◆目的

本稿は、小嶋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・構想まとめ

◆タイトル

有制約最適化のためのMOEA/Dに基づく制約対処法の検討

◆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　問題分割に基づく制約対処法の解析

　3.1節　MOEA/Dに基づく制約対処法

　3.2節　先行研究と課題

3.3節　数値実験検証

4章　有制約最適化のための正規化法

　4.1節　正規化法の先行研究

　4.2節　有制約最適化のための正規化法

5章　数値実験検証

6章　おわりに

◆構想案

●1章　はじめに

・多目的最適化のための問題分割は、異なる重みベクトルとスカラ化関数によって、複数のサブ単一目的問題に分割して解く方法で、MOEA/D[1]が代表的である。

・問題分割ベースのCHTは、問題分割を有制約最適化へ拡張し、実行可能な大域的最適解を探索する。

・Wangら[2]や安田ら[3]は、MOEA/Dを有制約最適化に拡張した方法を提案している。

・しかしながら、多目的最適化のためのMOEA/Dは、各軸のスケールや難易度が異なる場合、パレートフロンティアを一様に近似できず、探索性能が劣化することが知られている[4]。

・本稿では、MOEA/Dを用いた問題分割ベースのCHTの課題について数値実験を通じて明らかにし、より有用性の高い方法の検討を行う。

●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、分離、多目的の3種類に大別できる。

・多目的最適化アプローチは、制約違反解の活用度合いが最も高いため、有制約最適化に有力である。

・さらに、多目的最適化アプローチは、パレートランキングと問題分割の方法に分類される。

・パレートランキングベースのCHTは、(f,v)空間上の優越関係や混雑距離に基づくランクで適合度を与える方法で、IDEA[4]が代表的である。

・有制約最適化にパレートランキングをそのまま応用すると、弱パレート解である実行可能解は探索過程で淘汰されやすいため、パレートランキングベースのCHTでは、実行可能解を一定数保持する工夫を加えられている。

・さらに、(f,v)空間の軸のスケールやパレートフロンティアの形状の影響を受けないという長所を持つ。

・しかしながら、実行可能領域が狭い場合、パレートフロンティア全体も広くなり、実行可能解の獲得が難しくなるという点で、有制約最適化への有効性に大きな課題がある。

・一方、問題分割ベースのCHTは、DeCODE[2]やAdaptive Weighted MOEA/D[3]がある。

・有制約最適化に問題分割をそのまま応用すると、パレートフロンティア全体を広く一様に近似するが、唯一の実行可能な大域的最適解を獲得する上で非効率であるため、問題分割ベースのCHTでは、探索過程で重みを調整する工夫がなされている。

・さらに、 (f,v)空間の軸のスケールやパレートフロンティアの形状の影響を受けるものの、基本的に重みとパレート解の対応が多少崩れても、実行可能解の獲得能力を発揮する点で、パレートランキングベースのCHTよりも有利だと考えられる。

・実際に、安田ら[3]は、パレートランキングベースのCHTであるIDEAよりも、問題分割ベースのCHTであるAdaptive Weighted MOEA/Dの探索性能が優れていることを数値実験的に示している。

・よって、本稿では、問題分割ベースのCHTに着目し、その性能改善を目指す。

●3.1節　MOEA/Dに基づく制約対処法

・MOEA/Dを有制約最適化に拡張した場合の数式を説明する。（とりあえず重み固定のバージョン）

●3.2節　先行研究と課題

・MOEA/Dの課題として、(1)目的関数空間の各軸のスケールが異なる場合、あるいは(2)スカラ化関数とパレートフロンティアの形状が合致しない場合、重みとパレート解の対応関係が崩れ、パレートフロンティアを一様に近似できないため、解の収束性が落ちることが指摘されている[1]。

・本稿では、MOEA/Dの一つ目の課題であるスケール性への対応性に着目する。

・先行研究では、MOEA/Dの原著論文[1]以外にも、Ishibuchi[5]らはMOEA/Dのスケール性の影響を数値実験的に検証しているが、多目的最適化における影響を指摘することに留まっており、有制約最適化へ応用した場合の影響は不明である。

・そこで本稿では、有制約最適化におけるスケール差が大きい場合の影響を明らかにする。

●3.3節　数値実験検証

・スケール差が無い場合と大きい場合で適用し、探索効率や探索の挙動を考察する。

・スケール差が大きいことが有制約最適化においても多大な影響を与えることを確認し、正規化法によってこの影響を緩和することが必要であることを述べる。

●4.1節　正規化法の先行研究

・多目的最適化における問題分割は、目的関数空間の各軸の値をスカラ化関数によって、スカラの適合度に変換して、解を比較する。

・しかしながら、各軸のスケールが異なると、各軸が適合度に与える影響に大きな差が生まれ、パレートフロンティアを一様に探索できない。

・そこで、正規化は、理想点（ideal point）・最悪点（nadir point）と呼ばれる基準点に基づき、目的関数空間の各軸の値をmin-maxスケーリングし、[0,1]の範囲に収める。

・3.3節で指摘したように、有制約最適化においても目的関数値と制約違反量のスケール差は発生しやすいことから、多目的最適化と同様に、正規化法を検討し、その影響の差を緩和することは意義が大きい。

・MOEA/Dに関する正規化法の先行研究について述べる。

・原著論文[1]では、簡潔な正規化法を示している。

・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。

●4.2節　有制約最適化のための正規化法

・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。



図1：正規化のための基準点の設定方法[6]

◆参考文献

1. Q. Zhang and H. Li: “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition,” IEEE Transactions on Evolutionary Computation, Vol. 11, No. 6, pp. 712-731 (2007)
2. 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)
3. 安田・熊谷・田村・安田：「MOEA/Dの有制約最適化への拡張と適応的重み調整に関する基礎検討」、電気学会論文誌C、Vol. 142、No. 1、pp.108-109（2022）
4. 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)
5. H. Ishibuchi, K. Doi, and Y. Nojima: “On the Effect of Normalization in MOEA/D for Multi-Objective and Many-Objective Optimization,” Complex & Intelligence Systems, Vol. 3, No. 4, pp. 279-294 (2017)
6. L. He, Y. Nan, K. Shang, and H. Ishibuchi: “A Study of the Naive Objective Space Normalization Method in MOEA/D,” Proceedings of the IEEE Symposium Series on Computational Intelligence, pp. 1834-1840 (2019)
7. 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)