◆目的

本稿は、佐藤君の原稿構想をサポートする目的で執筆したものである。

◆スケジュール

8月31日～9月3日：発表

6月15日：原稿投稿締切

6月5日：実験結果まとめる、研究構想まとめる、原稿執筆（可能な箇所から）

◆タイトル

有制約最適化のためのDifferential Evolutionの基礎検討

◆構成案

1章　はじめに

2章　有制約最適化

　2.1節　有制約最適化問題の定式化

　2.2節　制約対処法

3章　有制約最適化のための近傍生成法の解析

　3.1節　先行研究の近傍生成法と課題

3.2節　探索戦略と多様化・集中化の評価指標

3.3節　数値実験検証

4章　有制約最適化のためのDifferential Evolution

　4.1節　有制約最適化へ拡張したDE

　4.2節　参照点の動的調整機能を有する有制約DE

5章　数値実験検証

6章　おわりに

Appendix１　Multiple Constraint Ranking

◆構想案

●1章　はじめに

・メタヒューリスティクス[1]は、自然現象や物理現象から着想を得た発見的な最適化手法であり、実数値Genetic Algorithm（実数値GA）[2]やDifferential Evolution（DE）[3]が知られている。

・さらに、直接探索・多点探索の特徴を有するため、ブラックボックス最適化や大域的最適化において有力な手法だが、無制約最適化での適用に限定されている。

・そこで、メタヒューリスティクスの直接探索・多点探索の維持・活用を念頭に置きながら、有制約最適化へ拡張するための手法群として、制約対処法（Constraint Handling Techniques：CHT）[4]が知られている。

・有制約最適化では、目的関数と制約を逸脱した量（制約違反量）の両方の景観やトレードオフ関係を考慮しながら、実行可能な最適解を求める必要がある。

・これは、次の候補解（近傍解）を生成する近傍生成法と、候補解の選択・比較において制約条件を考慮するCHTを組み合せることで実現できる。

・代表的なCHTとして、Adaptive Penalty[5]や制約法[6]、Multiple Constraint Handling[7]、Infeasibility Driven Evolutionary Algorithm[8]などが知られており、数多く提案されている。

・これらのアルゴリズムは、有制約最適化へ拡張したメタヒューリスティクスという観点で見ると、独自のCHTと実数値GAで構成されており、特に近傍生成法はビットストリング交叉やSimulated Binary Crossoverなど非常に単純な操作を採用するケースがほとんどである。

・しかしながら、高次元や多数制約など複雑な問題では、CHTだけでなく、近傍生成も探索効率に多大な影響を与える可能性が高い。

・そこで本稿では、多様化・集中化の観点から、有制約最適化における近傍生成の影響を明らかにする。

・さらに、無制約最適化で有力なDEをベースとして、有制約最適化のための近傍生成法を検討し、その探索性能について検証する。

●2.1節　有制約最適化問題の定式化

・有制約最適化問題を記述し、notationを示す。

●2.2節　制約対処法

・CHTのアプローチは、ペナルティベース、分離ベース、多目的最適化ベースの3種類に大別できる。

・ペナルティベースは、制約違反量の総和をペナルティとして目的関数に追加し、無制約最適化問題として解く方法である。

・古典的な方法として、拡張Lagrange関数法が知られているが、ブラックボックス・非凸な目的関数や制約関数が含まれる場合、局所解に停滞する可能性が高い、あるいはペナルティ係数の適切な設定が困難であるなどの問題がある。

・そこで、探索過程で適応的にペナルティ係数を調整するAdaptive Penalty（APM）[5]が提案されている。

・他にも、分離ベースとして制約法[6]やMultiple Constraint Handling（MCR）[7]、多目的最適化ベースとしてInfeasibility Driven Evolutionary Algorithm（IDEA）[8]やDeCODE[9]が知られている。

・CHTは、目的関数と制約違反量の両方の景観を考慮するために、メタヒューリスティクスのアルゴリズム中の解の選択・比較方法を工夫する。

・本稿では、解の選択・比較はMCRによって行うことにする。具体的には、アルゴリズムで使用されている目的関数値の順序関係を、MCRによって定められる適合度の順序関係に置き換える。MCRの具体的なアルゴリズムはAppendix 1を参照されたい。

●3.1節　先行研究の近傍生成法と課題

・2章で挙げたCHTの先行研究で使用されている実数値GAの近傍生成法について述べる。

・実数値GAの近傍生成は交叉と突然変異で構成される場合があるが、交叉が主要な影響を与える一方、突然変異はあくまで補助的な手段で、探索性能への影響は比較的小さい[2]ため、本稿では交叉のみに注目する。

・Lemongeら[5]や高濱ら[6]は、ビットストリング型の交叉（一点交叉や一様交叉）を使用している。

・ビットストリング型の交叉は、実数値ベクトルをビットストリングに変換してから要素を組み替えるため、解空間上で目的関数の景観や探索点の分布をほぼ考慮しない近傍解が生成される。

・Garcia ら[7]は、Blend Crossover-α（BLX-α）、Rayら[8]らは、Simulated Binary Crossover（SBX）を使用している。

・これらの交叉は二つの探索点を基準に、要素毎に確率分布（BLX-αは一様分布、SBXは多項式分布）に基づいて近傍解を生成するが、探索点間の相互作用の活用度合いは低い。

・このように、これらの先行研究では、独自のCHTを提案している一方、近傍生成法は非常に簡易的な方法を採用しているが、数値実験は20次元などの比較的易しい条件での性能検証に留まっている。

・しかしながら、高次元などの複雑な条件になった場合、CHTによって制約条件を考慮した解の比較・選択が可能になっているものの、これらの交叉では、有望な近傍解が生成できないため、探索効率が低下すると考えられる。

・そこで本稿では、有制約最適化における近傍生成法の影響について明らかにする。

●3.2節　探索戦略と多様化・集中化の評価指標

・メタヒューリスティクスは近接最適性原理（POP）に基づき、解空間の偏り構造を把握・活用することで、効率的な探索を行う。

・POPを効果的に活用する探索戦略として、多様化・集中化が知られている[1]。

・多様化は、探索範囲・探索点分布を広くしておくことで、解空間の偏り構造を把握する指針で、集中化は、把握した偏り構造を活用し、探索範囲・探索点分布を有望領域に徐々に狭めて解の改善を狙う指針である。

・探索過程で多様化・集中化がどの程度実現されているのかを評価するための指標として、本稿では下記の探索点間の平均距離を用いる。

ただし、（）は探索点の位置、はユークリッド距離である。式中の分数は正規化のためにかけている。

・探索点間の平均距離が大きい場合、探索点群が解空間内に広く分布しているため多様化、探索点間の平均距離が小さい場合、探索点群が解空間内に狭く分布しているため集中化というように、探索状態を評価することが期待できる。

・さらに、有制約最適化では実行可能な大域的最適解を探索するため、探索点群が実行可能領域の境界に到達しているかどうかは重要である。

・本稿では下記の実行可能解の割合を用いる。

ただし、は内部の条件を満たす集合の要素数である。が1に近い場合、探索点群は可能解の割合が高く、が0に近い場合、探索点群は可能解の割合が低い。

・本稿では、上記の指標を用いることで、有制約最適化における可能解の獲得状況や多様化・集中化の実現状態を評価する。

●3.3節　数値実験検証

・次元数が低い場合と高い場合を比較し、探索性能が一気に落ちていることを指摘する。

・探索過程の最良解の推移と、評価指標（平均距離や可能解の割合）の推移を示し、多様化・集中化の実現状態と探索効率の関係を指摘する。

・CHTを組み合わせても、実数値GAの近傍生成法を使用する場合、高次元では多様化・集中化の観点で探索効率が落ち、実行可能解が得られないことを指摘する。

●4.1節　有制約最適化へ拡張したDE

・DEは、何種類かの演算モデルがあり、X/Y/Zで表される。Xは突然変異の演算モデル、Yは突然変異における差分ベクトルの個数、Zは交叉の演算モデルである。

・DEのアルゴリズムを示し、次世代の選択部分を「探索点群と近傍解集合を合わせた解集合から、適合度に基づいて上位個を選び、次世代の探索点群とする。」という記述にしておく。

・適合度はMCRによって与えられるものとする。

・DEは探索点間の差分ベクトルを使用するため、探索点間の相互作用が比較的強い上に、探索点分布のスケールや偏りが近傍に反映されやすい。

・このため、3章で述べた実数値GAの簡易的な交叉よりも、多様化・集中化の実現が期待できる。

●4.2節　参照点の動的調整機能を有する有制約DE

・有制約DEの先行研究について述べる。

・Wangら[9]は、多目的最適化ベースのCHTとDEを組み合わせたDeCODEを提案している。

・DeCODEは、解の比較・選択において、問題分割に基づく適合度を利用している。

・一方、近傍生成では、DEの2種類の演算モデルを確率的に選択・使用する。片方の演算モデルはrand-to-best/1/bin、もう片方の演算モデルはcurrent-to-rand/1（交叉無し）である。最良な探索点の選び方においても、制約条件を考慮した適合度を使用している。

・さらに、探索過程で選択確率を線形に増加させ、後者のモデルから前者のモデルに徐々に移行することで、多様化・集中化の戦略実現を狙う。

・また、Heら[6]は、正規化法の先行研究について体系的に整理し、それらの影響を数値実験的に検証し、正規化法の影響が大きいことを指摘している。

・しかしながら、これらの先行研究は、多目的最適化における正規化法の影響に関する考察や数値実験検証に留まっている。

・さらに、Wangら[2]は、MOEA/Dを有制約最適化に応用した方法を提案しており、簡潔な正規化法を使用しているが、有制約最適化における正規化の影響については詳細に述べられていない。

・上記の先行研究に基づき、本稿では、有制約最適化において有効な正規化法を検討し、正規化法の影響について明らかにする。

・下記では、有制約最適化における正規化法について述べる。

・目的関数値と制約違反量のWeighted Sumのスカラ化関数は式(1)で表される。

・正規化は、スカラ化関数におけるとを[0,1]の範囲のとで置換する。

・簡潔な正規化法としては、とを下記のとで置換する方法が考えられる。

ただし、は分母が0になることを回避するための微小値、とは正規化の基準点であり、多目的最適化における理想点と最悪点に対応している。

・実問題では真の基準点を入手することは困難であるため、探索過程で基準点を逐次的に求め、使用する必要がある。

・図1は、多目的最適化の先行研究で使用されている基準点の設定方法を示している[6]。

・Heらは、これらの基準点を利用した正規化方法を多目的最適化問題に適用・比較し、方法3が平均的に優れていることを確認している。

・図1の4種類の基準点を有制約最適化に応用すると、下記のように書ける。

・方法1　理想点：最小点 in 探索履歴

最悪点：最大点 in 現在の探索点群

・方法2　理想点：最小点 in 現在の探索点群

最悪点：最大点in 現在の探索点群

・方法3　理想点：最小点 in 探索履歴

最悪点：最大点 in 現在の探索点群の非優越解

・方法4　理想点：最小点 in 現在の探索点群

最悪点：最大点 in 現在の探索点群の非優越解

・本稿では、上記の正規化法をMOEA/DのCHTに適用する。

・なお、DeCODE[2]やAdaptive Weighted MOEA/D[3]では、探索過程で重みも同時に調整することで、実行可能解への収束性を狙っているが、本稿では、スケール性の課題に対する正規化の効果を調べるために、重みを固定した設定とする。

・ただし、αによって重みの散らばりを調整する方式で、α固定がやりやすいかも。

●5章　数値実験検証

・スケール差が無い場合、スケール差が大きい場合の問題について、4種類の正規化法を適用して比較する。

・探索性能だけでなく、探索挙動・軌跡も注目する。（実行可能解付近のパレートフロンティアの一様性に影響するかどうかがポイントだから）

●6章　おわりに

・今後は、簡潔な正規化法だけでなく、工夫した正規化法を適用することも考えられる。

・例えば、Heらは、正規化の悪影響は下記の説明によって、最悪点の不正確な推定に起因することを指摘し、正規化の程度を動的に制御する方法を提案している[7]。

・初期世代では、探索点群がパレートフロンティアから大きく離れているため、推定された最悪点はパレートフロンティアの上界を近似することができない。このように、最悪点の推定が不正確な場合、大きくスケールアップされた目的関数空間は探索の方向を誤らせる。

・本稿は、単一制約での正規化法のみを示したが、多数制約への拡張の検討が考えられる。

●Appendix１　Multiple Constraint Ranking

・MCRは、解集合内で目的関数値のランクや制約違反量のランクなどを算出し、そのトータルランクに基づいて適合度を与える制約対処法の一種である。具体的には、ある解集合に含まれる解同士で比較・評価するとき、あるの適合度を下記で与える。

ただし、は目的関数値のランク、は制約違反量のランク、は制約違反数のランクであり、各ランクは解集合におけるの順位である。本稿では、無制約メタヒューリスティクスのアルゴリズムの中で使用されている目的関数値の順序関係を、上記の適合度の順序関係に書き換えることで、制約条件を扱う。



図1：正規化のための基準点の設定方法[6]

◆参考文献

1. 相吉・安田 編著：「メタヒューリスティクスと応用」、オーム社（2007）
2. 小野・山村：喜多：「実数値GAとその応用」、人工知能学会誌、Vol. 15、No. 2、pp. 259-266（2000）
3. R. Storn and K. Price: “Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces,” Journal of Global Optimization, Vol. 11, No. 4, pp. 341-359 (1997)
4. E. Mezura-Montes and C. A. C. Coello: “Constraint-Handling in Nature-Inspired Numerical Optimizations: Past, Present and Future,” Swarm and Evolutionary Computation, Vol. 1, No. 4, pp. 173-194 (2011)
5. A. Lemonge and H. Barbosa: “An Adaptive Penalty Scheme for Genetic Algorithms in Structural Optimization,” International Journal for Numerical Methods in Engineering, Vol. 59, No. 5, pp. 703-736 (2004)
6. 高濱・阪井：「制約遺伝的アルゴリズムによる制約付き最適化」、情報処理学会論文誌、Vol. 47、No. 6、pp. 1861-1871（2006）
7. R. Garcia, B. Lima, A. Lemonge, and B. Jacob: “A Rank-Based Constraint Handling Technique for Engineering Design Optimization Problems Solved by Genetic Algorithms,” Computers and Structures, Vol. 188, pp. 77-87 (2017)
8. T. Ray, H.K. Singh, A. A. Isaacs, and W. Smith: “Infeasibility Driven Evolutionary Algorithm for Constrained Optimization,” Constraint-Handling in Evolutionary Optimization, pp. 145-165 (2009)
9. B. Wang, H. Li, Q. Zhang, and Y. Wang: “Decomposition-Based Multiobjective Optimization for Constrained Evolutionary Optimization,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 51, No. 1, pp. 574-587 (2021)
10. L. He, H. Ishibuchi, A. Trivedi, and D. Srinivasan: “Dynamic Normalization in MOEA/D for Multiobjective Optimization,” Proceedings of the 2020 IEEE Congress on Evolutionary Computation, pp. 1-8 (2020)