

平成 27 年度 修士論文

適応化のフレームワークの構築と
適応型メタヒューリスティクスの開発
に関する研究

首都大学東京大学院 理工学研究科 電気電子工学専攻

熊谷 渉

論文要旨

最適化は工学に限定されることなく、経済学、社会学などの幅広い分野に及んで応用され、その重要性が広く認識されている。近年のシステムの大規模化・複雑化により、システムの設計・運用・制御に対する要求は高度化している。また、シミュレーション技術・計測技術などの最適化分野の周辺技術は急速に発展している。さらに、コンピュータパワーの飛躍的な増大が、数値計算に多大な貢献をしている。このように、最適化分野を取り巻く環境の変化に応じた実用的かつ新たな最適化手法への要求が高まっている。

上述の背景に対応可能な最適化手法の枠組みとして、メタヒューリスティクスが注目されている。従来から使用されてきた数理計画法は、対象関数の解析的情報（勾配や Hesse 行列など）を要することに対して、メタヒューリスティクスは、決定変数値情報と評価値情報のみを用いる。また、実用的な時間に応じて最適性の高い近似解を求めることができる。さらに、多くの手法は生物現象などの経験的に優れたメカニズムからのアナロジーにより構築されている。例えば、Cuckoo Search (CS) はカッコウの托卵行動のメカニズムに基づいており、解が実数空間で定義される連続値最適化問題を対象とした手法である。

メタヒューリスティクスは調整可能なパラメータを有しているが、十分な探索性能の発揮には、パラメータ設定に関する使用者の高い専門的な知識・経験や試行錯誤を必要とする。一方、多様な環境下での使用が想定されるメタヒューリスティクスは、多様な環境下においても高い探索性能を発揮可能な適応能力が不可欠である。従って、メタヒューリスティクスの適応能力の向上、及び高い適応能力を有する新たなメタヒューリスティクスの開発には、適応能力向上を実現し得る「汎用的フレームワークの構築」が最も重要な課題となる。

まず本論文では、アナロジーではなく、探索構造の観点から、様々なメタヒューリスティ

クスを解析した。そして、構造解析を通じて得た「最適化手法として優れた戦略」に基づき、様々なメタヒューリスティクスに適用可能な、適応化のフレームワークを構築した。さらに、適応化のフレームワークに基づき、連続値最適化問題を対象とする適応型メタヒューリスティクスの開発を行った。

メタヒューリスティクスでは、「探索序盤では多様化の実現、探索終盤では集中化の実現」という探索戦略が知られている。本論文では、様々なメタヒューリスティクスの構造解析を通じて、多様化・集中化の実現状態（探索状態）を調整する操作を定義・分類した。このように、共通概念である多様化・集中化に影響を与える操作を定義・分類しておくことで、各手法の多様化・集中化に対する解析が可能となる。さらに、メタヒューリスティクスは探索過程で多様化・集中化を適切に実現することで、問題構造や探索条件に適応すると同時に、優れた探索性能を発揮する。従って、定義した多様化・集中化に影響を与える操作を巧みに調整することで、高い適応能力・探索性能が期待できる。

メタヒューリスティクスの更なる有用性の向上のために、これまでに適応化のためのアプローチはいくつか提案されている。しかし、これらのアプローチの多くは、① 統一的な視点に基づく適応能力を具備したメタヒューリスティクスの設計論が存在せず、既存の手法に対するパラメータ調整機能の付加のみに留まっていること、② 対象の手法に特化した観点に基づいており、明確な多様化・集中化の観点に基づいていないこと、などの課題が存在する。そこで本論文では、多様化・集中化に基づく適応化のための汎用的フレームワークを構築した。適応化のフレームワークは、既存のアプローチとは異なり、① 既存の手法に対して適用可能であること、② 新たな手法を構築し、その手法に対して適用可能であること、から、汎用性が高い。既存の手法に対しては、① 探索状態を定量的に評価すること、② 探索状態の指標をフィードバックし、パラメータを調整することで、探索状態を制御すること、を実現する。また、優れた適応性を有する新たな手法を構築し、上述の方法でその手法を適応化する。このように、探索状態の評価と制御に基づく探索能力を開発・付加することで、適応型メタヒューリスティクスを構築する枠組みを「適応化のフレームワーク」とする。これは、パラメータ調整機能の付加に限らず、高い適応能力・探索性能を具備した新たな適応型メタヒューリスティクスの設計に繋がる。

上述した通り、適応化のフレームワークに基づくことで、既存の手法の系統的な適応化が可能となり、適応能力・探索性能の向上が期待できる。そこで本論文では、適応化のフレームワークに基づき、適応型 CS の開発を行った。

さらに、適応化のフレームワークに基づくと、新たな適応型メタヒューリスティクスの開発が可能となる。そこで本論文では、適応性に優れた探索構造を具備する新たなメタヒューリスティクスを構築した。さらに、この手法を基礎とすることで、より優れた適応型メタヒューリスティクスの開発を実現した。

最後に、典型的なベンチマーク関数を対象とした数値実験を通じて、既存手法と比較することで、提案手法の探索性能・適応能力を検証した。

本論文では、メタヒューリスティクスの更なる有用性の向上のために、適応化のフレームワークを構築した上で、このフレームワークに基づき、適応型 CS や新たな適応型メタヒューリスティクスを開発した。最後に、数値実験を通じて提案手法の探索性能を検証し、適応化のフレームワークの有用性を確認した。

目次

論文要旨	i
1 はじめに	1
1.1 本研究の背景	1
1.2 本論文の目的と位置付け	4
1.3 本論文の構成	6
1.4 表記の諸定義	7
2 最適化手法の基礎と数理	13
2.1 最適化問題の数理	13
2.1.1 最適化問題の定式化	13
2.1.2 局所的最適解と大域的最適解	14
2.1.3 最適化問題の分類	15
2.2 最適化手法の数理	17
2.2.1 最適化手法の基本構造	17
2.2.2 特殊な解・探索点	18
2.3 数理計画法	19
2.3.1 数理計画法と最適化問題の関連性	19
2.3.2 降下法	20
2.3.3 勾配法	21
2.3.4 最急降下法	23
2.3.5 Newton-Raphson 法	23

2.3.6	準 Newton 法	24
2.4	メタヒューリスティクス	26
3	メタヒューリスティクスの探索構造の解析	29
3.1	メタヒューリスティクスの基本的な探索構造	29
3.2	Particle Swarm Optimization	31
3.2.1	Particle Swarm Optimization の概要	31
3.2.2	Particle Swarm Optimization のアルゴリズム	32
3.2.3	Particle Swarm Optimization の探索構造の解析	34
3.3	Firefly Algorithm	35
3.3.1	Firefly Algorithm の概要	35
3.3.2	Firefly Algorithm のアルゴリズム	36
3.3.3	Firefly Algorithm の探索構造の解析	38
3.4	Differential Evolution	39
3.4.1	Differential Evolution の概要	39
3.4.2	Differential Evolution のアルゴリズム	40
3.4.3	Differential Evolution の探索構造の解析	42
3.5	Artificial Bee Colony Algorithm	43
3.5.1	Artificial Bee Colony Algorithm の概要	43
3.5.2	Artificial Bee Colony Algorithm のアルゴリズム	43
3.5.3	Artificial Bee Colony Algorithm の探索構造の解析	47
3.6	Cuckoo Search	48
3.6.1	Cuckoo Search の概要	48
3.6.2	Lévy Flight	49
3.6.3	Cuckoo Search のアルゴリズム	50
3.6.4	Cuckoo Search の探索構造の解析	52
3.7	メタヒューリスティクスの探索戦略	53
3.8	メタヒューリスティクスの探索構造の分類	55
3.8.1	近傍の生成方法	55
3.8.2	ベクトルの種類	56

3.8.3	乱数の使用箇所	58
3.9	メタヒューリスティクスの探索戦略と探索構造	60
4	適応化のフレームワーク	62
4.1	メタヒューリスティクスの適応能力と探索構造	62
4.2	適応化のための汎用的フレームワークの構築	64
4.3	適応化設計論	66
5	適応化のフレームワークに基づく 適応型 Cuckoo Search	71
5.1	Cuckoo Search のパラメータ調整方法	71
5.2	Cuckoo Search の解析	72
5.2.1	Cuckoo Search の特徴	73
5.2.2	Cuckoo Search の探索ダイナミクスの解析	73
5.2.3	Cuckoo Search のパラメータ解析	74
5.3	Cuckoo Search の多様化・集中化の評価指標	75
5.3.1	評価指標の提案	75
5.3.2	評価指標の数値実験的検証	77
5.4	探索状態の評価と制御に基づく適応型 Cuckoo Search	81
5.4.1	多様化・集中化の適応的制御	81
5.4.2	適応型 Cuckoo Search	82
5.5	数値実験による検証	85
5.5.1	数値実験条件	85
5.5.2	実験結果・考察	86
5.6	本章のまとめ	87
6	適応化のフレームワークに基づく メタヒューリスティクス	91
6.1	適応化設計論に基づくメタヒューリスティクスの設計	91
6.2	適応化のフレームワークに基づくメタヒューリスティクスの多様化・集 中化の検証	94

6.3	数値実験による検証	99
6.3.1	数値実験条件	99
6.3.2	実験結果・考察	99
7	適応化のフレームワークに基づく適応型 メタヒューリスティクス	112
7.1	探索状態の評価と制御に基づく適応型メタヒューリスティクス	112
7.2	数値実験による検証	115
7.2.1	数値実験条件	115
7.2.2	実験結果・考察	116
8	おわりに	121
8.1	本論文のまとめ	121
8.2	今後の展望	123
	謝辞	124
	参考文献	125
A	Lévy Flight に関する定量的解析	132
B	ベンチマーク関数	135

1

はじめに

1.1 本研究の背景

最適化とは、「ある制約条件の下で、多数の選択肢の中から、目的に対して最適な成果を出す選択肢（解）を選ぶ行為」を指す。最適化は工学、経済学、社会学などの幅広い分野に及んで応用され、その重要性が広く認識されている。特に工学では、計画問題や製品設計問題など、「問題毎に設定された目的に対する成果（目的関数）を最大化あるいは最小化する問題」を、最適化問題として数理的に記述することが可能である。具体的な例として、工業製品の寸法・形状・トポロジー最適設計問題 1 [2], 生産分野の工場の生産計画最適化問題 3 [4 [5], エネルギー分野の設備最適運用問題 6 [7 [8], 機械学習分野のニューラルネットワークの学習問題 9], 金融工学分野のポートフォリオ最適化問題 10 [11] などが挙げられる。これらの最適化問題は、システムの出力（目的関数）が目的に対して最適な値となるように、入力（解）を決定する問題として置き換えられる。システムとは、「相互に作用し合う、複数の要素から構成される集合体」を指す。歴史的には、Dantzig により単体法 12 が開発されてから、最適化手法の開発・改良や、適用範囲の拡張が進められてきた。現代では、一般のシステムを対象とする最適化（システム最適化）は、現代工学の基盤技術の一つとして確立されている。

工学における最適化の形態は、下記の要素から構成されている。

- (1) 対象となる実システム
- (2) 実システムを表現した数式モデル

(3) 数式モデルに適用可能な最適化手法

本来，上記の最適化の形態において，(2)では，有効な最適化手法が適用できるように，数式モデルの定式化を行い，(3)では，対象の問題に有効な最適化手法を選択することが常道である。したがって，(2)と(3)は密接な関係にある。

最適化の理論は，凸解析[13]と呼ばれる数学の分野から発展した後，解の最適性や最適化問題の双対性など，最適化に関する重要な諸概念が整備され，現在でも，最適化問題に対する解析や最適化手法の設計において，重要な役割を果たしている。このように，最適化問題に対して，上記の理論に基づき最適な解を導くための方法論を数理計画といい，この方法論の下で構築・設計された最適化手法を数理計画法[14][15][16][17][18][19][20][21]という。数理計画法は，最適化の理論に従い，数式モデルの解析的情報（勾配やHesse行列など）を用いて，数学的に保証された解を求める。代表的な数理計画法として，線形計画問題を対象とする単体法や，非線形計画問題を対象とするNewton-Raphson法が歴史적으로よく知られている。これまで数理計画法に対して，解の収束性・最適性など，多くの数理的な知見や性質が明らかにされていることから，上記の最適化の形態の(3)において，従来から数理計画法が使用されてきた。

しかし，数理計画法は，最適化問題の解析的な情報を必要とするため，手法の種類によって，適用可能な最適化問題のクラスが規定される。これは，上記の最適化の形態の(3)において数理計画法を採用した場合，(2)と(3)の依存関係が互いに強いため，(2)と(3)の適用可能なクラスが限定されることを示している。このため，数理計画法を用いた最適化では，以下の問題が生じる。

- 手法の規定により，実システムが数式モデルへ表現可能であることや，数式モデルの微分可能性・連続性を有していること，などの様々な条件が必要となるため，最適化の適用範囲が限定される。
- 数式モデルを手法に合致するように表現した結果，実システムと数式モデルの間に乖離が生じるため，解の有用性や実行可能性が不十分となる。

このように，数理計画に基づく最適化では，手法が最適化の形態に制約条件を課すことから，その実用化には課題が多い。

一方，近年のシステムや工業製品の大規模化・複雑化により，システムの設計・運用・

制御では高効率化・高信頼化・高機能化の実現が困難となっている。また，現実の最適化では，長時間をかけて厳密な最適解のみを求めることよりも，実用的な時間内に十分な最適性を有する解（準最適解）を求めることに対するニーズが多い。さらに近年では，コンピュータパワーの飛躍的な増大が，最適化アルゴリズムやシミュレーションなどの数値計算に多大な貢献をしている。このように，最適化分野を取り巻く環境の変化（① 実システムの大規模化・複雑化，② 計算時間の制約，③ 周辺技術の発展）に応じた，実用的かつ新たな最適化手法の必要性が高まっている。

上述の「実用上における従来の最適化手法の課題」や「最適化分野の環境の変化」に対応可能な最適化手法の枠組みとして，近年ではメタヒューリスティクス（発見的近似解法）が注目されている[22][23][24][25][26][27]。メタヒューリスティクスは，① 解直接探索法，② 実用的な近似手法，③ 発見的手法，という特徴を有している。解直接探索法とは，最適化問題の解情報と目的関数値情報のみを用いて最適化を行う手法である。最適化の形態の(3)において解直接探索法であるメタヒューリスティクスを採用した場合，数理計画法とは異なり，手法が数式モデルに対して制約条件を課さない。そのため，高精度なモデリングや，シミュレータ・計測機器からの直接的な目的関数値の獲得など，最適化の形態が柔軟となる。また，実用的な計算時間に応じて最適性の高い近似解を求めることができる。さらに，多くの手法は生物現象・物理現象などの経験的に優れたメカニズムに基づき，準最適解を探索する。例えば，Particle Swarm Optimization[28]は鳥や魚の群としての採餌行動，Differential Evolution[29]は生物の進化過程のメカニズムに基づく手法である。また，メタヒューリスティクスは調整可能なパラメータを有しており，パラメータの自由度を活用し，問題構造や探索条件に応じて適切に設定することで効率的に探索を行うことができる。このように，優れた汎用性・探索性能を有するメタヒューリスティクスは，工学的な価値が高い最適化手法として注目されている[22][23][24]。

メタヒューリスティクスという枠組みの登場以来，2005年のArtificial Bee Colony Algorithm[30]，2008年のFirefly Algorithm[27][31][32]，2009年のCuckoo Search[27][33]など，新たな手法の開発はますます活発となっている。手法の種類が増加する一方で，メタヒューリスティクスは，① 数学的な背景を有していないため，得られた解に対する最適性は保証されていないこと，② 十分な探索性能の発揮には，パラメータ設定に関する使用者の専門的な知識・経験や試行錯誤を必要とすること，が課題点としてしばしば指摘される。そこで近年では，メタヒューリスティクスの実システムへの応用や更なる有用性の

向上のために、探索ダイナミクスに対する数理的・数値実験的解析や、解析に基づく手法の改良・開発に関する研究が盛んになっている[34][35][36][37][38][39][40][41][42]。今後も、数値実験や解析を通じて、① パラメータの設定・調整に対する有効な知見の抽出による系統的な分類・整理、② 数理的な根拠に基づくパラメータの調整方法の開発、③ メタヒューリスティクスの設計論の確立、などの更なる発展が期待されている。

さらに、上述の通り、多様な環境下での使用が想定されるメタヒューリスティクスは、多様な環境下においても高い探索性能を発揮可能な適応能力が不可欠である。メタヒューリスティクスの更なる有用性の向上のために、これまでに適応化のためのアプローチはいくつか提案されている。しかし、これらのアプローチの多くは、① 統一的な視点に基づく適応能力を具備したメタヒューリスティクスの設計論が存在せず、既存の手法に対するパラメータ調整機能の付加のみに留まっていること、② 対象の手法に特化した観点に基づいており、メタヒューリスティクスに共通する普遍的な観点に基づいていないこと、などの課題が存在する。したがって、メタヒューリスティクスの適応能力の向上、および高い適応能力を有する新たなメタヒューリスティクスの開発において、適応能力の向上を実現し得る「汎用的フレームワーク」が最も重要な役割を果たす。

本論文では、上述のメタヒューリスティクスの適応能力に関する課題解決に資するため、「適応化のフレームワーク」の構築や、適応型メタヒューリスティクスの開発を行う。

1.2 本論文の目的と位置付け

本論文では、メタヒューリスティクスの適応能力の向上・開発のために、様々な手法に適用可能な適応化のフレームワークの構築と、適応化のフレームワークに基づく適応型メタヒューリスティクスの開発を目的とする。

はじめに、自然現象や物理現象などの経験的に優れたメカニズムの観点ではなく、最適化手法としての探索構造の観点から、様々なメタヒューリスティクスを解析する。メタヒューリスティクスでは、「探索序盤では多様化の実現、探索終盤では集中化の実現」という探索戦略が知られている。本論文では、様々なメタヒューリスティクスの構造解析を通じて、多様化・集中化の実現状態（探索状態）を調整する操作を定義・分類する。このように、共

通概念である多様化・集中化に影響を与える操作を定義・分類しておくことで、各手法の多様化・集中化に対する解析が可能となる。さらに、メタヒューリスティクスの多様化・集中化の実現状態と適応能力の関連性に対する考察を通じて、優れた探索構造を抽出する。

次に、多様化・集中化に基づく適応化のための汎用的フレームワークを構築する。適応化のフレームワークは、既存のアプローチとは異なり、① 既存の手法に対して適用可能であること、② 新たな手法を構築し、その手法に対して適用可能であること、から、汎用性が高い。既存の手法に対しては、多様化・集中化の実現状態の評価と制御に基づく探索能力を付加することで、適応化する。また、優れた適応性を有する新たな手法を構築し、上述の方法でその手法を適応化する。このように、探索状態の評価と制御に基づく探索能力を開発・付加することで、適応型メタヒューリスティクスを構築する枠組みを「適応化のフレームワーク」とする。

さらに、これまでに提案されてきた適応化のためのアプローチの多くは、① 既存の手法に対するパラメータ調整機能の付加のみに留まっていること、② 対象の手法に特化した観点に基づいていること、などの課題が存在していた。これに対して、本論文で構築する適応化のフレームワークは、① パラメータ調整機能の付加に限らず、高い適応能力・探索性能を具備した新たな適応型メタヒューリスティクスの柔軟な設計が可能となること、② 多くのメタヒューリスティクスの普遍的な探索構造・戦略に立脚するというアプローチに基づいていること、などの点で、これまでのアプローチとは大きく異なる上に、これまでのアプローチにおける課題解決に資する可能性が高い。本論文では、適応化のフレームワークに基づき、具体的に、① 既存の手法に対する適応化、② 新たに構築した手法に対する適応化、を行い、フレームワークの有用性を検証する。

上述の通り、適応化のフレームワークに基づくことで、既存の手法の系統的な適応化が可能となり、適応能力・探索性能の向上が期待できる。そこで本論文では、適応化のフレームワークに基づき、適応型 Cuckoo Search (CS) の開発を行う。解析を通じて、他のメタヒューリスティクスに比べて、CS の多様化・集中化の調整能力が低いことを指摘する。その後、適応化のフレームワークに従い、適応型 CS を構築する。最後に、数値実験を通じて、適応型 CS の探索性能・適応能力を検証する。

さらに、適応化のフレームワークに基づく、新たな適応型メタヒューリスティクスの設計・開発が可能となる。そこで本論文では、適応性に優れた探索構造を具備する新たなメタヒューリスティクスを構築する。その後、適応化のフレームワークに従い、この手法

を基礎とした適応型メタヒューリスティクスを構築する。最後に、数値実験を通じて、新たな適応型メタヒューリスティクスの探索性能・適応能力を検証する。

以上の成果により、適応化のフレームワークに従う場合、十分優れた適応能力・探索性能を有する適応型メタヒューリスティクスを開発することが可能となる。この研究成果は、1.2のメタヒューリスティクスの適応能力に関する課題に対して大きく資すると考えられる。

1.3 本論文の構成

本節では、本論文の構成について述べる。本章の以降の部分では、本論文で使用する数学的用語とその定義、および表記法をまとめる。

2章では、本論文で扱う最適化問題、および代表的な数理計画法やメタヒューリスティクスのアルゴリズムを記述する。

3章では、メタヒューリスティクスの探索構造を解析し、メタヒューリスティクスに共通する探索戦略を抽出する。ここでは、4章以降でメタヒューリスティクスに対して統一的な視点で議論するための土台を形成する。

4章では、構造解析を通じて得た探索戦略の観点から、メタヒューリスティクスの適応能力と探索性能の関係について議論する。その議論を基に、メタヒューリスティクスの適応能力の向上のための「汎用的フレームワーク」を構築する。

5章では、適応化のフレームワークを既存手法である Cuckoo Search (CS) に適用することで、適応型 CS を開発する。ここでは、CS の探索ダイナミクスやパラメータについて解析する。また、CS の探索状態の評価指標を提案し、数値実験を通じてその有効性を検証する。最後に、典型的なベンチマーク関数を対象とした数値実験を通じて、オリジナル CS との比較により、適応型 CS の有用性を検証する。

6章では、適応化のフレームワークに基づき、適応性に優れた探索構造を有する新たなメタヒューリスティクスを構築する。最後に、典型的なベンチマーク関数を対象とした数値実験を通じて、他の手法との比較により、提案したメタヒューリスティクスの有用性を検証する。

7章では、適応化のフレームワークに従い、6章で構築した手法を基礎とした適応型メタ

ヒューリスティクスを開発する。最後に、典型的なベンチマーク関数を対象とした数値実験を通じて、既存の手法との比較により、提案した適応型メタヒューリスティクスの有用性を検証する。

最後に8章では、本論文の内容をまとめると同時に、今後の課題や研究展望について述べる。

1.4 表記の諸定義

本節では、本論文で扱う数式表記の諸定義を行う。

(a) 集合

集合とは、いくつかのもののから構成される集まりのことである。集合を構成する個々のものを元、あるいは要素という。

ある元 a が集合 A を構成するものの一つであるとき、「 a は集合 A に属する」といい、 $a \in A$ と表す。また、ある2つの集合 A, B について、 A に属する全ての元が B にも属する、すなわち $a \in A \Rightarrow a \in B (\forall a)$ が成立するとき、「 A は B の部分集合である」、あるいは「 B は A を包含している」といい、 $A \subset B$ あるいは $A \subseteq B$ と表す。

一般的な集合の記述方法として、外延的記法と内包的記法が存在する。外延的記法とは、集合の全ての元を列挙することで、集合を記述する方法である。例えば、 $1, 3, 5, 7, 10$ から構成される集合を $\{1, 3, 5, 7, 10\}$ と表す。内包的記法とは、元が対象の集合に属するための必要十分条件を与えることで、集合を記述する方法である。例えば、条件 $P(x)$ を満たす元だけを全て集めた集合を $\{x | P(x)\}$ と表す。

有限個の元から構成される集合を有限集合、有限集合でない集合を無限集合という。「集合 A に属する元の個数」を拡張させたものを濃度、あるいは基数といい、 $|A|$ と表す。有限集合の濃度は、集合に属する元の個数と一致する。

本論文で扱う特別な集合の表記を下記の通り定義する。

- \mathbb{R} : 実数全体からなる集合

- \mathbb{Z} : 整数全体からなる集合
- \mathbb{N} : 自然数全体からなる集合
- \emptyset : 空集合

(b) ベクトル

本論文では、特に断りのない限り、ベクトルとは列ベクトルを指す。 $N \in \mathbb{N}$ 次元のベクトルを式 (1.1) で表す。

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \quad (1.1)$$

ここで、 $x_n (n = 1, \dots, N)$ をベクトル $\boldsymbol{x} \in \mathbb{R}^N$ の第 n 要素という。転置記号 T を用いて、 $\boldsymbol{x} = (x_1, x_2, \dots, x_N)^T$ と表す場合もある。特に、全ての要素が0であるベクトルを零ベクトルといい、 $\mathbf{0}$ と表す。

(c) 行列

$N, M \in \mathbb{N}$ に対して、 N 行 M 列の行列 $\boldsymbol{A} \in \mathbb{R}^{N \times M}$ を式 (1.2) で表す。

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NM} \end{pmatrix} \quad (1.2)$$

ただし、 $a_{nm} (n = 1, \dots, N; m = 1, \dots, M)$ を行列 \boldsymbol{A} の n 行 m 列成分という。全ての成分が0である行列を零行列といい、 \mathbf{O} と表す。

特別な性質を有する行列を下記の通り定義する。

- 転置行列：行列 $\boldsymbol{A} \in \mathbb{R}^{N \times M}$ の行・列成分を逆にした M 行 N 列の行列を転置行列といい、 \boldsymbol{A}^T と表す。
- 対称行列： $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ を正方行列といい、 $\boldsymbol{A} = \boldsymbol{A}^T$ を満たす正方行列 \boldsymbol{A} を対称行列という。対称行列 $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ について、二次形式 $\boldsymbol{z}^T \boldsymbol{A} \boldsymbol{z} > 0$ ($\forall \boldsymbol{z} \in \mathbb{R}^N, \boldsymbol{z} \neq \mathbf{0}$) が成り立つとき、 \boldsymbol{A} を正定値行列といい、 $\boldsymbol{A} > 0$ と表す。

- 対角行列：対角成分以外の成分が全て0である正方行列 $D \in \mathbb{R}^{N \times N}$ を対角行列という。特に，全ての対角要素が1である対角行列を単位行列といい， $I \in \mathbb{R}^{N \times N}$ と表す。
- 正則行列：正方行列 $A, B \in \mathbb{R}^{N \times N}$ ，単位行列 $I \in \mathbb{R}^{N \times N}$ について，式(1.3)の関係を満たす行列 B が存在するとき， A を正則行列という。

$$AB = BA = I \quad (1.3)$$

正則行列 A について，式(1.3)の関係を満たす行列 B が存在するとき， B を A の逆行列といい， $B = A^{-1}$ と表す。一般的に，正方行列 A について，「逆行列 A^{-1} が存在する（ A の行列式が0でない）こと」と，「 A が正則行列であること」は同値である。行列 $A \in \mathbb{R}^{N \times N}$ が正定値行列ならば， A は正則である。

(d) 関数

ある f が始域（定義域） A から終域 B への写像であるとき， $f: A \rightarrow B$ と表す。数の集合を終域に持つ写像を関数という。特に，実数空間 \mathbb{R} を終域に持つ関数を実数値関数といい，本論文では，特に断りのない限り，関数とは実数値関数を指す。

一つの変数 $x \in \mathbb{R}$ により，値が決定される関数 $f: \mathbb{R} \rightarrow \mathbb{R}$ を一変数関数といい， $f(x)$ と表す。複数の変数を要素とするベクトル変数 $x \in \mathbb{R}^N$ により，値が決定される関数 $f: \mathbb{R}^N \rightarrow \mathbb{R}$ を多変数関数といい， $f(x)$ と表す。

$f: \mathbb{R}^N \rightarrow \mathbb{R}$ のように，写像の終域が1次元空間の場合， f をスカラー値関数という。
 $f: \mathbb{R}^N \rightarrow \mathbb{R}^M$ のように，写像の終域が多次元空間の場合， f をベクトル値関数という。

(e) 距離空間

集合 X とその直積上の関数 $d: X \times X \rightarrow \mathbb{R}$ について，距離の公理[†] を満たす関数 d を距離関数，組 (X, d) を距離空間という。

[†] $\forall x, y, z \in X$ について，

- (1) $d(x, y) \geq 0$ (正定値性)
 - (2) $d(x, y) = 0 \Leftrightarrow x = y$ (非退化性)
 - (3) $d(x, y) = d(y, x)$ (対称性)
 - (4) $d(x, y) + d(y, z) \geq d(x, z)$ (三角不等式)
- が成立するとき， d を X 上の距離という。

この距離空間を用いることで、いくつかの距離概念を定義することができる。本論文では、代表的な距離空間として、実数に限定した L^p 空間を取り扱う。 L^p 空間 (\mathbb{R}^N, d) における距離関数 $d(x, y; p)$ ($x, y \in \mathbb{R}^N$; $p \in \mathbb{R}$) を L^p -ノルム (p -ノルム) といい、式 (1.4) で定義される。

$$d(x, y; p) = \left(\sum_{n=1}^N |x_n - y_n|^p \right)^{1/p} \quad (1.4)$$

式 (1.4) の L^p -ノルムにおいて $p = 1$ のとき、 $d(x, y; 1)$ は Manhattan 距離といい、 $d_M(x, y)$ と表す。 $p = 2$ のとき、 $d(x, y; 2)$ は Euclidean 距離といい、 $d_E(x, y)$ と表す。 $p \rightarrow \infty$ のとき、 $d(x, y; \infty)$ は Chebyshev 距離 (L^∞ 距離) という。本論文では、特に断りが無い限り、距離とは Euclidean 距離を指し、ベクトル $x \in \mathbb{R}^N$ に対する Euclidean ノルムを $\|x\|$ と表す。

距離空間 (X, d) のある点 $x \in X$ と正数 $\varepsilon > 0$ について、開集合 $B(x; \varepsilon)$ を ε -近傍 (x を中心とする半径 ε の開球) といい、式 (1.5) で表す。

$$B(x; \varepsilon) = \{y \in X \mid d(x, y) < \varepsilon\} \subseteq X \quad (1.5)$$

式 (1.5) より、距離空間 (\mathbb{R}^N, d_E) において、ある点 $x^* \in \mathbb{R}^N$ に対する ε -近傍 $B(x^*; \varepsilon)$ は式 (1.6) で表すことができる。

$$B(x^*; \varepsilon) = \{x \in \mathbb{R}^N \mid \|x^* - x\| < \varepsilon\} \quad (1.6)$$

(f) 微分

一変数関数 $f(x)$ ($x \in \mathbb{R}$) に対する微分作用素 (微分演算子) を d/dx と表し、多変数関数 $f(x)$ ($x \in \mathbb{R}^N$) に対する偏微分作用素を $\partial/\partial x_n$ ($n = 1, \dots, N$) と表す。各変数に対する偏微分作用素を要素とするベクトル ∇ をナブラ、あるいは Hamilton の演算子といい、式 (1.7) で表す。

$$\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_N} \right)^T \quad (1.7)$$

ナブラ ∇ を多変数関数 $f(x)$ に作用させた $\nabla f(x) \in \mathbb{R}^N$ を勾配ベクトルといい、各変数に対する一階偏微分 $\partial f(x)/\partial x_n = f_{x_n}$ ($n = 1, \dots, N$) を要素とする。勾配ベクトル $\nabla f(x)$ を式 (1.8) で表す。

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_N} \right)^T \quad (1.8)$$

各変数に対する二階偏微分 $h_{nm} = \partial^2 f(x)/\partial x_n \partial x_m = f_{x_n x_m}(n, m = 1, \dots, N)$ を要素とする行列 $H(f(x)) \in \mathbb{R}^{N \times N}$ を Hesse 行列といい, 式 (1.9) で表す。

$$H(f(x)) = \begin{pmatrix} f_{x_1 x_1} & \cdots & f_{x_N x_1} \\ \vdots & \ddots & \vdots \\ f_{x_1 x_N} & \cdots & f_{x_N x_N} \end{pmatrix} \quad (1.9)$$

f が 2 回連続的微分可能 (f が C^2 級) ならば, $\partial^2 f(x)/\partial x_n \partial x_m = \partial^2 f(x)/\partial x_m \partial x_n$ が成り立つ (Schwarz の定理)。よって, $f(x)$ が C^2 級ならば, Hesse 行列は対称行列となる。

任意の x における多変数関数 $f(x)$ のベクトル $v \in \mathbb{R}^N$ に対する偏微分 $\partial f(x)/\partial v$ を方向微分といい, 式 (1.10) で定義される。

$$\frac{\partial f(x)}{\partial v} = \lim_{h \rightarrow 0} \frac{f(x + hv) - f(x)}{h} \quad (1.10)$$

式 (1.10) の方向微分について, f が x において全微分可能ならば式 (1.11) の定理が成立する。

$$\frac{\partial f(x)}{\partial v} = \nabla f(x)^T v \quad (1.11)$$

(g) 確率分布

変数 $x \in \mathbb{R}$ あるいは, $x \in \mathbb{R}^N$ の値が確率的に定まるとき, これらを確率変数という。確率変数 x がある確率分布 \mathcal{P} に従って分布することを $x \sim \mathcal{P}$ と表し, 確率分布 \mathcal{P} の確率密度関数を $f_{\mathcal{P}}$ と表す。本論文で扱う確率分布の確率密度関数を, 式 (1.12), 式 (1.13), 式 (1.14), 式 (1.15) で定義する。

- 実数値一様分布 $\mathcal{U}_{\mathbb{R}}(a, b)$:

$$f_{\mathcal{U}_{\mathbb{R}}}(x; a, b) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}, x \in \mathbb{R} \quad (1.12)$$

式 (1.12) において, $a, b \in \mathbb{R}$ は分布の下限值, 上限値である。特に, $a = 0, b = 1$ の場合, 実数値標準一様分布という。

- 整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(a, b)$:

$$f_{\mathcal{U}_{\mathbb{Z}}}(x; a, b) = \begin{cases} \frac{1}{b-a+1}, & x \in \{a, \dots, b\} \\ 0, & \text{otherwise} \end{cases}, x \in \mathbb{Z} \quad (1.13)$$

式 (1.13) において, $a, b \in \mathbb{Z}$ は分布の下限值, 上限値である。

- 正規分布 $\mathcal{N}(\mu, \sigma^2)$ (Gauss 分布) :

$$f_{\mathcal{N}}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (1.14)$$

ただし, μ は平均, σ^2 は分散である。特に, $\mu = 0, \sigma^2 = 1$ の場合, 標準正規分布という。

- (狭義) Lévy 分布 $\mathcal{L}(\mu, c)$:

$$f_{\mathcal{L}}(x; \mu, c) = \begin{cases} \sqrt{\frac{c}{2\pi}} \frac{\exp(-c/(2(x-\mu)))}{(x-\mu)^{3/2}}, & \mu \leq x \\ 0, & \text{otherwise} \end{cases} \quad (1.15)$$

ただし, μ は位置パラメータ, c は尺度パラメータである。

2

最適化手法の基礎と数理

本論文の主題となるメタヒューリスティクスの解析のための準備として、一般の最適化手法の基本構造を数学的に整理し、統一的に記述する[14][15][16][17][18][19][20][21]。まず、本論文で扱う最適化問題の具体的な定式化を与える。次に、最適化問題を解くための最適化手法の構造を数学的に記述する。最適化手法を概観することで、3章で行うメタヒューリスティクスの探索構造の解析のための土台を形成する。

2.1 最適化問題の数理

2.1.1 最適化問題の定式化

最適化問題とは、与えられた制約条件の下で、目的を最適に達成するための数理モデルである。本論文で扱う最適化問題は、実数値目的関数 $f: \mathbb{R}^N \rightarrow \mathbb{R}$ と、空集合でない閉集合 $S \subseteq \mathbb{R}^N$ が与えられたとき、集合 S 上で目的関数 $f(x)$ を最小化する連続（実数）最適化問題とし、式 (2.1) として定式化される。

$$\begin{cases} \underset{x}{\text{minimize}} & f(x) \\ \text{subj. to} & x \in S \subseteq \mathbb{R}^N \end{cases} \quad (2.1)$$

目的関数 $f(x)$ の最大化問題を扱う場合であっても、 $f(x)$ の符号を反転させることで、式 (2.1) の最小化問題に帰着できる。

S を実行可能領域といい、その要素 $x \in S$ を実行可能解という。実行可能領域 S は、目的関数に対する制約条件 C と、解に対する上下限制約条件 S_x を用いて、式 (2.2) として定義される。

$$S = S_x \cup C \quad (2.2)$$

目的関数に対する制約条件 C は、ベクトル値不等式制約関数 $g: \mathbb{R}^N \rightarrow \mathbb{R}^L$ と、ベクトル値等式制約関数 $h: \mathbb{R}^N \rightarrow \mathbb{R}^M$ を用いて、式 (2.3) として表す。

$$C = \{x \in \mathbb{R}^N \mid g(x) \geq 0, h(x) = 0\} \quad (2.3)$$

式 (2.3) において、 $g(x) \geq 0$ を不等式制約条件、 $h(x) = 0$ を等式制約条件という。

解に対する上下限制約条件 S_x は、上下限制約 $a, b \in \mathbb{R}^N$ を用いて、式 (2.4) として表す。

$$S_x = \{x \in \mathbb{R}^N \mid a_n \leq x_n \leq b_n, n = 1, \dots, N\} \quad (2.4)$$

2.1.2 局所的最適解と大域的最適解

式 (2.1) の最適化問題において、ある実行可能解 $x^o \in S$ が式 (2.5) を満たす場合、解 x^o を式 (2.1) の最適化問題の大域的最適解といい、大域的最適解 x^o の目的関数値 $f(x^o)$ を大域的最適値という。

$$f(x^o) \leq f(x), \quad \forall x \in S \quad (2.5)$$

式 (2.5) を満たす大域的最適解 x^o の集合を S^o と定義する。

式 (2.1) の最適化問題において、ある実行可能解 $x^* \in S$ が式 (2.6) を満たす場合、解 x^* を式 (2.1) の最適化問題の局所的最適解といい、局所的最適解 x^* の目的関数値 $f(x^*)$ を局所的最適値という。

$$f(x^*) \leq f(x), \quad \forall x \in S \cap B(x^*; \varepsilon) \quad (2.6)$$

ただし、 $B(x^*; \varepsilon)$ は、式 (1.6) の距離空間 (\mathbb{R}^N, d_E) において、ある点 $x^* \in \mathbb{R}^N$ に対する ε -近傍である。式 (2.6) を満たす局所的最適解 x^* の集合を S^* と定義する。

一般的に式 (2.1) の最適化問題は、複数の大域的最適解と局所的最適解を有する。大域的最適解集合 S^o と局所的最適解集合 S^* について、式 (2.7) の関係式が成立する。

$$S^o \subseteq S^* \quad (2.7)$$

ただし, $S^o = \emptyset$ あるいは, $S^* = \emptyset$ となる場合も存在する。 $|S^*| \geq 1$ ($S^* \neq \emptyset$) を仮定して, 大域的最適解あるいは局所的最適解を求めるためのアルゴリズムを最適化手法という。

2.1.3 最適化問題の分類

最適化問題は, 実行可能領域 S や目的関数 f の性質により, いくつかの種類に分類される。特定の種類の問題に対しては, 大域的最適解を得るための有力な最適化手法が知られている。特に, 最適化問題を解くとき, その問題のクラスが既知であれば, そのクラスに合わせた適切な最適化手法を適用することが可能である。

(a) 有制約最適化問題と無制約最適化問題

式 (2.1) の最適化問題に対して, 目的関数に対する制約条件 C あるいは, 解に対する上下限制約条件 S_x が課される場合, 有制約最適化問題という。一方, C と S_x のどちらか一方も課されない, すなわち $S = \mathbb{R}^N$ の場合, 無制約最適化問題といい, 式 (2.8) として定式化される。

$$\underset{x}{\text{minimize}} \quad f(x), \quad x \in \mathbb{R}^N \quad (2.8)$$

現実には存在する多くの最適化問題は, 無制約最適化問題ではなく, 有制約最適化問題に分類される。しかし, 一般の有制約最適化問題において, 制約条件の侵害量をペナルティとして目的関数に与え, 新たな拡大目的関数を構成することで, 無制約最適化問題に変換するアプローチが存在する。これを外点ペナルティ関数法 [14] [15] [16] [19] [20] という。外点ペナルティ関数法を用いることで, 式 (2.3) の目的関数に対する制約条件 C を陽に考慮せずに, 最適化条件を満足する解を得ることができる。

また, 一般に最適化手法によって得る解は, 式 (2.4) の上下限制約条件 S_x を常に満たすとは限らない。そこで, 解 $x \in \mathbb{R}^N$ の暫定位置 $u \in \mathbb{R}^N$ を残しておき, 暫定位置の各要素 $u_n (n = 1, \dots, N)$ に対して適当な閉じ込め関数 $\ell : \mathbb{R} \rightarrow [a_n, b_n]$ を作用させたものを正式な解として採用するアプローチが存在する。閉じ込め関数の例として, 式 (2.9) のリミッティ

ング関数などが挙げられる。

$$x = \ell(u) = \begin{cases} a, & u < a \\ u, & a \leq u \leq b \\ b, & u > b \end{cases} \quad (2.9)$$

(b) 線形計画問題と非線形計画問題

式(2.1)の最適化問題において、目的関数 f 、制約関数 h, g がいずれも線形の場合、線形計画問題という。一方、目的関数 f 、制約関数 h, g のいずれか1つ以上が非線形の場合、非線形最適化問題（非線形計画問題）という。線形計画問題に対して有効な最適化手法は、単体法や内点法などが知られている。しかし、非線形最適化問題に対して有効な最適化手法は、凸性や制約の有無により異なる。

(c) 凸最適化問題と非凸最適化問題

式(2.1)の最適化問題において、目的関数 f が凸関数であり、実行可能領域 S が凸集合である場合、凸最適化問題（凸計画問題）という。代表的な凸最適化問題として、凸2次計画問題が知られている。凸最適化問題において、局所的最適解集合 S^* と大域的最適解集合 S° は一致する。大域的収束性を有する最適化手法を適用すれば、大域的最適解 x° を求めることができる。凸最適化問題の中でも、特に目的関数 f が狭義凸関数であり、実行可能領域 S が凸集合である場合、狭義凸最適化問題という。狭義凸最適化問題は唯一の大域的最適解 x° を有する。

一方、目的関数 f が非凸関数、または実行可能領域 S が非凸集合である場合、非凸最適化問題（非凸計画問題）という。一般の非凸最適化問題において、局所的最適解集合 S^* と大域的最適解集合 S° は一致しない。問題によっては大域的最適解 x° を確実に求めることは困難である。全ての線形計画問題は凸最適化問題だが、非線形最適化問題の中でも、凸最適化問題に分類されるのは、凸2次最適化問題など、一部に限定されている。しかし、非凸計画問題を解く場合でも、凸最適化問題に対する最適化手法を拡張して用いることは有効なアプローチとなることが多い。

2.2 最適化手法の数理

2.2.1 最適化手法の基本構造

本論文では、非凸最適化問題の複数の局所的最適解の中から、目的関数値が最小となるような解を求める最適化手法（大域的最適化手法）について扱う。目的関数 f 、制約条件関数 g, h の連続性・微分可能性は特に仮定しないが、必要に応じて、連続性・微分可能性が仮定される場合に適用可能な最適化手法についても考察する。また本論文では、式 (2.3) の上下限制約条件 S_x が存在しない上に、外点ペナルティ関数法により、式 (2.1) の有制約最適化問題は式 (2.8) の無制約最適化問題に変換されているものとする。したがって、本論文で扱う最適化手法は、無制約最適化問題を対象とする手法に限定する。

一般に、式 (2.8) の最適化問題に対する解法は、直接解法と反復解法に大別される。直接解法は、有限回の解析的な手順で大域的最適解を直接求める方法であり、単体法や Lagrange の未定乗数法などが挙げられる。反復解法は、探索空間内に適当な初期点を与え、アルゴリズムの規則により、点の位置を反復的に更新し、これらを最適解へ近接・収束させる方法である。このときに移動する点を探索点、あるいは個体という。直接解法は適用可能な最適化問題のクラスが極一部に限定される上にその種類が少ないため、一般的な最適化手法は反復法のアプローチにより構成される。反復法における探索点 $x \in \mathbb{R}^N$ の更新式は、式 (2.10) で表される。

$$x(k+1) = x(k) + v(k) \quad (2.10)$$

ただし、 $x(k)$ は反復回数 k における探索点、 $v(k)$ は $x(k)$ から $x(k+1)$ までの移動ベクトルである。最適化手法特有の移動ベクトルを生成することで、探索ダイナミクスは決定される。

反復解法を行う最適化手法の中でも、一つの探索点によって最適解の探索を行う最適化手法を単点型最適化手法という。単点型最適化手法では、探索空間 \mathbb{R}^N 内に、探索点の初期位置 $x \in \mathbb{R}^N$ を与え、アルゴリズムの規則によって、探索点の位置 x を反復的に更新する。この反復の計算で得られる点列 $\chi(k) = \{x(1), x(2), \dots, x(k)\} = \{x(k)\}_{k=1}^k$ を反復回数 k までの探索点 x の探索履歴という。

また、複数の探索点によって最適解の探索を行う最適化手法を多点型最適化手法という。多点型最適化手法では、探索空間 \mathbb{R}^N 内に、複数の探索点の初期位置 $x^i \in \mathbb{R}^N (i = 1, \dots, m)$

を与え、アルゴリズムの規則によって、探索点の位置 x^i を反復的に更新する。ここで、 $m \in \mathbb{N}$ を探索点数、 $i \in \{1, \dots, m\}$ を探索点番号という。探索点同士は独立しておらず、相互に作用し合う要素が含まれる。 m 個の探索点の集団を探索点群といい、反復回数 k における m 個の探索点 $x^i(k) (i = 1, \dots, m)$ の探索点群 $X(k)$ は式 (2.11) で表される。

$$X(k) = \{x^1(k), x^2(k), \dots, x^m(k)\} \quad (2.11)$$

単点の場合と同様に、探索点群に対しても探索履歴を考えることができる。

$\chi(k) = \{X(1), X(1), \dots, X(k)\} = \{X(\kappa)\}_{\kappa=0}^k$ を反復回数 k までの探索点群 X の探索履歴という。

2.2.2 特殊な解・探索点

2.2.1 項で定義した探索点群や探索履歴を用いて、最適化手法において使用する特殊な解・探索点を定義する。本論文で扱う特殊な解・探索点を、式 (2.12)、式 (2.13)、式 (2.14)、式 (2.15)、式 (2.16)、式 (2.17) で定義する。

- p-best (personal best solution : 個別最良解)

$$x^{p-best, i}(k) = \underset{x^i(\kappa) \in \chi^i(k)}{\operatorname{argmin}} \{f(x^i(\kappa)) \mid \kappa = 0, \dots, k\} \quad (2.12)$$

探索点群 $X(k)$ のある探索点 $x^i(k) \in X(k)$ について、反復回数 k までの探索履歴 $\chi^i(k) = \{x^i(1), \dots, x^i(k)\}$ の中で、最も目的関数値が小さな解を p-best といい、 $x^{p-best, i}(k)$ と表す。

- g-best (global best solution : 最良解)

$$x^{g-best}(k) = \underset{x^{p-best, i}(k) \in \chi(k)}{\operatorname{argmin}} \{f(x^{p-best, i}(k)) \mid i = 1, \dots, m\} \quad (2.13)$$

探索点群 $X(k)$ の反復回数 k までの探索履歴 $\chi(k) = \{X(0), \dots, X(k)\}$ の中で、最も目的関数値が小さな解を g-best といい、 $x^{g-best}(k)$ と表す。

- c-best (current best search point : 最良探索点)

$$x^{c-best}(k) = \underset{x^i(k) \in X(k)}{\operatorname{argmin}} \{f(x^i(k)) \mid i = 1, \dots, m\} \quad (2.14)$$

探索点群 $X(k)$ の中で，最も目的関数値が小さな探索点を $c\text{-best}$ といい， $x^{c\text{-best}}(k)$ と表す。

- $c\text{-worst}$ (current worst search point : 最悪探索点)

$$x^{c\text{-worst}}(k) = \operatorname{argmax}_{x^i(k) \in X(k)} \{f(x^i(k)) \mid i = 1, \dots, m\} \quad (2.15)$$

探索点群 $X(k)$ の中で，最も目的関数値が大きな探索点を $c\text{-worst}$ といい， $x^{c\text{-worst}}(k)$ と表す。

- better (better search point : 優良探索点)

$$\begin{aligned} \text{Better}^i(k) = \{x^\ell(k) \in X(k) \mid f(x^\ell(k)) < f(x^i(k)); \\ \ell = 1, \dots, m\} \end{aligned} \quad (2.16)$$

$$x^{\text{better}, i}(k) \in \text{Better}^i(k) \quad (2.17)$$

探索点群 $X(k)$ のある探索点 $x^i(k) \in X(k)$ について $x^i(k)$ よりも目的関数値が小さな探索点群を優良探索点群といい， $\text{Better}^i(k) \subseteq X(k)$ と表す。優良探索点群 $\text{Better}^i(k)$ の中の各探索点を better といい， $x^{\text{better}, i}(k)$ と表す。ただし， $x^i(k) = x^{c\text{-best}}(k)$ ならば， $x^{\text{better}, i}(k) = x^{c\text{-best}}(k)$ とする。

2.3 数理計画法

2.3.1 数理計画法と最適化問題の関連性

数理計画法は，最適解が満たすべき数理的な最適条件を満たす集合に漸近するような反復のアルゴリズムから構成される最適化手法である。数理計画法のアルゴリズムでは，対象問題の解情報，目的関数値情報，解析の情報（勾配や Hesse 行列など）を用いて，数学的に保証された解を求める。数理計画法では，対象問題の解析的情報を要することから，特定の条件や構造を満たす対象問題のクラスに対して適用可能な解法が存在する。例えば，線形計画問題に対する解法は線形計画法，無制約非線形問題に対する解法は無制約非線形計画法のように対応している。また，最適化問題における最適性を活用して解析的に最適

解を求める解法も存在する。これらも最適化問題の制約条件の有無や，等式制約関数あるいは不等式制約関数の有無によって解法が分類される。代表的には，Karush-Kuhn-Tucker条件を活用した解法（Lagrangeの未定乗数法を含む）や双対問題を活用した解法が知られている[14][15][16][19][20]。本論文では，式(2.8)の無制約の非凸かつ非線形最適化問題に対する反復解法のみを扱う。

2.3.2 降下法

多くの無制約の非線形最適化問題に対する反復解法は，単点型最適化手法である。これらの解法では，式(2.10)における移動ベクトルをステップサイズ $\alpha(k) \in \mathbb{R}$ ，探索方向ベクトル $s(k) \in \mathbb{R}^N$ を用いて， $v(k) = \alpha(k)s(k)$ と表す。このため，これらの解法における探索点 $x(k)$ の更新式は，式(2.18)と表すことができる。

$$x(k+1) = x(k) + \alpha(k)s(k) \quad (2.18)$$

式(2.18)の更新式を用いて， $f(x(k+1)) < f(x(k))$ を満たすように探索点を移動させる解法を降下法という。また，降下法において，ステップサイズ $\alpha(k)$ は式(2.19)の直線探索問題を解くことで決定する。

$$\alpha(k) = \underset{\alpha}{\operatorname{argmin}} \{f(x(k) + \alpha s(k)) \mid \alpha \geq 0\} \quad (2.19)$$

直線探索問題は1次元の α に関する最適化問題であり，囲い込みと補間法によって解かれる。本論文では，囲い込みと補間法のアルゴリズムについては記述しない。

以下に降下法のアルゴリズムを示す。アルゴリズムの終了条件を $\|\nabla f(x)\| < \varepsilon$ ($\varepsilon > 0$: 停止基準)とする。

【降下法のアルゴリズム】

Step 0:[初期化]

停止基準 $\varepsilon > 0$ を定め，反復回数を $k = 1$ とする。探索点の初期位置 $x(k)$ を与える。

Step 1:[終了判定]

$\|\nabla f(x(k))\| < \varepsilon$ ならば，探索を終了する。

Step 2:[降下方向の決定]

$x(k)$ における降下方向 $s(k)$ を決定する。

Step 3:[ステップサイズの決定]

直線探索問題

$$\alpha(k) = \underset{\alpha}{\operatorname{argmin}} \{f(x(k) + \alpha s(k)) \mid \alpha \geq 0\}$$

を解き , ステップサイズ $\alpha(k)$ を求める。

Step 4:[探索点の更新]

$$x(k+1) = x(k) + \alpha(k)s(k)$$

より , 探索点 $x(k)$ を更新する。

Step 5:[反復回数の更新]

$k := k + 1$ として **Step 1** へ戻る。

2.3.3 勾配法

降下法の中でも , 探索方向ベクトル s に目的関数の勾配ベクトル $\nabla f(x)$ を使用する解法を勾配法という。勾配法において , 探索方向ベクトル s が降下方向となるための必要十分条件は , 式 (1.10) の $f(x)$ の s に対する方向微分が負となることである。これは式 (2.20) で与えられる。

$$\frac{\partial f(x)}{\partial s} < 0 \Leftrightarrow \nabla f(x)^T s < 0 \quad (2.20)$$

以降では , 勾配法に分類される最適化手法を概観する。勾配法を適用するには , 目的関数 $f(x)$ が連続微分可能である条件が必要である。

式 (2.18) で表される勾配法の更新式により生成される点列 $\{x(k)\}$ について , Zoutendijk 条件が成り立つことが知られており , 式 (2.21) あるいは式 (2.22) で表される。

$$\sum_{k=0}^{\infty} \left(\frac{\nabla f(x(k))^T s(k)}{\|s(k)\|} \right)^2 < \infty \quad (2.21)$$

$$\sum_{k=0}^{\infty} (\|\nabla f(x(k))\| \cos \theta(k))^2 < \infty \quad (2.22)$$

式 (2.22) の $\theta(k)$ は $-\nabla f(x(k))$ と探索方向ベクトル $s(k)$ とのなす角であり, 式 (2.23) で定義される。

$$\cos \theta(k) = \frac{(-\nabla f(x(k)))^T s(k)}{\|\nabla f(x(k))\| \|s(k)\|} \quad (2.23)$$

式 (2.21) あるいは式 (2.22) の Zoutendijk 条件において, 無限級数が収束するための必要条件より, 式 (2.24) あるいは式 (2.25) が成り立つ。

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x(k))^T s(k)}{\|s(k)\|} = 0 \quad (2.24)$$

$$\lim_{k \rightarrow \infty} \|\nabla f(x(k))\| \cos \theta(k) = 0 \quad (2.25)$$

式 (2.24) あるいは式 (2.25) が勾配法における点列 $\{x(k)\}$ の大域的収束性を示している。

勾配法では, 目的関数 $f(x)$ を近似したモデルを局所的に最小化するように, 探索方向ベクトル s を決定する方法がよく用いられる。モデル関数として, 目的関数 $f(x)$ を $x = x(k)$ 付近で Taylor 展開した式 (2.26) が用いられる。

$$f(x(k) + s) \approx f(x(k)) + \nabla f(x(k))^T s + \frac{1}{2} s^T H(f(x(k))) s + \cdots \quad (2.26)$$

ただし, $\nabla f(x(k))$ は式 (1.8) の勾配ベクトルであり, $H(f(x(k)))$ は式 (1.9) の Hesse 行列である。

式 (2.26) の 1 次の項まで Taylor 展開した 1 次近似モデル $\ell(s)$ は式 (2.27) で表され, 2 次の項まで Taylor 展開した 2 次近似モデル $q(s)$ は式 (2.28) で表される。

$$\ell(s) = f(x(k)) + \nabla f(x(k))^T s \quad (2.27)$$

$$q(s) = f(x(k)) + \nabla f(x(k))^T s + \frac{1}{2} s^T H(f(x(k))) s \quad (2.28)$$

2.3.4 最急降下法

最急降下法は、探索方向ベクトル s として、式 (2.27) の 1 次近似モデル $\ell(s)$ を局所的に最小にする方向を設定する勾配法である。つまり、 s は、方向微分が最も小さくなる方向に設定される。これは、式 (2.23) の $-\nabla f(x(k))$ と $s(k)$ とのなす角について $\theta(k) = 0$ となることであるから、 s は式 (2.29) で与えられる。

$$s(k) = -\nabla f(x(k)) \quad (2.29)$$

式 (2.20) より、最急降下法の探索方向は目的関数の降下方向となる。最急降下法における探索方向ベクトルは、目的関数の等高線に対する法線ベクトルとなる。

s が式 (2.29) で与えられたとき、式 (2.24) あるいは式 (2.25) の Zoutendijk 条件を満たす。これは、最急降下法の点列は大域的収束性が保証されていることを示している。最急降下法のアルゴリズムは、降下法のアルゴリズムの探索方向ベクトル s を式 (2.29) で決定したものであるため、具体的なアルゴリズムについては記述しない。

2.3.5 Newton-Raphson 法

Newton-Raphson 法は、式 (2.28) の 2 次近似モデル $q(s)$ の最小化に基づく勾配法である。最適性条件より、 $q(s)$ を最小化するためには、 $\nabla q(s) = 0$ を満たすベクトル s を求めればよい。この方程式を Newton 方程式といい、Newton 方程式の解 s を Newton 方向という。Newton-Raphson 法は、探索方向ベクトル $s(k)$ を Newton 方向に設定する。

式 (2.28) の 2 次近似モデル $q(s)$ をベクトル s で微分すると、式 (2.30) が得られる。

$$\nabla q(s) = \nabla f(x(k)) + H(f(x(k)))s \quad (2.30)$$

ただし、 $H(f(x))$ は、式 (1.9) の Hesse 行列である。式 (2.30) を用いると、Newton 方程式は式 (2.31) として表すことができる。

$$H(f(x(k)))s = -\nabla f(x(k)) \quad (2.31)$$

式 (2.31) を解くことで、Newton 方向は式 (2.32) として求められる。

$$s(k) = -H(f(x(k)))^{-1} \nabla f(x(k)) \quad (2.32)$$

ただし, Hesse 行列 $H(f(x(k)))$ は正則であるとする。さらに, Hesse 行列 $H(f(x(k)))$ が正定値ならば, 式 (2.32) の Newton 方向を用いると, 方向微分について式 (2.33) が成立する。

$$\nabla f(x(k))^T s(k) = -\nabla f(x(k))^T H(f(x(k)))^{-1} \nabla f(x(k)) < 0 \quad (2.33)$$

式 (2.20) より, Hesse 行列が正定値ならば, Newton 方向は目的関数の降下方向となる。Newton 方向は, 点 $x(k)$ において目的関数の等高線の接線と曲率を共有する楕円を考え, 点 $x(k)$ から楕円の中心へ向かう方向ベクトルとなる。

本論文では証明を行わないが, Newton-Raphson 法の点列は 2 次局所的収束性が保証されている。そのため, 収束速度の点で優れた代表的な最適化手法である。しかし一般に, ① Hesse 行列は正則である保証はないため, 探索方向を計算できるとは限らない, ② Hesse 行列は正定値である保証はないため, 探索方向が目的関数の降下方向になるとは限らない。Newton-Raphson 法のアルゴリズムは, 降下法のアルゴリズムの探索方向ベクトル s を式 (2.32) で決定したものであるため, 具体的なアルゴリズムについては記述しない。

2.3.6 準 Newton 法

準 Newton 法は, Newton-Raphson 法の収束速度を活用しながら, 計算負荷に関する欠点を軽減するように改良された勾配法である。具体的には, Hesse 行列を正定値対称行列で近似することで, 探索方向が目的関数の降下方向になると同時に, Hesse 行列の計算負荷を軽減できる。

準 Newton 法では, Hesse 行列を近似する場合と Hesse 行列の逆行列を近似する場合がある。本論文では, Hesse 行列を正定値対称行列 $B(k)$ で近似する場合を考える。式 (2.28) の 2 次近似モデル $q(s)$ における Hesse 行列を正定値対称行列 $B(k) \in \mathbb{R}^{N \times N}$ に置き換えた, 式 (2.34) で表される新たな 2 次近似モデル $Q(s)$ を考える。

$$Q(s) = f(x(k)) + \nabla f(x(k))^T s + \frac{1}{2} s^T B(k) s \quad (2.34)$$

Newton-Raphson 法と同様に, 2 次近似モデル $Q(s)$ に対する Newton 方程式を解くことで, 準 Newton 法の探索方向ベクトル s は式 (2.35) で与えられる。

$$s(k) = -B(k)^{-1} \nabla f(x(k)) \quad (2.35)$$

$B(k)$ は正定値対称行列であるため、方向微分は負となる。したがって、式 (2.20) より、準 Newton 法の探索方向ベクトルは目的関数の降下方向となる。また、正定値行列は正則であるため、 $B(k)$ は正則行列となる（逆行列が存在する）。

$B(k+1) \in \mathbb{R}^{N \times N}$ が正定値対称行列となるためには、セカント条件を満たす必要がある。セカント条件は式 (2.36) で与えられる。

$$B(k+1)(x(k+1) - x(k)) = \nabla f(x(k+1)) - \nabla f(x(k)) \quad (2.36)$$

準 Newton 法では、式 (2.36) のセカント条件を満たすように $B(k)$ を更新して、 $B(k+1)$ が生成される。セカント条件を満たす $B(k+1)$ の更新式はこれまでに多く提案されている。代表的な更新式として、DFP (Davidon Fletcher-Powell) 公式や BFGS (Broyden Fletcher Goldfarb Shanno) 公式が知られている[14][15][16][20]。Hesse 行列の逆行列を近似する場合は、正定値対称行列 $B(k)$ の逆行列を直接置き換えた行列に対して更新式を与えればよい。この場合、逆行列の計算が必要なくなるため、計算負荷をさらに軽減させることができる。本論文では証明を行わないが、BFGS 公式を用いた準 Newton 法の点列は局所的に超 1 次収束し、大域的収束性を満たすことが保証されており、他の公式においてもこれと類似の性質が示されている[14][20]。このことから、準 Newton 法は Newton-Raphson 法の局所的収束の速さを活かしながら、大域的収束性を満たすため、既存の無制約最適化手法の中でも最も有効な手法として知られている。

以下に準 Newton 法のアルゴリズムを示す。アルゴリズムの終了条件を $\|\nabla f(x)\| < \varepsilon$ ($\varepsilon > 0$) とする。本論文では記述しないが、方向微分の符号が正になった場合、近似行列 $B(k)$ を初期化するアルゴリズムも存在する。

【Hesse 行列を近似する準 Newton 法のアルゴリズム】

Step 0:[初期化]

停止基準 $\varepsilon > 0$ を定め、反復回数を $k = 1$ とする。探索点の初期位置 $x(k)$ 、近似行列 $B(k) = I$ を与える。ただし、 $I \in \mathbb{R}^{N \times N}$ は単位行列である。

Step 1:[終了判定]

$\|\nabla f(x(k))\| < \varepsilon$ ならば、探索を終了する。

Step 2:[降下方向の決定]

$x(k)$ における降下方向 $s(k)$ を

$$s(k) = -B(k)^{-1} \nabla f(x(k))$$

より決定する。

Step 3:[ステップサイズの決定]

直線探索問題

$$\alpha(k) = \underset{\alpha}{\operatorname{argmin}} \{f(x(k) + \alpha s(k)) \mid \alpha \geq 0\}$$

を解き，ステップサイズ $\alpha(k)$ を求める。

Step 4:[探索点の更新]

$$x(k+1) = x(k) + \alpha(k)s(k)$$

より，探索点 $x(k)$ を更新する。

Step 5:[近似行列の更新]

公式を用いて，近似行列 $B(k)$ を更新する。

Step 6:[反復回数の更新]

$k := k + 1$ として **Step 1** へ戻る。

2.4 メタヒューリスティクス

メタヒューリスティクスは，生物現象や物理現象などの経験的に優れたメカニズムに基づくアルゴリズムから構成される最適化手法である。メタヒューリスティクスのアルゴリズムでは，対象問題の解情報と目的関数値情報のみを用いて，探索点が解空間内を逐次移動し，最適解あるいは準最適解を探索する。特に多点型最適化手法に分類されるメタヒューリスティクスを多点探索型メタヒューリスティクスという。数理計画法とは異なり，求める解に数学的な保証がないため，探索点同士に経験的に優れたメカニズムに基づく相互作用

用を与えることで、多様なダイナミクスを生み出し、そのダイナミクスを解の探索に巧みに活用する。

代表的なメタヒューリスティクスとして、Particle Swarm Optimization[28]や、Differential Evolution[29]が知られている。1.1節でも述べたように、これらの手法は、高い探索性能・汎用性を有していることから、近年では、産業応用の場面でも使用されることが多い。しかし、メタヒューリスティクスの課題点として、① 数学的な背景を有していないため、得られた解に対する最適性は保証されていないこと、② 十分な探索性能の発揮には、パラメータ設定に関する使用者の専門的な知識・経験や試行錯誤を必要とすること、がよく指摘される。これに対して、近年では、メタヒューリスティクスの実システムへの応用や更なる有用性の向上のために、探索ダイナミクスに対する数理的・数値実験的解析や、解析に基づく手法の改良・開発に関する研究が盛んになっている[34][35][36][37][38][39][40][41][42]。今後も、数値実験や解析を通じて、① パラメータの設定・調整に対する有効な知見の抽出による系統的な分類・整理、② パラメータの調整方法の開発、③ メタヒューリスティクスの設計論の確立、などの更なる発展が期待されている。例えば、数理計画法においても、効率的な探索を行うための理論・技術は整備されている。関数の解析的情報に基づかない理論・技術であれば、メタヒューリスティクスに対して適用することは、上述の課題解決に資すると考えられる。

また、2.3節まで様々な数理計画法を概観したように、数理計画法は「対象の問題の特徴を大いに把握し、その情報を最大限に活用して数理的に保証された方向へ探索を行う」ことが基本概念である。これは、1.1節でも述べたように、対象の問題の特徴を大いに把握するために、あるいは数理的に解の最適性・収束性を保証するために、探索に関数の解析的情報が必要となった結果、「適用可能な問題のクラス」が限定されることを意味する。言い換えると、数理計画法は「適用範囲は限定されるが、適用範囲内の問題に対する究極的な適応能力を有している」といえる。これに対して、メタヒューリスティクスは、数理計画法とは構築のプロセスが全く異なるため、勾配に基づく解の最適性・収束性を議論することができない。そのため、「探索点群による多様なダイナミクスによって、対象の問題の特徴を把握し、その情報を最大限に活用した探索を行う」ことが基本概念である。したがって、数理計画法とメタヒューリスティクスの共通点を考えると、優れた探索性能を有する最適化手法を構築するには、優れた適応能力を具備することが必要となる。

本論文では、メタヒューリスティクスの探索構造と適応能力の関係に着目する。様々な

メタヒューリスティクスの探索構造の解析を行い，統一的な視点に基づき，メタヒューリスティクスの適応能力・探索性能に資する探索戦略を系統的に分類・定義する。これらの解析を踏まえることで，上記の課題の③であるメタヒューリスティクスの設計論となる汎用的フレームワークの確立と，汎用的フレームワークに基づく優れた適応能力を有するメタヒューリスティクスの開発を目的とする。

3

メタヒューリスティクス の探索構造の解析

本章では、これまでに提案されているメタヒューリスティクスのアルゴリズムを概観する。さらに、探索構造を解析し、メタヒューリスティクスに共通する探索戦略を抽出する。4章以降でメタヒューリスティクスに対して統一的な視点で議論するための土台を形成する。

3.1 メタヒューリスティクスの基本的な探索構造

本章では、様々なメタヒューリスティクスのアルゴリズムを概観しながら、探索構造を解析する。近年ではメタヒューリスティクスの探索ダイナミクスを数理的に明らかにする研究も増えているが、本論文では更新式に従って生成される探索点の位置について定性的に考察する。

図 3.1 にメタヒューリスティクスの基本的な探索構造を示す。図 3.1 は、 x_1, x_2 軸で構成される 2 次元の解空間における探索の様子を示している。メタヒューリスティクスは、① 近傍の生成、② 探索点の更新、という普遍的な探索構造を有している[22][23]。近傍の生成では、探索点 $x(k)$ が移動する可能性がある領域（近傍）を生成し、近傍の中で解 $\hat{x}(k)$ （近傍解）を生成する。探索点の更新では、探索点 $x(k)$ が近傍解 $\hat{x}(k)$ へ更新し、 $x(k+1)$ となる。この更新式は、移動ベクトル $v(k)$ を用いて、 $x(k+1) = x(k) + v(k)$ と表される。これは、式 (2.10) の反復解法の更新式と一致する。このように、メタヒューリスティクスでは、「近傍解 $\hat{x}(k)$ を生成し、探索点 $x(k)$ が近傍解へ更新する」という動作を繰り返すことで、

解空間内を探索する。

また，探索点の更新方法は大きく「絶対移動」と「改善移動」の二種類に分類される。絶対移動では，近傍解 $\hat{x}(k)$ の目的関数値に関係なく，探索点 $x(k)$ は近傍解へ移動する。これに対して，改善移動では，探索点 $x(k)$ の目的関数値よりも近傍解 $\hat{x}(k)$ の目的関数値が良い，つまり $f(\hat{x}(k)) < f(x(k))$ を満たす場合のみ，近傍解へ移動する。それぞれの場合の移動ベクトルは，式 (3.1)，式 (3.2) として表される。

$$v(k) = \hat{x}(k) - x(k) \quad (3.1)$$

$$v(k) = \begin{cases} \hat{x}(k) - x(k), & f(\hat{x}(k)) < f(x(k)) \\ 0, & \text{otherwise} \end{cases} \quad (3.2)$$

改善移動では，近傍解 $\hat{x}(k)$ の目的関数値によって，探索点 $x(k)$ が移動しない場合もある。さらに，更新方法の種類によって，2.2.2 項で述べた特殊な解・探索点の捉え方が異なる。絶対移動では，2.2.2 項で述べた特殊な解・探索点の捉え方をそのまま活用できる。これに対して，改善移動では，目的関数値が改善しなければ探索点 $x(k)$ は移動しないため，各探索点 $x(k)$ は式 (2.12) の p-best，式 (2.14) の c-best は式 (2.13) の g-best と一致する。

しかし，どちらの更新方法においても，更新が行われる場合の移動ベクトルは $v(k) = \hat{x}(k) - x(k)$ となるため，近傍解の生成方法が探索ダイナミクスを決定づけるといえる。さらに，多くの手法の更新式では，乱数が使用され，近傍解の多様性の獲得に貢献している。本論文では，乱数などの確率的要素によって拡大された近傍を摂動と呼ぶことにする。したがって，手法の探索性能に対しては，摂動に基づく近傍が重要な役割を果たすことが考えられる。

ところで，工学における最適化問題の多くは偏り構造を有しており，近接最適性原理 (Proximate Optimality Principle: POP) が成り立つことが知られている[22] [23]。POP とは「良い解同士は類似構造を持つ」という原理のことである。本論文では，連続値最適化問題における「良い解」と「類似構造」を以下のように解釈する。

- 良い解：優れた評価値を持つ解
- 類似構造：解同士の距離が近いこと

数理計画法では問題の解析的情報を用いることで問題構造を把握し，その情報を活用することで優れた解を求めている。これに対して，メタヒューリスティクスは自然現象・物理

現象などの経験的・発見的に優れたメカニズムを基に構築されており，最適化手法としては問題の解析的情報を用いずに探索を行う解直接探索法である。メタヒューリスティクスでは，POPに基づいて探索構造が構成されており，解空間の偏り構造を把握・活用することで，解直接探索法として効率的な探索を行う。

以上から本論文では，探索構造の観点から様々なメタヒューリスティクスを解析し，解空間の偏り構造を把握・活用するための，近傍解 $\hat{x}(k)$ の生成方法について重点的に考察する。解析・考察の結果から，メタヒューリスティクスの優れた探索構造・戦略を抽出する。

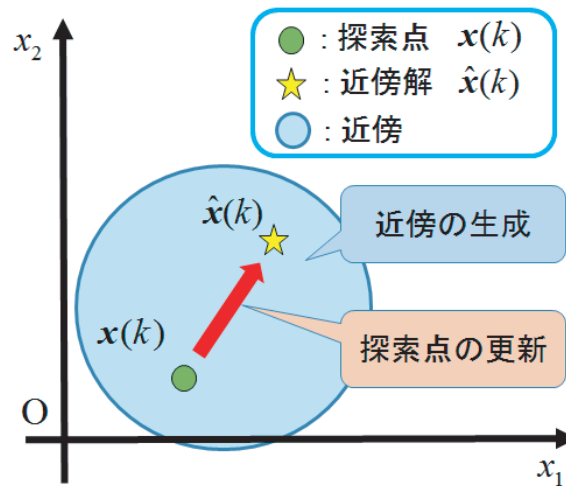


図 3.1：メタヒューリスティクスの探索構造

3.2 Particle Swarm Optimization

3.2.1 Particle Swarm Optimization の概要

Particle Swarm Optimization (PSO) [28] は，1995年にJames KennedyとRussell Eberhartにより開発された，連続値最適化問題を対象とする多点探索型メタヒューリスティクスである。PSOは鳥や魚の群としての採餌行動のメカニズムに基づいており，群全体で情報を共有する探索を行う。これは，探索点間で互いに良い情報を共有し，それを活用している。

また，PSO は，簡潔なアルゴリズムでありながらも，目的関数の勾配情報を活用せず，大域的最適化性能を有することから，実問題への応用研究が盛んになっている[1][8][22]。

3.2.2 Particle Swarm Optimization のアルゴリズム

PSO の更新式を式 (3.3)，式 (3.4) に示す。

$$v^i(k+1) = w v^i(k) + c_1 R_1^i u^{p-best} + c_2 R_2^i u^{g-best} \quad (3.3)$$

$$x^i(k+1) = x^i(k) + v^i(k+1) \quad (3.4)$$

ここで， $w, c_1, c_2 \geq 0$ は各項の係数パラメータ， $u^{p-best} = x^{p-best, i}(k) - x^i(k)$ ， $u^{g-best} = x^{g-best}(k) - x^i(k)$ ， $x^{p-best, i}(k)$ は式 (2.12) で定義される p-best， $x^{g-best}(k)$ は式 (2.13) で定義される g-best， $R_1^i, R_2^i \in \mathbb{R}^{N \times N}$ は対角行列であり，各対角要素は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従う乱数である。

PSO では，移動ベクトル $v(k+1)$ を，移動ベクトル $v(k)$ ， $x^i(k)$ から $x^{p-best, i}(k)$ に向かう差分ベクトル u^{p-best} ， $x^i(k)$ から $x^{g-best}(k)$ に向かう差分ベクトル u^{g-best} の線形結合として表す。

以下に PSO のアルゴリズムを示す。アルゴリズムの終了条件を $k = k_{\max}$ (k_{\max} ：最大反復回数) とする。目的関数値の計算回数を評価回数 T といい， $T = m(k_{\max} + 1)$ となる。本論文では，探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。このとき， IS の各次元のサイズは $b = |c - a|$ となる。

【Particle Swarm Optimization のアルゴリズム】

Step 0:[準備]

探索点数 m ，パラメータ $w, c_1, c_2 \geq 0$ ，最大反復回数 k_{\max} を定め，反復回数を $k = 1$ とする。

Step 1:[初期化]

探索点の初期位置 $x^i(k)$ ($i = 1, 2, \dots, m$) を初期配置領域 IS 内にランダムに与え，探索点の初期移動ベクトル $v^i(k) \in \mathbb{R}^N$ をランダムに与え，初期の p-best を

$\mathbf{x}^{p-best, i}(k) = \mathbf{x}^i(k)$ とする。探索点群の初期の g-best を

$$\mathbf{x}^{g-best}(k) = \underset{\mathbf{x}^{p-best, i}(k)}{\operatorname{argmin}} \{f(\mathbf{x}^{p-best, i}(k)) \mid i = 1, \dots, m\}$$

とする。

Step 2:[移動ベクトルの生成]

各探索点の移動ベクトルを

$$\mathbf{v}^i(k+1) = w \mathbf{v}^i(k) + c_1 \mathbf{R}_1^i \mathbf{u}^{p-best} + c_2 \mathbf{R}_2^i \mathbf{u}^{g-best}$$

より, 生成する。ただし, $\mathbf{u}^{p-best} = \mathbf{x}^{p-best, i}(k) - \mathbf{x}^i(k)$, $\mathbf{u}^{g-best} = \mathbf{x}^{g-best}(k) - \mathbf{x}^i(k)$, $\mathbf{R}_1^i, \mathbf{R}_2^i \in \mathbb{R}^{N \times N}$ は対角行列であり, 各対角要素は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従う乱数である。

Step 3:[探索点の更新]

各探索点を

$$\mathbf{x}^i(k+1) = \mathbf{x}^i(k) + \mathbf{v}^i(k+1)$$

より更新する。

Step 4:[p-best, g-best の更新]

各探索点について, p-best よりも優れている探索点番号の集団を

$$P = \{i \mid f(\mathbf{x}^i(k+1)) < f(\mathbf{x}^{p-best, i}(k)); i = 1, \dots, m\}$$

とする。各探索点の p-best, 探索点群の g-best を

$$\mathbf{x}^{p-best, i}(k+1) = \begin{cases} \mathbf{x}^i(k+1), & i \in P \\ \mathbf{x}^{p-best, i}(k), & \text{otherwise} \end{cases}$$

$$\mathbf{x}^{g-best}(k+1) = \underset{\mathbf{x}^{p-best, i}(k+1)}{\operatorname{argmin}} \{f(\mathbf{x}^{p-best, i}(k+1)) \mid i = 1, \dots, m\}$$

より更新する。

Step 5:[終了判定]

$k = k_{\max}$ ならば, 探索を終了する。さもなければ, $k := k+1$ とし, Step 2 へ戻る。

3.2.3 Particle Swarm Optimization の探索構造の解析

PSO の更新式に含まれる一様乱数は一定の摂動を与えることができるため、PSO は乱数によって、探索の多様性を獲得し、探索性能に貢献している。しかし、乱数の使用により、探索点の動特性の解析を困難としており、試行錯誤的にパラメータを設定する必要がある。そこで、探索点を大域的最適解あるいは局所的最適解へ安定的に収束させるために、PSO の探索モデルを力学系と捉えて、その安定性の解析を行う研究が行われている。これを発端とし、乱数を定数と近似した縮約モデルや、離散時間線形確率システムとして PSO を捉えることで、探索点の収束性を数理的に導出しようという研究が増えている。また、式 (3.3) の更新式に含まれる wv の項を慣性項という。最急降下法などの更新式に慣性項を付加した手法をモーメント法といい、探索の振動抑制や探索性能の向上に寄与することが検証されている。このように、PSO の探索ダイナミクスを数理的に明らかにすることで、有用性を向上させようとする研究は盛んになっている[35][36][37][38]

これに対して、本論文では、探索ダイナミクスを数理的に明らかにすることまでに至らないが、更新式に従って生成される近傍について重点的に考察する。図 3.2 に、PSO の近傍生成について示す。式 (3.3) の更新式で表されるように、移動ベクトル $v(k+1)$ は、移動ベクトル $v(k)$ 、 $x^i(k)$ から $x^{p-best, i}(k)$ に向かう差分ベクトル u^{p-best} 、 $x^i(k)$ から $x^{g-best}(k)$ に向かう差分ベクトル u^{g-best} の線形結合として表され、近傍解 $\hat{x}^i(k)$ は乱数による摂動に基づく近傍内に生成される。

また、PSO は絶対移動であるため、探索点 $x^i(k)$ は近傍解 $\hat{x}^i(k)$ へ必ず更新する。しかし、PSO の更新式には、良い解である p-best と g-best が含まれている。 $x^{g-best}(k)$ と $x^{p-best, i}(k)$ が移動する場合、式 (3.5)、式 (3.6) が成立する。

$$f(x^{g-best}(k+1)) \leq f(x^{g-best}(k)) \quad (3.5)$$

$$f(x^{p-best, i}(k+1)) \leq f(x^{p-best, i}(k)) \quad (3.6)$$

式 (3.5) と式 (3.6) は、探索点 $x^i(k)$ は必ず改善する方向へ移動しないが、g-best と p-best は必ず改善する方向へ移動することを意味している。g-best は c-best を近傍解、p-best は探索点 $x^i(k)$ を近傍解とみなすことができ、g-best と p-best は目的関数値が改善した場合に移動する「改善移動」である。つまり、g-best と p-best に関しては弱い降下条件を満たすた

め、探索点群としては降下方向に移動する。これは、「探索過程で得た g-best と p-best によって問題構造を把握し、その情報を活用した探索を行う」という POP に基づく探索構造であることを示している。したがって、PSO は探索点 $x^i(k)$ は必ずしも降下条件を満たさないが、弱い降下条件を満たす g-best と p-best に向かう差分ベクトルを活用することで優れた近傍を生成し、探索点群として降下条件を満たすように移動する。

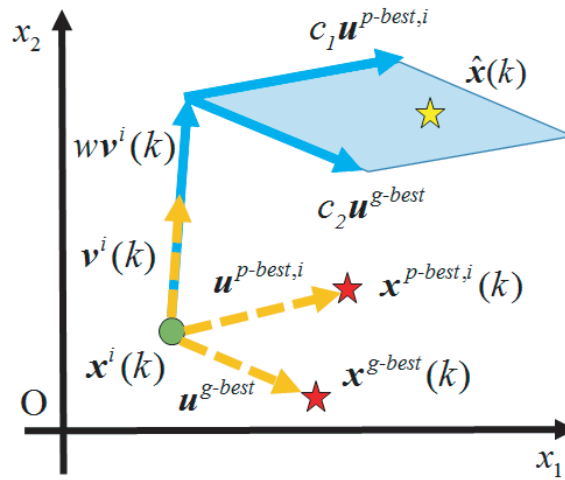


図 3.2 : Particle Swarm Optimization の近傍の生成

3.3 Firefly Algorithm

3.3.1 Firefly Algorithm の概要

Firefly Algorithm (FA) [27][31] は、2008 年に Xin-She Yang により開発された、連続値最適化問題を対象とする多点探索型メタヒューリスティクスである。FA は蛍の求愛行動のメカニズムに基づいており、相対的に強い光を発する蛍に近づく探索を行う。FA では、探索点群が複数の群に分かれる性質を持つため、この性質を活用した改良研究や応用研究も行われている。

3.3.2 Firefly Algorithm のアルゴリズム

FA では各探索点 $x^i(k)$ は固有の光強度 $I^i(k)$ を有しており，他の探索点 $x^s(k) \in X(k)$ を参照して移動する。探索点 $x^i(k)$ の移動は，光強度 $I^i(k)$ に基づく条件を満たす他の探索点 $x^s(k)$ を全て参照するために複数回行われ，全ての探索点が移動し終えたら，探索点を評価する。光強度 $I^i(k)$ は式 (3.7) として定義される。

$$I^i(k) = \frac{1}{|f(x^{c-best}(k)) - f(x^i(k))| + 1} \quad (3.7)$$

ここで， $x^{c-best}(k)$ は式 (2.14) で定義される c-best である。つまり，探索点群 $X(k)$ において探索点 $x^i(k)$ の目的関数値が小さければ，光強度 $I^i(k)$ は大きくなることから，探索点の良さが蛍が発する光強度に相当する。上述の光強度 $I^i(k)$ に基づく条件は $I^i(k) < I^s(k)$ であるため，この条件は $f(x^i(k)) > f(x^s(k))$ の関係と等価である。したがって，探索点 $x^i(k)$ が参照する探索点 $x^s(k)$ は，式 (2.17) で定義される better と等価である。

FA の更新式を式 (3.8)，式 (3.9)，式 (3.10) に示す。

$$v^i(k) = \beta u^{better} + s \quad (3.8)$$

$$\beta = \beta_0 \exp[-\gamma \|u^{better}\|^2] \quad (3.9)$$

$$x^i(k) := x^i(k) + v^i(k) \quad (3.10)$$

ここで， $s = \alpha \varepsilon^i$ ， $\alpha, \beta_0, \gamma \geq 0$ はパラメータ， $u^{better} = x^{better, i}(k) - x^i(k)$ ， $x^{better, i}$ は式 (2.17) で定義される better， $\varepsilon^i \in \mathbb{R}^N$ は一様乱数ベクトルであり，各要素 ε_n^i は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(-0.5, 0.5)$ に従う乱数である。

FA では，移動ベクトル v を， x^{better} 方向の差分ベクトル u^{better} と，乱数ベクトル s の線形結合で表す。

以下に FA のアルゴリズムを示す。アルゴリズムの終了条件を $k = k_{\max}$ とする。評価回数は $T = m(k_{\max} + 1)$ となる。探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。

Step 0:[準備]

探索点数 m , パラメータ $\alpha, \beta_0, \gamma \geq 0$, 最大反復回数 k_{\max} を定め , 反復回数を $k = 1$ とする。

Step 1:[初期化]

探索点の初期位置 $x^i(k)$ ($i = 1, 2, \dots, m$) を初期配置領域 IS 内にランダムに与える。保存位置 $y^i(k) = x^i(k)$ とする。

Step 2:[探索点のランキング]

最良探索点 $x^{c-best}(k)$ と各探索点の光強度 $I^i(k)$ を

$$x^{c-best}(k) = \underset{x^i(k)}{\operatorname{argmin}} \{f(x^i(k)) \mid i = 1, \dots, m\}$$

$$I^i(k) = \frac{1}{|f(x^{c-best}(k)) - f(x^i(k))| + 1}$$

より求め , $I^i(k)$ の非減少順に探索点 $x^i(k), y^i(k)$ を並べ替える。 $i = 1$ とする。

Step 3:[探索点の移動]

$I^i(k) < I^s(k)$ を満たす全ての探索点 $y^s(k)$ の集団を優良探索点群 $Better^i(k)$ とする。

$s = i + 1$ とする。

Step 3-1:[探索点の移動]

$i < m$ ならば , 優良探索点群 $Better^i(k)$ に含まれる参照点 $y^s(k)$ を参照し , 探索点 $x^i(k)$ を

$$v^{is}(k) = \beta(y^s(k) - x^i(k)) + \alpha\epsilon^{is}$$

$$\beta = \beta_0 \exp[-\gamma \|y^s(k) - x^i(k)\|^2]$$

$$x^i(k) := x^i(k) + v^{is}(k)$$

より移動する。

$i = m$ ならば , 探索点 $x^i(k)$ を

$$x^i(k) := x^i(k) + \alpha\epsilon^{is}$$

より移動する。

$\varepsilon^{is} \in \mathbb{R}^N$ は一様乱数ベクトルであり, 各要素 ε_n^{is} は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(-0.5, 0.5)$ に従う乱数である。

Step 3-2:[移動する探索点の変更]

$s < m$ ならば, $s := s + 1$ とし, Step 3-1 へ戻る。 $s = m$ かつ $i < m$ ならば, $i := i + 1$ とし, Step 3 へ戻る。

Step 4:[探索点の更新]

各探索点 $x^i(k)$ ($i = 1, \dots, m$) と保存位置 $y^i(k)$ を

$$x^i(k+1) = x^i(k)$$

$$y^i(k+1) = x^i(k)$$

より更新する。

Step 5:[終了判定]

$k = k_{\max}$ ならば, 探索を終了する。さもなければ, $k := k + 1$ とし, Step 2 へ戻る。

3.3.3 Firefly Algorithm の探索構造の解析

更新式に従って生成される近傍について重点的に考察する。FA では各探索点が複数回移動するが, ここでは一度の移動である式 (3.8) を扱う。図 3.3 に, FA の近傍生成について示す。式 (3.8) の更新式で表されるように, 移動ベクトル v は, x^i から $x^{\text{better}, i}$ に向かう差分ベクトル u^{better} , 一様乱数ベクトル ε^i の線形結合として表され, 近傍解 \hat{x}^i は ε^i による摂動に基づく近傍内に生成される。FA は絶対移動であるため, 探索点 x^i は近傍解 \hat{x}^i へ必ず移動する。

さらに, 差分ベクトル u^{better} の係数パラメータである β は, 式 (3.9) で表されるように, $\|u^{\text{better}}\|$ を変数とする正規分布に従う。このため, x^i と $x^{\text{better}, i}$ の距離が小さければ, β は大きくなり, 距離が大きければ, β は小さくなる。これは, 式 (3.9) における第一の項 βu^{better} を決定付ける。つまり, x^i と $x^{\text{better}, i}$ の距離が極端に小さい場合, $\beta u^{\text{better}} \simeq \beta_0 u^{\text{better}}$ となり, $\beta_0 = 1$ のとき x^i は $x^{\text{better}, i}$ の近くへ移動する。これに対して, x^i と $x^{\text{better}, i}$ の距離が極端に大きい場合, $\beta u^{\text{better}} \simeq 0$ となり, 第一の項 βu^{better} は x^i の移動にほぼ影響を与えない。

したがって、FA は、距離が近い探索点同士が集まるダイナミクスとなり、探索点群は複数の群に分かれる性質を有している。

FA の更新式には、良い解である *better* が含まれている。そのため、各探索点は探索点群の中でも比較的解が良く、距離が近い探索点の付近に集まる。また、差分ベクトル u^{better} は探索過程でノルムが変化するため、探索点群のスケールに適した近傍解を生成することが可能である。これは、「*better* によって問題構造を把握し、その情報を活用した探索を行う」という POP に基づく探索構造であることを示している。したがって、FA は探索点 x^i は必ずしも降下条件を満たさないが、良い探索点である *better* に向かう差分ベクトルを活用することで優れた近傍を生成し、探索点群として良い領域に集まるように移動する。

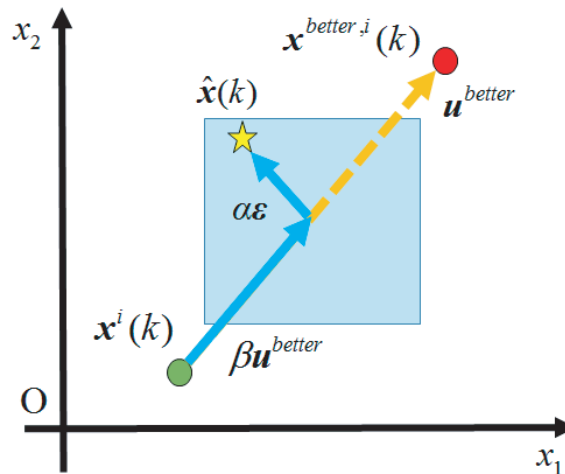


図 3.3 : Firefly Algorithm の近傍の生成

3.4 Differential Evolution

3.4.1 Differential Evolution の概要

Differential Evolution (DE) [27] [31] は、1995 年に Rainer Storn と Kenneth Price により開発された、連続値最適化問題を対象とする多点探索型メタヒューリスティクスである。

DE は生物の進化過程のメカニズムに基づいており，突然変異，交叉，選択による探索を行う。PSO と同様に，DE についても数理的な解析に関する研究や応用研究がますます増えている[39] [40] [41] [42]。

3.4.2 Differential Evolution のアルゴリズム

DE の更新式を式 (3.11)，式 (3.12)，式 (3.13)，式 (3.14) に示す。

$$\mathbf{y}^i(k) = \mathbf{x}^{r_1}(k) + F\mathbf{u}^{between} \quad (3.11)$$

$$\hat{\mathbf{x}}_n^i(k) = \begin{cases} \mathbf{y}_n^i(k), & \theta < CR \cup n = \lambda \\ \mathbf{x}_n^i(k), & \text{otherwise} \end{cases} \quad (3.12)$$

$$\mathbf{v}^i(k) = \begin{cases} \hat{\mathbf{x}}^i(k) - \mathbf{x}^i(k), & f(\hat{\mathbf{x}}^i(k)) < f(\mathbf{x}^i(k)) \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (3.13)$$

$$\mathbf{x}^i(k+1) = \mathbf{x}^i(k) + \mathbf{v}^i(k) \quad (3.14)$$

ただし， $F, CR \geq 0$ はパラメータ， $\mathbf{u}^{between} = \mathbf{x}^{r_2}(k) - \mathbf{x}^{r_3}(k)$ ， r_1, r_2, r_3 ($r_1 \neq r_2 \neq r_3 \neq i$) は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数， θ は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従う乱数， λ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, N)$ に従う乱数である。

DE では，解 \mathbf{y} を，探索点 \mathbf{x}^{r_1} を基準として，探索点間の差分ベクトル $\mathbf{u}^{between}$ で生成する。近傍解 $\hat{\mathbf{x}}$ を探索点 \mathbf{x} と解 \mathbf{y} から生成する。移動ベクトル \mathbf{v} を，目的関数値により，近傍解 $\hat{\mathbf{x}}$ と探索点 \mathbf{x} の差分ベクトルか，零ベクトル $\mathbf{0}$ のどちらかを選択する。

以下に DE のアルゴリズムを示す。アルゴリズムの終了条件を $k = k_{\max}$ とする。評価回数は $T = m(k_{\max} + 1)$ となる。探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。

【Differential Evolution のアルゴリズム】

Step 0:[準備]

探索点数 m ，パラメータ $F, CR \geq 0$ ，最大反復回数 k_{\max} を定め，反復回数を $k = 1$ とする。

Step 1:[初期化]

探索点の初期位置 $x^i(k)$ ($i = 1, 2, \dots, m$) を初期配置領域 IS 内にランダムに与える。

Step 2:[突然変異]

各探索点 $x^i(k)$ について, 解 $y^i(k)$ を

$$y^i(k) = x^{r_1}(k) + F(x^{r_2}(k) - x^{r_3}(k))$$

より生成する。ただし, r_1, r_2, r_3 ($r_1 \neq r_2 \neq r_3 \neq i$) は整数値一様分布 $\mathcal{U}_{\mathbb{N}}(1, m)$ に従う乱数である。

Step 3:[交叉]

各探索点 $x^i(k)$ と $y^i(k)$ について, 近傍解 $\hat{x}(k)$ を

$$\hat{x}_n^i(k) = \begin{cases} y_n^i(k), & \theta < CR \cup n = \lambda \\ x_n^i(k), & \text{otherwise} \end{cases}$$

より要素ごとに生成する。ただし, θ は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従う乱数, λ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, N)$ に従う乱数である。

Step 4:[探索点の更新 (選択)]

各探索点 $x^i(k)$ について,

$$v^i(k) = \begin{cases} \hat{x}^i(k) - x^i(k), & f(\hat{x}^i(k)) < f(x^i(k)) \\ 0, & \text{otherwise} \end{cases}$$

$$x^i(k+1) = x^i(k) + v^i(k)$$

より更新する。

Step 5:[終了判定]

$k = k_{\max}$ ならば, 探索を終了する。さもなければ, $k := k + 1$ とし, **Step 2** へ戻る。

3.4.3 Differential Evolution の探索構造の解析

更新式に従って生成される近傍について重点的に考察する。図 3.4 に、DE の近傍生成について示す。DE の近傍生成は、式 (3.11) の突然変異と式 (3.12) の交叉によって行われる。突然変異では、探索点 x^{r_1} を基準として、探索点間の差分ベクトル $u^{between}$ によって、解 y^i を生成する。探索点間の差分ベクトル $u^{between}$ を用いることで、探索過程の探索点分布のスケールや探索点の配置状態に適した位置に解を生成することが可能となる。交叉では、探索点 x^i と解 y^i を要素毎に組み合わせることで、近傍解 \hat{x}^i を生成する。各次元で組み合わせるときに採用する要素は乱数によって決定されるため、近傍解 \hat{x}^i はこの乱数による摂動に基づく近傍内に生成される。

DE は改善移動であるため、探索点 x^i は目的関数値が改善した場合、近傍解 \hat{x}^i へ移動する。このため、探索点 x^i は式 (2.12) の p-best、式 (2.14) の c-best は式 (2.13) の g-best とみなすことができる。つまり、DE では各探索点 x^i が降下条件を満たす場合のみ移動を行い、優れた探索点を次の探索点群とする。その後、突然変異では優れた探索点群から解 y^i を生成するため、解 y^i が優れている可能性が高い。さらに、交叉では優れた探索点 x^i と解 y^i を組み合わせることで、近傍解 \hat{x}^i はさらに優れた領域に生成されることが期待できる。これは、「改善移動を行う探索点群によって問題構造を把握し、その情報を活用した探索を行う」という POP に基づく探索構造であることを示している。したがって、DE は、降下条件を満たす探索点 x^i の移動、探索点間の差分ベクトルの活用、解同士の組み合わせによって、優れた近傍を生成することで、探索点群として良い領域に集まる。

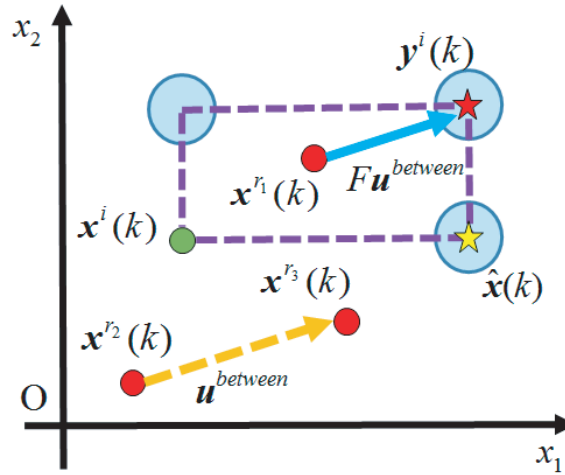


図 3.4 : Differential Evolution の近傍の生成

3.5 Artificial Bee Colony Algorithm

3.5.1 Artificial Bee Colony Algorithm の概要

Artificial Bee Colony Algorithm (ABC) [27] [31] は, 2005 年に Dervis Karaboga により開発された, 連続値最適化問題を対象とする多点探索型メタヒューリスティクスである。ABC は蜜蜂の採餌行動のメカニズムに基づいており, 働き蜂, 傍観蜂, 斥候蜂により機能を分担した探索を行う。ABC の探索性能はメタヒューリスティクスの中でも比較的高いことが示されており, 更なる改良が目指されている [43] [53]。

3.5.2 Artificial Bee Colony Algorithm のアルゴリズム

ABC では, 働き蜂, 傍観蜂, 斥候蜂がそれぞれ異なる操作を行う。近傍の生成と探索点の更新は, 主に働き蜂と傍観蜂が行う。働き蜂の探索では, 全ての探索点が探索を行う。傍観蜂の探索では, ルーレット選択で選ばれた探索点が探索を行う。斥候蜂の探索では, 停滞回数が規定回数 $Limit \geq 0$ ($m_o, Limit \in \mathbb{Z}$) よりも上回った場合のみ, ランダムに更新

する。

ABC の更新式を式 (3.15) , 式 (3.16) , 式 (3.17) に示す。

$$\hat{x}_n^i = \begin{cases} x_\lambda^i + \phi^i u_\lambda^{random}, & n = \lambda \\ x_n^i, & \text{otherwise} \end{cases} \quad (3.15)$$

$$v^i = \begin{cases} \hat{x}^i - x^i, & f(\hat{x}^i) < f(x^i) \\ 0, & \text{otherwise} \end{cases} \quad (3.16)$$

$$x^i := x^i + v^i \quad (3.17)$$

ただし, $m_o \in [0, m]$ はパラメータ, $u^{random} = x^r - x^i$, $r \neq i$ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数, ϕ^i は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(-1, 1)$ に従う乱数, λ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, N)$ に従う乱数である。

ABC では, 近傍解 \hat{x} を, 探索点 x^r を基準として, 探索点間の差分ベクトル u^{random} で生成する。移動ベクトル v を, 目的関数値により, 近傍解 \hat{x} と探索点 x の差分ベクトルか, 零ベクトル 0 のどちらかを選択する。

以下にABCのアルゴリズムを示す。アルゴリズムの終了条件を評価回数 $T \geq T_{\max}$ (T_{\max} : 最大評価回数) とする。探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。

【Artificial Bee Colony Algorithm のアルゴリズム】

Step 0:[準備]

探索点数 (働き蜂数) m , 傍観蜂数 $m_o \in [0, m] \subset \mathbb{Z}$, 規定回数 $Limit \geq 0$ ($Limit \in \mathbb{Z}$), 最大評価回数 T_{\max} を定め, 評価回数を $T = 0$ とする。

Step 1:[初期化]

探索点の初期位置 x^i ($i = 1, \dots, m$) を初期配置領域 IS 内にランダムに与え, 評価回数 $T = m$ とする。停滞回数を $s_i = 0$ とする。

Step 2:[働き蜂による探索]

各探索点 x^i ($i = 1, \dots, m$) について, 近傍解 \hat{x}^i を生成し, 更新する。

Step 2-1:[近傍の生成]

各探索点 x^i について, 近傍解 \hat{x}^i を

$$\hat{x}_n^i = \begin{cases} x_\lambda^i + \phi^i(x_\lambda^r - x_\lambda^i), & n = \lambda \\ x_n^i, & \text{otherwise} \end{cases}$$

より要素ごとに生成する。ただし, $r \neq i$ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数, ϕ^i は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(-1, 1)$ に従う乱数, λ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, N)$ に従う乱数である。

Step 2-2:[探索点の更新]

各探索点 x^i , 停滞回数 s^i について

$$v^i = \begin{cases} \hat{x}^i - x^i, & f(\hat{x}^i) < f(x^i) \\ 0, & \text{otherwise} \end{cases}$$

$$s^i := \begin{cases} 0, & f(\hat{x}^i) < f(x^i) \\ s^i + 1, & \text{otherwise} \end{cases}$$

$$x^i := x^i + v^i$$

より更新する。評価回数 $T := T + m$ とする。

Step 3:[傍観蜂による探索]

最良探索点 x^{c-best} , 各探索点 x^i の適合度 fit^i , 適合度に基づく選択確率 p^i を

$$x^{c-best} = \underset{x^i}{\operatorname{argmin}} \{f(x^i) \mid i = 1, \dots, m\}$$

$$fit^i = \frac{1}{|f(x^{c-best}) - f(x^i)| + 1}$$

$$p^i = fit^i / \sum_{i=1}^m fit^i$$

より求める。 $j = 1$ とする。

Step 3-1:[近傍の生成]

選択確率 p^i ($i = 1, \dots, m$) に基づくルーレット選択により, 移動する探索点 x^ℓ を選択する。

探索点 x^ℓ について, 近傍解 \hat{x}^ℓ を

$$\hat{x}_n^\ell = \begin{cases} x_\lambda^\ell + \phi^\ell(x_\lambda^r - x_\lambda^\ell), & n = \lambda \\ x_n^\ell, & \text{otherwise} \end{cases}$$

より要素ごとに生成する。ただし, $r \neq \ell$ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数, ϕ^ℓ は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(-1, 1)$ に従う乱数, λ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, N)$ に従う乱数である。

Step 3-2:[探索点の更新]

探索点 x^ℓ , 停滞回数 s^ℓ について

$$v^\ell = \begin{cases} \hat{x}^\ell - x^\ell, & f(\hat{x}^\ell) < f(x^\ell) \\ 0, & \text{otherwise} \end{cases}$$

$$s^\ell := \begin{cases} 0, & f(\hat{x}^\ell) < f(x^\ell) \\ s^\ell + 1, & \text{otherwise} \end{cases}$$

$$x^\ell := x^\ell + v^\ell$$

より更新する。評価回数 $T := T + 1$ とする。

$j < m_o$ ならば, $j := j + 1$ とし, Step 3-1 へ戻る。

Step 4:[斥候蜂による探索]

各探索点について, 停滞回数が規定回数を超えた探索点番号の集合を

$$P = \{i \mid s^i \geq Limit; i = 1, \dots, m\}$$

とする。 $M = |P|$ とし, 探索点 x^h ($\forall h \in P$) を初期配置領域 IS 内にランダムに更新し, 評価回数 $T := T + M$ とする。

Step 5:[終了判定]

$T \geq T_{\max}$ ならば, 探索を終了する。さもなければ, Step 2 へ戻る。

3.5.3 Artificial Bee Colony Algorithm の探索構造の解析

更新式に従って生成される近傍について重点的に考察する。図 3.5 に、ABC の近傍生成について示す。式 (3.15) で表されるように、探索点 x^i からランダムな探索点 x^r に向かう差分ベクトル u^{random} を活用して、近傍解 \hat{x}^i は乱数による摂動に基づく近傍内に生成される。このとき、近傍解 \hat{x}^i は、ランダムに選ばれた要素のみに摂動を加えて、それ以外の要素を x^i の要素とする。このため、解空間において、ある単一の軸に沿って、摂動が加えられる。これは、DE の交叉と同様に、摂動が加えられた要素と探索点の要素の組み合わせで、近傍解 \hat{x}^i が生成される。また、探索点間の差分ベクトル u^{random} を用いることで、探索過程の探索点分布のスケールや探索点の配置状態に適した位置に解を生成することが可能となる。

ABC は改善移動であるため、探索点 x^i は目的関数値が改善した場合、近傍解 \hat{x}^i へ移動する。このため、探索点 x^i は式 (2.12) の p-best、式 (2.14) の c-best は式 (2.13) の g-best とみなすことができる。つまり、ABC では各探索点 x^i が降下条件を満たす場合のみ移動を行い、優れた探索点を次の探索点群とする。その後の近傍の生成では、近傍解 \hat{x}^i は、優れた探索点群によって、単一の軸に沿った摂動が加えられるため、優れた領域に生成されることが期待できる。これは、「改善移動を行う探索点群によって問題構造を把握し、その情報を活用した探索を行う」という POP に基づく探索構造であることを示している。したがって、ABC は、降下条件を満たす探索点 x^i の移動、探索点間の差分ベクトルの活用、解同士の組み合わせによって、優れた近傍を生成することで、探索点群として良い領域に集まる。

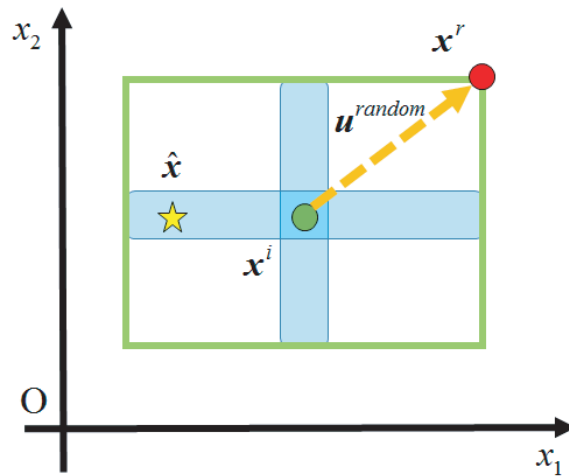


図 3.5 : Artificial Bee Colony Algorithm の近傍の生成

3.6 Cuckoo Search

3.6.1 Cuckoo Search の概要

Cuckoo Search (CS) [27][33] は, 2009 年に Xin-She Yang と Suash Deb により開発された, 連続値最適化問題を対象とする多点探索型メタヒューリスティクスである。CS は鳥のカッコウの托卵行動から着想を得ており, 以下のルールを「Lévy Flight」, 「更新」, 「排斥」として表現した操作により探索を行う。

- 各カッコウはランダムに選んだ巣に, 一つ卵を産み付ける。
- 質の良い卵を持つ巣は次の世代へと持ち越される。
- 質の悪い卵は仮親に発見されると, 巣から弾き飛ばされたり, 仮親が新たな巣を作ることで見捨てられる。

3.6.2 Lévy Flight

Lévy Flight は、ランダムウォークの一種である。Lévy Flight では、式 (1.15) の Lévy 分布に従う乱数によりステップサイズが決定する。正規分布は、分布の裾が指数関数的に減衰する対称分布であることに対して、Lévy 分布は、分布の裾が正規分布よりも緩やかに減衰するヘヴィーテイル型の非対称分布である。ヘヴィーテイル型の分布に従う乱数は、小さな値を中心にとりながらも、まれに大きな値をとることが特徴である。また、Lévy 分布は、生物の飛行パターンや採餌行動など、様々な自然現象や物理現象における確率的変動を表現できるとされている[33][45]。さらに最適化において、景観が未知で広い領域の探索を行う場合、Lévy Flight を用いることで、式 (1.14) の正規分布に従う乱数によるランダムウォーク (Gauss Flight) を用いる場合に比べ、効率的な探索を行うことができるとされている[27][45]。

Lévy 分布に従う乱数の発生方法はいくつか提案されているが、本論文では CS で代表的に使用されている、Mantegna's Algorithm (MA)[27][45][50] を用いる。Lévy 分布は確率変数が $[0, \infty]$ で定義される非対称な分布だが、MA では $x = 0$ に関して対称な分布 Le を擬似的に生成し、それに従う乱数 $L(\beta) \in [-\infty, \infty]$ を発生させる。本論文ではこの分布を近似 Lévy 分布 Le と表現する。近似 Lévy 分布 Le の形状は分布調整変数 $\beta \in [0.3, 1.99]$ の値により変化し、 $0.3 \leq \beta \leq 1.99$ であるとき、確率変数が $|x| \gg 0.1$ の範囲で Lévy 分布を近似する。図 3.6 に $\beta = 1.5, 1.0, 0.5$ のときの近似 Lévy 分布を示す。乱数 $L(\beta)$ は式 (3.18) で計算される。

$$L(\beta) = \frac{u}{|v|^{1/\beta}} \quad (3.18)$$

ここで、 u は式 (1.14) の正規分布 $N(0, \sigma_u^2)$ に従う乱数として、式 (3.19) で表される。ただし、平均 0、分散 σ_u^2 である。 v は標準正規分布に従う乱数として、式 (3.20) で表される。なお、分散 σ_u^2 は式 (3.21) であり、 Γ はオイラーのガンマ関数である。

$$u \sim N(0, \sigma_u^2) \quad (3.19)$$

$$v \sim N(0, 1) \quad (3.20)$$

$$\sigma_u^2 = \left[\frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma[(1 + \beta)/2] 2^{(\beta-1)/2} \beta} \right]^{1/\beta} \quad (3.21)$$

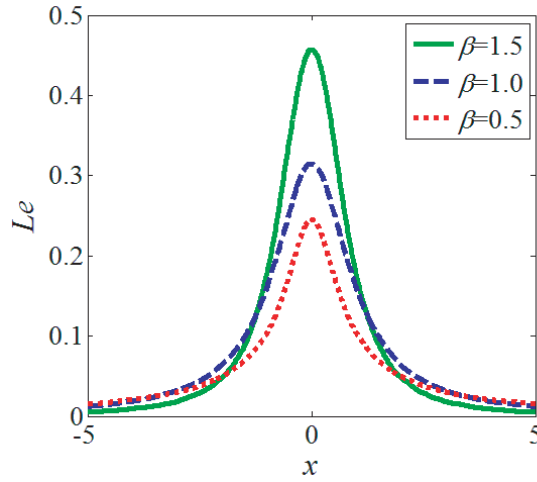


図 3.6： 近似 Lévy 分布

3.6.3 Cuckoo Search のアルゴリズム

CS では, Lévy Flight で近傍の生成, 更新と排斥で探索点の更新が行われる。Lévy Flight では, ランダムに選ばれた探索点 (参照点) を参照して, 近傍解を生成する。更新では, ランダムに選ばれた探索点 (更新点) が, 目的関数値が改善する場合, 近傍解へ更新する。排斥では, 排斥確率 $P_a \in [0, 1]$ を満たす場合, 式 (2.14) の最悪探索点 c-worst が絶対移動として更新する。排斥の更新式はいくつか存在するが, 本論文では Lévy Flight と同様の式を用いる。

CS の更新式を式 (3.22), 式 (3.23), 式 (3.24) に示す。

$$\hat{x} = x^{r_1} + s \quad (3.22)$$

$$v^{r_2} = \begin{cases} \hat{x} - x^{r_2}, & f(\hat{x}) < f(x^{r_2}) \\ 0, & \text{otherwise} \end{cases} \quad (3.23)$$

$$x^{r_2} := x^{r_2} + v^{r_2} \quad (3.24)$$

ただし, $\alpha \geq 0$, $\beta \in [0.3, 1.99]$ はパラメータである。 $s = \alpha L(\beta)$ はステップサイズ, r_ℓ ($\ell \in 1, 2$) は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数, $L(\beta) \in \mathbb{R}^N$ は Lévy 乱数ベクトルであり, 各要素 $L_n(\beta)$ は近似 Lévy 分布に従う乱数である。

CS では、近傍解 \hat{x} を、探索点 x^{r_1} を基準として、Lévy 乱数ベクトル $L(\beta)$ で生成する (Lévy Flight)。移動ベクトル v^{r_2} として、目的関数値によって、近傍解 \hat{x} と探索点 x^{r_2} の差分ベクトルか、零ベクトル 0 のどちらかを選択する。

以下に CS のアルゴリズムを示す。アルゴリズムの終了条件を $T \geq T_{\max}$ とする。また、探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。

【Cuckoo Search のアルゴリズム】

Step 0:[準備]

探索点数 m , パラメータ $\alpha \geq 0$, $\beta \in [0.3, 1.99]$, 排斥確率 $P_a \in [0, 1]$, 最大評価回数 T_{\max} を定め、評価回数を $T = 0$ とする。

Step 1:[初期化]

探索点の初期位置 x^i ($i = 1, 2, \dots, m$) を初期配置領域 IS 内にランダムに与え、評価回数を $T = m$ とする。

Step 2:[Lévy Flight]

参照探索点 x^{r_1} を基準として、近傍解 \hat{x} を

$$\hat{x} = x^{r_1} + \alpha L(\beta)$$

より、生成する。ただし、 r_1 は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数、 $L(\beta) \in \mathbb{R}^N$ は Lévy 乱数ベクトルであり、各要素 $L_n(\beta)$ は近似 Lévy 分布に従う乱数である。

Step 3:[探索点の更新]

更新探索点 x^{r_2} を

$$v^{r_2} = \begin{cases} \hat{x} - x^{r_2}, & f(\hat{x}) < f(x^{r_2}) \\ 0, & \text{otherwise} \end{cases}$$

$$x^{r_2} := x^{r_2} + v^{r_2}$$

より、更新する。ただし、 r_2 は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数である。評価回数 $T := T + 1$ とする。

Step 4:[排斥]

排斥確率 P_a に従い、以下の操作を行う。

最悪探索点 $x^{c-worst}$ を

$$x^{c-worst} = \underset{x^i}{\operatorname{argmax}} \{f(x^i) \mid i = 1, \dots, m\}$$

とし,

$$x^{c-worst} := x^{c-worst} + \alpha L(\beta)$$

より, 更新する。ただし, $L(\beta) \in \mathbb{R}^N$ は Lévy 乱数ベクトルであり, 各要素 $L_n(\beta)$ は近似 Lévy 分布に従う乱数である。 $T := T + 1$ とする。

Step 5:[終了判定]

$T \geq T_{\max}$ ならば, 探索を終了する。さもなければ, Step 2 へ戻る。

3.6.4 Cuckoo Search の探索構造の解析

更新式に従って生成される近傍について重点的に考察する。図 3.7 に, CS の近傍生成について示す。式 (3.18) で表されるように, 近傍解 \hat{x} は, ランダムな探索点 x^{r_1} を参照し, Lévy 乱数による摂動に基づく近傍内に生成される。この摂動は, ランダムウォークである Lévy Flight によって発生するため, 大きさも方向も乱数によって定まる。さらに, Lévy 分布はヘヴィーテイル型であるため, 近傍解 \hat{x} は解空間のどの範囲でも生成される可能性がある。しかし, その生成される確率は Lévy 分布に従うため, 探索点 x^{r_1} からの距離によってその確率は異なる。近傍解 \hat{x} は探索点 x^{r_1} から近い範囲では生成されやすく, 探索点 x^{r_1} から遠い範囲では生成されにくい。

CS は改善移動であるため, ランダムな探索点 x^{r_2} は目的関数値が改善した場合, 近傍解 \hat{x} へ移動する。このため, 探索点 x^{r_2} は式 (2.12) の p-best, 式 (2.14) の c-best は式 (2.13) の g-best とみなすことができる。つまり, CS では探索点 x^{r_2} が降下条件を満たす場合のみ移動を行い, 優れた探索点を次の探索点群とする。その後の近傍の生成では, 近傍解 \hat{x} は, 優れた探索点 x^{r_1} を参照して, 主に小さな摂動が発生するため, 優れた領域 (優れた探索点付近) に生成されることが期待できる。これは, 「改善移動を行う探索点群によって問題構造を把握し, その情報を活用した探索を行う」という POP に基づく探索構造であること

を示している。したがって、CS は、降下条件を満たす探索点 x の移動、小さな摂動によって、優れた近傍を生成することで、探索点群として良い領域に集まる。

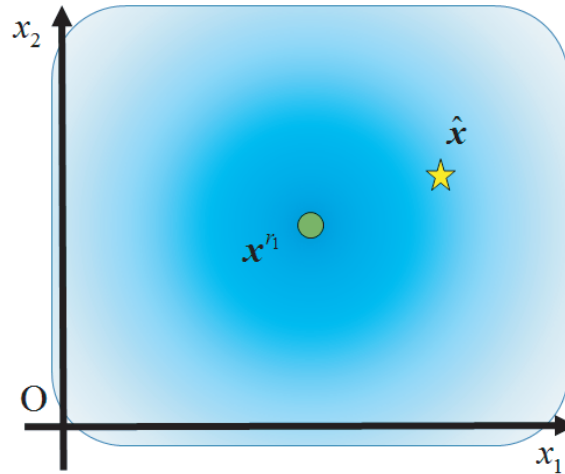


図 3.7 : Cuckoo Search の近傍の生成

3.7 メタヒューリスティクスの探索戦略

3.6 節まで、様々なメタヒューリスティクスの探索構造の解析を行った。解析の結果、3.1 節で述べたように、メタヒューリスティクスは、POP を巧みに活用する探索構造として構成されており、解空間の偏り構造を把握・活用することで、解直接探索法として効率的な探索を行っているといえよう。この解析を基に、メタヒューリスティクスの探索戦略に影響を与える操作を定義・分類する。

POP を効果的に活用するメタヒューリスティクスの探索指針として、多様化 (Diversification)・集中化 (Intensification) が知られている[22][23][24][25][26][27][35][39][54][55][56][57][58][59]。多様化・集中化は抽象的な概念であるため、様々な解釈が存在している。このため、各個人で多様化・集中化に対する解釈が異なる場合が多い。また、3.6 節まで、探索構造について解析したように、多様化・集中化の実現方法は、各手法によっても異なる。このように、多様化・集中化に対する明確な定義が存在しないため、メタヒュー

リスティクスの設計・開発において、統一的な視点で多様化・集中化を議論することは困難である。したがって、メタヒューリスティクスに対する汎用性が一定以上確保される範囲で、多様化・集中化の概念を具体化する必要がある。抽象的な概念・状態を実現するための操作を具体的に分類しておくことで、「多様化・集中化に対する解釈」と「多様化・集中化を生み出す操作」の乖離を埋めることができる。これによって、統一的な視点で各手法に対する多様化・集中化の解析が期待できる。さらに、この統一的な視点に基づいて、メタヒューリスティクスの汎用的な設計論を構築することが期待できる。

本論文では、多様化・集中化を表3.1のように定義する。多様化・集中化の目的をそれぞれ長期的な解の改善、短期的な解の改善とする。この定義では、多様化・集中化の実現状態を「探索範囲・探索点分布の広さ」に対応させる。多様化により、探索範囲・探索点分布が広がり、集中化により、探索範囲・探索点分布が狭まる。さらに、多くのメタヒューリスティクスは、探索構造の中で、① 特定の領域・方向への指向性と、② 摂動・探索点分布の広さ、を調整する操作を行うことで、多様化・集中化の実現状態が変化する。つまり、多様化を「特定の領域・方向への指向の抑制」や「摂動・分布の拡大」を行う操作とすることに対して、集中化を「特定の領域・方向への指向の促進」や「摂動・分布の縮小」を行う操作とする。特に、探索構造において、「特定の領域」として良い解を設定することや、探索点群が良い解付近に分布することで、集中化を実現させる操作は、POPを効果的に活用している。

また、多くのメタヒューリスティクスでは、限られた時間内で効率的な探索を行うために、「探索序盤では多様化、探索終盤では集中化」という探索戦略の実現を目指している[22][23][24][35][39][54][55][56][57][58][59][60]。図3.8に、2次元空間の 2^N minima Functionにおける探索戦略のイメージを示す。探索序盤では、探索範囲・探索点分布を広くしておくことで、解空間の偏り構造を把握している。探索が進むにつれて、把握した偏り構造を活用し、探索範囲・探索点分布を有望な領域に徐々に狭めていく。探索終盤では、探索範囲・探索点分布を有望な領域に狭めることで、より優れた解を探索する。このようにメタヒューリスティクスでは、探索過程で多様化・集中化を適切に実現する探索戦略が重要となる。

表 3.1：多様化・集中化の定義

探索状態	目的	効果	操作	
	解の改善	探索範囲・探索点分布	① 特定の領域・方向に対する指向性	② 摂動・分布
多様化	長期的	広域化	抑制	拡大
集中化	短期的	狭域化	促進	縮小

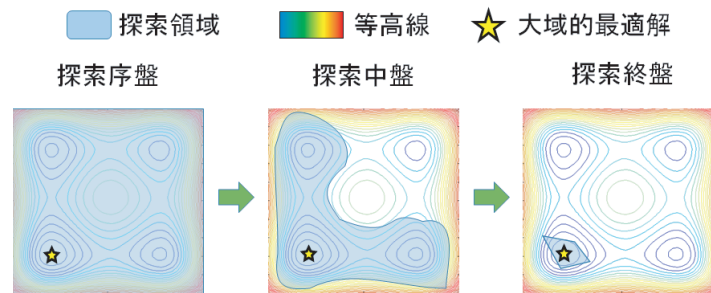


図 3.8：メタヒューリスティクスの探索戦略のイメージ

3.8 メタヒューリスティクスの探索構造の分類

3.8.1 近傍の生成方法

3.6節まで行った解析を基に，メタヒューリスティクスの探索構造，特に近傍の生成のパターンを分類し，考察する。

まず，近傍の生成方法について着目する。近傍は，探索点 x を基準とした，更新式に含まれる乱数に基づく摂動によって生成される。この摂動は，移動ベクトル v の終点となる近傍解 \hat{x} が生成される範囲となる。すなわち，近傍は更新される場合の移動ベクトル $v = \hat{x} - x$ の生成方法に依存する。 $v = \hat{x} - x$ の生成方法は，(a) 基底ベクトルの線形結合で生成する場合 (PSO, FA)，(b) 探索点同士の組み合わせで生成する場合 (DE, ABC)，(c) 特定の確率分布に従うランダムウォークで生成する場合 (CS)，の三つに分かれる。

(a) 基底ベクトルの線形結合

(a) の場合，図 3.2，図 3.3 で示されるように，近傍解 \hat{x} は，基底ベクトルの線形結合により張られる空間内に生成される。近傍は，基底ベクトルの係数や要素として使用する，乱数に基づく摂動によって得る空間となる。このため，近傍の偏りは，基底ベクトルの方向や摂動のスケールによって調整される。

(b) 解同士の組み合わせ

(b) の場合，図 3.4，図 3.5 で示されるように，近傍解 \hat{x} は，解同士の組み合わせによって生成される可能性がある空間内に生成される。近傍は，「解の中で組み合わせる要素の選択」や「要素の組み合わせ」において使用する，乱数に基づく摂動によって得る空間となる。このため，近傍の偏りは，解同士の要素の組み合わせ方，摂動のスケール，組み合わせる解同士の距離によって調整される。

(c) 特定の確率分布に従うランダムウォーク

(c) の場合，図 3.6 で示されるように，近傍解 \hat{x} は，参照点を始点とする乱数ベクトル s によって生成される。近傍は，乱数ベクトル s の要素として使用する，乱数に基づく摂動によって得る空間となる。このため，近傍の偏りは，参照点の選び方や，摂動のスケール（確率分布の種類や形状）によって調整される。(a) や (b) の場合に比べて，摂動によって得る近傍が広い。

3.8.2 ベクトルの種類

$v = \hat{x} - x$ の生成において，活用されるベクトルは，(a) 差分ベクトル u (PSO, FA, DE, ABC) と，(b) 乱数ベクトル s (FA, CS), の二つに分かれる。

(a) 差分ベクトル u

(a) の場合，差分ベクトル u が活用される箇所は，3.8.1 項の (a) (PSO, FA), DE, ABC で分けられる。3.8.1 項の (a) では，基底ベクトルとして差分ベクトル u を使用する。DE

では、解同士の組み合わせで用いる解を生成するときに差分ベクトル u を使用する。ABC では、解同士の組み合わせが行われる要素における摂動として差分ベクトル u を使用する。

差分ベクトル u の使用は、3.7 節で定義した多様化・集中化の実現状態へ影響を与えることが考えられる。差分ベクトル u を使用する場合、その終点として特殊な解・探索点を設定する。PSO では、探索点から式 (2.12) で定義される p-best までの差分ベクトル u^{p-best} と、探索点から式 (2.13) で定義される g-best までの差分ベクトル u^{g-best} を使用する。FA では、探索点から式 (2.17) で定義される better までの差分ベクトル u^{better} を使用する。DE では、ランダムに選ばれた探索点間の差分ベクトル $u^{between}$ を使用する。ABC では、探索点からランダムに選ばれた探索点までの差分ベクトル u^{random} を使用する。

差分ベクトルの活用は、基本的に探索点分布を縮小する操作となるため、探索過程で徐々に特定の領域へ近傍が狭まる。しかし、終点として設定する解の種類によって、多様化・集中化の実現状態に対する影響の強さは異なる。例えば、g-best や better などの良い解までの差分ベクトルの活用は、特定の領域に対する指向性を促進する操作となる。探索点間の差分ベクトルの活用は、探索序盤では探索点分布が広いため、特定の領域・方向に対する指向性を抑制する操作となり、探索終盤では探索点分布が狭いため、特定の領域・方向に対する指向性を促進する操作となる。したがって、差分ベクトルの活用は、多様化・集中化の実現状態へ影響を与える上に、探索過程でその状態を調整する効果を持つことから、3.7 節で述べた探索戦略の実現へ貢献することが考えられる。しかし、終点として設定する解が探索過程で同一の場合、その領域に対する指向性が促進され、短期的に探索点分布が縮小する傾向がある。このため、探索過程で適切な多様化・集中化を実現するために、差分ベクトル以外のベクトルの活用や、差分ベクトルのスケールの調整が重要である。

(b) 乱数ベクトル s

(b) の場合、乱数ベクトル s が活用される箇所は、3.8.1 項の (a) あるいは (b) (FA)、3.8.1 項の (c) (CS) で分けられる。3.8.1 項の (a) あるいは (b) では、基底ベクトルの線形結合や解同士の組み合わせにおいて、乱数ベクトル s を使用する。例として、FA では、基底ベクトルとして一様乱数ベクトル ε を使用する。3.8.1 項の (c) では、ランダムウォークにおいて、乱数ベクトル s を使用する。例として、CS では、Lévy 乱数ベクトル $L(\beta)$ を使用する。

乱数ベクトルはスケールや方向が確率的に決定されるため、ベクトルの終点を任意に設

定できず、特定の領域・方向への指向性が存在しない。乱数ベクトルの活用は、探索点分布の状態に関わらず、一定以上の広がりを持つ近傍を生成するため、摂動・分布を拡大する操作となる。一方で、乱数ベクトルを活用しても、探索過程で摂動・分布の拡大・縮小を自律的に調整することはできない。このため、乱数ベクトルの活用は、多様化・集中化の実現状態の調整に対する貢献度が低いことが考えられる。

しかし、乱数が従う分布やステップサイズのスケールは、3.7節で定義した多様化・集中化の実現状態へ影響を与えることが考えられる。CSでは、ヘヴィーテイル型のLévy分布を使用している。式(1.12)で定義される実数値一様分布 $\mathcal{U}_{\mathbb{N}}$ を使用した場合、確率分布に偏りがないため、多様化・集中化の実現状態へ影響を与えることができない。これに対して、式(1.14)で定義される正規分布 \mathcal{N} や、ヘヴィーテイル型の確率分布は、参照点から距離が近い位置の確率が高く、参照点から距離が離れた位置の確率が低い形状となるため、特定の領域に対する指向性を促進する効果を生み出す。特に、ヘヴィーテイル型の分布では、距離が離れた位置の確率がある程度残るため、探索点分布・摂動を拡大する、特定の領域に対する指向性を抑制する効果も生み出す。さらに、乱数が従う分布やステップサイズのスケールを変更することで、摂動の傾向をある程度調整することが可能である。このため、探索過程で摂動の傾向を動的に変更することで、多様化・集中化を調整する効果を生み出すことが考えられる。

3.8.3 乱数の使用箇所

数理計画法とは異なり、連続値最適化問題を対象としたメタヒューリスティクスは乱数を使用する。乱数は v の生成において使用される。乱数が使用される箇所は、(a) ベクトルの係数 (PSO, ABC), (b) ベクトルの要素 (乱数ベクトル) (FA, CS), (c) 解の組み合わせ (DE, ABC), の三つに分かれる。

(a) ベクトルの係数

(a) の場合、ベクトルの係数における乱数は多様化・集中化の実現に対して様々な効果を生み出す。ベクトルの係数が定数のみである場合、ベクトルの終点は一意に決定されるため、乱数に基づく摂動を生み出すことができない。特定の領域・方向に対する指向性を促

進する操作となる。これに対して、基底ベクトルや差分ベクトルの係数に乱数を含む場合、ベクトルの終点は一定以上の摂動の中で生成されるため、係数が定数である場合に比べて、多様な近傍を生み出すことができる。これは、摂動を拡大する操作、特定の領域・方向に対する指向性を抑制する操作となる。

また、ベクトルの係数として使用する乱数は、式(1.12)の実数値一様分布 $\mathcal{U}_{\mathbb{R}}$ 、式(1.14)の正規分布 N 、式(1.15)の Lévy 分布 \mathcal{L} などの確率分布に従う。一様乱数の区間のサイズが1以下の場合 (PSO や FA) や、乱数の大きさが小さな値である確率が高い (CS) 場合、探索点分布よりも小さなスケールで摂動が起こる可能性が高い。これは、探索点分布を縮小する操作となる。これに対して、一様乱数の区間のサイズが1以上であり、負の値もとる場合 (ABC) は、探索点分布よりも小さなスケールで摂動が起こる可能性もあれば、大きなスケールで摂動が起こる可能性もある。これは、摂動・探索点分布を拡大・縮小する操作となる。

さらに、ベクトルの係数に含まれる乱数の影響は探索過程で変化する。探索序盤では、使用するベクトルのノルムが大きいいため、摂動が大きい。探索過程で、徐々にベクトルのノルムが小さくなるため、乱数の影響が軽減された結果、探索終盤では特定の領域に探索点分布が縮小する。したがって、ベクトルの係数における乱数の活用は、多様化・集中化の実現状態へ影響を与える上に、探索過程でその状態を調整する効果を持つことから、3.7節で述べた探索戦略の実現へ貢献することが考えられる。

(b) ベクトルの要素 (乱数ベクトル)

(b) の場合、参照点から直接摂動を与え、参照点を中心とした近傍を生成する。ベクトルの要素として乱数を使用した場合、探索点分布の状態に関わらず、一定以上の広がりを持つ近傍を生成するため、摂動・分布を拡大させる操作となる。

しかし、3.8.2 項の (b) で述べたように、近傍解 \hat{x} への方向を設定することができないため、特定の領域・方向に対する指向性が存在しない。また、乱数の種類により、近傍内で近傍解 \hat{x} が生成される確率に偏りが生じる。このため、乱数ベクトルの活用は、多様化・集中化の実現状態の調整に対する貢献度が低いことが考えられる。

一方、乱数が従う分布やステップサイズのスケールを変更することで、近傍をある程度調整することが可能であることから、探索過程で摂動の傾向を動的に変更することで、多

様化・集中化の調整する効果を生み出すことが考えられる。

(c) 解の組み合わせ

(c) の場合、組み合わせる解の要素の選択において、乱数を使用する。特定の要素ではなく、様々な要素を組み合わせる可能性を持たせることで、摂動を与えることができる。組み合わせる要素をどちらか片方に偏らせることは、特定の領域に対する指向性を促進する操作となり、組み合わせる要素を偏らせないことは、特定の領域に対する指向性を抑制する操作となる。ABC では、解のうち単一の要素のみを組み合わせるが、他の要素は全て変わらないため、特定の領域に対する指向性を促進させている。DE では、解の全ての要素を組み合わせるため、特定の領域に対する指向性を抑制させている。

組み合わせる解の要素の選択における乱数の影響は探索過程で変化する。探索序盤では探索点分布が広いため、組み合わせた解の候補（近傍解）は広い領域で分布する。探索過程で、徐々に探索点分布が狭まるため、組み合わせた解の候補（近傍解）は狭い領域で分布する。したがって、組み合わせる解の要素の選択における乱数の活用は、多様化・集中化の実現状態へ影響を与える上に、探索過程でその状態を調整する効果を持つことから、3.7 節で述べた探索戦略の実現へ貢献することが考えられる。

3.9 メタヒューリスティクスの探索戦略と探索構造

メタヒューリスティクスで重要な点は、解直接探索法の枠組みの下で、特徴的な近傍の生成を行うことで、3.7 節で述べた探索戦略を実現可能な能力を有することである。このため、優れた探索には、探索戦略の観点で効果的な近傍の生成が必要である。また、3.8 節では、様々なメタヒューリスティクスにおける近傍の生成のパターンを分類し、それぞれの近傍が探索へ与える影響を考察した。

本章の考察を通じて抽出した、多様化・集中化の観点から優れた探索構造の必要条件を以下に示す。

- 特定の領域・方向に対する指向を生み出す操作。さらに、探索過程におけるその指

向性の調整。特に、「特定の領域」として良い解を設定することで、POPを効果的に活用する探索となる。

- 摂動・探索点分布の一定の広さを生み出す操作。さらに、探索過程におけるその広さの調整。特に、その領域を良い解に狭めることで、POPを効果的に活用する探索となる。
- 上述の指向性や広さに影響を与える乱数の使用。探索過程で過度な「特定の領域・方向に対する指向性」、「摂動・探索点分布の縮小」を緩和する役割。

以上から、メタヒューリスティクスの探索構造には、特に近傍の生成において、多様化・集中化を実現するための操作が埋め込まれていることが考えられる。

さらに、優れたメタヒューリスティクスでは、探索構造に埋め込まれた各操作が探索過程で多様化・集中化の実現状態を調整することで、探索戦略を実現させる能力を有していることが考えられる。例えば、差分ベクトルを活用することで、探索過程で多様化・集中化の調整を行うPSOやDEは、優れていることが考えられる。一方、多様化・集中化を実現させるための操作は埋め込まれているが、多様化・集中化の調整能力を有していない手法もあり、これらの探索構造は比較的優れていないことが考えられる。例えば、乱数ベクトルによるランダムウォークを使用することで近傍の生成を行うCSが挙げられる。したがって、メタヒューリスティクスの探索性能の向上・高い探索性能の実現において、メタヒューリスティクスの探索構造、特に近傍の生成が、多様化・集中化に基づく探索戦略とどのような関連性を有しているかに着目することは極めて重要である。

以上から、抽象的な多様化・集中化を実現するための操作を定義・分類しておくことで、各手法に対して、統一的な視点で探索構造と探索戦略の関連性を議論することができた。

4

適応化のフレームワーク

3章では，メタヒューリスティクスの探索構造の解析によって，探索構造と探索戦略の関連性を抽出・分類した。以上の解析に基づき，本章では，メタヒューリスティクスの適応能力の向上，および高い適応能力を有する新たなメタヒューリスティクスの開発において，適応能力の向上を実現し得る「適応化のフレームワーク」を構築する。さらに，適応化のフレームワークに基づく設計・解析を行うための「適応化設計論」を構築する。

4.1 メタヒューリスティクスの適応能力と探索構造

3.7 節で述べたように，メタヒューリスティクスの探索戦略は，「探索序盤では多様化，探索終盤では集中化の実現」である。この戦略は，解空間の偏り構造を把握し，活用することで効率的な探索の実現を狙う。この戦略における「問題構造の把握・活用」とは，「環境・状況に対する適応」を指す。したがって，本論文では，「多様化・集中化の実現状態（探索状態）が環境・状況に適応する能力」を「メタヒューリスティクスの適応能力」と定義する。メタヒューリスティクスは，探索過程で多様化・集中化を適切に実現することで，環境・状況に適応すると同時に，優れた探索性能を発揮する。これは，環境・状況に適応した多様化・集中化の調整能力（適応能力）が，探索性能へ寄与していることを示している。以上から，メタヒューリスティクスにおいて，適応能力の向上が極めて重要である。

パラメータ調整則の付加などの適応化のアプローチは，様々なメタヒューリスティクス

に対する適用や、探索性能・適応能力の向上が可能である。しかし、従来の適応化のアプローチでは、手法の「潜在的な素質としての適応能力」には注目していない。素質とは、「持って生まれた性質」や「将来優れた能力が発揮される素となる性質や能力」を指す。したがって、潜在的な素質としての適応能力とは、「優れた適応能力が発揮可能な素の能力」を意味する。上述の観点にしたがうと、「潜在的な素質としての適応能力」が探索性能に貢献するといえる。さらに、既存の手法を適応化する場合、適応能力の向上を目的とすると、「潜在的な素質としての適応能力」が高いことが適応化に適するための必要条件となることが考えられる。この考えに基づき、適応能力が高い探索構造、適応能力が低い探索構造を以下に示す。

- 適応能力が高い探索構造：探索過程で多様化・集中化を徐々に調整することが可能な構造
- 適応能力が低い探索構造：探索過程で多様化・集中化を一定の状態から調整することが不可能な構造

から、メタヒューリスティクスの「潜在的な素質としての適応能力」は、多様化・集中化に基づく探索戦略の実現と関連があることが考えられる。3章の解析では、メタヒューリスティクスの探索構造の優劣は、多様化・集中化の実現状態へ影響を与える操作との関連によって決定されることを指摘した。以上から、「潜在的な素質としての適応能力」は、多様化・集中化の実現状態へ影響を与える操作を含む探索構造と密接な関係にある。

例えば、CSのように、移動ベクトル v として乱数ベクトル s を使用する手法は、探索過程で特定の領域・方向への指向性や摂動・探索点分布を調整する操作を具備していないため、適応能力が低いと推測される。また、PSOやDEのように、移動ベクトル v の生成において、差分ベクトル u を活用する手法は、探索過程で特定の領域・方向への指向性や摂動・探索点分布を調整する操作を具備しているため、適応能力が高いと推測される。したがって、メタヒューリスティクスでは、多様化・集中化の実現状態へ影響を与える操作が、潜在的な適応能力へ貢献することが考えられる。

4.2 適応化のための汎用的フレームワークの構築

多様な環境下での使用が想定されるメタヒューリスティクスは、多様な環境下においても高い探索性能を発揮可能な適応能力が不可欠である。メタヒューリスティクスの更なる有用性の向上のために、これまでに適応化のためのアプローチはいくつか提案されている。しかし、これらのアプローチの多くは、① 統一的な視点に基づく適応能力を具備したメタヒューリスティクスの設計論が存在せず、既存の手法に対するパラメータ調整則の付加のみに留まっていること、② 対象の手法に特化した観点に基づいており、明確な多様化・集中化の観点に基づいていないこと、などの課題が存在する。したがって、メタヒューリスティクスの適応能力の向上、および高い適応能力を有する新たなメタヒューリスティクスの開発において、適応能力の向上を実現し得る「汎用的フレームワーク」が最も重要な役割を果たす。本論文では、上述のメタヒューリスティクスの適応能力に関する課題解決に資するため、「適応化のフレームワーク」の構築や、適応型メタヒューリスティクスの開発を行う。

文献[35]、[39]では、メタヒューリスティクスに対して多様化・集中化の観点に基づく適応化を行い、適応能力・探索性能の向上を確認している[35][39]。これらは、「潜在的な素質として有する探索状態の調整能力の向上」を行っている。したがって、多様化・集中化の観点に基づく適応化は、メタヒューリスティクスの適応能力・探索性能の向上において有効であることが考えられる。しかし、これらの研究成果においても、既存の手法に対するパラメータ調整則の開発・付加に限定していた。

以上を踏まえ、本章では、多様化・集中化の観点に基づく適応化のフレームワークを構築する。図4.1に、適応化のフレームワークを示す。適応化のフレームワークは、既存のアプローチとは異なり、① 既存の手法に対して適用可能であること、② 新たな手法を構築し、その手法に対して適用可能であること、から、汎用性が高い。既存の手法に対しては、以下の能力を付加する。

- 探索における探索状態（多様化・集中化）を定量的に評価すること。
- 探索状態を表す指標をフィードバックし、パラメータを調整することで、探索状態を制御すること。

また、潜在的に優れた適応能力を有する新たな手法を構築し、上述の方法でその手法を適

応化する。このように，探索状態の評価と制御に基づく探索能力を開発・付加することで，適応型メタヒューリスティクスを構築する枠組みを「適応化のフレームワーク」とする。

さらに，適応化のフレームワークでは，既存の手法に対する解析や新たな手法の構築を，3章の探索構造の解析によって得た統一的な視点・方法論に基づいて行う。この統一的な視点・方法論を「適応化設計論（Adaptation Design Methodology: ADM）」と呼ぶこととする。ADMでは，統一的な視点に基づき，潜在的な素質としての適応能力に重点を置いた，メタヒューリスティクスの解析・設計を行う。ADMの詳細は，4.3節で述べる。

上述のように，これまでに提案されてきた適応化のためのアプローチの多くは，① 既存の手法に対するパラメータ調整則の付加のみに留まっていること，② 対象の手法に特化した観点に基づいていること，などの課題が存在していた。これに対して，本論文で構築する適応化のフレームワークは，① パラメータ調整則の付加に限らず，高い適応能力・探索性能を具備した新たな適応型メタヒューリスティクスの柔軟な設計が可能となること，② 多くのメタヒューリスティクスの普遍的な探索構造・戦略に立脚するというアプローチに基づいていること，などの点で，これまでのアプローチとは大きく異なる上に，これまでのアプローチにおける課題解決に資する可能性が高い。

本論文の5章以降では，適応化のフレームワークを用いた適応型メタヒューリスティクスの開発を行う。5章では，適応化のフレームワークを既存の手法である Cuckoo Search (CS) に対して適用することで，適応型 CS を開発する。6章では，適応化のフレームワークに基づき，ADMによって潜在的な適応能力を有する新たなメタヒューリスティクスを構築する。さらに，7章では，適応化のフレームワークに従い，この手法を基礎とした適応型メタヒューリスティクスを開発する。

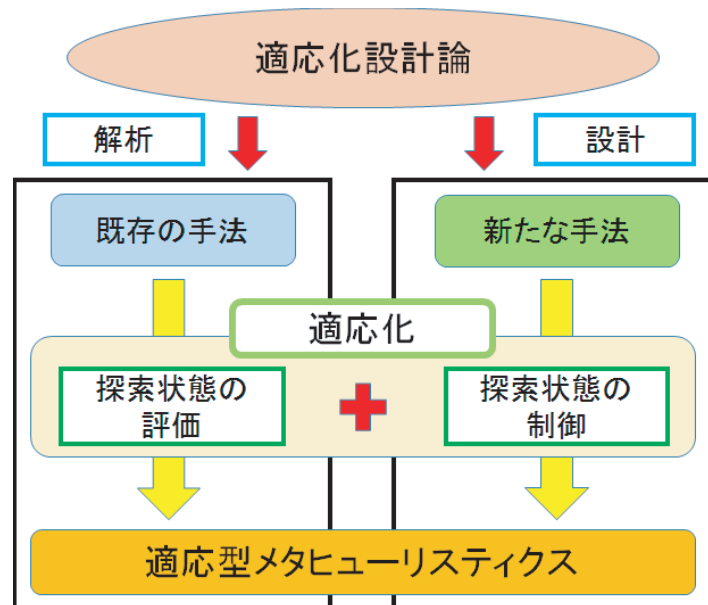


図 4.1： 適応化のフレームワーク

4.3 適応化設計論

本節では、4.3 節で述べた適応化設計論（Adaptation Design Methodology: ADM）について説明する。ADM では、統一的な視点に基づき、潜在的な素質としての適応能力に重点を置いた、メタヒューリスティクスの解析・設計を行う。この視点は、3 章の探索構造の解析結果に基づいている。3 章の探索構造の解析によって抽出した優れた探索構造の必要条件を以下に示す。

- 特定の領域・方向に対する指向を生み出す操作。さらに、探索過程におけるその指向性の調整。特に、「特定の領域」として良い解を設定することで、POP を効果的に活用する探索となる。
- 摂動・探索点分布の一定の広さを生み出す操作。さらに、探索過程におけるその広さの調整。特に、その領域を良い解に狭めることで、POP を効果的に活用する探索となる。

- 上述の指向性や広さに影響を与える乱数の使用。探索過程で過度な「特定の領域・方向に対する指向性の促進」,「摂動・探索点分布の縮小」を緩和する役割。

4.1 節で述べたように,これは潜在的な素質としての適応能力を生み出すための源となる。ADMでは,①既存の手法に対して,適応能力に対する潜在的な素質を解析すること,②潜在的な素質としての適応能力が発揮可能な手法を新たに設計すること,が可能となる。

3.9 節で述べたように,メタヒューリスティクスの探索性能の向上において,探索構造,特に近傍の生成と多様化・集中化に基づく探索戦略との関連性に注目することは極めて重要である。手法が,探索戦略を適切に実現するための適応能力を素質として具備するには,上述の優れた探索構造の必要条件を可能な限り多く満たす必要がある。図 4.2, 図 4.3, 図 4.4 に,必要条件を満たす場合の探索のイメージを示す。図 4.2 は,特定の領域に対する指向が生み出される場合の探索と,特定の方向に対する指向が生み出される場合の探索のイメージを示している。図 4.3 は,摂動の一定の広さが生み出される場合の探索と,探索点分布の一定の広さが生み出される場合の探索のイメージを示している。図 4.4 は,乱数を使用することで,探索過程で上述の指向性の抑制や摂動・探索点分布の拡大が生み出される場合の探索のイメージを示している。いずれの場合も,探索に与える影響が探索過程で調整され,探索戦略に貢献する操作が望ましい。これらは,探索戦略を実現するための具体的な操作の例として挙げられる。

さらに,3.9 節で述べたように,近傍の生成において 差分ベクトルや解の組み合わせを行う操作,またはこれらの操作における乱数の活用が探索戦略の実現に対して多大な貢献をすることが考えられる。この考えに基づくと,DE と ABC は上述の必要条件を全て満たしていることから,潜在的な素質として高い適応能力を備えた探索構造を有している。新たな手法においても,上述の必要条件を可能な限り多く満たすように設計を行うことが,適応能力に対する潜在的な素質を決定づけることになる。したがって,6 章では,以上の ADM によって潜在的な適応能力を具備するように,新たなメタヒューリスティクスを設計する。

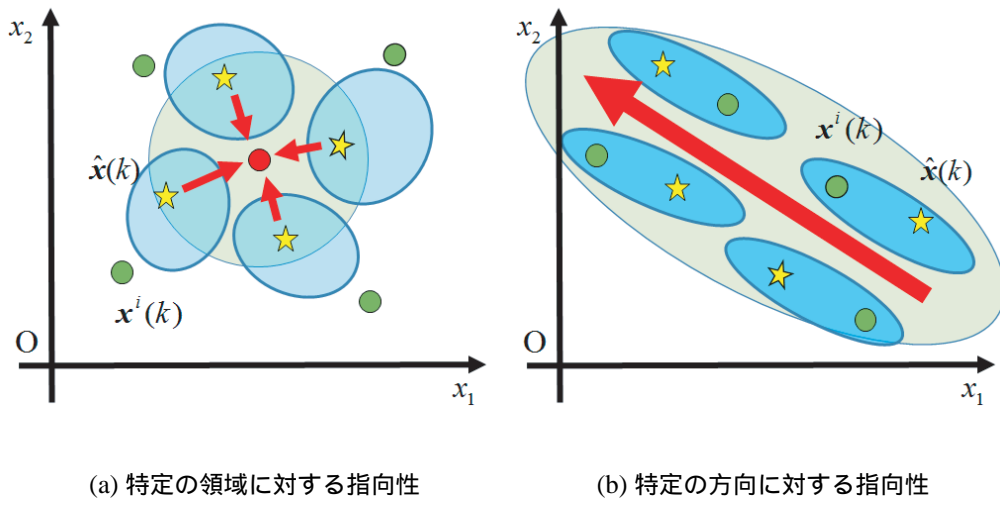


図 4.2 : 特定の領域・方向に対する指向性が生み出される探索のイメージ

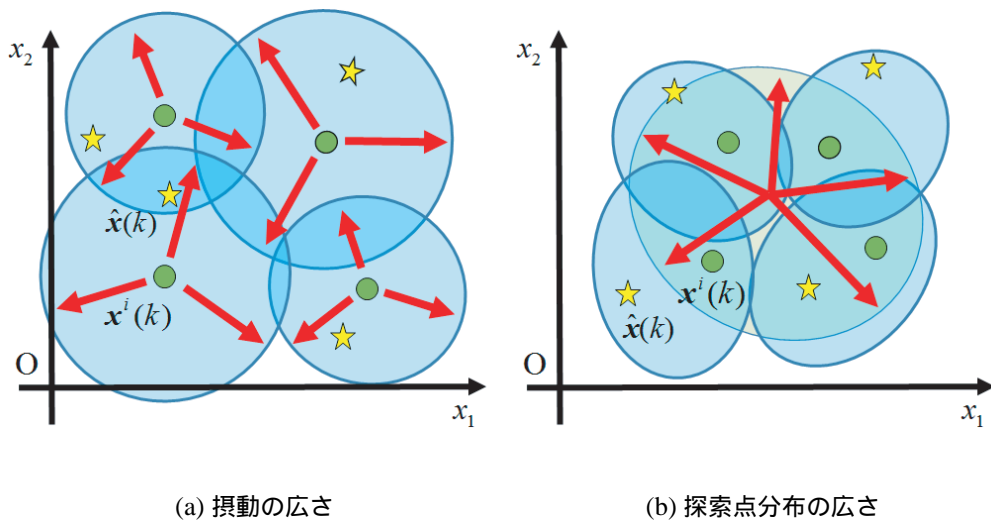


図 4.3 : 摂動・探索点分布の広さが生み出される探索のイメージ

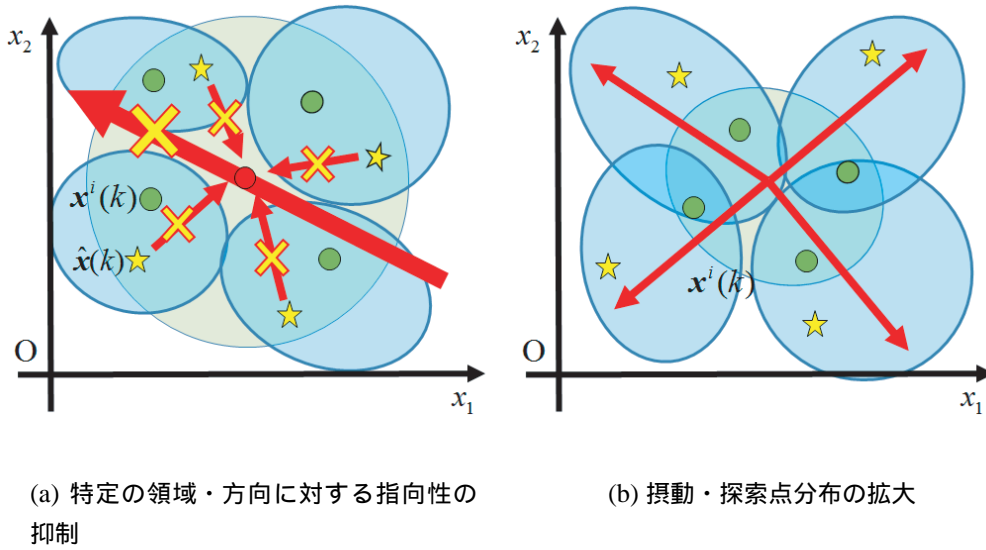


図 4.4：乱数が使用される探索のイメージ

以下に，ADMに基づくメタヒューリスティクスの設計指針を示す。図3.1で示したように，メタヒューリスティクスは，① 近傍の生成，② 探索点の更新，という普遍的な探索構造を有している。また，3章で述べたように，探索点の更新は「絶対移動」と「改善移動」の二種類の方法によって行われることに対して，近傍の生成は多様な方法が考えられる上に，どちらの更新方法においても，近傍の生成方法が探索ダイナミクスを決定づける。以上から，ADMでは，事前に定められたアルゴリズムの枠組みに基づき，近傍の生成方法と探索点の更新方法を決定することで，手法を設計する。特に，近傍の生成方法は自由度が高いが，上述したように，潜在的な適応能力に貢献する要素を活用するように近傍生成を決定することが，ADMの特徴である。

以下に，適応化のフレームワークに基づくメタヒューリスティクスの枠組みとなるアルゴリズムを示す。アルゴリズムの終了条件を $k = k_{\max}$ とする。評価回数は $T = m(k_{\max} + 1)$ となる。探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。

【適応化設計論に基づくメタヒューリスティクスのアルゴリズム】

Step 0: [準備]

探索点数 m ，パラメータ，最大反復回数 k_{\max} を定め，反復回数を $k = 1$ とする。

Step 1: [初期化]

探索点の初期位置 $x^i(k)$ ($i = 1, 2, \dots, m$) を初期配置領域 IS 内にランダムに与える。

Step 2: [近傍の生成]

各探索点 $x^i(k)$ について, 近傍の生成方法に従い, 近傍解 $\hat{x}^i(k)$ を生成する。

Step 3: [探索点の更新]

各探索点 $x^i(k)$ を, 探索点の更新方法に従い, 更新する。

Step 4: [終了判定]

$k = k_{\max}$ ならば, 探索を終了する。さもなければ, $k := k + 1$ とし, **Step 2** へ戻る。

5

適応化のフレームワーク に基づく 適応型 Cuckoo Search

本章では、適応化のフレームワークを既存の手法である Cuckoo Search (CS) に対して適用することで、適応型 CS を開発する。多様化・集中化の観点から探索ダイナミクスの解析を行うことで、CS の多様化・集中化の調整能力が低いことを指摘する。この問題を踏まえ、適応化のフレームワークに基づき、CS に多様化・集中化の調整能力を付加させることで、適応能力の高い CS を構築する。

5.1 Cuckoo Search のパラメータ調整方法

メタヒューリスティクスは調整可能なパラメータを有しており、パラメータによりその探索ダイナミクスは大きく変化する。しかし実際には、メタヒューリスティクスが探索性能を十分に発揮するために、対象問題や探索条件に応じて使用者による試行錯誤に基づく設定が必要となる。そのため、数値実験や解析を通じてパラメータの設定・調整に対する有効な知見を抽出し、パラメータの適応的な調整方法の開発が求められている。CS はパラメータとして、ステップサイズ調整変数 α 、排斥確率 P_a を有している。さらに、MA を用いてレヴィ分布に従う乱数を発生させた場合、分布調整変数 β も CS のパラメータに含まれる。以上から、使用者への負担軽減と CS の探索性能の向上には、これらのパラメータの調整方法を検討することが重要である。

これまで CS のパラメータ調整方法はいくつか報告されており、代表的な方法として以

下の4つが挙げられる。

- 文献 [46]：予め設定されたスケジュールにしたがって，探索過程で α を減少させる。
- 文献 [47]：予め設定されたスケジュールにしたがって，探索過程で α と P_a を減少させる。
- 文献 [48]：参照する探索点の評価値に応じて変化するスケジュールを作成し，作成したスケジュールにしたがって探索過程で α を減少させる。
- 文献 [49]：参照する探索点の評価値にしたがって，① 予め設定されたスケジュールによって探索過程で α を減少させるか，② 目的関数値が良いほど α を小さな値へ調整するか，を選択する。

どのパラメータ調整方法も，探索過程で α を減少させることが共通している。これは，探索序盤では大きな摂動（多様化）を促し，探索終盤では小さな摂動（集中化）を促すことで，3.7 節で述べた探索戦略の実現を目指している。さらに，文献 [48]，[49] の調整方法は，探索点群における参照点 x^r の良さに応じて α を調整するため，探索過程で得た情報に基づく適応的なパラメータ調整方法である。したがって，以上の方法は，適応能力・探索性能の向上を目的として，探索過程で α を調整する。

しかし，以上の方法では，多様化・集中化の実現状態が明確にされておらず，その状態に応じてパラメータを調整していない。さらに， α と β は CS の探索ダイナミクスへ影響を与える重要なパラメータだが， β の有効な知見・調整方法については明らかになっていない。したがって，以上のパラメータ調整方法では，① 適応能力の向上に余地があること，② 各パラメータの効果と多様化・集中化の関係を明確に意識していないこと，が課題であるといえる。

5.2 Cuckoo Search の解析

本章では，3.7 節で述べたメタヒューリスティクスの探索構造の観点から，CS の特徴を明らかにする。また，多様化・集中化の観点から探索ダイナミクスの解析を行うことで，CS の多様化・集中化の調整能力が低いことを指摘する。さらに，パラメータ解析を行い，

パラメータがCSの探索ダイナミクスへ与える影響を明らかにする。

5.2.1 Cuckoo Search の特徴

本節では、探索構造の観点からCSの特徴を明らかにする。一般的なメタヒューリスティクスの探索構造の観点から解析すると、CSは「近傍の生成」に特徴を有していると考えられる。多くの多点探索型メタヒューリスティクスでは、更新点 $x^{r_2} \in X$ が更新先（近傍解 \hat{x} ）を生成するとき、更新点 x^{r_2} の情報を参照する。一方、CSでは、近傍解 \hat{x} を生成するとき、更新点 $x^{r_2} \in X$ の情報とは限らず、探索点群のいずれかの探索点 $x^{r_1} \in X$ の情報を参照する[†]。以上のCS特有の「近傍の生成」が、カッコウの托卵行動のメカニズムを表現していると考えられる。

5.2.2 Cuckoo Search の探索ダイナミクスの解析

本節では、一般的なメタヒューリスティクスの探索戦略やCSの特徴を基に、CSの探索ダイナミクスを解析する。CSの探索ダイナミクスに大きな影響を与える操作は、Lévy Flightによる「近傍の生成」と「解の更新」である。Lévy Flightでは、探索点群の参照点 $x^{r_1} \in X$ を参照し、Lévy乱数 $L(\beta)$ による摂動を加えることで、近傍解 \hat{x} を生成する。解の更新では、探索点群から更新点 $x^{r_2} \in X$ を選び、 $f(\hat{x})$ が $f(x^{r_2})$ よりも良い場合、 $\hat{x} \leftarrow x^{r_2}$ を更新する。これらの操作から、近傍の生成において以下の傾向が考えられ、以下の傾向を考察することで、探索ダイナミクスを解析する。

- 参照点 x^{r_1} に加えられる摂動の大きさによって近傍解 \hat{x} の生成範囲が異なる。
- 参照点 x^{r_1} の目的関数値の良さによって解の更新が行われる可能性が異なる。

まず、摂動が比較的小さい場合を考える。参照点 x^{r_1} が探索点群で比較的良好の場合、POPの観点から、近傍解 \hat{x} は比較的良好な可能性が高い。また、 x^{r_1} が更新点 x^{r_2} よりも良い場

[†] CSのアルゴリズムにおける参照解の選び方は、本論文で採用している方法以外にも「探索点群の中で目的関数値が最も良い探索点を参照する」などの方法がいくつか存在するが、いずれの方法においても更新を行う探索点のみを参照して近傍解を生成することはない。

合、解の更新が行われる可能性が高い。そのため、探索過程で目的関数値が良い探索点の周辺に探索点群が集まる。特に、CS では探索序盤においてもこの現象が起きる。次に、摂動が比較的大きい場合を考える。解が更新される場合、更新点 x^{r_2} は他の探索点とは離れた位置にある優良な領域に移動する可能性が高い。他の探索点がこの x^{r_2} を参照して近傍解を生成することで、別の更新点もその離れた位置にある優良な領域に移動する。そのため、CS では探索点群が特定の狭い領域に集中していても、別の優良な領域への移動が可能である。さらに、近似 Lévy 分布の形状から、摂動は大きい場合よりも小さい場合が多い。以上の解析から、CS は良い解を積極的に活用すると同時に、持続可能な探索を行うことで、高い探索性能を発揮できると考えられる。

ところで、表 3.1 で定義したように、多様化・集中化の実現状態は、① 特定の領域・方向に対する指向性と、② 摂動・探索点分布の広さ、の調整により変化する。この定義に基づく、CS は「特定の領域（良い解）に対する指向性の促進」と「探索点分布の縮小」による集中化を積極的に実現するが、「特定の方向に対する指向性の促進・抑制」、「摂動の拡大・縮小」、「探索点分布の拡大」による多様化・集中化を実現できない。したがって、CS の探索性能の向上には、探索過程で多様化・集中化を適切に実現できるように、CS が「特定の領域・方向に対する指向性の促進・抑制」、あるいは「摂動・探索点分布の拡大・縮小」の調整能力を付加することが必要となる。

5.2.3 Cuckoo Search のパラメータ解析

α は摂動のスケールを調整し、 β は摂動の傾向を調整するパラメータであることが想定されるため、CS の多様化・集中化の能力を向上させる上で重要な役割を果たすことが考えられる。しかし、5.1 節で述べたように、 β に対する有効な知見・調整方法については明らかにされていない。以上から、本節では β に関する解析を行うことで、CS の多様化・集中化に対する影響を明らかにする。

まず、定性的解析により、 β と摂動の関係を調べる。CS では近似 Lévy 分布に従う Lévy Flight を行うことで、近傍解 \hat{x} を生成する。図 3.6 で示すように、近似レヴィ分布の形状は分布調整変数 β により決定される。 β が大きい場合、乱数 $L(\beta)$ が大きい値をとる確率は低く、小さい値をとる確率は高くなる。一方、 β が小さい場合、乱数 $L(\beta)$ が小さい値をと

る確率は低く、大きい値をとる確率は高くなる。これは、 β が小さいほど、参照点の解 x^{r_1} に加えられる摂動が大きくなることで、近傍解 \hat{x} の生成範囲が広くなり、 β が大きいほど、摂動が小さくなることで、近傍解 \hat{x} の生成範囲が狭くなることを表す。以上から、 β が変化すると、参照点 x^{r_1} を中心とする摂動が拡大・縮小するため、近傍解 \hat{x} の生成範囲の広さも変化することが明らかになった。

次に、定量的解析により、 β と摂動の関係を調べる。具体的には、 β に対応する乱数 $L(\beta)$ の標本の統計量を計算し、 β の特性を調べる。定量的解析の方法と結果については、付録1を参照されたい。付録1の定量的解析から、 β の値と Lévy 乱数 $L(\beta)$ (摂動量) の間に関係性が確認できる。これは、初期配置領域 IS のサイズ b に応じて適切な β の範囲が存在することを示している。

以上のパラメータ解析から、① β が大きい場合、摂動が狭くなり(集中化)、 β が小さい場合、摂動が広くなる(多様化)ことと、② IS のサイズ b と適切な β の範囲の間には定量的関係が存在すること、を明らかにした。したがって、解析を通じて、 β は CS の摂動の広さを決定付けるパラメータであることがいえる。

5.3 Cuckoo Search の多様化・集中化の評価指標

メタヒューリスティクスの探索性能の向上には、多様化・集中化の実現が重要だが、5.2節の解析では、CS は多様化・集中化の調整能力を有していないことを指摘した。本章では、探索過程における探索状態を把握するために、CS における多様化・集中化の評価指標を新たに定義する。さらに、数値実験を通じて、評価指標が CS の多様化・集中化の実現状態を定量的に評価可能であることを示す。

5.3.1 評価指標の提案

多様化・集中化の実現状態は、① 特定の領域・方向に対する指向性と、② 摂動・探索点分布、を調整することで変化する。また、5.2.2 項では、CS は「特定の領域・方向に対する指向性の促進・抑制」、あるいは「摂動・探索点分布の拡大・縮小」による多様化・集中

化の調整能力を有していないことを推測した。このCSの多様化・集中化に対する解釈は、手法固有の観点に基づくのではなく、一般的なメタヒューリスティクスの観点に基づいている。さらに、5.2.3項では、 β がCSの摂動の広さを決定付けるパラメータであることが明らかになった。

上記の推測に基づくと、CSに β の変化を活用し、多様化・集中化の調整能力を付加させることで、より明確な多様化・集中化の実現が期待でき、その結果、探索性能の向上が見込める。多様化・集中化の実現状態の調整には、「摂動・探索点分布」に対する評価指標が必要となる。本節では、5.2.3項のパラメータの解析に基づき、「摂動・探索点分布の拡大」を多様化、「摂動・探索点分布の縮小」を集中化とした場合の多様化・集中化の実現状態に対する評価を考える。「摂動」と「探索点分布」の評価指標を定義し、その指標を基に多様化・集中化の評価指標を定義する。

まず、摂動の評価指標について考える。3.1節や3.8節で述べたように、近傍は、 $v = \hat{x} - x^{r_1}$ の生成方法に依存する。また、本論文では、摂動は乱数によって拡大された近傍のことを指す。3.6.4項で述べたように、CSはLévy乱数ベクトル $L(\beta)$ が $\hat{x} - x^{r_1}$ となるため、CSにおける摂動は、 $\hat{x} - x^{r_1} = L(\beta)$ となる。このため、摂動の評価指標 P を式(5.1)と定義する。 P は近傍解 \hat{x} の生成範囲の広さを評価する。CSでは実質一つの探索点しか動かないため、摂動の評価指標 P を式(5.2)とする。

$$P = \sqrt{\frac{1}{Nm} \sum_{i=1}^m \sum_{n=1}^N (\hat{x}_n^i - x_n^i)^2} \quad (5.1)$$

$$P = \sqrt{\frac{1}{N} \sum_{n=1}^N s_n^2}, \quad s_n = \min \{ \alpha |L_n(\beta)|, b \} \quad (5.2)$$

ただし、初期配置領域 $IS = [a, c]^N$ 、 S のサイズ $b = |c - a|$ とする。Lévy乱数 $L(\beta)$ が大きければ、 P は大きく（生成範囲が広く）、 $L(\beta)$ が小さければ、 P は小さい（生成範囲が狭い）。

次に、探索点分布の評価指標について考える。式(1.4)より、 x^i と x^k に関するManhattan距離 $d_M(x^i, x^k)$ は式(5.3)で表される。

$$d_M(x^i, x^k) = \sum_{n=1}^N d_n(x^i, x^k), \quad d_n(x^i, x^k) = |x_n^i - x_n^k| \quad (5.3)$$

式(5.3)の Manhattan 距離の各要素 $d_n(x^i, x^k)$ を用いて, 探索点分布の評価指標 D を式(5.4)と定義する。 D は ℓ_n の平均であり, 探索点分布の広さを評価する。

$$\begin{cases} D = \frac{1}{N} \sum_{n=1}^N \ell_n, \\ \ell_n = \left(\sum_{i=1}^{m-1} \sum_{k=i+1}^m d_n(x^i, x^k) \right) / \sum_{h=1}^{m-1} h \end{cases} \quad (5.4)$$

ただし, ℓ_n は全ての探索点間における $d_n(x^i, x^k)$ の平均である。 ℓ_n が大きければ, D は大きく(点分布が広く), ℓ_n が小さければ, D は小さい(点分布が狭い)。

以上を踏まえ, 本論文では, CS の多様化・集中化の評価指標 I を式(5.5)と定義する。 I は, 摂動と探索点分布の範囲を評価する。

$$I = (P + D)/2 \quad (5.5)$$

多様化・集中化の実現状態は摂動と探索点分布の範囲によって変化するため, I が小さい場合を集中化, I が大きい場合を多様化として対応付けることで, I が CS の多様化・集中化の評価指標として期待できる。また, 評価指標 I は, CS 固有の多様化・集中化ではなく, メタヒューリスティクスの一般的な多様化・集中化に基づき, CS の多様化・集中化を評価する点で新しいといえる。

5.3.2 評価指標の数値実験的検証

本節では数値実験を通じて, 評価指標 I と探索ダイナミクスとの関係を明らかにすることで, 評価指標 I が CS の多様化・集中化を評価可能であることを示す。Rastrigin Function, Griewank Function を用いた数値実験を行い, オリジナル CS における評価指標 I を検証する。共通の実験条件として, 探索点数 $m = 10$, 次元数 $N = 50$, ステップサイズ調整変数 $\alpha = 0.1$, 排斥確率 $P_a = 0$, 関数毎に共通の初期配置領域 IS を用い, 終了条件を評価回数 $T_{\max} = 1000$ とする。 IS のサイズ b は, Rastrigin Function が $b = 10$, Griewank Function が $b = 100$ である。

図 5.1, 図 5.2 に, $\beta = 0.5, 1.0, 1.5$ に設定した場合の評価指標 I と最良解の評価値 $f(x^{g-best})$ の推移を示す。 β が大きい場合, 摂動の評価指標 P と探索点分布の評価指標 D が小さな値

で一定となるため、多様化・集中化の評価指標 I も小さくなり、 β が小さい場合、 P と D が大きな値で一定となるため、 I も大きくなることが確認できる。これは、5.2.3 項で示したように、 β が大きい場合、摂動・探索点分布が狭い状態が続き、 β が小さい場合、摂動・探索点分布が広い状態が続くことを示している。5.2.2 項で指摘したように、 I の数値実験的検証から、CS の多様化・集中化の調整能力が低いことが明らかになった。また、 $f(x^{g-best})$ が改善する場合、 I が探索序盤では大きく、探索過程で減少する傾向が確認できる。これは、CS で効率が良い探索を行う場合、「探索序盤では多様化、探索終盤では集中化」という探索戦略がある程度実現されることを示している。しかし、 S のサイズ b によって、 β が I へ与える影響は異なることが確認できる。これは、5.2.3 項で示したように、 IS のサイズ b に応じて探索点分布・摂動の適切な範囲が異なるためだと考えられる。

したがって、数値実験を通じて、① 本論文で定義した評価指標 I は、CS の多様化・集中化（探索状態）を評価可能であることと、② β の値により CS の多様化・集中化の実現状態が異なるが、その状態が探索過程で続くこと、を明らかにした。以上の検証から、探索条件や探索状態に応じて β を調整し、探索過程で多様化・集中化を適切に実現することで、CS の探索性能・適応能力の向上が期待できる。

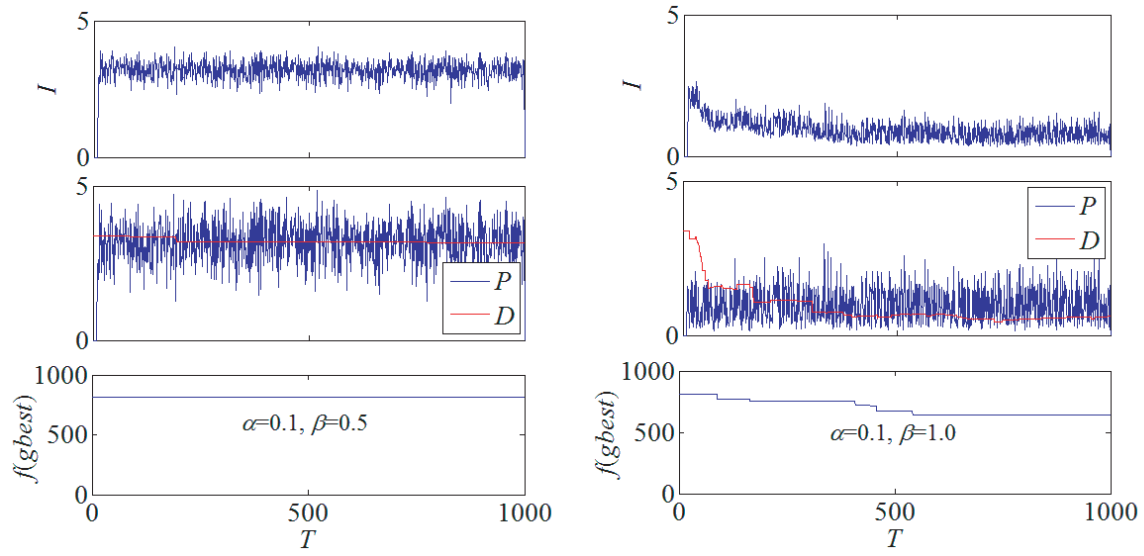
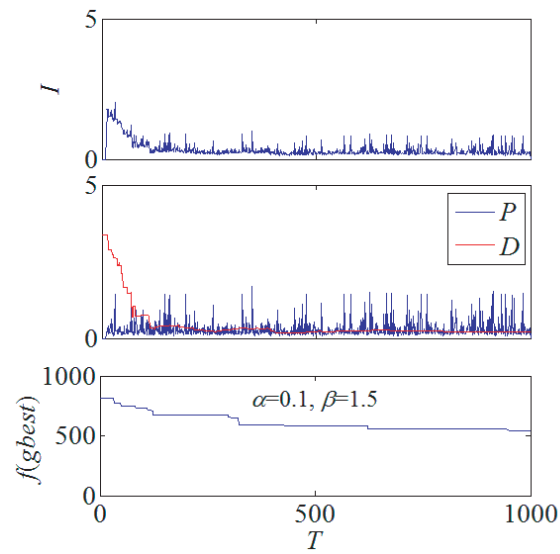
(a) $\beta = 0.5$ (Rastrigin)(b) $\beta = 1.0$ (Rastrigin)(c) $\beta = 1.5$ (Rastrigin)

図 5.1 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (Cuckoo Search , Rastrigin , $N = 50$)

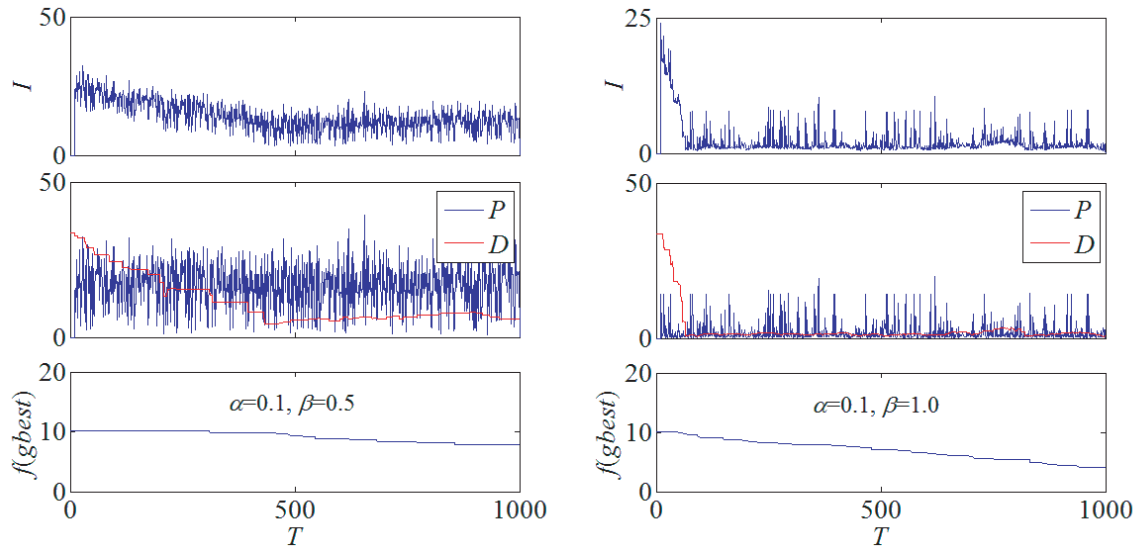
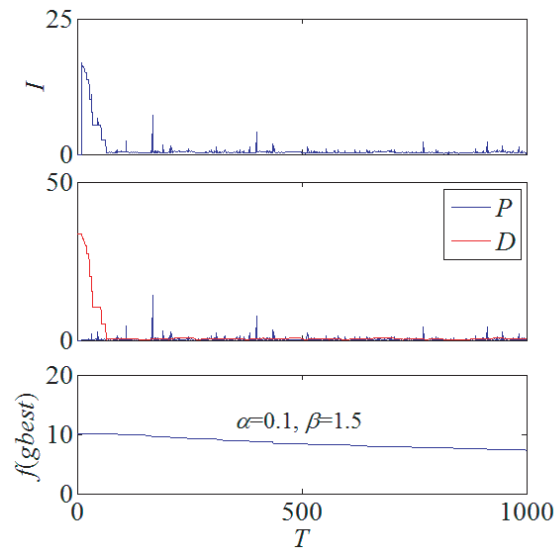
(a) $\beta = 0.5$ (Griewank)(b) $\beta = 1.0$ (Griewank)(c) $\beta = 1.5$ (Griewank)

図 5.2 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (Cuckoo Search, Griewank, $N = 50$)

5.4 探索状態の評価と制御に基づく適応型 Cuckoo Search

5.4.1 多様化・集中化の適応的制御

多様な環境・状況に対して、高い適応能力・探索性能を有するメタヒューリスティクスの開発・改良が求められる。さらに、メタヒューリスティクスの探索性能の向上において、探索過程で多様化・集中化を適切なバランスで実現することが重要である。従来のCSの適応化のアプローチでは、探索過程で α が減少することで探索状態が多様化から集中化へ推移するが、多様化・集中化が適切に実現されているとはいえない[46][47][48][49]。また、文献[35]、[39]では、PSOやDEに対して多様化・集中化の観点に基づく適応化を行い、適応能力・探索性能の向上を確認している[35][39]。これらは、従来から具備している「探索状態の調整能力の向上」を目的として、以下の能力を付加することで、多様化・集中化を適切に実現している。

- 探索における探索状態（多様化・集中化）を定量的に評価する。
- 探索状態を表す指標をフィードバックし、パラメータを調整することで、探索状態を制御する。

上述のように、これまでに提案されてきた適応化のためのアプローチの多くは、①既存の手法に対するパラメータ調整機能の付加のみに留まっていること、②対象の手法に特化した観点に基づいていること、などの課題が存在していた。

これに対して、4章で構築した適応化のフレームワークは、①パラメータ調整則の付加に限らず、高い適応能力・探索性能を具備した新たな適応型メタヒューリスティクスの柔軟な設計が可能となること、②多くのメタヒューリスティクスの普遍的な探索構造・戦略に立脚するというアプローチに基づいていること、などの点で、これまでのアプローチとは大きく異なる上に、これまでのアプローチにおける課題解決に資する可能性が高い。したがって、5.4節では、適応化のフレームワークに従い、普遍的な探索戦略の観点からCSの適応能力・探索性能を向上させる。CSは既存手法であるため、CSの適切な多様化・集中化が実現可能なパラメータ調整則を開発・付加する。以上を踏まえ、5.4.2項では、多様化・集中化の定量的評価と適応的制御を行う適応型CSの開発を行う。CSは探索状態の調整能力が低いため、本論文における適応型CSの開発は、「探索状態の調整能力の付加」を

目的としている。

5.4.2 適応型 Cuckoo Search

5.2 節と 5.3 節で述べたように, CS では, ① β の値により, 多様化・集中化の実現状態 (探索状態) は異なることと, ② 「摂動・探索点分布の拡大・縮小」を行う操作が存在しないため, 探索状態が変化しにくいこと, を確認した。したがって, 探索状態の評価と制御に基づき β を調整することで, 適応能力・探索性能の向上が期待できる。

以上を踏まえ, 本節では探索状態の評価と制御に基づく適応型 CS を提案する。そのためには, 5.4.1 項で述べたように, 探索状態の評価指標と制御式を与える必要がある。すでに, 5.2 節と 5.3 節では, CS の多様化・集中化の評価指標を提案し, 数値実験を通じて, 指標が CS の多様化・集中化を定量的に評価できることを確認した。また, CS では, 探索過程において, β が大きい場合, 摂動・探索点分布が狭い状態 (集中化) が続き, β が小さい場合, 摂動・探索点分布が広い状態 (多様化) が続くことを明らかにした。したがって, 探索過程における CS の多様化・集中化の実現状態を評価しながら, 事前に設定した目標値に評価指標が追従するように β を調整することで, CS の多様化・集中化を制御することが可能となる。

具体的な β の調整則を式 (5.6) に示す。事前に評価指標の目標値スケジュール $I_{\text{target}}(T)$ ($T = 1, 2, \dots, T_{\text{max}}$) を与えておき, 各評価回数 T における評価指標 I の値と比較し, I が $I_{\text{target}}(T)$ に追従するように調整幅 $\Delta\beta > 0$ で β を調整する。

$$\beta := \begin{cases} \min \{\beta + \Delta\beta, \beta_{\text{max}}\}, & I \geq I_{\text{target}}(T) \\ \max \{\beta - \Delta\beta, \beta_{\text{min}}\}, & I < I_{\text{target}}(T) \end{cases} \quad (5.6)$$

この調整則では, 目標値スケジュール I_{target} を適切に設定することで, 各探索段階における理想の探索状態へ制御することができる。3.7 節で述べたように, メタヒューリスティクスでは「探索序盤では多様化の実現, 探索終盤では集中化の実現を目指す」戦略が有効であることが知られている。本論文では目標値スケジュール I_{target} を探索序盤では大きな値に, 探索終盤では小さな値に設定することで, 多様化・集中化を適切なバランスで制御することが期待できる。図 5.3 に, 本論文で使用する目標値 I_{target} の線形および指数減少スケジュールを示し, 式 (5.7), 式 (5.8) にそれぞれのスケジュールの式を示す。

$$I_{\text{target}}(T) = \max \left\{ 0, I_{\text{start}} \left(1 - \frac{T}{T_{\text{end}}} \right) \right\} \quad (5.7)$$

$$I_{\text{target}}(T) = I_{\text{start}} \left(\frac{I_{\text{end}}}{I_{\text{start}}} \right)^{\frac{T}{T_{\text{max}}}} \quad (5.8)$$

なお, 提案手法における β の調整則では, 新たに下限値 β_{\min} , 上限値 β_{\max} , 調整幅 $\Delta\beta$, T_{end} , I_{start} , I_{end} というパラメータが導入されている[‡]。しかし, 許容される評価回数に応じて評価指標の目標値スケジュールを設定し, このスケジュールに追従させることによって, 最大評価回数の制約の下で多様化・集中化を確実に実現し, 多くの対象問題・探索条件に対して, 高い適応性・探索性能を維持することが期待できる。さらに, β_{\min} と β_{\max} の設定において適切な値が推測できない場合, β の定義域($\beta_{\min} = 0.3, \beta_{\max} = 1.99$)に設定すれば, オリジナルCSの β の設定の場合に比べて, 使用者によるパラメータの設定にかかる負担を軽減できる。

以下に, 本論文で提案する適応型CSのアルゴリズムを示す。アルゴリズムの終了条件を $T \geq T_{\text{max}}$ とする。また, 探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。

【適応型 Cuckoo Search のアルゴリズム】

Step 0: [準備]

探索点数 m , パラメータ $\alpha \geq 0$, β の下限値 $\beta_{\min} \in [0.3, 1.99)$, 上限値 $\beta_{\max} \in (0.3, 1.99]$, 調整幅 $\Delta\beta \in (0, \beta_{\max} - \beta_{\min}]$, 排斥確率 $P_a \in [0, 1]$, 最大評価回数 T_{max} を定め, 評価回数を $T = 0$ とする。評価指標の目標値 $I_{\text{target}}(T)$ ($T = 1, 2, \dots, T_{\text{max}}$)を与える。

Step 1: [初期化]

探索点の初期位置 x^i ($i = 1, 2, \dots, m$)を初期配置領域 IS 内にランダムに与え, 評価回数を $T = m, \beta = \beta_{\min}$ とする。

Step 2: [Lévy Flight]

参照探索点 x^{r_1} を基準として, 近傍解 \hat{x} を

$$\hat{x} = x^{r_1} + \alpha L(\beta)$$

[‡] 著者らの数値実験に基づいた, 適応型CSのパラメータ推奨値は, $T_{\text{end}} = 0.9 \cdot T_{\text{max}}$, $I_{\text{start}} = 0.1 \cdot x_{\text{width}}$, $I_{\text{end}} = 0.001 \cdot x_{\text{width}}$, $x_{\text{width}} = |x_{\text{max}} - x_{\text{min}}|$ ($x_{\text{max}} = \max\{x_n^i | n = 1, \dots, N; i = 1, \dots, m; T = 0\}$, $x_{\text{min}} = \min\{x_n^i | n = 1, \dots, N; i = 1, \dots, m; T = 0\}$)である。

より, 要素ごとに生成する。ただし, r_1 は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数, $L(\beta) \in \mathbb{R}^N$ は Lévy 乱数ベクトルであり, 各要素 $L_n(\beta)$ は近似 Lévy 分布に従う乱数である。

Step 3: [更新]

更新探索点 x^{r_2} を

$$v^{r_2} = \begin{cases} \hat{x} - x^{r_2}, & f(\hat{x}) < f(x^{r_2}) \\ 0, & \text{otherwise} \end{cases}$$

$$x^{r_2} := x^{r_2} + v^{r_2}$$

より, 更新する。ただし, r_2 は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数である。評価回数 $T := T + 1$ とする。

$$\beta := \begin{cases} \min \{ \beta + \Delta\beta, \beta_{\max} \}, & I \geq I_{\text{target}}(T) \\ \max \{ \beta - \Delta\beta, \beta_{\min} \}, & \text{otherwise} \end{cases}$$

より β を更新する。

Step 4: [排斥]

排斥確率 P_a に従い, 以下の操作を行う。

最悪探索点 $x^{c\text{-worst}}$ を

$$x^{c\text{-worst}} = \underset{x^i}{\operatorname{argmax}} \{ f(x^i) \mid i = 1, \dots, m \}$$

とし,

$$x^{c\text{-worst}} := x^{c\text{-worst}} + \alpha L(\beta)$$

より, 更新する。ただし, $L(\beta) \in \mathbb{R}^N$ は Lévy 乱数ベクトルであり, 各要素 $L_n(\beta)$ は近似 Lévy 分布に従う乱数である。 $T := T + 1$ とする。

$$\beta := \begin{cases} \min \{ \beta + \Delta\beta, \beta_{\max} \}, & I \geq I_{\text{target}}(T) \\ \max \{ \beta - \Delta\beta, \beta_{\min} \}, & \text{otherwise} \end{cases}$$

より β を更新する。

Step 5: [終了判定]

$T \geq T_{\max}$ ならば, 探索を終了する。さもなければ, Step 2 へ戻る。

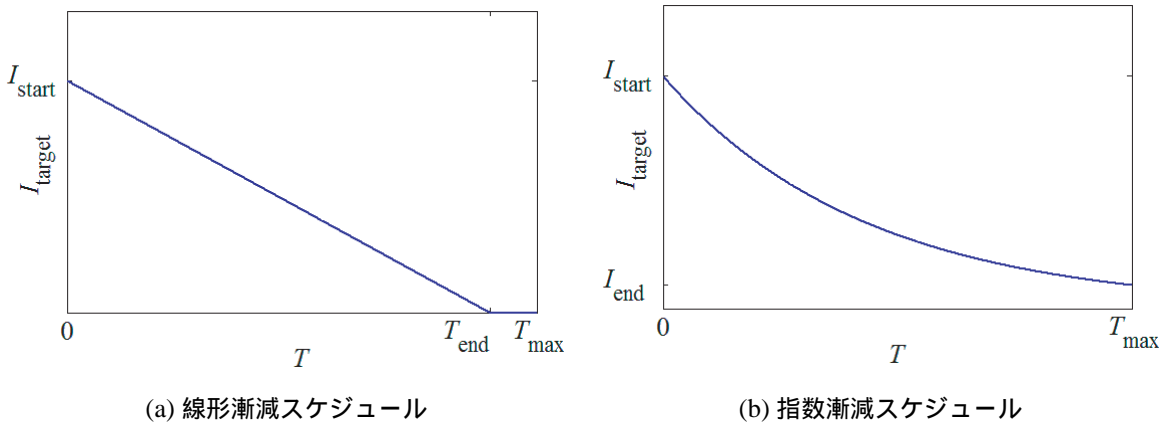


図 5.3 : 評価指標の目標値スケジュール $I_{\text{target}}(T)$

5.5 数値実験による検証

5.5.1 数値実験条件

9 種類の典型的なベンチマーク関数を用いた数値実験を行い, オリジナル CS と比較することで適応型 CS (提案手法) の探索性能を検証する。表 5.1 に数値実験に用いるベンチマーク関数を示す。関数 1, 2, 5 は単峰性関数, それ以外の関数は多峰性関数である。また, 関数 2, 5, 6, 8 は変数間に依存関係を有する関数である。関数 2 は非線形の稜構造, 関数 5, 6 は線形の稜構造, 関数 8 は大域的最適解付近で線形の稜構造を有している。

共通条件として, 探索点数 $m = 10$, 次元数 $N = 10, 50, 100, 300$, ステップサイズ調整変数 $\alpha = 0.1$, 排斥確率 $P_a = 0$, 関数毎に共通の初期配置領域 IS を用い, 終了条件を評価回数 $T_{\max} = 1000, 10000$ とする。また, オリジナル CS では $\beta = 0.3, 0.35, \dots, 1.95, 1.99$ の 35 通りの値を用い, 適応型 CS では下限値 $\beta_{\min} = 0.3$, 上限値 $\beta_{\max} = 1.99$, 調整幅 $\Delta\beta = 0.05$ を用いる。 β 以外のパラメータは, 様々な問題を対象とした数値実験を通じて平均的に優良な結果が得られた値に設定する。50 試行の平均値を実験結果とする。

本実験では、適応型 CS の探索性能を、オリジナル CS の 35 通りの β の値における結果と比較することで評価する。オリジナル CS では、35 通りの結果の中で、最も優れている結果（最良値）を 1st、8 番目に優れている結果（第一四分位数）を 8th、17 番目に優れている結果（中央値）を 17th、35 通りの結果の平均値を Mean と表現する。手法や問題に対する知識や経験を有していない使用者は、 β を定義域の中からランダムに決定することが考えられる。そのため、Mean はこのような使用者による平均的な結果と想定できる。また、オリジナル CS の結果には設定した β の値も示す。

適応型 CS では、 I_{target} を図 5.3 の線形および指数減少スケジュールに設定した条件で得た結果をそれぞれ Lin, Exp と表現する。また、オリジナル CS の 35 通りと適応型 CS を合わせた 36 通りの結果における、適応型 CS の順位を Rank と表現する。Rank の値は、適応型 CS がオリジナル CS のどのような順位より優れているかを示す。

5.5.2 実験結果・考察

表 5.2、表 5.3 に数値実験結果を示す。表 5.2、表 5.3 では、適応型 CS が、オリジナル CS の Mean よりも優れている場合は ▲ を、17th よりも優れている場合は ■ を、8th よりも優れている場合は ◆ を、1st よりも優れている場合は ● を付す。

表 5.2 より、適応型 CS はほぼ全ての条件において、Mean、17th よりも優れていることが確認できる。この結果は、対象問題の種類、次元数、終了条件などに関わらず、適応型 CS では、オリジナル CS でパラメータをランダムに設定した場合よりも得られる結果の期待値が高いことを表す。さらに、適応型 CS はいくつかの実験条件（特に $T_{\text{max}} = 10000$ ）において、8th や 1st よりも優れていることが確認できる。以上の結果から、適応型 CS がオリジナル CS よりも全体的に探索性能が優れているといえる。

また、図 5.4 に Rosenbrock's Saddle Function、Rastrigin Function、Griewank Function ($N = 50$) を対象とした場合の β 、多様化・集中化の評価指標 I 、摂動の評価指標 P 、探索点分布の評価指標 D 、最良解の目的関数値 $f(x^{g-\text{best}})$ の推移を示す。図 5.4 より、適応型 CS では、探索過程で I が目標値 I_{target} に追従することで徐々に漸減すると同時に、 $f(x^{g-\text{best}})$ が改善する様子を確認できる。5.2 節と 5.3 節で述べたように、関数毎に設定される初期配置領域 IS のサイズによって適切な β の値は異なるため、オリジナル CS では探索条件に応じ

て β を設定する必要がある。さらに、オリジナルCSでは探索過程で探索状態が大きく変化しないため、探索戦略が適切に実現されない。一方で、適応型CSでは、本実験の結果から、 I を目標値に追従させることで、探索序盤で初期配置領域 IS のサイズに応じた値へ β を適応的に調整しながら、探索過程で探索状態に応じて適切な多様化・集中化を実現していることがいえる。したがって、探索状態の評価と制御に基づくパラメータの調整則を導入することで、CSの多様化・集中化の調整能力が向上した結果、CSの適応能力・探索性能を向上させることができた。

5.6 本章のまとめ

本章では、適応化のフレームワークに従い、一般的なメタヒューリスティクスにおける探索戦略に基づくアプローチにより、高い適応能力・探索性能を有する適応型CSを開発した。はじめに、CSの解析を通じて、多様化・集中化の観点からパラメータと探索ダイナミクスの関係を明らかにした。さらに、解析結果に基づき、CSの多様化・集中化の実現状態の評価指標を提案し、数値実験を通じて評価指標の有用性を検証した。そして、メタヒューリスティクスで有効な探索戦略に基づき、事前に設定された目標値スケジュールに、評価指標が追従する適応型CSを提案した。典型的なベンチマーク関数を用いた数値実験により、提案手法の汎用性・有用性を確認した。

表 5.1：ベンチマーク関数

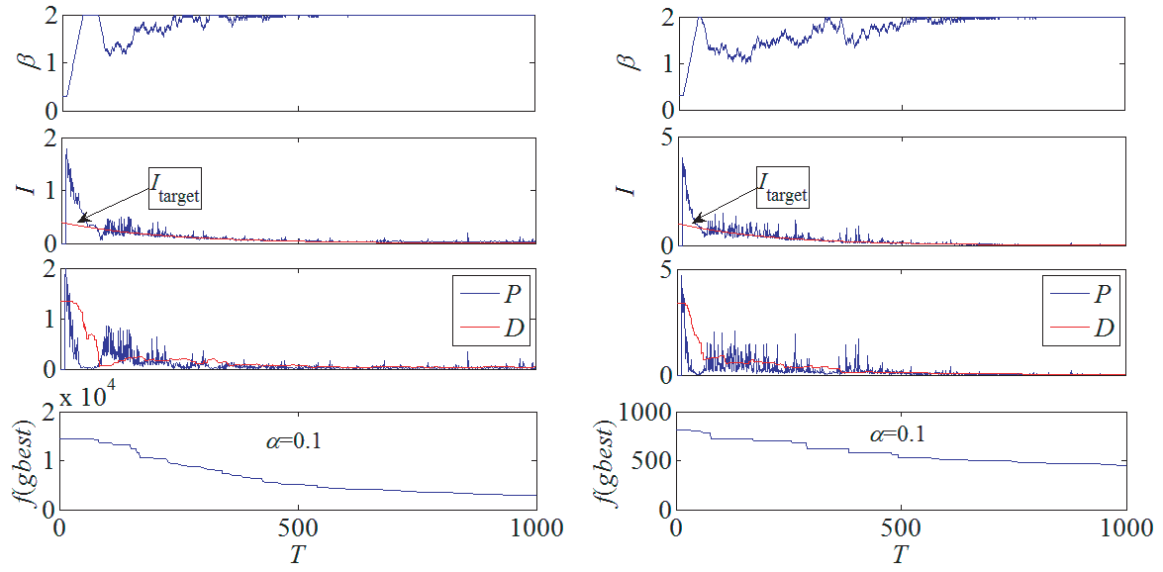
関数	目的関数 $f(x)$	大域的最適解 x^o	大域的最適値 $f(x^o)$	初期配置領域 IS
1. Parabola	$f_1(x) = \sum_{n=1}^N x_n^2$	$(0, \dots, 0)$	0	$[-5 \ 5]^N$
2. Rosenbrock	$f_2(x) = \sum_{n=1}^{N-1} [100(x_n^2 - x_{n+1})^2 + (1 - x_n)^2]$	$(1, \dots, 1)$	0	$[-2 \ 2]^N$
3. 2^N minima	$f_3(x) = \sum_{n=1}^N \{x_n^4 - 16x_n^2 + 5x_n\}$	$\approx (-2.9, \dots, -2.9)$	$\approx -78N$	$[-5 \ 5]^N$
4. Rastrigin	$f_4(x) = \sum_{n=1}^N \{x_n^2 - 10 \cos(2\pi x_n) + 10\}$	$(0, \dots, 0)$	0	$[-5 \ 5]^N$
5. Schwefel	$f_5(x) = \sum_{n=1}^N (\sum_{i=1}^n x_i)^2$	$(0, \dots, 0)$	0	$[-5 \ 5]^N$
6. Lévy	$f_6(x) = \frac{\pi}{N} \left\{ \sum_{n=1}^{N-1} [(x_n - 1)^2 (1 + 10 \sin^2(\pi x_{n+1}))] + 10 \sin^2(\pi x_1) + (x_N - 1)^2 \right\}$	$(1, \dots, 1)$	0	$[-5 \ 5]^N$
7. Ackley	$f_7(x) = -20 \exp \left\{ -0.2 \sqrt{\frac{1}{N} \sum_{n=1}^N x_n^2} \right\} - \exp \left\{ \frac{1}{N} \sum_{n=1}^N \cos 2\pi x_n \right\} + 20 + e$	$(0, \dots, 0)$	0	$[-5 \ 5]^N$
8. Griewank	$f_8(x) = \frac{1}{4000} \sum_{n=1}^N x_n^2 - \prod_{n=1}^N \cos \left(\frac{x_n}{\sqrt{n}} \right) + 1$	$(0, \dots, 0)$	0	$[-50 \ 50]^N$
9. Alpine	$f_9(x) = \sum_{n=1}^N x_n \sin x_n + 0.1 x_n $	$(0, \dots, 0)$	0	$[-10 \ 10]^N$

表 5.2 : 数値実験結果 (適応型 Cuckoo Search , 評価回数 $T = 1000$)

関数	N	オリジナル Cuckoo Search (35 種類 β)						適応型 Cuckoo Search				
		1st	(β)	8th	(β)	17th	(β)	Mean	Lin	Rank	Exp	Rank
1. Parabola	10	0.08	(1.60)	0.11	(1.35)	0.29	(1.00)	3.51	0.09	2	0.22	14
	50	68.8	(1.50)	82.0	(1.65)	123	(1.85)	177	117	16	150	20
	100	320	(1.60)	344	(1.35)	482	(1.10)	522	452	16	459	17
	300	1740	(1.75)	1827	(1.90)	2192	(1.20)	2109	2061	14	1990	13
2. Rosenbrock	10	15.6	(1.75)	17.9	(1.60)	29.7	(1.20)	148	13.3	1	23.6	15
	50	1649	(1.85)	2232	(1.95)	5320	(1.20)	8279	2454	10	3151	12
	100	9509	(1.90)	13712	(1.60)	28983	(1.20)	25863	14797	9	13614	8
	300	76273	(1.95)	104251	(1.65)	119655	(1.20)	111864	96619	7	87326	5
3. 2^N minima	10	-718	(0.95)	-702	(0.85)	-685	(0.65)	-665	-699	10	-681	21
	50	-2778	(1.60)	-2731	(1.40)	-2471	(1.95)	-2117	-2532	16	-2512	17
	100	-4675	(1.75)	-4493	(1.50)	-3576	(1.15)	-3243	-3906	16	-4047	15
	300	-9477	(1.85)	-8141	(1.60)	-4592	(1.20)	-5853	-6335	14	-7650	10
4. Rastrigin	10	41.9	(1.00)	43.7	(1.25)	46.7	(1.45)	52.8	30.8	1	37.8	1
	50	467	(1.95)	539	(1.60)	585	(1.25)	628	515	6	455	1
	100	1157	(1.99)	1307	(1.65)	1404	(1.20)	1456	1279	7	1181	3
	300	4294	(1.99)	4669	(1.65)	4985	(1.20)	4947	4628	8	4403	4
5. Schwefel	10	5.98	(1.35)	7.09	(1.65)	13.4	(1.75)	91.7	8.69	13	18.9	21
	50	37208	(1.45)	44502	(1.25)	68744	(1.85)	116606	67886	16	83286	19
	100	748580	(1.60)	823863	(1.35)	1184932	(1.10)	1459967	1117188	16	1217199	18
	300	42969321	(1.65)	45610753	(1.85)	58315435	(1.99)	57522198	54229869	15	52207603	14
6. Levy	10	3.62	(0.90)	5.19	(0.80)	9.26	(0.55)	11.9	7.55	15	12.7	21
	50	40.1	(1.95)	53.8	(1.65)	60.3	(1.15)	74.5	47.9	5	39.6	1
	100	59.2	(1.99)	79.9	(1.65)	94.9	(1.20)	103	75.6	7	62.7	3
	300	97.3	(1.99)	118	(1.65)	137	(1.20)	137	116	7	101	3
7. Ackley	10	1.38	(1.35)	1.64	(1.10)	2.54	(0.75)	3.35	1.52	5	2.87	19
	50	6.47	(1.35)	6.80	(1.15)	7.74	(0.95)	8.05	7.24	12	7.66	16
	100	8.25	(1.40)	8.45	(1.65)	8.93	(1.90)	9.16	8.70	12	8.71	12
	300	9.63	(1.65)	9.68	(1.90)	10.0	(1.20)	10.0	9.83	14	9.69	10
8. Griewank	10	0.44	(0.80)	0.57	(0.55)	0.92	(1.25)	0.95	0.56	7	0.41	1
	50	3.46	(0.80)	4.03	(1.05)	6.16	(0.55)	6.3	4.88	12	5.13	14
	100	10.6	(0.95)	11.8	(1.15)	14.9	(0.65)	15.0	13.9	15	13.5	13
	300	47.0	(1.10)	49.1	(0.95)	53.6	(1.65)	53.7	54.6	20	52.1	14
9. Alpine	10	1.39	(1.85)	1.80	(1.55)	2.24	(1.15)	3.11	1.63	6	1.08	1
	50	49.9	(1.85)	53.9	(1.55)	64.2	(1.15)	75.4	60.9	17	50.3	4
	100	150	(1.85)	157	(1.55)	181	(1.15)	198	176	16	155	7
	300	638	(1.90)	667	(1.60)	738	(1.20)	751	722	16	668	9

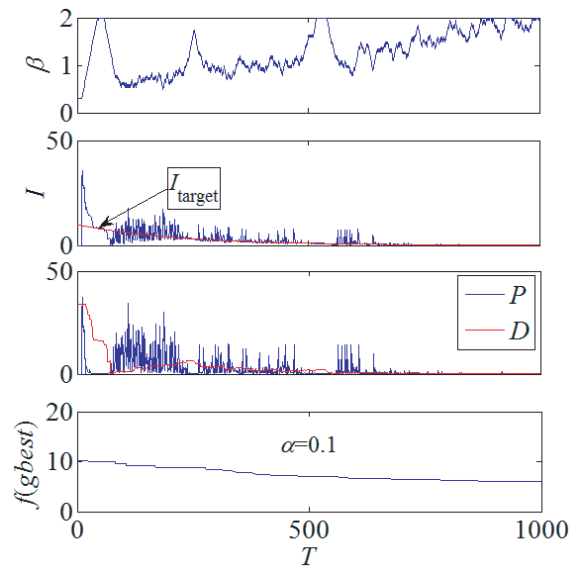
表 5.3 : 数値実験結果 (適応型 Cuckoo Search , 評価回数 $T = 10000$)

関数	N	オリジナル Cuckoo Search (35 種類 β)				適応型 Cuckoo Search			
		1st (β)	8th (β)	17th (β)	Mean	Lin	Rank	Exp	Rank
1. Parabola	10	0.00 (1.99)	0.01 (1.65)	0.02 (1.25)	0.24	0.00	2	0.00	2
	50	0.11 (1.99)	1.13 (1.65)	5.86 (1.20)	78.8	0.17	2	0.05	1
	100	1.87 (1.95)	11.6 (1.60)	59.8 (1.20)	266	6.80	6	0.66	1
	300	146 (1.90)	353 (1.60)	1433 (1.20)	1413	677	13	323	8
2. Rosenbrock	10	5.87 (1.99)	7.67 (1.65)	9.48 (1.10)	26.1	6.22	2	6.03	2
	50	76.9 (1.99)	214 (1.65)	791 (1.20)	5119	91.1	3	74.6	1
	100	269 (1.99)	1396 (1.65)	10686 (1.20)	17549	501	3	300	2
	300	4805 (1.99)	44367 (1.65)	119185 (1.20)	90762	16878	5	6429	3
3. 2^N minima	10	-782 (1.10)	-781 (1.20)	-775 (0.60)	-750	-783	1	-776	17
	50	-3362 (1.65)	-3338 (1.45)	-3252 (1.20)	-2728	-3432	1	-3415	1
	100	-6508 (1.95)	-6328 (1.65)	-5550 (1.20)	-4541	-6443	6	-6616	1
	300	-17825 (1.95)	-15274 (1.65)	-8091 (1.20)	-9200	-14641	10	-17013	5
4. Rastrigin	10	15.9 (1.20)	17.1 (1.45)	19.0 (0.85)	23.2	9.80	1	10.6	1
	50	333 (1.99)	405 (1.70)	438 (1.20)	509	232	1	243	1
	100	766 (1.99)	1094 (1.65)	1193 (1.25)	1291	761	1	699	1
	300	2973 (1.99)	4324 (1.65)	4750 (1.20)	4710	3689	3	3304	2
5. Schwefel	10	0.01 (1.99)	0.21 (1.65)	0.58 (1.20)	5.55	0.02	2	0.01	2
	50	394 (1.95)	1218 (1.65)	4414 (1.20)	53810	1315	9	577	3
	100	13689 (1.95)	41408 (1.60)	161847 (1.20)	765599	55417	10	25845	5
	300	3620172 (1.90)	8112337 (1.60)	29494433 (1.20)	37266326	13136408	13	6662217	7
6. Levy	10	0.01 (1.40)	0.02 (1.65)	0.09 (1.55)	1.39	0.00	1	0.00	1
	50	24.6 (1.99)	30.4 (1.30)	34.3 (1.55)	51.0	17.3	1	18.6	1
	100	30.2 (1.99)	58.9 (1.65)	70.5 (1.20)	84.9	34.8	2	30.4	2
	300	47.0 (1.99)	101 (1.65)	124 (1.20)	125	69.8	3	57.0	2
7. Ackley	10	0.12 (1.80)	0.18 (1.60)	0.31 (1.15)	0.87	0.03	1	0.03	1
	50	1.90 (1.70)	2.50 (1.40)	3.60 (1.05)	4.99	1.65	1	1.20	1
	100	3.40 (1.70)	3.88 (1.40)	6.03 (1.05)	6.66	3.49	2	3.44	2
	300	6.74 (1.65)	7.42 (1.85)	9.10 (1.20)	8.95	8.02	11	7.54	10
8. Griewank	10	0.13 (0.60)	0.16 (0.45)	0.31 (1.15)	0.49	0.10	1	0.11	1
	50	0.18 (1.45)	0.26 (1.20)	0.82 (0.85)	1.81	0.81	17	0.10	1
	100	0.95 (1.25)	1.07 (1.45)	1.65 (1.60)	5.43	1.63	17	0.89	1
	300	5.93 (1.20)	8.07 (1.05)	21.8 (0.85)	28.9	27.7	20	14.1	14
9. Alpine	10	0.16 (1.99)	0.60 (1.60)	0.73 (0.65)	0.87	0.08	1	0.13	1
	50	10.3 (1.99)	24.3 (1.65)	36.7 (1.20)	48.8	18.0	5	13.1	3
	100	41.6 (1.99)	85.6 (1.65)	123 (1.20)	146	75.3	6	54.2	3
	300	297 (1.99)	476 (1.65)	630 (1.20)	641	472	8	365	3



(a) Rosenbrock

(b) Rastrigin



(c) Griewank

図 5.4 : 評価指標 I と最良解の目的関数値 $f(x^{g-\text{best}})$ の推移 (適応型 Cuckoo Search , $N = 50$)

6

適応化のフレームワーク に基づく メタヒューリスティクス

本章では，7章で適応化のフレームワークに基づき適応型メタヒューリスティクスを開発することを踏まえ，適応化設計論に基づく新たなメタヒューリスティクスを設計する。高い適応能力を有するように設計したメタヒューリスティクスを適応化すれば，優れた適応能力・探索性能を有する適応型メタヒューリスティクスの構築が期待できる。

6.1 適応化設計論に基づくメタヒューリスティクスの設計

4章で述べたように，メタヒューリスティクスは環境・状況に適応するように多様化・集中化の実現状態を調整する操作を具備している。そこで，適応化のフレームワークに従い，多様化・集中化に基づく高い適応能力・探索性能を潜在的に有する手法を新たに構築することで，優れた適応型メタヒューリスティクスの開発が期待できる。本章では，その第一歩として，適応化設計論（ADM）に基づき，高い適応能力・探索性能を有する新たなメタヒューリスティクスを構築する。

4.3節で述べたように，ADMでは，適応化のフレームワークに基づくメタヒューリスティクスの枠組みとなるアルゴリズムを用いて，近傍の生成と探索点の更新を決定する。提案手法の更新式を式(6.1)，式(6.2)，式(6.3)に示す。

$$\hat{x}^i(k) = x^i(k) + \alpha R u^{better} + \beta \phi u^{random} \quad (6.1)$$

$$v^i(k) = \begin{cases} \hat{x}^i(k) - x^i(k), & f(\hat{x}^i(k)) < f(x^i(k)) \\ 0, & \text{otherwise} \end{cases} \quad (6.2)$$

$$x^i(k+1) = x^i(k) + v^i(k) \quad (6.3)$$

ただし, $\alpha, \beta \geq 0$ はパラメータ, $u^{better} = x^{better, i}(k) - x^i(k)$, $u^{random} = x^r - x^i$, $x^{better, i}$ は式 (2.17) で定義される better, $r \neq i$ は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数, $R, \phi \in \mathbb{R}^{N \times N}$ は対角行列であり, R の対角要素は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従う乱数, ϕ の対角要素は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(-0.5, 0.5)$ に従う乱数である。

図 6.1 に, 提案手法の近傍生成について示す。式 (6.1) で表されるように, 提案手法では, $\hat{x}^i - x_i^i$ が, 探索点 x^i からランダムな探索点 x^r に向かう差分ベクトル u^{random} と, 探索点 x^i から better $x^{better, i}$ に向かう差分ベクトル u^{better} の線形結合で表され, 近傍解 \hat{x}^i は基底ベクトルの係数の乱数 R, ϕ による摂動に基づく近傍内に生成される。提案手法は改善移動であるため, 探索点 x^i は目的関数値が改善した場合, 近傍解 \hat{x}^i へ移動する。

この探索構造は, 4.3 節の ADM の説明で述べた「優れた探索構造となる必要条件」に基づいて構成されている。探索点間の差分ベクトル u^{better} , u^{random} を用いることで, 良い解を活用しながら, 特定の領域・方向に対する指向性を促進することが可能となる。そして, 基底ベクトルの係数に一様乱数を用いることで, 摂動・探索点分布の縮小が可能となる。また, 一様乱数 ϕ の区間を $[-0.5, 0.5]$ とすることで, u^{random} とは逆の方向にも摂動を生じさせ, 一定以上の摂動の拡大が可能となる。さらに, これらの操作は探索過程で多様化・集中化に対する影響が変化するため, 有効な探索戦略が実現可能な構造である。以上から, 提案手法は, 3 章の解析に基づく優れた探索構造を有しているため, 高い適応能力・探索性能が期待できる。

以下に, 本論文で提案した適応化のフレームワークに基づくメタヒューリスティクス (提案手法) のアルゴリズムを示す。アルゴリズムの終了条件を $k = k_{\max}$ とする。評価回数は $T = m(k_{\max} + 1)$ となる。探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。

【提案手法のアルゴリズム】

Step 0: [準備]

探索点数 m , パラメータ $\alpha, \beta \geq 0$, 最大反復回数 k_{\max} を定め, 反復回数を $k = 1$ とする。

Step 1: [初期化]

探索点の初期位置 $x^i(k)$ ($i = 1, 2, \dots, m$) を初期配置領域 IS 内にランダムに与える。

Step 2: [近傍の生成]

各探索点 $x^i(k)$ について, 優良探索点群 $Better^i$, 近傍解 $\hat{x}^i(k)$ を

$$Better^i = \{x^\ell(k) \mid f(x^\ell(k)) < f(x^i(k)); \ell = 1, \dots, m\}$$

$$\hat{x}^i(k) = \begin{cases} x^i(k) + \alpha R u^{better} + \beta \phi u^{random}, & Better^i \neq \emptyset \\ x^i(k) + \beta \phi u^{random}, & \text{otherwise} \end{cases}$$

より, 生成する。

ただし, $u^{better} = x^{i,better}(k) - x^i(k)$, $u^{random} = x^r(k) - x^i(k)$, $x^{i,better}(k) \in Better^i$ は優良探索点群 $Better^i$ からランダムに選ばれた探索点, \emptyset は空集合, r は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数, $R, \phi \in \mathbb{R}^{N \times N}$ は対角行列であり, R の対角要素は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従う乱数, ϕ の対角要素は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(-0.5, 0.5)$ に従う乱数である。

Step 3: [探索点の更新]

各探索点 $x^i(k)$ について,

$$v^i(k) = \begin{cases} \hat{x}^i(k) - x^i(k), & f(\hat{x}^i(k)) < f(x^i(k)) \\ 0, & \text{otherwise} \end{cases}$$

$$x^i(k+1) = x^i(k) + v^i(k)$$

より, 更新する。

Step 4: [終了判定]

$k = k_{\max}$ ならば, 探索を終了する。さもなければ, $k := k + 1$ とし, Step 2 へ戻る。

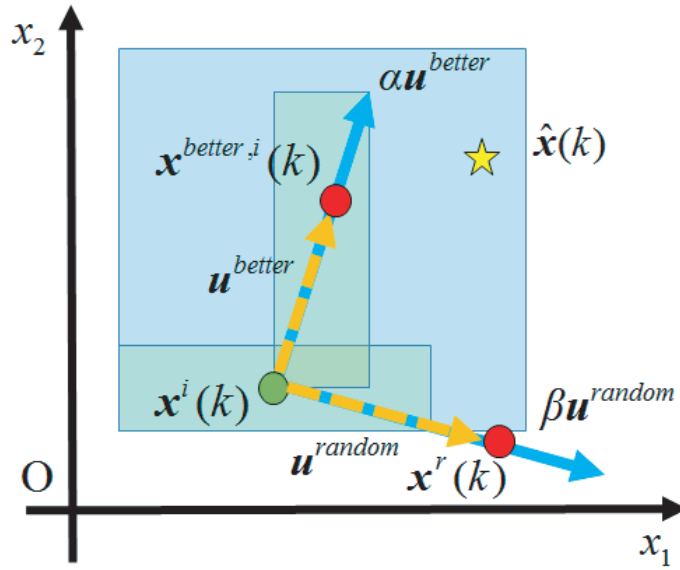


図 6.1：提案手法の近傍の生成

6.2 適応化のフレームワークに基づくメタヒューリスティクスの多様化・集中化の検証

本節では、6.1 節で構築した提案手法における多様化・集中化の数値実験的検証を行う。検証には、式 (5.5) で定義された、摂動・探索点分布に対する評価指標 I を用いる。Rosenbrock's Saddle Function, Rastrigin Function, Griewank Function を用いた数値実験を行い、評価指標 I と探索ダイナミクスとの関係を明らかにすることで、提案手法における多様化・集中化の調整能力を検証する。共通の実験条件として、探索点数 $m = 20$ 、次元数 $N = 50$ 、パラメータ $\alpha = 1.2$ 、関数毎に共通の初期配置領域 IS を用い、終了条件を反復回数 $k_{\max} = 1000$ とする。

図 6.2, 図 6.3, 図 6.4 に、 $\beta = 1.5, 2.0, 2.5$ に設定した場合の評価指標 I と最良解の評価値 $f(x^{g-best})$ の推移を示す。どの β の値でも、多様化・集中化の評価指標 I は探索過程で徐々に漸減しながら、 $f(x^{g-best})$ が改善する様子を確認できる。これは、3.7 節で述べた探索戦略を実現していることを示している。6.1 節でも述べたように、近傍の生成において、探索

点間の差分ベクトルや良い解を用いたため、探索序盤では多様化、探索過程で多様化から徐々に集中化へ推移し、探索終盤では集中化を実現したことが考えられる。

さらに、パラメータ β の値が異なると、評価指標 I の漸減速度が異なることがわかる。特に、 β が大きい場合、評価指標 I の漸減速度が遅く、 β が小さい場合、 I の漸減速度が速い。これは、3.7節で定義したように、 β が大きい場合、探索過程で摂動・探索点分布が広い状態が続くために、特定の領域に対する指向性を抑制し、 β が小さい場合、探索序盤から摂動・探索点分布が狭い状態のために、特定の領域に対する指向性を促進していると考えられる。 I の数値実験的検証から、提案手法は多様化・集中化の調整能力が高く、パラメータ β の値により、その漸減速度が異なることが明らかになった。また、初期配置領域 IS のサイズ b が異なると、評価指標 I の漸減速度が異なることがわかる。これは、CSと同様に、 IS のサイズ b に応じて探索点分布・摂動の適切な範囲が異なるためだと考えられる。

したがって、数値実験を通じて、① 提案手法は多様化・集中化（探索状態）の調整能力が高いことと、② β の値により、提案手法の多様化・集中化の実現状態へ与える影響力が異なること、を明らかにした。以上の検証から、近傍の生成において多様化・集中化の調整能力に影響を与える操作は、高い適応能力の実現に貢献することが期待できる。

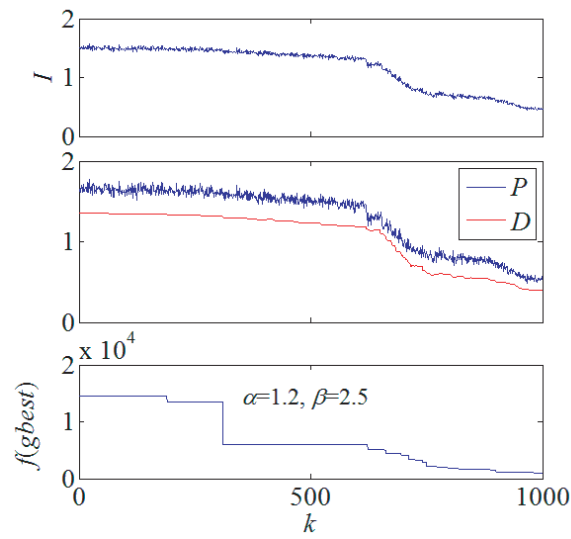
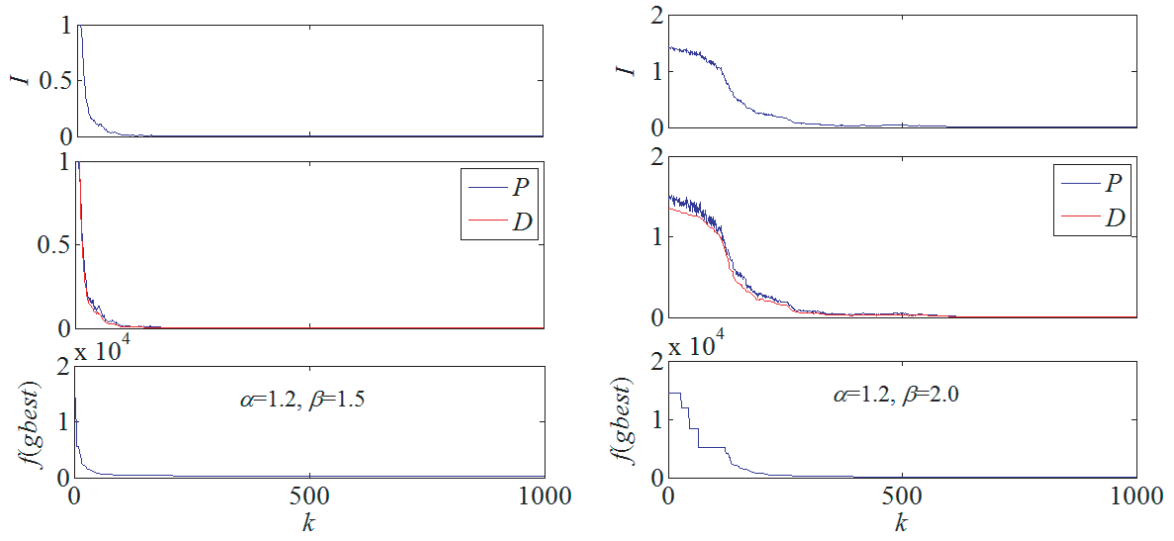
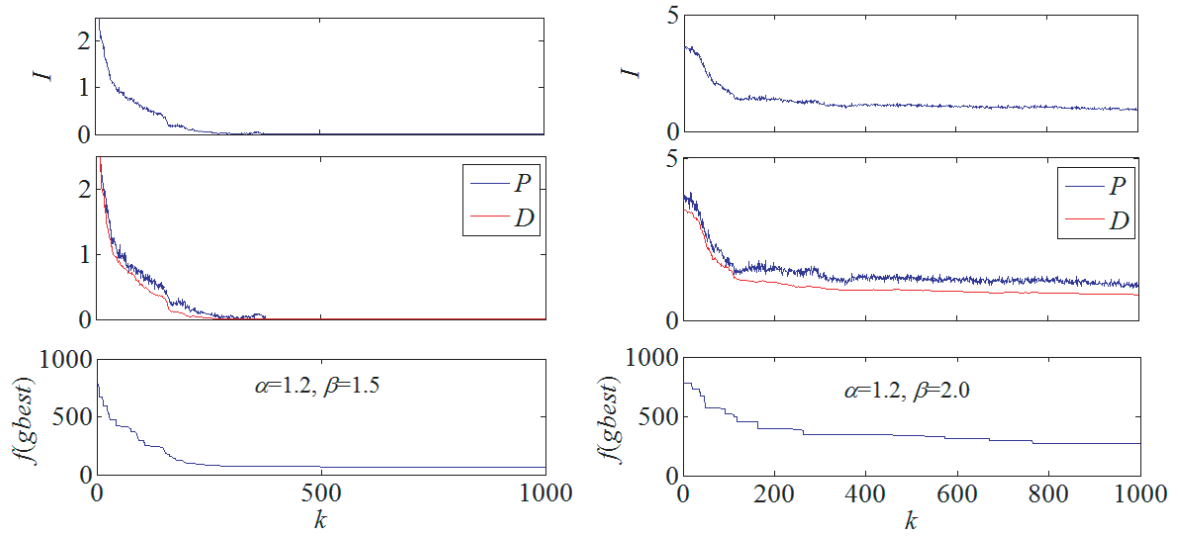
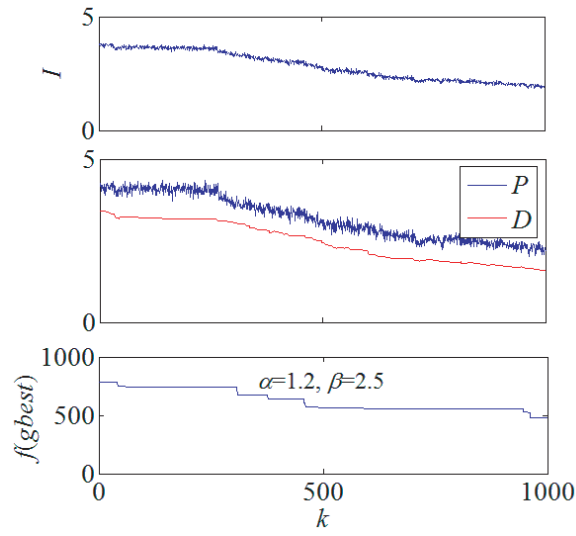
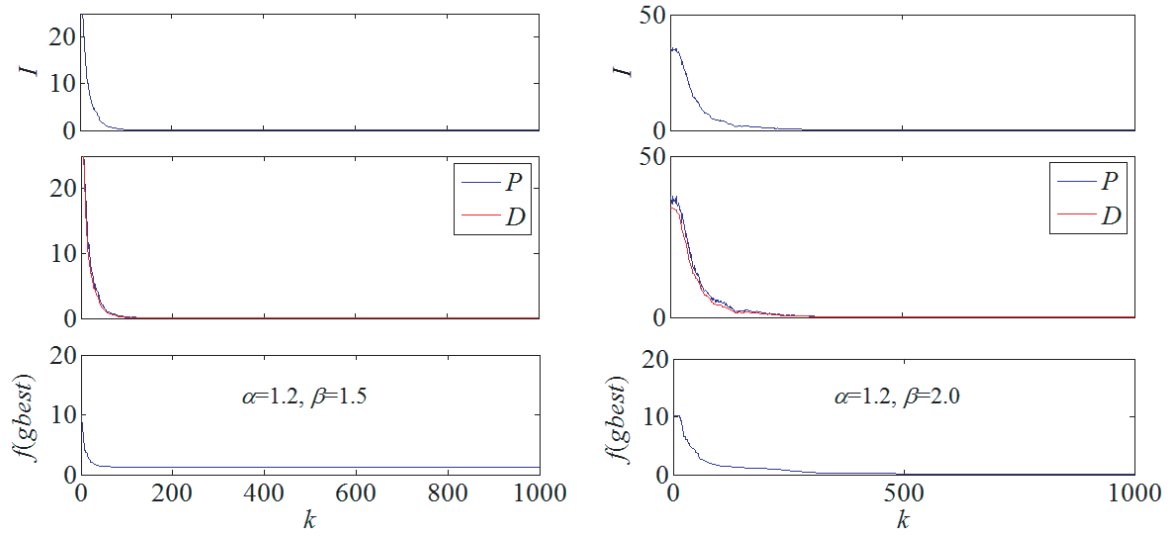
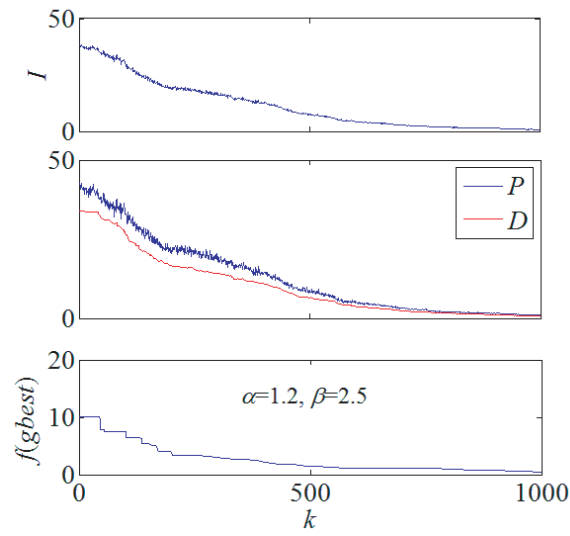


図 6.2 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (提案手法, Rosenbrock, $N = 50$)

(a) $\beta = 1.5$ (Rastrigin)(b) $\beta = 2.0$ (Rastrigin)(c) $\beta = 2.5$ (Rastrigin)図 6.3 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (提案手法, Rastrigin, $N = 50$)

(a) $\beta = 1.5$ (Griewank)(b) $\beta = 2.0$ (Griewank)(c) $\beta = 2.5$ (Griewank)図 6.4 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (提案手法, Griewank, $N = 50$)

6.3 数値実験による検証

6.3.1 数値実験条件

典型的なベンチマーク関数を用いた数値実験を行い、代表的な手法である PSO, DE, FA, ABC と比較することで、提案手法の探索性能を検証する。数値実験には、表 5.1 のベンチマーク関数を用いる。共通条件として、探索点数 $m = 20$ 、次元数 $N = 10, 50, 100, 300$ 、関数毎に共通の初期配置領域 IS を用い、終了条件を評価回数 $T_{\max} = 2000, 20000$ とする。また、パラメータとして、提案手法 (Proposed Method: PM) では $\alpha = 1.2$, $\beta = 1.2, 1.4, 2.0$ の 3 通り、PSO では Constriction Method ($w = 0.729, c_1 = c_2 = 1.4955$) [22][35]、DE/rand/1 では膨張率 $F = 0.4$ 、交叉率 $CR = 0.4$ 、FA では $\alpha = 0.2, \beta_0 = 1, \gamma = 0.0001$ 、ABC では $Limit = 60, m_o = 20$ を用いる。各パラメータは、様々な問題を対象とした数値実験を通じて平均的に優良な結果が得られた値に設定する。50 試行の平均値を実験結果とする。

6.3.2 実験結果・考察

表 6.1, 表 6.2 に PSO, DE, FA, ABC, PM ($\beta = 1.2$), PM ($\beta = 1.4$), PM ($\beta = 2.0$) の数値実験結果を示す。表 6.1, 表 6.2 では、七種類の手法の中で最も結果が優れている手法に ●, 二番目に優れている手法に ◆, 三番目に優れている手法に ■, 四番目に優れている手法に ▲ を付す。また、Rosenbrock's Saddle Function, Rastrigin Function, Griewank Function を対象としたとき、図 6.5, 図 6.6, 図 6.7 に提案手法の場合、図 6.8, 図 6.9, 図 6.10 に PSO, DE, FA, ABC の場合の評価指標 I と最良解の評価値 $f(x^{g-best})$ の推移を示す。

表 6.1 (評価回数 $T = 2000$) では、 $\beta = 1.2, 1.4$ に設定した提案手法は、全体的に他の手法と同等あるいは同等以上であることが確認できる。特に PSO と比べると全体的に優れている。これは、図 6.5, 図 6.6, 図 6.7 からわかるように、評価回数 $T = 2000$ の探索において、 $\beta = 1.2, 1.4$ の設定では、評価指標が探索過程で徐々に漸減し、探索終盤では集中化を実現しているためだと考えられる。しかし、 $\beta = 2.0$ に設定した場合の提案手法は、他の手法と比べて劣る場合がある。これは、図 6.5, 図 6.6, 図 6.7 からわかるように、評価回数 $T = 2000$ の探索において、 $\beta = 2.0$ の設定では十分に集中化を実現できない場合があるためだと考えられる。また、表 6.2 (評価回数 $T = 20000$) では、 $\beta = 2.0$ に設定した提案手法

は、全体的に PSO や FA と同等程度であることが確認できる。これは、図 6.5、図 6.6、図 6.7 からわかるように、評価回数 $T = 20000$ の探索において、 $\beta = 2.0$ の設定では、評価指標が探索過程で徐々に漸減し、探索終盤では集中化を実現しているためだと考えられる。

しかし、提案手法は、高次元において他の手法と比べて劣ることが多い。図 6.11 に Rastrigin Function を対象とした場合の、各次元数における提案手法の評価指標の推移を示す。図 6.11 より、 β が小さい場合は、どの次元においても評価指標が探索過程で漸減することに対して、 β が大きい場合は、高次元になるほど、探索過程における評価指標の漸減速度が遅くなる、つまり十分に集中化を実現できていない様子が確認できる。しかし、評価回数 $T = 20000$ においては、 β が小さい場合は、どの次元においても探索過程における評価指標の漸減速度が速い、つまり探索序盤から集中化を実現していることに対して、 β が大きい場合は、 $N = 100$ までの次元では、探索過程で評価指標が徐々に漸減している様子が確認できる。これは、パラメータ、評価回数、次元数によって、多様化・集中化の実現状態の調整が変化することを示している。

図 6.8、図 6.9、図 6.10 から、PSO、DE、FA、ABC では、探索過程で徐々に探索状態が推移する様子が確認できる。これは、3 章で解析したように、メタヒューリスティクスは潜在的な適応能力を有しているためだと考えられる。以上から、各手法の潜在的な適応能力を向上させることで、探索性能の向上へ貢献することが期待できる。

図 6.12 に Rastrigin Function を対象とした場合の、各次元数における PSO、DE、FA、ABC の評価指標の推移を示す。図 6.12 より、各手法の探索性能と関連性が確認できる。提案手法と同様に、PSO では、高次元になるほど、探索過程における評価指標の漸減速度が遅くなることから、十分に集中化を実現していない様子が確認できる。また、FA では、次元による影響を受けていないが、評価指標は探索過程で徐々に漸減していないことから、十分に集中化を実現していない様子が確認できる。これらに対して、DE、ABC では、高次元になるほど、探索過程における評価指標の漸減速度は多少遅くなるものの、探索終盤の評価指標の値は十分小さいことから、探索戦略を高いレベルで実現している様子が確認できる。これは、DE と ABC は、3.9 節で探索構造が優れていると結論付けたように、評価回数、次元数に依存せずに、多様化・集中化の実現状態の調整がされる、つまり潜在的な素質としての適応能力が高いことを示している。

以上から、提案手法は、他の優れたメタヒューリスティクスと同様に、探索戦略を実現可能な探索構造を有していることが考えられる。また、適応化のフレームワークに従い、

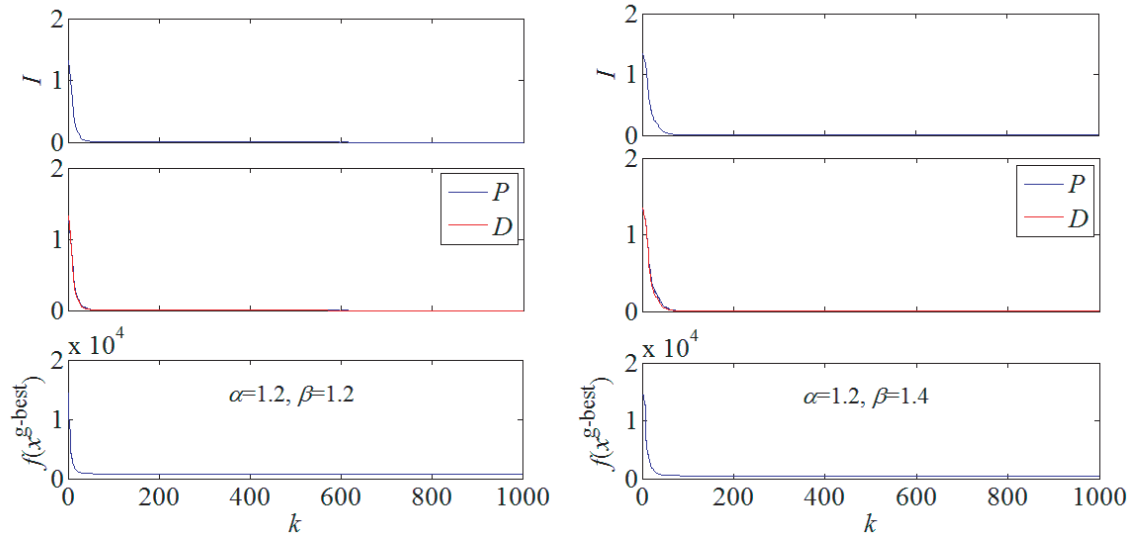
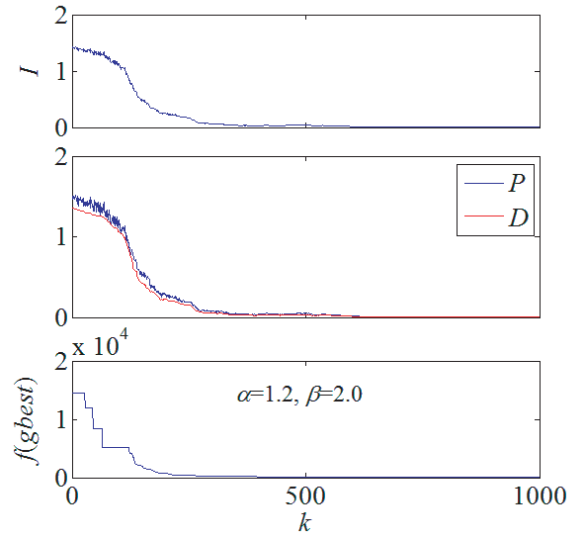
適応能力を備えた探索構造を簡潔に構築しただけで、他の優れたメタヒューリスティクスと同等、あるいは同等以上の探索性能を実現することができた。この結果から、適応化のフレームワークに従うメタヒューリスティクスの設計は、一定の有用性を有しているといえる。一方で、提案手法は、次元数、評価回数、初期配置領域によって、適切なパラメータを設定しなければ探索性能を十分に発揮していないため、その適応能力はまだ向上の余地がある。しかしながら、「高い適応能力を備えた探索構造」となるような具体的な条件・操作をより豊かにすることができれば、この提案手法よりもはるかに優れた探索性能・適応能力を有するメタヒューリスティクスの設計が期待できる。

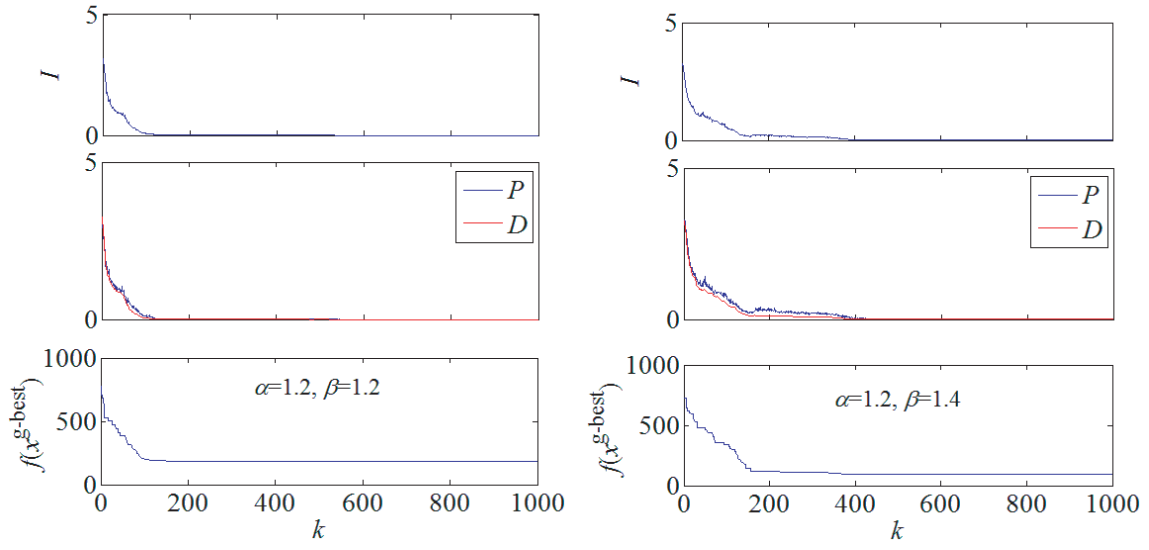
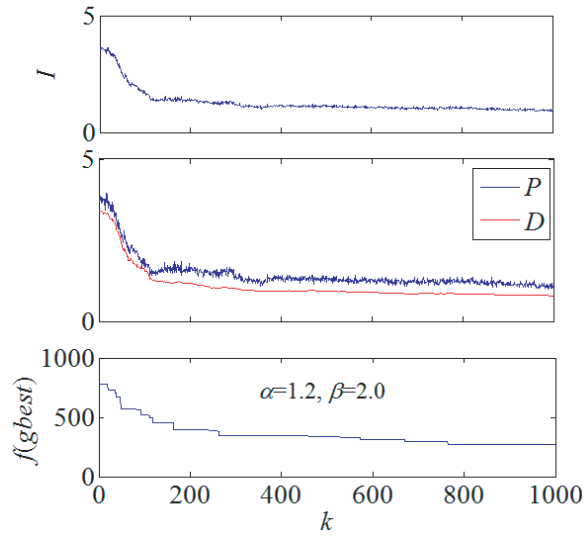
表 6.1: 数値実験結果 (提案手法, 評価回数 $T = 2000$)

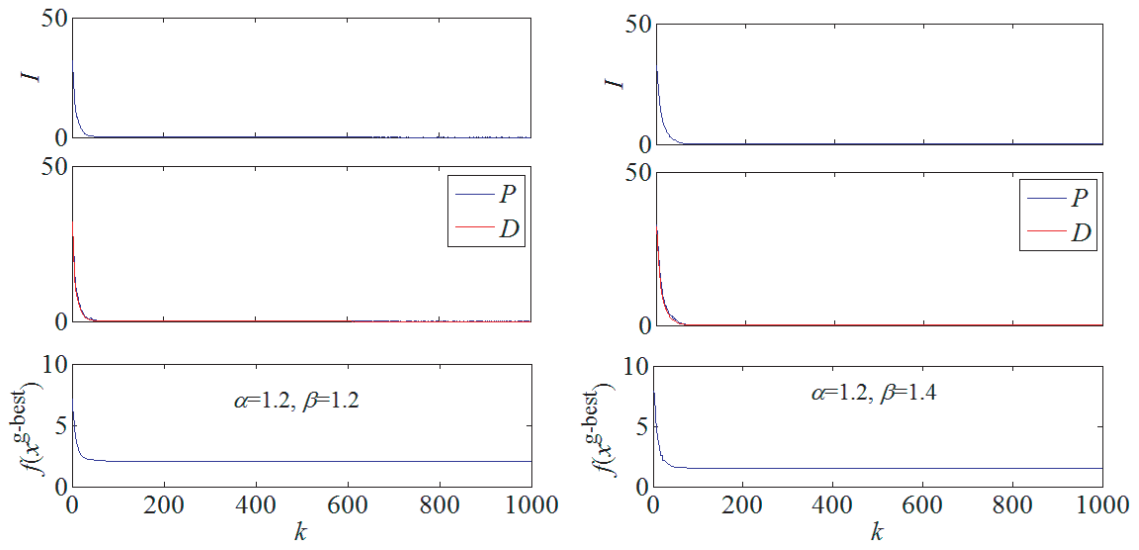
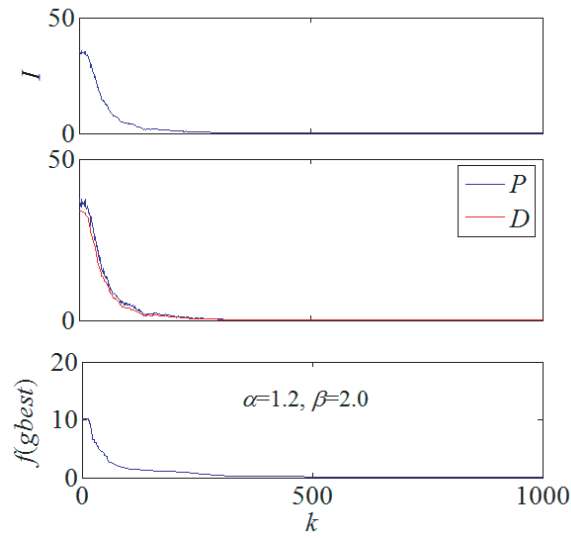
関数	N	PSO	DE	FA	ABC	PM		
						$\beta = 1.2$	$\beta = 1.4$	$\beta = 2.0$
1. Parabola	10	0.00 ■	0.00 ◆	0.01	0.01 ▲	0.10	0.03	0.00 ●
	50	31.4	15.9 ■	42.9	3.49 ●	22.5	14.9 ◆	18.1 ▲
	100	221	161	161 ▲	58.6 ●	100 ■	80.7 ◆	425
	300	1630	1759	492 ●	947 ▲	613 ◆	671 ■	2275
2. Rosenbrock	10	8.18 ◆	8.63 ▲	11.1	6.02 ●	15.3	9.78	8.47 ■
	50	1375	998	224 ●	297 ◆	469 ▲	414 ■	2145
	100	18155	10212	1380 ●	1866 ◆	2447 ▲	1970 ■	32723
	300	112538	113003	19714 ●	34778 ■	20069 ◆	90172 ▲	115963
3. 2^N minima	10	-717	-782 ●	-665	-776 ◆	-727	-735 ▲	-757 ■
	50	-2502	-2295	-3062 ◆	-3539 ●	-2841 ▲	-3049 ■	-1414
	100	-2592	-2578	-5526 ◆	-6448 ●	-4792 ■	-3107 ▲	-1977
	300	-5034 ■	-4520 ▲	-13525 ◆	-14035 ●	-4520 ▲	-4520 ▲	-4520 ▲
4. Rastrigin	10	18.1 ▲	23.1	56.2	2.29 ●	9.47 ◆	14.6 ■	33.7
	50	428	435	388 ▲	89.9 ●	178 ◆	231 ■	512
	100	1144	1156	877 ▲	387 ●	490 ◆	696 ■	1469
	300	4457	4683	2785 ■	2696 ◆	2493 ●	3088 ▲	5165
5. Schwefel	10	0.02 ■	0.02 ◆	4.40	0.03 ▲	2.51	0.37	0.00 ●
	50	14112	7734 ■	20577	2426 ●	12282	9064 ▲	7671 ◆
	100	474080	342511 ▲	347566	168767 ●	245606 ■	186178 ◆	589659
	300	40188152	38039143	11016695 ●	27540850 ▲	14537627 ◆	15574649 ■	64363154
6. Levy	10	0.11	0.00 ●	17.1	0.03 ■	0.16	0.07 ▲	0.02 ◆
	50	29.5	21.7 ▲	33.0	2.38 ●	4.75	3.54 ◆	31.3
	100	59.7	57.4	40.2 ▲	12.0 ■	11.4 ●	11.7 ◆	102
	300	110	122	40.8 ◆	55.7 ▲	31.0 ●	41.5 ■	151
7. Ackley	10	0.15 ■	0.02 ●	6.57	0.70	0.69	0.33 ▲	0.03 ◆
	50	4.97	3.90 ■	7.63	7.00	3.68 ◆	3.21 ●	3.94 ▲
	100	7.12 ▲	6.33 ■	7.63	8.67	4.94 ◆	4.56 ●	8.18
	300	9.09 ▲	9.37	6.85 ■	9.80	6.56 ●	6.66 ◆	10.2
8. Griewank	10	0.22 ▲	0.35	0.15 ■	0.46	0.11 ◆	0.11 ●	0.34
	50	1.78	1.39 ◆	1.56	3.15	1.54 ▲	1.39 ●	1.43 ■
	100	6.49 ▲	5.16 ■	18.3	9.12	3.39 ◆	3.01 ●	11.4
	300	40.8 ■	45.0	57.9	41.1 ▲	16.5 ●	17.5 ◆	57.9
9. Alpine	10	0.40	0.22 ▲	1.21	0.30	0.04 ◆	0.01 ●	0.09 ■
	50	51.0	50.9	26.2 ▲	7.40 ●	13.3 ■	9.69 ◆	68.5
	100	156	163	58.6 ▲	39.0 ●	51.5 ◆	52.8 ■	221
	300	669	732	141 ●	352 ■	312 ◆	394 ▲	822

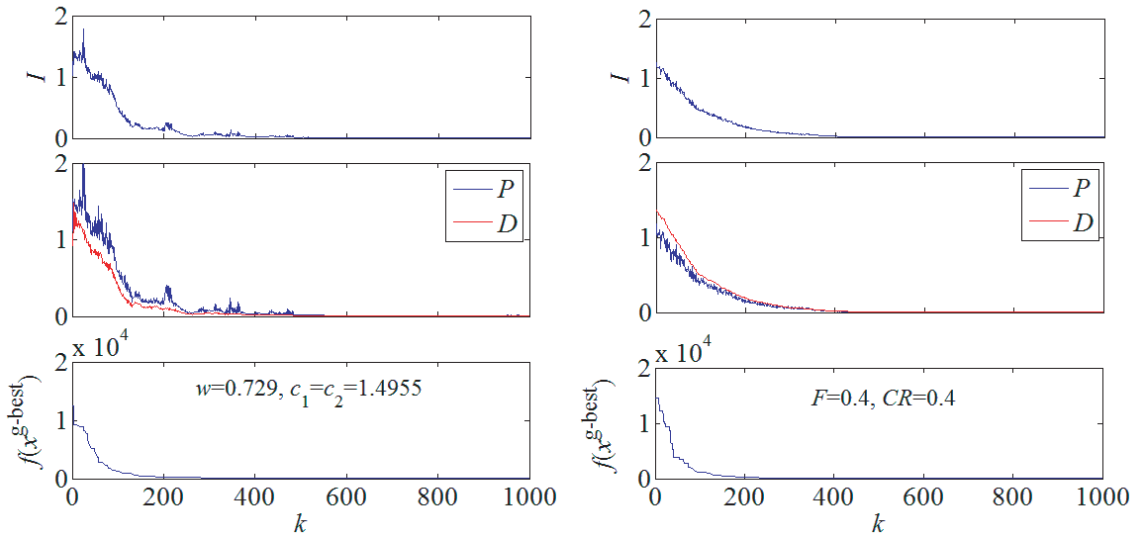
表 6.2 : 数値実験結果 (提案手法, 評価回数 $T = 20000$)

関数	N	PSO	DE	FA	ABC	PM		
						$\beta = 1.2$	$\beta = 1.4$	$\beta = 2.0$
1. Parabola	10	0.00 ◆	0.00 ▲	0.00	0.00 ■	0.06	0.02	0.00 ●
	50	0.03 ▲	0.01 ■	0.36	0.00 ◆	23.6	14.3	0.00 ●
	100	9.87	0.18 ■	1.78 ▲	0.00 ●	91.4	80.1	0.18 ◆
	300	495 ▲	21.3 ■	19.0 ◆	1.52 ●	628	621	2193
2. Rosenbrock	10	1.71 ◆	6.25	3.53 ■	0.98 ●	14.3	10.1	5.89 ▲
	50	79.0 ■	59.8 ◆	84.1 ▲	101	528	390	56.1 ●
	100	537 ▲	153 ●	289 ■	245 ◆	2161	1830	1112
	300	111566	3559 ■	3075 ◆	1241 ●	19140 ▲	19592	115963
3. 2^N minima	10	-722	-782 ◆	-666	-783 ●	-737	-738 ▲	-758 ■
	50	-3361 ▲	-3839 ◆	-3265	-3914 ●	-2878	-3112	-3539 ■
	100	-3835	-7055 ◆	-6460 ■	-7782 ●	-4881	-5441 ▲	-1990
	300	-5044 ▲	-4520	-18987 ◆	-22053 ●	-5304 ■	-4520	-4520
4. Rastrigin	10	8.78	0.57 ◆	55.5	0.00 ●	5.52	5.20 ▲	4.88 ■
	50	162	257	354	0.81 ●	133 ■	102 ◆	149 ▲
	100	483 ▲	728	778	12.8 ●	441 ■	345 ◆	678
	300	3247	3037	2740 ▲	265 ●	2399 ■	2202 ◆	5061
5. Schwefel	10	0.00 ◆	0.00 ▲	0.16	0.00 ■	2.42	0.60	0.00 ●
	50	23.1 ▲	12.3 ■	326	0.02 ◆	12643	7124	0.00 ●
	100	20326	662 ■	6250 ▲	12.0 ●	240955	174274	140 ◆
	300	6903009 ▲	370917 ◆	581117 ■	52882 ●	15100250	14179676	18563442
6. Levy	10	0.01 ▲	0.00 ◆	9.60	0.00 ●	0.24	0.07	0.01 ■
	50	3.14	0.00 ◆	24.6	0.00 ●	5.11	3.08 ▲	0.03 ■
	100	12.4	0.03 ◆	32.5	0.00 ●	11.3	8.30 ▲	1.60 ■
	300	45.9	17.0 ◆	38.0	0.36 ●	30.1 ▲	27.9 ■	139
7. Ackley	10	0.22 ▲	0.00 ◆	6.62	0.00 ●	0.84	0.27	0.02 ■
	50	3.11 ▲	0.02 ●	6.46	0.17 ◆	3.61	3.17	0.41 ■
	100	4.77	0.19 ●	4.35 ▲	2.14 ■	4.86	4.44	1.57 ◆
	300	7.21	2.19 ●	3.90 ◆	6.66	6.47 ■	6.48 ▲	10.1
8. Griewank	10	0.10	0.00 ●	0.11	0.01 ◆	0.10	0.04 ■	0.05 ▲
	50	0.11	0.01 ●	0.02 ◆	0.11 ▲	1.54	1.35	0.03 ■
	100	1.23 ▲	0.09 ●	18.1	0.43 ■	3.57	2.92	0.13 ◆
	300	9.60 ■	1.51 ◆	57.7	1.23 ●	17.0	16.7 ▲	56.8
9. Alpine	10	0.00 ■	0.00 ●	0.52	0.00 ▲	0.07	0.00	0.00 ◆
	50	2.04 ▲	0.01 ◆	11.4	0.10 ■	11.7	7.10	0.00 ●
	100	27.3 ▲	0.16 ●	27.0	0.53 ◆	50.4	34.9	21.7 ■
	300	315	237 ■	70.6 ◆	13.5 ●	300	261 ▲	774

(a) $\beta = 1.2$ (Rosenbrock)(b) $\beta = 1.4$ (Rosenbrock)(c) $\beta = 2.0$ (Rosenbrock)図 6.5 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (提案手法, Rosenbrock, $N = 50$)

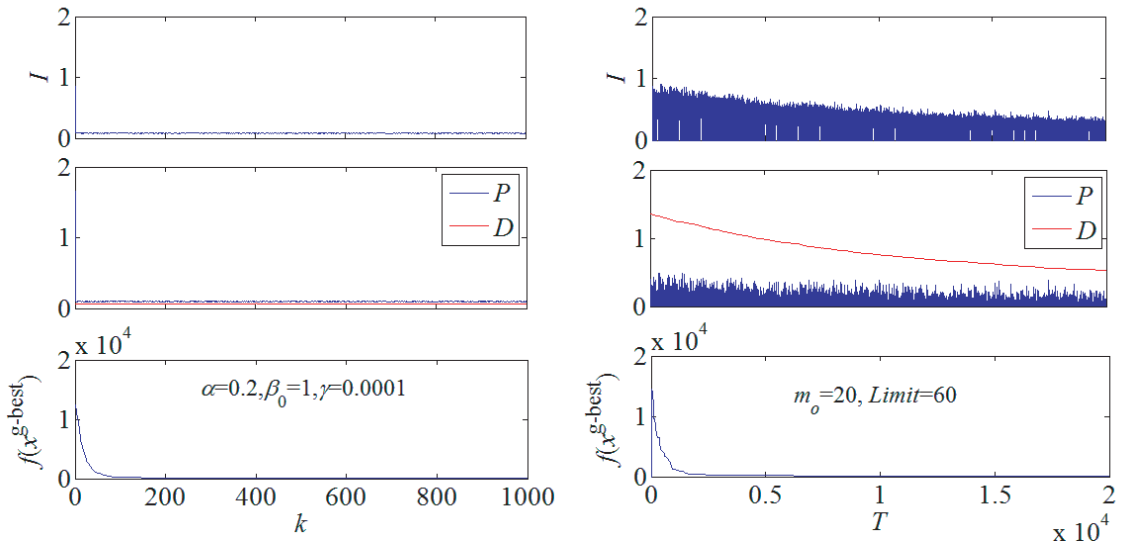
(a) $\beta = 1.2$ (Rastrigin)(b) $\beta = 1.4$ (Rastrigin)(c) $\beta = 2.0$ (Rastrigin)図 6.6 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (提案手法, Rastrigin, $N = 50$)

(a) $\beta = 1.2$ (Griewank)(b) $\beta = 1.4$ (Griewank)(c) $\beta = 2.0$ (Griewank)図 6.7 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (提案手法, Griewank, $N = 50$)



(a) Particle Swarm Optimization

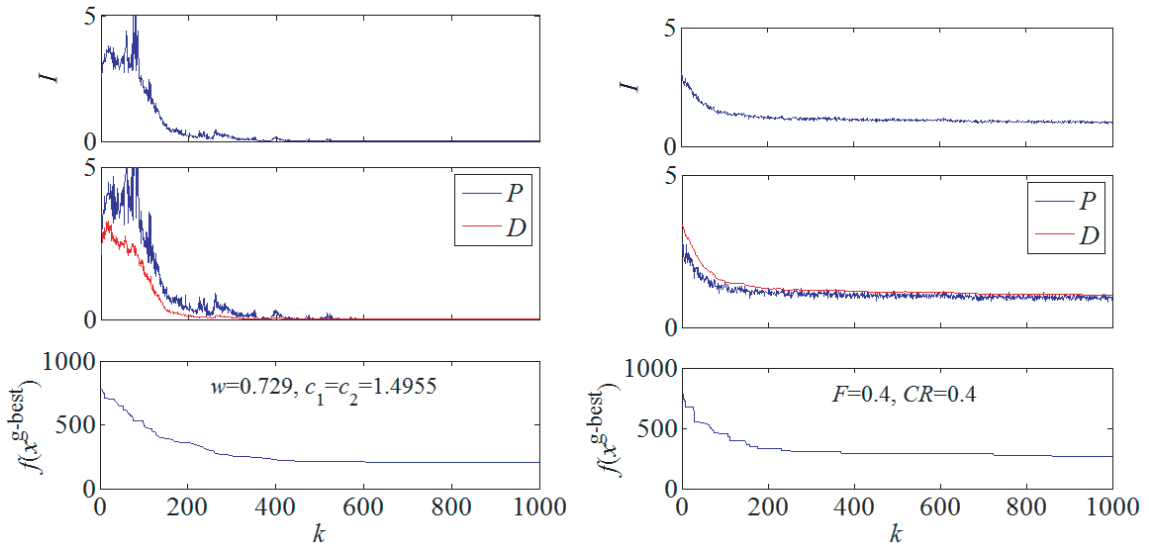
(b) Differential Evolution



(c) Firefly Algorithm

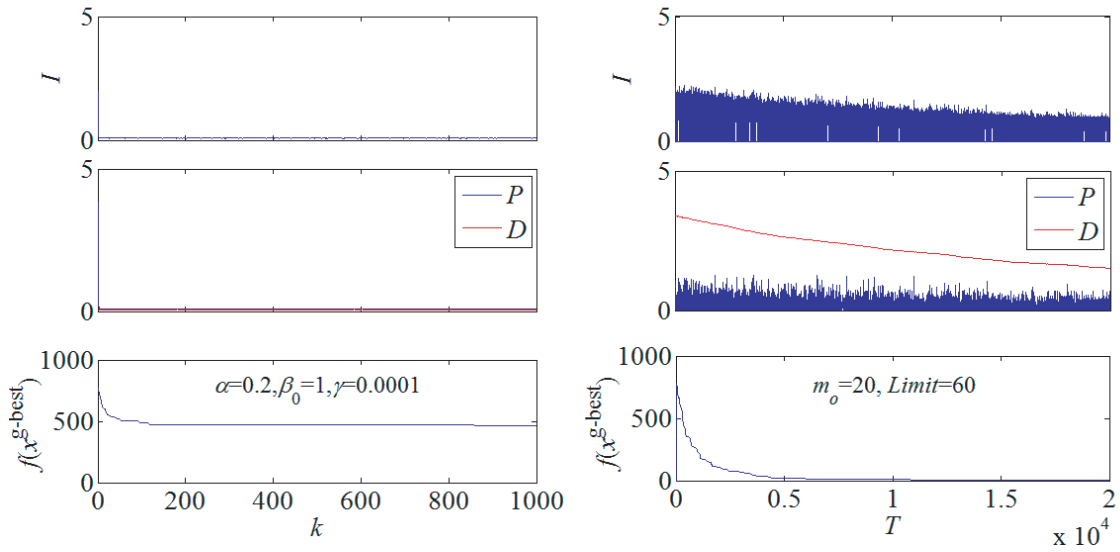
(d) Artificial Bee Colony Algorithm

図 6.8 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (PSO, DE, FA, ABC, Rosenbrock, 反復回数 $k = 1000$ (評価回数 $T = 20000$))



(a) Particle Swarm Optimization

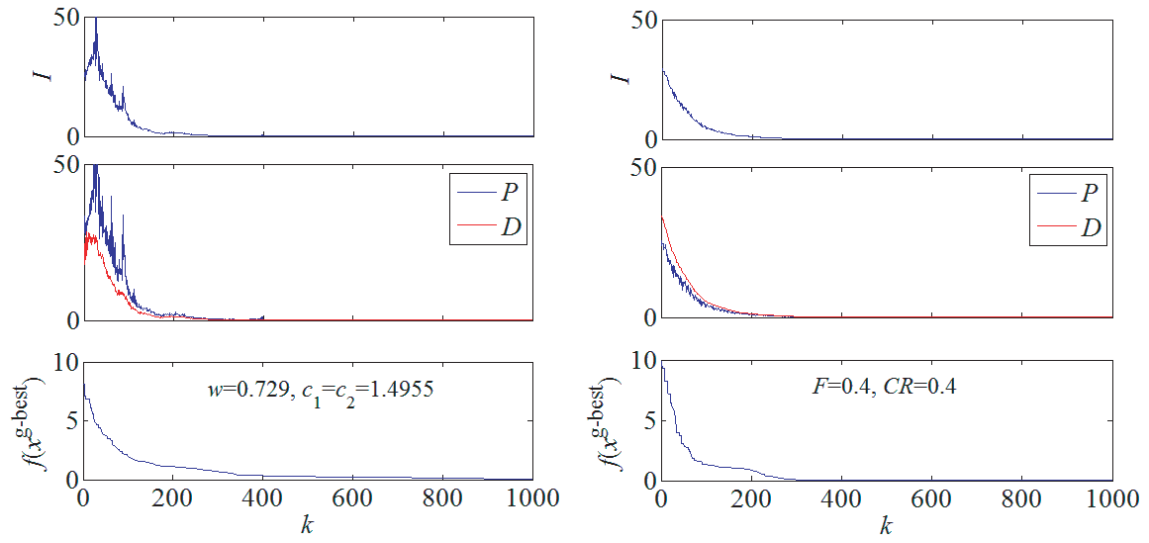
(b) Differential Evolution



(c) Firefly Algorithm

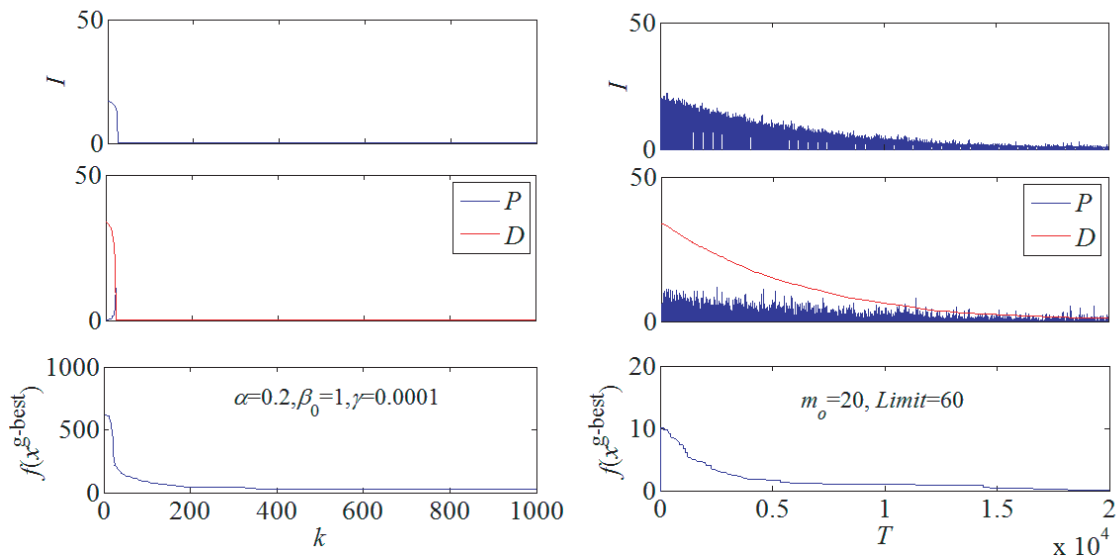
(d) Artificial Bee Colony Algorithm

図 6.9 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (PSO, DE, FA, ABC, Rastrigin, 反復回数 $k = 1000$ (評価回数 $T = 20000$))



(a) Particle Swarm Optimization

(b) Differential Evolution



(c) Firefly Algorithm

(d) Artificial Bee Colony Algorithm

図 6.10 : 評価指標 I と最良解の目的関数値 $f(x^{g-best})$ の推移 (PSO, DE, FA, ABC, Griewank, 反復回数 $k = 1000$ (評価回数 $T = 20000$))

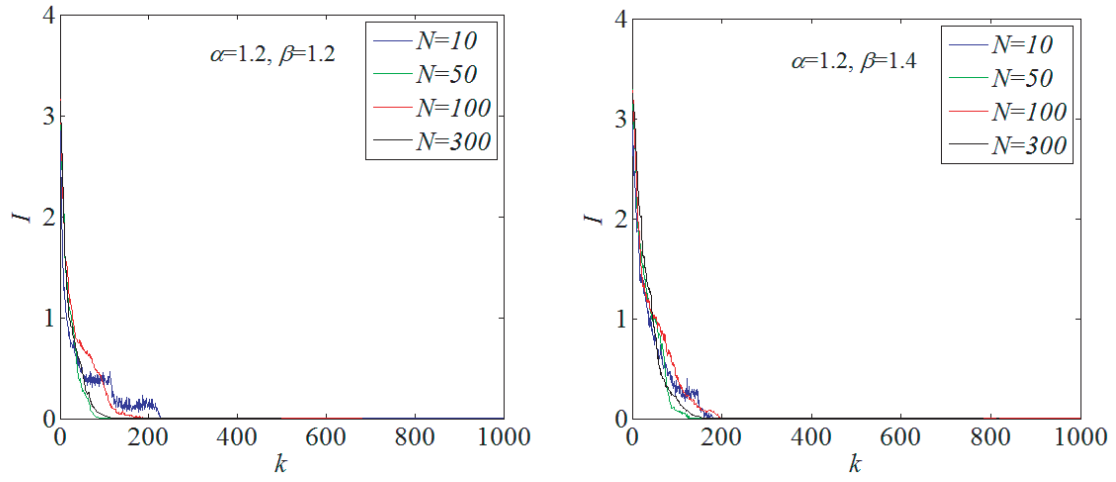
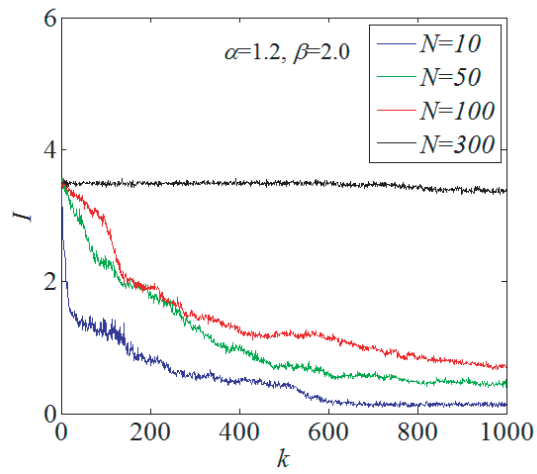
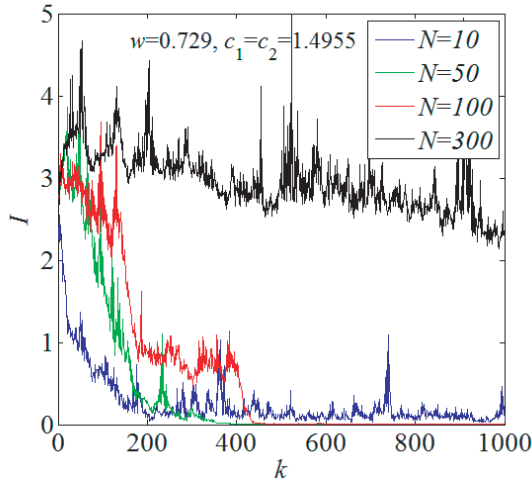
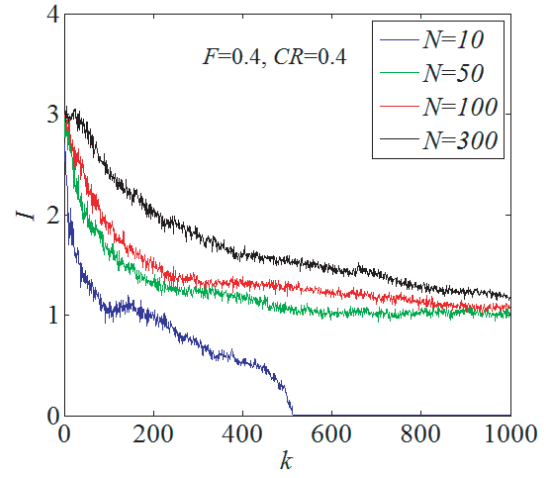
(a) $\beta = 1.2$ (Rastrigin)(b) $\beta = 1.4$ (Rastrigin)(c) $\beta = 2.0$ (Rastrigin)

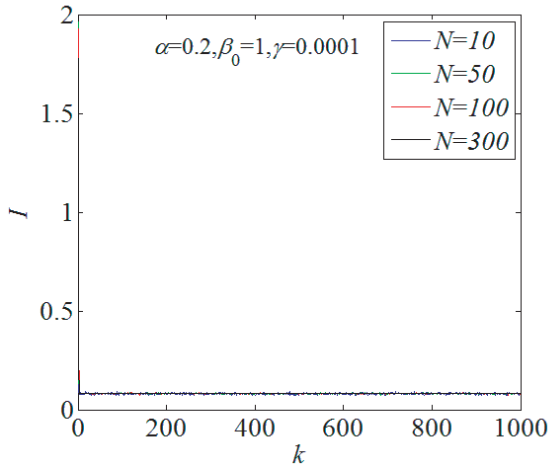
図 6.11 : 各次元数における評価指標 I の推移 (提案手法, Rastrigin, 反復回数 $k = 1000$ (評価回数 $T = 20000$))



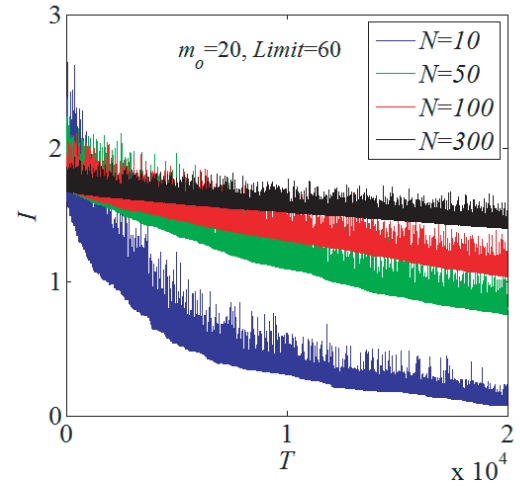
(a) Particle Swarm Optimization



(b) Differential Evolution



(c) Firefly Algorithm



(d) Artificial Bee Colony Algorithm

図 6.12 : 各次元数における評価指標 I の推移 (PSO , DE , FA , ABC , Rastrigin , 反復回数 $k = 1000$ (評価回数 $T = 20000$))

7

適応化のフレームワーク に基づく適応型 メタヒューリスティクス

6章では，適応化のフレームワークに従い，高い適応能力を素質として備えた新たなメタヒューリスティクスを構築した。本章では，6章の提案手法を基礎とし適応化することで，適応型メタヒューリスティクスを構築する。

7.1 探索状態の評価と制御に基づく適応型メタヒューリスティクス

適応化のフレームワークでは，高い適応能力を素質として備えた新たなメタヒューリスティクスを適応化することで，さらに優れた適応能力・探索性能を有する適応型メタヒューリスティクスの構築が期待できる。6章では，適応化のフレームワークの適応化設計論に基づき，高い適応能力を素質として備えた新たなメタヒューリスティクスを構築した。数値実験の結果や評価指標の推移から，提案手法は探索戦略をある程度実現しながら，他のメタヒューリスティクスと同等の探索性能を有することを確認した。6章の数値実験結果から，提案手法は素質として一定以上の適応能力を備えていることを示した。本章では，6章の提案手法を適応化することで，さらに優れた適応型メタヒューリスティクスを構築する。

図 6.2，図 6.3，図 6.4 から，提案手法は，パラメータ β の値により，多様化・集中化（探索状態）の調整能力が異なることを確認した。この検証から，探索条件や探索状態に応じて β を調整し，探索過程で多様化・集中化を適切に実現することで，提案手法の探索性能・

適応能力の向上が期待できる。適応型メタヒューリスティクスの構築のためには、5.4.1項で述べたように、探索状態の評価指標と制御式を与える必要がある。すでに、6.2節では、数値実験を通じて、評価指標が提案手法の多様化・集中化を定量的に評価できることを確認した。したがって、探索過程における提案手法の多様化・集中化の実現状態を評価しながら、事前に設定した目標値に評価指標が追従するように β を調整することで、提案手法の多様化・集中化を確実に制御することが可能となる。本論文では、この手法を「適応型提案手法」、6章で提案した手法を「オリジナル提案手法」と呼ぶことにし、区別する。

具体的な β の調整則を式(7.1)に示す。適応型CSと同様に、多様化・集中化の評価指標として、式(5.4)の評価指標 I を使用する。事前に評価指標の目標値スケジュール $I_{\text{target}}(k)$ ($k = 1, 2, \dots, k_{\text{max}}$)を与えておき、各反復回数 k における評価指標 I の値と比較し、 I が $I_{\text{target}}(k)$ に追従するように調整幅 $\Delta\beta > 0$ で β を調整する。

$$\beta := \begin{cases} \max \{\beta - \Delta\beta, \beta_{\min}\}, & I \geq I_{\text{target}}(k) \\ \min \{\beta + \Delta\beta, \beta_{\max}\}, & \text{otherwise} \end{cases} \quad (7.1)$$

この調整則では、目標値スケジュール I_{target} を適切に設定することで、各探索段階における理想の探索状態へ制御することができる。3.7節で述べたように、メタヒューリスティクスでは「探索序盤では多様化の実現、探索終盤では集中化の実現を目指す」戦略が有効であることが知られている。本論文では目標値スケジュール I_{target} を探索序盤では大きな値に、探索終盤では小さな値に設定することで、多様化・集中化を適切なバランスで制御することが期待できる。図7.1に、本論文で使用する目標値 I_{target} の線形および指数減少スケジュールを示し、式(7.2)、式(7.3)にそれぞれのスケジュールの式を示す。

$$I_{\text{target}}(k) = \max \left\{ 0, I_{\text{start}} \left(1 - \frac{k}{k_{\text{end}}} \right) \right\} \quad (7.2)$$

$$I_{\text{target}}(k) = I_{\text{start}} \left(\frac{I_{\text{end}}}{I_{\text{start}}} \right)^{\frac{k}{k_{\text{max}}}} \quad (7.3)$$

なお、適応型提案手法における β の調整則では、新たに下限値 β_{\min} 、上限値 β_{\max} 、調整幅 $\Delta\beta$ 、 k_{end} 、 I_{start} 、 I_{end} というパラメータが導入されている*。

* 著者らの数値実験に基づいた、適応型提案手法のパラメータ推奨値は、 $k_{\text{end}} = 0.95 \cdot k_{\text{max}}$ 、 $I_{\text{start}} = 0.2 \cdot x_{\text{width}}$ 、 $I_{\text{end}} = 0.0001 \cdot x_{\text{width}}$ 、 $x_{\text{width}} = |x_{\text{max}} - x_{\text{min}}|$ ($x_{\text{max}} = \max\{x_n^i | n = 1, \dots, N; i = 1, \dots, m; k = 1\}$ 、 $x_{\text{min}} = \min\{x_n^i | n = 1, \dots, N; i = 1, \dots, m; k = 1\}$)である。

以下に、本論文で提案した適応化のフレームワークに基づく適応型メタヒューリスティクス（適応型提案手法）のアルゴリズムを示す。アルゴリズムの終了条件を $k = k_{\max}$ とする。評価回数は $T = m(k_{\max} + 1)$ となる。探索点の初期位置を初期配置領域 $IS = [a, c]^N \subset \mathbb{R}^N$ 内に与える。

【適応型提案手法のアルゴリズム】

Step 0: [準備]

探索点数 m , パラメータ $\alpha \geq 0$, β の下限値 $\beta_{\min} \in [0, \beta_{\max})$, 上限値 $\beta_{\max} \in (\beta_{\min}, \infty]$, 調整幅 $\Delta\beta \in (0, \beta_{\max} - \beta_{\min}]$, 最大反復回数 k_{\max} を定め、反復回数を $k = 1$ とする。評価指標の目標値 $I_{\text{target}}(k)$ ($k = 1, 2, \dots, k_{\max}$) を与える。

Step 1: [初期化]

探索点の初期位置 $x^i(k)$ ($i = 1, 2, \dots, m$) を初期配置領域 IS 内にランダムに与える。

Step 2: [近傍の生成]

各探索点 $x^i(k)$ について、優良探索点群 $Better^i$, 近傍解 $\hat{x}^i(k)$ を

$$Better^i = \{x^\ell(k) \mid f(x^\ell(k)) < f(x^i(k)); \ell = 1, \dots, m\}$$

$$\hat{x}^i(k) = \begin{cases} x^i(k) + \alpha R u^{better} + \beta \phi u^{random}, & Better^i \neq \emptyset \\ x^i(k) + \beta \phi u^{random}, & \text{otherwise} \end{cases}$$

より、生成する。

ただし、 $u^{better} = x^{i,better}(k) - x^i(k)$, $u^{random} = x^r(k) - x^i(k)$, $x^{i,better}(k) \in Better^i$ は優良探索点群 $Better^i$ からランダムに選ばれた探索点、 \emptyset は空集合、 r は整数値一様分布 $\mathcal{U}_{\mathbb{Z}}(1, m)$ に従う乱数、 $R, \phi \in \mathbb{R}^{N \times N}$ は対角行列であり、 R の対角要素は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従う乱数、 ϕ の対角要素は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(-0.5, 0.5)$ に従う乱数である。

Step 3: [探索点の更新]

各探索点 $x^i(k)$, β について、

$$v^i(k) = \begin{cases} \hat{x}^i(k) - x^i(k), & f(\hat{x}^i(k)) < f(x^i(k)) \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbf{x}^i(k+1) = \mathbf{x}^i(k) + \mathbf{v}^i(k)$$

$$\beta := \begin{cases} \max \{\beta - \Delta\beta, \beta_{\min}\}, & I \geq I_{\text{target}}(k) \\ \min \{\beta + \Delta\beta, \beta_{\max}\}, & \text{otherwise} \end{cases}$$

より，更新する。

Step 4: [終了判定]

$k = k_{\max}$ ならば，探索を終了する。さもなければ， $k := k + 1$ とし，**Step 2** へ戻る。

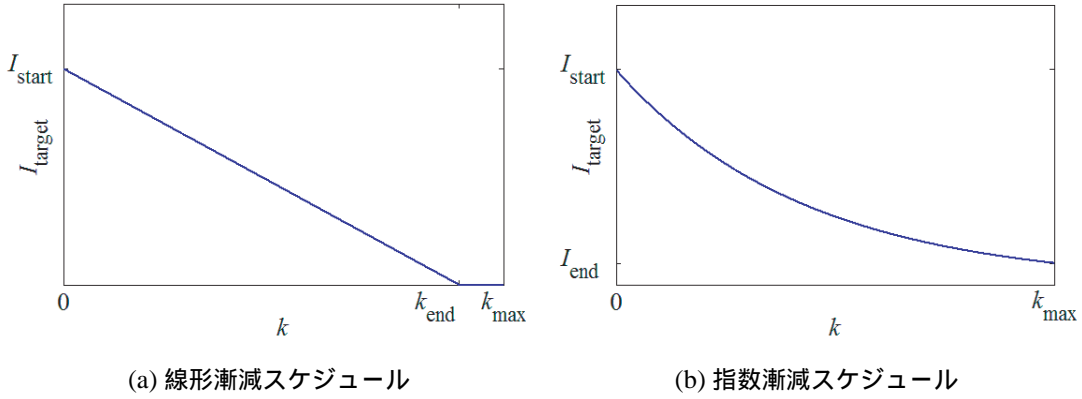


図 7.1：評価指標の目標値スケジュール $I_{\text{target}}(k)$

7.2 数値実験による検証

7.2.1 数値実験条件

9種類の典型的なベンチマーク関数を用いた数値実験を行い，オリジナル提案手法と比較することで，適応型提案手法の探索性能を検証する。数値実験には，表 5.1 のベンチマーク関数を用いる。

共通条件として，探索点数 $m = 20$ ，次元数 $N = 10, 50, 100, 300$ ，パラメータ $\alpha = 1.2$ ，関数毎に共通の初期配置領域 IS を用い，終了条件を反復回数 $k_{\max} = 100, 1000$ とする。また，オリジナル提案手法では $\beta = 1.0, 1.2, \dots, 3.0$ の 11 通りの値を用い，適応型提案手法では下限値 $\beta_{\min} = 1.0$ ，上限値 $\beta_{\max} = 3.0$ ，調整幅 $\Delta\beta = 0.2$ を用いる。 β 以外のパラメータ

は、様々な問題を対象とした数値実験を通じて平均的に優良な結果が得られた値に設定する。50 試行の平均値を実験結果とする。

本実験では、適応型提案手法の探索性能を、オリジナル提案手法の 11 通りの β の値における結果と比較することで評価する。オリジナル提案手法では、11 通りの結果の中で、最も優れている結果（最良値）を 1st、3 番目に優れている結果（第一四分位数）を 3rd、6 番目に優れている結果（中央値）を 6th、11 通りの結果の平均値を Mean と表現する。手法や問題に対する知識や経験を有していない使用者は、 β をランダムに決定することが考えられる。そのため、Mean はこのような使用者による平均的な結果と想定できる。また、オリジナル提案手法の結果には設定した β の値も示す。

適応型提案手法では、 I_{target} を図 7.1 の線形および指数減少スケジュールに設定した条件で得た結果をそれぞれ Lin、Exp と表現する。また、オリジナル提案手法の 11 通りと適応型提案手法を合わせた 12 通りの結果における、適応型提案手法の順位を Rank と表現する。Rank の値は、適応型提案手法がオリジナル提案手法のどのような順位より優れているかを示す。

7.2.2 実験結果・考察

表 7.1、表 7.2 に数値実験結果を示す。表 7.1、表 7.2 では、適応型提案手法が、オリジナル提案手法の Mean よりも優れている場合は ▲ を、6th よりも優れている場合は ■ を、3rd よりも優れている場合は ◆ を、1st よりも優れている場合は ● を付す。

表 7.1、表 7.2 より、適応型提案手法はほぼ全ての条件において、Mean、6th よりも優れていることが確認できる。この結果は、対象問題の種類、次元数、終了条件などに関わらず、適応型提案手法では、オリジナル提案手法でパラメータをランダムに設定した場合よりも得られる結果の期待値が高いことを表す。さらに、適応型 CS はいくつかの実験条件（特に $k_{\text{max}} = 1000$ ）において、3rd や 1st よりも優れていることが確認できる。以上の結果から、適応型提案手法がオリジナル提案手法よりも全体的に探索性能が優れているといえる。また、5 章で構築した適応型 CS よりも、本章の適応型提案手法の方が、オリジナルの手法からの探索性能の向上が大きい。これは、適応化のフレームワークに従って構築したオリジナル提案手法は、素質として適応能力が高いために、オリジナル提案手法に対する

適応化の効果が高かったことが考えられる。

また，図 7.2 に Rosenbrock's Saddle Function，Rastrigin Function，Griewank Function ($N = 50$) を対象とした場合の β ，多様化・集中化の評価指標 I ，摂動の評価指標 P ，探索点分布の評価指標 D ，最良解の評価値 $f(x^{g-best})$ の推移を示す。図 7.2 より，適応型提案手法では，探索過程で I が目標値 I_{target} に追従することで徐々に漸減すると同時に， $f(x^{g-best})$ が改善する様子を確認できる。図 6.2，図 6.3，図 6.4 より，オリジナル提案手法では，探索過程で多様化から集中化へ実現状態を推移することができるが， β の値，対象問題の種類，初期配置領域 IS によって，その推移するスピードが大きく異なることを確認した。これは，パラメータ調整無しで探索戦略の実現が可能なことから，素質として適応能力を有しているが，パラメータや実験条件によって，探索過程で適応能力を十分に発揮できない場合が存在することを示している。これに対して，適応型提案手法は，オリジナル提案手法を基礎とすると同時に，探索過程で多様化・集中化（探索状態）を制御することで，確実に探索戦略を実現するため，素質として適応能力が高い上に，その適応能力が飛躍的に向上したといえる。

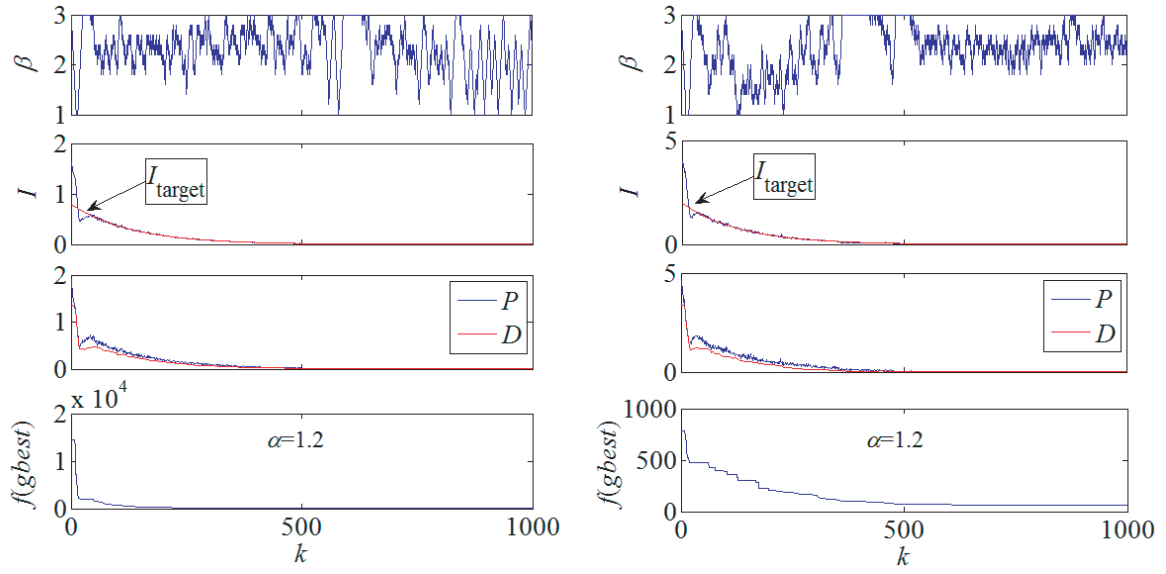
以上から，本論文で構築した適応化のフレームワークは，既存の手法に対して適応能力を向上させることが可能であると同時に，素質として高い適応能力を備えた手法を設計することが可能である。さらに，適応化のフレームワークに従って新たに設計・開発した手法は，適応型メタヒューリスティクスとしてさらに優れた探索性能・適応能力を発揮することが期待できる。

表 7.1: 数値実験結果 (適応型提案手法, 反復回数 $k = 100$)

関数	N	オリジナル提案手法 (11 種類 β)						適応型提案手法				
		1st	(β)	3rd	(β)	6th	(β)	Mean	Lin	Rank	Exp	Rank
1. Parabola	10	0.00	(2.0)	0.00	(2.2)	0.01	(2.6)	0.06	0.00 ■	6	0.00 ■	4
	50	7.60	(1.8)	14.9	(1.4)	29.0	(1.0)	116	8.17 ◆	2	26.2 ■	6
	100	67.8	(1.6)	100	(1.2)	425	(2.0)	393	58.4 ●	1	108 ■	6
	300	565	(1.0)	671	(1.4)	2275	(2.0)	1538	495 ●	1	591 ◆	2
2. Rosenbrock	10	7.48	(2.2)	8.47	(2.0)	9.78	(1.4)	12.1	7.49 ◆	2	11.1 ▲	8
	50	319	(1.6)	414	(1.4)	2145	(2.0)	5906	297 ●	1	540 ■	6
	100	1970	(1.4)	2447	(1.2)	32723	(2.0)	21256	1460 ●	1	2283 ◆	2
	300	17432	(1.0)	90172	(1.4)	115963	(2.0)	95943	14833 ●	1	16936 ●	1
3. 2^N minima	10	-757	(2.0)	-754	(2.2)	-735	(1.4)	-721	-749 ■	4	-724 ▲	8
	50	-3049	(1.4)	-2682	(1.6)	-1414	(2.0)	-1886	-3097 ●	1	-2650 ■	4
	100	-4792	(1.2)	-3107	(1.4)	-1977	(2.0)	-2559	-5166 ●	1	-4437 ◆	3
	300	-6132	(1.0)	-4520	(1.4)	-4520	(2.0)	-4667	-5918 ◆	2	-5719 ◆	2
4. Rastrigin	10	9.03	(1.0)	14.6	(1.4)	33.7	(2.0)	31.2	13.9 ◆	3	11.4 ◆	3
	50	166	(1.0)	231	(1.4)	512	(2.0)	495	217 ◆	3	172 ◆	2
	100	490	(1.8)	696	(1.4)	1469	(2.0)	1221	593 ◆	3	535 ◆	3
	300	2493	(1.2)	3088	(1.4)	5165	(2.0)	4401	2662 ◆	3	2582 ◆	3
5. Schwefel	10	0.00	(2.0)	0.00	(2.2)	0.09	(1.6)	1.30	0.04 ■	5	0.13 ▲	7
	50	3807	(1.8)	7671	(2.0)	17695	(1.0)	66559	3884 ◆	2	18682 ▲	7
	100	149871	(1.6)	212083	(1.8)	589659	(2.0)	1047926	132624 ●	1	291219 ■	6
	300	14144176	(1.0)	15574649	(1.4)	64363154	(2.0)	45579151	11845238 ●	1	15101423 ◆	3
6. Levy	10	0.00	(1.6)	0.03	(1.8)	0.07	(1.4)	0.39	0.00 ●	1	0.14 ▲	7
	50	3.41	(1.6)	4.75	(1.2)	31.3	(2.0)	47.2	3.58 ◆	3	7.08 ■	5
	100	11.4	(1.2)	13.9	(1.0)	102	(2.0)	78.8	11.3 ●	1	13.5 ◆	3
	300	29.7	(1.0)	41.5	(1.4)	151	(2.0)	114	31.0 ◆	2	32.0 ◆	3
7. Ackley	10	0.03	(2.4)	0.03	(2.0)	0.32	(1.6)	0.36	0.08 ■	4	0.19 ■	6
	50	2.73	(1.6)	3.21	(1.4)	4.05	(1.0)	5.74	2.83 ◆	3	3.86 ■	5
	100	4.53	(1.6)	4.94	(1.2)	8.18	(2.0)	7.47	4.32 ●	1	5.13 ■	5
	300	6.34	(1.0)	6.66	(1.4)	10.2	(2.0)	8.9	6.19 ●	1	6.45 ◆	2
8. Griewank	10	0.11	(1.4)	0.14	(1.6)	0.34	(2.0)	0.38	0.42	7	0.11 ◆	2
	50	1.18	(1.8)	1.39	(1.4)	1.74	(1.0)	3.87	1.20 ◆	2	1.66 ■	6
	100	2.66	(1.6)	3.39	(1.2)	11.4	(2.0)	10.8	2.41 ●	1	3.71 ■	6
	300	15.3	(1.0)	17.5	(1.4)	57.9	(2.0)	43.5	13.6 ●	1	16.0 ◆	2
9. Alpine	10	0.01	(1.6)	0.01	(1.8)	0.11	(1.0)	1.51	0.02 ■	4	0.02 ■	4
	50	9.69	(1.4)	0.01	(1.8)	0.11	(1.0)	1.51	0.02 ■	4	0.02 ■	4
	100	51.5	(1.2)	62.6	(1.0)	221	(2.0)	173	54.7 ◆	3	64.4 ■	4
	300	309	(1.0)	394	(1.4)	822	(2.0)	673	316 ◆	3	326 ◆	3

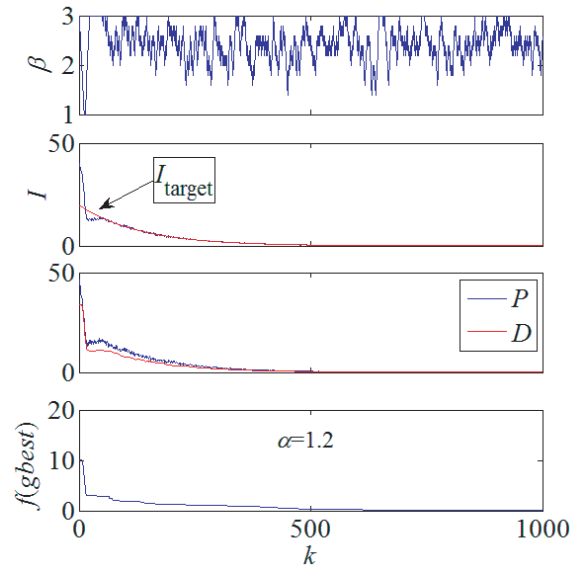
表 7.2 : 数値実験結果 (適応型提案手法 , 反復回数 $k = 1000$)

関数	N	オリジナル提案手法 (11 種類 β)						適応型提案手法				
		1st	(β)	3rd	(β)	6th	(β)	Mean	Lin	Rank	Exp	Rank
1. Parabola	10	0.00	(2.0)	0.00	(1.8)	0.00	(2.8)	0.03	0.00 ▲	7	0.00 ▲	7
	50	0.00	(2.0)	0.00	(2.4)	4.26	(1.6)	35.0	0.08 ■	5	0.00 ◆	3
	100	0.18	(2.0)	6.15	(2.2)	91.4	(1.2)	259	3.16 ◆	3	0.25 ◆	2
	300	243	(1.8)	547	(1.0)	2193	(2.0)	1461	184 ●	1	142 ●	1
2. Rosenbrock	10	3.81	(2.6)	3.96	(3.0)	5.89	(2.0)	7.88	4.15 ■	4	4.64 ■	5
	50	56.1	(2.0)	77.3	(1.8)	390	(1.4)	2545	61.9 ◆	3	49.0 ●	1
	100	378	(1.8)	1112	(1.2)	2383	(1.0)	15792	316 ●	1	206 ●	1
	300	16911	(1.0)	19592	(1.4)	115963	(2.0)	87982	5935 ●	1	4256 ●	1
3. 2^N minima	10	-775	(3.0)	-767	(2.6)	-758	(2.0)	-753	-762 ■	5	-752 ■	7
	50	-3539	(2.0)	-3364	(1.6)	-2878	(1.2)	-2528	-3594 ●	1	-3511 ◆	2
	100	-5981	(1.6)	-4881	(1.2)	-1990	(2.0)	-3191	-6834 ●	1	-6856 ●	1
	300	-10640	(1.0)	-4520	(1.4)	-4520	(2.0)	-5148	-16440 ●	1	-16270 ●	1
4. Rastrigin	10	4.55	(1.6)	4.88	(2.0)	6.03	(2.2)	9.37	3.98 ●	1	4.03 ●	1
	50	77.5	(1.8)	102	(1.4)	163	(1.0)	300	66.5 ●	1	64.7 ●	1
	100	225	(1.8)	345	(1.4)	678	(2.0)	906	250 ◆	2	170 ●	1
	300	1761	(1.6)	2399	(1.2)	5061	(2.0)	3872	1633 ●	1	1154 ●	1
5. Schwefel	10	0.00	(2.0)	0.00	(2.4)	0.00	(1.8)	0.90	0.00 ▲	7	0.00 ▲	8
	50	0.00	(2.0)	2.39	(2.4)	1964	(1.6)	13732	7.72 ■	4	0.01 ◆	3
	100	140	(2.0)	4578	(1.8)	203644	(2.4)	607496	2810 ◆	2	3930 ◆	3
	300	3642499	(1.8)	13928168	(1.0)	18563442	(2.0)	36584224	3204716 ●	1	4050201 ◆	2
6. Levy	10	0.00	(2.4)	0.00	(2.2)	0.00	(1.6)	0.07	0.01 ▲	8	0.00 ■	5
	50	0.01	(2.2)	0.14	(1.8)	4.90	(2.4)	20.5	0.04 ◆	3	0.03 ◆	2
	100	1.37	(1.8)	4.86	(1.6)	12.9	(1.0)	53.0	1.20 ●	1	1.01 ●	1
	300	18.8	(1.6)	29.9	(1.0)	139	(2.0)	94.5	12.7 ●	1	9.94 ●	1
7. Ackley	10	0.00	(2.4)	0.00	(2.6)	0.02	(2.2)	0.22	0.00 ■	4	0.00 ■	4
	50	0.00	(2.2)	0.41	(2.0)	1.04	(1.6)	1.50	0.26 ■	5	0.01 ●	1
	100	1.57	(2.0)	2.57	(1.8)	4.86	(1.2)	5.69	1.51 ●	1	1.36 ●	1
	300	5.29	(1.8)	6.37	(1.0)	10.1	(2.0)	8.32	4.46 ●	1	4.05 ●	1
8. Griewank	10	0.02	(1.6)	0.04	(1.4)	0.10	(1.2)	0.13	0.15	8	0.05 ■	5
	50	0.01	(2.2)	0.08	(1.8)	1.04	(1.6)	1.50	0.26 ■	5	0.01 ●	1
	100	0.13	(2.0)	1.05	(2.2)	3.57	(1.2)	7.43	1.06 ■	4	0.25 ◆	2
	300	7.01	(1.8)	14.7	(1.0)	56.8	(2.0)	37.7	5.54 ●	1	4.45 ●	1
9. Alpine	10	0.00	(2.2)	0.00	(2.0)	0.00	(2.6)	0.04	0.00 ▲	7	0.00 ■	6
	50	0.00	(2.0)	1.84	(1.6)	11.7	(1.2)	36.0	0.36 ◆	3	0.01 ◆	2
	100	2.13	(1.8)	21.7	(2.0)	60.8	(1.0)	123	8.07 ◆	2	0.42 ●	1
	300	188	(1.6)	290	(1.8)	774	(2.0)	569	142 ●	1	84.6 ●	1



(a) Rosenbrock

(b) Rastrigin



(c) Griewank

図 7.2 : 評価指標 I と最良解の目的関数値 $f(\mathbf{x}^{g\text{-best}})$ の推移 (適応型提案手法, $N = 50$)

8

おわりに

8.1 本論文のまとめ

本論文では、メタヒューリスティクスの適応化・設計論をテーマとして、適応化のための汎用的フレームワークである「適応化のフレームワーク」を構築した。さらに、適応化のフレームワークに基づき、① Cuckoo Search (CS) の適応能力の向上を目的とした適応型 CS の開発、② 高い適応能力を素質として備えた新たなメタヒューリスティクスの設計、③ その手法を基礎とした適応型メタヒューリスティクスの開発、を行った。

「実用上における従来の最適化手法の課題」や「最適化分野の環境の変化」に対応可能な最適化手法の枠組みとして登場したメタヒューリスティクスは、工学的な価値が高い最適化手法として注目され、手法の開発・改良に関する研究がますます増えている。多様な環境下での使用が想定されるメタヒューリスティクスは、多様な環境下においても高い探索性能を発揮可能な適応能力が不可欠であり、更なる有用性の向上のために、これまでに適応化のためのアプローチはいくつか提案されているが、① 統一的な視点に基づく適応能力を具備したメタヒューリスティクスの設計論が存在せず、既存の手法に対するパラメータ調整機能の付加のみに留まっていること、② 対象の手法に特化した観点に基づいており、メタヒューリスティクスに共通する普遍的な観点に基づいていないこと、などの課題が存在している。そこで本論文では、メタヒューリスティクスの適応能力の向上、および高い適応能力を有する新たなメタヒューリスティクスの開発において、適応能力の向上を実現し得る「汎用的フレームワーク」の構築を行い、そのフレームワークに基づき適応型メタ

ヒューリスティクスの開発を行った。

以下では、本論文の内容を改めてまとめる。

1章では、最適化分野の研究に関する歴史的経緯を述べ、近年注目されているメタヒューリスティクスにおける実用上の課題と、適応能力の向上、および高い適応能力を備えるメタヒューリスティクスの設計論の必要性を指摘した上で、「解析に基づくメタヒューリスティクスの改良・設計」という本論文の目的・位置付けを明確にした。

2章では、具体的な議論に入る前に、本論文で扱う最適化問題、および代表的な数理計画法やメタヒューリスティクスのアルゴリズムを記述した。また、数理計画法やメタヒューリスティクスの特徴を概観した上で、最適化手法における適応能力の重要性を述べた。

3章では、代表的なメタヒューリスティクスの探索構造を解析し、メタヒューリスティクスに共通する探索戦略を抽出した。さらに、4章以降で適応型メタヒューリスティクスの設計に対して統一的な視点で議論するために、抽象的な多様化・集中化を実現するための操作を定義・分類した。

4章では、構造解析を通じて得た探索戦略の観点から、メタヒューリスティクスの探索戦略と適応能力の関係について議論した。その議論を基に、メタヒューリスティクスの適応能力の向上のための汎用的フレームワークとして、「適応化のフレームワーク」を構築した。

5章では、適応化のフレームワークを既存手法であるCSに適用することで、適応型CSを開発した。また、探索状態の評価指標を提案し、数値実験を通じてその有効性を検証した。最後に、数値実験を通じて、オリジナルCSとの比較により、適応型CSの有用性を確認した。

6章では、適応化のフレームワークに基づき、適応能力に優れた探索構造を有する新たなメタヒューリスティクスを構築した。最後に、数値実験を通じて、他の手法との比較により、提案したメタヒューリスティクス（オリジナル提案手法）の有用性を確認した。

7章では、適応化のフレームワークに従い、6章で構築した手法を基礎とした適応型メタヒューリスティクスを開発した。最後に、数値実験を通じて、オリジナル提案手法との比較により、提案した適応型メタヒューリスティクスの有用性を確認した。

以上より、本論文で構築した適応化のフレームワークは、メタヒューリスティクスの適応能力の向上、および高い適応能力を備えたメタヒューリスティクスの開発・設計において、有効性が高いことを検証した。

8.2 今後の展望

適応化のフレームワークは、柔軟性・汎用性が高い。本論文では限られた数の適用例しか示すことができなかったが、この枠組みでは、既存の手法に対する適応化や、優れた適応型メタヒューリスティクスの設計を容易に行うことが可能となる。したがって、適応化のフレームワークの有効性を高めるために、他の手法に対しても適応化のフレームワークを適用し、その有用性を検証することが期待できる。さらに、適応化のフレームワークに従ったメタヒューリスティクスの設計例を数多く示すことで、適応型メタヒューリスティクスの設計において有用な情報を得ることが期待できる。

メタヒューリスティクスは数理計画法とは異なり、数理的な根拠を有していないが、本論文でテーマとした、メタヒューリスティクスの適応能力は、最適化手法の探索性能の向上に繋がるための能力として、極めて重要である。したがって、適応化のフレームワークで行われる適応型メタヒューリスティクスの設計は、その過程で何らかの数理的な根拠に基づいていることが望ましい。近年では、メタヒューリスティクスの探索ダイナミクスに対する数理的・数値実験的な解析に関する研究も行われている。1章や2章で述べたように、数理計画法とメタヒューリスティクスのアルゴリズムは全く異なるが、メタヒューリスティクスの実システムへの応用や更なる有用性の向上のためには、数学的知見の活用は有効であると考えられる。例えば、数理計画法における知見（降下方向、共役方向、凸性など）や、メタヒューリスティクスの探索モデルは離散時間確率システムとして捉えることができるため、確率システムに対する数学的知見（マルコフ連鎖、システムの安定性、確率統計・推定など）の活用が考えられる。PSOやDEなどの既存の手法に対する数理的解析のアプローチは、新たなメタヒューリスティクスの設計においても、収束安定性を高める上で貢献することが期待できる。さらに、メタヒューリスティクスの適応能力と収束安定性の関連性を明らかにすることで、本論文の抽象的な多様化・集中化に対する解釈・定義よりも、さらに有効な解釈・定義を行うことが可能となり、「数理的な根拠に基づく適応型メタヒューリスティクスの設計論」の構築が期待できる。

謝辞

本論文は、著者が首都大学東京大学院 理工学研究科 博士前期課程において、首都大学東京大学院 理工学研究科 電気電子工学専攻 安田 恵一郎 教授の指導の下で行ったメタヒューリスティクスに関する研究成果である。

本研究の遂行および本論文の作成にあたり、日頃からご指導頂いている安田 恵一郎 先生をはじめ、助教 土屋 淳一 先生、田村 健一 先生、システム制御工学研究室の方々には、多くの御指導、御助言を頂きました。心より御礼申し上げます。

参考文献

参考文献

- [1] 土屋淳一・鈴木隆之・安田恵一郎：「サーフェスモータの最適設計への統合的最適化の適用」, 電気学会 電子・情報・システム部門誌, Vol.130, No.1, pp.108-109 (2010)
- [2] A. Takezawa, S. Nishiwaki, and M. Kitamura: “Shape and Topology Optimization Based on the Phase Field Method and Sensitivity Analysis,” Journal of Computational Physics, Vol.229, No.7, pp.2697-2718 (2010)
- [3] 森川克己・中村信人：「ジョブショップの生産計画とスケジューリングに対する最適化アプローチ」, 日本経営工学会論文誌, Vol.51, No.3, pp.168-176 (2000)
- [4] 藤田薫・江本源一・竹下聡彦・佐中俊哉：「化学産業における最適化技術の適用」, オペレーションズ・リサーチ：経営の科学, Vol.48, No.8, pp.549-554 (2003)
- [5] 黒田充・村松健児 編：「生産スケジューリング」, 朝倉書店 (2002)
- [6] 加藤政一：「電力システム運用における最適化技術」, 電気学会 電力・エネルギー部門誌, Vol.121, No.2, pp.160-163 (2001)
- [7] 森啓之：「電力系統におけるメタヒューリスティクス応用の動向」, 電気学会 電力・エネルギー部門誌, Vol.123, No.10, pp.1120-1123 (2003)
- [8] 福山良和：「メタヒューリスティック手法の電力・エネルギー分野への適用例」, 電気学会 電力・エネルギー部門誌, Vol.124, No.5, pp.679-682 (2004)

- [9] 得居誠也：「最適化から見たディープラーニングの考え方」，オペレーションズ・リサーチ：経営の科学，Vol.60，No.4，pp.191-197（2015）
- [10] H. Konno and H. Yamazaki: “Mean-Absolute Deviation Portfolio Optimization Model and Its Application to Tokyo Stock Market,” Management Science, Vol.37, No.5, pp.519-531 (1991)
- [11] 若井亮介・新里隆・嶋崎善章：「ランダム行列を用いたポートフォリオ最適化解析」，日本経営工学会論文誌，Vol.65，No.1，pp.17-28（2014）
- [12] G. Dantzig: “Linear Programming and Extensions,” Princeton University Press (1998)
- [13] 田中謙輔 著：「凸解析と最適化理論」，牧野書店（1994）
- [14] 今野浩・山下浩 編著：「非線形計画法」，日科技連（1978）
- [15] 志水清孝・相吉英太郎 共著：「数理計画法」，昭晃堂（1984）
- [16] 坂和正敏 著：「非線形システムの最適化 <一目的から多目的へ>」，森北出版（1986）
- [17] 今野浩 著：「数理決定法入門 キャンパスのOR」，朝倉書店（1992）
- [18] 福島雅夫 著：「非線形最適化の基礎」，朝倉書店（2001）
- [19] 玉置久 編著：「システム最適化」，オーム社（2005）
- [20] 矢部博 著：「工学基礎 最適化とその応用」，数理工学社（2006）
- [21] 藤澤克樹・梅谷俊治 著：「応用に役立つ 50 の最適化問題」，朝倉書店（2009）
- [22] 相吉英太郎・安田恵一郎 編著：「メタヒューリスティクスと応用」，電気学会，オーム社（2007）
- [23] 安田恵一郎：「メタヒューリスティクスの現在と未来」，計測と制御，Vol.47，No.6，pp.453-458（2008）

- [24] 安田恵一郎：「進化論的計算手法とメタヒューリスティクス」, 電気学会 電子・情報・システム部門誌 , Vol.122 , No.3 , pp.320-323 (2002)
- [25] I. Boussaïd, J. Lepagnot, and P. Siarry: “A Survey on Optimization Metaheuristics,” Information Sciences, Vol.237, pp.82-117, Elsevier Publisher (2013)
- [26] 柳浦睦憲・茨木俊秀 著：「組合せ最適化 メタ戦略を中心として」, 朝倉書店(2001)
- [27] X. S. Yang: “Nature-Inspired Metaheuristic Algorithms,” Second Edition, Luniver Press (2010)
- [28] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” Proceedings of IEEE International Conference on Neural Networks, Vol.4, pp.1942-1948 (1995)
- [29] R. M. Storn and K. V. Price: “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Space,” Journal of Global Optimization, Vol.11, No.4, pp.341-359 (1997)
- [30] D. Karaboga: “An Idea Based on Honey Bee Swarm for Numerical Optimization,” Technical Report TR06, Computer Engineering Department, Engineering Faculty, Er-ciyes University (2005)
- [31] X. S. Yang: “Nature-Inspired Metaheuristic Algorithms,” Luniver Press (2008)
- [32] X. S. Yang: “Firefly Algorithms for Multimodal Optimization,” Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Sciences, Vol.5792, pp.169-178 (2009)
- [33] X. S. Yang and S. Deb: “Cuckoo Search via Lévy Flights,” Proceedings of World Congress on Nature and Biologically Inspired Computing 2009, pp.210-214 (2009)
- [34] 染谷博司：「進化型計算の挙動制御のためのダイナミクス解析」, システム/制御/情報 , Vol.57 , No.2 , pp.67-72 (2013)
- [35] K. Yasuda, N. Iwasaki, G. Ueno, and E. Aiyoshi: “Particle Swarm Optimization: A

- Numerical Stability Analysis and Parameter Adjustment Based on Swarm Activity,” IEEJ Transactions on Electrical and Electronic Engineering, Vol.3, No.6, pp.642-659 (2008)
- [36] 小熊祐司・相吉英太郎：「 Particle Swarm Optimization の力学系に対する乱数を考慮した安定性解析：持続探索のための最良パラメータ」, 電気学会 電子・情報・システム部門誌, Vol.130, No.1, pp.29-38 (2010)
- [37] 小熊祐司・相吉英太郎：「乱数係数をもつ Particle Swarm Optimization の力学系に対する統計的安定性解析」, 電気学会 電子・情報・システム部門誌, Vol.131, No.5, pp.1020-1030 (2011)
- [38] 若佐裕治・田中幹也・明石卓也：「 Particle Swarm Optimization アルゴリズムの安定性解析」, システム制御情報学会論文誌, Vol.23, No.1, pp.9-15 (2010)
- [39] K. Tamura, K. Makise, and K. Yasuda: “Concept of Feedback Controlled Differential Evolution and Its Realization,” IEEJ Transactions on Electrical and Electronic Engineering, Vol.10, No.4, pp.423-437 (2015)
- [40] K. Masuda, H. Yokota, and K. Kurihara: “An Empirical Study on the Search Directions of Differential Evolution,” Proceedings of the Congress on Evolutionary Computation 2011, pp.2550-2557 (2011)
- [41] D. Zaharie: “Influence of Crossover on the Behavior of Differential Evolution Algorithms,” Applied Soft Computing, Vol.9, No.3, pp.1126-1138 (2009)
- [42] 小熊祐司・相吉英太郎：「最大エントロピー法に基づく Differential Evolution の探索点分布解析」, 電気学会 電子・情報・システム部門誌, Vol.134, No.9, pp.1341-1347 (2014)
- [43] P. Civicioglu and E. Besdok: “A Conceptual Comparison of The Cuckoo-Search, Particle Swarm Optimization, Differential Evolution and Artificial Bee Colony Algorithms,” Artificial Intelligence Review, Vol.39, No.4, pp.315-346 (2013)

- [44] F. Wang, X. S. He, Y. Wang, and S. M. Yang: “Markov Model and Convergence Analysis Based on Cuckoo Search Algorithm,” *Computer Engineering*, Vol.38, No.11, pp.180-185 (2012) (in Chinese)
- [45] X. S. Yang and S. Deb: “Cuckoo Search: Recent Advances and Applications,” *Neural Computing and Applications*, Vol.24, No.1, pp.169-174 (2014)
- [46] S. Walton, O. Hassan, K. Morgan, and M. R. Brown: “Modified Cuckoo Search : A New Gradient Free Optimisation Algorithm,” *Chaos, Solitons & Fractals*, Vol.44, No.5, pp.710-718 (2011)
- [47] E. Valian, S. Mohanna, and S. Tavakoli: “Improved Cuckoo Search Algorithm for Global Optimization,” *International Journal of Communications and Information Technology*, Vol.1, No.1, pp.31-44 (2011)
- [48] M. Naik, M. R. Nath, A. Wunnava, S. Sahany, and R. Panda: “A New Adaptive Cuckoo Search Algorithm,” *Proceedings of 2015 IEEE 2nd International Conference on Recent Trends in Information Systems*, Vol.49, No.5, pp.4677-4683 (2015)
- [49] P. Ong: “Adaptive Cuckoo Search Algorithm for Unconstrained Optimization,” *The Science World Journal*, Vol.2014, pp.1-8 (2014)
- [50] R. N. Mantegna: “Fast, Accurate Algorithm for Numerical Simulation of Lévy Stable Stochastic Processes,” *Physical Review E*, Vol.49, No.5, pp.4677-4683 (1994)
- [51] T. Takahama and S. Sakai: “Solving Nonlinear Optimization Problems by Differential Evolution with a Rotation-Invariant Crossover Operation using Gram-Schmidt process,” *Proceedings of World Congress on Nature and Biologically Inspired Computing 2010*, pp.526-533 (2010)
- [52] 金政実・相吉英太郎:「回転不変性を考慮した超球交叉による Differential Evolution の改良」, 計測自動制御学会論文集, Vol.51, No.7, pp.503-511 (2015)
- [53] 森大輔・山口智:「主成分分析を取り入れた Artificial Bee Colony アルゴリズム」, 電

気学会 電子・情報・システム部門誌, Vol.135, No.4, pp.423-435 (2015)

著者の研究業績

学術論文（論文審査有り）

- [54] 熊谷渉・田村健一・土屋淳一・安田恵一郎：「Cuckoo Search のパラメータ解析と適応化に関する基礎検討」, 電気学会 電子・情報・システム部門誌, Vol.135, No.6, pp.721-722 (2015)
- [55] 熊谷渉・田村健一・土屋淳一・安田恵一郎：「探索状態の評価と制御に基づく適応型 Cuckoo Search」, 電気学会 電子・情報・システム部門誌 (投稿中)

国際会議（論文審査有り）

- [56] W. Kumagai, K. Tamura, and K. Yasuda: “Search Dynamics Analysis and Adaptive Parameter Adjustment of Cuckoo Search,” Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, pp.2634-2639 (2015)

学会発表（論文審査無し）

- [57] 熊谷渉・田村健一・安田恵一郎：「パラメータ調整機能を有する Cuckoo Search に関する検討」, 計測自動制御学会 システム・情報部門 学術講演会 2014, GS1-1, pp.1-6 (2014)
- [58] 熊谷渉・田村健一・安田恵一郎：「Cuckoo Search の探索ダイナミクスの解析と適応化に関する検討」, 進化計算学会 第 8 回進化計算研究会, P2-5, pp.113-120 (2015)
- [59] 熊谷渉・田村健一・安田恵一郎：「Cuckoo Search の探索ダイナミクスの解析と適応的パラメータ調整に関する基礎検討」, 平成 27 年電気学会全国大会, 3-025, pp.29-30

(2015) 【優秀論文発表賞 受賞】

- [60] W. Kumagai, K. Tamura, K. Yasuda: “Parameter Analysis and Adaption of Cuckoo Search,” Proceedings of 2015 IEEE Conference on Electronics, Information, and Systems, SS4-2, pp.1619-1620 (2015)



Lévy Flight に関する定量的解析

Lévy Flight に関する定量的解析方法と結果を述べる。具体的には, Cuckoo Search (CS) のパラメータ β と MA に従う Lévy 乱数 $L(\beta)$ の標本の統計量の関係について, 数値実験を通じて考察する。これによって, CS のパラメータ β と摂動の関係を明らかにする。以下に, 乱数の統計量を計算するアルゴリズムを示す。

【Lévy 乱数 $L(\beta)$ の統計量の計算アルゴリズム】

Step 0:[準備]

分布調整変数 β の下限値 β_{\min} , 上限値 β_{\max} , 調整幅 $\Delta\beta$, 乱数の上限 b , 乱数データのサイズ N を設定し, $\beta = \beta_{\min}$, 乱数データを $s(\beta) \in \mathbb{R}^N$ とする。

Step 1:[乱数の発生]

Lévy 乱数ベクトル $L(\beta) \in \mathbb{R}^N$ を発生させ, 乱数の上限 b を与えた後の乱数データ $s(\beta)$ の要素を

$$s_n(\beta) = \min \{ |L_n(\beta)|, b \}, \quad (n = 1, \dots, N)$$

とする。

Step 2:[乱数の統計量の計算]

乱数データ $s(\beta)$ の平均と標準偏差を

$$\mu(\beta) = \frac{1}{N} \sum_{n=1}^N s_n(\beta)$$

$$\sigma(\beta) = \sqrt{\frac{1}{N} \sum_{n=1}^N (s_n(\beta) - \mu(\beta))^2}$$

より求める。

Step 3:[終了判定]

$\beta = \beta_{\max}$ ならば終了する。さもなければ, $\beta := \min \{\beta + \Delta\beta, \beta_{\max}\}$ とし, Step 1 へ戻る。

Lévy 分布は平均値と標準偏差は ∞ だが, 上記のアルゴリズムで得た乱数の統計量を式 (1.14) の正規分布関数 $f_N(x; \beta)$ に代入することで, β と $|L(\beta)|$ の定量的関係が明らかになる。本論文では, $\beta_{\min} = 0.3$, $\beta_{\max} = 1.99$, $\Delta\beta = 0.05$, $b = 10$, $N = 10^6$ とした。図 A.1 に乱数の統計量, 図 A.2 に乱数データの正規分布の確率密度関数 $f_N(x; \beta)$ を示す。図 A.1 と図 A.2 から, 5.2.3 の定性的解析で明らかにした傾向に加えて, β と乱数 $|L(\beta)|$ の平均値 $\mu(\beta)$ や標準偏差 $\sigma(\beta)$ の間に関係性が確認できる。以上の解析結果から, β と摂動量は定量的関係が存在し, 初期配置領域 IS のサイズ b によって適切な β の範囲が異なることが明らかになった。

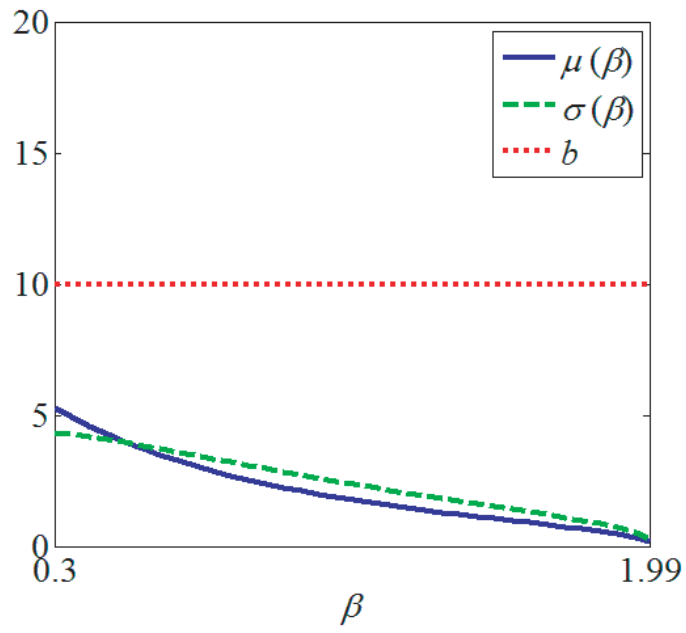


図 A.1 : Lévy 乱数 $|L(\beta)|$ の統計量

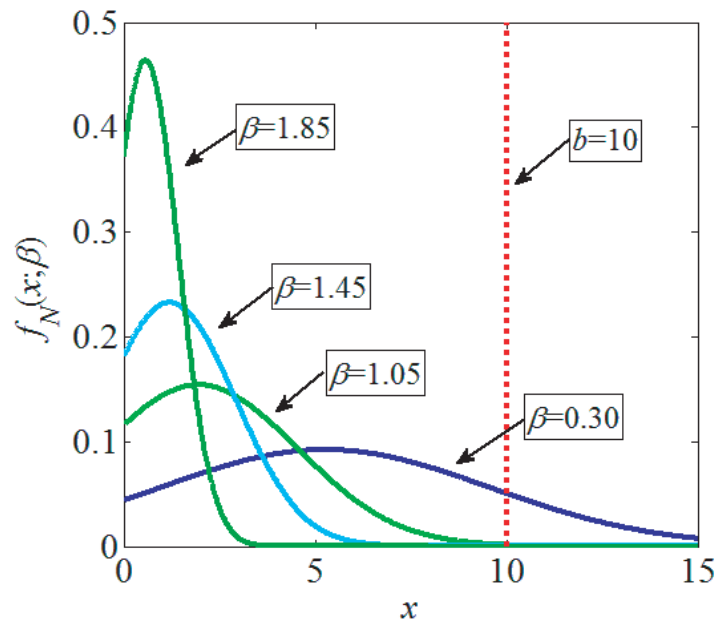


図 A.2 : Lévy 乱数 $|L(\beta)|$ の正規分布 $f_N(x; \beta)$

B

ベンチマーク関数

本論文で用いたベンチマーク関数について記述する。図中の★は大域的最適解 \boldsymbol{x}^o を表す。

1. Parabola Function

$$f_1(\boldsymbol{x}) = \sum_{n=1}^N x_n^2 \quad (\text{B.1})$$

$$\boldsymbol{x}^o = (0, 0, \dots, 0), \quad f(\boldsymbol{x}^o) = 0$$

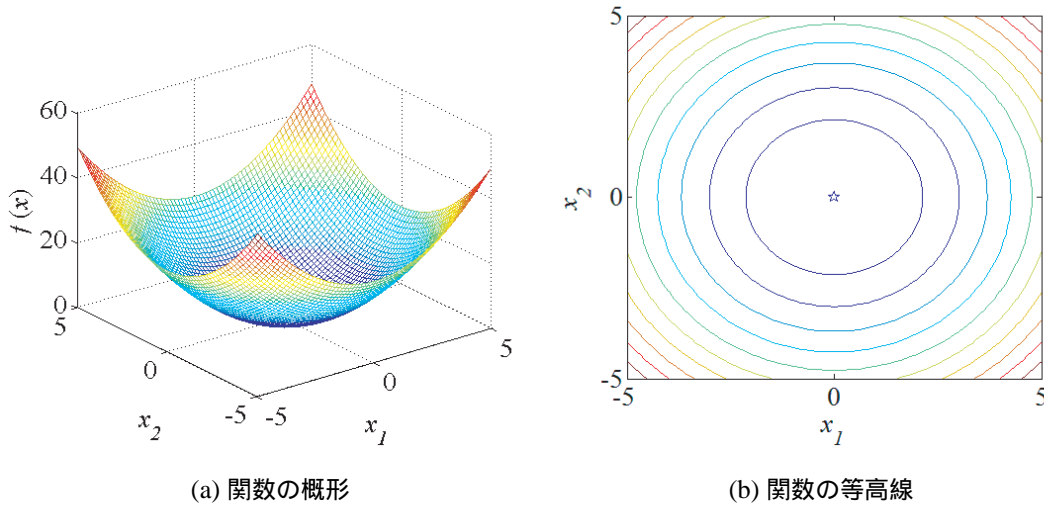


図 B.1 : Parabola Function

2. Rosenbrock's Saddle Function

$$f_2(\mathbf{x}) = \sum_{n=1}^{N-1} 100(x_n^2 - x_{n+1})^2 + (x_n - 1)^2 \quad (\text{B.2})$$

$$\mathbf{x}^o = (1, 1, \dots, 1), \quad f(\mathbf{x}^o) = 0$$

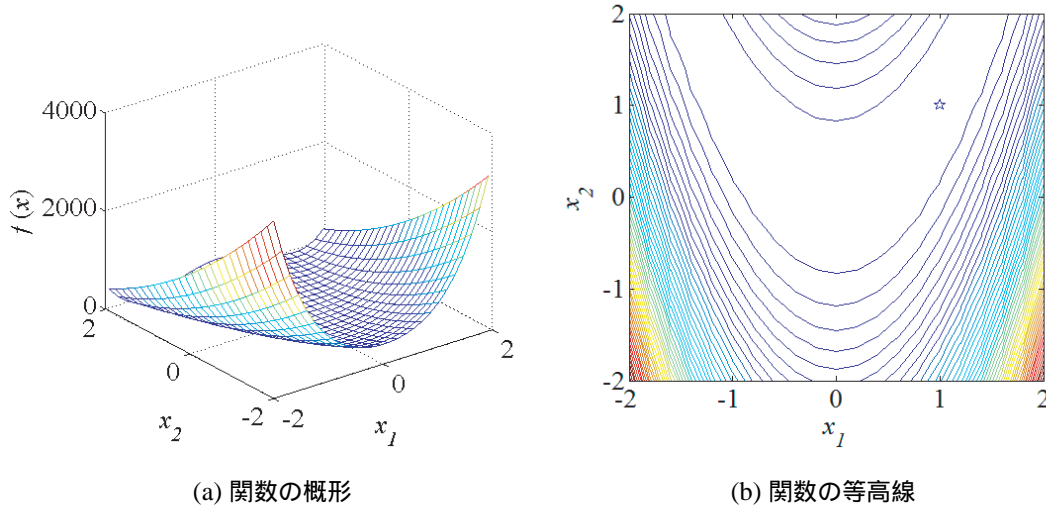


図 B.2 : Rosenbrock's Saddle Function

3. 2^N minima Function

$$f_3(\mathbf{x}) = \sum_{n=1}^N (x_n^4 - 16x_n^2 + 5x_n) \quad (\text{B.3})$$

$$\mathbf{x}^o \approx (-2.92, -2.92, \dots, -2.92), \quad f(\mathbf{x}^o) \approx -78N$$

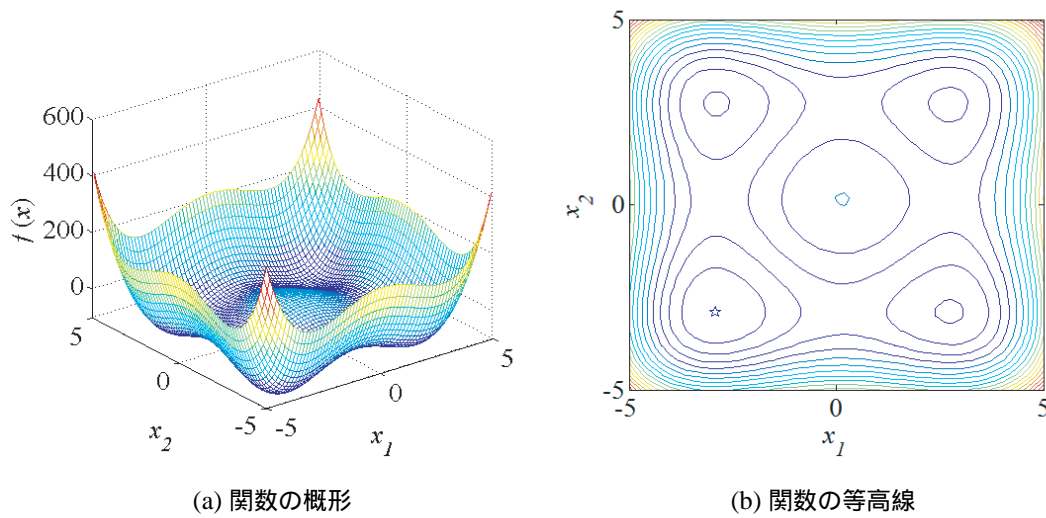


図 B.3 : 2^N minima Function

4. Rastrigin Function

$$f_4(\mathbf{x}) = \sum_{n=1}^N (x_n^2 - 10 \cos(2\pi x_n) + 10) \quad (\text{B.4})$$

$$\mathbf{x}^o = (0, 0, \dots, 0), \quad f(\mathbf{x}^o) = 0$$

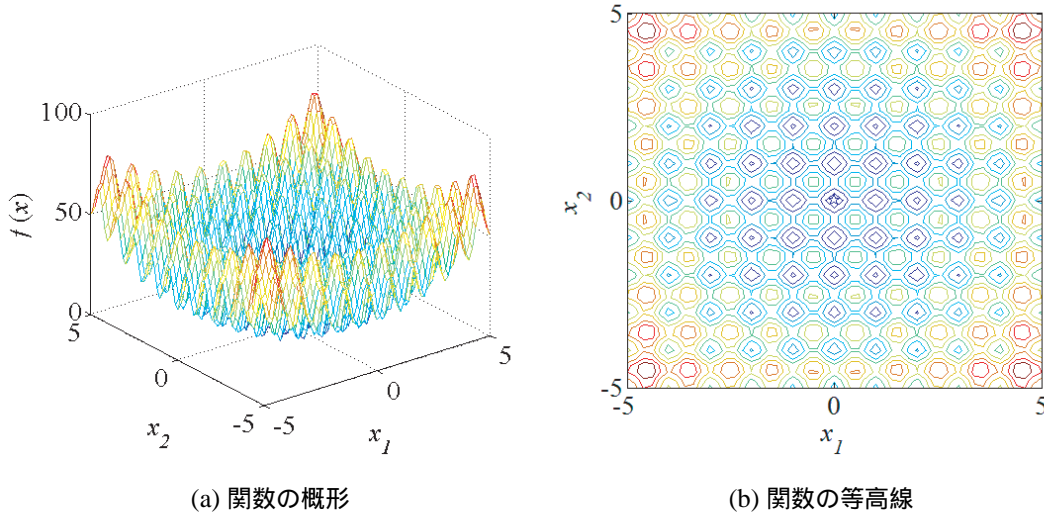


図 B.4 : Rastrigin Function

5. Schwefel Function

$$f_5(\mathbf{x}) = \sum_{n=1}^N \left(\sum_{i=1}^n x_i \right)^2 \quad (\text{B.5})$$

$$\mathbf{x}^o = (0, 0, \dots, 0), \quad f(\mathbf{x}^o) = 0$$

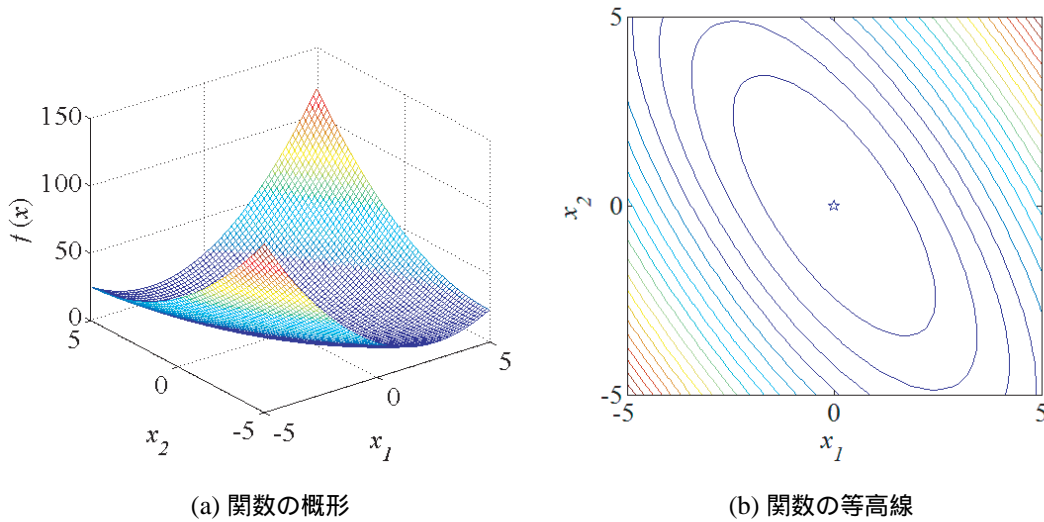
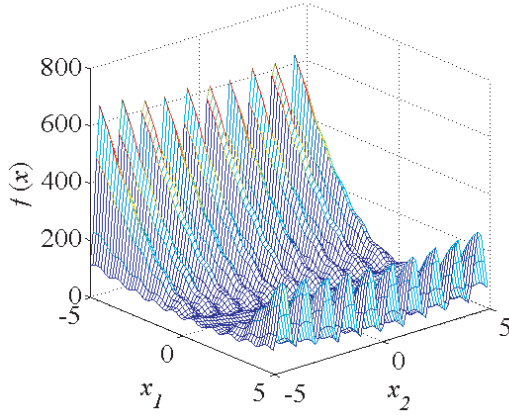


図 B.5 : Schwefel Function

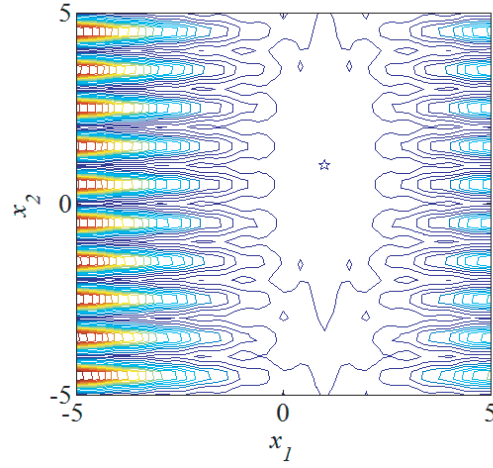
6. Lévy Function

$$f_6(\mathbf{x}) = \frac{\pi}{N} \left(\sum_{n=1}^{N-1} \{(x_n - 1)^2 (1 + 10 \sin^2(\pi x_{n+1}))\} + 10 \sin^2(\pi x_1) + (x_N - 1)^2 \right) \quad (\text{B.6})$$

$$\mathbf{x}^o = (1, 1, \dots, 1), \quad f(\mathbf{x}^o) = 0$$



(a) 関数の概形



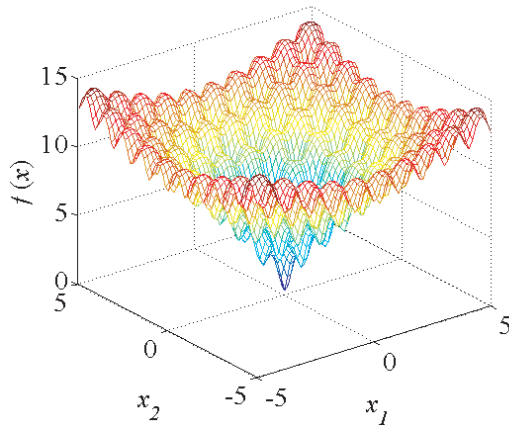
(b) 関数の等高線

図 B.6 : Lévy Function

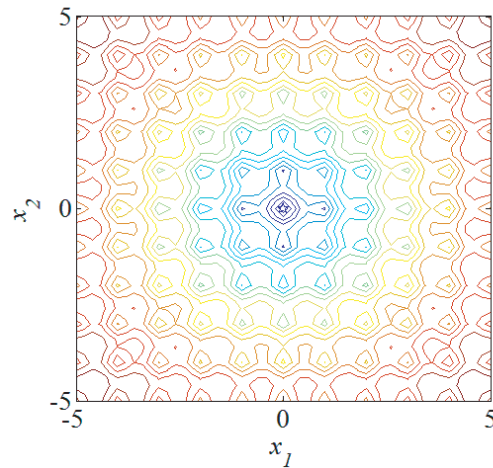
7. Ackley Function

$$f_7(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{1/N \sum_{n=1}^N x_n^2} \right) - \exp \left(1/N \sum_{n=1}^N \cos 2\pi x_n \right) + 20 + \exp(1) \quad (\text{B.7})$$

$$\mathbf{x}^o = (0, 0, \dots, 0), \quad f(\mathbf{x}^o) = 0$$



(a) 関数の概形



(b) 関数の等高線

図 B.7 : Ackley Function

8. Griewank Function

$$f_8(\mathbf{x}) = \frac{1}{4000} \sum_{n=1}^N \{x_n + \prod_{n=1}^N \cos(x_n / \sqrt{n}) + 1\} \quad (\text{B.8})$$

$$\mathbf{x}^o = (0, 0, \dots, 0), \quad f(\mathbf{x}^o) = 0$$

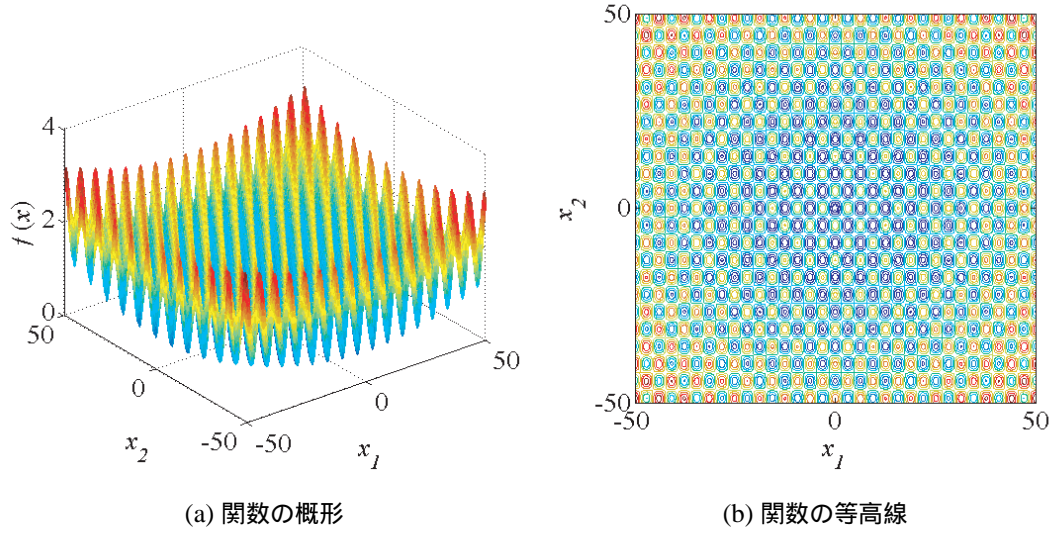


図 B.8 : Griewank Function

9. Alpine Function

$$f_9(\mathbf{x}) = \sum_{n=1}^N |x_n \sin x_n + 0.1x_n| \quad (\text{B.9})$$

$$\mathbf{x}^o = (0, 0, \dots, 0) \quad f(\mathbf{x}^o) = 0$$

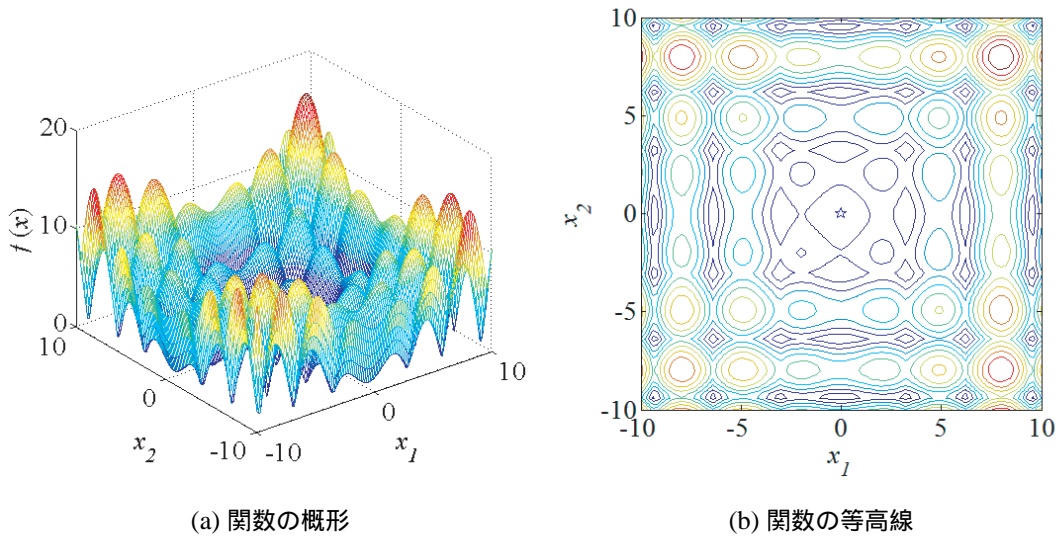


図 B.9 : Alpine Function