

# 工場における需給連携最適化方式とメタヒューリスティック手法の適用に関する研究

メタデータ	言語: Japanese 出版者: 公開日: 2024-03-27 キーワード (Ja): キーワード (En): 作成者: 川口, 嵩平 メールアドレス: 所属:
URL	<a href="http://hdl.handle.net/10291/0002000362">http://hdl.handle.net/10291/0002000362</a>

明治大学大学院先端数理科学研究科

2023年度

博士学位請求論文

工場における需給連携最適化方式と  
メタヒューリスティック手法の適用に関する研究

A Study on Supply-Demand Coordinated Optimization Schemes  
and Application of Meta-heuristic Methods in Factories

学位請求者 ネットワークデザイン専攻

川 口 嵩 平



## 【目次】

### 第1章 序論

1. 1 研究の目的.....	1-1
1. 2 従来研究の概要と課題 .....	1-1
1. 2. 1 現状の組立加工系工場における需給計画立案業務の位置づけ .....	1-1
1. 2. 2 工場のエネルギー削減の課題 .....	1-2
1. 2. 3 近年の研究動向と課題 .....	1-5
1. 3 本研究の提案.....	1-8
1. 4 各章の概要.....	1-9
1章の参考文献.....	1-12

### 第2章 需給連携最適化方式の改良

2. 1 改良方式の概要 .....	2-1
2. 2 生産シミュレータの概要.....	2-4
2. 3 主問題の定式化 .....	2-5
2. 3. 1 生産シミュレータの各種生産パラメータと主問題との関係、及びJSPとの比較.....	2-6
2. 3. 2 決定変数 .....	2-12
2. 3. 3 目的関数 .....	2-12
2. 3. 4 制約条件 .....	2-14
2. 4 副問題の定式化 .....	2-14
2. 4. 1 決定変数 .....	2-15
2. 4. 2 目的関数 .....	2-15
2. 4. 3 制約条件 .....	2-15
2. 4. 4 副問題に対する解法アルゴリズム .....	2-19
2. 5 需給連携最適化実現に向けた並列分散需給連携シミュレータの提案 .....	2-20
2章の参考文献.....	2-23

### 第3章 改良需給連携最適化方式の主問題に対するメタヒューリスティック手法の提案

3. 1 生産シミュレータを用いた生産計画最適化問題に対する整数型PBILの適用 .....	3-1
3. 1. 1 PBILの概要 .....	3-1
3. 1. 2 整数型PBILの概要 .....	3-3
3. 1. 3 生産計画最適化問題に対する整数型PBILを用いた解法アルゴリズム .....	3-4
3. 1. 4 シミュレーション .....	3-6
3. 1. 5 整数型PBILの有効性と課題.....	3-8
3. 2 生産シミュレータを用いた生産計画最適化問題に対する整数型APBILの適用の提案 .....	3-9

3. 2. 1	整数型 APBIL の提案.....	3-9
3. 2. 2	適応関数の適用 .....	3-9
3. 2. 3	生産計画最適化問題に対する整数型 APBIL を用いた解法アルゴリズム	3-10
3. 2. 4	シミュレーション.....	3-11
3. 2. 5	整数型 APBIL の有効性と課題.....	3-15
3. 3	改場需給連携最適化方式の主問題に対する改良整数型 APBIL の適用の提案	3-16
3. 3. 1	改良整数型 APBIL の提案 .....	3-16
3. 3. 2	改良適応関数の適用 .....	3-16
3. 3. 3	主問題に対する改良整数型 APBIL を用いた解法アルゴリズム.....	3-18
3. 3. 4	シミュレーション.....	3-19
3. 3. 5	改良整数型 APBIL の有効性と課題.....	3-29
3. 4	改良需給連携最適化方式の主問題に対する改良整数型 APBIL-RTS の適用の提案	3-30
3. 4. 1	改良整数型 APBIL-RTS の提案.....	3-30
3. 4. 2	探索後半におけるローカルサーチの適用 .....	3-30
3. 4. 3	主問題に対する改良整数型 APBIL-RTS を用いた解法アルゴリズム ..	3-31
3. 4. 4	シミュレーション.....	3-32
3. 5	3章の結論.....	3-39
	3章の参考文献.....	3-41
<b>第4章 需給連携最適化方式の主問題に対するメタヒューリスティック手法の提案</b>		
4. 1	需給連携方式における従来研究の課題と提案手法の概要 .....	4-1
4. 1. 1	需給連携方式における従来研究の課題.....	4-1
4. 1. 2	提案手法の概要 .....	4-3
4. 2	主問題の定式化 .....	4-3
4. 2. 1	決定変数 .....	4-3
4. 2. 2	目的関数 .....	4-3
4. 2. 3	制約条件 .....	4-4
4. 2. 4	生産計画の表現方法.....	4-6
4. 3	主問題に対する改良並列 RHPSO の適用の提案.....	4-6
4. 3. 1	RHPSO.....	4-6
4. 3. 2	改良型 RHPSO の提案.....	4-7
4. 3. 3	改良並列 RHPSO の提案 .....	4-10
4. 4	副問題に対する解法アルゴリズム.....	4-10
4. 5	シミュレーション.....	4-10
4. 5. 1	シミュレーション条件 .....	4-11

4. 5. 2 シミュレーション結果 .....	4-12
4. 6 4章の結論.....	4-18
4章の参考文献.....	4-19
5. 結論 .....	5-1
謝辞	
研究業績	

## 1. 序論

### 1. 1 研究の目的

組立加工系工場は、大きく分けて製品を製造する生産設備と、生産設備にエネルギーを供給するエネルギー供給設備が存在する。工場全体のエネルギー購入コストや労務費を最小化するためには、生産設備と供給設備の各計画である生産計画とエネルギー設備運用計画の両方を同時に最適化する必要がある。

従来研究では、生産計画最適化問題の一種である Job-shop Scheduling Problem(以下, JSP)を主問題, エネルギー設備最適運用計画問題を副問題と定義し, 相互に連携することで両方の計画を同時に最適化する需給連携最適化方式を提案し, 検証により本方式の有効性を確認した<sup>(1-1)</sup>。しかし, 従来研究は, 主問題を JSP として定義しているが, JSP は, 実際の組立加工系工場の生産現場が有する制約条件の一部のみを考慮し, 理想化されたモデルに基づいた生産計画最適化問題であるため, 生産現場の条件が合えば適用できるが, 汎用性が低いという問題があった。また, 従来研究の主問題である JSP を解くための適用手法に解の質の面で改善の余地があった<sup>(1-1)</sup>。

本論文では, 様々な生産現場に汎用的に適用することを目的として, 従来研究の需給連携最適化方式の改良を行う。具体的には, 実際の組立加工系工場の生産現場の制約条件を全て考慮するため, 従来研究の需給連携最適化方式の主問題に対して, 生産現場で実際に利用している生産計画ツールをベースとした生産シミュレータを適用する。また, 改良需給連携最適化方式を実現することと処理時間を削減することを目的として, 改良需給連携最適化方式の主問題に用いる生産シミュレータの改良開発を行う。そして, 改良方式の主問題の解の質向上と, 主問題に適用するメタヒューリスティック手法のパラメータ調整工数削減を目的として, 改良方式の主問題に対し適用するメタヒューリスティック手法を提案する。更に, 従来研究の需給連携最適化方式の主問題の解の質向上を目的として, 従来研究の方式の主問題に対し適用するメタヒューリスティック手法を提案する。

### 1. 2 本研究の課題と概要

#### 1. 2. 1 現状の組立加工系工場における需給計画立案業務の位置づけ

これまで, 日本の組立加工系工場では, 顧客納期を守りながら可能な限り多くの製品を製造すること, つまり生産性を第一優先として製造を行っていた。図 1.1 に, 現状の組立加工系工場における業務フロー図を示す。まず初めに, 生産現場を統括管理する生産管理者が, 顧客からの製造依頼に基づいて, 製造すべき製品種や製品数, 製造納期といった製

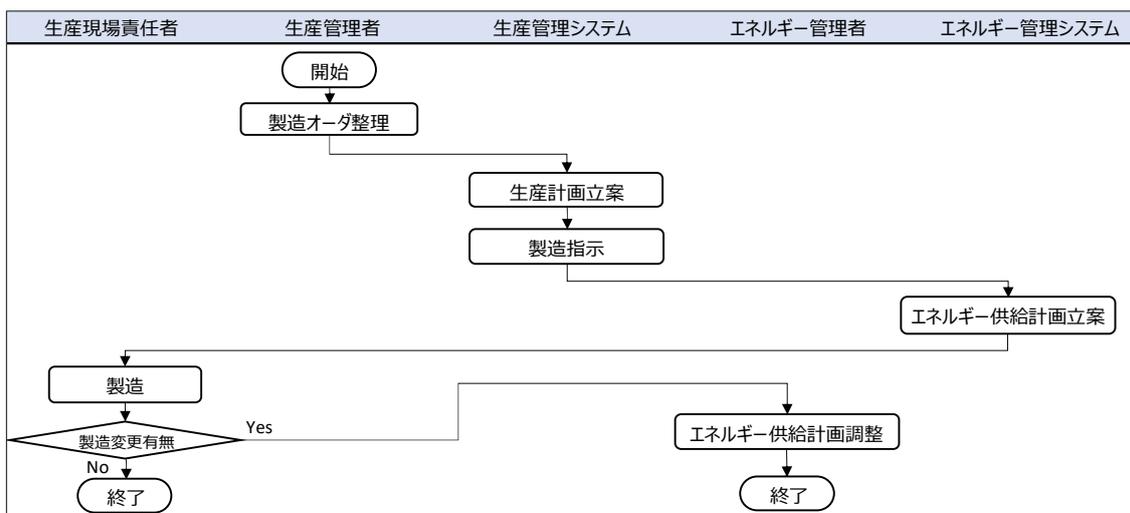


図 1.1 現状の組立加工系工場における業務フロー図

造オーダーを整理し生産管理システムにオーダー情報等の計画情報を入力する。次に、生産管理システムが、入力された製造オーダーを基に顧客納期を順守した生産計画を立案する。生産計画が立案されると、生産管理システムは、製造指示を出力する。製造指示を受けた生産現場責任者は、生産計画に基づき製造を行う。また、エネルギー管理システムが、生産計画情報を基に生産設備が必要とするエネルギー需要を満たすエネルギー供給が可能なエネルギー供給設備運用計画を立案する。そして、エネルギー供給設備を統括管理するエネルギー管理者が、適宜計画を調整する。つまり、現状の組立加工系工場における需給計画立案業務は、最初に需要側の生産計画を立案し、その後に立案された生産計画に基づいて供給側のエネルギー供給設備運用計画を立案するという順序で計画立案が行われている。

## 1. 2. 2 工場のエネルギー削減の課題

2015 年に国連サミットにて提唱された持続可能な開発目標 Sustainable Development Goals(SDGs)の特に目標 7 と目標 13 に関連し、カーボンニュートラルの実現に向けた取り組みが世界的に推進されている<sup>(1-2)</sup>。このような状況下で、先進的企業は、具体的な数値目標を掲げて CO<sub>2</sub>削減に取り組んでいる<sup>(1-3,1-4)</sup>。カーボンニュートラルの目標達成には、CO<sub>2</sub>発生原因の主要素である電力消費量削減が必須である。日本国内企業の業種別電力消費量を含むエネルギー消費割合の推移を図 1.2 に、全業種のうち産業部門(製造業、農林水産業、鉱業、建設業)と業務他部門に限定したエネルギー消費量割合の推移<sup>(1-5)</sup>を図 1.3 に示す。図 1.2 より、2020 年度の全業種における産業部門と業務他部門の合計割合は、61.9%である。また、図 1.3 より、産業部門と業務他部門において、製造業の占める割合は、68.1%に達し最大である。つまり、全業種における製造業のエネルギー消費割合は、61.9%×68.1%で計算でき、約 42.2%に達していることから、製造業における省エネルギーの取り組みが、カーボンニュートラルの実現に向けて重要であるといえる。また、工場の生産現場の運営において、

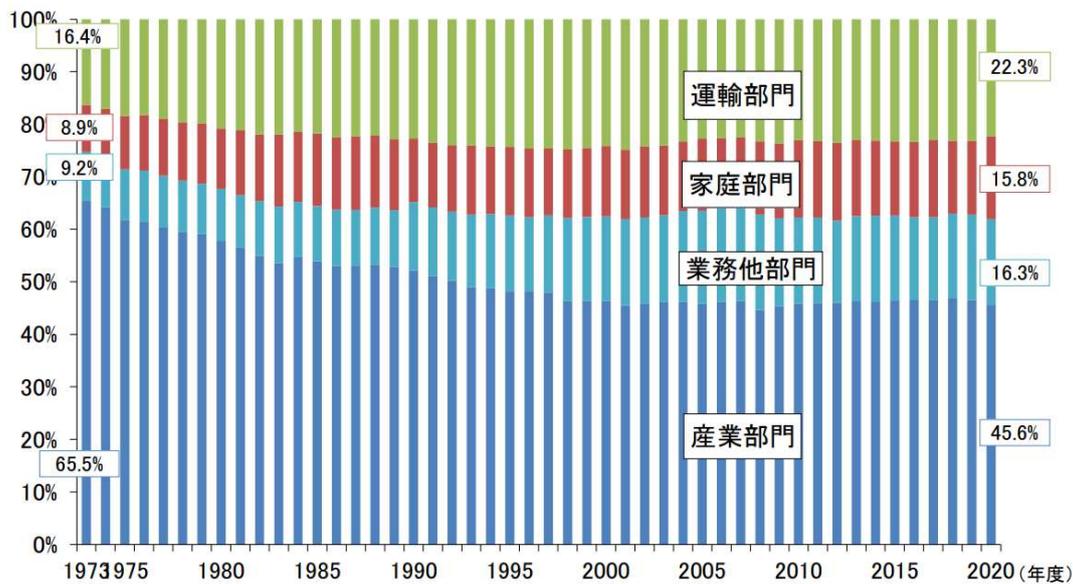


図 1.2 日本の全業種におけるエネルギー消費割合の推移<sup>(1-5)</sup>

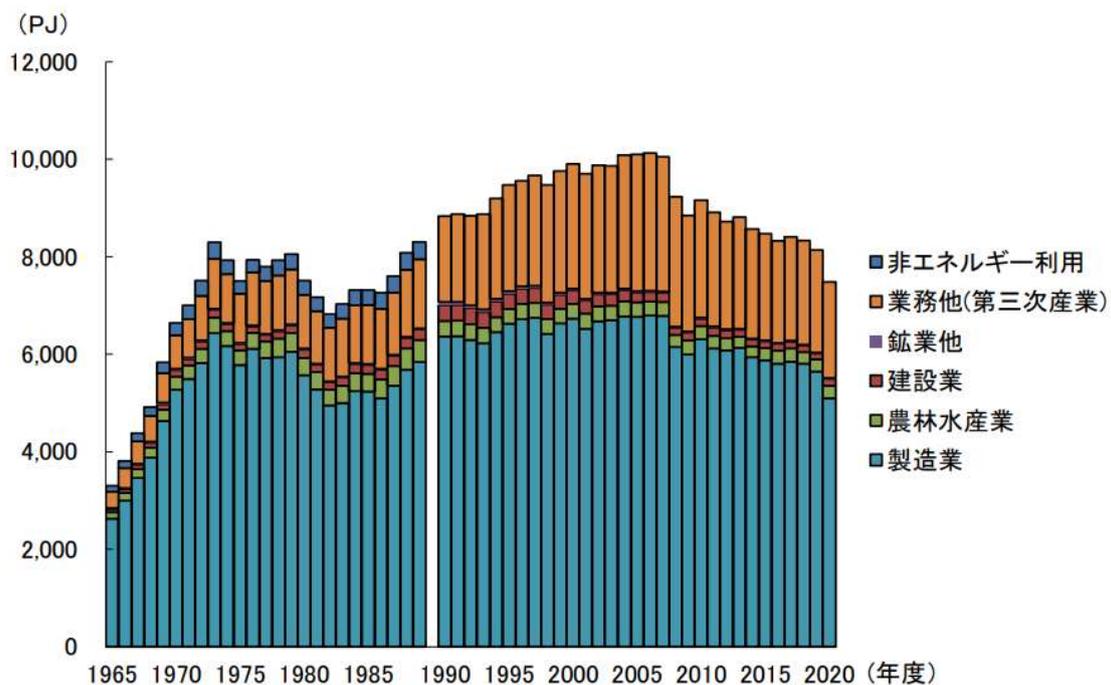


図 1.3 日本の産業部門と業務他部門における業種別エネルギー消費割合の推移<sup>(1-5)</sup>

経営的観点から見ると作業者の労務費やエネルギー購入コストを抑えることは重要である。つまり、労務費やエネルギー購入コスト、CO<sub>2</sub>排出量をコスト換算したCO<sub>2</sub>排出コストを下げるのが重要である。

製造業では、Industry4.0やSociety5.0の実現に向けた技術革新が進められている<sup>(1-6,1-7)</sup>。この状況下で、組立加工系工場の需要側である生産設備では、従来から取り組まれてきた顧

客納期遵守を最優先とする生産時間最小化を目的関数とした生産計画最適化の研究<sup>(1-8)</sup>だけでなく、生産設備の電力消費量や CO<sub>2</sub> 排出量の最小化を目的関数に追加した生産計画最適化の研究が行われてきた<sup>(1-9-1-17)</sup>。しかし、これらの研究は、生産設備のみを対象範囲として生産計画が解かれており、工場のエネルギー供給設備を対象としていなかった。

図 1.4 に工場におけるエネルギー供給会社から生産設備までのエネルギー遷移とエネルギー購入コスト決定順序を示す<sup>(1-18)</sup>。中・大規模工場の一部は、電力や天然ガスの供給会社から購入した電気や天然ガス等(以下、二次エネルギー)を使って、生産設備に必要な電気、熱、蒸気、圧縮空気等(以下、最終消費エネルギーを効率的に供給することを目的として、発電機や冷凍機、蓄熱槽等で構成されたエネルギープラントを有する。エネルギープラントでは、二次エネルギーを入力として、各種設備によって生産設備が要する最終消費エネルギーに変換し、生産設備に供給する(図 1.4 上段参照)。エネルギー購入コスト決定の点に着目すると、工場では、決定した生産計画を実現するために必要な最終消費エネルギー消費量を 1 時間毎に求め、この 1 時間毎の最終消費エネルギー消費量を基にエネルギープラント運用を行う。エネルギープラントの各種設備の運用が決まることで、エネルギープラントが必要な 1 時間毎の二次エネルギー消費量が決まる。そして、二次エネルギー消費量を基に二次エネルギー購入コストを決定する(図 1.4 下段参照)。また、二次エネルギー購入コストが決まることで、CO<sub>2</sub> 排出コストが決定する。つまり、生産計画の変更により、エネルギープラントが対象とする 1 時間毎の最終消費エネルギー消費量は変化する。しかし、エネルギープラント最適運用の従来研究では、供給側のエネルギープラントのみを対象とし、生産設備の最終消費エネルギー消費量は固定値として問題が解かれていた<sup>(1-19-1-24)</sup>。

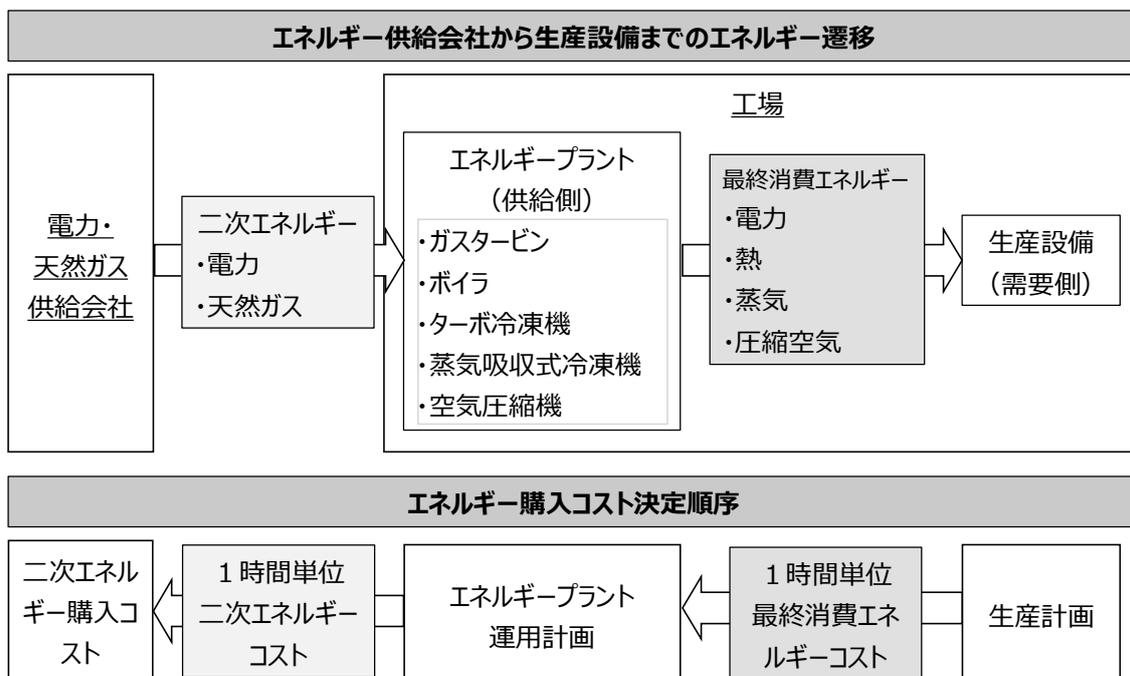


図 1.4 工場におけるエネルギー供給会社から生産設備までのエネルギー遷移とエネルギー購入コスト決定順序<sup>(1-18)</sup>

### 1. 2. 3 近年の研究動向と課題

1.2.2 節の説明より、工場の需要側と供給側を対象とした研究は、それぞれ需要側と供給側を別問題とされていたことが分かる。しかし、1.2.2 節で示した通り、工場の供給設備と需要設備は密に連携し、二次エネルギーコストやそれに伴う CO<sub>2</sub> 排出コストが決まるという観点では、工場の需要側と供給側全体を対象として最適化すべきである。そこで、電子情報技術産業協会(JEITA)の制御・エネルギー管理専門委員会 WG1(省エネルギー)は、工場の需要側と供給側の不調和による無駄を減らすため、需要側と供給側の各設備を連携し、需給両方を最適に制御する連携制御という概念を提唱した<sup>(1-25)</sup>。連携制御には様々な形態があるが、その中で二次エネルギーの削減に最も寄与する連携方式として、需給双方向連携がある(図 1.5 参照)。需給双方向連携は、生産設備の需要量に適合するためにエネルギープラントの運用を最適化すると同時に、生産設備の運転も調整することで、需要側の生産設備と供給側のエネルギープラントの最適化を図る連携制御である。近年、著者らの従来研究を含め、組立加工系工場の需要側の生産計画最適化問題と供給側のエネルギープラント最適運用計画問題の両方を対象として需給双方向連携を実現する研究はいくつか行われている<sup>(1-18,1-26)</sup>。更に、著者らは、従来研究として、生産計画最適化問題の一種である Job-shop Scheduling Problems(以下、JSP)を主問題、エネルギープラント最適運用計画問題を副問題として定義した需給連携最適化方式を提案した<sup>(1-18)</sup>。

図 1.6 に、従来研究として文献(1-18)で提案した、需給連携最適化方式を適用した場合の組立加工系工場の業務フロー図を示す。図 1.1 で示す現状の組立加工系工場の業務フロー図に対して、生産計画最適化エンジンと JSP 指標算出システムが追加されている。また、需給連携最適化方式は主問題において、JSP の生産計画立案を行うため、図 1.1 の生産管理システムは、図 1.6 では JSP 計画立案システムに変更している。同様に、副問題において、エネルギープラントを対象として最適運用を行うため、図 1.1 のエネルギー管理システムは、図 1.6 ではエネルギープラント最適運用シミュレータに変更している。図 1.1 は、最初

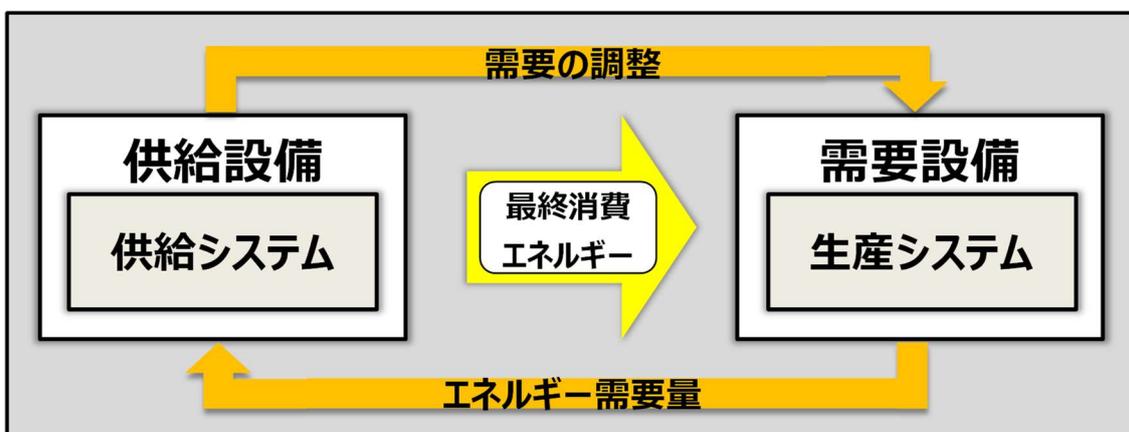


図 1.5 需給双方向連携概要図

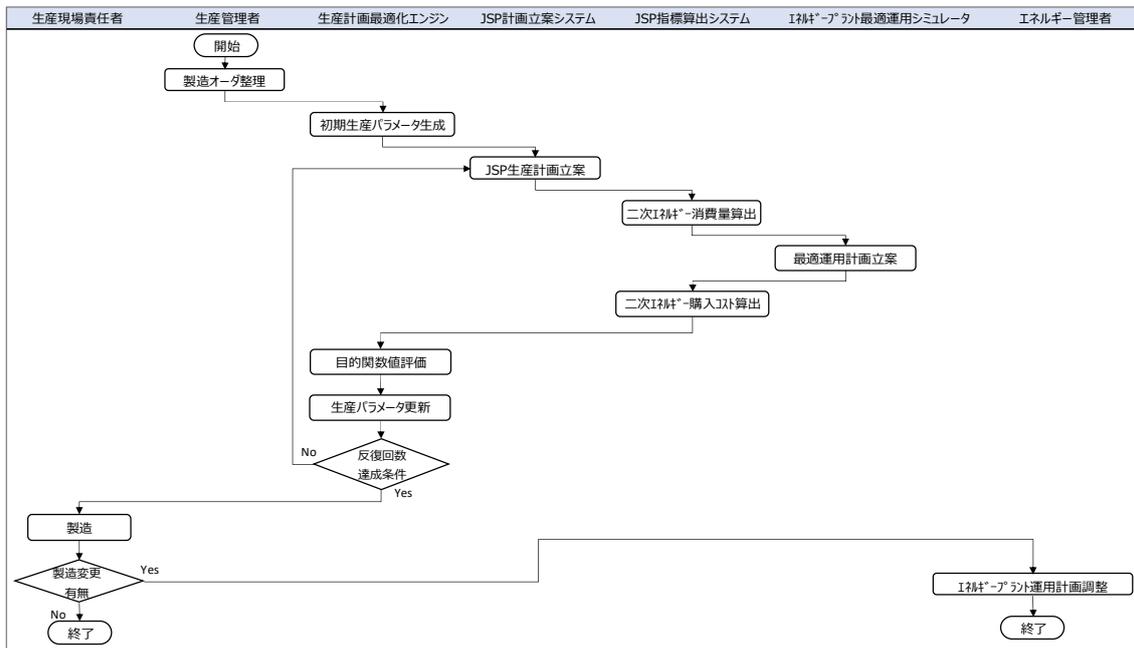


図 1.6 需給連携最適化方式を適用した場合の組立加工系工場における業務フロー図

に生産計画が立案され、立案された生産計画の二次エネルギー消費量を基にエネルギー供給計画を立案するという単方向のフローであった。一方、図 1.6 は、生産計画とエネルギープラント最適運用計画を立案し、2つの計画を基に算出した目的関数値を基に、再度2つの計画を更新する。つまり、需給連携最適化方式は、生産計画とエネルギープラント運用計画の両方を同時に評価し、相互連携しながら最適化するフローであることが特徴である。これにより、従来実現できなかった工場全体の最適化が実現できる。

しかし、これらの従来研究には2点の問題があった。1つ目の問題は、従来研究で提案した需給連携最適化方式は、様々な生産現場に適用できないという点である。従来研究は、主問題の生産計画最適化問題を JSP や Flow-shop Scheduling Problems(以下、FSP)として定式化し、ベンチマーク問題により解の質を比較している。JSP や FSP は、制約条件として、ジョブの工程処理順序順守制約や機械によるジョブの同時処理禁止制約、各種時間制約が存在する。一方、実際の組立加工系工場の生産現場は、JSP や FSP で定義する制約以外に、機械毎・製品毎の生産可否制約や設備能力制約、労働者の労働時間制約等、様々な制約が存在する。また、JSP や FSP は、組立加工系工場の理想化されたモデルに基づいた生産計画最適化問題である。つまり、従来研究の主問題で扱う JSP や FSP は、組立加工系工場の生産現場の一部で限定的に適用できる問題であり、全ての組立加工系工場への適用はできないといえる。2つ目の問題は、従来研究で適用したメタヒューリスティック手法では解の質に改善の余地があるという点である。従来研究で扱う需給連携最適化方式は、工場の需要側と供給側の計画最適化問題を両方対象として、相互に連携しながら工場全体の最適化を図る方式として、著者らにより世界で初めて提案した。つまり、従来研究で扱う需給連携最

適化方式は、適用範囲が限定的であるが、工場のエネルギー需給全体を考慮して工場の二次エネルギーコストの削減や CO<sub>2</sub> 排出量の削減を検討できるという点で意義があるといえる。著者らの従来研究では、JSP を解くために開発された Hybrid Particle Swarm Optimization (以下, HPSO)<sup>(1-27)</sup>の改良手法である並列 Reactive HPSO (以下, 並列 RHPSO) を提案し、これまでの著者らの従来研究の提案手法と比較し、解の質の向上を確認した<sup>(1-1)</sup>。しかし、従来研究の並列 RHPSO は、近傍生産計画生成方法として、交換可能な隣同士のジョブを交換する方法のみを用いていたため、クリティカルブロックを用いた近傍生産計画生成に多様性がなかった。

1 点目の問題を解決するため、著者らは、組立加工系工場の需要側のみを対象として生産現場の様々な制約を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化することにより精緻に再現可能な生産シミュレータ<sup>(1-28)</sup>を開発した。著者らの従来研究では、この生産シミュレータ上で生産現場をデジタルツインモデルとして表現し、ライン間の生産数量割合を決定変数とした生産計画最適化問題を対象問題として、進化計算手法を適用し有効性を確認した<sup>(1-29)</sup>。従来研究である文献<sup>(1-29)</sup>を通して、生産シミュレータの 1 回当たりの処理時間が掛かることを確認した。一方、現場適用を考えると、特急オーダーの調整や計画変更等の外乱を考慮し、1 日で複数回最適生産計画を立案する必要がある。つまり、生産シミュレータを用いた生産最適化問題は、生産シミュレータの実行回数が少ない中で質の高い解を導出する必要がある。更に、生産シミュレータの処理時間を削減する必要がある。しかし、それらを考慮したメタヒューリスティック手法の適用や生産シミュレータの実現はできていなかった。

以上より、従来研究が解決すべき課題は、以下の 4 点である。

【課題 1】 需給連携最適化方式の汎用性向上

【課題 2】 需給連携最適化方式の主問題である JSP に対するメタヒューリスティック手法の適用による解の質の向上

【課題 3】 生産シミュレータの処理時間削減

【課題 4】 生産シミュレータの実行回数が少ない条件下の生産計画最適化問題に対するメタヒューリスティック手法の適用による解の質の向上

### 1. 3 本研究の提案

本研究では、1.2.3 節で説明した 4 点の課題を解決するため、以下の提案を行う。図 1.7 に、従来研究の課題と本研究の全体像を示す。1 つ目の提案は、需給連携最適化方式の主問題を、組立加工系工場の生産現場の様々な制約を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化することにより精緻に再現可能な生産シミュレータを用いた生産計画最適化問題に改良した、改良需給連携最適化方式の提案である（図 1.7 中、提案 1）。2 つ目の提案は、生産シミュレータをベースとして、並列分散機能を持ち、且つ需給連携最適化を実現する並列分散需給連携シミュレータの提案である（図 1.7 中、提案 2）。3 つ目の提案は、改良需給連携最適化方式の主問題に対する整数型 Adaptive Population-Based Incremental Learning（以下、整数型 APBIL）、改良整数型 Adaptive Population-Based Incremental Learning（以下、改良整数型 APBIL）、及び改良整数型 Adaptive Population-Based Incremental Learning-Reactive Tabu Search（以下、改良整数型 APBIL-RTS）の提案である（図 1.7 中、提案 3-1、3-2、3-3）。そして、4 つ目の提案は、需給連携最適化方式の主問題に対する改良並列 Reactive Hybrid Particle Swarm Optimization（以下、改良並列 RHPSO）の適用の提案である（図 1.7 中、提案 4）。

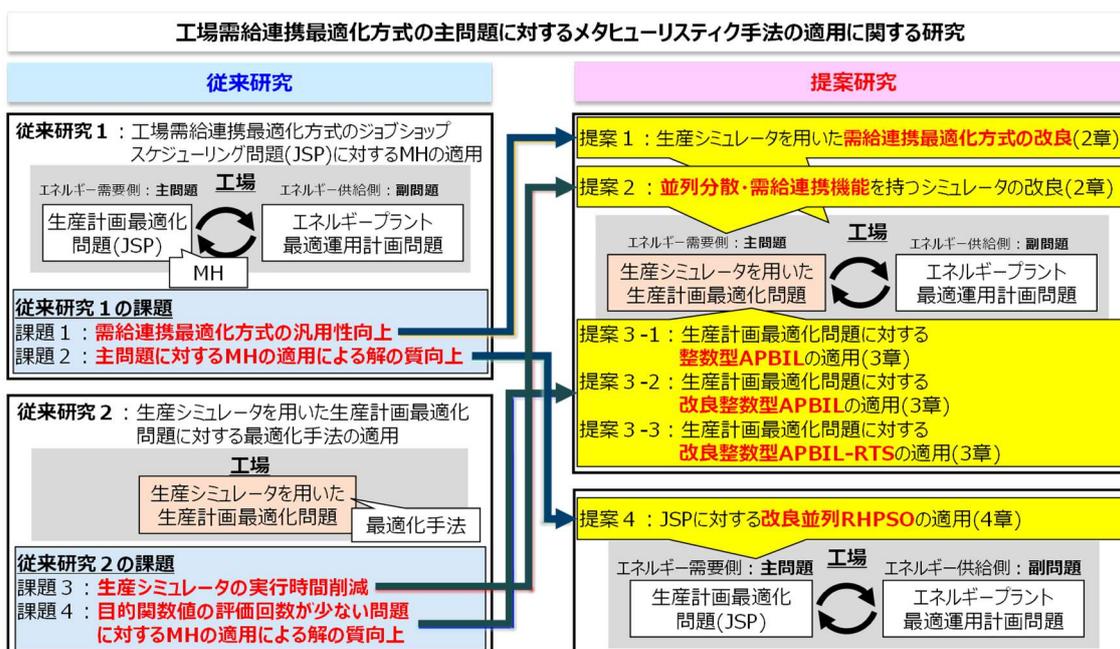


図 1.7 従来研究の課題と本研究の全体像

## 1. 4 各章の概要

以下、章ごとに本論文の概要を示す。

### 第2章の概要

第2章では、需給連携最適化方式の主問題を、組立加工系工場の生産現場の様々な制約を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化することにより精緻に再現可能な生産シミュレータを用いた生産計画最適化問題に改良した改良需給連携最適化方式の提案を行う。

従来研究の需給連携最適化方式の主問題は、JSPを用いているため、実際の組立加工系工場の生産現場で扱う制約条件の一部のみを考慮し、理想化されたモデルに基づいて最適化していた。一方で、実際の組立加工系工場にて生産計画を立案する場合、生産計画ツールを用いて対象の生産現場をモデル化する必要がある。生産現場のモデルは、対象の工場によって適切な粒度（建屋⇒生産ライン⇒工程⇒設備・作業員）で構築する必要があり、設備や工程や生産ラインや建屋を工場毎に適切な粒度でモデル構築するのが一般的である。更に、生産計画ツールを用いて構築された生産現場のモデルを用いて、生産計画最適化シミュレーションを行うためには、目的関数、制約条件、及び決定変数を定式化し、生産現場の全ての制約条件を考慮する必要がある。目的関数値に関しては、生産計画ツールにて設定した、製品種、製造数量、製造納期等の生産パラメータによって求めた生産計画を基に、工場毎に設定した評価指標を利用して計算することで求める。この際、生産計画ツールのみでの評価指標出力が不十分な場合、別途生産計画ツールの出力情報から評価指標を計算する補助的なシミュレータを考慮する必要がある。制約条件に関しては、生産計画ツールで内包的に表現されており、これを数式化することは困難である。つまり、生産計画ツールをブラックボックスとして定義する。決定変数に関しては、生産計画ツールで設定する生産パラメータのうち、顧客からの製造オーダーを満たしつつ生産計画の評価指標の改善に寄与するパラメータを決定変数として設定することで、一意の生産計画が得られるブラックボックス最適化問題と考える方がより自然である。上記背景より、従来研究の主問題として定式化していたJSPは、実際の組立加工系工場の生産現場で扱う制約条件の一部のみを考慮し、理想化されたモデルに基づいて最適化しているため、実際の組立加工系工場の生産現場の一部に適用可能である一方、汎用性が低いといえる。この課題（課題1）を解決するため、需給連携最適化方式の主問題を、著者らが開発した生産計画立案機能と評価指標算出機能を有する生産シミュレータ<sup>(1-28)</sup>を用いてブラックボックス最適化問題として定式化した生産計画最適化問題に変更した改良需給連携最適化方式を提案する。

また、著者らの従来研究により、生産現場の様々な制約を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化することにより精緻に再現可能な生産シミュレータ<sup>(1-28)</sup>を用いた生産計画最適化問題に対して最適化

手法を適用し、有効性を確認した<sup>(1-29)</sup>。しかし、1.2.3節で述べた通り、文献(1-29)の検証により、課題3があることを確認した。そこで、課題3を解決するため、改良需給連携最適化と処理時間削減を達成する並列分散需給連携シミュレータを提案する。

### 第3章の概要

第3章では、改良需給連携最適化方式の主問題に対する整数型 APBIL（提案手法 3-1）、改良整数型 APBIL（提案手法 3-2）、及び改良整数型 APBIL-RTS（提案手法 3-3）の提案を行う。

第3章で提案する改良需給連携最適化方式の主問題として新たに定義した、生産シミュレータ上で表現する生産計画最適化問題について、決定変数の選定と選定した決定変数を用いた生産計画最適化に関する研究を著者らの研究も含め実施しており、ライン間の生産割合と、ライン毎の製品の生産順序と、製品の生産開始時刻の3種類の生産パラメータが、決定変数として生産計画最適化を行うことで生産計画の改善に寄与することを確認している<sup>(1-29-1-31)</sup>。この中で、ライン毎の製品の生産順序という決定変数は、従来研究の需給連携最適化方式の主問題として扱った JSP の決定変数と同じ生産順序を扱うものである。更に、製品機種が異なる際発生する治具交換等の段取り替えに掛かる時間（以下、段取り替え時間）は生産順序によって変わるため、改良需給連携最適化方式の主問題においても、生産順序の最適化が適切に行われているかを確認することは重要である。そこで、3章では、主問題の生産計画最適化問題で扱う決定変数を、ライン毎の製品の生産順序とする。また、ライン毎の製品の生産順序は離散値として扱うことができるため、組合せ最適化問題として定義する。生産シミュレータを用いたこれまでの従来研究により、生産シミュレータを用いた生産計画最適化問題は、最適生産計画立案に要する時間が限られ、且つ1回当たりの生産シミュレータの処理時間が数十秒掛かることから、生産シミュレータの実行回数が少ない条件下でより質の高い解を導出する必要があった（課題4）。

3章では、上述の課題4を解決するため、改良需給連携最適化方式の主問題に対して、以下のような特徴を持つ3つのメタヒューリスティック手法を提案する。

- ・解候補の確率モデルを用いた分布推定アルゴリズムをベースとし、ブラックボックス最適化問題に対して、効率的な探索が可能である手法（全手法（提案手法 3-1, 3-2, 3-3））。
- ・学習率を探索に応じて適応的に変更するアルゴリズムの開発により、様々な工場にパラメータチューニングせずに適用可能である手法（整数型 APBIL（提案手法 3-1））。
- ・初期学習率の設定によらず、探索に応じて学習率を適切且つ適応的に変更するアルゴリズムの開発により、様々な工場にパラメータチューニングせずに適用可能である手法（改良整数型 APBIL（提案手法 3-2））。
- ・探索の前半に探索の多様化を強め、探索の後半に探索の集中化を強めることで、目的関数値の評価回数が少ない問題に対して効率的に探索可能である手法（改良整数型 APBIL-RTS（提案手法 3-3））。

検証では、実際の組立加工系工場の研磨加工工程をモデルとした生産計画最適化問題に適用し、従来手法と解の質の比較検証を行うことにより有効性を示す。更に、生産計画最適化問題を表現している生産シミュレータに並列分散機能を付加し、従来のシミュレータとの全処理時間の比較検証を行うことにより有効性を示す。

#### 第4章の概要

第4章では、需給連携最適化方式の主問題に対する改良並列RHPSOの適用の提案を行う。

従来研究で適用したメタヒューリスティック手法である並列RHPSOの近傍生産計画生成法として用いていた方法は、クリティカルブロック内の隣同士の工程のジョブの処理順序を入れ替えるAdjacent Swapping近傍（以下、AS近傍）と呼ばれる手法である<sup>(1-32)</sup>。そのため、近傍生産計画の数が限られ、多様な近傍生産計画を生成できず効率的な探索が出来ていない可能性があった。上記考察により定めた課題2を解決するため、提案手法である改良型並列RHPSOでは、RTSに適用する効果的な方法を検討した結果、以下の二つの手法を適用する。一つは、クリティカルブロック内の工程をクリティカルブロック内の工程の先頭または最後尾に移動するCritical Block近傍（以下、CB近傍）という手法<sup>(1-33)</sup>である。CB近傍は、RTSの適用で有効性が確認できた。もう一つは、クリティカルブロック内の先頭と最後尾にある工程を入れ替える手法<sup>(1-34)</sup>に加えて、工程の集合を入れ替えるTop End近傍（以下、TE近傍）である。これらのCB近傍及びTE近傍は、AS近傍に比べダイナミックに工程を変化させることができる特徴を持つ。先行研究では、これら3種類の近傍解生成法のうちそれぞれ1種類のみ適用していたが、これらを組み合わせた手法は提案されていなかった。

4章では、AS近傍にTE近傍を加えた近傍生産計画生成法（提案手法4-1）と、AS近傍にCB近傍を加えた近傍生産計画生成法（提案手法4-2）と、AS近傍にCB近傍とTE近傍両方を加えた近傍生産計画生成法（提案手法4-3）を提案する。これらの提案手法を10機械10ジョブのベンチマーク問題（abz6<sup>(1-35)</sup>）に適用し、従来手法と解の質の比較検証を行うことにより提案手法の有効性を示す。

#### 第5章の概要

5章は結論であり、本論文の成果を総括する。

以上、本論文では、従来研究で示した工場の需給連携最適化方式に対し、実現性向上を実現するために改良した需給連携最適化方式の提案について述べる。また、両方の方式に対するメタヒューリスティック手法の改良を行い、解の質の面と処理時間の面での有効性について述べる。

## 第 1 章の参考文献

- (1-1) S. Kawaguchi and Y. Fukuyama, "Parallel Hybrid Particle Swarm Optimization for Integration Framework of Optimal Operational Planning Problem of an Energy Plant and Production Scheduling Problem," IEEE The 1st International Conference on Artificial Intelligence in Information and Communication (2019).
- (1-2) Carbon Neutrality by United Nations, <https://sdgs.un.org/partnerships/carbon-neutrality>.
- (1-3) 例えば, Google Co. Commitment, <https://sustainability.google/commitments/carbon/#>.
- (1-4) 例えば, SoftBank Co. Commitment, <https://www.softbank.jp/en/corp/sustainability/special/carbon-neutral2030/>.
- (1-5) 経済産業省資源エネルギー庁:「令和 3 年度エネルギーに関する年次報告(エネルギー白書 2022)」, 第 2 部 第 1 章, <https://www.enecho.meti.go.jp/about/whitepaper/2022/pdf/> (2022).
- (1-6) T. Stock and G. Seliger, "Opportunities of Sustainable Manufacturing in Industry 4.0," Journal of Procedia CIRP, Vol. 40, pp.536-541.
- (1-7) 内閣府: Society 5.0, [https://www8.cao.go.jp/cstp/society5\\_0/](https://www8.cao.go.jp/cstp/society5_0/).
- (1-8) D. Applegate and W. Cook, "A computational study of the job-shop scheduling problem," ORSA Journal on Computing, Vol. 3, No.2, pp. 85-176 (1991).
- (1-9) K. Fang, N. Uhan, F.Zhao, and J. Sutherland, "A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction," Journal of Manufacturing Systems Vol. 30, Issue 4, pp. 234-240 (2011).
- (1-10)S. Kemmoe and N. Tchernev, "A GRASPxEELS for Scheduling of Job Shop Like Manufacturing Systems and CO2 Emission Reduction," IFIP Advances in Information and Communication Technology, Vol. 439. pp 130-137 (2014).
- (1-11)Y. Liu, H. Dong, N. Lohse, S. Petrovic, and N. Gindy, "An investigation into minimizing total energy consumption and total weighted tardiness in job shops," Journal of Cleaner Production, Vol. 65, pp. 87-96 (2014).
- (1-12)G. May, B. Stahl, M. Taisch, and V. Prabhu, "Multi-objective genetic algorithm for energy-efficient job shop scheduling," International Journal of Production Research, Vol. 53, No. 23, pp. 7071-7089 (2015).
- (1-13)C. Zang, P. Gu, and P. Jiang, "Low-carbon scheduling and estimating for a flexible job shop based on carbon footprint and carbon efficiency of multi-job processing," Proceedings of the Institution of Mechanical Engineers, Journal of Engineering Manufacture Vol. 229. Issue, 2, pp. 328-342 (2015).

- (1-14)J. Y. Ding, S. Song, R. Zhang, R. Chiong, and C. Wu, "Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches," *IEEE Transactions on Automation Science and Engineering*, Vol. 13, No. 2, pp.1138–1154 (2016)
- (1-15)X. Liao, R. Zhang, and R. Chiong, "Multi-objective optimization of single machine scheduling with energy consumption constraints," *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp.2922-2929 (2017)
- (1-16)T. Jiang and G. Deng, "Optimizing the Low-Carbon Flexible Job Shop Scheduling Problem Considering Energy Consumption," *IEEE Access*, Vol. 6. pp. 46346-46355 (2018)
- (1-17)X Yang, J Zhang, N Zhang, and Y Li, "Low Carbon Multi-Objective Shop Scheduling Based on Genetic and Variable Neighborhood Algorithm," *First International Conference on Computer Applied Science and Information Technology* (2020).
- (1-18)川口嵩平・福山良和：「改良近傍スケジューリング生成法を用いた改良型並列 Reactive Hybrid Particle Swarm Optimization による最適生産計画とエネルギープラント最適運用計画の統合最適化」, *電学論 D*, Vol. 139, No. 12, pp.1005-1014 (2019).
- (1-19)H. Ravn and J. M. Rygaard, "Optimal scheduling of coproduction with a storage," *Journal of engineering optimization*, Vol. 22, pp. 267-281 (1994).
- (1-20)K. Ito, T. Shiba, and R. Yokoyama, "Optimal Operation of a Cogeneration Plant in Combination with Electric Heat Pumps," *Journal of Energy Resources Technology*, Vol. 116, No. 1, pp. 56-64 (1994).
- (1-21)R. Suzuki, F. Kawai, S. Kitagawa, T. Matsui, K. Matsumoto, D. Xiang, and Y. Fukuyama, "The  $\epsilon$  constrained differential evolution approach for optimal operational planning of energy plants," *Proc. of IEEE Congress on Evolutionary Computation*, pp. 1-6 (2010).
- (1-22)N. Nishimura, Y. Fukuyama, and T. Matsui, "Dependable Parallel Differential Evolutionary Particle Swarm Optimization for On-line Optimal Operational Planning of Energy Plants," *IEEEJ Trans. on EIS*, Vol. 137, No. 11 (2017) (in Japanese).
- (1-23)西村典大・福山良和・松井哲郎：「エネルギープラントのオンライン最適運用計画に対するディペンダブルな並列 DEEPSO による方式の提案」, *電学論 C*, Vol. 137, No. 11 (2017).
- (1-24)K. Arai, Y. Fukuyama, T. Iizaka, and T. Matsui, "Optimal Operational Planning of Energy Plants Considering Uncertainty of Renewable Energy Outputs by Global-

- best Brain Storm Optimization," *IFAC-PapersOnLine*, Vol. 52, Issue 4, pp. 153-158 (2019).
- (1-25)電子情報技術産業協会(JEITA) : 「連携制御ガイドブック」 ,  
[https://home.jeita.or.jp/upload\\_file/20170329160814\\_Wv0SQHuAn1.pdf](https://home.jeita.or.jp/upload_file/20170329160814_Wv0SQHuAn1.pdf) (2017).
- (1-26)藪内雅幸・貝原俊也・藤井信忠・國領大介 : 「サステナブル生産の実現を目指す生産計画と工場エネルギー供給計画の連携手法に関する研究」, 日本機械学会生産システム部門研究発表講演会(2022).
- (1-27)D. Y. Sha, and C. Hsu, "A hybrid particle swarm optimization for job shop scheduling problem," *Journal of Computers & Industrial Engineering*, Vol. 51, Issue 4, pp. 791-808 (2006).
- (1-28)三菱電機株式会社ニュースリリース : 「「環境配慮型生産コストシミュレーション技術」を開発」 , [1023.pdf \(mitsubishielectric.co.jp\)](#) (2019).
- (1-29)K. Takahashi, Y. Fukuyama, S. Kawaguchi, and T. Sato, "Optimal Production Scheduling using a Production Simulator and Multi-population Global-best Modified Brain Storm Optimization," Proc. of IEEE World Congress on Computational Intelligence (2022).
- (1-30)川口嵩平・福山良和・佐藤隆臣 : 「実用的な生産シミュレータと整数型 Adaptive Population Based Incremental Learning を用いた最適生産順序導出手法の適用」, 電気学会システム/スマートファシリティ合同研究会(2022).
- (1-31)R. Katagiri, Y. Fukuyama, S. Kawaguchi, K. Takahashi, and T. Sato, "Optimal Production Scheduling by Integer Form of Population-Based Incremental Learning with Initial Probability Matrix Setting Methods and a Practical Production Simulator", Proc. of IEEE Symposium Series on Computational Intelligence (2023).
- (1-32)E. Balas, "Machine Sequencing via Disjunctive Graphs: An Implicit Enumeration Algorithm," *Operations Research*, Vol. 17, No. 6, pp. 941-957 (1969).
- (1-33)山田武士・中野良平 : 「確率的探索と確定的探索の組合せによるジョブショップスケジューリング問題の解法」, 情報処理学会論文誌, Vol. 37, No. 4, pp. 597-604 (1996).
- (1-34)横山想一郎・飯塚博幸・山本雅人 : 「JSP に対する部分的な割り付け規則の適用による局所探索法の提案」, 日本機械学会ロボティクス・メカトロニクス講演会講演論文集 (2014).
- (1-35)Beasley: OR-library Website, <http://people.brunel.ac.uk/~mastijb/jeb/info.html>.

## **2. 需給連携最適化方式の改良**

文献(2-1)に示す従来研究の需給連携最適化方式は、主問題を JSP として定式化し、ベンチマーク問題により解の質を比較している。そのため、主問題で定義した問題では、組立加工系工場の生産現場の制約条件の一部のみしか考慮していない。また、JSP は、理想化されたモデルに基づいて最適化している。つまり、文献(2-1)の需給連携最適化方式は、組立加工系工場の様々な生産現場に対して汎用性が低いといえる。

上記の従来研究の課題を解決し、組立加工系工場の現場に適用可能な需給双方向連携を実現する連携制御を実現するためには、供給側と需要側を精緻にモデル化した上で、両方のモデルを連携するデジタルツインが必要である。デジタルツインを構築するために、供給側は、エネルギープラントのボイラや蓄熱槽等の各設備をモデル化し、設備間の接続状況を基に設備モデルのネットワークを忠実に再現する必要がある。そのネットワークモデルを基にエネルギープラント運用計画を最適化する研究は行われており、実用化されている(2-2, 2-3)。つまり、需給双方向連携に利用可能な、供給モデルは既に存在している。需要側は、著者らの従来研究により、組立加工系工場の生産現場の様々な制約を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化することにより精緻に再現可能な生産シミュレータ(2-4)を用いて、工場の需要側のみを対象として、生産時間や労務費といった生産性の向上と、電力消費量や CO<sub>2</sub> 排出量といった環境負荷量の低減の観点で現場に適用可能な生産計画を最適化する研究を行っている(2-5, 2-6)。本シミュレータは、工場現場を忠実に再現できることから、工場現場に対して本シミュレータを用いた現場に適用可能な生産計画立案の実証が行われ、効果を確認している(2-4)。つまり、需給双方向連携に利用可能な、需要モデルは既に存在している。以上より、工場現場を精緻にモデル化し、需給双方向連携に利用可能な需給モデルは既に存在するといえる。

本章では、従来の需給連携最適化方式の主問題において、組立加工系工場の生産現場の様々な制約を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化した生産システムを基に生産計画最適化を行うため、主問題を JSP から生産シミュレータ上で表現した工場現場の生産計画最適化問題に変更した需給連携最適化方式の改良を行う。

### **2. 1 改良方式の概要**

図 2.1 に、文献(2-1)の従来の需給連携最適化方式(以下、従来方式)と、生産シミュレータを用いた現場適用可能な生産計画最適化問題を主問題、エネルギープラント最適運用計画問題を副問題とする提案する改良需給連携方式(以下、提案方式)の概要を示す。提案方式は、従来方式の主問題のジョブショップスケジューリング問題を、生産現場の様々な制約を全

て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化することにより精緻に再現可能な生産シミュレータによる生産計画最適化問題に変更することで、様々な組立加工系工場の生産現場に適用することができる。提案方式における主問題の生産計画最適化部と、副問題のエネルギープラント運用計画最適化部の構成と入出力データは、従来方式と変わらない仕様としている。

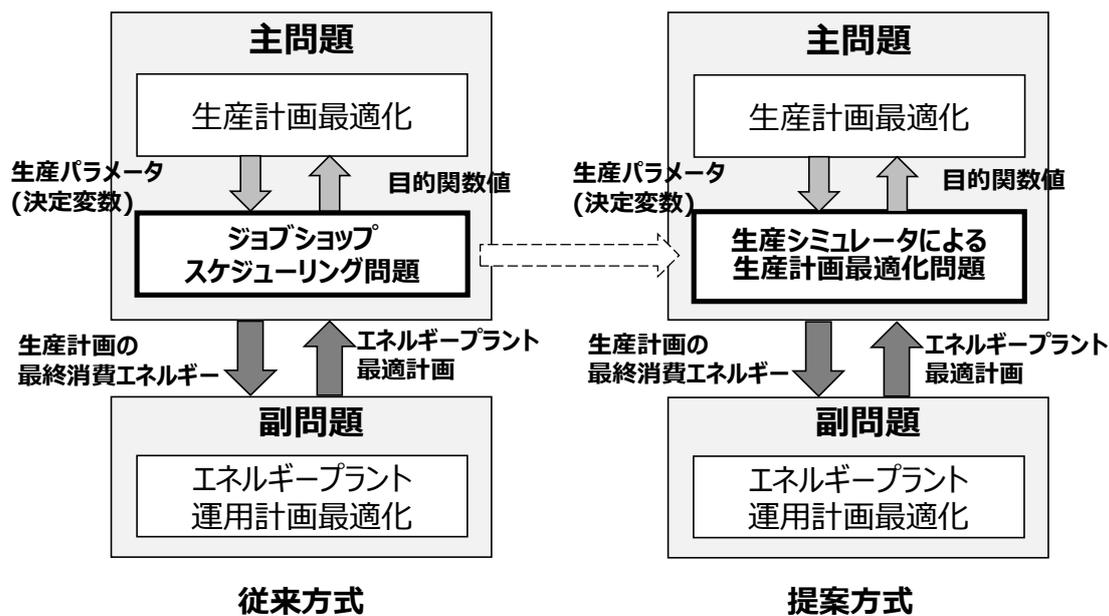


図 2.1 従来研究の需給連携方式と提案する需給連携方式比較

図 2.1 の右側に示す提案方式のアルゴリズムを以下に示す。

- Step.1 世代数を 1 とし、主問題において、生産計画最適化部が、乱数により求めた生産パラメータを生産シミュレータに渡す。
- Step.2 主問題において、生産シミュレータが、生産パラメータを基に求めた生産計画から 1 時間単位の最終消費エネルギー消費量を求め、副問題に渡す。
- Step.3 副問題において、エネルギープラント運用計画最適化部が、最終消費エネルギー消費量を基に最適なエネルギープラント運用計画を求め、生産シミュレータに渡す。
- Step.4 主問題において、生産シミュレータが、エネルギープラント最適運用計画と生産計画を基に、目的関数値を算出し、生産計画最適化部に渡す。
- Step.5 主問題において、生産計画最適化部が、目的関数値を基に生産パラメータを更新する。
- Step.6 世代数が設定した最大世代数と同じ値になった場合、終了する。それ以外の場合、世代数を 1 加算し Step.2 に戻る。

提案方式は、対象問題を主問題と副問題に分け、副問題を解くことで得た最適なエネルギープラント最適運用計画を主問題に送り、主問題にて生産計画とエネルギープラント最適運用計画を基に目的関数値を計算している。生産計画最適化部では、生産シミュレータによる生産計画最適化問題から送られる目的関数値を入力として生産計画最適化処理を行い、決定変数である生産パラメータを出力する。この 2 つのシステムはファイル連携しており、一方のシステムが処理している間、決められたファイルパス上でフラグファイルを監視し、フラグファイルが存在することをトリガーとして所定のファイルパスに保存されたデータファイルを読み込む。データファイル読み込み後、すべての処理が終了すると、完了のフラグファイルを生成し、再びフラグファイル監視に遷移する。エネルギープラント運用計画最適化では、生産シミュレータによる生産計画最適化問題から送られる生産計画の最終消費エネルギー負荷を入力として運用計画最適化処理を行い、エネルギープラント最適運用計画を出力する。この 2 つのシステムは、生産計画最適化部と同様の仕様でファイル連携している。提案方式の適用により、工場全体のエネルギー購入コストを含む生産コストを最小化することが期待できる。

図 2.2 に、提案方式を適用した場合の組立加工系工場の業務フロー図を示す。提案方式では、需給連携シミュレータを用いてモデル化した生産計画最適化問題とエネルギープラント最適運用計画問題を解く仕様のため、JSP 計画立案システムが生産管理システム、JSP 指標算出システムが二次エネルギー消費量計算機能と生産コスト計算機能にそれぞれ置き換わっている。基本的なフローについては、図 1.6 に示す従来の需給連携最適化方式のフローと同様である。

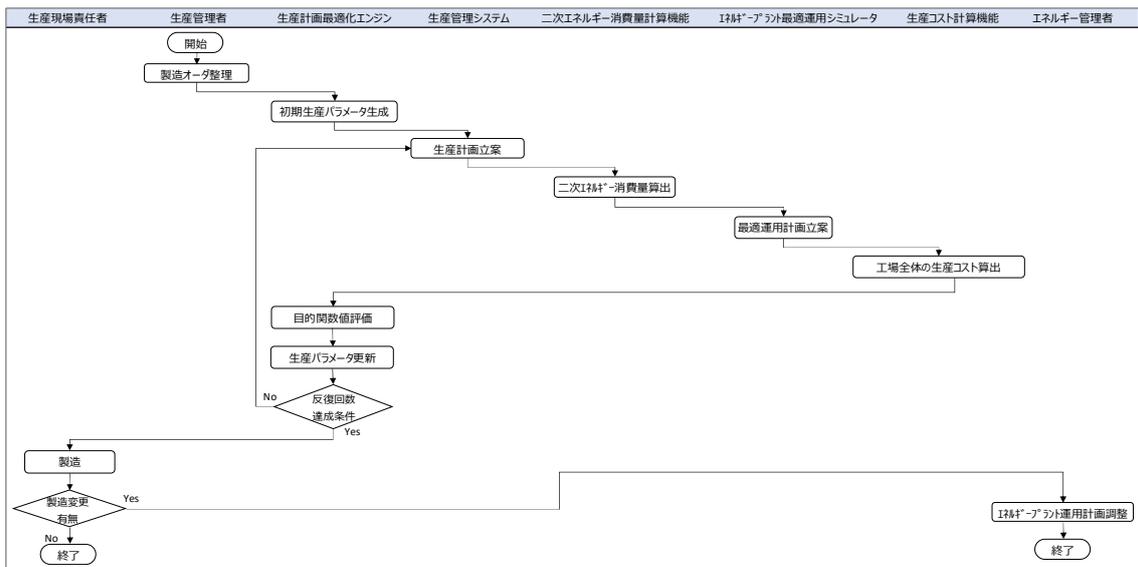


図 2.2 提案方式を適用した場合の組立加工系工場における業務フロー図

## 2.2 生産シミュレータの概要

生産シミュレータ<sup>(2-4)</sup>は、工場設備の稼働シミュレーションにより、評価方法が異なる生産性・エネルギー効率・環境負荷を、統一指標として生産コストへ換算し、高精度な予測を行うデジタルツールである。図 2.3 に生産シミュレータ概要図を示す。生産シミュレータ利用時は、事前に工場の設備構成や表 2.1 に示す設備毎の生産性や環境負荷評価用パラメータを設定し、生産シミュレーションモデルを構築する。続いて、生産管理システムにより立案した生産計画と設備稼働計画を入力として、設備稼働を模擬して、設備の運転状態、労働者

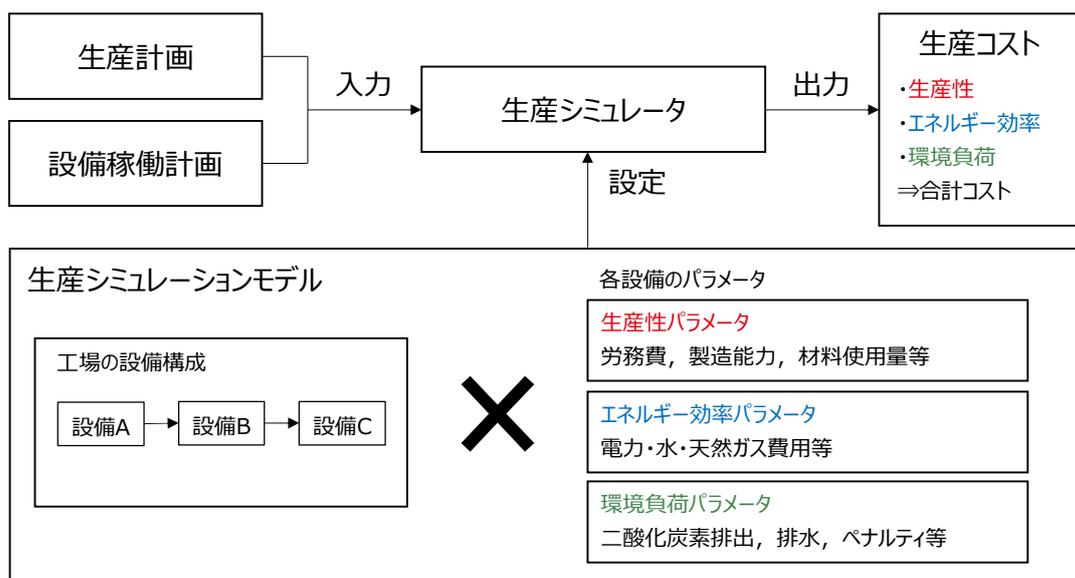


図 2.3 生産シミュレータの概要

表 2.1 生産シミュレータの入力パラメーター一覧

生産システムパラメータ	環境負荷/コストパラメータ	生産運用計画データ
<ul style="list-style-type: none"> <li>・ 生産設備構成</li> <li>・ 製造能力</li> <li>・ 生産モデル</li> <li>・ 材料使用量</li> <li>・ 作業員数</li> <li>・ 段取り替え時間</li> <li>・ シフト情報</li> <li>・ カレンダー情報</li> <li>・ 製品単価</li> </ul>	<ul style="list-style-type: none"> <li>・ 各生産設備の運転状態毎の環境負荷量</li> <li>・ 電力、排水、天然ガスに係る費用</li> <li>・ 労務費</li> </ul>	<ul style="list-style-type: none"> <li>・ 生産計画情報</li> <li>・ 設備稼働計画情報</li> </ul>

の稼働人数、環境負荷排出量の時系列データを算出する。最後に、前述の時系列データから、生産計画に係る生産コストを出力する。本研究では、生産管理システムとして、日本国内で広いシェアを持つ市販の生産計画立案ツール<sup>(27)</sup>を使用している。生産シミュレータは、設定により、全ての制約条件を考慮することができる。また、生産計画ツールで設定する生産パラメータのうち、顧客からの製造オーダーを満たしつつ生産計画の評価指標の改善に寄与するパラメータを決定変数として設定することで、一意の生産計画が得ることができる。つまり、生産シミュレータをブラックボックス最適化問題と扱う方がより自然である。本研究では、生産シミュレータで扱う生産計画最適化問題を、生産パラメータを入力データ、生産計画を基に算出した生産コストを出力データとするブラックボックス最適化問題として定義する。

シミュレータの精度については、組立加工系工場を対象として検証を行った。具体的には、特定のラインに対して、製品機種毎に1年間分の生産数量及び電力消費量データを基に、運転状態（生産中、立上げ、立下げ、段取り替え等）別の電力消費量を学習し、1週間分の生産計画を用いて電力消費量を予測した結果、対象とした1週間の実績値とシミュレーション予測データの誤差が日平均で約1.04%であり、生産計画に影響がないことを確認した。また、対象工場担当者からのフィードバックにより、シミュレーションによる予測値が妥当である評価を受けた。つまり、生産計画を立案する上で、本シミュレータが、妥当な精度を得られていることを確認した。

## 2.3 主問題の定式化

主問題では、複数の生産設備、作業員を有する組立加工系工場の生産システムを生産シミュ

シミュレータ<sup>(2-4)</sup>によりモデル化した上で、作業者の労務工数に応じた労務費、電力消費量に応じた買電コスト、電力起源のCO<sub>2</sub>排出量に応じたCO<sub>2</sub>排出コストから算出した生産コスト指標が最小となる生産計画を立案する。以下に、主問題である生産シミュレータで表現する生産計画最適化問題の定式化を示す。

### **2. 3. 1 生産シミュレータの各種生産パラメータと主問題との関係, 及び JSP との比較**

#### **(1) JSP の定式化上の特徴**

以下に、実際の生産現場を模擬した生産シミュレータと、現場の一部の制約条件のみを考慮し、理想化したモデルを用いる JSP との違いを明確化する上で必要な JSP の定式化上の特徴を示す。尚、JSP の詳細な定式化は、4.2 節に示す。

- ① 各ジョブ(生産ライン)に対する設備の使用順序と生産時間は予め決まっている。
- ② 設備の運転状態は製造状態と停止状態の 2 種類のみ存在し、それら運転状態間の状態遷移の定義や、状態遷移に掛かる時間は考慮されていない。

上記 JSP の定式化上の特徴と、提案法で用いる定式化上の違いを、以下に示す。

#### **(2) 生産シミュレータの各種生産パラメータと、決定変数及び制約条件との関係**

生産計画は、生産管理システムとして用いている市販の生産計画立案ツール<sup>(2-7)</sup>にて表現される。生産計画立案ツール<sup>(2-7)</sup>では、生産計画の立案に際し 8 種類のテーブルを事前に設定する。3 章の検証で用いた生産計画の設定例を以下に示す。尚、各テーブルは、より多くの設定列を有するが、簡単のため特に重要な設定列のみを示す。

##### **① 資源テーブル**

資源テーブルでは、表 2.2 に示す 5 項目を設定する。生産計画ツールでは、設備を主資源、労働者を副資源と定義している。資源コード列は、生産計画の対象となる設備名と労働者名を設定する。資源種別列は、資源コードが主資源(設備)か、副資源(労働者)かを設定する。コスト種別列は、副資源において請負単価か、時間単価かを設定する。本研究では、労働者に係る労務費は時間単価を用いてコスト換算するため、時間単価を設定する。時間単価列は、副資源の時間単価を設定する。適用可能シフト列は、後述のシフトテーブルで設定するシフトコードのうち、副資源が適用するシフトを設定する。

表 2.2 資源テーブル設定例

資源コード	資源種別	コスト種別	時間単価	適用可能シフト
Aライン設備	主資源	時間単価		
Aライン担当作業	副資源	時間単価	1500	1シフト;2シフト
Bライン設備	主資源	時間単価		
Bライン担当作業	副資源	時間単価	1500	1シフト;2シフト

② 工程テーブル

工程テーブルでは、表 2.3 に示す 2 項目を設定する。工程コード列は、生産計画の対象となる工程に紐づくコードを設定する。工程名列は、工程コードの工程名称を設定する。

④ 品目テーブル

品目テーブルでは、表 2.4 に示す 2 項目を設定する。品目コード列は、生産計画の対象となる製品名を設定する。品目種別列は、品目コードの状態を「原材料、購買品、半製品、中間品、製品」のうちから設定する。

⑤ 品目段取りテーブル

品目段取りテーブルでは、表 2.5 に示す 4 項目を設定する。資源コード列は、表 2.2 で設定した資源名称を設定する。前品目/後品目列は、表 2.4 で設定した品目コード名称を設定する。段取り時間列は、該当の資源における前後品目間の段取り時間(治具の取り換えや設備入れ替え等の時間)を設定する。

⑥ シフトテーブル

シフトテーブルでは、表 2.6 に示す 2 項目を設定する。シフトコード列は、生産計画対象のシフトの名称を設定する。パターン列は、該当シフトのシフト時間を設定する。

⑦ カレンダーテーブル

カレンダーテーブルでは、表 2.7 に示す 4 項目を設定する。資源列は、表 2.2 で設定した資源名称を設定する。日指定列は、該当の資源における期間を設定する。これは、生産計画の期間を表している。シフトコード列は、表 2.6 で設定したシフトコードを設定する。資源量列は、該当の副資源の人数を設定する。

表 2.3 工程テーブル設定例

工程コード	工程名
A_工程1	Aラインの組立工程
B_工程1	Bラインの組立工程

表 2.4 品目テーブル設定例

品目コード	品目種別
田製品A-加工完	半製品
製品A-加工完-1	中間品
製品A-加工完-100	中間品
製品A-加工完-2	中間品
製品A-加工完-200	中間品
製品A-加工完-3	中間品
製品A-加工完-300	中間品
製品A-未加工	購買品
田製品B-加工完	半製品
製品B-加工完-1	中間品
製品B-加工完-100	中間品
製品B-加工完-2	中間品
製品B-加工完-200	中間品
製品B-加工完-3	中間品
製品B-加工完-300	中間品
製品B-未加工	購買品

表 2.5 品目段取りテーブル設定例

資源	前品目	後品目	段取り時間
Aライン設備	製品A-加工完-100	製品C-加工完-100	4H
Aライン設備	製品A-加工完-100	製品B-加工完-100	30M
Aライン設備	製品A-加工完-100	製品D-加工完-100	3H
Aライン設備	製品B-加工完-100	製品A-加工完-100	3H
Aライン設備	製品B-加工完-100	製品C-加工完-100	2H
Aライン設備	製品B-加工完-100	製品D-加工完-100	20M
Aライン設備	製品C-加工完-100	製品A-加工完-100	20M
Aライン設備	製品C-加工完-100	製品B-加工完-100	20M
Aライン設備	製品C-加工完-100	製品D-加工完-100	3H
Aライン設備	製品D-加工完-100	製品A-加工完-100	10M
Aライン設備	製品D-加工完-100	製品B-加工完-100	3H
Aライン設備	製品D-加工完-100	製品C-加工完-100	1H
Bライン設備	製品A-加工完-300	製品C-加工完-300	4H
Bライン設備	製品A-加工完-300	製品B-加工完-300	2H
Bライン設備	製品A-加工完-300	製品D-加工完-300	3H
Bライン設備	製品B-加工完-300	製品A-加工完-300	20M
Bライン設備	製品B-加工完-300	製品C-加工完-300	40M
Bライン設備	製品B-加工完-300	製品D-加工完-300	10M
Bライン設備	製品C-加工完-300	製品A-加工完-300	30M
Bライン設備	製品C-加工完-300	製品B-加工完-300	3H
Bライン設備	製品C-加工完-300	製品D-加工完-300	10M
Bライン設備	製品D-加工完-300	製品A-加工完-300	30M
Bライン設備	製品D-加工完-300	製品B-加工完-300	3H
Bライン設備	製品D-加工完-300	製品C-加工完-300	10M

表 2.6 シフトテーブル設定例

シフトコード	パターン
終日	0:00-24:00
1シフト	8:30-12:15;12:45-17:00;17:30-20:00
2シフト	20:00-24:00;0:30-4:15;4:45-7:30
3シフト	0:30-4:15;4:45-7:30;8:30-12:15;12:45-17:00;17:30-24:00

表 2.7 カレンダーテーブル設定例

資源	日指定	シフトコード	資源量
*	*	3シフト	1
Bライン設備	2019/08/01-2019/08/08	3シフト	1
Aライン設備	2019/08/01-2019/08/08	3シフト	1
Aライン担当作業	2019/08/01-2019/08/08	1シフト;2シフト	1
Bライン担当作業	2019/08/01-2019/08/08	1シフト;2シフト	1

⑧ 製造 BOM テーブル

製造 BOM テーブルでは、表 2.8 に示す 13 項目を設定する。ここで、BOM とは Bill of Materials の略称であり、製品を製造するのに必要な部品情報を一覧として管理するものである。本研究で用いる生産計画立案ツール<sup>(2-7)</sup>では、表 2.8 に示すような製造 BOM を設定することで「ライン>工程>設備・労働者」という階層構造や、製品毎の設備の製造能力(製品 1 個当たりどのくらいの製造時間が掛かるかの値)を網羅的に表現する。表 2.8 の設定例をグラフ表示した結果を図 2.4 に示す。設定例では、2つの独立した工程 A と工程 B が存在し、それぞれの工程で製品 A を製造可能であることをグラフで表している。

⑨ オーダテーブル

オーダテーブルでは、表 2.9 に示す 8 項目を設定する。オーダコード列は、オーダ名称を設定する。品目列は、表 2.4 で設定した品目コードを設定する。最早開始日時列、納期列、オーダ数量列は、オーダコードに紐づくオーダ情報をそれぞれ設定する。ライン A 優先度列とライン B 優先度列は、それぞれライン A とライン B に対する製品の生産順序を設定する。

表 2.8 製造 BOM テーブル設定例

品目	工程番号	工程コード	指図種別	前段取り	製造	後段取り	有効条件式	指図コード	品目/資源	先行工程番号	必要資源量	重なり方法		
製品A-加工	1	start_dummy	入力指図					In	製品A-未加工	-1		ES		
			使用指図	0s				M	start_dummy			1		
			出力指図		OTHER.Order.NumSpec1		ME.NumSpec1>0		Out1	製品A-加工完-1				
			出力指図		OTHER.Order.NumSpec4		ME.NumSpec4>0		Out4	製品A-加工完-3				
	130	A_工程1	入力指図				ME.NumSpec1>0	In	製品A-加工完-1	1		ES		
			使用指図	0	10sp				M	Aライン設備			1	
			使用指図	0	0				S0	Aライン担当作業者			1	
	440	B_工程1	入力指図				ME.NumSpec4>0	In	製品A-加工完-3	1		ES		
			使用指図	0	20sp				M	Bライン設備			1	
			使用指図	0	0				S0	Bライン担当作業者			1	
	9999	end_dummy	入力指図		OTHER.Order.NumSpec1		ME.NumSpec1>0	In1	製品A-加工完-100	130		SS		
			入力指図		OTHER.Order.NumSpec4		ME.NumSpec4>0		In4	製品A-加工完-300	440		SS	
			使用指図		0s				M	end_dummy			1	

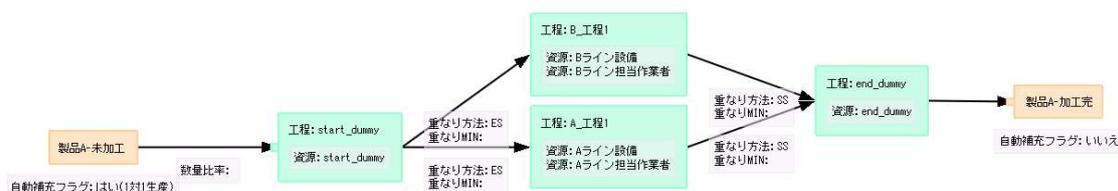


図 2.4 製造 BOM テーブルのグラフ表示結果

表 2.9 オーダテーブル設定例

オーダーコード	品目	最早開始日時	納期	オーダー数量	ラインA優先度	ラインB優先度
田製品A_1	製品A-加工完	2019/08/01 08:00:00	2019/08/02 08:00:00	700	5	4
田製品A_2	製品A-加工完	2019/08/01 08:00:00	2019/08/02 08:00:00	700	2	7
田製品B_1	製品B-加工完	2019/08/01 08:00:00	2019/08/02 08:00:00	600	1	1
田製品B_2	製品B-加工完	2019/08/01 08:00:00	2019/08/02 08:00:00	600	3	5
田製品C_1	製品C-加工完	2019/08/01 08:00:00	2019/08/02 08:00:00	500	4	8
田製品C_2	製品C-加工完	2019/08/01 08:00:00	2019/08/02 08:00:00	1200	7	2
田製品C_3	製品C-加工完	2019/08/01 08:00:00	2019/08/02 08:00:00	1100	8	3
田製品C_4	製品C-加工完	2019/08/01 08:00:00	2019/08/02 08:00:00	1100	6	6

図 2.5 に、上記設定をすることにより表現した生産計画例を示す。例えば、製造 BOM テーブルの製品毎の設備の製造能力とオーダーテーブルのオーダー数量が決まることで、製品毎の各生産時間が確定する。また、オーダーテーブルの各ラインの優先度と品目段取りテーブルの品目間の段取り替え時間が決まることで、製品の生産順序とそれに伴う段取り替え時間が確定する。図 2.6 に、図 2.5 の生産計画で設定した表 2.9 のオーダーテーブルのライン A 優先度列とライン B 優先度列の整数値を変更したときの生産計画例を示す。ライン A 優先度列とライン B 優先度列の整数値を変更することで、製品機種の生産順序が変更される。製品機種の生産順序が変わることで、表 2.5 に示す製品間の段取り替え時間が変わり、生産計画の終了時間が変わる。以上より、この問題の決定変数は、製品機種の生産順序になる。

JSP では、2.3.1 節(1)に示のように、事前に各ジョブ（ライン）に対する設備の技術的順序と生産時間は予め決まっているという制約を持つ。一方、実際の生産現場では、製品を製造する優先度に応じて製品の生産順序が変わり、それに伴い製品間の組合せに応じて決められた段取り替え時間が変わる。そのため、生産順序は状況に応じて変化し、生産順序が変わることで段取り替え時間を含む生産時間が変わる。また、JSP では、設備の運転状態は製造状態と停止状態の 2 種類のみ存在し、それら運転状態間の状態遷移の定義や、状態遷移に掛かる時間は考慮されていない。一方、実際の生産現場では、図 2.7 に示すように、製造状

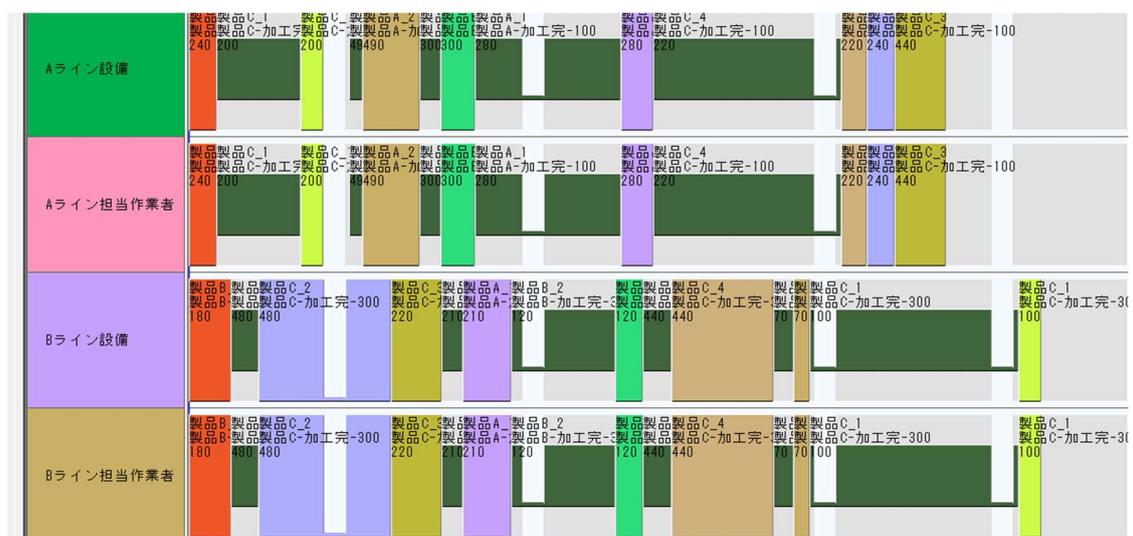


図 2.5 2.3.1 節で示すテーブル設定を適用した生産計画例

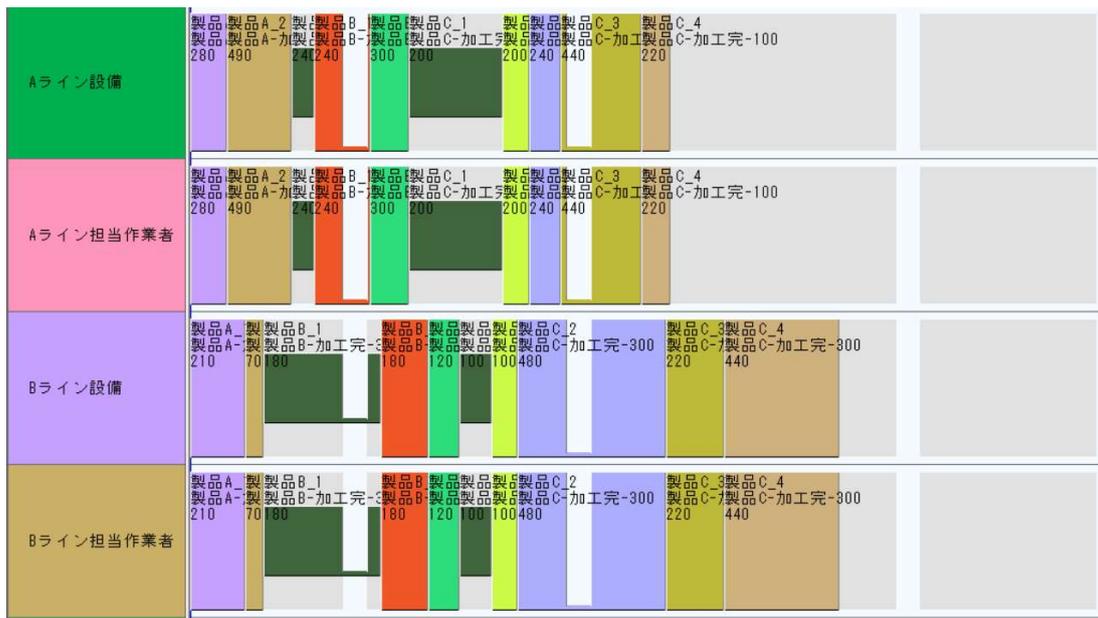


図 2.6 製品機種の生産順序を変更した際の生産計画例

態と停止状態の他に、設備の電源が入っておりいつでも生産可能な状態である待機状態の計 3 種類の運転状態が存在する。更に、生産現場では、設備の待機時間が一定時間を超えると設備を停止させることがある。そして、設備が停止した状態から起動させて生産可能な状態になるための立上げ時間は、設備の性能によって異なる。つまり、運転状態間の状態遷移に掛かる時間が定義されており、これによっても生産時間が変わってしまう。

上述のような実際の生産現場が持つ設備特性や運転状態、状態遷移に関する情報といった製造ルールを各種テーブルに事前設定することで、対象とする組立加工系工場の生産現場の制約条件を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化した一意の生産計画を表現することができる。つまり、この問題の制約条件は、上述の各種テーブルを設定することで考慮することができる。

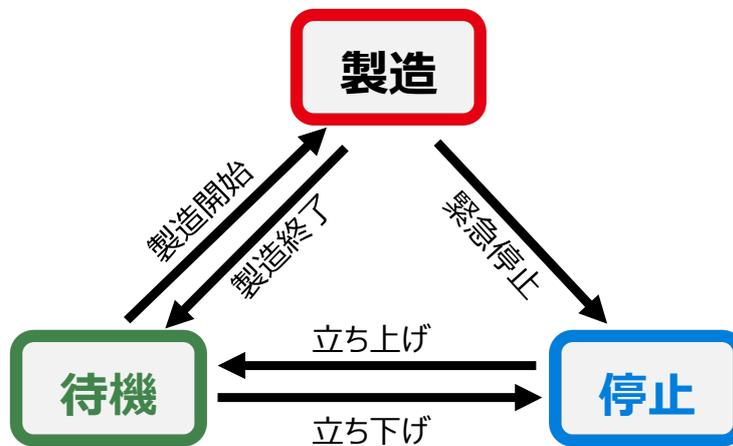


図 2.7 生産シミュレータによる設備の運転状態遷移ルール

### (3) 生産シミュレータの各種生産パラメータと、目的関数との関係

上述の生産計画ツール上の各種テーブルを設定することで、対象とする組立加工系工場の生産現場の制約条件を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化した一意の生産計画を表現することができる。生産シミュレータは、図 2.7 に示す設備の運転状態遷移ルールを基に、生産計画情報と設定ファイルで設定した設備の運転条件を入力として、設備毎の運転状態時系列データを算出する。そして、設備毎の運転状態時系列データと設定ファイルで設定した設備毎・エネルギー種別毎のエネルギー消費量と作業員数情報を入力として、エネルギー消費量時系列データと作業員数時系列データを算出する。次に、エネルギー消費量時系列データとライン毎の作業員数時系列データとエネルギー購入単価情報と労務費単価情報を入力として、生産計画の労務費、エネルギー購入コスト、CO<sub>2</sub>排出コストを算出する。最後に、各種コスト情報と生産計画の負荷時間と製造数量を入力として、生産コストに基づく評価指標を導出する。この問題の目的関数値は、上述の処理により算出した生産コストに基づく評価指標とする。

## 2. 3. 2 決定変数

2.3.1 節 (2) に述べたように、生産シミュレータ上で変更する生産パラメータは、各ラインの製品機種毎の製造優先度、つまり製品機種毎の生産順序であるため、主問題における決定変数は、製品機種の生産順序  $x_{ij}$  ( $i = 1, \dots, I, j = 1, \dots, J$ ) である。ここで、 $x_{ij}$  は  $i$  ラインにおける製品  $j$  の処理順序、 $I$  は生産ライン数、 $J$  は製品機種数である。尚、上述のように、生産順序が変わることにより段取り替え時間や立上げ・立下げ時間が変わり、全体の生産時間が変わるため、定式化全体では、JSP のような理想化されたモデルに基づいた生産計画最適化問題とは異なる定式化となっている。

## 2. 3. 3 目的関数

2.3.1 節 (3) に述べたように、生産シミュレータで最終的に出力する値は、生産コストに基づく評価指標であるため、主問題における目的関数は、生産性と環境負荷の指標毎の生産コストに基づく評価指標の最小化である。尚、本問題は一般的には多目的関数となるが、この 2 つの目的関数をスカラー化手法により 1 つの目的関数とした。

$$\min. (w_1 \times Pro(x_{ij}) + w_2 \times Ene(x_{ij}) + w_3 \times Env(x_{ij})) + P_1 + P_2 \quad (2.1)$$

$$Pro(x_{ij}) = \frac{\sum(LCost(x_{ij})) \times LTime}{PNum} \quad (2.2)$$

$$Ene(x_{ij}) = \frac{\sum(ECost(x_{ij})) \times LTime}{PNum} \quad (2.3)$$

$$Env(x_{ij}) = \frac{\sum(CCost(x_{ij})) \times LTime}{PNum} \quad (2.4)$$

$$LTime = OTime - STime \quad (2.5)$$

$$P_1 = (100 - CRate)^2 \quad (2.6)$$

$$P_2 = (PPF(x_{ij}) - FTh) \times 100 \quad (PPF(x_{ij}) > FTh) \quad (2.7)$$

ここで、 $Pro(x_{ij})$ :生産性評価指標、 $Ene(x_{ij})$ :エネルギー効率評価指標、 $Env(x_{ij})$ :環境負荷評価指標、 $w_1, w_2$ :目的関数の各項に対する重み係数( $w_1 + w_2 = 1$ )、 $P_1$ :生産計画の納期制約違反割合[%]を二乗したペナルティ値、 $P_2$ :買電コストから買電コスト閾値を減算し100を乗算したペナルティ値、 $LCost(x_{ij})$ :労務費、 $LTime$ :負荷時間、 $PNum$ :製造数量、 $ECost(x_{ij})$ :電力購入コスト、 $CCost(x_{ij})$ :CO<sub>2</sub>排出コスト、 $OTime$ :操業時間、 $STime$ :計画休止時間、 $CRate$ :生産計画の納期遵守率、 $PPF(x_{ij})$ :買電コスト、 $FTh$ :買電コスト閾値

$LCost(x_{ij})$ は、2.3.1節で述べたように、生産計画を基に算出した作業数時系列データと、事前に設定した労務費単価情報を入力として、作業数時系列データと時間帯ごとの労務費単価を乗算することで導出することができる。 $ECost(x_{ij})$ は、2.3.1節で述べたように、生産計画を基に算出した電力消費量時系列データと、事前に設定した電力単価情報を入力として、電力消費量時系列データと時間帯ごとの電力購入単価を乗算することで導出することができる。 $ECost(x_{ij})$ は、上述の方法で算出した $ECost(x_{ij})$ に対してCO<sub>2</sub>排出係数を乗算することでCO<sub>2</sub>排出量を算出し、CO<sub>2</sub>排出量にCO<sub>2</sub>排出費用係数を乗算することで導出することができる。

重み係数は、設定ファイルにより設定する仕様としており、生産現場毎にユーザが設定できるようにした。例えば、工場Aのユーザは生産性を優先した最適生産計画を、工場Bのユーザはエネルギー効率を優先した最適生産計画を、工場Cのユーザは3指標を均等に扱う最適生産計画を導出したいとする。その場合、工場Aのユーザは、[生産性,エネルギー効率,環境負荷]をそれぞれ[1,0,0]と設定し、工場Bのユーザは、[生産性,エネルギー効率,環境負荷]をそれぞれ[0,1,0]と設定し、工場Cのユーザは、[生産性,エネルギー効率,環境負荷]をそれぞれ[0.33,0.33,0.33]と設定することで、各ユーザが所望の生産計画を得ることができる。ペナルティ値 $P_1$ 、 $P_2$ は外点ペナルティ法により計算される。具体的に説明すると、生産計画立案時に算出される生産計画の納期遵守率と、エネルギープラント最適運用計画立案時に算出される買電コスト、及び買電コストの閾値を用いて、納期遵守違反率を二乗したペナルティ値(式(2.7)参照)と買電コスト閾値違反量に係数を乗算したペナルティ値(式(2.7)参照)を目的関数値に付加する。

## 2.3.4 制約条件

2.3.1 節 (2)に述べたように、実際の生産現場が持つ設備特性や運転状態、状態遷移に関する製造ルールを各種テーブルに事前設定することで、対象とする組立加工系工場の生産現場の制約条件を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化した一意の生産計画を表現することができる。尚、生産シミュレータで表現する生産計画最適化問題は、ブラックボックス最適化問題として定式化されるため、制約条件を含めた現場の生産システムを、生産シミュレータを用いてモデル化し、現場の工場をデジタルツインとして再現している。

## 2.4 副問題の定式化

副問題であるエネルギープラントは、電気学会産業応用部門「情報知能システムの新展開とその産業応用調査専門会」がまとめた「産業応用のための最適化ベンチマーク問題集」の3.2 節「エネルギープラント運用計画のための最適化ベンチマーク問題」にて提案されたモデルである<sup>(2-8)</sup>。そのエネルギープラントを図 2.8 に示す。このプラントは、電力会社からの購入電力とガスを購入し、ガスタービンの発電により電力エネルギーを供給し、蒸気負荷に対しては、ガスタービンの排熱とボイラでの蒸気生成、熱負荷にはターボ冷凍機、蒸気吸収式冷凍機によって冷水を供給する構成となっている。本論文では副問題を線形計画問題として定式化し、数理計画法のソルバーの1つである GLPK<sup>(2-9)</sup>を用いて解いている。

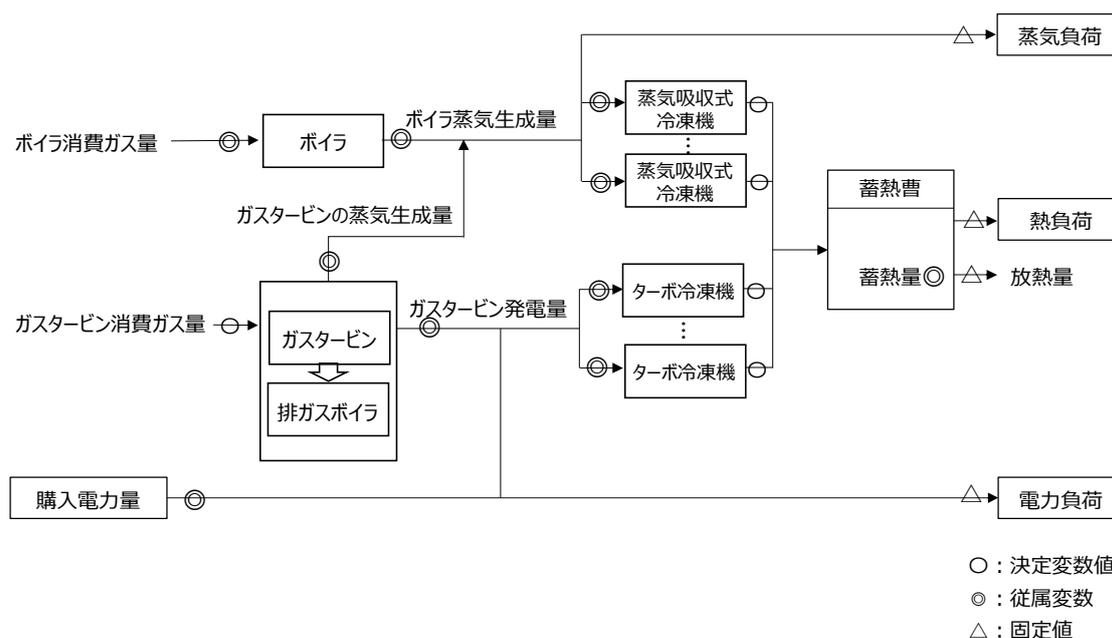


図 2.8 対象としたエネルギープラント

## 2. 4. 1 決定変数

副問題における決定変数を以下に示す，決定変数は，1日24時間，つまり24点の入出力の連続値であり，各変数は1日当たり24点( $K = 24$ )持っている。

- ・ ターボ冷凍機の熱出力量  $x_{ti}^k$  ( $k = 1, \dots, K, i = 1, \dots, N_t$ )  
ここで， $N_t$ ：ターボ冷凍機の台数
- ・ 蒸気吸収式冷凍機の熱出力量  $x_{si}^k$  ( $k = 1, \dots, K, i = 1, \dots, N_s$ )  
ここで， $N_s$ ：蒸気吸収式冷凍機の台数
- ・ ガスタービンの消費ガス量  $x_{gi}^k$  ( $k = 1, \dots, K, i = 1, \dots, N_g$ )  
ここで， $N_g$ ：ガスタービンの台数

## 2. 4. 2 目的関数

副問題における目的関数は，エネルギープラントにおける1日の運用コスト（購入電力量と購入ガス量の和）の最小化とする。

$$\min EC = \sum_{k=1}^K \left\{ C_{Ep}^k e_p^k + C_{Gp}^k \left( \sum_{i=1}^{N_g} x_{gi}^k + \sum_{i=1}^{N_b} g_{bi}^k \right) \right\} \quad (2.8)$$

ここで， $K$ :1日の分割点数， $C_{Ep}^k$ :時刻 $k$ における電気購入単価， $e_p^k$ :時刻 $k$ における電気購入量， $C_{Gp}^k$ :時刻 $k$ におけるガス購入単価， $N_g$ :ガスタービン数， $N_b$ :ボイラ数， $g_{bi}^k$ :時刻 $k$ における $i$ 台目のボイラにおける消費ガス量

$e_p^k$ は従属変数であり， $x_{ti}^k$ によって計算されるターボ冷凍機の消費電力量と電力供給量によって計算される。 $g_{bi}^k$ も同様に従属変数であり， $x_{si}^k$ により導出される。

## 2. 4. 3 制約条件

副問題における制約条件を以下に示す。

### ① 需給バランス制約

電力，蒸気，熱の供給量は，需要設備が必要とする電力，蒸気，熱の各負荷とバランスを取る必要がある。電力は購入電力量，蒸気はボイラ出力量，そして，熱は蓄熱量で均衡

を保っている。例えば、熱の負荷予測値がずれた場合は、蓄熱槽に貯められた熱で対応することで熱バランスをとっており、このようなしわ取りを考慮した需給バランス制約を以下に示す。

➤ 電力バランス制約

電力バランスに関する制約を以下に示す。

$$e_p^k + \sum_{i=1}^{N_g} \{f_{ge}(x_{gi}^k)\} = \sum_{i=1}^{N_t} \{f_{ti}(x_{ti}^k)\} + E_L^k \quad (k = 1, \dots, K) \quad (2.9)$$

ここで、 $f_{ge}(x_{gi}^k)$ ：時刻 $k$ における $i$ 台目のガスタービン発電量、 $f_{ti}(x_{ti}^k)$ ：時刻 $k$ における $i$ 台目のターボ冷凍機電気入力、 $E_L^k$ ：時刻 $k$ における電力負荷

(2.9)式は、各時刻における電力供給と電力需要は等しいことを表している。

➤ 蒸気バランス制約

上記バランスに関する制約を以下に示す。

$$\sum_{i=1}^{N_g} \{f_{gs}(x_{gi}^k)\} + \sum_{i=1}^{N_b} \{f_b(x_{bi}^k)\} = \sum_{i=1}^{N_s} \{f_{sj}(x_{si}^k)\} + S_L^k \quad (k = 1, \dots, K) \quad (2.10)$$

ここで、 $f_{gs}(x_{gi}^k)$ ：時刻 $k$ における $i$ 台目のガスタービン蒸気生成量、 $f_b(x_{bi}^k)$ ：時刻 $k$ における $i$ 台目のボイラ蒸気生成量、 $f_{sj}(x_{si}^k)$ ：時刻 $k$ における $i$ 台目の蒸気吸収式冷凍機蒸気入力、 $S_L^k$ ：時刻 $k$ における蒸気負荷（固定値）

(2.10)式は、各時刻における蒸気入力と蒸気出力は等しいことを表している。

➤ 熱バランス制約

熱バランスに関する制約を以下に示す。

$$Q_{ts}^0 = Q_{ts}^{init} \quad (2.11)$$

$$Q_{ts}^k = Q_{ts}^{k-1} + \sum_{i=1}^{N_t} \{x_{ti}^k\} + \sum_{i=1}^{N_s} \{x_{si}^k\} - Q_L^k - Q_{loss} \quad (k = 1, \dots, N) \quad (2.12)$$

ここで、 $Q_{ts}^{init}$ ：初期蓄熱量、 $Q_{ts}^k$ ：時刻 $k$ における蓄熱槽の蓄熱量、 $Q_L^k$ ：時刻 $k$ における

熱負荷（固定値）， $Q_{loss}$ ：単位時間あたりの蓄熱ロス（固定値）

(2.12)式は，各時刻における蓄熱量は，前時刻の蓄熱量に熱入力を加え，熱負荷と蓄熱ロスを引いたものと等しいことを表している。

## ② 機械特性

各機器の入出力特性（線形）と上下限值に関する機械特性制約は，以下のように表現できる。

### ➤ ガスタービン

ガスタービンの機械特性制約を以下に示す。

$$f_{ge}(x_{gi}^k) = a_{gei}x_{gi}^k \quad (k = 1, \dots, K, i = 1, \dots, N_g) \quad (2.13)$$

$$E_{gi}^{min}y_{gi}^k \leq f_{ge}(x_{gi}^k) \leq E_{gi}^{max}y_{gi}^k \quad (k = 1, \dots, K, i = 1, \dots, N_g) \quad (2.14)$$

ここで， $a_{gei}$ ： $i$ 台目のガスタービン発電量係数， $a_{gsi}$ ： $i$ 台目のガスタービン蒸気生成量係数， $E_{gi}^{min}$ ： $i$ 台目のガスタービン発電量下限値， $E_{gi}^{max}$ ： $i$ 台目のガスタービン発電量上限値， $y_{gi}^k \in \{0,1\}$ ：時刻 $k$ における $i$ 台目のガスタービンの起動停止状態

(2.13)式は，時刻 $k$ における $i$ 台目のガスタービン発電量は線形であることを表している。

(2.14)式は，時刻 $k$ における $i$ 台目のガスタービン発電量は，同時刻における $i$ 台目のガスタービンの起動停止状態と $i$ 台目のガスタービン発電量下限値を乗算した値以上であり，且つ同時刻における $i$ 台目のガスタービンの起動停止状態と $i$ 台目のガスタービン発電量上限値を乗算した値以下であることを表している。

### ➤ ターボ冷凍機

ターボ冷凍機の機械特性制約を以下に示す。

$$f_{ti}(x_{ti}^k) = a_{ti}x_{ti}^k \quad (k = 1, \dots, K, i = 1, \dots, N_t).. \quad (2.15)$$

$$Q_{ti}^{min}y_{ti}^k \leq x_{ti}^k \leq Q_{ti}^{max}y_{ti}^k \quad (k = 1, \dots, K, i = 1, \dots, N_t) \quad (2.16)$$

ここで， $a_{ti}$ ： $i$ 台目のターボ冷凍機電気入力係数， $Q_{ti}^{min}$ ： $i$ 台目のターボ冷凍機熱出力下限値， $Q_{ti}^{max}$ ： $i$ 台目のターボ冷凍機熱出力上限値， $y_{ti}^k \in \{0,1\}$ ：時刻 $k$ における $i$ 台目のターボ冷凍機の起動停止状態

(2.15)式は，時刻 $k$ における $i$ 台目のターボ冷凍機電気入力は線形であることを表してい

る。(2.16)式は，時刻 $k$ における $i$ 台目のターボ冷凍機の熱出力量は，同時刻における $i$ 台目

のターボ冷凍機の起動停止状態と*i*台目のターボ冷凍機熱出力下限値を乗算した値以上であり、且つ同時刻における*i*台目のターボ冷凍機の起動停止状態と*i*台目のターボ冷凍機熱出力上限値を乗算した値以下であることを表している。

➤ 蒸気吸収式冷凍機

蒸気吸収式冷凍機の機械特性制約を以下に示す。

$$f_{si}(x_{si}^k) = a_{si}x_{si}^k \quad (k = 1, \dots, K, i = 1, \dots, N_s). \quad (2.17)$$

$$Q_{si}^{min}y_{si}^k \leq x_{si}^k \leq Q_{si}^{max}y_{si}^k \quad (k = 1, \dots, K, i = 1, \dots, N_s) \quad (2.18)$$

ここで、 $a_{si}$  : *i*台目の蒸気吸収式冷凍機蒸気入力係数、 $Q_{si}^{min}$  : *i*台目の蒸気吸収式冷凍機熱出力下限値、 $Q_{si}^{max}$  : *i*台目の蒸気吸収式冷凍機熱出力上限値、 $y_{si}^k \in \{0,1\}$  : 時刻*k*における*i*台目の蒸気吸収式冷凍機の起動停止状態

(2.17)式は、時刻*k*における*i*台目の蒸気吸収式冷凍機蒸気入力は線形であることを表している。(2.18)式は、時刻*k*における*i*台目の蒸気吸収式冷凍機の熱出力量は、同時刻における*i*台目の蒸気吸収式冷凍機の起動停止状態と*i*台目の蒸気吸収式冷凍機熱出力下限値を乗算した値以上であり、且つ同時刻における*i*台目の蒸気吸収式冷凍機の起動停止状態と*i*台目の蒸気吸収式冷凍機熱出力上限値を乗算した値以下であることを表している。

➤ ボイラ

ボイラの機械特性制約を以下に示す。

$$f_b(x_{bi}^k) = a_{bi}x_{bi}^k \quad (k = 1, \dots, K, i = 1, \dots, N_b). \quad (2.19)$$

$$S_{bi}^{min}y_{bi}^k \leq f_b(x_{bi}^k) \leq S_{bi}^{max}y_{bi}^k \quad (k = 1, \dots, K, i = 1, \dots, N_b) \quad (2.20)$$

ここで、 $a_{bi}$  : *i*台目のボイラ蒸気生成量係数、 $S_{bi}^{min}$  : *i*台目のボイラ蒸気生成量下限値、 $S_{bi}^{max}$  : *i*台目のボイラ蒸気生成量上限値、 $y_{bi}^k \in \{0,1\}$  : 時刻*k*における*i*台目のボイラの起動停止状態

(2.19)式は、時刻*k*における*i*台目のボイラ蒸気生成量は線形であることを表している。(2.20)式は、時刻*k*における*i*台目のボイラ蒸気生成量は、同時刻における*i*台目のボイラの起動停止状態と*i*台目のボイラ蒸気生成量下限値を乗算した値以上であり、且つ同時刻における*i*台目のボイラの起動停止状態と*i*台目のボイラ蒸気生成量上限値を乗算した値以下であることを表している。

➤ 蓄熱曹

蓄熱曹の機械特性制約を以下に示す。

$$Q_{ts}^{min} \leq Q_{ts}^k \leq Q_{ts}^{max1} \quad (k = 1, \dots, K - 1) \quad (2.21)$$

$$Q_{ts}^{min} \leq Q_{ts}^k \leq Q_{ts}^{max2} \quad (k = K) \quad (2.22)$$

ここで、 $Q_{ts}^{min}$  : 蓄熱量下限値、 $Q_{ts}^{max1}$  : 時刻 $k = 1, \dots, K - 1$ での蓄熱量上限値、  
 $Q_{ts}^{max2}$  : 時刻 $k = K (= 24)$ での蓄熱量上限値

(2.21)(2.22)式は、各時刻における蓄熱量は蓄熱量下限値以上であり、且つ蓄熱量上限値以下の値を推移することを表している。

上記定式化は、線形計画問題であるため、各種線形計画法の適用が可能となる。本方式では、副問題にて求めた二次エネルギーコストは主問題に送られ、主問題にて二次エネルギーコストを考慮した工場における生産コストの評価が行われる。

#### 2. 4. 4 副問題に対する解法アルゴリズム

副問題であるエネルギープラントの最適運用は、主問題から送られてきた生産計画に対し最終消費エネルギー消費量を求め、その求めた最終消費エネルギー消費量をエネルギー負荷としてエネルギープラントの最適運用解を求める。以下に手順を示す。

**Step. 1** 主問題から送られてきた生産計画に対し、設備の運転状態に応じた1時間毎の最終消費エネルギー量を求める。

**Step. 2** Step. 1で求めた最終消費エネルギー消費量を負荷として、エネルギープラントの最適運用解を求め、二次エネルギーコストを求める。

**Step. 3** Step. 2で求めた二次エネルギーコストを主問題に送る。

Step. 1においては、対象となる生産計画で利用する設備のエネルギー消費量から最終消費エネルギーを求める。

Step. 2においては、副問題となるエネルギープラント最適運用は線形計画問題であるので、ここでは Gnu Linear Programming Kit (以下、GLPK) <sup>(2-9)</sup>を用いている。

## 2. 5 需給連携最適化実現に向けた並列分散需給連携シミュレータの提案

2章で提案する需給連携最適化方式の実現のため、工場の需要側を対象としてモデリングを行い、生産計画のコスト算出が可能な生産シミュレータ<sup>(2-4)</sup>を、工場の需要側と供給側の両方を対象にシミュレーション可能にするよう機能拡張したシミュレータ(以下、需給連携シミュレータ)を新たに開発した<sup>(2-10)</sup>。需給連携シミュレータは、工場における生産設備の稼働シミュレーションにより、評価方法が異なる生産性指標や環境負荷指標を、統一指標として生産コストへ換算し、高精度な予測を行う生産シミュレータに対し、生産計画に応じた電力消費量データを入力データとして最適エネルギープラント運用計画データを出力するエネルギープラントシミュレータを連携し、最適な需給計画を導出するデジタルツールである。需給連携シミュレータの処理フロー図を図 2.9 に示す。需給連携シミュレータは、最適化手法(図 2.9 中、最適化手法)、生産シミュレータ(図 2.9 中、生産シミュレータ)、エネルギープラント最適運用シミュレータ(図 2.9 中、エネルギープラントシミュレータ)の3つの

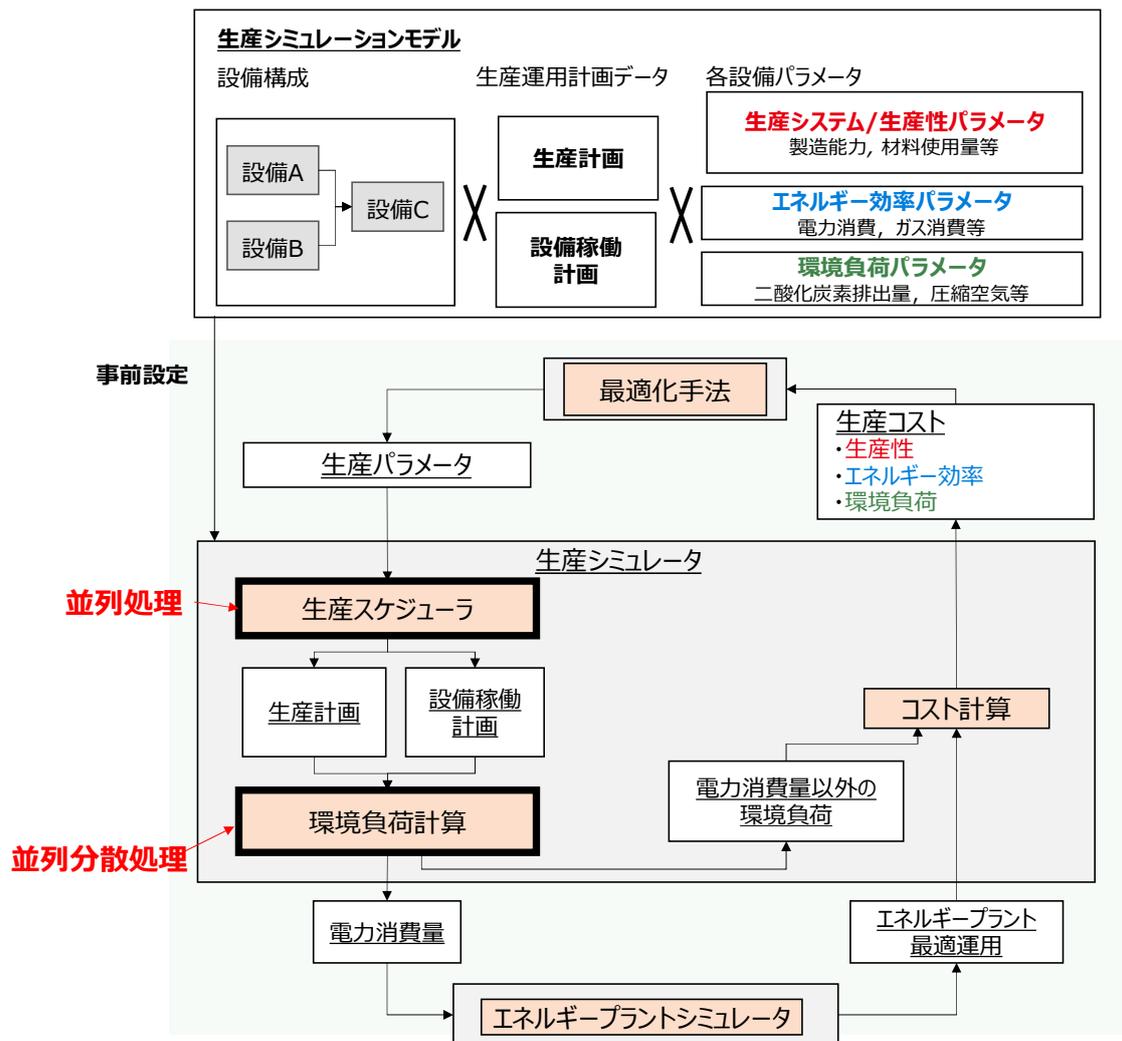


図 2.9 需給連携シミュレータ機能構成図

システムから構成される。需給連携シミュレータ活用時は、生産シミュレータでは、2.2節に示すように、事前に工場の設備構成や設備毎の生産性パラメータや環境負荷/コストパラメータ、生産運用計画データを設定し、シミュレーションモデルを構築する。エネルギープラント最適運用シミュレータでは、事前にエネルギープラントの設備構成や設備毎の運転特性データを設定し、シミュレーションモデルを構築する。需給両方のモデルを構築した後の需給連携シミュレータの処理手順を以下に示す。

- Step.1 最適化手法が、初期生産パラメータ(図 2.9 中、生産パラメータ)を生産シミュレータに渡す。
- Step.2 生産シミュレータの生産計画立案部(図 2.9 中、生産シミュレータ)が、生産パラメータを基に生産計画(図 2.9 中、生産計画)と設備稼働計画(図 2.9 中、設備稼働計画)を求める。
- Step.3 生産シミュレータの環境負荷量計算部(図 2.9 中、環境負荷計算部)が、2つの計画を基に1時間単位の環境負荷量を求め、そのうち電力消費量(図 2.9 中、電力消費量)をエネルギープラント最適運用シミュレータ(図 2.9 中、エネルギープラントシミュレータ)に渡す。
- Step.4 エネルギープラント最適運用シミュレータが、電力消費量を基にエネルギープラント最適運用計画を求め、生産シミュレータの生産コスト計算部(図 2.9 中、コスト計算部)に渡す。
- Step.5 生産シミュレータの生産コスト計算部が、エネルギープラント最適運用計画と生産計画を基に目的関数値(図 2.9 中、生産コスト)を算出し、最適化手法に渡す。
- Step.6 最適化手法が、目的関数値を基に生産パラメータを更新する。
- Step.7 世代数が事前に設定した最大世代数に達したら終了する。そうでなければ、Step.2に戻る。

尚、実際の工場では生産計画の変化により、電力負荷以外に蒸気負荷や熱負荷も変動するが、本研究では簡単のため、蒸気負荷と熱負荷は一定とし、電気負荷を生産計画により変動させる仕様とした。

本研究で適用するメタヒューリスティック手法は全て多点探索手法であるため、改良需給連携最適化方式の生産シミュレータ部は、 $K$ 個の個体の目的関数値を計算する際、 $K$ 回生産シミュレータを実行する必要がある。従来のシミュレータは、個体毎の処理を逐次処理していた。更に、環境負荷量計算部(図 2.9 中、環境負荷量計算部)は、全処理の中で計算機負荷が最も大きく、計算時間のボトルネックとなっていた。具体的には、事前シミュレーションにより、生産シミュレータの環境負荷量計算部の処理時間は、全処理時間の90%以上を占めている。一方、提案する並列分散型生産シミュレータは、GUIで設定したスレッド数に応じて、各個体に対応する生産シミュレータの生産計画立案部(図 2.9 中、生産計画立案部)と

環境負荷量計算部(図 2.9 中, 環境負荷量計算部)を並列処理することができる。更に, 環境負荷量計算部(図 2.9 中, 環境負荷量計算部)は, 複数の計算機で分散処理することもできる。これにより, 全処理時間短縮を実現することができる。尚, メタヒューリスティック手法による処理時間は全処理時間の 1%程度であり, それに対し生産シミュレータによる処理時間は全処理時間の 90%以上を占めていることから, 本研究では改善効果が大きい生産シミュレータのみに対して並列分散処理を行った。

提案する並列分散需給連携シミュレータを用いたシミュレーション結果は, 3.3.4 節にて説明する。

## 第2章の参考文献

- (2-1) 川口嵩平・福山良和：「改良近傍スケジューリング生成法を用いた改良型並列 Reactive Hybrid Particle Swarm Optimization による最適生産計画とエネルギープラント最適運用計画の統合最適化」, 電学論 D, Vol. 139, No. 12, pp.1005-1014 (2019).
- (2-2) 項東輝・川森亨・福山良和：「原動力設備プラントの最適運用と適用事例」, 富士技報, Vol.77, No.2, pp.166~170 (2004).
- (2-3) 横河電機株式会社データ駆動型 プラント最適操業支援ソリューション (DDMOnEX) 事例紹介, <https://www.yokogawa.co.jp/solutions/solutions/ddmonex/>.
- (2-4) 佐藤隆臣・川口嵩平・菊地健斗：「環境配慮型生産改善フレームワーク」, 三菱電機技報, Vol.96, No.9, pp.42-45 (2022).
- (2-5) K. Takahashi, Y. Fukuyama, S. Kawaguchi, and T. Sato, "Optimal Production Scheduling using a Production Simulator and Multi-population Global-best Modified Brain Storm Optimization," Proc. of IEEE World Congress on Computational Intelligence (2022).
- (2-6) 片桐瑠星・福山良和・川口嵩平・佐藤隆臣・高橋賢二郎：「実用的な生産シミュレータと Modified Brain Storm Optimization を用いた生産開始時間最適化の提案」, 2022 年度電気学会産業応用部門大会, 5-17 (2022).
- (2-7) 生産スケジューラ Asprova ウェブサイト, <https://www.asprova.jp/>.
- (2-8) 電気学会 情報知能システムの新展開とその産業応用調査専門委員会：「産業応用のための最適化ベンチマーク問題集」, 電気学会技術報告 第 1287 号, 3.2 節 (2013).
- (2-9) GLPK ウェブサイト, <https://en.wikibooks.org/wiki/GLPK>.
- (2-10) 川口嵩平・福山良和：「改良整数型 Adaptive Population-Based Incremental Learning と生産シミュレータによる工場需給連携最適化フレームワーク」, 電学論 D, Vol. 143, No. 11, pp. 728-741 (2023).

### **3. 改良需給連携最適化方式の主問題に対するメタヒューリスティック手法の提案**

生産シミュレータを用いたこれまでの従来研究により、生産シミュレータを用いた生産計画最適化問題は、最適生産計画立案に要する時間が限られ、且つ1回当たりの生産シミュレータの処理時間が数十秒掛かることから、生産シミュレータの実行回数が少ない条件下でより質の高い解を導出する必要があった（課題4）。

本章では、上述の課題4を解決するため、2章で説明した改良需給連携最適化方式の主問題である生産計画最適化問題に対して、以下のような特徴を持つ3つのメタヒューリスティック手法を提案する。

- ・解候補の確率モデルを用いた分布推定アルゴリズムをベースとし、ブラックボックス最適化問題に対して、効率的な探索が可能である手法（全手法（提案手法 3-1, 3-2, 3-3））。
- ・学習率を探索に応じて適応的に変更するアルゴリズムの開発により、様々な工場にパラメータチューニングせずに適用可能である手法（整数型 APBIL（提案手法 3-1））。
- ・初期学習率の設定によらず、探索に応じて学習率を適切且つ適応的に変更するアルゴリズムの開発により、様々な工場にパラメータチューニングせずに適用可能である手法（改良整数型 APBIL（提案手法 3-2））。
- ・探索の前半に探索の多様化を強め、探索の後半に探索の集中化を強めることで、目的関数値の評価回数が少ない問題に対して効率的に探索可能である手法（改良整数型 APBIL-RTS（提案手法 3-3））。

提案手法は、2.2節で説明した生産シミュレータを用いて、組立加工系工場の一部工程を対象としてモデル化を行った生産計画最適化問題に適用し、解の質の向上と計算時間の高速化の観点で改善可能かを検証した。尚、3.1節及び3.2節は、生産シミュレータを用いた生産計画最適化問題を対象問題とし、3.3節及び3.4節は、改良需給連携最適化方式を対象問題として、メタヒューリスティック手法の比較検証を行った。

#### **3. 1 生産シミュレータを用いた生産計画最適化問題に対する整数型 PBILの適用**

##### **3. 1. 1 PBILの概要**

PBILは、1994年に S. Baluja らによって提案された探索手法<sup>(3-1)</sup>で、解空間を効率的に探索することを目的として解候補の確率モデルを用いた分布推定アルゴリズム<sup>(3-2)</sup>の一種である。本手法は、遺伝的アルゴリズム(以下、GA)と同様、2進数で表現した解要素(以下、遺伝子座)を0か1で表現し、1つの解を表現する(表 3.1 参照)。GAは、交叉や突然変異、

自然淘汰により解そのものを改善することで解空間を探索する。一方、PBILは、解の各遺伝子座が、0か1のどちらが良い解かを遺伝子座毎に確率で表した確率ベクトルを生成する(表3.2参照)。確率ベクトルによって生成した複数の解のうち、最も良い解の遺伝子座の値(0,1)に従って確率ベクトルを更新することで、生成される解が解空間を探索する。これら解の分布を考慮した最適化手法は、解自体を更新する遺伝的アルゴリズム等の従来の最適化手法に比べ、ブラックボックス最適化問題を含むあらゆる最適化問題の解空間をより効率的に探索できることが確認されている。例えば、連続変数に対する最適化手法であるCovariance Matrix Adaptation Evolution Strategy (以下、CMA-ES)は、進化計算分野で権威ある国際学会のひとつであるIEEE Congress on Evolutionary Computationのコンペティションにおいて、近年優秀な成績を残している(3-3)。また、著者らの研究室における先行研究では、様々な対象問題に対して、PBIL系手法の有効性を確認している(3-4,5)。これら背景を踏まえ、本研究では、ブラックボックス最適化問題と定義可能な生産シミュレータを用いた生産計画最適化問題に対して、PBIL系手法を適用した。

確率ベクトルの各確率を更新する際に用いる競合学習の考え方を取り入れたPBILの確率ベクトル更新式を以下に示す。

$$P_n^{gen+1} = (1.0 - \epsilon) \times P_n^{gen} + best_n^{gen} \times \epsilon \quad (n = 1, \dots, N) \quad (3.1)$$

ここで、 $P_n^{gen}$  : 世代数 $gen$ で $n$ 番目の遺伝子座の値が1になる確率、 $gen$  : 世代数、 $N$  : 遺伝子座の数、 $\epsilon$  : 学習率、 $best_n^{gen}$  : 世代数 $gen$ で生成された遺伝子集団の内、最良目的関数値を持つ個体の $n$ 番目の遺伝子座の値

突然変異は、0から1の一様乱数が突然変異確率より小さければ、以下の式にて $P_n^{gen+1}$ を修正する。

$$P_n^{gen+1} = (1.0 - SHIFT) \times P_n^{gen+1} + Random(0 \text{ or } 1) \times (SHIFT) \quad (n = 1, \dots, N) \quad (3.2)$$

ここで、 $SHIFT$  : 確率に対して突然変異が影響する量、 $Random(0 \text{ or } 1)$  : 一様乱数により0か1か決定する数値

表 3.1 PBIL の決定変数例

$x_1$	$x_2$	$x_3$	$x_4$
1	1	0	0

表 3.2 PBIL の確率ベクトル例

$x_1$	$x_2$	$x_3$	$x_4$
0.7	0.6	0.3	0.1

PBIL アルゴリズムを以下に示す。

- Step.1 世代数を  $gen = 1$  とし、確率ベクトルの各要素を 0.5 で初期化する。
- Step.2 確率ベクトルを用いて、 $K$ 個の個体を生成する。
- Step.3 生成した個体毎に目的関数値を求め、個体の中で最良値を持つ最良個体を決定する。
- Step.4 最良個体に従って、(3.1)式にて確率行列を更新する。
- Step.5 突然変異確率に従って、(3.2)式にて確率行列を突然変異する。
- Step.6 世代数が設定した最大世代数と同じ値になった場合、全探索中の最良解を出力し終了する。それ以外の場合、世代数を  $gen = gen + 1$  とし、Step.2 へ戻る。

上述のアルゴリズムにより、確率行列によって生成した複数の解の適応度に従って確率行列を更新することで、解空間の探索を行う。PBIL は、複数の連続最適化問題に対して遺伝的アルゴリズムより優れていることを確認している<sup>(3-1)</sup>。

### 3. 1. 2 整数型 PBIL の概要

PBIL は決定変数をバイナリ型で表現するが、決定変数が整数値である最適化問題に対して適用可能とするため、1999年に H. Zhenya らによって整数型 PBIL が提案されている<sup>(3-6)</sup>。整数型 PBIL は、決定変数を整数値型で表現し、決定変数が整数値の問題に適用できる<sup>(3-6)</sup>。本研究で扱う生産順序問題を例とすると、各決定変数は処理順序として異なる整数値を持ち、表 3.3 の例では、製品 3、製品 4、製品 1、製品 2 の順序で処理することを表す。また、整数型 PBIL では、各決定変数が解となり得る整数値数毎の確率を確率行列で表現し、表 3.4 の確率行列の例では、決定変数  $x_1$  は順序 3 が選択される確率が最も高いことを示す。整数値型の決定変数に適用した整数型 PBIL の確率行列初期設定式、更新式、突然変異式、及び正規化式を以下に示す。

表 3.3 整数型 PBIL の決定変数例

$x_1$	$x_2$	$x_3$	$x_4$
3	4	1	2

表 3.4 整数型 PBIL の確率行列例

	$x_1$	$x_2$	$x_3$	$x_4$
1	0.05	0.1	0.85	0.2
2	0.1	0.1	0.1	0.6
3	0.8	0.1	0.02	0.1
4	0.05	0.7	0.03	0.1

- 確率行列初期設定式

$$P_{ij}^{gen} = 1/J \quad (i = 1, \dots, I, j = 1, \dots, J) \quad (3.3)$$

ここで,  $P_{ij}^{gen}$  : 世代数  $gen$  における決定変数  $i$  の  $j$  番目の解となり得る整数値の確率値,  
 $I$  : 決定変数の数,  $J$  : 解となり得る整数値数

- 更新式

$$P_{ij}^{gen+1} = P_{ij}^{gen} + \varepsilon \quad (i = 1, \dots, I, j = LocalBest) \quad (3.4)$$

ここで,  $\varepsilon$  : 学習率 (定数),  $LocalBest$  : 最良個体の決定変数

- 突然変異式

$$P_{ij}^{gen+1} = (1.0 - SHIFT) \times P_{ij}^{gen+1} + Random(0 \text{ or } 1) \times (SHIFT) \quad (i = 1, \dots, I, j = 1, \dots, J) \quad (3.5)$$

- 正規化式

$$P_{ij}^{gen+1} = \frac{P_{ij}^{gen+1}}{\sum_{j=1}^J P_{ij}^{gen+1}} \quad (i = 1, \dots, I, j = 1, \dots, J) \quad (3.6)$$

以下に, 整数型 PBIL のアルゴリズムを示す。

- Step.1 世代数を  $gen = 1$  とし, 確率行列の各要素を(3.3)式にて初期設定する。
- Step.2 確率行列を用いて,  $K$  個の個体を生成する。
- Step.3 生成した個体毎に目的関数値を求め, 個体の中で最良値を持つ最良個体を決定する。
- Step.4 最良個体の決定変数に従って, (3.4)式にて確率行列を更新する。
- Step.5 突然変異確率に従って, (3.5)式にて確率行列を突然変異する。
- Step.6 (3.6)式にて, 確率行列の正規化を行う。
- Step.7 世代数が設定した最大世代数と同じ値になった場合, 全探索中の最良解を出力し終了する。それ以外の場合, 世代数を  $gen = gen + 1$  とし, Step.2 へ戻る。

### 3. 1. 3 生産計画最適化問題に対する整数型 PBIL を用いた解法アルゴリズム

本研究で扱う生産計画最適化問題は, 製品機種 of 処理順序を決定変数としており, 決定変数は 1 から生産機種数までの異なる整数値の組合せとなる。従って, 一度使用した整数値は使用できないようにする必要がある。そこで, 生産ライン  $z$  の決定変数における未使用整

数値リスト( $L_{uuz}$ )と生産ライン $z$ の決定変数における使用済み整数値リスト( $L_{uz}$ )を利用する。決定変数として既に選択された整数値は $L_{uz}$ に追加され、 $L_{uuz}$ から削除される。そして、 $L_{uuz}$ の中から次の決定変数が選択される。

生産計画最適化問題に対して整数型 PBIL を適用したアルゴリズムを以下に示す。

Step.1 世代数を  $gen = 1$  とする。

Step.2 ライン番号を  $l = 1$ ,  $L_{used} = \{\}$ ,  $L_{unused} = \{1, \dots, N_l\}$  とする。ここで、 $N_l$  はライン  $l$  の製品機種数である。

Step.3 確率行列の各要素を(3.7)式にて初期設定する。

$$P_{ijl}^{gen} = \frac{1}{N_l} (i = 1, \dots, N_l, j = 1, \dots, N_l, l = 1, \dots, L) \quad (3.7)$$

ここで、 $P_{ijl}^{gen}$  : 世代数  $gen$  におけるライン  $l$  の製品  $i$  の生産順序が  $j$  番目になり得る整数値の確率値、 $L$  : ライン総数

Step.4 個体番号を  $k = 1$  とする。

Step.5 製品番号を  $i = 1$  とする。

Step.6 (3.8)式にて、確率行列の正規化を行う。

$$P_{ijl}^{gen} = \frac{P_{ijl}^{gen}}{\sum_{j=1}^{N_l} P_{ijl}^{gen}} (i = 1, \dots, N_l, j \in L_{unused}, l = 1, \dots, L) \quad (3.8)$$

Step.7 正規化された確率行列  $P_{ijl}^{gen} (j \in L_{unused})$  を用いて、ライン  $l$  の製品  $i$  の生産順序  $j$  を決定する。

Step.8 Step.7 で決定したライン  $l$  の製品  $i$  の生産順序  $j$  を  $L_{used}$  に保存する。

Step.9 製品番号が  $i = N_l$  であれば Step.10 へ、そうでなければ、製品番号を  $i = i + 1$  とし Step.6 へ戻る。

Step.10 個体番号が  $k = K$  であれば Step.11 へ、そうでなければ、個体番号を  $k = k + 1$  とし Step.5 へ戻る。

Step.11 Step.4~10 で求めた生産順序を用いて各個体の目的関数値を求め、最良値を示す個体を最良個体として選択する。

Step.12 Step.11 で選択した最良個体の決定変数に従って、(3.9)式にて確率行列を更新する。

$$P_{ijl}^{gen+1} = P_{ijl}^{gen} + \varepsilon (i = 1, \dots, I, j = LocalBest, l = 1, \dots, L) \quad (3.9)$$

Step.13 突然変異確率に従って、(3.10)式にて確率行列を突然変異する。

$$P_{ijl}^{gen+1} = (1.0 - SHIFT) \times P_{ijl}^{gen+1} + Random(0 \text{ or } 1) \times (SHIFT) \\ (i = 1, \dots, I, j = 1, \dots, J, l = 1, \dots, L) \quad (3.10)$$

Step.14 (3.11)式にて、確率行列の正規化を行う。

$$P_{ijl}^{gen+1} = \frac{P_{ijl}^{gen+1}}{\sum_{j=1}^{N_l} P_{ijl}^{gen+1}} (i = 1, \dots, N_l, j = 1, \dots, N_l, l = 1, \dots, L) \quad (3.11)$$

Step.15 ライン番号が $l = L$ であれば、Step.16 へ、そうでなければ、ライン番号を $l = l + 1$ として Step.3 へ戻る。

Step.16 世代数が設定した最大世代数と同じ値になった場合、全探索中の最良生産順序を出力し終了する。それ以外の場合、世代数を $gen = gen + 1$ とし、Step.2 へ戻る。

### 3. 1. 4 シミュレーション

#### ① シミュレーション条件

シミュレーションでは、実際の組立加工系工場の一部工程を対象としてモデル化を行った生産計画最適化問題を用いて検証を行う。具体的には、同時に同品種の製品を製造することができる独立した 2 ラインの生産ラインそれぞれに対し、5 種類の製品における製品機種の生産順序(10 変数)を対象として、Tabu Search (以下、従来手法 3-1) (3-7)と整数型 PBIL を適用し、解の質について比較検証した。

共通パラメータ、従来手法 3-1、及び整数型 PBIL のパラメータを下記する。尚、従来手法 3-1 と整数型 PBIL の評価回数は同一としている。

##### (1) 共通パラメータ

試行回数：30 回，最大世代数：30 回，ライン数：2 ライン，生産オーダー数：5 オーダー

##### (2) 従来手法 3-1 パラメータ

近傍数：50，タブー長：10

##### (3) 整数型 PBIL パラメータ

サンプル数(K)：50，学習率( $\epsilon$ )：0.06，突然変異確率( $MUT\_Prob$ )：0.02，突然変異量( $SHIFT$ )：0.02

#### ② シミュレーション結果

表 3.5 に、整数型 PBIL と従来手法 3-1 の試行回数 30 回あたりの目的関数の平均値と標準偏差値の比較結果と、Mann-Whitney の U 検定の p 値を示す。シミュレーション結果の正規性は、D'Agostino and Pearson 検定及び Anderson and Darling 検定により認められず、従来手法 3-1 と整数型 PBIL では初期解が異なり対応関係が無いいため、Mann-Whitney の U 検定を適用した。表 3.5 より、整数型 PBIL は従来手法 3-1 と比較し、平均値と標準

表 3.5 整数型 PBIL と従来手法 3-1 による目的関数値比較

手法	平均値	標準偏差	p 値
整数型 PBIL	<b>0.2285</b>	<b>0.0050</b>	0.0097
従来手法 3-1	0.2461	0.0344	

偏差値が共に優れていることを確認した。また、 $p$  値が 0.0097 であることから、有意水準 5% で有意でないとはいえないことを確認した。つまり、整数型 PBIL の有効性を確認した。

図 3.1 に、試行回数 30 回における整数型 PBIL と従来手法 3-1 の箱ひげ図による目的関数値評価結果を示す。図 3.2 より、従来手法 3-1 を適用した場合、目的関数値にばらつきがあり、安定して質の高い解を導出できていないことを確認した。一方、整数型 PBIL は、30 回試行のうち 1 回を除いてすべての試行で 2 手法間での最良解を導出できることを確認した。これにより、整数型 PBIL の有効性を確認した。

図 3.2 に、整数型 PBIL と従来手法 3-1 の世代数毎の目的関数平均値の収束特性を示す。図 3.2 より、従来手法 3-1 は世代数 10 回あたりまで目的関数値の改善が見られるものの、11 回以降は局所解に収束する傾向にあることを確認した。一方、整数型 PBIL は世代数 13



図 3.1 整数型 PBIL と従来手法 3-1 の箱ひげ図による目的関数値評価

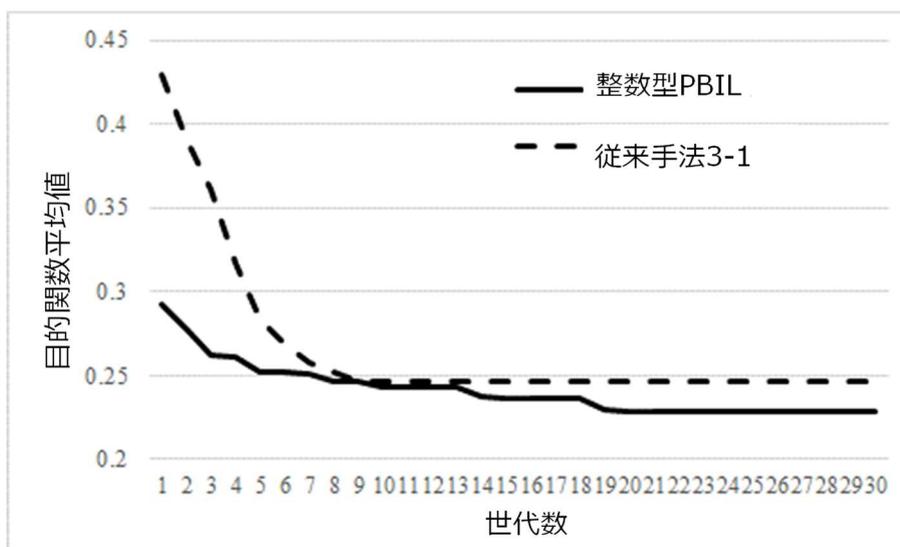


図 3.2 整数型 PBIL と従来手法 3-1 の世代数毎の目的関数平均値の収束特性

回以降も改善を続け、従来手法 3-1 より優れた解を取得していることを確認した。これらの結果の考察として、従来手法 3-1 は、近傍探索手法であるため、局所的な探索領域のみしか探索できなかったのに対し、整数型 PBIL は確率行列を用いて決定変数を決定するため、局所的な探索領域だけでなく大域的な探索領域を探索できたためであるといえる。更に、確率行列の更新処理により探索の集中化を強めつつ、確率行列要素の突然変異処理により確率的に探索の多様化を強めるアルゴリズムにより、効率的な探索が出来たと考えられる。

### **3. 1. 5 整数型 PBIL の有効性と課題**

本節では、生産コストシミュレータと整数型 PBIL を用いて、製品毎の生産順序を最適化する最適生産順序導出手法の有効性と課題を確認した。

シミュレーション検証では、実際の組立加工系工場の一部工程をモデル化した生産計画最適化問題に対し、従来手法 3-1 と整数型 PBIL を適用し、整数型 PBIL の有効性を確認した。しかし、整数型 PBIL の性能は、(3.9)式に示す整数型 PBIL が決定変数の決定に用いる確率行列の更新パラメータである学習率 $\epsilon$ に大きく依存するため、学習率 $\epsilon$ を変えると解が変化してしまう。つまり、実際の工場現場で運用することを考えると、対象となる生産計画最適化問題が変わる度に、事前に最適な学習率 $\epsilon$ を調整する必要がある。以上より、整数型 PBIL は、学習率 $\epsilon$ の調整にかかる工数の削減が課題であるといえる。

## 3. 2 生産シミュレータを用いた生産計画最適化問題に対する整数型 APBIL の適用の提案

### 3. 2. 1 整数型 APBIL の提案

整数型 PBIL は、分布推定アルゴリズムの考えに基づき、確率行列により確率的に決定変数を決定する PBIL を、組合せ最適化問題に適用できるよう改良した手法である。また、整数型 PBIL は、確率行列の更新パラメータである学習率 $\epsilon$ が固定値として設定される。そのため、学習率 $\epsilon$ の値に応じて解の質が変動する。これは、実際の工場現場での適用を考えると、対象とする生産計画に応じて事前パラメータ調整工数が掛かるといえる。つまり、実工場適用に適した手法とはいえない。

本節では、整数型 PBIL の課題である学習率 $\epsilon$ の調整工数削減という課題を解決するため、整数型 APBIL<sup>(3-8)</sup>を提案する。

### 3. 2. 2 適応関数の適用

バイナリ型の決定変数を扱う PBIL は、整数型 PBIL と同様、2章で示す(3.9)式の学習率 $\epsilon$ の値が固定である。学習率 $\epsilon$ は、一般的に値が小さいと全体探索の傾向が強くなり収束プロセスが遅くなる。一方、値が大きいと局所探索の傾向が強くなり高品質の解を見つけることが難しくなる。従って、アルゴリズムが効率的に探索するためには事前に最適な学習率 $\epsilon$ の値を導出する必要がある。一方、APBIL<sup>(3-9)</sup>は、エントロピーを使用して探索の進化度合いを表現し、進化度合いに応じて世代ごとの学習率 $\epsilon^{gen}$ を自動的に調整する適応関数を採用する。これにより、初期学習率 $\epsilon_0$ を様々な値で設定した場合でも解の質を保つ効果が期待できる。(3.12)式と(3.13)(3.14)式に、エントロピーと世代ごとの学習率 $\epsilon^{gen}$ の算出式を示す。(3.12)式で導出されるエントロピー $E$ は、世代数ごとに確率行列 $P_{ij}$ の値を用いて計算され、探索が進むにつれて値が減少する。(3.13)(3.14)式で導出される世代ごとの学習率 $\epsilon^{gen}$ は、多様性を確保するために、探索の初期は世代ごとの学習率 $\epsilon^{gen}$ を小さくする。探索が進むにつれて進化の加速を促すため、世代数の中間付近では世代ごとの学習率 $\epsilon^{gen}$ を大きくし、世代数の後期では世代ごとの学習率 $\epsilon^{gen}$ を中間付近よりわずかに小さくする。このように、整数型 APBIL は、探索期間に応じて世代ごとの学習率 $\epsilon^{gen}$ を変化させる戦略（以下、APBIL 探索戦略）を取る。

$$E^{gen} = -\sum_{i=1}^I \sum_{j=1}^J (P_{ij}^{gen} \times \ln P_{ij}^{gen}) \quad (3.12)$$

ここで、 $E^{gen}$ : 世代数 $gen$ におけるエントロピー

$$\varepsilon^{gen} = \begin{cases} \frac{\varepsilon_0}{1+k_1 \times (1-E^{gen}/E_{th})^2} & (E^{gen} \leq E_{th}) \\ \frac{\varepsilon_0}{1+k_2 \times (1-E^{gen}/E_{th})^2} & (E^{gen} > E_{th}) \end{cases} \quad (3.13)$$

$$E_{th} = \alpha \times E_0. \quad (3.14)$$

ここで、 $\varepsilon^{gen}$ ：世代数 $gen$ における学習率、 $\varepsilon_0$ ：初期学習率、 $k_1, k_2, \alpha$ ：定数、 $E_{th}$ ：エントロピー閾値、 $E_0$ ：初期確率ベクトルのエントロピー値

本節では、3.1節で説明した整数型PBILに対して、APBILが持つ学習率 $\varepsilon^{gen}$ を自動的に調整する機能を追加した整数型APBILを提案する。以下に、提案する整数型APBILのアルゴリズムを示す。

- Step.1 (3.3)式を用いて初期確率行列 $P_{ij}^{gen}$ を設定する。世代数 $gen = 1$ とする。
- Step.2  $P_{ij}^{gen}$ を用いて、 $K$ 個の決定変数の組合せを生成する。
- Step.3 (3.12)式を用いて、 $E$ を算出する。
- Step.4 (3.13)(3.14)式を用いて、学習率 $\varepsilon^{gen}$ を更新する。
- Step.5 生成した決定変数の組合せを用いて目的関数を計算し、目的関数値が最も良い解 $B$ と(3.4)式を用いて、 $P_{ij}^{gen}$ を更新する。
- Step.6 突然変異確率 $MUT\_Prob$ に従って、(3.5)式を用いて $P_{ij}^{gen}$ に突然変異を起こす。
- Step.7 (3.5)式を用いて、 $P_{ij}^{gen}$ の正規化を行う。
- Step.8 世代数 $gen$ が事前に設定した最大世代数( $maxgen$ )に達したら、これまで得られた解の中で最も良い解を出力し、終了する。そうでなければ、 $gen = gen + 1$ とし、Step.2へ戻る。

### 3.2.3 生産計画最適化問題に対する整数型APBILを用いた解法アルゴリズム

本研究で扱う生産計画最適化問題は、製品機種の処理順序を決定変数としており、決定変数は1から生産機種数までの異なる整数値の組合せとなる。従って、3.1.3節と同様、使用した整数値を記録し、未使用の整数値を決定変数として選択可能なアルゴリズムを追加した。

生産計画最適化問題に対する整数型APBILを用いた解法アルゴリズムを以下に示す。

- Step.1 世代数を $gen = 1$ とする。
- Step.2 ライン番号を $l = 1$ 、 $L_{used} = \{\}$ 、 $L_{unused} = \{1, \dots, N_l\}$ とする。ここで、 $N_l$ はライン $l$ の製品機種数である。
- Step.3 確率行列の各要素を(3.7)式にて初期設定する。

- Step.4 個体番号を $k = 1$ とする。
- Step.5 製品番号を $i = 1$ とする。
- Step.6 (3.8)式にて、確率行列 $P_{ijl}^{gen}$ の正規化を行う。
- Step.7 正規化された確率行列 $P_{ijl}^{gen}$  ( $j \in L_{unused}$ )を用いて、ライン $l$ の製品 $i$ の生産順序 $j$ を決定する。
- Step.8 Step.7で決定したライン $l$ の製品 $i$ の生産順序 $j$ を $L_{used}$ に保存する。
- Step.9 製品番号が $i = N_l$ であれば Step.10 へ、そうでなければ、製品番号を $i = i + 1$ として Step.6 へ戻る。
- Step.10 個体番号が $k = K$ であれば Step.11 へ、そうでなければ、個体番号を $k = k + 1$ として Step.5 へ戻る。
- Step.11 (3.12)(3.13)(3.14)式にて、世代ごとの学習率 $\varepsilon^{gen}$ を更新する。
- Step.12 Step.4~10 で求めた生産順序を用いて各個体の目的関数値を求め、最良値を示す個体を最良個体として選択する。
- Step.13 Step.11 で選択した最良個体の決定変数に従って、(3.9)式にて確率行列 $P_{ijl}^{gen}$ を更新する。
- Step.14 突然変異確率に従って、(3.10)式にて確率行列 $P_{ijl}^{gen}$ を突然変異する。
- Step.15 (3.11)式にて、確率行列 $P_{ijl}^{gen}$ の正規化を行う。
- Step.16 ライン番号が $l = L$ であれば、Step.17 へ、そうでなければ、ライン番号を $l = l + 1$ として Step.3 へ戻る。
- Step.17 世代数が設定した最大世代数と同じ値になった場合、全探索中の最良生産順序を出力し終了する。それ以外の場合、世代数を $gen = gen + 1$ とし、Step.2 へ戻る。

### **3. 2. 4 シミュレーション**

本節では、整数型 PBIL<sup>(3-6)</sup>と整数型 APBIL<sup>(3-8)</sup> (提案手法 3-1) を組立加工系工場の一部工程を対象としてモデル化した生産計画最適化問題に適用し、目的関数値の比較を行った。

#### **① シミュレーション条件**

本論文では、2ラインにおける6種類の製品における製品機種の処理順序(12変数)を対象として、従来手法と提案手法を用いて生産計画の最適化を実施し、出力となる生産コスト指標を比較する。

共通パラメータ、整数型 PBIL、及び提案手法 3-1 のパラメータを下記する。本節において、提案手法 3-1 パラメータの $\varepsilon$ 値は、定数の学習率 $\varepsilon$ を表し、提案手法 3-1 の $\varepsilon$ 値は、初期学習率 $\varepsilon_0$ を表す。尚、整数型 PBIL と提案手法 3-1 の評価回数は同一としている。

(1) 共通パラメータ

試行回数：30回，最大世代数 $maxgen$ ：30回，ライン数：2ライン，製品数：6製品，サンプル数 $K$ ：30，突然変異確率 $MUT\_Prob$ ：0.02，突然変異量 $SHIFT$ ：0.02

(2) 整数型 PBIL パラメータ

学習率 $\varepsilon$ ：0.1~0.5(0.1刻み)

(3) 提案手法 3-1 パラメータ

初期学習率 $\varepsilon_0$ ：0.1~0.5(0.1刻み)， $k_1$ ：5.0， $k_2$ ：0.5， $\alpha$ ：0.8

実行環境は，C言語(Microsoft Visual studio 2019 Visual C++)，計算機 (Windows 10 home, Intel® Core™i9-10980XE CPU @3.0GHz) である。

② シミュレーション結果

表 3.6 に，提案手法 3-1 と整数型 PBIL の試行回数 30 回あたりの $\varepsilon$ 値を 0.1 から 0.5 まで変化させたときの目的関数の平均値と標準偏差値の比較結果と，各 $\varepsilon$ 値における 2 手法に対する Wilcoxon の符号検定の p 値を示す。シミュレーション結果の正規性は，D'Agostino and Pearson 検定及び Anderson and Darling 検定により認められず，提案手法 3-1 と整数型 PBIL では初期解が同じとなるので，Wilcoxon の符号検定を適用した。尚， $\varepsilon$ 値は，整数型 PBIL の場合定数を示し，提案手法 3-1 の場合初期値を示す。

表 3.6 より，提案手法 3-1 を用いた提案手法は，整数型 PBIL を用いた方法と比較し，全ての $\varepsilon$ 値において，提案手法 3-1 が優れた平均値を算出することを確認した。特に， $\varepsilon$ 値が 0.1,0.2 の場合，提案手法 3-1 の平均値は整数型 PBIL と比べ半分以下となり，より質の高

表 3.6 提案手法 3-1 と整数型 PBIL による目的関数値比較

$\varepsilon$ 値	手法	平均値	標準偏差	p 値
0.1	提案手法 3-1	<b>0.125</b>	$2.70 \times 10^{-6}$	$1.44 \times 10^{-7}$
	整数型 PBIL	0.239	<b><math>1.11 \times 10^{-16}</math></b>	
0.2	提案手法 3-1	<b>0.125</b>	<b><math>1.11 \times 10^{-16}</math></b>	$4.32 \times 10^{-8}$
	整数型 PBIL	0.239	<b><math>1.11 \times 10^{-16}</math></b>	
0.3	提案手法 3-1	<b>0.239</b>	<b><math>1.11 \times 10^{-16}</math></b>	$1.96 \times 10^{-7}$
	整数型 PBIL	0.242	0.008	
0.4	提案手法 3-1	<b>0.239</b>	<b><math>1.11 \times 10^{-16}</math></b>	$5.53 \times 10^{-7}$
	整数型 PBIL	0.242	0.006	
0.5	提案手法 3-1	<b>0.239</b>	<b><math>1.11 \times 10^{-16}</math></b>	$1.44 \times 10^{-7}$
	整数型 PBIL	0.240	0.004	

い解を導出することを確認した。標準偏差値の観点では、 $\epsilon$  値が 0.2~0.5 において、提案手法 3-1 が整数型 PBIL と比べ同等、または小さい値を導出することを確認した。また、 $p$  値が全ての $\epsilon$ 値で 0.05 以下であることから、有意水準 5%で有意でないとはいえないことを確認した。

図 3.3 に、試行回数 30 回における  $\epsilon$  値を変化させたときの提案手法 3-1 と整数型 PBIL の箱ひげ図による目的関数値評価結果を示す。図 3.3 より、整数型 PBIL を適用した場合、 $\epsilon$  値が 0.1 と 0.2 のとき両手法間での最良解を試行回数すべてで導出することができることを確認した。しかし、 $\epsilon$  値が 0.3 以上のとき、両手法間での最悪解を試行回数すべてで導出することを確認した。一方、提案手法である提案手法 3-1 を適用した場合、 $\epsilon$  値が 0.1 と 0.2 のとき同様に両手法間での最良解を試行回数すべてで導出し、更に  $\epsilon$  値が 0.3 以上の時も試行回数の半数以上で 2 手法間での最良解を導出できることを確認した。つまり、整数型 PBIL を実工場の生産計画に適用した場合、 $\epsilon$  値を一定値以上に設定すると質の悪い解を導出する可能性があることを示す。一方で、提案手法 3-1 を実工場の生産計画に適用した場合、 $\epsilon$  値を一定値以上に設定してもある程度質の高い解を導出できる可能性があることを示す。これにより、提案手法 3-1 の有効性を確認した。

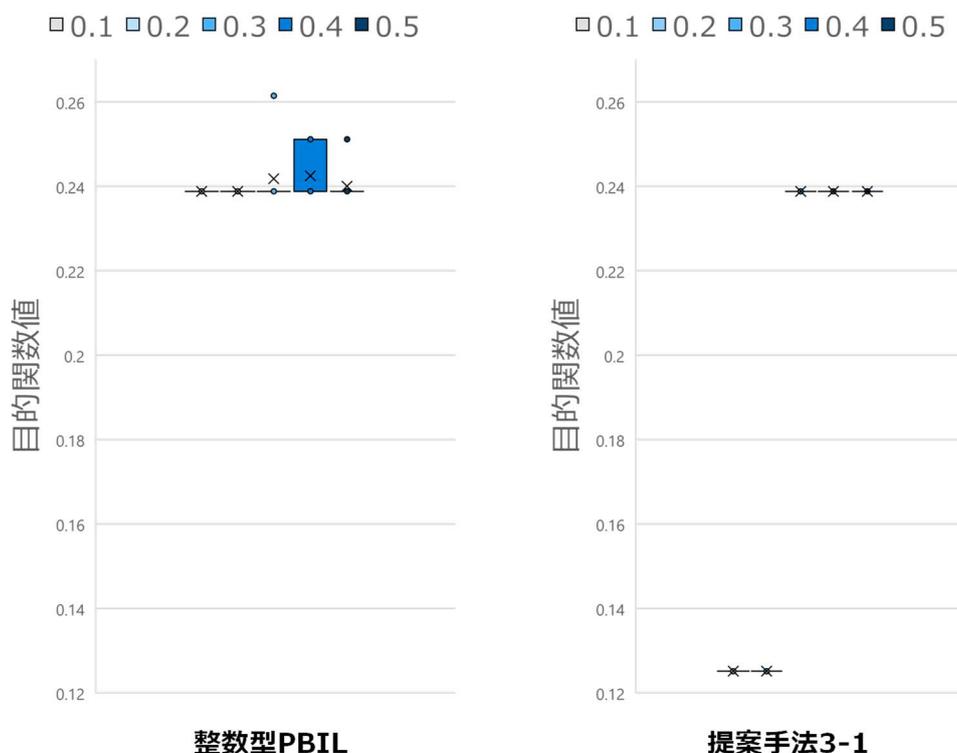


図 3.3 提案手法 3-1 と整数型 PBIL の箱ひげ図による目的関数値比較

### ③ シミュレーション結果に対する考察

$\epsilon$  値が 0.3 以上の時に、提案手法 3-1 と整数型 PBIL で平均値に大きな差異が出なかった原因として、提案手法 3-1 で用いた適応関数の性質上、初期学習率 $\epsilon_0$ が一定の数値以上になった場合、想定通りの APBIL 探索戦略を再現できないことが考えられた。そこで、初期学習率 $\epsilon_0$ を 0.1 から 0.5 まで 0.1 刻みで変えたときの学習率 $\epsilon^{gen}$ の推移について確認した。その結果を図 3.4 に示す。ここで、APBIL 探索戦略が再現できるという定義は、世代数 1 から探索の後半(以下、最大世代数の 50%~90%)まで学習率 $\epsilon^{gen}$ が正の傾きを取り、最大世代数の 50%~90%の間で一度学習率 $\epsilon^{gen}$ を負の傾きにした場合、最大世代数まで学習率 $\epsilon^{gen}$ が負の傾きを取り続けることとする。図 3.4 では、最大世代数の 10%刻みでこの定義に基づいた学習率 $\epsilon^{gen}$ の正の傾き及び負の傾きが実現されているかを矢印で示す。図 3.4 中、正の傾き及び負の傾きが実現されている場合、青の矢印とし、実現されていない場合、赤の矢印としている。図 3.4 より、提案手法 3-1 では、初期学習率 $\epsilon_0$ が 0.1 のときは APBIL 探索戦略を再現できているが、0.3 以上のときから APBIL 探索戦略を全く再現できておらず、0.1 の時と比べ解の質が悪くなることを確認した<sup>(3-8)</sup>。つまり、初期学習率 $\epsilon_0$ を変化させたときの学習率推移の確認により、初期学習率 $\epsilon_0$ が一定の数値以上になった場合、想定通りの APBIL 探索戦略を再現できないことを確認した。今回の問題の規模では、初期学習率 $\epsilon_0$ を 0~0.2 の間で設定することで、より提案手法の効果が発揮されるといえる。異なる規模の問題に対して、様々な初期学習率 $\epsilon_0$ を与えても質の高い解を導出可能とするためには、初期学習率 $\epsilon_0$ を探索に応じて変化させる適応関数の改良が必要である。

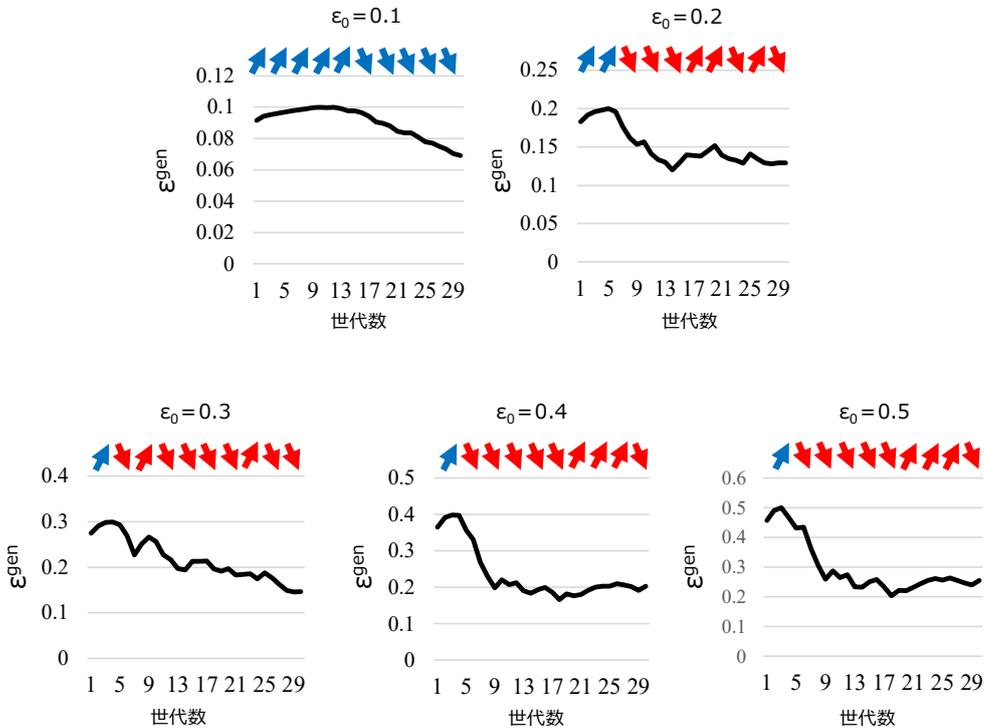


図 3.4 初期学習率 $\epsilon_0$ を変化させたときの提案手法 3-1 による学習率推移

### **3. 2. 5 整数型 APBIL の有効性と課題**

本節では、生産コストシミュレータと提案手法 3-1 を用いて、製品毎の生産順序を最適化する最適生産順序導出手法を提案した。シミュレーション検証では、実際の組立加工系工場の一部工程をモデル化した生産計画最適化問題に対し、整数型 PBIL と提案手法 3-1 を適用し、提案手法 3-1 の有効性を確認した。

しかし、提案手法 3-1 は、初期学習率 $\epsilon_0$ がある一定以上の値になると解の質が落ちることを確認した。解の質が悪いときの学習率推移を確認した結果、想定通りの APBIL 探索戦略を再現できないことを確認した。再現できない原因を調査した結果、提案手法 3-1 の適応関数が、特定の初期学習率 $\epsilon_0$ に対しては APBIL 探索戦略に沿った学習率推移を取る一方で、それ以外の初期学習率 $\epsilon_0$ に対しては APBIL 探索戦略に沿わない学習率推移を取るためであることを確認した。つまり、エンジニアリング業務を削減するために、初期学習率 $\epsilon_0$ を変えても質の高い解を導出可能とするには、初期学習率 $\epsilon_0$ によらず APBIL 探索戦略を再現することができる適応関数の改良が必要である。

### 3.3 改良需給連携最適化方式の主問題に対する改良整数型 APBIL の適用の提案

#### 3.3.1 改良整数型 APBIL の提案

整数型 APBIL は、整数型 PBIL に対して APBIL が持つ適応関数を適用し、探索に応じて学習率 $\varepsilon^{gen}$ を更新することができるように改良した手法である。3.2 節のシミュレーション検証により、整数型 APBIL は、初期学習率 $\varepsilon_0$ の値により解の質が変動し、解の質が悪いとき学習率推移を調べた結果、探索の初期は学習率 $\varepsilon^{gen}$ を小さくし、探索が進むにつれて学習率 $\varepsilon^{gen}$ を大きくし、探索の後期で学習率 $\varepsilon^{gen}$ を中間付近よりわずかに小さくする APBIL 探索戦略が再現できていないことを確認した。つまり、エンジニアリング業務を削減するために、初期学習率 $\varepsilon_0$ を変えても質の高い解を導出可能とするには、初期学習率 $\varepsilon_0$ によらず APBIL 探索戦略を再現することができる適応関数の改良が必要である。

本節では、整数型 APBIL の課題である解の質の向上と初期学習率 $\varepsilon_0$ の調整工数削減を達成するため、改良整数型 APBIL<sup>(3-10)</sup> (提案手法 3-2) を提案する。

#### 3.3.2 改良適応関数の適用

本節で提案する改良整数型 APBIL は、3.4 節で説明した、整数型 APBIL の初期学習率 $\varepsilon_0$ によって APBIL 探索戦略が再現できず、解の質がばらつく問題を解決し、学習率更新式を改良した手法である。提案する改良整数型 APBIL の学習率更新式を以下に示す。

$$\varepsilon^{gen} = \begin{cases} \frac{\log(1/\varepsilon_0) \times N_{col}}{(maxgen)^2} \times gen + \varepsilon_0 (gen < gen_{th}) \\ - \frac{\log(1/\varepsilon_0) \times N_{col}}{2 \times (maxgen)^2} \times (gen - gen_{th}) \\ + \frac{\log(1/\varepsilon_0) \times N_{col}}{(maxgen)^2} \times gen_{th} + \varepsilon_0 (gen \geq gen_{th}) \end{cases} \quad (3.15)$$

$$gen_{th} = Roundup(maxgen \times \beta) \quad (3.16)$$

ここで、 $N_{col}$ : 確率行列の列要素数(決定変数の数)、 $gen_{th}$ : 学習率 $\varepsilon^{gen}$ を減少させる  $gen$  閾値、 $\beta$ : 定数、 $Roundup()$ : 小数点切り上げ関数

尚、文献(3-8~3-10)より、初期学習率 $\varepsilon_0$ の設定範囲は、 $0 < \varepsilon_0 \leq 0.5$ とする。(3.15)式は、世代数が閾値 $gen_{th}$ 未満の時は学習率 $\varepsilon^{gen}$ が増加し、閾値 $gen_{th}$ 以上の時は学習率 $\varepsilon^{gen}$ が増加時の傾きの 1/2 分減少する。また、(3.15)式は、初期学習率 $\varepsilon_0$ を切片としている。図 3.5 に、提案する改良整数型 APBIL の(3.15)式で表される学習率推移の例を示す。(3.15)式の傾きとした理由は、初期学習率 $\varepsilon_0$ や最大世代数の大小に関わらず、学習率 $\varepsilon^{gen}$ の最大値(図 3.5 の例

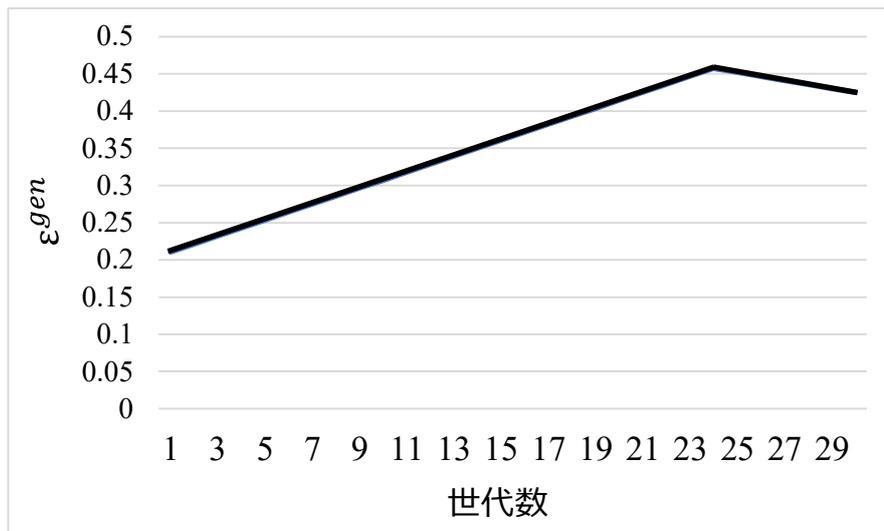


図 3.5 改良整数型 APBIL による学習率推移の例 ( $\beta = 0.8$ )

では約 0.47)を一定範囲内にすることで、解のばらつきを抑えられるという仮説を立てたためである。例として初期学習率 $\epsilon_0$ について説明する。(3.15)式は、初期学習率 $\epsilon_0$ が切片であることから、初期学習率 $\epsilon_0$ が大きい場合、世代数が 1 のときの学習率 $\epsilon^{gen}$ は既に大きいため、(3.15)式により傾きを小さくする。一方、初期学習率 $\epsilon_0$ が小さい場合、世代数が 1 のときの学習率 $\epsilon^{gen}$ は小さいため、(3.15)式により傾きを大きくする。これにより、学習率 $\epsilon^{gen}$ の最大値のばらつきを抑制している。上記仮説に基づく改良により、様々な組合せ最適化問題に対しても、APBIL の戦略に沿った学習率 $\epsilon^{gen}$ の変動を再現し、適切な探索を可能とする。

提案する改良整数型 APBIL アルゴリズムを以下に示す。

- Step.1 世代数を  $gen = 1$  とし、確率行列の各要素を(3.3)式にて初期設定する。
- Step.2 確率行列を用いて、 $K$ 個の個体を生成する。
- Step.3 (3.15)(3.16)式にて、学習率 $\epsilon^{gen}$ を更新する。
- Step.4 生成した個体毎に目的関数値を求め、個体の中で最良値を持つ最良個体を決定する。
- Step.5 Step.3 で更新した学習率 $\epsilon^{gen}$ と Step.4 で決定した最良個体の決定変数に従って、(3.4)式にて確率行列を更新する。
- Step.6 突然変異確率に従って、(3.5)式にて確率行列を突然変異する。
- Step.7 (3.6)式にて、確率行列の正規化を行う。
- Step.8 世代数が設定した最大世代数と同じ値になった場合、全探索中の最良解を出力し終了する。それ以外の場合、世代数を  $gen = gen + 1$  とし、Step.2 へ戻る。

### 3. 3. 3 主問題に対する改良整数型 APBIL を用いた解法アルゴリズム

本研究で扱う改良需給連携最適化方式の主問題は、製品機種の処理順序を決定変数としており、決定変数は 1 から生産機種数までの異なる整数値の組合せとなる。従って、3.1.3 節と同様、使用した整数値を記録し、未使用の整数値を決定変数として選択可能なアルゴリズムを追加した。

改良需給連携最適化方式の主問題に提案する改良整数型 APBIL を適用した際のアルゴリズムを以下に示す。

- Step.1 世代数を  $gen = 1$  とする。
- Step.2 ライン番号を  $l = 1$ ,  $L_{used} = \{\}$ ,  $L_{unused} = \{1, \dots, N_l\}$  とする。ここで、 $N_l$  はライン  $l$  の製品機種数である。
- Step.3 確率行列の各要素を(3.7)式にて初期設定する。
- Step.4 個体番号を  $k = 1$  とする。
- Step.5 製品番号を  $i = 1$  とする。
- Step.6 (3.8)式にて、確率行列の正規化を行う。
- Step.7 正規化された確率行列  $P_{ijl}^{gen} (j \in L_{unused})$  を用いて、ライン  $l$  の製品  $i$  の生産順序  $j$  を決定する。
- Step.8 Step.7 で決定したライン  $l$  の製品  $i$  の生産順序  $j$  を  $L_{used}$  に保存する。
- Step.9 製品番号が  $i = N_l$  であれば Step.10 へ、そうでなければ、製品番号を  $i = i + 1$  として Step.6 へ戻る。
- Step.10 個体番号が  $k = K$  であれば Step.11 へ、そうでなければ、個体番号を  $k = k + 1$  として Step.5 へ戻る。
- Step.11 (3.15)(3.16)式にて、学習率  $\epsilon^{gen}$  を更新する。
- Step.12 Step.4~10 で求めた生産順序を用いて各個体の目的関数値を求め、最良値を示す個体を最良個体として選択する。
- Step.13 Step.11 で更新した学習率  $\epsilon^{gen}$  と Step.12 で選択した最良個体の決定変数に従って、(3.9)式にて確率行列を更新する。
- Step.14 突然変異確率に従って、(3.10)式にて確率行列を突然変異する。
- Step.15 (3.11)式にて、確率行列の正規化を行う。
- Step.16 ライン番号が  $l = L$  であれば、Step.16 へ、そうでなければ、ライン番号を  $l = l + 1$  として Step.3 へ戻る。
- Step.17 世代数が設定した最大世代数と同じ値になった場合、全探索中の最良生産順序を出力し終了する。それ以外の場合、世代数を  $gen = gen + 1$  とし、Step.2 へ戻る。

### 3. 3. 4 シミュレーション

#### ① シミュレーション条件

本節では、主問題として実際の組立加工系工場の一部工程を対象としてモデル化と検証を行う。具体的には、3ラインにおける6種類の製品における製品機種の処理順序(18変数)を対象として、整数型 APBIL<sup>(3-8)</sup>(提案手法 3-1)と提案手法である改良整数型 APBIL(提案手法 3-2)を適用して最適生産計画を立案し、生産コスト指標を比較する。主問題は、 $I$ 台の生産ライン数に対し $J$ 種類の製品機種数を扱う場合、処理順序は $(J!)^I$ 通りの組合せ数になる。つまり、本問題の処理順序の組合せは、 $(6!)^3 \approx 3.7 \times 10^8$ 通りである。

以下に、共通パラメータ、提案手法 3-1、及び提案手法 3-2 のパラメータを示す。尚、提案手法 3-1 と提案手法 3-2 の評価回数は同一としている。

#### (1) 共通パラメータ

試行回数：30，最大世代数：30，個体数 $K$ ：50，突然変異確率：0.02，定数 $SHIFT$ ：0.02，初期学習率 $\varepsilon_0$ ：0.01, 0.1, 0.2, 0.3, 0.4, 0.5，初期確率行列：全て1/6

#### (2) 提案手法 3-1 パラメータ

定数 $k_1, k_2$ ：0.5，定数 $\alpha$ ：0.8

#### (3) 提案手法 3-2 パラメータ

定数 $\beta$ ：0.5, 0.6, 0.7, 0.8, 0.9

実行環境は、C 言語(Microsoft Visual studio 2019 Visual C++)，C #(Microsoft Visual studio 2019 Visual C#)，Microsoft SQL Server 2016 と以下の計算機である。

- ① Windows 11, Intel® Core™ i9-10980XE CPU @3.0GHz,
- ② Windows 11, Intel® Core™ i9-10980XE CPU @3.0GHz,
- ③ Windows Server 2016 Standard, Intel® Core™ i7-7700K CPU @4.2GHz,
- ④ Windows 10, Intel® Core™ i7-8700 CPU @3.2GHz,
- ⑤ Windows 10, Intel® Core™ i5-8500 CPU @3.0GHz,
- ⑥ Windows 10, Intel® Core™ i5-6500 CPU @3.2GHz,
- ⑦ Windows 10, Intel® Core™ i5-6500 CPU @3.2GHz

本節では、上記複数の計算機を用いて、解の質については図 2.7 全体を計算機①に実装して並列分散処理を実現し、環境負荷量計算部(図 2.7 中、環境負荷計算部)を計算機②に実装して分散処理を実現し評価した。また、並列分散処理の効果については、図 2.6 全体を計算機③に実装して並列分散処理を実現し、環境負荷量計算部(図 2.6 中、環境負荷計算部)を計算機④～⑦に実装して LAN 接続により分散処理を実現し評価した。分散処理の評価方法とし

て、計算機④～⑦を1台ずつ追加したときの全処理時間を計測した。尚、副問題は、線形計画問題として扱うことが可能なため、線形計画法ソルバーである GLPK で解く<sup>(3-11)</sup>。

## ② シミュレーション結果

提案手法 3-2 と提案手法 3-1 の初期学習率 $\epsilon_0$ を 0.01 から 0.5 まで変えたときの試行回数 30 回あたりの目的関数の平均値と標準偏差値の比較結果を表 3.7 に、箱ひげ図による目的関数値の比較結果を図 3.6 に示す。尚、太字は各手法の最良平均値を示している。表 3.7 及び図 3.6 より、提案手法 3-2 は、提案手法 3-1 と比較し、全ての初期学習率 $\epsilon_0$ で優れた値を算出することを確認した。特に、初期学習率 $\epsilon_0$ が 0.01 や 0.5 の場合、提案手法 3-1 は解のばらつきが大きく解の質が悪くなっている一方、提案手法 3-2 は解のばらつきを抑え解の質を保っており、過度なパラメータチューニングは必要ないことを確認した。つまり、初期学習率 $\epsilon_0$ に対するエンジニアリングが削減可能であることを確認した。標準偏差値の観点でも、全ての初期学習率 $\epsilon_0$ で提案手法 3-2 が優れており、提案手法 3-1 より安定して質の高い解を導出することを確認した。

また、表 3.7 の両手法の最良平均値となる初期学習率 $\epsilon_0$ が 0.1 の 2 手法に対する検定を行った。シミュレーション結果の正規性は、D'Agostino and Pearson 検定及び Anderson and Darling 検定により認められず、提案手法 3-2 と提案手法 3-1 は初期解が同じとなり、対応関係があるので、Wilcoxon の符号検定を適用した。Wilcoxon の符号検定の p 値を算出したところ、p 値が 0.00785 であり 0.05 以下であることから、提案手法 3-2 が提案手法 3-1 に

表 3.7 提案手法 3-2 と提案手法 3-1 の試行回数 30 回あたりの  
目的関数の平均値と標準偏差値の比較結果

手法	$\epsilon_0$	平均値	標準偏差
提案手法 3-2	0.01	4358.756	20.480
	0.1	<b>4352.899</b>	<b>2.157</b>
	0.2	4355.317	15.178
	0.3	4366.386	28.981
	0.4	4364.974	29.661
	0.5	4368.428	32.127
従来手法 3-1	0.01	4382.757	20.210
	0.1	<b>4356.518</b>	<b>15.382</b>
	0.2	4365.374	29.571
	0.3	4380.050	37.976
	0.4	4368.828	38.791
	0.5	4394.948	45.250

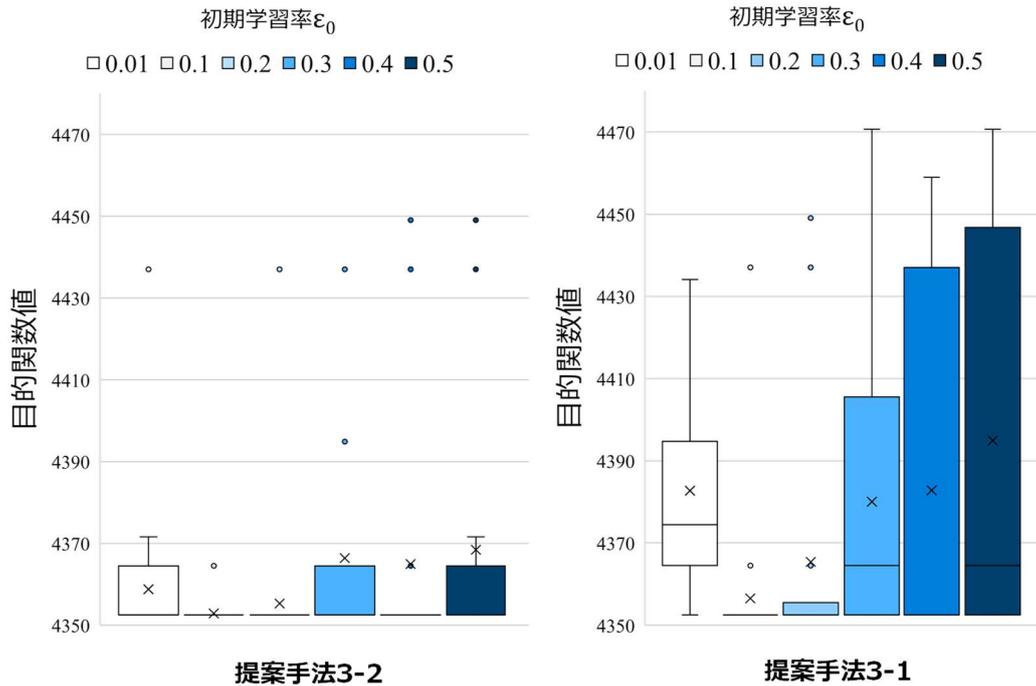


図 3.6 箱ひげ図による提案手法 3-2 と提案手法 3-1 の試行回数 30 回あたりの目的関数値の比較

対し、有意水準 5% で有意でないとは言えないことを確認した。以上により、提案手法 3-2 の有効性を確認した。

図 3.7 に、提案手法 3-2 と提案手法 3-1 の初期学習率  $\epsilon_0$  を 0.01 から 0.5 まで変えたときの学習率推移結果を示す。図 3.7 は、図 3.4 と同様、最大世代数の 10% 刻みで APBIL 探索戦略に基づいた学習率  $\epsilon^{gen}$  の正の傾き及び負の傾きが実現されているかを青矢印(再現可)または赤矢印(再現不可)で示す。図 3.7 の提案手法 3-1 の学習率推移に着目すると、他の初期学習率  $\epsilon_0$  に比べ解の質が良い  $\epsilon_0 = 0.1, 0.2$  のときは、APBIL 探索戦略を再現できていた割合がそれぞれ 90%, 80% である。しかし、それ以外の初期学習率  $\epsilon_0$  のときは、APBIL 探索戦

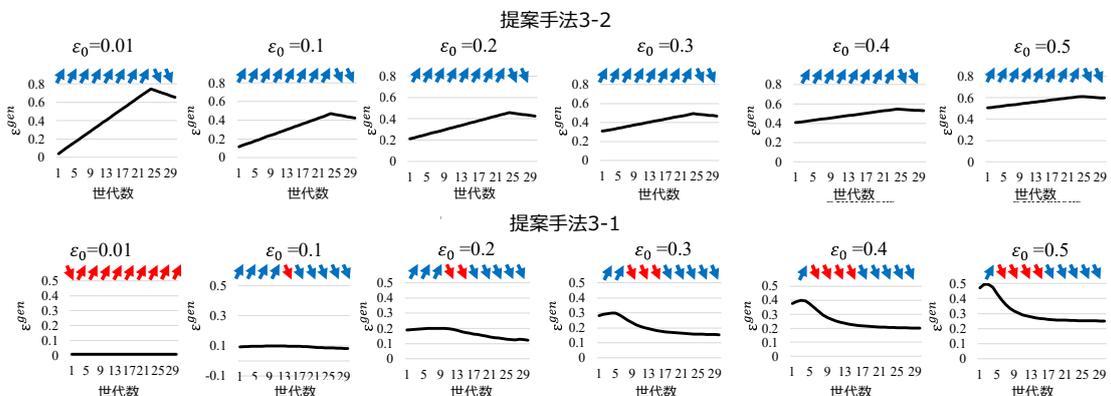
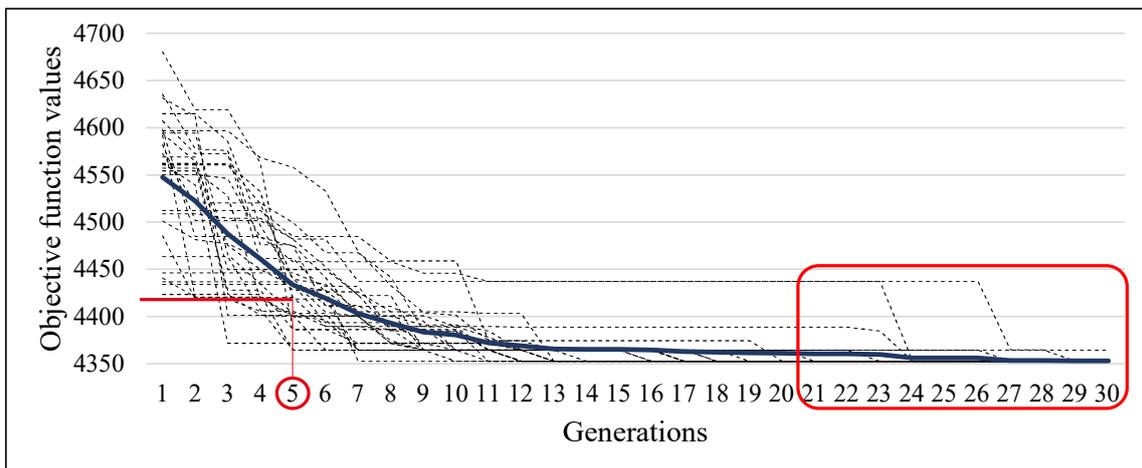


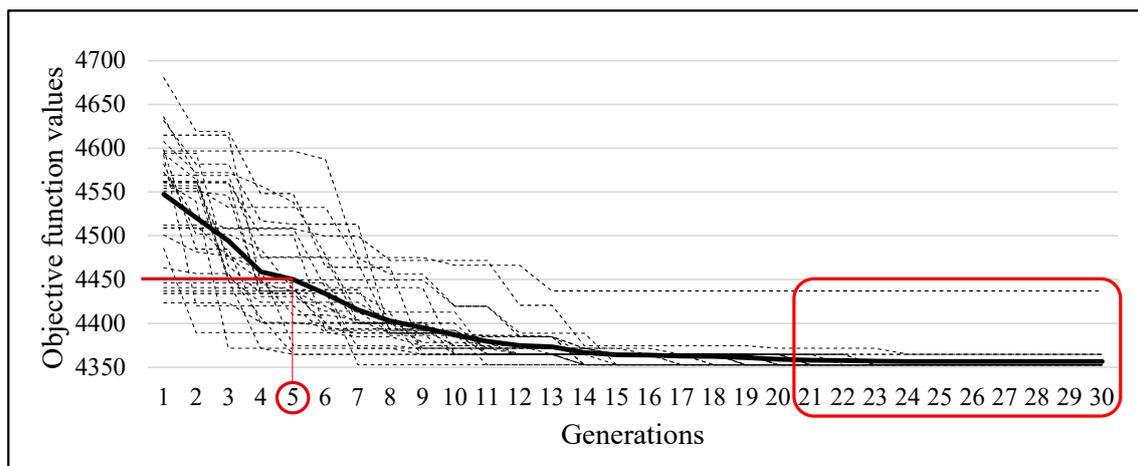
図 3.7 初期学習率  $\epsilon_0$  を変化させたときの提案手法 3-2 と提案手法 3-1 の学習率推移

略を再現できている割合が、初期学習率 $\epsilon_0$ が小さい順に 0%, 70%, 60%, 60%であることを確認した(図 3.7 下段参照)。一方、提案手法 3-2 の学習率推移は、(3.15)(3.16)式を用いることで、どの初期学習率 $\epsilon_0$ でも APBIL 探索戦略を再現できていることを確認した(図 3.7 上段参照)。このことから、APBIL 探索戦略の活用が解の質向上に寄与するといえる。また、初期学習率 $\epsilon_0$ 毎の学習率最大値の範囲に着目すると、提案手法 3-1 では(3.12)~(3.14)式より、学習率 $\epsilon^{gen}$ の最大値が初期学習率 $\epsilon_0$ に近い値を取るため、学習率 $\epsilon^{gen}$ の最大値は初期学習率 $\epsilon_0$ を変化させることで変動し、約 0.01~0.5 となることを確認した(図 3.7 下段参照)。一方、提案手法 3-2 では(3.15)式より、初期学習率 $\epsilon_0$ によらず学習率 $\epsilon^{gen}$ の最大値が約 0.5~約 0.7 に収まっていることを確認した(図 3.7 上段参照)。更に、表 3.7 及び図 3.7 より、学習率最大値を一定範囲内に収めることで、解のばらつきを抑えられるという仮説を立証した。

初期学習率 $\epsilon_0$ を 0.1 とした提案手法 3-2 による全 30 回試行の収束特性とその平均収束特性を図 3.8 (a)に、初期学習率 $\epsilon_0$ を 0.1 とした提案手法 3-1 による全 30 回試行の収束特性とその平均収束特性を図 3.8 (b)に示す。尚、破線が各 30 回分の収束特性を表し、実線が平均



(a) 提案手法 3-2 の収束特性



(b) 提案手法 3-1 の収束特性

図 3.8 提案手法 3-2 と提案手法 3-1 の収束特性

収束特性を表す。図 3.8 中の世代数が 5 世代時点での目的関数平均値を比較すると、提案手法 3-2 は約 4440、提案手法 3-1 は約 4450 となり、提案手法 3-2 の方が早い段階で質の高い解を導出していることを確認した。この結果の考察として、提案手法 3-2 は、提案手法 3-1 に比べ、初期探索時の学習率 $\epsilon^{gen}$ の正の傾きが大きく、探索の前半で大域的に探索する APBIL 戦略を実現できたためであると考えられる。また、探索の終盤に着目すると(図 3.8 中赤枠内参照)、提案手法 3-2 は、30 回中 29 回最良値を導出する一方、提案手法 3-1 は、30 回中 26 回最良値を導出することを確認した。この結果の考察として、提案手法 3-2 は提案手法 3-1 に比べ、探索終盤の学習率 $\epsilon^{gen}$ を小さくして探索の後半に探索の集中化を促すことで、APBIL 戦略を実現できたためであると考えられる。

図 3.9 に、世代数が 1 時点の初期生産計画 No. 1(図 3.9 (a))と、初期学習率 $\epsilon_0$ を 0.1 とした提案手法 3-2 により導出した最良生産計画(図 3.9 (b))を示す。図 3.9 より、最良生産計画は、初期生産計画 No. 1 に比べ全ラインで電力単価が最も高い時間帯(13:00~16:00)にて、電力消費が小さい段取り替え(図 3.9 中段取り替え時間)を長く行い電力消費が大きい生産を

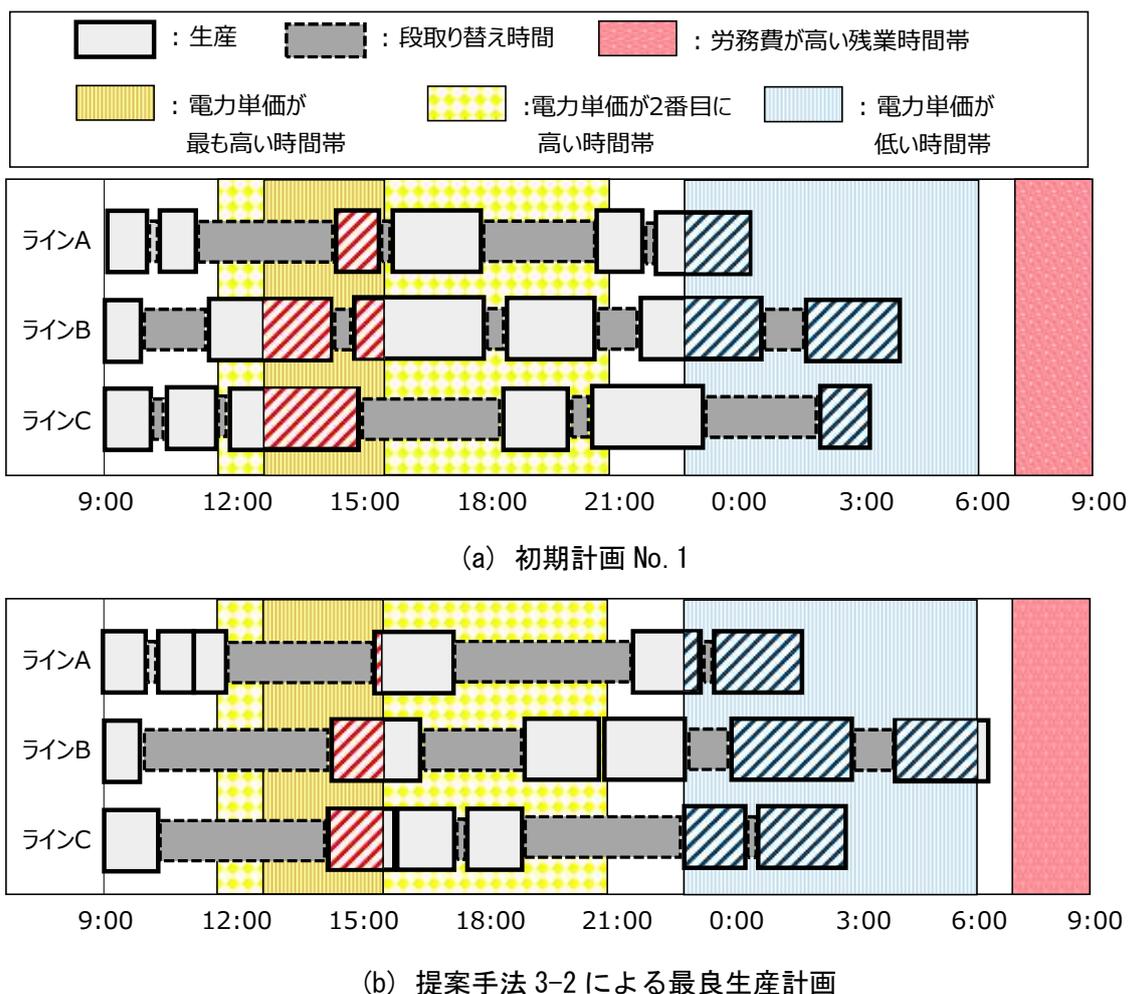
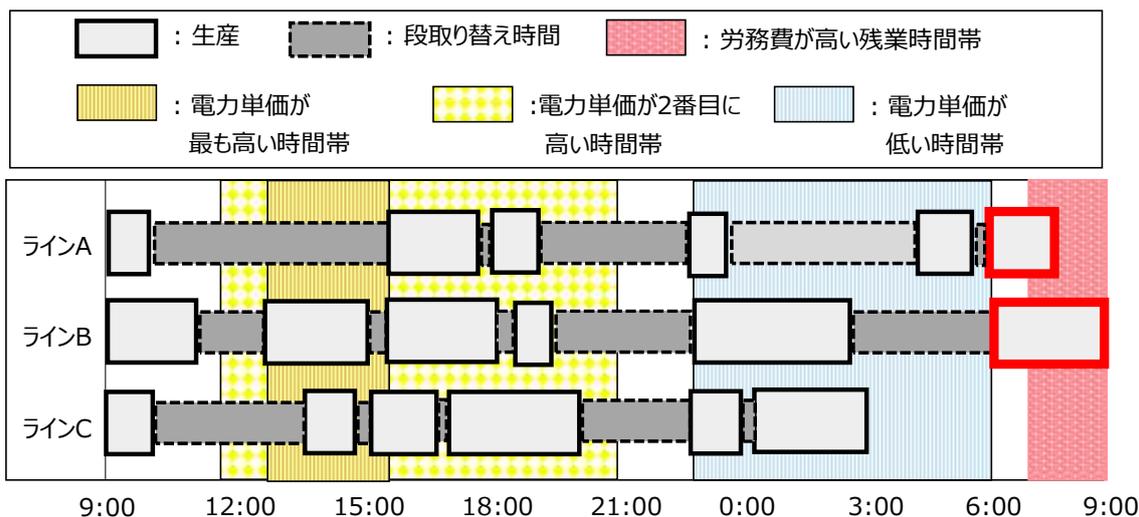


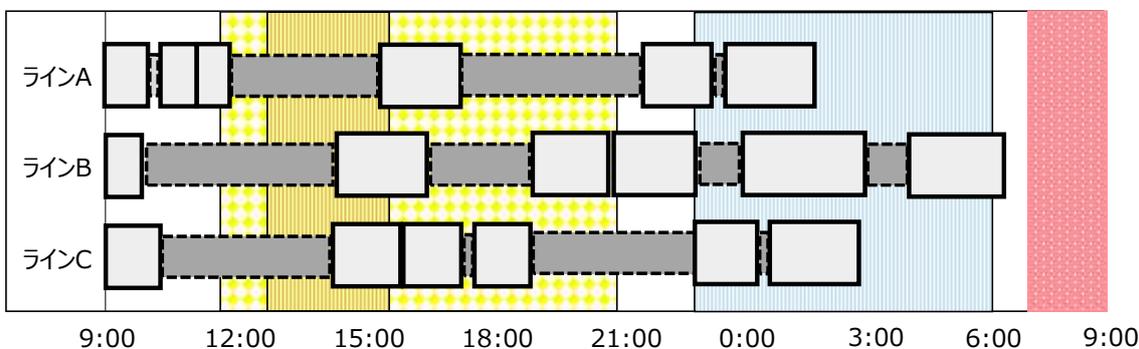
図 3.9 初期計画 No. 1 と提案手法 3-2 による最良生産計画例

抑えている(図 3.9 中赤色斜線参照)。更に、最良生産計画は、初期生産計画 No. 1 に比べ電力単価が低い夜間帯(23:00~6:00)にて、電力消費が大きい生産(図 3.9 中生産)を長く行っている(図 3.9 中青色斜線参照)。これは、目的関数値を生産性とエネルギー効率と環境負荷の各指標のコストを合計した生産コストの最小化としているため、電力費用と労務費の両方の観点で最適な計画を導出できていることを示す。更に、電力単価が最も高い時間帯の付近の生産計画に着目すると、最良生産計画は、敢えて段取り替え時間が長い製品間の組合せを設定し、消費電力量が生産時よりも低い段取り替え時間を電力単価が最も高い時間帯に当たるような計画が導出出来ていることが確認できる。これにより、生産設備全体の買電コストを抑えていることを確認した。

図 3.10 に、世代数が 1 時点の初期生産計画 No. 2(図 3.10(a))と、初期学習率 $\epsilon_0$ を 0.1 とした提案手法 3-2 により導出した最良生産計画(図 3.10 (b))を示す。図 3.10 より、初期生産計画 No. 2 は、ライン A, B で労務費が高い残業時間帯(7:00~9:00)に生産している(図 3.10 中赤枠)。一方、最良生産計画は、全ラインで労務費が高い残業時間帯(7:00~9:00)の生産を回避している。これは、電力消費量が生産時よりも低い段取り替え時間を可能な限り電力単



(a) 初期計画 No. 2



(b) 提案手法 3-2 による最良生産計画

図 3.10 初期計画 No. 2 と提案手法 3-2 による最良生産計画例

価が最も高い時間帯に割り当てるように段取り替え時間が長い生産順序を設定しつつも、労務費が高い残業時間帯に割り当たらないように電力単価が低い時間帯における段取り替え時間を可能な限り短くなる生産順序を設定していることを確認した。これにより、労務費を抑えていることを確認した。

提案手法 3-2 の初期学習率 $\epsilon_0$ を 0.1 とした最良解のときの、需給連携シミュレータにより導出した需要電力量、購入電力量、ガスタービン発電量、及びターボ冷凍機入力電力量の時間系列データを図 3.11 に示す。図 3.11 より、ガスタービンは、電力単価が高い時間帯で発電量上限である 20kWh 発電し、購入電力量を抑えた運転をしていることを確認した。また、電力単価が高い時間帯では、電力を利用するターボ冷凍機への電気入力が 0kWh となり、電力消費を抑えた運用をしていることを確認した。ターボ冷凍機への電機入力が 0kWh となる時間帯では、蒸気吸収式冷凍機が代わりに稼働し、熱出力する運用になっていることを確認した。本研究では、電力消費量に応じて  $\text{CO}_2$  排出コストを換算しているため、電力消費量を抑えることで、 $\text{CO}_2$  排出コストも削減することができる。以上より、生産計画から出力された需要電力量に対して、エネルギープラント最適運用計画を立案出来ていることを確認した。そして、本研究の目的である需給連携シナリオの実現が達成された。

提案手法 3-2 の初期学習率 $\epsilon_0$ を 0.1 として、定数 $\beta$ を 0.5 から 0.9 まで 0.1 刻みで変更し、試行回数 30 回当たりの目的関数値を示す箱ひげ図を図 3.12 に示す。図 3.12 では、全ての定数 $\beta$ において、ほぼ全ての試行回数で最良値(目的関数値が 4352.899)を導出しているため、箱ひげ図の箱部が明確に分かるようになっていない。そして、1 点または 2 点のみ最良値が得られていないため、平均値(図 3.9 中の×印)が少しだけ最良値よりも大きな値になっている。図 3.12 より、定数 $\beta$ を変えても目的関数値の差は小さく、安定して質の良い解を導出で

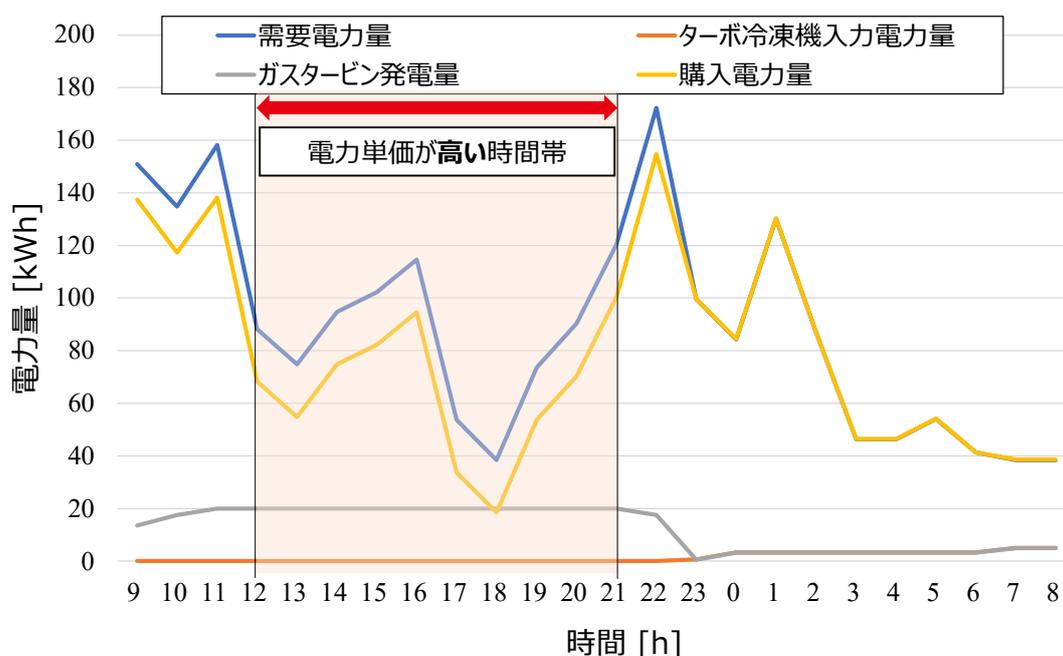


図 3.11 電力需要量、電力購入量、及び供給設備による電力供給量の推移 (提案手法 3-2 の最良解)

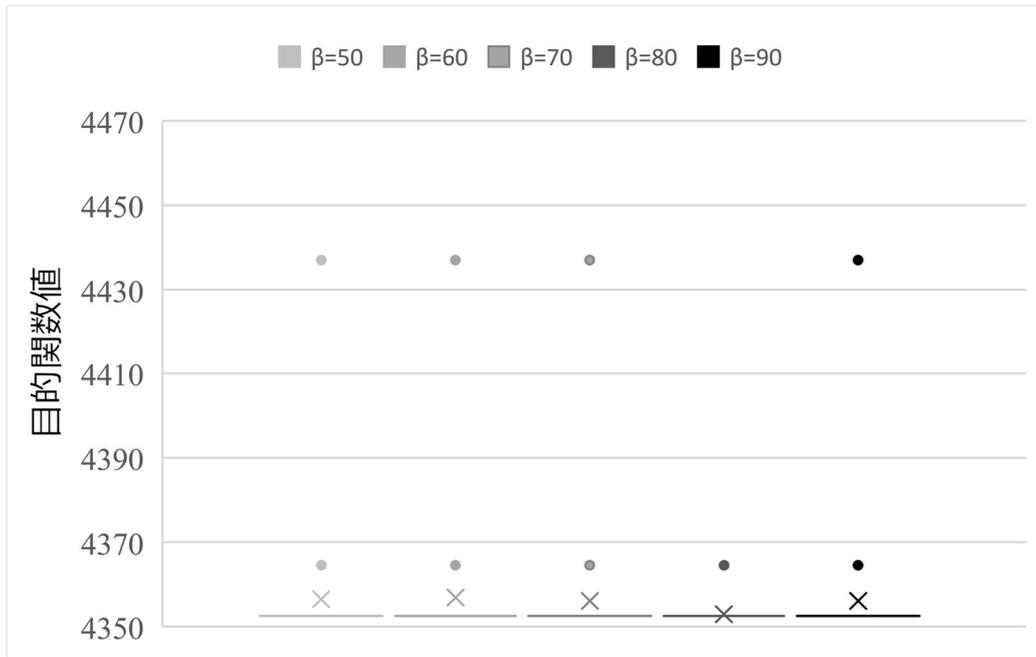


図 3.12 β値を変化させたときの提案手法 3-2 における目的関数値

きているため、提案手法 3-2 における定数 $\beta$ は、過度なパラメータチューニングは必要ないことを確認した。つまり、提案手法 3-2 を用いることで、初期学習率 $\epsilon_0$ に対するエンジニアリングが削減可能であることを確認した。

提案手法 3-2 を適用した並列分散型需給連携シミュレータにおけるスレッド数毎の全処理時間を図 3.13 に示す。尚、環境負荷量計算部は、並列分散型需給連携シミュレータを起動する計算機（以下、ホスト計算機）に導入したものとする。従来の実行方法であるスレッド数が 1 のときの全処理時間と比較し、スレッド数が 8 のときの全処理時間は、約 43.7% 削減することを確認した。また、スレッド数が 16 のときの全処理時間は、スレッド数 8 のときと比べ、約 1.3% 大きくなった。原因として、計算機③におけるスレッド数毎の CPU 負荷率に着目すると、スレッド数が 16 のときは CPU 負荷率が 100% となり、環境負荷量計算部での処理が発生したためと考えられる。

スレッド数が 8 のときの環境負荷量計算部処理用計算機台数毎の全処理時間を図 3.14 に示す。尚、0 台のときは、ホスト計算機のみを使用して実行した。ホスト計算機に環境負荷量計算部を導入したときの全処理時間と比較し、計算機④に環境負荷量計算部を導入したとき（環境負荷量計算部処理用計算機台数が 1 のとき）の全処理時間は、約 16.5% の削減を確認した。この原因として、環境負荷量計算部を計算機④とすることで、ホスト計算機の CPU 負荷が下がり、負荷分散を実現したためと考えられる。また、環境負荷量計算部を処理する計算機台数を増やすにつれて全処理時間が減少し、計算機を 4 台にした場合、1 台のときと比較して約 18% の削減を確認した。

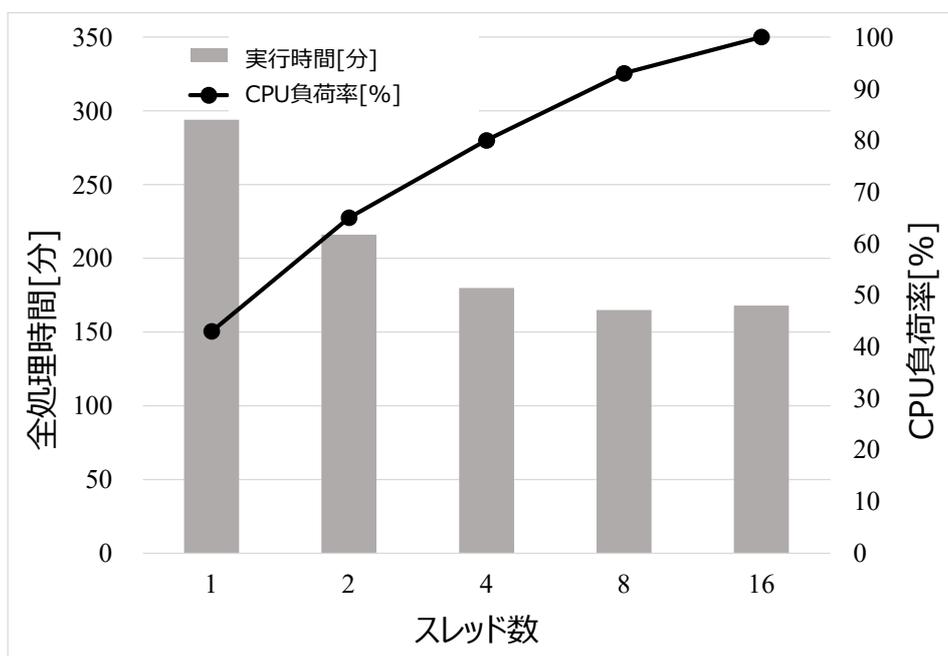


図 3.13 提案手法 3-2 を適用した並列分散型需給連携シミュレータにおけるスレッド数毎の全処理時間

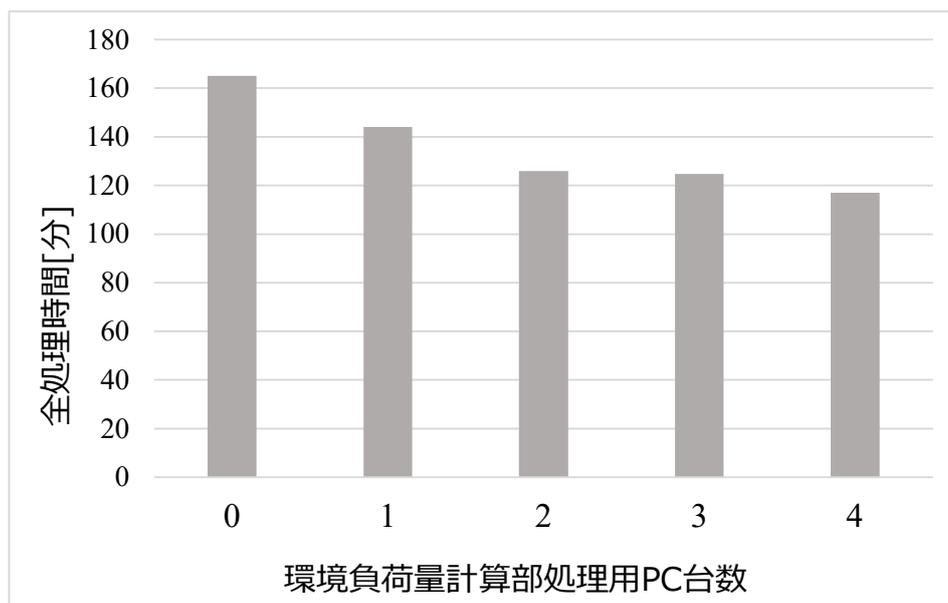


図 3.14 スレッド数が 8 のときの環境負荷量計算部処理用計算機台数毎の全処理時間

全処理時間の検証を通して、最速であったスレッド数を 8 とし計算機を 4 台にした方式の場合(117 分(図 3.14 の環境負荷量計算部処理用の計算機台数が 4 のとき)), 従来の実行方式(294 分(図 3.13 のスレッド数が 1 のとき))と比較し、約 60%の全処理時間削減が可能であることを確認した。つまり、最速の方式を用いることで、約 8 時間で複数回計画立案が可能であることを確認した。以上より、提案する並列分散型需給連携シミュレータの実用性を確認した。

### ③ シミュレーション結果に対する考察

表 3.8 に、提案手法 3-2 による最大世代数時における確率行列の例を示す。一般的に、整数型 PBIL 系統の手法では、確率行列の確率要素のうち目的関数値が優れている決定変数の値に対応する確率要素の値が 1 に近づいていき、探索の終盤では各決定変数で 1 つの確率要素が 0.9 以上の値になり、それ以外の確率要素が 0 に近い値になる傾向がある。しかし、本研究で扱う生産計画最適化問題は、短い時間である程度良質な需給計画を導出する必要があることから、目的関数値の評価回数を 1500 回としており、一般的な多点型最適化手法で設定される目的関数値の評価回数よりも少ない数を設定している。そのため、表 3.8 ではすべての決定変数における確率要素の最大値が 0.9 以下となっており、特に決定変数  $x_2$  のとき、確率要素の最大値は 0.614823 となり、0.7 以下となっている。これは、確率行列の更新が足りておらず、探索の集中化が十分に行われていないことを示す。つまり、提案手法 3-2 は、解の質に改善の余地があり、探索の後半に探索の集中化を強めることで、より効率的に探索できる可能性がある。

表 3.8 提案手法 3-2 による最大世代数時の確率行列例

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
1	0.000001	0.026136	0.000004	0.000022	0.101346	<b>0.87249</b>
2	0.000003	0.137582	<b>0.799088</b>	0.002208	0.060247	0.000872
3	0.011316	0.003567	0.15042	0.071428	<b>0.763112</b>	0.000157
4	0.17818	0.02562	0.008649	<b>0.787433</b>	0.000088	0.000031
5	0.158266	<b>0.614823</b>	0.11591	0.002789	0.000245	0.107967
6	<b>0.755342</b>	0.000204	0.016677	0.032473	0.137599	0.057706

### **3. 3. 5 改良整数型 APBIL の有効性と課題**

本節では、3ラインにおける6種類の製品における製品機種の処理順序を決定変数とした生産計画最適化問題を対象として、提案手法3-1と提案手法3-2を適用して最適生産計画を立案し、生産コスト指標を比較した。検証の結果、提案手法3-2は、提案手法3-1に比べ質の高い解を導出でき、解のばらつきを抑えることが可能であることを確認した。

一般的に、最適化の分野では、良い解の周辺には良い解が存在すると言われる (Proximate Optimality Principle)<sup>(3-12)</sup>。更に、最適化手法は一般的に、効率的に探索するため、探索の多様化と集中化のバランスが重要である。整数型 PBIL 系手法は、確率行列が持つ確率によって決定変数を決定し解探索を行う。探索前の確率行列は初期確率行列として全要素に同じ値が設定され、探索が進むにつれて世代数毎の最良解の決定変数に該当する確率行列要素を更新していくため、解探索の前半は探索の多様化が進み、解探索の後半は確率行列の更新が十分に行われる場合、探索の集中化が進む。しかし、本研究で扱う生産計画最適化問題は、短い時間である程度良質な需給計画を導出する必要があることから、整数型 PBIL 系手法で設定される一般的な最大世代数や個体数より小さい値を設定している。そのため、本節の検証結果より、提案手法3-2は確率行列の十分な更新による探索の集中化が未達成であることを確認した。従って、本研究で扱うような一般的な最適化問題に比べ最大世代数や個体数が少ない、言い換えると、目的関数値の評価回数が少ない生産計画最適化問題を対象とした場合、提案手法3-2は、導出する解の質に改良の余地がある。

### 3. 4 改良需給連携最適化方式の主問題に対する改良整数型 APBIL-RTS の適用の提案

#### 3. 4. 1 改良整数型 APBIL-RTS の提案

改良整数型 APBIL は、整数型 APBIL に対して APBIL が持つ適応関数を改良し、初期学習率 $\varepsilon_0$ によらず、探索に応じて学習率 $\varepsilon^{gen}$ を適切かつ適応的に変更させ、APBIL 探索戦略を再現可能となるようにした手法である。3.3 節のシミュレーション検証により、改良整数型 APBIL は、目的関数値の評価回数が少ない問題に適用する場合、確率行列の各値が収束しきらないことを確認した。つまり、目的関数値の評価回数が少ない問題において、探索の集中化を強めるためのアルゴリズムの改良が必要である。

本節では、改良整数型 APBIL の課題である探索の多様化と集中化のバランスを取ることに由来する解の質の向上という課題を解決するため、改良整数型 APBIL-RTS<sup>(3-13)</sup> (提案手法 3-3) を提案する。

#### 3. 4. 2 探索後半におけるローカルサーチの適用

改良整数型 APBIL の課題を解決するため、探索の前半で改良整数型 APBIL により探索の多様化を促し、探索の後半で探索の集中化を強めることが有効であるといえる。生産シミュレータは、工場の生産管理者が使用することを想定しており、生産管理者が生産シミュレータを使用する場合、事前に設定するパラメータ数を減らすことで、パラメータチューニング工数を削減することができる。探索の集中化を実現できる近傍探索手法の中で、Reactive Tabu Search (以下、RTS)<sup>(3-14)</sup>は、パラメータが初期タブーリスト長のみであり、且つ探索に応じてタブーリスト長を調整する機能を持つため、あらゆる初期タブーリスト長を与えても安定して良質な解を導出できる特徴を持つ。つまり、生産管理者が生産シミュレータを使用する際、RTS を用いることでパラメータチューニング工数を抑えることができる。そこで、本節では、改良整数型 APBIL に対し、探索の後半に RTS を適用した手法を提案する。

RTS のアルゴリズムを以下に示す。

- Step 1 初期解を生成する。
- Step 2 現在の解から個体数 $N_{neighbor}$ 個の近傍解を生成する。
- Step 3 個体毎に目的関数値を求め、個体間で評価する。
- Step 4 個体毎にタブー条件を違反しているか判定する。
- Step 5 タブー条件を違反していない個体の中から最良個体を選択する。
- Step 6 選択された個体が、決められた世代数以内で再び探索された場合、タブーリスト長

を長くする。

- Step 7 探索された個体が再び出現するまでの世代数の移動平均より多い世代数でタブーリスト長が調整されない場合、タブーリスト長を短くする。
- Step 8 予め設定した世代数が最大世代数に達した場合、これまで探索した中での最良解を出力し、探索を終了する。それ以外の場合、世代数を1追加し、Step 2に戻る。

### 3. 4. 3 主問題に対する改良整数型 APBIL-RTS を用いた解法アルゴリズム

本研究で扱う改良需給連携最適化方式の主問題は、製品の生産順序を決定変数とする。決定変数は1から製品機種数までの異なる整数値の組合せである。従って、3.1.3節と同様、使用した整数値を記録し、未使用の整数値を決定変数として選択可能なアルゴリズムを追加した。

改良需給連携最適化方式の主問題に提案する改良整数型 APBIL-RTS を適用した際のアルゴリズムを3つのパートに分けて以下に示す。

〈3・1〉 初期化 初期化のアルゴリズムを以下に示す。

- Step 1 改良整数型 APBIL 世代数 $iter_1$ 及び RTS 世代数 $iter_2$ を1とする。
- Step 2  $z = 1, L_{uz} = \{\}, L_{uuz} = \{1, \dots, p_z\}$ とする。  
ここで、 $p_z$ は生産ライン $z$ の製品数
- Step 3 確率行列の各要素を(3.7)式により初期化する。
- Step 4 個体数 $in$ 及び生産順序 $y$ を1とする。

〈3・2〉 改良整数型 APBIL 改良整数型 APBIL アルゴリズムを以下に示す。

- Step 1 確率行列 $P_{xyz}^{iter_1}(x \in L_{uuz})$ を用いて、生産ライン $z$ における生産順序 $y$ 番目となる製品 $x$ を決定する。
- Step 2 Step 1で決定した生産ライン $z$ における製品 $x$ をリスト $L_{uz}$ に保存し、リスト $L_{uuz}$ から除外する。
- Step 3 生産順序 $y$ が $p_z$ と同じ場合、Step 4に移動する。それ以外の場合、生産順序 $y$ を $y + 1$ とし、Step 1に戻る。
- Step 4 個体数 $in$ が $N_{ind}$ と同じ場合、Step 5に移動する。それ以外の場合、生産順序 $in$ を $in + 1$ とし、Step 1に戻る。
- Step 5 (3.15)(3.16)式により、学習率 $\varepsilon^{gen}$ を更新する。生産順序 $y$ が $p_z$ と同じ場合、Step 4に移動する。それ以外の場合、生産順序 $y$ を $y + 1$ とし、Step 1に戻る。

- Step 6 Step 1~Step 5 で得た生産順序を用いて、個体毎に目的関数を算出し、個体間で最も優れた目的関数値を持つ個体を最良個体とする。
- Step 7 Step 5 で更新した学習率 $\varepsilon^{gen}$ と、Step 6 で選択した最良個体の決定変数を用いて、(3.9)式により確率行列を更新する。
- Step 8 (3.10)式により、突然変異確率 $MUT\_P$ に従って確率行列を突然変異させる。
- Step 9 (3.11)式により、確率行列を正規化する。
- Step 10 生産ライン数 $z$ が $L$ と同じ場合、Step 11 に移動する。それ以外の場合、生産ライン数 $z$ を $z + 1$ とし、Step 1 に戻る。
- Step 11 世代数 $iter_1$ が $im_{iiapbil}$ と同じ場合、これまで探索した中での最良解の生産順序を $IIAPBIL_{best}$ に保存し、RTS パートに移動する。それ以外の場合、世代数 $iter_1$ を $iter_1 + 1$ とし、Step 1 に戻る。

〈3・3〉 RTS RTS アルゴリズムを以下に示す。

- Step 1 初期解を $IIAPBIL_{best}$ とする。
- Step 2 生産順序の近傍生成により、現在の解から個体数 $N_{neighbor}$ 個の生産順序を生成する。
- Step 3 個体毎に目的関数値を求め、個体間で評価する。
- Step 4 個体毎にタブー条件を違反しているか判定する。
- Step 5 タブー条件を違反していない個体の中から最良個体を選択する。
- Step 6 選択された生産順序が、決められた世代数以内で再び探索された場合、タブーリスト長を長くする。
- Step 7 探索された生産順序が再び出現するまでの世代数の移動平均より多い世代数でタブーリスト長が調整されない場合、タブーリスト長を短くする。
- Step 8 世代数 $iter_2$ が $im_{rts}$ と同じ場合、これまで探索した中での最良解の生産順序と生産計画を出力し、探索を終了する。それ以外の場合、世代数 $iter_2$ を $iter_2 + 1$ とし、Step 2 に戻る。

### 3. 4. 4 シミュレーション

#### ① シミュレーション条件

主問題は、実際の組立加工系工場の組立工程の一部をモデル化し検証する。具体的には、2 ラインにおける 8 種類の製品の処理順序(16 変数)である。改良整数型 APBIL(提案手法 3-2)と RTS(以下、従来手法 3-2)、改良整数型 APBIL-RTS(提案手法 3-3)を主問題に適用し、導出した生産コストを比較する。

共通パラメータ，従来手法 3-2，提案手法 3-2，及び提案手法 3-3 の各パラメータを以下に示す。これらのパラメータはシミュレーション前に決定する。また，本検証における各種適用手法の評価回数は同一としている。

(1) 共通パラメータ

試行回数：30， $\alpha_1, \alpha_2, \alpha_3$ ：全て 0.333， $p_z$ ：8， $L$ ：2

(2) 提案手法 3-2・提案手法 3-3 パラメータ

初期確率行列の要素：全て 1/6，初期学習率 $\epsilon_0$ ：0.1， $MUT_P$ ：0.02， $S$ ：0.02， $\beta$ ：0.8， $N_{neighbor}$ ：50

(3) 提案手法 3-2 パラメータ

(最大世代数 30)  $im_{iiapbil}$ ：30，(最大世代数 50)  $im_{iiapbil}$ ：50

(4) 従来手法 3-2 パラメータ

$N_{ind}$ ：50，(最大世代数 30)  $im_{rts}$ ：30，(最大世代数 50)  $im_{rts}$ ：50

(5) 提案手法 3-3 パラメータ

(最大世代数 30)  $im_{iiapbil}$ ：10, 15, 20, 24， $im_{rts}$ ：20, 15, 10, 6，

(最大世代数 50)  $im_{iiapbil}$ ：25, 30, 40， $im_{rts}$ ：10, 20, 25

実行環境は，Intel Core i9-10980XE (3.00GHz) の計算機における C 言語(Microsoft Visual studio 2019 Visual C++)，C# (Microsoft Visual studio 2019 Visual C#)，GLPK (VS2012 ARM Cross Tools Command Prompt)，Microsoft SQL Server 2016 である。副問題は，線形計画問題として定式化可能であるため，線形計画法ソルバー(GLPK)<sup>(3-11)</sup>を用いる。

## ② シミュレーション結果

表 3.9 に，30 回試行における提案手法 3-3 と提案手法 3-2 と従来手法 3-2 による目的関数の平均値・標準偏差値と，最小解の導出数の比較を示す。表 3.9 より， $im_{iiapbil}$ が 15, 20, 24 のとき，提案手法 3-3 による目的関数値の平均値は，提案手法 3-2 及び従来手法 3-2 よりも小さく， $im_{iiapbil}$ が 15 のとき，提案手法 3-3 による目的関数値の標準偏差値は，提案手法 3-2 及び従来手法 3-2 よりも小さいことを確認した。また，最良解導出回数では，従来手法 3-2 では 3 回，提案手法 3-2 では 2 回最良解を導出した。一方，提案手法 3-3 ではすべての $im_{iiapbil}$ で 3 回以上最小解を導出し，特に $im_{iiapbil}$ が 24 のとき，6 回最良解を導出できることを確認した。これらの結果から，多様化と集約化のバランスを取ることで，より効率的な探索が可能であることを確認した。

表 3.9 提案手法 3-3, 提案手法 3-2, 及び従来手法 3-2 による目的関数の平均値・標準偏差値, 最良解導出数の比較

手法	$im_{iiapbil}$	平均値	標準偏差	最良解導出回数
提案手法 3-3	10	2536.014	30.344	3
	15	2521.930	<b>27.915</b>	4
	20	<b>2519.220</b>	30.202	5
	24	2521.810	33.542	<b>6</b>
提案手法 3-2	30	2522.179	30.194	2
従来手法 3-2	0	2575.742	57.690	3

また, 提案手法 3-3 で最も良質な解を得た  $im_{iiapbil}$  が 20 のとき, 提案手法 3-2 及び従来手法 3-2 を対象に, 検定を行った。各手法による結果の正規性は, D'Agostino & Pearson 検定と Anderson & Darling 検定で確認されなかった。また, 3 手法で対応関係が無いから, Kruskal-Wallis 検定を適用した。その結果, p 値が 1.37E-06 であり 0.05 以下であることから, 各手法間に, 有意水準 5% で有意な差がないとは言えないことを確認した。しかし, どの 2 手法間に有意な差がないとは言えないかは, 明確ではないため, 事後検定を行った。提案手法 3-3 と提案手法 3-2 の間には, 対応関係があり, 提案手法 3-3 と従来手法 3-2, 及び提案手法 3-2 と従来手法 3-2 の間には, 対応関係がない。従って, 提案手法 3-3 と従来手法 3-2, 及び提案手法 3-2 と従来手法 3-2 に対しては, Holm 法による補正を用いた Mann-Whitney の U 検定を行った。提案手法 3-3 と提案手法 3-2 に対しては, Holm 法による補正を用いた Wilcoxon の符号検定を行った。検定の結果を表 3.10 に示す。表 3.10 より, p 値が全ての組合せで 0.05 以下であることから, 提案手法 3-3 と提案手法 3-2, 及び提案手法 3-3 と従来手法 3-2 の 2 手法間に有意な差がないとは言えないことを確認した。

図 3.15 に 30 回試行した際の  $im_{iiapbil}$  が 20 のときの提案手法 3-3 の平均収束特性 (赤線) と提案手法 3-2 の平均収束特性 (青線) を示す。世代数が 1~20 のとき, 提案手法 3-3 は改

表 3.10 提案手法 3-3, 提案手法 3-2, 及び従来手法 3-2 の全組合せにおける 2 手法間の検定結果

手法	$im_{iiapbil}$	提案手法 3-3	提案手法 3-2	従来手法 3-2
提案手法 3-3	20		S	S
提案手法 3-2	30	0.00924		S
従来手法 3-2	0	3.17E-05	1.26E-04	

S: Significance, NS: Non Significance

良整数型 APBIL による探索を行うため、提案手法 3-3 と提案手法 3-2 の最良解の推移は同じである。世代数が 21 回以降の時に着目すると、提案手法 3-3 は、提案手法 3-2 より高品質な解を探索していることを確認した。この結果の考察として、目的関数値の評価回数が少ない問題では、探索の後半に RTS によるローカルサーチを行うことで、改良整数型 APBIL による探索に比べて探索の集中化がより強くなり、結果として効率的な探索が出来たと考えられる。従って、RTS による探索の集中化による有効性を確認した。

図 3.16 に 30 回試行における提案手法 3-3 と提案手法 3-2 と従来手法 3-2 の箱ひげ図による目的関数値比較結果を示す。図 3.16 の凡例における括弧内の数字は、改良整数型 APBIL の世代数  $im_{iiapbil}$  を表している。図 3.16 より、箱ひげ図中の×印で表される目的関数値の平均値は、 $im_{iiapbil}$  が 20 のときの提案手法 3-3 を適用した方法が最も優れていることを確認した。また、箱ひげ図の箱の中の横線で表される目的関数値の中央値も、同様に  $im_{iiapbil}$  が 20 のときの提案手法 3-3 を適用した方法が最も優れていることを確認した。目的関数値の最小値は、全手法で同一の解を導出することを確認した。一方で、目的関数値の最大値は、 $im_{iiapbil}$  が 15 のときの提案手法 3-3 を適用した方法が最も優れていることを確認した。これら結果を総合して評価すると、提案手法 3-3 が解の質の面で提案手法 3-2 と従来手法 3-2 より優れていることを確認した。

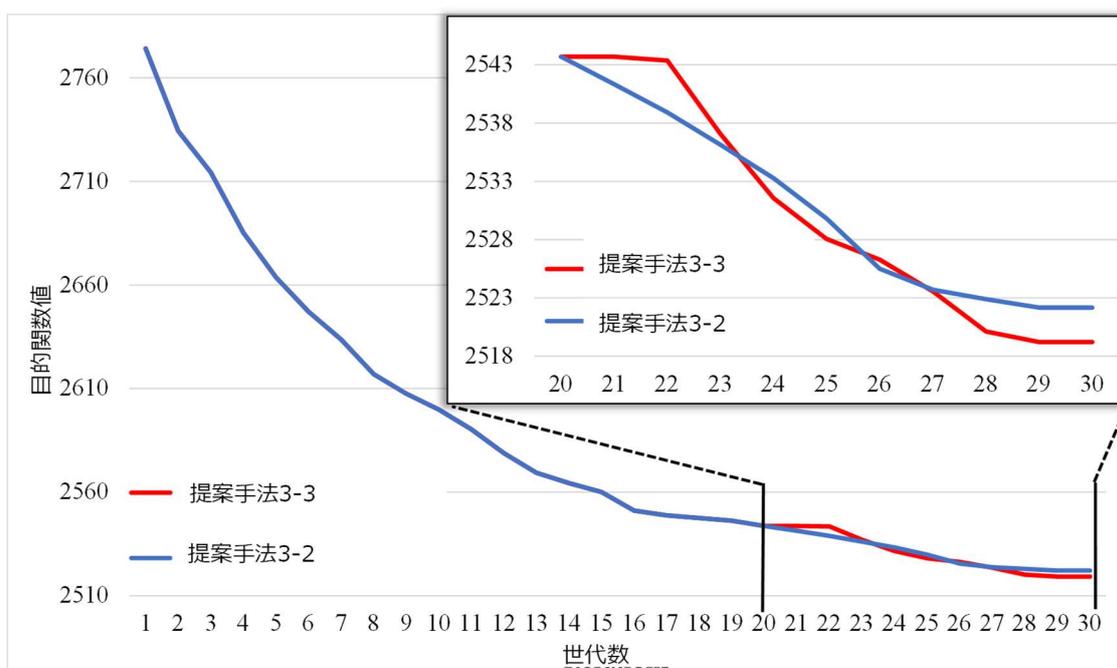


図 3.15 提案手法 3-3 と提案手法 3-2 による収束特性

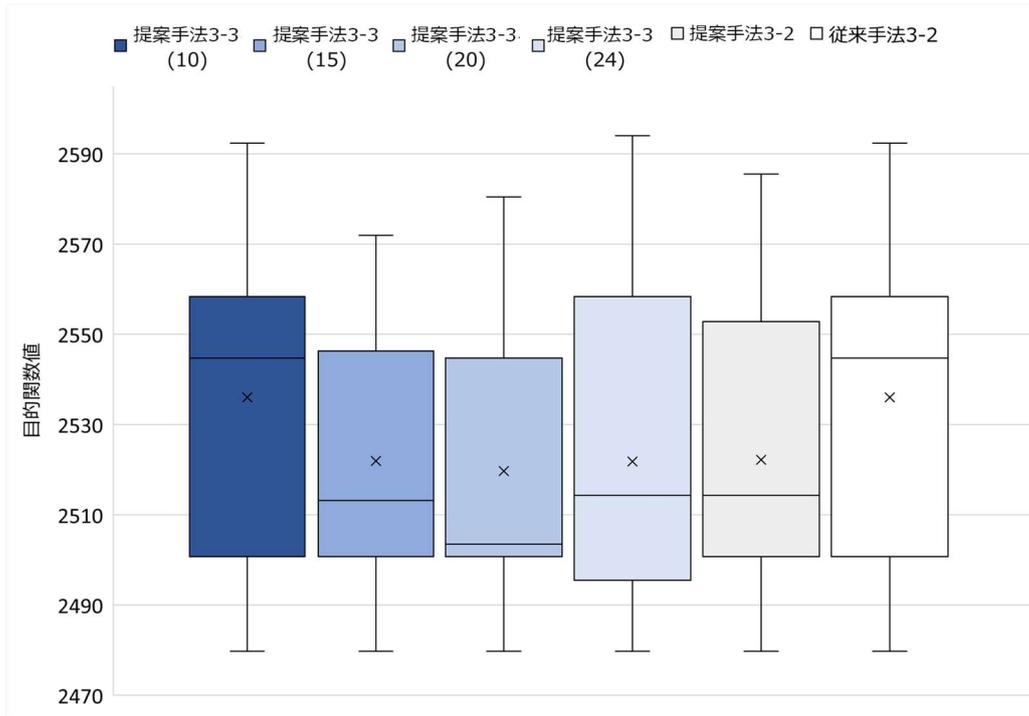


図 3.16 提案手法 3-3 と提案手法 3-2 と従来手法 3-2 の箱ひげ図による目的関数値比較

$im_{iipbit}$  が 20 のときの提案手法 3-3 を適用した最良解のときの、需給連携シミュレータにより導出した需要電力量、購入電力量、ガスタービン発電量、及びターボ冷凍機入力電力量の時系列データを図 3.17 に示す。図 3.17 より、ガスタービンは、図 3.11 と同様、電力単価が高い時間帯で発電量上限である 20kWh 発電し、可能な限り購入電力量を抑えた運転

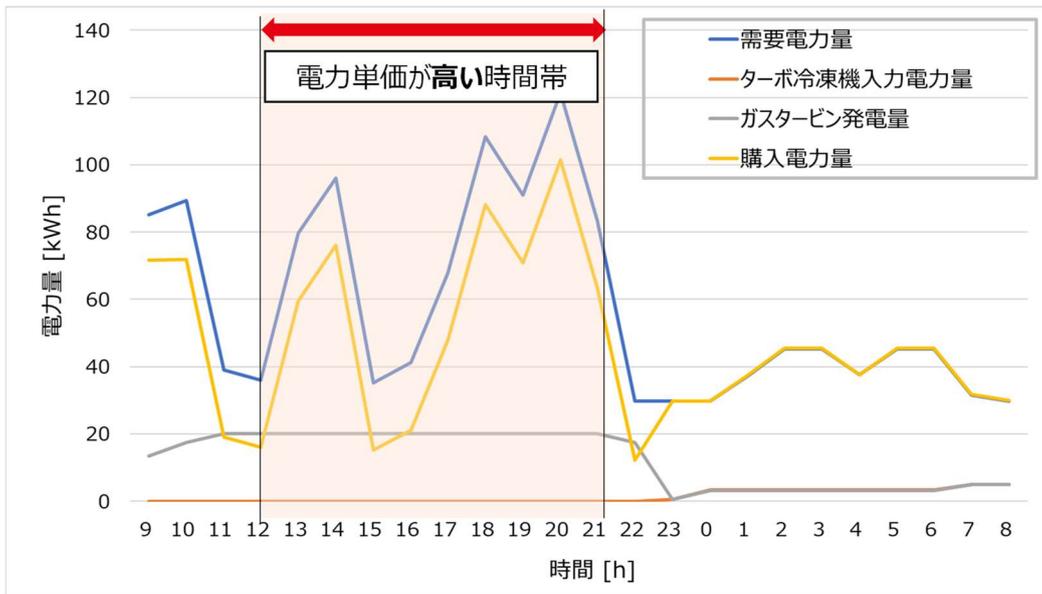


図 3.17 電力需要量、電力購入量、及び供給設備による電力供給量の推移 (提案手法 3-3 の最良解)

をしていることを確認した。また、電力単価が高い時間帯では、電力を利用するターボ冷凍機への電気入力が 0kWh となり、電力消費を抑えた運用をしていることを確認した。ターボ冷凍機への電機入力が 0kWh となる時間帯では、蒸気吸収式冷凍機が代わりに稼働し、熱出力する運用になっていることを確認した。本研究では、電力消費量に応じて CO<sub>2</sub> 排出コストを換算しているため、電力消費量を抑えることで、CO<sub>2</sub> 排出コストも削減することができる。以上より、図 3.11 で示した需要電力量の時系列データとは異なる需要電力量が入力されても、各種設備が購入電力量を最小化するための運用が達成されていることを確認し、本研究の目的である需給連携シナリオの実現が達成された。

これまでのシミュレーション検証により、最大世代数を 30 回としたときの提案手法 3-3 の有効性を確認した。しかし、図 3.15 では、世代数 30 回時点の目的関数値は提案手法 3-3 の方が提案手法 3-2 よりも優れているものの、世代数の途中で目的関数値は提案手法 3-2 の方が優れていることも確認した。そこで、世代数を 30 回から 50 回に増やし、 $im_{iiapbil}$  を 25, 30, 40 に設定した提案手法 3-3 と提案手法 3-2 の目的関数値を評価することとした。図 3.18 に、最大世代数を 50 回にした時の提案手法 3-4 と提案手法 3-3 の目的関数値比較結果を示す。尚、凡例の括弧内の数値は、改良整数型 APBIL の世代数  $im_{iiapbil}$  を表している。図 3.18 より、 $im_{iiapbil}$  を 30 または 40 に設定したときの提案手法 3-3 の目的関数値平均値（図 3.18 中×印）が、提案手法 3-2 よりも優れていることを確認した。また、箱ひげ図の箱の下部、つまり試行回数 30 回のうち上位 25% の目的関数値に着目すると、 $im_{iiapbil}$  を 25 に設定したときの提案手法 3-3 と提案手法 3-2 は、約 2500 であることを確認した。一方で、 $im_{iiapbil}$  を 30 または 40 に設定したときの提案手法 3-3 は、約 2480

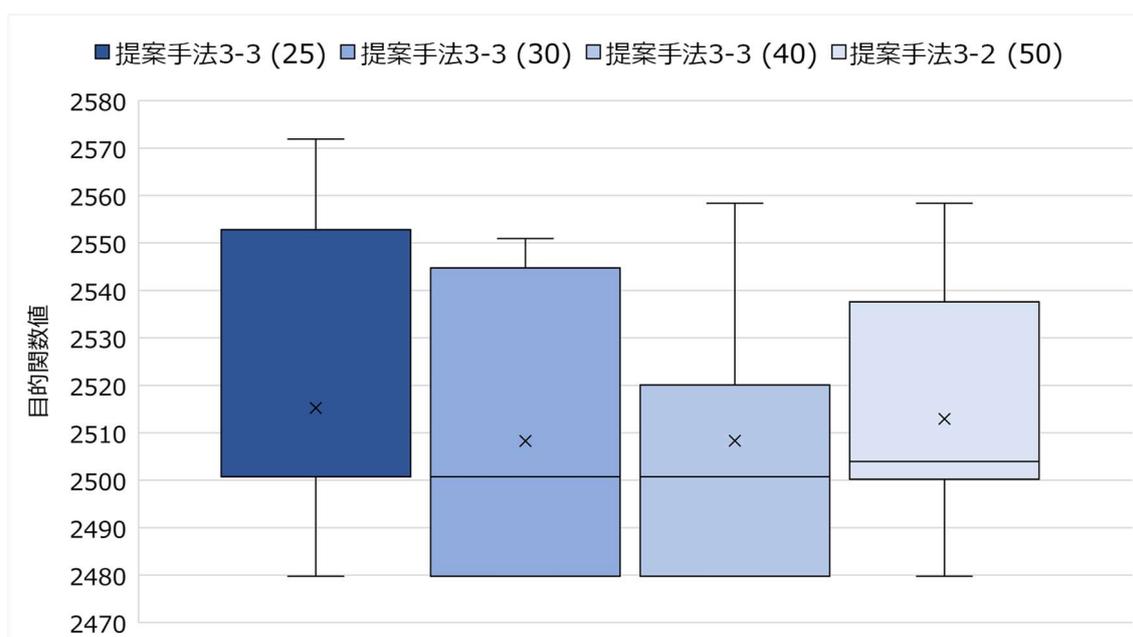


図 3.18 最大世代数を 50 回にしたときの提案手法 3-3 と提案手法 3-2 の箱ひげ図による目的関数値比較

であり、 $im_{iiapbil}$ を25に設定したときの提案手法3-3と提案手法3-2より優れた解であることを確認した。尚、約2480という数値は、本シミュレーションにおける目的関数値の最良値であり、 $im_{iiapbil}$ を30または40に設定したときの提案手法3-3は、試行回数30回のうち25%を占める約7回以上で本シミュレーションの目的関数値最良値を導出できていることを示す。これらの結果より、世代数を増やした場合でも、提案手法3-3の有効性を確認した。更に、これまでのシミュレーション検証により、最大世代数が30のときも50のときも、 $im_{iiapbil}$ を最大世代数の70%~80%とすることで、質の高い解を導出できることが示された。

### 3. 5 3章の結論

本章では、課題4を解決するため、主問題を生産シミュレータで模擬した実際の組立加工系工場の加工工程を対象とした生産計画最適化問題、供給側をエネルギープラント運用計画のための最適化ベンチマーク問題<sup>(3-15)</sup>とした改良需給連携最適化方式の主問題に対して、整数型PBILとその改良手法を提案し、解の質の面で比較検証を行った。

3.1.1節、3.1.2節では、PBIL及び整数型PBILの概要について説明し、3.1.3節では、シミュレータ上で表現した生産計画最適化問題に対して整数型PBILを用いた解法アルゴリズムについて説明した。3.1.4節では、整数型APBIL（提案手法3-1）とTS（従来手法3-1）を10変数の生産順序を扱う生産計画最適化問題に適用しシミュレーションを行った。シミュレーション結果により、提案手法3-1が従来手法3-1より質の高い解を導出できることを確認した。3.1.5節では、提案手法3-1の有効性を示しつつ、提案手法3-1が決定変数の決定に用いる確率行列の更新パラメータである初期学習率 $\epsilon_0$ に大きく依存し、初期学習率 $\epsilon_0$ を変えると解の質が変わるため、実際の工場現場での運用を考えると、問題が変わる度に事前に最適な初期学習率 $\epsilon_0$ を調整する必要があることを確認した。つまり、提案手法3-1は、初期学習率 $\epsilon_0$ の調整工数の削減が課題であることを確認した。

3.2.1節では、3.1.5節で示した提案手法3-1の課題を解決するための手法として、整数型APBILを提案した。3.2.2節では、初期学習率 $\epsilon_0$ を探索に応じて変動させることができる適応関数について説明した。3.2.3節では、改良需給連携最適化方式の主問題に対して整数型APBILを用いた解法アルゴリズムについて説明した。3.2.4節では、提案する整数型APBIL（提案手法3-2）と提案手法3-1を12変数の生産順序を扱う生産計画最適化問題に適用しシミュレーションを行った。シミュレーション結果により、提案手法3-2が提案手法3-1より質の高い解を導出できることを確認した。3.2.5節では、提案手法3-2の有効性を示しつつ、提案手法3-2は、初期学習率 $\epsilon_0$ がある一定以上の値になると解の質が落ちることを確認した。解の質が悪いときの学習率推移を確認した結果、想定通りのAPBIL探索戦略を再現できないことを確認した。再現できない原因を調査した結果、提案手法3-2の適応関数が、特定の初期学習率 $\epsilon_0$ に対してはAPBIL探索戦略に沿った学習率推移を取る一方で、それ以外の初期学習率 $\epsilon_0$ に対してはAPBIL探索戦略に沿わない学習率推移を取るためであることを確認した。つまり、エンジニアリング業務を削減するために、初期学習率 $\epsilon_0$ を変えても質の高い解を導出可能とするには、初期学習率 $\epsilon_0$ によらずAPBIL探索戦略を再現することができる適応関数の改良が必要であることを確認した。

3.3.1節では、3.2.5節で示した提案手法3-2の課題を解決するための手法として、改良整数型APBILを提案した。3.3.2節では、どの初期学習率 $\epsilon_0$ を設定しても、学習率 $\epsilon^{gen}$ を探索の前半から少しずつ学習率 $\epsilon^{gen}$ を増加させ、探索の後半に少し減少させるAPBIL探索戦略を再現可能な改良適応関数について説明した。3.3.3節では、改良需給連携最適化方式の主問題に対して改良整数型APBILを用いた解法アルゴリズムについて説明した。

3.3.4 節では、提案する改良整数型 APBIL (提案手法 3-3) と提案手法 3-2 を 18 変数の生産順序を扱う生産計画最適化問題に適用しシミュレーションを行った。シミュレーション結果により、異なる初期学習率 $\epsilon_0$ を与えても、提案手法 3-3 が安定して質の高い解を導出し、提案手法 3-2 より質の高い解を導出できることを確認した。また、APBIL 探索戦略の実現が、解の質向上に寄与することを確認した。また、提案手法 3-3 のパラメータである  $\beta$  を変化させても、安定して質の高い解を導出でき、ロバストであることを確認した。

3.3.5 節では、提案手法 3-3 の有効性を示しつつ、提案手法 3-3 は、本研究で扱うような目的関数値の評価回数が少ない問題に対して適用する場合、確率行列の要素が収束しきらず、結果として探索の後半に探索の集中化が働かず、解の質に改良の余地があることを確認した。つまり、解の質の向上に向けて、探索の後半に探索の集中化を取り入れる必要があることを確認した。

3.4.1 節では、3.3.5 節で示した提案手法 3-3 の課題を解決するための手法として、改良整数型 APBIL-RTS を提案した。3.4.2 節では、探索後半に適用するローカルサーチとして、他の手法よりパラメータ数が少ない RTS について説明した。3.4.3 では、改良需給連携最適化方式の主問題に対して改良整数型 APBIL-RTS を用いた解法アルゴリズムについて説明した。3.4.4 節では、提案する改良整数型 APBIL-RTS (提案手法 3-4) と提案手法 3-3 と RTS (従来手法 3-2) を 16 変数の生産順序を扱う生産計画最適化問題に適用しシミュレーションを行った。シミュレーション結果により、最大世代数 30 回のうち、改良整数型 APBIL を 20 回、RTS を 10 回としたときの提案手法 3-4 が、提案手法 3-3 や従来手法 3-2 と比較して質の高い解を導出できることを確認した。これにより、探索の前半に改良整数型 APBIL による探索の多様化を強め、探索の後半に RTS によるローカルサーチを行うことで探索の集中化を強めることで、探索の多様化と集中化のバランスを取り、効率的に探索することができるという仮説を立証した。更に、最大世代数が 30 のときのシミュレーション検証により、提案手法 3-3 と提案手法 3-4 の収束特性では世代数によっては提案手法 3-3 の方が優れていることを確認したため、最大世代数 50 のときの結果を確認したところ、提案手法 3-4 が最も優れた質の解を導出できることを確認し、提案手法 3-4 の有効性を確認した。

以上より、改良需給連携最適化方式の主問題に対する整数型 APBIL, 改良整数型 APBIL, 及び改良整数型 APBIL-RTS の適用により、課題 4 を解決した。

### 第3章の参考文献

- (3-1) S. Baluja, "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," Technical Report. Carnegie Mellon University, USA (1994).
- (3-2) I. Inza, P. Larranaga, and B. Sierra, "Feature Subset Selection by Estimation of Distribution Algorithms," Estimation of Distribution Algorithms, pp. 269-293 (2003).
- (3-3) D. Molina, F. Moreno-Garcia, and F. Herrera, "Analysis among winners of different IEEE CEC competitions on Real-parameters Optimization: Is there always improvement?," Proc. of 2017 IEEE Congress on Evolutionary Computation (2017).
- (3-4) K. Matsumoto, et al., "Connection Phase Estimation of Pole Mounted Distribution Transformers by Integer Form of Population Based Incremental Learning Considering Measurement Errors and Outliers by Correntropy," Proc. of 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (2021).
- (3-5) Y. Kobayashi, Y. Fukuyama, K. Seki, A. Oi, H. Yoshida, T. Jintsugawa, and H. Fujimoto, "Transposition Point Optimization for Voltage Unbalance Mitigation in Electric Power Distribution Systems by Integer Form of Population-Based Incremental Learning," Proc. of IEEE Congress on Evolutionary Computation (2023).
- (3-6) H. Zhenya, et al., "A New Population-based Incremental Learning Method for the Traveling Salesman Problem," Proc. of IEEE World Congress on Computational Intelligence (1999).
- (3-7) F. Glover and Manuel Laguna, "Tabu Search," Handbook of Combinatorial Optimization, Kluwer Academic Publishers, pp. 2093-2229 (1997).
- (3-8) 川口嵩平・福山良和 : 「実用的な生産シミュレータと整数型 Adaptive Population Based Incremental Learning を用いた最適生産順序導出手法の適用」, 電気学会スマートファシリティ研究会 (2022).
- (3-9) H. Pang, K. Hu, Z. Hong, "Adaptive PBIL algorithm and its application to solve scheduling problems," IEEE International Conference on Computer-Aided Design (2006).
- (3-10) 川口嵩平・福山良和 : 「改良整数型 Adaptive Population-Based Incremental Learning と生産シミュレータによる工場需給連携最適化フレームワーク」, 電学論 D, Vol. 143, No. 11, pp.728-741 (2023).
- (3-11) GLPK ウェブサイト, <https://en.wikibooks.org/wiki/GLPK>.

- (3-12) D. Niizuma, et al., "Multipoint Tabu Search based on proximate optimality principle—application of parts concept", *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 2, No. 6, pp. 635–642 (2007).
- (3-13) S. Kawaguchi and Y. Fukuyama, "Improved Adaptive Integer form of Population-Based Incremental Learning and Reactive Tabu Search for an Integrated Energy Supply and Demand Optimization Framework in Factories," *IEEE ICAIIC 2024*, (2024).
- (3-14) R. Battiti, "The Reactive Tabu Search," *ORSA Journal on Computing*, vol. 6, No. 2, pp. 126-140 (1994).
- (3-15) 電気学会 情報知能システムの新展開とその産業応用調査専門委員会：「産業応用のための最適化ベンチマーク問題集」, 電気学会技術報告 第 1287 号, 3.2 節 (2013).

## **4. 需給連携最適化方式の主問題に対する改良並列 RHPSO の提案**

JSP を主問題として需給連携最適化方式における従来研究では、近傍生産計画生成方法として、交換可能な隣同士のジョブを交換する方法のみを用いていたため、クリティカルブロックを用いた近傍生産計画生成に多様性がなかった（課題 2）。

本章では、上述の課題 2 を解決するため、需給連携最適化方式<sup>(4-1)</sup>の主問題に対して、以下のような近傍生産計画生成法を持つメタヒューリスティック手法を提案する<sup>(4-2)</sup>。尚、近傍生産計画生成法の詳細は、4.3.2 節を参照されたい。

- ・ AS 近傍に TE 近傍を加えた近傍生産計画生成法  
（改良並列 RHPSO（提案手法 4-1））
- ・ AS 近傍に CB 近傍を加えた近傍生産計画生成法  
（改良並列 RHPSO（提案手法 4-2））
- ・ AS 近傍に CB 近傍と TE 近傍両方を加えた近傍生産計画生成法  
（改良並列 RHPSO（提案手法 4-3））

### **4. 1 需給連携最適化方式における従来研究の課題と提案手法の概要**

#### **4. 1. 1 需給連携最適化方式における従来研究の課題**

著者らが従来研究により提案した工場における需給連携最適化方式<sup>(4-1)</sup>は、Generalized Benders Decomposition (GBD)<sup>(4-3)</sup>などで利用されている、最適化問題を階層的に主問題と副問題に分割し、それぞれの問題を階層的に解く方法を採用している。具体的には、二次エネルギーコストの最小化を考慮した JSP を主問題、エネルギープラント最適運用計画問題を副問題と定義し、双方の問題を階層的に解きながら目的関数値を評価する。この方式により、最適生産計画と最適エネルギープラント最適運用計画を同時に導出することが可能となる。

従来研究では、JSP を解くために開発された Hybrid Particle Swarm Optimization（以下、HPSO）<sup>(4-4)</sup>の改良手法である並列 Reactive HPSO（以下、並列 RHPSO）を提案し、これまでの著者らの従来研究の提案手法と比較し、解の質の向上を確認した<sup>(4-5)</sup>。しかし、これらの従来研究には問題があった。従来研究で適用したメタヒューリスティック手法では解の質に改善の余地があるという点である。

生産計画最適化問題における近傍生産計画導出手法は、これまでに様々な手法が提案されてきている。例えば、ジョブごとの工程順序を順列として表現し、工程順序を入れ替えることで近傍生産計画を導出する研究が行われている<sup>(4-6,7)</sup>。更に、近年では、生産計画のクリ

ティカルブロックに着目し、クリティカルブロックの中の工程を入れ替えることで近傍生産計画を導出する研究が行われている(4-8,9)。先行文献により、クリティカルブロックの中の工程を入れ替える方法は、実行可能解のみを解候補として生成することができるため、より効率的に探索することが可能であることが示されている。これら先行文献の知見に基づき、従来研究で提案している並列 RHPSO では、近傍生産計画導出手法として、クリティカルブロック近傍を用いている。ここで、クリティカルブロックについて簡単に説明する。生産計画の中で、工程の処理開始時間を遅らせることで、生産計画全体の終了時間が遅れるような特定の工程群の総称をクリティカルパスと呼ぶ(図 4.1 黄枠参照)。クリティカルパスの中で、同じ機械同士が時間的に隣接している複数工程をクリティカルブロックと呼ぶ(図 4.2 赤枠参照)。図 4.2 の生産計画の例では、マシン 2 におけるジョブ 2 とジョブ 3、マシン 2

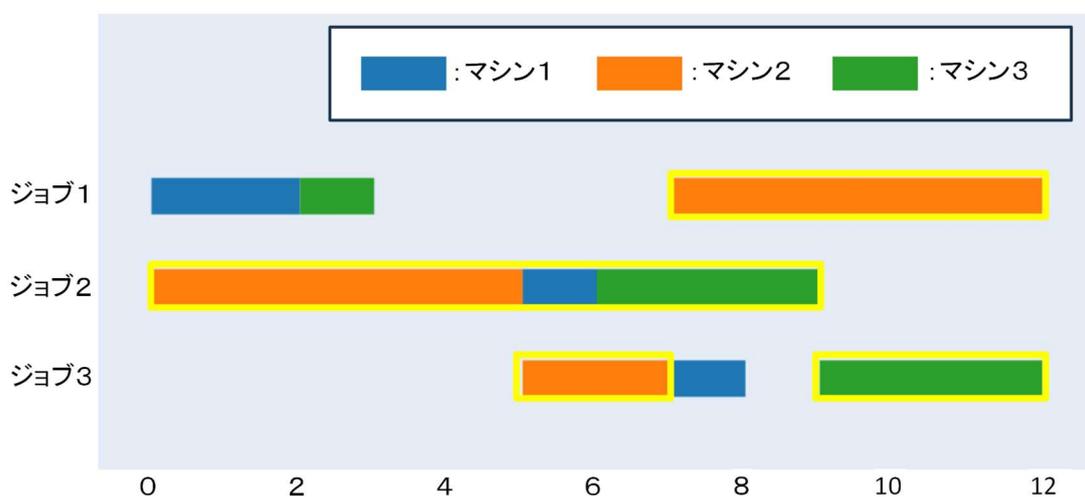


図 4.1 生産計画におけるクリティカルパス

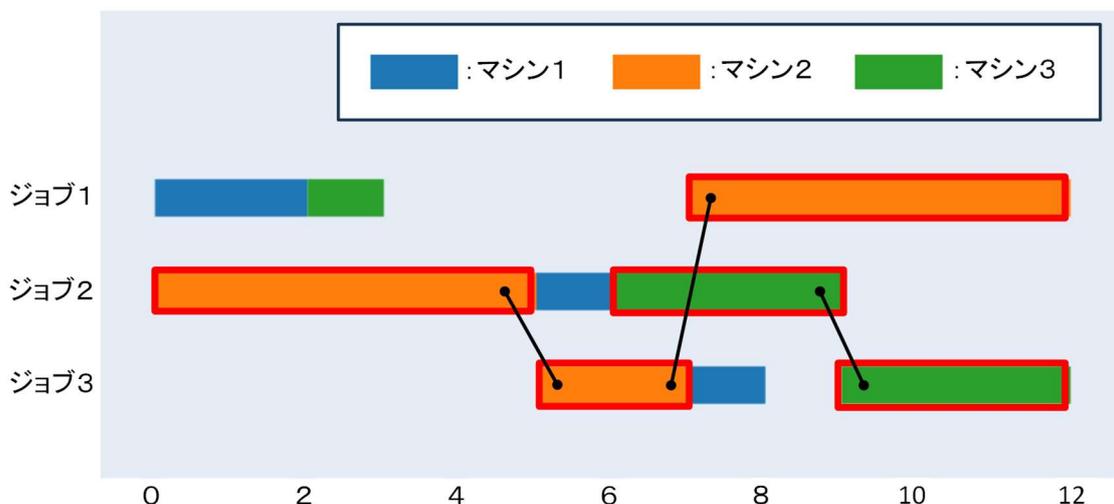


図 4.2 生産計画におけるクリティカルブロック

におけるジョブ 3 とジョブ 1, そしてマシン 3 におけるジョブ 2 とジョブ 3 の計 3 つのクリティカルブロックが存在している。このクリティカルブロックを用いたクリティカルブロック近傍において, 従来研究の並列 RHPSO では, クリティカルブロックの隣同士のジョブの処理順序を交換する方法のみを用いていたため, クリティカルブロックを用いた近傍生産計画生成に多様性を与える必要があった。

#### **4. 1. 2 提案手法の概要**

従来法の課題を解決するために, 本章では改良並列 RHPSO を提案する。提案手法である改良型並列 RHPSO では, 従来手法で用いていた AS 近傍に加えて, 以下 2 種類の近傍生産計画生成法を適用する。一つは, クリティカルブロック内の工程をクリティカルブロック内の工程の先頭または最後尾に移動する CB 近傍である。もう一つは, クリティカルブロック内の先頭と最後尾にある工程または工程の集合を入れ替える TE 近傍である。これらの CB 近傍及び TE 近傍は, AS 近傍に比べてダイナミックに工程を変化させることができる特徴を持つ。本章では, AS 近傍に CB 近傍を加えた近傍生産計画生成法による RHPSO (提案手法 4-1) と, AS 近傍に TE 近傍を加えた近傍生産計画生成法による RHPSO (提案手法 4-2) と, AS 近傍に CB 近と TE 近傍の両手法を加えた近傍生産計画生成法による RHPSO (提案手法 4-3) を提案する。シミュレーション検証では, 提案手法 4-1~4-3 を JSP に適用した際の解の質を評価する。更に, 提案手法 4-1~4-3 のローカルサーチとして用いている各探索点の RTS 処理に対して, 並列化処理を適用した。シミュレーション検証では, スレッド数を変更した際の全処理時間を評価する。

#### **4. 2 主問題の定式化**

以下に, 需給連携最適化方式の主問題の定式化を示す。

##### **4. 2. 1 決定変数**

主問題における決定変数は, 各工程の作業終了時間  $ET(O_{jm})$  ( $j = 1, \dots, J, m = 1, \dots, M$ ) である。ここで,  $O_{jm}$  はジョブ  $j$  の機械  $m$  の工程,  $J$  はジョブ数,  $M$  は機械台数である。従って, 決定変数の数は,  $J \times M$  となる。この  $J \times M$  の決定変数が決まることで, 解となる生産計画が決定する。

##### **4. 2. 2 目的関数**

主問題における目的関数は, 生産計画に存在する全工程の作業終了時の最大時間 (以下,

メイクスパン)の最小化と、電力使用のピーク時間帯における総電力消費量の最小化を目的関数とする。なお、本問題は一般的には多目的関数となるが、2つの目的関数をスカラー化手法により1つの目的関数とした。

$$\text{minimize } w_1 \times ET(O_{jm})_{max} + w_2 \times \sum_{k=ps}^{pe} EC(k) \quad (j = 1, \dots, J, m = 1, \dots, M) \quad (4.1)$$

$$EC(k) = \sum_{j=1}^J \sum_{m=1}^M \sum_{k=1}^{ET(O_{jm})_{max}} EU_{jm} \times ON_{jmk} \quad (k = 1, 2, \dots, ET(O_{jm})_{max}) \quad (4.2)$$

$$ON_{jmk} = \begin{cases} 1 & (O_{jm} \text{ exists}) \\ 0 & (O_{jm} \text{ not exists}) \end{cases} \quad (4.3)$$

ここで、 $ET(O_{jm})_{max}$ :メイクスパン、 $ps$ :ピーク時間帯の開始時刻、 $pe$ :ピーク時間帯の終了時刻、 $w_1, w_2$ :目的関数の各項における重み係数( $w_1 + w_2 = 1$ )、 $EC(k)$ :時間 $k$ の総電力消費量、 $EU_{jm}$ :ジョブ $j$ の機械 $m$ の単位時間1あたりの電力消費量、 $ON_{jmk}$ :各時刻における $O_{jm}$ の有無

### 4. 2. 3 制約条件

制約条件を以下に示す。

#### ① メイクスパン制約

メイクスパンに関する制約を以下に示す。

$$FOE(O_{jm})_{max} \geq FOE(O_{jm}) \quad (j = 1, \dots, J, m = 1, \dots, M) \quad (4.4)$$

(4.4)式は、各工程 $O_{jm}$ の作業終了時間( $ET(O_{jm})$ )は、全作業の生産終了時間を示すメイクスパン( $FOE(O_{jm})_{max}$ )以下であることを表している。

#### ② 作業開始時間制約

作業開始時間に関する制約を以下に示す。

$$ST(O_{jm}) \geq 0 \quad (j = 1, \dots, J, m = 1, \dots, M) \quad (4.5)$$

ここで、 $ST(O_{jm})$ :各工程の作業開始時間

(4.5)式は、各工程の作業開始時間は、全て0以上であることを表している。

③ 作業終了時間制約

作業終了時間に関する制約を以下に示す。

$$ET(O_{jm}) = ST(O_{jm}) + PT(O_{jm}) \quad (j = 1, \dots, J, m = 1, \dots, M) \quad (4.6)$$

ここで、 $PT(O_{jm})$ ：各工程の処理時間

(4.6)式は、各工程の作業終了時間は、対象工程の作業開始時間に対象工程の処理時間を足したものに等しいことを表している。

④ 同じジョブにおける作業時間制約

同じジョブにおける作業時間に関する制約を以下に示す。

$$ST(O_{j(m+1)}) \geq ET(O_{jm}) \quad (j = 1, \dots, J, m = 1, \dots, M) \quad (4.7)$$

(4.7)式は、同じジョブにおいて、工程の作業終了時間は、対象工程の次に処理される工程の作業開始時間と等しい、もしくは小さいということを表す。

⑤ 電力消費量上限制約

電力消費量の上限に関する制約を以下に示す。

$$EPC(k) \leq EC_{max} \quad (k = ps, \dots, pe) \quad (4.8)$$

ここで、 $EC_{max}$ ：ピーク時間帯における許容最大電力消費量

(4.8)式は、電力使用のピーク時間帯における電力消費量は、最大許容電力消費量を超えないということを表す。

⑥ メイクスパン上限制約

メイクスパンの上限に関する制約を以下に示す。

$$MS \leq MSlimit \quad (4.9)$$

ここで、 $MSlimit$ ：最大許容メイクスパン

(4.9)式は、メイクスパンは、最大許容メイクスパンを超えないということを表す。尚、制約条件(4.8)(4.9)式を満たさない場合、目的関数値にペナルティ値を付加する。

#### ⑦ 機械特性制約

機械特性に関する制約を以下に示す。

- 機械は故障しないことを前提とする。
- 1つの機械は、同時に2つ以上のジョブを処理できない。
- 一度処理が開始された機械は、途中で中断できない。

### 4. 2. 4 生産計画の表現方法

JSPは、文献(4-10)で説明しているように、事前に各ジョブに対する機械の技術的順序と処理時間が与えられ、各工程は機械番号と処理時間を持つ。各工程は、これらの情報を用いることで、(機械番号, ジョブ番号,  $ET(O_{jm})$ )という情報を一意に決めることができる。各工程の情報により、一意の生産計画を表現することができる。本章では、これに加え、各工程の情報として電力消費量を予め設定している。つまり、各工程が、機械番号, 処理時間, 電力消費量の3種類の情報を与えられることになる。これらの情報を用いることで、一意の生産計画に加え、生産計画の単位時間当たりの電力消費量を求めることができる。

### 4. 3 主問題に対する改良並列RHPSOの適用の提案

本節では、4.1節で説明した需給連携最適化方式の主問題を解くために、改良並列RHPSOを用いたアルゴリズムを提案する。

#### 4. 3. 1 RHPSO

RHPSOは、JSPに適用させるためにエージェントの速さや更新方法を改良し、優先リストを用いたPSOと、局所探索法として知られているTSを組み合わせた手法として開発されたHPSO<sup>(4-4)</sup>のTSによるローカルサーチをRTSに改良した手法である<sup>(4-11)</sup>。RHPSOでは、各エージェントが生産計画・優先リスト・速さを持つ。優先リストを以下に示す。

$$\mathbf{Y}^i = \begin{bmatrix} y_{11}^i & \cdots & y_{1J}^i \\ \vdots & \ddots & \vdots \\ y_{M1}^i & \cdots & y_{MJ}^i \end{bmatrix} \quad (4.10)$$

ここで、 $i$ :エージェント番号,  $\mathbf{Y}^i$ : $i$ 番目のエージェントの優先リスト,  $y_{mj}^i$ : $i$ 番目のエージェントの機械 $m$ ジョブ $j$ の工程における優先度

優先リストにおいて、同じ機械の中で優先度 $y_{mj}^i$ が小さい工程ほど優先度が高い。また、優先リストが決まると一つの生産計画が決定される。工程の速さを以下に示す。

$$\mathbf{v}^i = \begin{bmatrix} v_{11}^i & \cdots & v_{1J}^i \\ \vdots & \ddots & \vdots \\ v_{M1}^i & \cdots & v_{MJ}^i \end{bmatrix} \quad (4.11)$$

ここで、 $i$ :エージェント番号、 $\mathbf{v}^i$ : $i$ 番目のエージェントの速さ群、 $v_{mj}^i$ : $i$ 番目のエージェントの機械 $m$ ジョブ $j$ の工程の速さ $v_{mj}^i \in \{0,1\}$

連続値の決定変数を用いた非線形最適化問題に対応したオリジナルの PSO の速さは、決定変数の変化量を表している(4-12)。一方、HPSO における速さは、優先リストの更新処理において工程を交換する際に交換できるかどうかを 0, 1 で示している(0:交換可能, 1:交換不可)。

以下に、RHPSO のアルゴリズムを示す。

- Step. 1 エージェント毎に Giffler & Thompson アルゴリズム(4-13)の改良アルゴリズム(以下、改良 G&T アルゴリズム)(4-5)により初期生産計画を生成する。
- Step. 2 エージェント毎に優先リストの優先度及び速さを初期化する。
- Step. 3 エージェント毎に初期生産計画の目的関数値及び優先リストを  $pbest$  に保存し、 $pbest$  の中で最も目的関数値が良いエージェントの目的関数値と優先リストを  $gbest$  に保存する。
- Step. 4 エージェント毎に乱数を用いて速さ 1 を持つ工程の速さを 0 にする。
- Step. 5 エージェント毎に交換操作及び突然変異を行い、優先リストを更新する。
- Step. 6 エージェント毎に優先リストを用いた改良 G&T アルゴリズムにより変更生産計画を生成する。
- Step. 7 エージェント毎に RTS を実行する。
- Step. 8 エージェント毎に  $pbest$  と  $gbest$  の更新を行う。
- Step. 9 探索回数が上限に達した場合終了し、それ以外は Step. 4 へ。

### 4. 3. 2 改良型 RHPSO の提案

提案する改良型 RHPSO は、探索の多様化を促すことを目的として、RHPSO の RTS 処理における近傍生産計画生成法を修正している。本節では、RTS によるローカルサーチ部について、具体的に説明する。以下に、改良型 RHPSO のローカルサーチに適用する RTS アルゴリズムを示す(4-14)。

- Step. 1 ハッシュ関数を用いた保存リストを用意し、初期 TL 長を設定する。

- Step. 2 生成された変更生産計画を保存リストに保存する。
- Step. 3 近傍生産計画を生成する。
- Step. 4 近傍生産計画の中でタブーでない最も優れた生産計画を選び移動する。(アスピレーション基準を満たす場合、タブーであっても移動する)
- Step. 5 選ばれた最良生産計画の変更要素をタブーリストに、選ばれた最良生産計画を保存リストに保存する。
- Step. 6 選ばれた最良生産計画が以前に探索された生産計画であり、ある一定以下の探索回数で再度探索された場合、探索がループとなっているとみなし TL 長を長くする。
- Step. 7 以前探索された生産計画が現れるまでの探索回数の移動平均より長い間 TL 長の修正がなかった場合、もしくはすべての近傍生産計画がタブーであった場合、TL 長を短くする。
- Step. 8 生産計画の更新が閾値  $Iter_{not\_improved}$  以上行われなかった場合 Step. 9 へ、そうでない場合 Step. 3 へ。
- Step. 9 終了。

上記 RTS アルゴリズムの Step. 3 の近傍生産計画生成法において、クリティカルブロックにおける工程の交換操作のバリエーションを増やすことで、従来の交換操作に比べ、より多様性を持った近傍生産計画生成を行い、結果として効率的な探索を行う改良型 RHPSO を提案する。

従来手法である RHPSO の近傍生産計画生成法として用いていた方法は、図 4.3 に示すクリティカルブロック内の隣同士の工程のジョブの処理順序を入れ替える AS 近傍<sup>(4-15,16)</sup>と呼ばれる手法である。一方、提案手法である改良型並列 RHPSO では、RTS に適用する効果的な方法を検討した結果、以下の二つの手法を適用する。一つは、図 4.4 に示すクリティカルブロック内の工程をクリティカルブロック内の工程の先頭または最後尾に移動する CB 近傍という手法<sup>(4-17)</sup>である。CB 近傍は、著者らの事前検証にて、RTS の適用で有効性が確認できた。もう一つは、図 4.5 に示すクリティカルブロック内の先頭と最後尾にある工程を入れ替える手法<sup>(4-18)</sup>に加えて、工程の集合を入れ替える TE 近傍という手法である。これらの CB 近傍及び TE 近傍は、AS 近傍に比べダイナミックに工程を変化させることができる特徴を持つ。先行研究では、これら 3 種類の近傍解生成法のうちそれぞれ 1 種類のみ適用していたが、これらを組み合わせた手法は提案されていなかった。そこで、本章では、より効率的な探索を目的として、以下のような組合せによる手法を提案する。

- ・ AS 近傍に TE 近傍を加えた近傍生産計画生成法 (提案手法 4-1)
- ・ AS 近傍に CB 近傍を加えた近傍生産計画生成法 (提案手法 4-2)
- ・ AS 近傍に CB 近傍と TE 近傍両方を加えた近傍生産計画生成法 (提案手法 4-3)

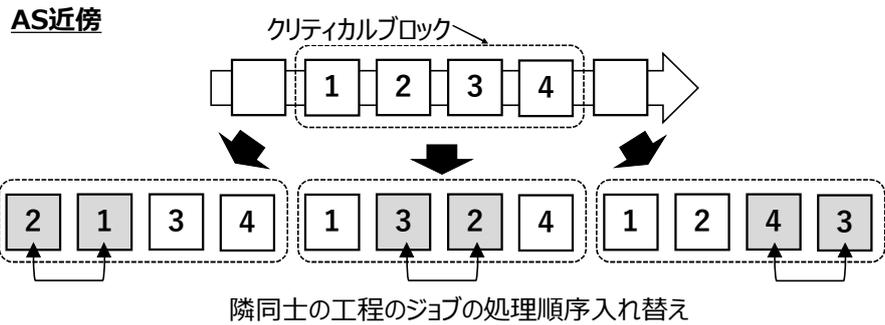


図 4.3 Adjacent Swapping 近傍例

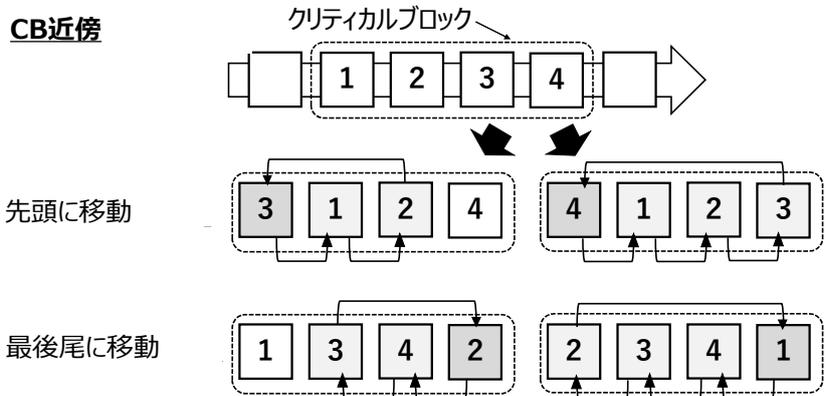


図 4.4 Critical Block 近傍例

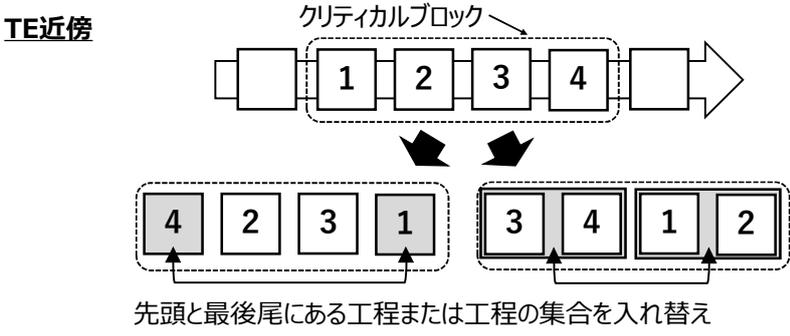


図 4.5 Top End 近傍例

### 4. 3. 3 改良並列 RHPSO の提案

ジョブショップ型生産計画は、多品種少量生産に移行しており、短時間に様々な生産計画を求めなければならず、高速に生産計画を求める必要がある。しかし、改良 RHPSO は各探索点が逐次的に計算しており、計算時間がかかってしまうという課題があった。また、2.3.2 節で説明した改良型 RHPSO アルゴリズムの Step. 7 のローカルサーチ部の RTS にかかる計算時間が全体計算時間の約 99.92% を占めており、この部分の計算時間の短縮化ができれば全体の計算時間短縮化に大きく寄与する。

RTS 部の処理時間短縮化の対策として、「RTS の世代数を減らす」ことや、「エージェントの個数を減らす」ことが考えられる。しかし、これらの対策によると効率的な探索ができず、結果として解の質の改善が達成できなかった。一方、「エージェントごとに並列化を行う」を考慮した際、RTS はエージェントごとに独立した処理を行うため、より広い解空間を探索でき解の質も並列前と同等の質が得られたため「並列化」を選択した。従って、提案手法では Step. 7 の RTS 部の並列化を行った。Step. 7 は、RTS の各エージェントの隣接生産計画数が異なり、且つ、終了条件を「生産計画の更新が閾値  $Iter_{not\_improved}$  以上行われない場合」としたため、各エージェントの計算時間が異なることになる。できるだけエージェント間の計算時間を同一にするためには、終了条件を一定の生産計画の更新回数にすることも考えられるが、最終的な解の質を上げることが出来なかったため、今回の終了条件としている。理論上エージェント毎のアンバランスな計算時間を考慮して、どのようにプロセスに配分するかを考慮する必要があると思われるが、実際には、エージェントに対する処理をいくつかのプロセスに等しく配分しても一定の高速化が得られたので、結果としては対象 Step 全てに対し、エージェントに対する処理をいくつかのプロセスに等しく配分する並列化を行った。この Step. 7 におけるエージェント毎の計算時間の違いによる計算時間の高速化特性については、シミュレーションにて考察を行った。

### 4. 4 副問題に対する解法アルゴリズム

本節で用いる副問題は、2.4.4 節で示す内容と同じため、2.4.4 節と同様とする。

### 4. 5 シミュレーション

本節では、10 機械 10 ジョブのベンチマーク問題 (abz6<sup>(4-19)</sup>) を用いて、二次エネルギーコスト最小化を考慮した JSP に対して、提案する改良並列 RHPSO (以下、提案手法 4-1~4-3) と従来手法である RHPSO<sup>(4-11)</sup> (以下、従来手法) 適用し、解の質の向上、及び OpenMP<sup>(4-20)</sup> を用いた並列計算による計算時間の高速化を検証した。更に、多品種少量生産への適用を考慮し、提案手法の様々なパラメータを適用した。

## 4. 5. 1 シミュレーション条件

abz6 の各ジョブの工程順序・処理時間・消費電力量を表 4.1 に示す。abz6 におけるメイクスパンの最小化のみを考慮した場合の最適解は 943 である<sup>(4-19)</sup>。最初の作業開始時間 0 を午前 8 時と設定し、単位時間 1 当たりの時間を 1 分（1 時間につき 60 コマ）と定めることで、現実的な工場を想定した。JSP は組合せ最適化問題であり、機械数とジョブ数によって全生産計画数が決定する。N 台の機械と K 個のジョブの場合、組合せの数は  $(K!)^N$  通り存在する。つまり、対象問題は  $(10!)^{10} \cong 3.96 \times 10^{65}$  通り存在する。従って、進化計算手法の適用が必要となる。

以下に、各パラメータを示す。

### (1) 共通パラメータ

探索回数 : 100,  $Iter_{not\_improved}$  : 100, 初期生産計画数  $N_a$  : 32, 試行回数 : 30,  $S$  : 0.5,  $C_1$  : 0.6,  $C_2$  : 0.1, TL 長の修正率  $\alpha$  : 0.1, CYCLEMAX : 30,  $N_g$  : 1,  $N_t$  : 1,  $N_s$  : 2,  $MSlimit$  : 960, 初期 TL 長 : 10~50 を 10 刻み, 重み係数  $w_1$  : 0.1~1.0 (0.1 刻み), 重み係数  $w_2$  : 0.1~0.9 (0.1 刻み), 重み係数  $w_p$  : 0.2,

### (2) 改良型並列 RHPSO パラメータ

スレッド数 : 1, 2, 4, 8, 16.

なお、本研究では重み係数を変更した際メイクスパンと二次エネルギーコストの影響度を等しくするため、第一項と第二項を正規化し、目的関数値がおよそ 100 になるように設定している。シミュレーションプログラムは、インテル Xeon E5-2660 (2.20GHz, 8 コア×2) のサーバ上の C 言語 (gcc ver 4.4.7 (CentOS ver. 6.9), OpenMP) 及び GLPK (Ver. 4.64)<sup>(4-21)</sup>を利用して構築した。

表 4.1 10 機械 10 ジョブ問題の設定情報

ジョブ	設定情報 (工程順序, 処理時間, 消費電力量)									
1	(8,62,10)	(9,24,5)	(6,25,15)	(4,84,10)	(5,47,5)	(7,38,15)	(3,82,25)	(1,93,10)	(10,24,10)	(2,66,5)
2	(6,47,5)	(3,97,15)	(9,92,10)	(10,22,5)	(2,93,25)	(5,29,25)	(8,56,10)	(4,80,15)	(1,78,15)	(7,67,15)
3	(5,45,5)	(8,46,15)	(7,22,10)	(3,26,10)	(10,38,10)	(1,69,15)	(5,40,15)	(4,33,20)	(9,75,15)	(6,96,5)
4	(5,85,10)	(9,76,5)	(6,68,10)	(10,88,25)	(4,36,20)	(7,75,25)	(3,56,5)	(2,35,15)	(1,77,20)	(8,85,10)
5	(9,60,5)	(10,20,10)	(8,25,5)	(4,63,10)	(5,81,5)	(1,52,15)	(2,30,25)	(6,98,10)	(7,54,20)	(3,86,15)
6	(4,87,20)	(10,73,15)	(6,51,5)	(3,95,10)	(5,65,20)	(2,86,5)	(7,22,5)	(9,58,5)	(1,80,10)	(8,65,15)
7	(6,81,5)	(3,53,15)	(8,57,10)	(7,71,10)	(10,81,5)	(1,43,15)	(5,26,15)	(9,54,10)	(4,58,5)	(2,69,15)
8	(5,20,5)	(7,86,15)	(6,21,10)	(9,79,20)	(10,62,20)	(3,34,10)	(1,27,5)	(2,81,5)	(8,30,5)	(4,46,5)
9	(10,68,15)	(7,66,5)	(6,98,10)	(9,86,15)	(8,66,10)	(1,56,15)	(4,82,10)	(2,95,15)	(5,47,5)	(3,78,15)
10	(1,30,5)	(4,50,15)	(8,34,15)	(3,58,10)	(2,77,5)	(6,34,15)	(9,84,5)	(5,40,10)	(10,46,10)	(7,44,10)

## 4. 5. 2 シミュレーション結果

表 4.2 に、従来手法である AS 近傍のみを適用した RHPSO と、改良型並列 RHPSO の AS 近傍に対して TE 近傍を加えた提案手法 4-1 と、CB 近傍を加えた提案手法 4-2 と、TE 近傍及び CB 近傍の両方を加えた提案手法 4-3 による初期 TL 長を 30 とした際の目的関数値の平均値を示す。なお、 $w_1 = 1$  のときの 99.263 という目的関数値は、最適メイクスパンである 943 が得られている数値である。Ave.列は、重み係数 $w_1 = 0.1\sim 0.9$ に対する目的関数値の平均値を表している。表中太字は、全手法の中で最も良い値を示す。表 4.2 より、全提案手法の目的関数値の平均値は、従来手法の目的関数値の平均値よりも優れていることを確認した。特に、提案手法 4-2 は $w_1 = 0.3\sim 0.5$ において、最も優れた解を導出し、提案手法 4-3 は $w_1 = 0.3\sim 0.5$ を除く全ての重み係数 $w_1$ で最も優れた解を導出することを確認した。更に、提案手法 4-3 は、全重み係数 $w_1$ の平均値も最良であることを確認した(太斜字)。尚、10 機械 10 ジョブのベンチマーク問題に対して、文献(4-5)では、HPSO が、TL 長が 40 の時のみ最適メイクスパンである 943 を得たものの、RHPSO は、全ての初期 TL 長で最適メイクスパンである 943 を得ることを確認した。同様に、本論文で提案する改良型並列 RHPSO の全提案手法でも、全ての初期 TL 長で最適メイクスパンを得ることを確認している。

表 4.3 に、従来手法、提案手法 4-1、提案手法 4-2、及び提案手法 4-3 の初期 TL 長を 30 とした際の目的関数の標準偏差値を示す。標準偏差値の観点で比較すると、重み係数 $w_1 = 0.2, 0.5$ 以外のすべての重み係数 $w_1$ で従来手法が最も優れていることを確認した。表 4.2 と

表 4.2 従来手法，提案手法 4-1，提案手法 4-2，及び提案手法 4-3 の  
目的関数平均値比較

$w_1$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	Ave.
従来手法	121.706	119.432	117.023	114.685	112.244	109.834	107.443	105.020	102.364	<b>99.263</b>	112.195
提案手法 4-1	121.761	119.385	117.019	114.633	112.296	109.898	107.448	104.976	102.324	<b>99.263</b>	112.193
提案手法 4-2	121.705	119.418	<b>116.979</b>	<b>114.597</b>	<b>112.217</b>	109.866	107.436	104.967	102.334	<b>99.263</b>	112.169
提案手法 4-3	<b>121.637</b>	<b>119.376</b>	117.016	114.628	112.234	<b>109.812</b>	<b>107.431</b>	<b>104.964</b>	<b>102.317</b>	<b>99.263</b>	<b>112.157</b>

表 4.3 従来手法，提案手法 4-1，提案手法 4-2，及び提案手法 4-3 の  
目的関数標準偏差値比較

$w_1$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	Ave.
従来手法	<b>0.221</b>	0.169	<b>0.150</b>	<b>0.097</b>	0.128	<b>0.092</b>	<b>0.054</b>	<b>0.050</b>	<b>0.055</b>	<b>0.000</b>	<b>0.102</b>
提案手法 4-1	0.320	0.292	0.230	0.190	<b>0.110</b>	0.163	0.090	0.114	0.124	<b>0.000</b>	0.163
提案手法 4-2	0.365	<b>0.166</b>	0.165	0.164	0.142	0.071	0.071	0.126	0.100	<b>0.000</b>	0.140
提案手法 4-3	0.277	0.279	0.171	0.183	0.141	0.110	0.072	0.138	0.099	<b>0.000</b>	0.147

合わせてこの結果を考察すると、従来手法では試行回数のほとんどで最良解ではない同一の局所解を導出していったといえる。一方で、提案手法 4-3 は、解のばらつきは従来手法に比べてあるものの、従来手法では到達できなかった質の高い解をいくつかの試行回数で導出することができているといえる。また、重み係数 $w_1$ が1のとき、つまりメイクスパンの最小化のみを考慮した場合、全ての適用手法において、試行回数すべてで最適解を導出できている一方、重み係数 $w_1$ が1以外のとき、つまり最終消費エネルギーコストとメイクスパンの最小化両方を考慮した場合、試行回数毎に解の質が異なり、ばらつきがある。これは、需給連携最適化方式で扱う問題は、一般的な JSP に比べ最適解を導出する難易度が高いことを示す。

図 4.6 に、提案手法 4-3 を適用した場合の様々なスレッド数に対する平均計算時間の推移を示す。図 4.6 は、例として $w_1 = 0.1$ とした場合の結果を示している。図 4.6 より、スレッド数を大きくするにつれ、並列処理による高速化が実現されていることがわかる。スレッド数を16とした場合には、従来手法と比較し約8.53倍の高速化が実現されることを確認した。RTS 処理において、各スレッドによる処理は、近傍生産計画数の差異が発生するため、スレッド間の待ち時間が発生する。そのため、全計算時間において、スレッド数が16倍になっても、8.52倍程度の高速化しか実現できていないことを確認した。一方、待ち時間を除いた並列計算時間を見てみると、スレッド数が2倍、4倍、8倍、16倍になると、並列化部の計算時間が1/2、1/4、1/8、1/16になっていることを確認した。

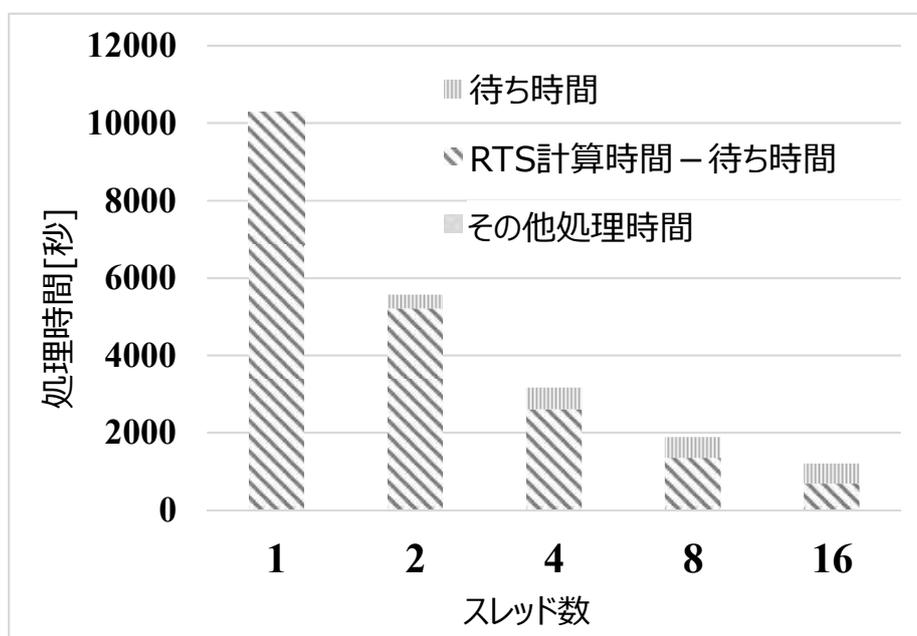


図 4.6 提案手法 4-3 のスレッド数に応じた平均計算時間推移

図 4.7 に、提案手法 4-3 を適用した場合の重み係数 $w_1$ を変化させた時のメイクスパンと二次エネルギーコストの推移を示す。図 4.7 より、 $w_1 = 1.0$ のときの生産計画は、最適メイクスパンを得ることができているが、二次エネルギーコストが高いことが確認できる。一方、 $w_1 = 0.1$ のときの生産計画は、メイクスパンの値は高くなっているものの、二次エネルギーコストを抑えることが出来ている。また、 $w_1 = 0.2$ から $w_1 = 0.8$ のときの生産計画に着目すると、全ての重み係数 $w_1$ で同じメイクスパンの値であり、且つ重み係数 $w_1$ が小さくなるにつれて二次エネルギーコストがわずかに小さくなっていることが確認できる。つまり、 $w_1 = 0.2$ から $w_1 = 0.8$ のときの生産計画の中では、 $w_1 = 0.2$ と設定したときの生産計画が最も優れていることを確認した。尚、重み係数 $w_1$ は、生産計画の最小化と二次エネルギーコストの最小化という一つの評価指標では考えられない意思決定に対して、経営状況や顧客への納期というものを総合的に判断できるように正規化した上で設定したものである。従って、実運用において何かの指標をもとにユーザが設定を行うもの、もしくは何かの結果に基づき自動的に定められるものではない。

図 4.8 に重み係数 $w_1$ を 0.1 から 0.9 まで変化させた時の提案手法 4-3 適用時の 1 時間ごとの最終消費エネルギー負荷推移を示す。図 4.8 より、重み係数 $w_1$ が 0.9 のときは電力購入単価が比較的高い昼間の時間帯の最終消費エネルギー消費量が大きくなっていることを確認した。重み係数 $w_1$ を小さくしていくにつれて、電力購入単価が比較的高い昼間の時間帯の最終消費エネルギー消費量が小さくなっていき、重み係数 $w_1$ が 0.1 のときは電力購入単価が比較的高い昼間の時間帯の最終消費エネルギー消費量は、その他の時間帯にピークシフト出来ていることを定量的に確認した。

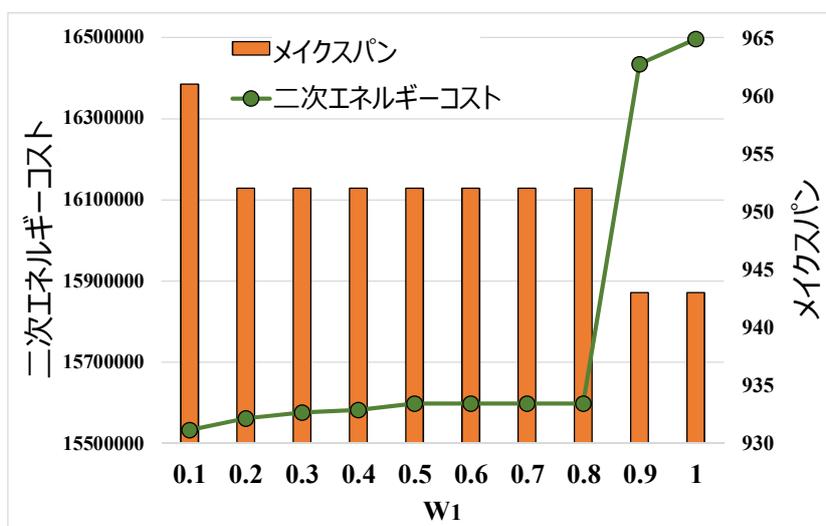


図 4.7 重み係数 $w_1$ を変化させた時の提案手法 4-3 適用時のメイクスパンと二次エネルギーコストの推移

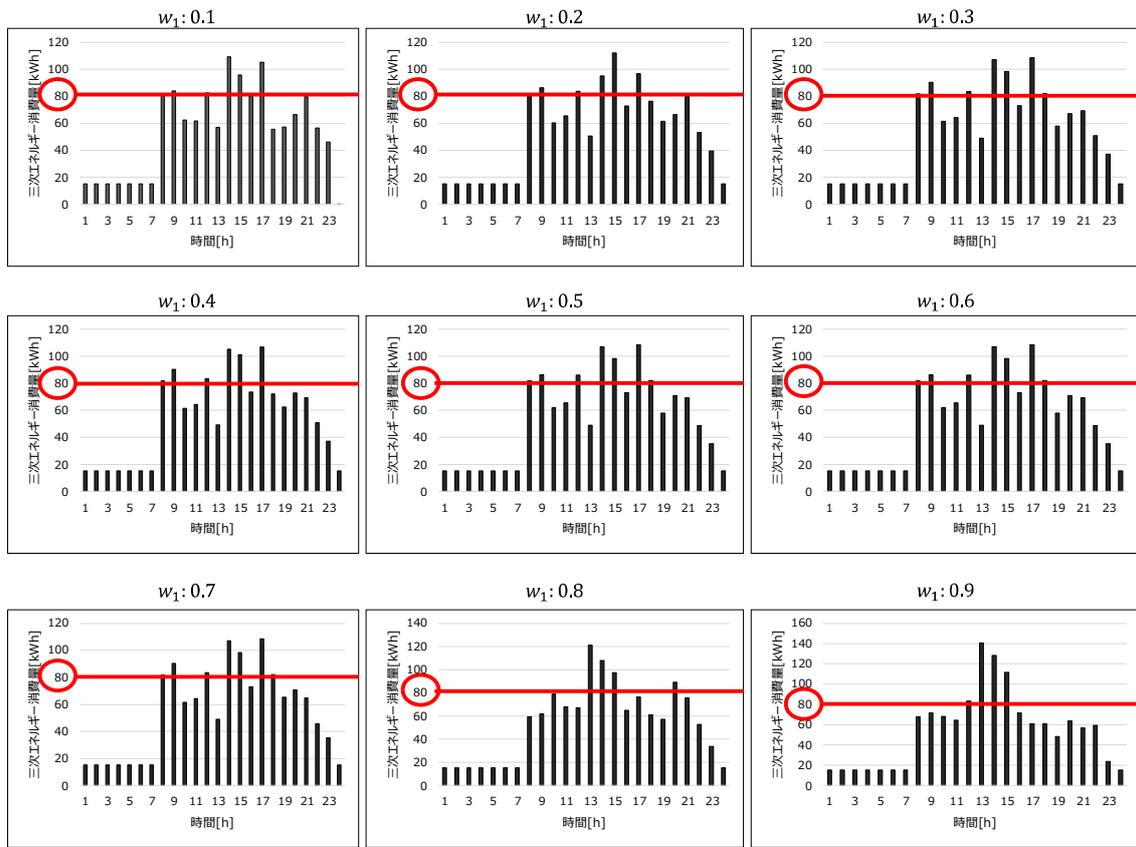


図 4.8 重み係数 $w_1$ を 0.1 から 0.9 まで変化させた時の  
提案手法 4-3 適用時の 1 時間ごとの最終消費エネルギー負荷推移

提案手法 4-3 を適用した場合の $w_1$ を変化させた時の工程の変化率及びクリティカルパスの変化比率を表 4.4 に示す。表 4.4 は、 $w_1 = 0.9$ から $w_1 = 0.1$ まで各重み係数に対する生産計画ごとの $w_1$ が 0.1 大きい生産計画に対する工程の変化率及びクリティカルパスの変化比率（「変化した工程数/クリティカルパスを構成する工程数」）を表す。図 4.7 及び表 4.4 より、 $w_1 = 0.2 \sim 0.8$ の生産計画と、 $w_1 = 0.9$ と $w_1 = 1.0$ の生産計画は、メイクスパンが同じであり、クリティカルパスを構成する工程も全て同じであることが確認できる。従って、abz6 のベンチマーク問題においては、 $w_1 = 0.1$ と $w_1 = 0.2 \sim 0.8$ と $w_1 = 0.9, 1.0$ の 3 つの生産計画に分類できるといえる。 $w_1 = 0.2 \sim 0.8$ において、クリティカルパス以外の工程の処理順序をかえることで、二次エネルギーコストを下げているといえる。

表 4.4 重み係数 $w_1$ を変化させた時の提案手法 4-3 適用時の  
工程の変化率及びクリティカルパスの変化比率

$w_1$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
工程の変化比率	13%	9%	2%	9%	4%	6%	4%	52%	10%	-
クリティカルパス の変化比率	11/15	0/13	0/13	0/13	0/13	0/13	0/13	11/13	0/13	-

図 4.9 に  $w_1 = 0.9$  と  $w_1 = 1.0$  の 1 時間ごとの最終消費エネルギー負荷の推移を示す。両ケースは、クリティカルパスは同じものの、図 4.9 より、 $w_1 = 0.9$  の最終消費エネルギー負荷は、 $w_1 = 1.0$  と比べて電力購入単価が比較的高い昼間の時間帯の負荷量が少なく、電力購入単価が比較的低い夜間の時間帯の負荷量が多くなり、エネルギー消費の観点から見て効率的な生産計画であり、結果として二次エネルギーコストが低くなっているといえる。

以上の二次エネルギーコストとメイクスパンの関係を総括すると、生産計画におけるメイクスパンと二次エネルギーコストはトレードオフの関係にあることが分かる。重み係数  $w_1$  を大きくすると、メイクスパンが可能な限り小さい一方で、ピーク時間帯の最終消費エネルギーが大きくなることで、二次エネルギーコストが大きい生産計画が導出される。反対に、重み係数  $w_1$  を小さくすると、最終消費エネルギーのピークシフトを実現し、二次エネルギーコストが可能な限り小さい一方で、その分メイクスパンが長い生産計画が導出される。この知見を基にユーザが希望する生産計画に合わせて重み係数を設定することで、ユーザの希望に合った生産計画を導出することが可能となる。

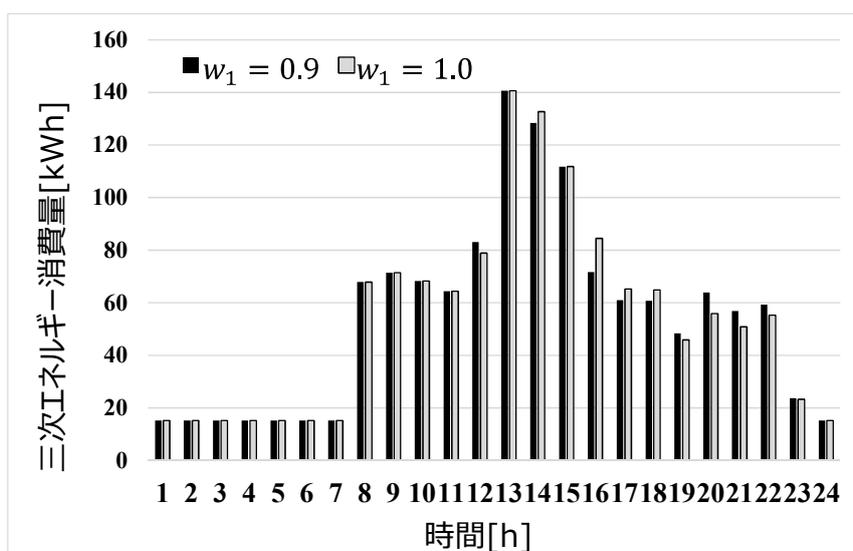


図 4.9 重み係数  $w_1$  を 1.0 と 0.1 で変化させた時の提案手法 4-3 適用時の 1 時間ごとの最終消費エネルギー負荷推移

更に、別の観点からシミュレーション結果を説明する。表 4.5 に、提案手法 4-3 による各種パラメータ  $S, C_1, C_2$  を変化させた時の、例として、 $w_1 = 1.0$  のときのメイクスパンの平均値及び標準偏差値結果を示す。表 4.5 より、提案手法 4-3 のパラメータである  $S, C_1, C_2$  を変化させても、提案手法 4-3 は最適メイクスパンを得ることができることを確認した。つまり、提案手法 4-3 のロバスト性を確認した。

表 4.5 提案手法 4-3 による各種パラメータ  $S, C_1, C_2$  を変化させた時のメイクスパンの平均値及び標準偏差値結果

S	C1	C2	平均値	標準偏差	S	C1	C2	平均値	標準偏差	S	C1	C2	平均値	標準偏差
0.4	0.1	0.1	943	0	0.5	0.1	0.1	943	0	0.6	0.1	0.1	943	0
		0.2	943	0			0.2	943	0			0.2	943	0
		0.3	943	0			0.3	943	0			0.3	943	0
	0.2	0.1	943	0		0.2	0.1	943	0		0.2	0.1	943	0
		0.2	943	0			0.2	943	0			0.2	943	0
		0.3	943	0			0.3	943	0			0.3	943	0
	0.3	0.1	943	0		0.3	0.1	943	0		0.3	0.1	943	0
		0.2	943	0			0.2	943	0			0.2	943	0
		0.3	943	0			0.3	943	0			0.3	943	0
	0.4	0.1	943	0		0.4	0.1	943	0		0.4	0.1	943	0
		0.2	943	0			0.2	943	0			0.2	943	0
		0.3	943	0			0.3	943	0			0.3	943	0
	0.5	0.1	943	0		0.5	0.1	943	0		0.5	0.1	943	0
		0.2	943	0			0.2	943	0			0.2	943	0
		0.3	943	0			0.3	943	0			0.3	943	0
	0.6	0.1	943	0		0.6	0.1	943	0		0.6	0.1	943	0
		0.2	943	0			0.2	943	0			0.2	943	0
		0.3	943	0			0.3	943	0			0.3	943	0
	0.7	0.1	943	0		0.7	0.1	943	0		0.7	0.1	943	0
		0.2	943	0			0.2	943	0			0.2	943	0
		0.3	943	0			0.3	943	0			0.3	943	0
	0.8	0.1	943	0		0.8	0.1	943	0		0.8	0.1	943	0
		0.2	943	0			0.2	943	0			0.2	943	0
	0.9	0.1	943	0		0.9	0.1	943	0		0.9	0.1	943	0

## 4. 6 4章の結論

本章では，需給連携最適化方式の主問題である最適生産計画問題に対し，良質な近傍生産計画の生成による課題2の解決に加えて，計算時間の高速化及び多品種少量生産への適用を目的として，改良近傍生産計画生成法を適用し，並列計算による計算時間の高速化を実現した改良型並列RHPSOを提案した。

10ジョブ10機械のJSP及びエネルギープラント最適運用計画問題のベンチマーク問題を用いて提案手法4-1~4-3と従来手法を比較することで，提案手法4-1~4-3，特に提案手法4-3の解の質の向上と計算時間の高速化を確認した。また，提案手法4-3は，パラメータを変更しても最適解を得ることができ，ロバストであることを確認した。

以上より，需給連携最適化方式の主問題に対する改良型並列RHPSOの適用の提案により，課題2を解決した。加えて，計算時間の高速化を実現した。更に，提案する改良型並列RHPSOのパラメータ変更が解の質に影響しないことを確認し，適用範囲が限定的であるが，多品種少量生産への適用が可能であることを確認した。

#### 第4章の参考文献

- (4-1) 川口嵩平・福山良和：「生産計画とエネルギープラント最適運用の統合による工場内エネルギーバリューチェーンの最適化」, 計測と制御 7 特集号, 第 57 卷, 第 7 号, pp. 509-513 (2018).
- (4-2) 川口嵩平・福山良和：「改良近傍スケジュール生成法を用いた改良型並列 Reactive Hybrid Particle Swarm Optimization による最適生産計画とエネルギープラント最適運用計画の統合最適化」, 電学論 D, Vol. 139, No. 12, pp.1005-1014 (2019).
- (4-3) A. M. Geoffrion, "Generalized benders decomposition," *Journal of Operation Theory and Applications*, vol. 10, no. 4, pp. 237–260, (1997).
- (4-4) D. Y. Sha, and C. Hsu, "A hybrid particle swarm optimization for job shop scheduling problem," *Journal of Computers & Industrial Engineering*, vol. 51, issue 4, December (2006).
- (4-5) S. Kawaguchi and Y. Fukuyama, "Parallel Hybrid Particle Swarm Optimization for Integration Framework of Optimal Operational Planning Problem of an Energy Plant and Production Scheduling Problem," *IEEE The 1<sup>st</sup> International Conference on Artificial Intelligence in Information and Communication (ICAIIIC 2019)* (2018).
- (4-6) M. Mastrolilli and L. M. Gambardella, "Effective Neighborhood Functions for the Flexible Job Shop Problem," *Journal of Scheduling*, Vol. 3, Issue 1, pp.3-20 (2000).
- (4-7) 飯間等・三宮信夫：「ジョブショップスケジューリング問題に対する局所探索法における新しい近傍設定法の提案」, 計測自動制御学会論文集, Vol. 39, No. 2, pp.185-191 (2003).
- (4-8) R. Ruiz, J. Antonio, and V. Rodriguez, "The hybrid flow shop scheduling problem," *European Journal of Operational Research*, Vol. 205, Issue 1, pp. 1-18 (2010).
- (4-9) S. Meeran and M. S. Morshed, "A Hybrid Genetic Tabu Search Algorithm for solving job shop scheduling problems – A case study", *Journal of Intelligent Manufacturing*, Vol. 23, pp. 1063-1078 (2011).
- (4-10) 鍋島一郎：「スケジューリング理論」, 森北出版(1993).
- (4-11) S. Kawaguchi and Y. Fukuyama, "Reactive Hybrid Particle Swarm Optimization Based Job-shop Scheduling Problems Considering Energy Management," *Proc. of IEEE TENCON 2018* (2018).
- (4-12) Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," *Proc. of the 1999 Congress on Evolutionary Computation* (1999).
- (4-13) B. Giffler and L. Thompson, "Algorithms for Solving Production Scheduling Problems," *Operations Research*, vol. 8, No. 8, pp. 487-503 (1960).
- (4-14) R. Battiti, "The Reactive Tabu Search," *ORSA Journal on Computing*, vol. 6, No. 2, pp. 126-140 (1994).

- (4-15)E. Balas, "Machine Sequencing via Disjunctive Graphs: An Implicit Enumeration Algorithm," *Operations Research*, Vol. 17, No. 6, pp. 941-957 (1969).
- (4-16)R. Zhang, S. Song, and C. Wu, "A hybrid artificial bee colony algorithm for the job shop scheduling problem," *International Journal of Production Economics* (2013).
- (4-17)山田武士・中野良平：「確率的探索と確定的探索の組合せによるジョブショップスケジューリング問題の解法」, 情報処理学会論文誌, Vol. 37, No. 4, pp. 597-604 (1996).
- (4-18)横山想一郎・飯塚博幸・山本雅人：「JSP に対する部分的な割り付け規則の適用による局所探索法の提案」, 日本機械学会ロボティクス・メカトロニクス講演会講演論文集 (2014).
- (4-19)Beasley: OR-library Website, <http://people.brunel.ac.uk/~mastijb/jeb/info.html>.
- (4-20)G. Chapman, et al., *Using OpenMP: Portable Shared Memory Parallel Programming*, The MIT Press, USA (2007).
- (4-21)GLPK ウェブサイト, <https://en.wikibooks.org/wiki/GLPK>.

## 5. 結論

本論文では、組立加工系工場の需給連携最適化方式における改良方式の提案と、生産計画最適化問題に適用するメタヒューリスティック手法の提案について論じた。これまで、組立加工系工場の需要側設備と供給側設備の両方を対象として、双方の計画を連携しながら最適化する研究が行われてきたが、これらの研究は、主問題を JSP や FSP として定式化していることから、一部の制約条件のみを考慮し、理想化されたモデルに基づいているため、現場適用に際し汎用性が低かった。また、著者らの従来研究では、需給連携最適化方式の主問題に適用したメタヒューリスティック手法では解の質に改善の余地があった。また、組立加工系工場の生産現場への現場適用に際し汎用性が低い問題を解決するため、従来研究では組立加工系工場の需要側のみを対象として生産現場の様々な制約を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化することにより精緻に再現可能な生産シミュレータを用いた生産計画最適化問題に対して進化計算手法を適用する研究が行われてきたが、これらの研究を通して、生産シミュレータを用いた生産最適化問題は、生産シミュレータの実行回数が少ない中で質の高い解を導出する必要があった。更に、生産シミュレータの処理時間を削減する必要があった。しかし、それらを考慮したメタヒューリスティック手法の適用や生産シミュレータの実現はできていなかった。

以上より、従来研究が解決すべき課題は、以下の4点である。

- 【課題1】 需給連携最適化方式の主問題に対する様々な工場に適用可能な生産シミュレータを用いた生産計画最適化問題の適用による汎用性の向上
- 【課題2】 需給連携最適化方式の主問題である JSP に適用するメタヒューリスティック手法の改良による解の質の向上
- 【課題3】 生産シミュレータの処理時間削減
- 【課題4】 生産シミュレータの実行回数が少ない条件下の生産計画最適化問題に対するメタヒューリスティック手法の適用による解の質の向上

2章では、課題1を解決するため、需給連携最適化方式の主問題を、組立加工系工場の生産現場の様々な制約を全て考慮し、生産現場における設備の特性や運転状態、製造ルールに沿ってデジタルツインとしてモデル化することにより精緻に再現可能な生産シミュレータを用いた生産計画最適化問題に改良した改良需給連携最適化方式を提案した。また、課題2を解決するため、改良需給連携最適化方式の実現に向けて、生産計画の立案とコス

ト換算を行う生産シミュレータを改良し、エネルギープラント最適運用計画問題を扱うことができ、処理時間高速化を実現可能な並列分散需給連携シミュレータを提案した。これにより、課題 1, 2 を解決した。

3 章では、課題 3 を解決するため、改良需給連携最適化方式の主問題に対して解の質向上を目的として、メタヒューリスティック手法の適用を提案した。まず、様々な組合せ最適化問題に対して有効性を確認した整数型 PBIL の生産計画最適化問題へ適用した。シミュレーション検証では、実際の組立加工系工場の一部工程をモデル化した生産計画最適化問題に対し、TS（従来手法 3-1）と整数型 PBIL を適用し、整数型 PBIL の有効性を確認した。次に、学習率の調整にかかる工数の削減という整数型 PBIL の課題を解決するための手法として、整数型 APBIL（提案手法 3-1）の生産計画最適化問題への適用を提案した。シミュレーション検証では、実際の組立加工系工場の一部工程をモデル化した生産計画最適化問題に対し、整数型 PBIL と提案手法 3-1 を適用し、提案手法 3-1 の有効性を確認した。次に、初期学習率 $\epsilon_0$ によらず APBIL 探索戦略を再現することができる適応関数の改良という提案手法 3-1 の課題を解決するための手法として、改良整数型 APBIL（提案手法 3-2）の生産計画最適化問題への適用を提案した。シミュレーション検証では、実際の組立加工系工場の一部工程をモデル化した生産計画最適化問題に対し、提案手法 3-1 と提案手法 3-2 を適用し、提案手法 3-2 の有効性を確認した。最後に、探索の多様化と集中化のバランスを取るという提案手法 3-2 の課題を解決するための手法として、改良整数型 APBIL-RTS（提案手法 3-3）の生産計画最適化問題への適用を提案した。シミュレーション検証では、実際の組立加工系工場の一部工程をモデル化した生産計画最適化問題に対し、提案手法 3-2 と提案手法 3-3 と RTS（従来手法 3-2）を適用し、提案手法 3-3 の有効性を確認した。これにより、課題 3 を解決した。

課題 4 を解決するため、4 章では、主問題を JSP、副問題をエネルギープラント最適運用計画問題とする需給連携最適化方式の主問題に対して、改良並列 RHPSO を用いた解法を提案した。改良並列 RHPSO では、AS 近傍に TE 近傍を加えた近傍生産計画生成法（提案手法 4-1）と、AS 近傍に CB 近傍を加えた近傍生産計画生成法（提案手法 4-2）と、AS 近傍に CB 近傍と TE 近傍両方を加えた近傍生産計画生成法（提案手法 4-3）を提案した。これらの提案手法を 10 機械 10 ジョブのベンチマーク問題に適用し、多品種少量生産に適用できる可能性があることを検証するとともに、目的関数の重み係数を変化させることにより、メイクスパンと二次エネルギーコストの減少率を調整できることを検証した。更に、並列計算による計算時間の高速化を実現したことを検証した。これにより、課題 4 を解決した。

世界的に、カーボンニュートラルの実現に向けた二酸化炭素排出量の抑制の動きが加速している。また、工場の経営的観点から、生産性のみだけでなくエネルギー消費量や環境負荷量に係るコストを含む生産コストの最小化は需要を増している。このような状況変化を考慮し、今後の課題を挙げる。

- (1) 組立加工系工場の様々な生産形態を対象とした改良需給連携最適化方式の実現
- (2) 生産期間を拡張した生産計画を対象とした改良需給連携最適化方式の実現
- (3) 組立加工系工場の様々な供給設備を対象とした改良需給連携最適化方式の実現
- (4) 生産計画最適化に際し調整する生産パラメータ(決定変数)の拡張を考慮した改良需給連携最適化方式の実現
- (5) 組立加工系工場のサプライチェーンを対象とした改良需給連携最適化方式の拡張

上記、いくつかの課題については検討を開始しており、近い将来、カーボンニュートラルの実現に向けた二酸化炭素排出量の抑制に対して、有益な結果が得られることを確信している。

## 謝辞

研究の立ち上げから現在までに渡り，研究活動に関する様々な面で手厚いご指導とご鞭撻を頂いた明治大学総合数理学部教授 福山 良和 先生に心からお礼申し上げます。

本論文の作成にあたり，ご指導とご鞭撻を頂いた明治大学総合数理学部教授 福山 良和 先生，吉田 明正 先生，櫻井 義尚 先生に心からお礼申し上げます。

研究に関わる共同研究において，担当学生として精力的に研究に取り組んで頂いた明治大学時代 高橋 賢二郎さん，明治大学先端数理科学研究科ネットワークデザイン専攻博士前期課程 2 年 片桐 瑠星さん，明治大学総合数理学部ネットワークデザイン学科 3 年 任 海 晟さんに心からお礼申し上げます。

また，博士後期課程進学に対し，快くご承諾頂いた三菱電機株式会社情報技術総合研究所コネクテッドインダストリーシステム技術部 長島 勝部長，研究や投稿論文の文章構成や表現に関するご指導を頂いた元三菱電機株式会社情報技術総合研究所コネクテッドインダストリーシステム技術部インダストリアル IoTG のグループリーダー，現三菱電機株式会社名古屋製作所 F A システム第一部社会計装システム技術課 三井 聡課長，三菱電機株式会社情報技術総合研究所コネクテッドインダストリーシステム技術部インダストリアル IoTG 所属 佐藤 隆臣さん，高橋 賢二郎さんに心からお礼申し上げます。