

学位論文 博士（工学）

変換不变性とパラメータ調整に基づく
メタヒューリスティクスの設計
に関する研究

2020 年度

東京都立大学大学院 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、社会学等の幅広い分野に及んで応用され、その重要性は広く認識されている。工学的応用を念頭においた最適化は、① 実システムの大規模さ・複雑さ、② システムのモデリング・シミュレーション技術、③ コンピュータの計算能力、④ 最適化理論および最適化手法、の4つの主要な要素が互いに強い影響を与えるながら発展してきた。近年のシステムの大規模化・複雑化やシステム・工業製品の性能に対する要求の高度化が進んでいる一方で、2010年頃から、第3次人工知能ブームを牽引する深層学習の登場や、General-Purpose Computing on Graphics Processing Unitsによる並列化・コンピュータパワーの増大などを背景にモデリング・シミュレーション技術も進化している。以上のように、最適化技術を取り巻く環境は急速に変化しており、これらの変化に対応可能な最適化技術が求められている。

古典的な最適化技法を集大成した数理最適化（数理計画法）は、1970年代の高速・大容量のコンピュータの出現を原動力にして、単体法や準Newton法をはじめとする実用的なアルゴリズムの構築と有用性の検証が急速に進展した。非線型連続最適化では、目的関数のクラスや性質（非凸性・悪スケール性・変数間依存性）が問題となるが、数理最適化は目的関数の解析的情報（勾配やHesse行列など）を使用することで、悪スケール性と変数間依存性に対処している。一方、近年、シミュレータなどのブラックボックスな入出力情報のみを用いて、良好な解を探索する問題、ブラックボックス最適化（Black-Box Optimization : BBO）の形態が、製品設計、機械学習のハイパーパラメータ決定、実験データベースを活用した新素材探索などにおいて増えている。BBOでは、対象の目的関数の解析的情報は得られず、数理最適化は適用できないため、入力（決定変数値情報）と出力（目的関数値情報）のみだけで探索が可能なアルゴリズムである、直接探索法の適用が必要である。代表

的な直接探索法の枠組みとしてメタヒューリスティクスが知られている。メタヒューリスティクスは、生物現象や自然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おり、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発見的近似解法である。また、多点探索・確率的探索であるため、並列処理に適している、非凸性に対応可能である点からも、環境変化に対応可能な最適化の枠組みとして期待できる。

しかし、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、十分な探索性能を発揮するには、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。さらに、BBOでは、目的関数の構造や性質が得られないため、これらの有無や変化に対してアルゴリズムの探索性能が影響されないことに加え、パラメータ設定がなるべく不要で探索性能を十分に発揮できることが望ましい。このため、BBOを解くアルゴリズムは、高いロバスト性や適応性を具備すべきである。ロバスト性とは、探索過程でアルゴリズムのパラメータを固定した状態でも、目的関数の構造や性質に対して探索性能が維持できる性質を表し、適応性とは、探索過程で逐次得られる問題構造の情報を活用して、アルゴリズムのパラメータを動的に調整することで、対象問題に適応する性質を表す。以上の背景から、高いロバスト性・適応性を具備するBBOのためのメタヒューリスティクスの開発は重要な課題となっている。

ロバスト性を実現する性質として変換不变性が、適応性を実現する機能としてパラメータ調整が挙げられる。よって、変換不变性とパラメータ調整に基づくメタヒューリスティクスは、BBOのための高いロバスト性・適応性を具備すると考えられる。以上の背景を踏まえ、本論文では、変換不变性とパラメータ調整に基づくメタヒューリスティクスの解析と開発に関する検討を行った。

メタヒューリスティクスの探索構造が変換不变性の定義を満たすかどうかを調べれば、その変換不变性の有無を指摘することが可能である。本論文では、このアプローチによって様々な既存手法の変換不变性の有無を明らかにすることで課題を指摘した後、代表的なメタヒューリスティクスである Particle Swarm Optimization (PSO), Artificial Bee Colony Algorithm (ABC), Cuckoo Search (CS)において欠如していた変換不变性を付加することで、不变性を有する手法を提案した。

また、本論文では、下記の(1)および(2)のロバスト性と適応性の観点に基づき、CSとPSOのパラメータ調整則を設計し、不变性を有する適応型メタヒューリスティクスを提案した。

- (1) 有効な探索戦略として多様化・集中化が知られており、多様化・集中化をより明確に実現することで、探索性能の向上が期待できる。多様化・集中化の観点からメタヒューリスティクスの探索状態を評価する指標を定義した後、メタヒューリスティクスが有するパラメータを解析し、評価指標に基づき探索過程で探索状態を有効に制御するパラメータ調整則を検討した。
- (2) メタヒューリスティクスの変換不变性は、パラメータ調整則の付加によって、その有無が変わることがある。これを用いて、①欠如している変換不变性を補完し、探索構造全体が多くの変換不变性を有しているかのように、パラメータ調整則を付加する、②有している変換不变性を維持するように、パラメータ調整則を付加する、という不变性に基づく二つの設計指針を提示した。これらの指針に従い、探索構造全体が多くの変換不变性を獲得するように、あるいは維持するようにパラメータ調整則を検討した。

さらに、本論文では、欠如している変換不变性を付加する上記のアプローチは、他の変換不变性を犠牲にすることが多く、より一般のアフィン変換不变性を獲得することが困難であることを指摘した。なお、アフィン変換は、回転変換、スケール変換、平行移動変換を含んでおり、解空間に施すことで変数間依存性、悪スケール性、原点依存性を発生させることができる。これを踏まえ、本論文では、アフィン変換不变性を具備する最適化アルゴリズムの「フレームワーク」を提案し、このフレームワークに対して不变性定理を与えた。この定理は、フレームワークの範囲で設計された全てのアルゴリズムに対して適用可能であり、要求条件さえ満たすようにアルゴリズムを設計すれば、アフィン変換不变性を具備することを表すものである。このフレームワークに従えば、アフィン変換不变性を具備することを保証しながら、様々な最適化能力を自由に付与でき、さらにパラメータ調整則を付加することも可能である。つまり、高いロバスト性と適応性の具備が可能な汎用的なフレームワークであるといえる。

本論文は、全7章より構成されており、各章の概要および得られた成果は以下の通りである。

第1章は序論であり、最適化のニーズや最適化アルゴリズムの発展を概観するとともに、BBOでの適用を目的とした最適化アルゴリズムが具備すべき基本的な要件を系統的に整理し、本論文の概要と位置付けを記載した。

第2章では、本論文の主眼としているBBOの最適化問題の性質と、BBOで使用される最適化アルゴリズムについて概観し、その数理的構造や特徴を記述している。特に、BBOのための最適化アルゴリズムには、ロバスト性と適応性が求められることを述べ、第3章以降へ繋がる問題提起を行った。

第3章では、第4章以降の数理的解析のための基盤を形成するために、既存のメタヒューリスティクスを概観しながら、その探索構造を解析し、共通の構造を抽出した。

第4章では、まず変換不变性の定義を述べ、様々な既存のメタヒューリスティクスの変換不变性について解析・評価した。そして、PSOに対して共分散行列を用いて回転不变性を付加した手法を提案し、数値実験を通じて回転変換に対してロバスト性が向上したことを確認した。さらに、ABCに対して超球を用いて回転不变性を付加した手法を提案し、数値実験を通じて回転変換に対してロバスト性が向上したことを確認した。最後に、CSに対して共分散行列を用いてアフィン変換不变性を付加した手法を提案し、数値実験を通じてアフィン変換に対してロバスト性が向上したことを確認した。

第5章では、既存のパラメータ調整則を概観しながら、変換不变性とパラメータ調整則の関連を明らかにし、不变性に基づくパラメータ調整則の設計指針を二つ提示した。一つ目の指針に従い、CSに対してスケール不变性の補完と適応性の向上が可能なパラメータ調整則を付加することで適応型CSを提案し、数値実験を通じてそのロバスト性・適応性を確認した。さらに、二つ目の指針に従い、第4章で提案した回転不变性を有するPSOに対して、不变性の維持と適応性の向上が可能なパラメータ調整則を付加することで、回転不变性を有する適応型PSOを提案し、数値実験を通じてそのロバスト性・適応性を確認した。

第6章では、第4章を含む、変換不变性を付加するアプローチは、他の変換不变性を犠牲にすることが多く、より一般のアフィン変換不变性を獲得することが困難であることを指摘した。そこで、アフィン変換不变性をすでに具備することが保証された、最適化アルゴリズムの「フレームワーク」を構築し、このフレームワークのもとで様々な最適化能力を付与することで、具体的なアルゴリズムを設計する新たなアプローチを提案した。このフレームワークに従って、PSOに着想を得た新たなアルゴリズムを設計し、数値実験を通じて、アフィン変換に対するロバスト性を有することを確認した。第7章は、本論文の結論であり、本論文で得られた研究成果および今後の研究の展望をまとめた。

目次

論文要旨	i
1 序論	1
1.1 本論文の背景 ······	1
1.2 本論文の目的と位置づけ ······	11
1.3 本論文の構成 ······	15
2 最適化アルゴリズムの基礎と数理	17
2.1 本章について ······	17
2.2 最適化問題 ······	18
2.2.1 最適化問題の定式化 ······	18
2.2.2 局所的最適解と大域的最適解 ······	18
2.2.3 最適化問題の分類 ······	19
2.3 ブラックボックス最適化において直面する最適化問題の性質 ······	21
2.3.1 ブラックボックス最適化 ······	21
2.3.2 変数間独立性／変数間依存性 ······	22
2.3.3 スケール性 ······	22
2.3.4 凸／非凸・单峰性／多峰性 ······	23
2.3.5 原点依存性 ······	24
2.4 最適化アルゴリズムの数理 ······	25
2.4.1 最適化アルゴリズムの基本構造と分類 ······	25
2.4.2 特殊な解・探索点 ······	27

2.5	数理最適化	28
2.5.1	数理最適化と最適化問題の関連性	28
2.5.2	勾配法	29
2.6	大域的最適化	37
2.6.1	確率的勾配降下法	37
2.6.2	力学系モデルに基づく大域的最適化手法	39
2.7	直接探索法	40
2.7.1	古典的な直接探索法	40
2.7.2	統計力学に基づくサンプリング	42
2.7.3	メタヒューリスティクス	46
2.8	逐次近似最適化	49
2.8.1	逐次近似最適化	49
2.8.2	Bayes 最適化	50
2.9	最適化アルゴリズムの最適化能力	51
2.10	最適化アルゴリズムのロバスト性と適応性	53
2.11	まとめ	59
3	メタヒューリスティクスの探索構造の解析	61
3.1	本章について	61
3.2	メタヒューリスティクスの基本的な探索構造	63
3.2.1	基本的な探索構造	63
3.2.2	探索点の更新則	65
3.3	Evolution Strategy の解析	67
3.3.1	Evolution Strategy	67
3.3.2	Evolution Strategy の探索構造の解析	68
3.4	Genetic Algorithm の解析	70
3.4.1	Genetic Algorithm	70
3.4.2	実数値 Genetic Algorithm	71
3.4.3	実数値 Genetic Algorithm の探索構造の解析	73
3.5	Simulated Annealing の解析	75

3.5.1	Simulated Annealing	75
3.5.2	実数値 Simulated Annealing	76
3.5.3	実数値 Simulated Annealing の探索構造の解析	76
3.6	Particle Swarm Optimization の解析	78
3.6.1	Particle Swarm Optimization	78
3.6.2	Pariticle Swarm Optimization の探索構造の解析	79
3.7	Differential Evolution の解析	82
3.7.1	Differential Evolution	82
3.7.2	Differential Evolution の探索構造の解析	83
3.8	Artificial Bee Colony Algorithm の解析	85
3.8.1	Artificial Bee Colony Algorithm	85
3.8.2	Artificial Bee Colony Algorithm の探索構造の解析	86
3.9	Firefly Algorithm の解析	89
3.9.1	Firefly Algorithm	89
3.9.2	Firefly Algorithm の探索構造の解析	91
3.10	Cuckoo Search の解析	93
3.10.1	Cuckoo Search	93
3.10.2	Cuckoo Search の探索構造の解析	95
3.11	メタヒューリスティクスの分類	97
3.11.1	多様化・集中化の定義	97
3.11.2	更新則の分類	99
3.12	まとめ	109
4	変換不变性に基づくメタヒューリスティクスのロバスト化	111
4.1	本章について	111
4.2	メタヒューリスティクスの変換不变性	112
4.2.1	変換不变性の定義	112
4.2.2	メタヒューリスティクスの変換不变性に関する先行研究	118
4.2.3	変換不变性と乱数ベクトルに関する重要な定理	123
4.3	変換不变性に基づくメタヒューリスティクスの解析	129

4.3.1	変換不变性に基づく Particle Swarm Optimization の解析 ······	129
4.3.2	変換不变性に基づく Differential Evolution の解析 ········	133
4.3.3	変換不变性に基づく Artificial Bee Colony Algorithm の解析 ···	138
4.3.4	変換不变性に基づく Firefly Algorithm の解析 ········	142
4.3.5	変換不变性に基づく Cuckoo Search の解析 ········	146
4.4	相関性を用いた回転不变性を有する Particle Swarm Optimization ·····	155
4.4.1	先行研究と背景 ········	155
4.4.2	相関性を用いた回転不变性を有する PSO ········	156
4.4.3	CRI-PSO の変換不变性の解析 ········	157
4.4.4	数値実験検証 ········	161
4.5	超球に基づく回転不变性を有する Artificial Bee Colony Algorithm ···	168
4.5.1	先行研究と背景 ········	168
4.5.2	超球に基づく回転不变性を有する ABC ········	169
4.5.3	HBC-ABC の変換不变性の解析 ········	169
4.5.4	数値実験 ········	174
4.6	基底変換を用いたアフィン変換不变性を有する Cuckoo Search ······	180
4.6.1	先行研究と背景 ········	180
4.6.2	アフィン変換不变性を有する Cuckoo Search ········	181
4.6.3	CMD-CS のアフィン変換不变性に関する解析 ········	183
4.6.4	数値実験検証 ········	185
4.7	まとめ ········	190
5	パラメータ調整に基づくメタヒューリスティクスの適応化	195
5.1	本章について ········	195
5.2	メタヒューリスティクスのパラメータ調整 ········	197
5.2.1	パラメータ調整方法の分類と一般形の定義 ········	197
5.2.2	Evolution Strategy のパラメータ調整方法 ········	199
5.2.3	Particle Swarm Optimization のパラメータ調整方法 ······	205
5.2.4	Differential Evolution のパラメータ調整方法 ········	210
5.2.5	Cuckoo Search のパラメータ調整方法 ········	215

5.2.6	変換不变性に基づくパラメータ調整方法の分類	216
5.3	相似変換不变性を補完する適応型 Cuckoo Search	223
5.3.1	背景	223
5.3.2	Cuckoo Search のパラメータ解析	223
5.3.3	相似変換不变性を補完する適応型 Cuckoo Search	229
5.3.4	変換不变性に基づく適応型 CS の解析	234
5.3.5	数値実験による検証	236
5.4	回転不变性を有する適応型 Particle Swarm Optimization	244
5.4.1	背景	244
5.4.2	CRI-PSO のパラメータ解析	245
5.4.3	適応型 CRI-PSO	248
5.4.4	変換不变性に基づく適応型 CRI-PSO の解析	250
5.4.5	数値実験検証	257
5.5	まとめ	263
6	変換不变性を有するメタヒューリстиクスのフレームワーク	269
6.1	本章について	269
6.2	変換不变性を有するメタヒューリстиクスのフレームワーク	271
6.2.1	共分散行列の固有値分解に基づく変換不变性を有する直接探索法	271
6.2.2	CMD-DS の特徴	272
6.3	CMD-DS の変換不变性定理	274
6.3.1	CMD-DS の変換不变性定理	274
6.3.2	CMD-DS の変換不变性の数値実験検証	276
6.4	CMD-DS に基づく PSO 着想アルゴリズム	278
6.5	数値実験検証	280
6.5.1	実験条件	280
6.5.2	探索性能のロバスト性に関する検証	280
6.5.3	全般的な性能に関する検証	282
6.6	アルゴリズムの探索性能に関する総合比較	287
6.7	まとめ	294

7 結論	297
7.1 本論文のまとめ	297
7.2 今後の展望	300
謝辞	305
参考文献	307
A 数学的表記と定理	329
A.1 数学的表記	329
A.2 線形代数に関する諸概念・定理	337
A.3 確率統計に関する諸概念・定理	342
B Mantegna's Algorithm	345
C 本論文で用いたベンチマーク関数	347

1

序論

1.1 本論文の背景

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。最適化は工学に限定されることなく、経済学、社会学等の幅広い分野に及んで応用され、その重要性は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。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している。

図 1.1 に示すように、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。

要素1 対象となる実システム

要素2 実システムを表現した数式モデル

要素3 数式モデルに適用可能な最適化アルゴリズム

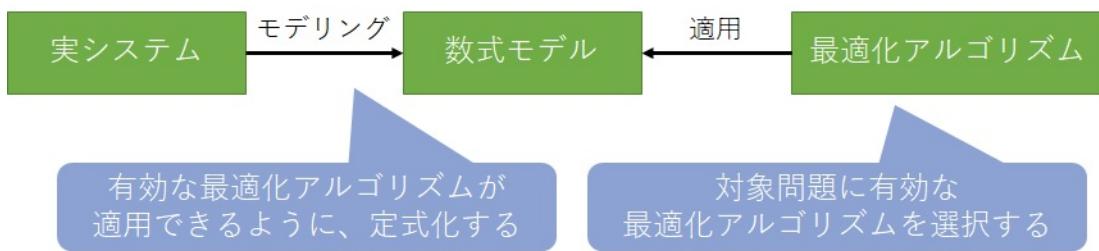


図 1.1：工学における最適化の形態

本来、上記の最適化の形態において、要素2では、有効な最適化アルゴリズムが適用できるように、数式モデルの定式化を行い、要素3では、対象の問題に有効な最適化アルゴリズムを選択することが常道である。これを踏まえ、工学の問題に最適化を適用するプロセスは、下記の5つのステップから構成されると考えられる。以下では、最適化アルゴリズムを研究開発・設計する立場の人間を「開発者」、開発されたアルゴリズムを使用して実際の最適化問題を解く立場の人間を「使用者（ユーザ）」と呼ぶことにする。

ステップ1 ユーザが原問題（要素1）のモデリングを行い、最適化問題（要素2）として定式化する。

ステップ2 開発者がある問題のクラスに適したアルゴリズムを開発する。

ステップ3 ユーザが問題に適したアルゴリズム（要素3）を選択する。

ステップ4 ユーザが問題に適するようにアルゴリズムのパラメータを調整する。

ステップ5 ユーザがアルゴリズムを適用し、得た解を吟味する。

このように最適化形態の要素2と要素3が密接な関係にあることから、ユーザは上記プロセスにおいて、対象問題に応じて適切な選択・作業が求められる。

さらに、工学的応用を念頭においていた最適化は、①実システムの大規模さ・複雑さ、②システムのモデリング・シミュレーション技術、③コンピュータの計算能力、④最適化理論および最適化アルゴリズム、の4つの主要な要素が互いに関係しており、これらが適切なバランスの上で存在することが理想である。これまでこれら要素の進歩・発展が、最適化における新たなバランスや各要素の新たな進歩・発展を促してきた。

①については、近年のシステムの大規模化・複雑化、システムの設計・運用・解析・制御や工業製品の性能に対する要求の高度化により、一層の高効率化・高信頼化・高機能化

が求められている。例えば、2020年に理化学研究所では、スーパーコンピュータを用いた新型コロナウイルス（COVID-19）の治療薬候補同定が行われたが、既存の薬剤約2000種類を候補としており、2011年のInternational Supercomputing Conferenceが発表したコンピュータの性能ランキング「TOP500 Lists」で1位だったスーパーコンピュータ「京」を用いて解いたとしても、1年以上要するほどの膨大な計算量であると発表された[12]。

②については、2010年頃の深層学習（Deep Learning, Deep Neural Network）[13]の成功が発端となり、それ以降第3次人工知能（Artificial Intelligence : AI）ブームが到来していることが強力な背景にある。深層学習は極めて複雑なシステムをモデリングできる能力を有しており、画像・音声認識[14, 15, 16]、分子の物性値予測[17]、タンパク質構造予測[18]、囲碁などのボードゲーム[19]などの様々なタスクで成果を挙げている。特に、囲碁は創造的かつ戦略的思考を必要とするボードゲームであるため、これまでではチェスなどと比較して、AIが囲碁で人間に勝つことは困難だと考えられていたが、2015年にGoogle DeepMind社のAlphaGoが当時世界トップの囲碁棋士に勝利したことで、囲碁界隈トップの人間に初めて勝利したコンピュータプログラムとなった[19]。また、タンパク質の構造は、アミノ酸配列から転写・翻訳されたペプチド鎖が物理法則に従い、折り畳まれる（Folding）ことで構成されるが、その物理法則は完全には解き明かされていない上に、与えられたアミノ酸配列に対応するタンパク質の構造の組み合わせも膨大なため、Foldingの予測は困難かつ重要な問題だと考えられ、Critical Assessment of Structure Prediction (CASP) というタンパク質構造予測コンペティションではその予測精度が競われてきたが、2018年のCASPでは、Google DeepMind社のAlphaFoldが他のチームを圧倒して勝利した[18]。AlphaGoやAlphaFoldを始めとする深層学習ベースのAIは、専用のタスクで目標達成のために人が作りこんだアルゴリズムではなく、ビッグデータに対して解析を行い、汎用のタスクを達成するためのアルゴリズムを自動的に抽出している点が、2020年現在のモデリング技術を象徴している。また、シミュレーション技術も、タスクを実行する前に発生する現象を把握することによる信頼性向上や、宇宙空間や超高压・高温などの特殊環境を含む多様なニーズへの対応など、非常に重要な位置付けとなっている。

③については、近年の高性能計算（High-Performance Computing : HPC）技術を搭載したスーパーコンピュータの進化は凄まじい。スーパーコンピュータは、従来CPUを基本としたベクトル型計算や並列型計算を採用していた[20]が、2010年頃から、安価・小型・高性能（理論演算性能）であるGeneral-Purpose Computing on Graphics Processing

Units (GPGPU) [21, 22, 23, 24]による並列計算や、超並列処理の方式が採用されている。2020年のInternational Supercomputing Conferenceが発表したコンピュータの性能ランキング「TOP500 Lists」では、理化学研究所と富士通の超並列処理方式の「富岳」が1位となった¹ [25]。2020年、上述のCOVID-19の治療薬候補同定問題に対して、「富岳」を用いて計算したところ、10日間で2000種類から数十種類の候補まで絞った上、その候補には実際に臨床試験が進んでいる薬剤も含まれていたことが発表され、2020年現在トップのHPC技術の計算速度の高さが証明された[12]。他にも、2020年の「TOP500 Lists」の上位層の多くでは、並列処理が得意なGPGPUが採用されている[25]。加えて、2020年現在、Amazon Web Services, Microsoft Azure, Google Cloud Platformなど、GPUコンピューティング環境のクラウドサービスが充実しているため、ユーザはハードウェアを準備せずにGPU環境を安価で利用可能な時代に突入している。これらのHPC技術の発展は、明らかに②のモデリング・シミュレーション技術に貢献していると同時に、第3次AIブームの要因の一つである。以上のように、④の最適化技術を取り巻く環境は急速に変化しており、これらの変化に対応可能な最適化技術が求められる。

線型最適化や凸最適化など、最適化問題のクラスに対応した理論に基づく最適化技法を数理最適化（数理計画法）という[26, 27, 28, 29, 30, 31, 32, 33, 34, 35]。連続最適化問題を対象とした数理最適化は、18世紀以前から最急降下法やNewton-Raphson法[26, 27, 28, 29, 30, 31, 32, 33]などが存在していたが、20世紀に入り、軍事用途などで最適化のニーズが高まるにつれて、線型最適化として1948年に単体法[34]が開発される、1960年代に凸最適化で重要な凸解析の理論が発展する[35]、など最適化理論の基盤が確立された。さらに、1970年代の高速・大容量のコンピュータの出現を原動力にして、準Newton法[26, 27, 28, 29, 30, 31, 32, 33]をはじめとする実用的なアルゴリズムの構築と有用性の検証が急速に進展した。数理最適化では、解への収束性や最適性に対する数学的な保証があるなど、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、上記の最適化形態の要素3において、従来から数理最適化が使用してきた。さらに、非線型連続最適化では、目的関数の構造や性質（悪スケール性・変数間依存性）などが問題となるが、数理最適化は目的関数の解析的情報（勾配やHesse行列など）を使用することで、ス

¹ 「富岳」は、ベクトル処理を拡張したScalable Vector Extensionを導入している。超並列処理などの登場により、そもそも処理方式自体も多様化しているが、GPGPUは使用されていない。

ケール性と変数分離可能性に対処している。しかしながら、上記の最適化形態の要素3、あるいはステップ3において数理最適化を採用する場合、下記の問題が生じる。

- 問題のクラスや性質（線形性・凸性・微分可能性）を前提に理論を構築している（要素2と要素3の依存関係が強い）ため、要素2のクラスが限定される。
- 手法の規定により、要素2、あるいはステップ1において、要素3の適用可能なクラス内に収まるように、実システムをモデリングする必要がある。
- 一方、要素1と要素2の間に乖離が生じるため、ステップ5で得た解の実行可能性や有用性が不十分となる。

この理由から、一般の非凸性を有する多峰構造（大域的最適化）に対する有効な数理最適化は未だに無く、凸最適化アルゴリズムを適用しても、大域的最適解に収束する保証はない。

近年、ブラックボックス関数の入出力情報のみを用いて、良好な解を自動的に獲得する問題である、ブラックボックス最適化（Black-Box Optimization : BBO）が登場している。BBOの目的関数は、シミュレーションなどの大規模・複雑な評価システムで、陽な定式化が不可能、あるいは困難なブラックボックス関数である。BBOの利用は、コンピュータやモデリング・シミュレーション技術の発展を背景に、製品設計[1, 36]やシステムオンライン運用・制御[37]など、より柔軟かつ複雑な形態で見られる。例えば、モデリング技術においては、2010年以降、深層学習やGPGPUによる並列処理が、第3次AIブームを牽引している。しかしながら、任意の問題（データセット）に対して、深層学習などの機械学習が性能を発揮するためには、いくつかのハイパーパラメータを適切に設定する必要がある。ハイパーパラメータのある値に設定し、機械学習アルゴリズムを適用するというプロセスを繰り替えすることで、その性能が最大化されるようにハイパーパラメータの値を探索する方法[38]があり、深層学習においてはネットワーク構造を含めて最適化することで構造自身も自動的に獲得する Neural Architecture Search (NAS) [39]の研究が知られている。これもハイパーパラメータ（入力）と性能（出力）の関係は不明であり、問題に応じて変わるために、BBOである。また、実験データベースやシミュレーション、AIによるモデリング、最適化を融合させ、化合物を探索することで、材料やタンパク質などの開発に活かす技術として、マテリアルズ・インフォマティクス（Materials-Informatics : MI）[40]やバイオ・インフォマティクス（Bio-Informatics : BI）[41]という用語が生まれている。従来の材料・素材開発では、理論計算、試作、実験を繰り返す過程で、研究者の経験と直

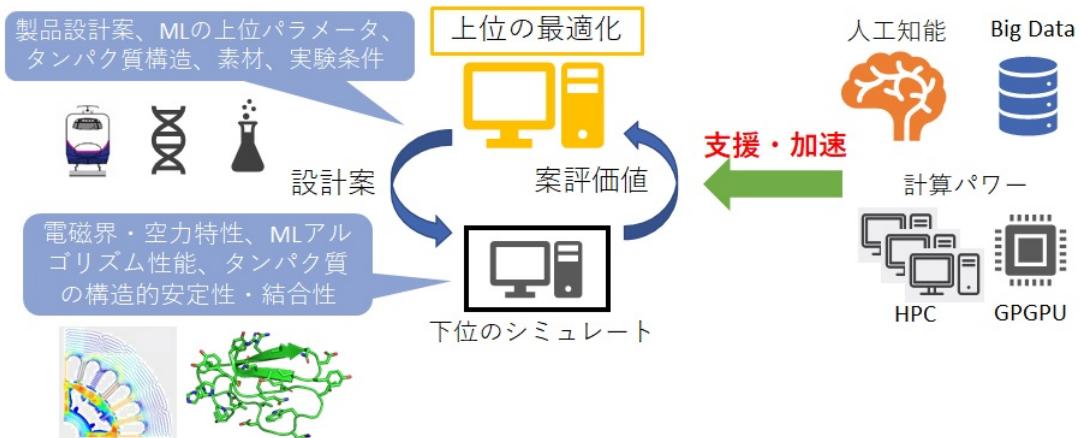


図 1.2： ブラックボックス最適化形態の例

感が頼りだったが、失敗も多く、精度の改善に多くの時間を費やしていた。しかしながら、MI や BI では、効果的、あるいは全く新しい機能を有する材料・タンパク質構造を効率的に探索することで、新素材創出や創薬支援が期待されている。以上から、図 1.2 に示すように、ビッグデータなどの扱う問題の大規模・複雑化、シミュレーション、AI、最適化などの計算技術、スーパーコンピュータや量子コンピュータなどのコンピュータの更なる進化などを背景にして、BBO の利用形態は今後もますます増えるであろう。

BBO は、ステップ 1においてユーザは原問題の様子を知ることができないため、最適化形態における要素 2 が無い。つまり、BBO では、対象の目的関数の入力（決定変数値情報）と出力（目的関数値情報）しか得られない上に、目的関数の景観が不明なため、目的関数の解析的情報が必須で凸に特化した数理最適化は適用できない。このため、決定変数値情報と目的関数値の情報だけで探索が可能なアルゴリズムの枠組みである、直接探索法 (Direct Search, Derivative-Free Optimization) [42]の適用が必要である。また、直接探索法は目的関数の解析的情報を用いないため、最適化形態における要素 3 において直接探索法を採用する場合、要素 2 と要素 3 の依存関係が無いため、適用可能な問題のクラスが規定されず、非凸の問題にも適用可能である。1960 年代に提案された決定的直接探索法として、Hooke-Jeeves 法[43, 44]や滑降シンプレックス法 (Nelder-Mead 法) [26, 45]がある。特に滑降シンプレックス法は、BBO や大域的最適化のアルゴリズムとして様々なソフ

トウェアで実装されている²。これらのアルゴリズムは、BBO で適用可能であり、10 次元以下の問題において良好な探索性能を示すが、これらのアルゴリズムは、次元数の増加に伴い探索性能が劣る上に、大域的最適化で有効とは言えない[48]。また、1958 年に提案されたランダムサーチは、確率的直接探索法である[49]。これは、ランダム性を利用することで局所的最適解に収束しにくくなり、無限に計算時間を与えた場合に大域的最適解に収束する保証がある。しかしながら、ランダムサーチは、その最適性が保証される条件で適用されることは困難であり、保証されない場合には効率的な探索を行うことができない。

以上のアルゴリズムとは異なる、新たな直接探索法の枠組みとして、メタヒューリスティクス (Metaheuristics, Metaheuristic Algorithm) [50, 51, 52, 53, 54, 55, 56] が知られている。代表的なアルゴリズムとして、Evolution Strategy (ES) [57, 58], Genetic Algorithm (GA) [59, 60, 61, 62], Simulated Annealing (SA) [63, 64, 65], Tabu Search (TS) [66, 67], Particle Swarm Optimization (PSO) [68], Differential Evolution (DE) [69], Artificial Bee Colony Algorithm (ABC) [70], Firefly Algorithm (FA) [71, 72], Cuckoo Search (CS) [55, 73, 74] などが挙げられる。ES, GA, DE は生物の進化過程、SA は金属の焼鈍し現象、TS は人間の記憶過程、PSO は鳥や魚の群れの採餌行動、ABC は蜜蜂の機能分担、FA は螢の求愛行動、CS はカッコウの托卵行動から着想を得ているため、発見的手法である側面が強調され、Nature-Inspired と称されることも多い。また、ES や GA のように、生物の進化過程に着想を得たアルゴリズムを進化計算 (Evolutionary Computation), PSO や FA のように、群れの知的行動に着想を得たアルゴリズムを群知能 (Swarm Intelligence) と呼ぶこともあるが、これらの呼称は 2020 年時点でメタヒューリスティクスとほぼ同義である。1960 年代に ES, GA が進化計算アルゴリズムとして登場した。1983 年に SA が、1989 年に TS が提案され、単純な局所的探索の上位に発見的操作を冠した、メタヒューリスティクスという用語が生まれた。この時代では、GA や SA が組合せ最適化アルゴリズムであることや、GA が生物が集団として進化することを模倣し、多点探索であることに対して、SA と TS が単点探索であることなど、枠組みとしてメタヒューリスティクスは体系づけられていなかった。しかしながら、1984 年に実数値 SA [75]、1990 年代に実数値 GA [76] など連続値最適化問題に適用できるように拡張され

² 例えば、Python のモジュール SciPy におけるライブラリ optimize [46] や、MATLAB の最適化用ソルバ fminsearch [47] では、滑降シンプレックス法が実装されている。

た。さらに、多点型 ES [58, 77] や ES with Covariance Matrix Adaptation (CMA-ES) [78, 79]などの進化計算が発展すると同時に、1995年に PSO と DE などの群知能アルゴリズムも提案されるなど、進化計算に限らず、群知能アルゴリズムもメタヒューリスティクスとして融合され始めた。それ以降も、ABC, FA, CS など、新たなメタヒューリスティクスの開発がますます盛んになっている。このような動向を背景に、2000年頃から、米国電気学会 (Institute of Electrical and Electronics Engineers : IEEE) の Congress on Evolutionary Computation (CEC)³ や Association for Computing Machinery (ACM) の Genetic and Evolutionary Computation Conference (GECCO)⁴ などの国際学会が開催されており、2020年現在では CEC と GECCO はこの分野においてトップの国際会議である。国内でも 2010年にこの分野に特化した進化計算学会が創設されるに至っている。また、IEEE の Computational Intelligence Society によると、計算知能の主要な研究分野は、ニューラルネットワーク、ファジィシステム、進化計算とされており、重要な分野として認識されている。

多くのメタヒューリスティクスの特徴は、① 最小の情報で最適化が可能な直接探索法、② 生物の進化過程や群知能などの経験則などをアノロジーとして構築される発見的手法、③ 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可能な近似解法、④ 確率的乱数を利用した確率的解法、⑤ 相互作用を及ぼす複数の探索点を利用する多点探索型手法、などが挙げられる。特に、多点探索型・確率的・直接探索法という側面が、古典的な数理最適化や直接探索法とはアルゴリズムの本質として異なる点である。解空間内で複数の探索点が移動する過程に確率的操作が含まれるため、初期点への依存性を解消することや、多峰性でも局所的最適解に陥りにくくなることが期待されている。また、多点探索であるため、並列処理が適しており、その進化の恩恵を受けやすい[80, 81, 82]。実際に、電磁アクチュエータ・エンジン設計、エネルギー設備運用計画、音声認識システム、機械学習のハイパーパラメータ探索、NAS など様々な分野での成果を挙げている [1, 6, 7, 8, 83, 84, 85, 86, 87]。以上のように、メタヒューリスティクスは大域的最適化や BBO において、有力な最適化アルゴリズムとして地位が確立している。

³ 1999年以降、前身である IEEE International Conference on Evolutionary Computation と Society of EP が CEC を共催することになった。

⁴ 1999年に、前身である International Conference on Genetic Algorithms と Genetic Programming Conference が統合され、GECCO となった。

メタヒューリスティクスは適用範囲が広く、有力な最適化アルゴリズムである一方で、アルゴリズムの探索性能は目的関数の構造や性質から影響を受ける、調整可能なパラメータを有しているが、その設定が適切でないと十分な探索性能を発揮できない、などの課題が挙げられる。特に BBO の環境下では、問題の情報が事前に得られないため、ユーザはプロセスの各ステップで適切な作業ができず、ステップ 5 で得た解を吟味した後に手戻りが必要となる場合がほとんどである。具体的には、ステップ 3 では問題に対して適切なアルゴリズムを選択できない、ステップ 4 では問題に対してアルゴリズムのパラメータを設定できない、などの理由で、ステップ 5 の後にステップ 3, 4 に戻って再度アルゴリズムの選択とパラメータを設定し直すことが必要となる。言い換えると、BBO では、下記の課題が挙げられる。

- 目的関数のクラスや性質の有無や変化に対して、アルゴリズムの探索性能が影響されないことが望ましいが、BBO ではユーザがそれに応じてアルゴリズムを適切に選択できない。
- 目的関数のクラスや性質の有無や変化に対して、パラメータ設定がなるべく不要で探索性能を十分に発揮できることが望ましいが、BBO ではそれに応じてユーザがパラメータを適切に設定できず、高い専門的な知識・経験や試行錯誤を必要とする。

このため、ステップ 2において、BBO を解くアルゴリズムは、高いロバスト性や適応性を具備すべきである。ロバスト性とは、探索過程でアルゴリズムのパラメータを固定した状態でも、目的関数の構造や性質の変化に対して探索性能が維持できる性質を表し、適応性とは、探索過程で得られる問題構造の情報を活用して、アルゴリズムのパラメータを動的に調整することで、対象問題に適応する性質を表す。以上の背景から、BBO のための高いロバスト性・適応性を具備するメタヒューリスティクスの開発は重要である。

ロバスト性を実現する性質として変換不变性が挙げられる。変換不变性は、疑似乱数系列が同じであれば、解空間や目的関数の変換に対してアルゴリズムの探索性能が再現性を有する、つまり同等あるいは類似の結果が得られる性質である。メタヒューリスティクスが有する変換不变性は手法によって様々である。実際、ある変換に対する不变性を欠如している手法は、その変換の有無によって、探索性能が大幅に変化することがしばしば見られる。具体的には、解空間の変換としてアフィン変換（スケール変換、回転変換、平行移動の合成変換）、目的関数の変換として単調増加変換が挙げられる。特に、悪スケール性の対

処としてスケール不変性、変数間依存性の対処として回転不変性、原点依存性の対処として平行移動不変性、などは有効であるため、アフィン変換不変性は重要である。変換不変性は1990年代にHansenらによって提案され、それをESに組み込むことでCMA-ESが開発された[88, 89]。それ以降、PSOやDEを中心に回転不変性などの欠如を指摘する、あるいは不変性を付加する研究がなされた[90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112]。

さらに、適応性を実現する機能としてパラメータ調整が挙げられる。これまでのメタヒューリスティクスにおけるパラメータ調整則は、適応性のレベルによって下記の3種類に分類できる。

固定型： パラメータはユーザが予め設定し、探索過程で同じ値で固定する機構。

スケジューリング的調整型： 事前に定められた時間経過に応じたスケジュールに従って、探索過程でパラメータを動的に調整する機構。

適応的調整型： 探索の時間経過に限らず、探索過程で得られた情報を活用して、パラメータを適応的に調整する機構。

加えて、パラメータの時系列を決定する方法は、下記の2種類に分類できる。

フィードフォワード型： 事前に与えた情報だけを用いて、パラメータの時系列を確定する。

フィードバック型： 探索過程で得られる情報も用いて、パラメータの時系列を逐次生成する。

固定型とスケジューリング的調整型はフィードフォワード型、適応的調整型はフィードバック型に分類される。これらの分類に基づくと、フィードバック型・適応的調整型のパラメータ調整則が最も適応性の発揮を期待できる。パラメータ設定・調整の研究は、1980年代のESの1/5ルール[58]などを発端として、1990年代では、CMA-ESのStep-Size Adaptation [78, 79]、PSOのConstriction Method [113]が開発された後、2000年以降は、PSOのパラメータ調整則として、Linearly Decreasing Inertia Weight Method [114]やLinearly Decreasing Vmax [115]、Activity Feedback PSO [116]など、DEのパラメータ調整則として、jDE [117]やJADE [118]、Success-History Based Adaptive DE [119]など、古くから2020年現在まで精力的に研究してきた。このように、パラメータ設定・調整に対する有効な知見を抽出し、系統的に分類・整理することや、パラメータの自律的調整の可能性を

検討することは、メタヒューリスティクスにおいて重要な課題である。特に、CMA-ESはアフィン変換不变性と適応的調整型パラメータ調整則を有している。実際、国際会議GECCOのワークショップ「Black-Box-Optimization-Benchmarking (BBOB) [120]」やCECのスペシャルセッション「Competition on Real-Parameter Single Objective Optimization [123]」では、BBOのコンペティションが開催されるが、CMA-ESや適応型DEは上位に位置している[121]。以上のように、変換不变性とパラメータ調整則を有するメタヒューリスティクスは、BBOのための高いロバスト性・適応性を具備すると考えられる。

1.2 本論文の目的と位置づけ

BBOでは変換不变性とパラメータ調整が重要である反面、先行研究ではメタヒューリスティクスのアルゴリズム毎にそれらを付加するアプローチが異なる。詳細は4章で述べるが、例えば、PSOの回転不变性を付加する先行研究は、①探索点の移動候補範囲を超球化する[95, 99]、②探索点の更新式に基底変換を導入する[91, 96, 98, 100, 101]、③探索点の更新式において係数をスカラ（線形結合）とする[90, 97, 102]、と分類できる。しかしながら、これらのアプローチは、他のアルゴリズムに適用しても回転不变性を獲得できるとは限らない、回転不变性を獲得できても他の不变性が犠牲となり、より一般のアフィン変換不变性を獲得できるとは限らない、などの問題がある。また、メタヒューリスティクスの変換不变性は、パラメータ調整則の付加によって、その有無が変わることがある。これを用いれば、①欠如している変換不变性を補完し、探索構造全体が多くの変換不变性を有しているかのように、パラメータ調整則を付加することが可能、②有している変換不变性を維持するように、パラメータ調整則を付加することが必要、という設計指針が立てられる。しかしながら、これらのアプローチを使ってメタヒューリスティクスのロバスト性・適応性の向上を行っている研究例は、著者の知る限り少ない。これらの問題は、メタヒューリスティクスの探索構造は発見的に開発されており、その探索構造毎に異なる特徴を有するが、それらには数理的根拠がないことに起因する。つまり、アルゴリズム毎に数理的構造を解析し、変換不变性の有無を判断した上で、元々具備している不变性を失わないようにながら、アフィン変換不变性を具備する方向へ改良する方法を慎重に選択、あ

るいは開発しなければならないが、ユーザにとっては困難である。

以上の課題を踏まえ、本論文では、下記4項目を目的とした、変換不变性とパラメータ調整に基づくメタヒューリスティクスの設計に関する研究を行う。

- (1) メタヒューリスティクスの変換不变性について汎用的に適用可能な解析方法を確立させた後、代表的なアルゴリズムについて適用し、それらの変換不变性の有無を調べることで、更新式の数理的構造と不变性の具備の関係について明らかにする。
- (2) 先行研究における不变性を付加するアプローチに基づき、いくつかの既存のアルゴリズムについて不变性を付加することで、不变性を有するアルゴリズムを提案する一方で、このアプローチの課題を明らかにする。
- (3) パラメータ調整則を含む更新式に対しても不变性の解析が可能であることを示すと同時に、この知見に基づいてパラメータ調整を活用することで、不变性の補完が可能であることと、ロバスト性を維持しながら適応性の向上が可能であることを示す。
- (4) アフィン変換不变性を具備するメタヒューリスティクスの「フレームワーク」を構築すると同時に、そのなかで最適化能力を自由に組み込むことでアルゴリズムを設計する考え方を示す。

目的(1)については、変換不变性の定義を示し、代表的なアルゴリズムがその定義を満たすかどうかを具体的に導出すれば良い。変換不变性の定義は、Hansen らにより提示されており、ES に適用している[88, 89, 93, ?]ため、本論文ではこれを活用する。また、Wilke らは、不变性を解析する方法を提示している[91]。これらの方法を用いて、PSO, DE, 実数値 GA, FA の変換不变性を解析した例はある[91, 92, 108, 122, 124]が、Wilke らの方法[91]は、平行移動不变性、回転不变性、相似変換不变性の3種類に限定しており、より一般のアフィン変換不变性や目的関数の単調増加変換不变性の有無を調べることはできない、Hansen らの方法[88, 89, 93, ?]は、定義を示しただけで具体的な導出例は ES に限定されている、などの課題がある。さらに、上記以外のアルゴリズムについて、変換不变性の有無を明らかにした例は、著者の知る限りない。以上から、本論文では、より一般のアフィン変換不变性や目的関数の単調増加変換不变性に関する解析の方法を確立した後、代表的なメタヒューリスティクスに適用し、不变性の有無を調べることで、更新式の数理的構造と不变性の具備の関係について明らかにし、後のアルゴリズム設計に活用する。

目的(2)については、先行研究と同様に、いくつかのアルゴリズムについて不变性の付

加を検討する。PSO や DE に対して回転不变性を付加した先行研究の中には、他の不变性を犠牲し、より一般のアフィン変換不变性を獲得できないケースがある[90, 94, 95, 96, 97, 98, 103, 107]。これは、① メタヒューリスティクスが発見的手法であるため、その数理的構造はあまり着目されず、数理的解析がなされる例は少ない、② 目的(1)で挙げたように、一般のアフィン変換不变性の有無について調べる具体的な方法が知られていない、などの理由のためだと考えられる。本論文では、先行研究において、アルゴリズムと変換不变性を付加するアプローチの関係を系統的に分類し、目的の不变性獲得のための方法に関して統一的な視点を得る。その後、PSO, ABC, CS に対して、各アルゴリズムにおいてはまだ適用されていない既存のアプローチを流用することで、不变性を有するアルゴリズムを提案する。なお、CS に関する不变性の解析、あるいは不变性を付加した先行研究は、著者の知る限りない。また、これらの検討を通じて、探索構造とアプローチの関係から不变性の有無への影響を具体的に明らかにしながら、このアプローチの課題を示す。

目的(3)については、適応性とロバスト性の観点に基づき、いくつかのアルゴリズムについてパラメータ調整則の付加を検討する。有効な探索戦略として多様化・集中化が知られており、多様化・集中化をより明確に実現することで、探索性能の向上が期待できる。Yasuda らは、多様化・集中化の観点から PSO や DE の探索状態を評価する指標を定義した後、評価指標に基づき探索過程で探索状態を有效地に制御する適応的パラメータ調整則を提案している[116, 125]。また、変換不变性は、パラメータ調整則の付加によって、その有無が変わることがある。金政らは、ESにおいて、相似変換不变性（スケール不变性）が欠如していることと、スケールパラメータの適応的調整によってスケール不变性を補完することが可能であることを指摘し、適応的調整則を検討している[126]。他にも、金政らは、DEにおいても、超球交叉を導入することで回転不变性を付加したが、スケール不变性を犠牲となることを指摘すると同時に、超球半径スケールパラメータを導入し、その適応的調整則を付加することで、スケール不变性も補完している[107]。秋本らは、実数値 GA の Real-Coded Ensemble Crossoverにおいて、交叉の拡張率の適応的調整則を検討しており、その際にパラメータ調整則の不变性とアルゴリズム全体の不变性の関係に留意している[127]。Hansen らは、CMA-ESにおいて、パラメータ調整則を有する状態でのアフィン変換不变性の具備を証明している[128, 163]。しかしながら、これらの先行研究では、特定のアルゴリズムの探索構造とパラメータ調整則の関係に関する考察に限定されており、一般的な更新式の数理的構造との関係は不明である。これを踏まえ、本論文では、代表的

なパラメータ調整則を概観すると同時に、パラメータ調整と変換不変性の関係を統一的な視点から整理する。その視点に基づき、CSに対して、アルゴリズム全体の不変性を補完しながら、多様化・集中化の探索戦略を活用した適応的パラメータ調整則を検討し、それを付加した適応型CSを提案する。さらに、目的(2)で検討した回転不変性を有するPSOに対して、アルゴリズム全体の不変性を維持しながら、多様化・集中化の探索戦略を活用した適応的パラメータ調整則を検討し、それを付加した回転不変性を有する適応型PSOを提案する。なお、CSやPSOに対して、アルゴリズム全体の不変性の関係に着目しながら、適応的パラメータ調整則を検討した先行研究は、著者の知る限りない。また、目的(2)と同様に、これらの検討を通じて、探索構造とアプローチの関係から不変性の有無への影響を具体的に明らかにしながら、このアプローチの課題を示す。

目的(4)については、目的(2), (3)を通じて明らかとなった、先行研究における不変性を付加するアプローチの課題の解決を検討する。従来のアプローチは、アルゴリズムの数理的構造の制約上、適用不可である場合があるため、適用可能な方法を選択する必要があるが、アルゴリズム毎に困難な数理的解析が必要であるという点から、汎用な方法だとはいえない。つまり、メタヒューリスティクスの更新式は、各アルゴリズム特有のアナロジーに基づいて最適化能力が設計されているため、変換不変性の付加アプローチもアルゴリズム毎に依存し、アルゴリズムのバリエーションまで考慮されていない。以上を踏まえ、本論文では、アフィン変換不変性を具備する最適化アルゴリズムの「フレームワーク」を提案し、このフレームワークに対して不変性定理を与える。この定理は、フレームワークの範囲で設計されたすべてのアルゴリズムに対して適用可能であり、要求条件さえ満たすようアルゴリズムを設計すれば、アフィン変換不変性を具備することを表すものである。このフレームワークに従えば、アフィン変換不変性を具備することを保証しながら、様々な最適化能力を自由に付与でき、さらにパラメータ調整則を付加することも可能である。つまり、高いロバスト性と適応性の具備が可能な汎用的なフレームワークであるといえる。

以上の成果により、BBOにおいて、ステップ2でロバスト性・適応性の高い最適化アルゴリズムを選択・適用することが可能となり、また、独自に最適化アルゴリズムを設計する場合でも、ロバスト性・適応性の高いメタヒューリスティクスの設計が可能となる。したがって、1.1節で挙げたBBOにおける課題解決に対して大きく資するものであると考えられる。

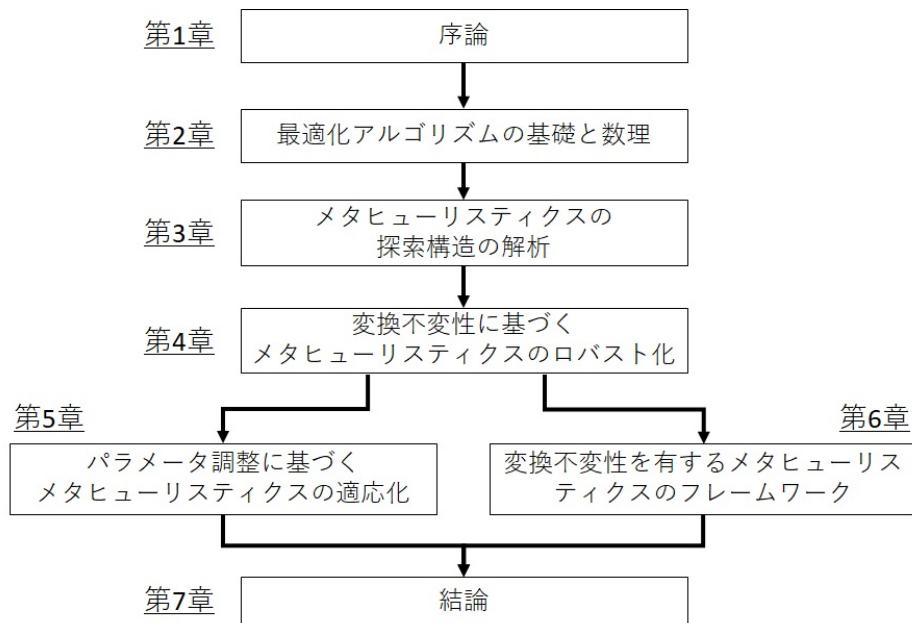


図 1.3： 本論文の章構成

1.3 本論文の構成

本論文の構成を以下に述べる。図 1.3 に本論文の章構成を示す。

第 2 章では、本論文の主眼としている BBO の最適化問題の性質と、BBO で使用される最適化アルゴリズムについて概観し、その数理的構造や特徴を記述する。特に、BBO のための最適化アルゴリズムには、ロバスト性と適応性が求められることを述べ、第 3 章以降へ繋がる問題提起を行う。

第 3 章では、第 4 章以降の数理的解析のための基盤を形成するために、既存のメタヒューリスティクスを概観しながら、その探索構造を解析し、共通の構造を抽出する。

第 4 章では、まず変換不変性の定義を述べ、変換不変性の汎用的な解析方法を確立する。その方法を用いて、様々な既存のメタヒューリスティクスの変換不変性について解析・評価する。また、先行研究における変換不変性を付加するアプローチを系統的に整理し、演算形式毎に異なることを指摘する。そして、PSO に対して共分散行列を用いて回転不変性を付加した手法を提案し、数値実験を通じて回転変換に対してロバスト性が向上することを確認する。さらに、ABC に対して超球乱数を用いて回転不変性を付加した手法を提案

し、数値実験を通じて回転変換に対してロバスト性が向上することを確認する。また、CSに対して共分散行列を用いて回転不变性とスケール不变性を付加した手法を提案し、数値実験を通じてアフィン変換に対してロバスト性が向上することを確認する。最後に、これらの検討を通じて、先行研究と第4章を含めた、変換不变性を付加するアプローチの課題を提起する。

第5章では、既存のパラメータ調整則を概観しながら、変換不变性とパラメータ調整則の関連について述べ、変換不变性に基づくパラメータ調整則の設計指針を2つ提示する。そして、1つ目の指針に従い、CSに対して、スケール不变性の補完と適応性の向上が可能な適応的パラメータ調整則を具備する、適応型CSを提案し、数値実験を通じてそのロバスト性・適応性が向上することを確認する。さらに、2つ目の指針に従い、第4章で提案した回転不变性を有するPSOに対して、変換不变性の維持と適応性の向上が可能な適応的パラメータ調整則を具備する、回転不变性を有する適応型PSOを提案し、数値実験を通じてそのロバスト性・適応性が向上することを確認する。

第6章では、第4章で指摘した、変換不变性を付加するアプローチは、他の変換不变性を犠牲にすることが多く、より一般のアフィン変換不变性を獲得することが困難であるという課題の解決を検討する。そこで、アフィン変換不变性をすでに具備することが保証された、最適化アルゴリズムの「フレームワーク」を構築し、このフレームワークのもとで自由に最適化能力を付与することで、具体的なアルゴリズムを設計する新たなアプローチを提案する。このフレームワークに従って、PSOに着想を得たサンプルアルゴリズムを設計し、数値実験を通じて、アフィン変換に対するロバスト性を有することを確認する。最後に、本論文で扱ったアルゴリズムの全般的な探索性能について総合比較し、評価する。

最後に第7章は、本論文の結論であり、本論文で得られた研究成果および今後の研究課題・展望をまとめた。

2

最適化アルゴリズムの基礎と数理

2.1 本章について

本章では、本論文の主眼としている BBO の最適化問題の性質と、BBO で使用される最適化アルゴリズムについて概観し、その数理的構造や特徴を記述する。特に、BBO のための最適化アルゴリズムには、ロバスト性と適応性が求められることを述べ、3 章以降へ繋がる問題提起を行う。[2.2](#) 節では、一般の最適化問題を定式化し、分類する。[2.3](#) 節では、特に BBO で直面する最適化問題の性質について概観する。[2.4](#) 節では、最適化アルゴリズムを大別し、特殊な解の表記を述べる。[2.5](#) 節では、凸最適化として使用される古典的な数理最適化について、[2.6](#) 節では、大域的最適化アルゴリズムについて、[2.7](#) 節では、代表的な直接探索法について、[2.8](#) 節では、BBO でよく使用される逐次近似最適化について概観する。[2.9](#) 節では、数理最適化や直接探索法の最適化能力について整理する。[2.10](#) 節では、BBO で要求される最適化アルゴリズムのロバスト性・適応性について述べる。

2.2 最適化問題

2.2.1 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。本論文で扱う連続最適化問題は、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 $f(\mathbf{x})$ 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。

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

目的関数 $f(\mathbf{x})$ の最大化問題を扱う場合であっても、 $f(\mathbf{x})$ 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 $\mathbf{x} \in S$ を実行可能解という。実行可能領域 S は、ベクトル値不等式制約関数 $\mathbf{g} : \mathbb{R}^N \rightarrow \mathbb{R}^L$ と、ベクトル値等式制約関数 $\mathbf{h} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ を用いて、式 (2.2) として表す。

$$S = \{\mathbf{x} \in \mathbb{R}^N \mid \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\} \quad (2.2)$$

式 (2.2)において、 $\mathbf{g}(\mathbf{x}) \geq \mathbf{0}$ を不等式制約条件、 $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ を等式制約条件という。實際には、不等式制約条件として、全ての解に対する上下限制約条件を与えることが多い。この制約は上下限 $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ を用いて、式 (2.3) として表される。

$$\{\mathbf{x} \in \mathbb{R}^N \mid a_n \leq x_n \leq b_n, n = 1, \dots, N\} \quad (2.3)$$

2.2.2 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 $\mathbf{x}^o \in S$ が式 (2.4) を満たす場合、解 \mathbf{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 \mathbf{x}^o の目的関数値 $f(\mathbf{x}^o)$ を大域的最適値という。

$$f(\mathbf{x}^o) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in S \quad (2.4)$$

式(2.4)を満たす大域的最適解 \mathbf{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 $\mathbf{x}^* \in S$ が式(2.5)を満たす場合、解 \mathbf{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 \mathbf{x}^* の目的関数値 $f(\mathbf{x}^*)$ を局所的最適値という。

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in S \cap B(\mathbf{x}^*; \varepsilon) \quad (2.5)$$

ただし、 $B(\mathbf{x}^*; \varepsilon)$ は、式(A.6)の距離空間 (\mathbb{R}^N, d_E) において、ある点 $\mathbf{x}^* \in \mathbb{R}^N$ に対する ε -近傍である。式(2.5)を満たす局所的最適解 \mathbf{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(2.6)の関係式が成立する。

$$S^o \subseteq S^* \quad (2.6)$$

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。

2.2.3 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アルゴリズム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、実行可能領域 S が課される場合、有制約最適化問題という。一方、 S が課されない、すなわち $S = \mathbb{R}^N$ の場合、無制約最適化問題といい、式(2.7)として定式化される。

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^N \quad (2.7)$$

現実に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[26, 27, 28, 31, 32]という。数理最適化では、外点ペナルティ関数法を用いることで、制約条件を陽に考慮せずに、最適化条件を満足する解を得ることができる。よって、これまで無制約最適化問題に対する最適化アルゴリズムを中心に開発されてきている。

また、一般に最適化アルゴリズムによって得る解は、上下限制約条件を常に満たすとは限らない。そこで、解 $\mathbf{x} \in \mathbb{R}^N$ の暫定位置 $\mathbf{u} \in \mathbb{R}^N$ 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(2.8)のリミッティング関数などが挙げられる。

$$x = \ell(u) = \begin{cases} a, & u < a \\ u, & a \leq u \leq b \\ b, & u > b \end{cases} \quad (2.8)$$

(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アルゴリズムは、単体法や内点法などが知られている[26, 27, 28, 31, 32, 34]。しかし、非線形最適化問題に対して有効な最適化アルゴリズムは、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 \mathbf{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 \mathbf{x}^o を有する。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次最適化問題などの一部に限定されている。

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^o は一致しない。非凸最適化問題において大域的最適解 \mathbf{x}^o を確実に求めることは困難であることから、大域的最適化（Global Optimization）と呼ばれる。

2.3 ブラックボックス最適化において直面する最適化問題の性質

2.3.1 ブラックボックス最適化

非線形最適化では、変数間依存性、悪スケール性、非凸性のような問題のクラスや性質が主要な問題であるが、目的関数の解析的情報（勾配や Hesse 行列など）を用いることで対処している。一方、ブラックボックス最適化（Black-Box Optimization : BBO）は、対象の目的関数のクラスや性質が不明であるため、これらの有無や強弱を判断できない上に、目的関数の解析的情報を用いることができないため、直接探索法では他の対処が必要となる。なお、多峰性・非凸な目的関数を対象とする大域的最適化では、勾配や Hesse 行列を用いることは可能だが、単に用いても局所的最適解への収束が問題となるため、大域的最適解を求めるための様々な対処が必要である。国際会議では、ブラックボックス最適化アルゴリズムのコンペティションが開催され、上記のクラス・性質を中心としたベンチマーク問題を整備している。例えば、国際会議 ACM の GECCO のワークショップでは「Black-Box Optimization Benchmarking」が開催されている[120]。他にも、国際会議 IEEE の CEC のスペシャルセッション「Competition on Real-Parameter Single Objective Optimization」が開催されている[123]。2.2 節では、おおまかに問題のクラスを分類したが、本節では、BBO で直面する問題のクラス・性質について概観する。

また、付録 C に本論文で使用するベンチマーク関数とその性質を記載している。

2.3.2 変数間独立性／変数間依存性

目的関数を $f(\mathbf{x}) = \sum_{n=1}^N f_n(x_n)$ のように、各変数のみに依存する目的関数に分解することで、 N 個の 1 次元最適化問題に分離できる性質を変数分離可能性 (Separability, Decomposability)，あるいは変数間独立性 (Independency) という。一方、複数の変数間で互いに依存関係があり、目的関数を各変数に分解できない性質を変数分離不可能性 (Non-Separability, Non-Decomposability)，あるいは変数間依存性 (Dependency, Epistasis) という。

一般的に、多くの最適化アルゴリズムは各変数を独立に探索するため、変数分離可能な関数の探索は容易であるが、変数間依存性を持つ関数の探索は困難とされている。変数間分離可能な関数の等高線を回転させると、変数間依存性が発生する場合がある¹。逆に、変数間分離不可能な関数の等高線を回転させると、変数間依存性が解消される場合がある。しかし、BBO では、事前に目的関数の性質が得られないため、事前に適切な回転を与えることはできない。この性質から、関数の変数間依存性に対するアルゴリズムの性能は、本論文の主眼となるアルゴリズムの回転不变性に依存しており、回転不变性を有するアルゴリズムは変数間依存性の影響を受けにくい（ロバストである）ことが知られている[93]。

2.3.3 スケール性

各変数に目的関数 f への貢献度合いが等しい性質を良スケール性 (Well-Scaling)，あるいは良条件 (Well-Conditioning) ということに対して、その貢献度合いが大きく異なる問題性質を悪スケール性 (Ill-Scaling)，あるいは悪条件 (Ill-Conditioning) という。悪スケール性は、例えば、ある変数が長さ [m] と別の変数が重さ [g] などのように、異なるスケールを持つ変数を持つ問題において現れる。特に悪条件という呼称は、正值対称行列 \mathbf{H} に対して目的関数が $f(\mathbf{x}) = 2\mathbf{x}^T \mathbf{H} \mathbf{x}$ と表せた時、 \mathbf{H} の条件数² が 1 より大きいときに使用することがあるが、ほとんどの場合、悪スケール性と同義である。

一般的に、多くの最適化アルゴリズムは各変数を等しいスケールで探索するため、良スケール性を有する関数の探索は容易であるが、悪スケール性を有する関数の探索は困難と

¹ 目的関数 $f(x_1, x_2)$ の項に $x_1 x_2$ などの交差項が含まれるときに、変数間依存性が発生する。

² 固有値の最大値と最小値の割合を条件数と呼ぶ。

されている。例えば、付録 C の Schwefel 関数は、目的関数の等高線が橿円形であり、変数間依存性と悪スケール性を有する。非線型最適化の基本的なアルゴリズムである最急下降法は、このような悪スケール性を有する関数を対象としたとき、直前の移動方向と直交する方向に移動、つまりジグザグに移動することになるため、多くの探索を要する。また、メタヒューリстиクスを悪スケール性を有する関数に適用したとき 目的関数への影響力が強い変数が優先的に最適化され、影響力の弱い変数の探索が十分に行われないため、Schwefel 関数のような单峰性関数であったとしても、探索点群が初期収束を起こしてしまう可能性がある。

良スケール性を有する関数を対象としたとき、その関数が有する変数に $\mathbf{x} \mapsto \mathbf{Cx}$ (\mathbf{C} は対角行列) というスケール変換を作用させると、悪スケール性が発生する。逆に、悪スケール性を有する関数に適切なスケール変換を作用させると、悪スケール性が解消される場合がある。しかし、BBO では、事前に目的関数の性質が得られないため、事前に適切なスケール変換を与えることは難しい。この性質から、目的関数の悪スケール性に対するアルゴリズムの性能は、本論文の主眼となるアルゴリズムのスケール不変性に依存しており、スケール不変性を有するアルゴリズムは悪スケール性の影響を受けにくい（ロバストである）ことが知られている[93]。また、目的関数が良スケール性を有していても、各変数に等しいスケーリング（相似変換） $\mathbf{x} \mapsto \delta\mathbf{x}$ ($\delta > 0$) を施すと、最適化アルゴリズムの性能は影響を受けることがある。つまり、最適化問題毎に持つ、変数固有のスケールに影響される場合がある。このように、各変数共通のスケールに対するアルゴリズムの性能は、アルゴリズムの相似変換不変性に依存しており、相似変換不変性を有するアルゴリズムはこのスケールの影響を受けにくい。ただし、各変数の共通のスケールは、変数の実験データや上下限制約などから予想しやすく、事前に大まかだが適切な範囲のスケーリングを与えることができるため、深刻な問題になりにくい。

2.3.4 凸／非凸・单峰性／多峰性

2.2 節で述べたように、非凸最適化問題を対象とした大域的最適化は、特に困難なクラスであることが知られている。凸最適化では、局所的最適解と大域的最適解は一致する。このとき、目的関数の景観は单一の峰から構成されていることになるため、单峰性 (Uni-Modal)

とも呼ばれる。一方、非凸最適化では、複数の局所最適解が存在し、目的関数の景観は多くの峰から構成されているため、多峰性（Multi-Modal）とも呼ばれる。

凸最適化問題では、凸性を活用した最適化アルゴリズム（数理最適化が中心）を適用すれば良いが、BBOでは、事前に凸性の有無を判断できないため、非凸・大域的最適化であることを前提にした最適化アルゴリズムを適用すべきである。

多峰性の更なる分類として、大域的単峰性（Single-Funnel）と大域的多峰性（大谷構造、騙し構造：Big-Valley Structure, Multi-Funnel）がある。大域的単峰性は小さな谷（局所的最適解）が複数存在しており、大域的な関数景観は一つの大きな谷の構造をしている性質である。一方、大域的多峰性は大域的最適解から離れた位置に大きな谷（有望な局所的最適解）が複数存在しており、大域的な関数景観は大きな谷が複数連なる構造をしている性質である。一般的に、大域的単峰性の目的関数は、アルゴリズムで探索点が大きな谷の底（大域的最適解）を目指すように設計されていれば、探索が容易だが、大域的多峰性の目的関数は、探索点が局所的最適解に陥りやすく、探索が困難なクラスだと言われている[203]。

2.3.5 原点依存性

付録Cに記載しているような非線型ベンチマーク関数は、最適解の位置が**0**であることが多いが、実問題ではそうであるとは限らない。このため、最適解の位置が**0**である非線型ベンチマーク関数において、事前に**0**付近を探索するように設計されたアルゴリズムは、その問題においては優れた性能が期待できるが、BBOでは望ましくない。このように、原点に依存するアルゴリズムの性質を原点依存性と呼ぶ。

原点移動は、目的関数の解空間を平行移動 $\mathbf{x} \mapsto \mathbf{x} - \mathbf{t}$ ($\mathbf{t} \in \mathbb{R}^N$) を施すことで可能である。しかしながら、初期配置領域内が最適解を覆っている限りは、初期点の位置を基準とした探索を行えば、この原点移動の影響を受けないため、ほとんどの最適化アルゴリズムは原点依存性を有しておらず、原点移動にロバストである。よって、原点依存性はアルゴリズム設計において必須な性質だが、優れた性能を発揮するためのアルゴリズムの研究においてはさほど重要でない。

2.4 最適化アルゴリズムの数理

2.4.1 最適化アルゴリズムの基本構造と分類

本論文では、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アルゴリズム（大域的最適化アルゴリズム）、あるいはBBOで適用可能な直接探索法について扱う。また本論文では、上下限制約条件が存在しない上に、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（目的関数変換法）ものとする。したがって、本論文で扱うのは、無制約最適化問題を対象とするアルゴリズムに限定する。ただし、メタヒューリスティクスのような多点型直接探索法を有制約最適化問題に適用する際は、目的関数変換法以外にも様々な方法があり、制約対処法として現在も精力的に研究されている[129, 130, 131, 132, 133, 134]。

(a) 直接解法／反復解法

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、一般的な最適化アルゴリズム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 $\mathbf{x} \in \mathbb{R}^N$ の更新式は、式(2.9)で表される。

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{v}(k) \quad (2.9)$$

ただし、 $\mathbf{x}(k)$ は反復回数 k における探索点、 $\mathbf{v}(k)$ は $\mathbf{x}(k)$ から $\mathbf{x}(k+1)$ までの移動ベクトルである。最適化アルゴリズム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。

(b) 単点探索／多点探索

反復解法を行う最適化アルゴリズムの中でも、一つの探索点によって最適解の探索を行う最適化アルゴリズムを単点型最適化アルゴリズムという。単点型最適化アルゴリズムでは、探索空間 \mathbb{R}^N 内に、探索点の初期位置 $\mathbf{x} \in \mathbb{R}^N$ を与え、アルゴリズムの規則によって、探索点の位置 \mathbf{x} を反復的に更新する。この反復の計算で得られる点列 $\chi(k)$ を反復回数 k までの探索点 \mathbf{x} の探索履歴といい、探索履歴 $\mathcal{X}(k)$ は式 (2.10) で表される。

$$\chi(k) = \{\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(k)\} = \{\mathbf{x}(\kappa)\}_{\kappa=1}^k \quad (2.10)$$

一方、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アルゴリズムという。多点型最適化アルゴリズムでは、探索空間 \mathbb{R}^N 内に、複数の探索点の初期位置 $\mathbf{x}^i \in \mathbb{R}^N (i = 1, \dots, m)$ を与え、アルゴリズムの規則によって、探索点の位置 \mathbf{x}^i を反復的に更新する。ここで、 $m \in \mathbb{N}$ を探索点数、 $i \in \{1, \dots, m\}$ を探索点番号といふ。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 $\mathbf{x}^i(k) (i = 1, \dots, m)$ の探索点群 $\mathcal{X}(k)$ は式 (2.12) で表される。

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

探索点群 $\mathcal{X}(k)$ に対応する目的関数値の集合 $\mathcal{F}(k)$ は式 (2.12) で表される。

$$\mathcal{F}(k) = \{f(\mathbf{x}^i(k)) | \mathbf{x}^i(k) \in \mathcal{X}(k)\} \quad (2.12)$$

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、反復回数 k までの探索点群 \mathcal{X} の探索履歴は式 (2.13) で表される。

$$\chi(k) = \{\mathcal{X}(1), \mathcal{X}(2), \dots, \mathcal{X}(k)\} = \{\mathcal{X}(\kappa)\}_{\kappa=0}^k \quad (2.13)$$

探索履歴 $\chi(k)$ に対応する目的関数値の集合 $\zeta(k)$ は式 (2.14) で表される。

$$\zeta(k) = \{\mathcal{F}(1), \mathcal{F}(2), \dots, \mathcal{F}(k)\} = \{\mathcal{F}(\kappa)\}_{\kappa=0}^k \quad (2.14)$$

数理最適化に分類されるアルゴリズムは、ほとんど単点探索型である一方、多点探索・直接探索法であるメタヒューリティクスは、探索点群の探索履歴を巧みに活用し、探索点群を更新させていく、良い解を求める。

(c) 決定的／確率的

探索点の移動過程で疑似乱数を利用してない最適化アルゴリズムを決定的, あるいは確定的という。決定的なアルゴリズムは, 同じ初期位置から探索を開始すれば, 探索過程で同じ軌道を辿り, 得る解も同じである。一方, 探索点の移動過程で疑似乱数を利用している最適化アルゴリズムを確定的という。確率的なアルゴリズムは, 同じ初期位置から探索を開始しても, 探索過程における軌道が異なり, 得る解も異なる。数理最適化に分類されるアルゴリズムの多くは決定的である一方, メタヒューリスティクスや大域的最適化の多くは確率的であり, 亂数を利用した摂動により局所解への収束を回避している³。

2.4.2 特殊な解・探索点

2.4.1 項で定義した探索点群や探索履歴を用いて, 最適化アルゴリズム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(2.15), 式(2.16), 式(2.17), 式(2.18)で定義する。

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

$$\mathbf{p}^i(k) = \underset{\mathbf{x}^i(\kappa) \in \chi^i(k)}{\operatorname{argmin}} \{f(\mathbf{x}^i(\kappa)) \mid \kappa = 0, \dots, k\} \quad (2.15)$$

多点探索において, 探索点群 $\mathcal{X}(k)$ に含まれる探索点 $\mathbf{x}^i(k)$ について, 反復回数 k までの探索履歴 $\chi^i(k) = \{\mathbf{x}^i(1), \dots, \mathbf{x}^i(k)\}$ の中で, 最も目的関数値が小さな解を p-best といい, $\mathbf{p}^i(k)$ と表す。また, 反復回数 k における p-best 集団を $\mathcal{P}(k) = \{\mathbf{p}^1(k), \dots, \mathbf{p}^m(k)\}$ と表す。

- g-best (global/group best solution : 最良解)

$$\mathbf{p}^g(k) = \underset{\mathbf{p}^i(k) \in \mathcal{P}(k)}{\operatorname{argmin}} \{f(\mathbf{p}^i(k)) \mid i = 1, \dots, m\} \quad (2.16)$$

多点探索において, 探索点群 $\mathcal{X}(k)$ の反復回数 k までの探索履歴 $\chi(k) = \{\mathcal{X}(0), \dots, \mathcal{X}(k)\}$ の中で, 最も目的関数値が小さな解を g-best といい, $\mathbf{p}^g(k)$ と表す。式(2.16)は,

³ Tabu Search [66, 67]は決定的なアルゴリズムだが, 探索履歴を巧みに活用することで局所解の脱出を実現している。

p-best 集団 $\mathcal{P}(k)$ の中で最も目的関数値が小さな解として定義しているため, g は $\mathcal{P}(k)$ の中の解の番号である。単点探索の場合は $m = 1$ のため, 常に $\mathbf{p}^g(k) = \mathbf{p}^i(k)$ である。

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

$$\mathbf{x}^b(k) = \underset{\mathbf{x}^i(k) \in \mathcal{X}(k)}{\operatorname{argmin}} \{f(\mathbf{x}^i(k)) \mid i = 1, \dots, m\} \quad (2.17)$$

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

- c-worst (current worst search point : 最悪探索点)

$$\mathbf{x}^w(k) = \underset{\mathbf{x}^i(k) \in \mathcal{X}(k)}{\operatorname{argmax}} \{f(\mathbf{x}^i(k)) \mid i = 1, \dots, m\} \quad (2.18)$$

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

2.5 数理最適化

2.5.1 数理最適化と最適化問題の関連性

数理最適化は, 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。数理最適化のアルゴリズムは, 目的関数の目的関数値情報や解析的情報（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。数理最適化では, 目的関数の解析的情報を要することから, 問題のクラス・性質毎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, Lagrange の未定乗数法, Karush-Kuhn-Tucker 条件を活用した解法, 双対問題を活用した解法が知られている[26, 27, 28, 31, 32]。数理最適化は, 目的関数, 制約条件関数の連続性・微分可能性が保証される場合にのみ適用可能である。

2.5.2 勾配法

(a) 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、単点型最適化アルゴリズムである。これらの解法では、式(2.9)における移動ベクトルをステップサイズを表すパラメータ $\alpha(k) \in \mathbb{R}$ 、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(2.19)と表すことができる。

$$x(k+1) = x(k) + \alpha(k)s(k) \quad (2.19)$$

式(2.19)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(2.20)の直線探索問題を解くことで決定する。

$$\alpha(k) = \underset{\alpha}{\operatorname{argmin}} \{f(x(k) + \alpha s(k)) \mid \alpha \geq 0\} \quad (2.20)$$

これは、 $x(k)$ の位置から $s(k)$ 方向において、最も目的関数値を小さくする α を求めている。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[31]。勾配法や、直接探索法の共役方向法[26]は降下法である。

(b) 勾配法

降下法の中でも、探索方向ベクトル s の生成に目的関数の勾配ベクトルや Hesse 行列を使用する解法を勾配法という。勾配法は凸最適化で最も使用される枠組みである。以降では、勾配法に分類される最適化アルゴリズムを概観する。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。

勾配法では、大域的収束性が保証されている。式(2.19)で表される勾配法の更新式により生成される点列 $\chi(k)$ について、Zoutendijk 条件に基づく定理が成り立つことが知られている[32]。これは、勾配法における直線探索問題において Wolfe 条件⁴を満たすように

⁴ Armijo 条件や曲率条件を合わせた条件。

$\alpha(k)$ を決定すると同時に、探索方向ベクトル s が降下方向（式 (A.10) の方向微分が負）となれば、Zoutendijk 条件（式 (2.21)，あるいは式 (2.22)）が成立するという定理である。

$$\sum_{k=0}^{\infty} \left(\frac{\nabla f(\mathbf{x}(k))^T s(k)}{\|s(k)\|_2} \right)^2 < \infty \quad (2.21)$$

$$\sum_{k=0}^{\infty} (\|\nabla f(\mathbf{x}(k))\|_2 \cos \theta(k))^2 < \infty \quad (2.22)$$

ただし、 $\nabla f(\mathbf{x}(k))$ は式 (A.8) の勾配ベクトルである。式 (2.22) の $\theta(k)$ は $-\nabla f(\mathbf{x}(k))$ 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。

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

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。

$$\lim_{k \rightarrow \infty} \frac{\nabla f(\mathbf{x}(k))^T s(k)}{\|s(k)\|_2} = 0 \quad (2.24)$$

$$\lim_{k \rightarrow \infty} \|\nabla f(\mathbf{x}(k))\|_2 \cos \theta(k) = 0 \quad (2.25)$$

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

勾配法では、目的関数 $f(\mathbf{x})$ 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 $f(\mathbf{x})$ を $\mathbf{x} = \mathbf{x}(k)$ 付近で Taylor 展開した式 (2.26) が用いられる。

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

ただし、 $\nabla f(\mathbf{x}(k))$ は式 (A.8) の勾配ベクトルであり、 $\mathbf{H}(f(\mathbf{x}(k)))$ は式 (A.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。

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

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

(c) 最急降下法

最急降下法は、探索方向ベクトル \mathbf{s} として、式 (2.27) の1次近似モデル $\ell(\mathbf{s})$ 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、 \mathbf{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 $-\nabla f(\mathbf{x}(k))$ と $\mathbf{s}(k)$ とのなす角について $\theta(k) = 0$ となることであるから、 \mathbf{s} は式 (2.29) で与えられる⁵。

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

最急降下法の探索方向は目的関数の降下方向となる。最急降下法は降下法の探索方向ベクトルとして目的関数の等高線に対する法線ベクトルを与えたアルゴリズムである。 \mathbf{s} が式 (2.29) で与えられたとき、Zoutendijk 条件を満たすため、最急降下法の点列は大域的収束性が保証されている。

付録 C の Sphere 関数のような、良スケール性・変数分離可能性・単峰性を有する目的関数の等高線は、座標系に直交かつ真円となる。このような等高線に対する最急降下法の降下方向は、円の中心に向かうベクトルとなる。しかしながら、Sphere 関数に $\mathbf{x} \mapsto \mathbf{B}\mathbf{D}\mathbf{x}$ (\mathbf{D} は対角行列、 \mathbf{B} は正規直交行列) という変換を作用させると、Schwefel 関数のように等高線が座標系に直交でない楕円となり、悪スケール性・変数間依存性が発生する。このとき、最急降下法の降下方向は、座標系に直交な方向にしか生成できず、効率の悪い探索となる。この現象については、最適化アルゴリズムのロバスト性と直結するため、2.10 節にて詳細を述べる。

Algorithm 2.5.2 に最急降下法の疑似コードを示す。

(d) Newton-Raphson 法

Newton-Raphson 法は、式 (2.28) の2次近似モデル $q(\mathbf{s})$ の最小化に基づく勾配法である。最適性条件より、 $q(\mathbf{s})$ を最小化するためには、 $\nabla q(\mathbf{s}) = \mathbf{0}$ を満たすベクトル \mathbf{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 \mathbf{s} を Newton 方向という。Newton-Raphson 法は、探索方向ベクトル $\mathbf{s}(k)$ を Newton 方向に設定する。

⁵ 実際には、 $\mathbf{s}(k) = -\frac{\nabla f(\mathbf{x}(k))}{\|\nabla f(\mathbf{x}(k))\|}$ として正規化した後の勾配を使用することが多い。

Algorithm 1 最急降下法

1: procedure GRADIENT DESCENT	テップサイズ $\alpha(k)$ を求める。
Step 0 : パラメータ設定	Step 4 : 探索点の更新
2: 停止基準 $\varepsilon > 0$ を設定する。	7: 式 (2.19) を用いて、探索点 $\mathbf{x}(k)$ を更新する。
Step 1 : 初期化	Step 5 : 終了判定
3: 探索点の初期位置 $\mathbf{x}(k)$ を与える。	8: if $\ \nabla f(\mathbf{x}(k))\ _2 < \varepsilon$ then
4: $k = 1$ とする。	9: 計算を終了する。
Step 2 : 降下方向の決定	10: else
5: 式 (2.29) を用いて、 $\mathbf{x}(k)$ における降下方向 $\mathbf{s}(k)$ を決定する。	11: $k := k + 1$ とし、 Step 2 へ戻る。
Step 3 : ステップサイズの決定	12: end if
6: 式 (2.20) の直線探索問題を解き、ス	13: end procedure

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

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

ただし、 $\mathbf{H}(f(\mathbf{x}))$ は、式 (A.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。

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

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

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

ただし、Hesse 行列 $\mathbf{H}(f(\mathbf{x}(k)))$ は正則であるとする。さらに、Hesse 行列 $\mathbf{H}(f(\mathbf{x}(k)))$ 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。

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

これは、Hesse行列が正定値ならば方向微分が負となることを表しているため、Newton 方向は目的関数の降下方向となる。

また、Newton 方向は、点 $\mathbf{x}(k)$ において目的関数の等高線の接線と曲率を共有する機能を考え、点 $\mathbf{x}(k)$ から機能の中心へ向かう方向ベクトルとなる。例えば、付録 C の Sphere 関数のような、良スケール性・変数分離可能性・単峰性を有する目的関数に対して $\mathbf{x} \mapsto \mathbf{B}\mathbf{D}\mathbf{x}$ (\mathbf{D} は対角行列、 \mathbf{B} は正規直交行列) という変換を作用させると、Schwefel 関数のように等高線が座標系に直交でない機能となり、悪スケール性・変数間依存性が発生する。このとき、最急降下法では探索効率が悪かったが、Newton 方向はその機能の中心へ向かうベクトルとなるため、Sphere 関数に対する降下方向と同様の効率を發揮できる。この現象については、最適化アルゴリズムのロバスト性と直結するため、2.10 節にて詳細を述べる。

Newton-Raphson 法の点列は 2 次局所的収束性が保証されている[32]。そのため、収束速度の点で優れた代表的な最適化アルゴリズムである。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、最急降下法のアルゴリズム Algorithm 2.5.2 における Step 2 において、探索方向ベクトル s を式 (2.32) で決定したものである。

(e) 共役勾配法

共役勾配法は共役勾配ベクトルを降下方向とした勾配法である。元々、連立 1 次方程式の解法として 1952 年に Hestenes と Stiefel によって提案されたもので、一般の非線型関数最小化問題への適用は 1964 年に Fletcher と Reeves によりはじめて行われた。共役勾配法は収束が速く、アルゴリズムが簡単で、記憶容量も少ない等の特長を有している。基本的な考え方は、互いに共役な探索ベクトルを次々に生成し、共役方向に沿って探索を繰り返すことにある。

正定値対称行列 \mathbf{Q} に対して \mathbf{x}, \mathbf{y} が $\mathbf{x}^T \mathbf{Q} \mathbf{y} = 0$ を満たすとき、 \mathbf{x}, \mathbf{y} を \mathbf{Q} に関して互いに共役という。 \mathbf{Q} が単位行列 \mathbf{I} のとき、共役は直交と等価である。共役勾配ベクトル \mathbf{m} は式 (2.34) で表される。

$$\mathbf{m}(k) = \nabla f(\mathbf{x}(k)) + \beta \mathbf{t}(k) \quad (2.34)$$

ここで、 $\mathbf{t}(k)$ は $\mathbf{x}(k)$ における目的関数の等高線の接ベクトルであり、 $\beta > 0$ は以下で導

出する。探索方向は共役勾配方向だから、 $s \propto \mathbf{m}$ である。また、式(2.32)のNewton方程式から、 $\mathbf{H}(f(\mathbf{x}(k)))\mathbf{m} \propto \nabla f(\mathbf{x}(k))$ となる。そして、接ベクトル $\mathbf{t}(k)$ と勾配ベクトル $\nabla f(\mathbf{x}(k))$ は直交するから、 $\mathbf{t}^T(k)\nabla f(\mathbf{x}(k)) = 0$ となるから、 $\mathbf{t}^T(k)\mathbf{H}(f(\mathbf{x}(k)))\mathbf{m} = 0$ となる。つまり、式(2.34)の \mathbf{m} を代入して変形すると、共役勾配ベクトルを決定づける β が式(2.35)として得られる。

$$\beta(k) = -\frac{\mathbf{t}^T(k)\mathbf{H}(f(\mathbf{x}(k)))\nabla f(\mathbf{x}(k))}{\mathbf{t}^T(k)\mathbf{H}(f(\mathbf{x}(k)))\mathbf{t}(k)} \quad (2.35)$$

さらに、 $\mathbf{m}(k)$ 方向に直線探索していくと、次回の接ベクトル $\mathbf{t}(k+1)$ 方向と一致することがわかる。言い換えると、接ベクトル $\mathbf{t}(k)$ は前回の共役勾配ベクトル $\mathbf{m}(k-1)$ である。そこで、初期の接ベクトル $\mathbf{t}(1) = \nabla f(\mathbf{x}(1))$ とし、それ以降の探索は $\mathbf{t}(k) = \mathbf{m}(k-1)$ とすれば、共役勾配ベクトルの更新式として式(2.36)が得られる。

$$\mathbf{m}(k) = \nabla f(\mathbf{x}(k)) - \frac{\mathbf{m}^T(k-1)\mathbf{H}(f(\mathbf{x}(k)))\nabla f(\mathbf{x}(k))}{\mathbf{m}^T(k-1)\mathbf{H}(f(\mathbf{x}(k)))\mathbf{m}(k-1)}\mathbf{m}(k-1) \quad (2.36)$$

$\mathbf{t}^T(k)\mathbf{H}(f(\mathbf{x}(k)))\mathbf{m} = 0$ においても同様に置き換えると、 \mathbf{m} はHesse行列 $\mathbf{H}(f(\mathbf{x}(k)))$ に関する接ベクトル $\mathbf{m}(k-1)$ と共役となることを示している。

目的関数が N 次元の付録CのSphere関数のような2次関数であれば、高々 N 回の反復で最適解に到達する[31]。通常の問題に対しては、後に述べる準Newton法の方が一般に有効であると言われているが（通常、準Newton法の2倍程度の計算時間がかかるとされている）、問題によっては共役勾配法の方が効果的である場合もあることが報告されている。どのようなときに共役勾配法が他の手法に比べて有効であるかについてはまだ明らかにされておらず、現在のところ実際に数値実験を行ってみる以外に方法はない。また、初期下降ベクトル（最初の探索方向）の選択方法や、一般の非線型関数に適用する際の再出発の頻度あるいは1次元探索に要求される精度等に関しても不明な点が多く、現在なお研究が行われている。なお、式(2.35)にはHesse行列が含まれているため、 β を勾配ベクトルだけを用いて近似的に決定する方法が知られており、Fletcher-Reevesの公式などがある[31]。

(f) 準Newton法

準Newton法[26, 27]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。元々、1959 年に Davidon により「Variable Metric Method for Minimization (可変計量法)」として提案され、その後、Fletcher と Powell によりさらに改良が加えられ、今日良く知られた DFP (Davidon-Fletcher-Powell) 公式の開発に至った。

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。以降では、Hesse 行列を正定値対称行列 $\mathbf{B}(k)$ で近似する場合を考える。式 (2.28) の 2 次近似モデル $q(\mathbf{s})$ における Hesse 行列を正定値対称行列 $\mathbf{B}(k) \in \mathbb{R}^{N \times N}$ に置き換えた、式 (2.37) で表される新たな 2 次近似モデル $Q(\mathbf{s})$ を考える。

$$Q(\mathbf{s}) = f(\mathbf{x}(k)) + \nabla f(\mathbf{x}(k))^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{B}(k) \mathbf{s} \quad (2.37)$$

Newton-Raphson 法と同様に、2 次近似モデル $Q(\mathbf{s})$ 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 \mathbf{s} は式 (2.38) で与えられる。

$$\mathbf{s}(k) = -\mathbf{B}(k)^{-1} \nabla f(\mathbf{x}(k)) \quad (2.38)$$

$\mathbf{B}(k)$ 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、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、 $\mathbf{B}(k)$ は正則行列となる（逆行列が存在する）。

$\mathbf{B}(k+1) \in \mathbb{R}^{N \times N}$ 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[32]。セカント条件は式 (2.44) で与えられる。準 Newton 法では、式 (2.45) のセカント条件を満たすように $\mathbf{B}(k)$ を更新して、 $\mathbf{B}(k+1)$ が生成される。セカント条件を満たす $\mathbf{B}(k+1)$ 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 [26, 27, 28, 32]。Hesse 行列の逆行列を近似する場合は、正定値対称行列 $\mathbf{B}(k)$ 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。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 [26, 32]。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。また、各探索ベクトルは共役勾配法と同様に互いに共役であり、目的関数が N 次元の付録 C の

Algorithm 2 Hesse 行列を近似する準 Newton 法

```

1: procedure QUASI-NEWTON METHOD      Step 4 : 探索点の更新
   Step 0 : パラメータ設定           8: 式 (2.19) を用いて、探索点  $\mathbf{x}(k)$  を更
2:    停止基準  $\varepsilon > 0$  を設定する。       新する。
   Step 1 : 初期化                  Step 4 : 近似行列の更新
3:    探索点の初期位置  $\mathbf{x}(k)$  を与える。  9: 公式を用いて、近似行列  $\mathbf{B}(k)$  を更新
4:    近似行列を  $\mathbf{B}(k) = \mathbf{I}$  と与える。       する。
5:     $k = 1$  とする。                  Step 5 : 終了判定
   Step 2 : 降下方向の決定          10: if  $\|\nabla f(\mathbf{x}(k))\|_2 < \varepsilon$  then
6:    式 (2.38) を用いて、 $\mathbf{x}(k)$  における降 11: 計算を終了する。
   下方向  $\mathbf{s}(k)$  を決定する。        12: else
   Step 3 : ステップサイズの決定      13:    $k := k + 1$  とし、Step 2 へ戻る。
7:    式 (2.20) の直線探索問題を解き、ス 14: end if
   テップサイズ  $\alpha(k)$  を求める。       15: end procedure

```

Sphere 関数のような 2 次関数であれば、高々 N 回の反復で最適解に到達する[31]。つまり、準 Newton 法は、Newton-Paphson 法と共に勾配法の優れた点をうまく組み合わせた手法と言うことができ、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。この現象については、最適化アルゴリズムのロバスト性と直結するため、2.10 節にて詳細を述べる。

Algorithm 2.5.2 に準 Newton 法の疑似コードを示す。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 $\mathbf{B}(k)$ を初期化するアルゴリズムも存在する。

2.6 大域的最適化

2.6.1 確率的勾配降下法

確率的勾配降下法 (Stochastic Gradient Descent : SGD) は、機械学習などのモデリングで使用される大域的最適化アルゴリズムである。ある関数 $H(\mathbf{p})$ のモデル $\hat{H}(\mathbf{p}; \mathbf{x})$ を L 個のデータ $\{\mathbf{p}_\ell\}_{\ell=1}^L$ を用いてフィッティングするように、モデルパラメータ \mathbf{x} を決定する問題を考える。この問題は、統計学やモデリングにおける最小二乗問題や最尤推定問題などとして頻出する最適化問題である。この問題の目的関数は、モデルと対象関数の間の誤差、具体的には各データ \mathbf{p}_ℓ を用いた誤差 $f_\ell(\mathbf{x})$ の和の形 $f(\mathbf{x}) = \sum_{\ell=1}^L f_\ell(\mathbf{x})$ として与えられ、この誤差を最小化するようなモデルパラメータ \mathbf{x} を求める。中でも、誤差関数を $f(\mathbf{x}; \mathbf{p}_\ell) = \sum_{\ell=1}^L \|H(\mathbf{p}_\ell) - \hat{H}(\mathbf{x}; \mathbf{p}_\ell)\|_2^2$ として表してから解くのが最小二乗法である⁶。なお、この問題は2次関数のように見えるが、一般的にそうではないことに注意されたい。例えば、学習データ $\{\mathbf{p}_\ell, y_\ell\}_{\ell=1}^L$ を用いた教師有り学習において、モデルが線型モデル $\hat{H}(\mathbf{p}; \mathbf{x}) = \mathbf{x}^\top \mathbf{p}$ として与えられる場合、最小二乗法の目的関数は $f(\mathbf{x}; \mathbf{p}_\ell) = \sum_{\ell=1}^L \|y_\ell - \mathbf{x}^\top \mathbf{p}_\ell\|_2^2$ となるため、目的関数は \mathbf{x} について線型だが、出力自身にさらに非線型変換 h が与えられる、つまりモデルが非線型モデル $\hat{H}(\mathbf{p}; \mathbf{x}) = h(\mathbf{x}^\top \mathbf{p})$ として与えられる場合、目的関数は $f(\mathbf{x}; \mathbf{p}_\ell) = \sum_{\ell=1}^L \|y_\ell - h(\mathbf{x}^\top \mathbf{p}_\ell)\|_2^2$ となるため、目的関数は \mathbf{x} について非線型となる。

このとき、最急降下法における目的関数の勾配ベクトルは、全てのデータを用いて、 $\nabla f(\mathbf{x}) = \sum_{\ell=1}^L \nabla f_\ell(\mathbf{x})$ と表される（バッチ学習）。これに対して、SGD では各反復でランダムに選択したデータを用いて、 $\nabla f(\mathbf{x}) \simeq \nabla f_\ell(\mathbf{x})$ として導出する（オンライン学習）⁷。このため、SGD は勾配ベクトルの計算量を削減しながら、ランダム性を含むため、局所的最適解への収束の強さを緩和できる。

また、代表的な機械学習であるニューラルネットワーク (Neural Network : NN) におけるモデルの学習では、SGD と誤差逆伝搬法 (Back Propagation) を使用して、最小二乗法

⁶ なお、この目的関数内の $\hat{H}(\mathbf{x}; \mathbf{p}_\ell)$ という表記は、;の後ろ側がパラメータ（母数）であるという確率統計で用いられる表記だが、このように使用しているのは、モデリングの観点では \mathbf{x} がパラメータで、データ \mathbf{p} が説明変数だが、最適化問題の観点ではデータ \mathbf{p}_ℓ がパラメータで、モデルパラメータ \mathbf{x} が決定変数となるためである。

⁷ バッチ学習とオンライン学習の間をとって、一部のデータをランダムに選択して、勾配ベクトルを計算する学習方法をミニバッチ学習という。

形式の問題を解くことでモデルパラメータを求める。NNは、入力層と出力層の間に中間層を導入し、中間層において入力層の出力同士の線形結合と非線型変換（活性化）を施す、より柔軟なモデリングを可能にした非線型モデルである。このため、NNの最小二乗法では、非線型最適化問題を解いている。NNはその柔軟なモデリング能力によって、1980年代以降の第2次人工知能（AI）ブームを牽引したが、様々な要因で下火となった。

さらに、2010年頃に深層学習（Deep Learning, Deep Neural Network）が登場し、画像・音声分野で成功とする同時に、2010年以降の第3次AIブームを起こした。それ以降も深層学習は飛躍的に発展し、2020年時点より広い分野への適用が検討されていることから、このブームを牽引しているのは明らかである。中間層が1層のNNに比べ、深層学習は中間層をより深くすることで表現力をより高めているため、学習問題がかなり複雑である。よって、通常のSGDは局所的最適解に収束する、鞍点付近で収束が不安定になる、目的関数の陵構造（Pathological Curvature）付近で探索点の軌道が振動する、などが原因で、学習が不十分となり、第2次AIブームでは形式としては認知されていたが、使用されていなかった。これを受け、Momentum法、AdaGrad（2011年）、RMSProp（2012年）、Adam（2014年）、Eve（2016年）などの様々なSGDベースの最適化アルゴリズムが提案され、その課題を解消したことから、深層学習の成功は、SGDベースの最適化アルゴリズムの貢献が大きい⁸。よって、これらの方は勾配を利用することが前提であるため、BBOでは利用できないが、大域的最適化としては期待される。

さらに、深層学習におけるSGDベースの最適化アルゴリズムは慣性項やパラメータ調整則の追加など、メタヒューリスティクスにおけるアイディアと類似するものが多い。例えば、Momentum法の更新式は式(2.39)、式(2.40)で表される。

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \alpha \boldsymbol{\nu}(k+1) \quad (2.39)$$

$$\boldsymbol{\nu}(k+1) = \beta \boldsymbol{\nu}(k) + (1 - \beta) \nabla f(\mathbf{x}(k)) \quad (2.40)$$

ただし、 α, β はパラメータである。通常のSGD（最急降下法）は式(2.39)の $\boldsymbol{\nu}(k+1)$ が勾配ベクトル $\nabla f(\mathbf{x}(k))$ に置き換えた式である。一方、式(2.40)の $\beta \boldsymbol{\nu}$ は慣性項と呼ばれ、

⁸ 最適化アルゴリズムに限らず、新たな活性化関数である正規化線形関数（Rectified Linear Unit : Relu関数、ランプ関数ともいう）やバッチ正規化の提案など、同時期に様々な研究成果が融合したことが最も大きな貢献である。

式(2.40)で更新されていくが、これは過去の勾配情報と現在の勾配情報を合わせたものを移動ベクトルとする、という意味である。このため、Momentum法における慣性項は、現在の勾配情報（正負の符号）だけに依存しないため、正負で振動していると打ち消しあって振動が小さくなる効果がある。この慣性項はメタヒューリスティクスのParticle Swarm Optimizationにおいても同様に導入されている。また、AdaGradの更新式は式(2.41)、式(2.42)、式(2.43)で表される。

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \alpha(k+1) \odot \nabla f(\mathbf{x}(k)) \quad (2.41)$$

$$\alpha_n(k+1) = \frac{\alpha_{\text{start}}}{\sqrt{r_n(k+1)}} \quad (2.42)$$

$$\mathbf{r}(k+1) = \mathbf{r}(k) + \nabla f(\mathbf{x}(k)) \odot \nabla f(\mathbf{x}(k)) \quad (2.43)$$

ただし、 α_{start} はパラメータである。なお、 \odot はHadamard積(Entrywise Product)で、 $\mathbf{x} \odot \mathbf{y} = [x_1 y_1, \dots, x_N y_N]^T$ である。通常のSGD(最急降下法)は係数のパラメータがスカラだが、式(2.41)ではベクトルであり、式(2.42)と式(2.43)によって適応的に調整される。このため、通常のSGD(最急降下法)は勾配が急な方向には早く収束するが、勾配が緩やかな方向には収束に時間がかかることが起り得ることに対して、AdaGradは要素毎に勾配のスケールを適応的に調整することで収束方向を工夫している。これは、メタヒューリスティクスのEvolution Strategyの適応的パラメータ調整則においてもパラメータを多次元にし、それを適応的に調整する工夫がみられる。このように、メタヒューリスティクスと深層学習のSGDベースの最適化アルゴリズムは非凸性の対処方法が独自に発展してきたが、これまであまり融合された事例はほとんどない。したがって、より効果的な最適化アルゴリズムを設計する上で、これらは相互に貢献することが期待される。

2.6.2 力学系モデルに基づく大域的最適化手法

統計力学やカオス現象などで見られる、系のエネルギー・エントロピーなどのスカラ量の挙動を記述する力学系モデルを、大域的最適化の計算モデルとして利用したアルゴリズムを、力学系モデル、あるいはPhysical-Inspiredアルゴリズム⁹という。本論文では詳細

⁹ メタヒューリスティクスなどを表すNature-Inspiredという呼称と対応させている。

を述べないが、勾配を使用できる場合は強力な大域的最適化アルゴリズムである。詳細は [135] を参考されたい。

2.7 直接探索法

2.7.1 古典的な直接探索法

本節では、古典的な直接探索法として、滑降シンプレックス法[26, 45]と Hook-Jeeves 法[43, 44]を挙げる。他の古典的な直接探索法には、共役方向法の一種である Powell 法[26]、ランダムサーチ[49]などがある。Python のモジュール SciPy におけるライブラリ optimize には、直接探索法用アルゴリズムとして滑降シンプレックス法と Powell 法の改良アルゴリズムが実装されている[46]。

(a) 滑降シンプレックス法

滑降シンプレックス法 (Nelder-Mead 法) は古典的な直接探索法のアルゴリズムである。探索は、鏡映操作、拡張・縮小操作の二段階から構成される。探索空間 \mathbb{R}^N において、 $N+1$ 個の頂点を有する単体（シンプレックス）を形成した後、最悪探索点を動かしながらシンプレックスを拡張・縮小させることで、有望な領域に徐々に収束させていく。

鏡映操作の更新式は式 (2.44)、式 (2.45) で表される。

$$\mathbf{y}(k) = \mathbf{x}^w(k) + \alpha_r(\mathbf{g} - \mathbf{x}^w(k)) \quad (2.44)$$

$$\mathbf{g} = \frac{1}{m-1} \sum_{i=1; i \neq w}^m \mathbf{x}^i(k) \quad (2.45)$$

ただし、 $i = 1, \dots, m$ 、 $m = N+1$ 、 $\mathbf{x}^w(k)$ は式 (2.18) の最悪探索点、 \mathbf{g} は $\mathbf{x}^w(k)$ を除いた探索点群の重心、 α_r はスケールパラメータである。

次に、拡張・縮小操作について述べる。(1) もし $f(\mathbf{x}^b(k)) \leq f(\mathbf{y}) < f(\mathbf{x}^w)$ ならば、 $\mathbf{x}^w(k+1) = \mathbf{y}$ とする。(2) もし $f(\mathbf{y}) < f(\mathbf{x}^b(k))$ ならば、拡張操作を行う。(3) (1), (2) でなければ、縮小操作を行う。ただし、 $\mathbf{x}^b(k)$ は式 (2.17) の最良探索点である。

拡張操作は、式(2.46)、式(2.47)で行われる。

$$\mathbf{u} = \mathbf{g} + \alpha_e(\mathbf{y} - \mathbf{g}) \quad (2.46)$$

$$\mathbf{x}^w(k+1) = \begin{cases} \mathbf{u}, f(\mathbf{u}) < f(\mathbf{v}) \\ \mathbf{y}, \text{otherwise} \end{cases} \quad (2.47)$$

ただし、 α_e はスケールパラメータである。

縮小操作は、式(2.48)で行われる。

$$\mathbf{u} = \mathbf{g} + \alpha_c(\mathbf{g} - \mathbf{x}^w(k)) \quad (2.48)$$

ただし、 α_c はスケールパラメータである。もし、 $f(\mathbf{u}) < f(\mathbf{x}^w(k))$ ならば、 $\mathbf{x}^w(k+1) = \mathbf{u}$ とする。そうでなければ、式(2.49)で全ての探索点を更新する。

$$\mathbf{x}^i(k+1) = \frac{1}{2}(\mathbf{x}^b(k) + \mathbf{x}^i(k)) \quad (2.49)$$

(b) Hooke-Jeeves 法

Hooke-Jeeves 法は古典的な直接探索法のアルゴリズムである。Explanatory Search Step と Pattern Move Step の二段階の探索を交互に繰り返す。Explanatory Search Step における更新は、各座標軸に沿って候補解を生成し、改善する場合のみ移動する。座標軸毎に、正の方向の改善を試み、改善しない場合は負の方向の改善を試みる。正方向の更新式は式(2.50)、式(2.51)で表され、負方向の更新式は式(2.50)の \mathbf{e}_n を $-\mathbf{e}_n$ と置き換えたものである。

$$\mathbf{y} = \mathbf{x} + \alpha \mathbf{e}_n \quad (2.50)$$

$$\mathbf{x} := \begin{cases} \mathbf{y}, f(\mathbf{y}) < f(\mathbf{x}) \\ \mathbf{x}, \text{otherwise} \end{cases} \quad (2.51)$$

ただし、標準基底 $E_0 = \{\mathbf{e}_1, \dots, \mathbf{e}_N\}$ 、 α はステップ幅であり、対角行列 $\text{diag}[\alpha_1, \dots, \alpha_N]^T$ と設定する場合もある。

Pattern Move Stepにおける更新は、差分ベクトルに沿って候補解を生成し、改善する場合のみ移動する。そうでなければ、 α を更新する。更新式は式(2.52)で表される。

$$\begin{cases} \mathbf{x} := \mathbf{x} + (\mathbf{x} - \mathbf{u}), f(\mathbf{x}) < f(\mathbf{u}) \\ \alpha := \beta\alpha, \text{otherwise} \end{cases} \quad (2.52)$$

ただし、 \mathbf{u} はExplanatory Search Step直前の \mathbf{x} 、 β は α 調整係数パラメータであり、 $\beta = 0.5$ が推奨値である。

滑降シンプレックス法と Hook-Jeeves 法などの古典的な直接探索法では、探索点の次の移動候補（近傍）の生成において、探索過程で得た特殊な解や差分ベクトルを活用することで、目的関数の解析的情報を用いずに、目的関数の景観を把握しようとしている。一方、これらのアルゴリズムは、次元数の増加に伴い探索性能が劣る上に、大域的最適化で有効とは言えない[48]。ランダムサーチは、ランダム性を利用することで局所的最適解に収束しにくくなり、無限に計算時間を与えた場合に大域的最適解に収束する保証がある一方、その最適性が保証される条件で適用されることは困難であると同時に、近傍生成において上記の工夫がないため、効率的な探索を行うことができない。以上のように、古典的な直接探索法は実問題に適用するには性能が不十分だが、メタヒューリスティクスのアルゴリズム設計において活用できるアイディアも多い。実際に、メタヒューリスティクスの設計や改良において、滑降シンプレックス法における操作や探索構造を活用している例もある[136, 137]。

2.7.2 統計力学に基づくサンプリング

Markov 連鎖モンテカルロ法 (Markov Chain Monte Carlo Method : MCMC) は多変量確率分布からのサンプリング方法である。元々 MCMC は、1960 頃から統計物理分野で多くの分子からなる集団の相転移や臨界現象をシミュレーションするための計算方法として誕生し、それ以降は統計科学の一手法として発展したが、その一部の方法は直接探索法との関連が深いため、簡潔に概観しておく。モンテカルロ法自体は多変量確率分布が与えられたとき、その分布に従う点をサンプリングする、その分布の期待値を近似的に計算する、などのために使用される。しかしながら、単純なモンテカルロ法は一様なランダムサンプリングであるため、非効率になることが多いなどの問題がある。また、多変量確率分

布の逆関数を求めるることは難しい上に、元々多変量確率分布が明示的に与えられることは少ない。そこで、Metropolis-Hastings 法 (MH 法) は重要な箇所を重点的にサンプリングすることで効率化を図っている。MH 法では、Markov 連鎖という確率過程を用いて、逐次的にサンプリングすることを考える。Markov 連鎖とは、直前の点（状態）のみに依存した確率分布に従うランダムウォークで生成される点列を仮定した確率過程である。結果的に、Markov 連鎖で生成される点列が目標である確率分布に従えば良いから、構成した分布関数に収束するように点（状態）の遷移確率を上手く決定すればよい。確率分布が定常分布となる（詳細つり合い条件），そしてエルゴード性を満たすという条件を与えれば、任意の初期状態から生成されるマルコフ連鎖から構成される事後分布が唯一の定常分布に収束することが示される。この点列から一定間隔を空けて抜き出した点集合を独立なサンプルとしてみなし、そこから統計量を計算する。このように、Markov 連鎖のもとで Bayes 推定のように事後分布を逐次的に生成していき、確率分布からサンプリングするという方法が MCMC の本来の特徴である。また、Bayes 推定は元々事後確率分布の期待値を求みたいが、高次元の積分計算が含まれるために、MCMC はその計算を解析的に取り扱わずに近似的に解くことが可能である。よって、MCMC の応用範囲が非常に広いことがわかる。

MCMC の方法は代替サンプリング、Gibbs サンプラー¹⁰（熱浴法）、MH 法に大別される。代替サンプリングはあまり使用されず、MH 法は Gibbs サンプラーを一般化した方法であり、最も使用されるため、本項では MH 法の説明に限定する。目標の確率分布 $\mathcal{P}(x)$ に従ってサンプル点を得ることを考える。状態空間 Ω 内の状態 $x \in \Omega$ があるとし、2 つの状態間の遷移確率を $\mathcal{P}(x(k+1)|x(k))$ とする。MH 法では、状態遷移確率を式 (2.53) とする。

$$\mathcal{P}(x(k+1)|x(k)) = g(x(k+1)|x(k))A(x(k+1)|x(k)) \quad (2.53)$$

ここで、 $g(x(k+1)|x(k))$ は提案分布で状態 $x(k)$ が与えられたときに状態 $x(k+1)$ を提案する条件確率分布、 $A((x(k+1)|x(k)))$ は採択確率で状態 $x(k)$ が与えられたときに状態 $x(k+1)$ を採択する条件確率分布である。ここで、詳細つり合い条件は式 (2.54) である。

$$\mathcal{P}(x(k))\mathcal{P}(x(k+1)|x(k)) = \mathcal{P}(x(k+1))\mathcal{P}(x(k)|x(k+1)) \quad (2.54)$$

¹⁰ 原理的には「Bayesian サンプラー」が正しいが、慣用的に「Gibbs サンプラー」という名称が使用されることが多い。

式(2.54)の詳細つり合い条件を満たすような採択確率は、Metropolis選択が知られている。Metropolis選択における採択確率は式(2.56)である。

$$A(x(k+1)|x(k)) = \min\left\{1, \frac{\mathcal{P}(x(k+1))}{\mathcal{P}(x(k))} \frac{g(x(k)|x(k+1))}{g(x(k+1)|x(k))}\right\} \quad (2.55)$$

式(2.56)の採択確率は、式(2.54)の詳細つり合い条件を満たすが、本論文では割愛する。したがって、MH法の計算手順は下記の通りに整理できる。

- ステップ1 ランダムに $x(1) \in \Omega$ を設定する ($k = 1$ とする)。
- ステップ2 $g(\hat{x}(k)|x(k))$ に従い $\hat{x}(k)$ を生成する。
- ステップ3 $A(x(k+1)|x(k))$ に従い $x(k+1) = \hat{x}$ として遷移する。採択しなければ $x(k+1) = x(k)$ として留まる (棄却する)。
- ステップ4 終了条件を満たす場合は終了し、そうでなれば $k := k + 1$ として②に戻る。

このように、採択確率を満たす場合のみ状態を遷移する動作を繰り返していく。

MCMCでは、目標の確率分布として Boltzmann 分布を仮定することが多い。Boltzmann 分布は統計力学においてミクロな集団から構成される統計集団のマクロな状態を表すときに使用する分布関数である。具体的には、統計集団（カノニカル集団）が熱平衡状態にあるとき、微視的状態 x が実現される確率は式(2.56)として表される。

$$\mathcal{P}(x; T) = \frac{1}{Z} \exp\left(-\frac{E(x)}{k_B T}\right) \quad (2.56)$$

ただし、 $E(x)$ は微視的状態 x のエネルギー、 k_B は Boltzmann 定数、 T は温度、 Z は分配関数である。特に、式(2.56)中の $\exp(-\frac{E(x)}{k_B T})$ が Boltzmann 分布だが、カノニカル集団の際には Boltzmann 因子と呼ばれる。また、分配関数 Z は確率の正規化定数の役割であるため、全ての微視的状態について Boltzmann 因子の総和をとったものだから、式(2.57)として表される。

$$Z = \sum_{x \in \Omega} \exp\left(-\frac{E(x)}{k_B T}\right) \quad (2.57)$$

式(2.56)は、エネルギー $E(x)$ と温度 T によって、状態 x が実現する確率が変わることを表している。さらに、物理学においては元々、目的関数をエネルギー関数とみなし、そのエネルギーを最小化するような考えがある。この Boltzmann 分布も最適化と関連させると、

エネルギー $E(x)$ を目的関数値 $f(x)$ と置き換えることができ、Boltzmann 定数 k_B は物理的意味があるが、最適化では意味がないため、 $k_B = 1$ とおくと、式(2.56)は式(2.58)として書き換えられる。

$$\mathcal{P}(x; T) = \frac{1}{Z} \exp\left(-\frac{f(x)}{T}\right) \quad (2.58)$$

これは、目的関数値が小さい箇所の確率が高くなる分布である。つまり、Boltzmann 分布を仮定した MH 法は、目的関数値が小さい箇所に重点的にサンプリングするような方法である。また、採択確率の式では、確率の比として利用されるため、分配関数 Z は消去され、実際には計算する必要はなく、目的関数値と温度だけで計算される。これらの仮定により、M-H 法を始めとする MCMC は最適化として利用できる形となる。

MCMC ベースのサンプリングによる最適化は、直接探索の枠組みのもとで、比較的計算量が抑えられると同時に、収束性が保証される点でメリットがある。上記の形はメタヒューリスティクスの一種である Simulated Annealing (SA) でも同様にみられる。ただし、SA では温度パラメータ T を探索序盤では高いため（活性状態）から、探索過程で徐々に下げる（冷却する）ことで、多様化・集中化の探索戦略を実現する。つまり、MCMC ベースだが、多峰性を仮定した最適化アルゴリズムとして設計されている。また、目的関数に関する知識をサンプルから学習し、提案分布の設計に活かすのは分布推定アルゴリズム (Estimation of Distribution Algorithm : EDA) [174] と類似している。EDA はメタヒューリスティクスの一種である実数値 Genetic Algorithm の一種としても捉えられることがある。このように、MCMC ベースのサンプリングによる最適化は、メタヒューリスティクスとの関連性は深い。

他にも集団モンテカルロ法がある。上記に挙げた M-H 法や SA は基本的に単点を想定したものだが、MCMC を集団で行う（多点探索）方法を集団モンテカルロ法、拡張アンサンブル法と呼ぶ。中でもパラレル・テンパリング法 (Parallel Tempering, レプリカ交換法 : Replica Exchange Method) が有名である。パラレル・テンパリング法は、異なる温度パラメータを有する探索点が複数あり、それぞれ独立に MCMC に従ってランダムウォークするが、途中で交換確率を満たした場合に、探索点間で温度パラメータを交換する方法である。SA が時系列で温度パラメータを下げることで多様化から集中化へ移行していたことに対して、パラレル・テンパリング法は探索点毎に多様化・集中化の探索状態が異なり、それらが交換することで相互作用を生み出すという点で異なる。他にも、拡張アンサンブル

ル法では、MCMC の発展形として、重点サンプリング法、リサンプリング、マルチカノニカル法などがある。集団モンテカルロ法は単なるサンプリング法として見られることが多い、メタヒューリスティクスと関連させた最適化アルゴリズムはほとんど事例がないが、多点探索という点でメタヒューリスティクスと親和性が高く、相互に発展する上で重要な知見の一つとして考えられる。

2.7.3 メタヒューリスティクス

メタヒューリスティクスは、生物現象等の経験的に優れたメカニズムが実装された直接探索法・大域的最適化アルゴリズムである[50, 51, 52, 53, 54, 55, 56]。これまで概観してきたアルゴリズムと大きく異なる点は、その最適化能力の設計が自然界などの優れたメカニズムから発想を得ていることである。メタヒューリスティクスという言葉は、上位を表すメタという接頭語（Meta）と発見的手法（Heuristics）から構成されている。このため、発見的操作の上位に、有力な上位概念を付け加えることで、アルゴリズム全体の秩序を形成していると考えられる。上述の「メタ」の概念の観点から、代表的なメタヒューリスティクスを捉えてみる。Genetic Algorithm (GA) [51, 59, 60, 61, 62]は、交叉・突然変異・淘汰の3つの操作から構成されている。最適化の観点からは、交叉は個体単体では生成することが不可能な近傍解の生成操作とみなすことができる。また、個体の遺伝子配列を部分的に変える操作である突然変異は、それぞれの個体が独立に作り出し得る近傍解の生成操作とみなせる。これらの操作はいずれもランダムかつ発見的な近傍探索である。解空間中のどこでこの発見的な近傍探索を行うかを決定する上位（メタ）操作が淘汰（選択）である。淘汰操作により優良解を次世代に継承し、優良解の近傍解を交叉操作と突然変異操作により生成するという上位（メタ）概念によってアルゴリズム全体の秩序が形成されている。また、Tabu Search [62, 67]は近傍探索に決定論的な山登り法を用い、さらにタブリストによる禁止操作を導入することで大域的な探索を可能とした手法である。GA と同様に、解空間中のどこで近傍探索を行うかを決定する上位（メタ）操作がタブリストに基づく禁止操作であり、この上位（メタ）概念によってアルゴリズム全体の秩序が形成されている。よって、メタヒューリスティクスの元々の特徴は、下位の発見的操作と上位概念のバランスによってアルゴリズムを設計することにある。

以降では、最適化アルゴリズムとしてのメタヒューリスティクスの特徴を概観していく。メタヒューリスティクスは探索点（個体）が発見的なメカニズムに基づく操作に従って、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。多くのメタヒューリスティクスに共通する特徴は、① 最小の情報で最適化が可能な直接探索法、② 生物の進化過程や群知能などの経験則などをアロジーとして構築される発見的手法、③ 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可能な近似解法、④ 確率的乱数を利用した確率的解法、⑤ 相互作用を及ぼす複数の探索点を利用する多点探索型手法、である。一方、2.5節の勾配法を中心とした数理最適化に共通する特徴は、上記と対照的に表すと、① 目的関数に対して微分可能性や連続性を要求する手法、② 数理的な根拠に基づき構築される手法、③ 目的関数がある条件を満たせば、厳密に近い最適解を求めることができ可能な手法、④ 確率的乱数を利用しない決定的解法、⑤ 単体の探索点により探索を進める単点探索型手法、である。これらの特徴の違いにより、メタヒューリスティクスは数理最適化に比べ、BBOの環境下での使用が想定される、2.8節の逐次近似最適化との親和性が高い、大域的最適化に向いた探索ダイナミクスを生み出せる、などの大きなメリットを有している。特に、多点探索は探索への貢献以外にも、下記のような応用面でのメリットも有している。

(1) 多点探索による探索のロバスト性の向上

直接探索法はシミュレータの出力や計測値を直接用いることが可能だが、最悪の場合、シミュレータの都合などで解の評価ができない可能性がある。しかしながら、メタヒューリスティクスのような多点探索であれば、一部の探索点における目的関数値の情報が欠落した場合でも他の探索点を用いて探索を継続できるため、探索におけるロバスト性が高い。

(2) 多点探索による非劣解の同時探索

複数の目的関数で構成される多目的最適化問題は、完全最適解が存在する特殊な場合を除いて、他の解に比べて劣っていない解の集合（非劣解集合：Pareto Frontier）が最適解となる。多点探索型のアルゴリズムは、各探索点を非劣解集合の部分集合の探索に対応付けることで、非劣解の部分集合を同時に探索することが可能である。このようなアプローチは多目的最適化[53]と呼ばれている。

このように、メタヒューリスティクスは数理最適化にはない特徴を有しており、BBOにおいて重要な役割を果たす可能性が高い。

上記以外のメタヒューリスティクスの特徴として、パラメータが持つ高い自由度を挙げることができる。パラメータが持つ高い自由度を活用することで、パラメータの少ないアルゴリズムに比べて、多種・多様な問題を効率的に解ける可能性がある。一方で、パラメータの自由度の高さは、メタヒューリスティクスにおけるパラメータ設定・調整の煩雑さを生じさせることが多い。よって、パラメータ設定・調整に有効な知見を抽出し、系統的に分類・整理することは重要である。特に、直接探索法であることから BBO の環境下で適用されることを念頭に置くと、目的関数のクラスや性質の有無や変化に対して、ユーザがパラメータを適切に設定できず、解を吟味した後に再度パラメータ設定のステップに戻る場合がほとんどである。また、多くのメタヒューリスティクスはアルゴリズムの探索性能は目的関数の構造や性質から影響を受けることが多い。特に、2.3 節の目的関数の悪スケール性や変数間依存性の有無による影響が大きい。BBO での環境下で適用されることを念頭に置くと、目的関数のクラスや性質の有無や変化に対して、ユーザがアルゴリズムを適切に選択できず、解を吟味した後に再度アルゴリズム選択のステップに戻る場合がほとんどである。以上をまとめると、BBO での環境での適用を想定した課題は下記の二つだと考えられる。

- 目的関数のクラスや性質の有無や変化に対して、パラメータ設定がなるべく不要で探索性能を十分に発揮できることが望ましいが、BBO ではそれに応じてユーザがパラメータを適切に設定できず、高い専門的な知識・経験や試行錯誤を必要とする。
- 目的関数のクラスや性質の有無や変化に対して、アルゴリズムの探索性能が影響されないことが望ましいが、BBO ではユーザがそれに応じてアルゴリズムを適切に選択できない。

このように、BBO 環境下での適用が期待される一方、課題も残されている。

2.8 逐次近似最適化

2.8.1 逐次近似最適化

メタヒューリスティクスのような直接探索法は、BBO のように、数式モデルの存在や連續性・微分可能性などの条件を必要とせず、対象の評価値情報のみで最適化を行える。このため、直接探索法はその特徴を最大限に活かし、モデリング・シミュレーション技術などと結合した最適化の新たな枠組みの構築が可能となる[54]。このような概念の一例が図 2.1 である。数理最適化を使用する場合、最適化アルゴリズムが規定するクラスの数式モデルが必要である。例えば、単体法を用いる場合、線形かつ連続な数式モデルが必要となる。このようなアプローチは、数式モデルの構築が容易な良構造の分野を除いて、モデルと実システムの乖離が大きくなり、得られた最適解の実行可能性や最適性が十分ではないなどの問題が生ずる。これに対して、直接探索法を用いれば、近年の進歩が著しいシミュレーション技術・計測技術などの活用が可能となり、シミュレータからの評価値情報を直接用いた最適化や計測値を直接用いたオンラインの最適化も実現できる。また、シミュレーションや計測に大きな負荷がかかる場合には、ニューラルネットワークやカーネル法などの数値モデリング技術を用いることにより、シミュレーションや計測の負荷を抑えつつ最適化が実現できる。

このように、BBO の環境下で得られるサンプル点とモデリングを併用し、目的関数を近似した代理モデルに対して逐次的に最適化を行うことで、全体の評価関数を削減するアプローチを逐次近似最適化 (Sequential Model Based Optimization : SMBO), 応答局面法 (Response Surface Methodology) という[138]¹¹。具体的には、① 探索過程で得ているサンプル点を用いて代理モデルを生成する、② 代理モデルに対して最適化を行う、③ 最適化によって得た解に基づき、新たにサンプリングする、という手順を繰り返す。これは元々実験計画法と呼ばれる、実験する条件をなるべく実験回数を増やすずに効率的に決定していくプロセスが起源である[139]。なお、先ほど述べた理由から、この最適化のステップにおいて直接探索法、あるいは大域的最適化を用いることで、代理モデルは非凸・多峰性関数として、より実際に近い高精度なモデルとすればよい。先行研究では、メタヒュー

¹¹ 逐次近似最適化自身も目的関数の解析的情報を用いないことから、直接探索法として分類する場合もある。

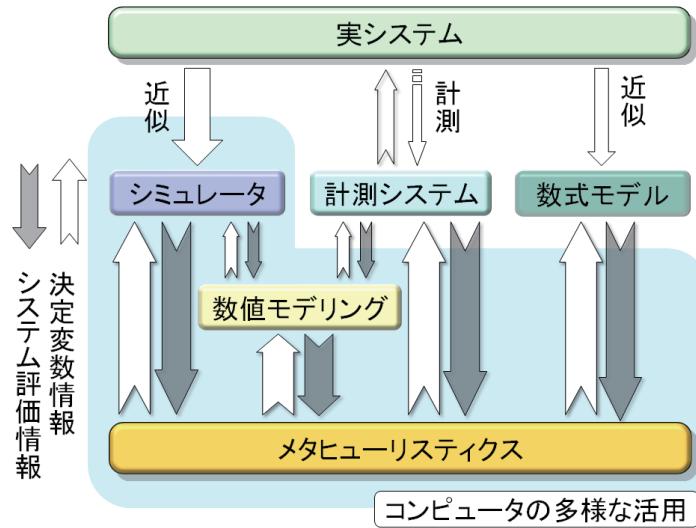


図2.1： モデリング・シミュレーション・計測技術と結合した最適化の枠組み[54]

リスティクスと Radial Basis Function Network (RBFN) を始めとする非線型モデリングを組み合わせた方法や、非線型モデルから有効な位置に直接サンプリングする方法など、様々な SMBO が検討されている[140, 141, 142, 143, 144]。これに加えて、2010 年以降の第3次人工知能（Artificial Intelligence : AI）ブームを牽引する深層学習[13]や Gauss 過程など、モデリング技術がさらに発展しているため、これらを組み込むことでより高精度な代理モデルを使用できると考えられる。なお、先行研究で実施されているのは RBFN を用いた方法が多いが、近年の第3次 AI ブームにおける最先端のモデリング技術が活用されているとは言い難い。Bayes 最適化で使用されるように、Gauss 過程やサンプリング方法を融合させることで、逐次近似最適化手法としてより発展することが期待できる。

2.8.2 Bayes 最適化

近年の SMBO の形態として、Bayes 最適化 (Bayes Optimization) [145, 146]¹² がよく見られる。Bayes 最適化は、SMBO のプロセスにおいて、① Gauss 過程 (Gaussian Process :

¹² 元々、Bayes 最適化はクリギング (Kriging) 法や Efficient Global Optimization という呼称として知られているが、現在は Bayse 最適化という呼称が主流である[147, 148, 149]。

GP) 回帰モデルなどを用いて代理モデルを生成した後, ② 獲得関数と呼ばれる関数を構成し, それを最大化する点を求める, ③ その最大点にサンプリングする, を実施する。

GP 回帰モデルは, 入出力データから, 出力に関する確率分布を得ることで予測する Bayes 的なカーネルマシンである。GP 回帰モデルの特徴として, 非線型モデル, 確率モデル, カーネル法, などが挙げられる。まず, 非線型な特徴量に変換するため, ニューラルネットワークの普遍性定理が適用されるため, 非常に柔軟性の高い非線型モデルである [150]。次に, Bayes 推定ベースの確率モデルであるため, 予測点ではなく, 予測範囲が得られる。獲得関数は, この予測範囲を活用して, 新たにサンプリングする際に得られる改善量の期待値や GP 回帰モデルから生成される信頼区間の上限値など, 改善に貢献する指標とできる[147, 148, 149]。サンプリング位置決定のための指標の方針として, 探索／活用 (Exploration／Exploitation) があり, 後述する多様化・集中化と類似している。さらに, カーネル法の一種であるため, サンプル点を無限次元の特徴量空間に写像し, その空間上でフィッティングすることで複雑なモデルを構築できる。RBFN も Gaussian 関数による非線型な特徴量を用いているが, これは有限次元の特徴量ベクトルであり, そのハイパープラメータもユーザが決定することに対して, GP 回帰はカーネルトリックにより特徴量ベクトルの次元数に依存せず (次元数の呪いの影響を受けない), 特徴量 (カーネル関数) に関するハイパープラメータを最適化する枠組みがあるため事前に設定する必要がない, という点で有利である。以上のように, BBO 環境下における SMBO の形態として, 直接探索法とモデリングと組み合わせた方法や, モデリングと直接的なサンプリングを組み合わせたなどの形態が挙げられるが, これらは SMBO として設計する際に有効な操作として類似しており, 親和性が高いことから, これらを融合させた新たな SMBO の開発などが期待される。

2.9 最適化アルゴリズムの最適化能力

2.5 節の勾配法を中心とした数理最適化は, 探索方向ベクトルとして目的関数の解析的情報 (勾配ベクトルや Hesse 行列) を使うことで降下条件を満たしている。これが探索点の駆動力となって, 探索ダイナミクスを生み出していることから, 解析的情報が数理最適化の

最適化能力といえよう。一方、2.7節のメタヒューリスティクスを始めとする直接探索法は、上記の解析的情報以外の情報を用いて最適化能力を生み出している。本節では、直接探索法における最適化能力について述べる。

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[53, 54]。POPが成立していれば、良い解における類似構造を活用することで、より良い解を探索できる可能性が高い。POPは漠然とした原理だが、下記のように解釈・定義することで、問題におけるPOPの成立の検証や最適化アルゴリズムへの反映が可能となる[54]。

- (1) 「優れた解」 = 優れた目的関数値を持つ解
- (2) 「類似構造」 = 解同士の距離が近いこと

直接探索法の探索構造は、POPに基づき構成されており、解空間の偏り構造を把握・活用することで、効率的な探索を行うことができる。例えば、2.7.1項で述べた滑降シンプレックス法は、探索点群内の最良探索点 c-best や最悪探索点 c-worst を用いている。鏡映操作や縮小操作では、c-worst からの差分ベクトルを使い、拡張操作では、鏡映操作で生成した、c-best よりも良い解に向かう差分ベクトルを使っている。さらに、近傍解が現在の探索点よりも優れた解である（降下条件を満たす）ときにのみ、探索点を近傍解に移動させる。このように、直接探索法は、2.4.2項で述べた探索点群や探索過程で得た特殊な解（探索履歴）を活用することで相対的に有望領域を見出すと同時に、その領域へ探索点を移動させるベクトルを用いる、あるいは降下条件を満たす場合に探索点を移動させることで、解の改善を図っている。以上の操作が探索点の駆動力となり、最適化能力を生み出しているといえる。

さらに、メタヒューリスティクスでは、POPを効果的に活用する探索指針として、多様化（Diversification, Exploration）・集中化（Intensification, Exploitation）¹³が知られている[53, 55]。多様化は、広い範囲を探索することで、解空間の偏り構造を把握する探索指針であり、集中化は、有望領域を重点的に探索することで、より優れた解を得る探索指針である。多様化・集中化は抽象的な概念だが、先ほどの考察と照らし合わせると、集中

¹³ Exploration / Exploitation（探索／活用）は Bayes 最適化におけるサンプリングの方針との類似性に由来した呼称である。

化は上記の駆動力による移動そのものであり、最適化能力の源だと解釈できる。しかしながら、非凸・多峰性の目的関数を対象とした場合、過度な集中化を実施しても、局所解に陥ってしまい、大域的最適解を得られない。このような考え方から、多様化は、解空間をなるべく広く把握し、大域的最適解を探索するための指針として位置づけられる。このために、メタヒューリスティクスでは、多点探索や確率的要素による摂動などにより、多様化に貢献する要素が探索構造に埋め込まれている。特に、多くのメタヒューリスティクスは、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。この戦略は、①探索序盤では、探索範囲を広くしておくことで、解空間の偏り構造を把握し、②探索が進むにつれて、把握した偏り構造を活用し、探索範囲を有望な領域に徐々に狭めていき、より優れた解を探索する。以上から、メタヒューリスティクスは、上記のPOPに基づく探索点群や探索履歴の情報に加えて、多点探索や確率的要素も探索に活用することで、多様化・集中化を実現している。

2.10 最適化アルゴリズムのロバスト性と適応性

本論文の目的は、BBOの環境下を想定したアルゴリズム設計であり、本章では、勾配法を中心とした数理最適化と直接探索法を概観・比較することで、BBOではメタヒューリスティクスが期待できる特徴を有していることを確認した。一方、[2.7.3項](#)で述べたように、BBOを念頭に置いたメタヒューリスティクスの課題として、①ユーザがパラメータを適切に設定できず、高い専門的な知識・経験や試行錯誤を必要とする、②目的関数のクラスや性質の有無や変化に対して、アルゴリズムの探索性能が影響されないことが望ましいが、BBOではユーザがそれに応じてアルゴリズムを適切に選択できず、試行錯誤を必要とする、が挙げられることも確認した。

そこで、BBOを解くアルゴリズムが具備する性質として、「ロバスト性」と「適応性」を定義する。[図2.2](#)に最適化アルゴリズムにおけるロバスト性と適応性のイメージを示す。[図2.2\(a\)](#)に示すように、ロバスト性とは、探索過程でアルゴリズムのパラメータを固定した状態でも、目的関数の構造や性質に対して探索性能が維持できる性質を指す。これに対して、[図2.2\(b\)](#)に示すように、適応性とは、探索過程で得られる問題構造の情報を活用し

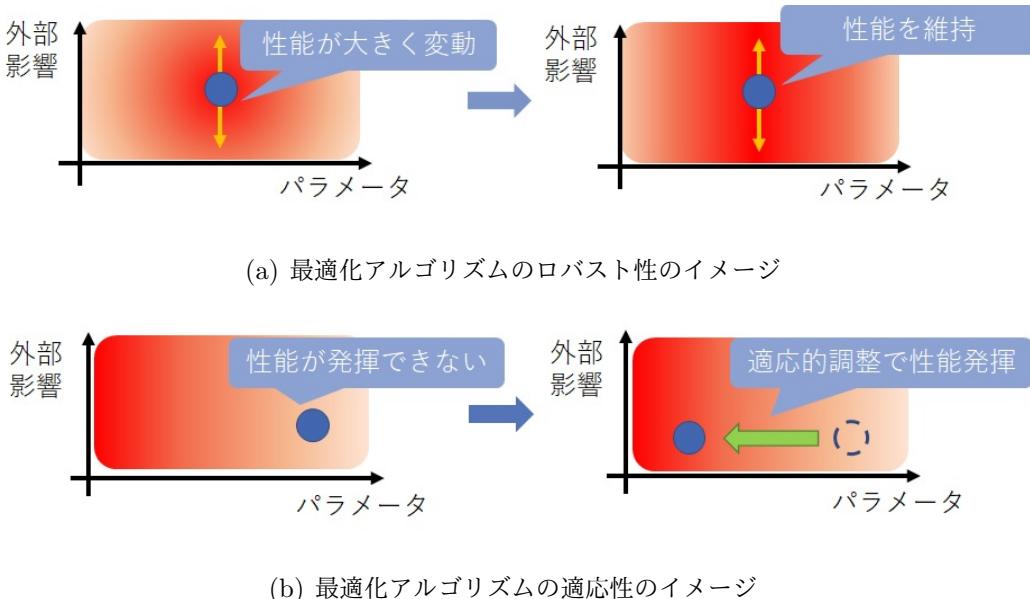


図 2.2：最適化アルゴリズムのロバスト性と適応性のイメージ

て、アルゴリズムのパラメータを動的に調整することで、対象問題に適応する性質を指す。なお、ロバスト性は現代制御理論のロバスト制御から着想を得ている。 H_∞ 制御のようなロバスト制御理論では、元々制御対象のある状態において、優れた制御性能（安定性など）を発揮できるようにコントローラーを設計しても、制御対象の特性（外乱・温度などに対する性質変化）によって、制御性能が大きく変動し、ピーク性能を維持できなくなることをロバスト性が低いと呼ぶ。そこで、この現象を防ぐために、その特性を事前に特定しておき、コントローラーのパラメータを固定した状態でも、制御性能のピークを維持するロバスト性が高いと呼び、これを実現しようとする考え方が、ロバスト制御である¹⁴。

メタヒューリスティクスは、パラメータ設定に対する根拠が希薄であるため、ロバスト性が高いパラメータ設定を与える、あるいは探索構造の理論的な設計はされていないことから、ロバスト性に課題があると考えられる。また、通常は調整可能なパラメータを探索過程で固定しているが、パラメータを探索過程における探索状態に応じて適応的に調整することで、問題構造や条件に応じて平均的に高い性能を発揮することが期待できることか

¹⁴ ロバスト制御では、制御対象とその特性を特定するため、ピーク性能をほぼ維持することが可能だが、ブラックボックス最適化では、対象問題の性質が不明なので、必ずしもピーク性能を維持するとは限らない点で異なることに注意されたい。

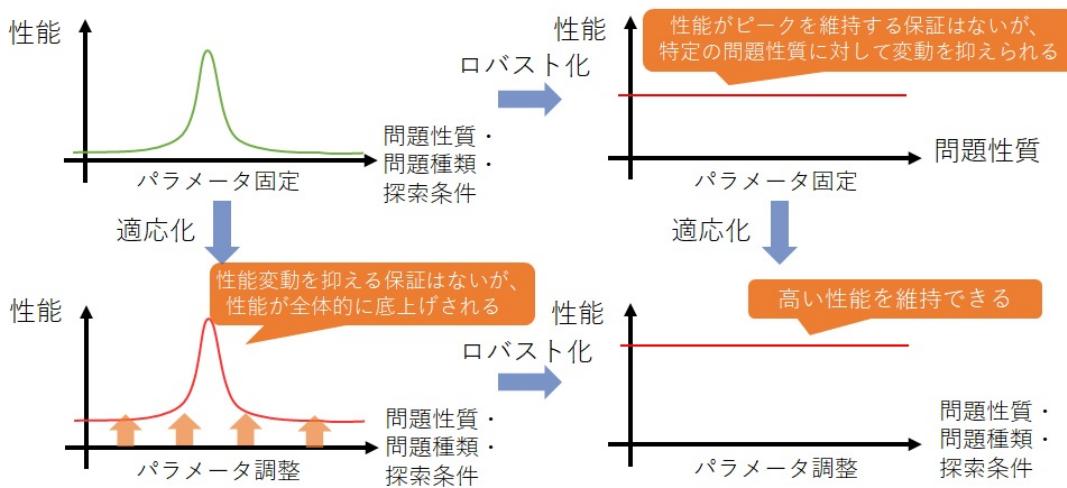


図 2.3：ロバスト性と適応性を具備する最適化アルゴリズムのイメージ

ら、適応性に改善の余地があると考えられる。そこで、ロバスト性・適応性の両方の課題解決がBBOにとって必要だと考えられる。図2.3にロバスト性と適応性の効果のイメージを示す。図2.3の左上のように、ロバスト性・適応性の両方を具備しない場合、問題の性質・種類・探索条件の変化に対して、探索性能が大きく変動してしまうと同時に、全体的に性能も低いケースが多くなってしまう。図2.3の右上のように、ロバスト性を具備する場合、問題の性質・種類・探索条件の変化に対して、探索性能を維持することが可能となる¹⁵が、ピーク性能を維持する保証はない。一方、図2.3の左下のように、適応性を具備する場合、様々な問題の性質・種類・探索条件においても全体的に性能向上することが期待されるが、それらの変化に対して、探索性能の変動を抑えられる保証はない。そして、図2.3の右下のように、ロバスト性・適応性の両方を具備する場合、問題の性質・種類・探索条件の変化に対して、探索性能を維持すると同時に、様々な問題の性質・種類・探索条件においても全体的に性能向上することで、結果的に多くのケースにおいて高い性能を発揮することが期待される。これらを踏まえると、BBOのための高いロバスト性・適応性を具備するメタヒューリстиクスの開発は重要である。

¹⁵ 一般に、問題の種類・探索条件に対して、理論的なアプローチにより確実にロバスト性を高めることは困難であるため、図2.3では横軸は問題性質に限定している。

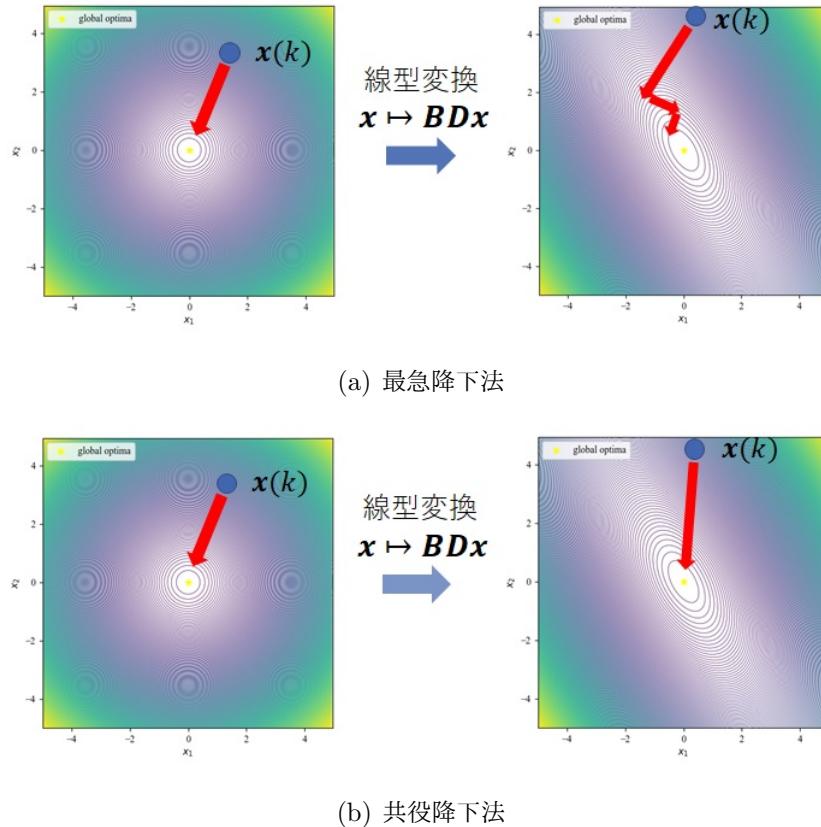


図 2.4：変換不変性とロバスト性の関係

ロバスト性を実現する性質として変換不変性が挙げられる。変換不変性は、疑似乱数系列が同じであれば、解空間や目的関数の変換に対してアルゴリズムの探索性能が再現性を有する、つまり同等あるいは類似の結果が得られる性質である。メタヒューリスティクス（最適化アルゴリズム）が有する変換不変性は手法によって様々である。実際、ある変換に対する不変性を欠如している手法は、その変換の有無によって、探索性能が大幅に変化することがしばしば見られる。図 2.4 に示す例を用いて、変換不変性とロバスト性の関係について説明する。線型変換不変性を具備していない最急降下法を用いた場合を図 2.4(a) に示し、線型変換不変性を具備する共役方向法を用いた場合を図 2.4(b) に示す。図 2.4(a), (b) の左図は、2 次元平面における付録 C の Sphere 関数の等高線を示している。Sphere 関数は $f(\mathbf{x}) = \mathbf{x}^T \mathbf{x}$ という 2 次形式（2 次関数）の標準型であるため、等高線が等方かつ座標系に直交している。つまり、変数間独立性かつ良スケール性を有する。これに対して、図 2.4(a), (b) の右図は、Sphere 関数を線形変換 ($\mathbf{x} \mapsto \mathbf{B}\mathbf{D}\mathbf{x}$, \mathbf{B} は 30 度の回転行列,

$\mathbf{D} = \text{diag}[1, 2]$ はスケーリング行列) した目的関数 $f(\mathbf{x}) = \mathbf{x}^T \mathbf{B}^{-1} \mathbf{D}^2 \mathbf{B} \mathbf{x}$ の等高線を示している。つまり、変数間依存性かつ悪スケール性を有する。図 2.4(a) の左図は Sphere 関数を対象としたときの最急降下法の例を示している。最急降下法で、探索点 $\mathbf{x}(k)$ の軌道は等高線の法線方向であるため、理論的に一度の反復で最適解に到達する。しかし、図 2.4(a) の右図のように線形変換が適用された 2 次関数を対象としたとき、最急降下法では探索点は等高線の法線方向しか進めないため、最適解に到達するには複数回の反復が必要となり、探索効率が極端に低下する。このように、2 次元部分空間に閉じ込められる現象は Stiefel の鳥籠と呼ばれる[26]。これは、本質的には探索方向が線型変換不变性を有さないことが原因である。一方、図 2.4(b) の左図のように、Sphere 関数を対象としたときに共役方向法が探索方向を生成したとする。そして、図 2.4(b) の右図のように線形変換が適用された 2 次関数を対象としたときでも、共役方向法の探索方向は線形変換に伴って変わり、左図と同等の効率で探索が可能である。これは、本質的には探索方向が線型変換不变性を有することに起因する。このため、共役方向法は線型変換不变性を具備するため、線形変換に対して探索効率が変わらない、つまりロバスト性が高いといえる。以上から、2 次関数においては最急降下法と共役方向法のロバスト性は線型変換不变性の有無によって完全に決定される。さらに、線型変換不变性は数理最適化では重要な性質であり、共役勾配法や Newton-Raphson 法も同様に有している。より複雑な関数においても、線型変換不变性の有無が主要因となって、最急降下法の性能は低いことと同時に、線型変換不变性を具備する数理最適化アルゴリズムの性能が高いことが数値実験的に知られている。よって、探索構造の不变性を具備するように設計する方針が、ロバスト性の発揮において重要となると考えられる。

このロバスト性の課題に対して、Particle Swarm Optimization (PSO) や Differential Evolution (DE), Evolution Strategy (ES), 実数値 GA, Firefly Algorithm の変換不变性の欠如を指摘する、あるいは不变性を付加する先行研究がある[88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 122, 124]。また、パラメータ設定のロバスト性に関する先行研究として、実験的・理論的に PSO と DE に対して、パラメータ設定に対する有効な知見を導出している[113, 116, 151, 152, 153, 154, 155, 156]。特に、小熊らの方法[154, 155, 156]は、安定性理論に基づき、PSO や DE の探索点群のダイナミクスに関する安定性指標を精悍に導出している。さらに、Koguma らは、これを逆に利用し、パラメータ設定に対する安定

性指標を完全に導出可能な正規乱数結合に基づくアルゴリズムのフレームワークを提案している[157]。このような不变性や安定性（収束性）は、勾配法を中心とする数理最適化において重要な性質であるため、これらの発展は元々数学的な根拠がないメタヒューリスティクスにおいて非常に大きく貢献すると考えられる。

また、適応性を実現する機能としてパラメータ調整が挙げられる。パラメータを探索過程における探索状態に応じて適応的に調整する、フィードバック型・適応的調整型のパラメータ調整則が最も適応性の発揮を期待できる。この適応性の課題に対して、先行研究では、ES や PSO, DE において、適応的パラメータ調整則を開発している[58, 78, 79, 114, 115, 116, 117, 118, 119]。さらに、金政らは、多点型 ES や PSO, 超球交叉を有する DE に対して、Genetic Programming (GP) を用いて適応的パラメータ調整則を自動的に設計するアプローチを提案している[107, 126, 158]。前者の先行研究は特定のメタヒューリスティクスに対して特有のパラメータ調整則を設計していることに対して、後者の金政らの方法[107, 126, 158]は、あるパラメータ調整則を構成する演算子などの集合を解とみなし、それを対象関数に対して実行したときの探索性能を得るというプロセスを複数回繰り替えず上位の最適化問題を GP が解くことで、使用したアルゴリズム・対象関数に対して最適なパラメータ調整則を自動的に設計するという点において、汎用的かつ BBO にとって有効なアプローチである。

また、変換不变性はパラメータ調整則の付加によって、その有無が変わることがある。これを活用すれば、①欠如している変換不变性を補完し、探索構造全体が多くの変換不变性を有しているかのように、パラメータ調整則を付加することが可能、②有している変換不变性を維持するように、パラメータ調整則を付加することが必要、という設計指針が立てられる。よって、探索構造全体が多くの変換不变性を獲得する、あるいは維持するようなパラメータ調整則を検討することは、ロバスト性・適応性の向上にとって有効だと考えられる。

2.11 まとめ

本章では、本論文で扱う BBO の最適化問題の構造や性質と、最適化技法の体系として発展してきた数理最適化のアルゴリズム、BBO で使用される直接探索法や逐次近似最適化について概観し、その数理的構造や特徴を記述した。また、数理最適化と比較し、メタヒューリスティクスの特徴が BBO において重要な役割を果たすことが期待されることを述べた。最後に、BBO において重要な性質としてロバスト性・適応性を挙げ、これを実現するアプローチとして変換不变性とパラメータ調整を挙げた。これらの背景を踏まえ、本論文では、変換不变性とパラメータ調整に基づくメタヒューリスティクスの設計を検討する。なお、不变性とパラメータ調整則に関する先行研究の詳細は、第 4 章と第 5 章にて述べる。

3

メタヒューリстиクス の探索構造の解析

3.1 本章について

メタヒューリстиクスは、Nature-Inspired な発見的手法の枠組みである。初期のメタヒューリстиクスは1960年～1970年頃に、Evolution Strategy(ES) や Genetic Algorithm(GA) が進化計算として登場した。1970年以降、Simulated Annealing(SA) や Tabu Search(TS) などの新たなメタヒューリстиクスが登場し、「メタヒューリстиクス」という用語が使用され始めた。1990年以降は実数値GA や多点型ESなどの進化計算が発展すると同時に、Particle Swarm Optimization や Differential Evolutionなどの群知能アルゴリズムが登場し、メタヒューリстиクスとして体系化された。2000年以降、Artificial Bee Colony Algorithm, Firefly Algorithm, Cuckoo Searchなど、様々なアナロジーに富んだメタヒューリстиクスの開発はますます盛んになった。

上記の歴史は、その潮流を踏まえると、草創期、発展期、全盛期に区分できると考えられる。草創期は、ES, GA, SA, TSなどの初期のメタヒューリстиクスが登場した1960年～1990年の期間、発展期は、初期の発展形や他のアルゴリズムも登場し、メタヒューリстиクスとして体系化され始めた1990年～2000年の期間、全盛期は、これまでのアルゴリズムの改良・解析・応用がなされると同時に、斬新なアナロジーに基づくアルゴリズムが開発され、さらに多様化が進み、メタヒューリстиクス全体が一大分野として確立した2000年～2010年の期間、だと呼ぶことができよう¹。特に、全盛期の研究は、アルゴ

¹ この区分を裏付ける文献を全て挙げることは難しいが、本論文の参考文献全体（約200件）がこの区分に

リズムの応用を除いて、1990年以前の古典的なアルゴリズムに着目した改良（ハイブリッド化含む）・解析か、斬新的なアナロジーに基づくアルゴリズムの開発、に明確に分かれている。一方、2010年以降、全盛期のスタイルを「そのまま」継承する一方、メタヒューリスティクス自体の研究事例は少なくなっている。この要因としては、①斬新的なアナロジーに基づくアルゴリズムは新たに提案されているものの、多くの古典的アルゴリズムの改良系よりも有用でない、あるいは有用性の検証が十分に実施されていない、②上記の状況から、この分野がすでに成熟しつつあると認識されると同時に、同時期に第3次人工知能ブームが到来した影響で深層学習や機械学習のアルゴリズムに研究の対象が移行した、などが考えられる。実際、IEEE CEC や ACM GECCO などの国際会議あるいは、電気学会 電子・情報・システム部門大会、計測自動制御学会 システム・情報部門 学術講演会、進化計算学会 進化計算シンポジウムなどの国内会議において、発展期・全盛期では、単純なメタヒューリスティクスのアルゴリズム開発に関する発表が盛んであったが、2010年以降は、その発表件数は徐々に減少し、深層学習や機械学習のアルゴリズムの開発、あるいはそれらとの親和性を前提としたメタヒューリスティクスの開発に関する発表が増えている。特に、①の本質的要因は、メタヒューリスティクス本来の特徴である「アナロジーの斬新さ」を重視する傾向が継承され、そのアプローチだけが多くなされていることだと考えられる。言い換えると、多くの既存のアルゴリズムよりも優秀な探索性能を発揮する、数理最適化のような何らかの理論的な保証がある、などの最適化アルゴリズムとしての有用性が不十分であることを表している。しかしながら、第1章で述べた通り、今後のシステムの大規模化・複雑化や周辺技術の発展などを踏まえると、環境変化に対応可能なブラックボックス最適化技術として、メタヒューリスティクスの開発は重要だと考えられる。ただし、上記で述べた通り、実応用上十分なメタヒューリスティクスを開発するには、斬新的なアナロジーに囚われず、多くの既存のアルゴリズムよりも優秀な探索性能を発揮する、何らかの理論的な保証がある、などの最適化アルゴリズムとしての十分な有用性が必要だと考えられる。

以上のメタヒューリスティクスの歴史や外部環境の変化を踏まえ、本論文では、ブラックボックス最適化のための、何らかの理論的な保証がありながら、優秀な探索性能を発揮するメタヒューリスティクスの設計を目指す。本章では、第4章以降の数理的解析のため

大まかに従っていることに基づいている。

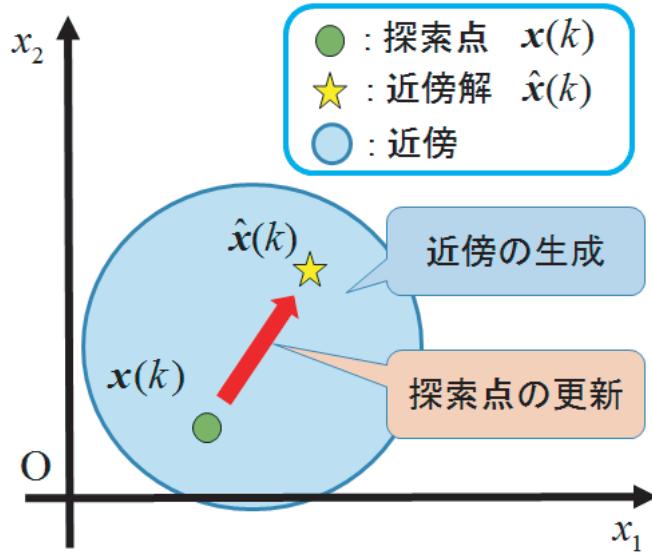
の基盤を形成するために、既存のメタヒューリスティクスとして、Evolution Strategy, 実数値 Genetic Algorithm, 実数値 Simulated Annealing, Particle Swarm Optimization, Differential Evolution, Artificial Bee Colony Algorithm, Firefly Algorithm, Cuckoo Search を概観しながら、多様化・集中化の探索戦略の観点から、その探索構造を解析・分類する。

3.2 メタヒューリスティクスの基本的な探索構造

3.2.1 基本的な探索構造

本節では、メタヒューリスティクスの基本的な探索構造を述べ、一般的な更新則を示す。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[53, 54]。近傍の生成では、探索点 $\mathbf{x}(k)$ 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 $\hat{\mathbf{x}}(k)$ （近傍解）を生成する。探索点の更新では、探索点 $\mathbf{x}(k)$ が近傍解 $\hat{\mathbf{x}}(k)$ へ更新し、 $\mathbf{x}(k+1)$ となる。この更新は、移動ベクトル $\mathbf{v}(k)$ を用いて、 $\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{v}(k)$ と表される。これは、式 (2.9) 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 $\hat{\mathbf{x}}(k)$ を生成し、探索点 $\mathbf{x}(k)$ 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。なお、これらの更新は探索点 $\mathbf{x}(k)$ だけではなく、アルゴリズムによっては特殊な解やパラメータを保存しておき、それを更新することもある。

また、探索点の移動方法は、基準の違いによって、「絶対移動」、「確率的移動」、「改善移動」の 3 種類に分類される。絶対移動では、近傍解 $\hat{\mathbf{x}}(k)$ の目的関数值に関係なく、探索点 $\mathbf{x}(k)$ は近傍解へ移動する。これに対して、改善移動では、探索点 $\mathbf{x}(k)$ の目的関数值よりも近傍解 $\hat{\mathbf{x}}(k)$ の目的関数值が良い、つまり $f(\hat{\mathbf{x}}(k)) < f(\mathbf{x}(k))$ を満たす場合のみ、近傍解へ移動する。確率的移動では、絶対移動と改善移動の間に位置する移動方法である。Simulated Annealing では、改善する場合か、乱数が事前に設定した閾値（受理確率）を下回る場合、は探索点が移動し、そうでなければ移動しない。それぞれの場合の移動方法



は、式(3.1)、式(3.2)、式(3.3)として表される。

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

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

$$\mathbf{v}(k) = \begin{cases} \hat{\mathbf{x}}(k) - \mathbf{x}(k), & f(\hat{\mathbf{x}}(k)) < f(\mathbf{x}(k)) \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (3.3)$$

ただし、 $P \in [0, 1]$ は受理確率、 $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ である。ど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 $\mathbf{v}(k) = \hat{\mathbf{x}}(k) - \mathbf{x}(k)$ となるため、近傍解 $\hat{\mathbf{x}}(k)$ の生成方法が探索ダイナミクスを決定づけるといえる。なお、更新方法の種類によって、2.4.2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、2.4.2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、改善移動を行うアルゴリズムでは、目的関数値が改善しなければ探索点 $\mathbf{x}(k)$ は移動しないため、各探索点 $\mathbf{x}(k)$ は式(2.15)の p-best と一致し、式(2.17)の c-best は式(2.16)の g-best と一致する。

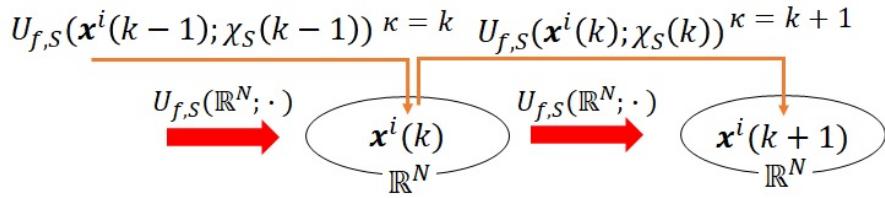


図 3.2： 更新則の写像図式

3.2.2 探索点の更新則

メタヒューリстиクスの探索点の更新則 $U_{f,S}$ を定義 3.1 で定義する。

定義 3.1 (探索点の更新則) 式 (2.1) の解空間 S , 目的関数 f の最小化問題を対象としたとき, 探索点 $\mathbf{x}^i(k)$ の更新則 $U_{f,S} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ は式 (3.4) で表される。

$$\mathbf{x}^i(k+1) = U_{f,S}(\mathbf{x}^i(k); \chi_S(k), \mathcal{R}(k), \mathcal{C}(k)) \quad (3.4)$$

ただし, $\chi_S(k)$ は探索点群 $\mathcal{X}(k)$ の探索履歴, $\mathcal{C}(k)$ は更新用パラメータ集合, $\mathcal{R}(k)$ は確率変数用パラメータ集合である。

式 (3.4) で文字の添字として S や f を付しているのは, 内部では S や f に依存していることを表す。例えば, $\chi_S(k)$ は解空間 S で探索してきた履歴であり, $U_{f,S}$ は解空間 S , 目的関数 f のもとでの更新である。探索履歴 $\chi_S(k)$ に対応する目的関数値 f の集合 $\zeta_f(k)$ も更新則 $U_{f,S}$ のパラメータだが引数として明記せず, $U_{f,S}$ で潜在的に使用することとする。なお, 定義 3.1 では $r_j(k) \in \mathcal{R}(k)$ をパラメータとしているが, 実際には確率変数用パラメータ $r_j(k)$ は確率分布 \mathcal{P}_j に従う疑似乱数を表し, 確率分布 \mathcal{P}_j の性質により, 各パラメータが実数値／離散値と定まる。本論文では, パラメータ $\mathcal{R}(k), \mathcal{C}(k)$ は解空間 S や目的関数 f の変換とは独立しているため, 場合によってはこれらを省略し, 式 (3.4) は $\mathbf{x}^i(k+1) = U_{f,S}(\mathbf{x}^i(k); \chi_S(k))$ と表記しても良いものとする。図 3.2 に定義 3.1 のメタヒューリстиクスの探索点の更新則 $U_{f,S}$ の写像図式を示す。この写像図式に示すように, 更新則 $U_{f,S}$ は探索履歴 $\chi_S(k)$ や $\mathcal{R}(k), \mathcal{C}(k)$ をパラメータとし, 探索点の $\mathbf{x}^i(k)$ の写像として定義されている。

また, 3.2.1 項に従うと, 定義 3.1 の更新則 $U_{f,S}$ は近傍生成と探索点の更新に分かれるた

め、定義 3.2 としても定義できる。

定義 3.2 (探索点の更新則) 式 (3.4) で表される、探索点 $\mathbf{x}^i(k)$ の更新則 $U_{f,S}$ は式 (3.5) で書き換えられることとする。

$$\mathbf{x}^i(k+1) = U_{\text{mov},f,S} \circ U_{\text{nei},f,S}(\mathbf{x}^i(k); \chi_S(k), \mathcal{R}(k), \mathcal{C}(k)) \quad (3.5)$$

ただし、 $U_{\text{nei}} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ は近傍解 $\hat{\mathbf{x}}^i(k)$ を生成する近傍生成則であり、式 (3.6) で表される。 $U_{\text{mov}} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ は近傍解 $\hat{\mathbf{x}}^i(k)$ に探索点 $\mathbf{x}^i(k)$ を移動させるかどうかを判断する探索点の移動則であり、式 (3.7) で表される。

$$\hat{\mathbf{x}}^i(k) = U_{\text{nei},f,S}(\mathbf{x}^i(k); \chi_S(k), \mathcal{R}(k), \mathcal{C}(k)) \quad (3.6)$$

$$\mathbf{x}^i(k+1) = U_{\text{mov},f,S}(\hat{\mathbf{x}}^i(k); \chi_S(k), \mathcal{R}(k), \mathcal{C}(k)) \quad (3.7)$$

定義 3.2 は、式 (3.5) の更新則 U が、近傍生成則 U_{nei} と探索点の移動則 U_{mov} の合成写像である（近傍を生成してから、移動させる）ことを表している²。また、探索点の移動則 U_{mov} は 3.2.1 項において述べた通り、3 種類のいずれかであるため、近傍生成則 U_{nei} がアルゴリズムによってバリエーションがある。メタヒューリスティクスでは、特殊な解を用いることで降下方向を生成していることが最適化能力に該当するため、次節以降は、特に近傍生成則 U_{nei} に着目する。

なお、先行研究では、更新則を Markov 連鎖モデルを用いて定義し、メタヒューリスティクスの大域的収束性などの解析に活用している[159, 160, 161]。定義 3.1 は、Markov 連鎖モデルによる定義と類似しており、更新則 U の引数はいわゆる状態変数の意味に近い。しかし、通常 Markov 連鎖モデルは乱数ベクトルの存在を前提とする状態遷移確率に着目しているが、メタヒューリスティクス全般には該当しない。よって、本論文では Markov 連鎖モデルではなく、定義 3.1、あるいは定義 3.2 を使用する。

本章では、メタヒューリスティクスを定義 3.2 と照らし合わせて、近傍生成則 U_{nei} につ

² 定義 3.2 は定義 3.1 と本質的な意味や役割は変わらないことに注意されたい。

いて重点的に解析することで、その探索構造を系統的に分類する。

3.3 Evolution Strategy の解析

3.3.1 Evolution Strategy

Evolution Strategy (ES) は、1960 年代に Rechenberg によって開発された後、Schwefel によって様々な改良が施された単点探索型メタヒューリスティクスである[57, 58, 162]。ES は生物の進化過程に基づいており、主に突然変異と淘汰（選択）の二つの操作で探索が行われる。最初の ES は単点探索で、その更新則は式 (3.8) と式 (3.9) で表される。

$$\hat{\mathbf{x}}(k) = \mathbf{x}(k) + \sigma \mathbf{s}(k) \quad (3.8)$$

$$\mathbf{x}(k+1) = \begin{cases} \hat{\mathbf{x}}(k), & f(\hat{\mathbf{x}}(k)) < f(\mathbf{x}(k)) \\ \mathbf{x}(k), & \text{otherwise} \end{cases} \quad (3.9)$$

ただし、 $\sigma > 0$ はパラメータ、 $\mathbf{s}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ は標準正規乱数ベクトルである。式 (3.8) が突然変異、式 (3.9) が選択に該当する。

ところが、スケールパラメータ σ を動的に調整しないと、ES は最適化アルゴリズムとして機能しないことが知られている。そこで、Schwefel は σ を動的に調整する 1/5 ルールを提案している[58]。1/5 ルールは、現時点から一定の反復回数分遡った期間において、探索点の改善が成功した頻度を $p_s(k)$ とし、 $p_s(k) > 1/5$ ならば $\sigma(k)$ を増加させ、そうでなければ、 $\sigma(k)$ を減少させる調整則である。改善頻度 $p_s(k; K)$ は式 (3.10) で表される。

$$p_s(k; K) = \sum_{\kappa=1}^K \mathbb{1}_{f(\mathbf{x}(k-\kappa+2)) < f(\mathbf{x}(k-\kappa+1))}(\chi(k)) \quad (3.10)$$

ただし、 $K \in \mathbb{N}$ は遡る反復回数、 $\mathbb{1}_{\{P\}}(A)$ は指示関数 $A \rightarrow \{0, 1\}$ であり、集合 A について条件 P を満たす場合 1、そうでなければ 0 を返す。代表的な 1/5 ルールとして、式 (3.11) が知られている。

$$\sigma(k+1) = \begin{cases} \sigma(k)/c, & p_s(k) > 1/5 \\ c\sigma(k), & \text{otherwise} \end{cases} \quad (3.11)$$

Algorithm 3 Evolution Strategy ((1+1)-ES)

```

1: procedure EVOLUTION STRATEGY   Step 3 : 解の選択
   ((1+1)-ES)                      7:    $\mathbf{x}(k)$  を式 (3.8) を用いて更新する。
   Step 0 : パラメータ設定          Step 4 :  $\sigma$  の更新
2:   パラメータ  $k_{\max}$ 、 $c$ 、 $\sigma_{\text{start}}$  を設定する。 8:    $\sigma(k)$  を式 (3.11) を用いて更新する。
   Step 1 : 初期化                  Step 5 : 終了判定
3:   探索点の位置  $\mathbf{x}(k)$  を初期配置領域  $S_0$  9:   if  $k = k_{\max}$  then
   内にランダムに配置する。          10:    計算を終了する。
4:    $\sigma(k) = \sigma_{\text{start}}$  とする。 11:   else
5:    $k = 1$  とする。                 12:     $k := k + 1$  とし、Step 2 へ戻る。
   Step 2 : 突然変異              13:   end if
6:    $\hat{\mathbf{x}}(k)$  を式 (3.9) を用いて生成する。 14: end procedure

```

ただし、 $0 < c < 1$ である。この $1/5$ ルールを具備した単点型 ES は (1+1)-ES と称されており、最も基本的なメタヒューリスティクスの一つである。

Algorithm 3.3.1 に (1+1)-ES の疑似コードを示す。なお、メタヒューリスティクスの終了条件は $k = k_{\max}$ (k_{\max} : 最大反復回数)、あるいは、目的関数値の計算回数である評価回数 (Function Calls) T が $T \geq T_{\max}$ (T_{\max} : 最大評価回数) を満たすことである。 (1+1)-ES の終了条件は反復回数であり、評価回数は $T = k_{\max}$ となる。また、探索点の初期位置を超直方体の初期配置領域 $S_0 = [\gamma_{\min,n}, \gamma_{\max,n}]^N \subset \mathbb{R}^N$ 内に与える。

実際には (1+1)-ES は単点型であることから使われておらず、 (μ, λ) -ES や Covariance Matrix Adaptation ES (CMA-ES) などの多点型 ES が使用される[58, 77, 78, 79]。なお、多点型 ES もパラメータ調整則を具備することが一般的である。

3.3.2 Evolution Strategy の探索構造の解析

定義 3.1 に従って、ES の更新則を解析する。まず、通常の ES の近傍生成則 U_{nei} は突然変異の式 (3.8) が該当し、探索点の移動則 U_{mov} は選択の式 (3.9) が該当する。よって、通常の

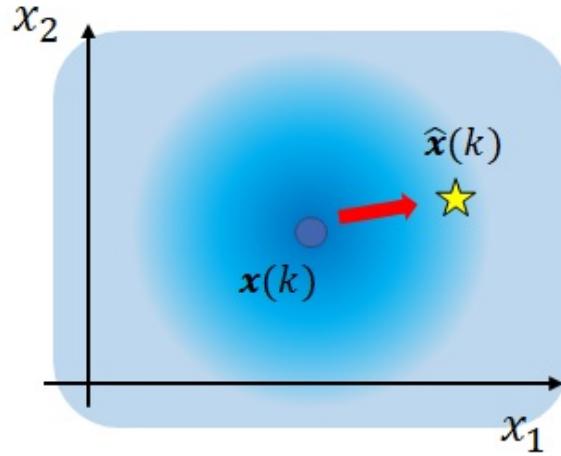


図 3.3：Evolution Strategy の近傍の生成

ES 全体の更新則 U (式 (3.4))において, $\mathcal{R}(k) = \{\mathbf{s}(k)\}$, $\mathcal{C}(k) = \{\sigma\}$, $\mathbf{x}(k) \in \chi(k)$ となる。よって, 探索点自身の位置情報と乱数ベクトルによる摂動によって近傍解を生成し, 改善するなら探索点を移動させる, という仕組みであることがわかる。さらに, (1+1)-ES は, 通常の ES に 1/5 ルール式 (3.11) が付加されたものだが, パラメータ調整則の $\sigma(k)$, K , c , σ_{start} などの量も更新に影響する。このように, パラメータ調整則が含まれる場合は, そのパラメータの状態も更新則に影響すると考えられる。パラメータ調整と更新則の関連に関する詳細は第 5 章で述べる。

図 3.3 に, ES の近傍生成について示す。図 3.3 中の色の濃淡は確率密度の高さを表す。ES は近傍生成則で探索点に摂動を与え, 移動則で改善移動するという単純なメカニズムである。しかしながら, 移動則で改善移動を行うことで, 探索点は有望領域に少しづつ移動していくことが最適化能力だとみなせるため, 通常の ES は集中化の能力しか有していないと考えられる。一方, 乱数ベクトルは解空間内で更新・生成される解とは独立に生成される。このため, 通常の ES における近傍生成則 U_{nei} では, 解空間のスケールとは独立に σ を与えるため, 適切な摂動を与えない。この理由から, (1+1)-ES の 1/5 ルールでは, 改善頻度 $p_s(k)$ に基づき, 改善が見込まれる場合は $\sigma(k)$ を小さく (集中化) し, そうでない場合は $\sigma(k)$ を大きく (多様化) するというように, $\sigma(k)$ を動的に調整することで, 探索過程で解空間のスケールに適応し, 多様化・集中化の能力を生み出している。以上から,

ES自身は多様化・集中化の能力が低いため、(1+1)-ESはパラメータ調整則によってそれを補完したと考えられる。

3.4 Genetic Algorithm の解析

3.4.1 Genetic Algorithm

Genetic Algorithm (GA) は、1960年代に Hollandにより開発された多点探索型メタヒューリスティクスである[59, 60, 61, 62]。GAは生物の進化過程に基づいており、交叉、突然変異、淘汰（選択）の三つの操作で探索が行われる。

交叉は二つの探索点（親）の要素を組み合わせて、解を生成する。交叉における探索点の選択は、非復元抽出が一般的である。具体的な選択方法としては、ランダム選択、ルーレット選択、トーナメント選択、ランキング選択などがある。ランダム選択は、探索点群からランダムに2つ選択する。トーナメント選択は、探索点群からいくつかランダムに選択し、その中で最も良い探索点を選択する。ルーレット選択とランキング選択は、各探索点の適合度 fit_i を計算し、適合度に従う各探索点の選択確率 p_i に基づき選択する。ある解集合 \mathcal{Z} に含まれる \mathbf{x}^i の選択確率 p_i は式(3.12)で表される。

$$p_i = fit_i / \sum_{j=1}^m fit_j \quad (3.12)$$

ルーレット選択における適合度 fit_i は式(3.13)で表され、ランキング選択における適合度 fit_i は式(3.14)で表される。

$$fit_i = \begin{cases} \frac{1}{1+f(\mathbf{x}^i)}, & f(\mathbf{x}^i) \geq 0 \\ 1 + |f(\mathbf{x}^i)|, & \text{otherwise} \end{cases} \quad (3.13)$$

$$fit_i = \frac{m - R_i(\mathcal{Z}) + 1}{m} \quad (3.14)$$

ただし、 $R_i(\mathcal{Z})$ は式(3.15)で表される。

$$R_i(\mathcal{Z}) = \sum_{j=1}^m \mathbb{1}_{\{f(\mathbf{x}^j) < f(\mathbf{x}^i)\}}(\mathcal{Z}) + 1 \quad (3.15)$$

Algorithm 4 ルーレット選択

1: procedure ROULETTE	SELEC-	Step 3 : 選択
TION($f(\mathbf{x}^i) \in \mathcal{F}$)	5:	for $i = 1, 2, \dots, m$ do
Step 0 : 準備	6:	$\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$
2: $m = \mathcal{F} $ とする。	7:	$S_i = \sum_{j=1}^i p_j$
Step 1 : 適合度の計算	8:	if $\theta \leq S_i$ then
3: 適合度 fit_i を式 (3.13) を用いて計算する。	9:	return i
Step 2 : 選択確率の計算	10:	end if
4: 選択確率 p_i を式 (3.12) を用いて計算する。	11:	end for
	12:	end procedure

ここで, $\mathbb{1}_{\{P\}}(A)$ は指示関数 $A \rightarrow \{0, 1\}$ であり, 集合 A について条件 P を満たす場合 1, そうでなければ 0 を返す。 $R_i(\mathcal{Z})$ はある解集合 \mathcal{Z} に対応する目的関数值に基づき $f(\mathbf{x}^1) < f(\mathbf{x}^2) < \dots < f(\mathbf{x}^m)$ となるように探索点番号を並び替えたときの \mathbf{x}^i の順位を返す関数である。交叉の方法は, 1 点交叉や一様交叉などが挙げられる。突然変異は交叉で生成された解を一定確率で多少変動させ, 近傍解（子集団）とする。淘汰は近傍解の集団（子集団）から次世代の探索点群（親集団）を選択する。淘汰においてもランダム選択やルーレット選択などが使用される。淘汰において選択の対象 \mathcal{Z} の設定方法はいくつかあるが, 通常は探索点群 $\mathcal{X}(k)$ と近傍解集合 $\hat{\mathcal{X}}(k)$ の和集合 $\mathcal{Z} = \mathcal{X}(k) \cup \hat{\mathcal{X}}(k)$ とするのが一般的である。

[Algorithm 3.4.1](#) にルーレット選択の疑似コードを示す。なお, ランキング選択は [Algorithm 3.4.1](#) の **Step 1** で使用する適合度の式を式 (3.14) に置き換えればよい。

3.4.2 実数値 Genetic Algorithm

最初の GA はバイナリ変数で構成した組合せ最適化問題向けのアルゴリズムであったが, 1990 年以降, 実数値最適化問題向けの実数値 GA (Real-Coded GA) が盛んに開発

された。実数値 GA の交叉は、最初は組合せ最適化に由来するビットストリング交叉[59]が使用されていたが、実数空間において自由なトポロジーにより解を生成できるため、これまでに多様な方法が提案されている。このように、多様な交叉方法があることから、突然変異は使用されず、交叉と選択のみで構成することが多い。交叉の方法として、Blend Crossover (BLX- α) [164], Simulated Binary Crossover (SBX) [165], Unimodal Normal Distribution Crossover (UNDX) [166], Simplex Crossover (SPX) [167], Parent Centric Crossover (PCX) [168], Parent Centric Blend Crossover (PBX- α) [169], Real-Coded Ensemble Crossover (REX) [170]など、多くの方法がある。なお、これらの交叉方法の分類として、子個体集団の中心が親個体集団の重心と一致する「Mean Centric Crossover」と、子個体集団の中心がいずれかの親個体と一致する「Parent Centric Crossover」がある[168]。BLX- α , UNDX, SPX, REX は前者に、SBX, PCX, PBX- α は後者に分類される。なお、REX は Mean Centric Crossover を一般化したものである。

ここでは、UNDX と PCX の交叉について述べる。UNDX における近傍生成は式(3.16)で表される。

$$\hat{\mathbf{x}}^i(k) = \frac{1}{2}(\mathbf{x}^j(k) + \mathbf{x}^h(k)) + \alpha \xi_1 (\mathbf{x}^h(k) - \mathbf{x}^j(k)) + \beta D \sum_{q=1}^{N-1} \xi_{2,q} \mathbf{a}^q \quad (3.16)$$

ただし、 $\xi_1, \xi_{2,q} \sim \mathcal{N}(0, 1)$ は標準正規乱数、 \mathbf{a}^q は $\mathbf{x}^h(k) - \mathbf{x}^j(k)$ に直交する部分空間の正規直交基底ベクトル、 $\alpha, \beta > 0$ はパラメータである³。 D は $\mathbf{x}^\ell(k)$ から $\mathbf{x}^h(k) - \mathbf{x}^j(k)$ に垂直に下ろした法線ベクトルの L^2 ノルムである。ここで、一般に、 \mathbf{x}^1 から $\mathbf{x}^2 - \mathbf{x}^3$ に垂直に下ろした法線ベクトルの L^2 ノルムを $D(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$ とすると、 $D(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$ は式(3.17)で表されるため、UNDX においては、 $D = D(\mathbf{x}^\ell(k), \mathbf{x}^h(k), \mathbf{x}^j(k))$ となる。

$$D^2(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3) = \|\mathbf{x}^1 - \mathbf{x}^3\|_2^2 - \left(\frac{(\mathbf{x}^2 - \mathbf{x}^3)^T (\mathbf{x}^1 - \mathbf{x}^3)}{\|\mathbf{x}^2 - \mathbf{x}^3\|_2} \right)^2 \quad (3.17)$$

PCX における近傍生成は式(3.18)で表される。

$$\hat{\mathbf{x}}^i(k) = \mathbf{x}^j(k) + \alpha \xi_1 (\mathbf{x}^j(k) - \boldsymbol{\mu}(\mathcal{Y})) + \beta \xi_2 D \sum_{q=1, q \neq j}^{\mu} \mathbf{a}^q \quad (3.18)$$

³ 探索点の差分ベクトルは主探索軸、主探索軸の直交部分空間は副探索空間と呼称される。

ただし、 $\xi_1, \xi_2 \sim \mathcal{N}(0, 1)$ は標準正規乱数、 \mathbf{a}^q は各探索点 $\mathbf{x}^\ell(k) \in \mathcal{Y} \setminus \{\mathbf{x}^j(k)\}$ から $\mathbf{x}^j(k) - \boldsymbol{\mu}(\mathcal{Y})$ に直交する部分空間の正規直交基底ベクトル、 $\mathcal{Y} = \{\mathbf{x}^1(k), \dots, \mathbf{x}^\mu(k)\}$ 、 $\boldsymbol{\mu}(\mathcal{Y})$ は平均ベクトル（式 (A.17)）、 $\alpha, \beta > 0$ 、 $2 \leq \mu < m$ はパラメータである。 D は各探索点 $\mathbf{x}^\ell(k)$ から $\mathbf{x}^j(k) - \boldsymbol{\mu}(\mathcal{Y})$ に垂直に下ろした法線ベクトルの L^2 ノルムの平均であるため、 $D = \frac{1}{\mu-1} \sum_{\ell=1, \ell \neq j}^{\mu-1} D_\ell$ 、 $D_\ell = D(\mathbf{x}^\ell(k), \mathbf{x}^j(k), \boldsymbol{\mu}(\mathcal{Y}))$ （式 (3.17)）で表される。

淘汰としては、通常の GA と同様にルーレット選択やトーナメント選択のほかに、世代交代モデルが知られている。代表的な世代交代モデルとして、Minimal Generation Gap (MGG) [171]、Just Generation Gap (JGG) [172]、Generalized Generation Gap (G3) [168] が挙げられる。詳細は割愛するが、淘汰は交叉と並んで重要な要素だと言われている。また、実数値 GA のアルゴリズムの設計指針として機能分担仮説が有名であり、これは多様化・集中化の探索戦略を実現するための条件を整備したものである [173]。

さらに、分布推定アルゴリズム (Estimation of Distribution Algorithm : EDA) もよく知られている [174]。EDA は探索過程で確率分布を学習し、その確率分布に従うサンプリングにより近傍解を生成するアルゴリズムである。上記に挙げた実数値 GA では、探索点分布に対する確率分布を直接使用しないが、機能分担仮説では確率分布を導入することで、その分布に従うサンプリングだと解釈することができる。この関連性から、EDA は実数値 GA の一種と称されることもある。以上のように、実数値 GA は非常にバリエーションが豊富であることから、詳細のアルゴリズムは記載しない。

3.4.3 実数値 Genetic Algorithm の探索構造の解析

定義 3.1 に従って、実数値 GA の更新則を解析する。まず、実数値 GA の近傍生成則 U_{nei} は交叉が該当し、探索点の移動則 U_{mov} は選択・世代交代モデルが該当する。UNDX の場合、近傍生成則 U_{nei} （式 (3.16)）において、 $\mathcal{R}(k) = \{\xi_1, \xi_{2,1}, \dots, \xi_{2,N-1}\}$ 、 $\mathcal{C}(k) = \{\alpha, \beta\}$ 、 $\{\mathbf{x}^j(k), \mathbf{x}^h(k), \mathbf{x}^\ell(k)\} \subseteq \chi(k)$ となる。また、PCX の場合、近傍生成則 U_{nei} （式 (3.18)）において、 $\mathcal{R}(k) = \{\xi_1, \xi_2\}$ 、 $\mathcal{C}(k) = \{\alpha, \beta\}$ 、 $\{\mathbf{x}^j(k), \mathbf{x}^h(k), \mathbf{x}^\ell(k)\} \subseteq \chi(k)$ となる。よって、交叉は探索点群の位置情報に基づく正規直交基底ベクトルと距離によって、探索点分布内に近傍解を生成するという仕組みであることがわかる。さらに、ルーレット選択やランキング選択の場合、探索点の移動則 U_{mov} （式 (3.13)、式 (3.14)）において、 $\mathcal{R}(k) = \{\theta\}$ 、

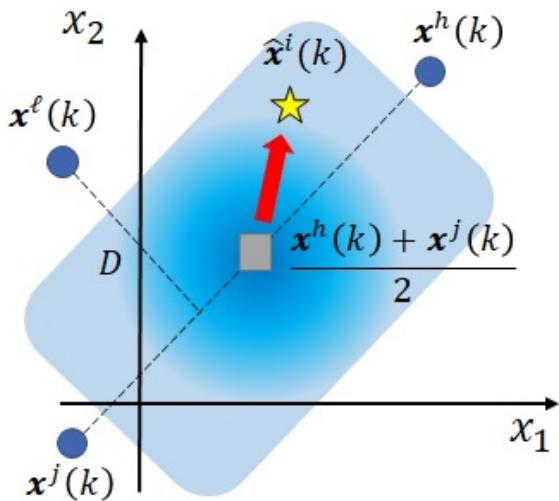


図 3.4： 実数値 GA (UNDX) の近傍の生成

$\mathcal{C}(k) = \emptyset$ となる。ルーレット選択は、目的関数値の大きさに依存し、優れている解がより選択されやすく、ランキング選択は、目的関数値の大小関係に依存し、優れている解がより選択されやすい。

図 3.4 に、UNDX の近傍生成について示し、図 3.5 に、PCX の近傍生成について示す。したがって、実数値 GA は近傍生成則で探索点群のスケールや散らばりなどに応じて近傍解を生成し、交叉における親の選択や、移動則で優れた探索点を優先的に選択するというメカニズムであることがわかる。また、UNDX は機能分担仮説に基づく理論的解析を通じて、探索点分布の多様性と目的関数多峰性に関する性質が解明されている[175]。優れた探索点の選択圧力を高めることや、分布付近の近傍生成が、有望領域に探索点群を移動させる集中化を生み出す操作であり、分布中心と良い探索点の位置のズレに基づき他の領域に拡張させる近傍生成が、多様化を生み出す操作であると考えられる。

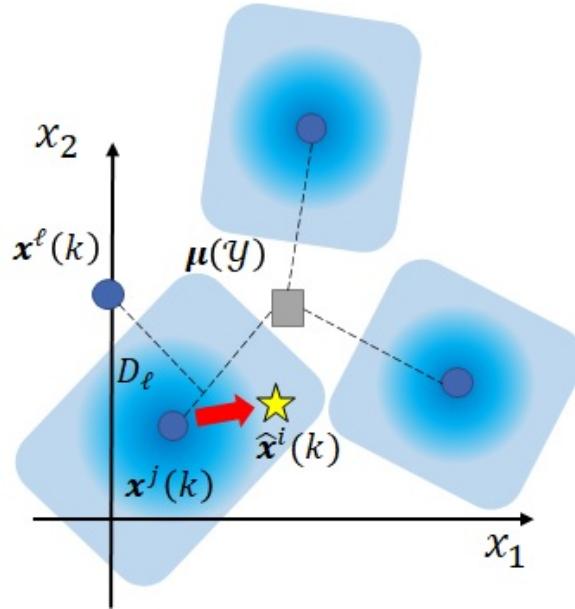


図 3.5： 実数値 GA (PCX) の近傍の生成

3.5 Simulated Annealing の解析

3.5.1 Simulated Annealing

Simulated Annealing (SA) は 1983 年に Kirkpatrick らによって開発された単点探索型メタヒューリстиクスである[63, 64, 65]。SA は金属の焼き鈍し現象に基づいている。SA は単点探索で、更新則は近傍生成と探索点の移動により構成されている。近傍生成は乱数に基づく摂動により局所探索を行う。探索点の移動は、式 (3.19) を用いて行う。

$$\mathbf{x}(k+1) = \begin{cases} \hat{\mathbf{x}}(k), & f(\hat{\mathbf{x}}(k)) < f(\mathbf{x}(k)) \vee P(k) \geq \theta \\ \mathbf{x}(k), & \text{otherwise} \end{cases} \quad (3.19)$$

ただし、 $P(k) \in [0, 1]$ は受理確率、 $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ である。特に、SA では $P(k; T(k))$ は Metropolis の受理確率の式 (3.20) で表される。

$$P(k; T(k)) = \exp \left(-\frac{f(\hat{\mathbf{x}}(k)) - f(\mathbf{x}(k))}{T(k)} \right) \quad (3.20)$$

ただし、 $T(k)$ は温度パラメータであり、その調整則として式 (3.21) が与えられる。

$$T(k+1) = T_{\text{start}} / \ln(k) \quad (3.21)$$

ただし、 T_{start} は温度パラメータの初期値である。なお、SA は Markov 連鎖モンテカルロ法と統計力学によるサンプリングの枠組みとしても位置づけられる。

3.5.2 実数値 Simulated Annealing

最初の SA は組合せ最適化問題向けのアルゴリズムであった。それ以降、実数値最適化問題向けの実数値 SA (Real-Coded SA) が開発された。実数値 SA における近傍生成は ES と同様に、式 (3.22) で表される。

$$\hat{\mathbf{x}}(k) = \mathbf{x}(k) + \sigma \mathbf{s}(k) \quad (3.22)$$

ただし、 $\sigma > 0$ はパラメータ、 $\mathbf{s}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ は標準正規乱数ベクトルである。探索点の移動は通常の SA と同様に式 (3.19) を用いて行う。

Algorithm 3.5.2 に実数値 SA の疑似コードを示す。実数値 SA の終了条件は反復回数であり、評価回数は $T = k_{\max}$ となる。

3.5.3 実数値 Simulated Annealing の探索構造の解析

定義 3.1 に従って、実数値 SA の更新則を解析する。まず、実数値 SA の近傍生成則 U_{nei} は式 (3.22) が該当し、探索点の移動則 U_{mov} は確率的移動の式 (3.19) が該当する。よって、受理確率 $P(k)$ が固定される場合、実数値 SA 全体の更新則 U (式 (3.4)) において、 $\mathcal{R}(k) = \{\mathbf{s}(k), \theta\}$, $\mathcal{C}(k) = \{\sigma\}$, $\mathbf{x}(k) \in \chi(k)$ となる。以上から、探索点自身の位置情報と乱数ベクトルによる摂動によって近傍解を生成し、改善するか受理確率 $P(k)$ を満たす場合に、探索点を移動させる、という仕組みであることがわかる。さらに、実数値 SA は、上記に受理確率 $P(k)$ の調整則 (式 (3.20)) と温度パラメータの調整則 $T(k)$ (式 (3.21)) が附加されたものだが、パラメータ調整則の $P(k), T(k), T_{\text{start}}$ などの量も更新に影響する。こ

Algorithm 5 実数値 Simulated Annealing

1: procedure 実数値 SIMULATED AN-	6: $\hat{x}(k)$ を式 (3.22) を用いて生成する。
NEALING	Step 3 : 解の選択
Step 0 : パラメータ設定	7: $P(k)$ を式 (3.20) を用いて計算する。
2: パラメータ k_{\max} , σ 、 T_{start} を設定する。	8: $x(k)$ を式 (3.19) を用いて更新する。
Step 1 : 初期化	Step 4 : 終了判定
3: 探索点の位置 $x(k)$ を初期配置領域 S_0 内にランダムに配置する。	9: if $k = k_{\max}$ then
4: $T(k) = T_{\text{start}}$ を設定し、式 (3.21) を用いて $T(k)$ のスケジュールを与える。	10: 計算を終了する。
5: $k = 1$ とする。	11: else
Step 2 : 近傍生成	12: $k := k + 1$ とし、 Step 2 へ戻る。
	13: end if
	14: end procedure

のように、パラメータ調整則が含まれる場合は、そのパラメータの状態も更新則に影響すると考えられる。パラメータ調整と更新則の関連に関する詳細は第 5 章で述べる。

実数値 SA の近傍生成は ES と同様なので、図 3.3 である。確率的移動である点を除けば、実数値 SA は ES と類似しているが、この確率的移動が SA の大きな特徴である。受理確率 $P(k)$ は式 (3.20) に従うため、探索点と近傍解の目的関数値の差が大きければ $P(k)$ が低くなり、差が小さければ $P(k)$ が高くなるように計算される。よって、近傍解が改善しない場合でも、解の悪化が大きくなれば、探索点の移動により悪化することも許している。これにより、改善移動に基づく集中化に加えて、悪化の許容に基づく多様化が含まれていることがわかる。さらに、式 (3.20) の受理確率 $P(k)$ は、温度パラメータ $T(k)$ が大きければ受理確率も高くなり、小さければ受理確率は低くなる。 $(1+1)$ -ES は通常の摂動に加えて、 $1/5$ ルールに従いパラメータ $\sigma(k)$ を動的に調整することで、摂動範囲の拡大・縮小が可能だが、SA も受理確率 $P(k)$ の温度パラメータ $T(k)$ の調整則として、式 (3.21) が付加されている。この調整則は、温度パラメータ $T(k)$ を探索が進むにつれて徐々に減少させていくスケジュールである。このアイディアは、金属の徐冷過程に基づいており、徐々に温度を下げ、内部エネルギーを低くすることで、物理状態が変動する範囲を徐序に狭めていく操

作に該当する。つまり、探索序盤では、温度パラメータ $T(k)$ を高くすることで、悪化の許容も積極的に促し（多様化）、探索終盤では、温度パラメータ $T(k)$ を低くすることで、改善移動のみに限定（集中化）する、という効果がある。つまり、基本的に局所探索を行うため、最適化能力を有していると同時に、パラメータ調整則により、探索過程で多様化から集中化に移行するという探索戦略を実現している。以上から、SA はパラメータ調整則によって探索戦略に基づく最適化能力を実現していると考えられる。

3.6 Particle Swarm Optimization の解析

3.6.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) は、1995 年に Kennedy と Eberhart により開発された多点探索型メタヒューリスティクスである[53, 54, 68]。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。PSO の更新則は式 (3.23), 式 (3.24) で表される⁴。

$$\mathbf{v}^i(k+1) = w\mathbf{v}^i(k) + c_1 \mathbf{R}_1(\mathbf{p}^i(k) - \mathbf{x}^i(k)) + c_2 \mathbf{R}_2(\mathbf{p}^g(k) - \mathbf{x}^i(k)) \quad (3.23)$$

$$\mathbf{x}^i(k+1) = \mathbf{x}^i(k) + \mathbf{v}^i(k+1) \quad (3.24)$$

ただし、 $w, c_1, c_2 > 0$ は PSO のパラメータ、 $\mathbf{R}_j = \text{diag}[R_{j1}, R_{j2}, \dots, R_{jN}]$ ($j = 1, 2$)、 $R_{jn} \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ である。特に、 \mathbf{v} を速度ベクトル、 w を慣性パラメータ、式 (3.23) の第一項を慣性項と呼ぶ。 \mathbf{p}^i は式 (2.15) の p-best、 \mathbf{p}^g は式 (2.16) の g-best である。なお、PSO の探索過程では、p-best を逐次更新させる式 (3.25) で表すことが多いが、式 (2.15) と等価である。

$$\mathbf{p}^i(k+1) = \begin{cases} \mathbf{x}^i(k+1), f(\mathbf{x}^i(k+1)) < f(\mathbf{p}^i(k)) \\ \mathbf{p}^i(k), \text{otherwise} \end{cases} \quad (3.25)$$

⁴ PSO の更新式にはいくつかの種類がある[176, 177]。代表的な更新式としては、Eberhart らにより 1996 年に提案された IWM (Inertia Weight Model) と、Eberhart と Shi によって修正された Constriction Coefficient Model (CCM) がある。本論文では、IWM を用いる。なお、PSO では探索点群の速度発散を防ぐために、速度上限 (V_{\max}) を制約として課すことがある。

Algorithm 6 Particle Swarm Optimization

```

1: procedure PARTICLE SWARM OPTI- 9:       $\mathbf{v}^i(k)$  を式 (3.23) を用いて更新す
   MIZATION                               る。
2:   Step 0 : パラメータ設定           10:       $\mathbf{x}^i(k)$  を式 (3.24) を用いて更新
   2:     パラメータ  $m$ 、 $k_{\max}$ 、 $w$ 、 $c_1$ 、 $c_2$  を設    する。
   定する。                                11:       $\mathbf{p}^i(k)$  を式 (3.25) を用いて更新
   3:     探索点の位置  $\mathbf{x}^i(k)$  を初期配置領域  $S_0$  12:      end for
   内にランダムに配置する。                13:       $\mathbf{p}^g(k)$  を式 (2.16) を用いて更新する。
   4:     探索点の速度  $\mathbf{v}^i(k)$  をランダムに与    Step 3 : 終了判定
   える。                                14:      if  $k \geq k_{\max}$  then
   5:      $\mathbf{p}^i(k) = \mathbf{x}^i(k)$  とする。          15:        計算を終了する。
   6:      $\mathbf{p}^g(k)$  を式 (2.16) を用いて計算する。 16:      else
   7:      $k = 1$  とする。                      17:         $k := k + 1$  とし、Step 2 へ戻る。
   8:     for  $i = 1, 2, \dots, m$  do           18:      end if
                                         19: end procedure

```

Algorithm 3.6.1 に PSO の疑似コードを示す。PSO の終了条件は反復回数であり、評価回数は $T = m(k_{\max} + 1)$ となる。

3.6.2 Pariticle Swarm Optimization の探索構造の解析

定義 3.1 に従って、PSO の更新則を解析する。PSO の全体の更新則 U (式 (3.4)) は、式 (3.23) と式 (3.24) を連立させた式 (3.26) である。

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

更新則 U において、 $\mathcal{R}(k) = \{\mathbf{R}_1, \mathbf{R}_2\}$, $\mathcal{C}(k) = \{c_1, c_2, w\}$, $\{\mathbf{x}^i(k), \mathbf{x}^i(k-1), \mathbf{p}^i(k), \mathbf{p}^g(k)\} \subseteq \chi(k)$ となる。図 3.6 に、PSO の近傍生成について示す。探索点から p-best と g-best へ向

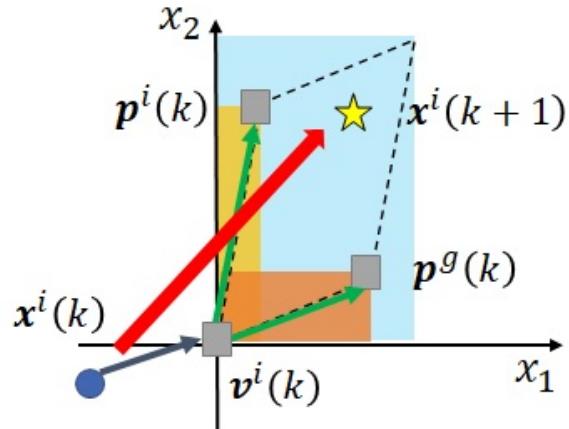


図 3.6 : Particle Swarm Optimization の近傍の生成

かう差分ベクトル, 慣性項による摂動, 係数の乱数によって, 近傍を生成し探索点を移動させる, という仕組みであることがわかる。また, PSO は絶対移動であるため, 探索点 $\mathbf{x}^i(k)$ は近傍解 $\hat{\mathbf{x}}^i(k)$ へ必ず更新する。しかし, $\mathbf{p}^g(k)$ が移動する場合, $f(\mathbf{p}^g(k+1)) \leq f(\mathbf{p}^g(k))$ が成立し, と $\mathbf{p}^i(k)$ が移動する場合, $f(\mathbf{p}^i(k+1)) \leq f(\mathbf{p}^i(k))$ が成立する。これは, 探索点は必ず改善する位置に移動せず, 改悪も許容するが, g-best と p-best は目的関数値が改善する場合に移動する「改善移動」であることを意味する。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。したがって, PSO は探索点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することが, 集中化の能力を生み出している。さらに, 慣性項と係数の乱数が上記の近傍生成に摂動を与えることで, 多様化の能力を生み出している。

メタヒューリстиクスは元々発見的手法だが, 先行研究では, PSO に対していくつか理論的・実験的な解析が行われ, その探索ダイナミクス (特に動特性) が明らかになりつつある [113, 153, 154, 155]。代表的な近似モデルとして, 離散時間線形時不変システムとしてみなした縮約モデルが知られている。まず, 式 (3.23) を式 (3.27) に変形する。

$$\mathbf{v}^i(k+1) = \Phi(\mathbf{z}^i(k) - \mathbf{x}^i(k)) + w\mathbf{v}^i(k) \quad (3.27)$$

ただし、 Φ , $\mathbf{z}^i(k)$ は式 (3.28), 式 (3.29) である。

$$\Phi = c_1 \mathbf{R}_1 + c_2 \mathbf{R}_2 \quad (3.28)$$

$$\mathbf{z}^i(k) = \Phi^{-1}(c_1 \mathbf{R}_1 \mathbf{p}^i(k) + c_2 \mathbf{R}_2 \mathbf{p}^g(k)) \quad (3.29)$$

Φ は対角行列の線形結合で表される乱数行列であるため、 $\Phi = \text{diag}[c_1 R_{11} + c_2 R_{21}, \dots, c_1 R_{1N} + c_2 R_{2N}]$ となる。縮約モデルは、式 (3.27), 式 (3.28), 式 (3.29) において、 $N = 1$ とし、p-best と g-best を定数としたものである。 $\mathbf{z}^i(k)$ は p-best と g-best を $c_2 R_2 : c_1 R_1$ の比に内分する点であり、優れた解とみなせる。また、 $\mathbf{y}^i(k) = \mathbf{z}^i(k) - \mathbf{x}^i(k)$ とおくとき、式 (3.27) は、有望方向に向かうベクトル $\mathbf{y}^i(k)$ を乱数行列で重みづけしたベクトルと慣性項の和で表される。式 (2.19) の勾配法に慣性項を付加した手法をモーメント法と呼ぶが、この縮約モデルは確率的なステップサイズを持ったモーメント法と類似しているとみなすことができる。

また、このモデルに変形することで、システム表現が可能となり、固有値解析による安定性解析が容易となる。これにより、PSO のパラメータ空間 $\{w, c = c_1 = c_2\}$ 内には安定／不安定領域が存在し、パラメータ設定が位置する領域によって安定性が決定されることがわかる [116, 153, 154, 155]。不安定である場合、探索点群は探索過程で速度を増しながら、発散していく。安定である場合、探索点群は p-best と g-best の内分点に収束していく。そして、パラメータを不安定領域に近い安定領域（弱い安定領域）に設定した場合、探索序盤では探索点群が解空間内を大きく動き回り、探索過程で徐々に収束していく。Constriction Method (CM) は、PSO のパラメータを $\{w, c = c_1 = c_2\} = \{0.729, 1.4955\}$ と設定する方法で、弱い安定領域に位置する [176, 177]。このように、PSO はパラメータが不安定領域の内部に近く位置するほど強い多様化、安定領域の内部に近く位置するほど強い集中化が生じる。さらに、弱い安定領域に位置すれば、探索過程で多様化から集中化に徐々に移行する探索戦略を実現できる。以上のように、PSO は動的なパラメータ調整がなくても、安定性を考慮して設定すれば、多様化・集中化の探索戦略を実現できる点が最適化能力を生み出していると考えられる。

3.7 Differential Evolution の解析

3.7.1 Differential Evolution

Differential Evolution (DE) は、1995 年に Storn と Price により開発された多点探索型メタヒューリスティクスである[69]。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。DE の突然変異や交叉は様々な方法があり、 $DE/X/Y/Z$ という呼称で具体的な方法を区別して表現する。 X は突然変異における解の選択方法、 Y は突然変異における差分ベクトルの数、 Z は交叉方法である。下記では、最も基本的な $DE/\text{rand}/1/\text{bin}$ を取り挙げる。DE の更新則は式 (3.30)，式 (3.31)，式 (3.32) を用いて表される。

$$\mathbf{u}^i(k) = \mathbf{x}^{r_1}(k) + F(\mathbf{x}^{r_2}(k) - \mathbf{x}^{r_3}(k)) \quad (3.30)$$

$$\hat{x}_n^i(k) = \begin{cases} u_n^i(k), & \theta < CR \cup n = \lambda \\ x_n^i(k), & \text{otherwise} \end{cases} \quad (3.31)$$

$$\mathbf{x}^i(k+1) = \begin{cases} \hat{\mathbf{x}}^i(k), & f(\hat{\mathbf{x}}^i(k)) < f(\mathbf{x}^i(k)) \\ \mathbf{x}^i(k), & \text{otherwise} \end{cases} \quad (3.32)$$

ただし、 $F, CR \geq 0$ はパラメータ、 $r_1, r_2, r_3 \sim \mathcal{U}_{\mathbb{Z}}(1, m)$ ($r_1 \neq r_2 \neq r_3 \neq i$)、 $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ 、 $\lambda \sim \mathcal{U}_{\mathbb{Z}}(1, N)$ に従う乱数である。

式 (3.30) は $\text{rand}/1$ の突然変異を表し、ランダムに選択した探索点を基本位置ベクトルとし、基本位置ベクトルと 1 つの差分ベクトルの線形結合で突然変異ベクトル $\mathbf{u}^i(k)$ を生成している。 X/Y が $\text{best}/1$ の場合、式 (3.30) の基本位置ベクトルを c-best とする。 X/Y が $\text{current-to-rand}/1$ の場合、式 (3.30) に探索点 $\mathbf{x}^i(k)$ からランダムに選択した探索点に向かう差分ベクトルを追加する。

式 (3.30) は $\text{bin}/1$ の交叉を表し、一様交叉 (Binomial Crossover) と呼ばれる操作である。各要素について、交叉率 CR を満たす場合、あるいは特定の要素の場合は突然変異ベクトル $\mathbf{u}^i(k)$ の要素を、そうでなければ探索点 $\mathbf{x}^i(k)$ の要素を組み込むことで、近傍解 $\hat{\mathbf{x}}^i(k)$ を生成する。 Z が exp の場合、指数交叉 (Exponential Crossover) と呼ばれる操作を行うが、先行研究では、指数交叉が変数間依存性を有する中でも特殊な関数のみの非現実的な性質を仮定・利用していることから、BBO では望ましい操作でないことが指摘されて

Algorithm 7 Differential Evolution

```

1: procedure DIFFERENTIAL EVOLUTION
   8:   end for
   TION
   Step 3: 位置の更新
Step 0: パラメータ設定
2:   パラメータ  $m$ 、 $k_{\max}$ 、 $F$ 、 $CR$  を設定
   9:   for  $i = 1, 2, \dots, m$  do
      10:      式(3.21)を用いて、 $\mathbf{x}^i$  を更新する。
         11:   end for
Step 1: 初期化
3:   探索点の位置  $\mathbf{x}^i(k)$  を初期配置領域  $S_0$ 
   内にランダムに配置する。
   12:   if  $k \geq k_{\max}$  then
      13:      計算終了する。
      14:   else
Step 2: 突然変異と交叉
5:   4:    $k = 1$  とする。
   5:   for  $i = 1, 2, \dots, m$  do
      6:      突然変異により、 $\mathbf{y}^i$  を生成する。
      7:      交叉により、近傍解  $\hat{\mathbf{x}}^i$  を生成する。
      15:       $k := k + 1$  とし、Step 2 に戻る。
      16:   end if
      17: end procedure

```

いる[178]。なお、DE の交叉は GA に着想を得ているが、実際には組合せ最適化向け GA の一様交叉や 1 点交叉を使用するため、実数空間のトポロジーを活用しているとは言い難い側面もある。このため、DE の交叉方法として、実数値 GA の多様な交叉を使用することも有効だと考えられる。

Algorithm 3.7.1 に DE の疑似コードを示す。評価回数は $T = m(k_{\max} + 1)$ となる。

3.7.2 Differential Evolution の探索構造の解析

定義 3.1 に従って、DE の更新則を解析する。まず、式(3.31)を多変量で表現すると、式(3.33)、式(3.34)に変形できる[126]。

$$\hat{\mathbf{x}}^i(k) = \mathbf{M}^1 \mathbf{x}^i(k) + \mathbf{M}^2 \mathbf{u}^i(k) \quad (3.33)$$

$$\mathbf{M}^1 + \mathbf{M}^2 = \mathbf{I} \quad (3.34)$$

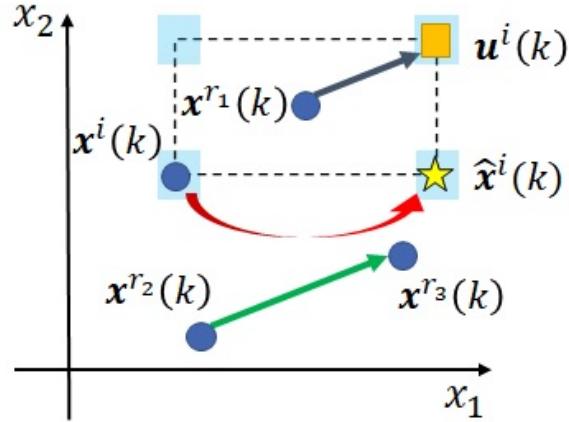


図 3.7： Differential Evolution の近傍の生成 (DE/rand/1/bin)

ただし， $\mathbf{M}^1, \mathbf{M}^2 \in \mathbb{R}^{N \times N}$ は対角行列， $\mathbf{I} \in \mathbb{R}^{N \times N}$ は単位行列である。 \mathbf{M}^1 の対角要素は $\{0, 1\}$ の値をとり，値は θ, λ で決まる。

DE の近傍生成則 U_{nei} は突然変異（式 (3.30)）と交叉（式 (3.33)）が該当し，探索点の移動則 U_{mov} は改善移動の式 (3.32) が該当する。よって，DE 全体の更新則 U において， $\mathcal{R}(k) = \{r_1, r_2, r_3, \theta, \lambda\}$ ， $\mathcal{C}(k) = \{F, CR\}$ ， $\{\mathbf{x}^i(k), \mathbf{x}^{r1}(k), \mathbf{x}^{r2}(k), \mathbf{x}^{r3}(k)\} \subseteq \chi(k)$ となる。よって，探索点間の差分ベクトルと交叉による組合せによって近傍解を生成し，改善する場合に，探索点を移動させる，という仕組みであることがわかる。

図 3.7 に，DE の近傍生成について示す。突然変異では，探索点間の差分ベクトルが探索点群のスケールや分布の情報を含む揃動であると同時に，その揃動を $F \in (0, 1)$ によって縮小させるため，突然変異ベクトル $\mathbf{u}^i(k)$ は集中化に貢献していると考えられる。一方，交叉では，突然変異ベクトル $\mathbf{u}^i(k)$ と探索点 $\mathbf{x}^i(k)$ を組み替えることで，有望な位置と探索点の位置の範囲で多様な位置に近傍生成しているため，多様化に貢献していると考えられる。さらに，改善移動であるため，探索点群は降下方向に改善していくことで，有望領域に集中化していく。また，突然変異において $F \in [0, 1]$ を小さくすれば，探索点群内の狭い領域に近傍解が生成されやすく，大きくすれば，探索点群と同じスケールの領域（あるいは探索点群を少しだけ覆う領域）に近傍解が生成されやすい。交叉において $CR \in [0, 1]$ を小さくすれば，現在の探索点の領域付近に近傍解が生成されやすく，大きくすれば，突

然変異ベクトルの領域付近に近傍解が生成されやすい。以上の点から、DEは多様化・集中化の能力を有していると解釈でき、パラメータによってその多様化・集中化を調整可能だと考えられる。

DEはPSOよりも探索ダイナミクスの解析が困難であることが知られているが、先行研究では、理論・実験的にその一部の傾向は明らかにされている[151, 152, 156, 178]。

3.8 Artificial Bee Colony Algorithm の解析

3.8.1 Artificial Bee Colony Algorithm

Artificial Bee Colony Algorithm (ABC) は2005年にKarabogaにより開発された多点探索型メタヒューリスティクスである[70]。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。働き蜂の探索では、 $m_e \in \mathbb{N}$ 個の探索点が探索を行う。働き蜂の更新則を式(3.35), 式(3.36), 式(3.37)に示す。

$$\hat{x}_n^i = \begin{cases} x_n^i + \phi^i(x_n^r - x_n^i), & n = \lambda \\ x_n^i, & \text{otherwise} \end{cases} \quad (3.35)$$

$$\mathbf{x}^i := \begin{cases} \hat{\mathbf{x}}^i, & f(\hat{\mathbf{x}}^i) < f(\mathbf{x}^i) \\ \mathbf{x}^i, & \text{otherwise} \end{cases} \quad (3.36)$$

$$C_i := \begin{cases} 0, & f(\hat{\mathbf{x}}^i) < f(\mathbf{x}^i) \\ C_i + 1, & \text{otherwise} \end{cases} \quad (3.37)$$

ただし、 $r \sim \mathcal{U}_{\mathbb{Z}}(1, m)$, $r \neq i$, $\phi \sim \mathcal{U}_{\mathbb{R}}(-1, 1)$, $\lambda \sim \mathcal{U}_{\mathbb{Z}}(1, N)$, C_i は探索点 \mathbf{x}^i の停滞回数である。

傍観蜂の探索では、探索点群の目的関数値集合 \mathcal{F} に対して、 $m_o \in \mathbb{N}$ 回ルーレット選択 (Algorithm 3.4.1) を行い、選ばれた探索点が探索を行う。更新則は働き蜂と同様に、式(3.35), 式(3.36), 式(3.37)である。

斥候蜂の探索では、停滞回数 C_i が事前に設定した規定回数 $Limit \in \mathbb{N}$ よりも上回った場合のみ、その探索点を初期配置領域内にランダムに配置する。初期配置領域 S_0 が超直方

体 $S_0 = [\gamma_{\min,n}, \gamma_{\max,n}]^N \subset \mathbb{R}^N$ として与えられる場合、近傍生成則は式(3.38)、式(3.39)で表され、移動則は絶対移動である。

$$\mathbf{x}^i := \mathbf{M}^1 \boldsymbol{\gamma}_{\max} + \mathbf{M}^2 \boldsymbol{\gamma}_{\min} \quad (3.38)$$

$$\mathbf{M}^1 + \mathbf{M}^2 = \mathbf{I} \quad (3.39)$$

ただし、 $\boldsymbol{\gamma}_{\max}, \boldsymbol{\gamma}_{\min}$ は初期配置領域 S_0 の角のベクトル、 $\mathbf{M}^1, \mathbf{M}^2 \in \mathbb{R}^{N \times N}$ は対角行列、 $\mathbf{I} \in \mathbb{R}^{N \times N}$ は単位行列である。 $M_n^2 \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ ($n = 1, 2, \dots, N$) であり、 M_n^1 は式(3.39)を満たすように決定されるため、 $M_n^1 \in [0, 1]$ となる。その後、配置し直した探索点に対応する停滞回数を $C_i := 0$ とリセットする。

Algorithm 3.8.1 にABCの疑似コードを示す。なお、ABCでは終了条件が反復回数ではなく、評価回数であるため、探索点の表記も \mathbf{x}^i としている。働き蜂数 m_e が他のメタヒューリスティクスにおける探索点数としてみなす。

なお、ABCは先行研究では、PSOや後述するCuckoo Searchよりも探索性能が優れていると検証されている[179]。

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

定義 3.1 に従って、ABCの更新則を解析する。まず、式(3.35)を多変量で表現すると、式(3.40)、式(3.41)、式(3.42)に変形できる。

$$\hat{\mathbf{x}}^i = \mathbf{M}^1 \mathbf{x}^i + \mathbf{M}^2 \mathbf{u}^i \quad (3.40)$$

$$\mathbf{M}^1 + \mathbf{M}^2 = \mathbf{I} \quad (3.41)$$

$$\mathbf{u}^i = \mathbf{x}^i + \phi(\mathbf{x}^r - \mathbf{x}^i) \quad (3.42)$$

ただし、 $\mathbf{M}^1, \mathbf{M}^2 \in \mathbb{R}^{N \times N}$ は対角行列、 $\mathbf{I} \in \mathbb{R}^{N \times N}$ は単位行列である。 \mathbf{M}^2 の対角要素はいずれか一つだけが 0、それ以外は全て 1 であり、その位置は λ で決まる。なお、この変形は[126]を参考にしているが、ABCの更新則をこの形式で表現している先行研究は、著者の知る限りない。

Algorithm 8 Artificial Bee Colony Algorithm

```

1: procedure ARTIFICIAL BEE COLONY      を更新する。
   ALGORITHM
   Step 0: パラメータ設定           12:      式(3.37)を用いて、 $C_i$ を更新する。
   2:       $m_e$ 、 $m_o$ 、 $T_{\max}$ 、 $Limit$ を設定する。    Step 4: 戾候蜂
   Step 1: 初期化                  13:      end for
   3:       $\mathbf{x}^i \in S_0 (i = 1, 2, \dots, m_e)$ をランダム 14:      for  $i = 1, 2, \dots, m_e$  do
   に与える。                           15:      if  $C_i \geq Limit$  then
   4:       $C_i := 0$ とする。             16:           $\mathbf{x}^i$ を $S_0$ にランダムに再配置
   5:      for  $i = 1, 2, \dots, m_e$  do        17:           $C_i := 0$ とする。
   6:          式(3.35)、式(3.36)を用いて、 $\mathbf{x}^i$  18:      end if
   を更新する。                         Step 5: 終了判定
   7:          式(3.37)を用いて、 $C_i$ を更新する。 20:      if  $T \geq T_{\max}$  then
   8:      end for                      21:          計算終了する。
   Step 3: 傍観蜂                 22:      else
   9:      for  $i = 1, 2, \dots, m_o$  do        23:          Step 2に戻る。
   10:         ルーレット選択により、探索点 $\mathbf{x}^i$  24:      end if
   を選択する。                         25: end procedure
   11:         式(3.35)、式(3.36)を用いて、 $\mathbf{x}^i$ 

```

働き蜂の近傍生成則 U_{nei} は式(3.41)、式(3.42)が該当し、探索点の移動則 U_{mov} は改善移動の式(3.36)が該当する。よって、働き蜂全体の更新則 U において、 $\mathcal{R}(k) = \{r, \phi\}$ 、 $\mathcal{C}(k) = \emptyset$ 、 $\{\mathbf{x}^i, \mathbf{x}^r\} \subseteq \chi(k)$ となる。よって、探索点間の差分ベクトルと解の組合せによって近傍解を生成し、改善する場合に、探索点を移動させる、という仕組みであることがわかる。傍観蜂の近傍生成則 U_{nei} と探索点の移動則 U_{mov} は働き蜂と同様だが、探索点 \mathbf{x}^i の選択方法はルーレット選択であるため、目的関数値が優れた探索点ほど行いやすい点で異なる。よって、傍観蜂の全体の更新則 U において、 $\mathcal{R}(k) = \{r, \phi, \theta\}$ 、 $\mathcal{C}(k) = \emptyset$ 、 $\{\mathbf{x}^i, \mathbf{x}^r\} \subseteq \chi(k)$

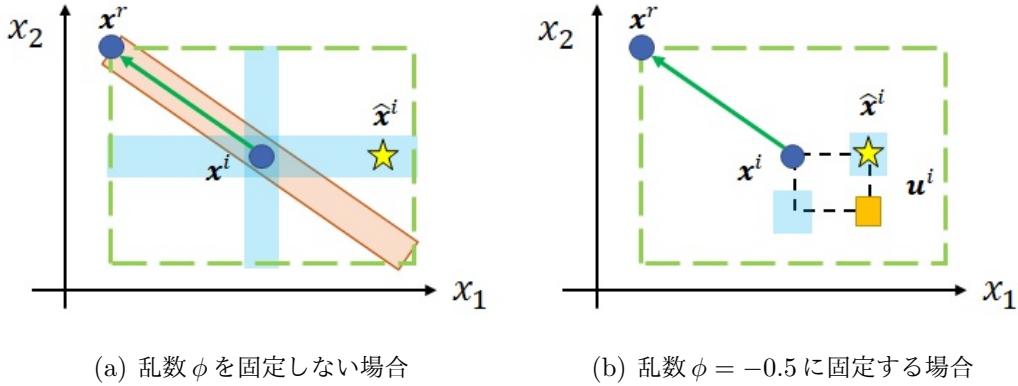


図 3.8 : Artificial Bee Colony Algorithm の近傍の生成

となる。斥候蜂の近傍生成則はランダム配置で、更新則は絶対移動である。斥候蜂では、停滞回数が多い、つまり有望でない探索点が移動する。よって、斥候蜂の全体の更新則 U において、 $\mathcal{R}(k) = \{M_1^2, M_2^2, \dots, M_N^2\}$, $\mathcal{C}(k) = \text{Limit}$, $\gamma_{\max}, \gamma_{\min} \in \chi(k)$ となる。

図 3.8 に、ABC の近傍生成について示す。ABCにおいて、近傍生成と探索点の移動は、主に働き蜂と傍観蜂が行う。働き蜂と傍観蜂の近傍生成（式 (3.42)）において、探索点間の差分ベクトルは探索点群のスケールや分布の情報を含む摂動であると同時に、その摂動を乱数 $\phi \in [-1, 1]$ によって拡大・縮小させるため、DE の突然変異ベクトルと同様に、 u^i は集中化、あるいは弱い多様化に貢献していると考えられる。一方、式 (3.41) は、DE の交叉と同様に、探索点 x^i と u^i を組み合わせることで、有望な位置と探索点の位置の範囲で近傍解を生成している。ただし、 u^i の要素は一つしか採用されないため、近傍解は解空間の中で探索点 x^i を基準として、单一の標準基底に沿って、摂動が加えられた位置に生成される。さらに、改善移動であるため、探索点群は降下方向に改善し、有望領域に集中化していく。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せで、近傍解 \hat{x}^i が生成される。特に、傍観蜂においては、選択される探索点は比較的優れているため、優れた探索点付近に近傍解が生成されやすくなる。一方、これらを繰り返すと、探索点群の内部に近傍解が生成されやすい、つまり徐々に集中化しやすい。そこで、斥候蜂では、改善が成功しない、つまり有望領域が付近にない探索点をランダムに再配置するという点が多様化に貢献している。以上の点から、ABC は各蜂において多様化・集中化の能力・機能を分担して実行していると考えられる。

3.9 Firefly Algorithm の解析

3.9.1 Firefly Algorithm

Firefly Algorithm (FA) は、2008年に Xin-She Yangにより開発された、多点探索型メタヒューリスティクスである[71, 72]。FA は螢の求愛行動のメカニズムに基づいており、強い光を発する螢に近づく探索を行う。FA の各探索点 $\mathbf{x}^i(k)$ は固有の光強度 $I^i(k)$ を有しており、光強度の非減少順に探索点番号をソートする。各探索点 $\mathbf{x}^i(k)$ は、光強度が相対的に強い他の探索点 $\mathbf{x}^s(k)$ を光強度の非減少順に参照して移動する。最も光強度の強い探索点は、乱数ベクトルが加わって移動する。ただし、探索点の評価は、全ての探索点が移動し終えた後であるため、実際には光強度を計算するタイミングで探索点群を $\mathcal{Y} = \mathcal{X}(k)$ として保存しておく、参照する解は $\mathbf{y}^s \in \mathcal{Y}$ とする。光強度 $I^i(k)$ は式 (3.43) として定義される⁵。

$$I^i(k) = \begin{cases} \frac{1}{1+f(\mathbf{x}^i(k))}, & f(\mathbf{x}^i(k)) \geq 0 \\ 1 + |f(\mathbf{x}^i(k))|, & \text{otherwise} \end{cases} \quad (3.43)$$

FA の更新則を式 (3.44), 式 (3.45), 式 (3.46) に示す。

$$\mathbf{x}^i(k) := \mathbf{x}^i(k) + \beta_i(\mathbf{y}^s - \mathbf{x}^i(k)) + \alpha \mathbf{s}^i(k) \quad (3.44)$$

$$\beta_i = \beta_0 \exp[-\gamma \|\mathbf{y}^s - \mathbf{x}^i(k)\|_2^2] \quad (3.45)$$

$$\mathbf{x}^i(k) := \mathbf{x}^i(k) + \alpha \mathbf{s}^i(k) \quad (3.46)$$

ここで、 $\mathbf{s}^i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ は多変量標準正規乱数ベクトル、あるいは $\mathbf{s}^i(k) \sim \mathcal{U}_{\mathbb{R}}(-0.5 \cdot \mathbf{1}, 0.5 \cdot \mathbf{1})$ は多変量標準一様乱数ベクトル、 $\alpha, \beta_0, \gamma \geq 0$ はパラメータである。

式 (3.44) の第3項において、 α は乱数ベクトルの係数パラメータであり、振動の大きさを調整することが可能である。乱数ベクトルは多様化を生み出すとも考えられるが、探索終盤では集中化を妨げることになり、望ましくない。そこで、提案者の Xin-She Yang ら

⁵ 式 (3.43) はルーレット選択の適合度の式だが、光強度 $I^i(k)$ はその大小関係でしか使用されないため、ランキング選択の適合度の式として定義することと等価である。

Algorithm 9 Firefly Algorithm

```

1: procedure FIREFLY ALGORITHM      10:           if  $I_i < I_s$  then
   Step 0: パラメータ設定          11:             式 (3.44)、式 (3.45) を用
2:     パラメータ  $m$ 、 $k_{\max}$ 、 $\alpha$ 、 $\beta_0$ 、 $\gamma$  を設    いて、 $\mathbf{x}^i(k)$  を更新する。
   定する。                               12:           end if
   Step 1: 初期化                  13:           end for
3:     探索点の位置  $\mathbf{x}^i(k) \in S_0$  をランダム 14:           end for
   に与える。                               15:             式 (3.46) を用いて、 $\mathbf{x}^m(k)$  を更新する。
4:      $k = 1$  とする。                   16:             各探索点を  $\mathbf{x}^i(k+1) := \mathbf{x}^i(k)$  と更新
   Step 2: 探索点のソーティング          する。
5:     式 (3.43) を用いて、各探索点の光強度      17:           if  $k \geq k_{\max}$  then
    $I_i$  を計算する。                     18:             計算を終了する。
6:      $I_i$  の非減少順に  $\mathbf{x}^i(k)$  をソートする。 19:           else
7:      $\mathbf{y}^i = \mathbf{x}^i(k)$  とする。               20:              $k := k + 1$  とし、Step 2 へ戻る。
   Step 3: 位置の更新                  21:           end if
8:     for  $i = 1, 2, \dots, m-1$  do          22:           end procedure
9:         for  $s = i+1, 2, \dots, m$  do

```

は、乱数ベクトルの係数パラメータ α を探索過程で減少することが望ましいと述べている [71, 180]。[71, 180]では、式 (3.47) の α 調整則が提案されている。

$$\alpha(k) = \alpha_0 \delta^k \quad (3.47)$$

ただし、 $\delta \in [0, 1]$ は指數の底パラメータ、 $\alpha_0 > 0$ は初期値パラメータである。

Algorithm 3.9.1 に FA の疑似コードを示す。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。

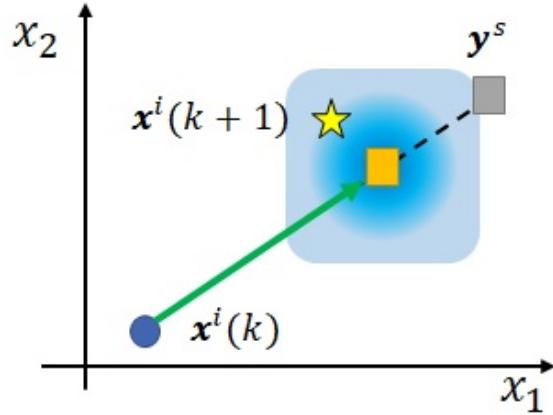


図 3.9 : Firefly Algorithm の近傍の生成（多変量標準正規乱数ベクトルの場合）

3.9.2 Firefly Algorithm の探索構造の解析

定義 3.1 に従って、FA の更新則を解析する。FA の全体の更新則 U (式 (3.4)) は、式 (3.44) (最も良い探索点の場合は式 (3.46)) である。このため、FA 全体の更新則 U において、 $\mathcal{R}(k) = \{\mathbf{s}^i(k)\}$, $\mathcal{C}(k) = \{\alpha, \beta_0, \gamma\}$, $\{\mathbf{x}^i(k), \mathbf{y}^s\} \subseteq \chi(k)$ となる。よって、探索点間の差分ベクトルと乱数ベクトルの線形結合によって近傍解を生成し、探索点を絶対移動させる、という仕組みであることがわかる。

図 3.9 に、FA の近傍生成について示す。光強度 $I^i(k)$ の式 (3.43) は、GA のルーレット選択における適合度 fit_i の式 (3.13) と同様であるため、目的関数値の大きさによって光強度が割り当てられる。しかしながら、探索過程で光強度が使用されるのは、 $I^i(k) < I^s(k)$ という大小関係であるため、 $f(\mathbf{x}^s(k)) < f(\mathbf{x}^i(k))$ という大小関係と等価である。式 (3.44) の探索点の保存位置 \mathbf{y}^s は $I^i(k) < I^s(k)$ を満たすため、探索点 $\mathbf{x}^i(k)$ よりも良い解である。このため、 $\mathbf{y}^s - \mathbf{x}^i(k)$ は探索点群の中の優れた解に向かう差分ベクトルであり、これが集中化の能力を生み出していると考えられる。一方、乱数ベクトル $\mathbf{s}^i(k)$ による摂動を与えていていることで、集中化の能力を抑制する多様化の能力を生み出していると考えられる。

上記の特徴は、p-best, g-best 方向への差分ベクトルと慣性項による摂動の和によって探索点が移動する PSO と類似しているが、FA では、差分ベクトルの係数 β_i が特徴的である。式 (3.45) は正規分布の確率密度関数 (式 (A.15)) であり、 $f_{\mathcal{N}}(x; \mu, \sigma^2)$ と照らし合わ

せると、 $x = \|\mathbf{y}^s - \mathbf{x}^i(k)\|_2, \mu = 0, \sigma^2 = 1/(2\gamma)$ となる⁶。つまり、 γ は分布の拡がりを調整するパラメータ、 β_0 は分布の高さを調整するパラメータであり、差分ベクトルの L^2 ノルムを変数とする正規分布の高さが β_i の値となる。よって、差分ベクトルの L^2 ノルムが大きければ、 β_i は小さくなり、 L^2 ノルムが小さければ、 β_i は大きく、 β_0 に近くなる。この影響から、式(3.44)において、差分ベクトルの L^2 ノルムが極端に大きいときは、差分ベクトルがある程度大きくても、係数 β_i が指数的に小さいことから、第2項はほぼ $\mathbf{0}$ となり、第3項の乱数ベクトルのみにほぼ依存する。つまり、近傍解が生成されやすい範囲は、差分ベクトルの L^2 ノルムが係数 β_i の減少効果を受けにくい程度の範囲に限定されるため、良い探索点の位置 \mathbf{y}^s が探索点 $\mathbf{x}^i(k)$ から遠ければ、探索点はその位置を参照しても \mathbf{y}^s に近づかず、乱数ベクトルの摂動のみによる移動となる。この効果により、比較的距離が近い範囲にある探索点群だけで、その範囲内の良い解付近に集中化することになり、各領域で探索点群が複数の群に分かれることが知られている[181]。この性質を活用したFAの改良研究も行われている[181]。よって、 β_i は、全体の集中化能力を抑制し、比較的距離が近い探索点同士で限定した集中化する能力を与えていていると考えられる。

さらに、 α は式(3.44)の第3項において乱数ベクトルの係数パラメータであり、摂動の大きさを調整することが可能である。しかしながら、探索終盤で探索点群が集中する、あるいは他の探索点が極端に遠いとき、第2項はほぼ $\mathbf{0}$ となり、第3項の乱数ベクトルのみで移動する状態となることに加えて、改善移動ではなく、絶対移動であることから、改悪も許容する一定の摂動が与えられることになる⁷。この挙動は多様化を生み出すとも考えられるが、探索終盤では集中化を妨げることになり、望ましくない。そこで、式(3.47)の α 調整則は、探索が経過するにつれて、 $\alpha(k)$ をスケジューリング的に減少させる調整則である。これは、探索序盤では乱数ベクトルの影響を強くし、探索過程で徐々にその影響を弱めていくことから、多様化・集中化を意識している。以上の点が、FAにおいて多様化・集中化の能力を生み出していると考えられる。

⁶ 確率密度関数は全ての範囲を積分をすると1となるように係数で正規化されているため、厳密には式(3.45)は正規分布と類似する性質を有する分布である。

⁷ PSOにおける摂動の役割を担う慣性項は、パラメータを安定領域に設定すれば、探索過程で徐々に小さくなることから、パラメータの動的な調整がなくても多様化・集中化を実現できる。

3.10 Cuckoo Search の解析

3.10.1 Cuckoo Search

Cuckoo Search (CS) は、2009年に Xin-She Yang と Suash Deb により開発された、多点探索型メタヒューリスティクスである[55, 73, 74]。CS は鳥のカッコウの托卵行動から着想を得ており、CS のアルゴリズムは、Lévy Flight と排斥から構成されており、各操作は Global Search と Local Search に対応づけられる⁸。CS の近似モデルは、Markov 連鎖モデルによって、大域的収束性が示されている[182]。

Lévy Flight は、ランダムウォークの一種である。Lévy Flight では、式 (A.17) 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[55, 73]。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 (A.15) の正規分布に従う乱数によるランダムウォーク (ES の式 (3.8) や実数値 SA の式 (3.22)) 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[55, 73]が、その詳細は確かではない。

Global Search における近傍生成は式 (3.48)，探索点の移動は式 (3.49) で表される。

$$\hat{\mathbf{x}}^i = \mathbf{x}^i + \alpha \mathbf{s}^i \quad (3.48)$$

$$\mathbf{x}^i := \begin{cases} \hat{\mathbf{x}}^i, & f(\hat{\mathbf{x}}^i) < f(\mathbf{x}^i) \\ \mathbf{x}^i, & \text{otherwise} \end{cases} \quad (3.49)$$

ただし、 $\alpha > 0$ はスケールパラメータ、 \mathbf{s}^i は後述する乱数ベクトルである。一方、Local Search では $\theta \leq P_a$ を満たす探索点のみ行う。ただし、 $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ 、 $P_a \in [0, 1]$ は排斥確

⁸ 原著論文では、Global Search しか式が提示されておらず、Local Search の式が不明だったが、それ以降に Xin-She Yang らは Local Search の式を提示している。

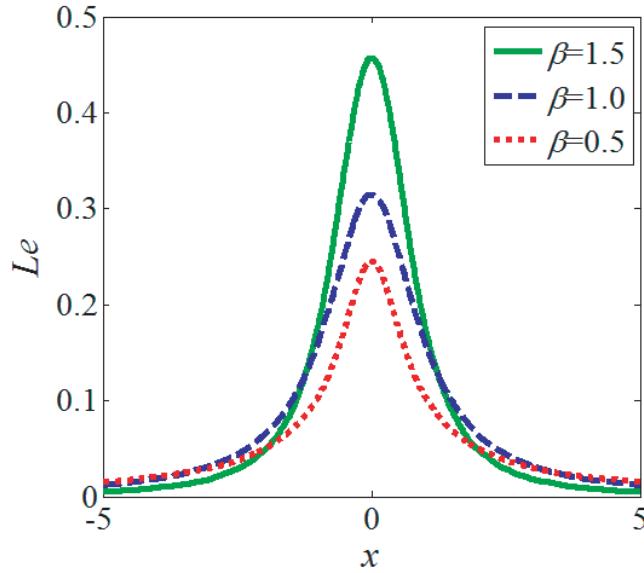


図 3.10：近似 Lévy 分布

率と呼ばれるパラメータである。Local Search における近傍生成は式 (3.50) で表され、探索点の移動は Global Search と同様に式 (3.49) で行われる。

$$\hat{\mathbf{x}}^i = \mathbf{x}^i + r(\mathbf{x}^{r_1} - \mathbf{x}^{r_2}) \quad (3.50)$$

ただし、 $r_1, r_2 \sim \mathcal{U}_{\mathbb{Z}}(1, m)$ ($r_1 \neq r_2$)、 $r \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ である。

乱数ベクトル \mathbf{s}^i は、式 (3.51) を使用する⁹。

$$\mathbf{s}^i = \mathbf{L} \quad (3.51)$$

ただし、 \mathbf{L} の各要素 L_n は、近似 Lévy 分布（安定分布）に従う乱数である。Lévy 分布は通常正の値の変数のみをとるが、最適化向けに正負の符号もまとめて乱数として発生させる必要がある。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) によって生成する [184, 185]。

⁹ 原著論文には記載されていないが、提案者の Xin-She Yang が公開している疑似コードでは、 $\mathbf{s}^i = \text{diag}[v_1 L_1, v_2 L_2, \dots, v_N L_N](\mathbf{x}^c - \mathbf{x}^i)$ とすることで、スケーリングの対策を施している [183]。ただし、 \mathbf{x}^c は c-best、 $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ である。しかし、これは差分ベクトルに乱数を乗じることで、方向に対して Lévy Flight でなくなるため、本稿では従来の式 (3.51) を使用する。

Algorithm 10 Cuckoo Search

```

1: procedure CUCKOO SEARCH           8:    $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ 
   Step 0: パラメータ設定         9:   if  $P_a \geq \theta$  then
2:   パラメータ  $m$ 、 $T_{\max}$ 、 $\alpha$ 、 $\beta$ 、 $P_a$  を設く 10:    $\mathbf{x}^i(k)$  を式 (3.49)、式 (3.50) を
   定する。                           用いて更新する。
   Step 1: 初期化               11:   end if
3:    $\mathbf{x}^i(k)$  を  $S_0$  にランダムに配置する。 12:   end for
   Step 2: Global Search          13:   if  $T \geq T_{\max}$  then
4:   for  $i = 1, 2, \dots, m$  do       14:   計算を終了する。
5:      $\mathbf{x}^i(k)$  を式 (3.48)、式 (3.49) を用        15:   else
   いて更新する。                     16:     Step 2 へ戻る。
6:   end for                      17:   end if
   Step 3: Local Search          18: end procedure
7:   for  $i = 1, 2, \dots, m$  do

```

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 3.10 に $\beta = 1.5, 1.0, 0.5$ のときの近似 Lévy 分布を示す。MA による乱数生成方法については付録 B を参照されたい。[185]では、CS における MA 以外の乱数発生方法について検討している。

Algorithm 3.10.1 に CS の疑似コードを示す。アルゴリズムの終了条件を $T \geq T_{\max}$ とする。

3.10.2 Cuckoo Search の探索構造の解析

定義 3.1 に従って、CS の更新則を解析する。Global Search の近傍生成則 U_{nei} は式 (3.48) が該当し、探索点の移動則 U_{mov} は改善移動の式 (3.49) が該当する。このため、Global Search

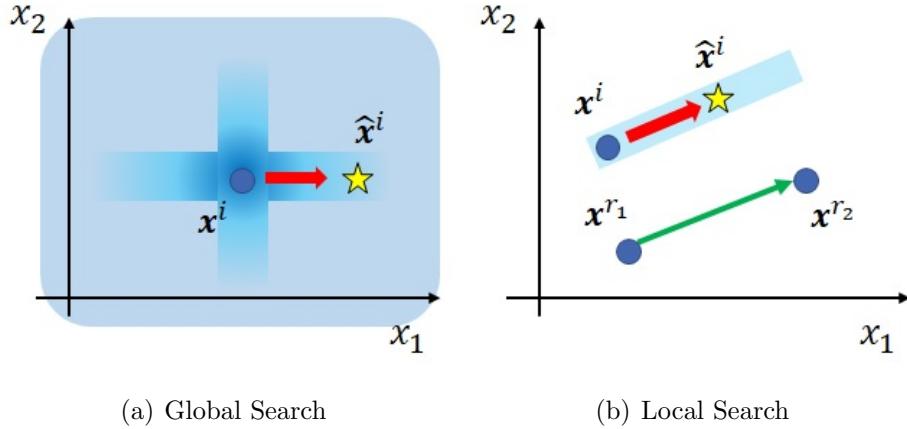


図 3.11 : Cuckoo Search の近傍の生成

全体の更新則 U において, $\mathcal{R}(k) = \{\mathbf{s}^i\}$, $\mathcal{C}(k) = \{\alpha, \beta\}$, $\{\mathbf{x}^i\} \subseteq \chi(k)$ となる。よって, 探索点から摂動を与えることで近傍解を生成し, 改善する場合に, 探索点を移動させる, という仕組みであることがわかる。Local Search の近傍生成則 U_{nei} は式 (3.50) が該当し, 探索点の移動則 U_{mov} は改善移動の式 (3.49) が該当する。このため, Local Search 全体の更新則 U において, $\mathcal{R}(k) = \{r, r_1, r_2\}$, $\mathcal{C}(k) = \{P_a\}$, $\{\mathbf{x}^i, \mathbf{x}^{r_1}, \mathbf{x}^{r_2}\} \subseteq \chi(k)$ となる。よって, 探索点間の差分ベクトルによって近傍解を生成し, 改善する場合に, 探索点を移動させる, という仕組みであることがわかる。

図 3.11 に, CS の Global Search と Local Search の近傍生成について示す。Global Search では, 探索点 \mathbf{x}^i に乱数ベクトルによる摂動を加えることで, 近傍を生成している。さらに, 乱数ベクトルは Lévy 乱数であるため, 近傍は探索点 \mathbf{x}^i から近い距離に生成される確率は高いが, 遠い距離でも生成される一定の確率がある。よって, Global Search は基本的に集中化の役割だが, 弱い多様化も実現している。一方, Local Search では, 排斥確率 P_a を満たす探索点について, 探索点間の差分ベクトルに乱数 $r \in [0, 1]$ をかけた摂動を探索点 \mathbf{x}^i を加えることで, 近傍を生成している。探索点間の差分ベクトルは探索点群のスケールや分布の情報を含む摂動であると同時に, その摂動を乱数 $r \in [0, 1]$ によって縮小させるため, Local Search は集中化の役割である。そして, Local Search では, 排斥確率 P_a の大きさを変えることで, その集中化を行う確率を調整できる。また, Global Search と Local Search の両方で, 改善移動であるため, 目的関数値が改善した場合に限り, 探索点は近傍

解へ移動する。つまり、CS では探索点群が降下条件を満たす場合のみ移動を行う。以上の点が、CS において多様化・集中化の能力を生み出していると考えられる。

3.11 メタヒューリスティクスの分類

3.11.1 多様化・集中化の定義

メタヒューリスティクスの一般的な解説は、主として各々のアルゴリズムが基礎としているアナロジー現象に着目した分類に基づいている。しかしながら、どのような現象を基礎としたかは別として、最適化アルゴリズムとして見た場合の分類や評価も重要である。ところが、メタヒューリスティクスの分類指針は、現状として十分に確立されてはいない。そこで、3.10 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結果、第 2 章で述べたように、メタヒューリスティクスは、近接最適性原理（POP）に基づく多様化・集中化の探索戦略を実現することで、直接探索法として効率的な探索を行っていることがわかった。一方、多様化・集中化は抽象的な概念であると同時に、多様化・集中化の実現方法は、アルゴリズムの種類によっても異なり、具体的な解析に活かしにくい。本論文の目的である、統一的な視点でメタヒューリスティクスに対する解析・設計を行うためには、「多様化・集中化に対する解釈・状態」と「多様化・集中化を生み出す操作」の対応関係を整備する必要がある。よって、本節では、3.10 節までの解析に基づき、上記の対応関係を整備した後、それが各アルゴリズムの更新則ではどのような操作が埋め込まれているのかを明確にし、分類する。

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[53, 55]。図 3.12 に、2 次元空間の 2^N minima 関数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。

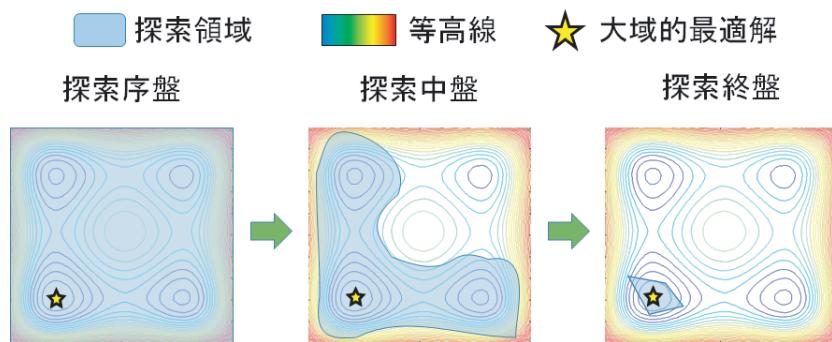


図 3.12：メタヒューリスティクスの探索戦略のイメージ

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	探索領域 / 探索点分布	① 特定領域 / 方向への指向性	② 摂動 / 分布
多様化	長期	拡大	抑制	拡張
集中化	短期	縮小	推進	限定

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、長期的な解の改善、短期的な解の改善とする[186]。この定義では、多様化・集中化の状態を「探索領域・探索点分布の広さ」に対応させる。探索領域は近傍（探索点の次回移動する候補解）を表し、探索点分布は探索点群の現在位置を表す。多様化により、探索領域・探索点分布が広がり、集中化により、探索領域・探索点分布が狭まる。探索構造では、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・方向への指向性の抑制」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・方向への指向性の促進」や「摂動・分布の縮小」を行う操作とする。特に、探索構造では、「特定の領域」として優れた解を設定することや、探索点群が優れた解付近に分布させるような集中化を実現する操作は、POPを効果的に活用している。

3.11.2 更新則の分類

(a) 演算形式の分類

3.10 節まで行った解析を基に、メタヒューリスティクスの更新則を考察し、分類する。3.1 節で述べたように、メタヒューリスティクスの更新則は近傍生成則と移動則の合成によって構成されているが、移動則は、無条件で探索点が近傍解に移動する絶対移動、目的関数値の比較によって探索点が近傍解に移動する改善移動、改善移動か受理確率を満たす場合に改悪を許容する確率的移動の3種類に分類されるため、メタヒューリスティクスの本質的な特徴が含まれていることは少ない¹⁰。よって、本節では、特に近傍生成則に着目し、多様化・集中化との関連性を中心に、その更新則を分類する。

近傍生成則は、探索点やその探索履歴に含まれる解などの位置ベクトルと、独立した確率分布に従う乱数ベクトルの線形演算で基本的に表現される。つまり、位置ベクトル、乱数ベクトル、演算の集合 $\{+, -, \times, /\}$ および単純な非線型関数の集合 $\{\sqrt{}, \exp, \log\}$ ¹¹ などの組合せによって表現される。さらに、使用されるベクトルの種類と乱数の使用箇所の観点から、更新則の演算形式を、(1) 基本ベクトルと差分ベクトルの線形結合、(2) 解同士の要素の組み替え、(3) ランダムウォーク、の3種類に分類できると考えられる。

(1) 基本ベクトルと差分ベクトルの線形結合

(1) は、式(3.52)のように、基本ベクトル \mathbf{a} と解の差分ベクトル $\mathbf{b}_j - \mathbf{c}_j$ の線形結合で表されるケースを示している。

$$\hat{\mathbf{x}} = \mathbf{a} + \sum_j \mathbf{A}_j (\mathbf{b}_j - \mathbf{c}_j) \quad (3.52)$$

ただし、 $\mathbf{a}, \mathbf{b}_j, \mathbf{c}_j \in \mathbb{R}^N, \mathbf{A}_j \in \mathbb{R}^{N \times N}$ である。 \mathbf{a} は基本ベクトルで、基本的な位置を表し、更新する探索点自身の位置ベクトルや解集団の平均ベクトルなどが使用される。 $\mathbf{b}_j - \mathbf{c}_j$ は差分ベクトルで、基本ベクトルに与える摂動を表し、探索点と優れた解の差分ベクトル、探索点間の差分ベクトルなどが使用される。差分ベクトルの効

¹⁰ ただし、GA では淘汰も重要な操作であり、淘汰における世代交代モデルは複雑な操作を含むため、この3種類に単純に分類されないが、基本的には目的関数値に着目している。

¹¹ 近傍生成則に非線型関数を含む場合はあるが、それはスカラ（ベクトルの係数など）を非線形変換するためであるため、ベクトルの線形演算であることには相違ない。

果については後述する。 \mathbf{A}_j は差分ベクトルの係数で、正方行列とスカラの場合がある¹²。係数は、乱数やパラメータで構成されることが多いが、平均をとるための正規化係数の場合もある。この場合の近傍は、基本的に、基本ベクトル \mathbf{a} と差分ベクトル $\mathbf{b}_j - \mathbf{c}_j$ を基底ベクトルとしてみなし、その基底ベクトルで張る空間である。ただし、具体的な形状は係数が行列かスカラなのに依存しており、この性質は回転不变性やスケール不变性と関連が深い。このタイプに分類されるのは、実数値 GA の交叉、PSO、DE の突然変異、ABC の働き蜂・傍観蜂、FA、CS の Local Search などが挙げられ、メタヒューリスティクスの最も基本的なタイプであるといえる。

(2) 解同士の組み合わせ

(2) は、式(3.53)、式(3.54)のように、解 \mathbf{a} と解 \mathbf{b} の線形結合で表されるケースを示している。

$$\hat{\mathbf{x}} = \mathbf{M}_1 \mathbf{a} + \mathbf{M}_2 \mathbf{b} \quad (3.53)$$

$$\mathbf{M}_1 + \mathbf{M}_2 = \mathbf{I} \quad (3.54)$$

ただし、 $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N, \mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{N \times N}$ である。 \mathbf{a}, \mathbf{b} は組み替え対象のベクトルで、探索点の位置ベクトルや、(1) で生成されたベクトルなどが使用される。 $\mathbf{M}_1, \mathbf{M}_2$ は式(3.54)で拘束される対角行列であり、組み替えを決定する。その対角要素が $\{0, 1\}$ であるとき、1 となった要素が近傍解の要素として採用される仕組みである。線型代数の視点では、(1) を線形結合としてみなすなら、(2) は凸結合¹³の頂点に生成される解釈に近い。この場合の近傍は、 \mathbf{a}, \mathbf{b} を結ぶ線分を対角線とし、それらを頂点とする超直方体を生成し、その超直方体の頂点である。ただし、具体的な頂点はアルゴリズムに依存している。一方、その対角要素が $[0, 1]$ であるとき、近傍は凸結合の範囲、つまり、上記の超直方体の内部全体である。これらの近傍解は必ず座標系に平行な超直方体上に生成されるため、回転不变性と関連が深い。このタイプに分

¹² 係数が正方行列の場合、厳密には式(3.52)を線形結合とは呼ばないが、メタヒューリスティクスの場合、係数は対角行列や回転行列である、つまり、ベクトルの座標変換（スケール変換や回転変換）を意図しているため、本質的には近傍生成則はそれらのベクトル同士の和として表現される。

¹³ 凸結合は、 $\sum_j \alpha_j \mathbf{x}_j, \alpha_j \geq 0, \sum_j \alpha_j = 1$ であり、 \mathbf{x}_j を頂点とする凸集合に生成される。

類されるのは、実数値GAのビットストリング交叉[59]、DEの一様交叉、ABCの働き蜂・傍観蜂などが挙げられる。

(3) ランダムウォーク

(3)は、式(3.55)のように、基本ベクトル \mathbf{a} と乱数ベクトル \mathbf{s} の線形結合で表されるケースを示している。

$$\hat{\mathbf{x}} = \mathbf{a} + \mathbf{A}\mathbf{s} \quad (3.55)$$

ただし、 $\mathbf{a}, \mathbf{s} \in \mathbb{R}^N, \mathbf{A} \in \mathbb{R}^{N \times N}$ である。 \mathbf{a} は基本ベクトルで、探索点の位置ベクトルや、解集団の平均ベクトル、(1)で生成される解などが使用される。 \mathbf{s} は乱数ベクトルで、 $\mathbf{s} \sim \mathcal{P}$ のようにある多変量確率分布 \mathcal{P} に従う。確率分布 \mathcal{P} は付録Aに挙げるような、正規分布、一様分布、Lévy分布が挙げられる。特に \mathbf{a} を探索点にした場合、探索点 \mathbf{a} が確率分布 \mathcal{P} に従ってランダムウォークするタイプである。係数 \mathbf{A} は、スカラのパラメータや乱数で構成されることが多いが、正方行列である場合もある¹⁴。この場合の近傍は、 \mathbf{a} を中心として、 \mathbf{s} が与える摂動範囲である。ただし、具体的な近傍は、 \mathbf{s} が従う確率分布 \mathcal{P} の性質に依存する。この性質が、スケール不变性や回転不变性と関連が深い。このタイプに分類されるのは、ES、実数値SA、FA、CSのGlobal Searchなどが挙げられる。

なお、DEとABCは(1)と(2)のハイブリッド、FAは(1)と(3)のハイブリッドである。このように、近傍生成則は3種類に分類されるが、多様化・集中化を生み出す操作はそれに含まれるベクトルや乱数によるものである。そこで以降では、この近傍則で使用されるベクトルや乱数を分類し、多様化・集中化との関連性を分析する。

(b) ベクトルの種類

ここでは、近傍生成則で使用されるベクトルの種類を分類し、多様化・集中化との関連性を分析する。(a)の3種類の近傍生成則で使用されるベクトルは、(1)探索点の位置ベクトル、(2)差分ベクトル、(3)乱数ベクトル、の3種類があることがわかる。

¹⁴ (1)と同様、正方行列の場合、式(3.55)は厳密には線形結合とは呼ばないが、この正方行列は対角行列や回転行列などであり、乱数ベクトルを変換することを意図している。

(1) 探索点の位置ベクトル

探索点の位置ベクトルは最も基本的なベクトルである。探索履歴の解もこれに分類されるが、式(3.52)の第1項の基本ベクトルのように単独で使用されるのは、探索点の現在の位置ベクトルか、探索点集団の平均ベクトルであることがほとんどである。これは、直接探索法において、探索点群の現在の情報を基準として近傍生成するためであり、多様化・集中化の意図は含まれていないと考えられる。

(2) 差分ベクトル

式(3.52)の第2項のように、メタヒューリスティクスの近傍生成則では、差分ベクトルが最もよく使用される。差分ベクトルの始点は探索点の位置ベクトルやランダムに選択した他の探索点の位置ベクトルが用いられ、終点は特殊な解・探索点を設定する。PSOやFAでは、探索点から p-best, g-best, 有望な探索点に向かう差分ベクトルを用いている。また、DEの突然変異やABCの働き蜂・傍観蜂、CSのLocal Searchでは、探索点間の差分ベクトルを用いている。差分ベクトルは、探索点分布のスケールや分布の指向性などが反映された情報であり、目的関数の解析的情報を用いない直接探索法において非常に重要である。そして、差分ベクトルの係数には、通常 $r \in [0, 1]$ のような小さな乱数やパラメータが用いられるため、基本的に探索点分布を縮小する操作となり、探索過程で特定の領域へ探索点群・近傍が徐々に狭まる効果を生み出す。このように、差分ベクトルの使用は、3.11.1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。

さらに、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 や有望な探索点などの良い解に向かう差分ベクトルの活用は、特定の有望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。ただし、探索過程で同じ解に対する差分ベクトルを使用する場合、その領域に対する指向性が常に促進されるため、短期的に探索点分布が縮小する傾向がある。逆に、ランダムに選択した探索点間の差分ベクトルを使用するだけでは、移動則において改善移動でなければ、探索点群は下降方向に移動しない。このように、直接探索法では目的関数の解析的情報を用いずに、探索過程で適切な多様化・集中化（最適化能力）を実現するために

は、POP のための良い解の埋め込みや移動則における効果の併用による集中化効果の創出、あるいは、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による多様化効果の創出、が重要である。

(3) 乱数ベクトル s

式(3.55)の第2項のように、メタヒューリスティクスの近傍生成則では、乱数ベクトルも使用される。乱数ベクトルは、式(3.55)のように、第1項を中心として、摂動として確実に与える効果を有する。ES, 実数値 SA, FA, CS の Global Search では、正規分布や Lévy 分布に従う乱数ベクトルを用いている。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有する近傍を生成するため、摂動・分布を拡大する操作（多様化）となる。さらに、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。このため、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。この性質から、単独で乱数ベクトルを活用するだけでは、多様化・集中化の探索状態を変動させることは困難であることが考えられる。そこで、(1+1)-ES や FA ではスケールパラメータを動的に調整するパラメータ調整則や、実数値 SA では受理確率を探索過程で減少させていくパラメータ調整則が付加されていることが多い。さらに、乱数ベクトルはそれが従う確率分布の性質に依存する。例えば、ES, 実数値 SA, FA における乱数ベクトルは多変量標準正規分布に従うが、正規分布の確率密度関数は中心から離れると指数的に減少するため、乱数ベクトルの生成範囲はある一定範囲内に収まり、その確率分布の偏りは等方的である。一方、CS の Global Search における乱数ベクトルは Lévy 分布（安定分布）に従うが、安定分布の確率密度関数は、中心から離れても指数的に減少せず、一定値が残るヘヴィーテイル型であるため、乱数ベクトルの生成範囲はかなり広く、その確率分布の偏りは異方的である。さらに、 (μ, λ) -ES や CMA-ES のような多点型 ES では、標準正規分布（等高線が真円）に従う乱数ベクトルに対角行列と回転行列を作用させることで、共分散行列を持つ多変量正規分布（等高線が座標系に対して斜向している楕円）に従う乱数ベクトルに変換している。このように、乱数ベクトルは確率分布の性質に依存するため、ステップサイズの調整や変換により、摂動の傾向を変動させることで、3.11 節で定義した多様化・集中化に貢献していることが考えられる。

(c) 乱数の使用箇所

数理最適化とは異なり、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では疑似乱数を使用することで多様性を生み出す。これは、直接探索法においてPOPに基づく集中化の効果を生み出すためには、良い解に向かう差分ベクトルの活用や改善移動などが有効であったが、これだけでは過度な集中化になってしまうためである。(a) の3種類の近傍生成則における乱数の使用箇所は、(1) 差分ベクトルの係数、(2) 乱数ベクトル、(3) 探索点や要素の選択、(4) 操作実施の判断、の4種類があることがわかる。

(1) 差分ベクトルの係数

実数値GA, PSO, ABC, CSのLocal Searchでは、差分ベクトルの係数に乱数を用いている。この乱数は多様化・集中化の実現に対して様々な効果を生み出す。差分ベクトルの係数が定数である場合、差分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ず、特定の領域・方向に対する指向性を過度に促進する操作となってしまう。これに対して、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、差分ベクトルの係数として使用する乱数 r は、実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従うものがほとんどである。これは、探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く、探索点分布を縮小する操作となる。これに対して、ABCでは、 $\mathcal{U}_{\mathbb{R}}(-1, 1)$ に従う乱数を使用しているため、乱数区間サイズが1以上であり、負の値もとることから、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、3.11.1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。

(2) 亂数ベクトル

乱数ベクトルは、各要素が確率分布に独立に従った乱数で構成されている。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有する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、(b)で述べたように、近傍の指向性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、乱数ベクトルが従う確率分布の種類により、近傍内で近傍解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・集中化の効果を生み出すことが考えられる。

(3) 探索点や要素の選択

探索点の選択、あるいは要素の選択において、乱数が使用されることがある。実数値 GA, DE の突然変異、ABC の働き蜂・傍観蜂、CS の Local Search では、差分ベクトルで使用する探索点はランダムに選択される。これは各探索点を一様に扱うこと、特定の領域に対する指向性を抑制している操作となる。また、実数値 GA や ABC の傍観蜂では、探索点の目的関数値による選択確率が重みづけされるルーレット選択が使用される。他にも、実数値 GA の世代交代モデルでは、単に優れた解のみを次世代の探索点とするわけではなく、ランダムに選択された探索点も次世代に移すことで、多様性を確保している。その他に、DE の交叉や ABC の働き蜂・傍観蜂では、組み合わせる解の要素の選択において、乱数を使用する。解同士の組合せで乱数を使用することは、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、解のうち单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 の交叉では、解の全ての要素を一様に組み合わせるため、特定の領域に対する指向性を抑制させている。このように、探索点や要素の選択において乱数を使用することは、近傍に多様性を与

えている。

(4) 操作実行の判断

ある操作を実行するかどうかの判断において、乱数が使用されることがある。例えば、CS の Local Search は、排斥確率 P_a を満たした場合のみ行う、実数値 SA の確率的移動は、受理確率を満たす場合は悪化を許容して探索点を移動させる、などが挙げられる。両方とも、多様化・集中化のどちらかの操作を常に実行しており、ある確率を満たす場合に他の操作を実行する仕組みになっている。つまり、探索過程で多様化・集中化を均等に実行しているわけではなく、確率的に操作を実行することで、探索の多様性を与える。

(d) 更新則の分類

以上のように、近傍生成則は演算形式が3種類に分類されると同時に、含まれる操作により更新則全体の多様化・集中化の能力が決定されることがわかった。これらは最適化アルゴリズムの振る舞いを決定づける重要な項目であることから、各種のメタヒューリスティクスにおけるこれらの違いが、各メタヒューリスティクスの最適化アルゴリズムとしての本質的相違を明らかにしている。

表 3.2 は、これまでの解析に基づき、各種のメタヒューリスティクスの更新則において、多様化・集中化を生み出す操作、そして多様化から集中化に移行する探索戦略を実現している方法を整理したものを見ている。下記に、各説明を示す。

(1+1)-ES 多変量正規乱数ベクトルによる摂動が多様化の操作で、改善移動が集中化の操作となる。探索過程で改善頻度に基づき、摂動スケールパラメータを増減させることで、摂動の拡大・限定ができる、多様化・集中化の調整が可能である。ただし、「探索序盤で多様化、探索終盤で集中化」という探索戦略は埋め込まれていない。

実数値 GA 世代交代モデルによる「特定領域に対する指向性」の抑制や、統計量の遺伝による「探索点分布の維持」が多様化の操作で、探索点分布内への近傍生成や、世代交代モデルによる「特定領域に対する指向性」の促進が集中化の操作となる。さらに、探索過程で探索点分布を徐々に縮小させていく点が探索戦略を実現している。

- 実数値 SA 多変量正規乱数ベクトルによる「摂動」や確率的な悪化の許容による「探索点分布の拡大」が多様化の操作で、基本的な改善移動が集中化の操作となる。さらに、温度パラメータの冷却スケジュールによって、探索過程で悪化を許容する確率を減少させていくことで、摂動範囲を徐々に縮小させる点が探索戦略を実現している。
- PSO 慣性項による「摂動」や、差分ベクトルの係数の乱数による「特定領域に対する指向性の抑制」が多様化の操作で、g-best と p-best に向かう差分ベクトルによる「特定領域に対する指向性の促進」が集中化の操作となる。さらに、PSO のパラメータを弱い安定領域に設定することで、探索序盤では探索点群が大きく動き、探索過程で徐々に移動量が狭まっていく点が探索戦略を実現している。
- DE 解の組み合わせる要素のランダム選択や探索点のランダム選択が多様化の操作で、探索点間の差分ベクトルによる「探索点分布の縮小」や、改善移動による「特定領域に対する指向性の促進」が集中化の操作となる。さらに、探索点間の差分ベクトルによって、探索過程の探索点群のスケールや分布の方向を反映させ、徐々に近傍を狭めていく点が探索戦略を実現している。
- ABC 解の組み合わせる要素のランダム選択や探索点のランダム選択、有望でない探索点の再配置が多様化の操作で、探索点間の差分ベクトルによる「探索点分布の縮小」や、改善移動による「特定領域に対する指向性の促進」、優れた探索点に重みづけした選択が集中化の操作となる。さらに、探索点間の差分ベクトルによって、探索過程の探索点群のスケールや分布の方向を反映させ、徐々に近傍を狭めていく点が探索戦略を実現している。
- FA 多変量正規乱数ベクトルによる「摂動の拡張」や、探索点間の距離に基づく差分ベクトルの係数の減衰による「特定領域への指向性の抑制」が多様性の操作となり、より優れた探索点に向かう差分ベクトルが集中化の操作となる。近傍生成則に乱数ベクトルと差分ベクトルの両方を使うことや、探索過程で摂動スケールパラメータを減少させていく点が探索戦略を実現している。
- CS 安定分布に従う乱数ベクトルによる「摂動の拡張」や、探索点のランダム選択、差分ベクトルの係数の乱数による「特定領域に対する指向性の抑制」が

表3.2：各種メタヒューリスティクスの特徴

アルゴリズム	多様化の操作	集中化の操作	探索戦略の実現方法
(1+1)-ES	乱数ベクトルによる摂動	改善移動	改善頻度に基づく摂動スケールの増減
実数値 GA	統計量の遺伝・世代交代モデル	探索点分布内への近傍生成 ・世代交代モデル	探索過程で分布を縮小
実数値 SA	乱数ベクトルによる摂動 ・確率的な悪化の許容	(基本的には) 改善移動	探索時間に基づく悪化確率の減少
PSO	慣性項・ 差分ベクトルの係数で乱数の使用	最良解と個別最良解に 向かう差分ベクトル	弱い安定状態
DE	解の組合せ方で乱数の使用 ・探索点のランダム選出	探索点間の差分ベクトル・ 改善移動	差分ベクトルによる 探索状態の反映
ABC	解の組合せ方で乱数の使用 ・探索点のランダム選出 有望でない探索点の再配置	探索点間の差分ベクトル・ 改善移動・ 優れた探索点に重みづけした選択	差分ベクトルによる 探索状態の反映
FA	乱数ベクトルによる摂動・ 探索点間の距離による係数の減衰	より優れた探索点に 向かう差分ベクトル	多様化・集中化の同時実現 探索時間に基づく摂動スケールの減少
CS	乱数ベクトルによる摂動・ 探索点のランダム選出・ 差分ベクトルの係数で乱数の使用	探索点間の差分ベクトル ・改善移動	多様化・集中化の同時実現

多様化の操作で、探索点間の差分ベクトルによる「探索点分布の縮小」や、改善移動による「特定領域に対する指向性の促進」が集中化の操作となる。1反復で、摂動の拡張と探索点分布の縮小の両方を行い、どちらも改善移動とする点が、多様化・集中化を同時に実現している。さらに、探索点間の差分ベクトルによって、探索過程の探索点群のスケールや分布の方向を反映させ、徐々に近傍を狭めていく点が探索戦略を実現している。

これらの解析結果に基づくと、メタヒューリスティクスの探索構造は、多点型かつ確率的な直接探索法という特徴を活用しているメカニズムを有していることが明らかとなった。

3.12 まとめ

本章では、既存のメタヒューリстиクスとして、ES、実数値GA、実数値SA、PSO、DE、ABC、FA、CSを概観しながら、その探索構造を解析・分類した。まず、更新則を近傍生成則と探索点の移動則の合成とみなし、各アルゴリズムがどのような情報を使用しているのかを確認した。次に、多様化・集中化の明確な定義を与え、それを実現する操作を分類した。この定義に基づき、主要な近傍生成則の演算形式を分類、使用されているベクトルや乱数を整理し、それぞれの多様化・集中化へ与える定性的な効果を明らかにした。最後に、これらの解析結果に基づき、各メタヒューリстиクスにおける多様化・集中化の操作や、「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方法を整理し、メタヒューリстиクスの探索構造が多点型かつ確率的な直接探索法という特徴を最大限に活かしたメカニズムを有していることを明らかにした。

4

変換不变性に基づくメタヒューリスティクスのロバスト化

4.1 本章について

本章では、メタヒューリスティクスの変換不变性を定義した後、各アルゴリズムの不变性の具備や欠如を明らかにする。さらに、代表的なメタヒューリスティクスである、Particle Swarm Optimization (PSO), Artificial Bee Colony Algorithm (ABC), Cuckoo Search (CS) に着目し、不变性の付加を検討する。具体的には、PSO と ABC は回転不变性を欠如、CS はスケール不变性と回転不变性を欠如している。そして、欠如している不变性を付加することで、回転不变性を有する PSO, 回転不变性を有する ABC, アフィン変換不变性を有する CS を提案する。また、証明や数値実験を通じて、各提案手法が欠如していた不变性が付加されていることと、その変換に対して探索性能のロバスト性が向上することを確認する。最後に、これらの変換不变性を付加するアプローチの課題を提起する。

本章の成果は、下記の点で新規性があると考える。

- PSO の不变性に関する先行研究は数多くあり、回転不变性を有する PSO も提案されているが、本章では、他のアルゴリズムに対する回転不变性の付加のために使用されている「共分散行列に基づく近傍の回転」を PSO に適用する。このアプローチによって、PSO の回転不变性を獲得した研究は著者の知る限りない。
- 回転不变性を有する ABC は、先行研究においても提案されている¹が、ABC の不变

¹ 先行研究の目的が回転不变性の獲得のためではなく、変数間依存性への対処だが、これらは本質的に同質であり、結果として回転不变性を有している。

性に関する研究は著者の知る限りない。

- 本章では、他のアルゴリズムに対する回転不变性の付加のために使用されている「近傍の超球化」を ABC に適用する。このアプローチによって、ABC の回転不变性を獲得した研究は著者の知る限りない。
- CS の不变性に関する先行研究も、アフィン変換不变性を有する CS に関する先行研究は著者の知る限りない。さらに、他のアルゴリズムに対する回転不变性の付加のために使用されている「共分散行列に基づく近傍の回転とスケーリング」を CS に適用する。このアプローチによって、CS のアフィン変換不变性を獲得した研究は著者の知る限りない。

なお、本章は著者の先行研究[210, 215, 217, 223, 224, 225, 227, 228]に基づいている。

4.2 メタヒューリスティクスの変換不变性

4.2.1 変換不变性の定義

最適化アルゴリズムの変換不变性を定義 4.1 として定義する。

定義 4.1 (最適化アルゴリズムの変換不变性) 最適化アルゴリズムの更新則 $U_{f,S}$ がある変換 $T_s : \mathbb{R}^N \rightarrow \mathbb{R}^N$, $T_f : \mathbb{R} \rightarrow \mathbb{R}$ に対して不变性を有するとは, $\forall k \in \mathbb{N}$ に対して式 (4.1) が成立することである。

$$U_{T_f \circ f, S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) = (T_s \circ U_{f,S})(\mathbf{x}^i(k); \chi_S(k)) \quad (4.1)$$

ただし, $\chi_{S'}(k) = \{T_s(\mathbf{x}^i(\kappa)) \mid i = 1, \dots, m; \kappa = 1, \dots, k\}$ である。また, T_s の逆変換 T_s^{-1} の存在を仮定すると, 式 (4.2) は式 (4.1) と等価である。

$$(T_s^{-1} \circ U_{T_f \circ f, S'})(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) = U_{f,S}(\mathbf{x}^i(k); \chi_S(k)) \quad (4.2)$$

解空間の変換 T_s に対する不变性の条件式は, 式 (4.1) あるいは式 (4.2) において目的関数の変換 T_f を恒等写像だと仮定した式であり, 目的関数の変換 T_f に対する不变性の条件式は, 式 (4.1) あるいは式 (4.2) において解空間の変換 T_s を恒等写像だと仮定した式である。図 4.1 に解空間の変換 T_s に対する不变性に関する可換図式, 図 4.2 に目的関数の変換 T_f に対する不变性に関する可換図式を示す²。図 4.1 において, $\mathbf{x}^i(k)$ を基本状態とすると, 式 (4.1) の左辺は $\mathbf{x}^{i''}(k)$ を経由して $\mathbf{x}^{i''}(k+1)$ まで辿るルートで, 右辺は $\mathbf{x}^i(k+1)$ を経由して $\mathbf{x}^{i''}(k+1)$ まで辿るルートを表すため, これらのルートが異なっても同じ位置に辿り着くことが可能であるという写像の可換性が解空間の変換不变性の条件となる。このときの左辺を \mathbf{x}_A と表し, 右辺を \mathbf{x}_B と表すこととする。図 4.2 において, 左上の $\{\mathbf{x}^i(k), f(\mathbf{x}^i(k))\}$ を基本状態とすると, 式 (4.1) の左辺は $\{\mathbf{x}^i(k+1), f'(\mathbf{x}^i(k+1))\}$ を経由して $\{\mathbf{x}^i(k+1), f(\mathbf{x}^i(k+1))\}$ まで辿るルートで, 右辺は直接 $\{\mathbf{x}^i(k+1), f(\mathbf{x}^i(k+1))\}$ まで辿るルートを表すため, これらのルートが異なっても同じ位置に辿り着くことが可能であるという写像の可換性が目的関数の変換不变性の条件となる。

第 2 章において, 非線形最適化では, 変数間依存性, 悪スケール性, 非凸性のような問題のクラスや性質が主要な問題だが, ① 目的関数の解析的情報 (勾配や Hesse 行列など)

² 図 4.1, 図 4.2 は, いわゆる圏論の可換図式である。通常の可換図式は恒等射を記載するが, 各対象において恒等射は定義されていると仮定する。なお, 図 4.1, 図 4.2 では対象が集合であるから, 恒等射は恒等写像を指す。

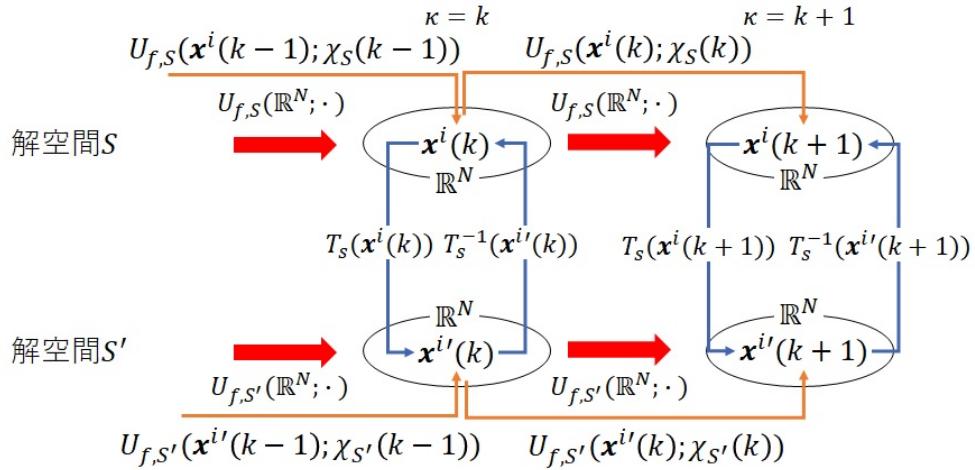


図 4.1：解空間の変換不変性の定義写像図式

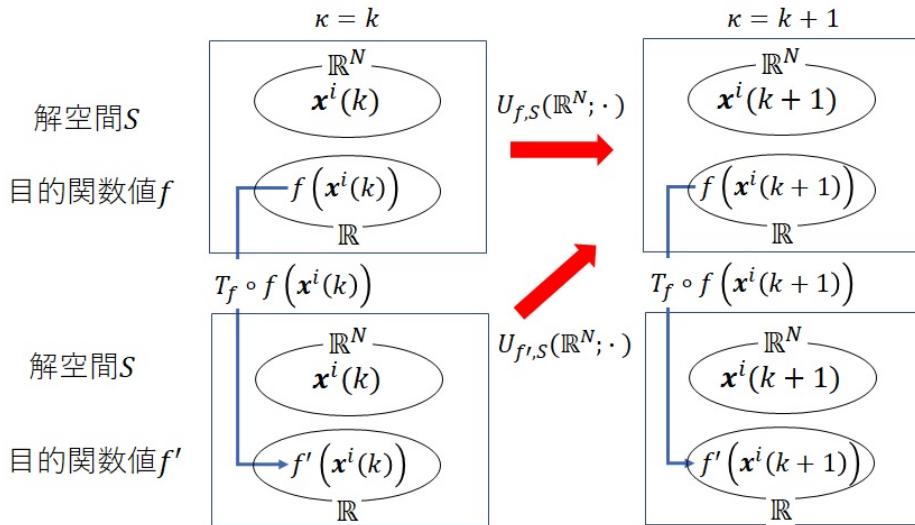


図 4.2：目的関数の変換不変性の定義写像図式

を用いることで対処可能だが、BBO では使用できない、② BBO では事前に目的関数の情報を知ることができないため、どの性質に特化してアルゴリズムを選択すればよいのかが不明であり、適切に選択できない、などの課題を挙げた。この課題を受け、BBO において重要な変換不変性は下記の通りである。

解空間の回転不変性： 解空間の回転変換 $T_s : \mathbf{x} \mapsto \mathbf{Bx}$ に対する不变性（回転不変性）。

ただし、 $\mathbf{B} \in \mathbb{R}^{N \times N}$ は正規直交行列 ($\mathbf{B}^T = \mathbf{B}^{-1}$) である。

解空間の相似変換不変性： 解空間の相似変換 $T_s : \mathbf{x} \mapsto \delta\mathbf{x}$ に対する不変性（相似変換不変性）。ただし， $\delta > 0$ である。

解空間のスケール不变性： 解空間のスケール変換 $T_s : \mathbf{x} \mapsto \mathbf{D}\mathbf{x}$ に対する不変性（スケール不变性）。ただし， $\mathbf{D} \in \mathbb{R}^{N \times N}$ は対角行列である。スケール変換は相似変換を一般化した変換である。

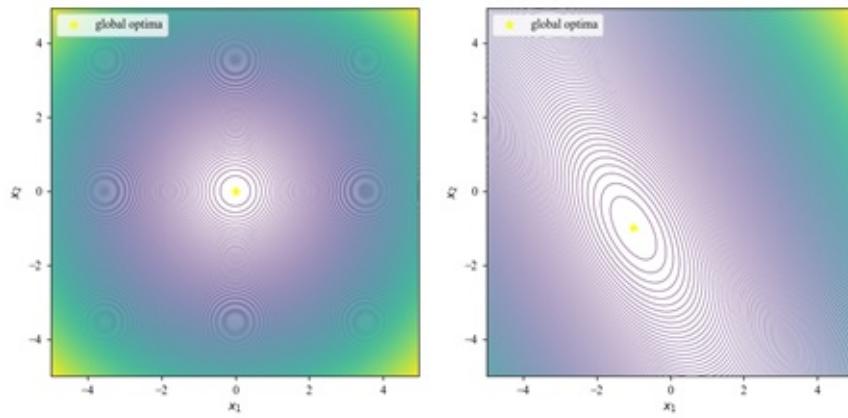
解空間の平行移動不变性： 解空間の平行移動変換 $T_s : \mathbf{x} \mapsto \mathbf{x} - \mathbf{t}$ に対する不変性（平行移動不变性）。ただし， $\mathbf{t} \in \mathbb{R}^N$ である。

解空間の線型変換不变性： 解空間の線型変換 $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x}$ に対する不変性（線型変換不変性）。ただし， $\mathbf{C} \in \mathbb{R}^{N \times N}$ ， \mathbf{C} は正則とする。線型変換は回転変換やスケール変換を一般化した変換である。

解空間のアフィン変換不变性： 解空間のアフィン変換 $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x} - \mathbf{t}$ に対する不変性（アフィン変換不变性）。ただし， $\mathbf{t} \in \mathbb{R}^N$ ， $\mathbf{C} \in \mathbb{R}^{N \times N}$ ， \mathbf{C} は正則とする。アフィン変換は線形変換と平行移動の合成変換である。

目的関数の単調増加変換： 目的関数の単調増加変換 $T_f : f \mapsto g \circ f$ に対する不変性（目的関数の単調増加変換不变性）。ただし， $g : \mathbb{R} \rightarrow \mathbb{R}$ は単調増加変換（Order-Preserving Transformation, Monotonically Increasing Transformation, Isotonic Transformation）であり，順序集合を維持する。

図を用いて，BBOにおいて重要な変換不变性の例を詳細に説明する。図 4.3 に解空間のアフィン変換を与えたときの目的関数の等高線を示す。例えば，図 4.3(a) の Sphere 関数（2次形式の標準型）の等高線は座標軸に等方で，直交した同心円であるため，変数間独立性と良スケール性を有しているが，解空間の変換 $T_s : \mathbf{x} \mapsto \mathbf{B}\mathbf{D}\mathbf{x}$ （スケール変換と回転変換の合成）を適用すると，図 4.3(b) のように，目的関数の等高線は座標軸に異方で，斜向した橙円となり，変数間依存性と悪スケール性を有するようになる。加えて，平行移動 $T_s : \mathbf{x} \mapsto \mathbf{x} - \mathbf{t}$ を適用すると，等高線全体が平行移動する。アフィン変換不变性を具備する最適化アルゴリズムは，この変換後の関数に適用した場合，変換前の単純な Sphere 関数に適用した場合と同様の探索を行うことが可能である。つまり，最適化アルゴリズムがアフィン変換不变性を具備することは，変数間依存性，悪スケール性，原点依存性に対する性能のロバスト性を有することを表している。さらに，図 4.4 に目的関数の単調増加変換を与えたときの目的関数を示す。図 4.4(a) のように，目的関数が全体的には单峰性だとし



(a) 座標軸に等方かつ直交する目的関数の等高線
(b) 座標軸に異方かつ斜交する目的関数の等高線

図 4.3：解空間のアフィン変換の例

ても、部分的に微分不可能・不連続な点を含む（非凸）場合、数理最適化は適用できない、あるいは目的関数に指數変換が適用されている場合、数理最適化の探索は影響を受ける。しかしながら、目的関数の単調増加変換不变性を有するアルゴリズムは、変換前の関数に適用した場合、図 4.4(b) の凸かつ連続という単純な関数に適用した場合と同様の探索を行うことが可能である。また、BBO ではない場合、目的関数のクラスや性質が変換に対応している問題クラスに対して性能の良し悪しが予想できるため、適切なアルゴリズムを選択することが可能となる。発見的手法であるメタヒューリスティクスの研究では、各種のアルゴリズムの性能と各種の目的関数の対応関係は数値実験から裏付けられることがほとんどであったことから、以上の変換不变性のように汎用が高く、何らかの保証が得られる理論的性質を具備することの意義は大きい。以上から、解空間のアフィン変換不变性と目的関数の変換单調増加変換不变性は、BBO において特に重要であり、メタヒューリスティクスのアルゴリズム設計において有望な性質だと考えられる。

定義 4.1 を用いて最適化アルゴリズムの更新則の変換不变性を明らかにするには、下記の手順を実行すれば良い。

Step1 : 更新則で使用されている各量が、ある変換 T_s , T_f の影響をどのように受けるのかを導出する。

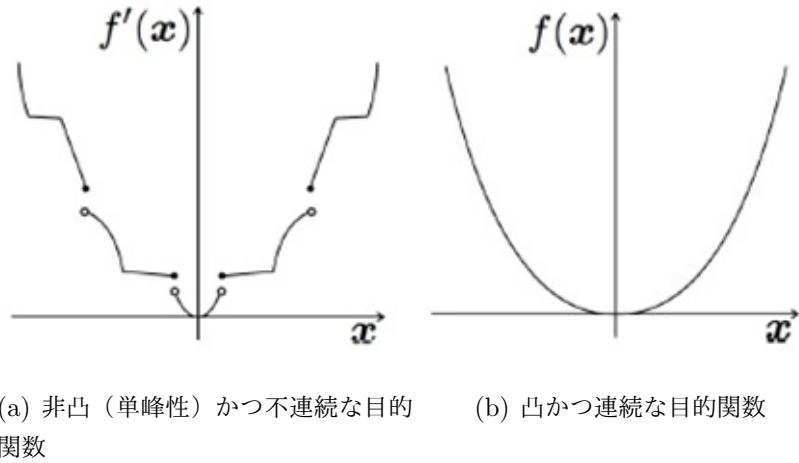


図 4.4：目的関数の単調増加変換の例

Step2 : Step1で求めた変換前後の量を用いて、式(4.1)あるいは式(4.2)の条件を満たすかどうかを具体的に導出する。

Step3 : 式(4.1)あるいは式(4.2)の条件を満たせば、その更新則はある変換 T_s , T_f に対する不变性を具備することが示され、そうでなければ、その不变性を欠如することが示される。

本論文では、この方法を「不变性の解析フレームワーク」と呼ぶことにする。定義 4.1 を「最適化アルゴリズムの変換不变性」と呼ぶ理由は、この定義を基にした不变性の解析フレームワークが勾配法、古典的な直接探索法、メタヒューリスティクスなど、一般の反復解法に属する最適化アルゴリズムに対して適用可能であるためである³。また、定義 4.1 は、Hansen らにより提示されている不变性の定義[93, 163]を参考にしている。[93, 163]は、 T_s, T_f の全体集合を用いて群として定義しており、具体的に適用する際にはその部分群を抜き出し、変換群として扱っているが、定義 4.1 と本質的な違いはない。しかしながら、Hansen の定義を用いて不变性を具体的に示した事例は、Covariance Matrix Adaptation ES (CMA-ES) のみに限定されている[93, 163]。他に、不变性の定義とその定義を活用した解析方法を提示しているのは、Wilke らの方法がある[91]が、解空間の平行移動不变性、回転不变性、相似変換不变性の 3 種類に限定しており、より一般のスケール不变性、ア

³ ただし、勾配ベクトルは目的関数の微分を用いるため、解空間の変換の影響に対する考え方には注意が必要である。

フィン変換不变性や目的関数の単調増加変換不变性の有無を調べることはできない。一方、本章では、定義 4.1に基づき、最適化アルゴリズムの更新則 U の不变性を解析する方法に従い、各種メタヒューリスティクスを対象として、より一般的かつ BBO で重要な、解空間のアフィン変換不变性と目的関数の単調増加変換不变性について証明によって明らかにする。場合によっては、より具体的な変換に対する不变性の具備を確認し、狭義な変換不变性の欠如を特定する。

なお、図 4.1 の下側は解空間の変換 T_s によって変換された探索空間上の更新だが、これに伴い、探索点の位置ベクトルだけでなく、探索履歴やその履歴を引数とする量も変換の影響を受ける。例えば、PSO における p-best や g-best は探索履歴に元々含まれ、FA における差分ベクトルの係数 β_i は探索点間の距離を変数とする正規分布関数である。また、図 4.2 の下側は目的関数の変換 T_f によって変換された目的関数值だが、これに伴い、目的関数值を引数とする量も変換の影響を受ける。例えば、PSO における p-best や g-best では目的関数值の大小関係を使用し、GA や ABC におけるルーレット選択では目的関数值に基づく適合度を使用する。本論文では、数学的な厳密さを欠くが、上記の量 h が T_s 、 T_f によって変換されたものを h' と便宜上表現する。

4.2.2 メタヒューリスティクスの変換不变性に関する先行研究

メタヒューリスティクスの変換不变性に関する先行研究について整理する。Evolution Strategy (ES) の変換不变性に関する先行研究は、Hansen らの研究が先駆的であり、最も有名である。Hansen らは、メタヒューリスティクスにおける変換不变性の概念を提示し、ES のスケール不变性の欠如を指摘すると同時に、それに基づき Covariance Matrix Adaptation ES (CMA-ES) を構築している[88]。第 3 章で説明した (1+1)-ES は、ES のスケールパラメータを探索過程で適応的に調整することで、スケール不变性を補完していると解釈できる。多点型 ES である (μ, λ) -ES や CMA-ES は、多変量標準正規乱数ベクトルに、対角行列と回転行列を作用させることで、共分散行列を有する多変量正規乱数ベクトルに変換することで、アフィン変換不变性を獲得している[88, 93]。また、Hansen は変換不变性の定義に従い、CMA-ES がアフィン変換不变性と目的関数の単調増加変換不变性を具備することを示している[93, 163]。

実数値 Genetic Algorithm (実数値 GA) の変換不変性に関する先行研究は、主に交叉の方法に注目している。小野らや Salomon らは、実数値 GA のビットストリング交叉[59]や、BLX- α [164], SBX [165]が目的関数の変数間依存性に対して性能が劣化することを指摘している[166, 187, 188]。Ras らも、Wilke らの方法[91]を用いて BLX- α の回転不変性の欠如を指摘すると同時に、超球を導入した Rotationally Invariant Continuous Parameter Genetic Algorithm を提案している[122]が、この方法はスケール不変性を有していない。阪井らは、BLX- α に、探索点群の共分散行列に基づく基底変換行列を導入した、Transformed Blend Crossover (TBLX- α) を提案している[189]。さらに、小野らは変数間依存性に着目し、UNDX を提案しており[166]、実数値 GA の交叉の中でも最も有名な方法となっている。しかし、Tsutsui らは、UNDX が悪スケール性に対処できないことを指摘すると同時に、アフィン変換不変性を有する SPX を提案した[167]。他にも、小林らは、SPX を一般化した REX を提案している[170]。SPX と REX は、線型結合のみを使用することで、アフィン変換不変性を獲得している。なお、ルーレット選択は目的関数値の単調増加変換不変性を有さないが、ランキング選択のような目的関数値のランキングに変更することで、容易に付加することが可能である。また、Ras らにより、Wilke らの方法[91]に従い、BLX- α が回転不変性を欠如することと、Rotationally Invariant Continuous Parameter Genetic Algorithm が回転不変性を具備することが示されている[122]。しかしながら、この方法はスケール不変性を有していない。

Particle Swarm Optimization (PSO) の変換不変性に関する先行研究も、多く存在している。Raquet らは PSO の差分ベクトルの係数をスカラにした Linear PSO を提案しており、回転不変性の獲得が目的ではないが、更新則が線形結合となったため、結果としてアフィン変換不変性を獲得している[90]。Bonyadi らは、Linear PSO はアフィン変換不変性を有しているが、多様性を欠如していることを指摘すると同時に、p-best と g-best に多変量標準正規乱数ベクトルの摂動を与えることで多様性を確保しながら回転不変性を有する Locally Convergence Rotationally Invariant PSO を提案している[97]。Wilke らは、不变性を解析する方法を PSO に適用することで、PSO が回転不変性を欠如していることを指摘すると同時に、差分ベクトルに乱数の回転角度を有する回転行列を作用することで、回転不変性を有する PSO を提案している[91]。同様に、Bonyadi らも、差分ベクトルに回転行列を作用することで、回転不変性を有する PSO を提案している[96]。他にも、Clerk は近傍生成に超球を導入した Standard PSO-2011 を提案[95]し、Hariya らは超立方体を導入し

た Norm Linked PSO [98] と Standard PSO にスケール変換を与える Modified Standard PSO を提案[99]し, Shindo らは p-best 集団の共分散行列を有する多変量正規乱数ベクトルを導入した Linear PSO を提案し[100, 101], 串田らは Differential Evolution の突然変異を導入した Linear PSO を提案している[102]。

Differential Evolution (DE) の変換不変性に関する先行研究については, 突然変異はアフィン変換不変性を有しているが, 交叉は一般に回転不変性を有していないため, 交叉の方法についていくつか存在している。Sutton らは, DE の一様交叉は交叉率 $CR = 1$ の場合のみ回転不変性を有しているが, $CR < 1$ のときは回転不変性を有していないことを数値実験的に指摘している[103]。Takahama らは, Gram-Schmidt の正規直交化法に基づく回転行列を生成し, DE の交叉に適用した DE with a Rotation-Invariant Crossover Operation (RIDE) を提案している[104]。Tvrdík らは, 滑降シンプレックス法の反射操作を DE の交叉とした DE を提案している[107]。Li ら, Wang ら, Guo らは, 探索点群の共分散行列に基づく回転行列を生成し, DE の交叉に適用した DE を提案している[105, 106, 109, 110]。金政らは, 超球交叉を導入した DE を提案しており[108], 大場らは, 超立方体交叉を導入した DE を提案している[190]。

Artificial Bee Colony Algorithm (ABC) の変換不変性に関する先行研究については, 著者の知る限りないが, 変数間依存性の課題を解消するためのアプローチが結果として回転不変性を具備している方法がいくつか存在している。He らは, 近傍生成則に g-best 方向に向かう差分ベクトルを導入した ABC を提案している[191]が, 交叉と同様の解の組合せを除外していないため, 完全には回転不変性を具備していない。鈴木らは, 近傍生成則における解の組合せを複数の次元で行う操作に変更した ABC を提案している[192]が, 完全には回転不変性を具備していない。Mori らは, 近傍生成則に探索点群の共分散行列による回転行列を導入した ABC を提案しており[193], 回転不変性を具備している。

他のメタヒューリスティクスに対する変換不変性に関する先行研究は少ない。Cleghorn らは, Firefly Algorithm (FA) に対して, Wilke らの方法[91]を用いて相似変換不変性・スケール変換不変性の欠如を指摘すると同時に, 数値実験を通じてスケール変換に対する性能がロバストではないことを指摘している[124]が, 著者の知る限り, [124]以外の FA の変換不変性に関する先行研究はない。Cuckoo Search や実数値 Simulated Annealing については, 著者の知る限り, 変換不変性に関する先行研究は全くない。

以上の先行研究は, 手法の変換不変性やロバスト性について数理的・数値実験的に解析

する、あるいは欠如している変換不変性を付加する点で共通している。アルゴリズムの種類毎に、欠如しているスケール不変性、あるいは回転不変性を付加するアプローチは、下記のように大別できる。

- ES はスケール不変性を欠如しているため、共分散行列の固有値行列や対角行列を多変量標準正規乱数ベクトルに作用することで、スケール不変性を付加する。しかし、この方法は、回転不変性が犠牲になるため、共分散行列の固有ベクトルに基づく基底変換行列や回転行列をさらに作用することで、アフィン変換不変性を付加している[88, 93]。
- 実数値 GA において、① 回転不変性が欠如している交叉に、超球を導入することで、回転不変性を付加している[122]、② 回転不変性が欠如している交叉において、探索点群の共分散行列の固有ベクトルや、正規直交基底ベクトルに基づく回転行列を導入することで、回転不変性を付加している[166, 189]、③ 交叉を、線型演算のみで構成することで、アフィン変換不変性を付加している[167, 170]。
- PSO は回転不変性を欠如しているため、近傍生成で、① 超球や超立方体を導入することで、回転不変性を付加している[95, 98, 99]、② 回転行列を作用することで、回転不変性を付加している[91, 96]、③ 線型演算のみで構成することで、アフィン変換不変性を付加している[90, 97, 100, 101, 102]。
- DE は回転不変性を欠如しているため、① 交叉に、超球や超立方体を導入することで、回転不変性を付加している[108, 190]、② 交叉に、共分散行列の固有ベクトルに基づく基底変換行列や回転行列を作用することで、回転不変性を付加している[104, 105, 106, 109, 110]、③ 交叉を線型演算のみで構成する式に変更することで、アフィン変換不変性を付加している[107]。ただし、③は、従来の解の組合せの操作から大きく異なる。
- ABC は回転不変性を欠如しているため、近傍生成で、① 共分散行列の固有ベクトルに基づく基底変換行列を作用することで、回転不変性を付加している[193]、② 線型演算のみで構成する式に変更することで、アフィン変換不変性を付加している[191, 192]。ただし、②は、従来の解の組合せの操作から大きく異なる。

上記のアプローチの分類は、3.11 節で分類した近傍生成則の演算形式と深い関連があることがわかる。

(1) 基本ベクトルと差分ベクトルの線形結合

実数値 GA, PSO, DE の突然変異, ABC の式(3.42), CS の Local Search のように, 近傍生成則が基本ベクトルと差分ベクトルの線形結合の場合である。回転不变性を付加するアプローチは, ① 超球を導入する, ② 式(3.52)の係数 \mathbf{A} に共分散行列の固有ベクトルに基づく基底変換行列や回転行列を作用する, である。アフィン変換不变性を付加するアプローチは, ③ 線型演算 $\mathbf{A} = \mathbf{I}$ とする, である。

(2) 解同士の組合せ

DE の交叉, ABC の式(3.40)のように, 近傍生成則が解同士の組合せの場合である。回転不变性を付加するアプローチは, ① 超球を導入する, ② 式(3.53)の係数 $\mathbf{M}_1, \mathbf{M}_2$ に共分散行列の固有ベクトルに基づく基底変換行列や回転行列を作用する, である。アフィン変換不变性を付加するアプローチは, ③ 線型演算 $\mathbf{M}_1 = \mathbf{M}_2 = \mathbf{I}$ とする, である。ただし, ③は, 従来の解の組合せの操作から大きく異なる。

(3) ランダムウォーク

ES, 実数値 Simulated Annealing, CS の Global Search のように, 近傍生成則がランダムウォークの場合である。回転不变性を付加するアプローチは, ① 式(3.55)において, 線型演算 $\mathbf{A} = \mathbf{I}$ とし, 乱数ベクトル s が従う確率分布として, 等方的な確率分布を使用する, ② \mathbf{A} に共分散行列の固有ベクトルに基づく基底変換行列や回転行列を作用する, である。スケール不变性を付加するアプローチは, ③ \mathbf{A} に共分散行列の固有値行列や対角行列を作用する, である。アフィン変換不变性を付加するアプローチは, ④ \mathbf{A} に対角行列と回転行列を同時に作用する, である。

上記の(1)と(2)は, 数理的には同じ線型結合であるため, 両方とも不变性を付加するアプローチはその係数に着目しているものであることがわかる。一方, (3)は探索過程の解とは独立な乱数ベクトルを含むため, (1)と(2)とは異なるアプローチが必要になる。さらに, アプローチの種類によっては, 完全にアフィン変換不变性を獲得することができず, 他の不变性を犠牲にすることもあることもわかる。以降の節では, 各アルゴリズムの変換不变性の有無を明らかにすると同時に, アルゴリズムと上記のアプローチの新たな組合せによって, 不变性を付加することを検討する。

4.2.3 変換不变性と乱数ベクトルに関する重要な定理

多くのメタヒューリスティクスの探索は、ある確率分布に従う乱数を用いることで、探索範囲の多様性を拡張させる。第3章では、更新則において演算形式の分類では乱数が重要であり、使用されるのは、① 差分ベクトルの係数、② 乱数ベクトルであったことを述べ、前項では、不变性の付加アプローチを分けるのは、更新則に乱数ベクトルを含むかどうかであった。本項では、不变性の有無における、乱数に関する重要な定理を示す。

確率変数ベクトルが多変量確率分布（同時確率分布）に従う、つまり $s \sim \mathcal{P}$ であるとき、同時確率分布 \mathcal{P} の性質は、各要素 s_n が従う单変量確率分布 \mathcal{P}_n の性質に依存する。さらに、付録 A に示すように、各要素 s_n が互いに独立であるとき、多変量確率分布は单変量確率分布の重ね合わせ $\mathcal{P} = \prod_n \mathcal{P}_n$ で表される。例えば、確率変数ベクトル s の各要素 s_n が正規分布 $\mathcal{N}(\mu_n, \sigma_n)$ に従い、かつ互いに独立であるとき、 s が従う同時確率分布は、多変量正規分布 $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_n \mathcal{N}(\mu_n, \sigma_n)$ となる。ここで、平均ベクトルは $\boldsymbol{\mu} = [\mu_1, \dots, \mu_N]^T$ 、共分散行列は $\boldsymbol{\Sigma} = \text{diag}[\sigma_1, \dots, \sigma_N]$ である。ただし、一般に共分散行列は、対角要素が分散であり、非対角要素が共分散だが、上記の共分散行列 $\boldsymbol{\Sigma}$ では、対角要素が互いに異なる分散 σ_n であり、かつ共分散が 0 であるような対角行列となる。

メタヒューリスティクスのような最適化アルゴリズムが探索する空間は多次元空間を想定しているため、更新則はベクトルや行列で表記できる。そして多くの場合、更新則内の乱数は要素毎に表記される、つまり、同一の確率分布に従って要素毎に独立に生成されるが、これらをまとめて、ベクトルや行列で表記できる。本論文では、各要素がある確率分布に従う乱数（確率変数）で構成されるベクトルを乱数ベクトル、行列を乱数行列と呼ぶことにする。多くのメタヒューリスティクスの更新式で使用される乱数ベクトルや乱数行列では、各要素の乱数が互いに独立している。例えば、PSO の更新則では、乱数行列 $\mathbf{R}_1, \mathbf{R}_2$ が用いられているが、これは一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に従う対角要素 $r_{1,n}, r_{2,n}$ を持つ対角行列であり、各要素は互いに独立している。他にも、単点型 ES の更新則では、乱数ベクトル s が用いられるが、これは多変量標準正規分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ に従っており、各要素 s_n は互いに独立で、单変量標準正規分布 $\mathcal{N}(0, 1)$ に従う。一方で、乱数ベクトルの各要素の乱数が互いに独立でない、つまり、乱数ベクトルが従う同時確率分布 \mathcal{P} が、各要素の乱数が従う单変量確率分布 \mathcal{P}_n の単なる重ね合わせで表現できず、条件付き確率分布となる場合もある。このとき、各変数間の関係性も表現した分布に従って乱数ベクトルを生成できるため、生成さ

れる範囲の柔軟性が高い。例えば、多点型ESである、CMA-ESの更新則では、多変量正規分布 $\mathcal{N}(\mu, \Sigma)$ に従う乱数ベクトルを探索に用いるが、共分散行列 Σ の非対角要素（共分散）を0とはせず、各変数間の相関を考慮することで、より効果的な乱数ベクトルを生成している。

よく使用される確率分布は、一様分布、正規分布、Lévy 分布などがあり、分布の性質が探索に活用されるが、この多変量確率分布の性質は、最適化アルゴリズムのスケール不变性や回転不变性の有無と直結する。まず、乱数ベクトル、あるいは多変量確率分布は、最適化アルゴリズムの探索で得られる解とは独立であるため、基本的にその解空間のスケール性を考慮できない。このため、乱数ベクトルが含まれる更新則は、スケール不变性を欠如している。さらに、確率分布の性質は、多変量確率分布の確率密度関数の等高線に影響を与える。例えば、各要素が互いに独立である多変量標準正規分布の確率密度関数の等高線は、原点を中心とし、座標軸に直交しながら等方な形状、つまり同心円状となる。一方、各要素が互いに独立だが、各標準偏差が異なる多変量正規分布の確率密度関数の等高線は、原点を中心とし、座標軸に直交しながら異方な形状、つまり橢円状となる。さらに、各要素が相關する多変量正規分布の確率密度関数の等高線は、原点を中心とし、座標軸に斜交した橢円状となる。これは、共分散行列の性質に依存するもので、共分散行列が単位行列 I である場合は、多変量標準正規分布であり、対角行列である場合は、各要素が独立な多変量正規分布であり、非対角要素が0でない対称行列である場合は、各要素が相關する多変量正規分布となる。違う例として、ヘヴィーテイル型の確率分布の場合、各要素が互いに独立していても、各標準基底に沿って確率分布が下る形状、つまり異方性となる。これらの例を見る限り、乱数ベクトルの生成は基本的に標準基底に依存しているため、異方性になりやすいと考えられる。このため、確率分布が異方性である乱数ベクトルが含まれる更新則は、回転不变性を欠如している。よって、更新則は乱数ベクトルを含む場合、基本的にスケール不变性や回転不变性を欠如する。以上の考察から、メタヒューリスティクスの更新則では、多変量標準正規分布を直接用いる場合を除き、更新式で用いられる乱数ベクトルや乱数行列において、各要素の確率変数が互いに独立しているケースが大半であるため、多くのアルゴリズムは、スケール不变性、あるいは回転不变性を欠如していると考えられる。

しかしながら、多変量確率分布の性質をアルゴリズムの探索過程で得られる解に依存することで、不变性を獲得できる場合がある。多変量正規分布に従う乱数ベクトルに関する

定理として、定理 4.1 が成り立つ。

定理 4.1 (多変量正規分布に従う乱数ベクトルに関する定理) 多変量標準正規分布に従う乱数ベクトル $s_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ にアフィン変換 $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x} + \mathbf{t}$ を作用させた乱数ベクトルは、 $\mathbf{s} \sim \mathcal{N}(\mathbf{t}, \mathbf{C}\mathbf{C}^T)$ とみなせる。ただし、 $\mathbf{t} \in \mathbb{R}^N$, $\mathbf{C} \in \mathbb{R}^{N \times N}$, \mathbf{C} は正則とする。

証明. アフィン変換を $T_s : \mathbf{z} \mapsto \mathbf{C}\mathbf{z} + \mathbf{t} = \mathbf{x}$ と表すことにする。付録 A から、多変量標準正規分布の確率密度関数は、下記で表せる。

$$f_{\mathcal{N}}(\mathbf{z}; \mathbf{0}, \mathbf{I}) = \frac{1}{(2\pi)^{N/2}} \exp\left(-\frac{1}{2}\mathbf{z}^T \mathbf{z}\right)$$

アフィン変換 T_s を作用させた乱数ベクトル \mathbf{s} が従う多変量正規分布を $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ と表す。ここで、アフィン変換に対する確率密度関数の変数変換のヤコビアンは、下記で表せる。

$$\left| \frac{\partial(x_1, \dots, x_N)}{\partial(z_1, \dots, z_N)} \right| = \det(\mathbf{C})$$

このとき、変数変換後の確率密度関数は下記で表せるから、代入すると

$$\begin{aligned} f_{\mathcal{N}}(\mathbf{x}) &= f_{\mathcal{N}}(\mathbf{C}^{-1}(\mathbf{x} - \mathbf{t})) \det(\mathbf{C}^{-1}) \\ &= \frac{1}{(2\pi)^{N/2} \det(\mathbf{C})} \exp\left(-\frac{1}{2}(\mathbf{C}^{-1}(\mathbf{x} - \mathbf{t}))^T (\mathbf{C}^{-1}(\mathbf{x} - \mathbf{t}))\right) \\ &= \frac{1}{(2\pi)^{N/2} \det(\mathbf{C})} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{t})^T (\mathbf{C}^{-1})^T \mathbf{C}^{-1} (\mathbf{x} - \mathbf{t})\right) \\ &= \frac{1}{(2\pi)^{N/2} \det(\mathbf{C})} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{t})^T (\mathbf{C}^T)^{-1} \mathbf{C}^{-1} (\mathbf{x} - \mathbf{t})\right) \\ &= \frac{1}{(2\pi)^{N/2} \det(\mathbf{C})} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{t})^T (\mathbf{C}\mathbf{C}^T)^{-1} (\mathbf{x} - \mathbf{t})\right) \end{aligned}$$

上記の正規分布の式と恒等的に比較すると、 $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ において、 $\boldsymbol{\mu} = \mathbf{t}$, $\boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^T$ が得られ、題意が示された。 \square

図 4.5 に定理 4.1 のイメージを示す。定理 4.1 は多変量正規分布に従う乱数ベクトルは、多変量標準正規分布に従う乱数ベクトルをアフィン変換を与えると得られることを表しているが、言い換えると、共分散行列を適切に与えれば、アフィン変換によって変換さ

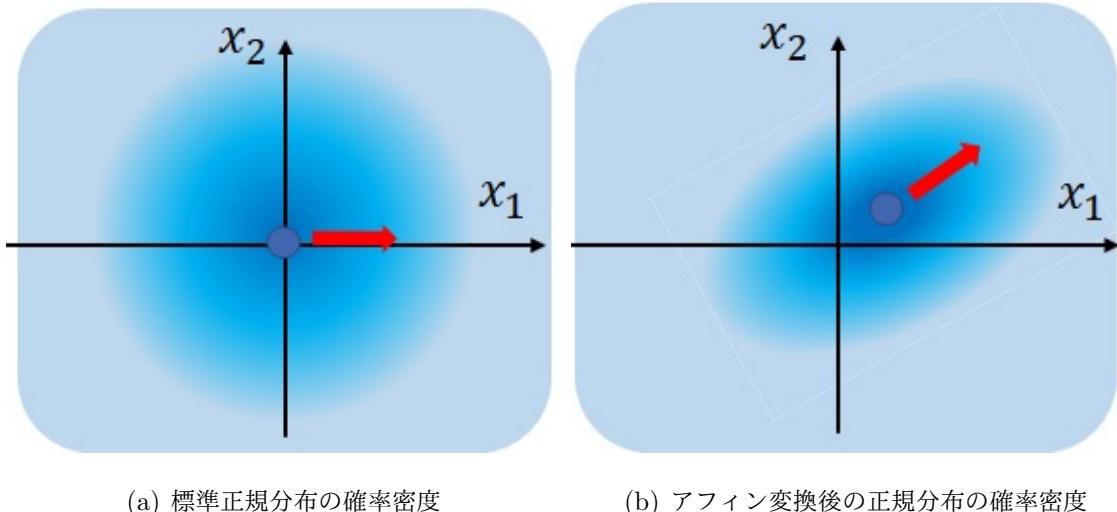


図 4.5： 正規乱数とアフィン変換の関係

れた多変量正規分布に従う乱数ベクトルを得られることを表す。通常、探索とは独立な乱数ベクトルを含む更新則（ランダムウォーク）は、線型変換不变性を失ってしまうが、図 4.5 の性質は乱数ベクトルを探索履歴に依存させ、更新則が線型変換不变性を獲得する効果が期待できる。

さらに、共分散行列 Σ を探索履歴 $\chi(k)$ に含まれる解集合（例えば探索点集団 \mathcal{X} ）を対象とした場合、共分散行列 $\Sigma(\mathcal{X})$ の非対称平方根 $\Sigma_*^{1/2}$ は、式 (4.3) と表せる（補助定理 A.2）。

$$\Sigma_*^{1/2} = \mathbf{P} \Lambda^{1/2} \quad (4.3)$$

ただし、 $\mathbf{P} = [\mathbf{a}_1, \dots, \mathbf{a}_N]$ は基底変換行列、 $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$ は固有値行列である。 $\{(\mathbf{a}_n, \lambda_n) \mid n = 1, 2, \dots, N\}$ ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$) は Σ の固有ベクトルと固有値であり、 Σ の固有値分解によって得られる。そして、非対称平方根 $\Sigma_*^{1/2}$ はアフィン変換に対して、式 (4.4) と変換される（補助定理 A.3）。

$$\Sigma_*^{1/2'} = \mathbf{C} \Sigma_*^{1/2} \quad (4.4)$$

補助定理 A.2、補助定理 A.3 の詳細は、付録 A に記載している。

式 (4.3)、式 (4.4) を利用すると、定理 4.2 が成り立つ。

定理 4.2 (多変量正規分布に従う乱数ベクトルを含む更新則の変換不変性に関する定理) 多変量正規分布に従う乱数ベクトル $\mathbf{s}(k) \sim \mathcal{N}(\mathbf{0}, \Sigma(\mathcal{X}(k)))$ を含むランダムウォークの更新則 $U_f(\mathbf{x}^i(k); \chi(k))$ は、アフィン変換不変性を有する。ただし、 $\mathcal{X}(k)$ は反復回数 k のときの探索点群である。

証明. 一般に、ランダムウォークによる近傍生成則 $U_{\text{nei}, f, S}(\mathbf{x}^i(k); \chi_S(k), \mathcal{R}(k))$ は下記の式である。

$$\hat{\mathbf{x}}^i(k) = \mathbf{x}^i(k) + \mathbf{s}(k)$$

ここで、 $\mathbf{s}(k) \sim \mathcal{N}(\mathbf{0}, \Sigma(\mathcal{X}(k)))$ について、定理 4.1 から、式 (4.5) が得られる。

$$\mathbf{s}(k) = \Sigma_*^{1/2} \mathbf{s}_0(k) \quad (4.5)$$

ただし、 $\mathbf{s}_0(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ である。式 (4.5) を用いて近傍生成則を書き直すと、下記の式となる。

$$\hat{\mathbf{x}}^i(k) = \mathbf{x}^i(k) + \Sigma_*^{1/2} \mathbf{s}_0(k)$$

ただし、 $\mathcal{R}(k) = \{\mathbf{s}_0(k)\}$ である。つまり、この近傍生成則は、 $\hat{\mathbf{x}}^i(k) \sim \mathcal{N}(\mathbf{x}^i(k), \Sigma(\mathcal{X}(k)))$ であることを表している。ここで、アフィン変換 $\mathbf{x} \mapsto \mathbf{C}\mathbf{x} - \mathbf{t}$ を作用すると、近傍生成で使用する量は変換される。探索点の位置ベクトルは $\mathbf{x}^{i'}(k) = \mathbf{C}\mathbf{x}^i(k) - \mathbf{t}$ となる。多変量標準正規分布に従う乱数ベクトルは探索過程の解とは独立しているため、変換の影響を受けないことから、 $\mathbf{s}'_0(k) = \mathbf{s}_0(k)$ となる。また、 $\Sigma_*^{1/2}$ は探索履歴を使用するため変換の影響を受け、式 (4.4) から $\Sigma_*^{1/2'} = \mathbf{C}\Sigma_*^{1/2}$ となる。これに伴い、式 (4.5) から、 \mathbf{s} は

$$\begin{aligned} \mathbf{s}'(k) &= \Sigma_*^{1/2'} \mathbf{s}'_0(k) \\ &= \mathbf{C}\Sigma_*^{1/2} \mathbf{s}_0(k) \\ &= \mathbf{C}\mathbf{s}(k) \end{aligned}$$

となる。

ここで、題意が成り立つには、近傍生成則 $U_{\text{nei},f,S}(\mathbf{x}^i(k); \chi_S(k), \mathcal{R}(k))$ について、式 (4.1) が成立すればよい（定義 4.1）。

$$\begin{aligned}
 U_{\text{nei},f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) &= \mathbf{x}^{i'}(k) + \Sigma_*^{1/2} \mathbf{s}'_0(k) \\
 &= (\mathbf{C}\mathbf{x}^i(k) - \mathbf{t}) + \mathbf{C}\Sigma_*^{1/2} \mathbf{s}_0(k) \\
 &= \mathbf{C}\mathbf{x}^i(k) + \mathbf{C}\Sigma_*^{1/2} \mathbf{s}_0(k) - \mathbf{t} \\
 &= \mathbf{C}(\mathbf{x}^i(k) + \Sigma_*^{1/2} \mathbf{s}_0(k)) - \mathbf{t} \\
 &= \mathbf{C}\hat{\mathbf{x}}^i(k) - \mathbf{t} \\
 &= T_s \circ U_{\text{nei},f,S}(\mathbf{x}^i(k); \chi_S(k))
 \end{aligned}$$

よって、近傍生成則 $U_{\text{nei},f,S}(\mathbf{x}^i(k); \chi_S(k), \mathcal{R}(k))$ について、式 (4.1) が成立することが示された。探索点の移動則 $U_{\text{mov},f,S}(\hat{\mathbf{x}}^i(k))$ を絶対移動とすれば、 $\mathbf{x}^i(k+1) = \hat{\mathbf{x}}^i(k)$ となるため、式 (4.1) が成立する⁴。よって、これらの合成変換である更新則 $U_{f,S}(\mathbf{x}^i(k); \chi_S(k), \mathcal{R}(k))$ についても、式 (4.1) が成立することから、題意が示された。□

定理 4.2 は、共分散行列 Σ を探索点集団 \mathcal{X} など、探索過程に得られる解集合を対象とした場合、多変量確率分布の性質をアルゴリズムの探索過程で得られる解に依存させることができあり、アフィン変換不变性を獲得できることを表している。CMA-ES では、多変量正規分布に従う乱数ベクトルが用いられているが、これと同様の原理によりアフィン変換不变性を獲得している。加えて、定理 4.2 の証明では、正規分布は自身が持つ共分散行列の特性から、式 (4.5) を導出して、計算過程で用いている。しかしながら、逆にこれを活用し、更新則に式 (4.5) を導入することで、共分散行列の特性のみを活用することができる。つまり、本質的には、定理 4.2 の証明において、 \mathbf{s}_0 を各要素毎が独立な乱数ベクトル（多変量確率分布が单変量確率分布の重ね合わせで表現できる）としても、図 4.5 の性質を獲得できる。これを踏まえると、乱数ベクトルを含む更新則においては、定理 4.2 は強力な定理であり、アフィン変換不变性を必ず具備することが可能である。

⁴ 当然、探索点の移動則において解空間のアフィン変換が関係すれば、影響を確認する必要があるが、改善移動の場合でも、一般的に目的関数値の大小関係のみしか含まないため、絶対移動と結果は同じである。

4.3 変換不变性に基づくメタヒューリスティクスの解析

本節では、4.2節で提示した、不变性の解析フレームワークに基づき、Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony Algorithm (ABC), Firefly Algorithm (FA), Cuckoo Search (CS)について、不变性の有無を明らかにする。

4.3.1 変換不变性に基づく Particle Swarm Optimization の解析

(a) Particle Swarm Optimization のアフィン変換不变性に関する証明

Particle Swarm Optimization (PSO)に関する定理として、定理 4.3, 定理 4.4, 定理 4.5を示すことで、PSOが解空間のスケール不变性・平行移動不变性・目的関数の単調増加不变性を有していることと、回転不变性を欠如（アフィン変換不变性を欠如）していることを明らかにする。

定理 4.3 (PSO の解空間のスケール不变性・平行移動不变性に関する定理) PSO の更新則 $U_{f,S}$ は解空間のスケール不变性・平行移動不变性を有する。ただし、スケール変換と平行移動の合成変換は $T_s : \mathbf{x} \mapsto \mathbf{Cx} - \mathbf{t}$, \mathbf{C} は対角行列である。

証明. 解空間のスケール変換・平行移動 T_s のもとで、更新則 $U_{f,S}$ の引数への影響を考える。探索点の位置ベクトル $\mathbf{x}^i(k)$ ($i = 1, \dots, m$), p-best $\mathbf{p}^i(k)$, g-best $\mathbf{p}^g(k)$ は、式(3.25), 式(2.16)より、 $\mathbf{x}^i(k), \mathbf{p}^i(k), \mathbf{p}^g(k) \in \chi_S(k)$ だから、 $\mathbf{x}^{i'}(k) = \mathbf{Cx}^i(k) - \mathbf{t}$, $\mathbf{p}^{i'}(k) = \mathbf{Cp}^i(k) - \mathbf{t}$, $\mathbf{p}^{g'}(k) = \mathbf{Cp}^g(k) - \mathbf{t}$ となる。探索点の移動ベクトル $\mathbf{v}^i(k+1)$ は式(3.24)より、 $\mathbf{v}^{i'}(k+1)$ は

$$\begin{aligned}\mathbf{v}^{i'}(k+1) &= \mathbf{x}^{i'}(k+1) - \mathbf{x}^{i'}(k) \\ &= (\mathbf{Cx}^i(k+1) - \mathbf{t}) - (\mathbf{Cx}^i(k) - \mathbf{t}) \\ &= \mathbf{C}(\mathbf{x}^i(k+1) - \mathbf{x}^i(k)) \\ &= \mathbf{Cv}^i(k+1)\end{aligned}$$

となるから、 $\mathbf{v}^{i'}(k) = \mathbf{Cv}^i(k)$ と表される。

ここで、題意が成り立つには、PSO の更新則 $U_{f,S}$ について、式 (4.1) が成立すればよい (定義 4.1)。よって、

$$\begin{aligned}
& U_{f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\
&= \mathbf{x}^{i\prime}(k) + w\mathbf{v}^{i\prime}(k) + c_1\mathbf{R}_1(\mathbf{p}^{i\prime} - \mathbf{x}^{i\prime}) + c_2\mathbf{R}_2(\mathbf{p}^{g\prime} - \mathbf{x}^{i\prime}) \\
&= (\mathbf{C}\mathbf{x}^i(k) - \mathbf{t}) + w\mathbf{C}\mathbf{v}^i(k) + c_1\mathbf{R}_1((\mathbf{C}\mathbf{x}^i(k) - \mathbf{t}) - (\mathbf{C}\mathbf{p}^i(k) - \mathbf{t})) \\
&\quad + c_2\mathbf{R}_2((\mathbf{C}\mathbf{x}^i(k) - \mathbf{t}) - (\mathbf{C}\mathbf{p}^g(k) - \mathbf{t})) \\
&= \mathbf{C}(\mathbf{x}^i(k) + w\mathbf{v}^i(k)) + c_1\mathbf{R}_1\mathbf{C}(\mathbf{x}^i(k) - \mathbf{p}^i(k)) + c_2\mathbf{R}_2\mathbf{C}(\mathbf{x}^i(k) - \mathbf{p}^g(k)) - \mathbf{t} \\
&= \mathbf{C}(\mathbf{x}^i(k) + w\mathbf{v}^i(k)) + c_1\mathbf{C}\mathbf{R}_1(\mathbf{x}^i(k) - \mathbf{p}^i(k)) + c_2\mathbf{C}\mathbf{R}_2(\mathbf{x}^i(k) - \mathbf{p}^g(k)) - \mathbf{t} \\
&= \mathbf{C}(\mathbf{x}^i(k) + w\mathbf{v}^i(k) + c_1\mathbf{R}_1(\mathbf{x}^i(k) - \mathbf{p}^i(k)) + c_2\mathbf{R}_2(\mathbf{x}^i(k) - \mathbf{p}^g(k))) - \mathbf{t} \\
&= \mathbf{C}\mathbf{x}^i(k+1) - \mathbf{t} \\
&= (T_s \circ U_{f,S})(\mathbf{x}^i(k); \chi_S(k))
\end{aligned}$$

対角行列の積は交換可能だから、 $\mathbf{C}\mathbf{R}_\ell = \mathbf{R}_\ell\mathbf{C}$ ($\ell = 1, 2$) であることを証明過程で利用した。以上の証明により、題意が示された。 \square

定理 4.4 (PSO の解空間の回転不変性に関する定理) PSO の更新則 $U_{f,S}$ は解空間の回転不変性を有さない。ただし、回転変換は $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x}$, \mathbf{C} は正規直交行列 ($\mathbf{C}^\mathrm{T} = \mathbf{C}^{-1}$) である。

証明. 解空間の回転変換 T_s のもとで、更新則 $U_{f,S}$ の引数への影響を考える。探索点の位置ベクトル $\mathbf{x}^i(k)$ ($i = 1, \dots, m$), p-best $\mathbf{p}^i(k)$, g-best $\mathbf{p}^g(k)$ は、式 (3.25), 式 (2.16) より、 $\mathbf{x}^i(k), \mathbf{p}^i(k), \mathbf{p}^g(k) \in \chi_S(k)$ だから、 $\mathbf{x}^{i\prime}(k) = \mathbf{C}\mathbf{x}^i(k)$, $\mathbf{p}^{i\prime}(k) = \mathbf{C}\mathbf{p}^i(k)$, $\mathbf{p}^{g\prime}(k) = \mathbf{C}\mathbf{p}^g(k)$ となる。探索点の移動ベクトル $\mathbf{v}^i(k+1)$ は式 (3.24) より、 $\mathbf{v}^{i\prime}(k+1)$ は

$$\begin{aligned}
\mathbf{v}^{i\prime}(k+1) &= \mathbf{x}^{i\prime}(k+1) - \mathbf{x}^{i\prime}(k) \\
&= \mathbf{C}\mathbf{x}^i(k+1) - \mathbf{C}\mathbf{x}^i(k) \\
&= \mathbf{C}(\mathbf{x}^i(k+1) - \mathbf{x}^i(k)) \\
&= \mathbf{C}\mathbf{v}^i(k+1)
\end{aligned}$$

となるから、 $\mathbf{v}^{i\prime}(k) = \mathbf{C}\mathbf{v}^i(k)$ と表される。

ここで、題意が成り立つには、PSO の更新則 $U_{f,S}$ について、式(4.1)が成立しなければよい（定義 4.1）。よって、

$$\begin{aligned}
 & U_{f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\
 &= \mathbf{x}^{i\prime}(k) + w\mathbf{v}^{i\prime}(k) + c_1\mathbf{R}_1(\mathbf{p}^{i\prime} - \mathbf{x}^{i\prime}) + c_2\mathbf{R}_2(\mathbf{p}^{g\prime} - \mathbf{x}^{i\prime}) \\
 &= \mathbf{C}\mathbf{x}^i(k) + w\mathbf{C}\mathbf{v}^i(k) + c_1\mathbf{R}_1(\mathbf{C}\mathbf{x}^i(k) - \mathbf{C}\mathbf{p}^i(k)) + c_2\mathbf{R}_2(\mathbf{C}\mathbf{x}^i(k) - \mathbf{C}\mathbf{p}^g(k)) \\
 &= \mathbf{C}(\mathbf{x}^i(k) + w\mathbf{v}^i(k)) + c_1\mathbf{R}_1(\mathbf{x}^i(k) - \mathbf{p}^i(k)) + c_2\mathbf{R}_2(\mathbf{x}^i(k) - \mathbf{p}^g(k)) - \mathbf{t} \\
 &\neq \mathbf{C}(\mathbf{x}^i(k) + w\mathbf{v}^i(k)) + c_1\mathbf{C}\mathbf{R}_1(\mathbf{x}^i(k) - \mathbf{p}^i(k)) + c_2\mathbf{C}\mathbf{R}_2(\mathbf{x}^i(k) - \mathbf{p}^g(k)) \\
 &= \mathbf{C}(\mathbf{x}^i(k) + w\mathbf{v}^i(k) + c_1\mathbf{R}_1(\mathbf{x}^i(k) - \mathbf{p}^i(k)) + c_2\mathbf{R}_2(\mathbf{x}^i(k) - \mathbf{p}^g(k))) \\
 &= \mathbf{C}\mathbf{x}^i(k+1) \\
 &= (T_s \circ U_{f,S})(\mathbf{x}^i(k); \chi_S(k))
 \end{aligned}$$

一般に行列の積は交換不可能だから、 $\mathbf{C}\mathbf{R}^\ell \neq \mathbf{R}^\ell\mathbf{C}$ ($\ell = 1, 2$)であることを証明過程で利用した。以上の証明により、題意が示された。 \square

定理 4.5 (PSO の目的関数の単調増加変換不变性に関する定理) PSO の更新則 $U_{f,S}$ は目的関数の単調増加変換不变性を有する。ただし、目的関数の単調増加変換は $T_f : f(\mathbf{x}) \mapsto g \circ f(\mathbf{x})$, g は単調増加変換である。

証明. 目的関数の単調増加変換 T_f のもとで、更新則 $U_{f,S}$ の引数の影響を考える。探索点の位置ベクトル $\mathbf{x}^i(k)$ ($i = 1, \dots, m$) は $\mathbf{x}^i(k) \in \chi_S(k)$ だから、 $\mathbf{x}^{i\prime}(k) = \mathbf{x}^i(k)$ となる。探索点の移動ベクトル $\mathbf{v}^i(k+1)$ は式(3.24)より、 $\mathbf{v}^{i\prime}(k+1) = \mathbf{x}^{i\prime}(k+1) - \mathbf{x}^{i\prime}(k) = \mathbf{x}^i(k+1) - \mathbf{x}^i(k) = \mathbf{v}^i(k+1)$ となる。

また、探索点と p-best の目的関数值 $f(\mathbf{x}^i(k)), f(\mathbf{p}^i(k))$ ($i = 1, \dots, m$) は、 $f(\mathbf{x}^i(k)), f(\mathbf{p}^i(k)) \in \zeta_f(k)$ だから、 $f'(\mathbf{x}^i(k)) = g \circ f(\mathbf{x}^i(k)), f'(\mathbf{p}^i(k)) = g \circ f(\mathbf{p}^i(k))$ となる。ただし、 $\zeta_f(k)$ は探索履歴 $\chi_S(k)$ に対応する目的関数值集合である。

下記では、 $f(\mathbf{x}^i(k+1)) \leq f(\mathbf{p}^i(k))$ が成立すると仮定する。このとき、

$$\begin{aligned}
 & f'(\mathbf{x}^i(k+1)) \leq f'(\mathbf{p}^i(k)) \\
 & \Leftrightarrow g \circ f(\mathbf{x}^i(k+1)) \leq g \circ f(\mathbf{p}^i(k)) \\
 & \Leftrightarrow f(\mathbf{x}^i(k+1)) \leq f(\mathbf{p}^i(k))
 \end{aligned}$$

となるから、 $f'(\mathbf{x}^i(k+1)) \leq f'(\mathbf{p}^i(k))$ が成り立つ。これは、 T_f に対して目的関数値の大小関係が不変であることを表す。 $f(\mathbf{x}^i(k+1)) \leq f(\mathbf{p}^i(k))$ が成り立つとき、式(3.25)から、 $\mathbf{p}^i(k+1) = \mathbf{x}^i(k+1)$ となり、 T_f が与えられたとき、 $f'(\mathbf{x}^i(k+1)) \leq f'(\mathbf{p}^i(k))$ となるから、 $\mathbf{p}^{i\prime}(k+1) = \mathbf{x}^i(k+1)$ となる。よって、 $\mathbf{p}^{i\prime}(k) = \mathbf{p}^i(k)$ となる。 $f(\mathbf{x}^i(k+1)) > f(\mathbf{p}^i(k))$ の場合も同様の手順によって、 $\mathbf{p}^{i\prime}(k) = \mathbf{p}^i(k)$ が得られる。

また、式(3.15)により、p-best 集合 $\mathcal{P}(k) = \{\mathbf{p}^1(k), \dots, \mathbf{p}^m(k)\}$ の各 p-best に目的関数値の順位を付け、順位の非減少順にソーティングする。このとき、式(2.16)から、 $\mathbf{p}^g(k)$ は $f(\mathbf{p}^g(k)) \leq f(\mathbf{p}^2(k)) \leq \dots \leq f(\mathbf{p}^m(k))$ を満たす。このとき、

$$\begin{aligned} f'(\mathbf{p}^g(k)) &\leq f'(\mathbf{p}^2(k)) \leq \dots \leq f'(\mathbf{p}^m(k)) \\ \Leftrightarrow g \circ f(\mathbf{p}^g(k)) &\leq g \circ f(\mathbf{p}^2(k)) \leq \dots \leq g \circ f(\mathbf{p}^m(k)) \\ \Leftrightarrow f(\mathbf{p}^g(k)) &\leq f(\mathbf{p}^2(k)) \leq \dots \leq f(\mathbf{p}^m(k)) \end{aligned}$$

となるから、 $\mathbf{p}^{g\prime}(k) = \mathbf{p}^g(k)$ となる。

ここで、題意が成り立つには、PSO の更新則 $U_{f,S}$ について、式(4.1)が成立すればよい(定義 4.1)。よって、

$$\begin{aligned} &U_{T_f \circ f, S}(\mathbf{x}^i(k); \chi_S(k)) \\ &= \mathbf{x}^{i\prime}(k) + w\mathbf{v}^{i\prime}(k) + c_1 \mathbf{R}_1(\mathbf{p}^{i\prime} - \mathbf{x}^{i\prime}) + c_2 \mathbf{R}_2(\mathbf{p}^{g\prime} - \mathbf{x}^{i\prime}) \\ &= \mathbf{x}^i(k) + w\mathbf{v}^i(k) + c_1 \mathbf{R}_1(\mathbf{p}^i - \mathbf{x}^i) + c_2 \mathbf{R}_2(\mathbf{p}^g - \mathbf{x}^i) \\ &= \mathbf{x}^i(k+1) \\ &= U_{f,S}(\mathbf{x}^i(k); \chi_S(k)) \end{aligned} \tag{4.6}$$

以上の証明により、題意が示された。 \square

(b) PSO の回転不変性に関する数値実験的検証

数値実験的検証を通じて、PSO の回転不変性の欠如を指摘する。具体的には、回転変換 $T_s : \mathbf{x} \mapsto \mathbf{B}(\theta)\mathbf{x}$ のもとで、 $\mathbf{x}_A \neq \mathbf{x}_B$ が成立することを示せばよい(定義 4.1)。なお、 $\mathbf{B}(\theta)$ は解空間を θ deg 回転させる回転行列であり、付録 A を参照されたい。

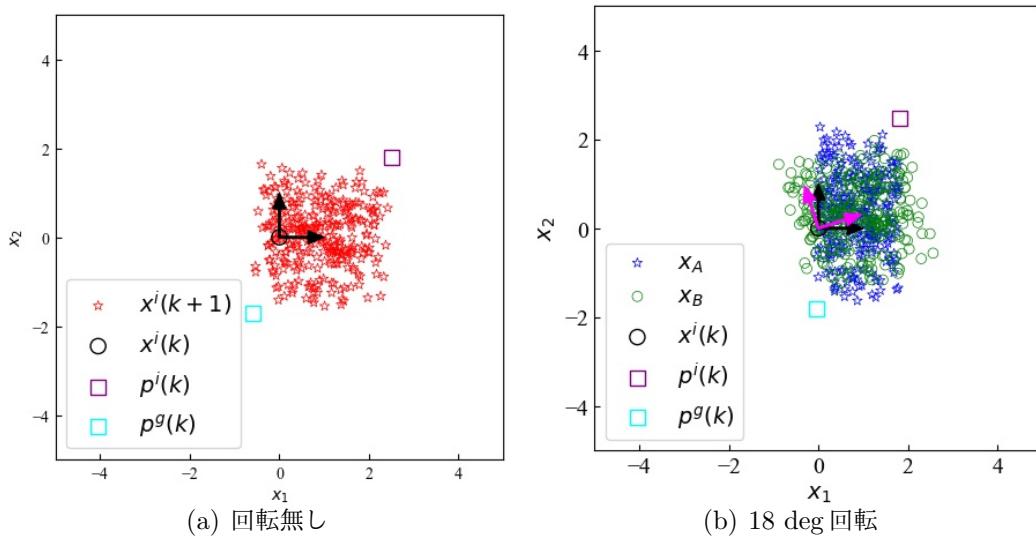


図 4.6：解空間の回転前後の近傍 (PSO)

図 4.6 に, \mathbb{R}^2 上の同じ探索点, p-best, g-best を用いて, PSO の更新則 $U_{f,S}$ で x_A と x_B を 500 回生成した図を示す。簡潔に議論するために, パラメータは $w = 0$, $c_1 = c_2 = 1$ とし, $\theta = 18\text{deg}$ とした。なお, 慣性項は回転不变性を有するため, $w = 0$ として慣性項の影響を除いても, 議論に支障はない。黒い矢印は標準基底, 桃色の矢印は回転した標準基底, 原点の \circ は探索点 $x^i(k)$, \triangle は p-best $p^i(k)$, \square は g-best $p^g(k)$ を表す。図 4.6(a) は, 解空間を回転させる前に生成した近傍, 図 4.6(b) は, 18 deg 回転後に生成した近傍 x_A と, 生成後の解 (図 4.6(a) の近傍) を 18 deg 回転させた近傍 x_B を示している。

図 4.6 から、PSOにおいて、 $\mathbf{x}_A \neq \mathbf{x}_B$ が成立することが確認できる。また、PSO の近傍は、解空間の回転に関わらず、標準基底に沿って生成されることも確認できる。これは、更新則における差分ベクトルの係数乱数が対角行列であることに起因する。以上から、前半で示した結果と同様に、数値実験的に PSO は回転不变性を有していないことが示された。この回転不变性の欠如が、解空間の回転前後で探索性能が大幅に変動する原因となる。

4.3.2 変換不变性に基づく Differential Evolution の解析

Differential Evolution (DE) に関する定理として、定理 4.6, 定理 4.7, 定理 4.8 を示すことで、DE の更新則が解空間のスケール不変性・平行移動不変性・目的関数の単調增加変換不変性を有していることと、解空間の回転不変性を欠如（アフィン変換不変性を

欠如) していることを明らかにする。

定理 4.6 (DE の解空間のスケール不変性・平行移動不変性に関する定理) DE の更新則 $U_{f,S}$ は解空間のスケール不変性・平行移動不変性に対する不変性を有する。ただし、スケール変換と平行移動の合成変換は $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x} - \mathbf{t}$, \mathbf{C} は対角行列である。

証明. 解空間のスケール変換・平行移動 T_s のもとで、近傍生成則 $U_{\text{nei},f,S}$ の引数の影響を考える。探索点の位置ベクトル $\mathbf{x}^j(k)$ ($j = i, r_1, r_2, r_3$) は $\mathbf{x}^j(k) \in \chi_S(k)$ だから、 $\mathbf{x}^{j'}(k) = \mathbf{C}\mathbf{x}^j(k) - \mathbf{t}$ となる。このとき、式 (3.30) の $\mathbf{u}^i(k)$ は

$$\begin{aligned}\mathbf{u}^{i'}(k) &= \mathbf{x}^{r_1'}(k) + F(\mathbf{x}^{r_2'}(k) - \mathbf{x}^{r_3'}(k)) \\ &= (\mathbf{C}\mathbf{x}^{r_1}(k) - \mathbf{t}) + F((\mathbf{C}\mathbf{x}^{r_2}(k) - \mathbf{t}) - (\mathbf{C}\mathbf{x}^{r_3}(k) - \mathbf{t})) \\ &= \mathbf{C}\mathbf{x}^{r_1}(k) + F\mathbf{C}(\mathbf{x}^{r_2}(k) - \mathbf{x}^{r_3}(k)) - \mathbf{t} \\ &= \mathbf{C}(\mathbf{x}^{r_1}(k) + F(\mathbf{x}^{r_2}(k) - \mathbf{x}^{r_3}(k))) - \mathbf{t} \\ &= \mathbf{C}\mathbf{u}^i(k) - \mathbf{t}\end{aligned}$$

となるから、 $\mathbf{u}^{i'}(k) = \mathbf{C}\mathbf{u}^i(k) - \mathbf{t}$ と表される。

DE の近傍生成則 $U_{\text{nei},f,S}$ による $\hat{\mathbf{x}}^i(k)$ は下記のように求められる。

$$\begin{aligned}\hat{\mathbf{x}}^{i'}(k) &= U_{\text{nei},f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\ &= \mathbf{M}^1(\mathbf{C}\mathbf{x}^i(k) - \mathbf{t}) + \mathbf{M}^2(\mathbf{C}\mathbf{u}^i(k) - \mathbf{t}) \\ &= \mathbf{M}^1\mathbf{C}\mathbf{x}^i(k) - \mathbf{M}^1\mathbf{t} + \mathbf{M}^2\mathbf{C}\mathbf{u}^i(k) - \mathbf{M}^2\mathbf{t} \\ &= \mathbf{C}\mathbf{M}^1\mathbf{x}^i(k) + \mathbf{C}\mathbf{M}^2\mathbf{u}^i(k) - (\mathbf{M}^1 + \mathbf{M}^2)\mathbf{t} \\ &= \mathbf{C}(\mathbf{M}^1\mathbf{x}^i(k) + \mathbf{M}^2\mathbf{u}^i(k)) - \mathbf{I}\mathbf{t} \\ &= \mathbf{C}\hat{\mathbf{x}}^i(k) - \mathbf{t}\end{aligned}$$

対角行列の積は交換可能だから、 $\mathbf{C}\mathbf{M}^\ell = \mathbf{M}^\ell\mathbf{C}$ ($\ell = 1, 2$) であることを計算過程で利用した。また、 $\mathbf{M}^1 + \mathbf{M}^2 = \mathbf{I}$ を利用した。

さらに、解空間のスケール変換・平行移動 T_s のもとで、探索点の移動則 $U_{\text{mov},f,S}$ の引数の影響を考える。式 (3.32) において、 $\mathbf{x}^{i'}(k) = \mathbf{C}\mathbf{x}^i(k) - \mathbf{t}$, $\hat{\mathbf{x}}^{i'}(k) = \mathbf{C}\hat{\mathbf{x}}^i(k) - \mathbf{t}$ となる。また、下記では、 $f(\hat{\mathbf{x}}^i(k)) \leq f(\mathbf{x}^i(k))$ が成立すると仮定し、解空間のスケール変換・平行移動 T_s の影響を受けないとする。DE の探索点の移動則 $U_{\text{mov},f,S}$ による $\mathbf{x}^i(k+1)$ は下記

のように求められる。

$$\begin{aligned}\mathbf{x}^{i\prime}(k+1) &= U_{\text{mov},f,S'}(T_s(\hat{\mathbf{x}}^i(k)); \chi_{S'}(k)) \\ &= \hat{\mathbf{x}}^{i\prime}(k) \\ &= \mathbf{C}\hat{\mathbf{x}}^i(k) - \mathbf{t} \\ &= \mathbf{C}\mathbf{x}^i(k+1) - \mathbf{t}\end{aligned}$$

ここで、題意が成り立つには、DE の更新則 $U_{f,S}$ について、式 (4.1) が成立すればよい (定義 4.1)。よって、

$$\begin{aligned}&U_{f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\ &= \left(U_{\text{mov},f,S'} \circ U_{\text{nei},f,S'} \right)(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\ &= \mathbf{x}^{i\prime}(k+1) \\ &= \mathbf{C}\mathbf{x}^i(k+1) - \mathbf{t} \\ &= \left(T_s \circ U_{\text{mov},f,S} \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i(k); \chi_S(k)) \\ &= \left(T_s \circ U_{f,S} \right)(\mathbf{x}^i(k); \chi_S(k))\end{aligned}$$

なお、これは $f(\hat{\mathbf{x}}^i(k)) > f(\mathbf{x}^i(k))$ としても成立する。以上の証明により、題意が示された。 \square

定理 4.7 (DE の解空間の回転不変性に関する定理) DE の更新則 $U_{f,S}$ は解空間の回転不変性を有さない。ただし、回転変換は $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x}$, \mathbf{C} は正規直交行列 ($\mathbf{C}^T = \mathbf{C}^{-1}$) である。

証明. 解空間の回転変換 T_s のもとで、近傍生成則 $U_{\text{nei},f,S}$ の引数の影響を考える。探索点の位置ベクトル $\mathbf{x}^j(k)$ ($j = i, r_1, r_2, r_3$) は $\mathbf{x}^j(k) \in \chi_S(k)$ だから、 $\mathbf{x}^{j\prime}(k) = \mathbf{C}\mathbf{x}^j(k)$ となる。このとき、式 (3.30) の $\mathbf{u}^i(k)$ は

$$\begin{aligned}\mathbf{u}^{i\prime}(k) &= \mathbf{x}^{r_1\prime}(k) + F(\mathbf{x}^{r_2\prime}(k) - \mathbf{x}^{r_3\prime}(k)) \\ &= \mathbf{C}\mathbf{x}^{r_1}(k) + F(\mathbf{C}\mathbf{x}^{r_2}(k) - \mathbf{C}\mathbf{x}^{r_3}(k)) \\ &= \mathbf{C}\mathbf{x}^{r_1}(k) + F\mathbf{C}(\mathbf{x}^{r_2}(k) - \mathbf{x}^{r_3}(k)) \\ &= \mathbf{C}(\mathbf{x}^{r_1}(k) + F(\mathbf{x}^{r_2}(k) - \mathbf{x}^{r_3}(k))) \\ &= \mathbf{C}\mathbf{u}^i(k)\end{aligned}$$

となるから、 $\mathbf{u}^{i\prime}(k) = \mathbf{C}\mathbf{u}^i(k)$ と表される。

ここで、題意が成り立つには、DE の更新則 $U_{f,S}$ について、式 (4.1) が成立しなければよい（定義 4.1）。DE の近傍生成則 $U_{\text{nei},f,S}$ について、

$$\begin{aligned} & U_{\text{nei},f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\ &= \mathbf{M}^1 \mathbf{C} \mathbf{x}^i(k) + \mathbf{M}^2 \mathbf{C} \mathbf{u}^i(k) \\ &\neq \mathbf{C} \mathbf{M}^1 \mathbf{x}^i(k) + \mathbf{C} \mathbf{M}^2 \mathbf{u}^i(k) \\ &= \mathbf{C} (\mathbf{M}^1 \mathbf{x}^i(k) + \mathbf{M}^2 \mathbf{u}^i(k)) \\ &= \mathbf{C} \hat{\mathbf{x}}^i(k) \\ &= (T_s \circ U_{\text{nei},f,S})(\mathbf{x}^i(k); \chi_S(k)) \end{aligned}$$

のように、式 (4.1) が成立しないことが確認される。一般に行列の積は交換不可能だから、 $\mathbf{C} \mathbf{M}^\ell \neq \mathbf{M}^\ell \mathbf{C}$ ($\ell = 1, 2$) であることを計算過程で利用した。よって、全体の更新則 $U_{f,S}$ についても、同様に式 (4.1) が成立しない。以上の証明により、題意が示された。 \square

定理 4.8 (DE の目的関数の単調増加変換不变性に関する定理) DE の更新則 U_f は目的関数の単調増加変換不变性を有する。ただし、目的関数の単調増加変換は $T_f : f(\mathbf{x}) \mapsto g \circ f(\mathbf{x})$, g は単調増加変換である。

証明. 目的関数の単調増加変換 T_f のもとで、近傍生成則 $U_{\text{nei},f,S}$ の引数の影響を考える。探索点の位置ベクトル $\mathbf{x}^j(k)$ ($j = i, r_1, r_2, r_3$) は $\mathbf{x}^j(k) \in \chi_S(k)$ だから、 $\mathbf{x}^{j\prime}(k) = \mathbf{x}^j(k)$ となる。式 (3.30) は目的関数値 $f(\mathbf{x})$ を含まないため、 $\mathbf{u}^{i\prime}(k) = \mathbf{u}^i(k)$ となる。DE の近傍生成則 $U_{\text{nei},f,S}$ による $\hat{\mathbf{x}}^i(k)$ は下記のように求められる。

$$\begin{aligned} \hat{\mathbf{x}}^{i\prime}(k) &= U_{\text{nei},T_f \circ f,S}(\mathbf{x}^i(k); \chi_S(k)) \\ &= \mathbf{M}^1 \mathbf{x}^{i\prime}(k) + \mathbf{M}^2 \mathbf{u}^{i\prime}(k) \\ &= \mathbf{M}^1 \mathbf{x}^i(k) + \mathbf{M}^2 \mathbf{u}^i(k) \\ &= U_{\text{nei},f,S}(\mathbf{x}^i(k); \chi_S(k)) \\ &= \hat{\mathbf{x}}^i(k) \end{aligned}$$

さらに、目的関数の単調増加変換 T_f のもとで、探索点の移動則 $U_{\text{mov},f,S}$ の引数の影響を考える。式 (3.32) において、 $\mathbf{x}^{i\prime}(k) = \mathbf{x}^i(k)$, $\hat{\mathbf{x}}^{i\prime}(k) = \hat{\mathbf{x}}^i(k)$ となる。 $f'(\mathbf{x}^i(k)) =$

$g \circ f(\mathbf{x}^i(k)), f'(\hat{\mathbf{x}}^i(k)) = g \circ f(\hat{\mathbf{x}}^i(k))$ となる。下記では、 $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定する。このとき、

$$\begin{aligned} & f'(\hat{\mathbf{x}}^i(k)) \leq f'(\mathbf{x}^i(k)) \\ \Leftrightarrow & g \circ f(\hat{\mathbf{x}}^i(k)) \leq g \circ f(\mathbf{x}^i(k)) \\ \Leftrightarrow & f(\hat{\mathbf{x}}^i(k)) \leq f(\mathbf{x}^i(k)) \end{aligned}$$

となるため、 $f'(\hat{\mathbf{x}}^i(k)) \leq f'(\mathbf{x}^i(k))$ が成り立つ。これは、 T_f に対して目的関数値の大小関係が不变であることを表す。また、式(3.32)で、 $f(\hat{\mathbf{x}}^i(k)) \leq f(\mathbf{x}^i(k))$ が成り立つとき、 $\mathbf{x}^i(k+1) = \hat{\mathbf{x}}^i(k)$ となる。DE の探索点の移動則 $U_{\text{mov},f,S}$ による $\mathbf{x}^i(k+1)$ は下記のように求められる。

$$\begin{aligned} \mathbf{x}^{i\prime}(k+1) &= U_{\text{mov},T_f \circ f}(\mathbf{x}^i(k); \chi_S(k)) \\ &= \hat{\mathbf{x}}^{i\prime}(k) \\ &= \hat{\mathbf{x}}^i(k) \\ &= U_{\text{mov},f}(\mathbf{x}^i(k); \chi_S(k)) \\ &= \mathbf{x}^i(k+1) \end{aligned}$$

ここで、題意が成り立つには、DE の更新則 $U_{f,S}$ について、式(4.1)が成立すればよい(定義 4.1)。よって、

$$\begin{aligned} & U_{T_f \circ f,S}(\mathbf{x}^i(k); \chi_S(k)) \\ &= \left(U_{\text{mov},T_f \circ f,S} \circ U_{\text{nei},T_f \circ f,S} \right)(\mathbf{x}^i(k); \chi_S(k)) \\ &= \mathbf{x}^{i\prime}(k+1) \\ &= \mathbf{x}^i(k+1) \\ &= \left(U_{\text{mov},f,S} \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i(k); \chi_S(k)) \\ &= U_{f,S}(\mathbf{x}^i(k); \chi_S(k)) \end{aligned}$$

また、 $f(\hat{\mathbf{x}}^i(k)) > f(\mathbf{x}^i(k))$ のときも上記の論証は成立する。以上の証明により、題意が示された。□

4.3.3 変換不変性に基づく Artificial Bee Colony Algorithm の解析

(a) Artificial Bee Colony Algorithm のアフィン変換不変性に関する証明

Artificial Bee Colony Algorithm (ABC) に関する定理として、定理 4.9, 定理 4.10, 定理 4.11 を示すことで、ABC の更新則が解空間のスケール不変性・平行移動不変性を有していることと、解空間の回転不変性・目的関数の単調増加変換不変性を欠如している（解空間のアフィン変換不変性を欠如）ことを指摘する。

定理 4.9 (ABC の解空間のスケール不変性・平行移動不変性に関する定理) ABC の更新則 $U_{f,S}$ は解空間のスケール不変性・平行移動不変性に対する不変性を有する。ただし、スケール変換と平行移動の合成変換は $T_s : \mathbf{x} \mapsto \mathbf{Cx} - \mathbf{t}$, \mathbf{C} は対角行列である。

証明. 解空間のスケール変換・平行移動 T_s のもとで、近傍生成則 $U_{\text{nei},f,S}$ の引数の影響を考える。探索点の位置ベクトル \mathbf{x}^j ($j = i, r$) は $\mathbf{x}^j \in \chi_S$ だから、 $\mathbf{x}^{j'} = \mathbf{Cx}^j - \mathbf{t}$ となる。このとき、式 (3.42) の \mathbf{u}^i は

$$\begin{aligned}\mathbf{u}^{i'} &= \mathbf{x}^{i'} + \phi(\mathbf{x}^{r'} - \mathbf{x}^{i'}) \\ &= \mathbf{Cx}^i - \mathbf{t} + \phi((\mathbf{Cx}^r - \mathbf{t}) - (\mathbf{Cx}^i - \mathbf{t})) \\ &= \mathbf{Cx}^i + \phi\mathbf{C}(\mathbf{x}^r - \mathbf{x}^i) - \mathbf{t} \\ &= \mathbf{C}(\mathbf{x}^i + \phi(\mathbf{x}^r - \mathbf{x}^i)) - \mathbf{t} \\ &= \mathbf{Cu}^i - \mathbf{t}\end{aligned}$$

となるから、 $\mathbf{u}^{i'} = \mathbf{Cu}^i - \mathbf{t}$ と表される。

ここで、ABC の近傍生成則 $U_{\text{nei},f,S}$ による $\hat{\mathbf{x}}^i$ は下記のように求められる。

$$\begin{aligned}\hat{\mathbf{x}}^{i'} &= U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \mathbf{M}^1(\mathbf{Cx}^i - \mathbf{t}) + \mathbf{M}^2(\mathbf{Cu}^i - \mathbf{t}) \\ &= \mathbf{M}^1\mathbf{Cx}^i - \mathbf{M}^1\mathbf{t} + \mathbf{M}^2\mathbf{Cu}^i - \mathbf{M}^2\mathbf{t} \\ &= \mathbf{CM}^1\mathbf{x}^i + \mathbf{CM}^2\mathbf{u}^i - (\mathbf{M}^1 + \mathbf{M}^2)\mathbf{t} \\ &= \mathbf{C}(\mathbf{M}^1\mathbf{x}^i + \mathbf{M}^2\mathbf{u}^i) - \mathbf{It} \\ &= \mathbf{C}\hat{\mathbf{x}}^i - \mathbf{t}\end{aligned}$$

対角行列の積は交換可能だから、 $\mathbf{C}\mathbf{M}^\ell = \mathbf{M}^\ell\mathbf{C}$ ($\ell = 1, 2$) であることを計算過程で利用した。また、 $\mathbf{M}^1 + \mathbf{M}^2 = \mathbf{I}$ を利用した。

さらに、解空間のスケール変換・平行移動 T_s のもとで、探索点の移動則 $U_{\text{mov},f,S}$ の引数の影響を考える。式(3.32)において、 $\mathbf{x}^{i\prime} = \mathbf{C}\mathbf{x}^i - \mathbf{t}$, $\hat{\mathbf{x}}^{i\prime} = \mathbf{C}\hat{\mathbf{x}}^i - \mathbf{t}$ となる。また、下記では、 $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定し、解空間のスケール変換・平行移動 T_s の影響を受けないとする。ABC の探索点の移動則 $U_{\text{mov},f,S}$ による $\mathbf{x}^i(k+1)$ は下記のように求められる。

$$\begin{aligned}\mathbf{x}^{i\prime} &:= U_{\text{mov},f,S'}(T_s(\hat{\mathbf{x}}^i); \chi_{S'}) \\ &= \hat{\mathbf{x}}^{i\prime} \\ &= \mathbf{C}\hat{\mathbf{x}}^i - \mathbf{t} \\ &= \mathbf{C}\mathbf{x}^i - \mathbf{t}\end{aligned}$$

ここで、題意が成り立つには、ABC の更新則 $U_{f,S}$ について、式(4.1) が成立すればよい(定義 4.1)。よって、

$$\begin{aligned}U_{f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) &= \left(U_{\text{mov},f,S'} \circ U_{\text{nei},f,S'} \right)(T_s(\mathbf{x}^i); \chi_{S'}) \\ &:= \mathbf{x}^{i\prime} \\ &= \mathbf{C}\mathbf{x}^i - \mathbf{t} \\ &= \left(T_s \circ U_{\text{mov},f,S} \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i; \chi_S) \\ &= \left(T_s \circ U_{f,S} \right)(\mathbf{x}^i; \chi_S)\end{aligned}$$

なお、これは $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ としても成立する。以上の証明により、題意が示された。□

定理 4.10 (ABC の解空間の回転不変性に関する定理) ABC の更新則 $U_{f,S}$ は解空間の回転不変性を有さない。ただし、回転変換は $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x}$, \mathbf{C} は正規直交行列 ($\mathbf{C}^T = \mathbf{C}^{-1}$) である。

証明. 解空間の回転変換 T_s のもとで、近傍生成則 $U_{\text{nei},f,S}$ の引数の影響を考える。探索点の位置ベクトル \mathbf{x}^j ($j = i, r$) は $\mathbf{x}^j \in \chi_S$ だから、 $\mathbf{x}^{j\prime} = \mathbf{C}\mathbf{x}^j$ となる。このとき、式(3.42)

の \mathbf{u}^i は

$$\begin{aligned}\mathbf{u}^{i\prime} &= \mathbf{x}^{i\prime} + \phi(\mathbf{x}^{r\prime} - \mathbf{x}^{i\prime}) \\ &= \mathbf{C}\mathbf{x}^i + \phi(\mathbf{C}\mathbf{x}^r - \mathbf{C}\mathbf{x}^i) \\ &= \mathbf{C}\mathbf{x}^i + \phi\mathbf{C}(\mathbf{x}^r - \mathbf{x}^i) \\ &= \mathbf{C}(\mathbf{x}^i + \phi(\mathbf{x}^r - \mathbf{x}^i)) \\ &= \mathbf{C}\mathbf{u}^i\end{aligned}$$

となるから、 $\mathbf{u}^{i\prime} = \mathbf{C}\mathbf{u}^i$ と表される。

ここで、題意が成り立つには、ABC の更新則 $U_{f,S}$ について、式 (4.1) が成立しなければよい（定義 4.1）。ABC の近傍生成則 $U_{\text{nei},f,S}$ について、

$$\begin{aligned}&U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \mathbf{M}^1 \mathbf{C} \mathbf{x}^i + \mathbf{M}^2 \mathbf{C} \mathbf{u}^i \\ &\neq \mathbf{C} \mathbf{M}^1 \mathbf{x}^i + \mathbf{C} \mathbf{M}^2 \mathbf{u}^i \\ &= \mathbf{C}(\mathbf{M}^1 \mathbf{x}^i + \mathbf{M}^2 \mathbf{u}^i) \\ &= \mathbf{C} \hat{\mathbf{x}}^i \\ &= \left(T_s \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i; \chi_S)\end{aligned}$$

のように、式 (4.1) が成立しないことが確認される。一般に行列の積は交換不可能だから、 $\mathbf{C} \mathbf{M}^\ell \neq \mathbf{M}^\ell \mathbf{C}$ ($\ell = 1, 2$) であることを計算過程で利用した。よって、全体の更新則 $U_{f,S}$ についても、同様に式 (4.1) が成立しない。以上の証明により、題意が示された。□

定理 4.11 (ABC の目的関数の単調増加変換不变性に関する定理) ABC の更新則 $U_{f,S}$ は目的関数の単調増加変換不变性を有さない。ただし、目的関数の単調増加変換は $T_f : f(\mathbf{x}) \mapsto g \circ f(\mathbf{x})$, g は単調増加変換である。

証明. 目的関数の単調増加変換 T_f のもとで、更新則 $U_{f,S}$ の引数の影響を考える。

$f'(\mathbf{x}^i) = g \circ f(\mathbf{x}^i)$, $f'(\hat{\mathbf{x}}^i) = g \circ f(\hat{\mathbf{x}}^i)$ となる。下記では、 $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定する。このとき、

$$\begin{aligned}f'(\mathbf{x}^i) - f'(\hat{\mathbf{x}}^i) &= g \circ f(\mathbf{x}^i) - g \circ f(\hat{\mathbf{x}}^i) \\ &= g \circ (f(\mathbf{x}^i) - f(\hat{\mathbf{x}}^i)) \geq 0\end{aligned}$$

となるから、 $f'(\hat{\mathbf{x}}^i) \leq f'(\mathbf{x}^i)$ が成り立つ。これは、 T_f に対して目的関数値の大小関係が不变であることを表す。

式 (3.40) において、 T_f に対して $\hat{\mathbf{x}}^i$ は目的関数値を含まないため、 $\hat{\mathbf{x}}^{i'} = \hat{\mathbf{x}}^i$ となる。また、式 (3.36) で、 $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成り立つとき、 $\mathbf{x}^i := \hat{\mathbf{x}}^i$ となる。また、 T_f が与えられたとき、 $f'(\hat{\mathbf{x}}^i) \leq f'(\mathbf{x}^i)$ が成り立つため、 $\mathbf{x}^{i'} := \hat{\mathbf{x}}^{i'} = \hat{\mathbf{x}}^i$ となる。

ここで、題意が成り立つには、ABC の更新則 $U_{f,S}$ について、式 (4.1) が成立しなければよい（定義 4.1）。よって、

$$\begin{aligned} U_{T_f \circ f, S}(\mathbf{x}^i; \chi_S) &= \hat{\mathbf{x}}^{i'} \\ &= \hat{\mathbf{x}}^i \\ &= U_{f, S}(\mathbf{x}^i; \chi_S) \end{aligned}$$

また、 $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ のときも上記の論証は成立する。よって、働き蜂と傍観蜂の式 (3.36) は目的関数の単調増加変換不变性を有することを示した。

次に、傍観蜂のルーレット選択の影響を考える。下記では、 $f(\mathbf{x}^i) > 0$ が成立すると仮定する。 T_f のとき、適合度 fit_i と選択確率 p_i は、

$$\begin{aligned} fit'_i &= \frac{1}{1 + g \circ f(\mathbf{x}^i)} \\ &\neq g \circ \left(\frac{1}{1 + f(\mathbf{x}^i)} \right) \\ &= g \circ fit_i \end{aligned}$$

$$\begin{aligned} p'_i &= \frac{fit'_i}{\sum_{j=1}^m fit'_j} \\ &\neq \frac{fit_i}{\sum_{j=1}^m fit_j} \\ &= p_i \end{aligned}$$

となる。また、 $f(\mathbf{x}^i) \leq 0$ のときも $p'_i \neq p_i$ となる。よって、 T_f のもとで、傍観蜂のルーレット選択によって選択される探索点は、不变でない。以上の証明により、題意が示された。□

(b) ABC の回転不变性に関する数値実験的検証

数値実験的検証を通じて、ABC の回転不变性の欠如を指摘する。具体的には、回転変換 $T_s : \mathbf{x} \mapsto \mathbf{B}(\theta)\mathbf{x}$ のもとで、 $\mathbf{x}_A \neq \mathbf{x}_B$ が成立することを示せばよい（定義 4.1）。なお、 $\mathbf{B}(\theta)$ は解空間を θ deg 回転させる回転行列であり、付録 A を参照されたい。

図 4.7 に、 \mathbb{R}^2 上の同じ探索点を用いて、ABC の更新則 $U_{f,S}$ で \mathbf{x}_A と \mathbf{x}_B を 300 回生成した図を示す。黒い矢印は標準基底、桃色の矢印は回転した標準基底、原点の \circlearrowleft は \mathbf{x}^i 、□は \mathbf{x}^r 、 $\theta = 18\text{deg}$ である。図 4.7(a) は、解空間を回転させる前に生成した近傍、図 4.7(b) は、18 deg 回転後に生成した近傍 \mathbf{x}_A と、生成後の解（図 4.7(a) の近傍）を 18 deg 回転させた近傍 \mathbf{x}_B を示している。

図 4.7 より、① ABC では $\mathbf{x}_A = \mathbf{x}_B$ が成立しないこと、② ABC では、 \mathbf{u}^i が \mathbf{x}^i と \mathbf{x}^r を通る線上に生成された後、近傍解 $\hat{\mathbf{x}}^i$ が \mathbf{u}^i を標準基底上に射影した箇所に生成されるため、近傍が標準基底に依存していること、という 2 つの結果が確認できる。また、図 4.8(a) は、 $\phi = 0.5$ に固定した状態で生成した ABC の近傍を示している。この図から、ABC の近傍は、半径が $\|(\mathbf{x}^i - \mathbf{u}^i)/2\|_2$ 、中心が $(\mathbf{x}^i + \mathbf{u}^i)/2$ である超球の表面上に近傍解が生成されるが、対角行列 $\mathbf{M}^1, \mathbf{M}^2$ をかけることで超球表面と標準基底の交点に拘束されることがわかる。以上から、数値実験的に ABC は回転不变性を有していないことが明らかとなった。

4.3.4 変換不变性に基づく Firefly Algorithm の解析

Firefly Algorithm (FA) に関する定理として、定理 4.12、定理 4.13、定理 4.14 を示すことで、FA の更新則が解空間の回転不变性・平行移動不变性・目的関数の単調増加変換不变性を有していることと、解空間の相似変換不变性・スケール不变性を欠如している（解空間のアフィン変換不变性を欠如）ことを明らかにする。

定理 4.12 (FA の解空間の回転不变性・平行移動不变性に関する定理) FA の更新則 $U_{f,S}$ は解空間の回転不变性・平行移動不变性に対する不变性を有する。ただし、回転変換と平行移動の合成変換は $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x} - \mathbf{t}$ 、 \mathbf{C} は正規直交行列 ($\mathbf{C}^T = \mathbf{C}^{-1}$) である。

証明. 解空間の回転変換・平行移動 T_s のもとで、更新則 $U_{f,S}$ の引数の影響を考える。探索

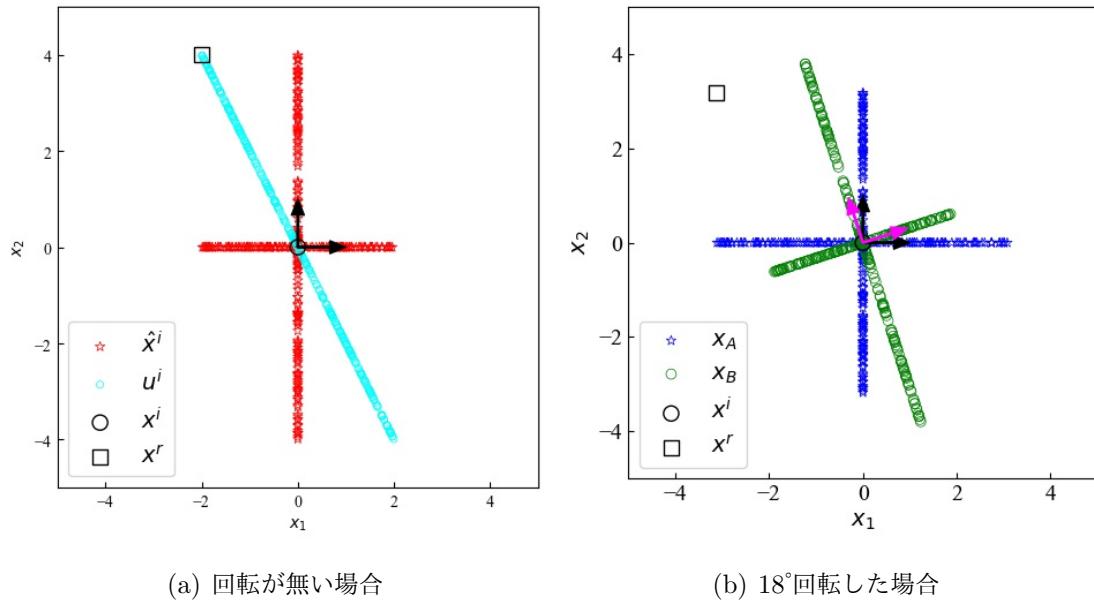
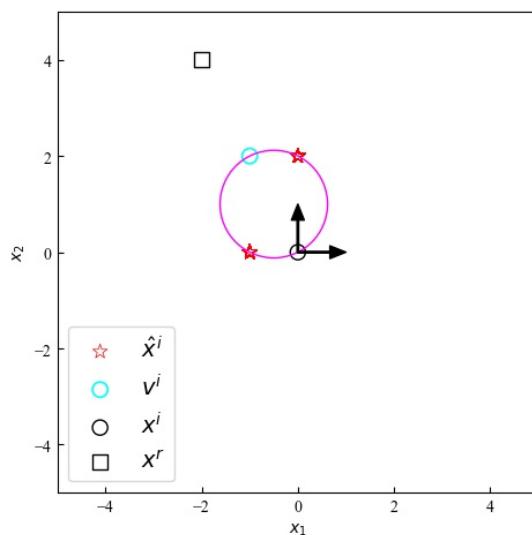


図 4.7：ABC の近傍生成

図 4.8： $\phi = 0.5$ に固定したときのABC の近傍生成

点の位置ベクトル $\mathbf{x}^i(k)$ と保存位置 \mathbf{y}^j は $\mathbf{x}^i(k), \mathbf{y}^j \in \chi_S(k)$ だから, $\mathbf{x}^{i\prime}(k) = \mathbf{C}\mathbf{x}^i(k) - \mathbf{t}$, $\mathbf{y}^{j\prime} = \mathbf{C}\mathbf{y}^j - \mathbf{t}$ となる。このとき, 式(3.45)の β は

$$\begin{aligned}\beta' &= \beta_0 \exp(-\gamma \| \mathbf{y}^{j\prime} - \mathbf{x}^{i\prime}(k) \|_2^2) \\ &= \beta_0 \exp(-\gamma \| (\mathbf{C}\mathbf{y}^j - \mathbf{t}) - (\mathbf{C}\mathbf{x}^i(k) - \mathbf{t}) \|_2^2) \\ &= \beta_0 \exp(-\gamma \| \mathbf{C}(\mathbf{y}^j - \mathbf{x}^i(k)) \|_2^2) \\ &= \beta_0 \exp(-\gamma \| \mathbf{y}^j - \mathbf{x}^i(k) \|_2^2) \\ &= \beta\end{aligned}$$

となるから, $\beta' = \beta$ と表される。計算過程で, 直交行列の性質 $\| \mathbf{C}(\mathbf{y}^j - \mathbf{x}^i(k)) \|_2 = \| \mathbf{y}^j - \mathbf{x}^i(k) \|_2$ であることを利用した。

また, \mathbf{s} は多変量標準正規分布 $N(\mathbf{0}, \mathbf{I})$ に従う乱数ベクトルであるため, 回転しても同じベクトルだとみなせる。よって, $\mathbf{s}' = \mathbf{Cs} = \mathbf{s}$ と表せる。

ここで, 題意が成り立つには, FA の更新則 $U_{f,S}$ について, 式(4.1)が成立すればよい(定義 4.1)。よって,

$$\begin{aligned}U_{f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\ &= \mathbf{x}^{i\prime}(k) + \beta'(\mathbf{y}^{j\prime} - \mathbf{x}^{i\prime}(k)) + \alpha\mathbf{s}' \\ &= \mathbf{C}\mathbf{x}^i(k) - \mathbf{t} + \beta((\mathbf{C}\mathbf{y}^j - \mathbf{t}) - (\mathbf{C}\mathbf{x}^i(k) - \mathbf{t})) + \alpha\mathbf{Cs} \\ &= \mathbf{C}\mathbf{x}^i(k) + \beta\mathbf{C}(\mathbf{y}^j - \mathbf{x}^i(k)) + \alpha\mathbf{Cs} - \mathbf{t} \\ &= \mathbf{C}(\mathbf{x}^i(k) + \beta(\mathbf{y}^j - \mathbf{x}^i(k)) + \alpha\mathbf{s}) - \mathbf{t} \\ &= \mathbf{C}\mathbf{x}^i(k+1) - \mathbf{t} \\ &= \left(T_s \circ U_{f,S}\right)(\mathbf{x}^i(k); \chi_S(k))\end{aligned}$$

以上の証明により, 題意が示された。 \square

定理 4.13 (FA の解空間の相似変換不変性・スケール不変性に関する定理) FA の更新則 $U_{f,S}$ は解空間の相似変換不変性・スケール不変性を有さない。ただし, 相似変換は $T_s : \mathbf{x} \mapsto \delta\mathbf{x}$, $\delta > 0$ である。

証明. 解空間の相似変換 T_s のもとで, 更新則 $U_{f,S}$ の引数の影響を考える。探索点の位置ベクトル $\mathbf{x}^i(k)$ と保存位置 \mathbf{y}^j は $\mathbf{x}^i(k), \mathbf{y}^j \in \chi_S(k)$ だから, $\mathbf{x}^{i\prime}(k) = \delta\mathbf{x}^i(k)$, $\mathbf{y}^{j\prime} = \delta\mathbf{y}^j$ と

なる。このとき、式(3.45)の β は

$$\begin{aligned}\beta' &= \beta_0 \exp[-\gamma \| \mathbf{y}^{j'} - \mathbf{x}^{i'}(k) \|_2^2] \\ &= \beta_0 \exp[-\gamma \| \delta \mathbf{y}^j - \delta \mathbf{x}^i(k) \|_2^2] \\ &= \beta_0 \exp[-\gamma \| \delta(\mathbf{y}^j - \mathbf{x}^i(k)) \|_2^2] \\ &= \beta_0 \exp[-\gamma \delta^2 \| \mathbf{y}^j - \mathbf{x}^i(k) \|_2^2] \\ &= \beta_0 (\exp[-\gamma \| \mathbf{y}^j - \mathbf{x}^i(k) \|_2^2])^{\delta^2} \\ &= \beta_0 \left(\frac{\beta}{\beta_0} \right)^{\delta^2} \\ &= (\beta_0)^{1-\delta^2} \beta^{\delta^2}\end{aligned}$$

となるから、 $\beta' = (\beta_0)^{1-\delta^2} \beta^{\delta^2}$ と表される。なお、 $\beta_0 > 0$ である。

また、 s は多変量標準正規分布 $N(\mathbf{0}, \mathbf{I})$ に従う乱数ベクトルであるため、スケール変換しても同じベクトルだとみなせる。よって、 $s' = s$ と表せる。

ここで、題意が成り立つには、FA の更新則 $U_{f,S}$ について、式(4.1)が成立しなければよい(定義 4.1)。よって、

$$\begin{aligned}&U_{f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\ &= \mathbf{x}^{i'}(k) + \beta'(\mathbf{y}^{j'} - \mathbf{x}^{i'}(k)) + \alpha s' \\ &= \delta \mathbf{x}^i(k) + (\beta_0)^{1-\delta^2} \beta^{\delta^2} (\delta \mathbf{y}^j - \delta \mathbf{x}^i(k)) + \alpha s \\ &= \delta \mathbf{x}^i(k) + (\beta_0)^{1-\delta^2} \beta^{\delta^2} \delta(\mathbf{y}^j - \mathbf{x}^i(k)) + \alpha s \\ &= \delta(\mathbf{x}^i(k) + (\beta_0)^{1-\delta^2} \beta^{\delta^2} (\mathbf{y}^j - \mathbf{x}^i(k))) + \alpha s \\ &\neq \delta(\mathbf{x}^i(k) + \beta(\mathbf{y}^j - \mathbf{x}^i(k)) + \alpha s) \\ &= \delta \mathbf{x}^i(k+1) \\ &= (T_s \circ U_{f,S})(\mathbf{x}^i(k); \chi_S(k))\end{aligned}$$

以上の証明により、FA が解空間の相似変換不変性を有していないことを示した。

さらに、相似変換は、スケール変換 $x \mapsto \mathbf{C}x$ の対角行列 \mathbf{C} が $\mathbf{C} = \delta \mathbf{I}$ とした場合である。このため、相似変換不変性を持たない場合、スケール不変性も持たない。以上の証明により、FA が解空間のスケール不変性を有していないことを示した。以上から、題意が示された。□

定理 4.14 (FA の目的関数の単調増加変換不变性に関する定理) FA の更新則 $U_{f,S}$ は目的関数の単調増加変換不变性を有する。ただし、目的関数の単調増加変換は T_f : $f(\mathbf{x}) \mapsto g \circ f(\mathbf{x})$, g は単調増加変換である。

証明. 目的関数の単調増加変換 T_f のもとで、更新則 $U_{f,S}$ の引数の影響を考える。式(3.45), 式(3.46)に目的関数値が含まれないが、式(3.43)の光強度には目的関数値が含まれる。式(3.44)における \mathbf{y}^s の選択は光強度の大小関係 $I^s(k) \geq I^i(k)$ に基づいてるが、 $I^s(k) \geq I^i(k) \Leftrightarrow f(\mathbf{y}^s) \leq f(\mathbf{x}^i(k))$ となる。 $f'(\mathbf{x}^i(k)) = g \circ f(\mathbf{x}^i(k))$, $f'(\mathbf{y}^s) = g \circ f(\mathbf{y}^s)$ となる。下記では、 $f(\mathbf{y}^s) \leq f(\mathbf{x}^i(k))$ が成立すると仮定する。このとき、

$$\begin{aligned} f'(\mathbf{y}^s) &\leq f'(\mathbf{x}^i(k)) \\ \Leftrightarrow g \circ f(\mathbf{y}^s) &\leq g \circ f(\mathbf{x}^i(k)) \\ \Leftrightarrow f(\mathbf{y}^s) &\leq f(\mathbf{x}^i(k)) \end{aligned}$$

となるため、 $f'(\mathbf{y}^s) \leq f'(\mathbf{x}^i(k))$ が成り立つ。これは、 T_f に対して目的関数値の大小関係が不变であることを表す。よって、 T_f に対して、 \mathbf{y}^s の選択は不变となる。

ここで、題意が成り立つには、FA の更新則 $U_{f,S}$ について、式(4.1)が成立すればよい(定義 4.1)。よって、

$$\begin{aligned} U_{T_f \circ f, S}(\mathbf{x}^i(k); \chi_S(k)) &= \mathbf{x}^{i\prime}(k+1) \\ &= \mathbf{x}^i(k+1) \\ &= U_{f,S}(\mathbf{x}^i(k); \chi_S(k)) \end{aligned}$$

また、 $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ のときも上記の論証は成立する。よって、題意が示された。 \square

4.3.5 変換不变性に基づく Cuckoo Search の解析

(a) Cuckoo Search のアフィン変換不变性に関する証明

Cuckoo Search (CS) に関する定理として、定理 4.15, 定理 4.16, 定理 4.17 を示すことで、CS の更新則が解空間の平行移動不变性・目的関数の単調増加変換不变性を有して

いることと、解空間の回転不変性・相似変換不変性・スケール不変性を欠如している（解空間のアフィン変換不変性を欠如）ことを明らかにする。

定理 4.15 (CS の解空間の平行移動不変性に関する定理) CS の更新則 $U_{f,S}$ は解空間の平行移動不変性に対する不変性を有する。ただし、平行移動は $T_s : \mathbf{x} \mapsto \mathbf{x} - \mathbf{t}$ である。

証明. 解空間の平行移動 T_s のもとで、更新則 $U_{f,S}$ の引数の影響を考える。探索点の位置ベクトル \mathbf{x}^i ($i = 1, 2, \dots, m$) は $\mathbf{x}^i \in \chi_S$ だから、 $\mathbf{x}^{i'} = \mathbf{x}^i - \mathbf{t}$ となる。また、 \mathbf{s} は安定分布に従う乱数ベクトルであるため、平行移動しても同じベクトルだとみなせる。よって、 $\mathbf{s}' = \mathbf{s}$ と表せる。

ここで、題意が成り立つには、CS の GS (式 (3.48)) と LS (式 (3.50)) の更新則 $U_{f,S}$ について、式 (4.1) が成立すればよい (定義 4.1)。まず、GS の近傍生成則 $U_{\text{nei},f,S}$ による $\hat{\mathbf{x}}^i$ は下記のように求められる。

$$\begin{aligned}\hat{\mathbf{x}}^{i'} &= U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \mathbf{x}^{i'} + \alpha \mathbf{s}' \\ &= \mathbf{x}^i - \mathbf{t} + \alpha \mathbf{s} \\ &= (\mathbf{x}^i + \alpha \mathbf{s}) - \mathbf{t} \\ &= \hat{\mathbf{x}}^i - \mathbf{t} \\ &= \left(T_s \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i; \chi_S)\end{aligned}$$

さらに、解空間の平行移動 T_s のもとで、探索点の移動則 $U_{\text{mov},f,S}$ について考える。下記では、 $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定し、解空間の平行移動 T_s の影響を受けないとする。GS の探索点の移動則 $U_{\text{mov},f,S}$ による \mathbf{x}^i は下記のように求められる。

$$\begin{aligned}\mathbf{x}^{i'} &:= U_{\text{mov},f,S'}(T_s(\hat{\mathbf{x}}^i); \chi_{S'}) \\ &= \hat{\mathbf{x}}^{i'} \\ &= \hat{\mathbf{x}}^i - \mathbf{t} \\ &= \mathbf{x}^i - \mathbf{t} \\ &= \left(T_s \circ U_{\text{mov},f,S} \right)(\mathbf{x}^i; \chi_S)\end{aligned}$$

ここで、題意が成り立つには、GS の更新則 $U_{f,S}$ について、式 (4.1) が成立すればよい（定義 4.1）。よって、

$$\begin{aligned}
 & U_{f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\
 &= \left(U_{\text{mov},f,S'} \circ U_{\text{nei},f,S'} \right)(T_s(\mathbf{x}^i); \chi_{S'}) \\
 &:= \mathbf{x}^{i'} \\
 &= \mathbf{x}^i - \mathbf{t} \\
 &= \left(T_s \circ U_{\text{mov},f,S} \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i; \chi_S) \\
 &= \left(T_s \circ U_{f,S} \right)(\mathbf{x}^i; \chi_S)
 \end{aligned}$$

なお、これは $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ としても成立する。よって、GS が解空間の平行移動不変性を有していることを示した。

また、LS の近傍生成則 $U_{\text{nei},f,S}$ による $\hat{\mathbf{x}}^i$ は下記のように求められる。

$$\begin{aligned}
 \hat{\mathbf{x}}^{i'} &= U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\
 &= \mathbf{x}^{i'} + r(\mathbf{x}^{r_1'} - \mathbf{x}^{r_2'}) \\
 &= \mathbf{x}^i - \mathbf{t} + r((\mathbf{x}^{r_1} - \mathbf{t}) - (\mathbf{x}^{r_2} - \mathbf{t})) \\
 &= (\mathbf{x}^i + r(\mathbf{x}^{r_1} - \mathbf{x}^{r_2})) - \mathbf{t} \\
 &= \hat{\mathbf{x}}^i - \mathbf{t} \\
 &= \left(T_s \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i; \chi_S)
 \end{aligned} \tag{4.7}$$

さらに、解空間の平行移動 T_s のもとで、探索点の移動則 $U_{\text{mov},f,S}$ については GS と同様である。ここで、題意が成り立つには、LS の更新則 $U_{f,S}$ について、式 (4.1) が成立すればよい（定義 4.1）が、GS と全く同様に求められる。よって、LS が解空間の平行移動不変性を有していることを示した。以上の証明により、題意が示された。□

定理 4.16 (CS の解空間の回転不変性に関する定理) CS の更新則 $U_{f,S}$ は解空間の回転不変性に対する不変性を有さない。ただし、回転変換は $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x}$, \mathbf{C} は正規直交行列 ($\mathbf{C}^T = \mathbf{C}^{-1}$) である。

証明. 解空間の回転変換 T_s のもとで, 更新則 $U_{f,S}$ の引数の影響を考える。探索点の位置ベクトル \mathbf{x}^i ($i = 1, 2, \dots, m$) は $\mathbf{x}^i \in \chi_S$ だから, $\mathbf{x}^{i'} = \mathbf{C}\mathbf{x}^i$ となる。また, \mathbf{s} は安定分布に従う乱数ベクトルであるため, 回転しても影響を受けない。よって, $\mathbf{s}' = \mathbf{s}$ と表せる。

ここで, 題意が成り立つには, CS の GS (式(3.48)) と LS (式(3.50)) の更新則 $U_{f,S}$ のどちらかについて, 式(4.1) が成立しなければよい (定義 4.1)。まず, GS の近傍生成則 $U_{\text{nei},f,S}$ について,

$$\begin{aligned} & U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \mathbf{x}^{i'} + \alpha \mathbf{s}' \\ &= \mathbf{C}\mathbf{x}^i + \alpha \mathbf{s} \\ &\neq \mathbf{C}\mathbf{x}^i + \alpha \mathbf{C}\mathbf{s} \\ &= \mathbf{C}(\mathbf{x}^i + \alpha \mathbf{s}) \\ &= \mathbf{C}\hat{\mathbf{x}}^i \\ &= \left(T_s \circ U_{\text{nei},f,S}\right)(\mathbf{x}^i; \chi_S) \end{aligned}$$

となり, 式(4.1) が成立しない。このため, GS の全体の更新則についても式(4.1) が成立しない。よって, GS が解空間の回転不変性を有していないことを示した。

一方, LS の近傍生成則 $U_{\text{nei},f,S}$ について,

$$\begin{aligned} & U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \mathbf{x}^{i'} + r(\mathbf{x}^{r_1} - \mathbf{x}^{r_2}) \\ &= \mathbf{C}\mathbf{x}^i + r(\mathbf{C}\mathbf{x}^{r_1} - \mathbf{C}\mathbf{x}^{r_2}) \\ &= \mathbf{C}\mathbf{x}^i + r\mathbf{C}(\mathbf{x}^{r_1} - \mathbf{x}^{r_2}) \\ &= \mathbf{C}(\mathbf{x}^i + r(\mathbf{x}^{r_1} - \mathbf{x}^{r_2})) \\ &= \mathbf{C}\hat{\mathbf{x}}^i \\ &= \left(T_s \circ U_{\text{nei},f,S}\right)(\mathbf{x}^i; \chi_S) \end{aligned}$$

となる。

さらに, 解空間の回転変換 T_s のもとで, 探索点の移動則 $U_{\text{mov},f,S}$ について考える。下記では, $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定し, 解空間の回転変換 T_s の影響を受けないとする。

LS の探索点の移動則 $U_{\text{mov},f,S}$ による \mathbf{x}^i は下記のように求められる。

$$\begin{aligned}\mathbf{x}^{i\prime}(k+1) &= U_{\text{mov},f,S'}(T_s(\hat{\mathbf{x}}^i); \chi_{S'}) \\ &= \hat{\mathbf{x}}^{i\prime} \\ &= \mathbf{C}\hat{\mathbf{x}}^i \\ &:= \mathbf{C}\mathbf{x}^i \\ &= (T_s \circ U_{\text{mov},f,S})(\mathbf{x}^i; \chi_S)\end{aligned}$$

以上から、LS の更新則 $U_{f,S}$ については、

$$\begin{aligned}&U_{f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= (U_{\text{mov},f,S'} \circ U_{\text{nei},f,S'})(T_s(\mathbf{x}^i); \chi_{S'}) \\ &:= \mathbf{x}^{i\prime} \\ &= \mathbf{C}\mathbf{x}^i \\ &= (T_s \circ U_{\text{mov},f,S} \circ U_{\text{nei},f,S})(\mathbf{x}^i; \chi_S) \\ &= (T_s \circ U_{f,S})(\mathbf{x}^i; \chi_S)\end{aligned}$$

となり、式(4.1)が成立する。なお、これは $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ としても成立する。よって、LS が解空間の回転不変性を有していることを示した。以上の証明により、題意が示された。

□

定理 4.17 (CS の解空間の相似変換不変性・スケール不変性に関する定理) CS の更新則 U_f は解空間の相似変換不変性・スケール不変性を有さない。ただし、相似変換は $T_s : \mathbf{x} \mapsto \delta\mathbf{x}$, $\delta > 0$ である。

証明. 解空間の相似変換 T_s のもとで、更新則 $U_{f,S}$ の引数の影響を考える。探索点の位置ベクトル \mathbf{x}^i ($i = 1, 2, \dots, m$) は $\mathbf{x}^i \in \chi_S$ だから、 $\mathbf{x}^{i\prime} = \mathbf{C}\delta\mathbf{x}^i$ となる。また、 s は安定分布に従う乱数ベクトルであるため、相似変換しても影響を受けない。よって、 $s' = s$ と表せる。

ここで、題意が成り立つには、CS の GS (式(3.48)) と LS (式(3.50)) の更新則 $U_{f,S}$ のどちらかについて、式(4.1)が成立しなければよい (定義 4.1)。まず、GS の近傍生成則

$U_{\text{nei},f,S}$ について、

$$\begin{aligned}
 & U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\
 &= \mathbf{x}^{i'} + \alpha s' \\
 &= \delta \mathbf{x}^i + \alpha s \\
 &\neq \delta \mathbf{x}^i + \alpha \delta s \\
 &= \delta(\mathbf{x}^i + \alpha s) \\
 &= \delta \hat{\mathbf{x}}^i \\
 &= \left(T_s \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i; \chi_S)
 \end{aligned}$$

となり、式(4.1)が成立しない。このため、GSの全体の更新則についても式(4.1)が成立しない。よって、GSが解空間の相似変換不变性を有していないことを示した。

一方、LSの近傍生成則 $U_{\text{nei},f,S}$ について、

$$\begin{aligned}
 & U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\
 &= \mathbf{x}^{i'} + r(\mathbf{x}^{r_1'} - \mathbf{x}^{r_2'}) \\
 &= \delta \mathbf{x}^i + r(\delta \mathbf{x}^{r_1} - \delta \mathbf{x}^{r_2}) \\
 &= \delta \mathbf{x}^i + r \delta(\mathbf{x}^{r_1} - \mathbf{x}^{r_2}) \\
 &= \delta(\mathbf{x}^i + r(\mathbf{x}^{r_1} - \mathbf{x}^{r_2})) \\
 &= \delta \hat{\mathbf{x}}^i \\
 &= \left(T_s \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i; \chi_S)
 \end{aligned}$$

となる。

さらに、解空間の相似変換 T_s のもとで、探索点の移動則 $U_{\text{mov},f,S}$ について考える。下記では、 $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定し、解空間の相似変換 T_s の影響を受けないとする。

LSの探索点の移動則 $U_{\text{mov},f,S}$ による \mathbf{x}^i は下記のように求められる。

$$\begin{aligned}
 \mathbf{x}^{i'}(k+1) &= U_{\text{mov},f,S'}(T_s(\hat{\mathbf{x}}^i); \chi_{S'}) \\
 &= \hat{\mathbf{x}}^{i'} \\
 &= \delta \hat{\mathbf{x}}^i \\
 &:= \delta \mathbf{x}^i \\
 &= \left(T_s \circ U_{\text{mov},f,S} \right)(\mathbf{x}^i; \chi_S)
 \end{aligned}$$

以上から, LS の更新則 $U_{f,S}$ については,

$$\begin{aligned} & U_{f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \left(U_{\text{mov},f,S'} \circ U_{\text{nei},f,S'} \right) (T_s(\mathbf{x}^i); \chi_{S'}) \\ &:= \mathbf{x}^{i\prime} \\ &= \delta \mathbf{x}^i \\ &= \left(T_s \circ U_{\text{mov},f,S} \circ U_{\text{nei},f,S} \right) (\mathbf{x}^i; \chi_S) \\ &= \left(T_s \circ U_{f,S} \right) (\mathbf{x}^i; \chi_S) \end{aligned}$$

となり, 式 (4.1) が成立する。なお, これは $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ としても成立する。よって, LS が解空間の相似変換不変性を有していることを示した。以上の証明により, CS が解空間の相似変換不変性を有していないことを示した。

さらに, 相似変換は, スケール変換 $\mathbf{x} \mapsto \mathbf{C}\mathbf{x}$ の対角行列 \mathbf{C} が $\mathbf{C} = \delta \mathbf{I}$ とした場合である。このため, 相似変換不変性を持たない場合, スケール不変性も持たない。以上の証明により, 題意が示された。□

定理 4.18 (CS の目的関数の単調増加変換不変性に関する定理) CS の更新則 $U_{f,S}$ は目的関数の単調増加変換不変性を有する。ただし, 目的関数の単調増加変換は $T_f : f(\mathbf{x}) \mapsto g \circ f(\mathbf{x})$, g は単調増加変換である。

証明. 目的関数の単調増加変換 T_f のもとで, 更新則 $U_{f,S}$ の引数の影響を考える。 $f'(\mathbf{x}^i) = g \circ f(\mathbf{x}^i)$, $f'(\hat{\mathbf{x}}^i) = g \circ f(\hat{\mathbf{x}}^i)$ となる。下記では, $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定する。このとき,

$$\begin{aligned} f'(\mathbf{x}^i) - f'(\hat{\mathbf{x}}^i) &= g \circ f(\mathbf{x}^i) - g \circ f(\hat{\mathbf{x}}^i) \\ &= g \circ (f(\mathbf{x}^i) - f(\hat{\mathbf{x}}^i)) \geq 0 \end{aligned}$$

となるから, $f'(\hat{\mathbf{x}}^i) \leq f'(\mathbf{x}^i)$ が成り立つ。これは, T_f に対して目的関数值の大小関係が不变であることを表す。

CS の GS (式 (3.48)) と LS (式 (3.50)) において, T_f に対して $\hat{\mathbf{x}}^i$ は目的関数值を含まないため, $\hat{\mathbf{x}}^{i\prime} = \hat{\mathbf{x}}^i$ となる。式 (3.49) で, $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成り立つとき, $\mathbf{x}^i := \hat{\mathbf{x}}^i$ となる。また, T_f が与えられたとき, $f'(\hat{\mathbf{x}}^i) \leq f'(\mathbf{x}^i)$ が成り立つため, $\mathbf{x}^{i\prime} := \hat{\mathbf{x}}^{i\prime} = \hat{\mathbf{x}}^i$ となる。

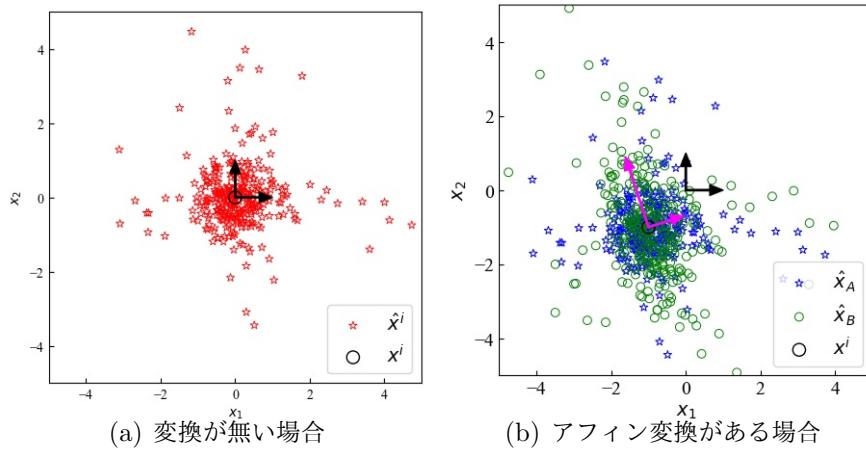


図 4.9：CS の Global Search における近傍生成

ここで、題意が成り立つには、CS の更新則 $U_{f,S}$ について、式 (4.1) が成立すればよい (定義 4.1)。よって、

$$\begin{aligned} & U_{T_f \circ f, S}(\mathbf{x}^i; \chi_S) \\ &= \mathbf{x}^{i\prime} \\ &= \mathbf{x}^i \\ &= U_{f,S}(\mathbf{x}^i; \chi_S) \end{aligned}$$

また、 $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ のときも上記の論証は成立する。以上の証明により、題意が示された。

□

(b) Cuckoo Search のアフィン変換不変性に関する数値実験的検証

数値実験的検証を通じて、CS のアフィン変換不変性の欠如を指摘する。具体的には、アフィン変換 $T_s : \mathbf{x}$ のもとで、 $\mathbf{x}_A \neq \mathbf{x}_B$ が成立することを示せばよい (定義 4.1)。ところで、アフィン変換は線形変換 $\mathbf{x} \mapsto \mathbf{Cx}$ と平行移動 $\mathbf{x} \mapsto \mathbf{x} - \mathbf{t}$ に分解できる。本実験でアフィン変換を与える行列とベクトルは $\mathbf{C} = \mathbf{B}(\theta)\mathbf{D}^\varepsilon, \mathbf{t} = \delta\mathbf{1}$ とする⁵。なお、

⁵ スケール変換を与える場合は \mathbf{C} を対角行列とし、回転変換を与える場合は \mathbf{C} を正規直交行列とすればよい。本実験では、 \mathbf{t} は平行移動、 \mathbf{D} はスケール変換、 $\mathbf{B}(\theta)$ は回転変換を与えていている。

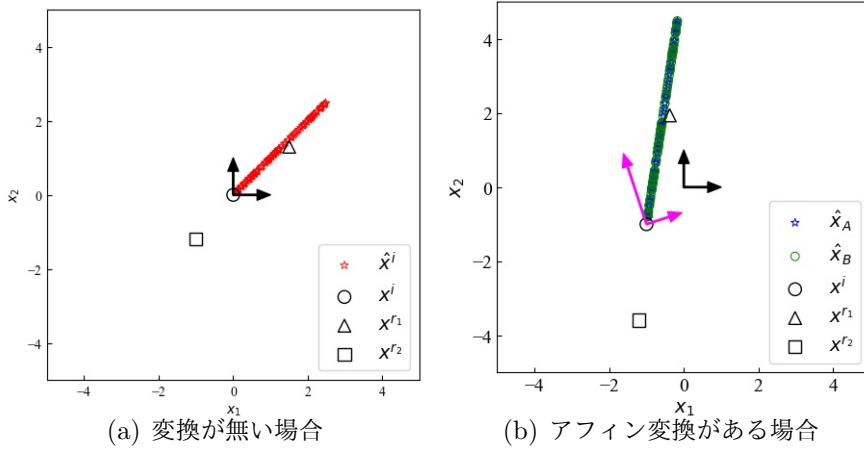


図 4.10：CS の Local Search における近傍生成

$D = \text{diag}(1, 2, \dots, N)$, $B(\theta)$ は解空間を θ deg 回転させる回転行列である。回転行列は付録 A を参照されたい。

図 4.9 に, \mathbb{R}^2 上の同じ探索点の位置から, CS の Global Search の更新則 $U_{f,S}$ により x_A と x_B を 300 回生成した図を示す。アフィン変換のパラメータは $\theta = 18, \delta = 1, \varepsilon = 1$, CS のパラメータは $\alpha = 0.75, \beta = 1.9$ とする。原点の \circlearrowleft は探索点 x^i , 黒い矢印は標準基底ベクトルを表す。図 4.9(a) は, 変換する前に生成した近傍解 \hat{x}^i , 図 4.9(b) は, 変換後に生成した近傍 x_A と, 生成後の解 (図 4.9(a) の近傍) を変換した近傍 x_B を示している。図 4.9 から, $x_A \neq x_B$ の成立が確認できる。また, Global Search の近傍は, 解空間のアフィン変換に関わらず, 標準基底に沿って生成されることも確認できる。同様に, 図 4.10 に, \mathbb{R}^2 上の同じ探索点の位置から, CS の Local Search の更新則 $U_{f,S}$ により x_A と x_B を 300 回生成した図を示す。パラメータと図中の記号は図 4.9 と同様であり, \triangle と \square は他の探索点 x^{r_1}, x^{r_2} を表す。図 4.9 から, $\hat{x}_A = \hat{x}_B$ の成立が確認できる。

Global Search の近傍が標準基底に依存し, アフィン変換不変性を失っている要因として, 下記の二つが考えられる。

- 乱数ベクトルが確率変数毎に独立に生成されること。回転不変性は, 乱数ベクトルが従う分布の等方性・異方性に依存する。安定分布はヘヴィテール型であり, 空間に於いて異方性となるため, 乱数ベクトルは回転不変性を欠如してしまう。
- 近傍生成で, 乱数ベクトルを単独の項として使用すること。乱数ベクトルによる摂

動量は、探索点の初期配置領域や探索過程における分布などに応じてスケーリングできないため、乱数ベクトルに基づく近傍もスケール不変性を欠如してしまう。

これらは、4.2節で述べた、乱数ベクトルの影響であることを表している。したがって、CSのGlobal Searchはアフィン変換不変性を具備せず、Local Searchはアフィン変換不変性を具備することが示された。よって、CSが完全にアフィン変換不変性を具備するには、Global Searchに不变性を付加する必要がある。

4.4 相関性を用いた回転不变性を有する Particle Swarm Optimization

4.4.1 先行研究と背景

これまでの節では、各種メタヒューリスティクスの変換不变性の有無について明らかにしてきたが、以降の節では、各アルゴリズムにおいてはまだ適用されていない既存のアプローチを流用することで、不变性の付加を検討する。

前節や先行研究[91]で指摘しているように、PSOは回転不变性を有しておらず、回転変換に対して性能がロバストでない。これは、BBOにおいて、目的関数の変数間依存性に対して不利であるが、アルゴリズムの適切な選択が困難となるため、ユーザに負担がかかるることを表している。そこで、先行研究では、PSOにおいて、①超球や超立方体を導入する[95, 98, 99]、②回転行列を作用する[91, 96]、③線型演算のみで構成する[90, 97, 100, 101, 102]ことで、回転不变性を付加している。これらは、4.2節で挙げた全てのアプローチについて事例がある。①超球や超立方体のアプローチは、Standard PSO-2011[95]のように単純に超球を導入しても多様性に欠けるため、他の先行研究では多様性を確保する工夫がされている[99]、超立方体は完全に回転不变性を付加しているわけではない[98]。③線型演算のアプローチは、Linear PSO[90]のように簡潔にアフィン変換不变性（回転不变性含む）を獲得できるが、多様性が失われることが指摘されている[97]ため、他の先行研究では多様性を確保する工夫がされている[90, 97, 100, 101, 102]。一方、②回転行列のアプローチは、上記の多様性の犠牲を最小限に抑えながら、回転不变

性を付加することが可能である。[91, 96]の方法は、ランダムな回転角を生成し、その回転角を回転させる行列を差分ベクトルに作用させるが、有効な回転角度を与えていとは言い難い。

以上の背景を踏まえ、本節では、共分散行列の固有ベクトルに基づく基底変換行列を生成し、それを差分ベクトルに作用することで、回転不変性を獲得するPSOを提案する。探索過程で得られる解集団の共分散行列の固有ベクトルを用いることで、探索にとってより有効な回転変換を与えることが期待できる。PSO以外の先行研究では、ESや、実数値GAの交叉、DEの交叉、ABCに対して、本章と全く同様のアプローチにより、回転不変性を獲得している[88, 93, 189, 105, 106, 109, 110, 193]が、PSOに対してこのアプローチが適用された先行研究は、著者の知る限りない。

本節の構成を述べる。4.4.2節では、提案手法について述べる。4.4.3節では、提案手法の回転不変性の具備を示す。4.4.4節では、数値実験により、提案手法の回転変換に対するロバスト性を示す。

4.4.2 相関性を用いた回転不変性を有するPSO

本項では、相関性を用いた回転不変性を有するPSO (PSO with Rotational Invariance Using Correlativity : CRI-PSO) を提案する。CRI-PSOの更新式は、PSOの更新則を式(3.23)から式(4.8)に変更したものである。

$$\begin{aligned} \mathbf{v}^i(k+1) = & w\mathbf{v}^i(k) + c_1 \mathbf{P} \mathbf{R}_1 \mathbf{P}^T (\mathbf{p}^i(k) - \mathbf{x}^i(k)) \\ & + c_2 \mathbf{P} \mathbf{R}_2 \mathbf{P}^T (\mathbf{p}^g(k) - \mathbf{x}^i(k)) \end{aligned} \quad (4.8)$$

$\mathbf{P} \in \mathbb{R}^{N \times N}$ は基底変換行列であり、式(4.9)で表される。

$$\mathbf{P} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N] \quad (4.9)$$

他の更新則はPSOと同様で、探索点の更新は式(3.24)、p-bestの更新は式(3.25)、g-bestの更新は式(2.16)を用いる。

CRI-PSOでは、標準基底 $E_0 = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N\}$ の座標系から新たな正規直交基底 $E = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N\}$ の座標系へ変換し、近傍解を生成した後、その解を元の標準基底 E_0

の座標系へ変換する。正規直交基底 E は、解集団 Z の共分散行列 $\Sigma(\mathcal{Z})$ の正規固有ベクトル（主成分ベクトル）で構成される。固有ベクトルは共分散行列 $\Sigma(\mathcal{Z})$ の固有値分解を行うことで得られる。特異値分解については付録 A を参照されたい。PSO は探索の各反復回数で同一の座標軸（標準基底 E_0 ）に沿った近傍生成を行うため、回転不変性を有さない。一方で、CRI-PSO は新たな正規直交基底 E への座標変換によって、探索の各反復回数で異なる座標軸に沿った近傍生成を行うため、回転不変性を有する。

定義 3.1 に従うと、CRI-PSO の更新則 $U_{f,S}$ として、式(4.8) と式(3.24) を連立させることで、式(4.10) が得られる。

$$\begin{aligned} \mathbf{x}^i(k+1) = & \mathbf{x}^i(k) + w\mathbf{v}^i(k) + c_1 \mathbf{P} \mathbf{R}_1 \mathbf{P}^T (\mathbf{p}^i(k) - \mathbf{x}^i(k)) \\ & + c_2 \mathbf{P} \mathbf{R}_2 \mathbf{P}^T (\mathbf{p}^g(k) - \mathbf{x}^i(k)) \end{aligned} \quad (4.10)$$

Algorithm 4.4.2 に CRI-PSO のアルゴリズムを示す。

4.4.3 CRI-PSO の変換不変性の解析

(a) CRI-PSO の変換不変性の証明

CRI-PSO に関して、定理 4.19, 定理 4.20, 定理 4.21 を示すことで、解空間の相似変換不変性・回転不変性・平行移動不変性・目的関数の単調増加変換不変性を有することを明らかにする。ただし、解空間のスケール不変性を失っていることも明らかにする。

定理 4.19 (CRI-PSO の解空間の回転不変性・相似変換不変性・平行移動不変性に関する定理) CRI-PSO の更新則 $U_{f,S}$ は解空間の回転不変性・相似変換不変性・平行移動不変性を有する。ただし、回転変換、相似変換、平行移動の合成変換は $T_s : \mathbf{x} \mapsto \delta \mathbf{C} \mathbf{x} - \mathbf{t}$, $\forall \delta > 0, \mathbf{t} \in \mathbb{R}^N$ であり、 $\mathbf{C} \in \mathbb{R}^{N \times N}$ は正規直交行列 ($\mathbf{C}^T = \mathbf{C}^{-1}$) である。

証明. 解空間の回転変換・相似変換・平行移動の合成変換 T_s のもとで、更新則 $U_{f,S}$ の引数の影響を考える。探索点の位置ベクトル $\mathbf{x}^i(k)$ ($i = 1, \dots, m$), p-best $\mathbf{p}^i(k)$, g-best $\mathbf{p}^g(k)$ は、式(3.25), 式(2.16) より、 $\mathbf{x}^i(k), \mathbf{p}^i(k), \mathbf{p}^g(k) \in \chi_S(k)$ だから、 $\mathbf{x}^{i'}(k) = \delta \mathbf{C} \mathbf{x}^i(k) - \mathbf{t}$, $\mathbf{p}^{i'}(k) = \delta \mathbf{C} \mathbf{p}^i(k) - \mathbf{t}$, $\mathbf{p}^{g'}(k) = \delta \mathbf{C} \mathbf{p}^g(k) - \mathbf{t}$ となる。探索点の移動ベクトル $\mathbf{v}^i(k+1)$ は

Algorithm 11 CRI-PSO

```

1: procedure CRI-PSO
2:   Step 0: パラメータ設定
3:   パラメータ  $m, k_{\max}, w, c_1, c_2$  を設定
4:   Step 1: 初期化
5:    $\mathbf{x}^i(k)$  を  $S_0$  内にランダムに配置する。
6:    $\mathbf{v}^i(k)$  をランダムに与える。
7:   p-best を  $\mathbf{p}^i(k) = \mathbf{x}^i(k)$  とする。
8:   g-best を  $\mathbf{p}^g(k)$  を式 (2.16) を用いて計算する。
9:   Step 2: 座標変換行列の計算
10:  共分散行列  $\Sigma(\mathcal{Z})$  を固有値分解する。
11:   $\mathbf{P}$  を式 (4.9) を用いて計算する。
12:  Step 3: 速度と位置の更新
13:   $\mathbf{v}^i(k)$  を式 (4.8) を用いて更新する。
14:   $\mathbf{x}^i(k)$  を式 (3.24) を用いて更新する。
15:   $\mathbf{p}^g(k)$  を式 (3.25) を用いて更新する。
16:  if  $k \geq k_{\max}$  then
17:    計算を終了する。
18:  else
19:     $k := k + 1$  とし、Step 2 へ戻る。
20:  end if
21: end procedure

```

式(3.24)より、 $v^{i\prime}(k+1)$ は

$$\begin{aligned}
\mathbf{v}^{i\prime}(k+1) &= \mathbf{x}^{i\prime}(k+1) - \mathbf{x}^{i\prime}(k) \\
&= (\mathbf{C}\mathbf{x}^i(k+1) - \mathbf{t}) - (\mathbf{C}\mathbf{x}^i(k) - \mathbf{t}) \\
&= \mathbf{C}(\mathbf{x}^i(k+1) - \mathbf{x}^i(k)) \\
&= \mathbf{C}\mathbf{v}^i(k+1)
\end{aligned}$$

となるから、 $\mathbf{v}^{i\prime}(k) = \mathbf{C}\mathbf{v}^i(k)$ と表される。

さらに、式(4.9)の基底変換行列 P について式(4.12)が成立する（補助定理 A.1）。

$$P' = CP \quad (4.11)$$

ここで、題意が成り立つには、CRI-PSO の更新則 $U_{f,S}$ について、式(4.1)が成立すれば

よい（定義 4.1）。よって、

$$\begin{aligned}
& U_{f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\
&= \mathbf{x}^{i'}(k) + w\mathbf{v}^{i'}(k) + c_1 \mathbf{P}' \mathbf{R}_1 \mathbf{P}'^T (\mathbf{p}^{i'}(k) - \mathbf{x}^{i'}(k)) + c_2 \mathbf{P}' \mathbf{R}_2 \mathbf{P}'^T (\mathbf{p}^{g'}(k) - \mathbf{x}^{i'}(k)) \\
&= (\delta \mathbf{C} \mathbf{x}^i(k) - \mathbf{t}) + w\delta \mathbf{C} \mathbf{v}^i(k) + c_1 (\mathbf{C} \mathbf{P}) \mathbf{R}_1 (\mathbf{C} \mathbf{P})^T ((\delta \mathbf{C} \mathbf{p}^i(k) - \mathbf{t}) - (\delta \mathbf{C} \mathbf{x}^i(k) - \mathbf{t})) \\
&\quad + c_2 (\mathbf{C} \mathbf{P}) \mathbf{R}_2 (\mathbf{C} \mathbf{P})^T ((\delta \mathbf{C} \mathbf{p}^g(k) - \mathbf{t}) - (\delta \mathbf{C} \mathbf{x}^i(k) - \mathbf{t})) \\
&= \delta (\mathbf{C} \mathbf{x}^i(k) + w \mathbf{C} \mathbf{v}^i(k) + c_1 \mathbf{C} \mathbf{P} \mathbf{R}_1 \mathbf{P}^T \mathbf{C}^T \mathbf{C} (\mathbf{p}^i(k) - \mathbf{x}^i(k)) \\
&\quad + c_2 \mathbf{C} \mathbf{P} \mathbf{R}_2 \mathbf{P}^T \mathbf{C}^T \mathbf{C} (\mathbf{p}^g(k) - \mathbf{x}^i(k))) - \mathbf{t} \\
&= \delta \mathbf{C} (\mathbf{x}^i(k) + w \mathbf{v}^i(k) + c_1 \mathbf{P} \mathbf{R}_1 \mathbf{P}^T (\mathbf{p}^i(k) - \mathbf{x}^i(k)) + c_2 \mathbf{P} \mathbf{R}_2 \mathbf{P}^T (\mathbf{p}^g(k) - \mathbf{x}^i(k))) - \mathbf{t} \\
&= \delta \mathbf{C} \mathbf{x}^i(k+1) - \mathbf{t} \\
&= (T_s \circ U_{f,S})(\mathbf{x}^i(k); \chi_S(k))
\end{aligned}$$

となり、式(4.1)を満たす。したがって、題意が成り立つことが示された。□

なお、証明の過程で用いた補助定理（補助定理 A.1）は付録 A に記している。

定理 4.20 (CRI-PSO の解空間のスケール不変性に関する定理) CRI-PSO の更新則 $U_{f,S}$ は解空間のスケール不変性を有さない。ただし、スケール変換は $T_s : \mathbf{x} \mapsto \mathbf{C} \mathbf{x}$, $\mathbf{C} \in \mathbb{R}^{N \times N}$ は対角行列である。

証明. 解空間のスケール変換 T_s のもとで、更新則 $U_{f,S}$ の引数の影響を考える。探索点の位置ベクトル $\mathbf{x}^i(k)$ ($i = 1, \dots, m$), p-best $\mathbf{p}^i(k)$, g-best $\mathbf{p}^g(k)$ は、式(3.25), 式(2.16)より, $\mathbf{x}^i(k), \mathbf{p}^i(k), \mathbf{p}^g(k) \in \chi_S(k)$ だから、 $\mathbf{x}^{i'}(k) = \mathbf{C} \mathbf{x}^i(k)$, $\mathbf{p}^{i'}(k) = \mathbf{C} \mathbf{p}^i(k)$, $\mathbf{p}^{g'}(k) = \mathbf{C} \mathbf{p}^g(k)$ となる。探索点の移動ベクトル $\mathbf{v}^i(k+1)$ は式(3.24)より、 $\mathbf{v}^{i'}(k+1)$ は

$$\begin{aligned}
\mathbf{v}^{i'}(k+1) &= \mathbf{x}^{i'}(k+1) - \mathbf{x}^{i'}(k) \\
&= \mathbf{C} \mathbf{x}^i(k+1) - \mathbf{C} \mathbf{x}^i(k) \\
&= \mathbf{C} (\mathbf{x}^i(k+1) - \mathbf{x}^i(k)) \\
&= \mathbf{C} \mathbf{v}^i(k+1)
\end{aligned}$$

となるから、 $\mathbf{v}^{i'}(k) = \mathbf{C} \mathbf{v}^i(k)$ と表される。

さらに、式(4.9)の基底変換行列 \mathbf{P} について、式(4.12)が成立する。

$$\mathbf{P}' = \mathbf{P} \quad (4.12)$$

これは、 \mathbf{P} は正規直交基底ベクトルで構成されているため、スケール変換の影響を受けないことを表している。

ここで、題意が成り立つには、CRI-PSOの更新則 $U_{f,S}$ について、式(4.1)が成立しなければよい(定義 4.1)。よって、

$$\begin{aligned} & U_{f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\ &= \mathbf{x}^{i\prime}(k) + w\mathbf{v}^{i\prime}(k) + c_1 \mathbf{P}' \mathbf{R}_1 \mathbf{P}'^T (\mathbf{p}^{i\prime}(k) - \mathbf{x}^{i\prime}(k)) + c_2 \mathbf{P}' \mathbf{R}_2 \mathbf{P}'^T (\mathbf{p}^{g\prime}(k) - \mathbf{x}^{i\prime}(k)) \\ &= \mathbf{C}\mathbf{x}^i(k) + w\mathbf{C}\mathbf{v}^i(k) + c_1 \mathbf{P}\mathbf{R}_1\mathbf{P}^T(\mathbf{C}\mathbf{p}^i(k) - \mathbf{C}\mathbf{x}^i(k)) + c_2 \mathbf{P}\mathbf{R}_2\mathbf{P}^T(\mathbf{C}\mathbf{p}^g(k) - \mathbf{C}\mathbf{x}^i(k)) \\ &= \mathbf{C}(\mathbf{x}^i(k) + w\mathbf{v}^i(k)) + c_1 \mathbf{P}\mathbf{R}_1\mathbf{P}^T\mathbf{C}(\mathbf{p}^i(k) - \mathbf{x}^i(k)) + c_2 \mathbf{P}\mathbf{R}_2\mathbf{P}^T\mathbf{C}(\mathbf{p}^g(k) - \mathbf{x}^i(k)) \\ &\neq \mathbf{C}(\mathbf{x}^i(k) + w\mathbf{v}^i(k)) + c_1 (\mathbf{C}\mathbf{P})\mathbf{R}_1\mathbf{P}^T\mathbf{C}^{-1}\mathbf{C}(\mathbf{p}^i(k) - \mathbf{x}^i(k)) \\ &\quad + c_2 (\mathbf{C}\mathbf{P})\mathbf{R}_2\mathbf{P}^T\mathbf{C}^{-1}\mathbf{C}(\mathbf{p}^g(k) - \mathbf{x}^i(k)) \\ &= \mathbf{C}\left(\mathbf{x}^i(k) + w\mathbf{v}^i(k) + c_1 \mathbf{P}\mathbf{R}_1\mathbf{P}^T(\mathbf{p}^i(k) - \mathbf{x}^i(k)) + c_2 \mathbf{P}\mathbf{R}_2\mathbf{P}^T(\mathbf{p}^g(k) - \mathbf{x}^i(k))\right) \\ &= \mathbf{C}\mathbf{x}^i(k+1) \\ &= (T_s \circ U_{f,S})(\mathbf{x}^i(k); \chi_S(k)) \end{aligned}$$

となり、式(4.1)を満たさない。したがって、題意が成り立つことが示された。 \square

定理 4.21 (CRI-PSO の目的関数の単調増加変換不変性に関する定理) CRI-PSO の更新則 $U_{f,S}$ は目的関数の単調増加変換不変性を有する。

証明. CRI-PSO の更新則は、PSO の更新則を式(3.23)から式(4.8)に変更したもので、 \mathbf{R} を $\mathbf{P}\mathbf{R}\mathbf{P}^T$ と置き換えただけである。この変更において目的関数値は無関係である。よって、PSO の定理 4.5 と全く同じ手順で式(4.1)を満たすことが導出できる。したがって、題意が成り立つことが示された。 \square

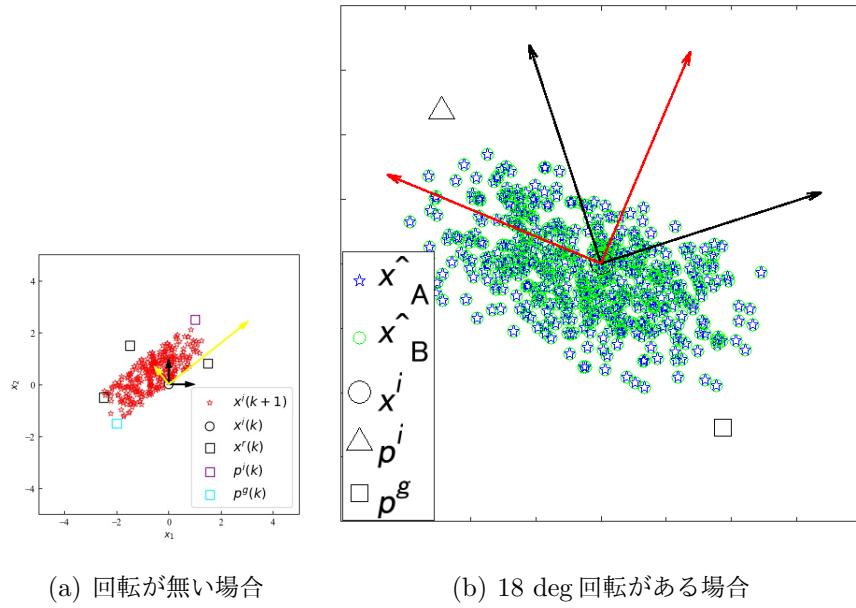


図 4.11：CRI-PSO の近傍

(b) 回転不变性に関する数値実験的検証

数値実験的検証を通じて CRI-PSO の回転不变性の有無を調べ、回転不变性の欠如を指摘する。図 4.11 に CRI-PSO の更新則で生成した \mathbf{x}_A , \mathbf{x}_B を示す。図 4.6 と同様の条件としており、赤い軸が新たな正規直交基底を成す固有ベクトル（主成分軸）である。図 4.11 から、 $\mathbf{x}_A = \mathbf{x}_B$ であることが確認できる。

PSO の近傍は、常に標準基底に沿って生成されるため、回転変換の前後で形状が全く異なる。一方で、CRI-PSO の近傍は、主成分軸に沿って生成するため、回転変換の前後で形状が不变となる。以上から、CRI-PSO は解空間の回転変換前後で探索性能がロバストになることが期待できる。

4.4.4 数値実験検証

(a) 共通の実験条件

典型的なベンチマーク関数を用いた数値実験を行い、CRI-PSO（提案手法）の探索性能を検証する。CRI-PSO と比較するアルゴリズムは、オリジナル PSO とする。各手法の共

通条件として、探索点数 $m = 20$ 、PSO のパラメータ $c_1 = c_2 = 1.4955$ 、関数毎に共通の初期配置領域 S_0 を用いる。また、各アルゴリズムのパラメータは、オリジナル PSO では $w = 0.729$ (CM [176, 177])、CRI-PSO では $w = 0.5$ を用いる。初期配置領域 S_0 内に初期探索点をランダムに配置した後、終了条件を満たすまで探索を進める試行を 50 回行い、50 試行の g-best の目的関数値 $f(\mathbf{p}^g(k_{\max}))$ の平均値を実験結果とする。数値実験で用いるベンチマーク関数については、付録 C を参照されたい。

なお、CRI-PSOにおいて、 \mathcal{Z} として任意の解集団を設定可能だが、各反復回数における探索点集団 $\mathcal{X}(k)$ と、各反復回数における p-best の集合 $\mathcal{P}(k)$ の二通り設定する。 $\mathcal{Z} = \mathcal{X}(k)$ の CRIPSO を「CRIPSO 1」、 $\mathcal{Z} = \mathcal{P}(k)$ の CRIPSO を「CRIPSO 2」と呼ぶことにする。

(b) アフィン変換に対するロバスト性に関する検証

数値実験を通じて、CRI-PSO のアフィン変換（回転変換を含む）に対する探索性能のロバスト性を検証する。本実験では、解空間にアフィン変換の一種である平行移動変換、スケール変換、回転変換を与えていく、その探索性能の変動を確認する。ベンチマーク関数として、変数分離可能・良スケール性・原点に最適解が存在・多峰性の 4 種類の目的関数を使用する。

各変換 T_s の詳細を下記に述べる。アフィン変換は線形変換 $\mathbf{x} \mapsto \mathbf{Cx}$ と平行移動 $\mathbf{x} \mapsto \mathbf{x} - \mathbf{t}$ に分解できる。本実験でアフィン変換を与える行列とベクトルは $\mathbf{C} = \mathbf{B}(\theta)\mathbf{D}^\varepsilon, \mathbf{t} = \delta\mathbf{1}$ とする⁶。なお、 $\mathbf{D} = \text{diag}(1, 2, \dots, N)$ 、 $\mathbf{B}(\theta)$ は解空間を θ deg 回転させる回転行列である。

実験 1 : $\theta = 0, \varepsilon = 0$ としたまま、 $\delta = 0, 1, 2, 3$ と変動させることで解空間を平行移動させる。解空間の原点依存性を発生させる変換である。

実験 2 : $\theta = 0, \delta = 0$ としたまま、 $\varepsilon = 0, 1, 2, 3$ と変動させることで解空間をスケール変換する。目的関数の悪スケール性を発生させる変換である。

実験 3 : $\delta = 0, \varepsilon = 0$ としたまま、 $\theta = 0, 30, 60, 90$ と変動させることで解空間を回転させる。目的関数の変数間依存性を発生させる変換である。なお、 $\theta = 90$ のときは、関数の対称性により、 $\theta = 0$ と同じ条件である。

なお、解空間の変換 T_s を与えるには、初期の探索点を $T_s^{-1}(\mathbf{x}^i(1)), f(\mathbf{x}(k))$ を $f(T_s(\mathbf{x}(k)))$

⁶ スケール変換を与える場合は \mathbf{C} を対角行列とし、回転変換を与える場合は \mathbf{C} を正規直交行列とすればよい。本実験では、 \mathbf{t} は平行移動、 \mathbf{D} はスケール変換、 $\mathbf{B}(\theta)$ は回転変換を与えていている。

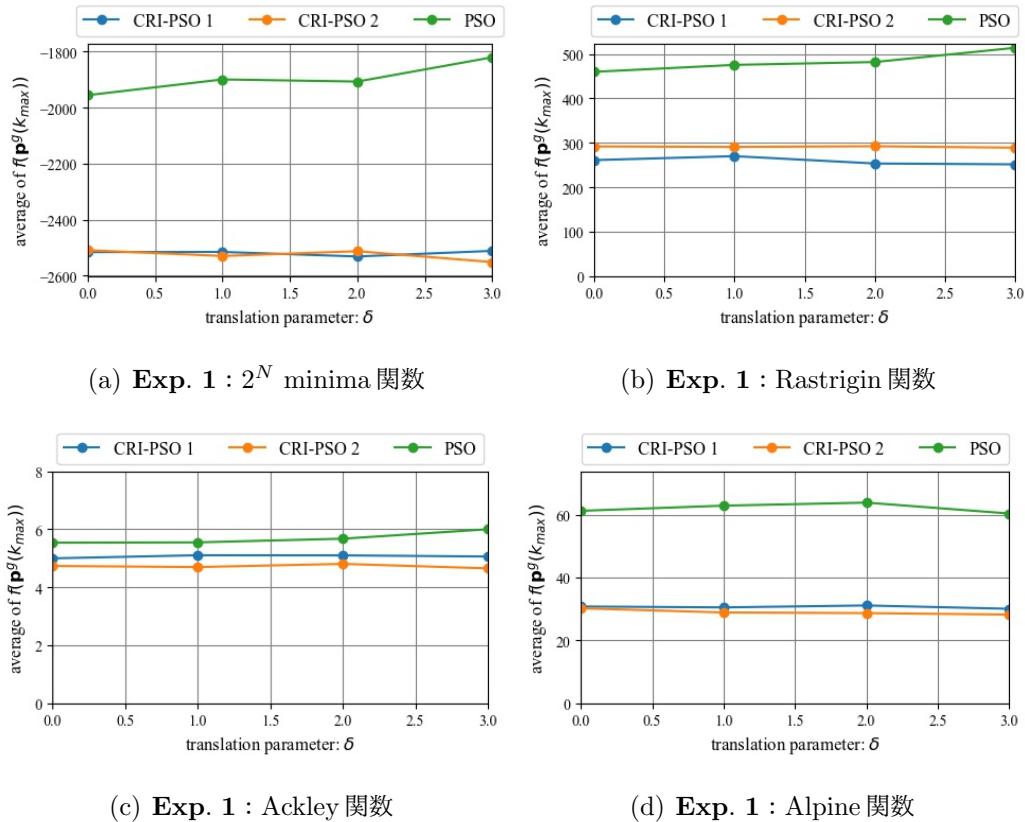


図 4.12：数値実験結果 (Exp. 1)：PSO と CRI-PSO の探索性能のロバスト性 ($N = 50$, $k_{\max} = 100$)

と置き換えればよい。共通条件として、次元数 $N = 50$ 、終了条件として評価回数 $k_{\max} = 100$ を用いる。

図 4.12, 図 4.13, 図 4.14 に数値実験結果を示す。各図の横軸は、平行移動パラメータ δ , スケール変換パラメータ ε , 回転角 θ であり、全ての図の縦軸が探索性能 ($f(\mathbf{p}^g(k_{\max}))$) の平均値である。図 4.12, 図 4.13, 図 4.14 より、下記の結果が確認できる。

- 解空間の平行移動変換に対して、全ての PSO は探索性能の変動が小さい。
- 解空間のスケール変換に対して、オリジナル PSO と CRI-PSO 2 は探索性能の変動が小さいことに対して、CRI-PSO 1 は、探索性能の変動が大きい。
- 解空間の回転変換に対して、オリジナル PSO は探索性能の変動が大きいことに対して、CRI-PSO は探索性能の変動が少ない。
- CRI-PSO 2 は、多くの条件で、探索性能が他の PSO よりも優れている。

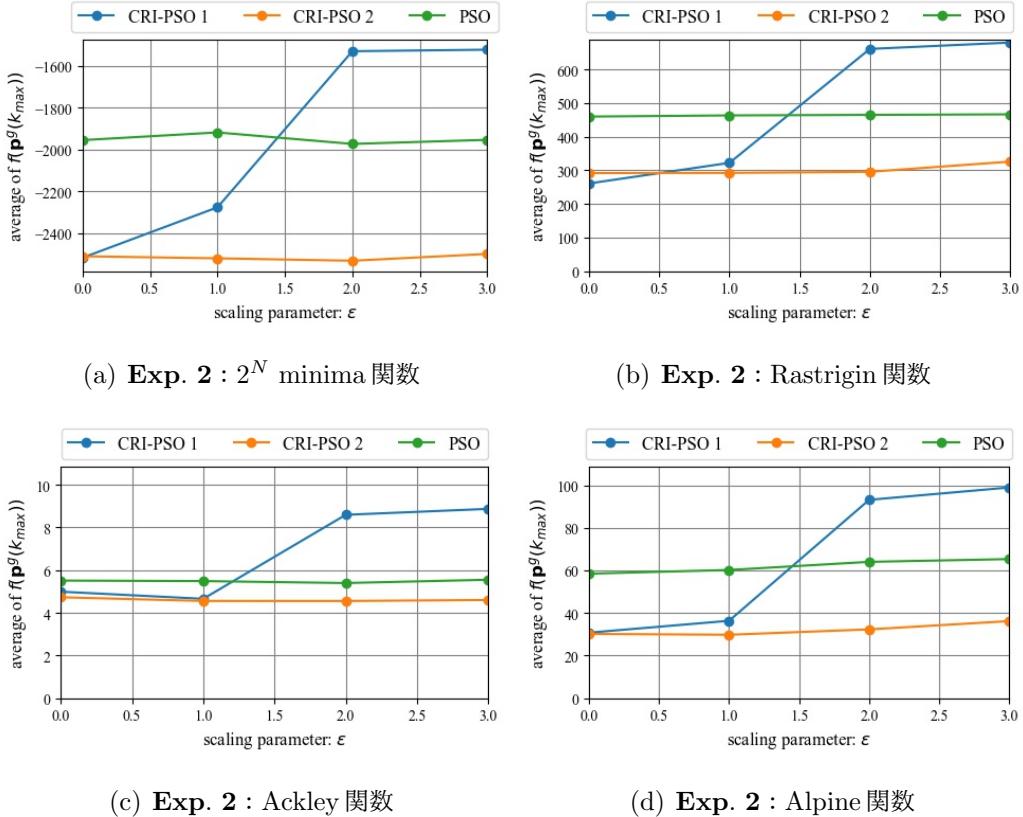


図 4.13 : 数値実験結果 (Exp. 2) : PSO と CRI-PSO の探索性能のロバスト性 ($N = 50$, $k_{\max} = 100$)

なお, $\theta = 90$ deg のときは, ベンチマーク関数の対称性により, $\theta = 0$ deg と同じ条件である。

図 4.14 の結果は, 回転不变性の有無が解空間の回転に対する探索性能のロバスト性に影響を与えることを示している。よって, PSO の探索性能は, 変数間依存性が発生すると悪化するため, 回転変換に対してロバストにならないことに対して, CRI-PSO の探索性能はロバストであることと, 多くの条件で PSO よりも優れていることも確認できる。一方, 図 4.13 の結果は, スケール不变性の有無が解空間のスケール変換に対する探索性能のロバスト性に影響を与えることを示している。しかしながら, スケール不变性を有していない, CRI-PSO 2 はスケール変換に対してもロバスト性を持つ。これは, $Z = \mathcal{P}(k)$ と設定することで, p-best の分布に沿った探索ができ, 性能が向上した結果, 悪スケール性にある程

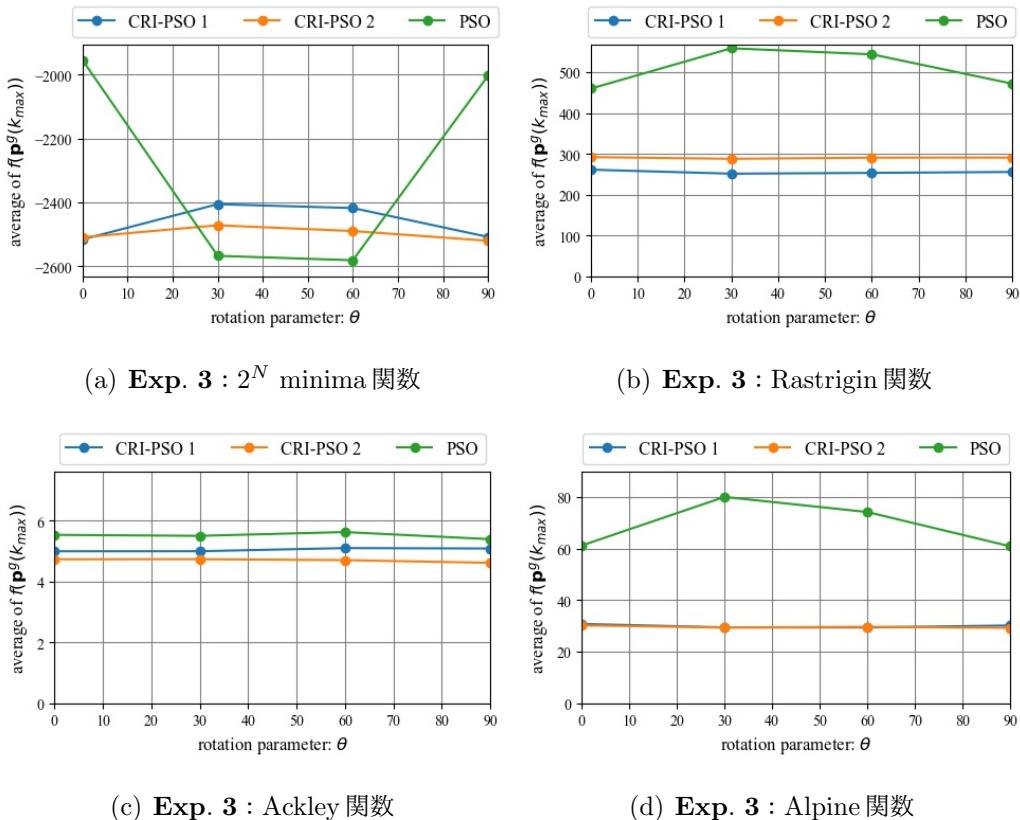


図 4.14 : 数値実験結果 (Exp. 3) : PSO と CRI-PSO の探索性能のロバスト性 ($N = 50$, $k_{\max} = 100$)

度対処できているためだと考えられる。

(c) 全般的な性能検証

数値実験を通じて, CRI-PSO の全般的な探索性能を検証する。数値実験には, 9種類のベンチマーク関数を用いる。共通条件として, 次元数 $N = 10, 50, 100, 300$, 終了条件を反復回数 $k_{\max} = 100, 1000$ とする。

数値実験結果を表 4.1 に示す。表 4.1 では, 三つの方法 (Original PSO, CRI-PSO 1, CRI-PSO 2) の結果を比較して, 最も良い結果に (*) を付しており, 二番目に良い結果

に（**）を付している⁷。

表4.1から、PSOの探索性能は、低次元、あるいは $k_{\text{kmax}} = 1000$ のとき、全般的に優れた性能を示すことが確認できる。これは、CMは低次元なら弱い安定領域内に位置し、持続的な探索が実行できるためだと思われる。一方で、CRI-PSOの探索性能は、高次元、あるいは $k_{\text{kmax}} = 100$ のとき、PSOよりも優れた性能を示すことが確認できる。これは、CRI-PSOでは、PSOとは安定・不安定領域の境界線が異なることが予想されることから、強い安定領域に位置したパラメータ設定であり、反復回数が短期に特化していたと考えられる。さらに、 $k_{\text{kmax}} = 100$ のときだけだが、関数2と関数5では、PSOに比べて顕著に改善していることが確認できる。これは、回転不変性を付加したことで、変数間依存性に対して効果を顕著に発揮したためだと思われる。さらに、全体的にCRI-PSO 2は、CRIPSO 1よりも探索性能が優れていることも確認できる。これは、PSOの性質として、 \varnothing を優れた解集団 $\mathcal{P}(k)$ と設定することで、より効率的な探索を行うことができるためだと考えられる。

これらの結果から、CRI-PSOは、パラメータを適切に設定すれば、回転不変性と共に分散行列の固有ベクトルの効果により、PSOよりも優れた性能を発揮するといえる。一方、パラメータが適切に設定できなければ、PSOよりも劣化するケースもみられる。これを解決するには、①CRI-PSOの安定・不安定領域の境界線の全体を把握し、弱い安定領域に設定することで多様化・集中化を十分に発揮できるようにする、②パラメータ調整則を付加することで、パラメータ設定の負荷を軽減すると同時に、探索状態に応じて多様なパラメータの効果を活用することで、多様化・集中化を発揮させる、が必要だと考えられる。この課題を受けて、第5章では、②のアプローチによって、CRI-PSOの適応性・平均的な性能の向上を図る。

⁷ なお、表4.1の一部では、同じ数値にも関わらず優劣がついている結果があるが、これは便宜上、小数点第三位以下を省略しているためであり、小数点以下の差異を確認した上で優劣を判断している。

表 4.1：全般的な探索性能に関する数値実験結果 (CRI-PSO)

関数	N	$k_{\max} = 100$			$k_{\max} = 1000$		
		PSO	CRI-PSO 1	CRI-PSO 2	PSO	CRI-PSO 1	CRI-PSO 2
1. Parabola	10	0.00 *	0.06	0.01	0.00 *	0.05	0.01
	50	43.2	49.7	41.8 *	0.01 *	49.9	40.8
	100	443	135	120 *	11.6 *	149	120
	300	2225	510	449 *	2221	516	437 *
2. Rosenbrock	10	8.60	18.6	8.35 *	1.31 *	10.7	10.2
	50	2236	805	606 *	99.6 *	966	624
	100	29456	2871	1970 *	1831 *	2627	2176
	300	111619	12399	9342 *	111776	12776	8829 *
3. 2^N minima	10	-723 *	-685	-682	-727 *	-675	-674
	50	-1955	-2516 *	-2509	-3344 *	-2519	-2533
	100	-2285	-4257 *	-4257	-2480	-4290 *	-4249
	300	-4931	-10266 *	-10197	-4898	-10146	-10313 *
4. Rastrigin	10	21.3	18.6	18.4 *	9.13 *	17.8	21.9
	50	460	261 *	292	186 *	256	291
	100	1364	718 *	766	592 *	733	752
	300	5029	2859	2809 *	5015	2850	2834 *
5. Schwefel	10	0.35	0.19	0.07 *	0.00 *	0.13	0.06
	50	297	85.2	70.7 *	34.1 *	84.1	69.2
	100	1469	379	314 *	363	353	324 *
	300	15518	3420	2878 *	4968	3584	2886 *
6. Levy	10	0.03 *	1.37	0.74	0.04 *	1.61	1.26
	50	40.5	13.8	12.7 *	3.83 *	12.7	13.4
	100	87.5	22.4	21.6 *	16.9 *	22.5	22.1
	300	143	36.8	33.5 *	136	36.1	32.8 *
7. Ackley	10	0.17 *	1.59	1.22	0.15 *	1.38	1.24
	50	5.53	4.99	4.73 *	3.13 *	4.96	4.68
	100	8.70	5.72	5.34 *	5.10 *	5.64	5.33
	300	10.1	6.23	5.92 *	10.1	6.17	5.91 *
8. Griewank	10	0.25	0.12	0.12 *	0.09 *	0.14	0.12
	50	2.16	2.25	2.00 *	0.10 *	2.22	2.00
	100	12.3	4.52	3.94 *	1.33 *	4.31	4.09
	300	56.6	13.2	11.7 *	56.6 *	13.7	12.1
9. Alpine	10	0.55 *	0.67	0.71	0.00 *	0.51	0.72
	50	61.2	30.8	30.2 *	2.91 *	29.4	28.3
	100	156	91.0	85.4 *	32.6 *	88.5	83.9
	300	669	361	328 *	618	359	322

4.5 超球に基づく回転不变性を有する Artificial Bee Colony Algorithm

4.5.1 先行研究と背景

4.3 節で指摘しているように、ABC は回転不变性と目的関数の単調増加変換不变性を有していない。後者は、ルーレット選択を実数値 GA のランキング選択に変更することで、容易に付加することが可能だが、前者の回転不变性は問題となる。これは、BBOにおいて、目的関数の変数間依存性に対して不利であるが、アルゴリズムの適切な選択が困難となるため、ユーザに負担がかかることを表している。

そこで、先行研究では、ABCにおいて、♠ 回転行列を作用する[194]、① 線型演算のみで構成する[191, 192]ことで、回転不变性を付加している。これらは、4.2 節で挙げた全てのアプローチのうち、「超球・超立方体の導入」以外のアプローチについて事例がある。♠ 回転行列のアプローチは、共分散行列の固有ベクトルに基づく基底変換行列を作用している[194]。① 線型演算のアプローチは、g-bestに向かう差分ベクトルの導入[191]、複数の次元で解の組合せを行う[191]、などがあるが、解の組合せの操作を排除していないため、回転不变性を完全に付加できているわけではない。しかしながら、解の組合せにおいて、係数がスカラである単なる線形結合にすると、多様性が失われるため、有用な探索を行うには他の工夫が必要となる。一方、ABCに対して、超球や超立方体のアプローチは、著者の知る限りなく、従来の近傍の多様性を比較的確保できると思われる。

以上の背景を踏まえ、本節では、超球表面上に生成される乱数ベクトルを導入した、回転不变性を獲得する ABC を提案する。ABC 以外の先行研究では、実数値 GA の交叉、PSO, DE の交叉に対して、本章と全く同様のアプローチにより、回転不变性を獲得している[122, 95, 99, 108, 190]が、ABC に対してこのアプローチが適用された先行研究は、著者の知る限りない。

本節の構成を述べる。4.5.2 節では、提案手法について述べる。4.5.3 節では、提案手法の回転不变性の具備を示す。4.5.4 節では、数値実験により、提案手法の回転変換に対するロバスト性を示す。

4.5.2 超球に基づく回転不变性を有するABC

ABCは式(3.40)のように解の組合せによって近傍生成を行うが、これは $\mathbf{x}^i, \mathbf{u}^i$ の線分を対角線とした超直方体の頂点に生成されるとみなすことができる。しかし、この超直方体は座標軸に平行であるため、回転不变性を欠如している。

そこで、本節では、超球に基づく回転不变性を有するABC (Hypersphere Based ABC with Rotational Invariance:HBC-ABC) を提案する。HBC-ABCは、ABCの更新式(3.40)から、式(4.13)に変更したものである。

$$\hat{\mathbf{x}}^i = (\mathbf{x}^i + \mathbf{u}^i)/2 + \|(\mathbf{x}^i - \mathbf{u}^i)/2\|_2 \mathbf{s}^i \quad (4.13)$$

ただし、 \mathbf{s}^i は原点中心の単位超球の表面上に一様に生成される乱数ベクトルであり、 $\mathbf{s}^i = \mathbf{z}/\|\mathbf{z}\|_2$ で生成される[194]。 \mathbf{z} は N 次元標準正規分布に従う乱数ベクトルである。式(4.13)以外は、ABCと同様の式を用いる。

式(4.13)は、中心が $(\mathbf{x}^i + \mathbf{u}^i)/2$ で、半径 $\|(\mathbf{x}^i - \mathbf{u}^i)/2\|_2$ の超球表面上に近傍解を生成することを表している。これは、式(3.40)の超直方体の解釈に着想を得ている。ABCの近傍生成がなす超直方体は、中心が $(\mathbf{x}^i + \mathbf{u}^i)/2$ で、対角線の長さ $\|\mathbf{x}^i - \mathbf{u}^i\|_2$ である。この超直方体は式(4.13)の超球に内接しているため、超直方体の頂点(ABCの近傍)は常に「超球表面上の一部」に生成されているとみなすことができる。式(4.13)は、「座標軸に平行である拘束」を解消し、近傍解をその超球表面上で一様に生成することに変更したものである。

Algorithm 4.5.2にHBC-ABCのアルゴリズムを示す。

4.5.3 HBC-ABCの変換不变性の解析

(a) HBC-ABCの変換不变性の証明

HBC-ABCに関して、定理 4.22, 定理 4.23, 定理 4.24を示すことで、解空間の相似変換不变性・回転不变性・平行移動不变性を有することを明らかにする。ただし、解空間のスケール不变性・目的関数の単調増加変換不变性を失っていることも明らかにする。なお、本節の目的は回転不变性の獲得であると同時に、目的関数の単調増加変換不变性はルー

Algorithm 12 HBC-ABC

```

1: procedure HBC-ABC
2:   Step 0: パラメータ設定
3:    $\mathbf{x}^i \in S_0 (i = 1, 2, \dots, m_e)$  をランダムに与える。
4:    $C_i := 0$  とする。
5:   for  $i = 1, 2, \dots, m_e$  do
6:     式(3.36)と式(4.13)を用いて、 $\mathbf{x}^i$  を更新する。
7:     式(3.37)を用いて、 $C_i$  を更新する。
8:   end for
9:   Step 3: 傍観蜂
10:  ルーレット選択により、探索点  $\mathbf{x}^i$  を選択する。
11:  式(3.36)と式(4.13)を用いて、 $\mathbf{x}^i$  を更新する。
12:  式(3.37)を用いて、 $C_i$  を更新する。
13: end for
14: for  $i = 1, 2, \dots, m_e$  do
15:   if  $s_i \geq Limit$  then
16:      $\mathbf{x}^i$  を  $S_0$  にランダムに再配置する。
17:    $C_i := 0$  とする。
18:   end if
19: end for
20: if  $T \geq T_{\max}$  then
21:   計算終了する。
22: else
23:   Step 2: 働き蜂に戻る。
24: end if
25: end procedure

```

レット選択をランキング選択に変更すれば、容易に獲得できるため、目的関数の単調増加変換不变性については詳細に考察しない。

定理 4.22 (HBC-ABC の解空間の相似変換不变性・回転不变性・平行移動不变性に関する定理) HBC-ABC の更新式 $U_{f,S}$ は解空間の相似変換不变性・回転不变性・平行移動不变性に対する不变性を有する。ただし、これらの合成変換は $T_s : \mathbf{x} \mapsto \delta \mathbf{C} \mathbf{x} - \mathbf{t}$, \mathbf{C} は正規直交行列 ($\mathbf{C}^T = \mathbf{C}^{-1}$), $\delta > 0$ である。

証明. 解空間の相似変換不变性・回転不变性・平行移動 T_s のもとで、近傍生成則 $U_{\text{nei},f,S}$ の

引数の影響を考える。探索点の位置ベクトル \mathbf{x}^j ($j = i, r$) は $\mathbf{x}^j \in \chi_S$ だから, $\mathbf{x}^{j\prime} = \mathbf{C}\mathbf{x}^j - \mathbf{t}$ となる。このとき, 式(3.42)の \mathbf{u}^i は

$$\begin{aligned}\mathbf{u}^{i\prime} &= \mathbf{x}^{i\prime} + \phi(\mathbf{x}^{r\prime} - \mathbf{x}^{i\prime}) \\ &= \delta\mathbf{C}\mathbf{x}^i - \mathbf{t} + \phi((\delta\mathbf{C}\mathbf{x}^r - \mathbf{t}) - (\delta\mathbf{C}\mathbf{x}^i - \mathbf{t})) \\ &= \delta\mathbf{C}\mathbf{x}^i + \phi\delta\mathbf{C}(\mathbf{x}^r - \mathbf{x}^i) - \mathbf{t} \\ &= \delta\mathbf{C}(\mathbf{x}^i + \phi(\mathbf{x}^r - \mathbf{x}^i)) - \mathbf{t} \\ &= \delta\mathbf{C}\mathbf{u}^i - \mathbf{t}\end{aligned}$$

となるから, $\mathbf{u}^{i\prime} = \delta\mathbf{C}\mathbf{u}^i - \mathbf{t}$ と表される。

また, \mathbf{z}^i は標準正規乱数ベクトルだから, $\mathbf{z}^{i\prime} = \mathbf{z}^i = \mathbf{C}\mathbf{z}^i$ となる。このため, \mathbf{s}^i は

$$\begin{aligned}\mathbf{s}^{i\prime} &= \frac{\mathbf{z}^{i\prime}}{\|\mathbf{z}^{i\prime}\|_2} \\ &= \frac{\mathbf{z}^i}{\|\mathbf{z}^i\|_2} \\ &= \mathbf{s}^i\end{aligned}$$

となるから, $\mathbf{s}^{i\prime} = \mathbf{s}^i = \mathbf{C}\mathbf{s}^i$ と表される。

ここで, HBC-ABC の近傍生成則 $U_{\text{nei},f,S}$ による $\hat{\mathbf{x}}^i$ は下記のように求められる。

$$\begin{aligned}\hat{\mathbf{x}}^{i\prime} &= U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \frac{1}{2}(\mathbf{x}^{i\prime} + \mathbf{u}^{i\prime}) + \frac{1}{2}\|\mathbf{x}^{i\prime} - \mathbf{u}^{i\prime}\|_2\mathbf{s}^{i\prime} \\ &= \frac{1}{2}((\delta\mathbf{C}\mathbf{x}^i - \mathbf{t}) + (\delta\mathbf{C}\mathbf{u}^i - \mathbf{t})) + \frac{1}{2}\|(\delta\mathbf{C}\mathbf{x}^i - \mathbf{t}) - (\delta\mathbf{C}\mathbf{u}^i - \mathbf{t})\|_2\mathbf{C}\mathbf{s}^i \\ &= \frac{1}{2}(\delta\mathbf{C}(\mathbf{x}^i + \mathbf{u}^i) - 2\mathbf{t}) + \frac{1}{2}\|\delta\mathbf{C}(\mathbf{x}^i - \mathbf{u}^i)\|_2\mathbf{C}\mathbf{s}^i \\ &= \frac{1}{2}\delta\mathbf{C}(\mathbf{x}^i + \mathbf{u}^i) + \frac{1}{2}\delta\|\mathbf{x}^i - \mathbf{u}^i\|_2\mathbf{C}\mathbf{s}^i - \mathbf{t} \\ &= \delta\mathbf{C}\left(\frac{1}{2}(\mathbf{x}^i + \mathbf{u}^i) + \frac{1}{2}\|\mathbf{x}^i - \mathbf{u}^i\|_2\mathbf{s}^i\right) - \mathbf{t} \\ &= \delta\mathbf{C}\hat{\mathbf{x}}^i - \mathbf{t}\end{aligned}$$

直交行列の性質から, $\|\mathbf{C}(\mathbf{x}^i - \mathbf{u}^i)\|_2 = \|\mathbf{x}^i - \mathbf{u}^i\|_2$ であることを証明過程で利用した。

さらに、解空間の相似変換不変性・回転不変性・平行移動 T_s のもとで、探索点の移動則 $U_{\text{mov},f,S}$ の引数の影響を考える。式(3.32)において、 $\mathbf{x}^{i'} = \delta \mathbf{C} \mathbf{x}^i - \mathbf{t}$, $\hat{\mathbf{x}}^{i'} = \delta \mathbf{C} \hat{\mathbf{x}}^i - \mathbf{t}$ となる。また、下記では、 $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定し、解空間の相似変換不変性・回転不変性・平行移動 T_s の影響を受けないとする。HBC-ABC の探索点の移動則 $U_{\text{mov},f,S}$ による $\mathbf{x}^i(k+1)$ は下記のように求められる。

$$\begin{aligned}\mathbf{x}^{i'} &:= U_{\text{mov},f,S'}(T_s(\hat{\mathbf{x}}^i); \chi_{S'}) \\ &= \hat{\mathbf{x}}^{i'} \\ &= \delta \mathbf{C} \hat{\mathbf{x}}^i - \mathbf{t} \\ &= \delta \mathbf{C} \mathbf{x}^i - \mathbf{t}\end{aligned}$$

ここで、題意が成り立つには、HBC-ABC の更新則 $U_{f,S}$ について、式(4.1)が成立すればよい（定義 4.1）。よって、

$$\begin{aligned}U_{f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) &= \left(U_{\text{mov},f,S'} \circ U_{\text{nei},f,S'} \right)(T_s(\mathbf{x}^i); \chi_{S'}) \\ &:= \mathbf{x}^{i'} \\ &= \delta \mathbf{C} \mathbf{x}^i - \mathbf{t} \\ &= \left(T_s \circ U_{\text{mov},f,S} \circ U_{\text{nei},f,S} \right)(\mathbf{x}^i; \chi_S) \\ &= \left(T_s \circ U_{f,S} \right)(\mathbf{x}^i; \chi_S)\end{aligned}$$

なお、これは $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ としても成立する。以上の証明により、題意が示された。□

定理 4.23 (HBC-ABC の解空間のスケール不変性に関する定理) HBC-ABC の更新則 $U_{f,S}$ は解空間のスケール不変性を有さない。ただし、これらの合成変換は $T_s : \mathbf{x} \mapsto \mathbf{C} \mathbf{x}$, \mathbf{C} は対角行列である。

証明. 解空間のスケール変換 T_s のもとで、近傍生成則 $U_{\text{nei},f,S}$ の引数の影響を考える。探索点の位置ベクトル \mathbf{x}^j ($j = i, r$) は $\mathbf{x}^j \in \chi_S$ だから、 $\mathbf{x}^{j'} = \mathbf{C} \mathbf{x}^j$ となる。このとき、式

(3.42) の \mathbf{u}^i は

$$\begin{aligned}\mathbf{u}^{i\prime} &= \mathbf{x}^{i\prime} + \phi(\mathbf{x}^{r\prime} - \mathbf{x}^{i\prime}) \\ &= \mathbf{C}\mathbf{x}^i + \phi(\mathbf{C}\mathbf{x}^r - \mathbf{C}\mathbf{x}^i) \\ &= \mathbf{C}\mathbf{x}^i + \phi\mathbf{C}(\mathbf{x}^r - \mathbf{x}^i) \\ &= \mathbf{C}(\mathbf{x}^i + \phi(\mathbf{x}^r - \mathbf{x}^i)) \\ &= \mathbf{C}\mathbf{u}^i\end{aligned}$$

となるから、 $\mathbf{u}^{i\prime} = \mathbf{C}\mathbf{u}^i$ と表される。

また、 \mathbf{z}^i は標準正規乱数ベクトルだから、 $\mathbf{z}^{i\prime} = \mathbf{z}^i = \mathbf{C}\mathbf{z}^i$ となる。このため、 \mathbf{s}^i は

$$\begin{aligned}\mathbf{s}^{i\prime} &= \frac{\mathbf{z}^{i\prime}}{\|\mathbf{z}^{i\prime}\|_2} \\ &= \frac{\mathbf{z}^i}{\|\mathbf{z}^i\|_2} \\ &= \mathbf{s}^i\end{aligned}$$

となるから、 $\mathbf{s}^{i\prime} = \mathbf{s}^i = \mathbf{C}\mathbf{s}^i$ と表される。

ここで、HBC-ABC の近傍生成則 $U_{\text{nei},f,S}$ について、

$$\begin{aligned}&U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \frac{1}{2}(\mathbf{x}^{i\prime} + \mathbf{u}^{i\prime}) + \frac{1}{2}\|\mathbf{x}^{i\prime} - \mathbf{u}^{i\prime}\|_2 \mathbf{s}^{i\prime} \\ &= \frac{1}{2}(\mathbf{C}\mathbf{x}^i + \mathbf{C}\mathbf{u}^{i\prime}) + \frac{1}{2}\|\mathbf{C}\mathbf{x}^i - \mathbf{C}\mathbf{u}^{i\prime}\|_2 \mathbf{C}\mathbf{s}^i \\ &= \frac{1}{2}\mathbf{C}(\mathbf{x}^i + \mathbf{u}^{i\prime}) + \frac{1}{2}\|\mathbf{C}(\mathbf{x}^i - \mathbf{u}^i)\|_2 \mathbf{C}\mathbf{s}^i \\ &\neq \frac{1}{2}\mathbf{C}(\mathbf{x}^i + \mathbf{u}^{i\prime}) + \frac{1}{2}\|(\mathbf{x}^i - \mathbf{u}^i)\|_2 \mathbf{C}\mathbf{s}^i \\ &= \mathbf{C}\left(\frac{1}{2}(\mathbf{x}^i + \mathbf{u}^{i\prime}) + \frac{1}{2}\|(\mathbf{x}^i - \mathbf{u}^i)\|_2 \mathbf{s}^i\right) \\ &= \mathbf{C}\hat{\mathbf{x}}^i \\ &= \left(T_s \circ U_{\text{nei},f,S}\right)(\mathbf{x}^i; \chi_S)\end{aligned}$$

となり、式 (4.1) が成立しない。一般に、 $\|\mathbf{C}(\mathbf{x}^i - \mathbf{u}^i)\|_2 \neq \|\mathbf{x}^i - \mathbf{u}^i\|_2$ であることを証明

過程で利用した。このため、全体の更新則についても式(4.1)が成立しない。以上の証明により、題意は示された。 \square

定理 4.24 (HBC-ABC の目的関数の単調増加変換不变性に関する定理) HBC-ABC の更新則 $U_{f,S}$ は目的関数の単調増加変換不变性を有さない。

証明. HBC-ABC の更新式は、ABC の更新則を式(3.40)から式(4.14)に変更しただけである。この変更において目的関数值は無関係である。よって、ABC の定理 4.11 と全く同じ手順で式(4.1)を満たすことが導出できる。したがって、題意が成り立つことが示された。 \square

(b) HBC-ABC の変換不变性の数値実験的検証

図 4.12 は提案手法の近傍を示しており、条件は図 4.7 と同じである。この図より、提案手法では $\mathbf{x}_A = \mathbf{x}_B$ が成立することが確認できる。また、図 4.13 は図 4.8 と同様の条件で生成した提案手法の近傍を示している。この図より、提案手法の近傍は上記の超球表面上に一様に生成され、標準基底に依存しないことが確認できる。よって、提案手法は回転不变性を有しているため、解空間の回転に対して、探索性能がロバストとなることが期待できる。

4.5.4 数値実験

(a) 共通の実験条件

典型的なベンチマーク関数を用いた数値実験を行い、HBC-ABC（提案手法）の探索性能を検証する。HBC-ABC と比較するアルゴリズムは、オリジナル ABC とする。各手法の共通条件として、探索点数 $m_e = m_o = 10$ 、PSO のパラメータ $limit = 5N$ 、関数毎に共通の初期配置領域 S_0 を用いる。初期配置領域 S_0 内に初期探索点をランダムに配置した後、終了条件を満たすまで探索を進める試行を 50 回行い、50 試行の g-best の目的関数值 $f(\mathbf{x}^g)$ の平均値を実験結果とする。数値実験で用いるベンチマーク関数については、付録 C を参照されたい。

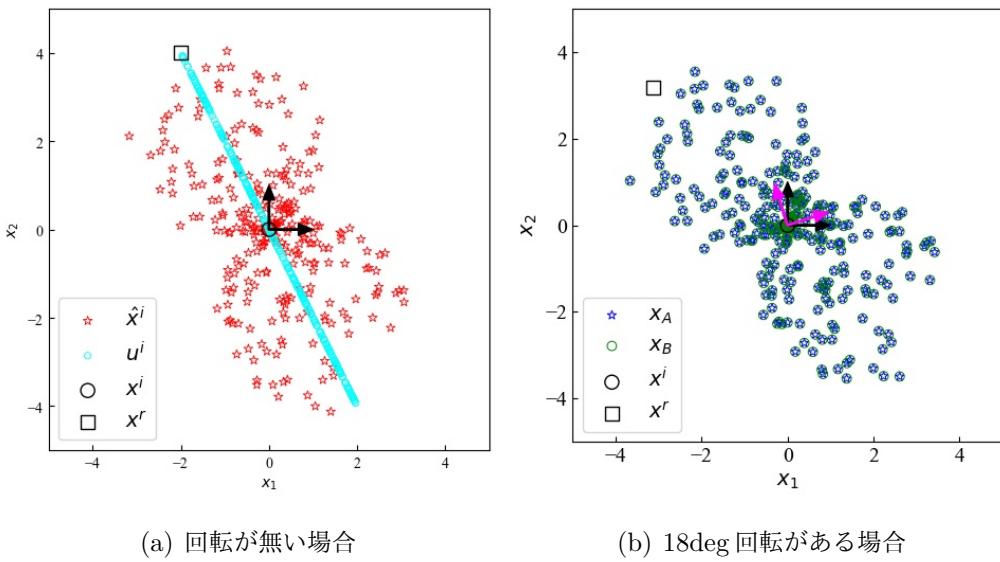


図 4.15：HBC-ABC の近傍生成

(b) アフィン変換に対するロバスト性に関する検証

数値実験を通じて、HBC-ABC のアフィン変換（回転変換を含む）に対する探索性能のロバスト性を検証する。本実験では、解空間にアフィン変換の一種である平行移動変換、スケール変換、回転変換を与えていく、その探索性能の変動を確認する。ベンチマーク関数として、変数分離可能・良スケール性・原点に最適解が存在・多峰性の 4 種類の目的関数を使用する。方法は、4.4.4 項と同様である。共通条件として、次元数 $N = 50$ 、終了条件として評価回数 $T_{\max} = 20000$ を用いる。

図 4.17、図 4.18、図 4.19 に数値実験結果を示す。各図の横軸は、平行移動パラメータ δ 、スケール変換パラメータ ε 、回転角 θ であり、全ての図の縦軸が探索性能 ($f(\mathbf{x}^g(T_{\max}))$) の平均値である。図 4.17、図 4.18、図 4.19 より、下記の結果が確認できる。

- 解空間の平行移動変換に対して、両方の ABC は探索性能の変動が小さい。
- 解空間のスケール変換に対して、オリジナル ABC は探索性能の変動が小さいことに対して、HBC-ABC は、探索性能の変動が大きい。
- 解空間の回転変換に対して、オリジナル ABC は探索性能の変動が大きいことに対して、HBC-ABC は探索性能の変動が少ない。

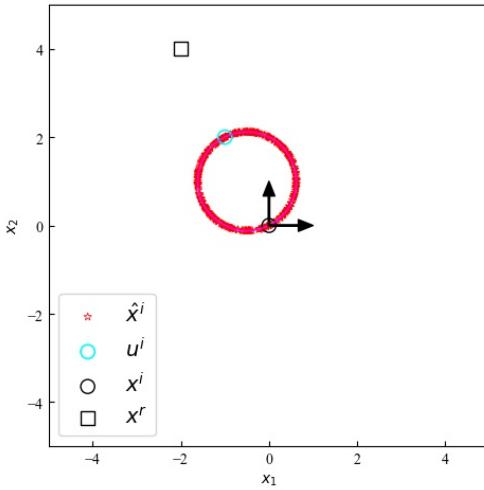


図 4.16： $\phi = 0.5$ に固定したときの HBC-ABC の近傍生成

なお、 $\theta = 90 \text{ deg}$ のときは、ベンチマーク関数の対称性により、 $\theta = 0 \text{ deg}$ と同じ条件である。

図 4.19 の結果は、回転不变性の有無が解空間の回転に対する探索性能のロバスト性に影響を与えることを示している。よって、ABC の探索性能は、変数間依存性が発生すると悪化するため、回転変換に対してロバストにならないことに対して、HBC-ABC の探索性能はロバストであることと、多くの条件で ABC よりも優れていることも確認できる。一方、図 4.18 の結果は、スケール不变性の有無が解空間のスケール変換に対する探索性能のロバスト性に影響を与えることを示している。よって、スケール不变性を具備しない HBC-ABC の探索性能は、スケール変換に対してロバストでなくなる。

(c) 全般的な性能検証

数値実験を通じて、HBC-ABC の全般的な探索性能を検証する。数値実験には、9種類のベンチマーク関数を用いる。共通条件として、次元数 $N = 10, 50, 100, 300$ 、終了条件を評価回数 $T_{\max} = 2000, 20000$ とする。

数値実験結果を表 4.2 に示す。表 4.2 では、二つの方法 (Original ABC, HBC-ABC) の結果を比較して、優れた結果に (*) を付している⁸。表 4.2 から、比較的多くの条件、特

⁸ なお、表 4.2 の一部では、同じ数値にも関わらず優劣がついている結果があるが、これは便宜上、小数点

表 4.2：全般的な探索性能に関する数値実験結果 (HBC-ABC)

関数	N	$T_{\max} = 2000$		$T_{\max} = 20000$	
		ABC	HBC-ABC	ABC	HBC-ABC
1. Parabola	10	0.00	0.00 *	0.00	0.00 *
	50	88.8	7.26 *	0.00	0.00 *
	100	397	49.5 *	0.09	0.00 *
	300	1921	329 *	251	9.91 *
2. Rosenbrock	10	9.95	8.17 *	2.32 *	3.48
	50	2736	198 *	102	48.6 *
	100	15367	920 *	408	102 *
	300	93025	6620 *	9028	743 *
3. 2^N minima	10	-763 *	-666	-783 *	-733
	50	-2835 *	-2306	-3889 *	-3054
	100	-4145 *	-3989	-7437 *	-5539
	300	-7104	-9932 *	-18749 *	-13038
4. Rastrigin	10	6.05 *	33.3	0.00 *	14.9
	50	318 *	372	10.3 *	210
	100	1047	874 *	83.3 *	583
	300	4475	2992 *	1306 *	2378
5. Schwefel	10	5.91	0.10 *	0.53	0.00 *
	50	317	38.3 *	126	2.90 *
	100	1246	193 *	541	34.1 *
	300	11541	1767 *	4627	452 *
6. Levy	10	0.05 *	0.79	0.00 *	0.009
	50	20.7	14.7 *	0.00 *	7.31
	100	55.1	23.2 *	0.15 *	12.7
	300	115	33.4 *	10.9 *	20.1
7. Ackley	10	0.32 *	0.56	0.00 *	0.17
	50	6.17	3.86 *	0.00 *	3.04
	100	8.12	4.87 *	1.03 *	3.74
	300	9.60	5.56 *	4.84	4.33 *
8. Griewank	10	0.15 *	0.34	0.01 *	0.02
	50	3.23	1.20 *	0.02	0.01 *
	100	10.8	2.26 *	0.32	0.05 *
	300	49.3	9.38 *	7.56	1.25 *
9. Alpine	10	0.15 *	1.94	0.00 *	0.01
	50	34.5	34.3 *	0.24 *	15.1
	100	137	81.5 *	3.37 *	48.6
	300	679	305 *	115 *	186

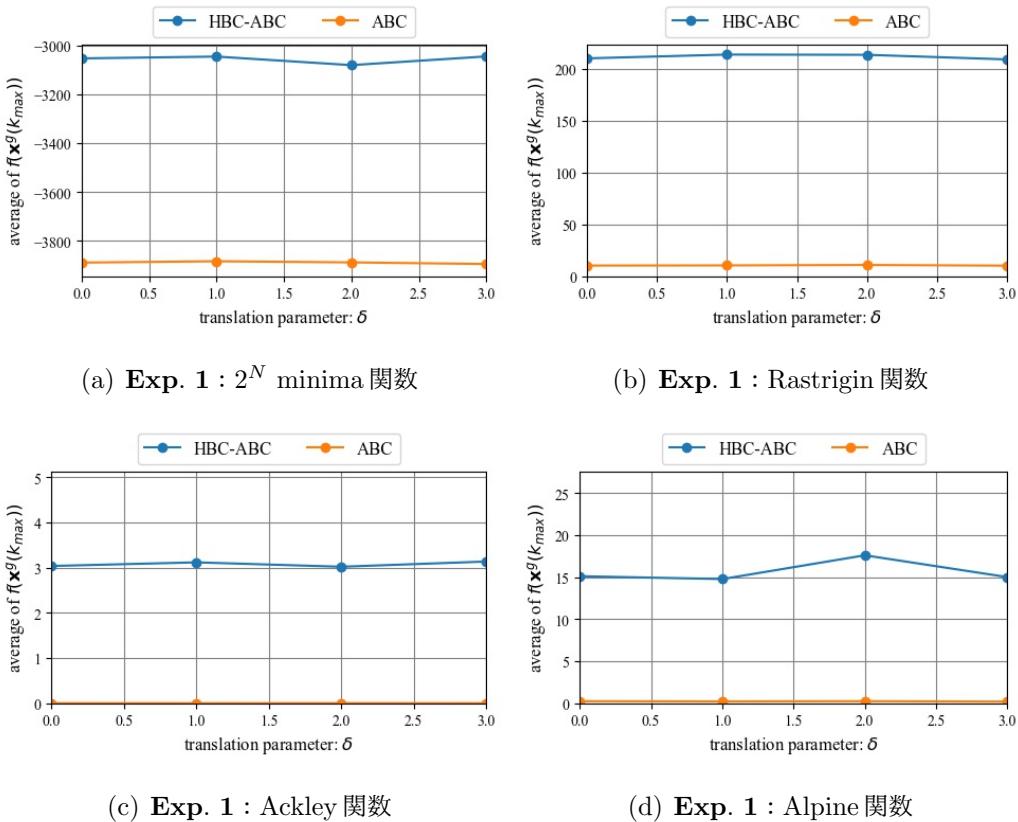


図 4.17 : 数値実験結果 (Exp. 1) : ABC と HBC-ABC の探索性能のロバスト性 ($N = 50$, $T_{\max} = 20000$)

に $T_{\max} = 2000$ かつ高次元において、HBC-ABC が ABC よりも優れていることがわかる。さらに、HBC-ABC が関数 2 と関数 5 において、顕著に優れた性能を示している。これは、回転不变性を附加したことで、変数間依存性に対して効果を顕著に発揮したためだと思われる。この結果は、ABC に回転不变性を附加することで、変数間依存性と変数間独立性の影響を受けにくい、つまり探索性能がロバストとなることを表している。

以上から、HBC-ABC は、超球を導入することで、ABC に回転不变性を附加することができる同時に、解空間の回転に対して探索性能がロバストとなることを確認した。一方、いくつか課題が残されている。CRI-PSO と同様に、短期と長期で差があるのは、①ABC

第三位以下を省略しているためであり、小数点以下の差異を確認した上で優劣を判断している。

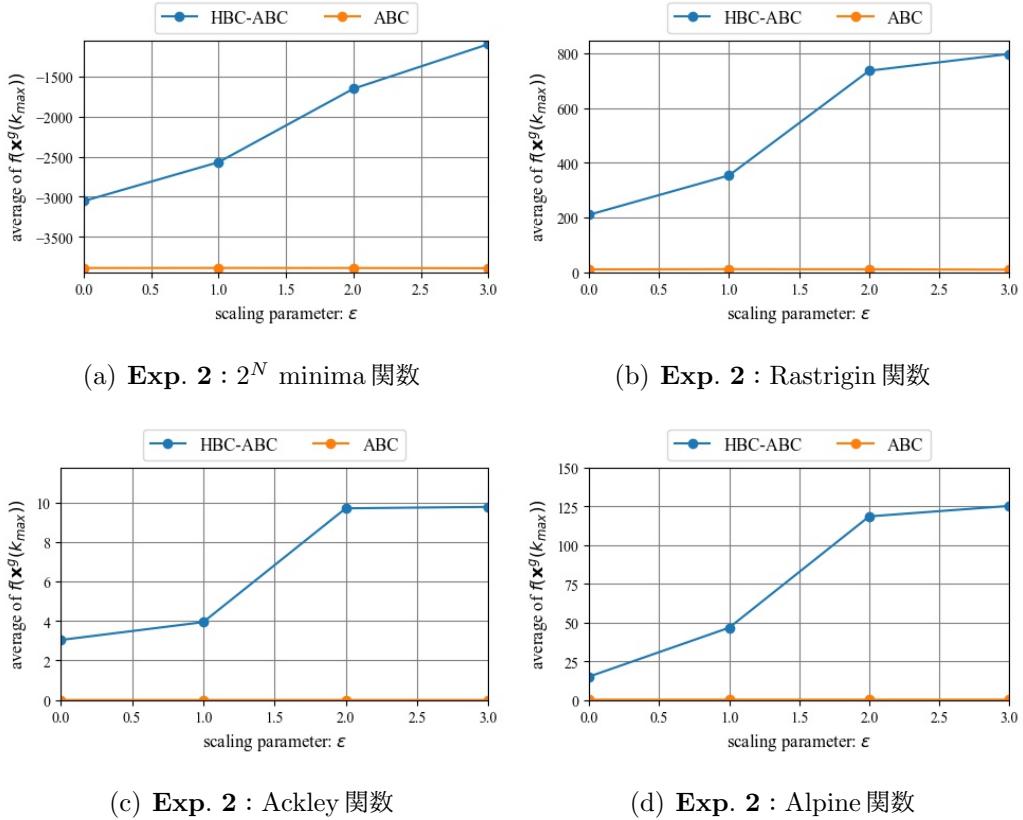


図 4.18 : 数値実験結果 (Exp. 2) : ABC と HBC-ABC の探索性能のロバスト性 ($N = 50$, $T_{\max} = 20000$)

はパラメータが少なく、停滯回数の閾値 *Limit* や探索点数しかないが、これらのパラメータによって差が生まれる、②超球に変更することで多様性を拡張したが、その一方で収束性の違いによって差が生まれる、だと考えられるが、その詳細はいまだに不明である。また、HBC-ABC はスケール不変性を有さないことから、悪スケール性に対応できない。解決策として、①乱数ベクトルに共分散行列に基づく基底変換を導入することで、超球を楕円状に変換し、悪スケール性に対応する、②超球の半径パラメータとそのパラメータ調整則を導入することで、悪スケール性を補完する[108]、などが考えられるが、今後の課題とする。

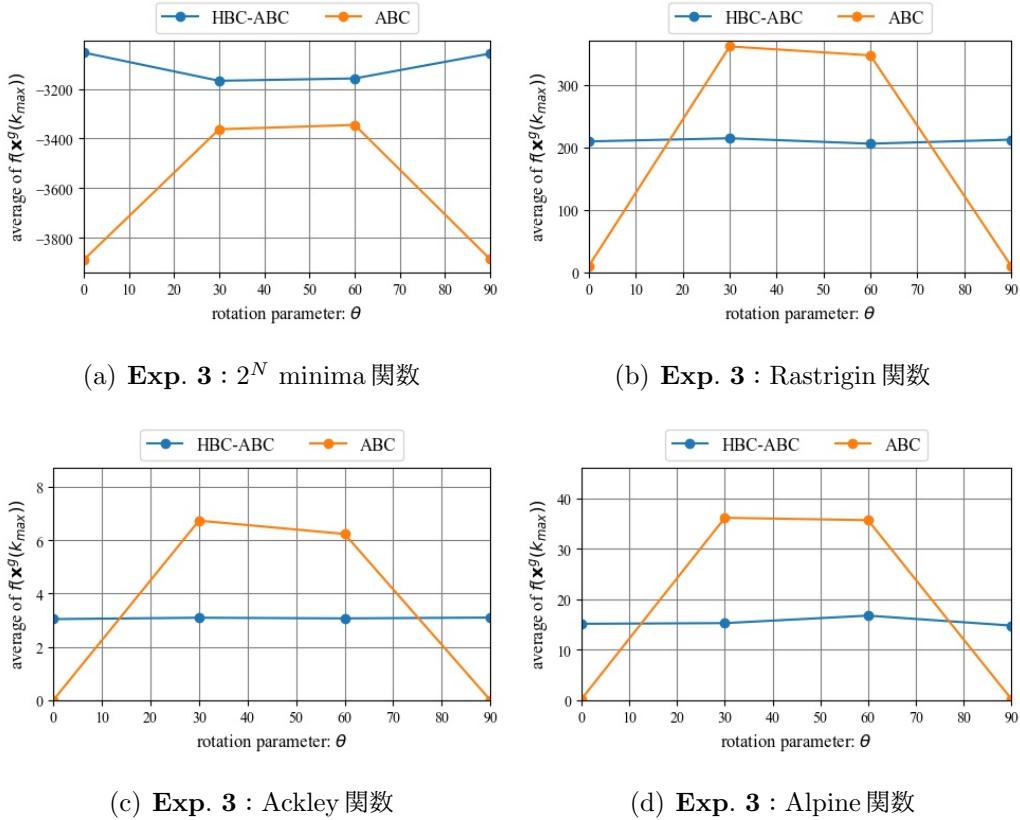


図 4.19 : 数値実験結果 (Exp. 3) : HBC-ABC の探索性能のロバスト性 ($N = 50, T_{\max} = 20000$)

4.6 基底変換を用いたアフィン変換不变性を有する Cuckoo Search

4.6.1 先行研究と背景

CS は乱数ベクトルを使用することから、スケール不变性と回転不变性を欠如している。しかしながら、CS の変換不变性に関する先行研究は、著者の知る限りなく、他のアプローチを流用することで、これらの不变性を付加できることが期待できる。CS の更新則には、乱数ベクトルを使用するため、定理 4.2 が適用できる。つまり、共分散行列に基づく基底変換を導入することで、乱数ベクトルにアフィン変換不变性を持たせることができる。こ

れに伴い、CS全体もアフィン変換不变性を獲得できると考えられる。

以上の課題を踏まえ、本節では、アフィン変換不变性を有するCSの開発を検討する。なお、ESにおいて、これと同様のアプローチが使用されている[88, 93]が、CSにおいて使用されている事例は、著者の知る限りない。

本節の構成を述べる。4.6.2項では、アフィン変換不变性を有するCSを提案する。4.6.3項では、提案手法のアフィン変換不变性の具備を示す。4.6.4項では、いくつかの変換に対する数値実験を通じて、提案手法の探索性能のロバスト性を検証する。

4.6.2 アフィン変換不变性を有する Cuckoo Search

本項では、アフィン変換不变性を有するCuckoo Searchとして、共分散行列の固有値分解に基づくCuckoo Search (Cuckoo Search based on eigenvalue Decomposition of Covariance Matrix : CMD-CS)を提案する。CMD-CSは、CSのGlobal Search (式(3.48))を式(4.14)に置き換えたアルゴリズムである。

$$\hat{\mathbf{x}}^i = \mathbf{x}^i + \alpha \Sigma_*^{1/2} \mathbf{s}^i \quad (4.14)$$

ただし、 \mathbf{s}^i は式(3.34)の乱数ベクトル、 $\alpha > 0$ はスケールパラメータである。 $\Sigma_*^{1/2}$ は $\Sigma(\mathcal{Z})$ の非対称平方根であり、式(4.15)で表される。

$$\Sigma_*^{1/2} = \mathbf{P} \Lambda^{1/2} \quad (4.15)$$

ただし、 $\Sigma(\mathcal{Z})$ は解集団 \mathcal{Z} の共分散行列、 $\mathbf{P} = [\mathbf{a}_1, \dots, \mathbf{a}_N]$ は基底変換行列、 $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$ は固有値行列である。 $\{(\mathbf{a}_n, \lambda_n) \mid n = 1, 2, \dots, N\} (\lambda_1 > \lambda_2 > \dots > \lambda_N)$ は $\Sigma(\mathcal{Z})$ の固有ベクトルと固有値であり、 $\Sigma(\mathcal{Z})$ の固有値分解によって得られる。なお、式(4.15)は式(4.3)と全く同じである。

また、解集団 \mathcal{Z} は任意に設定できるが、本節では**Step 2**における探索点群 $\mathcal{X} = \{\mathbf{x}^i \mid i = 1, \dots, m\}$ とする。CMD-CSのGlobal Searchの更新則は $H = \{\mathbf{x}^i, \mathbf{s}^i, \Sigma_*^{1/2}\}, C = \{\alpha, \beta\}$ である。ただし、Local Searchはすでにアフィン変換不变性を具備するため、CSと同様であることに注意されたい。

CSのGlobal Searchの更新則では、確率変数間が独立した乱数ベクトルを単独の項として含むため、その近傍も標準基底に依存すると同時に解空間のスケールに対応できず、ア

Algorithm 13 CMD-CS

```

1: procedure CMD-CS
2:   Step 0: パラメータ設定
3:   パラメータ  $m, k_{\max}, w, c_1, c_2$  を設定
4:   する。
5:   Step 1: 初期化
6:    $\mathbf{x}^i$  を  $S_0$  にランダムに配置する。
7:   Step 2: 座標変換行列の計算
8:   共分散行列  $\Sigma(\mathcal{Z})$  を固有値分解する。
9:   for  $i = 1, 2, \dots, m$  do
10:     $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ 
11:    if  $P_a \geq \theta$  then
12:       $\mathbf{x}^i(k)$  を式 (3.34)、式 (3.35) を
13:      用いて更新する。
14:    end if
15:    if  $T \geq T_{\max}$  then
16:      計算を終了する。
17:    else
18:      Step 2 へ戻る。
19:    end if
20:  end procedure

```

Step 4: Local Search

アフィン変換不变性を具備しない。これに対して、CMD-CS の Global Search の更新則では、その乱数ベクトルを探索点の情報を用いて基底変換するため、近傍は新たな正規直交基底に依存するようになり、アフィン変換不变性を具備すると考えられる。よって、CMD-CS の探索性能は、解空間のアフィン変換に対してロバスト性が高いことが期待される。

また、 $\Sigma_*^{1/2}$ が Σ の非対称平方根であることを考慮せずに、式 (4.9) で $\Sigma_*^{1/2} = \mathbf{P}$ とおくと、回転不变性を有するが、スケール不变性を欠如した更新則となり、逆に $\Sigma_*^{1/2} = \Lambda^{1/2}$ とおくと、スケール不变性を有するが、回転不变性を欠如した更新則となる。

Algorithm 4.6.2 に CMD-CS の疑似コードを示す。

4.6.3 CMD-CS のアフィン変換不变性に関する解析

(a) CMD-CS の変換不变性の証明

CMD-CS に関して、定理 4.25、定理 4.26 を示すことで、解空間のアフィン変換不变性と目的関数の単調増加変換不变性を有することを明らかにする。

定理 4.25 (CMD-CS の解空間のアフィン変換不变性に関する定理) CMD-CS の更新則 $U_{f,S}$ は解空間のアフィン変換不变性に対する不变性を有する。ただし、アフィン変換は $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x} - \mathbf{t}$ である。

証明. 解空間のアフィン変換 T_s のもとで、更新則 $U_{f,S}$ の引数の影響を考える。 $\mathbf{x}^{i\prime} = \mathbf{C}\mathbf{x}^i - \mathbf{t}$ となる。また、 \mathbf{s} は安定分布に従う乱数ベクトルであるため、アフィン変換しても同じベクトルだとみなせる。よって、 $\mathbf{s}' = \mathbf{s}$ と表せる。さらに、 $\Sigma_*^{1/2}$ は $\Sigma_*^{1/2\prime} = \mathbf{C}\Sigma_*^{1/2}$ となる（補助定理 A.3）。

ここで、題意が成り立つには、CMD-CS の GS（式 (4.14)）と LS（式 (3.50)）の更新則 $U_{f,S}$ について、式 (4.1) が成立すればよい（定義 4.1）。まず、GS の近傍生成則 $U_{\text{nei},f,S}$ では、

$$\begin{aligned} & U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\ &= \mathbf{x}^{i\prime} + \alpha \Sigma_*^{1/2\prime} \mathbf{s}' \\ &= \mathbf{C}\mathbf{x}^i - \mathbf{t} + \alpha \mathbf{C}\Sigma_*^{1/2} \mathbf{s} \\ &= \mathbf{C}(\mathbf{x}^i + \alpha \mathbf{s}) - \mathbf{t} \\ &= \mathbf{C}\hat{\mathbf{x}}^i \\ &= (T_s \circ U_{\text{nei},f,S})(\mathbf{x}^i; \chi_S) \end{aligned}$$

となり、式 (4.1) が成立する。また、移動則 $U_{\text{mov},f,S}$ はアフィン変換に依存しない。このため、GS 全体の更新則 $U_{f,S}$ についても式 (4.1) が成立する。よって、GS が解空間のアフィン変換不变性を有していることを示した。

また, LS の近傍生成則 $U_{\text{nei},f,S}$ では,

$$\begin{aligned}
 & U_{\text{nei},f,S'}(T_s(\mathbf{x}^i); \chi_{S'}) \\
 &= \mathbf{x}^{i'} + r(\mathbf{x}^{r_1'} - \mathbf{x}^{r_2'}) \\
 &= \mathbf{C}\mathbf{x}^i - \mathbf{t} + r(\mathbf{C}\mathbf{x}^{r_1} - \mathbf{t} - \mathbf{C}\mathbf{x}^{r_2} - \mathbf{t}) \\
 &= \mathbf{C}\mathbf{x}^i + r\mathbf{C}(\mathbf{x}^{r_1} - \mathbf{x}^{r_2}) - \mathbf{t} \\
 &= \mathbf{C}(\mathbf{x}^i + r(\mathbf{x}^{r_1} - \mathbf{x}^{r_2})) - \mathbf{t} \\
 &= \mathbf{C}\hat{\mathbf{x}}^i - \mathbf{t} \\
 &= \left(T_s \circ U_{\text{nei},f,S}\right)(\mathbf{x}^i; \chi_S)
 \end{aligned}$$

となり, 式(4.1)が成立する。また, 移動則 $U_{\text{mov},f,S}$ はアフィン変換に依存しない。このため, LS 全体の更新則 $U_{f,S}$ についても式(4.1)が成立する。よって, LS が解空間のアフィン変換不变性を有していることを示した。以上の証明により, 題意が示された。□

定理 4.26 (CMD-CS の目的関数の単調増加変換不变性に関する定理) CMD-CS の更新則 $U_{f,S}$ は目的関数の単調増加変換不变性を有する。ただし, 目的関数の単調増加変換は $T_f : f(\mathbf{x}) \mapsto g \circ f(\mathbf{x})$, g は単調増加変換である。

証明. 目的関数の単調増加変換 T_f のもとで, 更新則 $U_{f,S}$ の引数の影響を考える。 $f'(\mathbf{x}^i) = g \circ f(\mathbf{x}^i)$, $f'(\hat{\mathbf{x}}^i) = g \circ f(\hat{\mathbf{x}}^i)$ となる。下記では, $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成立すると仮定する。このとき,

$$\begin{aligned}
 f'(\mathbf{x}^i) - f'(\hat{\mathbf{x}}^i) &= g \circ f(\mathbf{x}^i) - g \circ f(\hat{\mathbf{x}}^i) \\
 &= g \circ (f(\mathbf{x}^i) - f(\hat{\mathbf{x}}^i)) \geq 0
 \end{aligned}$$

となるから, $f'(\hat{\mathbf{x}}^i) \leq f'(\mathbf{x}^i)$ が成り立つ。これは, T_f に対して目的関数値の大小関係が不变であることを表す。

CMD-CS の GS (式(4.14)) と LS (式(3.50)) において, T_f に対して $\hat{\mathbf{x}}^i$ は目的関数値を含まないため, $\hat{\mathbf{x}}^{i'} = \hat{\mathbf{x}}^i$ となり, 近傍生成則は式(4.1)を満たす。式(3.49)で, $f(\hat{\mathbf{x}}^i) \leq f(\mathbf{x}^i)$ が成り立つとき, $\mathbf{x}^i := \hat{\mathbf{x}}^i$ となる。また, T_f が与えられたとき, $f'(\hat{\mathbf{x}}^i) \leq f'(\mathbf{x}^i)$ が成り立つため, $\mathbf{x}^{i'} := \hat{\mathbf{x}}^{i'} = \hat{\mathbf{x}}^i$ となる。

ここで、題意が成り立つには、CMD-CS の更新則 $U_{f,S}$ について、式(4.1)が成立すればよい（定義 4.1）。よって、

$$\begin{aligned} & U_{T_f \circ f, S}(\mathbf{x}^i; \chi_S) \\ &= \mathbf{x}^{i\prime} \\ &= \mathbf{x}^i \\ &= U_{f,S}(\mathbf{x}^i; \chi_S) \end{aligned}$$

また、 $f(\hat{\mathbf{x}}^i) > f(\mathbf{x}^i)$ のときも上記の論証は成立する。以上の証明により、題意が示された。

□

(b) CMD-CS の変換不变性に関する数値実験的検証

数値実験的に CMD-CS のアフィン変換不变性を検証する。Local Search は CS と同様であるため、Global Search の更新則だけを検証すればよい。図 4.14 に、CMD-CS の Global Search の更新則に対して、図 4.9 と同様の条件・方法で行った図を示す。原点の○は探索点 \mathbf{x}^i 、それ以外の○は他の探索点 $\mathbf{x}^j (j \neq i)$ 、黒い矢印は標準基底ベクトル、灰色の矢印は固有ベクトルを表す。図 4.14(a) は近傍解 $\hat{\mathbf{x}}^i$ 、図 4.14(b) はアフィン変換に対する $\hat{\mathbf{x}}_A$ と $\hat{\mathbf{x}}_B$ を示している。図 4.14 から、 $\hat{\mathbf{x}}_A = \hat{\mathbf{x}}_B$ の成立が確認できる。また、近傍は、アフィン変換に関わらず、固有ベクトルに沿って生成されることも確認できる。したがって、CMD-CS の Global Search はアフィン変換不变性を具備することが示された。よって、CMD-CS はアルゴリズム全体としてアフィン変換不变性を具備する。

4.6.4 数値実験検証

(a) 共通の実験条件

典型的なベンチマーク関数を用いた数値実験を行い、CMD-CS（提案手法）の探索性能を検証する。CMD-CS と比較する手法は、オリジナル CS とする。各手法の共通条件として、探索点数 $m = 20$ 、CS のパラメータ $\alpha = 0.1, \beta = 1.9, P_a = 0.25$ 、関数毎に共通の初期配置領域 S_0 を用いる。初期配置領域 S_0 内に初期探索点をランダムに配置した後、終了

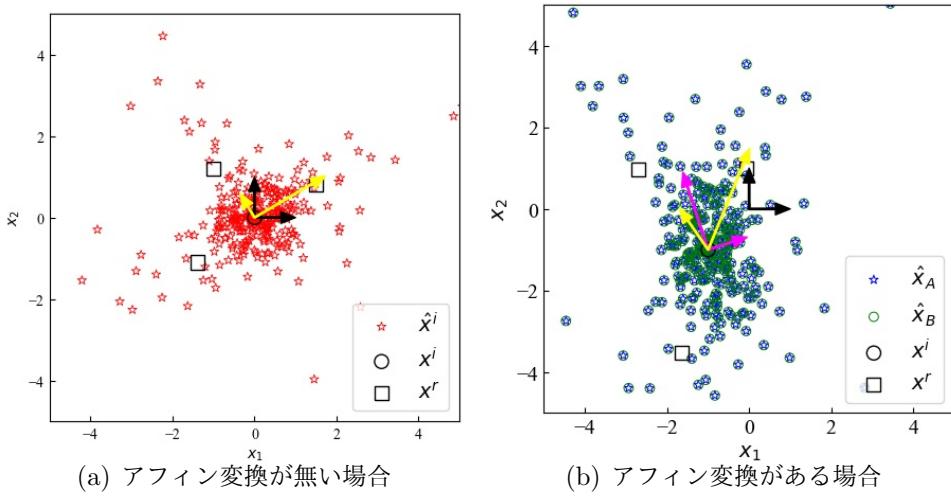


図 4.20 : CMD-CS の Global Search の近傍生成

条件を満たすまで探索を進める試行を 50 回行い、50 試行の g-best の目的関数値 $f(\mathbf{x}^g)$ の平均値を実験結果とする。数値実験で用いるベンチマーク関数については、付録 C を参照されたい。

(b) アフィン変換に対するロバスト性に関する検証

数値実験を通じて、CMD-CS のアフィン変換に対する探索性能のロバスト性を検証する。本実験では、解空間にアフィン変換の一種である平行移動変換、スケール変換、回転変換を与えていく、その探索性能の変動を確認する。ベンチマーク関数として、変数分離可能・良スケール性・原点に最適解が存在・多峰性の 4 種類の目的関数を使用する。方法は、4.4.4 項と同様である。共通条件として、次元数 $N = 50$ 、終了条件として評価回数 $T_{\max} = 20000$ を用いる。

図 4.21、図 4.22、図 4.23 に数値実験結果を示す。各図の横軸は、平行移動パラメータ δ 、スケール変換パラメータ ε 、回転角 θ であり、全ての図の縦軸が探索性能 ($f(\mathbf{x}^g)$ の平均値) である。図 4.21、図 4.22、図 4.23 より、下記の結果が確認できる。

- 解空間の平行移動変換に対して、両方の CS は探索性能の変動が小さい。
- 解空間のスケール変換に対して、オリジナル CS は探索性能の変動が大きいことに

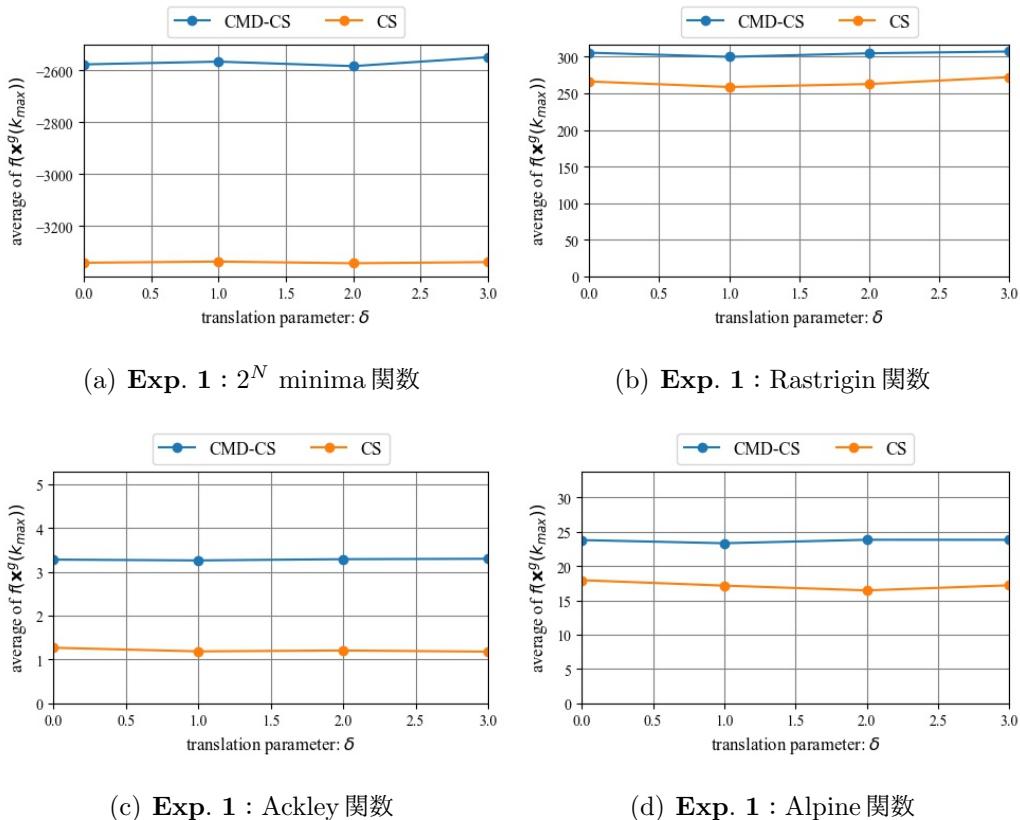


図 4.21： 数値実験結果（Exp. 1）：CS と CMD-CS の探索性能のロバスト性 ($N = 50$, $T_{\max} = 20000$)

対して、CMD-CS は、探索性能の変動が小さい。

- 解空間の回転変換に対して、オリジナル CS は探索性能の変動が大きいことに対して、CMD-CS は探索性能の変動が少ない。

図 4.22, 図 4.23 の結果は、スケール不変性や回転不変性の有無が解空間のスケール変換や回転変換に対する探索性能のロバスト性に影響を与えることを示している。よって、CS の探索性能は、変数間依存性や悪スケール性が発生すると悪化するため、ロバストにならないことに対して、CMD-CS の探索性能はロバストであることが確認できる。なお、スケール変換よりも回転変換のほうが CS の変動が比較的小さい理由は、Lévy 乱数ベクトルは大規模的には異方性（回転不変性を欠如している）だが、局所的には正規乱数ベクトルと同様

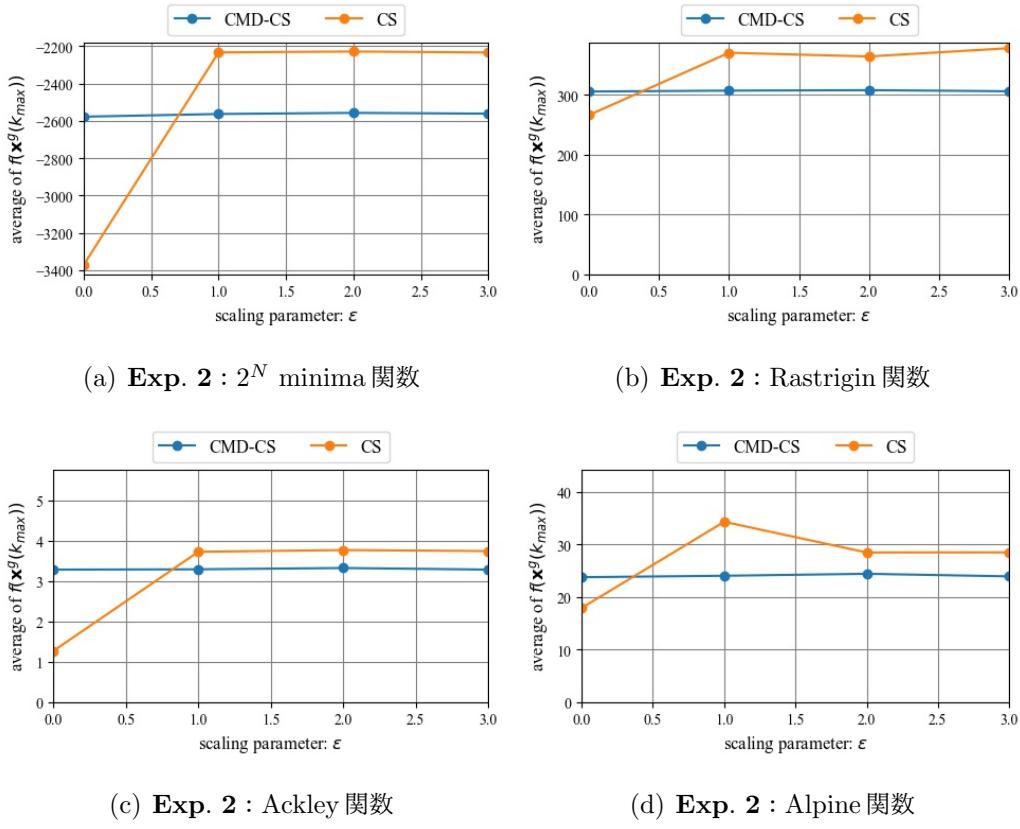


図 4.22：数値実験結果（Exp. 2）：CS と CMD-CS の探索性能のロバスト性 ($N = 50$, $T_{\max} = 20000$)

に等方性（回転不变性を有している）とみなすことができる。よって、CSの回転変換に対するロバスト性はそこまで低くないと思われる。このように、不变性の具備によって極端な結果が出るわけではなく、実質的な近傍生成を確認しなければ、具体的なロバスト性の高さは判断しにくい。

(c) 全般的性能に関する検証

数値実験を通じて、CMD-CS の全般的な探索性能を検証する。数値実験には、9種類のベンチマーク関数を用いる。共通条件として、次元数 $N = 10, 50, 100, 300$ 、終了条件を評価回数 $T_{\max} = 2000, 20000$ とする。

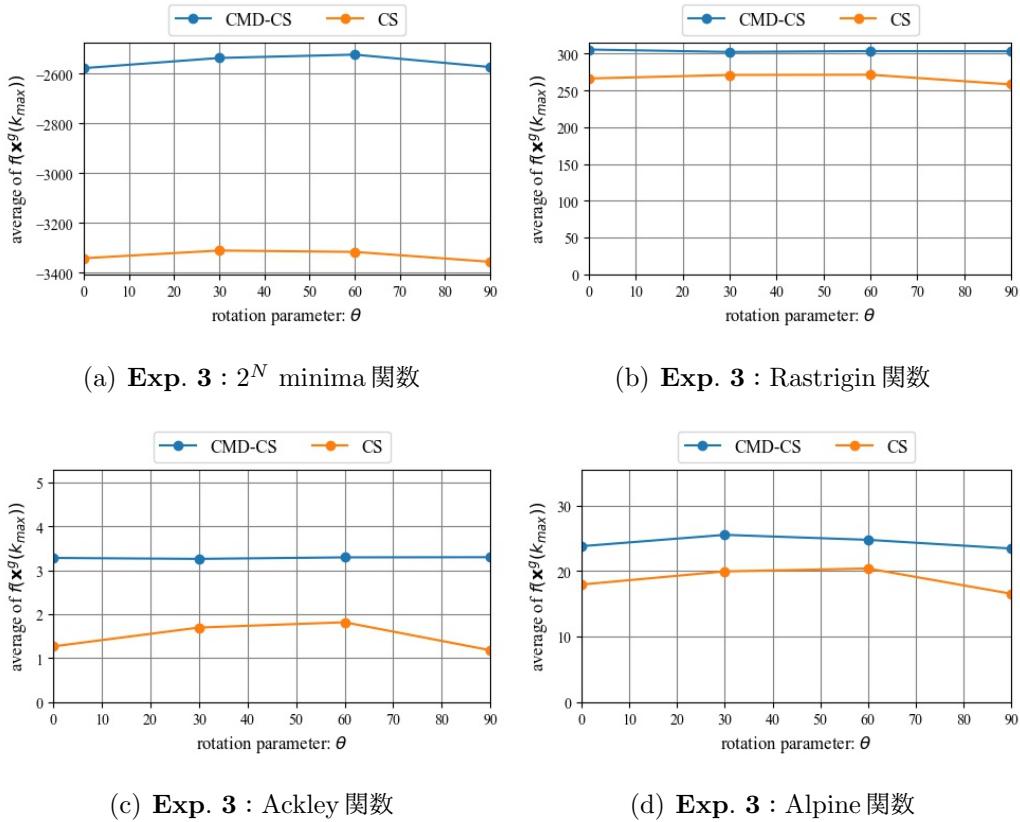


図 4.23 : 数値実験結果 (Exp. 3) : CS と CMD-CS の探索性能のロバスト性 ($N = 50$, $T_{\max} = 20000$)

数値実験結果を表 4.3 に示す。表 4.3 では、二つの方法 (CS, CMD-CS) の結果を比較して、優れた結果に (*) を付している⁹。表 4.3 から、 $T_{\max} = 2000$ において、CMD-CS が CS よりも顕著に優れていることがわかる。一方、 $T_{\max} = 20000$ のほとんどの条件においては、CS が CMD-CS よりも優れていることがわかる。この結果から、CRI-PSO, HBC-ABC と同様に、変換不变性の付加により、パラメータの影響と探索状態の関係が変わり、今回使用したパラメータでは、CMD-CS が短期で有利、CS が長期で有利、ということが発生したと考えられる。しかしながら、今回使用したパラメータにおいて、 $T_{\max} = 2000$ における CS と CMD-CS の性能差は極端に大きい（例えば関数 2, $N = 300$ では 2 衡異なる）

⁹ なお、表 4.3 の一部では、同じ数値にも関わらず優劣がついている結果があるが、これは便宜上、小数点第三位以下を省略しているためであり、小数点以下の差異を確認した上で優劣を判断している。

一方で、 $T_{\max} = 2000$ におけるCSとCMD-CSの性能差は比較的小さいため、平均的な性能ではCMD-CSのほうが上回っているといえる。

以上から、CMD-CSは、解空間のアフィン変換に対してCSよりも高いロバスト性を示すことを確認した。しかし、CMD-CSは探索性能がロバストとなった一方で、いくつかの条件ではCSよりも全体的に探索性能が悪化しているなどの課題がある。これは、共分散行列の各固有値で、対応する正規直交基底ベクトルをスケーリングしているが、探索過程で各固有値の比が極端に大きくなってしまい、多様性の低い近傍生成となっていたため、探索効率が低下したのだと考えられる。この解決策として、CMA-ESと同様に、スケールパラメータの調整則を付加することが考えられる。この方法は、上記の課題を問題を解消すると同時に、平均的な性能・適応性の向上が期待できる。

4.7まとめ

本章では、最適化アルゴリズムの変換不变性の定義に基づき、各種のメタヒューリスティクスの不变性を明らかにすると同時に、PSO、ABC、CSに対して、他の不变性の付加アプローチを流用することで、欠如していた不变性を付加したアルゴリズムを提案した。

表4.4に本章の提案手法を含む、各種メタヒューリスティクスの変換不变性の有無を示す。○は該当する不变性を具備しており、×は該当する不变性を欠如していることを表す。表4.4から、既存のメタヒューリスティクスは回転不变性か、スケール不变性を欠如していることが多いことと、本章のように欠如している不变性を付加しても、他の不变性を犠牲にしやすく、アフィン変換不变性の獲得が困難であることがわかる。これは先行研究における、不变性を付加するアプローチにおいても共通することである。

本章の成果は、下記の点で新規性を有すると考えられる。

- (1) メタヒューリスティクスが発見的手法であるが故に、各種アルゴリズムの不变性の有無や先行研究における不变性を付加するアプローチが明確でないことを指摘し、統一的な視点でその数理的構造に着目しながら不变性に関する先行研究を整理することで、不变性が欠如する構造や不变性を付加するアプローチが分類できることを明らかにした。

表 4.3 : 数値実験結果 (CMD-CS)

関数	N	$T_{\max} = 2000$		$T_{\max} = 20000$	
		CS	CMD-CS	CS	CMD-CS
1. Parabola	10	15.7	0.74 *	0.00	0.00 *
	50	295	29.9 *	0.18 *	14.1
	100	689	65.4 *	2.09 *	35.5
	300	2264	219 *	50.2 *	123
2. Rosenbrock	10	6941	95.3 *	4.63	4.19 *
	50	459018	7697 *	79.0 *	278
	100	1215876	19129 *	301 *	719
	300	4371190	70585 *	2427 *	2599
3. 2^N minima	10	-555	-670 *	-744 *	-734
	50	-554	-2330 *	-3371 *	-2578
	100	1170	-4150 *	-6158 *	-4417
	300	10619	-11185 *	-14714 *	-11611
4. Rastrigin	10	90.7	52.7 *	22.3 *	25.9
	50	778	473 *	266 *	306
	100	1605	1034 *	687 *	727
	300	5275	3178 *	2614	2572 *
5. Schwefel	10	21.0	1.13 *	0.00	0.00 *
	50	850	72.0 *	6.15 *	14.7
	100	2928	292 *	36.4 *	63.6
	300	27265	2903 *	375 *	627
6. Levy	10	43.24	11.86 *	0.64 *	1.20
	50	110	34.0 *	10.9 *	11.4
	100	139	37.6 *	14.0 *	14.7
	300	157	35.8 *	20.7	20.2 *
7. Ackley	10	7.30	3.36 *	0.04 *	0.11
	50	9.60	4.82 *	1.27 *	3.28
	100	9.92	4.83 *	2.29 *	3.70
	300	10.3	4.82 *	3.56 *	4.01
8. Griewank	10	0.27	0.16 *	0.06 *	0.14
	50	1.07	0.66 *	0.83 *	1.36
	100	1.16	0.75 *	1.21 *	1.89
	300	1.58	0.91 *	2.74 *	4.08
9. Alpine	10	3.17	1.86 *	0.48 *	1.59
	50	56.6	28.2 *	17.9 *	23.8
	100	127	57.7 *	45.0 *	57.2
	300	428	170 *	191 *	208

表 4.4：各種メタヒューリスティクスの変換不变性

手法	T_s : アフィン変換不变性				T_f : 目的関数の 単調増加変換不变性	
	線型変換不变性					
	平行移動不变性	回転不变性	スケール不变性	相似変換不变性		
ES	○	○	×	×	○	
PSO	○	×	○	○	○	
DE	○	×	○	○	○	
ABC	○	×	○	○	×	
FA	○	○	×	×	○	
CS	○	×	×	×	○	
CRI-PSO	○	×	○	○	○	
HBC-ABC	○	×	○	○	×	
CMD-CS	○	○	○	○	○	

- (2) 先行研究では、各種メタヒューリスティクスに対して、一部の不变性が主に実験的に示されてきたが、数学的な証明はされていないケースがほとんどであると同時に、特に、比較的新しい ABC や CS に関しては全く不变性が不明であったことを指摘し、数学的証明を通じて、各種メタヒューリスティクス (PSO, DE, ABC, FA, CS) の不变性の有無を具体的に明らかにした。
- (3) 先行研究において、他のアルゴリズムにおいてみられるが、PSOにおいてはまだ事例がないアプローチである「共分散行列の固有ベクトルに基づく基底変換」を流用することで、PSO に回転不变性を付加した CRI-PSO を提案した。さらに、数値実験を通じて、CRI-PSO は回転変換に対するロバスト性が向上すると同時に、全般的な性能も平均的に（特に変数間依存性において）向上することを確認した。
- (4) 先行研究において、他のアルゴリズムにおいてみられるが、ABCにおいてはまだ事例がないアプローチである「超球表面上の近傍生成」を流用することで、ABC に回転不变性を付加した HBC-ABC を提案した。さらに、数値実験を通じて、HBC-ABC は回転変換に対するロバスト性が向上すると同時に、全般的な性能も平均的に（特に変数間依存性において）向上することを確認した。

- (5) 先行研究において、他のアルゴリズムにおいてみられるが、CSにおいてはまだ事例がないアプローチである「共分散行列の固有ベクトルと固有値に基づく基底変換」を流用することで、CSにアフィン変換不変性を付加したCMD-CSを提案した。さらに、数値実験を通じて、CMD-CSはアフィン変換に対するロバスト性が向上すると同時に、全般的な性能が平均的に向上する（長期探索ではCSに劣っているが、全体的な性能差を見るとCMD-CSが優れている）ことを確認した。

一方、本章の検討を通じて、下記の課題があることが明らかとなった。

- (1) 本章では、先行研究における不变性の付加アプローチとアルゴリズムの組合せが新規であるものを選択し、適用することで対象の不变性を獲得することが可能であることを確認した。一方、CRI-PSOとHBC-ABCは従来有していたスケール不变性を欠如しているなど、アプローチによっては他の不变性を犠牲にしていることから、完全にアフィン変換不变性を付加することは困難であることも確認した。これに対して、BBOではアフィン変換不变性が重要であるため、より実用的なアルゴリズムを設計するためには、この欠如を補完するアプローチも重要である。
- (2) 本章で提案したCRI-PSO, HBC-ABC, CMD-CSに共通する傾向として、パラメータ設定が探索に与える効果がオリジナルのアルゴリズムと同様でないことを確認した。具体的には、長期探索と短期探索での性能の違いが顕著であったが、これはパラメータと多様化・集中化の関連性に強く依存するものだと考えられる。つまり、不变性の獲得は、想定した性質（変数間依存性、悪スケール性、原点依存性）に対するロバスト性が向上するという点において有効であったが、他の条件（探索時間、多峰性などの問題構造）に対しては有効であるとは限らないという点において、適応性の付加により、アルゴリズムの平均的な性能を向上させる必要がある。
- (3) 本章では各種アルゴリズムの不变性と数理的構造に着目し、不变性を付加するアプローチを適切に選択しているが、アルゴリズムに対する理解が薄いユーザにとってこの作業は困難である。

これらを受けて、課題(1)と(2)の一部に対して、第5章で解決することを検討し、課題(1)と(3)に対して、第6章で解決することを検討する。

5

パラメータ調整に基づく メタヒューリстиクス の適応化

5.1 本章について

メタヒューリстиクスは、ブラックボックス最適化 (Black-Box Optimization : BBO) のような多様な環境下での適用が想定されるため、環境変化に対する適応性が必要である。また、メタヒューリстиクスは、調整可能なパラメータを有しており、ユーザは探索条件に合わせてパラメータを適切に設定することで、その手法は十分な探索性能を発揮する。一方、BBOにおいて、これを実現するにはユーザに負担がかかるため、効率的なパラメータの設定・調整に関する方法がこれまでに開発してきた。その中でも、適応的調整型は、探索過程における探索状態を評価しながら、その状態に応じてパラメータを適応的に調整する、つまり探索状態を制御する方法である。さらに、多様化・集中化の戦略に対応させて、探索状態の評価指標を定義することで、多様化・集中化の実現能力が高まり、探索性能の改善が期待できる。よって、メタヒューリстиクスに対して、適応的調整則に分類されるパラメータ調整則を付加することは、BBO の環境下でも、パラメータの設定・調整に関するユーザの負担が一定以上解消されると同時に、平均的に探索性能を向上させることができる。一方で、パラメータ調整則の付加は、メタヒューリстиクスのロバスト性を保証する変換不变性の有無に影響を与えることがある。このため、ロバスト性と適応性の両立には、アルゴリズム全体の変換不变性を維持するようにパラメータ調整則を設計する必要がある。

本章では、メタヒューリстиクスの既存のパラメータ調整則を概観し、ロバスト性と

適応性の両立には、アルゴリズム全体の変換不变性を維持するようにパラメータ調整則を設計する必要があることを指摘する。さらに、変換不变性の観点から、適応的調整型に分類されるパラメータ調整則を付加することで、CS がロバスト性と適応性が向上することや、PSO がロバスト性を維持しながら適応性が向上することを確認する。最後に、変換不变性の観点からパラメータ調整則を付加するアプローチの課題を提起する。

本章の成果は、下記の点で新規性があると考える。

- パラメータ調整則の一般形を提示し、その統一的な観点から先行研究のパラメータ調整則を分類すると同時に、パラメータ調整則の数理的構造と変換不变性との関連性を示す。
- パラメータ調整則と不变性の関係性を活用したパラメータ調整則の設計指針を新たに 2 つ提示する。従来は発見的手法であるメタヒューリスティクスにおけるパラメータ調整則も発見的に設計されていることに対して、この設計指針に基づくアプローチは、理論的な不变性に基づく点で新しい。
- 相似変換不变性を欠如している CS に対して、スケールパラメータの適応的調整型のパラメータ調整則を付加することで、相似変換不变性を補償すると同時に、多様化・集中化に基づき適応性を向上させる。このアプローチによって、CS の変換不变性・適応性を同時に獲得した研究は著者の知る限りない。
- 回転不变性を有する PSO に対して、適応的調整型のパラメータ調整則を付加することで、回転不变性を維持すると同時に、多様化・集中化に基づき適応性を向上させる。このアプローチによって、PSO の変換不变性・適応性を同時に獲得した研究は著者の知る限りない。

なお、本章は著者の先行研究[210, 211, 212, 213, 214, 216, 218, 219, 220, 221, 222, 226, 227]に基づいている。

5.2 メタヒューリスティクスのパラメータ調整

5.2.1 パラメータ調整方法の分類と一般形の定義

(a) パラメータ調整方法の分類

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。

最適化アルゴリズムのパラメータ設定・調整方法はいくつかの側面で分類できる。まず、オフラインチューニングとオンラインチューニングがある[204]。オフラインチューニングは、パラメータをある値に固定した下で探索を実行し、結果を確認した後、パラメータを異なる値に設定し直す、ということを繰り返すことで、パラメータを良い値に調整していく。一方、オンラインチューニングは、ある指針に基づき、探索過程でパラメータの値を動的に調整する。さらに、オンラインチューニングのパラメータ調整方法は、最適化アルゴリズムに与える適応性のレベルに応じて、下記の3種類に分類できる¹。

固定型: パラメータはユーザが予め設定し、探索過程で同じ値で固定する機構。

スケジューリング的調整型: 事前に定められた時間経過に応じたスケジュールに従って、探索過程でパラメータを動的に調整する機構。

適応的調整型: 探索の時間経過に限らず、探索過程で得られた情報を活用して、パラメータを適応的に調整する機構。

一般的に、メタヒューリスティクスが有するパラメータは探索過程で一定であるため、「パラメータ固定型」に分類される。また、パラメータの時系列を決定する方法は、下記の2

¹ 適応的調整型は、探索内で予め定めた何らかの条件を満たした際にパラメータを調整する機構である「反応的調整型」、予め外部から探索の指針となる何らかの情報を与えた上で、探索過程で得られた情報から、探索の指針に従うようにパラメータを調整する機構である「適応的調整型」、外部信号を一切利用せず、探索過程で得られた内部情報のみから指針を作成および改善し、パラメータを調整する機構である「自律適応的調整型」の3種類に詳細に分類することもある[211]が、ここではこれらをまとめて扱う。

種類に分類できる。

フィードフォワード型： 事前に与えた情報だけを用いて、パラメータの時系列を確定する。

フィードバック型： 探索過程で得られる情報も用いて、パラメータの時系列を逐次生成する。

これらの分類に基づくと、固定型は静的・フィードフォワード型、スケジューリング的調整型は動的・フィードフォワード型、適応的調整型は動的・フィードバック型、というよう²に解釈できる²。

メタヒューリスティクスのパラメータは、その多様化・集中化に多大な影響を与え、その探索性能を左右するため、優れた性能を発揮するパラメータ設定は対象問題や探索条件に依存することが、オフラインチューニングで困難な点である。一方、オンラインチューニングは、パラメータを探索過程で動的に変動させることで、多様化・集中化の「多様性」を拡大させることができることから、適切なチューニングを与えれば、ユーザの負荷を軽減すると同時に、平均的な性能向上が期待できる。中でも、適応的調整型は、探索過程で逐次得られる情報をパラメータの時系列の生成に活用することから、最も適応性の発揮が期待できるといえよう。このような背景から、パラメータ調整則に関する先行研究では、オンラインチューニングが中心に開発されてきた。以降では、オンラインチューニングのパラメータ調整則を単に「パラメータ調整則」と呼ぶことにする。

(b) パラメータ調整則の一般形の定義

次節以降、代表的なパラメータ調整則を概観・分類すると同時に、変換不变性との関連性を見出す。パラメータ調整則と更新則全体の変換不变性の関連性を確認するには、パラメータ調整則の数理的構造に着目する必要がある。

そこで、パラメータ調整則の一般形として、定義 5.1 で定義する。

² また、[205]では、オンラインチューニングのパラメータ調整則は、決定的パラメータ制御、適応的パラメータ制御、自己適応的パラメータ制御と分類できると指摘しているが、実質的には後者二つの明確な区別ではなく、決定的パラメータ制御が「フィードフォワード型」、後者二つが「フィードバック型」と同義である。

定義 5.1 式 (2.1) の目的関数 f の最小化問題を対象としたとき、パラメータ $\alpha \in \mathbb{R}$ がパラメータ調整則 $F_{f,S}$ によって時系列で調整されるとき、パラメータ調整則 $F_{f,S}$ は式 (5.1) で表される。

$$\alpha(k+1) = F_{f,S}(\alpha(k); \chi_S(k), \mathcal{R}_F(k), \mathcal{C}_F(k)) \quad (5.1)$$

ただし、 $\chi_S(k)$ は探索点群 $\mathcal{X}(k)$ の探索履歴、 $\mathcal{C}_F(k)$ は更新用パラメータ集合、 $\mathcal{R}_F(k)$ は確率変数用パラメータ集合である。

定義 5.1 は、探索点の更新則の定義 3.1 と照らし合わせると、式 (5.1) は「パラメータに関する更新則」だと解釈できる。つまり、 $k+1$ 回目の探索で使用するパラメータ $\alpha(k+1)$ は、 $\alpha(k)$ の他に、探索履歴の情報を用いて、調整則 F によって生成される。パラメータ調整則 F は、更新則 U と連動して機能するため、更新則全体として見ると、パラメータ調整則 F の $\mathcal{R}_F(k), \mathcal{C}_F(k)$ は、更新則 U_f の $\mathcal{R}(k), \mathcal{C}(k)$ に包含されるとみなす。

さらに、定義 5.1 は先ほどの分類を包含している。固定型は、 α の初期値 α_{start} だけを与える、式 (5.1)において、 $\alpha(k) = \alpha_{\text{start}}, \forall k, \mathcal{C}_F(k) = \{\alpha_{\text{start}}\}$ とした場合だとみなせる。ただし、初期値 α_{start} の決定は、初期配置領域のサイズなどから導出する場合もある。スケジューリング的調整型は、 α の初期値 α_{start} や終端値 α_{end} などを与え、式 (5.1)において、 k と $\mathcal{C}_F(k) = \{\alpha_{\text{start}}, \alpha_{\text{end}}\}$ を引数とした場合だとみなせる。つまり、これは探索過程において、時間経過の情報をのみを使用することから、事前に時系列が確定している。適応的調整型は、式 (5.1)を満たす様々な形態を考えられ、探索過程で得られる探索履歴 $\chi_S(k)$ の情報を含むことから、その探索状態に応じて、次のパラメータを逐次生成していくことがわかる。以降では、先行研究におけるパラメータ調整則を定義 5.1 に照らし合わせて、分類していく。

5.2.2 Evolution Strategy のパラメータ調整方法

ES のパラメータ調整に関する研究は古くから行われており、その中でも優秀な方法は 2020 年現在でも使用されている。最初の単点型 ES に対して、Schwefel により $1/5$ ルールが付加された $(1+1)$ -ES が開発され、1990 年以降は多点型 ES が主要な研究対象となった。

ES や GA などの進化計算において、近傍生成の元となる探索点集団を親集団と呼び、近傍解集団を子集団と呼ぶが、ES の $(\mu + \lambda)$ などの表記は、主に探索点の移動（選択）の際に、 μ 個の親集団と λ 個の子集団をどのように組み合わせて選択の母集団を作成するかを意味している。このため、(1+1)-ES という表記は、探索点自身と近傍解を合わせた 2 個から良い解を次の探索点とすることを表す。従来の ES のパラメータ調整は適応的調整型のパラメータ調整則がほとんどであるため、下記ではこの部類に特化して説明していく。

ES の適応的調整型として代表的な設定方法は、(1+1)-ES の 1/5 ルール[58]や、多点型 ES の (μ, λ) -ES [58]と Covariance Matrix Adaptation-ES (CMA-ES) が知られている。1/5 ルールは現時点から一定の反復回数分遡った期間において、探索点の改善が成功した頻度を $p_s(k)$ とし、 $\sigma(k)$ を式 (5.21) を用いて調整する。

$$\sigma(k+1) = \begin{cases} \sigma(k)/c, & p_s(k) > 1/5 \\ c\sigma(k), & \text{otherwise} \end{cases} \quad (5.2)$$

ただし、 $0 < c < 1$ である。1/5 ルールは、 $p_s(k)$ が $1/5$ となるように $\sigma(k)$ を適応的に増減させる最も簡単な方法であるが、多点型 ES では使用できない。 (μ, λ) -ES では、突然変異に加えて、GA と同様に交叉も行われる。交叉は、パラメータ $C_c \in [0, 1]$ を設定し、 $C_c \leq \theta_c$ ($\theta_c \sim \mathcal{U}_{\mathbb{R}}(0, 1)$) を満たした場合に行い、 $\mathbf{u}^i(k)$ を生成する。更新則中の交叉の方法として、実数値 GA と同様に式 (5.22) や式 (5.23) などがある。

$$\mathbf{u}^i(k) = \frac{1}{2}(\mathbf{x}^{r_1}(k) + \mathbf{x}^{r_2}(k)) \quad (5.3)$$

$$u_n^i(k) = \begin{cases} x_n^{r_1}(k), & \theta < 1/2 \\ x_n^{r_2}(k), & \text{otherwise} \end{cases} \quad (5.4)$$

ただし、 $r_1, r_2 \in \mathcal{U}_{\mathbb{Z}}(1, m)$ ($r_1 \neq r_2$)、 $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ とする。なお、 $C_c > \theta_c$ の場合、交叉を行わず、 $\mathbf{u}^i(k) = \mathbf{x}^{r_1}$ か、 $\mathbf{u}^i(k) = \mathbf{x}^{r_2}$ とする。更新則中の突然変異は、式 (5.24) で行われる。

$$\hat{\mathbf{x}}^i(k) = \mathbf{u}^i(k) + \mathbf{B}(\boldsymbol{\alpha}^i)\Sigma^{1/2,i}\mathbf{s}^i(k) \quad (5.5)$$

ただし、 $\mathbf{s}^i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 、 $\Sigma^{1/2,i} = \text{diag}[\sigma_1^i, \dots, \sigma_N^i]^T$ 、 $\Sigma^{1/2,i}$ は Σ^i の平方根行列で、 $\Sigma^{1/2,i} = \text{diag}[\sigma_1^{1/2,i}, \dots, \sigma_N^{1/2,i}]^T$ である。 $\mathbf{B}(\boldsymbol{\alpha}^i)$ は付録 A の回転行列であり、回転角ベクトル $\boldsymbol{\alpha}^i =$

$[\alpha_{1,2}^i, \alpha_{2,3}^i, \dots, \sigma_{N-1,N}^i] \in \mathbb{R}^{N-1}$ を入力すると、式 (A.32) の内部が $\mathbf{B}^{i,j}(\alpha_{i,j}^{i'})^3$ とする回転行列が output される。式 (5.24) は実質的に $\hat{\mathbf{x}}^i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{B}(\boldsymbol{\alpha}^i)\Sigma^i\mathbf{B}^T)(\boldsymbol{\alpha}^i)$ と等価である。なお、この共分散行列 $\mathbf{B}(\boldsymbol{\alpha}^i)\Sigma^i\mathbf{B}(\boldsymbol{\alpha}^i)^T$ は、付録 A の補助定理 A.2 と定理 4.1 を活用することで導出できる⁴。選択では、近傍解集団の中から、目的関数値に基づくランクが上位 m 個の解を探索点群とする。

(μ, λ) -ES の更新則では、対角行列 Σ^i と、回転角ベクトル $\boldsymbol{\alpha}^i$ がパラメータとして導入されている。これらのパラメータ調整則は、式 (5.25), 式 (5.26) で行われる。

$$\sigma_n^i(k+1) = \sigma_n^i(k) \exp(\tau^1 + \tau_n^2) \quad (5.6)$$

$$\alpha_{\ell,j}^i(k+1) = \alpha_{\ell,j}^i(k) + \tau_{\ell,j}^3 \quad (5.7)$$

ただし、 $\tau^1 \sim \mathcal{N}(0, \gamma^1)$, $\tau_n^2 \sim \mathcal{N}(0, \gamma^2)$, $\tau_{\ell,j}^3 \sim \mathcal{N}(0, \gamma^3)$ である。推奨値は、 $\gamma^1 = 1/(\sqrt{2\sqrt{N}})$, $\gamma^2 = 1/(\sqrt{2N})$, $\gamma^3 = (5/360) \times 2\pi \approx 0.0873$ である。このように、 (μ, λ) -ES は、悪スケールかつ変数間依存性を持つ目的関数に対処するために、 σ を要素毎に用意し、回転パラメータを導入すると同時に、1/5 ルールの代わりに正規乱数で調整することにしている。なお、 $(\mu + \lambda)$ -ES は交叉と突然変異の方法は (μ, λ) -ES と同様だが、選択の際に親集団と子集団を合わせた集団の中から、目的関数値に基づくランクが上位 m 個の解を探索点群とする。

Covariance Matrix Adaptation-ES (CMA-ES) はこれまでに多様な方法が提案されている。ここでは最も標準的に使用されている、 (μ_w, λ) -CMA-ES について取り上げる。 (μ_w, λ) -CMA-ES では、Weighted Recombination, Cumulative Step-Size Adaptation, Hybrid Covariance Matrix Adaptation が組み合わさっている。なお、CMA-ES における λ の表記は探索点数の意味で使用されるが、本論文全体で統一するために以降では探索点数 m を使用する。 (μ_w, λ) -CMA-ES では、次の探索点 $\mathbf{x}^i(k+1)$ は下記の通り生成される。

$$\mathbf{x}^i(k+1) = \mathbf{m}(k) + \sigma(k) \mathbf{z}^i(k) \quad (5.8)$$

³ 指数部を i' としているが、更新則中の探索点番号 i と式 (A.32) 中の要素 i が混同しないように、探索点番号を i' と置き換えているだけである。

⁴ 線型変換 \mathbf{C} を与えたとき、標準正規分布が共分散行列 \mathbf{CC}^T を持つ正規分布に変換されるため、 $\mathbf{CC}^T = \mathbf{B}\Sigma^{1/2,i}(\mathbf{B}\Sigma^{1/2,i})^T = \mathbf{B}\Sigma^{1/2,i}(\Sigma^{1/2,i})^T\mathbf{B}^T = \mathbf{B}\Sigma^{1/2,i}\Sigma^{1/2,i}\mathbf{B}^T = \mathbf{B}\Sigma^i\mathbf{B}^T$ と導出される。ただし、対角行列の性質 ($(\Sigma^{1/2,i})^T = \Sigma^{1/2,i}$) を利用した。

$$\mathbf{z}^i(k) = \mathbf{C}_*^{1/2}(k) \mathbf{s}^i(k) \quad (5.9)$$

ただし、 $\mathbf{s}^i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 、 $\mathbf{C}_*^{1/2}(k)$ は $\mathbf{C}(k) = \mathbf{C}_*^{1/2}(k)(\mathbf{C}_*^{1/2}(k))^T$ を満たす行列、 $\mathbf{m}(k)$ は平均ベクトル、 $\mathbf{C}(k)$ は共分散行列パラメータである。これは、実質的に $\mathbf{x}^i(k+1) \sim \mathcal{N}(\mathbf{m}(k), \sigma^2(k)\mathbf{C}(k))$ 、 $\mathbf{z}^i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{C}(k))$ というサンプリングと等価である。下記の $\mathbf{C}_*^{1/2}(k)$ は $\mathbf{C}(k) = \mathbf{C}_*^{1/2}(k)(\mathbf{C}_*^{1/2}(k))^T$ を満たす。

$$\mathbf{C}_*^{1/2}(k) = \mathbf{P} \boldsymbol{\Lambda}^{1/2} \quad (5.10)$$

$\mathbf{P} = [\mathbf{a}_1, \dots, \mathbf{a}_N]$ は基底変換行列、 $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N)$ は固有値行列である。 $\{(\mathbf{a}_1, \lambda_1), (\mathbf{a}_2, \lambda_2), \dots, (\mathbf{a}_N, \lambda_N)\}$ ($\lambda_1 > \lambda_2 > \dots > \lambda_N$) は $\mathbf{C}(k)$ の固有ベクトルと固有値であり、 $\mathbf{C}(k)$ の固有値分解によって得られる。

平均ベクトル \mathbf{m} は下記の Weighted Recombination により更新する。

$$\mathbf{m}(k+1) = \sum_{i=1}^{\mu} w_i \mathbf{x}^{i:m}(k+1) \quad (5.11)$$

ただし、 $1 < \mu < m$ は上位点数である。 $\mathbf{x}^{i:m}(k+1)$ は探索点群 $\mathcal{X}(k+1)$ で目的関数値 $f(\mathbf{x}^i(k+1))$ の昇順に $\mathbf{x}^{i:m}(k+1)$ ($i = 1, 2, \dots, m$) をソートし、新たに探索点番号を割り当てた後の i 番目の探索点である。つまり、探索点群 $\mathcal{X}(k+1)$ 内で i 個目に優れた探索点である。なお、 $\sum_{i=1}^{\mu} w_i = 1$ 、 $w_i > 0$ ($i = 1, 2, \dots, \mu$)、 $w_i = 0$ ($i = \mu + 1, \mu + 2, \dots, m$) である。 $\mathbf{x}^{i:m}$ に対応する $\mathbf{s}^i, \mathbf{z}^i$ も、 $\mathbf{s}^{i:m}, \mathbf{z}^{i:m}$ と表記する。下記により平均ベクトル \mathbf{m} の変位 \mathbf{y} が得られる。

$$\mathbf{y}(k+1) = \frac{\mathbf{m}(k+1) - \mathbf{m}(k)}{\sigma(k)} \quad (5.12)$$

ここから、 $\mathbf{m}(k+1)$ を消すように変形すると、

$$\mathbf{y}(k+1) = \sum_{i=1}^{\mu} w_i \mathbf{z}^{i:m}(k) \quad (5.13)$$

このように変形すると、変位ベクトル \mathbf{y} は上位 μ 個の乱数ベクトル $\mathbf{z}^i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{C}(k))$ の加重平均であることがわかる。また、 $\mathbf{z}^{i:m}$ は上位探索点と平均ベクトルの差分ベクトルであることから、Successful Variations と呼ばれる。

ES 系におけるステップサイズ σ の調整は Step Size Adaptation (SSA) と呼ばれる。 (μ_w, λ) -CMA-ES における SSA として、ステップサイズ σ は下記の Cumulative Step-Size Adaptation (CSA) により更新する。

$$\mathbf{p}^\sigma(k+1) = (1 - c_\sigma)\mathbf{p}^\sigma(k) + \sqrt{c_\sigma(2 - c_\sigma)\mu_w}\mathbf{s}_w(k) \quad (5.14)$$

$$\mathbf{s}_w(k) = \sum_{i=1}^{\mu} w_i \mathbf{s}^{i:m}(k) \quad (5.15)$$

$$\sigma(k+1) = \sigma(k) \cdot \exp \frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}^\sigma(k+1)\|_2}{\xi_d} - 1 \right) \quad (5.16)$$

ただし、 $c_\sigma > 0$ は学習率パラメータ、 $\mu_w = 1 / \sum_{i=1}^{\mu} w_i^2$ である。 $\xi_d = \mathbb{E}[\|\mathbf{s}^i\|_2] = \mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2] \approx \sqrt{N}(1 - \frac{1}{4N} + \frac{1}{21N^2})$ である。 \mathbf{p}^σ は進化パス (Evolution Pass) と呼ばれる、CMA-ES において重要な量であり、共分散行列パラメータ \mathbf{C} が \mathbf{I} に規格化された状態で、特定の期間にわたって平均ベクトルがどの程度変位したかを表す。CSA は、平均ベクトルがランダムに変位しているなら、 σ はほぼ変化させず、平均ベクトルの変位が小さければ、 σ を小さく、平均ベクトルの変位が特定方向であれば、 σ を大きくする。つまり、平均ベクトルの変位が特定方向であるとき、大きく改善が見込める方向に探索点群が向かっているため、探索点のサンプリング範囲を広げ（多様化）、逆に平均ベクトルの変位が小さいときは、サンプリング内部に改善が見込める領域があるとみなし、サンプリング範囲を狭める（集中化）、という戦略である。

CMA-ES 系における共分散行列パラメータ \mathbf{C} の調整は Covariance Matrix Adaptation (CMA) と呼ばれる。最初の CMA のバージョンとして、少ない探索点数を想定した Rank-One Update が提案されたが、後に多い探索点数を想定した Rank- μ Update が提案された。さらに、Hybrid Covariance Matrix Adaptation (Hybrid CMA) は Rank-One Update と Rank- μ Update を組み合わせることで、探索点数を少なくとも多くても従来の探索効率を著しく落とすことがないと言われている。 (μ_w, λ) -CMA-ES における共分散行列パラメータ \mathbf{C} は、Hybrid CMA により更新する。Rank-One Update の進化パスは下記の通りである。

$$\mathbf{p}^C(k+1) = (1 - c_C)\mathbf{p}^C(k) + h_\sigma(k+1)\sqrt{c_C(2 - c_C)\mu_w}\mathbf{y}(k+1) \quad (5.17)$$

$$h_\sigma(k) = \mathbb{1}_{\frac{\|\mathbf{p}^{\sigma(k)}\|_2}{\sqrt{1-(1-c_C)^{2k}}} < \frac{\epsilon_d}{2}(3 + \frac{4}{2N-1})} \quad (5.18)$$

ただし、 $c_C > 0$ は進化パスの学習率パラメータ、 $\mathbb{1}_P$ は指示関数であり、条件 P が真ならば 1、そうでなければ 0 を返す。Rank- μ Update では、下記の \mathbf{C}_μ を利用する。

$$\mathbf{C}_\mu = \sum_{i=1}^{\mu} w_i \mathbf{z}^{i:m}(k) (\mathbf{z}^{i:m}(k))^T \quad (5.19)$$

Hybrid CMA による共分散行列パラメータ \mathbf{C} の更新は下記で行われる。

$$\begin{aligned} \mathbf{C}(k+1) = & (1 - c_{\text{cov}})\mathbf{C}(k) + c_{\text{cov}}\left(1 - \frac{1}{\mu_C}\right)\mathbf{C}_\mu \\ & + \frac{c_{\text{cov}}}{\mu_C}(\mathbf{p}^C(k+1)(\mathbf{p}^C(k+1))^T + (1 - h_\sigma(k+1))\mathbf{C}(k)) \end{aligned} \quad (5.20)$$

ただし、 $c_{\text{cov}} > 0$ は CMA の学習率パラメータである。なお、 $\mathbf{x}\mathbf{y}^T$ は \mathbf{x} と \mathbf{y} の直積（外積、テンソル積）を表しており、ベクトルの要素同士の積を各要素とする行列となる。例えば、 $\mathbf{x} \in \mathbb{R}^{N_1}, \mathbf{y} \in \mathbb{R}^{N_2}$ の場合、直積は $\mathbf{x}\mathbf{y}^T = [(x_i y_j)_{i,j}] \in \mathbb{R}^{N_1 \times N_2}$ という行列となる。ベクトル自身同士の直積、つまり $\mathbf{x} \in \mathbb{R}^N$ 自身同士の直積は $\mathbf{x}\mathbf{x}^T = [(x_i^2)_{i,j}] \in \mathbb{R}^{N \times N}$ という、ベクトルの各要素を二乗とした値を要素とする行列となる。Rank-One Update では、進化パスに平均ベクトルの変位 $\mathbf{y}(k+1)$ を学習させることで、共分散行列パラメータ \mathbf{C} を変位方向の相関を強める役割を持つ。Rank- μ Update では、Successful Variations $\mathbf{z}^{i:m}(k)$ から計算される重み付きの共分散行列 \mathbf{C}_μ を導出しておき、共分散行列パラメータ \mathbf{C} を \mathbf{C}_μ に近づける役割を持つ。どちらも、改善が見込める量や方向を共分散行列に学習させ、サンプリング範囲（楕円形）を変形させている。

CMA-ES では、各パラメータの推奨設定が与えられていることが多い。 (μ_w, λ) -CMA-ES のパラメータでは、Weighted Recombination における探索点数は $m = 4 + \lfloor 3 \ln(N) \rfloor$ 、上位点数は $\mu = \lfloor m/2 \rfloor$ 、重みは $w_i = (\ln(\mu+1) - \ln(i))/(\mu \ln(\mu+1) - \sum_{j=1}^{\mu} \ln(j))$ で、CSA における進化パスの学習率は $c_\sigma = (\mu_w + 2)/(N + \mu_w + 3)$ 、減衰パラメータは $d_\sigma = 1 + d_\sigma + 2 \max(0, \sqrt{(\mu_w - 1)/N} - 1)$ で、Rank-One Update における進化パスの学習率は $c_C = 4/(N + 4)$ 、Rank- μ Update における進化パスの学習率は $\mu_C = \mu_w$ 、CMA における学習率は $c_{\text{cov}} = \frac{2}{\mu_C(N+\sqrt{2})^2} + (1 - \frac{1}{\mu_C}) \min(1, \frac{2\mu_w - 1}{(N+1)^2 + \mu_w})$ が推奨設定方法として与えられている。調整されるパラメータの初期値は、 $\mathbf{p}^\sigma(1) = \mathbf{p}^C(1) = \mathbf{0}$ 、 $\mathbf{C}(1) = b\mathbf{m}\mathbf{I}$ 、

$\mathbf{m}(1) = (\gamma_{\max} + \gamma_{\min})/2$, $\sigma(1) = (1/N) \sum_{n=1}^N (\gamma_{\max,n} - \gamma_{\min,n})/2$ とする場合が多い。なお, $\gamma_{\max}, \gamma_{\min}$ は初期配置領域 $S_0 = [\gamma_{\min,n}, \gamma_{\max,n}]^N$ から決めている。推奨値については [113]に詳しく記載されている。

また, 他の CMA-ES として, $(\mu/\mu_w, \lambda)$ -CMA-ES がよく使用される[113]。元々, CMA-ES は单峰性を仮定した理論的な背景もあり, 单峰性関数に対しては著しく良好な性能を発揮したが, 多峰性関数で性能を発揮するには探索点数を推奨値よりも大きくする必要であることが知られている。これらを踏まえ, 多峰性関数に対しても有効な CMA-ES として, 探索点数を増やしたときの CSA の課題を解決した機能分担 CMA-ES [113]や, リスタート毎に探索点数を増加させる IPOP-CMA-ES [113], 目的関数のノイズなどの不確定性へ対処した UH-CMA-ES [113], マルチスタートと IPOP-CMA-ES を組み合わせた BI-POP-CMA-ES [113], 非効率なリスタートを解消するための探索点数の適応的調整則を備えた PSA-CMA-ES [113]など, 非常に多くのバリエーションが提案されている。

CMA-ES はアフィン変換不变性と目的関数の単調増加変換不变性を具備することが示されており, これらの不变性と上記のような適応的パラメータ調整則を具備するメタヒューリスティクスである。また, 正規分布の豊富かつ良好な性質を利用可能であるため, メタヒューリスティクスの中では最も理論的研究が進んでいるといえる。中でも, 自然勾配系との類似により, 共分散行列が目的関数の Hesse 行列に近似的に適応することが示されており, 近年の有力な理論となっている。国際会議 ACM GECCO にて開催された BBO のコンペティション Workshop on Real-Parameter Black-Box Optimization Benchmarking (BBOB) では, 2009 年時点では BIPOP-CMA-ES が 1 位であることが示され[48], 国際会議 IEEE CEC2018 の Competition on Real-Parameter Single Objective Optimization では CMA-ES の亜種が 1 位に位置することが数値実験的に示された[48]。以上から, CMA-ES は, 不変性や適応的パラメータ調整則, 自然勾配系との類似性を具備することで, state of the art なメタヒューリスティクスであるといえる。

5.2.3 Particle Swarm Optimization のパラメータ調整方法

先行研究において, PSO のパラメータ調整則は多く提案されている。特に, 多様化・集中化の探索戦略に基づいて設計されているものが多い。5.2.1 項の分類方法に従って, PSO

表 5.1：代表的な PSO のパラメータ調整則

パラメータ固定型	Constriction Method [176, 177]
スケジューリング的調整型	Random Inertia Weight Method [177]
	Linearly Decreasing Inertia Weight Method [177, 195]
	Linearly Decreasing Vmax Method [196]
	Increasing Neighborhood Size Method [197]
適応的調整型	Activity Feedback PSO [116]
	Evolving PSO Tuning Using Genetic Programming [158]

のパラメータ調整則に関する代表的な先行研究を分類・整理したものを表 5.1 に示す。

PSO の固定型調整則として代表的な設定方法は、Constriction Method (CM) [176, 177] が挙げられ、パラメータを $\{w, c = c_1 = c_2\} = \{0.729, 1.4955\}$ と設定する。CM は比較的多様化・集中化の探索戦略が実現しやすく、低次元では優れた性能を発揮する方法として知られている。

PSO のスケジューリング的調整型として代表的な設定方法は、Random Inertia Weight Method (RIWM) [177] や Linearly Decreasing Inertia Weight Method (LDIWM) [177, 195]、Linearly Decreasing Vmax Method (LDVM) [196]、Increasing Neighborhood Size Method (INSM) [197] が知られている。RIWM のパラメータ調整則 F_f は式 (5.21) で表される。

$$w(k+1) \sim \mathcal{U}_{\mathbb{R}}(0.5, 1.0) \quad (5.21)$$

ただし、 $c_1 = c_2 = 1.4955$ としており、 w の探索過程における平均値 0.75 である。つまり、 c_1, c_2 に CM と合わせ、 w は CM の設定を中心として、探索過程で摂動させることで、多様化・集中化の能力を向上させていく。LDIWM のパラメータ調整則 F_f は式 (5.22) で表される。

$$w(k+1) = w_{\text{start}} + \frac{k}{k_{\text{max}}} (w_{\text{end}} - w_{\text{start}}) \quad (5.22)$$

ただし、 $c_1 = c_2 = 2.0$ である。 w を大きな値から探索過程で線形に減少させていくことで、多様化から集中化へ移行させている。IDVM のパラメータ調整則 F_f は、速度上限パ

ラメータ V_{\max} を導入し、式(5.23)を用いて調整する。

$$V_{\max}(k+1) = V_{\text{start}} + \frac{k}{k_{\max}}(V_{\text{end}} - V_{\text{start}}) \quad (5.23)$$

つまり、 $\{w, c_1, c_2\}$ は大きな値に設定しておき、 V_{\max} を探索過程で線形に減少させていくことで、多様化から集中化へ移行させている。INSM では、更新則として local-best solution (l-best) $\mathbf{L}(k)$ を導入した Lbest モデルを使用する。Lbest モデルは式(5.24)、式(5.25)で表される。

$$\mathbf{x}^i(k+1) = \mathbf{x}^i(k) + w\mathbf{v}^i(k) + c_1\mathbf{R}_1(\mathbf{p}^i(k) - \mathbf{x}^i(k)) + c_2\mathbf{R}_2(\mathbf{L}^j(k) - \mathbf{x}^i(k)) \quad (5.24)$$

$$\mathbf{L}^j(k) = \underset{\mathbf{p}^i(k) \in \mathcal{L}^j(k)}{\operatorname{argmin}} \{f(\mathbf{p}^i(k)) \mid i = 1, \dots, M\} \quad (5.25)$$

ただし、グループ番号 $j = 1, \dots, J$, \mathcal{L}^j は p-best 集合 $\mathcal{P}(k)$ を J 個のグループに分割したときのグループ j の l-best 集合、 $J = \lfloor m/M \rfloor$, M は近傍パラメータである。このように、Lbest モデルは複数の群に分割し、最良解を共有する範囲を限定する。INSM のパラメータ調整則 F_f では、 $M, w, c = c_1 = c_2$ を式(5.26)、式(5.22)、 $c = c_1 = c_2$ を式(5.27)を用いて調整する。

$$M(k+1) = M_{\text{start}} + \frac{k}{k_{\max}}(M_{\text{end}} - M_{\text{start}}) \quad (5.26)$$

$$c(k+1) = c_{\text{start}} + \frac{k}{k_{\max}}(c_{\text{end}} - c_{\text{start}}) \quad (5.27)$$

$M_{\text{start}} = 3, M_{\text{end}} = m > 3$ がよく使用される。つまり、Lbest モデルから探索を開始することで、最良解の情報を多様化させておき、徐々に Gbest モデルに近づけることで探索点群全体を集中化させている。

PSO の適応的調整型のパラメータ調整則として、Activity Feedback PSO (AFPSO) [116] や、金政らの[158]の方法が知られている。AFPSO のパラメータ調整則 F_f では、 w

を式(5.28), 式(5.29), 式(5.30)を用いて調整する。

$$w(k+1) = \begin{cases} \min\{w(k) + \Delta w, w_{\max}\}, & P(k) \leq P_{\text{target}}(k) \\ \max\{w(k) - \Delta w, w_{\min}\}, & \text{otherwise} \end{cases} \quad (5.28)$$

$$P(k) = \frac{1}{m\sqrt{N}} \sum_{i=1}^m \| \mathbf{v}^i(k) \|_2 \quad (5.29)$$

$$P_{\text{target}}(k) = P_{\text{start}} \left(\frac{P_{\text{end}}}{P_{\text{start}}} \right)^{\frac{k}{k_{\max}}} \quad (5.30)$$

ただし, $\Delta w > 0$ は調整幅, w_{\max}, w_{\min} は w の上下限値, $P_{\text{start}}, P_{\text{end}} > 0$ は目標値 $P_{\text{target}}(k)$ の初期値と終端値である。特に, $P(k)$ を群の活性度 (Swarm Activity) と呼び, 多様化・集中化の評価指標として用いている⁵。AFPSO では, 式(5.30)のように, 目標値 P_{target} の指数減少スケジュール⁶を生成しておき, 活性度 $P(k)$ が目標値 P_{target} に追従するように, w を適応的に調整する。これにより, 許容される反復回数のもとで多様化・集中化の探索戦略を確実に実現する方法である。

[158]では, パラメータ調整則を構成する演算子などの集合を解とみなし, それを対象関数に対して実行したときの探索性能を得るというプロセスを複数回繰り替えず上位の最適化問題を Genetic Programming (GP) が解くことで, 使用したアルゴリズム・対象関数に対して最適なパラメータ調整則を自動的に設計する方法を用いて, PSO の適応的調整型のパラメータ調整則を獲得している。それを簡潔な関数のみで近似した式は, 式(5.31), 式(5.49)で表される。

$$c_2(k) = \exp(0.6484 - 1.025 \| \mathbf{v}^i(k) \|_2 - 1.182 \Delta_g(k) + 0.02597 / \Delta_g(k)) \quad (5.31)$$

$$\Delta_g(k) = \frac{1}{m\sqrt{N}} \sum_{i=1}^m \| \mathbf{p}^g(k) - \mathbf{x}^i(k) \|_2 \quad (5.32)$$

⁵ 活性度は熱・統計力学分野における Maxwell–Boltzmann の速度分布則に着想を得ており, 系の温度が高ければ気体分子の運動エネルギーも高く, 温度が低ければ運動エネルギーも低いことに対応させて解釈している。また, Simulated Annealing における温度パラメータが高いと探索点が解空間の広い範囲を動き, 温度パラメータが低いと範囲が狭くなることにも対応させている。

⁶ [116]では, 線形減少スケジュールも実験しているが, 指数減少スケジュールのほうが優れていることを確認している。

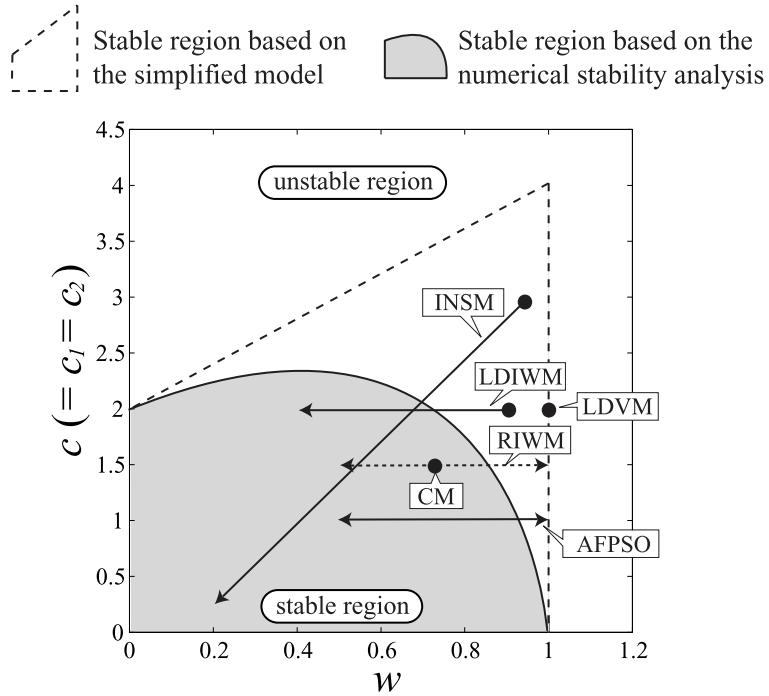


図 5.1：PSO のパラメータ空間における安定・不安定領域[116]

ただし, $w = 0.769, c_1 = 1.752$, $\Delta_g(k)$ は探索点と g-best の間の平均 L^2 ノルムの平均である。探索点群が密集していない状態では ($\Delta_g(k)$ 大), $c_2(k) = 0.0$ となり, w, c_1 による慣性と p-best への差分ベクトルで駆動する（集中化）ことに対して、探索点群が密集している状態では ($\Delta_g(k)$ 小), $0.02597/\Delta_g(k)$ の項が大きくなり、指標関数で增幅され、結果として発散状態（多様化）へ移行することがわかる。なお、式(5.31)は人間が設計した調整則ではなく、GP によって特定の関数に対して優れた性能を出すように自動的に設計された調整則であることに注意されたい。

さらに, RIWM, LDIWM, LDVM, INSM のパラメータ調整則は、多様化・集中化の探索戦略の実現を目指す点で共通している一方、数値実験のみにより有効性が主張されているが、その有効性に対する理論的な根拠が明確にされていなかった。3.6 節で述べたように、PSO の探索ダイナミクスを解析する方法として、縮約モデルが知られており、PSO のパラメータ空間 $\{w, c_1 = c_2\}$ 内に安定・不安定領域を理論的に導出できる。しかしながら、縮約モデルは近似や仮定が多く含まれる簡易なモデルであるため、その安定・不安定領域は正確なものではなかった。そこで、Yasuda らは、活性度 $P(k)$ を安定性の評価指標

として使用することで、数値実験的に安定・不安定領域を描画できることを示した[116]。その安定・不安定領域を図5.1に示す。Yasudaらは、図5.1に上記のパラメータ調整方法を対応させることで、下記のように、より正確な多様化・集中化の実現度合いを確認できることを明らかにした。

CM： パラメータを弱い安定領域 ($\{w, c = c_1 = c_2\} = \{0.729, 1.4955\}$) に設定する。

RIWM： パラメータを安定・不安定領域の境界線 ($\{w, c = c_1 = c_2\} \in (\{0.5, 1.4955\}, \{1.0, 1.4955\})$) を挟むように、ランダムに設定する。

LDIWM： パラメータを不安定領域 ($\{w, c = c_1 = c_2\} = \{0.9, 2.0\}$) から安定領域 ($\{w, c = c_1 = c_2\} = \{0.4, 2.0\}$) へ線形に減少させる。

LDVW： パラメータ $\{w, c = c_1 = c_2\}$ を不安定領域に設定し、 V_{\max} を線形に減少させる。

INSM： 各 p-best の近傍パラメータ M を探索過程で増加させ、 $\{w, c = c_1 = c_2\}$ を減少させる。

さらに、Yasudaらは、CMが高次元において性能が劣化する理由が、安定・不安定領域の境界線が高次元になると安定領域側にやや小さくなり、CMの位置が弱い安定領域から不安定領域に近づくためであることを明らかにした。AFPSOでは、上記の解析結果に従い、 w_{\max} を不安定領域に、 w_{\min} を安定領域に設定することで、多様化・集中化の探索戦略を実現できる。

5.2.4 Differential Evolution のパラメータ調整方法

DEはパラメータとして、突然変異パラメータ F 、交叉率 CR を有している。しかしながら、PSOやESと比べて、DEはオフラインチューニング（固定型）と探索性能の関係が、探索条件や問題の性質に強く依存し、明確な関係がないことが知られている[206]。したがって、DEのパラメータ調整・設定方法は、オフラインチューニングで有効な方法が開発されておらず、オンラインチューニングの方法が中心に開発されている。そこで、本節ではDEのスケジューリング的調整型と適応的調整型のパラメータ調整則を概観する。

DEのスケジューリング的調整型として代表的な設定方法は、Self-Adaptive DE (SaDE)

[207] や Bimodal Distribution Parameter Setting DE (BiDE) [208] が知られている。SaDE では、各探索点 $\mathbf{x}^i(k)$ に異なるパラメータ F^i を割り当て、正規分布に従いランダムに生成する。SaDE のパラメータ調整則は $F^i \sim \mathcal{N}(0.5, 0.3)$ である。BiDE では、各探索点 $\mathbf{x}^i(k)$ に異なるパラメータ F^i と C^i を割り当て、各パラメータを二峰性の Cauchy 分布に従いランダムに生成する。BiDE のパラメータ調整則は式 (5.50), 式 (5.51) で表される。

$$F^i = \begin{cases} r_c(0.65, 0.1), & \theta < 0.5 \\ r_c(1.0, 0.1), & \text{otherwise} \end{cases} \quad (5.33)$$

$$C^i = \begin{cases} r_c(0.1, 0.1), & \theta < 0.5 \\ r_c(0.95, 0.1), & \text{otherwise} \end{cases} \quad (5.34)$$

ここで、 $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$, $r_C(\mu, \gamma) \sim \mathcal{C}(\mu, \gamma)$ は位置パラメータ μ と尺度パラメータ γ の Cauchy 分布 (式 (A.21)) に従う乱数である。このように、DE のスケジューリング的調整型のパラメータ調整則は、PSO の IWM と同様に、探索過程で一様にランダムに与えるものが多い。

DE の適応的調整型として代表的な設定方法は、jDE [117]⁷ や JADE [118]⁸, Success-History Based Adaptive DE (SHADE) [119], Linear Population Size Reduction Based SHADE (L-SHADE) [209], Feedback-Controlled DE (FCDE) [?]など、多くの方法が知られている。jDE のパラメータ調整則は式 (5.52), 式 (5.53), 式 (5.55) で表される。

$$\hat{F}^i(k) = \begin{cases} r_F, & \theta < \tau_F \\ F^i(k), & \text{otherwise} \end{cases} \quad (5.35)$$

$$\hat{C}^i(k) = \begin{cases} r_C, & \theta < \tau_C \\ C^i(k), & \text{otherwise} \end{cases} \quad (5.36)$$

⁷ jDE は原著論文では単に「Adaptive DE」と呼称されていたが、それ以降の多くの適応型 DE と区別するために、jDE という呼称が使用されたことがきっかけでその呼称が定着している。jDE の「j」は、第一著者 (Janez Brest) のファーストネームの頭文字であると思われる。

⁸ JADE の「JA」は、提案者である Jingqiao Zhang と Arthur C. Sanderson のファーストネームの頭文字を合わせたものであると考えられる。

$$\{F^i(k+1), C^i(k+1)\} = \begin{cases} \{\hat{F}^i(k), \hat{C}^i(k)\}, & f(\hat{\mathbf{x}}^i(k)) \leq f(\mathbf{x}^i(k)) \\ \{F^i(k), C^i(k)\}, & \text{otherwise} \end{cases} \quad (5.37)$$

ただし, $r_F \sim \mathcal{U}_{\mathbb{R}}(0.1, 1)$, $\theta, r_C \sim \mathcal{U}_{\mathbb{R}}(0, 1)$, $\{F^i(1), C^i(1)\} = \{0.5, 0.9\}$ である。jDE では, 各探索点 $\mathbf{x}^i(k)$ に異なるパラメータ F^i と C^i を割り当て, 各反復でランダムに与える一方で, 探索点の更新が成功したら, F^i と C^i も更新する方法である。jDE を始めとする, DE のパラメータ調整則では, 探索点だけではなく, パラメータにも進化計算特有の選択を与えることで, 探索過程で優れたパラメータを「進化」させる方法が多い。jDE は 2006 年に提案されたパラメータ調整則だが, 簡潔でありながら優れた性能を示すことから, 近年の適応型 DE の標準的な方法として比較対象などで使用されることが多い。JADE の更新則では, 突然変異として式 (3.30) ではなく, 式 (5.38) (current-to-pbest) が使用される。

$$\mathbf{u}^i(k) = \mathbf{x}^i(k) + F(\mathbf{x}^{better}(k) - \mathbf{x}^i(k)) + F(\mathbf{x}^{r_1}(k) - \mathbf{y}(k)) \quad (5.38)$$

$\mathbf{x}^{better}(k)$ は探索点群 $\mathcal{X}(k)$ の目的関数値に基づくランクの上位 ℓ 個からランダムに選択した個体である。ただし, $\ell = \max(\lfloor m \times p \rfloor, 2)$, $p \in [0, 1]$ である。 $\mathbf{y}(k)$ は, 集合 $\mathcal{X}(k) \cup A(k)$ からランダムに選択した個体である。ただし, アーカイブ $A(k)$ は式 (3.32) の生存選択にて, 探索点 $\mathbf{x}^i(k)$ が改善に成功し, $\mathbf{x}^i(k+1) = \hat{\mathbf{x}}^i(k)$ と移動したとき, $\mathbf{x}^i(k)$ を $A(k)$ に追加していく。 $A(k)$ のサイズ s_A をパラメータとし, $\text{card}(A(k)) > s_A$ の場合, $A(k)$ の要素を $\text{card}(A(k)) = s_A$ となるようにランダムに除外する。JADE のパラメータ調整則では, 各探索点 $\mathbf{x}^i(k)$ に異なるパラメータ $F^i(k)$ と $C^i(k)$ を割り当て, $F^i(k) \sim \mathcal{C}(\mu_F(k), 0.1)$ と $C^i(k) \sim \mathcal{N}(\mu_C(k), 0.1)$ のようにランダムに与える。ただし, $F^i(k) > 1$ となった場合, $F^i(k) = 1$ とし, $F^i(k) \leq 0$ となった場合は再度生成し, $C^i(k) > 1$ となった場合, $C^i(k) = 1$ とし, $C^i(k) < 0$ となった場合, $C^i(k) = 0$ とする。各探索点の F が小さな値に集中すると, 探索点群が初期収束しやすいため, ヘヴィー・テイル型の Cauchy 分布 (式 (A.21)) に従うことでなるべく多様な F を生成することで, 初期収束を防ぐ意図がある。追加されたパラメータ μ_F と μ_C は探索過程で調整され, その調整則は式 (5.39), 式 (5.40) で表される。

$$\mu_F(k+1) = (1 - c)\mu_F(k) + c\mu(S^F) \quad (5.39)$$

$$\mu_C(k+1) = (1 - c)\mu_C(k) + c\mu_L(S^C) \quad (5.40)$$

ただし, $c \in [0, 1]$ はパラメータであり, $\mu_F(1), \mu_C(1) = 0.5$ である。 $S^F(k), S^C(k)$ は探索点群 $\mathcal{X}(k)$ を更新したときに改善した探索点 $\mathbf{x}^i(k)$ に対応する $F^i(k), C^i(k)$ の集合であり, 各反復でリセットされる。 μ_L は階数 2 の Lehmer 平均 (非調和平均) であり, 集合 S に対する階数 $j \in \mathbb{N}$ の Lehmer 平均は $\mu_L(S) = \sum_{F \in S} F^j / \sum_{F \in S} F^{j-1}$ によって計算される。集合の高い値に偏りやすい Lehmer 平均を用いることで, F が小さな値に集中することを防いでいる。このように, JADE では, 探索過程で改善に成功したパラメータ集合に徐々に近づくことで, その問題に特化したパラメータに調整している。SHADE のパラメータ調整則では, 各探索点 $\mathbf{x}^i(k)$ に異なるパラメータ $F^i(k)$ と $C^i(k)$ を割り当て, $F^i(k), C^i(k)$ を大きさ H のメモリ $\mathbf{m}^F(k), \mathbf{m}^C(k) \in \mathbb{R}^H$ を用いて更新する。 $\mathbf{m}^F(1), \mathbf{m}^C(1) = 0.5 \times \mathbf{1}$ とする。 $F^i(k) \sim \mathcal{C}(m_{r_i}^F(k), 0.1)$ と $C^i(k) \sim \mathcal{N}(m_{r_i}^C(k), 0.1)$ のようにランダムに与える。ただし, $r_i \sim \mathcal{U}_{\mathbb{Z}}(1, H)$ である。 $F^i(k), C^i(k)$ が $[0, 1]$ を超えた場合, JADE と同様の処理により $[0, 1]$ に収める。追加されたメモリパラメータ \mathbf{m}^F と \mathbf{m}^C は探索過程で調整され, その調整則は式 (5.41), 式 (5.42) で表される。

$$m_j^F(k+1) = \mu_L(S^F(k)) \quad (5.41)$$

$$m_j^C(k+1) = \mu_L(S^C(k)) \quad (5.42)$$

ただし, $S^F(k), S^C(k)$ は探索点群 $\mathcal{X}(k)$ を更新したときに改善した探索点 $\mathbf{x}^i(k)$ に対応する $F^i(k), C^i(k)$ の集合であり, μ_L は階数 2 の Lehmer 平均である。反復回数 k において, $(h-1) \times H + 1 < k \leq h \times H$ を満たす $h \in \mathbb{N}$ を求め, $j = \min\{k - (h-1) \times H, k_{\max}\}$ とする。JADE では, パラメータ $\mu_F(k), \mu_C(k)$ が改善に成功した F, C に収束しやすいが, 理想的とは限らない一方で, SHADE では, メモリパラメータ $\boldsymbol{\mu}^F(k), \boldsymbol{\mu}^C(k)$ の一要素だけを更新することで, 理想とは大きく異なる F, C が改善に成功し, S^F, S^C に含まれたとしても, 望ましくない値に集中する効果を緩和している。SHADE は, BBOB のベンチマーク問題集を用いて評価したところ, jDE や JADE などの代表的な適応型 DE に比べて, 多くの条件で優れていたという報告されている[119]。L-SHADE は決定的集団数減少法 (Linear Population Size Reduction : LPSR) を用いた SHADE であり, その調整則は式 (5.43) で表される。

$$m(k+1) = \lfloor (m_{\text{end}} - m_{\text{start}}) \frac{T}{T_{\text{end}}} + -m_{\text{start}} \rfloor \quad (5.43)$$

$m_{\text{start}}, m_{\text{end}}$ は探索開始時と探索終了時の探索点数である。SHADE に加えて、評価回数 T に応じて、 $m(k)$ がスケジューリング的に減少していく調整則が追加されている。FCDE では、 F の調整則は式 (5.44), 式 (5.45), 式 (5.46), 式 (5.47), 式 (5.48) で表される。

$$F(k+1) = F(k) + K_F F(k) D(k) \quad (5.44)$$

$$CR(k+1) = CR(k) + K_{CR} CR(k) D(k) \quad (5.45)$$

$$D(k) = I_{\text{target}}(k) - I(k) \quad (5.46)$$

$$I(k) = \frac{\sum_{i=1}^m \| \mathbf{x}^i(k) - \mathbf{x}^b(k) \|_1}{\sum_{i=1}^m \| \mathbf{x}^i(1) - \mathbf{x}^b(1) \|_1} \quad (5.47)$$

$$I_{\text{target}}(k) = I_{\text{start}} \left(\frac{I_{\text{end}}}{I_{\text{start}}} \right)^{\frac{k}{k_{\text{max}}}} \quad (5.48)$$

ただし、 $I_{\text{start}} = 1$ であり、 $F(k) \in [F_{\min}, F_{\max}], CR(k) \in [CR_{\min}, CR_{\max}]$ となるように、収める。さらに、ゲインパラメータ K_F, K_{CR} は $1/K_F \geq h, 1/K_{CR} \geq h, -1/K_F + 1/K_{CR} \geq 2h$ を満たすように設定すれば、上記のフィードバックによって追従できることが示されている。 $h > 0$ は偏差 $D(k)$ の推定可動範囲を意味するパラメータである。このように、FCDE は目標値生成器、コントローラ、評価器に基づくパラメータ調整則を構成し、DE に陽なフィードバック制御を行うことで、多様化・集中化に基づく探索戦略を確実に実現する。

国際会議 IEEE CEC にて開催された CEC2014 Competition on Real-Parameter Single Objective Optimization (RPSO) では L-SHADE が 1 位、CEC2015 の RPSO においても L-SHADE に基づくアルゴリズムが 1, 2, 3 位、CEC2016 の RPSO においても L-SHADE に基づくアルゴリズムが 1 位、CEC2018 Competition on Large-Scale Global Optimization (LSGO) においても SHADE に基づくアルゴリズムが 1 位、CEC2019 の LSGO においても SHADE に基づくアルゴリズムが 1 位を占めている[48]。また、SHADE と L-SHADE は、BIPOP-CMA-ES に匹敵する性能を具備することが数値実験的に示されている[48]。以上から、DE の適応型パラメータ調整則の設計は難しい一方で、DE の最適化能力のポテンシャルは高く、適応的調整型のパラメータ調整則の付加によって、メタヒューリスティクスの中でも最も優れている部類に入るといえる。

5.2.5 Cuckoo Search のパラメータ調整方法

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。

これまでに CS のパラメータ調整方法はいくつか提案されている[199, 200, 201]。5.2.1 項の分類方法に従って、PSO のパラメータ調整則に関する代表的な先行研究を分類・整理する。

スケジューリング的調整則として、[199]の方法がある。[199]のパラメータ調整則は、式(5.49)、式(5.50)で表される。

$$\alpha = \alpha_{\max} \left(\frac{\alpha_{\min}}{\alpha_{\max}} \right)^{\frac{T}{T_{\max}}} \quad (5.49)$$

$$P_a = P_{a,\max} - \frac{T}{T_{\max}} (P_{a,\max} - P_{a,\min}) \quad (5.50)$$

これは、探索過程で α を指数的に減少させていくと同時に、 P_a を線形に現象させていく。

適応的調整則として、[200]や[201]の方法がある。[200]のパラメータ調整則は、式(5.51)で表される。

$$\alpha^i = \begin{cases} \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \frac{f(\mathbf{x}^i) - f(\mathbf{x}^b)}{f_{\text{ave}} - f(\mathbf{x}^b)}, & f(\mathbf{x}^i) \leq f_{\text{ave}} \\ \frac{\alpha_{\max}}{\sqrt{T}}, & \text{otherwise} \end{cases} \quad (5.51)$$

ただし、 f_{ave} は探索点群の評価値集合 \mathcal{F} の平均値である。よって、探索点毎に α^i を割り当てる多様性を増加させた上で、探索点群内で平均的に優れているなら、優れている度合いが高いほど α^i を小さく、平均に近いと α^i を多少大きくすることに対して、平均的に優れていなければ、探索過程で α^i を指数的に減少させていく。[201]のパラメータ調整則は、式(5.52)で表される。

$$\alpha^i = \left(\frac{1}{T} \right)^{\frac{|f(\mathbf{x}^i) - f(\mathbf{x}^b)|}{|f(\mathbf{x}^w) - f(\mathbf{x}^b)|}}, \quad (5.52)$$

よって、探索点毎に α^i を割り当てる多様性を増加させた上で、探索過程で α^i を指数的に減少させていくが、探索点群内で優れているほど、指数部分が 0、 $\alpha^i \sim 1$ となり、劣っているほど、指数部分が 1、 $\alpha^i \sim 1/T$ となる。

どのパラメータ調整方法も、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、探索戦略の実現を目指している。さらに、[200]や[201]の方法は、探索点群における探索点 x^i の良さに応じて α^i を調整するため、自律適応的な方法である。ただし、パラメータ調整則で使用する量としては、「探索時間の経過」と「探索点群内の相対的な良さ」に限られている。

5.2.6 変換不变性に基づくパラメータ調整方法の分類

(a) パラメータ調整則の変換不变性の定義

これまで、パラメータ調整則と変換不变性については独立的に述べてきた。第4章では、変換不变性は、アルゴリズムのロバスト性と直結する性質であり、アルゴリズムに欠如している変換不变性を具備させることで、探索性能のロバスト性向上が可能であることを述べた。一方、パラメータ調整則は、アルゴリズムの適応性を向上させる機能であり、アルゴリズムに付加し、有機的に連携させることで、探索性能を引き出すことが可能である。代表的なパラメータ調整則の中でも不变性を満たしていない例があることから、その設計指針を与えることは重要である。本章では、これらのパラメータ調整則と変換不变性が独立的ではなく、密接な関係にあることを指摘し、高いロバスト性・適応性を有するアルゴリズム設計を目指すためには、それらの相互関係を把握する必要があることを示す。

パラメータ調整則は、探索過程でパラメータを動的に変化させることで、パラメータの効果により探索ダイナミクスを変化させる方法だが、更新則の変換不变性に影響を与える。パラメータ調整則と変換不变性の相互関係が発生するケースは、下記の二点である。

ケース(1) 更新則は変換不变性を有しているが、パラメータ調整則自身はその不变性を欠如している場合

ケース(2) 更新則が変換不变性を欠如しているが、パラメータ調整則の付加によって、その不变性を補償する場合

ケース(1)は、アルゴリズムの更新則がすでにある変換不变性を有する場合、その不变性を維持するには、付加するパラメータ調整則もその変換に対して不变である必要があることを表す。裏を返せば、パラメータ調整則が不变でなければ、アルゴリズム全体としての

変換不変性も失うことを意味する。ケース(2)は、アルゴリズムがある変換不変性を失っている場合、パラメータ調整則による影響によってその不変性を補完することも可能であることを表す。よって、アルゴリズム全体の変換不変性を獲得するには、アルゴリズム自身の更新則とパラメータ調整則の相互の影響を考慮しながら、設計する必要がある。これらのケースを踏まえると、変換不変性に基づくパラメータ調整則の設計指針として、下記の二点が考えられる。

指針(1) 更新則は変換不変性を具備するとき、その変換不変性を維持するように、その変換不変性を有するパラメータ調整則を付加する

指針(2) 更新則が変換不変性を欠如するとき、その変換不変性を補償するように、その変換不変性に影響を与えるパラメータ調整則を付加する

本章では、これらの変換不変性に基づく設計指針に従い、適応的調整型のパラメータ調整則を設計することで、高いロバスト性・適応性を有するメタヒューリスティクスを開発する。

以降では、一般的なパラメータ調整則の数理的構造を示し、パラメータ調整則が変換不変性を具備するための条件を導出する。アルゴリズムの更新則がある変換 T に対して不変性を有する場合、アルゴリズム全体の不変性維持という要求から、パラメータ調整則に対して数理的構造の条件が与えられる。更新則自身は、ある変換 T に対する不変性を有すると仮定したため、動的に変化するパラメータ α も変換 T に対して不変である必要がある。これを「パラメータの不変性」と呼ぶこととし、定義 5.2 として定義する。

定義 5.2 パラメータ調整則 $F_{f,S}$ が変換不変性を有するとは、ある変換 T_s, T_f に対して、パラメータ調整則 F が式(5.53)、あるいは式(5.54)の条件を満たすことである。

$$F_{f,S'}(\alpha(k); \chi_{S'}(k)) = F_{f,S}(\alpha(k); \chi_S(k)) \quad (5.53)$$

$$F_{T_f \circ f, S}(\alpha(k); \chi_S(k)) = F_{f,S}(\alpha(k); \chi_S(k)) \quad (5.54)$$

ここで、左辺を $\hat{\alpha}(k+1)$ とおくと、これは $\hat{\alpha}(k+1) = \alpha(k+1) \Leftrightarrow \hat{\alpha}(k) = \alpha(k)$ となる。つまり、式(5.53)あるいは式(5.54)は変換 T_s, T_f に対して α が不変であることを表してい

る。加えて、パラメータの不变性を満たすための調整則 F の数理的構造の条件を導出する。調整則 F は、式 (5.53) を満たすとき不变となるから、そのためには、式 (5.55) の成立が十分条件となる。

$$T \circ H(k) = H(k) \quad (5.55)$$

言い換えると、 T に対して α が不变である必要十分条件は、 F に含まれる量 $H(k)$ について、式 (5.55) が成立することである。

(b) パラメータ調整則で使用する変数と変換不变性の関係

以降では、パラメータ調整則と変換不变性の関係が密接であることを、既存のパラメータ調整則を例示して指摘する。まず、パラメータ調整則 F_f において一般的な変数 $H(k)$ として用いられる情報と、変換不变性の関係について整理する。

前項まで既存のパラメータ調整則を概観したように、パラメータ調整則では探索点の位置ベクトル、経過時間の割合、探索点の更新割合、目的関数値などの量が使用されることが多い。式は、要素と演算の集合 $\{+, -, \times, /\}$ 、が組み合わさって構成される。また、If-then ルールを用いて複数のパラメータ調整則を使い分ける場合もあり、その分岐条件として大小関係が用いられる。

探索点群の状態： 探索点群の状態変数として、探索点や解の位置情報が用いられる。特に、それらの差分ベクトルのノルムがよく用いられる。これは、平行移動不变性を保つが、スケール不变でなくなる。さらに、その中でもよく使用されるノルムは、 L^2 ノルム（ユークリッド距離）である。ただし、差分ベクトルのマラハラノビス距離はアフィン変換に対して不变な量である。

探索の経過時間： 探索の経過時間の状態変数として、経過の割合が用いられる。その中でも、 k/k_{\max} のような経過の割合は、最もよく使用される。これは、アフィン変換や目的関数の単調増加変換に対して不变である。

各探索点の目的関数値の改善 各探索点の目的関数値の改善の状態変数として、各探索点の更新割合（改善頻度）がよく用いられる。これは、アフィン変換や目的関数の単調増加変換に対して不变である。

複数探索点の目的関数値： 複数探索点の目的関数値を用いる場合である。目的関数値の

大小比較のような、探索点や解同士の優劣やランク（全順序関係）だけを情報として用いる場合は、目的関数値そのものではなく、順序関係のみがパラメータ調整則に影響する。これは、アフィン変換や目的関数の単調増加変換に対して不変である。一方で、GA や ABC のルーレット選択、Natural Evolution Strategy [202] のような、直接目的関数値をパラメータ調整則に使用する場合、アフィン変換に対しては不変だが、目的関数の単調増加変換に対して不変でない。

1つ目の探索点群の状態について確認する。例えば、探索点が探索過程で得た解の差分ベクトル $\mathbf{x} - \mathbf{y}$ の L^2 ノルム $\|\mathbf{x} - \mathbf{y}\|_2$ を変数 $h \in H(k)$ とするパラメータ調整則 $F_{f,S}$ を考える。相似変換を与えると、

$$\begin{aligned} h' &= \|\mathbf{x}' - \mathbf{y}'\|_2 \\ &= \|\delta\mathbf{x} - \delta\mathbf{y}\|_2 \\ &= \|\delta(\mathbf{x} - \mathbf{y})\|_2 \\ &= \delta\|\mathbf{x} - \mathbf{y}\|_2 \\ &= \delta h \end{aligned}$$

となり、不变性の条件を満たさず、パラメータ調整に影響を与える。また、回転変換を与えると、

$$\begin{aligned} h' &= \|\mathbf{x}' - \mathbf{y}'\|_2 \\ &= \|\mathbf{C}\mathbf{x} - \mathbf{C}\mathbf{y}\|_2 \\ &= \|\mathbf{C}(\mathbf{x} - \mathbf{y})\|_2 \\ &= \|\mathbf{x} - \mathbf{y}\|_2 \\ &= h \end{aligned}$$

となり、不变性の条件を満たし、パラメータ調整に影響を与えない。これは L^2 ノルムと直交行列の性質を利用している。よって、 L^2 ノルムは平行移動不变性と回転不变性を有するが、それ以外のノルムは平行移動不变性しか有していない。ただし、大小関係でしか使用しない場合に限り、相似変換不变性も有する。

2つ目と3つ目は自明である。4つ目の複数探索点の目的関数値について確認する。例えば、探索点 \mathbf{x}^i の目的関数值 $f(\mathbf{x}^i)$ を変数 $h \in H(k)$ とするパラメータ調整則 $F_{f,S}$ を考え

る。目的関数の単調増加変換を与えたとき、

$$\begin{aligned} h' &= f'(\mathbf{x}^i) \\ &= g \circ f(\mathbf{x}^i) \\ &= g \circ h \end{aligned}$$

となり、不变性の条件を満たさない。ただし、目的関数値を不等式でしか使用しなければ、

$$\begin{aligned} h^{1'} &\leq h^{2'} \\ \Leftrightarrow f'(\mathbf{x}^1) &\leq f'(\mathbf{x}^2) \\ \Leftrightarrow g \circ f(\mathbf{x}^1) &\leq g \circ f(\mathbf{x}^2) \\ \Leftrightarrow f(\mathbf{x}^1) &\leq f(\mathbf{x}^2) \\ \Leftrightarrow h^1 &\leq h^2 \end{aligned}$$

となり、不变となり、パラメータ調整に影響を与えない。

以上をまとめると、パラメータ調整則に(2)と(3)の情報を用いるときは変換不变性を満たすが、(1)と(4)の情報を用いるときは変換不变性を常に満たさないことを確認した。具体的に満たさないケースは下記の通りである。

- (1)の情報として、解の位置ベクトルに関する L^2 ノルムを用いる場合はスケール不变でない
- (1)の情報として、解の位置ベクトルに関する L^1 ノルムを用いる場合はスケール不变でなく、かつ回転不变でない
- (4)の情報として、目的関数値情報を直接用いる場合は目的関数の単調増加変換に対して不变でない

さらに、下記のいずれかの条件を満たす要素で構成されるとき、解空間のアフィン変換あるいは、目的関数の単調増加変換に対するパラメータ不变性を満たす。

- (1) 全ての解空間や目的関数の変換とは無関係な量：経過時間の割合、探索点の更新割合
- (2) 解空間のアフィン変換に対して不变な量：位置ベクトル間のマハラノビス距離
- (3) 目的関数の単調増加変換に対して不变な量：目的関数値の全順序関係

以降では、これらの量が不変性を具備することを具体的に確認する。

(1)に分類される量として、「経過時間の割合 k/k_{\max} 」、「探索点の更新割合 $p(k)$ 」が挙げられる。これを導入したパラメータ調整則の例としては、 $\alpha(k) = (\alpha_{\text{start}} - \alpha_{\text{end}})(1 - \frac{k}{k_{\max}}) + \alpha_{\text{end}}$, $\alpha(k) = \alpha_{\text{start}}(\frac{\alpha_{\text{end}}}{\alpha_{\text{start}}})^{\frac{k}{k_{\max}}}$, ES の 1/5 ルールなどが挙げられる。元々変換とは無関係であるため、不変となることは自明である。

(2)に分類される量として、「位置ベクトル間のマハラノビス距離 $d_M(\mathbf{x}, \mathbf{y})$ 」が挙げられる。これを導入したパラメータ調整則の例としては、AFPSO の活性度（式(5.29)）を $I(k) = \frac{1}{m\sqrt{(N)}} \sum d_M(\mathbf{x}, \mathbf{y})$ とおいて、式(5.28), 式(5.30)を用いるなどが挙げられる。他にも、式(5.31)を簡略化した、 $\alpha(k+1) = \exp(\frac{1}{I(k)} - \frac{I(k)}{c^2})$ や、 $\alpha(k+1) = \alpha(k)\sqrt{\frac{D(k)}{I(k)}}$ も考えられる。ただし、 $D(k)$ は評価値上位の探索点に限定した評価指標である。マハラノビス距離 $d_M(\mathbf{x}, \mathbf{y})$ にアフィン変換を与えると、

$$\begin{aligned} d_M^2(\mathbf{x}, \mathbf{y}) &= (\mathbf{x}' - \mathbf{y}')^T \Sigma^{-1} (\mathbf{x}' - \mathbf{y}') \\ &= (\mathbf{C}\mathbf{x} - \mathbf{C}\mathbf{y})^T (\mathbf{C}\Sigma\mathbf{C}^T)^{-1} (\mathbf{C}\mathbf{x} - \mathbf{C}\mathbf{y}) \\ &= (\mathbf{C}(\mathbf{x} - \mathbf{y}))^T (\mathbf{C}^T)^{-1} \Sigma^{-1} \mathbf{C}^{-1} \mathbf{C}(\mathbf{x} - \mathbf{y}) \\ &= (\mathbf{x} - \mathbf{y})^T \mathbf{C}^T (\mathbf{C}^T)^{-1} \Sigma^{-1} (\mathbf{x} - \mathbf{y}) \\ &= (\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y}) \end{aligned} \quad (5.56)$$

$$d_M^2(\mathbf{x}, \mathbf{y}) \Leftrightarrow d'_M(\mathbf{x}, \mathbf{y}) = d_M(\mathbf{x}, \mathbf{y}) \quad (5.57)$$

となり、マハラノビス距離はアフィン変換不変性を有する。

(3)に分類される量として、「探索点 \mathbf{x}^i のランク」が挙げられる。これを導入したパラメータ調整則の例としては、 $\alpha_i(k+1) = (\alpha_{\max} - \alpha_{\min})\frac{R_{i-1}}{m-1} + \alpha_{\min}$ などが考えられる。ただし、 $R_i \in [1, m]$ は探索点群の中で各探索点を $f(\mathbf{x}^1) < f(\mathbf{x}^2) < \dots < f(\mathbf{x}^m)$ となるように並び替えたときの探索点 \mathbf{x}^i のランクである。不変性の成立は上記で示した通りである。

以上の分類・整理から、その設計指針として、パラメータ調整則 $F_{f,S}$ の中で、これらのいずれかの条件を満たす要素のみで構成すれば、パラメータの不変性を満たすことがわかる。そして、パラメータの不変性を満たすようにパラメータ調整則 $F_{f,S}$ を設計しておけば、 $F_{f,S}$ を有するアルゴリズム全体の不変性にも影響を与えずに設計することが可能となる。

以上から、パラメータ調整則と変換不変性が密接な関係にあり、その相互関係を把握しながら使い分けることは、アルゴリズム設計において極めて重要であることがわかる。パ

パラメータ調整則の指針(1)「更新則は変換不变性を具備するとき、その変換不变性を維持するように、その変換不变性を有するパラメータ調整則を付加する」に従う場合、不变性を有する（影響を与えない）調整則を用いる。これは、経過時間の割合、探索点の更新割合、位置ベクトル間のマハラノビス距離、目的関数値の全順序関係などを用いればよい。ただし、 L^1 ノルムや L^2 ノルムを用いてしまうと、アルゴリズムのスケール不变性や回転不变性を失う、目的関数値を直接用いてしまうと、アルゴリズムの目的関数の単調増加変換に対する不变性を失う、などに注意する必要がある。例えば、AFPSOのパラメータ調整則が代表的である。[5.4](#)節で指摘するが、この調整則は相似変換・平行移動・回転変換に対して不变だが、スケール変換に対して不变でない。一方、パラメータ調整則の指針(2)「更新則が変換不变性を欠如するとき、その変換不变性を補償するように、その変換不变性に影響を与えるパラメータ調整則を付加する」に従う場合、不变性を有さない（影響を与える）調整則を用いる。これは、 L^2 ノルムを適切に使用することで、アルゴリズムの相似変換不变性を補完することが考えられる。例えば、ESの1/5ルールが代表的である。ESは正規分布の標準偏差に相当するパラメータとして σ を備えているが、解空間の相似変換を与えたとき、その変換に応じて適切な値に設定しなければ、探索性能が大きく変動してしまう（相似変換不变でない）。この原因は、パラメータ σ が固定型調整型に従っていることである、として捉えられる。このため、パラメータ σ を探索状況に応じて適応的に調整することで、相似変換に対するロバスト性を補完することが可能である。この影響で、多くのESに関する先行研究は、1/5ルールといった、 σ の調整則に関するものである。以上のように、パラメータ調整則がアルゴリズムの変換不变性を補償することが可能である。ただし、これは補完可能な不变性がかなり限定されており、汎用なアプローチではない。

このように、パラメータ調整則とアルゴリズムの変換不变性の関係は密接な関係にあると同時に、パラメータ調整則で用いる情報を適切に選ぶことで、アルゴリズムの変換不变性に対する補完や保持が行えることを確認した。したがって、本論文における、パラメータ調整則の役割を下記のように設定し、アルゴリズム設計に活かす。

- 主な役割は、アルゴリズムの適応性の向上のための機能である
- サブの役割は、アルゴリズムの変換不变性を補完する機能である
- ただし、アルゴリズムが元々有している変換不变性に影響を与えないように注意する

5.3 相似変換不变性を補完する適応型 Cuckoo Search

5.3.1 背景

5.2 節で述べたように、これまでに CS のパラメータ調整則は多く検討されてきた。一方、4.3 節で指摘したように、CS は回転不变性と相似変換不变性（スケール不变性）を欠如しているため、4.6 節ではアフィン変換不变性を有する CS として新たに CMD-CS を提案すると同時に、アフィン変換に対するロバスト性が向上することと、全般的な性能がパラメータに依存するため、適応性に課題があることを確認した。

そこで、本節では、BBO での適用を想定し、高いロバスト性・適応性を有する CS として、変換不变性を有する適応型 CS を提案する。具体的には、5.2 節の設計指針(2)に従い、CS に対して、相似変換不变性を補完するように適応的調整型のパラメータ調整則を付加する。先行研究では、①4.2 節で概観したように、変換不变性に関する CS の研究はない、②5.2 節で概観したように、いくつかパラメータ調整則を有する CS を提案しているが、変換不变性を観点で行われていない、ということに対して、本節の提案手法は、変換不变性と適応的調整型のパラメータ調整則を同時に有する CS に分類される点で、新しい。

5.3.2 項では、CS のパラメータ解析を行い、パラメータと多様化・集中化の関係を明らかにする。5.3.3 項では、相似変換不变性を補完した適応型 CS を提案する。5.3.4 項では、提案手法の変換不变性の解析を行う。5.3.5 項では、数値実験を通じて、提案手法のロバスト性・適応性を検証する。

5.3.2 Cuckoo Search のパラメータ解析

(a) Lévy 乱数の解析

本項では、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。

CS の Global Search (GS) は、各探索点自身に乱数ベクトルに基づく摂動を与える操作、CS の Local Search (LS) は、確率的に、各探索点自身に探索点間の差分ベクトルに基づく摂動を与える操作である。表 3.1 で定義したように、多様化・集中化の実現状態は、

① 良い解の活用と、② 摂動・探索点分布、の調整により変化する。この定義に基づくと、LS・GSでは、探索点の更新において改善移動のみにおいて、良い解を活用しており、LSは摂動の大きさを差分ベクトルによって調整されるが、GSは一切摂動を調整できない。さらに、LSでは、差分ベクトルの方向に指向性があるが、GSでは探索点に関する指向性がなく、Lévy乱数による軸に沿った指向性しかない。よって、CSは、多様化・集中化を調整する操作が弱いと考えられる。そこで本節では、CSのパラメータと密接な関係にある「摂動・探索点分布の拡大・縮小」に限定して検討する。

α は摂動のスケールを調整し、MAにおける β は摂動の傾向を調整するパラメータであるため、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。しかし、 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。まず、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 $\hat{\mathbf{x}}$ を生成する。図 3.10 で示すように、近似 Lévy 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 \mathcal{L} 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 \mathcal{L} 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 \mathbf{x}^i に加えられる摂動が大きくなることで、近傍解 $\hat{\mathbf{x}}$ 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 $\hat{\mathbf{x}}$ 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 \mathbf{x}^i を中心とする摂動が拡大・縮小するため、近傍解 $\hat{\mathbf{x}}$ の生成範囲の広さも変化することが明らかになった。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）ことを明らかにし、② S_0 のサイズ $b = \gamma_{\max} - \gamma_{\min}$ によって、適切な β の範囲が異なることを推測した。

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 $|\mathcal{L}(\beta)|$ （摂動量）の標本の統計量を計算し、 β の特性を調べる。 β と摂動に関する定量的解析方法と結果を述べる。Lévy 乱数の絶対値 $|\mathcal{L}(\beta)|$ を摂動とみなし、ある確率密度関数に従う乱数 $|\mathcal{L}(\beta)|$ と、その乱数が従う確率密度関数のパラメータ β の定量的関係を明らかにする。この場合、発生させた乱数 $|\mathcal{L}(\beta)|$ をデータ（標本）として扱い、推定によって、この乱数データの統計的性質（母集団の性質）を調べることで、上記の定量的関係を明らかにすることが可能である。

代表的な推定方法は、パラメトリック推定とノンパラメトリック推定に大きく分類され

る。パラメトリック推定は、確率密度関数が明らかである必要があるが、簡潔な方法で、少ないデータで十分な精度の推定が可能であることに対して、ノンパラメトリック推定は、確率密度関数を使用しない分、複雑な方法で、十分な精度で推定を行うには多くのデータを必要とする。加えて、最も代表的なパラメトリック推定である最尤推定法では、確率密度関数が正規分布関数の場合、最尤推定量として、データ $\mathbf{x} \in \mathbb{R}^M$ の標本平均 μ と標本標準偏差 σ (統計量) を解析的に求めることができ、それらは式(5.58), 式(5.59)で表される。

$$\mu = \frac{1}{M} \sum_{j=1}^M x_j \quad (5.58)$$

$$\sigma = \sqrt{\frac{1}{M} \sum_{j=1}^M (x_j - \mu)^2} \quad (5.59)$$

一方、本論文では、Lévy 乱数 $\mathcal{L}(\beta)$ を Mantegna's Algorithm (付録 B) に従って生成するため、乱数 $|\mathcal{L}(\beta)|$ が従う確率密度関数は明らかではない (観測現象の母集団が明確ではない)。また、本論文のパラメータ β に対する解析は基礎的検討であるため、十分に近似精度の高い関係が得られなくとも、簡潔な方法で大まかな定量的関係を明らかにできれば良い。以上から、ここでは、乱数 $|\mathcal{L}(\beta)|$ が正規分布関数に従うと仮定し、パラメトリック推定である最尤推定法の結果を利用して、統計量を計算した後、その統計量が代入された正規分布関数を乱数 $|\mathcal{L}(\beta)|$ とパラメータ β の定量的関係として得る。

Algorithm 5.3.2 に乱数の統計量を計算するアルゴリズムを示す。上記のアルゴリズムで得た乱数の統計量を正規分布関数 $g(x; \beta)$ に代入することで、 β と $|\mathcal{L}(\beta)|$ の定量的関係を明らかにする。本論文では、 $\beta_{\min} = 0.3$, $\beta_{\max} = 1.99$, $\Delta\beta = 0.05$, $b = 10$, $M = 10^6$ とした。乱数の統計量を図 5.2 に、乱数データの正規分布関数 $g(x; \beta)$ を図 5.3 に示す。図 5.2 と図 5.3 から、定性的解析で明らかにした傾向に加えて、 β と乱数 $|\mathcal{L}(\beta)|$ 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。以上の解析から、 β と摂動量の定量的関係を明らかにした。さらに、この解析結果から、初期配置領域 S_0 のサイズ b によって、適切な摂動量が異なることに伴い、適切な β の範囲も異なることが推測できる。

以上の定量的解析から、 β の値と乱数 $|\mathcal{L}(\beta)|$ の間に関係が確認できる。さらに、初期配置領域 S_0 のサイズ $b = \gamma_{\max} - \gamma_{\min}$ によって、適切な摂動量が異なることと、この解析結果から、 b によって適切な β の範囲が異なることが推測できる。

Algorithm 14 亂数 $|\mathcal{L}(\beta)|$ の統計量の計算アルゴリズム

```

1: procedure COMPUTE_STATICS_OF Step 2: 亂数の統計量の計算
    RANDOM NUMBERS
    Step 0: 準備
    2:    $\beta$  の下限値  $\beta_{\min}$ , 上限値  $\beta_{\max}$ , 調整幅
         $\Delta\beta$ , 亂数の上限  $b$ , 発生させる乱数の個
        数  $M$  を設定する。
    3:    $\beta = \beta_{\min}$  とする。
    Step 1: 亂数の発生
    4:   亂数  $\mathcal{L}_j(\beta)$  ( $j = 1, \dots, M$ ) を発生させ
    5:    $G(\beta) = \{G_j(\beta)\}, j = 1, \dots, M\}$  と
    6:   式(5.58), 式(5.59)を用いて、乱数データ  $G(\beta)$  の平均  $\mu(\beta)$  と標準偏差  $\sigma(\beta)$  を
        求める。
    Step 3: 終了判定
    7:   if  $\beta = \beta_{\max}$  then
    8:     計算を終了する。
    9:   else
    10:     $\beta := \min\{\beta + \Delta\beta, \beta_{\max}\}$  とし,
        Step 1へ戻る。
    11:  end if
    12: end procedure

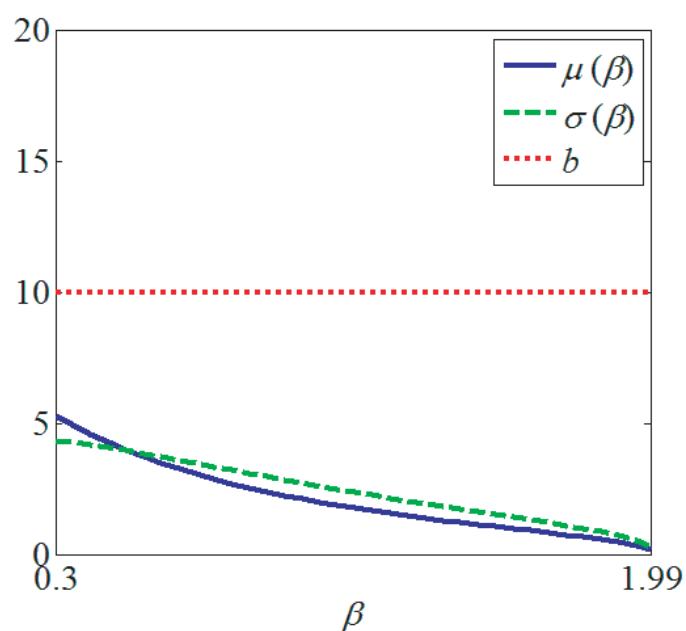
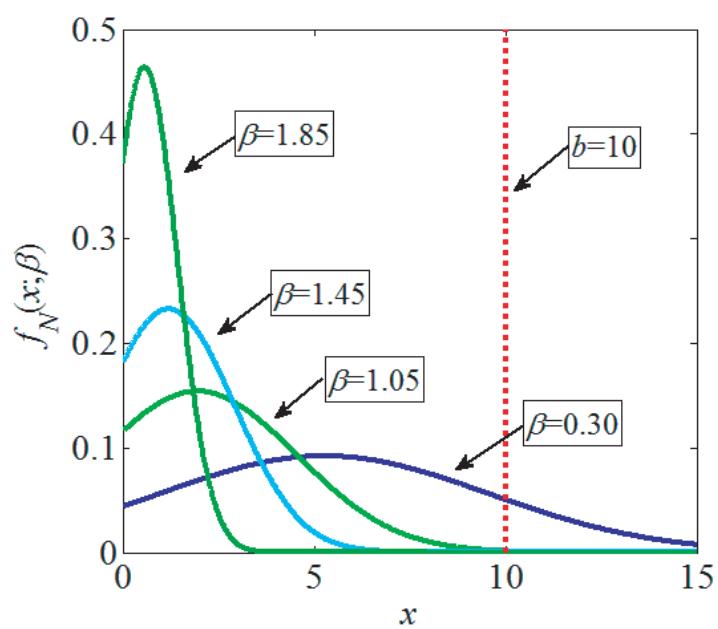
```

以上の解析から、 β を動的に調整することで、摂動の大きさを変化させることが可能である。しかしながら、 β と摂動の大きさの関係は指数的に減少するため、使用しづらい。よって、本節では、 β の値は平均的に優れた性能である $\beta = 1.9$ としておき、 α によって摂動の大きさを調整することを考える。

(b) Cuckoo Search のパラメータ解析

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、(c) 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。ここでは、数値実験を通じて、評価指標を用いて、CS の多様化・集中化の実現状態を定量的に評価することで、CS に対して、 α の変化を活用した多様化・集中化の調整能力を付加することで、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ことを示す。

多様化・集中化の評価指標として、5.2 節で述べた AFPSO の活性度 P を用いる。活性

図 5.2 : $|\mathcal{L}(\beta)|$ の統計量図 5.3 : $|\mathcal{L}(\beta)|$ の正規分布

度 P は移動ベクトル \mathbf{v}^i の L^2 ノルムの平均だが, CS における移動ベクトルは, 近傍生成則における $\mathbf{v}^i = \hat{\mathbf{x}}^i - \mathbf{x}^i$ とし, 式 (5.60) で表される。

$$P = \frac{1}{\sqrt{N}m} \sum_{i=1}^m \|\hat{\mathbf{x}}^i - \mathbf{x}^i\|_2 \quad (5.60)$$

さらに, GS では, $\hat{\mathbf{x}}^i = \mathbf{x}^i + \alpha \mathcal{L}$ であるため, 活性度 P は式 (5.61) で表される。

$$P = \frac{1}{\sqrt{N}m} \sum_{i=1}^m \|\alpha \mathcal{L}\|_2 \quad (5.61)$$

また, LS では, $\hat{\mathbf{x}}^i = \mathbf{x}^i + r(\mathbf{x}^{r_1} - \mathbf{x}^{r_2})$ であるため, 活性度 P は式 (5.62) で表される。

$$P = \frac{1}{\sqrt{N}m} \sum_{i=1}^m \|r(\mathbf{x}^{r_1} - \mathbf{x}^{r_2})\|_2 \quad (5.62)$$

P は, 摂動の大きさ, 近傍解 $\hat{\mathbf{x}}$ の生成範囲の広さを評価していることがわかる。LS では探索過程で探索点間の距離が小さくなるため, これに伴い, P も小さくなっていくが, GS では探索過程で乱数ベクトルは一定なので, P も一定の大きさとなり, α がそのスケールを担うパラメータとなる。よって, α が大きければ, P は大きく (生成範囲が広く), α が小さければ, P は小さい (生成範囲が狭い) と対応づけることで, P 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 P 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。

まず, α によって, P の値や推移が異なることを検証する。Rastrigin 関数と Ackley 関数を対象とし, α を変化させて実験を行う。ベンチマーク関数は付録 C を参照されたい。共通の実験条件として, 探索点数 $m = 20$, 次元数 $N = 50$, $\beta = 1.9$, 排斥確率 $P_a = 0.25$ とし, 関数毎に共通の初期配置領域 S_0 を用い, 終了条件を評価回数 $T_{\max} = 2000$ とする。 S_0 のサイズ b は $b = 10$ と固定する。

図 5.4, 図 5.5 に, $\alpha = 0.1, 0.5, 1.0$ に設定した場合の評価指標 P と最良解の評価値 $f(\mathbf{x}^g)$ の推移を示す。 α が小さい場合, 多様化・集中化の評価指標 P は小さな値で一定となることに対して, α が大きい場合, P は大きな値で一定となることが確認できる。これは, α が小さい場合, 摂動が狭い状態が続き, α が大きい場合, 摂動が広い状態が続くことを示し

ている。先ほど指摘したように、 P 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。

次に、 S_0 のサイズ b によって、 α が P へ与える影響が異なることを検証する。Ackley 関数を対象とし、 b を変化させて実験を行う。共通の実験条件として、 $m = 20$ 、 $N = 50$ 、 $\alpha = 0.1$ 、 $\beta = 1.9$ 、 $P_a = 0$ を用い、終了条件を $T_{\max} = 2000$ とする。

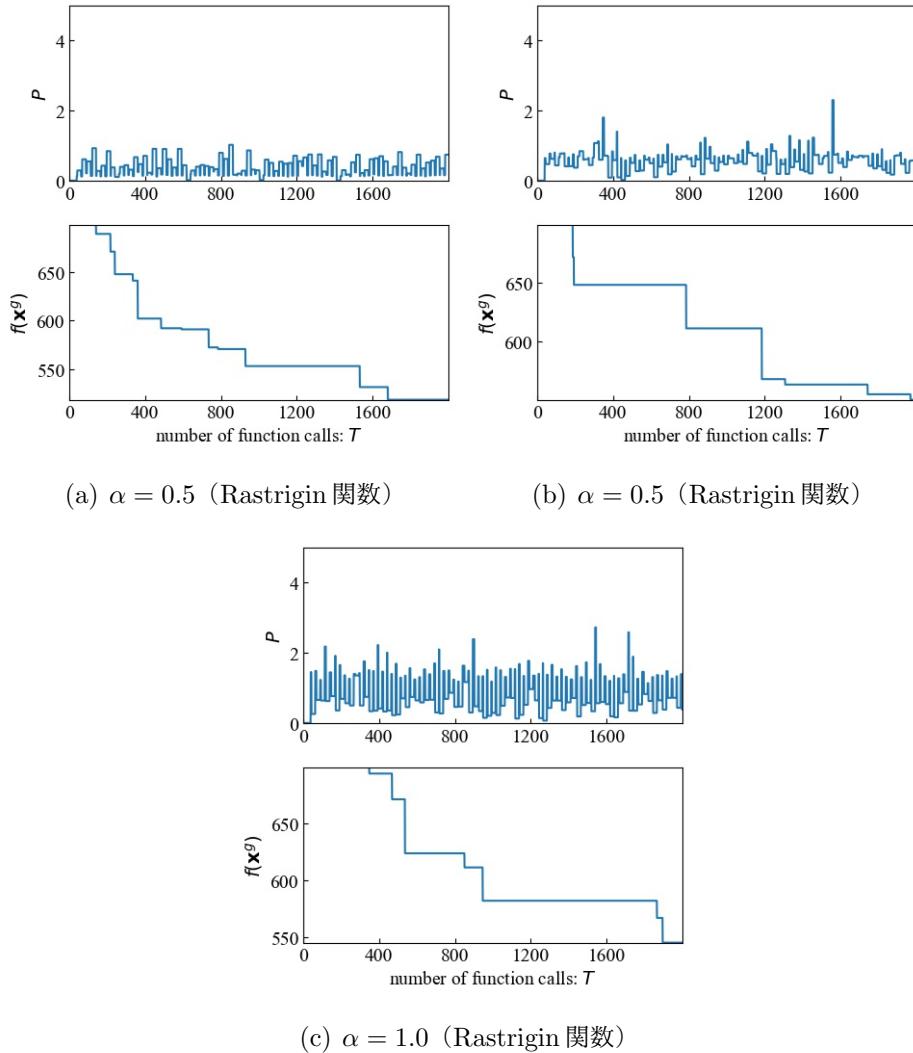
図 5.6 に、 $b = 20, 200, 2000$ に設定した場合の評価指標 P と最良解の評価値 $f(\mathbf{x}^g)$ の推移を示す。 b が大きい場合、 P は探索過程で漸減することに対して、 b が小さい場合、 P は探索過程で一定となることが確認できる。この結果は、先ほど推測したように、 S_0 のサイズ b によって、 α が P へ与える影響は異なることを示している。これは、 b によって、適切な摂動の範囲が異なることに応じて、 α の適切な値も変化するためだと考えられる。

したがって、数値実験を通じて、 α 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 α 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。

5.3.3 相似変換不变性を補完する適応型 Cuckoo Search

CS は相似変換不变性や回転不变性など、多くの不变性を欠如している。また、従来の CS のパラメータ調整方法では、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、フィードフォワード型の調整則であるため、多様化・集中化が適切に実現されているとはいえない。一方、Yasuda らは、PSO、DE に対して、探索状態（多様化・集中化）を定量的に評価しながら、それをフィードバックし、パラメータを調整する、適応的調整型のパラメータ調整則を付加することで、適応型メタヒューリスティクスを開発し、多様化・集中化を適切に実現することで、適応能力・探索性能の向上が可能であることを示してきた[116, ?]。したがって、このアプローチと同様に、CS の不变性を補完すると同時に、適切な多様化・集中化が実現可能なパラメータ調整方法を開発することで、CS の適応能力・探索性能の向上が期待できる。

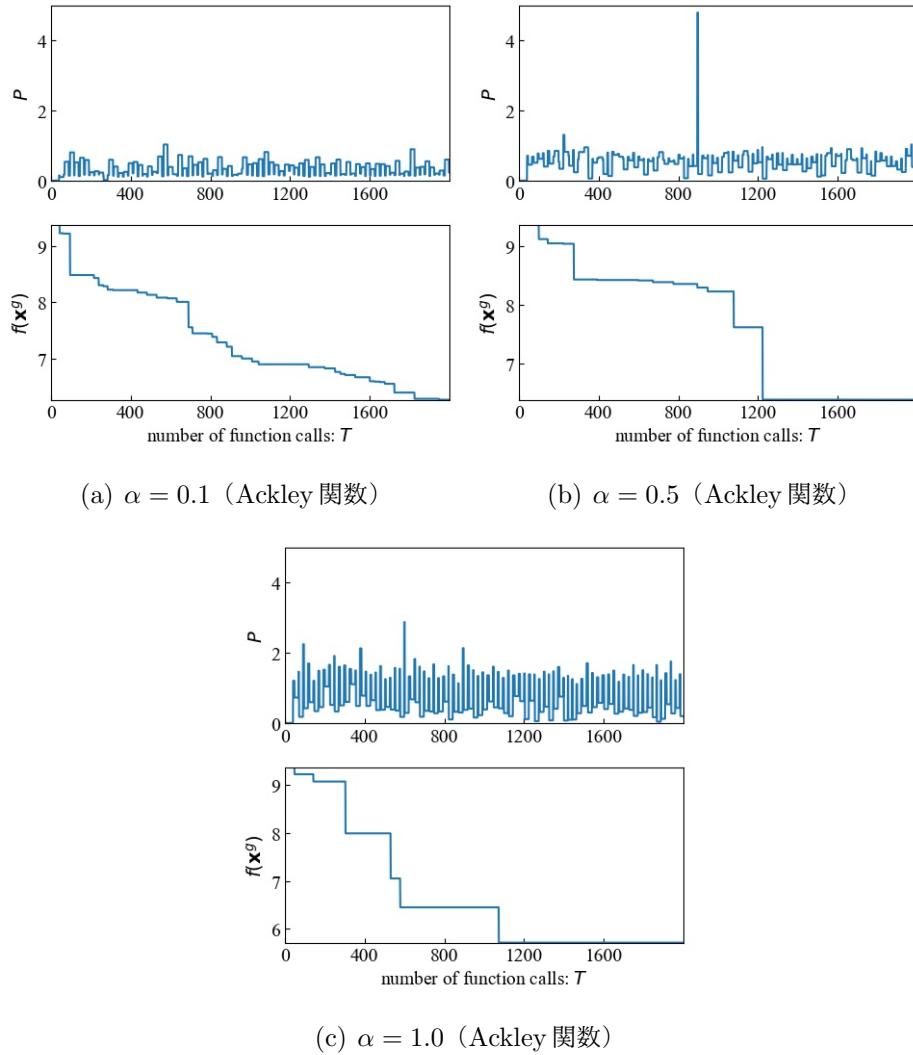
以上を踏まえ、本節では、相似変換不变性の補完と多様化・集中化の定量的評価と適応的制御に基づく適応型 CS を提案する。適応型 CS は CS の更新則と同様だが、GS における

図 5.4：CS の探索過程における P と $f(\mathbf{x}^g)$ の推移 ($N = 50$ 、 b 固定、Rastrigin 関数)

るパラメータ α を動的に調整する。 α の調整則は式 (5.63) で表される。

$$\alpha := \begin{cases} \min \{\alpha + \Delta\alpha, \alpha_{\max}\}, & (P \geq P_{\text{target}}^T) \\ \max \{\alpha - \Delta\alpha, \alpha_{\min}\}, & (P < P_{\text{target}}^T) \end{cases} \quad (5.63)$$

これは、事前に評価指標の目標値スケジュール P_{target}^T ($T = 1, 2, \dots, T_{\max}$) を与えておき、各評価回数 T における評価指標 P の値と比較し、 P が P_{target}^T に追従するように調整幅 $\Delta\alpha > 0$ で α を調整する。式 (5.63) の調整則は、AFPSO の式 (5.28) とロジックは同様である。なお、LS は CS と全く同様である。

図 5.5：CS の探索過程における P と $f(\mathbf{x}^g)$ の推移 ($N = 50$ 、 b 固定、Griewank 関数)

この調整則では、目標値スケジュール P_{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。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。このため、目標値 P_{target} として探索過程で一様に減少する形状を使用することで、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。本節では、目標値 P_{target} として、AFPSO と同様に、一様に減少する最も単純なスケジュールの一つである指数減少スケジュール（式(5.30)）を使用する⁹。本論文で使用する目標値 P_{target} の指数減少スケ

⁹ 本節では基礎的検討として、指数減少スケジュールを使用するが、これ以外のスケジュールとして、線形

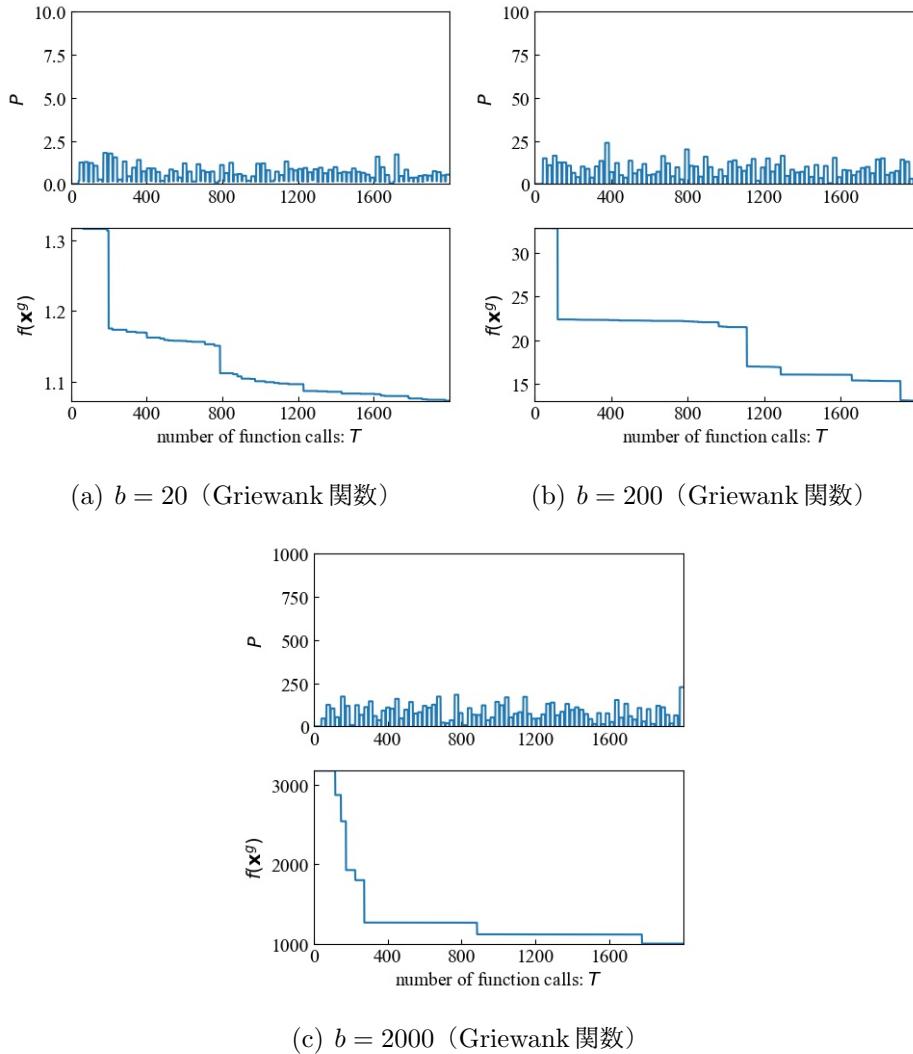


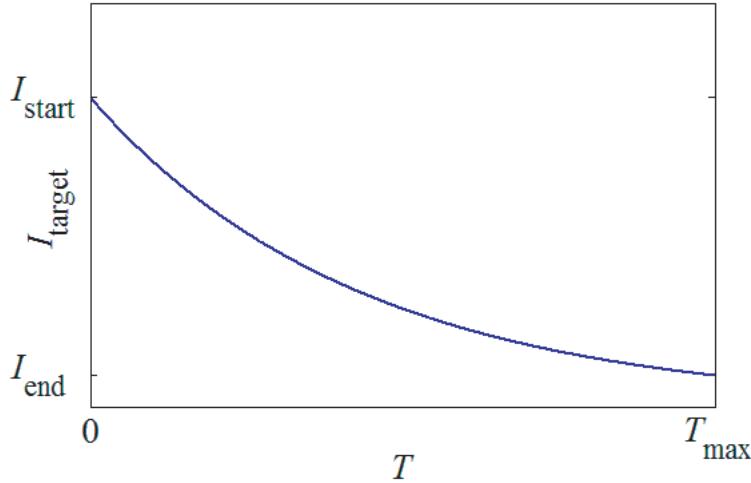
図 5.6：CS の探索過程における P と $f(\mathbf{x}^g)$ の推移 ($N = 50$ 、 $\alpha = 0.1$ 、Griewank 関数)

ジユールを図 5.7 に示し、式 (5.64) にスケジュールの式を示す。

$$P_{\text{target}}^T = P_{\text{start}} \left(\frac{P_{\text{end}}}{P_{\text{start}}} \right)^{\frac{T}{T_{\max}}} \quad (5.64)$$

この調整則は、許容される反復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、最大反復回数の制約の下で多様化・集中化を

減少スケジュールや、探索過程で値の減少を周期的に繰り返す（のこぎり波など）スケジュールも考えられる。

図 5.7：目標値スケジュール P_{target}

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。これまでに Yasuda らは、PSO と DE に対して、多様化・集中化の観点から、同様のパラメータ調整則を付加することで、適応性・探索性能の向上が可能であることを確認した[116, 125]。よって、本論文においても同様に、パラメータ調整則の付加によって、CRI-PSO の適応性・探索性能の向上が期待できる。

なお、目標値 P_{target} では、新たに P_{start} , P_{end} というパラメータが導入されている。探索戦略の観点から、 P_{start} を多少大きく、 P_{end} を十分小さく設定すればよい。本章では、著者の数値実験に基づき、提案手法のパラメータ推奨値は、 $P_{\text{start}} = \varepsilon_{\text{start}} x_{\text{width}}$, $P_{\text{end}} = \varepsilon_{\text{end}} x_{\text{width}}$, $\varepsilon_{\text{start}} = 0.25$, $\varepsilon_{\text{end}} = 0.001$ とする。 x_{width} は式(5.65)で表される。

$$x_{\text{width}} = \frac{1}{\sqrt{N}} \| \boldsymbol{\gamma}_{\max} - \boldsymbol{\gamma}_{\min} \|_2 \quad (5.65)$$

ただし、初期配置領域 $S_0 = [\gamma_{\min,n}, \gamma_{\max,n}]^N$ である。また、これらは、パラメータ調整則を有するアルゴリズムの変換不変性の有無に影響を与えるため、後で述べる条件式を満たすように設定すればよい。

Algorithm 5.3.3 に適応型 CS の疑似コードを示す。

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。

Algorithm 15 適応型 CS

```

1: procedure ADAPTIVE CS
   10:  $\theta \sim \mathcal{U}_{\mathbb{R}}(0, 1)$ 
      Step 0: パラメータ設定
      11: if  $P_a \geq \theta$  then
      12:    $\mathbf{x}^i$  を式 (3.34)、式 (3.35) を用
            $\Delta\alpha$ 、 $P_a$  を設定する。
           いて更新する。
      13:   end if
      14: end for
      15:  $\alpha$  を式 (5.63) を用いて更新する。
      Step 4:  $\alpha$  の更新
      16: if  $T \geq T_{\max}$  then
      17:   計算を終了する。
      18:   else
      19:     Step 2 へ戻る。
      20:   end if
      21: end procedure

```

5.3.4 変換不变性に基づく適応型 CS の解析

本節では、適応性の向上と相似変換不变性の補完のために、CS にパラメータ調整則を付加する。5.2 節で述べたように、変換不变性に基づくパラメータ調整則の指針は 2 つあるが、本節では、指針 (2) 「更新則が変換不变性を欠如するとき、その変換不变性を補完するように、その変換不变性に影響を与えるパラメータ調整則を付加する」に従って、パラメータ調整則を設計する。このため、前項のパラメータ調整則が、CS の解空間の相似変換不变性を補完するのかを簡潔に確認する。

定義 5.1 に従うと、式 (5.63) のパラメータ調整則 $F_{f,S}$ において、 P, P_{target} は探索履歴 χ_S に依存し、 $\alpha_{\min}, \alpha_{\max} \in \mathcal{C}_F$ となる。

目標値の式を変形すると,

$$\begin{aligned}
 P_{\text{target}}^T &= P_{\text{start}} \left(\frac{P_{\text{end}}}{P_{\text{start}}} \right)^{\frac{T}{T_{\max}}} \\
 &= \varepsilon_{\text{start}} x_{\text{width}} \left(\frac{\varepsilon_{\text{end}} x_{\text{width}}}{\varepsilon_{\text{start}} x_{\text{width}}} \right)^{\frac{T}{T_{\max}}} \\
 &= \frac{\varepsilon_{\text{start}}}{\sqrt{N}} \| \boldsymbol{\gamma}_{\max} - \boldsymbol{\gamma}_{\min} \|_2 \left(\frac{\varepsilon_{\text{end}}}{\varepsilon_{\text{start}}} \right)^{\frac{T}{T_{\max}}} \\
 &= \mathcal{E}_1 \| \boldsymbol{\gamma}_{\max} - \boldsymbol{\gamma}_{\min} \|_2
 \end{aligned}$$

である。ただし、定数部分を $\mathcal{E}_1 = \frac{\varepsilon_{\text{start}}}{\sqrt{N}} \left(\frac{\varepsilon_{\text{end}}}{\varepsilon_{\text{start}}} \right)^{\frac{T}{T_{\max}}}$ とおいた。また、式(5.61)の評価指標 P は、 $P = \frac{1}{\sqrt{N}m} \sum_{i=1}^m \| \alpha \mathcal{L} \|_2 = \mathcal{E}_2 \sum_{i=1}^m \| \alpha \mathcal{L} \|_2$ である。ただし、定数部分を $\mathcal{E}_2 = 1/(\sqrt{N}m)$ とおいた。ここで、式(5.63)のパラメータ調整則 F_f は P を P_{target} に追従するから、十分に追従したと仮定すると、 $P \sim P_{\text{target}}$ である。よって、

$$\begin{aligned}
 \mathcal{E}_1 \| \boldsymbol{\gamma}_{\max} - \boldsymbol{\gamma}_{\min} \|_2 &= \mathcal{E}_2 \sum_{i=1}^m \| \alpha \mathcal{L} \|_2 \\
 \Leftrightarrow \alpha &= \frac{\mathcal{E}_1}{\mathcal{E}_2 \sum_{i=1}^m \| \mathcal{L} \|_2} \| \boldsymbol{\gamma}_{\max} - \boldsymbol{\gamma}_{\min} \|_2 \\
 &= \mathcal{E} \| \boldsymbol{\gamma}_{\max} - \boldsymbol{\gamma}_{\min} \|_2
 \end{aligned}$$

となる。ただし、定数部分を $\mathcal{E} = \mathcal{E}_1 / (\mathcal{E}_2 \sum_{i=1}^m \| \mathcal{L} \|_2)$ とおいた。つまり、このとき、 α は初期配置領域のスケールに合わせることを表している。ここで、相似変換を与えると、 $\alpha' = \delta \alpha$ となり、パラメータの不变性条件を満たさないが、近傍生成則では、

$$\begin{aligned}
 \hat{\mathbf{x}}^{i\prime} &= \mathbf{x}^{i\prime} + \alpha' \mathbf{s}^{i\prime} \\
 &= \delta \mathbf{x}^i + \delta \alpha \mathbf{s}^i \\
 &= \delta (\mathbf{x}^i + \alpha \mathbf{s}^i) \\
 &= \delta \hat{\mathbf{x}}^i
 \end{aligned}$$

となることから、更新則全体の不变性を有することがわかる。したがって、このパラメータ調整則はCSの相似変換不变性を補完していることを示した。

5.3.5 数値実験による検証

(a) 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、適応型CS（提案手法）の探索性能を検証する。適応型CSと比較する手法は、オリジナルCSとする。数値実験に用いるベンチマーク関数は付録Cを参照されたい。

共通条件として、探索点数 $m = 20$, $\beta = 1.9$, 排斥確率 $P_a = 0.25$, 関数毎に共通の初期配置領域 S_0 を用い、終了条件を評価回数 $T_{\max} = 2000, 20000$ とする。適応型CSでは下限値 $\alpha_{\min} = 0.01$, 上限値 $\alpha_{\max} = 1.0$, 調整幅 $\Delta\alpha = 0.1$ を用いる。このため、 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。初期配置領域 S_0 内に初期探索点をランダムに配置した後、終了条件を満たすまで探索を進める試行を 50 回行い、50 試行の g-best の目的関数値 $f(\mathbf{x}^g)$ の平均値を実験結果とする。

(b) アフィン変換に対するロバスト性に関する検証

数値実験を通じて、適応型CSのアフィン変換（スケール変換を含む）に対する探索性能のロバスト性を検証する。本実験では、解空間にアフィン変換の一種である平行移動変換、スケール変換、回転変換を与えていき、その探索性能の変動を確認する。ベンチマーク関数として、変数分離可能・良スケール性・原点に最適解が存在・多峰性の4種類の目的関数を使用する。実験方法は、4.4.4項と同様である。共通条件として、次元数 $N = 50$, 終了条件を評価回数 $T_{\max} = 20000$, CSのパラメータは $\alpha = 0.1$ とする。

図5.8, 図5.9, 図5.10に数値実験結果を示す。各図の横軸は、平行移動パラメータ δ , スケール変換パラメータ ε , 回転角 θ であり、全ての図の縦軸が探索性能 ($f(\mathbf{x}^g)$ の平均値) である。図5.8, 図5.9, 図5.10より、下記の結果が確認できる。

- 解空間の平行移動変換に対して、オリジナルCSと適応型CSは探索性能の変動が小さい。
- 解空間のスケール変換に対して、オリジナルCSと適応型CSは探索性能の変動が大きいが、適応型CSのほうが変動が多少抑えられている。

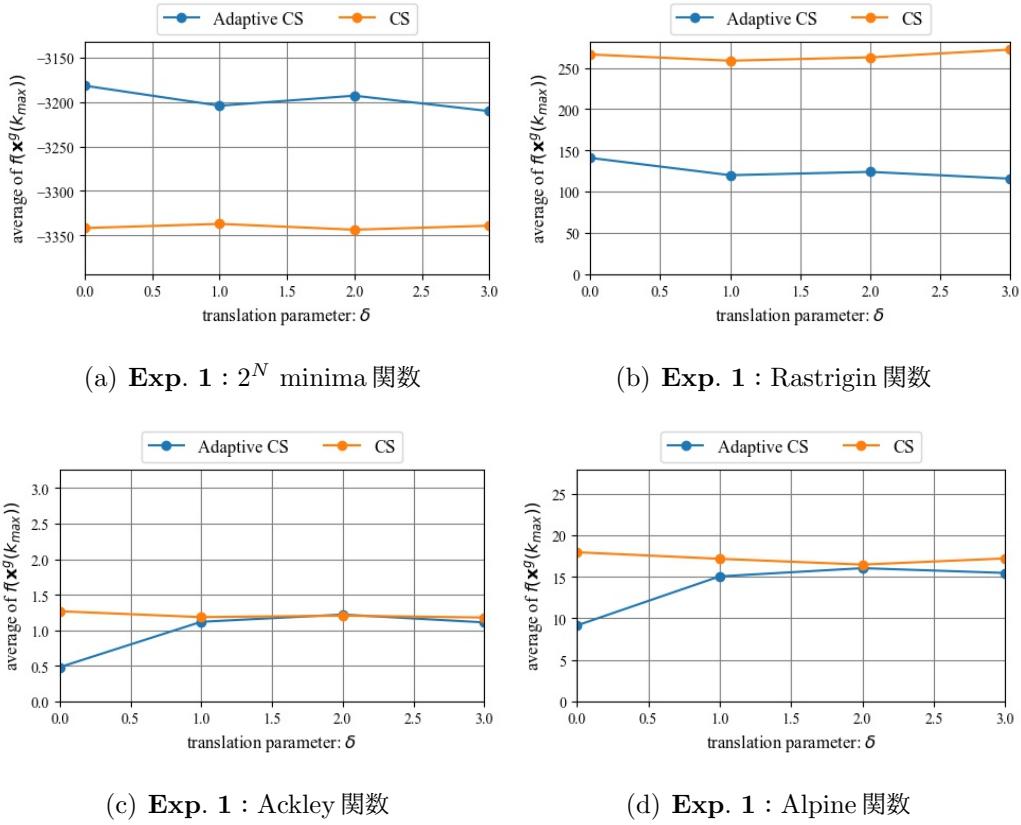


図 5.8 : 数値実験結果 (Exp. 1) : CS と適応型 CS の探索性能のロバスト性 ($N = 50$, $T_{\max} = 20000$)

- 解空間の回転変換に対して、オリジナル CS と適応型 CS は探索性能の変動が大きい。
- 適応型 CRI-PSO は、多くの条件で、探索性能がオリジナル CS よりも優れている。

なお、 $\theta = 90 \text{ deg}$ のときは、ベンチマーク関数の対称性により、 $\theta = 0 \text{ deg}$ と同じ条件である。

図 5.9 の結果は、相似変換不变性の有無が解空間のスケール変換に対する探索性能のロバスト性に影響を与えることを示している。よって、適応型 CS は、パラメータ調整則の付加によって、相似変換不变性・スケール不变性を補完しているため、解空間のスケール変換に対する探索性能のロバスト性も多少向上していることが示された。しかしながら、その改善は大きなものではなく、あくまで「補完」であることがわかる。また、図 5.10 の結果は、回転不变性の有無が解空間の回転変換に対する探索性能のロバスト性に影響を与えることを示している。このため、オリジナル CS も適応型 CS も、回転変換に対してはロ

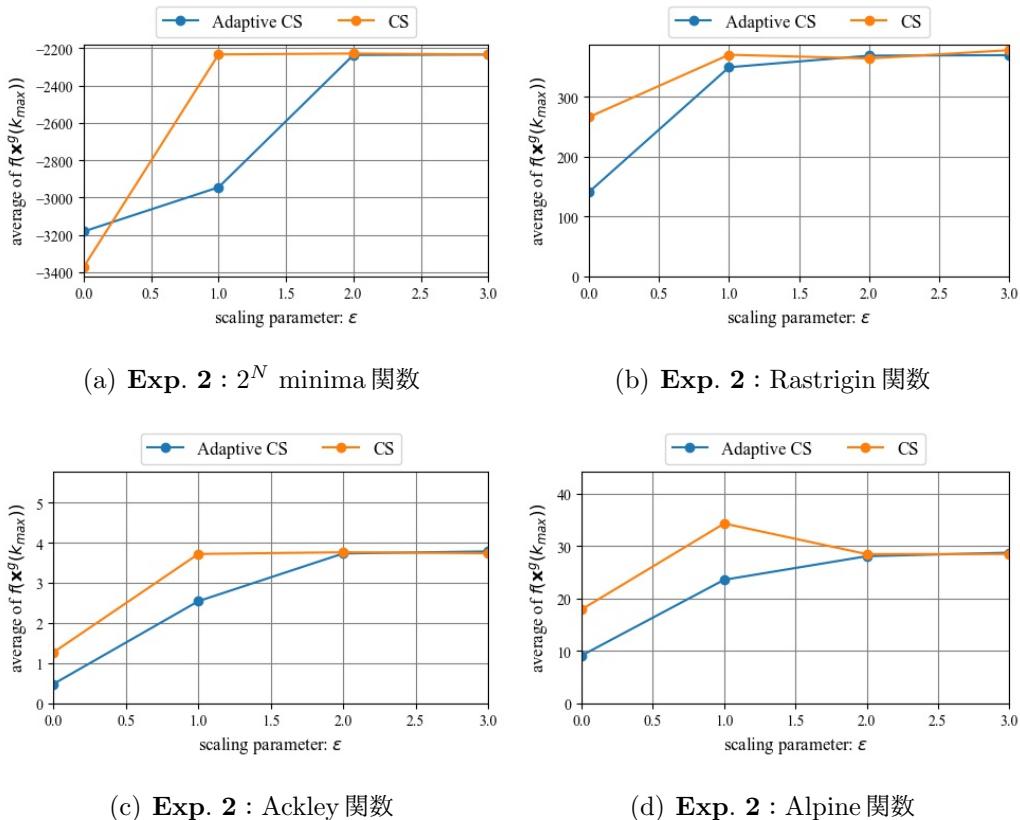


図 5.9： 数値実験結果 (Exp. 2)：CS と適応型 CS の探索性能のロバスト性 ($N = 50$, $T_{\max} = 20000$)

バスト性を失ったままであることも示している。しかしながら、適応型 CS の探索性能は、多くの条件で CS よりも優れている。

(c) 全般的な探索性能に関する検証

本実験では、数値実験を通じて、適応型 CS の全般的な探索性能を検証する。数値実験には、9種類のベンチマーク関数を用いる。共通条件として、次元数 $N = 10, 50, 100, 300$ 、終了条件を評価回数 $T_{\max} = 2000, 20000$ とする。オリジナル CS では、 $\alpha = 0.01, 0.1, 0.2, \dots, 1.0$ の11通りの値を用いる。

本実験では、適応型 CS の探索性能を、オリジナル CS の11通りの α の値における結果

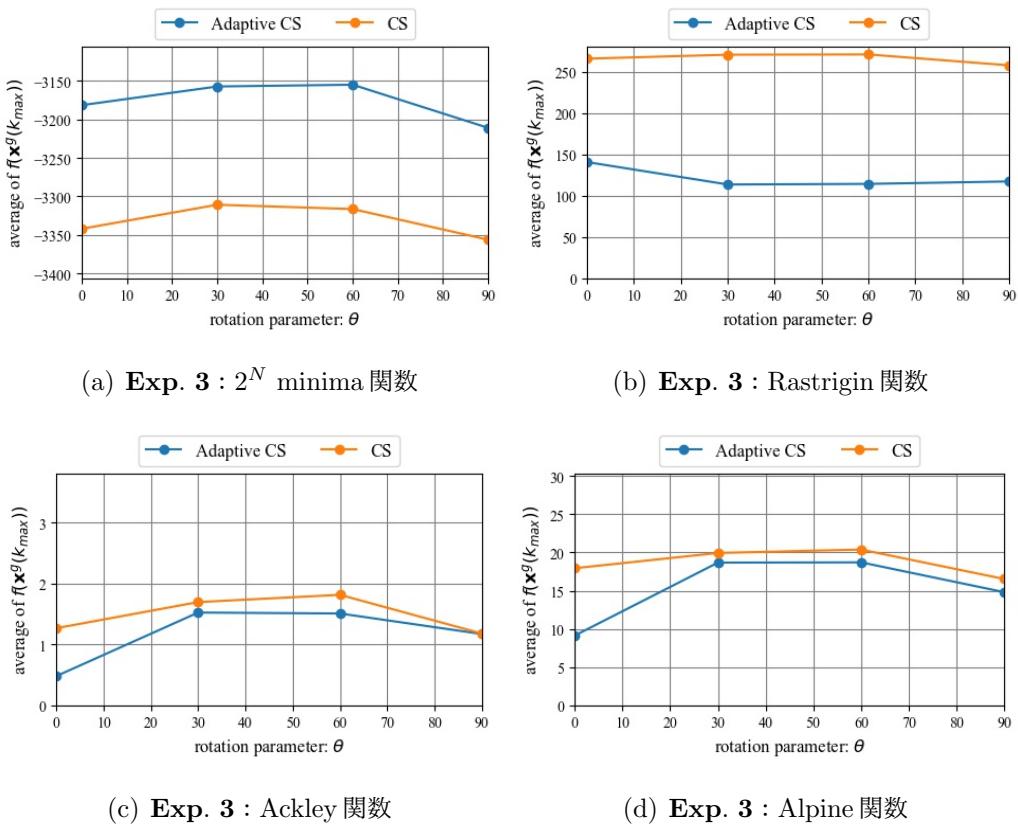


図 5.10： 数値実験結果 (Exp. 3) : CS と適応型 CS の探索性能のロバスト性 ($N = 50$, $T_{\max} = 20000$)

と比較することで評価する。オリジナル CS では、11通りの結果の中で、最も優れている結果（最小値）を1位、3番目に優れている結果（第一四分位数）を3位、5番目に優れている結果（中央値）を5位と表現する。アルゴリズムや問題に対する知識や経験を有していないユーザは、 α をランダムに決定することが考えられる。そのため、中央値はこのようなユーザ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 α の値も示す。適応型 CS では、 P_{target} を図 5.7 指数減少スケジュールに設定した条件で得た結果を「指数」と表現する。また、オリジナル CS の 11通りと適応型 CS を合わせた 12通りの結果における、適応型 CS の順位を「順位」と表現する。「順位」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。

表 5.2 に $T_{\max} = 2000$ の数値実験結果を、表 5.3 に $T_{\max} = 20000$ の数値実験結果を示す。

これらの表において、適応型CSの順位が、5位と3位の間の場合■を、3位と1位の間の場合◆を、1位の場合●を付す。よって、何らかのマークが付されているなら、中央値よりも優れているといえる。

表5.2、表5.3より、適応型CSは多くの条件において、中央値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 $T_{\max} = 20000$ ）において、3位や1位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??にRosenbrock関数、Rastrigin関数、Griewank関数（ $N = 50$ ）を対象とした場合の α 、多様化・集中化の評価指標 P 、最良解の評価値 $f(x^g)$ の推移を示す。図??より、適応型CSでは、探索過程で P が目標値 P_{target} に追従することで徐々に漸減すると同時に、 $f(x^g)$ が改善する様子を確認できる。5.3.2項で述べたように、関数毎に設定される初期配置領域 S_0 のサイズによって適切な α の値は異なるため、オリジナルCSでは探索条件に応じて α を設定する必要がある。Griewank関数ではサイズが $b = 100$ 、Alpine関数ではサイズが $b = 20$ 、他の関数ではサイズが $b = 10$ だが、Griewank関数以外では、CSの優れているパラメータは $\alpha = 0.1$ など、小さな値が多いことに対して、Griewank関数では、 $\alpha = 0.5$ など、比較的大きな値が多い。これは、パラメータ解析の通り、探索過程で探索状態が大きく変化しないため、サイズに適応することができず、探索戦略が適切に実現されないことを示している。一方で、適応型CSでは、本実験の結果から、 P を目標値に追従させることで、探索序盤で初期配置領域 S_0 のサイズに応じた値へ α 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といえる。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の相似変換不变性を補完すると同時に、多様化・集中化の調整能力を向上させた結果、CSのロバスト性・適応性を向上させることができた。

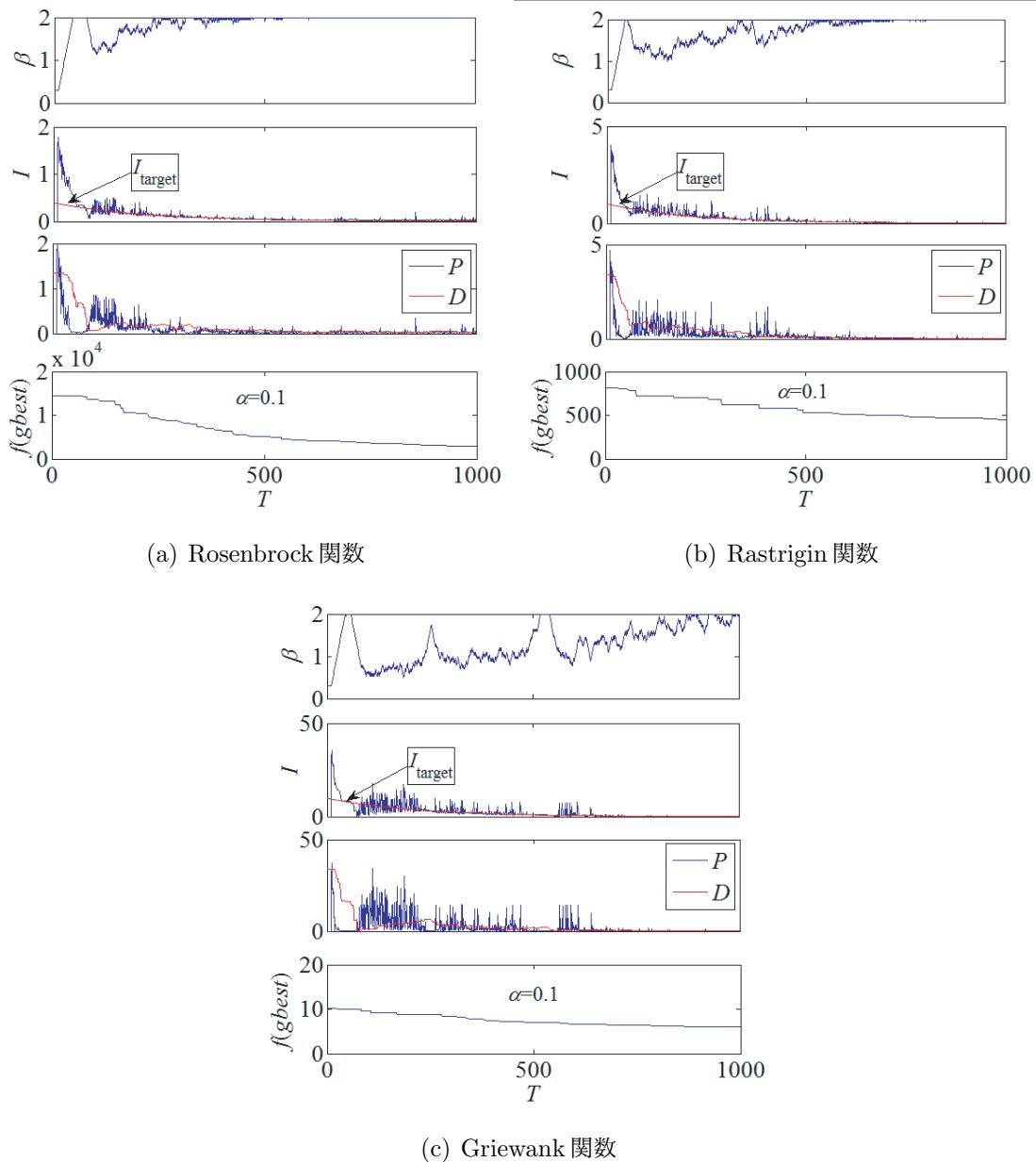
図 5.11：適応型 CS の探索過程における I と $f(\mathbf{x}^g)$ の推移 ($N = 50$)

表5.2：数値実験結果：適応型CSの全般的性能 ($T_{\max} = 2000$)

関数	N	CS (11種類の α)						適応型CS	
		1位	α	3位	α	5位	α	指數	順位
1. Parabola	10	0.57	0.3	0.64	0.2	1.08	0.7	2.63	10
	50	51.9	0.2	59.2	0.1	69.8	0.7	126	11
	100	153	0.1	165	0.3	181	0.01	131	• 1
	300	549	0.1	602	1.0	621	0.8	416	• 1
2. Rosenbrock	10	21.3	0.1	29.3	0.3	55.4	0.6	43.2	■ 5
	50	1204	0.1	1907	0.3	2415	0.6	1638	◆ 3
	100	4359	0.1	4888	0.01	5388	1.0	3596	• 1
	300	17798	0.1	18170	0.8	18623	1.0	24957	11
3. 2^N minima	10	-692	0.3	-686	0.2	-670	0.6	-643	10
	50	-2448	0.2	-2357	0.1	-2134	0.6	-2208	6
	100	-4126	0.2	-3992	0.3	-3630	0.01	-3337	11
	300	-10587	0.1	-10092	0.01	-9938	0.8	-9412	11
4. Rastrigin	10	42.5	0.1	56.0	0.3	60.9	0.5	49.2	◆ 3
	50	444	0.1	505	0.01	528	0.5	446	◆ 2
	100	1020	0.1	1079	0.01	1147	0.4	1004	• 1
	300	3380	0.1	3505	0.2	3596	0.9	3224	• 1
5. Schwefel	10	2.02	0.3	2.32	0.5	2.90	0.7	5.51	11
	50	143	0.2	156	0.3	172	0.5	102	• 1
	100	571	0.2	629	0.1	778	0.6	710	6
	300	5059	0.1	5293	0.3	7517	0.6	3564	• 1
6. Levy	10	7.74	0.8	9.12	0.9	9.87	0.4	9.22	■ 4
	50	35.6	0.1	41.2	0.3	46.5	0.6	30.4	• 1
	100	44.2	0.1	50.0	0.01	55.3	0.7	49.5	◆ 3
	300	52.7	0.01	57.8	0.2	59.5	0.9	75.8	12
7. Ackley	10	2.84	0.5	2.96	0.3	3.05	0.8	3.16	7
	50	5.73	0.2	5.85	0.4	6.04	0.6	6.06	7
	100	6.23	0.1	6.34	0.2	6.50	0.01	6.68	12
	300	6.52	1.0	6.53	0.1	6.64	0.5	7.44	12
8. Griewank	10	0.79	0.6	0.81	0.7	0.83	0.8	0.76	• 1
	50	2.47	1.0	2.70	0.8	2.79	0.7	2.56	◆ 2
	100	4.73	0.9	4.91	1.0	5.05	0.7	4.58	• 1
	300	14.4	0.8	14.8	0.4	15.2	0.5	16.0	11
9. Alpine	10	3.13	0.1	3.67	0.3	4.95	0.6	2.15	• 1
	50	57.3	0.1	58.9	0.3	64.9	0.6	74.0	12
	100	128	0.1	137	0.4	148	0.6	136	◆ 3
	300	436	0.3	450	0.2	469	0.6	475	8

表 5.3： 数値実験結果：適応型 CS の全般的性能 ($T_{\max} = 20000$)

関数	N	CS (11種類の α)						適応型 CS	
		1位	α	3位	α	5位	α	指數	順位
1. Parabola	10	0.00	0.01	0.00	0.2	0.00	0.40	0.00	9
	50	0.18	0.1	0.62	0.2	3.09	0.5	0.10 ●	1
	100	2.09	0.1	8.51	0.0	25.1	0.5	3.84 ◆	2
	300	50.2	0.1	100	0.2	143	0.4	60.3 ◆	2
2. Rosenbrock	10	3.50	0.01	5.73	0.2	7.26	0.5	2.78 ●	1
	50	49.5	0.01	145	0.2	280	0.5	50.5 ◆	2
	100	135	0.01	563	0.2	841	0.6	143 ◆	2
	300	931	0.01	3062	0.9	3067	1.0	1010 ◆	2
3. 2^N minima	10	-755	0.4	-752	0.3	-744	0.6	-783 ●	1
	50	-3371	0.1	-3113	0.3	-2847	0.5	-3182 ◆	3
	100	-6158	0.1	-5291	0.3	-4772	0.5	-5745 ◆	3
	300	-14714	0.1	-13163	0.01	-11190	0.5	-12891 ■	4
4. Rastrigin	10	17.0	0.01	22.6	0.2	30.0	0.5	12.0 ●	1
	50	101	0.01	339	0.2	397	0.5	141 ◆	2
	100	287	0.01	818	0.2	886	0.9	350 ◆	2
	300	1593	0.01	2776	1.0	2855	0.2	1960 ◆	2
5. Schwefel	10	0.00	0.01	0.00	0.2	0.00	0.5	0.00 ■	4
	50	6.15	0.1	9.47	0.2	15.7	0.5	9.12 ◆	3
	100	35.0	0.01	48.4	0.2	78.6	0.5	49.1 ■	4
	300	339	0.01	479	0.2	836	0.5	612 ■	5
6. Levy	10	0.05	0.4	0.06	0.8	0.07	0.5	0.00 ●	1
	50	6.12	0.01	8.54	0.8	8.76	0.6	5.71 ●	1
	100	7.74	0.01	14.7	0.3	15.2	0.5	8.38 ◆	2
	300	12.9	0.01	22.8	0.2	23.6	0.8	15.5 ◆	2
7. Ackley	10	0.03	0.2	0.04	0.4	0.05	0.6	0.01 ●	1
	50	1.27	0.1	1.99	0.3	2.52	0.5	0.48 ●	1
	100	2.29	0.1	2.91	0.01	3.63	0.5	2.77 ◆	3
	300	3.56	0.1	3.96	0.2	4.24	0.7	3.62 ◆	3
8. Griewank	10	0.06	0.1	0.09	0.3	0.11	0.5	0.04 ●	1
	50	0.27	0.3	0.30	0.2	0.35	0.7	0.15 ●	1
	100	0.83	0.4	0.84	0.5	0.97	0.8	0.71 ●	1
	300	1.77	0.4	1.82	0.3	2.04	0.2	2.12	8
9. Alpine	10	0.30	0.01	0.92	0.2	1.64	0.5	0.06 ●	1
	50	16.3	0.01	23.9	0.2	32.3	0.5	9.13 ●	1
	100	45.0	0.1	60.2	0.2	76.0	0.5	57.3 ◆	3
	300	185	0.01	227	0.2	252	0.9	222 ■	4

5.4 回転不变性を有する適応型 Particle Swarm Optimization

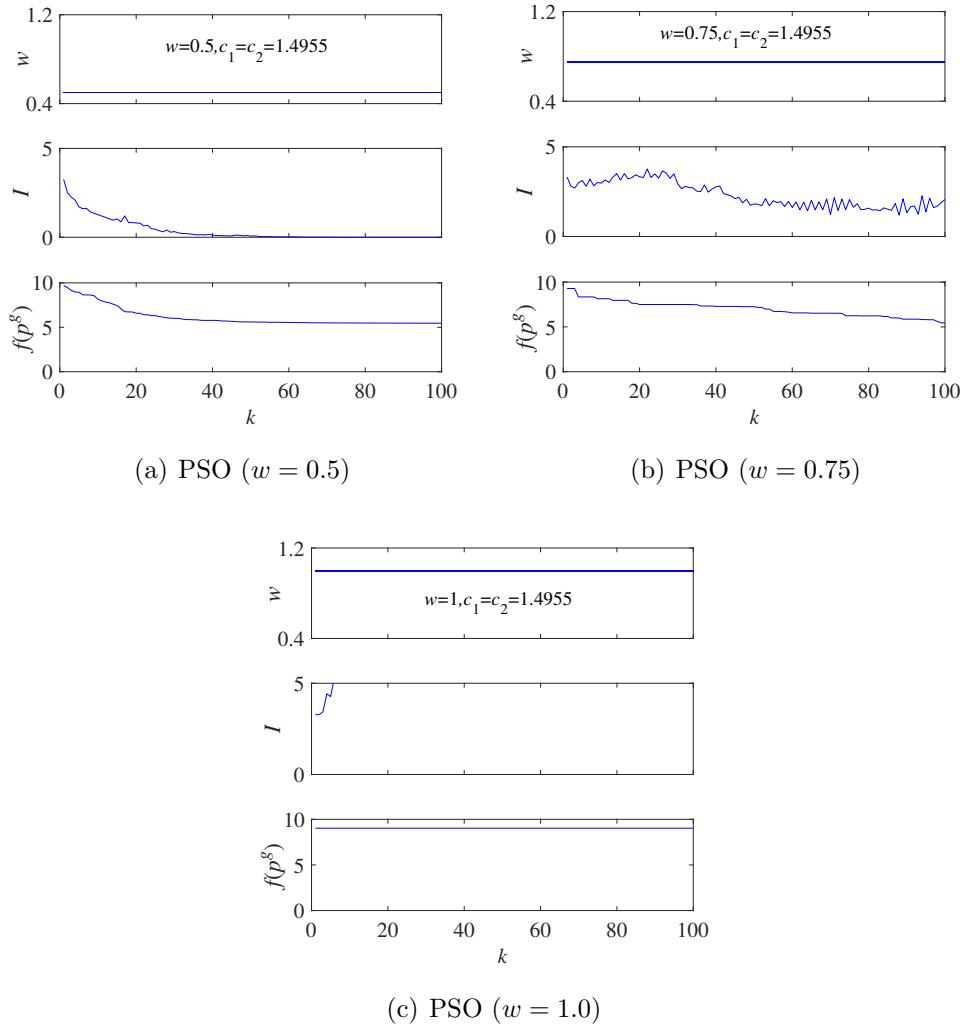
5.4.1 背景

5.2 節で述べたように、これまでに PSO のパラメータ調整則は多く検討されてきた。一方、PSO は回転不变性を欠如しているため、4.4 節では回転不变性を有する PSO として新たに CRI-PSO を提案すると同時に、回転変換に対するロバスト性が向上することと、全般的な性能も平均的に PSO よりも優れていることを確認したが、その性能がパラメータに依存するため、適応性に課題があった。

そこで、本節では、BBO での適用を想定し、高いロバスト性・適応性を有する PSO として、回転不变性を有する適応型 PSO を提案する。具体的には、5.2 節の設計指針(1)に従い、4.4 節で提案した CRI-PSO に対して、変換不变性を維持するように適応的調整型のパラメータ調整則を付加する。先行研究では、①4.2 節で概観したように、回転不变性を有する PSO を提案しているが、パラメータ調整則を具備していない、②5.2 節で概観したように、様々なパラメータ調整則を有する PSO を提案しているが、回転不变性を具備していない、ということに対して、本節の提案手法は、回転不变性と適応的調整型のパラメータ調整則を同時に有する PSO に分類される点で、新しい。

なお、PSO 以外では ES にパラメータ調整則と変換不变性を同時に付加する研究があり、その中でも Hansen らの CMA-ES [88, 89] は強力なメタヒューリスティクスとして代表的である。国際会議 ACM GECCO のワークショップ「Black-Box-Optimization-Benchmarking (BBOB)」[120] や IEEE CEC のスペシャルセッション「Competition on Real-Parameter Single Objective Optimization」[123] では、BBO のコンペティションが開催されるが、CMA-ES ベースの手法や適応型 DE は上位に位置している [121]。本節の提案手法は CMA-ES との類似点からも優れた探索性能が期待できる。

5.4.2 項では、CRI-PSO のパラメータ解析を行い、パラメータと多様化・集中化の関係を明らかにする。5.4.3 項では、回転不变性を有する適応型 PSO として、適応型 CRI-PSO を提案する。5.4.4 項では、適応型 CRI-PSO の変換不变性の解析を行う。5.4.5 項では、数值実験を通じて、適応型 CRI-PSO のロバスト性・適応性を検証する。

図 5.12：PSO の探索過程における I と $f(\mathbf{p}^g)$ の推移 ($N = 50$ 、Ackley 関数)

5.4.2 CRI-PSO のパラメータ解析

本項では、数値実験を通じて、多様化・集中化の評価指標を用いて、慣性パラメータ w と CRI-PSO の探索状態の関係を明らかにする。多様化・集中化の評価指標として、AFPSO で使用されている活性度 P (式 (5.29)) を使用する。活性度 P が大きければ多様化の状態、活性度 P が小さければ集中化の状態と対応付けて、探索状態を評価できる。Yasuda らは、活性度 P を用いることで、数値実験的に安定・不安定領域 (図??) を描画でき、多様化・集中化を評価できることを示した[116]。このため、PSO と同様に、CRI-PSO に対しても、

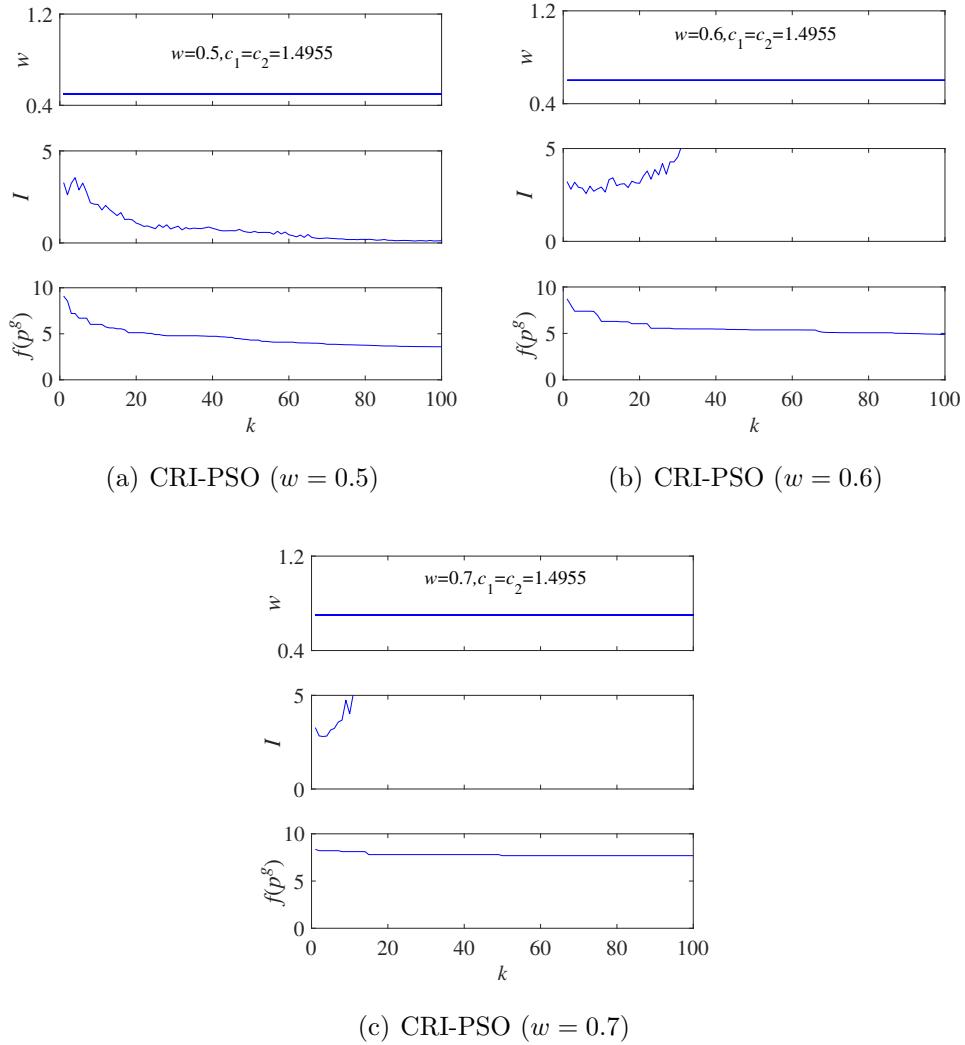


図 5.13： CRI-PSO の探索過程における I と $f(\mathbf{p}^g)$ の推移 ($N = 50$ 、Ackley 関数)

探索状態の評価が可能であると考えられる。

PSO と同様に CRI-PSOにおいて、 w の値によって、活性度 P の推移が異なることを数值実験的に検証する。共通の実験条件として、対象関数は Ackley 関数（付録 C を参照）、探索点数 $m = 20$ 、次元数 $N = 50$ 、PSO のパラメータ $c_1 = c_2 = 1.4955$ 、終了条件を反復回数 $k_{\max} = 100$ とする。

図?? (a), (b), (c) に、PSO の $w = 0.5, 0.75, 1.0$ に設定した場合の活性度 P と g-best の目的関数値 $f(\mathbf{p}^g)$ の推移を示す。これらの図から下記の様子が確認できる。

- $w = 1.0$ の場合, 活性度 P は探索序盤から大きな値に発散していく。
- $w = 0.75$ の場合, 活性度 P は探索過程で発散しないが, まだ収束せずに探索点群の動きが続く。
- $w = 0.5$ の場合, 活性度 P は探索過程で徐々に収束する。

これらは, 図??の PSO の安定／不安定領域と対応付けて解釈ができる。

図?? (a), (b), (c) に, CRI-PSO の $w = 0.5, 0.6, 0.7$ に設定した場合の活性度 P と g-best の目的関数値 $f(\mathbf{p}^g)$ の推移を示す。これらの図から下記の様子が確認できる。

- $w = 0.6, 0.7$ の場合, 活性度 P は探索序盤から大きな値に発散していく。
- $w = 0.5$ の場合, 活性度 P は探索序盤から徐々に収束し, そのまま推移し続ける。
さらに, 他のパラメータ設定に比べて, $f(\mathbf{p}^g)$ が大きく改善する。

これらは, ① w が大きい場合, 探索序盤から多様化の状態が続くこと, ② w が小さい場合, 「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たことで, 効率の良い探索ができていること, を示している。ただし, 著者らは事前の数値実験を通じて, 他の関数においても同様の傾向が得られることと, 次元数 N が変わると活性度 P の収束速度が多少異なること, を確認している。

したがって, ① 活性度 P が CRI-PSO の探索状態を評価できること, ② CRI-PSO においては, $(w, c = c_1 = c_2) \in [(0.5, 1.4955), (0.6, 1.4955)]$ の範囲で, 安定／不安定領域の境界線が存在すること, がわかる。よって, 探索状態が多様化から集中化に遷移する, つまり活性度 P が大きな値から徐々に減少するように, パラメータ w を安定／不安定領域の境界線を跨ぐ範囲で適応的に調整することで, CRI-PSO の適応性の向上が期待できる。なお, 本節は適応型 CRI-PSO の基礎的検討であるため, パラメータ空間の安定／不安定領域の境界線はその一部だけが確認できれば十分である。図??と同様の, CRI-PSO のパラメータ空間の安定／不安定領域に関する解析は今後の課題とする。

5.4.3 適応型 CRI-PSO

(a) 探索状態の評価と制御に基づくパラメータ調整則

本項では、適応型 CRI-PSO の開発にあたって、パラメータ w の調整則を設計する。パラメータ w の調整則を有する CRI-PSO の更新則を式 (5.66), 式 (5.67) に示す。

$$\begin{aligned}\mathbf{v}^i(k+1) &= w(k)\mathbf{v}^i(k) + c_1 \mathbf{P} \mathbf{R}_1 \mathbf{P}^T (\mathbf{p}^i(k) - \mathbf{x}^i(k)) \\ &\quad + c_2 \mathbf{P} \mathbf{R}_2 \mathbf{P}^T (\mathbf{p}^g(k) - \mathbf{x}^i(k))\end{aligned}\quad (5.66)$$

$$\mathbf{x}^i(k+1) = \mathbf{x}^i(k) + \mathbf{v}^i(k+1) \quad (5.67)$$

これらを連立すると、更新則 $U_{f,S}$ として式 (5.68) が得られる。

$$\begin{aligned}\mathbf{x}^i(k+1) &= \mathbf{x}^i(k) + w(k)\mathbf{v}^i(k) + c_1 \mathbf{P} \mathbf{R}_1 \mathbf{P}^T (\mathbf{p}^i(k) - \mathbf{x}^i(k)) \\ &\quad + c_2 \mathbf{P} \mathbf{R}_2 \mathbf{P}^T (\mathbf{p}^g(k) - \mathbf{x}^i(k))\end{aligned}\quad (5.68)$$

ただし、式 (5.68) は式 (4.10) の w を時系列表記 $w(k)$ としただけである。

w の調整則として、探索状態の評価と制御に基づくパラメータ調整則である AFPSO の式 (5.28) を用いる。この調整則 F を式 (5.69) に再掲する。

$$w(k+1) = \begin{cases} \min\{w(k) + \Delta w, w_{\max}\}, & P(k) \leq P_{\text{target}}(k) \\ \max\{w(k) - \Delta w, w_{\min}\}, & \text{otherwise} \end{cases} \quad (5.69)$$

式 (5.69) は、事前に評価指標の目標値 $P_{\text{target}}(k)$ ($k = 1, 2, \dots, k_{\max}$) を与えておき、各反復回数 k における評価指標 $P(k)$ の値と比較し、 P が P_{target} に追従するように調整幅 $\Delta w > 0$ で w を調整する。

これ以降は有効な探索戦略と変換不变性の観点から、式 (5.69) のパラメータ調整則の詳細を設計する。なお、 w の調整則には、新たに下限値 w_{\min} 、上限値 w_{\max} 、調整幅 Δw というパラメータが導入されている。 Δw は、評価指標 P が目標値 P_{target} を十分追従できる値に設定すればよい。また、 w_{\min} 、 w_{\max} は 5.4.2 項で述べた通り、パラメータ空間における安定／不安定領域の境界線を跨ぐように設定すればよい。よって、これらの推奨値は、より詳細な安定性解析に基づくことが望ましいが、ここでは基礎的な検討とするため、5.4.5 項では著者らの数値実験に基づく推奨値とし、より詳細な安定性解析に基づくパラメータの設定は今後の課題とする。

(b) 探索戦略に基づくパラメータ調整

式(5.67)の w の調整則では、目標値 P_{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。また、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を目指す」戦略が有効であることが知られている。このため、目標値 P_{target} として探索過程で一様に減少する形状を使用することで、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。本節では、目標値 P_{target} として、AFPSOと同様に、一様に減少する最も単純なスケジュールの一つである指数減少スケジュール（式(5.30)）を使用する¹⁰。指数減少スケジュールを図??に示す。この調整則は、許容される反復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、最大反復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。これまでにYasudaらは、PSOとDEに対して、多様化・集中化の観点から、同様のパラメータ調整則を付加することで、適応性・探索性能の向上が可能であることを確認した[116, ?]。よって、本論文においても同様に、パラメータ調整則の付加によって、CRI-PSOの適応性・探索性能の向上が期待できる。

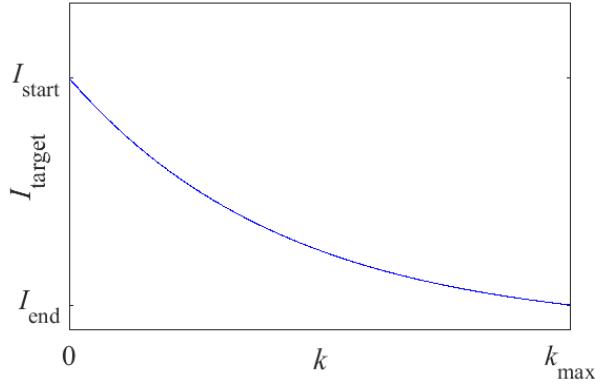
なお、目標値 P_{target} では、新たに P_{start} , P_{end} というパラメータが導入されている。探索戦略の観点から、 P_{start} を多少大きく、 P_{end} を十分小さく設定すればよい。本章では、著者の数値実験に基づき、提案手法のパラメータ推奨値は、 $P_{\text{start}} = \epsilon_{\text{start}} x_{\text{width}}$, $P_{\text{end}} = \epsilon_{\text{end}} x_{\text{width}}$, $\epsilon_{\text{start}} = 0.25$, $\epsilon_{\text{end}} = 0.001$ とする。 x_{width} は式(5.70)で表される。

$$x_{\text{width}} = \frac{1}{\sqrt{N}} \| \gamma_{\max} - \gamma_{\min} \|_2 \quad (5.70)$$

ただし、初期配置領域 $S_0 = [\gamma_{\min,n}, \gamma_{\max,n}]^N$ である¹¹。また、これらは、パラメータ調整

¹⁰ 本節では基礎的検討として、指数減少スケジュールを使用するが、これ以外のスケジュールとして、線形減少スケジュールや、探索過程で値の減少を周期的に繰り返す（のこぎり波など）スケジュールも考えられる。

¹¹ 本論文では初期配置領域が解空間の変換の影響を受けることにしており、ただし、これは本来ユーザが決定するパラメータであるため、変換不变性の維持が目的であれば、初期の探索点位置から決定しても良い。例えば、初期探索点の平均ベクトルを計算し、平均ベクトルから最も離れている探索点を選ぶと同時に、平均ベクトルに関して対称な位置ベクトルを選び、それらの差分ベクトルの L^2 ノルムから決定すれば上記と同じの不变性を具備する。

図 5.14：目標値スケジュール I_{target}

則を有するアルゴリズムの変換不变性の有無に影響を与えるため、後で述べる条件式を満たすように設定すればよい。

5.4.4 変換不变性に基づく適応型 CRI-PSO の解析

(a) パラメータ調整則を有する CRI-PSO の変換不变性の証明

4.4 節では、ロバスト性向上のために、PSO に回転不变性を付加する一方で、本節では、適応性向上のために、回転不变性を有する CRI-PSO にパラメータ調整則を付加する。5.2 節で述べたように、変換不变性に基づくパラメータ調整則の指針は 2 つあるが、本節では、指針(1)「更新則は変換不变性を具備するとき、その変換不变性を維持するように、その変換不变性を有するパラメータ調整則を付加する」に従って、パラメータ調整則を設計する。このため、前項のパラメータ調整則が、証明によって、CRI-PSO の解空間の相似変換不变性、回転不变性、平行移動不变性、目的関数の単調増加変換不变性を維持するのかを確認する。

定義 5.1 に従うと、式(5.69)のパラメータ調整則 $F_{f,S}$ において、 $P(k), P_{\text{target}}$ は探索履歴 $\chi_S(k)$ に依存し、 $w_{\min}, w_{\max} \in \mathcal{C}_F(k)$ となる。

パラメータ調整則 $F_{f,S}$ を有する CRI-PSO の更新式 $U_{f,S}$ がある変換 T に対する不变性を有する条件について、定理 5.1 が成り立つ。

定理 5.1 (パラメータ調整則 $F_{f,S}$ を備えた CRI-PSO が変換不变性を有するための条件) CRI-PSO の更新式 $U_{f,S}$ が変換 T に対する不变性を有するとき, パラメータ調整則 $F_{f,S}$ を備えた CRI-PSO の更新則 $U_{f,S}$ がある変換 T に対する不变性を有するための条件は, 式 (5.71) が成立することである。

$$P'(k) \leq P'_{\text{target}}(k) \quad (5.71)$$

ただし, $P(k) \leq P_{\text{target}}(k)$ とし, $P'(k), P'_{\text{target}}(k)$ は, 変換 T を与えたときの $P(k), P_{\text{target}}(k)$ である。

証明. パラメータ調整則 $F_{f,S}$ を備えた更新則 $U_{f,S}$ は, 更新則 $U_{f,S}$ の w を時系列表記 $w(k)$ に置き換えただけである。よって, 更新則 $U_{f,S}$ が変換 T に対する不变性を有するとき, パラメータ調整則 $F_{f,S}$ を備えた更新式 $U_{f,S}$ は, 変換 T によってパラメータ調整則 $F_{f,S}$ が影響を受けなければ, 変換 T に対する不变性を有する。

定義 5.2 から, パラメータ調整則 $F_{f,S}$ において, パラメータの不变性条件 (式 (5.50)) が成立するための条件を導出すればよい。式 (5.69) のパラメータ調整則 $F_{f,S}$ において, $P(k)$ と $P_{\text{target}}(k)$ が変換 T によって変化しなければよい。つまり, 式 (5.72), 式 (5.73) が成立すればよい。

$$P'(k) = P(k) \quad (5.72)$$

$$P'_{\text{target}}(k) = P_{\text{target}}(k) \quad (5.73)$$

しかしながら, 式 (5.69) のパラメータ調整則 F_f において, これらの量は $P(k) \leq P_{\text{target}}(k)$ という不等式として使用されている。よって, 変換 T によってこの大小関係が不变であっても, パラメータの不变性条件を満たす。前提条件から $P(k) \leq P_{\text{target}}(k)$ であり, 変換 T を与えたとき, 式 (5.71) が成立するとしているから, 大小関係は不变であることを指している。よって, 式 (5.71) が成立するならば, パラメータ調整則 $F_{f,S}$ について式 (5.51) が成立することが導出された。したがって, 題意が示された。 \square

また、パラメータ調整則 $F_{f,S}$ を有する適応型 CRI-PSO に関して、定理 5.2、定理 5.3、定理 5.4 を示すことで、解空間の回転不变性・相似変換不变性・平行移動不变性・目的関数の単調増加変換不变性を有することを明らかにする。ただし、解空間のスケール不变性を失っていることも明らかにする。

定理 5.2 (適応型 CRI-PSO の解空間の回転不变性・相似変換不变性・平行移動不变性に関する定理) 目標値 P_{target} が式 (5.74) を満たすとき、適応型 CRI-PSO の更新則 $U_{f,S}$ は解空間の回転不变性・相似変換不变性・平行移動不变性を有する。

$$P'_{\text{target}}(k) = \delta P_{\text{target}}(k) \quad (5.74)$$

ただし、 $P'_{\text{target}}(k)$ は変換 T_s を与えたときの $P_{\text{target}}(k)$ 、回転変換・相似変換・平行移動の合成変換は $T_s : \mathbf{x} \mapsto \delta \mathbf{C} \mathbf{x} - \mathbf{t}$, $\forall \delta > 0, \mathbf{t} \in \mathbb{R}^N$ であり、 $\mathbf{C} \in \mathbb{R}^{N \times N}$ は直交行列である。

証明. まず、CRI-PSO の更新則 $U_{f,S}$ は解空間の回転不变性、相似変換不变性、平行移動不变性を有する (定理 4.19)。また、このとき、式 (5.71) が成立するならば、パラメータ調整則 $F_{f,S}$ を備えた CRI-PSO の更新則 $U_{f,S}$ は解空間の回転不变性、相似変換不变性、平行移動不变性を有する (定理 5.1)。よって、変換 $T_s : \mathbf{x} \mapsto \delta \mathbf{C} \mathbf{x} - \mathbf{t}$ が与えられ、式 (5.74) が成立するとき、式 (5.71) の成立を導出できれば、題意は成り立つ。

解空間の回転変換・相似変換・平行移動の合成変換 T_s のもとで、パラメータ調整則 $F_{f,S}$ の引数の影響を考える。変換 T_s を与えたとき、 $P'(k)$ は

$$\begin{aligned} P'(k) &= \frac{1}{m\sqrt{N}} \sum_{i=1}^m \| \mathbf{v}^{ii}(k) \|_2 \\ &= \frac{1}{m\sqrt{N}} \sum_{i=1}^m \| \delta \mathbf{C} \mathbf{v}^i(k) \|_2 \\ &= \delta \frac{1}{m\sqrt{N}} \sum_{i=1}^m \| \mathbf{v}^i(k) \|_2 \\ &= \delta P(k) \end{aligned}$$

となる。ただし、計算過程で直交行列の性質 $\| \mathbf{C} \mathbf{v}^i(k) \|_2 = \| \mathbf{v}^i(k) \|_2$ を用いた。さらに、

式 (5.74) が成立するから,

$$\begin{aligned} P'_{\text{target}}(k) - P'(k) &= \delta P_{\text{target}}(k) - \delta P(k) \\ &= \delta (P_{\text{target}}(k) - P(k)) \geq 0 \end{aligned}$$

となり、式 (5.71) を満たすことが導出される。したがって、題意が示された。 \square

なお、目標値 P_{target} は、初期配置領域 S_0 に属するベクトルの L^2 ノルムから設定すれば、式 (5.72) の条件式を満たす。このため、5.4.3 項における設定が、式 (5.72) を満たすこととは、補助定理 5.1 から導出される。

補助定理 5.1 解空間の回転変換・相似変換・平行移動の合成変換 T_s が与え、式 (5.28) の目標値 P_{target} に含まれるパラメータを式 (5.70) と設定したとき、式 (5.74) が成り立つ。

証明. 解空間の回転変換・相似変換・平行移動の合成変換 T_s のもとで、パラメータ調整則 $F_{f,S}$ の引数の影響を考える。 γ_{\min} , γ_{\max} は解空間に属するベクトルなので、変換 T_s を与えた後、 $\gamma'_{\min} = \delta C \gamma_{\min} - t$, $\gamma'_{\max} = \delta C \gamma_{\max} - t$ となる。同様に、 x'_{width} は

$$\begin{aligned} x'_{\text{width}} &= \frac{1}{\sqrt{N}} \| \gamma'_{\max} - \gamma'_{\min} \|_2 \\ &= \frac{1}{\sqrt{N}} \| (\delta C \gamma_{\max} - t) - (\delta C \gamma_{\min} - t) \|_2 \\ &= \delta \frac{1}{\sqrt{N}} \| C(\gamma_{\max} - \gamma_{\min}) \|_2 \\ &= \delta \frac{1}{\sqrt{N}} \| \gamma_{\max} - \gamma_{\min} \|_2 \\ &= \delta x_{\text{width}} \end{aligned}$$

となる。ただし、計算過程で直交行列の性質 $\| C(\gamma_{\max} - \gamma_{\min}) \|_2 = \| \gamma_{\max} - \gamma_{\min} \|_2$ を

用いた。よって、パラメータ P'_{start} , P'_{end} と、目標値 P'_{target} は、

$$\begin{aligned} P'_{\text{start}} &= \varepsilon_{\text{start}} x'_{\text{width}} = \delta \varepsilon_{\text{start}} x_{\text{width}} = \delta P_{\text{start}}, \\ P'_{\text{end}} &= \varepsilon_{\text{end}} x'_{\text{width}} = \delta \varepsilon_{\text{end}} x_{\text{width}} = \delta P_{\text{end}}, \\ P'_{\text{target}}(k) &= P'_{\text{start}} \left(\frac{P'_{\text{end}}}{P'_{\text{start}}} \right)^{\frac{k}{k_{\max}}} \\ &= \delta P_{\text{start}} \left(\frac{\delta P_{\text{end}}}{\delta P_{\text{start}}} \right)^{\frac{k}{k_{\max}}} \\ &= \delta P_{\text{start}} \left(\frac{P_{\text{end}}}{P_{\text{start}}} \right)^{\frac{k}{k_{\max}}} \\ &= \delta P_{\text{target}}(k) \end{aligned}$$

となる。以上から、題意が示された。 \square

定理 5.3 (適応型 CRI-PSO の解空間のスケール不変性に関する定理) 適応型 CRI-PSO の更新式 $U_{f,S}$ は解空間のスケール不変性を有さない。ただし、スケール変換は $T_s : \mathbf{x} \mapsto \delta \mathbf{C} \mathbf{x}$, $\mathbf{C} \in \mathbb{R}^{N \times N}$ は対角行列である。

証明. CRI-PSO の更新則 $U_{f,S}$ は解空間のスケール不変性を有さない (定理 4.20)。また、このとき、パラメータ調整則 $F_{f,S}$ に関して式 (5.71) が成立するならば、パラメータ調整則 F_f は不変性定理が成立する (定理 5.1)。よって、変換 $T_s : \mathbf{x} \mapsto \delta \mathbf{C} \mathbf{x} - \mathbf{t}$ が与えられたとき、式 (5.71) が成立しないことを導出できれば、題意は成り立つ。

解空間のスケール変換 T_s のもとで、更新則 $U_{f,S}$ の引数の影響を考える。変換 T_s を与えたとき、 $P'(k)$ は

$$\begin{aligned} P'(k) &= \frac{1}{m\sqrt{N}} \sum_{i=1}^m \| \mathbf{v}^{i\prime}(k) \|_2 \\ &= \frac{1}{m\sqrt{N}} \sum_{i=1}^m \| \mathbf{C} \mathbf{v}^i(k) \|_2 \\ &\neq \frac{1}{m\sqrt{N}} \sum_{i=1}^m \| \mathbf{v}^i(k) \|_2 \\ &= P(k) \end{aligned}$$

となる。つまり、式(5.72)が成立しない。よって、 $P_{\text{target}}(k)$ がどのような式でも、 $\mathbf{C} = \delta \mathbf{I}$ でない限り、式(5.71)は成立しない。ただし、ここでは \mathbf{C} は対角行列である。したがって、題意が示された。□

定理 5.4 (適応型 CRI-PSO の目的関数の単調増加変換不变性に関する定理) 適応型 CRI-PSO の更新式 $U_{f,S}$ は目的関数の単調増加変換不变性を有する。

証明. パラメータ調整則 $F_{f,S}$ において目的関数値を使用していないことから、パラメータの不变性定理が成立する。また、適応型 CRI-PSO の更新則は、CRI-PSO の更新則を式(4.8)から式(5.68)に変更したもので、 w を $w(k)$ と置き換えただけである。この変更において目的関数値は無関係である。よって、PSO の定理 4.5 と全く同じ手順で式(4.1)を満たすことが導出できる。したがって、題意が示された。□

(c) 適応型 CRI-PSO

本章では適応型 CRI-PSO を提案する。適応型 CRI-PSO は、CRI-PSO に探索状態の評価と制御に基づくパラメータ w の調整則を付加したものである。適応型 CRI-PSO の更新則は式(5.66)と式(5.67)、 w の調整則は式(5.69)であり、他の式は CRI-PSO と同様である。Algorithm 5.4.4 に適応型 CRI-PSO のアルゴリズムを示す。初期の慣性パラメータ $w(1)$ を $w(1) = w_{\max}$ とする。

適応型 CRI-PSO の特徴を下記に示す。

- 有効な探索戦略（多様化・集中化）に基づくパラメータ調整則を CRI-PSO に付加することで、適応型 CRI-PSO は高い適応性・探索性能が期待できる。
- CRI-PSO は回転不变性が付加された PSO だが、パラメータ調整則の付加により、その変換不变性を失うことがある。そこで、探索構造の変換不变性が維持されるようにパラメータ調整則を付加することで、CRI-PSO は変換不变性を失わず、適応型 CRI-PSO は高いロバスト性が期待できる。

既存のアルゴリズムとの違いについて記述する。PSO のアルゴリズムは、algorithm 4.4.2 の CRI-PSO のアルゴリズムの Step 2において、 $\mathbf{P} = \mathbf{I}$ とすれば一致する。ただし、

Algorithm 16 適応型 CRI-PSO

```

1: procedure ADAPTIVE CRI-PSO           13:       $\mathbf{v}^i(k)$  を式 (5.66) を用いて更新す
   Step 0 : パラメータ設定             る。
2:   パラメータ  $m, k_{\max}, c_1, c_2, w_{\max}, w_{\min}, \Delta w$  を設定する。          14:       $\mathbf{x}^i(k)$  を式 (5.67) を用いて更新
   Step 1 : 初期化                   15:       $\mathbf{p}^i(k)$  を式 (3.25) を用いて更新
3:    $\mathbf{x}^i(k)$  を  $S_0$  内にランダムに配置する。          16:      end for
4:    $\mathbf{v}^i(k)$  をランダムに与える。          17:       $\mathbf{p}^g(k)$  を式 (2.16) を用いて更新する。
5:   p-best を  $\mathbf{p}^i(k) = \mathbf{x}^i(k)$  とする。          18:       $P(k)$  を式 (5.29) を用いて更新する。
6:   g-best を  $\mathbf{p}^g(k)$  を式 (2.16) を用いて計          19:       $w(k)$  を式 (5.69) を用いて更新する。
   算する。          Step 4 : パラメータの更新
7:    $P_{\text{target}}(k)$  を与える。          20:      if  $k \geq k_{\max}$  then
8:    $w(k) = w_{\max}$  とする。          21:      計算を終了する。
9:    $k = 1$  とする。          22:      else
   Step 2 : 座標変換行列の計算          23:       $k := k + 1$  とし、Step 2 へ戻る。
10:  共分散行列  $\Sigma(\mathcal{Z})$  を固有値分解する。          24:      end if
11:   $\mathbf{P}$  を式 (4.9) を用いて計算する。          25: end procedure
   Step 3 : 速度と位置の更新
12:  for  $i = 1, 2, \dots, m$  do

```

$\mathbf{I} \in \mathbb{R}^{N \times N}$ は単位行列である。同様に, Activity Feedback PSO (AFPSO) [116]のアルゴリズムは, algorithm 5.4.4 の適応型 CRI-PSO のアルゴリズムの **Step 2**において, $\mathbf{P} = \mathbf{I}_N$ とすれば一致する。AFPSO も目標値 P_{target} として式 (??) を用いており, そのパラメータも $P_{\text{start}} = 0.25x_{\text{width}}, P_{\text{end}} = 0.001x_{\text{width}}$ としている。ただし, x_{width} として

$$\begin{aligned}
x_{\text{width}} &= |x_{\max} - x_{\min}|, \\
x_{\max} &= \max\{\gamma_{\max,n} \mid n = 1, 2, \dots, N\}, \\
x_{\min} &= \min\{\gamma_{\min,n} \mid n = 1, 2, \dots, N\}
\end{aligned}$$

を用いており, 適応型 CRI-PSO とは異なる。それ以外のパラメータは適応型 CRI-PSO と

同様である。なお、適応型 CRI-PSOにおいて上記の x_{width} を使用しない理由は、 L^1 ノルムは回転不变性の条件を満たさず、回転不变性を維持できないためである。このため、補助定理 5.1 で示したように、適応型 CRI-PSO では回転不变性を維持できる L^2 ノルムの x_{width} を使用した。

5.4.5 数値実験検証

(a) 共通の実験条件

典型的なベンチマーク関数を用いた数値実験を行い、適応型 CRI-PSO（提案手法）の探索性能を検証する。適応型 CRI-PSO と比較する手法は、オリジナル PSO, Activity Feedback PSO (AFPSO) [116], CRI-PSO とする。各手法の共通条件として、探索点数 $m = 20$, PSO のパラメータ $c_1 = c_2 = 1.4955$, 関数毎に共通の初期配置領域 S_0 を用いる。また、各手法のパラメータは、オリジナル PSO では $w = 0.729$ (CM [176, 177]), CRI-PSO では $w = 0.5$, AFPSO と適応型 CRI-PSO では、下限値 $w_{\min} = 0.5$, 上限値 $w_{\max} = 1.0$, 調整幅 $\Delta w = 0.1$, 評価指標の目標値スケジュール P_{target} は図??の指数減少スケジュールを用いる。初期配置領域 S_0 内に初期探索点をランダムに配置した後、終了条件を満たすまで探索を進める試行を 50 回行い、50 試行の g-best の目的関数値 $f(\mathbf{p}^g(k_{\max}))$ の平均値を実験結果とする。数値実験で用いるベンチマーク関数については、付録 C を参照されたい。

なお、CRI-PSO と適応型 CRI-PSO において、 \mathcal{Z} として任意の解集団を設定可能だが、4.4 節の数値実験を通じて、 $\mathcal{Z} = \mathcal{P}(k)$ と設定する場合が最も優れていることを確認した。このため、本章においても $\mathcal{Z} = \mathcal{P}(k)$ と設定する。ただし、 $\mathcal{P}(k)$ は各反復回数における p-best 集合である。

(b) アフィン変換に対するロバスト性に関する検証

数値実験を通じて、適応型 CRI-PSO のアフィン変換（回転変換を含む）に対する探索性能のロバスト性を検証する。本実験では、解空間にアフィン変換の一種である平行移動変換、スケール変換、回転変換を与えていき、その探索性能の変動を確認する。ベンチマーク関数として、変数分離可能・良スケール性・原点に最適解が存在・多峰性の 4 種類の目

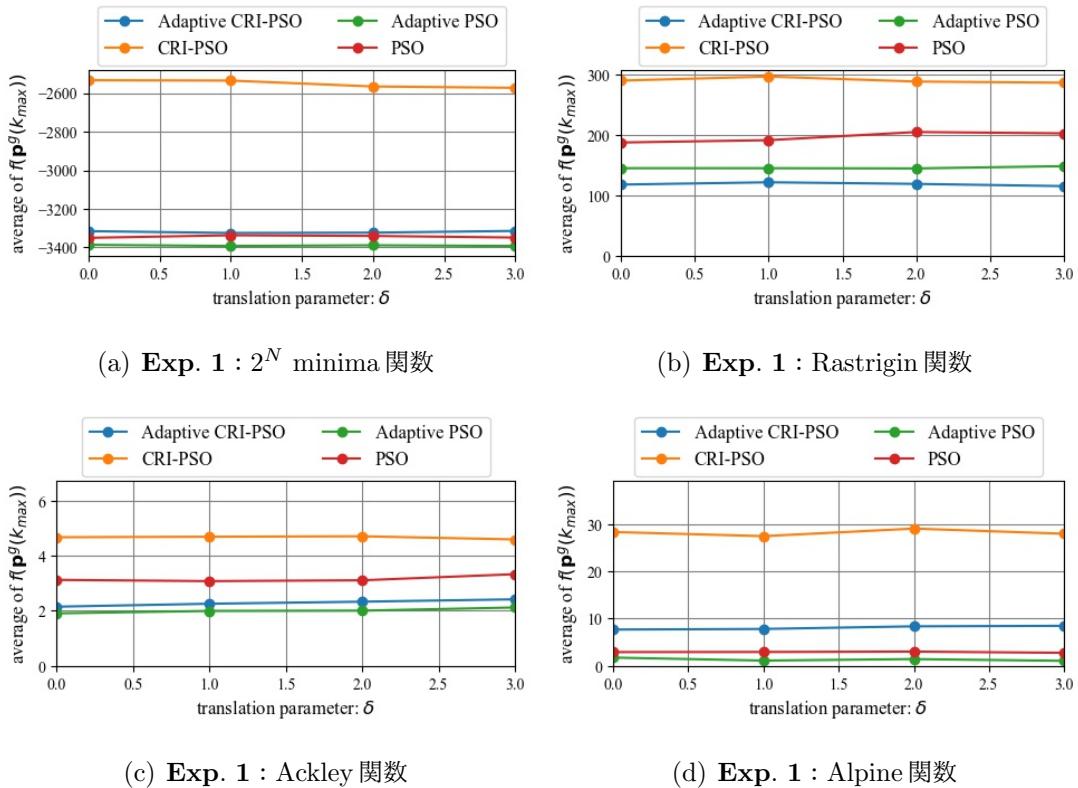


図 5.15： 数値実験結果（Exp. 1）：4 種類の PSO の探索性能のロバスト性 ($N = 50$ 、 $k_{\max} = 1000$)

的関数を使用する。実験方法は、4.4.4 項と同様である。共通条件として、次元数 $N = 50$ 、終了条件を反復回数 $k_{\max} = 1000$ とする。

図 5.15, 図 5.16, 図 5.17 に数値実験結果を示す。各図の横軸は、平行移動パラメータ δ 、スケール変換パラメータ ε 、回転角 θ であり、全ての図の縦軸が探索性能 ($f(\mathbf{p}^g(k_{\max}))$) の平均値) である。図 5.15, 図 5.16, 図 5.17 より、下記の結果が確認できる。

- 解空間の平行移動変換に対して、4 種類の PSO は探索性能の変動が小さい。
- 解空間のスケール変換に対して、オリジナル PSO と CRI-PSO は探索性能の変動が小さいことに対して、AFPSO と適応型 CRI-PSO は、探索性能の変動が大きい。
- 解空間の回転変換に対して、オリジナル PSO と AFPSO は探索性能の変動が大きいことに対して、CRI-PSO と適応型 CRI-PSO は、解空間の回転に対して探索性能の

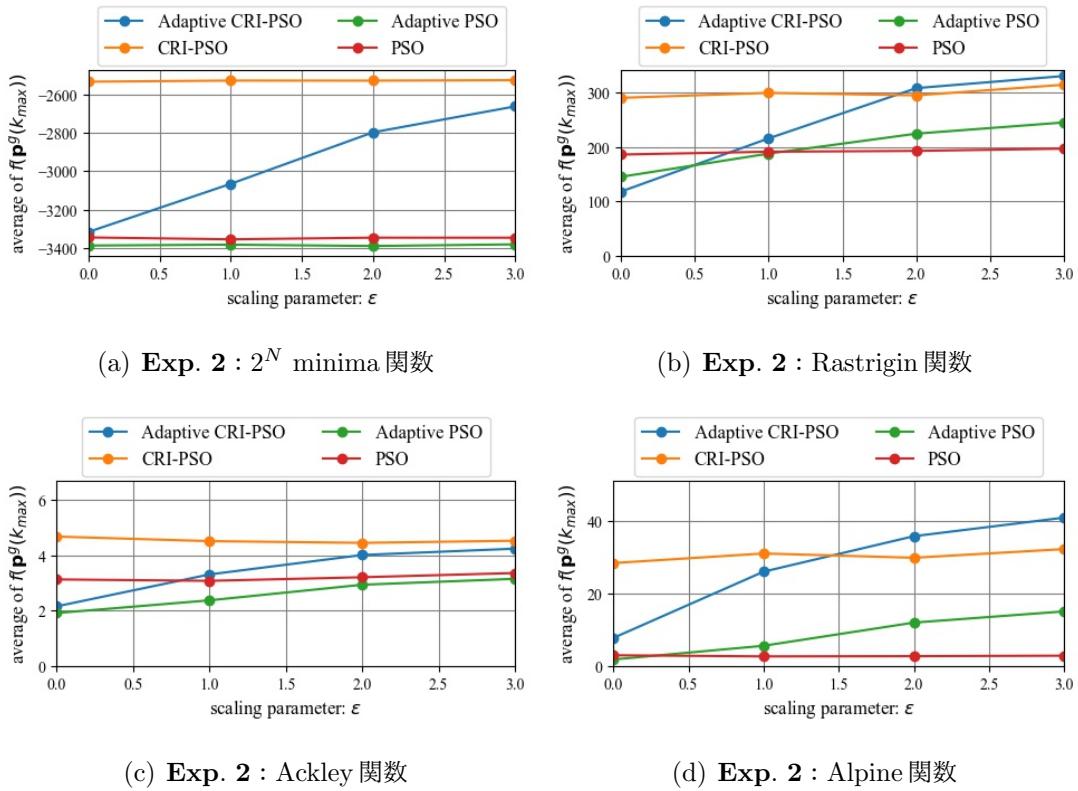


図 5.16 : 数値実験結果 (Exp. 2) : 4 種類の PSO の探索性能のロバスト性 ($N = 50$, $k_{\max} = 1000$)

変動が少ない。

- CRI-PSO は、多くの条件で、探索性能が他の PSO よりも劣っている。
- 適応型 CRI-PSO は、多くの条件で、探索性能が他の PSO よりも優れている。

なお、 $\theta = 90 \text{ deg}$ のときは、ベンチマーク関数の対称性により、 $\theta = 0 \text{ deg}$ と同じ条件である。

図 5.17 の結果は、回転不变性の有無が解空間の回転に対する探索性能のロバスト性に影響を与えることを示している。よって、適応型 CRI-PSO は、パラメータ調整則の付加によって、回転不变性を失っていないため、解空間の回転に対する探索性能のロバスト性も維持していることが示された。一方、図 5.16 の結果は、スケール不变性の有無が解空間のスケール変換に対する探索性能のロバスト性に影響を与えることを示している。このため、

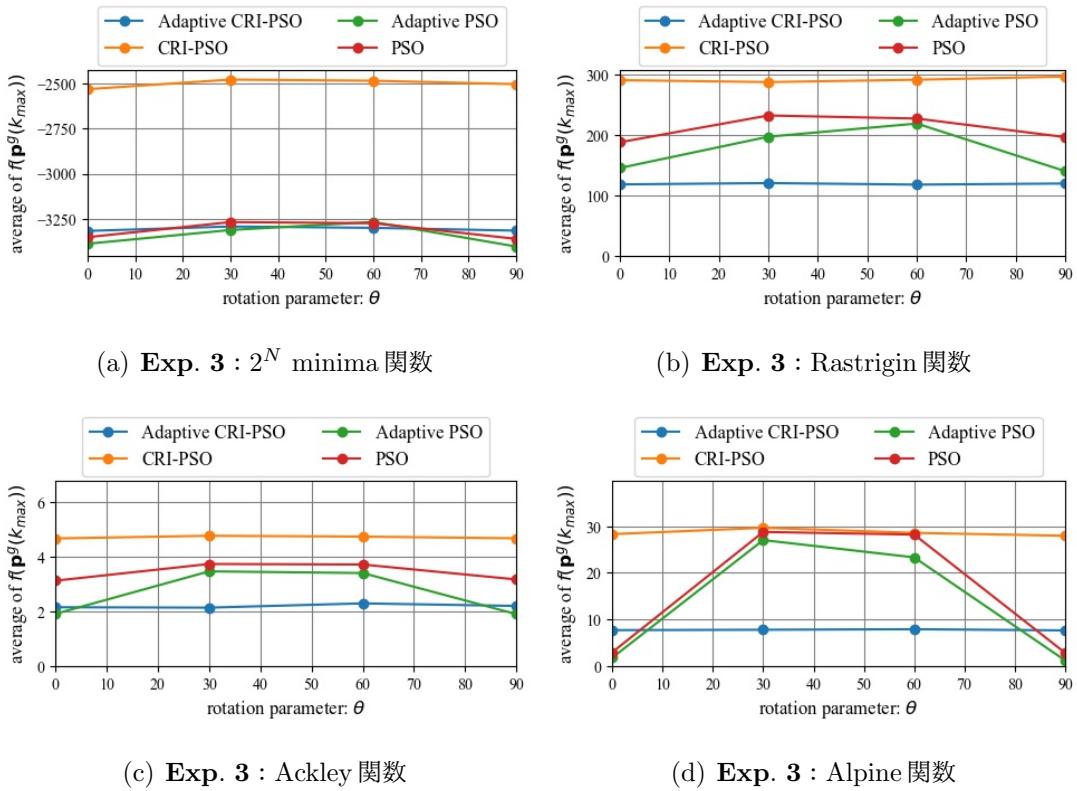


図 5.17： 数値実験結果 (Exp. 3)：4 種類の PSO の探索性能のロバスト性 ($N = 50$ 、 $k_{\max} = 1000$)

適応型 CRI-PSO は、スケール変換に対してはロバスト性を失ったままであることも示している。なお、スケール変換については、CRI-PSO と適応型 CRI-PSO はスケール不变性を失っているにも関わらず、CRI-PSO は適応型 CRI-PSO よりもロバスト性が高いが、適応型 CRI-PSO のパラメータ調整則自身もスケール不变性を失っているため、ロバスト性もより低下したのだと考えられる。しかしながら、CRI-PSO は全ての条件で探索性能に課題があるため、有用性に欠ける。

(c) 全般的な探索性能に関する検証

本実験では、数値実験を通じて、適応型 CRI-PSO の全般的な探索性能を検証する。数値実験には、9 種類のベンチマーク関数を用いる。共通条件として、次元数 $N = 10, 50$,

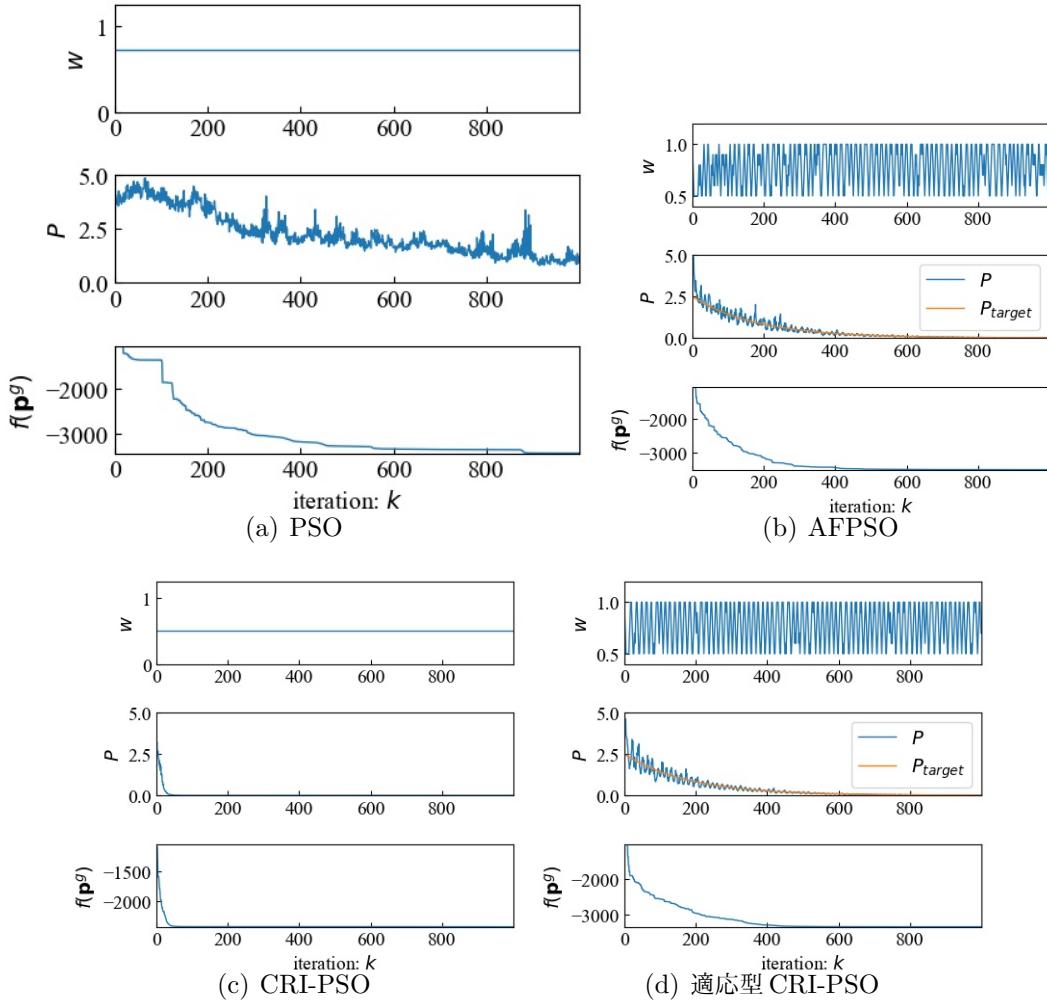


図 5.18：4 種類の PSO の探索過程における $P(k)$ と $f(\mathbf{p}^g(k))$ の推移 ($N = 50, 2^N$ minima 関数)

100, 300, 終了条件を反復回数 $k_{\max} = 100, 1000$ とする。

表 5.4 に $k_{\max} = 100$ の場合の数値実験結果を示し、表 5.5 に $k_{\max} = 1000$ の場合の数値実験結果を示す。表 5.4, 表 5.5 では、4 種類の PSO (オリジナル PSO, AFPSO, CRI-PSO, 適応型 CRI-PSO) の結果を比較して、最も良い結果に (*) を付しており、2 番目に良い結果に (**) を付している¹²。表 5.4, 表 5.5 より、下記の結果が確認できる。

¹² なお、表 5.4, 表 5.5 の一部では、同じ数値にも関わらず優劣がついている結果があるが、これは便宜上、

オリジナル PSO : $N = 10$ においてはAFPSOの次に優れている。 $N = 50, 100, 300$ においては, $k_{\max} = 100$ では最も劣り, $k_{\max} = 1000$ ではAFPSOと適応型CRI-PSOよりも劣っている。

AFPSO : $N = 10$ においては最も優れている。 $N = 50, 100$ においては, $k_{\max} = 100$ ではCRI-PSOと適応型CRI-PSOよりも劣り, $k_{\max} = 1000$ では適応型CRI-PSOの次に優れている。 $N = 300$ においては, CRI-PSOと適応型CRI-PSOよりも劣っている。

CRI-PSO : $N = 10$ においては最も劣っている。 $N = 50, 100$ においては, $k_{\max} = 100$ では最も優れているが, $k_{\max} = 1000$ では最も劣っている。 $N = 300$ においては, $k_{\max} = 100$ では最も優れているが, $k_{\max} = 1000$ では適応型CRI-PSOの次に優れている。

適応型CRI-PSO : $N = 10$ においてはオリジナルPSOとAFPSOよりも劣っている。 $N = 50, 100, 300$ においては, $k_{\max} = 100$ ではCRI-PSOの次に優れており, $k_{\max} = 1000$ では最も優れている。

この結果から, パラメータ固定型(PSO, CRI-PSO)は, 関数の種類, 次元数, 反復回数などの条件の設定によっては高い探索性能を示すが, 様々な条件で行う場合は全体的に探索性能が劣ることがわかる。一方で, パラメータ調整則を有するPSO(AFPSO, 適応型CRI-PSO)は, 様々な条件に関わらず, 全体的に探索性能を発揮できることがわかる。以上から, 適応的調整型のパラメータ調整則を有するPSOは適応性が高い。

また, 図5.18に, Rastrigin関数(回転角 $\theta = 0 \text{ deg}$)を対象としたときの各手法の w , 多様化・集中化の評価指標 P , g-bestの目的関数値 $f(\mathbf{p}^g)$ の推移を示す。なお, 図5.18(b),(d)では, P に重なって目標値 P_{target} のスケジュールが緑の曲線で描かれている。図5.18より, 下記の結果が確認できる。

- PSOとCRI-PSOは, 探索序盤では P が漸減すると同時に, $f(\mathbf{p}^g)$ が改善していく。一方, 探索中盤以降では P が低いままで一様となり, $f(\mathbf{p}^g)$ の更新が停滞する。
- AFPSOと適応型CRI-PSOは, 探索過程で P が目標値 P_{target} に追従することで徐々に漸減すると同時に, $f(\mathbf{p}^g)$ が改善していく。

小数点第三位以下を省略しているためであり, 小数点以下の差異を確認した上で優劣を判断している。

パラメータ固定型（PSO, CRI-PSO）のアルゴリズムは、関数の種類、次元数、反復回数などの探索条件や探索過程の探索状態に応じてパラメータ調整が行われないため、探索戦略が適切に実現されない。特に、CRI-PSO の探索性能が、 $k_{\max} = 100$ で優れているが、 $k_{\max} = 1000$ で比較的劣るのは、本実験のパラメータ設定では、図 5.18(c) の探索序盤 ($0 \leq k < 250$) で十分な集中化が達成され、探索中盤以降 ($250 \leq k$) では過度な集中化が継続するためであると考えられる。ただし、5.4.2 項で述べた通り、事前実験を通じて、解空間の次元数 N が変わると、この傾向が多少異なることを確認しているため、この傾向を明らかにするには詳細な解析が必要であり、今後の課題とする。

一方で、適応的調整型（AFPSO, 適応型 CRI-PSO）のアルゴリズムは、探索戦略に基づく目標値を生成し、 P を目標値に追従させることで、探索戦略を確実に実現している。特に、これらの探索性能が $k_{\max} = 1000$ で比較的優れているのは、目標値を k_{\max} に応じて生成することで、 k_{\max} に関わらず探索戦略の実現を可能とする点が、パラメータ固定型（PSO, CRI-PSO）のアルゴリズムにはないためであると考えられる。

さらに、適応型 CRI-PSO は多くの条件において、AFPSO よりも探索性能が優れている。これは、4.4 節で確認したように、 $\mathcal{Z} = \mathcal{P}(k)$ と設定することで、p-best 集合 \mathcal{P} が探索過程で優れた領域に分布していくにつれて、その分布に沿った基底を生成し、有望領域の重点的な探索を実現しているためだと考えられる。

したがって、有効な探索戦略（多様化・集中化）と変換不变性に基づくパラメータ調整則を付加することで、適応型 CRI-PSO は、解空間の回転に対して高いロバスト性を有することと、様々な探索条件に対して高い適応性を有することが示された。

5.5 まとめ

本章では、高いロバスト性・適応性を具備するメタヒューリスティクスの設計を目指して、相似変換不变性を補完する適応型 CS と、回転不变性を有する適応型 PSO を設計した。本章の成果は、下記の点で新規性を有すると考えられる。

- (1) ロバスト性・適応性の観点から、変換不变性とパラメータ調整則の関係性を指摘し、「パラメータ調整則の不变性」の定義を新たに提示した。また、これを活用す

ることで変換不变性に基づくパラメータ調整則の設計指針を新たに二つ与えた。さらに、代表的なパラメータ調整則が有する量に注目し、その量とパラメータ調整則の不变性の関係性を具体例を用いて確認した。従来では、発見的手法であるメタヒューリスティクスと同様、そのパラメータ調整則も発見的に設計されていたことに対して、本章のように、理論的な不变性の観点から、パラメータ調整則を設計する点で新しい。

- (2) 一つ目の設計指針である「更新則がある不变性を欠如するとき、その不变性を補完するように、その不变性に影響を与えるパラメータ調整則を付加する」に従い、相似変換不变性を欠如しているCSに対して、その不变性に影響を与えながら、多様化・集中化に基づく適応的調整型のパラメータ調整則を付加することで、相似変換不变性を補完した適応型CSを提案した。さらに、数値実験を通じて、適応型CSは相似変換（スケール変換）に対するロバスト性が向上すると同時に、全般的な性能も向上することを確認した。
- (3) 二つ目の設計指針である「更新則が不变性を具備するとき、その不变性を維持するように、その不变性を有するパラメータ調整則を付加する」に従い、4章で提案した回転不变性を有するPSO（CRI-PSO）に対して、その不变性を維持しながら、多様化・集中化に基づく適応的調整型のパラメータ調整則を付加することで、回転不变性を有する適応型PSOを提案した。さらに、数値実験を通じて、回転不变性を有する適応型PSOは回転変換に対するロバスト性を維持すると同時に、全般的な性能も平均的に向上することを確認した。

これらの成果は、第4章の課題(1)(2)の解決に大きく資するものである。

一方で、本章の検討を通じて、下記の課題があることが明らかとなった。

- (1) 両者とも、一部の変換不变性（スケール不变性や回転不变性）を欠如したままで、アフィン変換不变性を具備していない。このため、これらの不变性をさらに付加する必要がある。しかしながら、更新則におけるスケール不变性と回転不变性の欠如は、パラメータ調整則では付加することは困難である。このため、一つ目のパラメータ調整則の設計指針「更新則がある不变性を欠如するとき、その不变性を補完するように、その不变性に影響を与えるパラメータ調整則を付加する」は、現実的に可能な対象は相似変換不变性など限定されていると考えられる。したがって、こ

の課題を解決しながら、高いロバスト性・適応性を具備するメタヒューリスティクスを設計するには、第4章のように、更新則にアフィン変換不变性を付加することでロバスト性を向上させた後、二つ目の設計指針である「更新則が不变性を具備するとき、その不变性を維持するように、その不变性を有するパラメータ調整則を付加する」に従い、適応的調整型のパラメータ調整則を付加することで適応性を向上させるアプローチが有力だと考えられる。

- (2) 設計したアルゴリズムの最適化能力は向上の余地がある。適応型CSの性能はCSに比べて向上したが、回転不变性を有する適応型PSOと比べて、高次元においては全体的に劣っている。これは既存のメタヒューリスティクスの最適化能力の違いに起因すると考えられる。このため、例えば、アフィン変換不变性が保証される範囲で、優秀な最適化能力を自由に組み込めるフレームワークが有効である。他にも、更新則にさらに親和性の高いパラメータ調整則を付加することが挙げられる。これは、本章のように、更新則の数理的構造や多様化・集中化の能力を確認した上で、可能な限り親和性の高いパラメータ調整則を付加する必要がある。例えば、本章では、CRI-PSOの安定・不安定領域はPSOとは異なることを指摘してきたが、その一部しか明らかとなっていない。そこで、[116]と同様に、安定・不安定領域全体を数値実験的に全体を明らかにすることで、さらに有用なパラメータ調整則を設計できると考えられる。一方、このアプローチは、参考にするパラメータ調整則の種類が多い上に、数値実験を通じてその親和性を確認するなど、ユーザにとっては負荷が大きい。そこで、親和性の課題については、上記のアプローチ以外にも、[108, 126, 158]が解決に貢献すると考えられる。これは、パラメータ調整則を構成する演算子などの集合を解とみなし、それを対象関数に対して実行したときの探索性能を得るというプロセスを複数回繰り替えず上位の最適化問題をGenetic Programming(GP)が解くことで、使用したアルゴリズム・対象関数に対して最適なパラメータ調整則を自動的に設計する方法であり、ユーザの負荷を削減すると同時に、最適なパラメータ調整則を使用できる点で有効だと考えられる。ただし、このアプローチを用いる場合、GPの解としてパラメータ調整則が含む量の候補を選ぶが、その量の候補と不变性の影響を考えながら選択する必要がある。
- (3) 第4章の課題(3)「変換不变性の具備を証明によって確認する必要があるが、ユーザにとっては困難である」は、第5章のパラメータ調整則の設計においても、同様

の課題である。

よって、第6章では、第4章の課題(1)(3)、上記の課題(2)(3)（(3)は同じ）を解決することを検討する。

表5.4：数値実験結果（反復回数 $k_{\max} = 100$ ）

関数	N	Orig. PSO (CM)	AFPSO	CRI-PSO	適応型 CRI-PSO
1. Parabola	10	0.00	0.01	0.00 *	0.00 **
	50	43.2	41.8	26.3 **	21.6 *
	100	443	120 **	169	83.9 *
	300	2225	449 **	1086	412 *
2. Rosenbrock	10	8.60	8.35 *	8.42 **	8.88
	50	2236	606	574 **	380 *
	100	29456	1970 **	3759	1570 *
	300	111619	9342 **	35519	7877 *
3. 2^N minima	10	-723 *	-682	-695 **	-685
	50	-1955	-2509	-2944 *	-2730 **
	100	-2285	-4257	-5001 *	-4595 **
	300	-4931	-10197 **	-8123	-10820 *
4. Rastrigin	10	21.3	18.4 **	15.0 *	18.4
	50	460	292	265 **	254 *
	100	1364	766 **	784	653 *
	300	5029	2809 **	3462	2660 *
5. Schwefel	10	0.35	0.07	0.04 **	0.01 *
	50	297	70.7 **	153	68.3 *
	100	1469	314 *	700	320 **
	300	15518	2878 *	7204	2891 **
6. Levy	10	0.03 *	1.04	0.12 **	0.66
	50	40.5	12.7 **	15.4	10.7 *
	100	87.5	21.6 **	30.1	16.6 *
	300	143	33.5 **	63.1	29.3 *
7. Ackley	10	0.17 *	1.22	0.28 **	0.66
	50	5.53	4.73 **	4.75	4.04 *
	100	8.70	5.34 **	6.36	5.05 *
	300	10.1	5.92 **	8.17	5.68 *
8. Griewank	10	0.25	0.12	0.11 *	0.11 **
	50	2.16	2.00	1.64 **	1.54 *
	100	12.3	3.94 **	5.13	3.20 *
	300	56.6	11.7 **	29.0	11.2 *
9. Alpine	10	0.55 **	0.71	0.08 *	0.58
	50	61.2	30.2	25.4 **	21.8 *
	100	191	85.4 **	96.6	68.1 *
	300	794	328 **	490	297 *

表 5.5 : 数値実験結果 (反復回数 $k_{\max} = 1000$)

関数	N	Orig. PSO (CM)	AFPSO	CRI-PSO	適応型 CRI-PSO
1. Parabola	10	0.00 *	0.01	0.00	0.00 **
	50	0.01 **	40.8	0.08	0.00 *
	100	11.6 **	120	12.8	0.03 *
	300	2221	437 **	481	49.7 *
2. Rosenbrock	10	1.31 **	10.2	4.13	0.41 *
	50	99.6	624	84.8 **	47.6 *
	100	1831	2176	531 **	104 *
	300	111776	8829 **	9170	1287 *
3. 2^N minima	10	-727 **	-674	-761 *	-689
	50	-3344 **	-2533	-3386 *	-3316
	100	-2480	-4249	-6451 **	-6573 *
	300	-4898	-10313	-15788 **	-17368 *
4. Rastrigin	10	9.13 **	21.9	5.17 *	16.5
	50	186	291	145 **	118 *
	100	592	752	458 **	250 *
	300	5015	2835	2591 **	1112 *
5. Schwefel	10	0.00 *	0.06	0.00	0.00 **
	50	34.1	69.2	24.9 **	1.02 *
	100	363	324	265 **	54.4 *
	300	4968	2886 **	3333	928 *
6. Levy	10	0.04	1.26	0.02 *	0.02 **
	50	3.83 **	13.4	3.05 *	4.97
	100	16.9	22.1	11.4 **	7.22 *
	300	136	32.8 **	34.8	13.3 *
7. Ackley	10	0.15	1.24	0.00 *	0.07 **
	50	3.13	4.68	1.91 *	2.15 **
	100	5.10	5.33	4.01 **	3.19 *
	300	10.1	5.91 **	6.83	4.58 *
8. Griewank	10	0.09 *	0.12	0.09 **	0.13
	50	0.10 **	2.00	0.13	0.01 *
	100	1.33	4.09	1.30 **	0.13 *
	300	56.6	12.1 **	12.8	2.24 *
9. Alpine	10	0.00 *	0.72	0.00 **	0.11
	50	2.91 **	28.3	1.74 *	7.67
	100	32.6	83.9	21.7 *	29.7 **
	300	618	322	286 **	166 *

6

変換不变性を有するメタヒューリスティクスのフレームワーク

6.1 本章について

第4章では、先行研究と同様に、既存のメタヒューリスティクスに変換不变性を付加するアプローチにより、不变性を有するアルゴリズムを設計すると同時に、このアプローチの課題として、①他の不变性を維持しながら、完全にアフィン変換不变性を付加することが困難であること、②想定した性質（変数間依存性、悪スケール性、原点依存性）に対するロバスト性は向上するが、他の条件に対して有効であるとは限らないため、適応性の付加により、平均的な性能をさらに向上させる必要があること、③各種のアルゴリズムの数理的構造と変換不变性の関係に着目しながら、適切なアプローチを選択することは、アルゴリズムに対する理解が薄いユーザにとって困難であること、を指摘した。この課題を受けて、第5章では、①②の課題解決のために、第4章で設計した不变性を有するアルゴリズムに対して適応的パラメータ調整則を付加することで、ロバスト性を維持しながら適応性の向上が可能であることを確認した。一方、第5章においても、アルゴリズム全体の変換不变性を考慮しながらパラメータ調整則を設計する点で、③の課題は残したままであった。③の課題は、発見的手法であるメタヒューリスティクスの設計自由度が極めて高いため、変換不变性の付加アプローチもアルゴリズム毎の数理的構造に依存し、アルゴリズムのバリエーションまで考慮されていないことが原因である。つまり、変換不变性という数理的性質に基づく、統一的な設計アプローチが存在しないためだと言い換えることができる。したがって、変換不变性を具備するメタヒューリスティクスの設計において、アルゴ

リズムの数理的構造と変換不変性の関連性を明らかにし、最適化能力の設計自由度を残しながらも変換不変性を具備する枠組みを確立することは重要である。

そこで、本章では、①③の課題解決のために、アフィン変換不変性の具備が保証されたアルゴリズムの「フレームワーク」を提案する。また、このフレームワークに対して、4.2節で整理した数理的構造と不変性の関連性を活用し、変換不変性定理を与える。このフレームワークに従えば、アフィン変換不変性を具備することを保証しながら、様々な最適化能力を自由に付与でき、さらにパラメータ調整則を付加することも可能である。つまり、高いロバスト性と適応性の具備が可能な汎用的なフレームワークであるといえる。そのフレームワークに基づき、サンプルアルゴリズムを設計し、数値実験を通じて、変換不変性定理の成立を確認する。最後に、そのフレームワークに基づき、PSOに着想を得たアルゴリズムを設計し、数値実験を通じて、ロバスト性を確認する。

本章の成果は下記の点で新しいと考える。

- アルゴリズムを構築した後に、変換不変性を付加するという既存のアプローチとは逆で、変換不変性の具備が保証されたアルゴリズムのフレームワークを作り、そのフレームワークに基づき新規のアルゴリズムを設計する。
- 特定の変換やアルゴリズムに限らず、より一般性の高いアフィン変換に関する、一般的の反復解法に適用可能な不変性定理を導出する。
- 特定の変換やアルゴリズムに限らず、より一般性の高いアフィン変換に関する、一般的の反復アルゴリズムに適用可能な不変性定理を導出する。

なお、本章は著者の先行研究[226]に基づいている。

6.2 変換不变性を有するメタヒューリスティクスのフレームワーク

6.2.1 共分散行列の固有値分解に基づく変換不变性を有する直接探索法

本節では、変換不变性を有するメタヒューリスティクスのフレームワークを提案する。このフレームワークにおける更新則 $U_{f,S}$ は式 (6.1) で表される。

$$\mathbf{x}^i(k+1) = \mathbf{u}^i(k) + \alpha \boldsymbol{\Sigma}_*^{1/2}(k) \mathbf{s}^i(k) \quad (6.1)$$

ただし、 $\mathbf{s}^i(k) \sim \mathcal{P}$ は摂動を定めるベクトル（乱数ベクトル）、 \mathcal{P} はある N 次元同時確率分布、 $\mathbf{u}^i(k) \in \mathbb{R}^N$ は \mathcal{P} の平均を定めるベクトル（中心ベクトル）、 α はパラメータである。 $\boldsymbol{\Sigma}_*^{1/2}(k)$ は $\boldsymbol{\Sigma}(\mathcal{Z})$ の非対称平方根であり、式 (6.2) で表される（補助定理 A.2）。

$$\boldsymbol{\Sigma}_*^{1/2}(k) = \mathbf{P} \boldsymbol{\Lambda}^{1/2} \quad (6.2)$$

ただし、 $\boldsymbol{\Sigma}(\mathcal{Z})$ は解集団 \mathcal{Z} の共分散行列であり、式 (A.18) で表される。 $\mathbf{P} = [\mathbf{a}_1, \dots, \mathbf{a}_N]$ は基底変換行列、 $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N)$ は固有値行列である。 $\{(\mathbf{a}_1, \lambda_1), (\mathbf{a}_2, \lambda_2), \dots, (\mathbf{a}_N, \lambda_N)\}$ ($\lambda_1 > \lambda_2 > \dots > \lambda_N$) は $\boldsymbol{\Sigma}(\mathcal{Z})$ の固有ベクトルと固有値であり、 $\boldsymbol{\Sigma}(\mathcal{Z})$ の固有値分解によって得られる。補助定理 A.2 は付録 A を参照されたい。

このフレームワークを、共分散行列の固有値分解に基づく直接探索法（Direct Search based on eigenvalue Decomposition of Covariance Matrix : CMD-DS）と呼ぶこととする。CMD-DS の更新則 U_f （式 (6.1)）を定義 4.1 と照らし合わせると、 $\mathbf{u}^i(k), \boldsymbol{\Sigma}_*^{1/2}(k)$ は探索履歴 $\chi(k)$ に依存し、 $\mathcal{C}(k) = \{\alpha\}$ である。既存のメタヒューリスティクスと同様に多点探索・確率的探索を継承しながら、固有ベクトルを活用した探索を実現する。なお、探索点と近傍解の優劣とは無関係に必ず探索点を更新する方法（絶対更新）を用いている。

Algorithm 6.2.1 に CMD-DS のアルゴリズムを示す。

中心ベクトル $\mathbf{u}^i(k)$ と乱数ベクトル $\mathbf{s}^i(k)$ の設定には下記の条件が課される。

条件 1: $\mathbf{u}^i(k)$ は、アフィン変換不变性を有する。

条件 2: $\mathbf{s}^i(k)$ が従う確率分布 \mathcal{P} は確率変数同士が独立であり、 \mathcal{P} のパラメータが探索過程で得られる解とは独立である。

Algorithm 17 CMD-DS

```

1: procedure CMD-DS
2:   Step 0 : パラメータ設定
3:   パラメータ  $m, k_{\max}, \alpha$  を設定する。
4:   Step 1 : 初期化
5:   探索点の位置  $\mathbf{x}^i(k)$  を初期配置領域  $S_0$  内にランダムに配置する。
6:    $k = 1$  とする。
7:   for  $i = 1, 2, \dots, m$  do
8:      $\mathbf{x}^i(k)$  を式 (6.1) を用いて更新す
9:   end for
10:  Step 4 : 終了判定
11:  if  $k \geq k_{\max}$  then
12:    計算を終了する。
13:  else
14:     $k := k + 1$  とし、Step 2 へ戻る。
15:  end if
Step 3 : 位置の更新
end procedure

```

この条件を満たすように $\mathbf{u}^i(k)$ と $\mathbf{s}^i(k)$ を設計すれば、そのアルゴリズム群はアフィン変換不变性の具備が保証される。この定理は 6.3 節で与える。また、更新則 U_f が目的関数値に関する解のランキングに基づくとき、目的関数の単調増加変換 T_f に対する不变性の具備が保証される。このため、中心ベクトル $\mathbf{u}^i(k)$ や乱数ベクトル $\mathbf{s}^i(k)$ の設計において、目的関数値に関する解のランキングを使用する。ただし、この性質は、多くのメタヒューリスティクスが具備するため、本章ではその詳細について議論せず、以降の証明では T_f は恒等写像として扱う。

6.2.2 CMD-DS の特徴

CMD-DS の特徴として三点が挙げられる。

一つ目は、変換不变性の具備を解析する必要がない点である。前節で述べた通り、既存のメタヒューリスティクスはアフィン変換不变性を具備しておらず、従来の変換不变性の付加アプローチは汎用な方法ではないという課題がある。一方、CMD-DS では、提示される条件を満たすようにアルゴリズムを設計すれば、変換不变性の具備が保証されるため、

個々のアルゴリズムについて変換不变性の有無を解析する必要がない。

二つ目は、最適化能力の設計自由度が高い点である。CMD-DS に従って設計されるアルゴリズム群は、既存の発見的手法と同様に、多点探索・確率的探索という特徴を有するが、生物の進化過程などの経験則に基づかない点で異なる。このため、アルゴリズムの詳細（線形結合の係数 α , 中心ベクトル $\mathbf{u}^i(k)$, 乱数ベクトル $\mathbf{s}^i(k)$ が従う確率分布 \mathcal{P} など）は定められておらず、最適化能力を有していない。しかしながら、提示された条件を満たす範囲で、探索戦略を適切に反映するように詳細構造を定めることで、最適化能力を有するアルゴリズムを構成できる。なお、本稿の CMD-DS の更新則（式 (6.1)）は絶対更新を用いているが、DE や ES のように近傍解の目的関数値が優れている場合にのみ探索点を更新する方法（改善更新）に変更することで、POP を反映することも可能である。

三つ目は、共分散行列の固有値分解に基づく基底変換を活用して、アフィン変換不变性を保障する点である。既存のメタヒューリスティクスがアフィン変換不变性を欠如する原因是、更新式内に変数の成分毎に異なる演算を含むことである。例えば、ES や実数値 SA, CS の更新則は、乱数ベクトルを単独の項として含むが、①乱数ベクトルの各確率変数同士が独立である、②乱数ベクトルが従う確率分布 \mathcal{P} のパラメータが設計変数とは独立である、などにより、線形変換不变性を失っている¹。一方、CMD-DS では、 $\Sigma_*^{1/2}$ を意図的に乱数ベクトル \mathbf{s}^i に適用することで、線形変換不变性を確実に補完する。CMD-DS の変換不变性定理については 6.3 節で示す。

4.2 節で整理したように、先行研究では、ES, 実数値 GA, DE, ABC に対して、共分散行列の固有値分解に基づく基底変換を活用して変換不变性を付加するアプローチがある [88, 93, 189, 105, 106, 109, 110, 193]。本論文の第 4 章で提案した CRI-PSO と CMD-CS も、これと同様のアプローチである。しかしながら、これらの先行研究は既存のメタヒューリスティクスに基づくため、最適化能力を決定づけるアルゴリズムの詳細構造において設計自由度が低い。一方、CMD-DS に基づくアプローチは高い設計自由度を活用して高度な方策を組み込むことで、変換不变性が保障されると同時に、最適化能力の向上が可能な点で本質的に異なる。

さらに、Koguma らは、安定性理論に基づき、アルゴリズムの正規乱数結合フレームワー-

¹ 確率分布 \mathcal{P} が全方向に一様（標準正規分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ など）である場合は、それに従う乱数ベクトル \mathbf{s}^i は回転不变性のみを有する。

クを提案しており、このフレームワークに従って設計したアルゴリズムは、パラメータ設定に対する安定性指標を厳密に導出可能である[157]。これは、理論的性質が保証されたアルゴリズムのフレームワークである点で本章で提案する CMD-DS と類似しているが、アフィン変換不变性に関する定理は導出されていない点で CMD-DS とは異なる。

6.3 CMD-DS の変換不变性定理

6.3.1 CMD-DS の変換不变性定理

CMD-DS のアフィン変換に関する不变性定理は定理 6.1 として与えられる。

定理 6.1 CMD-DS の更新則 $U_{f,S}$ が条件 1 と条件 2 を満たすとき、アフィン変換不变性を有する。

証明. アフィン変換 T_s によって、更新則 $U_{f,S}$ の引数がどのように変換されるのかを考える。アフィン変換 T_s に対して、 $\mathbf{u}^i(k)$ は $\mathbf{u}^{ii}(k) = \mathbf{C}\mathbf{u}^i(k) - \mathbf{t}$ であり（条件 1）、 $\mathbf{s}^i(k)$ は $\mathbf{s}^{ii}(k) = \mathbf{s}^i(k)$ である（条件 2）。一方、アフィン変換 T_s に対して $\Sigma_*^{1/2}(k)$ は $\Sigma_*^{1/2i}(k) = \mathbf{C}\Sigma_*^{1/2}(k)$ である（補助定理 A.3）。よって、

$$\begin{aligned} & U_{f,S'}(T_s(\mathbf{x}^i(k)); \chi_{S'}(k)) \\ &= \mathbf{u}^{ii}(k) + \alpha\Sigma_*^{1/2i}(k)\mathbf{s}^{ii}(k) \\ &= \mathbf{C}\mathbf{u}^i(k) - \mathbf{t} + \alpha\mathbf{C}\Sigma_*^{1/2}(k)\mathbf{s}^i(k) \\ &= \mathbf{C}(\mathbf{u}^i(k) + \alpha\Sigma_*^{1/2}(k)\mathbf{s}^i(k)) - \mathbf{t} \\ &= \mathbf{C}\mathbf{x}^i(k+1) - \mathbf{t} \\ &= (T_s \circ U_{f,S})(\mathbf{x}^i(k); \chi_S(k)) \end{aligned}$$

は式 (4.1) を満たす。以上から、題意が示された。 \square

補助定理 A.3 は付録 A を参照されたい。

中心ベクトル \mathbf{u}^i の設計方針（条件 1 の成立条件）を与えるために、 \mathbf{u}^i が代表的な形式である場合の変換不变性定理を記述する。 \mathbf{u}^i の代表的な形式は、 $\mathbf{u}^i = \mathbf{A}\mathbf{v}$ と $\mathbf{u}^i = \sum_j \mathbf{A}_j(\mathbf{x}_j - \mathbf{y}_j) + \mathbf{v}$ （基本ベクトルと差分ベクトルの線形結合）に大別される。ただし、

$\mathbf{A}, \mathbf{A}_j \in \mathbb{R}^{N \times N}$, $\mathbf{x}_j, \mathbf{y}_j, \mathbf{v} \in \mathbb{R}^N$ である。これは、3.11.2 項において更新則を演算形式に基づき分類したものを参考にしている。

アフィン変換に関する不变性定理が前者の形式の場合は定理 6.2 として与えられ、後者の形式の場合は定理 6.3 として与えられる。

定理 6.2 CMD-DS の更新則 $U_{f,S}$ において、 \mathbf{u}^i の形式が $\mathbf{u}^i = \mathbf{Av}$ であるとき、 \mathbf{u}^i がアフィン変換不变性を有するための条件は、 $\mathbf{A} = \mathbf{I}$ が成立することである。ただし、 $\mathbf{I} \in \mathbb{R}^{N \times N}$ は単位行列である。

証明. アフィン変換 T_s に対して、 $\mathbf{v}' = \mathbf{Cv} - \mathbf{t}$ となる。また、 $\mathbf{A}' = \mathbf{I} = \mathbf{A}$ である。よって、

$$\begin{aligned}\mathbf{u}'(k) &= \mathbf{A}'\mathbf{v}' \\ &= \mathbf{I}(\mathbf{Cv} - \mathbf{t}) \\ &= \mathbf{Cv} - \mathbf{t} \\ &= \mathbf{CIv} - \mathbf{t} \\ &= \mathbf{CAv} - \mathbf{t} \\ &= \mathbf{Cu}^i(k) - \mathbf{t}\end{aligned}$$

である。以上から、題意が示された。 \square

定理 6.3 CMD-DS の更新則 $U_{f,S}$ において、 \mathbf{u}^i の形式が $\mathbf{u}^i = \sum_j \mathbf{A}_j(\mathbf{x}_j - \mathbf{y}_j) + \mathbf{v}$ であるとき、 \mathbf{u}^i はアフィン変換不变性を有するための条件は、 $\mathbf{A}_j = \delta_j \mathbf{I}$ が成立することである。ただし、 $\delta_j > 0$ である。

証明. アフィン変換 T_s に対して $\mathbf{x}_j, \mathbf{y}_j, \mathbf{v}$ は $\mathbf{x}' = \mathbf{Cx} - \mathbf{t}$ となる。また、 $\mathbf{A}'_j = \delta_j \mathbf{I} = \mathbf{A}_j$

である。よって,

$$\begin{aligned}
 \mathbf{u}^i(k) &= \sum_j \mathbf{A}'_j(\mathbf{x}'_j - \mathbf{y}'_j) + \mathbf{v}' \\
 &= \sum_j (\delta_j \mathbf{I}) \left((\mathbf{C}\mathbf{x}_j - \mathbf{t}) - (\mathbf{C}\mathbf{y}_j - \mathbf{t}) \right) + \mathbf{C}\mathbf{v} - \mathbf{t} \\
 &= \sum_j (\delta_j \mathbf{I}) \mathbf{C}(\mathbf{x}_j - \mathbf{y}_j) + \mathbf{C}\mathbf{v} - \mathbf{t} \\
 &= \mathbf{C} \sum_j (\delta_j \mathbf{I})(\mathbf{x}_j - \mathbf{y}_j) + \mathbf{C}\mathbf{v} - \mathbf{t} \\
 &= \mathbf{C} \left(\sum_j \mathbf{A}_j(\mathbf{x}_j - \mathbf{y}_j) + \mathbf{v} \right) - \mathbf{t} \\
 &= \mathbf{C}\mathbf{u}^i(k) - \mathbf{t}
 \end{aligned}$$

である。以上から、題意が示された。 \square

上記の \mathbf{u}^i の形式は、メタヒューリスティクスに限らず、定義 4.1 に従う反復法のアルゴリズムに立脚しているため、定理 6.2 と定理 6.3 を適用することで、多くの最適化アルゴリズムのアフィン変換不变性の有無を指摘することが可能である。また、条件 2 を満たす確率分布 \mathcal{P} の例として、標準正規分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 、多変量実数値一様分布 $\mathcal{U}_{\mathbb{R}}(\mathbf{a}, \mathbf{b})$ などが挙げられ、多様な確率分布を採用できる。

6.3.2 CMD-DS の変換不变性の数値実験検証

(a) 簡潔なサンプルアルゴリズム

本節では、CMD-DS に従ってサンプルアルゴリズムを設計し、数値実験を通じて、そのアルゴリズムのアフィン変換不变性を検証する。CMD-DS に基づく具体的なアルゴリズムとして、簡潔なアルゴリズム $\mathcal{A}_{\text{test}}$ を構築する。 $\mathcal{A}_{\text{test}}$ の更新則 $U_{f,S}$ は、CMD-DS の更新則（式 (6.1)）における $\mathbf{u}^i, \mathbf{s}^i$ を式 (6.3), 式 (6.4) と設定したものである。

$$\mathbf{u}^i(k) = \mathbf{x}^i(k) \quad (6.3)$$

$$\mathbf{s}^i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (6.4)$$

ただし, $\mathcal{Z} = \mathcal{X}(k) = \{\mathbf{x}^i(k) \mid i = 1, \dots, m\}$, $\mathcal{N}(\mathbf{0}, \mathbf{I})$ は N 次元標準正規分布である。

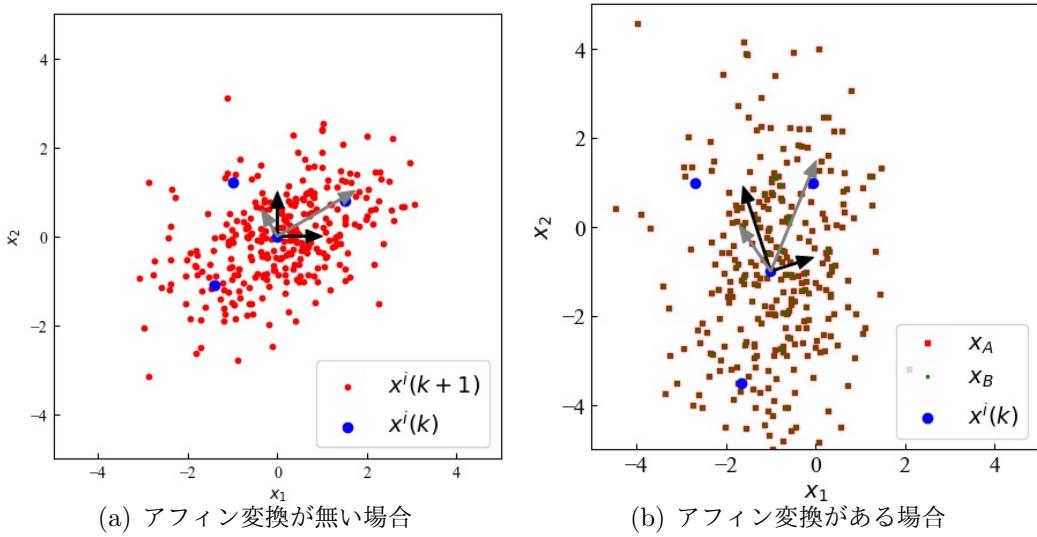
変換不变性定理の条件式の成立について議論する。 \mathbf{u}^i は定理 6.2 の形式に該当する。このとき, \mathbf{u}^i がアフィン変換不变性を具備するには, $\mathbf{A} = \mathbf{I}$ (係数が単位行列) が成立すればよい (定理 6.2)。この更新則の \mathbf{x}^i の係数は単位行列であるため, \mathbf{u}^i はアフィン変換不变性を有する (条件 1 を満たす)。また, 標準正規分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ は, 確率変数間が独立しており, パラメータが設計変数とは独立している (条件 2 を満たす)。したがって, $\mathcal{A}_{\text{test}}$ の更新則は, 条件 1 と条件 2 を満たすことから, 解空間のアフィン変換不变性を有する (定理 6.1)。

なお, サンプルアルゴリズム $\mathcal{A}_{\text{test}}$ は最も簡潔な実装例であり, POP の思想が導入されていないため, 最適化能力を有さない点に注意されたい。

(b) 変換不变性の実験的検証

数値実験を通じて, $\mathcal{A}_{\text{test}}$ のアフィン変換不变性を検証する。CMD-DS のサンプルアルゴリズム $\mathcal{A}_{\text{test}}$ と \mathcal{A}_{PSO} の数値実験を通じて, CMD-DS の有効性を検証する。ところで, アフィン変換は線形変換 $\mathbf{x} \mapsto \mathbf{Cx}$ と平行移動 $\mathbf{x} \mapsto \mathbf{x} - \mathbf{t}$ に分解できる。本実験で用いる線形変換は, 大域的最適化で困難とされる性質を再現する変換 (スケール変換, 回転変換) に限定する。つまり, アフィン変換を与える行列とベクトルは $\mathbf{C} = \mathbf{B}(\theta)\mathbf{D}^\varepsilon, \mathbf{t} = \delta\mathbf{1}$ とする。ただし, \mathbf{t} は平行移動を与えるベクトル $\mathbf{t} = \mathbf{1} = (1, 1, \dots, 1)^T$, \mathbf{D} はスケール変換を与える行列 $\mathbf{D} = \text{diag}(1, 2, \dots, N)$, $\mathbf{B}(\theta)$ は解空間を θ deg 回転させる回転行列である。本実験では, 2 次元平面の解空間にアフィン変換 T_s を与えた場合の $\mathbf{x}_A = \mathbf{x}_B$ (式 (4.1)) の成立を確認する。ただし, \mathbf{x}_A はアフィン変換後に生成した近傍解であり, \mathbf{x}_B は近傍解をアフィン変換した解を意味する。アフィン変換のパラメータは $\theta = 18, \delta = \varepsilon = 1$, $\mathcal{A}_{\text{test}}$ のパラメータは $\alpha = 1$ とする。

図 6.1 に, 同じ探索点から, $\mathcal{A}_{\text{test}}$ の更新則により近傍生成を 500 回行った図を示す。原点の \circlearrowleft は探索点 \mathbf{x}^i , それ以外の \circlearrowleft は他の探索点 $\mathbf{x}^j (j \neq i)$, 黒い矢印は標準基底ベクトル, 灰色の矢印は固有ベクトルを表す。図 6.1(a) は近傍解 $\hat{\mathbf{x}}^i$, 図 6.1(b) は \mathbf{x}_A と \mathbf{x}_B を示している。図 6.1 から, $\mathbf{x}_A = \mathbf{x}_B$ の成立が確認できる。また, $\mathcal{A}_{\text{test}}$ の近傍は, 解空間のアフィン変換に関わらず, 固有ベクトルに沿って生成されることも確認できる。したがって, 数値実験を通じて, CMD-DS のサンプルアルゴリズム $\mathcal{A}_{\text{test}}$ はアフィン変換不变性を具備

図 6.1: $\mathcal{A}_{\text{test}}$ の近傍生成

すること、あるいは不変性定理の成立例を示した。

6.4 CMD-DS に基づく PSO 着想アルゴリズム

4.3 節で確認したように、PSO は発見的手法であるため、変換不変性の観点から構築されておらず、平行移動不変性やスケール不変性を有するが、回転不変性を欠如していることから、より一般化したアフィン変換不変性を欠如している。そこで、CMD-DS に基づく具体的なサンプルアルゴリズムとして、PSO に着想を得たアルゴリズム \mathcal{A}_{PSO} を構築する。 \mathcal{A}_{PSO} の更新則 U_f は、CMD-DS の更新則（式 (6.1)）における $\mathbf{u}^i, \mathbf{s}^i$ を式 (6.5), 式 (6.6) と設定したものである。

$$\mathbf{u}^i(k) = \mathbf{x}^i(k) + c_1 r_1 (\mathbf{p}^i(k) - \mathbf{x}^i(k)) + c_2 r_2 (\mathbf{p}^g(k) - \mathbf{x}^i(k)) \quad (6.5)$$

$$\mathbf{s}^i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (6.6)$$

ただし、 $\alpha = \beta(1 - c_1 r_1 - c_2 r_2)$, $\mathcal{Z}(k) = \mathcal{P}(k)$, $r_1, r_2 \sim \mathcal{U}(0, 1)$, $\mathbf{p}^i(k)$ は p-best, $\mathbf{p}^g(k)$ は g-best である。このとき、 $\alpha \in [\beta(1 - c_1 - c_2), \beta]$, $\mathcal{C}(k) = \{c_1, c_2, \beta\}$ である。改めて、

式(6.1)に式(6.5), 式(6.6)を代入して, 更新則を書き直すと, 式(6.7)と表せる。

$$\begin{aligned}\mathbf{x}^i(k+1) &= \mathbf{x}^i(k) + c_1 r_1 (\mathbf{p}^i(k) - \mathbf{x}^i(k)) + c_2 r_2 (\mathbf{p}^g(k) - \mathbf{x}^i(k)) \\ &\quad + \beta(1 - c_1 r_1 - c_2 r_2) \boldsymbol{\Sigma}_*^{1/2}(k) \mathbf{s}^i(k)\end{aligned}\quad (6.7)$$

ここで, Linear PSO [90]の更新則を変形すると, \mathcal{A}_{PSO} と類似の式が導出できる。式(6.7)は, 式(6.8), 式(6.9)を導入すると, 式(6.10)へ変形できる。

$$\mathbf{y}^i(k) = \mathbf{x}^i(k) + \beta \boldsymbol{\Sigma}_*^{1/2}(k) \mathbf{s}^i(k) \quad (6.8)$$

$$\mathbf{z}^i(k) = \frac{c_1 r_1 \mathbf{p}^i(k) + c_2 r_2 \mathbf{p}^g(k)}{\phi} \quad (6.9)$$

$$\begin{aligned}\mathbf{x}^i(k+1) &= \mathbf{y}^i(k) + c_1 r_1 (\mathbf{p}^i(k) - \mathbf{y}^i(k)) + c_2 r_2 (\mathbf{p}^g(k) - \mathbf{y}^i(k)) \\ &= (1 - \phi) \mathbf{y}^i(k) + \phi \mathbf{z}^i(k)\end{aligned}\quad (6.10)$$

ただし, $\phi = c_1 r_1 + c_2 r_2$ である。Linear PSO の更新則は式(6.11)と表される。

$$\mathbf{x}^i(k+1) = \mathbf{x}^i(k) + w \mathbf{v}^i(k) + c_1 r_1 (\mathbf{p}^i(k) - \mathbf{x}^i(k)) + c_2 r_2 (\mathbf{p}^g(k) - \mathbf{x}^i(k)) \quad (6.11)$$

これは, PSO の更新則において, 係数の乱数行列をスカラとしただけである。式(6.11)を変形すると, 式(6.12)と表せる。

$$\mathbf{x}^i(k+1) = (1 - \phi) \mathbf{x}^i(k) + \phi \mathbf{z}^i(k) + w \mathbf{v}^i(k) \quad (6.12)$$

Linear PSO はパラメータ設定が安定領域に位置する場合, 慣性項は $\mathbf{v}^i(k) \rightarrow_{k \rightarrow \infty} \mathbf{0}$ と収束するため, 探索点は $\mathbf{x}^i(k)$ と $\mathbf{z}^i(k)$ の内分比 $\phi : (1 - \phi)$ の位置が平衡点となり, p-best と g-best の内分点に収束する。このように, 有望な領域である p-best と g-best の位置に探索点が収束していく点が PSO の最適化能力といえる。しかしながら, Linear PSO は乱数を要素毎に統一しているため, 通常の PSO に比べて多様性が低く, 性能が悪化する場合がある。一方, \mathcal{A}_{PSO} において, $\mathbf{y}^i(k) \sim \mathcal{N}(\mathbf{x}^i(k), \beta^2 \boldsymbol{\Sigma})$ で, 探索点は $\mathbf{y}^i(k)$ と $\mathbf{z}^i(k)$ の内分比 $\phi : (1 - \phi)$ の位置が平衡点となり, $\mathbf{y}^i(k)$ が $\mathbf{y}^i(k) \rightarrow_{k \rightarrow \infty} \mathbf{x}^i(k)$ と収束する場合, p-best と g-best の内分点に収束する。また, 乱数ベクトルは要素とは無関係に運動を与えるため, 通常の PSO が作る乱数行列による運動と同様の効果が得られると考えられる。したがって, \mathcal{A}_{PSO} は PSO の最適化能力に着想を得ており, 乱数の多様性を継承しているといえる。

変換不変性定理の条件式の成立について議論する。 \mathbf{u}^i は定理 6.3 の形式に該当する。このとき、 \mathbf{u}^i がアフィン変換不変性を具備するには、 $\mathbf{A}_j = \delta_j \mathbf{I}$ （差分ベクトルの係数がスカラ）が成立すればよい（定理 6.3）。この更新則の差分ベクトルの係数 $c_1 r_1, c_2 r_2$ はスカラであるため、 \mathbf{u}^i はアフィン変換不変性を有する（条件 1 を満たす）。また、確率分布 \mathcal{P} は標準正規分布 $\mathcal{N}(0, \mathbf{I})$ であるため、 \mathcal{A}_{PSO} と同様に、条件 2 を満たす。したがって、 \mathcal{A}_{PSO} の更新則は、条件 1 と条件 2 を満たすことから、解空間のアフィン変換不変性を有する（定理 6.1）。

サンプルアルゴリズム \mathcal{A}_{PSO} は、探索点が p-best と g-best の存在する領域に向かうという PSO における POP の思想が導入されているため、PSO と同等の最適化能力を有すると同時に、オリジナル PSO の多様性を継承すると期待される。さらに、CMD-DS に従って構築されているため、解空間のアフィン変換に対する探索性能のロバスト性が高いことが期待される。

6.5 数値実験検証

6.5.1 実験条件

本節では、数値実験を通じて、CMD-DS のサンプルアルゴリズム \mathcal{A}_{PSO} の性能を検証する。 \mathcal{A}_{PSO} と比較する手法は PSO と Linear PSO [90] とする。共通条件として、探索点数 $m = 20$ 、PSO のパラメータ $c_1 = c_2 = 1.4955$ を用いる。 \mathcal{A}_{PSO} のパラメータは $\beta = 2$ 、PSO と Linear PSO の慣性パラメータは $w = 0.729$ (Constriction Method : CM) を用いる。初期配置領域 S_0 内に初期探索点をランダムに配置した後、終了条件を満たすまで探索を進める試行を 50 回行い、50 試行の g-best の目的関数値 $f(\mathbf{p}^g(k_{\max}))$ の平均値を探索性能とする。ベンチマーク関数は、付録 C を参照されたい。

6.5.2 探索性能のロバスト性に関する検証

本項の数値実験では、 \mathcal{A}_{PSO} のアフィン変換に対する探索性能のロバスト性を検証する。本実験では、解空間にアフィン変換の一種である平行移動変換、スケール変換、回転変

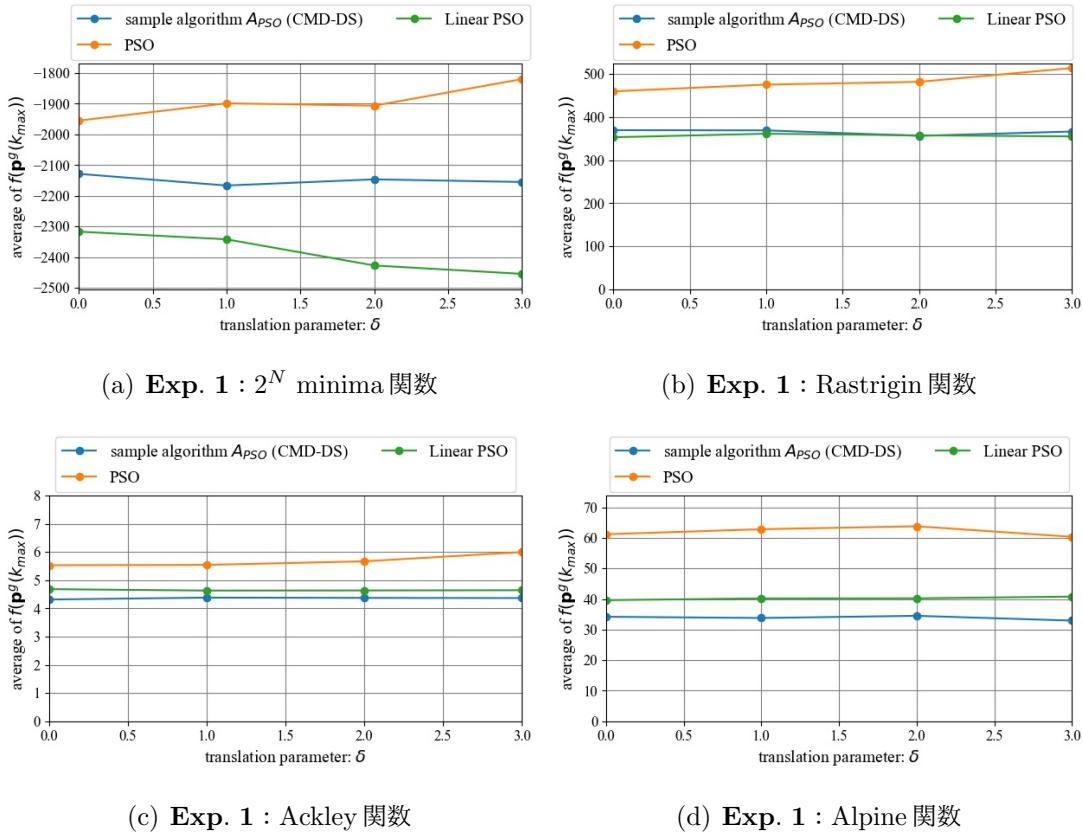


図 6.2： 数値実験結果 (Exp. 1)： A_{PSO} と PSO の探索性能のロバスト性 ($N = 50$ 、 $k_{\max} = 100$)

換を与えていき、その探索性能の変動を確認する。ベンチマーク関数として、変数分離可能・良スケール性・原点に最適解が存在・多峰性の4種類の目的関数を使用する。実験方法は、4.4.4項と同様である。共通条件として、次元数 $N = 50$ 、終了条件として反復回数 $k_{\max} = 100$ を用いる。

図 6.2、図 6.3、図 6.4 に数値実験結果を示す。各図の横軸は、平行移動パラメータ δ 、スケール変換パラメータ ε 、回転角 θ であり、全ての図の縦軸が探索性能 ($f(\mathbf{p}^g(k_{\max}))$) の平均値) である。図 6.2、図 6.3、図 6.4 より、下記の結果が確認できる。

- PSO と A_{PSO} の探索性能は、平行移動とスケール変換に対して変動が少ない。
- PSO の探索性能は回転変換に対して大きく変動する一方、 A_{PSO} は変動が少ない。
- A_{PSO} の探索性能は、ほぼ全ての条件において PSO よりも優れている。

これは、PSO は平行移動不変性とスケール不変性を具備するが、回転不変性を欠如している。

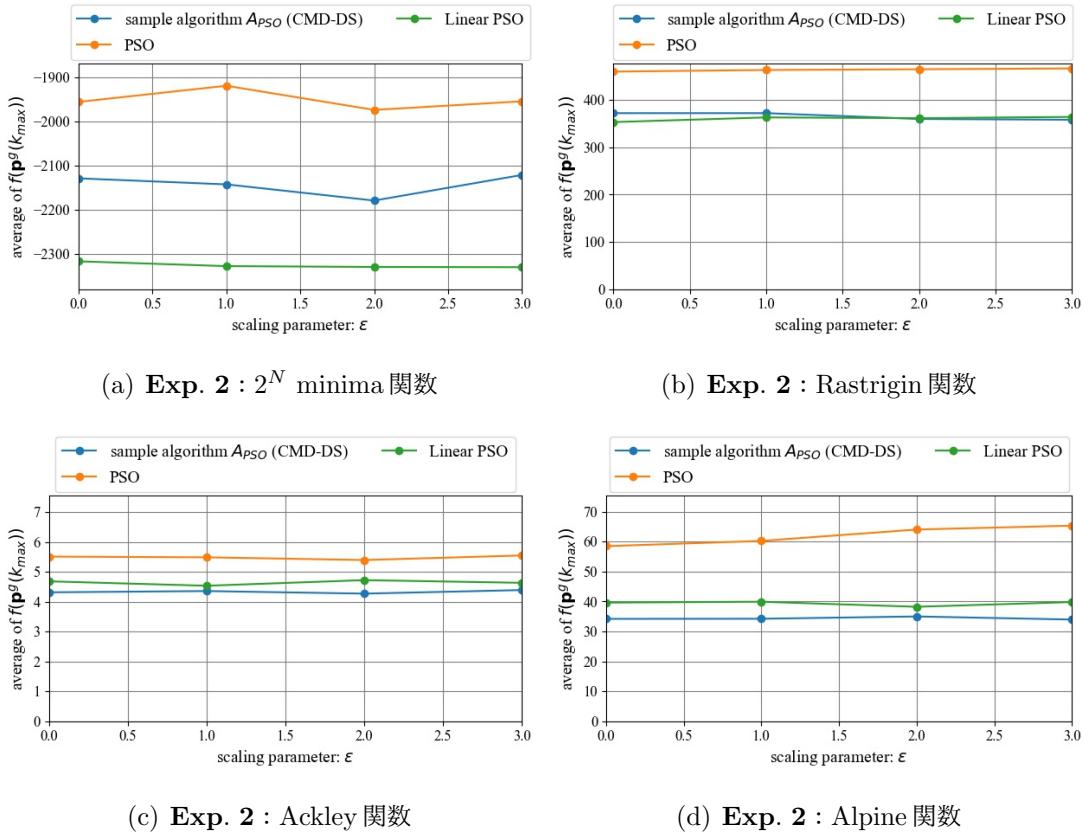


図 6.3 : 数値実験結果 (Exp. 2) : A_{PSO} と PSO の探索性能のロバスト性 ($N = 50$, $k_{\max} = 100$)

ることに対して, A_{PSO} は平行移動, スケール変換, 回転変換を含むアフィン変換に対する不变性を具備するためだと考えられる。この結果は, 不变性の有無が探索性能のロバスト性に影響を与えることを示している。よって, CMD-DS のサンプルアルゴリズム A_{PSO} は, 代表的な手法である PSO と同等の最適化能力を具備すると同時に, 解空間のアフィン変換に対する探索性能の高いロバスト性を示すことが確認された。

6.5.3 全般的な性能に関する検証

数値実験を通じて, CMD-DS の全般的な探索性能を検証する。数値実験には, 9種類のベンチマーク関数を用いる。共通条件として, 次元数 $N = 10, 50, 100, 300$, 終了条件を反復回数 $k_{\max} = 100, 1000$ とする。

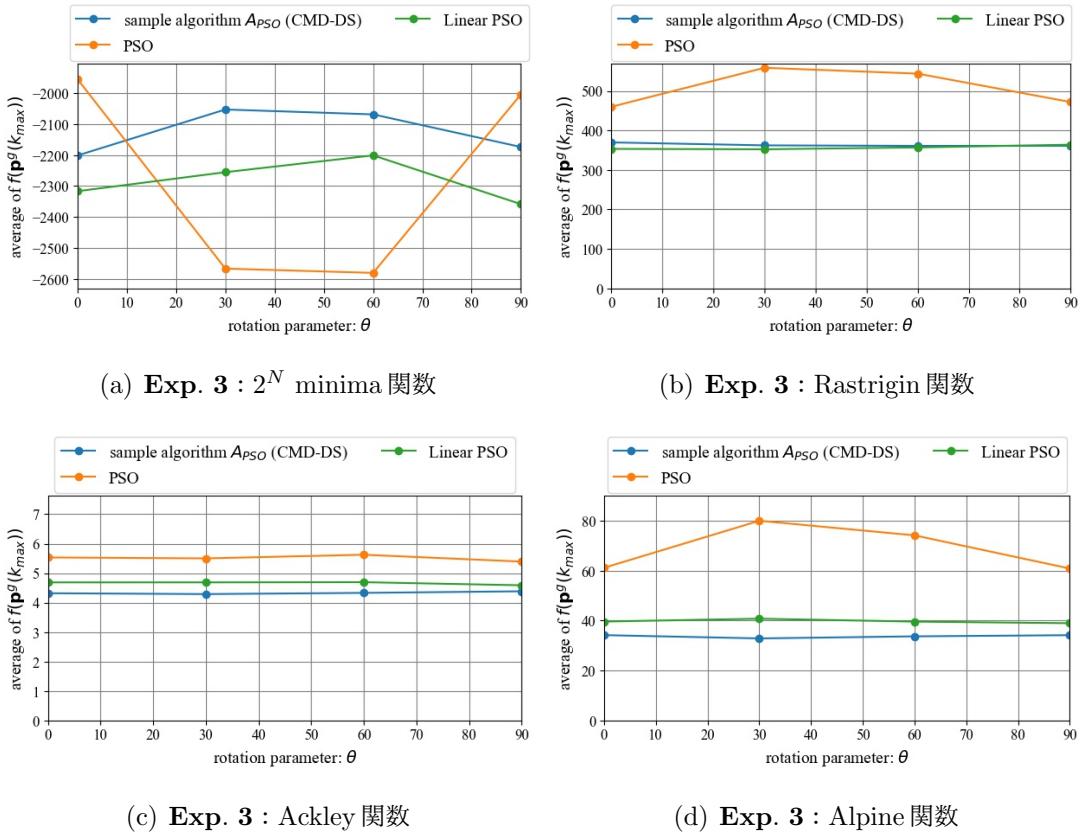


図 6.4 : 数値実験結果 (Exp. 3) : A_{PSO} と PSO の探索性能のロバスト性 ($N = 50$ 、
 $k_{\max} = 100$)

数値実験結果を表 6.1 に示す。表 6.1 では、3 つの方法 (PSO, Linear PSO, A_{PSO}) の結果を比較して、最も優れた結果に (*) を付し、二番目に優れた結果に (**) を付している。表 6.1 から下記の結果が確認できる。

- (1) 表中の * と ** の合計の数は、PSO が 32 個、Linear PSO が 46 個、 A_{PSO} が 66 個であることから、全般的な探索性能は $A_{PSO} > \text{Linear PSO} > \text{PSO}$ という優劣関係である。
- (2) 表中の * と ** の数は、 $k_{\max} = 100$ のとき、PSO が 9 個、Linear PSO が 31 個、 A_{PSO} が 32 個、 $k_{\max} = 1000$ のとき、PSO が 23 個、Linear PSO が 15 個、 A_{PSO} が 34 個であることから、短期では、 $A_{PSO} > \text{Linear PSO} > \text{PSO}$ という優劣関係で、長期では、 $A_{PSO} > \text{PSO} > \text{Linear PSO}$ という優劣関係である。

(3) 表中の * と ** の数は, $N = 10$ のとき, PSO が 18 個, Linear PSO が 9 個, \mathcal{A}_{PSO} が 9 個, $N = 50$ のとき, PSO が 9 個, Linear PSO が 11 個, \mathcal{A}_{PSO} が 16 個, $N = 100$ のとき, PSO が 5 個, Linear PSO が 13 個, \mathcal{A}_{PSO} が 18 個, $N = 300$ のとき, PSO が 0 個, Linear PSO が 13 個, \mathcal{A}_{PSO} が 23 個であることから, 低次元では PSO > $\mathcal{A}_{\text{PSO}} = \text{Linear PSO}$ という優劣関係で, 高次元では $\mathcal{A}_{\text{PSO}} > \text{Linear PSO} > \text{PSO}$ という優劣関係である。

(1) の結果は, \mathcal{A}_{PSO} が PSO の多様性を維持しながら, 最適化能力を基本的に引き継ぐと同時に, アフィン変換不変性を具備することに起因していると考えられる。

(2) の理由は, 3.4 節で述べたように, CM に基づく PSO のパラメータ設定は, パラメータ空間において弱い安定領域に位置しており, 持続的な多様化・集中化の探索戦略を実現できる一方, \mathcal{A}_{PSO} は PSO に着想を得ているが, 乱数ベクトルなどの違いにより, 安定・不安定領域とパラメータ設定の関係が PSO とは多少異なり ($\beta = 2.0$ は CM とは異なり), 短期に特化とした(集中化に特化)設定であったためだと考えられる。

次に (3) の考察を行う。PSO のパラメータ空間における安定・不安定領域の境界線は, 高次元になると安定領域側に多少小さくなり, CM は弱い安定領域から不安定領域に近づくため, 高次元においては高い性能を発揮できないことが指摘されている。これは, 上記の安定・不安定領域の導出が Linear PSO の更新則(差分ベクトルの係数の乱数がスカラ)で確認されているものだが, 通常の PSO では乱数行列であり, 高次元になると全ての要素において安定な探索ダイナミクスとなる状態が実現しにくくなり, 集中化の能力が弱くなるためだと言われている。一方, \mathcal{A}_{PSO} は(詳細には) Linear PSO の更新則を参照しているため, \mathcal{A}_{PSO} と Linear PSO では, 上記の現象が起きず, 高次元においても多様化・集中化の探索戦略が実現できたと考えられる。これらの理由から, 低次元では PSO の弱い安定状態が上手く探索に寄与したために, \mathcal{A}_{PSO} と Linear PSO よりも高い性能を示したが, 大差がない一方, 高次元では \mathcal{A}_{PSO} と Linear PSO は PSO よりも大差をつけて優れている。加えて, \mathcal{A}_{PSO} と同様に, Linear PSO はアフィン変換不変性を有するが, 差分ベクトルの係数乱数がスカラであるため, 近傍の多様性を失うことに対して, \mathcal{A}_{PSO} は乱数ベクトルに基づく摂動により, 近傍の多様性を維持している。以上の理由から, 全般的な性能は(特に高次元においては), 平均的に \mathcal{A}_{PSO} が優れていると考えられる。

以上から, \mathcal{A}_{PSO} は近傍の多様性を失わずにいながら, PSO の基本的な探索性能を引き

継ぐと同時に、アフィン変換不変性を具備しているといえる。さらに、本実験を通じて、CMD-DSは優秀な既存のアルゴリズムの最適化性能を埋め込むことで、アフィン変換不変性の具備が保証されながら、優秀な最適化能力を発揮するポテンシャルを有することを示した。

表6.1：全般的な探索性能に関する数値実験結果（CMD-DS のサンプルアルゴリズム \mathcal{A}_{PSO} ）

関数	N	$k_{\max} = 100$			$k_{\max} = 1000$		
		PSO	Linear PSO	\mathcal{A}_{PSO}	PSO	Linear PSO	\mathcal{A}_{PSO}
1. Sphere	10	0.00 *	0.27 **	1.09	0.00 *	0.23 **	0.93
	50	43.2	30.2 *	30.5 **	0.01 *	32.1	29.1 **
	100	443	74.3 **	63.5 *	11.6 *	72.8	64.2 **
	300	2225	258 **	202 *	2221	254 **	210 *
2. Rosenbrock	10	8.60 *	11.4 **	16.3	1.31 *	12.3 **	17.5
	50	2236	602 **	577 *	99.6 *	590	538 **
	100	29456	1528 **	1286 *	1831	1521 **	1366 *
	300	111619	5426 **	4457 *	111776	5666 **	4185 *
3. 2^N minima	10	-723 *	-662	-635 **	-727 *	-663 **	-639
	50	-1955	-2317 *	-2178 **	-3344 *	-2356 **	-2187
	100	-2285	-4101 *	-3821 **	-2480	-4125 **	-3892 *
	300	-4931	-11013 *	-10665 **	-4898	-11091 *	-10784 **
4. Rastrigin	10	21.3 *	26.3 **	32.6	9.13 *	23.1	21.7 **
	50	460	353 *	368 **	186 *	354	350 **
	100	1364	870 **	831 *	592 *	841	823 **
	300	5029	2964 **	2858 *	5015	2956 **	2813 *
5. Schwefel	10	0.35 *	0.75	0.62 **	0.00 *	0.60 **	0.72
	50	297	60.3 **	56.4 *	34.1 *	60.6 **	63.2
	100	1469	248 *	285 **	363	236 *	248 **
	300	15518	2090 *	2476 **	4968	2141 *	2455 **
6. Levy	10	0.03 *	3.86	2.72 **	0.04 *	3.48	2.69 **
	50	40.5	19.1 **	16.3 *	3.83 *	18.5	15.2 **
	100	87.5	24.6 **	21.5 *	16.9 *	25.4	20.8 **
	300	143	32.1 **	27.2 *	136	30.5 **	26.9 *
7. Ackley	10	0.17 *	2.57	2.16 **	0.15 *	2.31	2.19 **
	50	5.53	4.69 **	4.32 *	3.13 *	4.58	4.28 **
	100	8.70	4.89 **	4.51 *	5.10	4.90 **	4.57 *
	300	10.1	5.03 **	4.69 *	10.08	5.05 **	4.70 *
8. Griewank	10	0.25 *	0.28 **	0.44	0.09 *	0.24 **	0.37
	50	2.16	1.80 **	1.73 *	0.10 *	1.78	1.71 **
	100	12.3	2.87 **	2.58 *	1.33 *	2.83	2.60 **
	300	56.6	7.31 **	6.30 *	56.6	7.46 **	6.16 *
9. Alpine	10	0.55 *	2.43	1.66 **	0.00 *	2.21	1.00 **
	50	61.2	39.6 **	33.8 *	2.91 *	37.9	32.2 **
	100	191	98.3 **	77.3 *	32.6 *	93.0	76.9 **
	300	794	316 **	265 *	618	300 **	259 *

6.6 アルゴリズムの探索性能に関する総合比較

BBO の環境下における最適化アルゴリズムのロバスト性と適応性の向上を目指し、第 4 章と第 5 章において、代表的なメタヒューリスティクスである PSO, ABC, CS に対して、変換不変性とパラメータ調整則を付加したアルゴリズムを設計した。さらに、第 6 章においては、変換不変性が保証されるアルゴリズムのフレームワークを新たに提案し、そのフレームワークに基づき、PSO に着想を得たサンプルアルゴリズムを設計した。また、それぞれ提案したアルゴリズムは、オリジナルのアルゴリズムよりも全般的な性能が平均的に優れていることを確認した。一方、ロバスト性・適応性の高さを評価するためには、アルゴリズム同士の総合的な比較が必要だと考えられる。

以上を踏まえ、本節では、本論文で提案したメタヒューリスティクスを総合的に比較する。なお、オリジナルのメタヒューリスティクスの同士の比較は非常に困難である。なぜなら、メタヒューリスティクスは、目的関数のクラス・性質や探索条件を絞れば、パラメータを相応しい値に設定でき、優れた探索性能を発揮するためである。しかしながら、本論文では、BBO のように少なくとも目的関数のクラス・性質が不明であり、次元数や解空間のスケールなどが問題に依存するため、探索条件が事前に絞れない状況を想定しているため、パラメータを最適な値に設定してはいけないが、様々な探索条件において、平均的に優れていた値を採用した場合をオリジナルの代表的な性能とする。

まず、本論文で提案したメタヒューリスティクスを含め、各種メタヒューリスティクスの変換不変性の有無について、整理する。各種メタヒューリスティクスの変換不変性の有無を表 6.2 に示す。

次に、全般的な探索性能を総合的に比較する。比較するアルゴリズムは、第 4 章、第 5 章、第 6 章で検証した、PSO, CRI-PSO, Activity Feedback PSO (AFPSO) [116], 適応型 CRI-PSO, CMD-DS のサンプルアルゴリズム \mathcal{A}_{PSO} , Linear PSO [90], ABC, HBC-ABC, CS, CMD-CS, 適応型 CS の 11 種類に加えて、アフィン変換不変性と適応的パラメータ調整則を具備する (μ_w, λ) -CMA-ES の合計 12 種類とする。ただし、パラメータは、各検証で用いた平均的に優れていた値を使用し、CRI-PSO, 適応型 CRI-PSO, \mathcal{A}_{PSO} では $\mathcal{Z} = \mathcal{P}(k)$ とし、 (μ_w, λ) -CMA-ES は 5.2 節の推奨値を使用する。共通条件として、探索点数 $m = 20$, 次元数 $N = 10, 50, 100, 300$, 終了条件は、評価回数 $T_{\max} = 2000$, $T_{\max} = 20000$ とし

表 6.2 : 本論文で提案したアルゴリズムを含めた各種メタヒューリスティクスの変換不変性 (* は本論文で提案したアルゴリズム)

手法	T_s : アフィン変換不変性				T_f : 目的関数の 単調増加変換不変性	
	線型変換不変性					
	平行移動不変性	回転不変性	スケール不変性	相似変換不変性		
ES	○	○	×	×	○	
PSO	○	×	○	○	○	
DE	○	×	○	○	○	
ABC	○	×	○	○	×	
FA	○	○	×	×	○	
CS	○	×	×	×	○	
CRI-PSO*	○	×	○	○	○	
HBC-ABC*	○	×	○	○	×	
CMD-CS*	○	○	○	○	○	
適応型 CS*	○	×	△	△	○	
AFPSO	○	×	×	○	○	
適応型 CRI-PSO*	○	○	×	○	○	
Linear PSO	○	○	○	○	○	
CMD-DS*	○	○	○	○	○	
CMD-ES	○	○	○	○	○	

て統一した。初期配置領域 S_0 内に初期探索点をランダムに配置した後、終了条件を満たすまで探索を進める試行を 50 回行い、50 試行の g-best の目的関数値 $f(\mathbf{p}^g(k_{\max}))$ の平均値を実験結果とする。他に、ベンチマーク関数などの条件は、各検証で実施した「全般的な探索性能に関する検証」で用いた条件を用いる。

これらのアルゴリズムの性能を総合的に比較した表を表 6.3 に示す。表 6.3 では、11 種類のアルゴリズムの結果を比較して、最も良い結果に (●) を付しており、2 番目に良い結果に (*) を付しており、3 番目に良い結果に (■) を付しており、4 番目に良い結果に (◆) を付することで、上位 4 位の結果をマークする²。表 6.3 中の ●, *, ■, ◆ の合計の数は下記の

² なお、表 6.3 の一部では、同じ数値にも関わらず優劣がついている結果があるが、これは便宜上、小数点

通りである。

全条件 PSO が 11 個, CRI-PSO が 13 個, AFPSO が 29 個, 適応型 CRI-PSO が 45 個, \mathcal{A}_{PSO} が 20 個, Linear PSO が 14 個, ABC が 33 個, HBC-ABC が 41 個, CS が 16 個, CMD-CS が 17 個, 適応型 CS が 20 個, CMA-ES が 29 個

$T_{\max} = 2000$ PSO が 4 個, CRI-PSO が 13 個, AFPSO が 17 個, 適応型 CRI-PSO が 23 個, \mathcal{A}_{PSO} が 20 個, Linear PSO が 14 個, ABC が 7 個, HBC-ABC が 21 個, CS が 3 個, CMD-CS が 14 個, 適応型 CS が 0 個, CMA-ES が 8 個

$T_{\max} = 20000$ PSO が 7 個, CRI-PSO が 0 個, AFPSO が 12 個, 適応型 CRI-PSO が 22 個, \mathcal{A}_{PSO} が 0 個, Linear PSO が 0 個, ABC が 26 個, HBC-ABC が 20 個, CS が 13 個, CMD-CS が 3 個, 適応型 CS が 20 個, CMA-ES が 21 個

$N = 10$ PSO が 9 個, CRI-PSO が 4 個, AFPSO が 13 個, 適応型 CRI-PSO が 5 個, \mathcal{A}_{PSO} が 0 個, Linear PSO が 0 個, ABC が 13 個, HBC-ABC が 6 個, CS が 1 個, CMD-CS が 1 個, 適応型 CS が 4 個, CMA-ES が 16 個

$N = 50$ PSO が 1 個, CRI-PSO が 4 個, AFPSO が 10 個, 適応型 CRI-PSO が 15 個, \mathcal{A}_{PSO} が 3 個, Linear PSO が 2 個, ABC が 9 個, HBC-ABC が 10 個, CS が 4 個, CMD-CS が 3 個, 適応型 CS が 3 個, CMA-ES が 8 個

$N = 100$ PSO が 1 個, CRI-PSO が 3 個, AFPSO が 5 個, 適応型 CRI-PSO が 13 個, \mathcal{A}_{PSO} が 8 個, Linear PSO が 4 個, ABC が 7 個, HBC-ABC が 11 個, CS が 3 個, CMD-CS が 6 個, 適応型 CS が 6 個, CMA-ES が 5 個

$N = 300$ PSO が 0 個, CRI-PSO が 2 個, AFPSO が 1 個, 適応型 CRI-PSO が 12 個, \mathcal{A}_{PSO} が 9 個, Linear PSO が 8 個, ABC が 4 個, HBC-ABC が 14 個, CS が 8 個, CMD-CS が 7 個, 適応型 CS が 7 個, CMA-ES が 0 個

以上の結果を踏まえると, 下記のように探索性能の優劣関係を付けられる。

全般 適応型 CRI-PSO > HBC-ABC > ABC > AFPSO = CMA-ES > 適応型 CS = \mathcal{A}_{PSO} > CMD-CS > CS > Linear PSO > CRI-PSO > PSO

短期 適応型 CRI-PSO > HBC-ABC > \mathcal{A}_{PSO} > AFPSO > Linear PSO > CRI-PSO = CMD-CS > CMA-ES > ABC > PSO > CS > 適応型 CS

第三位以下を省略しているためであり, 小数点以下の差異を確認した上で優劣を判断している。

長期 ABC > 適応型 CRI-PSO > CMA-ES > 適応型 CS = HBC-ABC > CS > AFPSO
 > PSO > CMD-CS > \mathcal{A}_{PSO} = Linear PSO = CRI-PSO

$N = 10$ CMA-ES > ABC = AFPSO > PSO > HBC-ABC > 適応型 CRI-PSO > 適応型 CS > CRI-PSO > CS > CMD-CS > \mathcal{A}_{PSO} = Linear PSO

$N = 50$ 適応型 CRI-PSO > AFPSO = HBC-ABC > ABC > CMA-ES > CS = CRI-PSO > 適応型 CS = CMD-CS = \mathcal{A}_{PSO} > Linear PSO > PSO

$N = 100$ 適応型 CRI-PSO > HBC-ABC > \mathcal{A}_{PSO} > ABC > 適応型 CS = CMD-CS > AFPSO = CMA-ES > Linear PSO > CRI-PSO = CS > PSO

$N = 300$ HBC-ABC > 適応型 CRI-PSO > \mathcal{A}_{PSO} > Linear PSO = CS > 適応型 CS = CMD-CS > ABC > CRI-PSO > AFPSO > PSO = CMA-ES

以上の結果から、下記のことが確認できる。

様々な探索時間や次元数において、適応型 CRI-PSO の性能が全般的に最も優れている。この理由は、回転不変性と適応的パラメータ調整則の付加、さらに $Z = \mathcal{P}(k)$ としたことによる p-best 集団の活用によって、最も高いロバスト性・適応性を具備すると同時に、最適化能力が向上したためだと考えられる。

次に性能が優れているのは、HBC-ABC と ABC である。これは、ABC はいずれかの標準基底の方向に沿って近傍を生成するため、少しづつ探索点を徐々に有望領域に移動できる上に、停滞した探索点を再配置することで、持続的な探索に貢献しているため、元々最適化能力が優れていると考えられる。HBC-ABC は、その ABC に対して回転不変性を付加することによって、さらに性能が改善すると考えられる。しかしながら、ABC の近傍生成はいずれかの標準基底の方向に沿って行うため、探索点群の収束が遅い上に、次元が増えるほどさらに遅くなるため、短期や高次元では、性能が劣化する。一方、HBC-ABC では、超球化することで、様々な方向に沿って近傍生成が可能となり、この現象が起らなければ、短期・高次元でも優れた性能を発揮するため、HBC-ABC のほうが優れていたと考えられる。

その次に性能が優れているのは、CMA-ES と AFPSO、適応型 CS で、適応的パラメータ調整則を具備するアルゴリズムである。適応的パラメータ調整則を付加することで、様々な条件においても「適応」し、多様化・集中化を確実に実現し、平均的な性能を底上げしていると考えられる。特に CMA-ES はその亜種が国際会議でのコンペティションにおいて最も

成果を挙げている, state of the art なメタヒューリスティクスであるといえる。CMA-ES は低次元において最も優れている。

それ以降のアルゴリズムは, オリジナルのメタヒューリスティクスか, 不変性を付加した CRI-PSO と CMD-CS, \mathcal{A}_{PSO} である。適応型メタヒューリスティクスに比べて, オリジナルの PSO や CS は, 探索条件が限定されている場合は優れているが, パラメータの値に依存しており, 適応性が低いためだと考えられる。また, 不変性を付加したアルゴリズムも, 目的関数の特定の性質に対してはロバストとなるが, それ以外の性質には対応できず, 適応性が低いことから, パラメータの問題はオリジナルと同様である。これは本論文の問い合わせである, ユーザにアルゴリズムの選択やパラメータ設定など, 試行錯誤がかかる음을表す。一方, 適応型 CRI-PSO や適応型 CS, HBC-ABC はロバスト性・適応性の比較的高いアルゴリズムであり, パラメータ設定を試行錯誤せずに平均的に優れた性能を発揮することを確認した。特に適応型 CRI-PSO はこの分野で優れている CMA-ES よりも全体的に優れていることを確認した。

以上から, オリジナルよりも高いロバスト性・適応性を具備するメタヒューリスティクスを設計できたといえる。

表 6.3：全般的な探索性能に関する数値実験結果（アルゴリズムの総合比較、評価回数 $T_{\max} = 2000$ ）

関数	N	PSO	CRL-PSO	AFPSO	適応型CRL-PSO	A_{PSO}	Linear PSO	ABC	HBC-ABC	CS	CMD-CS	適応型CS	CMA-ES
			0.00	0.00	*	0.09	0.27	0.00	0.00	0.71	0.74	2.63	0.00
1. Sphere	10	0.00	0.01	0.00	*	30.5	30.2	88.8	7.26	295	29.9	126	128
	50	43.2	41.8	26.3	■	21.6	*	74.3	397	49.5	689	65.4	131
	100	443	120	169	■	83.9	*	1528	■	329	2264	219	416
2. Rosenbrock	300	2225	449	1086	■	412	*	202	■	1921	19.71	43.2	2263
	10	8.60	8.35	■	8.42	◆	8.88	16.3	11.4	9.95	8.17	6.93	*
	50	2236	606	574	■	380	*	577	◆	602	2736	198	459018
3. 2 ^N minima	100	29456	1970	3759	■	1570	*	1286	■	15367	920	1215876	19129
	300	111619	9342	35519	■	7877	*	4545	●	5426	93025	6620	4371190
	10	-723	■	-682	◆	-695	◆	-685	-635	-763	●	-666	-555
4. Rastrigin	50	-1955	-2509	◆	-2944	●	-2730	■	-2178	-2317	*	-2835	*
	100	-2285	-4257	■	-5001	●	-4595	*	-3821	-4101	-4145	-3989	1170
	300	-4931	-10197	-8123	■	-10820	*	-10665	◆	-11013	*	-7104	-9932
5. Schwefel	10	21.2	18.40	■	15.0	*	18.4	◆	32.6	26.3	6.05	●	90.7
	50	460	292	■	265	*	254	●	368	353	318	◆	372
	100	1364	766	*	784	■	653	●	831	◆	1047	874	1605
6. Levy	300	5029	2809	*	3462	■	2660	●	2858	■	2964	◆	4475
	10	0.35	0.07	◆	0.04	■	0.01	*	0.62	0.75	5.91	0.10	21.0
	50	297	70.7	153	■	68.3	◆	56.4	*	60.3	■	317	38.3
7. Ackley	100	1469	314	700	■	320	*	285	■	248	*	1246	193
	300	15518	2878	■	7204	■	2891	◆	2476	■	2090	*	11541
	10	0.03	*	1.04	◆	0.12	◆	0.66	2.72	3.86	0.05	0.79	43.2
8. Griewank	50	40.5	12.7	*	15.4	◆	10.7	*	16.3	19.1	20.7	14.7	110
	100	87.5	21.6	■	30.1	■	16.6	*	21.5	24.6	55.1	23.2	139
	300	143	33.5	63.1	■	29.3	*	27.2	●	32.1	■	115	33.4
9. Alpine	10	0.17	*	1.22	0.28	■	0.66	2.16	2.57	0.32	0.56	7.30	3.36
	50	5.53	4.73	4.75	4.04	*	4.32	■	4.69	6.17	3.86	●	9.60
	100	8.70	5.34	6.36	5.05	*	4.51	●	4.89	8.12	4.87	■	9.92
300	10.1	5.92	8.17	5.68	4.69	*	4.69	●	5.03	9.60	5.56	◆	10.3
	10	0.25	0.12	◆	0.11	*	0.11	■	0.44	0.28	0.15	0.34	0.27
	50	2.16	2.00	1.64	1.54	◆	1.73	*	1.73	1.80	3.23	1.20	1.07
300	100	12.3	3.94	5.13	3.20	*	2.58	◆	2.87	10.8	2.26	■	1.16
	10	56.6	11.6	29.0	11.2	*	6.30	■	7.31	◆	49.3	9.38	1.58
	50	0.55	0.71	0.08	*	0.58	1.66	2.43	0.15	1.94	3.17	1.94	0.03
300	100	191	85.4	96.6	68.1	*	21.8	●	33.8	39.6	34.5	34.3	36.6
	10	794	328	490	297	*	265	■	265	*	316	305	428
	50	61.2	30.2	25.4	*	21.8	●	77.3	■	98.3	137	81.5	127

表6.4：全般的な探索性能に関する数値実験結果（アルゴリズムの総合比較、評価回数 $T_{\max} = 20000$ ）

関数	N	PSO		CRL-PSO	AFPSO	適応型CRL-PSO	A_{PSO}	Linear PSO	ABC	HBC-ABC	CS	CMD-CS	適応型CS	CMA-ES	
		0.00	0.01	0.00	0.00	0.93	0.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
1. Sphere	10	0.00	■	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	●	
	50	0.01	40.8	0.08	0.00	◆	29.1	32.1	0.00	0.00	0.18	14.1	0.10	0.00	
	100	11.6	120	12.8	0.03	*	1.31	72.8	0.09	0.00	2.09	35.5	3.84	1.24	
2. Rosenbrock	300	2221	437	481	49.7	*	210	254	2.32	0.91	50.2	123	60.3	◆	
	10	1.31	■	10.2	4.13	0.41	*	17.5	12.3	2.32	3.48	4.63	4.19	2.78	
	50	100	624	84.8	47.6	*	538	590	102	48.6	79.0	278	50.5	◆	
3. 2 ^N minima	100	1831	2176	531	104	*	1366	1521	408	102	301	719	143	■	
	300	111776	8829	9170	1287	■	4185	5666	9028	743	2427	2599	1010	*	
	10	-727	-674	-761	■	-689	-639	-663	-783	*	-744	-734	-783	*	
4. Rastrigin	50	-3344	-2533	-3386	■	-3316	-2187	-2556	-3889	*	-3054	-3371	-2578	*	
	100	-2480	-4249	-6451	◆	-6573	*	-3892	-4125	-7437	*	-5539	-6158	-4417	■
	300	-4898	-10313	-15788	■	-17368	*	-10784	-11091	-18749	*	-13038	-14714	-11611	*
5. Schwefel	10	9.13	◆	21.9	5.17	*	16.5	21.66	23.1	0.00	14.9	22.3	25.9	6.41	
	50	186	291	145	◆	*	118	350	354	10.3	210	266	306	210	
	100	592	752	458	◆	250	*	823	841	83.3	583	687	727	350	
6. Levy	300	5015	2835	2591	1112	●	2813	2956	1306	*	2378	2614	2572	1960	
	10	0.00	*	0.06	0.00	0.00	0.72	0.60	0.53	0.00	◆	0.00	0.00	●	
	50	34.1	69.2	24.9	1.02	*	63.2	60.6	126	2.90	6.15	14.7	9.12	45	
7. Ackley	100	363	324	265	54.4	◆	248	236	541	34.1	*	36.4	63.6	49.1	
	300	4968	2886	3333	928	*	2455	2141	4627	452	*	375	627	612	
	10	0.04	1.26	0.02	0.02	2.69	3.48	0.00	*	0.01	◆	0.64	1.20	0.00	
8. Griewank	50	3.83	◆	13.4	3.05	■	4.97	15.2	18.5	0.00	*	7.31	10.9	5.71	
	100	16.9	22.1	11.4	7.22	■	20.8	25.4	0.15	12.7	14.0	14.7	8.38	◆	
	300	136	32.8	34.8	13.3	*	26.9	30.5	10.9	20.1	◆	20.7	20.2	15.5	
9. Alpine	10	0.15	1.24	0.00	0.07	2.19	2.31	0.00	*	0.17	0.04	0.11	0.01	◆	
	50	3.13	4.68	1.91	2.15	■	4.28	4.58	0.00	*	3.04	1.27	3.28	0.48	
	100	5.10	5.33	4.01	3.19	*	4.57	4.90	1.03	*	3.74	2.29	3.70	2.77	
300	10.1	5.91	6.83	4.58	4.70	5.05	4.84	4.33	◆	3.56	*	4.01	3.62	*	
	10	0.09	0.12	0.09	0.13	0.37	0.24	0.01	*	0.02	■	0.06	0.14	0.04	
	50	0.10	2.00	0.13	0.01	■	1.71	1.78	0.02	◆	0.01	0.83	1.36	0.15	
300	100	1.33	4.09	1.30	0.13	*	2.60	2.83	0.32	◆	0.05	1.21	1.89	0.71	
	10	0.00	*	0.72	0.00	◆	0.11	1.00	2.21	0.00	■	0.01	0.48	0.00	
	50	2.91	28.3	1.74	■	7.67	32.2	37.9	0.24	*	15.1	17.9	23.8	0.07	
300	100	32.6	83.9	21.7	*	29.7	76.9	93.0	3.37	*	48.6	45.0	57.2	85.8	
	10	618	322	286	166	*	259	300	115	186	■	191	208	222	
	50	618	322	286	166	*	259	300	115	186	■	191	208	222	

6.7 まとめ

本章では、多様な環境下での高いロバスト性の発揮を可能にするメタヒューリスティクスのフレームワークとしてCMD-DSを提案した。さらに、CMD-DSの変換不变性定理を導出すると共に、CMD-DSに基づきPSOに着想を得たサンプルアルゴリズムを設計した後、数値実験を通じてその有効性を確認した。また、本論文で提案した方法を含む各種メタヒューリスティクスの全般的性能に関する総合比較を行い、提案手法の有効性を確認した。

本章の成果は、下記の点で新規性を有すると考えられる。

- (1) 既存のアプローチのように、メタヒューリスティクスを発見的手法として構築した後に不变性を付加するのではなく、不变性の具備が理論的に示されているアルゴリズムのフレームワークに従うことで、具体的にメタヒューリスティクスを設計する方法を提案している。
- (2) フレームワークが規定する条件を満たす範囲であれば自由にアルゴリズムを設計することが可能で、アルゴリズムに高度な最適化能力を付与することができる。例えば、PSOやABCなど、既存の優秀なメタヒューリスティクスの最適化能力をフレームワークに組み込むことで、その最適化能力を継承しながら、不变性を具備するアルゴリズムの設計が可能である。

既存のアプローチの場合、メタヒューリスティクスにアフィン変換不变性を付加することは困難であるため、BBOの最適化プロセスにおいて、ユーザが適切にアルゴリズムを選択する試行錯誤が必要である。これに対して、本章で提案したCMD-CSに従って設計されたアルゴリズムは、アフィン変換不变性の具備が理論的に保証され、変数間依存性・悪スケール性などの目的関数の性質に対する高いロバスト性を発揮できるため、アルゴリズム選択に関する労力を削減でき、最適化アルゴリズムの設計に大きく貢献すると思われる。

CMD-DSに基づくアルゴリズム設計の発展的な方向性としては、下記が挙げられる。

- (1) もしBBOでなければ、直接探索法である必要性ではなく、目的関数の解析的情報（勾配やHesse行列など）が使用できるため、この情報を最適化能力としてフレームワークに組み込むことも可能である。
- (2) 第5章のように、CMD-DSに対してパラメータ調整則により適応性を付加する

ことで、平均的な性能を更に向上する、つまり、BBOにおける高いロバスト性・適応性を具備するアルゴリズム設計が期待できる。一方、適応的パラメータ調整則の設計は自由度が残されているため、本論文のアプローチを使用する以外に、[108, 126, 158]のように、パラメータ調整則を構成する演算子などの集合を解とみなし、それを対象関数に対して実行したときの探索性能を得るというプロセスを複数回繰り替えず上位の最適化問題を Genetic Programming が解くことで、使用したアルゴリズム・対象関数に対して最適なパラメータ調整則を自動的に設計する方法も有効だと考えられる。ただし、この方法は、大量試行が可能な状況でしか使用できないという欠点があるが、BBOにとって有効なアプローチであり、本論文の研究の有用性の更なる向上が期待できる。

一方、CMD-DS の課題として、計算コストの高さが挙げられる。CMD-DS では、各反復において共分散行列の固有値分解（Cholesky 分解など）が必要となるが、この計算量は通常、次元数 N に対して $\mathcal{O}(N^3)$ のオーダーとなる。このため、長期探索や高次元において適用する際には、計算コストが比較的高くなることから、計算速度の削減が望まれる。この課題に対してはいくつか解決策が挙げられる。

- (1) CMD-DSにおいて共分散行列を使用するのは、アフィン変換不变性の具備を保証するためだが、実用的には、その役割を損なわない範囲で適切に厳密性を欠如させてもよいと考えられる。例えば、固有値分解の近似アルゴリズムなどを導入することで、計算を加速することが期待できる。
- (2) 6.4 節において、CMD-DSに基づきサンプルアルゴリズムを設計したとき、共分散行列の対象となる解集団 \mathcal{Z} を p-best の集団 $\mathcal{P}(k)$ と設定したが、この設定を工夫すれば計算量の削減が可能である。例えば、一つ前の反復における p-best の集団 $\mathcal{P}(k-1)$ などに設定して、探索点の更新則と共に分散行列の固有値分解の計算を GPU などによって並列化することが考えられる。他にも、共分散行列の固有値分解の計算を各反復で行うのではなく、一定間隔の反復だけで行うことにして、その間は同じ固有値・固有ベクトルを使用することが考えられる。これらの方法は、CMD-DS のアフィン変換不变性の具備が保証されると同時に、全体の計算時間を減らすことが可能である。

7

結論

7.1 本論文のまとめ

拡大しつつある最適化周辺技術の一層の活用を実現するためには、モデリング・シミュレーション技術との結合を前提とした新たな最適化アルゴリズムの開発と、最適化アルゴリズムとの結合を前提とした新たなモデリング・シミュレーション技術の開発を同時にかつ双方向・相補的に進め、最適化に関わる技術を発展的に再構築することが重要な課題となっている。特に、ブラックボックス最適化（BBO）では、人工知能（AI）やコンピュータパワーの飛躍的進化を背景にしたモデリング・シミュレーション技術の併用がますます進むと考えられる一方、最適化アルゴリズムは目的関数の解析的情報（勾配や Hesse 行列）を必要しない直接探索法が望ましい。BBO での環境下での適用を想定すると、メタヒューリスティクスを始めとする直接探索法には、ロバスト性と適応性の両方の具備が必要であり、それぞれに対しては変換不变性とパラメータ調整が貢献すると考えられる。

本論文では、以上の背景を踏まえ、BBO での高いロバスト性・適応性を具備した最適化アルゴリズムの開発を目指し、変換不变性とパラメータ調整を活用した、メタヒューリスティクスの設計に関する検討を行った。第 2 章では、BBO で問題となる目的関数の性質を述べた後、メタヒューリスティクスの最適化能力が近接最適性原理や多様化・集中化に基づいていることを述べた。第 3 章では、各種メタヒューリスティクスのアルゴリズムを具体的に挙げながら、その探索構造を解析した後、多様化・集中化の観点で、各アルゴリズムがどのように最適化能力を生み出しているのかを整理した。第 4 章では、本論文で中

核となる変換不变性の定義を挙げ、統一的な視点で数理的構造に着目しながら先行研究を整理することで、不变性が欠如する構造や不变性を付加するアプローチを分類すると同時に、数理的証明を通じて、各種メタヒューリスティクスの不变性の有無を具体的に明らかにした。その後、Particle Swarm Optimization (PSO), Artificial Bee Colony Algorithm (ABC), Cuckoo Search に対して、不变性を付加したCRI-PSO, HBC-ABC, CMD-CS を提案し、数値実験を通じて変換に対するロバスト性の向上を確認した。第5章では、本論文で中核となる先行研究におけるパラメータ調整を整理し、統一的な視点でパラメータ調整と不变性の関連性を明らかにし、不变性に基づくパラメータ調整則の指針を提示した。その後、各指針に従い、CS に対して、適応的パラメータ調整則を付加した相似変換不变性を補完した適応型 CS と、と第4章で提案したCRI-PSO に対して、適応的パラメータ調整則を付加した回転不变性を有する適応型 PSO を提案し、数値実験を通じてロバスト性の維持と適応性の向上を確認した。第6章では、第4章の既存のアルゴリズムに変換不变性を付加するアプローチとは逆に、アフィン変換不变性が保証されるアルゴリズムの「フレームワーク」を提示し、そのフレームワーク内で自由にアルゴリズムを設計する方法を提案した。このフレームワークに従い、サンプルアルゴリズムを設計し、数値実験を通じてアフィン変換不变性の具備を確認した後、PSO に着想を得たアルゴリズムを設計し、数値実験を通じて高いロバスト性の具備を確認した。

本論文で得られた主要な研究成果は、以下の通りである。

- (1) メタヒューリスティクスが発見的手法であるが故に、各種アルゴリズムの不变性の有無や先行研究における不变性を付加するアプローチが明確でないことを指摘し、統一的な視点でその数理的構造に着目しながら不变性に関する先行研究を整理することで、更新則の数理的構造と不变性の具備の関係を明らかにした。
- (2) 先行研究では、発見的手法である各種メタヒューリスティクスに対して、一部の不变性が主に実験的に示されてきたが、数学的な証明はされていないケースがほとんどであると同時に、特に、比較的新しいABC やCS に関しては全く不变性が不明であったことを指摘し、数学的証明を通じて、各種メタヒューリスティクス (PSO, DE, ABC, FA, CS) の不变性の有無を具体的に明らかにした。
- (3) 先行研究において他のアルゴリズムにおいてみられるが、そのアルゴリズムにおいてはまだ事例がないアプローチを流用することで、「共分散行列の固有ベクトルに

基づく基底変換」に基づき回転不变性を付加した CRI-PSO, 「超球表面上の近傍生成」に基づき回転不变性を付加した HBC-ABC, 「共分散行列の固有ベクトルと固有値に基づく基底変換」に基づきアフィン変換不变性を付加した CMD-CS を提案した。

- (4) (3) の検討を通じて, 先行研究を含めた, 既存のアルゴリズムに対して不变性を付加するアプローチは, ①従来有していた他の不变性を犠牲にすることがあり, 完全にアフィン変換不变性を付加することは困難であること, ②想定した性質(変数間依存性, 悪スケール性, 原点依存性)に対するロバスト性が向上するという点において有効であったが, 他の条件(探索時間, 多峰性などの問題構造)に対しては有効であるとは限らないこと, ③各種アルゴリズムの不变性と数理的構造に着目し, アプローチを適切に選択しているが, アルゴリズムに対する理解が薄いユーザにとってこの作業は困難であること, を指摘した。
- (5) (4) の①②の課題解決のために, 統一的な視点で変換不变性とパラメータ調整則の関係を明らかにすることで, 不变性に基づくパラメータ調整則の設計指針を提示した。
- (6) 不变性に基づくパラメータ調整則の設計指針に基づき, 適応的パラメータ調整則によるロバスト性の維持と適応性の向上を目的として, CS に対して適応的パラメータ調整則を付加することで相似変換不变性を補完した適応型 CS と, (3) で提案した CRI-PSO に対して, 適応的パラメータ調整則を付加した回転不变性を有する適応型 PSO を提案した。
- (7) (4) の①③の課題解決のために, (3) の既存のアルゴリズムに変換不变性を付加するアプローチとは逆に, アフィン変換不变性が保証されるアルゴリズムの「フレームワーク」を提示し, そのフレームワーク内で自由にアルゴリズムを設計する方法を提案した。これはフレームワークの範囲で設計されたすべてのアルゴリズムに対して適用可能であり, 要求条件さえ満たすようにアルゴリズムを設計すれば, アフィン変換不变性を具備することを表すものである。このフレームワークに従えば, アフィン変換不变性を具備することを保証しながら, 様々な最適化能力を自由に付与でき, さらにパラメータ調整則を付加することも可能である。つまり, 高いロバスト性と適応性の具備が可能な汎用的なフレームワークであるといえる。

7.2 今後の展望

また、当該分野における今後の主要な課題としては、以下の事項を挙げることができる。

- (1) 変換不变性を有するフレームワークに基づく新たなメタヒューリстиクスの設計
本論文の第7章では、変換不变性を有するフレームワークとしてCMD-DSを構築した後、CMD-DSの条件を満たす範囲でPSOの最適化能力を埋めこむことで、変換不变性を有する新たなメタヒューリстиクス \mathcal{A}_{PSO} を設計した。CMD-DSのフレームワークの汎用性を活用すれば、これと同様に、他のメタヒューリстиクスの最適化能力を埋め込むことで、様々なアルゴリズムを設計することが可能である。例えば、 \mathcal{A}_{PSO} では探索点の更新方法を絶対移動としていたが、Differential Evolution (DE) や Cuckoo Search では改善移動とすることで最適化能力を生み出している。CMD-DSのフレームワークにおいて探索点の更新方法を改善移動に変更しても、変換不变性定理の成立には影響しないため、そのまま拡張することが可能である。また、CMD-DSの \mathbf{u} の設計において、DEやArtificial Bee Colony Algorithm (ABC) のように、探索点間の差分ベクトルを導入するなどが考えられる。差分ベクトルの係数がスカラであれば、CMD-DSが課す条件を満たすため、容易にDEやABCの操作を取り入れることが可能である。他にも、第2章では、大域的最適化アルゴリズムや古典的な直接探索法を概観し、第5章では、CMA-ESなどのパラメータ調整則を概観したが、条件を満たす範囲であれば、既存のメタヒューリстиクスに限らず、これらのアルゴリズムの操作も導入することが可能である。例えば、滑降シンプレックス法における最悪点から探索点群の重心に向かう差分ベクトルや、CMA-ESのWeight RecombinationにおけるRankに基づく加重付き平均ベクトル、これを発展させて、p-bestのRankに基づく加重付き平均ベクトルなど、有用な量を導入することが考えられる。本論文ではPSOの最適化能力を導入した設計例しか示せなかったが、上記のように、CMD-DSの高い汎用性を活用し、メタヒューリстиクスに限定せずに様々なアルゴリズムの概念と融合することで、より有用なブラックボックス最適化向けの直接探索法を設計することが期待されるため、これらの検討が重要な課題である。
- (2) 変換不变性の他の問題クラスへの拡張

本論文では、单一目的・連続変数・無制約最適化問題のみを扱ったが、実応用上では多目的最適化、離散変数を扱う組合せ最適化、有制約最適化などが重要である。本論文で中核の概念であった変換不变性は、これらの問題クラスにおいても適用・拡張が可能だと考えられる。解空間の変換不变性は実数空間において定義されるため、現状では連続変数を扱う問題にしか適用できないと考えられる。一方、多くの最適化問題では目的関数の出力は実数である ($f : S \rightarrow \mathbb{R}$) ため、目的関数の変換不变性は上記の問題クラスにそのまま適用・拡張可能である。例えば、連続変数の多目的最適化では、決定変数空間上の操作は解空間の変換不变性が適用でき、目的関数空間上の操作は目的関数の変換不变性が適用することが可能である。他にも、有制約への拡張方法として制約対処法 (Constraint Handling) が知られているが、この制約対処法のアルゴリズムにおいても、連続変数の決定変数空間上の操作は解空間の変換不变性が適用できると同時に、複数の制約違反量関数に対しても変換を与えることを考えると、目的関数の変換不变性の概念をそのまま拡張することで、「制約違反量関数の変換不变性」の導入が可能である。組合せ最適化では、決定変数が組合せ的であるため、本論文の解空間の変換不变性は適用不可だが、目的関数の変換不变性はそのまま適用可能であると同時に、有制約組合せ最適化に拡張しても連続・有制約最適化と同様に、複数の制約違反量関数に対する変換不变性の議論が可能である。このように、一部限定されるが、実応用上により近い問題クラスにおいても高いロバスト性を具備するアルゴリズムを設計するためには、変換不变性の適用・拡張は重要な課題である。

(3) 他の研究成果における理論との融合

本論文の中核となる概念であった変換不变性は最適化アルゴリズム中の写像に対する理論であるため、元々発見的手法として誕生したメタヒューリティクスに対して強力な理論となり得る。一方で、近年、メタヒューリティクスの探索、動特性・安定性・収束性に関する理論的研究も進んでいる。例えば、メタヒューリティクスの更新則を Markov 連鎖とみなすことで、収束性を導出した研究がある。なお、この見方は、本論文の第 3 章における更新則の定義と類似している。他にも、本論文の CMD-DS は乱数ベクトルを扱うため、Evolution Strategy (ES) の成果が一部活用できることが考えられる。ES の理論的研究はメタヒューリティクスの中でも最も進んでいるといえる。変換不变性に関する理論とこれらの理論

を共存させることで、「理論的な保証がされたメタヒューリスティクス」という新たな最適化アルゴリズムのパラダイムへとシフトさせられる可能性があるため、これらは重要な課題である。

(4) 高いロバスト性・適応性を具備する逐次近似最適化の開発

直接探索型の最適化手法の特徴を最大限に活かした、最適化技術とモデリング・シミュレーション技術を結合した最適化の新たな枠組みの構築、さらにはモデリング・シミュレーション技術との結合を前提とした最適化手法の開発と、最適化手法との結合を前提としたモデリング・シミュレーション技術の開発による最適化技術の発展的再構築においてメタヒューリスティクスは主要な技術となり得る。第2章で述べた逐次近似最適化は、この枠組みによく合致する。さらに、逐次近似最適化では、本論文で提案した最適化アルゴリズムだけでなく、サンプリングアルゴリズムも使用するが、全体として高いロバスト性・適応性を具備するためには、サンプリングアルゴリズム自身もロバスト性・適応性に着目することも重要な課題だと考えられる。また、従来の逐次近似最適化では、モデリング方法として Radial Basis Neural Network が主流であったが、Bayes 最適化で使用されるような Gauss 過程などのカーネル法や深層学習など、第3次 AI ブームを牽引する技術を積極的に導入することも考えられる。このように、最適化アルゴリズムに限らず、モデリング技術などの外部環境を広く概観した上で、総合的な最適化技術を構築することが重要な課題である。

(5) 変換不变性を有するメタヒューリスティクスのフレームワークとパラメータ調整則の融合

本論文では、①既存のアルゴリズムに対する不变性の付加、②①のアルゴリズムに対するパラメータ調整則の付加、③変換不变性のフレームワークの提案、を行ってきたが、更なるロバスト性・適応性の向上のためには、③で設計した具体的なアルゴリズムに対してパラメータ調整則を付加することが考えられる。これは、アフィン変換不变性が保証されていながら、自由に最適化能力を埋め込めるため、②よりも優秀なアルゴリズムが設計できることが期待できる。一方、適応的パラメータ調整則の設計は自由度が残されているため、本論文のアプローチを使用する以外に、[108, 126, 158]のように、パラメータ調整則を構成する演算子などの集合を解とみなし、それを対象関数に対して実行したときの探索性能を得るというプロセスを

複数回繰り替えず上位の最適化問題を Genetic Programming が解くことで、使用したアルゴリズム・対象関数に対して最適なパラメータ調整則を自動的に設計する方法も有効だと考えられる。ただし、この方法は、大量試行が可能な状況でしか使用できないという欠点があるが、BBO にとって有効なアプローチであり、本論文の研究の有用性の更なる向上が期待できる。

謝辞

本論文は、著者が東京都立大学大学院 理工学研究科 電気電子工学専攻¹において、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、多くのご指導、ご助言を頂きました。心より御礼申し上げます。

東京都立大学大学院 理工学研究科 栄久保 文嘉 教授には、学位論文の審査委員として、主として最適化の応用の視点から具体的かつ有益なご助言をいただきました。東京都立大学大学院 理工学研究科 相馬 隆郎 准教授には、学位論文の審査委員として、本研究の位置づけや意義を中心に学位論文の構成に関する具体的かつ有益なご助言をいただきました。心より御礼申し上げます。

東京都立大学大学院 システムデザイン研究科 電子情報システム工学域 システム制御工学研究室² の皆様には、学位論文の作成の過程で多大なるご協力をいただきました。心より御礼申し上げます。

社会人博士課程制度の利用にあたり、横河電機株式会社 マーケティング本部 イノベー

¹ 東京都立大学（旧 首都大学東京）は、2018年4月の再編と2020年4月の名称変更があり、2020年5月現在、旧組織と新組織が混合して存在している。著者が入学した2017年10月時点の所属は「首都大学東京大学院 理工学研究科 電気電子工学専攻」であった。2018年4月に旧 首都大学東京では学部再編があり、当研究室の所属は新組織において「首都大学東京大学院 システムデザイン研究科 電子情報システム工学域」となったため、それ以降に入学した学生は新組織所属となったが、それ以前に入学した著者を含む学生は旧組織所属のままであった。よって、当研究室は旧組織と新組織の両方に所属していたことになる。さらに、2020年4月以降、「首都大学東京」は「東京都立大学」に名称変更したことに伴い、新組織名称は「東京都立大学大学院 システムデザイン研究科 電子情報システム工学域」に、旧組織名称は「東京都立大学大学院 理工学研究科 電気電子工学専攻」に変更された。よって、著者の所属は後者である。

² 現在の学生は全員新組織所属であるため。

ションセンターの皆様には業務の面で多大なる便宜を図っていただきましたことを付記して、関係各位への謝辞といたします。

参考文献

1. 参考文献

- [1] 土屋淳一, 鈴木隆之, 安田恵一郎:「サーフェスマ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, 竹下聰彦, 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] 理化学研究所 計算科学研究センター 「富岳」による新型コロナウイルスの治療薬候補同定, <https://www.r-ccs.riken.jp/jp/fugaku/projects/okuno.html> (2020年8月4日アクセス)
- [13] Y. LeCun, U. Bengio, and G.Hinton: “Deep Learning,” Nature, Vol.521, pp.436-444 (2015)
- [14] 岡谷貴之:「画像認識のための深層学習」, 人工知能学会誌, Vol.28, No.6, pp.962-974 (2013)
- [15] 藤吉弘亘, 山下隆義:「深層学習による画像認識」, 日本ロボット学会誌, Vol.35, No.3, pp.180-185 (2017)
- [16] 久保陽太郎:「音声認識のための深層学習」, 人工知能学会誌, Vol.29, No.1, pp.62-71 (2014)
- [17] M. Tsubaki and T. Mizoguchi: “Fast and Accurate Molecular Property Prediction: Learning Atomic Interactions and Potentials with Neural Networks,” Journal of Physical Chemistry Letters, Vol.9, No.19, pp.5733-5741 (2018)
- [18] W. Andrew, E. Richard, J. John, K. James, S. Laurent, G. Tim, Q. Chongli, Ž. Augustin, W. R. N. Alexander, B. Alex, P. Hugo, P. Stig, S. Karen, C. Steve, K. Pushmeet, T. J. David, S. David, K. Koray, and H. Demis: “Improved Protein Structure Prediction Using Potentials from Deep Learning,” Nature, Vol.577, pp.706-710 (2020)
- [19] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Driessche, T. Graepel, and D. Hassabis: “Mastering the Game of Go without Human Knowledge,” Nature, Vol.550, No.7676, pp.354-359 (2017)
- [20] 田中輝雄, 深川正一, 井村淳一, 後保範, 玉置由子, 榊原忠幸, 稲上泰弘:「クラスタ型ベクトル並列スーパコンピュータ A - 3000 クラスタシステムのアーキテクチャと性能評価」, 情報処理学会 研究報告ハイパフォーマンスコンピューティング, Vol.1995, No.81(1995-HPC-057), pp.49-54 (1995)
- [21] 宮田一乗, 高橋誠史, 黒田篤:「GPU コンピューティングの動向と将来像」, 芸術科学会論文誌, Vol.4, No.1, pp.13-19 (2005)
- [22] 新庄貞昭, 高橋誠史, 木村秀敬, 白井暁彦, 宮田一乗:「GPU の先駆的利用の研究動向と将来像」, 芸術科学会論文誌, Vol.6, No.3, pp.167-178 (2007)

- [23] 藤本典幸：「GPU を用いた超並列高速計算入門—I—GPU ハードウェアの概要」，システム／制御／情報，Vol.60，No.2，pp.79-85（2016）
- [24] 青木尊之：「GPU を用いた超並列高速計算入門—IV—GPU スパコンによる大規模物理シミュレーション」，システム／制御／情報，Vol.60，No.8，pp.350-357（2016）
- [25] Top500, <https://www.top500.org/lists/2019/11/> (Last accessed 1-May-2020)
- [26] 今野浩，山下浩 編著：「非線形計画法」，日科技連（1978）
- [27] 志水清孝，相吉英太郎 共著：「数理計画法」，昭晃堂（1984）
- [28] 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）
- [29] 今野浩 著：「数理決定法入門 キャンパスの OR」，朝倉書店（1992）
- [30] 福島雅夫 著：「非線形最適化の基礎」，朝倉書店（2001）
- [31] 玉置久 編著：「システム最適化」，オーム社（2005）
- [32] 矢部博 著：「工学基礎 最適化とその応用」，数理工学社（2006）
- [33] 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）
- [34] G. Dantzig: "Linear Programming and Extensions," Princeton University Press (1998)
- [35] 田中謙輔 著：「凸解析と最適化理論」，牧野書店（1994）
- [36] 棣田充俊，高橋則雄，村松和弘，山田忠治，小川誠，小林晋，桑原徹：「三次元有限要素法と Rosenbrock 法による永久磁石式リターダの最適設計」，電気学会 産業応用部門誌，Vol.118，No.6，pp.780-787（1998）
- [37] 西田英幸，中沢親志，福山良和：「オンライン最適化技術と制御プラットフォーム「FeTOP」」，富士時報，Vol.79，No.3，pp.274-278（2006）
- [38] J. Bergstra and Y. Bengio: "Random Search for Hyper-Parameter Optimization," Journal of Machine Learning Research, Vol.13, pp.281-305 (2012)
- [39] T. Elsken, J. H. Metzen, and F. Hutter: "Neural Architecture Search: A Survey," Journal of Machine Learning Research, Vol.20, pp.1-21 (2019)
- [40] 知京豊裕：「マテリアルズインフォーマティクスの現状と課題」，情報知識学会誌，Vol.27，No.4，pp.297-304（2017）

- [41] 阿久津達也, 秋山泰:「バイオインフォマティクスと人工知能の相互作用」, 人工知能学会誌, Vol.22, No.1, pp.42-48 (2007)
- [42] R. M. Lewis, V. Torczon, and M. W. Trossetca: “Direct Search Methods: Then and Now,” Journal of Computational and Applied Mathematics, Vol.124, No.1–2, pp.191-207 (2000)
- [43] R. Hooke and T. A. Jeeves: ““Direct Search” Solution of Numerical and Statistical Problems,” Journal of the ACM, Vol.8, No.2, pp.212-229 (1961)
- [44] 伊藤駒之:「Hooke-Jeeves の方法の一変種」, 国民経済雑誌, Vol.133, No.5, pp.52-66 (1976)
- [45] J. A. Nelder and R. Mead: “A Simplex Method for Function Minimization,” Computer Journal, Vol.7, No.4, pp.308–313 (1965)
- [46] python SciPy.org, <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-neldermead.html> (Last accessed 1-May-2020)
- [47] MATLAB fminsearch ソルバ, <https://jp.mathworks.com/help/matlab/ref/fminsearch.html> (2020年5月1日アクセス)
- [48] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Posík: “Comparing Results of 31 Algorithms from the Black-box Optimization Benchmarking BBOB-2009,” Proceedings of Genetic and Evolutionary Computation Conference, pp.1689-1696 (2010)
- [49] S. H. Brooks: “A Discussion of Random Methods for Seeking Maxima,” Operations Research, Vol.2, No.6, pp.244–251 (1958)
- [50] 室田一雄:「離散構造とアルゴリズム IV」, 近代科学社, 1995.
- [51] 柳浦睦憲, 萩木俊秀 著:「組合せ最適化—メタ戦略を中心として—」, 朝倉書店 (2001)
- [52] 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)
- [53] 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)
- [54] 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)
- [55] X. S. Yang: “Nature-Inspired Metaheuristic Algorithms,” Second Edition, Luniver Press (2010)

- [56] I. Boussaïd, J. Lepagnot, and P. Siarry: “A Survey on Optimization Metaheuristics,” *Information Sciences*, Vol.237, pp.82-117, Elsevier Publisher (2013)
- [57] H. -P. Schwefel: “Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie,” Birkhäuser (1977) (in German)
- [58] H. -P. Schwefel: “Evolution and Optimum Seeking,” Wiley (1995)
- [59] J. H. Holland: “Adaptation in Natural and Artificial Systems,” University of Michigan Press (1975)
- [60] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning,” Addison-Wesley (1989)
- [61] 三宮信夫, 喜多一, 玉置久, 岩本貴司:「遺伝アルゴリズムと最適化」, 朝倉書店 (1998)
- [62] 伊庭齊志:「遺伝的アルゴリズムの基礎」, オーム社 (1998)
- [63] S. Kirkpatrick, D. Gelatt Jr., and M. P. Vecchi: “Optimization by Simulated Annealing,” *Science*, Vol.220, No.4598, pp.671-680 (1983)
- [64] E. H. L. Aarts and J. H. M. Korst, “Simulated Annealing and Boltzmann Machines,” John Wiley & Sons (1989)
- [65] E. H. L. Aarts and J. K. Lenstra, eds., “Local Search in Combinatorial Optimization,” John Wiley & Sons (1997)
- [66] F. Glover: “Future Paths for Integer Programming and Links to Artificial Intelligence,” *Computers & Operations Research*, Vol.13, No.5, pp.533-549 (1986)
- [67] F. Glover and M. Laguna, “Tabu Search,” Kluwer Academic Publishers, 1997.
- [68] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” Proceedings of IEEE International Conference on Neural Networks, Vol.4, pp.1942-1948 (1995)
- [69] 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)
- [70] D. Karaboga: “An Idea Based on Honey Bee Swarm for Numerical Optimization,” Technical Report TR06, Computer Engineering Department, Engineering Faculty, Erçiyas University (2005)
- [71] X. S. Yang: “Nature-Inspired Metaheuristic Algorithms,” Luniver Press (2008)

- [72] X. S. Yang: "Firefly Algorithms for Multimodal Optimization," Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Sciences, Vol.5792, pp.169-178 (2009)
- [73] 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)
- [74] X. S. Yang and S. Deb: "Cuckoo Search: Recent Advances and Applications," Neural Computing and Applications, Vol.24, No.1, pp.169-174 (2014)
- [75] D. Vanderbilt and S. G. Louie: "A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables," Journal of Computational Physics, Vol.56, pp.259-271 (1984)
- [76] L.J. Eshelman and J.D. Schaffer: "Real-Coded Genetic Algorithms and Interval-Schemata," Foundations of Genetic Algorithms, Vol.2, pp.187-202 (1993)
- [77] T. Bäck and H. -P. Schwefel: "An Overview of Evolutionary Algorithms for Parameter Optimization," Evolutionary Computation, Vol.1, No.1, pp.1-23 (1993)
- [78] N. Hansen, A. Ostermeier, and A. Gawelczyk: "On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation," Proceedings of the 6th International Conference on Genetic Algorithms, pp.57-64 (1995)
- [79] N. Hansen and A. Ostermeier: "Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation," Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, pp.312-317 (1996)
- [80] C. -P. Erick: "Designing Efficient and Accurate Parallel Genetic Algorithms," Genetic Algorithms and Evolutionary Computation, Springer (2001)
- [81] 筒井茂義:「並列処理による進化計算の高速化について」, 阪南論集 社会科学編, Vol.45, No.3, pp147-164 (2010)
- [82] 筒井茂義:「GPU を用いた超並列高速計算入門ーIVー進化計算への適用」, システム／制御／情報, Vol.60, No.10, pp.443-450 (2016)
- [83] 小川雅俊, 鈴木泰政, 大貝晴俊, 草鹿仁:「エンジン応答局面モデルと制御パラメータ最適化手法の開発」, 計測自動制御学会論文集, Vol.47, No.10, pp.501-510 (2011)
- [84] 鈴木亮平, 北川慎治, 松井哲郎, 福山良和:「 ε 制約メタヒューリстиクスを用いたエネルギー・プラン最適運用」, 電気学会研究会資料 産業計測制御研究会, IIC-11-49, pp.1-6 (2011)

- [85] 篠崎隆宏, 渡部晋治:「音声認識とブラックボックス最適化」, 日本音響学会誌, Vol.72, No.10, pp.644-652 (2016)
- [86] I. Loshchilov and F. Hutter: "CMA-ES for Hyperparameter Optimization of Deep Neural Networks," Proceedings of the 33th International Conference on Machine Learning (ICML 2016), Workshop Track, pp.1-8 (2016)
- [87] Y. Akimoto, S. Shirakawa, N. Yoshinari, K. Uchida, S. Saito, and K. Nishida: "Adaptive Stochastic Natural Gradient Method for One-Shot Neural Architecture Search," Proceedings of the 36th International Conference on Machine Learning, pp.171-180 (2019)
- [88] N. Hansen: "Invariance, Self-Adaptation and Correlated Mutations in Evolution Strategies," Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, pp.355–364 (2000)
- [89] N. Hansen and A. Ostermeier: "Completely Derandomized Self-Adaptation in Evolution Strategies," MIT Press, Evolutionary Computation, Vol.9, No.2, pp.159-195 (2001)
- [90] U. Paquet and A. P. Engelbrecht: "A New Particle Swarm Optimiser for Linearly Constrained Optimisation," Proceedings of the 2003 IEEE Congress on Evolutionary Computation, pp.1839-1844 (2003)
- [91] D. N. Wilke, S. Kok, and A. A. Groenwold: "Comparison of Linear and Classical Velocity Update Rules in Particle Swarm Optimization: Notes on Scale and Frame Invariance," International Journal for Numerical Methods in Engineering, Vol.70, pp.985-1008 (2007)
- [92] N. Hansen, R. Ros, N. Mauny, M. Schoenauer, and A. Auger: "PSO Facing Non-Separable and Ill-Conditioned Problems," RR-6447, INRIA, (2008), <https://hal.inria.fr/inria-00250078v1t> (Last accessed 1-May-2020)
- [93] N. Hansen, R. Ros, N. Mauny, M. Schoenauer, and A. Auger: "Impact of Invariance in Search: When CMA-ES and PSO Face Ill-Conditioned and Non-Separable Problems," Journal of Applied Soft Computing, Elsevier, pp.5755-5769 (2011)
- [94] M. Clerc: "Beyond Standard Particle Swarm Optimisation," International Journal of Swarm Intelligence, Vol.1, No.4, pp.46–61 (2010)
- [95] M. Clerc: "Standard Particle Swarm Optimisation," HAL Open Access Archive (2012), <https://hal.archives-ouvertes.fr/hal-00764996/document> (Last accessed 1-May-2020)

- [96] M. R. Bonyadi, Z. Michalewicz, and X. Li: “An Analysis of the Velocity Updating Rule of the Particle Swarm Optimization Algorithm,” *Journal of Heuristics*, Vol.20, No.4, pp.417-452 (2014)
- [97] M. R. Bonyadi and Z. Michalewicz: “A Locally Convergent Rotationally Invariant Particle Swarm Optimization Algorithm,” *Swarm Intelligence*, Vol.8, No.3, pp.159–198 (2014)
- [98] Y. Hariya, T. Kurihara, T. Shindo, and K. Jin’no: “A Study of Robustness of PSO for Non-Separable Evaluation Functions,” *Proceedings of 2015 International Symposium on Nonlinear Theory and its Applications*, pp.724-727 (2015)
- [99] Y. Hariya, T. Shindo, and K. Jin’no: “An Improved Rotationally Invariant PSO: Modified Standard PSO-2011,” *Proceedings of 2016 IEEE Congress on Evolutionary Computation*, pp.1839-1844 (2016)
- [100] T. Shindo and K. Jin’no: “A Novel Particle Swarm Optimization Algorithm for Non-Separable and Ill-Conditioned Problems,” *Proceedings of 2016 IEEE International Conference on Systems, Man, and Cybernetics*, pp.2110-2115 (2016)
- [101] Y. Hariya, T. Shindo, and K. Jin’no: “On a Rotationally Invariant of PSO,” *Proceedings of 2016 International Symposium on Nonlinear Theory and its Applications*, pp.1839-1844 (2016)
- [102] 串田淳一, 原章, 高濱徹行:「回転不変性を考慮した Particle Swarm Optimization の改良」, 進化計算学会 進化計算シンポジウム 2016, pp.88-95 (2016)
- [103] A. M. Sutton, M. Lunacek, and L. D. Whitley: “Differential Evolution and Non-separability: Using Selective Pressure to Focus Search,” *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pp.1428–1435 (2007)
- [104] 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)
- [105] Y. -L. Li, W. -N. Chen, and J. Zhang: “Differential Evolution Algorithm with PCA-based Crossover,” *Proceedings of GECCO’ 12 Companion*, pp.1509-1510 (2012)
- [106] Y. Wang, H. -X. Li, T. Huang, L. Li: “Differential Evolution Based on Covariance Matrix Learning and Bimodal Distribution Parameter Setting,” *Applied Soft Computing*, Vol.18, No.C, pp.232–247 (2014)

- [107] J. Tvrdík and R. Poláková: "Differential Evolution with Rotation-Invariant Mutation and Competing-Strategies Adaptation," Proceedings of 2014 IEEE Congress on Evolutionary Computation, pp.2253-2258 (2014)
- [108] 金政実, 相吉英太郎:「回転不变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)
- [109] S-M. Guo and C-C. Yang: "Enhancing Differential Evolution Utilizing Eigenvector-Based Crossover Operator," IEEE Transactions on Evolutionary Computation, Vol.19, No.1, pp.31-49 (2015)
- [110] Y. Wang, Z. -Z., Liu, J. Li, H. -X. Li, and G. G. Yenf: "Utilizing Cumulative Population Distribution Information in Differential Evolution," Applied Soft Computing, Vol.48, No.C, pp.329-346 (2016)
- [111] F. Caraffni and F. Neria: "A Study on Rotation Invariance in Differential Evolution," Swarm and Evolutionary Computation, Vol.50, No.100436, pp.1-36 (2018)
- [112] F. Caraffni and F. Neria: "Rotation Invariance and Rotated Problems: An Experimental Study on Differential Evolution," Proceedings of International Conference on the Applications of Evolutionary Computation, pp 597-614 (2018)
- [113] M. Clerc and J. Kennedy: "The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space," IEEE Transactions on Evolutionary Computation, Vol.6, No.1, pp.58-73 (2002)
- [114] R. C. Eberhart and Y. Shi: "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization," Proceedings of the 2000 Congress on Evolutionary Computation, pp.84-88 (2000)
- [115] J. F. Schutte and A. A. Groenwold: "A Study of Global Optimization Using Particle Swarms," Jounal of Global Optimization, Vol.31, pp.93-108 (2005)
- [116] 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)
- [117] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer: "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," IEEE Transactions Evolutionary Computation, Vol.10, No.6, pp.646-657 (2006)

- [118] J. Zhang and A. C. Sanderson: "JADE: Adaptive Differential Evolution With Optional External Archive," IEEE Transactions Evolutionary Computation, Vol.13, No.5, pp.945-958 (2009)
- [119] R. Tanabe and A. Fukunaga: "Success-History Based Parameter Adaptation for Differential Evolution," Proceedings of the 2013 IEEE Congress on Evolutionary Computation, pp.71-78 (2013)
- [120] The Black-box Optimization Benchmarking (BBOB) Workshop, <http://numbbo.github.io/workshops/index.html> (Last accessed 1-May-2020)
- [121] 田邊遼司, 串田淳一, 畠中利治:「関数最適化における進化計算」, 計測自動制御学会 計測と制御, Vol.54, No.8, pp.567-572 (2015)
- [122] M. N. Ras, D. N. Wilke, and A. A. Groenwold: "On Rotationally Invariant Continuous-Parameter Genetic Algorithms," Journal of Advances in Engineering Software, Vol.78, pp.52-59 (2014)
- [123] Special Session & Competitions on Real-Parameter Single Objective Optimization, https://www.ntu.edu.sg/home/epnsugan/index_files/CEC2018/CEC2018.htm (Last accessed 24-July-2020)
- [124] C. W. Cleghorn and A. P. Engelbrecht: "Firefly Optimization: A Study on Frame Invariance," Proceedings of 2017 IEEE Symposium Series on Computational Intelligence, pp.423-428 (2017)
- [125] R. Ji, K. Tamura, and K. Yasuda: "Adaptive Parameter Adjustment of Differential Evolution," Proceedings of the 2013 IEEE International Conference on Systems, Man and Cybernetics, pp.3915-3920 (2013)
- [126] 金政実, 相吉英太郎:「多点型進化戦略と遺伝的プログラミングによる創発的パラメータ調整ルールの設計」, 電気学会 電子・情報・システム部門誌, Vol.135, No.3, pp.321-330 (2015)
- [127] 秋本洋平, 永田裕一, 佐久間淳, 小野功, 小林重信:「適応的実数地交叉 AREX の提案と評価」, 人工知能学会論文誌, Vol.24, No.6A, pp.446-458 (2009)
- [128] N. Hansen: "Adaptive Encoding: How to Render Search Coordinate System Invariant," Proceedings of the 10th International Conference on Parallel Problem Solving from Nature, Vol.5199 ,pp.205-214 (2008)
- [129] C. A. Coello-Coello: "Theoretical and Numerical Constraint-Handling Techniques Used with Evolutionary Algorithms: a Survey of the State of the Art," Computer Meth-

- ods in Applied Mechanics and Engineering, Vol.191, No.11–12, pp.1245-1287 (2002)
- [130] E. Mezura-Montes and C. A. Coello-Coello: “Constraint-Handling in Nature-Inspired Numerical Optimization: Past, Present and Future,” Swarm and Evolutionary Computation, Vol.1, No.4, pp.173-194 (2011)
- [131] S. L. Digabel and S. M. Wildy: “A Taxonomy of Constraints in Simulation-Based Optimization,” arXiv preprint (2015), <https://arxiv.org/abs/1505.07881> (Last accessed 1-May-2020)
- [132] 伊藤功樹, 阪本直氣, 田中清, 秋本洋平:「進化計算のための拡張ラグランジュ制約対処法の考察」, 計測自動制御学会 システム・情報部門 学術講演会 2018, SS14-04, pp.- (2018)
- [133] X.-S. He, Q. -W. Fan, M. Karamanoglu, and X. -S. Yang: “Comparison of Constraint-Handling Techniques for Metaheuristic Optimization,” Proceedings of the 19th International Conference on Computational Science, Vol.11538, pp 357-366 (2019)
- [134] N. Sakamoto and Y. Akimoto: “Adaptive Ranking Based Constraint Handling for Explicitly Constrained Black-Box Optimization,” GECCO ’19: Proceedings of the Genetic and Evolutionary Computation Conference, pp.700–708 (2019)
- [135] 相吉英太郎, 岡本卓, 安田恵一郎:「最適化手法の基礎 力学モデルによる理解と実装」, 森北出版 (2014)
- [136] 上山大地, 田村健一, 安田恵一郎:「シンプレックス法を取り入れた Differential Evolution」, 電気学会 電子・情報・システム部門誌, Vol.130, No.7, pp.1271-1272 (2010)
- [137] 小熊祐司, 古林史裕, 相吉英太郎:「Nelder-Mead 法に基づく新しい大域的最適化手法」, 平成 20 年電気学会電子・情報・システム部門大会, OS7-5, pp.457-462 (2008)
- [138] 関根嵩之, 相吉英太郎:「改良型応答局面法を用いたサポートベクターマシンの最適ハイパーパラメータ決定法」, システム制御情報学会論文誌, Vol.29, No.3, pp.105-113 (2016)
- [139] 田口玄一:「実験計画法 (上下)」, 丸善株式会社 (1976)
- [140] 中山弘隆, 荒川雅生, 佐々木理恵:「RBF ネットワークと遺伝アルゴリズムによる未知目的関数の最適化」, システム制御情報学会論文誌, Vol.13, No.3, pp.152-154 (2000)
- [141] 北山哲士, 安田恵一郎, 山崎光悦:「RBF ネットワークと Particle Swarm Optimization による統合的最適化」, 電気学会 電子・情報・システム部門誌, Vol.128, No.4, pp.636-645 (2008)

- [142] 河原林雅, 安田恵一郎:「Particle Swarm Optimization とモデリングを用いた統合的最適化」, 計測自動制御学会論文集, Vol.45, No.11, pp.1-8 (2008)
- [143] 北山哲士, 荒川雅生, 山崎光悦:「RBF ネットワークによる逐次近似最適化（サンプル関数の基礎的検討）」, 日本機械学会論文集 (C編), Vol.76, No.768, pp.98-107 (2010)
- [144] 片田喜章:「目的関数の近似に用いる RBF ネットワークの基底関数の半径に関する解析的考察」, 計測自動制御学会 システム・情報部門 学術講演会 2016, GS01-11, pp.32-36 (2016)
- [145] 松井孝太, 金森研太, 豊浦和明, 竹内一郎:「ベイズ最適化の基礎と材料工学への応用」, 日本金属学会 までりあ, Vol.58, No.1, p.12-16 (2019)
- [146] 下山翔, 野村将寛:「ベイズ最適化における代理モデルの性能比較」, 電子情報通信学会 第 22 回情報論的学習理論ワークショップ, 信学技報, Vol.119, No.89, pp.43-50 (2019)
- [147] A. Hebbal, L. Breault, M. Balesdent, E. Taibi, and N. Melab : “Efficient Global Optimization using Deep Gaussian Processes,” Proceedings of the IEEE Congress on Evolutionary Computation, pp.1-8 (2018)
- [148] 土屋耕治, 森田浩:「最適化のための効率的なサンプリングに関する考察」, 数理解析研究所講究録, Vol.1584, pp.122-131 (2008)
- [149] 下山幸治, 鄭信圭, 大林茂:「多目的最適化における Kriging 応答曲面法のためのサンプル追加指標の比較」, 進化計算学会論文誌, Vol.3, No.3, pp.173-184 (2012)
- [150] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, J. S.-Dickstein: “Deep Neural Networks as Gaussian Processes,” Proceedings of the 6th International Conference on Learning Representations, pp.1-17 (2018)
- [151] D. Zaharie: “Influence of Crossover on the Behavior of Differential Evolution Algorithms,” Applied Soft Computing, Vol.9, No.3, pp.1126-1138 (2009)
- [152] 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)
- [153] 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)
- [154] 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)

- [155] 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)
- [156] 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)
- [157] Y. Koguma and E. Aiyoshi: “A New Framework with a Stability Theory for Multipoint-Type and Stochastic Meta-Heuristic Optimization Algorithms,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E98.A, No.2, pp.700-709 (2015)
- [158] M. Kanemasa and E. Aiyoshi: “Algorithm Tuners for PSO Methods and Genetic Programming Techniques for Learning Tuning Rules,” IEEJ Transactions on Electrical and Electronic Engineering, Vol.9, No.4, pp.407-414 (2014)
- [159] X. He, X. -S. Yang, M. Karamanoglu, and Y. Zhao: “Global Convergence Analysis of the Flower Pollination Algorithm: A Discrete-Time Markov Chain Approach,” Proceedings of the International Conference on Computational Science, Vol.108, pp.1354-1363 (2017)
- [160] A. Atamna, A. Auger, and N. Hansen: “On Invariance and Linear Convergence of Evolution Strategies with Augmented Lagrangian Constraint Handling,” Theoretical Computer Science, Elsevier (2018)
- [161] Y. Akimoto, A. Auger, N. Hansen: “An ODE Method to Prove the Geometric Convergence of Adaptive Stochastic Algorithms,” arXiv preprint (2018), <https://arxiv.org/abs/1811.06703> (Last accessed 1-May-2020)
- [162] I. Rechenberg: “Evolutions Strategien,” Springer (1978)
- [163] N. Hansen and A. Auger: “Principled Design of Continuous Stochastic Search: From Theory to Practice,” Theory and Principled Methods for the Design of Metaheuristics, pp 145-180 (2013)
- [164] L. J. Eshelman and J. D. Schaffer: “Real-Coded Genetic Algorithms and Interval-Schemata,” in FOGA, pp.187-202 (1992)
- [165] K. Deb and R. B Agrawal: “Simulated binary crossover for continuous search space,” Complex Systems, Vol.9, pp.115-148 (1995)
- [166] I. Ono, H. Satoh, and S. Kobayashi: “A Real-Coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover,” Proceedings of the 7th

- International Conference on Genetic Algorithms, pp.246-253 (1997)
- [167] S. Tsutsui, M. Yamamura, and T. Higuchi: "Multi-Parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms," In GECCO, Vol.1, pp.657-664 (1999)
- [168] K. Deb, A. Anand, and D. Joshi: "A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization," Evol. Comput., Vol.10, No.4, pp.345-369 (2002)
- [169] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina: "Real-Coded Memetic Algorithms with Crossover Hill-Climbing," Evol. Comput., Vol.12, No.3, pp.273-302 (2004)
- [170] 小林重信:「実数値GAのフロンティア」, 人工知能学会論文誌, Vol.24, No.1, pp.147-162 (2009)
- [171] H. Satoh, M. Yamamura, and S. Kobayashi: "Minimal generation gap model for gas considering both exploration and exploitation," In the IIZUKA: Methodologies for the Conception, Design, and Application of Intelligent Systems, pp.494-497 (1996)
- [172] 秋本洋平, 永田裕一, 佐久間淳, 小野功, 小林重信:「実数値GAにおける生存選択モデルとしてのMGGとJGGの挙動解析」, 人工知能学会論文誌, Vol.25, No.2, pp.281-289 (2010)
- [173] 喜多一, 山村雅幸:「機能分担仮説に基づくGAの設計指針」, 計測自動制御学会 計測と制御, Vol.38, No.10, pp.612-617 (1999)
- [174] 佐久間淳, 小林重信:「確率分布推定に基づく実数値GAの新展開」, 人工知能学会誌, Vol.18, No.5, pp.479-486 (2003)
- [175] H. Kita, I. Ono and S. Kobayashi: "Theoretical Analysis of the Unimodal Normal Distribution Crossover for Realcoded Genetic Algorithms," Proceedings of ICEC'98, pp.529-534 (1998)
- [176] J. Kennedy and R. C. Eberhart : "Swarm Intelligence," Morgan Kaufmann Publishers, San Francisco (2001)
- [177] Y. Shi: "Particle Swarm Optimization," IEEE Connections, Vol.2, No.1, pp.8-13 (2004)
- [178] R. Tanabe and A. Fukunaga: "Reevaluating Exponential Crossover in Differential Evolution," In Proceedings of the Parallel Problem Solving from Nature, pp.201-210 (2014)

- [179] 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)
- [180] X. -S. Yang and X. He: "Firefly Algorithm: Recent Advances and Applications," *International Journal of Swarm Intelligence*, Vol.1, No.1, pp.36–50 (2013)
- [181] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Adaptive FA Based on Evaluation and Control of Search State for Superior Solution Set Search Problem," *IEEJ Transactions on Electrical and Electronic Engineering*, Vol.13, No.12, pp.1783-1794 (2018)
- [182] X. -S. Yang: "Nature-Inspired Algorithms and Applied Optimization," *Studies in Computational Intelligence*, Vol.744, pp.53-67 (2017)
- [183] MATLAB File Exchange, "Cuckoo Search Algorithm," <https://jp.mathworks.com/matlabcentral/fileexchange/29809-cuckoo-search-cs-algorithm> (Last accessed 1-May-2020)
- [184] 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)
- [185] H. R. Soneji and R. C. Sanghvi: "Towards the Improvement of Cuckoo Search Algorithm," *International Journal of Computer Information Systems and Industrial Management Applications*, Vol.6, pp.77-88 (2014)
- [186] 森田真英, 落合広樹, 田村健一, 安田恵一郎:「問題構造の概形の推定機構を有する多点探索型組合せ最適化手法」, 電気学会電子・情報・システム部門誌, Vol.136, No.7, pp.963-976 (2016)
- [187] R. Salomon: "Re-Evaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions: A Survey of Some Theoretical and Practical Aspects of Genetic Algorithms," *Biosystems*, Vol.39, No.3, pp.263-278 (1996)
- [188] 小野功, 山村雅幸, 喜多一:「実数値GAとその応用」, 人工知能学会論文誌, Vol.15, No.2, pp.259-266 (2000)
- [189] 阪井節子, 高濱徹行:「斜交座標系に基づく回転不变なブレンド交叉の提案」, 情報処理学会研究報告, Vol.2017-MPS-113, pp.1-6 (2017)
- [190] 大場和久, 串田淳一:「超立方体交叉手法を用いた Differential Evolution の提案」, 日本福祉大学健康科学論集, Vol.16, pp.9-17 (2013)
- [191] C. He, N. Noman, and H. Iba: "An Improved Artificial Bee Colony Algorithm with

- Non-separable Operator,” Convergence and Hybrid Information Technology, International Conference on Hybrid Information Technology 2012, Lecture Notes in Computer Science, Vol.7425, pp.203-210 (2012)
- [192] 鈴木真, 中野秀洋, 宮内新:「相対評価と多次元更新を導入した人工蜂コロニーアルゴリズム」, 電子情報通信学会 NLP 研究会, 信学技報, Vol.114, No.145, pp.27-32 (2014)
- [193] D. Mori and S. Yamaguchi: “Artificial Bee Colony Algorithm with Principal Component Analysis,” Journal of Electronics and Communications in Japan, Vol.99, No.9, pp.47-61 (2016)
- [194] G. Marsaglia: “Choosing a Point from the Surface of a Sphere,” Annals of Mathematical Statistics, Vol.43, No.2, pp.645–646 (1972)
- [195] Y. Shi and R. C. Eberhart: “Parameter Selection in Particle Swarm Optimization,” Evolutionary Programming VII: Proceedings of the EP 98, pp.591-600, Springer-Verlag (1998)
- [196] J. F. Schutte and A. A. Groenwold: “A Study of Global Optimization Using Particle Swarms,” J.Global Opt., Vol.31, pp.93-108 (2005)
- [197] P. N. Suganthan: “Particle Swarm Optimiser with Neighborhood Operator,” Proc. of the 1999 Congress on Evolutionary Computation, Vol.3 (1999)
- [198] 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)
- [199] P. Ong: “Adaptive Cuckoo Search Algorithm for Unconstrained Optimization,” The Science World Journal, Vol.2014, pp.1-8 (2014)
- [200] 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)
- [201] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber: “Natural Evolution Strategies,” Proceedings of the IEEE Congress on Evolutionary Computation, pp.3381-3387 (2008)
- [202] 下山幸治, 鄭信圭, 大林茂:「大域的多峰性関数最適化のための実数値 GA の枠組み Big Valley Explorer の提案」, 進化計算学会論文誌, Vol.4, No.1, pp.1-12 (2013)
- [203] G. Francesca, P. Pellegrini, T. Stützle, and M. Birattari: “NOFF-line and On-line Tun-

- ing: A Study on Operator Selection for a Memetic Algorithm Applied to the QAP,” Proceedings of the 11th European Conference on Evolutionary Computation in Combinatorial Optimization, pp.203-214 (2011)
- [204] A. E. Eiben, R. Hinterding, and Z. Michalewicz: “Parameter Control in Evolutionary Algorithms,” IEEE Transactions on Evolutionary Computation, Vol.3, No.2, pp.124-141 (1999)
- [205] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer: “Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems,” IEEE Transactions on Evolutionary Computation, Vol.10, No.6, pp.646-657 (2006)
- [206] A. K. Qin, V. L. Huang, and P. N. Suganthan: “Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization,” IEEE Transactions on Evolutionary Computation, Vol.13, No.2, pp.398-417 (2009)
- [207] Y. Wang, H. Li, T. Huang, and L. Li: “Differential Evolution Based on Covariance Matrix Learning and Bimodal Distribution Parameter Setting,” Journal of Applied Soft Computing, Elsevier, Vol.18, pp. 232-247 (2014)
- [208] R. Tanabe and A. S. Fukunaga: “Improving the Search Performance of SHADE Using Linear Population Size Reduction,” Proceedings of the IEEE Congress on Evolutionary Computation, pp.1658-1665 (2014)
- [209] 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)

2. 本研究に関する著者の研究業績

2.1 学術論文（査読有り）

- [210] 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 (2015.6)

- [211] 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, Vol.136, No.11, pp.1596-1609 (2016.11)
- [212] W. Kumagai and K. Yasuda: "Making Rotational Invariance of Particle Swarm Optimization Based on Correlativity," IEEJ Transactions on Electrical and Electronics Engineering, Vol.12, No.S2, pp.131-132 (2017.12)
- [213] Y. Miyake, W. Kumagai, K. Tamura, and K. Yasuda: "Search Point Ranking-Based Adaptive Cuckoo Search," IEEJ Transactions on Electrical and Electronics, Vol.13, No.7, pp.1075-1076 (2018.7)
- [214] 熊谷渉, 安田恵一郎:「回転不変性を有する適応型 Particle Swarm Optimization」, 電気学会電子・情報・システム部門誌, Vol.139, No.10, pp.1201-1214 (2019.10)
- [215] W. Kumagai, X. Zheng, K. Tamura, and K. Yasuda: "Artificial Bee Colony Algorithm with Rotational Invariance Based on Hypersphere," IEEJ Transactions on Electrical and Electronics, Vol.15, No.9 (2020.9) [in press]

2.2 国際会議（査読有り）

- [216] W. Kumagai, K. Tamura, and K. Yasuda: "Search Dynamics Analysis and Adaptive Parameter Adjustment of Cuckoo Search," in Proceedings of the 2015 IEEE International Conference on Systems, Man and Cybernetics, pp.1700-1705 (2015.10)
- [217] W. Kumagai and K. Yasuda: "Particle Swarm Optimization with Rotational Invariance using Correlativity," in Proceedings of the 2018 IEEE International Conference on Systems, Man and Cybernetics, pp.1197-1204 (2018.10)

2.3 国内学会発表（査読無し）

- [218] 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 (2014.11)
【15 件の優秀論文に選出（総論文数 350 件）】
- [219] 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 (2015.3)

- [220] 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 (2015.3)
- [221] 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 (2015.3) 【電気学会 優秀論文発表賞 (A 賞) 受賞】
- [222] W. Kumagai, K. Tamura, and K. Yasuda: “Parameter Analysis and Adaptation of Cuckoo Search,” 2015 IEEJ Conference on Electronics, Information and Systems, SS4-2, pp.1619-1620 (2015.8)
- [223] 三宅雄貴, 熊谷渉, 田村健一, 安田恵一郎:「回転不变性を有する Cuckoo Search についての基礎検討」, 計測自動制御学会 システム・情報部門 学術講演会 2017, SS11-6 pp.1-6 (2017.11)
- [224] 熊谷渉, 安田恵一郎:「相関性を用いた回転不变性を有する Particle Swarm Optimization」, 平成 30 年電気学会全国大会, 3-023, pp.28-29 (2018.3) 【電気学会 優秀論文発表賞 (A 賞) 受賞】
- [225] 熊谷渉, 安田恵一郎:「相関性を用いた回転不变性を有する Particle Swarm Optimization」, 平成 30 年電気学会電子・情報・システム部門大会, MC1-2, pp.28-29 (2018.9)
- [226] 熊谷渉, 安田恵一郎:「変換不变性を有するメタヒューリスティクスのフレームワークの基礎的検討」, 第 62 回 自動制御連合講演会, 1K1-03 (2019.11) 【優秀発表賞 受賞】
- [227] 熊谷渉, 安田恵一郎:「基底変換に基づくアフィン変換不变性を有する Cuckoo Search」, 計測自動制御学会 システム・情報部門 学術講演会 2019, GS01-01, pp.136-141 (2019.11)
- [228] 徐鄭, 熊谷渉, 田村健一, 安田恵一郎:「ロバスト性を考慮した回転不变性を有する Artificial Bee Colony Algorithm」, 計測自動制御学会 システム・情報部門 学術講演会 2019, GS01-04, pp.189-194 (2019.11)
- [229] 安田雄佑, 熊谷渉, 田村健一, 安田恵一郎:「メタヒューリスティクスのための制約対処法に関する解析とその課題」, 令和 2 年電気学会全国大会, 3-018, pp.24-25 (2020.3)

3. 本研究以外の著者の研究業績

3.1 学術論文（論文審査あり）

- [230] 松田健一, 熊谷渉, 近藤良, 増澤徹:「IPM型5軸制御セルフベアリングモータの小型化とコギングトルク特性改善に関する研究」, 日本機械学会, Vol.81, No.827, p.14-00567 (2015年7月)
- [231] 大隅竜太, 熊谷渉, 田村健一, 安田恵一郎:「単一目的最適化における優良解集合探索問題とFirefly Algorithmに基づく解法」, 電気学会 電子・情報・システム部門誌, Vol.136, No.10, pp.1497-1498 (2016.10)

3.2 国際会議（査読有り）

なし。

3.3 国内学会発表（査読無し）

- [232] 大隅竜太, 熊谷渉, 田村健一, 土屋淳一, 安田恵一郎:「優良解集合の提案とFirefly Algorithmに基づく最適化手法の提案」, 進化計算学会 進化計算シンポジウム 2016, P1-03, pp.12-20 (2016.12)

3.4 解説論文

- [233] 安田 恵一郎, 熊谷 渉:「ブラックボックス最適化とその応用」, 計測と制御, Vol.59, No.12 (2020.12) 【掲載決定】

4. 受賞歴

[234] 電気学会 優秀論文発表賞（A賞） 受賞

受賞論文：「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」

受賞日：2016年3月17日

[235] 電気学会 優秀論文発表賞（A賞） 受賞

受賞論文：「相関性を用いた回転不変性を有する Particle Swarm Optimization」

受賞日：2019年3月13日

[236] 第62回 自動制御連合講演会 優秀発表賞 受賞

受賞論文：「変換不変性を有するメタヒューリティクスのフレームワークの基礎的検討」

受賞日：2019年12月16日

A

数学的表記と定理

A.1 数学的表記

本付録では、本論文で扱う数学的表記の諸定義を行う。

(a) 集合

有限集合 A に属する元の個数を $|A|$ 、あるいは $\text{card}(A)$ と表す。

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 (A.1) で表す。

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \quad (\text{A.1})$$

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

(c) 行列

$N, M \in \mathbb{N}$ に対して、 N 行 M 列の行列 $\mathbf{A} \in \mathbb{R}^{N \times M}$ を式 (A.2) で表す。

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NM} \end{bmatrix} \quad (\text{A.2})$$

ただし、 $a_{nm} (n = 1, \dots, N; m = 1, \dots, M)$ を行列 \mathbf{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 $\mathbf{A} \in \mathbb{R}^{N \times M}$ 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、 \mathbf{A}^T と表す。
- 対称行列： $\mathbf{A} \in \mathbb{R}^{N \times N}$ を正方行列といい、 $\mathbf{A} = \mathbf{A}^T$ を満たす正方行列 \mathbf{A} を対称行列という。対称行列 $\mathbf{A} \in \mathbb{R}^{N \times N}$ について、二次形式 $\mathbf{z}^T \mathbf{A} \mathbf{z} > 0$ ($\forall \mathbf{z} \in \mathbb{R}^N, \mathbf{z} \neq \mathbf{0}$) が成り立つとき、 \mathbf{A} を正定値行列といい、 $\mathbf{A} \succ 0$ 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 $\mathbf{A} \in \mathbb{R}^{N \times N}$ 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、 $\mathbf{I} \in \mathbb{R}^{N \times N}$ と表す。
- 正則行列：正方行列 $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times N}$ 、単位行列 $\mathbf{I} \in \mathbb{R}^{N \times N}$ について、式 (A.3) の関係を満たす行列 \mathbf{B} が存在するとき、 \mathbf{A} を正則行列という。

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I} \quad (\text{A.3})$$

正則行列 \mathbf{A} について、式 (A.3) の関係を満たす行列 \mathbf{B} が存在するとき、 \mathbf{B} を \mathbf{A} の逆行列といい、 $\mathbf{B} = \mathbf{A}^{-1}$ と表す。一般的に、正方行列 \mathbf{A} について、「逆行列 \mathbf{A}^{-1} が

存在する（ \mathbf{A} の行列式が 0 でない）こと」と、「 \mathbf{A} が正則行列であること」は同値である。行列 $\mathbf{A} \in \mathbb{R}^{N \times N}$ が正定値行列ならば、 \mathbf{A} は正則である。

ただし、「逆行列の存在」の問題においては、工学的の問題において行列が正則であることは保証されないことが多い、正則行列でなければ、逆行列の計算ができない。その場合は、Moore-Penrose の擬似逆行列（一般逆行列）が使用され、定義 A.1 によって定義される。

定義 A.1 ある行列 \mathbf{A} について、下記の 4 条件を満たす \mathbf{A}^\dagger を Moore-Penrose の擬似逆行列という。

- (1) $\mathbf{A}\mathbf{A}^\dagger\mathbf{A} = \mathbf{A}$ (疑似逆行列の必要十分条件)
- (2) $\mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger = \mathbf{A}^\dagger$ (反射型疑似逆行列)
- (3) $(\mathbf{A}^\dagger\mathbf{A})^T = \mathbf{A}^\dagger\mathbf{A}$ (最小ノルム型疑似逆行列)
- (4) $(\mathbf{A}\mathbf{A}^\dagger)^T = \mathbf{A}\mathbf{A}^\dagger$ (最小二乗型疑似逆行列)

定義 A.1 を満たす式として、式 (A.4) で表される。

$$\mathbf{A}^\dagger = \mathbf{C}^T (\mathbf{C}\mathbf{C}^T)^{-1} (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \quad (\text{A.4})$$

ただし、 \mathbf{B}, \mathbf{C} は $\mathbf{A} = \mathbf{BC}$ と階数因数分解したときに得られる行列である。また、他の定義 A.1 を満たす式として、特異値分解を用いた方法があり、式 (A.5) で表される。

$$\mathbf{A}^\dagger = \mathbf{V} \Delta \mathbf{U}^T \quad (\text{A.5})$$

ただし、 \mathbf{U}, \mathbf{V} は $\mathbf{A} = \mathbf{U} \Delta \mathbf{V}^T$ と特異値分解したときに得られる行列である。 $\Delta \in \mathbb{R}^{N \times M}$ は式 (A.6) で表される。

$$\Delta = \begin{bmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (\text{A.6})$$

ただし、 $\mathbf{A} \in \mathbb{R}^{N \times M}$, $\mathbf{U} \in \mathbb{R}^{N \times N}$, $\mathbf{V} \in \mathbb{R}^{M \times M}$, $\Lambda_r = \text{diag}[\sigma_1, \sigma_2, \dots, \sigma_r]$, σ_n は特異値, $r = \text{rank}(\mathbf{A})$ である。

(d) 関数

ある f が始域（定義域） \mathbf{A} から終域 \mathbf{B} への写像であるとき, $f : \mathbf{A} \rightarrow \mathbf{B}$ 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 $\mathbf{x} \in \mathbb{R}^N$ 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, $f(\mathbf{x})$ 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。

アフィン変換を $T_s : \mathbf{x} \mapsto \mathbf{C}\mathbf{x} - \mathbf{t}$ と表す。ただし, $\forall \mathbf{t} \in \mathbb{R}^N$, $\mathbf{C} \in \mathbb{R}^{N \times N}$ は正則である。

(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 A.1 を満たす関数 d を距離関数, 組 (X, d) を距離空間という。

公理 A.1 (距離の公理) $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in X$ について,

- (1) $d(\mathbf{x}, \mathbf{y}) \geq 0$ (正定値性)
- (2) $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$ (非退化性)
- (3) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ (対称性)
- (4) $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$ (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 $d(\mathbf{x}, \mathbf{y}; p)$ ($\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$; $p \in \mathbb{R}$) を L^p -ノルム (p -ノルム) といい, 式 (A.7) で定義される。

$$d(\mathbf{x}, \mathbf{y}; p) = \left(\sum_{n=1}^N |x_n - y_n|^p \right)^{1/p} \quad (\text{A.7})$$

式 (A.7) の L^p -ノルムにおいて $p = 1$ のとき, $d(\mathbf{x}, \mathbf{y}; 1)$ は Manhattan 距離と表す。 $p = 2$ のとき, $d(\mathbf{x}, \mathbf{y}; 2)$ は Euclidean 距離といい, $d_E(\mathbf{x}, \mathbf{y})$ と表す。 $p \rightarrow \infty$ のとき, $d(\mathbf{x}, \mathbf{y}; \infty)$ は Chebyshev 距離 (L^∞ 距離) という。あるいは, L^p -ノルムを $\|\cdot\|_p$ と表記する。

距離空間 (\mathbf{X}, d) のある点 $x \in \mathbf{X}$ 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 (A.8) で表す。

$$B(x; \varepsilon) = \{y \in \mathbf{X} \mid d(x, y) < \varepsilon\} \subseteq \mathbf{X} \quad (\text{A.8})$$

式 (A.8) より, 距離空間 (\mathbb{R}^N, d_E) において, ある点 $\mathbf{x}^* \in \mathbb{R}^N$ に対する ε -近傍 $B(\mathbf{x}^*; \varepsilon)$ は式 (A.9) で表すことができる。

$$B(\mathbf{x}^*; \varepsilon) = \{\mathbf{x} \in \mathbb{R}^N \mid \|\mathbf{x}^* - \mathbf{x}\| < \varepsilon\} \quad (\text{A.9})$$

(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 $f(\mathbf{x})(\mathbf{x} \in \mathbb{R}^N)$ 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 (A.10) で表す。

$$\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_N} \right)^T \quad (\text{A.10})$$

ナブラ ∇ を多変数関数 $f(\mathbf{x})$ に作用させた $\nabla f(\mathbf{x}) \in \mathbb{R}^N$ を勾配ベクトルといい, 各変数に対する一階偏微分 $\partial f(\mathbf{x})/\partial x_n = f_{x_n}(n = 1, \dots, N)$ を要素とする。勾配ベクトル $\nabla f(\mathbf{x})$ を式 (A.11) で表す。

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_N} \right)^T \quad (\text{A.11})$$

各変数に対する二階偏微分 $h_{nm} = \partial^2 f(\mathbf{x})/\partial x_n \partial x_m = f_{x_n x_m}(n, m = 1, \dots, N)$ を要素とする行列 $\mathbf{H}(f(\mathbf{x})) \in \mathbb{R}^{N \times N}$ を Hesse 行列といい, 式 (A.12) で表す。

$$\mathbf{H}(f(\mathbf{x})) = \begin{pmatrix} f_{x_1 x_1} & \cdots & f_{x_N x_1} \\ \vdots & \ddots & \vdots \\ f_{x_1 x_N} & \cdots & f_{x_N x_N} \end{pmatrix} \quad (\text{A.12})$$

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

任意の \mathbf{x} における多変数関数 $f(\mathbf{x})$ のベクトル $\mathbf{v} \in \mathbb{R}^N$ に対する偏微分 $\partial f(\mathbf{x})/\partial \mathbf{v}$ を方向微分といい, 式 (A.13) で定義される。

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{v}} = \lim_{h \rightarrow 0} \frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h} \quad (\text{A.13})$$

式 (A.13) の方向微分について, f が \mathbf{x} において全微分可能ならば式 (A.14) の定理が成立する。

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{v}} = \nabla f(\mathbf{x})^T \mathbf{v} \quad (\text{A.14})$$

(g) 確率分布

変数 $x \in \mathbb{R}$ あるいは, $\mathbf{x} \in \mathbb{R}^N$ 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 (A.15), 式 (A.16), 式 (A.17), 式 (A.18), 式 (A.19), 式 (A.20) で定義する。

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

$$f_{\mathcal{U}_{\mathbb{R}}}(x; a, b) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}, x \in \mathbb{R} \quad (\text{A.15})$$

式 (A.15) において, $a, b \in \mathbb{R}$ は分布の下限値, 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。

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

$$f_{\mathcal{U}_{\mathbb{R}}}(\mathbf{x}; \mathbf{a}, \mathbf{b}) = \prod_{n=1}^N f_{\mathcal{U}_{\mathbb{R}}}(x_n; a_n, b_n), \quad \mathbf{x} \in \mathbb{R}^N \quad (\text{A.16})$$

式 (A.16) において, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ は分布の下限値ベクトル, 上限値ベクトルである。特に, $\mathbf{a} = \mathbf{0}, \mathbf{b} = \mathbf{1}$ の場合, 多変量実数値標準一様分布という。

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

$$f_{\mathcal{U}_{\mathbb{Z}}}(x; a, b) = \begin{cases} \frac{1}{b-a+1}, & x \in \{a, \dots, b\} \\ 0, & \text{otherwise} \end{cases}, x \in \mathbb{Z} \quad (\text{A.17})$$

式 (A.17)において, $a, b \in \mathbb{Z}$ は分布の下限値, 上限値である。

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

$$f_{\mathcal{N}}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (\text{A.18})$$

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。

- 多変量正規分布 $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$f_{\mathcal{N}}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) \quad (\text{A.19})$$

ただし, $\boldsymbol{\mu} \in \mathbb{R}^N$ は平均ベクトル, 共分散行列 $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times N}$ は共分散行列である。特に, $\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \mathbf{I}$ の場合, 多変量標準正規分布という。

- (狭義) Lévy 分布 $\mathcal{L}(\mu, c)$:

$$f_{\mathcal{L}}(x; \mu, c) = \begin{cases} \sqrt{\frac{c}{2\pi}} \frac{\exp(-c/(2(x-\mu)))}{(x-\mu)^{3/2}}, & \mu \leq x \\ 0, & \text{otherwise} \end{cases} \quad (\text{A.20})$$

ただし, μ は位置パラメータ, c は尺度パラメータである。

- Cauchy 分布 $\mathcal{C}(\mu, \gamma)$:

$$f_{\mathcal{C}}(x; \mu, \gamma) = \frac{1}{\pi} \frac{\gamma}{(x-\mu)^2 + \gamma^2} \quad (\text{A.21})$$

ただし, μ は位置パラメータ, γ は尺度パラメータである。

- 安定分布 \mathcal{S} :

$$f_{\mathcal{S}}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \varphi(t) \exp(-ixt) dt \quad (\text{A.22})$$

ただし, i は虚数単位, $\varphi(t)$ は特性関数であり, 解析的に記述することができない。特性関数 $\varphi(t; \alpha, \beta, \gamma, \delta)$ は式 (A.23), 式 (A.24) で表される。

$$\varphi(z; \alpha, \beta, \gamma, \delta) = \exp(i\delta z - \gamma \|z\|^\alpha(1 + i\beta \operatorname{sgn}(z)\omega(z; \alpha))) \quad (\text{A.23})$$

$$\omega(z; \alpha) = \begin{cases} \tan(\frac{\pi\alpha}{2}), & \alpha \neq 1 \\ \frac{2}{\pi} \ln \|z\|, & \text{otherwise} \end{cases} \quad (\text{A.24})$$

ただし, $\alpha \in (0, 2]$ は特性指数, $\beta \in [-1, 1]$ は歪度指数, $\gamma > 0$ は尺度パラメータ, $\delta \in \mathbb{R}$ は位置パラメータ, z は複素数, sgn は符号関数である。なお, $\alpha = 2$ のとき, 式 (A.18) の正規分布に一致し, $\alpha = 0.5, \beta = 1$ のとき, 式 (A.20) の（狭義）Lévy 分布に一致し, $\alpha = 1, \beta = 0$ のとき, 式 (A.21) の Cauchy 分布に一致する。 α は付録 B の Mantegna's Algorithm における β に対応しており, $\alpha < 2$ のとき, 安定分布はヘヴィーテイル型の分布となる。

(h) 共分散行列と平均ベクトル

解集団 $\mathcal{X} = \{\mathbf{x}^i | i = 1, \dots, m\}$ の平均ベクトル $\boldsymbol{\mu}(\mathcal{X})$ と分散共分散行列（単に共分散行列と呼ぶ） $\boldsymbol{\Sigma}(\mathcal{X})$ は, 式 (A.25), 式 (A.26) で表される。

$$\boldsymbol{\mu}(\mathcal{X}) = \frac{1}{m} \sum_{\mathbf{x}^i \in \mathcal{X}} \mathbf{x}^i \quad (\text{A.25})$$

$$\boldsymbol{\Sigma}(\mathcal{X}) = \frac{1}{m} \sum_{\mathbf{x}^i \in \mathcal{X}} (\boldsymbol{\mu}(\mathcal{X}) - \mathbf{x}^i)(\boldsymbol{\mu}(\mathcal{X}) - \mathbf{x}^i)^T \quad (\text{A.26})$$

また, ある群 \mathcal{X} に属する点 \mathbf{x} について, 群に対する Mahalanobis 距離 $d_M(\mathbf{x}, \mathbf{y})$ は式 (A.27) で表される。

$$d_M = (\mathbf{x} - \boldsymbol{\mu}(\mathcal{X}))^T \boldsymbol{\Sigma}^{-1}(\mathcal{X}) (\mathbf{x} - \boldsymbol{\mu}(\mathcal{X})) \quad (\text{A.27})$$

ただし, $\boldsymbol{\Sigma}(\mathcal{X})$ は正則とする。Mahalanobis 距離は, $\boldsymbol{\Sigma}(\mathcal{X})$ に属する \mathbf{x}, \mathbf{y} の二点間距離 $d_M(\mathbf{x}, \mathbf{y})$ としても定義でき, その場合は式 (A.28) で表される。

$$d_M(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \boldsymbol{\Sigma}^{-1}(\mathcal{X}) (\mathbf{x} - \mathbf{y}) \quad (\text{A.28})$$

A.2 線形代数に関する諸概念・定理

(a) 特異値分解

特異値分解 (Singular Value Decomposition : SVD) は、特異値分解定理に基づき、行列を分解する方法の一つ。SVDにより、実行列 $\mathbf{A} \in \mathbb{R}^{N \times M}$ を分解すると、式 (A.29) で表される。

$$\mathbf{A} = \mathbf{U} \Delta \mathbf{V}^T \quad (\text{A.29})$$

ただし、 $\mathbf{U} \in \mathbb{R}^{N \times N}$, $\mathbf{V} \in \mathbb{R}^{M \times M}$, $r = \text{rank}(\mathbf{A})$ である。 $\Delta \in \mathbb{R}^{N \times M}$ は式 (A.30) で表される。

$$\Delta = \begin{bmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (\text{A.30})$$

ここで、 $\Lambda_r = \text{diag}[\sigma_1, \sigma_2, \dots, \sigma_r]$ であり、 σ_n は特異値である。

行列 \mathbf{A} が正方行列の場合、特異値は固有値となり、SVD は固有値分解 (Eigen Value Decomposition : EVD) となる。EVD により、正方行列 $\mathbf{A} \in \mathbb{R}^{N \times N}$ を分解すると、式 (A.31) で表される。

$$\mathbf{A} = \mathbf{P} \Delta \mathbf{P}^T \quad (\text{A.31})$$

ただし、 $\mathbf{P} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]$, $\Lambda = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_N]$, $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N$ は固有ベクトル, $\lambda_1, \lambda_2, \dots, \lambda_N$ は固有値である。よって、EVD によって行列を分解する操作は、固有ベクトルと固有値を求めるうことになる。

(b) 解空間の回転行列

N 次元の解空間を θ deg 回転させる行列 $\mathbf{B}(\theta, N)$ は式 (A.32) で表される。

$$\mathbf{B}(\theta, N) = \prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{B}^{i,j}(\theta) \quad (\text{A.32})$$

ただし、 $\mathbf{B}^{i,j}(\theta)$ は $r_{q,\ell}^{i,j}(\theta)$ ($q, \ell = 1, 2, \dots, N$) を各要素とする行列であり、 $r_{q,\ell}^{i,j}(\theta)$ は式 (A.33) で表される。

$$r_{q,\ell}^{i,j}(\theta) = \begin{cases} \cos \theta, & q = i \wedge \ell = i \\ -\sin \theta, & q = i \wedge \ell = j \\ \sin \theta, & q = j \wedge \ell = i \\ \cos \theta, & q = j \wedge \ell = j \\ 1, & q = \ell \neq i \wedge q = \ell \neq j \\ 0, & \text{else} \end{cases} \quad (\text{A.33})$$

(c) 共分散行列の基底変換行列

共分散行列 $\Sigma(\mathcal{Z})$ の基底変換行列 $\mathbf{P} = [\mathbf{a}_1, \dots, \mathbf{a}_N]$ について、補助定理 A.1 が成り立つ。

補助定理 A.1 回転変換・相似変換・平行移動の合成変換 T_s に対して、共分散行列 $\Sigma(\mathcal{Z})$ の基底変換行列 \mathbf{P} は式 (A.34) となる。

$$\mathbf{P}' = \mathbf{C}\mathbf{P} \quad (\text{A.34})$$

証明. サンプル解集団 \mathcal{Z} の共分散行列 Σ と平均ベクトル μ は

$$\Sigma = \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} (\mathbf{z}^i - \boldsymbol{\mu})(\mathbf{z}^i - \boldsymbol{\mu})^T, \quad \boldsymbol{\mu} = \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} \mathbf{z}^i$$

で表される。ただし、 $\mathbf{z}^i \in \mathcal{Z}$ は解集団 \mathcal{Z} に属するサンプル点である。サンプル点 \mathbf{z}^i は解空間に属する解なので、変換 T_s を与えた後、 $\mathbf{z}^{i'} = \delta \mathbf{C} \mathbf{z}^i - \mathbf{t}$ 、 $\mathcal{Z}' = \{\mathbf{z}^{i'} \mid i = 1, \dots, |\mathcal{Z}|\}$

となる。同様に、 $\boldsymbol{\mu}'$ と Σ' は

$$\begin{aligned}\boldsymbol{\mu}' &= \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} \mathbf{z}^{i\prime} \\ &= \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} (\delta \mathbf{C} \mathbf{z}^i - \mathbf{t}) \\ &= \delta \mathbf{C} \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} \mathbf{z}^i - \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} \mathbf{t} \\ &= \delta \mathbf{C} \boldsymbol{\mu} - \mathbf{t}\end{aligned}$$

$$\begin{aligned}\Sigma' &= \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} (\mathbf{z}^{i\prime} - \boldsymbol{\mu}') (\mathbf{z}^{i\prime} - \boldsymbol{\mu}')^\text{T} \\ &= \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} ((\delta \mathbf{C} \mathbf{z}^i - \mathbf{t}) - (\delta \mathbf{C} \boldsymbol{\mu} - \mathbf{t})) ((\delta \mathbf{C} \mathbf{z}^i - \mathbf{t}) - (\delta \mathbf{C} \boldsymbol{\mu} - \mathbf{t}))^\text{T} \\ &= \delta^2 \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} \mathbf{C} (\mathbf{z}^i - \boldsymbol{\mu}) (\mathbf{C} (\mathbf{z}^i - \boldsymbol{\mu}))^\text{T} \\ &= \delta^2 \mathbf{C} \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} (\mathbf{z}^i - \boldsymbol{\mu}) (\mathbf{z}^i - \boldsymbol{\mu})^\text{T} \mathbf{C}^\text{T} \\ &= \delta^2 \mathbf{C} \Sigma \mathbf{C}^\text{T}\end{aligned}$$

となる。

ここで、共分散行列 Σ の固有値 λ_n と固有ベクトル \mathbf{a}_n の組 $\{(\lambda_1, \mathbf{a}_1), (\lambda_2, \mathbf{a}_2), \dots, (\lambda_N, \mathbf{a}_N)\}$ ($n = 1, \dots, N$) について、式 (A.35) の固有方程式を満たす。

$$(\Sigma - \lambda_n \mathbf{I}) \mathbf{a}_n = \mathbf{0} \quad (\text{A.35})$$

ただし、 $\mathbf{I} \in \mathbb{R}^{N \times N}$ は単位行列、 $\mathbf{0} \in \mathbb{R}^N$ は零ベクトルである。同様に、変換 T_s を与えた後も、共分散行列 Σ' の固有値 λ'_n と固有ベクトル \mathbf{a}'_n の組 $\{(\lambda'_1, \mathbf{a}'_1), (\lambda'_2, \mathbf{a}'_2), \dots, (\lambda'_N, \mathbf{a}'_N)\}$ について、式 (A.36) の固有方程式を満たす。

$$(\Sigma' - \lambda'_n \mathbf{I}) \mathbf{a}'_n = \mathbf{0} \quad (\text{A.36})$$

式 (A.36) の固有方程式に Σ' を代入すると,

$$\begin{aligned}
 & (\delta^2 \mathbf{C} \Sigma \mathbf{C}^T - \lambda'_n \mathbf{I}) \mathbf{a}'_n = \mathbf{0} \\
 \Leftrightarrow & \left(\mathbf{C} \Sigma \mathbf{C}^T - \frac{\lambda'_n}{\delta^2} \mathbf{I} \right) \mathbf{a}'_n = \mathbf{0} \\
 \Leftrightarrow & \left(\mathbf{C}^T \mathbf{C} \Sigma \mathbf{C}^T - \frac{\lambda'_n}{\delta^2} \mathbf{C}^T \right) \mathbf{a}'_n = \mathbf{0} \\
 \Leftrightarrow & \left(\Sigma - \frac{\lambda'_n}{\delta^2} \mathbf{I} \right) \mathbf{C}^T \mathbf{a}'_n = \mathbf{0}
 \end{aligned} \tag{A.37}$$

式 (A.34) と式 (A.37) を恒等的に比較すると,

$$\begin{aligned}
 \lambda_n &= \frac{\lambda'_n}{\delta^2}, \quad \mathbf{a}_n = \mathbf{C}^T \mathbf{a}'_n \\
 \Leftrightarrow \lambda'_n &= \delta^2 \lambda_n, \quad \mathbf{a}'_n = \mathbf{C} \mathbf{a}_n
 \end{aligned}$$

が得られる。

よって, 変換 T_s を与えた後, 基底変換行列 \mathbf{P} は

$$\begin{aligned}
 \mathbf{P}' &= [\mathbf{a}'_1, \mathbf{a}'_2, \dots, \mathbf{a}'_N] \\
 &= [\mathbf{C} \mathbf{a}_1, \mathbf{C} \mathbf{a}_2, \dots, \mathbf{C} \mathbf{a}_N] \\
 &= \mathbf{C} [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N] \\
 &= \mathbf{C} \mathbf{P}
 \end{aligned}$$

となる。以上から, 題意が示された。 \square

(d) 共分散行列の非対称平方根行列

正方行列 $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times N}$ について, $\mathbf{A} = \mathbf{B}^2$ が成立するとき, \mathbf{B} は \mathbf{A} の平方根行列という。行列 \mathbf{A}, \mathbf{B} について, $\mathbf{A} = \mathbf{B} \mathbf{B}^T$ が成立するとき, \mathbf{B} は \mathbf{A} の非対称平方根行列という。共分散行列 $\Sigma(\mathcal{X})$ の非対称平方根について, 補助定理 A.2, 補助定理 A.3 が成り立つ。

補助定理 A.2 共分散行列の非対称平方根行列 $\Sigma_*^{1/2}$ は式 (A.38) と表される。

$$\Sigma_*^{1/2} = \mathbf{P} \boldsymbol{\Lambda}^{1/2} \quad (\text{A.38})$$

ただし、 $\mathbf{P} = [\mathbf{a}_1, \dots, \mathbf{a}_N]$ は基底変換行列、 $\boldsymbol{\Lambda} = \text{diag}[\lambda_1, \dots, \lambda_N]$ は固有値行列、 $\{(\mathbf{a}_1, \lambda_1), (\mathbf{a}_2, \lambda_2), \dots, (\mathbf{a}_N, \lambda_N)\} (\lambda_1 > \lambda_2 > \dots > \lambda_N)$ は $\Sigma(\mathcal{X})$ の固有ベクトルと固有値の組である。

証明. 共分散行列 $\Sigma(\mathcal{X})$ の非対称平方根の定義は、式 (A.39) を満たす $\Sigma_*^{1/2}$ である。

$$\Sigma(\mathcal{X}) = \Sigma_*^{1/2} (\Sigma_*^{1/2})^T \quad (\text{A.39})$$

また、共分散行列 $\Sigma(\mathcal{X})$ は実対称行列であるため、 $\Sigma(\mathcal{X}) = \mathbf{P} \boldsymbol{\Lambda} \mathbf{P}^T$ と対角化できる。ただし、 \mathbf{P} は正規直交行列 ($\mathbf{P}^{-1} = \mathbf{P}^T$) である。これを変形すると、式 (A.40) となる。

$$\begin{aligned} \Sigma(\mathcal{X}) &= \mathbf{P} \boldsymbol{\Lambda}^{1/2} \boldsymbol{\Lambda}^{1/2} \mathbf{P}^T \\ &= \mathbf{P} \boldsymbol{\Lambda}^{1/2} (\boldsymbol{\Lambda}^{1/2})^T \mathbf{P}^T \\ &= \mathbf{P} \boldsymbol{\Lambda}^{1/2} (\mathbf{P} \boldsymbol{\Lambda}^{1/2})^T \end{aligned} \quad (\text{A.40})$$

ただし、計算過程で $\boldsymbol{\Lambda}$ が対角行列 ($\boldsymbol{\Lambda}^T = \boldsymbol{\Lambda}$) であることを用いた。よって、式 (A.39) と式 (A.40) を恒等的に比較すると、 $\Sigma_*^{1/2} = \mathbf{P} \boldsymbol{\Lambda}^{1/2}$ が得られる。以上から、題意が示された。 \square

補助定理 A.3 アフィン変換 T_s に対して、共分散行列の非対称平方根は式 (A.41) となる。

$$\Sigma_*^{1/2'} = \mathbf{C} \Sigma_*^{1/2} \quad (\text{A.41})$$

証明. アフィン変換 T_s によって影響を受ける量 h を h' と表す。解集団 \mathcal{X} に属する \mathbf{x} は、線形変換 T_s に対して $\mathbf{x}' = \mathbf{C}\mathbf{x}$ となる。このとき、共分散行列 $\Sigma(\mathcal{X})$ は $\Sigma'(\mathcal{X}) = \mathbf{C}\Sigma(\mathcal{X})\mathbf{C}^T$

となる。また、 T_s が平行移動である場合、 $\mathbf{x} \in \mathcal{X}$ は $\mathbf{x}' = \mathbf{x} - \mathbf{t}$ 、 $\Sigma(\mathcal{X})$ は $\Sigma'(\mathcal{X}) = \Sigma(\mathcal{X})$ となる。このため、 T_s がアフィン変換である場合も同様に $\Sigma'(\mathcal{X}) = \mathbf{C}\Sigma(\mathcal{X})\mathbf{C}^T$ が成立する。これを式 (A.39) を用いて変形すると、

$$\begin{aligned}\Sigma' &= \mathbf{C}\Sigma_*^{1/2}(\Sigma_*^{1/2})^T\mathbf{C}^T \\ &= \mathbf{C}\Sigma_*^{1/2}(\mathbf{C}\Sigma_*^{1/2})^T \\ &= \Sigma_*^{1/2'}(\Sigma_*^{1/2'})^T\end{aligned}$$

となる。よって、 $\Sigma_*^{1/2'} = \mathbf{C}\Sigma_*^{1/2}$ が得られる。以上から、題意が示された。 \square

A.3 確率統計に関する諸概念・定理

(a) 多変量確率分布

確率変数ベクトル $\mathbf{x} \in \mathbb{R}^N$ の各要素 x_n が单変量確率分布 \mathcal{P}_n に従うとき、 \mathbf{x} は同時確率分布 \mathcal{P} に従う。ただし、同時確率分布（多変量確率分布） \mathcal{P} は、各次元の单変量確率分布 \mathcal{P}_n の条件付き確率分布で表される。さらに、確率変数ベクトルの各要素である確率変数 x_n が互いに独立しているとき、同時確率分布は单変量確率分布の重ね合わせ $\mathcal{P} = \prod_n \mathcal{P}_n$ で表される。

(b) 主成分分析

主成分分析 (Principal Component Analysis : PCA) は、多変量解析の一つで、相関のある多数の変数から相関のない少数で全体の散らばりを最も良く表す変数（主成分）を合成する。データ空間の次元を削減するために使用される。主成分は $t_n = \mathbf{a}_n^T \mathbf{x}$ で表され、最も分散が大きい主成分を第 1 主成分とし、第 1 主成分軸 \mathbf{a}_1 を得る。 \mathbf{a}_1 に直交しながら分散が大きい主成分を第 2 主成分とする、という操作を繰り替えすることで、第 N 主成分までを得る。この操作は、実質的に、対象データの共分散行列を固有値分解することで固有値と固有ベクトルを導出し、その固有値と固有ベクトルから、主成分への基底変換を求めることと同じであり、 n 番目に大きい固有値 λ_n （散らばり）を持つ固有ベクトルが第 n 主成分軸ベクトル \mathbf{a}_n に対応している。第 n 主成分の寄与率 c_n は、第 n 固有値 λ_n を用いて、

$c_n = \lambda_n / \sum_i^N \lambda_i$ と表せ, 累積寄与率 cc_n は, $cc_n = \sum_i^n \lambda_i / \sum_i^N \lambda_i$ と表せる。次元削減として使用する場合は, 寄与率が低い主成分を棄却し, 他の主成分だけを用いることで, データの情報を可能な限り落とさず, より低い次元の空間で表現することが可能である。

PCA では共分散行列を対角行列にする変換を求めるが, これは信号処理の分野では, データの無相関化・白色化とも呼ばれる。また, 各軸が元の標準基底の固有値 $\sqrt{\lambda_n}$ 倍されている超楕円体でデータをフィッティングする操作という解釈でもできる。この変換は, 補助定理 A.2 で示したように, $\Sigma_*^{1/2} = P\Lambda^{1/2}$ をベクトルに作用する操作そのものである。

B

Mantegna's Algorithm

Mantegna's Algorithm (MA) は左右対称な安定分布に従う乱数を生成するアルゴリズムである [2, 23]。MA は式 (B.1) を用いて乱数 $\mathbf{L} = [L_1, L_2, \dots, L_N]^T$ を生成する。

$$L_n = \frac{u_n}{|v_n|^{1/\beta}} \quad (\text{B.1})$$

ただし、 $\beta > 0$ は安定分布の形状を決定するパラメータ、 $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ である。 σ^2 は式 (B.2) で表される。

$$\sigma^2 = \left(\frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma((1 + \beta)/2) 2^{(\beta-1)/2} \beta} \right)^{1/\beta} \quad (\text{B.2})$$

ただし、 $\Gamma(\cdot)$ は Euler のガンマ関数である。安定分布は正規分布や Cauchy 分布を包含した分布であり、 $\beta = 2$ のとき正規分布と一致し、 $\beta = 1$ のとき Cauchy 分布と一致する。特に、 $\beta \in [0.3, 1.99]$ のとき、裾が重いヘヴィテール型の左右対称の分布となるため、それに基づく乱数は基本的に小さな値が発生するが、まれに比較的大きな値も発生する [23]。最適化においてヘヴィテール型の分布に従う乱数を使用することは、広範囲の探索に寄与すると言われている [2]。

C

本論文で用いたベンチ マーク関数

本論文の数値実験で用いた9種類のベンチマーク関数 $f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$ を示す。全ての関数は単峰性／多峰性（多峰性は、さらに大域的単峰性／大域的多峰性）、変数分離可能性（変数間独立性）／変数間依存性、良スケール性／悪スケール性によって分類できる。 x^* は大域的最適解である。

また、表 C.1 にベンチマーク関数の性質を記載する。

表 C.1：ベンチマーク関数の一覧と性質

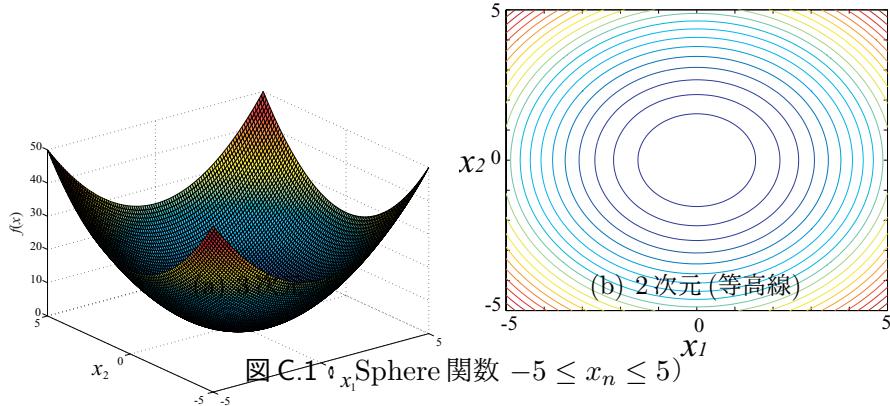
関数名	多峰性	大域的構造	変数分離可能性	スケール性	初期配置領域 S_0
f_1 : Sphere	单峰	—	可能	良スケール	$[-5, 5]^N$
f_2 : Rosenbrock	多峰	大域的单峰性	不可能	良スケール	$[-2, 2]^N$
f_3 : 2^N minima	多峰	大域的多峰性	可能	良スケール	$[-5, 5]^N$
f_4 : Rastrigin	多峰	大域的多峰性	可能	良スケール	$[-5, 5]^N$
f_5 : Schwefel	单峰	—	不可能	悪スケール	$[-5, 5]^N$
f_6 : Lévy	多峰	大域的多峰性	不可能	良スケール	$[-5, 5]^N$
f_7 : Ackley	多峰	大域的单峰性	可能	良スケール	$[-5, 5]^N$
f_8 : Griewank	多峰	大域的多峰性	不可能	良スケール	$[-50, 50]^N$
f_9 : Alpine	多峰	大域的单峰性	可能	良スケール	$[-10, 10]^N$

1. Sphere 関数

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

$$\mathbf{x}^o = \mathbf{0}, \quad f_1(\mathbf{x}^o) = 0$$

Sphere 関数は単峰性・変数分離可能・良スケール性を有する最もシンプルな関数である。さらに、大域的最適解に関して対称性がある上に、いわゆる 2 次形式標準形の関数である。

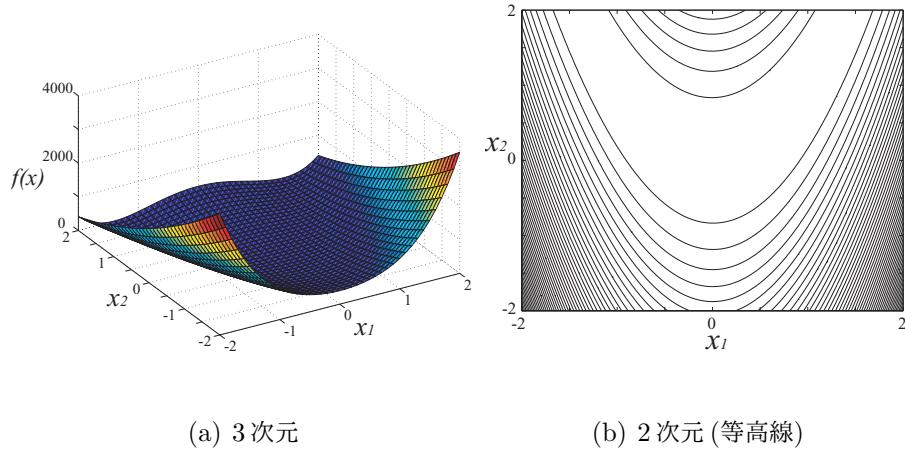


2. Rosenbrock's Saddle 関数

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

$$\mathbf{x}^o = \mathbf{1}, \quad f_2(\mathbf{x}^o) = 0$$

Rosenbrock's Saddle 関数は多峰性・大域的単峰性・変数分離不可能・良スケール性を有する関数で、変数間依存性の関数として代表的である。さらに、 -1×1 の位置に局所的最適解がある。 \sum 内の第 1 項 $100(x_n^2 - x_{n+1})^2$ により、第 2 項の二次関数に対して $x_n^2 = x_{n+1}$ の制約が課され、曲線 $C(t) = [t, t^2, \dots, t^{2^{N-1}}]$ に沿った稜線が作られるため、等高線上では緩い谷が非線型性状に現れる。

図 C.2 : Rosenbrock's Saddle 関数 $-2 \leq x_n \leq 2$

3. 2^N minima 関数

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

$$\mathbf{x}^o \approx -2.9 \times \mathbf{1}, \quad f_3(\mathbf{x}^o) \approx -78N$$

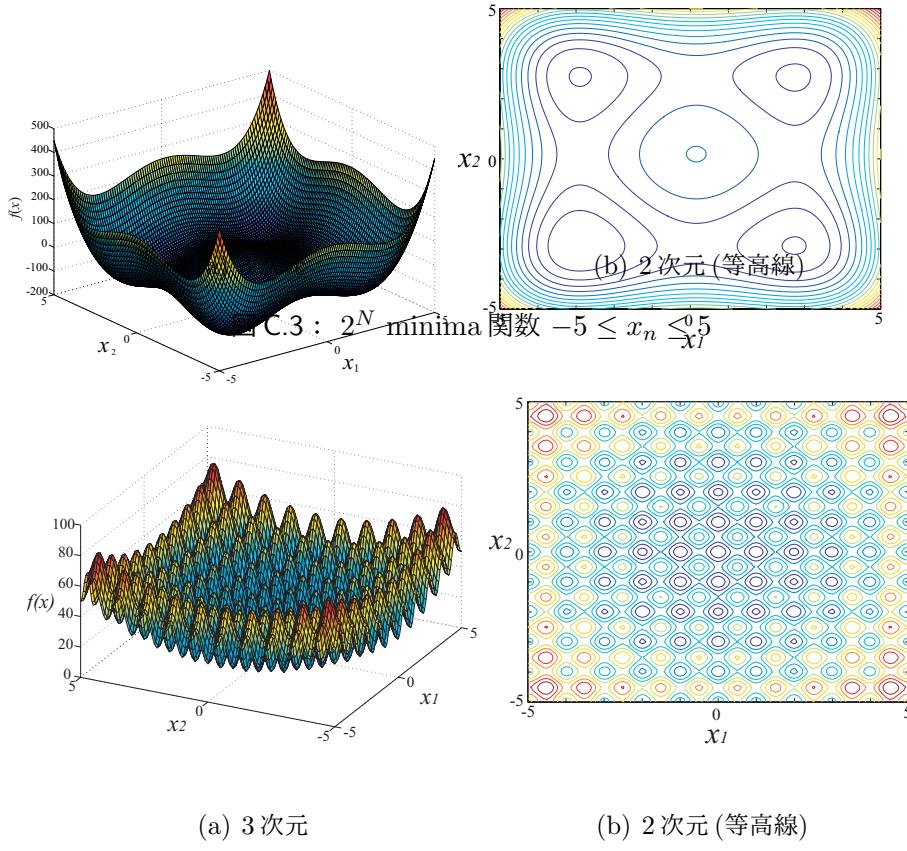
2^N minima 関数は多峰性・大域的多峰性・変数分離可能・良スケール性を有する関数で、多峰性の関数として代表的である。さらに、名称の通り、 2^N 個の局所的最適解が存在するため、次元数 N が増えるにつれて、指数的に個数が増加する。各軸・原点に関して、大域的最適解の位置の対称位置に局所的最適解があるため、大域的最適解と局所的最適解の距離が離れるという大域的多峰性の性質を有する。

4. Rastrigin 関数

$$f_4(\mathbf{x}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad (\text{C.4})$$

$$\mathbf{x}^o = \mathbf{0}, \quad f_4(\mathbf{x}^o) = 0$$

Rastrigin 関数は多峰性・大域的多峰性・変数分離可能・良スケール性を有する関数で、多峰性の関数として代表的である。さらに、多くの局所的最適解が格子状に配置されており、原点対称である。大域的最適解付近も深い局所的最適解が存在する。

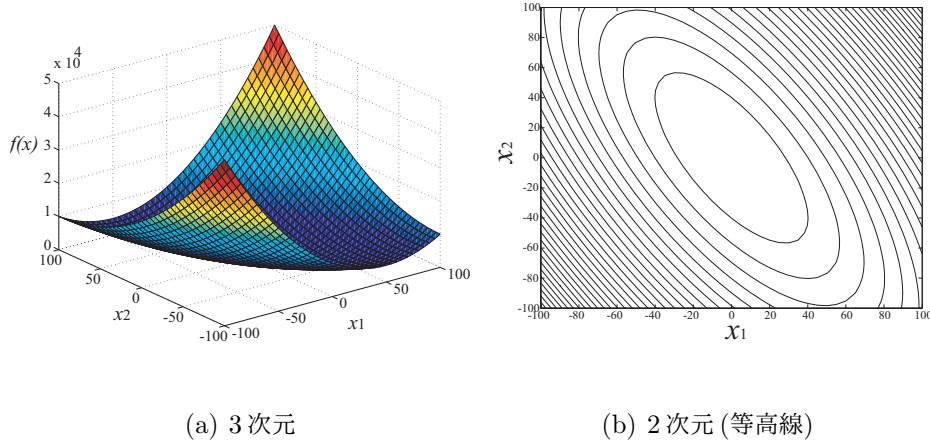


5. Schwefel 関数

$$f_5(x) = \sum_{n=1}^N \left(\sum_{j=1}^n x_j \right)^2 \quad (C.5)$$

$$\mathbf{x}^* = \mathbf{0}, \quad f_5(\mathbf{x}^o) = 0$$

Schwefel 関数は単峰性・変数分離不可能・悪スケール性を有する関数で、分離不可能かつ悪スケールな関数として代表的である。

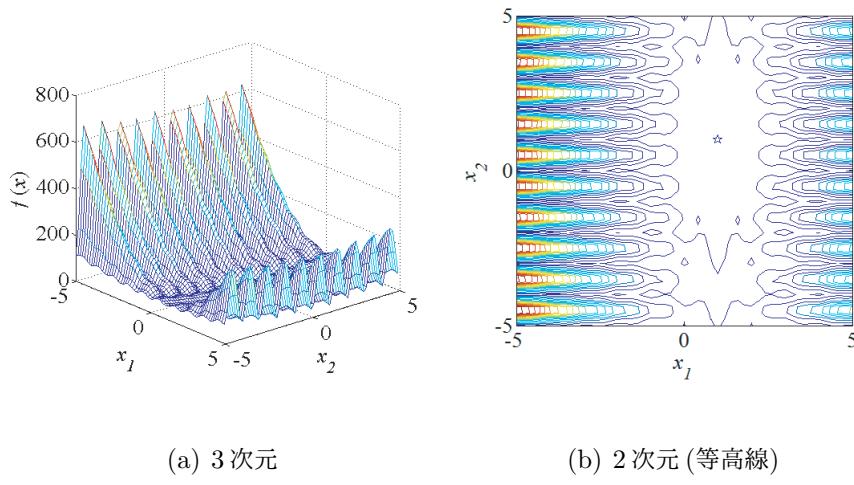
図 C.5 : Schwefel 関数 $-100 \leq x_n \leq 100$

6. Lévy 関数

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

$$\mathbf{x}^o = 0.25 \times \mathbf{1}, \quad f_6(\mathbf{x}^o) = 0$$

Lévy 関数は多峰性・大域的多峰性・変数分離不可能・良スケール性を有する関数である。さらに、多くの局所的最適解が格子状に配置されており、原点対称である。大域的最適解付近も深い局所的最適解が存在する。

図 C.6 : Lévy 関数 $-5 \leq x_n \leq 5$

7. Ackley 関数

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

$$\mathbf{x}^o = \mathbf{0}, \quad f_7(\mathbf{x}^o) = 0$$

Ackley 関数は多峰性・大域的単峰性・変数分離可能・良スケール性を有する関数である。複数の浅い局所的最適解が配置されており、原点対称である。

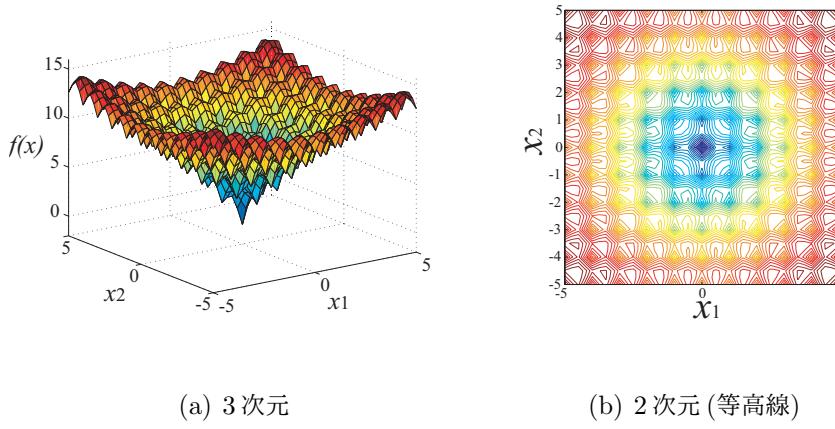


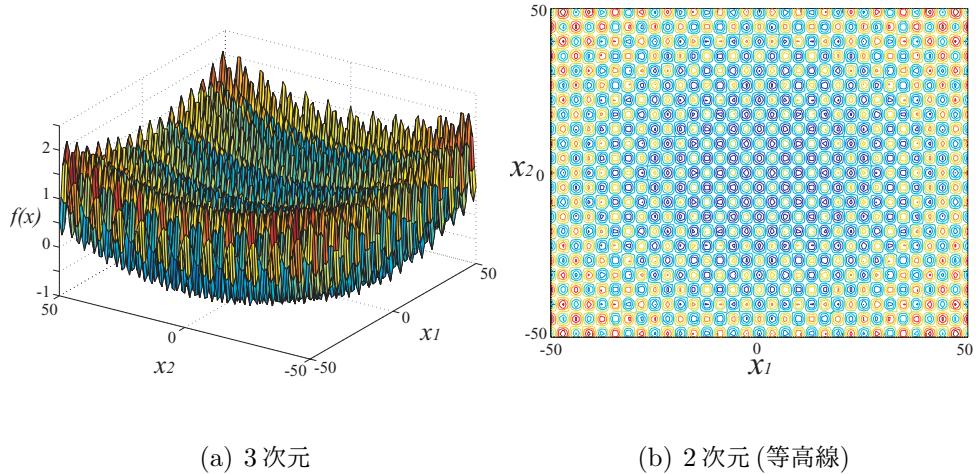
図 C.7 : Ackley 関数 $-5 \leq x_n \leq 5$

8. Griewank 関数

$$f_8(\mathbf{x}) = \frac{1}{4000} \sum_{n=1}^N x_n^2 - \prod_{n=1}^N \cos \left[\frac{x_n}{\sqrt{n}} \right] + 1 \quad (\text{C.8})$$

$$\mathbf{x}^o = \mathbf{0}, \quad f_8(\mathbf{x}^o) = 0$$

Griewank 関数は多峰性・大域的多峰性・変数分離不可能・良スケール性を有する関数である。Rastrigin に近いが、変数分離不可能である。三角関数の積和公式を利用して、第 2 項を変形すると、 $\prod_{n=1}^N \cos \left[\frac{x_n}{\sqrt{n}} \right] + 1 = \frac{1}{2} \sum_{k_n=0}^{\ell} \cdots \sum_{k_N=0}^{\ell} \cos \left[\sum_{n=1}^N \frac{(-1)^{k_n}}{\sqrt{n}} x_n \right]$ となるため、直線 $\sum_{n=1}^N (\pm 1/\sqrt{n}) x_n = 0$ 上に局所的最適解が存在する。

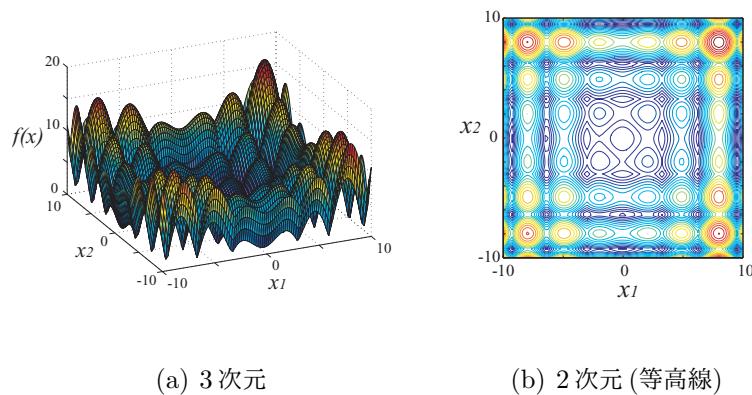
図 C.8 : Griewank 関数 $-50 \leq x_n \leq 50$

9. Alpine 関数

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

$$\boldsymbol{x}^o = \mathbf{0}, \quad f_9(\boldsymbol{x}^o) = 0$$

Alpine 関数は多峰性・大域的単峰性・変数分離可能・良スケール性を有する関数である。複数の浅い局所的最適解が配置されており、原点対称である。ただし、原点付近は凹凸が小さい。

図 C.9 : Alpine 関数 $-5 \leq x_n \leq 5$

熊谷 渉

1991年7月8日生。2010年3月 北海道北広島高等学校卒業。2014年3月 茨城大学 工学部 機械工学科卒業。2016年3月 首都大学東京大学院 理工学研究科 電気電子工学専攻 博士前期課程修了。同年4月 横河電機株式会社に入社。2020年9月 東京都立大学大学院（旧 首都大学東京大学院） 理工学研究科 電気電子工学専攻 博士後期課程修了、現在に至る。システム最適化（主にメタヒューリスティクス）、機械学習およびエネルギー最適化に関する研究に従事。電気学会の会員。博士（工学）。