

安定性解析に基づく多点型確率的最適化 アルゴリズムの設計に関する研究

2014 年度

小 熊 祐 司

学位論文 博士（工学）

安定性解析に基づく多点型確率的最適化
アルゴリズムの設計に関する研究

2014 年度

慶應義塾大学大学院 理工学研究科

小 熊 祐 司

目次

第1章	はじめに	1
1.1	本研究の背景	1
1.2	本研究の目的と位置づけ	4
1.3	本論文の構成	5
1.4	表記法について	6
第2章	多点型確率的最適化アルゴリズムの基礎と数理	9
2.1	本章について	9
2.2	最適化問題の数理	9
2.2.1	最適化問題の定式化	9
2.2.2	局所的最適解と大域的最適解	10
2.2.3	最適化問題の分類	10
2.3	多点型確率的最適化アルゴリズムの数理	13
2.3.1	多点型確率的最適化アルゴリズムの基本構造	13
2.3.2	エリート探索点関数	14
2.3.3	擬似乱数と確率的最適化	15
2.3.4	上下制限約内部への閉じ込め	16
2.4	局所的探索方策と大域的探索方策	16
2.4.1	大域的最適化アルゴリズムにおける「集中化と多様化」	16
2.4.2	局所的最適化アルゴリズムの基本構造	18
2.4.3	局所的最適化アルゴリズムの収束性	19
2.4.4	目的関数の解析的情報を用いた局所的最適化アルゴリズム	20
2.4.5	目的関数の解析的情報を用いない局所的最適化アルゴリズム	21
2.4.6	直線探索法	24
2.4.7	大域的探索方策	24
2.4.8	局所的探索と大域的探索の位置づけ	25
2.5	メタヒューリスティクス	26
2.5.1	多点型確率的最適化アルゴリズムとしてのメタヒューリスティクス	26
2.5.2	Particle Swarm Optimization	26
2.5.3	Differential Evolution	27
2.5.4	その他の手法	29
2.5.5	メタヒューリスティクスの課題	29
2.6	本章のまとめ	29

第 3 章	Particle Swarm Optimization の収束性解析	30
3.1	本章について	30
3.2	PSO の探索点位置更新式の確率システム表現	31
3.3	PSO システムの安定性解析	32
3.3.1	PSO システムの安定性の定義	32
3.3.2	慣性項がある場合の安定性解析	34
3.3.3	慣性項がない場合の安定性解析	36
3.4	安定性指標に基づく探索点の収束挙動の解析と持続的探索パラメータ	38
3.5	数値実験による持続的探索パラメータに基づく PSO の性能評価	40
3.5.1	数値実験条件	40
3.5.2	推奨値パラメータと持続的探索パラメータの比較	40
3.5.3	持続的探索パラメータの最適性	42
3.5.4	持続的探索パラメータ同士の比較	43
3.6	本章のまとめ	65
第 4 章	Differential Evolution の収束性解析	67
4.1	本章について	67
4.2	DE の近似モデルに対する探索点分布更新式	67
4.3	最大エントロピー法に基づく探索点分布解析	69
4.3.1	最大エントロピー法	69
4.3.2	独立な確率変数の和と特性関数	71
4.3.3	DE の探索点分布解析手順	72
4.4	数値実験による提案解析手法の有効性検証	72
4.4.1	数値実験条件	72
4.4.2	モーメント最大次数	73
4.4.3	実際の DE の探索点分布との比較	73
4.4.4	初期収束回避可能なパラメータ設定	74
4.5	本章のまとめ	74
第 5 章	数理的動特性解析可能な多点型確率的最適化アルゴリズムのフレームワーク	82
5.1	本章について	82
5.2	正規乱数線形結合型フレームワーク	83
5.3	正規乱数線形結合型フレームワークの安定性	84
5.3.1	正規乱数線形結合系フレームワークに対する安定性定理	84
5.3.2	数値実験による安定性定理の確認	89
5.4	正規乱数線形結合型フレームワークに基づく最適化アルゴリズムの例	90
5.4.1	PSO に着想を得た最適化アルゴリズム	90
5.4.2	数値実験条件	93
5.4.3	最適化性能の評価	94
5.4.4	パラメータと最適化性能の関係	94
5.5	本章のまとめ	109

第 6 章	おわりに	111
6.1	本論文のまとめ	111
6.2	今後の展望	112
付 録 A	確率変数・確率分布に関する諸性質	124
A.1	独立同一分布にしたがう確率変数の積の期待値と分散	124
A.2	独立同一分布にしたがう確率変数の標本分散・不偏分散の期待値と分散	125
A.3	対数正規分布の期待値と分散	129
付 録 B	最適化アルゴリズムの性能評価とベンチマーク問題	131
B.1	座標軸を回転させたベンチマーク問題の作成	131
B.2	ベンチマーク問題とその特徴	131
付 録 C	PSO の安定性指標の数値計算法	137
付 録 D	数値実験環境	138

第1章 はじめに

1.1 本研究の背景

工学的な問題のうち、製品設計問題や計画問題、意思決定問題など「多数の選択肢のなかから最も“良い”解」を選び出す問題の多くは、問題ごとに設定された評価指標（目的関数）に対する最大化あるいは最小化問題、すなわち最適化問題として数理的に記述することができる。具体的な最適化問題としては、たとえば、工業製品の寸法・形状・トポロジー最適化（最適設計）問題^[1]、流通生産分野における配車計画問題^[2,3]やコンテナ積付け最適化問題^[4]、工場の生産計画最適化問題^[5,6]、エネルギー分野における設備最適運用問題^[7]、機械学習分野におけるニューラルネットワークの学習問題^[8]、金融工学におけるポートフォリオ最適化問題^[9]などがあげられる。

本論文で扱う「最適化アルゴリズム」とは、上述したような各種の最適化問題を解くための数値計算法である。最適化アルゴリズムの歴史は古く、計算機が今日のかたちで発達する以前からすでに最急降下法やNewton法が存在していたが、現代に連なる研究の発端は、Dantzigによる線形計画問題に対する解法である単体法^[10]であると考えられる。当時のDantzigの研究の対象は、第2次世界大戦中のオペレーションズ・リサーチ、すなわち軍事上の兵站計画やスケジューリングに関する問題であったが、戦後、計算機の演算性能の向上と相まって、Dantzigの開発した単体法、またそこから発展した各種最適化アルゴリズムの適用の場は非線形計画問題をも対象として大きく拡がり、現代では、最適化アルゴリズムは工学を支える基盤技術のひとつとしてその地位を確立している。

工学的な問題に対する最適化は、4つのプロセス

- (1) 原問題のモデリング¹をおこない、最適化問題として定式化する
- (2) 定式化された最適化問題を解くための最適化アルゴリズムを設計する
- (3) (1)で定式化された最適化問題に対して、(2)で設計された最適化アルゴリズムを適用してこれを解き、最適解を得る
- (4) 得られた解の工学的・経済的妥当性を吟味して設計に反映させる

からなると考えられる。本論文では、おもに(2)のプロセスに相当する「最適化アルゴリズムの設計」を扱うが、上述したように、最適化アルゴリズムの設計は、最適化の一連のプロセスのなかの一部に過ぎず、優れた最適化アルゴリズムの設計が、直ちに最適な製品や計画を実現することとはならないことに注意する必要がある。なお、本来(1)と(2)は表

¹ここで、「モデリング」とは、原問題の本質的な部分を抽出して数式で表現することである。最適化において、原問題のうち本質的な部分とは問題の評価関数や制約条件といった数理的構造であり、その他の情報は除外できる。

裏一体の関係にあり、「有効なアルゴリズムを適用できるように」定式化をおこない、また問題に合わせたアルゴリズムを選ぶのが常道である。たとえば、制御分野におけるモデル予測制御や、機械学習分野におけるサポートベクターマシンは、凸2次計画とよばれる、「高速かつ確実に最適解を求めることができるアルゴリズムが存在する」かたちで定式化が試みられる^[11, 12]。

近年では、計算機の演算速度、市販の最適化ソルバの最適化性能ともに向上しており、大規模な線形計画や凸2次計画といった問題に対して高速に最適解を求めることが可能である。また、目的関数や制約関数が一般の非線形関数で表される場合でも、目的関数や制約条件式の解析的情報（勾配やHesse行列など）が利用できるならば、逐次2次計画法^[13]や信頼領域法^[14]といった実用上有効なアルゴリズムを用いることができる。これらのアルゴリズムは、基本的に唯一の最適解をもつ最適化問題を想定した手法であるが、近年では、カオス力学系の知見を利用し、局所的最適解を多数含む最適化問題であっても、そのなかでより優れた解を提示するアルゴリズムが考案されている^[15, 16]。このように、最適化可能な問題のクラスや規模は確実に拡大しつつあるが、それでもアルゴリズムの解きやすい形で定式化することが重要であるのは今後も変わらないと考えられる。

ところで、今日の大規模化・複雑化する工学上の諸問題において、モデリングが困難な問題に対する最適化のニーズが高まっている。こういった問題に対しては、設計変数とそれに対する目的関数値という入出力関係のみを用いた最適化アルゴリズムが必要となる。このような最適化アルゴリズムを直接探索アルゴリズムとよぶ。直接探索アルゴリズムでよく知られたものとしてはNelder-Mead法^[17]があげられる。Nelder-Mead法は、シンプルなアルゴリズムでありながらも頑健性があり、統計処理ソフトウェアであるR²の標準最適化アルゴリズムとして組み込まれているなど、現代においても有効な直接探索アルゴリズムのひとつとして広く用いられている。

最適化に関する理論は、1960年代に凸解析^[18]とよばれる数学分野が発展し、それとともに、解の最適性や最適化問題の双対性、感度解析などの最適化に関する重要な諸概念が整備され、以来、最適化問題の解析やアルゴリズムの設計において強力な道具となっている。一方で、近年の計算機演算能力の発達を背景に、「自然界における最適化過程の模倣」という観点から最適化アルゴリズムを構築する試みがあり、1970年代から2000年代にかけて、遺伝的アルゴリズム(GA)^[19]、進化戦略(ES)^[20]、Particle Swarm Optimization(PSO)^[21, 22]、Differential Evolution(DE)^[23, 24]など、このアプローチに基づくアルゴリズムが多数提案されてきた。これらのアルゴリズムに共通した特徴は、いずれも厳密な数学的背景を有しておらず、得られた解に対する最適性の証明もないが、実用的な計算時間で良質な解を提示することが可能という点であり、それぞれ有効な最適化性能をもつことが数値計算結果から示されている。

数学的厳密性にとらわれず、短時間で良質な解を求めるために問題に応じて設計されたアルゴリズムをヒューリスティクスという。上述したアルゴリズムは、いずれも最適化問題が具体的にどのような目的関数式や制約条件式で記述されるかという情報を使用しない直接探索アルゴリズムであり、問題に対する事前知識がなくても幅広い問題への適用が可能であることから、とくにメタヒューリスティクスとよばれる。

メタヒューリスティクスのアルゴリズムの構造としては、(1)複数の探索点を用いた多

²<http://www.r-project.org>

点型探索, (2) 擬似乱数を用いた確率的探索, といった特徴があげられ, その数理的構造に着目する場合は多点型確率的最適化アルゴリズムともよばれる.

初期のメタヒューリスティクスの成功を背景に, 1990年代の後半から, 初期のメタヒューリスティクスの改良手法や, まったく新しいメタヒューリスティクスなど, 様々な最適化アルゴリズムが提案され, 今日においてもなお新たなアルゴリズムの開発が盛んである. しかし, 新たな最適化アルゴリズムの開発が活発である一方で, 開発された新しい最適化アルゴリズムの産業応用は著者の知る限り少なく, オリジナルのPSOやDEを適用している例が大半である. 著者はその原因を以下の5点にあると考えている. 以下, 便宜上, 最適化アルゴリズムを研究・開発する人間を「開発者」, 開発されたアルゴリズムを利用して実際の最適化問題を解く人間を「利用者」とよぶこととする.

- (1) 最適化アルゴリズムのパラメータチューニングが困難である
- (2) 汎用的な最適化アルゴリズムを利用するより, 問題に対する専門家が, 独自に問題に特化したアルゴリズムを構築したほうが良好な解が得られる
- (3) 有望とされる最適化アルゴリズムと実問題の相性が悪い
- (4) 有力な最適化アルゴリズムが利用者に知れ渡っていない
- (5) (5)とは逆に, 最適化アルゴリズムに関する情報が多すぎて, 最適化アルゴリズムを専門としていない利用者にとって, アルゴリズムの選択基準がない

(1)について, より高度なアルゴリズムとなればなるほど内包するパラメータの数は増える傾向にある. アルゴリズムの性能を発揮するためには, パラメータを適切に調整する必要があるが, その調整則が数理的な根拠をもたない, あるいは問題に応じて性能を発揮するパラメータが大きく異なるなど, 利用者にとってはパラメータ調整という本来の業務とは異なる領域で負担を強いることとなる. この課題に対しては, パラメータの調整則を理論的に導く^[25–28], あるいはパラメータ自動調整能力をもつアルゴリズムを構築する^[29–32]ことで, 利用者の負担を減らすことが目指されている.

(2)について, 一般論として, 問題を限定しない汎用的な最適化アルゴリズムよりも, 問題に対する事前知識を組み入れた最適化アルゴリズムの方が, 良質な解を提示すると考えられる. つまり, 実問題における最適化においてより良い解を希求するのであれば, 究極的には問題に合わせた最適化アルゴリズムの改良や新規構築が必要となる. 具体例としては, たとえば原子炉炉心設計最適化などがあげられる. 原子炉炉心設計最適化では, 目的関数の計算コストがきわめて大きく, 問題に対する専門家の知識を活用した近似計算, IF-THEN ルールの適用, 最適解探索範囲の枝刈りを用いることで現実的な時間内での最適化を実現している^[33–35].

(3)について, 通例, 最適化アルゴリズムの性能評価はベンチマーク問題への適用を通して行われるが, そのベンチマーク問題の構造, 規模, 計算時間が現実問題のそれと大きく乖離している可能性がある. この課題に対して, 産業界から抽出した実問題からベンチマーク問題集およびプログラムを作成し, より実問題を志向したアルゴリズムの設計を促す試みがある^[36, 37]. 現状, エネルギープラント運用計画, 上水道送水ポンプ運用計画, 自動ピッキングシステム運用計画, 運転時の不確実性を考慮したエネルギー計画, ボイラ制

御における PID コントローラ設計，太陽光発電モジュール直並切替に関するの計 6 つの問題が Web で公開中あるいは準備中である³。

(4) と (5) については本質的には同じ問題であり，アルゴリズムの開発者と利用者の距離が遠いことが原因であると考えられる。最適化アルゴリズムの産業応用という観点からは，開発者は産業界のニーズを捉えたアルゴリズムの開発と情報発信，利用者は最新の最適化アルゴリズムの動向調査と自身の携わる問題への適用性調査をそれぞれ精力的におこなう必要があると考えられる。これらの課題については本論文では以降とくに言及しない。

本論文では，上述した課題のうち (1) および (2) の課題解決に資するため，数理的解析に基づく最適化アルゴリズムの設計論やパラメータチューニングの方策について考察していく。

1.2 本研究の目的と位置づけ

本研究では，以下で述べる 2 つのアプローチにより，メタヒューリスティクスを「多点型確率的最適化アルゴリズム」と解釈し，その動特性解析をおこない，最適化問題を解く際のアルゴリズム設計やパラメータ調整の数理的根拠を提供することを目的とする。

第 1 のアプローチとして，有力なメタヒューリスティクスである PSO と DE に対し，動特性の数理的解析によってそれらの収束性を解析し，複数の局所的最適解を有する最適化問題に対する大域的最適化性能に優れたパラメータの設定指針を提供する。PSO のように探索点位置の更新式中に擬似乱数が内在されるアルゴリズムでは，その動特性を決定論的に解析することは原理的に不可能であるが，本研究では，最適化アルゴリズムが多数回の計算の反復を前提とすることに着目し，巨視的な時間間隔での動特性を論じる。より具体的に，PSO の場合では，各探索点位置の更新式を等価な確率システムとして記述し，その安定性解析をおこなう。確率システムおよびその安定性は，1940 年代の Wiener の研究を端緒として Kushner や Wonham らにより精力的に研究されており，今日では確率システム理論として体系化されている^[38]。しかしながら，本研究で対象とするシステムにおいては，たとえば乱数が正規分布にしたがうことを仮定できないなど，既存の確率システムの理論体系の成果を直接用いてその性質を解析することは容易ではない。したがって，その解析に際しては，アルゴリズムごとに，それぞれに特化した解析手法を構築していくことが求められる。

第 2 のアプローチとして，数理的に動特性解析が可能な多点型確率的最適化アルゴリズムの「フレームワーク」を具体的に提案するとともに，そのなかで解きたい問題に合わせてアルゴリズムを設計する考え方を示す。そのうえで，このフレームワークに対して，探索点位置の分散共分散行列を指標とする探索点分布に関する安定性定理を与える。この定理は，このフレームワークに基づいて構築されるアルゴリズムすべてに対して適用可能なものであり，その適用結果を用いれば，個々のアルゴリズムに対して探索点分布の制御則を与えることが可能となる。このフレームワークに従えば，たとえば勾配や Hesse 行列といった目的関数に対する解析的情報を組み入れつつ，かつ数理的に解析可能な動特性を有するアルゴリズムの設計が可能となる。

³<http://www.ssl.te.chiba-u.jp/okamoto/BP-IA/index.html>

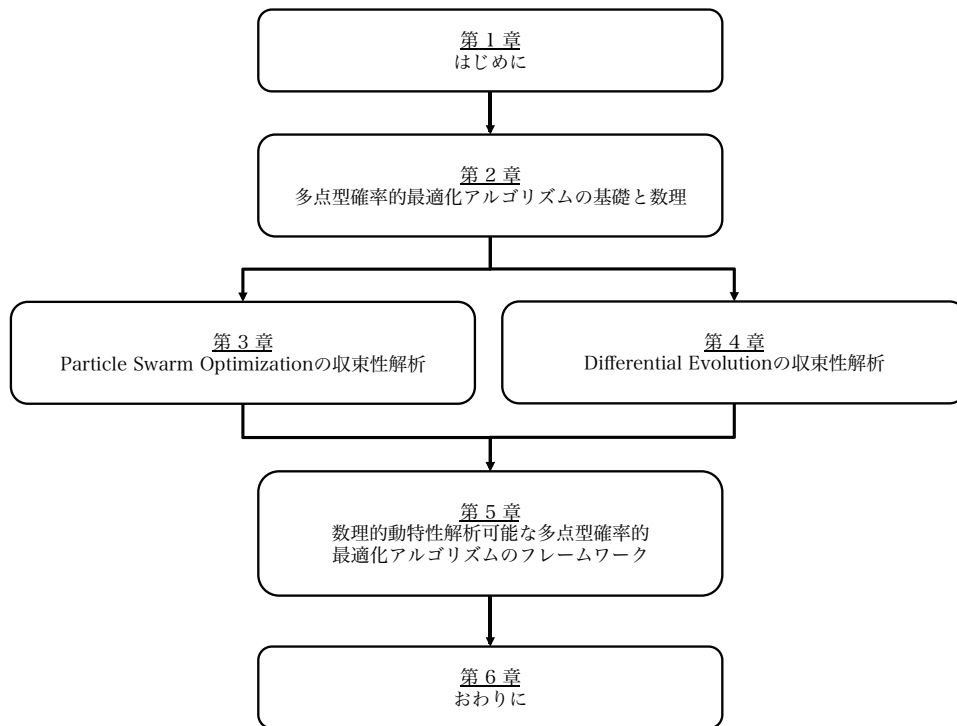


Fig. 1.1 Outline of dissertation.

以上の成果により、最適化問題に対して PSO や DE を適用する場合、数理的な根拠・指針をもとにパラメータを調整することが可能となり、また、問題にあわせて独自に最適化アルゴリズムを設計する場合でも、その動特性の数理解析に基づく適切なパラメータの設定が可能となる。これらのことは、1.1 節であげ産業上の実問題最適化に対する課題のうち (1) と (2) に対しても大きく資するものであると考えられる。

1.3 本論文の構成

本論文の構成を以下に述べる。Fig. 1.1 に、本論文の章構成を示す。本章の以降の部分を用いて、本論文を通じて使用する数学的用語とその定義、および表記法をまとめる。

第2章では、本論文で扱う最適化問題、およびそれを解く最適化アルゴリズムの構造を数学的に記述し、第3章以降の議論のための土台を形成する。また、既存の多点型確率的最適化アルゴリズムの例や応用研究や改良研究について概観するとともに、これらの手法に数学的裏付けが必ずしもないこと、またそれに起因する実用上の課題を指摘し、本論文における問題の提起をおこなう。

この課題に対して、第3章および第4章では、代表的な多点型確率的最適化アルゴリズムのひとつである Particle Swarm Optimization (PSO) および Differential Evolution (DE) に対する収束性解析をおこない、パラメータとアルゴリズムの動特性に関係について、数学的に議論する。とくに第3章では、収束性解析の結果に基づき、優れた最適化性能を実現するパラメータ設定の具体的な例を与え、数値実験によりその有効性を確認する。

第5章は本論文の主要部分にあたる。第3章および第4章における多点型確率的最適化アルゴリズムの収束性解析は、それぞれ個別のアルゴリズムに特化したものであり、その解析手法は拡張性に乏しいといえるものである。そこで第5章では、第3章および第4章の知見を踏まえつつも、まったく逆のアプローチとして、まず数理的にその動特性を解析可能な多点型確率的最適化アルゴリズムの「フレームワーク」を構築し、つぎのそのフレームワークのもとで具体的な最適化アルゴリズムを構築するという方策を提唱する。また、ここで提案するフレームワークに基づく具体的な最適化アルゴリズムの一例として、PSOに着想を得た新しいアルゴリズムを構築し、数値実験によりその有効性を確認する。

第6章では、本論文の内容をまとめるとともに、残された課題や今後の研究展望を述べる。

また、確率変数・確率分布に関する諸性質、本論文中の数値実験で用いる最適化ベンチマーク問題集、第3章で導かれるPSOの安定性指標の数値計算法、および数値実験環境をそれぞれ付録としてまとめている。

1.4 表記法について

ベクトル 実数全体からなる集合を $\mathbb{R}$ 、整数全体からなる集合を $\mathbb{Z}$ 、自然数全体からなる集合を $\mathbb{N}$ で表す。本論文では、とくに断りのない限り、ベクトルとは列ベクトルをさすものとし、 $N \in \mathbb{N}$ 次元のベクトルを、たとえば

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

と表す。ここで $x_n (n = 1, \dots, N)$ を $\mathbf{x}$ の第 n 要素という。 N 次元ベクトル全体の集合を N 次元ユークリッド空間といい $\mathbb{R}^N$ で表す。以降、紙面の都合上、列ベクトルを成分を含めて記述する際、転置記号 $\top$ を用いて、 $\mathbf{x} = (x_1, x_2, \dots, x_N)^\top$ と表す場合もある。

特殊なベクトルとして、すべての要素が0のベクトル $(0, 0, \dots, 0)^\top$ 、すべての要素が1のベクトル $(1, 1, \dots, 1)^\top$ を導入し、それぞれ $\mathbf{0}, \mathbf{1}$ と表すこととする。

2つのベクトル $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ に対する内積を $\mathbf{a}^\top \mathbf{b}$ と表す。また、ベクトル $\mathbf{x} \in \mathbb{R}^N$ に対するユークリッドノルムを $\|\mathbf{x}\|$ と表す。

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、

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

と表す。ただし、文脈に応じて、行列 $\mathbf{A}$ の成分に注目し、 $\mathbf{A}$ を成分 $a_{nm} (n = 1, 2, \dots, N, m = 1, 2, \dots, M)$ を用いて $\mathbf{A} = (a_{nm})$ と書く場合もある。行列 $\mathbf{A} \in \mathbb{R}^{N \times M}$ の行・列成分を逆にした M 行 N 列の転置行列を、転置記号 $\top$ を用いて $\mathbf{A}^\top$ と表す。すべての要素が0である行列を零行列といい $\mathbf{O}$ で表す。

行列の一記法として、要素数が等しい列ベクトル $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$ を列方向に並べた表記

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M) \quad (1.3)$$

および、要素数が等しい行ベクトル $\mathbf{x}_1^\top, \mathbf{x}_2^\top, \dots, \mathbf{x}_M^\top$ を行方向に並べた表記

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_M^\top \end{pmatrix} \quad (1.4)$$

を必要に応じて採用する。

正方行列のうち、対角成分以外の成分がすべて0であるような行列 $\mathbf{D} \in \mathbb{R}^{N \times N}$ を対角行列といい、対角成分 $d_{11}, d_{22}, \dots, d_{NN}$ を用いて $\text{diag}[d_{11}, d_{22}, \dots, d_{NN}]$ と表す。とくに、すべての対角要素が1である対角行列を単位行列といい $\mathbf{I}$ で表す。

正方行列 $\mathbf{A} \in \mathbb{R}^{N \times N}$ に対して、行列式を $\det \mathbf{A}$ 、階数を $\text{rank } \mathbf{A}$ で表す。 $\text{rank } \mathbf{A} = N$ となる正方行列 $\mathbf{A} \in \mathbb{R}^{N \times N}$ を正則行列といい、正則行列 $\mathbf{A}$ に対する逆行列を $\mathbf{A}^{-1}$ と表す。

$\mathbf{B}^\top \mathbf{B} \in \mathbb{R}^{M \times M}$ が正則となる $N \geq M$ の一般の N 行 M 列行列 $\mathbf{B} \in \mathbb{R}^{N \times M}$ に対して、Moore-Penrose の一般逆行列を $\mathbf{B}^\dagger \in \mathbb{R}^{M \times N}$ で表す。ただし、 $\mathbf{B}^\dagger$ は、

$$\mathbf{B}^\dagger = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \quad (1.5)$$

で与えられる [39, 40]。

正方行列 $\mathbf{Q} \in \mathbb{R}^{N \times N}$ において、任意の n, m ($n, m = 1, 2, \dots, N$) に対して $q_{nm} = q_{mn}$ がなりたつとき $\mathbf{Q}$ を対称行列という。対称行列 $\mathbf{Q} \in \mathbb{R}^{N \times N}$ が任意の $\mathbf{x} \in \mathbb{R}^N$ に対して $\mathbf{x}^\top \mathbf{Q} \mathbf{x} \geq 0$ を満たすとき、 $\mathbf{Q}$ を半正定値行列といい、 $\mathbf{Q} \succeq \mathbf{O}$ と表す。とくに、 $\mathbf{x} = \mathbf{0}$ 以外の任意の $\mathbf{x} \in \mathbb{R}^N$ に対して $\mathbf{x}^\top \mathbf{Q} \mathbf{x} > 0$ を満たすとき、 $\mathbf{Q}$ を正定値行列といい、 $\mathbf{Q} \succ \mathbf{O}$ と表す。

確率変数 スカラー変数 $X \in \mathbb{R}$ 、あるいはベクトル変数 $\mathbf{X} \in \mathbb{R}^N$ 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 $\mathbf{X}$ の期待値および分散をそれぞれ $\mathbb{E}[\mathbf{X}]$ 、 $\mathbb{V}[\mathbf{X}]$ と書くこととする。また、 P 個の確率変数 $\mathbf{X}^{(p)}$ ($p = 1, \dots, P$) に対して、その平均 $\sum_{p=1}^P \mathbf{X}^{(p)}$ を $\bar{\mathbf{X}}$ と書くこととする。

確率分布 確率変数 $\mathbf{X}$ がある確率分布 $\mathcal{P}$ にしたがって分布することを $\mathbf{X} \sim \mathcal{P}$ と書き、確率分布 $\mathcal{P}$ の確率密度関数を $f_{\mathcal{P}}(x)$ (確率変数がスカラーの場合) あるいは $f_{\mathcal{P}}(\mathbf{x})$ (確率変数がベクトルの場合) と書く。本論文で利用するおもな確率分布とその確率密度関数を以下に示す。

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

$$f_{\mathcal{N}}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (1.6)$$

(1.6) 式において、 $\mu \in \mathbb{R}$ および $\sigma^2 > 0$ はそれぞれ分布の平均、分散である。

(2) 対数正規分布： $\mathcal{LN}(\hat{\mu}, \hat{\sigma}^2)$

$$f_{\mathcal{LN}}(x; \hat{\mu}, \hat{\sigma}^2) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}x} \exp\left(-\frac{(\log(x) - \hat{\mu})^2}{2\hat{\sigma}^2}\right), \quad x > 0 \quad (1.7)$$

(1.7) 式において, $\hat{\mu} \in \mathbb{R}$ および $\hat{\sigma}^2 > 0$ はそれぞれ分布の位置母数, 尺度母数である.

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

$$f_{\mathcal{U}_{\mathbb{R}}}(x; a, b) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}, \quad x \in \mathbb{R} \quad (1.8)$$

(1.8) 式において, $a, b \in \mathbb{R}$ はそれぞれ分布の下限值, 上限値である.

(4) 整数値一様分布： $\mathcal{U}_{\mathbb{Z}}(m, n)$

$$f_{\mathcal{U}_{\mathbb{Z}}}(x; m, n) = \begin{cases} \frac{1}{m-n+1}, & x \in \{m, \dots, n\} \\ 0, & \text{otherwise} \end{cases}, \quad x \in \mathbb{Z} \quad (1.9)$$

(1.9) 式において, $m, n \in \mathbb{Z}$ はそれぞれ分布の下限值, 上限値である.

(5) N 次元正規分布： $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$f_{\mathcal{N}}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{(\sqrt{2\pi})^{-N}}{\sqrt{\det \boldsymbol{\Sigma}}} \exp\left(-\frac{(\mathbf{x} - \boldsymbol{\mu})^{\top} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}{2}\right) \quad (1.10)$$

(1.10) 式において, $\boldsymbol{\mu} \in \mathbb{R}^N, \boldsymbol{\Sigma} \in \mathbb{R}^{N \times N}$ はそれぞれ分布の平均ベクトル, 分散共分散行列である.

確率密度関数のモーメント スカラーの確率密度関数 $f(x)$ の m 次のモーメントとして, μ_m, μ'_m および α_m を, それぞれ

$$\mu_m = \int_{-\infty}^{\infty} x^m f(x) dx \quad (1.11a)$$

$$\mu'_m = \int_{-\infty}^{\infty} (x - \mu_1)^m f(x) dx \quad (1.11b)$$

$$\alpha_m = \frac{\mu'_m}{(\sqrt{\mu'_2})^m} \quad (1.11c)$$

と定義する. また, 確率分布の特徴を示す量として, 平均 μ , 分散 σ^2 , 歪度 β_1 , および尖度 β_2 を, それぞれ

$$\mu = \mu_1 \quad (1.12a)$$

$$\sigma^2 = \mu_2 - \mu_1^2 \quad (1.12b)$$

$$\beta_1 = \alpha_3 \quad (1.12c)$$

$$\beta_2 = \alpha_4 \quad (1.12d)$$

と定義する⁴.

⁴本論文では, 正規分布の尖度を 3 とする定義を採用する.

第2章 多点型確率的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、本論文の主題である多点型確率的最適化アルゴリズムの数理解析のための準備として、多点型確率的最適化アルゴリズムの構造や最適化方策を数学的に整理し、また統一的に記述する。まず、2.2節では、本論文で扱う最適化問題の具体的な定式化を与える。つぎに2.3節では、2.2節で定義した最適化問題を解くための最適化アルゴリズムの構造を議論し、とくに本論文で中心的に論じる多点型確率的最適化アルゴリズムについて数学的に記述する。2.4節では、2.3節で述べたアルゴリズムの構造に則った具体的な最適化アルゴリズムの最適化方策を論じる。2.5節では、多点型確率的最適化アルゴリズムの例として、既存の有効なメタヒューリスティクスである Particle Swarm Optimization と Differential Evolution を取りあげ、これらのアルゴリズムを概説したうえで、実問題への応用状況や、手法の改良研究、課題についてまとめる。最後に2.6節では、本章の内容をまとめるとともに、2.5節であげたメタヒューリスティクスの課題を基に、第3章以降へ繋がる問題提起をおこなう。

2.2 最適化問題の数理

2.2.1 最適化問題の定式化

N 次元実数空間 $\mathbf{R}^N$ を定義域とする実数値目的関数 $f : \mathbf{R}^N \rightarrow \mathbf{R}$ と、空でない閉集合 $S \subseteq \mathbf{R}^N$ が与えられたとき、集合 S 上で目的関数 $f(\mathbf{x})$ を最小化する問題

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (2.1a)$$

$$\text{subj. to } \mathbf{x} \in S \quad (2.1b)$$

を本論文で扱う最適化問題として定義する。目的関数 f の最大化を考える場合であっても、 $f(\mathbf{x})$ の符号を反転させることにより、一般性を失わずに最適化問題 (2.1) に帰着できる。よって、以降は関数の最小化について議論していくこととする¹。

最適化問題 (2.1) において、集合 S を実行可能解集合といい、その要素 $\mathbf{x} \in S$ を実行可能解という。一般に、実行可能解集合 S は、不等式制約関数 $\mathbf{g} : \mathbf{R}^N \rightarrow \mathbf{R}^L$ および等式制約関数 $\mathbf{h} : \mathbf{R}^N \rightarrow \mathbf{R}^M$ を用いて、

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

¹本節の議論は、文献 [41, 42] を参考にまとめた。

と表現される。(2.2)式において、 $g(x) \leq 0$ を不等式制約条件、 $h(x) = 0$ を等式制約条件という。

2.2.2 局所的最適解と大域的最適解

最適化問題 (2.1) の解について議論する。ある $x^* \in \mathbb{R}^N$ と正数 $\epsilon > 0$ に対して、開集合

$$B(x^*; \epsilon) = \{x \in \mathbb{R}^N \mid \|x - x^*\| < \epsilon\} \quad (2.3)$$

を x^* の近傍とよぶ。ある実行可能解 x^* が、

$$f(x^*) \leq f(x), \forall x \in B(x^*; \epsilon) \cap S \quad (2.4)$$

を満たすとき、 x^* を最適化問題 (2.1) の局所的最適解といい、局所的最適解 x^* における目的関数値 $f(x^*)$ を局所的最適値という。また、局所的最適解 x^* が、

$$f(x^*) < f(x), \forall x \in B(x^*; \epsilon) / x^* \quad (2.5)$$

を満たすとき、 x^* を最適化問題 (2.1) の狭義局所的最適解という。

ある実行可能解 $x^o \in S$ が、

$$f(x^o) \leq f(x), \forall x \in S \quad (2.6)$$

を満たすとき、 x^o を最適化問題 (2.1) の大域的最適解といい、大域的最適解 x^o に対する目的関数値 $f(x^o)$ を大域的最適値という。また、大域的最適解 x^o が唯一性

$$f(x^*) < f(x), \forall x \in S / x^o \quad (2.7)$$

を満たすとき、 x^o は唯一の大域的最適解であるという。

最適化問題 (2.1) において、局所的最適解および大域的最適解は一般的に複数個存在しうる。そこで (2.4) 式を満足する局所的最適解の集合を S^* 、(2.6) 式を満足する大域的最適解の集合を S^o と定義する。このとき、局所的最適解集合 S^* と大域的最適解の集合 S^o に対して、

$$S^o \subseteq S^* \quad (2.8)$$

の関係が成立する。ただし、最適化問題 (2.1) において、局所的最適解と大域的最適解は必ずしも存在するとは限らず、局所的最適解集合 S^* および大域的最適解集合は S^o は空集合 $\emptyset$ となりうることに注意する必要がある。なお、以降とくに断りのないかぎり、「最適解」という語句は任意の局所的最適解を指すものとする。

最適化問題 (2.1) に対して、1つ以上の局所的最適解の存在を仮定して、局所的最適解あるいは大域的最適解を求めるためのアルゴリズムを最適化アルゴリズムという。最適化アルゴリズムの数理的構造については次節で詳しく述べる。

2.2.3 最適化問題の分類

最適化問題 (2.1) は、その目的関数と制約条件の性質によりいくつかの種類に分類されており、そのなかの一部の問題に対しては、大域的最適解を得るための有力な最適化アルゴリズムが知られている。最適化問題を解くとき、対象となる問題の性質が既知であれば、それに合わせて適切な最適化アルゴリズムを選択することができる。

線形計画問題と非線形計画問題 最適化問題の目的関数 f , 制約条件式 g, h がいずれも線形である最適化問題を線形計画問題という。線形計画問題は, ベクトル $c \in \mathbb{R}^N, b \in \mathbb{R}^M$ および行列 $A \in \mathbb{R}^{N \times M}$ を用いて,

$$\min_x c^\top x \quad (2.9a)$$

$$\text{subj. to } Ax = b \quad (2.9b)$$

$$x \geq 0 \quad (2.9c)$$

と記述される。一方, 目的関数および制約関数のいずれか1つ以上が非線形である場合の最適化問題を非線形最適化問題 (非線形計画問題) という。線形計画問題に対しては, 単体法^[10] や内点法^[43] をはじめとする有効な最適化アルゴリズムが知られているが, 非線形最適化問題に対する有効なアルゴリズムは, 後述する凸性の有無などにより異なる。

凸最適化問題と非凸最適化問題 目的関数 f が凸関数であり, 実行可能解集合 S が凸集合である場合の最適化問題を凸最適化問題 (凸計画問題) という。凸最適化問題の代表的な例としては, 凸2次計画問題があげられる。凸2次計画問題は, ベクトル $c \in \mathbb{R}^N, b_1 \in \mathbb{R}^L, b_2 \in \mathbb{R}^M$, 行列 $Q \in \mathbb{R}^{N \times N} \succeq 0, A_1 \in \mathbb{R}^{L \times N}, A_2 \in \mathbb{R}^{M \times N}$ を用いて,

$$\min_x \frac{1}{2} x^\top Q x + c^\top x \quad (2.10a)$$

$$\text{subj. to } A_1 x \leq b_1 \quad (2.10b)$$

$$A_2 x = b_2 \quad (2.10c)$$

と記述される。凸最適化問題においては, 局所的最適解集合 S^* と大域的最適解集合 S^o は一致し, 2.4.3 項で述べる大域的収束性を有するアルゴリズムを用いれば, 大域的最適解を求めることができる。凸最適化問題のうち, とくに目的関数 f が狭義凸関数であり, かつ実行可能解集合 S が凸集合である場合の最適化問題を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をもつ。

一方, 目的関数 f が非凸関数, または実行可能解集合 S が非凸集合である場合の最適化問題を非凸最適化問題 (非凸計画問題) という。一般の非凸最適化問題においては, 局所的最適解集合 S^* と大域的最適解集合 S^o は一致せず, 問題によっては大域的最適解を確実に求めることは困難である。すべての線形計画問題は凸最適化問題であるが, 非線形最適化問題のうち凸最適化問題となるのは上述した凸2次計画問題など, ごく一部に限られる。しかしながら, 非凸計画問題を解く場合にも, 凸最適化問題, とくに凸2次計画問題に対するアルゴリズムを拡張して用いることは有効なアプローチとなりうる^[13]。

無制約最適化問題 最適化問題 (2.1) において, 実行可能解集合 S が $\mathbb{R}^N$ 全体である場合, すなわち,

$$\min_x f(x) \quad (2.11)$$

の形式で記述できる場合の最適化問題を無制約最適化問題という。現実的には, 制約をまったく有しない最適化問題は少ないと考えられる。しかしながら, 一般の制約条件を有する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値に計上した関数を

構成し、これを拡大目的関数として無制約のもとで最適化するという方法が考えられる。この方法を用いれば、制約を侵害した領域の拡大目的関数値を選択的に大きく設定することができ、制約を陽に考慮せずとも、最適化制約条件を満足（あるいは制約条件の侵害量が僅かな）解を自然に導くことができる。このような手法を総称して外点ペナルティ法とよぶ。

目的関数 $f: \mathbb{R}^N \rightarrow \mathbb{R}$ および制約条件関数 $g: \mathbb{R}^N \rightarrow \mathbb{R}^L, h: \mathbb{R}^N \rightarrow \mathbb{R}^M$ を用いて定義される非凸最適化問題

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (2.12a)$$

$$\text{subj. to } g(\mathbf{x}) \leq \mathbf{0} \quad (2.12b)$$

$$h(\mathbf{x}) = \mathbf{0} \quad (2.12c)$$

を考える。最適化問題 (2.12) を外点ペナルティ法を用いて無制約最適化問題に緩和する場合、拡大目的関数の構成手法としていくつかの種類が考えられるが、そのなかでもつぎの2つがよく知られている。

- (1) 制約条件の侵害量の2乗をペナルティとして拡大目的関数 $\hat{f}_1(\mathbf{x})$ を構成する手法

$$\hat{f}_1(\mathbf{x}) = f(\mathbf{x}) + s_1 \sum_{l=1}^L (\max\{0, g_l(\mathbf{x})\})^2 + s_2 \sum_{m=1}^M h_m^2(\mathbf{x}) \quad (2.13)$$

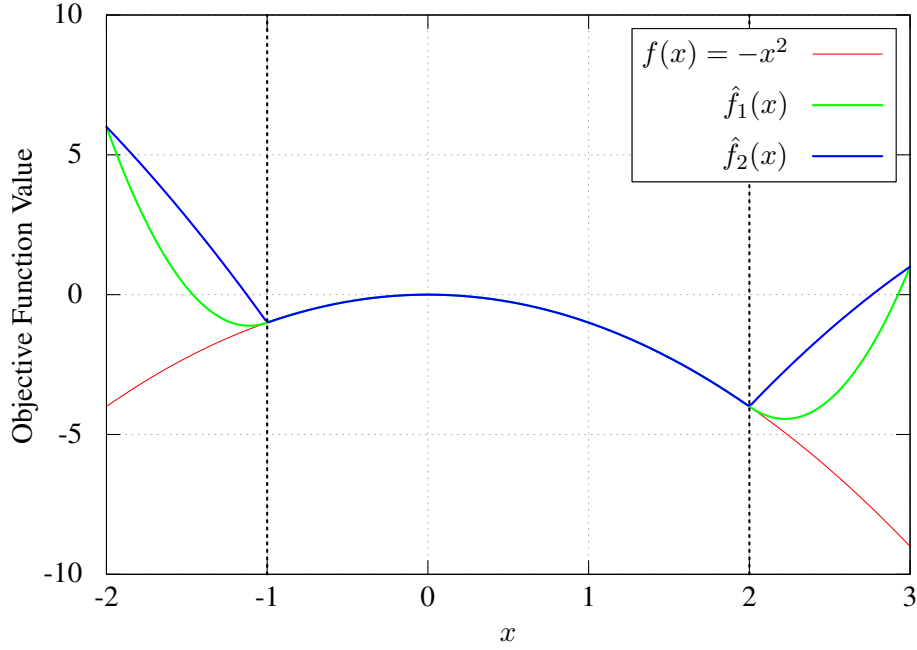
- (2) 制約条件の侵害量の絶対値をペナルティとして拡大目的関数 $\hat{f}_2(\mathbf{x})$ を構成する手法

$$\hat{f}_2(\mathbf{x}) = f(\mathbf{x}) + s_1 \sum_{l=1}^L \max\{0, g_l(\mathbf{x})\} + s_2 \sum_{m=1}^M |h_m(\mathbf{x})| \quad (2.14)$$

ただし、(2.13) 式および (2.14) 式において、 s_1, s_2 はペナルティ係数であり、 $s_1, s_2 \gg 1$ とする。(2.13) 式を用いて構成された拡大目的関数は、制約境界において関数が滑らかであるため微分可能である。しかしながら、元問題 (2.12) のある局所的最適解が制約条件の境界上に位置する場合、同じ局所的最適解に対応する拡大目的関数の局所的最適解は、元の制約条件を僅かに逸脱したところに存在する。ペナルティ係数 s_1, s_2 を十分に大きくとれば、この制約逸脱量を小さくしていくことができるが、ペナルティ係数が大きすぎると目的関数の構造を大きく崩してしまい、効率的な探索を阻害する場合がある。

一方、(2.14) 式を用いて構成された拡大目的関数は、元問題 (2.12) のある局所的最適解が制約境界上に位置する場合でも、ペナルティ係数を一定以上の値とすれば、拡大目的関数の局所的最適解を元の制約条件を完全に満足したものとすることができる。このことから (2.14) 式を用いて拡大目的関数を構成する手法を精密ペナルティ法^[44] という。精密ペナルティ法を用いた場合、制約条件境界上の最適解において目的関数値が滑らかではなくなる、つまり微分不可能となるため、目的関数の勾配を用いた最適化アルゴリズムとの相性が悪い。

Fig. 2.1 は、目的関数 $f(x) = -x^2$ に、 $-1 \leq x \leq 2$ という不等式制約条件を想定し、(2.13) 式、(2.14) 式を用いてペナルティを付与した拡大目的関数のそれぞれの概形である。この最適化問題は本来 $x = 2.0, -1.0$ という制約条件の境界上に局所的最適解を持つが、(2.13) 式によりペナルティを付与した場合は、局所的最適解がどちらもそれぞれ、わずかに制約条件の外側に位置してしまうことがわかる。


 Fig. 2.1 Augmented objective functions ($s_1 = 10$).

2.3 多点型確率的最適化アルゴリズムの数理

2.3.1 多点型確率的最適化アルゴリズムの基本構造

以降、本論文では一般の目的関数 $f: \mathbb{R}^N \rightarrow \mathbb{R}$ および制約条件関数 $\mathbf{g}: \mathbb{R}^N \rightarrow \mathbb{R}^L, \mathbf{h}: \mathbb{R}^N \rightarrow \mathbb{R}^M$ を用いて定義される非凸最適化問題

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (2.15a)$$

$$\text{subj. to } \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \quad (2.15b)$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \quad (2.15c)$$

の最適解を求める大域的最適化アルゴリズム、すなわち複数の局所最適解のなかから目的関数値が最小となるような解を提示することを目的としたアルゴリズムについて考察する。目的関数 f 、および制約条件関数 $\mathbf{g}, \mathbf{h}$ の連続性、微分可能性はいずれもとくに仮定しないが、必要に応じて連続微分可能性が仮定される場合に適用することが可能なアルゴリズムについても論じていくこととする。また、以下では最適化問題 (2.15) は、2.2.3 項で述べた外点ペナルティ法により無制約最適化問題に変換されているものとし、とくに断りのないかぎり、この場合の目的関数とは拡大目的関数を指すこととする。

最適化問題 (2.15) があたえられたとき、その最適解を解析的な手順により直接導くことができるケースは稀である。そこで、一般的な（単一の探索点による）最適化アルゴリズムでは、まず探索空間 $\mathbb{R}^N$ 内に探索点の初期位置 $\mathbf{x}(0) \in \mathbb{R}^N$ を設定し、アルゴリズムの固有の規則により探索点の位置 $\mathbf{x}$ を反復的に更新し、これらを最適解へ近接・収束させる手順をとる。ここで、初期位置 $\mathbf{x}(0)$ から k 回目までの計算の反復において、 k を探索時刻

といひ、計算の反復により得られる点列 $\mathcal{X}(k) = \{\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(k)\} = \{\mathbf{x}(\kappa)\}_{\kappa=0}^k$ を、第 k 探索時刻における探索点の探索履歴という。

最適化アルゴリズムのうち、同一探索時刻に探索空間 $\mathbf{R}^N$ 内に複数の探索点を配置し、各探索点の保持する「情報」を統合・活用して、探索点の集団的働きにより最適解を探索するものを多点型最適化アルゴリズムという。本論文では、ある探索時刻 k に探索空間 $\mathbf{R}^N$ 内に配置される各探索点の個数を $P \in \mathbf{N}$ としたとき、それぞれを $\mathbf{x}^{(1)}(k), \mathbf{x}^{(2)}(k), \dots, \mathbf{x}^{(P)}(k) \in \mathbf{R}^N$ と表記して区別する²。多点型最適化アルゴリズムにおいて、 P 個の探索点をまとめて探索集団といひ、探索時刻 k における P 個の探索点の位置 $\mathbf{x}^{(p)}(k)$ ($p = 1, \dots, P$) をまとめて、

$$\mathbf{X}(k) = (\mathbf{x}^{(1)}(k), \mathbf{x}^{(2)}(k), \dots, \mathbf{x}^{(P)}(k)) \in \mathbf{R}^{N \times P} \quad (2.16)$$

と書く。さらに、 $\mathbf{x}^{(p)}(k)$ ($p = 1, \dots, P$) に対する目的関数値 $f(\mathbf{x}^{(p)}(k))$ ($p = 1, \dots, P$) をまとめて、

$$\mathbf{f}(k) = (f(\mathbf{x}^{(1)}(k)), f(\mathbf{x}^{(2)}(k)), \dots, f(\mathbf{x}^{(P)}(k)))^\top \in \mathbf{R}^P \quad (2.17)$$

と書く。

探索点と同様に、探索集団に対しても探索履歴を考えることができる。

$\mathcal{X}(k) = \{\mathbf{X}(0), \mathbf{X}(1), \dots, \mathbf{X}(k)\} = \{\mathbf{X}(\kappa)\}_{\kappa=0}^k$ を、第 k 探索時刻における探索集団 $\mathbf{X}$ の探索履歴という。

2.3.2 エリート探索点関数

前項で述べた探索履歴・探索集団の概念を用いて、最適化アルゴリズムに構成に有用となる種々のエリート探索点関数を定義する。本項で定義するエリート探索点は既存の最適化アルゴリズムから着想を得たものであり、各項目において着想元の研究をそれぞれ引用で示す。

まず、ある単一の探索点 p に着目したエリート探索点関数として、探索時刻 k までの探索履歴 $\mathcal{X}^{(p)}(k)$ のなかで最も目的関数値が小さい座標を返す関数

$$\mathbf{x}^{(p\text{-best})}(k) = \phi(\mathcal{X}^{(p)}(k)) = \arg \min_{\mathbf{x}^{(p)}(\kappa)} \{f(\mathbf{x}^{(p)}(\kappa)) \mid \kappa = 0, \dots, k\} \quad (2.18)$$

を定義できる。(2.18) 式で定義した $\phi : \mathbf{R}^{N \times (k+1)} \rightarrow \mathbf{R}^N$ は、最適化計算アルゴリズム PSO^[21, 22] において p -best (personal best) とよばれる重要な役割をもつ。以降では (2.15) 式で定義された関数 ϕ を、PSO にならい p -best 関数とよぶこととする。

また、探索集団 $\mathbf{X}(k)$ および探索履歴 $\mathcal{X}(k)$ に対して、つぎのようなエリート探索点関数を定義できる。

(1) c-best (current best) 関数^[17]

$$\mathbf{x}^{(c\text{-best})}(k) = \alpha(\mathbf{X}(k)) = \arg \min_{\mathbf{x}^{(p)}(k)} \left\{ f(\mathbf{x}^{(p)}(k)) \mid p = 1, \dots, P \right\} \quad (2.19)$$

²本論文では、上付き添字のうち、丸括弧で囲われた数字や記号は探索点番号を表すものとし、それ以外は指数を表すものとする。

(2) c-worst (current worst) 関数^[17]

$$\mathbf{x}^{(\text{c-worst})}(k) = \beta(\mathbf{X}(k)) = \arg \max_{\mathbf{x}^{(p)}(k)} \left\{ f(\mathbf{x}^{(p)}(k)) \mid p = 1, \dots, P \right\} \quad (2.20)$$

(3) g-best (global best) 関数^[21, 22]

$$\mathbf{x}^{(\text{g-best})}(k) = \gamma(\mathcal{X}(k)) = \arg \min_{\phi(\mathcal{X}^{(p)}(k))} \left\{ f(\phi(\mathcal{X}^{(p)}(k))) \mid p = 1, \dots, P \right\} \quad (2.21)$$

(4) worst p -best (worst personal best) 関数^[45]

$$\mathbf{x}^{(\text{worst } p\text{-best})}(k) = \delta(\mathcal{X}(k)) = \arg \max_{\phi(\mathcal{X}^{(p)}(k))} \left\{ f(\phi(\mathcal{X}^{(p)}(k))) \mid p = 1, \dots, P \right\} \quad (2.22)$$

を定義することができる。これらエリート探索点関数を用いた最適化アルゴリズムの探索方策については、2.4.5 項で改めて述べることにする。

2.3.3 擬似乱数と確率的最適化

探索点 $\mathbf{x}(k)$ あるいは探索集団 $\mathbf{X}(k)$ の時間発展を決定論的に定めず、適当な確率分布にしたがう擬似乱数を用いて確率的に定めるものを、確率的最適化アルゴリズムという。擬似乱数の使われ方はアルゴリズムにより様々である。具体的な用途としては、たとえば、

- (1) 探索点座標 $\mathbf{x}^{(p)}(k)$ の更新則が差分方程式で記述されており、その係数に擬似乱数が含まれる (Particle Swarm Optimization^[21, 22])
- (2) 探索点座標 $\mathbf{x}^{(p)}(k)$ における数値微分の際、 $\mathbf{x}^{(p)}(k)$ の特定の要素をランダムに選んで微小変化幅を与える (同時摂動勾配法^[46–49])
- (3) 探索点座標 $\mathbf{x}^{(p)}(k)$ を適当な確率分布にしたがう擬似乱数で与える (LNR^[50])
- (4) 多点型最適化アルゴリズムにおいて、適当な探索点番号 p をランダムに選択する (Differential Evolution^[23, 24])
- (5) 探索点座標 $\mathbf{x}^{(p)}(k)$ 更新の際、アルゴリズムの更新則により得られた探索時刻 $k+1$ での座標を一度暫定的に $\mathbf{y}(k+1)$ とおき、 $f(\mathbf{y}(k+1)) < f(\mathbf{x}(k))$ を満たすときに限り、一定確率 $0 \leq p \leq 1$ で更新後の座標 $\mathbf{y}(k+1)$ を受理して $\mathbf{x}(k+1) = \mathbf{y}(k+1)$ とする (Simulated Annealing^[51], Differential Evolution)

といった例があげられる。いずれの用途にも共通していえるのは、探索点の挙動の正確な予測は実質的に不可能であり、アルゴリズムの動特性の解析が困難であるという点である。しかしながら、長い期間の探索点の挙動の考察を通じてアルゴリズムの性質を導くことができる例もある^[52, 53]。

2.3.4 上下限制約内部への閉じ込め

最適化問題 (2.15) を解くとき、効率的な最適解探索を目的として、最適解が上下限制約

$$D = \{ \mathbf{x} \mid a_n \leq x_n \leq b_n, n = 1, 2, \dots, N \} \quad (2.23)$$

の内部に存在すると想定し、探索範囲を D の内部に限定することがしばしばなされる。この上下限制約内部の領域 D を探索領域とよぶ。一般に、最適化アルゴリズムの反復により生成される探索点位置は、必ずしも探索領域 D の内点にとられるとは限らない。そこで、探索時刻 $k \rightarrow k+1$ において最適化アルゴリズムにより計算された新しい探索点位置を、いったん暫定位置として $\mathbf{u}(k+1)$ とおき、 $\mathbf{u}(k+1)$ の各要素 $u_1(k+1), \dots, u_N(k+1)$ に対して、適当な閉じ込め関数 $l: \mathbf{R} \rightarrow [a_n, b_n]$ を作用させたものを正式な位置 $\mathbf{x}(k+1)$ として採用する。閉じ込め関数 l の例としては、たとえば、

(1) リミッティング型関数 (limiting)

$$x = l_l(u) = \begin{cases} u, & a \leq u \leq b \\ b, & u > b \\ a, & u < a \end{cases} \quad (2.24)$$

(2) トーラス型関数 (torus)^[54]

$$x = l_t(u) = \begin{cases} u, & a \leq u \leq b \\ a + [(u - b) \bmod (b - a)], & u > b \\ b - [(a - u) \bmod (b - a)], & u < a \end{cases} \quad (2.25)$$

(3) 反射型関数 (reflection)^[55]

$$x = l_r(u) = \begin{cases} u, & a \leq u \leq b \\ b - [(u - b) \bmod (b - a)], & u > b \\ a + [(a - u) \bmod (b - a)], & u < a \end{cases} \quad (2.26)$$

などをあげることができる。Figs. 2.2–2.4 に、 $a = 0.0, b = 1.0$ とした場合の、これらの関数の概形を示す。

2.4 局所的探索方策と大域的探索方策

2.4.1 大域的最適化アルゴリズムにおける「集中化と多様化」

本節では、前節で述べた多点型確率的最適化アルゴリズムの構造を前提として、具体的に「どのように（大域的）最適解を探索するか」について、すなわち探索方策について議論する。大域的最適化アルゴリズムに期待される性質は、複数存在する局所的最適解のなかから、目的関数値が最小となる解を提示することである。その性質を実現するため、大域的最適化アルゴリズムには、

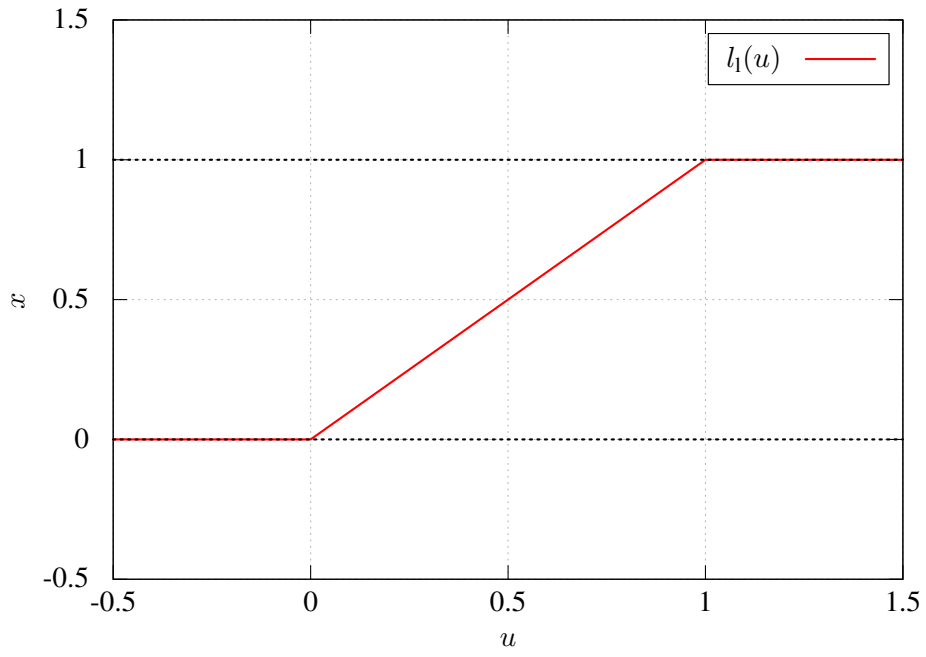


Fig. 2.2 Limiting function Eq. (2.24) ($a = 0.0, b = 1.0$).

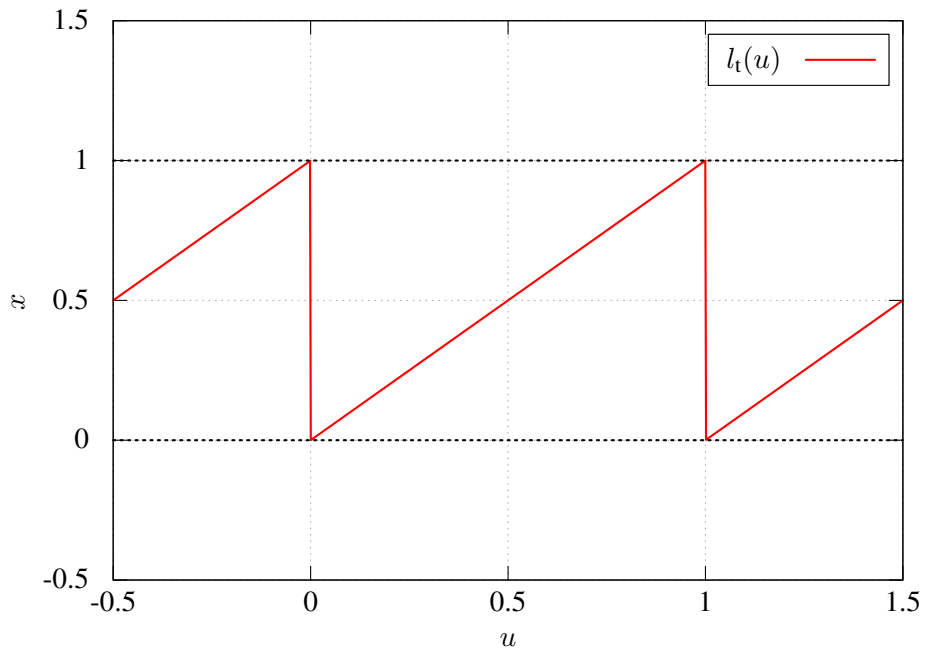
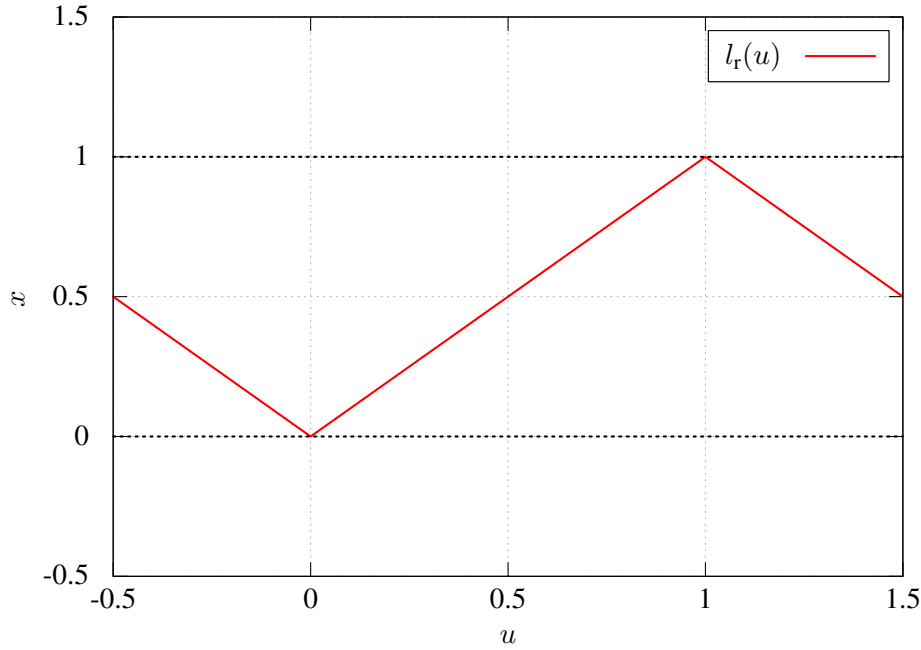


Fig. 2.3 Torus function Eq. (2.25) ($a = 0.0, b = 1.0$).


 Fig. 2.4 Reflect function Eq. (2.26) ($a = 0.0, b = 1.0$).

- 局所的探索方策

現在の探索点位置の近傍領域を探索することにより，局所的最適解へ到達するための方策

- 大域的探索方策

探索点位置の近傍領域を含む広範囲の領域をくまなく探索することにより，大域的最適解を発見するための方策

という2つの探索方策を同時に具備させる必要がある。

両者の方策はそれぞれ「集中化と多様化」とよばれており^[56]，大域的最適化アルゴリズムを構築する際の重要なキーワードとなっている。本節では，まず局所的探索方策を既存の有力な局所的最適化アルゴリズムとして概観したうえで，つぎに局所的探索方策の拡張としての大域的探索方策について述べる。なお，局所的探索方策は，目的関数の勾配や Hesse 行列などの解析的情報が利用されるが，本節では，目的関数の解析的情報が利用できない場合についても検討する。

2.4.2 局所的最適化アルゴリズムの基本構造

大域的最適化アルゴリズムにおける局所的探索方策としては，局所的最適化アルゴリズムをそのまま用いることが可能である。局所的最適化アルゴリズムの基本構造は，探索時刻 k における探索点位置を $\mathbf{x}(k)$ として，

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \alpha(k)\mathbf{d}(k) \quad (2.27)$$

で与えられる³。(2.27)式において、 $\mathbf{d}(k)$ は移動方向であり、次項以降で述べる適当なアルゴリズムにより与えられる。また $\alpha(k) > 0$ は、ステップサイズとよばれるパラメータである。局所的最適化では、まず移動方向 $\mathbf{d}(k)$ を適当な手法により定め、つぎに2.4.6項で述べる直線探索法とよばれるサブルーチンを用いてステップサイズ $\alpha(k)$ を定める構造となる。局所的最適化アルゴリズムでは、

$$f(\mathbf{x}(k+1)) = f(\mathbf{x}(k) + \alpha(k)\mathbf{d}(k)) < f(\mathbf{x}(k)) \quad (2.28)$$

が微小な $\alpha(k) > 0$ に対して成立するように移動方向 $\mathbf{d}(k)$ を定める。(2.28)式が成立するとき $\mathbf{d}(k)$ を降下方向とよぶ。

2.4.3 局所的最適化アルゴリズムの収束性

ある局所的最適化アルゴリズム $\mathcal{A}$ があたえられたとき、局所的最適解への収束に関して、以下の2つの性質が議論される。

大域的収束性 任意の位置を初期値として局所的最適化アルゴリズム $\mathcal{A}$ を実行し、その結果得られる探索履歴 $\mathcal{X}(k) = \{\mathbf{x}(\kappa)\}_{\kappa=0}^k$ の集積点がいずれかの局所的最適解であることが保証されるとき、 $\mathcal{A}$ は大域的収束性をもつという⁴。

局所的収束性 局所的収束性は、局所的最適解近傍での探索点の収束の様子を定量的に示す概念であり、おもに1次収束性、超1次収束性、2次収束性の三種類があげられる。以下、局所的最適化アルゴリズム $\mathcal{A}$ の実行により得られる探索履歴を $\mathcal{X}(k) = \{\mathbf{x}(\kappa)\}_{\kappa=0}^k$ 、その集積点である局所的最適解を $\mathbf{x}^*$ とする。

(1) 1次収束性

$$\|\mathbf{x}(k+1) - \mathbf{x}^*\| \leq \beta \|\mathbf{x}(k) - \mathbf{x}^*\|, \forall k > K \quad (2.29)$$

を満たす定数 $0 < \beta < 1$ と有限の自然数 K が存在するとき、局所的最適化アルゴリズム $\mathcal{A}$ は1次収束性をもつという。

(2) 超1次収束性

(2.29)式において、 β が探索時刻 k に依存し、 $\lim_{k \rightarrow \infty} \beta(k) = 0$ 、すなわち

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}(k+1) - \mathbf{x}^*\|}{\|\mathbf{x}(k) - \mathbf{x}^*\|} = 0 \quad (2.30)$$

を満たすとき、局所的最適化アルゴリズム $\mathcal{A}$ は超1次収束性をもつという。

(3) 2次収束性

$$\|\mathbf{x}(k+1) - \mathbf{x}^*\| \leq \beta \|\mathbf{x}(k) - \mathbf{x}^*\|^2, \forall k > K \quad (2.31)$$

を満たす定数 $0 < \beta < 1$ と有限の自然数 K が存在するとき、局所的最適化アルゴリズム $\mathcal{A}$ は2次収束性をもつという。

³本項で述べる局所的最適化方策は、いずれも単一探索点の位置の更新則として議論するものであり、探索点番号に関する添字は省略している。当然、本項の議論をそのまま多点型最適化手法に拡張することも可能である。

⁴大域的最適解へ収束する性質ではないことに注意されたい。

2.4.4 目的関数の解析的情報を用いた局所的最適化アルゴリズム

目的関数 $f(\mathbf{x})$ が連続微分可能である場合、以下で述べるように、 $f(\mathbf{x})$ の解析的情報である勾配や Hesse 行列を活用した効果的な降下方向 $\mathbf{d}(k)$ の生成手段が知られている。

最急降下法 目的関数 $f(\mathbf{x})$ の、 $\mathbf{x} = \mathbf{x}(k)$ を中心とした 1 次の項までの Taylor 展開

$$\bar{f}(\mathbf{x}) = f(\mathbf{x}(k)) + \nabla f(\mathbf{x}(k))\mathbf{d}(k) \quad (2.32)$$

を考える。 (2.32) 式において、 $\bar{f}(\mathbf{x})$ の最急降下方向をあたえる移動方向 $\mathbf{d}(k)$ は、

$$\mathbf{d}(k) = -\nabla^\top f(\mathbf{x}(k)) \quad (2.33)$$

で与えられる。 $\mathbf{d}(k)$ を (2.33) 式により定める方法を最急降下法という。ステップサイズ $\alpha(k) > 0$ が十分に小さく、かつ $\nabla f(\mathbf{x}(k)) = \mathbf{0}$ でないとき、

$$f(\mathbf{x}(k+1)) = f(\mathbf{x}(k)) - \alpha(k)\|\mathbf{d}(k)\|^2 < f(\mathbf{x}(k)) \quad (2.34)$$

が成立し、移動方向 $\mathbf{d}(k)$ は降下方向となる。

最急降下法は、直線探索による適切なステップサイズ設定のもとで大域的収束性が保証されており、局所的最適解への収束率は 1 次収束であることが知られている。

Newton 法 目的関数 $f(\mathbf{x})$ が 2 階微分可能であると仮定したもとで、 $\mathbf{x} = \mathbf{x}(k)$ を中心とした 2 次の項までの Taylor 展開

$$\bar{f}(\mathbf{x}) = f(\mathbf{x}(k)) + \nabla f(\mathbf{x}(k))\mathbf{d}(k) + \frac{1}{2}\mathbf{d}^\top(k)\nabla^2 f(\mathbf{x}(k))\mathbf{d}(k) \quad (2.35)$$

を考える。(2.35) 式において、Hesse 行列 $\nabla^2 f(\mathbf{x}(k))$ が正定値行列となるとき、 $\bar{f}(\mathbf{x})$ を最小化する移動方向

$$\mathbf{d}^*(k) = (\nabla^2 f(\mathbf{x}(k)))^{-1} \nabla^\top f(\mathbf{x}(k)) \quad (2.36)$$

が存在する。 $\mathbf{d}(k)$ を (2.36) 式により定める方法を Newton 法という。Newton 法を用いた場合の局所的最適解への収束率は 2 次収束であり、きわめて高速な局所的最適化アルゴリズムとして知られている。しかしながら、大域的収束性は保証されていないため、目的関数の形状によっては局所的最適解へ収束しない場合もある。

(2.36) 式では、移動量まで含めた移動方向 $\mathbf{d}(k)$ の最適解を求めており、ステップサイズ α は 1.0 としてよいが、目的関数の減少性を保証するため、最急降下法と同様ステップサイズを別途定めてもよい。ただしこの場合でも、目的関数の Hesse 行列の正定値性は必要である。

準 Newton 法 Newton 法はきわめて速い局所的収束性をもつが、Newton 法を用いて最適解を得るためには、Hesse 行列の正定値性が保証されている必要があり、また、各探索時刻において Hesse 行列およびその逆行列を計算する必要があるため、とくに大規模な問題においては多大な計算コストが必要となる。

準 Newton 法とは、2 次収束性と引き換えに上述した Newton 法の欠点を改良した手法であり、(2.36) 式における Hesse 行列あるいはその逆行列を、目的関数の 1 階微分の情報までを用いて逐次近似する手法の総称である。

以下、 $(\nabla^2 f(\mathbf{x}(k)))^{-1}$ の近似行列を $\mathbf{H}(k)$ とおき、また、

$$\mathbf{s}(k) = \mathbf{x}(k+1) - \mathbf{x}(k) \quad (2.37a)$$

$$\mathbf{y}(k) = \nabla^\top f(\mathbf{x}(k+1)) - \nabla^\top f(\mathbf{x}(k)) \quad (2.37b)$$

とおく。ここでは詳細な導出や理論は本論文では割愛し、比較的よく用いられる 2 つの Hesse 逆行列近似公式を示すにとどめる。

(1) DFP (Davidon-Fletcher-Powell) 公式^[57, 58]

$$\mathbf{H}(k+1) = \mathbf{H}(k) + \frac{\mathbf{y}(k)\mathbf{y}^\top(k)}{\mathbf{s}^\top(k)\mathbf{y}(k)} - \frac{\mathbf{H}(k)\mathbf{y}(k)\mathbf{y}^\top(k)\mathbf{H}(k)}{\mathbf{y}^\top(k)\mathbf{H}(k)\mathbf{y}(k)} \quad (2.38)$$

(2) BFGS (Broyden-Fletcher-Goldfarb-Shanno) 公式^[59–63]

$$\begin{aligned} \mathbf{H}(k+1) = \mathbf{H}(k) + & \left(1 + \frac{\mathbf{y}^\top(k)\mathbf{H}(k)\mathbf{y}(k)}{\mathbf{s}^\top(k)\mathbf{y}(k)}\right) \frac{\mathbf{y}(k)\mathbf{y}^\top(k)}{\mathbf{s}^\top(k)\mathbf{y}(k)} \\ & - \frac{\mathbf{H}(k)\mathbf{y}(k)\mathbf{s}^\top(k) + \mathbf{s}(k)^\top\mathbf{y}(k)\mathbf{H}(k)}{\mathbf{s}^\top(k)\mathbf{y}(k)} \end{aligned} \quad (2.39)$$

なお、近似 Hesse 逆行列 $\mathbf{H}$ の初期値 $\mathbf{H}(0)$ は単位行列とするのが通例である。

準 Newton 法は、直線探索による適切なステップサイズ設定のもとで大域的収束性が保証されており、局所的最適解への収束率は超 1 次収束であることが知られている。上記のような良い性質から、準 Newton 法は実用上きわめて重要かつ有効な手法となっており、制約条件付き最適化手法としても、準 Newton 法をもとにして逐次 2 次計画法^[13] や信頼領域法^[14] が有効な手法として確立されており、市販の最適化ソルバにも採用されている⁵。

2.4.5 目的関数の解析的情報を用いない局所的最適化アルゴリズム

前項では、目的関数の解析的情報である勾配と Hesse 行列を用いて降下方向を生成する手法を述べたが、一般の最適化問題 (2.15) において、勾配や Hesse 行列は必ずしも利用できるとは限らない。そこで本項では、目的関数の値のみを用いて局所的探索を実現する方策を述べる⁶。2.5 節で述べるメタヒューリスティクスは、目的関数の勾配や Hesse 行列を利用しないため幅広い最適化問題に対して適用可能である。それを実現するための探索方策として、とくに本項で述べるエリート探索点関数を用いた方法が利用される。目的関数の値のみを用いた局所的探索方策は、大きく分けて、

- (1) 探索履歴や探索集団の情報を用い、既得の目的関数値の小さい領域、あるいは目的関数値が小さいと期待される領域に探索点を移動させる方法

⁵たとえば、MATLAB (<http://www.mathworks.co.jp/products/matlab/>) や Numerical Optimizer (<http://www.msi.co.jp/nuopt/>) など (いずれも 2014 年 9 月現在)。

⁶目的関数の記述が複雑であり、勾配や Hesse 行列の計算が困難であるというだけであるならば、たとえば自動微分^[64]を用いた方法によりこれらを利用することが可能となる。

- (2) 探索履歴や探索集団の情報を用い、また必要に応じて適宜追加計算をおこない、目的関数の勾配を近似的に計算する方法

の2種類に分類することができる。

以下で述べる手法のうち、エリート探索点関数を用いた方法は前者、探索集団情報を用いた勾配近似法は後者にそれぞれ分類される。なお、本論文では直接取り扱わないが、前者に類される手法としては古典的な局所的最適化アルゴリズムである Nelder-Mead 法^[17]があり、後者に分類される手法としては、同時摂動勾配法^[46-49]や数値微分があげられる。

エリート探索点関数を用いた方法 2.3.2 項で述べたエリート探索点関数を用いて、たとえば移動方向をつぎのように生成することができる。ただし、これらの方法で生成される移動方向が必ずしも降下方向となるとは限らないことに注意されたい。

- (1) p -best への移流

$$\mathbf{d}_{p\text{-best}}^{(p)}(k) = \mathbf{x}^{(p\text{-best})}(k) - \mathbf{x}^{(p)}(k) \quad (2.40)$$

- (2) g -best への移流

$$\mathbf{d}_{g\text{-best}}^{(p)}(k) = \mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k) \quad (2.41)$$

- (3) c -best への移流

$$\mathbf{d}_{c\text{-best}}^{(p)}(k) = \mathbf{x}^{(c\text{-best})}(k) - \mathbf{x}^{(p)}(k) \quad (2.42)$$

- (4) 重み付き p -best への移流^[45]

$$\mathbf{d}_{\text{weighted } p\text{-best}}^{(p)}(k) = \frac{\sum_{p \in S_P} \omega^{(p)}(k) \mathbf{x}^{(p\text{-best})}}{\sum_{p \in S_P} \omega^{(p)}(k)} - \mathbf{x}^{(p)}(k) \quad (2.43a)$$

$$\text{where } \omega^{(p)}(k) = (\mathbf{x}^{(\text{worst } p\text{-best})}(k) - \mathbf{x}^{(p\text{-best})}(k))^\gamma, \quad p \in S_P, \quad \gamma > 0 \quad (2.43b)$$

- (5) 重み付き c -best への移流

$$\mathbf{d}_{\text{weighted } c\text{-best}}^{(p)}(k) = \frac{\sum_{p \in S_P} \omega^{(p)}(k) \mathbf{x}^{(p)}}{\sum_{p \in S_P} \omega^{(p)}(k)} - \mathbf{x}^{(p)}(k) \quad (2.44a)$$

$$\text{where } \omega^{(p)}(k) = (\mathbf{x}^{(c\text{-worst})}(k) - \mathbf{x}^{(p)}(k))^\gamma, \quad p \in S_P, \quad \gamma > 0 \quad (2.44b)$$

なお、(2.43) 式および (2.44) 式において、 S_P は $\{1, \dots, P\}$ のなかの 1 つ以上の数字を含む添字集合であり、たとえば、「自身以外の探索点の重み付き平均」などのバリエーションを含んで表現するためにこのような定式化としている。

探索集団情報を用いた勾配近似法 多点型最適化アルゴリズムにおいて、ある探索時刻 k に P 個の探索点がすべて $\mathbf{a} \in \mathbb{R}^N$ の近傍に密集しているとき、 $\mathbf{a}$ における勾配 $\nabla f(\mathbf{a})$ を、探索集団行列 $\mathbf{X}(k)$ とその目的関数値ベクトル $\mathbf{f}(k)$ を用いて近似的に求める方法を与える。まず、目的関数 $f(\mathbf{x})$ を $\mathbf{x}^a$ の近傍で1次の項まで Taylor 展開すると、

$$f(\mathbf{a} + \mathbf{d}) \simeq f(\mathbf{a}) + \nabla f(\mathbf{a}) \mathbf{d} \quad (2.45)$$

を得る。ここで、探索点 $p = 1, \dots, P$ に対して、

$$\mathbf{d}^{(p)} = \mathbf{x}^{(p)}(k) - \mathbf{a} \quad (2.46)$$

とおき、また近似勾配ベクトルを $\nabla f(\mathbf{a}) = \mathbf{g}^\top \in \mathbb{R}^N$ とおき、 P 個の式からなる N 元の連立方程式

$$f(\mathbf{x}^{(p)}(k)) \simeq f(\mathbf{a}) + \mathbf{g}^\top (\mathbf{x}^{(p)}(k) - \mathbf{a}) \quad (2.47)$$

を解くことによって $\mathbf{g}^\top$ を定める。連立方程式 (2.47) は、

$$\mathbf{f}^\top(k) - \mathbf{1}^\top f(\mathbf{a}) = \mathbf{g}^\top (\mathbf{X}(k) - \mathbf{1}^\top \mathbf{a}) \quad (2.48)$$

と書ける。連立方程式 (2.48) を満たす唯一の解が存在するのは、行列 $\mathbf{A} = (\mathbf{X}(k) - \mathbf{1}^\top \mathbf{a})$ に対して $\text{rank } \mathbf{A} = N$ のときに限り、 $\text{rank } \mathbf{A} < N$ であれば連立方程式 (2.48) を満たす解は無数に存在し、 $\text{rank } \mathbf{A} > N$ であれば連立方程式 (2.48) を満たす解は存在しない。そこで、連立方程式 (2.48) を厳密に満たす解を求めるのではなく、右辺と左辺の誤差自乗和関数

$$e(\mathbf{g}^\top) = \left\| \mathbf{f}^\top(k) - \mathbf{1}^\top f(\mathbf{a}) - \mathbf{g}^\top (\mathbf{X}(k) - \mathbf{1}^\top \mathbf{a}) \right\|^2 \quad (2.49)$$

を定義し、最小自乗問題

$$\min_{\mathbf{g}^\top} e(\mathbf{g}^\top) \quad (2.50)$$

を解き、 $\mathbf{g}^\top$ を定める。最小自乗問題 (2.50) は解析的にとくことができ、(2.49) 式を最小化する近似勾配ベクトル $\mathbf{g}^\top$ は、

$$\mathbf{g}^\top(k) = ((\mathbf{X}(k) - \mathbf{1}^\top \mathbf{a})^\dagger (\mathbf{f}(k) - \mathbf{1} f(\mathbf{a})))^\top \quad (2.51)$$

で与えられる。

(2.51) 式で求まる近似勾配ベクトル $\mathbf{g}^\top$ は、行列 $\mathbf{A} = (\mathbf{X}(k) - \mathbf{1}^\top \mathbf{a})$ にランクに応じてつぎのような性質をもつ。

- (1) $\text{rank } \mathbf{A} = N$ のとき、自乗誤差 $e(\mathbf{g}^\top)$ がゼロとなる $\mathbf{g}^\top$ が唯一の解が求まる。
- (2) $\text{rank } \mathbf{A} < N$ のとき、自乗誤差 $e(\mathbf{g}^\top)$ がゼロとなる $\mathbf{g}^\top$ のうち、ノルム $\|\mathbf{g}^\top\|$ が最小となる解が求まる。
- (3) $\text{rank } \mathbf{A} > N$ のとき、自乗誤差 $e(\mathbf{g}^\top)$ を最小化する $\mathbf{g}^\top$ が求まる。

本法を用いた勾配近似において、 $\mathbf{a} \in \mathbb{R}^N$ の定め方は任意である。探索集団のなかから目的関数値を計算済みの既得の座標を選んでもよく、探索集団の分布をもとに、たとえば全探索点の位置の平均などを用いて新たに選んでもよい。ただし、新たに選ぶ場合は、その座標に対応する目的関数 $f(\mathbf{a})$ の計算が必要となる。

2.4.6 直線探索法

前節までは、移動ベクトルの定め方について議論していたが、局所的探索アルゴリズムにおいてはステップサイズの適切な設定も重要であり、うまく移動ベクトルを決定しても、ステップサイズの設定が適切でない場合は、効率の悪い探索となったり、目的関数値の減少が実現されない場合もある。実用的なステップサイズの調整方法として、

- (1) ステップサイズを十分に小さい定数に設定し、目的関数値の減少性を保証することを優先する。
- (2) ステップサイズ α を設計変数とする 1 次元の最適化問題を各探索時刻において解き、 α を動的に設定する。

の 2 つの方法があげられる。後者の方法により最適なステップサイズを求める方法を直線探索法という。直線探索法では、つぎの 1 次元の最適化問題

$$\min_{\alpha(k)} f(\mathbf{x}(k) + \alpha(k)\mathbf{d}(k)) \quad (2.52)$$

を各探索時刻 k ごとに解く。直線探索法を採用する場合、最適なステップサイズ $\alpha(k)$ を都度求めることにより、探索時刻で見た場合の探索効率は向上するが、一方で各探索時刻ごとに別の最適化問題を解くこととなるため、計算コストの増加には注意を払う必要がある。直線探索法には、以下で示すように厳密な直線探索とラフな直線探索の 2 種類が存在する。

厳密な直線探索 厳密な直線探索は、1 次元の最適化問題 (2.52) の厳密な最適解を求める手法である。具体的なアルゴリズムとしては、たとえば二分法や黄金分割法^[65]を用いたものが考えられる。また、目的関数 $f(\mathbf{x})$ の勾配情報を利用できるのであれば、最適性条件として、

$$(\nabla f(\mathbf{x}(k) + \alpha(k)\mathbf{d}(k))) \mathbf{d}(k) = 0 \quad (2.53)$$

を与えることができる。

ラフな直線探索 直線探索の目的は、1 回の計算の反復における目的関数値減少量を可能な限り大きくすることであって、必ずしも厳密に最適なステップサイズを求める必要はない⁷。そこで、ラフな直線探索として、少ない計算量で十分な目的関数値減少を実現するため、たとえば Table 2.1 ようなアルゴリズムが考えられる。

2.4.7 大域的探索方策

大域的探索方策は大きく分けて、

⁷本論文では詳細を省略しているが、準 Newton 法において近似 Hesse 逆行列 H の正定値性を保証するためには、各探索時刻で厳密に最適なステップサイズが求められている必要がある。しかしながら、準ニュートン法の適用にあたりラフな直線探索を用いた場合でも、 H の正定値性を都度確認し、必要に応じて正定値性が保たれるよう補正するような機構を設けておけば、実用上問題になることは少ない。

Table 2.1 Example of Linear Search Algorithm

-
- Step 1** $0 < \beta < 1, \alpha_{\text{init}} > \alpha_{\text{term}} > 0$ を適当に定めておく. 降下方向 $\mathbf{d}(k)$ および目的関数値 $f(\mathbf{x}(k))$ を計算しておき, 暫定解として $\bar{f} = f(\mathbf{x}(k)), \bar{\alpha} = 0$ を保存しておく. また, $h = 0$ とおく.
- Step 2** $f = f(\mathbf{x}(k) + \alpha_{\text{init}}\beta^h \mathbf{d}(k))$ を計算する. $f < \bar{f}$ であれば **Step 3** へ進み, $f \geq \bar{f}$ であれば **Step 4** へ進む.
- Step 3** $\bar{f} \leftarrow f, \bar{\alpha} \leftarrow \alpha_{\text{init}}\beta^h$ として **Step 5** へ進む.
- Step 4** 最適ステップサイズとして $\bar{\alpha}$ を出力してアルゴリズムを終了する.
- Step 5** $h \leftarrow h + 1$ とする. $\alpha_{\text{init}}\beta^h < \alpha_{\text{terminal}}$ であれば直線探索失敗とみなし, 最適ステップサイズとして 0 を出力してアルゴリズムを終了する. それ以外の場合は **Step 2** へ進む.
-

- (1) 局所的探索の試行回数を増やす, 探索点数を増やす, あるいは初期位置のバリエーションを増やすことで探索領域を広くする方法
- (2) 局所的探索により生成される移動方向を用いつつも, 目的関数値が一時的に増加することも許容し, ステップサイズをあえて大きくとるなどする方法

の2種類に分けられると考えられる. (1) の例としては, たとえばマルチスタート法^[66, 67]や, 多点型最適化手法全般があげられる. (2) の例としては, 離散化勾配カオスモデル^[68–70]やその改良手法である水抜き法^[16]があげられる.

大域的探索方策の有効性は目的関数の構造に大きく依存するため, 局所的探索方策と比較して, 数学的に議論され構築された探索方策は数少ない. 大域的最適化方策の実現にあたって重要となるのは「特定の局所的最適解へとらわれる」ことをいかに回避するかという点であるとされ, メタヒューリスティクスの改良研究で多く見られるように, 局所的最適解から脱出して大域的な探索を続ける方法に関して, さまざまなアイデアが提案されている. 詳細については2.5節で再度述べることにする.

2.4.8 局所的探索と大域的探索の位置づけ

本節では, 局所的探索方策と大域的探索方策を分けて述べたが, これらの大域的最適化アルゴリズムのなかでの位置づけについては, 大別してつぎの3つの考え方が存在する.

- (1) 探索状況に応じて, あるいは確率的方策を用いて, 同じダイナミクスのもとで局所的探索と大域的探索の双方を実現できるようにする.
- (2) パラメータ調整等により, 大域的探索から徐々に局所的探索へシフトさせていくような方策とする.

(3) 大域的最適化アルゴリズムのなかで、「局所的探索」というフェーズと「大域的探索」という2つのフェーズに明示的に分割する。

(1) の例としては、たとえば PSO をあげることができる。PSO は、探索のおもな駆動力として (2.40) 式および (2.41) 式という、エリート探索点に対する移流項を用いているが、これらの移流項に対して乱数が付与されていることにより、局所的探索（小さい乱数が与えられた場合）と大域的探索（大きい乱数が与えられた場合）の両方を実現できる構造となっている。(2) の例としては、Simulated Annealing や離散化勾配カオスモデルがあげられる。いずれも、大域的探索と局所的探索は同一のダイナミクスを用いておこなわれるアルゴリズムであり、これらを制御するダイナミクス内のパラメータを経時的に調整することで、探索初期には大域的探索をおこない、探索が進むにしたがい局所的探索へ推移するといった方策をとる。この過程は、上述した Simulated Annealing にならい「徐冷 (Annealing)」とよばれ、これらの手法に限らず、さまざまな大域的最適化アルゴリズムに応用されている。(3) の例としては、たとえば大域的探索の各過程でその都度局所的最適化をおこなう手法が考案されており、文献 [55] ではこの手法が一種の目的関数の変換手法に相当すると指摘している。

2.5 メタヒューリスティクス

2.5.1 多点型確率的最適化アルゴリズムとしてのメタヒューリスティクス

本節では、実装の容易さや、ベンチマーク問題への適用を通して検証された大域的最適化性能などから近年着目を浴びている最適化アルゴリズムである Particle Swarm Optimization と Differential Evolution を取りあげ、これらのアルゴリズムを概説したうえで、実問題への応用状況や、手法の改良研究についてまとめる。これらのアルゴリズムはいずれも「メタヒューリスティクス」[56, 71, 72] とよばれる計算パラダイムに属する手法としてもよく知られている。メタヒューリスティクスとは、問題に対する事前知識を必要せず、幅広い問題に適用可能なアルゴリズムの総称であり、具体的には、目的関数の勾配情報を利用せず、目的関数の値のみから最適化計算が可能である点がしばしば強調される。多くのメタヒューリスティクスでは、目的関数の勾配情報を利用しない代わりに、探索の多様性を確保する手段として、「多点による探索」「確率的探索」を備えることが多く、これらの特徴の両方を備えるアルゴリズムは、多点型確率的最適化アルゴリズムとみなすことができる。本論文では、同じアルゴリズムを指しつつも、注目している性質に応じて、アルゴリズムの適用可能範囲や、アルゴリズムがもともと属する計算パラダイムに着目する場合はメタヒューリスティクスとよび、具体的なアルゴリズムの実装方法や数学的構造に着目する場合は多点型確率的最適化アルゴリズムとよぶこととする。

2.5.2 Particle Swarm Optimization

Particle Swarm Optimization (以下 PSO) は、魚や鳥の群れが餌場を探す社会的行動に着想を得たメタヒューリスティクスであり、その探索点位置更新式は、

$$\mathbf{v}^{(p)}(k+1) = \lambda \mathbf{v}^{(p)}(k) + c_1 \mathbf{R}_1^{(p)}(k)(\mathbf{x}^{(p\text{-best})}(k) - \mathbf{x}^{(p)}(k))$$

$$+c_2\mathbf{R}_2^{(p)}(k)(\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k)) \quad (2.54a)$$

$$\mathbf{x}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k+1) \quad (2.54b)$$

で与えられる。ただし、 $k = 0, 1, \dots$ は探索時刻、上付き添字 $p = 1, \dots, P$ は探索点番号をそれぞれ意味する。 $\mathbf{R}_1^{(p)}(k), \mathbf{R}_2^{(p)}(k)$ はいずれも対角行列であり、各対角要素は実数値一様分布 $\mathcal{U}_R(0, 1)$ にしたがう乱数によって、探索時刻、探索点、および探索空間の次元ごとにそれぞれ独立して決定される。(2.54a) 式において、 $\mathbf{x}^{(p\text{-best})}(k)$ および $\mathbf{x}^{(g\text{-best})}(k)$ は、それぞれ (2.18) 式および (2.21) 式で定義される p -best, g -best である。PSO の最適化性能は、パラメータである λ, c_1, c_2 に大きく依存する。よく使われるパラメータ設定例としては、たとえば $\lambda = 0.729$, $c_1 = c_2 = 1.49455$ ^[73], $\lambda = 0.6$, $c_1 = c_2 = 1.7$ ^[74] がある。

PSO は、簡便なアルゴリズムでありながらも、目的関数の勾配情報を利用せず、かつ優良な大域的最適化性能をもつことから多く注目を集めており、実問題への応用研究も盛んである。一例としては、モデル予測制御のパラメータ最適化^[75] や、電力系統の無効電力配分計画問題^[76, 77]、吸着冷凍機の性能向上^[78]、太陽光発電における最大電力点追従^[79]、機械学習^[80, 81]、クラウドコンピューティングにおける計算資源割当て^[82] などがあげられる。いずれも、目的関数が多峰性である、あるいは勾配情報が利用できないなど、既存の目的関数の解析的情報を用いた局所的最適化アルゴリズムの適用が難しい問題であり、こういった問題に対して PSO は成果をあげている。

一方で、いったん特定の局所的最適解近傍へすべての探索点が収束してしまうと、探索が完全に停止してしまい、大域的探索が継続できなくなるという課題が指摘されており、アルゴリズムの改良研究も数多く存在する。おもな改良アプローチとしては以下のようなものがあげられる。

- (1) パラメータの適切な設定により持続的な探索の実現を図るもの^[52, 53]
- (2) パラメータを探索状況に合わせて動的に変化させるもの^[29, 73, 83],
- (3) g -best の情報を保持しつつ大域的探索から局所的探索への推移を繰り返しおこなうもの^[84, 85]
- (4) 探索点群を複数のグループに分割して情報交換のトポロジーを変化させるもの^[86–88]
- (5) 現実の魚や鳥の群れの行動特性をより現実に近い形でモデリングするもの^[89]
- (6) p -best, g -best への移流項以外の新たな項を付け加える、あるいは他の最適化手法とのハイブリッドにより探索方策の拡充をはかるもの^[90–94]

2.5.3 Differential Evolution

Differential Evolution (以下 DE) は、いわゆる進化的アルゴリズムの発展として確立された大域的最適化アルゴリズムであり、その探索点位置更新式は、

$$\mathbf{v}^{(p)}(k+1) = \mathbf{x}^{(r_1^{(p)}(k))}(k) + c_f \left(\mathbf{x}^{(r_2^{(p)}(k))}(k) - \mathbf{x}^{(r_3^{(p)}(k))}(k) \right) \quad (2.55a)$$

$$u_n^{(p)}(k+1) = \begin{cases} v_n^{(p)}(k+1), & (\theta^{(p)}(k) < \lambda) \cup (n = d^{(p)}(k)) \\ x_n^{(p)}(k), & \text{otherwise} \end{cases} \quad (2.55b)$$

$$\mathbf{x}^{(p)}(k+1) = \begin{cases} \mathbf{u}^{(p)}(k+1), & f(\mathbf{u}^{(p)}(k+1)) < f(\mathbf{x}^{(p)}(k)) \\ \mathbf{x}^{(p)}(k), & \text{otherwise} \end{cases} \quad (2.55c)$$

で与えられる。ただし、 $k = 0, 1, \dots$ は探索時刻，上付き添字 $p = 1, \dots, P$ は探索点番号，下付き添字 $n = 1, \dots, N$ は探索空間の次元をそれぞれ意味する。 $r_1^{(p)}(k), r_2^{(p)}(k), r_3^{(p)}(k)$ は、着目する探索点番号を p として、 $p, r_1^{(p)}(k), r_2^{(p)}(k), r_3^{(p)}(k)$ がいずれも異なる値をとるように整数値一様分布 $\mathcal{U}_{\mathbb{N}}(1, P)$ にしたがう乱数により決定する。また、 $\theta^{(p)}(k)$ は実数値一様分布 $\mathcal{U}_{\mathbb{R}}(0, 1)$ に、 $d^{(p)}$ は整数値一様分布 $\mathcal{U}_{\mathbb{N}}(1, P)$ にしたがう乱数でそれぞれ決定される。 c_f および λ は DE の最適化性能を担うパラメータであり、それぞれスケールリングファクタ、交叉率とよばれる。

DE の探索点位置更新式 (2.55) の各プロセスには、進化計算に由来する名称がつけられており、(2.55a) 式は「突然変異 (mutation)」(2.55b) 式は「交叉 (crossover)」，(2.55c) 式は「選択 (selection)」とよばれている。また、DE の探索の主たる駆動力は (2.55a) 式の右辺第 2 項の $\mathbf{x}^{(r_2^{(p)}(k))}(k) - \mathbf{x}^{(r_3^{(p)}(k))}(k)$ であり、このベクトルは「差分ベクトル」とよばれている。

DE は提案当初よりさまざまなバリエーションアルゴリズムが考案されており、各バリエーションアルゴリズムは「DE/base/num/crossover」と表記され区別される。ここで *base* は、(2.55a) 式における探索点番号の選び方であり、ランダムに探索点番号を選択する「rand」のほか、目的関数値のもっとも小さい位置に存在する探索点番号を選択する「best」が存在する。*num* は、(2.55a) 式における差分ベクトルの数である。*crossover* は、(2.55b) 式における交叉方法であり、ベクトル $\mathbf{v}^{(p)}(k+1)$ と $\mathbf{x}^{(p)}(k)$ の全要素を独立に交差させる「bin (binary)」のほか、いくつかの番号の連続した要素のみを交差させる「exp (exponential)」がある。(2.55) 式に示した探索点位置更新式は DE/rand/1/bin とよばれる、DE のもっとも基本的なアルゴリズムである。

DE は PSO と同様、確率的かつ多点の大域的最適化手法であるが、PSO は探索点の位置更新に制限がないのに対して、DE では (2.55c) 式的作用により、各探索点はつねに目的関数値の小さくなる方向にしか移動しない。換言すれば、DE の探索点はつねに p -best の位置にあるといえる。したがって、DE では探索点位置が発散することはないため、パラメータ設定に対して最適化性能が頑健である。この性質に着目した実問題への応用研究としては、たとえば、多目的貯水槽の最適運用計画問題への適用^[95]や、発電機のゲイン調整^[96]、複数の三次元形状計測データの位置合わせ^[97]、マルチバンドマイクロ波フィルタの設計への適用^[98]、化合物の分子配列の決定^[99]などがあげられる。また、最適化性能の頑健性が高いことから、市販の最適化ソルバの大域的最適化アルゴリズムとしても採用されている⁸。

DE の抱える課題のひとつとして、探索空間の次元ごとの交叉を積極的に活用するため、問題の座標の取り方に最適化性能が大きく左右され、とくに変数間依存の強い問題に対して相性が悪いという点がある。これに対して、たとえば回転不変性を有する交叉の導入

⁸Maple (<http://www.cybernet.co.jp/maple/>) など (2014 年 9 月現在)。

[100, 101], Nelder-Mead 法に基づく降下方向の概念の組み込み [102] などが試みられている。なお、この課題は、探索空間の次元ごとに異なる乱数を用いる PSO においても同様の指摘がされており、文献 [52] では、すべて探索空間の次元で同じ乱数係数を用いることで、座標の取り方による最適化性能への影響が緩和されることが実験的に示されている。

その他、DE の最適化性能の向上を図った研究として、パラメータの自動調整に関する研究 [30–32] や、目的関数計算回数の削減に関する研究 [103] などがあげられる。

2.5.4 その他の手法

本節では、代表的なメタヒューリスティクスの例として PSO と DE を取り上げたが、このほかにもたとえば Cucko Search Algorithm [104], FireFly Algorithm [105], Artificial Bee Algorithm [106, 107] などが存在する。これらの多くは自然界における何かしらの最適化過程を模倣したものであり、文献 [108] では、さらに細かい分類や各アルゴリズムの類似性の指摘がなされている。

2.5.5 メタヒューリスティクスの課題

PSO や DE をはじめとするメタヒューリスティクスは、アルゴリズムが単純であり、かつ目的関数の勾配情報を利用せずに大域的最適化を試みることができるため、きわめて汎用性の高いアルゴリズム（群）であるといえる。また、問題設定に応じて、アルゴリズムを自由に改良できる柔軟さを持ち合わせている。しかしその一方で、アルゴリズムの構築に際して数学的な裏付けは必ずしもなく、また多点型確率的最適化アルゴリズムの形態をとるため、複数の探索点間の相互作用や確率的挙動といった数学的に取り扱いにくい要素を多分に含むため、動特性の把握やパラメータ調整に関してアルゴリズムを利用する人間が試行錯誤する必要がある。これに対して、たとえば、パラメータを探索状況に応じて自動的に調整する機能をもつアルゴリズムの研究 [29, 32] や、アルゴリズムの数学的挙動を調査する研究 [109] などがあり、研究の大きな潮流として、より実践的で使いやすいアルゴリズムが目指されている。

2.6 本章のまとめ

本章では、本論文で扱う最適化問題、および多点型確率的最適化アルゴリズムの構造を整理したうえで、既存の有力なメタヒューリスティクスである PSO や DE を多点型確率的最適化アルゴリズムと捉え、これらのアルゴリズムについて概説した。2.5.5 項で述べたように、メタヒューリスティクスの多くは、きわめて高い設計の自由度を有し、問題にあわせてアルゴリズムを自由に構築できる柔軟さを持ち合わせているが、数学的な裏付けが必ずしもないため、たとえばパラメータ設定などにおいて常に試行錯誤を求められているという実運用上の課題がある。この課題に対して、3 章および 4 章では PSO と DE のそれぞれについて、理論的なアプローチによりアルゴリズムの動特性を調べ、パラメータ設定の明確な指針を与えていく。

第3章 Particle Swarm Optimizationの収束性解析

3.1 本章について

本章では、Particle Swarm Optimization (PSO) の動特性解析として、探索点の収束性とパラメータの関係を考察する。PSO の最適化性能は探索点の収束性と強い関係があり、また探索点の収束性はパラメータ設定に依存するため、PSO を最適化問題へ適用する際には、大域的探索と局所的探索の所望のバランスや、許容される計算時間などを考慮して、パラメータを適切に調整することが求められる。このことから、PSO の動特性、とくに探索点の収束性とパラメータとの関係には関心が集められており、多くの研究や報告がある。

PSO の収束性解析上、数学的な取り扱いが困難とされるのはつぎの2点である。

- (1) p -best や g -best といったエリート探索点座標が、目的関数の形状や探索状況によって定まる点
- (2) 探索点座標更新式に乱数が含まれている点

これらのうち、(1) に関しては、 p -best や g -best を定数として近似するのが通例である^[110, 111]。PSO を実際の最適化問題に適用すると、 p -best や g -best は探索初期では頻繁に更新されるものの、探索が進むと既得の解より優れた解が見つかりづらくなるため、 p -best や g -best の更新は停滞する。PSO に求められる理想的な性質は、 p -best や g -best の更新が停滞した状況から抜け出して、より優れた解を見つけ出すことであり、この状況での動特性がもっとも重要となる。したがって、 p -best や g -best を定数とおくことは合理的な近似であるといえる。(2) に関しては、乱数の影響を直接考慮するのではなく、乱数を確率変数とみなし、たとえば期待値に代表させて定数とおく手法が代表的である^[73, 74]。乱数を定数とおき、さらに p -best と g -best を定数と仮定すれば、PSO の探索点位置更新式は、線形時不変システムとして記述できるため、固有値解析に基づく安定性解析手法を用いて、パラメータと安定性の関係を導くことが可能である。しかしながら、近似モデルと実システムの乖離は小さくなく、実用上影響のある誤差があることが指摘されている^[112]。近年では、さらに乱数がしたがう確率分布の高次のモーメントまで考慮して解く^[111, 113]、あるいは確率システム安定論の考察に基づき、PSO と等価な離散時間確率システムの自乗平均安定について論じた研究があり^[114]、これらの解析から導かれる安定／不安定の境界は、数値実験結果^[112]と相当に近いものとなっている。しかしながら、PSO のパラメータと収束性に関する厳密な関係は未だに得られていない。

そこで、本論文では、PSO のアルゴリズムを確率システムとしてみなす点については従来の研究と同様のアプローチに基づきつつも、確率変数の有限次までのモーメントを用いた議論はせず、確率分布を精緻に考慮した解析をおこなう。乱数の影響を受ける PSO のア

ルゴリズムにおいて、微視的な時間時刻スケールでは探索点の挙動を解析することは原理的に不可能であるが、PSO のアルゴリズムそのものは時間に対して不変であることから、巨視的な時間間隔で探索点の挙動を観察すると、毎回の計算の反復における個々の乱数の影響が相対的に小さくなるため、探索点の動特性を統計的に捉えることが可能となる。この統計的な動特性を数学的に記述することにより、PSO の安定条件のみならず、「どの程度の速さで探索点が収束（あるいは発散）するか」を精緻に予測する指標もあわせて導くことが可能となる。

本章ではまず、3.2 節で、PSO の探索点位置更新式を確率システムとして表す。3.3 節では、3.2 節で導いた確率システムに対する安定性を定義したうえで、具体的な PSO の動特性として、探索点の収束性の解析をおこなう。3.4 節では、提案解析手法が有効であることを確認するとともに、提案解析手法の応用として、局所的探索と大域的探索を持続的におこなうことを可能とする「持続的探索パラメータ」の例を具体的に与える。3.5 節では、数値実験により、持続的探索パラメータの有効性を既知の推奨パラメータ設定との比較を通じて確認する。最後に 3.6 節では、本章の内容をまとめるとともに、提案解析手法の有する課題について述べる。

3.2 PSO の探索点位置更新式の確率システム表現

PSO の探索点位置の更新式は、第 2 章で示したとおり、

$$\begin{aligned}
 \mathbf{v}^{(p)}(k+1) = & \lambda \mathbf{v}^{(p)}(k) + c_1 \mathbf{R}_1^{(p)}(k)(\mathbf{x}^{(p\text{-best})}(k) - \mathbf{x}^{(p)}(k)) \\
 & + c_2 \mathbf{R}_2^{(p)}(k)(\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k))
 \end{aligned} \quad (3.1a)$$

$$\mathbf{x}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k+1) \quad (3.1b)$$

で与えられる。(3.1a) 式において、探索が停滞した状況を想定して $p\text{-best}$ および $g\text{-best}$ を時間変化しない定数とみなし、

$$\mathbf{x}^{(p\text{-best})}(k) = \mathbf{x}_C^{(p\text{-best})} \quad \forall k = 0, 1, 2, \dots \quad (3.2a)$$

$$\mathbf{x}^{(g\text{-best})}(k) = \mathbf{x}_C^{(g\text{-best})} \quad \forall k = 0, 1, 2, \dots \quad (3.2b)$$

とおくと、PSO の探索点座標更新式 (3.1) は、探索空間の次元 ($n = 1, \dots, N$) ごとに、

$$\begin{aligned}
 \begin{pmatrix} y_n^{(p)}(k+1) \\ y_n^{(p)}(k) \end{pmatrix} = & M(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k); \lambda, c_1, c_2) \begin{pmatrix} y_n^{(p)}(k) \\ y_n^{(p)}(k-1) \end{pmatrix} \\
 & + u_n^{(p)}(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k)) \begin{pmatrix} 1 \\ 0 \end{pmatrix}
 \end{aligned} \quad (3.3a)$$

$$M(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k); \lambda, c_1, c_2) = \begin{pmatrix} 1 + \lambda - c_1 R_{1n}^{(p)}(k) - c_2 R_{2n}^{(p)}(k) & -\lambda \\ 1 & 0 \end{pmatrix} \quad (3.3b)$$

$$y_n^{(p)}(k) = x_n^{(p)}(k) - z_n^{(p)} \quad (3.3c)$$

$$z_n^{(p)} = \frac{c_1 x_{Cn}^{(p\text{-best})} + c_2 x_{Cn}^{(g\text{-best})}}{c_1 + c_2} \quad (3.3d)$$

$$u_n^{(p)}(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k)) = \frac{c_1 c_2 (x_{Cn}^{(p\text{-best})} - x_{Cn}^{(g\text{-best})})(R_{1n}^{(p)}(k) - R_{2n}^{(p)}(k))}{c_1 + c_2} \quad (3.3e)$$

と変形でき、係数行列 $M(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k); \lambda, c_1, c_2)$ が乱数 $R_{1n}^{(p)}(k), R_{2n}^{(p)}(k)$ により探索時刻 k ごとに変化する確率システムが得られる¹。

3.3 PSO システムの安定性解析

3.3.1 PSO システムの安定性の定義

PSO のパラメータと収束性の関係は、システム (3.3) 式の内部安定性解析から導くことができる。本論文では、(3.3a) 式の右辺第 2 項の入力をゼロとしたシステム

$$\begin{pmatrix} y_n^{(p)}(k+1) \\ y_n^{(p)}(k) \end{pmatrix} = M(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k); \lambda, c_1, c_2) \begin{pmatrix} y_n^{(p)}(k) \\ y_n^{(p)}(k-1) \end{pmatrix} \quad (3.4)$$

に注目する。なお、(3.4) 式で記述されるシステムは、探索点および探索空間の次元ごとに独立かつ共通したものであることから、特定の探索点の特定の探索空間の次元に着目した安定性解析であっても一般性は失われない。そこで、以下の議論では、探索点番号を示す添字 p および探索空間の次元を示す添字 n の表記を省略し、さらに $\mathbf{p}(k) = (p_1(k), p_2(k))^T = (y_n^{(p)}(k), y_n^{(p)}(k-1))^T$ とおき、システム (3.4) を、

$$\mathbf{p}(k+1) = M(R_1(k), R_2(k); \lambda, c_1, c_2) \mathbf{p}(k) \quad (3.5)$$

と書くこととする。

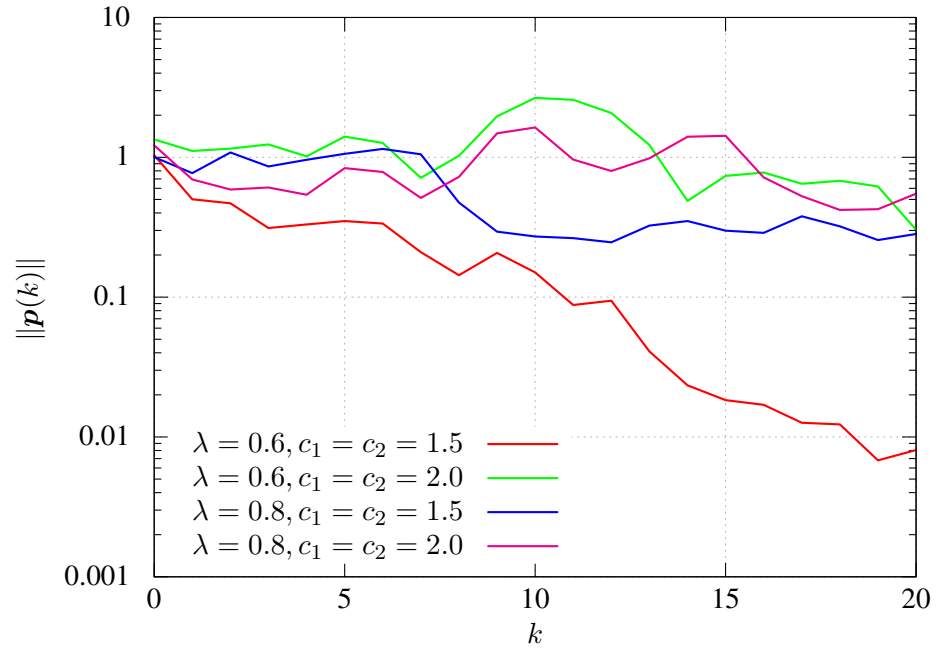
システム (3.5) が確率漸近安定条件、すなわち任意の正数 $\varepsilon > 0$ に対して、

$$\lim_{K \rightarrow \infty} \Pr \left\{ \sup_{k > K} \|\mathbf{p}(k)\| > \varepsilon \right\} = 0 \quad (3.6)$$

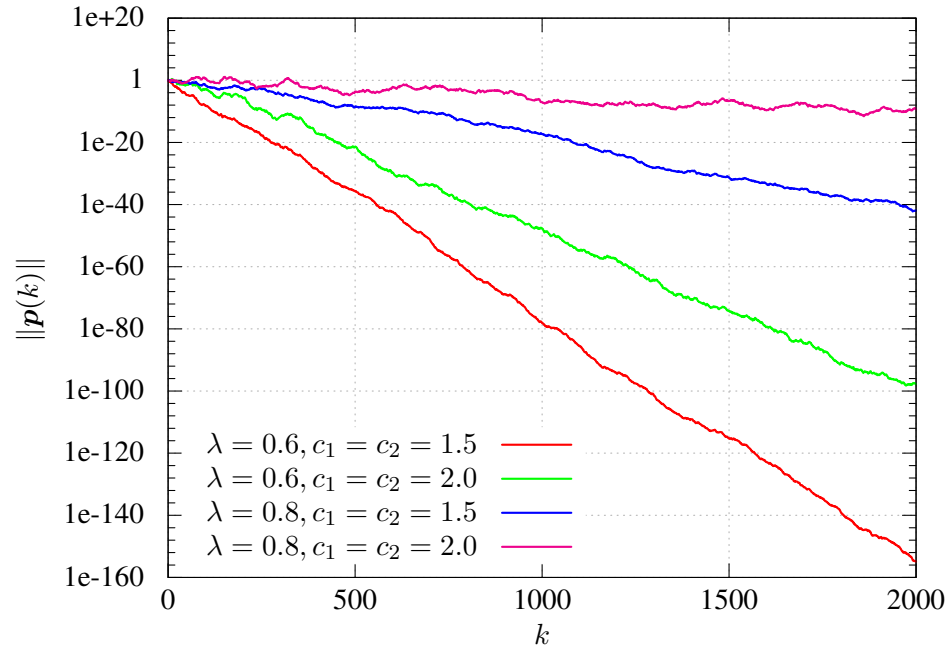
が成立する条件を考える。パラメータ λ, c_1, c_2 に対して、システム (3.4) 式が確率漸近安定条件 (3.6) を満たすとき、PSO は、システム (3.4) 式と等価な内部安定性を有することから、その挙動は安定的であり、とくに $\mathbf{x}^{p\text{-best}}(k) = \mathbf{x}^{g\text{-best}}(k)$ であるとき、探索点は特定の座標に確率的に収束する。ただし、収束する座標の局所的最適性は保証されないことに注意が必要である。なお、目的関数の違いによる PSO の動特性の違いは、 $p\text{-best}$ と $g\text{-best}$ の座標とその時間変化のみに現れるが、システム (3.5) は、 $p\text{-best}$ と $g\text{-best}$ を含まない形式で記述されており、システム (3.5) の安定性解析から得られる PSO のパラメータと収束性の関係は、あらゆる目的関数に対して適用することができる。

Fig. 3.1 に、いくつかのパラメータ設定に対するシステム (3.5) の挙動を示す。本章冒頭で述べたとおり、微視的な探索時刻スケールでみた場合は個々の探索時刻における乱数の影響が強いものの、巨視的な探索時刻スケールでみた場合は個々の乱数の影響が相対的に薄まり、パラメータ設定によって、システムは一定の傾向を示すことがわかる。

¹係数行列 $M(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k); \lambda, c_1, c_2)$ の引数に含まれる λ, c_1, c_2 は定数であるが、システムの挙動がこれらの値に依存するという点を強調するため、変数である $R_{1n}^{(p)}(k), R_{2n}^{(p)}(k)$ とセミコロン (;) で区切って記載した。



(a) Micro-time behaviour



(b) Macro-time behaviour

Fig. 3.1 System behaviours of system (3.5) for several parameter settings.

3.3.2 慣性項がある場合の安定性解析

システム (3.5) の状態量 $\mathbf{p}(k+1)$ は、初期値を $\mathbf{p}(0)$ とすると、

$$\mathbf{p}(k+1) = \left(\prod_{\kappa=0}^k M(R_1(k-\kappa), R_2(k-\kappa); \lambda, c_1, c_2) \right) \mathbf{p}(0) \quad (3.7)$$

と書ける. (3.7) 式において、係数行列 $M(R_1(k-\kappa), R_2(k-\kappa); \lambda, c_1, c_2)$ は、異なる κ に対して互いに非可換であるため、探索時刻 k における状態量 $\mathbf{p}(k)$ は、乱数 R_1, R_2 が探索時刻 k までにとった値と、さらにその順序に依存する. そこで本論文では、システム (3.5) の安定性を考察するため、状態量 $\mathbf{p}$ の 2 乗ノルム $\|\mathbf{p}\|$ の確率漸近安定条件を考える. そのために、2 次元の状態量 $\mathbf{p}$ を極座標上で考え、これと同じ向きの単位ベクトルとその偏角

$$\mathbf{q}(\theta) = (\cos(\theta), \sin(\theta))^T = \frac{\mathbf{p}}{\|\mathbf{p}\|} \quad (3.8a)$$

$$\theta = \arg(\mathbf{p}) = \begin{cases} \arccos\left(\frac{p_1}{\|\mathbf{p}\|}\right), & p_2 \geq 0 \\ 2\pi - \arccos\left(\frac{p_1}{\|\mathbf{p}\|}\right), & p_2 < 0 \end{cases} \quad (3.8b)$$

を導入すると、 $\|\mathbf{p}(k+1)\|$ は、

$$\begin{aligned} \|\mathbf{p}(k+1)\| &= \|M(R_1(k), R_2(k); \lambda, c_1, c_2)\mathbf{p}(k)\| \\ &= \|M(R_1(k), R_2(k); \lambda, c_1, c_2)\mathbf{q}(\theta(k))\| \times \|\mathbf{p}(k)\| \\ &= \left(\prod_{\kappa=0}^k \|(M(R_1(\kappa), R_2(\kappa); \lambda, c_1, c_2)\mathbf{q}(\theta(\kappa)))\| \right) \|\mathbf{p}(0)\| \end{aligned} \quad (3.9)$$

と書くことができる. (3.9) 式において、

$$D(k) = \prod_{\kappa=0}^k \|(M(R_1(\kappa), R_2(\kappa); \lambda, c_1, c_2)\mathbf{q}(\theta(\kappa)))\| > 0 \quad (3.10)$$

とおくと、システム (3.5) の確率漸近安定条件は、任意の正数 $\varepsilon > 0$ に対して、

$$\lim_{K \rightarrow \infty} \Pr \left\{ \sup_{k > K} \|\mathbf{p}(k+1)\| > \varepsilon \right\} = \lim_{K \rightarrow \infty} \Pr \left\{ \sup_{k > K} (D(k)\|\mathbf{p}(0)\|) > \varepsilon \right\} = 0 \quad (3.11)$$

であり、 $\|\mathbf{p}(0)\|$ は定数であることから、 $D(k)$ を求めることができれば、システム (3.5)、すなわち PSO の安定性の判別が可能となる. $D(k)$ の総積記号内部はスカラーの積であり、その値は積の順序に依存しない. よって、各探索時刻における乱数 R_1, R_2 や、単位ベクトル $\mathbf{q}$ の偏角 θ の値が具体的に与えられなくても、その分布がわかれば $D(k)$ を計算することができる. 偏角 θ の分布密度関数 $\rho(\theta)$ は、探索時刻 $k=0$ から探索時刻 $k \rightarrow \infty$ の間に $\theta(k)$ がとる値の分布と、ある探索時刻 k ($k \gg 1$) において偏角 $\theta(k)$ がとる値の確率分布が一致する、すなわちエルゴード性の仮定のもとで、Dirac のデルタ関数を用いて、

$$\rho(\theta) = \int_0^1 dR_1 \int_0^1 dR_2 \int_0^{2\pi} \delta(\theta - \arg(M(R_1, R_2; \lambda, c_1, c_2)\mathbf{q}(\theta'))) \rho(\theta') d\theta' \quad (3.12)$$

で与えられる。以降では、議論簡単化のため、探索が停滞し、 p -best と g -best の更新が止まり、かつ $\rho(\theta)$ が十分に定常に達した探索時刻を $k = 0$ とする。 $D(k)$ は、

$$\begin{aligned} D(k) &= \prod_{\kappa=0}^k \|(M(R_1(\kappa), R_2(\kappa); \lambda, c_1, c_2) \mathbf{q}(\theta(\kappa)))\| \\ &= \exp \left(k \frac{1}{k} \sum_{\kappa=0}^k \log \|(M(R_1(\kappa), R_2(\kappa); \lambda, c_1, c_2) \mathbf{q}(\theta(\kappa)))\| \right) \end{aligned} \quad (3.13)$$

と変形でき、ここで $R_1(k), R_2(k)$ が $[0,1]$ 間の一様分布で定まり、またこれらは $\theta(k)$ とは独立に定まることに注意すると、大数の弱法則より、任意の正数 $\delta > 0$ に対して、

$$\lim_{k \rightarrow \infty} \Pr \left\{ \left| \left(\frac{1}{k} \sum_{\kappa=0}^k \log \|(M(R_1(\kappa), R_2(\kappa); \lambda, c_1, c_2) \mathbf{q}(\theta(\kappa)))\| \right) - \mu \right| > \delta \right\} = 0 \quad (3.14)$$

が成り立つ。ただし、(3.14) 式において、 μ は $\log \|(M(R_1(\kappa), R_2(\kappa); \lambda, c_1, c_2) \mathbf{q}(\theta(\kappa)))\|$ の期待値であり、

$$\mu = \int_0^1 dR_1 \int_0^1 dR_2 \int_0^{2\pi} \log \|M(R_1, R_2; \lambda, c_1, c_2) \mathbf{q}(\theta)\| \rho(\theta) d\theta \quad (3.15)$$

で与えられる²。(3.13) 式、(3.14) 式および (3.15) 式より、

$$\lim_{k \rightarrow \infty} \Pr \{ \exp(k(\mu - \delta)) < D(k) < \exp(k(\mu + \delta)) \} = 1 \quad (3.16)$$

が得られる。ここで、 $\delta > 0$ は任意に小さくすることが可能であり、 $D(k)$ は、 $k \gg 1$ のとき、

$$D(k) \approx \exp(k\mu) \quad (3.17)$$

と見積ることが可能である。ここで、

$$\mu = \int_0^1 dR_1 \int_0^1 dR_2 \int_0^{2\pi} \log \|M(R_1, R_2; \lambda, c_1, c_2) \mathbf{q}(\theta)\| \rho(\theta) d\theta < 0 \quad (3.18)$$

を満たすとき、 $D(k)$ の見積りは k に対して単調減少となる。したがって、任意の $\varepsilon > 0$ に対して K を十分大きくとれば任意の $k > K$ に対して $D(k) \|\mathbf{p}(0)\| > \varepsilon$ となる確率を十分に小さくでき、システム (3.5) は確率漸近安定条件 (3.11) を満足する。

ところで、(3.15) 式の値は、システムの安定判別のみならず、収束あるいは発散の速さの指標としても利用することができる。いま、

$$\zeta(\lambda, c_1, c_2) = \exp \left(\int_0^1 dR_1 \int_0^1 dR_2 \int_0^{2\pi} \log \|M(R_1, R_2; \lambda, c_1, c_2) \mathbf{q}(\theta)\| \rho(\theta) d\theta \right) \quad (3.19)$$

とおけば、(3.9) 式は、十分大きい $k \gg 1$ に対して、

$$\|\mathbf{p}(k)\| \approx (\zeta(\lambda, c_1, c_2))^k \|\mathbf{p}(0)\| \quad (3.20)$$

² $\log \|(M(R_1(\kappa), R_2(\kappa); \lambda, c_1, c_2) \mathbf{q}(\theta(\kappa)))\|$ は負の無限大の値をとりうる（特異点をもつ）ため、本来 (3.15) 式は広義積分として定義し、有限な期待値が存在することを証明する必要がある。本論文では、期待値の存在性は証明できていないが、後の数値計算結果により、(3.15) 式が有限の値として求まることを確認している。

と近似することができ、 $\zeta(\lambda, c_1, c_2)$ の値と探索時刻 k からシステムの収束挙動（発散挙動）を予測することができる。

なお、本節で導入した偏角の密度分布 $\rho(\theta)$ や安定性指標 $\zeta(\lambda, c_1, c_2)$ は解析的に求めることは困難であり、数値的に求める必要がある。具体的な計算方法については Appendix C を参照されたい。

3.3.3 慣性項がない場合の安定性解析

PSO の探索点座標更新式 (3.1) は、一般的に慣性項をもった構造をしており、そのシステム表現は2階の差分方程式となり、係数は行列で与えられる。ここでは、PSO の特殊なケースとして慣性項の係数がゼロの場合、すなわち (3.1a) 式において $\lambda = 0$ の場合を考える。慣性項の係数がゼロのときは、システムの係数をスカラーで表現することができ、確率漸近安定条件を解析的に求めることが可能となる。

(3.1a) 式において $\lambda = 0$ としたとき、(3.3a) 式～(3.3e) 式にならって (3.1a) 式を変形すると、

$$y_n^{(p)}(k+1) = (1 - c_1 R_{1n}^{(p)}(k) - c_2 R_{2n}^{(p)}(k)) y_n^{(p)}(k) + u_n^{(p)}(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k)) \quad (3.21a)$$

$$y_n^{(p)}(k) = x_n^{(p)}(k) - z_n^{(p)} \quad (3.21b)$$

$$z_n^{(p)} = \frac{c_1 x_{Cn}^{(p\text{-best})} + c_2 x_{Cn}^{(g\text{-best})}}{c_1 + c_2} \quad (3.21c)$$

$$u_n^{(p)}(R_{1n}^{(p)}(k), R_{2n}^{(p)}(k)) = \frac{c_1 c_2 (x_{Cn}^{(p\text{-best})} - x_{Cn}^{(g\text{-best})}) (R_{1n}^{(p)}(k) - R_{2n}^{(p)}(k))}{c_1 + c_2} \quad (3.21d)$$

となり、 $\lambda \neq 0$ の場合とは異なり、係数がスカラーで定まる確率システムが得られる。(3.21a) 式において入力項を除き、また状態変数を 3.3.2 項の議論にならって $p(k) = y(k)$ とおいたシステム表現は、

$$p(k+1) = (1 - c_1 R_1(k) - c_2 R_2(k)) p(k) \quad (3.22)$$

となる。システム (3.22) の確率漸近安定条件は、任意の正数 $\varepsilon > 0$ に対して、

$$\lim_{K \rightarrow \infty} \Pr \left\{ \sup_{k > K} |p(k)| > \varepsilon \right\} = 0 \quad (3.23)$$

が成り立つことである。

システム (3.22) を初期値 $p(0)$ と時変のスカラー係数を用いて書き下すと、

$$p(k+1) = \left(\prod_{\kappa=0}^k (1 - c_1 R_1(\kappa) - c_2 R_2(\kappa)) \right) p(0) \quad (3.24)$$

となる。ここで、

$$D(k) = \prod_{\kappa=0}^k (1 - c_1 R_1(\kappa) - c_2 R_2(\kappa)) \quad (3.25)$$

とおけば、確率漸近安定条件 (3.23) は具体的に、

$$\lim_{K \rightarrow \infty} \Pr \left\{ \sup_{k > K} |p(k)| > \varepsilon \right\} = \lim_{K \rightarrow \infty} \Pr \left\{ \sup_{k > K} (|D(k)| |p(0)|) > \varepsilon \right\} = 0 \quad (3.26)$$

で表され、 $p(0)$ は定数であることから、 $|D(k)|$ を求めることができれば慣性項のない PSO と等価なシステム (3.22) の安定性の判別が可能となる。 $|D(k)|$ は、

$$\begin{aligned} |D(k)| &= \prod_{\kappa=0}^k |1 - c_1 R_1(\kappa) - c_2 R_2(\kappa)| \\ &= \exp \left(k \frac{1}{k} \sum_{\kappa=0}^k \log |1 - c_1 R_1(\kappa) - c_2 R_2(\kappa)| \right) \end{aligned} \quad (3.27)$$

と変形でき、ここで、 $R_1(k), R_2(k)$ が $[0,1]$ 間の一様分布で定まることに注意すると、大数の弱法則より、任意の正数 $\delta > 0$ に対して、

$$\lim_{k \rightarrow \infty} \Pr \left\{ \left| \left(\frac{1}{k} \sum_{\kappa=0}^k \log |1 - c_1 R_1(\kappa) - c_2 R_2(\kappa)| \right) - \mu \right| > \delta \right\} = 0 \quad (3.28)$$

が成り立つ。ただし、(3.28) 式において、 μ は $\log |1 - c_1 R_1(\kappa) - c_2 R_2(\kappa)|$ の期待値であり、

$$\mu = \int_0^1 dR_1 \int_0^1 \log |1 - c_1 R_1(\kappa) - c_2 R_2(\kappa)| dR_2 \quad (3.29)$$

で与えられる。

以降、3.3.2 項と同様の議論により、システム (3.22) の確率漸近安定条件は、

$$\mu = \int_0^1 dR_1 \int_0^1 \log |1 - c_1 R_1 - c_2 R_2| dR_2 < 0 \quad (3.30)$$

となる。また、安定性指標 $\zeta(c_1, c_2)$ は、

$$\zeta(c_1, c_2) = \exp \left(\int_0^1 dR_1 \int_0^1 \log |1 - (c_1 R_1 + c_2 R_2)| dR_2 \right) \quad (3.31)$$

で与えられる。

(3.31) 式の積分は解析的に計算することができる。たとえば、 $c_1 = 0, c_2 \neq 0$ のとき、システム (3.22) の確率漸近安定条件は、

$$(c_2 - 1) \log |c_2 - 1| - c_2 < 0 \quad (3.32)$$

であり、(3.32) 式を c_2 について解くと $c_2 < 4.5911$ が得られる。また、 $c_1 = c_2 = c \neq 0$ のとき、システム (3.22) の確率漸近安定条件は、

$$(2c - 1)^2 \log |2c - 1| - (3c^2 + 2(c - 1)^2 \log |c - 1|) < 0 \quad (3.33)$$

であり、(3.33) 式を c について解くと $c < 2.3196$ が得られる。ここで求めた値はすなわち、PSO のシステムが安定となるパラメータ設定の上限値である。

Table 3.1 Examples of index of stability.

No.	Parameter Setting		$\zeta(\lambda, c_1, c_2)$
1 ^[74]	$\lambda = 0.60000$	$c_1 = c_2 = 1.70000$	0.84876
2 ^[73]	$\lambda = 0.72900$	$c_1 = c_2 = 1.49445$	0.90713
3 ^[52]	$\lambda = 0.83210$	$c_1 = c_2 = 2.00000$	0.99991

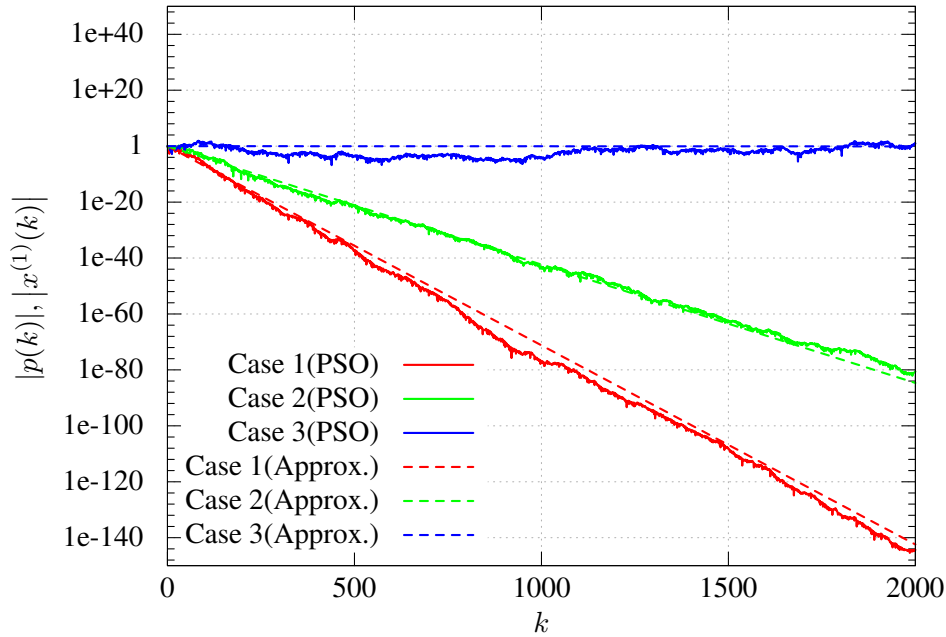


Fig. 3.2 Comparison between actual PSO behaviours and predictions on Eq. (3.20).

3.4 安定性指標に基づく探索点の収束挙動の解析と持続的探索パラメータ

Table 3.1 に、複数のパラメータ設定に対するシステム (3.5) の安定性指標 ζ の値を示す。Table 3.1 の No. 1 および No. 2 は、既存研究により提案された推奨パラメータ設定であり、No. 3 は、後述する「持続的探索パラメータ設定」である。また Fig. 3.2 に、実際の PSO の探索点の収束挙動 (図中、PSO と記された系列) と、Table 3.1 の安定性指標 ζ の値をもとに (3.20) 式で見積った PSO の収束挙動 (図中、Approx. と記された系列) を比較した結果を示す。ここで PSO で最適化する目的関数 $f(x)$ は、理論的予測と数値実験結果の比較を容易にするために、単純な形状の 1 変数 2 次関数 $f(x) = x^2$ を採用している。Fig. 3.2 より、(3.20) 式を用いて PSO の収束挙動を良い精度で解析できていることがわかる。

2.4.1 項で述べたように、複数の局所的最適解を有する多峰性目的関数の大域的最適解を探索するためには、「集中化と多様化」とよばれる 2 つの相反する方策をアルゴリズムに同時に具備させることが必要となる。「集中化」とは、既得の最適解候補の近傍を集中的に探索することで、より優れた解を得る働きを指し、「多様化」とは、探索領域全体を広く探索することで「集中化」により得た解以上に優れた解を発見する働きを指す。PSO は、

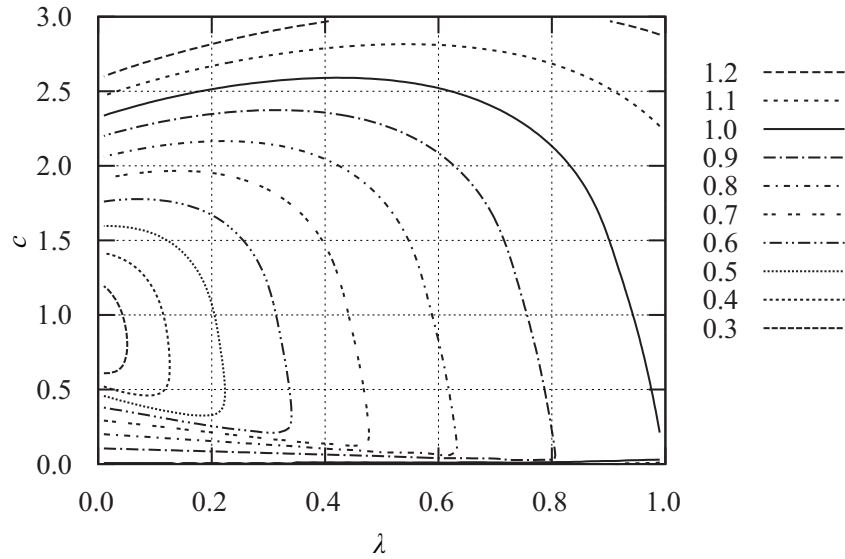


Fig. 3.3 Contour of index of stability for several parameter settings.

目的関数の勾配や Hesse 行列を利用しないため、集中化方策を実現するためには、システム (3.5) が安定的となるパラメータを設定し、 p -best や g -best の近傍を集中的に探索させる必要がある。一方で、安定的なパラメータを設定すると、探索領域の一部である p -best や g -best 近傍に探索点が集中し、探索領域全体に探索点が行きわたらなくなるため、大域的探索能力は弱まる。そこで、安定性指標 ζ の値が 1.0 に近くなるようにパラメータを設定し、システムの安定性を安定と不安定の境界上におくことで、収束（集中化）と発散（多様化）の繰り返しを促し、持続的な探索を図る方法が考えられる。

Fig. 3.3 に、パラメータ $\lambda, c = c_1 = c_2$ に対する安定性指標 $\zeta(\lambda, c = c_1 = c_2)$ の値を等高線で示す。また Table 3.2 に、 $\zeta = 1.0$ となり持続的探索を実現する具体的なパラメータ（以下、持続的探索パラメータ） λ, c_1, c_2 の設定例を示す。

ところで、PSO のシステムの挙動はもともと探索空間の次元ごとに独立であるため、持続的探索パラメータを設定した場合、問題の次元が大きくなるにしたがい、「すべての次元に関して収束的挙動である」という状態が実現されにくくなるため、集中的な探索が実現されなくなると考えられる。そこで、持続的探索パラメータを利用する場合は、PSO の探索点位置更新式を、

$$\begin{aligned} \mathbf{v}^{(p)}(k+1) = & \lambda \mathbf{v}^{(p)}(k) + c_1 R_1^{(p)}(k) (\mathbf{x}^{(p\text{-best})}(k) - \mathbf{x}^{(p)}(k)) \\ & + c_2 R_2^{(p)}(k) (\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k)) \end{aligned} \quad (3.34a)$$

$$\mathbf{x}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k+1) \quad (3.34b)$$

とし、(3.34a) 式の右辺第 2 項および第 3 項の乱数係数 $R_1^{(p)}(k), R_2^{(p)}(k)$ を、スカラーとして実数値一様分布 $\mathcal{U}_{\mathbf{R}}(0, 1)$ にしたがう乱数で決定し、全次元の乱数を同期することが望ましいと考えられる。本方策の有効性については、3.5 節の数値実験で検証する。

Table 3.2 Examples of parameter settings for sustainable search.

λ	$c_1 = c_2$	λ	$c_1 = c_2$	λ	$c_1 = c_2$	λ	$c_1 = c_2$	λ	$c_1 = c_2$
0.02	2.3501	0.22	2.5268	0.42	2.5913	0.62	2.5026	0.82	2.0547
0.04	2.3732	0.24	2.5384	0.44	2.5908	0.64	2.4802	0.84	1.9623
0.06	2.3948	0.26	2.5489	0.46	2.5888	0.66	2.4544	0.86	1.8490
0.08	2.4154	0.28	2.5583	0.48	2.5851	0.68	2.4247	0.88	1.7040
0.10	2.4346	0.30	2.5666	0.50	2.5797	0.70	2.3907	0.90	1.5128
0.12	2.4527	0.32	2.5738	0.52	2.5725	0.72	2.3518	0.92	1.2822
0.14	2.4697	0.34	2.5798	0.54	2.5632	0.74	2.3074	0.94	1.0375
0.16	2.4856	0.36	2.5846	0.56	2.5518	0.76	2.2568	0.96	0.7574
0.18	2.5004	0.38	2.5882	0.58	2.5381	0.78	2.1990	0.98	0.4239
0.20	2.5141	0.40	2.5904	0.60	2.5217	0.80	2.1323	1.00	0.0000

Table 3.3 Common numerical simulation conditions.

Item	Value
Max Iteration (K)	5000
Trial	200
Number of Search Points (P)	20
Parameter (c_1, c_2, λ)	Shown in Each Numerical Simulation
Link of Random Coefficients ($R_1^{(p)}, R_2^{(p)}$)	Shown in Each Numerical Simulation
Initial Condition	$x_n^{(p)}(0) \sim \mathcal{U}_{\mathbb{R}}(-5.0, 5.0)$
Bounding Method	Torus

3.5 数値実験による持続的探索パラメータに基づく PSO の性能評価

3.5.1 数値実験条件

本節では、最適化ベンチマーク問題への PSO の適用を通して、持続的探索パラメータに基づく PSO の最適化性能を評価する。Table 3.3 に、本節における数値実験の共通条件を示す。Table 3.3 で“Shown in Each Numerical Simulation”となっている条件は、以降、各数値実験の項において個別に示す。また、Table 3.4 に、本節において用いるベンチマーク問題の一覧を示す。

3.5.2 推奨値パラメータと持続的探索パラメータの比較

Tables 3.5–3.10 に、Table 3.1 に示した 3 つのパラメータ設定 (Fig. 3.7) の PSO を用いて、Table 3.4 に示す各ベンチマーク問題を解いた結果を示す。各表では最適化性能として、得られた $f(\mathbf{x}^{(\text{g-best})})$ の平均、標準偏差および最大、最小を示している。また、Figs. 3.7–3.12 に、200 回の試行のうち、ある 1 回の試行における探索時刻に対する $f(\mathbf{x}^{(\text{g-best})})$ の推移を

Table 3.4 Benchmark problems.

No.	Problem	Dimension (N)	Rotation Angle (α)
1	Prob. 1 (Sphere Function)	20	0.0
2	Prob. 2 (Rosenbrock's Function)	20	0.0
3	Prob. 3 (2^N minima Function)	20	0.0
4	Prob. 3 (2^N minima Function)	20	0.3
5	Prob. 4 (Rastrigin's Function)	20	0.0
6	Prob. 4 (Rastrigin's Function)	20	0.3

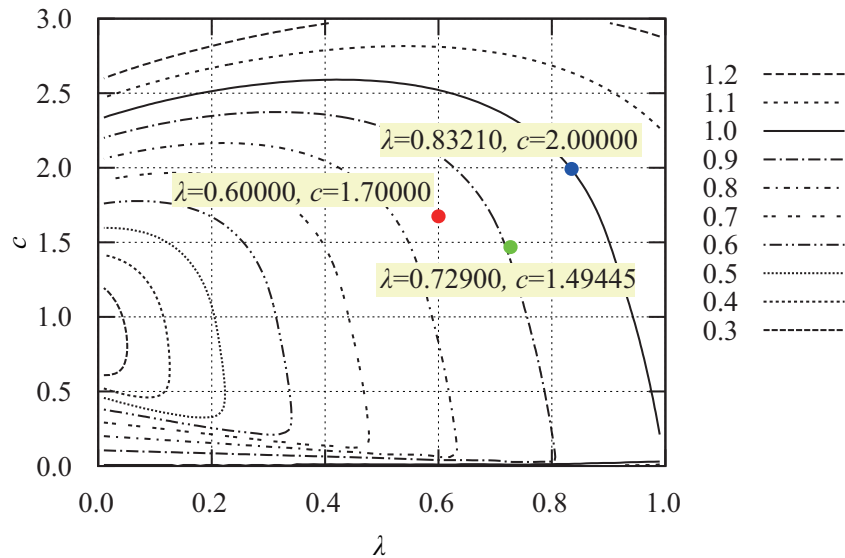


Fig. 3.4 Parameter settings map.

(1) Comparison between conventional and sustainable search parameters

示す。なお、本項の数値実験では、いずれのパラメータ設定に対しても乱数を同期させた場合、同期させない場合の両方について結果を示している。

Tables 3.5–3.10 より、持続的探索パラメータを設定した PSO は、乱数を各次元で同期させたほうが、同期させない場合と比較して性能が顕著に優れている。乱数を同期させない場合は、Table 3.5(a) に示すとおり、単峰性凸関数のきわめて最適化が容易な Prob. 1 (Sphere Function) においても最適解を得ることができていない。これは、3.5 節で考察したとおり、持続的探索パラメータを設定して、かつ乱数を同期させない場合、「探索空間の全次元に関してシステムが安定的である」という状態が実現されにくくなり、局所的探索が十分になされていないためであると考えられる。

一方で、安定傾向を示すパラメータの場合では、乱数は各次元で同期させない方が良好な結果が得られている。これは、システムの安定性が保証されたうえで、かつ乱数を探索空間の次元ごとに異なるものを用いることで、探索の多様性を獲得しているためと考えられる。

問題ごとに各パラメータ設定の最適化性能をみると、持続的探索パラメータ ($c_1 = c_2 =$

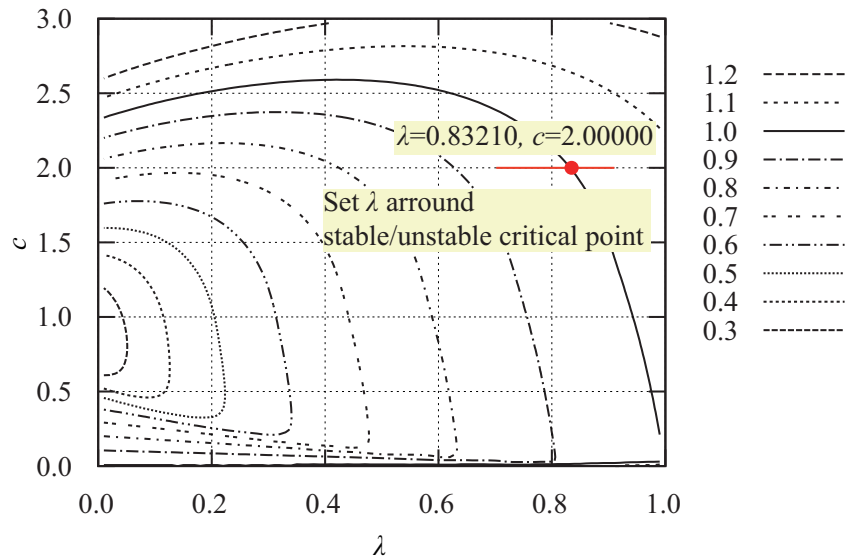


Fig. 3.5 Parameter settings map.

(2) Sensitivity analysis of sustainable search parameter

2.00000, $\lambda = 0.83210$) のうち、乱数を同期させた場合の最適化性能が平均的に優れており、とくに多数の局所的最適解を有する Prob. 3 において好成績を残している。Figs. 3.7–3.12 より、従来推奨値パラメータ ($c_1 = c_2 = 1.49445$, $\lambda = 0.72900$ および $c_1 = c_2 = 1.70000$, $\lambda = 0.60000$) を用いた場合では、いずれの問題においても探索点（群）は探索の早い段階で特定の座標に収束し、以降 $\mathbf{x}^{(g\text{-best})}$ の更新がないのに対して、持続的探索パラメータを用いた場合では、収束の速度は従来推奨値パラメータより劣るものの、継続的な $\mathbf{x}^{(g\text{-best})}$ の更新がなされ、とくに Fig. 3.11(b) などでは、最終的には従来推奨値パラメータより良い解が得られている。これは、持続的探索パラメータの有する特徴である「集中化と多様化」が有効に作用した結果と考えられる。

ただし、Prob. 1 や Prob. 2 などのように、局所的最適解の数が唯一の問題では局所的探索を集中的におこなうべきであるのにも関わらず「集中化と多様化」を繰り返すパラメータ設定となっているため、安定傾向、すなわち収束傾向の強いパラメータと比較してやや局所的探索が不十分な結果となっている。

3.5.3 持続的探索パラメータの最適性

Tables 3.11–3.16 および Figs. 3.13–3.18 に、 $c_1 = c_2 = 2.0$ を固定として、 λ を持続的探索パラメータ ($\lambda = 0.8321$) 付近で増減させた場合 (Fig. 3.5) における、各ベンチマーク問題に対する最適化性能を示す。Tables 3.11–3.16 および Figs. 3.13–3.18 の各図表より、問題に応じて最適化性能の全体的な傾向は異なるものの、どの問題においても持続的探索パラメータ ($\lambda = 0.8321$) 付近においてよい最適化性能を示していることがわかる。Prob. 1 (Sphere Function) に対する結果をみると、同問題は単峰性凸関数のきわめて簡単な最適化問題であるにもかかわらず、 λ の値が小さいと最適解を得ることができていない。これは、 $\lambda < 0.8321$ の値が小さいとき、PSO のシステムは安定的となり、最適解に到達する前に特

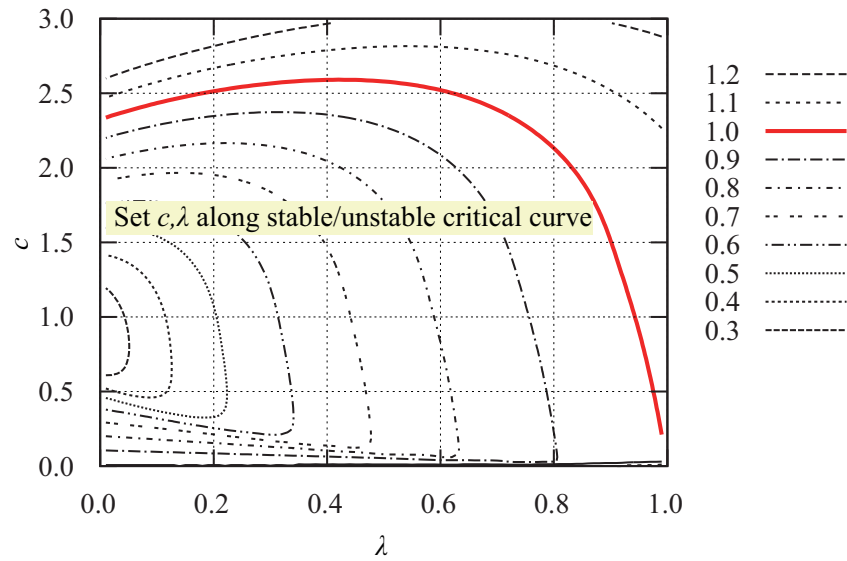


Fig. 3.6 Parameter settings map.

(3) Comparison between sustainable search parameters

定の座標に収束しているためと考えられる。一方で、 λ の値が大きいと、再び PSO の最適化性能は悪化している。これは λ の値が小さいときとは逆に、PSO のシステムが不安定的となり、局所的探索が十分におこなわれていないためと考えられる。

3.5.4 持続的探索パラメータ同士の比較

Tables 3.17–3.22 および Figs. 3.19–3.24 に、Table 3.2 で示した持続的探索パラメータ同士 (Fig. 3.6) の最適化性能の比較結果を示す。Tables 3.17–3.22 および Figs. 3.19–3.24 の各図表より、持続的探索パラメータ同士の優劣は問題に応じて異なるものの、 λ が過度に小さい場合や大きい場合は最適化性能が悪化する傾向にあり、 λ が 0.4～0.6 程度の値のとき、全ベンチマーク問題を通して安定した性能を発揮することがわかった。

Table 3.5 Optimization performance of PSO for Prob. 1 ($N = 20, \alpha = 0.0$).

(1) Comparison between conventional parameters and sustainable search parameter

(a) Random coefficients link disabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	8.438e-109	1.174e-107	1.653e-126	1.664e-106
2	1.70000	1.70000	0.60000	3.098e-18	3.330e-17	3.534e-150	4.602e-16
3	2.00000	2.00000	0.83210	5.690e+00	2.671e+00	1.081e+00	1.323e+01

(b) Random coefficients link enabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	2.298e+00	1.140e+00	4.178e-01	7.178e+00
2	1.70000	1.70000	0.60000	3.660e+00	1.698e+00	7.383e-01	9.928e+00
3	2.00000	2.00000	0.83210	1.153e-04	9.456e-05	8.918e-06	6.572e-04

Table 3.6 Optimization performance of PSO for Prob. 2 ($N = 20, \alpha = 0.0$).

(1) Comparison between conventional parameters and sustainable search parameter

(a) Random coefficients link disabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	2.279e+00	2.545e+00	1.962e-05	1.253e+01
2	1.70000	1.70000	0.60000	1.257e+01	1.149e+01	3.533e-05	9.540e+01
3	2.00000	2.00000	0.83210	1.462e+03	8.280e+02	2.598e+02	4.443e+03

(b) Random coefficients link enabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	3.346e+02	1.730e+02	8.614e+01	1.035e+03
2	1.70000	1.70000	0.60000	5.138e+02	3.047e+02	9.819e+01	1.803e+03
3	2.00000	2.00000	0.83210	1.882e+01	9.685e+00	5.991e+00	7.579e+01

Table 3.7 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.0$).

(1) Comparison between conventional parameters and sustainable search parameter

(a) Random coefficients link disabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	-1.487e+03	4.166e+01	-1.567e+03	-1.397e+03
2	1.70000	1.70000	0.60000	-1.464e+03	5.011e+01	-1.567e+03	-1.312e+03
3	2.00000	2.00000	0.83210	-1.309e+03	6.520e+01	-1.443e+03	-1.168e+03

(b) Random coefficients link enabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	-1.281e+03	6.895e+01	-1.441e+03	-1.110e+03
2	1.70000	1.70000	0.60000	-1.241e+03	7.121e+01	-1.441e+03	-1.013e+03
3	2.00000	2.00000	0.83210	-1.419e+03	5.620e+01	-1.533e+03	-1.218e+03

Table 3.8 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.3$).

(1) Comparison between conventional parameters and sustainable search parameter

(a) Random coefficients link disabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	-1.457e+03	7.774e+01	-1.567e+03	-1.083e+03
2	1.70000	1.70000	0.60000	-1.448e+03	8.872e+01	-1.538e+03	-1.096e+03
3	2.00000	2.00000	0.83210	-1.159e+03	1.209e+02	-1.401e+03	-6.903e+02

(b) Random coefficients link enabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	-1.171e+03	1.381e+02	-1.483e+03	-6.712e+02
2	1.70000	1.70000	0.60000	-1.101e+03	1.616e+02	-1.451e+03	-4.327e+02
3	2.00000	2.00000	0.83210	-1.415e+03	1.136e+02	-1.538e+03	-1.038e+03

Table 3.9 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.0$).

(1) Comparison between conventional parameters and sustainable search parameter

(a) Random coefficients link disabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	2.402e+01	7.025e+00	1.094e+01	4.577e+01
2	1.70000	1.70000	0.60000	2.570e+01	6.990e+00	8.955e+00	4.577e+01
3	2.00000	2.00000	0.83210	7.675e+01	1.843e+01	3.312e+01	1.265e+02

(b) Random coefficients link enabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	6.315e+01	1.607e+01	2.308e+01	1.053e+02
2	1.70000	1.70000	0.60000	7.231e+01	1.452e+01	3.990e+01	1.088e+02
3	2.00000	2.00000	0.83210	7.663e+00	2.451e+00	2.025e+00	1.417e+01

Table 3.10 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.3$).

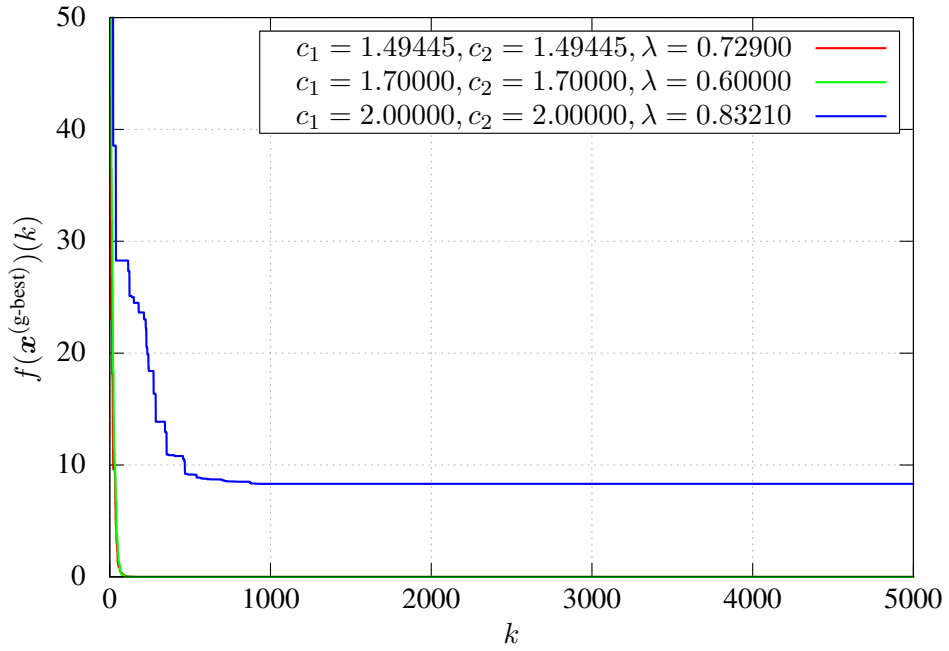
(1) Comparison between conventional parameters and sustainable search parameter

(a) Random coefficients link disabled

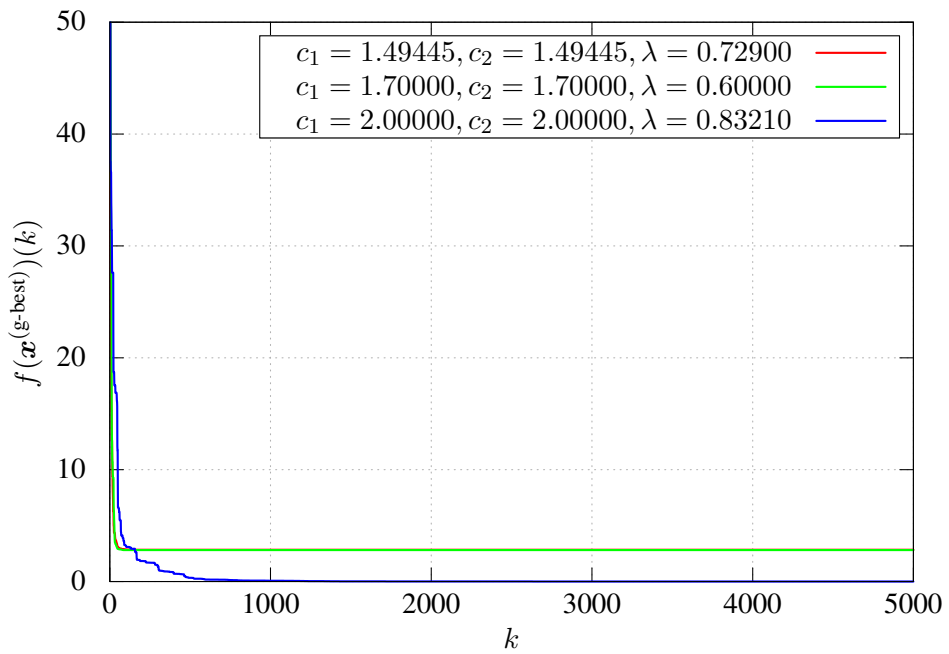
No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	3.971e+01	1.176e+01	1.592e+01	7.860e+01
2	1.70000	1.70000	0.60000	4.173e+01	1.261e+01	1.791e+01	8.955e+01
3	2.00000	2.00000	0.83210	1.091e+02	2.068e+01	6.272e+01	1.589e+02

(b) Random coefficients link enabled

No.	c_1	c_2	λ	mean	stdev	min	max
1	1.49445	1.49445	0.72900	7.163e+01	1.740e+01	3.025e+01	1.153e+02
2	1.70000	1.70000	0.60000	7.971e+01	1.700e+01	4.280e+01	1.308e+02
3	2.00000	2.00000	0.83210	3.711e+01	2.157e+01	1.112e+01	1.221e+02



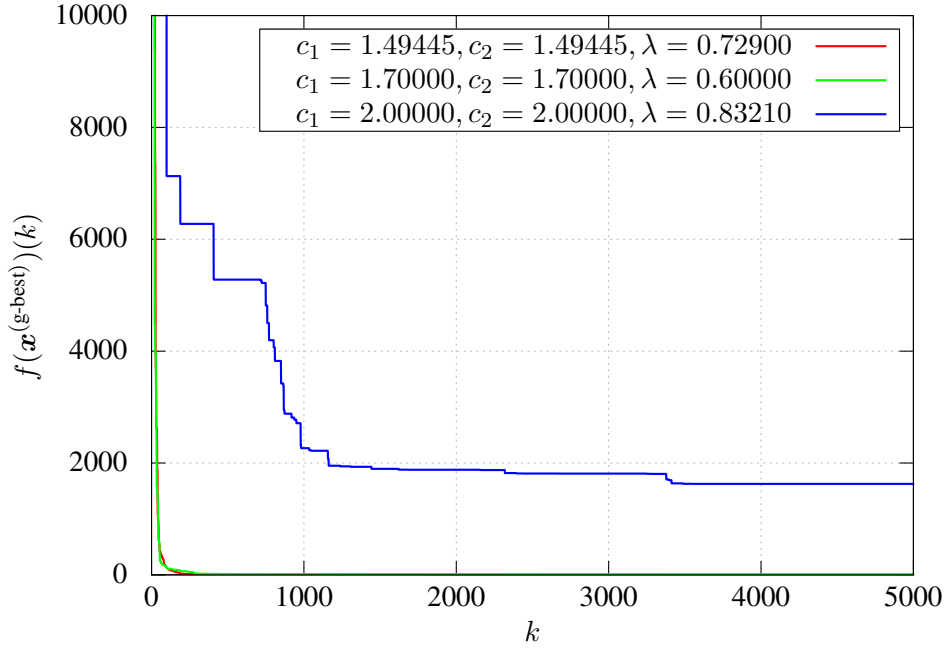
(a) Random coefficients link disabled



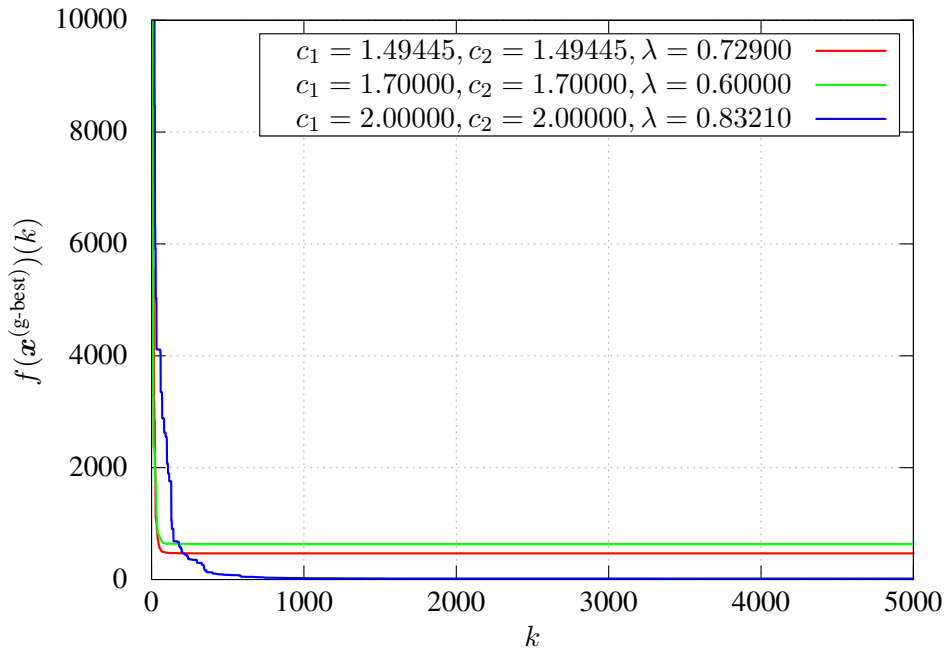
(b) Random coefficients link enabled

Fig. 3.7 Time history of $f(\mathbf{x}^{(g\text{-best})})$ of PSO for Prob. 1 ($N = 20, \alpha = 0.0$).

(1) Comparison between conventional parameters and sustainable search parameter



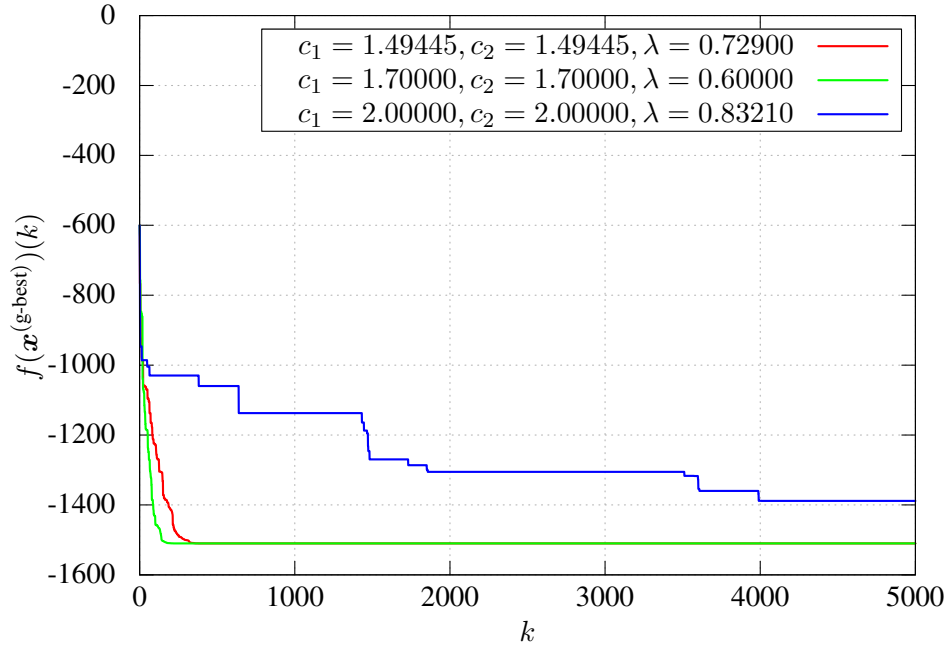
(a) Random coefficients link disabled



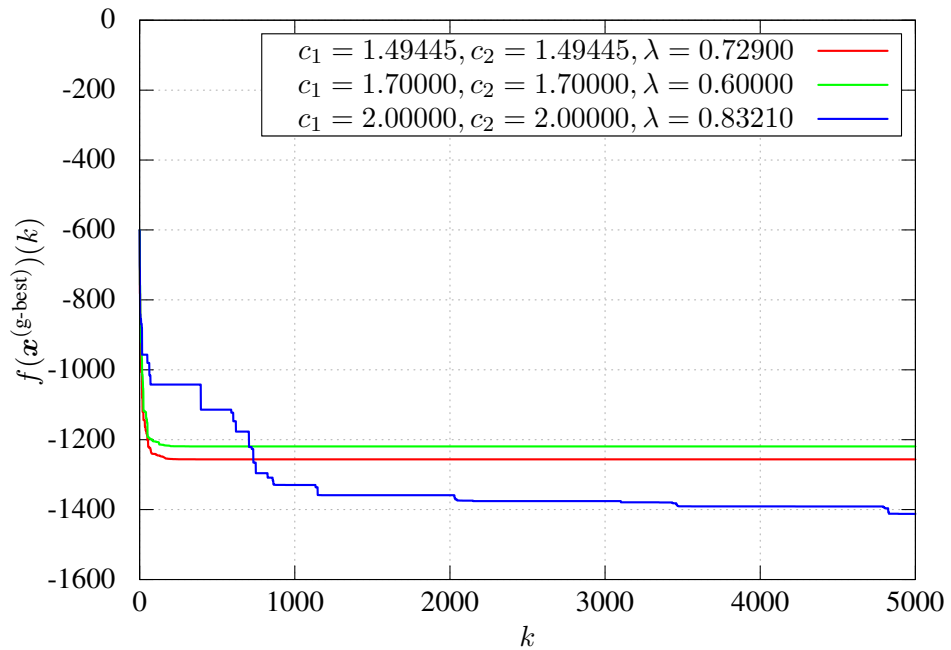
(b) Random coefficients link enabled

Fig. 3.8 Time history of $f(\mathbf{x}^{(g\text{-best})})$ of PSO for Prob. 2 ($N = 20, \alpha = 0.0$).

(1) Comparison between conventional parameters and sustainable search parameter



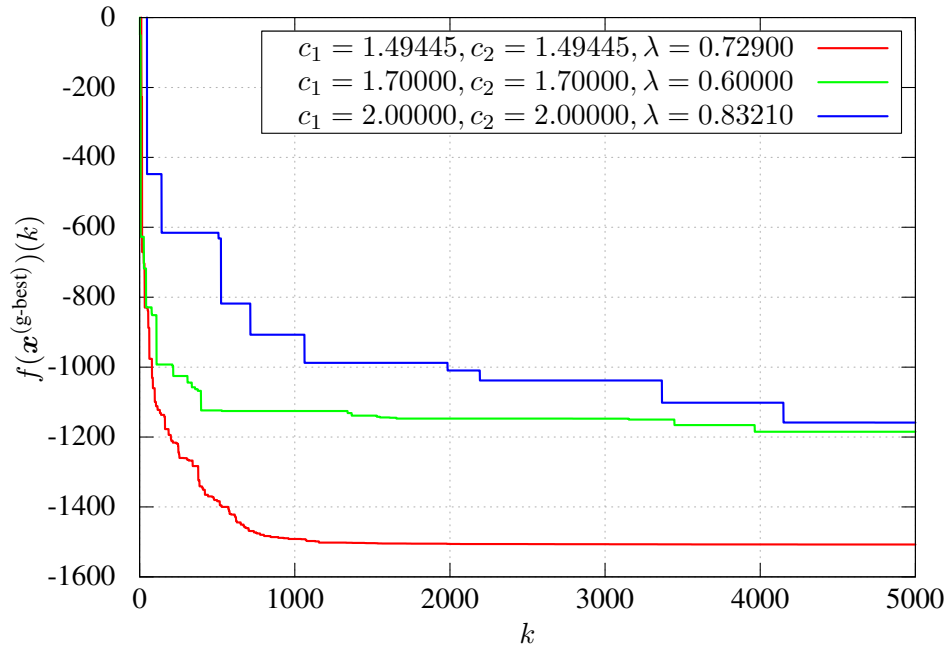
(a) Random coefficients link disabled



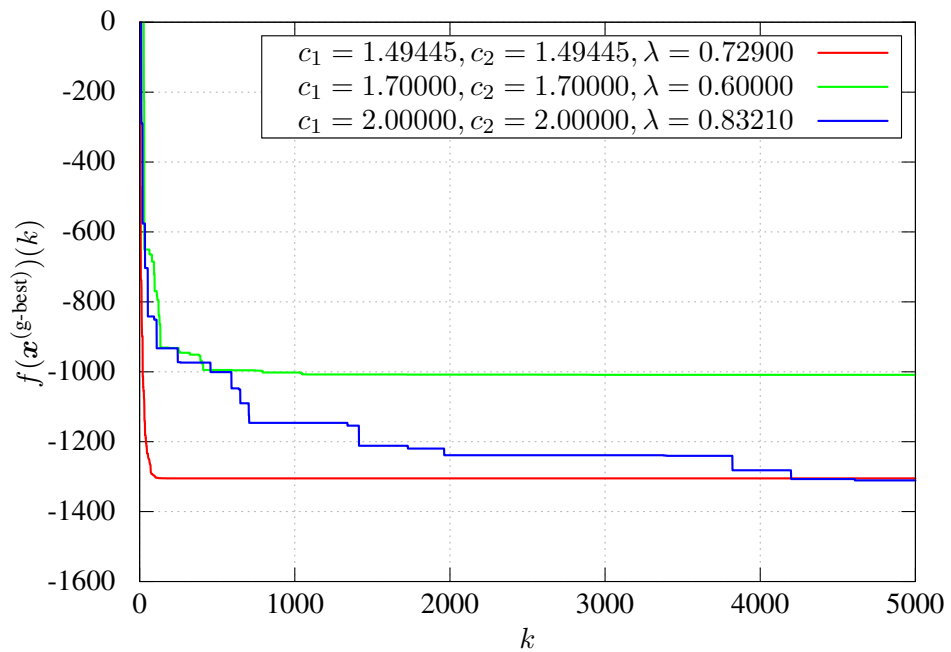
(b) Random coefficients link enabled

Fig. 3.9 Time history of $f(\mathbf{x}^{(g\text{-best})})$ of PSO for Prob. 3 ($N = 20, \alpha = 0.3$).

(1) Comparison between conventional parameters and sustainable search parameter

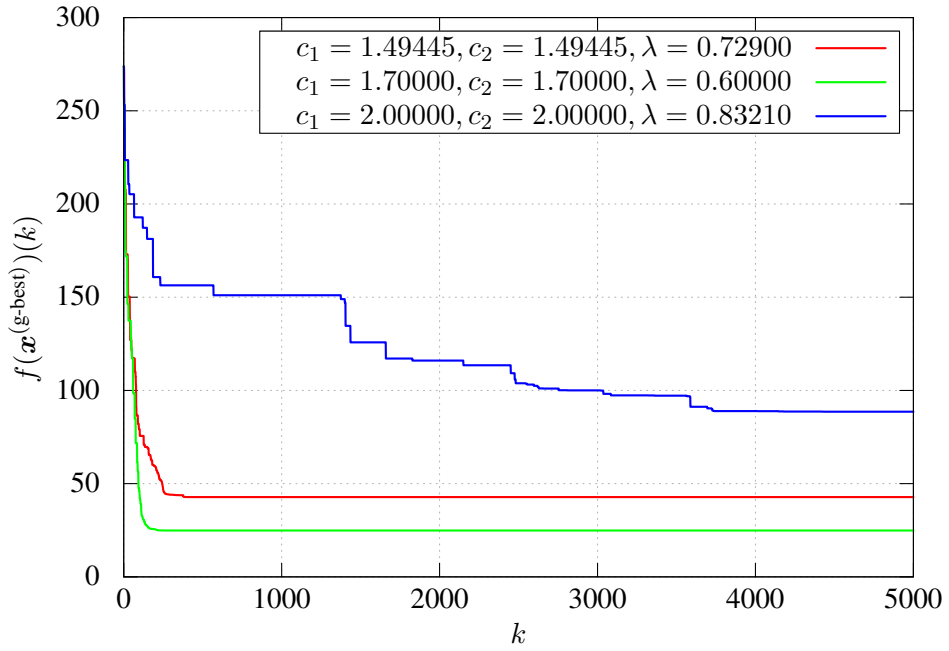


(a) Random coefficients link disabled

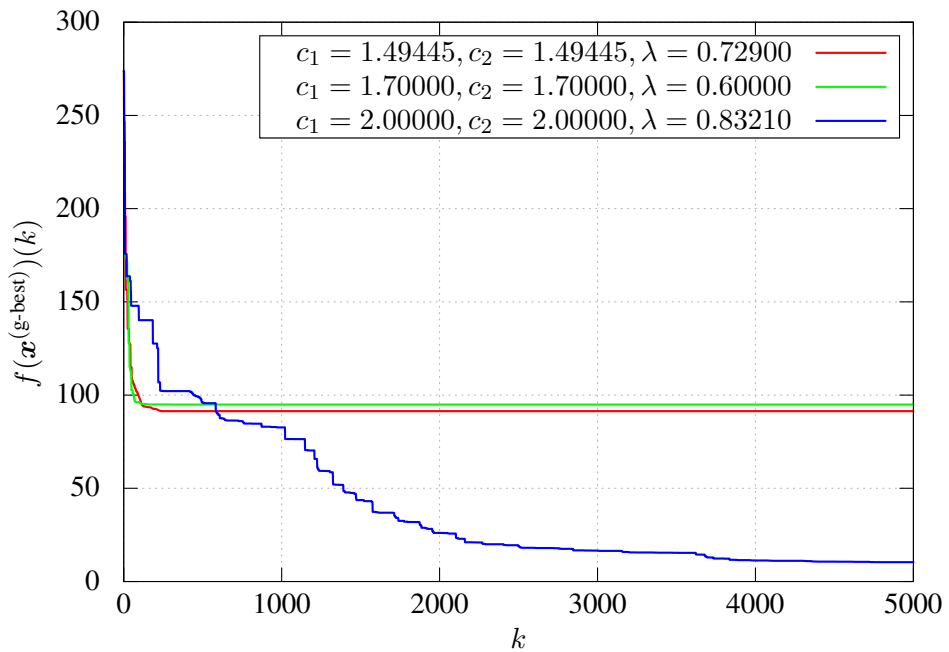


(b) Random coefficients link enabled

Fig. 3.10 Time history of $f(\mathbf{x}^{(g\text{-best})})$ of PSO for Prob. 3 ($N = 20, \alpha = 0.0$).
 (1) Comparison between conventional parameters and sustainable search parameter

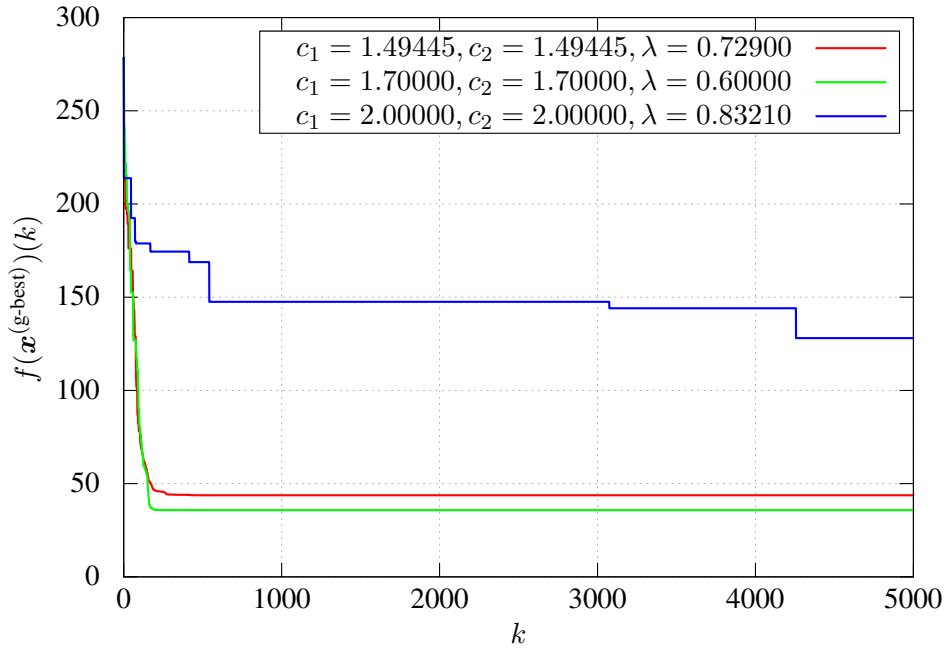


(a) Random coefficients link disabled

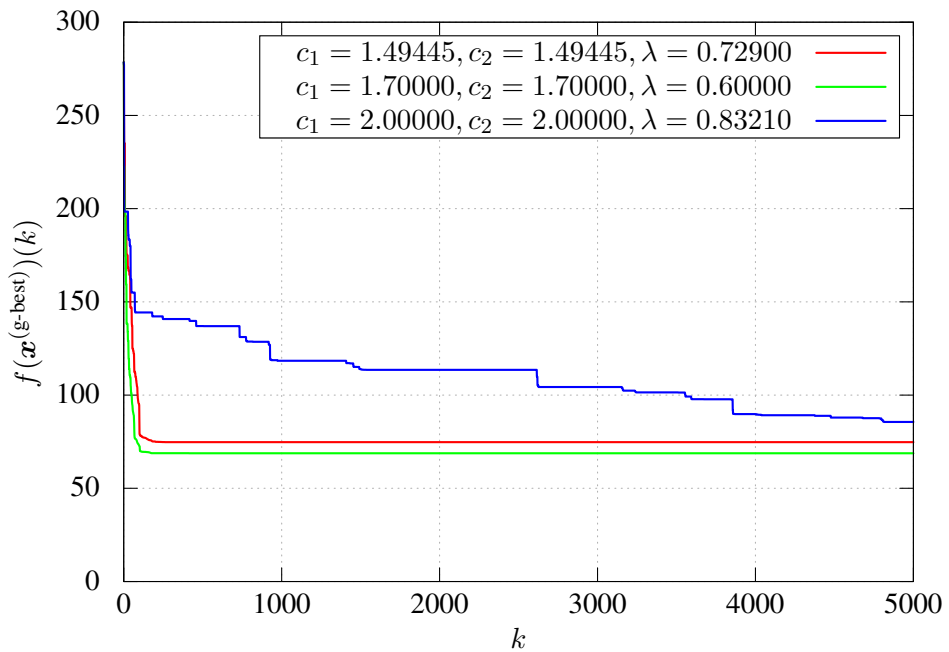


(b) Random coefficients link enabled

Fig. 3.11 Time history of $f(\mathbf{x}^{(g\text{-best})})$ of PSO for Prob. 4 ($N = 20, \alpha = 0.0$).
(1) Comparison between conventional parameters and sustainable search parameter



(a) Random coefficients link disabled



(b) Random coefficients link enabled

Fig. 3.12 Time history of $f(\mathbf{x}^{(g\text{-best})})$ of PSO for Prob. 4 ($N = 20, \alpha = 0.3$).
(1) Comparison between conventional parameters and sustainable search parameter

Table 3.11 Optimization performance of PSO for Prob. 1 ($N = 20, \alpha = 0.0$).

(2) Sensitivity analysis of sustainable search parameter

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.00000	2.00000	0.70000	8.964e-01	5.054e-01	1.701e-01	2.802e+00
2	2.00000	2.00000	0.72000	6.647e-01	3.917e-01	7.511e-02	3.459e+00
3	2.00000	2.00000	0.74000	2.896e-01	2.178e-01	2.245e-02	1.217e+00
4	2.00000	2.00000	0.76000	6.768e-02	6.874e-02	5.905e-03	5.245e-01
5	2.00000	2.00000	0.78000	6.023e-03	5.070e-03	4.488e-04	3.370e-02
6	2.00000	2.00000	0.80000	5.645e-04	4.777e-04	4.393e-05	3.301e-03
7	2.00000	2.00000	0.82000	1.127e-04	8.203e-05	9.733e-06	6.579e-04
8	2.00000	2.00000	0.83210	1.153e-04	9.456e-05	8.918e-06	6.572e-04
9	2.00000	2.00000	0.84000	1.701e-04	1.221e-04	9.663e-06	6.958e-04
10	2.00000	2.00000	0.86000	1.151e-03	7.958e-04	1.797e-04	7.250e-03
11	2.00000	2.00000	0.88000	1.504e-02	8.514e-03	2.430e-03	5.095e-02
12	2.00000	2.00000	0.90000	2.033e-01	1.082e-01	3.300e-02	5.809e-01

Table 3.12 Optimization performance of PSO for Prob. 2 ($N = 20, \alpha = 0.0$).

(2) Sensitivity analysis of sustainable search parameter

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.00000	2.00000	0.70000	1.543e+02	8.477e+01	2.447e+01	4.940e+02
2	2.00000	2.00000	0.72000	1.156e+02	6.403e+01	2.287e+01	3.913e+02
3	2.00000	2.00000	0.74000	6.720e+01	3.634e+01	2.076e+01	2.025e+02
4	2.00000	2.00000	0.76000	3.267e+01	1.950e+01	1.524e+01	9.874e+01
5	2.00000	2.00000	0.78000	2.100e+01	1.258e+01	1.201e+01	9.812e+01
6	2.00000	2.00000	0.80000	1.853e+01	7.690e+00	1.002e+01	7.208e+01
7	2.00000	2.00000	0.82000	1.835e+01	8.856e+00	9.202e+00	7.576e+01
8	2.00000	2.00000	0.83210	1.882e+01	9.685e+00	5.991e+00	7.579e+01
9	2.00000	2.00000	0.84000	1.881e+01	8.739e+00	4.596e+00	7.547e+01
10	2.00000	2.00000	0.86000	2.108e+01	1.295e+01	9.168e+00	1.073e+02
11	2.00000	2.00000	0.88000	3.061e+01	2.092e+01	1.540e+01	1.587e+02
12	2.00000	2.00000	0.90000	7.964e+01	3.996e+01	2.297e+01	2.705e+02

Table 3.13 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.0$).

(2) Sensitivity analysis of sustainable search parameter

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.00000	2.00000	0.70000	-1.431e+03	5.477e+01	-1.552e+03	-1.296e+03
2	2.00000	2.00000	0.72000	-1.458e+03	5.850e+01	-1.562e+03	-1.250e+03
3	2.00000	2.00000	0.74000	-1.495e+03	4.740e+01	-1.566e+03	-1.324e+03
4	2.00000	2.00000	0.76000	-1.526e+03	3.586e+01	-1.566e+03	-1.398e+03
5	2.00000	2.00000	0.78000	-1.536e+03	3.575e+01	-1.567e+03	-1.416e+03
6	2.00000	2.00000	0.80000	-1.531e+03	3.076e+01	-1.566e+03	-1.418e+03
7	2.00000	2.00000	0.82000	-1.470e+03	4.420e+01	-1.554e+03	-1.328e+03
8	2.00000	2.00000	0.83210	-1.419e+03	5.620e+01	-1.533e+03	-1.218e+03
9	2.00000	2.00000	0.84000	-1.369e+03	5.519e+01	-1.511e+03	-1.215e+03
10	2.00000	2.00000	0.86000	-1.284e+03	6.067e+01	-1.442e+03	-1.137e+03
11	2.00000	2.00000	0.88000	-1.203e+03	5.520e+01	-1.376e+03	-1.069e+03
12	2.00000	2.00000	0.90000	-1.155e+03	4.824e+01	-1.286e+03	-1.029e+03

Table 3.14 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.3$).

(2) Sensitivity analysis of sustainable search parameter

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.00000	2.00000	0.70000	-1.367e+03	7.914e+01	-1.524e+03	-1.163e+03
2	2.00000	2.00000	0.72000	-1.399e+03	8.413e+01	-1.536e+03	-1.145e+03
3	2.00000	2.00000	0.74000	-1.412e+03	9.323e+01	-1.537e+03	-1.053e+03
4	2.00000	2.00000	0.76000	-1.426e+03	8.530e+01	-1.538e+03	-1.162e+03
5	2.00000	2.00000	0.78000	-1.444e+03	8.265e+01	-1.538e+03	-1.135e+03
6	2.00000	2.00000	0.80000	-1.448e+03	8.330e+01	-1.538e+03	-1.107e+03
7	2.00000	2.00000	0.82000	-1.436e+03	1.101e+02	-1.538e+03	-1.016e+03
8	2.00000	2.00000	0.83210	-1.415e+03	1.136e+02	-1.538e+03	-1.038e+03
9	2.00000	2.00000	0.84000	-1.391e+03	1.284e+02	-1.538e+03	-1.028e+03
10	2.00000	2.00000	0.86000	-1.344e+03	1.409e+02	-1.535e+03	-1.008e+03
11	2.00000	2.00000	0.88000	-1.225e+03	1.268e+02	-1.501e+03	-8.830e+02
12	2.00000	2.00000	0.90000	-1.124e+03	1.204e+02	-1.374e+03	-7.937e+02

Table 3.15 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.0$).

(2) Sensitivity analysis of sustainable search parameter

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.00000	2.00000	0.70000	4.475e+01	1.239e+01	1.871e+01	9.425e+01
2	2.00000	2.00000	0.72000	4.016e+01	1.190e+01	1.443e+01	7.727e+01
3	2.00000	2.00000	0.74000	3.355e+01	1.164e+01	1.151e+01	7.208e+01
4	2.00000	2.00000	0.76000	2.068e+01	7.606e+00	6.966e+00	4.243e+01
5	2.00000	2.00000	0.78000	1.117e+01	3.993e+00	4.288e+00	3.576e+01
6	2.00000	2.00000	0.80000	8.707e+00	2.107e+00	3.055e+00	1.620e+01
7	2.00000	2.00000	0.82000	7.479e+00	2.408e+00	2.007e+00	1.405e+01
8	2.00000	2.00000	0.83210	7.663e+00	2.451e+00	2.025e+00	1.417e+01
9	2.00000	2.00000	0.84000	7.982e+00	2.441e+00	2.568e+00	1.481e+01
10	2.00000	2.00000	0.86000	1.197e+01	4.623e+00	3.704e+00	3.971e+01
11	2.00000	2.00000	0.88000	3.471e+01	1.603e+01	1.018e+01	9.611e+01
12	2.00000	2.00000	0.90000	7.945e+01	2.237e+01	3.606e+01	1.460e+02

Table 3.16 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.3$).

(2) Sensitivity analysis of sustainable search parameter

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.00000	2.00000	0.70000	5.981e+01	1.624e+01	2.336e+01	1.078e+02
2	2.00000	2.00000	0.72000	5.638e+01	1.459e+01	2.405e+01	9.708e+01
3	2.00000	2.00000	0.74000	5.144e+01	1.592e+01	1.558e+01	9.791e+01
4	2.00000	2.00000	0.76000	4.191e+01	1.448e+01	1.410e+01	8.995e+01
5	2.00000	2.00000	0.78000	3.509e+01	1.588e+01	1.042e+01	1.067e+02
6	2.00000	2.00000	0.80000	3.312e+01	1.436e+01	1.097e+01	1.031e+02
7	2.00000	2.00000	0.82000	3.650e+01	1.825e+01	9.017e+00	1.120e+02
8	2.00000	2.00000	0.83210	3.711e+01	2.157e+01	1.112e+01	1.221e+02
9	2.00000	2.00000	0.84000	3.786e+01	2.125e+01	1.263e+01	1.136e+02
10	2.00000	2.00000	0.86000	5.821e+01	3.286e+01	1.053e+01	1.318e+02
11	2.00000	2.00000	0.88000	9.558e+01	2.963e+01	2.219e+01	1.564e+02
12	2.00000	2.00000	0.90000	1.253e+02	1.836e+01	5.520e+01	1.578e+02

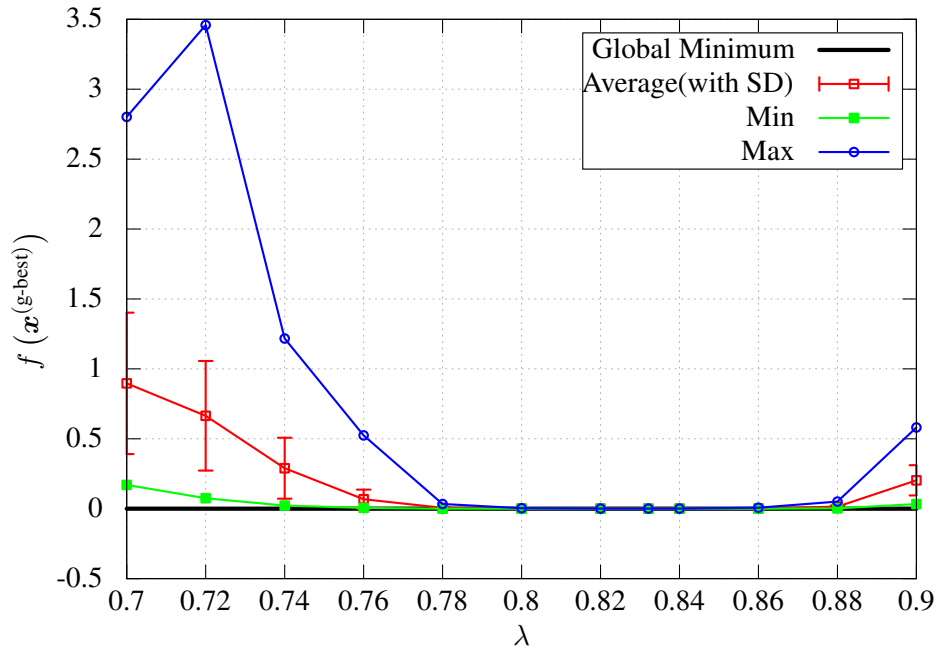


Fig. 3.13 Optimization performance of PSO for Prob. 1 ($N = 20, \alpha = 0.0$).

(2) Sensitivity analysis of sustainable search parameter

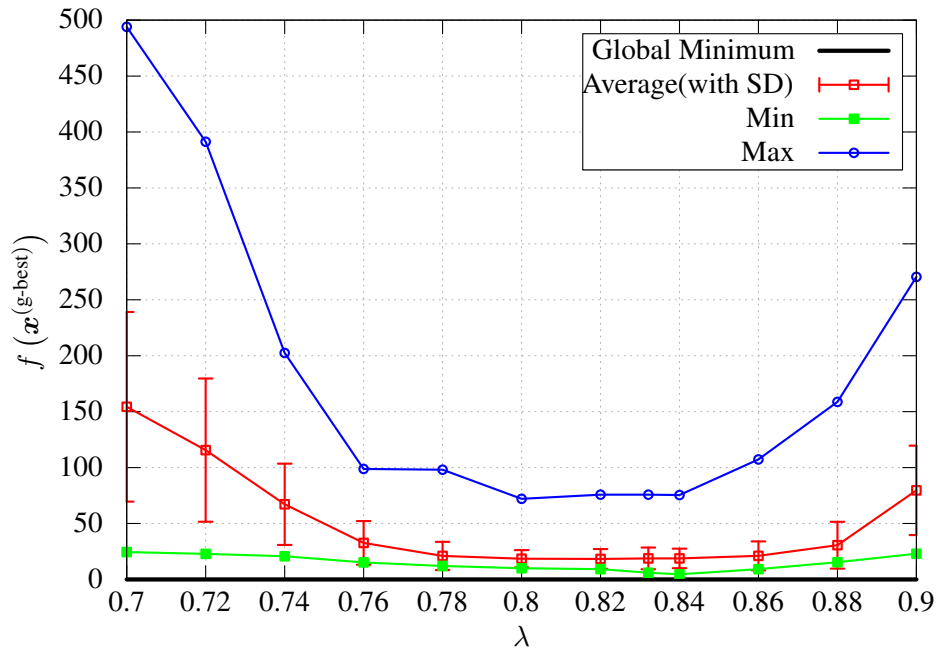


Fig. 3.14 Optimization performance of PSO for Prob. 2 ($N = 20, \alpha = 0.0$).

(2) Sensitivity analysis of sustainable search parameter

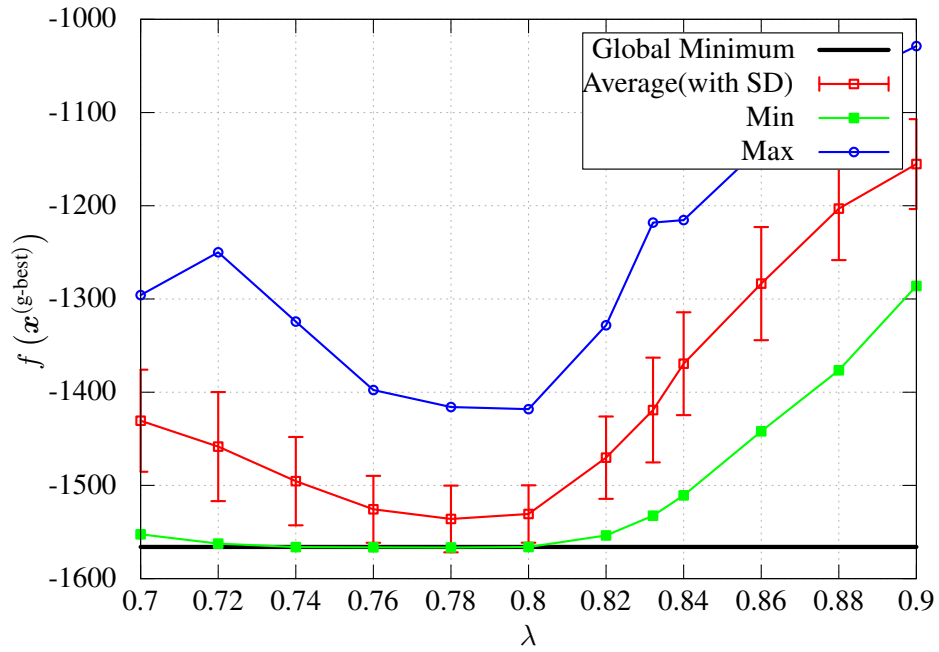


Fig. 3.15 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.0$).

(2) Sensitivity analysis of sustainable search parameter

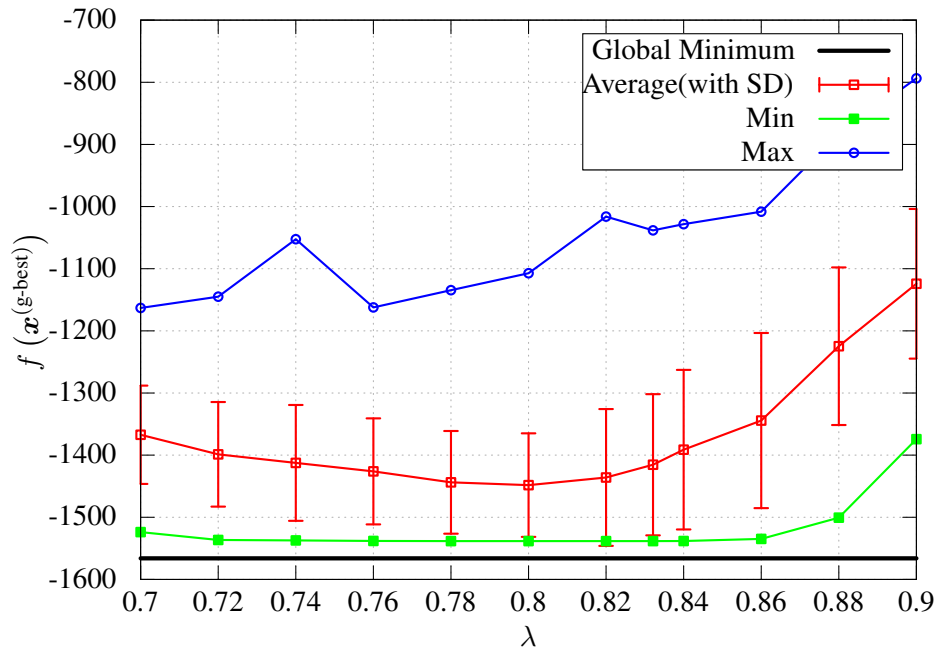


Fig. 3.16 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.3$).

(2) Sensitivity analysis of sustainable search parameter

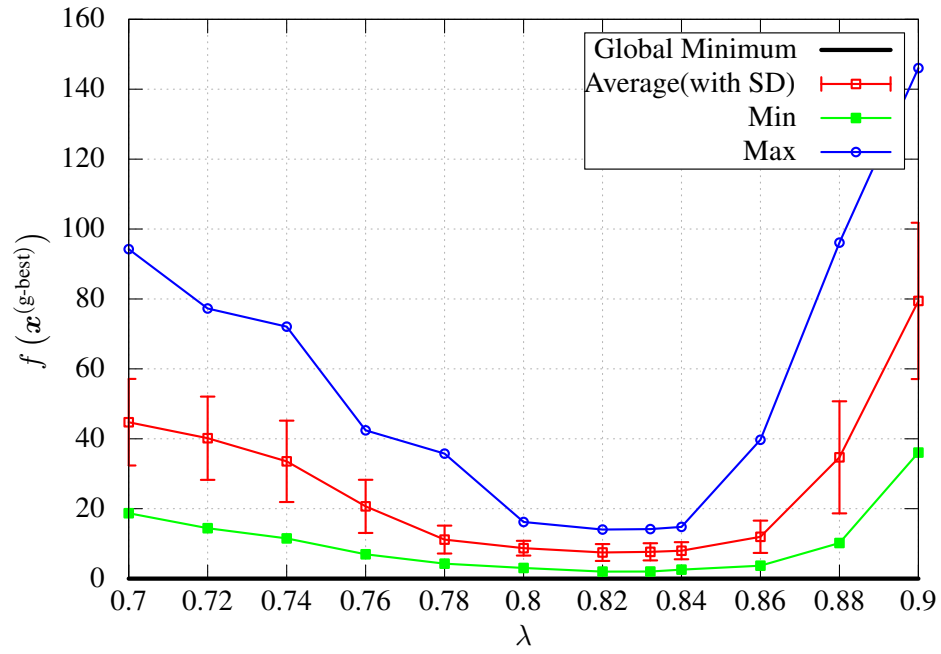


Fig. 3.17 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.0$).

(2) Sensitivity analysis of sustainable search parameter

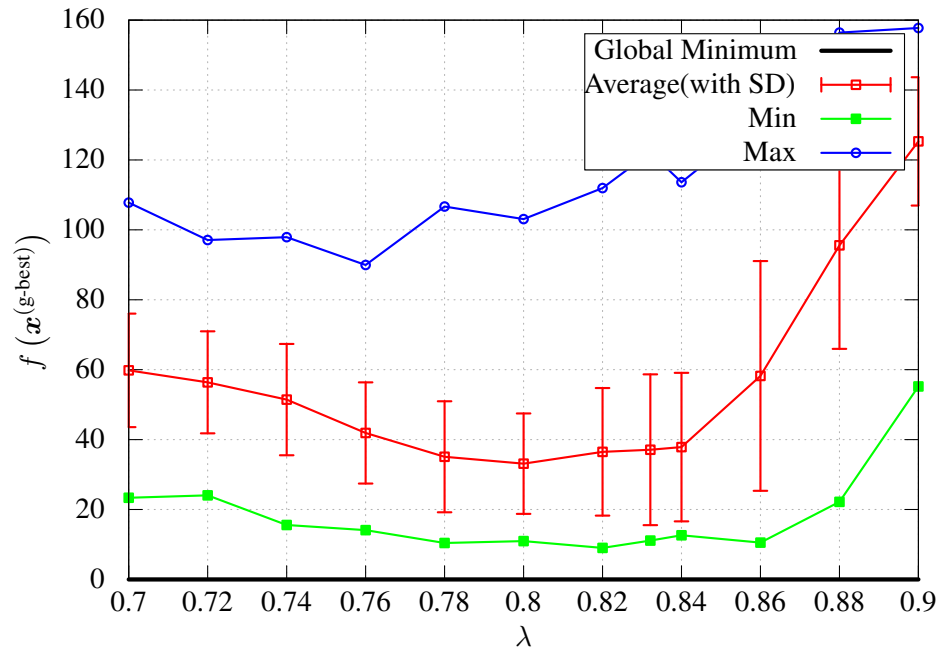


Fig. 3.18 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.3$).

(2) Sensitivity analysis of sustainable search parameter

Table 3.17 Optimization performance of PSO for Prob. 1 ($N = 20, \alpha = 0.0$).

(3) Comparison between sustainable search parameters

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.31960	2.31960	0.00000	1.618e-06	1.693e-06	9.456e-08	1.226e-05
2	2.43460	2.43460	0.10000	1.271e-07	1.085e-07	8.860e-09	6.025e-07
3	2.51410	2.51410	0.20000	9.164e-08	8.405e-08	6.742e-09	4.968e-07
4	2.56660	2.56660	0.30000	1.203e-07	1.205e-07	4.712e-09	7.890e-07
5	2.59040	2.59040	0.40000	2.953e-07	2.301e-07	1.904e-08	1.010e-06
6	2.57970	2.57970	0.50000	9.211e-07	9.832e-07	4.725e-08	6.789e-06
7	2.52170	2.52170	0.60000	2.982e-06	3.120e-06	1.141e-07	3.281e-05
8	2.39070	2.39070	0.70000	1.329e-05	1.174e-05	1.578e-06	9.392e-05
9	2.13230	2.13230	0.80000	6.611e-05	5.841e-05	4.602e-06	3.639e-04
10	1.51280	1.51280	0.90000	5.686e-04	4.319e-04	6.297e-05	2.796e-03

Table 3.18 Optimization performance of PSO for Prob. 2 ($N = 20, \alpha = 0.0$).

(3) Comparison between sustainable search parameters

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.31960	2.31960	0.00000	1.797e+01	1.166e+00	1.296e+01	1.950e+01
2	2.43460	2.43460	0.10000	1.831e+01	6.966e+00	1.197e+01	7.779e+01
3	2.51410	2.51410	0.20000	1.670e+01	1.375e+00	1.231e+01	1.950e+01
4	2.56660	2.56660	0.30000	1.699e+01	6.734e+00	6.918e+00	7.127e+01
5	2.59040	2.59040	0.40000	1.698e+01	7.978e+00	7.106e+00	7.279e+01
6	2.57970	2.57970	0.50000	1.647e+01	4.480e+00	7.103e+00	7.576e+01
7	2.52170	2.52170	0.60000	1.703e+01	5.362e+00	9.205e+00	7.597e+01
8	2.39070	2.39070	0.70000	1.686e+01	4.157e+00	7.173e+00	6.999e+01
9	2.13230	2.13230	0.80000	1.789e+01	6.210e+00	7.224e+00	7.167e+01
10	1.51280	1.51280	0.90000	2.169e+01	1.553e+01	6.239e+00	1.096e+02

Table 3.19 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.0$).

(3) Comparison between sustainable search parameters

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.31960	2.31960	0.00000	-1.452e+03	4.677e+01	-1.567e+03	-1.340e+03
2	2.43460	2.43460	0.10000	-1.538e+03	2.814e+01	-1.567e+03	-1.454e+03
3	2.51410	2.51410	0.20000	-1.562e+03	1.036e+01	-1.567e+03	-1.538e+03
4	2.56660	2.56660	0.30000	-1.566e+03	3.437e+00	-1.567e+03	-1.538e+03
5	2.59040	2.59040	0.40000	-1.567e+03	1.211e-03	-1.567e+03	-1.567e+03
6	2.57970	2.57970	0.50000	-1.567e+03	1.316e-02	-1.567e+03	-1.567e+03
7	2.52170	2.52170	0.60000	-1.566e+03	3.007e+00	-1.567e+03	-1.537e+03
8	2.39070	2.39070	0.70000	-1.531e+03	2.650e+01	-1.565e+03	-1.421e+03
9	2.13230	2.13230	0.80000	-1.443e+03	4.890e+01	-1.541e+03	-1.311e+03
10	1.51280	1.51280	0.90000	-1.367e+03	6.249e+01	-1.495e+03	-1.210e+03

Table 3.20 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.3$).

(3) Comparison between sustainable search parameters

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.31960	2.31960	0.00000	-1.461e+03	6.351e+01	-1.538e+03	-1.256e+03
2	2.43460	2.43460	0.10000	-1.469e+03	5.842e+01	-1.538e+03	-1.291e+03
3	2.51410	2.51410	0.20000	-1.467e+03	5.934e+01	-1.538e+03	-1.291e+03
4	2.56660	2.56660	0.30000	-1.462e+03	5.949e+01	-1.538e+03	-1.259e+03
5	2.59040	2.59040	0.40000	-1.474e+03	6.171e+01	-1.538e+03	-1.262e+03
6	2.57970	2.57970	0.50000	-1.467e+03	6.396e+01	-1.538e+03	-1.231e+03
7	2.52170	2.52170	0.60000	-1.463e+03	7.941e+01	-1.538e+03	-1.201e+03
8	2.39070	2.39070	0.70000	-1.443e+03	9.006e+01	-1.538e+03	-1.163e+03
9	2.13230	2.13230	0.80000	-1.416e+03	1.134e+02	-1.538e+03	-9.393e+02
10	1.51280	1.51280	0.90000	-1.402e+03	1.163e+02	-1.538e+03	-9.654e+02

Table 3.21 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.0$).

(3) Comparison between sustainable search parameters

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.31960	2.31960	0.00000	3.028e+01	8.878e+00	1.094e+01	5.870e+01
2	2.43460	2.43460	0.10000	9.432e+00	3.371e+00	1.991e+00	1.791e+01
3	2.51410	2.51410	0.20000	7.581e+00	3.126e+00	1.990e+00	1.990e+01
4	2.56660	2.56660	0.30000	6.614e+00	2.513e+00	9.958e-01	1.393e+01
5	2.59040	2.59040	0.40000	6.263e+00	2.492e+00	9.963e-01	1.293e+01
6	2.57970	2.57970	0.50000	6.167e+00	2.648e+00	1.942e-03	1.493e+01
7	2.52170	2.52170	0.60000	6.225e+00	2.169e+00	6.091e-03	1.294e+01
8	2.39070	2.39070	0.70000	6.631e+00	2.378e+00	1.038e+00	1.394e+01
9	2.13230	2.13230	0.80000	7.103e+00	2.384e+00	2.018e+00	1.501e+01
10	1.51280	1.51280	0.90000	8.407e+00	2.526e+00	2.310e+00	1.460e+01

Table 3.22 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.3$).

(3) Comparison between sustainable search parameters

No.	c_1	c_2	λ	mean	stdev	min	max
1	2.31960	2.31960	0.00000	4.354e+01	1.657e+01	1.094e+01	1.085e+02
2	2.43460	2.43460	0.10000	4.688e+01	1.574e+01	1.094e+01	8.457e+01
3	2.51410	2.51410	0.20000	4.252e+01	1.564e+01	1.293e+01	9.054e+01
4	2.56660	2.56660	0.30000	3.887e+01	1.408e+01	1.194e+01	9.114e+01
5	2.59040	2.59040	0.40000	3.319e+01	1.346e+01	9.950e+00	8.463e+01
6	2.57970	2.57970	0.50000	3.212e+01	1.063e+01	1.194e+01	6.469e+01
7	2.52170	2.52170	0.60000	3.075e+01	1.211e+01	1.094e+01	8.399e+01
8	2.39070	2.39070	0.70000	3.225e+01	1.443e+01	8.956e+00	9.840e+01
9	2.13230	2.13230	0.80000	3.622e+01	2.228e+01	3.995e+00	1.263e+02
10	1.51280	1.51280	0.90000	3.720e+01	1.998e+01	5.072e+00	9.958e+01

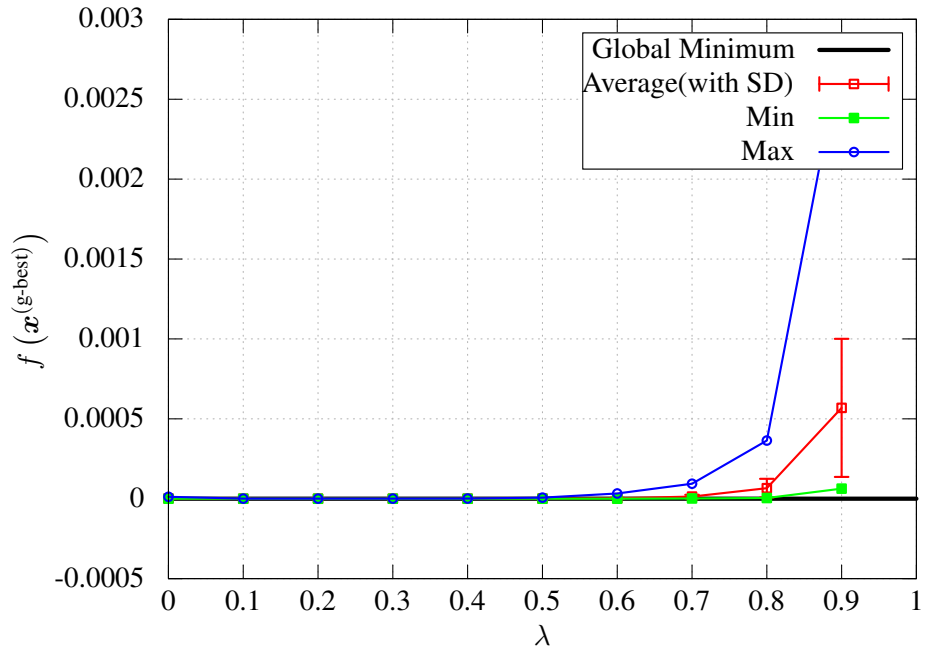


Fig. 3.19 Optimization performance of PSO for Prob. 1 ($N = 20, \alpha = 0.0$).
(3) Comparison between sustainable search parameters

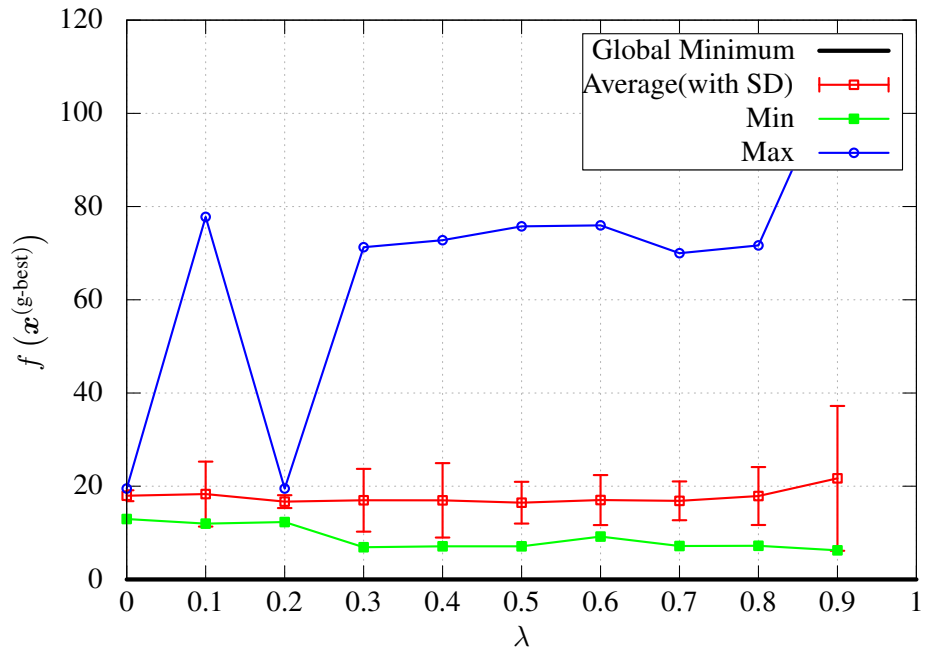


Fig. 3.20 Optimization performance of PSO for Prob. 2 ($N = 20, \alpha = 0.0$).
(3) Comparison between sustainable search parameters

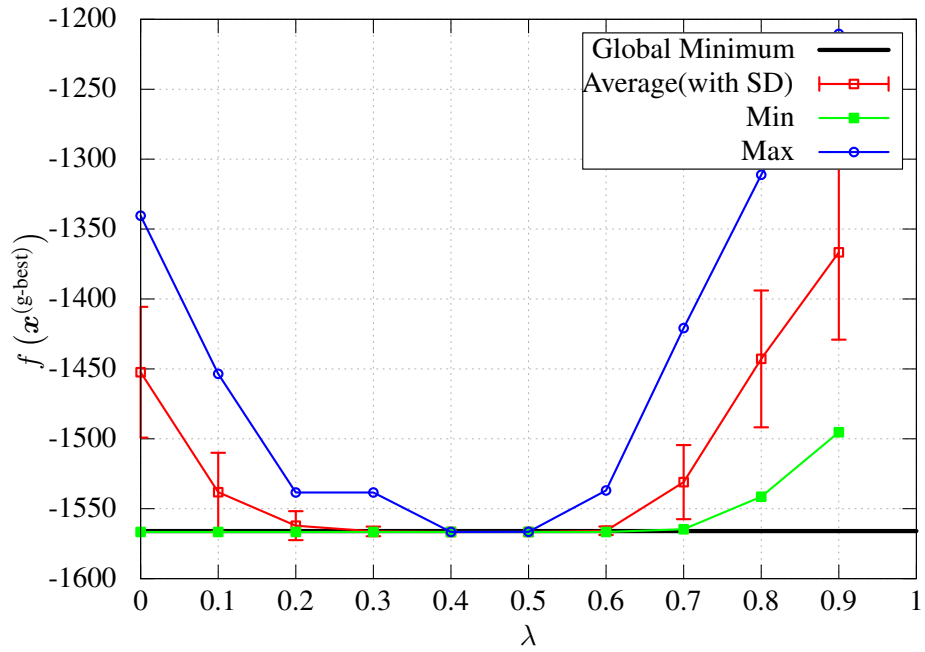


Fig. 3.21 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.0$).
(3) Comparison between sustainable search parameters

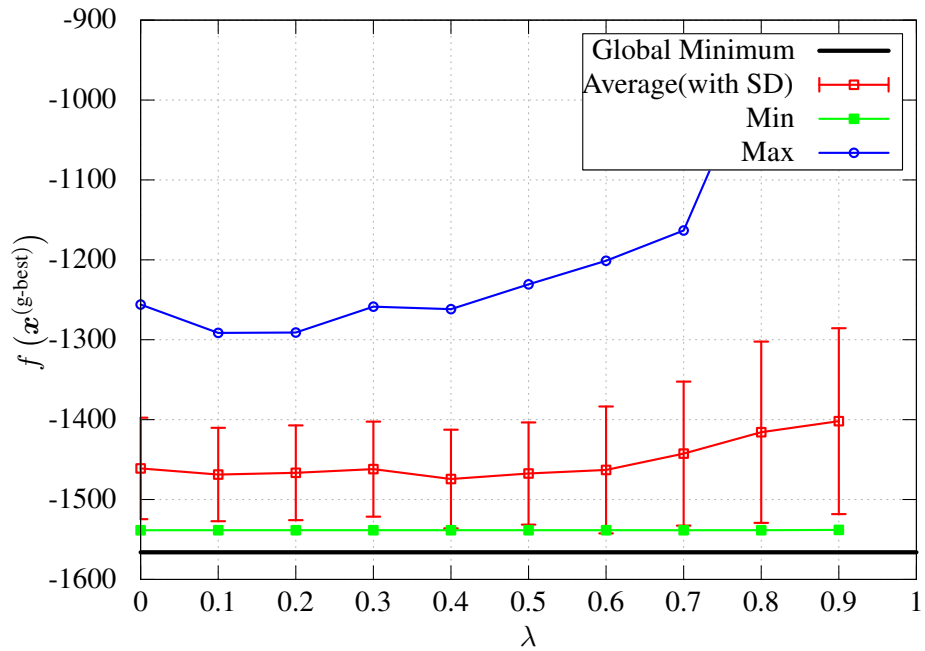


Fig. 3.22 Optimization performance of PSO for Prob. 3 ($N = 20, \alpha = 0.3$).
(3) Comparison between sustainable search parameters

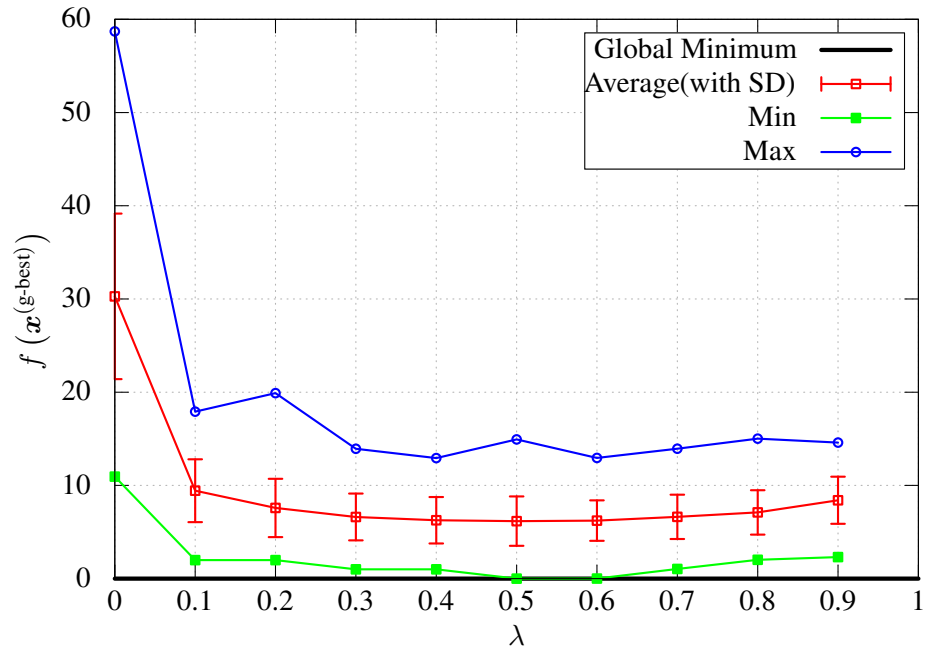


Fig. 3.23 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.0$).

(3) Comparison between sustainable search parameters

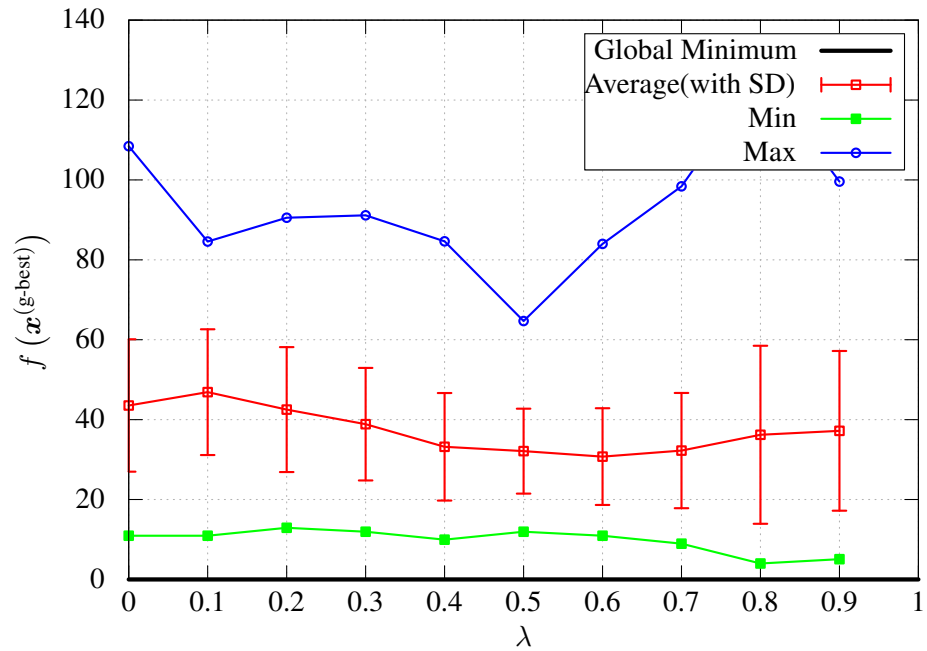


Fig. 3.24 Optimization performance of PSO for Prob. 4 ($N = 20, \alpha = 0.3$).

(3) Comparison between sustainable search parameters

3.6 本章のまとめ

本章では、代表的なメタヒューリスティクスである PSO に対して乱数の影響を考慮した安定性解析をおこない、その収束特性を定量的に評価した。また、安定性解析の知見をもとに、持続的に局所的探索と大域的探索を繰り返す「持続的探索パラメータ設定」を提案し、数値実験を通してその有効性を確認した。

ここで、本章で示した PSO の安定性解析手法（以下、提案解析手法）の適用範囲と、PSO の改良アルゴリズムへの寄与について言及しておく。本章では、PSO と等価な安定性を有する確率システム (3.5) を導入し、このシステムに対する安定性解析をおこなった。したがって、PSO の改良アルゴリズムのなかで (3.5) 式の形で記述できるものはすべて提案解析手法の適用対象となる。たとえば、g-best の代わりに隣接する探索番号の p-best への移流ベクトルを組み込んだアルゴリズムも解析可能である。また、PSO で用いる乱数係数 R_1, R_2 を $[0,1]$ 間の一様乱数としているが、この分布は任意のものでよい。乱数係数 R_1, R_2 がしたがう確率分布の確率密度関数をそれぞれ $f_1(x), f_2(x)$ とすると、(3.18) 式はより一般性の高い形として、

$$\int_0^1 f_1(R_1) dR_1 \int_0^1 f_2(R_2) dR_2 \int_0^{2\pi} \log \|M(R_1, R_2; \lambda, c_1, c_2) \mathbf{q}(\theta)\| \rho(\theta) d\theta < 0 \quad (3.35)$$

で記述できる。(3.35) 式を用いれば、たとえば PSO の乱数係数を正規乱数とした改良アルゴリズム^[115]の安定性解析も可能となる。

本章で示した提案解析手法は、パラメータがすべて定数であることを想定しているが、PSO の改良アルゴリズムのなかには、パラメータを探索状況や探索時刻に応じて動的に変化させるものも存在する^[29, 73, 83]。提案解析手法は、このようなアルゴリズムの動特性を明らかにすることはできないが、たとえばパラメータの調整や妥当性確認において有効と考えられる。具体的な例として、PSO の代表的な改良アルゴリズムである Inertia Weight Approach (IWA)^[73] のパラメータ設定について考える。IWA は、PSO のパラメータ λ を、最大探索時刻を K として、

$$\lambda(k) = \left(1 - \frac{k}{K}\right) (\lambda_{\max} - \lambda_{\min}) + \lambda_{\min} \quad (3.36)$$

と経時的に変化させることにより、大域的探索から局所的探索へ徐々にシフトさせていくアルゴリズムである。IWA のパラメータの数は、通常の PSO より 1 つ多く、慣性項の係数の初期値（最大値）と最終値（最小値） $\lambda_{\max}, \lambda_{\min}$ をそれぞれ適切に定める必要がある。文献[73]によると、 $\lambda_{\max}, \lambda_{\min}$ の推奨値パラメータはそれぞれ $\lambda_{\max} = 0.9, \lambda_{\min} = 0.4, c_1 = c_2 = 2.0$ であり、本章の安定性解析の結果より、 $c_1 = c_2 = 2.0$ の場合の安定／不安定の境界は $\lambda = 0.83210$ であり、推奨値パラメータは $\lambda_{\min} < 0.8321 < \lambda_{\max}$ を満たしており、所望の特性である「探索初期に大域的探索、探索末期に局所的探索」が実現可能であることを確認することができる。

本章で示した提案解析手法は、上述したように PSO の改良アルゴリズムに対する多少の柔軟性は有しているものの、すべての改良アルゴリズムに対して適用できるものではなく、アルゴリズムと安定性が等価な確率システムを (3.5) 式の形式で記述できないアルゴリズムに対しては、個別に新たな解析手法を検討する必要がある。メタヒューリスティク

スの数理的な取り扱いを目指すうえで、適用可能対象が一部に限られた解析手法は、設計の自由度を特徴とするメタヒューリスティクスと主旨を反するものと考えられ、より幅広いアルゴリズムに適用可能な解析手法の確立が期待される。

本章で示した提案解析手法およびその結果は、章中で強調したとおり、解くべき最適化問題の目的関数に依存しないものであり、幅広く適用可能なものである。しかしながら、最適化性能という観点からみれば、解析結果を踏まえ推奨されるパラメータ設定については目的関数に依存する。たとえば3.5.2項で考察したとおり、多峰性関数の最適化問題に対しては、システムを安定と不安定の境界におくようなパラメータ設定が有効であるが、一方でこのようなパラメータ設定は単峰性関数に対しては効果が薄く、システムを安定状態におくパラメータ設定を用いて、局所的最適解（＝大域的最適解）付近を集中的に探索するほうが良い結果が得られる。この点に関する今後の展望としては、安定性解析結果を踏まえつつ、さらに目的関数の形状³に応じた効果的なパラメータ設定指針の確立が期待される。

³たとえば単峰性／多峰性、探索次元小／大、悪条件問題／良条件問題など

第4章 Differential Evolution の収束性解析

4.1 本章について

本章では、第3章でPSOに対しておこなったのと同様、Differential Evolution (DE) の収束性とパラメータの関係を考察する。DE は、第2章で述べたとおり、パラメータに対する最適化性能が頑健であることが知られている。しかしながら、パラメータと収束性の関係については明らかにされておらず、不適当なパラメータを設定すると、局所的最適解でない点に収束する（初期収束）場合があることが指摘されている [31, 116]。

PSO と DE は、ともに目的関数値のみを参照する多点型確率的最適化アルゴリズムであるが、動特性の解析に際しては大きな違いが存在する。PSO に含まれる確率的要素は、計算の反復における探索点位置更新式の係数部分のみであり、動特性の解析では、個々の探索点を独立に扱うことができる。それに対して DE では、計算の反復において、各探索点の位置情報が確率的に組み替えられて新しい探索点位置を生成するため、個々の探索点の位置のみならず、探索点の分布を捉えた解析が必要となる。

DE の探索点分布を考慮した収束性解析としては、文献 [117] において、DE の計算過程のうち「交叉」とよばれる操作を除外し、探索点分布を正規分布と仮定した場合についての研究がなされている。これに対して本論文では、探索点分布の正規性を仮定せず、分布の特徴を有限次数のモーメントに代表させ、最大エントロピー法 [118] を用いて探索点分布をその形状も含めて逐次「推定」する方法を提案する。

本章の構成を以下に述べる。まず、4.2 節では、いくつかの仮定のもとで DE のアルゴリズムを変形した近似モデルに対する探索点分布の更新式を与える。4.3 節では、最大エントロピー法を用いて DE の探索点分布を逐次推定するアルゴリズムを構築し、4.4 節では、数値実験にて本章の解析手法の検証をおこない、また初期収束性を回避するためのパラメータ設定例をあげる。最後に 4.5 節では、本章の内容をまとめるとともに、提案解析手法の有する課題について述べる。

4.2 DE の近似モデルに対する探索点分布更新式

DE の探索点座標更新式は、第2章で示したとおり、

$$\mathbf{v}^{(p)}(k+1) = \mathbf{x}^{(r_1^{(p)}(k))}(k) + c_f \left(\mathbf{x}^{(r_2^{(p)}(k))}(k) - \mathbf{x}^{(r_3^{(p)}(k))}(k) \right) \quad (4.1a)$$

$$u_n^{(p)}(k+1) = \begin{cases} v_n^{(p)}(k+1), & (\theta^{(p)}(k) < \lambda) \cup (n = d^{(p)}(k)) \\ x_n^{(p)}(k), & \text{otherwise} \end{cases} \quad (4.1b)$$

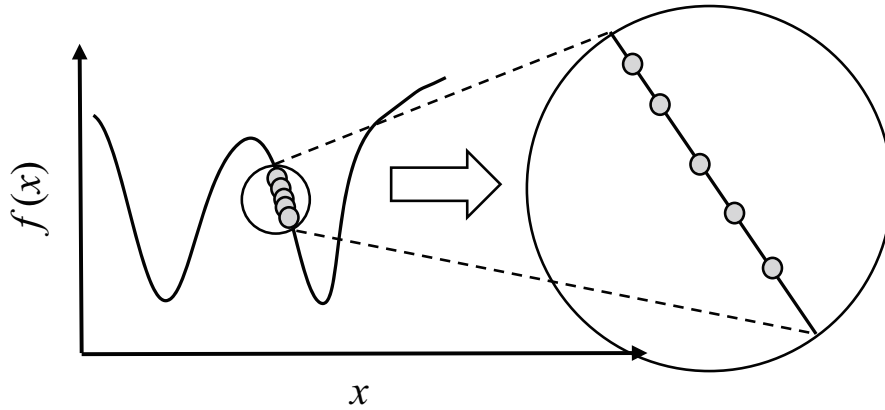


Fig. 4.1 Locally distributed search points.

$$x^{(p)}(k+1) = \begin{cases} u^{(p)}(k+1), & f(u^{(p)}(k+1)) < f(x^{(p)}(k)) \\ x^{(p)}(k), & \text{otherwise} \end{cases} \quad (4.1c)$$

で与えられる.

本章では, DE の探索点分布解析の端緒として 1 次元の問題を考える. また, DE の更新式は目的関数 $f(x)$ に依存した大小比較 (4.1c) を含んでおり, そのままの形で取り扱うことは困難である. そこで, 探索点が Fig. 4.1 のように探索空間上の局所的領域に密集した状況を想定した近似モデルを構築する. 4.1 節で述べたとおり, DE では初期収束が問題となるが, Fig. 4.1 のような状況から探索点分布が拡散する性質を示すのであれば, 初期収束は回避できると考えられる.

いま, 探索点がある座標 x^0 の近傍に密集しているものとする. ただし, $df(x^0)/dx \neq 0$ とする. 目的関数 $f(x)$ を x^0 の周りで Taylor 展開して 2 次以上の項を無視すると,

$$f(x) \simeq f(x^0) + \frac{df(x^0)}{dx}(x - x^0) \quad (4.2)$$

を得る. (4.2) 式を用いると, 1 次元問題に対する (4.1c) 式は,

$$x^{(p)}(k+1) \simeq \begin{cases} u^{(p)}(k+1), & \frac{df(x^0)}{dx}u^{(p)}(k+1) < \frac{df(x^0)}{dx}x^{(p)}(k) \\ x^{(p)}(k), & \text{otherwise} \end{cases} \quad (4.3)$$

と近似することができる. さらに, (4.3) 式は, x^0 における目的関数の微分係数 $df(x^0)/dx$ の正負に応じて,

(a) $df(x^0)/dx > 0$ のとき,

$$x^{(p)}(k+1) \simeq \min(u^{(p)}(k+1), x^{(p)}(k)) \quad (4.4)$$

(b) $df(x^0)/dx < 0$ のとき,

$$x^{(p)}(k+1) \simeq \max(u^{(p)}(k+1), x^{(p)}(k)) \quad (4.5)$$

となり、上述の探索点分布状況のもとで、目的関数の具体的な式 $f(x)$ を含まない形で「選択」プロセスを記述することができる。なお (4.4) 式は、設計変数 x の正負を逆とすれば (4.5) 式と同一の式となるため、探索点の動特性はケース (a)(b) で変わらない。そこで、以下ではケース (b) に代表させて探索点の動特性を議論していくこととする。

(4.5) 式による近似が成立する範囲における探索点分布の時間発展について考察する。まず、(4.5) 式における変数 $x^{(p)}, u^{(p)}, v^{(p)}$ を確率変数とみなして大文字で $X^{(p)}, U^{(p)}, V^{(p)}$ と書くこととする。すべての探索点は同一のアルゴリズムにより駆動することから、確率変数 $X^{(p)}, U^{(p)}, V^{(p)}$ を与える確率分布はすべての p に対して同一となると考えられる。そこで、確率変数 $X^{(p)}, U^{(p)}, V^{(p)}$ を定める確率分布の確率密度関数を、上付き添字 p を外して f_X, f_U, f_V と書くこととする。それぞれの探索点同士が独立とみなせるという仮定のもとで、更新式 (4.5) に対応する確率密度関数の更新式は、

$$f_V(x; k+1) = f_X(x; k) * f_X(c_f x; k) * f_X(-c_f x; k) \quad (4.6a)$$

$$f_U(x; k+1) = \lambda f_V(x; k+1) + (1 - \lambda) f_X(x; k) \quad (4.6b)$$

$$f_X(x; k+1) = f_U(x; k+1) \int_{-\infty}^x f_X(x'; k) dx' + f_X(x; k) \int_{-\infty}^x f_U(x'; k+1) dx' \quad (4.6c)$$

で与えられる。ここで、演算子 $*$ は畳み込み積分

$$f_A(sx) * f_B(tx) = \int_{-\infty}^{\infty} f_A\left(\frac{x - tx'}{s}\right) f_B(x') dx' \quad (4.7)$$

を意味する。

本章で想定した状況からの探索点分布の時間発展は、適当な初期分布 $f_X(x; 0)$ のもとで式 (4.6) 式を逐次解くことで得られる。ただし、(4.6) 式では各探索時刻の探索点分布 $f_X(x; k)$ 自体の不確実性は考慮しておらず、 $f_X(x; k)$ は探索時刻 k に対して一意に定まるものと暗黙に仮定している。この仮定を満たすためには、探索点数 P を十分大きくとる必要がある。

(4.6) 式の計算を解析的におこなうことは難しく、 x 空間を離散化した数値積分が必要である。とくに (4.6a) 式の畳み込みの計算を数値積分で直接計算する場合、 x 空間の離散化数の 2 乗に比例する計算量が必要となる。

一般に、確率分布という連続的な概念を数値的に扱うことは難しい。そこで本論文では、確率分布を厳密に計算するのではなく、その特徴のうち利用可能な情報のみを使って確率分布を最大エントロピー法により「推定」する考え方を採用する。具体的には、(4.6a) 式の計算で得られる確率分布を最大エントロピー法を用いて推定する。

4.3 最大エントロピー法に基づく探索点分布解析

4.3.1 最大エントロピー法

本節では、最大エントロピー法を用いて確率変数の線形結合に対する確率密度関数を推定する手法を概説し、またそれに基づいた DE の探索点分布解析について述べる。

$(-\infty, \infty)$ を台とする確率変数 X がしたがう確率分布 $\mathcal{P}_X$ を一意に推定することを考える。いま、確率分布 $\mathcal{P}_X$ について、性質 $l_0, l_1, \dots, l_M$ が既知であるとする。確率分布 $\mathcal{P}_X$

は、既知の性質 $l_0, l_1, \dots, l_M$ から必ずしも一意に定まるとは限らないため、特定の尺度に基づいて性質 $l_0, l_1, \dots, l_M$ を満たす確率分布 $\mathcal{P}_X$ の集合の要素に優劣をつけ、最適な確率分布を選ぶ必要がある。最大エントロピー法の主張は、確率分布を一意に推定するにあたり、既知の性質 $l_0, l_1, \dots, l_M$ から定まる部分以外については、「不確かさ」の尺度である(微分)エントロピー

$$S(f(\cdot)) = - \int_{-\infty}^{\infty} f(x) \log(f(x)) dx \quad (4.8)$$

が最大となるように定めるべきというものである¹。ここで、 $f(\cdot)$ は確率分布 $\mathcal{P}_X$ の確率密度関数である。確率分布 $\mathcal{P}_X$ に対する既知の情報としては、 m 次のモーメント

$$\mu_m = \int_{-\infty}^{\infty} x^m f(x) dx \quad (4.9)$$

が考えられる。0次²から M 次までのモーメントを既知とすると、最大エントロピー法に基づく確率分布決定問題は、各次のモーメントを等式制約条件とする最大化問題

$$\max_{f(\cdot)} S(f(\cdot)) = - \int_{-\infty}^{\infty} f(x) \log(f(x)) dx \quad (4.10a)$$

$$\text{subj.to } \mu_m = \int_{-\infty}^{\infty} x^m f(x) dx, \quad m = 0, \dots, M \quad (4.10b)$$

として定式化できる。最大化問題 (4.10) は Lagrange の未定乗数法を用いて解くことができる。Lagrange 関数 L を、

$$L(f(\cdot), \boldsymbol{\lambda}) = - \int_{-\infty}^{\infty} f(x) \log(f(x)) dx + \sum_{m=0}^M \lambda_m \left(\mu_m - \int_{-\infty}^{\infty} x^m f(x) dx \right) \quad (4.11)$$

とおき、Lagrange 関数 L が極値をとる条件を考える。まず、求める確率密度関数 $f(x) > 0$ に対して、適当な関数 $h: \mathbb{R} \rightarrow \mathbb{R}$ とスカラー $\epsilon \in \mathbb{R}$ を用いて、変分

$$\begin{aligned} \delta L(f(\cdot), h(\cdot), \boldsymbol{\lambda}, \epsilon) &= L(f(\cdot) + \epsilon h(\cdot), \boldsymbol{\lambda}) - L(f(\cdot), \boldsymbol{\lambda}) \\ &= \left[- \int_{-\infty}^{\infty} (f(x) + \epsilon h(x)) \log(f(x) + \epsilon h(x)) dx \right. \\ &\quad \left. + \sum_{m=0}^M \lambda_m \left(\mu_m - \int_{-\infty}^{\infty} x^m (f(x) + \epsilon h(x)) dx \right) \right] \\ &\quad - \left[- \int_{-\infty}^{\infty} f(x) \log(f(x)) dx + \sum_{m=0}^M \lambda_m \left(\mu_m - \int_{-\infty}^{\infty} x^m f(x) dx \right) \right] \end{aligned} \quad (4.12)$$

を定義する。Lagrange 関数 L が極値をとるときの確率密度関数 $f(x)$ の条件は、

$$\lim_{\epsilon \rightarrow 0} \delta L(f(\cdot), h(\cdot), \boldsymbol{\lambda}, \epsilon) = 0 \quad (4.13)$$

¹(4.8) 式の $S(f(\cdot))$ は、エントロピーが関数 f を引数とする汎関数であることを示している。

²本論文では、確率分布の正規化条件 $\int_{-\infty}^{\infty} f(x) dx = 1$ を既知の 0 次のモーメントとして考える。

で与えられる. (4.12) 式の右辺に対し, $\log(1+x)$ の Taylor 展開

$$\log(x+1) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} (x)^n}{n} \quad (4.14)$$

を用い, ϵ に関する 2 次以上の項を無視すると, 具体的な極値条件

$$\lim_{\epsilon \rightarrow 0} \left[- \int_{-\infty}^{\infty} \epsilon h(x) \left(1 + \log(f(x)) - \sum_{m=0}^M \lambda_m x^m \right) dx \right] = 0 \quad (4.15)$$

を得る. (4.15) 式において, 任意の関数 $h(x)$ に対して等号が成立する条件は,

$$1 + \log(f(x)) - \sum_{m=0}^M \lambda_m x^m = 0 \quad (4.16)$$

であり, (4.16) 式を $f(x)$ について解くと,

$$f(x) = \exp \left(-1 + \sum_{m=0}^M \lambda_m x^m \right) \quad (4.17)$$

を得る. 一方, Lagrange 関数 L が極値をとるときの Lagrange 乗数 λ の条件は,

$$\begin{aligned} \frac{\partial L(f(\cdot), \lambda)}{\partial \lambda_m} &= \mu_m - \int_{-\infty}^{\infty} x^m f(x) dx \\ &= \mu_m - \int_{-\infty}^{\infty} x^m \exp \left(-1 + \sum_{m=0}^M \lambda_m x^m \right) dx = 0, \quad m = 0, \dots, M \end{aligned} \quad (4.18)$$

で与えられる. 連立方程式 (4.18) の解となる Lagrange 乗数 λ_m ($m = 0, \dots, M$) を (4.17) 式に代入すると, M 次までのモーメントを制約条件としたもとで, エントロピー $S(f(\cdot))$ が最大となる確率密度関数 $f(x)$ の表式が具体的に得られる.

4.3.2 独立な確率変数の和と特性関数

確率変数 X を与える確率分布 $\mathcal{P}_X$ の確率密度関数 $f_X(x)$ に対する特性関数 $\psi_X(t)$ は,

$$\psi_X(t) = \int_{-\infty}^{\infty} \exp(itx) f_X(x) dx \quad (4.19)$$

で定義される. 特性関数を用いると, 独立な確率変数 $X_1, X_2, \dots, X_L$ の線形結合 $S = \sum_{l=1}^L a_l X_l$ に対して,

$$\psi_S(t) = \prod_{l=1}^L \psi_{X_l}(a_l t) \quad (4.20)$$

がなりたつ. (4.19) 式は, 指数関数部を Taylor 展開すると, 確率密度関数 $f_X(x)$ の m 次のモーメント μ_{Xm} を用いて,

$$\psi_X(t) = \sum_{m=0}^{\infty} \frac{(it)^m}{m!} \mu_{Xm} \quad (4.21)$$

と書くことができる。よって、(4.20) 式は確率密度関数 $f_{X_l}(x)$ の m 次のモーメント μ_{X_lm} を用いて、

$$\psi_S(t) = \prod_{l=1}^L \left(\sum_{m=0}^{\infty} \frac{(ia_l t)^m}{m!} \mu_{X_lm} \right) \quad (4.22)$$

と表現される。確率変数 $X_1, X_2, \dots, X_L$ の線形結合 $S = \sum_{l=1}^L a_l X_l$ を与える確率分布 $\mathcal{P}_S$ の確率密度関数 $f_S(x)$ に対する特性関数 $\psi_S(t)$ は、

$$\psi_S(t) = \sum_{m=0}^{\infty} \frac{(it)^m}{m!} \mu_{Sm} \quad (4.23)$$

であるから、(4.22) 式と (4.23) 式を t の多項式とみなして右辺の係数をそれぞれ比較することにより、確率変数 S の m 次のモーメント μ_{Sm} は、

$$\mu_{Sm} = \sum_{\sum_{l=1}^L n_l = m} \prod_{l=1}^L \frac{(a_l)^{n_l}}{n_l!} \mu_{X_l n_l} \quad (4.24)$$

と計算することができる。

4.3.3 DE の探索点分布解析手順

これまでの議論をまとめると、最大エントロピー法に基づく DE の探索点分布解析は Table 4.1 のような手順となる。

4.4 数値実験による提案解析手法の有効性検証

4.4.1 数値実験条件

本節では、まず、探索点分布を精度よく推定するために必要なモーメント最大次数について確認し、つぎに提案手法を用いて推定した探索点分布と、実際の DE の探索点分布の比較をおこない、提案手法の有効性を確認する。また、パラメータ設定と探索点分布の収束性／発散性の関係を調べ、初期収束を回避可能なパラメータ設定について考察する。

本数値実験で最適化する目的関数 $f(x)$ としては、(4.1c) 式を大域的に (4.5) 式で近似できる関数を採用する。4.3.3 項の手順から得られる探索点分布は、このような条件を満たす目的関数の最適化問題に対して DE を適用した場合の探索点分布に相当する。上述の条件を満たす目的関数は、1次元かつ単調減少であれば任意の関数を採用できる。本章では、DE を用いて解く最適化問題を、

$$\min_x f(x) = -x \quad (4.25)$$

とした。最適化問題 (4.25) は、無制約のもとで最適解をもたないため、適切にパラメータが設定された DE を適用すれば、探索点は $+\infty$ に発散する。しかし、パラメータが不適切な場合は、全探索点は有限な点に収束し探索が停止する、すなわち初期収束が生じることとなる。なお、以降の数値実験では共通して、初期状態 ($k = 0$) における探索点分布は平均 0、分散 1 の正規分布とした。

Table 4.1 Procedure of analysis of search point distribution of DE with maximum entropy method.

- Step 1** (初期条件) 反復回数 $k = 0$ とし, 適当な初期探索点分布 $f_X(x; 0)$ を定める.
- Step 2** (終了条件) 反復回数 k があらかじめ定められた上限値 K に達した場合は計算の反復を終了する.
- Step 3** 探索点分布 $f_X(x; k)$ のモーメント $\mu_{Xm}(m = 0, 1, \dots, M)$ を数値積分により計算する.
- Step 4** (4.24) 式を用いて, 確率密度関数 $f_V(x; k+1)$ のモーメント $\mu_{Vm}(m = 0, 1, \dots, M)$ を計算する.
- Step 5** 連立方程式 (4.12) を解き Lagrange 乗数 $\lambda_m(m = 0, 1, \dots, M)$ を求めて (4.13) 式に代入し, 確率密度関数 $f_V(x; k+1)$ を得る.
- Step 6** (4.6b) 式および (4.6c) 式を数値積分で計算し, $f_X(x; k+1)$ を求める.
- Step 7** $k = k+1$ とし, **Step 2** へ戻る.
-

4.4.2 モーメント最大次数

Fig. 4.2(a) および Fig. 4.2(b) に, $k = 300$ におけるモーメント最大次数 M と最適化問題 (4.25) に対する探索点分布の分散 σ^2 の関係を示す. Fig. 4.2 より, $M \geq 6$ の領域ではどのケースでも探索点の分散はほぼオーダーレベルで収束していることがわかる. 提案手法を用いた探索点分布の推定では, $M \geq 6$ とするのが望ましいと考えられる.

4.4.3 実際の DE の探索点分布との比較

Fig. 4.3 に, 最適化問題 (4.25) に対する, 提案手法を用いて推定した探索点分布の分散と, DE を適用した結果得られた探索点分布の分散 σ^2 の $k = 300$ までの推移を示す. 提案手法で考慮するモーメント最大次数 M は, 次数の打ち切りによる誤差を小さくするため, 4.4.2 項の検討結果に対して余裕をとり $M = 10$ とした.

Fig. 4.3 より, いずれのパラメータ設定に対しても, 探索点数 P が十分大きければ, 提案手法を用いて探索点分布を精度よく推定できていることがわかる. 一方, 探索点数 P が少ない場合は, 実際の DE の探索点分布の分散と提案手法による推定分布の分散に乖離がみられる. これは, 探索点が少ない場合, 分布を特徴付けるモーメントの値自体にばらつきが生じ, (4.6) 式における探索点分布の確率密度関数 $f_X(x; k)$ が一意に定まらず, それに起因する誤差が計算反復回数とともに発展することが原因であると考えられる.

Table 4.2 Critical parameter settings which enable DE to avoid premature convergence.

λ	0.300	0.500	0.700	0.900
c_f	0.316	0.363	0.406	0.441

4.4.4 初期収束回避可能なパラメータ設定

Fig. 4.4 に、最適化問題 (4.25) に対する、 $k = 300$ における、交叉率 $\lambda = 0.3, 0.5, 0.7, 0.9$ に対するスケールングファクタ c_f と探索点分布の分散 σ^2 の関係を示す。提案手法で考慮するモーメント最大次数 M は、4.4.3 項と同様 $M = 10$ とした。

Fig. 4.4 より、いずれの交叉率 λ に対しても、スケールングファクタ c_f が大きいほど探索点分布の分散が大きくなることがわかる。一方で、探索点分布の分散が大きくなる交叉率 λ はスケールングファクタ c_f の値によって異なり、スケールングファクタ c_f が小さい領域では交叉率 λ が小さいほど探索点分布の分散は大きく、逆にスケールングファクタ c_f が大きい領域では交叉率 λ が小さいほど探索点分布の分散は小さい傾向にあることがわかる。

Fig. 4.4 の図中、探索点分布の分散 σ^2 が初期状態より小さい、すなわち 1.0 未満となっているプロットは、対応するパラメータ設定 (c_f, λ) が収束的であることを示しており、逆に 1.0 より大きいプロットは、対応するパラメータ設定が発散的、すなわち初期収束回避可能なパラメータ設定であることを示している。これを受けて、各交叉率 ($\lambda = 0.3, 0.5, 0.7, 0.9$) 対して、収束と発散の境界を与える、すなわち探索点分布の分散が 1.0 となるようなスケールングファクタ c_f を調査した結果を Table 4.2 に示す。Table 4.2 の数値より、交叉率 λ を大きく設定した場合、それに応じてスケールングファクタ c_f を大きくとらなければ、初期収束を回避することができないことがわかる。このことは、本章の解析と数値実験によって確認された DE の性質の一つである。

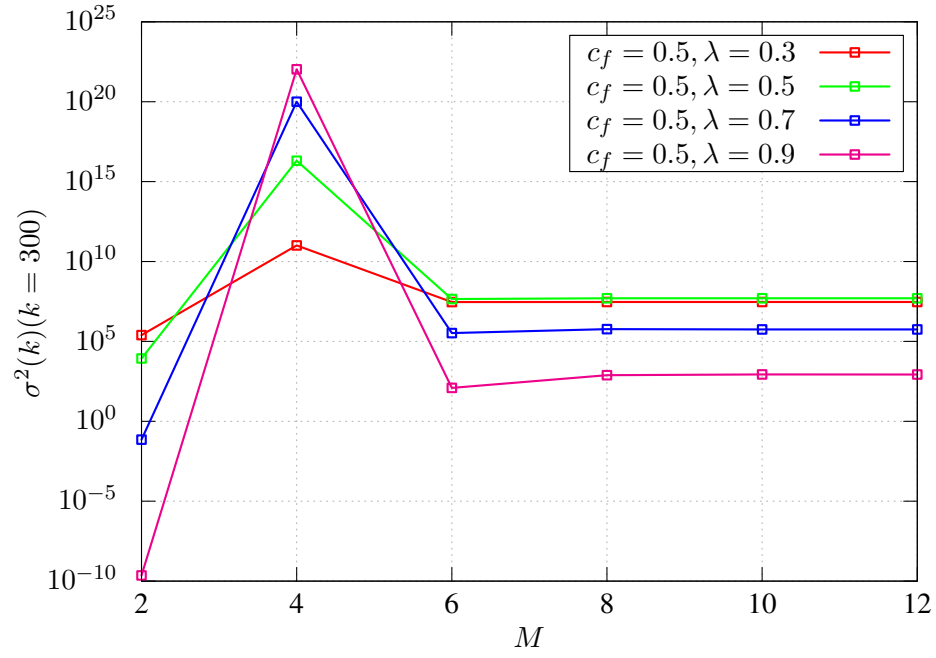
4.5 本章のまとめ

本章では、最大エントロピー法を用いた DE に対する探索点分布解析をおこない、数値実験により、いくつかの仮定のもとで精度の良い探索点分布解析が可能であることを確認した。また、その結果を用いて、1 次元の最適化問題に対して初期収束を回避可能なパラメータ設定の数値的条件の例を示した。

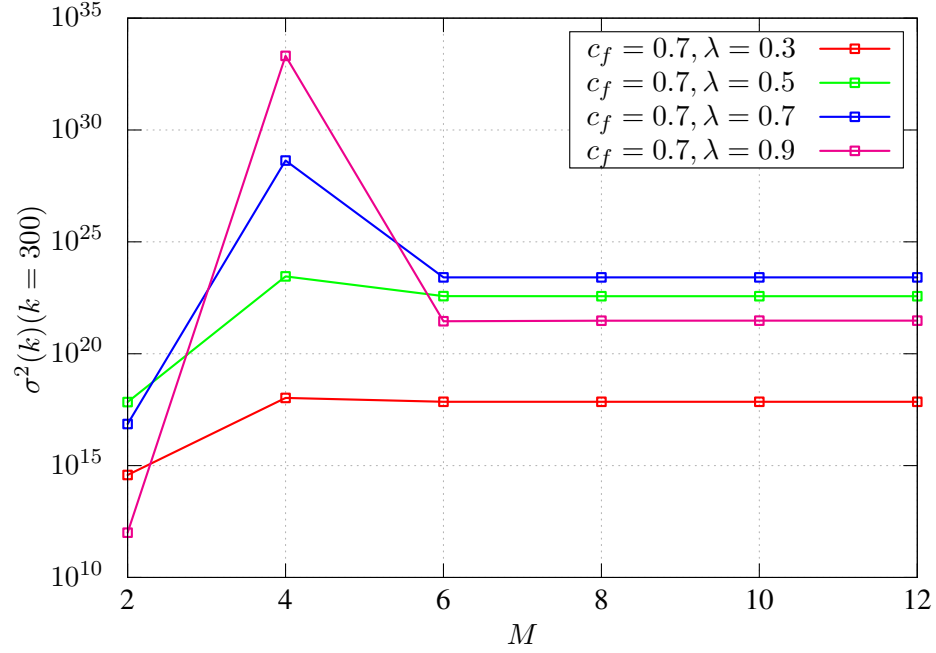
本章の解析の目指すところは、DE を用いて最適化をおこなったときに得られる解の最適性を保証することである。たとえば、初期収束の可能性のあるパラメータ設定で DE を適用した結果、すべての探索点が探索空間上のある 1 点に収束した場合、得られた解が局所的最適解であるのか、DE の枠組みのなかでは判断できない³。しかしながら、初期収束を回避可能なパラメータ設定であるという保証のもと、すべての探索点が探索空間上のある 1 点に収束しているのならば、それは局所的最適解であると判断することができ、実用上きわめて有利な情報となる。

³DE の枠組みに頼らないのであれば、たとえば数値微分を用いて勾配を計算することにより数値的に最適性のチェックが可能である。

しかしながら，本章では，(1) 探索点数が十分大きく，かつ (2) 1 次元の問題に対する問題のみを考える，といった仮定を設けたうえで DE 近似モデルを構築しており，現状では本論文の解析結果の適用範囲は限定的である．また，第 3 章における PSO の収束性解析と同様，アルゴリズムのバリエーションなどを考慮した解析手法として確立されていないという課題を残している．

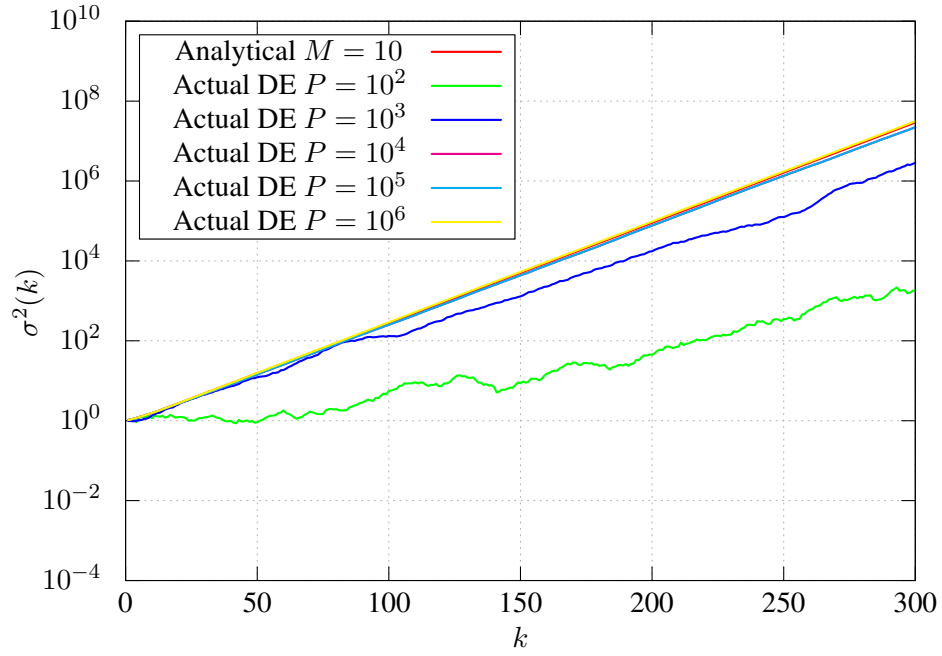


(a) $c_f = 0.5$

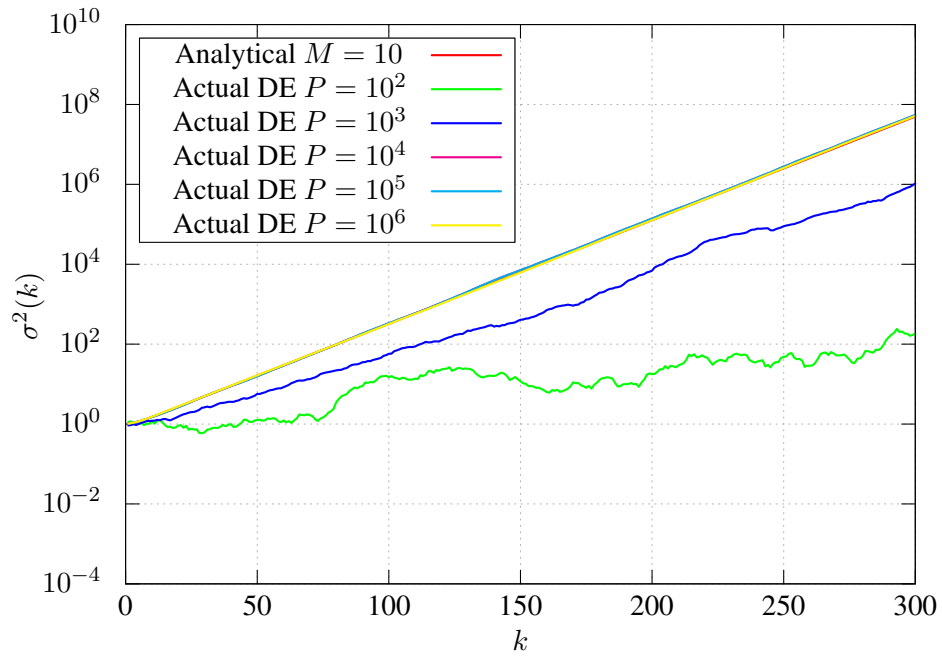


(b) $c_f = 0.7$

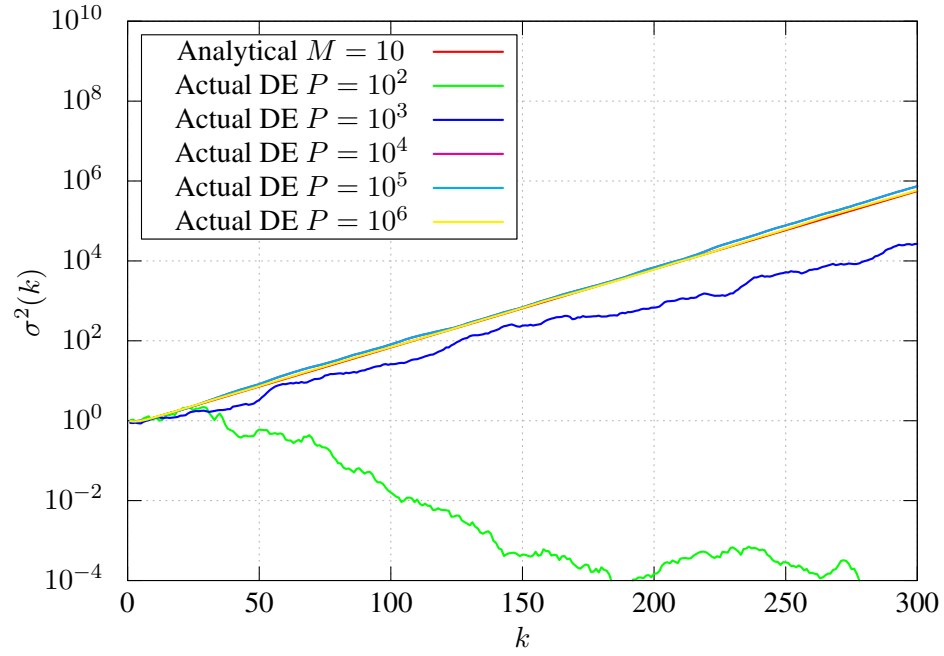
Fig. 4.2 Relationship between maximum moment order and estimated variance.



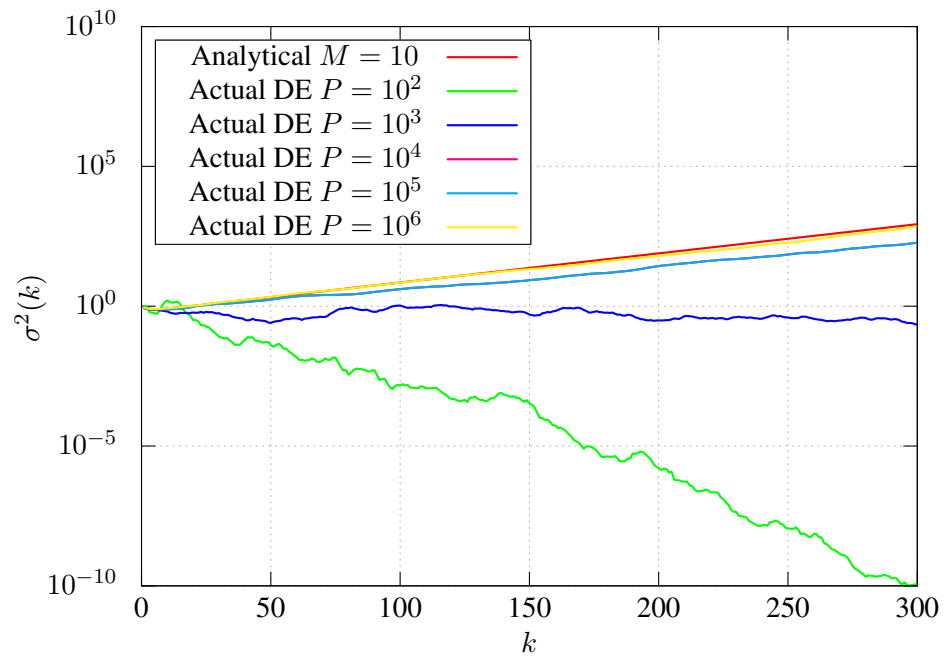
(a) $c_f = 0.5, \lambda = 0.3$



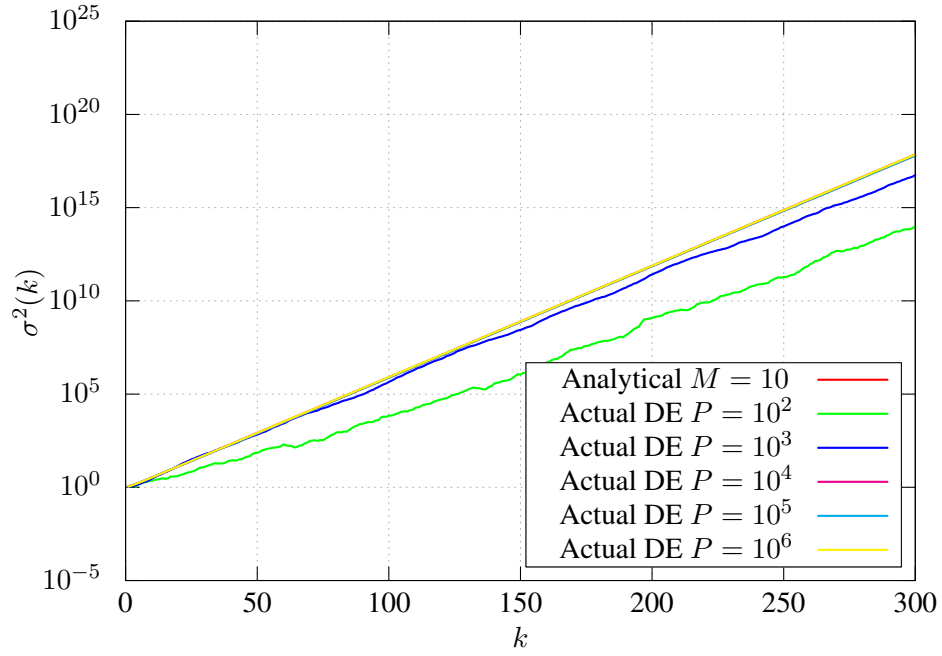
(b) $c_f = 0.5, \lambda = 0.5$



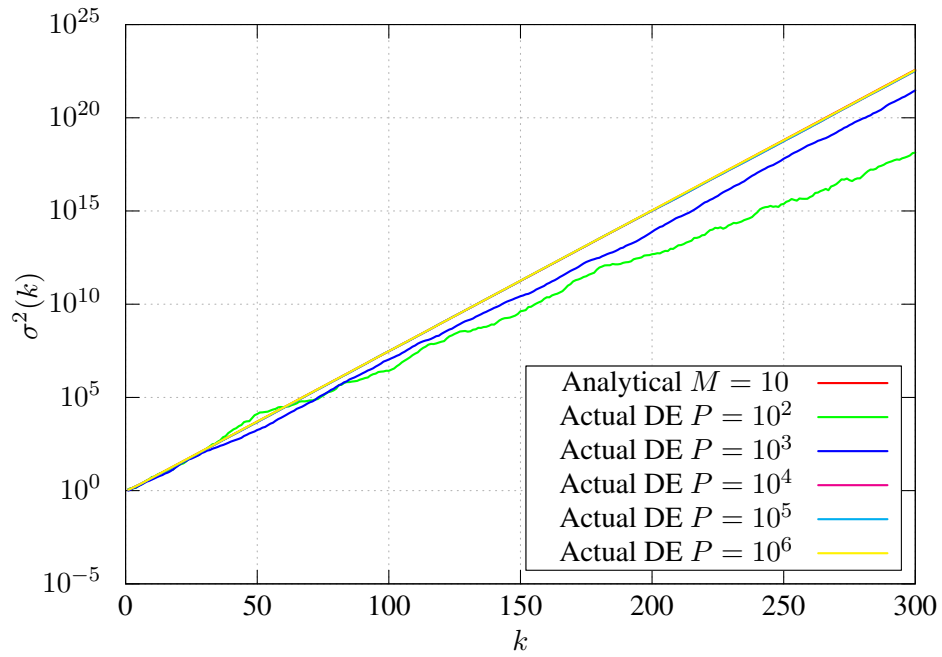
(c) $c_f = 0.5, \lambda = 0.7$



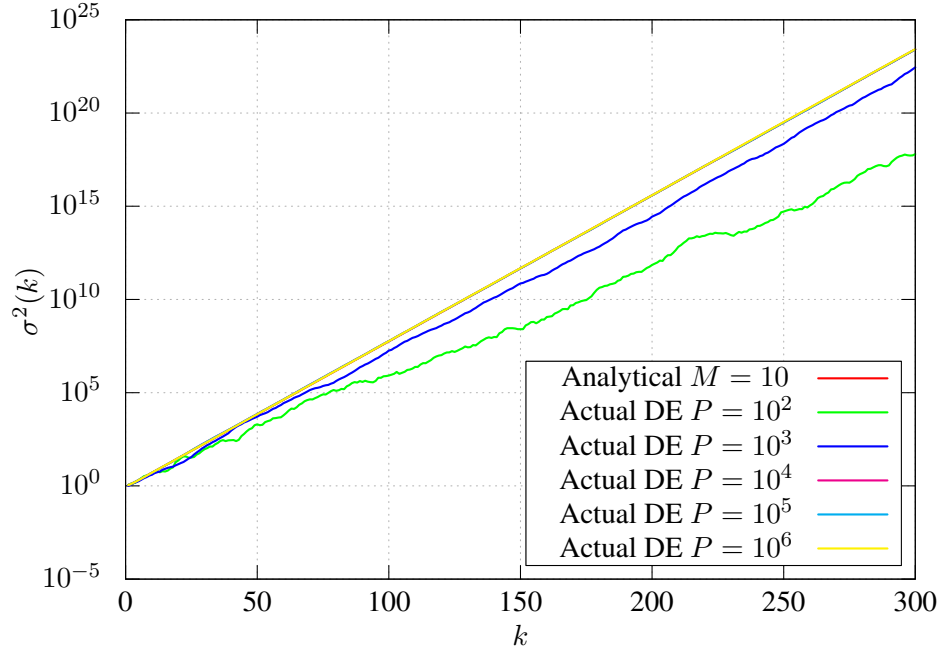
(d) $c_f = 0.5, \lambda = 0.9$



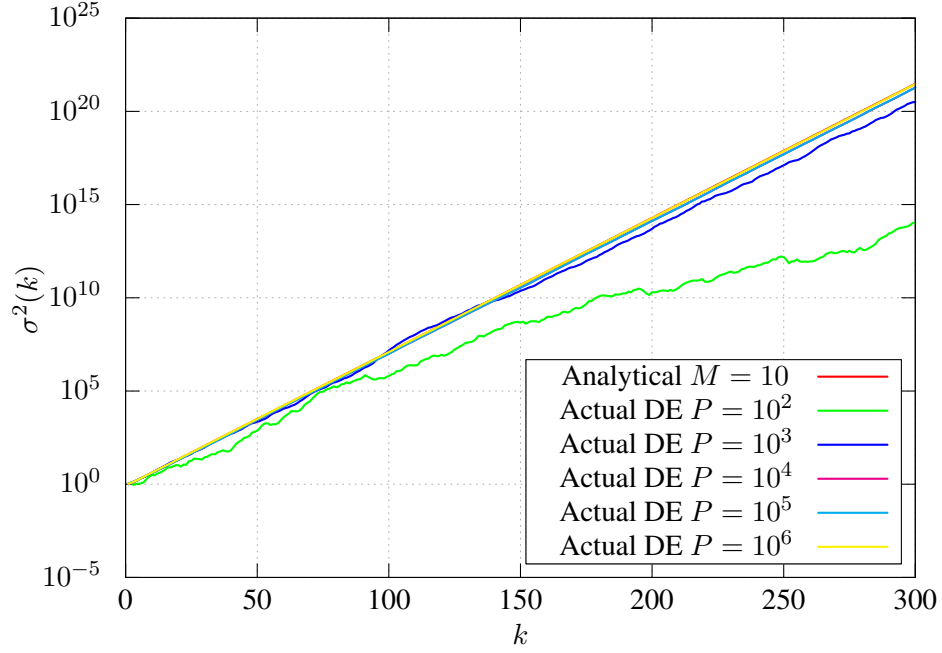
(e) $c_f = 0.7, \lambda = 0.3$



(f) $c_f = 0.7, \lambda = 0.5$



(g) $c_f = 0.7, \lambda = 0.7$



(h) $c_f = 0.7, \lambda = 0.9$

Fig. 4.3 Comparison between actual DE and estimation by proposed method.

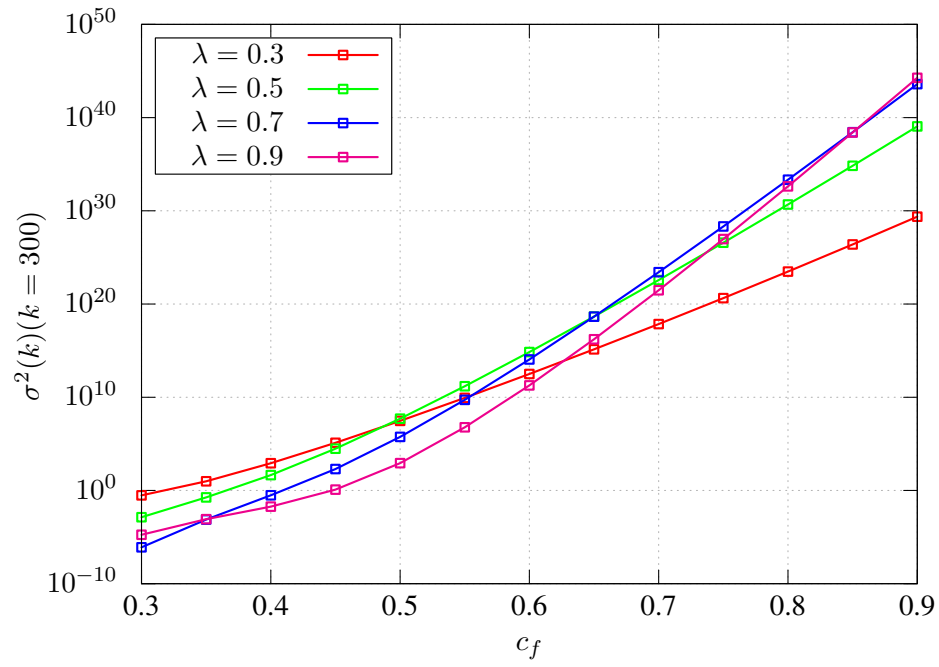


Fig. 4.4 Relation between parameter setting and estimated variance.

第5章 数理的動特性解析可能な多点型確率的 最適化アルゴリズムのフレームワーク

5.1 本章について

第3章および第4章では、既存の有力な多点型確率的最適化アルゴリズムである Particle Swarm Optimization (PSO) と Differential Evolution (DE) に対する収束性解析をおこない、パラメータと各アルゴリズムにおける探索点収束性の関係について考察した。しかしながら、それぞれの章末で言及したとおり、これらの解析は対象とするアルゴリズムに特化したものであり、アルゴリズムのバリエーションまで考慮した解析手法とはなっていなかった。また、とくに第4章でおこなった DE の収束性解析に関しては、解析のために近似や仮定を多く設けており、解析の成果を実際の最適化に適用しづらいという課題も残している。

上述の課題に対して、本章では、アルゴリズムを構築してから個別にその解析をおこなうのではなく、まず数理的に動特性解析可能なアルゴリズムのフレームワークを作り、そのなかでアルゴリズム群を構築していくアプローチを提案する。ここでいう「フレームワーク」とは、アルゴリズムの上位的概念であり、具体的にはアルゴリズムの構造や具備すべき性質などを定めるものである^[119, 120]。

本章では、多くのメタヒューリスティクスで採用される、(1) 複数の探索点を用いた多点型探索、(2) 擬似乱数を用いた確率的探索を基本方策とし、各計算の反復において、各探索点の位置が正規乱数の線形結合および定常入力項により与えられる構造を考え、これを一般化したものをアルゴリズムのフレームワークとして提案する。このフレームワークにおいて、その詳細（正規乱数の中心や線形結合の係数、定常入力項など）は具体的に定められておらず、またフレームワーク自体は最適化能力を有していない。しかし、所与の制約のもとで、探索方策を適切に反映し詳細構造を定めることで、最適化能力をもつアルゴリズムを構成することができる。

また、本章では、上述のフレームワークに対して、探索点位置の分散共分散行列を指標とする探索点分布に関する安定性定理を与える。この定理は提案フレームワークに基づき構築されるアルゴリズムすべてに対して適用可能なものであり、その結果を用いれば、個々のアルゴリズムに対して探索点分布の制御則を与えることが可能となり、第2章でも述べた、大域的探索に必要とされる「集中化と多様化」の実現に大きく寄与する。

本章の構成を以下に述べる。まず、5.2節では、本論文で提案する「正規乱数線形結合型フレームワーク」の構造を示す。5.3節では、提案フレームワークに基づくすべてのアルゴリズムに適用可能な安定性定理を示したうえで、数値実験によりその妥当性を検証する。5.4節では、提案フレームワークに基づく具体的な最適化アルゴリズムとして、PSO に着想を得たアルゴリズムを示し、その最適化性能を数値実験により確認する。最後に5.5節では、本章の内容をまとめるとともに、提案フレームワークの発展に関する展望を述べる。

5.2 正規乱数線形結合型フレームワーク

多点型確率的最適化アルゴリズムのフレームワークとして,

$$\mathbf{x}^{(p)}(k+1) = \sum_{m=1}^M A_m^{(p)}(k) \mathbf{Y}_m^{(p)}(k) + \mathbf{c}^{(p)}(\mathcal{X}(k), k) \quad (5.1a)$$

$$\mathbf{Y}_m^{(p)}(k) \sim \mathcal{N}(\mathbf{b}_m^{(p)}(\mathcal{X}(k), k), \mathbf{\Sigma}(k)) \quad (5.1b)$$

$$\mathbf{\Sigma}(k) = \frac{1}{P-1} \sum_{p=1}^P (\mathbf{x}^{(p)}(k) - \bar{\mathbf{x}}(k))(\mathbf{x}^{(p)}(k) - \bar{\mathbf{x}}(k))^\top \quad (5.1c)$$

で定義される構造を考える. ここで, $k = 0, 1, \dots$ は探索時刻, 上付き添字 $p = 1, \dots, P$ は探索点番号, $\mathbf{x}^{(p)}(k) \in \mathbb{R}^N$ は p 番目の探索点の探索時刻 k における位置をそれぞれ意味する. $A_m^{(p)}(k) \in \mathbb{R}$ は確率変数であり, 期待値および分散が存在する任意の確率分布 $\mathcal{P}_{Am}$ にしたがうものとする. また, $\mathbf{b}_m^{(p)}(\mathcal{X}(k), k) \in \mathbb{R}^N$, $\mathbf{c}^{(p)}(\mathcal{X}(k), k) \in \mathbb{R}^N$ は, いずれも探索履歴 $\mathcal{X}(k)$ や探索時刻 k によって定まる適当な関数であり, 必要に応じて乱数を組み入れてもよい. ただし, ある $\mathbf{c} \in \mathbb{R}^N$ が存在して,

$$\lim_{k \rightarrow \infty} \Pr\{\mathbf{b}_m^{(p)}(\mathcal{X}(k), k) = \mathbf{0}\} = 1 \quad (5.2a)$$

$$\lim_{k \rightarrow \infty} \Pr\{\mathbf{c}^{(p)}(\mathcal{X}(k), k) = \mathbf{c}\} = 1 \quad (5.2b)$$

となることが保証されるものであるとし¹, また実用上有用な定理を導くため, (5.2a) 式および (5.2b) 式を強めた条件として, 任意の $k \geq K$ に対して,

$$\mathbf{b}_m^{(p)}(\mathcal{X}(k), k) = \mathbf{0} \quad (5.3a)$$

$$\mathbf{c}^{(p)}(\mathcal{X}(k), k) = \mathbf{c} \quad (5.3b)$$

となるような $K \geq 0$ が存在すると仮定する.

本フレームワークの中心的な式である (5.1a) 式はおもに正規乱数の線形結合で構成されている. このことから, 本フレームワークを LNR (Linear combination of Normal Random variables) とよぶこととする.

LNR の更新式 (5.1) 式を用いた探索点位置 $\mathbf{x}^{(p)}(k)$ ($p = 1, \dots, P$) の具体的な更新手順はつぎのようになる.

- (1) P 個の探索点の位置 $\mathbf{x}^{(p)}(k)$ ($p = 1, \dots, P$) に対して, (5.1c) 式を用いて分散共分散行列 $\mathbf{\Sigma}(k)$ を計算する.
- (2) (1) で計算した分散共分散行列 $\mathbf{\Sigma}(k)$ と (5.1b) 式を用いて, N 次元正規乱数 $\mathbf{Y}_m^p(k)$ ($p = 1, \dots, P, m = 1, \dots, M$) を生成する.
- (3) (2) で生成した N 次元正規乱数 $\mathbf{Y}_m^p(k)$ ($p = 1, \dots, P, m = 1, \dots, M$) と (5.1a) 式を用いて新たな探索点位置 $\mathbf{x}^{(p)}(k+1)$ ($p = 1, \dots, P$) を計算する.

¹乱数の内在を考慮したうえで, 確率変数列の概収束性をもって定義した.

LNR の最大の特徴は、探索時刻が k から $k+1$ に更新されるとき、探索点位置を直接保存せず、探索点位置の分散にしたがう正規乱数を用いて探索点を再配置する点にある。第4章でも議論したとおり、多点型確率的最適化アルゴリズムにおいて、探索点の分布形状を掴むことは困難である。LNR では、探索点分布の正規性がアルゴリズムの側から保証され、正規分布のもつ「良い」数学的性質の活用が仮定や近似なしに可能であり、このフレームワークに基づく最適化アルゴリズムの数学的特性を詳細に議論することができる。

5.3 正規乱数線形結合型フレームワークの安定性

5.3.1 正規乱数線形結合系フレームワークに対する安定性定理

本節では、LNR に基づくすべてのアルゴリズムに適用可能な探索点分布の安定性定理を導く。本節の証明では、既存の多点型確率的最適化アルゴリズムの安定性解析や第3章と同様、探索が停滞した状態、すなわち (5.3) 式がすでに成立した状況を仮定する。また、本節では簡単に変数が1次元の場合を考える。(5.3) 式が成立した状況では、LNR のダイナミクスは方向依存性をもたない。したがって、1次元の場合における LNR の安定性定理はそのまま変数が多次元の場合に拡張することができる。

まず、LNR の確率漸近安定条件と、探索点位置の分散の「拡大率」を定義する。

定義 5.1 (LNR の確率漸近安定性) (5.1) 式で与えられる LNR に基づくアルゴリズム $\mathcal{A}$ において、探索点位置の分散共分散行列 $\Sigma(k)$ の任意の要素 $\sigma_{mn}(k)$ ($m, n = 1, \dots, N$) が、任意の正数 $\varepsilon > 0$ に対して、

$$\lim_{K \rightarrow \infty} \Pr \left\{ \sup_{k > K} |\sigma_{mn}(k)| > \varepsilon \right\} = 0 \quad (5.4)$$

を満たすとき、アルゴリズム $\mathcal{A}$ は確率漸近安定（あるいは単純に安定）であると定義する。

(5.4) 式は、すべての探索点が同一座標に集中することを意味している。上述したとおり、本節では変数が1次元の場合を考えるため、(5.1c) 式を、とくに

$$\sigma^2(k) = \frac{1}{P-1} \sum_{p=1}^P (x^{(p)}(k) - \bar{x}(k))^2 \quad (5.5)$$

と書き、その確率漸近安定条件

$$\lim_{K \rightarrow \infty} \Pr \left\{ \sup_{k > K} |\sigma^2| > \varepsilon \right\} = 0 \quad (5.6)$$

について議論する。

定義 5.2 (拡大率) (5.3a) 式および (5.3b) 式が成立するとき, $r^{(p)}(k+1)$ を,

$$r^{(p)}(k+1) = \frac{1}{\sqrt{\sigma^2(k)}} \sum_{m=1}^M A_m^{(p)}(k) Y_m^{(p)}(k) + c = \sum_{m=1}^M A_m^{(p)}(k) Y_m'^{(p)}(k) + c \quad (5.7)$$

とおき, $r^{(p)}(k+1)$ の不偏分散

$$s_r^2(k+1) = \frac{1}{P-1} \sum_{p=1}^P (r^{(p)}(k+1) - \bar{r}(k+1))^2 \quad (5.8)$$

を LNR の拡大率と定義する. ただし, (5.7) 式において, $Y_m'^{(p)}(k) = Y_m^{(p)}(k)/\sqrt{\sigma^2(k)}$ である.

(5.7) 式において, 正規乱数 $Y_m'^{(p)}(k)$ の分散は 1 であり, 拡大率 $s_r^2(k+1)$ は, 探索時刻が k から $k+1$ の更新されるときに探索点の位置の分散がどれだけ拡大あるいは縮小するかを記述する量となっている. 探索時刻 $k \gg 1$ のとき, $\sigma^2(k)$ は, 拡大率 $s_r^2(\kappa)$ ($\kappa = 1, \dots, k$) を用いて,

$$\sigma^2(k) \approx \left(\prod_{\kappa=1}^k s_r^2(\kappa) \right) \sigma^2(0) \quad (5.9)$$

と見積ることができる.

つぎに, LNR の安定性定理を与えるため, つぎの 2 つの補題を用意する.

補題 5.1 分散 σ^2 , 尖度 β_2 の任意の確率分布にしたがう乱数 $X^{(p)}$ ($p = 1, \dots, P$) を独立に生成する操作を k 回 ($k \gg 1$) 繰り返すとき,

$$\prod_{\kappa=1}^k s^2(\kappa) \approx \sigma^{2k} \left(1 + \frac{(P-1)\beta_2 - P + 3}{P(P-1)} \right)^{-\frac{k}{2}} \quad (5.10)$$

が成り立つ. ただし, (5.10) 式において, $s^2(\kappa)$ は $X^{(p)}(\kappa)$ の不偏分散, すなわち

$$s^2(\kappa) = \frac{1}{P-1} \sum_{p=1}^P (X^{(p)}(\kappa) - \bar{X}(\kappa))^2 \quad (5.11)$$

である.

証明 $s^2(\kappa)$ ($\kappa = 1, \dots, k$) は正の値をとる互いに独立な確率変数であるから, $k \gg 1$ としたとき, その総積 $\prod_{\kappa=1}^k s^2(\kappa)$ は, 中心極限定理により対数正規分布に収束する.

(A.21a) 式および (A.21b) 式より, 位置母数が $\hat{\mu}$, 尺度母数が $\hat{\sigma}^2$ で与えられる対数正規分布 $\mathcal{LN}(\hat{\mu}, \hat{\sigma}^2)$ の期待値 $\mu_{\mathcal{LN}}$ と分散 $\sigma_{\mathcal{LN}}^2$ は, それぞれ

$$\mu_{\mathcal{LN}} = \exp \left(\hat{\mu} + \frac{1}{2} \hat{\sigma}^2 \right) \quad (5.12a)$$

$$\sigma_{\mathcal{LN}}^2 = \exp(2\hat{\mu} + \hat{\sigma}^2) (\exp(\hat{\sigma}^2) - 1) \quad (5.12b)$$

で与えられる。一方で, (A.1a) 式と (A.1b) 式より, $\prod_{\kappa=1}^k s^2(\kappa)$ の期待値 $E \left[\prod_{\kappa=1}^k s^2(\kappa) \right]$ と分散 $V \left[\prod_{\kappa=1}^k s^2(\kappa) \right]$ は, それぞれ

$$E \left[\prod_{\kappa=1}^k s^2(\kappa) \right] = E^k [s^2] \quad (5.13a)$$

$$V \left[\prod_{\kappa=1}^k s^2(\kappa) \right] = (E^2 [s^2] + V [s^2])^k - E^{2k} [s^2] \quad (5.13b)$$

で与えられる。(5.13a) 式と (5.13b) 式の右辺において, $s^2(\kappa)$ の期待値と分散は試行番号 κ に依存しないことから, これを省略して記述している。さらに, (A.11a) 式と (A.11b) 式より, $E [s^2]$ と $V [s^2]$ は, それぞれ

$$E [s^2] = \sigma^2 \quad (5.14a)$$

$$V [s^2] = \frac{\sigma^4 ((P-1)\beta_2 - P + 3)}{P(P-1)} \quad (5.14b)$$

となる。(5.14a) 式と (5.14b) 式を (5.13a) 式と (5.13b) 式に代入すると,

$$E \left[\prod_{\kappa=1}^k s^2(\kappa) \right] = \sigma^{2k} \quad (5.15a)$$

$$V \left[\prod_{\kappa=1}^k s^2(\kappa) \right] = \sigma^{4k} \left(\left(1 + \frac{(P-1)\beta_2 - P + 3}{P(P-1)} \right)^k - 1 \right) \quad (5.15b)$$

を得る。ここで, (5.12a) 式と (5.15a) 式, (5.12b) 式と (5.15b) 式をそれぞれ等しくおき, 連立方程式を立てて対数正規分布の位置母数 $\hat{\mu}$ と尺度母数 $\hat{\sigma}^2$ について解くと,

$$\hat{\mu} = k \log \left(\sigma^2 \left(1 + \frac{(P-1)\beta_2 - P + 3}{P(P-1)} \right)^{-\frac{1}{2}} \right) \quad (5.16a)$$

$$\hat{\sigma}^2 = k \log \left(1 + \frac{(P-1)\beta_2 - P + 3}{P(P-1)} \right) \quad (5.16b)$$

が得られる。対数正規分布の定義より, (5.16a) 式で求まる位置母数 $\hat{\mu}$ は, $\log \prod_{\kappa=1}^k s^2(\kappa)$ の期待値となる。そこで, $\log \prod_{\kappa=1}^k s^2(\kappa) \approx \hat{\mu}$ と見積ると,

$$\begin{aligned} \prod_{\kappa=1}^k s^2(\kappa) &= \exp \left(\log \prod_{\kappa=1}^k s^2(\kappa) \right) \\ &\approx \exp (\hat{\mu}) \\ &= \sigma^{2k} \left(1 + \frac{(P-1)\beta_2 - P + 3}{P(P-1)} \right)^{-\frac{k}{2}} \end{aligned} \quad (5.17)$$

となり (5.10) 式を得る。 □

補題 5.2 (5.1) 式で与えられる LNR に基づくアルゴリズム $\mathcal{A}$ において, (5.3) 式が成立するとき, (5.7) 式で定めた確率変数 $r^{(p)}(k+1)$ の分散 $\sigma_r^2(k+1)$ と尖度 $\beta_{2r}(k+1)$ は, それぞれ時間に依存しない定数として,

$$\sigma_r^2 = \sum_{m=1}^M \mathbb{E}[A_m^2] \quad (5.18a)$$

$$\beta_{2r} = \frac{1}{\sigma_r^4} \left(3 \sum_{m=1}^M \mathbb{E}[A_m^4] + \sum_{m \neq l} \mathbb{E}[A_m^2] \mathbb{E}[A_l^2] \right) \quad (5.18b)$$

で与えられる.

証明 定義に基づき $\sigma_r^2(k+1)$ を展開すると,

$$\begin{aligned} \sigma_r^2(k+1) &= \mathbf{V} \left[\sum_{m=1}^M A_m^{(p)}(k) Y_m'^{(p)}(k) + c \right] \\ &= \sum_{m=1}^M \mathbf{V}[A_m^{(p)}(k)] \mathbf{V}[Y_m'^{(p)}(k)] + \sum_{m=1}^M \mathbb{E}^2[A_m^{(p)}(k)] \mathbf{V}[Y_m'^{(p)}(k)] \\ &\quad + \sum_{m=1}^M \mathbf{V}[A_m^{(p)}(k)] \mathbb{E}^2[Y_m'^{(p)}(k)] + \mathbf{V}[c] \end{aligned} \quad (5.19)$$

となる. ここで, $\mathbb{E}[Y_m'^{(p)}(k)] = 0$, $\mathbf{V}[Y_m'^{(p)}(k)] = 1$, $\mathbf{V}[c] = 0$ を用い, また $\mathbf{V}[A_m^{(p)}(k)]$, $\mathbb{E}[A_m^{(p)}(k)]$ は探索時刻 k にも探索点番号 p にも依存しないことから, それぞれ $\mathbf{V}[A_m]$, $\mathbb{E}[A_m]$ と略記すると, $\sigma_r^2(k+1)$ は時間に依存しない定数 σ_r^2 として,

$$\sigma_r^2 = \sum_{m=1}^M (\mathbf{V}[A_m] + \mathbb{E}^2[A_m]) = \sum_{m=1}^M \mathbb{E}[A_m^2] \quad (5.20)$$

と求まる.

一方で, $\beta_{2r}(k+1)$ は尖度の定義より,

$$\begin{aligned} \beta_{2r}(k+1) &= \frac{1}{\sigma_r^4(k)} \mathbb{E} \left[\left(\sum_{m=1}^M A_m^{(p)}(k) Y_m'^{(p)}(k) + c - \mathbb{E} \left[\sum_{m=1}^M A_m^{(p)}(k) Y_m'^{(p)}(k) + c \right] \right)^4 \right] \end{aligned} \quad (5.21)$$

で与えられる. $\sigma_r^2(k+1)$ の場合と同様に, $\mathbb{E}[Y_m'^{(p)}(k)] = 0$, $\mathbf{V}[Y_m'^{(p)}(k)] = 1$, $\mathbf{V}[c] = 0$ を用

いると,

$$\begin{aligned}
 & \mathbb{E} \left[\left(\sum_{m=1}^M A_m^{(p)}(k) Y_m'^{(p)}(k) + c - \mathbb{E} \left[\sum_{m=1}^M A_m^{(p)}(k) Y_m'^{(p)}(k) + c \right] \right)^4 \right] \\
 &= \mathbb{E} \left[\left(\sum_{m=1}^M A_m^{(p)}(k) Y_m'^{(p)}(k) \right)^4 \right] \\
 &= \mathbb{E} \left[\sum_{m=1}^M (A_m^{(p)}(k))^4 (Y_m'^{(p)}(k))^4 \right] + \mathbb{E} \left[\sum_{m \neq l} (A_m^{(p)}(k))^2 (A_l^{(p)}(k))^2 (Y_m'^{(p)}(k))^2 (Y_l'^{(p)}(k))^2 \right] \\
 &= \sum_{m=1}^M \mathbb{E} \left[(A_m^{(p)}(k))^4 \right] \mathbb{E} \left[(Y_m'^{(p)}(k))^4 \right] \\
 &\quad + \sum_{m \neq l} \mathbb{E} \left[(A_m^{(p)}(k))^2 \right] \mathbb{E} \left[(A_l^{(p)}(k))^2 \right] \mathbb{E} \left[(Y_m'^{(p)}(k))^2 \right] \mathbb{E} \left[(Y_l'^{(p)}(k))^2 \right] \\
 &= 3 \sum_{m=1}^M \mathbb{E} \left[(A_m^{(p)}(k))^4 \right] + \sum_{m \neq l} \mathbb{E} \left[(A_m^{(p)}(k))^2 \right] \mathbb{E} \left[(A_l^{(p)}(k))^2 \right]
 \end{aligned} \tag{5.22}$$

となる. なお, (5.22) 式の6行目では, $Y_m'^{(p)}(k)$ が分散1の正規分布にしたがい $\mathbb{E} \left[(Y_m'^{(p)}(k))^4 \right] = 3$ となることを用いている. (5.22) 式を (5.21) 式に代入し, また $\mathbb{E} \left[(A_m^{(p)}(k))^4 \right], \mathbb{E} \left[(A_l^{(p)}(k))^4 \right]$ は探索時刻 k にも探索点番号 p にも依存しないことから, それぞれ $\mathbb{E} [A_m^4], \mathbb{E} [A_l^4]$ と略記すると, $\beta_{2r}(k+1)$ は, 時間に依存しない定数 β_{2r} として,

$$\beta_{2r} = \frac{1}{\sigma_r^4} \left(3 \sum_{m=1}^M \mathbb{E} [A_m^4] + \sum_{m \neq l} \mathbb{E} [A_m^2] \mathbb{E} [A_l^2] \right) \tag{5.23}$$

と求まる. □

補題 5.1 および 5.2 を用いて, LNR の安定性に関する定理を導く.

定理 5.1 (5.1) 式で与えられる LNR に基づくアルゴリズム $\mathcal{A}$ において,

$$\sigma_r^2 = \sum_{m=1}^M \mathbb{E} [A_m^2] \tag{5.24a}$$

$$\beta_{2r} = \frac{1}{\sigma_r^4} \left(3 \sum_{m=1}^M \mathbb{E} [A_m^4] + \sum_{m \neq l} \mathbb{E} [A_m^2] \mathbb{E} [A_l^2] \right) \tag{5.24b}$$

とし,

$$\sigma_r^2 \left(1 + \frac{(P-1)\beta_{2r} - P + 3}{P(P-1)} \right)^{-\frac{1}{2}} < 1 \tag{5.25}$$

が成り立つならば, $\mathcal{A}$ は確率漸近安定である.

証明 探索時刻 k における探索点位置の不偏分散は, (5.9) 式で見積ることができる. (5.9)

式に対し、補題 5.1 を適用すると、

$$\sigma^2(k) \approx \sigma_r^{2k} \left(1 + \frac{(P-1)\beta_{2r} - P + 3}{P(P-1)} \right)^{-\frac{k}{2}} \sigma^2(0) \quad (5.26)$$

を得る。(5.26) 式において、 σ_r^2, β_{2r} は直接書き下していないが、それぞれ補題 5.2 を用いて、(5.18a) 式および (5.18b) 式により求めることができる。ここで、

$$\zeta = \sigma_r^2 \left(1 + \frac{(P-1)\beta_{2r} - P + 3}{P(P-1)} \right)^{-\frac{1}{2}} \quad (5.27)$$

とすると、(5.26) 式は、

$$\sigma^2(k) \approx \zeta^k \sigma^2(0) \quad (5.28)$$

と書くことができる。 $\zeta < 1$ であるとき、 $\sigma^2(k)$ の見積りは k に対して単調減少となる。したがって、任意の $\varepsilon > 0$ に対して K を十分大きくとれば、任意の $k > K$ に対して $\sigma^2(k) \|\mathbf{p}(0)\| > \varepsilon$ となる確率を十分に小さくでき、アルゴリズム $\mathcal{A}$ は確率漸近安定条件 (5.4) を満足する。□

なお、(5.28) 式より、 ζ は第 3 章で PSO に対して導入した安定性指標と同様、LNR の探索点分布に関する安定性の指標として用いることができる。 ζ は、探索点数 P や LNR に基づき実装された具体的なアルゴリズムに内在するパラメータに依存するため、以降では、たとえばアルゴリズムのパラメータを c とするとき、 $\zeta(c, P)$ を書くこととする。

5.3.2 数値実験による安定性定理の確認

数値実験を通して、前項で構築した LNR の安定性定理の妥当性を、多次元の場合も含めて確認する。本項では、 $M = 2, \mathbf{c}^{(p)}(\mathcal{X}(k), k) = \mathbf{b}_m^{(p)}(\mathcal{X}(k), k) = \mathbf{0}$ とし、

$$\mathbf{x}^{(p)}(k+1) = \sum_{m=1}^2 A_m^{(p)}(k) \mathbf{Y}_m^{(p)}(k) \quad (5.29a)$$

$$\mathbf{Y}_m^{(p)}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}(k)) \quad (5.29b)$$

$$\mathbf{\Sigma}(k) = \frac{1}{P-1} \sum_{p=1}^P (\mathbf{x}^{(p)}(k) - \bar{\mathbf{x}}(k))(\mathbf{x}^{(p)}(k) - \bar{\mathbf{x}}(k))^\top \quad (5.29c)$$

で定義される、LNR のきわめて単純な実装 $\mathcal{A}_{\text{test}}$ を利用する。ただし、(5.29a) 式において、 $A_1^{(p)}(k), A_2^{(p)}(k)$ はいずれも実数値一様分布 $\mathcal{U}_{\mathbf{R}}(0, c)$ にしたがう乱数で決定されるものとする。 $\mathcal{A}_{\text{test}}$ は、実装を単純なものとすることで LNR の妥当性確認をおこなうことを目的としており、最適化方策は組み込んでいないことに注意されたい。定理 5.1 を使うと、 $\sigma_r^2(c, P)$ 、 $\beta_{2r}(c, P)$ および安定性指標 $\zeta(c, P)$ は、それぞれ

$$\sigma_r^2(c, P) = \frac{2c^2}{3} \quad (5.30a)$$

$$\beta_{2r}(c, P) = \frac{16}{5} \quad (5.30b)$$

Table 5.1 Numerical simulation conditions.

Item	Value
Dimension (N)	20
Max Iteration (K)	500
Number of Search Points (P)	10, 20, 100
Parameter (c)	1.0, 1.5, 2.0
Initial Condition	$\mathbf{x}_n^{(p)}(0) \sim \mathcal{N}(0.0, 1.0)$

 Table 5.2 Calculated stability indices of $\mathcal{A}_{\text{test}}$.

c	$\zeta (P = 10)$	$\zeta (P = 20)$	$\zeta (P = 100)$
1.0	0.59815	0.63128	0.65939
1.5	1.34583	1.42037	1.48361
2.0	2.39259	2.52511	2.63755

$$\zeta(c, P) = \frac{2c^2}{3} \left(1 + \frac{(P-1)16/5 - P + 3}{P(P-1)} \right)^{-\frac{1}{2}} \quad (5.30c)$$

と解析的に導くことができる。

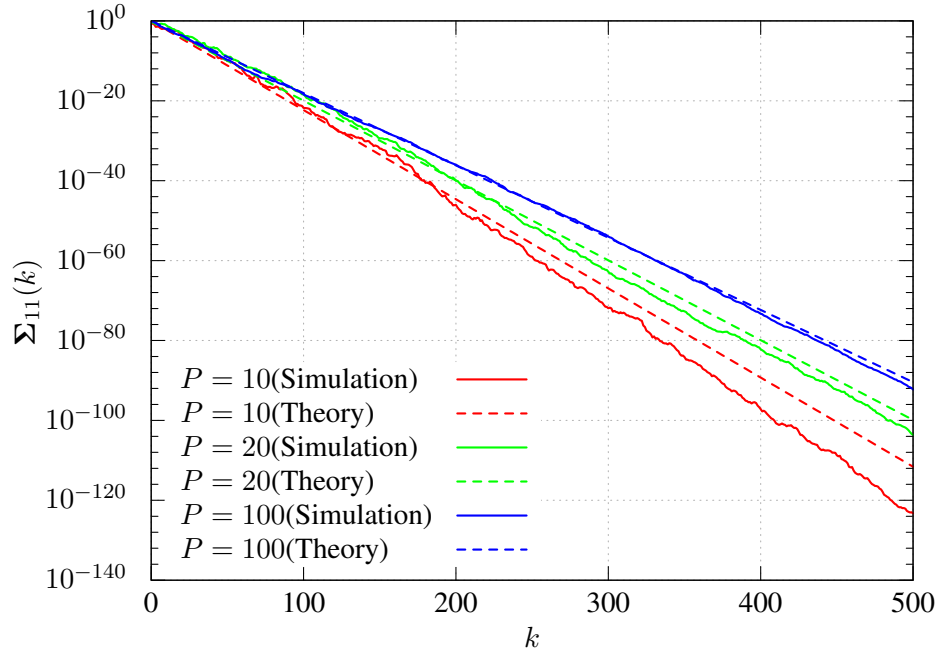
本数値実験では、 $\mathcal{A}_{\text{test}}$ における実際の探索点位置の分散の時間発展と、(5.28) 式と (5.30c) 式を用いて理論的に計算された探索点位置の分散の時間発展の比較をおこなう。Table 5.1 に数値実験条件を示す。

Fig. 5.1 に、 $\mathcal{A}_{\text{test}}$ の実行により得た実際の探索点位置の分散の時間発展（実線）と、(5.28) 式と (5.30c) 式を用いて理論的に計算された探索点位置の分散の時間発展（破線）を合わせて示す。また、Table 5.2 に、(5.30c) 式を用いて計算された各パラメータ設定に対する安定性指標 $\zeta(c, P)$ の値を示す。図より、数値実験の結果、実際の探索点位置の分散は理論的な解析値にオーダーの範囲内でおおむね一致していることがわかる。なお、探索点数 P の値が小さいと、探索点位置の分散の時間発展中の「揺らぎ」が目立つ。これは、(A.19b) 式が示す通り、 P が小さくなると拡大率の分散が大きくなるためであると考えられる。探索点分布の精緻な解析という視点からは、 P は大きい方が望ましいと考えられるが、一方で、最適化、とくに「集中化と多様化」という観点からは、ここでいう「揺らぎ」は局所的探索と大域的探索の同時実現に寄与すると考えられるため、 P を大きくすることは必ずしも有効な方策ではないと考えられる。

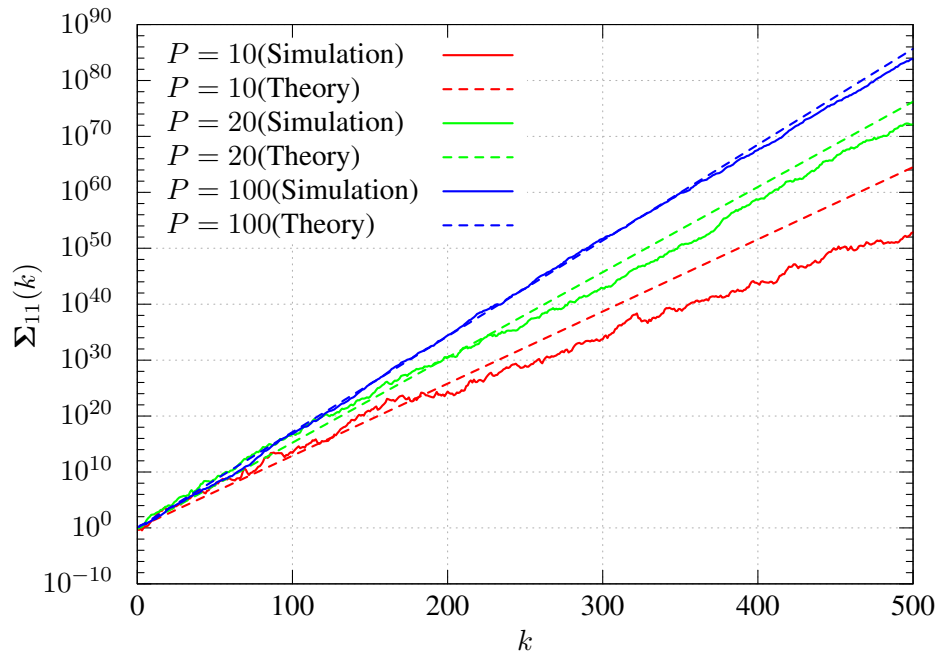
5.4 正規乱数線形結合型フレームワークに基づく最適化アルゴリズムの例

5.4.1 PSO に着想を得た最適化アルゴリズム

LNR に基づく具体的な最適化アルゴリズムとして、Particle Swarm Optimization (PSO) に着想を得たアルゴリズムを構築する。PSO の探索点座標更新式は、第 2 章で示したと



(a) $c = 1.0$



(b) $c = 1.5$

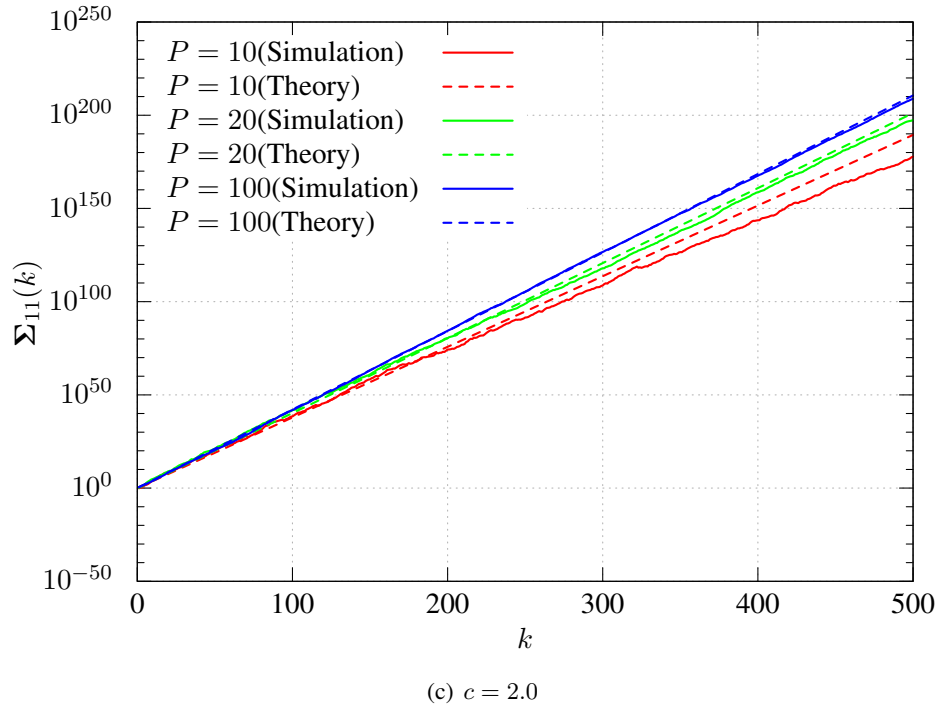


Fig. 5.1 Time history of the variance of the search point positions.

おり,

$$\begin{aligned} \mathbf{v}^{(p)}(k+1) = & \lambda \mathbf{v}^{(p)}(k) + c_1 \mathbf{R}_1^{(p)}(k) (\mathbf{x}^{(p\text{-best})}(k) - \mathbf{x}^{(p)}(k)) \\ & + c_2 \mathbf{R}_2^{(p)}(k) (\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k)) \end{aligned} \quad (5.31a)$$

$$\mathbf{x}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k+1) \quad (5.31b)$$

で与えられる.

PSO に着想を得た LNR に基づく最適化アルゴリズムとして, つぎのようなアルゴリズム $\mathcal{A}_{\text{PSO}}$ を考える.

$$\mathbf{x}^{(p)}(k+1) = A^{(p)}(k) \mathbf{Y}^{(p)}(k) + \mathbf{c}^{(p)}(\mathcal{X}(k), k) \quad (5.32a)$$

$$\mathbf{Y}^{(p)}(k) \sim \mathcal{N}(\mathbf{b}^{(p)}(\mathcal{X}(k), k), \mathbf{\Sigma}(k)) \quad (5.32b)$$

$$\mathbf{\Sigma}(k) = \frac{1}{P-1} \sum_{p=1}^P (\mathbf{x}^{(p)}(k) - \bar{\mathbf{x}}(k)) (\mathbf{x}^{(p)}(k) - \bar{\mathbf{x}}(k))^{\top} \quad (5.32c)$$

ここで, LNR の任意部分 $A^{(p)}(k)$, $\mathbf{b}^{(p)}(\mathcal{X}(k), k)$, $\mathbf{c}^{(p)}(\mathcal{X}(k), k)$ はそれぞれ

$$A^{(p)}(k) = 1 - c_1 R_1^{(p)}(k) - c_2 R_2^{(p)}(k) \quad (5.33a)$$

$$\mathbf{b}^{(p)}(\mathcal{X}(k), k) = \mathbf{0} \quad (5.33b)$$

$$\mathbf{c}^{(p)}(\mathcal{X}(k), k) = \frac{1}{2} (\mathbf{x}^{(p\text{-best})}(k) + \mathbf{x}^{(g\text{-best})}(k))$$

$$\begin{aligned}
 & + \frac{c_1 R_1^{(p)}(k)}{2} (\mathbf{x}^{(p\text{-best})}(k) - \mathbf{x}^{(g\text{-best})}(k)) \\
 & + \frac{c_2 R_2^{(p)}(k)}{2} (\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p\text{-best})}(k))
 \end{aligned} \tag{5.33c}$$

で定められるものとする。ただし、(5.33)式において、 $c_1, c_2 \geq 0$ はパラメータであり、また $R_1^{(p)}(k), R_2^{(p)}(k)$ は実数値一様分布 $\mathcal{U}(0, 1)$ にしたがる乱数により決定される。

(5.33a) 式、(5.33b) 式、および (5.33c) 式を (5.32a) 式に代入して整理すると、PSO の更新式と似た更新式

$$\begin{aligned}
 \mathbf{x}^{(p)}(k+1) = & \mathbf{z}^{(p)}(k) + c_1 R_1^{(p)}(k) (\mathbf{x}^{(p\text{-best})}(k) - \mathbf{z}^{(p)}(k)) \\
 & + c_2 R_2^{(p)}(k) (\mathbf{x}^{(g\text{-best})}(k) - \mathbf{z}^{(p)}(k))
 \end{aligned} \tag{5.34}$$

が得られる。ただし、(5.34) 式において、

$$\mathbf{z}^{(p)}(k) = \mathbf{Y}^{(p)}(k) + \frac{1}{2} (\mathbf{x}^{(p\text{-best})}(k) + \mathbf{x}^{(g\text{-best})}(k)) \tag{5.35}$$

である。

$\mathcal{A}_{\text{PSO}}$ における (5.3a) 式および (5.3b) 式の成立性を議論する。(5.3a) 式については、(5.33b) 式で定めた $\mathbf{b}^{(p)}(\mathcal{X}(k), k)$ の定義より明らかに満足する。一方、(5.3b) 式については、 $\mathcal{A}_{\text{PSO}}$ による最適化過程に対する考察が必要である。 $\mathcal{A}_{\text{PSO}}$ による最適化において、十分に探索が進んだ状態では、すべての $p\text{-best}$ は $g\text{-best}$ に収束する。このとき、(5.33c) 式において、乱数を含む項はいずれもゼロとなるため、 $\mathbf{c}^{(p)}(\mathcal{X}(k), k) = \mathbf{x}^{(g\text{-best})}(k)$ となる。 $\mathbf{x}^{(g\text{-best})}(k)$ は探索が停滞するとほぼ定数とみなすことができるため、結果として $\mathcal{A}_{\text{PSO}}$ は (5.3b) 式も満足する。

$\mathcal{A}_{\text{PSO}}$ に対して、簡単に $c_1 = c_2 = c$ として定理 5.1. を用いると、 $\sigma_r^2(c, P)$, $\beta_{2r}(c, P)$, および安定性指標 $\zeta(c, P)$ は、それぞれ

$$\sigma_r^2(c, P) = 1 - 2c + \frac{7c^2}{6} \tag{5.36a}$$

$$\beta_{2r}(c, P) = \frac{-1 + 2(c-1)^6 + (2c-1)^6}{10c^2(1-2c+7c^2/6)^2} \tag{5.36b}$$

$$\zeta(c, P) = \left(1 - 2c + \frac{7c^2}{6}\right) \left(1 + \frac{-1 + 2(c-1)^6 + (2c-1)^6}{10Pc^2(1-2c+7c^2/6)^2} + \frac{-P+3}{P(P-1)}\right)^{-\frac{1}{2}} \tag{5.36c}$$

と導くことができる。安定・不安定の境界は、 $P = 20$ の場合、 $c = \hat{c} = 1.71840$ であり、探索点数 P が 20 のとき、 $c < \hat{c}$ であれば $\mathcal{A}_{\text{PSO}}$ は安定であり、逆に $c > \hat{c}$ であれば $\mathcal{A}_{\text{PSO}}$ は不安定となる。

5.4.2 数値実験条件

5.4 節で構築した LNR に基づくサンプル最適化アルゴリズムを最適化ベンチマーク問題へ適用し、(1) 本フレームワークのもとで有効な最適化アルゴリズムが構築できること、(2) 構築した最適化アルゴリズムが優れた最適化性能を発揮する場合のパラメータが数学

Table 5.3 Common numerical simulation conditions

Item	Value
Max Iteration (K)	5000
Trial	200
Number of Search Points (P)	20
Parameter (c_1, c_2)	Shown in Each Numerical Simulation
Initial Condition	$x_n^{(p)}(0) \sim \mathcal{U}(-5.0, 5.0)$
Bounding Method	Torus

的に根拠のあるものであること、の2点を確認する。Table 5.3 に、本節における数値実験の共通条件を示す。Table 5.3 にて“Shown in Each Numerical Simulation”となっている条件は、以降、各数値実験の項において個別に示す。なお、本節で用いるベンチマーク問題は Table 3.4 に示したものをを用いる。

5.4.3 最適化性能の評価

パラメータ c の最適化性能への感度をみるため、(1) $c_1 = c_2 = c$ を広域 (1.0~2.0) で変化させた場合、(2) $c_1 = c_2 = c$ を安定・不安定の境界 ($\hat{c} = 1.71480$) の近傍 (1.75~1.85) で変化させた場合、の両方について、 $\mathcal{A}_{\text{PSO}}$ の最適化性能を調査した。

Tables 5.4(a)–5.9(a) および Figs. 5.2(a)–5.7(a) に、Table 3.4 に示す各ベンチマーク問題に対して、 $\mathcal{A}_{\text{PSO}}$ を、パラメータ $c_1 = c_2 = c$ を広域で分布させた場合の最適化性能を示す。また、Tables 5.4(b)–5.9(b) および Figs. 5.2(b)–5.7(b) に、Table 3.4 に示す各ベンチマーク問題に対して、 $\mathcal{A}_{\text{PSO}}$ を、パラメータ $c_1 = c_2 = c$ を安定・不安定の境界の近傍で分布させた場合の最適化性能を示す。

LNR に基づく最適化アルゴリズムの基本的な最適化性能を確認する。Table 5.10 に、Table 3.4 に示す各ベンチマーク問題に対して、 $\mathcal{A}_{\text{PSO}}$ と、第3章で提案した持続的探索型 PSO ($c_1 = c_2 = 2.00000, \lambda = 0.83210$) を適用した場合の最適化性能をまとめる。ここで、 $\mathcal{A}_{\text{PSO}}$ のパラメータ c_1, c_2 は、どの問題に対しても安定的に良い探索性能を示したパラメータである $c_1 = c_2 = c = 1.80000$ とした。Table 5.10 より、 $\mathcal{A}_{\text{PSO}}$ は、LNR に基づくアルゴリズムの例として、PSO に着想を得て単純に構築された最適化アルゴリズムでありながらも、持続的探索型 PSO とほぼ同等の性能を示しており、多峰性でかつ変数間依存を有する問題 No. 6 では、平均値、最小値ともに持続的探索型 PSO より優れた性能を示している。LNR はフレームワークそれ自体に最適化方策を組み込んでいないが、適当な最適化方策を与えることで、良好な最適化性能をもつアルゴリズムを構築できることが示された。

5.4.4 パラメータと最適化性能の関係

まず、パラメータと最適化性能の大域的な関係について確認する。Tables 5.4(a)–5.9(a) および Figs. 5.2(a)–5.7(a) より、パラメータ $c_1 = c_2 = c$ を広域で変化させたとき、最適化性能は顕著に変化することがわかる。いずれのベンチマーク問題においても、共通して $c = 1.0$ から $c = 1.8$ にかけて性能は徐々に向上しており、一方で $c = 1.8$ から $c = 2.0$ の

あいだでは、性能は大差ないか、あるいは Fig. 5.8(a) のように再び性能が悪化するという傾向がみられる。ここで着目したいのは、Prob. 1 (Sphere 関数) に対する最適化性能である。Prob. 1 はきわめて単純な単峰性凸関数であり唯一の局所的最適解が大域的最適解となるが、Table 5.4(a) および Fig. 5.2(a) より、 $c \leq 1.7$ の領域では、上述の大域的最適解が得られていないことがわかる。これは、Prob. 3 や Prob. 4 の多峰性関数の場合のように大域的最適解でない局所的最適解に捕捉されたのではなく、アルゴリズム A_{PSO} が過度に安定的（安定／不安定の境界は $c_1 = c_2 = \hat{c} = 1.78140$ ）であり、探索点が探索の早い段階で最適解でない1点の座標に初期収束してしまったためと考えられる。一方で、 $c \geq 1.8 \geq \hat{c}$ の領域では、アルゴリズム A_{PSO} は不安定となるため、探索点は特定の座標に収束することなく、また、上下限制約手段の存在のもとで探索領域内の探索を継続するため、最適解に近い値が得られている。ただし、アルゴリズムが不安定となることで、「集中化と多様化」のうち「集中化」の機能が損なわれるため、たとえば探索領域内部にきわめて多数の局所的最適解をもち、「集中化」による探索が必須となる Prob. 4 (Rastrigin's 関数) では、Fig. 5.8(a) のように、安定／不安定の境界（ $c_1 = c_2 = \hat{c} = 1.78140$ ）を大きく超えてパラメータを設定すると、再び最適化性能が落ちていくと考えられる。

つぎに、パラメータと最適化性能の大域的な関係について考察し、LNR に基づくアルゴリズムが数理的に動特性解析可能であることを確認する。Tables 5.4(b)–5.9(b) および Figs. 5.2(b)–5.7(b) より、どのベンチマーク問題に対しても、パラメータ $c_1 = c_2$ を、安定／不安定の境界 $\hat{c} = 1.78140$ をやや上回る程度の値に設定することで初期収束を回避、あるいは良好な最適化性能が得られていることがわかる。前者については単峰性関数である Prob. 1 が、後者について多数の局所的最適解をもつ多峰性関数である Prob. 4 においてとくに顕著である。理論上の安定／不安定の境界を与えるパラメータは、第3章で提案した持続的探索をあたえる PSO のパラメータと同質のものであり、このようなパラメータを設定することで、持続的探索型 PSO と同様、「集中化と多様化」を同時に実現することが可能となる。数値実験の結果、理論上の安定／不安定の境界を与えるパラメータよりやや不安定よりのパラメータを設定したほうがよい結果が得られているが、これは上下限制約内に探索点を閉じ込めるトーラス変換が、探索点位置の分散の発散を制限する機能として働くため、理論上の安定／不安定の境界を与えるパラメータはやや安定よりの傾向をもたらすためと考えられる。

以上の結果より、LNR の理論的安定性解析から得られる安定／不安定の境界を与えるパラメータ設定は、アルゴリズムを最適化問題に適用する際のよい指針となると考えられる。

Table 5.4 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 1 ($N = 20, \alpha = 0.0$).

 (a) Wide-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.00000	1.00000	1.365e+01	4.569e+00	4.440e+00	3.060e+01
2	1.10000	1.10000	1.207e+01	4.649e+00	3.177e+00	3.284e+01
3	1.20000	1.20000	1.083e+01	4.243e+00	2.800e+00	2.763e+01
4	1.30000	1.30000	9.410e+00	3.641e+00	2.576e+00	2.244e+01
5	1.40000	1.40000	7.742e+00	2.900e+00	2.890e+00	1.654e+01
6	1.50000	1.50000	6.607e+00	2.650e+00	1.627e+00	1.646e+01
7	1.60000	1.60000	5.010e+00	2.209e+00	9.263e-01	1.291e+01
8	1.70000	1.70000	3.032e+00	1.402e+00	7.375e-01	7.621e+00
9	1.80000	1.80000	2.634e-04	4.681e-04	2.229e-05	5.477e-03
10	1.90000	1.90000	4.667e-04	1.817e-04	1.159e-04	1.046e-03
11	2.00000	2.00000	8.952e-04	3.010e-04	3.544e-04	2.014e-03

 (b) Narrow-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.75000	1.75000	1.509e+00	1.083e+00	1.373e-01	6.132e+00
2	1.76000	1.76000	1.032e+00	9.029e-01	2.412e-02	4.455e+00
3	1.77000	1.77000	6.010e-01	6.187e-01	5.648e-03	3.549e+00
4	1.78000	1.78000	1.475e-01	3.284e-01	6.411e-04	2.597e+00
5	1.78140	1.78140	9.620e-02	2.106e-01	5.562e-04	1.924e+00
6	1.79000	1.79000	8.761e-03	5.281e-02	4.148e-05	7.364e-01
7	1.80000	1.80000	2.634e-04	4.681e-04	2.229e-05	5.477e-03
8	1.81000	1.81000	1.158e-04	6.573e-05	2.065e-05	5.471e-04
9	1.82000	1.82000	1.294e-04	5.463e-05	3.595e-05	3.246e-04
10	1.83000	1.83000	1.605e-04	6.796e-05	3.946e-05	4.757e-04
11	1.84000	1.84000	2.025e-04	8.002e-05	6.480e-05	4.687e-04
12	1.85000	1.85000	2.531e-04	9.648e-05	7.063e-05	6.563e-04

Table 5.5 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 2 ($N = 20, \alpha = 0.0$).

 (a) Wide-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.00000	1.00000	3.675e+03	2.151e+03	5.964e+02	1.376e+04
2	1.10000	1.10000	2.859e+03	1.575e+03	5.422e+02	8.589e+03
3	1.20000	1.20000	2.609e+03	1.707e+03	4.029e+02	1.128e+04
4	1.30000	1.30000	1.927e+03	1.277e+03	3.547e+02	7.963e+03
5	1.40000	1.40000	1.434e+03	9.705e+02	1.782e+02	8.590e+03
6	1.50000	1.50000	1.083e+03	6.432e+02	1.784e+02	3.605e+03
7	1.60000	1.60000	7.379e+02	4.946e+02	1.709e+02	3.591e+03
8	1.70000	1.70000	4.298e+02	3.326e+02	9.542e+01	2.572e+03
9	1.80000	1.80000	1.922e+01	8.797e+00	1.055e+01	7.352e+01
10	1.90000	1.90000	1.923e+01	6.784e+00	1.430e+01	7.518e+01
11	2.00000	2.00000	2.084e+01	1.161e+01	1.363e+01	9.301e+01

 (b) Narrow-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.75000	1.75000	2.030e+02	1.476e+02	2.786e+01	1.183e+03
2	1.76000	1.76000	1.526e+02	1.135e+02	1.992e+01	6.775e+02
3	1.77000	1.77000	1.044e+02	9.108e+01	1.988e+01	7.787e+02
4	1.78000	1.78000	4.844e+01	3.909e+01	1.443e+01	2.730e+02
5	1.78140	1.78140	4.396e+01	4.599e+01	1.529e+01	4.124e+02
6	1.79000	1.79000	2.255e+01	1.572e+01	1.337e+01	1.513e+02
7	1.80000	1.80000	1.922e+01	8.797e+00	1.055e+01	7.352e+01
8	1.81000	1.81000	1.912e+01	8.642e+00	9.845e+00	7.728e+01
9	1.82000	1.82000	1.990e+01	1.080e+01	8.019e+00	7.633e+01
10	1.83000	1.83000	1.875e+01	5.446e+00	1.165e+01	7.145e+01
11	1.84000	1.84000	1.827e+01	4.178e+00	8.125e+00	7.306e+01
12	1.85000	1.85000	1.962e+01	9.274e+00	1.115e+01	7.861e+01

Table 5.6 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 3 ($N = 20, \alpha = 0.0$).

 (a) Wide-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.00000	1.00000	-9.897e+02	6.396e+01	-1.210e+03	-8.251e+02
2	1.10000	1.10000	-1.003e+03	6.246e+01	-1.183e+03	-8.181e+02
3	1.20000	1.20000	-1.020e+03	6.639e+01	-1.187e+03	-8.527e+02
4	1.30000	1.30000	-1.044e+03	6.725e+01	-1.252e+03	-9.024e+02
5	1.40000	1.40000	-1.077e+03	6.567e+01	-1.290e+03	-9.185e+02
6	1.50000	1.50000	-1.106e+03	7.233e+01	-1.324e+03	-9.247e+02
7	1.60000	1.60000	-1.131e+03	7.160e+01	-1.317e+03	-9.396e+02
8	1.70000	1.70000	-1.175e+03	7.183e+01	-1.329e+03	-9.764e+02
9	1.80000	1.80000	-1.322e+03	6.892e+01	-1.482e+03	-1.177e+03
10	1.90000	1.90000	-1.336e+03	6.524e+01	-1.510e+03	-1.198e+03
11	2.00000	2.00000	-1.343e+03	6.891e+01	-1.510e+03	-1.142e+03

 (b) Narrow-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.75000	1.75000	-1.236e+03	7.450e+01	-1.455e+03	-1.025e+03
2	1.76000	1.76000	-1.274e+03	6.946e+01	-1.453e+03	-1.076e+03
3	1.77000	1.77000	-1.302e+03	7.068e+01	-1.453e+03	-1.110e+03
4	1.78000	1.78000	-1.317e+03	6.585e+01	-1.482e+03	-1.147e+03
5	1.78140	1.78140	-1.307e+03	6.645e+01	-1.453e+03	-1.109e+03
6	1.79000	1.79000	-1.323e+03	6.547e+01	-1.482e+03	-1.173e+03
7	1.80000	1.80000	-1.322e+03	6.892e+01	-1.482e+03	-1.177e+03
8	1.81000	1.81000	-1.333e+03	7.333e+01	-1.510e+03	-1.152e+03
9	1.82000	1.82000	-1.337e+03	6.809e+01	-1.510e+03	-1.182e+03
10	1.83000	1.83000	-1.337e+03	6.566e+01	-1.538e+03	-1.179e+03
11	1.84000	1.84000	-1.339e+03	6.791e+01	-1.510e+03	-1.171e+03
12	1.85000	1.85000	-1.334e+03	6.515e+01	-1.510e+03	-1.199e+03

Table 5.7 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 3 ($N = 20, \alpha = 0.3$).

 (a) Wide-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.00000	1.00000	4.034e+02	8.939e+02	-1.137e+03	2.849e+03
2	1.10000	1.10000	6.466e+01	7.279e+02	-1.126e+03	2.018e+03
3	1.20000	1.20000	-1.527e+02	6.728e+02	-1.150e+03	2.385e+03
4	1.30000	1.30000	-4.719e+02	4.819e+02	-1.303e+03	1.269e+03
5	1.40000	1.40000	-7.135e+02	3.981e+02	-1.349e+03	1.284e+03
6	1.50000	1.50000	-9.197e+02	2.465e+02	-1.299e+03	1.108e+02
7	1.60000	1.60000	-1.101e+03	1.704e+02	-1.451e+03	-4.027e+02
8	1.70000	1.70000	-1.246e+03	1.233e+02	-1.513e+03	-7.318e+02
9	1.80000	1.80000	-1.477e+03	5.841e+01	-1.538e+03	-1.202e+03
10	1.90000	1.90000	-1.488e+03	4.041e+01	-1.538e+03	-1.382e+03
11	2.00000	2.00000	-1.491e+03	5.017e+01	-1.538e+03	-1.305e+03

 (b) Narrow-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.75000	1.75000	-1.363e+03	9.273e+01	-1.535e+03	-1.123e+03
2	1.76000	1.76000	-1.411e+03	9.021e+01	-1.535e+03	-1.085e+03
3	1.77000	1.77000	-1.438e+03	6.966e+01	-1.538e+03	-1.115e+03
4	1.78000	1.78000	-1.465e+03	6.302e+01	-1.538e+03	-1.230e+03
5	1.78140	1.78140	-1.465e+03	6.382e+01	-1.538e+03	-1.240e+03
6	1.79000	1.79000	-1.467e+03	6.010e+01	-1.538e+03	-1.287e+03
7	1.80000	1.80000	-1.477e+03	5.841e+01	-1.538e+03	-1.202e+03
8	1.81000	1.81000	-1.483e+03	5.415e+01	-1.538e+03	-1.230e+03
9	1.82000	1.82000	-1.483e+03	4.904e+01	-1.538e+03	-1.316e+03
10	1.83000	1.83000	-1.484e+03	4.606e+01	-1.538e+03	-1.313e+03
11	1.84000	1.84000	-1.477e+03	5.327e+01	-1.538e+03	-1.243e+03
12	1.85000	1.85000	-1.488e+03	4.896e+01	-1.538e+03	-1.281e+03

Table 5.8 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 4 ($N = 20, \alpha = 0.0$).

 (a) Wide-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.00000	1.00000	1.160e+02	1.720e+01	6.794e+01	1.654e+02
2	1.10000	1.10000	1.177e+02	1.700e+01	5.767e+01	1.573e+02
3	1.20000	1.20000	1.112e+02	1.606e+01	6.654e+01	1.474e+02
4	1.30000	1.30000	1.043e+02	1.741e+01	5.908e+01	1.553e+02
5	1.40000	1.40000	1.002e+02	1.665e+01	5.323e+01	1.448e+02
6	1.50000	1.50000	8.824e+01	1.609e+01	4.302e+01	1.301e+02
7	1.60000	1.60000	7.920e+01	1.412e+01	4.518e+01	1.166e+02
8	1.70000	1.70000	6.033e+01	1.351e+01	2.478e+01	1.036e+02
9	1.80000	1.80000	1.156e+01	3.604e+00	5.021e+00	2.841e+01
10	1.90000	1.90000	3.091e+01	1.168e+01	9.042e+00	6.397e+01
11	2.00000	2.00000	3.183e+01	1.200e+01	1.008e+01	6.888e+01

 (b) Narrow-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.75000	1.75000	4.632e+01	1.341e+01	1.954e+01	8.471e+01
2	1.76000	1.76000	4.113e+01	1.387e+01	1.327e+01	7.783e+01
3	1.77000	1.77000	3.085e+01	1.147e+01	6.789e+00	6.078e+01
4	1.78000	1.78000	1.870e+01	9.349e+00	3.799e+00	5.117e+01
5	1.78140	1.78140	1.651e+01	9.174e+00	4.454e+00	6.949e+01
6	1.79000	1.79000	1.145e+01	4.675e+00	4.028e+00	3.973e+01
7	1.80000	1.80000	1.156e+01	3.604e+00	5.021e+00	2.841e+01
8	1.81000	1.81000	1.492e+01	6.046e+00	5.140e+00	4.422e+01
9	1.82000	1.82000	1.788e+01	7.028e+00	5.259e+00	4.795e+01
10	1.83000	1.83000	2.235e+01	9.884e+00	4.028e+00	6.786e+01
11	1.84000	1.84000	2.596e+01	1.129e+01	8.014e+00	8.582e+01
12	1.85000	1.85000	2.705e+01	1.145e+01	8.224e+00	6.499e+01

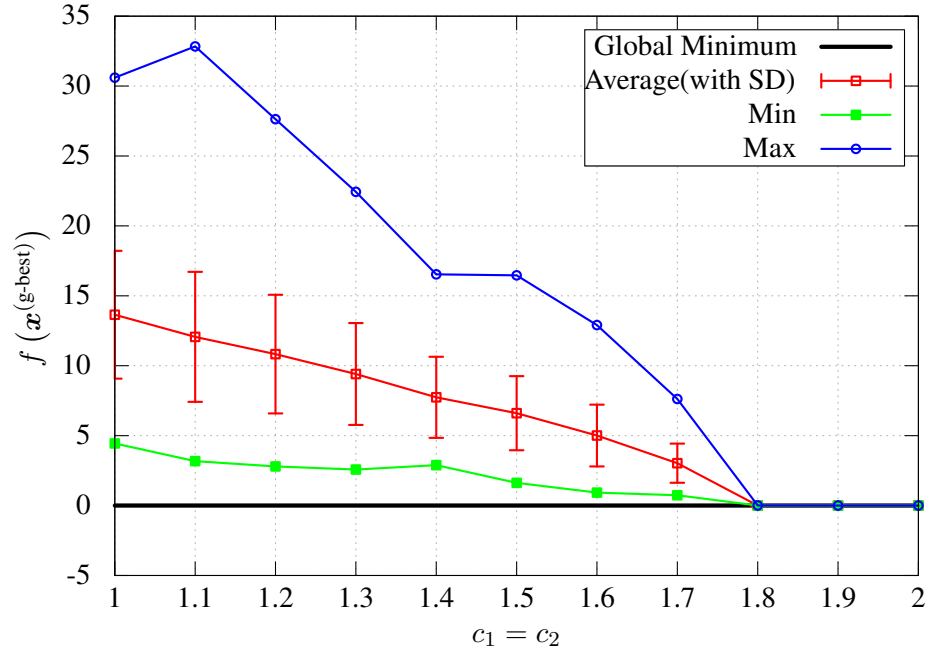
Table 5.9 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 4 ($N = 20, \alpha = 0.3$).

(a) Wide-ranged c

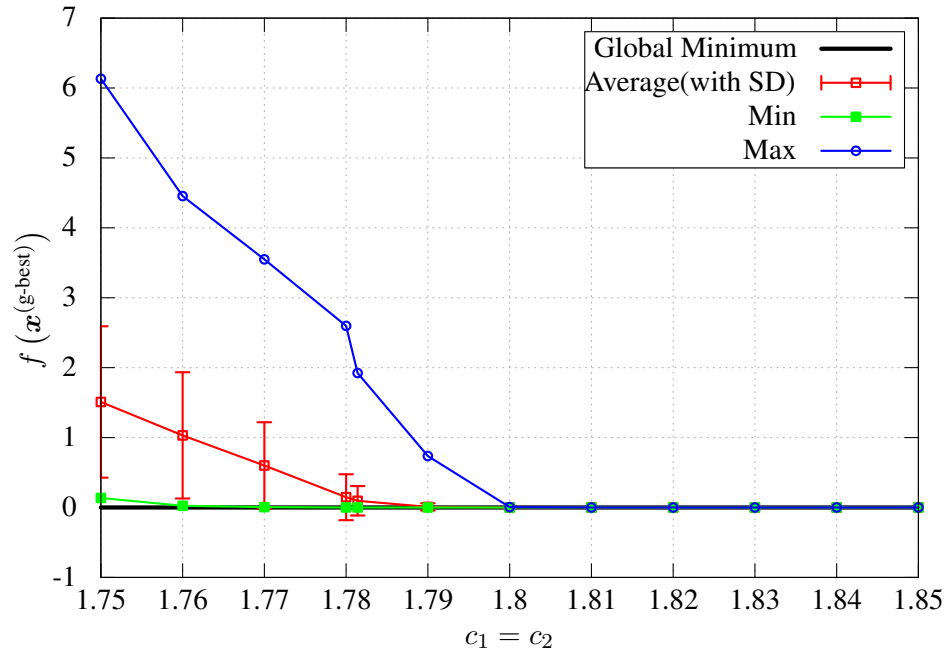
No.	c_1	c_2	mean	stdev	min	max
1	1.00000	1.00000	1.183e+02	1.727e+01	7.463e+01	1.613e+02
2	1.10000	1.10000	1.144e+02	1.758e+01	6.526e+01	1.662e+02
3	1.20000	1.20000	1.131e+02	1.698e+01	6.350e+01	1.547e+02
4	1.30000	1.30000	1.073e+02	1.665e+01	6.376e+01	1.489e+02
5	1.40000	1.40000	1.026e+02	1.596e+01	6.012e+01	1.540e+02
6	1.50000	1.50000	9.257e+01	1.634e+01	4.651e+01	1.353e+02
7	1.60000	1.60000	8.590e+01	1.533e+01	4.172e+01	1.247e+02
8	1.70000	1.70000	6.833e+01	1.441e+01	2.931e+01	9.985e+01
9	1.80000	1.80000	3.143e+01	1.107e+01	8.976e+00	6.738e+01
10	1.90000	1.90000	3.179e+01	1.235e+01	9.991e+00	7.087e+01
11	2.00000	2.00000	3.210e+01	1.383e+01	8.098e+00	9.390e+01

(b) Narrow-ranged c

No.	c_1	c_2	mean	stdev	min	max
1	1.75000	1.75000	5.683e+01	1.422e+01	2.253e+01	9.774e+01
2	1.76000	1.76000	5.303e+01	1.415e+01	1.508e+01	9.829e+01
3	1.77000	1.77000	4.429e+01	1.474e+01	1.395e+01	9.629e+01
4	1.78000	1.78000	3.618e+01	1.303e+01	1.207e+01	7.441e+01
5	1.78140	1.78140	3.386e+01	1.187e+01	1.214e+01	6.857e+01
6	1.79000	1.79000	3.213e+01	1.245e+01	6.505e+00	8.663e+01
7	1.80000	1.80000	3.143e+01	1.107e+01	8.976e+00	6.738e+01
8	1.81000	1.81000	3.255e+01	1.241e+01	9.974e+00	8.443e+01
9	1.82000	1.82000	3.217e+01	1.228e+01	7.996e+00	7.853e+01
10	1.83000	1.83000	3.132e+01	1.317e+01	7.975e+00	1.028e+02
11	1.84000	1.84000	3.261e+01	1.223e+01	1.197e+01	7.491e+01
12	1.85000	1.85000	3.315e+01	1.213e+01	1.003e+01	7.477e+01

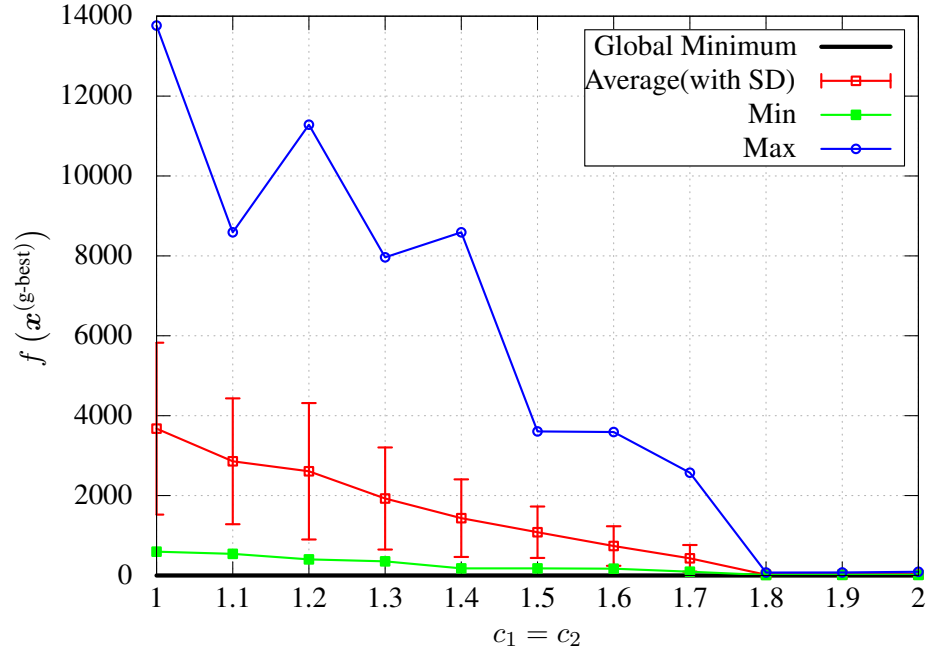


(a) Wide-ranged c

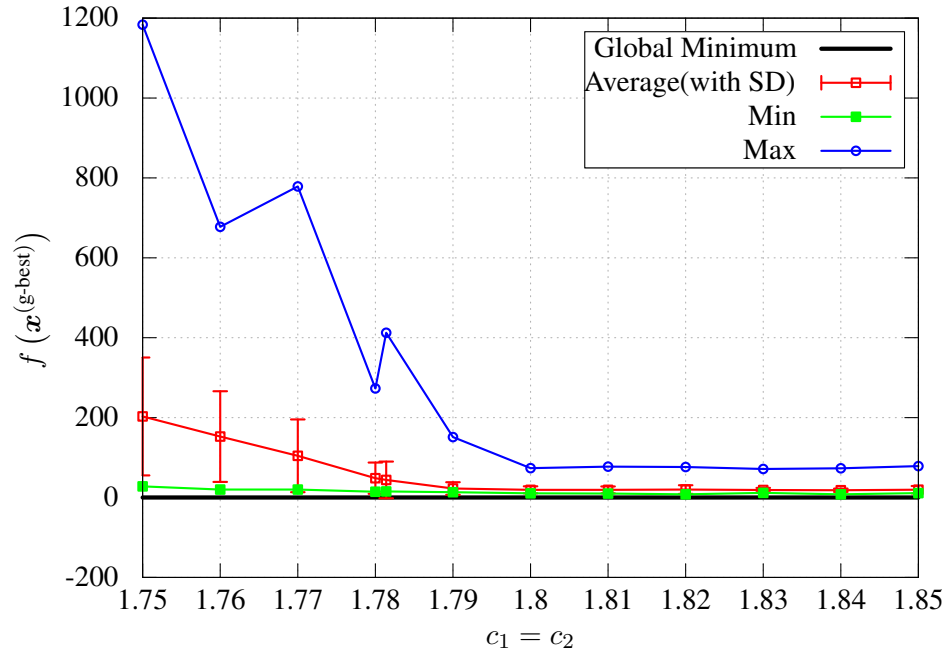


(b) Narrow-ranged c

Fig. 5.2 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 1 ($N = 20, \alpha = 0.0$).

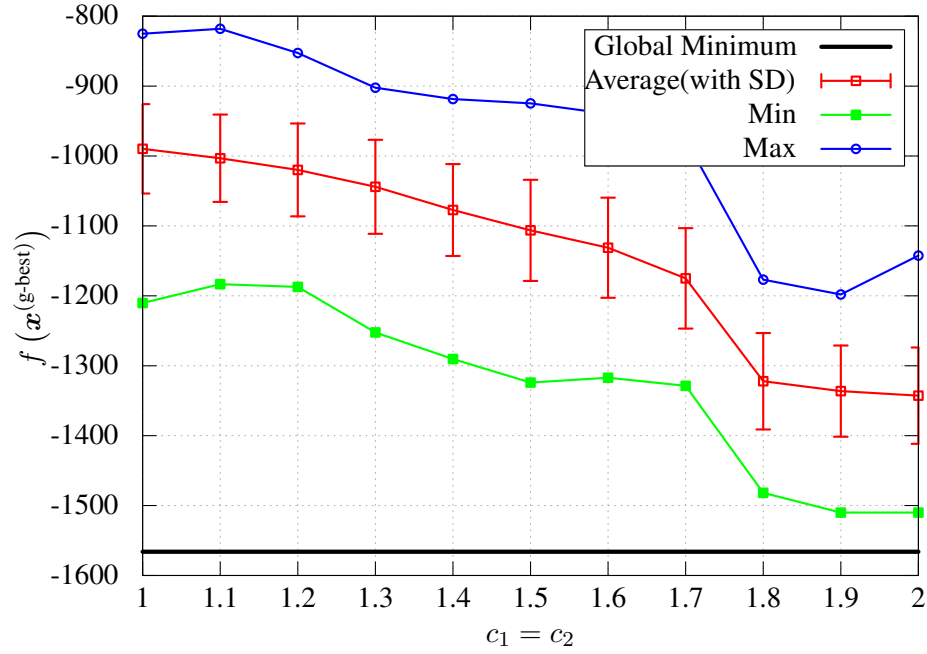


(a) Wide-ranged c

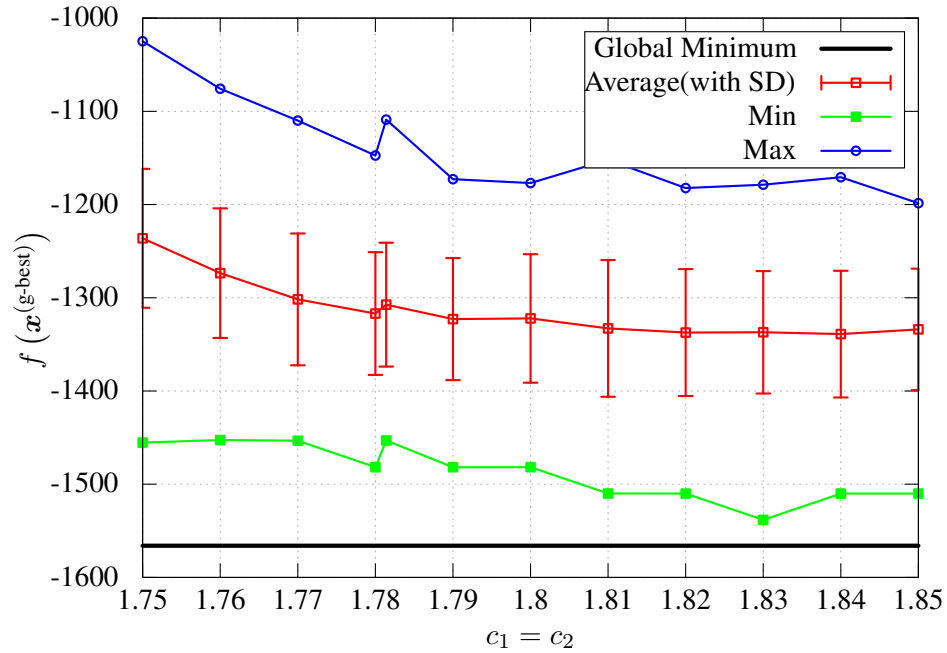


(b) Narrow-ranged c

Fig. 5.3 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 2 ($N = 20, \alpha = 0.0$).

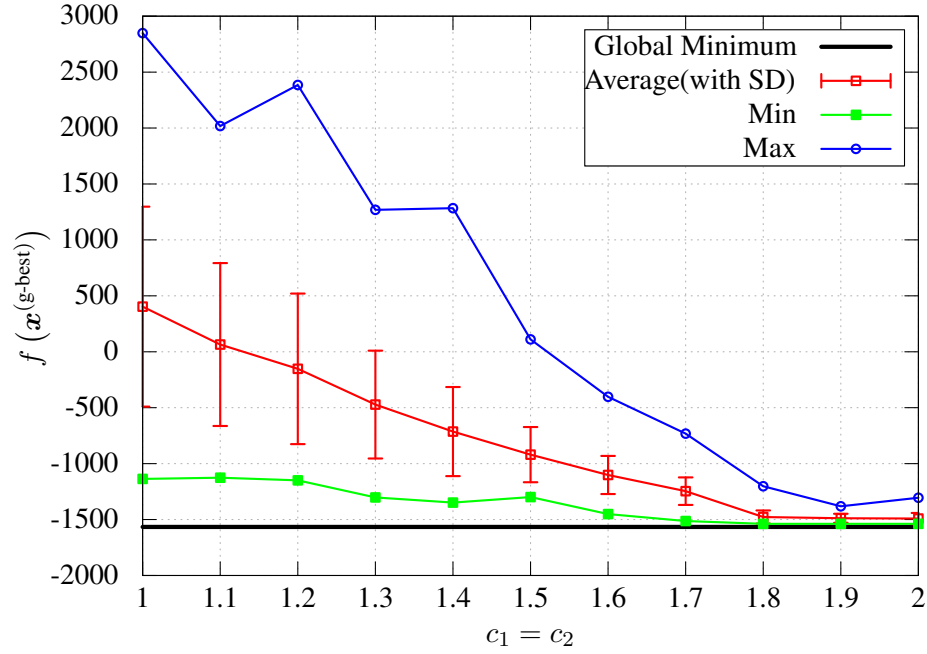


(a) Wide-ranged c

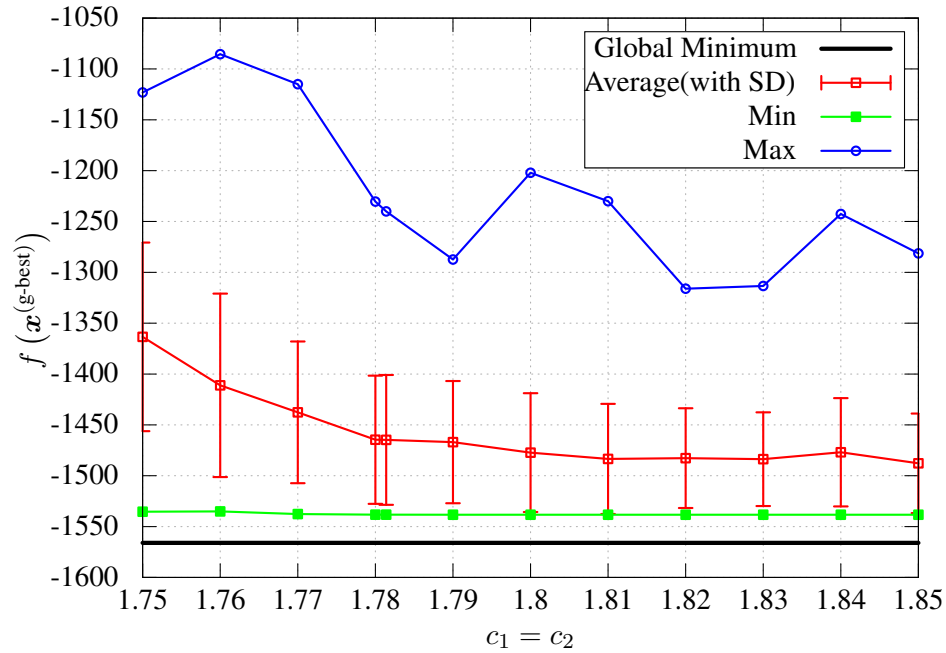


(b) Narrow-ranged c

Fig. 5.4 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 3 ($N = 20, \alpha = 0.0$).

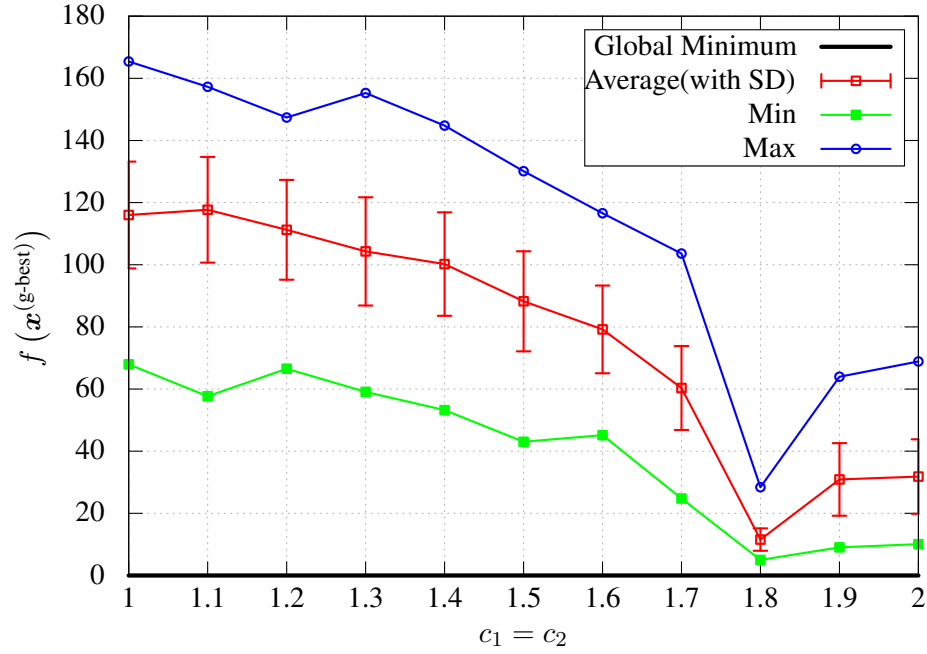


(a) Wide-ranged c

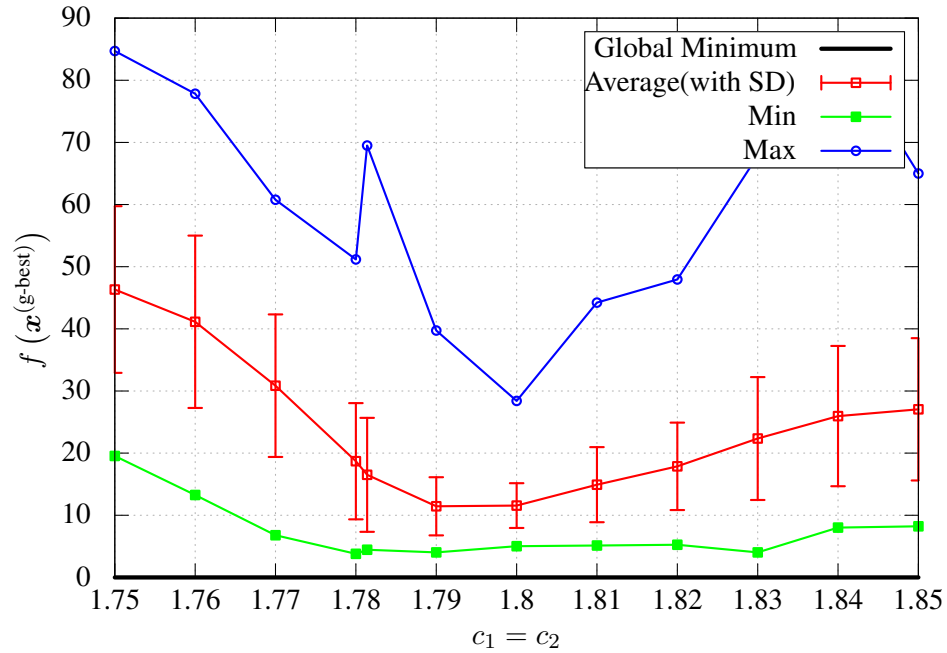


(b) Narrow-ranged c

Fig. 5.5 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 3 ($N = 20, \alpha = 0.3$).

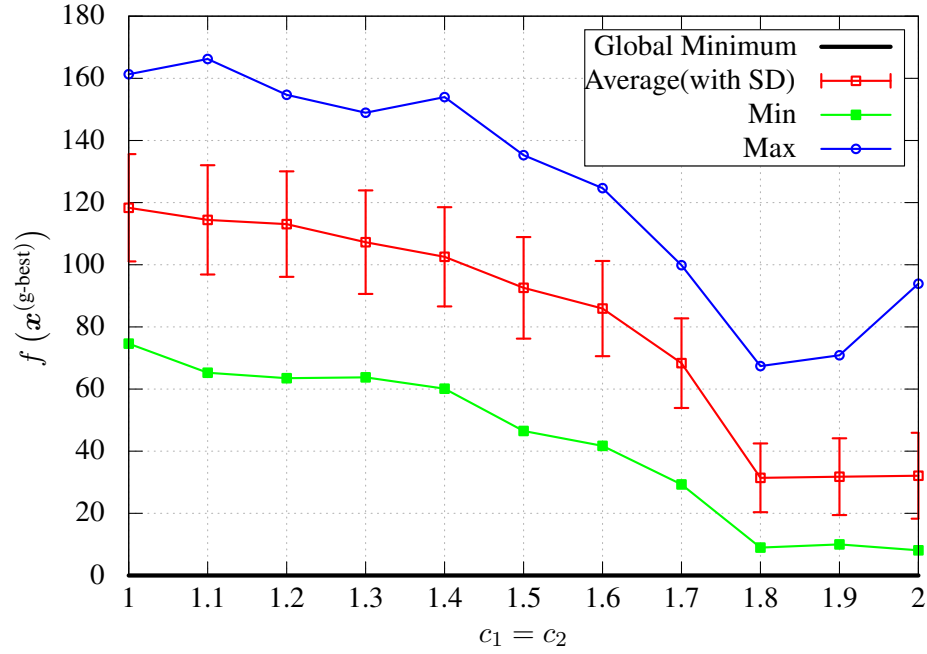


(a) Wide-ranged c

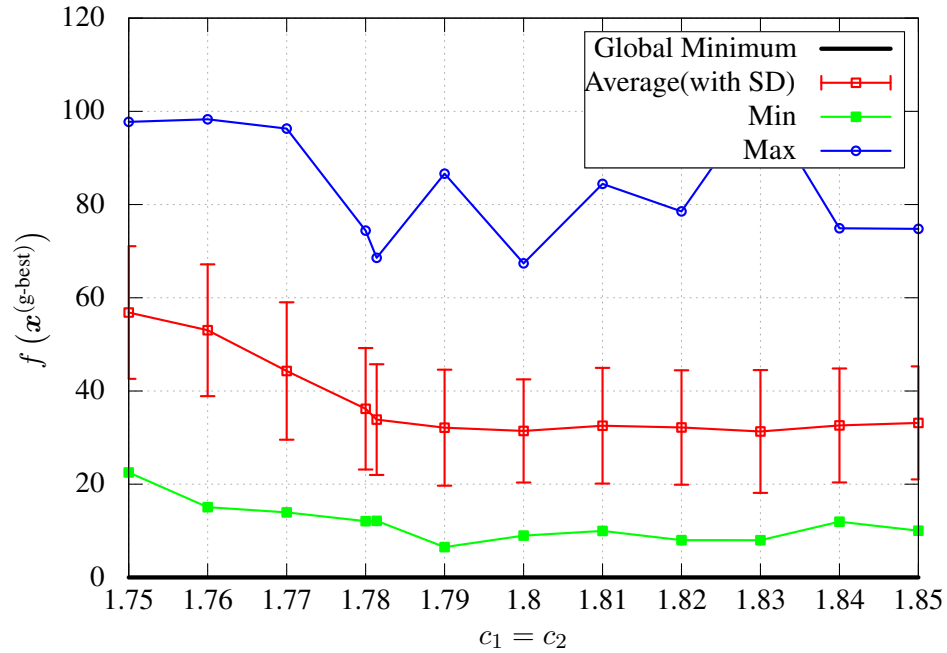


(b) Narrow-ranged c

Fig. 5.6 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 4 ($N = 20, \alpha = 0.0$).



(a) Wide-ranged c



(b) Narrow-ranged c

Fig. 5.7 Optimization performance of $\mathcal{A}_{\text{PSO}}$ for Prob. 4 ($N = 20, \alpha = 0.3$).

Table 5.10 Comparison between sustainable PSO and $\mathcal{A}_{\text{PSO}}$ in optimization performance.

(a) Prob. 1 (Sphere Function, $N = 20, \alpha = 0.0$)

No.	Algorithm	mean	stdev	min	max
1	Sustainable PSO	1.153e-04	9.456e-05	8.918e-06	6.572e-04
2	$\mathcal{A}_{\text{PSO}}$	2.634e-04	4.681e-04	2.229e-05	5.477e-03

(b) Prob. 2 (Rosenbrock's Function, $N = 20, \alpha = 0.0$)

No.	Algorithm	mean	stdev	min	max
1	Sustainable PSO	1.882e+01	9.685e+00	5.991e+00	7.579e+01
2	$\mathcal{A}_{\text{PSO}}$	1.922e+01	8.797e+00	1.055e+01	7.352e+01

(c) Prob. 3 (2^N minima Function, $N = 20, \alpha = 0.0$)

No.	Algorithm	mean	stdev	min	max
1	Sustainable PSO	-1.419e+03	5.620e+01	-1.533e+03	-1.218e+03
2	$\mathcal{A}_{\text{PSO}}$	-1.322e+03	6.892e+01	-1.482e+03	-1.177e+03

(d) Prob. 3 (2^N minima Function, $N = 20, \alpha = 0.3$)

No.	Algorithm	mean	stdev	min	max
1	Sustainable PSO	-1.415e+03	1.136e+02	-1.538e+03	-1.038e+03
2	$\mathcal{A}_{\text{PSO}}$	-1.477e+03	5.841e+01	-1.538e+03	-1.202e+03

(e) Prob. 4 (Rastrigin's Function, $N = 20, \alpha = 0.0$)

No.	Algorithm	mean	stdev	min	max
1	Sustainable PSO	7.663e+00	2.451e+00	2.025e+00	1.417e+01
2	$\mathcal{A}_{\text{PSO}}$	1.156e+01	3.604e+00	5.021e+00	2.841e+01

(f) Prob. 4 (Rastrigin's Function, $N = 20, \alpha = 0.3$)

No.	Algorithm	mean	stdev	min	max
1	Sustainable PSO	3.711e+01	2.157e+01	1.112e+01	1.221e+02
2	$\mathcal{A}_{\text{PSO}}$	3.143e+01	1.107e+01	8.976e+00	6.738e+01

5.5 本章のまとめ

本章では、第3章や第4章のように、既存の最適化アルゴリズムの動特性解析を目指すのではなく、その逆のアプローチとして、数理的動特性解析可能な新しいメタヒューリスティクスのフレームワークを構築し、そのなかで最適化アルゴリズムを問題に合わせて設計する方法を提案した。具体的には、各計算の反復において、各探索点の位置が正規乱数の線形結合および定常入力項により与えられる構造からなるフレームワーク「LNR」を提案し、同フレームワークに対する安定性解析をおこなった。また、PSOに着想を得た具体的なアルゴリズムの構築をおこない、本フレームワークに基づき有効な最適化アルゴリズムが構築でき、かつ本章で示した安定性定理に基づき、有効な最適化性能をもたらすパラメータ設定が可能であることを示した。

複雑な構造を通するメタヒューリスティクスのアルゴリズムは、そのパラメータをどのように設定すべきかという点に試行錯誤を要する。これに対して、本章で構築したフレームワークを用いれば、パラメータ設定の指針が理論的に与えられることとなり、パラメータ調整に関する労力を削減でき、実問題を解く最適化アルゴリズムの設計にも大きく貢献すると考えられる。

なお、本章でも言及しているとおり、LNRはPSOの探索方策に限らず多様な最適化方策を組み込むことが可能としており、より高度かつ最適化性能の高いアルゴリズムをLNRのもとで構築できると考えている。具体的なアイデアとしては、たとえばつぎのようなものが考えられる。

- (1) 目的関数の勾配やHesse行列といった解析的情報が利用できる場合、これらを用いてより局所的最適化性能を向上させる
- (2) 目的関数の勾配が利用できない場合、2.4.5項で述べた探索集団情報を用いた勾配近似法を用いて局所的最適化性能を向上させる
- (3) (2)のさらなる発展として、複数の探索点の探索履歴を活用して、目的関数を逐次的に近似する手法を内在させる。この方策は、応答曲面法やSurrogate-Modelとよばれ、目的関数の計算1回あたりの計算負荷が大きく、トータルでの目的関数計算回数を削減したい場合に有効とされている。

また、本論文では、LNRの本来の思想である、「問題に合わせて固有の探索方策を組み込みアルゴリズムを設計する」という点に関して、具体的な問題やそれに対するアルゴリズムを例示するまでには至っていない。これに対しては、とくに産業応用上の最適化問題に対して個別のアルゴリズム設計を試み、その有効性を評価したいと考えている。ただし、本論文ではLNRに基づくアルゴリズムが数理的に動特性解析可能であることは保証しているが、具体的にどのように「個別の問題に対する事前知識を反映するか」という指針までは与えておらず、この点の検討については今後の課題としたい。このほか、本論文では多峰性関数の大域的最適化に関して、全体を通して、アルゴリズムの安定性解析結果に基づいた「集中化と多様化」すなわち、局所的探索と大域的探索の両立を志向しているが、アルゴリズムの設計自由度が高くなれば、それ以外の方策をとることもできる。たとえば分枝限定法^[121]など、逐次探索範囲の枝刈りを併用して効率的に大域的探索をおこなうアルゴリズムをLNRで実装することも考えられる。

さらに、LNR 自体にも課題があり、改良の余地や実装上の工夫があると考えている。まず第一に、探索点位置の分散共分散行列や、それにしたがう多変数正規乱数の計算コストが高いことがあげられる。とくに後者では各探索時刻において分散共分散行列の Cholesky 分解（計算量は探索次元数 N に対して $O(N^2)$ ）の計算が必要であり、大規模な問題において計算速度への影響が懸念される。しかしながら、現状の LNR では、探索点位置の分散共分散行列の位置づけは、探索点分布の安定性を担う内部状態量に過ぎず、その役割を損なわない範囲で適当に厳密性を欠如させてもよいと考えられる。上述の課題に対する改善案としては、たとえばつぎのような方法が考えられる。

- (1) 正規乱数の生成に際して、1 探索時刻分古い探索点位置の分散共分散行列を用い、探索点位置の更新と分散共分散行列の計算を並列しておこなう。具体的には、更新式を

$$\mathbf{x}^{(p)}(k+1) = \sum_{m=1}^M A_m^{(p)}(k) \mathbf{Y}_m^{(p)}(k) + \mathbf{c}^{(p)}(\mathcal{X}(k), k) \quad (5.37a)$$

$$\mathbf{Y}_m^{(p)}(k) \sim \mathcal{N}(\mathbf{b}_m^{(p)}(\mathcal{X}(k), k), \Sigma(k-1)) \quad (5.37b)$$

$$\Sigma(k) = \frac{1}{P-1} \sum_{p=1}^P (\mathbf{x}^{(p)}(k) - \bar{\mathbf{x}}(k))(\mathbf{x}^{(p)}(k) - \bar{\mathbf{x}}(k))^{\top} \quad (5.37c)$$

とする（(5.37b) 式の右辺 $\Sigma(k-1)$ に注目されたい）。オリジナルの LNR では、(1) (5.1c) 式を用いて探索点位置の分散共分散行列 $\Sigma(k)$ を更新、(2) (1) で求めた $\Sigma(k)$ に基づき (5.1b) 式を用いて多変数正規乱数を発生させて $\mathbf{Y}_m^{(p)}(k)$ を生成、(3) 更新式 (5.1a) 式を用いて $\mathbf{x}^{(p)}(k+1)$ を更新、という手順を直列的にこなう必要があったが、上述の更新式では、(5.37a) 式と (5.37b) 式の計算は互いに依存しないため、(1)(2) 探索点位置の分散共分散行列の計算および正規乱数発生に必要な Cholesky 分解の計算と、(3) 探索点位置の更新を、たとえばマルチコア CPU 等を用いて同時並列的に計算でき、計算の効率化が見込める。

- (2) 探索点位置の分散共分散行列の非対角要素を計算せず、探索空間の各次元で独立に探索点位置の分散を計算し、その分散を用いて正規乱数を発生させる。この場合は Cholesky 分解は不要である。
- (3) 探索点位置の分散共分散行列の更新を探索時刻ごとではなく、適当な探索時刻間隔でおこなう。この場合は分散共分散行列の計算および乱数発生に必要な Cholesky 分解の計算の回数を単純に減らすことができる。

第6章 おわりに

6.1 本論文のまとめ

本論文では、数理的解析に基づくメタヒューリスティクスの設計、パラメータ調整をテーマとして、既存の代表的なメタヒューリスティクスである PSO や DE の収束性解析と、また数理的に動特性解析可能な多点型確率的最適化アルゴリズムのフレームワークとして「LNR」を提案した。

多くのメタヒューリスティクスは、(1) 複数の探索点を用いた多点型探索、(2) 擬似乱数を用いた確率的探索、といった数学的に取り扱いづらい探索方策を採用しており、とくに後者に関しては、探索点の位置を決定論的に解析することを原理的に不可能としている。そこで本研究では、これらの特徴からメタヒューリスティクスを「多点型確率的最適化アルゴリズム」として数学的に解釈したうえで、最適化アルゴリズムが多数回の計算の反復を前提とすることに着目し、その巨視的な時間間隔での動特性の考察を通してアルゴリズムの収束性解析を試みた。この考え方は、本論文第3章の PSO の収束性解析や、第5章の数理的解析可能な多点型確率的最適化アルゴリズムのフレームワークの安定性定理の導出において利用されており、本研究の中核をなす概念となっている。

以下、各章の内容を改めてまとめる。

第1章では、まず最適化問題と最適化アルゴリズムの研究に関する歴史的経緯を述べ、近年開発が活発化しているメタヒューリスティクスには数理的根拠が必ずしもないこと、またそれを理由とする実用上の課題を指摘したうえで、「数理的解析に基づくメタヒューリスティクスの設計、パラメータ調整」という本論文の目的、位置づけについて明確にした。第2章では、具体的な議論に入る前に、本論文では扱う最適化問題、およびそれを解く多点型確率的最適化アルゴリズムの構造を数学的に記述した。また、第3章および第4章で取り扱う PSO、DE についても、その概要や応用研究や改良研究について概観するとともに、これらの手法に数学的裏付けがないこと、またそれに起因する実用上の課題を指摘するなど、改めて問題の提起をおこなった。

第3章では、代表的な多点型確率的最適化アルゴリズムのひとつである PSO に対して、等価な動特性を有する確率システムを構築し、その安定性解析をおこない、パラメータとアルゴリズムの動特性について数学的に議論した。また、その副産物として、持続的に局所的探索と大域的探索の繰り返しにより高い大域的最適化性能を実現する「持続的探索パラメータ」の具体例を提案し、数値実験によりその有効性を確認した。ここで導いた具体的なパラメータ設定値は、PSO の固有のダイナミクスのみに依存し、最適化対象の目的関数の形状等に依存しないため、本質的にあらゆる最適化問題に対して適用できるという点をあらためて強調しておく。

第4章では、代表的な多点型確率的最適化アルゴリズムのひとつである DE に対して、

PSOと同様、パラメータとアルゴリズムの動特性について数学的に議論した。DEはPSOとは異なり、複数の探索点の探索空間上の分布がその動特性に大きく影響するため、解析が困難であり、PSOと同程度の精緻な解析や、実用上有益なパラメータ設定を提案するまでは至らなかった。しかしながら、「探索点分布」という連続的な概念を、問題が1次元かつ探索点数が多数という仮定のもとで、最大エントロピー法を用いて推定することが可能であることを確認した。

第5章では、第3章と第4章でおこなった「既存のアルゴリズムの解析」とは逆のアプローチとして、数理的に動特性解析可能な多点型確率的最適化アルゴリズムの「フレームワーク」のもとで具体的な最適化アルゴリズムを構築するという方策を提唱し、具体的なフレームワークであるLNRや、その探索点分布に関する安定性定理を示した。また、LNRに基づく具体的な最適化アルゴリズムの例として、PSOの探索方策に着想を得た A_{PSO} を例示し、PSOと同様、探索点分布が収束と発散、すなわち局所的探索と大域的探索を繰り返すようなパラメータ設定を与えることで、高い最適化性能を示すことを数値実験により確認した。この「局所的探索と大域的探索を繰り返すようなパラメータ設定」を数理的に導くことが可能という点がLNRの最大の特長であり、実問題に対して問題の事前知識を活用して最適化アルゴリズムを構築する場合でもこの特長を享受できる。

6.2 今後の展望

1.1節でも述べたとおり、最適化問題を解くにあたって「最適化アルゴリズムで解きやすいような問題の定式化」と「問題に合わせた最適化アルゴリズムの選択・設計」は、有効な最適化を実現するための両輪である。アルゴリズムの設計では、メタヒューリスティクスの設計指針に従えば、さまざまな形で問題に合わせた最適化方策を組み込むことが可能であるが、その一方で、アルゴリズムの動特性については検討が難しく、設計されたアルゴリズムのパラメータ調整など、本来労力を要するべきでない部分で新たな課題が生じることとなる。本論文で提案した多点型確率的最適化アルゴリズムのフレームワークであるLNRは、最適化アルゴリズム設計に数理的な根拠を与える手段のひとつであり、かつ最適化アルゴリズムの側からより多くの問題に対してその問題知識を十分に活用できるよう構築されている。具体的な今後の展望としては、様々な最適化問題に対してLNRを用いて個別の最適化アルゴリズムを設計・適用し、LNRの有効性を検証していくことを考えている。また、より広く、メタヒューリスティクスのアルゴリズムの設計柔軟性と設計・パラメータ調整時の数理的な根拠の両立に資する研究に今後も尽力していきたい。

謝辞

本研究を行うにあたって、指導教員である慶應義塾大学大学院 理工学研究科 基礎理工学専攻 後期博士課程在学中、慶應義塾大学理工学部 物理情報工学科 相吉英太郎 教授 に多くの御指導、御教示を賜りました。相吉教授には、著者の慶應義塾大学物理情報工学科 4 年次在籍中より同大学大学院基礎理工学専攻修士課程を経て、同大学院同専攻後期博士課程にいたるまで長きに亘り多大なる御世話を賜りました。この場を借りて謹んで感謝の意を申し上げます。とくに、修士課程修了後、一度大学を離れ、企業にて技術者として働きつつ研究を進めていた著者に対し、社会人として後期博士課程に入学することを快くご了解頂いたことに対して、重ねて御礼申し上げます。

また、本論文の審査に際して、副査として御指摘、御指導を賜りました、慶應義塾大学理工学部物理情報工学科 本多敏 教授、足立修一 教授、山本直樹 准教授、首都大学東京大学院理工学研究科 安田恵一郎 教授、各先生に対して厚く謝意を表します。理論面での不備の指摘や改善案のご提案や、本研究の応用や発展性の示唆など、多くの貴重かつ有益な御意見を賜ることができ、本論文の完成度をより高いものとすることができました。

本研究は、上述したとおり、著者が社会人として慶應義塾大学大学院理工学研究科 後期博士課程に在籍し、そのなかで行われたものであります。後期博士課程在学中、株式会社テプコシステムズ 原子力炉心技術部、および株式会社 IHI 技術開発本部 総合開発センター 制御技術開発部の皆様には、在職のまま社会人博士課程へ進学することに対して深いご理解を頂き、また在学中も様々な面でご支援を賜りました。厚く謝意を表します。

最後に、個別に御名前を挙げることは差し控えさせていただきますが、本研究を進めるにあたって貴重な議論、御意見を賜るとともに、友人として親しくさせて頂いた慶應義塾大学理工学部 相吉研究室の先輩、同期、後輩の皆様に対し、深く感謝の意を申し上げます。いつの日か再びめぐり逢い、またその喜びを分かち合えることを、心より待ち望んでおります。

2015 年 2 月 慶應義塾大学 矢上キャンパスの研究室にて

小熊 祐 司

参考文献

- [1] 山川宏, 最適設計ハンドブックー基礎・戦略・応用一, 朝倉書店, 2003.
- [2] P. Toth, and D. Vigo, eds., The Vehicle Routing Problem, SIAM, 2001.
- [3] 西田大, 中川賀津也, 相田剛, 熊本和浩, 小西伸之, “最適輸配送計画問題への数理計画法の適用,” オペレーションズ・リサーチ, vol.47, no.1, pp.22–26, 2002.
- [4] 宮本定明, “積み付け問題の現状と展望,” 知能と情報, vol.18, no.6, pp.821–827, 2006.
- [5] 山下勝比拡, 原裕淳, “大規模生産計画のリアルタイム最適化,” 電気学会論文誌, vol.107, no.12, pp.1265–1268, 1987.
- [6] 藤田薫, 江本源一, 竹下聡彦, 佐中俊哉, “化学産業における最適化技術の適用,” オペレーションズ・リサーチ, vol.48, no.8, pp.549–554, 2003.
- [7] 福山良和, “メタヒューリスティック手法の電力エネルギー分野への適用例,” 電気学会論文誌 B, vol.124, no.5, pp.679–682, 2004.
- [8] 電気学会 GA・ニューロを用いた学習法とその応用調査専門委員会, 学習とそのアルゴリズム, 森北出版, 2002.
- [9] 枇々木規雄, 田辺隆人, ポートフォリオ最適化, 朝倉書店, 2005.
- [10] G. Dantzig, Linear Programming and Extensions, Princeton University Press, 1998.
- [11] 大嶋正裕, “モデル予測制御ー理論の誕生・展開・発展一,” 計測と制御, vol.39, no.5, pp.321–325, 2000.
- [12] 津田宏治, “サポートベクターマシン: 最適化からのアプローチ,” オペレーションズ・リサーチ, vol.46, no.5, pp.249–254, 2001.
- [13] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons, 1987.
- [14] R. Byrd, R. Schnabel, and G. Shultz, “A Trust Region Algorithm for Nonlinearly Constrained Optimization,” SIAM Journal on Numerical Analysis, vol.24, no.5, pp.1152–1170, 1987.
- [15] 渡辺祐樹, 岡本卓, 相吉英太郎, “目的関数変換による分岐調整を利用したカオス最適化手法: 大域的最適化手法「水抜き法」の提案,” 電気学会論文誌 C, vol.126, no.12, pp.1559–1568, 2006.

- [16] T. Okamoto, and E. Aiyoshi, "The Improved Draining Method and Its Application to Proper Benchmark Problems," Proceedings of SICE-ICASE International Joint Conference 2006, pp.2190–2195, 2006.
- [17] J.A. Nelder, and R. Mead, "A Simplex Method for Function Minimization," The Computer Journal, vol.7, pp.308–313, 1965.
- [18] R. Rockafellar, Convex Analysis, NJ: Princeton University Press, 1997.
- [19] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., 1989.
- [20] H.P. Schwefel, Evolution and Optimum Seeking, John Wiley & Sons, Inc., 1995.
- [21] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization," Proceedings of International Conference on Neural Networks 1995, vol.4, pp.1942–1948, 1995.
- [22] Y. Shi, and R. Eberhart, "A Modified Particle Swarm Optimizer," Proceedings of IEEE International Conference on Evolutionary Computation 1998, 1998.
- [23] R. Storn, and K. Price, "Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," TR-95-012, pp.1–12, 1995.
- [24] R. Storn, and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," Journal of Global Optimization, vol.11, no.4, pp.341–359, 1997.
- [25] S. Dasgupta, S. Das, A. Biswas, and A. Abraham, "On Stability and Convergence of the Population-Dynamics in Differential Evolution," AI Communications, vol.22, no.1, pp.1–20, 2009.
- [26] S. Ghosh, S. Das, A. Vasilakos, and K. Suresh, "On Convergence of Differential Evolution Over a Class of Continuous Functions With Unique Global Optimum," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol.42, no.1, pp.107–124, Feb. 2011.
- [27] E. Campana, G. Fasano, and A. Pinto, "Dynamic Analysis for the Selection of Parameters and Initial Population, in Particle Swarm Optimization," Journal of Global Optimization, vol.48, no.3, pp.347–397, 2010.
- [28] C. Chen, and Y. Chen, "Convergence Time Analysis of Particle Swarm Optimization Based on Particle Interaction," Advances in Artificial Intelligence, vol.2011, 2011.
- [29] K. Yasuda, and N. Iwasaki, "Adaptive Particle Swarm Optimization Using Velocity Information of Swarm," Proceedings of IEEE International Conference on Systems, Man, and Cybernetics 2004, vol.4, pp.3475–3481, 2004.

- [30] J. Zhang, and A. Sanderson, “JADE: Adaptive Differential Evolution With Optional External Archive,” IEEE Transactions on Evolutionary Computation, vol.13, no.5, pp.945–958, 2009.
- [31] F. Neri, and V. Tirronen, “Scale Factor Local Search in Differential Evolution,” Memetic Computing, vol.1, no.2, pp.153–171, Feb. 2009.
- [32] R. Tanabe, and A. Fukunaga, “Success-History Based Parameter Adaptation for Differential Evolution,” IEEE Congress on Evolutionary Computation 2013, pp.71–78, 2013.
- [33] S. Jagawa, T. Yoshii, and A. Fukao, “Boiling Water Reactor Loading Pattern Optimization Using Simple Linear Perturbation and Modified Tabu Search Methods,” Nuclear Science and Engineering, vol.138, no.1, pp.67–77, 2001.
- [34] Y. Kobayashi, and E. Aiyoshi, “Optimization of Water Reactor Loading Pattern Using Two-Stage Genetic Algorithm,” Nuclear Science and Engineering, vol.142, pp.119–139, 2002.
- [35] Y. Kobayashi, and E. Aiyoshi, “Optimization of Boiling Water Reactor Loading Pattern Using an Improved Genetic Algorithm,” Nuclear Technology, vol.143, pp.144–151, 2002.
- [36] 石亀篤志, “産業応用のためのシステム最適化とベンチマーク問題,” 平成 26 年電気学会全国大会講演論文集, 2014.
- [37] 岡本卓, “ベンチマーク問題の実装と公開,” 平成 26 年電気学会全国大会講演論文集, 2014.
- [38] 大住晃, 確率システム入門, 朝倉書店, 2002.
- [39] E.H. Moore, “On the Reciprocal of the General Algebraic Matrix,” Bulletin of the American Mathematical Society, vol.26, no.9, pp.394–395, 1920.
- [40] R. Penrose, and J.A. Todd, “A Generalized Inverse for Matrices,” Mathematical Proceedings of the Cambridge Philosophical Society, vol.51, p.406, 1955.
- [41] 福島雅夫, 非線形最適化の基礎, 朝倉書店, 2001.
- [42] 茨木俊秀, 最適化の数学, 共立出版, 2011.
- [43] N. Karmarkar, “A New Polynomial-Time Algorithm for Linear Programming,” Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, pp.302–311, 1984.
- [44] 相吉英太郎, 志水清孝, 数理計画法演習, 朝倉書店, 1985.
- [45] 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・システム部門大会講演論文集, pp.457–462, 2008.

- [46] J.C. Spall, “A Stochastic Approximation Technique for Generating Maximum Likelihood Parameter Estimates,” 1987 American Control Conference, 1987.
- [47] J. Alspector, R. Meir, B. Yuhua, A. Jayakumar, and D. Lippe, “A Parallel Gradient Descent Method for Learning in Analog VLSI Neural Networks,” NIPS-1992, pp.836–844, 1993.
- [48] G. Cauwenberghs, “A Fast Stochastic Error-Descent Algorithm for Supervised Learning and Optimization,” Advances in Neural Information Processing Systems, vol.244-251, 1993.
- [49] 前田裕, “同時摂動型最適化法とその応用,” システム/制御/情報, vol.52, no.2, pp.47–53, 2008.
- [50] 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 Science, vol.E98-A, no.2, p.(in press), 2015.
- [51] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, “Optimization by Simulated Annealing,” Science (New York, N.Y.), vol.220, pp.671–680, 1983.
- [52] 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,” 電気学会論文誌 C, vol.130, no.1, pp.29–38, 2010.
- [53] 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,” 電気学会論文誌 C, vol.131, no.5, pp.1020–1030, 2011.
- [54] T. Okamoto, and E. Aiyoshi, “Global Optimization Using a Synchronization of Multiple Search Points Autonomously Driven by a Chaotic Dynamic Model,” Journal of Global Optimization, vol.41, no.2, pp.219–244, Aug. 2008.
- [55] 小熊祐司, 古澤敦郎, 相吉英太郎, “離散構造制約条件付き最適化問題に対する PSO を用いた進化計算,” 計測自動制御学会論文集, vol.45, no.10, pp.512–521, 2009.
- [56] 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,” 電気学会論文誌 C, vol.122, no.3, pp.320–323, 2002.
- [57] W. Davidon, “Variable Metric Method for Minimization,” SIAM Journal on Optimization, vol.1, no.1, pp.1–17, 1991.
- [58] R. Fletcher, and M. Powell, “A Rapidly Convergent Descent Method for Minimization,” The Computer Journal, vol.6, no.2, pp.163–168, 1963.
- [59] C. Broyden, “The Convergence of a Class of Double-Rank Minimization Algorithms 1. General Considerations,” IMA Journal of Applied Mathematics, vol.6, no.1, pp.76–90, 1970.

- [60] C. Broyden, "The Convergence of a Class of Double-Rank Minimization Algorithms 2. The New Algorithm," *IMA Journal of Applied Mathematics*, vol.6, no.3, pp.222–231, 1970.
- [61] R. Fletcher, "A New Approach to Variable Metric Algorithms," *The Computer Journal*, vol.13, no.3, pp.317–322, 1970.
- [62] D. Goldfarb, "A Family of Variable-Metric Methods Derived by Variational Means," *Mathematics of Computation*, vol.24, no.109, pp.23–26, 1970.
- [63] D. Shanno, "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computation*, vol.24, no.111, pp.647–656, 1970.
- [64] 久保田光一, 伊理正夫, アルゴリズムの自動微分と応用, コロナ社, 1998.
- [65] J. Kiefer, "Sequential Minimax Search for a Maximum," *Proceedings of the American Mathematical Society*, vol.4, no.3, pp.502–506, 1953.
- [66] 柳浦睦憲, 茨木俊秀, 組合せ最適化 —メタ戦略を中心として—, 朝倉書店, 2001.
- [67] 外崎真造, 久野誉人, "マルチスタート単体法による多峰関数の最適化," *数理解析研究所講究録*, vol.1629, pp.142–151, 2009.
- [68] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic Neural Networks," *Physics letters A*, vol.144, no.6-7, pp.333–340, 1990.
- [69] I. Tokuda, and K. Onodera, "Global Bifurcation Scenario for Chaotic Dynamical Systems that Solve Optimization Problems and Analysis of Their Optimization Capability," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol.81, no.2, pp.1–12, 1998.
- [70] K. Masuda, and E. Aiyoshi, "Global Optimization Method Using Chaos of Discrete Gradient Dynamics," *Proceedings of IFAC Workshops ALCOSP and PSYCO*, pp.825–830, 2004.
- [71] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers & Operations Research*, vol.13, no.5, pp.533–549, 1986.
- [72] 相吉英太郎, "メタヒューリスティックスとは何か Particle Swarm Optimization を中心として," *電気学会誌*, vol.126, no.9, pp.614–617, 2006.
- [73] R. 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.
- [74] I.C. Trelea, "The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection," *Information Processing Letters*, vol.85, pp.317–325, 2003.

- [75] 鈴木亮平, 河合富貴子, 中沢親志, 松井哲郎, 相吉英太郎, “PSO を用いたモデル予測制御のパラメータ最適化,” 電気学会論文誌 C, vol.129, no.3, pp.432–440, 2009.
- [76] 福山良和, “メタヒューリスティック手法の電力・エネルギー分野への適用例,” 電気学会論文誌 C, vol.124, no.5, pp.679–682, 2004.
- [77] 福山良和, “電力系統の無効電力配分計画問題への PSO の適用比較検討,” 電気学会論文誌 C, vol.124, no.5, pp.690–696, 2004.
- [78] 宮崎隆彦, 秋澤淳, 柏木孝夫, 赤平亮, “大域的最適化手法の適用による吸着冷凍機の性能向上: Particle Swarm Optimization によるサイクルタイムの最適化,” 日本冷凍空調学会論文集, vol.25, no.1, pp.9–19, 2008.
- [79] M. Miyatake, M. Veerachary, F. Toriumi, F. Nobuhiko, and K. Hideyoshi, “Maximum Power Point Tracking of Multiple Photovoltaic Arrays: A Particle Swarm Optimization Approach,” IEEE Transactions on Aerospace and Electronic Systems, vol.47, no.1, pp.367–380, 2011.
- [80] 飯間等, 黒江康明, “Particle Swarm Optimization に基づく強化学習,” システム/制御/情報: システム制御情報学会誌, vol.57, no.10, pp.408–414, 2013.
- [81] R. Mendes, P. Cortez, M. Rocha, and J. Neves, “Particle Swarms for Feedforward Neural Network Training,” Proceedings of the 2002 International Joint Conference on Neural Networks, vol.2, pp.1895–1899, 2002.
- [82] S. Pandey, L. Wu, S.M. Guru, and R. Buyya, “A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments,” 2010 24th IEEE International Conference on Advanced Information Networking and Applications, pp.400–407, 2010.
- [83] 村田秀樹, 安田恵一郎, 相吉英太郎, “非線形散逸項を有する Particle Swarm Optimization 法の提案,” 電気学会論文誌 C, vol.127, no.5, pp.787–792, 2007.
- [84] 平岡創土, 岡本卓, 相吉英太郎, “繰り返し型探索指針による Particle Swarm Optimization の改良,” 電気学会論文誌 C, vol.128, no.7, pp.1143–1153, 2008.
- [85] 石井良尚, 岡本卓, 相吉英太郎, “大域的持続探索のための非同期世代交代型 Particle Swarm Optimization,” 電気学会論文誌 C, vol.131, no.3, pp.626–634, 2011.
- [86] J. Kennedy, and R. Mendes, “Population Structure and Particle Swarm Performance,” Proceedings of the 2002 Congress on Evolutionary Computation, vol.2, pp.1671–1676, 2002.
- [87] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J. Pan, “Diversity Enhanced Particle Swarm Optimization with Neighborhood Search,” Information Sciences, vol.223, no.20, pp.119–135, 2013.

- [88] C. Cleghorn, and A. Engelbrecht, "Particle Swarm Convergence: Standardized Analysis and Topological Influence," *Swarm Intelligence Lecture Notes in Computer Science* Volume 8667, pp.134–145, 2014.
- [89] J. Wang, and D. Wang, "Particle Swarm Optimization with a Leader and Followers," *Progress in Natural Science*, vol.18, no.11, pp.1437–1443, Nov. 2008.
- [90] W. Zhang, and X. Xie, "DEPSO: Hybrid Particle Swarm with Differential Evolution Operator," *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics* 2003, vol.4, pp.3816–3821, 2003.
- [91] M. Noel, and T. Jannett, "Simulation of a New Hybrid Particle Swarm Optimization Algorithm," *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory* 2004, pp.150–153, 2004.
- [92] N. Nakagawa, A. Ishigame, and K. Yasuda, "Particle Swarm Optimization with Approximate Gradient," *IEEJ Transactions on Electrical and Electronic Engineering*, vol.3, no.5, pp.590–592, Sept. 2008.
- [93] V. Plevris, and M. Papadrakakis, "A Hybrid Particle Swarm-Gradient Algorithm for Global Structural Optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol.26, no.1, pp.48–68, Apr. 2011.
- [94] E. Kita, and Y.b. Shin, "Effectiveness of Second Best Particle Information for Particle Swarm Optimization," *Journal of Computer Science*, vol.9, no.11, pp.1461–1471, 2013.
- [95] D.G. Regulwar, "Differential Evolution Algorithm with Application to Optimal Operation of Multipurpose Reservoir," *Journal of Water Resource and Protection*, vol.2, no.6, pp.560–568, 2010.
- [96] J.V. Sickel, K. Lee, and J. Heo, "Differential Evolution and Its Applications to Power Plant Control," *Proceedings of International Conference on Intelligent Systems Applications to Power Systems*, pp.1–6, 2007.
- [97] 澤井陽輔, 篠原悠, 小野智司, 中山茂, 川崎洋, "3次元形状位置合わせにおける進化計算アルゴリズムの比較検討と全周復元への応用," *情報処理学会論文誌 数理モデル化と応用*, vol.6, no.2, pp.104–118, 2013.
- [98] S. Goudos, Z. Zaharis, and T. Yioultsis, "Application of a Differential Evolution Algorithm with Strategy Adaptation to the Design of Multi-Band Microwave Filters for Wireless Vommunications," *Progress In Electromagnetics Research*, vol.109, pp.123–137, 2010.
- [99] S. Kukkonen, "Solving the Molecular Sequence Alignment Problem with Generalized Differential Evolution 3 (GDE3)," *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, pp.302–309, 2007.

- [100] 阪井節子, 高濱徹行, “Differential Evolution による非線形最適化:直交ベクトルを用いた回転不変性を有する交叉の実現—,” 数理解析研究所講究録, vol.1734, pp.188–195, 2011.
- [101] 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.
- [102] 芦田高志, ラックターウォンマット, “局所的降下方向と差分変異個体の混合による Differential Evolution の改良,” 電子情報通信学会論文誌 D, vol.J96-D, no.3, pp.432–440, 2013.
- [103] 串田淳一, 大場和久, 亀井且有, “REAL: Differential Evolution における関数評価回数の削減の提案,” 進化計算学会論文誌, vol.1, no.1, pp.79–88, 2010.
- [104] X. Yang, and S. Deb, “Cuckoo Search via Lévy Flights,” World Congress on Nature & Biologically Inspired Computing, pp.210–214, 2009.
- [105] X. Yang, “Firefly Algorithms for Multimodal Optimization,” Stochastic algorithms: foundations and applications, vol.5792, pp.169–178, 2009.
- [106] D. Karaboga, “An Idea Based on Honey Bee Swarm for Numerical Optimization,” Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department,, 2005.
- [107] D. Karaboga, and B. Basturk, “On the Performance of Artificial Bee Colony (ABC) Algorithm,” Applied Soft Computing, vol.8, no.1, pp.687–697, Jan. 2008.
- [108] J.A. Ruiz-Vanoye, and O. Díaz-Parra, “Similarities between Meta-Heuristics Algorithms and the Science of Life,” Central European Journal of Operations Research, vol.19, no.4, pp.445–466, 2011.
- [109] 染谷博司, “進化型計算の挙動制御のためのダイナミクス解析,” システム/制御/情報, vol.57, no.2, pp.67–72, 2013.
- [110] M. Clerc, “Stagnation Analysis in Particle Swarm Optimisation or What Happens When Nothing Happens,” Department of Computer Science, University of Essex, Technical Report CSM-460, no.1, 2006.
- [111] R. Poli, “Mean and Variance of the Sampling Distribution of Particle Swarm Optimizers During Stagnation,” IEEE Transactions on Evolutionary Computation, vol.13, no.4, pp.712–721, Aug. 2009.
- [112] K. Yasuda, and N. Iwasaki, “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.652–659, 2008.

- [113] R. Poli, “Dynamics and Stability of the Sampling Distribution of Particle Swarm Optimisers via Moment Analysis,” *Journal of Artificial Evolution and Applications*, vol.2008, pp.1–10, 2008.
- [114] 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.
- [115] L. Zhang, F. Yang, and A. Elsherbeni, “An Improved PSO Method Using Gaussian Distributed Random Variables for Electromagnetic Optimization,” *Proceedings of IEEE Antennas and Propagation Society International Symposium*, pp.1–4, July 2008.
- [116] 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, June 2011.
- [117] J. Zhang, and A. Sanderson, “An Approximate Gaussian Model of Differential Evolution with Spherical Fitness Functions,” *Proceedings of the Congress on Evolutionary Computation 2007*, pp.2220–2228, 2007.
- [118] E. Jaynes, “Information Theory and Statistical Mechanics,” *Physical review*, vol.106, no.4, pp.620–630, 1957.
- [119] M. Milano, and A. Roli, “MAGMA: A Multiagent Architecture for Metaheuristics,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.34, no.2, pp.925–41, 2004.
- [120] F. Fernandes, S. Souza, M. Silva, H. Borges, and F. Ribeiro, “A Multiagent Architecture for Solving Combinatorial Optimization Problems through Metaheuristics,” *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics 2009*, pp.3071–3076, 2009.
- [121] A. Land, and A. Doig, “An Automatic Method of Solving Discrete Programming Problems,” *Econometrica: Journal of the Econometric Society*, vol.28, no.3, pp.497–520, 1960.
- [122] J. Kenney, and E. Keeping, *Mathematics of Statistics, part 2*, D. Van Nostrand Company Inc. 2nd edition, 1962.
- [123] J. Aitchison, and J. Brown, *The Lognormal Distribution*, Cambridge University Press, 1957.
- [124] J. Liang, P. Suganthan, and K. Deb, “Novel Composition Test Functions for Numerical Global Optimization,” *Proceedings of IEEE Symposium on Swarm Intelligence 2005*, pp.68–75, 2005.
- [125] H.H. Rosenbrock, “An Automatic Method for Finding the Greatest or Least Value of a Function,” *The Computer Journal*, vol.3, pp.175–184, 1960.

- [126] A.I. Oyman, and H.G. Beyer, “Analysis of the $(1, \lambda)$ -ES on the Parabolic Ridge,” *Evolutionary Computation*, vol.8, no.3, pp.267–289, 2000.
- [127] L. Rastrigin, *Systems of Extremal Control*, Nauka, 1974.
- [128] K.D. Boese, “Cost Versus Distance in the Traveling Salesman Problem 1 Introduction,” Tech. Rep. TR-950018, UCLA CS Department, 1995.

付 録 A 確率変数・確率分布に関する諸性質

A.1 独立同一分布にしたがう確率変数の積の期待値と分散

補題 A.1 $X^{(1)}, X^{(2)}, \dots, X^{(P)} \in \mathbb{R}$ を, 期待値 μ , 分散 σ^2 の同一の確率分布にしたがう独立な確率変数とする. このとき, $X^{(1)}, X^{(2)}, \dots, X^{(P)}$ の積 $Z^{(P)} = \prod_{p=1}^P X^{(p)}$ の期待値 $E[Z^{(P)}]$ と分散 $V[Z^{(P)}]$ は, それぞれ

$$E[Z^{(P)}] = \mu^P \quad (\text{A.1a})$$

$$V[Z^{(P)}] = (\mu^2 + \sigma^2)^P - \mu^{2P} \quad (\text{A.1b})$$

で与えられる.

証明 独立な確率変数 X, Y の積 XY に対する期待値と分散の性質

$$E[XY] = E[X] E[Y] \quad (\text{A.2a})$$

$$V[XY] = V[X] V[Y] + V[X] E^2[Y] + V[Y] E^2[X] \quad (\text{A.2b})$$

を用いる,

$$Z^{(P)} = X^{(P)} Z^{(P-1)} \quad (\text{A.3})$$

であることから, (A.1a) 式および (A.1b) 式より,

$$E[Z^{(P)}] = E[X^{(P)}] E[Z^{(P-1)}] = \mu E[Z^{(P-1)}] \quad (\text{A.4a})$$

$$\begin{aligned} V[Z^{(P)}] &= V[X^{(P)}] V[Z^{(P-1)}] + V[X^{(P)}] E^2[Z^{(P-1)}] + V[Z^{(P-1)}] E^2[X^{(P)}] \\ &= \sigma^2 V[Z^{(P-1)}] + \sigma^2 E^2[Z^{(P-1)}] + V[Z^{(P-1)}] \mu^2 \end{aligned} \quad (\text{A.4b})$$

が成立する. (A.4a) 式および (A.4b) 式を行列を用いて

$$\begin{pmatrix} E^2[Z^{(P)}] \\ V[Z^{(P)}] \end{pmatrix} = \begin{pmatrix} \mu^2 & 0 \\ \sigma^2 & \mu^2 + \sigma^2 \end{pmatrix} \begin{pmatrix} E^2[Z^{(P-1)}] \\ V[Z^{(P-1)}] \end{pmatrix} \quad (\text{A.5})$$

と整理し,

$$E[Z^{(1)}] = E[X^{(1)}] = \mu \quad (\text{A.6a})$$

$$V[Z^{(1)}] = V[X^{(1)}] = \sigma^2 \quad (\text{A.6b})$$

に注意して $E^2[Z^{(P)}], V[Z^{(P)}]$ 一般解を求めると,

$$\begin{aligned} \begin{pmatrix} E^2[Z^{(P)}] \\ V[Z^{(P)}] \end{pmatrix} &= \begin{pmatrix} \mu^2 & 0 \\ \sigma^2 & \mu^2 + \sigma^2 \end{pmatrix}^{P-1} \begin{pmatrix} E^2[Z^{(1)}] \\ V[Z^{(1)}] \end{pmatrix} \\ &= \begin{pmatrix} \mu^{2(P-1)} & 0 \\ (\mu^2 + \sigma^2)^{P-1} - \mu^{2(P-1)} & (\mu^2 + \sigma^2)^{P-1} \end{pmatrix} \begin{pmatrix} \mu^2 \\ \sigma^2 \end{pmatrix} \\ &= \begin{pmatrix} \mu^{2P} \\ (\mu^2 + \sigma^2)^P - \mu^{2P} \end{pmatrix} \end{aligned} \quad (\text{A.7})$$

となり, これより (A.1a) 式および (A.1b) 式が得られる. $\square$

A.2 独立同一分布にしたがう確率変数の標本分散・不偏分散の期待値と分散

補題 A.2 $X^{(1)}, X^{(2)}, \dots, X^{(P)} \in \mathbf{R}$ を, 分散 σ^2 , 尖度 β_2 の同一の確率分布にしたがう独立な確率変数とする. このとき, $X^{(1)}, X^{(2)}, \dots, X^{(P)}$ の標本分散

$$S^2 = \frac{1}{P} \sum_{p=1}^P \left(X^{(p)} - \frac{1}{P} \sum_{p=1}^P X^{(p)} \right)^2 \quad (\text{A.8})$$

の期待値 $E[S^2]$, 分散 $V[S^2]$ はそれぞれ

$$E[S^2] = \frac{P-1}{P} \sigma^2 \quad (\text{A.9a})$$

$$V[S^2] = \frac{\sigma^4}{P^3} ((P-1)^2 \beta_2 - (P-1)(P-3)) \quad (\text{A.9b})$$

で与えられる. また, 不偏分散

$$s^2 = \frac{1}{P-1} \sum_{p=1}^P \left(X^{(p)} - \frac{1}{P} \sum_{p=1}^P X^{(p)} \right)^2 \quad (\text{A.10})$$

の期待値 $E[s^2]$, 分散 $V[s^2]$ はそれぞれ

$$E[s^2] = \sigma^2 \quad (\text{A.11a})$$

$$V[s^2] = \frac{\sigma^4}{P(P-1)} ((P-1)\beta_2 - (P-3)) \quad (\text{A.11b})$$

で与えられる ^[122].

証明 まず, 標本分散の期待値および分散を計算する. 標本分散の期待値 $E[S^2]$ は, 標

本分散の表式 (A.8) を直接書き下し、確率変数の 2 次項と双 1 次項を区別して整理すると、

$$\begin{aligned}
\mathbb{E}[S^2] &= \mathbb{E}\left[\frac{1}{P}\sum_{p=1}^P\left(X^{(p)} - \frac{1}{P}\sum_{p=1}^P X^{(p)}\right)^2\right] \\
&= \mathbb{E}\left[\frac{1}{P}\sum_{p=1}^P\left(X^{(p)}\right)^2 - \left(\frac{1}{P}\sum_{p=1}^P X^{(p)}\right)^2\right] \\
&= \mathbb{E}\left[\frac{1}{P}\sum_{p=1}^P\left(X^{(p)}\right)^2 - \frac{1}{P^2}\left(\sum_{p=1}^P\left(X^{(p)}\right)^2 + \sum_{p \neq q}^P X^{(p)} X^{(q)}\right)\right] \\
&= \frac{1}{P}\mathbb{E}\left[\sum_{p=1}^P\left(X^{(p)}\right)^2\right] - \frac{1}{P^2}\mathbb{E}\left[\sum_{p=1}^P\left(X^{(p)}\right)^2\right] - \frac{1}{P^2}\mathbb{E}\left[\sum_{p \neq q}^P X^{(p)} X^{(q)}\right] \\
&= \frac{(P-1)}{P}\mu_2 - \frac{(P-1)}{P}\mu_1 \\
&= \frac{(P-1)}{P}\sigma^2
\end{aligned} \tag{A.12}$$

となり、(A.9a) 式が得られる。

標本分散の分散 $\mathbb{V}[S^2]$ は、 $Y^{(p)} = X^{(p)} - \mu$ とおくと、

$$\begin{aligned}
\mathbb{V}[S^2] &= \mathbb{E}[S^4] - \mathbb{E}^2[S^2] \\
&= \mathbb{E}\left[\left(\frac{1}{P}\sum_{p=1}^P\left(X^{(p)}\right)^2 - \left(\frac{1}{P}\sum_{p=1}^P X^{(p)}\right)^2\right)^2\right] - \frac{(P-1)^2}{P^2}\sigma^4 \\
&= \mathbb{E}\left[\left(\frac{1}{P}\sum_{p=1}^P\left(Y^{(p)}\right)^2 - \left(\frac{1}{P}\sum_{p=1}^P Y^{(p)}\right)^2\right)^2\right] - \frac{(P-1)^2}{P^2}\sigma^4
\end{aligned} \tag{A.13}$$

であり、また $\mathbb{E}[Y^{(p)}] = 0, \mathbb{E}[(Y^{(p)})^2] = \sigma^2, \mathbb{E}[(Y^{(p)})^4] = \beta_2\sigma^4$ に注意して右辺第 1 項を展開すると、

$$\begin{aligned}
\mathbb{V}[S^2] &= \mathbb{E}\left[\left(\frac{1}{P}\sum_{p=1}^P\left(Y^{(p)}\right)^2 - \left(\frac{1}{P}\sum_{p=1}^P Y^{(p)}\right)^2\right)^2\right] \\
&= \mathbb{E}\left[\frac{1}{P^2}\left(\sum_{p=1}^P\left(Y^{(p)}\right)^2\right)^2 - \frac{2}{P^3}\sum_{p=1}^P\left(Y^{(p)}\right)^2\left(\sum_{p=1}^P Y^{(p)}\right)^2 + \frac{1}{P^4}\left(\sum_{p=1}^P Y^{(p)}\right)^4\right] \\
&= \frac{1}{P^2}\mathbb{E}\left[\left(\sum_{p=1}^P\left(Y^{(p)}\right)^2\right)^2\right] - \frac{2}{P^3}\mathbb{E}\left[\sum_{p=1}^P\left(Y^{(p)}\right)^2\left(\sum_{p=1}^P Y^{(p)}\right)^2\right] \\
&\quad + \frac{1}{P^4}\mathbb{E}\left[\left(\sum_{p=1}^P Y^{(p)}\right)^4\right]
\end{aligned} \tag{A.14}$$

を得る．ここで

$$A = \mathbb{E} \left[\left(\sum_{p=1}^P \left(Y^{(p)} \right)^2 \right)^2 \right] \quad (\text{A.15a})$$

$$B = \mathbb{E} \left[\sum_{p=1}^P \left(Y^{(p)} \right)^2 \left(\sum_{p=1}^P Y^{(p)} \right)^2 \right] \quad (\text{A.15b})$$

$$C = \mathbb{E} \left[\left(\sum_{p=1}^P Y^{(p)} \right)^4 \right] \quad (\text{A.15c})$$

とおくと，それぞれ

$$\begin{aligned} A &= \mathbb{E} \left[\left(\sum_{p=1}^P \left(Y^{(p)} \right)^2 \right)^2 \right] \\ &= \mathbb{E} \left[\sum_{p=1}^P \left(Y^{(p)} \right)^4 + \sum_{p \neq q} \left(Y^{(p)} \right)^2 \left(Y^{(q)} \right)^2 \right] \\ &= \mathbb{E} \left[\sum_{p=1}^P \left(Y^{(p)} \right)^4 \right] + \mathbb{E} \left[\sum_{p \neq q} \left(Y^{(p)} \right)^2 \left(Y^{(q)} \right)^2 \right] \\ &= P\beta_2\sigma^4 + P(P-1)\sigma^4 \\ &= P(\beta_2 + P-1)\sigma^4 \end{aligned} \quad (\text{A.16a})$$

$$\begin{aligned} B &= \mathbb{E} \left[\sum_{p=1}^P \left(Y^{(p)} \right)^2 \left(\sum_{p=1}^P Y^{(p)} \right)^2 \right] \\ &= \mathbb{E} \left[\sum_{p=1}^P \left(Y^{(p)} \right)^4 + 2 \sum_{p \neq q} \left(Y^{(p)} \right)^3 Y^{(q)} \right. \\ &\quad \left. + \sum_{p \neq q} \left(Y^{(p)} \right)^2 \left(Y^{(q)} \right)^2 + \sum_{p \neq q \neq l} \left(Y^{(p)} \right)^2 Y^{(q)} Y^{(l)} \right] \\ &= \mathbb{E} \left[\sum_{p=1}^P \left(Y^{(p)} \right)^4 \right] + 2\mathbb{E} \left[\sum_{p \neq q} \left(Y^{(p)} \right)^3 Y^{(q)} \right] \\ &\quad + \mathbb{E} \left[\sum_{p \neq q} \left(Y^{(p)} \right)^2 \left(Y^{(p)} \right)^2 \right] \mathbb{E} \left[\sum_{p \neq q \neq l} \left(Y^{(p)} \right)^2 Y^{(q)} Y^{(l)} \right] \end{aligned} \quad (\text{A.16b})$$

$$\begin{aligned} &= P\beta_2\sigma^4 + P(P-1)\sigma^4 \\ &= P(\beta_2 + P-1)\sigma^4 \end{aligned} \quad (\text{A.16c})$$

$$C = \mathbb{E} \left[\left(\sum_{p=1}^P Y^{(p)} \right)^4 \right]$$

$$\begin{aligned}
&= \mathbb{E} \left[\sum_{p=1}^P \left(Y^{(p)} \right)^4 + 4 \sum_{p \neq q} \left(Y^{(p)} \right)^3 Y^{(q)} + 3 \sum_{p \neq q} \left(Y^{(p)} \right)^2 \left(Y^{(q)} \right)^2 \right. \\
&\quad \left. + 6 \sum_{p \neq q \neq l} \left(Y^{(p)} \right)^2 Y^{(q)} Y^{(l)} + \sum_{p \neq q \neq l \neq m} Y^{(p)} Y^{(q)} Y^{(l)} Y^{(m)} \right] \\
&= \mathbb{E} \left[\sum_{p=1}^P \left(Y^{(p)} \right)^4 \right] + 4 \mathbb{E} \left[\sum_{p \neq q} \left(Y^{(p)} \right)^3 Y^{(q)} \right] + 3 \mathbb{E} \left[\sum_{p \neq q} \left(Y^{(p)} \right)^2 \left(Y^{(q)} \right)^2 \right] \\
&\quad + 6 \mathbb{E} \left[\sum_{p \neq q \neq l} \left(Y^{(p)} \right)^2 Y^{(q)} Y^{(l)} \right] + \mathbb{E} \left[\sum_{p \neq q \neq l \neq m} Y^{(p)} Y^{(q)} Y^{(l)} Y^{(m)} \right] \\
&= P \beta_2 \sigma^4 + 3P(P-1) \sigma^4 \\
&= P(\beta_2 + 3P - 3) \sigma^4 \tag{A.16d}
\end{aligned}$$

とそれぞれ変形できる．(A.14) 式に (A.16) 式で得られた結果を代入すると，

$$\begin{aligned}
\mathbb{V} [S^2] &= \frac{1}{P^2} A - \frac{2}{P^3} B + \frac{1}{P^4} C \\
&= \frac{\sigma^4}{P^3} ((P-1)^2 \beta_2 - (P-1)(P-3)) \tag{A.17}
\end{aligned}$$

となり，(A.9b) 式が得られる．

不偏分散の期待値と分散については，不偏分散と標本分散の関係

$$s^2 = \frac{P}{P-1} S^2 \tag{A.18}$$

より，

$$\mathbb{E} [s^2] = \mathbb{E} \left[\frac{P}{P-1} S^2 \right] = \sigma^2 \tag{A.19a}$$

$$\mathbb{V} [s^2] = \mathbb{V} \left[\frac{P}{P-1} S^2 \right] = \frac{\sigma^4}{P(P-1)} ((P-1)\beta_2 - (P-3)) \tag{A.19b}$$

となり，(A.11) 式が得られる． $\square$

A.3 対数正規分布の期待値と分散

補題 A.3 対数正規分布の確率密度関数は

$$f_{\mathcal{LN}}(x; \hat{\mu}, \hat{\sigma}^2) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}x} \exp\left(-\frac{(\log(x) - \hat{\mu})^2}{2\hat{\sigma}^2}\right) \quad (\text{A.20})$$

で定義される．対数正規分布の期待値 $\mu_{\mathcal{LN}}$ および分散 $\sigma_{\mathcal{LN}}^2$ は，それぞれ

$$\mu_{\mathcal{LN}} = \exp\left(\hat{\mu} + \frac{1}{2}\hat{\sigma}^2\right) \quad (\text{A.21a})$$

$$\sigma_{\mathcal{LN}}^2 = \exp(2\hat{\mu} + \hat{\sigma}^2) (\exp(\hat{\sigma}^2) - 1) \quad (\text{A.21b})$$

で与えられる [123]．

証明 対数正規分布の確率密度関数 (A.20) の 1 次，2 次のモーメント $\mu_{1\mathcal{LN}}, \mu_{2\mathcal{LN}}$ は，それぞれ

$$\begin{aligned} \mu_{1\mathcal{LN}} &= \int_0^\infty x f_{\mathcal{LN}}(x; \hat{\mu}, \hat{\sigma}^2) dx \\ &= \int_0^\infty x \left(\frac{1}{\sqrt{2\pi\hat{\sigma}^2}x} \exp\left(-\frac{(\log(x) - \hat{\mu})^2}{2\hat{\sigma}^2}\right) \right) dx \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp\left(-\frac{(\log(x) - \hat{\mu})^2}{2\hat{\sigma}^2}\right) dx \end{aligned} \quad (\text{A.22a})$$

$$\begin{aligned} \mu_{2\mathcal{LN}} &= \int_0^\infty x^2 f_{\mathcal{LN}}(x; \hat{\mu}, \hat{\sigma}^2) dx \\ &= \int_0^\infty x^2 \left(\frac{1}{\sqrt{2\pi\hat{\sigma}^2}x} \exp\left(-\frac{(\log(x) - \hat{\mu})^2}{2\hat{\sigma}^2}\right) \right) dx \\ &= \int_0^\infty \frac{x}{\sqrt{2\pi\hat{\sigma}^2}} \exp\left(-\frac{(\log(x) - \hat{\mu})^2}{2\hat{\sigma}^2}\right) dx \end{aligned} \quad (\text{A.22b})$$

で与えられる．(A.22) 式は， $\log(x) = t$ と置換することにより，

$$\begin{aligned} \mu_{1\mathcal{LN}} &= \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp(t) \exp\left(-\frac{(t - \hat{\mu})^2}{2\hat{\sigma}^2}\right) dt \\ &= \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp\left(-\frac{(t - \hat{\mu} - \hat{\sigma}^2)^2}{2\hat{\sigma}^2} + \hat{\mu} + \frac{1}{2}\hat{\sigma}^2\right) dt \\ &= \exp\left(\hat{\mu} + \frac{1}{2}\hat{\sigma}^2\right) \end{aligned} \quad (\text{A.23a})$$

$$\begin{aligned} \mu_{2\mathcal{LN}} &= \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp(2t) \exp\left(-\frac{(t - \hat{\mu})^2}{2\hat{\sigma}^2}\right) dt \\ &= \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp\left(-\frac{(t - \hat{\mu} - 2\hat{\sigma}^2)^2}{2\hat{\sigma}^2} + 2\hat{\mu} + 2\hat{\sigma}^2\right) dt \\ &= \exp(2\hat{\mu} + 2\hat{\sigma}^2) \end{aligned} \quad (\text{A.23b})$$

と計算できる．対数正規分布の平均 $\mu_{\mathcal{LN}}$ および分散 $\sigma_{\mathcal{LN}}^2$ は，それぞれの定義より，

$$\mu_{\mathcal{LN}} = \mu_{1\mathcal{LN}} = \exp\left(\hat{\mu} + \frac{1}{2}\hat{\sigma}^2\right) \quad (\text{A.24a})$$

$$\sigma_{\mathcal{LN}}^2 = \mu_{2\mathcal{LN}} - \mu_{1\mathcal{LN}}^2 = \exp\left(2\hat{\mu} + \hat{\sigma}^2\right) \left(\exp\left(\hat{\sigma}^2\right) - 1\right) \quad (\text{A.24b})$$

で与えられ，(A.21) 式が得られる．

□

付 録 B 最適化アルゴリズムの性能評価とベンチマーク問題

B.1 座標軸を回転させたベンチマーク問題の作成

従来用いられてきた最適化ベンチマーク問題のなかには、変数間依存性をもたない独立変数を有する目的関数のもの多い。こういった問題に対しては、成分ごとに独立したダイナミクスをもつ手法が有効であること、またそれゆえこういった構造の目的関数がベンチマーク問題として不適であるという指摘がある^[124]。そこで本論文では、元ベンチマーク問題に加えて、文献[54]にならい、探索空間の座標系を、後述する回転行列 $R(\alpha)$ により回転させることにより、上述の問題を改善した問題もあわせて採用する。

回転行列 $R(\alpha)$ はつぎのように計算される。

$$R(\alpha) = R^{1,2}(\alpha)R^{1,3}(\alpha) \cdots R^{1,N}(\alpha)R^{2,3}(\alpha)R^{2,N}(\alpha) \cdots R^{N-2,N}(\alpha)R^{N-1,N}(\alpha) \quad (\text{B.1a})$$

$$r_{kl}^{i,j}(\alpha) = \begin{cases} \cos(\alpha), & k = i, l = i \\ -\sin(\alpha), & k = i, l = j \\ \sin(\alpha), & k = j, l = i \\ 1, & k = l \neq i, k = l \neq j \\ 0, & \text{else} \end{cases} \quad (\text{B.1b})$$

オリジナルの目的関数 $f(\mathbf{x})$ とその大域的最適解 $\mathbf{x}^*$ に対し、回転行列 $R(\alpha)$ を施した新しい問題はつぎのように設計される。

$$\begin{cases} \min_{\mathbf{z}(\mathbf{x})} f(\mathbf{z}(\mathbf{x})) \\ \text{subj. to } a_n \leq x_n \leq b_n, \quad n = 1, 2, \dots, N \\ \text{where } \mathbf{z}(\mathbf{x}) = R(\alpha)(\mathbf{x} - \mathbf{x}^*) + \mathbf{x}^* \end{cases} \quad (\text{B.2})$$

B.2 ベンチマーク問題とその特徴

Prob. 1 Sphere Function

$$\begin{cases} \min_{\mathbf{x}} f(\mathbf{x}) = \sum_{n=1}^N x_n^2 \\ \text{subj. to } -5.0 \leq x_n \leq 5.0, \quad n = 1, 2, \dots, N \\ \text{where } \mathbf{x}^* = \mathbf{0.0}, \quad f(\mathbf{x}^*) = 0.0 \end{cases} \quad (\text{B.3})$$

Prob. 1 は、単純な凸 2 次関数であり、唯一の大域的最適解をもつ。最適化アルゴリズムの基本的な最適化性能の確認のために用いられる。Fig. B.1 に、 $N = 2$ の場合の目的関数の概形を示す。

Prob. 2 Rosenbrock's Function^[125]

$$\begin{cases} \min_{\mathbf{x}} f(\mathbf{x}) = \sum_{n=1}^{N-1} \left((1.0 - x_n)^2 + 100.0 (x_{n+1} - x_n^2)^2 \right) \\ \text{subj. to } -5.0 \leq x_n \leq 5.0, \quad n = 1, 2, \dots, N \\ \text{where } \mathbf{x}^* = \mathbf{1.0}, \quad f(\mathbf{x}^*) = 0.0 \end{cases} \quad (\text{B.4})$$

Prob. 2 は、最適解が湾曲した谷のなかに存在するいわゆる綾構造^[126]の問題であり、最適解を得るためには有効な局所的最適化戦略が必要である。勾配を用いた最適化アルゴリズムを用いた場合でも、前述の関数形状のため直線探索のステップ幅が小さく定まりやすく、最適解を得るまでに多数の計算の反復が必要となる。Fig. B.2 に、 $N = 2$ の場合の目的関数の概形を示す。

Prob. 3 (Rotated) 2^N minima Function^[29]

$$\begin{cases} \min_{\mathbf{z}(\mathbf{x})} f(\mathbf{z}(\mathbf{x})) = \sum_{n=1}^N (z_n^4(\mathbf{x}) - 16z_n^2(\mathbf{x}) - 5z_n(\mathbf{x})) \\ \text{subj. to } -5.0 \leq x_n \leq 5.0, \quad n = 1, 2, \dots, N \\ \text{where } \mathbf{z}(\mathbf{x}) = R(\alpha)(\mathbf{x} - \mathbf{x}^*) + \mathbf{x}^*, \quad \mathbf{x}^* = -\mathbf{2.90353}, \quad f(\mathbf{z}(\mathbf{x}^*)) = -78.33233N \end{cases} \quad (\text{B.5})$$

Prob. 3 は、 2^N minima Function の名の示すとおり、変数次元 N に対し 2^N 個の局所的最適解をもつ問題であり、基本的な大域的最適化性能のテスト関数として用いられる。ただし、各変数はそれぞれ依存性を持たないため、変数ごとに問題を分割して解くような手法できわめて容易に解ける。そこで本論文では、B.1 節で述べた回転行列 $R(\alpha)$ を用いて座標系を回転された問題を合わせて用いる。Fig. B.3 に、 $N = 2$ の場合の目的関数の概形を示す。

Prob. 4 (Rotated) Rastrigin's Function^[127]

$$\begin{cases} \min_{\mathbf{z}(\mathbf{x})} f(\mathbf{z}(\mathbf{x})) = \sum_{n=1}^N (z_n^2(\mathbf{x}) - 10.0 \cos(2\pi z_n(\mathbf{x})) + 10.0) \\ \text{subj. to } -5.0 \leq x_n \leq 5.0, \quad n = 1, 2, \dots, N \\ \text{where } \mathbf{z}(\mathbf{x}) = R(\alpha)(\mathbf{x} - \mathbf{x}^*) + \mathbf{x}^*, \quad \mathbf{x}^* = \mathbf{0.0}, \quad f(\mathbf{z}(\mathbf{x}^*)) = 0.0 \end{cases} \quad (\text{B.6})$$

Prob. 4 は、多数の局所的最適解が大谷構造^[128]のなかに配置された問題であり、「すでに得た最良解を中心に周辺を探索する」手法と相性が良い。しかしながら、変数間依存性を持たないため、ベンチマーク問題として Prob. 3 と同様の問題をもつ。本論文では、Prob. 3 と同様、回転行列 $R(\alpha)$ を用いて座標系を回転された問題を併せて用いる。Fig. B.4 に、 $N = 2$ の場合の目的関数の概形を示す。

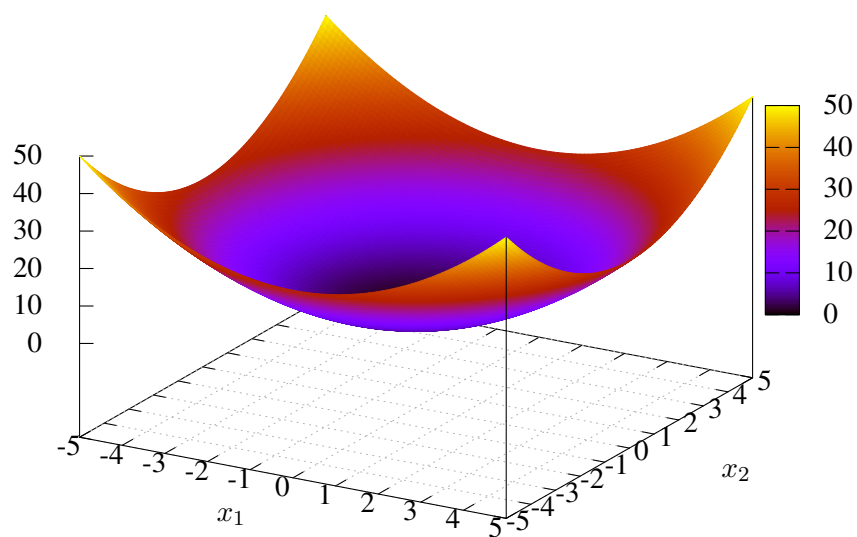


Fig. B.1 Landscape of sphere function.

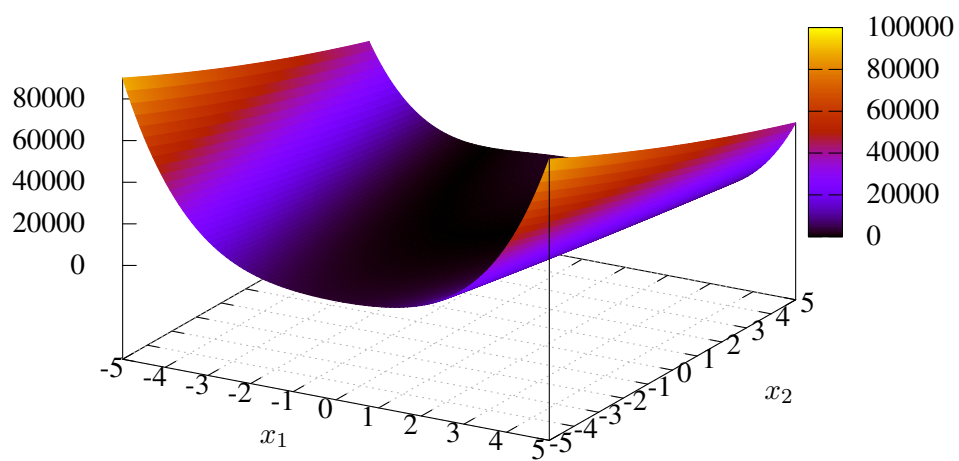


Fig. B.2 Landscape of Rosenbrock's Function.

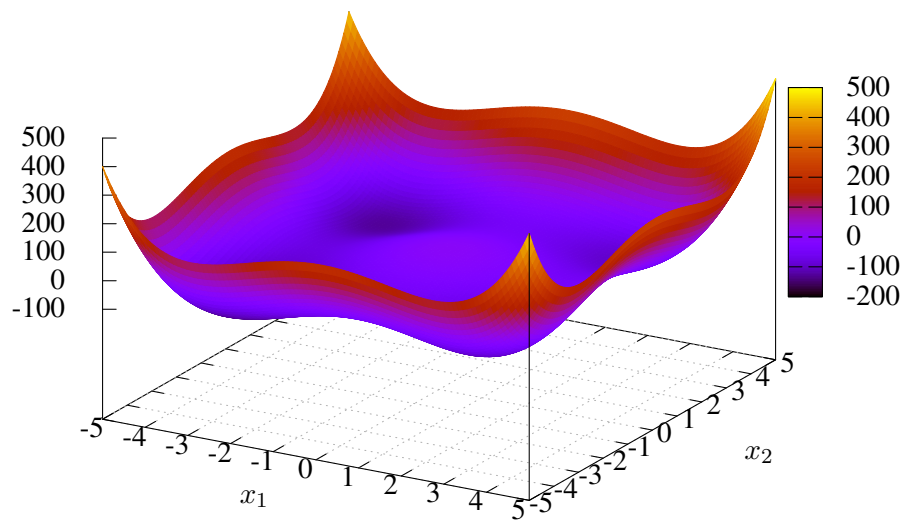


Fig. B.3 Landscape of 2^N minima Function.

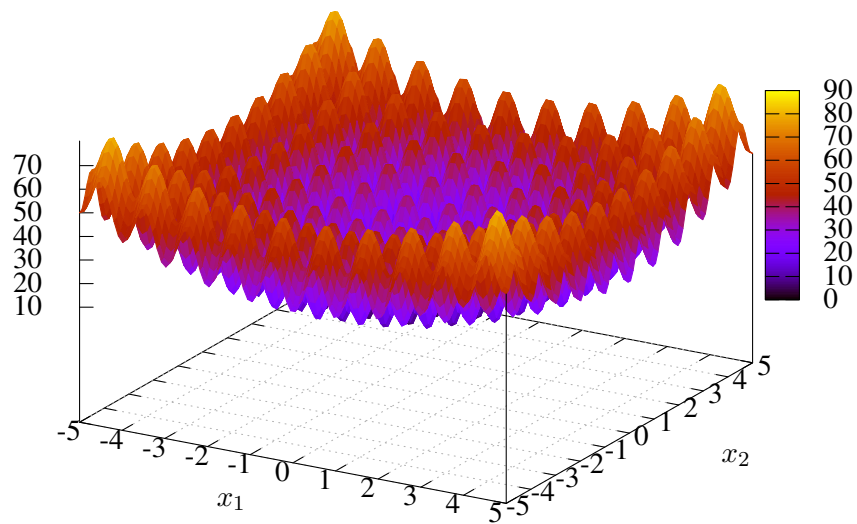


Fig. B.4 Landscape of Rastrigin's Function.

付録C PSOの安定性指標の数値計算法

3.3.2 項で定義した, PSO と等価な安定性を有するシステム (3.5) における偏角の密度分布

$$\rho(\theta) = \int_0^1 dR_1 \int_0^1 dR_2 \int_0^{2\pi} \delta(\theta - \arg(M(R_1, R_2; \lambda, c_1, c_2)\mathbf{q}(\theta')))\rho(\theta')d\theta' \quad (\text{C.1})$$

および安定性指標

$$\zeta(\lambda, c_1, c_2) = \exp \left(\int_0^1 dR_1 \int_0^1 dR_2 \int_0^{2\pi} \log \|M(R_1, R_2; \lambda, c_1, c_2)\mathbf{q}(\theta)\| \rho(\theta) d\theta \right) \quad (\text{C.2})$$

の数値的計算手法を与える.

3.3.2 項の議論どおり, 偏角の密度分布 $\rho(\theta)$ は定常に達していると仮定する. このとき, ある連続する探索時刻間で偏角が θ' から θ に遷移する確率を $g(\theta, \theta')$ として,

$$\rho(\theta) = \int_0^{2\pi} g(\theta, \theta') \rho(\theta') d\theta' \quad (\text{C.3})$$

が成り立つ. $g(\theta, \theta')$ は, 行列 $M(R_1, R_2; \lambda, c_1, c_2)$ 内のパラメータ λ, c_1, c_2 によって定まるが, $g(\theta, \theta')$ を解析的な手段により決定するのは困難である. そこで本論文では, $g(\theta, \theta')$ を離散化して数値的に計算する方法を考える. いま, θ の値域 $[0, 2\pi)$ を離散化して N_θ 個の領域に均等に分割し, 偏角 θ の確率分布 $\rho(\theta)$ を, N_θ 次元のベクトルに対応させ, $\boldsymbol{\rho} = (\rho_1, \dots, \rho_{N_\theta})^\top$ と書けば, (C.3) 式を離散化した式として,

$$\boldsymbol{\rho} = \mathbf{G}\boldsymbol{\rho} \quad (\text{C.4})$$

が得られる. ただし, 行列 $\mathbf{G} \in \mathbb{R}^{N_\theta \times N_\theta}$ の要素 g_{mn} は, ある連続する探索時刻間で偏角が $2\pi n/N_\theta$ から半閉区間 $\left[\frac{2\pi m}{N_\theta}, \frac{2\pi[(m+1) \bmod (N_\theta)]}{N_\theta} \right)$ に遷移する確率を表す. $\mathbf{G}$ は, θ と同様に乱数 R_1, R_2 の値域 $[0, 1]$ を離散化し, それぞれの分割数を $L_1 + 1, L_2 + 1$ とすれば,

$$g_{mn} = \frac{1}{L_1 L_2 N_\theta} \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{L_2} \sum_{n=0}^{N_\theta-1} \delta \left(m - \left\lfloor \frac{1}{2\pi} \arg \left(M \left(\frac{l_1}{L_1}, \frac{l_2}{L_2}; \lambda, c_1, c_2 \right) \mathbf{q} \left(2\pi \frac{n}{N_\theta} \right) \right) \right\rfloor \right) \quad (\text{C.5})$$

で数値的に計算できる. (C.5) 式で求めた $\mathbf{G}$ を用いて, (C.2) 式を $\boldsymbol{\rho}$ について解けば, 離散化された偏角 θ の密度分布が得られる.

また, 安定性指標 ζ は, 上で求めた $\boldsymbol{\rho}$ を用いて,

$$\zeta(\lambda, c_1, c_2) = \exp \left(\frac{1}{L_1 L_2} \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{L_2} \sum_{n=1}^{N_\theta-1} \log \left\| M \left(\frac{l_1}{L_1}, \frac{l_2}{L_2}; \lambda, c_1, c_2 \right) \mathbf{q} \left(2\pi \frac{n}{N_\theta} \right) \right\| \rho_n \right) \quad (\text{C.6})$$

で数値的に計算できる.

付 録D 数値実験環境

Table D.1 に，本論文の数値実験環境を示す。

Table D.1 Numerical calculation environment.

Item	Value
Computer	Apple Mac Pro (Early 2009)
Operating System	Mac OS X 10.9.5
CPU	Quad-Core Intel Xeon 2.66GHz
RAM	32GB 1066MHz DDR3
C/C++ Compiler	GNU Compiler Collection (GCC) 4.8.2 (with OpenMP) ¹
Library	Eigen3 (Linear Algebra Library) ²

¹<https://gcc.gnu.org>

²<http://eigen.tuxfamily.org>