

# アーキテクチャ理論に基づく新製品開発 のモデリングとその解法に関する研究

姜 東暁

秋田県立大学大学院  
システム科学技術研究科  
総合システム科学専攻

2014年3月20日

# 概要

競争的ポジショニング戦略で明らかなように、マーケット・フォロワーと違いマーケット・リーダーである企業では市場シェアを維持し続けるために利益に結び付けられる新製品の開発・設計が重要な役割を担っていると考えられる。

製品設計のアイデアの良さに関わらず、利益につながる製品設計案か、そうでない製品設計案か、という判断は製品設計の評価と呼ばれる。しかし製品設計案の評価は、中小企業はもちろん、大手企業でも熟練者の暗黙知や経験に大きく依存していることが一般的である。そのため、暗黙知に頼らずに熟練した設計者だけでなく経験の浅い設計者に対しても普遍的な議論を行える手法が必要である。それを実現するためには、誰でも扱える知識に還元しなければならない。先行研究として、製品設計における数理評価アプローチの構築は活発に議論されている。製品設計段階において、主に VE (Value Engineering)、QFD (Quality Function Deployment)、DSM (Design Structure Matrix) と LCC (Life Cycle Cost) の提案がある。一方、製品生産段階において、品質工学における田口メソッド及び類似部品加工法も含めて、多角度から製品設計案の評価手法を数理的に定義されている。しかし、既存製品または機能間のすりあわせ(インテグラル)による新製品を作り出そうとしているとき、上の評価手法のいずれもこのインテグラルの効果を取り入れた製品設計案を評価することができない。

この不足を踏まえて、製品アーキテクチャ理論に基づいて、本研究では製品設計案に対する定量的な見方を提供する数理評価アプローチを提案することを目的とする。製品アーキテクチャは「製品の基礎的な設計思想」の一つである。製品分解に基づいて、機能を代表するデバイス間のつながり状態をインターフェースで表し、製品の全般的なインターフェースの配置は製品設計案(製品アーキテクチャーともいい)として定義される。異なる製品設計案の総合収益が違うことは評価値で表される。一番良い評価値をもたらす製品設計案は最適設計として選ばれる。本学位論文では以下の流れで製品設計案の評価モデルの構築と実用性の考慮でモデルの数理解析を取り組む。

第1章「はじめに」の部分では、研究背景、研究目的について、また、主な現存の製品設計の数理手法 QFD と DSM のメリットとデメリットを紹介する上で、製品アーキテクチャ理論とその分類、モジュール型(くみあわせ)、インテグラル型(すりあわせ)およびハイブリッド型の簡単な説明を展開する。

第2章「製品設計のための最適化モデルの提案」の部分では、製品設計案の評価問題の定式化を行う。任意2つの機能デバイスをつなぐインターフェースそのものの単独効用に加えて、従来研究に考えられなかったインターフェース同士間の相互効用も考慮に入れたバイナリ2次組合せ最適化モデルを提案する。モデルパラメータである単独効用と相互効用の値が与えられた上で、最大の総合収益に対応する製品設計案を最適解として選ぶ。この提案モデルの新規性として、モジュール型の設計案のみでなく、インテグラル型も評価することができるようになる。

第3章「分枝限定法による2次モデルの厳密解析」の部分では、第2章で提案する2次組合せ最適化モデルの実用性を検証するためにモデルの厳密解析法を提案する。提案モデルの非線形項部分の特徴により、目的関数の凸性を保証できないため、従来の勾配法で得られた解が最適解である保証はない。そのため本章ではこの2次モデルにおける非線形項と等価である線形項及び不等式制約を導入し、2次組合せ最適化モデルの1次等価変換モデルを提案する。さらに等価性を示す上で、1次等価変換モデルを解析する分枝限定法 (Branch and Bound, BAB) を提案する。数値実験により列挙法より提案手法では製品設計の最適化解析を実用レベルに向上したことを検証する。

第4章「平均場近似法による製品設計の大規模問題の近似解析」の部分では、第3章で提案した厳密解析法でも解けない規模の製品設計問題の近似解法を考案する。統計力学における多体システムの状態を確率的に求められる平均場近似法 (Mean Field Approach, MFA) を援用する場合、一つのインターフェースの試行分布の平均値は、自身の単独効用とほかのすべての近傍であるインターフェースからもたらす相互効用の総和によって決まるものである。平均場近似法ではモデルの計算量が  $O(2^N)$  から  $O(N^2)$  へ低減することを期待できる。数値実験により提案手法では機能デバイス数  $M = 100$ 、インターフェース数  $N = 4950$  のサイズまでの近似解析を行って提案手法の有効性を示す。

第5章「製品設計の組合せ最適化モデルに適用する一点変動近傍アルゴリズムの提案」の部分では、モデルの特徴に合わせて、暫定状態の情報を用いた問題専用の高速の一点変動近傍法 (One-Bit- Variant Neighborhood Algorithm) を援用するメタヒューリスティクスアルゴリズムを提案する。数値実験によって提案手法の計算量を  $O(2^N)$  から  $O(N^2)$  へ低減できることを証明する。

第6章「ボルツマンマシンによるモデルパラメータの推定解析」の部分では、第2章で提案する製品設計の最適化モデルのパラメータの推定解析を行う。モデルのパラメータ (インターフェースの単独効用とインターフェース間の相互効用) の値が正規分布に従ってランダムに与えられている。それは設計者の暗黙知によってパラメータを精確に定めることが簡単ではないためである。本章では設計現場で提供しやすい製品設計案とそのカテゴリ評価値という観測データを利用して、パラメータの推定モデルを提案する。また、パラメータの数より少ない観測データしか取らない場合を勘案し、ボルツマンマシンを用いて学習誤差が0の下で汎化誤差を最小にする推定問題を解いた。数値実験によって提案手法の有効性を検証する。

第7章「Pメジアン法による最適解のクラスタリング」においては次の議論を行う。単一生産拠点での生産能力は限界があり、全体としてのシステムの効率化を考慮して、多くの機能デバイスからなる完成品を競争的な中間製品へ分割することも必要であるが、先行研究ではこの問題に対して解答することができていない。そこで本章は、機能情報のある経験則の解決策とクラスタリング手法の一つであるPメジアン法 (p-median) を用いて、クラスタ内部のインターフェース間のつながり密度ができるだけ大きく、クラスタ間のつながり密度ができるだけ小さくなる中間製品を導出するアーキテクチャ理論を改良する。

第8章「提案する製品設計アプローチの有効性を検証するための数値例」の部分では、小規模

の実例を用いて、製品設計の2次組合せ最適化モデルの構築（第2章の内容）、ボルツマンマシンによるモデルパラメータの推定解析（第6章の内容）と推定したパラメータに基づいて分枝限定法による2次モデルの厳密解析（第3章の内容）を実施した。結果として最適解とみられる実状態を再現したことが分かる。

第9章はまとめと今後の課題について述べている。今後の課題について、まず、本論文ではモデルのパラメータとなる単独効用と相互効用を与えられたもとで1次等価変換モデルの性能について議論してきたが、実用に向けて、利益、コストなどの指標を用いて単独効用と相互効用の定義を特定する必要もあるだろう。次は、本論文で議論した単独効用と相互効用の値は互いに独立で正規分布に従うとしたが、しかし現実の状況を考えると、異なる分布に従う可能性がある。そのため、この議論に従って提案モデルを改善する必要がある。また、製品設計の従来研究QFD,DSMなどと比較して本研究の実用性と優位さを検証することも重要である。

本学位論文では、製品アーキテクチャ理論に基づいて提案した製品設計アプローチを導入することによって、人のノウハウよりもっと広い範囲の評価が可能になり、暗黙知に頼らずに熟練した設計者だけでなく経験の浅い設計者に対しても普遍的な議論が可能となる。本研究で提案する製品設計アプローチが設計者の意思決定の支援システムの一つとして貢献し、試作品を利益に結び付ける可能性を高めることが期待できる。

# 目次

第1章	はじめに	9
1.1	研究背景	9
1.2	研究目的	9
1.3	主な現存の製品設計の数理手法	10
1.3.1	QFD(Quality Function Deployment)	11
1.3.2	DSM(Design Structure Matrix)	11
1.4	製品アーキテクチャ理論	12
1.4.1	製品アーキテクチャとは	12
1.4.2	アーキテクチャ分類	13
第2章	製品設計のための最適化モデルの提案	15
2.1	問題定義	15
2.2	製品設計問題のモデリング化	15
2.3	バイナリ2次組合せ最適化モデル	17
第3章	分枝限定法による2次モデルの厳密解析	19
3.1	1次等価変換モデルの提案	19
3.2	1次等価変換モデルの最適化解析	21
3.3	数値実験による提案法の評価	24
3.3.1	実験設計	24
3.3.2	分枝限定法の能力限界	24
3.3.3	行列 $G$ の特性と探索コスト	24
3.3.4	$\alpha, \beta, \alpha', \beta'$ と探索効率	26
3.4	まとめ	26
第4章	平均場近似法による製品設計の大規模問題の近似解析	29
4.1	製品設計のコスト最小化問題	29
4.2	統計力学に基づく平均場近似	33
4.2.1	統計力学アプローチ (Statistical Mechanical Approach)	33
4.2.2	最適化 (Optimization)	34
4.2.3	平均場近似 (Mean Field Approximation)	35

4.3	数値実験	36
4.3.1	機能デバイス数 $M = 4$ から $M = 7$ の小型例	36
4.3.2	均一 (Homogeneous) の例	38
4.3.3	機能デバイス数 $M = 100$ の大型例	39
4.3.4	応用例	40
4.4	まとめ	42
<b>第5章</b>	<b>製品設計の組合せ最適化モデルに適用する一点変動近傍アルゴリズムの提案</b>	<b>45</b>
5.1	製品設計の2次組合せ最適化モデルの特徴	45
5.2	一点変動近傍法 (One-Bit-Variant Neighborhood Algorithm)	49
5.3	数値実験	50
5.3.1	機能デバイス $M = 5, 10, 15$ の小規模の数値例	50
5.3.2	機能デバイス $M = 25, 40$ の中型規模の数値例	52
5.3.3	機能デバイス $M = 70, 100$ の大規模の数値例	53
5.4	まとめ	53
<b>第6章</b>	<b>ボルツマンマシンによるモデルパラメータの推定解析</b>	<b>55</b>
6.1	デザインとカテゴリ評価による推定モデルの提案	55
6.2	推定モデルのボルツマンマシン学習	58
6.2.1	ボルツマンマシンの定義	58
6.2.2	ボルツマンマシンの最適化問題	58
6.2.3	ボルツマンマシンの平衡状態の実現	60
6.2.4	確率伝搬法	61
6.2.5	サンプルデータ $S$ の生成	64
6.3	推定したパラメータを製品設計モデルに適用するパラメータへの転換	66
6.4	数値実験	66
6.4.1	実験設計	66
6.4.2	小規模の不均一の例	67
6.4.3	小規模の均一の例	68
6.4.4	大規模の均一の例	68
6.5	まとめ	69
<b>第7章</b>	<b>Pメジアン法による最適解のクラスタリング</b>	<b>71</b>
7.1	中間製品のクラスタ識別モデル	71
7.1.1	機能情報のある経験則のクラスタリング	72
7.2	P-median法による中間製品のクラスタ識別モデル	76
7.2.1	整数計画問題のP-median法	77

	7
7.3 数値実験 . . . . .	79
7.3.1 7個の機能デバイスを持つ懐中電灯の例 . . . . .	79
7.3.2 16個の機能デバイスを持つカーエアコンの例 . . . . .	82
7.3.3 非対称の隣接行列を持つ数値例 . . . . .	83
7.4 まとめ . . . . .	84
<b>第8章 提案する製品設計アプローチの有効性を検証するための数値例</b>	<b>87</b>
8.1 実験設計 . . . . .	87
8.2 実験結果 . . . . .	89
<b>第9章 まとめと今後の課題</b>	<b>91</b>





# 第1章 はじめに

## 1.1 研究背景

製品とは多様な顧客ニーズを満足でき、市場の利益に結びつける機能を有形な機能デバイスで具体化するものである [1]。どんな機能を選び、機能の依存関係をいかに機能デバイスのレイアウトで構築するのが製品設計であると考えられる。競争的ポジショニング戦略で明らかのように、マーケット・フォロワーと違いマーケット・リーダーである企業においては市場シェアを維持し続けるために、利益に結び付ける新製品の開発・設計が重要な役割を担っていることが分かる [1][2][3]。

製品設計案のアイデアの良さに関わらず、利益につながる製品設計案か、そうでない製品設計案か、という判断は製品設計の評価と呼ばれる。例えば、製品の目標機能を達成できるか、スペースの小型化が実現されるかなどが製品設計案の評価に影響する要素の1つである。しかし、製品設計案の評価は、中小企業はもちろん、大手企業でも熟練者の暗黙知や経験に大きく依存していることが一般的である。例として、本体が壊れにくい冷蔵庫の丈夫さを追求する設計を考えた場合、冷蔵庫の丈夫さは市場の利益に結びつかないパフォーマンスの一つである。経験の浅い設計者はそれを適正に評価できない恐れがある。もう1つは、新技術の導入により、熟練した設計者が新しい技術に対して詳しくないため、製品設計を適正に判断できない可能性が高い [4]。

一方、先行研究により、製品設計における数理評価アプローチの構築は活発に議論されているが、既存製品または機能間のインテグラル（すりあわせ）による新製品を作り出そうとしているとき、既存の評価手法のいずれもインテグラルを取り入れた製品設計案の効果を評価することができない。そのため、インテグラルを取り入れる製品設計案に対する定量的な見方を提供する数理評価アプローチを提案することが望まれる。

## 1.2 研究目的

本論文は詳細設計を展開する前の概念段階の製品設計を研究対象とする。熟練した設計者の暗黙知/経験に頼った設計は限定的な検討しか行えないものが、数理評価アプローチを導入することによって、暗黙知に頼らずに熟練した設計者だけでなく経験の浅い設計者に対してもより広い範囲の検討が可能になった。

製品設計における数理評価モデルの構築に関する従来の研究について、たとえばバリューエンジニアリング (VE) や品質機能展開 (QFD)、設計構造マトリクス (DSM)、およびライフサイクルコスト (LCC) などがある。これらの方法では多角度から製品設計を数理的に定義している。

しかしながら、既存製品または機能間のインテグラル（すりあわせ）による新製品を作り出そうとしているとき、上の手法のいずれもインテグラルを取り入れた製品設計を評価することには不十分である。この不足を踏まえて、製品アーキテクチャ理論に基づく製品設計を評価する手法が導入される [5][6][7][8][9][10][11][12]。製品アーキテクチャは「製品の設計思想」の一つである [1]。製品分解に基づいて、機能を代表するデバイス間のつながり状態をインターフェースで表し、製品の全般的なインターフェースの意思決定は製品アーキテクチャ（製品設計案とも呼ぶ）であると定義される。既存製品または機能間のインテグラルをインターフェースの再配置で表すことができる。一つのインターフェースの配置の優劣は総合性能の評価値で計算できる。そして、本論文において、すべてのインターフェースの可能配置の中、一番良い総合性能をもたらすのが最適な製品設計として考えられる。これを実現することが本論文の目的である。

具体的に、既存の機能情報に加えて、新の機能情報を盛り込もうとしている新製品設計問題を考える。まず、既存製品の機能と新しい要素を製品レベルからサブシステムレベルへ分解し、さらにサブシステムを機能デバイスへ分解する。次は、2つの機能デバイスをつなぐインターフェースの意思決定を変数にしてすべてのインターフェースの配置を数理的に定式化する。提案するモデルでは上述のインテグラルを取り入れた新製品設計を評価できるようになる。また、このモデルの数理解析（厳密解析と大規模問題の近似解析）及びモデルのパラメータ推定解析などの内容も本論文の中に議論される。

本研究の貢献は、提案手法では暗黙知に頼らずに熟練した設計者だけでなく経験の浅い設計者に対しても製品アーキテクチャ理論に基づく製品設計の普遍的な議論の実現を可能とする最初のステップである。

### 1.3 主な既存の製品設計の数的手法

従来の研究により、製品設計の数値評価モデルの構築は活発に議論されている。一番早いのが GE の L.D. マイルズにより VE(Value Engineering) の観点から「価値=満足度/コスト」という簡単なモデルで製品の「価値」をその「機能がもたらす満足度」と「コストの妥当さ」の比率で表し、価値を向上させるために満足度とコストのバランスを調整することを目的とする評価モデルである [13]。品質と機能を体系的に分析して管理できるために、VE に基づいて赤尾と水野により QFD(Quality Function Deployment) が提案されてきた。QFD における HOQ(House of Quality) では、言語変数 (linguistic variable) で表す顧客ニーズから変換した品質特徴と機能間の依存関係、及び機能同士の相互影響を転化値で記録し、多様な顧客ニーズを最大限に満足しようとする機能の優先度順序を解析した [14][15]。また、製品分解に基づいて、機能デバイスの最良レイアウトを評価できる DSM(Design Structure Matrix) が MIT により提案されている [16][17][18]。DSM により機能を反映するデバイス間の依存関係を 5 段階スコアリングで表し、さらにクラスタリング技術でデバイス同士をグループ化する (モジュール化と呼ぶ) ことで、既存製品のアップグレードを実現できるようになった。さらに、この 20 年間、循環型社会の実現に関わる話しが盛んでき

て、製品設計においては、目標機能の達成のみではなく製品のライフサイクルの経済性も製品設計の優劣の評価指標の1つとなってきた。製品のライフサイクルの総コスト(LCC)を段階別(設計, 生産, メンテナンスとリサイクル)に定着し分析する評価モデルも提案される[19][20]。

一方、製品生産段階(manufacturing process)において2つの数理評価手法が存在する。一つは、品質工学(quality engineering)において、製品出荷後に目標品質からの偏位を表す損失関数による要求品質を最適に設定する田口メソッド法である[21]。もう一つは、製品設計段階のモジュール化を扱うDSMと近いグループテクノロジーGT(Group Technology)技術である[22][23]。GT技術は、製品の種類に関わらず機能デバイスには共通性のあるものを一つのグループ(part family)に分類して加工する手法である。さらに、製品設計の段階において主な2つの数理手法を紹介する。

### 1.3.1 QFD(Quality Function Deployment)

VEに基づいて赤尾と水野によりQFD(Quality Function Deployment)が提案されてきた[14][15]。QFDにおけるHOQ(House of Quality)を用いて、言語変数(linguistic variable)で表す顧客ニーズと機能間の影響関係、及び機能同士の相互作用を転化値で記録し、多様な顧客ニーズの満足度を最大化しようとする機能の優先度順序を解析した。しかし、QFDにより得られた結果は設計ルール of 優先度順序であり、最適な製品設計そのものではない。また、設計ルールの優先度順序から間接的に得られる製品設計の総合収益も示されていないため、製品設計の優劣を比較することができない。

### 1.3.2 DSM(Design Structure Matrix)

処理の簡単さ及び並行処理の効率性という二つの利点があるので、製品の分解で複雑な製品全体を分析することが容易になる。図1.1のように一つの製品をサブシステムレベルへ分解し、さらに単一機能のデバイスレベルまでに分解する。さらに機能デバイス間の依存関係を表す指標を用いて、2次元のDSMを作成する。MITにより、依存度は四つの基準(材料, 空間, 情報とエネルギー)で表現され、それぞれの基準は5段階(-2, -1, 0, +1, +2)に分けて評価されている[16][17][18]。表1.1のスコアリングは機能デバイス間のインターフェースが存在しているべきかどうか及びその程度について表す[16]。例えば本研究によって分解した懐中電灯とナイトライトという二つの機能を持っている多機能懐中電灯のDSMを図1.2に展示される。行と列には対応な機能を並んで、セルの中で左上空間の近さ, 右上エネルギーの伝達, 左下情報の流れ, 右下材料の共有という四つの基準による依存度データを記録する。さらに、機能デバイスの依存関係を記録する静態なDSMだけでなく、タスクの情報流れを記録する工程スケジュールDSM及びグループ間の交流を記録する組織構造DSMなどの動態なDSMも存在している[17][18]。

また、DSMのクラスタ化を扱うクラスタリング(clustering)は、データマイニングのデータ解析手法の一つとして、分類対象を類似性やある定められた距離尺度に基づいていくつかのクラスタ

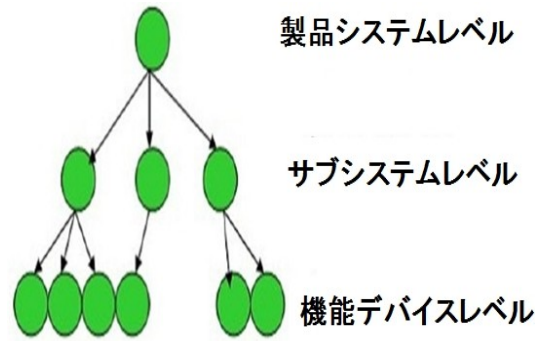


図 1.1: 製品分解の概念図

表 1.1: 5段階スコアリング

- 2	インターフェースがマイナス収益をもちながら，機能実現も無理
- 1	インターフェースがマイナス収益をもつが，機能の実現は可能
0	インターフェースが機能や収益と関係がない
+ 1	インターフェースがプラス収益をもつが，機能の実現には必要ではない
+ 2	インターフェースがプラス収益をもちながら，機能の実現には必要

に分割し，クラスタ中の分類対象の類似性が高い，クラスタ間の分類対象の類似性が低いことを目指す．DSMにおけるクラスタリングは，DSMの行と列の順番の再整理(reorder)で，対角線に沿って類似な機能を含むいくつかのクラスタが生成されて，クラスタの間に機能的類似性ができる限り存在しないことが望まれている．このクラスタ化の過程は，シミュレーテッドアニーリング法(SA)などに代表されるヒューリスティックアルゴリズムで実現される[17]．これ以外，クラスタリング識別アルゴリズム(CI)[23][24]，K-meansアルゴリズムとSOMの結合[25]及びP-median法[26]も推奨された．

しかし，製品または機能のインテグラルを考慮に入れた新製品設計に対して，上の数理モデルのいずれも評価することができないため，本研究で製品アーキテクチャに基づく製品設計手法を提案する．次の部分では製品アーキテクチャについて説明する．

## 1.4 製品アーキテクチャ理論

### 1.4.1 製品アーキテクチャとは

製品アーキテクチャ理論は藤本などに提唱されたものであり，企業における製品戦略を評価するものである[1]．さらに，組織体系や仕事のやり方と関連して企業の総合活力を評価することもできる．製品分解に基づく製品アーキテクチャの概念は，製品設計思想の一つであり，(1)機能の定義，(2)機能とデバイスとのマッピング，及び(3)デバイス間のインターフェースの設計から成

DSM(C)	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
C1 蓋	1 0	2 0										1 0			
C2 白色LEDの反射鏡	2 0	0 0	2 2	2 2								0 0			
C3 アンバーLEDの反射鏡		2 2	0 0		2 2							1 0			
C4 白色LEDライト		0 0			-1 0		1 0	0 0	0 0	0 0	0 0	1 0	2 0		
C5 アンバーLEDライト			2 2	-1 0		0 0	1 0	0 0	0 0	0 0	0 0	1 0	2 0		
C6 明暗センサー					0 0		1 0			0 0	0 0		2 2	0 0	
C7 電気回路板				1 0	1 0	1 0				1 0	1 0	2 0	2 0	1 0	2 0
C8 切替スイッチ				0 0	0 0				2 0	1 0		1 0	0 0	0 0	
C9 スイッチの支え								2 0				0 0			
C10 ニッケル水素電池				0 0	0 0	0 0	1 0	1 0			2 0	1 0	2 0	-1 0	
C11 電池の支え				0 2	0 2	2 0	0 2	2 0			0 2	0 2	0 2	0 2	0 2
C12 本体	1 0		1 0	1 0	1 0		2 0	1 0	1 0	1 0	1 0	1 0	0 0	1 0	2 0
C13 コード	0 0		0 0	0 2	0 2		0 2	0 0	0 0	0 0	0 2	0 0	0 2	0 2	0 0
C14 コンセント							0 0	1 0	0 0	-1 0		1 0	2 0		
C15 ネジ							2 0					2 0			

図 1.2: 懐中電灯とナイトライトという二つの機能を持っている多機能懐中電灯の DSM

る [27] . 製品アーキテクチャによる製品設計は、機能を反映するデバイス間のインターフェースの状態によって決まる . つまり各インターフェースをつなぐか否か、及びつなぎ方 (インテグラルかモジュール) を性能の評価値で表し、製品設計の優劣を総合収益により評価する .

製品アーキテクチャの基本タイプとしてモジュール型とインテグラル型に分けられる . 前者はそれぞれの機能デバイスが独立な機能を持っているため、事前に別々に設計しておいたデバイスを事後的に組合せていた製品 (例えばデスクトップパソコン、コンポーネントステレオ) である . 一方、後者は機能の発揮が全体的なデバイスに依存し、一つのデバイスの更新で製品全体の性能に影響を及ぼす製品 (例えば車) である . 日本の場合、統合型ものづくりの組織能力と相性の良いインテグラル型のモノづくりが得意である . そして、インテグラル型の製品設計に対して最適な製品設計を見出す問題しか存在していない .

また、図 1.3 のアーキテクチャ理論に関する先行研究のまとめにおいて、モジュール型のアーキテクチャの理論研究も実例研究も少なくないが、インテグラル型のアーキテクチャが主に概念についてのみ行われており、数理的な枠組みに対する研究が不十分である [28][29][30][31][32][33][34] . 一方、製品設計と作業組織との相性を分析する製品アーキテクチャ理論も研究されるようになってきた .

### 1.4.2 アーキテクチャ分類

モジュール型 (くみあわせ型) は、機能とデバイスをできるだけ一対一に対応させる . その代表製品は、デスクトップ型パソコンなどがある . 例えば入力機能、記憶機能、表示機能、情報処理機

作者	年間	モジュール	インテグラル
K. Ulrich et al.	1995	アーキテクチャ	
C.C. Huang.	2000	概念	概念
A. Kusiak	2002	概念	概念
D. M. Sharman et al.	2002	特性	特性
目代武史	2003		くみあわせからすりあわせへの変化
藤本隆宏	2004	アーキテクチャ理論	
Y. Yin.	2006		幾何構造に基づく製品設計手法の定式化
I. Kaku et al.	2010		インテグラルの意思決定モデル

図 1.3: アーキテクチャ理論の主な先行研究

能などのサブ機能がそれぞれキーボード、ハードディスクまたはメモリ、モニター、CPU などのように対応している。モジュールアーキテクチャ製品に対して、ある機能を向上したい場合、インターフェースが業界標準化ルールを守る限り、任意サプライヤーからもらった相応なデバイスを加えたり取り替えたりするだけで済む。

他方、インテグラル型(すりあわせ型)とは、機能とデバイス間の複雑な相互依存に対応させる。代表製品はノートブック型パソコンなどがある。例えば入力、処理、出力などデスクトップ型と同様の機能のほかに、省スペース、軽量で、手軽に持ち運びできるという機能特徴を追求するために、デバイスのレイアウトの最適化も求められる。

さらに、研究上モジュール型とインテグラル型を完全に分けて分析するが、現実上双方を併せ持つ型「近モジュール型」または「近インテグラル型」のハイブリッドアーキテクチャの製品がよく見られる。ハイブリッドアーキテクチャとは、機能とデバイスの対応関係も、それらの設計を行う組織も、モジュール型とインテグラル型との双方の特徴を併せ持つことと定義された [35]。

## 第2章 製品設計のための最適化モデルの提案

### 2.1 問題定義

ここで既存製品または機能間のインテグラル（すりあわせ）による新製品を作り出そうとしている設計問題を考える．新たな製品は両方の機能を併せ持つようになるから，製品全体の性能を向上させることが期待される．しかし，そうでない，両方の合併によって前より悪い総合性能のある新製品を生成する恐れもある．そのため，新製品の総合性能をいかに適正に評価できるかが問題点である．本論文では2つの機能デバイスのつながり関係を表すインターフェースを用いて，インターフェースの全配置によって新製品の総合性能を算出する仕組みを考える．既存製品の中にインターフェースの配置が分かったが，新製品または新機能をその上にすり合わせることで，存在していなかった新しいインターフェース関係が生成される．また，既存製品の中のインターフェース関係がそれによって再定義する必要がある．すなわち，両方の機能デバイスを含むすべてのインターフェースの再構築と評価問題は本論文で定式化される．

### 2.2 製品設計問題のモデリング化

ここでは図 2.1 のように， $M$  個の機能デバイスから構成される製品について考える．2つのデバイスを1つのインターフェースでつなげることから，インターフェースの最大個数  $N$  は  $N = M(M-1)/2$  となる．例えば， $M = 1, 2, 3, 4, 5, \dots$  のとき  $N = 1, 3, 6, 10, 15, \dots$  である．製品アーキテクチャ理論によって一般にインターフェースにはインテグラル型（すりあわせ型ともよぶ）とモジュール（組み合せ型ともよぶ）がある．インテグラル型のデバイス間には強い依存関係が存在し，モジュール型のデバイスは互いに弱い依存関係がある．例えばデバイスのインターフェースの幾何構造を調整することにより，製品全体のサイズの大きさを改善するとき，インターフェースの幾何構造を調整するとインテグラル型となり，調整しないとモジュール型となる [32]．このインターフェースに対して，インテグラル型とモジュール型により異なる収益がもたらされることが考えられ， $Y_{in}$  は幾何構造の調整に基づいてインターフェース  $i$  がもたらした収益  $R_i$  を式 (2.1) で表した [32] ．

$$R_i = \max \{0, p_i E_i - C_i\} \quad (2.1)$$

式 (2.1) において， $E_i$  は幾何構造を調整したとき得た利益， $C_i$  は幾何構造の調整にかかるコストを表す．また， $p_i \in [0, 1]$  はインターフェース  $i$  の幾何構造の調整を実現する可能性を表す．た

だし、幾何構造を調整しない(モジュール型) インターフェイスをつないだ時の効用を0であると定義し、それを基準としてインテグラル型のインターフェイスの効用を  $R_i = p_i E_i - C_i$  で表した。

また、図 2.2 に示すように、2つの機能デバイス  $\mu$  と  $\nu$  間のインターフェイス  $i$  の意思決定は  $x_i$  で表し、存在すると  $x_i = 1$  で表し、存在しないと  $x_i = 0$  で表すとする。

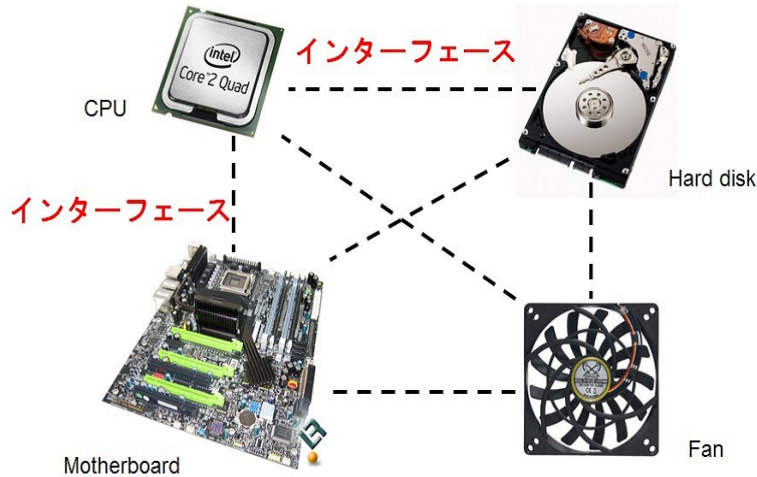


図 2.1: M=4 個のデバイスのある製品設計問題

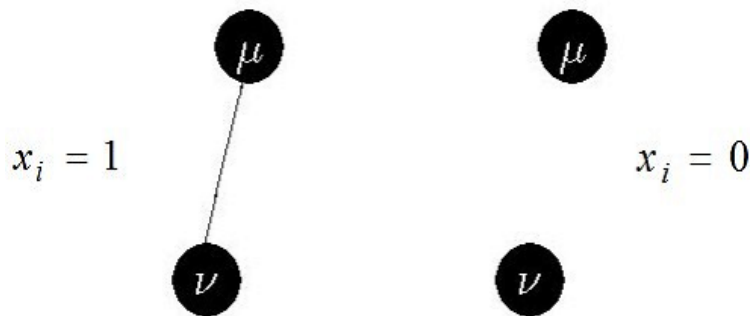


図 2.2: インターフェイスの状態 (連結するか否か)

一方、1つのデバイスは1つ以上のインターフェイスを持つことが可能であり、2つのインターフェイス間に共有デバイスが存在する場合もある。そのため、インターフェイス間の相互作用も考慮する必要がある。しかし、前期となる先行研究 (Yin) は幾何構造の調整に基づいたインターフェイスの単独効用  $R_i$  しか考慮されておらず、より一般的な製品設計を表すモデルの表現力を向上させるためには、KakuらとJiangらは単独効用  $h_i$  と相互作用項  $g_{ij}$  を導入し、製品設計の総合収益  $TR(\mathbf{x})$  を2次計画問題として式 (2.2) のように定義した [36] [37]。図 2.3 に4つの機能デバイス  $\mu, \nu, \rho, \delta$  からなるインターフェイスの組  $i, j$  間の相互効用  $g_{ij} x_i x_j$  のイメージを示す。つまり、提案の製品設計モデルはYinらの製品設計モデルを一般化したものであり、 $p_i, E_i, C_i$  の複雑な関係式を含む形で  $h_i = h_i(p_i, E_i, C_i)$  とした。これ以降本論文の主旨とずれるため、単独効用を  $h_i$  だけとする。同じように、相互効用  $g_{ij} = g_{ij}(p_i, E_i, C_i; p_j, E_j, C_j)$  とし、本論文では  $g_{ij}$  だけで表す。



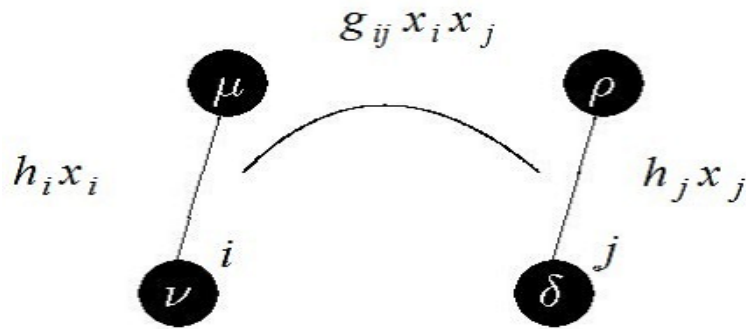


図 2.3: 相互効用のイメージ図

## 2.3 バイナリ 2 次組合せ最適化モデル

$$TR(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^N \sum_{j(\neq i)}^N x_i g_{ij} x_j + \sum_{i=1}^N h_i x_i \quad (2.2)$$

そして最適化問題は

$$\text{Max} \quad TR(\mathbf{x}) \quad (2.3)$$

$$\text{s.t.} \quad 0 \leq \sum_{i=1}^N x_i \leq a, a \leq N \quad (2.4)$$

$$x_i = \{0, 1\}, i = 1, 2, \dots, N \quad (2.5)$$

となる。ただし、ベクトル変数  $\mathbf{x} = (x_1, \dots, x_N) \in \{0, 1\}^N$  はインターフェース  $i$  を接続する ( $x_i = 1$ ) か否か ( $x_i = 0$ ) を表す。インターフェース  $i$  自身によりもたらされる利益  $h_i$  で定義されるベクトル  $\mathbf{h} = (h_1, \dots, h_N) \in \mathbb{R}^N$  と、インターフェース  $i, j$  間の相互作用  $g_{ij}$  で定義される  $N \times N$  次対称行列  $G = (g_{ij})$  が与えられたもとで、式 (2.3) の総合収益  $TR(\mathbf{x})$  を最大にする  $\mathbf{x}^*$  が最適解として求められる。そのうち、相関効用  $g_{ij}$  は任意の 2 つのインターフェースの間で生成される効用である。 $g_{ij}$  の値は正も負も取ることができる。例えば、限られたスペースの中にデバイスを組み込むことで、小型化を行うことができるが、デバイスのサイズや形によって他のデバイスの組み込みに悪影響をもたらすため、負の表現も許容する必要があるからである。本論文で相互効用項を新たに加えるすることによって、提案するモデルはモジュール型のみでなくインテグラル型の製品アーキテクチャを評価できるようになるのがモデルの新規性である。また、企業能力と製品特性などの制約により、式 (2.4) に存在するインターフェースの許容個数を定数  $a$  で制限される。また、行列  $G$  はインターフェース  $i$  と  $j$  が同時に存在するとき ( $x_i = x_j = 1$ )、総合収益へ  $x_i \frac{(g_{ij} + g_{ji})}{2} x_j = g_{ij}$  の利益分を寄与し、それ以外は 0 になる。ここでインターフェース自身の相関効用を表す対角成分は  $g_{ii} = 0$  とする。

$\mathbf{h}$  と  $G$  は製品設計の現場から得られたデータを用いることが望ましいが、熟練した設計者でさえデバイス数の多い製品の  $G$  を決定することは難しい。しかし、DSM による複数の評価指標

(例えば材料, 空間, 情報, エネルギーなど) を援用して, あるいは QFD による顧客ニーズから製品機能への数値転換プロセスによって,  $h$  や  $G$  を定量的に見積もることが可能であると思われる. 本論文の第6章で詳しくこれらのモデルパラメータの推定解析を説明する. 従って製品設計の最適化問題を定式化した上で, この2次組合せ最適化モデルをどのように解くのが次の課題となる.

## 第3章 分枝限定法による2次モデルの厳密解析

製品アーキテクチャによる製品設計について，Yin はインテグラル型とモジュール型のインターフェースの効用を評価できる1次モデルを議論した [32]．また，Kaku と Jiang などがそれを発展し，インターフェースの単独効用のみでなく，インターフェース間の相互作用も式 (2.2) で考慮に入れた2次組合せ最適化モデルを提案した [36][37]．このモデルにより総合収益  $TR(\mathbf{x})$  を最大にするインターフェースの配置  $\mathbf{x}^*$  が最適解として求められる．新規性として，この提案モデルにおいて相互効用項の設定によりモジュール型のみでなくインテグラル型の製品アーキテクチャを評価できるようになる．

しかしながら，ここで1点注意する必要がある． $\text{trace}(G) = \sum_{i=1}^N g_{ii} = 0$  であるため，一般的に  $G$  は非負定値行列ではない．そのため，最急降下法などの勾配法を用いて得られた解が最適解である保証はない [38]．また，2次モデルの子問題（一部の変数の値が固定化された部分問題）では，固定化されていない変数に対応する  $G'$  は  $G$  の一部に相当し，子問題のトレースも  $\text{trace}(G') = 0$  であるため， $G'$  も非負定値行列ではない．そのため，子問題の緩和値が極値である保証はない．つまり整数計画問題によく使われている分枝限定法でこの2次モデルの最適化を解析することは困難である．

そこで本章では，2次製品設計モデルにおける非線形項とある条件下で等価な意味を持つ線形項を導入し，新たな製品設計モデルを提案する．2次モデルとその1次等価変換を一致させるために，新たな不等式制約を用意する．さらに，分枝限定法を用いて1次等価モデルを解くことにより，大規模問題の最適解を得ることが期待できる．

本章の構成は以下の通りである．第1節では2次モデルに対応する1次等価変換モデルを提案する．第2節ではモデルの最適化解析に適用する分枝限定法を述べる．第3節では数値実験でモデルの最適化解析を行い，その効果を考察する．最後に第4節では結論と今後の課題を述べている．

### 3.1 1次等価変換モデルの提案

2次モデルの最適化解析を回避するために，変数  $y_{ij}$  を導入して式 (2.2) の2次項  $x_i x_j$  を書き換える．つまり，総合収益  $TR(\mathbf{x}, Y)$  は式 (3.1) のようになる [39]．

$$TR(\mathbf{x}, Y) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N g_{ij} y_{ij} + \sum_{i=1}^N h_i x_i \quad (3.1)$$

そして最適化問題は

$$\text{Max} \quad TR(\mathbf{x}, Y) \quad (3.2)$$

$$\text{s.t.} \quad 0 \leq \sum_{i=1}^N x_i \leq a, a \leq N \quad (3.3)$$

$$\alpha(x_i + x_j) + \beta \leq y_{ij} \leq \alpha'(x_i + x_j) + \beta' \quad (3.4)$$

$$x_i \in \{0, 1\}, i = 1, 2, \dots, N \quad (3.5)$$

$$y_{ij} \in \{0, 1\}, i, j = 1, 2, \dots, N, (i < j) \quad (3.6)$$

となる。ただし、 $(\alpha, \beta)$  と  $(\alpha', \beta')$  は

$$\begin{cases} 0 < 2\alpha + \beta \leq 1, \\ \alpha + \beta \leq 0, \\ \beta \leq 0. \end{cases} \quad (3.7)$$

$$\begin{cases} 2\alpha' + \beta' \geq 1, \\ 0 \leq \alpha' + \beta' < 1, \\ 0 \leq \beta' < 1. \end{cases} \quad (3.8)$$

を満たす組を選び、 $\frac{1}{2}x_i(g_{ij} + g_{ji})x_j = g_{ij}x_ix_j$  を用いた。ここで  $y_{ij}$  も 0-1 変数であるが、 $y_{ij} \neq x_ix_j$  (例えば  $y_{ij} = 1, x_i = 1, x_j = 0$ ) となる実行可能解は意味を持たないことに注意する。そのため  $y_{ij}$  と  $x_ix_j$  が一致するために、 $y_{ij} = x_ix_j$  の非線形関係を式 (3.4) の  $x_i$  と  $x_j$  の線形不等式で再構築し、制約式 (3.6) との併用で  $y_{ij}$  の値と  $x_ix_j$  の値を一致させる。式 (3.4) に 4 つの実行可能解  $(x_i, x_j, y_{ij}) = (1, 1, 1), (1, 0, 0), (0, 1, 0), (0, 0, 0)$  を代入し、式 (3.7) と式 (3.8) が得られる。式 (3.7) と式 (3.8) において、 $y_{ij} = x_ix_j$  となるようなパラメータ  $\alpha$  と  $\beta$  及び  $\alpha'$  と  $\beta'$  の値の取りうる範囲 (図 3.1 と図 3.2 の点線を除く陰影の部分) は以下のように決定する。

1.  $x_i = x_j = 1$  のとき、式 (3.4) は  $2\alpha + \beta \leq y_{ij} \leq 2\alpha' + \beta'$  になる。 $x_ix_j = 1$  と等価な  $y_{ij} = 1$  を保証するために、 $y_{ij}$  の下限条件  $0 < 2\alpha + \beta \leq 1$  と上限条件  $2\alpha' + \beta' \geq 1$  を同時に満たさなくてはならない。
2.  $x_i = 1, x_j = 0$  あるいは  $x_i = 0, x_j = 1$  のとき、式 (3.4) は  $\alpha + \beta \leq y_{ij} \leq \alpha' + \beta'$  になる。 $x_ix_j = 0$  と等価な  $y_{ij} = 0$  を保証するために、 $y_{ij}$  の下限条件  $\alpha + \beta \leq 0$  と上限条件  $0 \leq \alpha' + \beta' < 1$  を同時に満たさなくてはならない。
3.  $x_i = x_j = 0$  のとき、式 (3.4) は  $\beta \leq y_{ij} \leq \beta'$  になる。 $x_ix_j = 0$  と等価な  $y_{ij} = 0$  を保証するために、 $y_{ij}$  の下限条件  $\beta \leq 0$  と上限条件  $0 \leq \beta' < 1$  を同時に満たさなくてはならない。

上の3条件を同時に満たす  $\alpha$  と  $\beta$  及び  $\alpha'$  と  $\beta'$  の値の範囲は、図 3.1 と図 3.2 の陰影の部分 (点線の上の値を除く) にあたる。この範囲内に制約式 (3.6) と併用すれば、 $y_{ij}$  の値と  $x_ix_j$  の値は一致することが分かる。

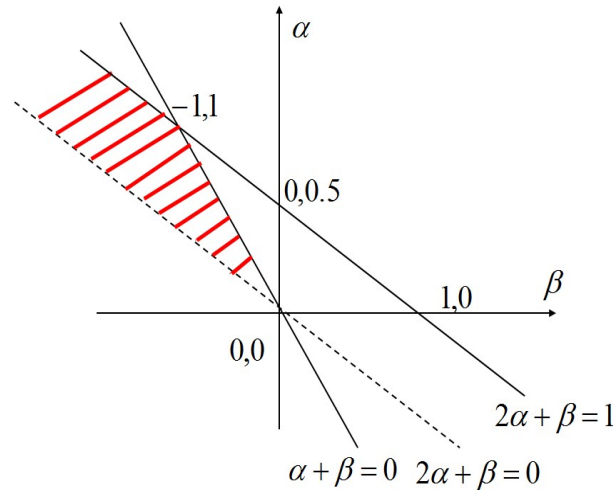


図 3.1:  $\alpha$  と  $\beta$  の値の取りうる範囲

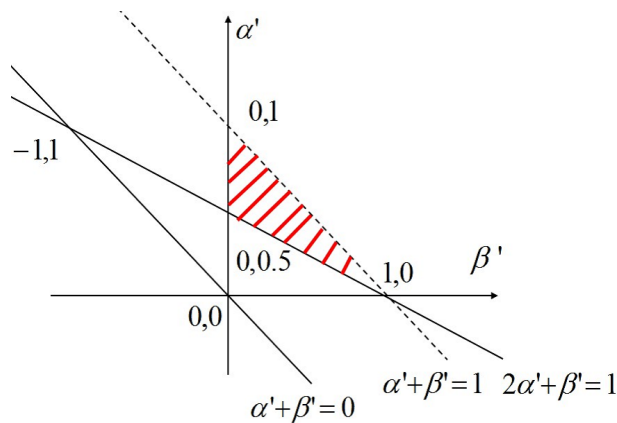


図 3.2:  $\alpha'$  と  $\beta'$  の値の取りうる範囲

### 3.2 1次等価変換モデルの最適化解析

0-1 整数線形計画モデルの緩和問題の最適解を求めるとき、変数の緩和範囲が  $[0, 1]$  であるならば、この最適解は実行可能解になることが知られている。しかし提案する1次等価変換モデルの緩和問題の最適解を求めるとき、変数の緩和範囲が  $[0, 1]$  に加え、制約条件 (3.4) が同時に課されるため、最適解は実数解になる可能性がある。このとき、変数の緩和範囲を満たす実行可能解の総合収益の値はこの実数解の総合収益の値より大きくなることはない。例として、2変数のある極めて簡単なケースを考える。組となるインターフェース1と2に対して、 $h_1 > 0, h_2 > 0, g_{12} < 0$ 、ただし  $|h_1| > |h_2| > |g_{12}|$  の場合を仮定し、線形関係により緩和問題の最適解  $x_1 = x_2 = 1, y_{12} = 0$  の値は  $h_1 + h_2$  である。しかし、 $[0, 1] \cap [\alpha(x_1 + x_2) + \beta, \alpha'(x_1 + x_2) + \beta']$  の共通部分に制限される  $y_{12}$  の緩和範囲は  $2\alpha + \beta \leq y_{12} \leq 1$  であるため、実際に緩和問題の最適解は実数解  $x_1 = x_2 = 1, y_{12} = 2\alpha + \beta$  にしか収束できず、緩和値  $h_1 + h_2 + (2\alpha + \beta)g_{12}$  は  $h_1 + h_2$  より小さい。一方、1次モデルの実行可能な最適解  $x_1 = x_2 = y_{12} = 1$  の値は  $h_1 + h_2 + g_{12}$  である。 $h_1 + h_2 + g_{12} < h_1 + h_2 + (2\alpha + \beta)g_{12}$  の関係によってこの問題の緩和解が上界となることが分

かる。

上述の特徴を踏まえて、ここでは勾配法による緩和問題と分枝限定法を用いて1次等価変換モデルの最適化解析を行う方法を提案する。0-1整数計画問題に対して、分枝限定法は計算量の指数爆発を克服でき、最適解を探索できるアルゴリズムの1つとして知られている [40][41][42][43]。基本的な考え方は、元の問題をいくつかの子問題に分割し、最大化問題を想定する場合、子問題の最適実数解を上界とし、暫定下界(実行可能解)とのギャップにより終端となる子問題を識別して、最適解の所在を絞り込むことである。1次線形モデルの子問題の緩和値が極値であるため、分枝限定法で得られる解が最適解となる。

**初期下界の確定** : 任意の実行可能解  $x$  に対して、ここでは1点変動近傍法により求めた局所最適解を初期下界  $L^0$  とする [44]。

**分枝操作** : 深さ優先探索と最良上界値優先探索の両方を採用する。前者の場合、変数の添字の昇順に従って1番目の整数でない変数を優先的に選び、0の場合(左枝)と1の場合(右枝)に分けてペアの子問題に分解する。後者の場合、暫定の最良上界値のある子問題を優先的に選ぶ。また、数値実験の結果により、中型と大型問題の場合でも、両者の平均探索時間には大きな差がなかったため、本章では説明が容易に行える深さ優先探索の方法と結果のみを記載する。また、変数  $y_{ij}$  が固定されたら  $y_{ji}$  も同じように固定される。

**限定操作** : 子問題の緩和値を内点法で計算する。すべての変数が0か1になる緩和解は実行可能解である。限定操作において、式(3.3)または式(3.4)を満たさない非実行可能解と暫定下界より小さい緩和値を持つ解を停止させる。

以下は分枝限定法のアルゴリズムを示す。図3.3は4デバイス、6インターフェース ( $M = 4, N = 6$ ) の1次モデルを例としてアルゴリズムの流れを説明する。この例の入力  $h = (0, 0, 0, 0, 16, 5)$ ,

$$G = \begin{pmatrix} 0 & 0 & 2 & -16 & -10 & 20 \\ * & 0 & 2 & 14 & 8 & -12 \\ * & * & 0 & -6 & 6 & 12 \\ * & * & * & 0 & -20 & -16 \\ * & * & * & * & 0 & 18 \\ * & * & * & * & * & 0 \end{pmatrix}$$

となる。

**ステップ0 (初期上界と下界)** : 1点変動近傍法により初期下界値  $L^0 = 69$  を得る。初期緩和値  $U^0 = 88.6$  を求めた。

**ステップ1 (収束条件)** : 限定操作をまだしていない子問題がなければ計算を終了する。暫定下界  $L$  が対応する解  $(x, Y)$  を最適解として出力する。限定操作をしていない子問題があればステップ2に進む。

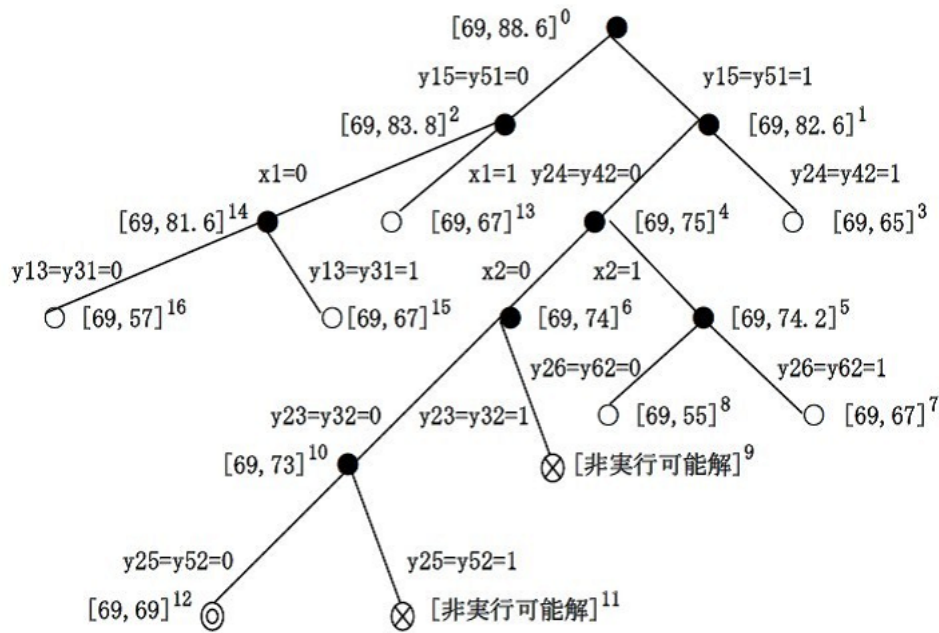


図 3.3: 探索過程の2分木 [暫定下界  $L$ , 上界  $U$ ]<sup>子問題の順番</sup> ○: 上界が暫定下界より小さいため終端となる子問題. ⊙: 最適解. ⊗: 非実行可能解.

ステップ2 (限定操作の対象の確定): 子問題集合の中に最後に生成された子問題を限定操作の対象となる.

ステップ3 (限定操作):

- $k$  番目の子問題  $(x, Y)^k$  は非実行可能解の場合, この子問題を終了させ, ステップ1に戻る. 例えば変数  $y_{23} = x_2 x_3 = 1$  を固定する前に  $x_2 = 0$  がすでに決められたため, 子問題9は非実行可能解となる.
- $k$  番目の子問題  $(x, Y)^k$  は実行可能解である場合, 実行可能解の  $U^k$  が  $U^k > L$  ならば暫定下界を更新 ( $U^k \rightarrow L$ ) し, この子問題を終了させ, ステップ1に戻る. 例えば実行可能解である子問題12は終了させたが,  $U^{12} = 69 = L$  のため暫定下界を更新しない.
- $k$  番目の子問題  $(x, Y)^k$  は実数解の場合, 緩和値  $U^k \leq L$  ならば (例えば子問題3) 終了させ, ステップ1に戻る. 一方  $U^k > L$  なら (例えば子問題4), 分枝操作の対象となり, ステップ4に進む.

ステップ4 (分枝操作): 分枝対象となる子問題における一番左の整数でない変数を分枝変数としてペア子問題を生成し, ステップ1に戻る.

図3.3のように, 分枝と限定を繰り返してステップ1の収束条件に満足すると計算を終了し, 最適解69を出力する. 図2の初期下界  $L^0 = 69$  のとき, 深さ優先探索により, 収束するまで16個の子問題をしか探索していない一方, 初期下界  $L^0 = 0$  とすると最適解に収束するまで探索した子問題の数は64個となる.

### 3.3 数値実験による提案法の評価

#### 3.3.1 実験設計

1次等価変換モデルの最適化解析の有効性を確認するために、以下の数値実験を設計する。(1)1次等価変換モデルに適用した分枝限定法の能力を確認する。(2)行列  $G$  の特性と探索コストとの関係を考察する。(3)パラメータ  $\alpha, \beta, \alpha', \beta'$  と探索効率との関係を考察する。計算は最適化ソフトウェア Lingo11 を用いて行い、計算精度は小数点以下5桁を設定し、子問題の緩和解法は Lingo11 に内蔵した内点法を使用した。コンピュータは HP Compaq 8100 Elite SFF PC (OS が Windows 7, CPU が Intel(R) Core(TM) i3@2.93GHz, メモリが 2.00GB である) を使用した。

#### 3.3.2 分枝限定法の能力限界

実験のパラメータは、(1)  $\alpha, \beta, \alpha', \beta'$  の値は式 (3.7) と式 (3.8) の範囲にあれば最適解に影響しないことから、 $\alpha = \alpha' = 0.5, \beta = -0.6, \beta' = 0$  を採用する。(2) インターフェース個数の制限は  $a = N$  とする。(3)  $h_i$  と  $g_{ij}$  は各々独立に平均0、分散5の正規分布に従ってランダムに生成させる [45]。(4)  $G$  の成分のゼロでない比率 (非零比率)  $r(G)$  を  $r(G) = 20\%$  とする。

図 3.4 は問題の規模による列挙法と分枝限定法の探索コストを示す。図 3.4 において、横軸はデバイス数 (インターフェース数) を示し、縦軸は 10 を底とする対数で表す計算時間 (秒) である。図 3.4 に示したように、列挙法はデバイス数の増加につれて急激に指数的に増加することに対して、分枝限定法は緩やかに指数的に増加する。これより、提案した分枝限定法は列挙法ができない実用レベルのモデル (16 デバイス, 120 インターフェース) の最適解析を実効的な時間で行うことができる。例として、Pimmler らの研究により、Ford Motor 製のカーエアコンシステムは 16 個のデバイス (中間製品) で構成されている [16]。そのため、上記のアルゴリズムで扱える 16 デバイス, 120 インターフェースは実用レベルであるといえる。

また、対象とする製品の設計期間によって求解時間の実用性も異なるが、数ヶ月程度の設計期間のある製品の場合、 $10^5$  秒 (約 28 時間) は合理的であると考えられる。

#### 3.3.3 行列 $G$ の特性と探索コスト

デバイスを連結するためのインターフェースが互いに影響しなければ、 $g_{ij} = 0$  となる。すなわち、ゼロでない成分を含む行列  $G$  の性質によって、アルゴリズムの性能がどのように影響するか、つまり、ゼロでない成分の比率  $r(G)$  による分枝限定法の探索能力に与える影響を考える。この小節では  $r(G)$  を変えて数値実験で検証する。

図 3.5 は非零比率  $r(G)$  と探索コストの変化を示す。数値実験においては  $h$  を固定して、 $G$  の非零比率  $r(G)$  のみを変えて生成する。図 3.5 において、横軸は  $r(G)$  であり、縦軸は探索コスト (秒) である。10 デバイスからなる製品 ( $M=10, N=45$ ) と 16 デバイスからなる製品 ( $M=16, N=120$ ) の



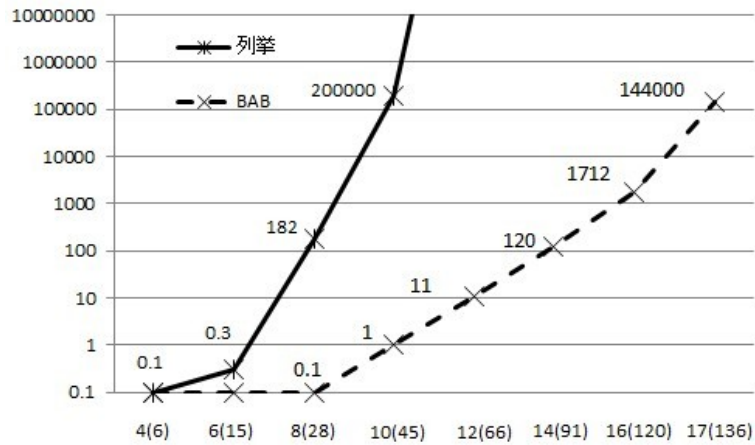


図 3.4: 列挙法と分枝限定法の探索コスト (秒)

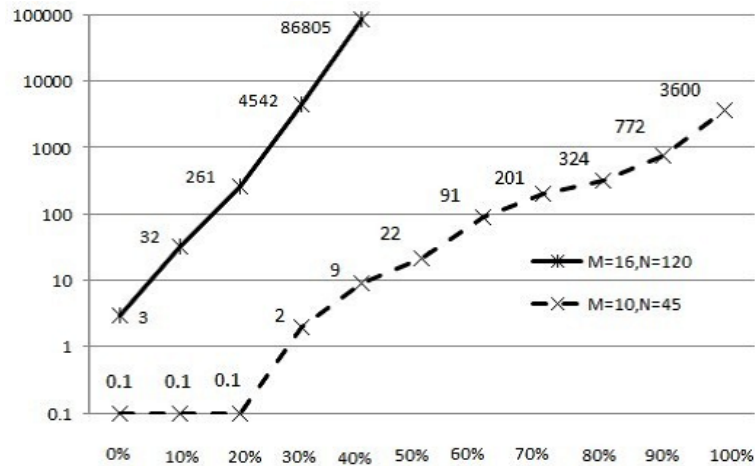


図 3.5: 非零比率  $r(G)$  と分枝限定法の探索コスト (秒)

2つの設計問題の結果を示した。図 3.5 から、 $r(G)$  の増加に伴い、探索コストも増大していることが分かる。また問題規模が大きくなると、 $r(G)$  の増大に従って探索コストが急激に増大する。これはインターフェースが互いに関連し合うと必要な分枝が多くなり、より多くの子問題を解かなければならないことから、探索コストが増大したと解釈できる。このことは場合によって解ける問題の規模に影響する。例えば図 3.5 から、16 デバイスの設計問題において  $r(G)$  が 40% を超えると実務的な時間内に解くことができない。実際、例えばパソコンの例を用いれば、ノートパソコンはデスクトップパソコンよりコンパクトにする必要があるため、インターフェース間の関連はもっと緊密となり (すなわち、ノートパソコンの非零比率よりデスクトップパソコンの非零比率が少なくなる)、より多くの関連デバイスを考慮しなければならないことが分かる。

表 3.1:  $\alpha, \beta, \alpha', \beta'$  の値

	$\alpha$	$\beta$	$\alpha'$	$\beta'$
端点	1	-1	0.5	0
境界線 1	1.5	-2	0.75	0
境界線 2	0.5	-0.5	0.25	0.5
内部 1	0.5	-0.75	0.75	0.1
内部 2	1	-1.5	0.25	0.6

### 3.3.4 $\alpha, \beta, \alpha', \beta'$ と探索効率

制約条件にある  $\alpha, \beta, \alpha', \beta'$  の値が式 (3.7) と式 (3.8) の範囲にあるとき, 1 次等価変換モデルは意味を持つが, それらによって制限されていた決定変数  $y_{ij}$  の値が変化し, 結果的に探索コストに影響することが考えられる. ここでは  $\alpha, \beta, \alpha', \beta'$  の値は端点とそうでないケースに分けて数値実験で考察する.

$\alpha, \beta, \alpha', \beta'$  の値は表 3.1 に示す. 端点, 境界線上 (2 種類) と内部 (2 種類) の計 5 ケースを用意して実験を行った. 図 3.6 と図 3.7 は非零比率  $r(G) = 20\%$  である場合と  $80\%$  である場合に対して, 5 ケースの探索コストを示した. 図 3.6 と図 3.7 において, 横軸はデバイス数 (インターフェース数) であり, 縦軸は探索コスト (秒) である. 図 3.6 と図 3.7 から, 端点や境界線のケースにおける探索コストがそうでない (内部) ケースと比べて高いことが分かる. 例えば, 図 3.6 から 16 デバイスを持つ設計問題に対して, この差は約 45% ぐらいである. また図 3.7 から 10 デバイスを持つ問題に対して, この差は約 100% となる. すなわち, 非零比率の多寡に係わらず, 値を取りやすい端点あるいは境界線にパラメータ  $\alpha, \beta, \alpha', \beta'$  を取ることは必ずしもよい探索効率を得るとは限らない. どのように  $\alpha, \beta, \alpha', \beta'$  を決定すれば探索効率が一番良いかについて研究する必要があり, 次の研究課題とする.

## 3.4 まとめ

製品アーキテクチャ理論に基づく 2 次組合せ最適化モデルは製品設計を評価することができるが, この 2 次モデルの非線形項係数  $G$  は非負定値行列ではないため, 従来の勾配法で得られた解が最適解である保証はない. そこで, 本章ではこの 2 次モデルにおける非線形項と等価である線形変数及び不等式制約を導入し, 1 次等価変換モデルを提案した. さらに 2 次モデルとの等価性を示した上, 1 次等価変換モデルを解析する分枝限定法を提案した. これにより製品設計問題の最適化解析は実用レベルに向上した. ここでの実用レベルは 16 個のデバイス (120 個のインターフェース) を超えない規模問題を指す. 例えば, 機能の少ない既存製品 (2 機能を持つ懐中電灯, 計 14 個のデバイス) と機能の多い製品のモジュール (カーの一部となるカーエアコンシステム, 計

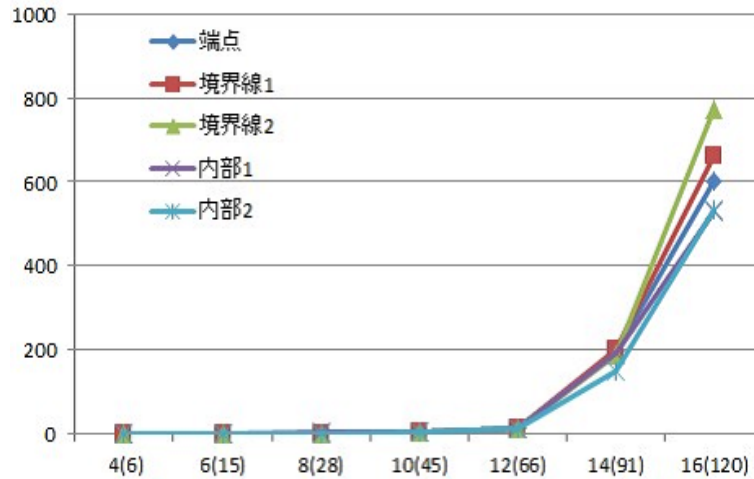


図 3.6: パラメータと探索効率 ( $r(G) = 20\%$ )

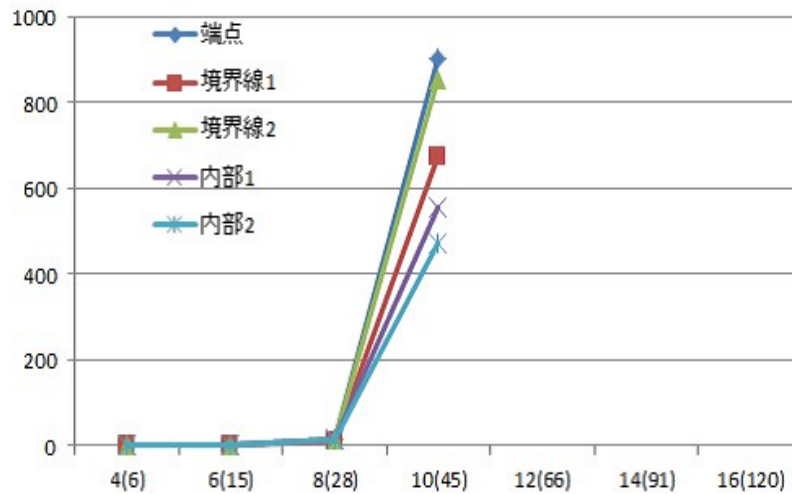


図 3.7: パラメータと探索効率 ( $r(G) = 80\%$ )

16 個のデバイス) を含むと考える．最後に数値実験で提案手法の有効性と分枝限定法アルゴリズムの有用性， $G$  の特性及びパラメータについて検証した．

今後の課題として，本研究では単独効用  $h_i$  と相関効用  $g_{ij}$  を与えられたもとの 1 次等価変換モデルの性能について議論してきたが，実用に向けて， $h_i, g_{ij}$  の可能性  $p_i$ ，利益  $E_i$ ，コスト  $C_i$  による関係式  $h_i = h_i(p_i, E_i, C_i)$  と  $g_{ij} = g_{ij}(p_i, E_i, C_i; p_j, E_j, C_j)$  を特定する必要もあるだろう．また，提案したモデルはデバイスをつながるインターフェースを決定するものであり，それに基づくデバイスのグループ化 (モジュール化) は次の研究目標である．また，製品設計の従来研究と比較して本研究の実用性を検証することも重要である．



## 第4章 平均場近似法による製品設計の大規模問題の近似解析

前章の議論で分枝限定法による実務レベル(デバイス数  $M = 16$ , インターフェース数  $N = 120$ )の製品設計問題の厳密解を解いた。しかしながら, 大規模問題(例えば  $M = 100, N = 4950$ )における決定変数の数  $N$  が非常に多い場合, 分枝限定法によっても式(2.2)の組合せ最適化問題の計算量は指数的に爆発したNP困難となることが分かる。

NP困難の証明について具体的には, 式(2.2)の一つの入力  $\mathbf{x}$  は多項式時間  $O(N^2)$  内に一つの出力  $TR(\mathbf{x})$  を生成する。この出力  $TR(\mathbf{x})$  は最適解であるのかが問われるとき, 最悪の場合, 解空間すべての出力を比較していく必要があり, 指数時間  $O(2^N)$  が要ると想像できる。そのため大規模問題の場合, 合理的な時間内に悪くない近似解を見出せる近似解法を考える意義がある。

一方, 問題を確率的に取り扱おうという方法論の一つとして統計力学における平均場近似 (Mean Field Approximation, MFA) が注目を集めつつある [46][47][48][49]。平均場近似とは多体問題を一体問題に帰着させる近似のことである。具体的に, 相互作用しているインターフェースのある一つの期待値を解けば, ほかのインターフェースがその解に従って変わる。そこで本章で平均場近似を援用して製品設計の大規模問題の近似解を求めてみる。

本章の構成は次の通りである。第1節では総合コストによる製品設計モデルの変形式を提案する。第2節では統計力学に基づく平均場近似のメカニズムを紹介した上で, 3節目に提案するモデルのNP困難を解決する。第4節では3つの数値実験で提案手法の効果を考察する。最後に第5節ではまとめを述べる。

### 4.1 製品設計のコスト最小化問題

前章の総合収益  $TR$  の最大化問題と違い, 本章で平均場近似法に適用する総合コストの最小化問題を議論する [37]。ここでインターフェースの変数値を改めて定義する。 $M$  個のデバイスの場合, 前章と同じ  $N = \frac{M(M-1)}{2}$  個のインターフェースがあり, ラベル  $\mu, \nu = 1, 2, \dots, M$  でインターフェースごとに標記する。図4.1に示すように, 2つのデバイス  $\mu, \nu$  を1つのインターフェース  $S_i$  でつなぐと連結された (coupled) とし, つながらないと連結されなかった (decoupled) とする。式(4.1)に連結されると  $S_i = -1$ , 連結しないと  $S_i = 1$  を定める。

$$S_i = \begin{cases} -1 & \text{coupled} \\ 1 & \text{decoupled} \end{cases} \quad (4.1)$$

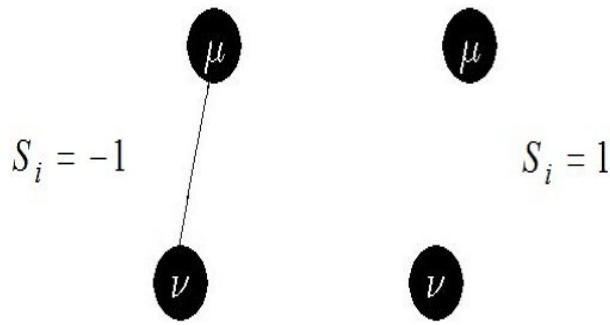


図 4.1: インターフェースの状態 (連結するか否か)

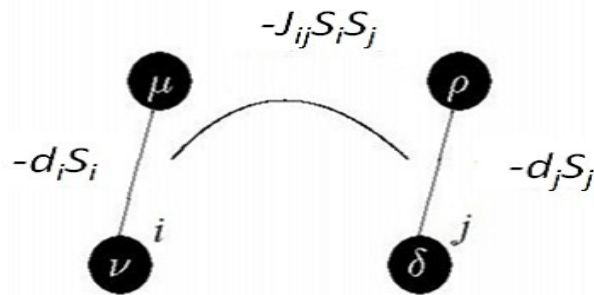


図 4.2: 相関コストのイメージ図

図 4.1 にノードでデバイスを表し、直線でインターフェースを表す．左側が連結されたインターフェースであり、右側がそうでないインターフェースである．インターフェースそのものの単独コスト (single cost) を式 (4.2) のように表すことができる．

$$-d_i S_i = \begin{cases} d_i & S_i = -1 \\ -d_i & S_i = 1 \end{cases} \quad (4.2)$$

ただし、 $d_i$  はインターフェース  $i$  の単独コストの係数である． $d_i > 0$  ならば、連結したインターフェースの単独コストはいつも連結しなかった方より小さくないことが分かった．逆にも、 $d_i < 0$  ならば、連結しなかったインターフェースの単独コストはいつも連結した方より小さくない．また、ここで  $S_i \in \{\pm 1\}$  とすることによって、 $d_i = 0$ 、すなわち、インターフェース  $i$  そのものがつながっていても単独コストを起こさない情報を失う可能性をなくした．そしてインターフェース間の相互作用を考慮することによって、インターフェースの相関コスト (correlation cost) を式 (4.3) のように定義する．

$$-J_{ij} S_i S_j = \begin{cases} J_{ij} & S_i \neq S_j \\ -J_{ij} & S_i = S_j \end{cases} \quad (4.3)$$

ただし、 $J_{ij}$  はインターフェース  $i$  と  $j$  間の相互作用コストの係数である． $J_{ij} > 0$  ならば、インターフェース  $i$  と  $j$  が異なる状態になる方 ( $S_i S_j = -1$ ) がそうでない方 ( $S_i S_j = 1$ ) より小さくないことが分かった．逆にも、インターフェース  $i$  と  $j$  が同じ状態になる方 ( $S_i S_j = 1$ ) がそうでな

い方 ( $S_i S_j = -1$ ) より小さくない．上の議論により，図 4.2 にインターフェース  $i$  と  $j$  の組の相関コストを示す． $-d_i S_i$  と  $-d_j S_j$  がそれぞれインターフェース  $i$  と  $j$  の単独コスト (single cost) を表し， $-J_{ij} S_i S_j$  がインターフェース  $i$  と  $j$  間の相関コスト (correlation cost) を表す．

ここで 2 つ注意するポイントがある．まず，コスト係数  $J_{ij}$  と  $d_i$  の値が正も負も取れる．例えば，インターフェース  $i$  と  $j$  が同時に存在する場合，もしこの組の相互作用が総合性能に良い効果をもたらしたら，両方の相互作用コスト  $J_{ij} < 0$  とするはずである．また， $J_{ij} = 0$  がインターフェース  $i$  と  $j$  が同時に存在しても直接に影響することがないを意味する．次は，異なる  $J_{ij}$  と  $d_i$  のセットに関連する最適設計も違うのため，適正な  $J_{ij}$  と  $d_i$  の値が望まれている．しかし， $J_{ij}$  と  $d_i$  の値を設計現場から簡単にもらうことが難しい．大量の設計実例から発見的に  $J_{ij}$  と  $d_i$  の値を統計的に帰納するか，あるいは顧客ニーズと市場動向に敏感，たくさんの設計経験を積んだ熟練した設計者から，経験的に  $J_{ij}$  と  $d_i$  の値を抽出する方法が，その一つの研究方向であるが，理論上に定義することが将来の研究の一つとする．

さらに， $J_{ij}$  と  $d_i$  に基づいて総合コスト関数  $\mathcal{H}(\mathbf{S})$  を式 (4.4) のように提案する．

$$\begin{aligned}\mathcal{H}(\mathbf{S}) &= -\sum_{i=1}^N d_i S_i - \sum_{i>j}^N J_{ij} S_i S_j \\ &= -\mathbf{d}^T \mathbf{S} - \frac{1}{2} \mathbf{S}^T \mathbf{J} \mathbf{S}\end{aligned}\quad (4.4)$$

ただしマトリクス  $J$  が対称的であり  $J_{ij} = J_{ji}$  を意味する．またインターフェース  $i$  が自身への影響がないと考えるため  $J_{ii} = 0$  とする． $T$  がベクトルとマトリクスの転置を意味する．そして，コスト関数による組合せ最適化問題が式 (4.5) で解き，最適解  $\mathbf{S}^*$  を求めることが目的である．

$$\mathbf{S}^* = \arg \min_{\mathbf{S}} \mathcal{H}(\mathbf{S}) \quad (4.5)$$

また， $\forall A > 0$ ，ベクトル  $\mathbf{S}$  に頼らない  $B \in \mathbb{R}$  の場合，式 (4.5) が  $\arg \min_{\mathbf{S}} \{A\mathcal{H}(\mathbf{S}) + B\}$  と等価することが分かる．そのため，連結したインターフェース  $S_i = -1$  の単独コスト  $d'_i$  の値が，連結しなかったインターフェース  $S_i = 1$  の単独コスト  $d''_i$  の値と異なる場合，単独コストが式 (4.6) のように書き直すことができる．

$$\begin{aligned}-d_i S_i &= \begin{cases} d'_i & S_i = -1 \\ -d''_i & S_i = 1 \end{cases} \\ &= \frac{d'_i - d''_i}{2} - \frac{d'_i + d''_i}{2} S_i\end{aligned}\quad (4.6)$$

そのうち，定数項  $\frac{d'_i - d''_i}{2}$  を無視して， $S_i = -1$  と  $S_i = 1$  に対して対称的な  $d_i = \frac{d'_i + d''_i}{2}$  を保証できる．同様に，相関コストも式 (4.7) のように書き直すことができる．

$$\begin{aligned}-J_{ij} S_i S_j &= \begin{cases} J'_{ij} & S_i \neq S_j \\ -J''_{ij} & S_i = S_j \end{cases} \\ &= \frac{J'_{ij} - J''_{ij}}{2} - \frac{J'_{ij} + J''_{ij}}{2} S_i S_j\end{aligned}\quad (4.7)$$

定数項  $\frac{J'_{ij}-J''_{ij}}{2}$  を無視して,  $S_i = -1$  と  $S_i = 1$  に対して対称的な  $J_{ij} = \frac{J'_{ij}+J''_{ij}}{2}$  を保証できる. 式 (4.6) と (4.7) によって, たとえ元のマトリクス  $J$  が非対称になっても, 非対称を対称にしたマトリクス  $J'$  と元の  $J$  が同じ効果を持っていることが分かる.

上の議論により, 単独コストと相関コストの定義を図 4.3 と図 4.4 で表す.

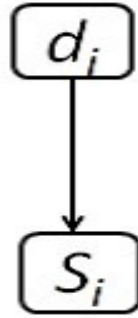


図 4.3: 単独コストのみが存在するケース

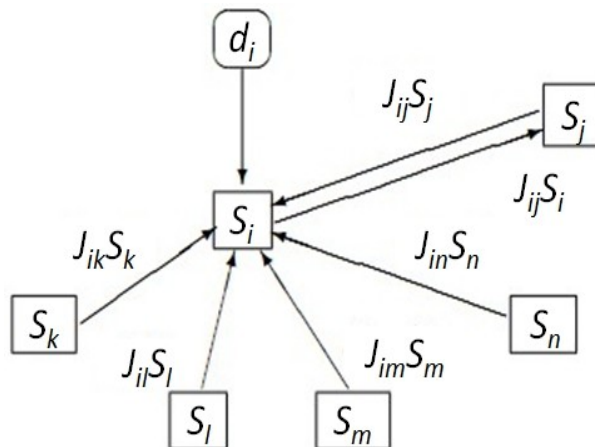


図 4.4: 近傍の相関コストを考慮に入れたケース

図 4.3 と図 4.4 における四角形でインターフェース  $i$  の意思決定を表し, 長円形で単独コストの係数  $d_i$  表すとする. 図 4.3 の特別なケースに対して, 任意インターフェースの組  $i$  と  $j$  の相互作用が存在しない, すなわち相互作用コスト係数  $J_{ij} = 0$  とするとき,  $S_i$  の意思決定は単独コスト係数  $d_i$  にしか影響されていない. このときの最適解は一意に式 (4.8) で求められる.

$$S_i^* = \begin{cases} -1 & d_i < 0 \\ 1 & \text{otherwise} \end{cases} \quad (4.8)$$

つまり,  $S_i^* = \text{sgn}(d_i)$ ,  $\mu \geq 0$  のとき  $\text{sgn}(\mu)$  は値 1 を取り, そうでないとき  $\text{sgn}(\mu)$  は値 0 を取る. また,  $J_{ij} = 0$  によって, 式 (4.4) の  $\mathcal{H}(\mathbf{S})$  を  $\mathcal{H}(\mathbf{S}) = -\sum_{i=1}^N d_i S_i = -d_1 S_1 - \sum_{i=2}^N d_i S_i$  のように書き換えることができる. そのため,  $\mathcal{H}(\mathbf{S}) = -d_1 S_1 + (\text{S}_1 \text{に関わらない項})$ , これによって各インターフェースの意思決定  $S_i$  はそれと関係ある係数  $d_i$  のみに決められる. このケースの場合, 1つのインターフェース  $i$  の値はオーダー  $O(N)$  の計算量で厳密的に求められる.



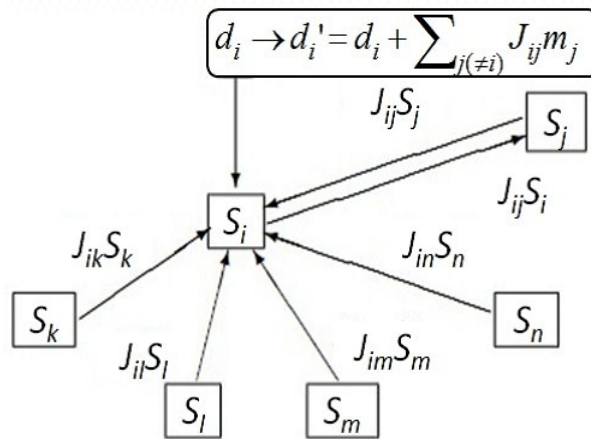


図 4.5: 平均場近似の概念図

しかしながら，もっと一般化の場合，すなわち，相関コストの係数  $J_{ij} \neq 0$  の場合，式 (4.4) の  $\mathcal{H}(\mathbf{S}) = -(d_i + \sum_{j(\neq i)} J_{1j} S_j) S_1 + (\text{S}_1 \text{に関わらない項})$  にまとめることができる．

$$S_i^* = \begin{cases} -1 & d_i + \sum_{j(\neq i)} J_{ij} S_j^* < 0, \\ 1 & \text{otherwise,} \end{cases} \quad (4.9)$$

つまり， $S_i^* = \text{sgn}(d_i + \sum_{j(\neq i)} J_{ij} S_j^*)$ ．インターフェースの組の意思決定  $S_i$  と  $S_j$  が影響し合うことにより， $S_i$  が係数  $d_i$  のみでなく，近傍のすべての相関コスト  $\sum_{j(\neq i)} J_{ij} S_j$  から影響されることを表す．このケースの場合，最適解を列挙法で厳密的に求めるのに計算量はオーダー  $O(2^N)$  が要ることが分かる．それで本章で近似解法である平均場近似法 MFA を援用してこの問題を解決する．

## 4.2 統計力学に基づく平均場近似

### 4.2.1 統計力学アプローチ (Statistical Mechanical Approach)

統計力学とは，システムのミクロな状態は確率的に出現することを基に，マクロな性質を導き出すための学問である．つまり，統計力学を確率論 (Probability Theory) の角度から解釈することができる．システムのミクロな状態  $\mathbf{S} = \{S_1, \dots, S_N\}^T$  を考え，状態  $\mathbf{S}$  が実現される確率分布  $P(\mathbf{S})$  が与えられているとき，目標関数  $\mathcal{H}(\mathbf{S})$ ，制約条件  $1 = \sum_{\mathbf{S}} P(\mathbf{S})$  があれば，期待値  $E = \sum_{\mathbf{S}} P(\mathbf{S}) \mathcal{H}(\mathbf{S})$  で定義される．ラグランジュ未定乗数法 (Lagrange multiplier method) により，ラグランジュ関数  $L$  は  $L = -\sum_{\mathbf{S}} P(\mathbf{S}) \log P(\mathbf{S}) + \lambda(1 - \sum_{\mathbf{S}} P(\mathbf{S})) + \beta(E - \sum_{\mathbf{S}} P(\mathbf{S}) \mathcal{H}(\mathbf{S}))$  のように定義できる． $P(\mathbf{S}), \beta, \lambda$  に対して関数  $L$  の微分の極値  $\frac{\partial L}{\partial P(\mathbf{S})} = 0, \frac{\partial L}{\partial \beta} = 0, \frac{\partial L}{\partial \lambda} = 0$  をとったとき，最適確率分布  $P(\mathbf{S}) = \frac{e^{-\beta \mathcal{H}(\mathbf{S})}}{Z}$  が導出される．ただし，パラメータ  $\beta$  の働きはシミュレート・アニーリング法における温度  $T$  の効果と一致する [50][51][52]． $\beta = \frac{1}{T}$  の関係により， $\beta$  の増大に従って分布はどんどん平衡状態に収束する．また，正規化定数  $Z = \sum_{\mathbf{S}} e^{-\beta \mathcal{H}(\mathbf{S})}$  で分配関数を定義する．符号  $\sum_{\mathbf{S}}$  は，すべて  $(2^N)$  個の状態を考慮する必要があることを意味する．

### 4.2.2 最適化 (Optimization)

$S_i$  の期待値を  $m_i$  で表し, すなわち  $m_i = \sum_{\mathbf{S}} P(\mathbf{S}) S_i$  である.  $\beta$  が十分に大きければ, 期待値  $\mathbf{m} = \{m_1, \dots, m_N\}^T \in \mathbb{R}^N$  が最適解  $\mathbf{S}^*$  に収束することが望まれている. ここで状態数  $i = 1, 2, 3$  の簡単な例を挙げて説明する. ボルツマン分布  $P_i = \frac{e^{-\beta \mathcal{H}_i}}{1+2^\beta+3^\beta}$ , 状態  $\mathcal{H}_1 = -\log 1, \mathcal{H}_2 = -\log 2, \mathcal{H}_3 = -\log 3$  と設定する. このケースの場合, 温度  $T \rightarrow 0$ , すなわち  $\beta = \frac{1}{T} \rightarrow \infty$  のとき,

$$P_i = \begin{cases} 1 & i = 3 \\ 0 & \text{otherwise} \end{cases} \quad (4.10)$$

を得る. 具体的に,  $\beta \rightarrow \infty$  のとき, 状態 3 の確率  $P_3 = \frac{3^\beta}{1+2^\beta+3^\beta} \rightarrow 1$  であることを意味する. 一方, 状態 1 と 2 の確率  $P_1 = \frac{1}{1+2^\beta+3^\beta} \rightarrow 0, P_2 = \frac{2^\beta}{1+2^\beta+3^\beta} \rightarrow 0$  であることが分かる. この証明から  $\mathcal{H}_1 > \mathcal{H}_2 > \mathcal{H}_3$  における状態 3 が最適状態として見出された. 要するに,

$$P(\mathbf{S}) = \begin{cases} 1 & \mathbf{S} = \arg \min_{\mathbf{S}} \mathcal{H}(\mathbf{S}) \\ 0 & \text{otherwise} \end{cases} \quad (4.11)$$

の結論が得られる. そして, 上記の証明により,  $\beta \rightarrow \infty$  となるとき, 期待値  $\mathbf{m} = \{m_1, \dots, m_N\}^T \in \mathbb{R}^N$  が最適解  $\mathbf{S}^*$  に近づくようになることが分かる.

また, 式 (4.4) に対して次の関係

$$m_i = -\frac{\partial F}{\partial d_i} \quad (4.12)$$

が存在する. ただし統計力学の分野では次に示す自由エネルギー (Helmholtz free energy) と呼ばれる  $F$  が

$$F = -\frac{1}{\beta} \log Z \quad (4.13)$$

である [53]. しかしながら,  $\sum_{\mathbf{S}} = \sum_{S_1=\pm 1} \sum_{S_2=\pm 1} \dots \sum_{S_N=\pm 1}$  は全確率変数の全配位に対する和を意味するから, 分配関数  $Z(\beta) = \sum_{\mathbf{S}} e^{-\beta \mathcal{H}(\mathbf{S})}$  の計算量はすべての  $J_{ij}$  が 0 でない限り, インターフェース数  $N$  の増加によって  $O(2^N)$  のオーダーで増大してしまう NP 困難な問題となる. 全数探索で  $P(\mathbf{S})$  の計算が必要である試行分布  $\langle S_i \rangle$  や  $\langle S_i S_j \rangle$  の計算は厳密に実行することは現実的ではないことが分かる.

一方, 状態  $\mathbf{S}$  の平衡状態  $\mathbf{S}^*$  の実現は, エルコード理論を応用する確率アルゴリズムを考える [54]. すべての  $K$  個のビットのなかから一様確率に従ってビット  $k$  を選び, 残りのビット全員はそのままにして, 選んだビット  $k$  の値  $S_k$  を確率

$$P_k = \frac{1}{1 + \exp\left(-\sum_{i=1}^K J_{ki} S_i - d_k\right)} \quad (4.14)$$

に従い 1 にし,  $1 - P_k$  の確率で 0 にすることを繰り返す.

このアルゴリズムは次のような意味をもっている． $S_k$  以外の  $K - 1$  個のビット  $S_{i \in K \setminus k}$  の値が固定されたとき  $S_k = 1$  となる条件つき確率  $p(S_k = 1 | S_{i \in K \setminus k}, J_{ki}, d_k)$  は，

$$\begin{aligned} p(S_k = 1 | S_{i \in K \setminus k}, J_{ki}, d_k) &= \frac{p(S_k = 1, S_{i \in K \setminus k} | J_{ki}, d_k)}{p(S_k = 1, S_{i \in K \setminus k} | J_{ki}, d_k) + p(S_k = 0, S_{i \in K \setminus k} | J_{ki}, d_k)} \\ &= \frac{1}{1 + \frac{p(S_k=0, S_{i \in K \setminus k} | J_{ki}, d_k)}{p(S_k=1, S_{i \in K \setminus k} | J_{ki}, d_k)}} \end{aligned} \quad (4.15)$$

である．ここで

$$\begin{aligned} \frac{p(S_k = 0, S_{i \in K \setminus k} | J_{ki}, d_k)}{p(S_k = 1, S_{i \in K \setminus k} | J_{ki}, d_k)} &= \exp(\mathcal{H}(p(S_k = 1, S_{i \in K \setminus k} | J_{ki}, d_k)) - \mathcal{H}(p(S_k = 0, S_{i \in K \setminus k} | J_{ki}, d_k))) \\ &= \exp\left(-\sum_{i=1}^K J_{ki} S_i - d_k\right) \end{aligned} \quad (4.16)$$

であるから， $p(S_k = 1 | S_{i \in K \setminus k}, J_{ki}, d_k) = P_k$  である．

上記のマルコフ連鎖過程 (Markov Process) において十分な時間を与えれば，時間が経過しても確率  $P_k$  が変化しない平衡分布に収束できる．しかし，モデルのサイズが大きくなることに従ってマルコフ連鎖過程の時間も指数的に長くなっていく．そのため，全数探索もマルコフ連鎖も解決できない試行分布  $\langle S_i \rangle$  や  $\langle S_i S_j \rangle$  の計算は効率的な統計的近似計算手法を用いる必要がある．

### 4.2.3 平均場近似 (Mean Field Approximation)

一般的に， $S_i$  や  $S_i S_j$  の平均値の評価の計算量は増大するため，いくつかの効率的な近似計算手法がある [53]．統計力学の観点に基づく平均場近似が扱いも容易であり汎用性も高い手法である．図 4.5 に示すように，平均場近似法によりインターフェース  $i$  の試行分布の平均値  $m_i = \langle S_i \rangle$  の係数は，インターフェース  $i$  そのものの単独コスト  $d_i$  とほかのすべての近傍インターフェース  $j$  からの相互作用  $J_{ij} S_j$ ，(ここで  $\beta \rightarrow \infty$  とし， $\delta S = S - m$  を無視して， $J_{ij} S_j$  を  $J_{ij} m_j$  で改めて表す) の総和  $\sum_{j=1}^N J_{ij} m_j$  という両方の和

$$d'_i = d_i + \sum_{j \neq i}^N J_{ij} m_j \quad (4.17)$$

に決められる．そして式 (4.4) のコスト関数は，

$$\mathcal{H}(\mathbf{S}) \simeq -\mathbf{S}^T (\mathbf{d} + \mathbf{J} \mathbf{m}) + \frac{1}{2} \mathbf{m}^T \mathbf{J} \mathbf{m} \quad (4.18)$$

に近似することができる．つまり，式 (4.18) は平均場近似の表現式である．そしてこれまでの議論により，分配関数  $Z$  は式 (4.19) のように相対的に簡単に計算できる．

$$\log Z \simeq \sum_{i=1}^N \log \left( 2 \cosh \left[ \beta d_i + \beta \sum_{j=1}^N J_{ij} m_j \right] \right) - \frac{\beta}{2} \mathbf{m}^T \mathbf{J} \mathbf{m} \quad (4.19)$$

ただしここで  $\theta_i = \beta d_i + \beta \sum_{j=1}^N J_{ij} m_j$  とし,  $\sum_{S_i=\pm 1} e^{\theta_i S_i} = e^{\theta_i} + e^{-\theta_i} = 2 \cosh \theta_i$  が用いられる. そして, 平均場近似のもとで最適解  $m_i$  は,

$$m_i = \tanh \left( \beta d_i + \beta \sum_{j=1}^N J_{ij} m_j \right) \quad (4.20)$$

で得られる. このときの計算量は  $O(2^N)$  から  $O(N^2)$  へ低減することを実現できる.

上の議論により, 以下は平均場近似のもとで最適の製品設計を求めたアルゴリズムをまとめる.

**ステップ0** 初期設定:

1.  $M$  個の機能デバイスがあるとする.
2. 任意2つの機能デバイス間のインターフェースにラベル  $i = 1, \dots, N$  を与える.
3. インターフェース  $i$  の単独コスト  $d_i$  と相関コスト  $J_{ij}$  を確定する.
4. 適当に収束条件の判定値, 小さな正の定数  $\delta > 0$  を入力する.
5. 初期状態  $m_i^0 = 0$  と更新回数  $t = 0$  を仮定する.

**ステップ1**  $m_i^{t+1}$  の値は,

$$m_i^{t+1} = \tanh \left( \beta d_i + \beta \sum_{j=1}^N J_{ij} m_j^t \right)$$

で与えられる.

**ステップ2**  $\Delta = \sum_{i=1}^N |m_i^{t+1} - m_i^t|$  が評価され,  $t \rightarrow t+1$ ,  $\Delta > \delta$  ならステップ1に戻して繰り返す. そうでないとステップ3に進む.

**ステップ3** このときの収束状態を最適解として出力する.

## 4.3 数値実験

### 4.3.1 機能デバイス数 $M = 4$ から $M = 7$ の小型例

ここで小型例を用いて提案手法の有効性を検証する. このレベルのサイズに対して, 例えば, 機能デバイス数  $M = 4$  のときインターフェース数  $N = 6$ , 機能デバイス数  $M = 7$  のときインターフェース数  $N = 21$  であることにより, 全数探索によって  $2^6 = 64$  から  $2^{21} = 2097152$  まで解空間のすべての状態を評価することができる. これで厳密解を求められる. しかしながら, 単独コスト  $d_i$  と相関コスト  $J_{ij}$  の値は設計現場で現実の物理意味を持っているはずであるが, ここでまだ精確に定義されていない. 例えば, 正の単独コストはこのインターフェースが存在すると機能デバイスの機能の発揮に悪影響を与えることを意味しながら, 負の単独コストはこのインターフェースが存在すると機能がうまくはたらせることを意味する. また一方, 負の相互作用はこのイ

インターフェースの組において, 1つのインターフェースの存在は必ずほかの1つのインターフェースの機能に悪影響を与えることを意味する.

そのため, ここで数理的に単独コスト  $d$  と相互作用  $J$  のデータは正規分布に従って生成されるとする. 例えば, 単独コスト  $S$  が平均  $\mu = 0$ , 分散  $\sigma^2 = 1$  の正規分布に従って生成するとし, 変数  $N$  がいくら大きくなっても相関コスト  $J_{ij}$  のオーダーと単独コスト  $d_i$  のオーダー  $O(1)$  と一致することを維持するために, 相互作用  $J_{ij}$  が平均  $\mu = 0$ , 分散  $\sigma^2 = \frac{1}{N}$  の正規分布に従って生成するとする. また, ここで正規分布に従うデータは一様分布に従う確率変数から正規分布に従う確率変数を生成させる手法であるボックス・ミュラー法 (Box-Muller transform) を用いて生成される [45]. ボックス・ミュラー法では正規分布  $N(\mu, \sigma^2)$  をす正規乱数  $\eta$  を生成する場合, 式 (4.21)

$$\eta = \mu + \sigma \sqrt{-2 \log X} \sin(2\pi Y) \quad (4.21)$$

あるいは式 (4.22)

$$\eta = \mu + \sigma \sqrt{-2 \log X} \cos(2\pi Y) \quad (4.22)$$

で表せる. ただし  $X$  と  $Y$  は  $[0, 1]$  間の独立な一様乱数である.

ボックス・ミュラー法によって 2000 セットの単独コスト  $d$  と相互作用  $J$  を生成する. この 2000 セットの数値実験がすべて全数探索法で得られた厳密解と比較し, 提案手法の有効性を証明する. 評価指標として, ここで  $q_{soft} = \frac{1}{N} \mathbf{m}^T \mathbf{S}^*$ ,  $q_{hard} = \frac{1}{N} \mathbf{M}^T \mathbf{S}^*$  と,

$$\cos \theta = \frac{\mathbf{m}^T \mathbf{S}^*}{|\mathbf{m}^T| |\mathbf{S}^*|} \quad (4.23)$$

を用いる. ただし  $\mathbf{S}^* = \{S_1^*, \dots, S_N^*\}^T \in \{\pm 1\}^N$  が全数探索による最適解であり,  $\mathbf{M} = \{M_1, \dots, M_N\}^T \in \{\pm 1\}^N$  が平均場近似による解 (hard decision solution) である. また,  $M_i = \text{sgn}(m_i)$  の関係があり, インターフェース  $i$  のハード解 (hard decision solution)  $M_i$  が式 (4.20) のソフト解 (soft decision solution)  $m_i$  によって得ることを意味する. また, 式 (4.23) における  $\theta$  が  $\mathbf{S}^*$  と  $\mathbf{m}$  間の角度 (angle) を表すことにより,  $\cos \theta$  が両方の近似さの度合を表すことができる.  $\cos \theta = 1$  になると最適解と一致することを意味する. それで, 評価指標  $q_{hard}$  によって, 最適解との一致する成功確率は,

$$P_{suc} = 1 - \frac{1}{N} \sum_{i=1}^N \frac{(S_i^* - M_i)^2}{4} = \frac{1 + q_{hard}}{2} \quad (4.24)$$

ように定義される.

機能デバイス数  $M = 4$ , インターフェース数  $N = 6$  から機能デバイス数  $M = 7$ , インターフェース数  $N = 21$  までの数値実験の結果として,  $q_{soft}$ ,  $q_{hard}$  と  $\cos \theta$  の平均値を図 4.6 に示した. 横軸の  $\beta$  値がどんどん増えていることに従って, 縦軸の  $q_{soft}$ ,  $q_{hard}$  と  $\cos \theta$  の平均値も安定に 1 に近づいている. また, たとえ  $\beta$  値が有限になっても  $q_{hard}$  からハード解  $M_i = \text{sgn}(m_i)$  が最適解を成功に見つけることが分かった. 要するに, これより提案手法が高い確率で本当の最適解を見出すことができることを証明した. さらに, 異なる  $M$  のもとで  $q_{soft}$ ,  $q_{hard}$ ,  $\cos \theta$  と  $\beta$  の関係を表 4.1, 表 4.2 と表 4.3 で詳しく記録した.

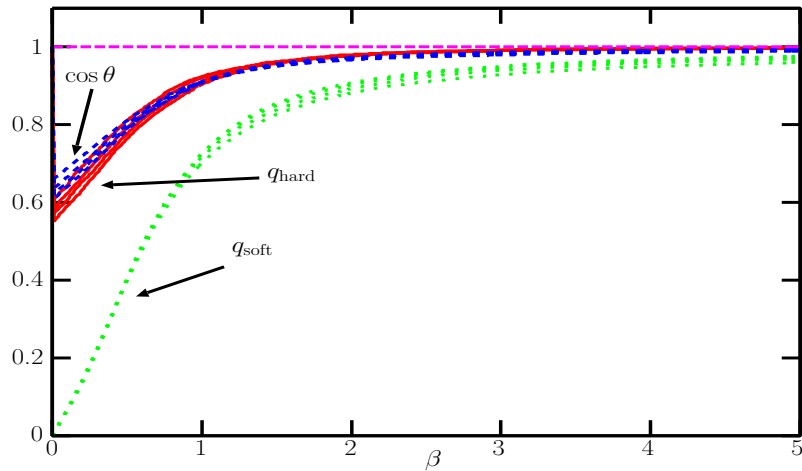


図 4.6: 機能デバイス数  $M = 4$ , インターフェース数  $N = 6$  から機能デバイス数  $M = 7$ , インターフェース数  $N = 21$  までの数値実験の結果として,  $\beta$  の増大につれて  $q_{soft}$ ,  $q_{hard}$ ,  $\cos \theta$  の遷移図

表 4.1: 異なる  $M$  のもとで  $q_{soft}$  と  $\beta$  の関係

$q_{soft}$	$\beta = 0.1$	$\beta = 0.5$	$\beta = 1.0$	$\beta = 2.0$	$\beta = 5.0$
$M = 4$	0.068	0.399	0.701	0.882	0.961
$M = 5$	0.066	0.403	0.717	0.893	0.969
$M = 6$	0.065	0.403	0.724	0.903	0.973
$M = 7$	0.064	0.401	0.726	0.908	0.976

### 4.3.2 均一 (Homogeneous) の例

均一のシステム  $d_i > 0, J_{ij} > 0$  に対して, 最適解は自明で  $S_i^* = 1$  であることが分かる. このとき, 組  $S_i = S_j = 1$  と  $S_i = S_j = -1$  がほかの 2 組  $S_i = 1, S_j = -1$  と  $S_i = -1, S_j = 1$  よりもっと理想な (より小さい) コスト関数  $\mathcal{H}(S)$  値をもたらすことによって, 最適解は,

$$m = \tanh(\beta \mathbf{d} + \beta J m) \quad (4.25)$$

で近似することができる. ただし, システムの対称性によって,  $m_i = m$  と  $m = \frac{1}{N} \sum_{i=1}^N m_i$  とする.

$M = 100, N = 4950$  のとき,  $d_i = 1, J_{ij} = \frac{1}{N}$  の数値例の結果を図 4.7 に示した. 横軸は  $\beta$  であり, 縦軸は  $m = \frac{1}{N} \sum_{i=1}^N m_i$  である.  $\beta$  の増大につれて  $m = \frac{1}{N} \sum_{i=1}^N m_i$  は 1 に近づいていることが分かる.  $m = 1$  になるとき, すべての  $m_i$  が  $m_i = 1$  に満たさないといけないうことによって, 自明の最適解  $S_i^* = 1$  を見つけたといえる.

表 4.2: 異なる  $M$  のもとで  $q_{hard}$  と  $\beta$  の関係

$q_{hard}$	$\beta = 0.1$	$\beta = 0.5$	$\beta = 1.0$	$\beta = 2.0$	$\beta = 5.0$
$M = 4$	0.644	0.812	0.924	0.981	0.998
$M = 5$	0.624	0.793	0.921	0.978	0.998
$M = 6$	0.607	0.777	0.911	0.978	0.997
$M = 7$	0.502	0.764	0.904	0.978	0.998

表 4.3: 異なる  $M$  のもとで  $\cos \theta$  と  $\beta$  の関係

$q_{soft}$	$\beta = 0.1$	$\beta = 0.5$	$\beta = 1.0$	$\beta = 2.0$	$\beta = 5.0$
$M = 4$	0.689	0.811	0.909	0.967	0.989
$M = 5$	0.666	0.801	0.912	0.969	0.992
$M = 6$	0.652	0.792	0.909	0.972	0.993
$M = 7$	0.640	0.786	0.908	0.973	0.994

### 4.3.3 機能デバイス数 $M = 100$ の大型例

非均一の大規模問題  $M = 100, N = 4950$  に対して、解空間の大きさは全数探索によって扱えない  $2^N \simeq 1.25 \times 10^{1490}$  である。しかし、平均場近似によって計算オーダー  $O(N^2)$  に低減できることによって、この規模問題の解析を可能にした。そして  $d_i \sim N(0, 1)$ ,  $J_{ij} \sim N(0, \frac{1}{N})$  とする。問題規模のため、上の3つの評価指標  $q_{soft}, q_{hard}, \cos \theta$  を厳密的に計算することができない。そのためここでほかの評価指標、 $m_i$  の密度関数  $P(m_i)$  で提案手法の近似効果を検証する。

$$\beta = 0.1$$

$\beta \simeq 0$  のケースに対して、式 (4.20) により  $m_i \simeq 0$  が得られる。図 4.8 に  $m_i$  の密度関数  $P(m_i)$  は正規分布  $\mu = 0, \sigma^2 = 0.01$  に従うことを示した。  $\theta$  が小さいとき、 $\tanh \theta \simeq \theta$  の関係によって、 $\mathbf{m} \simeq \beta \mathbf{d} + \beta J \mathbf{m}$  が得られ、 $\mathbf{m} \simeq (1 + \beta J + \beta^2 J^2 + \dots), \beta \mathbf{d} \simeq \beta \mathbf{d} + O(\beta^2)$  が得られる。これで  $m_i$  が正規分布  $\mu = 0, \sigma^2 = \beta^2$  に従う証明を提供した。

$$\beta = 0.5$$

図 4.9 に示したように、 $m_i$  の密度関数  $P(m_i)$  は2つのピークのある分布  $P(m_i) \simeq (0.552 + 0.228m_i^2 + 0.670m_i^4 - 1.541m_i^6)e^{-0.293m_i^2}$  に従うことが分かった。  $\beta = 0.1$  に比べて  $\beta = 0.5$  のケースには、単独コストが存在するからボルツマン分布の対称性を損なうようになる。  $m_i = 0$  以外、多個の安定状態が存在することによって  $P(m_i)$  が図 4.9 の分布になる。

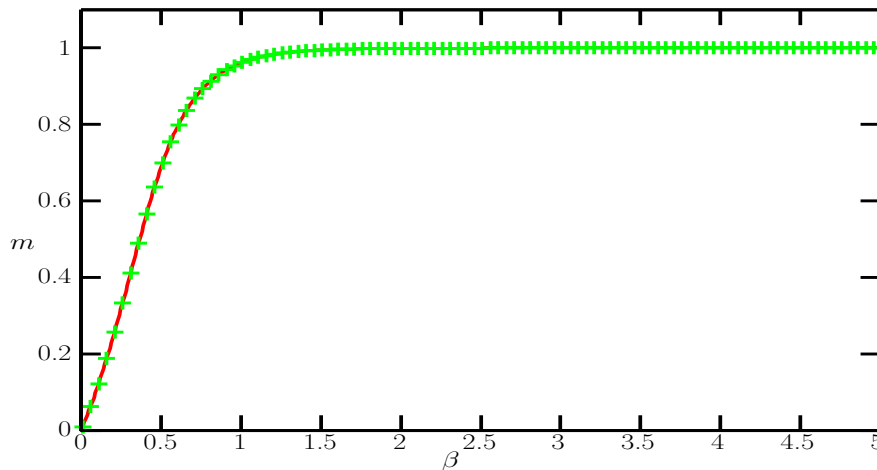


図 4.7: 機能デバイス数  $M = 100$ , インターフェース数  $N = 4950$  の大規模問題の結果として,  $\beta$  の増大につれて  $m = \frac{1}{N} \sum_{i=1}^N m_i$  の遷移図

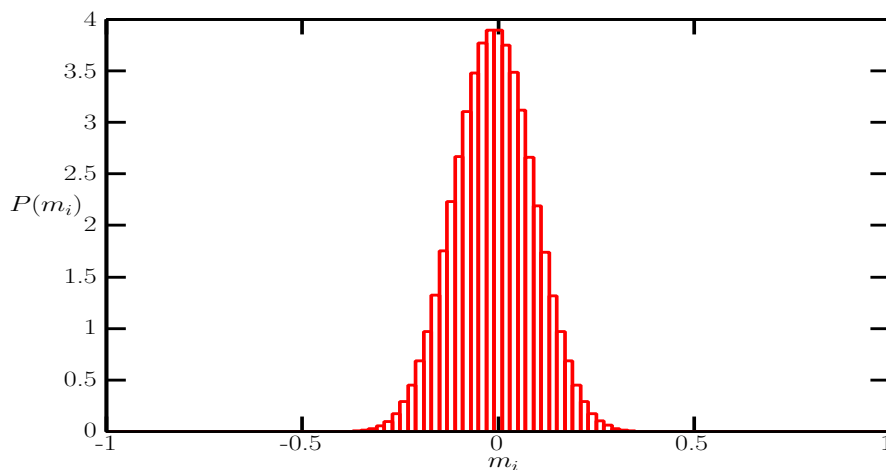


図 4.8:  $\beta = 0.1$  のとき  $m_i$  の密度関数  $P(m_i)$

$\beta = 1.0, \beta = 2.0$  と  $\beta = 5.0$

図 4.10 に  $\beta = 1.0, \beta = 2.0$  と  $\beta = 5.0$  のとき  $m_i$  の密度関数  $P(m_i)$  を示した.  $\beta$  の増大に従って,  $\lim_{\beta \rightarrow \infty} \tanh(\beta\mu) = \text{sgn}(\mu)$  によって,  $m_i = \lim_{\beta \rightarrow \infty} \tanh\left(\beta d_i + \beta \sum_{j=1}^N J_{ij} m_j\right) = \text{sgn}\left(d_i + \sum_{j=1}^N J_{ij} m_j\right)$  であることが得られる. つまり,  $m_i$  はどんどん  $+1$  か  $-1$  に収束することを証明した.

#### 4.3.4 応用例

今のモデルはインターフェースそのものの単独コスト  $d_i$  とインターフェースの組の相互作用  $J_{ij}$  という2つの効用しか考えられないものであるが, もっと一般的な状況を考えると, 3つあるいはそれ以上のインターフェースのグループ間の効用も考える必要がある. このように考えると, 提



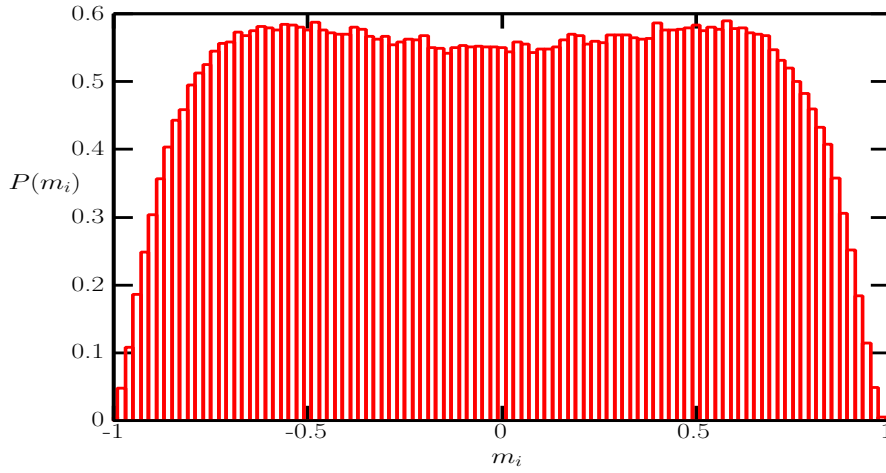


図 4.9:  $\beta = 0.5$  のとき  $m_i$  の密度関数  $P(m_i)$

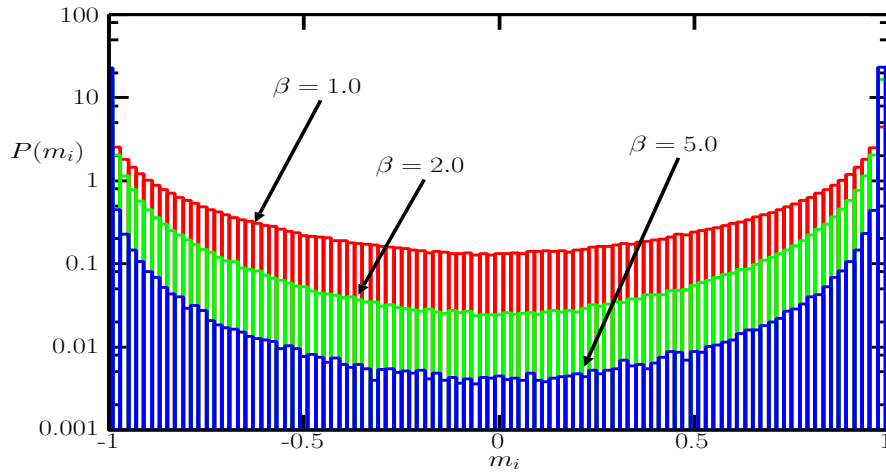


図 4.10:  $\beta = 1.0, \beta = 2.0, \beta = 5.0$  のとき  $m_i$  の密度関数  $P(m_i)$

案モデルは,

$$\mathcal{H}(\mathbf{S}) = - \sum_{i=1}^N d_i S_i - \sum_{i>j}^N J_{ij} S_i S_j - \sum_{i>j>k}^N r_{ijk} S_i S_j S_k \quad (4.26)$$

ように展開することができる．ただし対称な  $r_{ijk} = r_{ikj} = \dots = r_{kji}$  は3つのインターフェースのグループの効用であり,  $J_{ii}$  ような自身への効用  $r_{iij} = r_{iji} = \dots = 0$  とする．さらに, 符号  $\sum_{i>j>k}$  は添え字  $i > j > k$  に満たすすべてのグループを表示する．平均場近似によって式 (4.25) における設計  $S_i$  の期待値  $m_i$  は,

$$m_i = \tanh(\beta d'_i), \quad (4.27)$$

$$d'_i = d_i + \sum_{j=1}^N J_{ij} m_j + \frac{1}{2} \sum_{j=1}^N \sum_{k=1}^N r_{ijk} m_j m_k \quad (4.28)$$

ように定義することができる．

## 4.4 まとめ

我々は、製品の最適設計を評価するコスト関数を提案した上に、適用の平均場近似を用いてこの2次組合せ最適化モデルの大規模問題(機能デバイス数  $M = 100$ , インターフェース数  $N = 4950$ )を成功に解析した。平均場近似法により、インターフェース  $i$  の試行分布の平均値  $m_i = \langle S_i \rangle = \sum_{\mathbf{S}} P(\mathbf{S}) S_i$  は、インターフェース  $i$  そのものの単独コスト  $d_i$  とほかのすべての近傍インターフェース  $j$  からの相関コスト  $J_{ij} m_j$  の総和  $\sum_{j=1}^N J_{ij} m_j$  という両方の和に決められる。これでモデルの計算量は  $O(2^N)$  から  $O(N^2)$  へ低減することを実現した。そして3つの数値実験を用いて提案手法の効果を検証した。(a) 機能デバイス  $M$  が小さい  $M = 4 \sim 7$  数値例に対して、 $\beta$  値が十分に大きくなれば提案手法により高い確率で最適解を見出すことができることを証明した。(b) 均一の数値例に対して、式(4.20)から式(4.25)を推定でき、式(4.25)で平均値  $m$  を評価することができることを証明した。(c) 非均一の機能デバイス数  $M = 100$ , インターフェース数  $N = 4950$  の全数探索によって扱えない大規模数値例に対して、各  $\beta$  値のとき  $m_i$  の密度関数  $P(m_i)$  を評価することにより提案手法の有効性を示した。

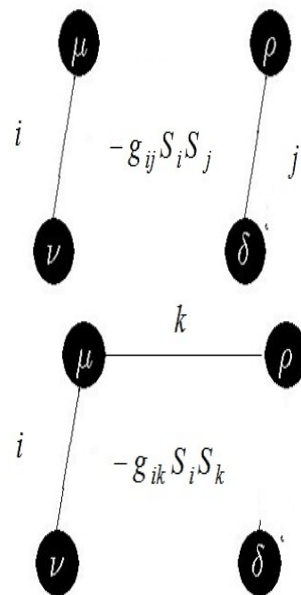


図 4.11: 直観的には、上と下の設計案は異なる分布を持つ可能性がある

将来の研究課題について、まず式(4.4)における決定変数は  $N$  個の単独コスト  $d_i$  と  $\frac{N(N-1)}{2}$  個の相関コスト  $J_{ij}$  であることによって、もし少なくとも  $N + \frac{N(N-1)}{2}$  組の製品設計案  $\mathbf{S}$  とコスト関数値  $\mathcal{H}(\mathbf{S})$  を用意すれば、これらの情報から最小2乗法 (least squares method) を用いて  $d_i$  と  $J_{ij}$  を近似に推定することができることが分かる。しかし、この仮説は理想の状態であるため、問題に合わせる推定モデルと適当なアルゴリズムを考える必要がある。次は、今の単独コスト  $d_i$  と相互作用  $J_{ij}$  の値は互いに独立で正規分布に従うとしたが、しかし現実の状況を考えると、例えば図4.11のように、左の設計案のインターフェースの組  $ij$  は共有の機能デバイスが存在しないことに対して、右の設計案のインターフェースの組  $ik$  は共有の機能デバイス  $\mu$  を持っていることが分

かる．直観的には，2つの設計案は異なる分布に従う可能性がある．そのため，この議論に従って提案モデルを改善する必要がある．



## 第5章 製品設計の組合せ最適化モデルに適用する一点変動近傍アルゴリズムの提案

前章に議論した統計力学の手法以外，2次組合せ最適化モデルの近似解析にしばしば用いられるのがメタヒューリスティクス（発見的な手法）における多種の探索法であることも知られていた．例えば主に遺伝的アルゴリズム，シミュレーテッド・エボリューション焼きなまし法，タブー探索，蟻コロニー最適化，Greedy Randomized Adaptive Search Procedure (GRASP)，及び一点変動近傍法などがある．

反復局所探索法 (Iterated Local Search) である変動近傍法 (Variable Neighborhood Search) を用いて大規模の組合せ最適化問題を解いてみた [55][56]．組合せ最適化問題において複数の近傍解を定義し，暫定解（それまでに得られた最良解）に対しある近傍解から1つの初期解を選択し，その解に対して局所探索を適用するという手順を繰り返す手法．しかし，一般的にメタヒューリスティクスはどんな問題にも汎用的な探索を行えるが，それらを有効に使用することで問題特徴に合わせる探索を設計する必要がある．本章で提案した2次組合せ最適化モデルの特徴に合わせる一点変動近傍法 (One-Bit-Variant Neighborhood Algorithm) を設計して，製品設計の大規模問題の近似探索を行う．

本章の構成は以下の通りである．1節では製品設計の2次組合せ最適化モデルの特徴を分析する．2節では1節の特徴に合わせる一点変動近傍法を設計する．3節で厳密解を見出す分枝限定法 BAB，メタヒューリスティックである遺伝的アルゴリズム GA との比較によって，小型，中型と大型の数値例から提案手法の有効性を検証する．最後に4節ではこの章をまとめる．

### 5.1 製品設計の2次組合せ最適化モデルの特徴

まず，ここで議論する2次組合せ最適化モデルは，前章で議論したコスト関数  $\mathcal{H}(S)$  でなく，第2章で議論した式 (2.2)～式 (2.5) を援用することを注意する．ここで決定変数は  $x_i$  を  $w_i$  で改めて

表すことにする．また，ほかの要素の定義は第2章のモデルと一致する．

$$\text{Max} \quad TR(\mathbf{w}) = \sum_{i=1}^N \sum_{j=1}^N w_i g_{ij} w_j + \sum_{i=1}^N h_i w_i \quad (5.1)$$

$$\text{s.t.} \quad 0 \leq \sum_{i=1}^N w_i \leq a, a \leq N \quad (5.2)$$

$$w_i = \{0, 1\}, i = 1, 2, \dots, N \quad (5.3)$$

一般的に，汎用的な変動近傍法において，暫定解  $\mathbf{w}$  とその近傍解  $\mathbf{w}'$  を比較してどちらが次の暫定解とすることを判断する指標は目標関数値  $TR(\mathbf{w})$  と  $TR(\mathbf{w}')$  である．例えば， $TR(\mathbf{w}) < TR(\mathbf{w}')$  に満たせれば，近傍解  $\mathbf{w}'$  をある確率で次の暫定解として更新する．しかし，この評価指標  $TR(\mathbf{w})$  による変動近傍法の計算量はオーダー  $O(2^N)$  であり，全数探索による計算量と同じレベルであることが分かる．そのため，ここでモデルの特徴に合わせる多項式時間内に探索を行える変動近傍法を設計し，有効な近似探索を達成することが本研究の貢献の1つである．

### 定義1

製品設計の2次組合せ最適化モデル  $TR(\mathbf{w})$  の展開は，

$$\begin{aligned} TR(\mathbf{w}) &= \sum_{i=1}^N h_i w_i + \sum_{i=1}^N \sum_{j=1}^N w_i g_{ij} w_j \\ &= (h_1 w_1 + g_{11} w_1 w_1 + \dots + g_{1s} w_s w_1 + \dots + g_{1N} w_N w_1) \\ &\quad + \dots \\ &= (h_s w_s + g_{s1} w_1 w_s + \dots + g_{ss} w_s w_s + \dots + g_{sN} w_N w_s) \\ &\quad + \dots \\ &= (h_N w_N + g_{N1} w_1 w_N + \dots + g_{Ns} w_s w_N + \dots + g_{NN} w_N w_N) \end{aligned} \quad (5.4)$$

ようになる．ただし，

1. 行  $s$  に関する項目の総和は  $r_s$  で定義され，

$$r_s = h_s w_s + \sum_{j \neq s}^N g_{sj} w_j w_s \quad (5.5)$$

で与えられる．

2. 列  $s$  に関する項目の総和は  $c_s$  で定義され，

$$c_s = \sum_{j \neq s}^N g_{js} w_s w_j \quad (5.6)$$

で与えられる．

3. 残りの項目の総和は  $U_s$  で定義される .

また , 上の  $r_s$  と  $c_s$  によって , インターフェース  $s$  の事前知識 (Prior knowledge)  $pk_s$  は ,

$$\begin{aligned} pk_s &= r_s + c_s \\ &= h_s + \sum_{j \neq s}^N g_{sj} w_j + \sum_{j \neq s}^N g_{js} w_j \end{aligned} \quad (5.7)$$

で定義される . 事前知識  $pk_s$  はここで暫定解とその近傍解を比較する新しい評価指標とする .

### 解釈

式 (5.4) を展開すると3つの部分に分けられると考える . インターフェース  $s$  に対して , 展開式における  $s$  行目は1個の単独収益項  $h_s w_s$  と  $(N - 1)$  個の相互作用項  $g_{sj} w_j w_s$  (インターフェース  $s$  からインターフェース  $j$  への影響を意味する) の総和である . 同様に , 展開式における  $s$  列目は  $(N - 1)$  個の相互作用項  $g_{js} w_s w_j$  (インターフェース  $j$  からインターフェース  $s$  への影響を意味する) の総和である . 添え字  $s$  が付かない残りの項目は , インターフェース  $s$  とは関係なく ,  $U_s$  でその総和を表すことができる .

インターフェース  $s$  の意思決定  $w_s$  は  $w_s \rightarrow w'_s$  に変わるとき , 暫定解  $\mathbf{w} = \{w_1, \dots, w_s, \dots, w_N\}$  とその一点変動近傍解  $\mathbf{w}' = \{w_1, \dots, w'_s, \dots, w_N\}$  の目標関数値  $TR(\mathbf{w})$  と  $TR(\mathbf{w}')$  を , それぞれ展開すると , 両方の区別は  $s$  行目と  $s$  列目の  $(2N - 1)$  個の項目しかないことが分かる . 残りの  $U_s$  の部分は  $TR(\mathbf{w})$  と  $TR(\mathbf{w}')$  の中に一致することによって , 計算において省略することが可能である . そのため , ここで式 (5.7) の項目和をインターフェース  $s$  の事前知識  $pk_s$  と呼ばれ , この事前知識で暫定解とその近傍解の優劣を評価する新しい指標となると考える . つまり , 暫定解  $\mathbf{w} = \{w_1, \dots, w_s, \dots, w_N\}$  とその一点変動近傍解  $\mathbf{w}' = \{w_1, \dots, w'_s, \dots, w_N\}$  のどちらが望ましいと聞くと , 目標関数値  $TR(\mathbf{w})$  と  $TR(\mathbf{w}')$  でなく , 事前知識  $pk_s$  から判断することである .

## 性質 1

暫定解  $\mathbf{w} = \{w_1, \dots, w_s, \dots, w_N\}$  とその一点変動近傍解  $\mathbf{w}' = \{w_1, \dots, w'_s, \dots, w_N\}$  の優劣を評価する指標は増量  $\Delta TR_s$  である .

$$TR(\mathbf{w}) = r_s(w_s) + c_s(w_s) + U_s \quad (5.8)$$

$$TR(\mathbf{w}'_s) = r_s(w'_s) + c_s(w'_s) + U_s \quad (5.9)$$

$$\begin{aligned} \Delta TR_s &= TR(\mathbf{w}'_s) - TR(\mathbf{w}) \\ &= (r_s(w'_s) - r_s(w_s)) + (c_s(w'_s) - c_s(w_s)) + (U_s - U_s) \\ &= \left( h_s + \sum_{j \neq s}^N g_{sj} w_j \right) (w'_s - w_s) + \sum_{j \neq s}^N g_{js} w_j (w'_s - w_s) \\ &= \left( h_s + \sum_{j \neq s}^N (g_{sj} + g_{js}) w_j \right) (w'_s - w_s) \end{aligned} \quad (5.10)$$

で与えられる . 式 (5.10) における  $\left( h_s + \sum_{j \neq s}^N (g_{sj} + g_{js}) w_j \right)$  の部分は事前知識  $pk_s$  である .

## 解釈

式 (5.8) と式 (5.9) はそれぞれ暫定解  $\mathbf{w}$  と  $\mathbf{w}'_s$  の目標関数値である . 式 (5.10) はインターフェース  $s$  の値を  $w_s \rightarrow w'_s$  に変わるとき増量  $\Delta TR_s$  の計算式である . 例えば , インターフェース  $i$  が存在している ( $w_s = 1$ ) とすると , その近傍解におけるインターフェース  $i$  が存在しない ( $w'_s = 0$ ) ことが分かる .

$$w_s = 1 \rightarrow w'_s = 0 \quad (5.11)$$

$$w'_s - w_s = -1 \quad (5.12)$$

が得られる . このとき ,

$$\Delta TR_s = pk_s (w'_s - w_s) \quad (5.13)$$

であるため ,  $pk_s < 0$  である限り増量  $\Delta TR_s > 0$  を保証でき , 近傍解が次の新しい暫定解として更新されることを判断できる . そして上の議論により , 事前知識  $pk_s$  で暫定解と近傍解を評価する計算量はオーダー  $O(N)$  ( $N - 1$  個の  $r_s$  の項目と  $N - 1$  個の  $c_s$  の項目 , 及び 1 つの項目  $h_s$  からなる  $2N - 1$  個の足し算) であることが分かる .

## 定義 2

すべてのインターフェース  $i$  の事前知識  $pk_i$  からなる事前知識のベクトルは  $PK = \{pk_1, \dots, pk_s, \dots, pk_N\}$  とする . このベクトルにおける最小の  $pk_i$  値を持つインターフェース  $i$  が最適のビット  $op$  と定義される . 最適のビット  $op$  の値を反転することでは , 最速ですべての近傍解の中 , 最適の近傍解を見出すことを保証できる .



## 性質 2

例として，すべてのビットが1である暫定解  $\mathbf{w} = \{1, 1, \dots, 1\}$  に対して，次の最適の近傍解を下記のように選ぶ．最小の  $pk_i$  値を持っているビット  $i$  が最適のビット  $op$  として選ばれてから，

$$\text{Min} (pk_i < 0) \rightarrow pk_{op} \quad (5.14)$$

$$w_{op} \rightarrow w'_{op} \quad (5.15)$$

最適のビット  $op$  の値  $w_{op} \rightarrow w'_{op}$  に反転する．

## 解釈

定義 1 に基づいて，暫定解は  $N$  次元の事前知識のベクトルは  $PK = \{pk_1, \dots, pk_i, \dots, pk_N\}$  を持っていることが分かる．例えば  $w_i = 1, pk_i < 0$  なら， $w_i : 1 \rightarrow 0$  をすることによって目標関数値  $TR(\mathbf{w})$  は暫定解の更新に従って  $-pk_i$  の増量を実現できる．

## 5.2 一点変動近傍法 (One-Bit-Variant Neighborhood Algorithm)

上の特徴と解釈により，計算の便利さを考えるため，ここで初期の暫定解をすべてのビットが1である  $\mathbf{w}^0 = \{1, 1, \dots, 1\}$  と設定する．次の最適の近傍解を見出すために，最小の  $pk_{op}$  値を持っているビット  $op$  を選び，ビット値  $w_{op}$  を反転することによって，目標関数値がこのときの最大の増量  $-pk_{op}$  を実現できる．提案する一点変動近傍法の計算量はオーダー  $O(2^N)$  から  $O(N^2)$  に低減できることが分かる．以下は一点変動近傍法のアルゴリズムをまとめる．

## ステップ 0 初期設定：

1.  $M$  個の機能デバイスがあるとする．
2. 任意 2 つの機能デバイス間のインターフェースにラベル  $i = 1, \dots, N$  を与える．
3. インターフェース  $i$  の単独コスト  $h_i$  と相互作用  $g_{ij}$  を入力する．
4. すべてのビットが1である初期解  $\mathbf{w}^0 = \{1, 1, \dots, 1\}$  を生成し，更新回数  $k = 0$  を仮定する．
5. 初期の事前知識ベクトル  $PK^0 = \{pk_1^0, \dots, pk_i^0, \dots, pk_N^0\}$  と初期の目標関数値  $TR(\mathbf{w}^0)$  を計算する．

## ステップ 1 近傍解が実行可能であるか否かは，

$$M - 1 \leq \sum_{i=1}^N w_i \leq N$$

で判断される．

ステップ2 最小の  $pk$  値をもたらすビット  $i$  を見出す .

$$\text{Min} (pk_i^k < 0) \rightarrow pk_{op}^k, k \rightarrow k + 1$$

ステップ3  $w_{op}^k$  を

$$w_{op}^k : 1 \rightarrow 0$$

のように更新する . また ,

$$TR(\mathbf{w}^k) = TR(\mathbf{w}^{k-1}) - pk_{op}^{k-1}$$

も行われる .

ステップ4  $pk_i^k \geq 0, i \in N$  ときの暫定解は収束解として出力する . そうでないとステップ1に戻して繰り返す .

## 5.3 数値実験

### 5.3.1 機能デバイス $M = 5, 10, 15$ の小規模の数値例

機能デバイス  $M = 5, 10, 15$  の小規模の数値例には , 厳密解は YALMIP において実装された分枝限定法 BAB で得られる . また , 近似探索の効果を近似探索の代表とするもう1つの強いツール , メタヒューリスティックである遺伝的アルゴリズム GA を C 言語で設計して3者の探索結果を比較する . ここで GA における交叉率を 0.5 , 突然変異率を 0.5 とする .

前章の議論したように , インターフェースそのものの収益  $h_i$  と相互作用  $g_{ij}$  の値は , 現実の意味を持っているが , 今の段階で相変わらず精確的に定義することができず , 数値的には解析しかできない . ここで収益  $h_i$  は正規分布  $h_i \sim N(0, 5)$  に従うとする . 一方 , 相互作用  $g_{ij}$  の密度とインターフェース間のインティグラル程度と正相関関係があることによって , 相互作用  $g_{ij}$  の値はここで3パターンの正規分布  $g_{ij} \sim N(-7, 5), g_{ij} \sim N(-5, 5), g_{ij} \sim N(-5, 6)$  に従うとする . これ  $G$  の密度を低 (Low)/中 (Moderate)/高 (High) を分けることができる . 具体的に ,  $G$  の密度が高いのは非0の値が多いことを意味する . このとき , 1つのインターフェースの意思決定を変えると , 多くのインターフェースの一連の変動を起こすことから , インターフェース間の関連性が高いとみられる . 同様に ,  $G$  の密度が低いとき , 1つのインターフェースの意思決定を変えると , ほかのインターフェースが連動的に変わることが少ない , インターフェース間の関連性が低いとみられる . 上の設計により , 低密度  $G \sim N(-7, 5)$  の場合 , ビット値が1であるインターフェースの数は総数  $N$  の 50% を超えないことを保証できる . また , 中密度  $G \sim N(-5, 5)$  の場合 , ビット

値が1であるインターフェースの数は総数  $N$  の 50% ~ 85% の範囲中に制限される．さらに，高密度  $G \sim N(-5, 6)$  の場合，ビット値が1であるインターフェースの数は総数  $N$  の 85% より低いことを意味する．

機能デバイス  $M = 15$  を例として，10回の数値実験の平均により，表 5.1，5.2 と 5.3 に提案する一点変動近傍法の探索効果と能力を検証する．

表 5.1: 高密度  $G \sim N(-5, 6)$  の探索効果

$M = 15(N = 105)$	的中率	最適解との偏差	時間
BAB	-	-	10000s
GA	100%	-	8s
1bitVN	100%	0.0	0.005

表 5.2: 中密度  $G \sim N(-5, 5)$  の探索効果

$M = 15(N = 105)$	的中率	最適解との偏差	時間
BAB	-	-	10000s
GA	100%	-	8s
1bitVN	10%	1.5	0.005

表 5.3: 低密度  $G \sim N(-7, 5)$  の探索効果

$M = 15(N = 105)$	的中率	最適解との偏差	時間
BAB	-	-	10000s
GA	20%	11.1	8s
1bitVN	20%	2.7	0.005

3つの表に示すように，最適解と一致する度合いを表す的中率，最適解と一致しなくてもその値の偏差及び運行時間を記録する．1bitVN で表す提案する一点変動近傍法は高密度  $G \sim N(-5, 6)$  のとき，的中率による探索の正確さも探索時間も両方が良いといえる．一方，中密度  $G \sim N(-5, 5)$  のとき，的中率により，GA は 100%の的中率を達成するのに提案手法 1bitVN で 10%しかない．しかし，最適解との偏差からみると，たとえ最適解を見出せなくても，提案手法 1bitVN では極めて少ない時間内に見出した局所的最良解の目標値と最適解の目標値の偏差は小さい (1.5%) ことが分かる．さらに，低密度  $G \sim N(-7, 5)$  のとき，GA も提案手法 1bitVN も 20%の的中率しか達

成できないが、最適解との偏差からみると、提案手法 1bitVN で見つけた最良解がより最適解に近いとみられる。

### 5.3.2 機能デバイス $M = 25, 40$ の中型規模の数値例

中型規模の問題  $M = 25, 40, N = 300, 780$  には、分枝限定法では扱えなくなることにより、ここで GA のみを用いて提案手法と比較する。図 5.1, 5.2 と 5.3 には  $G$  の各密度のとき、10 回実験の目標関数  $TR$  の平均値とその偏差を記録する。横軸は問題規模  $M = 5, 10, 15, 25, 40$  の値であり、縦軸は目標関数  $TR$  の平均値である。M=15 より大きい規模問題の場合、分枝限定法 BAB の値を記録することが可能ではない。また、プロットされた値の上の方は、BAB, GA と提案手法三者の中に最大の平均値を表し、その下の値は最適解との偏差である。

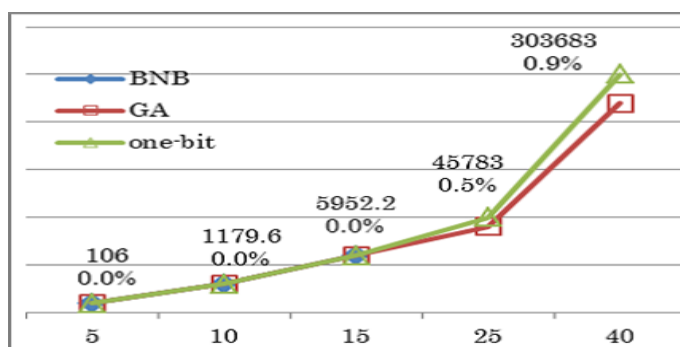


図 5.1: 高密度  $G \sim N(-5, 6)$  の平均値と偏差

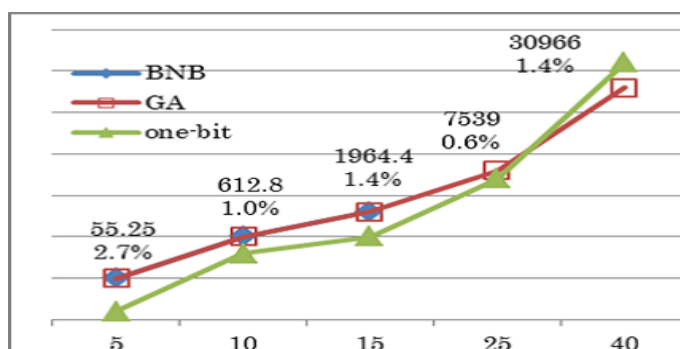


図 5.2: 中密度  $G \sim N(-5, 5)$  の平均値と偏差

図 5.1 の高密度の場合、提案手法の効果はいつも BAB と GA より悪くないとみられる。特に機能デバイスの増大につれて、提案手法の精度と速度両方の良さがどんどん著しくなる。一方、図 5.2 の中密度の場合、小規模問題には提案手法は BAB と GA よりそれほど良くないが、最適解との偏差からみるとその格差は少ない。そして時間的には、提案手法が強い探索能力を示した。GA より 260 倍速いことが分かる。また、機能デバイスの増大に従って、提案手法もどんどん良くなるとみられる。さらに、図 5.3 の低密度の場合、中密度とほぼ同じの効果を証明した。

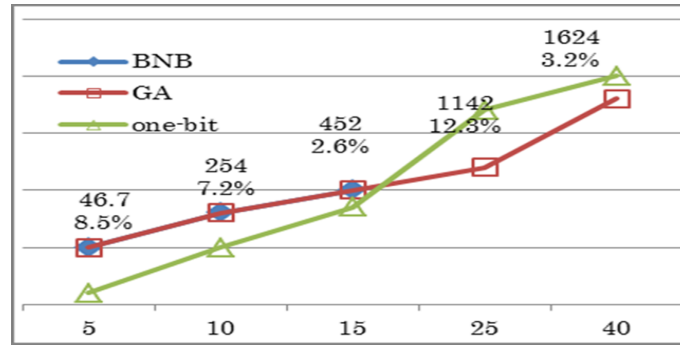


図 5.3: 低密度  $G \sim N(-7, 5)$  の平均値と偏差

### 5.3.3 機能デバイス $M = 70, 100$ の大規模の数値例

機能デバイス  $M = 70, N = 2415$  の大規模問題には，GA での探索は 15 時間がかかる必要がある．一方，提案する一点変動近傍法での探索は 81 秒ほどしかかからないことが分かる．また，さらに大きい規模問題  $M = 100, N = 4950$  には，合理的な時間内に GA での探索も不可能になったが，提案手法での探索は 30 分ぐらい完成できる．数値実験のまとめとして，(1)G の密度が高いほど，提案手法の効果がより良い関係が存在する．これは初期設定のすべてのビットが 1 となる  $w^0 = \{1, 1, \dots, 1\}$  を用いるためであると考えられる．(2)G の中密度と低密度の場合の探索効果は高密度に比べてそれほど良くないが，最適解との偏差がきわめて少ないとみられる．

## 5.4 まとめ

本章では，製品設計の 2 次組合せ最適化モデルに専用する一点変動近傍法を設計する．もともとの変動近傍法によって，暫定解とその一点変動近傍解を比較するのが目標関数値であるが，ここでインターフェースの事前知識で暫定解とその近傍解の優劣を評価する新しい指標となると考える．これで計算量を  $O(2^N)$  から  $O(N^2)$  へ低減できる．

将来の研究について 1 点のみでなく，2 点を互いに反方向に変動する変動近傍法では，より良い効果をもたらすかもしれないと考えるため，今の一点変動近傍法から発展する 2 点変動近傍法の効果を研究する意味がある．2 ビットが同時に反転することは式 (5.11) と式 (5.12) で表すと，

$$\begin{aligned} \Delta TR(\mathbf{w}^k) &= \left( h_s + \sum_{j \neq s}^N (g_{sj} + g_{js}) w_j \right) (w'_s - w_s) \\ &+ \left( h_k + \sum_{j \neq k}^N (g_{kj} + g_{jk}) w_j \right) (w'_k - w_k) - (g_{ks} + g_{sk}) \\ (w'_s - w_s) &= -(w'_k - w_k) \end{aligned} \quad (5.16)$$

$$(w'_s - w_s) = -(w'_k - w_k) \quad (5.17)$$

ようになる．



## 第6章 ボルツマンマシンによるモデルパラメータの推定解析

第3章の厳密解析において、第2章提案した2次組合せ最適化モデルの計算に用いたパラメータ(インターフェースそのものの効用とインターフェース間の相互作用)の値が正規分布によって与えられていた。また、設計現場の経験の浅い設計者が実数のパラメータを定めることが困難である。それに加えて熟練な設計者でさえ数多くのデバイスで構成された製品のパラメータを決定することも難しい。

従って、本研究では製品のデザインとそのカテゴリ評価からパラメータを抽出することを目的として、パラメータの推定モデルを提案する。また、未知変数(パラメータ)の数よりも少ない観測データしか提供されないとき、解の一意性が保証されないことを勘案し、ボルツマンマシンの近似学習を援用してこの問題を解決する [53]。

本章の構成は次の通りである。第1節ではデザインとそのカテゴリ評価に基づくパラメータの推定モデルを提案する。第2節では推定モデルに適用するボルツマンマシンのアルゴリズムを述べる。第3節では推定したパラメータを製品設計モデルに適用するパラメータへの転換を考える。そして第4節では数値実験でパラメータ推定を行い、その効果を考察する。最後に第5節では結論と今後の課題を述べる。

### 6.1 デザインとカテゴリ評価による推定モデルの提案

ここでは  $M$  個のデバイスで構成された製品を考える。この  $M$  個のデバイス間のそれぞれのインターフェースをつなげるかどうかの設計指針全体を本稿では製品設計と呼ぶことにし、デバイス間のインターフェースの総数を  $N = M(M - 1)/2$  で表すことにする。さて  $R$  個のデータセット  $D = \{d^1, d^2, \dots, d^R\}$  を以下のように用意する。

$$d^\mu = (\mathbf{S}^\mu, K^\mu), \mu = 1, \dots, R \quad (6.1)$$

$$\mathbf{S}^\mu = (S_1^\mu, \dots, S_N^\mu)^T \in \{\pm 1\}^N \quad (6.2)$$

ただし  $\mathbf{S}^\mu = (S_1^\mu, \dots, S_N^\mu)^T \in \{\pm 1\}^N$  は  $\mu$  番目の製品設計の  $N$  個のインターフェースの状態を表し、 $K^\mu$  は製品設計  $\mathbf{S}^\mu$  のカテゴリ評価値であり、 $d^\mu = (\mathbf{S}^\mu, K^\mu)$  は  $\mu$  番目の製品設計とその評価値の組を表す。また  $S_i^\mu = 1$  はインターフェース  $i$  をつなげることを表し、 $S_i^\mu = -1$  はつなげない

ことを表す．さらに製品設計の評価は次のコスト関数を用いて表されていると仮定する．

$$\mathcal{H}(\mathbf{S}) = - \sum_{i=1}^N d_i S_i - \frac{1}{2} \sum_{i=1}^N \sum_{j(\neq i)}^N S_i J_{ij} S_j \quad (6.3)$$

しかしながら，実際の設計現場では製品を特徴づける  $\mathbf{d}, J$  が既知であるとは限らず，さらに，製品設計の評価値  $\mathcal{H}(\mathbf{S})$  においても，実際の評価値が与えられておらず，むしろ従来研究 (QFD と DSM) により，設計者が提供する評価は言語変数 (優, 可, 劣など) あるいはカテゴリ評価 (何段階スコアリング) である場合が少なくない．そのため，本研究では設計  $\mathbf{S}^\mu$  のコスト関数  $\mathcal{H}(\mathbf{S}^\mu)$  とそのカテゴリ評価値  $k^\mu$  の関係式  $k^\mu = f\left(\frac{1}{N}\mathcal{H}(\mathbf{S}^\mu)\right)$  を構築し，カテゴリ  $k^\mu$  と観測カテゴリ評価値  $K^\mu$  との誤差を最小にするパラメータ  $\mathbf{d}^*, J^*$  が最適パラメータとして推定する．

ここで関係式  $f(\rho)$  は多段階関数の形であると仮定する．例えば，2段階関数の場合， $\rho$  がある閾値  $Q$  より大きくなった場合 0 を出力し，小さくなった場合 1 を出力する．一方，解析的に問題を解くためには微分可能な階段関数である必要がある．そのため，ここで製品設計の評価値  $\mathcal{H}(\mathbf{S}^\mu)$  のカテゴリ  $k^\mu$  は  $p$  個のシグモイド関数  $f_a(\rho)$  の総和からなるカテゴリ関数  $f(\rho)$  で定義されると仮定する [57]．

$$k^\mu = f\left(\frac{1}{N}\mathcal{H}(\mathbf{S}^\mu)\right) \quad (6.4)$$

$$f(\rho) = \sum_{a=1}^p f_a(\rho) \quad (6.5)$$

$$f_a(\rho) = \frac{1}{1 + e^{\gamma_a(\rho - \theta_a)}}, \gamma_a > 0 \quad (6.6)$$

ただし  $f_a(s)$  は微分可能なシグモイド関数である [57]． $p$  はカテゴリ数を表し， $\gamma_a \rightarrow \infty$  で  $f(s)$  は多段階関数になることが分かる．例えば  $p = 1$  の2段階カテゴリ評価  $k^\mu = \lim_{\gamma_1 \rightarrow \infty} f_1\left(\frac{1}{N}\mathcal{H}(\mathbf{S}^\mu)\right) \in \{0, 1\}$  である．そのうち， $k^\mu = 1$  はデザイン  $\mathbf{S}^\mu$  が良い， $k^\mu = 0$  は逆の意味を表す．一方， $p = 2$  としたら  $k^\mu \in \{0, 1, 2\}$  の3段階カテゴリ評価となる．また  $\frac{\partial f_a(s)}{\partial s} = -\gamma_a f_a(s)(1 - f_a(s)) \leq 0$  のため，コスト関数  $\mathcal{H}(\mathbf{S}^\mu)$  の値の小さいデザイン  $\mathbf{S}^\mu$  が高いカテゴリ評価  $k^\mu$  を持つ関係を保証する．図 6.1 に  $p = 2$ ,  $f(s) = \frac{1}{1 + e^{\gamma_1(s - \theta_1)}} + \frac{1}{1 + e^{\gamma_2(s - \theta_2)}}$ ,  $\theta_1 = 1, \theta_2 = 2$  を固定する場合， $\gamma_1 = \gamma_2 = 10, 20, \infty$  それぞれの3段階カテゴリ関数を示す．図からみると， $\gamma$  の増大に従って，カテゴリ関数  $f(\rho)$  が多段階関数に近くなることを確かめた．さらに  $f_a(\rho)$  は単調非増加関数であるため，コスト関数  $\mathcal{H}(\mathbf{S}^\mu)$  の値の小さい設計  $\mathbf{S}^\mu$  が高いカテゴリ評価値  $k^\mu$  を持つことを意味する．

上の議論を踏まえて， $R$  セットの観測データにおけるカテゴリ  $K^\mu$  とシグモイド関数による  $k^\mu$  の誤差を次のような目標関数  $C$  で表すことにする．

$$C = \frac{1}{2} \sum_{\mu=1}^R (k^\mu - K^\mu)^2 \quad (6.7)$$

推定の目標は，

$$\arg \min_{\mathbf{d}, J, \gamma, \theta} C = (\mathbf{d}^*, J^*, \gamma^*, \theta^*) \quad (6.8)$$



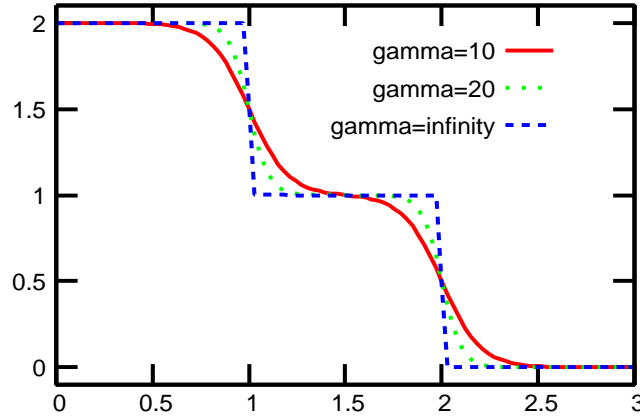


図 6.1:  $p = 2, \gamma_1 = \gamma_2 = 10, 20, \infty$  の場合それぞれの 3 段階カテゴリ関数

である．具体的に，パラメータ  $d_i, J_{ij}$  及びカテゴリ関数におけるパラメータ  $\gamma_a, \theta_a$  を調整することによって， $C$  を最小にすることを達成でき， $C$  を最小にしたときの  $d, J$  を最適パラメータ  $d^*, J^*$  として出力する．パラメータ  $d_i, J_{ij}, \gamma_a, \theta_a$  を最適になるように調整することは，

$$\frac{\partial C}{\partial d_i} = -\frac{1}{N} \sum_{\mu=1}^R (f(\rho^\mu) - K^\mu) S_i^\mu \frac{\partial f(\rho^\mu)}{\partial \rho^\mu} \quad (6.9)$$

$$\begin{aligned} \frac{\partial C}{\partial J_{ij}} &= -\frac{1}{N} \sum_{\mu=1}^R (f(\rho^\mu) - K^\mu) \\ &\quad \times S_i^\mu S_j^\mu \frac{\partial f(\rho^\mu)}{\partial \rho^\mu} \end{aligned} \quad (6.10)$$

$$\begin{aligned} \frac{\partial C}{\partial \gamma_a} &= \sum_{\mu=1}^R (f(\rho^\mu) - K^\mu) (\rho^\mu - \theta_a) \\ &\quad \times f_a(\rho^\mu) (1 - f_a(\rho^\mu)) \end{aligned} \quad (6.11)$$

$$\begin{aligned} \frac{\partial C}{\partial \theta_a} &= \sum_{\mu=1}^R (f(\rho^\mu) - K^\mu) f_a(\rho^\mu) \\ &\quad \times (1 - f_a(\rho^\mu)) \end{aligned} \quad (6.12)$$

の更新則を用いて求められる．ただし， $\rho^\mu = \frac{1}{N} \mathcal{H}(\mathbf{S}^\mu)$ ， $\frac{\partial f(\rho^\mu)}{\partial \rho^\mu} = \sum_{a=1}^p \gamma_a f_a(\rho^\mu) (f_a(\rho^\mu) - 1)$  である．

しかし，推定モデルは連立方程式でパラメータを推定するため，すべての未知変数（パラメータ）を特定できない可能性がある．すなわち， $C = 0$  のとき得られた解（1つのみではない）の学習誤差が最小になったが，汎化誤差が最小になったとは限らない．そのため，確率的に振る舞うボルツマンマシンを援用して，観測のセット数  $R$  が不足であるとき，学習誤差が 0 のもとで汎化誤差を最小にする推定手法を 2 節にて提案する．

## 6.2 推定モデルのボルツマンマシン学習

### 6.2.1 ボルツマンマシンの定義

ボルツマンマシン (Boltzmann machine) は、統計物理学において研究されているスピンステムに基づいて発展されたモデルである [54]。ボルツマンマシンでは観測データを生成する分布がボルツマン分布であると仮定し、観測データに応じてパラメータ推定問題を解くことを考える [53][57]。ここでボルツマンマシンを用いた学習は、 $N$ 次元分布  $P_0(\mathbf{S})$  に従うデータ  $\mathbf{S}$  が  $R$  セット  $\{S_i^\mu | \mu = 1, 2, \dots, R, i = 1, 2, \dots, N\}$  を用意されたとき、 $\mathbf{S}$  のボルツマン分布

$$P(\mathbf{S}|\mathbf{h}, G) = \frac{e^{-\beta\mathcal{H}(\mathbf{S})}}{Z(\beta)} \quad (6.13)$$

が  $R$  セット  $\{S_i^\mu | \mu = 1, 2, \dots, R, i = 1, 2, \dots, N\}$  から求まる経験分布

$$P_0(\mathbf{S}) = \frac{1}{R} \sum_{\mu=1}^R \prod_{i \in N} \delta(S_i, S_i^\mu) \quad (6.14)$$

に近くなるようにパラメータ  $\mathbf{d}$  と  $J$  の値を修正することで達成される。また、

$$Z(\beta) = \sum_{\mathbf{S}} e^{-\beta\mathcal{H}(\mathbf{S})} \quad \beta > 0 \quad (6.15)$$

は分配関数と呼ばれる規格化定数である。そして、 $\delta(n, m)$  はクロネッカーのデルタ (Kronecker delta function) であり、

$$\delta(n, m) = \begin{cases} 1 & n = m \\ 0 & n \neq m \end{cases} \quad (6.16)$$

であることが知られている [53]。

### 6.2.2 ボルツマンマシンの最適化問題

学習誤差が 0 のもとで汎化誤差を最小にする問題を扱えるために、経験分布  $P_0(\mathbf{S})$  とボルツマン分布  $P(\mathbf{S}|\mathbf{d}, J)$  の近さを表すカルバックライブラー情報量

$$KL = \sum_{\mathbf{S}} P_0(\mathbf{S}) \log \left( \frac{P_0(\mathbf{S})}{P(\mathbf{S}|\mathbf{d}, J)} \right) \quad (6.17)$$

の最小化問題を考える。

$$\text{Min} \quad KL \quad (6.18)$$

$$\text{s.t.} \quad K^\mu = f \left( \frac{1}{N} \mathcal{H}(\mathbf{S}^\mu) \right), \mu = 1, \dots, R \quad (6.19)$$

推定の目標は，

$$\arg \min_{\mathbf{d}, J, \gamma, \theta} KL = (\mathbf{d}^*, J^*) \quad (6.20)$$

である．この最適化問題をラグランジュ関数  $L$  でまとめれば，

$$\begin{aligned} L = & \sum_{\mathbf{S}} P_0(\mathbf{S}) \log \left( \frac{P_0(\mathbf{S})}{P(\mathbf{S}|\mathbf{d}, J)} \right) \\ & + \frac{1}{2} \sum_{\mu=1}^R \varepsilon^\mu \left( f \left( \frac{1}{N} \mathcal{H}(\mathbf{S}^\mu) \right) - K^\mu \right)^2 \end{aligned} \quad (6.21)$$

となる．また， $P(\mathbf{S}|\mathbf{d}, J)Z(\beta) = e^{-\beta \mathcal{H}(\mathbf{S})}$  の両辺の対数を取り，ラグランジュ関数  $L$  は，

$$\begin{aligned} L = & \sum_{\mathbf{S}} P_0(\mathbf{S}) \log P_0(\mathbf{S}) \\ & - \beta \sum_{\mathbf{S}} P_0(\mathbf{S}) \left( \mathbf{d}^T \mathbf{S} + \frac{1}{2} \mathbf{S}^T J \mathbf{S} \right) + \log Z(\beta) \\ & + \frac{1}{2} \sum_{\mu=1}^R \varepsilon^\mu \left( f \left( \frac{1}{N} \mathcal{H}(\mathbf{S}^\mu) \right) - K^\mu \right)^2 \end{aligned} \quad (6.22)$$

と展開できる．これより  $L$  の最小値は，

$$d_i^{t+1} = d_i^t + \eta_d \left( \beta (\langle S_i \rangle_0 - \langle S_i \rangle) - \frac{\partial C}{\partial d_i} \right) \quad (6.23)$$

$$J_{ij}^{t+1} = J_{ij}^t + \eta_J \left( \beta (\langle S_i S_j \rangle_0 - \langle S_i S_j \rangle) - \frac{\partial C}{\partial J_{ij}} \right) \quad (6.24)$$

$$\gamma_a^{t+1} = \gamma_a^t - \eta_\gamma \left( \frac{\partial L}{\partial \gamma_a} \right) \quad (6.25)$$

$$\theta_a^{t+1} = \theta_a^t - \eta_\theta \left( \frac{\partial L}{\partial \theta_a} \right) \quad (6.26)$$

$$\varepsilon^{\mu, t+1} = \varepsilon^{\mu, t} + \eta_\varepsilon \left( \frac{\partial L}{\partial \varepsilon^\mu} \right) \quad (6.27)$$

$$\beta^{t+1} = \beta^t - \eta_\beta \left( \frac{\partial L}{\partial \beta} \right) \quad (6.28)$$

の更新則で得られる．ただし， $\gamma_a$   $\theta_a$   $\varepsilon^\mu$  が  $KL$  と関わらないため，

$$\frac{\partial L}{\partial \gamma_a} = \frac{\partial C}{\partial \gamma_a} \quad (6.29)$$

$$\frac{\partial L}{\partial \theta_a} = \frac{\partial C}{\partial \theta_a} \quad (6.30)$$

$$\frac{\partial L}{\partial \varepsilon^\mu} = \frac{\partial C}{\partial \varepsilon^\mu} \quad (6.31)$$

である．また， $KL$  におけるパラメータ  $\beta$  の学習は，

$$\begin{aligned} \frac{\partial L}{\partial \beta} = & - \sum_{i=1}^N d_i^t (\langle S_i \rangle_0 - \langle S_i \rangle_t) \\ & - \frac{1}{2} \sum_{i \neq j} J_{ij}^t (\langle S_i S_j \rangle_0 - \langle S_i S_j \rangle_t) \end{aligned} \quad (6.32)$$

のようになる．ただし，

$$\langle S_i \rangle_0 = \sum_{\mathbf{S}} S_i P_0(\mathbf{S}) \quad (6.33)$$

$$\langle S_i S_j \rangle_0 = \sum_{\mathbf{S}} S_i S_j P_0(\mathbf{S}) \quad (6.34)$$

が経験分布  $P_0(\mathbf{S})$  の期待値であり，

$$\langle S_i \rangle = \sum_{\mathbf{S}} S_i P(\mathbf{S}|\mathbf{d}, J) \quad (6.35)$$

$$\langle S_i S_j \rangle = \sum_{\mathbf{S}} S_i S_j P(\mathbf{S}|\mathbf{d}, J) \quad (6.36)$$

が試行分布の平均値である．そして，小さな正の学習率（ステップ幅） $\eta_d, \eta_J, \eta_\gamma, \eta_\theta, \eta_\varepsilon, \eta_\beta$  は小さすぎれば推定速度に悪影響をもたらし，大きすぎれば求めたい最適解から脱出する可能性が高いことが分かる．そのため，ここでは  $\eta_d, \eta_J, \eta_\gamma, \eta_\theta, \eta_\varepsilon, \eta_\beta = 10^{-6}$  を用いる．

### 6.2.3 ボルツマンマシンの平衡状態の実現

しかしながら， $\sum_{\mathbf{S}} = \sum_{S_1=\pm 1} \sum_{S_2=\pm 1} \cdots \sum_{S_N=\pm 1}$  は全確率変数の全配位に対する和を意味するから，ボルツマン分布  $P(\mathbf{S}|\mathbf{d}, J) = \frac{e^{-\beta \mathcal{H}(\mathbf{S})}}{Z(\beta)}$  における分配関数  $Z(\beta) = \sum_{\mathbf{S}} e^{-\beta \mathcal{H}(\mathbf{S})}$  の計算量はすべての  $J_{ij}$  が0でない限り，インターフェース数  $N$  の増加によって  $O(2^N)$  のオーダーで増大してしまう NP 困難な問題となる．全数探索で  $P(\mathbf{S})$  の計算が必要である試行分布  $\langle S_i \rangle$  や  $\langle S_i S_j \rangle$  の計算は厳密に実行することは現実的ではないことが分かった．

一方，ボルツマンマシンの平衡状態を実現するために，以下のような確率アルゴリズムを考える．

1. ランダムに  $\mathbf{S}$  の初期値  $\mathbf{S}^0$  を定める．
2. 今の状態であるデザイン  $\mathbf{S}^a$  とその候補，すべての  $N$  個の変数の中から一様確率に従って変数  $n$  を選択し，残りの変数は変わらなくて，選んだ変数の値を反転するデザイン  $\mathbf{S}^b$  は確率，

$$P_n = \frac{P(\mathbf{S}^b)}{P(\mathbf{S}^a)} = e^{-\beta(\mathcal{H}(\mathbf{S}^b) - \mathcal{H}(\mathbf{S}^a))} \quad (6.37)$$

に従い， $\mathcal{H}(\mathbf{S})$  は小さいほど  $P(\mathbf{S})$  が高いことによって， $P_n > 1$  であれば  $\mathcal{H}(\mathbf{S}^b) < \mathcal{H}(\mathbf{S}^a)$  のため，候補  $\mathbf{S}^b$  を次の状態として採用する．そうでなければ，ランダムに候補  $\mathbf{S}^b$  を次の状態として採用する．

3. 2 の手続きを繰り返す．

ただし，パラメータ  $\beta$  の働きはシミュレーテッド・アニーリング法における温度  $T$  の効果と一致する．ただ， $\beta = \frac{1}{T}$  の関係により，ボルツマンマシンの平衡状態は，温度  $T \rightarrow 0$ ，すなわち  $\beta \rightarrow \infty$  のとき，上記の2の操作が変化しない状態でなければならないことから定まる．この証明過程の詳細は4.2.2に参考してよい．要するに，ボルツマンマシンの平衡状態の実現は，モデルのサイズに対して  $P(\mathbf{S})$  の計算が効率的な統計的近似計算手法を用いる必要がある．

### 6.2.4 確率伝搬法

$\langle S_i \rangle$  や  $\langle S_i S_j \rangle$  の評価の計算量はインターフェースの数  $N$  の増大に従って指数的に膨大する計算困難問題を解決するために、いくつかの効率的な近似計算手法が提案された [53] . 例えば、大規模な確率モデルを取り扱う上で統計力学における平均場近似 (Mean Field Approximation, MFA) がそのうちの 1 つである . 平均場近似とは、相互作用しているインターフェース同士は要素とみなし、製品設計はすべてのインターフェース要素からなる多体システムとみなすことによって、この多体システムに対する相互作用をある平均的なもの (平均場) による近似である . 具体的に、平均場近似法により、試行分布の期待値  $\langle S_i \rangle$  は、製品設計  $S$  におけるインターフェース  $i$  の単独コスト  $d_i$  と近傍のすべてのインターフェース  $j$  から  $i$  に向かう平均影響  $\sum_{j=1}^N J_{ij} \langle S_j \rangle$ 、この両方からなる平均場方程式  $\langle S_i \rangle = \tanh \beta \left( d_i + \sum_{j=1}^N J_{ij} \langle S_j \rangle \right)$  で計算されるという構造になっているものである [53] .

しかしながら、平均場近似法は試行分布の期待値  $\langle S_i \rangle$  の計算の簡便さを重視するが、近傍のインターフェース同士の相関を考慮していない欠点が存在する . そのためここでは、近傍のインターフェース同士の相関を周辺分布として求められる確率伝搬法 (Belief Propagation, BP) を用いて、平均場近似法より高い近似精度の近似を期待できると考える .

確率伝搬法の周辺分布の求め方は、製品設計  $S$  におけるインターフェース  $i$  からインターフェース  $j$  への平均影響は  $j$  以外から  $i$  に与える平均影響から計算される [53] . ここでは確率伝搬法を用いてアルゴリズムを構成する . まず  $-\beta \mathcal{H}(S)$  を展開する .

$$\begin{aligned}
 -\beta \mathcal{H}(S) &= \frac{\beta}{2} \sum_{i,j} J_{ij} S_i S_j + \beta \sum_{i=1}^N d_i S_i \\
 &= \frac{\beta}{2} \sum_{i \neq j} (J_{ij} S_i S_j + d_i S_i + d_j S_j) - \beta(N-1) \sum_{i=1}^N d_i S_i + \beta \sum_{i=1}^N d_i S_i \\
 &= \beta \sum_{i>j} (J_{ij} S_i S_j + d_i S_i + d_j S_j) + \beta(2-N) \sum_{i=1}^N d_i S_i \tag{6.38}
 \end{aligned}$$

となる . これより真の分布  $P(S) = \frac{e^{-\beta \mathcal{H}(S)}}{Z(\beta)}$  のボルツマン分布に近い試験分布

$$Q(S) = \frac{1}{Z_Q} \prod_{i>j} \frac{b_{ij}(S_i, S_j)}{b_i(S_i) b_j(S_j)} \prod_{i=1}^N b_i(S_i) = \frac{1}{Z_Q} \prod_{i>j} b_{ij}(S_i, S_j) \prod_{i=1}^N b_i^{1-(N-1)}(S_i) \tag{6.39}$$

を求める問題を考える . ただし分母の  $Z_Q$  は

$$Z_Q = \sum_{\mathbf{S}} \prod_{i>j} b_{ij}(S_i, S_j) \prod_{i=1}^N b_i^{1-(N-1)}(S_i) \tag{6.40}$$

である規格化定数である . さらに  $b_i(S_i)$  と  $b_{ij}(S_i, S_j)$  の間には

$$b_i(S_i) = \sum_{S_j=\pm 1} b_{ij}(S_i, S_j) \tag{6.41}$$

の関係が成り立っていると仮定する．ここで  $P(S)$  と  $Q(S)$  の部分ごとに，

$$b_{ij}(S_i, S_j) \propto e^{\beta J_{ij} S_i S_j + \beta d_i S_i + \beta d_j S_j} \quad (6.42)$$

$$b_i(S_i) \propto e^{\beta d_i S_i} \quad (6.43)$$

が対応しているとして，Bethe 自由エネルギー  $F$  を

$$\begin{aligned} F &= \sum_{i>j} \sum_{S_i=\pm 1, S_j=\pm 1} b_{ij}(S_i, S_j) \log \frac{b_{ij}(S_i, S_j)}{e^{\beta J_{ij} S_i S_j + \beta d_i S_i + \beta d_j S_j}} + (2 - N) \sum_{i=1}^N \sum_{S_i=\pm 1} b_i(S_i) \log \frac{b_i(S_i)}{e^{\beta d_i S_i}} \\ &+ \sum_{i=1}^N \sum_{j=1}^N \sum_{S_i=\pm 1} \lambda_{ji}(S_i) \left[ \sum_{S_j=\pm 1} b_{ij}(S_i, S_j) - b_i(S_i) \right] \end{aligned} \quad (6.44)$$

で定義し， $F$  を最小にする  $b_i(S_i), b_{ij}(S_i, S_j)$  を求める．まず  $F$  を  $b_i(S_i)$  と  $b_{ij}(S_i, S_j)$  で微分して，

$$\frac{\partial F}{\partial b_{ij}(S_i, S_j)} = \log \frac{b_{ij}(S_i, S_j)}{e^{\beta J_{ij} S_i S_j + \beta d_i S_i + \beta d_j S_j}} + 1 + \lambda_{ji}(S_i) + \lambda_{ij}(S_j) = 0 \quad (6.45)$$

$$\frac{\partial F}{\partial b_i(S_i)} = (2 - N) \left( \log \frac{b_i(S_i)}{e^{\beta d_i S_i}} + 1 \right) - \sum_{j \in \mathcal{L}(i)} \lambda_{ji}(S_i) = 0 \quad (6.46)$$

となる．ただし記法  $\sum_{j \in \mathcal{L}(i)}$  はインターフェース  $i$  と連結した ( $J_{ij} \neq 0$ ) インターフェース  $j$  全体で和を取ることを表す．これより，

$$b_{ij}(S_i, S_j) \propto \exp [\beta J_{ij} S_i S_j + \beta d_i S_i + \beta d_j S_j - \lambda_{ji}(S_i) + \lambda_{ij}(S_j)] \quad (6.47)$$

$$b_i(S_i) \propto \exp \left[ \beta d_i S_i + \frac{1}{2 - N} \sum_{j \in \mathcal{L}(i)} \lambda_{ji}(S_i) \right] \quad (6.48)$$

が得られる．ただし  $S_i, S_j$  に関連しない定数は無視した．次に新たな関数として，

$$\tilde{\lambda}_{ji}(S_i) = \lambda_{ji}(S_i) + \frac{1}{2 - N} \sum_{j \in \mathcal{L}(i)} \lambda_{ji}(S_i) \quad (6.49)$$

を定義する．またインターフェース  $i$  と連結したインターフェース全体で和を取ると，

$$\sum_{j \in \mathcal{L}(i)} \tilde{\lambda}_{ji}(S_i) = \sum_{j \in \mathcal{L}(i)} \lambda_{ji}(S_i) + \frac{N - 1}{2 - N} \sum_{j \in \mathcal{L}(i)} \lambda_{ji}(S_i) = \frac{1}{2 - N} \sum_{j \in \mathcal{L}(i)} \lambda_{ji}(S_i) \quad (6.50)$$

となる．これより， $\tilde{\lambda}_{ji}(S_i) = \lambda_{ji}(S_i) + \sum_{j \in \mathcal{L}(i)} \tilde{\lambda}_{ji}(S_i)$  を変形して，

$$\lambda_{ji}(S_i) = \tilde{\lambda}_{ji}(S_i) - \sum_{j \in \mathcal{L}(i)} \tilde{\lambda}_{ji}(S_i) \quad (6.51)$$

となる．さらに  $\tilde{\lambda}_{ji}(S_i)$  は  $S_i = \pm 1$  の関数なので，

$$\tilde{\lambda}_{ji}(S_i) = \tilde{d}_{ji} S_i \quad (6.52)$$

とおくことができる．ここでは定数項を無視した．またインターフェース  $i$  と連結したインターフェース全体で和を取って，

$$y_i = \sum_{j \in \mathcal{L}(i)} \tilde{d}_{ji} \quad (6.53)$$

とおく．これより，

$$\lambda_{ji}(S_i) = \tilde{d}_{ji}S_i - y_iS_i \quad (6.54)$$

となる．また  $b_{ij}(S_i, S_j), b_i(S_i)$  は，

$$b_{ij}(S_i, S_j) \propto e^{\beta J_{ij}S_iS_j + \beta d_iS_i + \beta d_jS_j + y_iS_i - \tilde{d}_{ji}S_i + y_jS_j - \tilde{d}_{ij}S_j} \quad (6.55)$$

$$b_i(S_i) \propto e^{\beta d_iS_i + y_iS_i} \quad (6.56)$$

となる．よって，

$$m_i = \tanh(\beta d_i + y_i) \quad (6.57)$$

となる．次に  $i, j$  の分配関数  $Z_{ij}$  は，

$$\begin{aligned} \log Z_{ij} &= \log \left( \sum_{S_i = \pm 1, S_j = \pm 1} e^{\beta J_{ij}S_iS_j + \beta d_iS_i + \beta d_jS_j + y_iS_i - \tilde{d}_{ji}S_i + y_jS_j - \tilde{d}_{ij}S_j} \right) \\ &= \log(e^{\beta J_{ij} + \beta d_i + \beta d_j + y_i - \tilde{d}_{ji} + y_j - \tilde{d}_{ij}} + e^{-\beta J_{ij} - \beta d_i + \beta d_j - y_i + \tilde{d}_{ji} + y_j - \tilde{d}_{ij}} \\ &\quad + e^{-\beta J_{ij} + \beta d_i - \beta d_j + y_i - \tilde{d}_{ji} - y_j - \tilde{d}_{ij}} + e^{\beta J_{ij} - \beta d_i - \beta d_j - y_i + \tilde{d}_{ji} - y_j + \tilde{d}_{ij}}) \end{aligned} \quad (6.58)$$

となる．これより，

$$\begin{aligned} m_i &= \frac{\partial}{\partial \beta d_i} \log Z_{ij} \\ &= \frac{e^{\beta d_i + y_i - \tilde{d}_{ji}} 2 \cosh(\beta d_j + y_j - \tilde{d}_{ij} + \beta J_{ij}) - e^{-(\beta d_i + y_i - \tilde{d}_{ji})} 2 \cosh(\beta d_j + y_j - \tilde{d}_{ij} - \beta J_{ij})}{e^{\beta d_i + y_i - \tilde{d}_{ji}} 2 \cosh(\beta d_j + y_j - \tilde{d}_{ij} + \beta J_{ij}) + e^{-(\beta d_i + y_i - \tilde{d}_{ji})} 2 \cosh(\beta d_j + y_j - \tilde{d}_{ij} - \beta J_{ij})} \\ &= \tanh \left[ \beta d_i + y_i - \tilde{d}_{ji} + \frac{1}{2} \log \frac{\cosh(\beta d_j + y_j - \tilde{d}_{ij} + \beta J_{ij})}{\cosh(\beta d_j + y_j - \tilde{d}_{ij} - \beta J_{ij})} \right] \end{aligned} \quad (6.59)$$

となる．ただし  $\tanh x = \frac{e^{2x} - 1}{e^{2x} + 1}$  を用いた．これより式 (6.40) と式 (6.42) の両方を比較して，

$$\tilde{d}_{ji} = \frac{1}{2} \log \frac{\cosh(\beta d_j + y_j - \tilde{d}_{ij} + \beta J_{ij})}{\cosh(\beta d_j + y_j - \tilde{d}_{ij} - \beta J_{ij})} \quad (6.60)$$

$$y_i = \sum_{j \in \mathcal{L}(i)} \tilde{d}_{ji} \quad (6.61)$$

となる．次に  $m_{ij}$  は，

$$\begin{aligned}
m_{ij} &= \frac{\partial}{\partial \beta J_{ij}} \log Z_{ij} \\
&= \frac{e^{\beta J_{ij}} 2 \cosh(\beta d_i + y_i - \tilde{d}_{ji} + \beta d_j + y_j - \tilde{d}_{ij}) - e^{-\beta J_{ij}} 2 \cosh(\beta d_i + y_i - \tilde{d}_{ji} - \beta d_j - y_j + \tilde{d}_{ij})}{e^{\beta J_{ij}} 2 \cosh(\beta d_i + y_i - \tilde{d}_{ji} + \beta d_j + y_j - \tilde{d}_{ij}) + e^{-\beta J_{ij}} 2 \cosh(\beta d_i + y_i - \tilde{d}_{ji} - \beta d_j - y_j + \tilde{d}_{ij})} \\
&= \tanh \left[ \beta J_{ij} + \frac{1}{2} \log \frac{\cosh(\beta d_i + y_i - \tilde{d}_{ji} + \beta d_j + y_j - \tilde{d}_{ij})}{\cosh(\beta d_i + y_i - \tilde{d}_{ji} - (\beta d_j + y_j - \tilde{d}_{ij}))} \right] \tag{6.62}
\end{aligned}$$

となる．

上の議論により，ここで  $S_i$  の試行分布に関する平均  $m_i = \langle S_i \rangle$ ,  $m_{ij} = \langle S_i S_j \rangle$  で書き換えると， $d_i$  と  $J_{ij}$  に対する  $L$  の学習則は，

$$\frac{\partial L}{\partial d_i} = -\beta \langle S_i \rangle_0 + \beta m_i + \frac{\partial C}{\partial d_i} \tag{6.63}$$

$$\frac{\partial L}{\partial J_{ij}} = -\beta \langle S_i S_j \rangle_0 + \beta m_{ij} + \frac{\partial C}{\partial J_{ij}} \tag{6.64}$$

となる．また確率伝搬法における  $m_i, m_{ij}$  は，

$$m_i = \tanh(\beta d_i + y_i), \tag{6.65}$$

$$m_{ij} = \tanh \left[ \beta J_{ij} + \frac{1}{2} \log \frac{\cosh(\beta d_i + y_i - \tilde{d}_{ji} + \beta d_j + y_j - \tilde{d}_{ij})}{\cosh(\beta d_i + y_i - \tilde{d}_{ji} - (\beta d_j + y_j - \tilde{d}_{ij}))} \right] \tag{6.66}$$

である．我々は  $d_i, J_{ij}$  が与えられたもとで，式 (6.43) と式 (6.44) を満たすように  $\tilde{d}_{ij}, y_i$  を決定し，それを用いて式 (6.48) と式 (6.49) の  $m_i, m_{ij}$  を計算し， $\frac{\partial L}{\partial d_i}, \frac{\partial L}{\partial J_{ij}}$  を評価することができる [53]．これより  $d_i, J_{ij}$  の更新則は，

$$d_i^{t+1} = d_i^t - \varepsilon_d \left( \frac{\partial L}{\partial d_i} \right) \tag{6.67}$$

$$J_{ij}^{t+1} = J_{ij}^t - \varepsilon_J \left( \frac{\partial L}{\partial J_{ij}} \right) \tag{6.68}$$

となる．

### 6.2.5 サンプルデータ $S$ の生成

提案手法の有効性を数値実験で検証するとき，サンプルデータ  $S$  の生成はここでメトロポリス・ヘイスティングス (M-H) 法で扱うとする [58]．M-H 法の目的は目標確率分布に従ったサンプル列の生成を行うことである．ここで用意した  $d_i, J_{ij}$  (真の値とも呼ぶ) の目標確率分布は  $P(S|d, J) = \frac{e^{-\beta \mathcal{H}(S|d, J)}}{\sum_S e^{-\beta \mathcal{H}(S|d, J)}}$ ,  $\beta > 0$  である．今の状態であるデザイン  $S^a$  とその候補，一つのビットの値を反転するデザイン  $S^b$  の確率分布の比  $r_P$  は，

$$r_P = \frac{P(S^b)}{P(S^a)} = e^{-\beta(\mathcal{H}(S^b) - \mathcal{H}(S^a))} \tag{6.69}$$



となる． $\mathcal{H}(S)$  は小さいほど  $P(S)$  が高いことによって， $r_P > 1$  であれば  $\mathcal{H}(S^b) < \mathcal{H}(S^a)$  のため，候補  $S^b$  を次の状態として採用する．そうでなければ，ランダムに候補  $S^b$  を次の状態に採用する．サンプルの数が充分であれば，その分布は  $P(S|d, J)$  に近似することが期待される．ここで M-H 法により生成した  $10^7$  個のサンプル列に対して，前の 20% を抜いて残りの 80% の個体の中にランダムに  $N$  個のデザインを実験のサンプルとして選ぶ．M-H 法で生成したサンプル列はランダムに生成する方より良い推定精度をもたらすことができると考える．

上の議論により，以下はボルツマンマシンのアルゴリズムをまとめる．

ステップ 0 (初期入力及び  $\langle S_i \rangle_0$  と  $\langle S_i S_j \rangle_0$  の計算) :

サンプル  $S^\mu$  とそのカテゴリ評価  $K^\mu$ ，及び  $d^0, J^0$  を入力し， $\gamma_a^0, \theta_a^0, \varepsilon^{\mu,0}, \beta^0$  を設定する．期待値  $\langle S_i \rangle_0 = \frac{1}{N} \sum_{\mu=1}^N S_i^\mu$  と  $\langle S_i S_j \rangle_0 = \frac{1}{N} \sum_{\mu=1}^N S_i^\mu S_j^\mu$  で得られる．

ステップ 1 (試行分布の平均値  $m_i^t, m_{ij}^t$  の明確) :

$d_i^t, J_{ij}^t$  が与えられたもとの式 (6.66) と式 (6.67) を満たすように  $\tilde{d}_{ij}^t, y_i^t$  を決定し，それを用いて式 (6.71) と式 (6.72) によって  $m_i^t, m_{ij}^t$  を計算する．

ステップ 2 (各パラメータの学習):

- 式 (6.73) によって  $d_i^t \rightarrow d_i^{t+1}$  を更新する．
- 式 (6.74) によって  $J_{ij}^t \rightarrow J_{ij}^{t+1}$  を更新する．
- 式 (6.25) によって  $\gamma_a^t \rightarrow \gamma_a^{t+1}$  を更新する．
- 式 (6.26) によって  $\theta_a^t \rightarrow \theta_a^{t+1}$  を更新する．
- 式 (6.27) によって  $\varepsilon^{\mu,t} \rightarrow \varepsilon^{\mu,t+1}$  を更新する．
- 式 (6.28) によって  $\beta^t \rightarrow \beta^{t+1}$  を更新する．

ステップ 3 (収束条件):

停止条件は  $\Delta = \sum_{i=1}^N |d_i^t - d_i^{t+1}| + \sum_{i>j} |J_{ij}^t - J_{ij}^{t+1}| + \sum_{a=1}^p (|\gamma_a^t - \gamma_a^{t+1}| + |\theta_a^t - \theta_a^{t+1}|) + \sum_{\mu=1}^N |\varepsilon^{\mu,t} - \varepsilon^{\mu,t+1}| + |\beta^t - \beta^{t+1}|$  が初めに設定した小さな正の定数  $\delta = 10^{-8}$  に対して  $\delta > \Delta$  になるときに停止する．このときの  $d^*, J^*$  を最適推定として出力する．そうでないとステップ 1 に戻る．

しかしながら  $d^*, J^*$  はデザイン  $S_i \in \{-1, 1\}^N$  に基づいて推定されたパラメータである．第 2 章で提案したデザイン  $x_i \in \{0, 1\}^N$  で表す製品設計の組合せ最適化モデル (式 (2.2)) のパラメータとして直ちに導入することができない．そのためセッション 6.3 では  $d^*, J^*$  を製品設計モデルに適用する  $h', G'$  への転換を行う方法を議論する．

## 6.3 推定したパラメータを製品設計モデルに適用するパラメータへの転換

製品設計  $\mathbf{S} = (S_1, \dots, S_N) \in \{-1, 1\}^N$  により推定されたパラメータを，設計  $\mathbf{x} = (x_1, \dots, x_N) \in \{0, 1\}^N$  で表す式 (2.2) の製品設計モデルに適用するパラメータとして導入する前に，以下の転換を行う必要がある．ここで  $d_i$  と  $J_{ij}$  を推定したパラメータとし， $h_i$  と  $g_{ij}$  を製品設計モデルに適用するパラメータとし，コスト関数

$$\mathcal{H}(\mathbf{S}) = -\sum_{i=1}^N d_i S_i - \frac{1}{2} \sum_{i=1}^N \sum_{j(\neq i)}^N S_i J_{ij} S_j \quad (6.70)$$

における  $S_i = 2x_i - 1$  の等価な関係により，コスト関数を

$$\begin{aligned} \mathcal{H}(\mathbf{S}) &= -\sum_{i=1}^N d_i (2x_i - 1) \\ &\quad - \frac{1}{2} \sum_{i=1}^N \sum_{j(\neq i)}^N (2x_i - 1) J_{ij} (2x_j - 1) \end{aligned} \quad (6.71)$$

のように書き直すことができる．そして式 (6.71) を式 (2.2) の総合収益の反対，コスト  $-TR(\mathbf{x})$  で表すと，

$$\begin{aligned} -TR(\mathbf{x}) &= -\sum_{i=1}^N h_i x_i - \frac{1}{2} \sum_{i=1}^N \sum_{j(\neq i)}^N x_i g_{ij} x_j \\ &\quad + \text{constant} \end{aligned} \quad (6.72)$$

となる．ただし，

$$h_i = (2d_i - \sum_{j \neq i}^N J_{ij}) \quad (6.73)$$

$$g_{ij} = 4J_{ij}, \quad (6.74)$$

$$\text{constant} = \left( \sum_{i=1}^N d_i - \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N J_{ij} \right) \quad (6.75)$$

である．しかし constant は  $x_i$  に依存しない項目であるため，計算するとき考慮しないことにする．式 (2.2) の製品設計モデルに導入するのが，推定された  $d_i$  と  $J_{ij}$  から転化した  $h_i$  と  $g_{ij}$  である．

## 6.4 数値実験

### 6.4.1 実験設計

提案する推定モデルとボルツマンマシンの有効性を確認するために，3つの数値実験を行う．(1) 小規模の不均一の問題の場合，推定回数の増加につれ推定値と真の値の近さの遷移を測る．(2)

小規模の均一の問題の場合，初期入力  $d_i^0, J_{ij}^0$  と真の  $d_i, J_{ij}$  とワンビットの値しか異なるケースに対して，この近さの遷移を測る．(3) 均一の大規模問題の場合，この近さと設計の的中率を測る．コンピュータは HP Compaq 8100 Elite SFF PC ( OS が Windows 7 , CPU が Intel(R) Core(TM) i3@2.93GHz, メモリが 2.00GB であるを使用した．

### 6.4.2 小規模の不均一の例

ここで推定回数  $t$  の増加につれて，推定値  $d_i^t, J_{ij}^t$  と真の  $d_i, J_{ij}$  との近さの変化を観察する．最終に推定値と真の値と一致するとき，式 (6.21) のラグランジュ関数  $L$  における  $C$  の  $\sum_{\mu=1}^R (f(\frac{1}{N}\mathcal{H}(\mathbf{S}^\mu)) - K^\mu)^2 \rightarrow 0, KL$  の  $\log\left(\frac{P_0(\mathbf{S})}{P(\mathbf{S}|\mathbf{d}, \mathbf{J})}\right) \rightarrow 0$  となることを予想できる．そのためここで， $\Delta_1 = \sum_{\mu=1}^R |f(\frac{1}{N}\mathcal{H}(\mathbf{S}^\mu)) - K^\mu|$  と  $\Delta_2 = \sum_{i=1}^N |\langle S_i \rangle_0 - m_i| + \sum_{j>i}^N |\langle S_i S_j \rangle_0 - m_{ij}|$ ，及びこの2つからなる総誤差  $\Delta^* = \Delta_1 + \Delta_2$  を用いて推定効果を考察する．

ここでインターフェース数  $N = 6, p = 1$  の2段階カテゴリ  $K^\mu \in \{0, 1\}$  の小規模問題に対し，未知変数の数  ${}_N C_1 + {}_N C_2 + 2p = 23$  よりも少ないデザイン数  $\mathcal{N} = 21$  がある場合，5回の数値実験により  $\Delta^*, \Delta_1, \Delta_2$  の平均遷移を図 6.2 に示す．この際に M-H 法のサンプル生成に用いられる  $d_i$  は平均 0，分散 5， $J_{ij}$  は平均  $\frac{0}{N}$ ，分散  $\frac{5}{N}$  の正規分布に従って生成し，ボルツマンマシン推定における  $d_i^0$  と  $J_{ij}^0$  も同じ正規分布によって生成する．また初期設定で  $\gamma^0 = \theta^0 = \varepsilon^{\mu,0} = \beta^0 = 1.0$  とする．

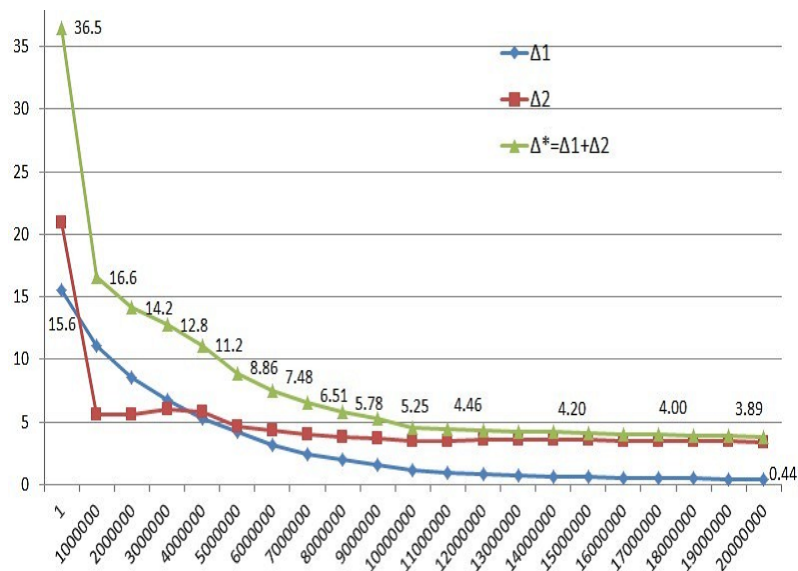


図 6.2: 推定値と真の値との近さの遷移

図 6.2 の横軸は推定回数  $t$  であり，縦軸は  $10^6$  回の推定おきに  $\Delta^*, \Delta_1, \Delta_2$  の値を記録する．6 時間ほどかかった  $t = 2.0 \times 10^7$  回の推定まで，推定の総誤差  $\Delta^*$  が  $t$  の増加につれて安定的に収束する傾向を明らかにした．また  $\Delta_1$  と  $\Delta_2$  の各自の遷移から， $KL$  を併用することは，たとえデザインの数決定変数の数より少なくても  $C$  の働きもうまくかけられることが分かる．さらに，推

定の早期段階 ( $t \leq 4.0 \times 10^6$ ) における  $\Delta_2$  には不安定な傾向が見える．これは  $\Delta^*$  の収束速度に影響するが収束トレンドに影響しない．しかも  $t > 4.0 \times 10^6$  から  $\Delta_2$  も遅くても安定的に減ることが分かる．

この数値実験により， $\Delta^* \rightarrow 0$  までの遷移が時間がかかるが，理論上には充分の時間を経れば提案手法で推定値が真の値に接近することが期待される．

### 6.4.3 小規模の均一の例

インターフェース数  $N = 6$ ， $p = 1$  の2段階カテゴリ  $K^\mu \in \{0, 1\}$  の小規模の均一の問題  $\mathbf{d} = \{1, 1, 1, 1, 1, 1\}$ ， $J = \{\frac{1}{6}\}_{6 \times 6}$  に対し，初期入力  $\mathbf{d} = \{0, 1, 1, 1, 1, 1\}$ ， $J = \{\frac{1}{6}\}_{6 \times 6}$  である数値実験により  $\Delta^*$ ， $\Delta_1$ ， $\Delta_2$  の平均遷移を図 6.3 に示す．ほかの設定は小規模の非均一の例と同じである．

図 6.3 の横軸は推定回数  $t$  であり，縦軸は  $\Delta^*$ ， $\Delta_1$ ， $\Delta_2$  の値を記録する． $t = 10^9$  回 (約 1 週間) の推定で，最後まで誤差  $\Delta_1$ ， $\Delta_2$  が 0 に近づいて，総合誤差  $\Delta^*$  も 1.0 以下に減ったことが分かる．また，推定の早期段階における  $\Delta_2$  には上の例と同じく不安定な傾向が見えるが， $\Delta^*$  の収束速度にしか影響していないことが分かる．

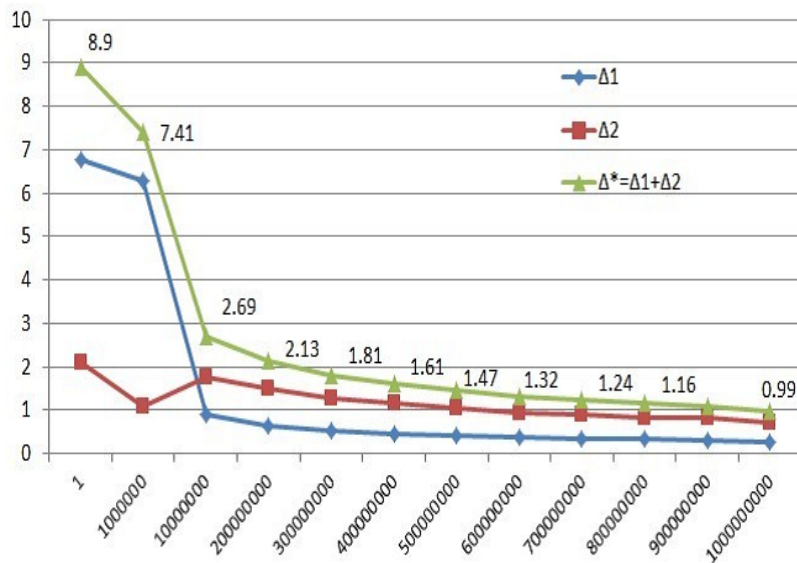


図 6.3:  $N = 6$  の均一の例の推定値と真の値との近さの遷移

### 6.4.4 大規模の均一の例

提案手法で大規模問題を扱う能力を検証するために，ここで  $N = 100$ ， $p = 3$ ， $\mathcal{N} = 5050 < {}_N C_1 + {}_N C_2 + 2p = 5056$  の問題を考察する．この際に M-H 法のサンプル生成に用いられる  $d_i = 1.0$ ， $J_{ij} = 1.0$  とする．式 (2.2) の製品設計モデルに対し，この均一の例の最適デザイン  $x_i = 1$  が自明である．ほかの設定は上と同じである．

$\Delta^*$  の遷移を図 6.4 に示す．横軸は推定回数  $t$  であり，縦軸は 5000 回の推定おきに  $\Delta^*$  の値である． $t = 10^5$  回の推定まで 30 時間ほどかかった．図 2 により推定回数の増加につれ  $\Delta^*$  が収束する傾向を見える．また図 6.5 に推定値で求めたデザインの的中率を記録する．この  $x^t$  を用いて的中率  $r_{ob}^t = \frac{\sum_{i=1}^{100} x_i^t}{100}$  を計算する．図 6.5 から推定回数の増大につれ的中率  $r_{ob}$  が改善し続け，最終に  $r_{ob} = 1.0$  を達成することが分かる．

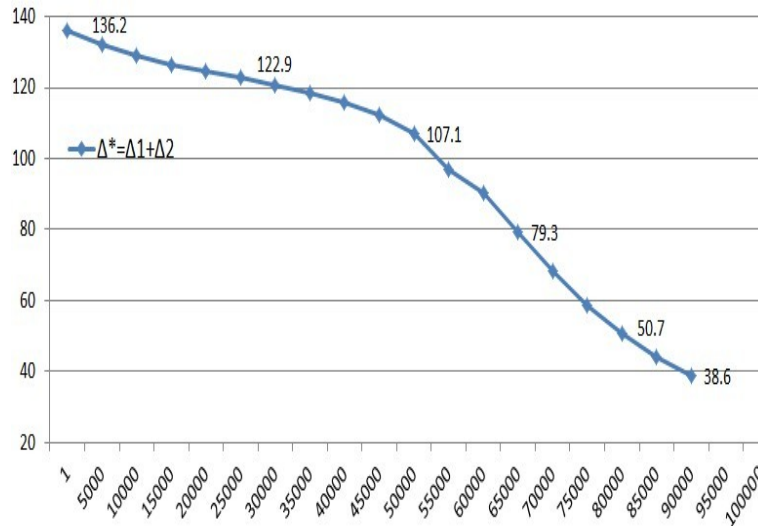


図 6.4:  $N = 100$  の均一の例の  $\Delta^*$  の遷移

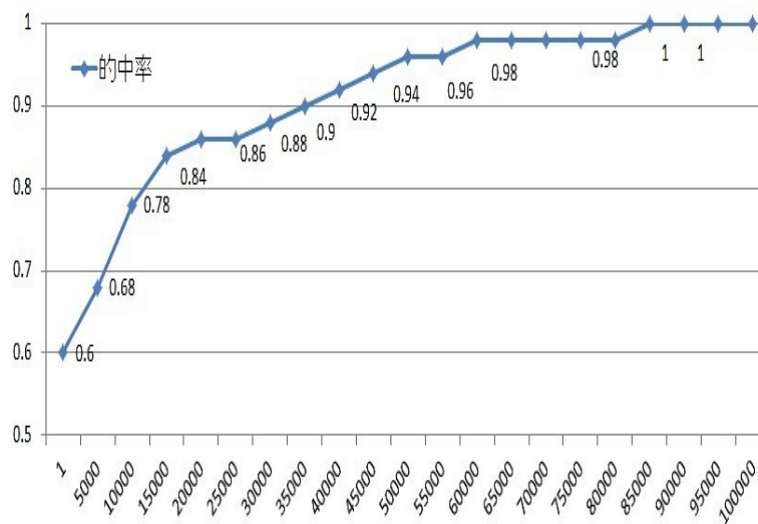


図 6.5:  $N = 100$  の均一の例の的中率の遷移

## 6.5 まとめ

我々が提案した製品設計の組合せ最適化モデルが解くことを解決されたが，計算に用いたパラメータ（インターフェースそのものの効用とインターフェース間の相互作用）の値が与えられてい

とは言えない．また設計者の暗黙知によってすべての実数のパラメータを精確に定めることが簡単ではないことが分かる．そこで，本研究では製品デザインとそのカテゴリ評価を利用してパラメータを数理的に見積もることを目的として，パラメータの推定モデルを提案する．しかし，提案する推定モデルは方程式の連立でパラメータを推定することにより，未知変数（パラメータ）の数よりも少ないデザインの数しか提供されない可能性がある．このときは解の一意性が保証されない．そこでさらに本研究でボルツマンマシンを援用して学習誤差が0の下で汎化誤差を最小にする推定問題を解いた．これにより製品設計問題における価値あるパラメータを定着することができるようになる．

小規模の数値実験によって提案手法の有効性を検証する．結果として，推定回数の増加につれて推定誤差が安定的に収束する傾向を明らかにした．また均一の大規模の例の推定誤差と的中率の遷移を観察することによって，提案手法で大規模問題を扱う能力を検証した．

設計者により提供するデザインとそのカテゴリ評価の生データを用いて，パラメータを推定する上に製品設計モデルを解くことにより，最適なデザインを再現することは次の研究目標である．また，製品設計の従来研究 QFD, DSM などと比較して本研究の実用性を検証することも重要である．

## 第7章 Pメジアン法による最適解のクラスタリング

製品設計に対する統一的な見方を提供するアーキテクチャ理論を構築することは、製品開発研究の中の重要な課題の1つである。種々の候補方案のうち、最適なアーキテクチャを見出すにあたって、これまで設計者の主観的な判断に依存しており、製品そのものの客観的な情報を用いて設計されているとは言いがたい。その問題に対して、我々は製品設計の組合せ最適化モデルを構築し、製品の機能同士間のつながり情報によって、インテグラルを考慮に入れた製品設計手法を定式化していた。さらに第3章と第4章で議論したように、数理解析手法である分枝限定法及び統計力学的アプローチを用いて最適的な製品設計を導出することが成功した。しかしながら、実際の製品設計に用いるためにはいくつかの点でモデルを改良する必要がある。例えば提案する製品設計の組合せ最適化モデルによって最適設計を見出したとする上で、「システムを構築する」という経営工学の観点から、製品設計の後工程を考える意義がある。なぜなら、1つの完成品には多くの部品と中間製品が含まれており、それらを特定し、生産拠点を分散させることで、単一拠点である生産能力の限界を避けられること、あるいは全体としてのシステムの効率化が図れることが知られているが、先行研究ではこの問題に対して解答することができていない。そこで我々は、機能情報のある経験則の解決策とクラスタリング手法であるPメジアン法(p-median)を用いて、中間製品を導出するアーキテクチャ理論を構築する[18][59]。また数値実験を用いて提案手法の有効性を検証する。

本章の構成は以下の通りである。第1節では機能情報のある経験則による最適的な製品設計のクラスタリングの可能性を分析する。第2節では整数計画のPメジアン法を援用し、機能同士間つなくかつながないかの客観情報を考慮にいれた最適的な中間製品を導出するクラスタ識別モデルを構築する。第3節では提案手法の効果を考察する。最後に第4節では結論と将来の課題を紹介する。

### 7.1 中間製品のクラスタ識別モデル

単一生産拠点である生産能力の限界と全体としてのシステムの効率化を考慮した上に、多くの機能デバイスからなる1つの完成品を競争的な中間製品への割り当てる能力も問われる。例えば、CPU、CPUクーラーなどがマザーボードをベースにして一つの間製品を、ハードディスクを別の中間製品を特定できると考えられる。最適的な製品設計を見出した上に、いくつかのクラス

タに切り分ける時，たとえば4つの機能を2つのクラスタに配分しようとする場合，共に ${}_4C_2 = 6$ 種の方案の中から，そもそもどちらが一番良いのだろうか．また，既存の2つのクラスタを3つのクラスタに更新しようとする場合を考えると，どのようなクラスタ方案につながるのだろうか．これらの問題を解決するために，クラスタ方案を評価できるクラスタリングアルゴリズムを設計しておかなくてはならない．

クラスタリングは分類対象を類似性やある定められた距離尺度に基づきいくつかのクラスタに分割し，クラスタ中の分類対象の類似性が高い，クラスタ間の分類対象の類似性が低いことを目指す．この技術では中間製品の識別を扱える．クラスタリングの主な先行研究を紹介する．早い段階でクラスタリング技術は人工的かつ主観的に大きく頼む段階に入っており，クラスタリングの自動化と共通化を実現させることが困難である．それから数理的な目標関数を提出することでクラスタリングの自動化と共通化の能力を向上したが，複雑な問題（例えばオーバーラップ）にはやはり能力不足であることが分かる．そして，図7.1に示すように従来のクラスタリング研究はDSMのデータだけを対象にするので，本研究に提案された製品設計問題のモデリングには合わない．

そこで本章では，提案した製品設計問題のモデリングに基づいて，機能情報のある経験則によるクラスタリングで製品設計の中間製品のクラスタ識別を行うことが可能であることを証明した上で，Pメジアン法を用いて，総合的にインタフェースのつながり情報を最大限に考慮できる最適な中間製品配置を導出するクラスタ識別モデルを提案し，これまでのアーキテクチャ理論を改良する．

作者	年間	クラスタリングに対する貢献
McCord and Eppinger.	1993	DSM概念の提出
Pimpler and Eppinger.	1994	部品関係をDSMと五段階スコアリングで表し，ヒューリスティックでクラスタリングを行う
Zakarian and Rushton.	2000	線引きのヒューリスティック法
Fernandez.	1998	DSMのデータに基づいてクラスタ方案を評価できる簡単な目標関数を構築することでクラスタリングを行う
Thebeau.	2001	

図 7.1: DSM のクラスタリングに関する先行研究

### 7.1.1 機能情報のある経験則のクラスタリング

製品設計者に対して，2つの単機能製品の機能情報によるすりあわせとは，類似機能を発揮している機能デバイスの合併の過程と見られる．ここで多機能懐中電灯の例を用いてこの合併の過程を再現するかを検証する．



## 機能デバイスの分解

比較のため家庭用単機能懐中電灯 (製品 A で表す) と多機能懐中電灯 (製品 C で表す) を選ぶ。選ぶ理由は簡単に手で分解することができるからである。多機能懐中電灯 (製品 C で表す) におけるナイトライト (製品 B で表す) と懐中電灯 (製品 A で表す) という 2 つの機能を持っている。

製品 A と製品 B を機能デバイスのレベルまでに分解し、それぞれのデバイス情報も図 7.2 に示される。

保護蓋	反射鏡	保護蓋	白色LED反射鏡 <sup>o</sup>
集光レンズ	ライト	アンバーLED反射鏡	白色LEDライト <sup>o</sup>
ライトの支え	スイッチ	アンバーLEDライト	明暗センサー <sup>o</sup>
スイッチチップ	スプリング	電気回路板	切り替えスイッチ <sup>o</sup>
陽極チップ	陰極チップ	スイッチの支え	ニッケル水素電池 <sup>o</sup>
乾電池	乾電池支え	電池の支え	本体 <sup>o</sup>
乾電池チップ	本体	コード	コンセント <sup>o</sup>
		ネジ <sup>o</sup>	

図 7.2: 単機能懐中電灯 (左) と多機能懐中電灯 (右) のデバイス分解

それで図 7.3 で製品 A と製品 B の合成 DSM を作る。図 7.3 の DSM に A と B それぞれの機能デバイスを行と列と対応的に並べ、セルの中に機能デバイス間の依存関係を記録する。製品 A と製品 B 各自の機能デバイス間に依存関係が存在する。すなわち図 7.3 から DSM の左上と右下の 2 つのデータの塊しか現れないことが分かる。一方、製品 A と製品 B の間に機能デバイスの依存関係はまだ明確していないことにより、図 7.3 の DSM の左下と右上が空白となることである。また、機能デバイス間の依存関係は四つの基準 (材料、空間、情報とエネルギー) で表現し、セルごとに左上は空間、左下は情報、右上はエネルギー、右下は材料の基準がそれぞれ 5 段階 (-2, -1, 0, +1, +2) に分けて評価している。機能デバイスの組をつなぐと負の依存が生成される可能性があるから、正の値も負の値もある。また、スコアリングのデータで依存関係の程度を多段階に分けて評価することもできる。例えば、製品 A のライトとライトの支えが空間上に近ければ近いほど、サイズは小さくなることによって、付加利益を生成することができるから、+2 を与える。一方、ライトとその支えのつながりでは、電流伝達によるライトの発光機能に悪影響を与えるため、材料基準をもとに -2 を与える。

一方、研究の関連性を考えると、DSM の依存関係のデータは、我々が提案したモデルにおけるインターフェース自身の単独収益  $h_i$  を反映できるものと考えられる。例えば、単独収益  $h_i > 0$  のとき、依存関係のデータは正であり、単独収益  $h_i < 0$  のとき、依存関係のデータは負である。

DSM(A)	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	B1	B2	B3	B4	B5	B6	B7	B8	B9
A1 蓋	2:0 0:0													1:0 0:0									
A2 反射鏡	2:0 0:0	2:0 0:0	2:2 0:0	2:2																			
A3 集光レンズ		2:0 0:0	2:2 0:0	2:2																			
A4 ライト		2:2 0:0	2:2 0:0	2:2	2:0 0:2	2:0 0:2		-2:0 0:2	2:0 0:2	-2:0 0:2	2:0 0:2	-2:0 0:2	-2:0 0:2										
A5 ライト 支え			2:0 0:2	2:0 0:2	2:0 0:2			2:0 0:2	2:0 0:2	2:0 0:2				1:0 0:0									
A6 イッチチップ			2:0 0:2	2:0 0:2	2:0 0:2			2:0 0:2	-2:0 0:2	-2:0 0:2	2:0 0:2	-2:0 0:2	-2:0 0:2	-2:0 0:2									
A7 スイッチ							2:0 0:0	2:0 0:0							1:0 0:0								
A8 スプリング							2:0 0:0	2:0 0:0						1:0 0:0	2:0 0:0								
A9 陽極チップ			-2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2		-2:0 0:2	2:0 0:2	2:0 0:2	-2:0 0:2	-2:0 0:2									
A10 陰極チップ			2:0 0:2	2:0 0:2	2:0 0:2			-2:0 0:2	2:0 0:2	2:0 0:2				2:0 0:2									
A11 乾電池			-2:0 0:2	-2:0 0:2	2:0 0:2	2:0 0:2		2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2								
A12 電池支え							1:0 0:0	2:0 0:0			2:0 0:2												
A13 電池チップ			-2:0 0:2	-2:0 0:2	2:0 0:2	2:0 0:2		-2:0 0:2	-2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2								
A14 本体	1:0 0:0				1:0 0:0			2:0 0:0		2:0 0:2			2:0 0:2	2:0 0:2									
B1 蓋															2:0 0:0								
B2 反射鏡															2:0 0:0	2:2 0:0							
B3 ライト															2:2 0:0	2:0 0:2	2:0 0:0	0:0 0:0	0:0 0:0	0:0 0:0	0:0 0:0	0:0 0:0	0:0 0:0
B4 ライト 支え															2:0 0:2	2:0 0:2	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0
B5 明暗センサー															2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0
B6 センサー 支え															2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0	2:0 0:0
B7 乾電池															0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2
B8 電池 支え															0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2
B9 コンセント															0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2	0:0 0:2
B10 コード															2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2
B11 本体															1:0 0:0	1:0 0:0	1:0 0:0	1:0 0:0	1:0 0:0	1:0 0:0	1:0 0:0	1:0 0:0	1:0 0:0

図 7.3: 製品 A と製品 B の従来 DSM

製品 A の機能デバイスのクラスタリング

ヒューリスティック方法を用いて分解した製品 A と製品 B 各自の機能デバイスのクラスタリングを行う。図 7.4 はこの方法の概念図を示す。

まず製品 A の従来 DSM セルの中で、空間基準における +2 となる機能デバイスを取り出し、図 7.4 のノード図を作成する。ノードでは機能デバイスを表し、実線(エッジ)では依存関係を表す。また、データ 2 で表す緊密な依存関係しか取らない理由は、機能を実現するためにこのインターフェースの存在が欠かせないことを意味する。次は、ノード図における左から右へいくつかの塊りを新しい DSM 内で順次に並べる。新しい DSM は図 7.5 の左上のようになる。そしてクラスタリングの結果は 1 つの塊りの中にある機能デバイスの順番に関わらないとする。それで、クラスタに搭載する機能別によって、3 つのクラスタはそれぞれコントロールクラスタ (A7, A8 からなる)、保護クラスタ (A1,A2,A3 と A4 からなる) と発光クラスタ (A4,A5,A6 などからなる) と呼ばれる。同じように製品 B のクラスタリングを行い、結果は図 7.5 の右下のようになる。そしてオーバーラップの機能デバイスが生成されたことが分かる。例えば、デバイス A4 のライトは保護クラスタと発光クラスタに含まれる。

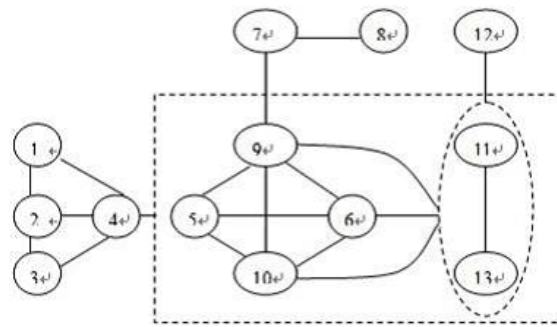


図 7.4: 製品 A の機能デバイスの依存関係を表すノード図

### 空間基準に基づく製品 B の割込みと合併

クラスタリング得られた製品 A の DSM(図 7.5 の左上) の中に，製品 B の類似機能を持つ機能デバイスを割込んで付け加えることを行う．具体的に，DSM の一基準 (4 つの基準の中から空間を選ぶこと) に基づいて下記の操作に従って行うことにする．

1. 製品 A 中にある機能デバイスが存在し，製品 B 中に同じ機能を持つ機能デバイスが存在する場合，製品 B の機能デバイスを対応の製品 A のデバイスの後に並べる．
2. 製品 A 中にある機能デバイスが存在しながら，製品 B 中に同じ機能を持つ機能デバイスが存在しない場合，B の対応の行と列を DSM から取られる．
3. 残りの製品 A と B 中に単独の機能を持つ機能デバイスを DSM の上から順番に並べる．

この操作でオーバーラップの機能デバイスが 1 つのクラスタの中に集中されることにより，クラスタのサイズが大きくなって，デバイスの距離も遠くなってしまふ．そのため，類似な機能デバイスを合併し 1 つのものにする必要がある．この合併過程は機能のすりあわせ (インテグラルともいう) と考えられる．図 7.6 にすりあわせ操作の結果を示す．結果からみると，単機能製品のすりあわせによって多機能製品を生成することが可能である．そして，新製品の中にコントロールクラスタ，保護クラスタと発光クラスタは保留されたが，各クラスタのサイズがむしろあまり大きくなることが分かった．

### 多重基準による結果

ほかの 3 つの基準，材料，情報とエネルギーを空間に加えて，4 基準に基づく製品のすりあわせの結果を考察する．具体的な操作過程は空間基準に基づく製品の合併操作と一致する．結果を図 7.7 に示す．

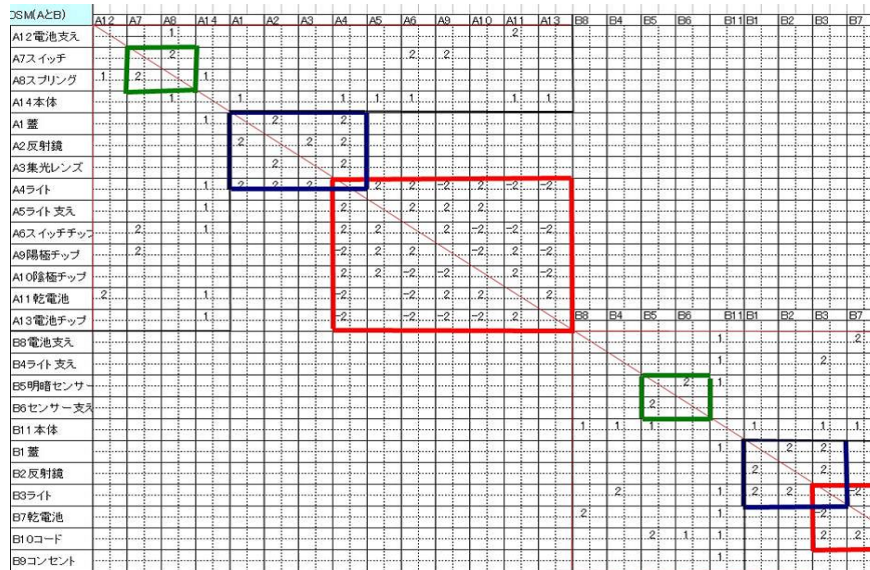


図 7.5: 製品 A と製品 B の各自クラスタリングを行った結果

## 結果考察

機能情報を用いたすりあわせ手法を検証するために、多機能 (懐中電灯とナイトライト) 懐中電灯完成品 C の DSM(図 7.8) を作成し、クラスタリングの結果を図 7.9 に示す。図 7.9 と図 7.7 を比較して、下記の 2 つの結論をまとめる。

1. 機能情報のすりあわせによって得られたクラスターと製品 C 自身のクラスターと比べ、主な機能クラスター (発光、コントロールと保護という 3 つのクラスター) は基本的に一致することを確認した。すなわち、空間の類似機能のすりあわせにより既存製品から新製品を作り出すことができると思われる。
2. 一方、単純な合併では生成された新製品は実の完成品と一致するとは限らない。例えば、懐中電灯のチップとナイトライトのコードが同じの機能を果たしているから両者を 1 つにする。しかし、実製品の中にこの機能を他のデバイス (電気回路板) で代替することが分かる。すなわち、機能デバイスのすりあわせにより製品の増設か取り替えが行う必要もある。

## 7.2 P-median 法による中間製品のクラスター識別モデル

機能情報のある経験則によるクラスタリングは可能であるが、統一的かつ普通的な議論はまだ導出されていないことが分かる。我々はクラスタリング手法である P メジアン法を用いて、最適な中間製品 (最適的とは、総合的にインターフェースのつながり情報を最大限に保留できる) を導出するクラスター識別モデルを提案し、製品設計の後工程の効率の向上を目指す [59]。最適設計の隣接行列をクラスタリングの対象とする。隣接行列は、インターフェースの隣接関係を表現する

DSM(AとB)	A1 2	B8	B4	A7 B5	A8 B6	A1 4	B1 1	A1 B1	A2 B2	A3	A4 B3	A5	A6 B1	A9 B1	A1 0 B1	A1 1 B1	A1 3 B1	B9
A1 2 電池支え				1														2
B8 電池支え							1											2
B4 ライト 支え								1			2							
A7 B5 スイッチ/センサー				2			1						2	2	2			2
A8 B6 スイッチ支え	1			2		1							1	1	1			1
A1 4 本体					1			1			1	1	1				1	1
B1 1 本体		1	1	1				1			1		1	1	1	1	1	1
A1 B1 蓋					1	1			2		2							
A2 B2 反射鏡									2		2	2						
A3 集光レンズ										2	2							
A4 B3 ライト			2		1	1			2	2	2		2	2	-2	2	-2	-2
A5 ライトの支え						1					2		2	2	2			
A6 B1 0 スイッチチップコード				2	1	1	1				2	2		2	-2	-2	-2	2
A9 B1 0 隔極チップコード				2	1		1				-2	2	2		2	2	-2	2
A1 0 B1 0 陰極チップコード				2	1		1				2	2	-2	-2		2	-2	2
A1 1 B7 乾電池	2	2				1	1				-2		-2	2	2			2
A1 3 B1 0 電池チップコード				2	1	1	1				-2		-2	-2	-2	2		2
B9 コンセント								1					2	2	2	2		2

図 7.6: 製品 A と製品 B のすりあわせによって生成した DSM

0-1 行列  $A = \{a_{ij}\}_{M \times M}$  であり、各成分は  $a_{ij} = \{0, 1\}, i = 1, 2, \dots, M$  である。M は機能デバイスの数である。最適な製品設計をいくつかのクラスタに切り分けようとするとき、依存関係の存在する機能デバイスをできるだけ同じクラスタ中に割り当てることを実現するのがクラスタリングの目的である。

### 7.2.1 整数計画問題の P-median 法

機能  $i$  と機能  $j$  間のインターフェースの配置を  $a_{ij} = \{0, 1\}$  で表す隣接行列  $A = \{a_{ij}\}_{M \times M}$  に対して、次の関数の最大化問題を考える。

$$\text{Max } P(E) = \sum_{i=1}^M \sum_{j=1}^M a_{ij} e_{ij} \tag{7.1}$$

ただし  $E = \{e_{ij}\}_{M \times M}$  は次の制約条件を満たす。

$$\sum_{i=1}^M e_{ii} = p, p \leq M \tag{7.2}$$

$$\sum_{i=1}^M e_{ij} = 1, j = 1, 2, \dots, M \tag{7.3}$$

$$e_{ij} \leq e_{ii}, i = 1, 2, \dots, M, j = 1, 2, \dots, M \tag{7.4}$$

$$e_{ij} = \{0, 1\}, i = 1, 2, \dots, M, j = 1, 2, \dots, M \tag{7.5}$$

クラスタリングの目標は、

$$\arg \max_{a_{ij}, e_{ij}} P(E) = (E^*) \tag{7.6}$$

DSM(AとB)	A12	B8	B4	A7B5	A8B6	A14	B11	A1 B1	A2B2	A3	A4B3	A5	A6A9	A11 B	B9
A12電池支え					1 0 0 0										2 0 0 -2
B8電池支え							1 0 0 0								2 0 0 -2
B4ライト支え							1 0 0 0				2 0 0 -2				
A7B5スイッチ/センサー					2 0 0 0		1 0 0 0						2 2 2 0		
A8B6スイッチ支え	1 0 0 0				2 0 0 0	1 0 0 0								1 0 0 0	
A14本体					1 0 0 0			1 0 0 0			1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 -2	
B11本体		1 0 0 0	1 0 0 0	1 0 0 0				1 0 0 0		1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 -2	1 0 0 0	1 0
A1 B1蓋						1 0 0 0	1 0 0 0		2 0 0 0		2 0 0 0	2 0 0 0			
A2B2反射鏡									2 0 0 0	2 0 0 0	2 2 0 0	2 2 0 0			
A3集光レンズ									2 0 0 0	2 2 0 0	2 2 0 0	2 0 0 0			
A4B3ライト			2 0 0 -2		1 0 0 0	1 0 0 0		2 0 0 0	2 0 0 0	2 2 0 0	2 2 0 0	2 0 0 0	2 0 0 -2	2 0 0 2	2 0
A5ライトの支え					1 0 0 0						2 0 0 -2	2 0 0 -2			
A6A9A10A13B10スイッチチップ					2 2 2 0	1 0 0 0	1 0 0 0	1 0 0 0			2 0 0 0	2 0 0 0		-2 0 0 2	2 0
A11 B7乾電池	2 0 0 -2	2 0 0 -2					1 0 0 -2	1 0 0 -2						-2 0 0 2	2 0
B9コンセント														2 0 0 2	2 0

図 7.7: 4 基準 (材料, 空間, 情報とエネルギー) のすりあわせの結果

である。

ただし,  $E$  はクラスタ識別のマトリクスである.  $p$  はメジアン点の個数, あるいはクラスタの数である.  $P(E)$  は各クラスタの中にインターフェース情報の緊密さを反映する総合程度である. 機能デバイス  $i$  と  $j$  がつながれば, 両方を 1 つのクラスタに属させることによって, 各クラスタの中にインターフェースの緊密程度が高くなる傾向がある. また,  $a_{ij} = \{0, 1\}$  は機能デバイスの組のインターフェース配置を表す変数である.  $e_{ii} = \{0, 1\}$  は対角線のデータであり,  $e_{ii}$  の値機能デバイス  $i$  をメジアン点として選ばれるかの判断を表す.  $e_{ii} = 1$  であれば機能デバイスがメジアン点として選び,  $e_{ii} = 0$  であればそうでないことを意味する. また,  $e_{ij} = \{0, 1\}$  の値は非メジアン点である機能デバイス  $j$  がメジアン点である機能デバイス  $i$  のクラスタの中に割り当てられるかの判断を表す.  $e_{ij} = 1$  であれば機能デバイス  $j$  がメジアン点  $i$  のクラスタの中に配置されることを意味する. 一方, 機能デバイスがメジアン点でない場合と, 機能デバイス  $j$  がメジアン点  $i$  のクラスタの中に配置されない場合  $e_{ij} = 0$  である.

制約条件 7.2 では, マトリクス  $E$  の対角線にある  $M$  個のデータから  $P$  個のメジアン点を選び,  $p$  個の  $e_{ii} = 1$  であることを制限する. 1 つのメジアン点ごとに 1 つのオーバーラップでないクラスタを持つ. 制約条件 7.3 では, マトリクス  $E$  の 1 つの列に並んでいるすべての行の値の和が 1 であることを制限する. すなわち任意の 1 つの機能デバイスが一つだけのクラスタの中に切り分けられ, 2 つのクラスタに共有するオーバーラップデバイスが存在しないことを確保する. 制約条件 7.4 では, ある機能デバイスが非メジアン点であれば, ほかのデバイスはこの非メジアン点に配置されることがないことを表す.

上の議論により, 提案するモデル 7.1 ではクラスタ各自の緊密さの和を計算できる. 具体的に, 非メジアン点  $j$  とメジアン点  $i$  のインターフェースが存在すれば  $a_{ij} = 1$  とし, 機能デバイス  $j$  がメジアン点  $i$  のクラスタの中に割り当てること ( $e_{ij} = 1$ ) でクラスタの緊密さを向上することが望

DSM(C)	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
C1 蓋	2:0 0:0											1:0 0:0			
C2 白色LEDの反射鏡	2:0 0:0		2:2 0:0	2:2 0:0											
C3 アンバーLEDの反射鏡	2:2 0:0				2:2 0:0							1:0 0:0			
C4 白色LEDライト	2:2 0:0				-1:0 0:-2		1:0 0:2	0:0 2:0		0:0 0:2		1:0 0:-2	2:0 0:2		
C5 アンバーLEDライト			2:2 0:0	-1:0 0:-2		0:0 2:0	1:0 0:2	0:0 2:0		0:0 0:2		1:0 0:-2	2:0 0:2		
C6 明暗センサー					0:0 2:0		1:0 0:0			0:0 2:0		2:2 0:0	0:0 0:2		
C7 電気回路板			1:0 0:2	1:0 0:2	1:0 0:2	1:0 0:0				1:0 0:2	1:0 0:-2	2:0 0:2	2:0 0:2	1:0 0:0	2:0 0:-2
C8 切替スイッチ			0:0 2:0	0:0 2:0					2:0 0:2	1:0 2:0		1:0 0:0	0:0 2:0	0:0 2:0	
C9 スwitchの支え								2:0 0:0				1:0 0:0			
C10 ニッケル水素電池			0:0 0:2	0:0 0:2	0:0 2:0	0:0 2:0	1:0 0:2	1:0 2:0		2:0 0:-2		1:0 0:-2	2:0 0:2	-1:0 0:2	
C11 電池の支え							1:0 0:-2			2:0 0:-2		1:0 0:0			
C12 本体	1:0 0:0		1:0 0:0	1:0 0:-2	1:0 0:-2		2:0 0:-2	1:0 0:0	1:0 0:0	1:0 0:-2	1:0 0:0	0:0 0:-2	0:0 0:2	1:0 0:-2	2:0 0:0
C13 コード			2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	2:0 0:2	0:0 2:0		2:0 0:2		0:0 0:-2	2:0 0:2	0:0 0:2	
C14 コンセント					0:0 2:0	1:0 0:2	0:0 2:0	0:0 2:0		-1:0 0:-2		1:0 0:-2	2:0 0:2		
C15 ネジ						2:0 0:-2						2:0 0:0			

図 7.8: 製品 C の従来 DSM

まれるようになる。

### 7.3 数値実験

整数計画の p-median 法の実行可能解は機能デバイスの数  $M$  とクラスタの数  $P$  の組合せによって求められる。例えば機能デバイスの数  $M = 5$ ，設定したクラスタの数  $p = 3$  の場合，対角線の 5 つのデータからそのうちの 3 つをランダムに選んでメジアン点とし，非メジアン点である残りの 2 つの機能デバイスをそれぞれ，3 つのクラスタのうちの任意 1 つの中に割り当てることを考える。このとき式 (7.6) のように，解候補の数は  $\Omega(5, 3) = {}_N C_3 ({}_3 C_1)^{5-3} = 90$  個である。三つの数値例で提案手法の有効性を証明する。

$$\Omega(M, p) = {}_M C_p ({}_p C_1)^{M-p} \tag{7.7}$$

#### 7.3.1 7 個の機能デバイスを持つ懐中電灯の例

懐中電灯の例を用いて具体的な操作を説明する。図 7.10 が 7 つの機能デバイスを含む最適設計案であるとする。

ここで最適解をグラフで表現するための隣接行列を生成する。最適設計における機能デバイス間のインターフェースが存在したら 1 で，そうでなければ 0 で表す。対角成分が  $a_{ii} = 0$  であり，

DSM(C)	C15	C11	C8	C9	C6	C12	C1	C2	C3	C4	C5	C7	C13	C10	C14
C15ネジ						2:0						2:0			
C11電池の支え						0:0						0:-2			
C8切替スイッチ			2:0	2:0	1:0					0:0	0:0		0:0	1:0	0:0
C9スイッチの支え			2:0	0:0						2:0	2:0		2:0	2:0	2:0
C6明暗センサー			0:0								0:0	1:0	2:2	0:0	0:0
C12本体	2:0	1:0	1:0	1:0			1:0	1:0	1:0	1:0	2:0	0:0	0:0	1:0	1:0
C1蓋	0:0	0:0	0:0	0:0			1:0				2:0	0:0	0:0	2:0	2:0
C2反射鏡1							2:0	2:2	2:2	2:2					
C3反射鏡2							0:0	0:0	0:0	0:0					
C4ライト1			0:0			1:0		2:2		-1:0	1:0	2:0	0:0		
C5ライト2			2:0			0:0		0:0		0:-2	0:2	0:2	0:2		
C7電気回路板	2:0	1:0			1:0	2:0				1:0	1:0		2:0	1:0	1:0
C13コード	0:-2	0:-2			0:0	0:0				0:2	0:2		1:0	0:2	0:2
C10ニッケル水素電池			0:0		2:2	2:0				2:0	2:0	2:0	2:0	2:0	2:0
C14コンセント			2:0		0:0	0:0				0:2	0:2		0:2	0:2	0:2
			0:0		2:0	2:0				0:0	0:0		1:0	2:0	-1:0
			2:0		2:0	2:0				0:2	0:2		0:2	0:2	0:2

図 7.9: 製品 C のクラスタリングの結果

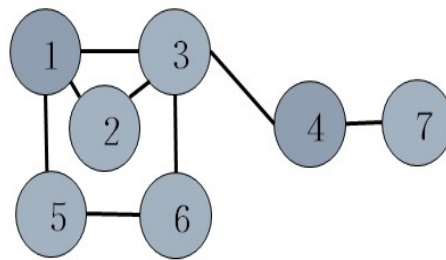


図 7.10: 仮定した 7 つの機能デバイスを含む最適設計案

対角でない成分が対称的である．図 7.10 の隣接行列  $a_{ij}$  は

$$a_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

である．例えば 7 つの機能デバイスを 2 つのクラスタに配分しようとするとき， $p = 2$  でモデル 7.1 の解析を行う．提案する P-median 法によるクラスタリングの結果として， $P(E) = 2$  が最大



値 4 を取るとき，最適的なクラスタ識別マトリクス  $E$  は下記

$$e_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

のようになる．この結果は線引きによるヒューリスティックの結果と一致する．対角線 1 番目と 4 番目のデータが 1 となることによって，それらに対応する機能デバイス 1 と 4 はメジアン点である．機能デバイス 1 と 4 がそれぞれ 1 つのクラスタを形成する．また，1 行目と 4 行目における対角線でないデータが 1 であれば，その列に対応する機能デバイスがデバイス 1 かデバイス 4 のクラスタに属することが分かった．図 7.11 は図 7.10 のクラスタリングの結果であり，(12356) と (47) の二つのクラスタを生成した．

表 7.1: 7 個の機能デバイスを持つ懐中電灯の例に対して，異なる  $p$  による最適的なクラスタ配置

$p$ の数	最適クラスタ	$P(E)$
$p=2$	(1,2,3,5,6)(4,7)	4
$p=3$	(1,2,3)(4,7)(5,6)	4
$p=4$	(1,3)(2)(4,7)(5,6)	3
$p=5$	(1),(2),(3)(4,7)(5,6)	2
$p=6$	(1),(2),(3)(4,7)(5),(6)	1

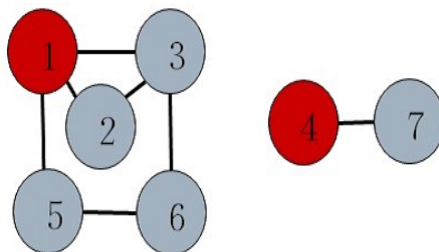


図 7.11: 機能デバイス 1 と 4 がメジアン点であるときの 2 つのクラスタ

さらに，異なる  $p$  による最適的なクラスタ配置を表 7.1 に示す．クラスタの個数を  $p = 4$  から  $p = 3$  に減ると，クラスタ (2) がクラスタ (1, 3) に合併し，一つのクラスタ (1, 2, 3) になる．図 7.10 に示すように，機能デバイス 1 と 3 の両方のつながりがあるから，機能デバイス 2 が 4, 5, 6 と 7 より一番先にクラスタ (1, 3) に受け入れられることが分かった．図 7.12 では 3 つのクラスタ

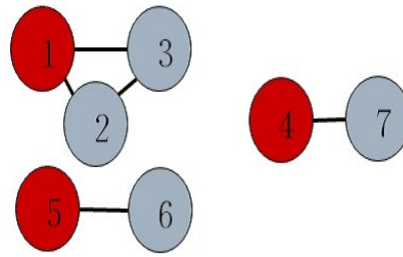


図 7.12: 機能デバイス 1, 4 と 5 がメジアン点であるときの 3 つのクラスタ

の結果を示す．また，クラスタの個数  $p = 3$  から  $p = 2$  に減ると，クラスタ (5, 6) がクラスタ (1, 2, 3) の中に合併される．機能デバイス 5 と 6 はクラスタ (1, 2, 3) とのつながりがあるからである．そのため，提案したクラスタ識別モデルでは合理的なクラスタを得ることができると考える．また，最大値  $P(E)$  をとる最適解が複数個存在することがある．この中にヒューリスティック法のみによって得られない最適解を見出す可能である．

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Radiator A		2 0			2 -2											
Engine Fan B	2 0				0 0								1 0			
Heater Core C				1 0			2 0	-1 0								0 0
Heater Hoses D			1 0		0 0			0 0	0 0		-1 0					0 2
Condenser E	2 -2	2 0				0 2		-2 2								
Compressor F						0 2		0 2	1 0	0 0	0 0		1 0			
Evaporator Case G			2 0			0 2		2 0	0 2	2 0	2 0			2 0	2 0	2 0
Evaporator Core H			0 0					0 0						0 0	0 0	0 2
Accumulator I				-1 0		1 0		1 0		1 0						
Refrigeration Controls J						0 0		0 0		1 0		0 0	1 0			
Air Controls K						2 0		0 0		0 0		2 0	1 0	0 0	0 0	
Sensors L												0 0	1 0			
Command Distribution M		1 0				1 0				1 0	1 0	1 0		1 0	1 0	1 0
Actuators N							2 0						0 0	1 0		
Blower Controller O							0 0					2 0	0 0			2 0
Blower Motor P			0 0				2 0	0 0					1 0		2 0	

図 7.13: 16 個の機能デバイスを持つカーエアコンの例 (Ford Motor 会社)[16]

### 7.3.2 16 個の機能デバイスを持つカーエアコンの例

MIT に開発されてきた DSM が用いられた Ford Motor 会社のカーエアコン (図 7.13) の例で提案手法の効果を検証する．このサイズの問題に対して， $M = 16, p = 2$  のときの解空間の数が  $1.97 \times 10^6$  であり， $M = 16, p = 3$  のときの解空間が  $8.93 \times 10^8$  である．前者の場合 120 秒ごろ最適解を探索できたが，後者には 5 日間の実行見通しがある．そのため，全数探索の能力を考慮した上でここで  $M = 16, p = 2$  のときのクラスタリングしか行わないことにする．DSM のデータから対応な隣接行列を作る方法として，DSM のデータが 2 の場合，隣接行列  $a_{ij} = 1$  で表す．図 7.14 はカーエアコンの隣接行列により生成する機能デバイス間の関連図である．また，ほかのすべて

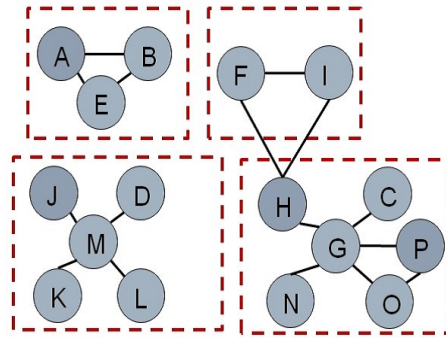


図 7.14: カーエアコンの隣接行列により生成する機能デバイス間の関連図

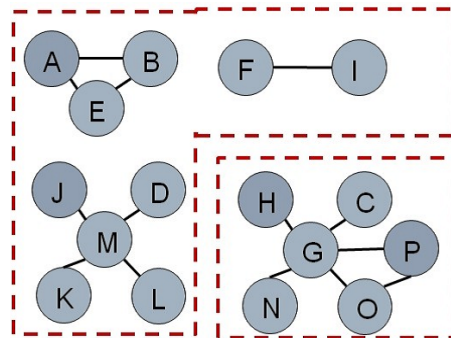


図 7.15:  $p = 2$  のとき, カーエアコンのクラスタリングの結果

のデータ  $0, -1, -2$  と  $1$  を  $a_{ij} = 0$  に転化させる.  $p = 2$  を設定するとしたら, 提案する P-median 法によるクラスタリングの結果 (図 7.15) はヒューリスティックの結果と一致し, (ABDEFIJKLM) と (CGHNOP) の 2 つのクラスタを識別した. しかしこの規模の問題に対して, 探索時間の制限で最適解のパラメータ  $p$  が分からないことが欠点である.

### 7.3.3 非対称の隣接行列を持つ数値例

最後に MIT の非対称の隣接行列を持つ 7 つの機能デバイスで効果を検証する. 図 7.16 は隣接行列の生成には参考になるつながりの配置図である. 符号  $\times$  が現れたセルには, 対応的に隣接行列のセルの値が  $1$  となることを意味する. 表 7.2 では異なる  $p$  による最適的なクラスタ配置を示す.  $p = 2$  以外の  $P(E)$  の値がそれより小さいことによって, クラスタの緊密さから  $p = 2$  以外のクラスタ結果が最適解でないことが分かった. また提案する P-median 法によるクラスタリング結果 (図 7.17) は (AEF) と (BCDG) の 2 つのクラスタを生成した. この結果は MIT によって提供した真の最適クラスタと一致する.

	A	B	C	D	E	F	G
A	■				X	X	
B		■		X			X
C		X	■	X			X
D		X	X	■	X		X
E				X	■	X	
F	X				X	■	
G		X	X	X			■

図 7.16: 非対称の隣接行列を持つ数値例の隣接行列

表 7.2: 非対称の隣接行列を持つ数値例に対して, 異なる  $p$  による最適的なクラスタ配置

$p$ の数	最適クラスタ	P(E)
$p=2$	(A,E,F)(B,C,D,G)	5
$p=3$	最適解ではない	4
$p=4$	最適解ではない	3
$p=5$	最適解ではない	2
$p=6$	最適解ではない	1

## 7.4 まとめ

単一拠点の生産能力の限界や生産拠点の分散などの理由で, 製品を機能デバイスから完成品までいくつかの企業に分け, 製品最適設計における中間製品を統一的な見方で提供することを議論した. 本研究では機能デバイス同士のつなぐかつながないかの客観的な情報を考慮に入れて, Pメジアン法を導入し, いくつかの中間製品の配置方案を考案するクラスター識別モデルを構築し, 3つの従来研究の数値例を用いて全数探索でその有効性を示した. このモデルの基本思想は, インターフェース間のつながり情報をクラスターを切り分けにしても最大限に考慮でき, 最も総合緊密さを持つクラスター案を求める Pメジアンモデルを構築した.

しかしながら, パラメータ  $p$  の設定は設計者の経験や暗黙知などに頼ることがある. 整数の制約条件 (5) を  $0 \leq e_{ii} \leq 1$  でファジデータに転化し, すなわちここで扱われた整数計画の Pメジアン法を非整数計画問題に拡大し, 内点法を用いてモデルの効果を向上していくことは今後の課題としたい. また, Pメジアンモデルでまだ解決できていない複雑な問題 (クラスター間のオーバーラップが存在すること) を考える必要がある. 最後に, Pメジアンモデルの NP 困難問題を解決するために, 遺伝的アルゴリズム GA などのメタヒューリスティックを援用して, 大型問題の最適クラスタ配置を見つけることも研究課題の 1つとしたい.

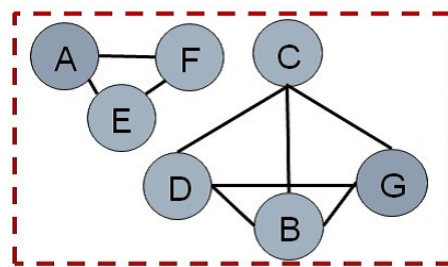


図 7.17: 得られた最適のクラスタリング結果



## 第8章 提案する製品設計アプローチの有効性を検証するための数値例

第2章「製品設計のための最適化モデルの提案」、第3章「分枝限定法による2次モデルの厳密解析」と第6章「ボルツマンマシンによるモデルパラメータの推定解析」の内容に基づいて、ここで1つの数値例を用いて提案手法では既存製品の最適設計が再現されることができるとを検証する。

### 8.1 実験設計

パソコンの4つの機能デバイス、CPU、ハードディスク、マザーボードと扇風機を選び、この4つの機能デバイスを1つの完成品あるいは完備機能を備えたモジュールを生成する問題を考える。機能デバイス  $M = 4$  のとき、インターフェースの最大個数  $N$  は  $N = M(M - 1)/2 = 6$  となる。すべてのインターフェースの配置からなる製品設計の候補案は合計  $2^6 = 64$  個がある。図 8.1 で既存のインターフェースの実状態を表す。ただし、データがインターフェースの番号である。実線でインターフェースが存在すべきことを意味し、点線でそうでないことを意味する。この実状態は提案する製品設計の2次組合せ最適化モデルの決定変数で表すと  $N$  は  $\mathbf{x}^* = \{1, 0, 1, 0, 1, 1\}^T$  となる。 $\mathbf{x}^* = \{1, 0, 1, 0, 1, 1\}^T$  も最適設計とみられる。

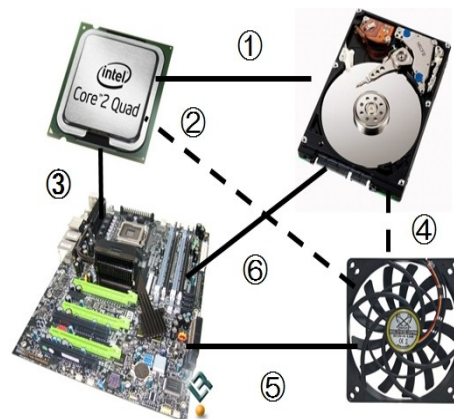


図 8.1: 数値例のインターフェース配置の実状態

この例の場合，製品設計のための2次組合せ最適化モデルは

$$\text{Max } TR(\mathbf{x}) = \sum_{i=1}^6 h_i x_i + \frac{1}{2} \sum_{i=1}^6 \sum_{j(\neq i)}^6 g_{ij} x_i x_j \quad (8.1)$$

$$\text{s.t. } 0 \leq \sum_{i=1}^N x_i \leq a, a \leq 6, \quad (8.2)$$

$$x_i = \{0, 1\}, i = 1, 2, \dots, 6 \quad (8.3)$$

となる．

さらに，この2次組合せ最適化モデルを解く前に，設計案  $S^\mu$  とそのカテゴリ評価値  $K(S^\mu)$  を用いてコスト関数  $\mathcal{H}(S^\mu)$  のパラメータを求めないといけない．

$$\text{Min } \mathcal{H}(S^\mu) = - \sum_{i=1}^6 d_i S_i^\mu - \frac{1}{2} \sum_{i=1}^6 \sum_{j(\neq i)}^6 J_{ij} S_i^\mu S_j^\mu \quad (8.4)$$

ここで実状態の情報に基づき，カテゴリ評価の観測値  $K(S^\mu)$  を生成することを考える．具体的に，実状態によってインターフェース  $i$  のつながりがあれば，例えばインターフェース 1, 3, 5 と 6 の場合， $x_i^* = 1$  である．これがインターフェースの存在が期待されるため，このインターフェース自身の収益  $h_i \geq 0$  を与えられると考える．一方，実状態のインターフェース  $i$  が存在しなければ，例えばインターフェース 2 と 4 の場合， $x_i^* = 0$  である．これがインターフェースの存在が期待されないため収益  $h_i < 0$  と設定してよい．式 (8.5) で実状態に基づく収益  $h_i$  の値の制約条件を示す．また，インターフェースの組には同様に，実状態によってインターフェース  $i$  と  $j$  が同時に存在していれば  $x_i^* x_j^* = 1$ ，その相互作用が総合収益に悪い影響を与えないため， $g_{ij} \geq 0$  を設定して良い．そうでないと， $x_i^* x_j^* = 0$  の場合  $g_{ij} < 0$  を設定して良い． $g_{ij}$  の値の制約条件は式 (8.6) で表す．

$$\begin{cases} x_i^* = 1 & h_i \geq 0 \\ x_i^* = 0 & h_i < 0 \end{cases} \quad (8.5)$$

$$\begin{cases} x_i^* x_j^* = 1 & g_{ij} \geq 0 \\ x_i^* x_j^* = 0 & g_{ij} < 0 \end{cases} \quad (8.6)$$

上の議論によって，実状態の情報に基づき，ここで設計案のカテゴリ評価の観測値  $K(S^\mu)$  を生成する  $\mathbf{h}$  と  $G$  は， $\mathbf{h} = \{1, 0, 1, 0, 0, 1\}^T$  と下記の  $G$  を設定する． $\mathbf{h}$  と  $G$  を式 (8.1) の製品設計のための2次モデルに代入し，モデルで実状態の最適解  $\mathbf{x}^* = \{1, 0, 1, 0, 1, 1\}^T$  を成功に再現した．

$$g_{ij} = \begin{pmatrix} 0 & -1 & 1 & -1 & 1 & 1 \\ -1 & 0 & -1 & 1 & -1 & -1 \\ 1 & -1 & 0 & -1 & 1 & 1 \\ -1 & 1 & -1 & 0 & -1 & 0 \\ 1 & -1 & 1 & -1 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 0 \end{pmatrix}$$



さらに，上の  $h$  と  $G$  によって設計案  $S^\mu$  とその設計案のカテゴリ評価値  $K(S^\mu)$  を計算する．ここで  $2^6 = 64$  個の設計案の中にランダムに重複でない下記の 21 個の設計案を選ぶ．21 という値は  ${}_1C_6 + {}_2C_6 = 21$  から計算され，パラメータ  $h$  と  $G$  の総数に等しいことを意味する．

$$\begin{aligned} S^1 &= (1, 1, -1, -1, -1, -1)^T \\ S^2 &= (1, -1, 1, -1, 1, 1)^T \\ &\vdots \\ S^{21} &= (1, 1, 1, 1, 1, 1)^T \end{aligned}$$

そして，式 (8.4) のコスト関数モデルを用いて 21 個の設計案のコスト関数値  $\mathcal{H}(S^\mu)$  を計算する．設計案  $S^\mu$  に対応する  $\mathcal{H}(S^\mu)$  の結果は下のようになる．

$$\begin{aligned} \mathcal{H}(S^1) &= -2 \\ \mathcal{H}(S^2) &= 18 \\ &\vdots \\ \mathcal{H}(S^{21}) &= 2 \end{aligned}$$

また，コスト関数値  $\mathcal{H}(S^\mu)$  によってカテゴリ評価値  $K(S^\mu)$  を計算することが  $p = 2$  のとき，式 (8.7)~式 (8.9) の 3 段階カテゴリ関数で実現する．

$$K(S^\mu) = f_1 \left( \frac{1}{N} \mathcal{H}(S^\mu) \right) + f_2 \left( \frac{1}{N} \mathcal{H}(S^\mu) \right) \quad (8.7)$$

$$f_1 \left( \frac{1}{N} \mathcal{H}(S^\mu) \right) = \frac{1}{1 + e^{\gamma_1 \left( \frac{1}{N} \mathcal{H}(S^\mu) - \theta_1 \right)}}, \gamma_1 = 100, \theta_1 = 0 \quad (8.8)$$

$$f_2 \left( \frac{1}{N} \mathcal{H}(S^\mu) \right) = \frac{1}{1 + e^{\gamma_2 \left( \frac{1}{N} \mathcal{H}(S^\mu) - \theta_2 \right)}}, \gamma_2 = 100, \theta_2 = 1 \quad (8.9)$$

$K(S^\mu)$  の結果は下記のようになる．

$$\begin{aligned} K(S^1) &= -1 \\ K(S^2) &= 1 \\ &\vdots \\ K(S^{21}) &= 1 \end{aligned}$$

## 8.2 実験結果

上で得られた 21 個の設計案  $S^\mu$  とそのカテゴリ評価値  $K(S^\mu)$  を用いて，モデルパラメータ  $d$  と  $J$  を推定する．具体的にまず，初期設定として，モデルパラメータの初期分布  $d^0$  と  $J^0$  を正規分布

$N(\mu = 0.0, \sigma^2 = 1.0)$  に従ってランダムに設定する． $\mathbf{d}^0 = (-0.36, -2.24, 1.56, 2.34, -2.09, 0.93)^T$  であり，

$$J_{ij}^0 = \begin{pmatrix} 0 & -1.23 & 2.29 & -0.59 & -0.54 & 0.65 \\ -1.23 & 0 & 0.05 & 0.959 & -0.36 & -0.36 \\ 2.29 & 0.05 & 0 & -0.87 & 1.92 & 1.64 \\ -0.59 & 0.96 & -0.87 & 0 & -1.2 & -0.22 \\ -0.54 & -0.36 & 1.92 & -1.2 & 0 & 0.02 \\ 0.65 & -0.36 & 1.65 & -0.22 & 0.02 & 0 \end{pmatrix}$$

である．

さらに，確率伝搬法によるボルツマンマシンを用いて，上の初期設定  $\mathbf{d}^0$  と  $J^0$  に基づく推定解析を行う．推定回数は  $T=20000000$  とし，推定結果  $\mathbf{d}^T = (1.24, -0.51, 1.96, 2.04, -1.92, 1.63)^T$  であり，

$$J_{ij}^T = \begin{pmatrix} 0 & -6.75 & -0.9 & 1.05 & 5.36 & -0.38 \\ -6.75 & 0 & -5.77 & 4.47 & 2.3 & -5.08 \\ -0.9 & -5.77 & 0 & 1.13 & 6.53 & 1.12 \\ 1.05 & 4.47 & 1.13 & 0 & -7.23 & 1.19 \\ 5.36 & 2.29 & 6.53 & -7.23 & 0 & 5.95 \\ -0.38 & -5.08 & 1.12 & 1.19 & 5.95 & 0 \end{pmatrix}$$

である．推定モデルの決定変数  $S \in \{1, -1\}$  と2次モデルの決定変数  $\mathbf{x} \in \{1, 0\}$  の違いによって，この推定結果  $\mathbf{d}^T$  と  $J^T$  を下式のように2次組合せ最適化モデルのパラメータ  $\mathbf{h}$  と  $G$  に転換する必要がある．

$$h_i = (2d_i - \sum_{j \neq i}^N J_{ij}) \quad (8.10)$$

$$g_{ij} = 4J_{ij} \quad (8.11)$$

結果として，2次組合せ最適化モデルに代入するパラメータは  $\mathbf{h} = (4.11, 9.8, 1.81, 3.46, -16.76, 0.45)^T$  であり，

$$g_{ij} = \begin{pmatrix} 0 & -26.99 & -3.61 & 4.2 & 21.45 & -1.54 \\ -26.99 & 0 & -23.08 & 17.87 & 9.19 & -20.31 \\ -3.62 & -23.08 & 0 & 4.5 & 26.1 & 4.48 \\ 4.2 & 17.87 & 4.5 & 0 & -28.92 & 4.79 \\ 21.45 & 9.19 & 26.1 & -28.92 & 0 & 23.81 \\ -1.54 & -20.31 & 4.48 & 4.79 & 23.81 & 0 \end{pmatrix}$$

である．

この  $\mathbf{h}$  と  $G$  のセットを式 (8.1) の2次モデルに代入し，最適解  $\mathbf{x} = (1, 0, 1, 0, 1, 1)^T$  が得られ，この最適解が実状態と一致することが確認された．この数値例によって，合理的な設計現場の情報を手に入れれば，提案手法で最適設計を再現できることを確認した．

## 第9章 まとめと今後の課題

製品設計では、アイデアの良さに関わらず利益に結びついた製品を作り出すことが重要である。しかし、試作品が利益につながるか否か、あるいは試作品の中のどれが一番大きい利益をもたらすかなどの判断は、製品設計の評価であると呼ばれる。この評価は、中小企業はもちろん大手企業でも、熟練した設計者の経験または暗黙知に依存しているのが一般的である。経験の浅い設計者の場合、あるいは熟練した設計者の知識範囲を超えたノウハウに結びつく新製品を開発するとき、良い設計結果は望まれない。

そのため、暗黙知に頼らずに熟練した設計者だけでなく経験の浅い設計者に対しても普遍的な議論を行え、誰でも扱える数理的な製品設計の評価手法の提案が本学位論文の目的であった。また、従来研究により、製品設計段階においても製品生産段階においても、多角度から製品設計案の評価手法を数理的に議論されているが、これらの評価手法のいずれも製品また機能のインテグラルを取り入れた新製品設計を評価することは充分ではない。本論文では製品アーキテクチャ理論に基づいて、インテグラルを考慮に入れた新製品設計の統一的な見方を提供した。また、この製品設計の評価モデルの数理解析を取り組んだ。具体的には下記の5つの課題である。

第2章の問題定義と定式化では、アーキテクチャ理論に基づき製品を機能デバイスに分解し、機能デバイスをつなげるものをインターフェースとして定義した。各インターフェースのつながりより、またインターフェース自身の単独効用と、本論文で定義したインターフェース同士の相互効用で、総合収益を最大にするインターフェース案（製品設計案）を最適解として選ぶ2次組合せ最適化モデルを提案した。

第3章では最適解析の方法について提案した。第2章で提案した2次組合せ最適化モデルはその非凸性により、現存の整数計画法のいずれも直接にこのモデルを解くことができない。そのため本章ではこの2次モデルにおける非線形項と等価な線形項を導入し、新たな1次等価変換モデルを提案した。さらに、分枝限定法（Branch and Bound）を用いて1次等価モデルを解くことにより、実務規模問題（中型規模）の厳密解を得ることを示した。厳密解法の効果について、製品設計問題の最適解析は実用レベル（16個のデバイス、120個のインターフェースを超えない規模問題）に向上した。

第4章と第5章の大規模問題の近似解析において、それぞれ統計力学における多体システムの状態を確率的に求められる平均場近似（MFA）を援用するアルゴリズムと、モデルの特徴に合わせ、現在の状態（個体）の情報を用いた高速の一点変動近傍法を援用するアルゴリズムを提案した。提案手法の効果について、適用の平均場近似法と一点変動近傍法両方がモデルの計算量は $O(2^N)$ から $O(N^2)$ へ低減することを実現した。数値実験により、2次組合せ最適化モデルの大規

模問題 (機能デバイス数  $M = 100$ , インターフェース数  $N = 4950$ ) を成功に解析した。

第6章のモデルのパラメータ推定において, 製品設計案とそのカテゴリ評価値を観測データとしてモデルのパラメータの推定モデルを提案した。さらに, 確率伝搬法に基づくボルツマンマシンを用いて学習誤差が0の下で汎化誤差を最小にする推定解析を行った。提案手法の効果について, 推定回数の増加につれて推定誤差が安定的に収束する傾向を明らかにした。また均一の大規模の例の推定誤差と的中率の遷移を観察したことによって, 提案手法で大規模問題を扱う能力があることを検証した。

第7章の設計後工程のクラスタリングにおいて, 提案したモデルはデバイスをつなげるインターフェースを決定するものであり, それに基づくデバイスのグループ化 (モジュール化) の研究も必要であると考え。現実的には単一生産拠点での生産能力は限界があり, 全体としてのシステムの効率化を考慮し, 2次モデルで見出した最適な製品設計案を競争的な中間製品へ分割する手法が望ましい。そのため, クラスタリング手法の一つであるPメジアン法を用いて, クラスタ中のインターフェースのつながり情報を最大限にあり, クラスタ間のつながり情報を最小限に抑える最適クラスタ識別モデルを提案した。提案手法の効果について, 従来研究の3つの数値例を用いて数値例のクラスタ結果を再現できることを検証した。

第8章「提案する製品設計アプローチの有効性を検証するための数値例」の部分では, 小規模の実例を用いて, 製品設計の2次組合せ最適化モデルの構築 (第2章の内容), ボルツマンマシンによるモデルパラメータの推定解析 (第6章の内容) と推定したパラメータに基づいて分枝限定法による2次モデルの厳密解析 (第3章の内容) を実施した。結果として最適解とみられる実状態を再現したことが分かる。

第9章はまとめと今後の課題について述べている。今後の課題として, まず, 本論文では単独効用  $h_i$  と相関効用  $g_{ij}$  を与えられたもとの1次等価変換モデルの性能について議論してきたが, 実用に向けて,  $h_i, g_{ij}$  の可能性  $p_i$ , 利益  $E_i$ , コスト  $C_i$  による関係式  $h_i = h_i(p_i, E_i, C_i)$  と  $g_{ij} = g_{ij}(p_i, E_i, C_i; p_j, E_j, C_j)$  を特定する必要もあるだろう。本論文では単独コスト  $h_i$  と相互作用  $g_{ij}$  の値は互いに独立で正規分布に従うとしたが, しかし現実の状況を考えると, 設計案のインターフェースの組に対して, 共有の機能デバイスが存在し, 2つの設計案は異なる分布に従う可能性がある。そのため, この議論に従って提案モデルを改善する必要がある。また, 製品設計の従来研究 QFD, DSM などと比較して本研究の実用性と優位さを検証することも重要である。

本論文では以上の研究内容を詳説した。製品設計の数値評価アプローチを導入する前に, 概念段階の製品設計は, 熟練した設計者の暗黙知あるいは経験に依存した限定な評価しか行えなかったが, 製品アーキテクチャ理論に基づいて提案した製品設計アプローチを導入したことによって, 人のノウハウよりもっと広い範囲の評価が可能になり, 暗黙知に頼らずに熟練した設計者だけでなく経験の浅い設計者に対しても普遍的な議論が実現された。本論文で提案している製品設計アプローチは設計者の意思決定の支援システムの一つとして貢献し, 試作品を利益に結び付ける可能性を高めることが期待できる。

## 関連図書

- [1] 藤本隆宏：「日本のもの造り哲学」，日本経済新聞社 (2004)
- [2] 大富浩一：“製品開発における上流設計の重要性とその方法”，東芝レビュー，Vol.60, No.1 (2005)
- [3] P, Kotler., and K, L, Keller.: Marketing Management, 1st ed., Prentice Hall (1967)
- [4] 及川忍，伊藤誠悟：“自動車部品における製品アーキテクチャの形成プロセスに関する一考察”，日本経営工学会論文誌，Vol.64, No.4 (2014)
- [5] K, Ulrich., and K, Tung.: “Fundamentals of Product Modularity” , Issues Design Manufacturing Integration, Vol.39, pp. 73–80 (1991)
- [6] P, O’Grady.: “The Age of Modularity: Using the New World of Modular Products to Revolutionize Your Corporation” , Adams and Steele (1999)
- [7] D, W, He., and A, Kusiak.: “Design of Assembly Systems for Modular Products” , IEEE Trans. Robot. Automat., Vol.13, pp. 646–655 (1994)
- [8] K, Ulrich., and S, D, Eppinger.: “Product Design and Development” , McGraw-Hill, New York (2011)
- [9] M, D, Guenov., and S, G, Barker.: “Application of Axiomatic Design and Design Structure Matrix to the Decomposition of Engineering Systems” , Systems Engineering, Vol. 8, No. 1, pp. 29–40 (2005)
- [10] P, Lanner., and J, Malmqvist.: “An Approach Towards Considering Technical and Economic Aspects in Product Architecture Design” , *The 2nd WDK-workshop on Product Structuring, TU Delft, The Netherlands* (1996)
- [11] K, Ulrich., and W, P, Seering.: “Function Sharing in Mechanical Design” , *AAAI-88 Proceedings* pp. 342–346 (1998)
- [12] K, B, Clark., and T, Fujimoto.: “The Power of Product Integrity” , Harvard Business Review, November-December, pp. 107–118 (1990)
- [13] 澤口学，大野高裕：“価値ある製品設計案選択に有効な意思決定モデル”，日本経営工学会誌，Vol.51, No.3, pp.279–292 (2000)

- [14] 水野滋, 赤尾洋二: 「品質機能展開」, 日科技連 (1978)
- [15] G, H, Lee., and A, Kusiak.: “The House of Quality for Design Rule Priority” , The International Journal of Advanced Manufacturing Technology, Vol.17, pp.288–296 (2001)
- [16] T, U, Pimmler and S, D, Eppinger.: “Integration Analysis of Product Decomposition” , *ASME Design Theory and Methodology Conference* (1994)
- [17] R. E. Thebeau: “Knowledge Management of System Interfaces and Interactions for Product Development Processes” , *Master 's Thesis. MIT.* (2001)
- [18] C, I, G, Fernandez: “Integration Analysis of Product Architecture to Support Effective Team Co-location” , *Master 's Thesis. MIT.* (1998)
- [19] W, J, Fabrycky., and B, S, Blanchard.: “Life Cycle Cost And Economic Analysis” , Prentice-Hall, Englewood Cliffs, NJ (1991)
- [20] I, Kaku., D, X, Jiang., T, Shinzato., M, Hoshino., Y, Yin., and Y, Yu.: “Definition, Model and Algorithm of New Product Design Considering Life Cycle and Architectures” , *International Symposium on Reliability Engineering and Risk Management(ISRERM)* (2012)
- [21] 人見勝人: 「生産システム工学第5版」, 共立出版 (2013)
- [22] A, Shahin., and N, Janatyan.: “Group Technology (GT) and Lean Production: A Conceptual Model for Enhancing Productivity” , *Journal of International Business Research*, Vol.3, No.4, pp.105–118 (2010)
- [23] A, Kusiak., and W, S, Chow.: “Efficient Solving of the Group Technology Problem” , *Journal of Manufacturing Systems*, Vol.6, No.2, pp. 117–124 (1987)
- [24] A, Zakarian., and G, J, Rushton.: “Development of Modular Electrical Systems” , *IEEE/ASME Transaction on Mechatronics*, Vol.6, No.4, pp. 507–520 (2001)
- [25] Y, B, Cho., Y, H, Cho., and S, H, Kim.: “Mining Changes in Customer Buying Behavior for Collaborative Recommendations” , *Expert Systems with Applications* 28, pp. 359–369 (2005)
- [26] A, Kusiak.: “The Part Families Problem in Flexible Manufacturing Systems” , *Annals of Operations Research* 3, pp. 279–300 (1985)
- [27] K, Ulrich.: “The Role of Product Architecture in the Manufacturing Firm” , *Research Policy*, Vol.24, pp.419–440 (1995)
- [28] C, C, Huang.: “Overview of Modular Product Development” , *Proc. Natl. Sci. Council. ROC(A)*, Vol.24, No.3, pp.149–165 (1999)

- [29] Z, Song., and A, Kusiak.: “Optimizing Product Configurations with a Data-Mining Approach”, *International Journal of Production Research*, Vol.47, No.7, pp.1733–1751 (2009)
- [30] K, Ulrich., and S, Pearson.: “Assessing the Importance of Design Through Product Archaeology”, *Management Science*, Vol.44, No.3, pp.352–369 (1998)
- [31] D, Robertson., and K, Ulrich.: Platform Product development. *Sloan Management Review* (1998)
- [32] Y, Yin.: “Product Architecture Design for Global Performance”, *Proceedings of the 51st Annual Meeting of the ISSS* (2007)
- [33] 目代武史: “製品アーキテクチャの変化と組織の動的対応 自動車部品の事例分析”, *日本経営工学会第77回大会報告要旨集*, pp.51–54 (2003)
- [34] D, M, Sharman., A A, Yassine., and P, Carlile.: “Characterizing Modular Architectures”, *ASME Design Theory and Methodology Conference* (2002)
- [35] 佐伯靖雄: “複合要素技術型製品の開発における製品と組織のアーキテクチャ分析”, *日本経営学会誌*, Vol.23, No.5, pp.25–36 (2009)
- [36] I, Kaku., D,X, Jiang., and T, Shinzato.: “A New Product Design Approach through Integral Architecture”, *Innovative Computing, Information and Control, Express Letters, An International Journal of Research and Surveys*, Vol. 5, No.5, pp. 1709–1716 (2011)
- [37] D,X, Jiang., T, Shinzato., I, Kaku., and Y, Yin.: “A Mathematical Model of Product Design through Integral Architecture”, *Proceedings of the 21th International Conference on Production Research* (2011)
- [38] 今野浩: 「線形計画法」, *日科技連* (1999)
- [39] 姜東暁, 新里隆, 星野満博, 張人千, 郭偉宏: “分枝限定法による2次新製品設計モデルとその1次等価変換の最適化解析”, *日本経営工学会誌*, 掲載決定, Vol.65, No.1 (2014)
- [40] P,J, Kolesar.: “A Branch and Bound Algorithm for the Knapsack Problem”, *Management Science*, Vol.13, No. 9, Series A, Sciences, pp. 723–735 (1967)
- [41] A, Caprara., D, Pisinger., and P, Toth.: “Exact Solution of the Quadratic Knapsack Problem”, *INFORMS Journal on Computing*, Vol.11, No.2, pp.125–137 (1999)
- [42] C, Helmberg., F, Rendl., and R, Weismantel.: “Quadratic Knapsack Relaxations Using Cutting Planes and Semidefinite Programming,” *Proceedings of the Fifth IPCO Conference, Lecture Notes in Computer Science*1084, 175–189, Springer Verlag (1996)

- [43] M, Bellare., O, Goldreich., and M, Sudan.: “PCPs and Non-Approximability-Towards Tight Results,” *Proceedings of the 36th FOCS Conference* IEEE Computer Society Press Springer Verlag.422–431(1995)
- [44] D,X, Jiang., T, Shinzato., H, Sugiyama., M, Hoshino., Y, Yu., and I, Kaku.: “A One-Bit-Variant Neighborhood Algorithm for Solving the Combinatorial Product Design Model,” *Proceedings of the 11th International Conference on Industrial Management*, pp.27–31 (2012)
- [45] T, Shinzato.: “Box Muller Method,” unpublished note, <http://www.akita-pu.ac.jp/system/mse/ie/shinzato/boxmuller.pdf> (2013)
- [46] 田中利幸：“平均場近似と最尤推定”，電子情報通信学会技術研究報告. NC, ニューロコンピューティング 98(674), 155–162 (1999)
- [47] R, J, Baxter.: “Exactly Solved Models in Statistical Mechanics” , Dover Publications (2008)
- [48] M, Opper., and D, Saad.: “Advanced Mean Field Methods: Theory and Practice” , MIT Press (2001)
- [49] H, Nishimori.: “Statistical Physics of Spin Glasses and Information Processing” , Oxford University Press (2001)
- [50] D, H, Ackley., G, E, Hinton., and T, J, Sejnowski.: “A Learning Algorithm for Boltzmann Machines”, *Cognitive Science*, Vol.9, pp.147–169 (1985)
- [51] M, Opper., and O, Winther.: “Tractable Approximations for Probabilistic Models: The Adaptive Thouless-Anderson-Palmer Mean Field Approach”, *Cognitive Science*, Vol.86, No. 17, pp.3695–3699 (2001)
- [52] M, Opper., and O, Winther.: “Adaptive and Self-averaging Thouless-Anderson-Palmer Mean-Field Theory for Probabilistic Modeling”, *Physical Review E*, Vol.64, pp.1–13 (2001)
- [53] 丹内隼也：“確率伝搬法を用いた統計的学習理論に関する研究”，東北大学大学院情報科学研究科応用情報科学専攻修士学位論文，(2009)
- [54] 渡辺透夫:「データ学習アルゴリズム」，共立出版 (2006)
- [55] P, Hansen., N, Mladenovic., and J.A.M, Perez.: “Variable neighbourhood search: methods and applications” , *Annals of Operations Research* 175, 367–407 (2010)
- [56] N, Mladenovic., and P, Hansen.: “Variable neighborhood search” , *Computers and Operations Research* 24 (11), 1097–1100 (1997)
- [57] 馬場則夫，小島史男，小澤誠一:「ニューラルネットの基礎と応用」，共立出版株式会社 (1994)



- [58] 伊庭幸人, 種村正美, 大森裕浩, 和合肇, 佐藤整尚, 高橋明彦: 「計算統計 マルコフ連鎖モンテカルロ法とその周辺」, 岩波書店 (2005)
- [59] J, Reese. : “Methods for Solving the  $p$ -Median Problem: An Annotated Bibliography” , Mathematics Faculty Research, paper 28, 2005.



# 謝辞

博士課程3年間の研究の集大成として卒論を仕上げました。本研究は研究計画の段階から検証、論文の作成に至るまで、先生方と友人から貴重な助言をいただきました。三年間星野満博准教授のゼミで発表した内容が、本論文の骨子となっています。三年間週に一回ゼミの際にいろいろご指導して下さった星野満博准教授、新里隆助教に心から感謝いたします。そして毎週研究内容について新里隆助教と打ち合わせをし続けて、中間発表や本審査の際にも新里隆助教には貴重な御指導をいただきました。そして中間発表の時に主査の三品勉教授と副査の木村寛准教授からいろいろとアドバイスをいただき、研究の視野が広がりました。上の先生方に深く感謝いたします。論文作成の段階に入ってから、外来の郭偉宏教授(東京都市大学)に懇親丁寧な御指導いただき、そのたびに的確なアドバイスを頂戴しました。きちんとした日本語での論文を書くことができている私が、本論文を論文の体裁に整えることができたのは、ひとえに新里隆助教の御指導があったこそであります。実をいえば指導を受けた瞬間には些末なことを指摘されたように感じたこともありました。ところが、それをいわれたとおりに書き直してみると、文章が整うだけでなく、自分の考えまでも明確に整理されていくのを実感したことが一度ならずありました。

末尾になりましたが、論文執筆の過程では、他にもたくさんの方のお世話になりました。同期で留学生であった私を特別扱いせず受け入れてくださった同級生や、諸先輩、後輩たち、援助までしてくださった家族に感謝の意を捧げたいと思います。そして在学中に多くのことを学ばせていただいた本荘公民館の日本語教室の先生、及び国際教養大学の相場先生にお礼を申し上げます。どうもありがとうございました。



## 業績

- [1] 姜東暁, 新里隆, 星野満博, 張人千, 郭偉宏: “分枝限定法による2次新製品設計モデルとその1次等価変換の最適化解析”, 日本経営工学会誌, 掲載決定, Vol.65, No.1, (2014)
- [2] Ikou Kaku, Dongxiao Jiang and Takashi Shinzato.: “A New Product Design Approach through Integral Architecture,” *Innovative Computing, Information and Control, Express Letters, An International Journal of Research and Surveys*, Vol. 5, No.5, pp. 1709-1716 (2011)

### 国際発表

- [1] Dongxiao Jiang, Takashi Shinzato, Mitsuhiro Hoshino, Ikou Kaku.: “Boltzmann Machine Approach with Mean Field Approximation for Solving Estimation of Product Design Parameters,” *Proceedings of the 2013 Asian Conference of Management Science and Applications*, 8 page (2013)
- [2] Ikou Kaku, Dongxiao Jiang, Renqian Zhang and Y. Yin.: “How to Solve the New Product Design Model Considered life-cycle and Product Architectures,” *Proceedings of the 11th Global Conference on Sustainable Manufacturing*, 6 page (2013)
- [3] Dongxiao Jiang, Shinzato, T., Sugiyama, H., Hoshino, M., Yu, Y. and Kaku, I.: “A One-Bit-Variant Neighborhood Algorithm for Solving the Combinatorial Product Design Model,” *Proceedings of the 11th International Conference on Industrial Management*, pp.27–31 (2012)
- [4] Ikou Kaku, Dongxiao Jiang, Takashi Shinzato, Mitsuhiro Hoshino, Y. Yin and Yang Yu.: “Definition, Model and Algorithm of New Product Design Considering Life Cycle and Architectures,” *Proceedings of the International Symposium on Reliability Engineering and Risk Management*, 6 page (2012)
- [5] Dongxiao Jiang, Shinzato, T., Kaku, I. and Yin, Y.: “A Mathematical Model of Product Design through Integral Architecture,” *Proceedings of the 21th International Conference on Production Research*, No. 30, 8 page.(2011)

It is selected as Young Scientists' Award .

## 国内発表

- [1] 姜東暁, 新里隆, 星野満博, 郭偉宏: “ボルツマンマシンによる新製品設計モデルのパラメータ推定”, 日本経営工学会平成 25 年度秋季研究大会予稿集, pp. 126-127(2013/11)
- [2] 姜東暁, 新里隆, 星野満博, 張人千, 郭偉宏: “1 次等価変換による 2 次新製品設計モデルの厳密解析”, 日本経営工学会平成 25 年度春季研究大会予稿集, pp. 126-127, (2013/05)
- [3] 姜東暁, 新里隆, 杉山博史, 星野満博, 郭偉宏: “製品設計の組合せ最適化モデルに適用する 1 点変動近傍法の提案”, 日本経営工学会平成 24 年度春季研究大会予稿集, pp. 78-79(2012/05)
- [4] 新里隆, 姜東暁, 星野満博, 嶋崎善章, 郭偉宏: “施設配置問題に対する平均場近似解析”, 日本経営工学会平成 24 年度春季研究大会予稿集, pp. 76-77(2012/05)
- [5] 姜東暁, 新里隆, 杉山博史, 郭偉宏: “中間製品を考慮に入れた最適製品設計のアーキテクチャ理論”, 日本経営工学会平成 23 年度秋季研究大会予稿集, pp. 232-233(2011/11)
- [6] 姜東暁, 新里隆, 杉山博史, 郭偉宏: “インテグラル概念を取り入れた製品設計の数理解析”, 日本経営工学会平成 23 年度春季研究大会予稿集, pp. 168-169(2011/05)
- [7] 姜東暁, 新里隆, 杉山博史, 郭偉宏: “アーキテクチャ理論に基づく製品設計戦略に関する研究”, 平成 22 年度日本経営工学会東北支部卒業研究発表会 (2011/03)