



Title	最適設計のための組合せを伴う解空間の構造分析に関する研究
Author(s)	戸井, 誠人
Citation	大阪大学, 2023, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/91927
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

博士学位論文

最適設計のための
組合せを伴う解空間の構造分析に関する研究

戸井 誠人

2023 年 1 月

大阪大学大学院工学研究科

目次

第 1 章	緒論	1
1.1	研究の背景と目的	1
1.2	本論文の構成	3
第 2 章	最適設計における解空間の構造分析の必要性	5
2.1	緒言	5
2.2	最適設計	5
2.2.1	システムの設計と最適化	5
2.2.2	設計問題の数理的な形式化	7
2.2.3	数理計画法の適用	8
2.2.4	最適設計の適用範囲の複雑化	9
2.3	設計問題における意思決定の在処	10
2.3.1	最適設計における意思決定の局面	10
2.3.2	複数の評価指標と競合	12
2.3.3	多目的設計問題への対応方法	13
2.3.4	意思決定の難しさと解空間の構造分析の役割	15
2.4	関連研究の動向	17
2.4.1	解空間の構造分析法と最適設計への応用	17
2.4.2	多目的最適設計での意思決定支援法	18
2.4.3	本研究の焦点: 組合せを伴う多目的最適設計	19
2.5	結言	20
第 3 章	組合せを伴う解空間の構造分析法	21
3.1	緒言	21
3.2	解空間の構造分析による意思決定	21
3.2.1	スキーマによる組合せを伴う解空間の構造表現	21
3.2.2	解空間構造の分析に基づいた意思決定の姿	23

3.3	解空間の構造分析法の基本構想	24
3.4	多目的最適化による代替案の生成と解空間構造の定義	26
3.4.1	多目的最適化	26
3.4.2	個別問題に応じたスキーマモデルの選択	27
3.5	クラスタリング法による代替案の構造化とビルディングブロックの顕在化	28
3.5.1	代替案の類似性の定量化	28
3.5.2	階層的クラスタリング法の活用	31
3.6	構造分析と結果の体系化	34
3.6.1	スキーマモデルに基づく構造分析	34
3.6.2	意思決定分岐図による分析結果の体系化	35
3.6.3	構造分析の意思決定への効果	37
3.7	結言	38
第 4 章	生産スケジューリング計画への適用	39
4.1	緒言	39
4.2	フローショップスケジューリング問題と最適設計	39
4.3	順列構造問題のスキーマモデルの構築	40
4.3.1	スキーマモデル A: Design Structure Matrix による方法	41
4.3.2	スキーマモデル B: ジョブの処理位置による方法	43
4.4	スキーマモデルに合わせた類似性尺度の選択	45
4.5	目的関数空間の可視化によるビルディングブロックの解釈	48
4.6	事例：順列フローショップスケジューリング問題	50
4.6.1	概要	50
4.6.2	問題設定と分析の準備	50
4.6.3	類似性尺度選択に関する調査	51
4.6.4	リンケージ手法選択に関する調査	55
4.6.5	解空間の構造分析結果	59
4.6.6	考察	59
4.7	結言	67
第 5 章	グローバルプロダクトファミリ設計への適用	69
5.1	緒言	69
5.2	グローバルプロダクトファミリの最適設計	69
5.2.1	グローバルプロダクトファミリ	69
5.2.2	最適設計問題の構築	70

5.2.3	定式化	72
5.2.4	グローバルプロダクトファミリ設計のための多目的最適化手法の構築 . . .	82
5.3	選択構造問題でのスキーマモデルの構築	83
5.4	多数評価指標でのビルディングブロックの解釈	84
5.4.1	ロジスティック回帰による分岐理由の考察	84
5.4.2	意思決定分岐図を用いた情報の整理	87
5.5	事例：自動車のグローバルプロダクトファミリ設計	88
5.5.1	概要	88
5.5.2	問題設定と計算条件	89
5.5.3	解空間の構造分析結果	91
5.5.4	考察	92
5.6	結言	103
第 6 章	モジュール化設計への適用	105
6.1	緒言	105
6.2	数理最適化によるモジュール化設計法とその課題	105
6.2.1	製品アーキテクチャとその意義	105
6.2.2	Design Structure Matrix を用いたモジュール化設計法	107
6.2.3	最適設計問題としてのモジュール化	108
6.2.4	モジュール化設計における競合と選定の難しさ	110
6.3	ネットワーク構造問題におけるスキーマモデルの構築	112
6.4	解空間構造の特徴に基づくビルディングブロックの解釈	113
6.4.1	解空間構造の特徴とその分析方法	113
6.4.2	Branching DSM を用いた分岐理由の考察	114
6.4.3	意思決定分岐図を用いた解空間構造の特徴の管理	117
6.5	事例：空調機のモジュール化設計	118
6.5.1	概要	118
6.5.2	問題設定と計算条件	119
6.5.3	解空間の構造分析結果	120
6.5.4	分析結果の考察	123
6.6	結言	127
第 7 章	組合せを伴う解空間の構造分析の適用における課題と展望	129
7.1	緒言	129
7.2	組合せを伴う解空間の構造分析法の適用条件と課題	129

7.2.1	提案手法での仮定と手法適用の条件と限界	129
7.2.2	構造分析法の効果と課題点	132
7.3	組合せを伴う解空間の構造分析法の最適設計への展開	134
7.3.1	組合せ最適化問題の種別に着眼した展開	134
7.3.2	最適設計の結果の評価方法に着眼した展開	136
7.4	結言	137
第 8 章	結論	139
	参考文献	143
	謝辞	153
	研究論文・発表リスト	155

第 1 章

緒論

1.1 研究の背景と目的

設計とは、人間が必要としている機能や役割を実現するため、1つの人工物として具体化していく過程として定義することができる。人間が必要としている機能や役割である「問題」に対して、それを実現する実体である「解」を見つけ出す過程こそが設計であるとすれば、設計はある種の問題解決である。今日の複雑な社会背景や課題意識では、設計者が認知すべき対象の全体像は膨大であり、全貌を一括して取り扱うことはもはや不可能であるため、複雑で大規模になった広範な内容をうまく処理していく必要がある。設計を進めていく際には、抽象的な解から始めて、段階的に具体的な解へと展開させ、最終的な解に収束させるという方式をとることは少なくなく、適宜視点や範囲を切り替え、部分を処理することになる。したがって、効果的に設計を進めていくには各部分における設計、つまり問題解決を段階的に行いながら、その過程で相互を摺り合わせることで、より大きな意味での問題解決を進めていくことになる。

製品やシステムの設計において、設計問題を数学的な表現へと書き換えることができれば、その見通しを明瞭にすることができる。その代表的な方法は、数理最適化を活用する最適設計である。最適設計とは、設計条件の下で、ある目的に沿って最適化をはかる設計である^[1]。最適設計では、対象の特徴や重要な部分を抽象するモデルを構築し、設計において決定すべき主要なパラメータを設計変数として選択する。安全性や有用性などを確保する制約条件を設定し、設計の優劣を定量的に判断する基準となる目的関数を設定する。このように設計問題を数学的な形式に書き換えることを定式化と呼び、問題そのものを最適化問題と呼ぶ。最適化問題を解くとは、制約条件を満たしながらも目的関数を最小化あるいは最大化する設計変数を決定することに相当する。ただし、設計問題として焦点を当てる対象に直接的な定量化の難しいものが含まれていることは稀ではなく、数学的な記述に馴染まない部分があることに注意されたい。数理的に記述できた部分に対しては、最適な内容を合理的な手続きによって定めることが可能となるため、強力なツールとして産業界で広く用いられている。

最適設計は、問題の範囲を設計変数や制約条件として明確に記述する特性上、その範囲は明瞭であるため、数学的な探索アルゴリズムを活用することで最適化問題の解あるいは解の集合を一意に定めることができる。しかし、ある設計問題に対して最適化問題を定式化することは、元々の設計問題からある部分問題を切り出すことに対応しているため、設計問題の捉え方に呼応してその最適化問題の形式も変化し、その解もまた変化することに注意しなければならない。つまり、見通しが明瞭であるのは数学的な形式の中であって、そもそもどのような形式にするかという点は設計者に拠るところが大きい。例えば、数理最適化で用いるモデルは操作や評価をより正確に行うために、できるだけ実体に忠実な方が望ましいように見える。しかし、過度に高価なモデルは操作や評価や解の探索の簡便さが損なわれてしまう。最適化を行うためには、本質的な部分を捉えた上で、その他を簡略化したモデルを用いることも必要であるため、モデルの忠実度の塩梅は設計者の判断に依存している。さらに、モデル化だけでなく、定式化にも意思決定の局面が潜んでいる。例えば、設計ではコストや品質、時間的な要求などのさまざまな評価指標を用意して、多角的に判断することになるが、評価指標1つをとっても、目的関数として設定するのか、許容できる閾値を設定して制約条件として設定するのかなど、さまざまな方式があり、そもそも何を設計変数や制約条件、目的関数にするのかについても判断が求められる。したがって、最適化問題として部分問題を考え、解を導出することは、ある段階で設計が可能になるという点で重要な役割を果たしている一方、最適化問題の形式によって切り出された、潜在的な設計解の部分集合を導出しているに過ぎず、ある種の概念によって規定された限定的なものであることに注意しなければならない。最適化問題の解は定式化の条件下では優れていることが保証されているものの、定式化の外に存在する要素は全く加味していない。また、そもそもモデル化や定式化が適切であるかどうかを保証するような規範は存在しないため、最適化結果を見ながら適宜修正を行うことが必要不可欠である。このように、「最適」解といえども、設計者の判断による定式化に限定された中での限定的な最適に過ぎないという事実から、最適化そのもので設計を完結させることは不可能であり、最適設計を通して優れた設計に至るには、最適化をどのように使うのかという意思決定の局面を合理的に進めることが必須であると考えられる。

最適設計では、設計者が部分問題をどのように定義するかという判断の部分で最適化問題として含める範囲を限定するが、計算機の性能向上や最適化技術の発展に起因し、近年ではその範囲が広がってきている。最適設計を行う場合は、先述のようなモデルの粒度に関するトレードオフのもと、求めたい解の水準やそれに向けて使える手段や時間的な制約を加味しつつ、適切なモデルと最適化手法の組合せを選択する。複雑で大規模な問題に対しても強力な計算機とアルゴリズムによって解くことができるのであれば、部分問題として最適化問題を定式化する上での考え方も変わってくる。典型的な例としては、目的関数の取り扱いによる方式の違いが挙げられる。本来、設計とは目的指向の営みであり、複数の評価指標を同時に達成することを目指す。古典的な最適化技法の場合は、目的関数が複数になると、それらを別々に扱う個別問題を考えたり、重み付けによって

ひとまとまりにしたりするなど、数理的に現実的な時間で解が求まるように前処理を行う必要があった。一方、多目的遺伝的アルゴリズム^[2]に代表されるような進化計算技術^[3]の発展に伴って、複雑な多目的問題であっても一網打尽に有望な解の集合、すなわちパレート最適解を導出することができるようになってきている。したがって、進化計算技術の発展は、多目的最適化問題として部分問題を定式化した上で、その元で得られる有望かつ多様な代替案の中から、意思決定者が嗜好にあった解を選ぶという方式を可能にした。このようなパラダイムは“design by shopping”と呼ばれている^[4]。この目的関数の取り扱い方式の例が示すように、最適化技術の発展により大規模で複雑な問題に対して、モデル化や定式化を数理的な操作が可能なレベルに調整する難しさは軽減しつつあるとみなすことができるだろう。従来の決めうちにしたモデル化や定式化と比較しても、より広い意味での解の候補を吟味できるため、より大局的な最適解に至ることが期待できる。しかしその一方で、最適化の結果として出てきた解もまた大規模で複雑であるため、それらを理解する部分で結果的にその難しさを負担することになる。複雑で大規模なモデルはその全貌を瞬時に把握することは難しく、そこから得られた結果についても、なぜその結果が優れているのか、重要な部分はどこなのかといった結果の解釈を単純な比較検討から行うことは容易ではない。

上述のような課題意識を受け、多目的最適設計での意思決定による判断の部分を支援する手法として、多目的設計探索^[5]や Trade Space Exploration^[6]が提案されているものの、これらの手法は設計変数や目的関数が連続である問題に限られた手法である。設計において対象が組合せ型をとる場合は少なくなく、システムのアーキテクチャ設計^[7]や組み立て手順の設計^[8]、生産のスケジュール計画^[9]等はすべて組合せを伴う設計問題である。組合せを伴う問題においては、代替案の関係性は離散的であり、また、等価解と呼ばれる目的関数は類似しながらも設計変数の構成が大きく異なる解が多数存在するため、従来の手法をそのまま用いることはできない。そこで本研究は、組合せ型の多目的最適設計を前提に、解空間の構造を分析する術を構築することで、意思決定者の解の選定と解釈を合理的に支援するための枠組みを提案することを目的とする。最適設計を対象にしているが、その部分問題の切り出し方は問題の形式に依存することを受け、本研究ではまず基礎理論としての解空間の構造分析法を提案し、手法を用いた意思決定の構想について述べた後に、いくつかの具体的な設計問題を念頭におきながら、その展開について述べ、事例への適用を通して有効性の検証を行う。

1.2 本論文の構成

以降の本論文の構成は以下の通りである。

第2章では、最適設計における解空間の構造分析の必要性について論じる。最適設計の具体的な進め方を整理して、意思決定の在処について述べ、本研究の着眼点を明確化する。組合せ型最適設計での意思決定を難しくする要因を指摘し、解空間の構造分析の意義や期待される効果について述

べる。また、関連する研究のレビューをもとに、本研究の立ち位置を明らかにする。

第3章では、第2章で述べた背景に基づいて、組合せを伴う解空間の構造分析法を提案し、具体的な方法と手続きについて述べる。組合せを伴う解空間構造を操作するための数学的な記法を導入するとともに、階層クラスタリングに基づく構造分析法の構想を述べた上で、具体的な実装のための3つのステップを示して、それぞれの詳細な手続きや細部の設定についての基本的な考え方を順に論じる。

第4章では、ベンチマーク問題として生産スケジューリング問題を例に引き、解空間の構造分析法を適用する一連の流れを確認する。事例では、順列フローショップスケジューリング問題を取り上げ、手法の有効性を確認する。スケジューリング問題の解空間構造を表現する記法として2つのモデルを導入し、モデルによる解釈や意思決定の違いを検証することで手法の効果と特徴について論じる。また、階層的クラスタリングでの非類似度の定義やリンケージ手法の選択による結果への影響は、規模の異なる2つのベンチマーク問題を通して検証された。

第5章では、解空間の構造分析法のグローバルプロダクトファミリ設計への展開を示す。グローバルプロダクトファミリ設計とは、製品系列とサプライチェーンネットワークとの相互作用を加味した統合的設計の枠組みである。グローバルプロダクトファミリの最適設計は多数目的の混合整数計画問題として定式化され、評価指標が多様であることから解空間構造の解釈が困難である課題を抱えている。混合整数計画問題の解空間構造を表現するモデルを新たに導入した上で、ロジスティック回帰分析を応用した多数評価指標問題での解釈法を新たに構築する。事例では、自動車のグローバルプロダクトファミリ設計問題を例に引いて、提案する方法の有効性を検証する。

第6章では、解空間の構造分析法の応用として製品のモジュール化設計への展開を示す。モジュール化設計は Design Structure Matrix(DSM) と呼ばれるネットワークモデリングツールを用いることとし、その解空間構造を表現するモデルを新たに提案する。モジュール化設計での評価指標は抽象度が高く、代替案の理解や選定としての基準として用いることができない。構造分析の結果を DSM モデルに還元することで、モデル化の仮定に立ち戻った具体的な解釈を可能にする分析方法を提案し、家庭用空調機の事例を通してその有効性を検証する。

第7章では、4章から6章での事例への展開を総括し、提案した枠組みをより広範な最適設計に展開していく上での課題について論じる。提案した構造分析法の前提や仮定を整理することで、個別事例の結果から課題設定の妥当性と構造分析の効果を考察し、手法の適用条件と現状での限界を克服するうえでの課題と可能性を論じる。

第8章では、各章の内容を要約した上で、本研究の成果を総括する。

第 2 章

最適設計における解空間の構造分析の 必要性

2.1 緒言

本章では、本研究の前提である最適設計に焦点を当て、その枠組みとしての考え方と意思決定の在処について述べ、解空間の構造分析の必要性について論じる。まず、最適設計の進め方の形式や方式について、設計過程や数学的な取り扱いの観点から論じ、設計における数理計画法の役割を整理する。さらに、最適設計での意思決定者の役割について論じ、意思決定の局面を合理的に進めることの難しさから、解空間の構造分析の必要性とその意義について指摘する。最後に、関連研究の動向を整理し、それらの課題と本研究の立ち位置を明らかにする。

2.2 最適設計

2.2.1 システムの設計と最適化

設計とは、人間が必要とする機能を 1 つの製品あるいはシステムとして具体化していく過程である^[10]。設計対象の規模は時代とともに大きくなってきており、それを構成する物理現象もまた大規模で複雑なものになってきている。1.1 節では、そのような複雑な対象を扱う設計問題では、全貌を一括して取り扱うことが難しいことから、部分問題へと分割して設計を進めざるを得ないことを指摘した。

複雑な設計対象を取り扱う際に、システムという抽象の力は効果的である。システムの階層性を念頭に置くと、部分問題へと順次視点を切り替える設計方法の効果を理解することができる。システムの複雑さは、「単純でない仕方で相互に関連しあう多数の部分から成り立つこと」として定義され、一般にその複雑さは階層的な形態をとると考えられている^[11]。ここでの階層的な形態とは、相互に関連するいくつかの下位システムから成立し、その下位システムもまたそれぞれがシステムか

ら成り立ち、最も低い基本的な下位システムに至るまで順次そのように連なっていることを指す。サイモン^[11]はこのような階層的に組織化されたシステムを準分解可能システムと呼び、階層はその行動を非常に単純化する特性、すなわち準分解可能性の特性を持っていると指摘した。準分解可能性は、次の2つの特性を備えている。(a) 準分解可能システムでは、構成要素である各下位システムの短期的な行動はほかの構成要素の短期的な行動からはほぼ独立している。(b) 長期的には、いかなる構成要素の行動も、ほかの構成要素の行動にただ集合的に依存するに過ぎない。さらに、システムの理解可能性に関しては、構成要素の内部構造に見られる階層の高頻度のダイナミクスを、構成要素間の相互作用での低頻度のダイナミクスから区別することで階層性を規定しているとすれば、システムを階層として表現しても、比較的わずかな情報しか失われまいと考えられている。したがって、階層的に組織化したシステムを用いることで、非階層なシステムと比べて効率よくその複雑さを処理することができる。これらの準分解可能性と理解可能性の観点を設計問題を対象に考えると、上位のから下位のシステムに視点を移すことは範囲を絞り込んで部分問題を考えること。つまり、設計の粒度を細かくすることに対応する。逆に、下位から上位に対象を移すことは、粒度の粗さを伴いながらも設計範囲を広げることや、それらの総合を考えることに対応する。

システムの設計において、設計問題を数理的な形式で記述することができれば、その中で合理的に設計を進めることができる。その代表的な方法は、対象とする問題を最適化問題として定式化する最適設計である。数学的な枠組みを用いるには、個別の問題が置かれている状況の中から本質的な内容を数学的な形式で記述する必要がある。最適設計では、本質的な内容を抽象したモデルを作成し、設計対象を設計変数、制約条件、目的関数という形式で数学的に記述する。このような形式に整えることを定式化と呼び、構築された問題は最適化問題と呼ばれる。最適化問題は数理計画法と呼ばれる技法によって求解され、目的関数を最小化、あるいは最大化するように探索アルゴリズムが最適解を探索する。このように、最適化問題を解くことによって行われる設計は最適設計と呼ばれる。

最適設計は、数理的に馴染まない部分は捨象せざるを得ないデメリットは存在するものの、最適化問題として定式化してしまえば、定式化の前提や背景からは独立した探索アルゴリズムによって、合理的に解を定めることができる。最適化は特定の解、あるいは解の集合を導き出すことができるため、ある部分問題で設計が可能になるという点で重要な役割を果たしている。しかし、モデル化や定式化自体の自由度は高く、どのようなモデルを用いるのか、さらには、どの要素を設計変数や制約条件、目的関数に割り当てるのかなど、最適化を使って何を解くのかという上位の問題では、唯一の答えが存在しない。そもそも何が問題であるかを理解し、その問題がどのように解決できるのかを明確にしておかなければならず、設計者の判断に委ねられている。このような設計者あるいは意思決定者と最適化計算の関係は2.3節で改めて論じる。本節では、定式化の詳細については2.2.2項、解の探索については2.2.3項と、最適設計での具体的な方策について述べる。

2.2.2 設計問題の数理的な形式化

ここまででは、数理的な形式という言葉を用いて最適設計の特徴について論じてきたが、ここで改めて最適設計での問題、つまり最適化問題の形式を定めておく。最適化問題は一般に、次のような形式で定式化される。

$$\begin{aligned}
 &\text{find} && \mathbf{x} = [x_1, x_2, \dots, x_n]^T \\
 &\text{minimize} && f_i(\mathbf{x}, \mathbf{p}) \\
 &\text{subject to} && g_j(\mathbf{x}, \mathbf{p}) \leq 0 \\
 &&& h_k(\mathbf{x}, \mathbf{p}) = 0
 \end{aligned} \tag{2.1}$$

ここで、 $\mathbf{x}$ は設計変数ベクトル、 $\mathbf{p}$ は設計パラメータのベクトルであり、 $f_i(\mathbf{x}, \mathbf{p})$ は目的関数、 $g_j(\mathbf{x}, \mathbf{p}) \leq 0$ は不等式制約条件、 $h_k(\mathbf{x}, \mathbf{p}) = 0$ は等式制約条件を表している。

設計変数 $\mathbf{x}$ は設計の内容に対応する変数であり、システムの何を決定すべきかを表している。一般に、設計変数の数が多くなると求解が難しくなる。設計変数の表現形式は対象の特性に依存し、連続変数ベクトルだけでなく、離散 (整数) 変数ベクトルや形状関数などを用いる場合もある^[12]。

目的関数 $f_i(\mathbf{x}, \mathbf{p})$ はシステムの評価指標に対応し、優劣を定量的に判断する役割を担う。基本的な形式では目的関数は設計変数 $\mathbf{x}$ のスカラー値関数であるが、設計とは目的指向の取り組みであることから、優劣の判断を 1 つの指標で行うことはほとんどなく、設計で考えるべき評価指標を複数取り入れる場合がある。最も直感的な方法は、重み付けによる総合評価である。例えば、線形加重法が挙げられる。線形加重和法は式 (2.2) に示されるように重みを付け合わせて単一目的の最適化問題に帰着させる手法である。

$$F = w_1 f_1 + w_2 f_2 + \dots + w_r f_r \tag{2.2}$$

線形加重和法では、式 (2.2) を最大化あるいは最小化した解として得られる結果と、設計者が設定する重み w_i との間には単調な関係がないことから、 w_i の調整に対して解の挙動が敏感であり、適切な重みを設定することが難しい。また、解集合が凸集合ではない場合ではうまくいかないなど、手軽である一方で課題も抱えている。最適化問題を定式化する際に、式 (2.1) のように複数の目的関数をそのまま設定することもでき、その場合は多目的最適化問題と呼ばれる。

$g_j(\mathbf{x}, \mathbf{p})$ と $h_k(\mathbf{x}, \mathbf{p})$ は制約条件であり、システムの実行可能性を規定する要素である。多くの場合、システムの果たすべき機能に関する制約等が等式制約として定式化され、システムの性能に関する制約が不等式制約として定式化される。等式制約は常にすべてが有効である。設計において複数の性能評価を行う場合、すべてを目的関数として取り扱うのではなく、その評価指標に対して、確保すべき水準などを決めた上で、不等式制約条件として制約条件式に含めることが行われることもある。

設計パラメータ p は対象システムの動作環境を表しているパラメータベクトルである。数理計画問題では設計パラメータを直接扱うことはなく、探索アルゴリズムは p を定数とみなして処理する。システムが置かれている環境や、サブシステムとしての階層的な関係性を踏まえると、設計において変動を想定する部分は対象の捉え方に応じて多様である。つまり、最適化問題ではパラメータベクトル p として固定された要素もより上位の問題では変動を想定できることも少なくない。設計変数ベクトル x とパラメータベクトル p の区別は定式化での判断によることから、 p を明示的にしておく本項の記法は、設計問題ならでの取り扱いともみれる。

実際の設計では随所に二者択一の選択が含まれていて、関連する部分の問題は組合せ型となる。そのような部分に対して最適設計の枠組みを用いる場合は、2進数変数や整数変数を用いて設計変数が定式化される。このような問題は、組合せ最適化問題と呼ばれる。例えば、サプライチェーンの設計^[13]では、生産施設の設立や設備の導入、輸送経路などの決定は二者択一の選択であり、2進数変数を用いることで設計変数を定義することができる。また、設計の上流段階では、設計対象をグラフやマトリクスなどの集約的なモデルを用いて抽象し、グラフ理論を応用した定量評価やモデル操作が行われる場合がある。これもまた、離散的なグラフ構造を計画する組合せ最適化問題である。例えば、製品や組織、プロセスのアーキテクチャ設計では Design Structure Matrix(DSM)と呼ばれるネットワークモデリングツールが用いられ、最適化によるモデル操作手法が提案されている^[14]。設計対象物の寸法や形状についても多くの場合は連続変数を用いて表現されるが、製品系列内での共通化や標準化を背景に、あらかじめ定められた規格の中から選択を行う組合せ問題として定式化される場合もある。このように、対象を数理的に操作する際に組合せ数を導入する場面は各種の設計問題でみられ、本研究ではそれらをまとめて組合せ型設計問題と称することにする。組合せ最適化問題では、その離散的な性質から勾配情報を用いた探索を行うことができない上に、設計変数の数が増えると、候補数が爆発的に増加することから、厳密解を得ることが難しい特徴がある。

2.2.3 数理計画法の適用

最適設計では、最適化問題をひとたび定式化すれば、そのモデルに応じて適切な最適化手法を選択し、これを解くことになる。しかし、あらゆる最適化問題を解くことができる万能なアルゴリズムは存在せず、最適化問題の特性を把握して、適切な手法を用意することが必要不可欠である。最適化手法の選択には、設計変数が連続変数であるか離散変数であるかや、制約条件の有無、対象モデルの非線形性などの視点から判断する^[12]。代表的なアルゴリズムとしては、シンプレックス法や双対法に代表されるような代数的関係に基づいた方法、ニュートン法やペナルティ法のような解析的關係に基づいた手法などが挙げられ、コンピュータの数値計算能力を生かした計算的手法が開発されている^[15]。また、近年は大規模かつ複雑化する最適化問題に対して正確ではなくとも実用に耐えうる最適解を現実的な処理時間で求めるため、メタヒューリスティクスが用いられてきている。例

例えば、生物の進化における自然淘汰の仕組みを模倣した遺伝的アルゴリズム (Genetic Algorithm, GA)^[16] や、金属材料の焼きなまし現象を模した模擬焼きなまし法 (Simulated Annealing, SA)^[17] などが広く用いられている。

式 (2.1) のように複数の評価指標を目的関数として扱う場合は、多目的最適化問題として定式化され、多目的最適化問題を解くための技法を用いることになる。目的関数は多くの場合互いに競合しており、最適化の結果はパレート最適解集合と呼ばれる非劣解集合である。解の競合関係については 2.3.2 項で詳しく説明する。パレート最適解集合を得るための簡便な方策の 1 つは、線形加重和法を用いながら、重みをさまざまに設定して複数回最適化を行うことである。しかし、2.2.3 項で指摘した観点から依然として扱いの難しさを抱えている。その他の手法として、 ε 制約法が挙げられる。複数の目的関数の中で 1 つの目的関数 f_p に注目し、その他の目的関数は制約条件 $f_i(\mathbf{x}) \leq \varepsilon_i, (i = 1, 2, \dots, l, i \neq p)$ として扱うことで単一目的最適化に帰着させ、 ε を調整しながら複数回計算を行うことでパレート最適解集合を獲得する。近年では、計算機の性能向上などに起因して、メタヒューリスティクスが主流である。例えば、多目的遺伝的アルゴリズム (Multi-objective Genetic Algorithm, MOGA)^[18] は広く用いられている。遺伝的アルゴリズムの多点探索を活かして、パレート最適解に広く分布させるように探索を行うことで、一網打尽にパレート最適解集合を導出することができる。

2.2.4 最適設計の適用範囲の複雑化

1.1 節では、最適設計における数理モデルの構築は設計者の判断に委ねられ、求める結果の精度とモデルの扱いやすさとのトレードオフの中で最適化に用いるモデルで含める範囲を定めることを述べたが、実際のところ、万能な最適化手法は存在しないことから、最適化手法の探索能の限界に規定されることが多い。一方で、今日の計算機の性能向上やメタヒューリスティックに代表される最適化手法の発展に伴い、複雑な問題を定式化したとしても、有意な時間で解を導出することができるようになってきている。設計過程としての段階的な詳細化を前提にすると、複雑な問題を解くことができるということは、部分問題としての切り出す対象の範囲を広げたり、より現実 に即した形へと深めたりすることができることを示している。このような最適設計でのモデル化や定式化の複雑化の動向は、次の 2 つの視点から整理することができる。

(a) モデルの忠実度の変化

より現実 に即した結果が欲しい場合は忠実度の高いモデルを構築することになる。内容の複雑さと呼応して、伴う現象も複雑化すれば局所的な非線形性が高まるため、数値計算モデルも複雑化し、計算コストが高くなる。例えば、最適化すべき要素は、流体、熱、電気系、制御系など、複数の領域にまたがっていることは少なくなく、それらは複雑に作用し合っている。これらを一括して扱う場合の最適化技法は複合領域最適化と呼ばれている^[19]。このよ

うな複雑化ではやはり、最適化手法の限界を見極めた上で、モデルの粒度を定めたり、代替モデルの活用や分解と統合によるアプローチなど、取り組み方を工夫することが求められる。

(b) モデルの範囲の変化

モデルの境界を広げることでより多くの概念を一括して扱うことができるので、広い意味での最適性を向上させることが期待できる。例えば、設計の過程に着眼すると、設計の過程のモデルはさまざまなものが考えられているが、基本構成として、(1) 製品企画、(2) 概念設計、(3) 基本設計、(4) 詳細設計といった段階に分けて整理される^[20]。詳細設計では、部品の寸法や表面特性など、解くべき問題が非常にはっきりしている。モデルの数学的な形式を想定しやすく、またその解釈も簡単であるため、最適設計は設計の下流部の詳細設計を中心に実用化されており、市販の解析プログラムも多く見られるようになってきている^[1]。最適設計の適用範囲をより上流へと切り替えていくと、その抽象度は高まり、最適設計としての難しさが顕在してくる。具体的には、上流の設計問題は多くの場合、多数の目的の中で右か左かを決めるような選択的な構造を抱えており、多目的かつ組合せ型の形式をとる。また、どのようなモデルを構築すべきか、どのような評価を行うべきかという選択肢が増大する。最適化問題を定式化し、高度な技術で最適化したとしても、その解釈の段階に課題を抱えることになる。

最適設計の枠組みは、設計問題から数理的に記述できる部分を抜き出したものにすぎないことを加味すれば、最適化結果の意味や効果はモデルの厳密さに紐づいている。最適設計を通して設計を進行させるには、設計者がその限定性を理解し、構築したモデルや最適化問題の定式化に立ち戻って結果を深く理解する必要がある。つまり、結果をさらなる設計に活かすには、ある種の知見獲得が必要であり、その方法も定式化した最適化問題の粒度や範囲に呼応して個別に考えなければならない。本研究の事例検証では、特に (b) 設計過程としての範囲の変化に焦点を当てて事例を展開する。第4章では、解くべき問題がはっきりしていて、評価指標からの解釈が容易な生産スケジューリング問題を取り上げる。第5章ではグローバルプロダクトファミリー、第6章では製品モジュール化設計へと、より抽象的かつ曖昧さを含む事例へと対象を展開していく。

2.3 設計問題における意思決定の在処

2.3.1 最適設計における意思意思決定の局面

最適設計は、数理で扱える部分問題に対して合理的な設計を進めることができる強力な手段である一方、その結果はあくまで定式化によって規定された限定的な解の1つにすぎず、最適化だけで設計を完結することはできないことを述べてきた。改めてシステム設計の進行における、最適設計の進行を整理し直すと、図2.1のようにまとめることができる。2.2.1項で述べたシステム設計の

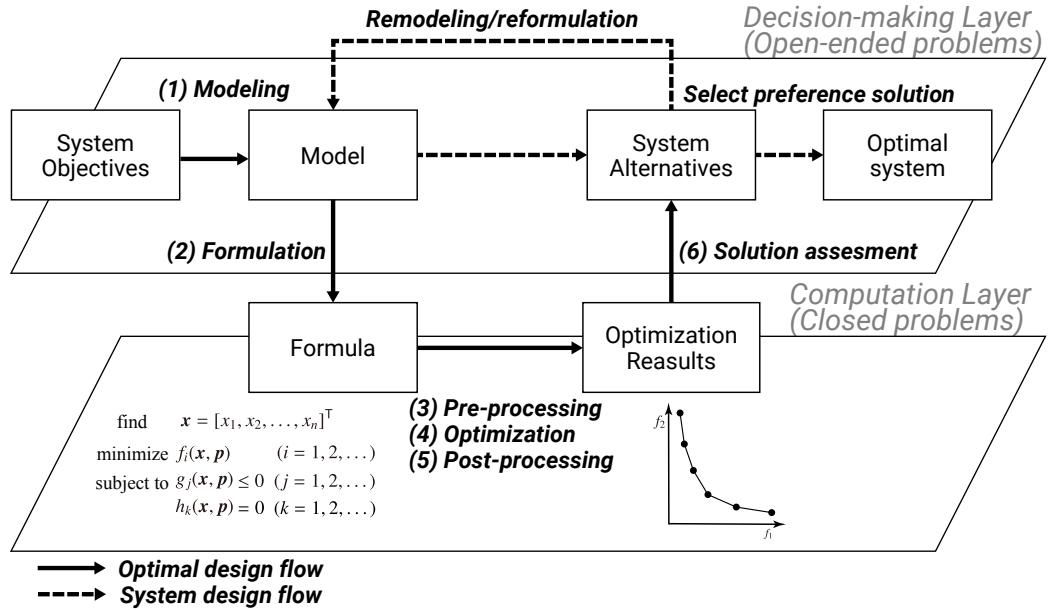


Fig. 2.1: Optimal design flow in the system design flow.

段階的な詳細化を前提とするならば、設計や計画の進捗に呼応しながら、システム代替案の生成と評価を繰り返して、設計解の表現やその評価に用いるモデルをより実際に近い詳細なものに漸次切り替えていくことにより、システムの実現にたどり着いていくことになる。詳細化のある一段階に焦点を当てて、最適設計を行う場合、図 2.1 に示されるように、(1) 対象のモデリング、(2) 問題の定式化、(3) プレプロセッシング、(4) 最適化計算、(5) ポストプロセッシング、(6) 結果の評価といった手順で進められる。

前述の通り、システムの設計において全貌を一括して取り扱うことはできないため、対象の一部を部分問題、ここでは最適化問題として切り出すことになる。まずは、求めたい解の詳細度や最適化として現実的な時間で求解できる方法があるかを鑑みつつ、内容や条件を簡略化したり、理想化したりすることで、対象をモデリングする。設計変数や目的関数、制約条件を設定し、式 (2.1) の形式で最適化問題を定式化する。

構成した最適化問題を最適化のアルゴリズムに載せるには、求解を行いやすくするために 2.2.3 項で示したような前処理 (プレプロセッシング) を行う場合がある。例えば、最適化問題 (主問題) に対して、補集合になる最適化問題である双対問題を解く処理方法がある。最適化計算が行われたのち、後処理 (ポストプロセッシング) を行い、最後に結果の評価を行う。

より大きなシステム設計の流れに焦点を当てると、最適化結果が導き出されたとしても、それで終わることはなく、同じく定式化の前提やパラメータ設定などを見つめ直して、結果の解釈を行うことになる。設計の段階的な詳細化の過程では、最適化の結果を踏まえて、相互のすり合わせを繰り返し、問題解決を進めていくことになるので、結果の評価の後、知見を抜き出して再度モデリングや定式化を改めて行ったり、最も好ましい解を選択して設計を具体化して次の設計過程へ進んだ

りする。例えば、多目的最適化問題を解いたとすると、パレート最適解と呼ばれる多様かつ有望な解の集合から、意思決定者の嗜好に合わせて選好解を定めることになる。選好解の選定には根拠が必要となるが、まずは結果からそれらの傾向を読み取ったり、そもそもの定式化の前提に立ち戻って問題を深く理解したりすることが効果的である。

意思決定者と最適化という関係に着目すれば、図 2.1 に照らし合わせて考えると、数理計算層と意思決定層の 2 つを想定することができ、それぞれで取り組む先を「閉じた問題」と「開いた問題」として区別して設計問題の構造を整理することができる。図 2.1 の下層に示される、数理的な形式での計算による問題解決では、解くべき問題があらかじめ与えられていて、解が何かについては、問題が用意された時点で実は 1 つあるいは複数に定められている。事実、2.2.3 項で紹介したような最適化手法はモデルの背景とは完全に独立していて、アルゴリズムによってあらかじめ用意されたものを探し当てるような構造になっている。コンピュータは閉じた状態空間の中で演算を繰り返す以外のことはできず、このような問題は「閉じた問題」と呼ばれる。一方で、これまで述べてきたように、最適設計は意思決定の側面を含んでいて、最適化問題をどのように定義するのかから議論を始めなければならないし、結果をどのように次に活かすのかについても考えなければならない。これらの問題は唯一の解が存在しない、「開いた問題」と呼ばれる。

最適設計は開いた問題であり、数理に基づくとは言え、その進行の形は 1 つに定まらない。2.2.4 項で示したように、詳細設計で解くべき問題がはっきりしている場合はまだしも、対象の抽象度が高まれば、その扱いにくさは高まる。抽象的に定式化した問題で得られる結果もまた抽象的であり、どのように結果を扱うのかという部分は、それ自体が大きな問題になっている。したがって、意思決定者の存在は最適設計を進める上で無視することができない要素であり、このような厄介な問題 (wicked problem) では、最適化技術の探究だけでなく、意思決定の局面を合理的に進めることが重要であることがわかる。

2.3.2 複数の評価指標と競合

最適設計において意思決定者の判断が必要となる代表例は、考えうる評価指標の間に競合関係が存在していて、一方を改善しようとする、他方が改悪するようなトレードオフの関係が潜んでいる際の取り扱いである。トレードオフの関係は、数理的な形式のもとでは、より明確に理解することができる。図 2.2 の左側に示されるように、制約条件は設計変数空間における実行可能領域を定める。図 2.2 では 2 つの目的関数 f_1 と f_2 を共に最大化したい場合を想定しているが、目的関数空間上の可能領域の右上側の境界部分では、 f_1 を改善しようすると f_2 が悪化し、逆に f_2 を改善しようすると f_1 が悪化するような、トレードオフの関係を視覚的に把握することができる。このような一連の解の集合をパレート最適解集合と呼び、それだけでは特定の解を定めることができず、意思決定者の判断が必要になる。

実行可能領域と可能領域の対応関係が複雑であることは稀でない。例えば、異なる設計変数を

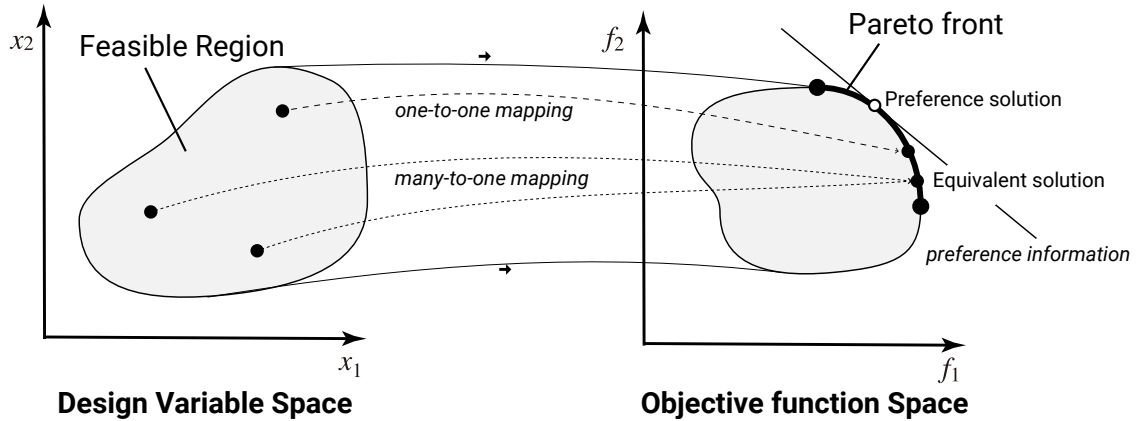


Fig. 2.2: Mapping between design variable space and objective function space.

持つ複数の解がほとんど同じ目的関数値をとる場合がある [21, 22]。そのような解は、等価解と呼ばれており、図 2.2 に示されるように、設計変数空間の多数の点が目的関数空間の 1 点に写像される [23]。特に、組合せ型の設計問題では、その離散的な性質から等価解が多数存在することがある。等価解の存在によって、最適化過程で局所最適解に陥りやすいことが指摘されており、進化計算を応用した、マルチモーダル多目的最適化問題が提案されてきている [24, 25, 26]。

パレート最適解は競合の関係によって導かれる集合であるので、唯一の解を定める場合には各評価項目の関係を調整する別の視点を導入しなければならない。嗜好情報をもとに意思決定者が選択した解は選好最適解、あるいは選好解と呼ばれる。例えば、選好情報として無差別曲線のようなものを定義すれば、図 2.2 のように接点から選好解を定めることができる。このように、多目的最適化では目的関数のトレードオフ関係から唯一の優れた解が存在しないため、追加で選好解を定めるための根拠や基準が必要になり、意思決定者の判断が必須である。

2.3.3 多目的設計問題への対応方法

最適設計は開いた問題であり、その内部に存在する閉じた問題、つまり最適化計算の部分を意思決定者がどのように扱うかについての万能な方策は存在しない。評価指標を同時に処理する多目的最適設計では、最適解として獲得できるものはパレート解集合と呼ばれる集団であるから、選好解を合理的に定めるためには、何らかの操作が不可欠である。多目的設計問題での意思決定の方式は次の 3 つの種別に大別される [27]。

(a) A priori approach

意思決定者があらかじめすべての条件を決定しておく方法。複数の目的関数の重要度や優先度を設定しておき、そのもとでの定式化を行う。最適化計算を実行することで、パレート最適解集合ではなく、特定の解を獲得することを目指す。典型的な方法は式 (2.2) を用いた重み付け線形和による目的関数値のスカラー化である。単一目的の最適化問題に帰着されるた

め、数理的な扱いが容易である一方で、重要度をうまく設定することが難しく、意思決定者が有効な非劣解を見落とす危険性が潜んでいる^[28]。このアプローチは選定の難しさというよりはむしろ、定式化に難しさがある。

(b) A posteriori approach

多目的最適化によってパレート最適解を生成した後、意思決定者が選好解を定める方法。多目的最適化問題をそのまま扱う解探索は数理的に難しいものの、得られる解の集合は多様性に富んでおり、代替案を豊富に用意することができる強みがある^[27, 29]。このアプローチは (a) とは逆に、選定に難しさを抱える。

(c) Interactive approach

意思決定者が解決の過程に選好情報を整理、あるいは追加することで解決 (最適化) の過程に介入する方法。意思決定者の理想との偏差を最小化しつつ、満足のいく妥協解を導出することができる。厳密な最適解が得られないものの、計算コストを大幅に抑えつつ満足のいく解を得ることができる^[30, 31]。

3つのアプローチの違いは、「閉じた問題」と「開いた問題」の境界をどのように設定するのかという観点で次のように解釈することができる。図 2.1 に示されるように、最適設計での2つの層は、(2) 定式化と (6) 結果の評価で接続されている。(a) a priori approach の場合は、図 2.1 の (2) 定式化に開いた問題としての性質を取り入れており、(6) 結果の評価の部分には意思決定の余地は殆ど残されていない。逆に (b) a posteriori approach の場合は (2) 定式化ではシステム的设计において曖昧な部分を残しながら、(6) 結果の評価の部分で開いた問題としての性質を取り入れている。さらに、(c) interactive approach は最適設計のすべてを開いた問題として扱っている。一般にはあらかじめ情報を決め打ちにしておく (a) a priori approach よりも (b) a posteriori approach の方が広範な解空間の探索が可能であると考えられるため、より優れた解に至ることが期待できる。

メタヒューリスティクスの応用などの近年の多目的最適化技術の発展により、複雑な多目的最適化問題のパレート最適解集合を一網打尽に生成することができるようになってきており、(b) a posteriori approach をとる場合が主流になりつつある。しかし、多目的最適化手法に関する多くの研究は、より優れたパレート最適解集合を短時間で獲得することを目指しており、選定や解釈の過程は依然として意思決定者に委ねられている。したがって、意思決定者は、目的関数を初めとするさまざまな根拠に基づいて、多目的最適化によって導出された選択肢を比較検討し、選好解を決定する。このような代替案生成と選定による最適設計のパラダイムは “design by shopping” と呼ばれている^[4]。“design by shopping” では、どの結果を選択するかにかかわらず、多くのトレードオフの重要な規定を特定し認識することで、問題そのものに対する洞察が育まれると考えられている。意思決定者の嗜好が設計の終始で一貫することは稀であり、選定の過程で嗜好が当初より改善されていくことが期待され^[32]、モデリングや定式化の改善に寄与すると考えられている。

用意された代替案から望ましいものを選ぶとは言っても、そう簡単なことではない。そもそも最

適化は定式化の外にあるものは全く考慮しないため、最適化結果をそのまま鵜呑みにすることはできない。2.3.1 項で述べたように、設計を進める上では、選好解を踏まえて相互のすり合わせを繰り返し、問題解決を進めていくことになるため、選好解の選定には、導き出されたパレート最適解を深く分析するだけでなく、最適化問題として何を捨象しているのかなど、数理的な形式の周辺も含めて、設計問題への洞察を深めておかなければならない。さらに、最適化技術の進歩によって複雑な問題を解くことができるようになった一方、導き出される結果もまた複雑であり、その理解や解釈も一筋縄ではいかない。代替案のすべてを横に並べて、単純な比較検討を行うことではその背景に潜む複雑な状況を把握することはできず、新たに意思決定者を支援する術が必要となる。

本研究では (b) a posteriori approach を念頭に置き、多目的最適化法によって得られた代替案の集合と意思決定者の関係に焦点を当てる。

2.3.4 意思決定の難しさや解空間の構造分析の役割

多目的最適化の結果から選好解を定めてゆく過程で、意思決定を合理的に行うことが重要であることを述べてきた。意思決定には、代替案の詳細を理解することはもちろんのこと、最適化問題ひいては設計問題そのものについても深い洞察を獲得することが重要である。本項では、最適設計の諸要素をシステム思考の冰山モデル^[33]を用いて改めて整理し直すことで、難しさをもたらす要因を指摘し、解空間の構造分析の役割を論ずる。

冰山は全体の 10% しか水面に出ておらず、90% は水中にある。その 90% に海流が作用し、氷山の先端部の挙動を作り出している。冰山モデルではシステムの挙動を同じように考え、システム要素を海上に氷山の先端部の見えている部分と、海面下にある常に見えていない部分とに区別して考える。見えていない海面下の部分についてアプローチすることで、システム挙動の根本原因を探ることができると考えられている。

システム思考の冰山モデルではいくつかのレベルに分けて思考を整理する。水面に出ている層は事象レベルであり、通常認識することができる部分である。事象レベルで観察される対象は単純な調整で対処できることは多い一方、冰山モデルはこのレベルだけで解決できると考えないようにするものでもある。事象レベルを観測しているといくつかのパターンを見出すことができ、その層をパターンレベルと呼ぶ。パターンの原因は何かを問うと、そこには何らかの構造を想定することができるため、パターンレベルの下には構造レベルが存在している。さらに、構造ではパターンに影響するさまざまな因子とそれらの相互関係が存在しているため、より深い層にはメンタルモデルが潜んでいる。システムそのものが成り立つ根底には、その前提や仮定、信念や価値観が潜んでいる。これらのいくつかの層の元で、根源的な変化を起こすには、一連の層のより深い部分にアプローチする必要がある。

上述のような考え方に倣った、最適設計の冰山モデルを図 2.3 に示す。事象レベルとして意思決定者が観測可能であることは、最適化結果として得られる設計変数や目的関数の値だけでなく、最

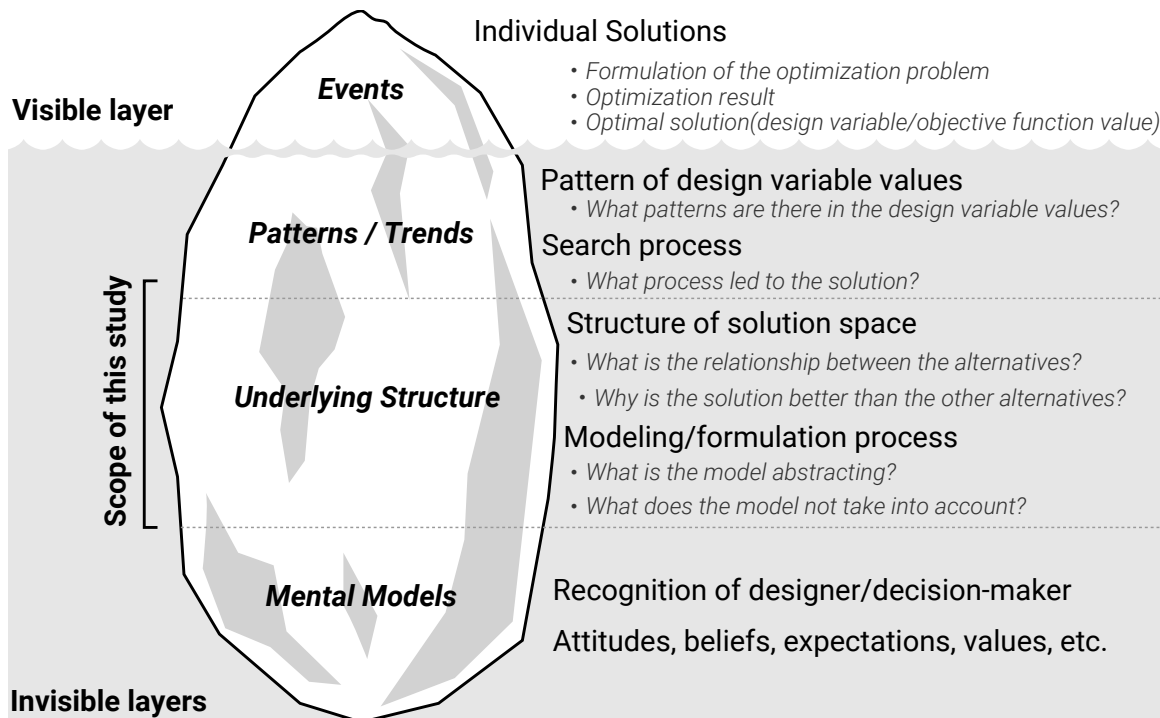


Fig. 2.3: The iceberg model of optimal design.

適化問題としての数学的形式も含まれている。観測可能な結果そのものもだけは、それらの関係や傾向を簡単には理解できないことは前述の通りであり、より深い部分にアプローチするため、本項では海面下に存在する要因を明示化し、それらの関係性を整理する。事象レベルの下には、設計変数のパターン、解探索過程が潜んでいる。さらに深くには解空間構造、構造をもたらすモデル化と定式化、さらには、モデル化と定式化の前提・仮定、メンタルモデルが存在していると考えられる。

最適解は、最適化手法に基づく解探索によって導かれる。どのようなアルゴリズムが用いられているにしろ、解探索で最適解に至るには、評価と比較に基づいた繰り返し操作が行われている。遺伝的アルゴリズムの考え方を借りるのであれば、そのような過程は自然淘汰に準えて理解でき、どのようにして最適解が導かれたか、という解探索の経緯が、設計変数や目的関数の値に傾向やパターンを生み出す。

解探索の過程を規定しているのは、その一層下にあたる解空間の構造である。解同士の優劣はそもそもにある解空間に規定されているため、探索の過程も代替案の関係を説明する解空間構造に規定されることになる。遺伝的アルゴリズムでは優劣を規定する重要な部分をビルディングブロックと呼び、その構造をスキーマを用いて説明している。つまり、最適解がなぜ優れているのか、という問いについては、構造の側面から説明可能であると考えられる。さらに、解空間の構造はモデルによって定義されるため、その下にはそもそものモデル構築での前提や仮定に規定されている。これまで述べてきたように、設計者がどのようにモデル化、定式化するかによってその最適化の様相は大きく変化する。また、そういった前提や仮定の根底には、もう一層下に潜んだメンタルモデ

ルに左右され、設計者がそもそも何を認識して、何を認識できていないのか。さらには設計者の意識や信念、期待、価値観の影響を受けている。

氷山モデルが示すように、最適化結果の背景には、さまざまな要素が複雑に絡み合いながら作用している。事実、最適化では、試験的に問題設定でのパラメータを変更させたり探索方法を変えてみたりした際に、結果が大きく変わることは稀ではない。海面下の要素を明示的に扱わなければ、事象レベルで観測できる結果を理解することができず、最適設計の枠組みを設計に活かす際に、課題を抱えることになる。

本研究では海面下の第2層である、解空間の構造に焦点を当てる。解空間の構造分析法を提案することで氷山モデルの海面を下げて解の解釈を可能にし、意思決定の過程を支援することを目指す。解空間の分析を行うことで、多目的最適化で得られる代替案の集合に対して、それらの特徴を明らかにすることができるだけでなく、その下層に位置するそもそもの問題構築に根ざす前提や仮定をより明確に把握することで、設計問題の性質を理解することにも繋がると考えられる。

2.4 関連研究の動向

2.4.1 解空間の構造分析法と最適設計への応用

2.3節では意思決定の難しさに焦点を当て、解空間の構造分析の必要性を述べてきた。最適化計算の中での探索効率を高めることを指向した、分析する手法がいくつか提案されている。

Design Space Exploration(DSE) は設計において変動を想定できる変数 (基本的には設計変数) で定義される設計空間に対して分析を行う手法である^[34]。DSE の適用により、変数が局所的あるいは包括的に及ぼす影響について洞察を得ることができる。DSE は広範な概念であり、数理最適化も一種の DSE とも位置付けられている^[35]。

DSE 研究は、設計空間の理解し、探索することで設計概念を探索することを目指しているが、いくつかの研究では、計算コストの高い最適化問題に対して計算負荷を下げるため、設計空間上の関心領域 (Regions of Interest, RoI) を特定することを目的としている。高価なモデルを用いて最適設計を行う場合、近似法などを導入して ROI を特定できれば、特定した部分絞って詳細に探索するなど、効率的に設計解を獲得することができるようになる。代表的な方法は、応答局面法やサロゲートモデルと呼ばれるメタモデルを構築することで、解空間の構造を予測する方法である^[19, 36]。また、メタモデリングの際に、同じサンプルサイズであれば、設計空間のサイズや次元が大きくなるにつれて精度が悪化するため、RoI をうまく特定できるように次元削減を行う場合もある^[37]。多目的最適化のために設計空間を部分空間に分解する手法^[38] や、計算幾何学的手法とサポートベクターマシンを用いて、設計空間を安全領域と失敗領域に分割する手法^[39]、自己組織化マップを応用したサンプリングによる手法^[40] などが提案されている。メタモデルを用いることは、最適解を導出することはもちろん、設計空間を理解する上でも効果が見込め、探索による情報獲得を目的

にモデル構築を行う場合もある [41, 42, 43]。

また、DSE の他の研究では、概念設計などの上流では ROI の特定が重要であることを受けて、非常に広範な設計空間を理解する、あるいは設計コンセプトの選定を支援するために、設計空間を広く探索する方法論が提案されている [35, 44, 45, 46]。近年では、データ駆動型アルゴリズムへの関心の高まりにより、メタモデルとは別の視点で設計空間の次元削減を行うことで、設計空間の理解を狙う DSE 手法が報告されてきている。高次元情報を低次元の表現に変換することは意思決定者の認知的負荷を下げることに貢献する。例えば、設計空間のパターンを即座に理解するため、自己組織化マップや [47] 主成分分析 [48]、変分オートエンコーダ [49] を用いた次元削減と可視化を一体にした方法が提案されている。

メタヒューリスティクスを活用した最適設計を背景に、探索性能の向上のため、解空間構造の分析を活かした手法が提案されてきている。マルチモーダル多目的最適化問題では、解空間の構造に等価解の存在を仮定すると、局所解に陥りやすい性質を持つことを背景に、解空間の構造を念頭においたアルゴリズムが開発されている。例えば、多目的進化アルゴリズムの多様性維持の仕組みに焦点を当て、ニッチング戦略を拡張するアルゴリズムが提案されている [25, 26, 50]。また、設計変数空間と目的関数空間の多様性を同時に測定したり [24]、設計変数と目的関数空間の適応度共有 [51] を活用することで [52] 効果的に解空間を探索することが可能になっている。さらに、遺伝的アルゴリズムでは、解空間構造を表現するビルディングブロックに焦点を当てたアルゴリズムが提案されている。ビルディングブロックとは、優れた解に共通する部分的構造であり、スキーマ定理 [53] では、ビルディングブロックを構成することで解の性能が向上すると仮定する。ビルディングブロックを直接操作する方法としては、messy GA [54, 55] や連鎖学習遺伝的アルゴリズム [56] が挙げられる。また、block mining と呼ばれる手法では、スキーマ定理に基づき、優れた解の染色体から共通の構造を探し出したり、新しいビルディングブロックを確立したりすることを通して解探索を行う [34, 57, 58]。

これらの手法から、図 2.3 の氷山モデルで示したように、最適化のプロセスの根底には解空間の構造が存在しており、解空間構造に注目することで探索性能を高めることができると改めて理解することができる。

2.4.2 多目的最適設計での意思決定支援法

多目的最適設計での選好解を定める難しさを背景に、最適化の高性能化というよりはむしろ、意思決定者の解釈や選定を支援することを目的とした研究がいくつか報告されている。

多目的設計探索 [5] は 3 つ以上の目的関数をもつ多数目的最適化問題に焦点を当て、多目的遺伝的アルゴリズムによって獲得したパレート最適解を分析する手法である。多数目的の場合、単純なパレート面の可視化ではその傾向を理解することは難しい。解空間をデータマイニング手法によって可視化し、トレードオフ情報を整理することでパレートフロントのスイートポイントを探索する

ことを目的としている。例えば、航空機の翼形状の最適化問題に対して、自己組織化マップを用いることで 2 次元のカラーマップに整理し、設計空間に関する知見を抽出する手法が提案されている [59, 60]。これらの取り組みは図 2.3 氷山モデルの事象レベルを深く調査しようとする取り組みである。

Trade Space Exploration^[6] は、“design by shopping”^[4] と呼ばれるパラダイムを背景に、目的関数のトレードオフ関係やそれに伴う設計変数の傾向を調査する手法である。多くの研究は、可視化と探索アルゴリズムを組合せることで、後天的に選好情報を組み込む手法を提案しており、多次元データの可視化には、平行座標プロットや散布図マトリクスを標準的な手法として採用している [61, 62, 63]。Trade Space Exploration で設計変数と目的関数の関係やパターンを調査することは、氷山モデルに対応させると、そのパターン層まで踏み込んだ研究として位置付けることができる。最近の研究では、可視化した結果もまた大規模であることを受けて、パレートフロント形状を定量化する手法が提案されている [64]。パレートフロント形状指標の導入により、探索が必要ないトレードオフ領域を削減し、注目すべき解空間の部分を優先度を設定することができる。

これらの研究の目的は単に選好解を定めることに留まらず、トレードオフ関係の把握やその背景に潜むパターンや関係性を明示的にすることで、解空間、引いては設計問題そのものに対する洞察を養うことにある。2.3.3 項で述べたように、分析を進めていくにつれて、意思決定者の嗜好もまた洗練されていく。

2.4.3 本研究の焦点: 組合せを伴う多目的最適設計

本研究では、先述の通り a posteriori approach を念頭に置き、多様な代替案の集合から解を選定する段階での結果の解釈や学習を支援する手法を提案する。特に、離散的な性質から解釈が困難になることが予想される、組合せを伴う多目的最適設計を研究の対象として設定し、解空間の構造分析法を提案することで最適設計の支援を行う。

2.4.2 項で示した解空間の構造を解探索や解の選定に活用する先行研究は、暗黙的に設計変数や目的関数が連続である場合を前提としており、組合せ型の最適設計にそのまま適用する場合、次のような課題を抱える。

- 組合せ型設計問題では解空間に連続的な変化を想定することができない。同等の目的関数値を持ちながらも、組合せのパターンが大きく異なる等価解が頻繁に生じる。このため、多目的設計探索^[5]に代表される目的関数を根拠にした分析では、最適解の特徴見出すことはできず、他の解と比べて優れている要因を直接特定することができない。
- 大規模化かつ複雑な組合せ型設計問題では、組合せが重層しているため、解の単純な比較検討では内部の傾向を掴むことが困難である。Trade Space Exploration^[6]のような可視化によって設計変数の傾向を獲得しようにも、設計変数は 2 進数あるいは整数の連続であるか

ら、それらの関係やパターンを見出すことが容易ではない。代替案の相違点は組合せ論的に増大するため、全貌を把握するには認知的負荷が高く、意思決定を行う上での介入点を明確にすることが難しい。

これらの理由から、従来の方法では組合せを伴う最適設計においては選好解を定める根拠を用意したり、反映させたりすることが容易ではなく、新たな意思決定者を支援する術が必要である。

上記のような組合せ型の最適設計の課題は、計算の結果として出てきた情報をそのまま扱おうとする限界に起因している。2.3.4 項で冰山モデルを用いて整理したように、最適設計では、目に見えない層でさまざまな要素が複雑に関係している。最適化結果の解釈を行うにも、設計変数や目的関数の値やそれらのパターンだけでなく、最適設計に潜んでいる構造の側面に焦点を当てなければ、意思決定を行うための重要な事柄を明示的に扱うことはできない。そこで本研究では、解空間の構造を定義、分析することで、代替案の理解を促し、意思決定者の着眼点を明示的にする新たな分析手法を構築することを目指す。優れた設計は重要な部分的構造 (ビルディングブロック) によって構成されていると仮定^[53] し、その部分的構造の集積と意思決定の階層性を対応させた解構造分析法を提案し、最適化結果や、問題そのものへの理解と解釈を可能にする。

2.5 結言

本章では、設計問題における最適設計の考え方と立ち位置を示し、解空間の分析法の必要性について論じた。まず、システムの設計としての最適化の役割を述べ、その形式化や最適化技術についてレビューを行いながら整理した。技術の発展に起因する最適化モデルの複雑化を背景に、最適設計としての範囲の広がり述べた。次に、最適設計での意思決定者の存在とその判断の難しさを論じ、設計問題の多目的性への対応方法を意思決定者の役割という観点から整理し、本研究での立ち位置を明らかにした。設計問題に潜む背景を冰山モデルを用いて整理し、本研究で提案しようとする解空間の構造分析の必要性を論じた。最後に、関連する研究をレビューし、それらの課題と本研究での焦点や新規性を明らかにした。

第 3 章

組合せを伴う解空間の構造分析法

3.1 緒言

本章では、第 2 章で述べた最適設計における意思決定者の役割を念頭におきながら、構造分析による意思決定支援法の骨子を新たに提案する。まず、組合せを伴う解空間の構造を数学的に記述する術を導入し、提案しようとする解空間の構造分析法によって達成される意思決定支援の枠組みについて構想を述べる。次に、意思決定の中核に位置する解構造の分析法についてその全体像を述べ、それを支える技術要素を 3 つの手順に対応させながら説明する。

3.2 解空間の構造分析による意思決定

3.2.1 スキーマによる組合せを伴う解空間の構造表現

ここまで、やや漠然と解空間という概念を用いてきたが、改めて解空間という概念の指す範囲を定めておく。本研究では解空間を、有望かつ多様な代替案の集合が織りなす空間として定義する。具体的には、解空間は DSE での設計空間の考え方^[34]を応用して、解空間 W は設計変数ベクトル $\mathbf{x}$ やパラメータベクトル $\mathbf{p}$ によって生成される集合として定義する。ここでの有望かつ多様な代替案とは、設計における目的が設定されて初めて規定される概念である。目的の設定を変えればその空間の範囲や構造も異なるものの、より広範な集合である、設計変数ベクトル $\mathbf{x}$ とパラメータベクトル $\mathbf{p}$ で生成される潜在的な解空間 V の部分空間として解釈可能である。つまり、解空間 W は $W \subset V$ を満たしている。最適設計では、有望かつ多様な代替案とは多目的最適化でのパレート最適解集合として解釈されるため、代替案を獲得するための手段として最適化手法が用いられることになる。図 3.1 に示されるように、そもそもの最適設計の進め方を念頭におけば、変動を想定できる変数のうち、何を設計変数に、何をパラメータとして扱うかや、何を設計の目的に設定するのかというモデリングや定式化での判断によって解空間は限定されている。さらに、より大きい視点で考えると、そもそも設計者が何を認識しているのかという部分でも解空間が限定されている。

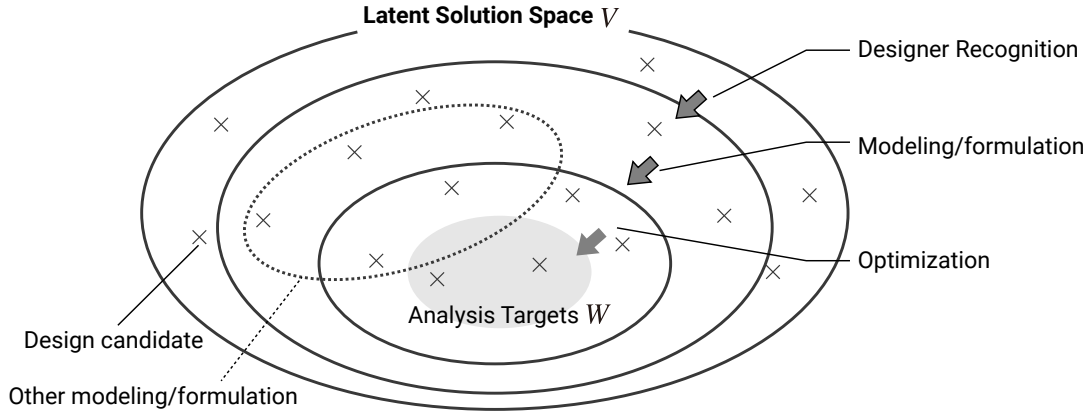


Fig. 3.1: Latent solution space and targets of proposed analysis.

本研究で分析の対象として取り上げる解空間とは、ある最適化問題の定式化のもと、最適化技法によって導かれる解の集合とし、具体的には多目的最適化のパレート最適解集合を扱うことにする。したがって、本研究での解空間の定義は図 2.2 に示されるような設計変数空間での制約条件によって規定される実行可能領域の概念とは異なることに注意されたい。図 2.2 でのパレート最適解のみが解空間の集合に含まれており、実行可能領域の大部分の解は劣解であるため、分析の対象ではない。設計変数や制約条件が変化すれば実行可能領域が変化するため、パレート最適解も変化する。また、同じ実行可能領域でも目的関数の設定によって得られるパレート最適解は異なる。このような定式化による範囲の絞り込み過程が図 3.1 での 2 段目に相当する。定式化そのものは 3.2.2 項で述べたように開いた問題であるので、定式化の方法に唯一の答えは存在しない。設計者の認識や定式化の方法としての広がりを加味した上での解の集合を潜在的な解空間 V として考えている。

本研究では、組合せを伴う解空間構造を理解するため、遺伝的アルゴリズムについてのスキーマ定理^[53]を取り上げ、スキーマを用いて解空間の構造を数学的に形式化する。以下では、その理論に基づいて提案する枠組みの前提となる考え方を説明する。図 3.2 にスキーマを用いた解空間構造の例を示す。スキーマとは、特定の文字列位置で類似性を持つ文字列の部分集合を識別する形式である。組合せ最適化問題の特定の結果を表現する記号、すなわち解空間の変数の集合に、*(don't care symbol)を加えた文字集合を用意し、それらの文字列でスキーマが定義される。例えば、ビット列を用いる場合は、スキーマは $\{1, 0, *\}$ で構成される文字列である。 $*$ は任意の記号を意味する文字であり、スキーマが表す部分集合で 1, 0 が多様に存在していることを意味する。5 つの設計変数を持つ場合は、解の全体集合はスキーマ $\{*****\}$ で表現される。スキーマの特定の部分を固定することによって解空間の部分集合を定義することができ、例えば、スキーマ $\{*1***\}$ とスキーマ $\{*0***\}$ は解空間を 2 つに分割する、補集合の関係にある部分集合である。さらに、スキーマ $\{***1*\}$ とスキーマ $\{***0*\}$ は先ほどの部分集合とは異なる分割を表し、それらを重ね合わせることでさらに細かい部分集合である $\{*1*0*\}$ や $\{*0*1*\}$ を定義することができる。つまり、 $*$ 以外の文字で固定された部分を増やすことによって、より細かな分割、より細かい部分集合を表現するこ

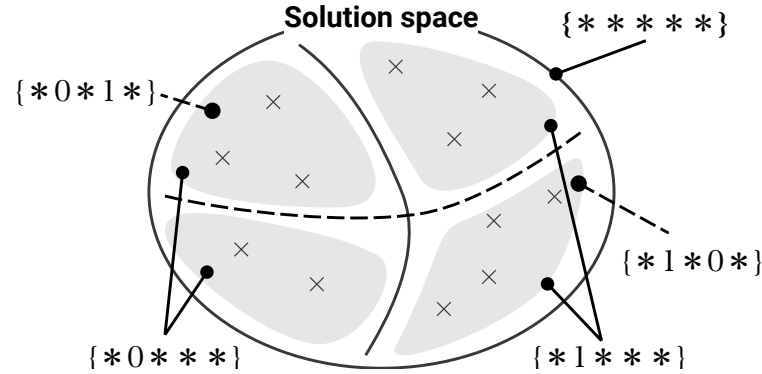


Fig. 3.2: Partitioning the solution space by schema.

とができることになる。スキーマには定義長と次数とよばれる量が定義されており、それぞれ、スキーマの * 以外の文字のうち左端と右端の距離、* 以外の文字の数を表す。

3.2.2 解空間構造の分析に基づいた意思決定の姿

多目的設計探査や Trade Space Exploration に代表される多くの意思決定支援手法は目的関数値に焦点を当てている一方、組合せ型設計問題では等価解の存在などを背景に、目的関数ベースの分析手法と馴染まない。そこで本研究では、目的関数だけでなく、解空間の構造に焦点を当てた、新たな意思決定の枠組みを構築する。スキーマ定理では、定義長が短く、次数が小さく、平均適応度が高いスキーマをビルディングブロックと呼び、遺伝的アルゴリズムの解探索はビルディングブロックの集積として解釈される。言い換えると、優れた解はビルディングブロックと呼ばれる優れた部分列を有しており、それらの集積で個々の最適解、さらには解空間の構造が形成されている。本研究ではこのようなビルディングブロック仮説を前提とし、解空間構造をスキーマを用いて分析することで、ビルディングブロックの集積過程から、最適化で特定の代替案に至った経緯や設計変数の重要部分を俯瞰的に把握することを可能にする。

2.2.1 項での複雑システムの準分解可能性とその理解可能性^[11]を念頭において、複雑な対象での問題解決、ここでは意思決定ということになるが、そのプロセスに階層性を想定する。大局を掴む判断から詳細を決める決定まで、一連した部分的意思決定の連続として構成することで認知的負荷を低減し、意思決定者の介入点を明確化することを狙う。最適化を潜在的な解空間全体をより細かい部分集合に絞り込む過程であると捉えると、解空間上でのビルディングブロックの重ね合わせが解空間の絞り込みに相当する。ビルディングの集積過程を意思決定の階層性に対応させ、図 3.3 に示されるような形で意思決定者に介入点を提示する探索木を構成する。

探索木の各分岐点では、構造特徴として、スキーマをどのように絞り込んでいるのかを記載する。スキーマの各ビットは特定の結果を表現する記号であるから、スキーマモデルの定義に応じて設計問題としての背景が存在している。例えば、サプライチェーンの最適化問題であれば、スキーマの

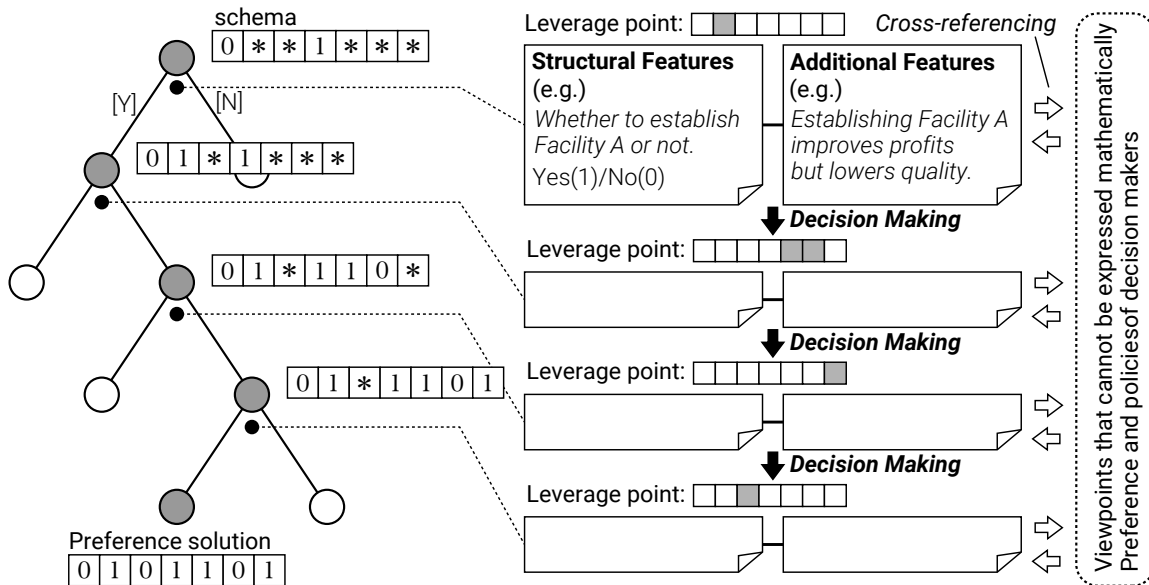


Fig. 3.3: Hierarchical decision-making concept.

ビット列である設計変数は施設や供給経路の選択に対応するだろう。右の分岐と左の分岐でスキーマを0で固定するか1で固定するかという絞り込み方の違いは、施設の設立の有無に対応する。さらに、スキーマが表す部分集合の目的関数値やその他特徴量の違いを分析し、追加特徴として記載する。最上位の分岐点から順にこれらの解空間構造に関する情報と数理に馴染まない要素や意思決定者の嗜好といったモデルの外側にある要素とを相互参照しながら意思決定を繰り返し、ビルディングブロックを積み上げることで、ある特定の選好解に辿り着くことができる。また、それぞれの分岐位置での解空間構造に関する情報はそれ自体が重要な設計知見であり、代替案の関係や問題の特徴を理解する際の着眼点が明示的であるので、膨大な情報を処理する認知的負荷を低減させることが期待できる。本章ではこのような構想をもとに、基本となる代替案の分析方法を構築する。

3.3 解空間の構造分析法の基本構想

本研究では、図 3.2 で示した、 $\{1, 0, *\}$ で構成されるバイナリー情報で定義されたスキーマを前提に解空間の構造を分析する。組合せ型の設計問題では、バイナリ変数を用いて設計変数を定義することが多く、そのままビット列として扱えば、簡単に解空間構造を設定することができる。順列の最適化などでは単純なビット列では表現できない場合もあり、*を含んだスキーマ表現への変換が必要なこともある。スキーマモデルを選択して解空間の構造を定義するにあたっては、設計変数などの数値をそのまま使うのではなく、代替案の関係性を言語化する上で重要になる要素を明示化し、意思決定者が抽象する要素を取捨選択する必要がある。具体的なモデル化方法については 3.4 節で論ずる。

ビルディングブロック仮説のもと、代替案は重要な部分列によって構成されていると考え、

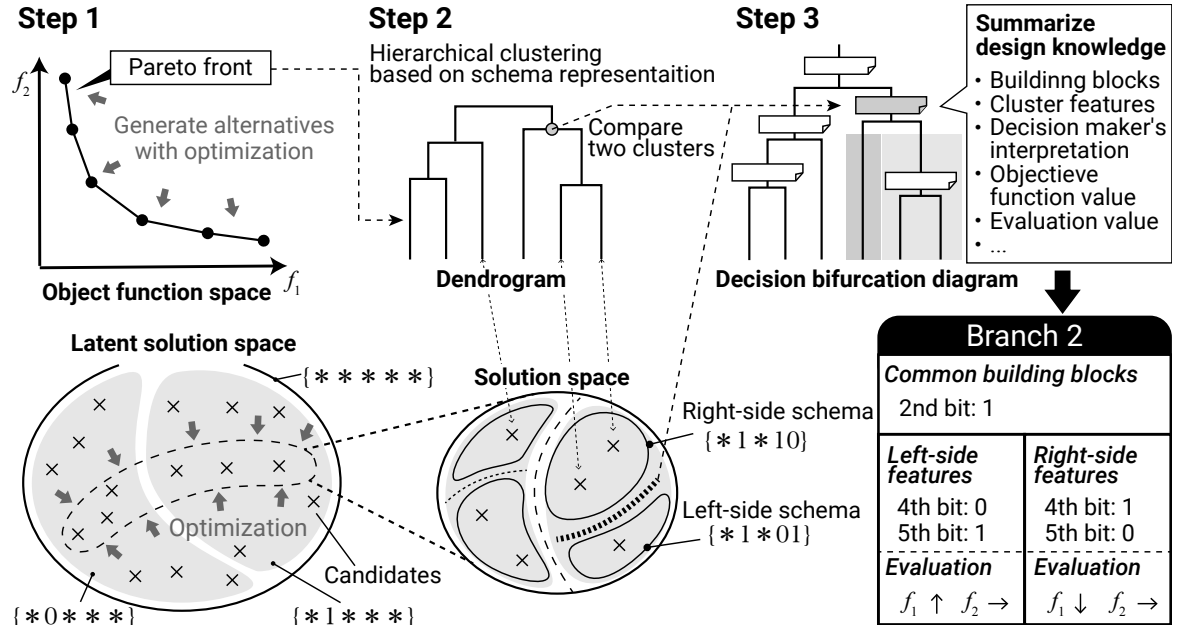


Fig. 3.4: Concept of the proposed method.

多目的最適化の場合はそれぞれの目的関数値に寄与するビルディングブロックを個々別々に仮定することができるため、パレート最適解集合に含まれるビルディングブロックは多様化することが予想される。解の特徴の理解や意思決定者の着眼点の明示化という動機のもとでは、パレート最適解集合でのビルディングブロックの分布を解析することができれば、解の重要な部分的組合せ構造を解の傾向を大きく変化させる組合せ構造の段階的な分岐として体系化することができると考えられる。以上を受けて、本研究では、代替案の集合から特定の解に至る解空間の分割プロセスをビルディングブロックの集積過程に対応させ、解の優劣や多様性に支配的に作用する部分的組合せ構造を階層的に整理することで、解空間の構造に解釈を与えることを目指す。

本研究での解空間の定義から、解空間は有望な代替案の集合であることを受けて、特定の集合を代表する部分列（スキーマの固定文字列位置）をその部分集合でのビルディングブロックとして定義する。代替案の中でのビルディングブロックの分布を通して、ビルディングブロックの集積過程として解空間構造を解き明かす。例えば、スキーマ $\{ * 1 * * * \}$ とスキーマ $\{ * 0 * * * \}$ は第 2 番目の文字列位置にのみに差異があり、解空間上では補集合の関係にあるため、2 つの集合の間には分割面を想定することができる。言い換えると、第 2 番目のビルディングブロックが分割面に関連する部分であることがわかる。特定のスキーマが表す集合を評価することで、2 つの部分集合の評価の比較から分割面を表すビルディングブロックの役割を考察する。

図 3.4 に構築する解空間の構造分析法の概要を示す。提案手法は次の 3 つの手順で構成される。

Step 1: 多目的最適化による代替案の生成と解空間構造の定義

設計対象の代表的なトレードオフ関係に着目し、多目的最適化問題を定式化、最適化を実行

することでパレート最適解を代替案として獲得する。対象とする設計問題における固有の性質を踏まえた上で、解空間構造を表現するスキーマを定義する。スキーマ表現は問題の性質に依存するが、多くの場合は設計変数に基づく。具体的には、代替案の関係性を説明するために、代替案の特徴を抽象し、それらをビット列へと変換する。次ステップではスキーマ表現に基づいて代替案の類似度を定義するため、定量評価が比較的容易な表現方法を用いるなど、代替案の類似度の設定方法を念頭に置いて定式化することが望ましい。

Step 2: クラスタリング法による代替案の構造化とビルディングブロックの顕在化

スキーマを入力に代替案の類似度を定義し、代替案全体を分類することで代替案の体系的な構造化を行う。スキーマ表現とその類似度の定義方法に基づいて解空間の構造が定まるため、スキーマモデルは慎重に選択しておかなければならない。しかし、設計問題の特性や性質を反映させることが重要であるため、スキーマの統一的な表現を想定することはできず、問題に合わせて柔軟に設定しなければならない。代替案の分類方法に制限はないが、本研究では、代替案の選定と解釈の意思決定過程に階層性が潜んでいることを想定し、それらをビルディングブロックの階層構造に対応させることを狙って、分類のアルゴリズムに階層的クラスタリング^[65]を採用する。

Step 3: 構造分析と結果の体形化

Step 2 の分類操作によって得られた木構造をもとに、ビルディングブロック集積の特徴を明らかにする。各分岐点で 2 つのクラスタを表現するスキーマを作成し、それらの固定文字列位置を比較することで、ビルディングブロックとその集積過程の変化を読み取る。分岐点でのビルディングブロックの排他的論理和は分岐位置での分岐理由に相当する。ビルディングブロックの違いやそれらに伴う特徴など、獲得された知見は意思決定分岐図としてデンドログラムに対応させて整理され、意思決定者はそれらを基にして分割面に解釈を与える。

以下の節では、それぞれのステップの詳細な手順と基盤となる技法について説明する。

3.4 多目的最適化による代替案の生成と解空間構造の定義

3.4.1 多目的最適化

Step 1 では、分析対象である解空間を用意し、その構造を定義する。

まず、有望かつ多様な代替案の集合を生成する。本研究では、有望かつ多様な代替案の集合を多目的最適化のパレート最適解集合と解釈し、多目的最適化問題を定式化し、最適化を実行する。目的関数を多数設定する場合、最適化問題は一般に式 (2.1) に示されるような形式で定式化される。目的関数が多くなればなるほどその最適化は困難になり、 $i \leq 3$ の場合は多目的最適化、 $i \geq 4$ の場合は多数目的最適化と呼称することで区別する場合もある。多目的最適化のパレート最適解は

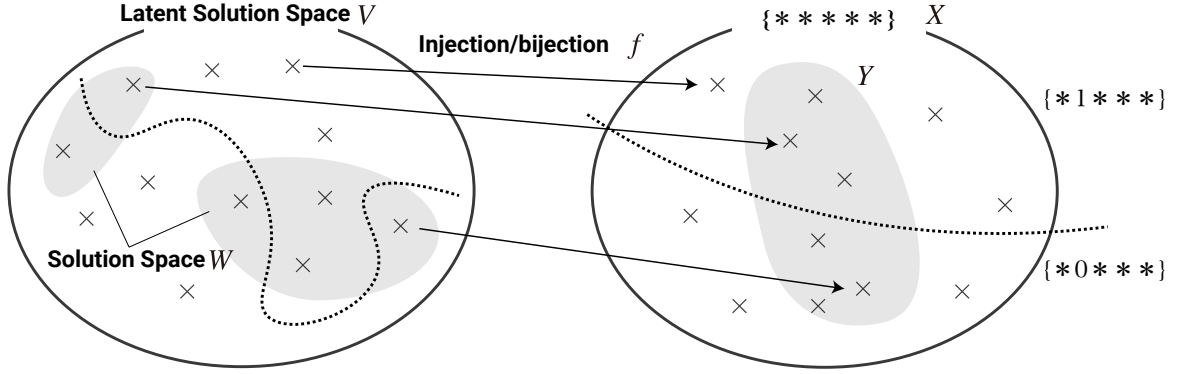


Fig. 3.5: Mapping relationship between the space generated by the schema and the design space.

2.2.3 項で紹介した ε 制約法や、多目的進化アルゴリズムを適用することで生成することができる。本研究では、パレート最適解の性能については焦点ではないので、理想的なパレートフロントが獲得されたものとして仮定し、解空間の構造分析を行うことにする。一般に、大規模な組合せ型の最適化問題は組合せ数が膨大であることから、最適化は困難であると考えられており、問題に合わせて適切なアルゴリズムを選択する必要があることに注意しなければならない。

3.4.2 個別問題に応じたスキーマモデルの選択

多目的最適化によって生成された代替案に対して、それらの集団の構造を表すスキーマモデルを定義する。スキーマモデルによって生成される空間と、解空間の関係は図 3.5 に示される。最適化問題が定式化された時の設計変数ベクトル $\mathbf{x}$ によって生成される潜在的な解空間全体を V 、スキーマモデルのビット列 δ で生成される空間を X とすると、代替案のスキーマモデルへの変換は写像 $f: V \rightarrow X$ として表現される。分析対象である解空間 W は定式化における有望な代替案の部分空間であり、 $W \subset V$ が成立している。本研究ではスキーマモデルを用いて解空間の構造分析を行うため、解空間 W を直接考えるのではなく、写像 f による像 $Y = f(W)$ を分析することになる。構造の分析から有望かつ多様な解の特徴や傾向を読みとり、意思決定の支援に役立てるためには、ここで定義する写像 f とスキーマモデル δ は次の 3 つの条件を満たさなければならない。

- (A) 写像 f は単射あるいは全単射である。
- (B) スキーマモデル δ の各ビットはそれぞれ設計上の判断に対応しており、それらは独立した関係にある。
- (C) $*$ を用いたスキーマモデルでの集合表現は集団の代表的な構成として特徴を表現することができる。

スキーマモデルを定義する際には、上述の性質に従い、代替案の特徴を表す変数が何かを、意思

決定者が判断することになる。基本的なスキーマ定義の方策は設計変数のビット列化である。目的関数に代表される設計の特徴量は設計変数の写像であることから、設計変数をそのまま用いることは自然ともいえる。スキーマモデルを定義する際に設計変数の情報を捨象しなければ (A) の単射性は保証される。写像が全単射である場合は、* を用いたスキーマでその集合の特徴を記述することができる。ただし、解空間は考えうる全ての代替案を含んだ集合ではないので、場合によっては* を用いたスキーマはそもそも解空間に含まれていない解を包含する可能性があることに注意しなければならない。設計変数の値に強い相関がある場合は、その背景にある設計上の判断に類似性があることが多い。次のステップでの類似性の定量化において、それらの感度が高くなるため、(B) に基づき、相関の強い設計変数を集約してスキーマを定義しなければならない。

スキーマモデルの最も簡単な定義方法はバイナリ変数で定式化された設計変数を、そのままビット列として転用する方法である。組合せ最適化問題の場合、設計変数はバイナリ変数を用いて定式化されることが多い。例えば、ナップサック問題であれば、ナップサックに入れるなら 1、そうでなければ 0 といった具合で定式化される。バイナリ変数であればそのままにビット列を形成することができ、スキーマとして集団を表現することができる。* はそれらの曖昧さを表現し、(C) の条件を満たす。設計変数そのものではビット列に変換できない、あるいは次のステップで類似度を定量化することが困難である場合、同値な中間モデルを導入して写像 f を定義する。例えば、解空間が順列構造やネットワーク構造のような複雑な構造を持つ場合、それと同値な 2 進数表現を用いた中間モデルを新たに定義してスキーマとして扱うことになる。中間モデルを検討する際には、遺伝的アルゴリズムでの符号化方法が参考になる場合がある。ただし、遺伝的アルゴリズムでの符号化ではビット列から設計変数への写像を考えるが、全射性は実行可能領域として保証するものの、単射である必要がない。ここでの写像 f は符号化の逆写像であり、そもそも写像 f を定義できない可能性がある。

スキーマモデルは設計問題としての特性や性質をうまく反映させることが重要である。スキーマの統一的な表現をあらかじめ想定することはできず、問題に合わせて柔軟に定義しなければならない。したがって、意思決定者が知りたい部分は何か、どこに特徴が現れる可能性があるのかなどを、あらかじめ整理した上で、その定義方法を議論する必要がある。

3.5 クラスタリング法による代替案の構造化とビルディングブロックの顕在化

3.5.1 代替案の類似性の定量化

Step 2 では、クラスタリング技術を応用することで解空間に属する代替案の集合を階層的に構造化する。クラスタリングを行う際には、一般に、観測値間の非類似度を定義しなければならない。本項では、解空間の構造を定義するスキーマのビット列を観測値として扱い、それぞれの代替案の類

Tab. 3.1: Counts of binary outcomes for two individuals^[67].

	Outcome	Individual δ		Total
		1	0	
Individual δ'	1	a	b	$a + b$
	0	c	d	$c + d$
	Total	$a + c$	$b + d$	$p = a + b + c + d$

似性を定量化する。つまり、クラスタ分析は元の解空間ではなく、写像 f の像であるスキーマモデルによって定義された空間 Y に対して行われる。クラスタ分析を成功させるためには、どのクラスタリング方法を選択するかよりも、オブジェクトやクラスタ間の非類似度の設定を適切に選択することが遥かに重要なことであると考えられている^[66] ため、本項では汎用的な方法論としての性質を担保するため、いくつかの類似性尺度の候補を列挙し、その選定の指針について論ずる。

スキーマで表現された観測値は全てバイナリデータである。バイナリデータについては多くの類似性尺度が提案されている^[65]。2つの文字列 (δ, δ') の2つのビット列の中の同位置のビット (δ_i, δ'_i) に注目すると表 3.1 に示されるように、 $(\delta_i, \delta'_i) = (1, 1), (0, 1), (1, 0), (0, 0)$ の4つの組合せパターンを想定することができる。ここで、全ビット列 $(i = 1, 2, \dots, I)$ でのそれぞれのパターンに当てはまるビットの総数を a, b, c, d とする。つまり、 $(a + b + c + d)$ はビットの長さ p である。Gower と Legendre^[67] は a, b, c, d の値を組合せて、バイナリデータ δ_s, δ_t 間の類似性尺度 s_{st} のリストを提供している。ここではその中から代表的な定義をいくつか取り上げ、表 3.2 にまとめる。なおこれらを非類似度、あるいは距離として扱う場合は、 $(1 - s_{st})$ を考えれば良い。

表 3.2 のように多様に類似性尺度が定義できることの背景には、類似点の取り扱いと、相違点の取り扱いが多様であるからである。どの類似性尺度を用いるかについてはバイナリ情報が持つ特徴やそもそもの意味に立ち返って議論する必要がある。

まず、類似性の取り扱いとして、1 で一致する数を表す a と 0 で一致する数を表す d を区別するかによって定義が大きく異なる。区別の有無については、バイナリ情報が持つ定性的な意味を踏まえて判断する必要がある。例えば、性別をバイナリ情報で識別する場合は、1, 0 は等価であるため、類似性尺度に含まれるべきである。つまり、 S_1 や S_2, S_5 が採用されることになる。一方で、先述のサプライチェーンの例のように設立の有無を考える場合などは、それらは区別する必要があり、 S_2, S_4, S_6 のような尺度を用いることになる。ただし、Sneath と Sokal^[68] は、0 一致の程度を表す d を含むかどうかに関しては、厳密な規則がないことを指摘しており、柔軟な判断が求められる。

次に、相違点の取り扱いについてであるが、バイナリデータの相違点とは基本的にはビット列での排他的論理和であるため、具体的には、 b と c の値を参照することになる。どの類似性尺度においても b, c の値が大きいほど類似度が小さくなるように設定されているが、相違点に対称性がある

Tab. 3.2: Formulation of similarity coefficients amongst binary variables^[67].

Measure	Formula
S_1	$s_{st} = \frac{a+d}{a+b+c+d}$
S_2	$s_{st} = \frac{a}{a+b+c}$
S_3	$s_{st} = \frac{a+d}{a+2(b+c)+d}$
S_4	$s_{st} = \frac{a}{a+2(b+c)}$
S_5	$s_{st} = \frac{a+d}{a+\frac{1}{2}(b+c)+d}$
S_6	$s_{st} = \frac{a}{a+\frac{1}{2}(b+c)}$
S_7	$s_{st} = \frac{1}{2} \left(\frac{a}{a+b} + \frac{a}{a+c} \right)$
S_8	$s_{st} = \frac{a}{\sqrt{(a+b)(a+c)}}$

か、すなわち、 $b = c$ を満たすか否かでその定義方法が異なる。 S_1 から S_6 までの類似性尺度は b, c を区別しない尺度であり、相違点の対称性を前提にしている。和 $(b + c)$ を用いて類似度を定義しており、それぞれでの相違点の感度の取り扱いで、感度が高い順に (S_3, S_4) 、 (S_1, S_2) 、 (S_5, S_6) と 3 つに大別することができる。一方、 S_7 と S_8 はそれらを区別することができる尺度である。 S_7 と S_8 は共に a と d を区別する尺度であるが、 b に焦点を当てた類似度と c に焦点を当てた類似度との、相加平均と相乗平均を算出している。なお、2 つのビット列の相違点に対称性があり、 $b = c$ を満たす場合は、 S_7 と S_8 は S_6 と同じ類似度を算出することになる。

実際には異なる類似性尺度を使用すると、 S_2 で類似していた 2 つが、 S_7 では類似していないなど、類似度が大きく異なる値を取ってしまうことは稀でない。スキーマの類似性は解空間での関係そのものであるので、Step1 で定義したスキーマモデルの特徴やそこに潜む背景に基づいて、認識や言語表現のレベルでの類似性とその距離の値を見比べながら個別に検討しなければならない。具体的な類似性尺度設定による構造分析結果への影響は第 4 章で事例を通して検証する。

3.5.2 項で後述するが、クラスタ間の距離を定量化する際に、クラスタ重心を定義する場合がある。バイナリ情報の重心を平均値として定義する場合は、表 3.2 で示すようなバイナリデータに基づく類似性尺度を用いることができないため、連続変数の距離定義を用いる。ここでは、連続変数の距離関数を取り上げ、バイナリ情報 $\delta_{i,j}$ を適用した場合での類似性尺度 $s_{s,t}$ を確認する。代表的な非類似度の定義としてユークリッド距離、市街地距離、コサイン距離を取り上げ、連続変数ベクトル $\mathbf{x}$ の非類似度を $d_{st}(\mathbf{x}_s, \mathbf{x}_t)$ とする。

- ユークリッド距離 (Euclidean distance)

ユークリッド距離は以下の式で定義される。

$$d_{s,t}^2 = (x_s - x_t)(x_s - x_t)' \quad (3.1)$$

ユークリッド距離は2点間の直線距離を表す。スキーマでの変数ベクトルは0か1かのバイナリ情報である。したがって、ベクトルの差分は実質的に(1,0)と(0,1)の組合せを数えていることになる。したがって、ビット列の類似性尺度は次式で表される

$$s_{st} = 1 - \sqrt{b+c} \quad (3.2)$$

- 市街地距離 (City block distance)

市街地距離は以下の式で定義される。

$$d_{s,t} = \sum_{j=1}^n |x_{sj} - x_{tj}| \quad (3.3)$$

市街地距離は各座標の差の総和を距離として定義している。ビット列では、ユークリッド距離と同様に相違点の数え出しに相当し、類似性尺度は次式で表される。

$$s_{st} = 1 - (b+c) \quad (3.4)$$

- コサイン距離 (Cosine distance)

コサイン距離は以下の式で定義される。

$$d_{st} = 1 - \frac{x_s x_t'}{\sqrt{(x_s x_s')(x_t x_t')}} \quad (3.5)$$

コサイン距離は、そのまま、ベクトル同士の成す角度の大きさを表現する。第2項は $\cos 0 = 1$ 、 $\cos \frac{\pi}{2} = 0$ であるので、0に近づくほど離れ、1に近づくほど近いことを意味する。1からの差分をとることにより、「類似度」ではなく「距離」として定義している。ベクトルがバイナリ情報のみで構成されている場合では、分子が(1,1)の総数を、分母が各ベクトルの1の数の相乗平均を表す次式で表され、類似性尺度は表3.2の S_8 と同値である。

$$s_{st} = \frac{a}{\sqrt{(a+b)(a+c)}} \quad (3.6)$$

3.5.2 階層的クラスタリング法の活用

本研究では、代替案の選定と解釈の過程に階層性を想定し、それらをビルディングブロックの集積過程に対応させることを狙って、分類のアルゴリズムに凝縮型の階層的クラスタリングを採用する。

階層的クラスタリングは段階的にクラスタを生成する手法である。その方法として、凝集型と分割型の2種類に大別することができる^[66]。本研究で用いる凝集型手法では最下層から始め、各階

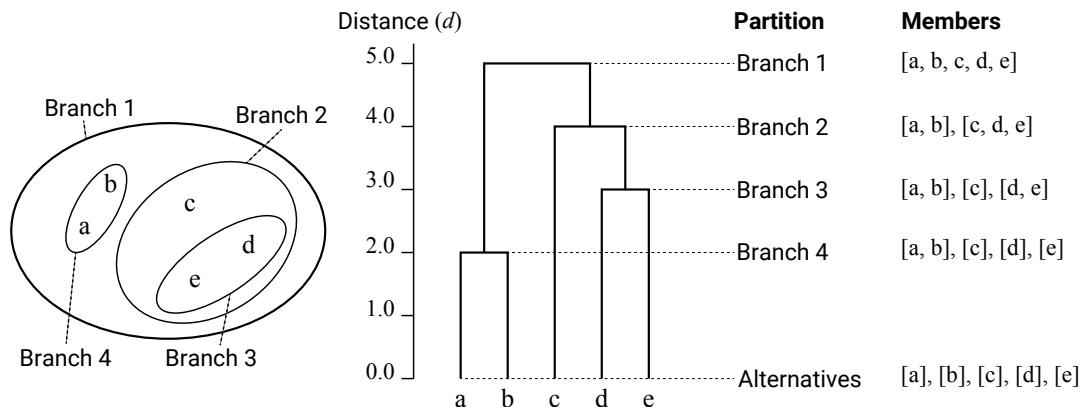


Fig. 3.6: Hierarchical clustering and dendrogram.

層で選択した2つのクラスタを1つのクラスタに再帰的に併合する。常に1つ上の階層にはクラスタ数が1つ少ないグループ集合が得られる。 N 個のオブジェクトから $N-1$ の階層が得られ、クラスタ分析の結果は樹形図またはデンドログラムと呼ばれるに二分木によって表現することができる。図3.6は、階層的クラスタリングの結果の例^[65]である。階層的クラスタリングにおける分割のプロセスには単調性があるため、併合したクラスタ間の非類似度は階層が上がるにつれて単調増加する。各頂点の高さを2つの子頂点間の非類似度に比例するようにして二分木を可視化したものが、図3.6の中央に位置するデンドログラムである。デンドログラムではクラスタ間の非類似度を結合点の高さで表現する。

凝縮型手法ではクラスタの併合を行うので、前項で論じた観測間の距離だけでなく、2つのクラスタ(観測グループ)の距離を定義しなければならない。クラスタ間の距離の定義はリンケージ手法と呼ばれ、さまざまな凝縮型クラスタリング手法が提案されている。本項では、代用的なリンケージ手法^[65]を取り上げる。以下では、その詳細について述べ、選定の指針を論じる。

- 単連結法 (Single linkage)

最近隣法とも呼ばれる。2つのクラスタからそれぞれ個体を選び、総当たりで個体間の距離を計算する。最も近い個体間の距離を、2つのクラスタの距離として定義する。

- 完全連結法 (Complete linkage)

最遠隣法とも呼ばれる。単連結法の対極にある方法で、2つのクラスタからそれぞれ個体を選び、総当たりで個体間の距離を計算する。最も遠い個体間の距離を、2つのクラスタの距離として定義する。

- 郡平均法 (Group average linkage)

単連結法と完全連結法を織り交ぜたような方法で、総当たりで求めた距離の平均値を2つのクラスタの距離として定義する。

- ウォード法 (Ward's method)^[69]

2つのクラスタの併合に、グループ内の分散に対するグループ間の分散が最大になるように

Tab. 3.3: Hierarchical agglomerative clustering methods: Lance-Williams parameters.

Method	α_i	α_j	β	γ
Single linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group average linkage	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Ward's method	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{n_j+n_k}{n_i+n_j+n_k}$	$\frac{n_k}{n_i+n_j+n_k}$	0

クラスタを形成する方法。

これらの標準的な階層的手法の数学的特性は、Lance と Williams^[70] の定式化のパラメータで定義することができこれらのクラスタ間の距離は以下の式でまとめられる。

$$d_{k(ij)} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{kj} - d_{ki}| \quad (3.7)$$

ここでの d_{ij} はクラスタ i とクラスタ j の距離であり、 α, β, γ はパラメータである。上述の方法とのパラメータとの関係は表 3.3 に示される。

ウォード法はこれらのリンケージ手法の中でも比較的分類感度が高く、広く用いられている手法である。ウォード法を用いる場合は、クラスタの重心を定義する必要がある。式 (3.7) の Lance と Williams^[70] の定式化では、平均ベクトルをクラスタ中心として扱っているものの、スキーマでは集団の中間値は * を用いて表現するため、重心を別途定義する必要がある。また、重心を定義したとしても、表 3.2 に示されるような距離関数はいずれもバイナリ情報に限られるため、別途、3.5.1 項で導入した連続変数に用いる距離関数を用いることになる。

距離の設定と同様に、リンケージ手法の選択によって結果が大きく変わることは少なくない。データ構造が複雑な場合は個体を入れ替えるだけでもデンドログラム形状が大きく変化する。デンドログラムの結果はあくまでもデータの視覚的な要約であり、結局のところ、さまざまな手法によって生成されるデンドログラムを横に見た上で意思決定者が望ましいものを選ばなければならない。デンドログラムを評価する基準の 1 つに、コーフェン相関係数が挙げられる。階層的クラスタリングでは、観測値から距離行列を生成した後に、リンケージ手法に応じてコーフェン行列を算出し、デンドログラムはコーフェン行列に基づいて描かれる。コーフェン相関係数とは距離行列とコーフェン行列の相関を求めたものであり、1 に近いほどデンドログラムの形状が実際の距離の関係と相関が取れていることを表している。ウォード法は分類感度が高い一方でコーフェン相関係数の値は小さくなる傾向にあるが、簡素な木構造を生成する特徴がある。また、その他のリンケージ手法によっては、コーフェン相関係数は高いものの、鎖効果や拡散現象、反転現象などの恣意的な構造を導く可能性もあり、実際のところは、複数のデンドログラムを横に並べて、解釈がしやすい

形状を選択することが多い^[71]。したがって、どの手法を用いるか、どの結果を信じるかについては、階層的クラスタ分析の結果だけでなく、他の解析方法などを取り入れながら探索的にデータを眺めた上で、総合的に判断することが重要である。本研究では、デンドログラムの形状は意思決定者の階層的な思考過程と 1 対 1 に対応するため、結果の正しさはもちろんのこと、解釈のしやすさという観点は手法の選択の重要な基準である。具体的なリンケージ手法の選択による構造分析結果への影響についてもまた、第 4 章で事例を通して検証する。

3.6 構造分析と結果の体系化

3.6.1 スキーマモデルに基づく構造分析

Step 3 ではクラスタ分析によって構造化された代替案がどのような構造を持っているのかをスキーマを用いて分析し、図 3.4 右のようにデンドログラムに対応させて、構造特徴と追加特徴を整理し、意思決定分岐図として木構造に体系化する。

階層的クラスタリングの結果であるデンドログラムは、上から順に各階層で既存のクラスタのうちの 1 つを 2 つの新しいクラスタに再帰的に分割していくため、図 3.5 に示されるスキーマモデルによって生成される空間 Y での階層的クラスタリングの結果は、写像 f の単射性から、解空間 W でも各分岐点に対応する 2 つの部分集合を定義でき、その分割面を想定することができる。解空間での構造はスキーマモデルを用いて表現されているので、その分割面の意味は 2 つの集合のスキーマの特徴の違いに対応するため、結果的にスキーマのビット列での比較に帰着する。また、分割が細くなるにつれてスキーマの固定部分が増えていくため、階層的クラスタリングの分割過程からスキーマの集積を理解することができる。

3.2.1 項のスキーマモデルの定義に立ち戻ると、スキーマは特定の代替案の集合を表していて、ビットを固定することは集団全てに共通する特徴があることを意味する。言い換えると、あるビットを 1 に固定しているスキーマでは、属するすべての代替案でそのビットに対応する意思決定について、全く同じ判断を下していることになる。一方で、固定せず * として表されているビット位置については集団でその判断が多様であることを意味している。本研究では 2 つの集団を表すスキーマの文字列の組合せをもとに、分割面に解釈を与える指針をテンプレートとして設定しておく。2 つのスキーマベクトル δ, δ' の i 番目のビットの組で次の 4 つのカテゴリに分類される。

Category A: $(\delta_i, \delta'_i) = (1, 1), (0, 0)$

2 つのスキーマの固定方法が同じ場合、その分岐位置での特徴を表すビルディングブロックとして考えることができる。 $(1, 1)$ と $(0, 0)$ とを区別するかどうかは、バイナリデータの距離の定義と同様に、解釈の側面からも個々の問題に合わせて判断が必要である。

Category B: $(\delta_i, \delta'_i) = (1, 0), (0, 1)$

左右の分岐での明確な差異であり、ビルディングブロック集積の分岐を表すレバレッジポイントである。一方は1、他方は0ということは、一方の集団すべてで行った判断と真逆の判断を別の集団で行っていることになり、解釈上では分割面の意味として捉えることができる。

Category C: $(\delta_i, \delta'_i) = (1, *), (*, 1), (0, *), (*, 0)$

一方は固定、他方は固定していないということは、分岐としては構造が曖昧であることを意味する。しかし、解釈の上では Category B の要素と組合せることで判断の方針の違いとして解釈することができる。例えば、デンドログラムで左に分岐した時のスキーマを $\{11*\}$ 、右を $\{0**\}$ であるとする、1 番目のビットは Category B、2 番目のビットが Category C として分類される。左クラスタでは Category C のビット列を 1 に固定するが、右のクラスタでは当該のビット列に対して 0 をとる代替案と 1 をとる代替案が混在していることを意味し、判断を先送りに行っていると解釈できる。この場合、Category C の 2 番目のビットは、1 番目のビットを 1 の固定した際の付随する一連の判断として結論づけることができる。左クラスタでは 1 番目のビットを 1 にすれば 2 番目も 1 に固定するが、右クラスタでは、1 番目のビットを 0 と固定した場合は、即座に 2 番目のビットを決めなくてもよい多様性を有している。したがって、Category C を分割面の意味として考察するならば、Category B に付随した意思決定方針の分岐として捉えることができる。

Category D: $(\delta_i, \delta'_i) = (*, *)$

どちらの集団でも判断が分かれていることを意味し、分岐理由には作用していないことがわかる。逆に言えば、分岐にはこのビットに関する判断が関連しないことが明らかであるため、分岐理由の根拠として除外することができる。

スキーマモデルの条件 (B) より、スキーマモデル δ の各ビットはそれぞれ設計上の判断に対応しているため、ビットが 1 か 0 あるいは * によってその意味が異なる。これらのテンプレートを参考にしながら、2 つのクラスタを比較することで、構造が分岐していることの意味を、着眼点を明確にしながらか解読できるようになる。4 つのカテゴリのそれぞれの重要性については、やはりスキーマモデルの定義に戻って個別に検討が必要であるが、Category A 及び Category B はビルディングブロックの集積過程の理解という観点で重要である。また、Category D 以外の 3 つでは 0 と 1 とで固定方法の違いがあり、距離の定義にも関連しながら、区別するか否かを考えておく必要がある。

3.6.2 意思決定分岐図による分析結果の体系化

デンドログラムによるビルディングブロックの構造化や、それらの 3.6.1 項で導入したテンプレートに基づいた解釈は、意思決定分岐図として整理される。意思決定分岐図はデンドログラムの形状をそのままに、それぞれの分岐点での左右の特徴の違いを記載した知識管理テンプレートである。図 3.3 で示した意思決定の構想に従って、それぞれの分岐位置では、階層的クラスタリングの

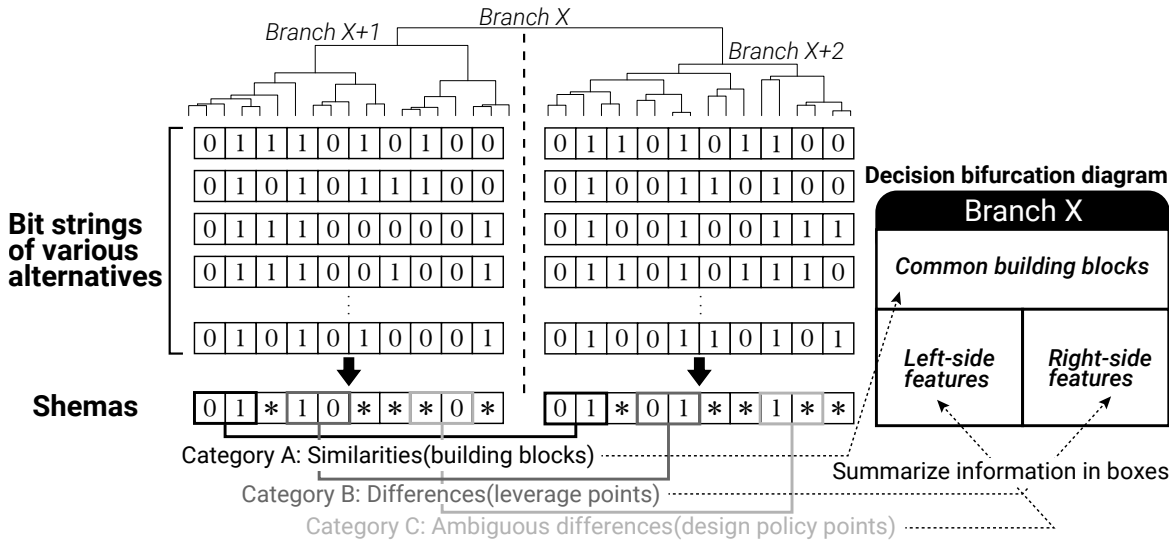


Fig. 3.7: Box of the decision bifurcation diagram and structural features.

分類結果からわかる構造特徴と、更なる分析や評価、意思決定者の解釈によって獲得した追加特徴の、2種の情報を記載する。前者は提案手法の一連の流れで定められる一方、後者は主観に委ねられる。

構造特徴の整理方法の一例を図 3.7 に示す。デンドログラムでのそれぞれの分岐点は代替案を左右2つのクラスタに分割する。それぞれのクラスタに属する代替案を表すスキーマを比較することで、構造の違いを整理する。図 3.7 では Branch X に焦点を当て、左右のビット列の例とそれによって形成されるスキーマとその比較結果を表している。デンドログラムは階層構造をとっているため、Branch X での左クラスタは次の Branch X+1, 右クラスタは Branch X+2 といった具合で、個別の代替案に至るまで分割が再帰的に行われることになる。4つのカテゴリのうちどれを重要視するかは、上述の通り、問題に依るところが大きい。Category A 及び Category B はビルディングブロックの集積過程を再現する上では重要な要素であり、意思決定分岐図に記載しておくことが望ましい。図 3.7 での第 1,2 番目のビットは共通ビルディングブロック、つまり Category A であり、クラスタの共通点特徴であるので、Branch X の代表的な特徴としてまとめられる。第 4,5 番目のビットはビルディングブロックの相違点であり Category B である。これらは左右クラスタの相違点を表すため、左側の特徴と右側の特徴として分けて記載する。問題によってはまた、第 8,9 番目のビットである Category C も合わせて記入する。階層性に着目すると、Branch B の要素は左右クラスタのビルディングブロックであるので、次の分岐点での共通ビルディングブロック (Category A) として取り扱われることになる。一般に、分岐が進むほどクラスタに属する代替案の数が少なくなるため、共通点が多くなり、スキーマの固定部分もまた多くなる。言い換えると、Category A や Category B の要素が多くなる。これは設計の詳細化に対応している。このような整理方法を保ったまま、全ての分岐点で情報を整理することで構造特徴が階層的に整理される。

意思決定分岐図では、図 3.3 で示した意思決定の構想のように、左右の構造特徴としてまとめた違いの意味を説明する追加特徴を追記することで、意思決定者の理解を支援する。集団の特徴を端的に説明する方法として、(1) 評価指標を用意して代表値として扱うことと、(2) 構造の違いとその意味を言語化することの 2 つの方法が考えられる。どのような情報を用いて 2 つのクラスタの特徴を説明するのかについては、これもまた汎用的な規範がないことから問題に合わせて柔軟に考える必要があるものの、最適設計問題のモデルの範囲や粒度からその方針を次のように定性的に定めておくことができる。まず、詳細設計に位置付けられるような、評価方法が明確な設計問題、例えば生産スケジュールの計画問題であれば、評価指標の値そのものが重要である。最適化に用いた目的関数やその他の評価指標を代表値として用いるに適している。一方、概念設計や基本設計などの設計段階の上流段階では、詳細が決まっていないことから、Design Structure Matrix(DSM) に代表されるような複雑な事象を 1 つに集約するモデルを用いて評価することになる。集約モデル上での評価値そのものには設計としての重要な知見はなく、むしろ最適化結果そのもの、つまり設計変数やパラメータに重要な知見が潜んでいることになる。この場合であれば構造に注目して特徴を整理することが望ましい。

目的関数などの評価指標を用いてクラスタの特徴を表現し、分岐理由を検討する場合、多目的設計探索^[5]や Trade Space Exploration^[6]と同様に、評価関数空間でのクラスタの分布を理解することが効果的である。最も基本的な理解の方法は、評価関数空間の可視化である。別の評価指標を用意しないのであれば、評価関数空間は目的関数空間である。評価関数が少数であれば、単純な目的関数空間の可視化で対応できるものの、評価関数が多数になる場合は工夫が必要である。Design Space Exploration^[57]や、Trade Space Exploration^[6]と同様に自己組織化マップなどの次元削減技術や回帰分析技術が有効である。

本論文では、第 4 章では具体的で評価指標からの解釈が容易例題としてスケジューリング問題を取り上げ、基本的な考え方の確認と視覚化によるシナリオ分類に取り組む。次に、第 5 章では評価指標を多用に想定できるグローバルプロダクトファミリー設計問題を取り上げ、回帰分析を用いた解釈法を提案する。最後に、第 6 章では集約モデルを用いた製品モジュール化設計を取り上げ、構造そのものに注目した解釈を行う。

3.6.3 構造分析の意思決定への効果

意思決定分岐図は構造分析の結果とそれに伴う部分的な解釈を 1 つにまとめた、分析過程の要約である。意思決定分岐図を作成することによる効果は次の 3 点にまとめられる。

- 代替案のすべてを体系的に整理することで、それぞれの結果に至る経緯をビルディングブロックの集積として解釈できるようになる。ビルディングブロックは解空間の構造そのものであるため、それぞれのブロックを表すビットは、スキーマの定義に立ち戻って考えること

で、その意味を明確にできるため、代替案それぞれの特徴や意味が明確になる。

- 意思決定分岐図をトップダウンに読み解くことで、選好解を定める上での認知的負荷を低減させることができる。トップダウンに辿ることは、クラスタリングの凝縮過程の逆に対応し、解集合を順に絞り込むことができる。それぞれの構造特徴と追加特徴に基づいて、右か左かの二者択一の選択を繰り返すことにより、特定の解に至ることができる。つまり、意思決定者の着眼点を明確にしながら、大局をつかむ意思決定から詳細を決める意思決定へと、体系的な手続きに従って選好解を定めることができる。
- 意思決定分岐図をボトムアップに読み解くことで、代替案の関係性を俯瞰的に理解することができる。任意の2つの解を選択すれば、それらの類似性を分岐位置から理解することができる。また、分岐位置での相違点に関する情報はそれらの違いを明快に説明する。

それぞれの手法適用による効果の妥当性については、第4章から第6章までの事例を通して、第7章で論じる。

3.7 結言

本章では、本研究の中核になる組合せを伴う解空間の構造分析法を提案し、構造分析を用いた意思決定の支援の形について論じた。まず、本研究で取り扱う組合せを伴う解空間を数理的に扱うため、スキーマに基づく定義方法を導入し、提案する構造分析とその解釈方法の構想を述べた。そして、構想に基づいて提案手法を3つのステップに分解し、それぞれの詳細な手順を述べた。ステップ1では、研究で考える解空間を有望かつ多様な代替案の集合と定義した上で、代替案を獲得する手段として多目的最適化を用いることを述べた。問題の特徴を踏まえたスキーマ定義のため、解空間に属する集合をスキーマで表現される集合へ写像する際に満たすべき3つの条件を整理し、分析対象の解空間を定義する方針について述べた。次に、ステップ2では、意思決定のプロセスに階層性を想定し、階層的クラスタリングを用いて解空間を階層的に分類することを述べた。スキーマモデルをもとにした代替案やクラスタ間の距離定義の方策について、具体的な事例を挙げながら、それぞれの特徴と選定の指針を整理した。最後に、ステップ3では構造化された代替案の集合から構造分析を行う具体的な方法について述べた。階層的クラスタリングの分割過程とビルディングブロック表現の絞り込み過程を対応させて解空間構造を明らかにし、スキーマの文字列の組合せに基づく解釈の方針を述べた。分析を通して得られた知見を要約するテンプレートとして意思決定分岐図を用意し、最後にその効果を述べた。

第 4 章

生産スケジューリング計画への適用

4.1 緒言

本章では、第 3 章で構築した組合せを伴う解空間の構造分析手法の有効性を検証するためのベンチマーク問題として、生産スケジューリング計画、特にフローショップスケジューリング問題を例に引く。フローショップスケジューリング問題はジョブの順序を決定する組合せ問題であり、その解空間は順列構造を有している。構築した解空間の構造分析法では、スキーマを用いた解空間の定義方法や、その中での代替案の距離やクラスタの距離の定義方法に自由度がある。フローショップスケジューリング問題に合わせた解空間構造や構造の分析法の考え方を論ずる。最後に事例としてスケジューリング問題の中でも最も単純な問題である、順列フローショップスケジューリング問題を例に引いて提案手法を適用する。解空間の構造分析手法の諸パラメータの影響を調査しつつ、手法の前提になっているビルディングブロック仮説と意思決定分岐図を作成する効果を具体的事例を通して検証する。

4.2 フローショップスケジューリング問題と最適設計

生産スケジューリング計画とは、ある製品の生産工程の順序とそれに必要な人手や機械、材料などのリソースを特定し、計画を立てることである。特に、フローショップスケジューリング問題は、全てのジョブが同一の機械順序で処理されるシステムに対するスケジューリングを指し、製造業における基本的な問題の 1 つである^[1]。フローショップスケジューリングはその理論的な背景はさながらに、産業的な効果からも注目を浴びる応用分野であり、写真フィルム製造^[72]や農業食品産業^[73]、自動車製造^[74]など、数理最適化を用いたさまざまな実践例が報告されている。

生産計画の現場では、さまざまな評価指標を鑑みて計画を行う必要がある。例えば、総流量時間 (total flowtime) の最小化、総完了時間 (total completion time) の最小化、総作業時間 (makespan) の最小化、納期遅れ (tardiness) の最小化、在庫コスト (inventory cost) の最小化など、多岐に渡

る^[9]。これらの評価指標はしばしば互いに相反するトレードオフの関係を持つため、生産のスケジュールを計画するマネージャーは複数の相反する判断基準をどのように扱うのかを判断しなければならない^[75]。数理最適化による計画支援を行う場合、これらの複数の評価指標は目的関数として扱われ、多目的最適化問題が定式化される。特に、目的関数を複数設定するスケジューリング問題は多目的スケジューリング問題（Multi-objective Scheduling Problem, MOSP）と呼ばれている。1995 年以前の MOSP に関する研究は、分枝限定法やプログラミング法などの解析的關係に基づくアプローチが主流であった^[76]。しかし、これらの焦点は単一機械や並列機械であり、生産現場から比較しても小規模な問題に限られていた。1995 年以降から遺伝的アルゴリズムを適用した手法が報告され始め、現在は、メタヒューリスティックが MOSP を解くための主要な方法として位置付けられている^[75]。

MOSP を解くために提案されてきたさまざまな手法は、より短時間で、あるいはより優れたパレート最適解集合を導出することを目的としてきた。しかしその一方で、選択肢の中から最も望ましい解を選択することについてはこれまで焦点が当てられてこなかった。解の違い、例えば、それぞれの解のメリットとデメリットをそのジョブの順列の意味にまで踏み込んで直感的に理解することは、これまでの研究が示してきたような組合せの複雑さ故に、困難であると考えられる。スケジューリング問題の場合はモデルの忠実度が高いため評価関数値を基準にして選好解を定めることが基本となるだろう。しかし、順列が異なっても目的関数の値が近いような等価解が存在していれば、評価関数からだけではそれらの違いを把握することはできない。つまり、選定を行うためには、代替案がなぜ優れているのかを多角的に理解する必要がある、各選択肢に潜んでいるスケジューリングにおける戦略を結果から解釈する必要がある。そこで本章では、代替案の選定を支援する組合せを伴う解空間の構造分析法の枠組みを適用し、その効果を確認する。

4.3 順列構造問題のスキーマモデルの構築

フローショップスケジューリング問題とは、全てのジョブが同一の機械順序で処理されるシステムに対するスケジューリングであるので、その設計変数はジョブの順序である。本研究ではこのような順序を定める最適化問題の解空間構造を順列構造として定義しておく。順列構造である場合、Step 1 で 3.4.2 項で定義したビット列に基づくスキーマモデルを用いて構造を表現しようにも、そのままではビット列の形式ではないので、何らかの変換操作を行い、中間モデルを用意することが必要となる。本節では、スケジューリング問題の特性を活かしつつ順列構造をスキーマモデルへと変換するモデルを新たに導入する。

解空間の構造をスキーマモデルとして定義するにあたって、ビルディングブロック仮説に基づき、こういった要因で優れた解が形成されるのかを整理しておかなければならない。つまり、ビルディングブロックの候補として用意しておくべきものは何かをあらかじめ検討する必要がある。ビ

ルディングブロックは評価指標の向上に寄与する部分列であるため、評価指標の側から検討を行う。スケジューリング問題の評価指標は次のような 3 つの種類に大別することができる。

- (a) 完了時間に基づく評価指標 (例えば、総完了時間や、総流量時間など)
- (b) 納期に基づく評価関数 (納期遅れ時間など)
- (c) 在庫コストや利用コストに基づく評価関数

ジョブの順序からこれらの指標が算出されるが、より具体的に踏み込めば、例えば、完了時間に基づく評価指標はそれぞれのジョブの順序関係が積み重なることで形成される。つまり、完了時間に関するビルディングブロックを知るためには、順序関係を抽象するようなビルディングブロックの候補を備えたスキーマモデルが望ましいと考えられる。このように、問題の特性や意思決定者が特に重要視する評価指標に基づいてスキーマモデルを選択しなければならない。実際には、3.4.2 項で述べたように、スキーマに基づく解空間構造の表現方法は遺伝的アルゴリズムから着想を得ているため、スキーマモデルの定義を考えるにあたっては、遺伝的アルゴリズムでの符号化方法が参考になる。

本章では、順列構造を表現するための 2 つのスキーマモデルを導入する。3.4.2 項で整理した 3 つのスキーマモデルが満たすべき 3 つの条件に基づいて、それらの特徴を述べる。

4.3.1 スキーマモデル A: Design Structure Matrix による方法

スケジューリング問題の評価関数のうち、(a) 完了時間に基づく評価指標は、先述の通り、ジョブの順序関係が積み重なることで評価値が算出される。完了時間を意思決定の基準として用いることを念頭に置くのであれば、順序関係を表現する要素をビルディングブロックの候補として定めることが妥当である。本項では、2 つのジョブの順序関係に焦点を当てた、Design Structure Matrix(DSM) を応用したスキーマモデルを導入する [77]。

DSM は、システムを構成する要素とそれらの相互作用を表すために用いられるネットワークモデリングツールであり、有向グラフをバイナリ情報のマトリクスへと変換することができる。本手法では、2 つのジョブの関係のみをビルディングブロック要素として抽出し、その順序を表す有効グラフを作成することで、順列構造を 2 進数表現に変換する。スキーマモデルの例を図 4.1 に示す。図 4.1 ではジョブ順列 {1,2,4,3,6,5} を例に引いてスキーマモデルを生成している。まず、ジョブ順列を {1,2}、{2,4}、{4,3}、{3,6}、{6,5} の 5 つのブロックに分割する。さらに、ビット列の距離計測の際に、最初と最後のジョブの入れ替え操作で類似性が小さくなることを防ぐため、最初と最後のジョブを識別するブロックである、{S,1}、{5,E} を追加し、合計 7 個のブロックを用意する。ブロックはジョブの順序関係を表すダイグラフの矢印に相当し、ダイグラフに対応する DSM が作成される。DSM の要素 δ_{ij} では、ジョブ j の後にジョブ i を実行する場合、 $\delta_{ij} = 1$ である。空白は $\delta_{ij} = 0$ であり、スキーマを 0 で固定することに対応している。行列をベクトルに変換したとし

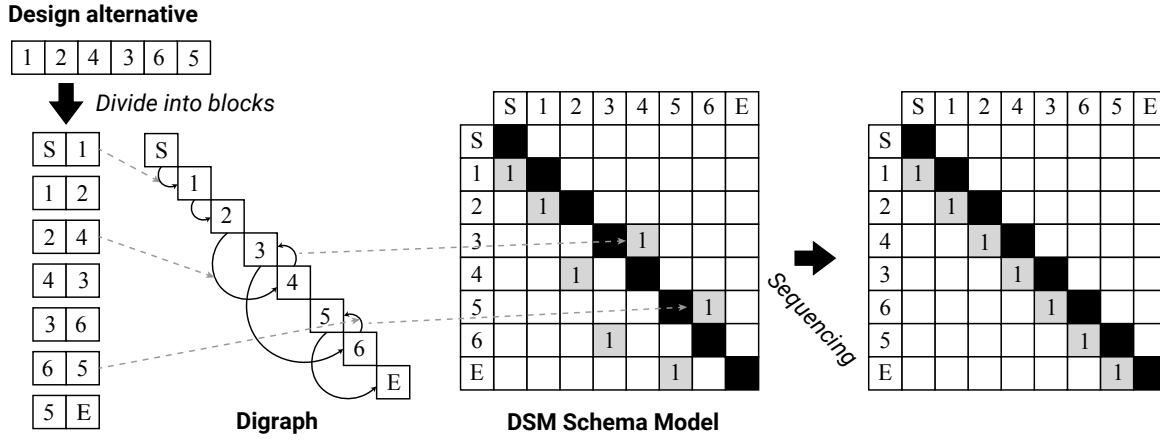


Fig. 4.1: Schema model construction using Design Structure Matrix(DSM).

てもその一般性は失わないため、本研究では δ_{ij} をそのままスキーマとして取り扱う。もう一方のスキーマモデルと区別のため、本研究ではスキーマモデル A と呼称する。

スキーマモデル A は次のような特徴を持つ。

- スキーマモデル A は DSM であるので、DSM の分析手法を応用することができる。例えば、パーティショニングと呼ばれる行列の並び替え操作によって元の順列を再現することができる。図 4.1 右側に示しているように、すべての要素をした三角成分に収まるように並び替えを行うシーケンシングと呼ばれる操作によって、元の順列が再構築される。一般に DSM の並び替え操作は手動で行うものの、近年はメタヒューリスティックを用いた計算機による処理が用いられている^[14]。
- ジョブ数 n のあらゆるジョブ順列に対して $(n+2) \times (n+2)$ の DSM として固有の形式で記述することができるため、その写像 f は単射である。また、DSM の要素 δ_{ij} にダイグラフとして矛盾が生じない次の条件を与えれば、全単射の関係にあり、(A) の条件を満たす。

$$\begin{aligned}
 \sum_{j=1}^{n+2} \delta_{1j} &= 0 \\
 \sum_{i=1}^{n+2} \delta_{i(n+2)} &= 0 \\
 \sum_{j=1}^{n+2} \delta_{ij} &= 1 \quad (i = 2, 3, \dots, n+2) \\
 \sum_{i=1}^{n+2} \delta_{ij} &= 1 \quad (j = 1, 2, \dots, n+1)
 \end{aligned} \tag{4.1}$$

- 各ビットはビルディング候補として注目したジョブの順序関係に対応しており、ジョブ i とジョブ j の順序関係の判断に対応していて、それ自体は独立しているため、(B) を満たして

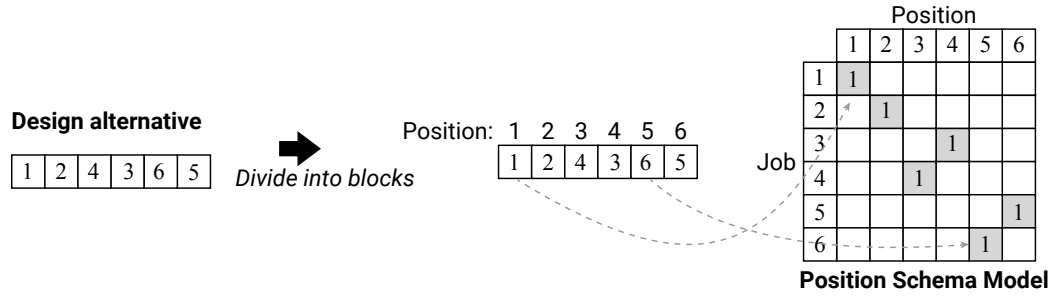


Fig. 4.2: Schema model construction based on the position of the job.

いる。

- 集団のすべての要素で平均を計算することで、重み付き DSM を作成することができる。重み付き DSM は * を用いたスキーマとは別の集団の表現方法であり、重心位置を定義することができ、Ward 法などのリンケージ手法を用いることができる。平均化したマトリクスは式 (4.1) を満たさないため、スキーマの意味としては存在しないジョブ順列を含んでいるものの、重み付き DSM としてシーケンシングすれば、集団を代表するようなジョブ列を生成することが可能であり、(C) の条件を満たす。

4.3.2 スキーマモデル B: ジョブの処理位置による方法

本項では、遺伝的アルゴリズムでの符号化方法を参考にすべく、ブロックマイニングによる最適化手法に焦点を当てる。ブロックマイニングとは、解空間の構造を表すビルディングブロックに焦点を当てて解探索を効率化する手法である。Chang らは^[57]、フローショップスケジューリング問題の解を表すジョブのシンボル列に着眼し、列の中でのジョブの位置をビルディングブロックの候補とした符号化を行っている。特定のジョブが何番目に処理されるのか、という情報は、(b) 納期に基づく評価関数に強く影響される。

Chang らの方法に着想を得て作成したスキーマモデルは図 4.2 に示される。図 4.2 では、先ほどの DSM スキーマモデルと同様に、ジョブ順列 {1,2,4,3,6,5} を例に引いてスキーマモデルを生成している。スキーマモデルは行にジョブの番号、列にその順番を割り当てて、 $n \times n$ のマトリクスで構成される。ジョブ i を j 番目に実行する場合、 $\delta_{ij} = 1$ となる。空白は $\delta_{ij} = 0$ を表し、スキーマを 0 で固定することに対応している。行列をベクトルに変換したとしてもその一般性は失わないため、本研究では δ_{ij} をそのままスキーマとして取り扱い、スキーマモデル B と呼称する。

スキーマモデル B は次のような特徴を持つ。

- ジョブ数 n のあらゆるジョブ順列に対して $n \times n$ の行列として同じ形式で記述することができるため、その写像 f は単射である。また、行列の要素 δ_{ij} として矛盾が生じない次の制約

Design alternatives

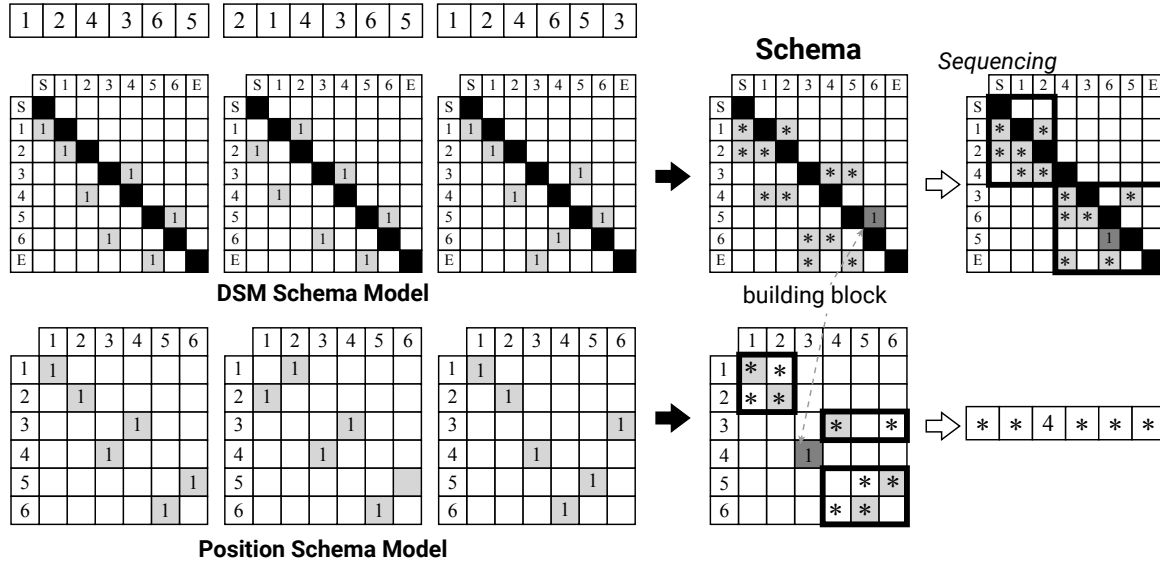


Fig. 4.3: Different ways of expressing alternatives and their populations.

を与えておけば、全単射の関係にあり、(A) の条件を満たす。

$$\sum_i^n \delta_{ij} = 1 \quad (j = 1, 2, \dots, n) \quad (4.2)$$

- 各ビットはビルディング候補として注目したジョブの処理タイミングに対応しており、ジョブ i をいつ遂行するのかの判断それ自体は独立しているため、(B) を満たしている。
- 集団のすべての要素で平均を計算することで、重心位置を定義することができ、Ward 法などのリンケージ手法を用いることができる。平均化したマトリクスは式 (4.2) を満たさないもので、スキーマの意味としては存在しないジョブ順列を含んでいるものの、それぞれのジョブについては処理タイミングの候補や傾向をスキーマの * から読み解くことができ、(C) の条件を満たしている。

図 4.3 は、 $\{1, 2, 4, 3, 6, 5\}$ に加え、 $\{2, 1, 4, 3, 6, 5\}$ 、 $\{1, 2, 4, 6, 5, 3\}$ を例に引いて、3 つの代替案の集合であるスキーマを 2 つの定義に従って変換している。図 4.3 上がスキーマモデル A、図 4.3 下がスキーマモデル B である。スキーマモデル A では、ほとんどの要素が 0 で固定されている。 $\delta_{5,6} = 1$ は唯一のスキーマ固定部分であり、全てに共通するビルディングブロックである。これは、ジョブ 6 の後にジョブ 5 を遂行することを意味している。シーケンシングによって代表的な構成 $\{1, 2, 4, 3, 6, 5\}$ が導き出されている。シーケンシング後のマトリクスから独立した 2 つの部分列を見出すことができ、ジョブ 4 以前のジョブ順序と以降のジョブ順序でそれぞれ分けて考えることができることが読み取れる。一方スキーマモデル B では、同様にほとんどの要素が 0 で固定されている一方で、 $\delta_{4,3} = 1$ が唯一のスキーマの固定部分であり、ジョブ 4 を 3 番目に遂行することを意味している。

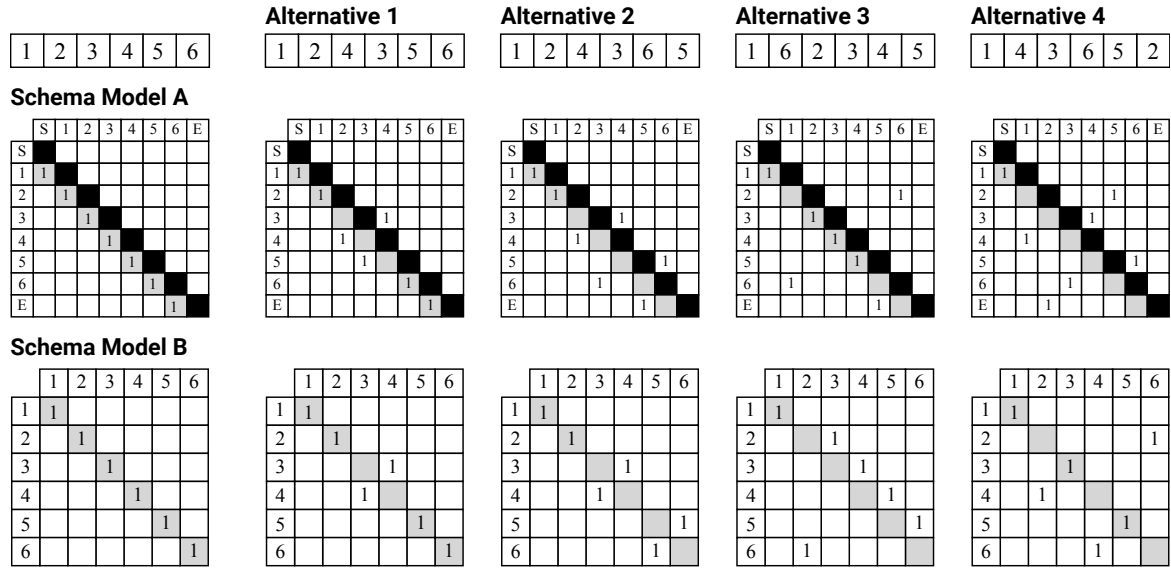


Fig. 4.4: Schema model representing alternatives.

こちらのスキーマでも同様の知見を獲得することができ、スキーマ内でのそれぞれのジョブの処理タイミングの候補の関係性を見ると、ジョブ 4 前後で独立していて、それぞれを分けて考えることがわかる。図 4.3 の例から、集団に対して似たような考察はできるものの、スキーマモデル A ならこの集団の特徴はジョブ 6 の後にジョブ 5 を遂行する集団としてラベルをつけることになるが、スキーマモデル B ではジョブ 4 を 3 番目に遂行する集団としてラベルをつけることになる。つまり、ビルディングブロックとして把握している部分要素は全く別であり、代替案の捉え方がスキーマモデルの定義に依存していることを確認できる。提案手法ではスキーマモデルに基づいて代替案の傾向や違いを理解することを狙っているため、スキーマの定義方法の影響は大きい。スキーマ定義もある種の抽象が行われているため、唯一の答えが存在しない「開いた問題」である。意思決定者が何を知りたいか、何を重視しているのか、などの嗜好情報を鑑みつつ、適切なスキーマ定義を行うことが望ましい。

4.4 スキーマモデルに合わせた類似性尺度の選択

本項では、実際の階層的クラスタリング実行に先立って、4.3 項で構築したスキーマモデルの意味の部分から、Step 2 での類似性尺度の選択について論じておく。どちらのスキーマモデルであってもビット列であるため、表 3.2 にまとめられた類似性尺度 s_{st} を用いる。なお、階層的クラスタリングで距離として用いる場合は $1 - s_{st}$ を計算する。3.5.1 項で言及したように、スキーマモデルの特性を踏まえて選択しなければならない。本項では、実際の階層的クラスタリング実行に先立って、類似性尺度の選択について論じておく。

6 つのジョブで構成される順列を例に引いて、それらの距離を計測することでスキーマモデ

ルと距離の関係を調査する。具体的には、 $\{1,2,3,4,5,6\}$ を基準に、4つの代替案 $\{1,2,4,3,5,6\}$ 、 $\{1,2,4,3,6,5\}$ 、 $\{1,6,2,3,4,5\}$ 、 $\{1,4,3,6,5,2\}$ との類似性尺度を計測する。ここでは、4つの代替案を順に、代替案 1、代替案 2、代替案 3、代替案 4 と呼称し、それぞれのスキーマモデルは図 4.4 にまとめられる。表 4.1 に代替案の類似性尺度の計算結果を示す。 S_1 から S_8 は類似性尺度なので、1 に近いほど似ていることを意味する。逆に、ユークリッド距離は小さいほど似ていることを意味している。

本研究で提案したスキーマモデルは共に相違点に対称性があるので、すべての結果で $(1,0)$ 、 $(0,1)$ の組の数 b と c の値は等しくなる。非対称性を前提とした指標である S_7 、 S_8 は S_6 と同じ値をとっていることが確認できる。以降では S_7 、 S_8 を除外することにする。

指標選択の上での重要な視点は、1 で一致する a と 0 で一致する d を区別するかである。区別しない場合は S_1 、 S_2 、 S_5 、区別する場合は、 S_2 、 S_4 、 S_6 を指標として用いることになる。それぞれのスキーマモデルの意味から検討すると、すべてのビットは選択の違いを表すため、1 と 0 は等価でなく、区別した指標を用いることが妥当であると考えられる。また、定量的な意味でも、これらのスキーマモデルは 1 で固定するビットの総数が固定されていて、その数も少量であるので、 d の値が極端に大きくなる傾向がある。計算結果からも分かるように、 d の値が大きい場合は類似度が高くなるため、相違点の感度が小さくなってしまう。これらの理由から、本研究は S_2 、 S_4 、 S_6 に焦点を当てることにする。3つの評価指標の違いは $(b+c)$ の値の感度の調整であり、スキーマモデルの特性を踏まえて検討する必要がある。まず、スキーマモデル A については、基準に対して代替案 1 と代替案 3 は等しい類似度であることに注意されたい。スキーマモデル A では 2つのジョブの順序関係のみを抽象しているので、あるジョブを抜き出してジョブの間に挿入するという操作によって得られる解はすべて同じ類似度になる。具体的には、代替案 1 はジョブ 3 をジョブ 4 とジョブ 5 の間に、代替案 3 はジョブ 6 をジョブ 1 とジョブ 2 の間に挿入している。挿入の視点で代替案を観測すると、代替案 2 は挿入操作を 2 回、代替案 4 は挿入操作を 3 回以上行った結果であることが分かる。ジョブの挿入の際には移動させるジョブの前後のブロックと挿入先の連結を表すブロックの、合計 3つのビットが変化する。スキーマの相違点は b と c とで 2度数えることになるため、それらを半分にする S_6 を適用することで、類似点と相違点を等価に扱うことができると考えられる。次に、スキーマモデル B については、ジョブの位置をブロックといて用いているため、 b あるいは c の数は、順番が一致していないジョブの数に相当する。スキーマモデル B でも相違点は b と c とで 2度数えることになるため、 S_6 を適用することが望ましいと考えられる。

モデルとしての特徴から考察すると S_6 を用いることが良いと結論づけたが、表 4.1 から、スキーマモデル A とスキーマモデル B は、それぞれの着眼点が違うことから、距離の値は異なっていることに注意する。まず、代替案 1 と代替案 2 は、基準となるジョブ順列からそれぞれ、1ヶ所、2ヶ所入れかえた順列である。入れ替え操作が増えるにつれてどちらのモデルでも類似性が単調減少していることを観測することができるためこれらは、直感に反しない結果であると考えられる。一方

Tab. 4.1: Distances between $\{1,2,3,4,5,6\}$ and alternatives.

Measure	Schema model A			
	Alternative 1	Alternative 2	Alternative 3	Alternative 4
	$\{1,2,4,3,5,6\}$	$\{1,2,4,3,6,5\}$	$\{1,6,2,3,4,5\}$	$\{1,4,3,6,5,2\}$
(a, b, c, d)	$(4, 3, 3, 46)$	$(2, 5, 5, 44)$	$(4, 3, 3, 46)$	$(1, 6, 6, 43)$
S_1	0.893	0.821	0.893	0.786
S_2	0.400	0.167	0.400	0.076
S_3	0.806	0.697	0.806	0.647
S_4	0.250	0.091	0.250	0.040
S_5	0.943	0.902	0.943	0.880
S_6	0.571	0.286	0.571	0.143
S_7	0.571	0.286	0.571	0.143
S_8	0.571	0.286	0.571	0.143
Euclidean distance	2.45	3.16	2.45	3.46
Measure	Schema model B			
	Alternative 1	Alternative 2	Alternative 3	Alternative 4
	$\{1,2,4,3,5,6\}$	$\{1,2,4,3,6,5\}$	$\{1,6,2,3,4,5\}$	$\{1,4,3,6,5,2\}$
(a, b, c, d)	$(4, 2, 2, 28)$	$(2, 4, 4, 26)$	$(1, 5, 5, 35)$	$(3, 3, 3, 27)$
S_1	0.889	0.778	0.783	0.833
S_2	0.500	0.200	0.091	0.333
S_3	0.800	0.636	0.643	0.714
S_4	0.333	0.111	0.048	0.200
S_5	0.941	0.875	0.878	0.909
S_6	0.667	0.333	0.167	0.500
S_7	0.667	0.333	0.167	0.500
S_8	0.667	0.333	0.167	0.500
Euclidean distance	2.00	2.83	3.16	2.45

で、代替案 3 と代替案 4 はスキーマによって類似度が大きく異なっている。上述の通り、スキーマモデル A では代替案 3 はジョブの挿入として理解できる一方、スキーマモデル B では、1 番目の

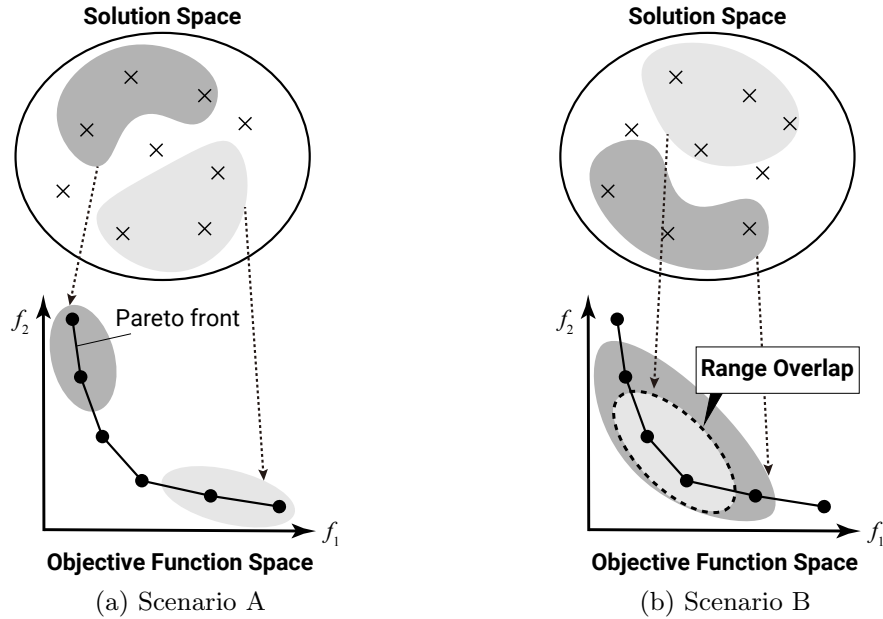


Fig. 4.5: Location of the two clusters in the objective function space:(a)trade-off relationship between evaluation values and (b)diversity not acting on the evaluation value.

ジョブ以外の全てが異なっていると判断する。逆に、代替案 4 についても、スキーマモデル A では異なる一方で、スキーマモデル B は似ていると判断している。これらの傾向から、どちらのスキーマモデルが優れているという結論に至ることはできず、あくまでも、何に焦点を当てるのか、という方針の違いでしかないことに注意したい。意思決定の方針に沿って、スキーマを柔軟に選択することが必要であることが距離の計測結果からも理解することができる。

4.5 目的関数空間の可視化によるビルディングブロックの解釈

Step 3 では階層的クラスタリングの結果から解空間構造の分析とその構造的相違点の意味を考察する。スキーマ定義や距離定義の際に議論したように、両スキーマモデルでビットの 0 一致が多いことを受け、本章ではビルディングブロックを表す Category A として 1 一致のみに焦点を当てることにする。デンドログラムの分岐の意味は、Category B として現れる構造特徴であるレバレッジポイントに着目しつつ、スキーマモデルそもそもの意味やその他評価指標を導入することで、意思決定者の手で追加特徴を検討する。フローショップスケジューリング問題は目的関数の値を選定の基準に用いることが適していることを受けて、評価指標の値に基づく特徴理解を目指す。

解空間上で 2 つに分けられたクラスタの目的関数空間での分布は、図 4.5 に示される 2 つのシナリオを想定することができ、それぞれの意味は次のように解釈可能である。

- (a) シナリオ 1: 左右のクラスタで目的関数の配分に明確に差がある場合、分析によって分岐理由として特定されたビルディングブロックの相違点 (構造特徴の左右の違い) は目的関数値

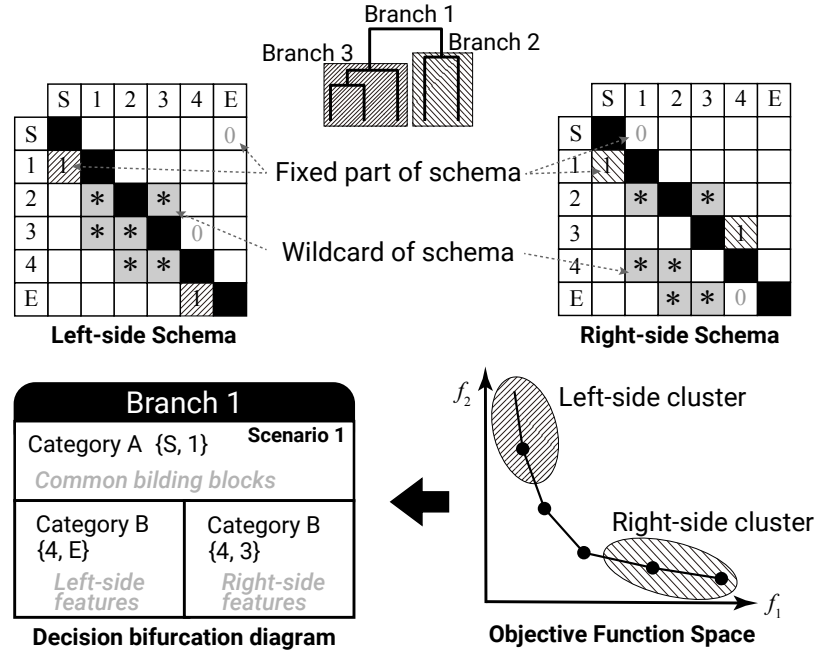


Fig. 4.6: Results of exploration of solution space and template of the decision bifurcation diagram.

の向上に寄与する要素として判断することができる。左右を判断する意思決定者にとってはトレードオフの配分を検討する。

- (b) シナリオ 2: 左右のクラスターで目的関数空間上の範囲に重複がある場合、ビルディングブロックの相違点は、同等の目的関数値を出すための組合せ方式の違いであると判断することができる。言い換えると、等価解のように似た目的関数を実現する方法が複数あることを意味する。意思決定者にとっては代替案の多様性を理解するに重要な要素であり、Category C の要素なども参照しながら方式を選択する。

上記の 2 つのシナリオの境界には明確な基準がなく、曖昧であるため、意思決定者の主観に依存することは忘れてはならない。それぞれのクラスターの目的関数の代表値や目的関数空間上の分布、その時点でのビルディングブロックの内容をもとに意思決定者が総合的に判断しなければならず、これらはあくまでも考察の指針に過ぎないことに注意しなければならない。

これらの議論を踏まえて、図 4.6 のような意思決定分岐図のテンプレートを用いて情報を整理する。図 4.6 では 4 つのジョブ列をスキーマモデル A で表現した場合の結果である。Branch 1 では代替案を 2 つのクラスターに分割し、それぞれのクラスターを表すスキーマを作成することができる。順列構造では、共通ビルディングブロックとして Category A のうち、1 で固定したビットのみ、意思決定分岐図に記録する。左右の構造特徴の違いとしては Category B の要素を記録するが、(1,0) の組のうち、1 として固定している側に特徴として記録する。図 4.6 の場合であれば、左のクラス

タでは $\delta_{6,5} = 1$ で固定されており、全ての代替案で最後に 4 のジョブを処理することが分かる。一方、右クラスタでは $\delta_{5,6} = 0$ であり、どの代替案でも 4 は最後に処理しないことを意味する。この情報は解空間構造の違いとして左の特徴として扱われる。同様に、 $\delta_{4,5}$ について右の特徴がまとめられている。さらに、これらの構造の違いの解釈のため、目的関数空間を可視化した上で、ふさわしいシナリオを意思決定者の目で判断し、記載される。図 4.6 では、シナリオ 1 として判断される。この結果から、ジョブ 4 を最後に行えば評価指標 f_1 の最小化に寄与できる一方で、ジョブ 4 の後にジョブ 3 を行えば、 f_1 は悪化するが、 f_2 の値を改善できることが分かる。

4.6 事例：順列フローショップスケジューリング問題

4.6.1 概要

本節では、フローショップスケジューリング問題の中でも最も基本的な問題の 1 つである、順列フローショップスケジューリング問題をベンチマークとして例に引き、事例を通して提案手法の有効性を確認する。

まず、本事例で取り扱うベンチマーク問題を定式化し、メタヒューリスティック手法による代替案の生成を行う。次に、構造分析法の中核を担う階層的クラスタリングでの距離定義やリンケージ手法による結果の違いを予備検討として調査する。最後に、4.3 項で導入した 2 つのスキーマモデルを用いて、解空間の構造分析を実施する。構造分析を通して、本研究で指摘した課題点の妥当性を確認した後、前提にあたるビルディングブロック仮説を検証する。そして、ある代替案に至る経緯を、ビルディングブロックの集積として再構成できることを確認し、2 つのスキーマ定義でのその解釈の違いの特徴をまとめた上で、それぞれの効果を論ずる。

4.6.2 問題設定と分析の準備

順列フローショップスケジューリング問題は、 n 個のジョブの順序を決定する組合せ最適化問題である。それぞれの機械でジョブの処理時間は異なっており、 m 台全ての機械は同じ順序でジョブを遂行する。したがって、順列フローショップスケジューリングの最適化問題の設計変数はジョブの順序である。上述のように目的関数を多様に設定することができ、生産管理者は多目的環境において意思決定を行わなければならない。本事例では、次式で表される Makespan C_{max} と Tardiness T の 2 目的の最適化問題に取り組む。ただし、 C_i はジョブ i の完了時間、 T_i はジョブ i の納期遅れ時間である。ジョブ i の納期を d_i とすれば、納期遅れは $T_i = \max \{C_i - d_i, 0\}$ で定義される。

$$C_{max} = \max \{C_i | i = 1, 2, \dots, n\} \quad (4.3)$$

$$T = \sum_{i=1}^n T_i \quad (4.4)$$

本事例では、Taillard のベンチマーク問題^[78]の中から問題の規模の異なる2つの問題を例に引く。引用する順列フローショップスケジューリング問題のベンチマークのうち、20 仕事 5 機械の問題をベンチマーク 1、50 仕事 20 機械の問題をベンチマーク 2 と呼称する。それぞれでのジョブの納期 d_i は、ジョブ番号の昇順で処理した際のそれぞれの完了時間として設定した。つまり、ジョブを昇順で処理した際の T_i は 0 である。Makespan を工場させるにはジョブの順序を変える必要がある一方で納期遅れが発生する恐れがあり、これらの2つの目的関数にはトレードオフの関係が存在している。

以上より、本事例での最適化問題は次のように定式化される。

$$\begin{aligned}
 & \text{find} && x_{ij} \quad (i = 1, \dots, n, j = 1, \dots, n, i \neq j) \\
 & \text{minimize} && f_1 = C_{max} \\
 & && f_2 = T \\
 & \text{subject to} && \sum_{j:j \neq i} x_{ij} = 1 \\
 & && \sum_{i:i \neq j} x_{ij} = 1 \\
 & && \sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \quad (S \subset \{1, \dots, n\}, S \neq \phi) \\
 & && x_{ij} \in \{1, 0\}
 \end{aligned} \tag{4.5}$$

ただし、 n はジョブの総数であり、 $x_{ij} = 1$ とはジョブ i の次にジョブ j を遂行することを意味する。また、3つ目の制約条件は部分巡回を回避するための制約である。この定式化はスキーマモデル A に準拠している。本質的には、最適化問題の設計変数はジョブの順序であり、遺伝的アルゴリズムなどでは、プレプロセッシングの時点でジョブの順序をシンボル列として符号化することが一般的である。この定式化はあくまでも1つの例であり、唯一の形式ではないことに注意されたい。

解空間構造の分析に用いる代替案集合を獲得するために、順列フローショップスケジューリング問題に対して、Makespan と Tardiness の多目的最小化問題を解いた。探索アルゴリズムには Genetic local search^[79] と NSGA-II^[2] を連携させたアルゴリズム^[80] を用いた。ベンチマーク 1 とベンチマーク 2 の最適化結果の散布図を図 4.7 に示す。ベンチマーク 1 からは 17、ベンチマーク 2 からは 61 のパレート最適解を獲得し、それぞれの代替案を Makespan の昇順に番号を割り当てた。以降の予備検討及び構造分析では、これらの代替案の集合を分析の対象とする。

4.6.3 類似性尺度選択に関する調査

4.4 節では、2つのスキーマモデルの定義に基づいて、バイナリデータの距離関数の意味を整理した上で類似性評価尺度のうち S_6 を用いることが適切であると結論づけた。ここでは、スキーマ定義からではなく階層的クラスタリングの側から、距離選択の影響を調査する。本項では、階層的

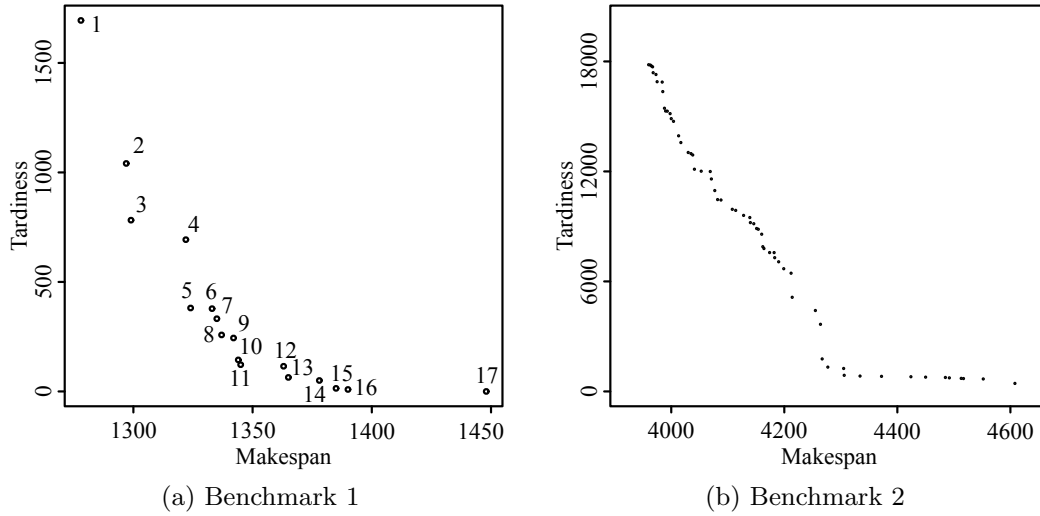


Fig. 4.7: Scatter plot of Pareto front.

クラスタリングのリンケージ手法には郡平均法を用いる。また、デンドログラムの描画の際には、分岐の枝葉の順序を反転させる操作を加え、隣接する枝葉で類似度を最大化する並び替え操作を行った。

まず、相違点に類似性を持つバイナリデータのための6つの類似性尺度とユークリッド距離を用いてベンチマーク1を対象に階層的クラスタリングを実行した。スキーマモデルAとスキーマモデルBのそれぞれでクラスタリングを行い、距離の定義をさまざまに変えた結果を図4.8、図4.9に示す。17のパレート最適解はデンドログラムによって段階的に分割され、16の分岐点で分けられる。それぞれの分岐位置でのクラスタ間距離はリンケージ手法によって計測され、縦軸に記載されている。

4.4節で論じたように、スキーマモデルAおよびBは0一致数を表す指標 d の値が大きくなる傾向にあることから、 d を加味する指標である S_1, S_3, S_5 を用いた結果の距離が小さくなっていることが確認できる。デンドログラムの分割過程に焦点を当てると、スキーマAにおいては、 S_1, S_3, S_5, S_6 を用いた結果は同じ形状であることがわかる。また、枝葉順序の反転を考慮しなければ、 S_2 とユークリッド距離を用いた結果も、 S_1, S_3, S_5, S_6 を用いた結果と同じ分割過程を辿っている。 S_4 のみ、他の類似性尺度と分割の方式が異なっている。他の代替案ではBranch 4で代替案5と代替案6を同じクラスタに分割している一方で、 S_4 では代替案5のみを除外している。これは、 S_4 が6つのバイナリデータ指標の中で最も類似点に対する相違点の感度が高いことが結果に作用していると考えられる。スキーマBでは、 S_2, S_4 、ユークリッド距離の組と S_1, S_6 の組でのデンドログラム形状が一致している他、枝葉順序の反転を考慮しなければ、 S_1, S_6 の組と S_5 を用いた結果が一致している。一方で S_3 を用いた結果は代替案10の取り扱いが独特であり、他の結果とは違って、スキーマAと比較しても多様な結果が得られていることがわかる。

次に、問題の規模が一段大きくなった、ベンチマーク2において同様の操作を行った結果を図

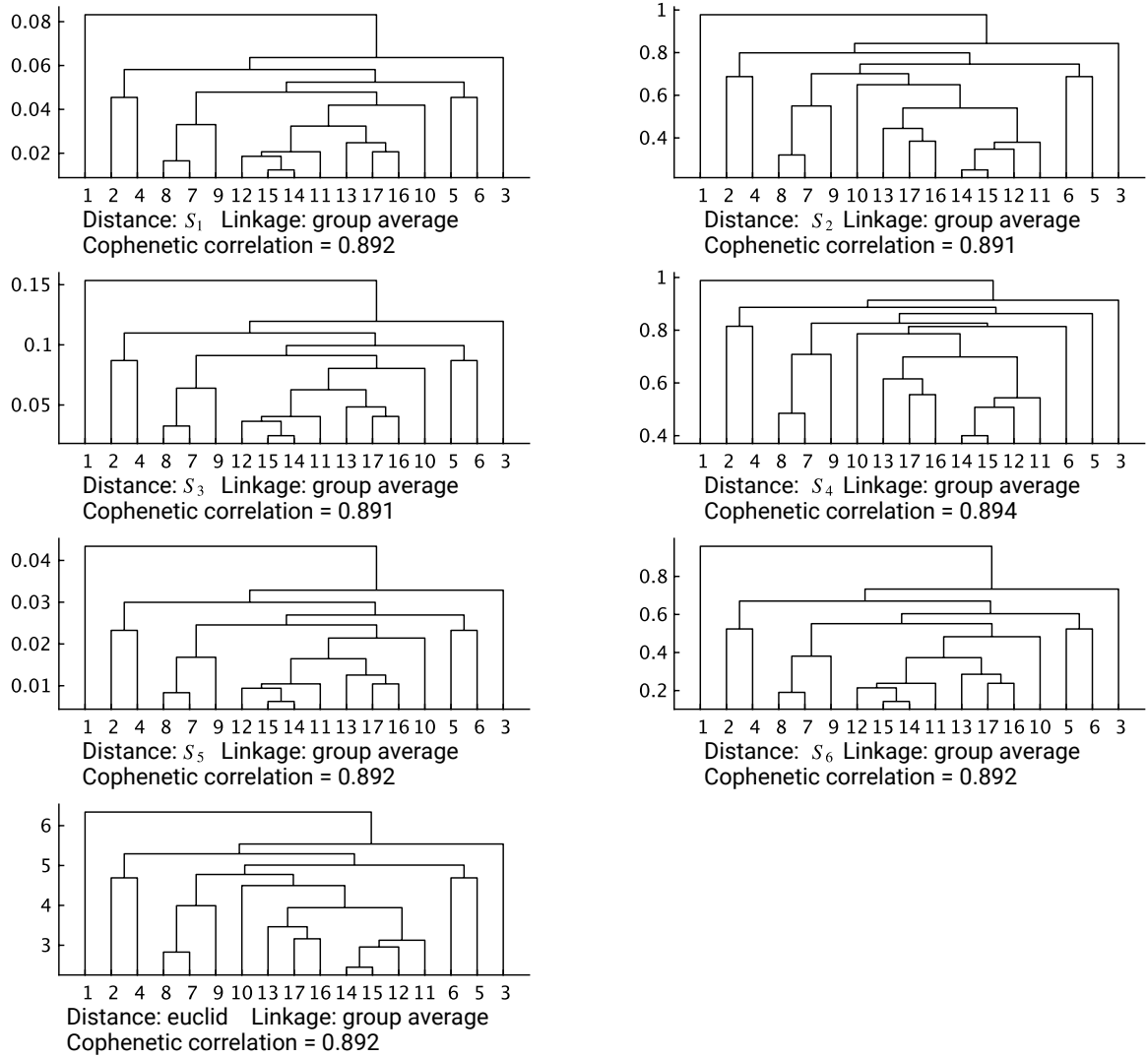


Fig. 4.8: Dendrograms by definition of distance in Benchmark 1 with schema model A.

4.10 及び、図 4.11 に示す。61 のパレート解は 60 の分岐点で階層的に分類される。

ベンチマーク 1 と同様に、デンドログラムの縦軸から、 d を加味した指標である S_1, S_3, S_5 の距離が小さくなっていることが確認できる。また、デンドログラムの分割過程に焦点を当てると、スキーマモデル A では代替案 3 と代替案 4 の分割過程が S_5, S_6 のみ異なっているものの、分割過程が 7 つの結果でほとんど一致していることを確認できた。一方、スキーマモデル B では、ベンチマーク 1 と同様に、 S_2, S_4 、ユークリッド距離のデンドログラム形状が同じことを確認することができるほか、 S_1 と S_3 の組と S_5 と S_6 の組の 3 つのパターンを確認できた。後者 2 つはデンドログラムの分割過程はほとんど一致しているものの、スキーマモデル A と同様に代替案 3 と代替案 4 の分割過程が異なっている。前者と後者 2 つとは代替案 45 の取り扱いが異なっている。これらの結果から、問題の規模が大きくなったとしてもスキーマ A の結果は距離定義に影響されない一方、スキーマ B の結果は距離定義に応じて姿を変えていることがわかる。

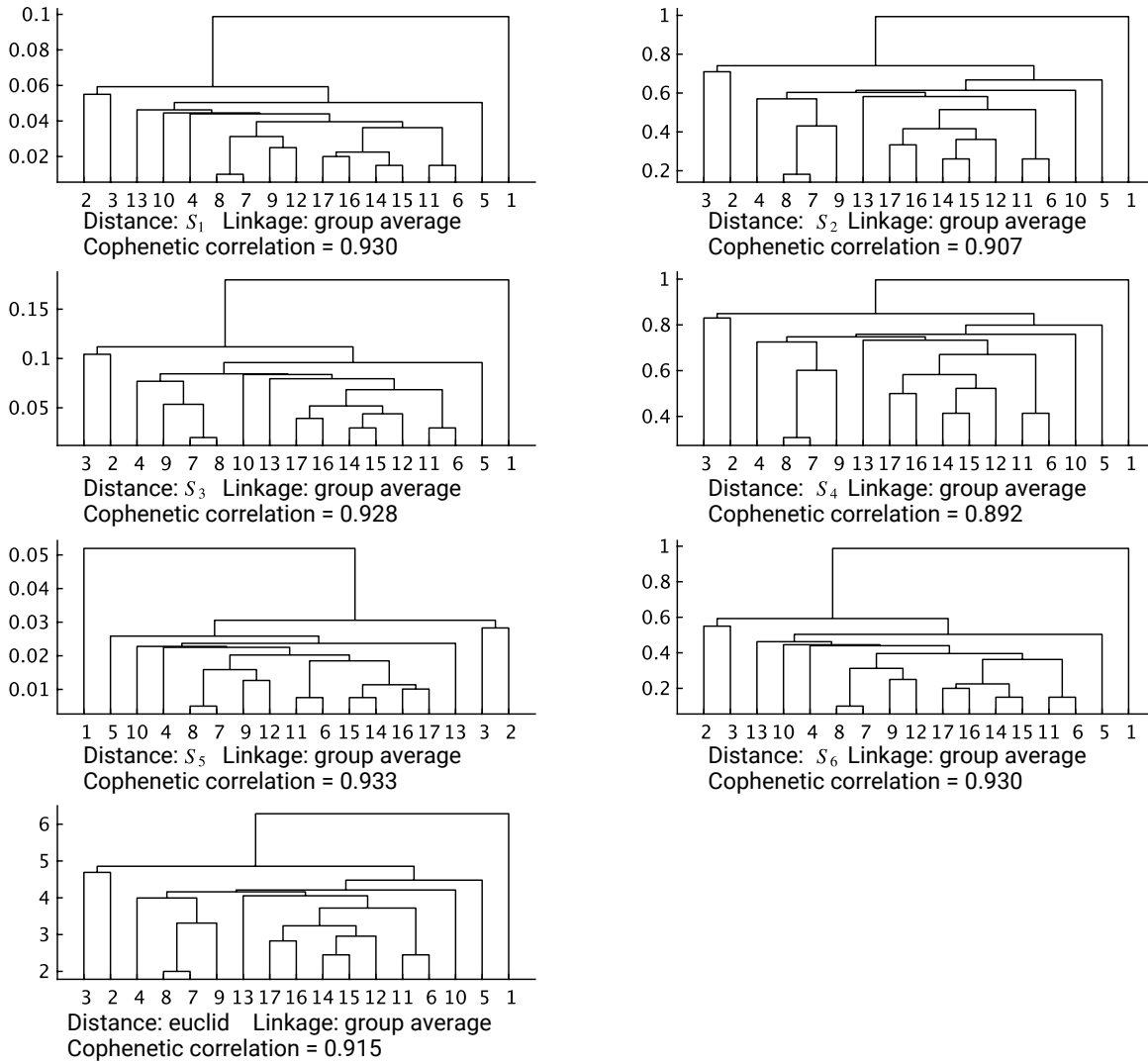


Fig. 4.9: Dendrograms by definition of distance in Benchmark 1 with schema model B.

以上の4つの結果から、距離関数がデンドログラムの形状に影響するかはスキーマの定義によってさまざまであることがわかる。結果の分割過程の違いは比較検討から理解することができるものの、これらのデンドログラムの形状のみからは距離の定義の良し悪しを判断することはできないため、階層的クラスタリングの結果の他に追加の選定根拠が必要であることがわかる。例えば、上述のようにデンドログラム形状の違いがどの代替案の取り扱いによるものかは上述の比較検討から特定できているので、言語表現や認識のレベルでの類似性とデンドログラムでの関係を照らし合わせて議論することで、どの結果を採用するかを判断することができると考えられる。また、4.4項で論じたようにスキーマモデルの特性を活かして類似性尺度の選択を判断することも重要な視点の1つであると考えられる。

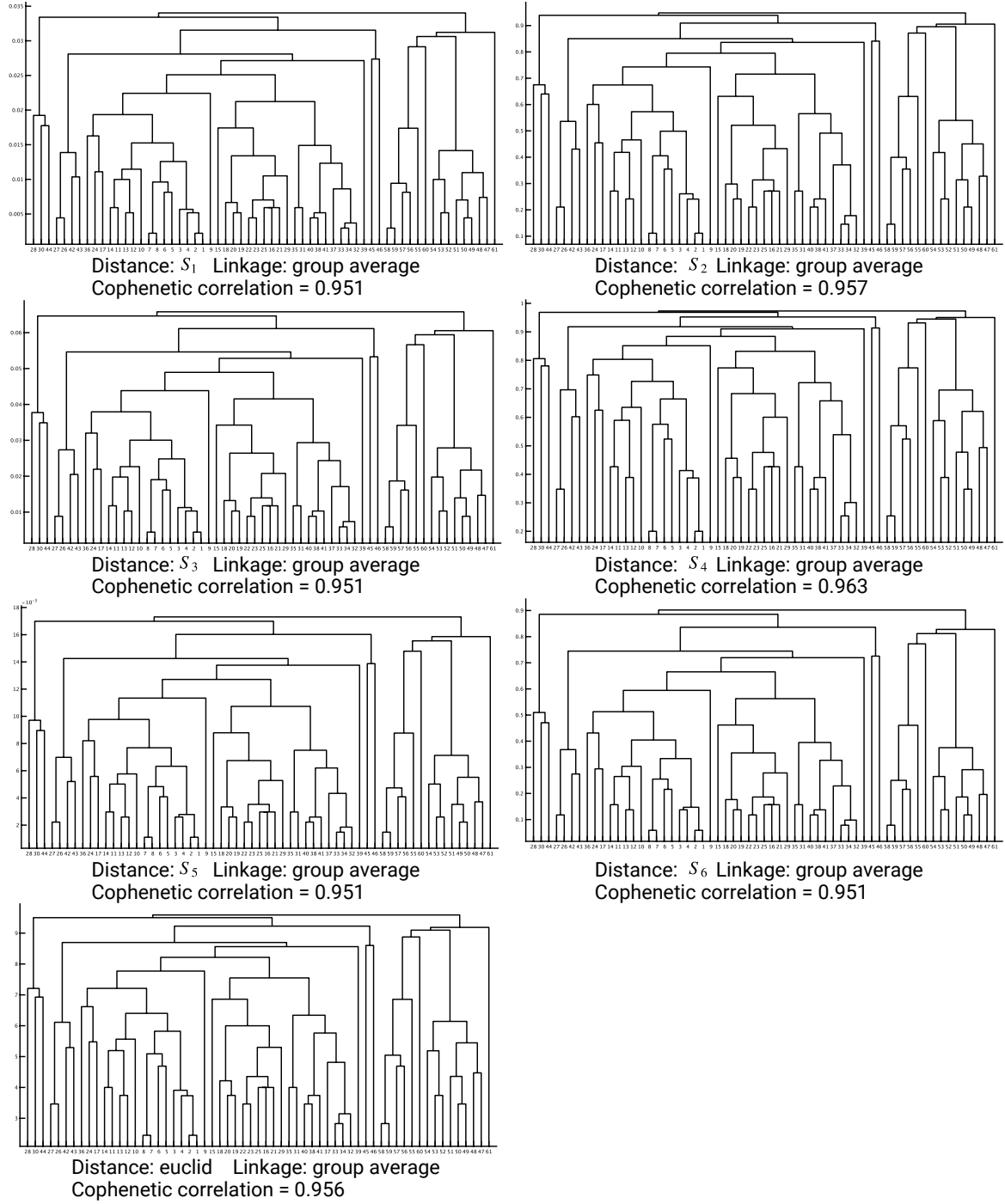


Fig. 4.10: Dendrograms by definition of distance in Benchmark 2 with schema model A.

4.6.4 リンケージ手法選択に関する調査

本項では、リンケージ手法による結果の違いを調査する。リンケージ手法は階層的クラスタリングの併合方法そのものであり、解空間の分割過程に直結するため、本研究で提案する解空間の構造

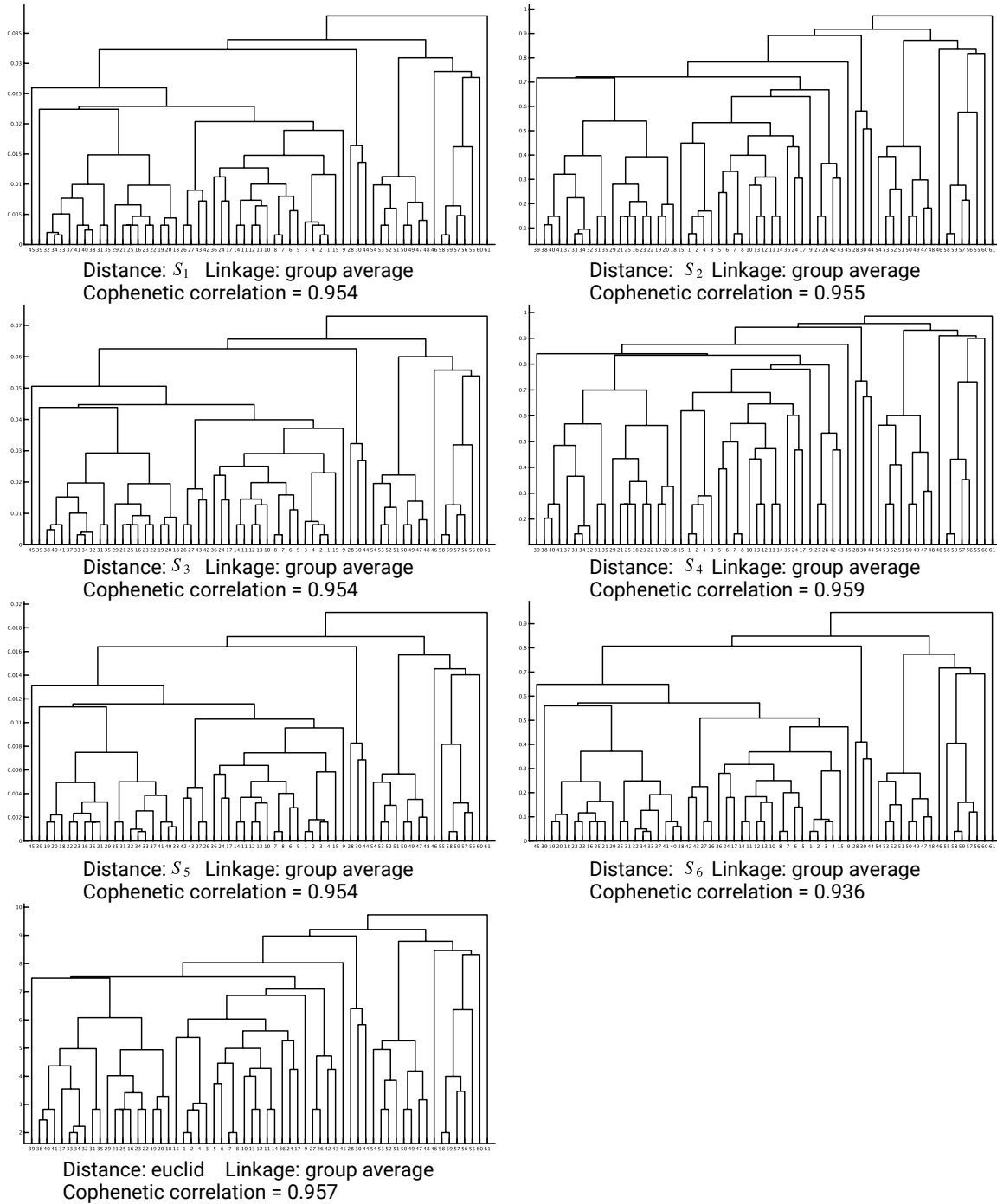
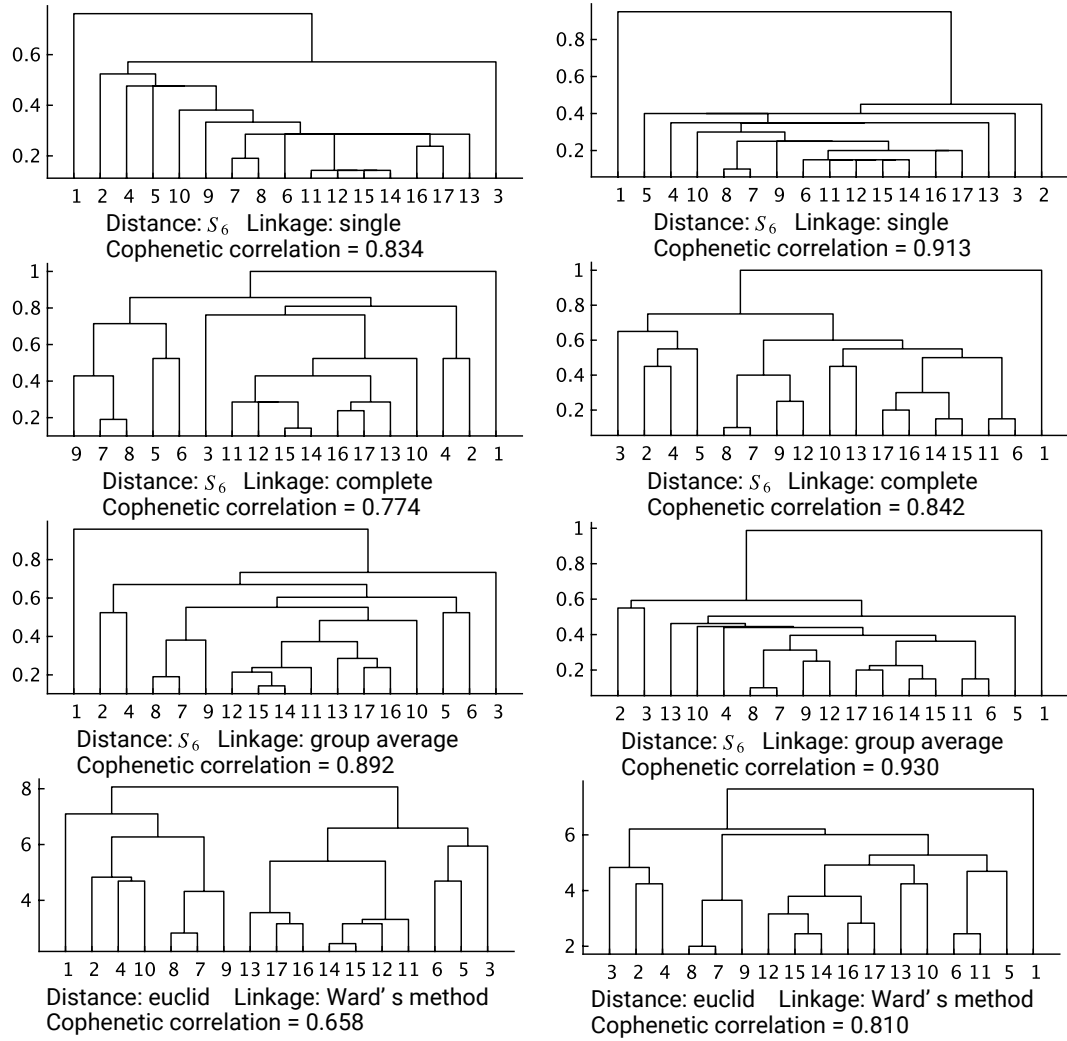


Fig. 4.11: Dendrograms by definition of distance in Benchmark 2 with schema model B.

分析の結果に大きく影響する。類似性尺度の選択と同様に、どの手法を用いるかを決める統一的な指標は存在しないものの、リンケージ手法によってデンドログラムの形状にどのように影響するかを、本検討を通して確認する。

4.3 節で提案した 2 つのスキーマモデルに対して、5 つのリンケージ手法を適用した。なお、4.6.3 項から、類似性尺度のデンドログラムでの分割過程への影響が小さいことを受けて、4.4 節で論じた



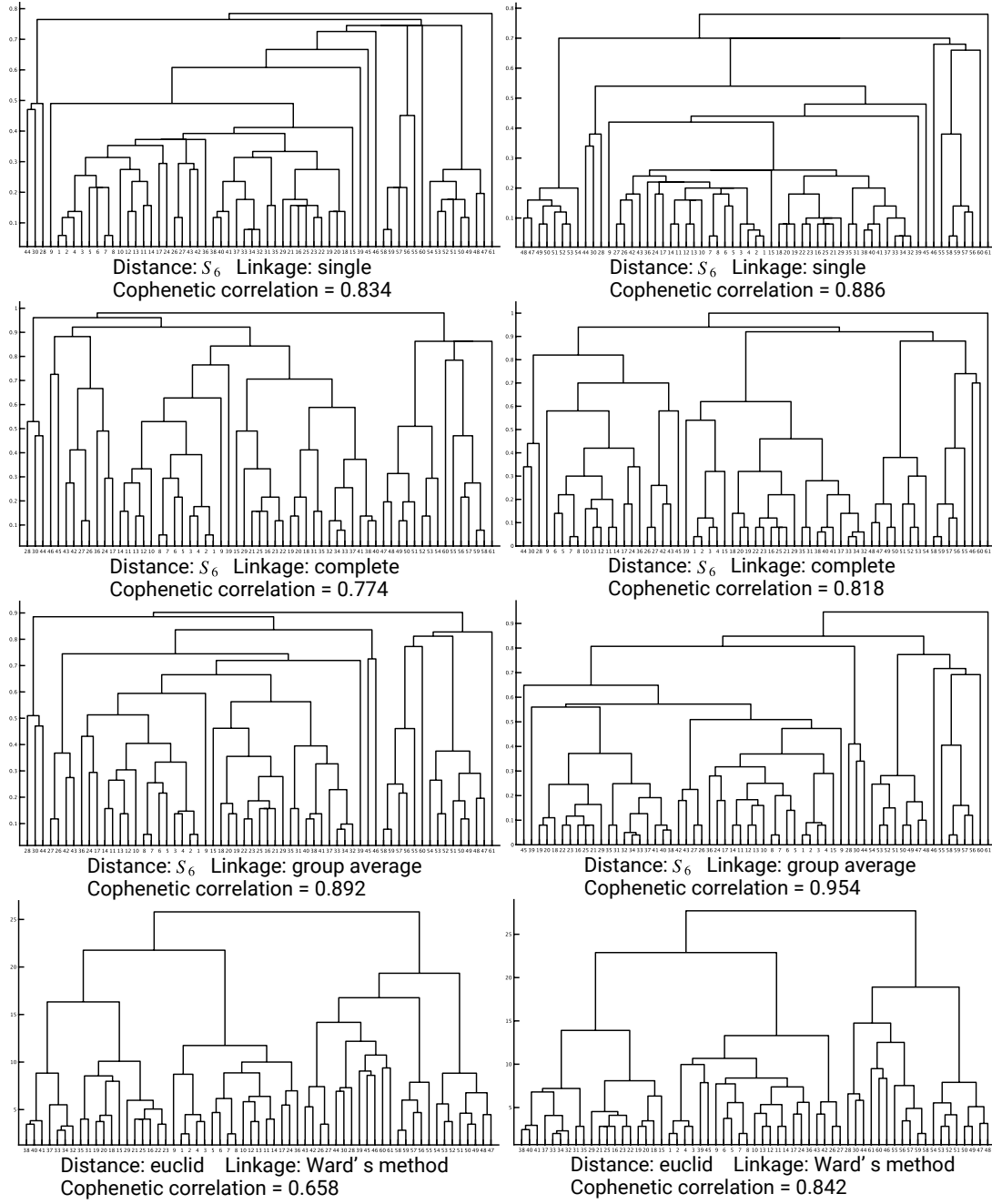
(a) Schema model A

(b) Schema model B

Fig. 4.12: Dendrograms by definition of linkage method in Benchmark 1: (a) schema model A and (b) schema model B.

スキーマモデルの特性から、本項では S_6 を類似性尺度に採用した。また、ウォード法ではスキーマの平均値をクラスターの重心として定義し、ユークリッド距離を用いて階層的クラスタリングを実行した。ベンチマーク 1 での計算結果は図 4.12 に、ベンチマーク 2 での計算結果は図 4.13 に示される。スキーマモデルの種類を問わず、リンケージ手法によってデンドログラムの形状が異なり、コーフェン相関係数の値もさまざまであることが確認できる。

最短距離法や最長距離法は、その距離測定法の性質から外れ値の影響を受けやすいことや、鎖効果や拡散現象を引き起こしやすいと言われており、望ましくない結果を導くことが報告されている^[71]。例えば、鎖効果とはある 1 つのクラスターに 1 つずつ順番に吸収されながらクラスターが形成される現象のことで、分類というよりは例外を除くような分割過程であることから、分析結果



(a) Schema model A

(b) Schema model B

Fig. 4.13: Dendrograms by definition of linkage method in Benchmark 2: (a) schema model A and (b) schema model B.

が判断しにくい。図 4.12 や図 4.13 でも最短距離の結果で、鎖効果を部分的に観測することができる。このようなリンケージ手法による恣意的な結果を引き起こさないため、郡平均法やウォード法を用いることが用いられることが一般的である。例えば、ウォード法はクラスター内の分散が最も

小さくなるようにクラスターを併合する。分類感度が高く、データのばらつきが抑えられる特徴を持つため、図 4.13 の結果のように、代替案が多量になる場合でもデンドログラムが見やすく、可読性が高い。

解空間の構造分析の視点では、デンドログラムの可読性や解釈の容易さは重要な手法選定の指針の 1 つであることは先に述べた通りである。本研究では、それぞれの分岐点での左右のスキーマでの共通あるいは相違する部分構造から、代替案がどのような重要な部分によって構成されているかを明らかにしようとしているため、鎖効果が発生しているデンドログラムのように、左右のクラスターで個体数に極端な差がある場合はその相違点を見出すことが困難になる。個体数の少ないクラスターのスキーマは固定部分が多量になることが予見でき、もう一方の集合のビルディングブロックを効果的に抜き出すことができなくなる。

4.6.5 解空間の構造分析結果

本節で取り組んできた、類似性尺度やリンケージ手法の選択による結果への影響を鑑みつつ、本項ではベンチマーク 1 のいくつかの階層的クラスタリングの結果に焦点を当て、構造分析を行い、結果を比較する。

まず、デンドログラム形状が距離の定義によって左右されないことから、スキーマモデル A に焦点を当てて構造分析を行った。リンケージ手法による結果の違いを確認するために、類似性尺度に S_6 を用いて郡平均法でクラスタリングを行った結果と、類似性尺度にユークリッド距離を用いてウォード法でクラスタリングを行った結果に焦点を当てる。それぞれの分析結果である意思決定分岐図を図 4.15、4.14 に示す。

次に、スキーマモデルの違いでの構造分析結果の違いを調査するため、スキーマモデル B でのウォード法クラスタリングを行った結果を取り上げ、構造分析を行った。作成された意思決定分岐図を図 4.16 に示す。

4.6.6 考察

前項で得られた結果において、提案する解空間の構造分析法の適用により、代替案の解釈と意思決定をどのように進めていくことができるかを考察する。まず、2.4.3 項で取り上げた組合せ型の最適設計が抱える課題を本ベンチマーク問題を通して確認した上で、提案手法の有効性を考察する。次に、提案手法の中核にあるビルディングブロック仮説に立ち戻って、代替案が重要な部分別の集積で構築されていることを例を挙げて確認し、抽出したビルディングブロックの妥当性を確認する。最後に、探索木を用いた代替案の選定プロセスを具体例を挙げて確認し、スキーマモデルの定義や階層的クラスタリングの方法で意思決定がどのように変化するかを考察する。

2.4.3 項では、組合せ型の最適設計では、目的関数の値は類似しているものの構造が異なる等価解

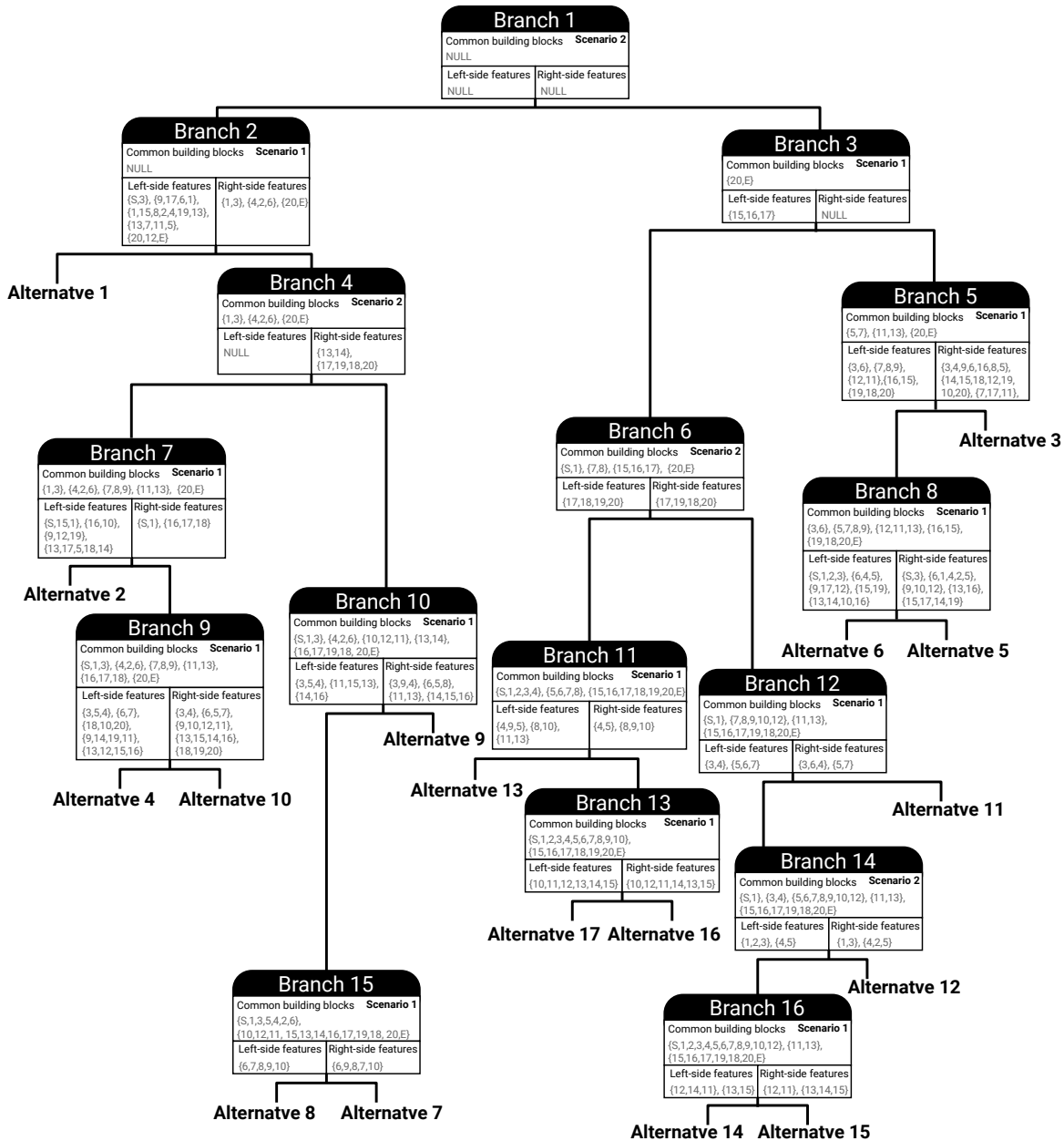


Fig. 4.14: Decision bifurcation diagram based on Ward's method with schema model A.

が存在していることから、従来の目的関数に基づく意思決定が機能しないことを指摘した。本項ではまず、図 4.7(a) の散布図での代替案 10 と代替案 11 の関係に焦点を当てて考察を行う。散布図からわかるように、2 つの代替案の目的関数値は類似しており、等価解とみなすことができる。しかし、解空間構造に焦点を当てた分析結果のいくつかでは、それらは異なるクラスタに分類されている。具体的には、図 4.14 の意思決定分岐図では初めの分岐である Branch 1 で別のクラスタにそれぞれが分類されている。したがって、目的関数に基づく判断では似ていると判断されるような代替案でも、解を構成する構造そのものに焦点を当てるとその性質が異なっていることがあり、従来の目的関数に基づく分析法では認識することができなかった代替案の特徴であると考えられる。ま

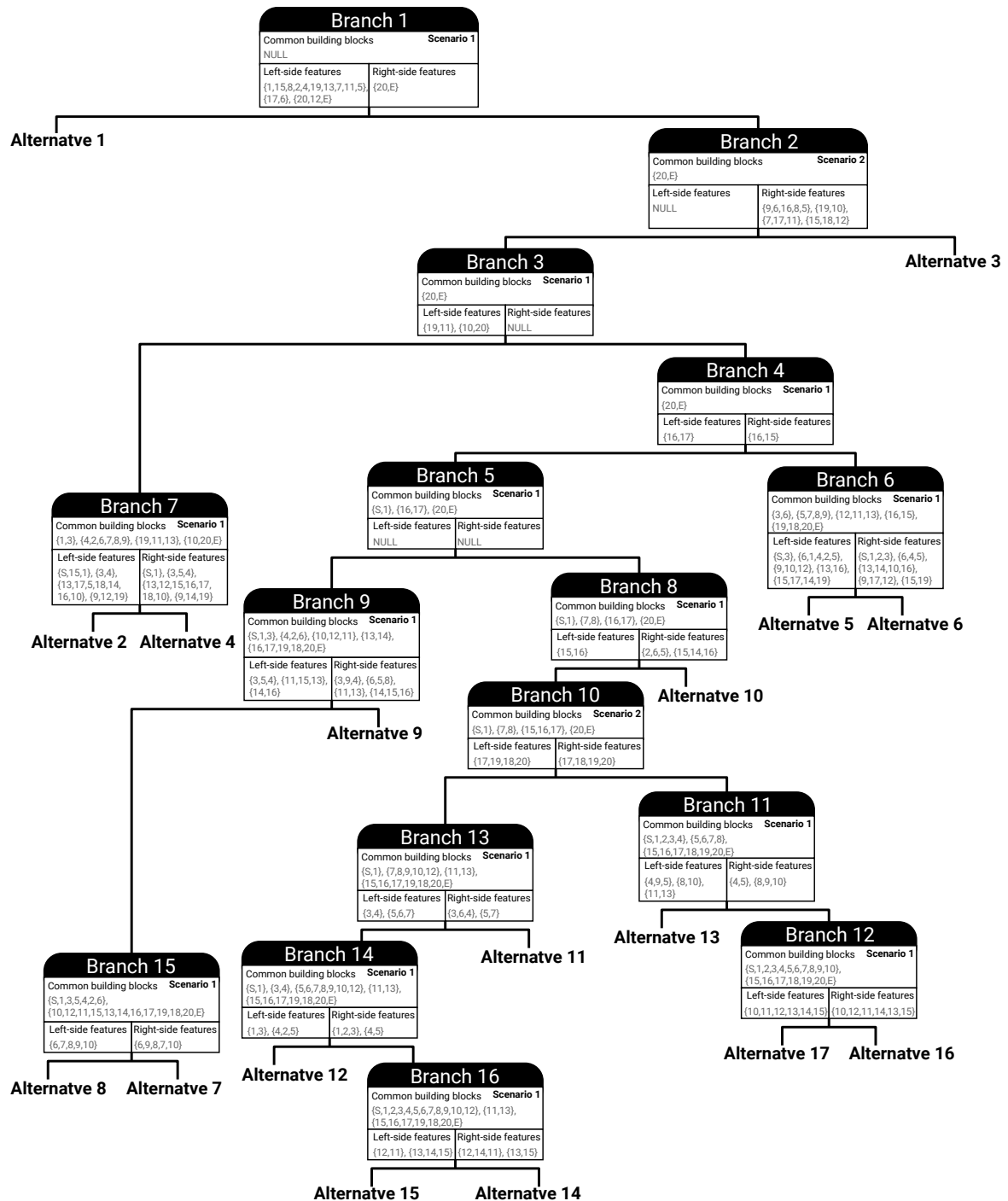


Fig. 4.15: Decision bifurcation diagram based on group average linkage with schema model A.

た、2.4.3 項で指摘した、単純な比較検討から意思決定の介入点を明らかにすることができない課題についてもこの代替案から確認する。2 つの代替案のジョブの順列はそれぞれ次のように表される。なお、それぞれのジョブの相違点は括弧で表される。

代替案 10 {1, (3, 4, 2, 6), 5, 7, 8, 9, 10, 12, 11, 13, (15, 14), 16, 17, (18, 19), 20}

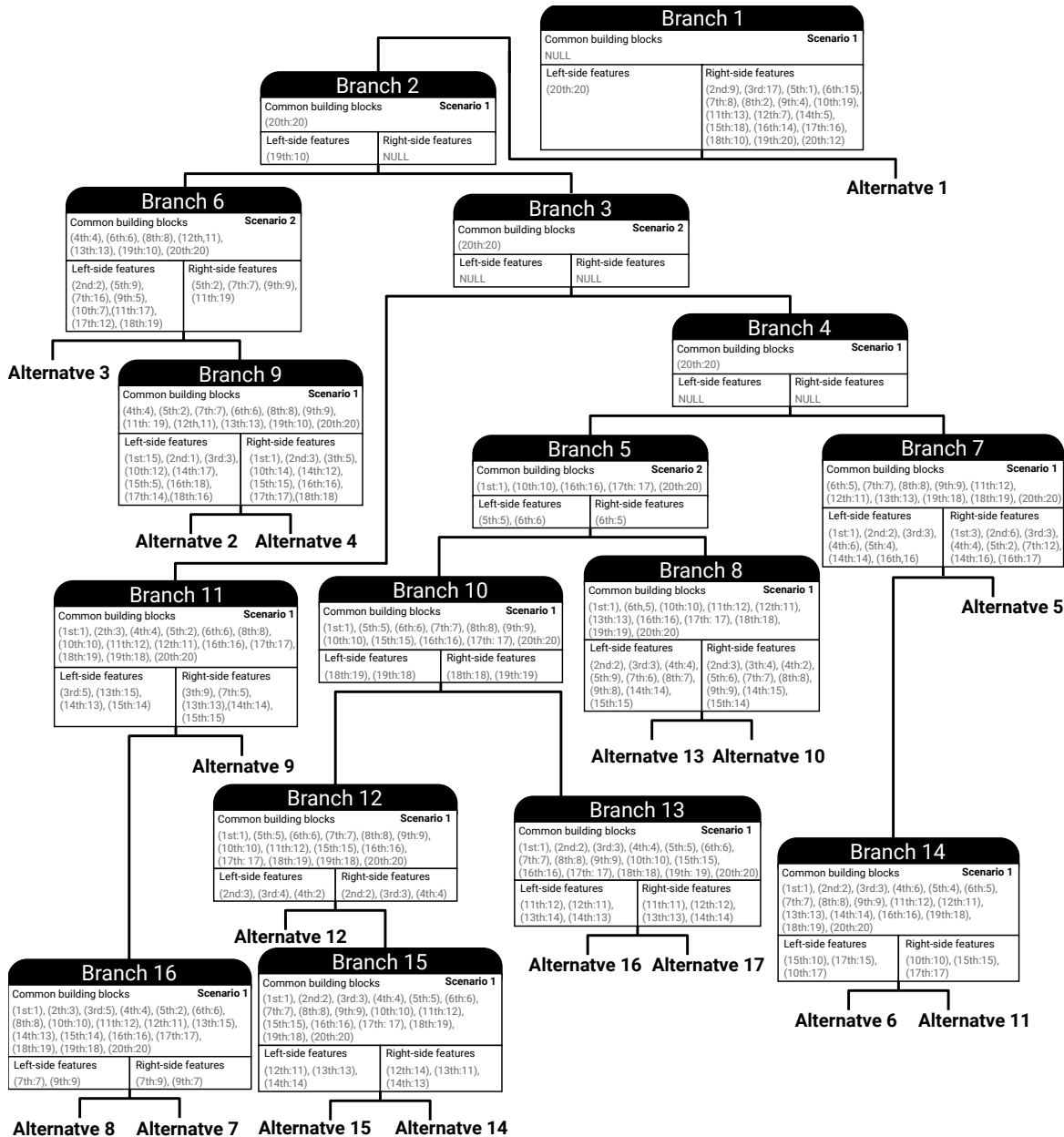


Fig. 4.16: Decision bifurcation diagram based on Ward's method with schema model B.

代替案 11 {1, (2, 3, 6, 4), 5, 7, 8, 9, 10, 12, 11, 13, (14, 15), 16, 17, (19, 18), 20}

2つのジョブは、ジョブ 2,3,4,6, の順序、ジョブ 14,15 の順序、ジョブ 18,19 の順序の3つの相違点が存在している。以降それぞれを相違点 A、B、C と呼称する。これらの文字列を単純に比較するだけでは、それぞれの意味は曖昧であり、どの相違点に注目すれば良いのかははっきりしない。本問題は規模が小さいので、このように相違点を列挙することができたが、実際の大規模な問題では組合せ数が指数関数で増大するため、このような考察も困難である。以上から、2.4.3 項で課題として指摘した点の本ベンチマークでの妥当性を確認することができた。

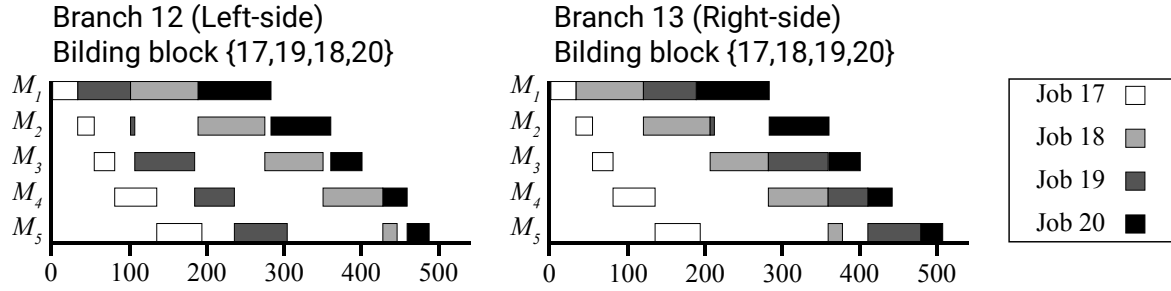


Fig. 4.17: Gantt chart of left-side cluster feature and right-side one at Branch 10 of decision bifurcation diagram based on Ward's method with schema model B.

次に、本研究で提案する意思決定分岐図に基づく解空間構造の分析によって、これら2つの代替案の解釈が可能になったことを確認する。代替案10と11の3つの相違点の優先順位は、意思決定分岐図の階層関係から整理することができる。例えば、図4.15に示される意思決定分岐図では、2つの代替案はBranch 8で2つに分けられている。ここでの左右の違いとして整理されている解空間構造の違いは、ジョブ14,15,16の関係である。具体的には、左に分岐する場合はジョブ15の直後にジョブ16を処理する一方、右に分岐する場合はジョブ15,14,16の順で処理する。これらの構造の違いは相違点Bに対応し、一番優先度が高いと結論づけることができる。さらに、次の分岐であるBranch 10では相違点C、さらに次の分岐であるBranch 13では相違点Aと、構造の違いが順に整理され、代替案11に至ることができる。あくまでもこれは解釈の1つではあるものの、着眼点を明確にして意思決定を円滑に進めるといふ点では効果があると考えられる。

提案した意思決定分岐図はビルディングブロック仮説に基づき、優れた代替案は重要な部分列の集積によって構成されていることを前提にしている。提案する分析法によって明らかになったビルディングブロックの効果を確認し、仮説を検証する。ここでは、図4.16で示される意思決定分岐図のBranch 10に焦点を当てる。Branch 10では、代替案12,14,15を左クラスタに、代替案16,17右クラスタに分割する分岐点である。それぞれの目的関数空間での範囲は、図4.7(a)の散布図から確認できるように範囲に重複がなく、ここでの分岐はシナリオ1としてラベル付けしている。具体的には、左クラスタは右クラスタに比べてMakespanの値は優れている一方でTardinessの値は悪化しているトレードオフの関係がある。意思決定分岐図のボックスからここでの分岐点でのレバレッジポイントはジョブ18とジョブ19の関係であるので、それぞれのビルディングブロックに対して目的関数への影響を調査する。Branch 10時点で17番目はジョブ17、20番目はジョブ20であるので、それぞれの部分的なジョブ列{17,19,18,20}と{17,18,19,20}のガントチャートを図4.17に示す。ガントチャートから、部分列においてもMakespanの値が左クラスタの方が小さいことを確認できる。また、本ベンチマークではジョブの納期がジョブの昇順に設定されているため、{17,19,18,20}はTardinessの値が悪化している。したがって、解空間の構造分析法によって抜き出したビルディングブロックが目的関数に寄与することを確認することができた。

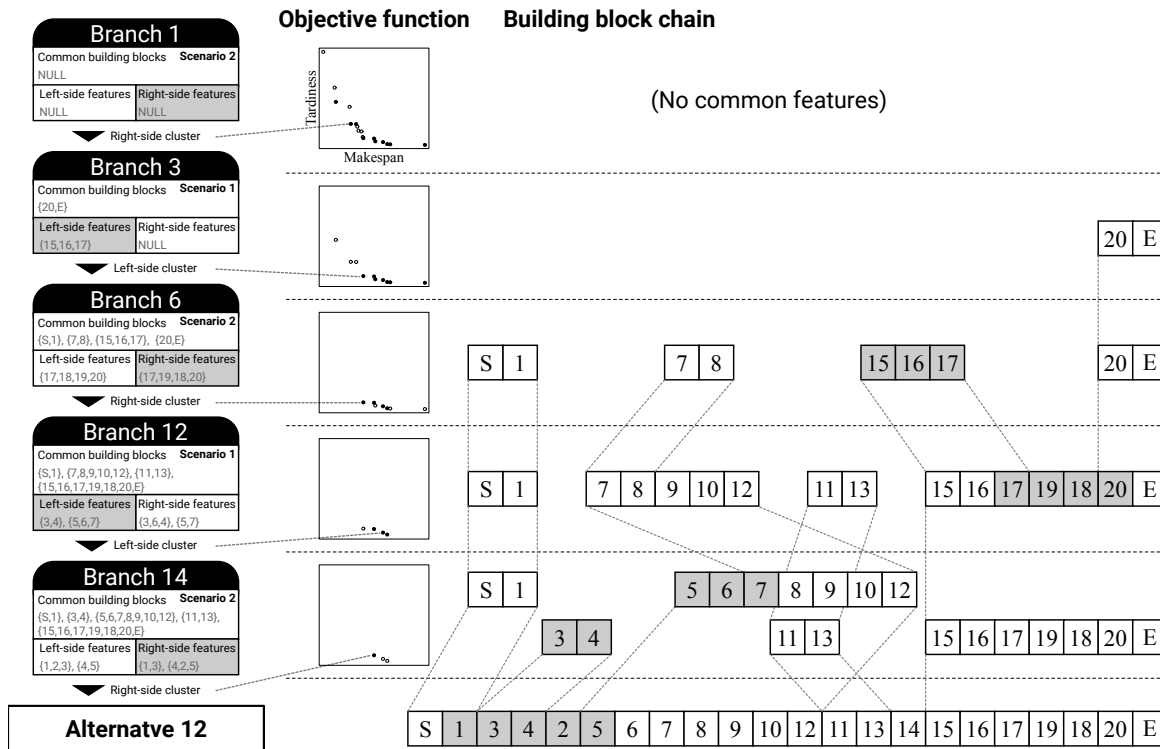


Fig. 4.18: Connection of building block's chain in Alternative 12 based on Ward's method with schema model A.

最後に、ビルディングブロックの集積で代替案に至ることができることを確認し、探索木を用いた代替案の選定プロセスを検証する。複数の結果を比較して、スキーマモデルやリンケージ手法による選定プロセスへの影響を考察する。ここでは、代替案 12 に焦点を当てて、図 4.14、4.15、4.16 に示される 3 つの意思決定分岐図それぞれでの結果を比較する。意思決定分岐図をトップダウンに読み解く過程をそれぞれ、図 4.18、4.19、4.20 に示す。図の左側には意思決定分岐図の各分岐点にあたる情報を示しており、代替案 12 に至る過程で選択するクラスタをハイライトしている。それぞれの分岐点で 2 つに分けられるクラスタを目的関数空間上に図示したものを中央に記載している。黒の点が代替案 12 を含むクラスタ、白の点がもう一方のクラスタに所属する代替案である。図の右側では共通ビルディングブロックとして記載されている部分を記載している。それぞれの上位の分岐で Category B に分類されていたブロックと Category C に分類されていたブロックを区別し、特に Category B に分類されていたブロックをハイライトしている。図 4.18、4.19 から、スキーマモデル A の場合は連続する 2 つのジョブの順列関係がビルディングブロックの候補であるため、ジョブを連結することで代替案 12 に至ることを確認できる。一方で、図 4.20 のスキーマモデル B の場合はジョブの位置関係がビルディングブロックの候補であるので、* を用いたスキーマモデルを用いて表現し、スキーマの固定ビットが増えていることを確認することができる。

まず、図 4.18 を用いて、代替案 12 に至る経緯を解釈できることを確認する。Branch 1 は左右

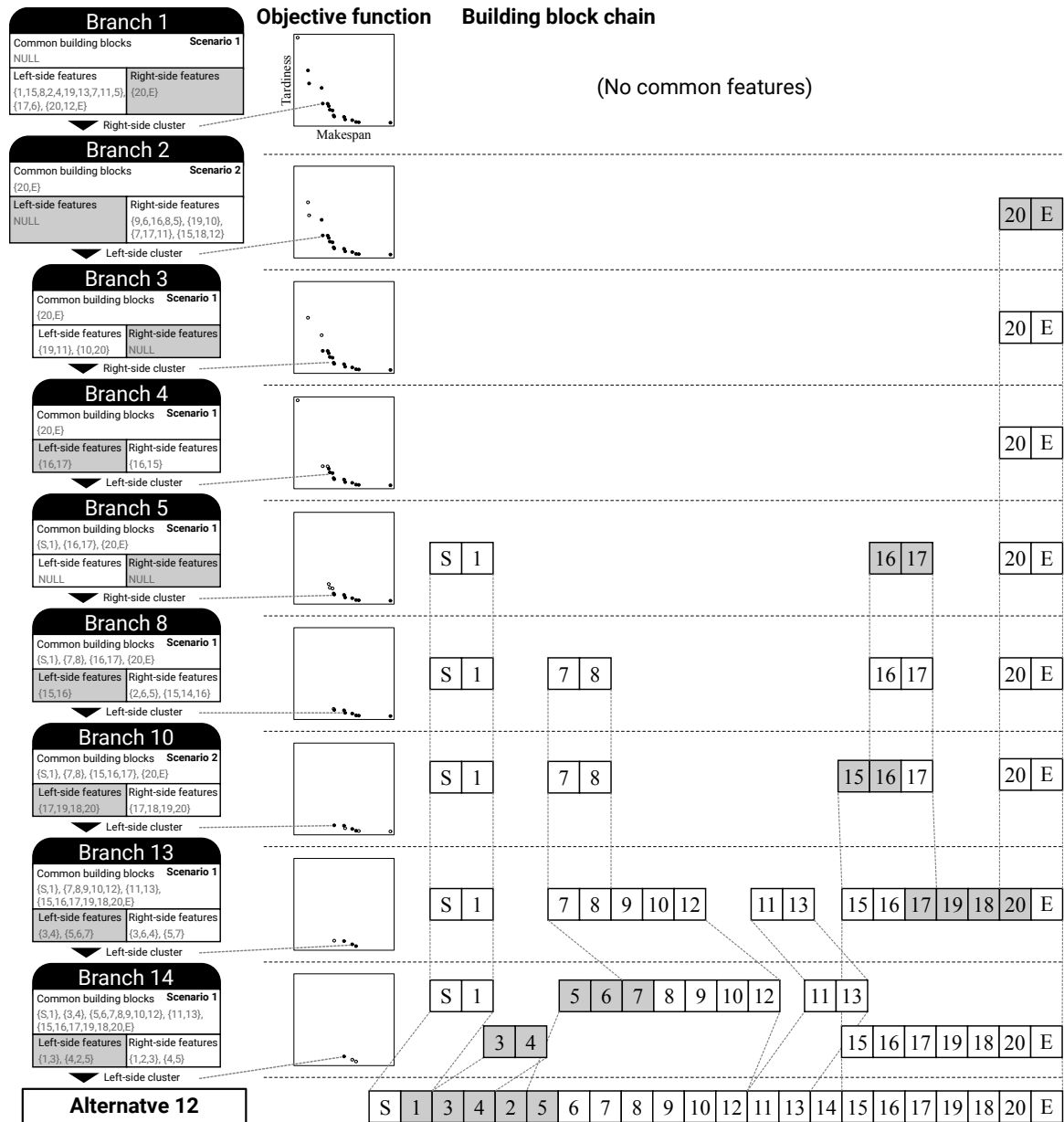


Fig. 4.19: Connection of building block's chain in Alternative 12 based on group average linkage with schema model A.

のクラスタを合わせて全ての代替案の集合を考慮している。ここでの共通するビルディングブロックがないことから、目的関数の配分によってジョブの順列が変化していることがわかる。Branch 1で左のクラスタを選択する場合は Branch 2、右のクラスタを選択する場合は Branch 3 へと検討を進めることになるが、代替案 12 は右クラスタに含まれるため、ここでは順に Branch 3 を検討することにする。Branch 3 では共通ビルディングブロックとして最後にジョブ 20 を処理することが明らかになっている。Branch 3 はシナリオ 1 の分岐点であるので、左右のクラスタにはトレードオフの関係がある。Makespan を優先する場合は右クラスタ、Tardiness を優先する場合は左クラス

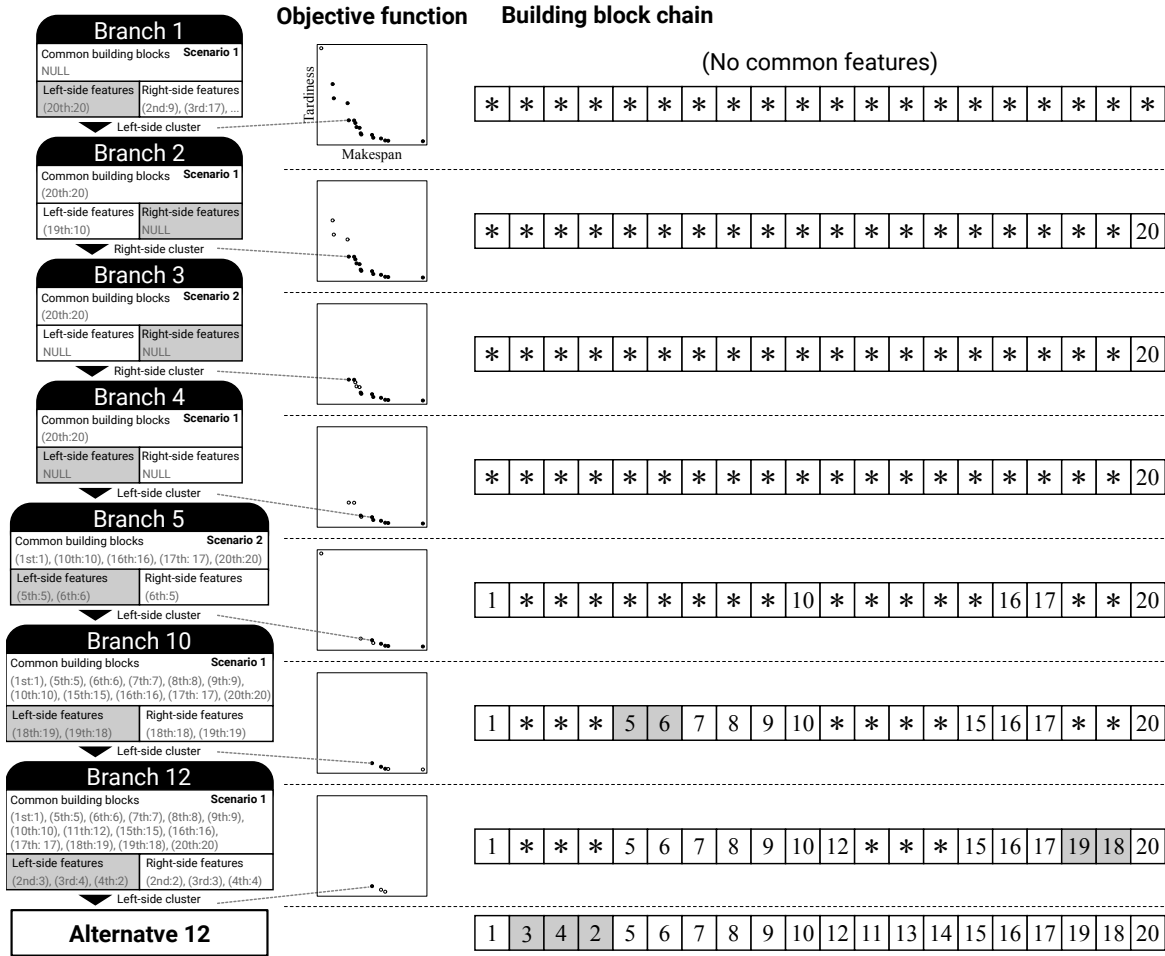


Fig. 4.20: Connection of building block's chain in Alternative 12 based on Ward's method with schema model B.

タを選択することになる。このような過程で Branch 6 に至れば、この時点で 4 つのビルディングブロック ($\{S,1\}$, $\{7,8\}$, $\{15,16,17\}$, $\{20,E\}$) が存在している。さらに左クラスターの Branch 12 へと検討を進めると、 $\{7,8\}$ のビルディングブロックが $\{7,8,9,10,12\}$ へと拡張されたり、 $\{15,16,17\}$ と $\{20,E\}$ にジョブ 19 と 18 を挿入した新たな部分列 $\{15,16,17,19,18,20,E\}$ が生成されたりと、より解が具体的になる。このように、目的関数空間と解の構造とを相互に参照しながら意思決定分岐図をトップダウンに辿ることで代替案 12 へと至ることができることを確認できた。

次に、リンケージ手法による結果の変化を考察する。図 4.18 はスキーマモデル A でウォード法を用いた場合、図 4.19 はスキーマモデル A で郡平均法を用いた場合である。リンケージ手法の変更に伴ってデンドログラムの形状が異なるため、意思決定分岐図も異なっている。ウォード法はデータのばらつきを抑えられる手法であることから、少ない分岐数で代替案 12 に至ることができている。一方で郡平均法では、図 4.8 からわかるように、主流のクラスターから個体数が少ないクラスターを除くような分岐構造をとっており、多数の分岐の末、代替案 12 に至ることになる。分

岐の数は異なるものの、その中身であるビルディングブロックの優先度は類似している。具体的には、ワード法の Branch 6 と郡平均法の Branch 10、ワード法の Branch 12 と郡平均法の Branch 13、ワード法の Branch 14 と郡平均法の Branch 14 で共通ビルディングブロックは全く同じである。つまり、スキーマモデルが同じである以上、代替案のビルディングブロックの集積過程は本質的に変わらないと考えられる。ワード法の結果では 3 つ目の分岐で 3 つの部分列 ($\{S,1\}$ 、 $\{7,8\}$ 、 $\{15,16,17\}$) が一度に共通ビルディングブロックとして形成されているが、郡平均法では、5 つ目の分岐から 7 つ目の分岐にかけて $\{S,1\}$ 、 $\{7,8\}$ 、 $\{15,16,17\}$ と順に形成されていることから、郡平均法で代替案 12 に至る分岐の数が増えることでその集積過程をより詳細に表現できていると考えることができる。したがって、スキーマモデルが同じであれば代替案に至るビルディングブロックの集積は変わらないものの、リンケージ手法を変えることで、他の代替案との関係を議論する際に異なる解釈を行うことができると考えられる。

最後に、スキーマモデル選択での結果の変化を考察する。図 4.20 はワード法を用いた分類結果で、スキーマモデル B で代替案 12 に至る経緯を表している。図 4.18 に示されるスキーマモデル A を用いた結果と比較する。スキーマモデル A ではビルディングブロックの連鎖を拡大する形で解釈を行ったが、スキーマモデル B ではジョブの位置を抽象している関係から Tardiness との親和性が高く、目的関数の影響と関連してビルディングブロックの集積を考察することができる。Branch 10 までの分岐ではそれぞれ Tardiness の値が向上するように分岐を辿っている。それに呼応してビルディングブロックの固定部分は Tardiness が向上するようにジョブの番号が納期順と一致するように構成されている。Branch 10 では Tardiness ではなく Makespan を向上させるように左クラスタを選択しているが、この時点で初めて Tardiness が悪化するジョブの降順のビルディングブロックが形成されていることがわかる。また、スキーマモデル A と比較することで、スキーマモデル選択による意思決定の着眼点の違いを確認することができる。スキーマモデル A の場合は上述の通り、3 つ目の分岐で 3 つの部分列 ($\{S,1\}$ 、 $\{7,8\}$ 、 $\{15,16,17\}$) が共通する特徴として抽象されている一方で、スキーマモデル B では Branch 5 で、1 番目にジョブ 1、10 番目にジョブ 10、16 番目にジョブ 16、17 番目にジョブ 17、の 4 つの部分列を特徴として取り上げている。ジョブ 10 に関する考察はスキーマモデル A では考慮されていないスキーマモデル B 特有の着眼点である。このように、スキーマモデルが異なれば、代替案の重要な部分列の意味も異なり、その集積過程の解釈の形も異なると考えられる。

4.7 結言

本章では、フローショップスケジューリング問題をベンチマークとして例に引き、第 3 章で構築した組合せを伴う解空間の構造分析法の有効性を確認した。まず、フローショップスケジューリング問題の解空間構造を順列構造として定義し、構造を表現するスキーマモデルを 2 つ提案した。そ

れぞれのスキーマモデルで抽象するビルディングブロックの候補を明らかにし、モデルの特徴を整理した。次に、モデルの特徴を踏まえて代替案の類似性尺度の設定を具体例を挙げながら論じた。フローショップスケジューリング問題では、評価関数そのものが意思決定を行う上での重要な視点であることを受け、目的関数空間の可視化に基づくクラスタの評価方法を提示し、2つのシナリオを定義した。最後に規模の異なる順列フローショップスケジューリング問題を2つ取り上げ、提案する構造分析法を適用した。階層的クラスタリングの設定パラメータである類似性尺度とリンケージ手法の選択による結果の変化について調査と検討を行った。提案手法の効果を検証するため、3つの結果を取り上げて構造分析を行い、意思決定分岐図を作成した。組合せ型設計問題の課題点を再確認した上で、分析条件の違いによる分析結果やその解釈性の違いを整理し、提案手法の有効性を確認した。

第 5 章

グローバルプロダクトファミリ設計への適用

5.1 緒言

本章では、異なる種別の組合せ型設計問題としてグローバルプロダクトファミリ設計を取り上げ、組合せを伴う解空間の分析法を応用する。グローバルプロダクトファミリ設計では、モジュール組合せによる製品系列展開とサプライチェーン設計を連携させた組合せ最適化問題が定式化されるため、その解空間構造は選択構造を有している。本章ではグローバルプロダクトファミリ設計での選択構造問題に合わせた解空間構造表現や構造分析の考え方を論じ、多数評価指標でのビルディングブロックの解釈方法を構築する。まず、グローバルプロダクトファミリ設計のための数理モデルと最適化手法を整理し、その特性を踏まえたスキーマモデルを提案する。ロジスティック回帰分析に基づく分岐理由考察の枠組みを提案し、自動車のグローバルプロダクトファミリ設計を例として引いて、提案する方法の妥当性と効果を検証する。

5.2 グローバルプロダクトファミリの最適設計

5.2.1 グローバルプロダクトファミリ

今日の製造業では、設計開発期間の短縮やコストの低減を図りながらも、多様な消費者ニーズに柔軟に対応することが求められている。そこで、背景となるコンセプトを共有する一連の製品群を製品系列 (プロダクトファミリ) として位置づけ、それらを統合的に考えて設計することが有望であると考えられている。このような設計方法論は、製品系列設計と呼ばれる^[81]。製品を独立したモジュールに分け、系列内の異なる製品間で一定の部分を共通化することで間接費を削減したり、モジュール組合せにより多様な製品を柔軟に提供したりすることで、コストの低減と設計の効率化を狙うことができるため、製品系列設計はマスカスタマイゼーション^[82]の要求に答えながらも製品

の競争力の向上に寄与する、重要な概念の1つであると考えられている^[83]。

従来の製品系列設計の研究では、製品系列の展開の内容に関するものが主であった。つまり、どのモジュールや属性を製品間で共通化するか、そして、共通化することで製品系列にコストや性能の面でどれほどの影響を与えるのかを論じてきた^[84, 85]。しかし近年では、製品系列展開の内容が、取り巻く他の要素との相互作用において、どのような様相を呈するのかという方向へと研究の主眼が広がってきている。その代表例が、適切な製品供給網を構築するサプライチェーンネットワークデザインを考慮した製品系列設計である^[86, 87, 88, 89, 90]。近年のグローバル化の進展に伴い、設計と製造が世界規模で行われるようになり、市場だけでなく、生産に必要な設備や拠点も世界中に広がりつつある^[91]。従来は、製品系列の展開のみに着目した研究^[84, 85]が多く行われていたが、このような背景のもと、多様な顧客要求に対応するための製品系列の設計だけでなく、さまざまな地域に分布している市場や生産拠点を適切に選択し、連携させるためのサプライチェーン設計を並行して行う必要があると考えられている。製品系列とサプライチェーンの同時設計は、製品の設計開発における新しい視点であり、より大局的な最適性をもたらす枠組みとして、グローバルプロダクトファミリー設計と呼ばれている^[87]。

グローバルプロダクトファミリーの設計では、さまざまな要因を含みながらもそれらの相互作用が作用して複雑になる上に、戦略的な意思決定が曖昧である難しさを抱えていることが指摘されており、数値的な支援だけでは最適な設計を見出すことは困難であると考えられている^[87]。このような状況に対して、本研究で提案する解空間の構造分析による意思決定支援の枠組みは有望であると考えられる。先行研究^[92]では、多目的遺伝的アルゴリズムを用いて大量のパレート最適解を生成した後、それらを小さなクラスタに分類し、その概念的な意味を特定する意思決定支援法を提案している。

5.2.2 最適設計問題の構築

グローバルプロダクトファミリー設計は、製品系列とサプライチェーンの相互作用を加味した、統合的設計の枠組みである。製品系列やサプライチェーンネットワークデザインのそれぞれは別々に最適化を用いた設計法が提案されているため、それらを連携させることによって大規模な数理計画問題として定式化する。本項では、Fujita ら^[88]が提案している、製品系列とサプライチェーンの同時設計問題を念頭に置き、本章で取り組む最適設計問題のモデル化と定式化について述べる。

製品系列設計の代表として、モジュール組合せによる設計を前提とする。何らかのアーキテクチャと形態を想定することができ、加えて各モジュールとして利用可能なものが事前に確定している場合、製品系列設計問題は製品を構成するために用いるモジュール群を、その候補群の中から取捨選択する問題になる。各製品はモジュールを実装するスロットを備えており、そのスロットにモジュール候補群の中からモジュールを選択することで製品が構成されるため、形式的には 0-1 整数計画問題になっており、組合せ最適化を用いることで最適な設計を定めることができる^[84]。

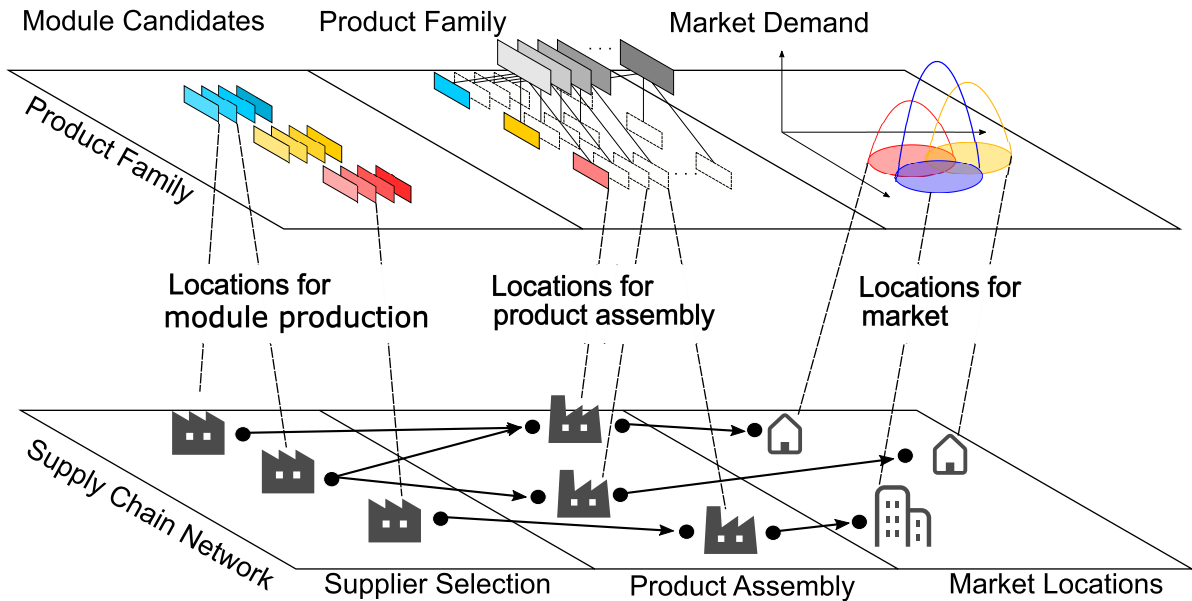


Fig. 5.1: Global product family model: product family layer and supply chain network layer.

先行研究で用いられている同時設計では、モジュール組合せ設計とそのプライチェーンネットワークとの対応関係に図 5.1 のような 2 層のモデルを想定し、最適設計問題を構築している。

上層は製品系列設計におけるモジュール組合せ設計の構成を表している。先述の通り、製品系列を構成する製品はモジュールを実装するスロットを備えており、あらかじめ想定したモジュール候補群の中から実装するモジュールを選択することで製品ラインアップが展開される。また、各製品はグローバルに広がるさまざまな市場を想定しており、それぞれの需要の分布からモジュール構成を決定する。このような生産形態を流れとして改めて整理すると、モジュール生産 (購入)、製品組み立て、製品販売の 3 つのプロセスとして構成されている。

下層はサプライチェーンネットワークの構成を表している。モジュール組合せ設計の流れは 3 つのプロセスで構成されるため、3 エシェロンのサプライチェーンネットワークを想定する。サプライチェーン設計では、モジュール生産、製品組み立て、製品販売のそれぞれを行うための設備投資を計画することになる。さらに、それぞれの施設での購入量、生産量、販売量だけでなく、それぞれが地理的に離れている場合は、各施設間の輸送量を決定しなければならない。数理的な形式で設計問題を定式化する場合は、設備投資などの選択の有無は 0-1 整数を用いて表現され、生産量や輸送量は連続変数を用いて表現されることが一般的であり、多くの場合は混合整数計画問題として取り扱われる。

サプライチェーン要素で取り扱うことができるモジュールや製品を固定し、上層の製品系列の内容と 1 対 1 に対応させることで上層と下層を同時に設計することができる。例えば、下層のサプライチェーンネットワーク構成として、あるモジュールを生産する設備を導入することは、上層のモジュール組合せ設計では、モジュール候補群から実装するモジュールを選択することに対応する。

製品生産施設の選択と製品系列展開のラインナップ、市場選択と市場への参入の選択についても、それぞれが同様に対応するので、サプライチェーンネットワークの設計を通して、製品間の共通化を見越しながら、製品系列内の製品の構成、その実現と展開を同時に考えることができる。

一般的なサプライチェーンネットワーク問題と比較して、本問題は次のような特徴を持つ。

- 同時設計モデルでは、考えうるモジュールの候補やその組合せで構成される製品の候補の全てを考慮しておかなければならないため、それらと 1 対 1 に対応するサプライチェーン要素数が組合せ的に増大する。最適化問題での設備導入の有無を表す設計変数の量が多量になるため、問題の非線形性が増大し、探索や結果の解釈が困難になる。
- 製品系列の構成はサプライチェーンでの設備選択と物流量から一意に決定される。サプライチェーンでの評価指標はコストや品質、デリバリータイムが用いられるため、共通化の程度など、サプライチェーンから独立した製品系列の性能評価指標を反映させる場合には、独自の評価が必要になる。

5.2.3 定式化

定式化の条件設定

グローバルプロダクトファミリーは広範な概念であるため、定式化に先立って内容を整理しておく必要がある。そこで、ここでは製品系列とサプライチェーンの同時設計を念頭において、考慮すべき内容について整理する。まず、グローバルプロダクトファミリーの性能評価には、一般に次のような指標が用いられる。

コスト: サプライチェーンにおける総コスト。もしくはそのサプライチェーンで生産する製品の設計開発コスト。

品質: 求められる製品の品質 (機能)。あるいは製造段階における不良率や歩留まりなどに対する信頼性。

リードタイム: 生産、流通、開発などの段階において、工程に着手してから全ての工程が完成するまでの所要期間。

リスク: 何らかの事故や災害などの不確実性による影響。システムとしての頑強性やレジリエンスを直接評価することもあれば、不確実性を想定した生産工場の分散やモジュールの代替性でリスクの評価を代替することもある。

市場への対応範囲: ある市場において求められる製品の種類、例えば、能力や価格、その環境に適した設計であるかなど、どれほど市場に適應できるか。

本研究では、簡単のため QCD と呼ばれる、上記で述べたコスト、品質、リードタイム (納期) の 3 つの評価指標用を目的関数として用いることにする。D は Delivery Time という意味での D だが、

ここではリードタイムとして用いる。グローバルプロダクトファミリの最適設計問題を定式化するに先立ち、本研究での前提と仮定を述べておく。

- サプライチェーンの諸要素

モジュールの生産、製品の組み立て、市場での販売の3エシェロンのサプライチェーンを前提とする。それぞれを実現するための施設や設備を配置する地点をいくつか想定する。各地点間におけるモジュールの輸送、製品の輸送を考慮する。倉庫における在庫は考えないものとする。

- 製品系列の諸要素

1つの製品系列を取り扱う。ただし、系列には複数の製品を含んでいる。アーキテクチャにはモジュラーアーキテクチャを想定する。製品を構成するモジュールはあらかじめ想定されており、その中での組合せを考える。

- 市場の多様性

市場の多様性として、各製品の需要のパターンが異なる市場を複数地点で考慮する。他社の動向やサプライチェーンとの相互関係は扱わないこととし、他社のシェアとの関係性は需要量という形で反映される。また、価格はどの市場でも固定の値として、変動は考慮しない。需要量は既知であり、各市場で固有に決定される。

- 生産における資源

企業における有限の資源は、モジュールの生産量・製品の生産量の制約で代替する。また、時間的な要素は考えないことにする。

- 意思決定モデル

サプライチェーン要素を意思決定要素として取り扱い、製品系列は副次的に決定される。各地点において、対象となるモジュールを生産するかどうか、製品を生産・販売するかどうかを決定する。また、それらの生産量や輸送量を決定する。各施設の選択はバイナリ変数を用いて定式化する。生産量や輸送量は、本来は0以上の整数であるが、大きな値をとるので、ここでは非負の連続変数として定式化する。用意されたモジュールの中での組合せを考えるため、考えうる全てのモジュールと製品に対応した設備に関する設計変数を網羅的に用意する。

- コストモデル

費用は、モジュールの購入における固定費と変動費、製品の生産における固定費と変動費、それぞれの輸送における変動費、販売における固定費を考慮する。固定費は、施設を開設するための施設費と、実際に施設を運用する運用費を考える。施設を開設しながらも生産を行わない場合は、施設費のみが固定費として発生する。

- 品質モデル

製品やモジュールの加工精度や耐久性などによるものを「品質」として呼ぶことにする。例

例えば、日本で生産すれば加工精度、耐久性という意味で良質なものであり、逆に新興国などでは良質なものが生産しにくい、というように、その地域での品質の差を考慮する。生産できる製品やモジュールの品質は、生産現場で働く人員の生産技術能力や電力や設備、またそこで手に入る材料の可能性など、その環境の依存するものと考えられる。本研究では輸送に関する品質劣化は考慮せず、モジュールの品質から最悪ケースで製品の品質を評価する。

- リードタイムモデル

モジュール生産時間と製品の組み立て時間、それらの輸送時間の総和を「リードタイム」として考える。モジュールと製品は、ある一定量の生産を行う場合の時間を考えることとし、ある輸送手段が一度に輸送できる量を基準にする。

以降の項では、グローバルプロダクトファミリモデルとその評価を行うコスト、品質、リードタイムに関する数理モデルについて説明する。なお、本項で用いる変数の一覧は表 5.1, 5.2 にまとめられる。

モジュール生産

モジュールスロット S_j の k 番目のモジュール候補 $m_{j,k}$ をサイト s ($s \in \text{Module site}$) で生産するか意思決定を次の 0-1 変数によって表す。これは製品系列設計の側から見れば、どのモジュール候補を用いるかのモジュール選択に対応する意思決定である。

$$\delta_{j,k}^s = \begin{cases} 1 & \dots & \text{モジュール } m_{j,k} \text{ をサイト } s \text{ で生産する} \\ 0 & \dots & \text{モジュール } m_{j,k} \text{ をサイト } s \text{ で生産しない} \end{cases} \quad (5.1)$$

また、サイト s でモジュール $m_{j,k}$ を生産できるかどうかについては係数 $\alpha_{j,k}^s$ を用いて以下のように制約される。ここでの生産可能性 $\alpha_{j,k}^s$ は、地域での資源の獲得可能性や技術的な能力を代表している。

$$\alpha_{j,k}^s = \begin{cases} 1 & \dots & \text{モジュール } m_{j,k} \text{ はサイト } s \text{ で生産可能} \\ 0 & \dots & \text{モジュール } m_{j,k} \text{ はサイト } s \text{ で生産不可能} \end{cases} \quad (5.2)$$

$$(1 - \alpha_{j,k}^s) \delta_{j,k}^s = 0 \quad (5.3)$$

モジュール輸送

モジュール $m_{j,k}$ のサイト s ($s \in \text{Module site}$) からサイト t ($t \in \text{Product site}$) への輸送量を $x_{j,k}^{s \rightarrow t}$ とする。連続性を担保するため、サイト s でのモジュール $m_{j,k}$ の生産量 $x_{j,k}^{s \rightarrow}$ は以下の制約を満たさなければならない。

$$x_{j,k}^{s \rightarrow} = \sum_t x_{j,k}^{s \rightarrow t} \quad (5.4)$$

Tab. 5.1: Nomenclature of design variables and parameters for cost model.

P_i	: Product ($i = 1, 2, \dots, I$)
S_j	: Module Slot ($j = 1, 2, \dots, J$)
$m_{j,k}$	: Module candidates ($k = 1, 2, \dots, K_j$)
M^l	: Market ($l = 1, 2, \dots, L$)
s, t, r	: Site candidates ($s \in \text{Module site}, t \in \text{Product site}, r \in \text{Market}$)
$\delta_{j,k}^s$	: 0-1 variable representing whether the module is produced or not
δ_i^t	: 0-1 variable representing whether the product is produced or not
$\delta_i^{t \rightarrow r}$	: 0-1 variable representing whether the product is sold or not
$x_{j,k}^{s \rightarrow}$	: Module production volume ($s \in \text{Module site}$) (Unit)
$x_{j,k}^{s \rightarrow t}$	: Module transportation volume (Unit)
$x_{j,k}^{\rightarrow t}$	: Module supply volume ($t \in \text{Product site}$) (Unit)
$x_{j,k \rightarrow i}^t$	: Module assignment volume ($t \in \text{Product site}$) (Unit)
$x_i^{t \rightarrow}$	: Product production volume (Unit)
$x_i^{t \rightarrow r}$	: Product transportation volume (Unit)
$x_i^{\rightarrow r}$	: Product supply volume (sales volume) (Unit)
$\alpha_{j,k}^s$	: Capability of module production
β_i^t	: Capability of product production
$\gamma_{j,k \rightarrow i}$	: Availability of module
D_i^r	: Product demand volume (Unit)
p_i^r	: Product price (Yen)
$w_{j,k}^s$	: Coefficient of constraint on module production volume
W^s	: Upper bound of module and product production volume
$w_{j,k}^{s \rightarrow t}$	: Coefficient on module transportation volume
$W_M^{s \rightarrow t}$	: Upper bound of module transportation volume
$W_P^{t \rightarrow r}$	: Upper bound of product transportation volume
w_i^t	: Coefficient of constraint on product production volume
$w_i^{t \rightarrow r}$	: Coefficient on product transportation volume

サイト s からサイト t への総モジュール輸送量は以下の式で制約される。

$$\sum_j \sum_k w_{j,k}^{s \rightarrow t} x_{j,k}^{s \rightarrow t} \leq W_M^{s \rightarrow t} \quad (5.5)$$

式 (5.5) は、総重量や総容積などの輸送容量の上限を表している。 $w_{j,k}^{s \rightarrow t}$ はその係数であり、モジュールの種類や地域によってさまざまである。 $W_M^{s \rightarrow t}$ はその上限であり、経路ごとに定められている。

製品組み立て

製品 P_i をサイト t ($t \in \text{Product site}$) で生産する (モジュールから組み立てる) かどうかの意思決定を次の 0-1 変数によって表す。これは製品系列設計の側から見れば、どの製品を組み上げるか

Tab. 5.2: Nomenclature of design parameters for quality and leadtime model.

C_{MP}	: Fixed cost of module production (Yen)
$C_{MP_{j,k}}$	: Facility cost for module production (Yen)
c_{MP}	: Variable cost of module production (Yen)
$c_{MP_{j,k}}$	: Unit cost of Module production (Yen/Unit)
c_{MT}	: Variable cost of module transportation (Yen)
$c_{MT_{j,k}}$	: Unit cost of module transportation (Yen/Unit)
$c_{MT}^{s \rightarrow t}$	: Constant for substituting distance in module transportation
C_{PA}	: Fixed cost of product production (Yen)
C_{PA_i}	: Facility cost for Product production (Yen)
C_S	: Facility cost of module and product production (Yen)
C_S^s	: Facility cost of module and product production per each factory (Yen)
c_{PA}	: Variable cost of product production (Yen)
c_{PA_i}	: Unit cost of product production (Yen/Unit)
c_{PT}	: Product transportation variable cost (Yen)
c_{PT_i}	: Unit cost of product transportation (Yen/Unit)
$c_{PT}^{t \rightarrow r}$	: Constant for substituting distance in product transportation
C_{PS}	: Product sales fixed cost (Yen)
C_{PS}^r	: Coefficient on fixed cost for sales per site (Yen)
C_{PS_i}	: Coefficient on fixed cost for sales of each product per site (Yen)
$q_{j,k,q}^s$	: Quality measure on module production at each site
q_i^t	: Quality measure on product production at each site
$l_{j,k}^s$	: Leadtime of module production at each site (hours)
l_i^t	: Leadtime of product production at each site (hours)
$l_m^{s \rightarrow t}$	: Leadtime of module transportation between a pair of sites (hours)
$l_P^{t \rightarrow r}$	: Leadtime of product transportation between a pair of sites (hours)
k_{MP}^s	: Adjustment factor of module production facility cost by each site
k_{mp}^s	: Adjustment factor of module production unit cost by each site
k_{mt}^s	: Adjustment factor of module transportation unit cost by each site
k_{pa}^t	: Adjustment factor of product production facility cost by each site
k_{PA}^t	: Adjustment factor of product production unit cost by each site
k_{pt}^t	: Adjustment factor of product transportation unit cost by each site
k_{PS}^r	: Adjustment factor of product sales fixed cost by each site

の選択に対応する意思決定である。

$$\delta_i^t = \begin{cases} 1 & \cdots & \text{製品 } P_i \text{ をサイト } t \text{ で生産する} \\ 0 & \cdots & \text{製品 } P_i \text{ をサイト } t \text{ で生産しない} \end{cases} \quad (5.6)$$

また、製品 P_i がサイト t で生産できるかどうかについては、係数 β_i^t を用いて、次のように制約さ

れる。ここでの組み立て可能性 β_i^t は、主に技術的な能力を代替する。

$$\beta_i^t = \begin{cases} 1 & \dots & \text{製品 } P_i \text{ はサイト } t \text{ で生産可能} \\ 0 & \dots & \text{製品 } P_i \text{ はサイト } t \text{ で生産不可能} \end{cases} \quad (5.7)$$

$$(1 - \beta_i^t) \delta_i^t = 0 \quad (5.8)$$

供給量の整合をとるため、サイト t でのモジュール $m_{j,k}$ の供給量 $x_{j,k}^{\rightarrow t}$ はサイト s ($s \in \text{Module site}$) からのモジュールの輸送量 $x_{j,k}^{s \rightarrow t}$ を用いて、次のように制約される。

$$x_{j,k}^{\rightarrow t} = \sum_s x_{j,k}^{s \rightarrow t} \quad (5.9)$$

モジュール組合せによる製品系列設計では、製品はモジュールスロットを備えており、モジュールを組み付けることで製品が構成される。モジュールごとに異なる性能を有しているとすれば、ある製品 P_i に求められる性能水準を満たすモジュール $m_{j,k}$ の種類も限定されている。製品 P_i へのモジュール $m_{j,k}$ の組み付け可能性を係数 $\gamma_{j,k \rightarrow i}$ を用いて設定する。

$$\gamma_{j,k \rightarrow i} = \begin{cases} 1 & \dots & \text{モジュール } m_{j,k} \text{ は製品 } P_i \text{ に組み付けることができる} \\ 0 & \dots & \text{モジュール } m_{j,k} \text{ は製品 } P_i \text{ に組み付けることができない} \end{cases} \quad (5.10)$$

モジュール $m_{j,k}$ の製品 P_i への割当量を $x_{j,k \rightarrow i}^t$ とすると、モジュール $m_{j,k}$ の供給量 $x_{j,k}^{\rightarrow t}$ に上限が制約される。

$$0 \leq x_{j,k \rightarrow i}^t \leq x_{j,k}^{\rightarrow t} \gamma_{j,k \rightarrow i} \beta_i^t \delta_i^t \quad (5.11)$$

また、モジュール割当量は供給量と整合しているため、次の式を満たす。

$$\sum_i x_{j,k \rightarrow i}^t = x_{j,k}^{\rightarrow t} \quad (5.12)$$

以上のもと、各製品 P_i の生産量 $x_i^{t \rightarrow}$ はすべてのモジュールスロット S_j に関して量的に整合していなければならないので、以下の関係式を満たす。

$$x_i^{t \rightarrow} = \sum_k x_{j,k \rightarrow i}^t \quad (5.13)$$

さらに、サイト s での総生産量について、地域での生産能力を限定する場合は、モジュールの生産量 $x_{j,k}^{s \rightarrow}$ と製品の生産量 $x_i^{s \rightarrow}$ の総和は以下の制約条件式を満たさなければならない。

$$\sum_j \sum_k w_{j,k}^s x_{j,k}^{s \rightarrow} + \sum_i w_i^s x_i^{s \rightarrow} \leq W^s \quad (5.14)$$

ここで、 $w_{j,k}^s$ と w_i^s はそれぞれのモジュールと製品の違いを代替する係数である。ここで、 $w_{j,k}^s$ と w_i^s はそれぞれの地域における、モジュールや製品の種類に応じた、生産に必要なリソースの違いを代替する係数である。例えば、使用可能な材料や労働力の量が抽象され、その上限を W^s で与える。

製品輸送

製品 P_i のサイト t ($t \in \text{Product site}$) からサイト r ($r \in \text{Market}$) への輸送量を $x_i^{t \rightarrow r}$ とする。その値は製品生産量と整合させるため、以下の式で制約される。

$$x_i^{t \rightarrow} = \sum_r x_i^{t \rightarrow r} \quad (5.15)$$

サイト t からサイト r への総製品輸送量はモジュール輸送と同様に、輸送能力の上限によって制約されている。

$$\sum_i w_i^{t \rightarrow r} x_i^{t \rightarrow r} \leq W_P^{t \rightarrow r} \quad (5.16)$$

ここで、 $w_i^{t \rightarrow r}$ はある経路での製品に固有の重量や容積を表す係数であり、 $W_P^{t \rightarrow r}$ は経路ごとに上限が設定されている。

製品の販売

製品 P_i をサイト r ($r \in \text{Market}$) で販売するかどうかについて次の 0-1 変数により表す。製品系列の側から見れば、市場投入を決定する変数である。

$$\delta_i^r = \begin{cases} 1 & \dots & \text{製品 } P_i \text{ をサイト } r \text{ で販売する} \\ 0 & \dots & \text{製品 } P_i \text{ をサイト } r \text{ で販売しない} \end{cases} \quad (5.17)$$

製品 P_i をサイト r で販売しない場合、その市場への製品 P_i の供給は不要であるため、以下の式が導かれる。

$$(1 - \delta_i^r) x_i^{\rightarrow r} = 0 \quad (5.18)$$

ここで、 $x_i^{\rightarrow r}$ はサイト r への製品 P_i の供給総量を表し、サイト t ($t \in \text{Product site}$) からの製品の輸送量 $x_i^{t \rightarrow r}$ を用いて次式によって連続性が保たれる。

$$x_i^{\rightarrow r} = \sum_t x_i^{t \rightarrow r} \quad (5.19)$$

地域ごとに製品の需要量は定められている。サイト r での製品 P_i についての需要と供給の関係から、次式が成り立つ。

$$x_i^{\rightarrow r} \leq D_i^r \delta_i^r \quad (5.20)$$

ここで、 D_i^r は製品 P_i のサイト r での潜在的な需要量である。

展開される全製品による総売上げ $Sales$ はサイト r での製品 P_i の価格 p_i^r を用いて次式により計算できる。

$$Sales = \sum_i \sum_r p_i^r x_i^{\rightarrow r} \quad (5.21)$$

コストモデル

サプライチェーンの総コストは固定費と変動費の総和として定義される。なお、固定費としては、モジュール生産、製品生産、製品の販売に関する設備費用および、サイト毎の施設費を想定する。また、変動費には、モジュール生産と製品組み立て量に比例する費用と、モジュールと製品の輸送費を想定する。

モジュール生産の固定費 C_{MP} は、各モジュールに対する設備費を想定する。モジュール生産の意思決定を表す変数 $\delta_{j,k}^s$ は設備導入の有無と対応するため、次式により定義される。

$$C_{MP} = \sum_j \sum_k \sum_s k_{MP}^s C_{MPj,k} \delta_{j,k}^s \quad (5.22)$$

ここで、 $C_{MPj,k}$ はモジュールの種類に応じた設備費であり、サイトごとの差異は物価や技術的なコストを加味して k_{MP}^s で調整される。

同様に、製品生産についての固定費 C_{PA} はモジュール生産の意思決定を表す変数 δ_i^t を用いて次式で定義される。

$$C_{PA} = \sum_i \sum_t k_{PA}^t C_{PAi} \delta_i^t \quad (5.23)$$

ここで、 C_{PAi} は製品の種類ごとに設定され、地域差は k_{PA}^t でサイト毎に調整される。

すべてのモジュール生産と製品組み立てのための施設費 C_S は、サイト毎に定義される。 $s = t$ を満たす場合は同じサイトで生産されている。2つの意思決定変数 $\delta_{j,k}^s$ と δ_i^t は、それぞれ施設導入の意思決定に対応するため、次式により与えられる。

$$C_S = \sum_s C_S^s \left(1 - \prod_j \prod_k (1 - \delta_{j,k}^s) \prod_i (1 - \delta_i^t) \right) \quad (5.24)$$

ここで、 C_S^s はサイト毎の施設費である。

製品販売のための固定費 C_{PS} は、サイト毎の施設費と製品毎の設備費の合計として定義される。

$$C_{PS} = \sum_r \left\{ C_{PS}^r \left(1 - \prod_i (1 - \delta_i^r) \right) + \sum_i k_{PS}^r C_{PSi} \delta_i^r \right\} \quad (5.25)$$

ここで、 C_{PS}^r と C_{PSi} はサイトの場所と製品の種類に応じた施設費と設備費である。設備費は地域による差を k_{PS}^r で代替している。

以上から、固定費 *Fixed cost* が次の式で算出される。

$$Fixed\ cost = C_{MP} + C_{PA} + C_S + C_{PS} \quad (5.26)$$

モジュール生産についての変動費 c_{MP} はその生産量に比例するため、次式で定義される。

$$c_{MP} = \sum_j \sum_k \sum_s k_{mp}^s c_{MPj,k} x_{j,k}^{s \rightarrow} \quad (5.27)$$

ここで、 $c_{MPj,k}$ はモジュールの種類に応じた単価である。地域による違いは係数 k_{mp}^s で調整される。

モジュール輸送についての変動費 c_{MT} はその輸送量に比例するため、次式で定義される。

$$c_{MT} = \sum_s \sum_t \sum_j \sum_k k_{mt}^s c_{MT}^{s \rightarrow t} c_{MTj,k} x_{j,k}^{s \rightarrow t} \quad (5.28)$$

ここで、 $c_{MTj,k}$ はモジュールの種類に応じた輸送の単価である。 $c_{MT}^{s \rightarrow t}$ は輸送距離に関する係数である。地域による違いは発送地毎に係数 k_{mt}^s を用いて調整される。

製品組み立てについての変動費 c_{PA} は次のように定義される。

$$c_{PA} = \sum_i \sum_t k_{pa}^t c_{PAi} x_i^{t \rightarrow} \quad (5.29)$$

ここで、 c_{PAi} は製品の種類毎の組み立て単価である。地域による組み立てコストの違いは k_{pa}^t で調整される。

製品輸送の変動費 c_{PT} は次式で定義される。

$$c_{PT} = \sum_t \sum_r \sum_i k_{pt}^t c_{PT}^{t \rightarrow r} c_{PTi} x_i^{t \rightarrow r} \quad (5.30)$$

ここで、 c_{PTi} は製品の輸送単価である。 $c_{PT}^{t \rightarrow r}$ は輸送距離に相当し、地域による違いは発送地毎に係数 k_{pt}^t を用いて調整される。

以上から、変動費 *Variable cost* が次の式で算出される。

$$\text{Variable cost} = c_{MP} + c_{MT} + c_{PA} + c_{PT} \quad (5.31)$$

利益モデル

利益を売上とコストの差として定義される。

$$\text{Profit} = \text{Sales} - \text{Fixed cost} - \text{Variable cost} \quad (5.32)$$

品質モデル

製品の品質とは本来、材料、加工、組み立て、輸送、配送など、製造の全ての工程が作用する。その指標はそれぞれの内容に基づいて決定されるべきであるが、本論文では、先述の条件設定のもと、品質指標 Q_i^t と呼ばれる抽象化された指標を導入する。品質指標の範囲は $[0,1]$ に設定する。

まず、サイト $s (s \in \text{Module site})$ でモジュール $m_{j,k}$ を生産する場合の、モジュールの品質を $q_{j,k}^s$ とする。製品の品質は組み立て過程で割り当てられるモジュールの品質に基づいて設定される。品質を評価するために、サイト s で生産されたモジュールがサイト t で製品 P_i を組み付けられるかどうかの割り当て可能性を次の 0-1 変数 $\epsilon_{j,k \rightarrow i}^{s \rightarrow t}$ で判断する。

$$\epsilon_{j,k \rightarrow i}^{s \rightarrow t} = \begin{cases} 1 & (x_{j,k}^{s \rightarrow t} x_{j,k \rightarrow i}^t > 0) \\ 0 & (x_{j,k}^{s \rightarrow t} x_{j,k \rightarrow i}^t = 0) \end{cases} \quad (5.33)$$

サイト $t(t \in \text{Product site})$ で製品 P_i が生産される場合、そのモジュールスロット S_j に割り当てられるモジュールの品質 $q_{j \rightarrow i}^t$ を割り当て可能性 $\epsilon_{j,k \rightarrow i}^{s \rightarrow t}$ を用いて次のように定義する。

$$q_{j \rightarrow i}^t = \min_{k,s} \{ q_{j,k}^s \epsilon_{j,k \rightarrow i}^{s \rightarrow t} \mid \epsilon_{j,k \rightarrow i}^{s \rightarrow t} = 1 \} \quad (5.34)$$

異なるサイトで生産されたモジュールや、異なる種類のモジュールが割り当てられる場合、それらの中の最小値を用いて評価している。

サイト t で生産される製品 P_i の品質の値 Q_i^t は、その組立 (生産) によって生じる品質の値を q_i^t として、モジュールの品質と組立により生じる品質の値の総積として定義される。

$$Q_i^t = q_i^t \left(\prod_j q_{j \rightarrow i}^t \right) \quad (5.35)$$

生産される全製品の品質 Q_i^t の中での最悪値 Q_{min} は次の式で算出される。

$$Q_{min} = \min_{t,i} \{ Q_i^t \} \quad (5.36)$$

リードタイムモデル

受注から顧客への納品にかかる時間は、コストと品質に次ぐ重要な判断基準である。

サイト $s(s \in \text{Module site})$ において生産されるモジュールの製造リードタイムを $l_{j,k}^s$ 、サイト s からサイト $t(t \in \text{Product site})$ までの輸送リードタイムを $l_m^{s \rightarrow t}$ 、サイト t で製品 P_i が生産される場合、そのモジュールスロット S_j に割り当てられるモジュールのリードタイム値 $l_{j \rightarrow i}^t$ (そのモジュールが組立拠点へ届くまでの時間) は割り当て可能性 $\epsilon_{j,k \rightarrow i}^{s \rightarrow t}$ を用いて次の式で定義される。

$$l_{j \rightarrow i}^t = \max_{k,s} \{ (l_{j,k}^s + l_m^{s \rightarrow t}) \epsilon_{j,k \rightarrow i}^{s \rightarrow t} \} \quad (5.37)$$

異なるサイトで生産されたモジュールや、異なる種類のモジュールが割り当てられる場合、それらの中の最大値を用いて評価している。

サイト t で生産される製品 P_i のリードタイム値 L_i^t は、その組立 (生産) によって生じるリードタイムを L_i^t として、モジュールのリードタイム値との総和で定義される。

$$L_i^t = \left(\max_j \{ l_{j \rightarrow i}^t \} + l_i^t \right) \quad (5.38)$$

ここで、サイト t において生産される製品 P_i の製造リードタイムは l_i^t 、サイト t からサイト r までの輸送リードタイムは $l_p^{t \rightarrow r}$ である。

サイト t で生産される製品 P_i がマーケットサイト $r(r \in \text{Market})$ まで輸送される最終的なリードタイム値 L_i^r を、構成されるモジュールのリードタイム値の中で最大値を用いて、組立生産時間と市場までの輸送時間との総和で次のように定義する。

$$L_i^r = \max_t \{ L_i^t + l_p^{t \rightarrow r} \} \quad (5.39)$$

生産される全製品の中で品質 L_i^r の最悪値 L_{max} は次の式で算出される。

$$L_{max} = \max_{r,i} \{ L_i^r \} \quad (5.40)$$

5.2.4 グローバルプロダクトファミリ設計のための多目的最適化手法の構築

定式化した最適計画問題は、設備導入などの選択を表す 0-1 変数と、生産量や輸送量を表す非負の連続変数を設計変数とした、混合整数計画問題である。本研究では、サプライチェーン設計における意思決定の階層性に着眼した多段階アルゴリズム^[89, 90]を用いて多目的最適化を行う。

サプライチェーン設計における意思決定は戦略的レベル (Strategic-level)、タクティカルレベル (Tactical-level)、オペレーショナルレベル (Operational-level) の 3 つに分類することができる^[93]。戦略的レベルは、企業に長期的に影響を及ぼす意思決定の領域である。たとえば、拠点の位置や数、それぞれの容量や処置能力などの決定が挙げられる。戦略的レベルの意思決定のいくつかは、大規模な投資を伴う。状況に合わせて頻繁に変更することができず、結果的に長期的な影響を及ぼす場合がある。タクティカルレベルは、一般に四半期から年に一度更新する意思決定の領域である。例えば、購買計画や生産決定、製品価格の決定、在庫政策の立案などが挙げられる。オペレーショナルレベルは日常的な意思決定であり、頻繁に変更できる要素である。例えば、スケジューリング、配送計画、積荷計画などが挙げられる。3 つの分類は階層的な関係にあることが知られている。タクティカルレベルの意思決定は戦略的レベルで決定された上での意思決定となるし、オペレーショナルレベルの意思決定は戦略的レベルとタクティカルレベルで決定されたことは前提として扱う。例えば、配送計画を策定する場合も、施設の位置や在庫政策はあらかじめ設定されていて、その上での意思決定を行うことになる。

多段階アルゴリズム^[89]は、3 つの意思決定レベルの階層性を利用し、設計変数を意思決定レベルに分割してそれぞれのレベルに応じた部分問題へと分割する手法である。本研究では、生産の有無を決定する 0-1 設計変数である $\delta_{j,k}^s, \delta_i^t, \delta_i^{r \rightarrow}$ を戦略的レベルの意思決定、その他の生産量や輸送量などを決定する連続設計変数 $x_{j,k}^{s \rightarrow}, x_{j,k}^{s \rightarrow t}, x_{j,k}^{t \rightarrow}, x_{j,k \rightarrow i}^t, x_i^{t \rightarrow}, x_i^{t \rightarrow r}, x_i^{r \rightarrow}$ をオペレーショナルレベルの意思決定として別々に考える。アルゴリズムのフローは図 5.2 に示される。戦略的レベルの意思決定には多目的遺伝的アルゴリズム^[18]、オペレーショナルレベルの意思決定にはシンプレックス法^[94]を用いて最適化を行う。多目的遺伝的アルゴリズムの適応度計算では、コスト、品質、リードタイムの 3 つを評価する必要がある。本問題は混合整数計画問題であるが、戦略的レベルの意思決定を固定すると、コストモデルをオペレーショナルレベルの意思決定を求める線形計画問題として定式化することができる。本章では、計算負荷低減のため、全ての指標で最適化を行うのではなく、最も重要な目的関数であるコストを最適化し、その場合での品質とリードタイムを順解析した結果を適応度として用いる。

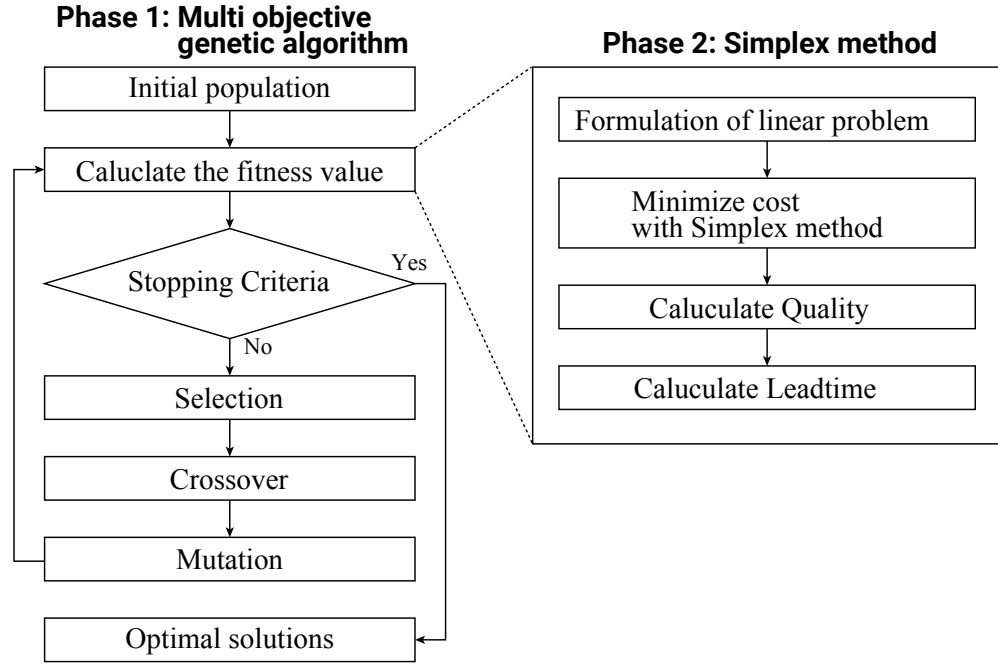


Fig. 5.2: Two-stage optimization Algorithm.

5.3 選択構造問題でのスキーマモデルの構築

本節では、提案する構造分析法を行うため、グローバルプロダクトファミリ設計問題の解空間構造を表現するスキーマモデルを定義する。グローバルプロダクトファミリ設計問題の多くの意思決定はそのまま 0-1 変数で表現することができる。本研究では、0-1 変数で構成されている解空間の構造を選択構造として定義し、グローバルプロダクトファミリの特徴を抽象するスキーマモデルを新たに定義する。

本研究の設計問題は 2 段階アルゴリズムのもと、ストラテジックレベルの設計変数とオペレーショナルレベルの設計変数とで構成されており、ストラテジックレベルの意思決定は 0-1 設計変数で定義されている。意思決定の階層性を加味すれば、オペレーショナルレベルの意思決定はストラテジックレベルの意思決定に対して従属した関係にあるので、ストラテジックレベルの意思決定のみを抜き出してスキーマを定義することができる。しかし、オペレーショナルレベルの意思決定を完全に捨象した場合、そのままではグローバルプロダクトファミリとしての特徴を把握することが難しい。そこで、連続変数のうち輸送量を表す連続変数を 0-1 変数の情報に変換する処理を加えて、混合整数計画問題のためのスキーマモデルを構築する。

スキーマモデルの満たすべき性質のうち、(B) スキーマモデル δ の各ビットはそれぞれの設計上の判断に対応しており、それらは独立した関係にあるという要件に着眼すると、連続変数のうち、モジュール生産量 $x_{j,k}^{s \rightarrow}$ 、製品生産量 $x_i^{t \rightarrow}$ 、製品販売量 $x_i^{r \rightarrow}$ は全て、製造の有無を判断する意思決

定 $\delta_{j,k}^s, \delta_i^t, \delta_i^{t \rightarrow r}$ と相関があることに注意する。ストラテジックレベルの設計変数 $\delta_{j,k}^s, \delta_i^t, \delta_i^{t \rightarrow r}$ に加えて、輸送量を表す設計変数 $x_{j,k}^{s \rightarrow t}, x_i^{t \rightarrow r}$ とモジュール割当量 $x_{j,k \rightarrow i}^t$ を次の式で 2 値化することで $\delta_{j,k}^{s \rightarrow t}, \delta_{j,k \rightarrow i}^t, \delta_i^{t \rightarrow r}$ の 3 つを定義し、6 つのバイナリ変数を連結することでスキーマを定義する。

$$\delta_{j,k}^{s \rightarrow t} = \begin{cases} 1 & (x_{j,k}^{s \rightarrow t} > 0) \\ 0 & (x_{j,k}^{s \rightarrow t} = 0) \end{cases} \quad (5.41)$$

$$\delta_{j,k \rightarrow i}^t = \begin{cases} 1 & (x_{j,k \rightarrow i}^t > 0) \\ 0 & (x_{j,k \rightarrow i}^t = 0) \end{cases} \quad (5.42)$$

$$\delta_i^{t \rightarrow r} = \begin{cases} 1 & (x_i^{t \rightarrow r} > 0) \\ 0 & (x_i^{t \rightarrow r} = 0) \end{cases} \quad (5.43)$$

式 (5.41)、(5.42)、(5.43) で定義されたスキーマモデルのビットは輸送経路の選択を意味するため、設計上の判断に対応しおり、(B) の条件を満たす。

式 (5.41)、(5.42)、(5.43) の 2 値化処理では連続設計変数の量的関係を捨象するため、この写像 f はそのままでは単射ではない。しかし、2 段階アルゴリズムに基づいて連続変数が定められることを前提にすれば、ストラテジックレベルの意思決定を 1 つに定めると 2 段階目の処理を通して一意に定めることができる。ストラテジックレベルの意思決定変数である $\delta_{j,k}^s, \delta_i^t, \delta_i^{t \rightarrow r}$ は全単射の関係があるので、このスキーマモデルへの写像 f もまた全単射であると考えてよい。また、施設選択と輸送経路選択を表す設計変数からスキーマを定義しているため、図 5.6、5.7 のように * を用いた集合表現から集団の代表的なサプライチェーン構成を再現することができる。

5.4 多数評価指標でのビルディングブロックの解釈

5.4.1 ロジスティック回帰による分岐理由の考察

提案手法の Step 3 では、階層的クラスタリングの結果を踏まえ、解空間構造の分析と構造的相違点の意味を考察する。解空間は構造を基準に階層的に分割されるが、3.6.2 項で述べたように、分割面での左右の違いの意味を考察するには、評価指標を用意することと、構造特徴を明らかにすることの 2 つの方法がある。グローバルプロダクトファミリの最適設計では、その概念が広範であることに起因して評価方法が多様である。5.2.3 項で述べた 5 つの性能評価指標に代表されるように、それぞれが設計における重要な意味を持つため、左右の違いを評価指標を用いて評価することが妥当であると考えられる。順列フローショップスケジューリング問題のように、評価関数が少ない場合は、4.5 節で導入した可視化に基づく分析方法である程度の考察が可能である一方、評価方法が多様である場合は、散布図行列等を用いて可視化をしたとしても認知的負荷が高く、評価関数の特徴を把握することが難しい。そこで本節では、評価方法が多様である場合で考察を可能にするた

め、Tsumoto ら^[95]によるロジスティック回帰分析を用いた設計探査手法を応用した構造分析法を導入する。

ロジスティック回帰分析とは、ベルヌーイ分布に従う変数の統計的回帰モデルの一種であり、いくつかの説明変数から2値の結果が起こる確率を説明する統計手法である^[96]。ある事象の起こりやすさを表すオッズ比は $\frac{p}{1-p}$ と書くことができ、 p はクラスラベル $y = 1$ に所属する確率を表す。ロジット関数は次の式で定義され、 $p \in [0, 1]$ のオッズ比を実数全体に変換する。

$$\text{logit}(p) = \log \frac{p}{1-p} \quad (5.44)$$

ロジスティック回帰では、この関数を用いて、次のように説明変数の値と対数オッズとの関係を表す。

$$\text{logit}(p(y = 1|\mathbf{x})) = \sum_{i=0}^n w_i x_i \quad (5.45)$$

ここで、 $p(y = 1|\mathbf{x})$ は、特徴量 $\mathbf{x}$ が与えられた時、クラス 1 に所属する条件付き確率である。あるデータ点が特定のクラスに所属している確率を予測するために、ロジット関数の逆関数である、ロジスティックスシグモイド関数を考える。ロジスティックスシグモイド関数は次の式で表される。

$$\phi(z) = \frac{1}{1 + e^{-z}} \quad (5.46)$$

ここでの z は重みと入力の線形結合であり、次式で表される。

$$z = \sum_{i=0}^n w_i x_i \quad (5.47)$$

ロジスティック回帰モデルは重み $\mathbf{w}$ をパラメータとして有しており、対数尤度を最大化する方針で重み $\mathbf{w}$ を学習する。シグモイド関数は $\phi(z) = 0.5$ を切片として、入力値を $[0,1]$ の値に変換することができるため、特徴量 $\mathbf{x}$ が重み $\mathbf{w}$ でパラメータ化されるとすれば、シグモイド関数の出力はあるデータ点がクラス 1 に所属している確率を表していると解釈できる。つまり、あるデータ点のクラスの判別 $\hat{y}$ は次のようにまとめられる。

$$\hat{y} = \begin{cases} 1 & \phi(z) \geq 0.5 \\ 0 & \phi(z) < 0.5 \end{cases} \quad (5.48)$$

さらに、シグモイド関数の特性を念頭におけば、次の関係が成立する。

$$\hat{y} = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases} \quad (5.49)$$

Tsumoto ら^[95]による方法では、ロジスティック回帰モデルの重み $\mathbf{w}$ に焦点を当てて、クラスラベルの特徴を説明する。評価指標 $\mathbf{f}$ を説明変数としてロジスティック回帰モデルを作成した場合、分類面は次の式で表される。

$$s = w_0 + w_1 f_1 + \cdots + w_n f_n \quad (5.50)$$

重み w の値から、次のように分類面の特徴を考察することができる。まず、重みの絶対値 $|w_i|$ は評価指標 f_i の影響度を評価する。ある評価指標 f_i の変動に対して重みの絶対値 $|w_i|$ が相対的に大きい場合は、 s の値が大きく変動するため、分類への影響が大きいと解釈することができ、分類におけるレバレッジポイントであると考えることができる。ただし、重みの絶対値は評価指標の大きさに影響されるため、全ての評価指標 f で正規化を行わなければならない。次に、重みの符号からクラスラベルの特徴を説明する。 $w_i > 0$ が成立する場合、評価指標 f_i の値と s とに正の相関があることから、 $y = 1$ のクラスラベルでは評価指標 f_i の値が比較的大きい集団であると解釈することができる。逆に、 $w_i < 0$ である場合は、 $y = 1$ のクラスラベルでは評価指標 f_i の値が小さくなることを意味するし、 y_0 のクラスラベルで評価指標 f_i の値が大きくなる。

本手法では、式 (5.50) に表されるロジスティック回帰モデルの定数項 w_0 の影響は解釈に用いない。式 (5.51) の変量変換を行うことで、分割面の式 (5.52) を得ることができる。

$$f'_i = f_i + \frac{w_0}{w_1 + \dots + w_n} \quad (5.51)$$

$$s = w_1 f'_1 + \dots + w_n f'_n \quad (5.52)$$

式 (5.51) の変量変換は観測値の並行移動にすぎず、重み w_i を用いて評価関数のトレードオフ関係などの相対的な関係を理解する場合は、 f_i と f'_i は強い正の相関があるので、 w_0 の項は解釈に影響しないと考えられる。

本研究では、デンドログラムの各分岐点で評価指標の値に応じて、左右どちらのクラスタに所属するのかを予測する。つまり、ここでの 2 値の結果とはどちらのクラスタに分岐するのかを表しており、ここでは、クラスラベル $y = 1$ を左クラスタに所属することとして考えることにする。ある分岐点 l での左右クラスタの代替案を入力に分割面を学習し、次のような分割面の式を得る。

$$s^{(l)} = w_0^{(l)} + w_1^{(l)} f_1^{(l)} + \dots + w_n^{(l)} f_n^{(l)} \quad (5.53)$$

それぞれの分岐点での重み $w_i^{(l)}$ の絶対値と符号から、左右のクラスタのビルディングブロックの違いの効果を解釈する。例えば、 $w_1^{(l)} > 0$ 、 $w_2^{(l)} < 0$ でありながら、どちらの絶対値も充分大きい場合、 f_1 と f_2 の評価指標にトレードオフの関係があることを読み取れる。 f_1 の指標を大きくすると $s^{(l)}$ が大きくなるので、意思決定者が f_1 を重要視する場合、 y_1 の左クラスタの中に嗜好に沿う解が含まれていることがわかる。逆に、 f_2 の指標を重要視する場合は、 f_2 の増加に伴って $s^{(l)}$ が小さくなるので意思決定者は右クラスタを選択して、次の分岐点へと移動する。重みの絶対値 $|w_i|$ が同程度である場合、等価解のように、評価関数の関係が同程度であることを表している。同等の評価値を実現するための方法が複数あることを意味しており、代替案の多様性をもたらすビルディングブロックとして解釈することができる。

5.4.2 意思決定分岐図を用いた情報の整理

それぞれの分岐位置におけるスキーマの相違点とロジスティック回帰によって明らかになったクラスタの特徴は意思決定分岐図として1つのテンプレートのもとで管理される。本章で用いる、グローバルプロダクトファミリ設計での意思決定分岐図のテンプレートを図5.3に示す。グローバルプロダクトファミリ設計でのスキーマモデルは主にストラテジックレベルの意思決定を抽象しているが、同決定事項は長期的に影響を及ぼすため、選択することと同様に、選択しないことも重要な意味をもつ。例えば、施設の設立を表す意思決定の場合、選択には莫大な費用が発生することは少なくなく、設立しないことの選択もコスト削減の観点から利益向上に貢献する。特にグローバルプロダクトファミリ設計では、サプライチェーンでの諸決定がスキーマに対応しているため、Category C に分類される $(*, 0)$ のビットの組合せは、潜在的なニーズに対してどのように供給経路や供給先を限定するのかという点で重要な知見が潜んでいる。そこで、意思決定分岐図ではこれまでの焦点であった Category A 及び、Category B の要素に加えて、Category C の $(*, 0)$ 要素に焦点を当てる。図5.3の下側ではある分岐点 Branch X での2つのサプライチェーンを表している。それぞれのビットに対応して、1のビットは黒、*のビットは灰色で示している。施設選択に関連するビットのみ0要素を点線で示している。左右のクラスタの共通する部分は地点 S^1 で製品 P_1 に関するモジュール生産と組み立てを行なっていることである。共通するビルディングブロックである Category A は1行目に整理される。左右の明確な違いは P_1 の販売地点であり、Category B として2行目に整理されている。*の要素はクラスタ内での意思決定が曖昧であることを示しているものの、サプライチェーンの展開の方針を読み取ることができる。左クラスタでは地点 S^1 で生産が閉じている一方、右クラスタの一部の解では製品 P_4 を地点 S^2 で生産している。また、販売の方針としても P_2 と P_3 を地点 S^1 で販売するか地点 S^2 で販売するかの違いがあり、これらは Category C として2行目に記載されている。

意思決定分岐図の3行目にはロジスティック回帰によって学習した重み $w^{(x)}$ が柱状グラフで示されている。ここでのロジスティック回帰によるクラスタの特徴量の解釈では、5.2.3項で構築した目的関数を評価関数として用いることにする。目的関数のうち、式(5.32)は3つの指標の線形和であることを受け、それぞれを分けて次のような5つの評価指標を用いる。なお、直感的な解釈のため、値が大きいほど評価関数値が良好になるように、符号を反転させる処理を加えている。

$$f_1 = Sales = \sum_i \sum_r p_i^r x_i^{\rightarrow r} \quad (5.54)$$

$$f_2 = -Variable\ cost = -(c_{MP} + c_{MT} + c_{PA} + c_{PT}) \quad (5.55)$$

$$f_3 = -Fixed\ cost = -(C_{MP} + C_{PA} + C_S + C_{PS}) \quad (5.56)$$

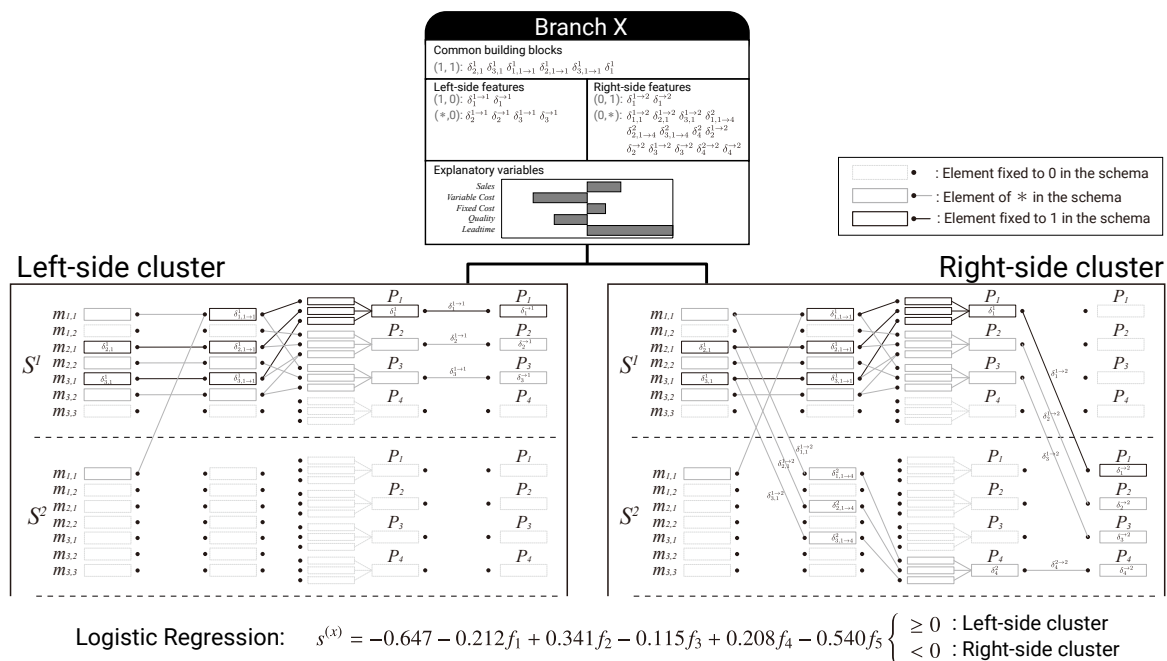


Fig. 5.3: Template for decision bifurcation diagram: left and right side supply chain schemas and histogram representing the results of the logistic regression.

$$f_4 = Q_{min} = \min_{t,i} \{ Q_i^t \} \quad (5.57)$$

$$f_5 = -L_{max} = \min_{r,i} \{ -L_i^r \} \quad (5.58)$$

棒グラフの長さはそれぞれの評価指標の影響度を表しており、重みの符号は柱の向きで表されている。図 5.3 の例であれば、リードタイム f_5 の重みが最も絶対値が大きいので、柱が長い。符号は負であるので、右クラスタはリードタイム値が良好である集団である。一方で、変動費 f_2 や品質 f_4 の指標は正の重みであることからトレードオフの関係があり、これらの値が良好であるのは左クラスタである。意思決定者はこれらの柱状グラフとサプライチェーンの図を参照しながら、嗜好に沿ったクラスタを逐次選択することになる。

5.5 事例：自動車のグローバルプロダクトファミリ設計

5.5.1 概要

本節では、提案する多数評価指標でのビルディングブロックの解釈法が機能することを自動車のグローバルプロダクトファミリー設計問題^[92]を例に引いて確認する。

自動車はグローバルに生産や販売を展開しながらも共通化や標準化による効果が発揮されている
 典型例である。多くの自動車メーカはプラットフォームの共通化を行い、一部のモジュールを変更

Tab. 5.3: Module implementability.

	P_1	P_2	P_3	P_4
$m_{1,1}$	1	0	1	0
$m_{1,2}$	0	1	1	1
$m_{2,1}$	1	1	0	0
$m_{2,2}$	0	0	1	1
$m_{3,1}$	1	1	0	0
$m_{3,2}$	1	1	1	0
$m_{3,3}$	1	1	0	1

することで製品のグレードや種類を増やしている^[97]。近年のグローバルな生産を想定するのであれば、基本骨格になるドアやシートなどは共通モジュールとして据えつつ、地域ごとの要求仕様に合わせてエンジンや変速機などのコアモジュールを柔軟に変更することでコストメリットを活かしながら生産を行うことができる。また、タイヤ等に代表される部分は、各地で他企業の製品を購入して最終的な製品が組み上げられる場合もある。つまり、汎用モジュールはいずれの地域でも生産可能という意味においてグローバルに生産が行われている一方で、コアモジュールはある限定された地域、つまりローカルに生産が行われている。このような考え方はグローバルプロダクトファミリー設計として世界に広がるサプライチェーンとプロダクトファミリーの統合による効果の1つである。

本節ではまず、5.2.3 項で構築したグローバルプロダクトファミリーの数理モデルをもとに問題設定を行い、多段階アルゴリズムによって多様かつ有望な代替案の集合を獲得する。5.3 節で導入した選択構造問題のスキーマモデルを用いて階層的クラスタリングを実施し、解空間の構造分析を行う。構造分析では5つの評価指標の関係を調査するためにロジスティック回帰分析が実施され、結果は意思決定分岐図にまとめられる。意思決定分岐図から代替案の関係性を理解できることを確認し、提案する方法によって得られた結果の妥当性と効果を考察する。

5.5.2 問題設定と計算条件

本事例で考える自動車の製品系列は3つのモジュールによって構成されており、2車種に対してそれぞれ2グレードを想定しての製品系列を展開する。車種によって共通の要素としてプラットフォームとボディが共通化できると仮定し、グレードに影響するモジュールは車種を問わず共通化できると仮定する。また、走行能力を決定するエンジンやタイヤは車両のグレードにある程度依存するものの、より高性能のモジュールで代替しても良いと仮定する。これらの仮定により、3つのモジュールに対して組合せを変えることで、4種類の製品が展開できるような製品系列を想定する。

Tab. 5.4: Outline of the example problem; differences among three sites.

Sites	Cost	Quality	Leadtime	Market size	Production capacity	Transportation capacity
S^1	Higher	Better	Moderate	Larger	Moderate	Larger
S^2	Lower	Worse	Longer	Larger	Larger	Larger
S^3	Moderate	Moderate	Shorter	Moderate	Smaller	Larger

具体的な製品系列の設定は次のようにまとめられる。

- 製品数: 4
- スロット数: 3
- 全モジュール候補数: 7
- スロット 1 内モジュール数: 2
- スロット 2 内モジュール数: 2
- スロット 3 内モジュール数: 3
- 生産サイト候補数: 3
- 市場候補数: 3

モジュールの製品への割り当て可能性 $\gamma_{j,k \rightarrow i}$ は表 5.3 にまとめられる。なお、モジュールスロット 1 はグレード、モジュールスロット 2 は車種を決定し、モジュールスロット 3 は走行性能を決定する。3 つの生産拠点と市場の特徴は表 5.3 のようにまとめられ、サプライチェーンや製品系列の諸パラメータを設定している。3 つの目的関数に対応して、それぞれ得意とする地点が異なっている。具体的にはコストを低く抑えたい場合は地点 S^2 を中心とする供給網を築くことが良い。同様に、地点 S^1 は品質に優れており、地点 S^3 はリードタイムに優れている。詳細なパラメータの設定は先行研究^[92]に従い、5.2.3 で構築したモデルを用いて、45 個の 0-1 設計変数と、249 個の実数変数の混合整数計画問題を次のように定式化する。

$$\begin{aligned}
& \text{find} && \boldsymbol{\delta} = [\delta_{j,k}^s, \delta_i^t, \delta_i^{\rightarrow r}] \\
& && \boldsymbol{x} = [x_{j,k}^{s \rightarrow}, x_i^{t \rightarrow}, x_i^{\rightarrow r}, x_{j,k}^{s \rightarrow t}, x_i^{t \rightarrow r}, x_{j,k \rightarrow i}^t] \\
& \text{minimize} && \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\delta}) = [-Prof_i, -Q_{min}, L_{max}] \\
& \text{subject to} && \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\delta}) \leq 0 \\
& && \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\delta}) = 0 \\
& && \boldsymbol{\delta} \in \{1, 0\} \\
& && \boldsymbol{x} \in \mathbb{R}
\end{aligned} \tag{5.59}$$

ここで、3 つの目的関数はそれぞれ式 (5.32)、(5.36)、(5.40) で定義されており、不等式制約条件 $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\delta})$ は、式 (5.5)、(5.11)、(5.14)、(5.16)、(5.20)、等式制約条件 $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\delta})$ は式 (5.3)、(5.4)、

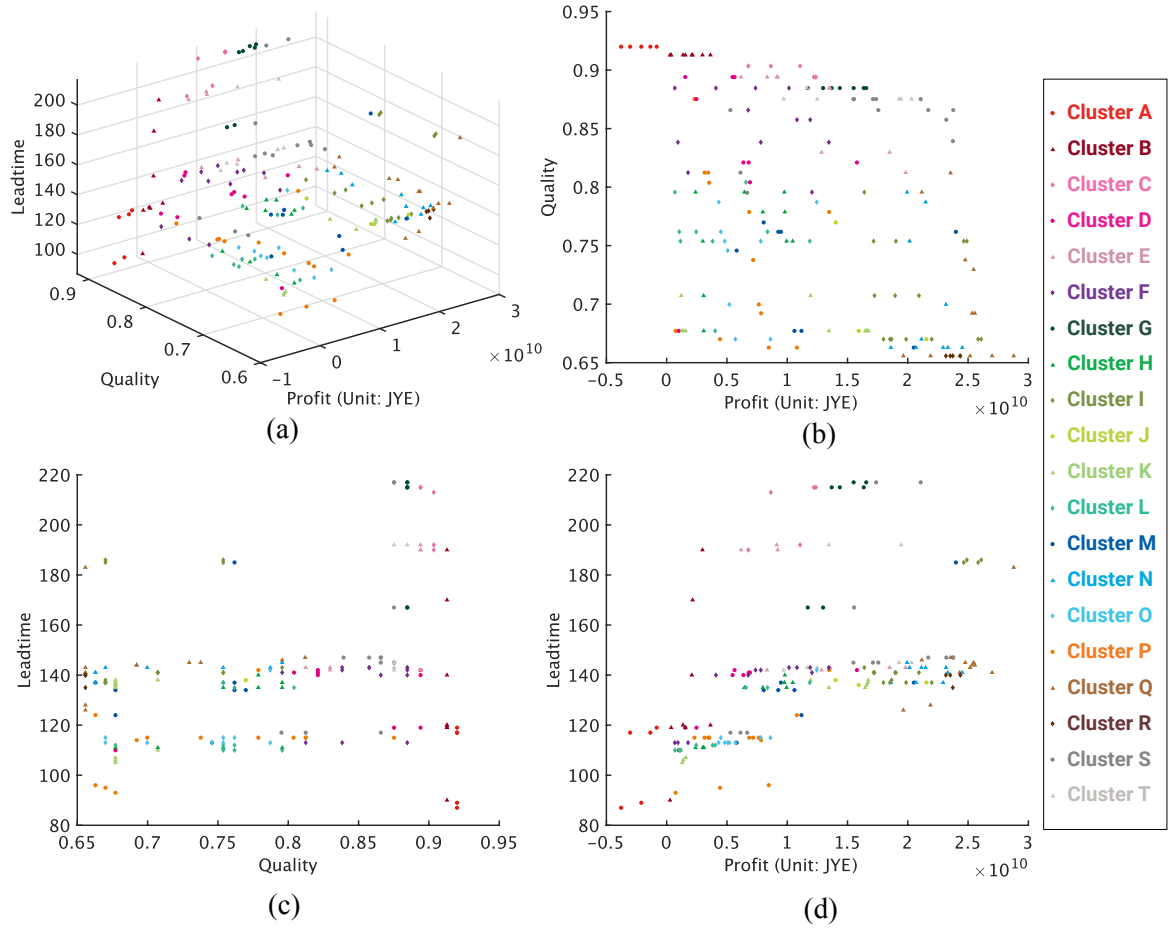


Fig. 5.4: Scatter plot of Pareto front: (a) cross-sectional perspective view, (b) plot to profit-quality section, (c) plot to quality-leadtime section and (d) plot to profit-leadtime section.

(5.8)、(5.9)、(5.12)、(5.13)、(5.15)、(5.18)、(5.19) である。

多段階アルゴリズムによってパレート最適解である 177 個の代替案を獲得した。多目的最適化のパレート最適解の散布図を図 5.4 に示す。なお、多段階アルゴリズムの上位問題には NCGA 法 (近傍培養型遺伝的アルゴリズム)^[98]、下位問題にはシンプレックス法を用いた。本事例では目的関数の 1 つである利益 *Profit* の昇順で代替案にラベルをつけ、以降は代替案の番号を用いて議論を展開する。

5.5.3 解空間の構造分析結果

177 の代替案に対してスキーマを定義し、階層的クラスタリングを実行した。デンドログラムの視認性を重視して、距離関数にユークリッド距離、リンケージ手法にウォード法を用いた。作成されたデンドログラムを図 5.5 に示す。177 の代替案に対して 176 の分岐点が存在するが、最下層の分岐構造は全体の振る舞いを左右するというよりは多様性をもたらす要素の分岐であると考え、図

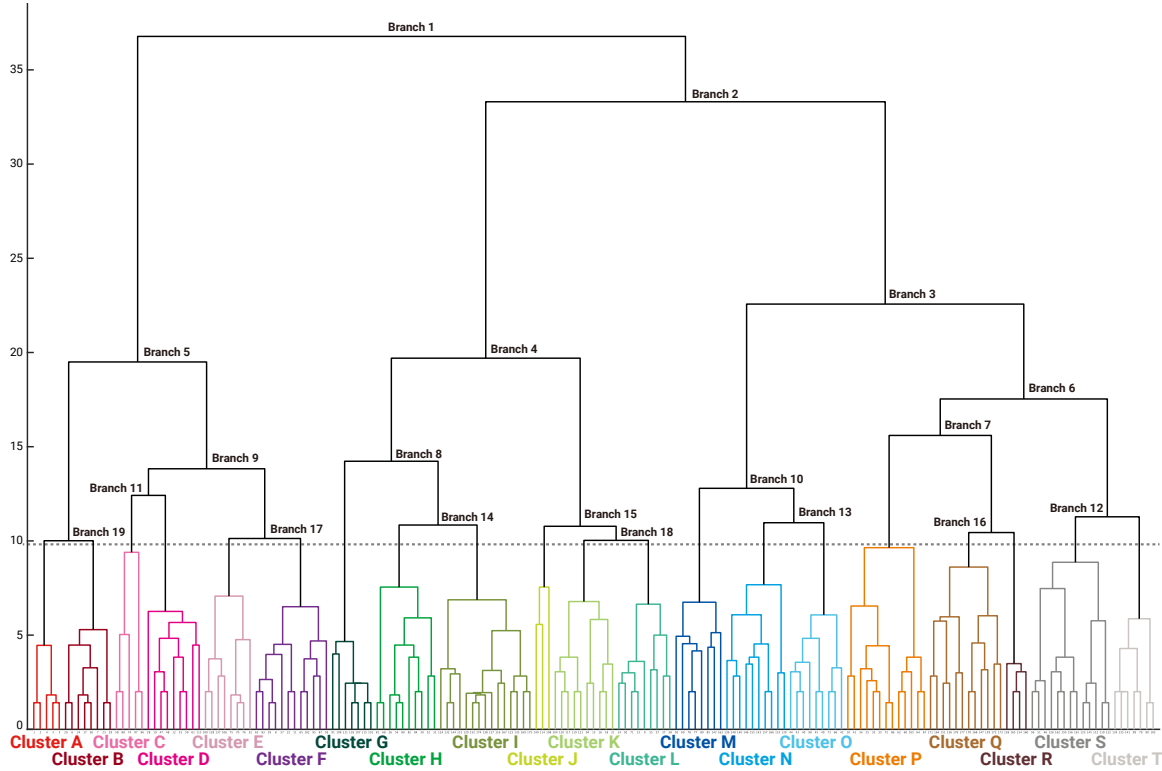


Fig. 5.5: Dendrogram.

5.5 の点線で分けられる 20 のクラスタに焦点を当て、19 の分岐点について考察を行う。それぞれのクラスタはデンドログラムで左から右に向かうように Cluster A から Cluster T までアルファベット順にラベルをつけた。なお、それぞれのクラスタに属する代替案の目的関数空間上での分布は図 5.4 から確認できる。20 のクラスタのスキーマを図 5.6, 5.7, 5.8, 5.9 に示す。

19 の分岐点についてスキーマの相違点とロジスティック回帰分析の学習結果をまとめた意思決定分岐図を作成した。分岐数が多い関係で 3 つの部分に分割し、Branch 1 で左に分岐した際の意思決定分岐図を図 5.10 に、Branch 1 で左に分岐し、Branch 2 で左に分岐した意思決定分岐図を図 5.11 に、Branch 2 で右に分岐した意思決定分岐図を図 5.12 に示す。なお、意思決定分岐図の各ブロックには対応するビットの通し番号を記載している。

5.5.4 考察

前項で得られた結果において、組合せを伴う解空間の特徴がグローバルプロダクトファミリー設計においても当てはまることを確認し、本章で提案した解空間の構造分析法が機能しているかを考察する。

まず、グローバルプロダクトファミリー設計の代替案の特徴を整理する。図 5.4 に示されるように、目的関数が 3 つの場合はパレート最適解は目的関数空間での曲面である。さらに、今回の評価指標

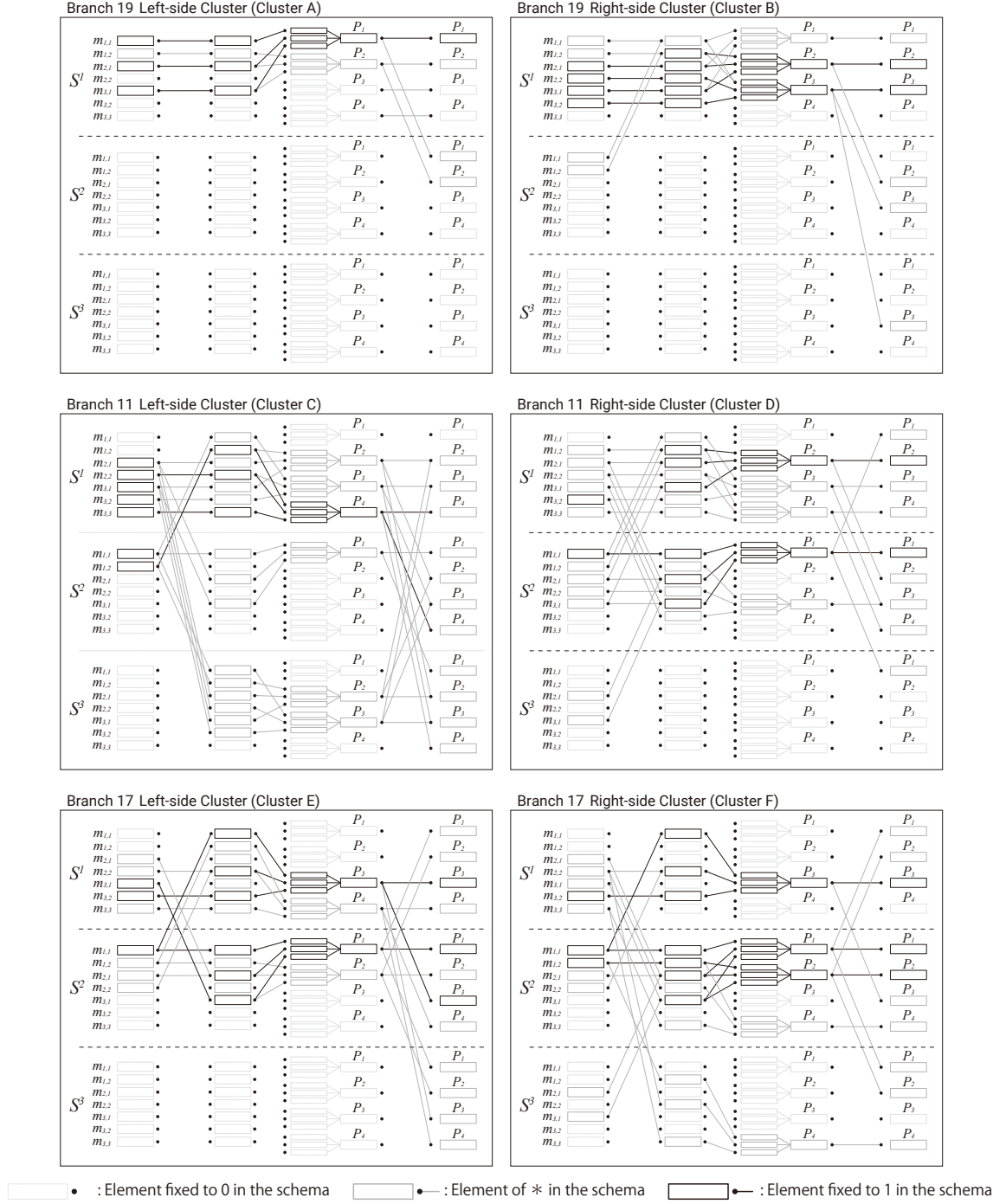


Fig. 5.6: Schemas of the 6 clusters: Cluster A, B, C, D, E and F.

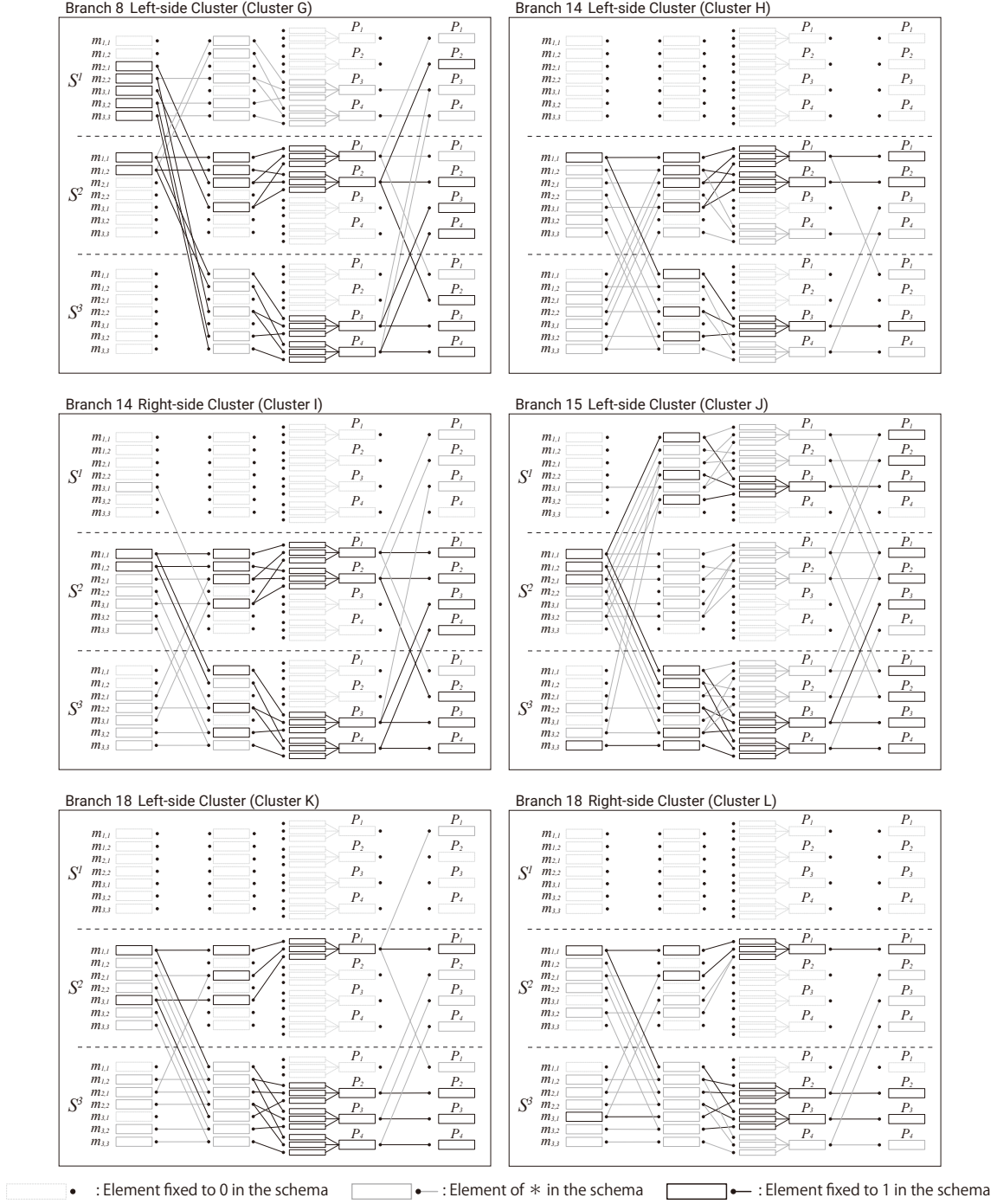


Fig. 5.7: Schemas of the 6 clusters: Cluster G, H, I, J, K and L.

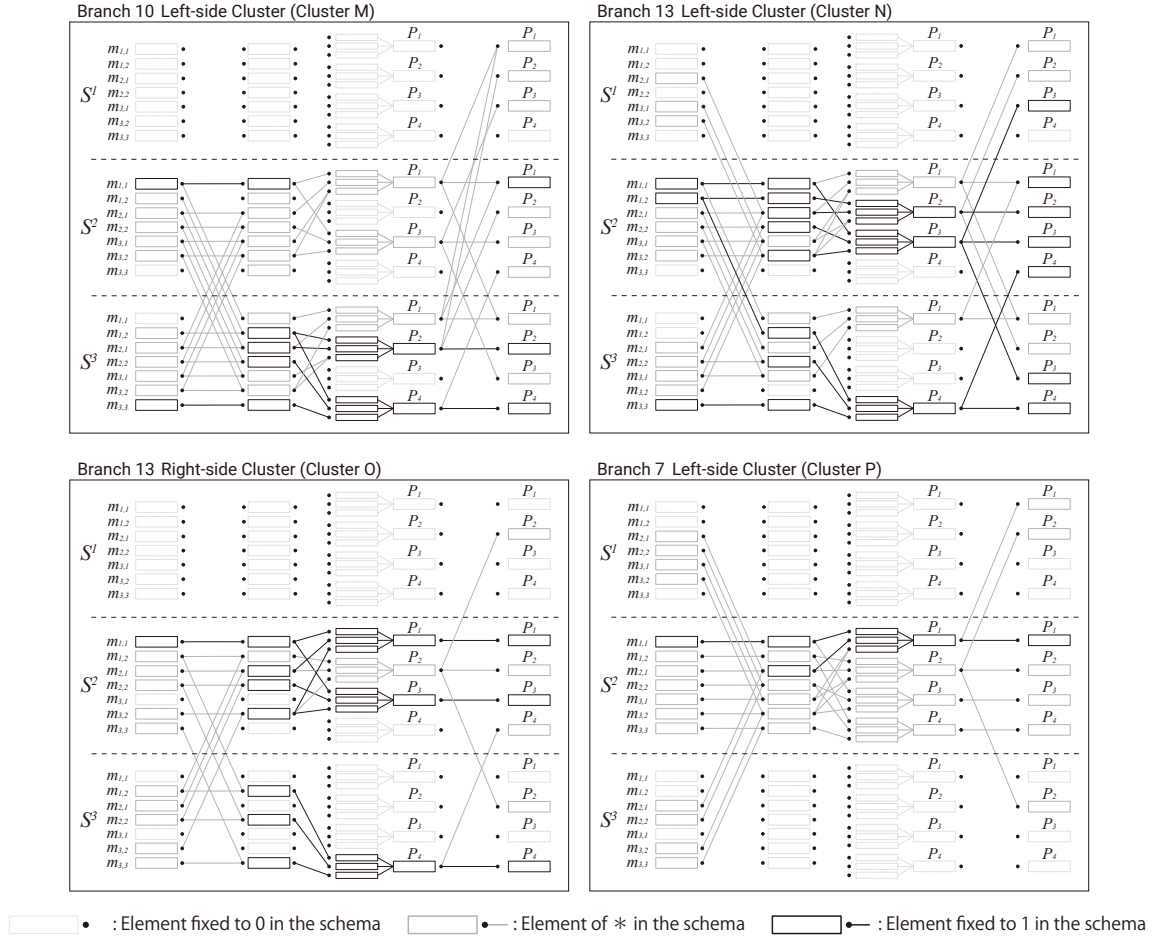


Fig. 5.8: Schemas of the 4 clusters: Cluster M, N, O and P.

空間であれば、5次元の高次元曲面になるため、単純な可視化からその分布の傾向を理解することが難しいことは多目的設計探索^[5]や Trade space exploration^[6]の述べる所と一致している。さらに、図 5.4(c) からわかるように、品質やリードタイムの目的関数では代替案が直線上に分布しており、それぞれの指標での等価解が複数存在していることがわかる。Cluster A や Cluster B では品質の値が同じ等価解の集合であり、解空間構造を入力にしたクラスタリングの結果で同じクラスに分類されていることから、解空間構造が目的関数値と 1:1 に対応したビルディングブロックの存在を観測できる。一方で、リードタイムは 120 付近と 140 付近に多くの代替案が混在しており、構造が異なりながらも類似した代替案が存在していることが観測できる。これらの結果は、解空間構造に着眼しなければ代替案の区別ができないことを示唆している。したがって、本事例で扱ったグローバルプロダクトファミリー設計でも 2.4.3 項で述べた等価解の存在による課題を抱えており、目的関数のみで代替案の解釈を行うことの限界を確認できる。

次に、本節で導入した多数評価指標でのビルディングブロックの解釈方法の妥当性を評価するため、Branch 12 を例に挙げて考察を行う。Branch 12 の意思決定分岐図要素と左右のクラス

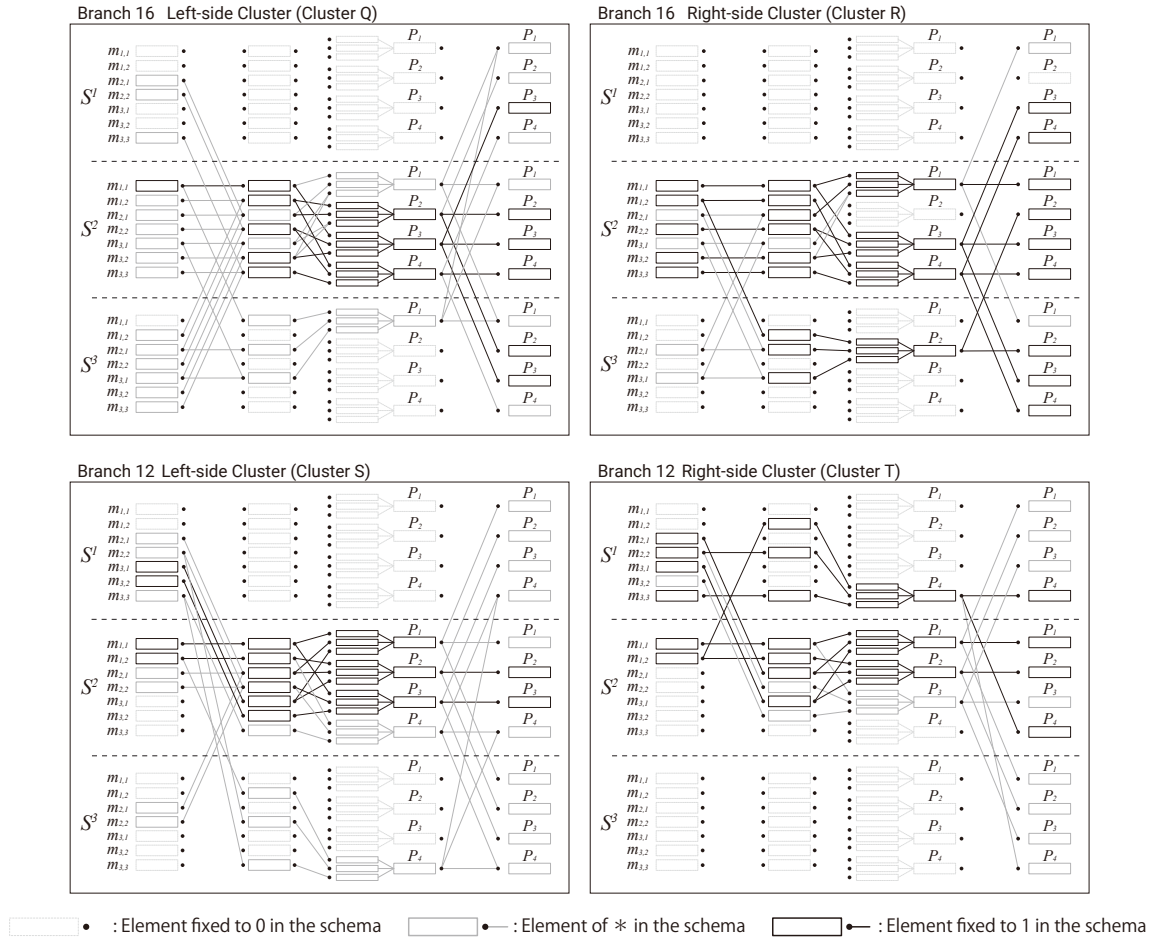


Fig. 5.9: Schemas of the 4 clusters: Cluster Q, R, S and T.

タのスキーマは図 5.13 にまとめられる。Branch 12 での左クラスタが Cluster S、右クラスタが Cluster T である。2 つのスキーマの相違点は意思決定分岐図にまとめられている。2 つの共通ビルディングブロックは太線で強調されており、製品 P_1 と P_2 を地点 S^2 で組み立てることに関連するビルディングブロックである。生産拠点の基本は S^1 であり、輸送コストの高いモジュールスロット 1 のみ地点 S^2 で生産している。Category B として記録されている要素は $(0, 1)$ の要素のみであり、Cluster T では製品 P_4 を地点 S^1 で組み立てることに関連するビルディングブロックである。Cluster S での製品 P_4 の扱いは多様であり、地点 S^2 か地点 S^3 、あるいはその両方で組み立てる代替案が混在している。Cluster T では地点 S^1 でのみ生産を行うため、これらの特徴は Category C である $(*, 0)$ のビットとして記録されている。ロジスティック回帰分析の学習結果では、品質 f_4 と変動費 f_2 の順で評価指標の影響が大きく、双方が高いほど Cluster T に分類される。一方で、売上 f_1 、固定費 f_3 、リードタイム f_4 は影響度が小さいものの、高いほど Cluster S に分類されるため、これらにはトレードオフの関係があると結論づけることができる。品質の影響度が高いことの原因としては、モジュール生産の拠点を S^1 中心にしていることが影響していると考え

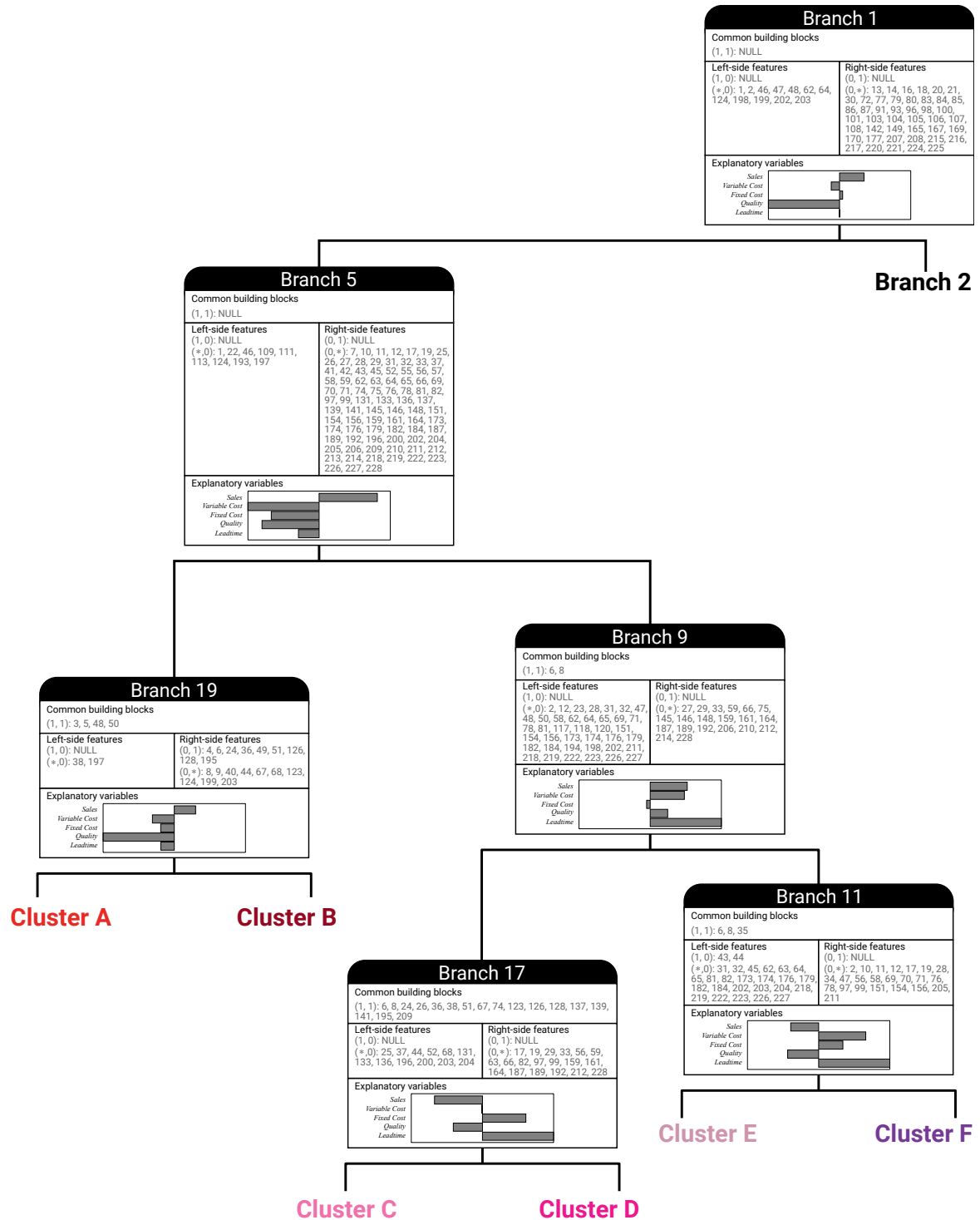


Fig. 5.10: Decision bifurcation diagram on the left side of Branch 1.

られる。表 5.3 より、 S^1 はコストが高いものの品質は良いという特徴がある。変動費については Cluster T で製品 P_3 を生産しない可能性があることなどから抑えられているものの、品質を引き換えに固定費が高くなり、トレードオフの関係が生じていると考えられる。以上より、Branch 12

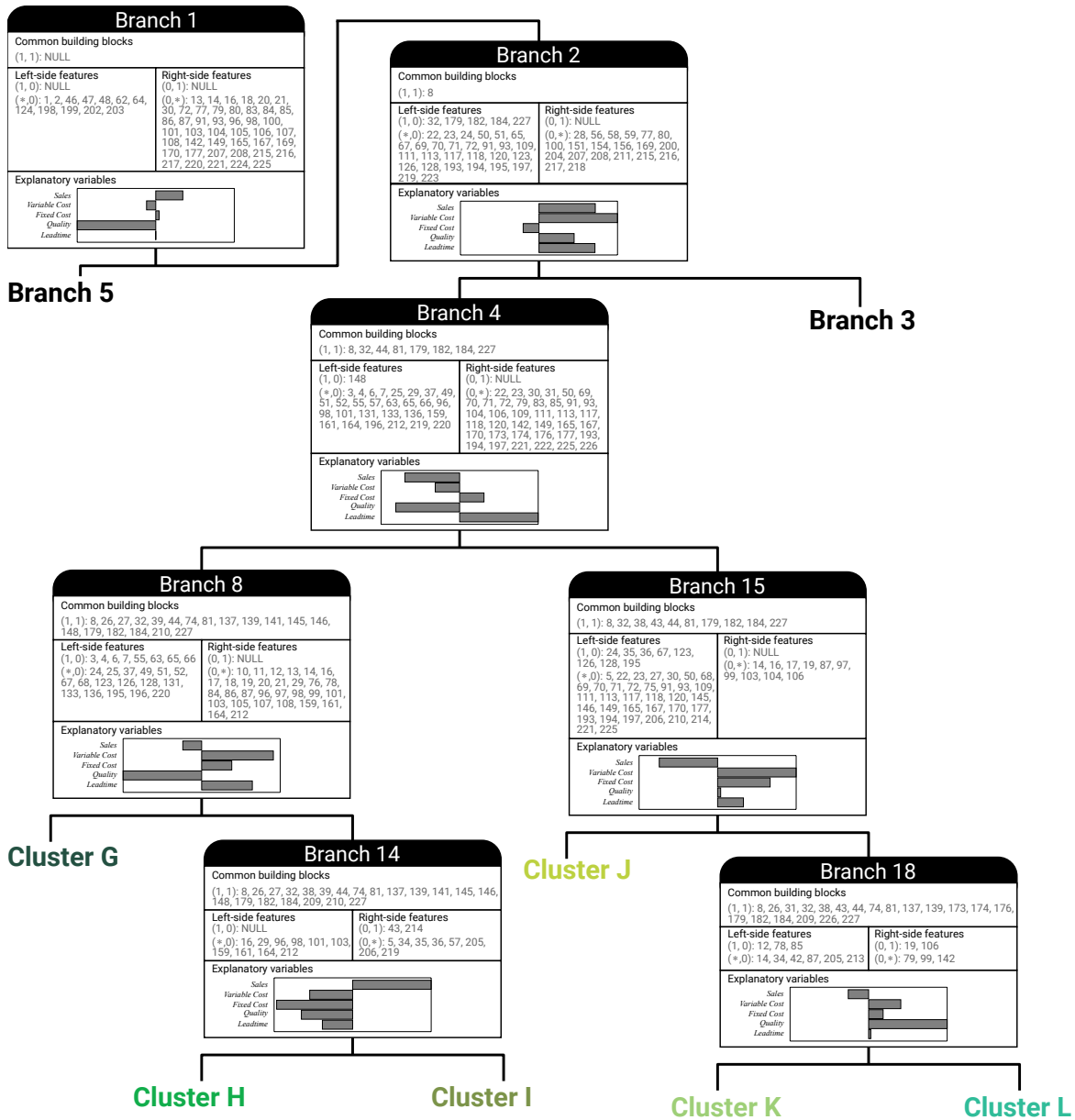


Fig. 5.11: Decision bifurcation diagram on the left side of Branch 2.

での意思決定の着眼点は製品 P_4 の組立拠点に関する取り扱いであり、品質を優先する場合は地点 S^1 で組立てを行い、合わせて上位段階で意思決定されていた製品 P_1 と P_2 に関連するモジュール生産も付随して S^1 で行うことになる。この場合の販売先や製品 P_3 の取り扱いは曖昧であり、Cluster T 以下の分岐での焦点である。逆に、品質を犠牲に売上やリードタイムを優先する場合は Cluster S に焦点を当て、製品 P_4 の組立て拠点を以降の分岐で決定することになる。したがって、ロジスティック回帰分析の結果の妥当性を確認し、それに伴う解釈可能性を確認できた。

次に、提案する枠組みを通して代替案の特徴を理解できるかを考察するため、最も利益の高い代替案 177 に焦点を当てる。意思決定分岐図では、Branch 1 から順に 9 回の分岐を経て、代替案

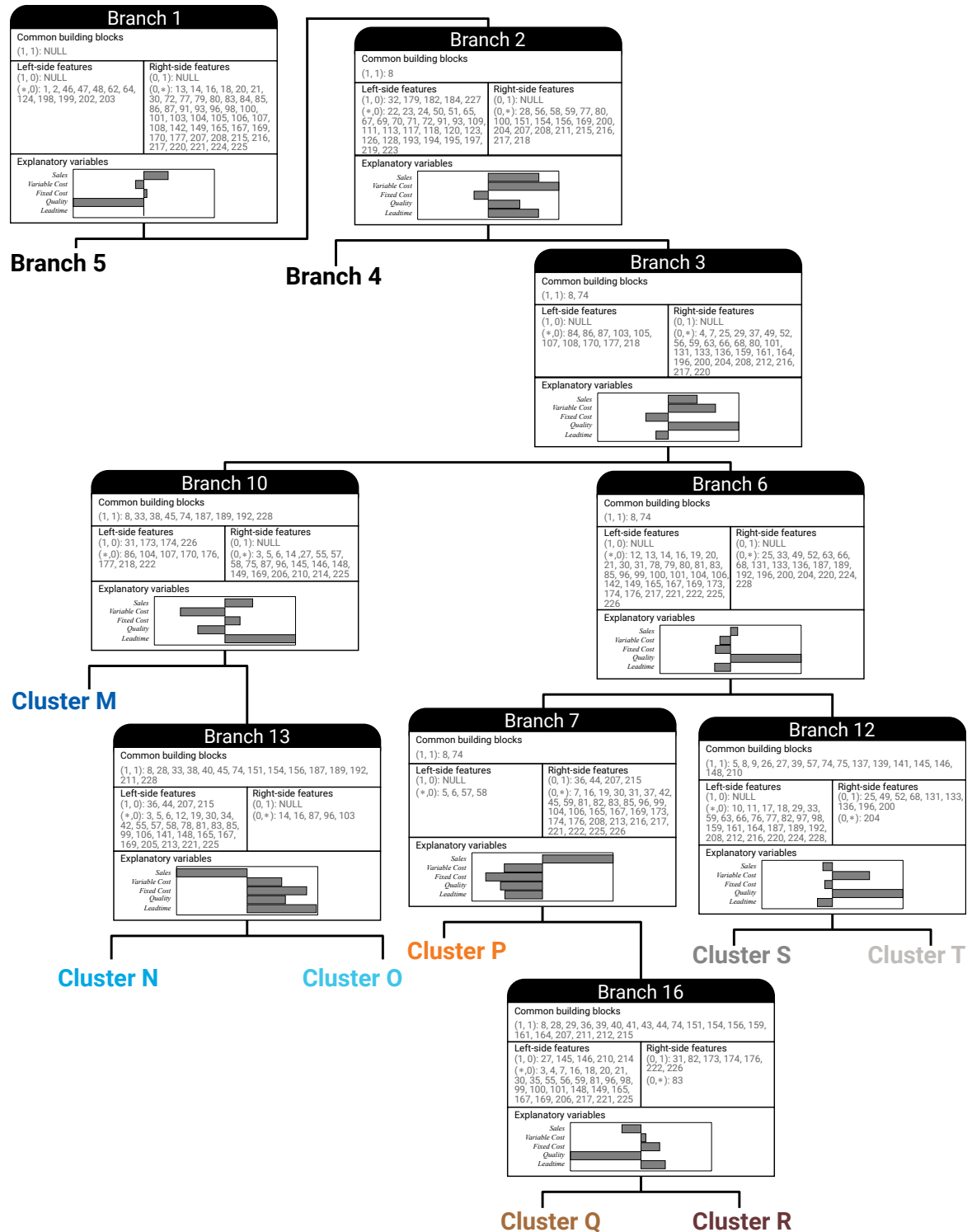


Fig. 5.12: Decision bifurcation diagram on the right side of Branch 2.

177 に至る。分岐の経緯とそれぞれの分岐点での意思決定分岐図を図 5.14 に、スキーマを図 5.15 に示す。Branch 101 では右クラスが代替案 177、左クラスが代替案 176 であり、それぞれが集団ではないことから、ロジスティック回帰分析を行わず、評価関数値を記載している。代替案

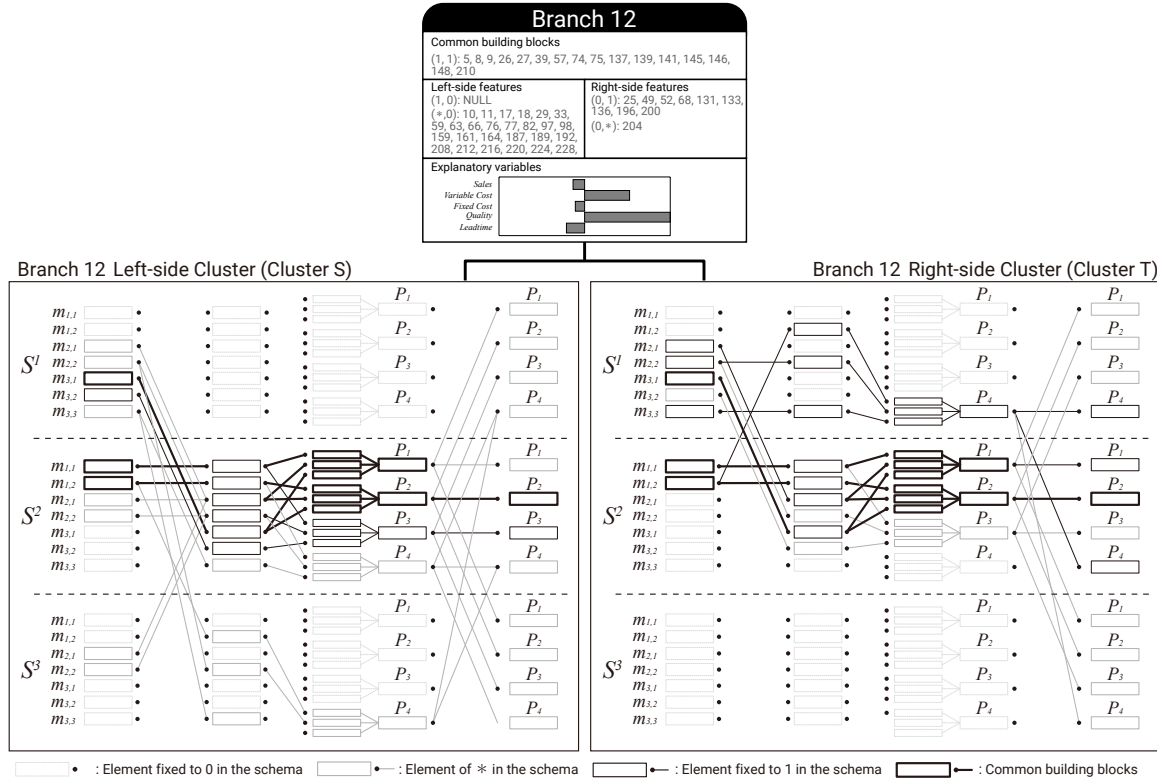


Fig. 5.13: Box in Branch 12 and schemas for left-side and right-side clusters.

177 のモジュール選択とその供給網は図 5.15 の右下に表されており、製品 P_2 、 P_3 、 P_4 を地点 S^2 で組み立て、製品 P_1 のみ地点 S^3 で組み立てた後、全ての市場に供給している。モジュール $m_{1,1}$ とモジュール $m_{2,1}$ は組み立て地点外から輸送されたモジュールを用いてグローバルな共通化が行われている。その他モジュールは組み立て地点で生産され、多くのモジュールは複数製品で共通化されている。代替案 177 の利益が高い理由はコストの低い S^2 を生産拠点の中心にしていることが考えられるものの、その詳細な戦略やその他の代替案との相違点はこれ自体では明確ではない。Branch 1 から Branch 101 までの分割過程からビルディングブロックの集積を明らかにすることで、代替案 177 に潜む戦略を考察する。

代替案 177 は大部分の組立てを地点 S^2 で行うことによって高い利益を出していることは前述の通りであるが、その絞り込みの過程は意思決定分岐図から解釈することができる。Branch 1、Branch 2、Branch 3 では全て売上やコストの評価が良好なクラスタへと分岐を進め、生産コストや組立てコストの高い地点 S^1 や地点 S^3 での拠点を選択しない方針のもと 0 で固定するビルディングブロックを集積させていることがわかる。Branch 6 では初めて売上が負に働く方向へと分岐を進めている。Branch 6 でのスキーマは図 5.16 に表される。Branch 6 での焦点は品質であり、右クラスタを選択すると品質を向上させることができる。Branch 6 の右クラスタでは地点 S^1 で生産した品質の良いモジュールをグローバルに展開している特徴がある。特にモジュールスロット 3 は全て地点 S^1 から供給しており、モジュール輸送費が高くつくものの、品質の底上げに貢献し

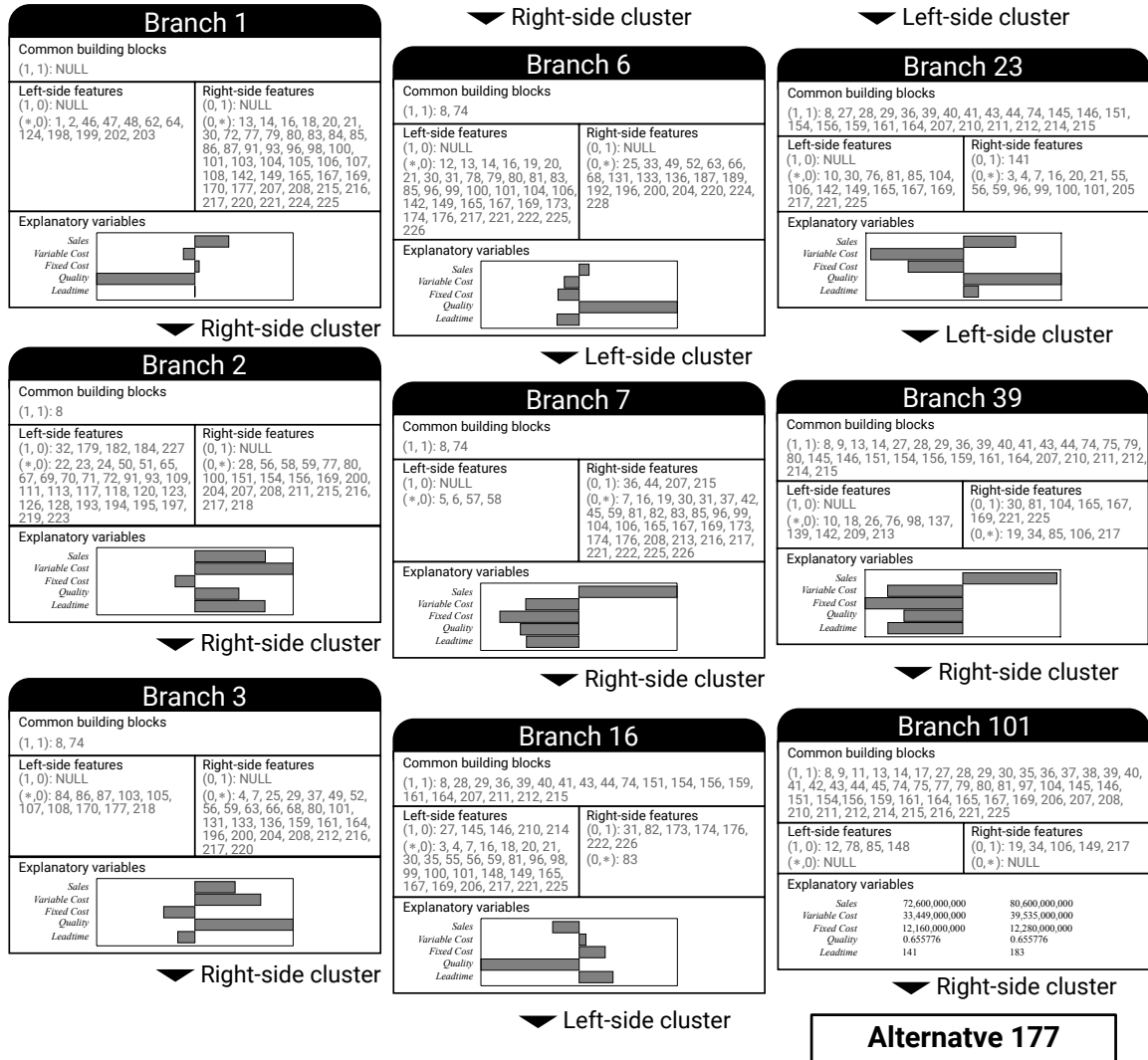


Fig. 5.14: The process leading to the Alternative 177 at decision bifurcation diagram.

ている。以上から、代替案 177 が含まれる Branch 6 の左クラスタは輸送費を抑えた構成であることが考察できた。ロジスティック回帰分析の結果からも、影響度は低いものの右クラスタを選択の方が売上を向上させることができるが、左クラスタは変動費と固定費の観点から有利であることを理解できる。Branch 6 時点で組み立て拠点は製品 P_1 、 P_2 のみ地点 S^3 の可能性が残されている。Branch 16 で品質と売上が向上する左クラスタを選択した時点で P_2 の生産拠点は S^2 に限られ、製品 P_1 の組立て拠点に関する意思決定は Branch 39 で行われている。Branch 39 の左クラスタでは製品 P_1 の組立て拠点を地点 S^1 としているものの、生産能力の制約によって市場 S^1 に供給を行っていない。売上を高めるため、左クラスタでは製品 P_1 の組立てのみを地点 S^3 で行うことによって、すべての市場への供給を可能にしている。以上から、代替案 177 を選択する経緯をビルディングブロックの集積過程として考察できることを確認できた。

最後に、グローバルプロダクトファミリの最適設計において、多目的最適化の結果を解釈する先

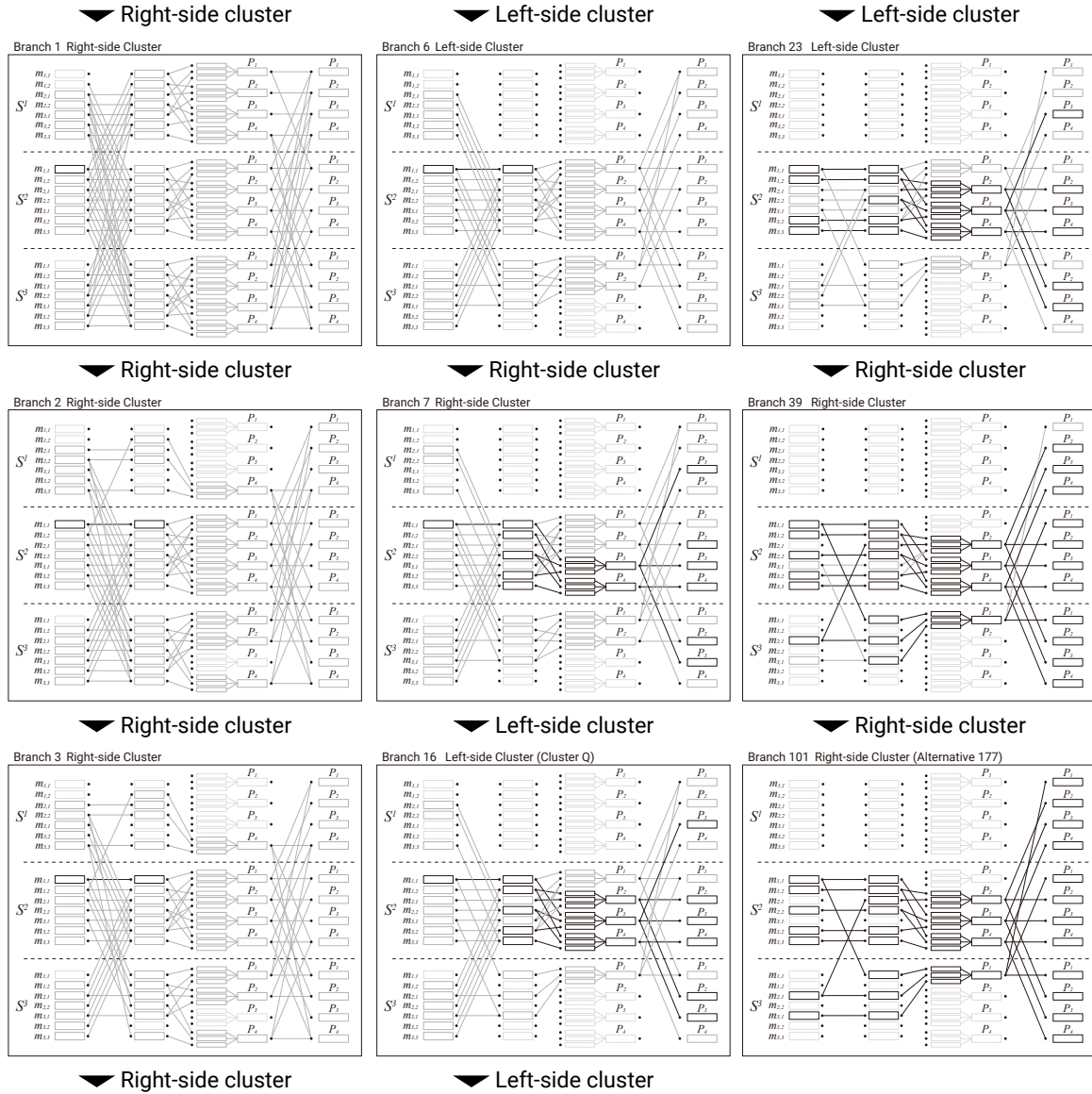


Fig. 5.15: Building block accumulation leading to the Alternative 177.

行研究^[92]と比較した提案手法との特徴について考察する。先行研究では、主成分分析を基盤にしたデータマイニング手法を応用することで、代替案の集合をいくつかのクラスタに分類する。それぞれのクラスタの中から代表的な構成を意思決定者が目で見えて選定し、全てクラスタでの目的関数の順位からクラスタの特徴を考察する。先行研究による方法と提案手法とを比較した時の、本研究の特徴と優位性は次の2点にまとめられる。

スキーマ定義による代替案集合の特徴の可読性: どちらの方法でも、クラスタの特徴を比較から検討するが、先行研究では代替案の集合から代表的な構成を人の目で見えて選定する。本手法で導入したスキーマモデルは*を用いて集団の代表的な構成を記述することができ、選定の負荷を提言させることができる。さらに、集団の特徴の違いの発見と解釈についても先行研

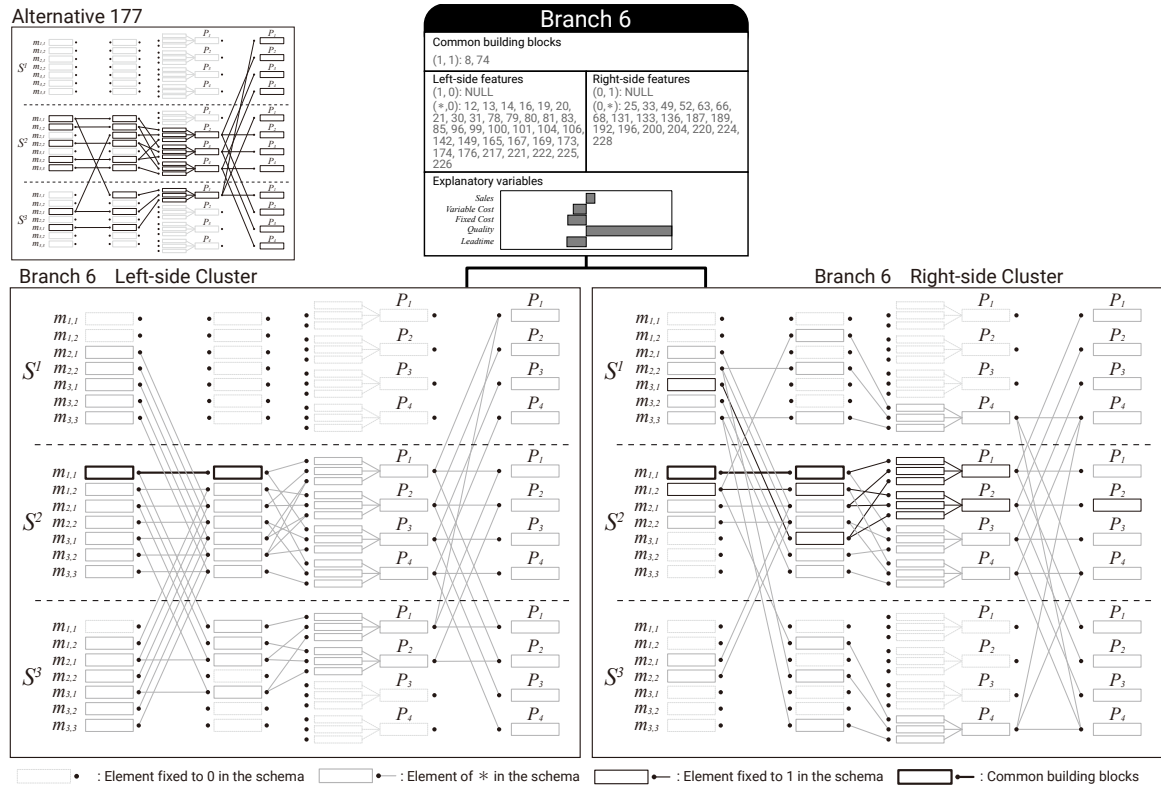


Fig. 5.16: Box in Branch 6 and schemas for left-side and right-side clusters.

究では人の手に委ねられていたが、提案手法によって数理的な手続きによるビルディングブロックの顕在化とその体系化が可能になった。多数の組合せ変数の中から注目すべき部分を意思決定者に提示することができ、論点を明確にする効果がある。

集合の関係性の量的な比較検討: 本研究では複数の評価指標に対して、ロジスティック回帰分析を応用したビルディングブロックの解釈法を提案した。先行研究の場合は、クラスターの目的関数値の平均値を代表値として扱い、評価関数での順位から定性的な解釈を行っており、多くの部分が人の手に委ねられていた。提案手法では2つの集団での考察にはなるが、複数の指標の優先度やトレードオフの関係を数理的な手続きによって明らかにすることができる。

5.6 結言

本章では、解空間に選択構造を持つ問題であるグローバルプロダクトファミリー設計問題を取り上げ、解空間構造の分析法の応用を示した。まず、グローバルプロダクトファミリーの最適設計問題を構築し、モジュール組合せによる製品系列展開とサプライチェーン設計を連携させた数理モデルを構築した。設計を表す要素としてストラテジックレベルの意思決定要素を抽象するスキーマモデルを構築し、選択構造の分析を可能にした。次に、グローバルプロダクトファミリーの性能評価指標

が多様であることを受けて、ロジスティック回帰分析に基づくビルディングブロックの解釈法を提案した。それぞれの分岐点での左右のクラスタの分類面を定義し、学習結果の重みの絶対値と符号からそれぞれの特徴を説明可能にした。最後に自動車のグローバルプロダクトファミリ設計問題を例に引いて、提案する解空間の構造分析法とロジスティック回帰分析による解釈法の妥当性と有効性を確認した。

第 6 章

モジュール化設計への適用

6.1 緒言

本章では、異なる種別の組合せ型設計問題として製品アーキテクチャ戦略の 1 つである、モジュール化設計法を取り上げ、組合せを伴う解空間の構造分析法を応用する。モジュール化問題の解空間は M 個のクラスタへの N 個のコンポーネントの最適配分を考える一種の割り当て問題であり、その解空間はネットワーク構造として定義される。本章では、モジュール化設計問題のネットワーク構造に合わせた解空間構造表現や構造分析法の考え方を論じ、解空間構造の特徴に基づく意思決定の枠組みを構築する。モジュール化設計問題は Design Structure Matrix と呼ばれる包括的なモデルを用いるため、モデルの評価指標そのものは解の選定の基準にはならず、モデル操作からの知見獲得やモデル化の前提に立ち戻った考察にこそ意味がある。そこで、解空間の構造分析結果から得られる構造の特徴に基づいて代替案の違いに解釈を与えるための、スキーマ表現に関連した新たなモデルを導入し、その分析方法を提案する。最後に、家庭用空調機を事例として取り上げ、提案手法の有効性と妥当性を確認する。

6.2 数理最適化によるモジュール化設計法とその課題

6.2.1 製品アーキテクチャとその意義

今日の製造業において、顧客ニーズの多様化に伴い製品の価格を抑えながらも多様な製品を展開することが求められてきている。従来の少品種大量生産を踏襲しながら多品種少量生産に移行すれば、さまざまな顧客ニーズに個別的に対応する形で設計を行うことになるため、設計や生産に要するコストが増加してしまう課題を抱える。そこで、個別の製品を対象として優れた製品を設計するのみならず、背景となるコンセプトを共有する一連の製品群を製品系列 (プロダクトファミリー) として俯瞰して捉え、統合的な設計を行うことが求められてきている。このような設計方法論は、製品系列設計と呼ばれている [81]。

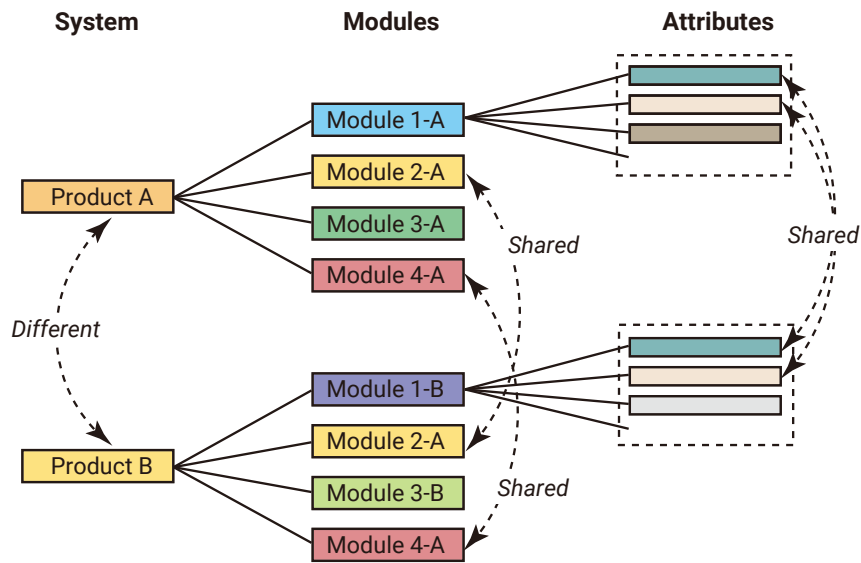


Fig. 6.1: Modular architecture.

製品系列内の関係を考える上では、製品の部分機能における構造と部分実体における構造との相互関係を整理しておくことが重要である。例えば、製品を機能や実体の構造の観点から独立した部分 (モジュール) に分けることで、共通化による間接費の削減や、モジュール組合せによる柔軟な製品提供により、コストの低減と設計の効率化を狙うことができる^[99]。このような部分機能と部分実体の関係性は製品アーキテクチャと呼ばれ、製品系列設計において重要な概念である。Ulrich^[7]は、製品アーキテクチャの設計は、(1) モジュールやコンポーネントへの階層的な分解と統合と、(2) モジュールとコンポーネントの相互関係の計画と、(3) モジュールとコンポーネントの機能の割り当ての3つの問題を包括していると考え、インテグラル型アーキテクチャとモジュラー型アーキテクチャという代表的な2タイプのアーキテクチャを示している。

モジュラー型アーキテクチャとは、ある部分機能を担うある部分実体が明確になっていて、両者の間の関係がおおよそ1対1になっているアーキテクチャである。製品の構造にモジュラーアーキテクチャを想定し、製品がシステム、モジュール、属性の3層で表されると考えれば、製品系列設計で行われる共通化は図6.1のように表される。ここでの属性を統一することの意味は、モジュールの背後にあるプロセスや部品などを共通化することに対応する。モジュール化戦略は性能効率の低下や全体サイズの増大といった欠点がある^[100]一方で、適切なアーキテクチャを導入することができれば、互いに独立性の高い部分実体を持つため、再構成可能性^[101]や、組み立て容易性^[102]、多様な顧客要求への柔軟な対応が可能になる^[103]など、さまざまな利点が期待できる。

本章においては、製品アーキテクチャ設計としてモジュール化設計を扱うこととし、ここでのモジュール化とはモジュラー型アーキテクチャをとることを指す。

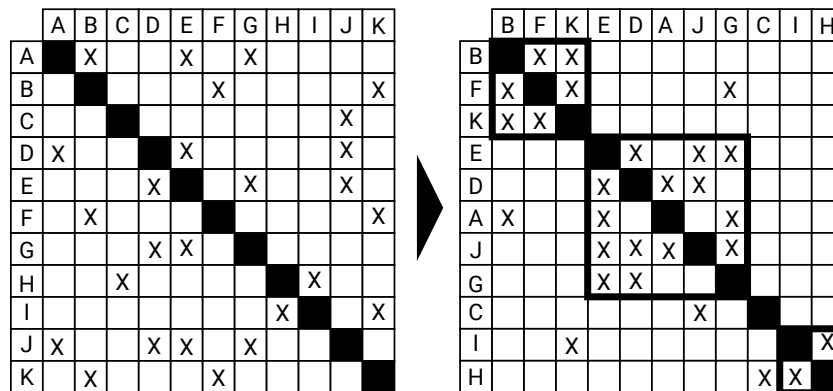


Fig. 6.2: Example of clustering result.

6.2.2 Design Structure Matrix を用いたモジュール化設計法

複雑なシステムのアーキテクチャを設計することとは、小さい要素にシステムの構造を分解し、システム全体としてより良い性能を発揮するためにその相互作用を計画することである。従来のモジュール化設計法では製品をシステムとして捉え、構成する部分機能や部分実体をシステムのコンポーネントとみなし、それらの関係性を包括的に抽象したモデルを構築することでその合理性を担保してきた。代表的な手法は、デザイン・ストラクチャ・マトリクス (Design Structure Matrix, DSM) を用いた方法論である^[14]。

DSM は、システムを構成する要素とそれらの相互作用を表すために用いられるネットワークモデリングツールである。マトリクスのノードにコンポーネントを対応させ、コンポーネント間の依存関係を記述する。行列の順序を入れ替える操作 (パーティショニング) でモデルを整理することで、さまざまな洞察を得ることが期待され、広い分野で応用されている。特に、製品アーキテクチャ設計で用いられる DSM は製品アーキテクチャ DSM と呼ばれ、産業界でもさまざまな対象に用いられてきた^[104, 105]。パーティショニングの基本は人の手によるのだが、近年は数値最適化技術の発展に伴って、メタヒューリスティクスを応用したアルゴリズムが提案されてきており、DSM を評価する指標がさまざまに提案されている^[106]。

製品アーキテクチャ DSM を用いてモジュール化設計を行う場合、クラスタリングと呼ばれるパーティショニングを行う。クラスタリングとは DSM の行および列を再整理することで要素をグループ化する分析方法である。クラスタリングを行う際には、クラスタの大きさ (クラスタ内の要素数) やクラスタの数を考慮しながら、クラスタの外側にある依存関係の数・大きさをなるべく小さくするようにグループを生成する。クラスタリングの例は図 6.2 に示される。図 6.2 の DSM は 11 のコンポーネントで構成されるシステムであり、クラスタリングを行うことで、3 つのクラスタにグループ分けされている。クラスタリングによって作成されたグループは、内部の依存関係は密にしつつも外部の依存関係は疎にする方針のもと生成されているため、製品アーキテクチャのモ

ジュールに対応する。

DSM に代表される包括モデルによるモジュール化の手法^[107]の利点は、(1) 比較的多くのコンポーネントとその関係をもつ複雑システムを簡素なモデルに落とし込んで考えることで俯瞰的な洞察を得ることができることと、(2) 分析を通してモジュラリティに影響するようなコンポーネントの重要な部分や相互関係のパターンを明らかにすることで判断の介入点を明確にすることにある。製品が大規模になり、部品点数の増加からそれらの関係性も複雑になるにしたがって、個人ではその全貌を把握することができないため、製品アーキテクチャ DSM の価値は高い。また、すべての情報をひとまとまりにした大規模な DSM を作成したとしても、認知的負荷の高さからその操作は容易ではなく、数学的なモデル評価指標とそれに基づくパーティショニングアルゴリズムは広く用いられている。

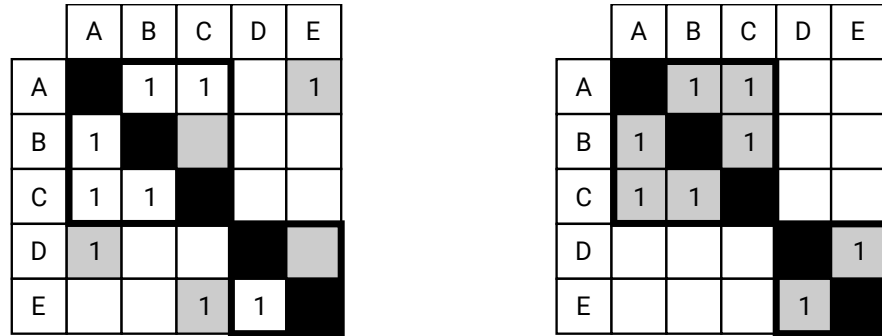
6.2.3 最適設計問題としてのモジュール化

クラスタリングは本質的には M 個のクラスタへの N 個のコンポーネントの最適配分を考える一種の割り当て問題である^[83]。DSM の分析の基本は、スプレッドシートの適用による行と列の手動操作であった。しかし、DSM が大きくなるにつれ手作業によって大域的な最適解に導くことは非常に難しく、コンピュータを利用した自動クラスタ法が利用されている。数理最適化を用いるにはモジュール性能を評価する必要がある、先行研究では、さまざまな評価方法や探索アルゴリズムが提案されている^[108, 109]。モジュールは他から独立した部分として定義されているが、モジュール性能の評価指標はアプローチによって多様である^[104, 110, 111, 112]。多くの研究ではモジュール内部の密な相互作用とモジュール間の緩い相互作用を表す 2 つの指標を用いてモジュール性を評価している^[106]。

Yu ら^[111]は最小記述長 (Minimal Discription Length, MDL) の考えに基づくモジュール性能評価指標を提案している。評価指標は式 (6.1) に表される。

$$f = (1 - \alpha - \beta)(c \log N + \log N \sum_{i=1}^c n_i) + \alpha[|S_1|(2 \log N + 1)] + \beta[|S_2|(2 \log N + 1)] \quad (6.1)$$

ここで、 c は DSM 全体のクラスターの数、 N は DSM の要素数、 n_i は i 番目のクラスタに含まれる要素の数であり、 α と β は設計者が決定する 0 から 1 の設計パラメータである。この評価指標は第 1 項に表される MDL 指標と第 2 項と第 3 項の 2 つのミスマッチ指標の組合せで構成されており、それらを最小化することで望ましいクラスタを生成することができる。MDL 指標は、モデルをできるだけ簡素な形で表現することを目指して式が設計されている。具体的には、第 1 項は DSM のクラスタの数とその要素の数に依存する形で表現されており、この項を最小化することでクラスタを横断する要素をできるだけ排除した上で、モジュールの大きさを小さくすることができる。第 2 項と第 3 項は 2 つのタイプのミスマッチに基づいて設定されている。それぞれの算出方法について、例を用いて説明する。図 6.3 は実際の DSM モデルと、クラスタリング結果から逆算さ



(a) DSM in which the clustering operation was performed: given data set. (b) DSM created based on the model description.

Fig. 6.3: Mismatch between calculation results and model description.

Tab. 6.1: Mismatched data description^[111]

	Hypothesized marks that do not exist in the real data set (Type I mismatch)	Real marks that the hypothesized model misses (Type II mismatch)
Inside clusters	(1) Row B, Column C (2) Row D, Column E	ϕ (always)
Outside clusters	ϕ (always)	(1) Row A, Column E (2) Row D, Column A (3) Row E, Column C

れる理想的な DSM モデルである。図 6.3(b) は、クラスタ内の依存関係が最も密でありながら、クラスタ間は完全に独立している理想状態を示している。ある計算結果として図 6.3(a) が導かれた場合、その設計情報は「クラスター 1 は {A, B, C} の 3 つの要素で構成され、クラスター 2 は {D, E} の 2 つの要素で構成される」というものである。この情報を伝達された側は図 6.3(b) のような状態を想像してしまうため、正しいデータを伝達するためには表 6.1 のような表を用いて整理を行う必要がある。これらのミスマッチの数をできるだけ減らすことが第 2 項と 3 項の目的である。目的関数の中では、ミスマッチのタイプ 1 はモジュール内に存在しない依存関係の数、タイプ 2 はモジュール間の依存関係の数を表し、それぞれ $|S_1|$ 、 $|S_2|$ で表される。

上述のように、式 (6.1) などを用いて DSM のモジュール性能を一度評価することができれば、最適化アルゴリズムを適用することで最適解を導くことが可能になる。具体的な方法としては遺伝的アルゴリズム^[111] や模擬焼きなまし法^[105] が挙げられる。

6.2.4 モジュール化設計における競合と選定の難しさ

Yu ら^[111]の方法に代表される DSM のクラスタリングの目的関数の多くは、設計者が設定するクラスタのサイズを調整するパラメータが含まれていて、DSM のクラスタリングには次のようなクラスタのサイズに関する 2 つの矛盾するトレードオフ関係が存在している。クラスタの外側にある依存関係を減らすことは、関係のある要素をクラスタの中に取り込んでいくことに相当するため、最小化を推し進めることはクラスタのサイズを大きくすることに直結する。これは式 (6.1) でも同様の解釈が可能である。クラスタのサイズを大きくすることはミスマッチタイプ 2 を小さくする一方で、内部の依存関係を疎にすることに繋がるため、ミスマッチタイプ 1 が大きくなってしまいう上に、MDL の指標も悪化する。したがって、クラスタリングにはクラスタの外側にある依存関係の数や強さの最小化とクラスタのサイズの最小化とがトレードオフの関係にある。

式 (6.1) のように、多くの研究で用いられているモジュール性能評価指標はトレードオフ関係にある複数指標の線形加重和法で定式化されているため、最適化結果に多くの多様性や競合関係、等価解の存在を見出すことができる。ここでは、具体的なクラスタリング結果の例を挙げてその特徴を指摘する。図 6.4 は 4 つの DSM 分析の結果である。まず、図 6.4(a) と図 6.4(b) の結果に焦点を当てる。(a) のミスマッチタイプ 1 は 4 である一方で、(b) のミスマッチタイプ 2 は 4 となっていて、これらは 2 つのミスマッチのトレードオフ関係から導かれた競合解であると考えられる。しかし、定式化の段階でミスマッチの重みを等しく設定した場合は、これらを等価解として導くことになり、目的関数設定次第ではこの競合関係を区別することができない。また、重みを変えていたとしても、図 6.4(c) と図 6.4(d) はクラスタリング結果は 2 つのミスマッチの値がどちらも等しく、等価解を容易に想定することができる。したがって、モジュール化の最適設計問題には多くの等価解の存在が示唆されており、解の多様性が高い問題であることがわかる。

上述のように数理最適化を利用した自動クラスタリング法が用いられている一方で、その最適化結果は設定によってさまざまであり、代替案の解釈と選定には設計者の勘と経験に頼らざるを得ない問題点を抱えている。DSM を用いたモジュール化設計には、6.2.4 項で述べたようなトレードオフの関係が潜んでおり、多目的の設計問題であることが伺える。つまり、2.3.2 項で論じてきたように、最適化による数学的操作からは直ちに最適なモジュール構成を定めることができず、他の指針が必要になる。さらに、DSM のクラスタリング操作は本質的には M 個のクラスタへの N 個のコンポーネントの最適配分を考える組合せ最適化問題であることから、最適化結果を織りなす組合せ構造は複雑であり、単純な比較検討からは代替案の傾向を理解することができない特性を持っている。

特に、DSM を用いたモジュール化設計に従来の目的関数に基づく代替案の分析法を用いる場合は、次のような課題に直面する。

	A	B	C	D	E	F	G
A		1	1	1			
B	1			1			
C	1			1			
D			1				
E						1	1
F					1		1
G					1	1	

(a) Clustering results: (Type I mismatch, Type II mismatch) = (4,0)

	A	B	C	D	E	F	G
A		1	1	1			
B	1			1			
C	1			1			
D			1				
E						1	1
F					1		1
G					1	1	

(b) Clustering results: (Type I mismatch, Type II mismatch) = (0,4)

	A	B	C	D	E	F	G
A		1	1	1			
B	1			1			
C	1			1			
D			1				
E						1	1
F					1		1
G					1	1	

(c) Clustering results: (Type I mismatch, Type II mismatch) = (4,4)

	A	B	C	D	E	F	G
A		1	1	1			
B	1			1			
C	1			1			
D			1				
E						1	1
F					1		1
G					1	1	

(d) Clustering results: (Type I mismatch, Type II mismatch) = (4,4)

Fig. 6.4: Equivalent solutions of clustering.

- DSM のクラスタリング結果には多数の等価解が存在しているので、評価関数からはそれらの違いを認識することができない。具体的な事例は図 6.4 が示している。
- そもそも DSM は多量の情報を一括して取り扱う包括的なモデルであるため、その評価指標そのものが抽象的である場合は少なくない。MDL についても 2 つのミスマッチの値そのものに設計上重要な情報は含まれておらず、それぞれのミスマッチが何を抽象しているのかに立ち戻った考察が必要である。つまり、目的関数そのものが選定の基準として適していないため、目的関数に基づく方法と馴染まない。

本章では、DSM のモジュール化設計問題を対象に組合せを伴う解空間の構造分析手法を適用することで代替案の解釈と選定を可能にする。本章では、上記課題の 2 点目に焦点を当てて、評価関数の値ではなく、解空間構造の特徴に基づいた解釈方法を新たに提案し、事例を通してその有効性を検証する。

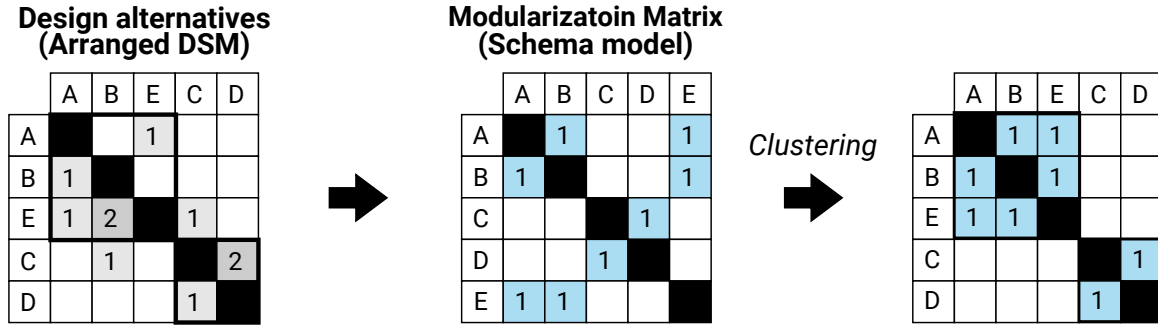


Fig. 6.5: Schema model construction using modularization matrix

6.3 ネットワーク構造問題におけるスキーマモデルの構築

本節では、本研究で提案する解空間の構造分析法を適用するため、モジュール化問題の解空間を表現するスキーマモデルを導入する。

DSM はシステムを構成する要素とそれらの相互作用を表すために用いられるネットワークモデリングツールであり、DSM の並び替え操作での設計変数とは、ノードの順序とそれらのグループである。本研究では、DSM 操作を表す最適化問題で考える解空間の構造をネットワーク構造として定義し、クラスタリングの結果を抽象するスキーマモデルを構築する。

DSM のクラスタリングは、本質的には N 個のコンポーネントを M 個のクラスタへと配分する組合せを考える割り当て問題であるが、製品アーキテクチャ設計での意思決定の観点から述べれば、「どの部品とどの部品を同じモジュールとしてまとめるか」という問いに答えることと等しい。この意思決定要素をビルディングブロックの候補として用意してスキーマモデルを定義する。スキーマモデルの作成方法を図 6.5 に示す。図 6.5 の左側のマトリクスは重み付き DSM のクラスタリング結果である。このシステムは 5 つの部品で構成されており、部品 A、部品 B、部品 E が 1 つのクラスタにまとめられ、部品 C、部品 D はもう一方のクラスタとしてまとめられており、合計 2 つのクラスタが生成されている。製品システムを構成するそれぞれの部品に着目して判断を書き下せば、部品 A は部品 B と部品 E と同じクラスタに割り当てる判断を行っている。これらの情報を DSM の依存関係として記述したマトリクスが図 6.5 の中央に示されている。本研究ではこのスキーマモデルを Modularization Matrix と呼称する^[113]。

Modularization Matrix の i 行 j 列の要素 $\delta_{i,j}$ は、以下のような性質を持つ。

$$\delta_{i,j} = \delta_{j,i} \quad (6.2)$$

$$\delta_{i,j} = 1, \quad \delta_{i,k} = 1 \rightarrow \delta_{j,k} = 1 \quad (6.3)$$

式 (6.2) は対称性に関する性質である。割り当て問題においてはその関係性に方向性を定義することができない。したがって、それぞれ双方向の関係性として依存関係を記述するため、クラスタマ

トリクスは対称になる。式 (6.3) は連続性に関する性質である。グループにまとめる操作において、部品 A と部品 B、部品 A と部品 E が同じクラスタにしながら部品 B と部品 E を別のクラスタに割り当てることはできない。図 6.5 の右側のマトリクスは、Modularization Matrix をクラスタリングした結果である。これは本来の最適化結果と同様の結果であることに注意されたい。

本節で提案したスキーマモデルへの写像 f について、3.4.2 項で述べた 3 つの条件との関係は次のようにまとめられる。

- Modularization Matrix はモジュール化結果の理想状態である。スキーマモデルの要素と元々の最適化結果は 1 対 1 の関係にあり、写像 f は単射である。
- 本スキーマモデルで注目するビルディングブロックの候補は「どの部品とどの部品を同じモジュールとしてまとめるか」という問いそのものであり、設計上の判断に対応している。
- スキーマとして集団を表現する場合には、これまでのスキーマモデルと同様に * を用いて表現することができるが、平均ベクトルを計算すれば、重み付き DSM として取り扱うことが可能になる。Modularization Matrix をクラスタリングすることで元の最適化結果が再現できるので、集団を表す重み付き DSM をクラスタリングすれば、集団の代表的な構成を知ることができる。本研究では集団を表現するスキーマである重み付き DSM を個別のマトリクスと区別して、Averaged Modularization Matrix と呼称する。多様な DSM のクラスタリング結果に横串を通して分析を行う際には、この性質を利用することで見通しが立ちやすくなる。

Modularization Matrix は 6.2.3 項で取り上げたモジュール性能評価指標である、MDL の考え方を基調にしている。Modularization Matrix は図 6.3(b) に対応し、クラスタリングの設計情報として伝達される情報そのものであると解釈することができる。Yu ら^[111] はその際に 2 種類のミスマッチに着目し、表の形式で整理することを推奨しており、これはクラスタマトリクスの特性を理解する上では重要な視点である。Modularization Matrix はどのコンポーネントをまとめるかという本質的な意思決定を抽象し、忠実に反映している一方で、2 つのミスマッチに関する情報を捨象していることに注意しなければならない。つまり、ミスマッチが多い Modularization Matrix は本来の製品システムの相互作用の関係とのギャップが広がることを意味する。

6.4 解空間構造の特徴に基づくビルディングブロックの解釈

6.4.1 解空間構造の特徴とその分析方法

6.3 節で導入した代替案の構造を表す Modularization Matrix や、代替案の集団を表す Averaged Modularization Matrix を比較することで、構造分析結果であるデンドログラムの分岐理由を考察する。これまでの事例では評価指標を用いてグループの特徴を説明していたが、製品アーキテク

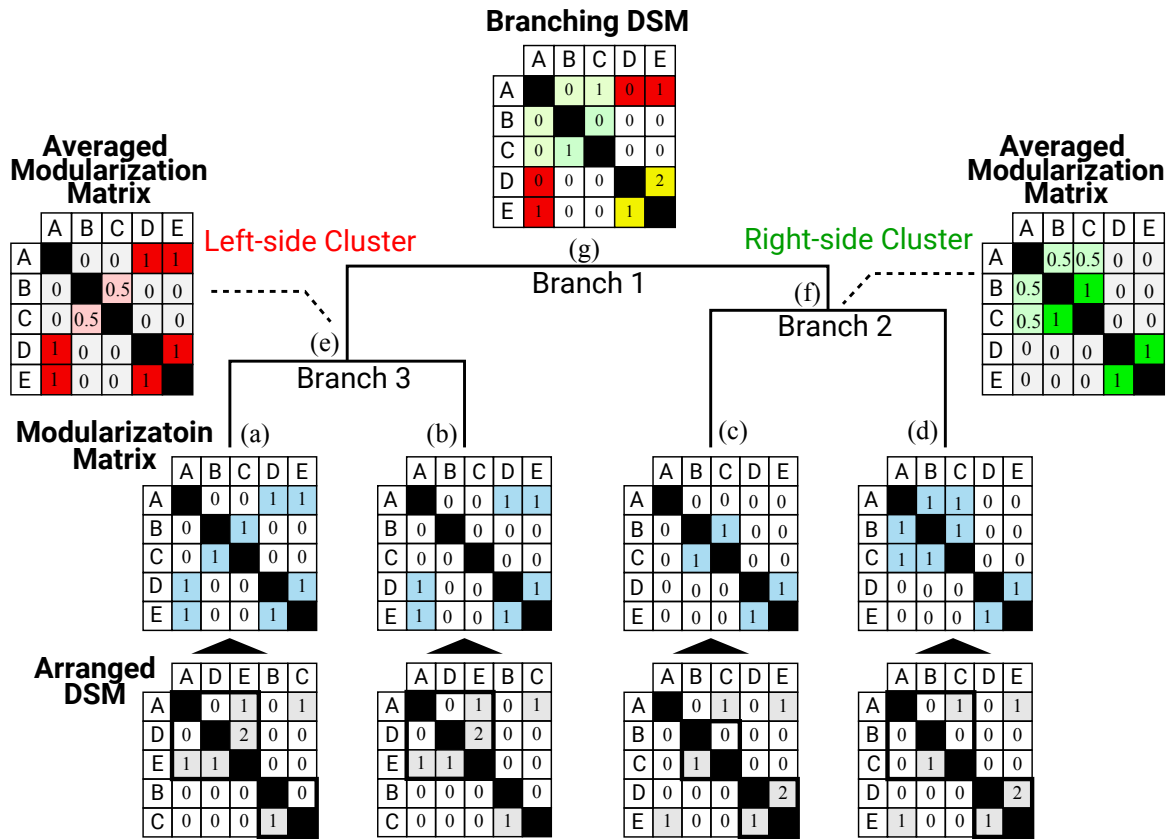


Fig. 6.6: An example of the dendrogram and matrices of respective nodes.

チャ設計では目的関数に基づく分岐理由の考察が馴染まないことを受け、本節では解空間の構造の特徴に注目した分岐理由の解釈法を提案する。

解空間の構造をより詳細に理解するため、Branching DSM と呼ばれる、2つのクラスタの差分の情報を取り入れた新しい DSM モデルを新しく導入する。それぞれの分岐点毎に Branching DSM を作成し、モデル操作を通して解釈を行う。Branching DSM の依存関係には分析対象の DSM と全く同じ値を用いる。デンドログラムで二分される2つのスキーマに注目し、それぞれの Averaged Modularization Matrix の値の組合せに応じて配色することによって、解空間構造の特徴を視覚的に重ね合わせて分析できるようにする。6.4.2 項では具体例を挙げながら Branching DSM の作成方法について述べる。6.4.3 項では Branching DSM を用いて解釈を行うための代表的な構成をテンプレートを整理して解釈の指針を示し、意思決定分岐図を用いた体系化について述べる。

6.4.2 Branching DSM を用いた分岐理由の考察

本項では、まず Branching DSM の作成方法について、例を挙げながら説明する。ある分岐点で分割された2つのクラスタを表す Averaged Modularization Matrix とそれらの違いを表す Branching DSM は図 6.6 にまとめらる。図 6.6 のデンドログラムでは、Branch 1 は、代替案

		Right-side Cluster		
		0	*	1
Left-side Cluster	0	Category A'	Category D	Category B
	*	Category D	Category E	Category C
	1	Category B	Category C	Category A

Fig. 6.7: Five cell categories of elements of the branching matrix.

(a) と (b) からなる左側のクラスタと、(c) と (d) からなる右側のクラスタに分割し、Branch 3 は (a) と (b) に分割、Branch 2 は (c) と (d) に分割する。図 6.6 の最下端に位置する (a)、(b)、(c)、(d) は DSM のクラスタリング結果と対応する Modularization Matrix である。それらの平均を取ることで Averaged Modularization Matrix が生成される。図 6.6 の (e)、(f) はそれぞれ Branch 3、Branch 2 をまとめるクラスタの Averaged Modularization Matrix である。Averaged Modularization Matrix の 0 あるいは 1 の要素はスキーマの固定部分であり、全ての代替案での共通要素である。図 6.6(g) が Branch 1 の分岐理由を考察する Branching DSM である。左右のクラスタの違いを考察するにあたって、モジュール構成のミスマッチやスキーマモデルの定義に立ち戻った考察が行えるように、Branch DSM には分析元の DSM モデルの依存関係を転写する。さらに、階層的クラスタリングに基づく構造分析の結果に関する情報は、3.6.1 項で導入したビット要素の組合せに応じたカテゴリに即して配色することで、視覚的に理解できるようにする。本章では、これまで Category C として扱ってきた $(1,*)$ と $(0,*)$ を区別し、新しくスキーマの固定方法の Category を設定する。具体的には、 $(1,*)$ と $(*,1)$ の組を Category C、 $(0,*)$ と $(*,0)$ の組を Category D、 $(*,*)$ を Category E とする。これらに応じた色のパターンは、図 6.7 に示される。

意思決定者は Branching DSM を操作することで解釈を行い、設計知識を獲得する。構造分析の結果である図 6.7 による配色のパターンから、解空間構造の重要な特徴を明らかにする。本項では、いくつかの分岐の典型的な配色パターンを示し、解釈のための指針を示す。Branching DSM 分岐のパターンを図 6.8 に示す。

図 6.8 は 4 つの部品で構成される製品の分岐マトリクスである。ある分岐 Branch n で分割される 2 つのクラスタについて、図 6.7 に基づいて、左を赤色、右を緑色でハイライトしている。マトリクスの下には、簡単なモジュールの模式図が示されている。分岐での構造の変化パターンには次の 2 つに大別される。

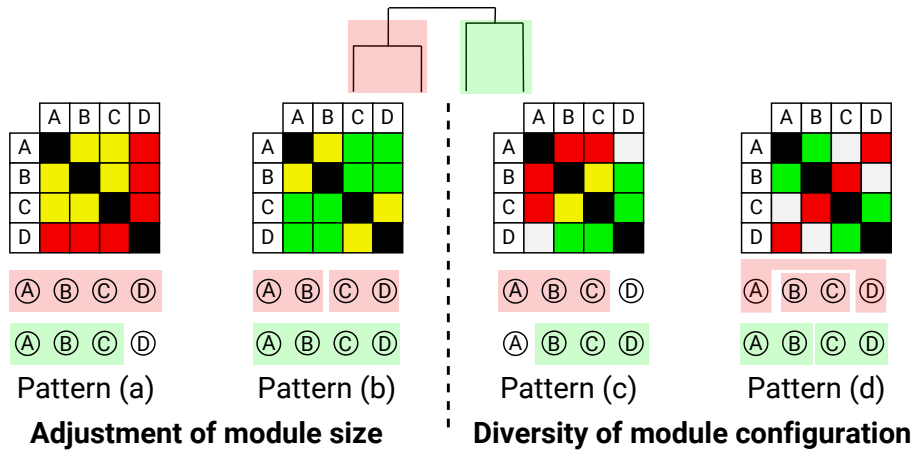


Fig. 6.8: Interpretation of colored patterns of branching DSM.

モジュールサイズに関する分岐: モジュールのサイズを大きくする戦略として、モジュールに属していない部品を取り込む方法と、小さなモジュールを統合する方法と2つの戦略がある。モジュールサイズはクラスタリングの最も基本的なトレードオフ関係であり、左右のクラスターで方針がことなる場合がある。

- (a) 図 6.8(a) は、モジュール拡大に関する意思決定である。モジュール拡大に関する意思決定: 左右どちらのクラスターにおいても、部品 A、部品 B、部品 C は同じモジュールとして扱っている一方、部品 D の取り扱いが異なる。左のクラスターでは他の部品と同じモジュールに追加している一方、右のクラスターでは単体の部品として取り扱っている。つまり、モジュールを拡大するならば左の分岐を辿ることになる。このパターンはクラスターサイズに関するトレードオフを操作しているとみなすことができる。
- (b) 図 6.8(b) は、モジュール統合に関する意思決定である。部品 A と部品 B、部品 C と部品 D について、どちらも同じモジュールとして扱っている。右のクラスターではそれらのモジュールを一体にして大きなモジュールを構築している一方で、左のクラスターでは別々のモジュールとして取り扱っている。モジュール統合の可能性はモジュールの総数が増える可能性がある。

モジュール構成に関する分岐: モジュールの拡張や構成部品の組み合わせによって、モジュールの構成が変化する。モジュール化の方針の違いは、モジュールそのものが持つ機能や役割が異なるため、モジュール化結果を左右するレバレッジポイントとして考えることができる。

- (c) 図 6.8(c) は、モジュール拡大の方針の違いに関する意思決定である。どちらのクラスターでも共通して部品 B と部品 C を同じモジュールとして扱っているが、そのほかの構成が異なる。左のクラスターでは部品 A と組にしている一方で、右のクラスターでは部品 D と組にしている。
- (d) 図 6.8(d) も同様にモジュールの構成が異なる意思決定である。マトリクス上で空間的

に交差しているように、左のマトリクスと右のマトリクスで4つの部品の取り扱いが全く異なることを意味している。

これらの4つのパターンは最小表現であり、実際の大規模な分岐マトリクスでは、それぞれのパターンの大きさが異なっていたり、それらが複合的に発生したりしていることがある。複合的に発生しているパターンを読み解くには、分岐マトリクスのノードの順番を動的に変化させ、パターンを発見しながら分析を行う操作が必須になる。

構造分析の結果として共通ビルディングブロックとして明らかにされた Category A の部分やレバレッジポイントとして明らかにされた Category B、それに関連する Category C と D の部分に焦点を当てながら、図 6.8 に示される分岐の構造的な特徴をもとに、解空間の分割過程に解釈を与えていく。また、DSM は包括的なモデルであるので、Branching DSM だけで完結せず、明らかになったビルディングブロックやレバレッジポイントは、元々の DSM モデルでのモデル化の過程に立ち戻ってその意味について考察を行うことになる。

6.4.3 意思決定分岐図を用いた解空間構造の特徴の管理

解空間の構造分析法では、スキーマの共通点や相違点をビルディングブロックやレバレッジポイントとして構造の違いとして把握した上で、関連する意思決定者の解釈や嗜好の情報を合わせて設計知見として扱い、意思決定分岐図で管理する。図 6.9 は本章で用いる意思決定分岐図のテンプレートである。モジュール化設計問題ではそれぞれの分岐の理由を構造から判断するため、上述の要素の色分けと配色パターンに基づいて、分岐の様子とその理由の考察結果を記載する。図 6.9 の例では、5つの部品で構成される Branching DSM を用いて、Branch 1 の分析と解釈を行なっている。左側のクラスタ、つまり Branch 3 では、部品 1,2,3,4 は同じモジュールに属している一方、右側のクラスタ、つまり Branch 2 では、部品 1 と 2、および部品 3 と 4 がそれぞれ同じモジュールに含まれているが、それらは独立している。これらは明確な違いであり、Category B に分類されるレバレッジポイントである。さらに、右側のクラスタの一部の結果では、部品 5 が部品 3、4 と同じモジュールに含まれており、Category B として配色されている。この Branching DSM からパターン (a) と (b) の2つの配色パターンを読み取ることができ、分岐理由が記載されている。なお、中括弧は同一モジュールに割り当てられた部品を示し、右クラスタと左クラスタのモジュール構成はスラッシュで区別され、意思決定分岐図に記されている。

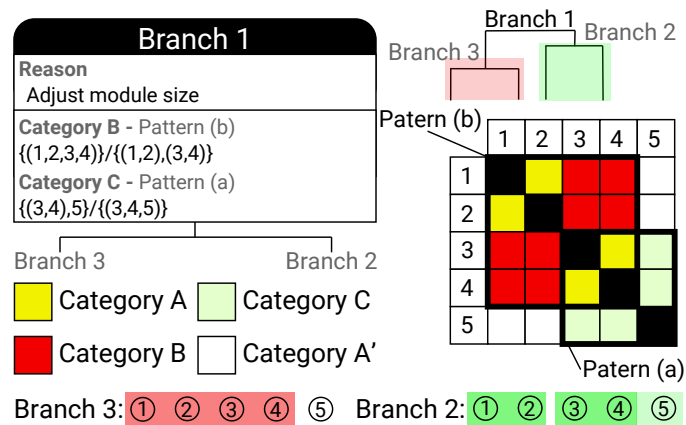


Fig. 6.9: Correspondence between decision bifurcation diagram, branching matrix and dendrogram.

6.5 事例：空調機のモジュール化設計

6.5.1 概要

本節では、家庭用空調機を例に引き、構築した手法を適用することでその有効性と結果の妥当性を考察する。

家庭用空調機は設置する部屋の面積や地域性に応じた製品展開が行われており、さらに空調性能や付加機能の有無などの機能性に沿って多種の製品ラインアップを有する多種展開品である。気候の変動によって需要が短期間で大きく変動する特性を持っている上に、その製品ラインアップの多さやサイズの大きさから完成品の在庫管理に際するコストが大きいため、適切な製品アーキテクチャを設計し、多種展開におけるモジュール性との整合性を図ることによって顧客要求に対する柔軟性を適切に確保することが重要である。空調機は主に室内機と室外機で構成され、それらは冷媒輸送のためのチューブで接続されている。物理的な構造に注目すれば室内外を分けた大きいモジュール構成を想定できるが、機能的な構造に焦点を当てればそれらを横断するようなモジュールを想定することもできる。本研究では家庭用空調機を部品レベルに分解し、50のコンポーネントで構成されるシステムとして取り扱う。

まず、本事例で取り扱う家庭用空調機の製品アーキテクチャ DSM を定義し、提案する構造分析法を適用する。式 (6.1) に潜む 2 つのミスマッチのトレードオフに注目して、遺伝的アルゴリズムを用いて有望かつ多様な解を生成する。デンドログラムのうち主要な分岐点を取り上げて、簡略化した意思決定分岐図を作成する。6.3 で提案したスキーマモデルと Branching DSM を用いた考察が有効に働くことを検証するため、作成された意思決定分岐図の妥当性やその効果を考察する。

ID	Components	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	Outdoor heat exchanger	■	1	1					1						1	1	1			1	1					2	1																								
2	Outdoor heat exchanger thermistor	1	■																																																
3	Expansion valve	1		■	1						1																1																								
4	Compressor			1	■	2	2								1	1	1										1																								
5	Suction pipe				2	■				2																																									
6	Discharge pipe				2		■	1	2																																										
7	Discharge pipe thermistor							1	■																																										
8	Four-way valve	1			2	2			■	1																		1																							
9	Three-way valve								1	■		2											1																												
10	Two-way valve			1							2												1																												
11	Liquid pipe											2	■											1									2																		
12	Gas pipe									2			■											1									2																		
13	Outdoor air temperature thermistor																										1																								
14	Partition plate	1			1													1										1																							
15	Anti-vibration rubber				1													1																																	
16	Outside chassis	1		1														1	1	■		1				1	1		1																						
17	Motor mounting plate	1																	1	■			1																												
18	Propeller fan motor																		1		1								1																						
19	Propeller fan	1																		2	■																														
20	Left side cabinet	1																								1	1		1																						
21	Right side cabinet	1									1	1																																							
22	Valve cover												1	1																																					
23	Front cabinet																																																		
24	Fan guard																																																		
25	Fin guard	1																																																	
26	Outdoor electrical equipment	1	1				1								1	1																																			
27	Reactor																																																		
28	Top cabinet																																																		
29	Indoor heat exchanger												2	2																																					
30	Indoor heat exchanger thermistor																																																		
31	Cross-flow fan																																																		
32	Cross-flow fan motor																																																		
33	Drain pan and body																																			</															

Fig. 6.10: Design Structure Matrix of home air-conditioner.

6.5.2 問題設定と計算条件

本事例で用いる普及機の重み付き DSM は図 6.10 に示される。DSM の 1 列目はコンポーネントの番号、2 列目は名称を表している。DSM には、部品間の物理的な接続関係、情報の流れ、エネルギーの流れ、空気の流れ、冷媒の流れ、電気信号の流れを記述した。複数の関係を持つ場合は重み付けをすることで非対称のマトリクスが構成されている。

本節では、式 (6.1) に示される MDL をモジュール性能評価に用いて、モジュールの大きさと独立性のトレードオフ関係に焦点を当てる。遺伝的アルゴリズムによるクラスタリングを行うことで

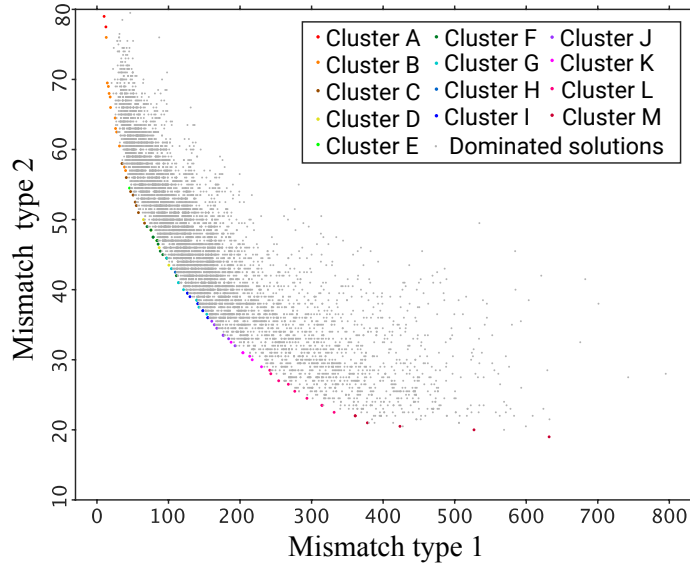


Fig. 6.11: Scatter plot of optimization results.

競合する解の集合を獲得する。DSM のクラスタリング操作は次のように定式化される。

$$\begin{aligned}
 &\text{find} && x_{ij} \quad (i = 1, \dots, c, j = 1, \dots, N) \\
 &\text{minimize} && f = (1 - \alpha - \beta)(c \log N + \log N \sum_{i=1}^c n_i) + \alpha[|S_1|(2 \log N + 1)] + \beta[|S_2|(2 \log N + 1)] \\
 &\text{subject to} && \sum_{i=1}^c x_{ij} = 1 \quad (i = 1, \dots, N) \\
 &&& x_{ij} \in \{1, 0\}
 \end{aligned} \tag{6.4}$$

ここで、 c は DSM 全体のクラスタの数、 N は DSM の要素数、 n_i は i 番目のクラスタに含まれる要素の数であり、 $x_{ij} = 1$ とはコンポーネント j を i 番目のクラスタに割り当てることを意味する。

6.5.3 解空間の構造分析結果

提案手法の 3 つのステップに従って解空間の構造分析を行った。

まず、Step 1 で最適化を実行し、有望かつ多様な代替案の集合を獲得した。モジュールの大きさと独立性に関するトレードオフを操作するため、式 (6.1) の 2 つのパラメータ α 、 β をさまざまに変化させながら複数回最適化を行うことで多様かつ有望な代替案を生成した。合計 8250 個の最適解から同一の結果を削除して、6756 個の設計候補を作成した。図 6.11 最適解の散布図である。横軸はミスマッチタイプ 1、縦軸はミスマッチタイプ 2 を表している。両者の値が小さいほど、目的関数が優れている。6750 の結果から、2 つのミスマッチのトレードオフ関係に着目して、87 件のパレート解を代替案として抽出した。図 6.11 において、色のついた点がパレート解、灰色の点が

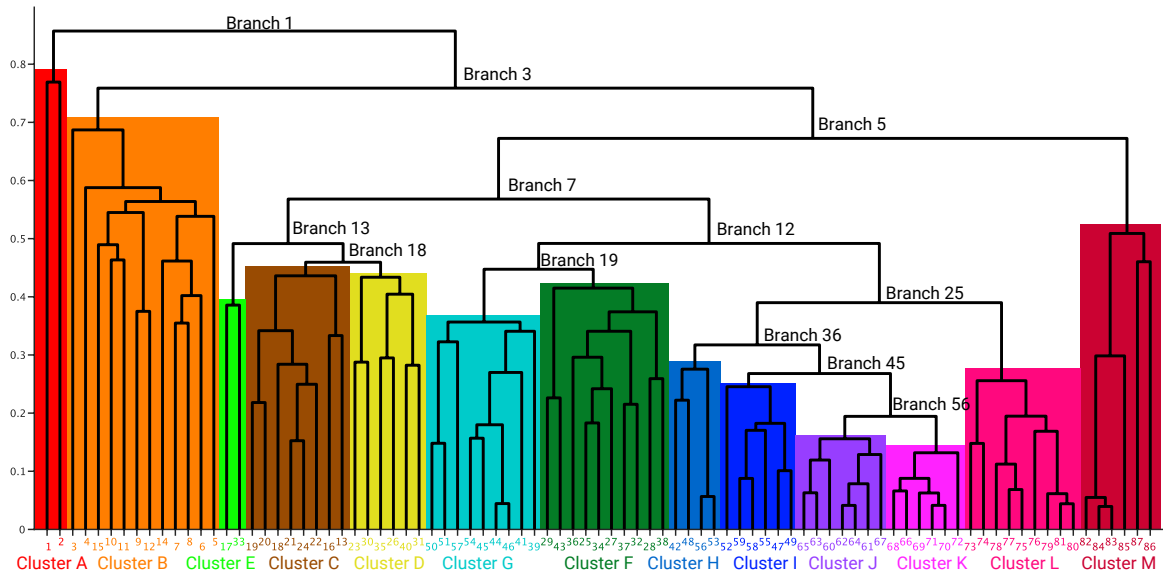


Fig. 6.12: Dendrogram.

非パレート解を表している。代替案全てについてミスマッチタイプ 1 の値の昇順にラベルをつけ、Modularization Matrix を作成することで解空間の構造を定義した。

Step 2 では、Modularization Matrix を用いて階層的クラスタリングを行った。Modularization Matrix の各要素はバイナリ情報であり、スキーマはビット列を用いて定義されている。本事例では、式 (6.2) に示される対称性を加味して、本章では表 3.2 にまとめられた類似性指標 s_{st} から S_2 を用いる。本事例では、スキーマ集積による構造の変遷を詳しく確認するため、リンケージ手法として郡平均法を採用した。階層的クラスタリングを実行した結果を図 6.12 に示す。デンドログラムの最下点は代替案であり、それぞれのラベル番号を記載している。

Step 3 では構造分析から意思決定支援を行うため、意思決定分岐図を作成した。提案手法の手順に従えば、全ての分岐点で構造分析を行うことができる。本事例では、モジュール構成を左右する有効なレバレッジポイントを特定できるかを確認するために、主要な分岐点に焦点を当てて簡略化した意思決定分岐図を作成することにする。デンドログラムの分岐点をトップダウンに辿りながら、より多くの個体が含まれている分岐点へと焦点を移しながら順にクラスタを分割し、全部で 13 のクラスタを定義した。それぞれのクラスタに属する代替案は図 6.11 上で確認することができる。

作成した意思決定分岐図を図 6.13 に示す。Branching DSM に基づいて、12 の分岐点を分析し、その結果に基づいて全てのボックスに解釈を記入した。この事例では、室外熱交換器 (1)、室外シャーシ (16)、左側キャビネット (20)、右側キャビネット (21) などの、室外機に関する構成要素が、上部に現れる重要な構成要素を構成している。その主な理由は、室外機が複雑であり、その構成部品が密に相互作用するためであると考えられる。一方で、室内機は構造が比較的単純であり、その構成部品のモジュール構成の違いは下段に表れている。

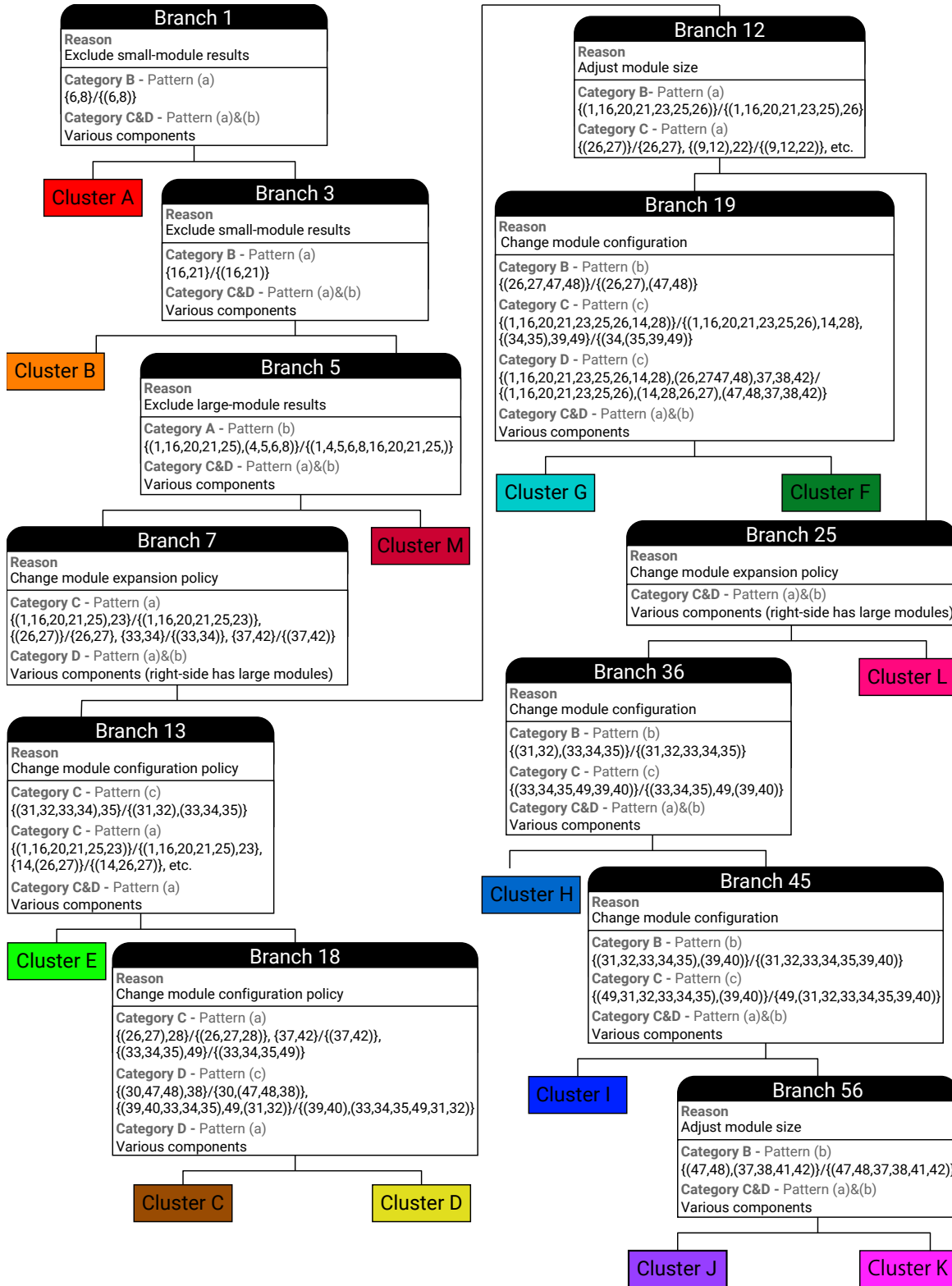


Fig. 6.13: Decision bifurcation diagram.

6.5.4 分析結果の考察

前項で得られた結果において、提案する解空間の構造分析法の適用により、モジュール構成を検討する上での着眼点を明確にしながら、大局をつかむ意思決定から詳細を決める意思決定へと、体系的な手続きに従って選好解を定めることができることを確認する。次に本章で新たに提案した、解空間構造の特徴に基づく解釈法を通じて、デンドログラム上での分岐理由の議論が可能になったかを確認し、それらの妥当性を考察する。

初めに、提案手法によって得られた意思決定分岐図が妥当な結果であるかどうかを検証し、Branching DSM を用いた知見獲得が可能であることを検証する。Branch 1 と Branch 3 は、微小なモジュールで構成される代替案で構成された Cluster A と Cluster B を除外する分岐である。目的関数ではミスマッチタイプ 2 を無視すると、クラスタサイズが極端に小さくなる結果が導かれる。モジュールや部品の統合の可能性を考慮しない結果であることから、分岐マトリクスでは Category C に分類されるセルが多くなっている。逆に、Branch 5 では、大型モジュールで構成される代替案で構成される Cluster M が除外されている。このクラスタでの結果はミスマッチタイプ 1 を無視した場合に生成される。これらの極端な結果は、他の結果とはかけ離れているため、デンドログラムの上部で分岐していると考えられる。

4 つ目の分岐である Branch 7 は、極端な結果を含まない結果の集合であるため、意思決定者が検討すべき具体的なモジュール構成の違いが浮かび上がってくる。図 6.14 は、Branch 7 の分岐マトリクスを整理したものである。図 6.14 で黄色にハイライトされている共通ビルディングブロックである Category A は、Branch 7 に含まれたすべての代替案で採用される基本モジュールである。具体的には、プロペラファンモジュール (18,19)、外装モジュール (1,16,20,21,25)、冷媒輸送管モジュール (4,5,6,8)、バルブモジュール (9,12)、クロスフローファンモジュール (31,32)、フラップモジュール (39,40)、室内電気モジュール (47,48) の 7 モジュールで構成されていることがわかる。これらのモジュールは依存関係が強い部品で構成されていることが確認できる。さらに、これらのモジュールを基調に他の構成要素が統合され、その結果、多様なモジュール構成が代替案として得られていると考えられる。Branching DSM で白にハイライトされた Category A'、つまり (0,0) で固定したビルディングブロックから、基本モジュールを軸にしたモジュール構成の多様性は、概ね図 6.14 の青枠で示した 8 つの範囲に限定されていると考えることができる。この範囲内で複数のモジュールに分けるか、モジュールを統合するかを下位の分岐点で判断することになる。また、図 6.14 では、青枠の範囲を横断する要素の存在を確認することができ、赤枠で示している。このように、Branching DSM を元に解析を行うことで製品のモジュール化にとって重要な知見を獲得することができる。

次に、意思決定分岐図をトップダウンに辿ることで大局をつかむ意思決定から詳細を決める意思決定へと焦点を切り替えながら意思決定が行えることを確認する。図 6.14 から特定したモジュール

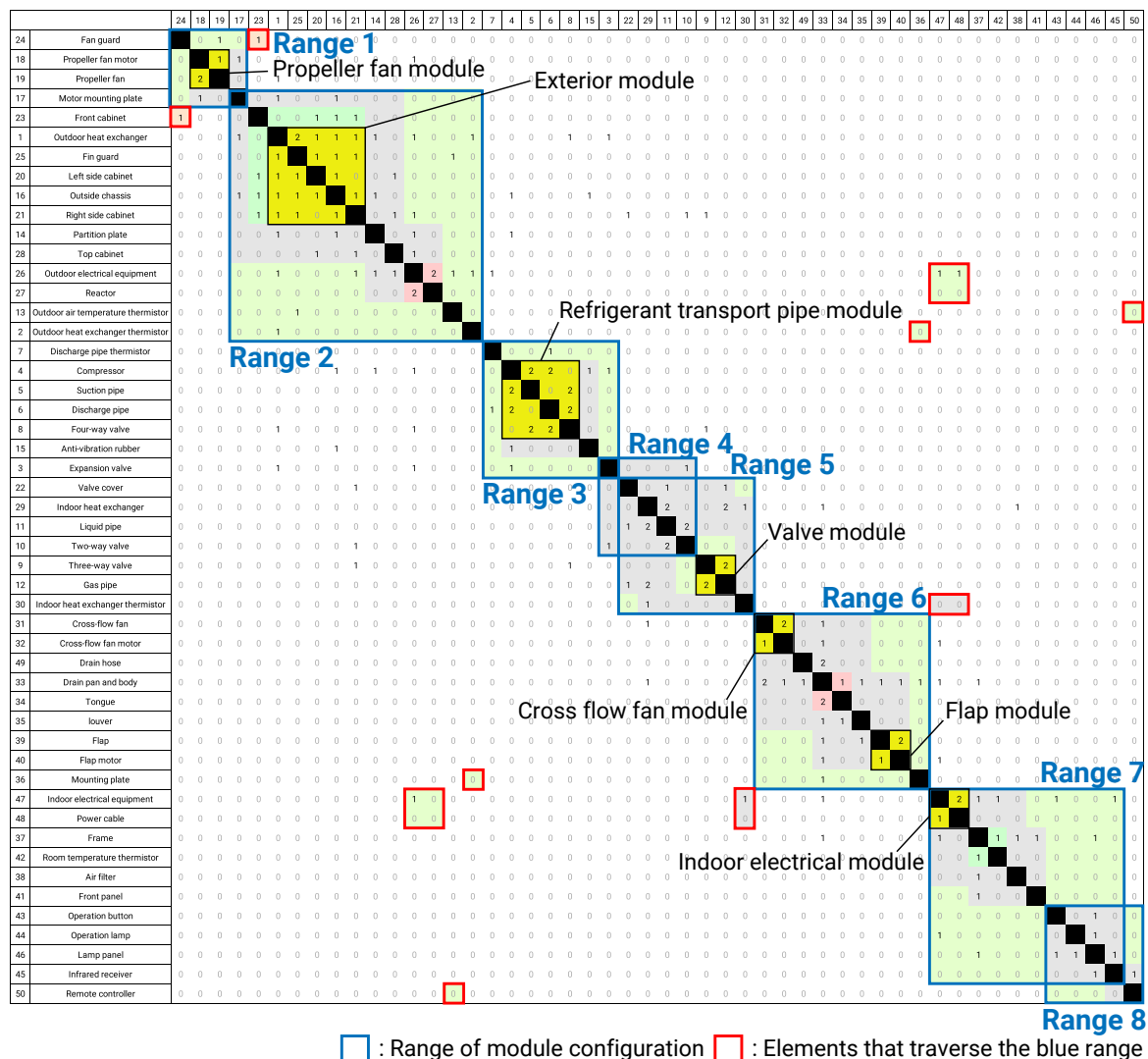


Fig. 6.14: Branching DSM of Branch 7.

ル構成の範囲を元に、これらの範囲の中でモジュール構成の判断が行われていることを、より下位の分岐での Branching DSM を取り上げて確認する。以下の考察では、図 6.14 の赤枠で示した関係のうち、(26,27) と (47,48) に注目することにし、Branch 19 に焦点を当てる。即ち、Branch 19 は Branch 7 の右側クラスターの部分集合である。図 6.13 に示されるように、Branch 7 で右クラスターを選択した場合、Branch 12 に遷移する。Branch 12 ではモジュールサイズを調整する意思決定が行われ、図 6.14 の青枠のうち、Range 2 と Range 7 の範囲を調整する。左クラスターを選択する場合はそれぞれの範囲を部分に分割することになり、Branch 19 に至る。図 6.15 は Branch 19 の Branching DSM であり、(26,27) と (47,48) の関係が Category B として指摘されている。これらは 4 つの配色パターンのうち (c) に分類され、モジュール構成の変化が 2 つ連なった構造になっている。具体的には、Branch12 で分割した Range 2 と Range 7 の部分構造の組み替えを検討することになる。一方で、Range 2 と Range 7 以外の範囲はそのまま保たれていることから、構造

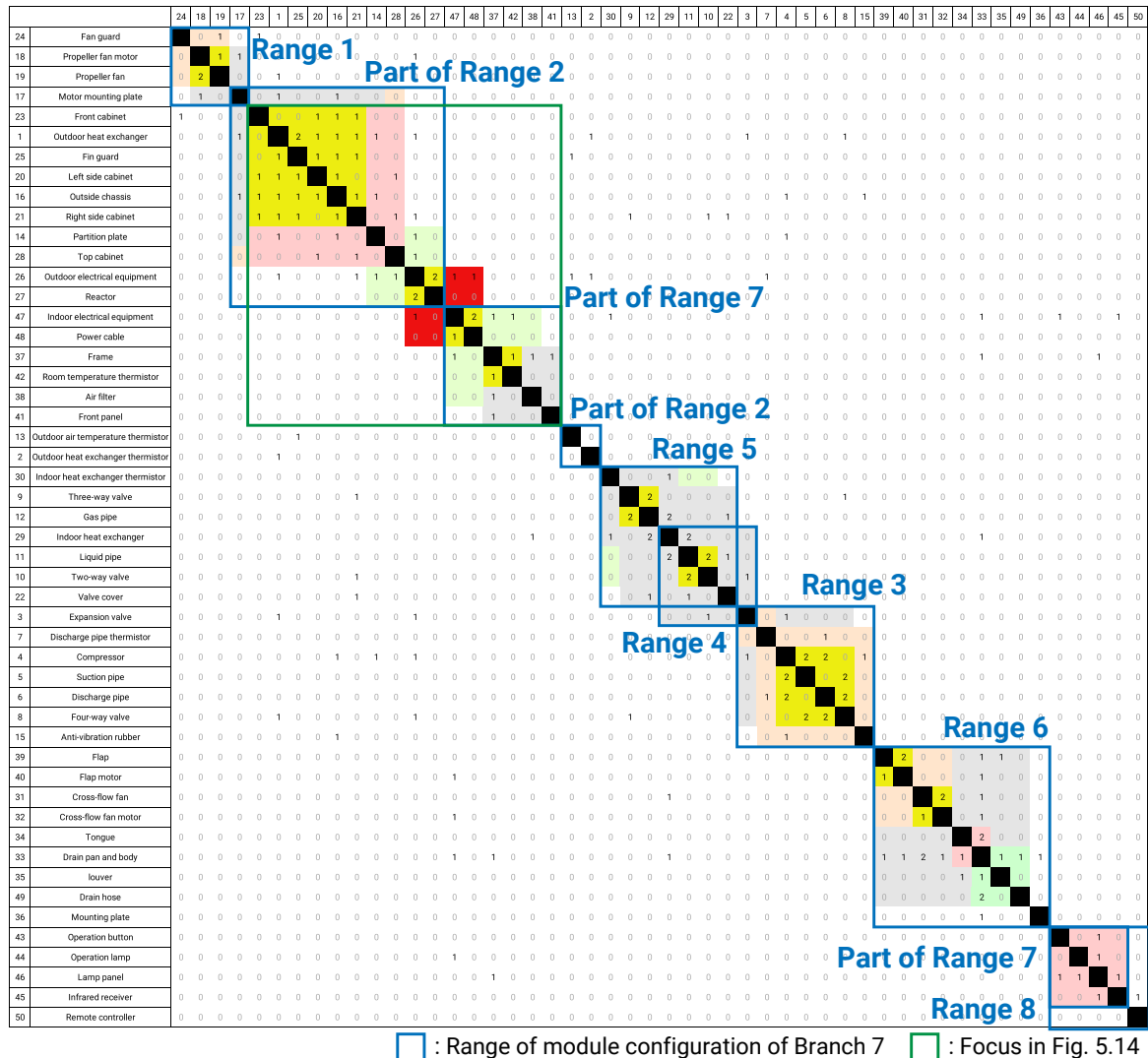


Fig. 6.15: Branching DSM of Branch 19.

分析によって階層的に構成することでモジュール構成の検討を焦点を絞りながら順に判断することができると考えられる。

最後に、Branch 19 の Branching DSM を用いて構造特徴に基づく解釈を通じて分岐理由の議論が可能になったかを確認する。Branch 19 の Branching DSM と左右クラスタの Averaged Modularization Matrix の 5 行列目から 19 行列目までの部分を抜き出し、図 6.16 に示す。当該の範囲は、図 6.15 では緑枠で示されている。図 6.16 の範囲では、次の 4 つのコンポーネント (仕切り板 (14)、トップキャビネット (28)、エアフィルター (38)、フロントパネル (41)) と 4 つのモジュール (外装モジュール (1,16,20,21,23,25)、屋外電装モジュール (26,27)、室内電装モジュール (47,48)、フレームモジュール (37,42)) が存在している。Branching DSM では、屋外電装モジュールと室内電装モジュールの関係が Category B として指摘されている。マトリクス上で赤くハイライトされている部分は左クラスタの共通する特徴を表しており、これらの 2 つのモジュールを統合

		23	1	25	20	16	21	14	28	26	27	47	48	37	42	38	41
23	Front cabinet	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
1	Outdoor heat exchanger	0	1	0	2	1	1	1	0	1	0	0	0	0	0	0	0
25	Fin guard	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
20	Left side cabinet	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
16	Outside chassis	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0
21	Right side cabinet	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0
14	Partition plate	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
28	Top cabinet	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
26	Outdoor electrical equipment	0	1	0	0	0	1	1	1	2	1	1	0	0	0	0	0
27	Reactor	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
47	Indoor electrical equipment	0	0	0	0	0	0	0	0	1	0	2	1	1	0	0	0
48	Power cable	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
37	Frame	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1
42	Room temperature thermistor	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
38	Air filter	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
41	Front panel	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

(a) Branching Matrix

		23	1	25	20	16	21	14	28	26	27	47	48	37	42	38	41
23	Front cabinet	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	Outdoor heat exchanger	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
25	Fin guard	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
20	Left side cabinet	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
16	Outside chassis	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
21	Right side cabinet	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
14	Partition plate	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
28	Top cabinet	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
26	Outdoor electrical equipment	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
27	Reactor	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
47	Indoor electrical equipment	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
48	Power cable	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
37	Frame	0	0	0	0	0	0	0	0	0	0	0	0	1	0.89	0.89	0.89
42	Room temperature thermistor	0	0	0	0	0	0	0	0	0	0	0	0	1	0.89	0.89	0.89
38	Air filter	0	0	0	0	0	0	0	0	0	0	0	0	0.89	0.89	0.89	0.78
41	Front panel	0	0	0	0	0	0	0	0	0	0	0	0	0.89	0.89	0.89	0.78

(b) Left-side Averaged Modularization Matrix

		23	1	25	20	16	21	14	28	26	27	47	48	37	42	38	41
23	Front cabinet	1	1	1	1	1	1	0.1	0.8	0	0	0	0	0	0	0	0
1	Outdoor heat exchanger	1	1	1	1	1	1	0.1	0.8	0	0	0	0	0	0	0	0
25	Fin guard	1	1	1	1	1	1	0.1	0.8	0	0	0	0	0	0	0	0
20	Left side cabinet	1	1	1	1	1	1	0.1	0.8	0	0	0	0	0	0	0	0
16	Outside chassis	1	1	1	1	1	1	0.1	0.8	0	0	0	0	0	0	0	0
21	Right side cabinet	1	1	1	1	1	1	0.1	0.8	0	0	0	0	0	0	0	0
14	Partition plate	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.8	0.8	0	0	0	0	0	0	0
28	Top cabinet	0.8	0.8	0.8	0.8	0.8	0.8	0.1	0.2	0.2	0	0	0	0	0	0	0
26	Outdoor electrical equipment	0	0	0	0	0	0	0	0.8	0.2	1	0	0	0	0	0	0
27	Reactor	0	0	0	0	0	0	0	0.8	0.2	1	0	0	0	0	0	0
47	Indoor electrical equipment	0	0	0	0	0	0	0	0	0	0	1	0.4	0.4	0.1	0	0
48	Power cable	0	0	0	0	0	0	0	0	0	0	1	0.4	0.4	0.1	0	0
37	Frame	0	0	0	0	0	0	0	0	0	0	0.4	0.4	1	0.6	0.6	0.6
42	Room temperature thermistor	0	0	0	0	0	0	0	0	0	0	0.4	0.4	1	0.6	0.6	0.6
38	Air filter	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0.6	0.6	0.6	0.5
41	Front panel	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.6	0.5	0.5

(c) Right-side Averaged Modularization Matrix

Fig. 6.16: Some details extracted from branching matrix of Branch 19 (a) branching matrix (b) left-side averaged modularization matrix (c) right-side averaged modularization matrix.

し、電装モジュールとしてまとめている。一方で、右側のクラスターでは全ての代替案で2つのモジュールを別のモジュールとして取り扱っている。さらに、この部分に関連する Category C と D は、意思決定者により多くの知見を与える。図 6.16 では、右側クラスターに注目すると、カテゴリー C は、屋外電装モジュールと仕切り板、トップキャビネットとを1つのモジュールにまとめる可能性を示している。また、フレームモジュールをエアフィルターと合わせて室内電装モジュールに割り当てる可能性を示している。一方、左側クラスターでは、Category B として電装モジュールを形成した後に、付随する要素として、Category C として仕切り板とトップキャビネットを外装モジュールに組み込んでいることを示している。さらに、ここで明らかになった構造の特徴から、DSM の要素に注目して、それらの意味を考察することができる。Category B として記述された要素である $\delta_{26,47}$ 、 $\delta_{26,48}$ 、 $\delta_{47,26}$ の3つの依存関係は電気信号の流れを記述した要素であり、左側クラスターは、これらをモジュール内部の依存関係として取り扱うため、電気信号に関する機能を優先

したモジュール構成となっていることがわかる。一方、右側のクラスタでは、 $\delta_{14,26}$ 、 $\delta_{26,14}$ 、 $\delta_{26,28}$ 、 $\delta_{28,26}$ 、 $\delta_{37,47}$ 、 $\delta_{47,37}$ は物理的な接触関係を表しており、すべて Category C として記述されていることから、物理的な位置関係を優先した構成であると考えられる。物理的な構造としては、モジュールは屋内と屋外に大別されるが、機能に着目すると、電気機器がそれらを横断している。屋内モジュール、屋外モジュール、電気機器の関係性は、モジュール構成の分岐点として適切な要素であるため、本事例の結果は妥当なものであると考えることができる。

6.6 結言

本章では、解空間にネットワーク構造を持つ問題であるモジュール化設計問題を取り上げ、解空間構造の分析法の応用を論じた。モジュール化設計は DSM と呼ばれる包括モデルが用いられる。数理最適化による DSM のクラスタリング操作に相当する解空間の構造を表現するため、Modularization Matrix と呼ばれるスキーマモデルを導入し、本研究で提案する解空間の構造分析を可能にした。

モジュール化設計では競合関係にある代替案をさまざまに想定できるものの、モジュール性能評価指標そのものが代替案の特徴を説明するものではなく、選定の基準として馴染まない。そこで、解空間の構造分析によって獲得された構造上の相違点の傾向を構造の特徴として扱い、代替案の解釈に用いる枠組みを提案した。構造の特徴を把握するため、2つのスキーマの相違点とそもそもの DSM モデルとを一括して参照できる Branching Matrix を導入し、その分析法を構築した。

家庭用空調機を例に引いた検証を通して、意思決定分岐図を用いてモジュール構成を左右する大局を決める決定から詳細な決定へと体形的な手順を追って意思決定が行えることを確認し、その結果の妥当性を示した。

第 7 章

組合せを伴う解空間の構造分析の適用 における課題と展望

7.1 緒言

本章では、第 4 章から第 6 章での 3 つの事例を総括して、組合せを伴う解空間の構造分析の適用における課題と展望について論ずる。本研究での提案手法は、最適設計という「開いた問題」のための方法であるため、対象を一般化してその有効性や妥当性を十分に満たしているかを確認することはできない。そこで本章では、個別事例へ適用したことによる成果を積み重ねて、その限定性を念頭におきながら、提案した枠組みをより広範な最適設計に展開していく上での課題や展望を論じる。まず、提案手法の実用的な意義や限定性を考察するため、個別の適用事例から手法適用に向けた前提や仮定を整理して適用の条件を論じた上で、提案手法の効果と課題について考察する。さらに、3 つの事例の位置付けについて確認し、更なる手法適用の展開や展望について考察する。

7.2 組合せを伴う解空間の構造分析法の適用条件と課題

7.2.1 提案手法での仮定と手法適用の条件と限界

本研究で提案した組合せを伴う解空間の構造分析法は、多くの仮定や前提のもとで成り立っている。本項では、個別の適用事例から手法の前提や仮定を改めて整理し直すことによって、提案手法の適用条件を明らかにし、技術的限界と課題について論じる。

提案手法を適用する上での仮定や前提は次の 4 点にまとめられ、それぞれの観点から構造分析法の適用条件や限界を理解することができる。

ビルディングブロック仮説 提案した解空間の分析法は、組合せ問題の最適解は解の優劣や多様性に影響する重要な部分的な組合せ構造 (ビルディングブロック) の集積で構成されている

という、ビルディングブロック仮説を前提としている。そもそも、重要な部分的組み合わせ構造を想定することができない問題であれば、提案手法による代替案理解の枠組みに馴染まないため、適用条件の1つとしてみなすことができる。本研究での3つの事例では、多目的遺伝的アルゴリズムによって導かれたパレート最適解を分析の対象としている。そもそも、遺伝的アルゴリズムは、遺伝的操作によってビルディングブロックをうまく処理することで最適解を効率よく探索する手法であるため、これらの対象では、ビルディングブロックが存在を前提としても一般性を失わないと考えられる。第4章では、提案手法によって抜き出したビルディングブロックの集積方法による違いを定量評価することで、目的関数への影響を調査し、その妥当性を確認した。

スキーマモデルとその解釈可能性に関する前提 提案手法では、解空間は設計において変動を想定することができる変数によって生成されるとし、遺伝的アルゴリズムのスキーマ定理^[53]を応用して、変数から解の特徴を抽象し、解空間の構造を定義した。本研究では構造を表すモデルをスキーマモデルと呼称している。分析を行うための技術的な制約から、スキーマモデルは次の3点を前提としている。(1) スキーマモデルはバイナリ変数で定義可能であること。(2) スキーマモデルによって代替案の特徴を十分に表現できていること。(3) 3.4.2項で述べたスキーマモデルの条件を満たしていること。

(1) 提案手法では2つのビット列のパターンで集団を比較するため、バイナリ変数でスキーマモデルを定義することを前提にしている。本研究でスキーマによる集団表現を行う理由は、相違点や共通点を顕在化することや、類似度を定量化することであり、第3章ではバイナリ変数を用いた場合での比較や定量化の方法を提案した。元々のスキーマの概念に立ち返れば、スケジューリング問題などの一部の組合せ最適化問題では、シンボル列を用いたスキーマ定義などを用いることが一般的であり、さまざまな符号化法が存在している。提案した方法をそのままバイナリ変数以外で定義されたスキーマモデルに転用することはできない。しかし、第4章と第6章では直接バイナリ変数を用いて表現できない事例を引いて、問題の特性を踏まえた中間モデルを構築することでバイナリ変数を用いたスキーマモデルへの変換が可能であることや、その元での代替案の解釈が可能であることを示した。分析の方法としては条件があるものの、事例のように問題の特性を加味しつつ柔軟に対応することで、提案手法は幅広く有用であると考えられる。

(2) 提案手法はスキーマモデルを用いて代替案の特徴やその他の案との関係性を論じるため、スキーマモデルを構築する際に捨象された事項は考慮することができず、解釈上の限定性が潜んでいる。スキーマモデルとして抽象する部分はビルディングブロックの候補であり、代替案の重要な特徴が何かについて考察した上で、問題に合わせて設定することが必要である。代替案のどの部分を抽象するのかは、意思決定者がどの部分に注目したいかに対応しているため、分析方法の特性を踏まえつつ柔軟に設定することができれば、提案する枠組

みは有用に働くと考えられる。第4章の事例では、最適設計での目的を踏まえて、ビルディングブロックの候補として2つの考え方を示し、それぞれに対応するスキーマモデルを提案した。それぞれのスキーマモデルで解空間の構造分析を行うことで、それぞれの視点から代替案の関係性やビルディングブロックの構成が解釈可能であることを示した。

(3)3.4.2項では、スキーマモデルが満たすべき条件を3つ設定した。条件を満たさないスキーマモデルは望まない結果を導く可能性があり、条件を満たすスキーマが定義可能であることそのものが手法を適用する上での条件になっていると考えられる。例えば、(B)スキーマモデルの各ビットはそれぞれの設計上の判断に対応しており、それらは独立した関係にあるという条件は、ビット列の固定方法の組合わせによって考察を行うための技術的限界による条件である。遺伝的アルゴリズムでは、整数変数や実数変数では、2進数に変換するか、グレイコードと呼ばれる符号化法を用いて遺伝子列を定義する。これらの場合は複数のビットで1つの情報を表すため、特定のビットの固定方法の意味を解釈することができない。したがって、設計変数の量的関係を代替案の特徴として抽象することができず、手法の限界の1つとして考えられる。

以上の3点の前提を破綻する場合、本研究で提案した方法をそのまま適用することができない。しかし、本研究で提案した分類操作や意思決定の枠組みは、スキーマの特性に関わらず適用可能であるので、スキーマの定義方法やビルディングブロックの解釈の方法を問題に合わせて工夫することで、これらの条件は克服可能であるため、提案する枠組みとしての基本的な構想は、機能すると考えられる。事例では、3つの種別の組合せ最適化問題を取り上げ、それぞれの問題の特徴を整理してスキーマモデルを個別に提案し、それぞれで手法の効果を確認した。

解空間の最適性に関する前提 本研究では、ビルディングブロックを組合せ問題の最適解は解の優劣や多様性に影響する重要な部分的な組合せ構造として定義していて、代替案を説明するスキーマモデルの固定文字列をビルディングブロックとして取り扱っている。これは、分析対象である代替案の集合が有望かつ多様であるという観念の上で成り立っており、解空間の最適性が担保されていることが前提となっている。最適化結果の質については本研究の焦点の外ではあるものの、局所最適解などの存在は解空間の構造の解釈に影響を与える。提案手法は目的関数の値は分類操作で参照しないので、目的関数値が大きく外れていることは分析には影響しないものの、提案手法では局所解を認識することはできず、大域的な最適解と同等に扱ってしまう。しかし、局所最適解の構造がその他と大きく異なる場合は、相違点の感度が高いため、第6章の事例で検証したように、ビルディングブロックでは上位の分岐でその他のクラスタとして排除されることになる。局所最適解が持つ部分的組合せ構造を把握できること自体は本手法の成果の1つとも考えられ、分析結果をモデル化や定式化に反映させることで、解探索の性能向上などに寄与する可能性を秘めている。

意思決定のプロセスに関する前提 本手法は代替案の体系化方法に階層性を前提にしている。根源的には、解を選定するという操作そのものは、比較によって行われている。特に複雑な対象においては、2.2.1 で言及したように、サイモン^[11]の準分解可能性の観点からも、グループに分けて考えるという方策が有効である。本手法では3.2.2で述べた構想のもと、比較の手続きに、対象を2つのグループに分けることを前提とした。グループ分けの分割数を暗黙的に固定することは柔軟性に欠け、この前提が解釈を制約する場合があると考えられる。例えば、三棘みなトレードオフの関係がある場合などは、2分割ではその特徴を顕在化することができないため、提案手法は解釈の解像度に限界がある。一方、本手法は、2分割を前提とするからこそ、認知的負荷を低減し、代替案の解釈を可能にしているという点に注意しなければならない。提案手法での2つのグループ比較ではスキーマのビットに応じて4つのカテゴリを設け、さらに、第6章では左右の違いも含めて9色を用いて可視化した。比較のグループ数が増えるとそのカテゴリも組合せ論的に増大し、煩雑になることは避けられない。本来の結果の解釈の難しさからわかるように、選択肢を広げることは必ずしも適当なことではなく、ある前提のもとで限定して議論することは、解釈が可能になるという点では意味がある。他の分類の過程を前提とした手法の拡張については今後の研究課題である。

7.2.2 構造分析法の効果と課題点

2.4.3 項では、組合せを伴う多目的最適設計において、2つの課題を抱えていることを指摘し、本研究の焦点を明確にした。また、3.6.3 項では、手法適用の効果について考察した。本節ではこれらを踏まえて、3つの事例を横断して、指摘した課題と手法適用による効果の妥当性を考察し、手法の限界と課題点について論じる。

2.4.3 項では、多目的組合せ型設計問題では、同等の目的関数値を持ちながらも組合せのパターンが異なる等価解が頻繁に生じるため、多目的設計探査などの目的関数による分析では代替案の特徴を見出すことができない課題を指摘した。本研究では、3つの事例全てで代替案の集合に等価解が潜んでいることを確認し、課題の妥当性を確認した。第5章では散布図からそれぞれの目的関数に組合せ的な傾向を確認し、等価解が頻繁に生じることを確認した。第6章では、小規模なDSMモデルを例に目的関数の設定によっては多くの等価解が存在する可能性があることを説明した。提案手法では、この課題を受け、設計において変動を想定できる部分から代替案の特徴を抽象し、解空間の構造を定義することで等価解の解釈を可能にした。第4章では等価解として目的関数値が類似した2つの代替案に焦点を当て、それらの違いについて解空間の構造からの解釈を行った。階層的クラスタリングではそれらが上流で分岐していることを確認し、解空間の構造に着眼することの意義と効果を検証した。等価解の特徴は解空間に焦点を当てることで説明可能になったものの、分析では目的関数に関する特徴は捨象するため、過度な抽象によってスキーマモデルを構築した場合、

例えば、解の単射性が保たれないなどの事態では、構造の意味での「等価解」が生じる可能性がある。第5章では、代替案をビルディングブロックの集積として説明可能にするだけでなく、ロジスティック回帰分析を用いて、それぞれのビルディングブロック集積の意図の考察を可能にする枠組みを提案した。この方法は目的関数が多数になった場合に有効な方法であり、多様な問題への応用が期待される。一方で、ロジスティック回帰分析の技術的課題により、最下層の分岐など要素数が少なくなる場合にうまく作動しない課題を抱えている。

2.4.3 項では、複雑な組合せ型設計問題である場合は、組合せが重層しているため、解の単純な比較検討では内部の傾向を掴むことが困難であることを指摘し、Trade Space Exploration のような可視化では、設計変数の傾向を獲得しようにも関係やパターンを見出すことが難しく、その認知的負荷の高さを課題として設定した。例えば、代替案が n 個存在する場合、それらの比較を行うには n^2 のオーダーで比較検討が必要であり、代替案が多いほど反復的な処理が増大する。さらに、組合せ最適化問題の設計変数ベクトル $\mathbf{x}$ が m 次元である場合、それらの部分的な共通点や相違点はビット列で定義されている場合であれば 2^m だけ存在することから、組合せ論的に比較検討の負荷が増加することがわかる。そこで提案手法では、統計的手法によって設計変数のパターンの理解を可能にし、着眼点を階層的に体系化することにより、認知的負荷の軽減を実現した。代替案の相違点は木構造に階層化して意思決定分岐図に整理されているので、意思決定分岐図をトップダウンに辿ることによって特定の代替案に至る場合、比較検討の繰り返し回数を最大でも $(n-1)$ 回に低減させることができた。第4章では、距離の定義やリンケージ手法の定義によって特定の解に至る分類の過程が異なることを示し、問題や意図に合わせて選択する必要があることを指摘した。また、それぞれの分岐点ではスキーマの排他的論理和を取ることによって、 2^m のビット列のうちどの部分に焦点当てれば良いかを提示することができる。第4章では、2つの等価解での相違点を3つの部分に分けて整理し、それらの違いに対して意思決定分岐図上での着眼点としての優先度を設定できることを確認した。このような比較検討の回数の低減と着眼点の提示という意味での認知的負荷の低減は、前項で述べたように、意思決定のプロセスとして2つのグループへの分類とそれらの比較を繰り返し行うことを前提としていることによる効果である。この前提は前項で述べたような技術的限界を抱えていることに注意しなければならない。

以上2つの課題解決に加えて、3.6.3 項では、提案手法の適用による効果として、まず、それぞれの結果に至る経緯をビルディングブロックの集積として解釈できるようになることを述べた。第4章と第5章では特定の代替案に焦点を当てて、その代替案に至る選定の経緯をビルディングブロックの集積として解釈可能であることを示した。第4章の事例では、代替案12の20の文字列に対して、いくつかの部分に分割し、それらの拡張や統合によって形成されることを確認した。いくつかのリンケージ手法とスキーマモデル定義によるビルディングブロックの集積過程の違いを検証することで、スキーマ定義に依存して代替案のビルディングブロックの構成が定まり、リンケージ手法によってその集積過程の詳細度が変化することを確認した。さらに、3.6.3 項では、提案手法の分析

結果をまとめるテンプレートである、意思決定分岐図について、トップダウンに読み解くこととボトムアップに読み解くことの2つの効果を論じた。トップダウンに辿ることによって、対局をつかむ意思決定から詳細を決める意思決定へと、体系的な手続きにしたがって選好解を定めることができることは、第6章の Branching Matrix で検証することができた。Branching Matrix では、図 6.14 からスキーマの固定方法からモジュール割り当ての範囲を定めることができることを確認し、さらに下位の分岐である図 6.15 ではその範囲の中での詳細を決定することができることを確認した。また、ボトムアップに辿ることによって代替案の関係性を俯瞰できることは、第4章の代替案 10 と代替案 11 の関係から検証することができた。これらの代替案は等価解であるものの、3つの部分的な相違点があった。図 4.15 の意思決定分岐図から、それらの違いに優先度をつけて解釈できることを示した。

提案した構造分析法は、最適設計の手順での、(6) 結果の解釈を支援する方法であることから、解釈から得られた知見をもとにモデル化や定式化を見直すことで構造分析の対象をピボットし、構造分析を反復的に行うことで、更なる効果を発揮すると考えられる。本研究では、分析の対象を組合せを伴う解空間として設定し、有望かつ多様な代替案の集合が織りなす空間として定義した。有望かつ多様とは設計対象が達成すべき機能や目的のもとでの概念であり、最適設計においては最適化問題として数学的な形式で記述することから、実用上は、有望かつ多様な代替案の集合は多目的最適化によって導かれるパレート最適解集合として解釈される。3つの事例では、ある定式化のもとでの多目的最適化を実行することで分析対象を1つに定めてその結果の考察を行った。しかし、2.3.1 項で最適設計が「開いた問題」としての性質を持つことを指摘したように、構造分析もまた、それ自体で優れた設計を行うことができない、ある意味での限定性があることに注意しなければならない。分析対象はパレート最適解集合として導かれた部分空間に過ぎず、図 3.1 で示した範囲に限定されている。実際のシステム設計の手順では、図 2.1 に示されるように、あるモデル化や定式化のもとで得られた結果をもとに、モデル化や定式化の方法を見直して、満足のいく結果が得られるまで反復的に設計が行われることになる^[12]。反復的な操作のそれぞれにおいて構造分析を施すことによって、より広範な解空間を探索することができるため、意思決定者の設計問題の性質と各代替案の特徴の理解を促すことができると考えられる。

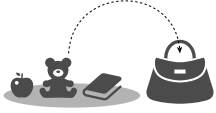
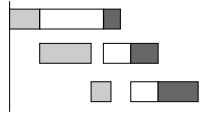
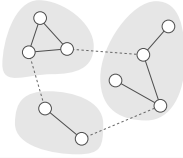
7.3 組合せを伴う解空間の構造分析法の最適設計への展開

7.3.1 組合せ最適化問題の種別に着眼した展開

本研究での事例は産業界の課題に偏っているものの、「最適設計」のための手法であり、数理最適化を活用する問題として設定可能な問題であれば、分野を問わず応用が可能である。

提案手法の分析対象である解空間は、多目的最適化問題のパレート最適解であるので、スキーマモデルとして抽象するビルディングブロックの候補の検討は、最適化問題の定式化法に依存すると

Tab. 7.1: Examples of categories of combinatorial design problems.

Problem category	Selection structure	Permutation structure	Network structure
			
Typical benchmark problem	Knapsack problem	Travelling salesman problem	Grouping problem
Examples	Module selection Facility location Supplier selection	Flow-shop scheduling Nurse scheduling Routing problem	Modular design Team formation Assembly line design
Schema definition	Direct definition	Introducing an intermediate model	

考えられる。本節では、最適設計での定式化方法をベンチマーク問題を起点に整理することで、その他の最適設計問題への展開の可能性について論じる。

第4章から第6章で扱った3つの事例は、それぞれ第4章は巡回セールスマン問題、第5章は輸送問題やナップサック問題、第6章はグループ化問題と、定式化の方法に対応して、基本となるベンチマーク問題を想定することができる。組合せ最適化問題のベンチマーク問題と、それらの解空間構造の一例は表7.1にまとめられる。

最も単純な解空間構造は第5章で扱った選択構造である。表7.1の左に位置している。ナップサック問題と呼ばれるベンチマーク問題は、あらかじめ用意された n 種類の品物に価値 v_i や、重量 w_i が与えられており、ナップサックに入れる品物を計画する問題である。重量の合計がナップサックの容量 W を超えない制約のもとで、価値の合計を最大化する品物の組合せを計画する。同じ種類の品物を1つまでしか入れられない場合は、設計変数は $x_i \in \{0, 1\}$ で定義される。選択を表す設計変数のバイナリ情報はそのままスキーマモデルとして転用可能であり、本研究ではこのスキーマモデルで定義された解空間構造を選択構造と定義した。第5章のグローバルプロダクトファミリー問題は混合整数計画問題として定式化されたものの、ストラテジックレベルの意思決定に、ナップサック問題と同様の設計変数の設定が可能であることに着眼し、選択構造を表すスキーマモデルを構築した。ナップサック問題と同様の定式化を行う問題としては、モジュール選択問題や施設配置問題、サプライヤ選択問題など、さまざまに想定することができ、これらへの展開を考える上では、スキーマモデル構築に関する考え方を転用することができると考えられる。

第4章で扱った生産スケジューリング計画の解空間構造は、順列構造として定義した。順列構造問題の代表的なベンチマークは巡回セールスマン問題である。巡回セールスマン問題とは、あらかじめ用意された都市のセットについて移動距離が最小になるように訪問する順序を決定する問題である。順列構造では設計変数は用意された集合の順序であり、そのままではバイナリとしてスキーマ

マを定義することができず、ある変換操作のもと、バイナリ変数の中間モデルをスキーマモデルとして用いることになる。第4章では、ビルディングブロック仮説に基づいて、目的関数の値に影響する要素は何かという視点から、2つのスキーマモデルを定義した。表7.1で示した、ナースケジューリング問題や、経路問題なども設計変数の定義方法や評価方法が類似しているため、本手法を展開する上では、第4章の方策を応用することができると考えられる。また、第6章で扱った製品モジュール化問題は、ネットワーク構造として定義されており、そのベンチマークはグループ化問題である。順列フローショップスケジューリング問題と同様にそのままではスキーマモデルを定義できず、第6章でのスキーマモデルの定義の方策が応用可能であると考えられる。例えば、グループ化問題では、モジュール化問題の他に、人員配置問題や配車計画問題、マーケットテリトリ決定問題などの応用例がある。

組合せ最適化問題のベンチマークは表7.1に限らない。例えば、最小全域木問題などのベンチマーク問題が知られており、その応用であるコンピュータネットワーク関連の設計問題、例えばSTPの通信経路などは本論文で取り上げていない種別の最適設計問題であり、提案手法の応用の1つとして考えられる。これらの問題に向けては、第3章の指針のもとスキーマモデルの定義や距離定義、リンケージ手法の選択、ビルディングブロックの解釈法を個別に検討することが求められ、異なる種別の問題への展開は今後の研究課題であると考えられる。

7.3.2 最適設計の結果の評価方法に着眼した展開

本研究では、設計過程でのモデルの範囲の異なる問題を取り上げることで、解釈の方法の観点からも多様な問題に取り組んできた。第4章は比較的規模の小さな問題であったので、評価関数空間の可視化による解釈法を用いたが、第5章での多数評価指標での解釈法や第6章での構造に基づく解釈法は問題の特性に合わせて解釈方法を柔軟に変えた結果である。本節では、3つの事例での評価方法の違いに焦点を当てて、設計過程での段階から手法の展開について論じる。

2.2.4節では、近年の計算機の性能向上や進化計算技術に起因して、大規模かつ複雑な問題でも有意な時間で解を導出できるようになってきていることを指摘した。部分問題として扱う問題を複雑化できることは、最適設計でのモデル化や定式化の段階でより忠実度の高いモデルを扱ったり、設計の範囲としてさまざまな事象を取り入れたりすることに繋がる。広範な概念を扱う場合はモデルそのものが大規模かつ複雑になり、評価指標も多様になる傾向にあり、設計の詳細化の過程に対応させて論じることができる。解くべき問題がはっきりしている詳細設計においては、その評価の方法も具体的である。第4章で扱った生産スケジューリング計画はその典型例であり、目的関数の値そのものが代替案の選定の理由になる。第4章ではビルディングブロックの意味を評価関数空間の可視化によって説明可能にしたが、単純な可視化が有効であるのはせいぜい3次元までである。一方、第5章で取り上げた、グローバルプロダクトファミリー設計などの問題は、基本設計などに相当し、抽象度が一段と上がるため、評価方法が多様になり、その意味も包括的な指標となる。第5章

では、代替案の選定の基準になる評価値を5つ用意して、ロジスティック回帰分析を応用した解釈法を提案した。これらの評価関数の考え方は前項の定式化法というよりむしろ、選定の方法に重点があり、評価指標を用いて選定を行う場合は同じ方法を異なる種別の例題に展開することが可能である。言い換えれば、ロジスティック回帰分析による解釈法はグローバルプロダクトファミリー設計に特化した方法ではなく、第4章の例題についてもロジスティック回帰分析を用いることが可能である。

さらに上流の設計では抽象度も一層高まるため、代替案を選定する方法そのものが変わってくる。第6章で扱った製品アーキテクチャの設計問題では、DSMと呼ばれる包括モデルを用いて計画を行った。最適設計で考える範囲が広い場合は、大胆な前提や仮定のもとでグラフやマトリクスなどの簡素なモデルを用いて抽象することが一般的であり、グラフ理論を応用してその優劣を定量化することで最適化の数理的枠組みを応用できるようになる。例えば、第6章と同じ製品のモジュール化についても、Modular Function Deployment(MFD)と呼ばれるQFDマトリクス^[114]を拡張したモデルを用いた手法^[115]や、有向グラフを用いた手法^[116, 117]が提案されている。これらの最適化で用いる目的関数はモデルを操作するための指標であるため、抽象度が非常に高く、選定の指標として適していない。第6章では最適化結果のモジュール構成の特徴を把握するため、顕在化したビルディングブロックを元のDSMモデルに反映することで特徴を説明可能にしたが、この方法もあくまでもDSMのクラスタリングに特化した方法であることに注意しなければならない。したがって、抽象度の高い問題に提案手法を応用する場合は、最適化問題の形式にとらわれず、モデル化の方法に立ち返って、どのように結果を解釈するのかを念頭におきながら、個別にビルディングブロックの解釈法を検討する必要があると考えられる。

7.4 結言

本章では、本研究で提案した組合せを伴う解空間の構造分析について、適用における課題と展望について論じた。まず、提案手法の実用的な意義や限定性を考察するため、個別の適用事例から手法適用に向けた前提や仮定を整理して適用の条件を論じた。手法適用に向けた条件として、ビルディングブロック仮説、スキーマモデルとその解釈性、解空間の最適性、意思決定プロセスの前提についてその詳細を述べ、限定性とそのもとでの手法の意義について論じた。さらに、2.4.3項や、3.6.3項で論じた、組合せ型最適設計での課題点や、手法適用による効果について、第4章から第6章までの3つの事例を束ねて、それらの妥当性について論じた。最後に、3つの事例の位置付けについて、定式化法と分析法の2つの視点から整理し、手法適用への展開や展望について考察した。

第 8 章

結論

本論文では、最適設計における意思決定者の役割に焦点を当てて、最適化計算によって得られた代替案から知見を獲得したり、選好解を定めたりする意思決定の局面の重要性を論じ、解空間の構造を分析することで、意思決定者が判断する着眼点を明らかにする手法を提案した。単純な比較検討から特徴や傾向を見出すことが困難である、組合せ型の多目的最適設計を対象に、組合せを伴う解空間の構造分析法の基本を構築した。構築した手法は異なる種別や特性を持つ最適設計問題への応用を通してその有効性が検証された。

組合せ型の多目的最適設計で解の候補として扱われる代替案の集合は、評価指標のトレードオフ関係によって導かれる。トレードオフでの競合関係はそれ自体では唯一の解を定めることができず、意思決定者の介入が必要である。組合せ型設計問題での秀でた解には、ビルディングブロックと呼ばれる優れた解に共通する部分的構造が含まれていると仮定し、ビルディングブロックの集積過程を代替案の絞り込み過程に連動させ、ビルディングブロックの顕在化と階層的な体系化に基づいた解空間構造の分析手法を構築した。ビルディングブロックに基づいて代替案の解釈を行うことで、代替案の特徴や他の代替案との関係を理解することが可能になり、意思決定者の着眼点を明確化することができるようになった。以下では、各章で得られた成果を総括する。

第 1 章では、設計における詳細化の過程での最適設計の位置付けを述べた上で、優れた最適化手法だけでは優れた解に至ることはできないことを指摘した。意思決定の局面が重要であることを論ずることにより、研究の背景と目的を示した。

第 2 章では、一般的な最適設計の形式やその進め方について述べ、その課題と解空間の構造分析法の必要性について論じた。最適設計の対象の複雑化を受け、定式化でのモデルの粒度の深まりや範囲の広がりを背景に、進化計算に代表される最適化技術の進化から複雑な問題でも一網打尽に結果を導出することができる一方で、その解釈の段階で難しさを抱えていることを指摘した。目的関数のトレードオフ関係に代表される競合の関係に焦点をあて、それ自身では唯一の解を定めることができないため、意思決定者の介入が必須であることを述べた。意思決定者の代替案理解を難しくする要因をシステム思考の氷山モデルを用いて整理し、背景に潜む構造に焦点を当てることの効果

を論じた。

第3章では、解空間の構造分析による意思決定者の構想を述べ、具体的な分析法を提案した。提案する組合せを伴う解空間構造は3つの手順から構成されている。Step 1では多目的最適化による代替案の生成を行い、解空間の構造を定義する。本論文での解空間とは設計において変動を想定できる要素によって定義される空間であり、有望かつ多様な代替案の集合を分析対象として設定した。多様かつ有望な代替案とは通常、多目的最適化のパレート最適解として解釈され、進化計算技術を応用して代替案が生成される。解空間の構造は、遺伝的アルゴリズムのスキーマ定義に着想を得て、バイナリ情報を用いたスキーマモデルを導入した。スキーマモデルは最適化問題の定式化に準拠し、個別に考える必要があることを指摘した。スキーマモデルを構築する際の満たすべき3つの条件を提示し、構築の指針を示した。Step 2では、クラスタリング法による代替案の構造化とビルディングブロックの顕在化を行う。ビルディングブロックの階層的な体系化を前提に、クラスタリング法では階層的クラスタリングを用いることにし、代替案の類似度の評価方法とリンケージ手法の選択の指針を示した。ビルディングブロックは階層的クラスタリングで二分されるクラスタのスキーマの固定部分として定義し、ビルディングブロックの相違点はスキーマの排他的論理和として定義することで、クラスタリング法によるビルディングブロックの特定を可能にした。Step 3では階層的クラスタリングの結果をもとに、ビルディングブロックの累積過程を分析し、意思決定分岐図として体系化する。解釈のためにスキーマのビットの組合せに応じた4つのカテゴリを定義し、分析の焦点を明確にした。意思決定分岐図にはビルディングブロックの累積による効果を説明するため、左右のクラスタの特徴を記載する。代表的な方法として2つの方針を述べ、構造分析による意思決定支援の効果について論じた。

第4章から第6章では種別の異なる組合せ型の多目的最適設計問題を取り上げ、それぞれへの応用の方法や解釈法を個別に提案した。それぞれでは事例を通した検証を経て、その結果の妥当性と手法の有効性を確認した。

第4章では、ベンチマーク問題として生産スケジューリング問題を例に引いて、提案手法の一連の流れを示し、結果の妥当性と有効性を確認した。生産スケジューリング問題はジョブの順序を決定する最適化問題として定式化される。本論文ではこの解空間の構造を順列構造と定義し、スキーマの定義方法によって解釈が変化することを確認するために、2つのスキーマモデルを導入した。さらに、顕在化したビルディングブロックの解釈には目的関数空間の可視化による方法を導入し、解釈のための2つのシナリオを提示した。事例では2目的の順列フローショップスケジューリング問題を取り扱い、提案手法によって得られる結果の妥当性と手法の有効性を確認した。階層的クラスタリングのパラメータ設定による結果への影響を検証するため、複数の類似性評価指標とリンケージ手法を用いて結果を比較した。類似性評価指標やリンケージ手法の選択はデンドログラムの形状が異なるため、代替案の関係性を議論する上では重要な要素であることが分かった。一方で、同じスキーマモデルを用いている場合であれば、同一の代替案に至るビルディングブロックの累積

は同じ経過を辿ることを確認した。明らかになったビルディングブロックの効果を確認したのち、特定の代替案に至る経緯をビルディングブロックの累積として再現できることを確認することで、手法の有効性を確認した。

第5章では、解空間の構造分析のグローバルプロダクトファミリ設計への応用を示した。グローバルプロダクトファミリ設計の数値モデルを構築し、ストラテジックレベルの意思決定に基づく解空間構造を選択構造として定義し、そのスキーマモデルを導入した。グローバルプロダクトファミリ設計は評価指標が多様であることを背景に、多数評価指標問題でのビルディングブロックの解釈法を新たに導入した。意思決定分岐図の各分岐点で二分される2つのクラスタをクラスラベルとしてロジスティック回帰分析を実行し、分類面の係数から2つのクラスタの特徴を説明可能にした。事例では自動車のグローバルプロダクトファミリ設計問題を例に引いて、構築した多数評価指標問題での解釈法によって得られた結果の妥当性を確認した。特定の代替案に至る経緯を解空間の構造分析結果から考察し、その代替案に潜む戦略やその他代替案との関係を考察できることを示した。

第6章では、解空間の構造分析のモジュール化設計問題への応用を示した。Design Structure Matrix(DSM)を用いたモジュール化法を示し、その解空間構造をネットワーク構造として定義した。DSMは包括モデルであり、その評価指標はシステムの性能を詳細に評価するためのものではないことから、代替案の解釈や選定には目的関数を用いることが適当ではないことを受け、解空間の構造の特徴による解釈方法を提案した。スキーマモデルをより詳細に議論するために Averaged Modularization Matrix と呼ばれるネットワークモデルを導入し、それらの違いを議論するための Branching DSM を提案し、Branching DSM を操作することによるモジュール拡張や組み替えのテンプレートを示した。事例では家庭用空調機のモジュール化設計を例に引いて、提案する構造分析法の有効性と結果の妥当性を検証した。階層関係にある複数の Branching DSM から、大局をつかむ意思決定から詳細を決める意思決定へと体系的な手続きに従って部分的なモジュール構成を決定することができることを確認した。ある分岐点で Branching DSM を用いた構造特徴の分析から得られた結果について、DSM のモデル化背景に立ち戻った分析を行うことで、Branching DSM が特定したレバレッジポイントの妥当性を確認した。

第7章では、第4章から第6章での事例を総括し、手法の限界とその展開について論じた。提案した解空間の構造分析法はそれ自体で閉じているようにも見えるものの、多くの前提や仮定のもとで成立しており、適用のための条件と課題を整理し、新ためて提案手法の効果について論じた。更なる手法適用の展開と展望について3つの事例の立ち位置から考察した。

以上より、本論文では最適化計算の外側にある意思決定の局面に焦点を当てて、組合せを伴う解空間の構造分析法を提案し、3つの種別の組合せ最適設計問題を例に手法の応用とその有効性を示した。実際の設計では、要素が組合せ構造をとることは稀ではなく、随所に二者択一の選択が潜んでいる。そのような部分に対して数値最適化による設計支援を行う場合は組合せ最適化問題が定式化され、その解空間の構造は組合せを伴う。第7章で述べたように、本論文で提案した分析法は、

さまざまな組合せ型最適設計問題を扱う上での基礎となるものであり、その適用先は数値例題で示したものに留まらない。さらに、本研究は意思決定者の支援を指向して解空間の構造分析法を提案したものの、最適設計とは「開いた問題」である。解空間の構造を表現するスキーマモデルの選択や代替案の類似度の評価方法、リンケージ手法、ビルディングブロックの解釈方法は問題に個別であり、分析者の意思決定が必要な部分であることに注意されたい。本論文では方法の選定の方針や基準について述べているものの、つまるところ個別問題の特性に依存するため、その応用の将来は、今後の研究課題として認識している。今後さまざまな設計問題に展開していく上では、それぞれの問題の種別に応じてスキーマモデルやその分析法を個別に議論する必要があるものの、本論文の成果は事例として、その基礎として位置付けられると考えている。

参考文献

- [1] 日本機械学会, 機械工学事典 (1997), 丸善.
- [2] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2 (2002), pp. 182–197.
- [3] Zhou, A., Qu, B., Li, H., Zhao, S., Suganthan, P. N. and Zhang, Q., Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm and Evolutionary Computation*, Vol. 1, No. 1 (2011), pp. 32–49.
- [4] Stump, G., Simpson, T. W., Yukish, M. and Bennett, L., Multidimensional visualization and its application to a design by shopping paradigm, in *Proceedings of 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (2002), p. 5622.
- [5] Obayashi, S., Jenong, S., Chiba, K. and Morino, H., Multi-objective design exploration and its application to regional-jet wing design, *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 50, No. 167 (2007), pp. 1–8.
- [6] Ross, A. M. and Hastings, D. E., The tradespace exploration paradigm, in *Proceedings of 5th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2005* (2005), pp. 1706–1718.
- [7] Ulrich, K., The role of product architecture in the manufacturing firm, *Research Policy*, Vol. 24, No. 3 (1995), pp. 419–440.
- [8] Fujimoto, H., Ahmed, A., Iida, Y. and Hanai, M., Assembly process design for managing manufacturing complexities because of product varieties, *International Journal of Flexible Manufacturing Systems*, Vol. 15, No. 4 (2003), pp. 283–307.
- [9] Yenisey, M. M. and Yagmahan, B., Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends, *Omega*, Vol. 45, (2014)pp. 119–135.
- [10] 赤木新介, 設計工学 (上) –新しいコンピュータ応用設計– (1991), コロナ社.
- [11] Simon, H. A., *The Sciences of the Artificial*, third edition (1999), The MIT Press, (第3版邦訳, 稲葉元吉・吉原英樹 (訳), システムの科学 (1999), パーソナルメディア).

- [12] 赤木新介, システム工学設計 –エンジニアリングシステムの解析と計画– (1992), 共立出版.
- [13] Govindan, K., Fattahi, M. and Keyvanshokoo, E., Supply chain network design under uncertainty: A comprehensive review and future research directions, *European Journal of Operational Research*, Vol. 263, No. 1 (2017), pp. 108–141.
- [14] Eppinger, S. D. and Browning, T. R., *Design Structure Matrix Methods and Applications* (2012), MIT Press.
- [15] 茨木俊秀, 福島雅夫, 最適化の手法 (1993), 共立出版.
- [16] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning* (1989), Addison-Wesley Professional.
- [17] Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., Optimization by simulated annealing, *Science*, Vol. 220, No. 4598 (1983), pp. 671–680.
- [18] Konak, A., Coit, D. W. and Smith, A. E., Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering & System Safety*, Vol. 91, No. 9 (2006), pp. 992–1007.
- [19] Simpson, T. W., Peplinski, J. D., Koch, P. N. and Allen, J. K., Metamodels for computer-based engineering design: Survey and recommendations, *Engineering with Computers*, Vol. 17, No. 2 (2001), pp. 129 – 150.
- [20] Pahl, G., Beitz, W., Feldhusen, J. and Grote, K. H., *Engineering Design –A Systematic Approach–*, third edition (2007), Springer, (金田徹, 他 (訳), エンジニアリングデザイン–工学設計の体系的アプローチ, 第3版 (2015), 森北出版).
- [21] Kudo, F., Yoshikawa, T. and Furuhashi, T., A study on analysis of design variables in Pareto solutions for conceptual design optimization problem of hybrid rocket engine, in *Proceedings of 2011 IEEE Congress of Evolutionary Computation (CEC)* (2011), pp. 2558–2562.
- [22] Schütze, O., Vasile, M. and Coello, C. A. C., Computing the set of epsilon-efficient solutions in multiobjective space mission design, *Journal of Aerospace Computing, Information, and Communication*, Vol. 8, No. 3 (2011), pp. 53–70.
- [23] Tanabe, R. and Ishibuchi, H., A review of evolutionary multimodal multiobjective optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 24, No. 1 (2020), pp. 193–200.
- [24] Deb, K. and Tiwari, S., Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization, *European Journal of Operational Research*, Vol. 185, No. 3 (2008), pp. 1062–1087.
- [25] Yue, C., Qu, B. and Liang, J., A multiobjective particle swarm optimizer using ring topol-

- ogy for solving multimodal multiobjective problems, *IEEE Transactions on Evolutionary Computation*, Vol. 22, No. 5 (2018), pp. 805–817.
- [26] Peng, Y. and Ishibuchi, H., Niching diversity estimation for multi-modal multi-objective optimization, in *Proceedings of Evolutionary Multi-Criterion Optimization* (2021), pp. 323–334.
- [27] Arroyo, J. C. and de Souza Pereira, A. A., A grasp heuristic for the multi-objective permutation flowshop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, Vol. 55, No. 5-8 (2011), pp. 741–753.
- [28] Allahverdi, A., A new heuristic for m-machine flowshop scheduling problem with bicriteria of makespan and maximum tardiness, *Computers & Operations Research*, Vol. 31, No. 2 (2004), pp. 157–180.
- [29] Rahimi-Vahed, A., Dangchi, M., Rafiei, H. and Salimi, E., A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, Vol. 41, No. 11 (2009), pp. 1227–1239.
- [30] Allouche, M. A., Aouni, B., Martel, J.-M., Loukil, T. and Rebaï, A., Solving multi-criteria scheduling flow shop problem through compromise programming and satisfaction functions, *European Journal of Operational Research*, Vol. 192, No. 2 (2009), pp. 460–467.
- [31] Afsar, B., Miettinen, K. and Ruiz, F., Assessing the performance of interactive multiobjective optimization methods: a survey, *ACM Computing Surveys*, Vol. 54, No. 4 (2021).
- [32] Fleming, P. J., Purshouse, R. C. and Lygoe, R. J., Many-objective optimization: An engineering design perspective, in *Proceedings of Evolutionary Multi-Criterion Optimization* (2005), pp. 14–32.
- [33] Gannon, T. and Monat, J., What is systems thinking? A review of selected literature plus recommendations, *International Journal of Systems Science*, Vol. 4, (2015)pp. 11–26.
- [34] Chang, P.-C., Chen, M.-H., Tiwari, M. K. and Iquebal, A. S., A block-based evolutionary algorithm for flow-shop scheduling problem, *Applied Soft Computing*, Vol. 13, No. 12 (2013), pp. 4536–4547.
- [35] Spencer, B., Steven, G., John, S., Christopher, T. and Christopher, R., Structural design space exploration using principal component analysis, *Journal of Computing and Information Science in Engineering*, Vol. 20, No. 6 (2020), p. 061014.
- [36] Wang, G. G. and Shan, S., Review of metamodeling techniques in support of engineering design optimization, *Journal of Mechanical Design*, Vol. 129, No. 4 (2006), pp. 370–380.
- [37] Wang, G. G. and Simpson, T. W., Fuzzy clustering based hierarchical metamodeling for

- design space reduction and optimization, *Engineering Optimization*, Vol. 36, No. 3 (2004), pp. 313–335.
- [38] Wang, G. G. and Shan, S., Design space reduction for multi-objective optimization and robust design optimization problems, *SAE Transactions*, Vol. 113, (2004)pp. 101–110.
- [39] Missoum, S., Ramu, P. and Haftka, R. T., A convex hull approach for the reliability-based design optimization of nonlinear transient dynamic problems, *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, No. 29 (2007), pp. 2895–2906.
- [40] Thole, S. P. and Ramu, P., Design space exploration and optimization using self-organizing maps, *Structural and Multidisciplinary Optimization*, Vol. 62, No. 3 (2020), pp. 1071–1088.
- [41] Shan, S. and Wang, G. G., Space exploration and global optimization for computationally intensive design problems: a rough set based approach, *Structural and Multidisciplinary Optimization*, Vol. 28, No. 6 (2004), pp. 427–441.
- [42] Curtis, S. K., Mattson, C. A., Hancock, B. J. and Lewis, P. K., Divergent exploration in design with a dynamic multiobjective optimization formulation, *Structural and Multidisciplinary Optimization*, Vol. 47, No. 5 (2013), pp. 645–657.
- [43] Gan, N. and Gu, J., Hybrid meta-model-based design space exploration method for expensive problems, *Structural and Multidisciplinary Optimization*, Vol. 59, No. 3 (2019), pp. 907–917.
- [44] Frank, C. P., Marlier, R. A., Pinon-Fischer, O. J. and Mavris, D. N., Evolutionary multi-objective multi-architecture design space exploration methodology, *Optimization and Engineering*, Vol. 19, No. 2 (2018), pp. 359–381.
- [45] Ölvander, J., Lundén, B. and Gavel, H., A computerized optimization framework for the morphological matrix applied to aircraft conceptual design, *Computer-Aided Design*, Vol. 41, No. 3 (2009), pp. 187–196, Computer Support for Conceptual Design.
- [46] Pepper, N., Montomoli, F. and Sharma, S., A non-parametric histogram interpolation method for design space exploration, *Journal of Mechanical Design*, Vol. 144, No. 8 (2022), p. 081703.
- [47] Harding, J., Dimensionality reduction for parametric design exploration, in *Proceedings of Advances in Architectural Geometry* (2016).
- [48] Brown, N. C. and Mueller, C. T., Design variable analysis and generation for performance-based parametric modeling in architecture, *International Journal of Architectural Computing*, Vol. 17, (2018)p. 147807711879949.
- [49] Danhaive, R. and Mueller, C. T., Design subspace learning: Structural design space exploration using performance-conditioned generative modeling, *Automation in Construction*,

Vol. 127, (2021)p. 103664.

- [50] Miller, B. and Shaw, M., Genetic algorithms with dynamic niche sharing for multimodal function optimization, in *Proceedings of IEEE International Conference on Evolutionary Computation* (1996), pp. 786–791.
- [51] Goldberg, D. E. and Richardson, J., Genetic algorithms with sharing for multimodal function optimization, in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application* (1987), pp. 41–49, USA, L. Erlbaum Associates Inc..
- [52] Liu, Y., Ishibuchi, H., Nojima, Y., Masuyama, N. and Shang, K., A double-niched evolutionary algorithm and its behavior on polygon-based problems, in *Proceedings of Parallel Problem Solving from Nature – PPSN XV* (2018), pp. 262–273, Springer International Publishing.
- [53] Holland, J. H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (1992), The MIT Press.
- [54] Goldberg, D. E., Korb, B. and Deb, K., Messy genetic algorithms: Motivation, analysis, and first results, *Complex Systems*, Vol. 3, (1989)pp. 493–530.
- [55] Kargupta, H., The gene expression messy genetic algorithm, in *Proceedings of IEEE International Conference on Evolutionary Computation* (1996), pp. 814–819.
- [56] Harik, G. R. and Goldberg, D. E., Linkage learning through probabilistic expression, *Computer methods in applied mechanics and engineering*, Vol. 186, No. 2 (2000), pp. 295–310.
- [57] Chang, P.-C., Huang, W.-H., Wu, J.-L. and Cheng, T., A block mining and re-combination enhanced genetic algorithm for the permutation flowshop scheduling problem, *International Journal of Production Economics*, Vol. 141, No. 1 (2013), pp. 45–55.
- [58] Hsu, C.-Y., Chang, P.-C. and Chen, M.-H., A linkage mining in block-based evolutionary algorithm for permutation flowshop scheduling problem, *Computers & Industrial Engineering*, Vol. 83, (2015)pp. 159–171.
- [59] Obayashi, S., Extraction of design rules from multi-objective design exploration (MODE) using rough set theory, *Fluid Dynamics Research*, Vol. 43, No. 4 (2011), p. 041404.
- [60] Jeong, S., Sakaki, T., Chae, S., Yee, K. and Aoyama, T., Design exploration of helicopter blades for HSI noise and aerodynamic performance, *Transaction of the Japan Society for Aeronautical and Space Sciences*, Vol. 54, No. 184 (2011), pp. 153–159.
- [61] Richardson, T., Nekolny, B., Holub, J. and Winer, E. H., Visualizing design spaces using two-dimensional contextual self-organizing maps, *AIAA Journal*, Vol. 52, No. 4 (2014),

pp. 725–738.

- [62] Winer, E. H. and Bloebaum, C. L., Visual design steering for optimization solution improvement, *Structural and Multidisciplinary Optimization*, Vol. 22, No. 3 (2001), pp. 219–229.
- [63] Stump, G., Lego, S., Yukish, M., Simpson, T. W. and Donndelinger, J. A., Visual steering commands for trade space exploration: User-guided sampling with example, *Journal of Computing and Information Science in Engineering*, Vol. 9, No. 4 (2009).
- [64] Unal, M., Warn, G. P. and Simpson, T. W., Quantifying the shape of Pareto fronts during multi-objective trade space exploration, *Journal of Mechanical Design*, Vol. 140, No. 2 (2017).
- [65] Everitt, B. S., Landau, S., Leese, M. and Stahl, D., *Cluster Analysis*, 5th edition (2011), John Wiley & Sons.
- [66] Hastie, T., Tibshirani, R. and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second edition (2009), Springer.
- [67] Gower, J. C. and Legendre, P., Metric and Euclidean properties of dissimilarity coefficients, *Journal of Classification*, Vol. 3, No. 1 (1986), pp. 5–48.
- [68] Sneath, P. H. A., *Numerical Taxonomy: The Principles and Practice of Numerical Classification* (1973), Freeman.
- [69] Ward, J. H., Jr., Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association*, Vol. 58, No. 301 (1963), pp. 236–244.
- [70] Lance, G. N. and Williams, W. T., A general theory of classificatory sorting strategies: 1. hierarchical systems, *The Computer Journal*, Vol. 9, No. 4 (1967), pp. 373–380.
- [71] 宮本定明, クラスター分析入門-ファジイクラスタリングの理論と応用 (1999), 森北出版.
- [72] Aghezzaf, E.-H. and Landeghem, H. V., An integrated model for inventory and production planning in a two-stage hybrid production system, *International Journal of Production Research*, Vol. 40, No. 17 (2002), pp. 4323–4339.
- [73] Boukef, H., Benrejeb, M. and Borne, P., A proposed genetic algorithm coding for flow-shop scheduling problems, *International Journal of Computers, Communications & Control (IJCCC)*, Vol. 2, No. 3 (2007), pp. 229–240.
- [74] Xu, J. and Zhou, X., A class of multi-objective expected value decision-making model with birandom coefficients and its application to flow shop scheduling problem, *Information Sciences*, Vol. 179, No. 17 (2009), pp. 2997–3017, Copulas, Measures and Integrals.
- [75] Lei, D., Multi-objective production scheduling: A survey, *The International Journal of Advanced Manufacturing Technology*, Vol. 43, No. 9 (2008), p. 926.

- [76] Nagar, A., Haddock, J. and Heragu, S., Multiple and bicriteria scheduling: A literature survey, *European Journal of Operational Research*, Vol. 81, No. 1 (1995), pp. 88–104.
- [77] Toi, M., Kaihara, M., Nomaguchi, Y. and Fujita, K., Application of dendrogram-directed design space exploration to bi-objective permutation flow-shop scheduling problem, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 16, No. 5 (2022), pp. 22–00221.
- [78] Taillard, E., Benchmarks for basic scheduling problems, *European Journal of Operational Research*, Vol. 64, No. 2 (1993), pp. 278–285, Project Management and Scheduling.
- [79] Ishibuchi, H. and Murata, T., A multi-objective genetic local search algorithm and its application to flowshop scheduling, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 28, No. 3 (1998), pp. 392–403.
- [80] Wang, R., An improved nondominated sorting genetic algorithm for multiobjective problem, *Mathematical Problems in Engineering*, Vol. 2016, (2016)p. 1519542.
- [81] 藤田喜久雄, 製品系列統合化設計における最適性と最適化法に関する研究 : 第 1 報, 最適化問題の構造と様相, 日本機械学会論文集 C 編, Vol. 68, No. 666 (2002), pp. 675–682.
- [82] Pine, B. J., *Mass Customization: the New Frontier in Business Competition* (1993), Boston: Harvard Business School Press.
- [83] Jiao, J. R., Simpson, T. W. and Siddique, Z., Product family design and platform-based product development: A state-of-the-art review, *Journal of Intelligent Manufacturing*, Vol. 18, No. 1 (2007), pp. 5–29.
- [84] 藤田喜久雄, 坂口久仁, 製品系列統合化設計における最適性と最適化法に関する研究 (第 2 報, モジュール組み合わせの最適化法), 日本機械学会論文集 C 編, Vol. 68, No. 666 (2002), pp. 683–691.
- [85] 藤田喜久雄, 吉田寛子, 製品系列統合化設計における最適性と最適化法に関する研究 (第 3 報, システム属性とモジュール組合せの同時最適化法), 日本機械学会論文集 C 編, Vol. 68, No. 668 (2002), pp. 1329–1337.
- [86] Lamothe, J., Hadj-Hamou, K. and Aldanondo, M., An optimization model for selecting a product family and designing its supply chain, *European Journal of Operational Research*, Vol. 169, No. 3 (2006), pp. 1030–1047.
- [87] Fujita, K., *Global product family design: Simultaneous optimal design of module commonalization and supply chain configuration* (2014), Springer.
- [88] Fujita, K., Amaya, H. and Akai, R., Mathematical model for simultaneous design of module commonalization and supply chain configuration toward global product family, *Journal of Intelligent Manufacturing*, Vol. 24, No. 5 (2013), pp. 991–1004.

- [89] Nomaguchi, Y., Osaki, D. and Fujita, K., Optimal design and robustness assessment of product family considering quantity discounts in supply chain, in *Proceedings of ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. Volume 2B: 43rd Design Automation Conference (2017), pp. DETC2017–67688.
- [90] Toi, M., Sawai, K., Nomaguchi, Y. and Fujita, K., Strategic-level robust optimal design method of product family and supply chain network, in *Proceedings of ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. Volume 2B: 44th Design Automation Conference (2018), pp. DETC2018–85242.
- [91] Simchi-Levi, D., Kaminsky, P. and Simchi-Levi, E., *Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies* (1999), Irwin Professional Pub.
- [92] Fujita, K., Nasu, K., Ito, Y. and Nomaguchi, Y., Global product family design: Multi-objective optimization and design concept exploration, in *Proceedings of ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. Volume 3: 38th Design Automation Conference, Parts A and B (2012), pp. DETC2012–70858.
- [93] Vidal, C. J. and Goetschalckx, M., Strategic production-distribution models: A critical review with emphasis on global supply chain models, *European Journal of Operational Research*, Vol. 98, No. 1 (1997), pp. 1–18.
- [94] Nelder, J. A. and Mead, R., A simplex method for function minimization, *The Computer Journal*, Vol. 7, No. 4 (1965), pp. 308–313.
- [95] Tsumoto, R., Fujita, K., Nomaguchi, Y., Yamasaki, S. and Yaji, K., Classification-Directed conceptual structure design based on topology optimization, deep clustering, and logistic regression, in *Proceedings of ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. Volume 3A: 48th Design Automation Conference (DAC) (2022), pp. DETC2022–88548.
- [96] Bishop, C., *Pattern Recognition and Machine Learning* (2012), Springer.
- [97] Shaik, A., Rao, V. and Rao, C., Development of modular manufacturing systems—a review, *The International Journal of Advanced Manufacturing Technology*, Vol. 76, (2015)pp. 789–802.
- [98] Watanabe, S., Hiroyasu, T. and Miki, M., Neighborhood cultivation genetic algorithm for multi-objective optimization problems, in *Proceedings of 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL-2002)* (2002), pp. 198–202.

- [99] Simpson, T. W., Jiao, J. R., Siddique, Z. and Hölttä-Otto, K. E., *Advances in Product Family and Product Platform Design Methods & Applications* (2014), Springer.
- [100] Hölttä-Otto, K. and Weck, de O., Degree of modularity in engineering systems and products with technical and business constraints, *Concurrent Engineering*, Vol. 15, No. 2 (2007), pp. 113–126.
- [101] Engel, A. and Browning, T. R., Designing systems for adaptability by means of architecture options, *Systems Engineering*, Vol. 11, (2008).
- [102] Suh, E. S. and Kott, G., Reconfigurable parallel printing system design for field performance and service improvement, *Journal of Mechanical Design*, Vol. 132, No. 3 (2010).
- [103] Simpson, T. W., Product platform design and customization: Status and promise, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 18, No. 1 (2004), pp. 3–20.
- [104] Sosa, M., Eppinger, S. and Rowles, C., Identifying modular and integrative systems and their impact on design team interactions, *Journal of Mechanical Design*, Vol. 125, No. 2 (2003).
- [105] Sawai, K., Nomaguchi, Y. and Fujita, K., Case study of extended product architecture design for modularization reflecting customer needs of industrial robots, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 11, No. 4 (2017), p. JAMDSM0050.
- [106] Chen, L. and Li, S., Analysis of decomposability and complexity for design problems in the context of decomposition, *Journal of Mechanical Design*, Vol. 127, No. 4 (2004), pp. 545–557.
- [107] Otto, K., Hölttä-Otto, K., Simpson, T. W., Krause, D., Ripperda, S. and Ki Moon, S., Global views on modular design research: Linking alternative methods to support modular product family concept development, *Journal of Mechanical Design*, Vol. 138, No. 7 (2016).
- [108] Kannan, R., Vempala, S. and Vetta, A., On clusterings: Good, bad and spectral, *J. ACM*, Vol. 51, No. 3 (2004), pp. 497–515.
- [109] Schaeffer, S. E., Survey: Graph clustering, *Computer Science Review*, Vol. 1, No. 1 (2007), pp. 27–64.
- [110] Sosa, M. E., Eppinger, S. D. and Rowles, C. M., A network approach to define modularity of components in complex products, *Journal of Mechanical Design*, Vol. 129, No. 11 (2007), pp. 1118–1129.
- [111] Yu, T.-L., Yassine, A. A. and Goldberg, D. E., An information theoretic method for developing modular architectures using genetic algorithms, *Research in Engineering Design*,

Vol. 18, No. 2 (2007), pp. 91–109.

- [112] Hölttä-Otto, K., Chiriac, N. A., Lysy, D. and Suh, E. S., Comparative analysis of coupling modularity metrics, *Journal of Engineering Design*, Vol. 23, No. 10-11 (2012), pp. 790–806.
- [113] Toi, M., Nomaguchi, Y. and Fujita, K., Dendrogram-directed design space exploration for DSM- based product modularization, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 16, No. 5 (2022), pp. 21–00343.
- [114] Akao, Y., *Quality Function Deployments* (1990), Productivity Press.
- [115] Gunnar, E., Modular function deployment(MFD), support for good product structure creation, in *Proceedings of 2nd WDK Workshop on Product Structuring* (1996), pp. 13–16.
- [116] Stone, R. B., Wood, K. L. and Crawford, R. H., A heuristic method for identifying modules for product architectures, *Design Studies*, Vol. 21, No. 1 (2000), pp. 5–31.
- [117] Gao, F., Xiao, G. and Simpson, T. W., Identifying functional modules using generalized directed graphs: Definition and application, *Computers in Industry*, Vol. 61, No. 3 (2010), pp. 260–269.

謝辞

本論文の作成にあたり、多くの方々にご指導、ご鞭撻を賜りました。

大阪大学大学院工学研究科機械工学専攻 藤田喜久雄教授には、指導教員として、研究姿勢から発表や論文として成果をまとめる技術まで、終始御懇切な御指導および御助言を賜りました。また、研究の研究活動だけでなく、在学中のさまざまな活動についてもあたたかいご指導を賜りました。ここに心より深く感謝の意を表します。今後も先生に教えていただいたことを糧に精進してまいります。

大阪大学大学院工学研究科機械工学専攻 高谷裕浩教授、小林英樹教授、井野秀一教授、野間口大准教授には、副査をお引き受け頂いたとともに、本論文を作成するにあたり、御懇切な御助言を賜りました。ここに深く感謝いたします。あわせて、野間口大准教授には、日頃の研究活動や学会発表でも多くのご助言を賜り、研究の細部に関して多角的にご議論いただきましたことに感謝いたします。

大阪大学大学院工学研究科機械工学専攻 矢地謙太郎助教、早稲田大学大学院情報生産システム研究科 山崎慎太郎教授には、本研究を進めるにあたり、随所において有益な御意見を頂きました。ここに深く感謝いたします。あわせて、矢地謙太郎助教には、日頃の雑談や学会を通して、研究活動の魅力や研究者としてのキャリア、生活について多くのご助言をいただきました。ここに深く感謝いたします。

本論文で取り上げた家庭用空調機の例題は、匿名の企業から御提供賜りました。関係各位には、本研究への御協力および御支援をいただきましたことに深く感謝申し上げます。

大阪大学工学部応用理工学科機械工学科目での卒業論文研究や大阪大学工学研究科機械工学専攻での修士論文への取り組みとして、本研究に参画いただきました、開原将貴氏、小林正秀氏に感謝いたします。ならびに、研究を進める上でのディスカッションに度々加わり、有益な御意見を提示いただきました澤井伽奈博士、丸山峻博士、津本燎氏に感謝いたします。澤井伽奈博士、丸山峻博士には博士課程での悩みについて何度もご相談に乗って頂きました。くじけず研究を進めてくれたことも、ご助言があったこそです。ここに深く感謝いたします。

機械工学専攻統合設計学系設計工学領域（藤田・野間口研究室）の各位には、研究のみならず、研究室でのさまざまな事柄において御協力、御助言をいただきました。皆様と過ごした日々はかけがえのないもので、みなさまのお陰で研究生活を楽しく過ごすことができました。ここに深く感謝

申し上げます。

大阪大学超域イノベーション博士課程プログラムの7期生各位には、専門分野は異なるものの、志を共にする仲間として大きな刺激を受け、共に切磋琢磨できたことは大きな励みになりました。深く感謝いたします。あわせて、大阪大学大学院工学研究科機械工学専攻統合設計学系サステナブルシステムデザイン学領域 川口太郎氏には、多くのご助言、激励をいただきました。ここに感謝いたします。

本研究の一部は、JST 次世代研究者挑戦的研究プログラム JPMJSP2138 の支援を受けたものです。

最後になりましたが、これまでの研究生活を支えてくれた家族と友人に、深く感謝いたします。

研究論文・発表リスト

学術誌掲載論文

- (1) Toi, M., Nomaguchi, Y. and Fujita, K., Dendrogram-directed design space exploration for DSM-based product modularization, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 16, No. 5 (2022), p. 21-00343.
- (2) 戸井誠人, 野間口大, 藤田喜久雄, 多目的組合せ型設計問題における意思決定のための階層的クラスター分析による解空間の構造解析, 日本機械学会論文集, Vol. 88, No. 914 (2022), p. 22-00173.
- (3) Toi, M., Kaihara, M., Nomaguchi, Y. and Fujita, K., Application of dendrogram-directed design space exploration to bi-objective permutation flow-shop scheduling problem, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 16, No. 5 (2022), p. 21-00221.

国際会議論文

- (4) Toi, M., Sawai, K., Nomaguchi, Y. and Fujita, K., Strategic-level robust optimal design method of product family and supply chain network, *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2018)* (2018), Paper No. DETC2018-85242.
- (5) Toi, M., Sawai, K., Kobayashi, M., Nomaguchi, Y. and Fujita, K., Integrative design of product architecture and assembly process plan, *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2019)* (2019), Paper No. IDETC2019-97338.
- (6) Toi, M., Nomaguchi, Y. and Fujita, K., Structural design exploration for product architecture design, *ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2020)* (2020), Paper No. IDETC2020-22599.
- (7) Toi, M., Kaihara, M., Nomaguchi, Y. and Fujita, K., An application of structural design exploration to multi-objective permutation flow shop scheduling problem, *International Symposium on Scheduling 2021 (ISS2021)* (2021), Paper No. TS2-3.

国内会議論文

- (2)' 戸井誠人, 野間口大, 藤田喜久雄, 多目的組合せ型設計問題における意思決定のための階層的クラスター分析による解空間の構造解析, 日本機械学会 第 31 回設計工学・システム部門講演会 (2021), Paper No. 1104. [改訂版が学術誌に (2) として掲載されている] .
- (8) 戸井誠人, 澤井伽奈, 野間口大, 藤田喜久雄, 製品系列設計とサプライチェーンネットワーク計画の統合のためのストラテジックレベルロバスト最適設計方法論の構築, 日本機械学会 第 28 回設計工学・システム部門講演会 (2018), Paper No. 1102.
- (9) 戸井誠人, 澤井伽奈, 小林正秀, 野間口大, 藤田喜久雄, 製品アーキテクチャと組立プロセスの統合的設計支援フレームワーク, 日本機械学会 第 29 回設計工学・システム部門講演会 (2019), Paper No. 2316.
- (10) 戸井誠人, 野間口大, 藤田喜久雄, 製品アーキテクチャのための解空間の構造分析に関する研究, 日本機械学会 第 30 回設計工学・システム部門講演会 (2020), Paper No. 2301.