



# 生産計画・オペレーション最適化シリーズ 製造現場の オペレーション最適化セミナー

2025年11月21日  
株式会社NTTデータ数理システム

# 「生産計画・オペレーション最適化」シリーズの内容

「生産における計画・最適化」にあらわれる2つの課題を扱う



## 中長期計画の立案

- 需要や納期を満たすスケジュールを策定
- 需要量や製造コストは固定値とみなす
- 数理最適化が適している。



時間スケール: 長  
解像度: 粗

## 現場オペレーションの実行

- 時々刻々の現場オペレーションを決定
- リアルタイムに発生する事象を考慮(需要の変動, 不良品, 再配達…)
- シミュレーションが適している。



時間スケール: 短  
解像度: 細

# 本セミナーの内容

1. シミュレーションとは？
2. 「オペレーション最適化」の適用対象と事例
3. 「オペレーション最適化」の解法
4. プロジェクトの流れ

ご自身の業務から課題を抽出できるようになる

「数理最適化」「シミュレーション」「強化学習」の位置づけを理解する

必要なデータや開発工程のイメージを得る

# 自己紹介

豊岡祥 株式会社NTTデータ数理システム  
シミュレーション＆マイニング部 グループリーダー



## 「シミュレーション・数理最適化・機械学習」を活用した受託開発

広告配信の効果予測と最適化、配送経路最適化(ラストワンマイル)

ECサイトのレコメンデーションモデル構築、法人営業データ分析

製造ラインシミュレーション、交通シミュレーション、感染症シミュレーション

## S<sup>4</sup> Simulation System の開発

2D人流シミュレーションに適した移動経路計算アルゴリズム

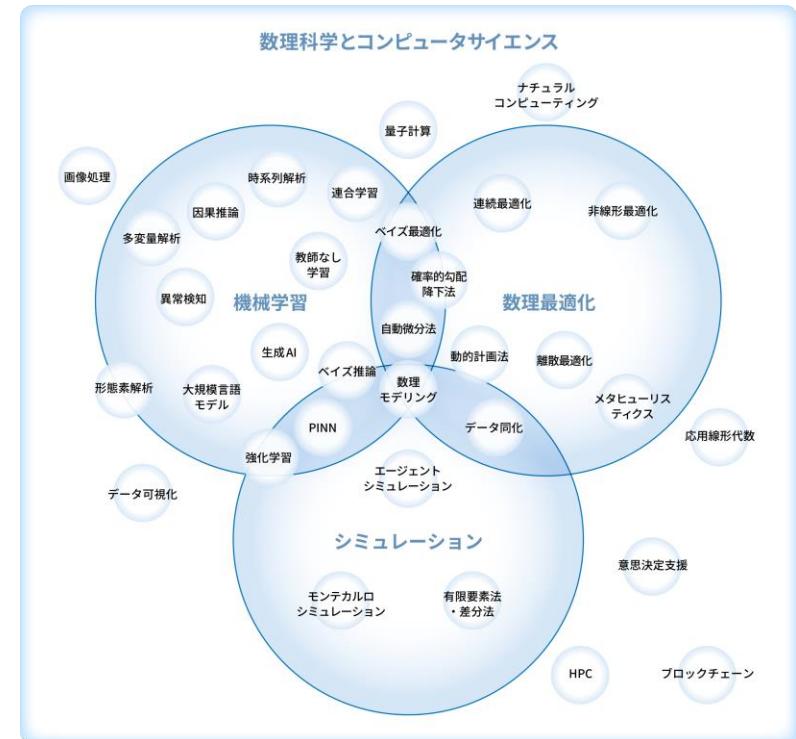
可視化ダッシュボード、シミュレータと強化学習との接合

## 執筆

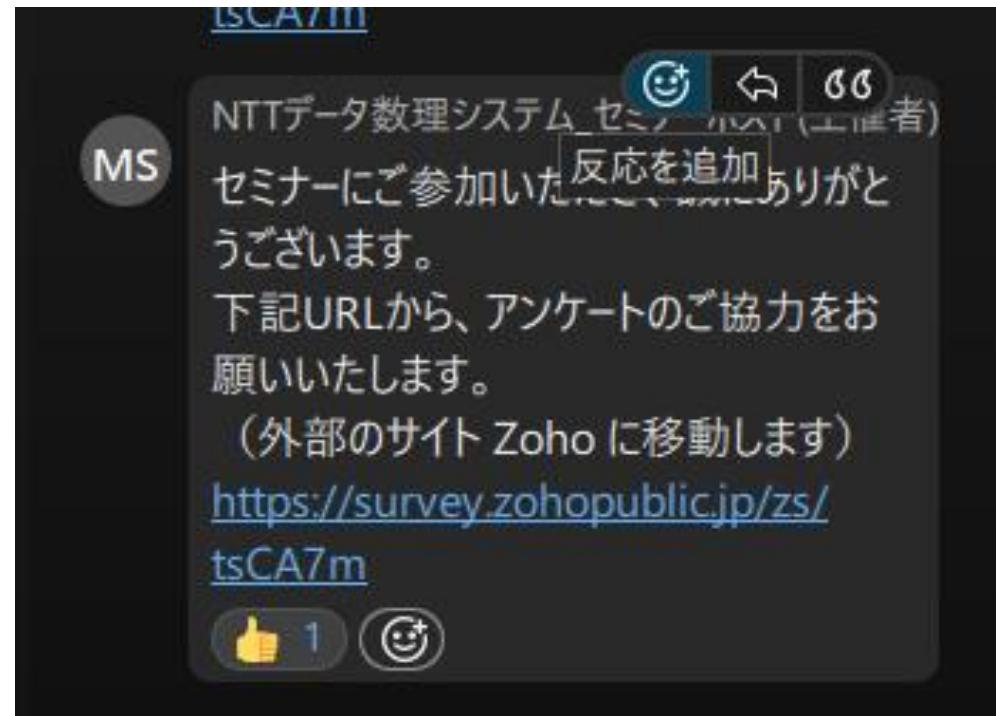
Python 講座 <https://www.msiism.jp/column/python-code-reading/>

XAI(説明可能なAI)——そのとき人工知能はどう考えたのか？

Python入門 AtCoder Programming Guide for beginners



# 質問・リアクションお待ちしております！





# 1

## シミュレーションとは？

# シミュレーションとは？

シミュレーション = 計算機上で現実を再現する技術

数理システムは色々な対象を表現可能な汎用シミュレーション技術を有している



待ち行列, 製造ライン, 物流,

離散イベントシミュレーション



人流



自動車交通

エージェントシミュレーション



S<sup>4</sup> Simulation System



社会（市場, 組織, 情報伝搬）…



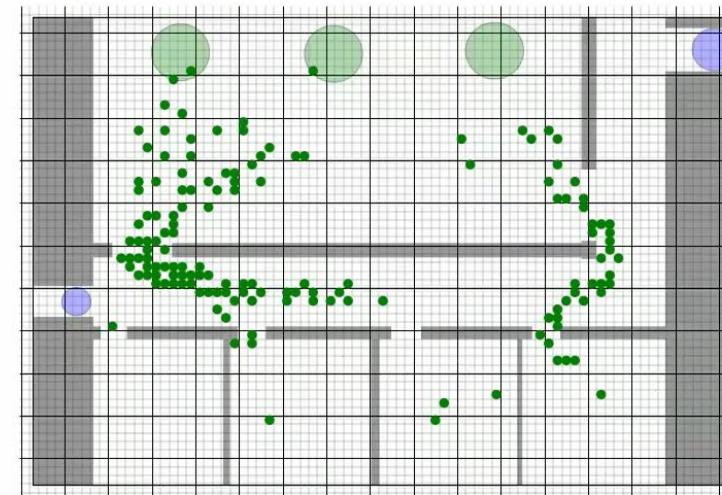
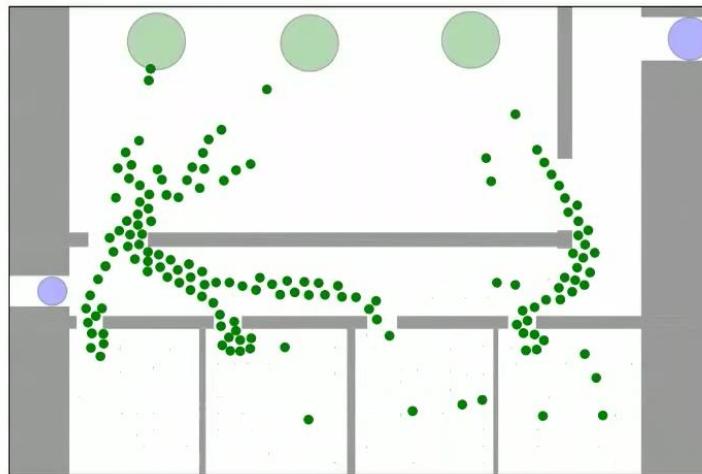
流体, 送電, 気象 …

連続シミュレーション

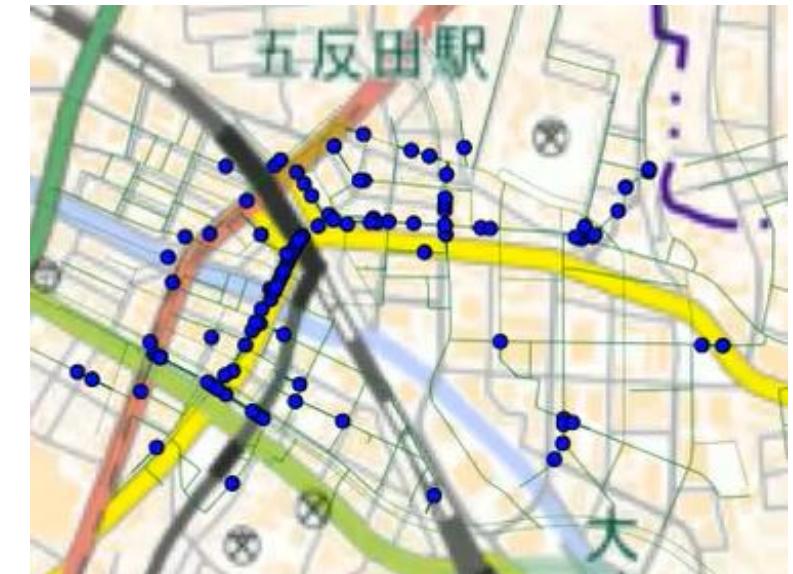
# 人流・交通シミュレーション

人や車の流れを再現する技術

人流シミュレーション



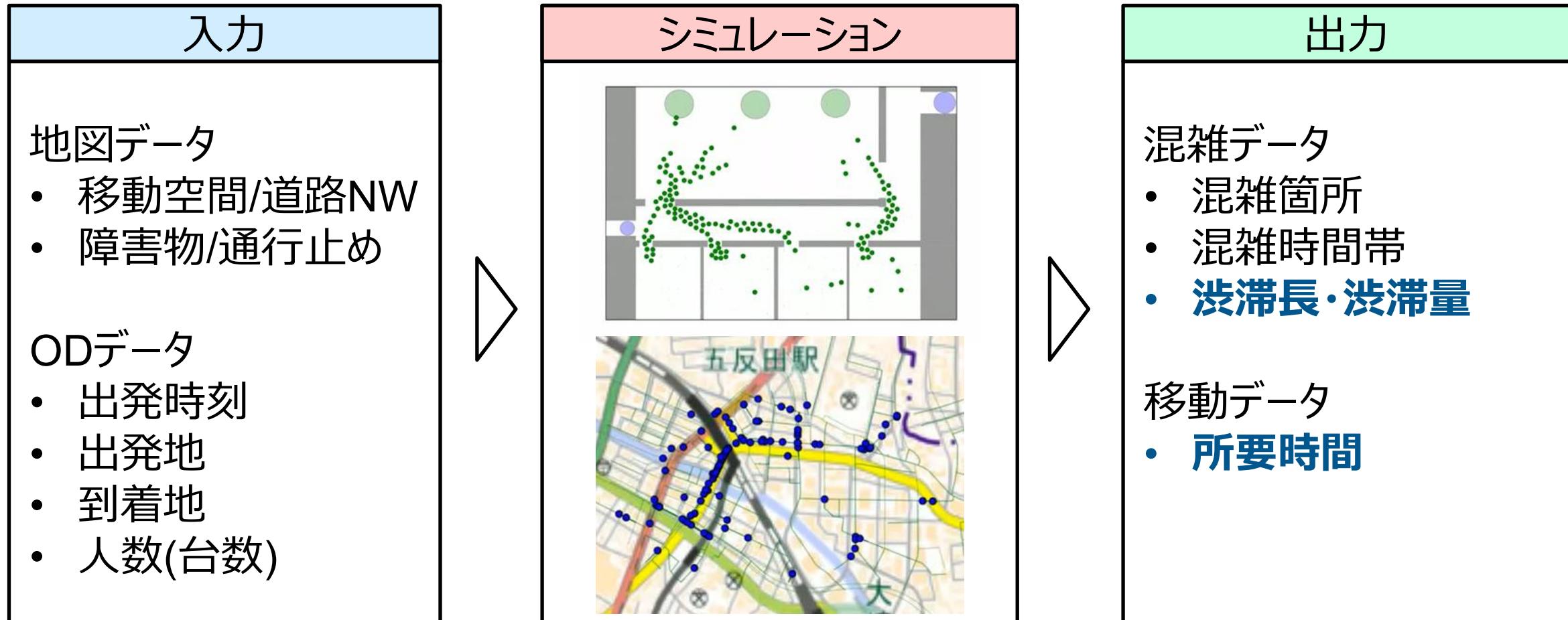
交通シミュレーション



出典：国土地理院ウェブサイト

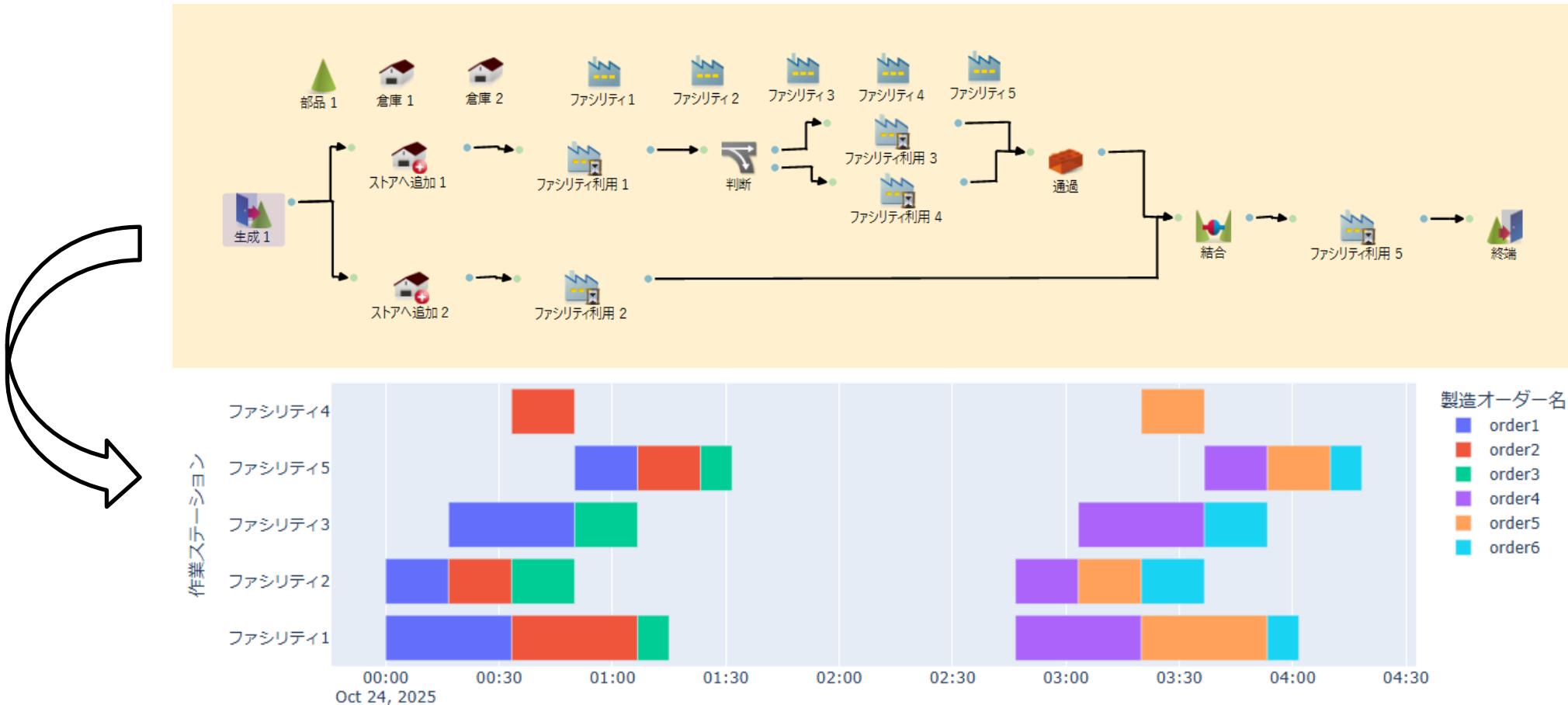
# 人流・交通シミュレーション

人や車の流れを再現する技術



## 製造ラインシミュレーション

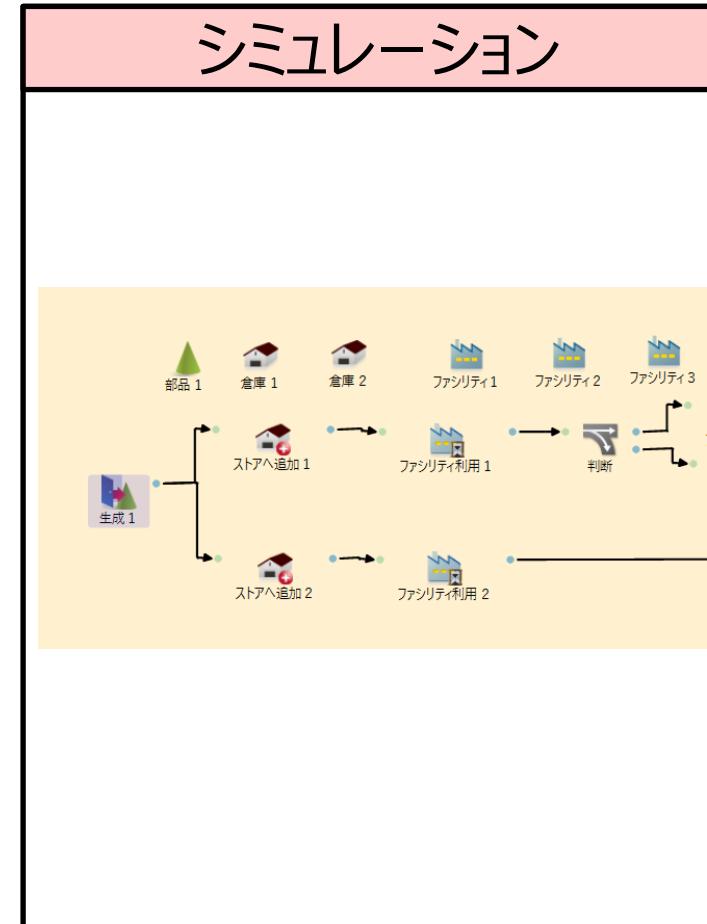
## 工場の製造ラインにおける機械の動作や部品の流れを再現する技術



# 製造ラインシミュレーション

工場の製造ラインにおける機械の動作や部品の流れを再現する技術

入力
設備データ • 作業機械 • シフト • 処理時間
生産計画 • 投入順序 • 分岐・合流条件 ...



出力
製造結果 • 生産効率 • ガントチャート
分析 • 機械の稼働率 • 滞留数・待ち時間 • ボトルネック可視化

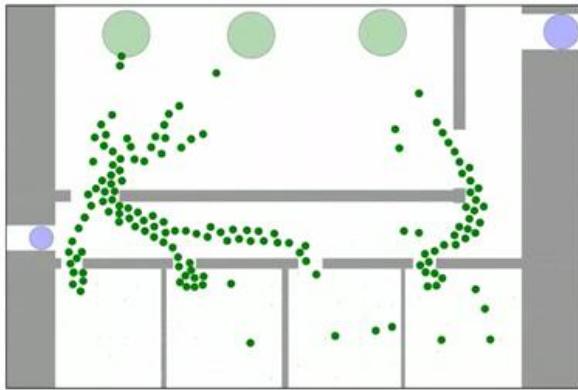


# 2

## 「オペレーション最適化」の 適用対象と事例

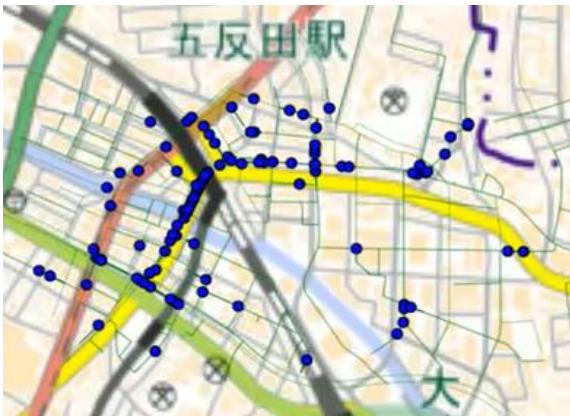
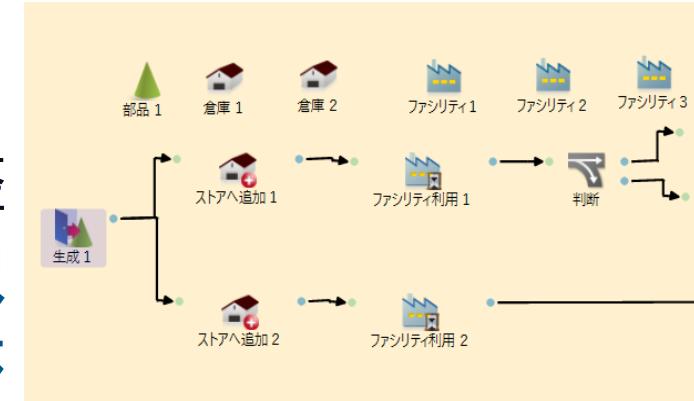
# 「オペレーション最適化」とは？

リアルタイムなオペレーションを最適化する問題

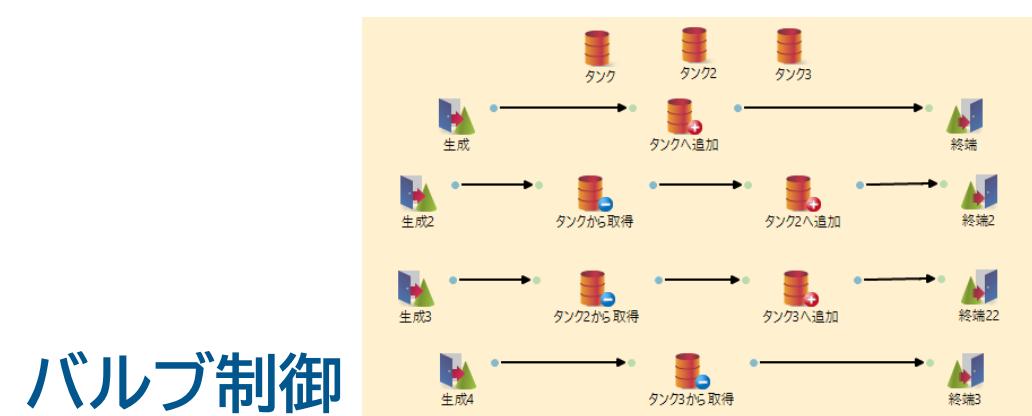


サイネージによる人流誘導

製造順序調整  
AGVの経路選択  
倉庫ピッキング順序



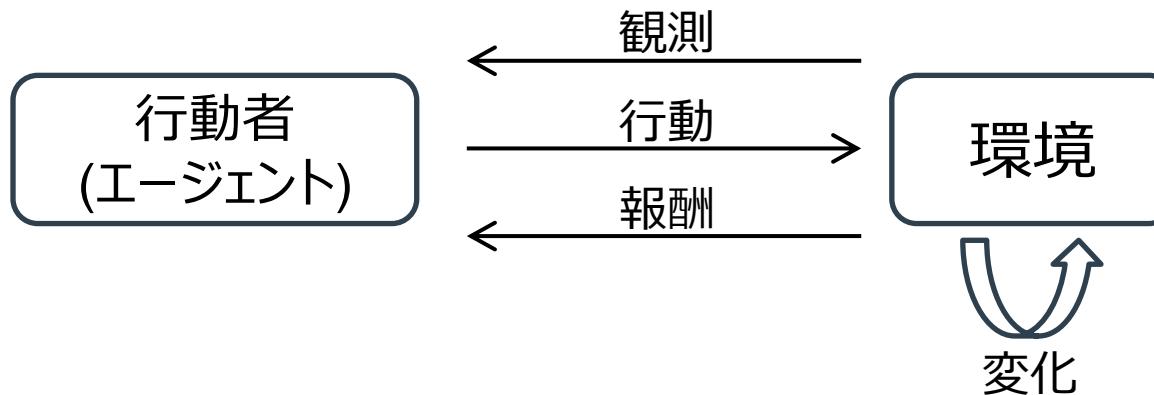
信号制御  
情報発信による経路誘導



# 「オペレーションの最適化」の対象

行動者(エージェント)と環境の相互作用で表現できるモデルを対象とする:

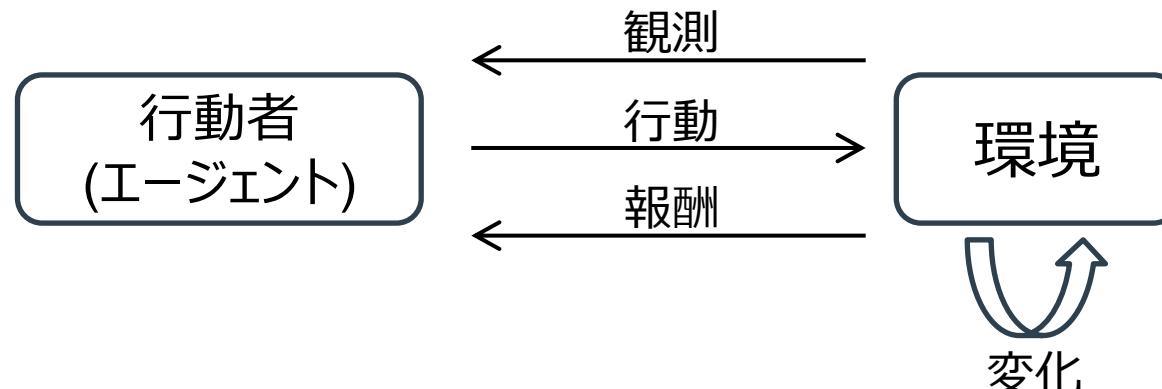
- ・ エージェントは環境を観測し、行動を選んで実行
- ・ 環境が変化し、報酬を返す
- ・ 長期的に報酬を最大化することが目標



# 「オペレーションの最適化問題」の対象

色々なシーンを扱うことができる

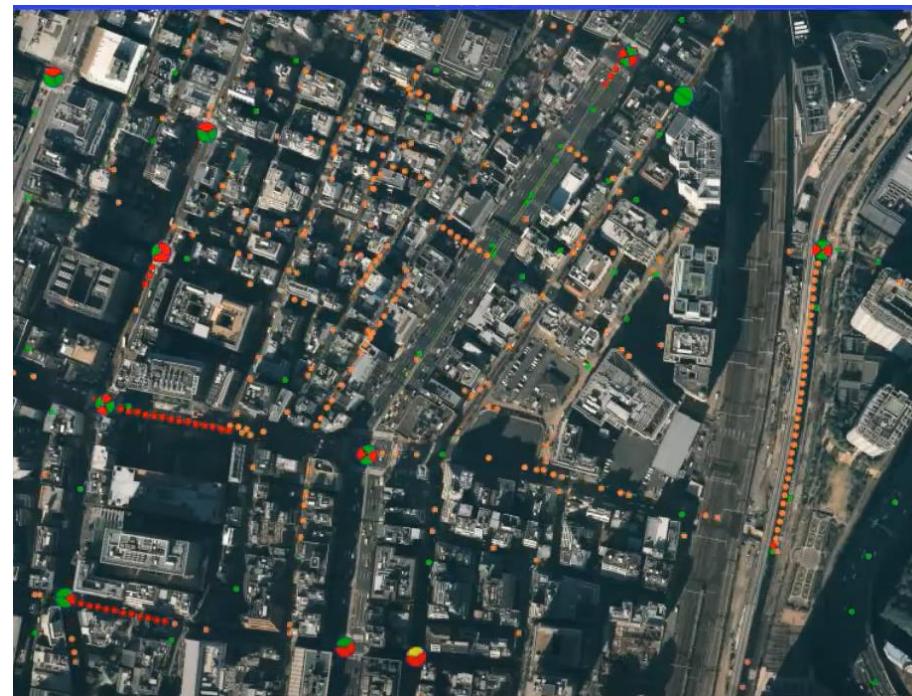
シーン	状態	行動	報酬
人流制御	各エリアの人口密度	サイネージの内容変更	人の移動時間
信号制御	各リンクの車両台数	信号の切り替え	車両待ち時間
製造ライン	各バッファ待ちロット数 機器稼働状況	次着手ロット選択 AGV 経路選択	工場全体のスループット
化学プラント	各タンクの空き容量 リンクの流量	バルブの開閉	生産量 安全性



# 過去事例：交通シミュレーション→信号制御効率化

## 背景：

- ・ 海外の大都市において渋滞が常態化している
- ・ 現在起きている渋滞だけでなく、GPSデータと連携し今後生じるであろう渋滞も考慮したリアルタイムな交通信号による制御を行いたい



# 過去事例：交通シミュレーション→信号制御効率化

## 取り組み：

1. 市街地の交通シミュレーションを構築
2. 複数の**信号制御方式の比較**

強化学習を活用する  
既存研究も多い

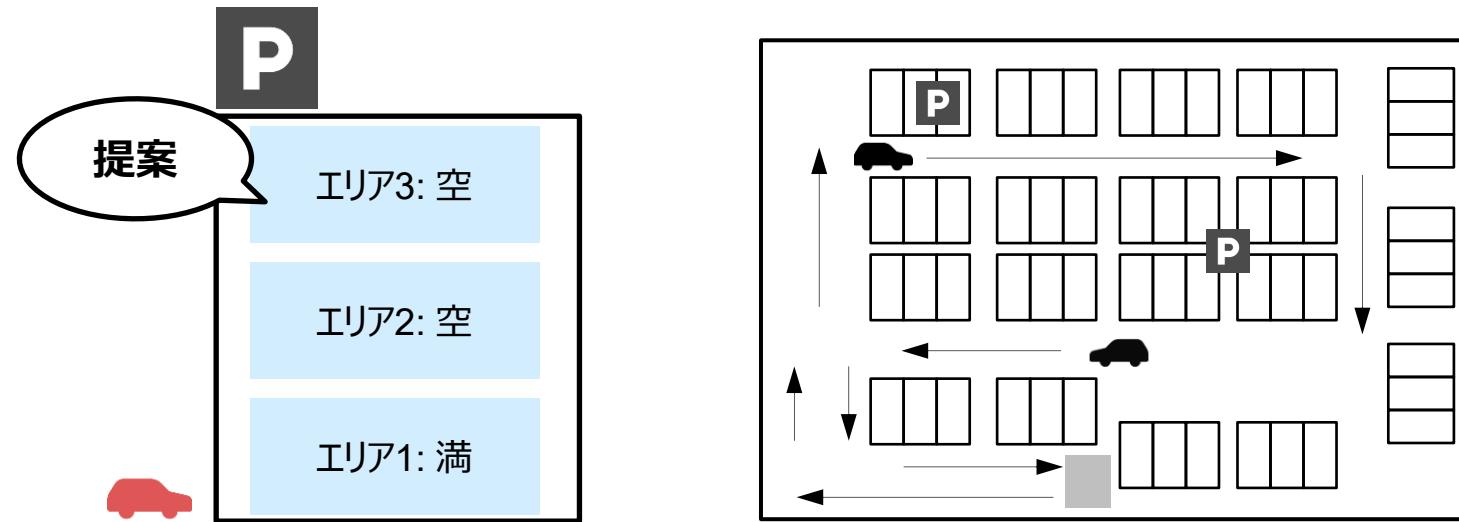
- ・ 時間式：あらかじめ決められた周期
- ・ 動的制御：車両が多くなりそうな方向を予測し、その方向の青時間を長くする



# 想定事例：駐車場における誘導

## 背景：

- ・ 大規模な駐車場で空きを見つけるための移動時間・待ち時間が課題となっている
  - ・ スマートな誘導を行いたい
    - ・ 空きを見つけやすいエリアを提案
    - ・ 現在は空きだが、近い将来満になりそうなエリアは避ける
- 「車両に対して行先を提案する」オペレーションを最適化したい

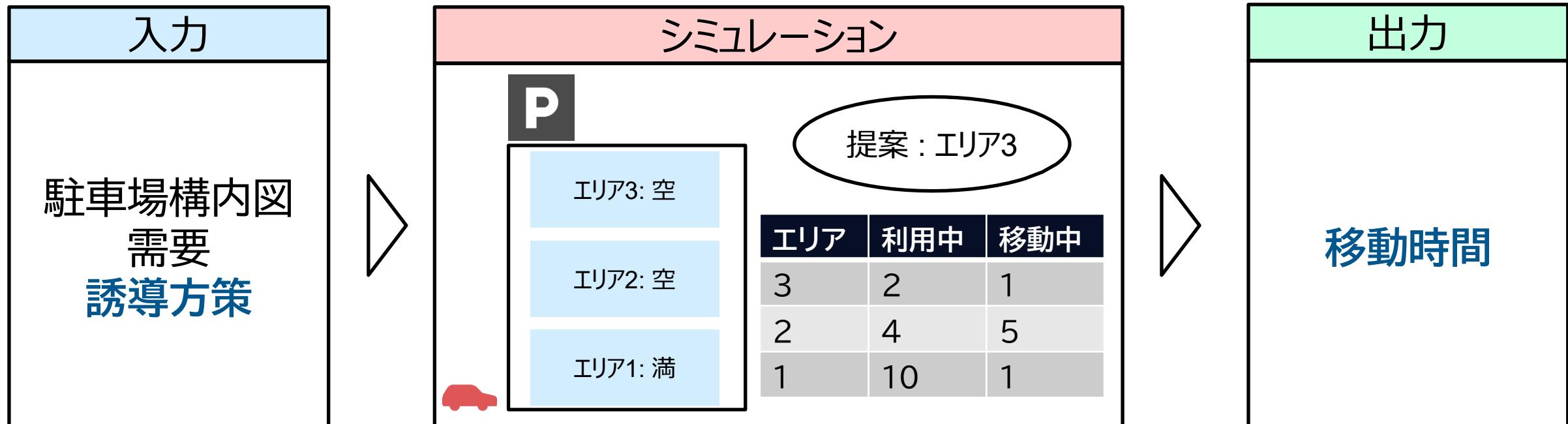


# 想定事例：駐車場における誘導

解決方法：シミュレーション + 強化学習

1. 駐車行動を再現するシミュレータを構築
2. 強化学習による誘導モデル構築

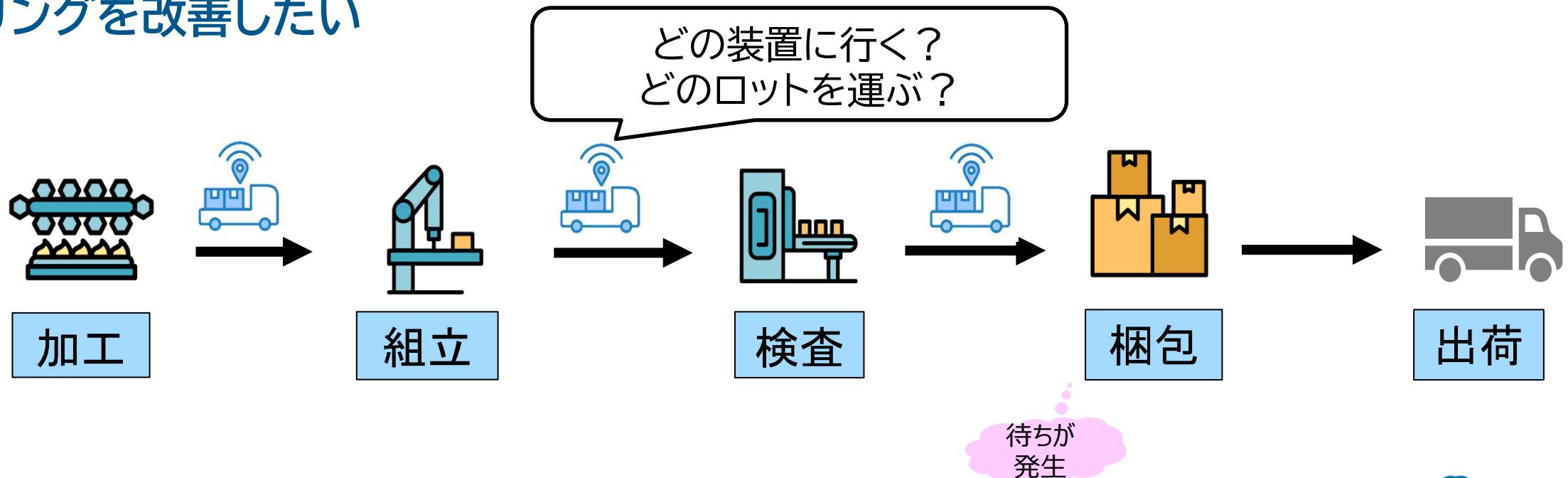
- ・構内の状態から提案内容を決める**方策**をランダムに初期化
- ・シミュレータ上で色々な行動を試し、利用者の**移動時間**が最小化するように学習



# 事例: 構内物流シミュレーション→ AGVの運用最適化

## 背景:

- 生産ラインにおいてAGV(無人搬送車)が工程間のロット輸送を行う
  - 固定巡回方式: 決まった経路を巡回する
  - 動的スケジューリング方式: 運ぶロットをリアルタイムに決める
- 輸送がボトルネックとなり全体のスループットが上げられないため、**動的スケジューリングを改善したい**



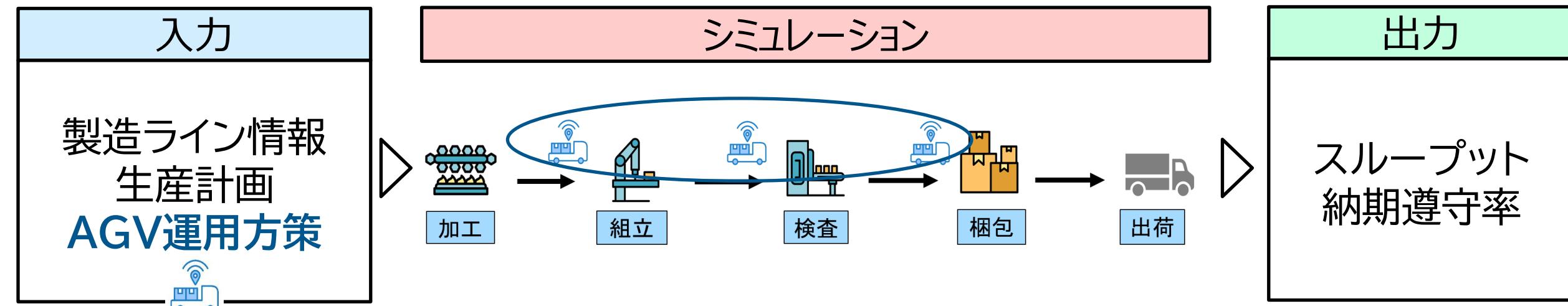
# 事例: 構内物流シミュレーション→ AGVの運用最適化

## 取り組み:

1. AGV を含む製造プロセスを再現するシミュレータを構築
2. シミュレータ上で AGV を動かす賢い方策を強化学習により構築

納期の近いロットから運ぶ  
空いている装置に運ぶ  
...

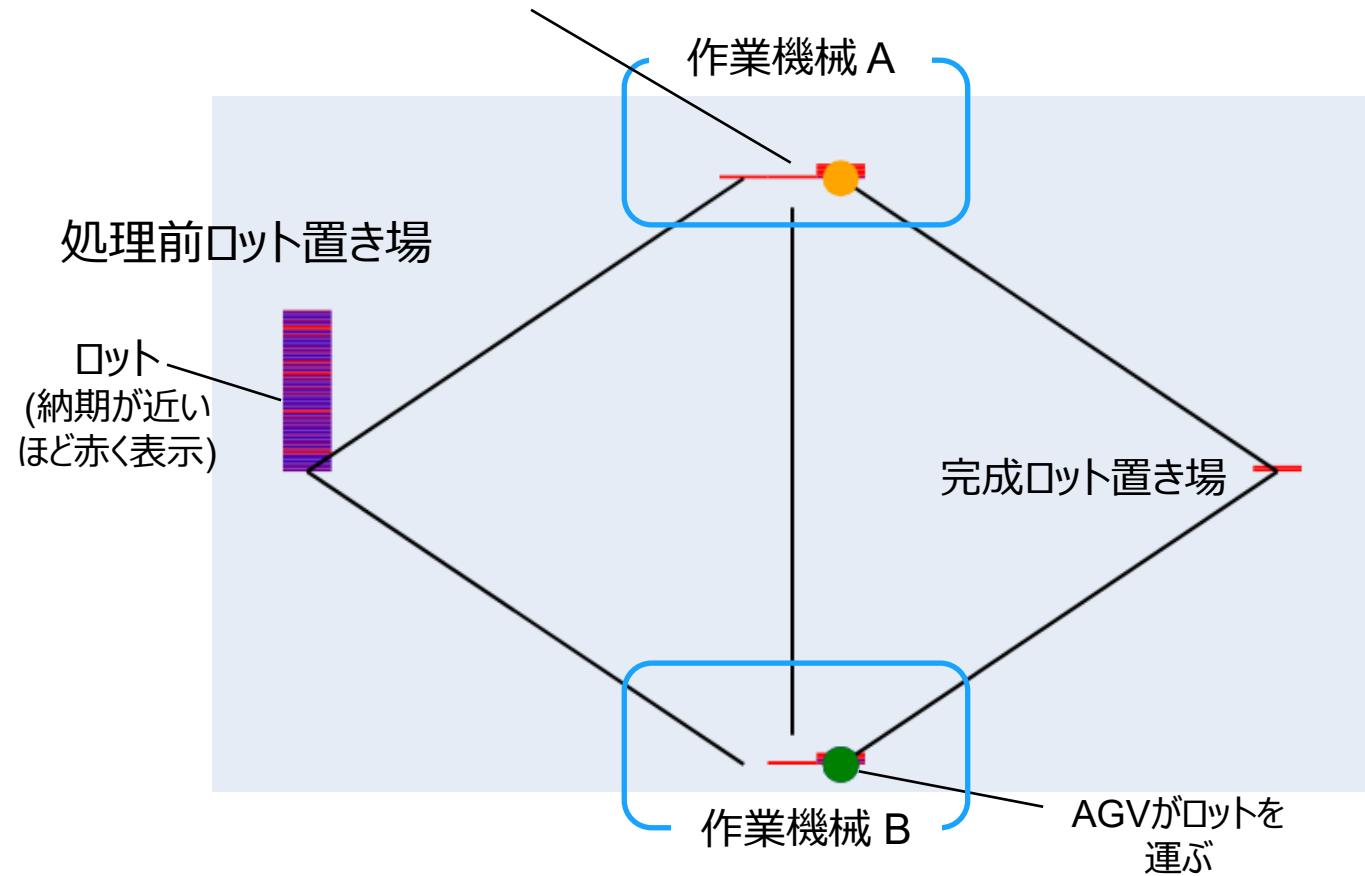
シミュレーションで試行錯誤しながら  
うまくいく方策を自動で学習させる方法



# 単純化した問題設定

- AGV (●○) でロット (—) を搬送
- 作業機械は来た順にロットを処理
- 納期に間に合うように完成させる
- AGVの搬送ロット選択を最適化

ロット (処理待ち / 処理中 / 処理済み)

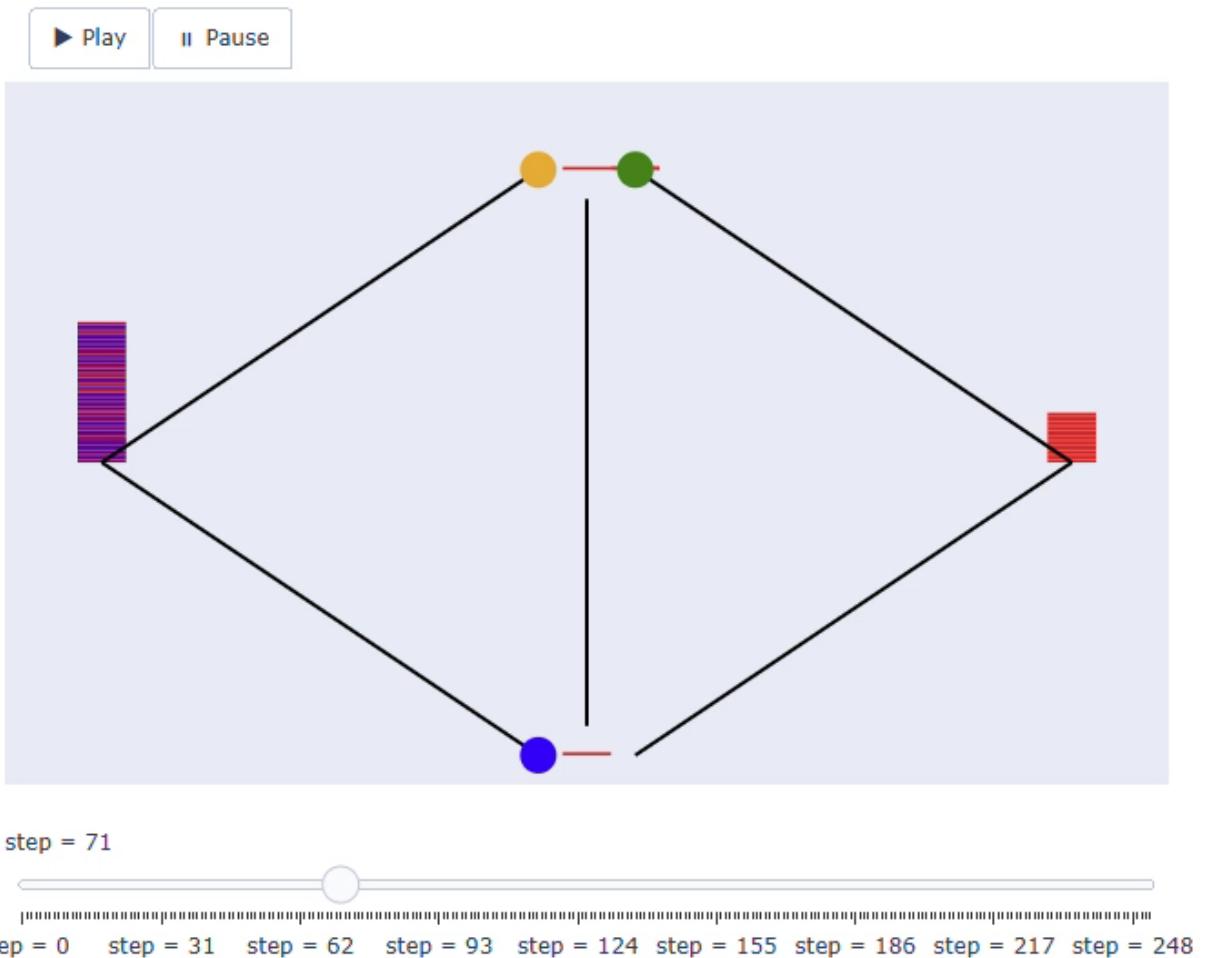


# 動作イメージ

何が「良い方策」かは一概には言えない

- ・ 納期が近いものの優先？
- ・ 残りの工程数が多いロットを優先？
- ・ 処理機械の稼働率を上げるには？
- ・ 複数のAGVで手分けして運ぶ？

→ どのように実現するか？





# 3

## 「オペレーション最適化」の解法

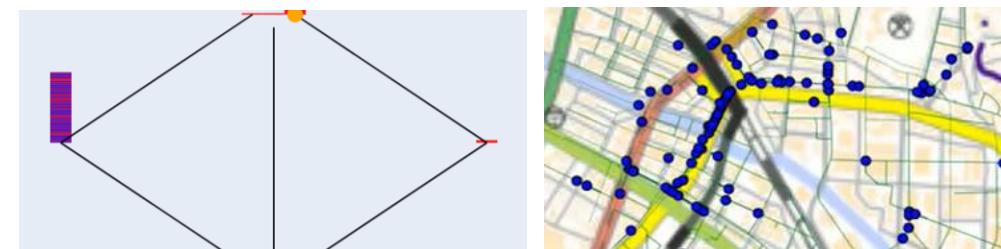
# 「オペレーションの最適化問題」の解き方

2通りのアプローチ:

## シミュレーション型アプローチ

対象を再現するシミュレータを作り、そのうえで試行錯誤して方策を求める。

ルールベース / 探索 / 強化学習などの方法がある。



## 計画型アプローチ

数式レベルに単純化した問題を解き、すべてのオペレーションを最初に決定する

混合整数計画法(ソルバ) / メタヒューリстиクス などの方法がある

AGV1	ロット1 運搬	移動	ロット5 運搬	...
AGV2	ロット2 運搬	移動	ロット3 運搬	...
AGV3	ロット4 運搬	移動	ロット6 運搬	...
機械1		ロット1 処理	ロット2 処理	...
機械2		ロット4 処理		ロット5 処理

# 計画型アプローチ

設定が単純な場合 (ex. 処理時間や移動時間が固定値)

→ 数式レベルに単純化した問題を解き、すべてのオペレーションを最初に決定する

(ジョブショップスケジューリング問題と呼ばれる: <https://www.msi.co.jp/solution/nuopt/docs/examples/html/02-25-00.html>)

## ① 最適化問題の定式化

**変数:** AGVの動き方とロット選択、機械のロット選択

**制約条件:** 機械処理時にロットが搬送済み / AGV運搬時にロットが処理済み / すべてのロットを完成させる

**目的関数(最小化):** 納期を超過するロット個数

## ③ 解から計画を復元

AGV1	ロット1 運搬	移動	ロット5 運搬	...
AGV2	ロット2 運搬	移動	ロット3 運搬	...
AGV3	ロット4 運搬	移動	ロット6 運搬	...
機械1		ロット1 処理	ロット2 処理	...
機械2		ロット4 処理		ロット5 処理



## ② 数理最適化による求解

(※) 数理計画法が適用できるかは「単純な四則演算程度で目的関数や制約条件を計算できるか」がおおまかな基準として考えられる

# 計画型アプローチで対処困難な要素

- AGV の移動時間が変動する
  - 狹い通路は同時に1台しか通行できない。2台以上通行すると待ちが発生する
  - 人が横切ると待つ
- 処理機械が所定の頻度で不定期に故障する
- ロットが動的に追加される



# シミュレーション型アプローチ

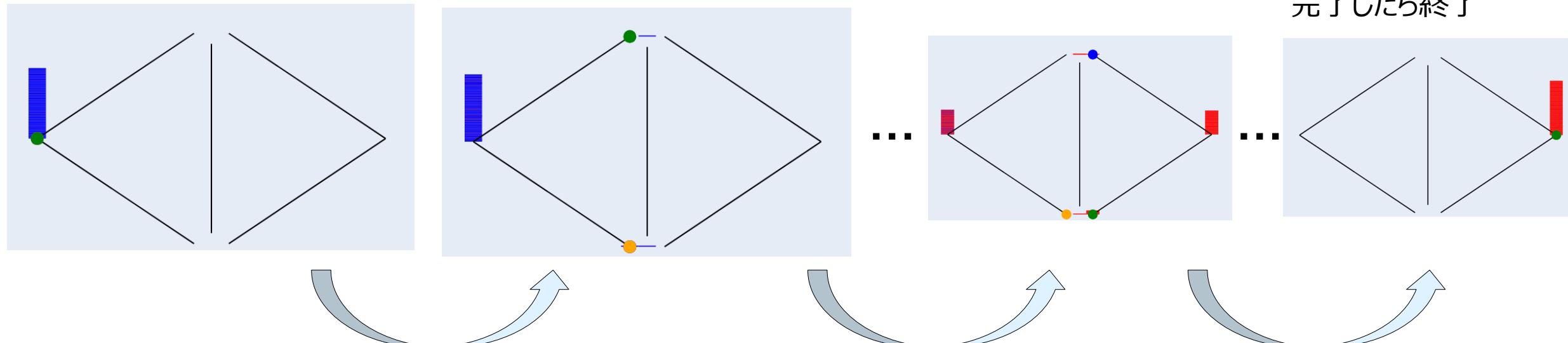
対象を再現するシミュレ

- ① 現在状態を取得  
(AGV・ロット位置, 機械の処理状態)
- ② 方策によりオペレーションを決定  
(AGV をどこに動かす, どのロットを処理する)



- ④ ①-③ を繰り返す

- ⑤ シミュレーションが完了したら終了



- ③ シミュレーションを1ステップ進める

# シミュレーション型アプローチの代表手法

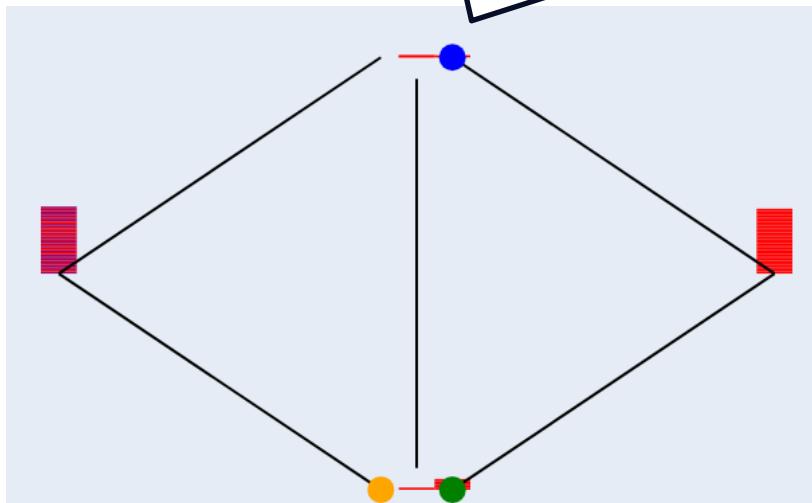
- A)ルールベース … 貪欲法
- B)探索ベース … プレイアウト、モンテカルロ木探索
- C)学習ベース … 強化学習

## A) ルールベース … 貪欲法

各オペレーションの良さを見積もる評価関数を設計し、評価値が最も良いオペレーションを行う。

例えば評価関数として「納期」を設定すると、「納期に最も近いロットを優先して運ぶ」というオペレーションになる。

搬送待ちロットのうち  
納期が最も早いものを選択

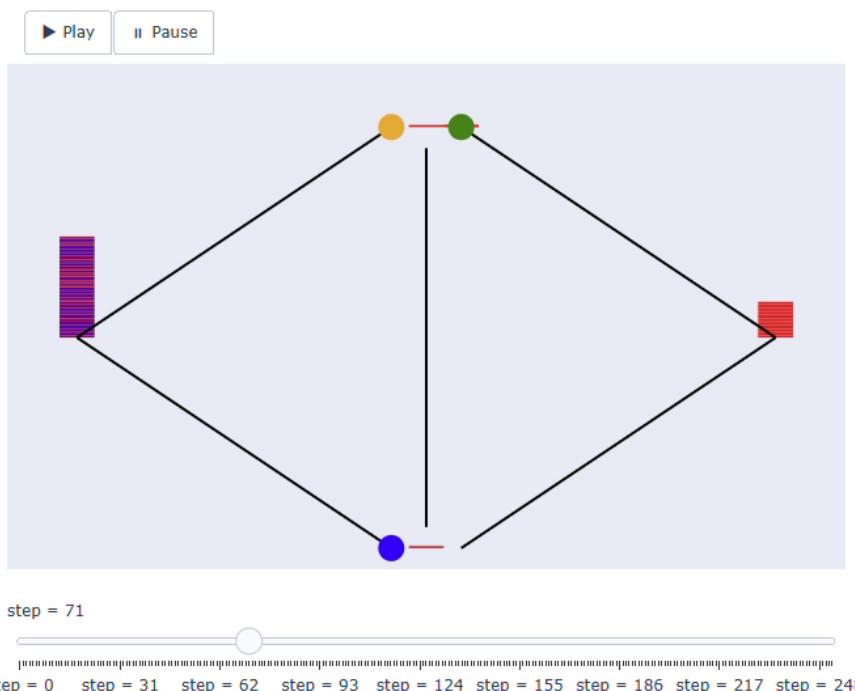


## A) ルールベースの利点と欠点

Good: 実装が容易. 計算が高速

Bad: 考慮したい要素が増えると設計が困難になる。性能に限界がある

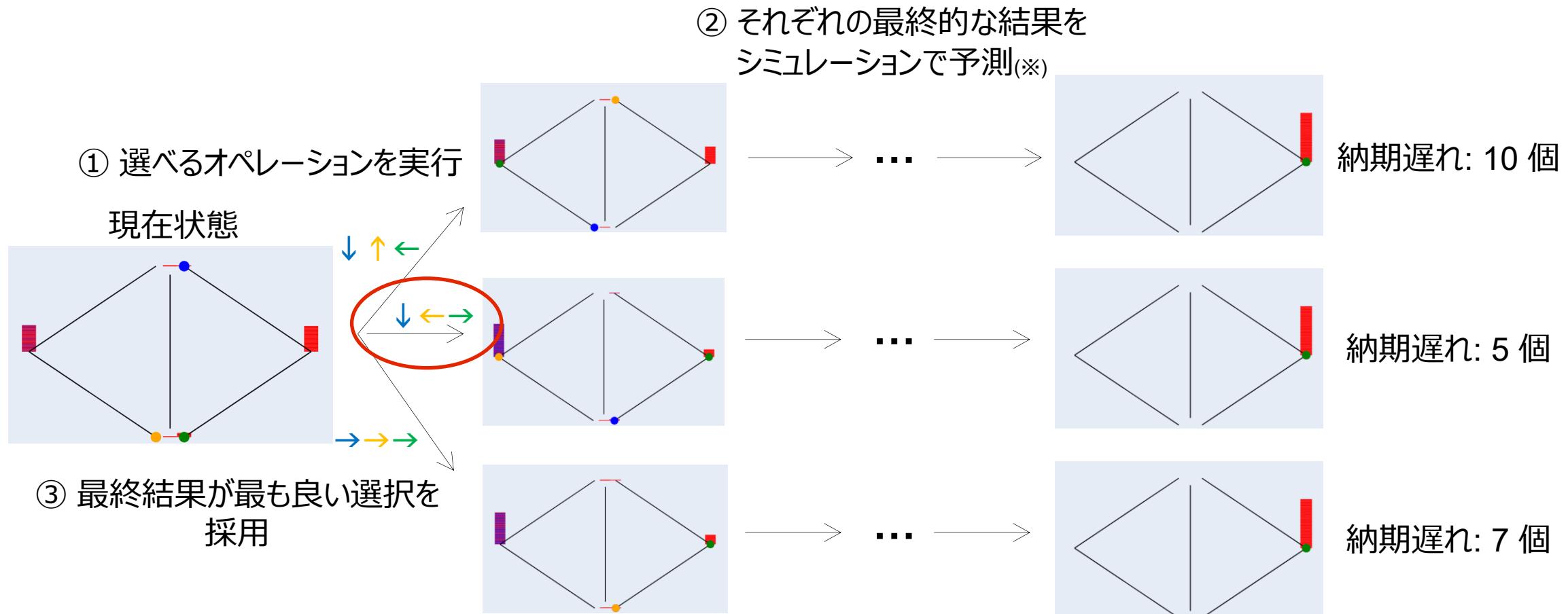
今回のケースで「納期最短」のルールベースを実行すると、機械が空くタイミングが存在する。これは明らかに効率が悪い。



## B) 探索ベースの方法

2手先以上の状態を考慮して良いオペレーションを選択する。

1ステップごとに以下のような処理を行う:



## B) 探索ベースの方法

Good: 性能が高い, 安定して動作する

Bad: 実装が困難, 計算コストが大きい

良い状態に行き着く操作を探索するため、安定して良いオペレーションを求めることができる。2016年に囲碁のトッププレイヤーを破って話題になった AlphaGo も探索ベースの方法※。

一方で複雑なシミュレーション上に探索アルゴリズムを実装することは実装難易度が高い。1オペレーションごとに探索を行うため計算コストも大きい。

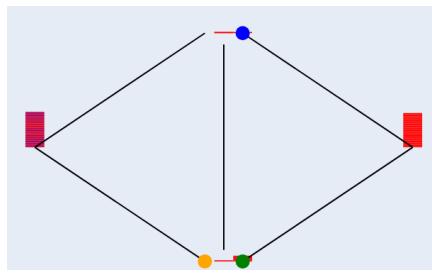
※「モンテカルロ木探索」と呼ばれるより工夫した探索アルゴリズムを用い、プレイアウト部分は強化学習(後述)エージェントを使っている  
このように、探索ベースの方法には他の選択アルゴリズムを組み合わせることが可能

## C) 学習ベース … 強化学習

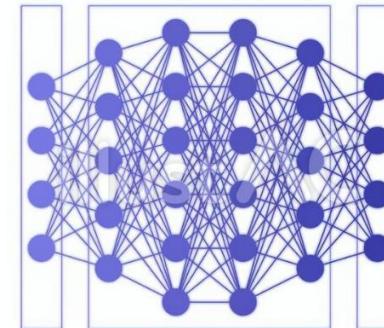
オペレーションを決める推論器をシミュレーションしながら学習させる方法。

推論器は現在状態を入力として受け取り、行動を出力する。表現能力が高いニューラルネットを用いるのが一般的。

① 現在状態を観測



② 推論器(ニューラルネット)の計算



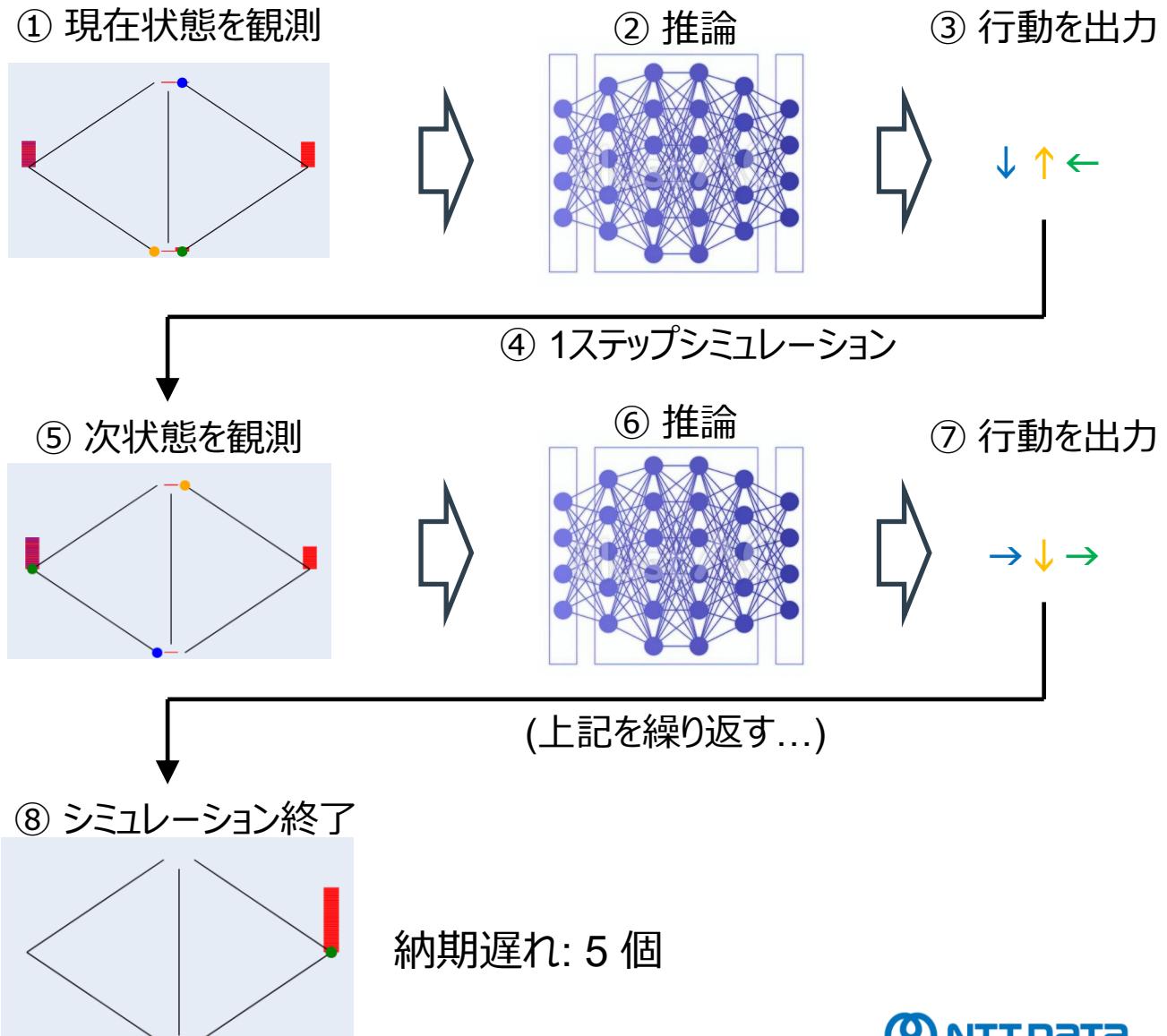
③ 行動を出力



# 強化学習の流れ

## Phase1. 学習

- i. シミュレーションの実行：①～⑧  
「状態の観測→ 推論→ 行動」を繰り返す



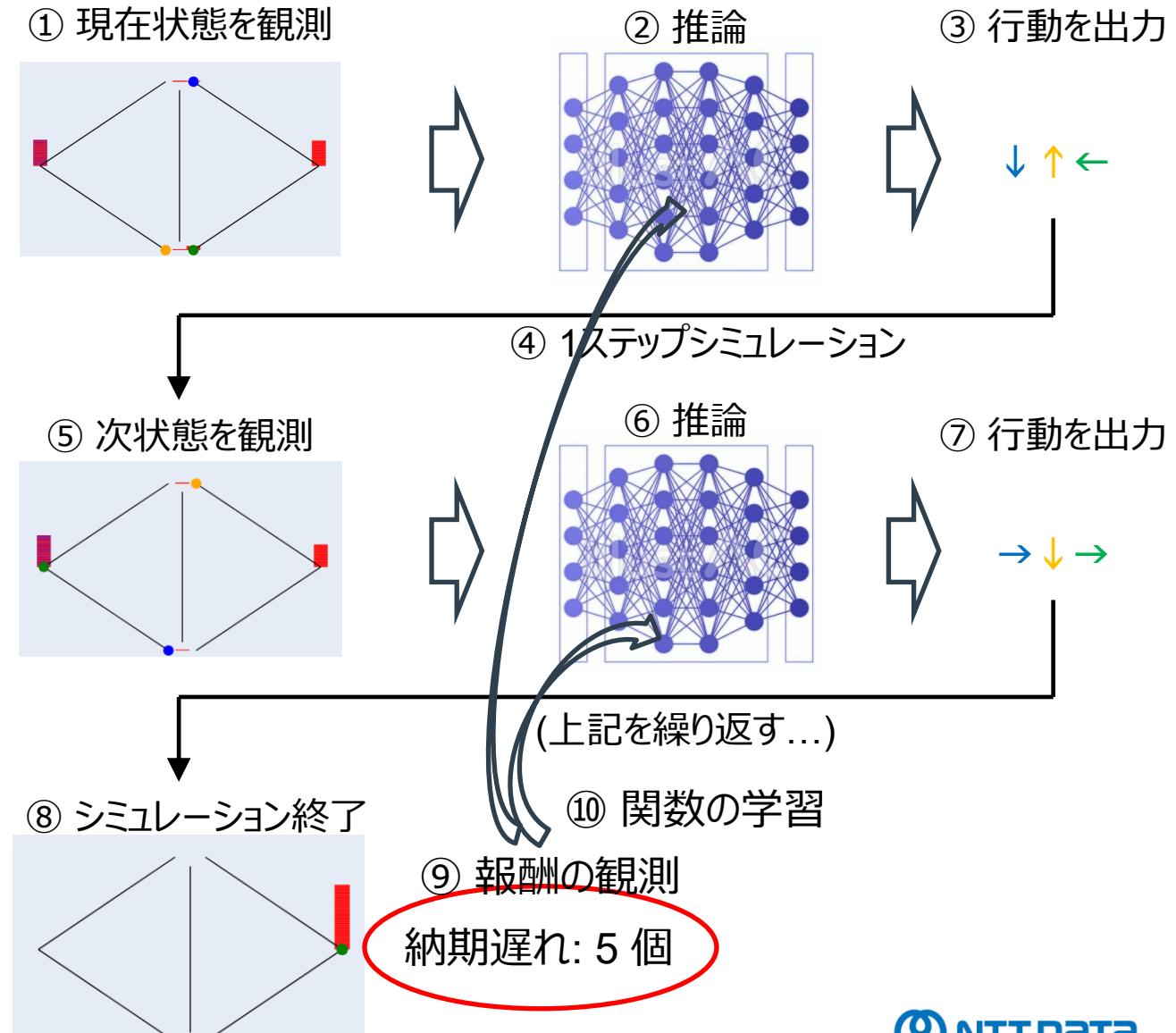
# 強化学習の流れ

## Phase1. 学習

- i. シミュレーションの実行：①～⑧  
「状態の観測→ 推論→ 行動」を繰り返す
- ii. 推論器の学習：⑨～⑩  
結果の良し悪しを推論器(ニューラルネット)にフィードバック

→ i. - ii. の繰り返しで推論器が良い行動を選ぶように学習される

※ 説明のために分けて記述しているが、実際はシミュレーションの途中にも学習は行われる。

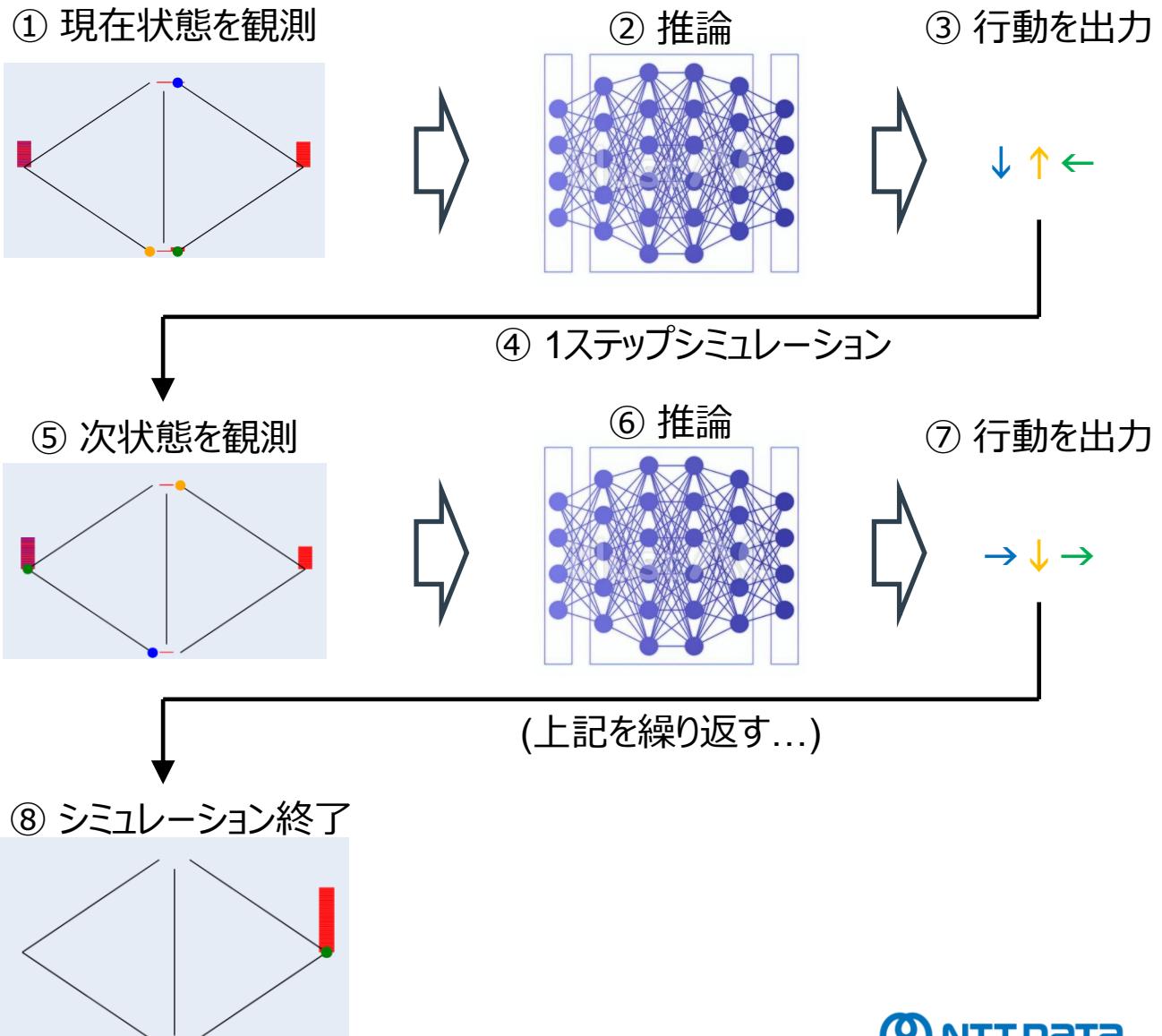


# 強化学習の流れ

## Phase2. 運用

学習した推論器を使ってオペレーションを行う。

※ (詳しい方向けの excuse) 推論器が policy そのものか状態価値関数かは  
この図では気にしていません (丸っと行動を選ぶロジック、くらいのイメージで描いてます)



# 強化学習の特徴

## Good:

うまく学習できれば性能は高い, 運用時の計算が高速(NNの推論計算のみ)

## Bad:

学習時の計算コストが大きい, 動作が不安定な可能性がある

「1オペレーションの決定に使える時間が数秒」のようなケースでも良い解を得られる可能性がある。

一方で、事前に時間をかけて学習させる必要がある点や、学習時と運用時で遭遇する状況の分布が異なると不安定な動作を示す可能性がある点がネック。

# オペレーションの最適化問題の解法：まとめ

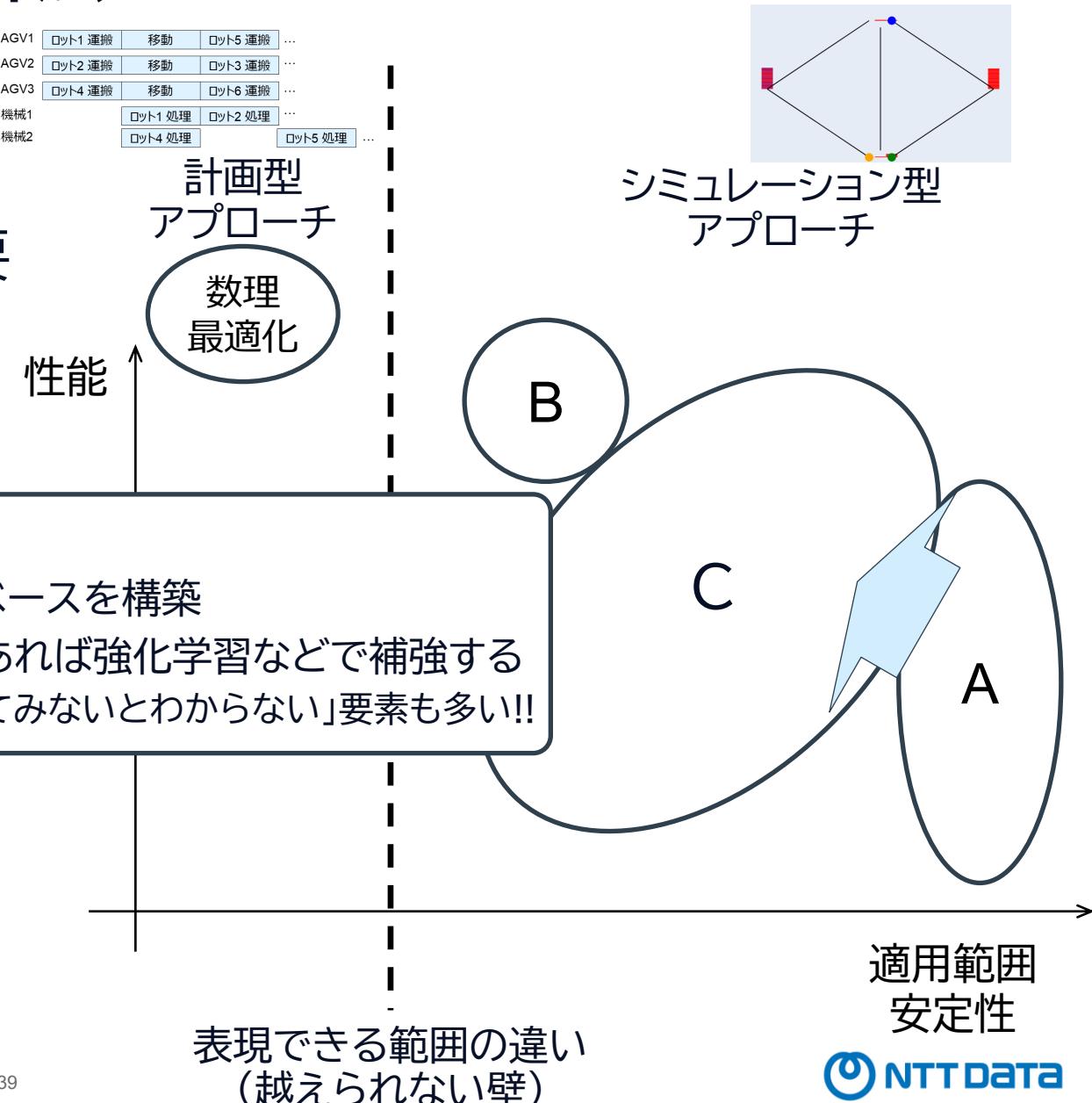
適切な方法を使い分けることが肝要

- ・ 数理最適化が適用できるならベスト
- ・ シミュレーションベースは試行錯誤が必要
  - ・ 簡単なロジックから作り徐々に作りこんでいくのがよい

手法	Good	Bad
数理最適化	高性能	単純な表現できず
A) ルールベース	実装容易・高速	性能限界
B) 探索ベース	高性能	実装難度・低速
C) 学習ベース	高性能・高速	学習時間・不安定

## 進め方の例

- ① まずはルールベースを構築
- ② 性能に不満があれば強化学習などで補強する  
※ 強化学習は「やってみないとわからない」要素も多い!!





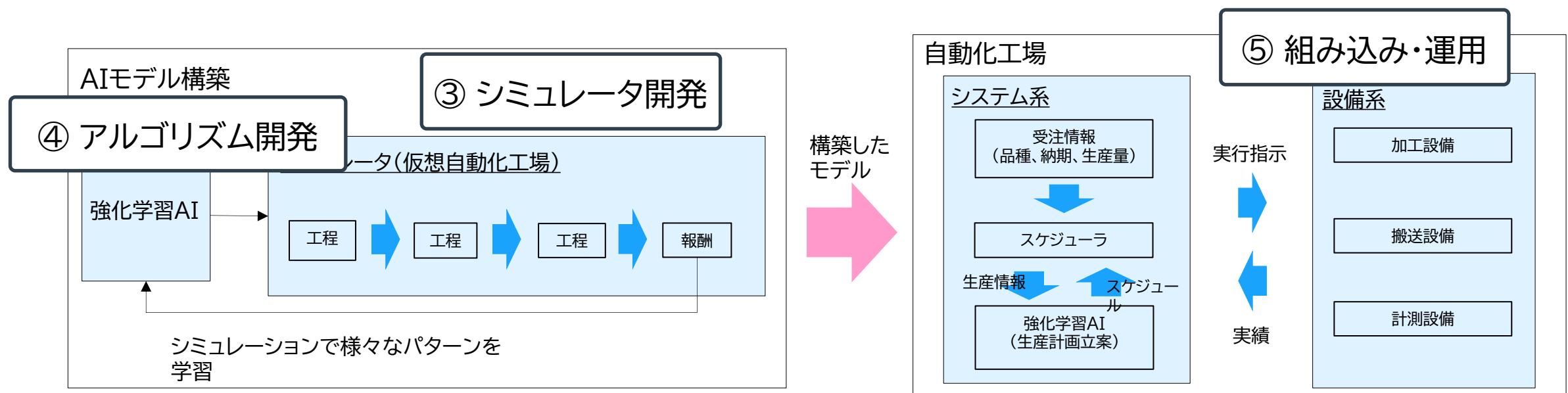
# 4

## プロジェクトの流れ

# 事例：大手ポンプ製造会社様

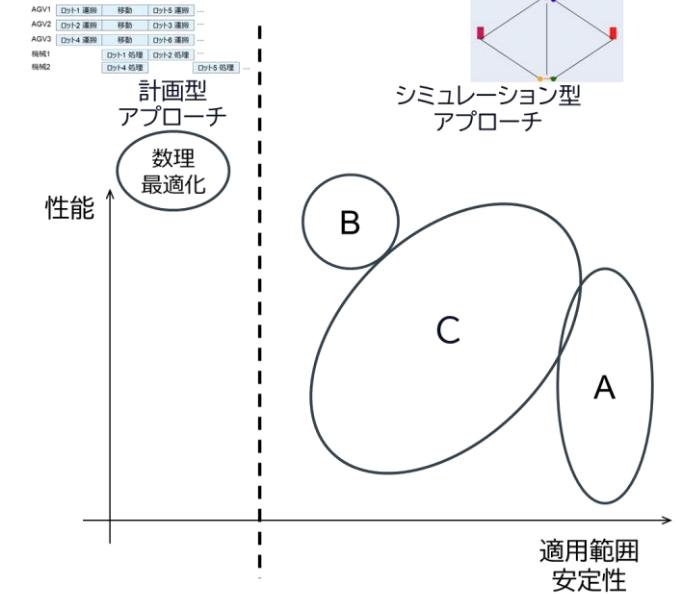
- 自動化工場の生産スケジュールを立案

背景と課題	<ul style="list-style-type: none"><li>需要の高まりに対応すべく、生産拠点の新設を計画。新たな取組として<b>自動化工場</b>を目指した</li><li><b>生産計画の立案</b>から<b>装置の制御</b>までを自動化したい</li><li>受注情報からリアルタイムに生産計画を立てたい</li><li>「<b>多品種短納期における段取替え</b>」、「<b>納期の遵守</b>」、「<b>生産量の最大化</b>」などが課題</li></ul>	① ヒアリング
活用	<ul style="list-style-type: none"><li>生産スケジューリングのためのAIとして、<b>強化学習</b>を採用</li><li>新規生産拠点の<b>シミュレータ</b>を構築。シミュレータでAIの学習データを生成し、AIの強化学習モデルを構築</li><li>自動化工場への実装を行った。</li></ul>	② 要件定義
お客様の声	<ul style="list-style-type: none"><li>取り組みに必要なシミュレーションと機械学習どちらの技術も持っている</li><li>両者に精通した技術者がシミュレーションモデルの仕様策定、PoC、AIモデルチューニング、実装まで対応</li></ul>	



# 最適化プロジェクトの流れ

- ① ヒアリング
- ② 要件定義・お見積り
- ③ シミュレータ開発
- ④ アルゴリズム開発
- ⑤ 組み込み・運用



現場の状況・課題感・最終的に達成したい状態をもとに**技術選定**を行います。

計画型アプローチ・シミュレーション型アプローチいずれも当社の技術領域ですので、幅広い課題に対応可能です。

要件定義の際には必要なデータの認識合せを行います。一般的には以下のようなデータが必要になります:

- 現場オペレーションの内容 (ロットの流れ方, AGV の移動履歴・搬送履歴, ...)
- 計画データ = オペレーションの**計画**
- 実績データ = オペレーションの**結果**

シミュレータ**構築**に必要

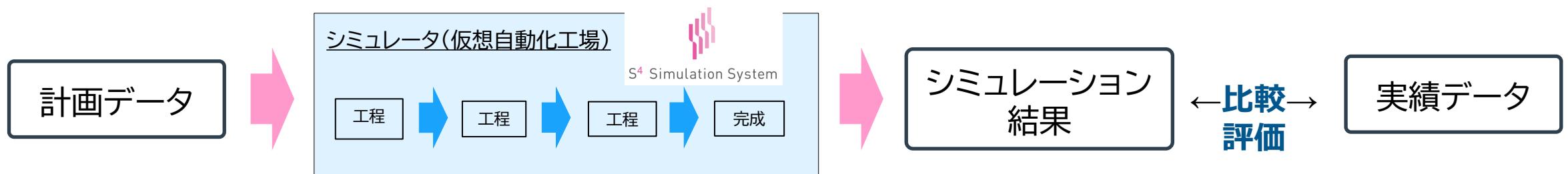
シミュレータ**検証**に必要

※データに限りがある場合でも、取得可能なデータのみで何ができるかを検討させていただくことは可能です。

# 最適化プロジェクトの流れ

- ① ヒアリング
- ② 要件定義・お見積り
- ③ シミュレータ開発
- ④ アルゴリズム開発
- ⑤ 組み込み・運用

シミュレータの開発には当社開発パッケージ S<sup>4</sup> Simulation System を用います。生産ラインをモデリングするためのパーツや API が豊富に用意されているため、スクラッチで実装するのに比べ効率よく実装できます。計画データを与えてシミュレーションした結果を実績データと比較・評価し、現況再現を行います。

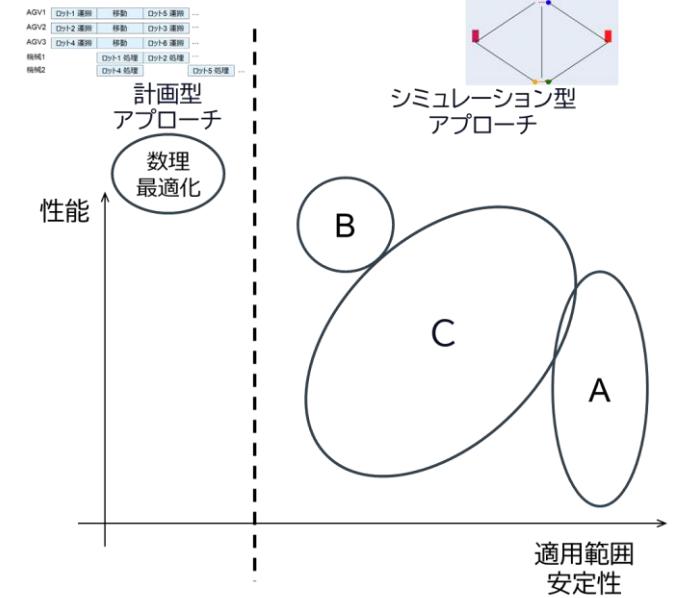


※ <https://gymnasium.farama.org/index.html>

# 最適化プロジェクトの流れ

- ① ヒアリング
- ② 要件定義・お見積り
- ③ シミュレータ開発
- ④ アルゴリズム開発
- ⑤ 組み込み・運用

ルールベースのロジック構築や強化学習アルゴリズムを実装し、シミュレータ上で効果を検証します。  
S4 で構築したシミュレータは強化学習の標準仕様である Gymnasium API※ に対応させることができたため、多くの既存手法を簡単に試すことが可能です。既にシミュレータを導入済みの場合、それを使って④を進めることも検討可能です。

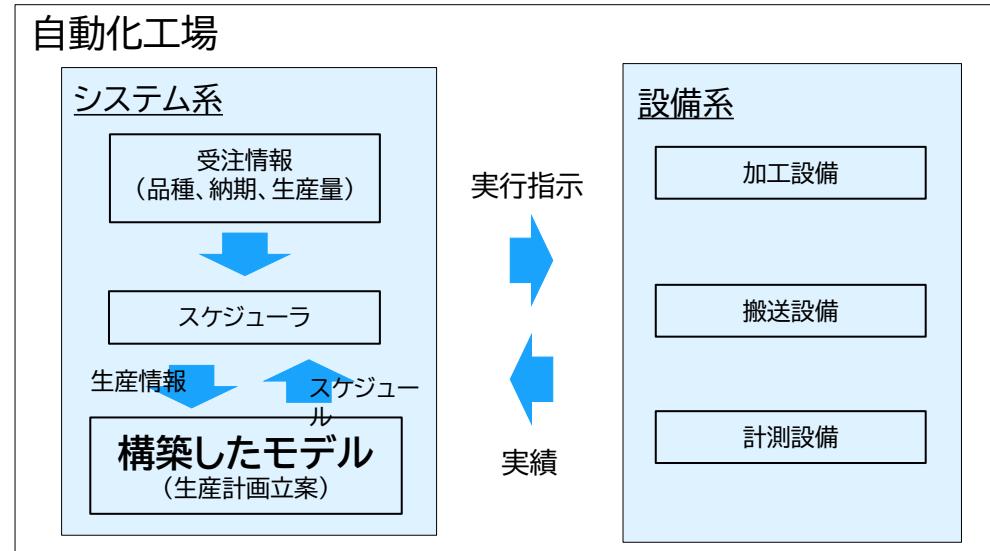


# 最適化プロジェクトの流れ

- ① ヒアリング
- ② 要件定義・お見積り
- ③ シミュレータ開発
- ④ アルゴリズム開発
- ⑤ 組み込み・運用

構築したアルゴリズムをシステムに組み込みます。

組み込める形でプログラム一式を納品する or 当社側で組み込み実装をするかについては前者のケースが殆どですが、後者も応相談です。



# 工数感

必要な開発期間は個別にお見積りいたします。以下に大まかなイメージを記載いたします。

① ヒアリング

② 要件定義・お見積り

→ 通常 1-2 週間以内で承ります

③ シミュレータ開発

→ 要件の複雑さによります。単純なもので2週間以内、複雑な場合3ヶ月～半年に及ぶこともあります。

複雑さはオペレーションのルールの種類に応じて決まります。全く同じルールで動作するものであれば機器の数が多くても開発工数には影響しません。

④ アルゴリズム開発

→ 単純なルールベース(要件に従って実装するだけ)であればシミュレータ開発 + 1週間程度で対応可能です。PoC 的に「うまくいくように試行錯誤」する場合、実装→ 実験のサイクルを回すために 1-3ヶ月程度で見積もることが多いです。

⑤ 組み込み・運用

→ 環境に応じて相談させていただきます。

# 最適化プロジェクトの流れ

- ① ヒアリング
- ② 要件定義・お見積り
- ③ シミュレータ開発
- ④ アルゴリズム開発
- ⑤ 組み込み・運用

弊社ではシミュレータ開発のみ、アルゴリズム開発のみ、のように一部工程のみの受託開発も承っております。  
(強化学習のように不確実な手法を試す場合、最初に PoC から着手することが多いです。)

上記の流れについて、弊社テックブログにてより詳しく記載しております:

<https://www.msiism.jp/article/reinforcement-learning-for-manufacturing.html>

お客様の課題に寄り添い  
適切な解決方法を提示させていただきます  
ご相談をお待ちしております

# FAQ (シミュレーション)

## 1. データが揃っていないと構築できない？

→ 製造シミュレーションには機器情報・製造計画情報が必要ですが、足りないものは経験知をもとに自動生成することも可能です。お手持ちのデータで実現可能な内容をご提案させていただきますので、お気軽に個別相談をお声掛けください。

## 2. 複数の工場に対して適用したい場合、シミュレータ作成は個別に行う必要がある？

→ ルール(部品の動き方)が共通であればシミュレータは横展開可能です。最初にシミュレータを構築する際に色々な工場に対応できるようルールを洗い出しておくと横展開しやすくなります。

## 3. シミュレータのカスタマイズは可能？

→ 入力データを差し替えることで処理機械の個数や処理機械の繋がりをユーザ側で変更できます。入力データの形式はエクセルの表形式(.csv など)をはじめ柔軟に対応可能です。

ルールを変更する場合、追加工数が発生しますが随時承ることが可能です。

## 4. 特別な計算環境が必要？

→ 通常の(ノート/デスクトップ)PCで動作する Python プログラムとなります。システムに組み込むことも可能です。

## 5. 計算時間は？

→ シミュレーション内容によりますが、例えばイベント数(全部品が経験する処理回数の総和)が10万程度であれば数分以内に完了することが期待できます。

# FAQ (最適化・強化学習)

1. 数理最適化が良いのかシミュレーションが必要かの目安は?  
→ 数理最適化が考慮できる要素には限りがあります(不確実性が少ない、リアルタイム性は扱わない等)。その範囲内で現実をモデリングできれば数理最適化を適用可能です。個別相談いただければ技術選定から行います。
2. 数理最適化と強化学習を横並びで比較するようなことは可能?  
→ 可能です。数理最適化による解 = 静的な方策(事前に決めた通りに行動する)とみなしてシミュレーションを行えば、強化学習で得た方策と直接比較することができます。
3. 強化学習の設定(状態、報酬)の設定方針は?  
→ 状態は機器の状態や滞留個数など、判断に寄与するものを網羅するように設定します。必要に応じてNNが学習しやすいように加工する工夫も考えられます(滞留数から見込みの待ち時間に変換する等)。  
報酬は最適化したい値(納期遅れ数)のほか、学習の初期ではそれに繋がる途中段階の値(完了した工程の数等)を設定することで学習が安定することも期待されます。
4. 強化学習の不安定さへの対処は?  
→ 以下のような方法が考えられます:
  - 複数の状況でシミュレーションを行いどのような状況であれば安定するかを把握する
  - 推論器が取れる行動の選択肢を事前に現実的なもののみにフィルタしておく(ルールベースに100個選択する、等)
  - 推論器の出力をチェックし異常な場合は安定して動作するロジック(ルールベース等)に切り替える
  - 推論器の挙動を説明する手法を使い、動作の根拠が人の直感に従うかを確認する

# お問い合わせ

株式会社 NTTデータ 数理システム

シミュレーション＆マイニング部

シミュレーションソリューション担当

TEL : 03 - 3358 - 6681

【URL】 <http://www.msi.co.jp/s4/>

【E-mail】 s4-info@ml.msi.co.jp

**まずはお気軽にご相談ください**

The logo for NTT Data is displayed in white text against a dark, blurred background of skyscrapers. The logo consists of a stylized 'O' shape containing a smaller circle, followed by the word "NTT data" in a bold, sans-serif font.

NTT data