ジョンソン法による解法



諸々の制約条件を考慮しながら,
生産設備の稼働効率を最大にしうる
総工期の計画値を定める

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

課題と提案

	M1	M2
ジョブ1	5	5
ジョブ2	6	4
ジョブ3	4	3
ジョブ4	2	8
ジョブ5	5	7
合計	22	27

n個をn個取って並べた総数

$$_{n}P_{n}$$

n:ジョブ数×タスク数

3,628,800通り

・実世界とAIの**モデル分離**によるソフトウェアミニマム

・**強化学習**で総当りを効率化

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

モデルの分離：スケジューリングモデル

A ⊗ E　⊗ 行列演算（AとEの合成）

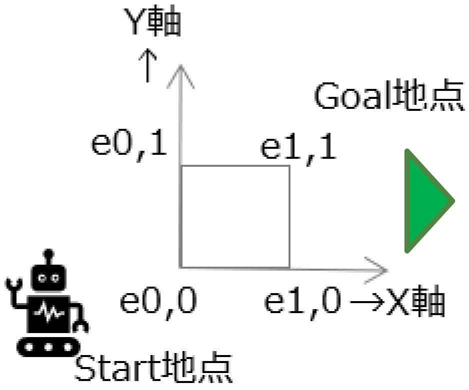
A：実世界のモデル（ローカル行列）

オーダ	タスク	要素	
		作業時間	タスク順序
#o0	#t0	1.0	null
#o0	#t1	0.5	null
#o1	#t0	0.5	null
#o1	#t1	0.1	null

$$\begin{bmatrix} a0,0 & a0,1 \\ a1,0 & a1,1 \end{bmatrix} = ai,j:[\text{作業時間},\text{タスク順序}]$$

$$= \begin{bmatrix} [1.0,null] & [0.5,null] \\ [0.5,null] & [0.1,null] \end{bmatrix}$$

B：AIのモデル（環境行列）



探索ルート

・e0,0→e0,1→e1,0→e1,1
・e0,0→e1,0→e0,1→e1,1

$$\left[\begin{bmatrix} e0,0,0 & e0,0,1 \\ e0,1,0 & e0,1,1 \end{bmatrix}, \begin{bmatrix} e1,0,0 & e1,0,1 \\ e1,1,0 & e1,1,1 \end{bmatrix} \right]$$

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

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

モデルの分離：スケジューリングモデル

A ⊗ E　　⊗ 行列演算（AとEの合成）

A:ローカル行列

$$\begin{bmatrix} [1.0,null] & [0.5,null] \\ [0.5,null] & [0.1,null] \end{bmatrix}$$

⊗

B:環境行列

$$\begin{bmatrix} \begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \end{bmatrix}$$

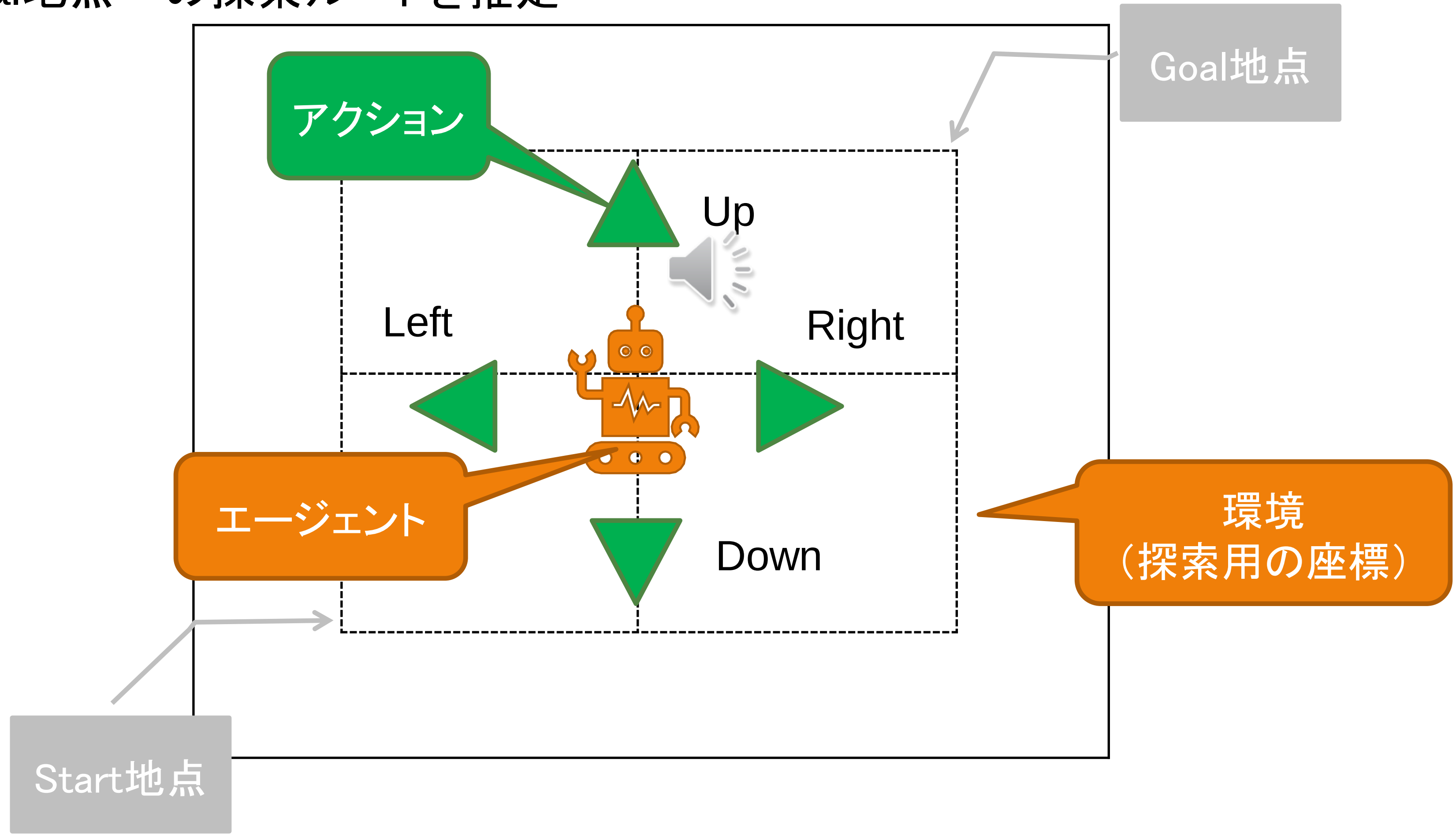
=

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

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

エージェントのアクション

ローカル行列(この場合はオーダ×タスク＝3×3)から
探索用の座標を決め，エージェントが座標決めのアクションを繰り返し，
Goal地点への探索ルートを推定

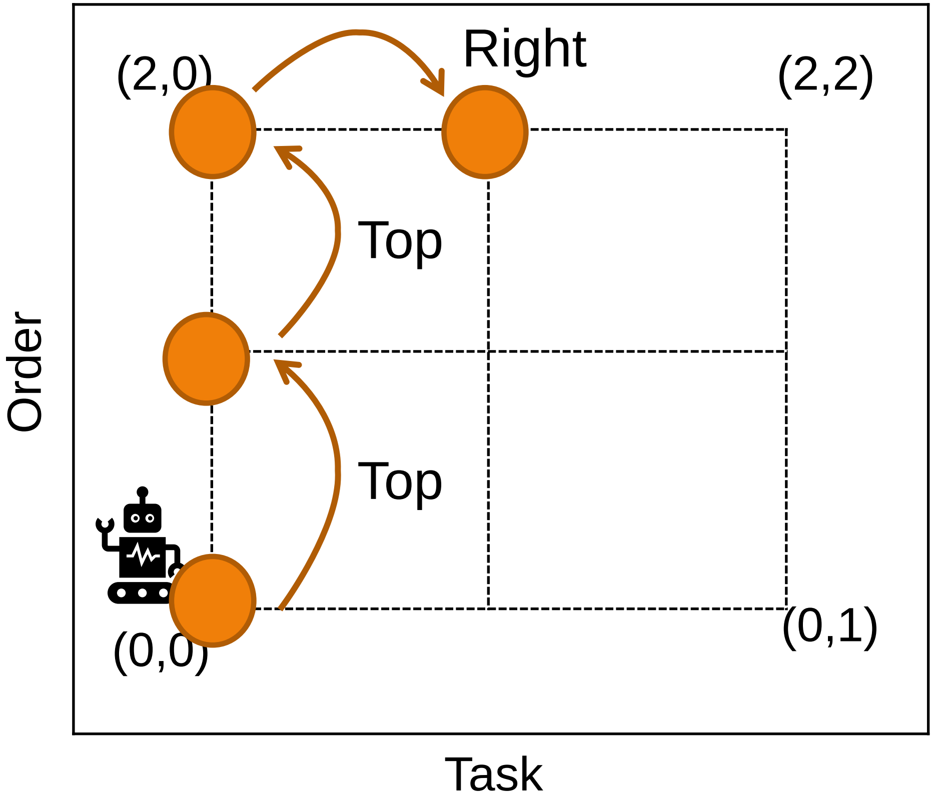


Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

報酬によるルート探索

報酬が高いルートを探索

- 壁なら-1，未通過があっても-1
- 逆走（同じX軸でのLeft）は-1
- それ以外は（作業時間÷総時間）×10（　　）



←左図における環境行列

実行順序	行	列
0(Start地点)	0	0
1	1	0
2	2	0
3	2	1

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

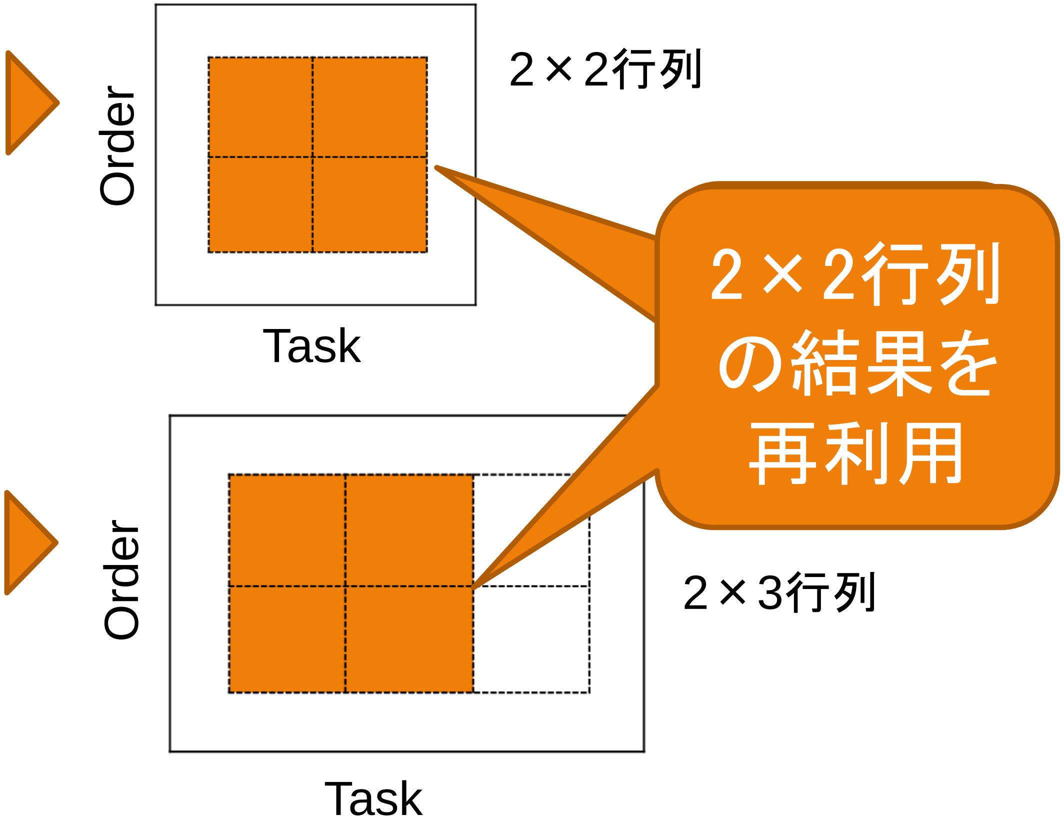
ルート探索の再利用

異なる生産計画（ローカル行列）

	タスクx	タスクy
オーダa	1	2
オーダb	6	5

	タスク1	タスク2
オーダ1	5	5
オーダ2	6	4
オーダ3	4	3

探索用の座標（環境行列）



Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

実験と考察

目的	評価観点	論証方針	実験法	結果
モデル分離の実用性	アルゴリズム	ベンチマーク（ホワイトボックク）	手計算	○
	データバリエーション	ベンチマーク（ブラックボックス）	サンプルプログラム出力	○
	ローコード	ソースプログラムの行数(LOC)比較	手計算	△
強化学習の効果	計算再利用	学習の収束性	サンプルプログラム出力	○
	正確さ	ログチェック	ログの目視チェック	○

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

モデル分離に対するベンチマーク

ジョンソン法との比較

Job	task_0	task_1	task_2
job_0	3	2	2
job_1	2	4	1
job_2	0	4	3

ジョンソン法の結果

	1	2	3	4	5	6	7	8	9	10	11	12	稼働率(%)
t_0			j_1	j_1		j_0	j_0	j_0					41.7%
t_1	j_2	j_2	j_2	j_2	j_1	j_1	j_1	j_1	j_0	j_0			83.3%
t_2					j_2	j_2	j_2		j_1		j_0	j_0	50.0%

	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

JIT計画法との比較

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

		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

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

ローコードに対するLOC比較

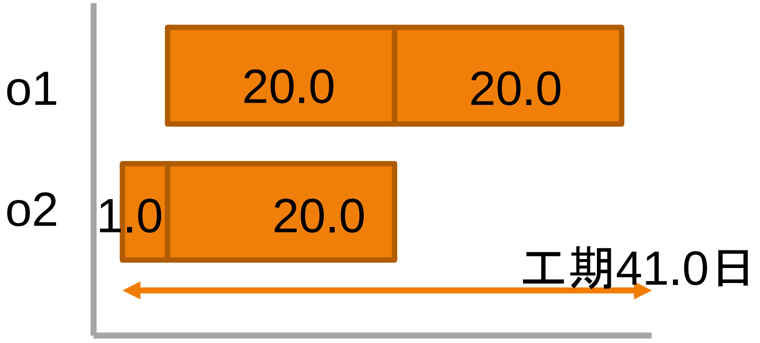
規模大

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**

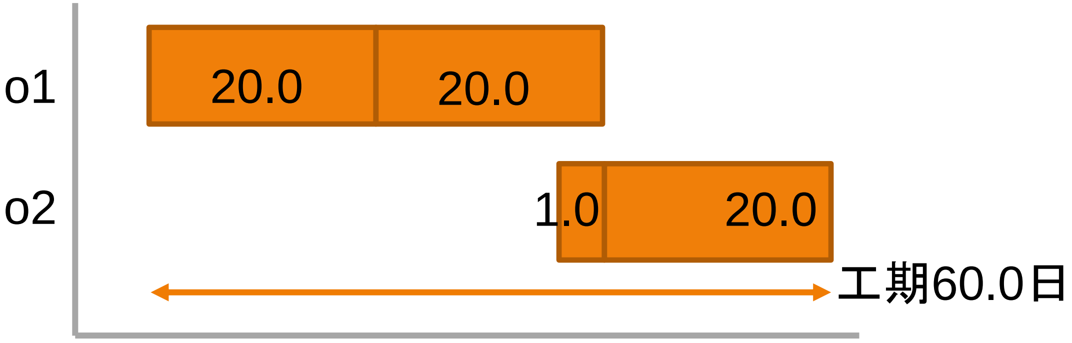
オーダ	タスク	作業時間
o1	t0	20.0
o1	t1	20.0
o2	t0	1.0
o2	t1	20.0

		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

AIスケジューラ



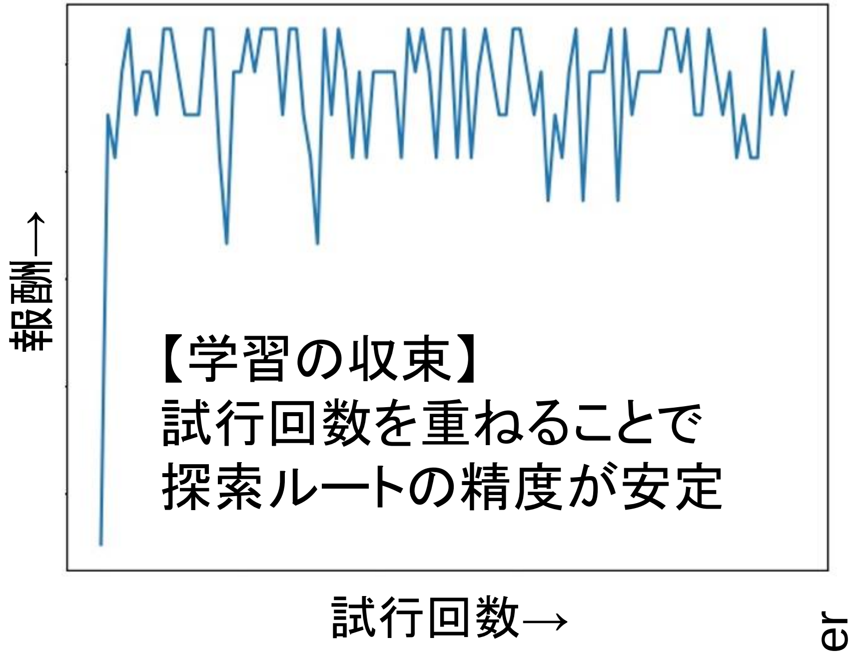
JIT計画法



Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

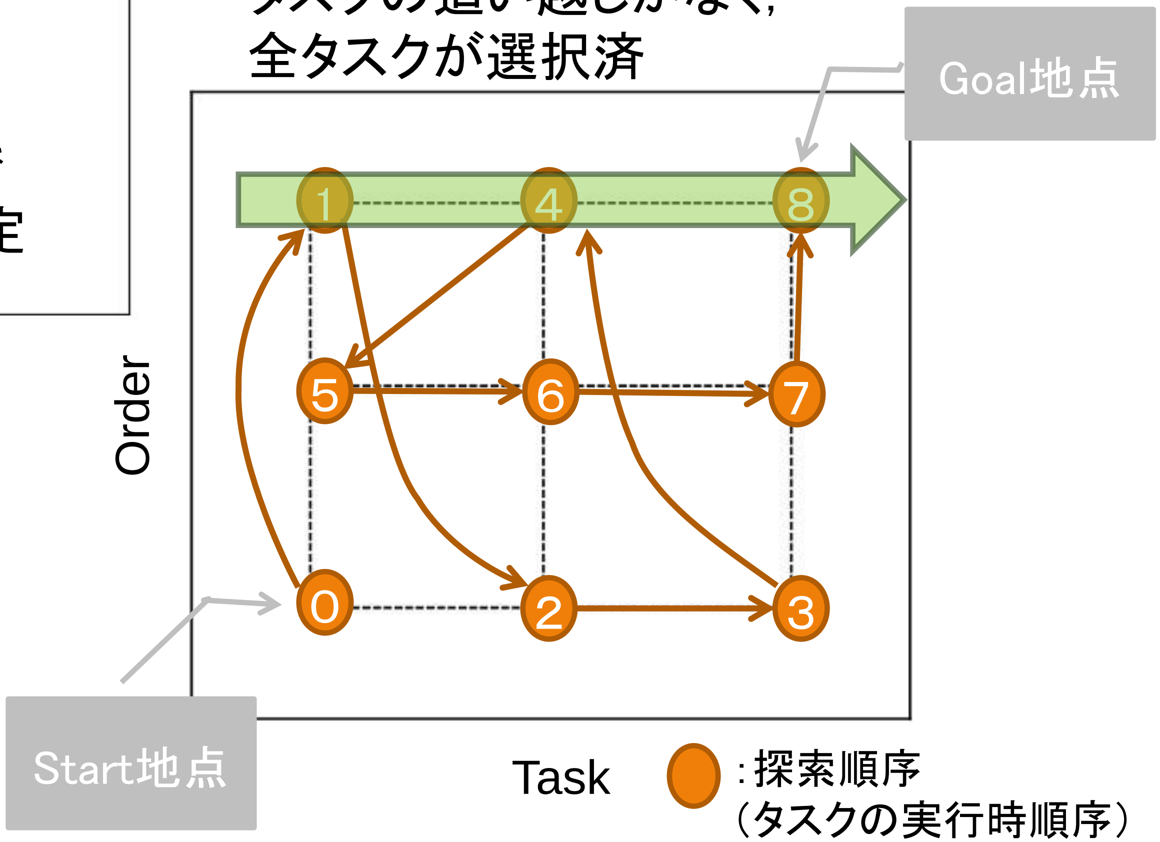
強化学習の効果について

計算の再利用



正確さ

【ログチェック】
タスクの追い越しがなく,
全タスクが選択済



Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

成果と展望

成果

スケジューリングが抱える総当たり問題は，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・肥大化の傾向にある支援ツールに対し，行列と行列演算を活用するモデルの分離が，アルゴリズムを単純にし，ローコードの可能性に一定の道筋をつけることができた．
今回の研究成果は，長期的には，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に貢献できる．

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標準化できたが，他の問題も標準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．

【謝辞】
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ました．ありがとうございます．

Copyright (C) 2022 ソフトウェア品質管理研究会 All Rights Reserved

ご清聴ありがとうございました

SQiP

Software Quality Profession