

数理最適化を用いたスケジューラによる 生産計画の自動作成

2024 年 10 月 11 日

株式会社 NTT データ数理システム 数理計画部 藤井 智仁

MSIISM Conference 2024

発表内容

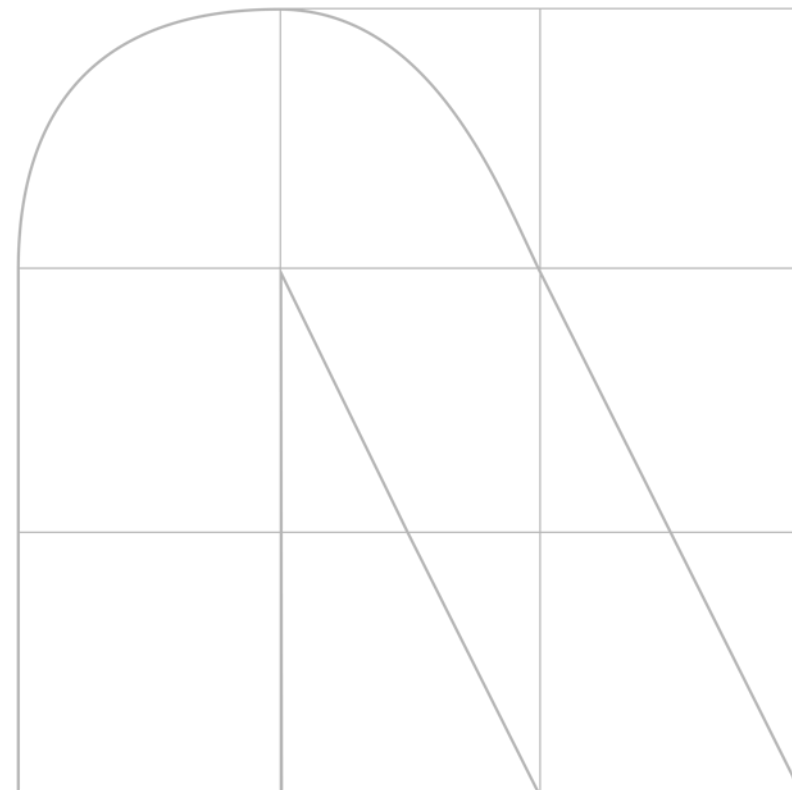
- **数理最適化の紹介**

- ・ 数理最適化でできることについて簡単に説明

- **ライオン様プロジェクトにおける数理最適化の適用イメージ**

- ・ 今回扱った問題の枠組みについて説明

数理最適化の紹介



数理最適化とは

大量にある候補の中で，一定のルールを満たしつつ最も良いものを求める

例	候補（変数）	ルール（制約条件）	良さの指標（目的関数）
	製造順序 生産量	生産能力 納期	在庫コスト 切替，稼働率
	訪問順序	訪問時間枠 積載量上限	移動時間
	人員配置	必要人数 休暇希望	負荷平準化

数理最適化がもたらす価値

長期の方針検討

- 投資判断

- 新しい設備を導入すれば生産効率はどう変わるか
- 物流拠点や倉庫をどこに新設すべきか

中期の計画立案

- コスト削減
- 環境変動への追従

- 中期の需要に対して早期に生産可否を判断
- 需要予測が変化したため生産計画を再立案する

日々の業務遂行

- 業務効率化
- 属人化の解消

- シフトを自動作成して空き時間を他業務に充てたい
- 熟練者でないと計画立案できないのを解消したい

参考：機械学習との違い

数理最適化では

満たすべきルールがわかっている課題に対して、意思決定を行う

数理最適化

機械学習

モデルの作成

ルールをもとに人手で構築

データからモデルを学習

モデルの利用

最適な出力が得られる
解を探索

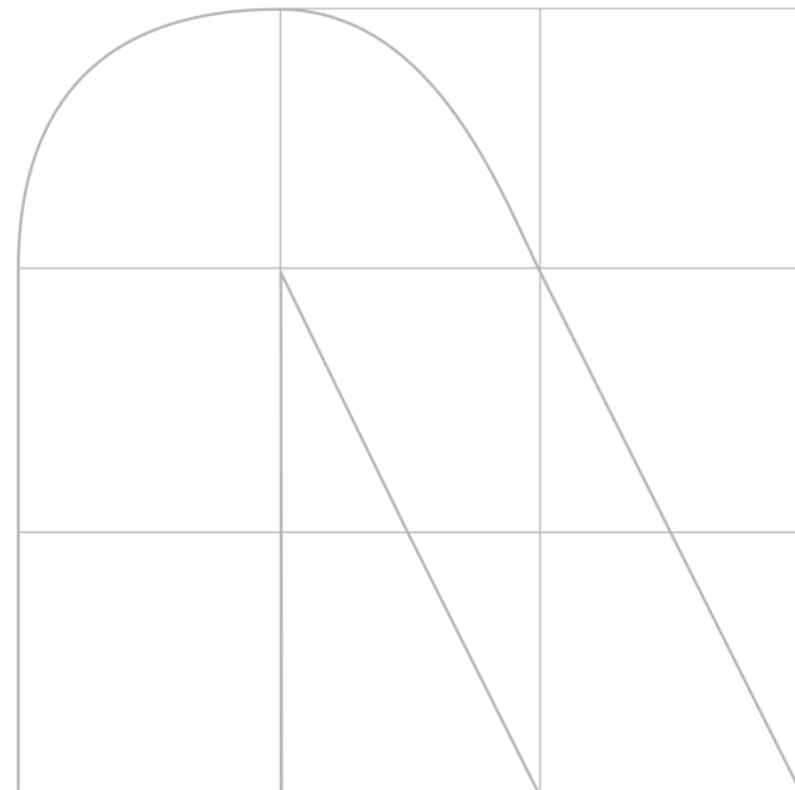
入力に対して出力を予測

利用例

生産計画の立案

需要予測

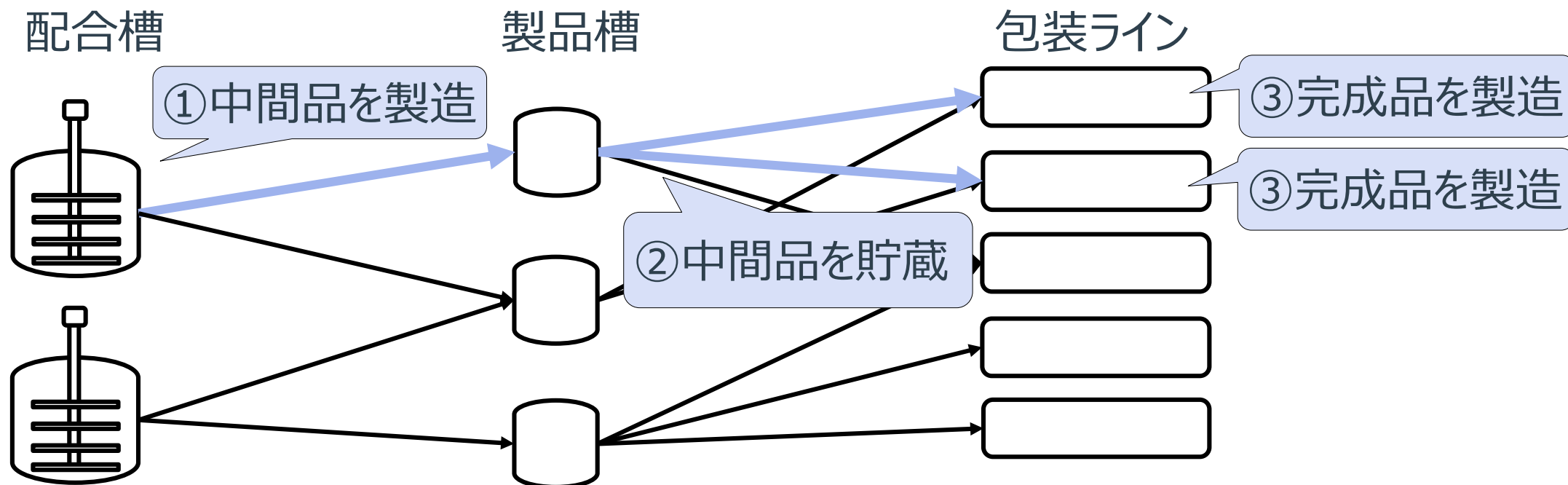
ライオン様プロジェクトにおける 数理最適化の適用イメージ



ライオン様プロジェクトで扱った生産計画問題 1/2

ライオン様側で整理いただき、全工場・カテゴリを共通の製造パターンに落とし込んだ

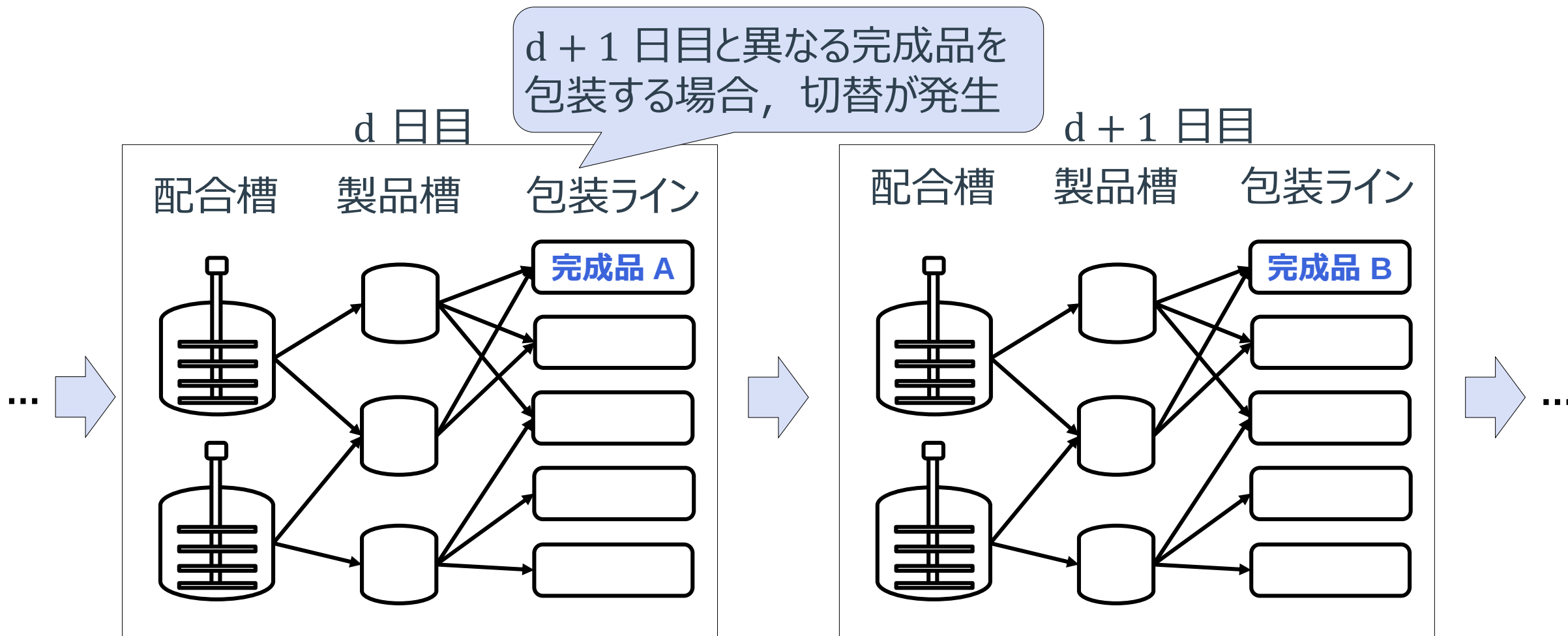
実際は工場によって生産方式が異なる：組立生産，プロセス生産，…



中間品の製造を考慮しつつ，ボトルネックとなる包装ラインの計画を求める

ライオン様プロジェクトで扱った生産計画問題 2/2

日を跨ぐ制約条件が存在するため、複数日を同時に考慮



問題の枠組み

今回扱う問題は「**多品目 多段階 多ラインのロットスケジューリング問題**」で、非常に難しい

切替無し（順序を重視しない）

切替有り（順序を重視）

易

教科書

1 品目 1 段階 1 ラインのロットサイズ決定問題

多品目 1 段階 1 ラインのロットサイズ決定問題

多品目 1 段階 1 ラインのロットスケジューリング問題

多品目 多段階 1 ラインのロットサイズ決定問題
多品目 1 段階 多ラインのロットサイズ決定問題

多品目 多段階 1 ラインのロットスケジューリング問題
多品目 1 段階 多ラインのロットスケジューリング問題

多品目 多段階 多ラインのロットサイズ決定問題

多品目 多段階 多ラインのロットスケジューリング問題

実務

難

最適化計算全体の流れ

6 か月分を一度の最適化計算で扱うのは厳しいため、1 か月ごとに分割して計算

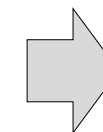
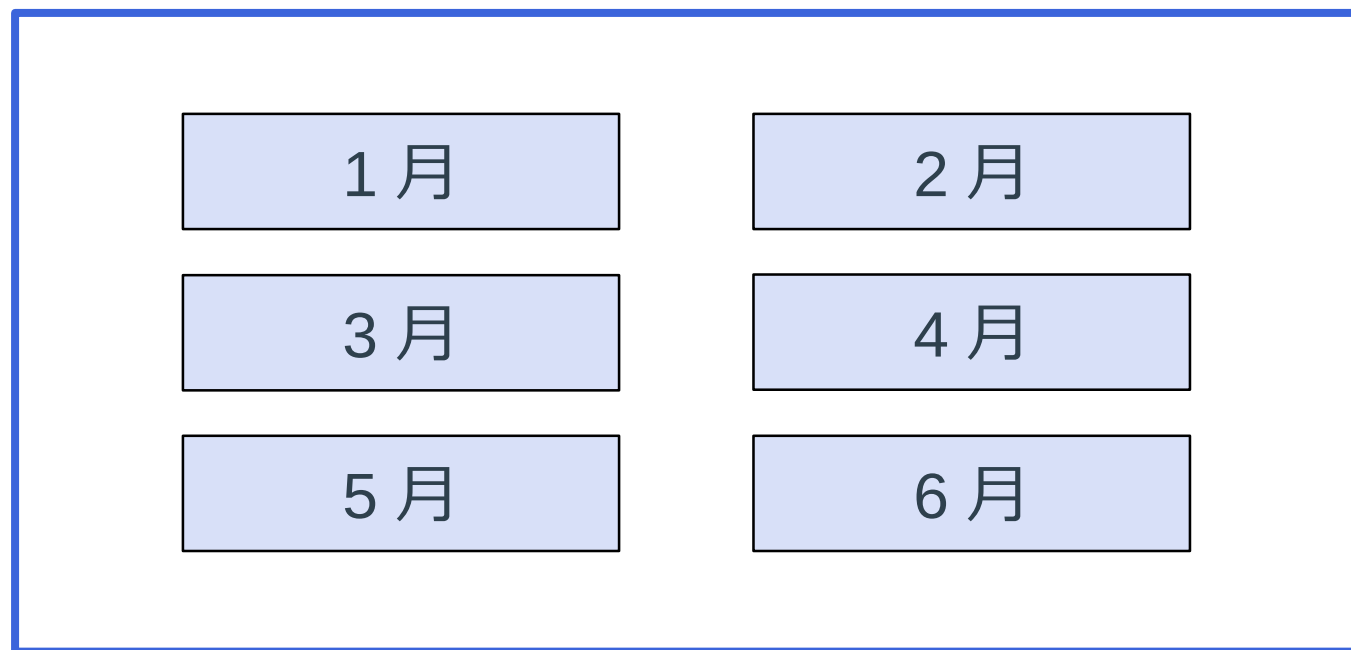
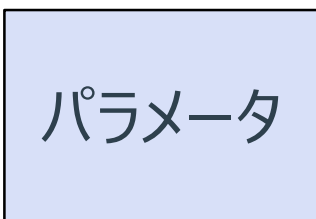
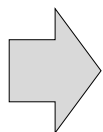
Excel から
取り込み



Nuorium
Optimizer

を用いた最適化計算

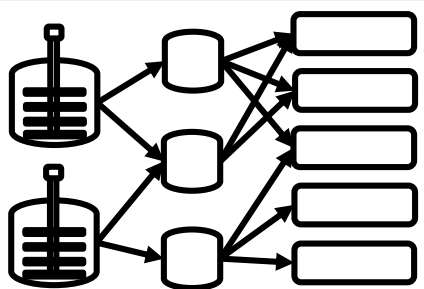
Excel
に出力



最適化計算のイメージ

数理最適化問題としてモデリング → Nuorium Optimizer により最適化計算

解きたい課題



```
problem = Problem(name='MinCostFlow', type=min)
```

```
City = Set(value=value.city)
```

```
i = Element(set=City)
```

```
j = Element(set=City)
```

```
k = Element(set=City)
```

```
upper = Parameter(index=(i,j), value=data.upper)
```

```
cost = Parameter(index=(i,j), value=data.cost)
```

```
supply = Parameter(index=k, value=data.supply)
```

```
f = Variable(index=(i,j), lb=0, ub=upper[i,j])
```

```
problem += Sum(cost[i,j]*f[i,j])
```

```
problem += Sum(f[k,j], j)-Sum(f[i,k], i) == supply[k]
```

```
problem.solve(**kwds)
```

今回は Python ベースのモデリング言語 **PySIMPLE** で記述

計算結果

	ライン 1	ライン 2	ライン 3
10/1	完成品 A	完成品 C	完成品 D
10/2	完成品 A	完成品 C	完成品 E
10/3	完成品 B	完成品 C	完成品 F
⋮			

※ 実際に本プロジェクトで記述したモデルとは異なります

まとめ

数理最適化とは

- **大量にある候補の中で，一定のルールを満たしつつ最も良いものを求める手法のこと**

ライオン様事例における数理最適化適用イメージ

- **PySIMPLE を用いて多品目 多段階 多ラインのロットスケジューリング問題としてモデリング → Nuorium Optimizer で最適化計算**

数理最適化に興味をお持ちの方へ…

- 是非当社で開催しているセミナーにご参加ください
数理最適化や Nuorium Optimizer に関するセミナーも実施しています
<https://www.msiism.jp/event/>

NTT Data