

jh250063

最適化ベンチマーク問題自動生成のための大規模言語モデル駆動型進化的

アルゴリズム

原田 智広 (埼玉大学)

概要

本研究では、大規模言語モデル (LLM) と進化計算を組み合わせ、最適化アルゴリズムの評価に用いるベンチマーク問題を自動生成する手法を構築した。提案手法では、数式やプログラムコードで表現されたベンチマーク問題を進化計算における個体とみなし、LLM を初期生成や交叉、突然変異に相当する生成演算子として用いる。これにより、特定のアルゴリズムが優れる問題、高難易度な問題、多目的最適化アルゴリズム間の性能差を引き出す問題を自動生成できることを示した。また、Python コードとして記述される目的関数の生成にも取り組み、従来の閉形式数式では表現しにくい複雑なベンチマーク問題を生成する基盤を構築した。これらの成果により、最適化アルゴリズムの特性分析と評価を支援する LLM 駆動型ベンチマーク生成の有効性を示した。

1. 共同研究に関する情報

(1) 共同利用・共同研究を実施している拠点名

東京大学 情報基盤センター

(2) 課題分野

大規模計算科学課題分野

(3) 参加研究者一覧と役割分担

原田 智広 (埼玉大学)・研究総括、提案法の基本設計

上野 史 (岡山大学)・大規模言語モデルを用いた処理の性能向上

高玉 圭樹 (東京大学)・最適化手法の性能向上

大野 愉展 (東京都立大学)・プログラム実装、結果の分析

2. 研究の目的と意義

○ 研究の目的

本研究では、ブラックボックス最適化の代表的な手法である進化的アルゴリズム (Evolutionary

Algorithm: EA) を含む最適化アルゴリズムの分析・評価に用いる最適化ベンチマーク問題を自動生成するためのアルゴリズムを提案し、その有効性を検証することを目的とする。

EA は、その汎用性の高さから、多様な実世界の最適化問題に適用されてきた。しかし、EA の探索性能や収束速度、計算コストを適切に評価するためには、多様な性質を有する最適化ベンチマーク問題が不可欠である。No Free Lunch 定理が示すように、すべての最適化問題に対して常に優れた性能を示す単一のアルゴリズムは存在しない。そのため、アルゴリズムの特性を理解するには、さまざまな問題特性を持つベンチマークセットによる包括的な分析が求められる。

これまでに最適化アルゴリズムの評価を目的とした多くのベンチマーク問題が提案されている [1] が、既存の人工的なベンチマーク問題は現実問題の多様な構造を十分に反映できていないという課題がある [2]。一方、実問題ベースのベンチマークは、現実的な構造を反映できるものの、実行時

間の長さや構築コストの高さに加え、機密性の問題から公開が難しいという制約を伴う。このような背景から、実問題に含まれる性質を備えながら、効率的に実行可能なベンチマーク問題の設計が課題となっている。

この課題を解決するために、本研究では大規模言語モデル (Large Language Model: LLM) と EA を統合したベンチマーク問題生成アルゴリズムを提案する。近年、LLM は自然言語処理分野で著しい進展を遂げており、EA との融合研究も進展している。例えば、LLM を用いた交叉や突然変異に相当する変異演算を実現する Language Model Crossover (LMX) [3]などの手法が提案されている。これらの成果を踏まえ、本研究ではベンチマーク問題を数式やプログラムコードとして表現し、LLM と EA の連携により多様性を有するベンチマーク問題を自動生成する枠組みを確立する。特に、ベンチマーク問題の評価基準を調整することで、所望の性質を持つベンチマーク問題を生成可能にする。

○ 研究の意義

本研究は、多様な性質を持つベンチマーク問題を自動的に生成可能なアルゴリズムを提供することにより、EA をはじめとする最適化アルゴリズムの包括的な分析と評価を可能にする点に意義がある。提案手法により、アルゴリズムの性能を多面的に検証できるだけでなく、その利点や欠点を明確にすることで、既存手法の改良や新たなアルゴリズムの開発を促進する。

LLM を活用する利点として、生成可能なベンチマークの表現の自由度が挙げられる。既存のベンチマーク自動生成手法は、事前に定めたパラメータ範囲や木構造表現に依存することが多く、生成可能な問題構造が制限される。一方、LLM-EBG では、数式をテキストベースの記号表現として扱うことで、より柔軟な表現空間において問題構造を進化させることができる。

さらに、生成されるベンチマーク問題は特定の課題や条件に応じた性質を意図的に反映させることが可能である。そのため、最適化アルゴリズム

の欠点を強調するような敵対的なベンチマーク問題の生成を通して、アルゴリズムの欠点を明確にし、それを克服する新しい最適化アルゴリズムの開発につなげることができる。このような自動生成の仕組みは、最適化アルゴリズムの設計や適用範囲の拡張に貢献し、最適化技術の発展を支える重要な基盤となる。

3. 当拠点の公募型共同研究として実施した意義

提案手法の実現のためには、EA の最適化過程において、LLM を用いた処理を多数回実行する必要がある。具体的には、初期ベンチマーク問題の生成に加え、交叉や突然変異といった各段階で LLM による処理を繰り返し適用する必要がある。特に、一般的な EA では多数の遺伝的操作と評価を繰り返す必要があり、本研究でも、ベンチマーク問題の生成・評価・選択を反復するため、多数の LLM 推論と最適化アルゴリズムの実行が必要となる。

一方、1 度の LLM 推論に数秒から数分の GPU 時間を要するため、提案手法の一連の処理を現実的な時間内に完了するためには LLM を並列実行する必要がある。また、近年の高性能な LLM は、70B や 405B の莫大なパラメータを有するため、実行には莫大なメモリ容量と高性能な GPU の利用が必須である。

これらの観点から、本研究の遂行は研究室レベルの計算資源では困難であり、本公募型共同研究を通じて提供される大規模な計算資源の利用（特に、高性能な GPU 資源を用いた並列処理）が不可欠である。

4. 前年度までに得られた研究成果の概要

該当なし

5. 今年度の研究成果の詳細

今年度の研究を通じて以下の成果を得た。

(1) 単目的最適化ベンチマーク問題の自動生成フレームワークの構築

本研究では、最適化アルゴリズムの性能評価に

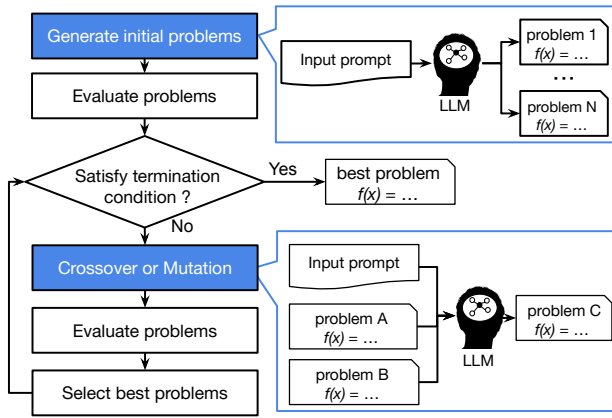


図 1 LLM-EBG の概略図

用いるベンチマーク問題を自動生成するため、LLM と EA を統合した LLM-driven Evolutionary Benchmark Generator (LLM-EBG) を構築した。LLM-EBG の全体像を図 1 に示す。本手法では、数式で表現されたベンチマーク問題を進化計算における個体とみなし、LLM を初期個体生成や交叉、突然変異に相当する生成演算子として用いる。生成された問題は、対象とする最適化アルゴリズム間の性能差に基づいて評価され、その評価値をもとに次世代の問題が選択される。この処理を繰り返すことで、特定のアルゴリズムの優位性や弱点を顕在化させる問題を自動的に探索する。

実験では、5次元の単目的連続値最適化問題を対象とし、遺伝的アルゴリズム (Genetic Algorithm: GA) [4] と差分進化 (Differential Evolution: DE) [5] の性能差が明確に現れる問題を生成した。具体的には、GA が DE より優れる問題を生成する場合と、DE が GA より優れる問題を生成する場合の 2 つのシナリオを設定した。その結果、GA を優位にするシナリオでは 10 試行中 8 試行、DE を優位にするシナリオでは 10 試行中 9 試行で、目標とするアルゴリズムが比較対象アルゴリズムを一貫して上回るベンチマーク問題を生成できた。

さらに、生成された問題に対して探索景観解析 (図 2) を行った結果、GA に有利な問題 (図 3) は変数スケールの影響が大きく、悪スケール性を有する傾向があることが確認された。一方、DE に有利な問題 (図 4) は、変数スケールの偏りが比較的小さく、DE が個体間の相対的な位置関

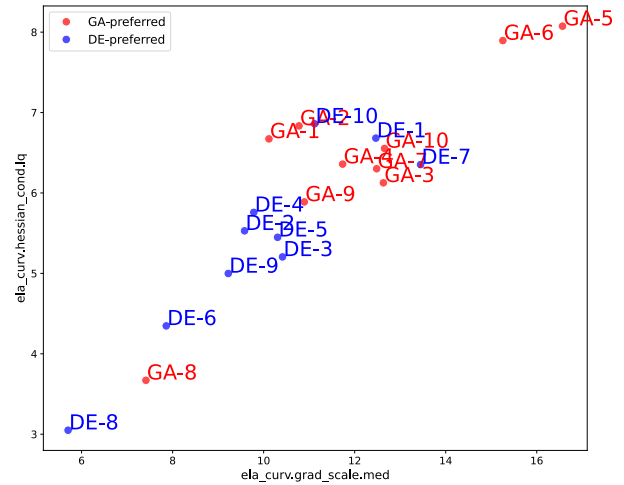


図 2 生成ベンチマークの探索景観特徴の分析

$$f(x) = \frac{x_0^2 + \sin(x_1)x_2 + |x_3 - x_4| + \sqrt{|x_0 - x_1|} + \frac{x_2x_3}{1 + x_4^2 + |\sin(x_0)\sinh(x_1)|} + \frac{|x_2 - x_3|^2}{1 + |x_4|} + \frac{x_0x_1x_2x_3x_4}{1 + \sum_{i=0}^4 |x_i|} + \sin(x_0)\sin(x_1)\sin(x_2)\sin(x_3)\sin(x_4) + \frac{|x_0 - x_1|^2}{1 + x_2^2} + \cos(x_0)\cos(x_1)\cos(x_2)\cos(x_3)\cos(x_4) + \frac{x_3x_4}{1 + |x_0| + |x_1| + |x_2|}$$

図 3 GA が優れる問題の生成例

$$f(x) = x_0^2 + |x_1x_2| + \sqrt{|x_3|} - \sin(x_4) + \sin(x_0x_1) + \cos(x_2x_3) + \frac{x_0}{1 + x_4^2} + \sinh(x_1x_2x_3) + |x_0 - x_1 + x_2 - x_3 + x_4| + \sqrt{x_0^2 + x_1^2 + x_2^2 + x_3^2 + x_4^2} + x_1 \sinh(x_0x_2) + \frac{|x_2 - x_3|}{\sqrt{1 + x_4^2}}$$

図 4 DE が優れる問題の生成例

係を利用しやすい構造を持つ傾向が見られた。このことから、LLM-EBG は単に性能差のある問題を生成するだけでなく、アルゴリズムの探索特性と対応する問題構造を明らかにする分析手法としても有効であることが示された。

(2) 高難易度な単目的最適化ベンチマーク問題の自動生成

ベンチマーク問題には、アルゴリズム間の性能差を明確化するだけでなく、単一のアルゴリズムの探索能力や頑健性を評価する役割も求められる。そこで本研究では、特定のアルゴリズムが安定して最良解に到達できないような、高難易度な単目的最適化ベンチマーク問題を自動生成する手法を構築した。

具体的には、図 5 に示すように生成された問題に対して最適化アルゴリズムを複数回独立に実行し、各試行で得られた最良目的関数値を比較することで、探索難易度を評価した。その上で、最良

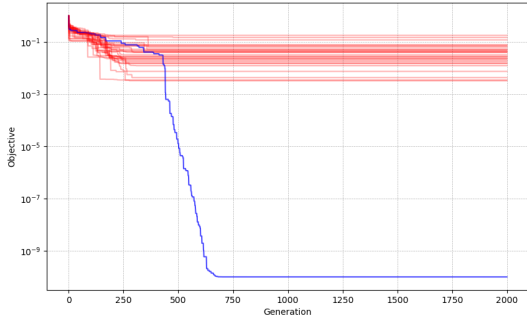


図 5 高難易度ベンチマークの探索イメージ

解と 2 番目に良い解の差が大きい問題を、高難易度な問題として評価した。これは、ごく少数の試行で非常に良い解に到達できるものの、多くの試行では同等の解に到達できない問題を探索するためである。このような問題では、大域的最適解が他の局所解から強く孤立している可能性が高く、アルゴリズムの探索能力をより明確に評価できる。

本手法では、ベンチマーク問題を数式として表現し、LLM が生成した数式を Python で安全に評価できる形式へ整形した。さらに、不適切な文字列や構文を除去する事前スクリーニングを行ったうえで評価した。この処理により、LLM が生成する多様な数式表現を活用しつつ、実験で評価可能なベンチマーク問題として扱えるようにした。

実験では、5 次元の単目的連続最小化問題を対象とし、DE を評価対象アルゴリズムとして高難易度ベンチマークの生成を行った。生成された 20 個のベンチマーク問題を評価した結果、多くの問題で探索成功率が極めて低くなり、17 個の問題では成功率が 1/30 から 5/30 の範囲にとどまった。この結果は、提案手法により、DE が安定して最良解に到達することが難しい問題を自動生成できたことを示している。

生成された問題(図 6)の構造を分析した結果、高難易度化に寄与する要素として、三角関数、双曲線関数、平方根、絶対値関数などの非線形・非平滑な構成要素が多く含まれていることが確認された。特に、周期性を生む \sin 項、急激なスケール変化を生む \sinh 項、非微分点を生む絶対値項が組み合わせられることで、平坦領域、多峰性、スケール不均衡、局所解の多さが生じ、探索を難しく

$$f(x) = (x_0^2 + x_1^2 - 10) + 100(|x_2 - x_3|^2 + |x_4 - x_0|^2) + \sin(x_1 x_4)^2 + \sqrt{x_2^2 + x_3^2}$$

図 6 DE が最適化困難なベンチマーク生成例していると考えられる。

さらに、生成された問題が DE に特化したものではないことを確認するため、GA および粒子群最適化 (Particle Swarm Optimization: PSO) [6] を用いた追加評価も行った。その結果、DE が最も良い評価値を示した問題は 20 問中 11 問であり、PSO は 5 問、GA は 4 問で最良の結果を示した。また、多くの問題で GA や PSO に対しても成功率が低く、生成された問題集合は DE だけでなく、異なる探索メカニズムを持つアルゴリズムに対しても高い探索難易度を持つことが確認された。

(3) 多目的最適化ベンチマーク問題の自動生成

(1)(2)の成果では単目的最適化問題の生成を対象としたのに対し、本成果では LLM-EBG を多目的最適化問題を生成可能にする拡張に取り組んだ。単目的最適化問題の自動生成では、数式で表現された単一の目的関数を個体として扱う枠組みであるのに対し、複数の目的関数から構成される多目的最適化問題を 1 つの個体として扱うことで、多目的進化計算アルゴリズムの性能差を引き出すベンチマーク問題を自動生成できるようにした。

多目的最適化では、単一の最適解ではなく、複数の目的間のトレードオフを表すパレート最適解集合を求める必要がある。そのため、アルゴリズムの性能評価には、パレートフロントの形状、目的関数間のスケール差、トレードオフ構造など、多様な問題特性を持つベンチマークが必要となる。既存の ZDT や DTLZ などの標準ベンチマークは、多目的進化計算の発展に大きく貢献してきた一方で、限られた問題構造に基づいて設計されているため、未知の問題構造に対するアルゴリズムの頑健性や一般化性能を十分に評価できない可能性がある。本成果は、この課題に対して、LLM-EBG を多目的問題に拡張し、多目的ベンチマーク問題を自動生成する枠組みを提案したものである。

提案手法では、2 目的最小化問題を対象とし、1

つの個体を 2 つの目的関数の組として表現した。例えば、2 目的最適化の場合、1 つのベンチマーク問題は $[f_1(x), f_2(x)]$ という 2 つの数式文字列の集合として表現される。LLM は、この複数目的関数の組に対して、初期個体生成、交叉、突然変異を行う。プロンプトでは、目的関数数、設計変数の次元数、使用可能な演算子、出力形式を明示し、さらに「目的関数が互いに競合すること」を指示することで、多目的最適化問題として意味のある問題を生成するようにした。

また、多目的問題では、単に数式が評価可能であるだけでは不十分であり、目的関数間に意味のあるトレードオフが存在する必要がある。そのため、本研究では生成された問題に対して事前検証および事後検証を導入した。事前検証では、ランダムに生成した多数の入力点に対して目的関数値を計算し、無限大、複素数、ゼロ除算などの不正な値が生じる問題を除外した。さらに、事後検証では、パレートフロントが単一解に潰れているような問題を排除し、トレードオフを持つ多目的ベンチマークのみを評価対象とした。

生成問題の評価には、代表的な多目的進化的アルゴリズムである NSGA-II[7]と MOEA/D[8]を用いた。NSGA-II はパレート支配に基づいて探索する代表的な手法であり、MOEA/D は多目的問題を複数のスカラー最適化問題に分解して解く分解型手法である。この 2 つは探索原理が異なるため、生成されたベンチマークがアルゴリズム特性の違いを引き出せるかを検証するのに適している。本研究では、NSGA-II が MOEA/D より優れる問題を生成するシナリオと、MOEA/D が NSGA-II より優れる問題を生成するシナリオの 2 つを設定した。

生成問題の評価には、多目的最適化における代表的な指標であるハイパーボリューム (HV) を用いた。各生成問題に対して NSGA-II と MOEA/D を複数回独立に実行し、各試行で得られた非劣解集合の HV を計算した。その後、両アルゴリズムの HV 値を順位付けし、順位和の差に基づいて生成問題の適応度を定義した。これにより、単一試行

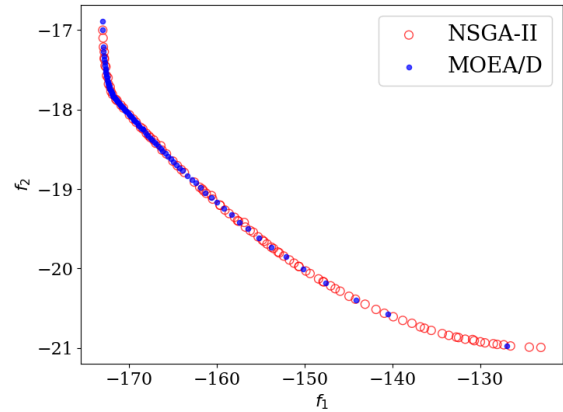


図 7 NSGA-II が優れる生成問題の最適化結果の結果ではなく、複数試行にわたって対象アルゴリズムが安定して優位となる問題を探索できるようにした。

実験では、2 目的 5 次元の連続最小化問題を対象とした。各生成問題の評価では NSGA-II と MOEA/D をそれぞれ 20 回独立に実行した。その結果、NSGA-II に有利な問題の生成では 5 試行すべてで最大適応度に到達した。一方、MOEA/D に有利な問題の生成では、5 試行中 2 試行で高い適応度に到達し、NSGA-II 優位問題の方が生成しやすいことが確認された。

生成された問題の分析から、NSGA-II に有利な問題 (図 7) では、2 つの目的関数のスケールが大きく異なることが明らかになった。このような問題では、MOEA/D で用いられるスカラー化関数である Tchebycheff 関数がスケールの大きい目的関数に支配されやすく、探索が一方の目的に偏る。その結果、MOEA/D はパレートフロント全体にわたる多様な解を得ることが難しくなる。これに対し、NSGA-II はパレート支配関係に基づいて探索するため、目的関数値の絶対的なスケール差に比較的頑健であり、広範囲のパレートフロントを発見できた。この結果は、提案手法が MOEA/D の既知の弱点である目的関数スケール差への感度を引き出す問題を生成できたことを示している。

一方、MOEA/D に有利な問題 (図 8) では、滑らかな凸型のパレートフロントを持つ問題が生成された。このような問題では、MOEA/D の参照方向に基づく分解型探索が有効に機能し、特に HV

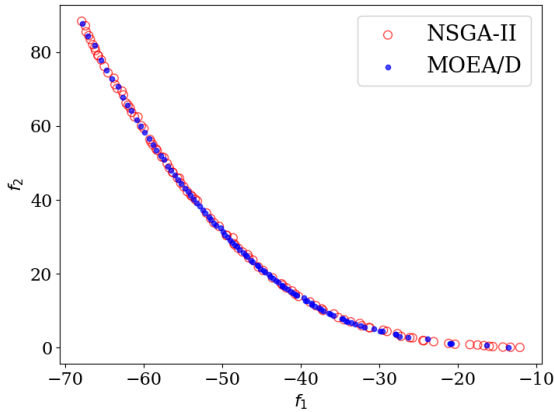


図 8 MOEA/D が優れる生成問題の最適化結果
 への寄与が大きいパレートフロント中央部を密に探索できた。その結果、NSGA-II よりも高い HV を達成した。NSGA-II は混雑距離に基づいて極端解を含む広範な解の分布を維持するため、HV 改善への寄与が相対的に小さい領域にも計算資源を割く。一方、MOEA/D は参照方向に沿って探索を進めるため、凸型フロントでは効率的に中央部の解を改善できたと解釈できる。

さらに、生成された問題を既存ベンチマークと比較するため、ZDT、DTLZ、BBOB-biobj の問題群とともにランドスケープ特徴量を計算し、t-SNE により可視化した。その結果、図 9 に示すように、MOEA/D に有利な問題は ZDT ベンチマークに近い領域に配置された一方で、NSGA-II に有利な問題は既存ベンチマークとは離れた領域にクラスターを形成した。このことは、既存ベンチマークが NSGA-II の利点が現れやすい特徴空間を十分にカバーしていない可能性を示しており、提案手法によって従来のベンチマーク集合に不足している問題構造を補完できることを示唆している。

(4) プログラムコード表現を用いたベンチマーク生成

本研究では、これまで数式表現を対象としてきた LLM-EBG をさらに発展させ、Python コードとして記述される最適化ベンチマーク問題を自動生成する枠組みを検討した。従来の数式ベースのベンチマーク生成では、目的関数を閉形式の数式として表現し、LLM によって数式文字列を交叉・

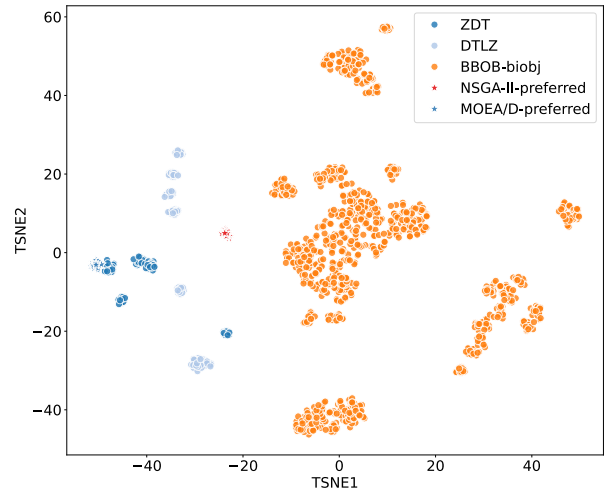


図 9 多目的ベンチマーク問題のランドスケープ特徴解析

突然変異させることで、多様なベンチマーク問題を生成していた。一方、実世界の最適化問題は、単純な閉形式の数式だけでなく、条件分岐、多段階の計算、内部状態を持つ処理、複数の補助関数から構成されることも多い。そのため、より現実的な問題構造を表現するためには、数式だけでなくプログラムコードをベンチマーク問題の表現として扱うことが重要となる。

この課題に対して、本研究では、Python で実装された目的関数コードを個体として扱い、LLM を用いてコードに対する交叉や突然変異に相当する変更を生成する。これにより、単なる数式の組合せにとどまらず、条件分岐、モジュール化された計算処理、確率的要素、複数段階の計算などを含む、より柔軟で表現力の高いベンチマーク問題を生成できる可能性がある。本成果の重要な点は、ベンチマーク問題の「表現空間」を拡張したことにある。数式ベースの生成では、三角関数、平方根、絶対値、四則演算などを組み合わせることで多様な関数景観を生成できるが、表現できる問題構造は基本的に 1 つの数式に限定される。一方、コードベース表現では、目的関数をアルゴリズム的な処理として記述できるため、実世界の評価関数に近い複雑な構造を扱うことができる。これは、単に難しい関数を作るだけでなく、最適化アルゴリズムの挙動をより実践的な条件下で評価するための基盤になる。

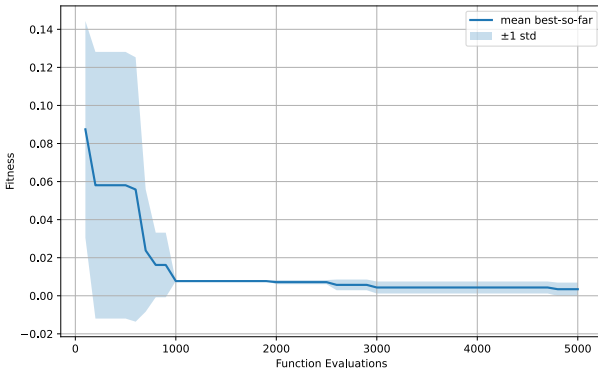


図 10 コードベース生成の目的関数推移

本研究では、コードベースのベンチマーク生成における評価基準として、複数の最適化アルゴリズムの相対的な性能の高さに着目した。まず、GA、ES、DE、PSO、CMA-ES、Nelder-Mead 法、焼きなまし法、ランダムサーチの 8 種類の代表的な最適化手法を対象とし、BBOB ベンチマーク問題上での性能に基づいて Elo レーティングを算出した。Elo レーティングは、チェスなどの対戦型競技で用いられる評価法であり、本研究ではアルゴリズム同士の性能比較を一種の対戦結果とみなすことで、各アルゴリズムの相対的な性能順位を定量化した。

提案手法では、この基準的な性能順位をもとに、既存ベンチマーク上で得られたアルゴリズム順位と逆の挙動を引き出すような問題の生成を目指した。すなわち、BBOB 上では高性能と評価されるアルゴリズムが必ずしも優位にならず、通常は低い順位にあるアルゴリズムが相対的に良好な性能を示す問題を探索することで、既存ベンチマークでは見えにくいアルゴリズム特性を明らかにすることを狙った。具体的には、統計的検定に基づく問題上のアルゴリズム順位と、Elo レーティングから期待される順位の不一致度を評価した。

実験では、LLM-EBG によってコードベースのベンチマークを探索し、図 10 に示すように世代が進むにつれて目的関数値が段階的に低下（アルゴリズム間の順位が逆転）しており、提案した評価基準に従って、より既存の性能順位を崩す問題が探索できる可能性が示された。

図 11 と図 12 に示した生成されたベンチマーク

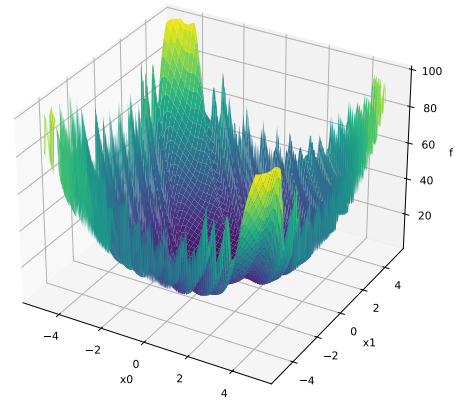


図 11 コードベースベンチマーク①の関数景観

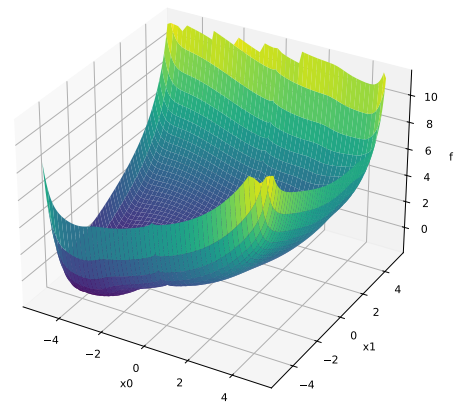


図 12 コードベースベンチマーク②の関数景観の 3 次元景観から、コードベースの生成では滑らかな単峰的構造だけでなく、複雑な凹凸や周期的・多峰的な構造を持つ問題が生成されていることが確認できる。また、これらのベンチマークを 8 種類の最適化アルゴリズムで最適化した結果を図 13 と図 14 に示す。既存の BBOB 上では CMA-ES が最も探索性能が高く、Elo レーティングが最上位であるが、生成されたベンチマークでは、CMA-ES が必ずしも最良の探索性能を示さず、GA や DE、PSO よりも劣る結果となることが確認できる。特に図 11 に示すベンチマーク①では、図 13 の結果のようにランダム探索が他の最適化アルゴリズムと比べて安定した性能を示しており、既存ベンチマークとは異なるアルゴリズム特性を引き出す特徴的な問題であることが確認された。

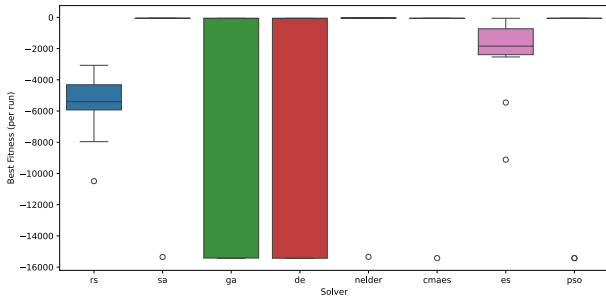


図 13 コードベースベンチマーク①の最適化結果比較

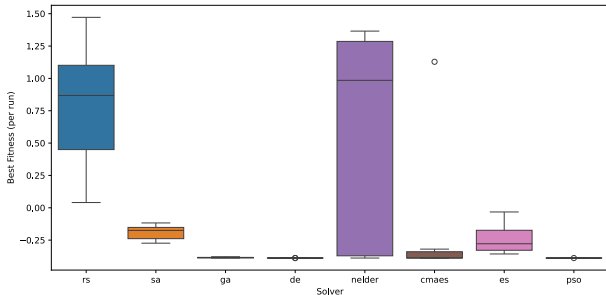


図 14 コードベースベンチマーク②の最適化結果

6. 進捗状況の自己評価と今後の展望

本研究を通じて、LLM と EA を組み合わせた最適化ベンチマーク問題の自動生成基盤を確立することができた。提案手法は、評価基準や問題表現を変更することで、アルゴリズム間の性能差を引き出す問題、高難易度問題、多目的問題、コードベース問題など、多様な性質を持つベンチマークを生成可能である。特に、数式ベースの問題表現に加えて、プログラムコード表現を用いるベンチマーク生成アルゴリズムの基盤を構築しただけでなく、実世界の最適化問題に頻出する多目的最適化問題の生成も実現した。これにより、既存ベンチマークでは捉えにくい最適化アルゴリズムの特性分析を支援できる可能性を示しており、LLM 駆動型ベンチマーク生成の基礎的な実証は達成できたと評価している。

一方、申請当初に予定していた実世界の最適化問題の特性を反映したベンチマーク問題の生成は未達成である。しかし、研究期間終了後にベンチマーク問題の特徴量に着目した自動生成手法の確立に着手しており、本研究で築いた基盤をもとに研究を進展させることができている。

今後は、実世界の最適化問題に頻出する制約条件を含む問題、複数の内部計算過程を持つ問題、外部入力を有する問題など、従来の閉形式数式では表現しにくいベンチマーク問題を生成できるという提案手法の強みを活かした方向性を探求する予定である。さらに、生成したベンチマークの一般公開を通じて、最適化研究全般の発展に寄与する。加えて、LLM 駆動型最適化の発展に向けて、使用する LLM モデルの違いが生成性能に与える影響や、プロンプト設計が生成されるベンチマークの性質に与える影響の解明にも取り組む予定である。

※7. 研究業績はウェブ入力です

参考文献

- [1] M. Hellwig and H.-G. Beyer, "Benchmarking evolutionary algorithms for single objective real-valued constrained optimization - a critical review," *Swarm and Evolutionary Computation*, 44:927-944, 2019.
- [2] R. Tanabe and A. Oyama, "A note on constrained multi-objective optimization benchmark problems," In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1127-1134, 2017.
- [3] E. Meyerson, et al, "Language Model Crossover: Variation through Few-Shot Prompting," *arXiv preprint arXiv:2302.12170*, 2023.
- [4] D. E. Goldberg, *Genetic algorithms*, Addison-Wesley, 1989.
- [5] R. Storn, K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization* 11, pp. 341-359 1997.
- [6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural*

Networks (ICNN '95), vol. 4, pp. 1942–1948, 1995.

- [7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [8] 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.