



単目的問題への分割に基づく多目的分枝限定法の提案とその応用

メタデータ	言語: Japanese 出版者: 公開日: 2025-06-11 キーワード (Ja): キーワード (En): 作成者: 下保, 知輝 メールアドレス: 所属:
URL	https://doi.org/10.15118/0002000336

博士論文

単目的問題への分割に基づく
多目的分枝限定法とその応用
**Multi-objective Branch-and-Bound Method
Based on Decomposition and Its Applications**

2021

下保知輝

Tomoki Kaho

2025 年 3 月

室蘭工業大学大学院 工学研究科 工学専攻 博士論文

指導教員 渡邊 真也 教授

計算知能研究室

目次

第 1 章	序論	1
1.1	本論文の背景と目的	1
1.2	本論文の構成	2
第 2 章	最適化	3
2.1	最適化とは	3
2.1.1	目的関数	4
2.1.2	制約条件	4
2.1.3	設計変数	5
2.1.4	最適化問題の分類	6
2.1.5	本稿における最適化問題	7
2.1.6	最適化手法	8
2.2	単目的最適化	8
2.2.1	はじめに	8
2.2.2	シンプレックス法	9
2.2.3	切除平面法	13
2.2.4	分枝限定法	18
2.3	多目的最適化	22
2.3.1	はじめに	22
2.3.2	本稿が対象とする問題設定	23
2.3.3	多目的最適化手法の分類	24
2.3.4	多目的最適化問題の分割（スカラー化）	25
2.3.5	多目的分枝限定法	28
第 3 章	Multi-Objective Branch-and-Bound based on Decomposition (MOBB/D)	32
3.1	はじめに	32
3.2	提案手法の設計	33
3.2.1	重みベクトルの生成	34
3.2.2	スカラー化関数の選択	34

3.2.3	探索情報の再利用	35
3.3	最適化の流れ	36
3.4	探索情報の例: シンプレックス表の再利用	37
3.4.1	概要	37
3.4.2	数値実験	39
3.4.3	まとめ	47
第 4 章	室蘭市のごみ収集問題	48
4.1	はじめに	48
4.2	経路計画問題	49
4.2.1	巡回セールスマン問題 (TSP)	50
4.2.2	配送経路問題 (VRP)	51
4.2.3	ごみ収集問題との関連性	52
4.3	単目的最適化としてのモデル化	52
4.3.1	モデル定式化	52
4.3.2	分枝カット法による解法	54
4.3.3	数値実験	55
4.4	多目的最適化としてのモデル化	59
4.4.1	多目的モデル化	59
4.4.2	切除平面共有の仕組み	62
4.4.3	数値実験	63
4.5	まとめと今後の課題	66
第 5 章	結論	67
5.1	本論文の成果	67
5.2	今後の課題	68
	謝辞	70
	参考文献	71
	付録	74
A	研究業績	74

目次

2.1	Example of a problem with only linear constraints	5
2.2	Example of a problem with nonlinear constraints	5
2.3	Example of a problem with continuous variables	6
2.4	Example of a problem with discrete variables	6
2.5	A conceptual diagram of the Simplex method	9
2.6	Feasible region of (IP)	14
2.7	Feasible region of (LP)	14
2.8	The concept of cutting plane method	16
2.9	The concept of branch-and-bound	19
2.10	The concept of dominance	23
2.11	Contours of scalarizing functions	28
2.12	Example of a situation where bounding can be performed in MOBB . .	30
2.13	Example of a situation where bounding cannot be performed in MOBB .	30
3.1	A conceptual diagram of MOBB/D.	33
3.2	Transition of hypervolume values at each timestamp for a small-scale problem.	43
3.3	Transition of hypervolume values at each timestamp for a large-scale problem.	43
4.1	Plot of Garbage Stations in Muroran City	48
4.2	Route of Actual GPS Data	56
4.3	Optimized Route by Proposed Model	56
4.4	Route Length Transition with Garbage Amount Variation	57
4.5	Target area of experiment for Multi-Objective Garbage Collection Problem	63
4.6	Pareto solutions for Multi-Objective Garbage Collection Problem	64
4.7	Route and assignment for residents ($\theta = 0.9$)	65
4.8	Route and assignment for workers ($\theta = 0.1$)	65

表目次

3.1	The result of execution times (millisecond) in each problem set.	41
3.2	The result of ratio of hypervolume values in each problem set.	42
3.3	An example of calculating the "ratio" of hypervolume value ($k = 2, m = 60$).	42
3.4	The result of average of number of update simplex table in each number of objectives.	45
3.5	The result of execution times (millisecond) in each problem set.	46
4.1	Comparison of execution time with/without cut sharing	64

第 1 章

序論

1.1 本論文の背景と目的

現代の多くの分野において、効率的な意思決定や資源配分を実現するために最適化問題が重要な役割を果たしている。最適化問題は工学、経済、物流、医療など幅広い分野で応用され、その影響力は年々増大している。特に、複数の評価基準が競合する状況下では、多目的最適化が不可欠である。たとえば、コストと品質、効率と環境負荷といったトレードオフが存在する問題において、単一の評価基準に基づく解法では十分な解を得ることが難しい。

多目的最適化では、一般に「パレート最適解」と呼ばれる解集合を得ることを目指す。パレート最適解集合は、各基準間のトレードオフを反映し、意思決定者に多様な選択肢を提供する。しかし、このような解集合を効率よく求めることは計算的に非常に困難である。特に、意思決定に整数制約が絡む多目的混合整数線形計画問題（Multi-Objective Mixed Integer Linear Programming; MOMILP）は、解の探索空間が膨大であり、その解法の開発は実世界の複雑な意思決定において極めて重要な課題となっている。

従来、MOMILP に対するアプローチとして、進化計算手法や多目的分枝限定法が用いられてきた。進化計算手法は計算効率が高く、高次元の問題にも対応可能であるが、厳密な最適解を保証することができない。一方、多目的分枝限定法は厳密解を得ることが可能であるが、計算コストが非常に高いという課題がある。これらを踏まえ、本研究では、MOMILP に対する新たな実用的厳密解法である Multi-Objective Branch-and-Bound based on Decomposition (MOBB/D) を提案し、その有用性を検証することを目的とする。

本研究では、多目的混合整数線形計画問題（MOMILP）に対して、分解ベースの手法と分枝限定法を統合した新しい解法である MOBB/D を提案する。提案手法は、分解ベースの進化計算手法の代表例である MOEA/D（Multi-Objective Evolutionary Algorithm based on Decomposition）に着想を得ており、MOMILP を複数の単目的部分問題に分解し、各

部分問題を厳密に解くことで、全体としてパレート最適解集合を構築する。

MOBB/D の特徴の 1 つは、各部分問題の最適化中に得られる探索情報を効果的に再利用する点である。具体的には、シンプレックス表や切除平面といった静的および動的な探索情報を保持し、これを他の部分問題の最適化に活用することで、探索効率を大幅に向上させることを目指す。これにより、従来の分枝限定法が抱える計算コストの高さを軽減しつつ、厳密解を得るという二重の目標を達成する。

本研究の目的は以下の 3 点に要約される。第一に、分解ベースの手法と分枝限定法を統合した新しい厳密解法を提案し、その探索効率と汎用性を高めることである。第二に、提案手法の有効性を、既存手法との比較実験や数値シミュレーションを通じて検証する。第三に、実問題への適用を通じて、MOMILP に対する提案手法の実用性を示すことである。本研究では、実問題として室蘭市のごみ収集問題を対象とし、EBPM (Evidence-Based Policy Making) のためにも活用できる手法の開発を目指す。

1.2 本論文の構成

本論文の構成を以下に示す。本論文は全 5 章から構成されている。

第 1 章では、本研究の背景、従来手法の課題、および提案手法の概要と研究目的について述べている。

第 2 章では、最適化問題の基礎を整理する。単目的最適化および多目的最適化の定式化とその解法について説明する。単目的最適化手法としては、提案手法とかわりか深いシンプレックス法、切除平面法、分枝限定法について述べる。多目的最適化手法としては、MOEA/D に代表される分割ベースの枠組みと、既存手法である多目的分枝限定法について述べる。

第 3 章では、提案手法 MOBB/D の詳細について述べる。具体的には、分解ベースのアプローチと分枝限定法の統合に関する手法設計、各部分問題における探索情報の再利用戦略、および全体の最適化プロセスを説明する。また、代表的な探索情報の活用例として、シンプレックス表の再利用を行う方法論および数値実験の結果を示す。

第 4 章では、提案手法を実問題に適用した結果を示す。室蘭市のごみ収集問題を対象とし、問題の概要および EBPM の重要性について述べ、この問題を単目的最適化問題としてモデル化したケースと、多目的最適化問題としてモデル化したケースについてそれぞれ説明する。

最後に、第 5 章では、本研究の成果をまとめるとともに、今後の課題について議論する。

第 2 章

最適化

2.1 最適化とは

最適化 (Optimization) とは、数学および応用科学の分野において、与えられた制約条件の下で、目的関数の値を最大化または最小化する解を探索することを指す。たとえば、ある製品の製造コストを最小化するために、原材料の配合比率を決定する問題や、ある製品の性能を最大化するために、製造工程のパラメーターを決定する問題などが挙げられる。

最適化問題は以下の基本要素から構成される。

- 目的関数 (Objective Function) : 最適化の対象となる指標を表す関数。
- 制約条件 (Constraint) : 設計変数が満たすべき等式や不等式。
- 設計変数 (Design Variable) : 最適化を行う対象を特徴づける未知変数。

たとえば、ある設計変数 $\boldsymbol{x} \in S$ に対して、制約条件 $g(\boldsymbol{x}) \leq 0$ を満たしながら目的関数 $f(\boldsymbol{x})$ を最小化する問題を考える。この場合、最適化問題は式 (2.1) のように表現される。

$$\begin{aligned} \min. \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}) \leq 0 \\ & \boldsymbol{x} \in S \end{aligned} \tag{2.1}$$

ここで、設計変数に具体的な値を代入したものを**解 (Solution)** と呼び、制約条件を満たす設計変数を**実行可能解 (Feasible Solution)**、実行可能解全体からなる集合を**実行可能領域 (Feasible Region)** と呼ぶ。実行可能解のなかで、目的関数の値が最小となる実行可能解を**最適解 (Optimal Solution)**、その時の目的関数値を**最適値 (Optimal Value)** と呼ぶ。

以下で最適化における基本要素の分類を記述する。

2.1.1 目的関数

目的関数は、最適化において最小化または最大化を目指すべき指標である。工学においては、製造コストやエネルギー消費量などの最小化、製品の性能や効率などの最大化が目的関数として設定されることが多い。

目的関数は主に以下の要素によって分類される。

- 目的関数の数: 単目的か多目的か。
- 線形性: 線形か非線形か。

特に、目的関数の数によって最適化の困難性が大きく異なる。単目的最適化では、目的関数の値の大小によって解の優劣が一意に定まる。一方、多目的最適化では、目的関数が 2 つ以上あるため、解の優劣が一意に定まらない。そのため、多目的最適化問題では、解の優劣を評価するための基準が必要となる。したがって、一般に多目的最適化問題は単目的最適化問題よりも困難である。詳細は 2.2 節と 2.3 節を参照されたい。

線形性について、線形関数は変数の 1 次式で表現される関数であり、非線形関数には 2 次以上の項や指数関数などを含む関数である（詳細は次節の 2.2 節を参照）。

2.1.2 制約条件

制約条件は、設計変数が満たすべき等式や不等式である。工学においては、使える資源の上限や解が満たすべき条件などを制約条件として記述する。

制約条件は主に以下の 2 つに分類される。

- 線形制約 (Linear Constraint) : 制約条件が線形の場合。
- 非線形制約 (Nonlinear Constraint) : 制約条件が非線形の場合。

線形制約の場合、特に制約条件は式 (2.2) のように表現される。

$$Ax \geq b \quad (2.2)$$

ここで、 A は係数行列、 x は設計変数ベクトル、 b は定数ベクトルである。線形制約は、設計変数空間を超平面 (hyperplane) によって区切るため、実行可能領域が凸集合となり比較的扱いやすい。

一方、非線形制約の場合は、制約条件が非線形関数によって表現される。非線形関数は一般に解析的に解くことが困難であるため、非線形最適化問題は解析的な解法が適用できない場合が多い。そのため、非線形最適化問題の解法は、数値的なアプローチが主流となる。したがって、一般に非線形最適化問題は線形最適化問題よりも困難である。

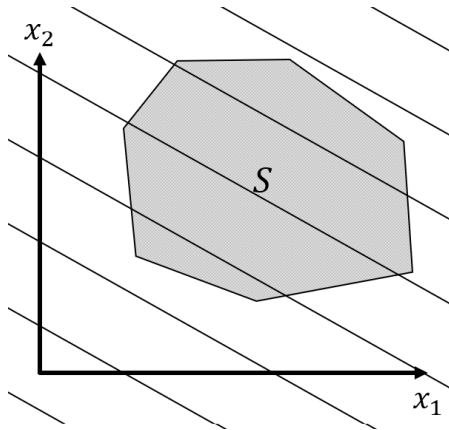


Fig. 2.1 Example of a problem with only linear constraints

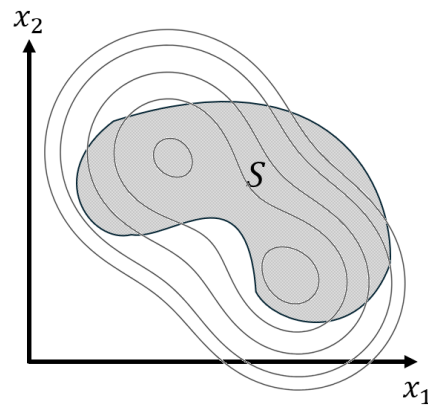


Fig. 2.2 Example of a problem with nonlinear constraints

Fig. 2.1, 2.2 に、線形制約のみを持つ問題と非線形制約を持つ問題それぞれの実行可能領域 S および目的関数の等高線をプロットした例を示す。Fig. 2.1 は線形制約のみを持つ問題の一例である。実行可能領域 S は線分で構成された凸多面体となっており、等高線は直線で等間隔になっている。一方、Fig. 2.2 は非線形制約を持つ問題の一例である。実行可能領域 S は非凸集合となっており、等高線は曲線で間隔が不均等になっている。

2.1.3 設計変数

設計変数は主に以下の 2 つに分類される。

- 連続変数 (Continuous Variable) : 設計変数が連続値を取る場合。
- 離散変数 (Discrete Variable) : 設計変数が離散値を取る場合。

なおこの分類は、設計変数に対し整数性という制約条件を設けるか否かによるものであり、制約条件分類としても考えることができる。しかし、一般に整数性は設計変数の取りうる値自体の制約を表すため、設計変数の分類として記述することが多い。

連続変数の場合、設計変数空間は連続的である。そのため、ある解に対し近傍の目的関数の値に基づいて解を探索するといったアプローチが容易である。

一方、離散変数の場合、設計変数空間は離散的である。そのため、解の探索は網羅的に行う必要があり、解の探索空間が大きい場合、探索が困難となる。したがって、一般に離散変数問題は連続変数問題よりも困難である。そこで、離散最適化問題に対しては、主に整数制約を緩和（削除）することで連続最適化問題として解き、その結果に基づいて整数制約を満たす解を探索するアプローチを取ることが多い。特に、(混合) 整数計画問題に対し整数制約を取り除いて線形計画問題へと変換することを**線形緩和**と呼ぶ。

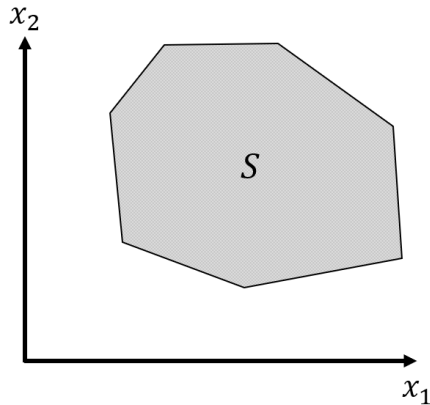


Fig. 2.3 Example of a problem with continuous variables

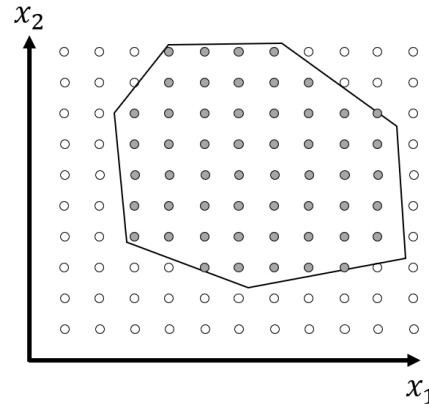


Fig. 2.4 Example of a problem with discrete variables

離散変数を導入することにより，探索空間は連続空間から離散的な構造へと変化する．この変化は，計算負荷の増加を招くが，実問題における制約を表現するために有用である．

Fig. 2.3, 2.4 に，連続変数からなる問題と離散変数からなる問題それぞれの実行可能領域のプロットの一例を示す．Fig. 2.3 は連続変数からなる問題の一例である．制約条件を表す境界上にも実行可能解が存在し，特に線形最適化においては実行可能領域が成す凸多面体の頂点が最適解となる．一方，Fig. 2.4 は離散変数からなる問題の一例である．制約条件によって定まる領域の内部に離散的に存在する解から最適解を探索する必要があり，探索が困難である．

2.1.4 最適化問題の分類

以上の基本要素の特性に基づいて，最適化問題は以下をはじめとしたさまざまな分類が存在する．なお，本稿で主に扱う項目は太字で示している．

- **線形計画問題 (Linear Programming; LP)** : 目的関数と制約条件が線形の場合．
- **整数計画問題 (Integer Programming; IP)** : 設計変数が整数値を取る場合．
- **混合整数計画問題 (Mixed Integer Programming; MIP)** : 設計変数が連続値と整数値を取る場合．
- 非線形計画問題 (Nonlinear Programming; NLP) : 目的関数または制約条件が非線形の場合．
- 二次計画問題 (Quadratic Programming; QP) : 目的関数が二次関数で表現される場合．
- 凸計画問題 (Convex Programming) : 目的関数が凸関数で表現される場合．
- 単目的最適化問題 (Single-Objective Optimization Problem; SOP) : 目的関数が 1

つの場合.

- 多目的最適化問題 (Multi-Objective Optimization Problem; MOP) : 目的関数が 2 つ以上ある場合 (主に制約なしを指すことが多い).
- 制約付き多目的最適化問題 (Constrained Multi-Objective Optimization Problem; CMOP) : 制約条件があることを明示した多目的最適化問題.
- 多目的線形計画問題 (Multi-Objective Linear Programming; MOLP) : 目的関数と制約条件が線形の多目的最適化問題.
- **多目的混合整数線形計画問題 (Multi-Objective Mixed Integer Linear Programming; MOMILP)** : 目的関数と制約条件が線形の多目的最適化問題で, 設計変数が連続値と整数値を取る場合.

離散最適化問題のなかで, 特に設計変数が要素の組み合わせの形で表現される問題を**組合せ最適化問題 (Combinatorial Optimization Problem)**と呼び, 実問題においても多く見られる. 以下に, 組合せ最適化問題の例を挙げる.

- 巡回セールスマン問題 (Traveling Salesman Problem; TSP) : 与えられた都市をすべて巡回するルートの総移動距離を最小化する問題.
- ナップサック問題 (Knapsack Problem) : 与えられた容量を超えないように, 価値の合計を最大化するアイテムの選択問題.
- グラフ彩色問題 (Graph Coloring Problem) : グラフの頂点に色を割り当てる問題.
- 最大流問題 (Maximum Flow Problem) : グラフの頂点間の最大流量を求める問題.
- 割当問題 (Assignment Problem) : 与えられたタスクを最小のコストで担当者に割り当てる問題.

2.1.5 本稿における最適化問題

本稿では, 目的関数および制約条件はいずれも線形の問題を取り扱う. これは以下の理由によるものである.

線形問題の数学的特性

線形問題では, 実行可能領域が凸集合となるため, 最適解が存在する場合, 必ず実行可能領域が成す超多面体 (polytope) の頂点に存在する. したがって, 線形問題は最適解を求めるためのアルゴリズムが効率的に設計できる.

実問題への適用性

線形問題は, 実問題においても物流最適化, タスク割り当て問題, ポートフォリオ問題, 交通計画問題など, 幅広い分野で適用することができる.

2.1.6 最適化手法

最適化問題を解くための手法は、目的関数や制約条件の特性によって異なる。最適化手法を大別すると、以下の 2 つに分類される。

- 厳密解法 (Exact Method) : 最適解を厳密に求める手法。
- ヒューリスティック (発見的) 解法 (Heuristic Method) : 近似解を効率的に求める手法。

厳密解法は、探索空間を網羅的に探索することや、最適性の理論的保証を行うことで最適解を求める手法であり、主に線形計画問題や整数計画問題に適用される。一方、ヒューリスティック解法は、問題の構造や経験則に基づいて近似解を効率的に求める手法であり、主に非線形計画問題や多目的最適化問題に適用される。

本稿では、厳密性に焦点を当て、線形計画問題や混合整数計画問題、多目的混合整数線形計画問題に対する最適化手法について説明する。

本章の以下の節では、単目的最適化問題、多目的最適化問題それぞれに対する最適化手法について説明する。

2.2 単目的最適化

2.2.1 はじめに

単目的最適化は、単一の目的関数を対象とした最適化問題である。複数の目的を同時に扱う必要がある多目的最適化とは異なり、解の評価と比較が明確である。そのため、単目的最適化問題は多目的最適化問題よりも比較的扱いやすい。

また、本稿では線形性を持つ問題を取り扱うため、単目的最適化問題においても目的関数および制約条件は線形である。そのため、特に線形性を明示するために式 (2.3) のように行列表記で問題を記述することが多い。

$$\begin{aligned} \min. \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \in S \end{aligned} \tag{2.3}$$

ここで、 $\mathbf{A}$ を係数行列、 $\mathbf{b}$ を定数ベクトル、 $\mathbf{c}$ を目的関数の係数ベクトルと呼ぶ。変数の取りうる値の集合 S は、主に実数集合 $\mathbb{R}$ や整数集合 $\mathbb{Z}$ 、その両者を変数ごとに指定することなどで表現される。

設計変数がとりうる値の種類によって、線形の単目的問題は主に以下の 3 つに分類される。

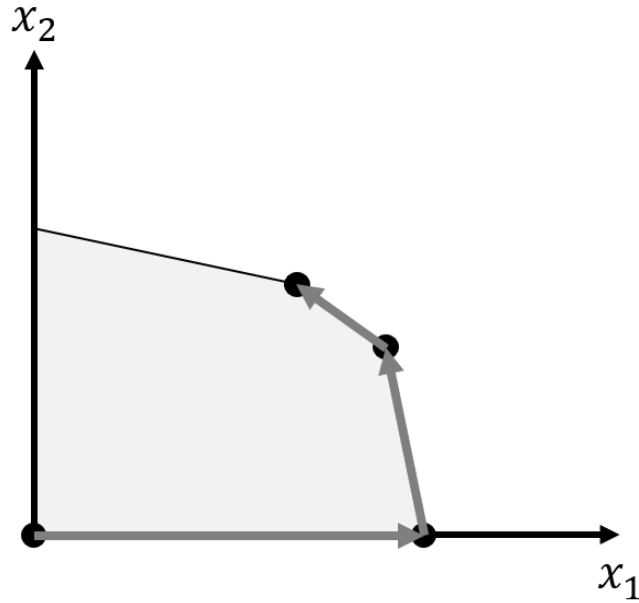


Fig. 2.5 A conceptual diagram of the Simplex method

- 線形計画問題 (Linear Programming; LP) : 設計変数が実数値を取る場合.
- 整数計画問題 (Integer Linear Programming; ILP) : 設計変数が整数値を取る場合.
- 混合整数計画問題 (Mixed Integer Linear Programming; MILP) : 設計変数が連続値と整数値を取る場合.

以下では、それぞれの問題に対する代表手法である、シンプレックス法、切除平面法、分枝限定法について説明する.

2.2.2 シンプレックス法

はじめに

シンプレックス法 (Simplex method, 単体法) は, 1947 年に Dantzig によって提案された線形計画問題を解くための効率的な反復法であり, 最適化理論の中核をなすアルゴリズムの 1 つである⁽¹⁾. Fig. 2.5 に示すように, シンプレックス法は, 実行可能解を表す凸多面体上の頂点を辿りながら目的関数の値を逐次的に改善する.

なお, 以下の説明では, 最大化問題 (LP) を取り扱う. 最小化問題を解く場合は目的関数の符号を反転させることで最大化問題に帰着できる.

$$\begin{aligned}
 (\text{LP}) \quad & \max. \quad \mathbf{c}^\top \mathbf{x} \\
 & \text{s.t.} \quad \mathbf{Ax} = \mathbf{b} \\
 & \quad \mathbf{x} \geq \mathbf{0}
 \end{aligned} \tag{2.4}$$

基底解

(LP)において、制約条件の数が m 個、設計変数の数が n 個であるとする。このとき、係数行列 A は $m \times n$ 行列、定数ベクトル b は m 次元ベクトル、目的関数の係数ベクトル c は n 次元ベクトルである。そして、簡単のため以降では係数行列 A は $\text{rank } A = m$ を満たすという仮定をおく。この仮定により、 A は線形独立となり基底ベクトル $a_{i_1}, a_{i_2}, \dots, a_{i_m}$ を取ることができる。

基底として選んだ添え字番号集合を $I_B = \{i_1, i_2, \dots, i_m\}$ 、選ばれなかった添え字番号集合を $I_N = \{i_{m+1}, i_{m+2}, \dots, i_n\}$ とおく。 A の $i_1, \dots, i_m$ 番目の列ベクトルを並べた行列を A_B 、 $i_{m+1}, \dots, i_n$ 番目の列ベクトルを並べた行列を A_N とする。同様に、設計変数ベクトル x を x_B, x_N 、目的関数の係数ベクトル c を c_B, c_N と分割する。なお x_B に含まれる変数を**基底変数**、 x_N に含まれる変数を**非基底変数**と呼ぶ。

このとき制約式は、 A_B が正則で逆行列が存在することに注意して、以下のように変形できる。

$$\begin{aligned} Ax &= b \\ \iff A_B x_B + A_N x_N &= b \\ \iff x_B &= A_B^{-1} b - (A_B^{-1} A_N) x_N \end{aligned} \quad (2.5)$$

したがって、 $(x_B, x_N) = (A_B^{-1} b, 0)$ は制約条件 $Ax = b$ を満たす解の1つとなる。これを A_B における**基底解**と呼ぶ。基底解が非負制約 $x \geq 0$ を満たすとき、この解は実行可能であり、特に**実行可能基底解**と呼ぶ。

目的関数に式 (2.5) を代入することで、以下のように定数項と非基底変数の項で記述できる。

$$\begin{aligned} c^\top x &= c_B^\top x_B + c_N^\top x_N \\ &= c_B^\top (A_B^{-1} b - (A_B^{-1} A_N) x_N) + c_N^\top x_N \\ &= c_B^\top A_B^{-1} b + (c_N^\top - c_B^\top A_B^{-1} A_N) x_N \end{aligned} \quad (2.6)$$

ピボット操作

シンプレックス法は、基底解から出発して、目的関数の値を改善する方向に基底解を移動させることで最適解を探索する。この移動は、基底変数と非基底変数を交換する操作である**ピボット操作**によって行われる。

ここで、問題 (LP) を式 (2.5), (2.6) を用いて以下のように記述しなおす。これを A_B に

おける**辞書**あるいは**基底形式**と呼ぶ。

$$\begin{aligned}
 (P') \quad & \max. \quad z_B + \mathbf{c}'_N{}^\top \mathbf{x}_N \\
 \text{s.t.} \quad & \mathbf{x}_B = \mathbf{b}' - \mathbf{A}'_N \mathbf{x}_N \\
 & \mathbf{x}_B \geq \mathbf{0}, \quad \mathbf{x}_N \geq \mathbf{0}
 \end{aligned} \tag{2.7}$$

なお, $z_B = \mathbf{c}_B^\top \mathbf{A}_B^{-1} \mathbf{b}$, $\mathbf{c}'_N = \mathbf{c}_N^\top - \mathbf{c}_B^\top \mathbf{A}_B^{-1} \mathbf{A}_N$, $\mathbf{b}' = \mathbf{A}_B^{-1} \mathbf{b}$, $\mathbf{A}'_N = \mathbf{A}_B^{-1} \mathbf{A}_N$ である。

式 (2.7) において, $\mathbf{c}'_N$ が正の成分を持つ場合, その成分を係数に持つ非基底変数を 0 から増加させることで目的関数の値を改善できる (最大化問題であることに注意)。具体的に, ある $q \in I_N$ で $(\mathbf{c}'_N)_q > 0$ であるとき, 非基底変数 x_q を増加させることを考える。このとき, ある基底変数 $x_i (i \in I_B)$ を含む制約式は式 (2.8) のように表される。

$$x_i = (\mathbf{b}')_i - (\mathbf{A}'_N)_{iq} x_q - \sum_{j \in I_N \setminus \{q\}} (\mathbf{A}'_N)_{ij} x_j \tag{2.8}$$

非負制約 $x_i \geq 0$ により, x_q は $\theta_{iq} := \sup\{x_q \mid (\mathbf{b}')_i - (\mathbf{A}'_N)_{iq} x_q \geq 0\}$ まで増加させることができる。なお, $(\mathbf{A}'_N)_{iq}$ が非正である場合, $\theta_{iq} = \infty$ となる。この θ_{iq} を x_q の x_i に対する**増加限界**と呼ぶ。

すべての $i \in I_B$ に対する増加限界の下限 $\inf_{i \in I_B} \{\theta_{iq}\}$ を取る番号を p としたとき, x_q を θ_{pq} まで増加させることができる。このとき, 基底変数 $x_i (i \in I_B)$ は式 (2.8) にしたがって $(\mathbf{A}'_N)_{pq} \theta_{pq}$ だけ減少する。なお, $\theta_{pq} = \infty$ であるとき, x_q をいくらでも増加させることができるため, 問題 (LP) は非有界となる。

基底変数 x_p と非基底変数 x_q の交換は, 式 (2.8) にて $i = p$ とし, x_q の式として式 (2.9) と記述しなおし, 式 (2.7) に代入することで実行することができる。

$$x_q = \frac{(\mathbf{b}')_p}{(\mathbf{A}'_N)_{pq}} - \frac{1}{(\mathbf{A}'_N)_{pq}} x_p - \sum_{j \in I_N \setminus \{q\}} \frac{(\mathbf{A}'_N)_{pj}}{(\mathbf{A}'_N)_{pq}} x_j \tag{2.9}$$

この操作により, 基底解は以下のように変化する。

- $i \in I_B \setminus \{p\}$ について $x_i = \mathbf{b}' - (\mathbf{A}'_N)_{iq} \theta_{pq}$
- $i = q$ について $x_q = \theta_{pq}$
- $i = p$ および $i \in I_N \setminus \{q\}$ について $x_i = 0$

このときの目的関数値は $z_B + (\mathbf{c}'_N)_q \theta_{pq}$ となる。

以上の操作を, 目的関数における非基底変数の係数に正の成分がなくなるまで繰り返すことで, 最適解を探索することができる。

シンプレックス表

シンプレックス法において, 基底形式を表形式で記述したものを**シンプレックス表 (Simplex Tableau)**と呼ぶ⁽²⁾。シンプレックス表は, 基底形式の制約式と目的関数の係数

と定数項をまとめて記述したものであり、基底形式の基底変数と非基底変数の係数を表形式で整理することで、ピボット操作を行いやすくするメリットがある。また、アルゴリズムの実装においても、シンプレックス表を用いることで計算を効率化することができる。

一般に表形式で記述する際は、目的関数を変数 $z = \mathbf{c}^\top \mathbf{x}$ として表現し、制約条件に $z - \mathbf{c}^\top \mathbf{x} = 0$ を追加してすべての制約式の係数と定数項を並べることで表を作成する。

$$\begin{array}{c|ccc}
 & z & \mathbf{x}_B & \mathbf{x}_N & \text{const.} \\
 \hline
 1 & 1 & -\mathbf{c}_B^\top & -\mathbf{c}_N^\top & 0 \\
 0 & 0 & \mathbf{A}_B & \mathbf{A}_N & \mathbf{b}
 \end{array} \quad (2.10)$$

また基底形式で記述すると式 (2.11) となる。

$$\begin{array}{c|ccc}
 & z & \mathbf{x}_B & \mathbf{x}_N & \text{const.} \\
 \hline
 1 & 1 & 0 & -\mathbf{c}'_N{}^\top & z_B \\
 0 & 0 & \mathbf{I} & \mathbf{A}'_N & \mathbf{b}'
 \end{array} \quad (2.11)$$

表を用いると、最適性の判定は目的関数に関する行がすべて非負であるかどうかを確認することで行うことができる。目的関数の行に負の値が q 列目にあった場合、各行に対し増加限界を計算し、その中で最も小さい値を取る行を p 行目とすると、 q 列目において p 行目のみを 1、それ以外を 0 とするように行基本変形を施すことで基底変数と非基底変数を交換することができる。

最適化の流れ

以上の説明をまとめると、シンプレックス法の最適化の流れは Algorithm 2.1 のようになる。

Algorithm 2.1 Simplex Method(P)

Input: Single-Objective Linear Programming Problem (P)

Output: Optimal solution $\mathbf{x}^*$

- 1: Initialize the Simplex Tableau from (P)
 - 2: **while** There exists a negative value in the top row **do**
 - 3: Select the column q with a negative value
 - 4: Calculate the increase limit θ_{iq} for each row i
 - 5: Select the row p with the smallest increase limit
 - 6: Pivot the row p and column q
 - 7: **end while**
 - 8: Obtain the optimal solution $\mathbf{x}^*$ from the Simplex Tableau
 - 9: **return** $\mathbf{x}^*$
-

注意点

解く対象の問題が不等式制約を含む場合、シンプレックス法は不等式制約を等式制約に変換する必要がある。この変換は、スラック変数 $\tilde{x}$ を導入することで行われる⁽²⁾:

$$Ax \geq b \rightarrow Ax - \tilde{x} = b, \quad \tilde{x} \geq 0 \quad (2.12)$$

また本稿では詳細を割愛するが、シンプレックス法ではさまざまなケースに応じていくつかのバリエーションが存在する。

- 基底解を導出することが困難である場合、人工変数を追加した補助問題を解く **2 段階シンプレックス法** により基底解を導出できる。
- 基底変数に 0 があるとき基底解が**退化**しているといい、更新が巡回してしまうことがある。この場合、ピボット操作において **Bland の最小添字規則**などの選択規則を用いることで巡回を防ぐことができる⁽³⁾。

シンプレックス法は、線形計画問題に対する効率的な解法であるが、整数計画問題に対しては適用できない。そのため、整数計画問題に対しては、シンプレックス法を用いて線形計画問題を解いた後、整数制約を満たす解を探索するアプローチが取られることが多い。

次節以降では、整数変数を持つ問題にも適用が可能である切除平面法と分枝限定法について説明する。

2.2.3 切除平面法

はじめに

切除平面法 (Cutting Plane Method) は、Gomory によって提案された整数計画問題の解法として広く用いられるアルゴリズムの 1 つである⁽⁴⁾。切除平面法では、実行可能解を削除しないように新たな制約条件を追加することで、徐々に実行可能領域を制限して整数解を導出する。特に、切除平面法は最適性を保証する理論的基盤を持つだけでなく、実用的なアルゴリズムとしても数理最適化の分野で多大な貢献を果たしている。また、切除平面法は非常に多くの制約条件を持つ問題を表現するためにも用いられる。

本節では、目的関数値を最大化する以下の整数計画問題 (IP) を例に、切除平面法の基本的な考え方とアルゴリズムについて説明する。

$$\begin{aligned} (\text{IP}) \quad & \max. \quad \mathbf{c}^\top \mathbf{x} \\ & \text{s.t.} \quad \mathbf{Ax} \leq \mathbf{b} \\ & \quad \mathbf{x} \in \mathbb{Z}_+^n \end{aligned} \quad (2.13)$$

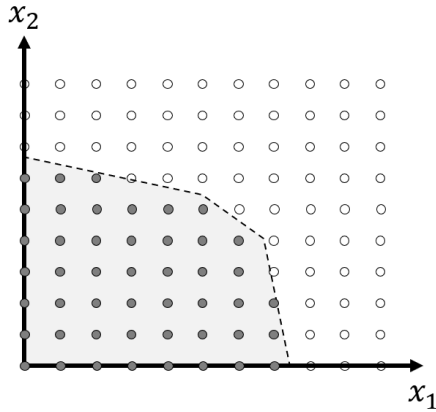


Fig. 2.6 Feasible region of (IP)

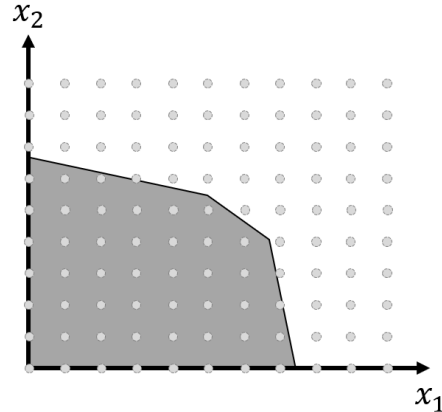


Fig. 2.7 Feasible region of (LP)

線形緩和問題

整数計画問題 (IP) では、設計変数 $\mathbf{x}$ が整数値を取る制約条件が課されている。この制約条件を緩和し、設計変数が実数値を取る問題を考えることで、整数計画問題 (IP) を線形計画問題 (LP) に変換することができる。この緩和された問題を**線形緩和問題**と呼ぶ。

線形緩和問題は、定義から以下の性質を持つ⁽⁵⁾。

- 線形緩和問題の最適値は、元の整数計画問題の最適値以上である。
- 線形緩和問題が実行不能ならば、元の整数計画問題も実行不能である。
- 線形緩和問題の最適解が整数解である場合、その解は元の整数計画問題の最適解である。

これは、Fig. 2.6, 2.7 に示すように、線形緩和問題 (LP) の実行可能領域が整数計画問題 (IP) の実行可能領域を包含していることから明らかである。

線形緩和問題は、前節のシンプレックス法を用いて効率的に解くことができるが、多くの場合は整数解を得ることができない。切除平面法では線形緩和問題を解いて得られる**緩和最適解**をもとに整数解を探索する。

切除平面

ある問題の実行可能領域 S について、任意の実行可能解 $\mathbf{x} \in S$ がある不等式 $\boldsymbol{\alpha}^\top \mathbf{x} \leq \beta$ を満たすとき、この不等式を**妥当不等式 (valid inequality)**と呼ぶ。定義から、妥当不等式を問題に追加しても S の実行可能解は削除されない。切除平面法では、緩和最適解が整数解でない場合、その解を削除する妥当不等式を生成し、問題に追加することで整数解を探索する。このような妥当不等式を**切除平面 (cutting plane)**あるいは単に**カット (cut)**と呼び、切除平面を生成することを**分割 (separation)**と呼ぶ。

切除平面法においては、切除平面の生成方法によってさまざまな手法が提案されているが、その中でも特に有名な手法として**ゴモリーカット (Gomory Cut)** が挙げられる。

ある整数計画問題 (IP) において、その線形緩和問題をシンプレックス法で解いたときに得られた緩和最適解が整数解でないならば、ある小数部を持つ変数 x_j が存在する。このとき、 x_j は 0 でないため解における基底変数であり、以下のような制約式 (2.14) が存在する。

$$x_j + \sum_{i \in I_N} \alpha_i x_i = \beta \quad (2.14)$$

なお、シンプレックス法では非基底変数 $x_i (i \in I_N)$ に 0 を代入し基底解を得るため、 $x_j = \beta$ となる。

ここで、各変数 x_i は非負であるから、 $\alpha_i x_i \geq \lfloor \alpha_i \rfloor x_i$ が成り立つ。ただし、 $\lfloor \cdot \rfloor$ は床関数 (floor function) であり、 $\lfloor \alpha_i \rfloor$ で α_i の小数部を切り捨てた値を表す。これを式 (2.14) に代入することで、以下のような不等式 (2.15) が得られる。

$$x_j + \sum_{i \in I_N} \lfloor \alpha_i \rfloor x_i \leq \beta \quad (2.15)$$

これは線形緩和問題 (LP) の妥当不等式である。さらに、この不等式の左辺は整数値を取るため、右辺値も整数値以下と限定した以下の式 (2.16) は (IP) の妥当不等式となる。

$$x_j + \sum_{i \in I_N} \lfloor \alpha_i \rfloor x_i \leq \lfloor \beta \rfloor \quad (2.16)$$

いま得られていた解は $x_j = \beta, x_i = 0 (i \in I_N)$ であったが、この解は不等式 (2.16) を満たさない。よってこの不等式を (LP) に追加することで、 x_j における整数制約違反を改善した新たな線形緩和問題を得ることができる。

以上の手順を繰り返すことで、線形緩和問題の解が整数解に近づくように制約を追加していく。切除平面を追加した線形緩和問題を解いた際に得られた解が整数解であれば、その解が元の整数計画問題 (IP) の最適解である。

Fig. 2.8 に、切除平面法の概念図を示す。得られた緩和最適解をもとに得られた切除平面を追加することで、実行可能領域を縮小することができることを示している。

切除平面は Gomory Cut 以外にもさまざまな手法が提案されており、問題の特性に応じて適切な切除平面を選択することが重要である。また、得られた解に基づいた切除平面の生成をすること自体を最適化問題として定式化することも可能である。たとえば、得られた緩和最適解を違反させる妥当不等式のうち、制約違反量を最大化する最適化問題を解くなどが考えられる。

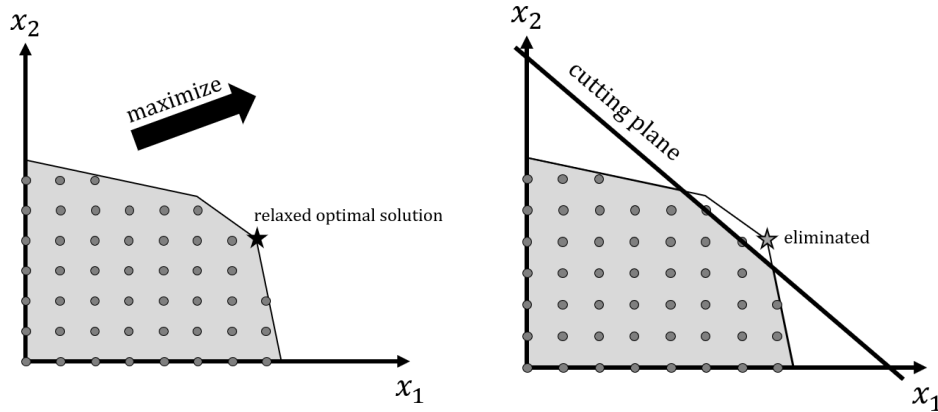


Fig. 2.8 The concept of cutting plane method

最適化の流れ

切除平面法は以下の Algorithm 2.2 のように記述される．これにより，切除平面法は段階的に解の領域を縮小し，最終的に整数解を得ることを目指す．

Algorithm 2.2 Cutting Plane Method (IP)

Input: Integer Programming Problem (IP)

Output: Optimal solution x^*

- 1: Generate the Linear Relaxation Problem (LP) from (IP)
 - 2: Solve the (LP) and obtain the relaxation optimal solution x
 - 3: **while** x is not integer **do**
 - 4: Generate a cutting plane based on x
 - 5: Add the cutting plane to the (LP)
 - 6: Solve the updated (LP) and obtain the relaxation optimal solution x
 - 7: **end while**
 - 8: **return** optimal solution x
-

遅延制約の実現

切除平面法の適用方法として，問題におけるすべての制約条件を考慮するのではなく，必要に応じて一部の制約を後から動的に追加する手法がある．この手法は Cplex や Gurobi などの数理最適化ソルバーで広く利用されており，**遅延制約 (lazy constraint)** と呼ばれることがある^(6,7)．この手法は，制約の数が非常に多く，初期の問題設定にすべてを含めることが計算量的に困難な場合に有効である．

代表的な例として，巡回セールスマン問題 (Travelling Salesman Problem, TSP) のような問題では，部分巡回路除去制約 (Subtour Elimination Constraint) を含むことで最適解が求められる．しかし，これらの制約は問題の規模に対して指数個存在するため，問題を

直接解くことは実質的に不可能である。

遅延制約を用いたアプローチでは、部分巡回路除去制約を事前にすべて問題に含めるのではなく、以下のように段階的に追加する。

1. 部分巡回路除去制約を除いた緩和問題を定式化し、解を求める。
2. 得られた緩和解が部分巡回路を含む場合、その部分巡回路を除外する制約を動的に生成する。
3. 新たな制約を問題に追加し、再度問題を解く。
4. この手順を、解が部分巡回路を含まなくなるまで繰り返す。

このアプローチでは、初期の定式化に含まれる制約の数を大幅に削減できるため、問題の計算効率を向上させることができる。また、得られる最終的な解は、すべての必要な制約を満たすため、元の問題の最適解となる。

注意点

切除平面法は、整数計画問題において強力かつ理論的に正当性のある解法の 1 つであるが、その適用に際していくつかの注意点が存在する。

まず、切除平面法では反復的に切除平面を生成・追加することで解を改善していくため、収束速度が問題の特性や切除平面の生成方法に大きく依存する。特に、生成される切除平面が解の改善に寄与しない場合、収束が遅くなる可能性がある。そのため、効率的な切除平面の選択・生成アルゴリズムが重要である。

次に、切除平面法によって追加される制約条件が多くなると、計算コストが増大する可能性がある。この問題に対処するため、不要になった切除平面を適切に削除する手法や、問題を部分的に解く手法が提案されている。

最後に、切除平面法を用いる際には数値安定性にも注意が必要である。特に、浮動小数点演算に起因する誤差によって、追加された切除平面が誤って妥当解を削除してしまうリスクがある。この問題を緩和するためには、数値誤差を考慮した制約の設定や、適切な精度のアルゴリズムの採用が重要である。

以上の点を踏まえ、切除平面法を適用する際には問題の特性に応じた工夫や、他の解法との併用を検討することが望ましい。

2.2.4 分枝限定法

はじめに

分枝限定法 (Branch and Bound) は, Land と Doig によって提案された, 混合整数計画問題, 特に組合せ最適化問題において広く用いられるアルゴリズムである. 基本的なアイデアは, 解空間を系統的に分割し (分枝), 特定の条件に基づいて不要な分枝を除外する (限定) ことで, 計算効率を向上させる点にある.

分枝限定法では, 問題を部分問題へと分割して行くことを再帰的に繰り返すため, この関係を木構造で表現することができる. この木構造を**探索木 (search tree)**と呼び, 各ノードは部分問題を表し, 枝分かれば部分問題の分割を表す.

分枝操作

分枝限定法では, 解空間を分割するための**分枝操作 (branching)**が中心となる. 分枝操作は, 探索木のノードを選択し, そのノードが表す部分問題をいくつかの部分問題に分割する操作である. この分割は通常, 設計変数の 1 つに値を割り当てることで行われる. たとえば, 整数計画問題において, 設計変数 x_i に整数値 k を割り当てることで, $x_i \leq k$ と $x_i \geq k + 1$ の 2 つの部分問題に分割することができる.

限定操作

分枝操作のみでは解の個数のオーダーのノードが生成されてしまうため, 計算コストが膨大になる可能性がある. この問題を解決するため, 分枝操作に加えて, 各部分問題に対する**限定操作 (bounding)**が行われる. 限定操作は, 各部分問題における目的関数の下界または上界を計算し, 現在得られている最良解と比較することで, 不要な分枝を除外する操作である.

たとえば, 最小化を目的とした混合整数計画問題 (P) を考える. あるノードについて, そのノードに対応する部分問題の線形緩和問題 (P') を考える. (P') はシンプレックス法などで効率的に解くことができ, その最適値 $z^*(P')$ は元の部分問題の最適値 $z^*(P)$ の下界を与える. つまり, $z^*(P) \leq z^*(P')$ が成り立つ. この下界が現在得られている最良値 $\tilde{z}$ 以上の場合, 式 (2.17) によりその部分問題が暫定最良解よりも優れた解を持つ可能性がないため, その部分問題を探索対象から除外することができる.

$$\tilde{z} \leq z^*(P) \leq z^*(P') \quad (2.17)$$

Fig. 2.9 に, 分枝限定法の概念図を示す. 探索木の各ノードに対して, 分枝操作により部分問題を分割し, 限定操作により不要な分枝を除外することで, 最適解を効率的に探索することができる.

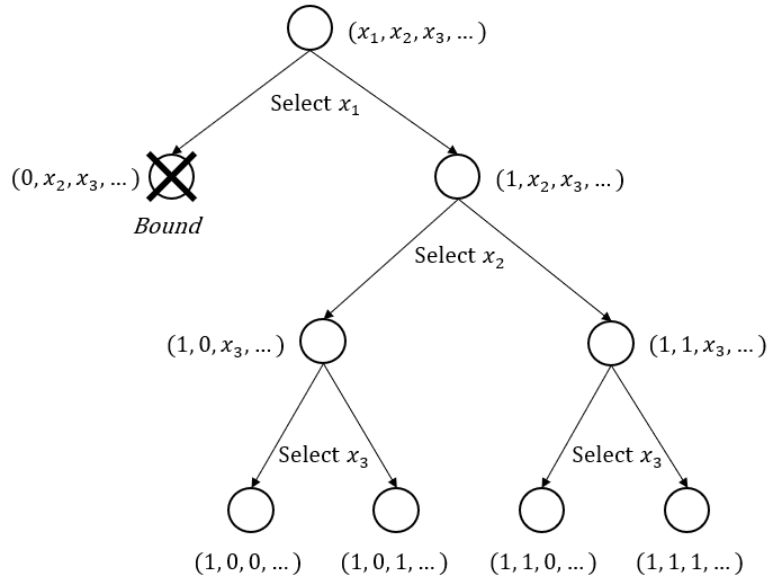


Fig. 2.9 The concept of branch-and-bound

分枝変数の選択方法

分枝限定法において、どの変数を分枝の基準として選択するかは、アルゴリズムの効率性に大きな影響を与える。これにはいくつかの戦略が提案されており、主なものを以下に示す。

Most Infeasible Branching

Most Infeasible Branching は、現在の部分問題における線形緩和問題の解を基に、最も整数解から遠い変数を分枝基準として選択する方法である。具体的には、変数 x_i の線形緩和問題の値 $\hat{x}_i$ について、上下の整数値との距離 $f_i = \min(\lceil \hat{x}_i \rceil - \hat{x}_i, \hat{x}_i - \lfloor \hat{x}_i \rfloor)$ を計算し、 f_i が最大となる変数を選択する。このアプローチの背景には、整数解から最も遠い変数が分枝後の制約を強化する可能性が高いという考え方がある。

Strong Branching

Strong Branching は、複数の候補変数について仮想的に分枝を試み、各場合における下界の改善幅を計算し、最も効果的な分枝を選択する方法である。この方法は、分枝後の探索効率を向上させる可能性があるが、仮想的な分枝を行うための計算コストが高い。そのため、実際には適切な候補変数を限定した上で適用することが多い。このアプローチは、特に探索木の初期段階や重要な部分問題に対して有効である。

これらの分枝変数の選択方法は、問題の特性や計算リソースに応じて適切に選択するこ

とが重要である。

次に分枝するノードの選択方法

分枝限定法において、次に分枝するノードを選択する方法も重要である。この選択方法にはいくつかの戦略が提案されており、主なものを以下に示す。

深さ優先探索 (Depth First Search, DFS)

深さ優先探索は、現在の探索パスをできるだけ深く進め、行き止まりに達した場合に戻る方法である。この方法は、必要なメモリ量が少なく、部分解を早期に見つける可能性がある。しかし、全体の探索が偏る可能性があり、特に探索木の一部に不要な分枝が多い場合には非効率になることがある。

幅優先探索 (Breadth First Search, BFS)

幅優先探索は、探索木の同じ深さにあるノードを順番に探索する方法である。この方法は、全体の探索構造を広くカバーするため、並列計算に適しているが、目的関数値の改善が遅くなる可能性がある。そのため、探索木が大きい場合には効率が低下することがある。また、未探索のノードをすべて保持する必要があるため、メモリ使用量が増大することがある。

最良下界優先探索 (Best First Search, BestFS)

最良下界優先探索は、現在の部分問題の評価値（通常は下界値）に基づいて、最も有望なノードを優先的に探索する方法である。目的関数の改善が早期に見込めるため、実用上はこの戦略がよく用いられる。ただし、幅優先探索と同様に、未探索のノードを保持する必要があるため、メモリ使用量が増大することがある。

最適化の流れ

分枝限定法は以下の Algorithm 2.3 のように記述される。これにより、分枝限定法は探索木を構築し、各ノードに対して分枝操作と限定操作を繰り返すことで最適解を探索する。

Algorithm 2.3 Branch and Bound Method(P)

Input: Mixed Integer Programming Problem (P)**Output:** Optimal solution x^*

- 1: Initialize the search tree with the root node
 - 2: Set the best solution $\tilde{x}$ to ∞
 - 3: **while** There are unexplored nodes **do**
 - 4: Select a node to branch
 - 5: Branch the selected node into multiple subproblems
 - 6: Solve the linear relaxation problem for each subproblem
 - 7: Update the best solution $\tilde{x}$ if a better feasible solution is found
 - 8: Perform bounding for each subproblem if necessary
 - 9: Add the subproblems to the search tree
 - 10: **end while**
 - 11: **return** best solution $\tilde{x}$
-

他の手法との組み合わせ

分枝限定法の性能を向上させるためには、他の手法と組み合わせる利用することが効果的である。特に、解空間の分割と計算効率の向上に寄与する手法を組み合わせることで、分枝限定法の適用範囲と実用性を大幅に広げることができる。以下に代表的な組み合わせの手法を示す。

- 切除平面法との組み合わせ: 切除平面法により緩和問題の精度を向上させることができるため、限定操作がより行われやすくなる。分枝限定法と切除平面法を組み合わせた手法は、特に**分枝カット法 (Branch and Cut)** として知られている。
- ヒューリスティック解法との組み合わせ: ヒューリスティック解法により高速に近似解を得ることができるため、初期解の生成に利用することで、探索効率を向上させることができる。

これらの手法を組み合わせることで、分枝限定法の計算効率を向上させることができる。

2.3 多目的最適化

2.3.1 はじめに

多目的最適化問題 (Multi-objective Optimization Problem, MOP) は、次の形式で表現される。

$$\min. \quad \mathbf{F}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_k(\mathbf{x}) \end{bmatrix} \quad (2.18)$$

$$\text{s.t.} \quad \mathbf{x} \in S \quad (2.19)$$

ここで、 $\mathbf{F}(\mathbf{x})$ は k 個の目的関数 $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})$ を持つベクトル値関数であり、 $\mathbf{x} \in \mathbb{R}^n$ は n 次元の設計変数ベクトルを表す。また、 $S \subseteq \mathbb{R}^n$ は設計変数の制約を満たす実行可能領域を示す。多目的最適化では、 $\mathbf{F}(\mathbf{x})$ の各成分 $f_i(\mathbf{x})$ を同時に最適化することを目指す。目的関数どうしがトレードオフの関係を持つことが一般的であるため、単一の最良解を求めることは困難である。

パレート最適性の概念

多目的最適化において、解 $\mathbf{x}_1, \mathbf{x}_2 \in S$ の優劣は支配関係 (dominance) によって定義される。解 $\mathbf{x}_1$ が解 $\mathbf{x}_2$ を**支配する** ($\mathbf{x}_1 \succ \mathbf{x}_2$) とは、次の2つの条件を同時に満たす場合を指す。

1. $\forall i \in \{1, 2, \dots, k\}, \quad f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2),$
2. $\exists j \in \{1, 2, \dots, k\}, \quad f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2).$

なお、最大化問題では符号が反転することに注意する。

Fig. 2.10 に、支配関係の概念を示す。2 目的の最小化問題を考え、各目的関数を各軸にとった目的関数空間を考えた場合、左下に位置する解 $\mathbf{x}_1$ は右上に位置する解 $\mathbf{x}_2$ を支配する。

支配される解よりも支配しない解が望ましいため、解集合 S の中で他の解に支配されない解を集めた集合 S^* を**パレート解集合 (Pareto-optimal set)** と呼ぶ。

$$S^* = \{\mathbf{x} \in S \mid \nexists \mathbf{y} \in S, \mathbf{y} \succ \mathbf{x}\}. \quad (2.20)$$

対応する目的空間において、パレート解集合の像 $\mathbf{F}(S^*)$ を**パレトフロント (Pareto front)** と呼ぶ。パレトフロントは、目的間のトレードオフ関係を視覚化する上で重要な役割を果たす。

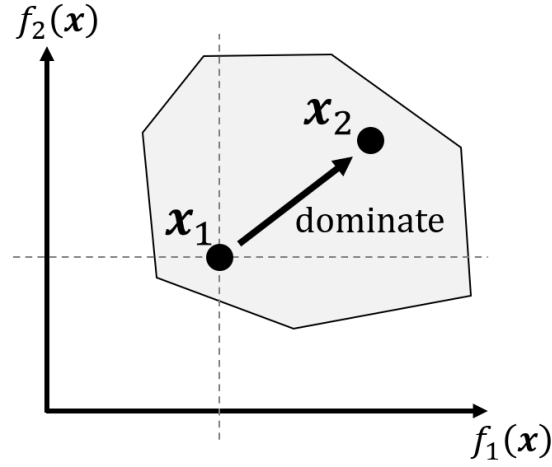


Fig. 2.10 The concept of dominance

解の形の特徴

多目的最適化における解は，単一のスカラー値ではなく，パレート解集合という形で得られる．そのため，アルゴリズムは以下の2つの性質を満たすことが求められる．

- **収束性 (convergence)**：パレートフロント上の理想的な解に近づくこと．
- **多様性 (diversity)**：パレートフロント上に広がる多様な解を探索すること．

これに加え，パレート解集合は，非劣解の個数が膨大になる場合があるため，その計算効率性や結果の解釈性を向上させる工夫が必要となる．

2.3.2 本稿が対象とする問題設定

本稿では，目的関数および制約が線形で表現される場合を仮定し，特に多目的混合整数線形計画問題（Multi-objective Mixed Integer Linear Programming, MOMILP）を対象とする．この問題は以下の形式で表される：

$$\min. \quad Cx = [c_1^\top x \quad c_2^\top x \quad \dots \quad c_k^\top x]^\top \quad (2.21)$$

$$\text{s.t.} \quad Ax \geq b \quad (2.22)$$

$$x \geq 0 \quad (2.23)$$

$$x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I}, \quad x_j \in \mathbb{R} \quad \forall j \notin \mathcal{I} \quad (2.24)$$

ここで， $x = (x_1, x_2, \dots, x_n)^\top$ は決定変数ベクトルを表し，そのうちの一部 $\mathcal{I} \subseteq \{1, 2, \dots, n\}$ は整数制約を受ける． C は k 個の目的関数の係数を並べた $k \times n$ 行列であり， A および b は線形制約を構成する係数行列および定数項ベクトルである．

この問題では，単目的混合整数計画問題と同様に，離散変数を含むために計算負荷が増大するが，さらに複数の目的を同時に最適化する必要があるため，探索空間の複雑性が格

段に増加する。これにより、効率的な探索手法が必要となる。

2.3.3 多目的最適化手法の分類

多目的最適化の手法は、目的空間における解集合をどのように探索するかに基づき、以下のように分類される。

近似解法

近似解法は、パレートフロントの近似解を求めることを目的とする。これらの手法は、厳密解法に比べて計算効率が高く、解空間が大規模な場合でも適用可能である。近似解法の多くは、進化計算手法に基づいており、主に以下の 3 つに大別される。

分割（スカラー化）ベース

分割ベースの手法は、多目的問題を複数の単目的問題に分割し、それらを並列的に解くことでパレートフロントを構築するものである。代表的な手法として MOEA/D (Multi-Objective Evolutionary Algorithm Based on Decomposition) ⁽⁸⁾ がある。この手法は、問題を分割することで各単目的問題の効率的な探索を可能とし、多目的最適化の計算効率を高める。

支配関係ベース

支配関係ベースの手法は、解間の支配関係に基づいて探索を行うものであり、代表的なアルゴリズムとして NSGA-II (Non-dominated Sorting Genetic Algorithm II) ⁽⁹⁾ が挙げられる。この手法では、支配される解を削除しながら解集合の多様性を維持する工夫がなされている。

集合ベース

集合ベースの手法は、解集合全体を直接操作することで探索を行うものである。たとえば、PSO (Particle Swarm Optimization) ⁽¹⁰⁾ は、粒子群の動きに基づいて解集合を探索する。これにより、パレートフロントの多様性を確保しつつ、近似解を効率的に求めることが可能となる。

厳密解法

厳密解法は、最適解を数学的に厳密に求める手法であり、特に多目的混合整数計画問題においては多目的分枝限定法 (Multi-objective Branch and Bound) が広く利用される。この手法は、探索空間を部分問題に分割し、効率的に解空間を探索することで、パレート最適解集合全体を求めることが可能である。しかし、解空間が大規模な場合、計算負荷が著しく増加するため、実行可能な問題規模が制限されるという課題がある。

以降の節では、多目的混合整数計画問題において、近似解法の中で重要な分割の考え方と、厳密解法である多目的分枝限定法について説明する。

2.3.4 多目的最適化問題の分割（スカラー化）

多目的最適化問題は、複数の目的関数を同時に最適化するため、解の優劣を一意に定めることができず、手法の選択や解の評価が困難である。この問題を解決するため、多目的最適化問題を複数の単目的問題に分割する方法が提案されている。

多目的問題を複数の単目的問題に分割するには、**重みベクトル (weight vector)** および **スカラー化関数 (scalarizing function)** を用いる方法が一般的である。

重みベクトル

重みベクトルは、多目的最適化問題の各目的関数に対して重み（優先度）を割り当てるためのベクトルである。目的関数の個数を k としたとき、重みベクトルは次のように表される：

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_k)^\top, \lambda_i \geq 0, \sum_{i=1}^k \lambda_i = 1 \quad (2.25)$$

重みベクトルを用いることで、多目的最適化におけるパレートフロントを近似することが可能となる。一般に、複数の重みベクトルを生成し、それぞれに対して単目的問題を解くことで、パレートフロントの近似解を求める。

ここで、重みベクトルで重みづけを行う前に、各目的関数の取りうる値を正規化しておくことが重要である。正規化を行うことで、各目的関数間のスケールの違いを補正し、重みベクトルの効果を適切に反映することができる。

重みベクトルの分布は、解の多様性に大きな影響を与えるため、適切な重みベクトルの選択が重要である。

スカラー化関数

スカラー化関数は、多目的最適化問題において複数の目的関数を 1 つのスカラー値に変換するための関数である。これにより、複数の目的を同時に考慮する代わりに、単一の目的関数を最適化する問題として解くことが可能となる。スカラー化関数を適切に設計することで、パレートフロント上の異なる解を導出することができる。以下では、代表的なスカラー化関数として **Weighted Sum**, **Tchebycheff**, **Penalty-based Boundary Intersection** を紹介する⁽⁸⁾。

Weighted Sum (加重和法)

Weighted Sum (加重和法) は、各目的関数に重みを割り当ててそれらを線形に結合する最も基本的なスカラー化手法である。重みベクトル λ に対し、Weighted Sum によるスカラー化関数は次のように定義される：

$$f^{\text{WS}}(x|\lambda) = \lambda^\top Cx \quad (2.26)$$

Weighted Sum の長所は、計算コストが低く、パレートフロントが凸の場合には均一に重みベクトルを選択することでパレートフロント全体を近似できることである。特に、線形性を仮定する場合には、目的関数の係数ベクトルを $\lambda^\top C$ とすることで、以下のような単純な混合整数計画問題として解くことができる。

$$\min. \quad (\lambda^\top C)x \quad (2.27)$$

$$\text{s.t.} \quad Ax \geq b \quad (2.28)$$

$$x \geq 0 \quad (2.29)$$

$$x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I}, \quad x_j \in \mathbb{R} \quad \forall j \notin \mathcal{I} \quad (2.30)$$

一方、短所は、非凸なパレートフロントに対しては非凸領域を探索することができないことである。そのため、得られた解がパレートフロント全体を網羅しているかどうかを確認することが重要であり、必要に応じて他のスカラー化関数を用いることが求められる。

Tchebycheff (チェビシェフ法)

Tchebycheff は、目的関数の最大偏差に着目するスカラー化手法である。ここで、**理想点 (ideal point)** を $z^* = (z_1^*, z_2^*, \dots, z_k^*)$ とし、各目的関数の最小値以下の値 ($z_i^* \leq \min_x \{c_i x\}$) で構成されたとする。重みベクトル λ を用いると、Tchebycheff によるスカラー化関数は次のように定義される：

$$f^{\text{Tch}}(x|\lambda, z^*) = \max_{i=1, \dots, k} \{\lambda_i |c_i^\top x - z_i^*|\} \quad (2.31)$$

Tchebycheff を用いることで、非凸なパレートフロントに対しても解を得ることができる。しかし、スカラー化関数が非線形であるため、線形性を仮定する場合は解の探索が難しくなることがある。

この場合、max を取る個所を変数で置き、max 条件を制約として記述することで、以下のような混合整数計画問題として解くことができる。

$$\min. \quad M \quad (2.32)$$

$$\text{s.t.} \quad \lambda(C^\top x - z^*) \leq M1 \quad (2.33)$$

$$Ax \geq b \quad (2.34)$$

$$x \geq 0 \quad (2.35)$$

$$x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I}, \quad x_j \in \mathbb{R} \quad \forall j \notin \mathcal{I} \quad (2.36)$$

ただし、 $\mathbf{1}$ は適切な次元の 1 ベクトルであり、 M は最大値を表現するための非負変数である。

Penalty-based Boundary Intersection (ペナルティに基づく境界交差法)

Penalty-based Boundary Intersection (PBI) は、Tchebycheff の変形であり、重みベクトル方向への指向性を持たせたスカラー化関数である。理想点 $\mathbf{z}^*$ を基準に、重みベクトル $\boldsymbol{\lambda}$ と重みベクトルへの指向性パラメーター $\theta \geq 0$ を用いると、PBI によるスカラー化関数は次のように定義される：

$$f^{\text{PBI}}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*, \theta) = d_1 + \theta d_2, \quad (2.37)$$

$$d_1 = \frac{\|\boldsymbol{\lambda}^\top (\mathbf{C}\mathbf{x} - \mathbf{z}^*)\|}{\|\boldsymbol{\lambda}\|}, \quad (2.38)$$

$$d_2 = \left\| \mathbf{C}\mathbf{x} - \left(d_1 \frac{\boldsymbol{\lambda}}{\|\boldsymbol{\lambda}\|} + \mathbf{z}^* \right) \right\| \quad (2.39)$$

ただし、 d_1 は解の値 $\mathbf{C}\mathbf{x}$ を重みベクトル方向に射影した点と理想点との距離を表し、 d_2 はその射影点と解の値との距離を表す。そして指向性パラメーター θ の値を大きくすることで、重みベクトル方向に対する指向性を強めることができる。

PBI は Tchebycheff に比べて、重みベクトル方向にある解を得やすいため、より望みどおりの解を得ることができる。しかし、Tchebycheff と同様に非線形な関数であるため、線形性を仮定する場合には解の探索が難しくなることがある。

PBI におけるノルムに L_1 ノルム（マンハッタン距離）を用いることで、以下のような混合整数計画問題として定式化することは可能であるが、目的数 k ごとの変数および制約を導入する必要があるため、計算コストが増大することがある。

$$\min. \quad \boldsymbol{\lambda}^\top (\mathbf{C}\mathbf{x} - \mathbf{z}^*) + \theta \mathbf{1}^\top \mathbf{M} \quad (2.40)$$

$$\text{s.t.} \quad \mathbf{C}\mathbf{x} - \left(\boldsymbol{\lambda}^\top (\mathbf{C}\mathbf{x} - \mathbf{z}^*) \boldsymbol{\lambda} + \mathbf{z}^* \right) \leq \mathbf{M} \quad (2.41)$$

$$\mathbf{A}\mathbf{x} \geq \mathbf{b} \quad (2.42)$$

$$\mathbf{x} \geq \mathbf{0} \quad (2.43)$$

$$x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I}, \quad x_j \in \mathbb{R} \quad \forall j \notin \mathcal{I} \quad (2.44)$$

なお、不等式 (2.41) は式変形により以下のように表すこともでき、これにより線形性を確認することができる：

$$(\mathbf{I} - \boldsymbol{\lambda}\boldsymbol{\lambda}^\top)\mathbf{C}\mathbf{x} \leq \mathbf{M} + (\mathbf{I} - \boldsymbol{\lambda}\boldsymbol{\lambda}^\top)\mathbf{z}^* \quad (2.45)$$

スカラー化関数は、多目的最適化問題においてパレートフロントを探索するための基本的なツールであり、各手法には特性に応じた適用範囲がある。問題の形状や特性（凸性や次元数など）を考慮し、適切なスカラー化関数を選択することが求められる。

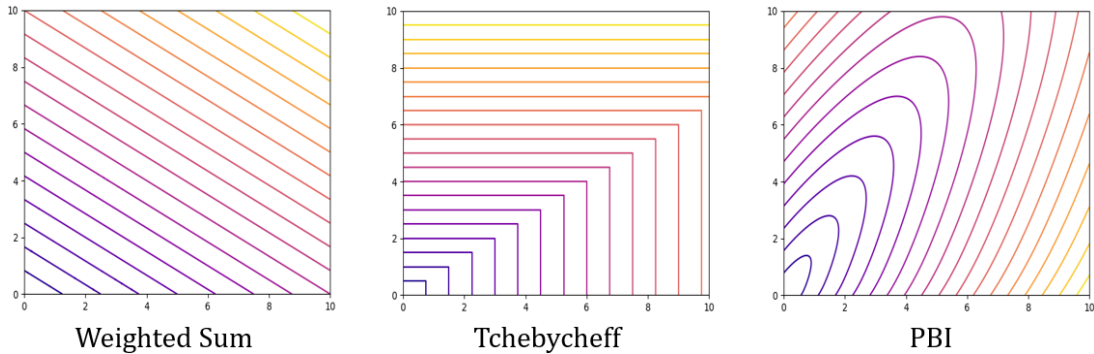


Fig. 2.11 Contours of scalarizing functions

スカラー化関数の比較

Weighted Sum, Tchebycheff, PBI は、それぞれ異なる特性を持つスカラー化関数である。Fig. 2.11 に、これらのスカラー化関数の等高線の比較を示す。Weighted Sum は、各目的関数の線形加重和を取ることから、等高線は直線状になる。一方、Tchebycheff は、各目的関数の最大偏差を取ることから、等高線は V 字型になる。PBI は、重みベクトル方向に対する指向性を持つため、等高線は重みベクトル方向に引き寄せられる。

これらのスカラー化関数は、目的関数の特性や問題の形状に応じて適切に選択することが重要である。

分割（スカラー化）を用いた多目的最適化手法

多目的最適化問題を複数の単目的問題に分割する手法は、多目的最適化の効率的な探索を可能とするため、多目的最適化手法に広く利用されている。代表的な手法として、進化計算手法である **MOEA/D** (Multi-Objective Evolutionary Algorithm based on Decomposition) が挙げられる。

MOEA/D では、生成した重みベクトルごとに解を生成し、進化計算手法により各重みベクトルに対する解を並列的に探索する。解評価をスカラー化関数により行うことで、パレートフロント全体を近似するような解集合を得ることが可能である。

本稿では進化計算手法である MOEA/D の基本的な概念の説明は割愛する。詳細は文献⁽⁸⁾を参照されたい。

2.3.5 多目的分枝限定法

多目的分枝限定法は、2.2.4 節で述べた分枝限定法を多目的最適化問題に拡張した手法である。通常分枝限定法と同様に、多目的分枝限定法は探索木を構築し、各ノードに対して分枝操作と限定操作を繰り返すことでパレート最適解集合を探索する。

多目的分枝限定法が単目的の分枝限定法と異なる点は、限定操作の困難性にある。単目的の場合、目的関数の値が単一の値であるため、限定操作は下界値と暫定最良値の大小関係を比較するだけでよく、比較的容易に行うことができる。しかし、多目的の場合、目的関数の値が複数の値で表現されるため、単純な大小比較では限定操作を行うことができない。

そこで、多目的分枝限定法では、下界値の代わりに**下界集合 (lower bound set)**を導入し、暫定最良解集合との比較を行うことで限定操作を行う。

下界集合

分枝操作により得られたあるノードに着目し、その問題に対応する部分問題 (P) を考える。この部分問題が元の問題の最適解を含む見込みがあるかを判断するために、(P) の緩和問題 (P') を考える。(P') は多目的線形計画問題であり、最適解集合の像はパレートフロントを成す。このように、あるノードに対しその部分問題 (P) の緩和問題に対する最適解集合を、そのノードおよび (P) に対する**下界集合**と呼び、(P) の任意の実行可能解は必ず下界集合が含むいずれかの解に支配される。

より形式的には、(P) の実行可能領域を S 、(P') の実行可能領域を S' とすると、 $S \subseteq S'$ である。(P) の下界集合を L とすると、パレート解集合の定義より $L = \{y \in S' \mid \nexists x \in S', x \succ y\}$ であることから、任意の $y \in L$ に対して、 S の任意の実行可能解 x は $x \succ y$ を満たす。

したがって、下界集合の任意の要素が暫定最良解集合のいずれかの要素に支配される場合、そのノードで得られる解はいずれも暫定解よりも劣るため、そのノードの探索を打ち切ることができる。

さて、下界集合 L は (P') の最適解集合であるため、一般に L の元は無数に存在する。Ehrgott と Gandibleux による提案では、2.3.4 節の重みベクトルと同様の概念を用いて、 L が支配する領域 L_+ を以下のように表現している⁽¹¹⁾。

$$L_+(\Lambda) = \bigcap_{\lambda \in \Lambda} \left\{ y \in \mathbb{R}^k \mid \lambda^\top y \geq \lambda^\top z_\lambda \right\} \quad (2.46)$$

ここで、 Λ は任意の重みベクトルを集めた集合であり、 z_λ は重みベクトル λ を用いて (P') を Weighted Sum により単目的化した問題を解いて得られた最適解である。各重みベクトル λ に対する集合は $\{y \in \mathbb{R}^k \mid \lambda^\top y \geq \lambda^\top z_\lambda\}$ であり、これは点 z_λ を通り λ に直交する超平面より上側の領域を表す。

式 (2.46) は、 L_+ を無限個の領域の共通部分として表現しているため、実際に計算することは困難である。しかし、重みベクトルの全体集合 Λ の代わりに、有限個の重みベクトルからなる集合 $\Lambda' \subset \Lambda$ を考えると、 $L_+(\Lambda) \subseteq L_+(\Lambda')$ であり、 $L_+(\Lambda')$ は S の任意の実

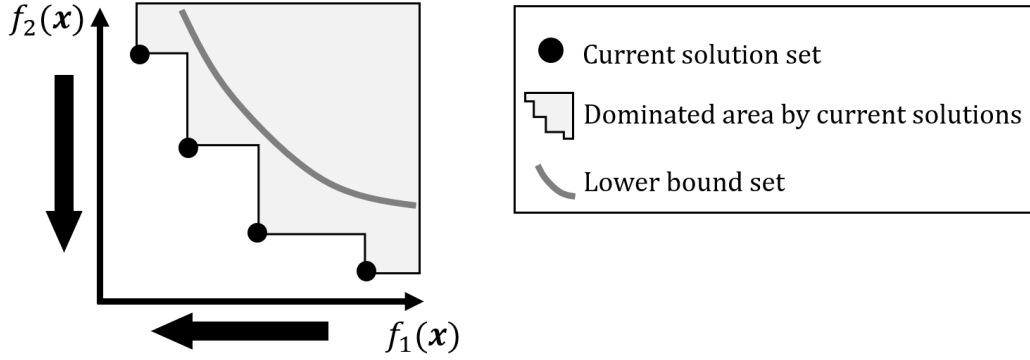


Fig. 2.12 Example of a situation where bounding can be performed in MOBB

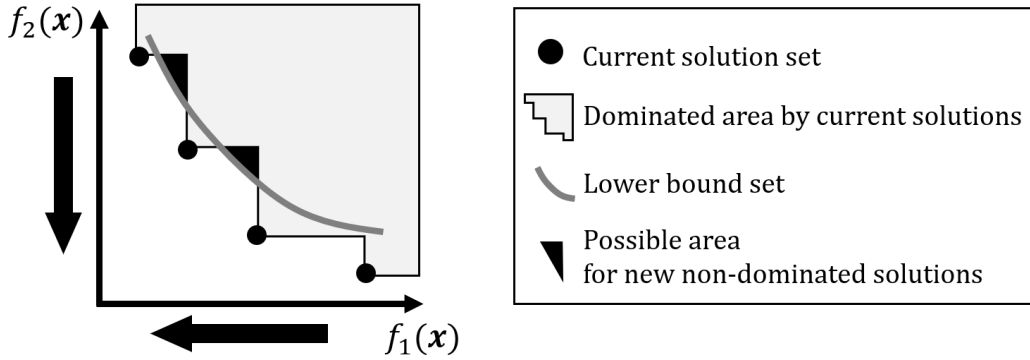


Fig. 2.13 Example of a situation where bounding cannot be performed in MOBB

行可能解を含む。

以上の包含関係をまとめると、次のようになる：

$$S \subseteq S' \subseteq L_+(\mathbf{\Lambda}) \subseteq L_+(\mathbf{\Lambda}') \quad (2.47)$$

これにより、暫定最良解集合の支配領域が $L_+(\mathbf{\Lambda}')$ を包含する場合、暫定最良解集合が着目ノードで得られる任意の解を支配することが保証されるため、そのノードの探索を打ち切ることができる。

Fig. 2.12 および Fig. 2.13 に、下界集合を用いた限定操作の例を示す。Fig. 2.12 では、暫定最良解集合が下界集合を包含しているため、着目しているノードで得られる解は暫定解よりも劣る。したがって、そのノードの探索を打ち切ることができる。一方、Fig. 2.13 では、暫定最良解集合が下界集合を包含していない。そのため、下界集合が支配し、かついずれの暫定解にも支配されない領域で新たな非支配解を得られる可能性がある。よって、着目ノードの探索を継続する必要がある。

したがって、実際の計算では、有限個の重みベクトルに対する下界集合 $L_+(\mathbf{\Lambda}')$ を求め、その上界を暫定最良解集合と比較することで限定操作を行う。

最適化の流れ

多目的分枝限定法の実行可能領域の像の凸包の頂点は、目的数 k が大きくなると指数関数的に増大する⁽¹²⁾。そのため、非支配的な解を列挙する本手法では探索時間が膨大になってしまう。

多目的分枝限定法の課題

多目的分枝限定法は、多目的混合整数線形計画問題に対して有効な手法であるが、以下の点に注意が必要である。

見つけるべきパレート最適解の個数が莫大

多目的最適化問題の実行可能領域の像の凸包の頂点は、目的数 k が大きくなると指数関数的に増大する⁽¹²⁾。そのため、非支配的な解を列挙する本手法では探索時間が膨大になってしまう。

限定操作が行われにくい

多目的分枝限定法における限定操作は、暫定最良解集合と下界集合の比較により行われる。しかし、一般に目的数 k が増えると支配関係は発生しにくくなる。これに加えて、集合どうしの比較には計算コストがかかる。下界集合を成すための重みベクトルの個数 $|\Lambda'|$ を減少させることで計算コストを抑えられるが、この場合より支配関係が成立しにくくなる。

結果として、多目的分枝限定法は、目的数が少なく探索空間が小さい場合に有効であるとされ、大規模な問題に対しては近似解法を用いることが一般的である。

第 3 章

Multi-Objective Branch-and-Bound based on Decomposition (MOBB/D)

3.1 はじめに

本章では、多目的混合整数線形計画問題 (MOMILP) に対する新たな厳密解法である **Multi-Objective Branch-and-Bound based on Decomposition (MOBB/D)** ⁽¹³⁾ を提案する。本手法は、2.3.4 節で述べた分割手法と 2.2.4 節で述べた分枝限定法を統合し、MOMILP の解法における計算効率とパレート解の多様性・収束性を両立させることを目的としている。

従来の多目的分枝限定法は、全てのパレート最適解を網羅的に探索するため、計算コストが指数的に増加する問題があった。一方、進化計算手法は計算効率が高く、高次元の問題にも適用可能であるが、厳密解を保証できず、得られる解の精度に限界がある。これらの課題を解決するため、本研究では、重みベクトルを用いて MOMILP を複数の単目的問題に分割し、各問題に対して分枝限定法を適用することで、厳密なパレート最適解を効率的に求める手法を開発した。

MOBB/D では、重みベクトルによるスカラー化を用いることで、解の個数を制御しつつ、厳密解法である分枝限定法を用いることで最適解を得ることが可能である。さらに、各単目的問題を解くにあたり、**探索情報**を収集・共有することで、計算効率の向上を図る。特に、重みベクトルが近接する単目的問題どうしは、問題構造や最適解が類似していると仮定できるため、最適解周辺の情報や探索途中に得られた知見を、探索情報として他の単目的問題を解く際に再利用することで、探索の効率化が期待できる。

分枝限定法は、単目的の混合整数線形計画問題 (MILP) に対する効率的な厳密解法であ

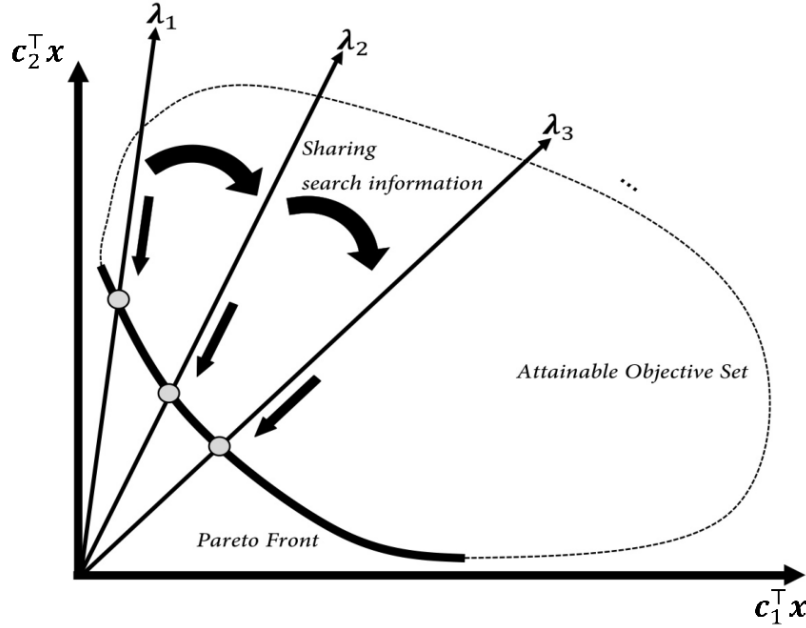


Fig. 3.1 A conceptual diagram of MOBB/D.

る。そのうえ、分枝限定法自体が拡張性に富み、問題特性に応じた改良を行いやすいという特徴がある。したがって、解く対象となる問題に応じて質の良い探索情報を採用することで、よりよい効率化を実現できると考えられる。

Fig. 3.1 に、MOBB/D の概念図を示す。この図では、重みベクトルを用いて多目的問題を複数の単目的問題に分割し、探索情報を共有しながら各単目的問題を解く流れを視覚的に表している。

本章では、まず提案手法の基本的な概念と流れを概説する。続いて、重みベクトルを用いた単目的化の手法と、探索情報の再利用戦略について詳細に述べる。さらに、シンプレックス表の再利用方法を数式について用いて形式的に説明し、提案手法の理論的な妥当性と有効性を示す。最後に、数値実験を通じて、MOBB/D の性能を既存手法と比較・検証する。

3.2 提案手法の設計

本節では、2.3.2 節で述べた多目的混合整数線形計画問題 (MOMILP) に対して、重みベクトルを用いたスカラー化により単目的化を行う手法について述べる。分割の詳細は 2.3.4 節を参照されたい。提案手法 MOBB/D では、この単目的化によって各部分問題を独立して解くことが可能となり、全体の計算効率を向上させる。

本節では、具体的な重みベクトルの生成方法および分割方法について述べる。

3.2.1 重みベクトルの生成

提案手法 MOBB/D では、パレートフロント上の多様な解を効率的に得るために、重みベクトルを用いて多目的混合整数線形計画問題 (MOMILP) を単目的問題に分割する。

重みベクトル $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)^\top \geq \mathbf{0}$ は、各目的関数に割り当てる重みを表す、要素の和が 1 である実数ベクトルである。異なる λ を用いることで、パレートフロント上の異なる解を探索することが可能となる。

実際に最適化を行う際には、まず重みベクトル集合 $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{|\Lambda|}\}$ を生成し、分割方向を定める必要がある。 Λ を生成する際には、以下の点を考慮する必要がある。

多様性の確保

重みベクトル集合はパレートフロント全体を網羅するように設計されるべきである。具体的には、重みベクトルが可能な限り均等に分布するように生成することで、パレートフロント上の多様な解を得ることが期待できる。たとえば、単位 $k-1$ 次元単体内の格子点を用いる方法や、球面上の等間隔分割を利用する方法が考えられる。

計算コストの制御

重みベクトルの数 $|\Lambda|$ を増加させることで、パレートフロント上の解の密度を向上させることができるが、計算コストが増大する。しかし、重みベクトルの数が少ないと、パレートフロントの一部しか探索できない可能性があるのに加え、近傍にある重みベクトルに対応する部分問題間の類似性が弱くなるため、探索情報の再利用が効果を発揮しにくくなってしまう。したがって、問題の規模や計算リソースに応じて $|\Lambda|$ を適切に選択する必要がある。

重みベクトルの生成は、MOBB/D の性能を左右する重要な要素である。適切な重みベクトル集合を設計することで、パレートフロントの多様性と収束性を両立しつつ、計算コストを制御することが可能となる。

3.2.2 スカラー化関数の選択

提案手法 MOBB/D におけるスカラー化関数の選択は、最適化プロセスの効率性とパレートフロントの構築精度に大きな影響を及ぼす。スカラー化関数自体の定義と特性については、2.3.4 節で詳細に述べた。本節では、提案手法において適切なスカラー化関数をどのように選択し、適用するべきかについて議論する。特に、Weighted Sum の優位性を強調しつつ、その限界を補うための戦略を提示する。

MOMILP に対する Weighted Sum の特性

Weighted Sum は、MOMILP を単目的化するためのスカラー化関数として、提案手法 MOBB/D の中心的な役割を担う。Weighted Sum の特性は、計算効率の向上と形式のシンプルさにおいて特に顕著であり、提案手法の設計方針と整合性が高い。

Weighted Sum は、目的関数を重みベクトルにより線形結合する方法であり、MOBB/D が対象とする線形性を仮定した MOMILP に適している。この手法では、スカラー化された単目的問題が線形または混合整数線形問題として扱われるため、探索情報の再利用が容易である。それに加え、目的関数が単純な線形結合として表されるため、計算コストを低減し、解探索の効率化が可能となる。

一方で、Weighted Sum では非凸パレートフロントへの対応できないという限界がある。解く対象の MOMILP のパレートフロントが非凸形状を持つ場合、非凸部分に含まれるパレート解を探索できない。これは、重みベクトルがカバーできる解の領域が凸集合に限定されるためである。

そのため、非凸部分が大きい場合、他のスカラー化関数を採用することで解消できる可能性がある。たとえば、Tchebycheff や PBI は、非凸パレートフロントのカバー範囲を拡張するために適している（詳細は 2.3.4 参照）。ただし、これらの手法は Weighted Sum に比べて計算コストが高く、探索情報の再利用が複雑化するため、適用範囲を慎重に選定する必要がある。スカラー化関数の選択は、MOBB/D の効率性とパレートフロントの品質に直結する重要な要素である。提案手法では、Weighted Sum を基本として使用し、非凸フロントなどの特殊な問題には Tchebycheff や PBI を必要に応じて適用することで、計算効率と探索精度の両立を図る。

3.2.3 探索情報の再利用

提案手法 MOBB/D では、重みベクトルを用いて MOMILP を複数の単目的問題へと分割する際、各単目的問題を解く過程で得られる情報を効果的に再利用することを重視する。ここでいう**探索情報**とは、分枝限定法やシンプレックス法などを通じて得られる、問題構造や部分解に関する有益な知見を指す。これらの情報を別の重みベクトルに対応する問題を解く際に活用することで、探索の初期段階から有効なヒントを得られ、計算コストの削減や解の収束性向上が期待できる。

探索情報には、多様な種類と活用方法が考えられるが、ここでは代表的な例をいくつか示す。

- **部分最適解およびその評価値の活用**: ある重みベクトルで定義された単目的問題を解く過程で得られた最適解は、類似した重みベクトルを持つ他の問題において、有

望な初期解として再利用できる。これにより、新たな問題をゼロから解き始める場合と比較して、早期に質の高い解へ到達することが期待される。

- **変数固定および制約緩和情報の再利用:** 過去の探索過程で特定の変数が固定されることによって収束性が向上した場合や、不要な制約を省略することで探索範囲が限定できた場合、その情報は新たな問題でも有用である可能性が高い。これらの情報を適切に適用することで、問題サイズや解探索空間を効果的に縮小し、分枝限定木の展開を抑制できる。
- **線形緩和の基底情報:** シンプレックス法で線形緩和問題を解くと、その過程で形成された基底情報（シンプレックス表）が得られる。この基底情報は、近い構造を持つ別の単目的問題に対して有望な初期基底を提供し、線形緩和問題の計算における反復を大幅に削減することが可能である。

これらの探索情報を再利用することによって得られる利点は以下のとおりである。

- **計算時間の短縮:** 類似した構造を持つ問題間で探索情報を共有することで、新たな問題に対して探索空間を初期段階から限定し、無駄な分枝や反復を削減できる。
- **収束性の向上:** 良質な初期解や有効な制約簡略化情報を活用することで、より早期にパレートフロント近傍の有望な解に収束でき、計算資源を効率的に活用できる。
- **効率的なパレートフロント構築:** 重みベクトルが近接する問題同士が探索情報を交換することで、幅広い範囲のパレートフロントを効率的にカバーし、多様なパレート解を得やすくなる。

本稿では、これらの概念的な探索情報再利用戦略を具現化する具体例として、3.4 節で**シンプレックス表の再利用手法**について説明する。すなわち、一度解いた単目的問題で得られた線形緩和最適解の基底情報（シンプレックス表）を他の単目的問題に適用し、再度のシンプレックス法実行時に高速な解探索を可能とする技術について解説する。こうした情報共有は、MOBB/D における計算効率・解の品質向上に不可欠な要素となり得る。

3.3 最適化の流れ

本節では、提案手法 MOBB/D による最適化のプロセスを Algorithm 3.4 に示す。探索情報を格納するための集合 S を用意し、各単目的問題を解く際にこの S を活用する。

はじめに多目的混合整数線形計画問題（MOMILP）を複数の単目的問題に分割するため、適切な重みベクトル集合を生成する。続いて、その重みベクトルごとに MOMILP を単目的混合整数線形計画（MILP）問題へ変換し、 S 内の探索情報を活用しながら分枝限定

法で最適化を行う。得られた最適解を集約することで、最終的にパレートフロント上にある解集合を構築できる。この反復過程で新たに得られた情報は順次 $\mathcal{S}$ へ蓄積・更新され、次以降の問題解法に役立てられる。

Algorithm 3.4 MOBB/D for a Given MOMILP

Input: A multi-objective mixed integer linear programming problem (MOMILP)

Output: A set of solutions $\mathcal{X}^*$ approximating the Pareto front

- 1: Generate the set of weight vectors Λ according to a suitable distribution
 - 2: Initialize the solution set $\mathcal{X}^*$ as empty
 - 3: Initialize the search information set $\mathcal{S}$ as empty
 - 4: **for all** $\lambda \in \Lambda$ **do**
 - 5: Formulate a single-objective MILP problem P_λ from MOMILP using λ
 - 6: Retrieve and prepare useful information from $\mathcal{S}$ for solving P_λ
 - 7: Apply branch-and-bound to solve P_λ utilizing the information prepared from $\mathcal{S}$
 - 8: Let $\tilde{x}_\lambda$ be the optimal solution obtained for P_λ ; update $\mathcal{X}^* \leftarrow \mathcal{X}^* \cup \{\tilde{x}_\lambda\}$
 - 9: Update $\mathcal{S}$ with the newly obtained solution and any additional useful information derived during the solving process
 - 10: **end for**
 - 11: **return** $\mathcal{X}^*$
-

上記の疑似コードは、MOBB/D における基本的なフローを簡潔に示している。各単目的問題解法時に $\mathcal{S}$ からの情報を用いることで、探索空間を狭め、計算時間の削減および収束性の向上が可能となる。また、問題解法後に得られた知見を $\mathcal{S}$ に蓄積することで、後続の問題解法時にさらなる効率化が期待できる。これらのプロセスを通じて、多様で質の高いパレート解が効率的に得られることが MOBB/D の特長である。

3.4 探索情報の例: シンプレックス表の再利用

3.4.1 概要

MOBB/D では、分割された部分問題（単目的問題）を厳密に最適化する手法として分枝限定法を用いている。分枝限定法のノード展開・探索過程では、混合整数線形計画問題を線形緩和した問題を繰り返し解くために**シンプレックス法**が適用される。通常、シンプレックス法は単目的問題をシンプレックス表と呼ばれる行列表現に変換し、行基本変形による更新操作を段階的に適用することで最適解を導出する。

しかし、そのままでは得られるシンプレックス表が特定の重みベクトルに対応した実数値行列となり、別の重みベクトルを持つ問題に直接適用することはできない。そこで、本

研究ではシンプレックス表の再利用手法として、**表の成分を重みベクトル λ の関数として表現する**アプローチを提案する。以下では、重みベクトル依存のシンプレックス表の構造および最適性の判定方法、近傍の重みベクトルとの関係、さらに適用範囲と課題について述べる。

重みベクトル依存のシンプレックス表

(3.1) 式に示すように、単目的問題の目的関数や制約係数をはじめとするシンプレックス表の各成分を、重みベクトル λ の線形結合、あるいはその他の関数形式として記述する。具体的には、重みベクトルに対応する各係数が式の中で変数扱いされ、任意の λ を代入すれば通常のシンプレックス表が再現できるように工夫する。このように表を重みベクトルの関数として管理しておくことで、異なる λ に差し替えるだけで同一の表構造を再利用しつつ、新たな単目的問題に応じたシンプレックス法の初期状態を得ることが可能となる。

$$\begin{array}{c|ccc}
 z & \mathbf{x}_B & \mathbf{x}_N & \text{const.} \\
 \hline
 1 & -\mathbf{c}_B^\top(\lambda) & -\mathbf{c}_N^\top(\lambda) & 0 \\
 0 & \mathbf{A}_B(\lambda) & \mathbf{A}_N(\lambda) & \mathbf{b}(\lambda)
 \end{array} \tag{3.1}$$

最適性の判定と近傍の重みベクトル

通常のシンプレックス法では、目的関数係数の符号などをもとに基底の最適性を判定する。本手法でも、更新操作（行基本変形）の手順は変わらないが、表の各成分が λ の関数であるため、判定時には現在着目している重みベクトルを代入して係数を実数値に変換し、その符号や大きさを確認する。具体的な判定方法に関しては、2.2.2 節を参照されたい。

一度ある重みベクトル λ で最適な基底を得た場合、近傍の重みベクトル λ' に対しても、その同一の表構造がそのまま最適解を与えることがある。この場合、一切の更新操作を行わずに解が得られ、探索コストを大きく削減できる。更新操作が必要な場合でも、最適基底解を初期解として利用できるため、ゼロベースでのシンプレックス法に比べ少ない反復回数で新たな最適解へ到達する可能性が高い。

適用範囲と課題

重みベクトルを目的関数にのみ導入する **Weighted Sum** によるスカラー化では、目的関数の係数が λ の線形結合となるため、シンプレックス表が常に線形の形を保つ。一方、Tchebycheff や PBI など他のスカラー化関数では、2.3.4 節で述べたように制約条件にも重みベクトルが登場する。この場合、更新操作を繰り返すうちに分数関数や高次の多項式関数が登場し、表の成分が肥大化・複雑化してしまう恐れがある。したがって、シンプレックス表の再利用手法を効率的かつ安定的に運用するためには、**Weighted Sum** を採用する

ことが推奨される。

以上のように、シンプレックス表を重みベクトルの関数として保持することで、一度構築した表を用いて複数の重みベクトルにわたる問題を効率的に解くことができる。特に、重みベクトルを段階的に変化させながら最適化を進める場合、シンプレックス表を切り替えるコストが大きく削減されるため、MOBB/D 全体の計算効率向上に大きく寄与する。

3.4.2 数値実験

本節では、提案手法 MOBB/D におけるシンプレックス表の再利用手法の有効性を 2 つの数値実験によって検証する。実験 1 では、提案手法 (MOBB/D) と従来手法 (多目的分枝限定法; MOBB) の性能比較を行う。実験 2 では、近傍情報の有無による探索性能の違いを検証する。

実験 1. 従来手法と提案手法の比較

本節では、提案手法 MOBB/D と従来手法 MOBB を比較し、以下を検証するための実験を実施した：

- 提案手法がパレートフロントに均一に近い解を効率的に発見できるか
- 提案手法が実用的な解を迅速に発見できるか

したがって、比較項目は実行時間および HyperVolume 値である。HyperVolume は、得られた解集合に含まれる各解について、その解が支配する領域の和集合の体積を計算したものであり、得られた解集合の多様性と収束性を示す指標である。なお、一般的な多目的最適化においては、パレートフロントへの近接性能を評価する指標として、他にも Generational Distance (GD) やその変種が用いられる。しかしながら、提案手法では任意の重みベクトルに対して最適解が得られるため、理論的に GD は 0 となる。したがって、本論文では GD の比較は行わない。

対象とするベンチマーク問題には、多目的整数計画問題 (MOIP) のベンチマークとして広く知られている多目的ナップサック問題を用いた。多目的ナップサック問題は以下のよう

$$\begin{aligned}
 \min \quad & \left(\sum_{j=1}^m v_j^i x_j \right)_{i=1}^k \\
 \text{s.t.} \quad & \sum_{j=1}^m w_j x_j \leq W \\
 & x_j \in \{0, 1\}.
 \end{aligned} \tag{3.2}$$

本実験では、先行研究⁽¹⁴⁾ に準じた方法でインスタンスを生成した。すなわち、重み w_j および価値 v_j^i は区間 $[1, 1000]$ からランダムに選択され、ナップサックの容量は $W = \lceil 0.5 \sum_{j=1}^m w_j \rceil$ とした。変数の数および目的関数の数が同一の設定で、各問題について 10 個のインスタンスを生成した。

詳細な実験設定は以下の通りである：

- 目的関数の数: 2, 3, 4, 5
- 設計変数の数: 10, 20, 30, 40, 50, 60, 70
- 重みベクトルの数: 500
- 実行時間制限: 1 時間

Table 3.1 には実行時間の結果を、Table 3.2 には HyperVolume 値の結果を示す。

実行時間について、">1hrs"は探索が実行時間制限に到達したことを意味し、 "-"は一部のインスタンスで実行時間制限に到達したため平均値が算出できないことを意味する。

HyperVolume 値については、インスタンス依存性が大きいいため、MOBB との比率で値を示す。比率が 1 未満であれば MOBB が優れており、1 を超える場合は MOBB/D が優れていることを意味する。比率は Table 3.3 に示すように計算され、その最小値、平均値、最大値が Table 3.2 にこの順で示される。HyperVolume 値は探索終了時点で算出される。つまり、実行時間制限に達した場合はその時点で得られた非支配解集合の HyperVolume 値を算出する。

Table 3.1 から、小規模なケースでは従来手法が探索を迅速に終了するが、中規模および大規模なケースでは提案手法が探索を迅速に終了することがわかる。特に、提案手法は目的関数や変数の増加に対して強く影響を受けない。これらの結果は、実験設定において重みベクトルの数を 500 に固定して得られたものであることに留意されたい。

Table 3.2 から、大規模問題では提案手法のほうが HyperVolume が優れていることがわかる。小規模問題では、従来手法が最適な非支配解集合を得ているため、提案手法は同等かそれ以下の品質しか達成できない。しかし、提案手法も最大値に近い HyperVolume 値を導出できていることがわかる。

これらの結果から、提案手法は短時間で高品質な非支配解集合を得ることができる。

Table 3.1 The result of execution times (millisecond) in each problem set.

#Variables	Applied method	#Objectives											
		2			3			4			5		
		min	ave	max	min	ave	max	min	ave	max	min	ave	max
10	MOBB	10	103.8	230	89	169.0	396	167	327.0	525	124	349.7	592
	MOBBD	906	2961.7	6964	979	2782.1	4929	1119	3526.0	5991	613	2121.5	3482
20	MOBB	527	1584.6	3733	1481	5610.5	13820	4265	14382.9	25338	11180	26819.0	53002
	MOBBD	4540	8115.5	12454	2312	8094.3	13686	2190	7807.8	17769	3103	7886.4	19091
30	MOBB	9469	23766.4	72151	19714	170385.2	468034	70830	272578.5	732893	141478	1139039.2	3183731
	MOBBD	5610	10817.3	26371	6325	19006.3	41706	6337	13026.7	21433	9217	17744.6	28755
40	MOBB	24741	164048.4	580123	577436	1586784.9	2528810	1852554	-	> lhrs	2321836	-	> lhrs
	MOBBD	11762	19942.1	35864	16308	21662.7	31536	19081	30876.8	55620	10023	27127.7	49623
50	MOBB	61043	858274.3	3342246	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs
	MOBBD	13157	23285.3	28461	23402	33912.9	62813	28472	39137.2	57944	22877	37598.9	59222
60	MOBB	1743324	-	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs
	MOBBD	23602	37477.8	55856	29741	41034.6	60448	24407	42201.7	56931	35765	51163.5	67676
70	MOBB	3168405	-	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs	> lhrs
	MOBBD	25941	46666.0	78335	40513	54302.3	89747	40991	66928.4	102493	32631	69893.7	140628

Table 3.2 The result of ratio of hypervolume values in each problem set.

#Variables	#Objectives											
	2			3			4			5		
	min	ave	max	min	ave	max	min	ave	max	min	ave	max
10	0.9880	0.9967	1.0000	0.9647	0.9908	0.9995	0.9726	0.9930	1.0000	0.9829	0.9910	0.9997
20	0.9970	0.9986	0.9998	0.9932	0.9968	0.9987	0.9824	0.9900	0.9983	0.9563	0.9748	0.9883
30	0.9949	0.9982	0.9994	0.9922	0.9954	0.9995	0.9833	0.9905	0.9964	0.9584	0.9733	0.9849
40	0.9982	0.9990	0.9996	0.9945	0.9956	0.9971	0.9928	1.0276	1.1239	0.9761	1.0919	1.2376
50	0.9975	0.9988	0.9994	1.0172	1.1377	1.3754	1.0429	1.2661	1.4647	1.1433	1.3891	1.7779
60	0.9989	1.0587	1.1820	1.0815	1.2103	1.4484	1.0986	1.3710	1.5944	1.3373	1.8158	2.1482
70	0.9992	1.1032	1.2044	1.2158	1.3336	1.7856	1.3221	1.5655	1.9308	1.4438	1.9160	2.3821

Fig. 3.2 および Fig. 3.3 は、異なる規模の問題に対する各タイムスタンプ時点での HyperVolume 値の推移を示している。なお、提案手法 MOBB/D が探索を早期に終了した場合、取得された HyperVolume 値は明確化のため点線で示している。"MOBB/D が探索を終了"とは、重みベクトルの数と同じ数のパレート最適解を得たことを意味する。

Fig. 3.2 から、小規模問題では提案手法の HyperVolume 値が初期段階で十分大きくなるが、最終的には従来手法がそれを上回る。この結果は、MOBB がすべてのパレート最適解を取得した一方で、MOBB/D は重みベクトルの数に相当する限定的な数のパレート最適解しか得られなかったことを示している。一方、Fig. 3.3 から、大規模問題では、従来手法が大きな HyperVolume 値を達成するのに非常に長時間を要するため、最終的には提案手法が優れていることがわかる。

Table 3.3 An example of calculating the "ratio" of hypervolume value ($k = 2, m = 60$).

Instance ID	MOBB	MOBBD	ratio
0	4.6828×10^8	5.5350×10^8	1.1820
1	5.7303×10^8	5.9221×10^8	1.0335
2	5.2470×10^8	5.5555×10^8	1.0588
3	6.7163×10^8	6.7089×10^8	0.9989
4	6.2102×10^8	6.2055×10^8	0.9992
5	5.6117×10^8	6.3275×10^8	1.1276
6	5.3710×10^8	5.6467×10^8	1.0513
7	5.5044×10^8	5.5020×10^8	0.9995
8	6.0439×10^8	6.1990×10^8	1.0257
9	4.1246×10^8	4.5799×10^8	1.1104

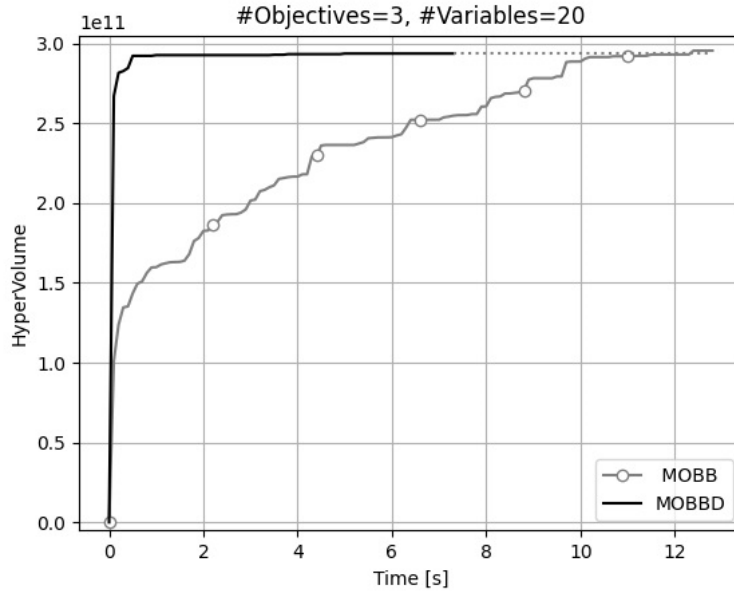


Fig. 3.2 Transition of hypervolume values at each timestamp for a small-scale problem.

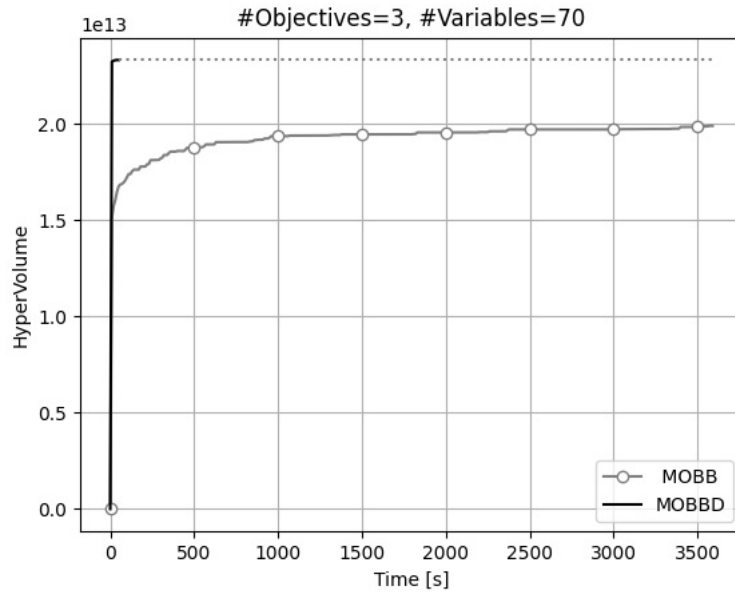


Fig. 3.3 Transition of hypervolume values at each timestamp for a large-scale problem.

実験 2. 近傍情報の有無による比較実験

本節では、以下を検証するための数値実験を実施した：

- MOBB/D を用いて現実的な時間で解ける問題の規模を明らかにする.
- 近傍情報を利用することで探索効率がどの程度向上するかを明らかにする.

本実験では、近傍情報を利用する場合と利用しない場合について、シンプレックス表の

更新回数と実行時間を比較した。なお、これらの手法で得られる解集合は同一であるため、解集合の HyperVolume 値の比較は行っていない。

問題セットは、先行研究⁽¹⁵⁾の“4.1 Randomly Generated Problems”節に基づいて生成した。これは、特に多目的混合整数線形計画問題 (MOMILP) で多数の目的関数を持つ場合に一般的に利用可能なベンチマーク問題が少ないためである。

詳細な実験設定は以下の通りである：

- 同一設定の問題を回数: 10
- 探索情報の利用：有／無
- 目的関数の数: 2, 3, 4, 5
- (制約条件の数) = (設計変数の数) = 10, 20, 30, 40, 50
- (連続変数の数) = (バイナリ変数の数)
- 重みベクトルの数: 5000
- 各変数の係数および定数の値：
 - 目的関数における実数変数の係数を $[-10, 10]$ の範囲から一様にランダムに選択
 - 目的関数におけるバイナリ変数の係数を $[-200, 200]$ の範囲から一様にランダムに選択
 - 制約条件における定数ベクトルの値を $[-100, -50]$ の範囲から一様にランダムに選択
 - 制約条件における変数の係数の値を $[-20, 1]$ の範囲から選択
 - 制約行列の密度は 25%（すなわち、25% の要素が非ゼロであり、75% の要素がゼロ）。

Table 3.4 にはシンプレックス表の更新回数を、Table 3.5 には実行時間の実験結果を示す。表から、すべてのケースで探索性能が向上していることがわかる。特に、目的関数の数が多いほど改善率が大きいことが示されている。

これらの結果から、近傍情報の利用が提案手法 MOBB/D の探索効率を向上させることが明らかになった。特に、目的関数の数が増加するにつれて、近傍情報を利用した手法の優位性が顕著になることが確認された。これは、高次元の多目的最適化問題において、近傍情報の利用が探索空間を効率的に絞り込むことに寄与している可能性を示唆している。

また、近傍情報を利用することで、問題の規模が大きくなるほど、探索の収束速度が向上する傾向も見られた。これは、複雑な問題において、近傍情報が探索の効率化において重要な役割を果たすことを示唆している。

Table 3.4 The result of average of number of update simplex table in each number of objectives.

#Variables		Reuse Tables		#Objectives									
				2			3			4			5
		min	ave	max	min	ave	max	min	ave	max	min	ave	max
10	Yes	3	35	78	525	2110	3237	649	4409	9021	1029	3114	7896
	No	13	4115	22449	849	4630	14863	3607	6433	11545	6467	11090	20863
20	Yes	63	146	285	5493	7127	10181	4177	28445	51391	13778	37135	91879
	No	3827	39222	115210	5700	25534	62714	11097	32727	102116	31617	55884	102120
30	Yes	250	462	765	9414	19194	27470	46999	86961	129682	81691	154751	204767
	No	89829	250250	706765	87821	151183	265211	95472	188243	329125	92991	218826	369609
40	Yes	523	945	1353	25466	45517	64947	81637	127730	195941	126325	311900	515642
	No	114671	538586	912993	194016	446806	653689	303015	466116	783145	163268	637272	1327463
50	Yes	960	2107	3138	52927	72576	91494	186445	272369	357548	356285	661357	858884
	No	510212	1467020	3237479	571082	1397150	2089634	538544	1239960	1939673	821458	1469280	2617848

3.4.3 まとめ

本章では、多目的混合整数線形計画問題 (MOMILP) に対する新たな解法である「Multi-Objective Branch-and-Bound based on Decomposition」(MOBB/D) を提案し、その設計思想、アルゴリズム、および探索情報の例としてシンプレックス表の再利用手法について詳細に述べた。また、提案手法の有効性を数値実験により検証した。

特に、従来手法である多目的分枝限定法 (MOBB) との比較実験を通じて、以下の点が明らかとなった。

- ベンチマーク問題において、高品質なパレートフロントを効率的に構築可能であること。
- 重みベクトルによる問題の分割と探索情報の再利用により、計算効率が大幅に向上すること。
- 高次元の目的関数や大規模な設計変数に対して特に有効な性能を示すこと。

これらの結果は、提案手法 MOBB/D がベンチマーク問題のような基本的な多目的最適化問題において、既存手法と比較して計算効率および解の品質の両面で優れた性能を有することを示している。

次章では、提案手法の応用例として室蘭市のごみ収集問題を取り上げる。この現実問題において、MOBB/D がどのように応用され、効率的かつ実用的な解を得ることが可能であるかを検討する。さらに、現実問題への適用を通じて、提案手法の有用性と拡張性について論じる。

第 4 章

室蘭市のごみ収集問題

4.1 はじめに

本研究では，室蘭市役所環境課との共同研究を通じて得られた実データを活用し，室蘭市のごみ収集問題に対する数理最適化手法の適用を検討する．室蘭市は近年，人口減少や高齢化が進行する一方でごみステーション数が増加するという深刻な状況に直面しており，収集事業が今後破綻する可能性が指摘されている．Fig. 4.1 に示すように，室蘭市内には多数のごみステーションが点在しており，これらを効率的に収集するための最適な計画が求められている．こうした社会背景のもと，ごみステーションの集約や経路の再編成など，収集業務の抜本的な見直しが求められている．

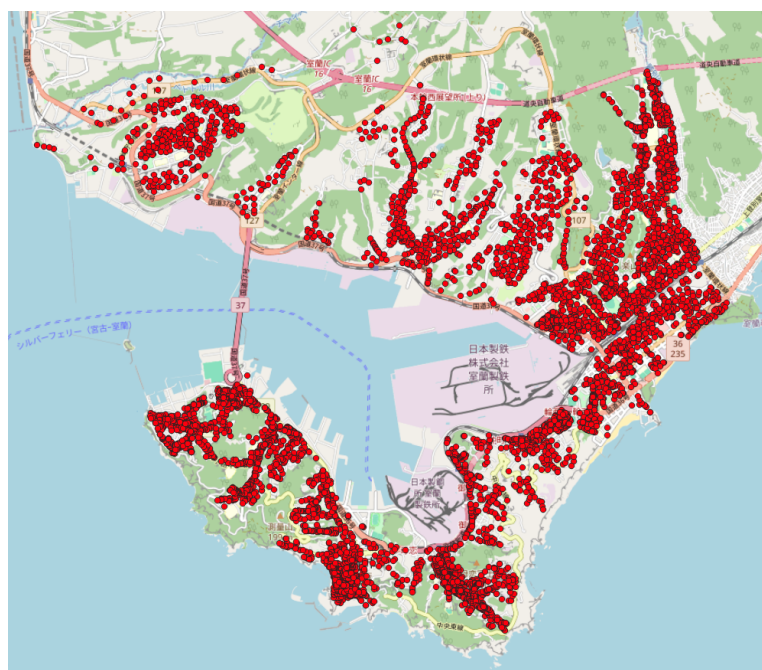


Fig. 4.1 Plot of Garbage Stations in Muroran City

しかし、ごみステーションを集約することによって単純にステーションの数を削減できたとしても、収集コストが必ずしも低減するとは限らない。また、住民にとってはごみ出しに要する距離が増加する懸念があり、安易な集約は受容されにくい。このように複数の利害が関わる意思決定を行うためには、**Evidence-Based Policy Making (EBPM)** の考え方にに基づき、根拠となる定量的な分析結果を提示する必要がある。

EBPM では、しばしば **Agent-based modeling** により個々の住民や事業者の行動を再現し、そこから得られるマクロ的效果を評価するアプローチが採られる^(16,17)。一方で、本研究では数理最適化の枠組みに着目し、ごみ収集問題を数学的に定式化して最適解を導出することで、施策決定の論拠となる定量的なデータの取得を目指す。具体的には、ごみの量といった不確実データが変化した際のルート長がどのように変化するかを分析することで、より効率的な収集戦略を考えることができる。また、ごみステーションの配置と収集経路の計画を同時に最適化することで、住民負担と収集業者負担の両側面を考慮に入れた総合的な評価を実施する。こうした取り組みは、収集業務の効率化を実現するとともに、行政と住民の合意形成に向けた具体的な根拠として活用できることが期待される。

本章の以下の構成としては、まず 4.2 節で経路計画問題一般の性質と部分巡回路除去制約の概要を説明する。続いて、4.3 節ではごみ収集問題を単目的最適化問題としてモデル化し、分枝カット法を用いた解法の妥当性を検証する。次に、4.4 節では多目的最適化問題としてモデル化し、ごみステーションの集約と経路計画を同時に最適化する際にどのように探索情報を活用できるかを示す。最後に、第 4.5 節で本章全体をまとめ、今後の課題と発展可能性について議論する。このように本章では、室蘭市のごみ収集問題における課題を単目的および多目的の両面から捉え、最適化手法が施策立案のための有力な手段となり得ることを示すことを目的としている。

4.2 経路計画問題

本節では、ごみ収集問題と密接に関連する**経路計画問題**について概説する。経路計画問題は、地図上に散在する拠点（頂点）を巡回しながら移動コストを最小化するようなルート（または複数のルート）を求める問題の総称であり、代表的な例として**巡回セールスマン問題**（Traveling Salesman Problem; TSP）や**配送経路問題**（Vehicle Routing Problem; VRP）が挙げられる。ごみ収集問題は、市内に分散するごみステーションを訪問しながら廃棄物を回収する点で、経路計画問題の一形態とみなすことができる。

4.2.1 巡回セールスマン問題 (TSP)

TSP は, n 個の頂点 (都市) をちょうど 1 回ずつ訪問し, 出発点に戻る閉路を得るようにルートを計画する問題である. 頂点の集合を $\{1, 2, \dots, n\}$ とし, 各頂点間に設定された移動コスト (または距離) を c_{ij} ($i, j = 1, \dots, n$) とする. TSP の典型的な定式化は以下のような混合整数線形計画問題 (MILP) として与えられる⁽¹⁸⁾. なお, 以下の定式化は後述する部分巡回路除去制約を含まない基本形であることに留意されたい.

$$\min. \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (4.1)$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{ij} = 1 \quad (i = 1, \dots, n) \quad (4.2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad (j = 1, \dots, n) \quad (4.3)$$

$$x_{ii} = 0 \quad (i = 1, \dots, n) \quad (4.4)$$

$$x_{ij} \in \{0, 1\} \quad (i, j = 1, \dots, n, i \neq j) \quad (4.5)$$

ここで, 決定変数 x_{ij} は

$$x_{ij} = \begin{cases} 1 & \text{頂点 } i \text{ から頂点 } j \text{ へ移動する場合,} \\ 0 & \text{それ以外の場合} \end{cases} \quad (4.6)$$

を表す. 目的関数は総移動コスト (距離) の最小化を意味し, 上記の 2 種類の総和制約は「各頂点に 1 回入って 1 回出る」ことを保証する.

部分巡回路除去制約 (Subtour Elimination Constraints)

TSP の解として望まれるのは, すべての頂点を含む 1 つの巡回路である. ところが, 上記の制約だけでは**部分巡回路** (部分巡回路, subtour) と呼ばれる独立したループが複数生じる解が許される可能性がある. すなわち, 一部の頂点のみを含む閉路が解中に含まれてはならない. これを排除するためには, 以下の**部分巡回路除去制約**が不可欠である.

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad (\forall S \subset \{1, \dots, n\}, 2 \leq |S| \leq n - 1). \quad (4.7)$$

ここで, S は頂点集合 $\{1, \dots, n\}$ の真部分集合であり, x_{ij} が S 内だけで閉路を作らないよう制限している. この形式を **DFJ 制約** (Dantzig–Fulkerson–Johnson 制約) と呼ぶ⁽¹⁸⁾. ただし, S の取り得る組合せは指数個に及ぶため, 通常は 2.2.3 節の遅延制約の項で述べたように, まず部分巡回路除去制約を削除してから最適化を実行し, 得られた解に部分巡回路が検出された時点で対応する制約を動的に追加していくことを繰り返す方法が用いられる.

MTZ 制約 (Miller–Tucker–Zemlin 制約)

一方, **MTZ 制約**⁽¹⁹⁾ では, 頂点 1 を基準として $u_1 = 0$ とし, 頂点 i を通過する順序を示す補助変数 u_i ($2 \leq i \leq n$) を導入することで部分巡回路除去を実現する. 具体的には「 $x_{ij} = 1$ ならば $u_j \geq u_i + 1$ 」という関係式を課す. ただし, 一般に **MTZ 制約は線形緩和最適解が元の実行可能解と大きく乖離するため, 下界値としての精度が低いことが知られており**⁽²⁰⁾, 大規模問題では解探索を著しく遅延させる場合が多い.

したがって, 本研究では, 部分巡回路除去制約の手段として **DFJ 制約**と**切除平面法**の組み合わせを主に用いる.

4.2.2 配送経路問題 (VRP)

TSP をさらに発展させた問題が**配送経路問題 (VRP)**である. VRP では, 1 つのデポ (拠点) と複数台の車両が存在し, 各顧客 (頂点) を一括して訪問する経路を求める必要がある. ここでは, すべての車両が同一の積載量 Q を持ち, 最大で m 台まで利用できると仮定する. 頂点集合を $\{0, 1, 2, \dots, n\}$ とし, 頂点 0 をデポ, $1, \dots, n$ を顧客 (またはごみステーション) とする. 各顧客 i の需要量 (ごみ量) を q_i とし, 決定変数 x_{ij} は

$$x_{ij} = \begin{cases} 1, & \text{頂点 } i \text{ から頂点 } j \text{ へ移動する経路を選択する場合,} \\ 0, & \text{それ以外の場合} \end{cases} \quad (i \neq j, i, j = 0, \dots, n) \quad (4.8)$$

とする. VRP の目的関数は走行コスト (距離) の総和を最小化するという TSP と同様の形式を持つ. さらに, 容量制約を **Rounded Capacity Inequalities (RCI)** によって課すことで, 大規模な VRP に対しても部分巡回路除去制約と同様に**切除平面法**との相性が良い定式化を得ることができる. 具体的には, 以下のように定式化される⁽²¹⁾.

$$\min. \quad \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \quad (4.9)$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{0j} = \sum_{i=1}^n x_{i0} = m \quad (4.10)$$

$$\sum_{j=0, j \neq i}^n x_{ij} = \sum_{j=0, j \neq i}^n x_{ji} = 1, \quad i = 1, \dots, n \quad (4.11)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S), \quad \forall S \subset \{1, \dots, n\}, 2 \leq |S| \leq n-1 \quad (4.12)$$

$$x_{ij} \in \{0, 1\}, \quad i \neq j, i, j = 0, \dots, n \quad (4.13)$$

ここで, (4.10) 式は, デポから出発するルートと戻るルートの本数を合わせて m 本にすることを意味し, (4.11) 式は, 各頂点 i をちょうど 1 度訪れることを強制している.

(4.12) 式は Rounded Capacity Inequalities (RCI) であり、各ルートの積載量が Q を超過しないよう制限する。 S は頂点集合 $\{1, \dots, n\}$ の真部分集合であり、 $r(S)$ は S 内の頂点を訪問するのに必要な車両台数を表す。ただし厳密な車両台数を求めるには、別途ビンパッキング問題 (Bin Packing Problem) と呼ばれる最適化問題を解く必要が生じてしまう。そのため、一般には緩和された以下の丸め込まれた (Rounded) 形式で定式化される。

$$r(S) = \left\lceil \frac{\sum_{i \in S} q_i}{Q} \right\rceil \quad (4.14)$$

これにより、 S だけで独立した巡回路が形成されることを防ぎつつ、かつ積載量 Q を超過するような需要は必ず複数のルートに分割して訪問しなければならないことを強制できる。この性質は切除平面法で動的に切除平面を生成・追加する際に非常に有効である。

4.2.3 ごみ収集問題との関連性

室蘭市のごみ収集問題は、市内に複数存在するごみステーションを訪問し、かつ車両（ごみ収集車）の積載制限や運転手の労働条件などを考慮しながら最適なルートを計画する点で、上記の TSP や VRP と同様の構造を持つ。さらに、4.4 節ではステーションの集約も考慮した最適化を行うため、**施設配置問題** (Location Problem) の要素も含む複合的な問題となる。いずれの場合も、部分巡回路除去制約や RCI など、部分集合 $S \subseteq \{1, \dots, n\}$ に着目した切除平面を動的に生成する切除平面法が鍵となり、ごみ収集に対する実行可能かつ効率的な解を得るための強力な手法として期待される。

本研究では、まず単目的の経路最適化としてモデル化を行い、部分巡回路除去制約を含む切除平面法による解法の有効性を検証する。そして多目的最適化へと拡張し、業者負担（走行コスト）と住民負担（歩行距離など）を同時に考慮した問題の解決を目指す。

4.3 単目的最適化としてのモデル化

4.3.1 モデル定式化

本節では、室蘭市のごみ収集問題を**単目的最適化問題として定式化**する。本研究は多目的の混合整数線形計画問題 (MOMILP) に対して探索情報を再利用する手法を提案することを最終目標としているが、まずは単目的でも現場に役立つデータが得られるかどうか、および分枝限定法と切除平面法を組み合わせた**分枝カット法**によって効率的に解けるかどうかを検証するために、本節のモデル構築および実験を行う。

本モデルでは、ごみステーションの集約自体にはあまり焦点を当てず、ルールベースに基づいた最低限の集約のみを行った状態で定式化を行う。そのため、問題構造は基本的に

は**経路計画問題**（車両経路問題あるいは巡回セールスマン問題に近い枠組み）として捉えられる。以下では、定式化および使用したデータセットと各種パラメーター設定について述べる。

定式化

本研究で採用する単目的経路計画問題の定式化は、4.2.2 節で述べた VRP の形式と同様である。すなわち、目的関数としてルート総移動距離の最小化を採用し、各種制約（次数制約、部分巡回路除去制約、車両数や容量制約）を組み込んだ混合整数線形計画問題（MILP）として定義される。特に、部分巡回路除去制約には DFJ 制約を用い、切除平面法により動的にこれを追加することで、部分巡回路を排除する。本モデルの定式化の詳細については、4.2.2 節を参照されたい。

データセットとパラメーター

ここでは、室蘭市のごみ収集に関する実データを用いたモデル化のため、以下の点を考慮する。

- **ごみステーション位置データ:** 室蘭市より提供された座標情報と、地理情報データベースである OpenStreetMap⁽²²⁾ による道路距離情報を用いて、頂点間コスト c_{ij} を算出する。
- **ルールベースによる最低限の集約:** 事前に一定距離以内にあるステーションをまとめる簡易的な集約が行われており、実質的には「集約後のステーション集合」を頂点集合 $\{1, \dots, n\}$ として扱う。
- **ごみ量や容量制約:** ごみ量 q_i は室蘭市の清掃業者によって調査された、ある日における各ごみステーションで収集されたごみ袋の数として提供されている。ごみ収集車の容量 Q は従業員による調査に基づいて平均的な値が設定される。

これらのデータを用いて上記の MILP を構築し、分枝カット法による最適化を行うことで、(i) 現行ルートとの比較実験（ルート長の削減度合い）、(ii) ごみ量変動時のルート再計算に関する分析を実施する。これらの実験結果については、次節 4.3.2 以降で詳述する。

以上のように、本研究の単目的モデルでは、室蘭市から提供されたごみステーションに関する情報とルールベースの最低限集約を前提としつつ、経路計画問題として定式化を行った。このモデルの解より得られるルートは、現行のごみ収集業務の効率化を定量的に評価し、EBPM の一環として実証データを提供する上で有用な指標となることが期待される。

4.3.2 分枝カット法による解法

本研究では、2.2.4 節および 2.2.3 節で述べた分枝限定法と切除平面法を組み合わせた**分枝カット法** (branch-and-cut) を用いて、前節 4.3.1 で示した単目的 VRP モデルを解く。すでに分枝限定法や切除平面法の理論については概説しているため、本節では、VRP に特化した切除平面の導入手順を中心に簡潔に述べる。

分枝限定法に切除平面法を組み込む流れ

分枝限定法は、一般に以下の手順で進行する。

1. 初期ノードで問題を線形緩和し、得られた解が整数解であれば最適解を更新。
2. 解が整数解でない場合は、変数の分枝 (branch) により問題を部分問題へ分割し、それぞれを再度緩和して評価。
3. サブノード同士の探索優先度を決めながら、限定 (bound) により不要ノードを刈り取り、探索効率を高める。

上記の緩和問題を解くステップにおいて、部分巡回路除去制約や容量制約などの切除平面 (カット不等式) を動的に追加する仕組みが切除平面法である。具体的には、中間解 (線形緩和解) を得るたびに、その解が部分巡回路を含むあるいは容量制約を破っているなどの制約違反を起こしているのであれば、それを排除する妥当不等式を分離 (separation) し、問題に追加して再度緩和問題を解く。これを繰り返すことで、**実問題に近く強い緩和**を構築し、探索効率を大幅に向上させる。

CVRPSEP ライブラリ

本研究では、切除平面を分離 (生成) するにあたり、**CVRPSEP**⁽²³⁾ という VRP 向けのオープンソースライブラリを使用している。CVRPSEP は、多様な VRP 固有の切除平面不等式を効率的に探索・生成する機能を備えており、本研究の実装にも容易に組み込むことができる。具体的には、容量制約をはじめとした、以下のような VRP に特化した切除平面の分離手法が用意されている。

- Capacity Inequalities
- Framed Capacity Inequalities (FCI)
- Strengthened Comb Inequalities
- Homogeneous Multistar and Partial Multistar Inequalities
- Generalized Large Multistar Inequalities (GLM)
- Hypotour Inequalities

これらの切除平面により、VRP 特有の部分巡回路や不正なフローを効果的に排除し、探索空間を大幅に削減できる。本研究では、**CVRPSEP に実装されている separation 関数**をブラックボックス的に利用し、探索の各ステップで線形緩和解を検証しては、違反解をしばしば排除していく。これにより、単純な部分巡回路除去制約（DFJ 制約）のみならず、VRP 全般に有効なカット不等式を柔軟に導入可能となっている。

なお、これらの切除平面の詳細な数理的背景や分離アルゴリズム（separation 手法）の仕組みについては、CVRPSEP の文献^(23,24)を参照されたい。本研究では、多くの既存研究と同様に、強力な既存ライブラリを活用し、高性能な分枝カット法を実現している。

4.3.3 数値実験

本章では、以下の 2 つの数値実験について説明する。

- 実験 1: 室蘭市の実際のごみ収集ルートと提案手法による最適ルートを比較し、移動距離の削減効果を評価することを目的とする。
- 実験 2: ごみ量の変動した場合のルート長を算出し、その推移を分析することで、本定式化で得られたデータを EBPM に活用する際の有用性を検証する。

これらの実験を通じて、本定式化の有効性と実用性を多角的に評価する。

実験 1: 現在の収集ルートと最適ルートの比較

本実験では、室蘭市で実際に運用されているごみ収集ルート（GPS データに基づくルート）と、本研究で最適化を行って得られたルートとを比較した。既存の GPS ルートは、収集車が定められたステーション群を巡回してデポに戻るまでの経路を示すものである。Fig. 4.2 に示すように、実際のルートは一部に大きな迂回や長い往復経路が含まれている。一方、最適化によって得られたルートは、Fig. 4.3 に示すように、総移動距離を大幅に短縮しつつ、すべてのごみステーションを効率的に訪問するように構築されている。

比較結果

実験では、単一の収集日における GPS ルートのデータ（約 7,020 m）と、提案モデルで求めたルート（約 5,261 m）とを比較した結果、およそ **25 % の距離削減**が得られた。以下に、主な発見をまとめる。

- **走行コストの削減:** 約 25 % の距離短縮は、燃料消費量の削減や人件費の軽減など、収集業者の運営コストに対しても大きなインパクトを与えられる。
- **実用面での考察:** 提案モデルによるルートは U ターンや狭い道路の通行など、現実



Fig. 4.2 Route of Actual GPS Data



Fig. 4.3 Optimized Route by Proposed Model

の交通制約を必ずしも十分考慮していない場合がある。実運用に当たっては、交通規制や道路幅員などの追加的な制約を盛り込む必要がある。

考察と今後の展望

本結果は、単に移動距離を削減するだけでなく、ごみ収集に要する時間やコストを低減できる可能性を示唆する。一方で、モデルが推奨する経路がすべて実用可能とは限らず、交通事情や物理的制約、さらには安全面の検討が不可欠である。今後は、より現実的な制約（交通規制、車両の走行特性、作業者の勤務条件など）を統合し、最適解の妥当性と実現性を高めることが重要な課題となる。

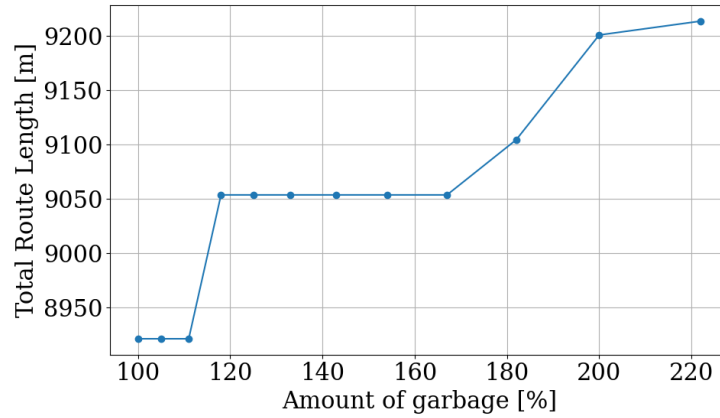


Fig. 4.4 Route Length Transition with Garbage Amount Variation

実験 2: ごみ量変動時のルート長推移

本実験では、ステーションごとのごみ量を変化させ、それに伴うルート長の推移を分析する。現実のごみ収集では、季節やイベントなどの要因によってごみ量が大きく変動することが知られており、それが収集効率にどのような影響を与えるかを考察することは、運用戦略の検討上非常に重要である。

実験内容

ベースラインとなるごみ量を「100%」とし、これを $\alpha \in [0.5, 2.0]$ 倍程度に変化させたときの最適ルートを算出し、ルート長がどのように変動するかを調べた。具体的には、ステーション i における需要量（廃棄物量） q_i を

$$q'_i = \alpha \cdot q_i \quad (4.15)$$

と設定し、 α を増減しながら繰り返し最適化を行う。

結果と考察

Fig. 4.4 は、 α に対する最適化後のルート長の変化をプロットしたものである。

- **ルート長の増加トレンド:** Fig. 4.4 からわかるように、 α が大きくなる（すなわちごみ量が増加する）に伴って、最適ルート長も全体的に増加傾向を示す。しかし、単純な線形関係ではなく、わずかにごみ量が増えただけでルート長が急増するポイントがある一方、ある範囲まではほとんどルート長が変化しない区間も見受けられる。
- **非線形的な影響:** このような非線形的増加は、地理的条件やステーション分布によるところが大きい。ある特定のステーションが容量超過（または仮想的に追加往復が必要となる状態）に近づくと、経路の再編が一気に進むため、ルート長が急に伸

びることがあると考えられる。

- **運用戦略への示唆:** わずかな需要増加でルート長が急増する箇所があるため、収集効率を維持するためには、ごみ量の増加に対して適切な運用戦略を取ることが重要である。

今後の課題

本結果は、ごみ量の増大に対して単純な一律拡大策をとるのではなく、地域特性や分布パターンを踏まえた戦略的な収集計画が重要であることを示唆する。たとえば、ルート長の増加が急激な箇所に対しては、収集回数を増やす、あるいは別の曜日に振り分けるなどの方法を検討できるなどがある。

本実験の知見は、ごみ量が多い時期に単純に車両台数や走行距離を増やすのではなく、最適化によるルート再編と収集スケジュールの調整を組み合わせることで、コスト増を最小限に抑える可能性を示している。

単目的モデル化のまとめ

本節では、室蘭市のごみ収集問題に対して単目的最適化モデルを構築し、分枝カット法を用いた実験を通じてその有効性を検証した。以下に、本実験の主な成果と考察をまとめる。

まず、実験1において提案モデルによる最適化ルートが実際のGPSルートと比較して約25%の距離削減を達成したことは、分枝カット法がごみ収集問題に対して**有効かつ効率的な解法**であることを示している。これは、収集業務の運用コスト削減や作業負荷軽減に直結する重要な成果である。一方で、提案モデルが推奨するルートには現実の交通規制や物理的制約を十分に反映していない部分があるため、実運用に際してはさらなる調整が必要であることが明らかとなった。

続いて、実験2ではごみ量の変動が最適ルート長に与える影響を分析した結果、ごみ量の増加に伴いルート長が非線形的に増加する傾向が確認された。特に、一定のごみ量を超えるとルート長が急増する現象が観察され、これは収集ルートの再編成や収集頻度の調整といった運用戦略の重要性を浮き彫りにした。この結果は、単目的最適化モデルがごみ収集業務の効率化において実用的な指標を提供することを裏付けるものである。

本研究では、単目的モデルを用いることで、まずは基本的な経路最適化の有効性を検証し、分枝カット法の適用可能性を確認した。この段階で得られた成果は、次の4.4節における多目的最適化モデルの構築および検証に向けた基盤となるものである。多目的モデルでは、業者負担（走行距離の最小化）と住民負担（歩行距離の最小化）という複数の評価基準を同時に考慮することで、より現実的かつバランスの取れたごみ収集計画の策定を目指す。

今後の課題としては、現実の運用に即した追加的な制約（交通規制や車両の運行時間制

限など)をモデルに組み込み、さらに多目的最適化の枠組みを拡張することである。これにより、本研究の成果が実際の自治体運営における意思決定支援ツールとしての実用性を高めることが期待される。

次節では、これらの課題に対応するための多目的モデルの詳細な定式化とその解法について述べる。

4.4 多目的最適化としてのモデル化

4.4.1 多目的モデル化

本節では、多目的最適化の枠組みを用いて、ステーション集約と経路計画を同時に扱うモデルを構築する。先に示した単目的モデル(4.2.2 節参照)では、ごみ収集車のルート長のみを最小化する問題を扱ったが、ここでは**住民負担(ステーションまでの移動距離)**も同時に考慮することで、業者と住民の双方にとって望ましい解集合を得ることを目指す。

2 目的の設定

まず、本研究が扱う多目的問題の2つの目的は以下のとおりである。

業者負担の最小化

収集車が全ステーションを巡回するときの**総走行距離**(あるいは時間や燃料消費量など)を最小化する。これは従来のVRPに相当する目的であり、主に収集業者や行政が負担するコストに対応する。

住民負担の最小化

各住民が割り当てられるステーションまで歩行する**増加距離**を最小化する。ステーションを集約するほど業者負担は下がる可能性がある一方で、住民が遠いステーションに行かなければならないというデメリットが生じる。この住民負担をもう一方の目的として組み込み、集約度合いと住民の利便性のトレードオフを分析可能にする。

本研究では、以上の2つの目的を同時に考慮し解探索することを目的とする。具体的には、本節で示す定式化をMOMILP(Multi-Objective Mixed Integer Linear Programming)として構築し、提案手法MOBB/Dを適用して解集合を導出する。

変数と制約の追加

単目的VRPでは、 $\{0, 1, \dots, n\}$ の頂点集合に対して「どのルートを選択するか」のみを決定すればよかった(**経路計画問題**)。一方、多目的モデルでは、**ステーションの集約**(ど

の候補地点を使用し、どの既存ステーションをどの候補地点に割り当てるか) も同時に扱うため、**施設配置問題**的な変数・制約が追加される。

まず、以下に本モデルで用いる記号とパラメーターを示す。

記号とパラメーター

- V_O : 現在のごみ収集所の集合 (頂点)。
- V_C : 統合後のごみ収集所の候補地点の集合 (頂点)。ノード 0 (x_0) は常に使用される出発点 (デポ)。
- K : ごみ収集車の台数。
- M : 使用する候補地点の数。
- c_{ij} : 現在のごみ収集所 $i \in V_O$ から候補地点 $j \in V_C$ への距離。
- d_{ij} : 候補地点 $i, j \in V_C$ 間の距離。
- θ : 2 つの目的の間の重み付けパラメーター ($\theta = 1$ に近いほど住民の負担を強調)。

これらを用いて、以下に、主要な変数と制約を示す。

(1) 候補地点と割り当て変数

- x_j : 候補ステーション (施設) j を使用するかどうかを示す 2 値変数。

$$x_j = \begin{cases} 1, & \text{候補地点 } j \text{ を使用する場合} \\ 0, & \text{それ以外} \end{cases}$$

- y_{ij} : 既存ステーション i が、候補地点 j へ割り当てられるかどうかを示す 2 値変数。

$$y_{ij} = \begin{cases} 1, & \text{既存ステーション } i \text{ が候補地点 } j \text{ に集約される場合} \\ 0, & \text{それ以外} \end{cases}$$

各ステーション i が必ず 1 つの候補地点 j に割り当てられるように、以下の制約

$$\sum_{j \in V_C} y_{ij} = 1 \quad (\forall i \in V_O),$$

かつ、候補地点 j が選択されたときのみ割り当て可能であることを指す以下の制約

$$y_{ij} \leq x_j, \quad \forall i \in V_O, j \in V_C$$

を課す。

(2) 経路選択変数

本研究では、選択された $\{x_j = 1\}$ 候補地点の集合を頂点とみなし、それに対して

VRP の枠組みで**走行経路**を決定する．すなわち， $\{0\} \cup \{j \mid x_j = 1\}$ を頂点集合とし，分枝カット法等で示した VRP の決定変数

$$e_{ij} = \begin{cases} 1, & \text{候補地点 } i \text{ から } j \text{ への経路を採用,} \\ 0, & \text{それ以外} \end{cases}$$

を用いる．部分巡回路除去制約 (DFJ 制約) や車両台数 K との整合性などは，単目的 VRP の場合と同様に設定する (詳細は 4.2.2 節を参照)．

(3) 使用候補数の制限

市の方針等によって「どの程度ステーションを集約できるか」に上限 (または下限) がある場合，

$$\sum_{j \in V_C} x_j = M$$

のように，開設可能な候補地点の総数を M 個に制限することも考慮できる．こうしたパラメーター M を変化させることで，集約度合いと住民負担のトレードオフを詳細に検討することが可能となる．

MOMILP としての定式化

以上の変数と制約を組み合わせると，以下のような **2 目的混合整数線形計画問題 (MOMILP)** が得られる．

$$\min. \quad \theta f_1(\mathbf{x}, \mathbf{y}, \mathbf{e}) + (1 - \theta) f_2(\mathbf{x}, \mathbf{y}, \mathbf{e}) \quad (4.16)$$

$$\text{s.t.} \quad \sum_{j \in V_C} x_j = M \quad (4.17)$$

$$\sum_{j \in V_C} y_{ij} = 1, \quad \forall i \in V_O \quad (4.18)$$

$$y_{ij} \leq x_j, \quad \forall i \in V_O, j \in V_C \quad (4.19)$$

$$\sum_{j=1}^n e_{0j} = \sum_{j=1}^n e_{j0} = K \quad (4.20)$$

$$\sum_{i \in S} \sum_{j \in S} e_{ij} \leq |S| - r(S), \quad \forall S \subset \{1, \dots, n\}, 2 \leq |S| \leq n - 1 \quad (4.21)$$

$$x_j, y_{ij}, e_{ij} \in \{0, 1\}, \quad (4.22)$$

ここで，2 つの目的関数 f_1, f_2 は以下のように定義する．

$f_1(\cdot)$: 住民負担 (歩行距離) の総和

$$f_1(\mathbf{x}, \mathbf{y}, \mathbf{e}) = \sum_{i \in V_O} \sum_{j \in V_C} c_{ij} y_{ij}.$$

c_{ij} は既存ステーション i から候補地点 j までの距離を表す．住民がその距離を歩くと想定することで，ステーション集約の影響を評価する．

$f_2(\cdot)$: 業者負担（ルート長）の総和

$$f_2(x, y, e) = \sum_{i \in V_C} \sum_{j \in V_C} d_{ij} e_{ij}.$$

d_{ij} は候補地点 i と j 間の距離であり、収集車が $e_{ij} = 1$ の辺を通過するときの移動距離を積算する。

この2つの目的を同時に最適化する MOMILP を解くことで、「住民負担は少ないが業者負担の大きい解」から「業者負担は少ないが住民負担の大きい解」まで、さまざまなトレードオフ解を列挙できる。その中から、行政と住民が協議しながら受容可能な案を選択するという枠組みが多目的最適化の利点である。

本研究では、提案手法である MOBB/D を用いて、この MOMILP を分割ベースで効率的に探索する。その際、各部分問題で生成された切除平面（特に部分巡回路除去制約や容量関連の切除平面）を他の部分問題へ再利用する仕組みを導入し、全体の計算時間を短縮することを狙う。詳細は後述の 4.4.2 節にて述べる。

4.4.2 切除平面共有の仕組み

分割ベースの多目的最適化手法では、重みベクトルが変化するたびに異なる部分問題を解く必要がある。本研究が提案する MOBB/D では、これらの部分問題を連続的に解いていく過程で、問題構造に強く依存する切除平面（カット）を再利用する仕組みを導入し、探索効率の向上を狙う。

ここで重要なのは、部分巡回路除去制約などの問題構造に根ざした不等式は、重みベクトルが変化しても不変であるという点である。すなわち、ある部分問題を解いている最中に発見された部分巡回路や容量超過を除去する切除平面は、他の重みベクトルに対する部分問題においても同様に有効である可能性が高い。

たとえば、 S をステーションの部分集合としたときに導入される部分巡回路除去制約は、目的関数の重み付け如何にかかわらず、「 S 内に独立した巡回路を形成してはならない」という意味を持つ。この論理的関係は重みベクトルが変わっても変化しないため、すでに検出済みの違反巡回路を除外する切除平面を次の部分問題でも活用することができる。

本研究の MOBB/D では、単目的の分枝限定法や分枝カット法と同様の手続きを、重みベクトルを変えながら反復的に実行する際に、切除平面情報をグローバルに記憶・再利用する仕組みを組み込んでいる。特に、ごみ収集問題で重要となる部分巡回路除去制約は、多くの部分問題で共通して必要となるため、ある1つの部分問題で見つかった巡回路の切除平面は、他の部分問題でも高い確率で再利用可能であると考えられる。

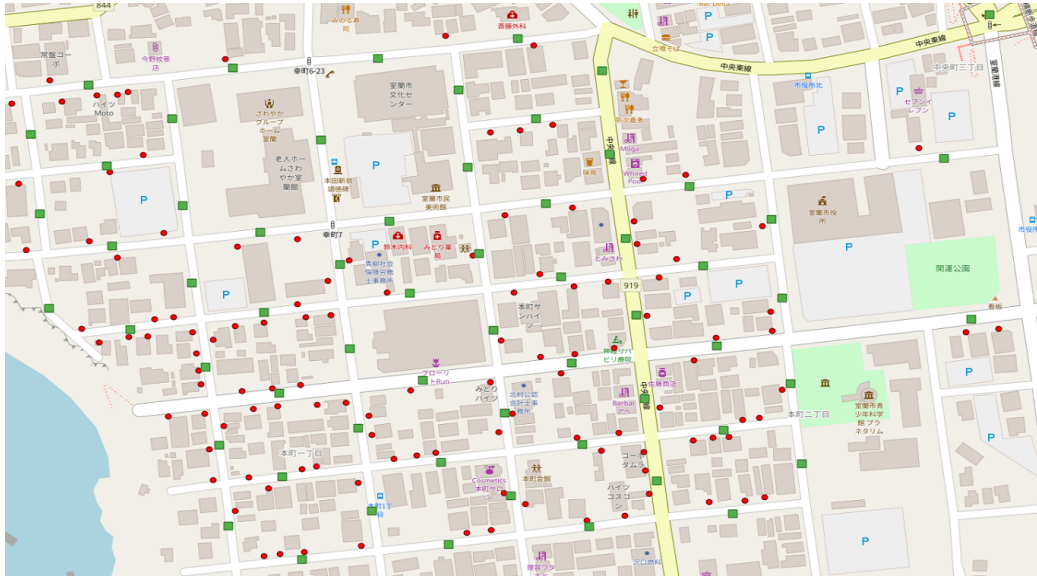


Fig. 4.5 Target area of experiment for Multi-Objective Garbage Collection Problem

4.4.3 数値実験

本節では、前節 4.4.1 で示した多目的混合整数線形計画問題 (MOMILP) を実際に解き、得られた数値結果を示す。まず、実験の設定と対象領域を示し、続いて切除平面の共有有無による探索効率の違いを比較し、最後に代表的な解の可視化例を提示する。

実験設定

室蘭市の一部地域を対象とし、既存ステーションの集合 V_O と候補地点の集合 V_C をそれぞれ準備した。今回の実験では、既存ステーション数を適度に絞り込むことで問題規模を管理しやすくしつつ、2 目的最適化 (住民負担最小化と業者負担最小化) の挙動を把握することを目指す。Fig. 4.5 に本実験で用いる領域の概略を示す。赤点が既存ステーション、緑四角が候補地点を表している。

切除平面共有の有無による探索効率比較

提案手法 MOBB/D では、部分巡回路除去制約などの切除平面を各部分問題間で共有する仕組みを導入している。この共有が探索効率に及ぼす影響を定量的に評価するため、切除平面共有なしの場合と共有ありの場合の実行時間を比較した。

重みベクトル θ を複数パターンに渡って変更しながらパレート解集合を構築した。Table 4.1 に結果を示す。

この結果から、切除平面を共有することによって約 10% 程度の計算時間削減が得られることがわかる。特に、問題構造が複雑になるほど (ステーション数や候補数が増加するほ

Table 4.1 Comparison of execution time with/without cut sharing

	Without Sharing	With Sharing
Execution Time (sec)	29796	26998

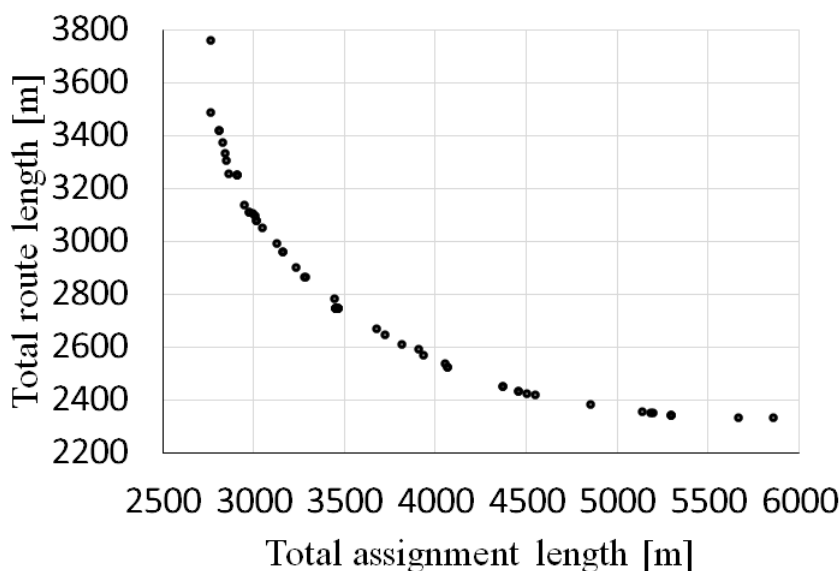


Fig. 4.6 Pareto solutions for Multi-Objective Garbage Collection Problem

ど) この効果は大きくなる傾向がある。以上より、探索情報として切除平面を共有する仕組みが、多目的最適化全体の計算コストを下げる上で有効であることが示唆される。

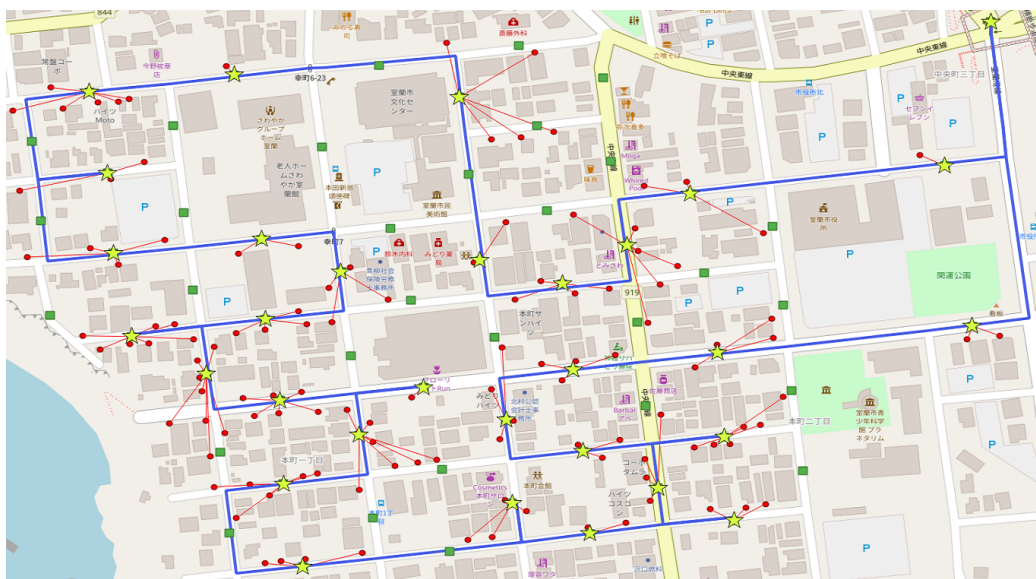
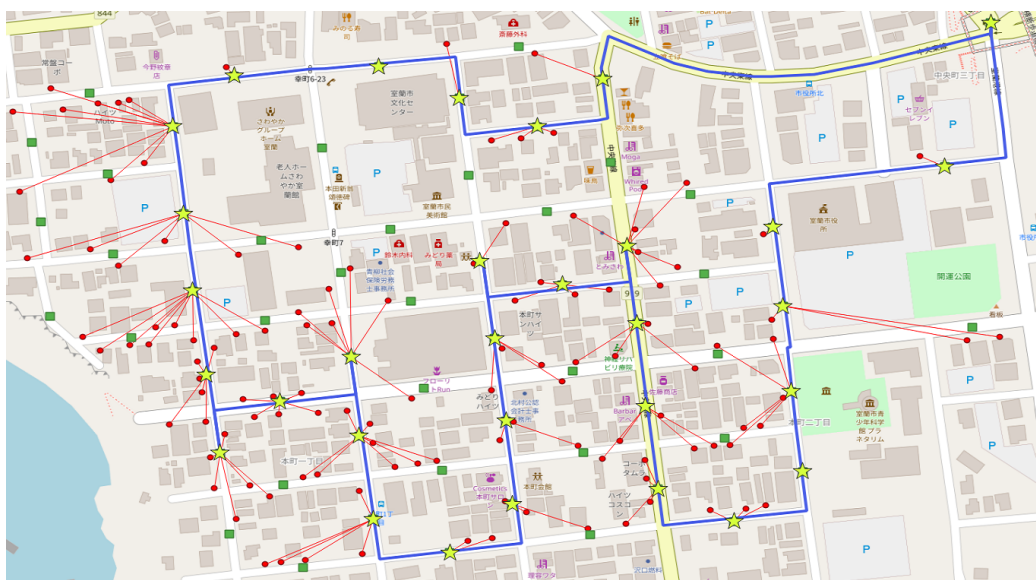
解集合の可視化

Fig. 4.6 に、複数の重みベクトル θ に対し得られたパレート解集合をプロットした例を示す。縦軸に業者負担（収集車の総ルート長）、横軸に住民負担（ステーションまでの歩行距離の総和）をとっており、左上から右下に向かってトレードオフ関係を示す点群が得られている。

さらに、この解集合から2点をピックアップし、住民優先 ($\theta = 0.9$) と業者優先 ($\theta = 0.1$) のそれぞれで得られたルートを可視化した結果を Fig. 4.7, Fig. 4.8 に示す。

Fig. 4.7 では、住民負担を重視するためにステーションの配置が分散傾向となり、ごみステーションへの歩行距離が短くなっている。一方で、収集車はより多くの地点を巡回する必要があるため、ルート長がやや大きくなる。

これに対し、Fig. 4.8 では、業者負担を低減するためにステーションを集約している傾向が見られ、ルート長が明らかに短くなっている。しかし、住民側からするとステーションまでの距離が長くなりやすいため、ある程度の負担増加を受け入れる必要がある。

Fig. 4.7 Route and assignment for residents ($\theta = 0.9$)Fig. 4.8 Route and assignment for workers ($\theta = 0.1$)

このように、多目的最適化の枠組みを用いることで、住民負担と業者負担のバランスを可視化でき、ステーション集約の最適な度合いを検討する上で有用な情報を得られると考えられる。また、切除平面共有による探索効率向上は、こうした複数の解を得る際の計算コストを削減する上で大いに寄与することが明らかとなった。

4.5 まとめと今後の課題

本章では、室蘭市のごみ収集問題を題材として、提案手法である MOBB/D が実用的な最適化課題にも適用可能であることを示すことを主眼とし、数理最適化モデルを構築して検証を行った。単目的モデルの段階では、実際のごみ収集ルートと比較して約 25% もの走行距離削減が実現可能であることが分かり、ごみ量の動的変動にも対応できる柔軟性が示された。この結果は、単なる事例研究にとどまらず、提案手法が比較的大規模かつ複雑な VRP 型問題に対しても有効であることを暗示している。

さらに、多目的モデルを導入することで、業者負担（走行距離）と住民負担（歩行距離）のトレードオフを可視化し、ステーションの集約と経路計画を同時に扱う複合的なモデルに対しても、MOBB/D が十分に適用可能であることを確認した。特に、分枝カット法による部分巡回路除去制約などの切除平面を複数の部分問題間で共有する仕組みを組み込み、重みベクトルを変化させながら連続的に問題を解いていくプロセスにおいても、カット情報を再利用することで計算時間を約 10% 削減できた。この成果は、多目的混合整数最適化問題の分割探索を高速化できるという MOBB/D の利点を、実問題ベースで明確に裏付けるものである。

今後の課題としては、まず現時点では最低限の交通事情しか考慮していないため、狭隘路や一方通行などの地理的制約をモデルにより正確に組み込み、最適ルートの現実性をさらに高める必要がある。また、室蘭市全域へ適用した場合にはステーション数や候補地点数が大幅に増加し、問題規模が膨大となる可能性があるが、今回の実験で得られた切除平面共有の効果が拡大条件下でも維持できるかを検証し、提案手法の計算効率をさらに高める方策を検討すべきである。これらの拡張に取り組むことで、MOBB/D の実運用に向けた適用可能性がいっそう明確になり、本研究が提案する多目的最適化フレームワークが多様な社会課題における合意形成の一助となることが期待される。

第 5 章

結論

5.1 本論文の成果

本研究では、多目的混合整数線形計画問題 (MOMILP) に対する新しい厳密解法として、分解ベースのアプローチと分枝限定法を統合した Multi-Objective Branch-and-Bound based on Decomposition (MOBB/D) を提案した。従来の進化計算手法が厳密解を保証できない点や、多目的分枝限定法が計算コストの高さから実問題への適用が難しい点を克服することを目的とした。特に、探索情報の再利用という新しいアプローチを導入し、各部分問題の最適化効率を向上させるとともに、全体として効率的かつ厳密なパレート最適解集合の構築を可能にした。

本研究では、序論で掲げた 3 つの研究目的を達成した。第一に、新たな厳密解法である MOBB/D を提案し、その探索効率と有効性を理論的に検証した。提案手法では、分解ベースのアプローチと分枝限定法を組み合わせることで、MOMILP における厳密解の効率的な探索を可能にした。特に、探索情報の再利用戦略を導入することで、計算コストを削減しつつ、既存手法と比較して優れた性能を示した。

第二に、数値実験を通じて、提案手法の性能を詳細に評価した。従来の多目的分枝限定法と比較した結果、提案手法が計算効率と解の質の両面で優れていることを示した。また、探索情報の再利用の有無を切り替えた実験により、この再利用戦略が提案手法の効率化に大きく寄与していることを確認した。

第三に、実問題への適用により、提案手法の実用性を明らかにした。室蘭市のごみ収集問題では、効率的な経路計画を実現する手段として MOBB/D が有効に機能することを示した。また、EBPM の観点においても、提案手法が適切な意思決定を支援する解法として活用可能であることを実証した。これらの結果は、提案手法の汎用性と実問題への適応力を裏付けるものである。

本研究の意義は、MOMILP に対する厳密解法の新たな可能性を切り開いた点にある。提案手法 MOBB/D は、分解ベースのアプローチと分枝限定法の統合により、厳密解法としての性能を維持しながら、従来の課題であった計算効率の低さを克服した。さらに、探索情報の再利用という戦略により、複数の部分問題を効率的に解くことが可能となり、計算資源の有効活用を実現した。

また、本手法はベンチマーク問題に対する数値実験のみならず、室蘭市のごみ収集問題といった実問題への適用を通じて、その実用性が確認された。これにより、本研究の成果は、学術的な厳密解法の枠を超え、現実世界での意思決定や最適化問題の解決に寄与する具体的な解法としての有用性を示した。これらの成果は、MOMILP への適用可能性を広げるとともに、多目的最適化全般における新たなアプローチの方向性を示唆するものである。

5.2 今後の課題

本研究では提案手法 MOBB/D の有効性を示したが、今後さらなる改善および発展の余地が残されている。以下の 3 つの観点から今後の課題を整理する。

第一に、さらなる近傍情報の適用方法の検討が必要である。本研究では分枝限定法における探索情報の再利用を中心に議論したが、従来の分枝限定法に組み込まれてきた列生成法やその他の効率化手法も適用可能と考えられる。これらの手法を提案手法に組み込むことで、さらなる計算効率の向上が期待されるため、有効性の検証が今後の課題となる。

第二に、近傍情報の取捨選択方法の検討が必要である。現状では、得られた探索情報をすべて保持して活用する形を取っているが、遠く離れた重みベクトル上の部分問題から得られた探索情報は、別の部分問題では有効に働かない可能性がある。そのため、しばらく利用されていない探索情報を適宜削除し、必要な情報のみを効率的に保持するアプローチの構築が求められる。

第三に、問題の適用範囲の拡大が挙げられる。本研究では問題の線形性を仮定して提案手法を構築したが、分枝限定法の拡張により、凸二次計画問題や、内点法を活用する一部の非線形計画問題への適用も考えられる。これにより、提案手法の適用範囲が大幅に広がり、さらなる実問題への対応が可能となる。

以上の課題に取り組むことで、提案手法 MOBB/D は多目的最適化の分野において、より一層の発展と汎用性の向上が期待される。

謝辞

本研究を遂行するにあたり、多くの方々から多大なるご指導とご支援を賜りました。ここに深く感謝の意を表します。

指導教員である渡邊真也教授には、本論文の作成をはじめ、日々の研究指導や学会参加における多方面でのサポートなどを頂きました。また、研究に関する議論だけでなく、個人的な相談にも丁寧に応じてくださり、博士課程を通じて多くの示唆と励ましを受けました。このような環境のもとで成長できたのは、渡邊先生の熱意あふれるご指導のおかげであり、心より感謝申し上げます。

さらに、富山県立大学の榊原一紀教授には、毎月のリモートミーティングにおいて数理最適化の立場から貴重な助言を頂きました。本研究の発展にあたり、異なる視点からの的確なアドバイスは大変有益であり、大きな学びとなりました。心より感謝申し上げます。

また、同研究室に在籍する学生の皆さん、そして既に卒業された学部・修士時代の同期の皆さんには、研究に関する意見交換や時には何気ない雑談を通じて精神的な支えとなっていました。ともに励まし合いながら研究活動を行えたことは、この博士論文を仕上げるうえで大きな力となりました。

最後に、大学生活の10年間を通じて、精神的・経済的に支えてくれた両親に深く感謝いたします。

参考文献

- [1] Dantzig, G. B. (1951) “Maximization of a linear function of variables subject to linear inequalities,” *Activity analysis of production and allocation*, **13**, 339–347.
- [2] Schrijver, A. (1998) *Theory of linear and integer programming*: John Wiley & Sons.
- [3] Bland, R. G. (1977) “New finite pivoting rules for the simplex method,” *Mathematics of operations Research*, **2**, No. 2, 103–107.
- [4] Wolsey, L. A. (2020) *Integer programming*: John Wiley & Sons.
- [5] 梅谷俊治 (2020) 『しっかり学ぶ数理最適化: モデルからアルゴリズムまで』, 講談社.
- [6] IBM (2022) *What are user cuts and lazy constraints?*, URL: <https://www.ibm.com/docs/en/icos/22.1.2?topic=pools-what-are-user-cuts-lazy-constraints>, Accessed: 2024-12-14.
- [7] Gurobi Optimization, LLC (2024) *Linear Constraint Attributes: Lazy*, URL: <https://docs.gurobi.com/projects/optimizer/en/current/reference/attributes/constraintlinear.html#lazy>, Accessed: 2024-12-14.
- [8] Zhang, Q. and Li, H. (2007) “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on evolutionary computation*, **11**, No. 6, 712–731.
- [9] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002) “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE transactions on evolutionary computation*, **6**, No. 2, 182–197.
- [10] Kennedy, J. and Eberhart, R. (1995) “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*, **4**, 1942–1948, ieee.
- [11] Ehrgott, M. and Gandibleux, X. (2001) “Bounds and bound sets for biobjective combinatorial optimization problems,” in *Multiple Criteria Decision Making in the New Millennium: Proceedings of the Fifteenth International Conference on*

- Multiple Criteria Decision Making (MCDM) Ankara, Turkey, July 10–14, 2000*, 241–253, Springer.
- [12] Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008) “Evolutionary many-objective optimization: A short review,” in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, 2419–2426, IEEE.
 - [13] Kaho, T., Watanabe, S., and Sakakibara, K. (2022) “Multi-objective Branch and Bound based on Decomposition,” *IEEJ Transactions on Electronics, Information and Systems*, **142**, No. 3, 373–381.
 - [14] Kirlik, G. and Sayın, S. (2014) “A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems,” *European Journal of Operational Research*, **232**, No. 3, 479–488.
 - [15] Mavrotas, G. and Diakoulaki, D. (1998) “A branch and bound algorithm for mixed zero-one multiple objective linear programming,” *European Journal of Operational Research*, **107**, No. 3, 530–541.
 - [16] Huang, Q., Parker, D. C., Filatova, T., and Sun, S. (2014) “A review of urban residential choice models using agent-based modeling,” *Environment and Planning B: Planning and Design*, **41**, No. 4, 661–689.
 - [17] Gostoli, U. and Silverman, E. (2022) “An agent-based model of social care provision during the early stages of Covid-19,” *Scientific Reports*, **12**, No. 1, 16534.
 - [18] Dantzig, G., Fulkerson, R., and Johnson, S. (1954) “Solution of a large-scale traveling-salesman problem,” *Journal of the operations research society of America*, **2**, No. 4, 393–410.
 - [19] Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960) “Integer programming formulation of traveling salesman problems,” *Journal of the ACM (JACM)*, **7**, No. 4, 326–329.
 - [20] Desrochers, M. and Laporte, G. (1991) “Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints,” *Operations Research Letters*, **10**, No. 1, 27–36.
 - [21] Dantzig, G. B. and Ramser, J. H. (1959) “The truck dispatching problem,” *Management science*, **6**, No. 1, 80–91.
 - [22] OpenStreetMap Foundation (2024) *OpenStreetMap*, URL: <https://www.openstreetmap.org/>, Accessed: 2024-12-14.
 - [23] Lysgaard, J. (2003) “CVRPSEP: A package of separation routines for the Capac-

tated Vehicle Routing Problem,” workingpaper.

- [24] Lysgaard, J., Letchford, A. N., and Eglese, R. W. (2004) “A new branch-and-cut algorithm for the capacitated vehicle routing problem,” *Mathematical programming*, **100**, 423–445.

付録

A 研究業績

1. 学術論文

1. Tomoki Kaho, Shinya Watanabe and Kazutoshi Sakakibara, “Multi-objective Branch and Bound based on Decomposition,” 電気学会論文誌C（電子・情報・システム部門誌）, 142 巻, 3 号, (頁 373 ～ 381), 2022 年 03 月, DOI: 10.1541/ieejeciss.142.373
2. Tomoki Kaho, Kazutoshi Sakakibara, Mikiharu Arimura and Shinya Watanabe, “Optimization of Garbage Collection Routes for Evidence-Based Policy Making,” *Artificial Life and Robotics*, (in press).

2. 一般公演

3. 下保知輝, 渡邊真也, 榊原一紀, “単目的問題への分割に基づく多目的分枝限定法の提案,” 情報処理学会 第 83 回全国大会, 情報処理学会, 情報処理学会 第 83 回全国大会 講演論文集, (p.1-163 - 1-164), 2021.03.18.
4. 佐藤 輝, 渡邊 真也, 下保 知輝, “多目的スクラップ配合問題に対する解法とその解析,” 第 135 回数理モデル化と問題解決研究発表会, 情報処理学会 数理モデル化と問題解決研究, 第 135 回数理モデル化と問題解決研究発表会予稿集, 2021.10.18
5. Tomoki Kaho and Shinya Watanabe, “Optimization of Garbage Collection Routes as Reference Data in Evidence-Based Policy Making,” Proceedings of AROB-ISBC-SWARM2024, (頁 1026 ～ 1030), 2024 年 01 月, Beppu, JAPAN
6. Tomoki Kaho, Kazutoshi Sakakibara, Mikiharu Arimura and Shinya Watanabe, “Multiobjective Branch-and-Cut Method with Shared Cutting Plane Information: A Case Study on Optimizing Garbage Collection,” 2024 Joint 13th International Conference on Soft Computing and Intelligent Systems and 25th International Symposium on Advanced Intelligent Systems (SCIS&ISIS), 2024.11.11. Himeji, Japan, pp. 1-6, DOI: 10.1109/SCISI-

SIS61014.2024.10760213.

3. ポスター

7. 下保 知輝, 黒丸 拓夢, 坂本 裕哉, 中田 涼介, 稲船 淳也, 渡邊真也, “進化型多目的最適化における同時摂動最適化法を利用した局所探索アプローチの提案,” 進化計算学会 進化計算シンポジウム 2017, 進化計算学会, 進化計算シンポジウム 2017 予稿集, (p.390-396), 2017.12.09.
8. 下保知輝, 渡邊真也, 榊原一紀, “分枝限定法における支配関係の概念導入による効果,” 計測自動制御学会 システム・情報部門 学術講演会 2018(SSI2018), 計測自動制御学会, 計測自動制御学会 システム・情報部門 学術講演会 2018 予稿集, (p.SS14-01), 2018.11.25.