

学位論文 博士（工学）

メタヒューリスティクスに対する
遺伝的プログラミングによる
創発的パラメータ調整則の自動設計

2015 年度

慶應義塾大学大学院理工学研究科

金 政 実

目次

第 1 章	はじめに	1
1.1	メタヒューリスティクスの登場とその課題	1
1.2	メタヒューリスティクスの課題解決と本論文の目的	3
1.3	本論文の構成	6
1.4	本論文が対象とする最適化問題とそこで用いられる記号	6
第 2 章	代表的なメタヒューリスティクスとそのパラメータ調整法	9
2.1	本章の内容	9
2.2	Particle Swarm Optimization の原理とそのパラメータ調整の課題	9
2.2.1	Particle Swarm Optimization の原理	9
2.2.2	Particle Swarm Optimization のパラメータ調整法	11
2.2.3	計算機実験による PSO のパラメータ調整法の課題提起	15
2.3	Evolution Strategy の原理とそのパラメータ調整の課題	24
2.3.1	単点型 Evolution Strategy の原理とそのパラメータ調整則	24
2.3.2	多点型 Evolution Strategy の原理とそのパラメータ調整則	25
2.3.3	(μ, λ) -ES	25
2.3.4	CMA-ES	27
2.3.5	計算機実験による ES のパラメータ調整法の課題提起	29
2.4	Differential Evolution の原理とそのパラメータ調整	34
2.4.1	Differential Evolution の原理	34
2.4.2	Differential Evolution のパラメータ調整法	35
2.4.3	計算機実験による DE のパラメータ調整法の課題提起	39
2.5	Firefly Algorithm の原理とそのパラメータ調整の課題	47
2.5.1	Firefly Algorithm の原理	47
2.5.2	改良型 Firefly Algorithm	48
2.5.3	計算機実験による FA のパラメータ調整法の課題提起	48
第 3 章	最適化アルゴリズムのパラメータ調整のためのメタ最適化	52
3.1	パラメータ調整法のメタ最適化	52
3.2	最適化アルゴリズムのパラメータ最適調整問題の定式化	53
3.2.1	最適化アルゴリズムの最適パラメータ決定問題とメタ最適化	53
3.3	メタヒューリスティクスのパラメータ最適調整問題とメタ最適化	56
3.4	メタ最適化問題の目的関数依存性と適合目的関数	61
3.5	メタヒューリスティクスのパラメータ調整変数の種類	65
3.6	複数の適合関数を考慮した多目的メタ最適化問題	68

3.7	遺伝的プログラミングによるメタ最適化問題の解法	70
第4章	メタヒューリスティクスとその調整則の獲得	75
4.1	メタヒューリスティクスのパラメータ調整法のメタ最適化	75
4.2	メタヒューリスティクスのパラメータ調整法の シミュレーション条件	76
4.3	Particle Swarm Optimization のパラメータ調整法のメタ最適化	80
4.3.1	Particle Swarm Optimization とその調整則の設計	80
4.3.2	Particle Swarm Optimization の調整則の進化結果	81
4.4	Evolution Strategy のパラメータ調整法のメタ最適化	87
4.4.1	多点型 Evolution Strategy とその調整則の設計	87
4.4.2	多点型 Evolution Strategy の調整則の進化結果	88
4.5	Differential Evolution のパラメータ調整法のメタ最適化	93
4.5.1	超球交叉を用いた Differential Evolution とその調整則の設計	93
4.5.2	Differential Evolution の調整則の進化結果	94
4.6	Firefly Algorithm のパラメータ調整法のメタ最適化	99
4.6.1	同期型 Firefly Algorithm とその調整則の設計	99
4.6.2	Firefly Algorithm の調整則の進化結果	100
4.7	新しい調整則を有するメタヒューリスティクスの総合比較	105
第5章	おわりに	110
付録A	最適化アルゴリズムの評価	120
A.1	一般的なベンチマーク問題	120
A.2	最適化アルゴリズム比較のための統計的検定	121
A.2.1	母集団の分布に対する仮定	124
A.2.2	片側検定	125
A.2.3	ランク化 t 検定	127
A.2.4	検定の多重性	129
A.2.5	まとめ	129
付録B	4章の詳細な結果	130
B.1	PSO の詳細な結果	130
B.2	ES の詳細な結果	135
B.3	DE の詳細な結果	140
B.4	FA の詳細な結果	145
B.5	横断的な比較の詳細な結果	150

第1章 はじめに

1.1 メタヒューリスティクスの登場とその課題

システム工学の重要な分野の一つである「システム最適化」における計算手法を分類する用語として、「メタヒューリスティクス」が登場し、とくに近年においてはそれに類する有力な計算手法が多く開発されて話題となっている。この用語が最初に登場したのは、1986年のGloverによる「タブーサーチ」の論文^[1]といわれている。そこでは、より良い解を求めて探索するという旧来の比較的単純なヒューリスティックな手法(日本語訳としては「発見的手法」という)に、直前に探索した点には遷移しないような上位のヒューリスティックである「タブー操作」を冠したことから、メタヒューリスティックという用語を使用した。一般的には、「メタ」は「高次の」という意味を表す接頭語であり、最初は「局所的探索のヒューリスティックに、局所最適解より離脱して大域的探索を行う上位のヒューリスティックを付与させた手法」をメタヒューリスティックな手法としていたが、時間とともに、「多点探索により局所最適解から離脱して大域的探索を行う手法」と変化し^[2]、近年においては「大域的探索能力を有するヒューリスティックな最適化手法」を「メタヒューリスティクス」と総称するようになった^{[3][4]}。

こうして局所最適解からの離脱による大域的最適解の探索機能が、タブーサーチにおいて強調されたこともあり、メタヒューリスティクスが有する機能の一つとして、「大域的最適解の探索機能」が強調されるようになった。最適化問題の目的関数に凸性の仮定がない場合、とくに多峰性の場合や探索空間が離散的な場合¹、その大域的最適解を厳密に求めることは、いわゆるNP困難な問題であり、計算を停止するための大域的最適解であるか否かの判定条件さえ与えることが原理的に困難である。そこで、限られた計算時間内であつ比較的高い確度で、大域的最適解の有力な近似解が得られる機能が、ヒューリスティックな最適化アルゴリズムに求められるようになった。

近年の定義で捉えられているメタヒューリスティクスが登場したのは1950~1960年代であり^{[5][6][7][8]}、こうしたアルゴリズムが積極的に評価される大きなきっかけとなったのが、「進化的アルゴリズム」と総称される遺伝的アルゴリズム(Genetic Algorithm: GA^[9])と進化戦略(Evolutional Strategy: ES^{[10][11][12]})の登場であるといえる。GAは1970年代から研究され始めていたが、1980年代にその大域的探索性能と有用性が、生物進化とのアナロジーが強調される形で、急速に認知されて世界的に話題となり、工学分野を幅広く横断する形で様々な設計問題に応用されて今日に至っている。また、ESも生物進化の過程を模倣したアルゴリズムとしてすでに1960年代に登場して1970年代に注目されるに至り、近年ではその探索点の多点化などによりその計算性能を高めるなど、急速に研究が進展することになった。たとえば、米国IEEEなどにおいて、Congress on Evolutionary Computation

¹ ただし近年においては離散凸解析という新しい理論体系が登場しており、離散的であっても大域的最適解が求まる目的関数が限定的ではあるが知られており、研究の対象となっている。

(CEC) と略称される国際会議が毎年開催され、我が国においてもこれらの分野に特化した「進化計算学会」などが創設されるに至っており、メタヒューリスティクスに属するアルゴリズムが有力な計算手法であるとしてその地位が確立している。

ところで、これら進化計算と称されるアルゴリズムは、本質的には最適化手法であり、メタヒューリスティクスの一つである。とくに GA の最適化手法としての斬新さが評価されたのは、探索空間の解候補として複数の探索点を用意し、それらを「交叉演算」と称する生物の遺伝操作を模擬した新しい演算を導入し、複数の解候補の間で情報交換する仕組みを最適化計算の過程に取り入れたことである。このような複数の解候補を用意し、これらを相互干渉させながら探索点の群れとして更新していくアルゴリズムの構造が、他の進化計算アルゴリズムを含めて多くのメタヒューリスティクスの特徴的な構造となり、この多点探索型構造によって大域的探索機能をもたせることが主流となっている。こうした流れの中で、本論文で取り上げる Particle Swarm Optimization (PSO)^[13] や Differential Evolution (DE)^[14] などの、実数値空間上の探索手法として有力なメタヒューリスティクスが 1990 年代後半に登場することとなった。こうしたメタヒューリスティクスでは、大域的最適解の探索性能が個々の探索点の挙動だけでなく、

- (1) 複数の探索点の群れによる目的関数の大域的景観の把握と、その情報を反映させた探索点群の更新

が、メタヒューリスティクスの解析や開発の鍵となっている。具体的には、探索点群の挙動を特徴付ける探索点群の「集中化 (Intensification) と分散化 (Diversification)」^[15] や、近接最適性の原理 (Proximate optimality principle: POP)^[16] という経験則がメタヒューリスティクスの解析と開発のキーになっており、こうした探索点の群れとしての特性に、生物の集団としての挙動の模倣や自然現象の模倣に基づいて各種各様の原理を導入し、様々なアルゴリズムが開発されている。

メタヒューリスティクスのもう一つの特徴は、確率的最適化手法として位置づけられることである。すなわち、GA や ES に代表されるように、それらの探索点の更新則に確率的要素が含まれ、探索点の更新が乱数を用いて確率的に決められることである。したがって、それらのアルゴリズムによる探索点が最適解に収束するとしても、その性能が確率的・統計的にしか保証されないことである。同時に、こうした確率的揺動によって生成された新しい解候補を直接目的関数で評価することで、その値に改善がなければその点を棄却したり、探索点の群れの中でより良い関数値の解候補だけを選択して残りは棄却したりするなど、探索点そのものの選択や棄却という数理的に扱うことが難しい操作を含ませているため、メタヒューリスティクスの解析を困難なものにしているといえる。しかも、こうした操作によるアルゴリズムの確率的・統計的探索性能は、それに含まれるパラメータに鋭敏に支配され、同時に前述した探索点群の「集中化と分散化」の特性を支配する結果となっている。したがって、

- (2) アルゴリズム中の確率的・統計的探索性能を支配するパラメータの調整

が、メタヒューリスティクスの解析や開発の大きな鍵となっている。

ところで、GA 系のアルゴリズムの発展過程で特筆すべきことは、1985 年に Cramer によって、探索空間の対象として 0-1 組合せベクトルの空間から木構造の空間へと概念が拡大され^[17]、1990 年に Koza によって「遺伝的プログラミング」(Genetic Programming:

GP)^[18]が登場したことである。GA系のアルゴリズムでは、「交叉演算」とよばれる探索点間の演算や、それに加えて探索点間での選択・棄却(GAでは「淘汰」という用語が使われている)という斬新な演算が用いられるとはいえ、通常のGAでの交叉演算は本質的に線形作用素であり、0-1組合せ空間での数値処理の一つに過ぎない。これに対してGPは、「木構造」で表されるグラフ空間上の解候補の探索を扱うようにしたため、数式や人工言語・自然言語だけでなく、回路や構造物そのものを直接探索するなど、最適化問題が扱う対象を数値的空間から記号的空間やグラフ的空間へと各段に広げることになった。したがってGPは、GAから派生したものではあるが、メタヒューリスティクスの中でも異質な存在として、他の数値处理的なメタヒューリスティクスからは区別して扱い、「メタ機能とは異次元の機能」、いわば「ハイパー機能」を有する手法として位置付けることができる。本論文では、GPがもつこのような記号处理的な特徴や機能を、数値处理的な機能しか有していない他のメタヒューリスティクスのさらなる上位機能として位置付けるものとする。なお、GPの性質上、応用問題は数式で表される問題ではなく、シミュレータの出力等を目的関数値とするBlack-Box Optimizationであるため計算時間の課題がその欠点としてあげられたが、1990年代中頃にBeowulf^[19]などのフリーのHPCクラスタ構築用のパッケージが登場し、特に2000年以降、HPCクラスタを用いてGPが様々な分野に応用されるに至り^{[20][21][22][23]}、現実に実用化されるに至っている^[24]。

近年においてはGoogleのMapReduce^[25]の論文をもとにオープンソースで実装された大規模分散処理フレームワークであるHadoop^[26]が登場し、これを用いたメタヒューリスティクスの研究も行われている^[27]。また、ビッグデータの普及とともに、大規模分散システムを用いた最適化が機械学習の分野において非常に重要となっているが、主に確率的勾配法に基づく最適化が使われており、この分野において分散処理に適しているメタヒューリスティクスの応用が期待される。

1.2 メタヒューリスティクスの課題解決と本論文の目的

複数の探索点の群れとしての特性を特徴付ける「独特の原理」に基づいて個々のメタヒューリスティクスが開発されてきており、それらアルゴリズムの更新手順は数理的根拠に基づいて導出されたものではない。そのため、その更新則の構造自体に汎用性が必ずしもなく、解きたい問題に対して適したものとは必ずしもいえないことも課題としてあげられる。また、それらの確率的・統計的特性が、アルゴリズム中のパラメータの値に大きく依存するが、その推奨値を与えるための理論的指針を与えることが容易ではない。したがって、解きたい問題ごとに最良の値を試行錯誤的にアルゴリズムのユーザが決める必要がある。また、複数の探索点の群れとしての挙動を、たとえば目的関数の大域的な景観把握のための分散的な状態から、集中状態に移行させて解の精度を高めるための局所的探索をさせたり、あるいは探索点が局所的最適解に停留しそうな安定状態になると、不安定化させて局所的最適解の引き込み領域から離脱させるなど、イテレーションの経過とともに探索点の挙動や様相を経時的に変化させることで、メタヒューリスティクスの性能を改善しようとする研究がなされている。

以上のようなメタヒューリスティクスの課題や研究状況を考慮し、本論文では、

1. 解きたい最適化問題ごとに特化する形で，メタヒューリスティクスの性能を向上させてより良い解を求めるためのパラメータの調整法
2. あるクラスに属する多くの最適化問題に対する汎化性をもたせたパラメータ調整法について考察する．このとき，パラメータの調整法を大別すると，

(A) フィードフォワード型調整法

(B) フィードバック型調整法

に分けることができる．(A)は，メタヒューリスティクスのアルゴリズム中のパラメータを時変係数とし，パラメータの値を直接時間の関数として与え，アルゴリズムのイテレーション(反復)の経過を時間とみなし，この経過とともにその関数にしたがってパラメータを変動させる方法である．これに対して(B)は，アルゴリズムの挙動特性を示す各種変数の関数としてアルゴリズム中のパラメータを規定し，本来のメタヒューリスティック手法のアルゴリズムのダイナミクスを対象としたフィードバック系を新たに構成してアルゴリズムを動作させる方法である．すなわち，(B)はアルゴリズムを動作させるパラメータを入力，それに対するアルゴリズムの特性を表す変数を出力とする対象に対し，後者を入力とし前者を出力とするいわば逆システムを付加してフィードバック系を構成することである．このことを「システム制御論」の観点から解釈すれば，固有の原理の下で作られたメタヒューリスティクスの構造による動特性そのものを，フィードバック系を構成することで構造的にその動特性を変更していると考えられ，こうすることによってたとえば「集中化と分散化」といった群れとして探索点の挙動も，その群れの状況に応じて自律的に設計することができる期待される．しかも，確率的・統計的な特性を有するメタヒューリスティクスの不確定的な挙動も，このフィードバック系で修正や改善をほどこすことで補償することができる期待される．以上のような観点から，本論文では一貫してメタヒューリスティクスに対して，システム制御論の観点から「フィードフォワード型パラメータ調整法」と「フィードバック型パラメータ調整法」を提案することを目的としている．特に，これらの調整法の部分を「パラメータ調整則」と称することにする．

ところで，上記のメタヒューリスティクスのパラメータ調整法に関する考察は古くからあり^[28]，その研究も多い^[29]．いずれにせよ，こうしたパラメータの調整方法の提案は，数多くのベンチマークに対するシミュレーションを通じた提案者の経験則やアイデアに基づくものであり，改良指針や設計指針も明確ではないなど，必ずしも科学的・合理的といえるものではない．本論文では，解きたい最適化問題とそれに適用する最適化手法のアルゴリズムが与えられたとき，特定の設計指針に関してアルゴリズム中の最適なパラメータを決定する問題を「メタ最適化問題」として定式化し，この問題にGPを適用することで，それがもつ記号処理のハイパー機能により，パラメータ調整則を規定する関数の生成をより合目的的に行う方法論を提唱する．具体的には，

- (a) メタヒューリスティクスの探索性能を向上させるパラメータ調整則
- (b) メタヒューリスティクスの不変性を保ちつつ性能を向上させるためのパラメータ調整則
- (c) メタヒューリスティクスの不変性を補償するためのパラメータ調整則

などの設計を、いわばメタレベルの最適化問題として定式化し、解きたい問題に適した (a) ~ (c) を生成する計算的な方法論の構築を目的する。以上の調整則や更新式などの生成において、それらの数式をまず木構造で表現し、その空間の探索に GP を用いることで、設計対象とするアルゴリズムの設計指針に合目的な調整則や更新式を発見しようとするのが本論文において一貫した方法論である。したがって、最適化アルゴリズムの改良であるいわゆるアルゴリズムの進化ないしは、解きたい最適化問題のために適用する最適化アルゴリズムを適応進化させるために GP を用いるという、計算論的にもアルゴリズム論的にも新たな概念の方法論の提唱により、この分野に一石を投じているものといえる。なお、関数の生成には、シグモイド基底関数やラジアル基底関数などを用い、その線形結合によって張られる関数空間に限定し、その結合パラメータの数値的探索によって最適な関数を生成することも考えられるが、これには

1. 得られる解の解釈が不可能である。
2. 解の多様性の維持が難しい。
3. 初期点と探索空間の設定が難しい。
4. 超高次元の非線形かつ変数分離でない最適化問題となり、解くのが非常に難しい。

といった問題点があげられる。1は、得られる解がパラメータの羅列であるため、得られるパラメータ調整則の関数形が不明であり、どのような原理のもとで動作しているのかの解釈が難しいからである。そのため、得られた解をもとに新たに理解が容易で実装が楽なパラメータ調整則を設計することが難しいということである。2はパラメータ調整則の関数形を探索するのに使われるであろう PSO, ES, DE 等の優秀なアルゴリズムにおいては最良点付近に解が集中するためであり、探索と言いつつも、実際はアルゴリズムによって割り振られたもっとも良かった初期点の局所的探索となり、実質的に探索できていない可能性があるということである。3は基底関数の形状によるものであり、ニューラルネットワークを問題に適用させる際に入力のスケールを規格化する必要があるのと同様に、パラメータ調整則で用いる入力の変数のスケールに応じて適切な初期値を与える必要があるということである。4は基底関数による関数近似特有の問題であり、次元数が低くとも十分な表現能力を得るためには多くのパラメータを必要とし、その多くのパラメータを最適化することは容易ではないということである。

これらに対し、GP を用いた木構造による関数生成による解法は

1. 得られる解の解釈が可能。
2. 解の多様性の維持が容易。

といった利点がある。1については、得られる解が簡単な場合は数式を読み解くことでパラメータの調整則の動作原理を理解することができる。また、数式が複雑である場合においても、数式の近似を行うことで、同様のことが可能となる。2については、多くの探索点や探索木を保持する GA, GP のようなアルゴリズムでは多くの解が保持され、大域的な探索が可能となる。これらのことから、GP を用いた木構造による関数生成は、より有力な関数の生成が可能であると考えられる。

具体的には (a) において、ヒューリスティックな探索法である Particle Swarm Optimization (PSO) に対し、文献 [30][31][32] においてパラメータ調整則を提案している。PSO では、局所最適解から離れている領域では、それに安定的に漸近して集中的な探索を行い、局所最適解にある一定距離以内に近づきすぎると、その局所最適解を不安定平衡点としてそれから離れて多様な探索を持続的に続ける動特性を、そのアルゴリズムに付与することが望ましいとされている。文献 [32] では、そのような動特性を PSO に付与のために、アルゴリズムの不変性を考慮しない、性能向上を目的としたパラメータ調整則の設計を行っている。

(b) においては従前の (1+1)-ES を多点化させた場合の多峰性関数に対して最適な 1/5 ルールにかわるものを設計している [33][34]。

(c) では、文献 [35][36] において DE のための回転不変な交叉を考慮した結果失われた不変性に対する補償を、パラメータによって行うと同時に、性能を向上させたパラメータ調整則の設計支援として GP を用いた。

文献 [37] では、(a), (b), (c) の全てに対して試行し、最終的に得られたもっとも良かったパラメータ調整則を近似、比較している。

類似した研究としてメタヒューリスティクスの更新式そのものを遺伝的プログラミングを用いて設計するというものがある。PSO に関しては文献 [38] において提案されており、DE に関しては文献 [39][40] において提案した。しかし、更新式そのものの設計は、最適化問題の目的関数の平行移動等の単純な変換に対してすらロバスト性を持たず、最適化アルゴリズムとして好ましくないものとなる。このことも、パラメータの調整則を設計する理由につながる。

1.3 本論文の構成

本論文の構成は以下のとおりである。

第2章においては本論文で扱う最適化アルゴリズムとそれらの改良手法と様々な既存のパラメータ調整則を紹介しつつ、それらの問題点を提示する。

第3章においてはパラメータ調整則設計をメタ最適化問題として定式化しつつ、この問題を人間が解く際に用いる設計思想と照らし合わせ、アルゴリズムを用いた解法を提示する。

第4章においては実際にシミュレーションを行い、具体的な結果を示す。それと同時に、得られた調整則の近似を行い、得られたアルゴリズムと従来手法間で性能比較を行う。

第5章においては、本論文の内容をまとめるとともに、今後の展望について述べる。また、付録として統計的検定を用いたアルゴリズム間の比較方法の他、本文で省略したシミュレーションの詳細な結果を示す。

1.4 本論文が対象とする最適化問題とそこで用いられる記号

本論文では、メタヒューリスティクスを対象とし、その更新式中のパラメータを自動調整するための方法論を提供することを主たる目的としている。そのため、解くべき最適化問題は、原則として単一目的関数を最小化する無制約最適化問題を考える。このとき目的

関数は N 次元ユークリッド空間 \mathbb{R}^N から実数値空間 \mathbb{R} への写像とし、それを $E(\mathbf{x})$ と表記し、とくに微分可能性などの解析的な性質の仮定は原則としておかないものとし、これらの仮定が必要な場合は、その都度言及する。そして、この無制約最適化問題を

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad (1.1)$$

と定式化する。したがって、変数 \mathbf{x} はベクトル変数で行ベクトルとし、その成分表現は

$$\mathbf{x} = (x_1, x_2, \dots, x_N)^T \quad (1.2)$$

とする。ただし、上付き文字 T は転置記号を表し、成分表現の場合は紙面の都合上、このように列ベクトル表現に転置記号を添えて表現する。なお、本来解きたい最適化問題が無制約であっても、メタヒューリスティクスでその大域的最適解を探索するとき、それらのアルゴリズムの適用の際に探索範囲を限定するため、変数の成分ごとに上下制限を課する場合があります、この制約については、

$$a_n \leq x_n \leq b_n, \quad n = 1, \dots, N \quad (1.3)$$

と表す。

また、上記の最適化問題を解くための最適化手法のアルゴリズムにおいては、 N 次元ユークリッド空間 \mathbb{R}^N にとられる探索点を、アルゴリズムの反復に伴う時系列とみなして、第 k イテレーションにおける探索点座標を $\mathbf{x}(k)$ 、それらの時系列全体を表す場合は $\mathbf{x}(k)$, $k = 1, 2, \dots$ 、ないしは $\mathbf{x}(\cdot)$ と記す。計算終了のためにイテレーションに上限を設ける場合は、その上限値を K で表す。なお、イテレーションの反復回数 k を離散時間系の時刻に対応づけることがある。

ところで、メタヒューリスティクスでは複数の探索点が用いられることが多いため、この場合には、探索点に番号を付し、それを括弧付きの上付き添字で表すことにし、

$\mathbf{x}^{(p)}(k)$: 第 k イテレーションにおける第 p 探索点の座標

P : 探索点の個数(ただし、 $P = 1$ の場合は、単に上付き添字 (p) を省略する)

$\mathbf{x}^{(p)}(k)$, $p = 1, \dots, P$: 探索点の集合(「探索点群」ともいう)

$x_n^{(p)}(k)$: 第 p 探索点の座標の第 n 成分

と約束する。また、これらの変数などに掛かる係数も同様の表記を行う。係数は原則として c などの小文字を使用し、イテレーション経過とは関係なく一定の値を維持する場合は引数 (k) を省略するが、イテレーション経過とともに変動する場合は時変係数と称し、 $c(k)$ と表す。また、探索点ごとに異なる係数は $c^{(p)}(k)$ 、あるベクトルの成分ごとに異なる係数の場合は $c_n^{(p)}(k)$ と区別を明確にする。なお、ある探索点 $\mathbf{x}^{(p)}$ に対応させてこれとは別の探索点 $\mathbf{x}^{(q)}$ を選ぶ場合があるが、このような場合、探索点番号 q が探索点番号 p に対応していることを明示するために、探索点番号を2重添字として $\mathbf{x}^{(q^{(p)})}$ と記すことにする。ただし、探索点 $\mathbf{x}^{(q)}$ の選択が探索点 $\mathbf{x}^{(p)}$ とは無関係に行われる場合はこの限りではない。

また、メタヒューリスティクスでは、探索点の生成・移動などにおいて、乱数などの確率変数が用いられることが多い。確率変数の場合、慣例では大文字が使用されるが、本論

文では行列との混同を避けるため、確率変数であっても小文字を用いることにする。そして、とくにアルゴリズム中のコンピュータプログラムの関数によって生成される乱数も、その添字などにおいて、

$r(k)$: 第 k イテレーションにおいて生成した一様乱数 (探索点やベクトルの成分に対しては共通)

$r^{(p)}(k)$: 第 k イテレーションにおいて第 p 探索点に対して生成した一様乱数 (ベクトルの成分に対しては共通)

$r_n^{(p)}(k)$: 第 k イテレーションにおいて第 p 探索点の座標ベクトルの第 n 成分に対して生成した一様乱数生成

と区別して表記する。なお、この乱数がどのような分布にしたがうかは、記号

$U_R(a, b)$: 実数空間 \mathbb{R} 上の閉区間 $[a, b]$ の一様分布

$U_Z(a, b)$: $b - a + 1$ 個の整数値集合 $\{a, a + 1, \dots, b\}$ の一様分布

$N(\mu, \sigma^2)$: 実数値空間 \mathbb{R} 上の平均 μ , 分散 σ^2 の正規分布

$\mathbf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: N 次元ベクトル空間上の平均ベクトル $\boldsymbol{\mu}$, $N \times N$ 分散共分散行列 $\boldsymbol{\Sigma}$ の N 次元正規分布

$C(\mu, \delta)$: 平均 μ , スケールパラメータ δ の Cauchy 分布

を用い、たとえば乱数 r が実数値空間上の閉区間 $[a, b]$ の一様分布にしたがう場合、 $r \sim U_R(a, b)$ と明示する。正規分布に従う乱数の場合の添字等の使い方も一様乱数と同じとする。なお、一様乱数の記号には r , その他の乱数の記号には s を用いることにする。

さらに、メタヒューリスティクスでは、たとえば探索点座標を

$$\mathbf{x} = \begin{cases} \mathbf{y} & \text{with probability } \theta \\ \mathbf{z} & \text{with probability } (1 - \theta) \end{cases}$$

と確率的に置き換えをする場合が多い。この場合も、閉区間 $[0, 1]$ の一様乱数 $r \sim U_R(0, 1)$ を生成し、その閾値を θ として

$$\mathbf{x} = \begin{cases} \mathbf{y}, & \text{if } r < \theta \\ \mathbf{z}, & \text{otherwise} \end{cases}$$

と表すことにする。

第2章 代表的なメタヒューリスティクスとそのパラメータ調整法

2.1 本章の内容

メタヒューリスティクスに対する計算論として提唱する「パラメータ調整則の自動設計」を適用する対象として、メタヒューリスティクスの代表的アルゴリズムである Particle Swarm Optimization^[13], Evolution Strategy^[12], Differential Evolution^[14], Firefly Algorithm^[41] を取り上げるが、本章ではこれらの手法の概要、とくにそれらの更新則中に含まれるパラメータに対する従来の調整法について概説する。これらの手法の多くは多点型確率的最適化アルゴリズムであるが、構造的に互いに異なる特徴を有しており、それらの大域的最適解の探索性能も異なっており、またそれらの手法ごとに改良を重ねる研究が行われている。これらを横断して計算性能を比較し、統一した考え方でパラメータ調整則を設計するという研究はない。本章では、これらの手法の特徴の概説だけでなく、ベンチマーク問題に対するシミュレーションによるそれらの横断的な計算性能の比較と、それらを基にしたパラメータ調整に関わる課題提起を行う。

また、本章のシミュレーションに用いた様々なメタヒューリスティクスのコードは Github (<https://github.com/kanemasa1987/Meatpie>) 上にオープンソースとして公開した。幅広く網羅されたメタヒューリスティクスのライブラリは存在しないため、本リポジトリがメタヒューリスティクスの発展に寄与することを望む。

2.2 Particle Swarm Optimization の原理とそのパラメータ調整の課題

2.2.1 Particle Swarm Optimization の原理

Particle Swarm Optimization (以降 PSO と略記) は、1995 年に Kennedy らによって、魚や鳥の餌食行動を模倣して開発され、大域的最適解の良好な近似解を比較的効率よく探索することができる手法として評価され、多くの研究者によって改良手法が提案され、実問題への適用も盛んにされてきている。PSO における第 k イテレーション時の第 p 探索点の位置座標 $\mathbf{x}^{(p)}(k)$ の更新式は、

$$\begin{aligned} v_n^{(p)}(k+1) = & c_0 v_n^{(p)}(k) + c_1 r_{1n}^{(p)}(k)(x_n^{(p\text{-best})}(k) - x_n^{(p)}(k)) \\ & + c_2 r_{2n}^{(p)}(k)(x_n^{(g\text{-best})}(k) - x_n^{(p)}(k)), n = 1, \dots, N \end{aligned} \quad (2.1a)$$

$$\mathbf{x}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k+1) \quad (2.1b)$$

$$\begin{aligned} \mathbf{x}^{(p\text{-best})}(k) &= \operatorname{argmin}_{\mathbf{x}^{(p)}(i)} \{E(\mathbf{x}^{(p)}(i)) | i = 0, 1, \dots, k\} \\ \mathbf{x}^{(g\text{-best})}(k) &= \operatorname{argmin}_{\mathbf{x}^{(g\text{-best})}(k)} \{E(\mathbf{x}^{(g\text{-best})}(k)) | q = 1, \dots, P\} \end{aligned} \quad (2.1c)$$

と表される. ここで, k はイテレーション回数, 括弧で括られた上付き添字 $p = 1, \dots, P$ は探索点番号, 下付き添え字 $n = 1, \dots, N$ はベクトルの要素を表し, $\mathbf{x}^{(p\text{-best})}(k)$ は第 p 探索点の第 k イテレーションまでの探索履歴での最良点で p -best (personal best) 座標, $\mathbf{x}^{(g\text{-best})}(k)$ は探索点群全体での第 k イテレーションまでの探索履歴における最良点で g -best (global best) 座標とよばれる. なお, 更新式 (2.1a) 中のパラメータ c_0, c_1, c_2 の値によっては探索点の挙動が不安定化して, 発散してしまう場合があり, このことについては簡単なシミュレーションにより確認する. また, $r_{1n}^{(p)} \sim U_R(0, 1), r_{2n}^{(p)} \sim U_R(0, 1)$ である.

大域的最適解の探索においては, 大域的最適解が一定の有界な領域の十分内部あることを想定し, この領域に限定して探索を行うことで, 不安定化を抑制して探索の効率化を図ることが行われる. この有界な探索領域として, たとえば上下限領域

$$S = \{\mathbf{x} \mid a_n \leq x_n \leq b_n, n = 1, \dots, N\} \quad (2.2)$$

を設定する場合, このような領域を考慮して, (2.1a) 式の変位 $\mathbf{v}^{(p)}(k+1)$ に制限を課すことがある [43]. この制限の方法は様々なものが考えられているが, たとえば $\mathbf{v}^{(p)}(k+1)$ を

$$v_n^{(p)}(k+1) = \begin{cases} \operatorname{sgn}(v_n^{(p)}(k+1)) \frac{b_n - a_n}{2}, & \text{if } \frac{b_n - a_n}{2} < |v_n^{(p)}(k+1)| \\ v_n^{(p)}(k+1), & \text{otherwise} \end{cases} \quad (2.3)$$

と置きなおすことにより [42], 探索点の過度な不安定化を抑制することができる. また, 別の方法としては探索点の座標に制限を課したり, ないしは探索点を探索領域に閉じ込めたりするものがある. たとえば前者には, 探索点の座標を

$$x_n^{(p)}(k+1) = \begin{cases} b_n, & \text{if } b_n < x_n^{(p)}(k+1) \\ a_n, & \text{else if } x_n^{(p)}(k+1) < a_n \\ x_n^{(p)}(k+1), & \text{otherwise} \end{cases} \quad (2.4)$$

と置きなおすものや, 後者には, トーラス化変換

$$x_n^{(p)}(k+1) = \begin{cases} a_n + [(x_n^{(p)}(k+1) - b_n) \bmod (b_n - a_n)], & \text{if } b_n < x_n^{(p)}(k+1) \\ b_n + [(a_n - x_n^{(p)}(k+1)) \bmod (b_n - a_n)], & \text{if } x_n^{(p)}(k+1) < a_n \\ x_n^{(p)}(k+1), & \text{otherwise} \end{cases} \quad (2.5)$$

により, 探索点を有界領域内に閉じ込める方法がある.

これらをまとめた PSO の基本的な更新手順は次の通りである.

Step 1 最大イテレーション K を設定し, P 個 (複数個) の探索点の初期座標

$\mathbf{x}^{(p)}(0), p = 1, \dots, P$ をランダムに与え, $k = 0$ とおく.

Step 2 (2.1a) 式を用いて変位 $\mathbf{v}^{(p)}(k+1), p = 1, \dots, P$ を生成する.

Step 3 たとえば (2.3) 式を用いて変位 $\mathbf{v}^{(p)}(k+1)$ に制限を課す.

Step 4 (2.1b) 式により, 探索点の座標を更新し, $\mathbf{x}^{(p)}(k+1)$, $p = 1, \dots, P$ を生成する.

Step 5 たとえば (2.4) 式や (2.5) 式を用いて探索点の座標に制限を課す.

Step 6 各探索点の目的関数値 $E(\mathbf{x}^{(p)}(k+1))$, $p = 1, \dots, P$ を計算し, (2.1c) 式を用いて p -best 座標, g -best 座標を更新する.

Step 7 $k \geq K$ ならば計算を終了し, そうでなければ, $k = k + 1$ として Step 2 へ戻る.

2.2.2 Particle Swarm Optimization のパラメータ調整法

PSO による最適解の探索性能は, 更新式 (2.1a) 中のパラメータ c_0 , c_1 , c_2 の値に大きく依存する. 特にパラメータ c_0 は, 慣性係数とよばれており, 探索点の挙動の安定性をより強く支配するパラメータとされている. これらのパラメータを一定値に設定する場合は, 安定的な収束性を示すパラメータの設定方法として, **constriction method**^{[43][44][45]} がある. 具体的には

$$c_0 = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}, \text{ where } \phi = c'_1 + c'_2, \phi > 4 \quad (2.6a)$$

$$c_1 = c_0 c'_1 \quad (2.6b)$$

$$c_2 = c_0 c'_2 \quad (2.6c)$$

によりパラメータを決定する. たとえば, 文献 [45] では $c'_1 = c'_2 = 2.05$ とし,

$$(c_0, c_1, c_2) = (0.729, 1.49455, 1.49455) \quad (2.7)$$

をパラメータとして用いている. この場合, 探索点の挙動は安定なため, 変位ベクトルや探索点の座標の制限は必要がないとされている.

他にも文献 [46] でも, パラメータの設定方法が考察されており, 問題ごとに試行錯誤的にパラメータ設定が必要であるとしつつも,

$$(c_0, c_1, c_2) = (0.6, 1.7, 1.7) \quad (2.8)$$

を安定的なパラメータとし, 良い探索結果を出す値としてシミュレーションに用いている.

これらに対して文献 [47] では, PSO の更新式中の一樣乱数の係数の影響を精緻に考慮した安定性解析により, 探索点の挙動を安定/不安定境界領域に設定することで, 不安定傾向の大域的探索と安定傾向の局所的探索を繰り返して持続的に探索し続けるためのパラメータの一つとして,

$$(c_0, c_1, c_2) = (0.83210, 2.0, 2.0) \quad (2.9)$$

が推奨されている. この手法では, 探索点を安定化させずに持続的に探索を行わせるため, 探索点の座標や変位の制限が必要となる. もとの文献においては (2.5) 式により, トーラス化した後に Step 5.5 として

$$\mathbf{v}^{(p)}(k+1) = \mathbf{x}^{(p)}(k+1) - \mathbf{x}^{(p)}(k) \quad (2.10)$$

とすることで、変位 $\mathbf{v}^{(p)}(k+1)$ を置きなおして安定化させている。また、探索の停滞を防ぐため、更新式は (2.1a) 式の代わりに微小乱数を加えた

$$\begin{aligned} v_n^{(p)}(k+1) &= c_0 v_n^{(p)}(k) + c_1 r_{1n}^{(p)}(k)(x_n^{(p\text{-best})}(k) - x_n^{(p)}(k)) \\ &+ c_2 r_{2n}^{(p)}(k)(x_n^{(g\text{-best})}(k) - x_n^{(p)}(k)) + 2(r_{3n}^{(p)}(k) - 0.5)(b_n - a_n)10^{-11} \end{aligned} \quad (2.11)$$

とした式を用い、乱数 $r_{1n}^{(p)}(k)$ と $r_{2n}^{(p)}(k)$ は成分ごとに同じものを使うことでより良い結果が得られたことが報告されている [47]。なお、この手法では探索終了後に $g\text{-best}$ 座標から局所探索をすることが推奨されている。また、 $r_{3n}^{(p)}(k)$ は乱数による摂動であるため、どの場合においても独立なものを使う。

一方において、一般に大域的最適解を探索するためのメタヒューリスティクスの探索点の挙動特性としては、現イテレーションでの探索点近傍を探索する局所的探索の特性と、その近傍を含むより広範囲の領域を比較的くまなく探索する大域的探索の特性とを併せもつことが望ましいとされている。このような観点から、イテレーション過程において探索特性に変化をつけるために、更新式中のパラメータをイテレーション経過とともに変動する調整法を用いる改良手法が提案されている。それらを分類すると以下のとおりである。

- (1) 大域的探索から局所的探索へと探索点の挙動特性を一方向的に変化させるようにパラメータ調整を行う (文献 [44][48][49] など)。
- (2) 大域的探索や局所的探索の挙動特性を示す指標の目標とする経時的変化を規定し、それに追従するようにパラメータ調整を行う (文献 [49] など)。
- (3) 大域的探索や局所的探索の挙動特性を探索点の挙動の状態に応じて変えるようパラメータ調整を行う (文献 [50] など)。

まず、(1) の代表例は文献 [44] で提案された慣性係数法 (Inertia Weight Method) で、IWA (Inertia Weight Approach) あるいは LDIW (Linearly Decreasing Inertia Weight) という名称で他の文献でしばしば引用される手法である。この手法では、慣性係数 c_0 をイテレーション回数 k に関する時変係数とし、

$$c_0(k) = c_{\text{start}} - (c_{\text{start}} - c_{\text{end}}) \frac{k}{K} \quad (2.12)$$

と線形的に減少させ、探索点の挙動に対して探索過程の序盤は不安定傾向をもたせて大域的探索を行い、探索過程の終盤では安定傾向を強めて局所的探索に移行する手法である。(2.12) 式でもさらに 2 種のパラメータ c_{start} , c_{end} を設定する必要があるが、文献 [44] では $c_{\text{start}} = 0.9$, $c_{\text{end}} = 0.4$, $c_1 = c_2 = 2.0$ と設定されている。

つぎに (2) の例として、文献 [49] の Activity feedback Particle Swarm Optimization (AF-PSO) をあげておく。この手法では、探索点ごとの変位 $\mathbf{v}^p(k)$ を用いて探索点群としての活性度

$$Act(k) = \sqrt{\frac{1}{PN} \sum_{p=1}^P \sum_{n=1}^N (v_n^{(p)}(k))^2} \quad (2.13)$$

を定義し、これを探索点群としての安定/不安定性を支配する指標とし、大域的探索過程から局所的探索過程へ移行するよう、 $Act(k)$ の目標値の経時的変化を

$$Act_{\text{target}}(k) = \max(0, Act_{\text{target}0}(1 - \frac{k}{K'})) \quad (2.14)$$

ないしは

$$Act_{\text{target}}(k) = Act_{\text{target}0} \left(\frac{Act_{\text{target}K}}{Act_{\text{target}0}} \right)^{\frac{k}{K}} \quad (2.15)$$

と線形的ないしは指数的に減少させつつ、イテレーションの更新と共に計算される活性度をもとに、序盤における不安定傾向の大域的探索から終盤における安定傾向の局所的探索に移行するよう、パラメータ c_0 を

$$c_0(k+1) = \begin{cases} \max(c_0(k) - \Delta c_0, c_{0\min}), & \text{if } Act(k) > Act_{\text{target}}(k) \\ c_0(k), & \text{else if } Act(k) = Act_{\text{target}}(k) \\ \min(c_0(k) + \Delta c_0, c_{0\max}), & \text{otherwise} \end{cases} \quad (2.16)$$

とイテレーション経過とともに試行錯誤的に変化させる手法である。この手法においても、活性度という指標の線形減少の更新式にはパラメータ K' , $Act_{\text{target}0}$, 指数的減少の更新式にはパラメータ $Act_{\text{target}0}$, $Act_{\text{target}K}$ を含み、さらに c_0 の更新則である (2.16) 式においてもパラメータ Δc_0 , $c_{0\max}$, $c_{0\min}$ の設定が必要であり、最良パラメータの設定が容易でないという課題がある。文献 [49] においては $K' = 0.9K$, $Act_{\text{target}0} = (b_n - a_n)/4$, $Act_{\text{target}K} = (b_n - a_n)/1000$, $c_0(0) = 1.0$, $c_1 = c_2 = 1.0$, $c_{0\max} = 1.0$, $c_{0\min} = 0.5$, $\Delta c_0 = 0.1$ と設定されている。

最後の (3) の例として、安定性を支配する慣性係数 c_0 を、イテレーション経過とともに不安定傾向の大域的探索から安定傾向の局所的探索に一方的に移行させるのではなく、探索点群の状態に応じて、直接的に慣性係数 c_0 を調整することで、安定/不安定性を変化させる手法「非線形散逸項を有する PSO (Nonlinear Dispersive Termed PSO: NDT-PSO)^[50]」をあげる。その係数 c_0 の調整則は、

$$c_0(k) = 1 - d_1 + d_1 d_0 \exp(-\frac{\|z\|}{d_2 d_0}) \quad (2.17)$$

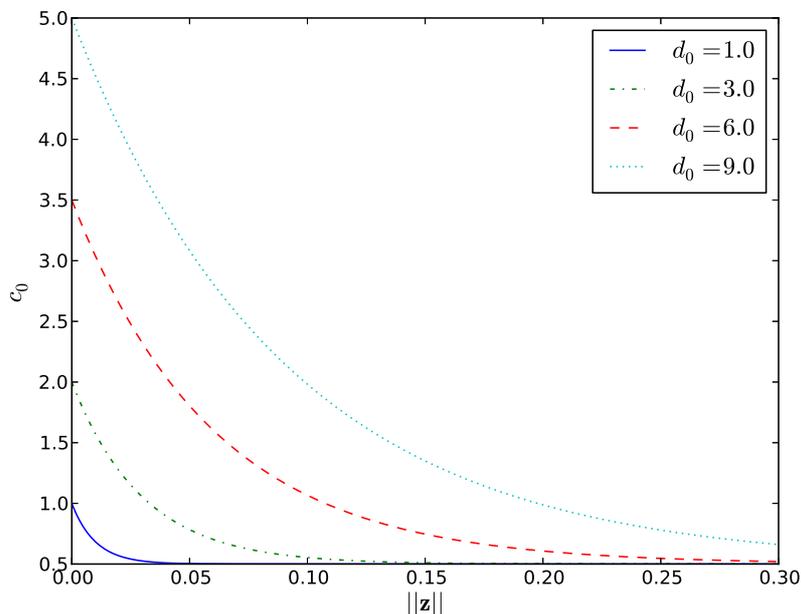
で与えられる。状態の大きさに対する $d_1 = 0.5$, $d_2 = 0.01$ とした場合の係数 c_0 のグラフを Fig. 2.1 に示す。このグラフより状態の大きさ $\|z\|$ が比較的大きいときは、慣性係数 c_0 が 1.0 未満でエネルギーを散逸させる安定状態であるのに対して、 $\|z\|$ が小さくなると慣性係数 c_0 が 1.0 以上になってエネルギーが補給されて不安定状態に移行することがわかる。文献 [50] では、探索点の状態 z を、探索点ごとに異なるものとして

$$z^{(p)}(k) = v^{(p)}(k) \quad (2.18a)$$

$$z^{(p)}(k) = x^{(p)}(k) - x^{(p\text{-best})}(k) \quad (2.18b)$$

$$z^{(p)}(k) = x^{(p)}(k) - x^{(g\text{-best})}(k) \quad (2.18c)$$

とすることが提案されている。(2.18a) 式では、探索点の変位 $v^{(p)}(k)$ を第 p 探索点の状態とし、その大きさが小さくなって停留傾向になるにしたがって不安定化するよう、慣性係


 Fig. 2.1: Shape of eq.(2.17) with $d_1 = 0.5$, $d_2 = 0.01$

数 c_0 が大きくなる. これに対して (2.18b) 式は, p -best 座標 $\mathbf{x}^{(p\text{-best})}(k)$ と現座標 $\mathbf{x}^{(p)}(k)$ の距離を, (2.18c) 式の方は, g -best 座標 $\mathbf{x}^{(g\text{-best})}(k)$ と現座標 $\mathbf{x}^{(p)}(k)$ の距離を第 p 探索点の状態とし, この距離が小さくなるにしたがい, すなわち現座標がこれら best 座標に近づくにしたがって, 慣性係数 c_0 を不安定領域に移行させるもので, 探索点の探索履歴の最良点に多様性がなくなってきたら探索点を不安定化させてその多様性の復帰を期待する調整則といえる. (2.18a)~(2.18c) 式を (2.17) 式に代入すると, 慣性係数 c_0 も引数 (k) と上付き添字 (p) を有する形で,

$$c_0^{(p)}(k) = 1 - d_1 + d_1 d_0(k) \exp\left(-\frac{\|\mathbf{v}^{(p)}(k)\|}{d_2 d_0(k)}\right) \quad (2.19a)$$

$$c_0^{(p)}(k) = 1 - d_1 + d_1 d_0(k) \exp\left(-\frac{\|\mathbf{x}^{(p)}(k) - \mathbf{x}^{(p\text{-best})}(k)\|}{d_2 d_0(k)}\right) \quad (2.19b)$$

$$c_0^{(p)}(k) = 1 - d_1 + d_1 d_0(k) \exp\left(-\frac{\|\mathbf{x}^{(p)}(k) - \mathbf{x}^{(g\text{-best})}(k)\|}{d_2 d_0(k)}\right) \quad (2.19c)$$

となり, イテレーションの経過とともに変動する時変係数になると同時に, 探索点ごとにその慣性係数が自律的に調整される. 以上のことから, 前述した (1), (2) の手法のパラメータ調整では, 不安定傾向から安定傾向への移行がイテレーション経過に関して一方向的であるのに対して, NDT-PSO における安定/不安定状態の移行は双方向的であり, 安定状態や不安定状態を不規則的に繰り返しながら持続探索を続けることが期待される手法といえる. なお, 上式の調整則中のパラメータ d_0 も時変パラメータとし,

$$d_0(k) = d_0 \left(1 - \frac{k}{K}\right) \quad (2.20)$$

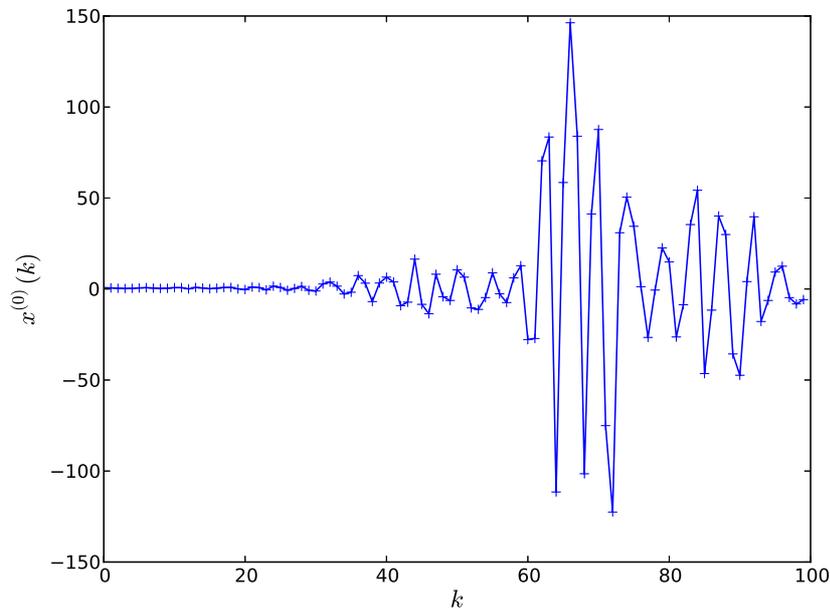


Fig. 2.2: Divergence of a search point of PSO

としている。したがって、調整則中にさらに3種類のパラメータを含むことになるが、文献[50]においては、それらの推奨値を $d_1 = 0.5$, $d_2 = 0.01(b_n - a_n)$, $d_0 \in [3.0, 10.0]$ としている。この手法は、一つのパラメータである慣性係数 c_0 の調整則中にさらにパラメータを複数種類含むことが難点であるが、それらのパラメータを適切に選べば、最大イテレーション回数の範囲内での持続探索によって、高い探索性能を実現することができる手法とされている。

2.2.3 計算機実験による PSO のパラメータ調整法の課題提起

PSO における探索点の不安定性は下記のシミュレーションで確認される。解きたい問題を

$$\underset{x}{\text{minimize}} \quad E(x) = (x - 0.5)^2 \quad (2.21)$$

とし、最大イテレーション $K = 100$, 探索点数 $P = 5$, パラメータ $c_0 = 0.8$, $c_1 = c_2 = 2.0$, 初期値を $x^{(p)}(0) = U_R(0, 1)$, $v^{(p)}(0) = U_R(0, 1) - 0.5$ とした場合、ある探索点の軌跡 $x^{(p)}(k)$ は Fig. 2.2 のようになる。探索点が不安定化し、発振していく様子が確認される。そのため、PSO のパラメータを適切に設定する、ないしは探索点が有界な探索領域から大きく離れない制限を課すなどの安定化操作が必要であることがわかる。

また、問題ごとに最良の設定が異なることを示すため、

1. 大域的な探索が必要となる 2^N -minima 関数
2. POP で多峰性の Rastrigin 関数

3. 単峰性で変数間依存性のある Rosenbrock 関数
4. 探索持続性の指標となる Fractal 関数
5. 1. の座標系を回転させた関数
6. 2. の座標系を回転させた関数
7. 4. の座標系を回転させた関数

の七つの異なる目的関数に対して

1. (2.7) 式のパラメータを用いた PSO (Param1)
2. (2.8) 式のパラメータを用いた PSO (Param2)
3. (2.9) 式のパラメータ, (2.11) 式の更新式, (2.10) 式による変位の置き換えと (2.5) 式による位置制限, 最終 10% のイテレーションは (2.7) 式のパラメータを用いて局所探索をした PSO (Param3)
4. (2.12) 式のパラメータ調整, (2.3) 式の変位制限と (2.5) 式による位置制限を用いた LDIW-PSO (LDIW-PSO)
5. (2.15) 式による目標活性度, (2.3) 式の変位制限と (2.5) 式による位置制限を用いた AF-PSO (AF-PSO)
6. (2.18c) 式を探索点の状態として使い, $d_0 = 2.0$ として, (2.3) 式の変位制限と (2.5) 式による位置制限を用いた NDT-PSO (NDT-PSO)

の 6 種類のアルゴリズムを適用し, $P = 20$, $K = 1,000$ の設定で 20,000 フェンクシオンコールとし, 指定がないパラメータ以外は推奨パラメータを用いてそれぞれ 50 回試行した. これらの試行で得られた目的関数値の中央値 (median), 平均 (mean), 標準偏差 (std), 最小値 (min), 最大値 (max) を目的関数ごとに Table 2.1~2.7 に示す. また, 性能比較の基準として中央値を考えた際の検定として, アルゴリズムのペアによりランク化 t 検定を行った結果を目的関数ごとに Table 2.8~2.14 に示す. この検定においては有意水準を 5% とし, Bonferroni 法を用いて P 値が $0.05/(6C_2 \times 7)$ より小さければ有意であるとし, そのパラメータ調整法のマスを網掛けにしてある. これらの検定結果のまとめは Table 2.15 に示す. alg1 と alg2 を比較して alg1 が統計的に優秀であると判断されれば L (Left), alg2 が優秀であると判断されれば R (Right), 判断がつかない場合は N (Neutral) とし, 最後の 2 列にすべての目的関数に対する L の数と R の数の合計を示してある.

以上のシミュレーションを Table 2.15 よりまとめると, もっとも多くの問題において良い性能を示したパラメータ調整法の PSO は NDT-PSO であるといえる. ただし, Rosenbrock 関数に関してだけは Param 1 と Param 2 の安定的なパラメータ設定の PSO の方が良い性能を示した. このように, 問題ごとにトレードオフの関係があり, 一概にどのアルゴリズム, パラメータを使えば良いかを事前に決定することは難しい. そのため, なるべく多くの関数に対して良い結果を出すパラメータ調整則の設計が望まれるが, 一般的に, 多様化と集中化の概念を反映する調整則は設計者の経験と試行錯誤によって行われている.

Table 2.1: Results for Rastrigin func. with 20,000 function calls

alg	median	mean	std	min	max
AF-PSO	4.97	4.82	6.48	0.00	11.94
LDIW-PSO	2.99	3.78	4.44	0.01	10.94
NDT-PSO	1.99	2.59	2.42	0.00	6.96
Param 1	7.96	9.37	19.36	3.98	21.89
Param 2	10.45	10.55	18.06	3.98	20.89
Param 3	5.21	5.67	7.38	2.11	17.96

Table 2.2: Results for 2^N -minima func. with 20,000 function calls

alg	median	mean	std	min	max
AF-PSO	-726.78	-738.65	655.50	-783.32	-670.23
LDIW-PSO	-755.05	-754.48	538.03	-783.32	-698.50
NDT-PSO	-740.91	-740.35	921.42	-783.32	-670.23
Param 1	-726.78	-724.51	973.62	-783.32	-670.23
Param 2	-726.78	-719.43	1494.69	-783.32	-613.68
Param 3	-770.88	-767.77	172.21	-783.06	-740.23

Table 2.3: Results for Rosenbrock func. with 20,000 function calls

alg	median	mean	std	min	max
AF-PSO	3.53	3.49	1.85	0.01	7.90
LDIW-PSO	4.89	5.02	1.52	2.67	8.84
NDT-PSO	4.74	4.35	2.82	0.15	8.56
Param 1	0.90	1.65	3.08	0.00	8.51
Param 2	2.69	2.51	3.82	0.00	7.68
Param 3	7.16	7.97	94.78	0.36	73.42

Table 2.4: Results for Fractal func. with 20,000 function calls

alg	median	mean	std	min	max
AF-PSO	-9.00	-9.00	0.86	-11.00	-7.00
LDIW-PSO	-20.00	-17.46	21.31	-23.00	-7.00
NDT-PSO	-12.00	-11.66	1.17	-14.00	-9.00
Param 1	-6.00	-7.46	29.15	-22.00	0.00
Param 2	-6.00	-6.34	21.41	-17.00	-1.00
Param 3	-3.00	-2.90	0.54	-5.00	-2.00

Table 2.5: Results for Rastrigin (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
AF-PSO	13.93	13.99	31.13	4.97	24.87
LDIW-PSO	8.96	10.02	16.30	1.99	19.50
NDT-PSO	13.43	13.99	39.61	3.98	32.83
Param 1	16.91	19.18	74.99	5.97	39.80
Param 2	17.41	21.05	152.59	4.97	63.68
Param 3	14.05	13.97	48.79	3.01	29.59

Table 2.6: Results for 2^N -minima (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
AF-PSO	-755.05	-738.40	720.74	-755.05	-638.66
LDIW-PSO	-755.05	-748.23	150.27	-755.05	-725.14
NDT-PSO	-755.05	-746.53	506.14	-783.32	-639.97
Param 1	-698.50	-697.94	1288.48	-783.32	-641.96
Param 2	-698.50	-695.11	1032.35	-755.05	-613.68
Param 3	-755.04	-747.63	534.97	-782.17	-618.63

Table 2.7: Results for Fractal (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
AF-PSO	-4.00	-4.24	4.47	-8.00	-1.00
LDIW-PSO	-4.00	-4.60	5.67	-13.00	-1.00
NDT-PSO	-7.50	-6.68	9.49	-11.00	-1.00
Param 1	-2.00	-2.20	3.02	-8.00	0.00
Param 2	-2.00	-1.94	2.67	-6.00	0.00
Param 3	-2.00	-2.14	0.65	-4.00	-1.00

Table 2.8: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rastrigin func. with 20,000 function calls

alg1	alg2	t-value	P -value
AF-PSO	LDIW-PSO	3.12	2.36e-03
AF-PSO	NDT-PSO	6.29	9.19e-09
AF-PSO	Param 1	-6.61	2.03e-09
AF-PSO	Param 2	-8.07	1.88e-12
AF-PSO	Param 3	-2.65	9.44e-03
LDIW-PSO	NDT-PSO	2.69	8.42e-03
LDIW-PSO	Param 1	-10.37	2.03e-17
LDIW-PSO	Param 2	-11.46	8.36e-20
LDIW-PSO	Param 3	-5.76	1.01e-07
NDT-PSO	Param 1	-13.63	2.32e-24
NDT-PSO	Param 2	-14.10	2.52e-25
NDT-PSO	Param 3	-9.74	4.59e-16
Param 1	Param 2	-1.59	1.14e-01
Param 1	Param 3	5.06	1.96e-06
Param 2	Param 3	6.59	2.29e-09

Table 2.9: Pairwise ranked Welch's t -test on algorithms for 10 dimensional 2^N -minima func. with 20,000 function calls

alg1	alg2	t-value	P -value
AF-PSO	LDIW-PSO	3.34	1.20e-03
AF-PSO	NDT-PSO	0.34	7.34e-01
AF-PSO	Param 1	-2.38	1.94e-02
AF-PSO	Param 2	-2.51	1.37e-02
AF-PSO	Param 3	5.80	9.84e-08
LDIW-PSO	NDT-PSO	-2.48	1.47e-02
LDIW-PSO	Param 1	-5.25	9.45e-07
LDIW-PSO	Param 2	-5.31	7.46e-07
LDIW-PSO	Param 3	1.60	1.14e-01
NDT-PSO	Param 1	-1.92	5.83e-02
NDT-PSO	Param 2	-2.20	2.99e-02
NDT-PSO	Param 3	4.01	1.32e-04
Param 1	Param 2	-0.43	6.66e-01
Param 1	Param 3	7.30	1.79e-10
Param 2	Param 3	7.38	9.39e-11

Table 2.10: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rosenbrock func. with 20,000 function calls

alg1	alg2	t-value	P -value
AF-PSO	LDIW-PSO	-8.93	2.54e-14
AF-PSO	NDT-PSO	-5.43	4.59e-07
AF-PSO	Param 1	5.76	1.03e-07
AF-PSO	Param 2	2.87	5.07e-03
AF-PSO	Param 3	-7.68	2.39e-11
LDIW-PSO	NDT-PSO	1.53	1.28e-01
LDIW-PSO	Param 1	11.36	3.03e-19
LDIW-PSO	Param 2	8.95	4.22e-14
LDIW-PSO	Param 3	-5.34	7.31e-07
NDT-PSO	Param 1	7.66	1.41e-11
NDT-PSO	Param 2	6.05	2.61e-08
NDT-PSO	Param 3	-6.59	3.34e-09
Param 1	Param 2	-1.99	4.98e-02
Param 1	Param 3	-11.12	4.72e-19
Param 2	Param 3	-9.28	5.56e-15

Table 2.11: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Fractal func. with 20,000 function calls

alg1	alg2	t-value	P -value
AF-PSO	LDIW-PSO	10.16	4.81e-16
AF-PSO	NDT-PSO	13.56	3.23e-24
AF-PSO	Param 1	-3.11	2.78e-03
AF-PSO	Param 2	-4.57	2.06e-05
AF-PSO	Param 3	-18.77	3.14e-34
LDIW-PSO	NDT-PSO	-7.57	7.54e-11
LDIW-PSO	Param 1	-9.60	1.07e-15
LDIW-PSO	Param 2	-11.82	1.47e-20
LDIW-PSO	Param 3	-18.03	8.74e-33
NDT-PSO	Param 1	-6.44	9.03e-09
NDT-PSO	Param 2	-7.48	1.09e-10
NDT-PSO	Param 3	-18.34	2.11e-33
Param 1	Param 2	-1.01	3.13e-01
Param 1	Param 3	-5.78	1.57e-07
Param 2	Param 3	-4.02	1.47e-04

Table 2.12: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rastrigin (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
AF-PSO	LDIW-PSO	3.57	5.63e-04
AF-PSO	NDT-PSO	0.36	7.19e-01
AF-PSO	Param 1	-2.78	6.59e-03
AF-PSO	Param 2	-3.00	3.49e-03
AF-PSO	Param 3	-0.05	9.62e-01
LDIW-PSO	NDT-PSO	-3.57	5.65e-04
LDIW-PSO	Param 1	-7.11	1.89e-10
LDIW-PSO	Param 2	-6.40	6.08e-09
LDIW-PSO	Param 3	-3.05	2.99e-03
NDT-PSO	Param 1	-3.09	2.64e-03
NDT-PSO	Param 2	-3.00	3.49e-03
NDT-PSO	Param 3	-0.14	8.86e-01
Param 1	Param 2	-0.25	8.00e-01
Param 1	Param 3	2.85	5.37e-03
Param 2	Param 3	3.00	3.39e-03

Table 2.13: Pairwise ranked Welch's t -test on algorithms for 10 dimensional 2^N -minima (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
AF-PSO	LDIW-PSO	6.24	1.10e-08
AF-PSO	NDT-PSO	4.92	3.45e-06
AF-PSO	Param 1	-5.60	2.07e-07
AF-PSO	Param 2	-6.44	4.37e-09
AF-PSO	Param 3	-2.55	1.26e-02
LDIW-PSO	NDT-PSO	-0.83	4.07e-01
LDIW-PSO	Param 1	-9.39	1.55e-14
LDIW-PSO	Param 2	-11.36	3.41e-19
LDIW-PSO	Param 3	-5.62	1.94e-07
NDT-PSO	Param 1	-7.69	1.41e-11
NDT-PSO	Param 2	-9.02	1.58e-14
NDT-PSO	Param 3	-4.64	1.23e-05
Param 1	Param 2	-0.14	8.87e-01
Param 1	Param 3	7.16	2.26e-10
Param 2	Param 3	8.76	6.73e-14

Table 2.14: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Fractal (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
AF-PSO	LDIW-PSO	0.44	6.62e-01
AF-PSO	NDT-PSO	4.85	5.36e-06
AF-PSO	Param 1	-5.44	3.95e-07
AF-PSO	Param 2	-6.09	2.20e-08
AF-PSO	Param 3	-6.34	1.07e-08
LDIW-PSO	NDT-PSO	4.04	1.13e-04
LDIW-PSO	Param 1	-6.61	2.06e-09
LDIW-PSO	Param 2	-7.13	1.85e-10
LDIW-PSO	Param 3	-8.48	3.25e-13
NDT-PSO	Param 1	-7.59	3.11e-11
NDT-PSO	Param 2	-8.53	3.39e-13
NDT-PSO	Param 3	-7.24	3.50e-10
Param 1	Param 2	-0.77	4.44e-01
Param 1	Param 3	0.47	6.39e-01
Param 2	Param 3	1.46	1.49e-01

Table 2.15: Pairwise ranked Welch's t -test for 10 dimensional functions with 20,000 function calls

alg1	alg2	Rastrigin	2^N -minima	Rosenbrock	Fractal	Rastrigin (Rot.)	2^N -minima (Rot.)	Fractal (Rot.)	L	R
AF-PSO	LDIW-PSO	N	N	L	R	N	R	N	1	2
AF-PSO	NDT-PSO	R	N	L	R	N	R	R	1	4
AF-PSO	Param 1	L	N	R	N	N	L	L	3	1
AF-PSO	Param 2	L	N	N	L	N	L	L	4	0
AF-PSO	Param 3	N	R	L	L	N	N	L	3	1
LDIW-PSO	NDT-PSO	N	N	N	L	N	N	R	1	1
LDIW-PSO	Param 1	L	L	R	L	L	L	L	6	1
LDIW-PSO	Param 2	L	L	R	L	L	L	L	6	1
LDIW-PSO	Param 3	L	N	L	L	N	L	L	5	0
NDT-PSO	Param 1	L	N	R	L	N	L	L	4	1
NDT-PSO	Param 2	L	N	R	L	N	L	L	4	1
NDT-PSO	Param 3	L	R	L	L	N	L	L	5	1
Param 1	Param 2	N	N	N	N	N	N	N	0	0
Param 1	Param 3	R	R	L	L	N	R	R	2	3
Param 2	Param 3	R	R	L	L	N	R	R	2	3

2.3 Evolution Strategy の原理とそのパラメータ調整の課題

2.3.1 単点型 Evolution Strategy の原理とそのパラメータ調整則

Evolution Strategy (以降 ES と略記) は, 1960 年代に Rechenberg^[10] によって初めて提案され, その後 Schwefel^[11] によって基本的な改良が施されて発展したもっとも代表的なメタヒューリスティクスの一つであり, 生物の進化を模倣して提案されたため, 遺伝的アルゴリズムと並んで進化計算と称せられる計算手法にも分類されている. しかし, 遺伝的アルゴリズムは原理的には 0-1 組合せ最適化問題を解く手法であるのに対して, ES は実数値変数に関する最適化問題を解く手法という点で大きく異なっている. ES における探索点の更新式自体はきわめて単純で, 第 k イテレーション時の探索点の位置座標を $\mathbf{x}(k)$ とすると, その更新式は,

$$\mathbf{u}(k+1) = \mathbf{x}(k) + \sigma(k)\mathbf{s}(k) \quad (2.22)$$

$$\mathbf{x}(k+1) = \begin{cases} \mathbf{u}(k+1), & \text{if } E(\mathbf{u}(k+1)) < E(\mathbf{x}(k)) \\ \mathbf{x}(k), & \text{otherwise} \end{cases} \quad (2.23)$$

で与えられる. ここで, $\mathbf{s}(k) \sim \mathbf{N}(\mathbf{0}, \mathbf{I})$ で, 平均が $\mathbf{0}$ で分散共分散行列が単位行列 \mathbf{I} の N 次元正規乱数 $\mathbf{s}(k)$ により変位させるのが (2.22) 式である. この変位によって得られるいわゆる突然変異点 $\mathbf{u}(k+1)$ が目的関数 $E(\mathbf{x})$ の値を $\mathbf{x}(k)$ より良くするのであれば, その突然変異点を受理して新しい探索点の位置座標とし, そうでなければ元の位置座標のままとするのが (2.23) 式である. 突然変異点 $\mathbf{u}(k+1)$ が目的関数 $E(\mathbf{x})$ の値を $\mathbf{x}(k)$ より良くする場合を「突然変異は成功」と称することにする.

ところで, この更新式が最適化手法としてある程度機能するためには, 時変パラメータであるすべての変数成分に共通の分散 $\sigma(k)$ を調整する必要がある. Schwefel は, 経験則による分散 $\sigma(k)$ の更新則として「1/5 ルール」を提案した. これは, 現在のイテレーション k から遡った一定のイテレーション期間で突然変異が成功した頻度を $p_s(k)$ とするとき, $p_s(k)$ が 1/5 以上ならば分散を増加させ, 1/5 未満ならば分散を減少させるという調整則であり, 具体的にはたとえば, $0 < c < 1.0$ とした

$$\sigma(k+1) = \begin{cases} \sigma(k)/c, & \text{if } 1/5 \leq p_s(k) \\ \sigma(k) \times c, & \text{otherwise} \end{cases} \quad (2.24)$$

や,

$$\sigma(k+1) = \sigma(k) \times \exp\left(\frac{1}{3} \times \frac{p_s(k) - 1/5}{1 - 1/5}\right) \quad (2.25)$$

などがある. 単一の探索点をこのような 1/5 ルールで更新する ES は, (1+1)-ES と称され, これらは $\sigma(k+1) = f(\sigma(k), p_s(k))$ と一般的に記述できる. この調整則を含む単点型 ES の基本的な更新手順を示すと, 以下のようになる

Step 1 最大イテレーション K を設定し, 探索点 $\mathbf{x}(0)$ をランダムに, 分散の初期値 $\sigma(0)$ を適当に与え, $k = 0$ とおく.

Step 2 1/5 ルール ((2.24) 式や (2.25) 式) を用いて $\sigma(k)$ を更新する.

Step 3 (2.22) 式を用いて突然変異点 $\mathbf{u}(k+1)$ を生成する.

Step 4 目的関数値 $E(\mathbf{u}(k+1))$ を計算し, (2.23) 式を用いて探索点 \mathbf{x} を更新する.

Step 5 $k \geq K$ ならば計算を終了し, そうでなければ $k = k+1$ として Step 2 へ戻る.

なお, 1/5 ルールは線形関数と2次関数に対する解析的な導出や, 経験則によって設計されたものである. 線形関数に対しては $p \approx 0.184$, 2次関数に対しては $p \approx 0.270$ が最適であると Rechenberg は導出し^[51], これらの値は他の関数に対しても経験的に良好であるとされている. 1/5 という数値は目安であって, 問題ごとに微妙に異なる値にする必要があり, 絶対的な値ではないことを注記しておく.

2.3.2 多点型 Evolution Strategy の原理とそのパラメータ調整則

メタヒューリスティクスの多くの手法が探索点を複数用いる多点探索法であり, 複数の探索点による目的関数の何らかの景観情報を用いることで大域的探索性能を高めている. そこで, ES においても探索点を多点化することで, その大域的探索性能を高めた手法が提案されている. その代表的手法が「パラータ内蔵型回転操作付き ES ((μ, λ)-ES)」と「共分散行列適応型 ES (Covariance Matrix Adaptation ES: CMA-ES)^[52]」である. 前者は, 慣例的に親の探索点数と子の探索点数をそれぞれ μ と λ で表記して, (μ, λ)-ES と称したため, 本論文でもこの略称を用いるが, 親探索点数と子探索点数については, 本論文での記述の統一性を意識してそれぞれ P と Q で表すことにする.

2.3.3 (μ, λ)-ES

(μ, λ)-ES のアルゴリズムは, 大きく分けて

1. 複数 (P 個) の探索点の中から選んだ親探索点の組 $(\mathbf{x}^{(p_1)}(k), \mathbf{x}^{(p_2)}(k))$ から子探索点 $\mathbf{x}^{(q)}(k)$ を生成する「組み換え操作」
2. 子探索点 $\mathbf{x}^{(q)}(k)$ の突然変異点を改めて子探索点 $\mathbf{x}^{(q)}(k+1)$ とする「回転変換付き突然変異操作」
3. 上記の子探索点生成を $Q (> P)$ 回繰り返して得られる子探索点 $\mathbf{x}^{(q)}(k+1)$, $q = 1, \dots, Q$ の目的関数値 $E(\mathbf{x}^{(q)}(k))$, $q = 1, \dots, Q$ の上位 P 個の子探索点のみを選択して残りを削除する「淘汰操作」

からなる. なお, 「淘汰操作」では, 生成した Q 個の子探索点と P 個の親探索点とを併せた合計個の中から, 目的関数値が上位個の探索点を選択して残りを削除する手法もあり, 前者と区別して「($\mu + \lambda$)-ES」と称されている.

「組み換え操作」の具体的な演算式は, 一定頻度 $C_c (0 < C_c < 1)$ で

$$\mathbf{x}^{(q)}(k) = (\mathbf{x}^{(p_1)}(k) + \mathbf{x}^{(p_2)}(k))/2 \quad (2.26)$$

と親探索点の重心を採用する場合や、やはり一定頻度 C_c で

$$x_n^{(q)}(k) = \begin{cases} x_n^{(p_1)}(k), & \text{if } U_R(0, 1) < 1/2 \\ x_n^{(p_2)}(k), & \text{otherwise} \end{cases} \quad (2.27)$$

$$n = 1, \dots, N$$

と成分ごとに両親探索点の成分を等確率で採用する方式がある。なお、残りの頻度 $1 - C_c$ ではこの組み換え操作を行わないが、この場合には、片方の親探索点をそのまま子探索点とする。

また、「回転変換付き突然変異操作」の具体的演算は、子探索点ごとに異なり、かつ成分ごとに異なる値の分散ベクトル $\boldsymbol{\sigma}^{(q)}(k) = (\sigma_1^{(q)}(k), \dots, \sigma_N^{(q)}(k))^T$ をもつ N 次元正規乱数 $\mathbf{s}^{(q)}(k) \sim N(0, \boldsymbol{\sigma}^{(q)}(k))$ に、やはりイテレーション k において子探索点ごとに異なる回転角度ベクトル $\boldsymbol{\alpha}^{(q)}(k) = (\alpha_{12}^{(q)}(k), \dots, \alpha_{N-1N}^{(q)}(k))^T$ を有する回転行列 $\mathbf{R}(\boldsymbol{\alpha}^{(q)}(k))$ を乗じて原点まわりに回転させた変位を子探索点ごとに加算する形で、突然変異点 $\mathbf{x}^{(q)}(k+1)$ を

$$\mathbf{x}^{(q)}(k+1) = \mathbf{x}^{(q)}(k) + \mathbf{R}(\boldsymbol{\alpha}^{(q)}(k))\mathbf{s}^{(q)}(k) \quad (2.28)$$

とする。ただし、回転行列 $\mathbf{R}(\boldsymbol{\alpha}^{(q)})$ は、

$$\mathbf{R}(\boldsymbol{\alpha}^{(q)}) = \prod_{m=1}^N \prod_{n>m}^N \mathbf{R}^{m,n}(\alpha_{mn}^{(q)})$$

$$r_{k,l}^{m,n}(\alpha_{mn}^{(q)}) = \begin{cases} \cos(\alpha_{mn}^{(q)}) & k = m, l = m \\ -\sin(\alpha_{mn}^{(q)}) & k = m, l = n \\ \sin(\alpha_{mn}^{(q)}) & k = n, l = m \\ \cos(\alpha_{mn}^{(q)}) & k = n, l = n \\ 1 & k = l \neq m, k = l \neq n \\ 0 & \text{otherwise} \end{cases} \quad (2.29)$$

で表される。ここで行列 $\mathbf{R}^{m,n}(\alpha_{mn}^{(q)})$ は m, n 軸に対する回転行列を表す。

(μ, λ) -ES のパラメータは、(2.26) 式や (2.27) 式による親探索点と異なる子探索点の生成頻度を表すパラメータ頻度 C_c 以外に、突然変異における分散ベクトル $\boldsymbol{\sigma}^{(q)}$ と回転角度ベクトル $\boldsymbol{\alpha}^{(q)}$ があり、これらはイテレーションの更新とともにやはり正規乱数 $s^{(q)}(k) \sim N(0, \gamma')$, $s_n^{(q)}(k) \sim N(0, \gamma)$, $\bar{s}_{mn}^{(q)}(k) \sim N(0, \beta)$ を用いて

$$\begin{aligned} \sigma_i^{(q)}(k+1) &= \sigma_i^{(q)}(k) \exp(s^{(q)}(k) + s_n^{(q)}(k)) \\ \alpha_{mn}^{(q)}(k+1) &= \alpha_{mn}^{(q)}(k) + \bar{s}_{mn}^{(q)}(k) \end{aligned} \quad (2.30)$$

と、子探索点ごとに、かつそれらの成分ごとに経時的に調整される。ここで、 γ, γ', β はパラメータであり、 $\gamma = 1/(\sqrt{2\sqrt{N}})$, $\gamma' = 1/(\sqrt{2N})$, $\beta = 5/360 \times 2\pi \approx 0.0873$ が推奨値である。

(1+1)-ES や、それを単純に多点化したアルゴリズムは等方向的な問題は比較的効率よく解くことができるが、(2.22) 式からわかるように σ を全成分で共通の値で用いるため悪条

件の問題を効率よく解くことができない。そこで、 σ を成分ごとに用意し、回転パラメータも導入して変数間に依存性のある悪条件の問題を解けるようにしようとしたのが (μ, λ) -ESであると考えられる。このとき、従前の1/5ルールが適用できなくなったため、 σ と α の値も正規乱数を用いて同時に試行錯誤的に調整することにした手法である。

これらの操作に淘汰操作も加えた (μ, λ) -ESの計算手順をまとめると以下ようになる。

Step 1 最大イテレーション K を設定し、 P 個(複数個)の探索点 $\mathbf{x}^{(p)}(0)$ 、 P 種のパラメータの初期値 $\sigma^{(p)}(0)$ 、 $\alpha^{(p)}(0)$ 、 $p = 1, \dots, P$ をランダムに与え、 $k = 0$ とおく。

Step 2 組み換え操作((2.26)式や(2.27)式)を用いて子探索点 $\mathbf{x}^{(q)}(k+1)$ のもととなる点 $\mathbf{x}^{(q)}(k)$ を Q 個生成する。

Step 3 (2.30)式を用いてパラメータの更新を行い、(2.28)式を用いて子探索点 $\mathbf{x}^{(q)}(k+1)$ を生成する。

Step 4 子探索点 $\mathbf{x}^{(q)}(k+1)$ の目的関数値 $E(\mathbf{x}^{(q)}(k+1))$ 、 $q = 1, \dots, Q$ を計算し、子探索点の中で優秀な P 探索点、もしくは親探索点と子探索点の中で優秀な P 探索点を残す。

Step 5 $k \geq K$ ならば計算を終了し、そうでなければ $k = k + 1$ としてStep 2へ戻る。

2.3.4 CMA-ES

CMA-ESのアルゴリズムは、

$$\mathbf{x}^{(p)}(k) = \mathbf{m}(k) + \sigma(k)\mathbf{N}^{(p)}(0, \mathbf{C}(k)), p = 1, 2, \dots, P \quad (2.31)$$

で表され、これに

1. 1/5ルールに替わるCumulative Step-Size Adaption(CSA)によるステップ幅 $\sigma(k)$ 更新
2. rank-one update と rank- μ update による分散共分散行列 $\mathbf{C}(k)$ の更新

によるパラメータ調整則が加わったアルゴリズムである。まずは従来の1/5ルールの考え方を改良したCumulative Step-Size Adaption(CSA)から説明する。CSAではまず P 個の親探索点の中から、目的関数値 $E(\mathbf{x}^{(p)}(k))$ が良い上位 Q 探索点の重心を、

$$\mathbf{m}(k+1) = \sum_{q=1}^Q w_q \mathbf{x}^{(q)}(k) \quad (2.32)$$

により求める。このときの個数 Q は $\mu = P/2$ が推奨されている。この重心の変位を

$$\mathbf{y}(k+1) = \frac{\mathbf{m}(k+1) - \mathbf{m}(k)}{\sigma(k)} \quad (2.33)$$

により求め、重心の変遷を

$$\mathbf{p}_\sigma(k+1) = (1 - c_\sigma)\mathbf{p}_\sigma(k) + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w} \mathbf{C}^{-1/2}(k) \mathbf{y}(k+1) \quad (2.34)$$

により更新する． \mathbf{C} は重心の変遷の分散共分散行列で， $\mathbf{p}_\sigma(k+1)$ は分散共分散行列が $\mathbf{N}(\mathbf{0}, \mathbf{I})$ となるように規格化して考えた際に，重心が特定の期間にわたってどの程度動いたのかを示すベクトルである．パラメータの推奨値は $c_\sigma \approx 4/N$ ， $\mu_w = \frac{1}{\sum_{q=1}^Q w_q^2} \approx \frac{P}{4}$ である．この重心の変遷をもとに，ステップ幅を

$$\sigma(k+1) = \sigma(k) \times \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma(k+1)\|}{E\|\mathbf{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right) \quad (2.35)$$

により計算するのが CSA である．ここで， $d_\sigma = 1 + 2\max(0, \sqrt{(\mu_w - 1)/(N + 1)} - 1) + c_\sigma$ ， $E\|\mathbf{N}(\mathbf{0}, \mathbf{I})\|$ は N 次元正規分布の2乗ノルムの期待値であり， $E\|\mathbf{N}(\mathbf{0}, \mathbf{I})\| \approx \sqrt{N}(1 - \frac{1}{4N} + \frac{1}{21N^2})$ である．分散共分散行列が $\mathbf{N}(\mathbf{0}, \mathbf{I})$ となるように規格化して考えた際に，重心がランダムに動いていれば $\sigma(k)$ はあまり変化せず，重心があまり動かなければ $\sigma(k)$ を小さく，重心が特定の方向に向かっていれば $\sigma(k)$ を大きくするという更新式になっている．

次に，rank-one update について説明する．rank-one update においては重心の変遷を元に分散共分散行列の更新を考える．(2.33) 式により， $\mathbf{y}(k+1)$ 方向へ重心が移動した際に，より $\mathbf{y}(k+1)$ 方向へ移動させたい ($\mathbf{y}(k+1)$ 方向の相関を強めたい) 場合は

$$\mathbf{C}(k+1) = (1 - c_{\text{cov}})\mathbf{C}(k) + c_{\text{cov}}\mu_w\mathbf{y}(k+1)\mathbf{y}^T(k+1) \quad (2.36)$$

により分散共分散行列を更新すればよい．ここで，パラメータ c_1 は $c_1 \approx 2/N^2$ が推奨されている．

rank- μ update では良い結果を残した正規分布乱数をもとに，分散共分散行列を更新する．上位 μ 個の正規分布乱数から計算される重み付の分散共分散行列に近づける．

$$\mathbf{C}_\mu(k+1) = \sum_{p=1}^{\mu} w_i \mathbf{N}^{(p)}(0, \mathbf{C}(k)) \mathbf{N}^{(p)}(0, \mathbf{C}(k))^T \quad (2.37)$$

$$\mathbf{C}(k+1) = (1 - c_{\text{cov}})\mathbf{C}(k) + c_{\text{cov}}\mathbf{C}_\mu(k+1) \quad (2.38)$$

$$c_{\text{cov}} \approx \mu_w/N^2, c_{\text{cov}} \leq 1 \quad (2.39)$$

これらの操作を統合した CMA-ES の計算手順をまとめると以下ようになる．

Step 1 最大イテレーション K を設定し，探索点の重心 $\mathbf{m}(1)$ をランダムに与え， $k = 1$ とおく．

Step 2 P 個 (複数個) の探索点 $\mathbf{x}^{(p)}(k)$ ， $p = 1, \dots, P$ を (2.31) 式により生成し，その目的関数値 $E(\mathbf{x}^{(p)}(k))$ を計算する．

Step 3 (2.32) 式を用いて重心を計算し，(2.33) 式により規格化された重心の変位ベクトルを求める．

Step 4 規格化された状況下での重心の変遷ベクトル

$$\mathbf{p}_\sigma(k+1) = (1 - c_\sigma)\mathbf{p}_\sigma(k) + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w} \mathbf{C}(k)^{-\frac{1}{2}} \mathbf{y}_w(k+1) \quad (2.40)$$

を求める．

Step 5 重心の変遷ベクトル

$$\begin{aligned} \mathbf{p}_c(k+1) &= (1 - c_c)\mathbf{p}_c(k) \\ &+ \mathbf{1}_{[0, 1.5\sqrt{N}]}(\|\mathbf{p}_\sigma(k+1)\|)\sqrt{1 - (1 - c_c)^2}\sqrt{\mu_w}\mathbf{y}_w(k+1) \end{aligned} \quad (2.41)$$

を求め,

$$\begin{aligned} \mathbf{C}(k+1) &= (1 - c_1 - c_\mu + c_s)\mathbf{C}(k) + c_1\mathbf{p}_c(k+1)\mathbf{p}_c(k+1)^T \\ &+ c_\mu \sum_{p=1}^{\mu} w_p \mathbf{N}(0, \mathbf{C}(k))\mathbf{N}(0, \mathbf{C}(k))^T \end{aligned} \quad (2.42)$$

により分散共分散行列を更新する. ここで,

$$c_s(k+1) = (1 - \mathbf{1}_{[0, 1.5\sqrt{N}]}(\|\mathbf{p}_\sigma(k+1)\|))c_1c_c(2 - c_c) \quad (2.43)$$

である.

Step 6 (2.34) 式によりステップ幅 σ を更新する.

Step 7 $k \geq K$ ならば計算を終了し, そうでなければ $k = k + 1$ として Step 2 へ戻る.

これらの操作により, CMA-ES は更新される方向に効率よく動くアルゴリズムとなっている.

2.3.5 計算機実験による ES のパラメータ調整法の課題提起

PSO と同様に

1. 大域的な探索が必要となる 2^N -minima 関数
2. POP で多峰性の Rastrigin 関数
3. 単峰性で変数間依存性のある Rosenbrock 関数
4. 探索持続性の指標となる Fractal 関数
5. 1. の座標系を回転させた関数
6. 2. の座標系を回転させた関数
7. 4. の座標系を回転させた関数

の七つの異なる目的関数に対して

1. 推奨値のパラメータを用いた (1+1)-ES
2. (1+1)-ES と同じパラメータと更新式を用いつつ, 探索点数を $P = 20$ とした (20+20)-ES

3. 推奨値のパラメータを用いた CMA-ES

の3種類のアプローチを適用する。(1+1)-ESは $P = 1$, $K = 20,000$, (20+20)-ESは $P = 20$, $K = 1,000$, CMA-ESは $P = 10N = 100$, $K = 200$ と設定し, 指定がないパラメータ以外は推奨パラメータを用いてそれぞれ50回試行した. 結果の中央値 (median), 平均 (mean), 標準偏差 (std), 最小値 (min), 最大値 (max) は目的関数ごとに Table 2.16~2.22 に示す. 性能比較の基準として中央値を考えた際の検定として, アルゴリズムのペアによりランク化 t 検定を行った結果は目的関数ごとに Table 2.23~2.29 に示す. これらの Table においては有意水準を5%とし, Bonferroni法を用いて P 値が $0.05/(3C_2 \times 7)$ より小さければ有意であるとし, マスを網掛けにしてある. 検定結果のまとめは Table 2.30 に示す. alg1 と alg2 を比較して alg1 が統計的に優秀であると言えれば L (Left), alg2 が優秀であると言えれば R (Right), 判断がつかない場合は N (Neutral) としてある. 最後の2列にはこれらの L の数と R の数を示す.

Table 2.30 を使ってこれらのシミュレーションをまとめると, もっとも多くの問題において良い結果を出したのは CMA-ES であることが確認される. ただし, CMA-ES は最悪値と分散が悪い場合が多いことが Table 2.16~2.22 より確認でき, 初期の探索が上手いかなければ悪い結果を得ることになることも併記しておく.

Table 2.16: Results for Rastrigin func. with 20,000 function calls

alg	median	mean	std	min	max
(1+1)-ES	72.63	73.31	589.60	28.85	123.37
(20+20)-ES	33.86	34.18	76.74	16.93	54.74
CMA-ES	4.97	17.88	361.42	0.99	75.96

Table 2.17: Results for 2^N -minima func. with 20,000 function calls

alg	median	mean	std	min	max
(1+1)-ES	-641.96	-648.74	2400.12	-755.05	-557.14
(20+20)-ES	-726.73	-730.52	427.22	-782.73	-670.16
CMA-ES	-726.78	-668.97	12083.63	-783.32	-456.87

Table 2.18: Results for Rosenbrock func. with 20,000 function calls

alg	median	mean	std	min	max
(1+1)-ES	5.63	12.87	621.32	0.82	118.55
(20+20)-ES	6.60	6.42	2.96	0.66	9.68
CMA-ES	7.73	228.31	1198391.73	0.76	5825.12

Table 2.19: Results for Fractal func. with 20,000 function calls

alg	median	mean	std	min	max
(1+1)-ES	0.00	0.00	0.00	0.00	0.00
(20+20)-ES	0.00	-0.02	0.02	-1.00	0.00
CMA-ES	-29.00	-26.08	81.59	-30.00	0.00

Table 2.20: Results for Rastrigin (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
(1+1)-ES	70.14	78.86	1000.28	18.90	164.17
(20+20)-ES	33.83	34.35	105.86	10.21	62.69
CMA-ES	3.48	18.75	609.08	0.00	109.21

Table 2.21: Results for 2^N -minima (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
(1+1)-ES	-698.50	-690.59	2252.64	-783.32	-585.41
(20+20)-ES	-755.03	-756.02	327.09	-783.32	-725.41
CMA-ES	-755.05	-674.08	21543.76	-783.32	-330.81

Table 2.22: Results for Fractal (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
(1+1)-ES	0.00	0.00	0.00	0.00	0.00
(20+20)-ES	0.00	-0.02	0.02	-1.00	0.00
CMA-ES	-29.00	-26.16	78.01	-30.00	0.00

Table 2.23: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rastrigin func. with 20,000 function calls

alg1	alg2	t-value	P -value
(1+1)-ES	(20+20)-ES	11.36	1.76e-19
(1+1)-ES	CMA-ES	13.16	2.28e-23
(20+20)-ES	CMA-ES	5.31	8.01e-07

Table 2.24: Pairwise ranked Welch's t -test on algorithms for 10 dimensional 2^N -minima func. with 20,000 function calls

alg1	alg2	t-value	P -value
(1+1)-ES	(20+20)-ES	9.83	4.32e-16
(1+1)-ES	CMA-ES	1.57	1.20e-01
(20+20)-ES	CMA-ES	0.10	9.18e-01

Table 2.25: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rosenbrock func. with 20,000 function calls

alg1	alg2	t-value	P -value
(1+1)-ES	(20+20)-ES	-1.12	2.67e-01
(1+1)-ES	CMA-ES	-3.03	3.20e-03
(20+20)-ES	CMA-ES	-4.83	5.08e-06

Table 2.26: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Fractal func. with 20,000 function calls

alg1	alg2	t-value	P -value
(1+1)-ES	(20+20)-ES	1.00	3.22e-01
(1+1)-ES	CMA-ES	16.94	3.97e-22
(20+20)-ES	CMA-ES	16.45	1.66e-22

Table 2.27: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rastrigin (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
(1+1)-ES	(20+20)-ES	11.29	2.82e-19
(1+1)-ES	CMA-ES	11.60	5.60e-20
(20+20)-ES	CMA-ES	5.27	1.05e-06

Table 2.28: Pairwise ranked Welch's t -test on algorithms for 10 dimensional 2^N -minima (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
(1+1)-ES	(20+20)-ES	7.43	6.85e-11
(1+1)-ES	CMA-ES	2.78	6.84e-03
(20+20)-ES	CMA-ES	1.40	1.66e-01

Table 2.29: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Fractal (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
(1+1)-ES	(20+20)-ES	1.00	3.22e-01
(1+1)-ES	CMA-ES	17.05	3.08e-22
(20+20)-ES	CMA-ES	16.55	1.26e-22

Table 2.30: Pairwise ranked Welch's t -test for 10 dimensional functions with 20,000 function calls

alg1	alg2	Rastrigin	2^N -minima	Rosenbrock	Fractal	Rastrigin (Rot.)	2^N -minima (Rot.)	Fractal (Rot.)	L	R
(1+1)-ES	(20+20)-ES	R	R	N	N	R	R	N	0	4
(1+1)-ES	CMA-ES	R	N	N	R	R	N	R	0	4
(20+20)-ES	CMA-ES	R	N	L	R	R	N	R	1	4

2.4 Differential Evolution の原理とそのパラメータ調整

2.4.1 Differential Evolution の原理

Differential Evolution は、1995年に Price と Storn^[14]によって提案された、やはり目的関数の勾配情報を必要としないヒューリスティックな実数値関数の多点型最適化アルゴリズムで、大域的最適解を比較的効率よく求めることができる手法として近年評価されている。DE のアルゴリズムの特徴は、複数の探索点のなかで第 k イテレーションにおける更新したい探索点を $\mathbf{x}^{(p)}(k)$ とするとき、それとは異なりかつ互いに異なる三つの探索点 $\mathbf{x}^{(q_1^{(p)})}(k)$, $\mathbf{x}^{(q_2^{(p)})}(k)$, $\mathbf{x}^{(q_3^{(p)})}(k)$ を探索点群からランダムに選び、それらを用いた差分情報から「突然変異操作」により変異点 $\mathbf{v}^{(p)}(k)$ を生成するステップ、および探索点 $\mathbf{x}^{(p)}(k)$ と変異点 $\mathbf{v}^{(p)}(k)$ との「一様交叉」により試験点 $\mathbf{u}^{(p)}(k)$ を生成するステップからなり、試験点が目的関数を改善する場合に、それを探索点 $\mathbf{x}^{(p)}(k)$ の更新点として採用するという単純なアルゴリズムである。ただし、突然変異点生成時に選ばれた三つの点の上付添字 $q_i^{(p)}$, $i = 1, 2, 3$ の記述については、それらが更新したい探索点 $\mathbf{x}^{(p)}(k)$ ごとにそれに対応して選ばれるため、括弧付き探索点番号 (p) が添字 q_i にさらに上付きで添えられていることに注意されたい¹。

「突然変異操作」は、数多く提案されており、それらを区別する名称が付けられている。Price と Storn によって初期に提案された変異点生成公式^[53]だけでも

DE/rand/1

$$\mathbf{v}^{(p)}(k) = \mathbf{x}^{(q_1^{(p)})}(k) + c(\mathbf{x}^{(q_2^{(p)})}(k) - \mathbf{x}^{(q_3^{(p)})}(k)) \quad (2.44)$$

DE/rand/2

$$\mathbf{v}^{(p)}(k) = \mathbf{x}^{(q_1^{(p)})}(k) + c_1(\mathbf{x}^{(q_2^{(p)})}(k) - \mathbf{x}^{(q_3^{(p)})}(k)) + c_2(\mathbf{x}^{(q_4^{(p)})}(k) - \mathbf{x}^{(q_5^{(p)})}(k)) \quad (2.45)$$

DE/best/1

$$\mathbf{v}^{(p)}(k) = \mathbf{x}^{(\text{best})}(k) + c(\mathbf{x}^{(q_1^{(p)})}(k) - \mathbf{x}^{(q_2^{(p)})}(k)) \quad (2.46)$$

DE/best/2

$$\mathbf{v}^{(p)}(k) = \mathbf{x}^{(\text{best})}(k) + c(\mathbf{x}^{(q_1^{(p)})}(k) + \mathbf{x}^{(q_2^{(p)})}(k) - \mathbf{x}^{(q_3^{(p)})}(k) - \mathbf{x}^{(q_4^{(p)})}(k)) \quad (2.47)$$

DE/rand to best/1

$$\mathbf{v}^{(p)}(k) = \mathbf{x}^{(p)}(k) + c_1(\mathbf{x}^{(\text{best})}(k) - \mathbf{x}^{(p)}(k)) + c_2(\mathbf{x}^{(q_1^{(p)})}(k) - \mathbf{x}^{(q_2^{(p)})}(k)) \quad (2.48)$$

がある。ここで、 $\mathbf{x}^{(\text{best})}(k)$ は、第 k イテレーションにおける全探索点の中での最良探索点で、

$$\mathbf{x}^{(\text{best})}(k) = \underset{\mathbf{x}^{(p)}(k)}{\operatorname{argmin}} \{E(\mathbf{x}^{(p)}(k)) | p = 1, \dots, P\} \quad (2.49)$$

¹ $q_i^{(p)}$ 自体もイテレーション経過とともに変動するので、正確には $q_i^{(p)}(k)$ と記すところであるが、煩雑さを避けるために $q_i^{(p)}$ の引数 (k) は省略する。

である。また、(2.44)~(2.48) 式の差分の部分に掛かる係数 c , c_1 , c_2 は、いずれも正の値をとるパラメータである。

試験点 $\mathbf{u}^{(p)}(k)$ を生成する $\mathbf{x}^{(p)}(k)$ と変異点 $\mathbf{v}^{(p)}(k)$ の一様交叉は、 N 次元ベクトルの少なくともランダムに選択された一つの成分は必ず交叉するもので、一様乱数 $r_n^{(p)}(k) \sim U_R(0, 1)$ と自然乱数 $r_0^{(p)}(k) \sim U_Z(1, N)$ を用いて、

$$\mathbf{u}_n^{(p)}(k) = \begin{cases} \mathbf{v}_n^{(p)}(k), & \text{if } r_n^{(p)}(k) < \theta \text{ or } n = r_0^{(p)}(k) \\ \mathbf{x}_n^{(p)}(k), & \text{otherwise} \end{cases} \quad (2.50)$$

$$n = 1, \dots, N, p = 1, \dots, P \quad (2.51)$$

により、試験点 $\mathbf{u}^{(p)}(k)$ を生成する。ただし、閾値 θ は交叉確率とよばれる。

以上の操作からなる DE の手順は以下のとおりである。

Step 1 最大イテレーション K を設定し、 P 個 (複数個) の探索点 $\mathbf{x}^{(p)}(0)$, $p = 1, \dots, P$ をランダムに与え、 $k = 0$ とおく。

Step 2 突然変異操作により変異点 $\mathbf{v}^{(p)}(k)$, $p = 1, \dots, P$ を生成する。

Step 3 探索点 $\mathbf{x}^{(p)}(k)$ と変異点 $\mathbf{v}^{(p)}(k)$ の一様交叉による試験点 $\mathbf{u}^{(p)}(k)$ の生成を $p = 1, \dots, P$ に対して繰り返す。

Step 4 必要に応じて試験点 $\mathbf{u}^{(p)}(k)$ の位置に、たとえば (2.4) 式や (2.5) 式を用いて制限をかける。

Step 5 試験点 $\mathbf{u}^{(p)}(k)$, $p = 1, \dots, P$ の目的関数値 $E(\mathbf{u}^{(p)}(k))$, $p = 1, \dots, P$ を計算し、

$$\mathbf{x}^{(p)}(k+1) = \begin{cases} \mathbf{u}^{(p)}(k), & \text{if } E(\mathbf{u}^{(p)}(k)) < E(\mathbf{x}^{(p)}(k)) \\ \mathbf{x}^{(p)}(k), & \text{otherwise, } p = 1, \dots, P \end{cases} \quad (2.52)$$

により探索点 $\mathbf{x}^{(p)}(k)$ を $\mathbf{x}^{(p)}(k+1)$ に更新する。

Step 6 $k \geq K$ ならば計算を終了し、そうでなければ、 $k = k+1$ として Step 2 へ戻る。

2.4.2 Differential Evolution のパラメータ調整法

DE の大域的探索性能は、PSO よりも優れているとされている。しかし、変異点生成のための突然変異の (2.44)~(2.48) 式の演算に含まれる係数 c , c_1 , c_2 や、一様交叉の乱数の閾値パラメータ θ がその大域的探索性能に大きく影響する。文献 [53] によると、パラメータの推奨値は $P = 10N$, $c \in [0.5, 1.0]$, $\theta \in [0.8, 1.0]$ 程度とされている。文献 [54] では、最大エントロピー法を用いた分布解析により、DE の探索点の初期収束を防いで持続探索をするためのパラメータの範囲が示されている。しかし、統計的に十分大きな個数の探索点や 1 変数を前提とした解析であり、実用上の課題が残されている。

最近では、DE の探索性能を向上させるため、更新式中のパラメータ c , θ をイテレーション経過とともに変動させる調整法を用いる改良手法が提案されている。それらを分類すると以下のとおりである。

- (1) 探索点の挙動とは関係なく乱数を用いて完全に試行錯誤的にパラメータを揺動させる (文献 [55][56] など).
- (2) 目的関数の改善の成否などの探索点の挙動を、乱数を用いたパラメータの揺動に対して反映させる (文献 [57][58] など).
- (3) 探索点の挙動特性を示す指標の目標とする経時的变化を規定し、それに追従するようにパラメータ調整を行う (文献 [59] など).

まず、(1) の代表例には、文献 [55] で提案された後に jDE と呼称されるようになった手法と、それを改良した自己適応型 DE (Self-Adaptive DE: 以降 SDE と略記する)^[56] とよばれる手法がある。まず、jDE の調整則は、パラメータ c や θ を探索点ごとに用意し、一様乱数 $r_1^{(p)}(k)$, $r_2^{(p)}(k)$, $r_3^{(p)}(k)$, $r_4^{(p)}(k)$ と閾値 θ'_1 , θ'_2 を用いて、

$$c^{(p)}(k+1) = \begin{cases} c_l + c_u \cdot r_1^{(p)}(k), & \text{if } r_2^{(p)}(k) < \theta'_1 \\ c^{(p)}(k), & \text{otherwise} \end{cases} \quad (2.53)$$

$$\theta^{(p)}(k+1) = \begin{cases} r_3^{(p)}(k), & \text{if } r_4^{(p)}(k) < \theta'_2 \\ \theta^{(p)}(k), & \text{otherwise} \end{cases} \quad (2.54)$$

と経時的に変動させる手法である。イテレーション過程において、 θ'_1 の頻度で区間 $[c_l, c_u]$ の一様乱数 $r_1^{(p)}(k)$ による揺動により $c^{(p)}(k+1)$ を与え、 $(1 - \theta'_1)$ 頻度で前イテレーションの値を引き継ぐというもので、 $\theta^{(p)}(k+1)$ についても同様である。探索点の状態をまったく考慮しないきわめて単純な調整則であるが、それらのパラメータに上付き添字 (p) が付されているように、探索点ごとに調整される。これらのパラメータの値については、文献 [55] においては $c_l = 0.1$, $c_u = 0.9$, $\theta_1 = \theta_2 = 0.1$ が使用されている。

これに対して SDE は、係数パラメータ $c^{(p)}$ の方を Cauchy 分布 $C(\mu, \delta^{(p)})$ で与え、さらにその Cauchy 分布のスケールパラメータ $\delta^{(p)}$ を探索点ごとに一様乱数を用いて、

$$\delta^{(p)}(k+1) = \delta_l + \delta_u \cdot r_1^{(p)}(k) \quad (2.55)$$

$$c^{(p)}(k+1) = \begin{cases} C(\mu, \delta^{(p)}(k+1)), & \text{if } r_2^{(p)}(k) < \theta'_1 \\ C(\mu, \delta^{(p)}(k)), & \text{otherwise} \end{cases} \quad (2.56)$$

$$\theta^{(p)}(k+1) = \begin{cases} r_3^{(p)}(k), & \text{if } r_4^{(p)}(k) < \theta'_2 \\ \theta^{(p)}(k), & \text{otherwise} \end{cases} \quad (2.57)$$

と与える手法で、そのパラメータ調整の設計思想は jDE と同じである。文献 [56] では、パラメータとして $\mu = 0.5, 0.8, 0.9$ の 3 種類、 $\delta_l = 0.1$, $\delta_u = 0.9$, $\theta'_1 = 0.1$, $\theta'_2 = 0.1$ を使用している。

以上の手法は DE/rand/1 の変異点生成公式を用い、探索点の挙動とは無関係にパラメータを揺動させて調整する手法であるが、探索点の挙動に応じてパラメータを揺動させる (2) に分類される手法として、(2.44)~(2.48) 式とは別の変異点生成公式を用いて改良した JADE^[57] と呼ばれる手法がある。この手法では、過去の探索点や、(2.52) 式において $E(\mathbf{u}^{(p)}(k)) < E(\mathbf{x}^{(p)}(k))$ となり目的関数値を改善する試験点 $\mathbf{u}^{(p)}(k)$ を生成したパラメー

タ c , θ が記憶され, 新たなパラメータと探索点の生成に使われる. この手法における変異点生成式は,

$$\mathbf{v}^{(p)}(k) = \mathbf{x}^{(p)}(k) + c^{(p)}(k)(\mathbf{x}^{(\text{best}_\rho)}(k) - \mathbf{x}^{(p)}(k)) + c^{(p)}(k)(\mathbf{x}^{(q_1^{(p)})}(k) - \mathbf{x}^{(q_2^{(p)})}(k)) \quad (2.58)$$

である. ここで, $\mathbf{x}^{(\text{best}_\rho)}(k)$ は第 k イテレーションにおける目的関数値の小さい順の上位 100 ρ % からランダムに選ばれた探索点, $\mathbf{x}^{(q_1^{(p)})}(k)$ は第 k イテレーションにおいてランダムに選ばれた $\mathbf{x}^{(p)}(k)$ と異なる探索点で, $\mathbf{x}^{(q_2^{(p)})}(k)$ は探索履歴の中で優秀とみなされた探索点の集合 $S_x(k)$ からランダムに選ばれた $\mathbf{x}^{(p)}(k)$ や $\mathbf{x}^{(q_1^{(p)})}(k)$ と異なる探索点である. また, パラメータ c , θ の更新則は, それぞれ Cauchy 分布と正規分布を用いて

$$\begin{aligned} c^{(p)}(k) &\sim C^{(p)}(\mu_c(k), 0.1) \\ \theta^{(p)}(k) &\sim N^{(p)}(\mu_\theta(k), 0.1) \end{aligned} \quad (2.59)$$

とランダムに選ばれるが, それらの分布の平均値はそれぞれ

$$\begin{aligned} \mu_c(k+1) &= (1 - \epsilon)\mu_c(k) + \epsilon \cdot \text{mean}_L(S_c(k)) \\ \mu_\theta(k+1) &= (1 - \epsilon)\mu_\theta(k) + \epsilon \cdot \text{mean}(S_\theta(k)) \end{aligned} \quad (2.60)$$

で更新される. 集合 $S_x(k)$, $S_c(k)$, $S_\theta(k)$ については, (2.52) 式において目的関数値を改善する試験点 $\mathbf{u}^{(p)}(k)$ が採用された場合, その試験点生成に用いられた探索点 $\mathbf{x}^{(p)}(k)$ を集合 $S_x(k)$ に記憶し, またその変位点 $\mathbf{v}^{(p)}(k)$ を与えた (2.58) 式の $c^{(p)}$ を集合 S_c に, (2.57) 式の $\theta^{(p)}$ を集合 S_θ に加える. これらを用いて, (2.59) 式によりパラメータを更新する. また, それらは (2.60) 式により決まるパラメータにより, 徐々に低い値を取る. なお, mean は算術平均であるのに対し, mean_L は非調和平均と呼ばれるもので,

$$\text{mean}_L(S_c) = \sum_{c \in S_c} c^2 / \sum_{c \in S_c} c \quad (2.61)$$

によって計算される. 非調和平均は算術平均より大きな値を取るため, アルゴリズムの収束に寄与するとされている. Step 2 においてはまず (2.59) 式によりパラメータ $c^{(p)}$, $\theta^{(p)}$ を更新し, Step 5 において (2.60) 式により Cauchy 分布や正規分布の平均値のパラメータ μ_c , μ_θ を更新する. なお, 過去の優秀な探索点の集合 S_x の要素数が, 探索点数 P より大きくなったら, ランダムに一つ削除する操作が行われる. パラメータの推奨値は, $\mu_c(0) = 0.5$, $\mu_\theta(0) = 0.5$, $1/\epsilon \in [5, 20]$, $\rho \in [0.05, 0.2]$ とされている. 実装上の注意点として, (2.59) 式により生成されるパラメータ $c^{(p)}(k)$ は稀に大きな値を取るため, $c^{(p)}(k)$ が 1.0 を超えた場合は $c^{(p)}(k) = 1.0$ とし, 負の値を取った場合は再度乱数を割り当てる.

(2) に分類される別の手法として, 試験点の目的関数値の評価による探索点更新の成否を考慮したパラメータ調整を jDE の調整則に付加する手法として, Adapt-DE がある. この調整則は,

$$c^{(p)}(k+1) = \begin{cases} r_1^{(p)}(k), & \text{if } E(\mathbf{u}^{(p)}(k)) > E(\mathbf{x}^{(p)}(k)) \text{ and } r_2^{(p)}(k) < \theta'_1 \\ c^{(p)}(k), & \text{otherwise} \end{cases} \quad (2.62)$$

$$\theta^{(p)}(k+1) = \begin{cases} r_3^{(p)}(k), & \text{if } E(\mathbf{u}^{(p)}(k)) > E(\mathbf{x}^{(p)}(k)) \text{ and } r_4^{(p)}(k) < \theta'_2 \\ \theta^{(p)}(k), & \text{otherwise} \end{cases} \quad (2.63)$$

であり、試験点 $\mathbf{u}^{(p)}(k)$ が探索点 $\mathbf{x}^{(p)}(k)$ より目的関数値を改善しない場合に、一定の頻度 θ'_1 や θ'_2 でパラメータ $c^{(p)}$ や $\theta^{(p)}$ を選びなおす調整則である。しかしながら、目的関数値の改善を考慮するにしても、探索点の状態に応じてより能動的にパラメータを調整するような機能はないといえる。

また、探索点の挙動に応じてより能動的にパラメータ調整を行う (3) に類する手法として、適応的パラメータ調整型 DE (Adaptive Parameter Adjustment of Differential Evolution: APADE)^[59] と称されている手法を示しておく。まず、(2.44) 式の変異点生成式のパラメータ c の更新のために、探索点の挙動特性を示す指標として、 $\mathbf{x}^{(\text{best})}(k)$ を除くすべての探索点間の距離の平均値

$$L_{\text{group}}(k) = \left(\sum_{i, i \neq (\text{best})}^{P-1} \sum_{j > i, j \neq (\text{best})}^P \|\mathbf{x}^{(i)}(k) - \mathbf{x}^{(j)}(k)\| \right) / \left(\frac{1}{2} P(P-1) \right) \quad (2.64)$$

を用いた探索点の群れの密集状態の指標として

$$I_{\text{di}}(k+1) = c(k) \cdot L_{\text{group}}(k+1) \quad (2.65)$$

を考える。(2.64) 式は $\mathbf{x}^{(\text{best})}(k)$ を除いた $P-1$ 個の探索点間の平均距離である。これを用いた探索状態 I_{di} の目標 I_{target} を最終イテレーション K まで

$$I_{\text{target}}(k+1) = I_{\text{target}}(k) \left(\frac{I_{\text{target}}(K)}{I_{\text{target}}(0)} \right)^{1/K} = I_{\text{target}}(0) \left(\frac{I_{\text{target}}(K)}{I_{\text{target}}(0)} \right)^{k+1/K} \quad (2.66)$$

と等比的に減少するよう規定し、探索点群が経時的にこの目標に可能な限り追従するよう、DE/best/1 による変異点生成のパラメータ調整を行う。具体的には、指標 I_{di} が目標値 I_{target} を下回ったら探索点の多様化のためにパラメータ c を増加させ、そうでなければ探索点の集中化のため、パラメータ c を減少させるように

$$c(k+1) = \begin{cases} \min(c(k) + \Delta c, c_{\text{max}}), & \text{if } I_{\text{target}}(k+1) > I_{\text{di}}(k+1) \\ \max(c(k) - \Delta c, c_{\text{min}}), & \text{otherwise} \end{cases} \quad (2.67)$$

と調整する手法である。つぎに、交叉確率 θ の調節については別のパラメータ C_{max} , $\theta_{C_{\text{max}}}$ や $P(k)$ に依存させて行う。 $\theta_{C_{\text{max}}}$ の調整のため、探索点の群れとしての目的関数値の改善度合いを

$$C(k+1) = \frac{\sum_{p=1}^P E(\mathbf{x}^{(p)}(k)) - \sum_{p=1}^P E(\mathbf{x}^{(p)}(k+1))}{|\sum_{p=1}^P E(\mathbf{x}^{(p)}(k+1))|} \quad (2.68)$$

と定義し、 $C(k+1)$ が $C_{\text{max}}(k)$ より大きい時に

$$C_{\text{max}}(k+1) = C(k+1), \quad \text{if } C(k+1) > C_{\text{max}}(k) \quad (2.69)$$

$$\theta_{C_{\text{max}}}(k+1) = \theta(k+1), \quad \text{if } C(k+1) > C_{\text{max}}(k) \quad (2.70)$$

により更新する。 $P(k)$ の調整は $I_{\text{di}}(k+1) = c(k)L_{\text{group}}(k+1)$ が $c(k)$ を更新することにより $c(k+1)L_{\text{group}}(k+1)$ が $I_{\text{target}}(k+1)$ に追従できたなら $P(k+1) = 1$ とし、それ以外の場合は $P(k+1) = P(k) + 1$ とする。 $P(k) \leq 2$ の場合は

$$\theta(k+1) = \begin{cases} \min(\theta_{C_{\text{max}}} + s(k) \cdot \Delta\theta, \theta_{\text{max}}), & \text{if } I_{\text{target}}(k) > I_{\text{di}} \\ \max(\theta_{C_{\text{max}}} + s(k) \cdot \Delta\theta, \theta_{\text{min}}), & \text{otherwise} \end{cases} \quad (2.71)$$

とし, $P(k) > 2$ の場合は

$$\theta(k+1) = \begin{cases} \min(\theta(k) + t(k) \cdot \Delta\theta, \theta_{\max}), & \text{if } I_{\text{target}}(k) > I_{\text{di}} \\ \max(\theta(k) + t(k) \cdot \Delta\theta, \theta_{\min}), & \text{otherwise} \end{cases} \quad (2.72)$$

とする. ただし $s(k)$ と $t(k)$ の更新式は省略する. これらの調整は Step 5 において付加的に行われる.

2.4.3 計算機実験による DE のパラメータ調整法の課題提起

PSO や ES と同様に

1. 大域的な探索が必要となる 2^N -minima 関数
2. POP で多峰性の Rastrigin 関数
3. 単峰性で変数間依存性のある Rosenbrock 関数
4. 探索持続性の指標となる Fractal 関数
5. 1. の座標系を回転させた関数
6. 2. の座標系を回転させた関数
7. 4. の座標系を回転させた関数

の七つの異なる目的関数に対して

1. (2.44) 式と $c = 0.5$, $\theta = 0.9$ のパラメータを用いた DE
2. 推奨値のパラメータを用いた jDE
3. 推奨値のパラメータを用いた SDE
4. 推奨値のパラメータを用いた AdaptDE
5. $\rho = 0.1$, $\epsilon = 0.05$ のパラメータを用いた JADE
6. 推奨値のパラメータを用いた適応的パラメータ調整型 DE

の 6 種類のアロリズムを適用し, $P = 20$, $K = 1,000$ の設定で, 指定がないパラメータ以外は推奨パラメータを用いてそれぞれ 50 回試行した. 結果の中央値 (median), 平均 (mean), 標準偏差 (std), 最小値 (min), 最大値 (max) は目的関数ごとに Table 2.31~2.37 に示す. 性能比較の基準として中央値を考えた際の検定として, アロリズムのペアによりランク化 t 検定を行った結果は目的関数ごとに Table 2.38~2.44 に示す. これらの Table においては有意水準を 5% とし, Bonferroni 法を用いて P 値が $0.05 / ({}_6C_2 \times 7)$ より小さければ有意であるとし, マスを網掛けにしてある. 検定結果のまとめは Table 2.45 に示す. alg1 と alg2 を比較して alg1 が統計的に優秀であると言えれば L (Left), alg2 が優秀であると言

えれば R (Right), 判断がつかない場合は N (Neutral) としてある. 最後の2列にはこれらの L の数と R の数を示してある.

Table 2.45 を使ってこれらのシミュレーションをまとめると, もっとも多くの問題において良い結果を出したのは jDE であるといえる. また, 特に Fractal 関数に関してはパラメータをフィードバックさせる改良手法よりも, パラメータを揺動させる改良手法において改善効果があったことが確認でき, この結果は問題のスケールに依存しないパラメータの揺動の方が, Fractal 関数に対しては合理的であると考えられることに一致する.

Table 2.31: Results for Rastrigin func. with 20,000 function calls

alg	median	mean	std	min	max
Adapt-DE	0.99	0.74	0.92	0.00	3.98
APADE	0.50	0.82	1.08	0.00	3.98
DE	3.87	4.16	4.99	0.99	13.02
JADE	0.00	0.02	0.01	0.00	0.40
jDE	0.00	0.08	0.07	0.00	0.99
SDE	0.00	0.32	0.30	0.00	1.99

Table 2.32: Results for 2^N -minima func. with 20,000 function calls

alg	median	mean	std	min	max
Adapt-DE	-783.32	-777.67	130.51	-783.32	-755.05
APADE	-783.32	-782.19	31.32	-783.32	-755.05
DE	-755.05	-763.38	494.44	-783.32	-698.50
JADE	-783.32	-782.19	31.32	-783.32	-755.05
jDE	-783.32	-778.80	142.26	-783.32	-726.78
SDE	-783.32	-781.63	46.01	-783.32	-755.05

Table 2.33: Results for Rosenbrock func. with 20,000 function calls

alg	median	mean	std	min	max
Adapt-DE	2.02	2.14	1.84	0.00	5.70
APADE	4.36	4.03	2.40	0.21	6.95
DE	7.64	8.63	52.36	0.00	40.57
JADE	2.45	2.55	1.09	0.36	5.12
jDE	1.26	1.64	1.76	0.00	6.65
SDE	1.44	1.54	0.87	0.01	6.15

Table 2.34: Results for Fractal func. with 20,000 function calls

alg	median	mean	std	min	max
Adapt-DE	-30.00	-29.18	18.19	-30.00	-3.00
APADE	-9.00	-9.92	8.69	-24.00	-6.00
DE	-9.50	-11.70	87.52	-30.00	0.00
JADE	-30.00	-29.64	6.48	-30.00	-12.00
jDE	-30.00	-29.62	7.22	-30.00	-11.00
SDE	-23.00	-23.32	0.67	-25.00	-21.00

Table 2.35: Results for Rastrigin (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
Adapt-DE	9.95	10.09	18.75	0.99	19.90
APADE	13.97	14.04	36.43	3.00	28.86
DE	9.04	10.44	36.28	1.99	28.27
JADE	20.16	20.11	19.10	4.54	29.39
jDE	11.12	12.13	38.47	3.03	27.87
SDE	10.51	11.01	26.65	3.98	24.03

Table 2.36: Results for 2^N -minima (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
Adapt-DE	-755.05	-648.15	155296.96	-783.32	1104.34
APADE	-755.04	-656.48	121674.17	-783.32	875.58
DE	-755.03	-747.29	331.69	-783.32	-698.48
JADE	-755.05	-751.66	86.14	-755.05	-726.78
jDE	-755.05	-719.78	40491.25	-783.32	669.27
SDE	-755.05	-718.31	39905.04	-783.32	657.91

Table 2.37: Results for Fractal (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
Adapt-DE	-21.00	-17.94	40.51	-25.00	-3.00
APADE	-5.00	-5.06	1.24	-8.00	-4.00
DE	-6.00	-8.22	34.34	-24.00	-1.00
JADE	-20.00	-20.32	1.20	-23.00	-18.00
jDE	-22.00	-21.00	12.73	-25.00	-4.00
SDE	-13.00	-12.88	1.37	-15.00	-8.00

Table 2.38: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rastrigin func. with 20,000 function calls

alg1	alg2	t-value	P -value
Adapt-DE	APADE	-3.59	5.19e-04
Adapt-DE	DE	-14.23	1.41e-25
Adapt-DE	JADE	1.66	1.02e-01
Adapt-DE	jDE	5.50	5.56e-07
Adapt-DE	SDE	2.61	1.05e-02
APADE	DE	-12.87	8.82e-23
APADE	JADE	6.98	3.50e-10
APADE	jDE	15.24	2.85e-27
APADE	SDE	7.78	8.37e-12
DE	JADE	17.27	1.72e-31
DE	jDE	21.75	6.52e-33
DE	SDE	18.29	1.43e-32
JADE	jDE	6.62	2.38e-09
JADE	SDE	2.11	3.76e-02
jDE	SDE	-2.71	8.27e-03

Table 2.39: Pairwise ranked Welch's t -test on algorithms for 10 dimensional 2^N -minima func. with 20,000 function calls

alg1	alg2	t-value	P -value
Adapt-DE	APADE	-5.72	1.62e-07
Adapt-DE	DE	-9.35	3.42e-15
Adapt-DE	JADE	2.51	1.42e-02
Adapt-DE	jDE	0.74	4.62e-01
Adapt-DE	SDE	2.11	3.83e-02
APADE	DE	-8.21	1.44e-12
APADE	JADE	12.79	4.96e-22
APADE	jDE	7.51	3.98e-11
APADE	SDE	11.32	1.86e-19
DE	JADE	15.37	1.59e-24
DE	jDE	10.49	1.20e-17
DE	SDE	14.20	1.24e-23
JADE	jDE	-1.77	8.06e-02
JADE	SDE	-0.45	6.50e-01
jDE	SDE	1.35	1.80e-01

Table 2.40: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rosenbrock func. with 20,000 function calls

alg1	alg2	t-value	P -value
Adapt-DE	APADE	-6.58	2.40e-09
Adapt-DE	DE	-12.18	4.45e-21
Adapt-DE	JADE	-2.00	4.82e-02
Adapt-DE	jDE	2.57	1.16e-02
Adapt-DE	SDE	2.98	3.67e-03
APADE	DE	-8.48	4.65e-13
APADE	JADE	5.84	7.94e-08
APADE	jDE	8.39	3.86e-13
APADE	SDE	9.17	1.10e-14
DE	JADE	11.60	1.21e-19
DE	jDE	12.90	1.14e-22
DE	SDE	13.54	6.75e-24
JADE	jDE	4.90	4.05e-06
JADE	SDE	6.08	2.37e-08
jDE	SDE	-0.56	5.75e-01

Table 2.41: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Fractal func. with 20,000 function calls

alg1	alg2	t-value	P -value
Adapt-DE	APADE	-18.58	5.31e-33
Adapt-DE	DE	-13.54	1.13e-21
Adapt-DE	JADE	0.58	5.63e-01
Adapt-DE	jDE	0.58	5.63e-01
Adapt-DE	SDE	-16.59	6.21e-30
APADE	DE	-0.59	5.59e-01
APADE	JADE	23.38	5.07e-31
APADE	jDE	22.67	2.44e-31
APADE	SDE	16.19	8.12e-29
DE	JADE	15.28	1.08e-21
DE	jDE	15.21	1.05e-21
DE	SDE	7.13	8.88e-10
JADE	jDE	-0.01	9.89e-01
JADE	SDE	-20.41	1.94e-36
jDE	SDE	-20.41	1.94e-36

Table 2.42: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Rastrigin (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
Adapt-DE	APADE	-3.99	1.25e-04
Adapt-DE	DE	0.16	8.75e-01
Adapt-DE	JADE	-12.02	6.14e-21
Adapt-DE	jDE	-1.46	1.47e-01
Adapt-DE	SDE	-0.87	3.88e-01
APADE	DE	3.22	1.76e-03
APADE	JADE	-6.21	1.41e-08
APADE	jDE	1.71	9.00e-02
APADE	SDE	3.00	3.39e-03
DE	JADE	-9.16	1.22e-14
DE	jDE	-1.52	1.31e-01
DE	SDE	-0.82	4.15e-01
JADE	jDE	7.51	3.27e-11
JADE	SDE	9.42	2.42e-15
jDE	SDE	0.71	4.78e-01

Table 2.43: Pairwise ranked Welch's t -test on algorithms for 10 dimensional 2^N -minima (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
Adapt-DE	APADE	-3.36	1.11e-03
Adapt-DE	DE	-2.82	5.78e-03
Adapt-DE	JADE	1.57	1.19e-01
Adapt-DE	jDE	0.63	5.30e-01
Adapt-DE	SDE	0.37	7.15e-01
APADE	DE	0.72	4.76e-01
APADE	JADE	6.38	1.32e-08
APADE	jDE	4.36	3.23e-05
APADE	SDE	3.88	1.88e-04
DE	JADE	6.25	1.35e-08
DE	jDE	3.60	5.00e-04
DE	SDE	3.01	3.31e-03
JADE	jDE	-1.00	3.19e-01
JADE	SDE	-1.26	2.10e-01
jDE	SDE	-0.26	7.93e-01

Table 2.44: Pairwise ranked Welch's t -test on algorithms for 10 dimensional Fractal (rotated) func. with 20,000 function calls

alg1	alg2	t-value	P -value
Adapt-DE	APADE	-10.41	1.18e-16
Adapt-DE	DE	-7.38	5.42e-11
Adapt-DE	JADE	-0.37	7.15e-01
Adapt-DE	jDE	2.71	7.90e-03
Adapt-DE	SDE	-6.16	4.35e-08
APADE	DE	2.47	1.60e-02
APADE	JADE	18.10	5.23e-33
APADE	jDE	15.32	2.01e-27
APADE	SDE	18.10	5.02e-33
DE	JADE	11.27	1.21e-18
DE	jDE	11.39	1.33e-19
DE	SDE	7.10	6.52e-10
JADE	jDE	4.77	6.76e-06
JADE	SDE	-18.06	6.05e-33
jDE	SDE	-13.40	7.00e-23

Table 2.45: Pairwise ranked Welch's t -test for 10 dimensional functions with 20,000 function calls

alg1	alg2	Rastrigin	2^N -minima	Rosenbrock	Fractal	Rastrigin (Rot.)	2^N -minima (Rot.)	Fractal (Rot.)	L	R
Adapt-DE	APADE	N	L	L	L	L	N	N	L	5
Adapt-DE	DE	L	L	L	L	N	N	N	L	5
Adapt-DE	JADE	N	N	N	N	L	N	N	N	1
Adapt-DE	jDE	R	N	N	N	N	N	N	N	0
Adapt-DE	SDE	N	N	N	L	N	N	L	L	2
APADE	DE	L	L	L	N	N	N	N	N	3
APADE	JADE	R	R	R	R	L	R	R	R	1
APADE	jDE	R	R	R	R	N	R	R	R	0
APADE	SDE	R	R	R	R	N	R	R	R	0
DE	JADE	R	R	R	R	L	R	R	R	1
DE	jDE	R	R	R	R	N	N	R	R	0
DE	SDE	R	R	R	R	N	N	R	R	0
JADE	jDE	R	N	R	N	R	N	R	R	0
JADE	SDE	N	N	R	L	R	N	L	L	2
jDE	SDE	N	N	N	L	N	N	L	L	2

2.5 Firefly Algorithm の原理とそのパラメータ調整の課題

2.5.1 Firefly Algorithm の原理

Firefly Algorithm (以降 FA と略記) はホタルの求愛行動を参考に Yang^{[41][60]} によって最初に提案されたアルゴリズムで、第 k イテレーションにおける第 p 探索点の位置座標 $\mathbf{x}^{(p)}(k)$ の更新式は、

$$x_n^{(p)}(k+1) = x_n^{(p)}(k) + v_n^{(p)}(k+1), \quad n = 1, \dots, N \quad (2.73a)$$

$$v_n^{(p)}(k+1) = \beta \exp(-\gamma \|\mathbf{x}^{(q)}(k) - \mathbf{x}^{(p)}(k)\|) (x_n^{(q)}(k) - x_n^{(p)}(k)) + \alpha(k) r_n(k), \quad (2.73b)$$

if $E(\mathbf{x}^{(q)}(k)) < E(\mathbf{x}^{(p)}(k))$, $n = 1, \dots, N$

で与えられる。ただし、 $\mathbf{x} \in R^N$ とし、括弧付きの上付き添字 (p), (q) は探索点番号を表し、 $p \neq q$ とする。 $r_n(k)$ は乱数変数で、 $\alpha(k)$ がその変動幅を決める時変係数パラメータであるのに対し、 β , γ は一定のパラメータである。 $\alpha(k)$ は一定とする場合もあるが、たとえば、

$$\alpha(k) = \alpha(0) \delta^{k/K} \quad (2.74)$$

として、イテレーション経過とともに減少させるのも提案されている。また、乱数 $r_n(k)$ に関しては、区間 $[-0.5, 0.5]$ の一様乱数 $U_R(-0.5, 0.5)$ を用いる場合が比較的多いが、正規乱数 $N(0, 1)$ を用いる場合もある。

ところで、(2.73b) 式の右辺の第 1 項の $\beta \exp(-\gamma \|\mathbf{x}^{(q)}(k) - \mathbf{x}^{(p)}(k)\|)$ 自体が、移流項 $(x_n^{(q)}(k) - x_n^{(p)}(k))$ に掛かる時変係数で、更新対象の探索点 $\mathbf{x}^{(p)}(k)$ と移動目標点 $\mathbf{x}^{(q)}(k)$ との距離に応じて移動幅を指数関数で調整する調整則とみなすことができるが、さらにこの調整則の中にパラメータ β , γ を含むため、それらの設定が必ずしも容易ではない。とくに γ に関しては、 $\gamma = 1.0$ としている場合が多い。

これらをまとめた FA の基本的な更新手順は次の通りである。

Step 1 ファンクションコールやプログラム実行時間等に関する終了条件、最大イテレーション K を設定し、 P 個 (複数個) の探索点 $\mathbf{x}^{(p)}(1)$, $p = 1, \dots, P$ をランダムに与え、 $k = 1$ とおく。

Step 2 更新対象の探索点 $\mathbf{x}^{(p)}(k)$ の番号を $p = \lceil k/P \rceil$ とおく。

Step 3 比較対象の探索点 $\mathbf{x}^{(q)}(k)$ の番号を $q = ((k-1) \bmod P) + 1$ とおく。

Step 4 もし $E(\mathbf{x}^{(q)}(k)) < E(\mathbf{x}^{(p)}(k))$ ならば、(2.73b)~(2.76) 式のいずれかにより変位 $v^{(p)}(k+1)$ を計算し、更新対象の探索点の新しい位置 $\mathbf{x}^{(p)}(k+1)$ を (2.73a) 式より求めて、目的関数値 $E(\mathbf{x}^{(p)}(k+1))$ を計算する。

Step 5 $k \geq K$ ならば計算を終了し、そうでなければ、 $k = k+1$ として Step 2 へ戻る。

なお、探索点数 P 個分に相当する P 回のイテレーションの更新期間中、Step 2 で選ばれる更新対象の探索点 $\mathbf{x}^{(p)}(k)$ の番号 p は、切り上げ演算 $\lceil \cdot \rceil$ によって一定のまま、比較対象の探索点 $\mathbf{x}^{(q)}(k)$ の番号 q のみがイテレーションの更新とともに、Step 3 での mod 演

算によって若い番号順に選ばれ、目的関数値の比較が行われる。このとき、番号 q が番号 p と重複する場合も許容されるが、この場合の (2.73b) 式では、乱数変数 $\alpha(k)r_n(k)$ の項のみによる駆動となる。

2.5.2 改良型 Firefly Algorithm

FA は最初から更新式にパラメータ調整則が含まれているアルゴリズムであるとも考えることができ、多くの派生型 FA もその調整則を継承している。また、Yang が考案した更新則を改良した手法がいくつか提案されている。その手法の一つとして更新則における変位が、

$$\begin{aligned}
 v_n^{(p)}(k+1) &= \beta \exp(-\gamma \|\mathbf{x}^{(q)}(k) - \mathbf{x}^{(p)}(k)\|) (x_n^{(q)}(k) - x_n^{(p)}(k)) \\
 &\quad + \beta \exp(-\gamma \|\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k)\|) (x_n^{(g\text{-best})}(k) - x_n^{(p)}(k)) \\
 &\quad + \alpha(k)r_n(k), \\
 &\quad \text{if } E(\mathbf{x}^{(q)}(k)) < E(\mathbf{x}^{(p)}(k)), n = 1, \dots, N \\
 \mathbf{x}^{(g\text{-best})}(k) &= \arg \min \{E(\mathbf{x}^{(p)}(i)) \mid p = 1, \dots, P, i = 1, \dots, k\}
 \end{aligned} \tag{2.75}$$

で与えられる改良手法がある^[61]。これは、各イテレーション時点での探索点群の中の最良点 $\mathbf{x}^{(g\text{-best})}(k)$ を目標点とする項を付加するものである。さらに、現時点での最良点 $\mathbf{x}^{(c\text{-best})}(k)$ を目標点とする項を付加し、改良した

$$\begin{aligned}
 v_n^{(p)}(k+1) &= \beta \exp(-\gamma \|\mathbf{x}^{(q)}(k) - \mathbf{x}^{(p)}(k)\|) (x_n^{(q)}(k) - x_n^{(p)}(k)) \\
 &\quad + \beta \exp(-\gamma \|\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k)\|) (x_n^{(g\text{-best})}(k) - x_n^{(p)}(k)) \\
 &\quad + \lambda r_n(k) (x_n^{(p)}(k) - x_n^{(c\text{-best})}(k)) \\
 &\quad + \alpha(k)r_n(k), \\
 &\quad \text{if } E(\mathbf{x}^{(q)}(k)) \leq E(\mathbf{x}^{(p)}(k)), n = 1, \dots, N \\
 \mathbf{x}^{(c\text{-best})}(k) &= \arg \min \{E(\mathbf{x}^{(p)}(k)) \mid p = 1, \dots, P\}
 \end{aligned} \tag{2.76}$$

も提案されており^[62]、同じく $r_i() \sim U_R(-0.5, 0.5)$ とされている。

2.5.3 計算機実験による FA のパラメータ調整法の課題提起

これまでに取り上げた手法と同様に

1. 大域的な探索が必要となる 2^N -minima 関数
2. POP で多峰性の Rastrigin 関数
3. 単峰性で変数間依存性のある Rosenbrock 関数
4. 探索持続性の指標となる Fractal 関数
5. 1. の座標系を回転させた関数

6. 2. の座標系を回転させた関数

7. 4. の座標系を回転させた関数

の七つの異なる目的関数に対して

1. 推奨値のパラメータを用いた FA

2. 推奨値のパラメータを用いた (2.76) 式の改良型 FA(Modified Firefly Algorithm: MFA)

の2種類のアプローチを適用し、 $P = 20$, $K = 1,000$ の設定で、指定がないパラメータ以外は推奨パラメータを用いてそれぞれ50回試行した。パラメータは、Rosenbrock 関数においていくらかの試行を行い、FA で $\alpha(k) = 0.25\exp(-5.0t(k))$, $\beta = 0.5$, $\gamma = 2.4/\sqrt{N}$, MFA で $\alpha(k) = 0.01\exp(-5.0t(k))$, $\beta = 0.02$, $\gamma = 16.0/\sqrt{N}$, $\lambda = 2.5$ とした。

結果の中央値 (median), 平均 (mean), 標準偏差 (std), 最小値 (min), 最大値 (max) は目的関数ごとに Table 2.46~2.51 に示す。性能比較の基準として中央値を考えた際の検定として、アルゴリズムのペアによりランク化 t 検定を行った結果は目的関数ごとに Table 2.53 に示す。これらの Table においては有意水準を5%とし、Bonferroni 法を用いて P 値が $0.05/(2C_2 \times 7)$ より小さければ有意であるとし、マスを網掛けにしてある。検定結果のまとめは Table 2.54 に示す。alg1 と alg2 を比較して alg1 が統計的に優秀であると言えれば L (Left), alg2 が優秀であると言えれば R (Right), 判断がつかない場合は N (Neutral) としてある。最後の2列にはこれらの L の数と R の数を示す。

Table 2.54 を使ってこれらのシミュレーションをまとめると、改良型 FA よりも従来の FA の方が良い結果となっている。ただし、この結果は Rosenbrock 関数に対して良い結果となるパラメータを設定したことにより、他の多くのテスト関数に対して MFA の汎用性が劣る結果となったと考えられる。このように、(2.76) 式では新たなパラメータ λ を増やし、右辺の項も増やして自由度をあげることで特定の問題に対する性能向上をもたらした一方、他の問題に対しては汎用性が失われる場合があることが確認される。

Table 2.46: Results for Fractal func. with 20,000 function calls

alg	median	mean	std	min	max
FA	-4.00	-4.30	0.21	-5.00	-4.00
MFA	-1.00	-2.46	9.68	-9.00	0.00

Table 2.47: Results for Fractal (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
FA	-4.00	-4.40	0.24	-5.00	-4.00
MFA	0.00	-0.24	0.19	-1.00	0.00

Table 2.48: Results for 2^N -minima func. with 20,000 function calls

alg	median	mean	std	min	max
FA	-755.00	-739.74	953.17	-783.29	-670.17
MFA	-726.78	-721.12	1239.87	-783.32	-641.96

Table 2.49: Results for 2^N -minima (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
FA	-755.01	-752.75	60.03	-755.04	-726.73
MFA	-726.78	-716.02	1504.18	-783.32	-613.10

Table 2.50: Results for Rastrigin func. with 20,000 function calls

alg	median	mean	std	min	max
FA	7.24	8.15	13.76	3.13	20.24
MFA	23.38	22.88	57.62	7.96	42.78

Table 2.51: Results for Rastrigin (rotated) func. with 20,000 function calls

alg	median	mean	std	min	max
FA	7.19	7.40	9.54	2.29	15.17
MFA	32.83	34.70	159.14	10.94	60.69

Table 2.52: Results for Rosenbrock func. with 20,000 function calls

alg	median	mean	std	min	max
FA	7.70	7.73	0.52	6.21	9.53
MFA	3.11	4.49	96.91	0.12	71.07

Table 2.53: Pairwise ranked Welch's t -test on algorithms for 10 dimensional functions with 20,000 function calls

obj. func.	alg1	alg2	t-value	P -value
Fractal	FA	MFA	-5.75	3.04e-07
Fractal (rotated)	FA	MFA	-21.54	1.39e-38
2^N -minima	FA	MFA	-0.99	3.27e-01
2^N -minima (rotated)	FA	MFA	-3.26	1.71e-03
Rastrigin	FA	MFA	-12.69	2.29e-22
Rastrigin (rotated)	FA	MFA	-16.32	1.07e-29
Rosenbrock	FA	MFA	10.35	1.43e-16

Table 2.54: Pairwise ranked Welch's t -test for 10 dimensional functions with 20,000 function calls

alg1	alg2	Fractal	Fractal (rot.)	2^N -minima	2^N -minima (rot.)	Rastrigin	Rastrigin (rot.)	Rosenbrock	L	R
FA	MFA	L	L	N	L	L	L	R	5	1

第3章 最適化アルゴリズムの パラメータ調整のためのメタ最適化

3.1 パラメータ調整法のメタ最適化

前章において、複数の探索点を用いるいわゆる確率的多点型最適化手法であるメタヒューリスティクスとして、PSO, ES, DE, FA を代表的な手法として取り上げ、さらにそのアルゴリズムの確率的・統計的探索性能を支配するパラメータ調整について、その重要性と課題を検討した。メタヒューリスティクスのアルゴリズム自体が、生物の進化や挙動、あるいは自然現象の模倣により試行錯誤的に構築されているうえに、その探索性能を支配するパラメータの調整法についても、やはり試行錯誤的に様々なものが提案されていることを述べた。とくに手法ごとにパラメータ調整法を分類したうえで、それらの代表的なパラメータ調整法の計算性能を統計的検定によって比較した。この結果、イテレーションの経過とともに、探索点の状態や挙動に応じてパラメータを動的に調整する「フィードバック型調整法」が一般的に高い計算性能と、解きたい問題に対して汎用性があることを確認した。

しかしながら、このようなフィードバック型調整法においても、これら調整則の設計は提案者の経験則やアイデアに基づくものであり、それらの改良指針や設計指針も明確でないものが多い。そこで本章では、本論文の核心部分として、メタヒューリスティクスのパラメータ調整問題を、最適化アルゴリズムの力学系の特性を調整するためにフィードバック系を設計してそのアルゴリズムの動特性を構造的に変更する問題、すなわち最適化アルゴリズムをメタレベルで最適化するためにフィードバック系を最適に設計する「メタ最適化問題」として捉え、この問題を定式化する。さらにこのフィードバック系の最適な調整則であるメタヒューリスティクスのパラメータ調整則を、解きたい最適化問題に適応させる形で経験則や試行錯誤によらずに行う方法論を提案する。

線形最適化問題に対する線形計画法や、凸型最適化問題に対する凸計画法は、解きたい問題の数理的構造が確定しており、その解に効率よく探索点を収束させる計算手法として理論的に開発され、その更新則に用いる設定しなければならないパラメータがほとんどないか、またはそれらの推奨値が解析的に求められていることが多い。これらの多くの解法がソルバーとして与えられ、ユーザーがそのアルゴリズムの中身を知らずとも解けるようにライブラリやパッケージとして実装されているのはこのためである。これが凸関数とは限らない目的関数を有する非線形最適化問題においては、とくにその非線形性の度合いや悪条件な構造を有する関数形の場合によっては、それに適したパラメータの設定を試行錯誤的に行う必要がある。メタヒューリスティクスによる多峰性関数の大域的最適解を求める場合がその極端な場合であり、ユーザーの試行錯誤による負担を軽減すべく、それらのパラメータ設定や調整を自動化することが本論文の目的である。

第2節において、アルゴリズムのパラメータの値をユーザーの使用環境に応じて最適に

設定する場合のメタ最適化の概念を解説したうえで、このメタ最適化問題を定式化する。そして、第3節において最適化アルゴリズムのイテレーション経過とともに、アルゴリズムの探索状態に応じてパラメータを動的に調整する調整則を獲得する問題にこのメタ最適化の概念を拡張し、その定式化を試みる。第4節ではメタヒューリスティクス設計思想について取り上げるとともに、新たにフラクタルを用いたベンチマーク問題を提案し、最適化問題を解くために必要なアルゴリズムの性質を考察する。第5節では調整則で用いることができるアルゴリズムの内部状態、ないしは調整則の引数を示したうえで、それらを最適化問題を解くための4節で紹介するアルゴリズムの性質と対比させる。第6節においては3節で定式化したパラメータ調整則設計のメタ最適化問題を多目的最適化問題として定式化し、最後に第7節において、この拡張されたメタ最適化問題の解法として遺伝的プログラミング (GP) を用いた解法を示す。

3.2 最適化アルゴリズムのパラメータ最適調整問題の定式化

3.2.1 最適化アルゴリズムの最適パラメータ決定問題とメタ最適化

最適化アルゴリズムを動作させるとき、様々なパラメータの設定が必要である。それらを分類すると、おもに

- (1) アルゴリズムの更新式中のパラメータ
- (2) イテレーション終了条件、探索点数等のパラメータ
- (3) 初期探索点の設定に関わるパラメータ

がある。(1)については、推奨値として一定の値に設定されるパラメータであり、メタヒューリスティクスはこのパラメータの設定に大きく依存することを前章においてシミュレーションによって示した。(2)については、アルゴリズムのユーザに許容されている計算時間に依存して決められるファンクションコール回数の上限值があり、また探索点の最適解への収束性や最適解近傍での停留が保証されているアルゴリズムであれば、その真の最適解と探索点との近さを表す精度を規定するパラメータがある。しかし、大域的最適解を求めることを目的としていながら、その解への理論的裏づけがないメタヒューリスティクスに関しては、イテレーション回数の上限值のみが主たる終了条件のパラメータとなり、ユーザが使用する計算機能力や最適化問題を解かねばならない期間など、ユーザの置かれている環境に依存して一意に決められ、探索点数等は経験則に基づく固定値を用いるのが一般的である。(3)については多くの最適化アルゴリズムが、ユーザによって探索点そのものの初期値が与えられるか、探索範囲を指定することでその範囲でランダムに自動的にとられることが多い。これらのことから、パラメータを操作することによるアルゴリズムの改良は、(1)を軸に行われる。

(1)の更新則中のパラメータをベクトル \mathbf{c} で表す。この次元は、パラメータの個数である。(2)のパラメータでイテレーション回数の上限值パラメータを K 、複数の探索点個数を P と表記し、いずれもスカラーとする。また、(3)のパラメータは探索点の初期座標や初期変位量などがあるが、ここでは便宜上探索点の初期座標のみとし、それを単点型アルゴリズム

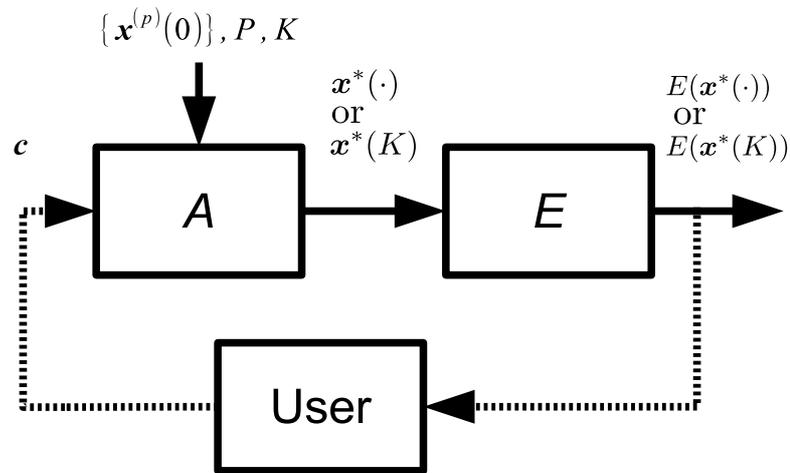


Fig. 3.1: Relations among algorithm, objective function and user.

ムでは単に $\mathbf{x}(0)$, 多点型アルゴリズムでは $\mathbf{x}^{(p)}(0)$, $p = 1, \dots, P$ と表すことにする. メタヒューリスティクスも含めて多くの最適化アルゴリズム A がソルバーとしてブラックボックス化されていることを考慮すると, Fig. 3.1 のような入出力関係の図で表すことができる. ここで, 各種のパラメータ \mathbf{c} が入力として与えられたもと, 最適化アルゴリズム A を動作させた結果得られる探索点の状態遷移を表す時系列 $\mathbf{x}^*(\cdot)$, ないしはその時系列の最終イテレーション K での解 $\mathbf{x}^*(K)$ が出力である. ユーザはこの時系列 $\mathbf{x}^*(\cdot)$ に対する目的関数 E の時系列 $E(\mathbf{x}^*(\cdot))$ ないしは, その最終イテレーション K での値 $E(\mathbf{x}^*(K))$ を評価し, 各種パラメータを設定しなおすことを行う. これはまさにユーザーを通したフィードバック系であり, Fig. 3.1 の破線の部分はこのことを表している.

ここで (2), (3) のパラメータは, ユーザによってあらかじめ与えられた所与のものとし (1) の更新則中のパラメータであるベクトル \mathbf{c} の最適化を考えるものとする. 最適化アルゴリズム A を動作させた結果得られる探索点の状態遷移を表す時系列を $\mathbf{x}^*(\cdot)$ とし, 以下では議論簡略化のため, Fig. 3.1 における出力は, この時系列のうちで探索点時系列の最終イテレーション $k = K$ での最適解あるいはその近似解 $\mathbf{x}^*(K)$ のみを考え, 時系列全体 $\mathbf{x}^*(\cdot)$ を目的関数で考慮する場合は考えないことにする. 具体的には, 更新式中にパラメータ \mathbf{c} を含むある最適化アルゴリズム A を実行したもとで得られる最適解ないしはその近似解 $\mathbf{x}^*(K)$ を, パラメータ \mathbf{c} に対するアルゴリズム A による写像とみなし,

$$\mathbf{x}^*(K) = A(\mathbf{c}) \tag{3.1}$$

とおく. ユーザが与える探索点の初期状態や計算の終了条件のもとで, パラメータ \mathbf{c} を含む最適化アルゴリズム A を用いて目的関数 $E(\mathbf{x})$ を最小化するとき, 得られる最適解あるいはその近似解に対する目的関数値 $E(\mathbf{x}^*(K))$ を最小にするようなアルゴリズムパラメータ \mathbf{c} を求める問題を, 解きたい最適化問題の目的関数 E とアルゴリズム A に対する「メ

「メタ最適化問題」と称し、

$$\begin{aligned} \min_{\mathbf{c}} E(\mathbf{x}^*(K)) \\ \text{where } \mathbf{x}^*(K) = A(\mathbf{c}) \end{aligned} \quad (3.2)$$

または、 $\mathbf{x}^*(K)$ を消去することにより、

$$\min_{\mathbf{c}} E(A(\mathbf{c})) \quad (3.3)$$

と定式化し、このように定式化される問題を意識して最適化アルゴリズムを設計することを単に「メタ最適化」と称することにする。なお、アルゴリズム写像 A で与えられる最適化アルゴリズムの力学系を改めて記号 A を用いて表し、さらに探索点の履歴

$$\chi(k) = \{\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(k)\} \quad (3.4)$$

にも依存させる形で具体的に、

$$\mathbf{x}(k+1) = A(\chi(k), \mathbf{c}), \quad k = 0, 1, \dots, K-1 \quad (3.5)$$

と表現すると、パラメータ \mathbf{c} に対する上記の差分方程式の解の時系列 $\mathbf{x}^*(\cdot)$ における、最終イテレーション $k = K$ での探索点 $\mathbf{x}^*(K)$ に対する目的関数値 $E(\mathbf{x}^*(K))$ を最小にするメタ最適化問題 (3.2) は、

$$\begin{aligned} \min_{\mathbf{c}} E(\mathbf{x}^*(K)) \\ \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ \mathbf{x}(k+1) = A(\chi(k), \mathbf{c}), \quad k = 0, 1, \dots, K-1 \\ \chi(0); \text{ given} \end{aligned} \quad (3.6)$$

と定式化しなおすことができる。

このような問題を意識したり定式化したりする意義は、古典的な局所最適化手法に関するかぎり必ずしも大きくはないが、パラメータ依存性が強いメタヒューリスティクスのアルゴリズムに対しては大きな意義をもつ。メタヒューリスティクスを適用する際、このようなアルゴリズムパラメータの決定問題が内在しているにも関わらず、推奨値とされるパラメータ値を用いるか、ユーザ自身の試行錯誤による決定に委ねられているのが現実である。しかし、パラメータの推奨値をアルゴリズムの開発者が決める作業や、解きたい問題に適したパラメータ値をユーザ自身が試行錯誤的に決める作業自体、工学的な設計問題の一つであり、上述のようなメタ最適化問題を暗に想定していることになる。本論文では、メタヒューリスティクスの性能を支配するアルゴリズムパラメータの値を、ユーザが解きたい問題に適応させ、かつ探索点の初期状態や計算終了条件に対応させて、最適に選択するアルゴリズム設計問題を工学的最適化問題として扱っている。とくにこのメタ最適化問題のパラメータに関する目的関数値を「解きたい最適化問題へのアルゴリズムパラメータの適合度」と称し、この適合度に関してパラメータ \mathbf{c} を最適調整することを、「アルゴリズム特性の進化」と称することにする。次節では、メタヒューリスティクスに対し、このメタ最適化によるアルゴリズム特性の進化の概念をより高機能化することを考える。

3.3 メタヒューリスティクスのパラメータ最適調整問題とメタ最適化

メタヒューリスティクスの代表的な最適化手法である PSO の改良手法である IWA-PSO や NDT-PSO は、慣性係数を調整する調整則を PSO 自体の更新式に連立させる形でアルゴリズムとして動作させる手法である。また、ES においては、目的関数の改善頻度に応じた更新式中のパラメータ調整が、その開発の初期段階から議論され、「1/5 ルール」がアルゴリズムの更新式に調整則として組み入れられてきた。さらに DE の改良手法についても、乱数を用いたかなり試行錯誤的な規則にせよ、目的関数を改善するか否かによるパラメータ調整則がアルゴリズムの更新則に加える形の改良手法が提案されている。FA においては、それ自体にパラメータ調整則がアルゴリズムに取り入れられており、乱数項に関しても時間と共に減少させていくことが提案されている。

PSO, ES, DE, FA のこうした改良手法の知見は、それぞれの手法ごとに関心をもつ別々の研究者による試行錯誤的な議論とそれらの計算機シミュレーションを通じた経験則によるものであり、これらのアルゴリズムパラメータの調整問題を工学的設計問題として横断的かつ統一的に捉えた議論がなされていないのが現状である。これらのパラメータ調整を伴う改良手法において共通する点は、まずアルゴリズムの更新式中のパラメータ \mathbf{c} がアルゴリズムのイテレーション経過とともに変動する時変パラメータであることである。そこで、このような場合、イテレーション回数 k を引数としてパラメータを $\mathbf{c}(k)$ と表し、イテレーション中のパラメータの時系列全体を表すときは、 $\mathbf{c}(\cdot)$ と表すことにする。ここで、最適化アルゴリズムの更新式中のパラメータ \mathbf{c} を時変パラメータの時系列 $\mathbf{c}(\cdot)$ として与えると、それを実行して得られる最終イテレーション $k = K$ での最適解ないしはその近似解 $\mathbf{x}^*(K)$ は、時変パラメータ $\mathbf{c}(\cdot)$ の写像として、

$$\mathbf{x}^*(K) = A(\mathbf{c}(\cdot)) \quad (3.7)$$

とおくことができる。したがって、この場合の問題 (3.2) に対する「メタ最適化問題」は、

$$\begin{aligned} \min_{\mathbf{c}(\cdot)} E(\mathbf{x}^*(K)) \\ \text{where } \mathbf{x}^*(K) = A(\mathbf{c}(\cdot)) \end{aligned} \quad (3.8)$$

または、

$$\min_{\mathbf{c}(\cdot)} E(A(\mathbf{c}(\cdot))) \quad (3.9)$$

となり、問題 (3.6) に対しては

$$\begin{aligned} \min_{\mathbf{c}(\cdot)} E(\mathbf{x}^*(K)) \\ \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k)), k = 0, 1, \dots, K-1 \\ \boldsymbol{\chi}(0); \text{ given} \end{aligned} \quad (3.10)$$

と定式化されることになる。

PSOの改良手法であるIWA-PSOは、調整パラメータ \mathbf{c} として慣性係数 c_0 のみを選び、その時系列

$$c_0(k) = c_{\text{start}} - (c_{\text{start}} - c_{\text{end}}) \frac{k}{K} \quad (3.11)$$

を、最適か否かは別としてある種のメタ最適化問題の解とみなしているものといえる。3.2節で定式化したメタ最適化問題においては、パラメータ \mathbf{c} を一定としているが、時系列 $\mathbf{c}(\cdot)$ が一定の値を取り続けるものとするれば、一定のパラメータの推奨値を決める問題(3.6)もパラメータ時系列を求めるメタ最適化問題(3.10)の特殊な場合とみなすことができる。こうして、パラメータを時系列化することで、たとえば、IWA-PSOのように大域的な探索から局所的探索へ移行させるなど、探索過程に探索点の挙動に変化を与えることで、メタ最適化の機能を高めることが可能である。

また、探索性能を評価する目的関数が、計算終了時のイテレーションの上限での探索点座標の目的関数値 $E(\mathbf{x}^*(K))$ ではなく、探索点群の群れとしての挙動特性を示す指標、たとえば探索点群の活性度を指標とし、その変動の目標時系列に追従するようなパラメータ調整を行う活性度調整型PSOに対して、その最適調整則を与えるメタ最適化問題を定式化すると、

$$\min_{\mathbf{c}(\cdot)} \sum_{k=1}^K |I(\mathbf{x}^*(k)) - I_R(k)| = J(\mathbf{x}^*(\cdot))$$

where $\mathbf{x}^*(\cdot)$ solution to

$$\begin{aligned} \mathbf{x}(k+1) &= A(\chi(k), \mathbf{c}(k)), \quad k = 0, 1, \dots, K-1 \\ \chi(0) &; \text{ given, } \mathbf{c}(0) ; \text{ given} \end{aligned}$$

となる。ここで、 $J(\mathbf{x}^*(\cdot))$ は探索点時系列 $\mathbf{x}^*(\cdot)$ の挙動を評価する指標であり、パラメータ時系列 $\mathbf{c}(\cdot)$ に対する最適化アルゴリズムによる探索点時系列 $\mathbf{x}^*(\cdot)$ に関する評価指標の時系列 $I(\mathbf{x}^*(\cdot))$ とその目標時系列 $I_R(\cdot)$ との偏差を最小にするように、パラメータ時系列 $\mathbf{c}(\cdot)$ を求める問題である。

一方、PSOのNDT-PSOやESおよびDEにおけるパラメータ調整則は、探索点の挙動状態に応じながらパラメータの値を変動させるものである。この場合もパラメータは時系列となるが、この場合のパラメータ \mathbf{c} の引数は、イテレーション回数 k だけでなく、探索点の挙動状態を表す探索点座標 \mathbf{x} も引数とする関数 \mathbf{F} を用いて、

$$\mathbf{c} = \mathbf{F}(k, \mathbf{x}) \quad (3.13)$$

とみなすことができる。イテレーション回数 k を引数にしない場合でも、探索点座標 \mathbf{x} が時系列であるため、パラメータ \mathbf{c} は結果的に

$$\mathbf{c}(k) = \mathbf{F}(\mathbf{x}(k)), \quad k = 0, 1, \dots, K \quad (3.14)$$

と時系列になる。この場合、最適化アルゴリズム A を実行して得られる最終イテレーション $k = K$ での最適解ないしはその近似解 $\mathbf{x}^*(K)$ は、直接パラメータ時系列に依存するのではなく、関数 $\mathbf{F}(\cdot)$ に依存するため、これを引数として、

$$\mathbf{x}^*(K) = A(\mathbf{F}(\cdot)) \quad (3.15)$$

と置くことができる。したがって、メタ最適化問題は目的関数値 $E(\mathbf{x}^*(K))$ を最小にするような関数 $\mathbf{F}(\cdot)$ を決定する問題となり、

$$\begin{aligned} \min_{\mathbf{F}(\cdot)} E(\mathbf{x}^*(K)) \\ \text{where } \mathbf{x}^*(K) = A(\mathbf{F}(\cdot)) \end{aligned} \quad (3.16)$$

または、

$$\min_{\mathbf{F}(\cdot)} E(A(\mathbf{F}(\cdot))) \quad (3.17)$$

と定式化される。これらの定式化において、 $\mathbf{F}(\cdot)$ の表記を $\mathbf{F}(\mathbf{x})$ と記述しないのは、前者は変数 \mathbf{x} の空間上で定義された関数自体を変数とする関数空間での最適化問題であるのに対して、後者では変数 \mathbf{x} のある値に対する関数 $\mathbf{F}(\mathbf{x})$ の値で最小化するという意味と解釈されるためである。この問題をメタ最適化問題 (3.10) に対応づけて定式化すると、

$$\begin{aligned} \min_{\mathbf{F}(\cdot)} E(\mathbf{x}^*(K)) \\ \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ \begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k)) \\ \mathbf{c}(k) = \mathbf{F}(\mathbf{x}(k)) \end{cases} \\ k = 0, 1, \dots, K-1 \\ \boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given} \end{aligned} \quad (3.18)$$

となる。

ところで、関数 \mathbf{F} はすべての探索点座標 \mathbf{x} をパラメータの値 \mathbf{c} に対応づけるベクトル関数で、これを改めて $\mathbf{c} = \mathbf{F}(\mathbf{x})$ と表現すると、探索点座標 \mathbf{x} を入力とし、それに応じてパラメータ \mathbf{c} の値が出力として決まる入出力関係である。したがって、パラメータ \mathbf{c} の空間の値を入力とし、 \mathbf{x} の空間の値を出力するような入出力関係 $\mathbf{x} = A(\mathbf{c})$ とは、その入出力関係が逆の関係であることがわかる。そこで、このようにパラメータ \mathbf{c} を \mathbf{x} の関数として与える問題を「フィードバック型メタ最適化問題」と称することにする。このようなフィードバックの関係を図示すると、Fig. 3.2 のようになる。最適化アルゴリズムの写像 A から生成される変数 \mathbf{x} の力学系に対して、関数 \mathbf{F} により \mathbf{x} から \mathbf{c} を生成する逆関数によってフィードバック系を構成することは、最適化アルゴリズムの写像 $A(\cdot)$ の力学系が有する動的特性に対して、写像 $A(\cdot)$ と関数 $\mathbf{F}(\cdot)$ の合成写像 $A(\mathbf{F}(\cdot))$ による新たな力学系を構成していることになり、本来のアルゴリズムの力学系の特性に対して本質的な変更を与えることに相当し、より高機能なアルゴリズムパラメータの調整を行っていることになる。このような例としては、NDT-PSO における調整則があげられ、たとえば (2.18c) 式において、 g -best 座標 $\mathbf{x}^{(g\text{-best})}$ を定数とみなし、かつ探索点番号の上付き添字 (p) を省略すると

$$F(\mathbf{x}) = 1 - d_1 + d_1 d_0 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}^{(g\text{-best})}\|}{d_2 d_0}\right) \quad (3.19)$$

とみなすことができる。

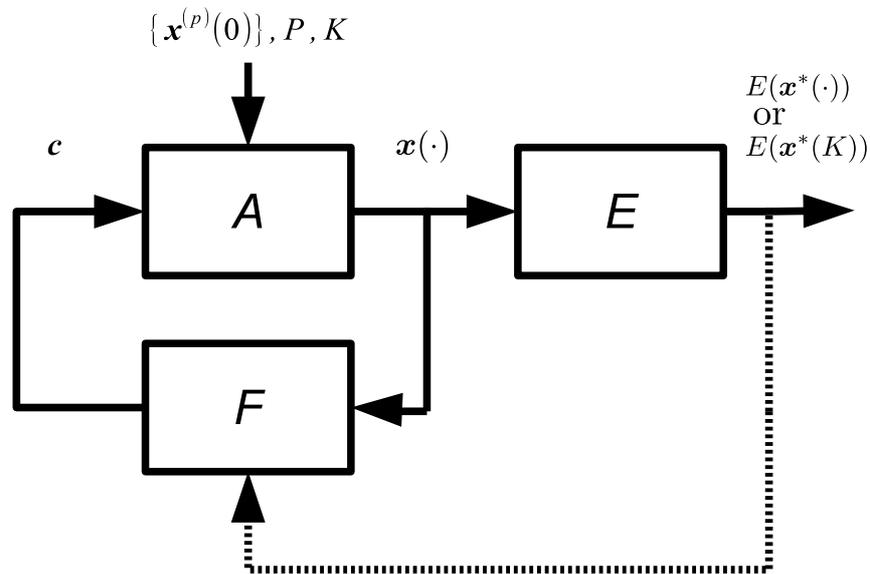


Fig. 3.2: Relations among algorithm, objective function and parameter tuning rule.

一方, ESにおける正規乱数の標準偏差に相当するパラメータ $\sigma(k)$ の調整則である (2.24), (2.25) 式は,

$$\sigma(k+1) = F(\sigma(k), p_s(k)) \quad (3.20)$$

と一般化することができる. ここで $p_s(k)$ は, 現在のイテレーション k から遡った一定のイテレーション期間で突然変異が成功した頻度であり, この遡る一定のイテレーション期間を l とすると, $p_s(k)$ は $\mathbf{x}(k)$ だけでなく, 過去に l 期遡った時系列 $\{\mathbf{x}(k-l), \mathbf{x}(k-l+1), \dots, \mathbf{x}(k)\}$ やその目的関数値を関数 F の引数にしていることになり, それを示しているのが破線の部分である. (試験点との目的関数値の比較にもよるので試験点の時系列を引数にする関数ともみなせるが, これについては議論簡略化のため省略する). また, 関数 F の引数に現イテレーションのパラメータ $\sigma(k)$ も含まれていることから, 関数 F による関数にも離散系のダイナミクスをもたせていることがわかる. これらのことを一般化すると, パラメータの更新則は離散力学系として

$$\mathbf{c}(k+1) = F(\mathbf{c}(k), \chi(k)), \quad k = 0, 1, \dots, K-1 \quad (3.21)$$

と一般化することができる. このような一般化された調整則の離散力学系を与える関数 F を用いて, 計算終了時のイテレーションの上限 $k = K$ での探索点 $\mathbf{x}^*(K)$ に対する目的関

数値 $E(\mathbf{x}^*(K))$ を最小にするようパラメータ調整則を求めるメタ最適化問題は、

$$\begin{aligned} & \min_{\mathbf{F}(\cdot, \cdot)} E(\mathbf{x}^*(K)) \\ & \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ & \begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k)) \\ \mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \boldsymbol{\chi}(k)) \end{cases} \quad (3.22) \\ & k = 0, 1, \dots, K-1 \\ & \boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given} \end{aligned}$$

とさらに一般化して定式化することができる。この問題を「最適化アルゴリズム A に対するパラメータの最適調整則 \mathbf{F} を与えるメタ最適化問題」と称することにする。なお、問題 (3.12) に対応する形で、パラメータ時系列 $\mathbf{c}(\cdot)$ に対する最適化アルゴリズムによる探索点の時系列 $\mathbf{x}^*(\cdot)$ に関する評価指標の時系列 $I(\mathbf{x}^*(\cdot))$ とその目標時系列 $I_R(\cdot)$ との偏差を目的関数として導入すると、これを最小にするようなパラメータ時系列 $\mathbf{c}(\cdot)$ を与えるフィードバック調整則 $\mathbf{F}(\cdot, \cdot)$ を求める、いわば問題 (3.22) に対応するメタ最適化問題は、

$$\begin{aligned} & \min_{\mathbf{F}(\cdot, \cdot)} \sum_{k=1}^K |I(\mathbf{x}^*(k)) - I_R(k)| \\ & \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ & \begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k)) \\ \mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \boldsymbol{\chi}(k)) \end{cases} \quad (3.23) \\ & k = 0, 1, \dots, K-1 \\ & \boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given} \end{aligned}$$

となる。

問題 (3.18) や問題 (3.22) のように、最適パラメータ時系列を与える調節則を与える関数 \mathbf{F} を求めるメタ最適化問題を「フィードバック型メタ最適化問題」ということにし、これに対して問題 (3.10) のような最適なパラメータ時系列を直接求める「フィードフォワード型メタ最適化問題」からは区別する。とくにフィードバック型問題は、関数 \mathbf{F} によるフィードバック関数が静的な問題 (3.18) のタイプと、パラメータ調整にダイナミクスをもたせて動的にフィードバックを行う問題 (3.22) のタイプを考えることができる。問題 (3.18) を「フィードバック型静的調整則決定メタ最適化問題」、問題 (3.22) を「フィードバック型動的調整則メタ最適化問題」と称することにする。

なお、目的関数 $E(\mathbf{x}^*(K))$ や $\sum_{k=1}^K |I(\mathbf{x}^*(k)) - I_R(k)|$ は、最適化アルゴリズムの時系列 $\mathbf{x}^*(\cdot)$ が決まるとそれに対応して実数スカラー値が決まる汎関数であるため、以降では便宜上のこの汎関数を $J(\mathbf{x}^*(\cdot))$ と統一的に表すことが可能である。この汎関数の表記を用い、以上において定式化したメタ最適化問題をまとめると以下のようになる。

パラメータ値メタ最適化問題:

$$\begin{aligned} & \min_{\mathbf{c}} J(\mathbf{x}^*(\cdot)) \\ & \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ & \quad \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}), \quad k = 0, 1, \dots, K-1 \\ & \quad \boldsymbol{\chi}(0); \text{ given} \end{aligned} \tag{3.24}$$

パラメータ時系列メタ最適化問題

$$\begin{aligned} & \min_{\mathbf{c}(\cdot)} J(\mathbf{x}^*(\cdot)) \\ & \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ & \quad \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k)), \quad k = 0, 1, \dots, K-1 \\ & \quad \boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given} \end{aligned} \tag{3.25}$$

フィードバック型静的調整則メタ最適化問題:

$$\begin{aligned} & \min_{\mathbf{F}(\cdot)} J(\mathbf{x}^*(\cdot)) \\ & \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ & \quad \begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k)) \\ \mathbf{c}(k) = \mathbf{F}(\mathbf{x}(k)) \end{cases} \\ & \quad k = 0, 1, \dots, K-1 \\ & \quad \boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given} \end{aligned} \tag{3.26}$$

フィードバック型動的調整則メタ最適化問題:

$$\begin{aligned} & \min_{\mathbf{F}(\cdot, \cdot)} J(\mathbf{x}^*(\cdot)) \\ & \text{where } \mathbf{x}^*(\cdot) \text{ solution to} \\ & \quad \begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k)) \\ \mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \boldsymbol{\chi}(k)) \end{cases} \\ & \quad k = 0, 1, \dots, K-1 \\ & \quad \boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given} \end{aligned} \tag{3.27}$$

3.4 メタ最適化問題の目的関数依存性と適合目的関数

更新式中にパラメータ \mathbf{c} を含むある最適化アルゴリズム A を実行したもとで得られる探索点の時系列 $\mathbf{x}^*(\cdot)$ は、パラメータ \mathbf{c} だけでなくメタヒューリスティクスを用いてユーザーが解きたい最適化問題の目的関数 E にも依存する。したがって、たとえば (3.7) 式は正確には、

$$\mathbf{x}^*(K) = A(\mathbf{c}(\cdot); E) \tag{3.28}$$

と表記しなければならない。3.3節で定式化したメタ最適化問題の解である最適パラメータの値、ないしはその最適時系列、あるいはそれを与える最適調整則は、いずれも解こうとしている最適化問題の目的関数 E に依存する。したがって、これらは最適化アルゴリズムのあるユーザが解こうとする最適化問題に適応した最適なパラメータの値、ないしはその最適時系列、さらにはそれを与える最適調整則であり、解こうとしている最適化問題を環境とみなし、それにあるアルゴリズムを適応させた結果がメタ最適化問題の解である最適パラメータの値、ないしはその最適時系列、あるいはそれを与える最適調整則といえる。したがって、メタ最適化問題の考え方は、本質的にユーザが解こうとしている問題に依存するものである。

以上のような考え方を基盤にすると、たとえば、文献 [16] では、

- (1) 「良い解の近傍にはさらにより良い解がある」といういわゆる近接最適性原理 (POP) が成り立つ目的関数に対して良好な計算性能を得ること

がアルゴリズムパラメータの調整指針の一つとして提唱されている。これに根拠をおくのが、探索点の挙動が発散傾向にある大域的探索と収束傾向にある局所的探索との異なる特性、あるいは複数の探索点の分散化 (Diversification) と集中化 (Intensification) の特性をアルゴリズムにもたせようとするパラメータ調整の設計指針である。PSOのパラメータ調整法において大域的探索から局所的探索に一方向的に移行させるパラメータ調整や、それらの探索性能を双方方向的に移行したりするようなパラメータ調整、さらには探索点の集団としての活性度を制御するようなパラメータ調整法がこれに相当する。POPが成り立つ目的関数として、大域的な谷の構造に細かな周期的波形が重なった **Rastrigin** 関数がある。しかし、POP自体が定性的な性質であり、PSOの各種のパラメータ調整法によって、POPが成り立つ目的関数に対して良好な探索性能がそれらのパラメータ調整によって獲得されている理論的保証はなく、ましてPOPの成立が保証されていない関数に対しては、発散と収束、ないしは分散化と集中化という設計指針は有効とは限らない。

これに対して近年では、たとえば文献 [52][63] のように、

- (2) 「変数空間に変換を施しても探索初期点に対して同じ変換を施せば、あるいは目的関数に変換を施せば、それに対する探索性能は不変である」という計算性能の「変換不変性」があること

が、アルゴリズムないしはそのパラメータ調整の設計指針として提唱されている。この不変性には、具体的に

- (a) 変数空間の平行移動に対し、それと同等の探索初期点の平行移動によりアルゴリズムの探索性能が不変 (平行移動不変性)
- (b) 変数空間の座標回転に対し、それと同等の探索初期点の回転によりアルゴリズムの探索性能が不変 (回転不変性)
- (c) 座標軸のスケール変換に対し、それと同等の探索初期点座標のスケール変換によりアルゴリズムの探索性能が不変 (座標スケール不変性)
- (d) 目的関数値のスケール変換に対し、アルゴリズムの探索性能が不変 (目的関数値スケール不変性)

をあげることができる。PSOやDEの更新式は、本質的には線形演算であるため、線形演算である平行移動や変数のスケール変換に対しては、その不変性を有するものといえる。しかし、回転不変性については、回転演算が線形であっても、PSOやDEの更新式が変数の成分ごとに記述されるため、変数成分間に相互干渉がない目的関数に対して良好な探索性能を有するようパラメータ調整を行った更新式を、回転操作を施した変数座標で同じ目的関数に対して適用してみると、回転不変性の欠如から探索性能が低下することが確認される。そこで、文献 [52] の CMA-ES のように、回転不変性をもたせた更新式とパラメータ調整法を有する ES の開発が行われている。ただし、変数座標に対して

$$\mathbf{R}(\alpha) = \prod_{i=1}^N \prod_{j=0}^N \mathbf{R}^{i,j}(\alpha_{ij})$$

$$r_{k,l}^{i,j}(\alpha_{ij}^q) = \begin{cases} \cos(\alpha_{ij}) & k = i, l = i \\ -\sin(\alpha_{ij}) & k = i, l = j \\ \sin(\alpha_{ij}) & k = j, l = i \\ \cos(\alpha_{ij}) & k = j, l = j \\ 1 & k = l \neq i, k = l \neq j \\ 0 & \text{otherwise} \end{cases} \quad (3.29)$$

によって回転操作を変数に施した目的関数に対して良好な探索性能を有するようにアルゴリズムを設計したとしても、それがユーザーが解きたい問題の目的関数に適合するアルゴリズムである保証はない。そのため、解きたい最適化問題の目的関数に適合したパラメータ調整法をメタ最適化の考え方によって発見することは、課題を克服するアプローチの一つであるといえる。

以上のような、POPに基づく設計や変換不変性の付与による設計のほかに、

(3) 「悪条件な目的関数形状」へ適合する計算性能があること

も設計指針としてあげることができる。この悪条件の例としては、たとえばある方向には急峻で別の方向には緩慢な細長い湾曲した形状をした谷を有する関数で、Rosenbrock 関数が有名である。

本論文では、メタヒューリスティクスのアルゴリズムが適合しなければならない新たな特徴をもつ目的関数として、またそのための設計指針として、「フラクタル的探索持続性」を提案する。具体的には、再帰的に定義される Fig. 3.3 のような形状のフラクタル関数

$$f(x) = \begin{cases} \frac{1}{3}f(3x), & \text{if } 0 \leq x < \frac{1}{3} \\ -\frac{1}{3} + \frac{1}{3}f(3x - 1), & \text{else if } \frac{1}{3} \leq x \leq \frac{2}{3} \\ \frac{1}{3}f(3x - 2), & \text{else if } \frac{2}{3} < x \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.30)$$

を用いて、 N 次元実空間で定義された目的関数

$$E(\mathbf{x}) = \sum_{n=1}^N f(x_n) \quad (3.31)$$

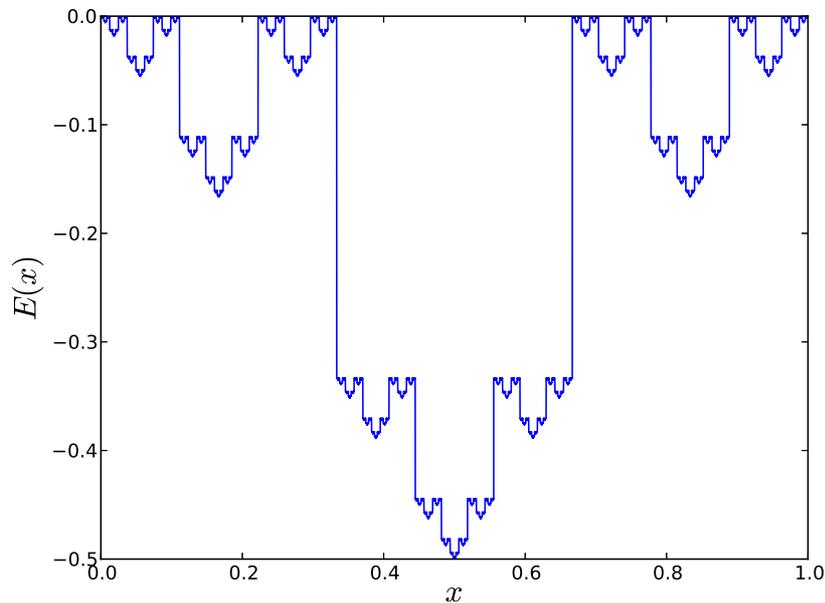


Fig. 3.3: Shape of Fractal function.

を考える. ここで, あるアルゴリズムで得られる解 \mathbf{x}^* に対応して

$$\left(\max_n \left| x_n^* - \frac{1}{2} \right| \right) \leq \frac{1}{2} \left(\frac{1}{3} \right)^{-m}, \quad n = 1, \dots, N \quad (3.32)$$

を満たす最小の整数 m を考えると, \mathbf{x}^* が第 $m - 1$ 回の再帰部分の領域 (第 $m - 1$ 階層の領域という) にあることを示す. したがって, この Fig. 3.3 で喩えれば, (3.31) 式の N 次元のフラクタル関数に適合するメタヒューリスティクスのアルゴリズムの性能として, イテレーションが進むに従って, その探索範囲の階層数が大きくなる領域を集中して探索していく性能が望まれる. すなわち, 最大イテレーション K の範囲で持続探索し続けて, 第 K イテレーション時に求まるアルゴリズムの解 $\mathbf{x}^*(K)$ に対応する整数の値 $-m$ が, 最大イテレーション回数の増加とともに大きくなるのが, メタヒューリスティクスの性能として望まれる. このような探索性能をメタヒューリスティクスの「フラクタル性」と称することにする.

本論文で提唱する「メタ最適化」の概念は, こうした様々な設計指針を前提としてメタヒューリスティクスのアルゴリズムの更新式や調整則を設計するのではなく, 「ユーザが解きたい個々の最適化問題の目的関数に適合したより良好な探索性能を獲得するよう, メタヒューリスティクスの更新式に対して最適な調整則を発見すること」ということができる. そこで本論文では, 本論文におけるメタ最適化問題において, アルゴリズムを進化的に適合させる対象の目的関数として,

- (a) 変数間依存を持たせるため回転変換した多峰性の Rastrigin 関数
- (b) 変数間依存を持たせるため回転変換した多峰性で局所的最適値に差が少ない Alpine 関数

(c) 変数間依存をさらに強くさせるため回転変換した多峰性の Levy 関数

(d) 変数間依存を持つ悪条件の Rosenbrock 関数

を取り上げることとする。メタヒューリスティクスの適合対象とするこのような目的関数をとくに「適合目的関数」と称することとする。

3.5 メタヒューリスティクスのパラメータ調整変数の種類

メタヒューリスティクスの更新式中のパラメータの調整則について統一的に議論した研究は、わずかに進化計算に対する検討が文献 [64] においてされているのみである。本節では、第 3.3 節で離散力学系の形で導入したアルゴリズムのパラメータ調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \boldsymbol{\chi}(k)), \quad k = 0, 1, \dots, K-1 \quad (3.33)$$

において、この離散力学系によるパラメータ調整の駆動力となる変数 $\boldsymbol{\chi}$ として用いる情報について考察する。まず、第 2 章で考察した各種の調整則や文献 [64] を参考にして、変数 $\boldsymbol{\chi}$ として用いる情報を次の 4 種類に大別する。

(1) 複数の探索点からなる群の状態情報

たとえば、探索点群の群れとしての特性を表す変数として探索点間の相対的な位置関係を変数とし、この値によって探索点群の集中度を調整するものである。一例として NDT-PSO や AF-PSO の調整則がこれに相当し、探索点間の距離が近い場合は多様化させて大域的探索へ移行し、探索点間の距離が離れている場合は集中化させて局所的探索へ移行するようにパラメータが調整される。

(2) イテレーション経過情報

局所的最適解ですらその停留条件を与えることができないメタヒューリスティクスでは、イテレーションの上限回数に達するまで更新し続けるようにするが、その設定したイテレーションの上限回数までのイテレーション経過割合 (以降では「経過時間」と称する) を調整則の変数とするものである。多くは経過時間の増加とともに探索点群を大域的な探索から局所的な探索へと移行するようにパラメータが調整される。

(3) 個々の探索点の目的関数値の改善情報

各探索点がイテレーションの更新時に目的関数を改善したかどうかの情報を、パラメータの調整のために利用するもので主として ES に用いられる。一例として 1/5 ルールでは目的関数値の改善が見られない、ないしは悪化をする場合には探索点の変位を抑え、目的関数値の改善される場合は、探索点の変位を大きくする。このように個々の探索点の挙動を目的関数値の更新情報によって調整する点で、探索点の群れとしての挙動を調整する (1) とは、その調整則の考え方が異なる。

(4) 複数探索点の目的関数値の情報

複数の探索点の目的関数値による優劣をパラメータ調整に用いるもので、Adapt-DE におけるパラメータ更新条件や、CMA-ES の重心を求めるときの重みパラメータの調整に用いられるもので、遺伝的アルゴリズムのルーレット選択等にもみられる。

以上のパラメータ調整に使われる情報の分類において、使われる情報によってはアルゴリズムの不変性に影響を与える。例えば(1)の複数探索点群の状態として探索点の位置情報を用いてパラメータ調整を行うことは、使い方によってはアルゴリズムの回転不変性は担保することができるが、スケール不変性は損なわれてしまう。例として変位ベクトルの大きさ $\|\mathbf{v}(k)\|$ に2乗ノルムを用いた調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \|\mathbf{v}(k)\|), k = 0, 1, \dots, K-1 \quad (3.34)$$

を考える。座標のアフィン変換を考え、変換行列を \mathbf{M} とすると、上式は

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \|\mathbf{M}\mathbf{v}(k)\|), k = 0, 1, \dots, K-1 \quad (3.35)$$

と書き換えられる。 \mathbf{M} が回転行列であれば、 $\|\mathbf{M}\mathbf{v}(k)\| = \|\mathbf{v}(k)\|$ となり、影響は与えないが、 \mathbf{M} が回転行列でなければ、 $\|\mathbf{M}\mathbf{v}(k)\| \neq \|\mathbf{v}(k)\|$ となり、パラメータ調整則 \mathbf{F} が座標系に依存することとなる。このため、パラメータ調整則 \mathbf{F} は座標の変換に対してロバストでなくなり、与えられた問題の座標系に応じて、アルゴリズムの性能が変化する。また、同様の議論で変位ベクトルの大きさに1乗ノルムを使用した場合、回転に対する不変性も満たさず、フラクタルな関数に対しても、大域的な探索と局所的な探索においてノルムが変化するため、フラクタル性も満たさない。

さらに、(4)において目的関数値そのものを用いる調整則も目的関数値に対する不変性を損なう。メタヒューリスティクスは一般的に、目的関数値の大小関係をもとに駆動し、大小関係が同じであれば同じ解を得る。単調増加の変換 h を考えた際に、目的関数値に依存するパラメータ調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), E(\mathbf{x}(k))), k = 0, 1, \dots, K-1 \quad (3.36)$$

を、変換関数 h を用いて目的関数のスケールを変換した場合

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), h(E(\mathbf{x}(k))))), k = 0, 1, \dots, K-1 \quad (3.37)$$

となるが、一般的に $E(\mathbf{x}(k)) \neq h(E(\mathbf{x}(k)))$ であるため、目的関数値のスケールに対する不変性が損なわれる。そのため、順位等の序数情報に変換する必要がある。

このように、(1)と(4)は解きたい最適化問題の変換に対してロバストでない場合があり、アルゴリズムの性能が問題に依存する。一方で、(2)と(3)による調整ルールは、アルゴリズムの内部状態(イテレーションの経過 $t(k)$ や、探索点の更新確率 $p_s(k)$) に依存し、解きたい最適化問題の変換によって陰に影響は受けるが(例えば問題の変換により、 $p_s(k)$ に影響が及び、結果として問題に対する性能が変化する)、(1)と(4)の様に施す変換から陽に影響は受けない。言い換えれば(2)と(3)の調整ルールによる性能変化はアルゴリズム内の問題であり、アルゴリズム内の調整則によって補償されるため、解きたい問題に依存せず、より問題に対してロバストであると考えられる。

以上により、本論文では、(2)と(3)で用いられる変数や情報、そして少なくとも回転に対する不変性が保存される(1)の2乗ノルムを用いた場合をパラメータ調整の基本と考える。まず、(1)に相当する探索点の群れとしての情報として、それらの集中・分散度の指標になるものとして、探索点の位置情報ではなく変位情報を探索点群で集約した

(a) 第 k イテレーションでの探索点の変位の大きさ $\|\mathbf{v}^{(p)}(k)\|$ の探索点群全体での平均

$$\|\mathbf{v}(k)\| = \frac{1}{P\sqrt{N}} \sum_{p=1}^P \|\mathbf{v}^{(p)}(k)\| \quad (3.38)$$

と、探索点と g -best 座標の距離の探索点群全体での平均

$$\Delta_g(k) = \frac{1}{P\sqrt{N}} \sum_{p=1}^P \|\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k)\| \quad (3.39)$$

を変数とすることを提案する．これらは文献 [49][50] などにおいて、PSO における探索点群の集中・分散度を示す指標として、また探索点の大域的探索と局所的探索性能を調整するための有力な指標として提案されたもので、手法間での公平な比較のために他のメタヒューリスティクスにも導入する．なお、探索点位置情報を調整則に用いることは、アルゴリズムのスケール不変性を損なう恐れがあるが、探索点群での最良の探索点との位置関係は、パラメータ調整にとって重要な情報であると考え、探索点群からの各探索点の群れとしての平均距離も併せて変数に採用する．

また、(2) の情報を用いた調整則として、

(b) あらかじめ設定された最大イテレーション回数 K に対する第 k イテレーション時の経過割合

$$t(k) = k/K \quad (3.40)$$

を、調整則のための重要な変数の一つとして設定する．文献 [44] において PSO にも導入されている調整則であるが、アルゴリズムの更新経過とともに大域的探索から局所的探索へとその探索の様相を変化させるのに重要な変数と捉え、他のメタヒューリスティクスにも導入する．

パラメータ調整のための第三の変数として、(3) の情報を用いた調整則変数として、個々の探索点が目的関数を改善した情報を群れとして集約するために、

(c) 第 k イテレーションから遡った一定イテレーション間で、目的関数値を改善させることに成功したイテレーション数の割合 $p_s^{(p)}(k)$, $p = 1, \dots, P$ の探索点群全体での平均

$$p_s(k) = \frac{1}{P} \sum_{p=1}^P p_s^{(p)}(k) \quad (3.41)$$

を採用する．探索点を多点化した多点型 ES アルゴリズムにおいて、この変数を用いた調整則を本論文の2章で用いたが、公平な比較のためにも PSO, DE, FA 等の異種のアルゴリズムに対し、これを変数に用いた調整則の設計を試みる．

以上により、本論文では、メタヒューリスティクスのアルゴリズムの共通のパラメータ調整則として、これらを変数とした3種類の調整則を取り上げ、その有効性を以下の章において確認する．具体的には、パラメータ調整を行いたいメタヒューリスティクスアルゴリズム A のパラメータを \mathbf{c} 、このパラメータをイテレーションの経過と伴に再帰的に調整する関数を \mathbf{F} とすると、

(a) 探索点変位および最良点との距離による調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \|\mathbf{v}(k)\|, \Delta_g(k)) \quad (3.42)$$

(b) イテレーション経過割合による調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), t(k)) \quad (3.43)$$

(c) 目的関数改善割合による調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), p_s(k)) \quad (3.44)$$

の3種類のフィードバック型動的調整則である。本論文では、与えられた解くべき最適化問題と、それに適用したいアルゴリズムにとって、そのアルゴリズム中のパラメータをイテレーション経過とともに調整するための調整則をGPにより進化的に獲得した結果を次章で示す。

3.6 複数の適合関数を考慮した多目的メタ最適化問題

ユーザが複数の最適化問題にアルゴリズムを適合させる場合や、複数のベンチマーク問題に対して最適化アルゴリズムのパラメータ設計を行う場合などは、これらのメタ最適化問題は多目的問題となる。具体的に、パラメータの設計や調整によって最適化アルゴリズムを適合させたい対象の目的関数が複数の場合、これら複数の目的関数を $E_1(\mathbf{x}), E_2(\mathbf{x}), \dots, E_R(\mathbf{x})$ とすると、メタ最適化問題はこれら複数の目的関数を対象した問題であること明記する形で

$$\min_{\mathbf{F}(\cdot, \cdot)} \begin{pmatrix} J(\mathbf{x}^*(K); E_1) \\ \vdots \\ J(\mathbf{x}^*(K); E_R) \end{pmatrix}$$

where $\mathbf{x}^*(\cdot)$ solution to

$$\begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k); E_r) \\ \mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \boldsymbol{\chi}(k)) \end{cases} \quad (3.45)$$

$$k = 0, 1, \dots, K-1$$

$$\boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given}$$

$$r = 1, \dots, R$$

と定式化される。 $r = 1$ のときが問題 (3.27) に対応する。関数 $J(\mathbf{x}^*(K); E_r)$ は、調整則 \mathbf{F} を有するメタアルゴリズム A を目的関数 E_r ごとに動作させて得られる最終イテレーション K での解 $\mathbf{x}^*(K)$ を、同一の関数 J で評価することを意味している。関数 J そのものを「評価関数」、関数 $J(\mathbf{x}^*(K); E_r)$ の値を調整則 \mathbf{F} の関数、つまり調整則を規定する関数 \mathbf{F} の汎関数とみなした場合を「目的関数」と称することにする。したがって、単に互いに異

なる目的関数が存在する多目的最適化問題ではなく、異なる複数種の評価対象を同一の評価関数で評価する多目的最適化問題であることに注意されたい。

一般に多目的最適化問題の合理的解は、より優れた許容解が他に存在しない解である Pareto 解(非劣解)とされるが、すべての目的関数に共通した最適解が存在しない場合、この Pareto 解は一般に無限個の要素をもつ集合となる。目的関数の個数が 2, 3 個の場合は、数多くの探索点からなる多点型探索法を用い、その Pareto 解集合全体を多くの探索点で被覆する手法が登場している。しかし、目的関数の個数が比較的多い場合には、解集合全体を多くの探索点で被覆しても、その解集合の広さのために特定の解を選好することが困難となり実用性に乏しい。このため一般的には、Pareto 解の集合内の限られた解を最適解として与えるという意味で等価な単一目的関数最適化問題を、多目的最適化問題に代替させて定式化して解くことが行われる。その代表例として、重み付け和最適化問題と重み付け最大成分最小化問題をあげておく。前者は、

$$\min_{\mathbf{F}(\cdot, \cdot)} \sum_{r=1}^R w_r J(\mathbf{x}^*(K); E_r)$$

where $\mathbf{x}^*(\cdot)$ solution to

$$\begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k); E_r) \\ \mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \boldsymbol{\chi}(k)) \end{cases} \quad (3.46)$$

$$k = 0, 1, \dots, K-1$$

$$\boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given, } w_r; \text{ given}$$

$$r = 1, \dots, R$$

と定式化されるのに対して、後者は

$$\min_{\mathbf{F}(\cdot, \cdot)} \max_r w_r J(\mathbf{x}^*(K); E_r)$$

where $\mathbf{x}^*(\cdot)$ solution to

$$\begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k); E_r) \\ \mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \boldsymbol{\chi}(k)) \end{cases} \quad (3.47)$$

$$k = 0, 1, \dots, K-1$$

$$\boldsymbol{\chi}(0); \text{ given, } \mathbf{c}(0); \text{ given, } w_r; \text{ given}$$

$$r = 1, \dots, R$$

と定式化される。ここで、重み係数 w_r , $r = 1, \dots, R$ は

$$(w_1, \dots, w_R)^T \in \left\{ \mathbf{w} \mid \sum_{r=1}^R w_r = 1, w_r \geq 0, r = 1, \dots, R \right\} \quad (3.48)$$

であり、(3.48) 式を満たす重み係数に対して、重み付け和最適化問題 (3.46) や最大成分最小化問題 (3.47) の最適解が Pareto 解になる。しかし、重み付け和最適化問題 (3.46) では、その目的関数が非凸の場合、任意の Pareto 解を与える重み係数が存在するのとは限らない欠点、重み付け最大成分最小化問題 (3.47) では、最大値関数の最小化のために勾配を用いた簡易的計算手法の適用が容易でない欠点が存在する。

ところで、複数の目的関数の関数形が陽に表現され、決定変数をそれに代入して評価を行う場合、計算の手間は比較的少ない。しかし、本論文で定式化した問題のように、計算機シミュレーションの実行結果・シミュレータや実システムの稼働結果を評価するとき、異なる評価対象が複数の存在する場合、それらすべての実行や稼働を1イテレーションの過程内においていわゆるバッチ処理で行うことは、著しく不効率になる場合がある。このことを問題(3.46)や(3.47)において具体的に当てはめると、調整則 \mathbf{F} を有するメタヒューリスティクス A を目的関数 E_r ごとに動作させることを評価対象の種類数の R 回繰り返し、それらの評価結果に対して重み付けして、問題(3.46)や(3.47)の目的関数値を求めて、それらを減少させるようにイテレーションを初めて更新することになる。

そこで、必ずしも多目的最適化問題の Pareto 解が得られる保証はないが、イテレーションごとに、複数の評価対象の中から単一の対象のみを選択し、それに対する評価値だけを改善するように決定変数を微小幅だけ更新することを、イテレーションの更新ごとに選択する評価対象を変更して繰り返す方式である。この場合、評価対象の選択は、複数の評価対象から無作為に選択する方式と、評価対象に付された番号順に選択する方式がある。本論文では、後者の方式を採用するものとする。多目的最適化問題(3.45)をメタ最適化問題として想定した場合、メタヒューリスティクスのアルゴリズム A のイテレーション番号 k と区別して、メタ最適化問題を解く評価対象逐次選択型解法のアルゴリズムのイテレーション番号にギリシャ文字 \mathcal{K} を用いると、以下のように記述することができる。

Step 1 評価対象の目的関数 E_1 を選択し、試行的な調整則 $F_{(1)}$ を設定し、 $\mathcal{K} = 0$ とおく。

Step 2 第 \mathcal{K} イテレーションに対応して $r = (\mathcal{K} \bmod R) + 1$ とおき、評価対象の目的関数 E_r を選択し、調整則 $\mathbf{F}_{(\mathcal{K})}$ を有するメタアルゴリズム A を目的関数 E_r に対して動作させる。すなわち

$$\begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k); E_r) \\ \mathbf{c}(k+1) = \mathbf{F}_{(\mathcal{K})}(\mathbf{c}(k), \boldsymbol{\chi}(k)) \end{cases}$$

$$k = 0, 1, \dots, K-1$$

を実行して解 $\mathbf{x}^*(K)$ を求め、その評価値 $J(\mathbf{x}^*(K); E_r)$ を得る。そして、この値を改善する新調整則 $\mathbf{F}_{(\mathcal{K}+1)}$ の発見を試みて、その成否に関わらず、次の Step へ行く。

Step 3 評価値 $J(\mathbf{x}^*(K); E_r)$ を改善する新調整則 $\mathbf{F}_{(\mathcal{K}+1)}$ が発見できれば $\mathcal{K} = \mathcal{K} + 1$ とし、そのような新調整則が発見できなければ、 $\mathbf{F}_{(\mathcal{K}+1)} = \mathbf{F}_{(\mathcal{K})}$ とし、Step 2 へ戻る。

このような評価対象逐次選択型解法では、イテレーション番号 \mathcal{K} の更新と評価対象として選択される目的関数の添字番号 r の更新が同期し、イテレーション番号が R 回更新されるごとに同じ目的関数が評価対象として選択されることが繰り返される。

3.7 遺伝的プログラミングによるメタ最適化問題の解法

前節で定式化した多目的メタ最適化問題(3.45)、あるいはその代替問題である単一目的関数メタ最適化問題(3.46)、(3.47)は、調整則を規定する関数 \mathbf{F} を生成する問題であり、

関数空間内でこの最適な関数を直接探索することは容易でない。そこで、何らかの近似関数を想定してこれを求めるいわゆる「関数近似問題」に置き換えて解くことが合理的である。パラメータ \mathbf{c} の次元 (すなわち調整したいパラメータの個数) を I 個とし、議論簡単化のために、関数 \mathbf{F} を単に $\mathbf{c}(k)$ から $\mathbf{c}(k+1)$ への関数とみなすと、関数近似問題に対しては、

- (1) 関数空間で互いに独立な M 個の基底関数 $f_m(\mathbf{c})$, $m = 1, \dots, M$ を用いて、

$$F_i(\mathbf{c}) = \sum_{m=1}^M \omega_{im} f_m(\mathbf{c}), \quad i = 1, \dots, I \quad (3.49)$$

と関数空間で1次結合し、この1次結合係数行列

$$\mathbf{\Omega} = \begin{pmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1M} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{I1} & \omega_{I2} & \dots & \omega_{IM} \end{pmatrix} \quad (3.50)$$

の最適な値を、たとえば最小自乗法を用いて決定する問題に置き換える。

という方法を採用するのが一般的である。この最小自乗法では、パラメータ \mathbf{c} の値の空間で、複数 (たとえば R 個の) サンプル点 $\mathbf{c}^{(r)}$, $r = 1, \dots, R$ とそれらの関数値 $\mathbf{F}(\mathbf{c}^{(r)})$, $r = 1, \dots, R$ の所望値 $\mathbf{d}^{(r)}$, $r = 1, \dots, R$ とが与えられ、評価対象とする R 個のサンプル点 $\mathbf{c}^{(r)}$, $r = 1, \dots, R$ に共通の自乗誤差関数

$$\begin{aligned} e(\mathbf{\Omega}; (\mathbf{c}^{(r)}, \mathbf{d}^{(r)})) &= \frac{1}{2} \left\| \sum_{m=1}^M \omega_{im} f_m(\mathbf{c}^{(r)}) - \mathbf{d}^{(r)} \right\|^2 \\ &= \frac{1}{2} \sum_{i=1}^I \left(\sum_{m=1}^M \omega_{im} f_m(\mathbf{c}^{(r)}) - d_i^{(r)} \right)^2, \quad r = 1, \dots, R \end{aligned} \quad (3.51)$$

を同時に結合係数行列 $\mathbf{\Omega}$ で最小化する多目的最適化問題を想定し、これを重みづけ和最適化問題に変換して解くのが最小自乗法である。このような考え方を用いているのが、ニューラルネットワークやラジアル基底ネットワークの学習問題や、サポートベクターマシンの識別問題といえる。

この近似方法を本論文において採用すると、調整則を規定する関数 \mathbf{F} をその関数空間で直接探索する代わりに、1次結合係数行列 $\mathbf{\Omega}$ の空間で探索することができるようになるが、本論文でのメタ最適化問題の場合は、1次結合係数行列 $\mathbf{\Omega}$ に対応して (3.49) 式で与えられる関数によって生成されるパラメータ時系列 $\{\mathbf{c}(k)\}$ が与えるメタヒューリスティクスアルゴリズム A による解 $\mathbf{x}^*(K)$ を関数 $J(\mathbf{x}^*(K); E_r)$ で評価することになる。具体的には、基底関数 $f_m(\mathbf{c}, \boldsymbol{\chi})$, $m = 1, \dots, M$ をまとめて、 $\mathbf{f}(\mathbf{c}, \boldsymbol{\chi}) = (f_1(\mathbf{c}, \boldsymbol{\chi}), \dots, f_M(\mathbf{c}, \boldsymbol{\chi}))^T$ とするとき、(3.49) 式に相当する近似式を

$$\mathbf{F}(\mathbf{c}, \boldsymbol{\chi}) = \mathbf{\Omega} \mathbf{f}(\mathbf{c}, \boldsymbol{\chi}) \quad (3.52)$$

と表して、これを問題 (3.45) に代入して得られる 1 次結合係数行列 Ω を求める多目的問題

$$\min_{\Omega} \begin{pmatrix} J(\mathbf{x}^*(K); E_1) \\ \vdots \\ J(\mathbf{x}^*(K); E_R) \end{pmatrix}$$

where $\mathbf{x}^*(\cdot)$ solution to

$$\begin{cases} \mathbf{x}(k+1) = A(\boldsymbol{\chi}(k), \mathbf{c}(k); E_r) \\ \mathbf{c}(k+1) = \Omega \mathbf{f}(\mathbf{c}(k), \boldsymbol{\chi}(k)) \end{cases} \quad (3.53)$$

$$k = 0, 1, \dots, K-1$$

$$\boldsymbol{\chi}(0); \text{ given}, \mathbf{c}(0); \text{ given}$$

$$r = 1, \dots, R$$

を想定することになるが、アルゴリズム A の実行によってその評価値 $J(\mathbf{x}^*(K); E_r)$ は定まるものの、その値が係数行列 Ω の関数として陽に表現できないことが大きな難点である。そこで、問題 (3.53) を解くためには、ヒューリスティックな手法を 1 次結合係数 Ω の空間で用いることになるが、もしメタヒューリスティクスを用いるとすると、そのためにもさらにパラメータ調整という本論文が提起している課題が、再帰的に生じることになる。また、この場合には、パラメータ \mathbf{c} と、 \mathbf{x} の時系列 $\boldsymbol{\chi}$ の空間でどのような基底関数 $f_m(\mathbf{c}, \boldsymbol{\chi})$, $m = 1, \dots, M$ を設定すればよいか不明である。仮にこれらを設定したとしても、それらの基底関数の 1 次結合で生成される関数の空間は必ずしも広いものとはいえない。

そこで、関数を近似的に生成する別の手法として、

- (2) 何種類かの数値の集合 $B = \{b_{m_1} | m_1 = 1, \dots, M_1\}$ や演算の集合 $O = \{+, -, \times, /\}$ および単純な関数の集合 $G = \{\exp, \log, \sin, \cos, \sqrt{\cdot}, \text{sig}\}$ などを用意し、それらの要素を組み合わせた木構造 $T_i(B, O, G)$, $i = 1, \dots, I$ によって関数 F_i , $i = 1, \dots, I$ を近似的に表現し、この木構造を更新して茂らせることによって表現する関数 F_i , $i = 1, \dots, I$ の近似精度をより高める。

という方法を採用する。パラメータ \mathbf{c} に関して定義された関数 $\mathbf{F}(\mathbf{c}) = (F_1(\mathbf{c}), \dots, F_I(\mathbf{c}))^T$ に対応して、複数の木構造をまとめて $\mathbf{T}(\mathbf{c}; B, O, G) = (T_1(\mathbf{c}; B, O, G), \dots, T_I(\mathbf{c}; B, O, G))^T$ とおき、要素の集合 (B, O, G) の \mathbf{T} による写像として関数 $\mathbf{F}(\mathbf{c})$ を近似的にみなし、

$$\mathbf{F}(\mathbf{c}) = \mathbf{T}(\mathbf{c}; B, O, G) \quad (3.54)$$

とおく。パラメータ \mathbf{c} の値の空間で、複数(たとえば R 個の)サンプル点 $\mathbf{c}^{(r)}$, $r = 1, \dots, R$ とそれらの関数値 $\mathbf{F}(\mathbf{c}^{(r)})$, $r = 1, \dots, R$ の所望値 $\mathbf{d}^{(r)}$, $r = 1, \dots, R$ とが与えられ、評価対象とする R 個のサンプル点 $\mathbf{c}^{(r)}$, $r = 1, \dots, R$ に対する関数近似のための共通の自乗誤差関数を考えるのであれば、

$$\begin{aligned} e(\mathbf{T}; (\mathbf{c}^{(r)}, \mathbf{d}^{(r)})) &= \frac{1}{2} \left\| \mathbf{T}(\mathbf{c}^{(r)}; B, O, G) - \mathbf{d}^{(r)} \right\|^2 \\ &= \frac{1}{2} \sum_{i=1}^I \left(T_i(\mathbf{c}^{(r)}; B, O, G) - d_i^{(r)} \right)^2 \end{aligned} \quad (3.55)$$

$$r = 1, \dots, R$$

と記述することができ、これらの自乗誤差関数を同時に木構造 \mathbf{T} の直接的な更新により最小化する多目的最適化問題を想定し、これを重みづけ和最適化問題に変換して解くことが、前述した最小自乗法に対応して考えることができる。このような近似方法は、最近ではたとえばランダムフォレスト^[65]とよばれる識別関数の構成法がある。

本論文では、多目的メタ最適化問題 (3.45) を想定するため、(3.54) 式の引数として \mathbf{c} に加えて χ も引数に入れて、(3.54) 式を問題 (3.45) に代入して得られる多目的問題

$$\min_{\mathbf{T}(\cdot)} \begin{pmatrix} J(\mathbf{x}^*(K); E_1) \\ \vdots \\ J(\mathbf{x}^*(K); E_R) \end{pmatrix}$$

where $\mathbf{x}^*(\cdot)$ solution to

$$\begin{cases} \mathbf{x}(k+1) = A(\chi(k), \mathbf{c}(k); E_r) \\ \mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \chi(k)) \end{cases} \quad (3.56)$$

$$= \mathbf{T}(\mathbf{c}(k), \chi(k); B, O, G)$$

$$k = 0, 1, \dots, K-1$$

$$\chi(0); \text{ given, } \mathbf{c}(0); \text{ given}$$

$$r = 1, \dots, R$$

を想定する。ただし、記述簡略化のため、木構造 $\mathbf{T}(\cdot)$ の要素を表す (B, O, G) は、木構造に常に共通であるため省略する。本論文ではこの問題を前節で言及した「評価対象逐次選択型解法」で解くために、木構造の空間において、そこで試行的にとられた木構造を更新する有力な計算手法として、遺伝的プログラミング (以降 GP と略記する) を用いることにする。GP は、木構造を変数とする最適化問題を解く唯一の計算手法とってよく、メタヒューリスティクスの一手法で多点型探索法であり、木構造の空間において複数種類の木構造が試行的に設定され、遺伝的アルゴリズムから派生した木構造に対する突然変異や交叉演算によって、新しい木構造の生成が繰り返される。したがって、その木構造に対応した関数が関数空間において直接的に生成され、関数近似において大きな自由度を有するものと考えられる。問題 (3.56) を想定し、「評価対象逐次選択型解法」として GP を適用した場合の計算手順は以下ようになる。

- Step 1** 評価対象とする最適化問題の目的関数の集合 $\{E_1, \dots, E_R\}$ を設定し、木構造の要素として数値の集合 B 、演算の集合 $O = \{+, -, \times, /\}$ 、 $G = \{\exp, \log, \sqrt{\cdot}, \text{sig}\}$ を用意する。
- Step 2** 遺伝的プログラミングの最大イテレーション K' を設定し、 \mathcal{P} 個 (複数個) の木構造 $\mathbf{T}^{(p)}(0)$ を初期化し、 $K = 0$ とおく。
- Step 3** 第 K イテレーションに対応して $r = (K \bmod R) + 1$ とおき、評価対象の目的関数 E_r を選択し、木構造 $\mathbf{T}^{(p)}(K)$ で表現した調整則 $\mathbf{F}_{(K)}$ を有するメタアルゴリズム A を目的関数 E_r に対して動作させる。
- Step 5** 評価結果をもとに、次世代の木構造 $\mathbf{T}^{(p)}(K+1)$ を生成する。

Step 6 $K \geq K'$ ならば計算を終了し，そうでなければ， $K = K + 1$ として Step 3 へ戻る．

なお，第4章では木から表現される関数 F の表記が具体的に与えられているが，この表記の仕方について注記しておく．たとえば，遺伝的プログラミングにおいて数式が木構造として表されている様子を示したのが Fig. 3.4 である．簡単な例は左の $1.0 + x$ であるが， $+$ と $-$ の演算子は単項としても使われており，同じ $1.0 + x$ の数式であっても木構造としては中央の構造を取ることも可能である．この場合の表記は，本論文では $1.0 - +(-x)$ となり，不等号は計算上は掛け算的に扱われる．この様な構造を取れることから突然変異により，一例として右の構造へと変異することが可能である．

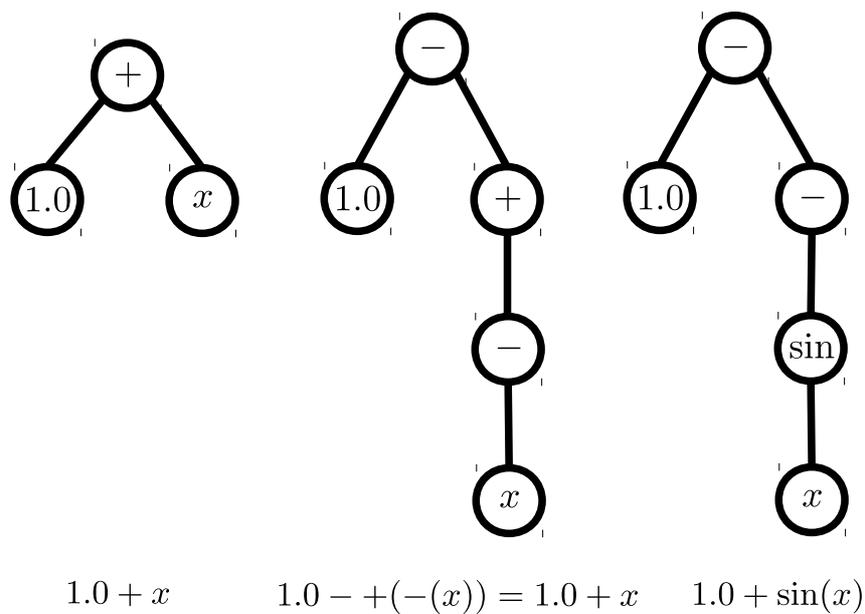


Fig. 3.4: Representation of expressions.

第4章 メタヒューリスティクスと その調整則の獲得

4.1 メタヒューリスティクスのパラメータ調整法のメタ最適化

本章では、メタヒューリスティクスの代表例として PSO, ES, DE, FA を取り上げ、それらの「パラメータ調整則」を遺伝的プログラミングによって進化的に獲得した結果を示す。いわば、本論文の核心部分の章である。アルゴリズムのパラメータ調整則を構築するために想定するメタ最適化問題として、前章で定式化したメタ最適化問題のうち多目的問題である問題 (3.45) を想定し、その中の調整則を獲得したい対象アルゴリズム A として、PSO, ES, DE, FA を適用する。本来ならば、メタヒューリスティクスのユーザにとっては、単一の解きたい問題に対してのみ、ソルバーから適当に選んだメタヒューリスティクスの調整則を進化させればよいが、ここでは解きたい問題を複数想定することで少しでも汎用性のある調整則を獲得するために、互いに異なる構造を有する目的関数を有する多目的問題をメタ最適化問題として想定する。PSO, ES, DE, FA の 4 種類の手法の調整則を GP で進化させた結果、獲得することができるこれら 4 種類の手法の計算性能について、4.3 節から 4.6 節まで順に考察する。したがって、各手法ごとのシミュレーションにおいて、以下の 2 段階の記述があることに注意されたい。

1. 各メタヒューリスティクスのアルゴリズムごとに、前章で導入した構造的に異なる 3 種の調整則を与える関数を、メタ最適化問題 (3.45) を想定して GP により進化的に獲得した結果について
2. 1. において進化的に獲得した複数種類の調整則を有する各メタヒューリスティクスごとに、COCO^[66] のベンチマークに適用して検証し、それらの検証結果を統計的に検定した結果について

以上の各手法のシミュレーションに関して、PSO は文献 [30][31][32]、ES は文献 [33][34]、DE は文献 [35][36] の手法の設計、FA は文献 [37] に基づくものであるが、これらの文献では、シミュレーション環境や GP のパラメータの設定、性能評価のための基準が異なるため、本論文ではそれらを統一して改めて計算した結果を示す。このような統一したシミュレーション環境や GP のパラメータ等については、次の 4.2 節でまとめてある。

ところで、進化的に獲得した調整則は、それを獲得するために用いた GP をブラックボックスとして、解きたい問題に適応した内部状態の木構造として自律的に獲得されていて、メタヒューリスティクスのユーザにとっては、陽に提示されるものではない。しかし、本論文では、手法ごとにまた調整則ごとに獲得された木構造を、念のため関数表現をすることにしていく。これらの木構造は、木のノードや深さに厳しい制限を設けてないために、比較的茂った複雑な構造となっている。このため調整則の関数表現も複雑な形となるが、

このような関数表現は、新しい調整則を付帯させたメタヒューリスティクスをユーザに供するという立場からすれば、好ましい関数形とはいえない。そこで獲得された調整則を規定する複雑な関数を簡単な関数で近似した調整則も、メタヒューリスティック手法ごとに提案し、これらをもって新しくかつ優れた計算性能を有するメタヒューリスティクスを提案し、さらにこれらに対して再度統計的検定を第4.7節で示すことで、もっとも優秀な手法を選びすぐっている。PSO, ES, DE, FA など、それぞれの手法ごとに改良が施され、それらの間で計算性能の比較がなされることはあっても、まったく異種の構造や特徴を有する手法の間で、横断的にそれらの計算性能を比較検討した例は少なく、またそれらの調整則を進化させて導出した新しい調整則を総合的に比較した研究も皆無である。

4.2 メタヒューリスティクスのパラメータ調整法のシミュレーション条件

調整則の遺伝子型といえる木構造を進化させる GP のパラメータは、本論文で対象とした PSO, ES, DE, FA の4種類の手法に対する GP のシミュレーションにおいて共通とし、次のように設定した。

(1) GP の探索個体数

$\mathcal{P} = 1,000$ 個体, すなわち GP の実行で用いた木構造の個体数が 1,000 種類

(2) GP の世代交代回数の上限

$\mathcal{K}' = 100$ 世代

(3) GP の試行回数

同一の調整則の引数, メタヒューリスティクスに対する GP の試行回数が 5 回

(4) 木構造の要素

非終端記号の演算子として $+, -, \times, /$, 関数として $\exp(\cdot), \sqrt{\cdot}, \text{sig}(\cdot) = \frac{1}{1+\exp(-\cdot)}$, 終端記号として実数, 0, 引数を使用。なお, 平方根の中身が負になる場合は, NaN(Not a Number) が返され, 結果として調整則が上手く動作せず, 大きい目的関数値となり性能が低い調整則であると評価される。

GP における探索個体数を $\mathcal{P} = 1,000$ 個体と大目に選んだ根拠は, 多様性の維持のため, そして一般的に GP の個体数 \mathcal{P} は数千個体とされるためであり^[24], 他にも文献 [20] では探索個体数 \mathcal{P} が 100,000 と大目に設定されていることなどを参考にした。最大世代数を $\mathcal{K}' = 100$ 世代としたのも比較的よく設定される最大世代数であり, これらの設定であればシミュレーションが 1 日程度で終わるという時間的な理由によるものでもある。GP の試行回数が 5 回なのも同様に時間的な理由によるものである。文献 [20] をはじめとする多くの GP の文献では, 最終的な成果物に関心が向いており, 終了条件が人間の判断となっていたり, 何回試行した結果, 得られた解であるかが不明瞭であることが多い。試行回数 5 回は統計的に我々の手法が良いかを議論するには少なく, **Selective inference**^[67](研究者は統計的検定が通る結果を探索して発表する傾向にあるため, 再現を行うと p 値から推定さ

れるより多くの発表が再現できないバイアス)の潜在的な統計的問題もあるが、少なくともどの程度のシミュレーションを行ったのかの参考になるため表記した。

次に GP の木構造の探索個体 1 個に対応して実行される PSO, ES, DE, FA の 4 種類の各アルゴリズムにおいても、

- (5) パラメータ調整則を GP で進化させるメタヒューリスティクスの探索点数
探索点数 $P = 20$ (ベンチマーク問題の変数次元に関係なく一定)
- (6) GP で進化させるメタヒューリスティクスによる目的関数 E の呼び出し回数の上限
最大イテレーション $K = 1,000$ とし, $K \times P = 20,000$ 回
- (7) 調整則 (c) を用いてパラメータ調整するときの引数である更新頻度 $p_s(k)$ の算出
現イテレーション k より $10N$ イテレーションまで遡って算出 (N は問題の変数次元)
- (8) 進化におけるメタヒューリスティクスの試行回数
すべての問題に対して 1 回
- (9) GP で進化させるメタヒューリスティクスが解く最適化問題の変数次元 N
進化においては $N = 20$ と固定
- (10) GP で進化させるメタヒューリスティクスが解く最適化問題の目的関数 E_1, \dots, E_R
目的関数については, $R = 4$ とし, E_1, \dots, E_R を順に 20 度回転させた Rastrigin 関数, Alpine 関数, Levy 関数, そして回転させていない Rosenbrock 関数とし, メタ最適化問題 (3.45) を想定

と共通にし, 調整則を進化させた. なお, (6) の条件が各メタヒューリスティクスのアルゴリズム A の終了条件であるが, アルゴリズムの性格上この終了判定条件を満たしていても, 大域的最適解は勿論のこと, 局所的最適解ですら得られている保証はない. そこで, この終了判定条件をみたしたイテレーション時までに見つけた最良の目的関数値を与える点のその値を, GP の探索個体である木構造に対応した調整則を有するアルゴリズムの評価値とし, GP の探索個体の適応度と見なすこととした. これらの目的関数の探索範囲は一般的に $-5.0 \leq x_n \leq 5.0$ とされることが多く, (a) のタイプのアルゴリズム A は座標系のスケールに依存するため, この制約の範囲を平行移動とリスケーリングによって $0.0 \leq x_n \leq 1.0$ に変換する. 初期値は全てのアルゴリズム A においてこの範囲に一様分布で配置され, アルゴリズム A が解く問題はこの制約に限定されることを想定することによりこの制約に最適な調整則を進化させ, 座標系の違いによるアルゴリズム A の性能変化を担保する. 新たな問題に関してはこの探索範囲に沿う様に変換を行う.

得られた調整則の最終的な評価には, ベンチマーク集である COCO^[66] にある関数のうち, ノイズなし関数であるインスタンス番号 1 の関数 $f_1 \sim f_{24}$ を目的関数として使用する. 問題は大きく分類すると,

$f_1 \sim f_5$:

変数分離可能な関数群で, このうち f_1 は Sphere 関数である. これら以外は, 変数分離不可能の問題である.

$f_6 \sim f_9$:

Low or moderate conditioning とよばれる低ないしは中程度の悪条件の関数群で、最適解付近の窪みが等方向的でない細長い溝になっていて、 f_8 が Rosenbrock 関数、 f_9 がこれを回転させた関数である。

$f_{10} \sim f_{14}$:

High conditioning and unimodal とよばれる、単峰性でありながらより強い悪条件の関数群で、 f_{10} は f_2 を回転させた関数である。

$f_{15} \sim f_{19}$:

Multi-modal functions with adequate global structure とよばれる、近接最適性の原理 (Proximate Optimality Principle: POP) がある程度成立する多峰性の形状をした関数群である。

$f_{20} \sim f_{24}$:

Multi-modal functions with weak global structure とされる目的関数で、多峰性の形状であってしかも POP の成立が弱い関数群である。

となる。

本論文では、これらのベンチマークに加えて、POP が理想的に成立し、大域的最適解への到達度の厳密な評価が可能な関数として、(3.31) 式のフラクタル関数もベンチマークに加え、この関数を 20 度回転させたものを f_{25} としてベンチマークに加えることとした。COCO のベンチマークは探索範囲が $-5.0 \leq x_n \leq 5.0$ とされているため、その範囲を平行移動とリスケーリングによって $0.0 \leq x_n \leq 1.0$ に変換した。変数 \mathbf{x} の次元は、進化させた環境と異なる場合でも動作することを確認するため $N = 30$ とした。これらの問題を $P = 20$, $K = 1,000$ と、GP でパラメータ調整則を進化させた際と同じ設定でアルゴリズムに解かせる。

ところで、メタヒューリスティクスはいずれもいわゆる確率的探索法であるため、これらの目的関数を用いた評価においては、すべてのベンチマークに対して、各アルゴリズムともに 50 回試行し、2 章におけるアルゴリズムの評価と同様に計算結果から、任意の 2 種のアルゴリズムの組合せの併合順位を算出し、Welch の t 検定で順位の平均値を検定する。なお、検定の詳細な結果や説明は紙面の都合上、付録に載せることとし、本章にはどのアルゴリズムが統計的に有意であったかの比較結果のみ示す。

以上の評価における設定をまとめると

- (11) 評価におけるメタヒューリスティクスの探索点数
探索点数 $P = 20$ (ベンチマーク問題の変数次元に関係なく一定)
- (12) 評価におけるメタヒューリスティクスによる目的関数 E の呼び出し回数の上限
最大イテレーション $K = 1,000$ とし、 $K \times P = 20,000$ 回
- (13) 調整則 (c) の更新頻度引数である更新頻度 $p_s(k)$
現イテレーション k より $10N$ イテレーションまで遡って算出 (N は問題の変数次元)

(14) メタヒューリスティクスの試行回数

すべての問題に対して 50 回

(15) 評価における問題の変数次元 N

評価においては $N = 30$ と固定

(16) 目的関数 E

COCO の $f_1 \sim f_{24}$, そしてフラクタル関数を 20 度回転させた f_{25}

となる. 以降の各メタヒューリスティクスにおけるシミュレーションの節においては, 上記の方法により 3.5 節の (a), (b), (c) 各 5 回ずつシミュレーションを行い得られた調整則の中で, 各設定における最良調整則の数式を示す. 有意水準を 5% と設定し, 3 つの組み合わせの検定を 25 関数に対して行うことから, Bonferroni 法により P 値が $0.05 / ({}_3C_2 \times 25)$ より小さい場合は, 統計的に有意な差があるとする. (a), (b), (c) のうちどれが最良な数式かは比較検定を行った上で検討するが, この詳細な検定結果の部分は論文の付録に添付することにし, 本章では検定でどの部分が統計的に有意であったかのみを示す.

4.3 Particle Swarm Optimizationのパラメータ調整法のメタ最適化

4.3.1 Particle Swarm Optimization とその調整則の設計

本節では、前章で定式化した最適調整則獲得のためのメタ最適化問題のうち多目的問題(3.45)を想定し、最適な調整則を獲得したい対象アルゴリズム A として PSO を適用し、GP を用いて獲得された調整則を示す。2.2 節において、PSO のもっとも基本的な更新則とその改良手法、およびその更新則中のパラメータ更新則で、その性能が評価されている各種の更新則を示し、計算機シミュレーションによる結果に対して統計的検定を行うことで、それらの計算性能を比較した。本節では、様々な研究者によって提案された改良手法ではなく、もっとも基本的な PSO の更新則である 2 章 (2.1) 式

$$v_n^{(p)}(k+1) = c_0 v_n^{(p)}(k) + c_1 r_{1n}^{(p)}(k)(x_n^{(p\text{-best})}(k) - x_n^{(p)}(k)) \\ + c_2 r_{2n}^{(p)}(k)(x_n^{(g\text{-best})}(k) - x_n^{(p)}(k)), n = 1, \dots, N$$

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

$$\mathbf{x}^{(p\text{-best})}(k) = \underset{\mathbf{x}^{(p)}(i)}{\operatorname{argmin}} \{E(\mathbf{x}^{(p)}(i)) | i = 0, 1, \dots, k\}$$

$$\mathbf{x}^{(g\text{-best})}(k) = \underset{\mathbf{x}^{(q\text{-best})}(k)}{\operatorname{argmin}} \{E(\mathbf{x}^{(q\text{-best})}(k)) | q = 1, \dots, P\}$$

を考える。基本的な更新則のパラメータ調整を考えるのは、メタ最適化問題の想定とそれへの GP の有効性を確認するためである。この中のパラメータ c_0, c_1, c_2 を時変パラメータと見なし、

$$\mathbf{c} = (c_0, c_1, c_2)^T \tag{4.2}$$

とにおいて、これらの調整則を 3.5 節で与えた (3.42)~(3.44) 式の 3 種類の再帰型調整則

(a) 探索点変位および最良点との距離による調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), \|\mathbf{v}(k)\|, \Delta_g(k)) \tag{4.3}$$

(b) イテレーション経過割合による調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), t(k)) \tag{4.4}$$

(c) 目的関数改善割合による調整則

$$\mathbf{c}(k+1) = \mathbf{F}(\mathbf{c}(k), p_s(k)) \tag{4.5}$$

を与える関数 $\mathbf{F} = (F_1, F_2, F_3)^T$ を、多目的メタ最適化問題(3.45)を想定して、それぞれ 3 種類の調整則ごとに GP を用いて自動的に生成した結果を示す。(a) は 2 章で紹介した NDT-PSO の調整則(2.17)式を一般化したものといえ、(b) はやはり 2 章で紹介した LDIW-PSO の調整則(2.12)式を一般化したものといえる。

4.3.2 Particle Swarm Optimization の調整則の進化結果

関数の具体的な数式を，GPにより進化的に獲得された木構造を基に表現すると以下のとおりである．GPによって木構造の形で獲得されたパラメータ調整則の内部状態の表現型であり，5回のGPの試行により，各試行における最終世代の個体の中で，最良の目的関数値を与えるものを示してある．また，特筆すると(a)の調整則の $c_1(k)$ は表現型の上では調整則となっているが，引数による影響が退化しており，実際にこの調整則を問題に適用すると常に約1.75243の定数値を返す．

(a) 探索点変位および最良点との距離による調整則

$$\begin{aligned}
 c_0(k+1) &= \text{sig}((\exp(\text{sig}(\text{sig}(0.0))) - \text{sig}(\text{sig}(\text{sig}(\text{sig}(0.813917))))) \approx 0.769249 \\
 c_1(k+1) &= (\text{sig}(\exp(\text{sig}(\text{sig}(\exp((\text{sig}(\exp((\text{sig}(\text{sig}(0.776912)) - \\
 &\quad \exp(\exp(\text{sig}((0.701999 - c_1(k)))))))))) \times ((0.175895 - \text{sig}(((0.706472 - \\
 &\quad \text{sig}(\exp(((0.706472 - \text{sig}(\text{sig}(\exp(((-0.354804 - \text{sig}(-(0.877894))) - \\
 &\quad ((0.706472 + 0.877894) \times \|\mathbf{v}(k)\|(k))) \times +((-0.175895 + \\
 &\quad ((\text{sig}(\text{sig}(\|\mathbf{v}(k)\|(k) + \exp(0.0)))) + (0.877894)))))))))) - 0.776912) - \\
 &\quad 0.780449))) + (0.877894)))))) + (0.877894) \approx 1.75243 \\
 c_2(k+1) &= (((-((0.604364/(0.0 + \sqrt{\|\mathbf{v}(k)\|})) \times (\Delta_g(k) - +((- (\|\mathbf{v}(k)\|/ \\
 &\quad (-0.374592 + ((\Delta_g(k) + (0.0610851 + 0.460483))/\sqrt{\|\mathbf{v}(k)\|})) \times \\
 &\quad ((-0.948512 + (0.0 - (-0.158034 - (\exp(0.801719) - ((-\text{sig}(- \\
 &\quad ((\text{sig}(-0.159738)/(-0.632893 \times \|\mathbf{v}(k)\|)))) \times \Delta_g(k) + \\
 &\quad 0.928146)))) \times 0.0)))) + (0.0 - (0.604364/(- (\|\mathbf{v}(k)\|/(0.0610851 + \\
 &\quad (1.12593/(\|\mathbf{v}(k)\| - \sqrt{\sqrt{\|\mathbf{v}(k)\|}))))) - \sqrt{\|\mathbf{v}(k)\|})))))
 \end{aligned} \tag{4.6}$$

(b) イテレーション経過割合による調整則

$$\begin{aligned}
 c_0(k+1) &= 0.844358 \\
 c_1(k+1) &= \sqrt{\sqrt{\sqrt{\sqrt{\text{sig}(c_1(k) + c_1(k))}}}} \\
 c_2(k+1) &= \exp(((0.37817 + (-0.0261027/(c_2(k) - (t(k)))))/(2.15234 \times \\
 &\quad (\exp(((\exp(c_2(k)) \times 0.398039) + (-0.0261027/0.876019))/ \\
 &\quad (((-\exp(0.5) \times 0.600798) - (+t(k))) \times -0.876019)) + \\
 &\quad ((-0.252496 + \text{sig}((\sqrt{0.662518}/0.37817))) + 0.896093))))))
 \end{aligned} \tag{4.7}$$

(c) 目的関数改善割合による調整則

$$\begin{aligned}
 c_0(k+1) = & \text{sig}((((0.78591 - (p_s(k) \times ((0.860709 \times -((\text{sig}(-0.36017)/((-0.0592678 - \\
 & 1.29494)/0.0)))))/(((0.440813 - (((\text{sig}(p_s(k)) \times 0.181351) + \\
 & 0.773033) + 0.197741) + (c_0(k))) + ((0.842895 \times p_s(k)) - \\
 & 0.00222484))) + ((\sqrt{2.61864} \times (-0.000980738 + (0.683255 - p_s(k))) \times \\
 & p_s(k)))/(-0.0592678/p_s(k))) - \text{sig}(p_s(k)))))) + 0.773033) - \\
 & (\text{sig}((0.704643 + (p_s(k) \times (\text{sig}(((p_s(k) - 0.00222484) \times ((p_s(k) \times \\
 & ((-0.837862 \times p_s(k)) \times -0.0768135)) - (p_s(k) - ((+(0.0 \times (\text{sig}(0.0)/ \\
 & ((-0.0592678 - p_s(k))/p_s(k)))))) \times -0.760866)))) - (\text{sig}((p_s(k) \times \\
 & -0.0592678))/\text{sig}((p_s(k) \times -0.0592678)))))) + 0.773033) \times \\
 & (-0.0768135)))/((-0.0592678 - p_s(k))/p_s(k)))) \\
 c_1(k+1) = & (\text{sig}((\text{sig}(0.967497) - \text{sig}(\sqrt{\text{sig}((0.278377 + \text{sig}(0.213981))))})) + 0.0) \\
 c_2(k+1) = & \text{sig}(\exp(\text{sig}((((0.0 - 0.137816) - 0.831617) \times +(\sqrt{c_2(k)})))
 \end{aligned} \tag{4.8}$$

これらの木構造を獲得するまでの GP の世代更新に伴う目的関数値の進化過程を 100 世代まで記した様子を Fig. 4.1~Fig. 4.3 に示す. メタ最適化問題として多目的問題 (3.45) を想定し目的関数の種類として $R = 4$ とし, 目的関数 $E_r(\mathbf{x}) = 1, \dots, R$ として, Rastrigin 関数, Alpine 関数, Rosenbrock 関数, Levy 関数を用い, 実際の GP の更新では世代ごとにこの順番で適応度とする関数を交互に交換する方式を採用し, GP の個体に対応する木構造で記述された調整則のうち最良の個体に対する目的関数値をプロットしてある. 4 種類の目的関数を交互に使用しているため, 100 世代では各目的関数が問題として 25 回使用される. また, 関数値のスケールが大きく異なるため片対数グラフを使用しており, 減少幅が少なく表示されている. GP の個体に対する調整則が与える目的関数値が世代更新とともに増減を繰り返しながら, 20 世代ぐらいまでにある程度減少しているプロットが多いことが確認され, とくに調整則 (c) の結果が顕著な進化の過程を示していることがわかる.

なお, プロットの各点は最適化アルゴリズムをそれぞれの木構造で与えられる調整則のもとで 1 回試行した結果である. そのため, 世代が進むにつれて調整則としては改善していたとしても, 図の上では乱数で与えられる初期点の影響により必ずしも改善が確認できるものではない. また, 目的関数ごとに得られる解の分布やスケールも異なるため, グラフが横ばいであっても実際には改善が起きている可能性があることを注記しておく.

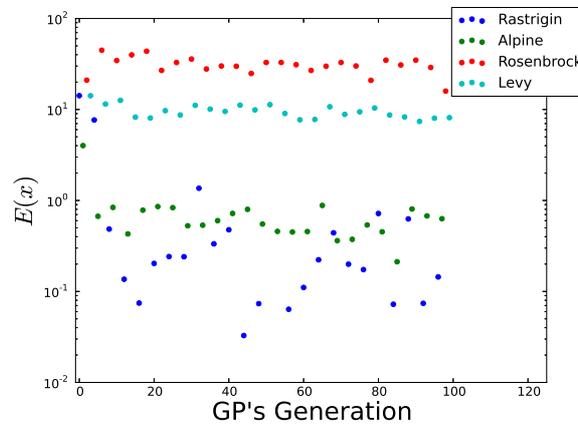


Fig. 4.1: $E(x)$ of Genetic Programming during evolving rule for PSO on setting (a).

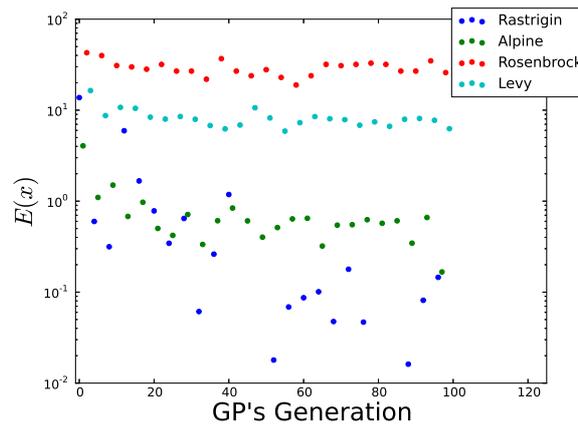


Fig. 4.2: $E(x)$ of Genetic Programming during evolving rule for PSO on setting (b).

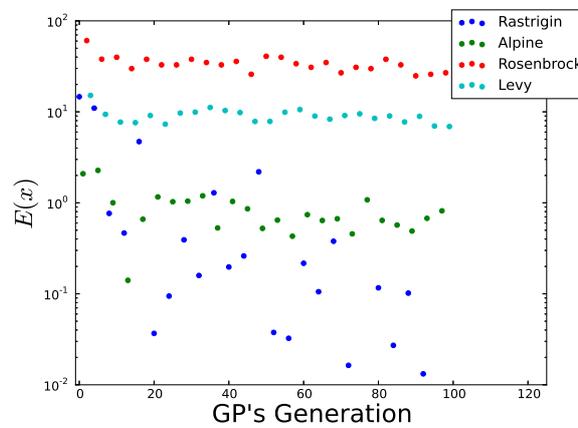


Fig. 4.3: $E(x)$ of Genetic Programming during evolving rule for PSO on setting (c).

(a)~(c)を比較検定した結果を Table 4.1 に示す. 結果の L と R の列から, (a) の設定において進化させた調整則がもっとも良いことがわかり, 特に, $f_1 \sim f_5$ の変数間依存性のない問題に対して強いことがわかる. これを解明するため, 調整則 (a) の近似を行い簡単な数式として表現し, 実装を簡単にしつつ, 数式の持つ意味を理解できる形に変換することを考える.

調整則 (a) を Rastrigin 関数に対して, 1,000 イテレーション動作させた結果, $c_0 = 0.769$, $c_1 = 1.752$ と定数であることが判明した. このことから $c_2(k)$ が実質的な (a) 式の調整則であることがわかる. そこで $\|\mathbf{v}(k)\|, \Delta_g(k), 1/\Delta_g(k)$ の3変数の線形結合を用いて特定の1,000 イテレーションの $c_2(k)$ を, 次の問題を解くことにより近似する. これらの変数を用いるのは, 著者らの研究^{[30][31][32]}において, これらによる簡単な調整則の数式を探索したことを参考にしてである. また, 対数スケールを用いるのは大きな値と小さな値を高いコントラストで近似するためである.

$$\min_{\mathbf{w}} \sum_{k=1}^{1000} (w_0 + w_1 \|\mathbf{v}(k)\| + w_2 \Delta_g(k) + w_3 / \Delta_g(k) - \ln(c_2(k)))^2 \quad (4.9)$$

これにより近似されるパラメータ調整則は

$$c_0 = 0.769 \quad (4.10a)$$

$$c_1 = 1.752 \quad (4.10b)$$

$$c'_2(k) = \exp(0.6484 - 1.025 \|\mathbf{v}(k)\| - 1.182 \Delta_g(k) + 0.02597 / \Delta_g(k)) \quad (4.10c)$$

であり, 実際の $c_2(k)$ のパラメータ値と, 近似した $c'_2(k)$ のパラメータ値 1,000 個の散布図は Fig. 4.4 となる. 元の調整則の出力と, 近似した調整則の出力のピアソン相関係数は 0.986 と大きく, 良い近似であると考えられる.

(4.10c) 式よりわかることは, 群が密集していない状態では ($\Delta_g(k)$ 大), $c_2(k) = 0.0$ とし, 定数 c_0, c_1 の慣性と p -best への移流で駆動し, 群が密集している状態では ($\Delta_g(k)$ 小), $0.02597 / \Delta_g(k)$ の項が大きくなり, 指数で増幅され, 結果として発散状態へ移行することがわかる. これはメタヒューリスティクスでいう「集中化と分散化」の概念を実装したものであり, 概念からアルゴリズムを設計したのではなく, 最適なパラメータ調整則を探索した結果, この概念を導出した結果となっている. ただし, これにより性能が向上するのは変数分離可能な問題に限定されているため, この概念は万能ではないことがわかる.

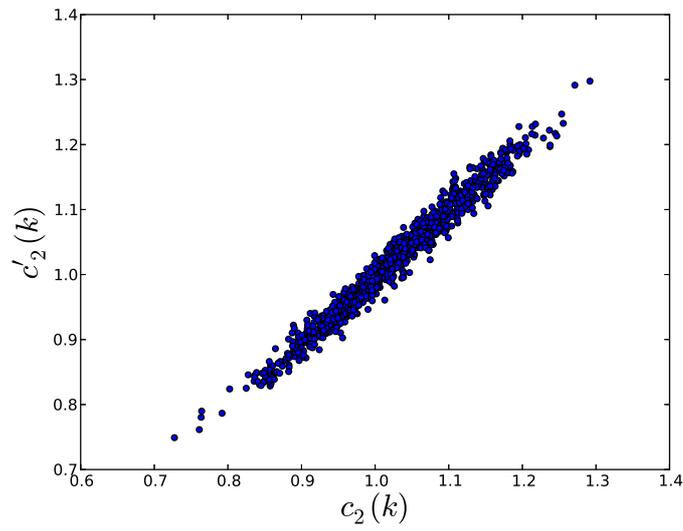


Fig. 4.4: Scatter plot of $c_2(k)$ and approximated tuning rule.

Table 4.1: Pairwise ranked Welch's t-test for 30 dimensional functions with 20,000 function calls of evolved PSO

alg1	alg2	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19	f20	f21	f22	f23	f24	f25	L	R
(a)	(b)	L	L	L	L	L	N	L	N	N	R	N	N	N	R	N	N	N	N	R	N	N	N	N	R	L	7	4
(a)	(c)	L	L	L	L	L	N	L	N	N	R	N	N	N	N	N	N	N	N	N	N	N	N	R	R	L	7	3
(b)	(c)	N	L	N	L	N	L	N	N	N	N	N	L	N	L	N	N	N	N	N	N	N	N	R	R	N	5	2

4.4 Evolution Strategy のパラメータ調整法のメタ最適化

4.4.1 多点型 Evolution Strategy とその調整則の設計

前節の PSO に対する調整則の設計においては、本来定数とされているパラメータをアルゴリズムの更新状態に対応して動的に変化させるようにしたため、その計算性能が改善されることはある意味自明であるといえる。これに対して、本節で扱う進化戦略 (Evolution Strategy: ES) は、最初に提案された (1+1)-ES とよばれる単点型アルゴリズム自体に、もともと「1/5 ルール^[10]」とよばれるパラメータ調整則が付帯している手法である。2.3 節で解説したように、これを出発点とする初期の ES の発展形は (μ, λ) -ES となり、1/5 ルールのようなパラメータ調整則が用いられなくなった。そこで本節では、(1+1)-ES の探索点数を単純に多点化した更新式

$$\mathbf{u}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \sigma(k)\mathbf{N}^{(p)}(\mathbf{0}, \mathbf{I}) \quad (4.11a)$$

$$\mathbf{x}^{(p)}(k+1) = \begin{cases} \mathbf{u}^{(p)}(k+1), & \text{if } E(\mathbf{u}^{(p)}(k+1)) \leq E(\mathbf{x}^{(p)}(k)) \\ \mathbf{x}^{(p)}(k), & \text{otherwise} \end{cases} \quad (4.11b)$$

$$p = 1, \dots, P$$

をアルゴリズム A として用い、この多点型の (1+1)-ES のための 1/5 ルールに相当するパラメータ調整則の設計を考える。ここでは、1 イテレーションの経過に伴って P 個の探索点が更新されて P 個の探索点が生成されるという意味で、 $(P+P)$ -ES と呼称することができる。

本節では、(4.11) 式中の時変パラメータ $\sigma(k)$ を、3.5 節で与えた (3.42)~(3.44) 式の 3 種類の再帰型調整則

- (a) 探索点変位および最良点との距離による調整則

$$\sigma(k+1) = F(\sigma(k), \|\mathbf{v}(k)\|, \Delta_g(k)) \quad (4.12)$$

- (b) イテレーション経過割合による調整則

$$\sigma(k+1) = F(\sigma(k), t(k)) \quad (4.13)$$

- (c) 目的関数改善割合による調整則

$$\sigma(k+1) = F(\sigma(k), p_s(k)) \quad (4.14)$$

を与える関数 F を、多目的メタ最適化問題 (3.45) を想定して、(a)~(c) の三つの場合を想定して多点型 ES の調整則を GP で自動的に生成した結果を示す。なお、PSO の場合とは異なり、調整パラメータは 1 種類のため、調整則を与える関数 F はスカラー関数である。PSO に対しては (a) の構造の調整則を進化させた結果が他の構造の調整則と比べて性能がよく、PSO には (a) の調整則が適していることが判明したが、PSO と同様の条件下で多点型 ES に対するこれらの調整則の適性を確認する。文献 [33][34] では、ES を単純に多点化することで、その大域的な探索性能が向上することを確認したうえで、調整則 (a) と (c) の性能について統計的検定により確認しているが、ここでは新たに (b) の性能についても確認する。

4.4.2 多点型 Evolution Strategy の調整則の進化結果

多点型 ES に対して進化的に獲得された調整則の具体的な数式を、GP により進化的に獲得された木構造を基に表現すると以下のとおりである。GP によって木構造の形で獲得されたパラメータ調整則の表現型であり、5 回の GP の試行により、各試行における最終世代の木構造の個体数の中で、最良の目的関数値を与えるものである。結果を評価したところ、Table 4.8 となり、(b) の設定において進化させた調整則がもっとも良いことがわかる。

(a) 探索点変位および最良点との距離による調整則

$$\begin{aligned} \sigma(k+1) = & (0.108486 \times \text{sig}(\text{sig}((-0.255446 + (0.38825/((0.00871860/ \\ & ((0.00884750 \times (-0.0757776 - (\sigma(k) + \|\mathbf{v}(k)\|)))) \times -1.111744) + \\ & 0.262112)) + 0.550570)))))) \end{aligned} \quad (4.15)$$

(b) イテレーション経過割合による調整則

$$\begin{aligned} \sigma(k+1) = & (\text{sig}(((((-0.230945 - ((\text{sig}((\text{sig}((\text{sig}(t(k)) + (\text{sig}(\sigma(k)) + ((-0.510822 - \\ & 0.0913668) \times -((t(k) + 0.288204) + 0.0568247)))))/(-((0.362239 - \\ & t(k)))/0.134624))) + (t(k)/ - 0.230945))) - 0.143257) \times 0.134624)) \times \\ & 0.012485) + 0.684864) - 0.130607) + ((t(k) + 0.684864)/ - 0.230945)))/ \\ & (((-0.0666374 - (t(k) + t(k))) - (((0.0366189 - \text{sig}(((0.0969042 + \\ & \text{sig}((-0.827956 \times ((-0.93084 - ((\text{sig}(((0.0 + \exp((0.0 - t(k)))) - \\ & \text{sig}(t(k) + (-0.330026)))) \times -0.246373)) + (-0.230945 \times t(k))) - \\ & (((t(k) + 0.684864)/(\text{sig}(t(k)) + ((0.0 - (t(k) + 0.684864)))))) \times \\ & (t(k) - 0.493738)) \times 0.67615))) \times -0.323205))) - (((t(k) + 0.203627) + \\ & \sigma(k)) \times (\text{sig}(\sqrt{t(k)}((- \sqrt{+}0.439565))) + (\text{sig}(+(t(k))) - 0.152922))) - \\ & (0.0 - (\sigma(k) \times t(k)))))) - 0.143257) \times 0.134624) - \sqrt{t(k)}(0.418429)) \end{aligned} \quad (4.16)$$

(c) 目的関数改善割合による調整則

$$\begin{aligned} \sigma(k+1) = & ((p_s(k) - ((0.110783 + ((p_s(k) + 0.0349683) - p_s(k))) + 0.0)) + \\ & ((p_s(k) + 0.0614613) - p_s(k))) \end{aligned} \quad (4.17)$$

これらの木構造を獲得するまでの GP の世代更新に伴う目的関数値の進化過程を 100 世代まで記した様子を Fig. 4.5~Fig. 4.7 に示す。メタ最適化問題として多目的問題 (3.45) を想定し、PSO に対するのと同様に目的関数 $E_r(\mathbf{x})$, $r = 1, \dots, 4$ として、Rastrigin 関数、Alpine 関数、Rosenbrock 関数、Levy 関数を用い、実際の GP の更新では世代ごとにこの順番で適応度とする関数を交互に交換する方式を採用し、GP の個体に対応する木構造で記述されたパラメータ調整則のうち最良の個体に対する目的関数値をプロットしてある。片対数グラフを使用しても、Fig. 4.6 において目的関数値調整則 (b) を用いた結果が顕著な

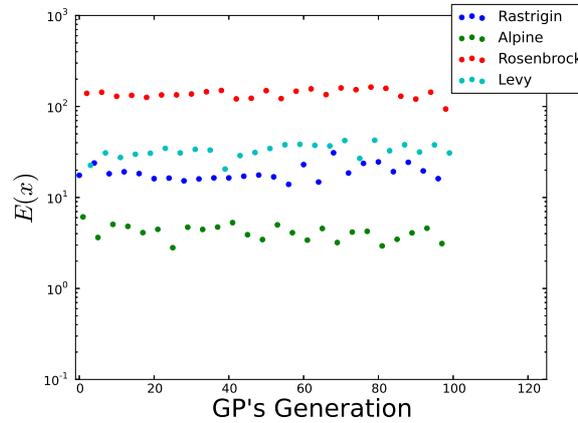


Fig. 4.5: $E(x)$ of Genetic Programming during evolving rule for ES on setting (a).

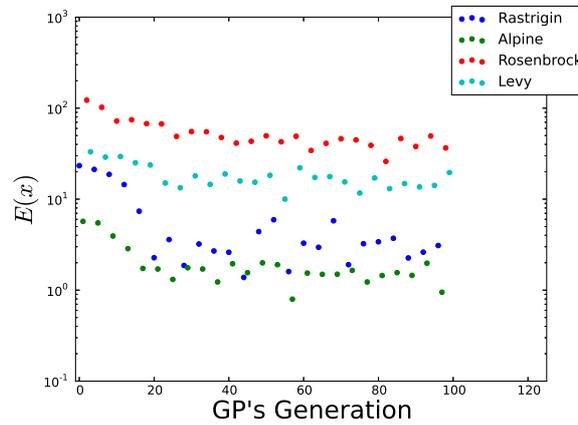


Fig. 4.6: $E(x)$ of Genetic Programming during evolving rule for ES on setting (b).

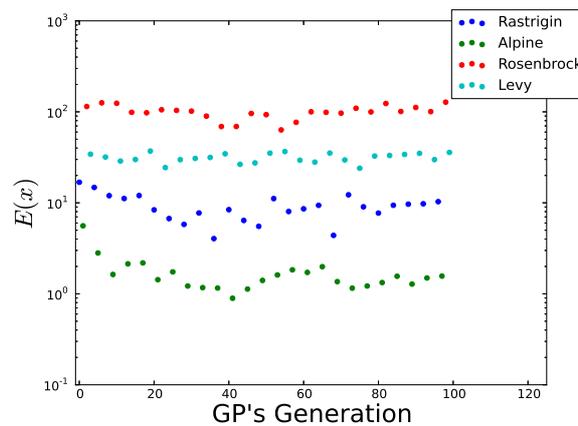


Fig. 4.7: $E(x)$ of Genetic Programming during evolving rule for ES on setting (c).

進化の過程を示しており、(b)の構造を有する調整則が多点型ESに適しているものと判断される。

(b)の構造の調整則を有する多点型ESでは、更新則中のステップ幅といえる分散共分散行列の大きさ $\sigma(k+1)$ の調整だけに限定し、しかも複数の探索点でこのパラメータを共有している。複数探索点がこのパラメータを共有することで、それらが干渉しているといえるが、パラメータの種類もそれを与える関数の引数も1種類のため、その大域的探索能力は限定的といえる。著者らの文献[34]では、(a)と(b)の構造を同時に考慮した3種類の引数を有する関数による調整則

$$\sigma(k+1) = F(\sigma(k), p_s(k), \|\mathbf{v}(k)\|, \Delta_g(k)) \quad (4.18)$$

を想定し、これをGPで進化させた調整則を有する多点型GPを得ることに成功している。

(a)~(c)の調整則を比較検定したのがTable 4.8であり、このLとRの列からも、(b)の手法が他の調整則と比べて良いことがわかるため、(b)の手法の近似を考える。(b)の数式は複雑であるが、この数式はイテレーション k とともに変化し、問題によらず一意の曲線を描く。そこで近似においては、このグラフを対数スケールで直線近似する。対数スケールを用いるのは、 σ が正規分布乱数の幅を決定するパラメータであるためであり、大域的な探索と局所的な探索を同等に行うべきであるというフラクタル性を考慮してである。近似した数式は

$$\sigma(k+1) = 0.1102\exp(-5.629t(k)) \quad (4.19)$$

となり、これを元の曲線と比較するとFig. 4.8のようになり、良く近似できていると考えられる。(b)の手法を近似した(4.19)式によると、時間と共にパラメータを小さくしていくと、本アルゴリズムでは調整が上手くいくと考えられる。これは「集中化と分散化」の概念と異なるものであり、焼き鈍し法^[68]に使われるアニーリングの概念に近く、「継続的な集中化」が本アルゴリズムに適した概念であると言える。

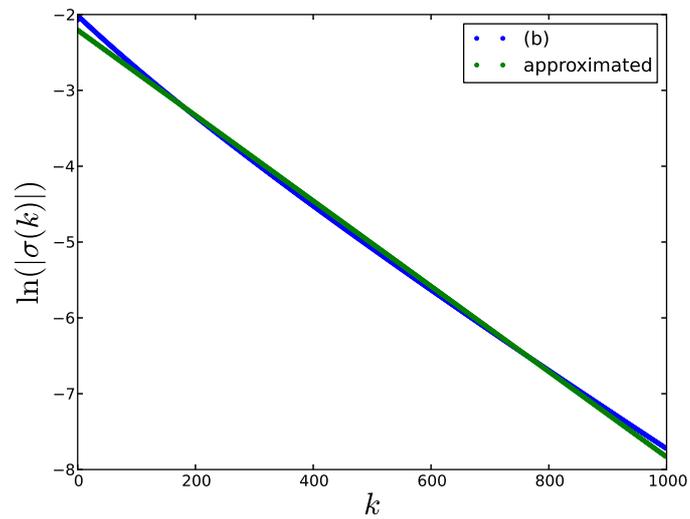


Fig. 4.8: Scatter plot of (b) and approximated tuning rule.

Table 4.2: Pairwise ranked Welch's t-test for 30 dimensional functions with 20,000 function calls of evolved ES

alg1	alg2	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19	f20	f21	f22	f23	f24	f25	L	R
(a)	(b)	R	R	R	N	R	R	N	R	R	R	N	R	R	R	N	R	N	N	N	R	R	R	R	R	N	0	17
(a)	(c)	R	R	L	L	N	R	N	R	R	R	N	R	R	R	L	R	L	L	L	R	R	R	R	L	N	7	14
(b)	(c)	L	N	L	L	N	N	N	N	L	N	N	L	L	L	L	L	L	L	N	N	N	N	L	L	N	13	0

4.5 Differential Evolution のパラメータ調整法のメタ最適化

4.5.1 超球交叉を用いた Differential Evolution とその調整則の設計

今まではアルゴリズム自体にはほとんど手を加えず、調整則の設計を行った。本節では、DEの交叉方法を変更し、DEに回転不変性を取り入れ、その新しい交叉方法のためのパラメータ調整則を設計する。回転不変性をもつ交叉方法としては文献[69]の超球を用いた方法や、文献[70]で挙げられているグラムシュミット直交化を用いた方法、文献[71]のように固有値分解を用いる方法があるが、本節では最初の超球交叉^[69]を用い、その超球の半径を適切に調整するためのパラメータ調整則を文献[35][36]と同様に設計する。

DEに回転不変性が備わっていないのは一様交叉、ないしは指数交叉によるものである。一様交叉は、 $\mathbf{M}_1 + \mathbf{M}_2 = \mathbf{1}$ を満たす0または1のみを成分とする二つの対角行列 $\mathbf{M}_1, \mathbf{M}_2$ を用いて、探索点を $\mathbf{x}^{(p)}(k)$ 、変異点を $\mathbf{v}^{(p)}(k)$ として

$$\mathbf{u}^{(p)}(k) = \mathbf{M}_1 \mathbf{x}^{(p)}(k) + \mathbf{M}_2 \mathbf{v}^{(p)}(k) = \mathbf{M}_1 (\mathbf{x}^{(p)}(k) - \mathbf{v}^{(p)}(k)) + \mathbf{v}^{(p)}(k) \quad (4.20)$$

と表すことができる。ここで、 $\mathbf{x}^{(p)}(k)$ は探索点、 $\mathbf{v}^{(p)}(k)$ は変異点、 $\mathbf{u}^{(p)}(k)$ は試験点である。座標系における \mathbf{a} の分の平行移動に対して、平行移動の後の交叉を行って生成される試験点 $\mathbf{u}'^{(p)}(k)$ は

$$\begin{aligned} \mathbf{u}'^{(p)}(k) &= \mathbf{M}_1 ((\mathbf{x}^{(p)}(k) - \mathbf{a}) - (\mathbf{v}^{(p)}(k) - \mathbf{a})) + (\mathbf{v}^{(p)}(k) - \mathbf{a}) \\ &= \mathbf{M}_1 (\mathbf{x}^{(p)}(k) - \mathbf{v}^{(p)}(k)) + \mathbf{v}^{(p)}(k) - \mathbf{a} = \mathbf{u}^{(p)}(k) - \mathbf{a} \end{aligned} \quad (4.21)$$

となり、交叉を行ってから平行移動する場合と同じ結果となり、平行移動に対する不変性が確認できる。

これに対して座標系の回転操作 \mathbf{R} に対しては、回転操作の後の交叉を行って生成される試験点 $\mathbf{u}''^{(p)}(k)$ は

$$\begin{aligned} \mathbf{u}''^{(p)}(k) &= \mathbf{M}_1 \mathbf{R} (\mathbf{x}^{(p)}(k) - \mathbf{v}^{(p)}(k)) + \mathbf{R} \mathbf{v}^{(p)}(k) \\ &\neq \mathbf{R} (\mathbf{M}_1 (\mathbf{x}^{(p)}(k) - \mathbf{v}^{(p)}(k)) + \mathbf{v}^{(p)}(k)) = \mathbf{R} \mathbf{u} \end{aligned} \quad (4.22)$$

であり、交叉してからの回転と一致せず、一様交叉は回転操作に対して不変的ではないことが確認できる。

この問題を解消したのが超球交叉であり、具体的にはDEの試験ベクトル $\mathbf{u}^{(p)}(k)$ を

$$\mathbf{u}^{(p)}(k) = \|(\mathbf{v}^{(p)}(k) - \mathbf{x}^{(p)}(k))/2\| \mathbf{S}() + (\mathbf{v}^{(p)}(k) + \mathbf{x}^{(p)}(k))/2 \quad (4.23)$$

により生成するものである。ここで、 $\mathbf{S}()$ は N 次元の超球上に一様に分布する乱数であり、 N 次元の乱数を生成し、ノルムを規格化することにより得ることができる^[72]。本交叉において、座標系の回転操作を施した後に超球交叉を行うと、

$$\begin{aligned} \mathbf{u}'''^{(p)}(k) &= \| \mathbf{R} (\mathbf{x}^{(p)}(k) - \mathbf{v}^{(p)}(k)) / 2 \| \mathbf{S}() + \mathbf{R} (\mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k)) / 2 \\ &= \| (\mathbf{x}^{(p)}(k) - \mathbf{v}^{(p)}(k)) / 2 \| \mathbf{S}() + \mathbf{R} (\mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k)) / 2 \end{aligned} \quad (4.24)$$

となるのに対して、超球交叉を行ってから座標系の回転操作を施すと、

$$\mathbf{u}''''^{(p)}(k) = \| (\mathbf{x}^{(p)}(k) - \mathbf{v}^{(p)}(k)) / 2 \| \mathbf{R} \mathbf{S}() + \mathbf{R} (\mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k)) / 2 \quad (4.25)$$

となるが, $RS()$ が単位超球面上の乱数であるため, $S()$ と同一と見なせ, 結果として回転操作に対して不変的であることが確認できる. 本シミュレーションにおいては超球の半径を調整するパラメータ調整則 $\sigma(\cdot)$ を導入し, 交叉の更新式を

$$\mathbf{u}^{(p)}(k) = \sigma(\cdot) \|(\mathbf{v}^{(p)}(k) - \mathbf{x}^{(p)}(k))/2\| S() + (\mathbf{v}^{(p)}(k) + \mathbf{x}^{(p)}(k))/2 \quad (4.26)$$

としたうえで 4.2 節の方法で最適な $\sigma(\cdot)$ を設計する.

4.5.2 Differential Evolution の調整則の進化結果

DE に対して進化的に獲得された調整則の具体的な数式を, GP により進化的に獲得された木構造を基に表現すると以下のとおりである. PSO と ES のシミュレーションと同様に, GP によって木構造の形で獲得されたパラメータ調整則の表現型であり, 5 回の GP の試行により, 各試行における最終世代の木構造の個体数の中で, 最良の目的関数値を与えるものである.

(a) 探索点変位および最良点との距離による調整則

$$\sigma(k+1) = -0.845073 \quad (4.27)$$

(b) イテレーション経過割合による調整則

$$\begin{aligned} \sigma(k+1) = & (\text{sig}(\sigma(k) - (((((0.106703 + \text{sig}(\text{sig}(0.422045)))) - (1.0 + (-1.01093 + \\ & \sigma(k)))))) \times 0.348373) / (((((0.652757 \times 0.0) + \exp((0.826845 + \\ & 0.652009))) - 7.74900) + (0.914288 \times (\text{sig}((t(k) + 0.0) + \\ & (\exp(0.715625) - 0.999969)) / 0.331957)))) - (\text{sig}(\sigma(k)) / 0.106703))) - \\ & (\text{sig}(\sigma(k) - (((((-0.343658 \times 0.0) + \text{sig}(\text{sig}(0.422045))) - (0.13012 / \\ & (0.13012 + ((\exp(t(k)) - \text{sig}(((0.826845 \times t(k)) \times \exp((\sigma(k) - t(k)))))) / \\ & \text{sig}(\sigma(k)))))) \times 0.348373) / (-1.71054 + (t(k) \times ((\text{sig}(((0.0 - \\ & (\text{sig}(((0.800094 \times 0.0) \times ((\text{sig}(0.0) + 0.991582) - ((-0.452505 \times \\ & (0.0 \times t(k))) + 0.830866))) \times \sigma(k) + (-0.368081 / t(k)))) / 0.331957)) - \\ & (-0.862655 - (\text{sig}(-0.797383) \times 0.106703))) \times \sigma(k) + 0.601206))) - \\ & (t(k) / 0.106703))) \times t(k)) \end{aligned} \quad (4.28)$$

(c) 目的関数改善割合による調整則

$$\begin{aligned}
 \sigma(k+1) = & (\text{sig}(-(\text{sig}(\text{sig}(((p_s(k) + \text{sig}(\text{sig}(0.135453)/\text{sig}(-(\text{sig}(0.832534)))))) \times \\
 & - 0.895216) + (\text{sig}(0.0/\text{sig}(\text{sig}(\text{sig}(-0.643055 + (-0.822369 + \\
 & (p_s(k)/(\sigma(k)/\sigma(k)))))) + 0.273953)/\text{sig}(-0.767463/(\text{sig}(-0.579801/ \\
 & ((\sigma(k)/0.72635)/\text{sig}(\text{exp}(\sigma(k)))))/\text{exp}(-0.804418))))/ \\
 & (\text{exp}(-0.597217) + \text{exp}(\sigma(k))))/(\text{exp}(0.571919)/0.894953) + \\
 & \text{sig}(\text{sig}(\text{exp}(0.937536)/(\text{exp}(-(\text{sig}(0.217976) + \text{sig}(\text{exp}(1.0)))) \times \\
 & - 0.895216) - ((-0.889697 + 0.246642) \times (\text{sig}(0.919793)/\sigma(k))/ \\
 & 0.120783)/\text{sig}(p_s(k))))/ - 0.879327)))/0.894953 + 0.75961)) - \\
 & (\text{sig}(0.0)))))))/p_s(k)/0.135453)))) + \text{exp}(\sigma(k)/ \\
 & 0.72635) - 38.3887) \times (p_s(k)/(0.103681 \times \text{sig}(p_s(k)))))) + \\
 & (p_s(k)/0.135453)
 \end{aligned}
 \tag{4.29}$$

これらの木構造を獲得するまでの GP の世代更新に伴う目的関数値の進化過程を 100 世代まで記した様子を Fig. 4.9～Fig. 4.11 に示す. PSO や ES と同様に多目的最適化問題 (3.45) を想定し, 同じ条件でシミュレーションしたものである. 最終的に定数となった (a) の調整則の GP 実行過程の Fig. 4.9 と比べて, Fig. 4.11 のグラフは Rosenbrock と Alpine 関数においてより良い結果となっていることが確認できる.

(a)～(c) を比較検定した結果が Table 4.3 であり, 結果から (c) の設定において進化させた調整則がもっとも良いことがわかる. 特に, 他の調整則より $f_6 \sim f_{14}$ の悪条件の問題を効率よく解くことができる傾向がみられる. また, (a) の位置情報を用いた調整則は定数となっており, 位置情報のフィードバックにより調整則を設計することは本アルゴリズムにおいて有効でないと考えられる. 同様に, 定数となった (a) に対して (b) も良い結果となっていないことから, 同様に時間と共に決定的にパラメータを変化させる調整則も, 本アルゴリズムにおいて有効でない結果となった.

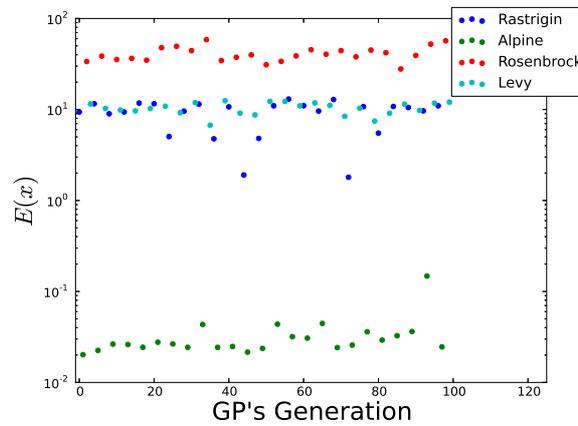


Fig. 4.9: $E(x)$ of Genetic Programming during evolving rule for DE on setting (a).

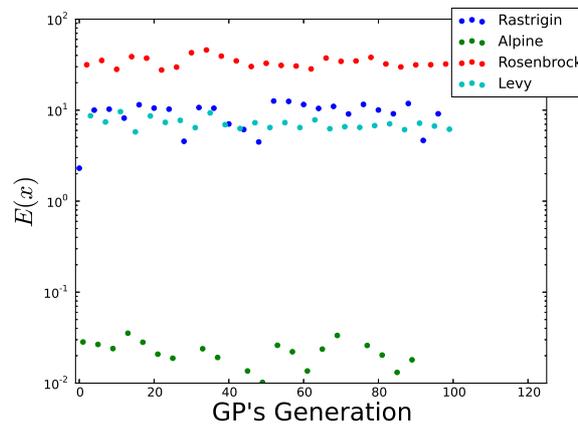


Fig. 4.10: $E(x)$ of Genetic Programming during evolving rule for DE on setting (b).

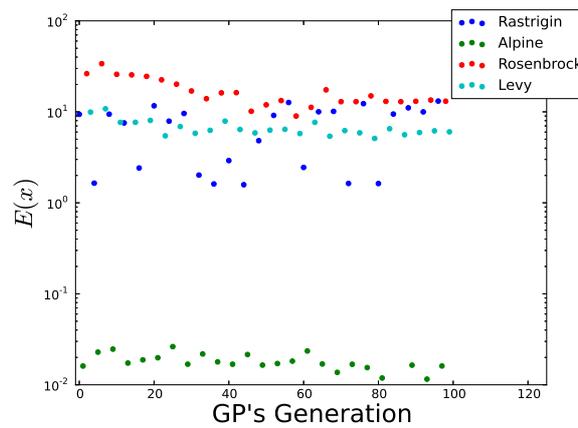


Fig. 4.11: $E(x)$ of Genetic Programming during evolving rule for DE on setting (c).

(c)の調整則がもっとも優秀であることから、この調整則の近似を行う。(c)の数式も一見複雑であるが、(c)の調整則を Rastrigin 関数に対して、1,000 イテレーション動作させ、 $p_s(k)$ と $\sigma(\cdot)$ の散布図をプロットすると Fig 4.12 のようになる。これを直線近似したものは

$$\sigma(\cdot) = 7.046p_s(k) + 0.4150 \quad (4.30)$$

となる。

(4.30) 式の意味を考えてみると、 $p_s(k)$ が大きい、つまり更新が上手くいっている場合は σ を増加させ、ステップ幅を大きくし、 $p_s(k)$ が小さい、つまり更新が停滞している場合は σ を減少させるというルールになっており、これは 1/5 ルールや CMA-ES の考え方に則したパラメータ調整則になっていることがわかる。

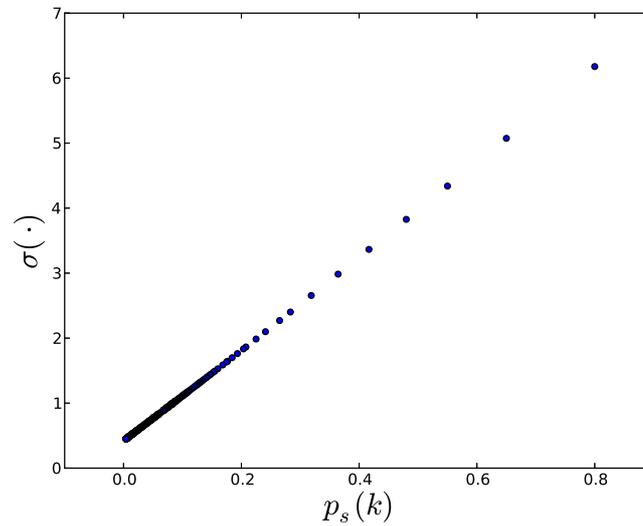


Fig. 4.12: Scatter plot of (c)

Table 4.3: Pairwise ranked Welch's t-test for 30 dimensional functions with 20,000 function calls of evolved DE

alg1	alg2	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19	f20	f21	f22	f23	f24	f25	L	R
(a)	(b)	L	R	R	R	L	L	L	N	N	R	R	L	L	L	R	N	L	L	R	N	N	L	N	R	L	11	8
(a)	(c)	L	R	R	R	N	R	R	R	R	R	R	L	R	L	R	N	N	N	N	N	R	L	N	N	N	4	12
(b)	(c)	L	R	N	N	R	R	R	R	R	R	N	L	R	R	N	N	R	R	N	L	R	L	N	N	R	4	13

4.6 Firefly Algorithm のパラメータ調整法のメタ最適化

4.6.1 同期型 Firefly Algorithm とその調整則の設計

最後に Firefly Algorithm (FA) のパラメータ調整則の設計を行う。FA は構造的には多点化した ES に best 座標へ向かう項を付与したものとなっている。また、従来の FA では、更新対象の探索点が更新するか否かに関わらず、比較対象の更新点との目的関数値の比較のたびにイテレーションが更新され、複数の探索点の更新に関して非同期的な更新が想定されている。しかし、PSO や DE 等のほかのメタヒューリスティクスと同様に多数の点を同時に更新することも考えられる。ここでは、探索点 $\mathbf{x}^{(p)}(k)$ を、下記の方法で、多数の点を同時に更新する同期型 FA^[37] を考える。

$$\begin{aligned} \mathbf{v}^{(p)}(k+1) &= \beta(\cdot)(\mathbf{x}^{(q\text{-best})}(k) - \mathbf{x}^{(p)}(k)) + \alpha(\cdot)\mathbf{N}(\mathbf{0}, \mathbf{I}), \\ &\text{where } E(\mathbf{x}^{(q\text{-best})}(k)) \leq E(\mathbf{x}^{(p\text{-best})}(k)) \end{aligned} \quad (4.31a)$$

$$\mathbf{x}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k+1) \quad (4.31b)$$

$$\begin{aligned} \mathbf{x}^{(p\text{-best})}(k) &= \operatorname{argmin}\{E(\mathbf{x}^{(p)}(i)) \mid i = 1, \dots, k\} \\ \mathbf{x}^{(q\text{-best})}(k) &= \operatorname{argmin}\{E(\mathbf{x}^{(q)}(i)) \mid i = 1, \dots, k\} \end{aligned} \quad (4.31c)$$

従来の FA と異なる点は、多数の点を同時に更新(同期)する点であり、これによりアルゴリズムの並列実行が可能となる。また、従来の FA が現時点での情報のみを用いるのに対し、本アルゴリズムは現イテレーション k における更新対象の探索点の位置を $\mathbf{x}^{(p)}(k)$ 、比較対象の探索点の位置を $\mathbf{x}^{(q)}(k)$ とすると、現イテレーション k から過去に遡った両探索点それぞれの最良点 $\mathbf{x}^{(p\text{-best})}(k)$ 、 $\mathbf{x}^{(q\text{-best})}(k)$ に対してそれらの目的関数値を比較し、 $E(\mathbf{x}^{(q\text{-best})}(k)) \leq E(\mathbf{x}^{(p\text{-best})}(k))$ を満たす p を含む探索点 q をランダムに選び、現イテレーションでの更新対象探索点の位置 $\mathbf{x}^{(p)}(k)$ を比較対象の探索履歴の最良点 $\mathbf{x}^{(q\text{-best})}(k)$ の方向に移流する手法となっている。設計対象とするパラメータは α と β とする。また、乱数項は正規分布乱数 $\mathbf{N}(\mathbf{0}, \mathbf{I})$ とし、回転不変性を保証する。このアルゴリズムの更新手順は以下となる。

Step 1 最大イテレーション K を設定し、 P 個(複数個)の探索点の初期座標 $\mathbf{x}^{(p)}(0)$, $p = 1, \dots, P$ をランダムに与え、 $k = 0$ とおく。

Step 2 $E(\mathbf{x}^{(q\text{-best})}(k)) < E(\mathbf{x}^{(p\text{-best})}(k))$ となる q を p も含めてランダムに決定し、更新対象探索点 $\mathbf{x}^{(p)}(k)$ の変位 $\mathbf{v}^{(p)}(k+1)$ を (4.31a) 式により生成し、更新対象の探索点の新しい位置 $\mathbf{x}^{(p)}(k+1)$ を求める。この操作を $p = 1, \dots, P$ に対して繰り返す。

Step 3 たとえば (2.4) 式や (2.5) 式を用いて探索点の座標に制限を課す。

Step 4 各探索点の目的関数値 $E(\mathbf{x}^{(p)}(k+1))$, $p = 1, \dots, P$ を計算し、各探索点の最良点を更新する。

Step 5 $k \geq K$ ならば計算を終了し、そうでなければ、 $k = k + 1$ として Step 2 へ戻る。

FA は ES と同様に，最初からパラメータ調整則が与えられており， β に関しては (a) による調整， α に関しては (b) による調整がアルゴリズムの作者より提案されている [41]．本シミュレーションは，文献 [37] と同じく (a) や (b) だけでなく，(c) の場合もシミュレーションを行い，同期型 FA に最適な調整則を探索する．

4.6.2 Firefly Algorithm の調整則の進化結果

他のメタヒューリスティクスに対するシミュレーションと同様に，進化的に獲得された調整則の具体的な数式を，GP により進化的に獲得された木構造を基に表現すると以下のとおりである．

(a) 探索点変位および最良点との距離による調整則

$$\begin{aligned} \alpha(k+1) = & (\alpha(k) / (+((((\exp(0.0403077) / (0.808896 + (\text{sig}(-0.097204) \times \\ & - (((0.923147 + ((((((1.0 / (0.808896 + \Delta_g(k))) - (((+((((1.0 / \\ & (-0.146655 + \Delta_g(k))) - 0.0) - 1.0)) \times ((\Delta_g(k) - 0.704573) \times \\ & 0.00421573)) / (\alpha(k) / 0.0566642)) \times (||\mathbf{v}(k)|| - \\ & ((0.534754 / (0.6219478 + \Delta_g(k))) + 0.274142)) \times \exp(||\mathbf{v}(k)||)))) - 1.0) \times \\ & (||\mathbf{v}(k)|| / ((0.808896 + \Delta_g(k)) - 0.677993)) / (0.538969 + \Delta_g(k))) + \\ & 0.943535)) - 0.677993)))))) \times 0.00421573) - 0.943535) - 0.0871938)) \\ \beta(k+1) = & 1.0 \end{aligned} \tag{4.32}$$

(b) イテレーション経過割合による調整則

$$\begin{aligned} \alpha(k+1) = & (((((0.456116 \times (-0.675137t(k) - (-0.675137 \times ((0.405215 \times (t(k) / \\ & \text{sig}(t(k)) - \sqrt{\text{sig}(t(k))})) \times (t(k)))))) + 0.226658) \times (((0.456116 \times \\ & ((t(k) \times -0.675137) - ((0.067446 \times ((0.456116 \times (-0.144339 - \alpha(k))) \times \\ & (-0.144339)) - \alpha(k)))) + 0.262688) / 0.737312)) \\ \beta(k+1) = & 1.0 \end{aligned} \tag{4.33}$$

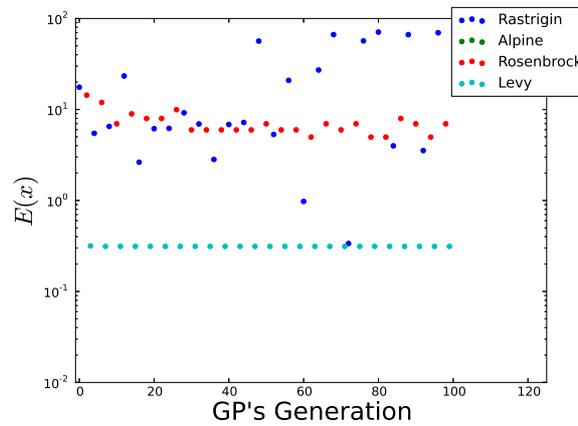


Fig. 4.13: $E(x)$ of Genetic Programming during evolving rule for FA on setting (a).

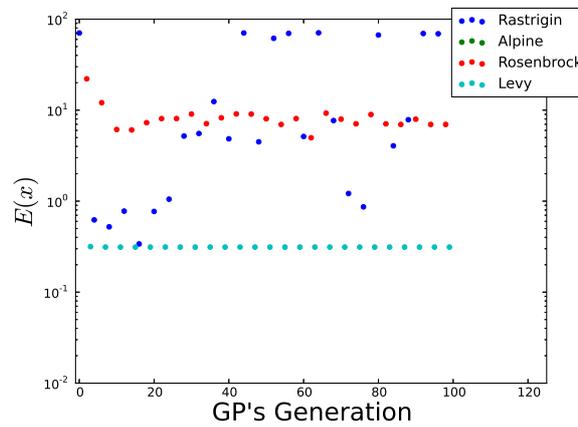


Fig. 4.14: $E(x)$ of Genetic Programming during evolving rule for FA on setting (b).

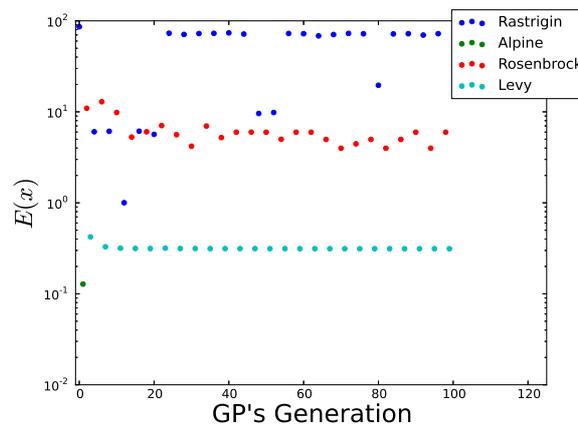


Fig. 4.15: $E(x)$ of Genetic Programming during evolving rule for FA on setting (c).

これら (a)~(c) の調整則を比較検定により評価したところ、Table 4.4 となり、(b) が一番良いことが確認される。(b) が良いという点において ES のパラメータ調整則の進化結果と同様であり、アルゴリズムの更新式が ES に似ていることによるものと考えられる。

(b) の設定において進化させた調整則がもっとも良いことから、この数式の近似を試みる。(b) の数式は ES のと同様に、イテレーション k とともに変化し、問題によらず一意の曲線を描く。そこで近似を行うが、 σ は正規分布の分散を調整するパラメータであり、大域的な探索、そして局所的な探索を同様に行うべきであるというフラクタル性に注目し、ES の調整則の近似と同様に、この調整則を対数スケールで直線近似する。これを示した図は Fig. 4.16 であり、青線が調整則のパラメータであり、緑線が近似したものである。近似した数式は

$$\alpha(k+1) = 0.25\exp(-7.72t(k)) \quad (4.35)$$

となり、ES と同様に「継続的な集中化」による調整則であることが確認できる。

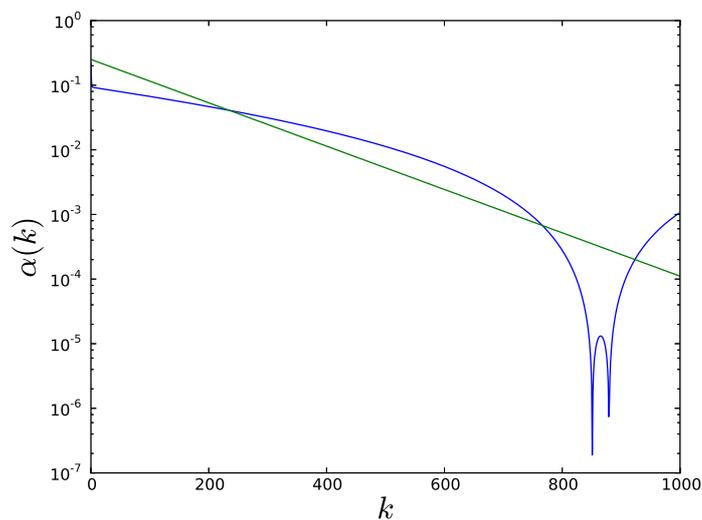


Fig. 4.16: Scatter plot of (c) and its approximation

Table 4.4: Pairwise ranked Welch's t-test for 30 dimensional functions with 20,000 function calls of evolved FA

alg1	alg2	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19	f20	f21	f22	f23	f24	f25	L	R
(a)	(b)	N	R	R	R	R	R	R	N	N	N	L	L	N	L	R	N	R	R	R	N	N	N	N	R	N	2	11
(a)	(c)	L	N	N	N	N	R	N	N	L	R	L	L	L	L	R	L	L	N	R	N	N	L	R	R	N	9	6
(b)	(c)	L	N	L	L	L	R	L	N	L	N	L	L	L	L	N	L	L	L	R	N	N	L	R	R	L	15	4

4.7 新しい調整則を有するメタヒューリスティクスの総合比較

第4.3節から第4.6節までにおいて、メタヒューリスティクスとして PSO, ES, DE, FA を取り上げ、それらの手法を高性能化するような調整則を GP を用いて進化的に獲得した結果と、その結果を基に得られた調整則の関数近似により導出した新しい調整則を有する高性能な手法を提案した。後者の近似的に導出した調整則とそれを有する更新則を、改めてまとめると以下のとおりである。ただし、これらの新しい更新則の引数は、第3.5節の繰り返しになるが、以下のとおりである。

$\|\mathbf{v}(k)\|$
 $\frac{1}{P\sqrt{N}} \sum_{p=1}^P \|\mathbf{v}^{(p)}(k)\|$ により計算される第 k イテレーションでの探索点の変位の大きさの探索点群全体での平均であり、アルゴリズムの探索点群の活発さを示す指標である。

$\Delta_g(k)$
 $\frac{1}{P\sqrt{N}} \sum_{p=1}^P \|\mathbf{x}^{(g\text{-best})}(k) - \mathbf{x}^{(p)}(k)\|$ により計算される探索点と $g\text{best}$ 座標の距離の探索点群全体での平均であり、アルゴリズムの探索点群の密集状態を示す指標である。

$t(k)$
 k/K により計算される最大イテレーション回数 K に対する第 k イテレーション時の経過割合、アルゴリズムの経過時間の指標である。

$p_s(k)$
 $\frac{1}{P} \sum_{p=1}^P p_s^{(p)}(k)$ により計算される第 k イテレーションから遡った一定イテレーション間で、目的関数値を改善させることに成功したイテレーション数の割合の探索点群での平均であり、アルゴリズムの直近の探索の成功度合いを示す指標である。

近似された調整則と、アルゴリズムの更新則は、以下のとおりとなる

1. 新調整則付き PSO(表などでは New-PSO と記す)

• 更新則

$$\begin{aligned} v_n^{(p)}(k+1) = & c_0 v_n^{(p)}(k) + c_1 r_{1n}^{(p)}(k) (x_n^{(p\text{-best})}(k) - x_n^{(p)}(k)) \\ & + c_2(k) r_{2n}^{(p)}(k) (x_n^{(g\text{-best})}(k) - x_n^{(p)}(k)), \quad n = 1, \dots, N \end{aligned} \quad (4.36a)$$

$$\mathbf{x}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k+1) \quad (4.36b)$$

$$\begin{aligned} \mathbf{x}^{(p\text{-best})}(k) = & \operatorname{argmin}_{\mathbf{x}^{(p)}(i)} \{E(\mathbf{x}^{(p)}(i)) | i = 0, 1, \dots, k\} \\ \mathbf{x}^{(g\text{-best})}(k) = & \operatorname{argmin}_{\mathbf{x}^{(q\text{-best})}(k)} \{E(\mathbf{x}^{(q\text{-best})}(k)) | q = 1, \dots, P\} \end{aligned} \quad (4.36c)$$

• 調整則

$$c_0 = 0.769 \quad (4.37a)$$

$$c_1 = 1.752 \quad (4.37b)$$

$$c_2(k) = \exp(0.6484 - 1.025\|\mathbf{v}(k)\| - 1.182\Delta_g(k) + 0.02597/\Delta_g(k)) \quad (4.37c)$$

2. 新調整則付き ES(表などでは New-ES と記す)

- 更新則

$$\mathbf{u}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \sigma(k)\mathbf{N}^{(p)}(\mathbf{0}, \mathbf{I}) \quad (4.38a)$$

$$\mathbf{x}^{(p)}(k+1) = \begin{cases} \mathbf{u}^{(p)}(k+1), & \text{if } E(\mathbf{u}^{(p)}(k+1)) \leq E(\mathbf{x}^{(p)}(k)) \\ \mathbf{x}^{(p)}(k), & \text{otherwise} \end{cases} \quad (4.38b)$$

- 調整則

$$\sigma(k+1) = 0.1102\exp(-5.629t(k)) \quad (4.39)$$

3. 新調整則付き DE(表などでは New-DE と記す)

- 更新則

$$\mathbf{v}^{(p)}(k) = \mathbf{x}^{(q_1^{(p)})}(k) + c(\mathbf{x}^{(q_2^{(p)})}(k) - \mathbf{x}^{(q_3^{(p)})}(k)) \quad (4.40a)$$

$$\mathbf{u}^{(p)}(k) = \sigma(k) \left(\|\mathbf{v}^{(p)}(k) - \mathbf{x}^{(p)}(k)\|/2 \|\mathbf{S}()\| + (\mathbf{v}^{(p)}(k) + \mathbf{x}^{(p)}(k))/2 \right) \quad (4.40b)$$

- 調整則

$$\sigma(k+1) = 7.046p_s(k) + 0.4150 \quad (4.41)$$

4. 新調整則付き FA(表などでは New-FA と記す)

- 更新則

$$\mathbf{v}^{(p)}(k+1) = (\mathbf{x}^{(q\text{-best})}(k) - \mathbf{x}^{(p)}(k)) + \alpha(k)\mathbf{N}(\mathbf{0}, \mathbf{I}), \quad (4.42a)$$

where $E(\mathbf{x}^{(q\text{-best})}(k)) \leq E(\mathbf{x}^{(p\text{-best})}(k))$

$$\mathbf{x}^{(p)}(k+1) = \mathbf{x}^{(p)}(k) + \mathbf{v}^{(p)}(k+1) \quad (4.42b)$$

$$\begin{aligned} \mathbf{x}^{(p\text{-best})}(k) &= \operatorname{argmin}\{E(\mathbf{x}^{(p)}(i)) \mid i = 1, \dots, k\} \\ \mathbf{x}^{(q\text{-best})}(k) &= \operatorname{argmin}\{E(\mathbf{x}^{(q)}(i)) \mid i = 1, \dots, k\} \end{aligned} \quad (4.42c)$$

- 調整則

$$\alpha(k+1) = 0.25\exp(-7.72t(k)) \quad (4.43)$$

これらの新しい調整則を有する新 PSO, 新 ES, 新 DE, 新 FA の間の計算性能だけでなく, 比較的高性能を有すると判断される NDT-PSO, CMA-ES, jDE, FA などの従来法とも比較対象とする. なお, シミュレーション条件は, CMA-ES を除いて 2 章・4 章の設定と同じとし, CMA-ES では, 探索点数を 2 章の設定の目的関数の変数次元 $P = N \times 10 = 300$ とするとファンクションコール数 20,000 の約数にならないため, $P = 250$ と設定し, $K = 80$ とする. 有意水準は 5% とし, 全部で 8 種類のアロリズムの対比較検定を 25 関数に対し

で行うことから、Bonferroni法を用いて P 値が $0.05/(8C_2 \times 25)$ より小さければ統計的に有意であるとする。

以上のように、単純化した近似調整則を有する新しいメタヒューリスティクスと、旧来のアルゴリズムでもっとも優れたアルゴリズムとみなされる手法の合計8種類のシミュレーション結果に対して、評価を行った結果が Table 4.5 である。表の第1・2列の手法の対比較に対する、表の最終の2列のLとRの個数により、不等号を用いて性能の優劣を表すと、もとなつたアルゴリズムの中で優秀なものより、GPで進化させ、近似した新手法の方が圧倒的に優れていることが、表の6, 10, 16, 22行目より読み取れる。また、表の1~3, 8, 9, 14行より、新手法間においては

$$\text{New-DE} > \text{New-FA} > \text{New-PSO} > \text{New-ES}$$

という優劣が、旧手法間では表の下から6行目までより

$$\text{FA} > \text{jDE} > \text{NDT-PSO} > \text{CMA-ES}$$

という性能差が確認され、さらに双方のグループの最上位であるNew-DEとFAの比較では、New-DEの方が一般的に良い結果を出すことが確認され、著者による文献[35][36]および本論文で提案したNew-DEがもっとも一般的に良い結果を出す優れた手法として推奨されることが判明した。

なお、各手法ごとに特筆すべきことをあげると以下のとおりである。

- (1) PSOにおいて(a)の引数を用いた調整ルールが最もよかつた理由
 - (b)と(c)による調整則の方が問題に対する依存性がなく、よりロバストであると3.5節において考えたが、今回のPSOのシミュレーションでは(a)の調整則の方が優れている結果となつた。Table 4.1を確認すると、(a)の調整則が特に優れているのは変数間の依存性がない $f_1 \sim f_5$ の問題に対してであり、探索の持続性とフラクタル性を見る f_{25} のフラクタル関数や、(a)が優れている f_2 を回転させた f_{10} では、(b)と(c)の方が優れている結果となっている。このことから、変数間依存性のない問題に対してのみ(a)の方が優れているといえ、一般的な問題に対しては3.5節で考察したように(b)や(c)の方が優れているといえる。
- (2) 新手法の中でESが最も劣つていた理由

ESの手法は、多点型ではあるものの、探索点間の相互作用や情報共有が、パラメータ調整則に与えられる(a),(c)の引数に限定されるため、その影響が非常に弱かつたと考えられる。
- (3) 新手法の中でDEが最も優れていて理由

DEはメタヒューリスティクスの中でも、特に優秀なアルゴリズムであり、関連する研究もPSO等と比べて多い。そのため、比較的古いPSOやESと比べて、今回は回転不変性を保証する交叉を取り入れたこともあり、本論文でも最終的にもっとも良い結果が得られたものと結論付けることができる。
- (4) FAが必ずしも優れているとは限らない理由

FAは優れた手法であるとして提案されているが、そのパラメータ推奨値が不明であるなど、それ自体は必ずしも有力な手法ではない。本シミュレーションにおい

ては調整則を進化させた結果、この欠点を克服することによって New-FA において New-DE とほぼ同等の性能を達成した。

以上の結果、従来の経験と試行錯誤によるアルゴリズムの改良ではなく、計算的に、機械的に目的をもってアルゴリズムを改良することにより、より良いアルゴリズムを設計できることが確認できた。人間がアルゴリズムを改良する際は、特定の問題、ないしは構造に対して良くなると信じる操作を導入するが、それが与える副作用により他の問題の性能が悪化することが多々ある。そのような人間のアルゴリズム設計における先入観、認知バイアス、探索のばらつきを排除することにより、本手法はより効率的に、アルゴリズムを改良することを確認しつつ、新たな手法を創出した。これらの手法は代表的な4種類のベンチマークを用いて得られたものにも関わらず、その他の様々なより高度な25種類のベンチマークに対しても汎用性と性能の高さを確保しており、遺伝的プログラミングを用いたパラメータ調整則の設計は、アルゴリズム設計において革新的な手法であると考えられる。

Table 4.5: Pairwise ranked Welch's t-test for 30 dimensional functions with 20,000 function calls of algorithms

alg1	alg2	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19	f20	f21	f22	f23	f24	f25	L	R	
New-DE	New-ES	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	R	N	R	R	L	L	L	19	4
New-DE	New-FA	R	N	R	N	L	N	L	L	L	L	L	R	N	L	R	R	L	L	L	R	N	R	R	L	L	N	11	7
New-DE	New-PSO	R	R	N	N	L	L	L	L	L	L	L	R	L	L	L	R	L	L	R	R	N	N	N	L	L	L	14	6
New-DE	CMA-ES	L	L	L	N	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	24	0
New-DE	FA	L	L	R	L	L	L	L	L	L	L	L	L	L	L	R	R	L	L	L	R	N	N	R	N	L	L	17	5
New-DE	jDE	R	R	R	N	R	L	L	L	L	L	L	R	L	L	L	L	L	L	R	R	N	N	L	L	L	L	16	6
New-DE	NDT-PSO	L	L	R	N	L	L	L	L	L	L	L	L	L	L	L	N	L	L	L	R	L	L	N	L	L	L	20	2
New-ES	New-FA	R	R	R	R	R	R	R	N	L	L	L	R	R	R	R	L	R	R	R	N	L	N	L	R	R	R	5	16
New-ES	New-PSO	R	R	R	R	R	R	R	N	L	L	L	R	R	R	R	L	R	R	R	R	L	L	L	L	N	R	5	16
New-ES	CMA-ES	L	N	L	L	N	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	N	20	0
New-ES	FA	R	N	R	R	N	R	R	N	L	L	L	R	R	R	R	R	R	R	R	L	L	N	R	R	R	R	4	16
New-ES	jDE	R	R	R	R	R	R	R	L	L	L	L	R	R	R	L	L	R	R	R	L	L	N	L	L	L	R	10	14
New-ES	NDT-PSO	L	N	R	R	L	L	L	L	L	L	L	R	L	L	N	L	L	R	R	R	L	L	L	R	R	R	13	9
New-FA	New-PSO	R	L	N	R	L	L	L	N	N	R	L	N	L	L	L	L	L	L	N	N	N	N	L	L	L	N	10	6
New-FA	CMA-ES	L	L	L	L	N	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	22	0
New-FA	FA	L	L	L	L	L	L	L	N	N	L	L	R	L	L	R	R	L	L	L	L	L	N	N	R	R	L	12	4
New-FA	jDE	R	R	L	L	R	L	L	L	N	L	L	R	L	L	L	L	L	L	L	L	N	N	L	L	L	N	16	4
New-FA	NDT-PSO	L	L	L	N	L	L	L	L	R	L	L	L	L	L	L	L	L	L	N	N	L	L	L	N	L	L	19	2
New-PSO	CMA-ES	L	L	L	L	N	L	L	L	L	N	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	22	0
New-PSO	FA	L	L	N	L	L	R	R	N	L	L	L	L	R	N	R	R	R	R	N	L	N	N	R	R	R	N	8	9
New-PSO	jDE	R	L	N	R	N	N	L	N	L	L	L	R	N	L	L	L	R	R	L	L	N	N	L	L	L	N	11	6
New-PSO	NDT-PSO	L	L	N	N	L	L	L	N	N	L	L	L	L	L	L	L	L	R	N	R	L	L	N	R	L	L	14	4
CMA-ES	FA	R	N	R	R	N	R	N	R	R	N	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	0	21
CMA-ES	jDE	R	R	R	R	N	R	N	R	R	N	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	0	22
CMA-ES	NDT-PSO	R	N	R	R	N	N	R	R	R	N	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	0	20
FA	jDE	R	R	N	R	R	L	L	L	N	L	L	R	N	L	L	L	L	L	L	L	N	R	N	L	L	R	12	7
FA	NDT-PSO	L	N	N	R	L	L	L	L	R	L	L	L	L	L	L	L	L	L	L	R	L	L	L	L	L	N	17	4
jDE	NDT-PSO	L	L	R	N	L	L	L	N	R	L	L	L	L	L	R	R	L	L	L	R	L	L	L	R	R	L	14	9

第5章 おわりに

大域的最適化手法として近年提案された多くのメタヒューリスティクスとその改良手法は、その名称のとおり経験と試行錯誤に基づいて開発された多点型確率的探索手法であり、数理的な理論に基づいて提案された手法ではない。このため、新しい更新則をもったメタヒューリスティクスが提案されると、解析的な手段を用いるのではなく、ベンチマークを用いた計算機シミュレーションによって、従来のメタヒューリスティクスとの性能比較が行われ、さらにその更新則の改良に関する提案や、更新則中に含まれているパラメータの調整法の提案が研究課題として取り上げられ、数多くの論文が執筆されるというのが、これらの分野の研究のスタイルとなっている。したがって、特定の名称が付されたメタヒューリスティックが登場とすると、それらの改良手法の間での性能比較がなされるのに対し、まったく構造や由来が互いに異なる手法の間で公平な観点から性能比較がされにくいという傾向がある。事実、Firefly Algorithm^[41]、Cuckoo Search^[60]、Bat Algorithm^[60]など、生物の名前を付したいくつものアルゴリズムを提案している Yang, X. S. 自身によっても、それらの異なる種類の手法の性能比較が十分になされているとは言い難い。

本論文では、特定の名称を関したメタヒューリスティクスに対し、計算機シミュレーションによる経験と試行錯誤によって改良する過程を自動化することを目的とした。そのために、メタヒューリスティクスがもつパラメータの調整則の中で、解きたい最適化問題に適した最良の調整則を求める問題をメタ最適化問題として定式化し、この問題を解いて改良した調整則を自動的に求める手段として遺伝的プログラミングを適用することを提案した。すなわち、特定のメタヒューリスティクスを改良しようとする研究者の問題意識を、そのアルゴリズムの最適パラメータ調整則を求めるメタ最適化問題として定式化し、特定のメタヒューリスティクスを改良しようとする研究者の試行錯誤の過程を遺伝的プログラミングに担わせたといえる。このような概念と方法論の提唱により、メタヒューリスティクスの改良を効率的かつ科学的・機械的にした試みは本論文の著者以外にない。

また、本論文では、その由来や発想がまったく異なる異種のメタヒューリスティクス間でそれらの性能比較を横断的かつ公平に行った。具体的には、メタヒューリスティクスとして PSO, ES, DE, FA を取り上げたが、とくに PSO, ES, DE はその由来や発想、そしてそのアルゴリズムの構造もまったく異にしている。FA は PSO に近い構造を有しているが、PSO より優れているとされて注目され始めた手法のためにあえて取り上げた。

各章の内容を振り返ると以下のとおりである。

第1章は、メタヒューリスティクスの歴史とともに、それらの手法を最適化問題に適用するときの課題について示し、同時に本論文の目的・意義・構成について説明した。

第2章では、本論文で扱う PSO, ES, DE, FA の解説とともに、それらの代表的な改良手法、およびパラメータ調整則を示した。これらは、手法ごとに計算性能の改良が試みられ、概してより複雑な構造や新しいパラメータ調整則が提案されており、そのこと自体に

存在する課題を提起した。

第3章は、メタヒューリスティクスに対して、解きたい最適化問題に適した最良の調整則を求める問題をメタ最適化問題として定式化し、この問題を遺伝的プログラミング (GP) の進化機能を用いて解くことによって調整則の改良を自動的に行う、すなわち「パラメータ調整則」を自動的に設計する概念を説明した。メタヒューリスティクスの改良を効率的かつ科学的・機械的に行う方法論を提唱した本論文の核心部分に当たる章である。

第4章において、第3章で提唱した概念と方法論を、PSO, ES, DE, FA に適用し、それらの最良なパラメータ調整則を GP によって進化的に獲得した結果を示し、これまで試行錯誤的に提案されてきた主たる改良手法より優れていることを統計的検定により確認した。さらに、GP によって獲得した調整則をより簡単な関数で近似することで、PSO, ES, DE, FA ごとに新しい簡単な調整則を提案し、それらに対して横断的かつ公平に統計的検定を施すことで選りすぐり、最良手法を結論づけた。

本論文では、メタヒューリスティクスのパラメータの最良調整則を発見するメタ最適化問題の概念の提唱とその遺伝的プログラミングによる解法を示した。アルゴリズムの開発、設計において本手法は有効であることを示せたが、一方でオフラインで調整則を設計し、問題に対して適用する必要があるという欠点がある。これを解決する枠組みとして、ニューラルネットワークを用いた Q-learning による強化学習の手法の応用^[73]が考えられる。最適化を行うと同時に、ニューラルネットワークを用いた Q-learning でアルゴリズムの学習を行うことができれば、任意の問題に対してオンラインで調整則を構築し、未知の問題に対して常に学習を行うアルゴリズムとなり得る。自身を最適化する最適化アルゴリズムの実現とその枠組みが、今後に残された課題といえよう。あるいはその枠組みも、本枠組みで実現できるかもしれない。

謝辞

本研究を行うにあたって、指導教員である相吉英太郎 教授には多大な御指導を頂きました。学部の授業も含めると 2007 年来の比較的長い期間であり、親身に対応していただき、改めて御礼を申し上げたいと思います。

また、本論文の審査に際して副査の方々からは中立的な立場から、固定観念により見落としがちだった様々な御指摘、御指南を頂きました。慶應義塾大学理工学部 物理情報工学科 田中敏幸 教授、内山孝憲 教授、情報工学科 萩原将文 教授、首都大学東京 安田恵一郎 教授には改めて御礼申し上げます。数多くの貴重な意見により、より論文の質を高めることができたと考えております。

本論文に至る私の人生における転換点は恐らく三つありました。一つ目は 1997 年にカスパロフがディーブ・ブルーにチェスで負けたことです。人間がコンピューターに負けるのであればコンピューターを用いてものを考えればよいという考えを持つようになり、コンピューターに興味を持ち始めました。二つ目は日経サイエンスの 2003 年 5 月号に載っていた遺伝的プログラミングの記事を本屋で偶然発見したことで、遺伝的プログラミングを用いればシミュレーターと組み合わせて人間に匹敵する発明をコンピューターに行わせることができると知りました。そこで現実の物理現象をコンピューター上で再現するための能力が必要だと考え、慶應義塾大学の物理情報工学科に進学しました。最後の転換点は相吉研究室に配属されたことであり、私の物事の捉え方のパラダイムを変革させるものでした。「全ての道はローマに通ず」という諺がありますが、今では「全ての問題は最適化に通ず」と確信しており、全ての問題は最適化問題として記述できると考えるまでになり、本論文は最適な最適化アルゴリズムを最適化で考えて導出するものとなりました。このパラダイムの習得は、研究生活で出会った先生、先輩、友人、後輩との相互作用の結果であり、皆様に感謝の意を示したいと思います。

参考文献

- [1] Glover, F.: “Future paths for integer programming and links to artificial intelligence.” *Computers & Operations Research*, Vol. 13, Issue 5, pp.533-549 (1986)
- [2] Glover, F., and Kochenberger, G. A. (Eds.): “Handbook of metaheuristics.” Springer (2003)
- [3] Gendreau, M., Potvin, J. Y.: “Handbook of metaheuristics.” Springer (2010)
- [4] Luke, S.: “Essentials of metaheuristics.” Lulu Enterprises (2013)
- [5] Robbins, H., and Sutton, M.: “A stochastic approximation method.” *The annals of mathematical statistics*, Vol. 22, No. 3, pp.400-407 (1951)
- [6] Rastrigin, L. A.: “The convergence of the random search method in the extremal control of a many parameter system.” In *Automation and Remote Control*, Vol. 24, No. 10, pp.1337-1342 (1963)
- [7] Nelder, J. A., and Mead, R.: “A simplex method for function minimization.” *Computer Journal*, Vol. 7, Issue 4, pp.308-313 (1965)
- [8] Fogel, L. J., Owens, A. J., and Walsh, M. J.: “Artificial Intelligence through Simulated Evolution.” Wiley (1966)
- [9] Holland, J. H.: “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.” MIT Press (1975)
- [10] Rechenberg, I.: “Evolutionsstrategien.” Springer (1978)
- [11] Schwefel, H. -P.: “Numerische optimierung von computer-modellen mittels der evolution-sstrategie.” Birkhuser (1977)
- [12] Schwefel, H. -P.: “Evolution and Optimum Seeking.” Wiley (1995)
- [13] Eberhart, R., and Kennedy, J.: “A new optimizer using particle swarm theory.” In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp.39-43 (1995)
- [14] Storn, R., and Price, K.: “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces.” *Journal of Global Optimization*, Vol. 11, Issue 4, pp.341-359 (1997)

- [15] Laguna, M., Marti, R., and Campos, V.: “Intensification and diversification with elite tabu search solutions for the linear ordering problem.” *Computers & Operations Research* , Vol. 26, Issue 12, pp.1217-1230 (1999)
- [16] Yasuda, K., and Kanazawa, T.: “Proximate optimality principle based Tabu Search.” *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, pp.1560-1565 (2003)
- [17] Cramer, N. L.: “A Representation for the Adaptive Generation of Simple Sequential Programs.” In *Proceedings of the First International Conference on Genetic Algorithms*, pp.183-187 (1985)
- [18] Koza, J. R.: “Genetic Programming. On the Programming of Computers by Means of Natural Selection.” MIT Press (1992)
- [19] Beowulf.org, <http://www.beowulf.org/>, (2015 Nov. 1)
- [20] Streeter, M. J., Keane, M. A. and Koza, J. R.: “Automatic synthesis using genetic programming of improved PID tuning rules.” *Preprints of the 2003 Intelligent Control Systems and Signal Processing Conference*, Springer Verlag, pp.494-499 (2003)
- [21] Koza, J. R., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J., and Lanza, G.: “Genetic Programming IV: Routine Human-Competitive Machine Intelligence.” Kluwer Academic Publishers (2003)
- [22] Lohn, J., Hornby, G., and Linden, D.: “Evolutionary antenna design for a NASA spacecraft.” *Genetic Programming Theory and Practice II*, pp.301-315, Springer (2004)
- [23] Spector, L., Barnum, H., and Bernstein, H. J.: “Genetic programming for quantum computers.” In *Proceedings of the Third Annual Conference of Genetic Programming*, pp.365-373 (1998)
- [24] Poli, R., Langdon, W. B., and McPhee, N.F.: “A Field Guide to Genetic Programming.” Lulu Enterprises (2008)
- [25] Dean, J., and Sanjay, G.: “MapReduce: simplified data processing on large clusters.” *Communications of the ACM*, Vol. 51, Issue 1, pp.107-113 (2008)
- [26] Welcome to Apache Hadoop!, <https://hadoop.apache.org/>, (2015 Nov. 1)
- [27] Sadasivam, G. S., and Selvaraj, D.: “A novel parallel hybrid PSO-GA using MapReduce to schedule jobs in Hadoop data grids.” *Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pp.377-382 (2010)
- [28] Cavicchio, D. J.: “Adaptive search using simulated evolution.” Ph.D Thesis, University of Michigan, Computer and Communication Sciences Department (1970)
- [29] Kramer, O.: “Evolutionary self-adaptation: a survey of operators and strategy parameters.” *Evolutionary Intelligence*, Vol. 3, Issue 2, pp.51-65 (2010)

- [30] 金政 実, 相吉 英太郎: 「遺伝プログラミングによるアルゴリズムコントローラ的设计」, 電気学会 産業計測制御研究会資料, IIC-09-12, pp.63-66 (2009)
- [31] 金政 実, 相吉 英太郎: 「遺伝的プログラミングによる PSO に対するアルゴリズム調整器の提案と设计」, 電気学会 電子・情報・システム部門大会 講演論文集, TC14-9, pp.544-549 (2012)
- [32] Kanemasa, M., and Aiyoshi, E.: “Algorithm Tuners for PSO Methods and Genetic Programming Techniques for Learning Tuning Rules.” *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 9, No. 4, pp.407-414 (2014)
- [33] 金政 実, 相吉 英太郎: 「遺伝的プログラミングを用いた進化戦略における創発的パラメータ調整機構的设计」, 電気学会 電子・情報・システム部門大会 講演論文集, TC15-3, pp.575-580 (2013)
- [34] 金政 実, 相吉 英太郎: 「多点型進化戦略と遺伝的プログラミングによる創発的パラメータ調整ルール的设计」, 電気学会論文誌 C, Vol. 135, No. 3, pp.321-330 (2015)
- [35] 金政 実, 相吉 英太郎: 「回転不変性を考慮した超球交叉による Differential Evolution の改良」, 第5回コンピューターショナル・インテリジェンス研究会, pp.5-9 (2014)
- [36] 金政 実, 相吉 英太郎: 「回転不変性を考慮した超球交叉による Differential Evolution の改良」, 計測自動制御学会論文集, Vol. 51, No. 7, pp.503-511 (2015)
- [37] 金政 実, 相吉 英太郎: 「同期型 Firefly Algorithm の提案と遺伝的プログラミングを用いた調整則设计」, インテリジェント・システム・シンポジウム (FAN), B603 (2015)
- [38] Poli, R., Langdon, W. B., and Holland, O.: “Extending particle swarm optimisation via genetic programming.” In *Proceedings of the 8th European Conference on Genetic Programming*, Lecture Notes in Computer Science, Vol. 3447, pp.291-300 (2005)
- [39] 金政 実, 相吉 英太郎: 「遺伝的プログラミングを用いた問題特化型 Differential Evolution の自動设计」, 計測自動制御学会 システム・情報部門学術講演会 (SSI2010) 講演論文集, 2D4-1 (2010)
- [40] Kanemasa, M. and Aiyoshi, E.: “Designing Optimal Updating Rule for Differential Evolution Using Genetic Programming.” In *Proceedings of SICE Annual Conference 2013*, pp.1548-1549 (2013)
- [41] Yang, X. S.: “Nature-inspired metaheuristic algorithms.” Luniver Press (2008)
- [42] Shi, Y., and Eberhart, R.: “Parameter selection in particle swarm optimization.” *Evolutionary Programming VII*, Springer, pp.591-600 (1998)
- [43] Clerc, M.: “The swarm and the queen: towards a deterministic and adaptive particle swarm optimization.” In *Proceedings of the congress on Evolutionary Computation*, Vol. 3, pp.1951-1957 (1999)

- [44] Eberhart, R. C., and Shi, Y.: “Comparing inertia weights and constriction factors in particle swarm optimization.” In *Proceedings of the congress on Evolutionary Computation*, Vol. 1, pp.84-88 (2000)
- [45] Clerc, M., and Kennedy, J.: “The particle swarm - explosion, stability, and convergence in a multidimensional complex space.” *IEEE Transactions on Evolutionary Computation*, Vol. 6, Issue 1, pp.58-73 (2002)
- [46] Trelea, I. C.: “The particle swarm optimization algorithm: Convergence analysis and parameter selection.” *Information Processing Letters*, Vol. 85, Issue 6, pp.317-325 (2003)
- [47] Koguma, Y., and Aiyoshi, E.: “Stability Analysis in Consideration of Random Numbers for Particle Swarm Optimization Dynamics : The Best Parameter for Sustainable Search.” *IEEJ Transactions on Electronics, Information and Systems*, Vol. 130, No. 1, pp.29-38 (2010)
- [48] Schutte, J. F., and Groenwold, A. A.: “A study of global optimization using particle swarms. *Journal of Global Optimization*.” Vol. 31, Issue 1, pp.93-108 (2005)
- [49] Yasuda, K., Iwasaki, N., Ueno, G., and Aiyoshi, E.: “Particle swarm optimization: a numerical stability analysis and parameter adjustment based on swarm activity.” *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 3, Issue 6, pp.642-659 (2008)
- [50] 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)
- [51] Bck, T., Hoffmeister, F., and Schwefel, H.-P.: “A survey of evolution strategies.” In *Proceedings of the Fourth International Conference on Genetic Algorithms*, Vol. 9, pp.2-9 (1991)
- [52] Ostermeier, A., Gawelczyk, A., and Hansen, N.: “A Derandomized Approach to Self-Adaptation of Evolution Strategies.” *Journal of Evolutionary Computation*, Vol. 2, pp.369-380 (1994)
- [53] Storn, R.: “On the usage of differential evolution for function optimization.” In *Proceedings of Biennial Conference of the North American Fuzzy Information Processing Society*, pp.519-523 (1996)
- [54] Koguma, Y., and Aiyoshi, E.: “Search Points Distribution Analysis for Differential Evolution Based on Maximum Entropy Method.” *IEEJ Transactions on Electronics, Information and Systems*, Vol. 134, No. 9, pp.1341-1347 (2014)
- [55] Brest, J., Greiner, S., Boskovic, B., Mernik, M., and Zumer, V.: “Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems.” *IEEE Transactions on Evolutionary Computation*, Vol. 10, Issue 6, pp.646-657 (2006)

- [56] Soliman, O. S., and Bui, L. T.: “A self-adaptive strategy for controlling parameters in Differential Evolution.” IEEE Congress on Evolutionary Computation, pp.2837-2842 (2008)
- [57] Zhang, J., and Sanderson, A. C.: “JADE: Adaptive differential evolution with optional external archive.” IEEE Transactions on Evolutionary Computation, Vol. 13, Issue 15, pp.945-958 (2009)
- [58] Wu, Z. F., Huang, H. K., Yang, B., and Zhang, Y.: “A modified differential evolution algorithm with self-adaptive control parameters.” In *Proceedings of 2008 3rd International Conference on Intelligent System and Knowledge Engineering*, Vol. 1, pp.524-527 (2008)
- [59] Ji, R., Tamura, K., and Yasuda, K.: “Adaptive parameter adjustment of Differential Evolution.” In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp.3915-3920 (2013)
- [60] Yang, X. S.: “Nature-inspired optimization algorithms.” Elsevier (2014)
- [61] Hassanzadeh T., and Meybodi M. R.: “A new hybrid approach for data clustering using firefly algorithm and K-means.” In *Proceedings of IEEE 16th CSI International Symposium on Artificial Intelligence and Signal*, pp.7-11 (2012)
- [62] Goel S., and Panchal, V. K.: “Performance evaluation of a new modified firefly algorithm.” In *Proceedings of IEEE 3rd Int. Conf. on Reliability, Infocom Technologies and Optimization*, pp.1-6 (2014)
- [63] Auger, A., and Hansen, N.: “Tutorial CMA-ES: evolution strategies and covariance matrix adaptation.” In *GECCO (Companion)*, pp.827-848 (2012)
- [64] Aleti, A., and Morser, I.: “Studying feedback mechanisms for adaptive parameter control in evolutionary algorithms.” IEEE Congress on Evolutionary Computation (CEC), pp.3117-3124 (2013)
- [65] Breiman, L.: “Random forests.” *Machine learning*, Vol. 45, Issue 1, pp.5-32 (2001)
- [66] COCO (COMparing Continuous Optimisers), <http://coco.gforge.inria.fr/>, (2015 Nov. 1)
- [67] Benjamini, Y, and Yekutieli, D.: “False discovery rate-adjusted multiple confidence intervals for selected parameters.” *Journal of the American Statistical Association*, Vol. 100, Issue 469, pp.71-81 (2005)
- [68] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P.: “Optimization by Simulated Annealing.” *Science News Series*, Vol. 220, No. 4598, pp.671-680 (1983)
- [69] Grueninger T., and Wallace, D.: “Multi-modal optimization using genetic algorithms.” *MIT CADlab-Technical Report: 96.02* (1996)

- [70] Takahama, T., and Sakai, S.: “Solving nonlinear optimization problems by Differential Evolution with a rotation-invariant crossover operation using Gram-Schmidt process.” In *Proceedings of Nature and Biologically Inspired Computing (NaBIC 2010)*, pp.526-533 (2010)
- [71] Guo, S., and Yang, C.: “Enhancing Differential Evolution Utilizing Eigenvector-Based Crossover Operator.” *IEEE Transactions on Evolutionary Computation*, Vol. 19, Issue 1, pp.31-49 (2014)
- [72] Marsaglia, G.: “Choosing a Point from the Surface of a Sphere.” *Annals of Mathematical Statistics*, Vol. 43, No. 2, pp.64-646 (1972)
- [73] Lin, L. J.: “Self-improving reactive agents based on reinforcement learning, planning and teaching.” *Machine learning*, Vol. 8, Issue 3, pp.293-321 (1992)
- [74] 相吉 英太郎, 安田 恵一郎: 「メタヒューリスティクスと応用」, オーム社 (2007)
- [75] CEC2014 (Congress on Evolutionary Computation 2014), http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014/CEC2014.htm, (2015 Nov. 1)
- [76] Sandgren, E.: “Nonlinear and Discrete Programming in Mechanical Design Optimization.” *Journal of Mechanical Design*, Vol. 112, Issue 2, pp.223-229 (1990)
- [77] 情報知能システムの新展開とその産業応用調査専門委員会: 「産業応用のための最適化ベンチマーク問題集」, 電気学会技術報告, No. 1287, pp.1-37 (2013)
- [78] 東京大学教養学部統計学教室: 「自然科学の統計学」, 東京大学出版会 (1992)
- [79] Student: “On the error of counting with hmcycrometer.” *Biometrika*, Vol. 5, Issue 3, pp.351-360 (1907)
- [80] Student: “The probable error of a mean.” *Biometrika*, Vol. 6, Issue 1, pp.1-25 (1908)
- [81] Welch, B. L.: “The significance of the difference between two means when the population variances are unequal.” *Biometrika*, Vol. 29, Issue 3-4, pp.1-25 (1938)
- [82] Welch, B. L.: “The generalization of ”Student’s” problem when several different population variances are involved.” *Biometrika*, Vol. 34, Issue 1-2, pp.28-35 (1947)
- [83] 森棟 公夫: 「統計学入門 (新経済学ライブラリ)」, 新世社 (2000)
- [84] Wineberg, M. and Christensen, S.: “Statistical analysis for evolutionary computation: advanced techniques.” In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pp.2661-2682 (2010)
- [85] Downey, A. B., 黒川 洋, 黒川 利明: 「Think Stats プログラマのための統計入門」, オライリージャパン (2012)
- [86] web 宝くじシミュレーター, <http://homepage2.nifty.com/kaz/takarakuji/>, (2015 Nov. 1)

- [87] Lombardi, C. M. and Hurlbert, S. H.: “Misprescription and misuse of one-tailed tests.” *Austral Ecology*, Vol. 34, Issue 4, pp.447-468 (2009)
- [88] Schulman, R. S.: “Statistics in Plain English.” Van Nostrand Reinhold (1992)
- [89] Mann, H. B., Whitney, D. R.: “On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other.” *Annals of Mathematical Statistics*, Vol. 18, No. 1, pp.50-60 (1947)
- [90] Ruxton, G. D.: “The unequal variance t-test is an underused alternative to Student’s t-test and the Mann-Whitney U test.” *Behavioral Ecology*, Vol. 17, Issue 4, pp.688-690, (2006)
- [91] Zimmerman, D. W.: “A note on consistency of non-parametric rank tests and related rank transformations.” *British Journal of Mathematical and Statistical Psychology*, Vol. 65, Issue 1, pp.122-144, (2012)
- [92] 高木 英行: 「使える！統計検定・機械学習 3群以上の場合の有意差検定」, システム制御情報学会誌, Vol. 58, No. 10, pp.432-438 (2014)

付録A 最適化アルゴリズムの評価

A.1 一般的なベンチマーク問題

最適化アルゴリズムの評価においてはベンチマーク問題が比較的良好に用いられる。文献 [74] より有名なものをあげると、

2^N -minima:

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

$$\mathbf{x}^{opt} \approx (-2.90, -2.90, \dots, -2.90)$$

Alpine:

$$E(\mathbf{x}) = \sum_{n=1}^N |x_n \sin x_n + 0.1x_n| \quad (\text{A.2})$$

$$\mathbf{x}^{opt} = (0, 0, \dots, 0)$$

Levy:

$$E(\mathbf{x}) = \frac{\pi}{N} \sum_{n=1}^{N-1} [(x_n - 1)^2 (1 + 10 \sin^2(\pi x_{n+1}))] + 10 \sin^2(\pi x_1) + (x_N - 1)^2 \quad (\text{A.3})$$

$$\mathbf{x}^{opt} = (1, 1, \dots, 1)$$

Rastrigin:

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

$$\mathbf{x}^{opt} = (0, 0, \dots, 0)$$

Rosenbrock:

$$E(\mathbf{x}) = \sum_{n=1}^{N-1} (100(x_n^2 - x_{n+1})^2 + (1 - x_n)^2) \quad (\text{A.5})$$

$$\mathbf{x}^{opt} = (1, 1, \dots, 1)$$

などがある。ただし、これらのベンチマーク問題の多くは等方向的、変数分離可能で、解が原点付近にある単調な問題であることが多い。そのため、次の回転

$$\mathbf{R}(\boldsymbol{\alpha}(k)) = \prod_{i=1}^N \prod_{j=0}^N \mathbf{R}^{i,j}(\alpha)$$

$$r_{k,l}^{i,j}(\alpha) = \begin{cases} \cos(\alpha) & k = i, l = i \\ -\sin(\alpha) & k = i, l = j \\ \sin(\alpha) & k = j, l = i \\ \cos(\alpha) & k = j, l = j \\ 1 & k = l \neq i, k = l \neq j \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.6})$$

の行列を計算し、目的関数を $E(\mathbf{R}(\boldsymbol{\alpha})\mathbf{x})$ と変換することや、最適解の平行移動等が推奨されるが、問題の等方向性は依然として残る。近年においてはさらに変換をかけ、悪条件の問題も多い COCO^[66] や、CEC^[75] 等のベンチマーク集、大会があり、それらの目的関数を使用することができる。アルゴリズムの評価においてはベンチマーク関数の目的関数値の平均値、最適解への収束確率が多くの場合において評価指針として使われる。

これらの問題は人工的に作られた問題だが、現実の応用問題のベンチマークとしては文献 [76][77] が参考となる。

A.2 最適化アルゴリズム比較のための統計的検定

研究の工程にデータの収集とその評価がある。かつてと変わらず今でも多くの研究はデータ収集において苦勞する。薬学、心理学、物性等、実験を伴う分野においてはデータ収集のために十分な準備が必要となり、実験に不手際があれば、再度実験を行う必要がある。一方、シミュレーションの分野は、実験のための下準備のプログラミング等は必要だが、それさえ行えばデータ収集はコンピューターが自動的に行う。そのため、大量のデータを、比較的短時間で入手することができる。

検定が注目を浴び、現在でも論文の採択基準として大きな比重を占めることになった要因の一つに、少ないデータから確からしい結論を導くことができることが挙げられる。一方、メタヒューリスティクスの分野においては、大量のデータを用意できることから、標本を統計的に比較することが軽んじられている傾向がある。また、標本サイズも論文によって異なり、標本サイズ 50~1,000、つまり 50~1,000 回の試行回数の標本平均をもとにアルゴリズムを比較し、評価している論文など様々である。

これらの論文の問題点として、評価の信頼性がわからないことが挙げられる。標本サイズ 50 に問題があり、標本サイズ 1,000 であれば十分ということではなく、結論がどの程度信頼できるのかわからない点が問題である。仮に標本平均を比べて、既存手法より良い結果が出ていたとしても、母集団の平均も同じ結果であるとは限らない。検定を用いれば、我々の結論が確からしい、ということを手張りで主張できる。

その点において統計は有用だが、いざ統計学の教科書を手本に学び始めると、我々の疑問の解決や実用には往々にして答えてくれない。例として文献 [78] では第 3 章の 79 ページ

Table A.1: Simulation results of PSO and DE on Rosenbrock function, 10 trials.

	PSO	DE
x_1	2.839	0.205
x_2	3.580	7.289
x_3	4.164	7.317
x_4	7.150	6.234
x_5	6.091	7.351
x_6	7.167	6.936
x_7	5.407	6.004
x_8	2.424	5.674
x_9	6.832	5.228
x_{10}	8.839	2.160
\bar{X}	5.449	5.440
s	2.128	2.405

目から我々に興味がある実験データの検定方法について書かれているが、用語、方法と手順が限りなく簡素に書かれてあり、何故検定を行うのか等については触れていないだけでなく、間違った検定を行うことになる。検定が何を計算しており、計算した結果出てくる数値の意味を理解するには文献を隅々まで読む必要がある。そこで、ここではメタヒューリスティクスにおけるアルゴリズムの評価に焦点を当てて、有用な統計手法と、それらの考え方や、数値の見方を例を用いて記す。

例として有名なアルゴリズムである PSO と DE を使い、特定のパラメータ設定において Rosenbrock 関数を解いた。どちらも個体数 $P = 20$ 、最大イテレーション $K = 1,000$ とし、ファンクションコール数は統一させた。10 回試行をおこなった結果は Table A.1 となった。この場合、統計学用語で、標本の大きさは 2 であり、標本サイズはそれぞれ 10 である。ここで、Table A.1 の標本平均 \bar{X} より、DE は PSO より優れていると判断するのは間違いである。DE の母集団の平均 μ_{DE} が PSO の母集団の平均 μ_{PSO} より小さければ、目的関数値の平均という観点から DE の方が優れているという結論は出せるが、我々が得られるのは標本平均だけである。そのため、母集団の平均に関する情報を得る必要がある。母集団の平均と標本平均の誤差について教えてくれるのが中心極限定理である。

母集団の平均を μ 、標本平均を \bar{x} 、母集団の分散を σ^2 、標本分散を s^2 とする。また、標本サイズは n とする。大数の法則によると、標本サイズ n を大きくすると、標本平均 \bar{x} は母集団の平均 μ に近づく。 \bar{x} と μ の誤差に関して教えてくれるのが中心極限定理であり、母集団の分布によらず (例外あり)、 n が十分大きければ、その標本平均の標本は平均が μ 、分散が σ^2/n の正規分布で近似できるというのが中心極限定理である。なお、母集団が正規分布であれば、標本サイズによらず、常に中心極限定理が成立する。なお、正規分布で近似できるまでの n がどの程度なのかに関しては、中心極限定理は触れていない。

例としてサイコロを考えてみる。 $n = 1$ の場合、この平均、つまりサイコロの目の標本はもとの一様な分布である。 $n = 10$ 、 $n = 100$ と増やすにつれて、その平均の標本、つまりサイコロを n 回振った平均を、例えば 500 個集めてヒストグラムにすると、その分布は

平均が $\mu = 3.5$ 、分散が $2.92/n$ の正規分布に近づく。つまり、 n が十分大きければ標本平均 \bar{x} は中心が $\mu = 3.5$ 、分散が $\sigma^2/n = 2.92/n$ の正規分布乱数の確率変数であると考えられる。

$$\bar{x} = N(\mu, \sigma^2/n) \quad (\text{A.7})$$

ここで式変形すると

$$Z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1) \quad (\text{A.8})$$

となる。つまり標準正規分布乱数を 1 個生成したら、偶然左辺の値になったということである。我々が関心を持っているのは μ だが、母分散 σ^2 も式に入っている。そこで、代わりに標本分散 s^2 の使用を考える。

$$Z' = \frac{\bar{x} - \mu}{s/\sqrt{n}} \quad (\text{A.9})$$

標本分散は、正規分布の二乗の和であり、これは χ^2 分布として知られている。これをもとに、 Z' は正規分布ではなく自由度 (パラメータ) が $n - 1$ の t 分布になると発見したのが Gosset (Student)^{[79][80]} である。

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} \quad (\text{A.10})$$

n が大きくなるにつれ、 t 分布は正規分布に近づく。式変形をすると、

$$\mu = \bar{x} - ts/\sqrt{n} \quad (\text{A.11})$$

が得られる。自由度が 9 の t 分布の両側 5% の棄却域、つまり原点付近の面積が 0.95 になるのは $t = \pm 2.262$ なので、

$$\mu = \bar{x} \pm 2.262s/\sqrt{10} \quad (\text{A.12})$$

母集団の平均は 95% の確率でこの範囲にあることがわかる。母集団の 95% がこの範囲にあるのではなく、母集団の平均が 95% の確率でこの範囲にある。

ここでもし、信頼区間が重なっていなければ、同じ水準でアルゴリズム間の比較ができることを説明する。アルゴリズム A と B があり、「A の目的関数値の平均の信頼区間 > B の目的関数値の平均の信頼区間」であるとすれば、「A の目的関数値の標本平均 > B の目的関数値の標本平均」の確率は少なくとも

$$(1 - \alpha/2)^2 > 1 - \alpha \quad (\text{A.13})$$

となり、 α の水準でアルゴリズムの優劣関係が成立する。

信頼区間が重なっている場合は、二つのアルゴリズムの標本平均の差を考える。中心極限定理が成立すれば、 \bar{x}_A は正規分布に従い、 \bar{x}_B も正規分布に従う。正規分布の重ね合わせは、正規分布になるため、二つの分布の分散が同じであるとすれば、

$$t = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{(s_A^2 + s_B^2)/n}} \quad (\text{A.14})$$

は自由度 $2n - 2$ の t 分布に従う。なお、これを近似を使い異分散でも検定できるようにしたのが Welch の t 検定 [81][82] である。Welch の t 検定では、別の自由度の t 分布を使い検定を行う。その自由度は A と B の標本サイズをそれぞれ n_A と n_B とした場合、

$$\nu = \frac{\left(\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}\right)^2}{\frac{s_A^4}{n_A^2(n_A-1)} + \frac{s_B^4}{n_B^2(n_B-1)}} \quad (\text{A.15})$$

とした、自由度 ν を使う。古い資料には、まず等分散かどうかを検定して、異分散であれば Welch の t 検定を使用するように書かれてあるが、検定の多重性の問題があるため、近年では等分散、異分散に関わらず Welch の t 検定を使用することが推奨されている。我々の分野においても、二つのアルゴリズムの分散が同じ、もしくはほぼ同じかどうかを問題ごとに考える手間も省けるため、平均値の検定を行う場合は、Welch の t 検定を推奨する。

A.2.1 母集団の分布に対する仮定

今までは述べなかったが、 t 検定を使用する前提として、次のように書かれている。

「母集団は正規分布に従う」

しかし、現実的には母集団が正規分布に従うことは稀であり、一見すると統計的検定は限定的な場合にのみ適用可能であると思われる。この一文が書かれているのは、中心極限定理が多くの場合、近似であるためである。母集団が正規分布であれば、正規分布の重ね合わせは正規分布であり、中心極限定理は近似ではなく、標本サイズによらず完璧に成立する。一方、母集団が正規分布に従わない場合、文献 [78] には次のように書いてある。「いま、正規分布の仮定が正しくなく (中略)。 n がある程度大きければ、中心極限定理より、(中略)。つまり t 検定を用いても結論の妥当性は失われまいということになる。」。また、文献 [83] には「母集団における分布が未知なら、(中略)。このような検定の手続きは、近似的な正規分布に依存しているが、結果として導かれる棄却域は正規母集団の場合と変化ない。」。これらの文章の意味は、母集団が正規分布に従わない場合であっても、 n が十分大きく、大数の法則と中心極限定理で標本平均の分布が正規分布で近似できれば、 t 検定を用いてもよいということである。ただし、我々の分野においては、特定の場合、この部分に関して注意が必要となることを後ほどシミュレーションを用いて説明する。

今、仮に Table A.1 のシミュレーションにおいて大数の法則と中心極限定理により標本平均の分布が正規分布で近似できているとする。その場合、式 (A.12) より、それぞれの母集団の平均は $\mu_{DE} = 5.440 \pm 1.632$ 、 $\mu_{PSO} = 5.449 \pm 1.444$ の間にある。そのため、二つの差に関して検定を行う必要がある (正確には信頼区間を見てから差に関して検定を行うと検定の多重性の問題があるため、正しくないが、ここでは説明のため、その議論は後半にまわす)。

差の検定の際には、流れとして二つの仮説を設定する。一つは帰無仮説 H_0 であり、二つの母集団の平均は等しい、つまり今回の場合は $H_0 : \mu_{PSO} = \mu_{DE}$ を設定する。二つ目は対立仮説 $H_1 : \mu_{PSO} \neq \mu_{DE}$ であり、一般的にはこちらが成立して欲しい仮説である。

Welch の t 検定の式に従い、 t 値を計算すると $t = 0.00935$ 、分布の自由度から計算した t 分布の両端の面積である P 値は $P = 0.992$ となる。もし、PSO と DE に大きな差があれば、 t 値の絶対値は大きくなり、両端の面積 P は小さい値を取る。 P が、我々が事前に設

定した有意水準 $\alpha = 0.05$ より小さければ、二つの間には有意な差があると考えるのが差の検定である。

今回、 t 値は PSO の標本平均の方がわずかに大きかったため、 t 値は正の値を取った。また、その絶対値は小さかったため、 $P = 0.992$ となり、二つの母集団に有意な差、つまり PSO と DE の間には確率的に考えて大きな差はなかったと判断される。

A.2.2 片側検定

上記の検定には哲学上、次の問題が生じる。我々が成立させようとした対立仮説 $H_1: \mu_{\text{PSO}} \neq \mu_{\text{DE}}$ は、両者に差があるということであり、PSO と DE のどちらが優秀かに関しては触れていない。一般的に我々の目的は新手法が既存手法より優秀だと示すことである。それであれば t 分布の両端を見るのではなく、片端のみを考え、対立仮説を設定するのが片側検定の考え方である。既存手法として PSO を考え、新手法の New Algorithm (NA) なるものが存在するとすると、対立仮説として $H_1: \mu_{\text{PSO}} > \mu_{\text{NA}}$ を設定することに相当する。これにより、今まで両端に振られていた α が片端に集中し、 P 値の基準を 2 倍緩くすることができる。教科書も片側検定に関してはこれ以上言及されておらず、一見するとアルゴリズムの比較には片側検定が適切に見えるが、文献 [87] によると今回の場合に限らず、多くの場合において片側検定は使うべきではない。理由は次の通りである。仮に新手法のアルゴリズム New Algorithm (NA) を開発し、それを PSO と比較するとする。我々のアルゴリズムの方が優秀であると予測を立て、有意水準 $\alpha = 0.05$ を設定し、帰無仮説 $H_0: \mu_{\text{PSO}} \leq \mu_{\text{NA}}$ 、対立仮説 $H_1: \mu_{\text{PSO}} > \mu_{\text{NA}}$ を設定し、 P 値を計算したところ、予想に反して $P = 0.98$ となった。この場合、結果の解釈や、それに対する対応に困るのである。対応としてもう一方の方向の片側に関して片側検定を同じ有意水準で行うと、検定の多重性により、 $(1 - \alpha)^2 \sim 1 - 2\alpha$ となり、有意水準が 2 倍になる (つまり 95% の正しさではなく、90% の正しさに減る)。文献 [87] では様々なシナリオや文献の引用を行い考察しているが、総括すると最初から両側検定をするべきであるというのが一般的な対応策であり、どちらが優秀かに関しては、研究者が理由をつけるべきであるとのことである。例えば、今回は改良を行ったことにより、 t 値の符合が我々の手法の方が良いとされる方向に出て、有意差が認められた等の理由をつけて手法の優劣に関して言及するということである。新薬開発においては、テスト項目が限られており、研究開発の成果であると主張すればいいかもしれないが、我々の分野においては、ベンチマークによって優劣が変わり、一括りに改良によるものと言及するのは好ましくないと考える。そこで、文献 [88] に賛同し、「 γ 過誤の可能性は無視できるとする」という一文を付け加え、 t 値の符合で優劣を判断し、残りは読者の判断に任せることにするのが妥当であると考えられる。

ここで新しく出た「 γ 過誤」を説明する前に、過誤について説明する。Mosteller によると統計には次の三つの過誤、つまり間違った結論の出し方がある。

- (a) 第一種過誤：真である帰無仮説を棄却する。つまり、本当はアルゴリズム間に差はないが、差があると判断する間違い (α)。
- (b) 第二種過誤：偽である帰無仮説を採択する。つまり、本当はアルゴリズム間に差があるのに、差がないと判断する間違い (β)。

Table A.2: Results on t -test at different sample size.

Sample Size	t -value	P -value
10	0.00938	0.993
50	-2.38	0.0209
100	-0.561	0.576
500	-4.10	4.60E-5
1000	-6.03	2.02E-9

(c) 第三種過誤：間違っただ理由で、正しく帰無仮説を棄却する (γ).

統計は第一種過誤に対して基準を設定し、判断をする。我々は(本当は)アルゴリズム間に差はないのに、間違っただと判断したくないということである。有意水準 $\alpha = 0.05$ は、この間違っただ判断をする確率を 5%にする。第二種過誤の β は α とは相反する関係にある。 α の基準を強くし、第一種過誤を起こりにくくすると、第二種過誤の可能性が起こりやすくなる。このため、 α は一概に厳しく設定することには問題があり、適切な値に設定する必要がある。第三種過誤の γ 過誤は次のようなものである。前節において、片側検定は使うべきではなく、両側検定を使うべきであると説明したが、それによって成立する対立仮説は $H_1: \mu_{\text{PSO}} \neq \mu_{\text{DE}}$ である。そこで、 t 値の符号から、どちらが優秀かを判断すると述べたが、確率的に考えれば、実は逆が正しい場合も存在し、この判断は間違っただいる可能性が少なからずある。そのため、前節の一文を追記することを推奨する。

検定の議論は母集団が正規分布でなくとも適用できることは説明したが、1点だけ注意する必要がある。大数の法則と中心極限定理によって標本平均が正規分布で近似される十分な n が必要がある。これが成立していない場合、検定に関する議論や、導出される数式、計算によって得られる値は意味がなく、間違っただものとなる。今回の Rosenbrock 関数における二つのアルゴリズムを例にすると、試行回数によって、検定の結果は次のように変化する。有意水準を $\alpha = 0.05$ とした場合、試行回数 50 において一旦有意差が認められるが、試行回数 100 に増やした際に認められない結果が出ており、問題があることがわかる。有意差が認められない点に問題があるのではなく、検定の結果が大きくブレている点に問題がある。

上記の結果は試行回数、つまり標本サイズによって結論が異なっている。この理由は、標本サイズが不十分で、中心極限定理により標本平均の分布が正規分布で近似されないことによるものである。文献 [85] によると、「中心極限定理が正規分布に収束するのに必要な標本サイズは分布の歪度に依存する。」とある。

中心極限定理により標本平均が正規分布で近似されるのが遅い例として宝くじが挙げられる。宝くじは大体 50% が税金として徴収されるため、期待値は 0.5 である。しかし、一例として Web 上の宝くじシミュレーター^[86] を用いてモンテカルロシミュレーションを行うと期待値は大抵の場合 0.2~0.3 を示す。宝くじは分布の裾の部分が大きく平均値に寄与する非対称な分布であり、十分な試行回数、ないしは標本サイズがなければ中心極限定理が成立しない。宝くじの期待値は確かに理論的に有限で 0.5 であり、分散も有限の値である。一方で、我々が観測する宝くじの期待値は、基本的には 0.2~0.3 程度の値となり、運が良ければ大きな値を取る。この差は、中心極限定理が成立していないことによるもので

あり、正しい期待値を計算するには十分な試行回数、ないしは標本サイズを用意する必要がある。

前節の Rosenbrock 関数も、アルゴリズムの試行によって得られる目的関数値の分布は裾が広いので、十分な標本サイズを取らなければ、検定の数式に当てはめるのは間違っており、数値にも意味がない。Table A.2 の結果を参考にすると、500 以上であれば収束しているように見られる。この結果を考慮し、Rosenbrock 関数の平均値は 500 以上の試行回数を行い、検定することを推奨する。

つまり、今回の例で挙げた 10 回試行での平均値の比較は、標本サイズが不十分であるため、間違っている。

A.2.3 ランク化 t 検定

今までは深く考えず、目的関数値の平均値がアルゴリズムの性能の指標であるかのように扱ってきたが、これは正しくなく、評価の基準はユーザーの解きたい問題の性質や考え方に依存する。多くの場合、目的関数値に意味はなく、その結果の良し悪しもリニアに影響してくるものではない。重量最小化問題があった場合、150.1g と 150g の解があった場合、重さの観点からは 0.1g の改善がある。そういう考え方もあるが、設計者の観点からは他者よりも優れている解を求めたいという考え方もある。その場合、平均値ではなく中央値を評価基準として使うべきである（平均値より良い解を得られる確率は 50% ではないため）。また、平均値を評価基準とした場合、目的関数を変換することによって、評価が変わる。本来の Rosenbrock 関数は次式で与えられる。

$$E(\mathbf{x}) = \sum_{n=1}^{N-1} (100(x_n^2 - x_{n+1})^2 + (1 - x_n)^2) \quad (\text{A.16})$$

この目的関数を変換し、次の Kanemasabrock 関数なるものを考えてみる。

$$E(\mathbf{x}) = \left(\sum_{n=1}^{N-1} (100(x_n^2 - x_{n+1})^2 + (1 - x_n)^2) \right)^2 \quad (\text{A.17})$$

すると、結果は Table 3 になる。Table A.1 と比べて標本平均の優劣が逆転する。これに関しては、目的関数の設定は与えられたものであり、その目的関数値の最小化に意味があり、目的関数に変換をかけて値を操作する余地はないという考え方もある。中央値を評価基準とした場合、このような変換に左右されない結果を得ることができる。また、これから説明するランク化 t 検定を用いれば Rosenbrock 関数を用いた場合でも 1,000 回試行せずに、検定を行うことができる。

正確な中央値の検定方法は存在しない。代替として、Mann-Whitney U 検定^[89](Wilcoxon 順位和検定)やランク化 t 検定が使われる。これらの手法は、どちらかの手法が一般的により良い結果を出しているかどうかを検出する。文献 [90][91]によると、ランク化 t 検定の方が良いとのことなので、今回はこの手法を説明する。手順としてはまず、データの併合順位を求める。同着のものがあれば、順位の平均を割り当てる（例えば同着 1 位が 3 個あれば、 $(1 + 2 + 3)/3 = 2$ より、順位は 3 個とも 2 を割り当てる）。今回の例では Table A.4 になる。その後、それぞれの手法の順位の平均と分散を求め、この平均値の差に対し

Table A.3: Simulation results of PSO and DE on Kanemasabrock function, 10 trials.

	PSO	DE
x_1	5.88	0.042
x_2	8.06	4.67
x_3	12.82	27.33
x_4	17.34	32.19
x_5	29.24	36.05
x_6	37.10	38.86
x_7	46.68	48.11
x_8	51.12	53.13
x_9	51.37	53.54
x_{10}	78.13	54.04
\bar{X}	33.77	34.80
s	23.39	19.53

Table A.4: Ranks of simulation results of PSO and DE on Rosenbrock function, 10 trials.

	PSO	DE
x_1	4	1
x_2	5	17
x_3	6	18
x_4	15	12
x_5	11	19
x_6	16	14
x_7	8	10
x_8	3	9
x_9	13	7
x_{10}	20	2
\bar{X}	10.1	10.9
s	5.78	6.33

て Welch の t 検定を行う。十分な試行回数があれば中心極限定理により順位の平均値は正規分布に近似し、Rosenbrock 関数のような、稀に極端な値を取る目的関数であっても、順位へと変換しているため、中心極限定理の成立も容易になるという考えに基づいた手法である。試行回数に関しては文献 [84] によると最低 30、できれば 50 回が推奨とのことである。今回のアルゴリズムで、試行回数を変えてランク化 t 検定を行った際の結果は Table A.5 のようになる。Table A.2 と比較して、 P 値が低く出ており、 t 値の符号の変動や、大きさの振動もない。結果として、より少ない試行回数でアルゴリズムの優劣を、中央値に似た基準で比較できることがわかる。

Table A.5: Results on Ranked t -test at different sample size.

Sample Size	t -value	P -value
10	-0.295	0.771
50	-4.51	1.82E-5
100	-6.66	2.70E-10
500	-17.3	9.92E-59
1000	-25.6	6.84E-125

A.2.4 検定の多重性

ひとつの論文，ないしは一連の実験で，特定の有意水準 α において N 回検定を行うと，すべての検定結果が正しい確率は $(1 - \alpha)^N \sim 1 - N\alpha$ となる．これが検定の多重性と呼ばれる問題であり，検定，ないしは信頼区間を複数計算する際は注意する必要がある．特に，項目が複数ある場合，組み合わせで検定を行った場合，検定の数是非常に大きくなり，全体での有意水準に影響を与える．分散分析などの手法は，項目間に差があるか効率よく計算するが，具体的にどの項目の組み合わせがどういう結果であるかに関しては教えてくれず，結果として我々の分野においては使いどころが難しい．興味があれば文献 [92][84] が参考になるが，ここでは紹介しない．

我々の目的はアルゴリズムの優劣関係であるため，対比較で検定を多数行う以外方法がなく，あまりに検定回数が大きくなるようであれば，重要な部分の検定のみを行うべきである．その際の有意水準の安易な補正として Bonferroni 法というものがある．具体的には P 値に検定回数 N をかける，ないしは有意水準 α を検定を行った回数と信頼区間を出した回数 N で割って $\alpha' = \alpha/N$ とするものである．検定が厳しくなるが，補正が楽であるという利点がある．つまり，すべての検定結果が正しい確率を $(1 - \alpha')^N = (1 - \alpha/N)^N \sim 1 - \alpha$ と補正し，全体で α の有意水準を保つ手法である．他の手法として Holm 法^[92] などがある．

A.2.5 まとめ

ここまでをまとめると，試行回数はなるべく 30 回以上で，ランク化 t 検定を用いた対比較をおこない，有意水準は Bonferroni 法で補正するのが安易な検定手順である．

付録B 4章の詳細な結果

B.1 PSOの詳細な結果

Table B.1: Results for separable functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{01}	(a)	7.9480e+01	7.9497e+01	6.0454e-03	7.9480e+01	7.9969e+01
f_{01}	(b)	7.9480e+01	7.9480e+01	1.4712e-07	7.9480e+01	7.9482e+01
f_{01}	(c)	7.9480e+01	7.9483e+01	2.9713e-04	7.9480e+01	7.9602e+01
f_{02}	(a)	-2.0962e+02	-1.9509e+02	2.8036e+03	-2.0988e+02	1.0441e+02
f_{02}	(b)	-2.0764e+02	-1.7080e+02	9.3535e+03	-2.0988e+02	3.6164e+02
f_{02}	(c)	-1.6942e+02	-4.6030e+01	8.8602e+04	-2.0974e+02	1.4259e+03
f_{03}	(a)	-3.7007e+02	-3.6045e+02	1.6669e+03	-4.2603e+02	-2.3804e+02
f_{03}	(b)	-3.0488e+02	-2.9851e+02	2.1122e+03	-3.7154e+02	-2.0128e+02
f_{03}	(c)	-2.6644e+02	-2.6369e+02	4.1509e+03	-3.7254e+02	-5.0189e+01
f_{04}	(a)	-3.4058e+02	-3.3439e+02	1.4194e+03	-4.0263e+02	-2.1031e+02
f_{04}	(b)	-2.5764e+02	-2.5104e+02	3.3497e+03	-3.5861e+02	-1.0689e+02
f_{04}	(c)	-1.8664e+02	-1.6484e+02	7.1316e+03	-3.2438e+02	3.6364e+01
f_{05}	(a)	4.4462e+01	4.4982e+01	6.8689e+02	7.3791e-01	1.1450e+02
f_{05}	(b)	6.2263e+01	6.2851e+01	4.6914e+02	2.6165e+01	1.1824e+02
f_{05}	(c)	7.4323e+01	7.4458e+01	5.2498e+02	2.2688e+01	1.3577e+02

Table B.2: Results for low or moderate conditioning functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{06}	(a)	1.1485e+02	1.3029e+02	3.3581e+03	4.1099e+01	2.9493e+02
f_{06}	(b)	8.5467e+01	1.1317e+02	4.1468e+03	4.1561e+01	2.6949e+02
f_{06}	(c)	1.4976e+02	1.6258e+02	4.3666e+03	7.0025e+01	3.9939e+02
f_{07}	(a)	1.3560e+02	1.4104e+02	5.0889e+02	1.0793e+02	2.1084e+02
f_{07}	(b)	1.7278e+02	1.8317e+02	1.9300e+03	1.2675e+02	3.6594e+02
f_{07}	(c)	1.8308e+02	1.9698e+02	2.6144e+03	1.3083e+02	3.7672e+02
f_{08}	(a)	2.2458e+02	2.3129e+02	3.3403e+03	1.5568e+02	4.5651e+02
f_{08}	(b)	2.2186e+02	2.1204e+02	1.2699e+03	1.5320e+02	2.8645e+02
f_{08}	(c)	2.3059e+02	2.3114e+02	2.0918e+03	1.5059e+02	3.7116e+02
f_{09}	(a)	1.6344e+02	1.9595e+02	4.0414e+03	1.4568e+02	4.1020e+02
f_{09}	(b)	1.5895e+02	1.8137e+02	1.1820e+03	1.4686e+02	2.5901e+02
f_{09}	(c)	1.5896e+02	1.8683e+02	2.6220e+03	1.4512e+02	4.1596e+02

Table B.3: Results for high conditioning and unimodal functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{11}	(a)	2.8768e+02	2.9476e+02	3.6437e+03	1.7426e+02	4.5508e+02
f_{11}	(b)	2.5487e+02	2.6089e+02	2.8055e+03	1.5966e+02	4.1555e+02
f_{11}	(c)	2.4711e+02	2.5583e+02	2.6358e+03	1.7646e+02	3.8714e+02
f_{12}	(a)	-5.7216e+02	4.3291e+03	1.0060e+09	-6.1879e+02	2.2397e+05
f_{12}	(b)	-6.0217e+02	1.0358e+03	1.0472e+08	-6.2100e+02	7.1846e+04
f_{12}	(c)	-5.0256e+02	1.0834e+04	3.1362e+09	-6.2017e+02	3.9075e+05
f_{13}	(a)	3.9805e+01	4.8863e+01	1.7933e+03	3.1637e+01	3.3036e+02
f_{13}	(b)	4.1005e+01	4.5567e+01	2.2198e+02	3.0628e+01	9.9369e+01
f_{13}	(c)	4.8158e+01	6.5611e+01	2.6501e+03	3.1797e+01	2.8095e+02
f_{14}	(a)	-5.2339e+01	-5.2329e+01	9.1705e-04	-5.2346e+01	-5.2172e+01
f_{14}	(b)	-5.2346e+01	-5.2345e+01	3.0353e-05	-5.2349e+01	-5.2322e+01
f_{14}	(c)	-5.2343e+01	-5.2277e+01	1.6764e-01	-5.2349e+01	-4.9445e+01

Table B.4: Results for multi-modal functions with adequate global structure functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{15}	(a)	1.2122e+03	1.2380e+03	7.1294e+03	1.1035e+03	1.4609e+03
f_{15}	(b)	1.2509e+03	1.2501e+03	5.1409e+03	1.1056e+03	1.4151e+03
f_{15}	(c)	1.2361e+03	1.2430e+03	4.9308e+03	1.0958e+03	1.4700e+03
f_{16}	(a)	8.7040e+01	8.7073e+01	2.3626e+01	7.8220e+01	9.8868e+01
f_{16}	(b)	8.5773e+01	8.5875e+01	1.5282e+01	7.7621e+01	9.8110e+01
f_{16}	(c)	8.4105e+01	8.5043e+01	1.3847e+01	7.7822e+01	9.4501e+01
f_{17}	(a)	-1.1542e+01	-1.1723e+01	2.5376e+00	-1.4568e+01	-8.2851e+00
f_{17}	(b)	-1.2141e+01	-1.1930e+01	2.1020e+00	-1.4689e+01	-8.6113e+00
f_{17}	(c)	-1.2227e+01	-1.2013e+01	1.2299e+00	-1.3908e+01	-9.5232e+00
f_{18}	(a)	7.2624e-01	1.5194e+00	4.0799e+01	-1.1528e+01	1.8255e+01
f_{18}	(b)	5.1949e-01	2.4876e-01	3.3991e+01	-1.0359e+01	1.4894e+01
f_{18}	(c)	4.4771e-01	6.5058e-01	2.1699e+01	-1.0099e+01	9.1940e+00
f_{19}	(a)	-9.6100e+01	-9.6061e+01	5.7818e-01	-9.7548e+01	-9.3731e+01
f_{19}	(b)	-9.6563e+01	-9.6630e+01	9.2597e-01	-9.8813e+01	-9.2954e+01
f_{19}	(c)	-9.6717e+01	-9.6827e+01	1.7412e+00	-1.0064e+02	-9.4557e+01

Table B.5: Results for multi-modal functions with weak global structure functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{20}	(a)	-5.4495e+02	-5.4496e+02	1.3136e-01	-5.4545e+02	-5.4318e+02
f_{20}	(b)	-5.4488e+02	-5.4492e+02	4.3415e-02	-5.4537e+02	-5.4439e+02
f_{20}	(c)	-5.4485e+02	-5.4485e+02	3.5494e-02	-5.4527e+02	-5.4445e+02
f_{21}	(a)	4.6561e+01	5.0600e+01	1.1902e+02	4.0780e+01	1.0167e+02
f_{21}	(b)	4.6561e+01	5.2489e+01	2.0782e+02	4.0780e+01	1.0167e+02
f_{21}	(c)	4.6565e+01	5.0678e+01	7.5422e+01	4.0780e+01	7.4211e+01
f_{22}	(a)	-9.8541e+02	-9.8448e+02	1.5424e+02	-9.9929e+02	-9.3784e+02
f_{22}	(b)	-9.8542e+02	-9.8496e+02	1.3855e+02	-9.9931e+02	-9.5706e+02
f_{22}	(c)	-9.8542e+02	-9.8475e+02	1.2862e+02	-9.9929e+02	-9.5707e+02
f_{23}	(a)	1.0109e+01	1.0037e+01	3.4453e-01	8.7881e+00	1.1114e+01
f_{23}	(b)	9.7230e+00	9.6697e+00	3.2981e-01	8.3882e+00	1.0707e+01
f_{23}	(c)	8.5800e+00	8.6568e+00	2.9710e-01	7.7340e+00	1.0267e+01
f_{24}	(a)	3.9287e+02	3.9151e+02	9.8619e+02	2.9940e+02	4.4935e+02
f_{24}	(b)	3.3785e+02	3.3935e+02	2.9694e+03	2.3254e+02	4.4363e+02
f_{24}	(c)	2.8401e+02	2.9320e+02	3.2370e+03	1.7057e+02	4.8264e+02

Table B.6: Results for fractal functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{25}	(a)	-1.0000e+00	-1.4600e+00	1.4371e+00	-4.0000e+00	0.0000e+00
f_{25}	(b)	0.0000e+00	-2.4000e-01	1.8612e-01	-1.0000e+00	0.0000e+00
f_{25}	(c)	0.0000e+00	-1.0000e-01	9.1837e-02	-1.0000e+00	0.0000e+00

Table B.7: Pairwise ranked Welch's t -test on algorithms for 30 dimensional separable functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{01}	(a)	(b)	-8.33	1.64e-12
f_{01}	(a)	(c)	-8.72	2.04e-13
f_{01}	(b)	(c)	-1.72	8.80e-02
f_{02}	(a)	(b)	-3.85	2.11e-04
f_{02}	(a)	(c)	-9.60	1.05e-15
f_{02}	(b)	(c)	-5.08	1.83e-06
f_{03}	(a)	(b)	-7.62	1.78e-11
f_{03}	(a)	(c)	-10.27	3.49e-17
f_{03}	(b)	(c)	-2.79	6.41e-03
f_{04}	(a)	(b)	-9.23	5.87e-15
f_{04}	(a)	(c)	-15.14	2.10e-27
f_{04}	(b)	(c)	-5.99	3.51e-08
f_{05}	(a)	(b)	-3.66	4.20e-04
f_{05}	(a)	(c)	-6.14	1.79e-08
f_{05}	(b)	(c)	-2.85	5.37e-03

Table B.8: Pairwise ranked Welch's t -test on algorithms for 30 dimensional low or moderate conditioning functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{06}	(a)	(b)	2.15	3.43e-02
f_{06}	(a)	(c)	-2.74	7.31e-03
f_{06}	(b)	(c)	-4.66	1.07e-05
f_{07}	(a)	(b)	-7.52	2.67e-11
f_{07}	(a)	(c)	-8.85	3.80e-14
f_{07}	(b)	(c)	-1.43	1.55e-01
f_{08}	(a)	(b)	1.17	2.43e-01
f_{08}	(a)	(c)	-1.05	2.97e-01
f_{08}	(b)	(c)	-2.52	1.34e-02
f_{09}	(a)	(b)	0.54	5.89e-01
f_{09}	(a)	(c)	0.56	5.74e-01
f_{09}	(b)	(c)	0.14	8.86e-01

Table B.9: Pairwise ranked Welch's t -test on algorithms for 30 dimensional high conditioning and unimodal functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{11}	(a)	(b)	2.97	3.72e-03
f_{11}	(a)	(c)	3.34	1.20e-03
f_{11}	(b)	(c)	0.56	5.79e-01
f_{12}	(a)	(b)	2.32	2.26e-02
f_{12}	(a)	(c)	-1.86	6.65e-02
f_{12}	(b)	(c)	-3.63	4.53e-04
f_{13}	(a)	(b)	-0.56	5.74e-01
f_{13}	(a)	(c)	-3.00	3.40e-03
f_{13}	(b)	(c)	-2.50	1.39e-02
f_{14}	(a)	(b)	9.02	2.58e-14
f_{14}	(a)	(c)	2.98	3.73e-03
f_{14}	(b)	(c)	-3.91	1.71e-04

Table B.10: Pairwise ranked Welch's t -test on algorithms for 30 dimensional multi-modal functions with adequate global structure functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{15}	(a)	(b)	-1.30	1.96e-01
f_{15}	(a)	(c)	-0.81	4.19e-01
f_{15}	(b)	(c)	0.76	4.47e-01
f_{16}	(a)	(b)	1.35	1.80e-01
f_{16}	(a)	(c)	2.44	1.66e-02
f_{16}	(b)	(c)	1.48	1.43e-01
f_{17}	(a)	(b)	0.75	4.55e-01
f_{17}	(a)	(c)	0.99	3.24e-01
f_{17}	(b)	(c)	0.41	6.86e-01
f_{18}	(a)	(b)	0.76	4.47e-01
f_{18}	(a)	(c)	0.38	7.07e-01
f_{18}	(b)	(c)	-0.77	4.43e-01
f_{19}	(a)	(b)	3.74	3.11e-04
f_{19}	(a)	(c)	3.18	1.97e-03
f_{19}	(b)	(c)	0.27	7.90e-01

Table B.11: Pairwise ranked Welch's t -test on algorithms for 30 dimensional multi-modal functions with weak global structure functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{20}	(a)	(b)	-1.89	6.14e-02
f_{20}	(a)	(c)	-3.15	2.15e-03
f_{20}	(b)	(c)	-1.59	1.15e-01
f_{21}	(a)	(b)	-0.49	6.27e-01
f_{21}	(a)	(c)	-0.60	5.47e-01
f_{21}	(b)	(c)	-0.19	8.48e-01
f_{22}	(a)	(b)	1.81	7.41e-02
f_{22}	(a)	(c)	0.25	8.05e-01
f_{22}	(b)	(c)	-1.30	1.95e-01
f_{23}	(a)	(b)	3.14	2.26e-03
f_{23}	(a)	(c)	12.10	3.72e-21
f_{23}	(b)	(c)	9.39	2.66e-15
f_{24}	(a)	(b)	6.04	3.05e-08
f_{24}	(a)	(c)	11.68	3.69e-20
f_{24}	(b)	(c)	4.27	4.48e-05

Table B.12: Pairwise ranked Welch's t -test on algorithms for 30 dimensional fractal functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{25}	(a)	(b)	-7.37	9.57e-11
f_{25}	(a)	(c)	-9.52	2.06e-14
f_{25}	(b)	(c)	-1.88	6.37e-02

B.2 ES の詳細な結果

Table B.13: Results for separable functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{01}	(a)	9.0950e+01	9.0903e+01	2.1267e+00	8.6487e+01	9.4398e+01
f_{01}	(b)	7.9482e+01	7.9482e+01	2.1996e-07	7.9481e+01	7.9484e+01
f_{01}	(c)	7.9495e+01	7.9497e+01	1.3433e-04	7.9482e+01	7.9534e+01
f_{02}	(a)	3.6870e+05	3.7184e+05	8.9513e+09	1.8671e+05	5.7001e+05
f_{02}	(b)	5.9520e+04	6.1851e+04	3.8562e+08	2.6369e+04	1.0191e+05
f_{02}	(c)	4.8499e+04	5.2253e+04	3.2398e+08	1.7348e+04	8.8931e+04
f_{03}	(a)	-1.1615e+02	-1.1370e+02	5.4317e+02	-1.7942e+02	-6.7996e+01
f_{03}	(b)	-1.8951e+02	-1.7697e+02	4.6377e+03	-2.8099e+02	2.9761e+01
f_{03}	(c)	-1.9090e+01	1.4188e+00	1.5314e+04	-2.1749e+02	3.1275e+02
f_{04}	(a)	-7.8889e+00	-1.1642e+01	1.4033e+03	-1.0329e+02	5.9623e+01
f_{04}	(b)	-5.3535e+00	-6.9674e+00	7.5098e+03	-1.8537e+02	1.7557e+02
f_{04}	(c)	2.1484e+02	2.4327e+02	1.7741e+04	2.5360e+01	5.8703e+02
f_{05}	(a)	2.1273e+02	2.1198e+02	2.5160e+02	1.6989e+02	2.5181e+02
f_{05}	(b)	1.8601e+02	1.8202e+02	6.4585e+02	1.3280e+02	2.3453e+02
f_{05}	(c)	1.9455e+02	1.9845e+02	1.2093e+03	1.1783e+02	2.7623e+02

Table B.14: Results for low or moderate conditioning functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{06}	(a)	7.5268e+03	7.9528e+03	2.0922e+07	1.4076e+03	2.3542e+04
f_{06}	(b)	1.4847e+02	1.4642e+02	1.6110e+03	5.8792e+01	2.4076e+02
f_{06}	(c)	1.1671e+02	1.2427e+02	1.5131e+03	5.9469e+01	2.4586e+02
f_{07}	(a)	2.1044e+02	2.1301e+02	3.8209e+02	1.6908e+02	2.5828e+02
f_{07}	(b)	2.0065e+02	2.0659e+02	2.0086e+03	1.1668e+02	3.3252e+02
f_{07}	(c)	1.9273e+02	1.9655e+02	9.0976e+02	1.4873e+02	2.7647e+02
f_{08}	(a)	2.4086e+03	2.4501e+03	1.7503e+05	1.7525e+03	3.4713e+03
f_{08}	(b)	2.0349e+02	2.0224e+02	5.6074e+02	1.7476e+02	2.5674e+02
f_{08}	(c)	1.8641e+02	1.8995e+02	1.7604e+02	1.7934e+02	2.3782e+02
f_{09}	(a)	2.2211e+03	2.1514e+03	1.2556e+05	1.4534e+03	2.9385e+03
f_{09}	(b)	1.5231e+02	1.5428e+02	8.1242e+01	1.4739e+02	2.0430e+02
f_{09}	(c)	1.5729e+02	1.5802e+02	2.2561e+01	1.5082e+02	1.7418e+02

Table B.15: Results for high conditioning and unimodal functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{11}	(a)	3.2215e+02	3.1725e+02	1.4276e+03	2.1329e+02	3.8786e+02
f_{11}	(b)	3.1326e+02	3.1541e+02	1.1400e+03	2.5545e+02	3.8921e+02
f_{11}	(c)	3.2759e+02	3.2280e+02	1.1354e+03	2.4289e+02	4.0343e+02
f_{12}	(a)	1.0854e+07	1.0929e+07	3.3643e+12	6.9133e+06	1.5528e+07
f_{12}	(b)	1.6694e+03	1.6063e+03	3.6776e+05	5.1794e+02	4.0425e+03
f_{12}	(c)	1.9408e+04	2.0703e+04	1.6095e+08	2.9521e+03	5.2281e+04
f_{13}	(a)	7.1188e+02	7.1539e+02	2.3334e+03	6.1817e+02	8.4420e+02
f_{13}	(b)	5.8772e+01	6.0657e+01	1.4658e+02	4.2263e+01	9.7580e+01
f_{13}	(c)	7.8296e+01	8.0066e+01	1.8540e+02	5.2075e+01	1.2349e+02
f_{14}	(a)	-4.7694e+01	-4.7626e+01	5.8462e-01	-4.9042e+01	-4.4522e+01
f_{14}	(b)	-5.2325e+01	-5.2319e+01	4.4110e-04	-5.2335e+01	-5.2195e+01
f_{14}	(c)	-5.2285e+01	-5.2278e+01	7.4443e-04	-5.2317e+01	-5.2197e+01

Table B.16: Results for multi-modal functions with adequate global structure functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{15}	(a)	1.3588e+03	1.3577e+03	1.2054e+03	1.2762e+03	1.4324e+03
f_{15}	(b)	1.3395e+03	1.3498e+03	5.3244e+03	1.1972e+03	1.5347e+03
f_{15}	(c)	1.5122e+03	1.5196e+03	2.0084e+04	1.2611e+03	1.8056e+03
f_{16}	(a)	9.9758e+01	9.9588e+01	8.8764e+00	9.1164e+01	1.0440e+02
f_{16}	(b)	7.8169e+01	7.8423e+01	2.0677e+00	7.5732e+01	8.2017e+01
f_{16}	(c)	8.7413e+01	8.7680e+01	1.0818e+01	8.1666e+01	9.5598e+01
f_{17}	(a)	-1.0017e+01	-1.0080e+01	9.7077e-01	-1.2844e+01	-7.3656e+00
f_{17}	(b)	-1.0317e+01	-1.0282e+01	9.6818e-01	-1.2585e+01	-8.0868e+00
f_{17}	(c)	-6.8978e+00	-6.8390e+00	1.9649e+00	-9.4626e+00	-3.3984e+00
f_{18}	(a)	7.6749e+00	8.6135e+00	2.6358e+01	-1.8841e+00	2.2030e+01
f_{18}	(b)	9.0638e+00	9.5687e+00	1.8320e+01	1.3310e+00	1.9010e+01
f_{18}	(c)	2.0763e+01	2.0377e+01	4.2939e+01	5.9672e+00	3.3908e+01
f_{19}	(a)	-9.4513e+01	-9.4655e+01	3.4599e-01	-9.6208e+01	-9.3890e+01
f_{19}	(b)	-9.4321e+01	-9.4238e+01	4.1952e-01	-9.5408e+01	-9.2359e+01
f_{19}	(c)	-9.4142e+01	-9.4204e+01	3.5089e-01	-9.5768e+01	-9.3160e+01

Table B.17: Results for multi-modal functions with weak global structure functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{20}	(a)	-5.0856e+02	-4.9618e+02	2.3362e+03	-5.4304e+02	-3.3686e+02
f_{20}	(b)	-5.4487e+02	-5.4488e+02	1.8778e-02	-5.4520e+02	-5.4460e+02
f_{20}	(c)	-5.4484e+02	-5.4483e+02	4.8358e-02	-5.4525e+02	-5.4404e+02
f_{21}	(a)	5.2593e+01	5.3444e+01	1.2956e+01	4.7441e+01	6.3301e+01
f_{21}	(b)	4.2027e+01	4.1883e+01	9.4146e-01	4.0787e+01	4.4741e+01
f_{21}	(c)	4.1993e+01	4.1795e+01	9.3503e-01	4.0782e+01	4.4816e+01
f_{22}	(a)	-9.8492e+02	-9.8468e+02	2.2253e+01	-9.9276e+02	-9.7335e+02
f_{22}	(b)	-9.9804e+02	-9.9758e+02	6.2102e+00	-9.9989e+02	-9.8541e+02
f_{22}	(c)	-9.9834e+02	-9.9773e+02	7.6316e+00	-9.9927e+02	-9.8541e+02
f_{23}	(a)	1.0727e+01	1.0673e+01	2.8936e-01	9.3635e+00	1.1765e+01
f_{23}	(b)	8.2348e+00	8.2346e+00	5.1779e-02	7.7656e+00	8.6565e+00
f_{23}	(c)	9.4095e+00	9.5027e+00	3.8110e-01	8.2598e+00	1.0924e+01
f_{24}	(a)	4.3892e+02	4.3568e+02	6.5315e+02	3.6966e+02	4.9397e+02
f_{24}	(b)	4.0658e+02	4.0673e+02	2.3386e+03	3.2221e+02	5.3444e+02
f_{24}	(c)	5.2881e+02	5.3388e+02	2.7908e+03	4.2076e+02	6.6100e+02

Table B.18: Results for fractal functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{25}	(a)	0.0000e+00	4.0000e-02	3.9184e-02	0.0000e+00	1.0000e+00
f_{25}	(b)	0.0000e+00	2.6000e-01	1.9633e-01	0.0000e+00	1.0000e+00
f_{25}	(c)	0.0000e+00	2.2000e-01	1.7510e-01	0.0000e+00	1.0000e+00

Table B.19: Pairwise ranked Welch's t -test on algorithms for 30 dimensional separable functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{01}	(a)	(b)	17.15	2.88e-31
f_{01}	(a)	(c)	17.15	2.88e-31
f_{01}	(b)	(c)	-15.22	1.60e-27
f_{02}	(a)	(b)	17.15	2.88e-31
f_{02}	(a)	(c)	17.15	2.88e-31
f_{02}	(b)	(c)	2.48	1.48e-02
f_{03}	(a)	(b)	6.15	2.90e-08
f_{03}	(a)	(c)	-6.90	9.77e-10
f_{03}	(b)	(c)	-9.93	1.74e-16
f_{04}	(a)	(b)	-0.03	9.73e-01
f_{04}	(a)	(c)	-16.37	8.63e-30
f_{04}	(b)	(c)	-13.05	3.65e-23
f_{05}	(a)	(b)	7.53	2.78e-11
f_{05}	(a)	(c)	2.42	1.76e-02
f_{05}	(b)	(c)	-2.61	1.05e-02

Table B.20: Pairwise ranked Welch's t -test on algorithms for 30 dimensional low or moderate conditioning functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{06}	(a)	(b)	17.15	2.88e-31
f_{06}	(a)	(c)	17.15	2.88e-31
f_{06}	(b)	(c)	3.12	2.37e-03
f_{07}	(a)	(b)	1.63	1.07e-01
f_{07}	(a)	(c)	3.41	9.79e-04
f_{07}	(b)	(c)	1.29	2.01e-01
f_{08}	(a)	(b)	17.15	2.88e-31
f_{08}	(a)	(c)	17.15	2.88e-31
f_{08}	(b)	(c)	1.27	2.07e-01
f_{09}	(a)	(b)	17.15	2.88e-31
f_{09}	(a)	(c)	17.15	2.88e-31
f_{09}	(b)	(c)	-7.25	1.25e-10

Table B.21: Pairwise ranked Welch's t -test on algorithms for 30 dimensional high conditioning and unimodal functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{11}	(a)	(b)	0.59	5.56e-01
f_{11}	(a)	(c)	-0.50	6.17e-01
f_{11}	(b)	(c)	-1.16	2.49e-01
f_{12}	(a)	(b)	17.15	2.88e-31
f_{12}	(a)	(c)	17.15	2.88e-31
f_{12}	(b)	(c)	-16.99	5.81e-31
f_{13}	(a)	(b)	17.15	2.88e-31
f_{13}	(a)	(c)	17.15	2.88e-31
f_{13}	(b)	(c)	-8.36	4.74e-13
f_{14}	(a)	(b)	17.15	2.88e-31
f_{14}	(a)	(c)	17.15	2.88e-31
f_{14}	(b)	(c)	-11.76	2.53e-20

Table B.22: Pairwise ranked Welch's t -test on algorithms for 30 dimensional multi-modal functions with adequate global structure functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{15}	(a)	(b)	1.26	2.12e-01
f_{15}	(a)	(c)	-8.10	3.97e-12
f_{15}	(b)	(c)	-7.57	2.41e-11
f_{16}	(a)	(b)	17.15	2.88e-31
f_{16}	(a)	(c)	16.27	1.33e-29
f_{16}	(b)	(c)	-17.04	4.60e-31
f_{17}	(a)	(b)	1.10	2.75e-01
f_{17}	(a)	(c)	-15.10	2.56e-27
f_{17}	(b)	(c)	-15.48	4.45e-28
f_{18}	(a)	(b)	-1.02	3.10e-01
f_{18}	(a)	(c)	-10.60	6.75e-18
f_{18}	(b)	(c)	-9.70	7.74e-16
f_{19}	(a)	(b)	-2.64	9.63e-03
f_{19}	(a)	(c)	-3.87	1.96e-04
f_{19}	(b)	(c)	-1.06	2.91e-01

Table B.23: Pairwise ranked Welch's t -test on algorithms for 30 dimensional multi-modal functions with weak global structure functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{20}	(a)	(b)	17.15	2.88e-31
f_{20}	(a)	(c)	17.15	2.88e-31
f_{20}	(b)	(c)	-0.81	4.19e-01
f_{21}	(a)	(b)	17.15	2.88e-31
f_{21}	(a)	(c)	17.15	2.88e-31
f_{21}	(b)	(c)	0.67	5.07e-01
f_{22}	(a)	(b)	15.18	1.92e-27
f_{22}	(a)	(c)	15.02	4.00e-27
f_{22}	(b)	(c)	0.97	3.34e-01
f_{23}	(a)	(b)	17.15	2.88e-31
f_{23}	(a)	(c)	10.43	1.39e-17
f_{23}	(b)	(c)	-15.31	1.06e-27
f_{24}	(a)	(b)	3.72	3.45e-04
f_{24}	(a)	(c)	-13.67	2.30e-24
f_{24}	(b)	(c)	-13.40	6.96e-24

Table B.24: Pairwise ranked Welch's t -test on algorithms for 30 dimensional fractal functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{25}	(a)	(b)	-3.21	2.06e-03
f_{25}	(a)	(c)	-2.75	7.59e-03
f_{25}	(b)	(c)	0.46	6.44e-01

B.3 DE の詳細な結果

Table B.25: Results for separable functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{01}	(a)	7.9480e+01	7.9480e+01	0.0000e+00	7.9480e+01	7.9480e+01
f_{01}	(b)	7.9480e+01	7.9480e+01	1.5449e-10	7.9480e+01	7.9480e+01
f_{01}	(c)	7.9480e+01	7.9480e+01	2.3667e-09	7.9480e+01	7.9480e+01
f_{02}	(a)	1.2523e+05	1.3080e+05	1.7602e+09	4.5599e+04	2.7231e+05
f_{02}	(b)	3.3885e+04	3.4383e+04	1.6507e+08	1.2755e+04	6.6600e+04
f_{02}	(c)	1.8372e+04	1.8977e+04	3.3077e+07	5.8663e+03	3.0373e+04
f_{03}	(a)	-2.6580e+02	-2.6457e+02	1.7650e+02	-2.9141e+02	-2.1852e+02
f_{03}	(b)	-3.3317e+02	-3.2746e+02	8.3867e+02	-3.8542e+02	-2.8173e+02
f_{03}	(c)	-3.0239e+02	-3.1038e+02	1.2197e+03	-4.4094e+02	-2.6727e+02
f_{04}	(a)	-2.3980e+02	-2.3941e+02	2.6668e+02	-2.7243e+02	-1.9123e+02
f_{04}	(b)	-3.2312e+02	-3.2201e+02	7.8196e+02	-3.7995e+02	-2.5048e+02
f_{04}	(c)	-3.1760e+02	-3.2512e+02	2.6364e+03	-4.1006e+02	-2.4298e+02
f_{05}	(a)	5.5031e+01	5.9465e+01	3.1765e+02	3.1460e+01	1.0780e+02
f_{05}	(b)	8.7164e+01	8.7091e+01	6.6630e+01	7.0339e+01	1.0540e+02
f_{05}	(c)	4.6829e+01	4.8883e+01	6.6521e+01	3.0629e+01	6.7845e+01

Table B.26: Results for low or moderate conditioning functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{06}	(a)	4.4010e+01	4.5962e+01	4.3033e+01	3.7995e+01	6.3154e+01
f_{06}	(b)	9.8126e+01	9.9749e+01	4.1153e+02	6.6704e+01	1.5406e+02
f_{06}	(c)	3.8762e+01	3.8941e+01	1.0979e+00	3.7297e+01	4.1890e+01
f_{07}	(a)	9.9993e+01	9.9921e+01	6.6263e+00	9.5279e+01	1.0532e+02
f_{07}	(b)	1.0253e+02	1.0349e+02	1.5944e+01	9.6016e+01	1.1490e+02
f_{07}	(c)	9.7642e+01	9.7756e+01	2.8700e+00	9.4534e+01	1.0179e+02
f_{08}	(a)	2.3112e+02	2.2033e+02	1.3316e+03	1.7591e+02	3.0762e+02
f_{08}	(b)	1.9435e+02	2.0864e+02	1.1713e+03	1.7615e+02	2.8016e+02
f_{08}	(c)	1.7698e+02	1.7838e+02	7.9678e+01	1.7624e+02	2.4007e+02
f_{09}	(a)	1.5248e+02	1.5429e+02	9.6789e+01	1.5025e+02	2.1096e+02
f_{09}	(b)	1.5271e+02	1.6171e+02	8.8494e+02	1.4905e+02	3.1065e+02
f_{09}	(c)	1.5175e+02	1.5168e+02	4.7836e-01	1.4962e+02	1.5326e+02

Table B.27: Results for high conditioning and unimodal functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{11}	(a)	2.1753e+02	2.1629e+02	4.5013e+02	1.6972e+02	2.6568e+02
f_{11}	(b)	1.6988e+02	1.7089e+02	2.2619e+02	1.3687e+02	2.0401e+02
f_{11}	(c)	1.7341e+02	1.7368e+02	2.0806e+02	1.3614e+02	2.0994e+02
f_{12}	(a)	-6.1683e+02	-6.1517e+02	2.3249e+01	-6.2085e+02	-5.9720e+02
f_{12}	(b)	-6.1088e+02	-5.8275e+02	7.5075e+03	-6.1890e+02	-2.1048e+02
f_{12}	(c)	-4.4197e+02	-4.3429e+02	7.8339e+03	-5.6153e+02	-7.1810e+01
f_{13}	(a)	4.1560e+01	4.3145e+01	7.7892e+01	3.1337e+01	6.8330e+01
f_{13}	(b)	7.3694e+01	7.6994e+01	4.0085e+02	4.1911e+01	1.3316e+02
f_{13}	(c)	3.3539e+01	3.3872e+01	1.0303e+00	3.2337e+01	3.7363e+01
f_{14}	(a)	-5.2347e+01	-5.2347e+01	6.6374e-07	-5.2348e+01	-5.2344e+01
f_{14}	(b)	-5.2339e+01	-5.2338e+01	5.5850e-05	-5.2346e+01	-5.2301e+01
f_{14}	(c)	-5.2345e+01	-5.2345e+01	6.0995e-07	-5.2347e+01	-5.2344e+01

Table B.28: Results for multi-modal functions with adequate global structure functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{15}	(a)	1.2061e+03	1.2040e+03	2.1050e+02	1.1686e+03	1.2314e+03
f_{15}	(b)	1.1625e+03	1.1583e+03	7.3209e+02	1.0921e+03	1.2021e+03
f_{15}	(c)	1.1775e+03	1.1744e+03	2.6439e+02	1.1320e+03	1.2012e+03
f_{16}	(a)	9.9624e+01	9.9687e+01	5.7136e+00	9.0761e+01	1.0462e+02
f_{16}	(b)	9.9205e+01	9.9000e+01	7.6377e+00	9.1657e+01	1.0321e+02
f_{16}	(c)	9.9292e+01	9.8551e+01	6.7885e+00	8.9981e+01	1.0200e+02
f_{17}	(a)	-1.6908e+01	-1.6896e+01	1.9763e-03	-1.6938e+01	-1.6717e+01
f_{17}	(b)	-1.6837e+01	-1.6807e+01	7.7920e-03	-1.6908e+01	-1.6545e+01
f_{17}	(c)	-1.6922e+01	-1.6918e+01	1.4256e-04	-1.6929e+01	-1.6864e+01
f_{18}	(a)	-1.6629e+01	-1.6463e+01	1.7245e-01	-1.6881e+01	-1.4947e+01
f_{18}	(b)	-1.6393e+01	-1.6302e+01	9.8935e-02	-1.6732e+01	-1.5268e+01
f_{18}	(c)	-1.6669e+01	-1.6635e+01	2.6416e-02	-1.6843e+01	-1.6194e+01
f_{19}	(a)	-9.7013e+01	-9.7045e+01	8.5013e-02	-9.7872e+01	-9.6592e+01
f_{19}	(b)	-9.7323e+01	-9.7345e+01	1.4839e-01	-9.8094e+01	-9.6488e+01
f_{19}	(c)	-9.7182e+01	-9.7163e+01	1.2861e-01	-9.7970e+01	-9.6551e+01

Table B.29: Results for multi-modal functions with weak global structure functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{20}	(a)	-5.4344e+02	-5.4344e+02	1.0739e-02	-5.4372e+02	-5.4326e+02
f_{20}	(b)	-5.4352e+02	-5.4355e+02	4.7841e-02	-5.4423e+02	-5.4327e+02
f_{20}	(c)	-5.4338e+02	-5.4341e+02	1.1026e-02	-5.4363e+02	-5.4320e+02
f_{21}	(a)	4.5337e+01	4.7835e+01	4.5108e+01	4.0780e+01	6.0026e+01
f_{21}	(b)	4.6694e+01	4.7949e+01	4.4045e+01	4.0780e+01	5.9865e+01
f_{21}	(c)	4.1184e+01	4.3657e+01	1.9227e+01	4.0780e+01	5.7892e+01
f_{22}	(a)	-9.8542e+02	-9.8567e+02	3.1786e+00	-9.9802e+02	-9.8542e+02
f_{22}	(b)	-9.8542e+02	-9.8578e+02	3.6285e+00	-9.9730e+02	-9.8538e+02
f_{22}	(c)	-9.8542e+02	-9.8586e+02	4.8362e+00	-9.9722e+02	-9.8541e+02
f_{23}	(a)	1.0220e+01	1.0189e+01	1.3043e-01	9.3267e+00	1.1039e+01
f_{23}	(b)	1.0156e+01	1.0158e+01	1.4024e-01	9.4311e+00	1.0982e+01
f_{23}	(c)	1.0301e+01	1.0166e+01	1.7207e-01	8.7501e+00	1.0769e+01
f_{24}	(a)	3.1689e+02	3.1689e+02	2.5636e+02	2.6333e+02	3.5162e+02
f_{24}	(b)	3.0537e+02	3.0683e+02	1.5986e+02	2.7305e+02	3.2960e+02
f_{24}	(c)	3.1114e+02	3.0971e+02	1.3117e+02	2.7875e+02	3.3027e+02

Table B.30: Results for fractal functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{25}	(a)	-5.0000e+00	-5.4600e+00	1.3886e+01	-1.8000e+01	-1.0000e+00
f_{25}	(b)	-3.0000e+00	-2.8400e+00	1.0759e+00	-5.0000e+00	-1.0000e+00
f_{25}	(c)	-6.0000e+00	-5.8600e+00	1.2286e-01	-6.0000e+00	-5.0000e+00

Table B.31: Pairwise ranked Welch's t -test on algorithms for 30 dimensional separable functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{01}	(a)	(b)	-24.25	5.97e-29
f_{01}	(a)	(c)	-24.25	5.97e-29
f_{01}	(b)	(c)	-17.15	2.88e-31
f_{02}	(a)	(b)	16.52	4.47e-30
f_{02}	(a)	(c)	17.15	2.88e-31
f_{02}	(b)	(c)	7.80	9.80e-12
f_{03}	(a)	(b)	16.52	4.46e-30
f_{03}	(a)	(c)	14.02	3.76e-25
f_{03}	(b)	(c)	-3.26	1.51e-03
f_{04}	(a)	(b)	16.22	1.66e-29
f_{04}	(a)	(c)	14.24	1.37e-25
f_{04}	(b)	(c)	0.12	9.07e-01
f_{05}	(a)	(b)	-9.10	2.93e-14
f_{05}	(a)	(c)	3.33	1.24e-03
f_{05}	(b)	(c)	17.15	2.88e-31

Table B.32: Pairwise ranked Welch's t -test on algorithms for 30 dimensional low or moderate conditioning functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{06}	(a)	(b)	-17.15	2.88e-31
f_{06}	(a)	(c)	10.05	1.53e-16
f_{06}	(b)	(c)	17.15	2.88e-31
f_{07}	(a)	(b)	-5.26	8.59e-07
f_{07}	(a)	(c)	4.63	1.17e-05
f_{07}	(b)	(c)	11.17	4.07e-19
f_{08}	(a)	(b)	0.90	3.69e-01
f_{08}	(a)	(c)	8.39	6.20e-13
f_{08}	(b)	(c)	9.44	2.33e-15
f_{09}	(a)	(b)	-1.45	1.51e-01
f_{09}	(a)	(c)	4.17	6.89e-05
f_{09}	(b)	(c)	5.90	6.10e-08

Table B.33: Pairwise ranked Welch's t -test on algorithms for 30 dimensional high conditioning and unimodal functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{11}	(a)	(b)	13.31	1.18e-23
f_{11}	(a)	(c)	12.23	2.12e-21
f_{11}	(b)	(c)	-0.93	3.55e-01
f_{12}	(a)	(b)	-4.96	2.92e-06
f_{12}	(a)	(c)	-17.15	2.88e-31
f_{12}	(b)	(c)	-11.14	1.91e-18
f_{13}	(a)	(b)	-12.23	2.04e-21
f_{13}	(a)	(c)	8.59	3.78e-13
f_{13}	(b)	(c)	17.15	2.88e-31
f_{14}	(a)	(b)	-16.37	8.63e-30
f_{14}	(a)	(c)	-12.29	2.76e-21
f_{14}	(b)	(c)	10.69	7.20e-18

Table B.34: Pairwise ranked Welch's t -test on algorithms for 30 dimensional multi-modal functions with adequate global structure functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{15}	(a)	(b)	11.94	8.18e-21
f_{15}	(a)	(c)	10.60	6.56e-18
f_{15}	(b)	(c)	-3.30	1.38e-03
f_{16}	(a)	(b)	1.08	2.81e-01
f_{16}	(a)	(c)	2.05	4.28e-02
f_{16}	(b)	(c)	0.83	4.11e-01
f_{17}	(a)	(b)	-7.82	6.23e-12
f_{17}	(a)	(c)	1.86	6.63e-02
f_{17}	(b)	(c)	15.14	2.12e-27
f_{18}	(a)	(b)	-3.97	1.47e-04
f_{18}	(a)	(c)	1.76	8.23e-02
f_{18}	(b)	(c)	7.66	1.39e-11
f_{19}	(a)	(b)	4.33	3.67e-05
f_{19}	(a)	(c)	1.81	7.30e-02
f_{19}	(b)	(c)	-2.53	1.31e-02

Table B.35: Pairwise ranked Welch's t -test on algorithms for 30 dimensional multi-modal functions with weak global structure functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{20}	(a)	(b)	2.79	6.31e-03
f_{20}	(a)	(c)	-1.83	6.96e-02
f_{20}	(b)	(c)	-4.14	7.42e-05
f_{21}	(a)	(b)	-0.73	4.68e-01
f_{21}	(a)	(c)	4.46	2.18e-05
f_{21}	(b)	(c)	4.83	5.07e-06
f_{22}	(a)	(b)	-15.52	2.38e-22
f_{22}	(a)	(c)	-15.52	2.38e-22
f_{22}	(b)	(c)	-4.05	1.14e-04
f_{23}	(a)	(b)	0.45	6.51e-01
f_{23}	(a)	(c)	-0.22	8.27e-01
f_{23}	(b)	(c)	-0.43	6.71e-01
f_{24}	(a)	(b)	3.79	2.61e-04
f_{24}	(a)	(c)	2.66	9.27e-03
f_{24}	(b)	(c)	-1.37	1.74e-01

Table B.36: Pairwise ranked Welch's t -test on algorithms for 30 dimensional fractal functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{25}	(a)	(b)	-4.54	1.75e-05
f_{25}	(a)	(c)	2.91	5.12e-03
f_{25}	(b)	(c)	20.76	1.15e-34

B.4 FA の詳細な結果

Table B.37: Results for separable functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{01}	(a)	7.9480e+01	7.9480e+01	0.0000e+00	7.9480e+01	7.9480e+01
f_{01}	(b)	7.9480e+01	7.9480e+01	0.0000e+00	7.9480e+01	7.9480e+01
f_{01}	(c)	7.9486e+01	7.9486e+01	7.8366e-07	7.9484e+01	7.9488e+01
f_{02}	(a)	2.0610e+04	2.6657e+04	1.1248e+09	5.6903e+03	2.3692e+05
f_{02}	(b)	1.5597e+04	1.6387e+04	3.1532e+07	9.1820e+03	3.9947e+04
f_{02}	(c)	1.7163e+04	1.7161e+04	3.5987e+07	6.5460e+03	3.4681e+04
f_{03}	(a)	-3.8497e+02	-3.7622e+02	1.1368e+03	-4.1133e+02	-1.9843e+02
f_{03}	(b)	-4.0657e+02	-4.0570e+02	1.3507e+02	-4.2800e+02	-3.7349e+02
f_{03}	(c)	-3.8545e+02	-3.8443e+02	3.2353e+02	-4.1844e+02	-3.2968e+02
f_{04}	(a)	-3.4866e+02	-3.3484e+02	2.6797e+03	-3.8249e+02	-3.6252e+01
f_{04}	(b)	-3.8276e+02	-3.7826e+02	2.4309e+02	-4.0111e+02	-3.3267e+02
f_{04}	(c)	-3.4533e+02	-3.4397e+02	7.1811e+02	-3.9398e+02	-2.8047e+02
f_{05}	(a)	1.1275e+02	1.2032e+02	1.6671e+03	6.5022e+01	2.5274e+02
f_{05}	(b)	7.1046e+01	7.3069e+01	3.9226e+02	3.2981e+01	1.1757e+02
f_{05}	(c)	9.8427e+01	1.0139e+02	4.9317e+02	5.6324e+01	1.5941e+02

Table B.38: Results for low or moderate conditioning functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{06}	(a)	5.8611e+01	7.5574e+01	4.7326e+03	3.9350e+01	4.3878e+02
f_{06}	(b)	3.8603e+01	4.0140e+01	1.4217e+01	3.6630e+01	4.9903e+01
f_{06}	(c)	3.7270e+01	3.7378e+01	4.1253e-01	3.6437e+01	4.0058e+01
f_{07}	(a)	1.2241e+02	1.2493e+02	2.6440e+02	1.0677e+02	2.1293e+02
f_{07}	(b)	1.1037e+02	1.1180e+02	3.8563e+01	1.0256e+02	1.3435e+02
f_{07}	(c)	1.1534e+02	1.1742e+02	6.2305e+01	1.0140e+02	1.3560e+02
f_{08}	(a)	1.9791e+02	2.1943e+02	1.0154e+03	1.8491e+02	3.0472e+02
f_{08}	(b)	1.9652e+02	2.0796e+02	6.9696e+02	1.9245e+02	3.1084e+02
f_{08}	(c)	1.9722e+02	2.0763e+02	5.3656e+02	1.8868e+02	2.6262e+02
f_{09}	(a)	1.7123e+02	1.8044e+02	9.0153e+02	1.6830e+02	3.1243e+02
f_{09}	(b)	1.7145e+02	1.7403e+02	1.5202e+02	1.6722e+02	2.3311e+02
f_{09}	(c)	1.7239e+02	1.7975e+02	5.8762e+02	1.6824e+02	2.9743e+02

Table B.39: Results for high conditioning and unimodal functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{11}	(a)	4.0811e+02	4.0592e+02	3.1455e+03	2.4000e+02	5.4561e+02
f_{11}	(b)	3.6848e+02	3.6951e+02	3.0242e+03	2.6049e+02	5.0097e+02
f_{11}	(c)	4.5228e+02	4.5827e+02	3.7945e+03	3.0922e+02	6.2831e+02
f_{12}	(a)	-6.2029e+02	1.6490e+06	1.2240e+14	-6.2098e+02	7.8202e+07
f_{12}	(b)	-6.1827e+02	-6.1807e+02	2.0257e+00	-6.2024e+02	-6.1421e+02
f_{12}	(c)	4.4844e+03	4.5458e+03	4.4641e+05	3.2516e+03	6.4795e+03
f_{13}	(a)	3.0892e+01	3.1588e+01	3.1777e+00	3.0063e+01	3.7581e+01
f_{13}	(b)	3.1682e+01	3.2016e+01	1.3412e+00	3.0546e+01	3.6530e+01
f_{13}	(c)	4.4460e+01	4.5052e+01	5.3243e+00	4.2221e+01	5.3597e+01
f_{14}	(a)	-5.2346e+01	-5.2247e+01	3.1313e-01	-5.2347e+01	-4.8523e+01
f_{14}	(b)	-5.2342e+01	-5.2342e+01	8.3288e-07	-5.2344e+01	-5.2340e+01
f_{14}	(c)	-5.2340e+01	-5.2340e+01	2.4946e-06	-5.2343e+01	-5.2334e+01

Table B.40: Results for multi-modal functions with adequate global structure functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{15}	(a)	1.2122e+03	1.2172e+03	4.0908e+03	1.0987e+03	1.4004e+03
f_{15}	(b)	1.1872e+03	1.1760e+03	1.7294e+03	1.0518e+03	1.2309e+03
f_{15}	(c)	1.1655e+03	1.1683e+03	1.9420e+03	1.0875e+03	1.2889e+03
f_{16}	(a)	9.3670e+01	9.3232e+01	2.3606e+01	8.2431e+01	1.0312e+02
f_{16}	(b)	9.0020e+01	9.0305e+01	2.1005e+01	7.8500e+01	1.0088e+02
f_{16}	(c)	1.0282e+02	1.0237e+02	1.4965e+01	9.0333e+01	1.1056e+02
f_{17}	(a)	-1.6168e+01	-1.6008e+01	7.7356e-01	-1.6928e+01	-1.3473e+01
f_{17}	(b)	-1.6891e+01	-1.6852e+01	1.8224e-02	-1.6926e+01	-1.6139e+01
f_{17}	(c)	-1.5549e+01	-1.4981e+01	2.4160e+00	-1.6905e+01	-1.1626e+01
f_{18}	(a)	-1.2676e+01	-1.2130e+01	1.2213e+01	-1.6484e+01	2.1370e-01
f_{18}	(b)	-1.6090e+01	-1.5971e+01	4.3694e-01	-1.6775e+01	-1.3520e+01
f_{18}	(c)	-1.2013e+01	-1.1175e+01	1.4857e+01	-1.6328e+01	1.6174e+00
f_{19}	(a)	-9.5789e+01	-9.5790e+01	9.6333e-02	-9.6507e+01	-9.5188e+01
f_{19}	(b)	-9.6760e+01	-9.6831e+01	3.5328e-01	-9.8635e+01	-9.5434e+01
f_{19}	(c)	-9.8153e+01	-9.8102e+01	4.4255e-01	-9.9525e+01	-9.6588e+01

Table B.41: Results for multi-modal functions with weak global structure functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{20}	(a)	-5.4500e+02	-5.4500e+02	2.0485e-02	-5.4536e+02	-5.4472e+02
f_{20}	(b)	-5.4507e+02	-5.4504e+02	2.9889e-02	-5.4539e+02	-5.4462e+02
f_{20}	(c)	-5.4506e+02	-5.4507e+02	2.0168e-02	-5.4541e+02	-5.4480e+02
f_{21}	(a)	4.3253e+01	4.3642e+01	4.6631e+00	4.0780e+01	5.3635e+01
f_{21}	(b)	4.3253e+01	4.4368e+01	1.1166e+01	4.2025e+01	5.7180e+01
f_{21}	(c)	4.3255e+01	4.4025e+01	7.5027e+00	4.0780e+01	5.3637e+01
f_{22}	(a)	-9.8541e+02	-9.8981e+02	7.1596e+01	-9.9931e+02	-9.6022e+02
f_{22}	(b)	-9.8541e+02	-9.9090e+02	4.3982e+01	-9.9931e+02	-9.7822e+02
f_{22}	(c)	-9.8541e+02	-9.9024e+02	3.8801e+01	-9.9931e+02	-9.8541e+02
f_{23}	(a)	9.5384e+00	9.6223e+00	6.9995e-01	8.2932e+00	1.1657e+01
f_{23}	(b)	9.4701e+00	9.5439e+00	4.7553e-01	8.4764e+00	1.1399e+01
f_{23}	(c)	8.9536e+00	8.9562e+00	3.6719e-01	7.5042e+00	1.0898e+01
f_{24}	(a)	5.3010e+02	5.2284e+02	1.0548e+03	4.4199e+02	5.7217e+02
f_{24}	(b)	3.9517e+02	3.8564e+02	1.1019e+03	3.1713e+02	4.4475e+02
f_{24}	(c)	3.0802e+02	3.0784e+02	8.1095e+02	2.2435e+02	3.5014e+02

Table B.42: Results for fractal functions with 20,000 function calls

$E(x)$	alg	median	mean	std	min	max
f_{25}	(a)	-5.0000e+00	-4.5000e+00	1.1929e+01	-9.0000e+00	-0.0000e+00
f_{25}	(b)	-5.0000e+00	-4.4600e+00	1.8045e+00	-6.0000e+00	-0.0000e+00
f_{25}	(c)	-1.0000e+00	-2.3600e+00	5.9494e+00	-5.0000e+00	-0.0000e+00

Table B.43: Pairwise ranked Welch's t -test on algorithms for 30 dimensional separable functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{01}	(a)	(b)	nan	nan
f_{01}	(a)	(c)	-24.28	5.70e-29
f_{01}	(b)	(c)	-24.28	5.70e-29
f_{02}	(a)	(b)	3.71	3.44e-04
f_{02}	(a)	(c)	2.66	9.05e-03
f_{02}	(b)	(c)	-0.92	3.58e-01
f_{03}	(a)	(b)	9.44	2.02e-15
f_{03}	(a)	(c)	0.83	4.11e-01
f_{03}	(b)	(c)	-7.19	1.36e-10
f_{04}	(a)	(b)	9.53	1.32e-15
f_{04}	(a)	(c)	0.38	7.02e-01
f_{04}	(b)	(c)	-8.34	4.86e-13
f_{05}	(a)	(b)	8.88	3.21e-14
f_{05}	(a)	(c)	2.51	1.37e-02
f_{05}	(b)	(c)	-7.11	1.90e-10

Table B.44: Pairwise ranked Welch's t -test on algorithms for 30 dimensional low or moderate conditioning functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{06}	(a)	(b)	11.60	4.24e-20
f_{06}	(a)	(c)	16.99	5.81e-31
f_{06}	(b)	(c)	5.01	2.71e-06
f_{07}	(a)	(b)	7.09	2.19e-10
f_{07}	(a)	(c)	3.02	3.24e-03
f_{07}	(b)	(c)	-4.07	9.63e-05
f_{08}	(a)	(b)	1.26	2.10e-01
f_{08}	(a)	(c)	0.34	7.37e-01
f_{08}	(b)	(c)	-1.63	1.07e-01
f_{09}	(a)	(b)	-0.59	5.56e-01
f_{09}	(a)	(c)	-4.02	1.17e-04
f_{09}	(b)	(c)	-4.24	5.11e-05

Table B.45: Pairwise ranked Welch's t -test on algorithms for 30 dimensional high conditioning and unimodal functions with 20,000 function calls

$E(x)$	alg1	alg2	t-value	P -value
f_{11}	(a)	(b)	3.41	9.47e-04
f_{11}	(a)	(c)	-4.64	1.06e-05
f_{11}	(b)	(c)	-7.98	2.79e-12
f_{12}	(a)	(b)	-5.17	1.76e-06
f_{12}	(a)	(c)	-13.05	2.14e-22
f_{12}	(b)	(c)	-17.15	2.87e-31
f_{13}	(a)	(b)	-3.18	2.08e-03
f_{13}	(a)	(c)	-17.15	2.88e-31
f_{13}	(b)	(c)	-17.15	2.88e-31
f_{14}	(a)	(b)	-10.52	9.14e-17
f_{14}	(a)	(c)	-11.28	1.08e-18
f_{14}	(b)	(c)	-8.11	1.92e-12

Table B.46: Pairwise ranked Welch's t -test on algorithms for 30 dimensional multi-modal functions with adequate global structure functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{15}	(a)	(b)	3.69	3.83e-04
f_{15}	(a)	(c)	4.58	1.40e-05
f_{15}	(b)	(c)	1.40	1.65e-01
f_{16}	(a)	(b)	3.20	1.85e-03
f_{16}	(a)	(c)	-10.85	1.79e-18
f_{16}	(b)	(c)	-13.91	6.17e-25
f_{17}	(a)	(b)	9.11	1.36e-14
f_{17}	(a)	(c)	-3.90	1.80e-04
f_{17}	(b)	(c)	-12.54	6.28e-22
f_{18}	(a)	(b)	9.07	1.81e-14
f_{18}	(a)	(c)	-1.36	1.76e-01
f_{18}	(b)	(c)	-13.15	2.64e-23
f_{19}	(a)	(b)	12.46	8.52e-22
f_{19}	(a)	(c)	17.15	2.88e-31
f_{19}	(b)	(c)	10.21	4.35e-17

Table B.47: Pairwise ranked Welch's t -test on algorithms for 30 dimensional multi-modal functions with weak global structure functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{20}	(a)	(b)	1.41	1.63e-01
f_{20}	(a)	(c)	2.26	2.61e-02
f_{20}	(b)	(c)	0.69	4.89e-01
f_{21}	(a)	(b)	-0.22	8.23e-01
f_{21}	(a)	(c)	-2.44	1.67e-02
f_{21}	(b)	(c)	-2.03	4.47e-02
f_{22}	(a)	(b)	0.47	6.43e-01
f_{22}	(a)	(c)	-3.77	2.98e-04
f_{22}	(b)	(c)	-4.27	5.03e-05
f_{23}	(a)	(b)	0.35	7.27e-01
f_{23}	(a)	(c)	4.63	1.12e-05
f_{23}	(b)	(c)	4.59	1.33e-05
f_{24}	(a)	(b)	17.04	4.60e-31
f_{24}	(a)	(c)	17.15	2.88e-31
f_{24}	(b)	(c)	12.72	1.91e-22

Table B.48: Pairwise ranked Welch's t -test on algorithms for 30 dimensional fractal functions with 20,000 function calls

$E(\mathbf{x})$	alg1	alg2	t-value	P -value
f_{25}	(a)	(b)	-0.29	7.74e-01
f_{25}	(a)	(c)	-3.44	8.80e-04
f_{25}	(b)	(c)	-4.16	6.96e-05

B.5 横断的な比較の詳細な結果

Table B.49: Results for f01 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	7.9480e+01	7.9480e+01	5.0639e-09	7.9480e+01	7.9480e+01
Approximated ES	7.9482e+01	7.9482e+01	1.6003e-07	7.9481e+01	7.9483e+01
Approximated FA	7.9480e+01	7.9480e+01	9.1837e-10	7.9480e+01	7.9480e+01
Approximated PSO	7.9480e+01	7.9480e+01	8.2818e-06	7.9480e+01	7.9500e+01
CMA-ES	2.6646e+02	2.4766e+02	3.6341e+03	7.9480e+01	3.0579e+02
FA	7.9482e+01	7.9482e+01	2.3596e-08	7.9481e+01	7.9482e+01
jDE	7.9480e+01	7.9480e+01	2.0000e-10	7.9480e+01	7.9480e+01
NDT-PSO	8.6517e+01	8.7105e+01	2.2408e+01	7.9595e+01	9.9233e+01

Table B.50: Results for f02 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	1.9344e+04	1.9453e+04	3.6522e+07	8.0413e+03	3.1907e+04
Approximated ES	5.9915e+04	5.9051e+04	3.6532e+08	1.8344e+04	1.0448e+05
Approximated FA	1.5881e+04	1.6252e+04	3.3441e+07	6.8359e+03	3.1255e+04
Approximated PSO	-2.0980e+02	-2.0784e+02	9.3345e+01	-2.0988e+02	-1.4160e+02
CMA-ES	5.2595e+04	1.8331e+06	4.3267e+12	7.5864e+03	6.8817e+06
FA	5.3782e+04	5.4552e+04	1.6862e+08	3.2051e+04	9.8853e+04
jDE	-2.0988e+02	-2.0987e+02	4.3896e-04	-2.0988e+02	-2.0977e+02
NDT-PSO	4.9315e+04	8.4539e+04	1.0801e+10	1.1279e+03	5.0844e+05

Table B.51: Results for f03 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-3.0327e+02	-3.0561e+02	7.2566e+02	-3.9622e+02	-2.6787e+02
Approximated ES	-1.7105e+02	-1.8422e+02	2.6844e+03	-3.3746e+02	-7.3630e+01
Approximated FA	-3.9630e+02	-3.9349e+02	2.6446e+02	-4.2017e+02	-3.3456e+02
Approximated PSO	-3.6667e+02	-3.6030e+02	1.0504e+03	-4.1533e+02	-3.0178e+02
CMA-ES	6.5917e+02	5.4367e+02	1.3429e+05	-4.0497e+02	9.2424e+02
FA	-3.4112e+02	-3.4410e+02	6.8399e+02	-3.9764e+02	-2.7930e+02
jDE	-3.3451e+02	-3.3317e+02	5.1657e+02	-3.8016e+02	-2.6394e+02
NDT-PSO	-3.6038e+02	-3.5531e+02	1.0096e+03	-4.1083e+02	-2.7837e+02

Table B.52: Results for f04 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-3.2418e+02	-3.3120e+02	2.4885e+03	-4.2127e+02	-2.4435e+02
Approximated ES	-7.3805e+00	-1.7011e+01	5.9701e+03	-1.9744e+02	1.7491e+02
Approximated FA	-3.6488e+02	-3.6411e+02	3.2208e+02	-3.9916e+02	-3.2169e+02
Approximated PSO	-3.3863e+02	-3.3125e+02	2.4254e+03	-4.0978e+02	-2.0322e+02
CMA-ES	1.1267e+03	8.8036e+02	4.6273e+05	-2.9538e+02	1.5990e+03
FA	-2.8401e+02	-2.8052e+02	1.0555e+03	-3.3447e+02	-1.7795e+02
jDE	-3.3200e+02	-3.3007e+02	4.8860e+02	-3.8152e+02	-2.8882e+02
NDT-PSO	-3.4406e+02	-3.4473e+02	9.0262e+02	-4.0355e+02	-2.8770e+02

Table B.53: Results for f05 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	4.8875e+01	5.0075e+01	9.1408e+01	2.7363e+01	7.2098e+01
Approximated ES	1.8469e+02	1.8415e+02	6.0928e+02	1.3743e+02	2.3179e+02
Approximated FA	8.3874e+01	9.0056e+01	5.3919e+02	5.0146e+01	1.4045e+02
Approximated PSO	4.4760e+01	4.3919e+01	4.4949e+02	6.5630e+00	1.0058e+02
CMA-ES	1.0995e+15	2.5315e+17	5.5314e+35	-9.2100e+00	3.9920e+18
FA	2.0254e+02	2.0386e+02	9.3195e+02	1.2265e+02	2.9562e+02
jDE	-2.4690e+00	-1.8662e+00	9.3851e+00	-6.1901e+00	6.1505e+00
NDT-PSO	3.0252e+02	2.9865e+02	5.8898e+02	2.3705e+02	3.5041e+02

Table B.54: Results for f06 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	3.8650e+01	3.9000e+01	1.5043e+00	3.7090e+01	4.2507e+01
Approximated ES	1.5002e+02	1.5273e+02	1.4420e+03	7.3743e+01	2.4035e+02
Approximated FA	3.7583e+01	3.9250e+01	2.3070e+01	3.5979e+01	5.8593e+01
Approximated PSO	1.3535e+02	1.3175e+02	2.8264e+03	5.0717e+01	2.5437e+02
CMA-ES	9.9959e+04	1.5709e+05	2.3993e+10	9.4080e+03	5.6663e+05
FA	5.5783e+01	5.7498e+01	1.5828e+02	3.8349e+01	9.5380e+01
jDE	1.0203e+02	1.1054e+02	1.0545e+03	4.7844e+01	2.4579e+02
NDT-PSO	3.6181e+04	4.4523e+04	9.3908e+08	1.5033e+02	1.2048e+05

Table B.55: Results for f07 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	9.7392e+01	9.7664e+01	4.2102e+00	9.4502e+01	1.0260e+02
Approximated ES	1.8742e+02	1.9028e+02	1.1262e+03	1.3111e+02	2.9526e+02
Approximated FA	1.2034e+02	1.2134e+02	6.1427e+01	1.0820e+02	1.4732e+02
Approximated PSO	1.4068e+02	1.4442e+02	4.8268e+02	1.1705e+02	1.9731e+02
CMA-ES	1.2046e+03	9.7840e+02	3.8721e+05	9.4601e+01	1.8581e+03
FA	1.1685e+02	1.1793e+02	5.8432e+01	1.0420e+02	1.3355e+02
jDE	1.2912e+02	1.2865e+02	9.7579e+01	1.0655e+02	1.5439e+02
NDT-PSO	2.0841e+02	2.2561e+02	6.7516e+03	1.2311e+02	5.8554e+02

Table B.56: Results for f08 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	1.7699e+02	1.7820e+02	7.6758e+01	1.7573e+02	2.3876e+02
Approximated ES	1.9255e+02	2.0388e+02	8.5557e+02	1.7573e+02	2.9264e+02
Approximated FA	1.9644e+02	2.0479e+02	4.1678e+02	1.9299e+02	2.7353e+02
Approximated PSO	2.3068e+02	2.3413e+02	2.5643e+03	1.5237e+02	3.6871e+02
CMA-ES	2.3073e+05	2.2096e+05	3.4401e+09	2.8268e+02	3.2566e+05
FA	1.9725e+02	2.0478e+02	6.2372e+02	1.9288e+02	3.2323e+02
jDE	2.5279e+02	2.6748e+02	1.5175e+03	2.0383e+02	4.1313e+02
NDT-PSO	2.5220e+02	2.9688e+02	2.6547e+04	1.8123e+02	1.2865e+03

Table B.57: Results for f09 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	1.5161e+02	1.5309e+02	1.0151e+02	1.5078e+02	2.2280e+02
Approximated ES	1.5283e+02	1.5408e+02	5.0761e+01	1.4571e+02	1.9345e+02
Approximated FA	1.7137e+02	1.7645e+02	2.9459e+02	1.6945e+02	2.3467e+02
Approximated PSO	1.6406e+02	1.9674e+02	4.0347e+03	1.4445e+02	4.4733e+02
CMA-ES	1.5425e+05	1.4795e+05	1.4692e+09	1.5294e+02	2.0766e+05
FA	1.7170e+02	1.7435e+02	1.7920e+02	1.6957e+02	2.4784e+02
jDE	1.7274e+02	1.9017e+02	9.6923e+02	1.6870e+02	2.8363e+02
NDT-PSO	1.5601e+02	1.7578e+02	1.4791e+03	1.5048e+02	2.9599e+02

Table B.58: Results for f10 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	1.3884e+04	1.4844e+04	2.2247e+07	6.8212e+03	2.6188e+04
Approximated ES	5.3171e+04	5.2466e+04	3.2316e+08	1.8023e+04	9.6283e+04
Approximated FA	5.3398e+04	5.5161e+04	2.6086e+08	1.8677e+04	9.4317e+04
Approximated PSO	2.8333e+04	3.1817e+04	2.8617e+08	1.0609e+04	1.0696e+05
CMA-ES	2.5176e+06	2.1170e+06	3.7565e+12	5.9947e+03	6.4072e+06
FA	6.2778e+04	6.0448e+04	3.4061e+08	2.2850e+04	1.0338e+05
jDE	1.1471e+05	1.2350e+05	1.6194e+09	5.1220e+04	2.2188e+05
NDT-PSO	2.4811e+05	3.0432e+05	5.0280e+10	2.2803e+04	1.0167e+06

Table B.59: Results for f11 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	1.7317e+02	1.7259e+02	1.9914e+02	1.4577e+02	2.1231e+02
Approximated ES	3.1759e+02	3.1341e+02	1.2048e+03	2.3307e+02	3.8026e+02
Approximated FA	3.7834e+02	3.8981e+02	4.1128e+03	2.7019e+02	5.6786e+02
Approximated PSO	3.0707e+02	2.9699e+02	2.7207e+03	1.6621e+02	3.7370e+02
CMA-ES	5.5166e+02	5.6515e+02	2.7809e+04	2.7495e+02	1.0027e+03
FA	3.8940e+02	3.8958e+02	2.1687e+03	2.9062e+02	5.0997e+02
jDE	3.9703e+02	4.0596e+02	4.4528e+03	2.7132e+02	5.9611e+02
NDT-PSO	1.9369e+02	1.9553e+02	4.5530e+02	1.6127e+02	2.7089e+02

Table B.60: Results for f12 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-4.4893e+02	-4.2847e+02	5.7032e+03	-5.5152e+02	-2.1606e+02
Approximated ES	1.2667e+03	1.3295e+03	1.8572e+05	5.8665e+02	2.3511e+03
Approximated FA	-5.0971e+02	-5.0877e+02	1.1219e+02	-5.2847e+02	-4.8510e+02
Approximated PSO	-5.6479e+02	-1.9071e+02	3.9201e+06	-6.1842e+02	1.3400e+04
CMA-ES	2.5608e+08	2.4051e+08	7.3674e+15	5.2051e+03	3.6339e+08
FA	7.9603e+02	8.1170e+02	2.0881e+04	5.2193e+02	1.1437e+03
jDE	-6.1640e+02	-6.1393e+02	3.9421e+01	-6.2097e+02	-5.9701e+02
NDT-PSO	3.1362e+05	1.5318e+06	7.0215e+12	3.1259e+04	1.2626e+07

Table B.61: Results for f13 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	3.3941e+01	3.3911e+01	5.8385e-01	3.2518e+01	3.5471e+01
Approximated ES	5.6711e+01	5.9206e+01	1.5661e+02	4.4683e+01	1.0083e+02
Approximated FA	3.3186e+01	3.4073e+01	5.2355e+00	3.2134e+01	4.2316e+01
Approximated PSO	4.6748e+01	4.6304e+01	8.4618e+01	3.1439e+01	6.7226e+01
CMA-ES	2.6942e+03	2.2388e+03	1.1042e+06	3.7090e+01	2.9789e+03
FA	3.9092e+01	3.9521e+01	3.2213e+00	3.7476e+01	4.6954e+01
jDE	3.9749e+01	4.0879e+01	3.9251e+01	3.2699e+01	5.8457e+01
NDT-PSO	8.8434e+02	8.8125e+02	4.5851e+04	4.2970e+02	1.4237e+03

Table B.62: Results for f14 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-5.2345e+01	-5.2345e+01	8.2434e-07	-5.2347e+01	-5.2343e+01
Approximated ES	-5.2322e+01	-5.2290e+01	2.4976e-02	-5.2336e+01	-5.1302e+01
Approximated FA	-5.2344e+01	-5.2344e+01	4.5830e-07	-5.2345e+01	-5.2343e+01
Approximated PSO	-5.2339e+01	-5.2338e+01	3.4928e-05	-5.2345e+01	-5.2316e+01
CMA-ES	6.5510e+00	-5.6583e+00	7.8811e+02	-5.2337e+01	3.2245e+01
FA	-5.2338e+01	-5.2339e+01	1.5611e-06	-5.2341e+01	-5.2335e+01
jDE	-5.2333e+01	-5.2332e+01	1.8278e-05	-5.2339e+01	-5.2320e+01
NDT-PSO	-4.8457e+01	-4.8308e+01	7.6674e+00	-5.2197e+01	-4.2507e+01

Table B.63: Results for f15 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	1.1762e+03	1.1719e+03	5.3039e+02	1.0577e+03	1.1958e+03
Approximated ES	1.3393e+03	1.3264e+03	4.0235e+03	1.1842e+03	1.4492e+03
Approximated FA	1.1513e+03	1.1470e+03	1.1639e+03	1.0561e+03	1.2154e+03
Approximated PSO	1.2541e+03	1.2614e+03	6.1554e+03	1.1337e+03	1.5819e+03
CMA-ES	2.3583e+03	2.2170e+03	1.5473e+05	1.1577e+03	2.6082e+03
FA	1.1167e+03	1.1162e+03	6.1782e+02	1.0490e+03	1.1735e+03
jDE	1.4054e+03	1.4130e+03	2.1254e+03	1.2374e+03	1.5310e+03
NDT-PSO	1.3084e+03	1.3261e+03	6.1864e+03	1.2001e+03	1.5336e+03

Table B.64: Results for f16 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	1.0012e+02	9.9568e+01	8.0426e+00	9.0044e+01	1.0387e+02
Approximated ES	7.8175e+01	7.8076e+01	2.0158e+00	7.3704e+01	8.1620e+01
Approximated FA	8.9944e+01	9.0677e+01	2.0149e+01	8.2572e+01	1.0070e+02
Approximated PSO	8.6325e+01	8.6567e+01	2.7647e+01	7.6590e+01	9.9362e+01
CMA-ES	1.0888e+02	1.0935e+02	1.3546e+01	1.0068e+02	1.1594e+02
FA	7.2839e+01	7.2885e+01	3.0364e-01	7.1931e+01	7.4109e+01
jDE	1.0420e+02	1.0408e+02	6.6055e+00	9.8789e+01	1.0915e+02
NDT-PSO	9.7622e+01	9.7825e+01	9.4594e+00	8.8891e+01	1.0454e+02

Table B.65: Results for f17 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-1.6921e+01	-1.6918e+01	9.5959e-05	-1.6928e+01	-1.6874e+01
Approximated ES	-1.0550e+01	-1.0463e+01	1.2231e+00	-1.2521e+01	-7.1262e+00
Approximated FA	-1.6816e+01	-1.6731e+01	7.8505e-02	-1.6919e+01	-1.5484e+01
Approximated PSO	-1.1467e+01	-1.1554e+01	1.8059e+00	-1.3722e+01	-7.7914e+00
CMA-ES	-2.1608e+00	-2.1225e+00	2.3946e+00	-5.3873e+00	1.3151e+00
FA	-1.6520e+01	-1.6463e+01	1.1958e-01	-1.6901e+01	-1.5213e+01
jDE	-1.5321e+01	-1.5254e+01	3.2583e-01	-1.6275e+01	-1.3722e+01
NDT-PSO	-1.3218e+01	-1.3022e+01	1.1101e+00	-1.4610e+01	-1.0527e+01

Table B.66: Results for f18 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-1.6699e+01	-1.6644e+01	2.9210e-02	-1.6826e+01	-1.5981e+01
Approximated ES	8.0641e+00	7.9746e+00	1.6905e+01	-1.4528e+00	1.8400e+01
Approximated FA	-1.5393e+01	-1.4883e+01	2.0266e+00	-1.6343e+01	-9.3788e+00
Approximated PSO	3.5428e+00	2.7460e+00	4.2621e+01	-1.0946e+01	1.4688e+01
CMA-ES	4.1376e+01	4.0213e+01	3.4624e+01	2.0683e+01	5.1487e+01
FA	-1.4705e+01	-1.4550e+01	9.3681e-01	-1.6172e+01	-1.1740e+01
jDE	-1.0937e+01	-1.0810e+01	3.3876e+00	-1.3594e+01	-5.4595e+00
NDT-PSO	6.8422e-01	9.9572e-02	1.8867e+01	-1.0754e+01	7.2654e+00

Table B.67: Results for f19 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-9.7201e+01	-9.7289e+01	2.2036e-01	-9.8334e+01	-9.6530e+01
Approximated ES	-9.4272e+01	-9.4310e+01	4.7531e-01	-9.6023e+01	-9.2740e+01
Approximated FA	-9.6343e+01	-9.6445e+01	3.7686e-01	-9.7540e+01	-9.5202e+01
Approximated PSO	-9.6093e+01	-9.5993e+01	4.4281e-01	-9.7445e+01	-9.4659e+01
CMA-ES	-7.9619e+01	-7.9929e+01	1.8322e+01	-9.7446e+01	-7.2838e+01
FA	-9.6244e+01	-9.6361e+01	4.3351e-01	-9.8034e+01	-9.5078e+01
jDE	-9.5382e+01	-9.5429e+01	1.7321e-01	-9.6835e+01	-9.4637e+01
NDT-PSO	-9.6516e+01	-9.6567e+01	2.7159e-01	-9.7798e+01	-9.5460e+01

Table B.68: Results for f20 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-5.4340e+02	-5.4342e+02	1.8472e-02	-5.4384e+02	-5.4323e+02
Approximated ES	-5.4492e+02	-5.4492e+02	2.4941e-02	-5.4528e+02	-5.4464e+02
Approximated FA	-5.4503e+02	-5.4504e+02	2.1179e-02	-5.4544e+02	-5.4472e+02
Approximated PSO	-5.4505e+02	-5.4509e+02	4.6553e-02	-5.4567e+02	-5.4460e+02
CMA-ES	7.5067e+04	6.5964e+04	8.7605e+08	-5.4324e+02	1.1340e+05
FA	-5.4451e+02	-5.4451e+02	2.5563e-02	-5.4497e+02	-5.4414e+02
jDE	-5.4459e+02	-5.4461e+02	1.2913e-01	-5.4547e+02	-5.4400e+02
NDT-PSO	-5.4511e+02	-5.4513e+02	4.0977e-02	-5.4550e+02	-5.4466e+02

Table B.69: Results for f21 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	4.2899e+01	4.5614e+01	3.4408e+01	4.0780e+01	5.7892e+01
Approximated ES	4.1988e+01	4.1741e+01	1.0790e+00	4.0783e+01	4.6565e+01
Approximated FA	4.3253e+01	4.3531e+01	4.5822e+00	4.0780e+01	5.3635e+01
Approximated PSO	4.6561e+01	4.9519e+01	1.1233e+02	4.0780e+01	1.0167e+02
CMA-ES	1.2078e+02	1.2048e+02	4.1672e+00	1.1342e+02	1.2348e+02
FA	4.3254e+01	4.3287e+01	7.0439e-01	4.0780e+01	4.8098e+01
jDE	4.3253e+01	4.3265e+01	7.9401e-02	4.2025e+01	4.3950e+01
NDT-PSO	6.5838e+01	6.7807e+01	1.7927e+02	4.2376e+01	1.0560e+02

Table B.70: Results for f22 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-9.8542e+02	-9.8592e+02	6.2501e+00	-9.9804e+02	-9.8541e+02
Approximated ES	-9.9804e+02	-9.9718e+02	1.0834e+01	-9.9949e+02	-9.8541e+02
Approximated FA	-9.8541e+02	-9.9060e+02	5.4933e+01	-9.9931e+02	-9.6404e+02
Approximated PSO	-9.8542e+02	-9.8478e+02	1.3697e+02	-9.9930e+02	-9.5709e+02
CMA-ES	-9.1727e+02	-9.1730e+02	1.2549e+00	-9.2019e+02	-9.1513e+02
FA	-9.8541e+02	-9.9074e+02	4.0067e+01	-9.9931e+02	-9.8541e+02
jDE	-9.8541e+02	-9.9072e+02	3.9646e+01	-9.9804e+02	-9.8541e+02
NDT-PSO	-9.6667e+02	-9.6452e+02	2.3158e+02	-9.9389e+02	-9.3406e+02

Table B.71: Results for f23 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	1.0106e+01	1.0082e+01	1.6265e-01	9.1809e+00	1.0883e+01
Approximated ES	8.1780e+00	8.1742e+00	4.0750e-02	7.7058e+00	8.5683e+00
Approximated FA	9.4132e+00	9.4433e+00	4.9285e-01	8.0665e+00	1.1503e+01
Approximated PSO	1.0259e+01	1.0132e+01	4.6230e-01	8.8858e+00	1.1664e+01
CMA-ES	1.2450e+01	1.2273e+01	6.0274e-01	1.0130e+01	1.3924e+01
FA	7.9031e+00	7.9164e+00	6.0882e-02	7.5376e+00	8.6876e+00
jDE	1.1070e+01	1.1006e+01	2.0302e-01	1.0276e+01	1.1852e+01
NDT-PSO	1.0076e+01	1.0102e+01	1.7662e-01	9.1041e+00	1.0862e+01

Table B.72: Results for f24 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	3.1023e+02	3.1095e+02	1.1345e+02	2.9289e+02	3.3723e+02
Approximated ES	4.2267e+02	4.1810e+02	2.2323e+03	3.1942e+02	5.3466e+02
Approximated FA	3.3518e+02	3.3596e+02	1.2181e+03	2.5551e+02	4.2732e+02
Approximated PSO	3.8790e+02	3.8899e+02	1.1145e+03	3.3206e+02	4.6201e+02
CMA-ES	9.1187e+02	9.0777e+02	3.9272e+03	7.5041e+02	1.0080e+03
FA	2.9976e+02	2.9395e+02	1.2439e+03	2.3073e+02	3.7060e+02
jDE	5.7018e+02	5.7260e+02	4.5733e+02	5.1991e+02	6.2963e+02
NDT-PSO	3.5931e+02	3.5506e+02	6.9386e+02	3.0247e+02	3.9308e+02

Table B.73: Results for f25 func. with 20,000 function calls

alg	median	mean	std	min	max
Approximated DE	-6.0000e+00	-5.6200e+00	2.4041e-01	-6.0000e+00	-5.0000e+00
Approximated ES	0.0000e+00	2.0000e-01	1.6327e-01	0.0000e+00	1.0000e+00
Approximated FA	-6.0000e+00	-3.8800e+00	7.9445e+00	-6.0000e+00	-0.0000e+00
Approximated PSO	-2.0000e+00	-1.7600e+00	1.7371e+00	-4.0000e+00	0.0000e+00
CMA-ES	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
FA	-1.0000e+00	-1.2400e+00	1.6555e+00	-5.0000e+00	-0.0000e+00
jDE	-2.0000e+00	-2.5000e+00	2.0918e+00	-7.0000e+00	-1.0000e+00
NDT-PSO	-5.0000e-01	-5.2000e-01	2.9551e-01	-2.0000e+00	0.0000e+00

Table B.74: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f01 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	10.59	1.36e-17
Approximated DE	Approximated PSO	7.63	3.66e-11
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-17.33	1.32e-31
Approximated DE	jDE	23.32	2.13e-30
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	21.52	1.68e-33
Approximated ES	Approximated PSO	14.81	2.21e-26
Approximated ES	CMA-ES	-9.87	1.72e-15
Approximated ES	FA	4.22	6.05e-05
Approximated ES	jDE	23.57	2.43e-30
Approximated ES	NDT-PSO	-17.15	2.88e-31
Approximated FA	Approximated PSO	5.63	5.81e-07
Approximated FA	CMA-ES	-21.52	1.68e-33
Approximated FA	FA	-21.89	3.32e-34
Approximated FA	jDE	33.91	2.79e-55
Approximated FA	NDT-PSO	-21.52	1.68e-33
Approximated PSO	CMA-ES	-16.72	1.83e-30
Approximated PSO	FA	-14.94	1.51e-26
Approximated PSO	jDE	19.75	1.64e-32
Approximated PSO	NDT-PSO	-17.15	2.88e-31
CMA-ES	FA	10.37	1.92e-16
CMA-ES	jDE	23.57	2.43e-30
CMA-ES	NDT-PSO	9.51	1.21e-14
FA	jDE	24.06	7.20e-31
FA	NDT-PSO	-17.33	1.32e-31
jDE	NDT-PSO	-23.57	2.43e-30

Table B.75: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f02 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-15.35	8.79e-28
Approximated DE	Approximated FA	2.60	1.09e-02
Approximated DE	Approximated PSO	17.15	2.88e-31
Approximated DE	CMA-ES	-5.35	8.01e-07
Approximated DE	FA	-17.15	2.88e-31
Approximated DE	jDE	17.63	4.17e-32
Approximated DE	NDT-PSO	-5.93	7.25e-08
Approximated ES	Approximated FA	15.94	5.99e-29
Approximated ES	Approximated PSO	17.15	2.88e-31
Approximated ES	CMA-ES	-0.18	8.59e-01
Approximated ES	FA	1.46	1.49e-01
Approximated ES	jDE	17.63	4.17e-32
Approximated ES	NDT-PSO	1.13	2.61e-01
Approximated FA	Approximated PSO	17.15	2.88e-31
Approximated FA	CMA-ES	-6.70	2.10e-09
Approximated FA	FA	-17.15	2.88e-31
Approximated FA	jDE	17.63	4.17e-32
Approximated FA	NDT-PSO	-7.01	6.24e-10
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	-17.15	2.88e-31
Approximated PSO	jDE	13.60	2.85e-24
Approximated PSO	NDT-PSO	-17.15	2.88e-31
CMA-ES	FA	0.12	9.02e-01
CMA-ES	jDE	17.63	4.17e-32
CMA-ES	NDT-PSO	1.79	7.67e-02
FA	jDE	17.63	4.17e-32
FA	NDT-PSO	0.67	5.07e-01
jDE	NDT-PSO	-17.63	4.17e-32

Table B.76: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f03 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-14.81	1.82e-26
Approximated DE	Approximated FA	15.48	4.73e-28
Approximated DE	Approximated PSO	9.12	1.01e-14
Approximated DE	CMA-ES	-11.12	1.39e-18
Approximated DE	FA	7.91	4.04e-12
Approximated DE	jDE	6.31	8.11e-09
Approximated DE	NDT-PSO	8.36	4.40e-13
Approximated ES	Approximated FA	17.10	3.64e-31
Approximated ES	Approximated PSO	16.37	8.71e-30
Approximated ES	CMA-ES	-8.54	9.91e-13
Approximated ES	FA	16.03	4.03e-29
Approximated ES	jDE	15.62	2.71e-28
Approximated ES	NDT-PSO	16.47	5.60e-30
Approximated FA	Approximated PSO	-6.29	1.01e-08
Approximated FA	CMA-ES	-15.35	1.05e-27
Approximated FA	FA	-11.94	8.23e-21
Approximated FA	jDE	-15.14	2.18e-27
Approximated FA	NDT-PSO	-8.27	7.25e-13
Approximated PSO	CMA-ES	-14.06	6.28e-25
Approximated PSO	FA	-2.52	1.34e-02
Approximated PSO	jDE	-4.53	1.77e-05
Approximated PSO	NDT-PSO	-0.72	4.72e-01
CMA-ES	FA	13.21	5.47e-23
CMA-ES	jDE	12.84	4.27e-22
CMA-ES	NDT-PSO	13.57	6.73e-24
FA	jDE	-2.16	3.31e-02
FA	NDT-PSO	2.14	3.49e-02
jDE	NDT-PSO	4.21	5.94e-05

Table B.77: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f04 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	3.43	9.68e-04
Approximated DE	Approximated PSO	0.16	8.70e-01
Approximated DE	CMA-ES	-13.37	7.98e-24
Approximated DE	FA	-5.49	3.47e-07
Approximated DE	jDE	-0.12	9.02e-01
Approximated DE	NDT-PSO	1.34	1.85e-01
Approximated ES	Approximated FA	17.15	2.88e-31
Approximated ES	Approximated PSO	17.15	2.88e-31
Approximated ES	CMA-ES	-4.99	4.53e-06
Approximated ES	FA	17.10	3.64e-31
Approximated ES	jDE	17.15	2.88e-31
Approximated ES	NDT-PSO	17.15	2.88e-31
Approximated FA	Approximated PSO	-3.76	3.06e-04
Approximated FA	CMA-ES	-17.15	2.88e-31
Approximated FA	FA	-16.93	7.32e-31
Approximated FA	jDE	-8.61	1.32e-13
Approximated FA	NDT-PSO	-3.72	3.40e-04
Approximated PSO	CMA-ES	-13.54	3.66e-24
Approximated PSO	FA	-6.37	8.61e-09
Approximated PSO	jDE	-1.06	2.94e-01
Approximated PSO	NDT-PSO	0.86	3.92e-01
CMA-ES	FA	10.13	7.90e-17
CMA-ES	jDE	16.88	9.22e-31
CMA-ES	NDT-PSO	16.93	7.32e-31
FA	jDE	10.01	1.18e-16
FA	NDT-PSO	11.43	1.01e-19
jDE	NDT-PSO	2.63	1.01e-02

Table B.78: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f05 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	-13.67	2.07e-24
Approximated DE	Approximated PSO	1.92	5.81e-02
Approximated DE	CMA-ES	-3.26	1.80e-03
Approximated DE	FA	-17.15	2.88e-31
Approximated DE	jDE	17.15	2.88e-31
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	17.04	4.60e-31
Approximated ES	Approximated PSO	17.15	2.88e-31
Approximated ES	CMA-ES	-3.26	1.80e-03
Approximated ES	FA	-3.34	1.20e-03
Approximated ES	jDE	17.15	2.88e-31
Approximated ES	NDT-PSO	-17.15	2.88e-31
Approximated FA	Approximated PSO	11.43	1.09e-19
Approximated FA	CMA-ES	-3.26	1.80e-03
Approximated FA	FA	-16.72	1.83e-30
Approximated FA	jDE	17.15	2.88e-31
Approximated FA	NDT-PSO	-17.15	2.88e-31
Approximated PSO	CMA-ES	-3.26	1.80e-03
Approximated PSO	FA	-17.15	2.88e-31
Approximated PSO	jDE	17.15	2.88e-31
Approximated PSO	NDT-PSO	-17.15	2.88e-31
CMA-ES	FA	3.26	1.80e-03
CMA-ES	jDE	3.26	1.80e-03
CMA-ES	NDT-PSO	3.26	1.80e-03
FA	jDE	17.15	2.88e-31
FA	NDT-PSO	-15.27	1.22e-27
jDE	NDT-PSO	-17.15	2.88e-31

Table B.79: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f06 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	3.27	1.58e-03
Approximated DE	Approximated PSO	-17.15	2.88e-31
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-14.27	1.47e-25
Approximated DE	jDE	-17.15	2.88e-31
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	17.15	2.88e-31
Approximated ES	Approximated PSO	2.41	1.77e-02
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	16.72	1.83e-30
Approximated ES	jDE	6.55	2.74e-09
Approximated ES	NDT-PSO	-15.89	7.74e-29
Approximated FA	Approximated PSO	-16.99	5.81e-31
Approximated FA	CMA-ES	-17.15	2.88e-31
Approximated FA	FA	-12.15	3.00e-21
Approximated FA	jDE	-16.93	7.32e-31
Approximated FA	NDT-PSO	-17.15	2.88e-31
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	12.26	1.91e-21
Approximated PSO	jDE	1.81	7.43e-02
Approximated PSO	NDT-PSO	-16.32	1.09e-29
CMA-ES	FA	17.15	2.88e-31
CMA-ES	jDE	17.15	2.88e-31
CMA-ES	NDT-PSO	3.48	7.73e-04
FA	jDE	-13.64	2.64e-24
FA	NDT-PSO	-17.15	2.88e-31
jDE	NDT-PSO	-16.88	9.22e-31

Table B.80: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f07 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	-17.15	2.88e-31
Approximated DE	Approximated PSO	-17.15	2.88e-31
Approximated DE	CMA-ES	-6.66	2.25e-09
Approximated DE	FA	-17.15	2.88e-31
Approximated DE	jDE	-17.15	2.88e-31
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	16.67	2.29e-30
Approximated ES	Approximated PSO	8.42	3.24e-13
Approximated ES	CMA-ES	-3.66	5.15e-04
Approximated ES	FA	16.99	5.81e-31
Approximated ES	jDE	14.58	2.92e-26
Approximated ES	NDT-PSO	-2.31	2.30e-02
Approximated FA	Approximated PSO	-8.07	1.89e-12
Approximated FA	CMA-ES	-3.66	5.15e-04
Approximated FA	FA	2.00	4.81e-02
Approximated FA	jDE	-4.27	4.53e-05
Approximated FA	NDT-PSO	-15.44	5.61e-28
Approximated PSO	CMA-ES	-3.66	5.15e-04
Approximated PSO	FA	9.87	2.39e-16
Approximated PSO	jDE	4.10	8.90e-05
Approximated PSO	NDT-PSO	-9.42	2.65e-15
CMA-ES	FA	3.66	5.15e-04
CMA-ES	jDE	3.66	5.15e-04
CMA-ES	NDT-PSO	3.66	5.15e-04
FA	jDE	-6.11	2.08e-08
FA	NDT-PSO	-16.12	2.54e-29
jDE	NDT-PSO	-13.47	7.20e-24

Table B.81: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f08 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-8.13	1.83e-12
Approximated DE	Approximated FA	-15.06	5.36e-27
Approximated DE	Approximated PSO	-5.75	1.92e-07
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-15.02	6.76e-27
Approximated DE	jDE	-16.47	5.60e-30
Approximated DE	NDT-PSO	-16.32	1.09e-29
Approximated ES	Approximated FA	-1.53	1.29e-01
Approximated ES	Approximated PSO	-3.00	3.54e-03
Approximated ES	CMA-ES	-17.10	3.64e-31
Approximated ES	FA	-1.35	1.81e-01
Approximated ES	jDE	-10.35	2.50e-17
Approximated ES	NDT-PSO	-9.21	6.53e-15
Approximated FA	Approximated PSO	-3.69	3.93e-04
Approximated FA	CMA-ES	-17.15	2.88e-31
Approximated FA	FA	-1.80	7.47e-02
Approximated FA	jDE	-11.31	2.52e-19
Approximated FA	NDT-PSO	-9.37	3.00e-15
Approximated PSO	CMA-ES	-16.77	1.46e-30
Approximated PSO	FA	4.07	1.05e-04
Approximated PSO	jDE	-4.58	1.46e-05
Approximated PSO	NDT-PSO	-4.06	9.84e-05
CMA-ES	FA	17.10	3.64e-31
CMA-ES	jDE	16.27	1.35e-29
CMA-ES	NDT-PSO	16.22	1.68e-29
FA	jDE	-11.45	2.11e-19
FA	NDT-PSO	-9.55	1.16e-15
jDE	NDT-PSO	0.12	9.07e-01

Table B.82: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f09 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-3.19	2.01e-03
Approximated DE	Approximated FA	-14.97	8.55e-27
Approximated DE	Approximated PSO	-8.58	4.23e-13
Approximated DE	CMA-ES	-17.04	4.60e-31
Approximated DE	FA	-14.89	1.37e-26
Approximated DE	jDE	-15.31	1.36e-27
Approximated DE	NDT-PSO	-12.66	2.55e-22
Approximated ES	Approximated FA	-13.81	2.44e-24
Approximated ES	Approximated PSO	-5.68	1.52e-07
Approximated ES	CMA-ES	-15.98	5.01e-29
Approximated ES	FA	-13.95	1.22e-24
Approximated ES	jDE	-14.81	1.28e-26
Approximated ES	NDT-PSO	-7.31	8.44e-11
Approximated FA	Approximated PSO	1.27	2.08e-01
Approximated FA	CMA-ES	-14.81	2.21e-26
Approximated FA	FA	-1.54	1.27e-01
Approximated FA	jDE	-3.92	1.65e-04
Approximated FA	NDT-PSO	5.19	2.03e-06
Approximated PSO	CMA-ES	-15.18	2.69e-27
Approximated PSO	FA	-1.15	2.56e-01
Approximated PSO	jDE	-2.11	3.81e-02
Approximated PSO	NDT-PSO	0.55	5.84e-01
CMA-ES	FA	14.81	2.21e-26
CMA-ES	jDE	14.81	2.21e-26
CMA-ES	NDT-PSO	14.93	1.08e-26
FA	jDE	-3.41	9.70e-04
FA	NDT-PSO	5.11	2.81e-06
jDE	NDT-PSO	5.79	1.52e-07

Table B.83: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f10 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-15.57	3.00e-28
Approximated DE	Approximated FA	-16.67	2.29e-30
Approximated DE	Approximated PSO	-8.95	3.00e-14
Approximated DE	CMA-ES	-7.33	1.04e-10
Approximated DE	FA	-16.99	5.81e-31
Approximated DE	jDE	-17.15	2.88e-31
Approximated DE	NDT-PSO	-16.99	5.81e-31
Approximated ES	Approximated FA	-0.65	5.20e-01
Approximated ES	Approximated PSO	6.68	1.48e-09
Approximated ES	CMA-ES	-2.18	3.30e-02
Approximated ES	FA	-2.13	3.54e-02
Approximated ES	jDE	-14.02	4.01e-25
Approximated ES	NDT-PSO	-12.66	8.19e-22
Approximated FA	Approximated PSO	8.14	1.31e-12
Approximated FA	CMA-ES	-1.99	5.12e-02
Approximated FA	FA	-1.61	1.12e-01
Approximated FA	jDE	-13.84	9.38e-25
Approximated FA	NDT-PSO	-12.43	3.28e-21
Approximated PSO	CMA-ES	-3.49	8.01e-04
Approximated PSO	FA	-9.05	1.43e-14
Approximated PSO	jDE	-15.89	7.44e-29
Approximated PSO	NDT-PSO	-14.02	5.37e-25
CMA-ES	FA	1.94	5.72e-02
CMA-ES	jDE	1.75	8.53e-02
CMA-ES	NDT-PSO	1.81	7.45e-02
FA	jDE	-12.35	1.24e-21
FA	NDT-PSO	-12.02	2.25e-20
jDE	NDT-PSO	-6.36	9.58e-09

Table B.84: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f11 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	-17.15	2.88e-31
Approximated DE	Approximated PSO	-14.50	5.42e-26
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-17.15	2.88e-31
Approximated DE	jDE	-17.15	2.88e-31
Approximated DE	NDT-PSO	-6.71	1.27e-09
Approximated ES	Approximated FA	-8.03	2.48e-12
Approximated ES	Approximated PSO	1.22	2.27e-01
Approximated ES	CMA-ES	-11.55	1.13e-19
Approximated ES	FA	-10.27	3.24e-17
Approximated ES	jDE	-9.10	1.36e-14
Approximated ES	NDT-PSO	16.88	9.22e-31
Approximated FA	Approximated PSO	8.85	3.92e-14
Approximated FA	CMA-ES	-7.03	3.90e-10
Approximated FA	FA	-0.47	6.37e-01
Approximated FA	jDE	-1.43	1.55e-01
Approximated FA	NDT-PSO	17.10	3.64e-31
Approximated PSO	CMA-ES	-11.96	1.14e-20
Approximated PSO	FA	-10.76	2.84e-18
Approximated PSO	jDE	-9.70	6.34e-16
Approximated PSO	NDT-PSO	11.94	1.76e-20
CMA-ES	FA	7.45	7.40e-11
CMA-ES	jDE	6.14	2.22e-08
CMA-ES	NDT-PSO	17.15	2.88e-31
FA	jDE	-1.21	2.30e-01
FA	NDT-PSO	17.15	2.88e-31
jDE	NDT-PSO	17.15	2.88e-31

Table B.85: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f12 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	10.73	5.01e-18
Approximated DE	Approximated PSO	5.50	4.54e-07
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-17.15	2.88e-31
Approximated DE	jDE	17.15	2.87e-31
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	17.15	2.88e-31
Approximated ES	Approximated PSO	14.69	3.71e-26
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	9.39	4.05e-15
Approximated ES	jDE	17.15	2.87e-31
Approximated ES	NDT-PSO	-17.15	2.88e-31
Approximated FA	Approximated PSO	2.71	8.50e-03
Approximated FA	CMA-ES	-17.15	2.88e-31
Approximated FA	FA	-17.15	2.88e-31
Approximated FA	jDE	17.15	2.87e-31
Approximated FA	NDT-PSO	-17.15	2.88e-31
Approximated PSO	CMA-ES	-17.04	4.60e-31
Approximated PSO	FA	-13.91	1.55e-24
Approximated PSO	jDE	10.01	1.32e-16
Approximated PSO	NDT-PSO	-17.15	2.88e-31
CMA-ES	FA	17.15	2.88e-31
CMA-ES	jDE	17.15	2.87e-31
CMA-ES	NDT-PSO	11.33	8.17e-19
FA	jDE	17.15	2.87e-31
FA	NDT-PSO	-17.15	2.88e-31
jDE	NDT-PSO	-17.15	2.87e-31

Table B.86: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f13 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	2.23	2.85e-02
Approximated DE	Approximated PSO	-9.12	4.62e-14
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-17.15	2.88e-31
Approximated DE	jDE	-10.78	7.51e-18
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	17.15	2.88e-31
Approximated ES	Approximated PSO	6.29	9.20e-09
Approximated ES	CMA-ES	-6.90	1.72e-09
Approximated ES	FA	16.88	9.22e-31
Approximated ES	jDE	12.18	2.99e-21
Approximated ES	NDT-PSO	-17.15	2.88e-31
Approximated FA	Approximated PSO	-9.09	2.86e-14
Approximated FA	CMA-ES	-16.32	1.07e-29
Approximated FA	FA	-12.57	1.38e-21
Approximated FA	jDE	-9.26	4.87e-15
Approximated FA	NDT-PSO	-17.15	2.88e-31
Approximated PSO	CMA-ES	-8.95	4.96e-14
Approximated PSO	FA	4.90	5.46e-06
Approximated PSO	jDE	3.26	1.53e-03
Approximated PSO	NDT-PSO	-17.15	2.88e-31
CMA-ES	FA	12.57	8.06e-22
CMA-ES	jDE	11.19	4.43e-19
CMA-ES	NDT-PSO	6.59	7.01e-09
FA	jDE	-0.03	9.78e-01
FA	NDT-PSO	-17.15	2.88e-31
jDE	NDT-PSO	-17.15	2.88e-31

Table B.87: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f14 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	-7.18	1.42e-10
Approximated DE	Approximated PSO	-14.09	2.83e-25
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-17.16	2.78e-31
Approximated DE	jDE	-17.15	2.85e-31
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	17.17	2.66e-31
Approximated ES	Approximated PSO	12.18	3.04e-21
Approximated ES	CMA-ES	-10.50	3.30e-17
Approximated ES	FA	17.05	4.44e-31
Approximated ES	jDE	7.87	5.23e-12
Approximated ES	NDT-PSO	-16.52	4.47e-30
Approximated FA	Approximated PSO	-9.76	7.37e-16
Approximated FA	CMA-ES	-17.17	2.66e-31
Approximated FA	FA	-17.18	2.57e-31
Approximated FA	jDE	-17.17	2.64e-31
Approximated FA	NDT-PSO	-17.17	2.66e-31
Approximated PSO	CMA-ES	-15.53	3.73e-28
Approximated PSO	FA	-0.09	9.29e-01
Approximated PSO	jDE	-7.10	2.84e-10
Approximated PSO	NDT-PSO	-17.15	2.88e-31
CMA-ES	FA	16.94	7.07e-31
CMA-ES	jDE	13.24	3.04e-23
CMA-ES	NDT-PSO	5.00	4.36e-06
FA	jDE	-14.64	2.35e-26
FA	NDT-PSO	-17.16	2.78e-31
jDE	NDT-PSO	-17.15	2.85e-31

Table B.88: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f15 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-16.37	8.69e-30
Approximated DE	Approximated FA	5.05	2.11e-06
Approximated DE	Approximated PSO	-8.45	6.63e-13
Approximated DE	CMA-ES	-12.69	6.12e-22
Approximated DE	FA	12.52	7.80e-22
Approximated DE	jDE	-17.15	2.87e-31
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	16.62	2.86e-30
Approximated ES	Approximated PSO	5.26	8.56e-07
Approximated ES	CMA-ES	-9.60	7.14e-15
Approximated ES	FA	17.15	2.88e-31
Approximated ES	jDE	-9.28	4.60e-15
Approximated ES	NDT-PSO	0.85	3.95e-01
Approximated FA	Approximated PSO	-11.45	1.04e-19
Approximated FA	CMA-ES	-14.73	1.44e-26
Approximated FA	FA	5.43	4.49e-07
Approximated FA	jDE	-17.15	2.87e-31
Approximated FA	NDT-PSO	-16.88	9.22e-31
Approximated PSO	CMA-ES	-10.19	2.50e-16
Approximated PSO	FA	16.03	3.87e-29
Approximated PSO	jDE	-12.29	1.87e-21
Approximated PSO	NDT-PSO	-4.65	1.03e-05
CMA-ES	FA	16.93	7.32e-31
CMA-ES	jDE	9.51	1.21e-14
CMA-ES	NDT-PSO	9.57	8.83e-15
FA	jDE	-17.15	2.87e-31
FA	NDT-PSO	-17.15	2.88e-31
jDE	NDT-PSO	6.81	1.21e-09

Table B.89: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f16 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	17.15	2.88e-31
Approximated DE	Approximated FA	12.26	1.73e-21
Approximated DE	Approximated PSO	14.20	1.66e-25
Approximated DE	CMA-ES	-14.69	1.70e-26
Approximated DE	FA	17.15	2.88e-31
Approximated DE	jDE	-8.80	5.36e-14
Approximated DE	NDT-PSO	3.06	2.87e-03
Approximated ES	Approximated FA	-17.15	2.88e-31
Approximated ES	Approximated PSO	-13.81	1.50e-24
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	16.88	9.22e-31
Approximated ES	jDE	-17.15	2.88e-31
Approximated ES	NDT-PSO	-17.15	2.88e-31
Approximated FA	Approximated PSO	4.50	1.93e-05
Approximated FA	CMA-ES	-17.10	3.64e-31
Approximated FA	FA	17.15	2.88e-31
Approximated FA	jDE	-16.67	2.29e-30
Approximated FA	NDT-PSO	-9.24	5.97e-15
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	17.15	2.88e-31
Approximated PSO	jDE	-17.10	3.64e-31
Approximated PSO	NDT-PSO	-12.10	5.74e-21
CMA-ES	FA	17.15	2.88e-31
CMA-ES	jDE	8.53	2.42e-13
CMA-ES	NDT-PSO	15.98	4.76e-29
FA	jDE	-17.15	2.88e-31
FA	NDT-PSO	-17.15	2.88e-31
jDE	NDT-PSO	12.12	3.26e-21

Table B.90: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f17 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	-13.34	9.55e-24
Approximated DE	Approximated PSO	-17.15	2.88e-31
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-16.62	2.86e-30
Approximated DE	jDE	-17.15	2.88e-31
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	17.15	2.88e-31
Approximated ES	Approximated PSO	4.44	2.40e-05
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	17.15	2.88e-31
Approximated ES	jDE	17.15	2.88e-31
Approximated ES	NDT-PSO	12.69	2.31e-22
Approximated FA	Approximated PSO	-17.15	2.88e-31
Approximated FA	CMA-ES	-17.15	2.88e-31
Approximated FA	FA	-5.71	1.19e-07
Approximated FA	jDE	-15.62	2.53e-28
Approximated FA	NDT-PSO	-17.15	2.88e-31
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	17.15	2.88e-31
Approximated PSO	jDE	17.15	2.88e-31
Approximated PSO	NDT-PSO	6.29	8.93e-09
CMA-ES	FA	17.15	2.88e-31
CMA-ES	jDE	17.15	2.88e-31
CMA-ES	NDT-PSO	17.15	2.88e-31
FA	jDE	-13.95	5.46e-25
FA	NDT-PSO	-17.15	2.88e-31
jDE	NDT-PSO	-15.44	5.48e-28

Table B.91: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f18 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	-16.67	2.29e-30
Approximated DE	Approximated PSO	-17.15	2.88e-31
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-17.04	4.60e-31
Approximated DE	jDE	-17.15	2.88e-31
Approximated DE	NDT-PSO	-17.15	2.88e-31
Approximated ES	Approximated FA	17.15	2.88e-31
Approximated ES	Approximated PSO	4.49	2.03e-05
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	17.15	2.88e-31
Approximated ES	jDE	17.15	2.88e-31
Approximated ES	NDT-PSO	10.35	2.37e-17
Approximated FA	Approximated PSO	-16.93	7.32e-31
Approximated FA	CMA-ES	-17.15	2.88e-31
Approximated FA	FA	-2.79	6.32e-03
Approximated FA	jDE	-12.99	8.92e-23
Approximated FA	NDT-PSO	-16.99	5.81e-31
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	17.15	2.88e-31
Approximated PSO	jDE	14.93	5.83e-27
Approximated PSO	NDT-PSO	2.24	2.78e-02
CMA-ES	FA	17.15	2.88e-31
CMA-ES	jDE	17.15	2.88e-31
CMA-ES	NDT-PSO	17.15	2.88e-31
FA	jDE	-14.35	8.50e-26
FA	NDT-PSO	-17.15	2.88e-31
jDE	NDT-PSO	-14.69	1.78e-26

Table B.92: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f19 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-17.15	2.88e-31
Approximated DE	Approximated FA	-7.44	4.25e-11
Approximated DE	Approximated PSO	-11.96	9.66e-21
Approximated DE	CMA-ES	-14.35	1.20e-25
Approximated DE	FA	-7.78	8.84e-12
Approximated DE	jDE	-16.47	5.58e-30
Approximated DE	NDT-PSO	-7.36	6.34e-11
Approximated ES	Approximated FA	15.80	1.09e-28
Approximated ES	Approximated PSO	12.99	5.04e-23
Approximated ES	CMA-ES	-13.05	2.14e-22
Approximated ES	FA	15.31	1.00e-27
Approximated ES	jDE	10.23	8.21e-17
Approximated ES	NDT-PSO	16.62	2.86e-30
Approximated FA	Approximated PSO	-3.31	1.30e-03
Approximated FA	CMA-ES	-13.24	6.49e-23
Approximated FA	FA	-0.84	4.03e-01
Approximated FA	jDE	-10.52	9.14e-18
Approximated FA	NDT-PSO	1.04	3.00e-01
Approximated PSO	CMA-ES	-13.08	1.75e-22
Approximated PSO	FA	2.31	2.30e-02
Approximated PSO	jDE	-4.90	4.10e-06
Approximated PSO	NDT-PSO	4.75	7.16e-06
CMA-ES	FA	13.18	9.65e-23
CMA-ES	jDE	13.05	2.14e-22
CMA-ES	NDT-PSO	13.18	9.65e-23
FA	jDE	-9.57	1.06e-15
FA	NDT-PSO	1.92	5.77e-02
jDE	NDT-PSO	12.49	6.63e-22

Table B.93: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f20 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	17.15	2.88e-31
Approximated DE	Approximated FA	17.15	2.86e-31
Approximated DE	Approximated PSO	17.15	2.88e-31
Approximated DE	CMA-ES	-17.04	4.60e-31
Approximated DE	FA	17.15	2.87e-31
Approximated DE	jDE	17.15	2.88e-31
Approximated DE	NDT-PSO	17.15	2.88e-31
Approximated ES	Approximated FA	3.65	4.28e-04
Approximated ES	Approximated PSO	4.60	1.28e-05
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	-14.09	2.97e-25
Approximated ES	jDE	-6.11	3.32e-08
Approximated ES	NDT-PSO	5.89	5.68e-08
Approximated FA	Approximated PSO	1.53	1.28e-01
Approximated FA	CMA-ES	-17.15	2.86e-31
Approximated FA	FA	-16.15	2.29e-29
Approximated FA	jDE	-7.71	2.95e-11
Approximated FA	NDT-PSO	3.01	3.33e-03
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	-15.31	9.76e-28
Approximated PSO	jDE	-8.09	2.95e-12
Approximated PSO	NDT-PSO	1.20	2.35e-01
CMA-ES	FA	17.15	2.87e-31
CMA-ES	jDE	17.15	2.88e-31
CMA-ES	NDT-PSO	17.15	2.88e-31
FA	jDE	1.18	2.41e-01
FA	NDT-PSO	15.62	2.44e-28
jDE	NDT-PSO	8.56	2.97e-13

Table B.94: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f21 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	1.89	6.24e-02
Approximated DE	Approximated FA	-0.18	8.58e-01
Approximated DE	Approximated PSO	-2.53	1.29e-02
Approximated DE	CMA-ES	-17.15	2.87e-31
Approximated DE	FA	-0.40	6.91e-01
Approximated DE	jDE	-0.61	5.47e-01
Approximated DE	NDT-PSO	-12.29	1.63e-21
Approximated ES	Approximated FA	-8.21	9.54e-13
Approximated ES	Approximated PSO	-7.11	3.15e-10
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	-10.63	5.59e-18
Approximated ES	jDE	-12.96	7.88e-22
Approximated ES	NDT-PSO	-16.22	1.65e-29
Approximated FA	Approximated PSO	-3.09	2.75e-03
Approximated FA	CMA-ES	-17.88	1.64e-32
Approximated FA	FA	-3.85	2.14e-04
Approximated FA	jDE	-1.14	2.56e-01
Approximated FA	NDT-PSO	-13.66	1.09e-23
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	2.92	4.65e-03
Approximated PSO	jDE	2.72	8.55e-03
Approximated PSO	NDT-PSO	-8.83	4.08e-14
CMA-ES	FA	17.43	9.04e-32
CMA-ES	jDE	19.02	7.96e-34
CMA-ES	NDT-PSO	17.15	2.88e-31
FA	jDE	5.15	1.42e-06
FA	NDT-PSO	-13.33	6.83e-23
jDE	NDT-PSO	-14.05	3.87e-23

Table B.95: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f22 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	10.27	5.89e-17
Approximated DE	Approximated FA	2.82	5.80e-03
Approximated DE	Approximated PSO	3.39	1.15e-03
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	-1.41	1.64e-01
Approximated DE	jDE	2.73	7.42e-03
Approximated DE	NDT-PSO	-10.67	3.50e-17
Approximated ES	Approximated FA	-2.07	4.15e-02
Approximated ES	Approximated PSO	-6.42	6.28e-09
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	-2.40	1.86e-02
Approximated ES	jDE	-2.72	7.80e-03
Approximated ES	NDT-PSO	-16.47	5.56e-30
Approximated FA	Approximated PSO	-1.26	2.12e-01
Approximated FA	CMA-ES	-18.19	6.13e-33
Approximated FA	FA	-3.14	2.35e-03
Approximated FA	jDE	-0.33	7.45e-01
Approximated FA	NDT-PSO	-12.42	1.42e-21
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	0.68	4.99e-01
Approximated PSO	jDE	1.02	3.09e-01
Approximated PSO	NDT-PSO	-7.87	4.88e-12
CMA-ES	FA	18.38	3.46e-33
CMA-ES	jDE	18.38	3.46e-33
CMA-ES	NDT-PSO	17.15	2.88e-31
FA	jDE	3.05	3.19e-03
FA	NDT-PSO	-13.14	1.11e-22
jDE	NDT-PSO	-13.14	1.11e-22

Table B.96: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f23 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	17.15	2.88e-31
Approximated DE	Approximated FA	5.83	8.29e-08
Approximated DE	Approximated PSO	-0.54	5.93e-01
Approximated DE	CMA-ES	-15.57	3.18e-28
Approximated DE	FA	17.15	2.88e-31
Approximated DE	jDE	-11.43	9.88e-20
Approximated DE	NDT-PSO	-0.21	8.37e-01
Approximated ES	Approximated FA	-13.91	1.22e-24
Approximated ES	Approximated PSO	-17.15	2.88e-31
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	6.19	1.51e-08
Approximated ES	jDE	-17.15	2.88e-31
Approximated ES	NDT-PSO	-17.15	2.88e-31
Approximated FA	Approximated PSO	-5.11	1.59e-06
Approximated FA	CMA-ES	-16.22	1.65e-29
Approximated FA	FA	15.98	4.76e-29
Approximated FA	jDE	-13.21	2.56e-23
Approximated FA	NDT-PSO	-6.15	1.91e-08
Approximated PSO	CMA-ES	-14.35	8.56e-26
Approximated PSO	FA	17.15	2.88e-31
Approximated PSO	jDE	-7.53	2.65e-11
Approximated PSO	NDT-PSO	0.42	6.76e-01
CMA-ES	FA	17.15	2.88e-31
CMA-ES	jDE	9.87	4.47e-16
CMA-ES	NDT-PSO	15.44	5.76e-28
FA	jDE	-17.15	2.88e-31
FA	NDT-PSO	-17.15	2.88e-31
jDE	NDT-PSO	10.91	1.28e-18

Table B.97: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f24 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-16.47	5.57e-30
Approximated DE	Approximated FA	-5.41	6.05e-07
Approximated DE	Approximated PSO	-16.93	7.32e-31
Approximated DE	CMA-ES	-17.15	2.88e-31
Approximated DE	FA	3.43	9.21e-04
Approximated DE	jDE	-17.15	2.88e-31
Approximated DE	NDT-PSO	-10.50	1.58e-17
Approximated ES	Approximated FA	10.33	2.51e-17
Approximated ES	Approximated PSO	3.48	7.57e-04
Approximated ES	CMA-ES	-17.15	2.88e-31
Approximated ES	FA	15.44	5.42e-28
Approximated ES	jDE	-17.10	3.64e-31
Approximated ES	NDT-PSO	8.13	1.88e-12
Approximated FA	Approximated PSO	-8.22	8.77e-13
Approximated FA	CMA-ES	-17.15	2.88e-31
Approximated FA	FA	6.34	7.22e-09
Approximated FA	jDE	-17.15	2.88e-31
Approximated FA	NDT-PSO	-3.10	2.55e-03
Approximated PSO	CMA-ES	-17.15	2.88e-31
Approximated PSO	FA	14.54	3.41e-26
Approximated PSO	jDE	-17.15	2.88e-31
Approximated PSO	NDT-PSO	5.47	3.59e-07
CMA-ES	FA	17.15	2.88e-31
CMA-ES	jDE	17.15	2.88e-31
CMA-ES	NDT-PSO	17.15	2.88e-31
FA	jDE	-17.15	2.88e-31
FA	NDT-PSO	-10.35	2.25e-17
jDE	NDT-PSO	17.15	2.88e-31

Table B.98: Pairwise ranked Welch's t -test on algorithms for 30 dimensional f25 func. with 20,000 function calls

alg1	alg2	t-value	P -value
Approximated DE	Approximated ES	-22.26	2.28e-39
Approximated DE	Approximated FA	-1.44	1.52e-01
Approximated DE	Approximated PSO	-18.87	4.48e-34
Approximated DE	CMA-ES	-28.84	2.12e-32
Approximated DE	FA	-18.43	2.54e-33
Approximated DE	jDE	-12.87	3.79e-21
Approximated DE	NDT-PSO	-19.99	2.50e-36
Approximated ES	Approximated FA	10.37	1.47e-16
Approximated ES	Approximated PSO	11.09	9.09e-18
Approximated ES	CMA-ES	3.50	1.00e-03
Approximated ES	FA	9.74	3.17e-15
Approximated ES	jDE	20.47	1.87e-35
Approximated ES	NDT-PSO	7.75	1.77e-11
Approximated FA	Approximated PSO	-3.86	2.40e-04
Approximated FA	CMA-ES	-10.06	1.66e-13
Approximated FA	FA	-4.29	5.06e-05
Approximated FA	jDE	-2.21	2.99e-02
Approximated FA	NDT-PSO	-5.26	1.25e-06
Approximated PSO	CMA-ES	-10.97	8.52e-15
Approximated PSO	FA	-2.13	3.57e-02
Approximated PSO	jDE	2.11	3.72e-02
Approximated PSO	NDT-PSO	-5.46	5.01e-07
CMA-ES	FA	9.34	1.85e-12
CMA-ES	jDE	25.24	9.67e-30
CMA-ES	NDT-PSO	6.98	7.04e-09
FA	jDE	5.23	1.03e-06
FA	NDT-PSO	-3.00	3.47e-03
jDE	NDT-PSO	-12.33	1.58e-21