

# サプライチェーンマネジメント最終レポート

中央大学理工学部  
ビジネスデータサイエンス学科  
23D7104001I 高木悠人

## 1. 実行環境

|                |  |
|----------------|--|
| PC (notebook): | Apple macbook air m4                     |
| Python 環境:     | venv 仮装環境(ローカルでの実行)                      |
| Python バージョン:  | python-3.13.9                            |
| 各種パッケージバージョン:  | requirements.txt リンクは付録の項に示すこととした        |
| セッション管理:       | Python Package(SessionSmith==2.0.0)を利用した |
| コードバージョン管理:    | GitHub(リンクは付録の項に示すこととした)                 |

## 2. 課題 1

### 課題 1 :

- (1) Jobshop型生産方式とそのシミュレーション・最適化で示したシミュレーションについて、各ジョブの内容を

```
ジョブ0 : [(0, 3), (1, 2), (2, 2)]
ジョブ1 : [(0, 2), (2, 1), (1, 4)]
ジョブ2 : [(1, 4), (2, 3)]
ジョブ3 : [(1, 2), (0, 1), (2, 4)]
ジョブ4 : [(2, 1), (0, 2), (1, 1)]
```

のようにしたときにFCFSの場合どのような結果になるか。また、OR-Toolsでスケジュール最適化を行った場合にFCFSとどのように結果が変わるかを確認・考察せよ。

- (2) それぞれのジョブに重みを付けた場合（3パターンほど、重みの値は各自自由で良い）に結果がどのように変わるかを確認・考察せよ。
- (3) 各自分で好きなジョブのパターンを作成し、その結果について確認・考察せよ。

### (1) FCFS と OR-Tools の結果の比較と考察

はじめに、FCFS によるシミュレーション結果と、OR-Tools を用いた最適化結果を比較考察した。それぞれのシミュレーション結果を以下に示す。

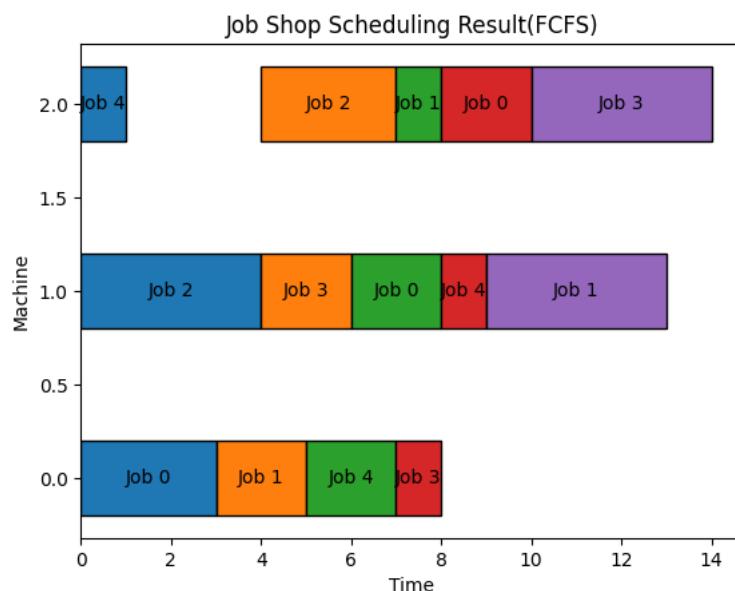


図 1. FCFS シミュレーション結果

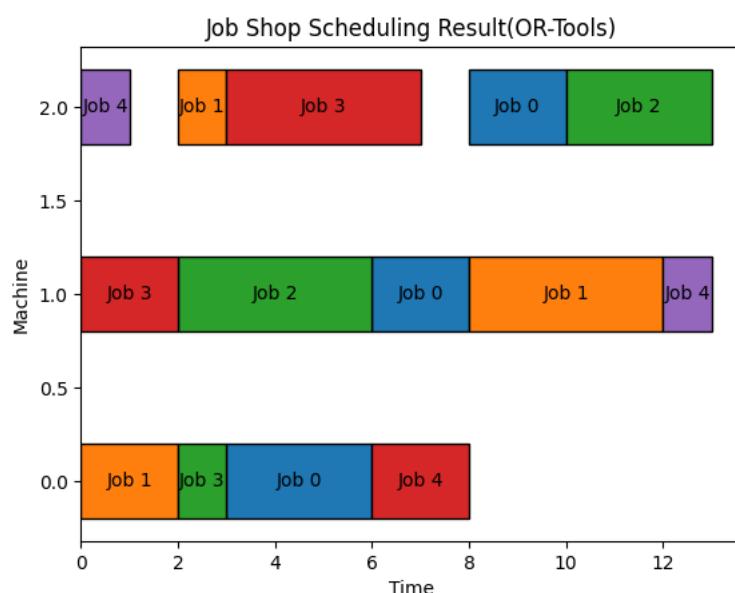


図 2. OR-Tools 最適化結果

図 1 に示す FCFS によるスケジューリング結果では、ジョブは到着順に処理されるため、各機械における処理順序が固定的となっている。その結果、ある機械では処理可能なジョブが存在するにもかかわらず、前工程の終了を待つためにアイドル時間が発生している様子が確認できる。特に機械 2 では、ジョブの処理が後半に集中しており、全体の完了時刻は 14 となっている。このように、FCFS は実装が容易である一方、Jobshop 型生産方式のように工程順序が複雑な場合には、機械資源を十分に活用できない可能性がある。

一方、図 2 に示す OR-Tools を用いた最適化結果では、各ジョブの工程順序制約および機械の非重複制約を満たしつつ、完了時刻までの時間最小化を目的としてスケジュールが再構成されている。その結果、ジョブの処理順序は FCFS の場合から大きく変化しており、機械ごとの空き時間が減少していることが分かる。特に、複数の機械において処理が前倒しされ、ジョブがより並行的に進行する構造となっている点が特徴的である。

この最適化の効果は、完了時刻までの時間の比較からも確認できる。FCFS では全ジョブの完了時刻が 14 であったのに対し、OR-Tools による最適化では完了時刻が 13 となっており、全体の生産完了時間が短縮されている。これは、OR-Tools が局所的な順序ではなく、全体最適の観点からジョブの割り当てを行った結果であり、Jobshop 問題における数理最適化の有効性を示している。

以上より、本課題の結果から、OR-Tools を用いたスケジュール最適化によって、FCFS と比較して生産完了時間を短縮できることが確認できた。ただし、短縮幅は 1 と比較的小さく、これは本課題のジョブ数および機械数が限定期であるためと考えられる。問題規模が拡大し、工程の組み合わせがより複雑になるほど、最適化手法の効果はより顕著に現れると考えられる。このことから、Jobshop 型生産方式においては、単純なヒューリスティック手法と数理最適化手法を問題特性に応じて使い分けることが重要であると考える。

## (2) 重み付け結果の比較と考察

次に、各ジョブに重みを付与した場合のスケジューリング結果について考察する。ここでは、重みの与え方を 3 通り設定し、それぞれの結果を比較した。重みはスケジューリングにおける優先度を表しており、重みの大きいジョブほど優先的に処理されるように設定されている。シミュレーション結果を以下に示す。

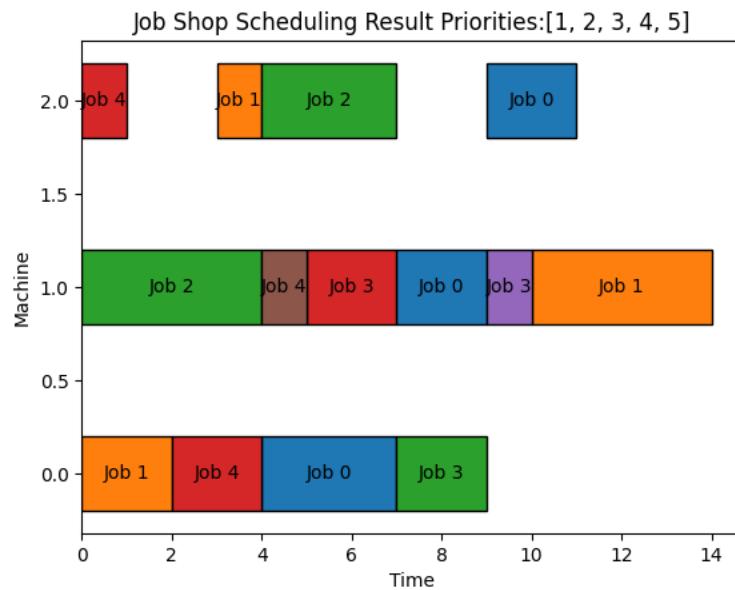


図 3. シミュレーション結果([1,2,3,4,5])

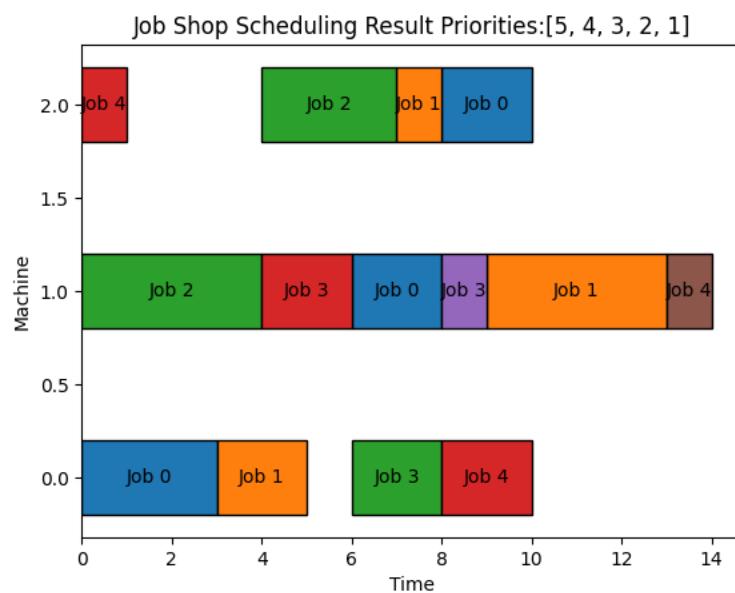


図 4. シミュレーション結果([5,4,3,2,1])

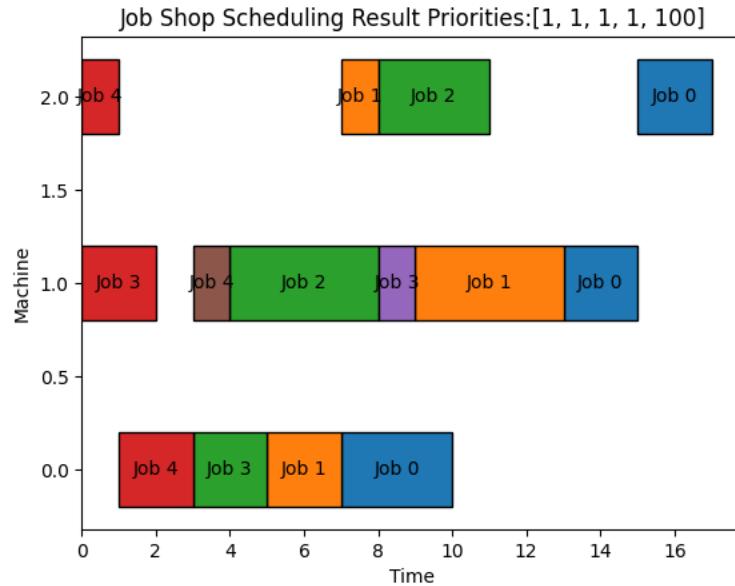


図 5. シミュレーション結果([1,1,1,1,100])

図 3 は、優先度を  $[1, 2, 3, 4, 5]$  とした場合のスケジューリング結果を示している。この設定では、ジョブ番号が大きいほど高い優先度を持つ。図から分かるように、高優先度のジョブは各機械において比較的早い時点で処理される一方、低優先度のジョブは後回しにされる傾向が見られる。その結果、全体の処理順序は FCFS や単純な最適化結果とは異なり、優先度を反映したスケジュールとなっている。ただし、特定のジョブを優先することで、他のジョブの待ち時間が増加している点も確認できる。

図 4 は、優先度を  $[5, 4, 3, 2, 1]$  と逆順に設定した場合の結果である。この場合、先ほどとは異なり、ジョブ 0 が最も高い優先度を持つ。図 3 と比較すると、ジョブ 0 が早期に処理される工程が増えており、優先度の設定がスケジュール全体に直接的な影響を与えていることが分かる。一方で、優先度の低いジョブは後半に集中して処理されるため、機械によっては処理が偏る区間が生じている。このことから、優先度の付け方によって、同じ Jobshop 問題であっても全く異なるスケジュールが生成されることが確認できる。

図 5 は、特定のジョブに非常に大きな重みを与えた場合の結果であり、優先度を  $[1, 1, 1, 1, 100]$  と設定している。この設定では、ジョブ 4 が他のジョブよりも極端に優先されるため、そのジョブの処理は可能な限り前倒しで配置されている。一方で、その他のジョブはその影響を受け、待ち時間が増加し、全体の完了時刻増大していることが確認できる。この結果から、優先度を極端に設定すると、個別ジョブの完了時間を短縮できる一方で、システム全体としての効率が低下する可能性があることが分かる。

以上の結果より、重み付きスケジューリングでは、どのジョブを優先するかによってスケジュールの性質が大きく変化することが明らかとなった。全体の生産効率を重視する場合と、特定のジョブの納期や重要度を優先する場合では、適切な重みの設定が異なる。したがって、実務においては、単に完了までの時間を最小化するだけでなく、納期遵守や顧客重要度などの要件を考慮し、目的に応じた重み付けを行うことが重要であると考察する。

### (3) 作成したジョブのパターン結果の確認と考察

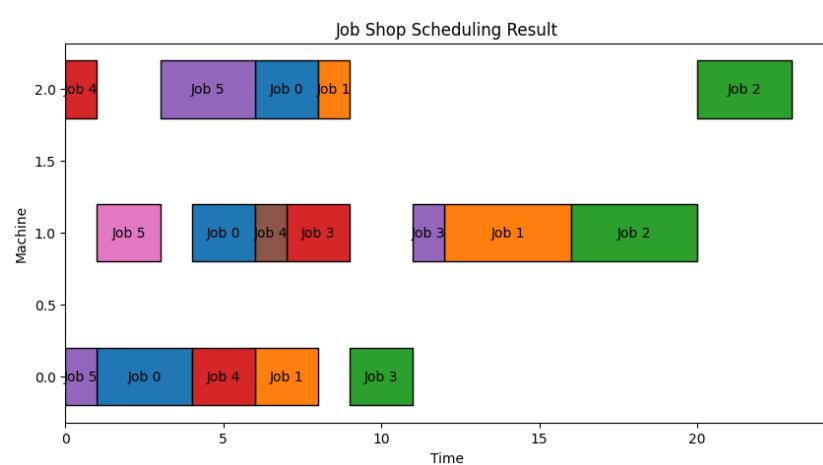
本課題では、以下のジョブパターンを作成した。

```
jobs_data = [
    [(0, 3), (1, 2), (2, 2)],
    [(0, 2), (2, 1), (1, 4)],
    [(1, 4), (2, 3)],
    [(1, 2), (0, 2), (1, 1)],
    [(2, 1), (0, 2), (1, 1)],
    [(0, 1), (1, 2), (2, 3)],
]

job_priorities = [100, 2, 1, 10, 30, 100]
num_machines = 3
```

図 6. 作成したジョブパターン

上記のパターンを用いてシミュレーションした結果を以下に示す。



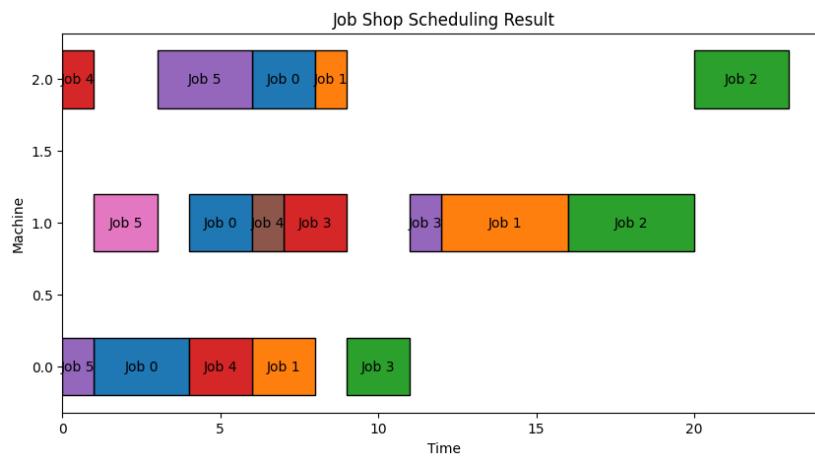


図 7. シミュレーション結果

本課題では、6つのジョブを任意に設定し、それぞれに異なる工程構成と重みを与えてスケジューリングを行った。特に、ジョブ0およびジョブ5には大きな重みを設定し、特定ジョブを強く優先させた場合の挙動を確認した。

結果を見ると、重みの大きいジョブは各機械において優先的に処理され、完了時刻が他のジョブよりも早くなっていることが分かる。一方で、重みの小さいジョブは後回しにされ、待ち時間が増加している。また、特定のジョブを優先した影響により、機械ごとの負荷に偏りが生じ、全体の完了時間が必ずしも短縮されていない点も確認できた。

以上より、重み付けによって特定ジョブの完了時間を短縮できる一方で、全体効率とのトレードオフが生じることが分かった。実務においては、納期や重要度を考慮しつつ、全体最適とのバランスを取った重み設定が重要であると考えられる。

### 3. 課題 2

#### 課題 2 :

サプライチェーン（生産・輸送・拠点配置）の領域におけるIoTセンシング・シミュレーション・最適化などのデータサイエンスの活用について、以下の流れで調査し文章にまとめよ。Q1)「サプライチェーン（生産・輸送・拠点配置）の領域におけるIoTセンシング・シミュレーション・最適化などのデータサイエンスの活用について具体例を示して調査せよ。」と生成AIに問いかけて、その出力結果を転載してください。Q2) その回答を自分なりに修正してください。誤っていると思われる点は修正し、深堀りした点は加筆してください。Q3) Q2の文章生成結果のどこに物足りなさや不適切さを見つめたか、その結果としてQ3で具体的に何を加筆修正したのかについて議論してください

#### 1. Q1) サプライチェーンの領域における DS 活用事例について AI に投げた結果

本課題では、Gemini を利用した。以下に生成結果を示す。

サプライチェーンマネジメント（SCM）におけるデータサイエンスの活用は、従来の「過去の実績に基づく計画」から、「リアルタイムデータに基づく動的制御と未来の予測」へと進化しています。特に「IoTセンシング」「シミュレーション」「数理最適化」の3つの技術を組み合わせることで、サプライチェーン全体の「デジタルツイン」を構築し、不確実性（需要変動、自然災害、交通網の混雑など）に強いオペレーションが可能になります。

生産・輸送・拠点配置の3つの領域に分け、具体的な活用例とともに解説します。

##### 1. 生産領域（Manufacturing & Operations）

～スマートファクトリーにおける歩留まり向上と生産スケジューリング～

生産領域では、工場内のあらゆるモノや動きをデータ化し、生産効率の最大化と品質の安定化を図ります。

- IoTセンシング：工作機械に振動・温度・音響センサーを取り付け、ミリ秒単位で稼働データを収集します。
- 具体例：半導体や自動車部品の製造ラインにおいて、ドリルなどの刃先の摩耗度合いをIoTで検知し、故障する前に部品交換を行う「予知保全」を実施。これにより突然的なライン停止を防ぎます。
- シミュレーション：工場のレイアウトや生産ラインのボトルネックを仮想空間上で検証します。
- 具体例：新しい製品ラインを導入する際、作業員の勤続、AGV（無人搬送車）の台数、バッファスペースの容量などをシミュレーションし、手戻りのないライン設計を行います。
- 数理最適化：数千に及ぶ注文と、数百の設備、人員のシフトを考慮し、最も効率的な生産スケジュールを自動生成します（ジョブショップ・スケジューリング問題）。

##### 2. 輸送領域（Transportation & Logistics）

～動的ルーティングとコールドチェーン（低温物流）の監視～

ドライバー不足（2024年問題）やカーボンニュートラルへの対応が急務となる中、積載率の向上とリアルタイムな配達管理が重要視されています。

- IoTセンシング：トラックのGPSトラッカー、RFIDタグ、温度センサーを活用します。
- 具体例（医薬品・食晶物流）：輸送中のコンテナ内の温度をIoTで常時監視。規定温度を逸脱しそうになった場合、即座にアラートを発報し、品質劣化を防ぎます。
- 数理最適化：配達先、荷物量、車両の積載上限、交通状況を考慮し、最適な配送ルートと積載計画を導き出します（配達経路最適化問題：VRP）。
- 具体例：宅配業者（ヤマト運輸や佐川急便など）が、その日の荷物量と配達指定時間に応じて、AIが数秒で全ドライバーの最適な巡回ルートを作成。走行距離とCO2排出量を削減します。
- シミュレーション：高速道路が事故で通行止めになった場合」や「台風が接近している場合」など、突然的な事象による遅延の影響を予測し、代替ルートや代替輸送手段（モードルート）の有効性を検証します。

##### 3. 拠点配置領域（Network Design & Facility Location）

～レジリエントな物流ネットワークの設計～

「どこに倉庫を構えれば最も物流コストを抑え、かつ災害時に機能不全に陥らないか」という戦略的な意思決定にデータサイエンスを用います。

- 数理最適化：将来の需要予測データ、土地代、人件費、各拠点間の輸送コストを入力とし、総コストを最小化する拠点数と立地を特定します（施設配置問題：FLP）。
- 具体例：大手EC事業者（Amazon等）が、全国の顧客に「翌日配送」を実現するため、人口動態や過去の購買データから重心法や数理最適化を用いて、最適なフルフィルメントセンター（物流拠点）の建設場所を決定します。
- シミュレーション：サプライチェーンのデジタルツイン上でストレステスト（What-if分析）を行います。
- 具体例：「特定の主要工場・倉庫が地震で1ヶ月操業停止した場合」の供給への影響をモンテカルロ・シミュレーション等で評価し、代替拠点への在庫の事前分散や、安全在庫基準の見直しを行います。
- IoTセンシング：直感的なセンシングではありませんが、各拠点のIoTから集まる入出庫データや在庫データが、将来の拠点配置を見直すための正確な基礎データとなります。

まとめ：各領域における技術の役割一覧

| 領域   | IoTセンシング（現状把握）      | シミュレーション（未来予測・検証） | 数理最適化（意思決定）   |
|------|---------------------|-------------------|---------------|
| 生産   | 設備の稼働状況、製品品質の監視     | ライン変更時のボトルネック検証   | 生産スケジュールの自動立案 |
| 輸送   | 車両位置、荷物の温湿度・衝撃検知    | 交通障害時の遅延影響分析      | 配達ルート・積載の効率化  |
| 拠点配置 | 拠点間のリアルタイムなモノの流れの把握 | 災害時のBCP（事業継続計画）検証 | 倉庫や工場の最適立地の決定 |

次のステップ

これらの技術は、個別で機能させるだけでなく、「サプライチェーン・デジタルツイン」としてプラットフォーム（Blue Yonder, o9 Solutions, Llamasoftなど）上で統合することで真価を発揮します。

特定の業界（例：半導体、食品、アパレルなど）に焦点を当てて、より詳細な技術的アーキテクチャや、導入による投資対効果（ROI）の事例について深掘り調査をいたしましょうか？ご関心のある業界があればお教えてください。

## 2. Q2) 回答の修正

サプライチェーン（生産・輸送・拠点配置）におけるデータサイエンスの活用は、従来の経験則や静的な計画に基づく運用から、リアルタイムデータを前提とした動的な意思決定と将来予測へと大きく転換しつつある。特に、IoT センシング、シミュレーション、数理最適化を組み合わせたアプローチは、サプライチェーン全体を仮想空間上で再現する「デジタルツイン」の構築を可能にし、不確実性の高い環境下でも柔軟な運用を実現する。以下では、生産・輸送・拠点配置の三領域に分け、データサイエンスがどのように活用されているかを具体的に述べる。

### 生産領域

生産領域では、IoT センサーによって設備の稼働状態や品質データをリアルタイムで取得し、それを基に生産効率と品質の最適化が行われている。例えば、工作機械の振動や温度データを継続的に監視することで、異常兆候を早期に検知し、突発的な設備故障を未然に防ぐ予知保全が可能となる。これは単なる保全コスト削減にとどまらず、生産計画全体の安定性向上に寄与する。

さらに、工場全体をモデル化したシミュレーションにより、工程変更や新製品投入時のボトルネック発生リスクを事前に検証できる。これに数理最適化を組み合わせることで、複数の設備制約や納期条件を考慮した生産スケジュールを自動生成することが可能となる。このように、生産領域では「センシングによる現状把握」「シミュレーションによる検証」「最適化による意思決定」が密接に連携している点が重要である。

### 輸送領域

輸送領域では、ドライバー不足や環境負荷低減といった社会的制約を背景に、配達効率の高度化が求められている。GPS や RFID、温湿度センサーなどの IoT デバイスを活用することで、車両位置や貨物状態をリアルタイムで把握できる。特に医薬品や食品物流では、温度逸脱の即時検知が品質保証の観点から極めて重要である。

これらのデータを基に、配送経路最適化問題 (VRP) を解くことで、走行距離や待機時間を最小化した配送計画が立案される。また、交通事故や自然災害といった突発事象を想定したシミュレーションにより、代替ルートや輸送手段の有効性を事前に評価できる。輸送領域におけるデータサイエンスの価値は、単なる効率化だけでなく、遅延リスクを定量的に把握し、迅速な判断を可能にする点にある。

## 拠点配置領域

拠点配置は、サプライチェーン全体の構造を左右する戦略的意思決定であり、短期的な効率だけでなく、長期的なレジリエンスが求められる。将来需要予測、輸送コスト、人件費、災害リスクなどのデータを入力とし、施設配置問題を解くことで、総コストを抑えつつ供給安定性を確保する拠点配置が導かれる。

さらに、デジタルツイン上でのシミュレーションを通じて、特定拠点が機能停止した場合の影響を定量的に評価できる。これにより、安全在庫の分散配置やバックアップ拠点の必要性を合理的に判断することが可能となる。拠点配置領域では IoT センシング自体が直接の主役ではないものの、日々蓄積される在庫・出入庫データが、配置戦略を見直すための重要な基盤となる。

以上より、サプライチェーンにおけるデータサイエンスの活用は、各領域で個別に機能するのではなく、相互に連携することで全体最適を実現する点に本質的な価値があると言える。

### 3. Q3) Q2 の生成結果のどこに物足りなさや不適切さを見つけたか

Gemini の回答は、IoT・シミュレーション・最適化という技術要素を網羅的に整理しており、全体像の把握という点では有用であった一方で、以下の点に物足りなさや不適切さを感じた。

はじめに、各技術がどのように連携して意思決定に結びつくのかという因果関係の説明が弱い点である。技術の列挙に留まり、「なぜそれが重要なのか」「どの段階で価値を生むのか」がやや曖昧であった。そのため Q2 では、「現状把握、検証、意思決定」という流れを明示し、技術同士の役割分担と接続関係を補足した。

次に、効率化以外の観点（リスク対応やレジリエンス）への言及が限定的であった点である。サプライチェーンでは、災害や需要変動といった不確実性への耐性が極めて重要であるため、Q2 ではシミュレーションやデジタルツインを「リスクの定量評価手段」として位置づけ直した。

最後に、戦略レベルとオペレーションレベルの区別が不明確であった点も修正対象とした。特に拠点配置は短期的な最適化ではなく長期的な意思決定であるため、その性質を明確にし、他領域との違いを強調した。

これらの修正により、Q2 では単なる技術紹介ではなく、「なぜデータサイエンスがサプライチェーン全体にとって不可欠なのか」を論理的に説明する構成とした。

## 4. 付録

本課題で利用したコードについて、GitHub にまとめているためリンクを以下に示す。  
(<https://github.com/yut0takagi/ass-supply-chain-management2>)