進捗報告

1 ネルダーミード法

ネルダーミード法 [1] は 1965 年に Nelder らが発表した最適化アルゴリズムである．n+1 個の頂点からなる n 次元の単体を反射，膨張，収縮させながら関数の最小値を探索する．

2 ネルダーミード法を用いたベンチマーク問題の実験

2.1 実験設定

ネルダーミード法の実験設定を表 1 に示す．

表 1: ネルダーミード法の実験設定

変化しない場合の許容回数	2000
変化とみなす閾値	10^{-11}
最大操作数	無限
α	1.0
γ	2.0
ρ	0.5
σ	0.5

$\boldsymbol{x}$ の初期位置は 0 から 1 のランダムな実数とする．

2.1.1 評価関数と制約違反

取り扱うベンチマーク問題では制約違反を考慮する必要があり，今実験ではその許容量を 1.0×10^{-10} とする．また，制約違反の合計値 V を用いて式 1 のようにネルダーミード法の目的関数 F を定義する．

$$F(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & (V < 1.0 \times 10^{-10}) \\ V + 10^7 & (otherwise) \end{cases} \tag{1}$$

ただし，ベンチマークの目的関数を f とする．

2.2 結果

表 2 に実験で得られた最終の目的関数値と制約違反の合計値，図 1 に目的関数値の推移を示す．縦軸は目的関数の値で，横軸は操作回数である．

表 2: 実験結果

目的関数値	制約違反
4200372.158	236.2943965

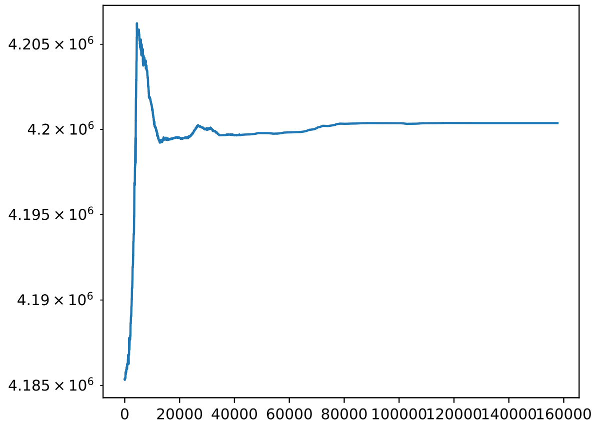

図 1: 目的関数値の推移

表 2 から，制約違反の合計値が許容量を超えていることがわかる．ネルダーミード法のパラメーターを変えるなど予備実験をしたが，許容量を満たす解を見つけることができなかった．120 次元という次元数の高い問題にネルダーミード法が適していないことが原因ではないかと考えた．今実験ではベンチマーク問題の目的関数を式 1 のように定義し，制約違反が許容量を下回る解を見つけられなかったため，図 1 は制約違反のみを最小化する際の目的関数値の推移となる．

3 CMA-ES

CMA-ES[2] は 1996 年に Hansen らが発表した，正規分布の共分散行列を学習する Covariance Matrix Adaptation(CMA) を用いた進化型計算である．変数分離不能，悪スケール，多峰性といった困難さをもつ連続最適化問題に対して効率的な探索ができる．

今回は，python の進化計算ライブラリである DEAP[3] を用いて CMA-ES を実装をした．

4 CMA-ES を用いたベンチマーク問題の実験

4.1 実験設定

表 3 に CMA-ES の実験設定を示す.

表 3: CMA-ES の実験設定

最大世代数	6000
入力次元数	120
1 世代あたりの個体数 λ	1200
μ	600
$\sigma^{(0)}$	0.05
$\mathbf{m}^{(0)}$	(5,...,5)

4.1.1 制約違反

今実験では制約違反の合計値を 1 つの目的関数 V とする．またその許容量は 1.0×10^{-10} とする．CMA-ES では，V を優先して最小化し許容量以下になった後にベンチマーク問題の目的関数 f を最小化するように設定する．

4.2 結果

表 4 に実験で得られた最終の目的関数値と制約違反の合計値，図 2 に目的関数値の推移を示す．縦軸は目的関数の値で，横軸は世代数である．

表 4: 実験結果

目的関数値	制約違反
4790204.092	$9.64638089 \times 10^{-11}$

表 4 より，CMA-ES を用いることで制約違反をクリアする解を見つけることができた．実験においてそのような解は，3005 世代目以降から見つかった．最適化の過程で，制約違反を許容量以下に抑えるまでは順調にできたが，目的関数は満足に最適化ができなかった．最適化の最後の方に目的関数が徐々に下がっていく傾向は見られたため，パラメータをうまく調整することで目的関数をさらに最適化できるのではないかと考えた．

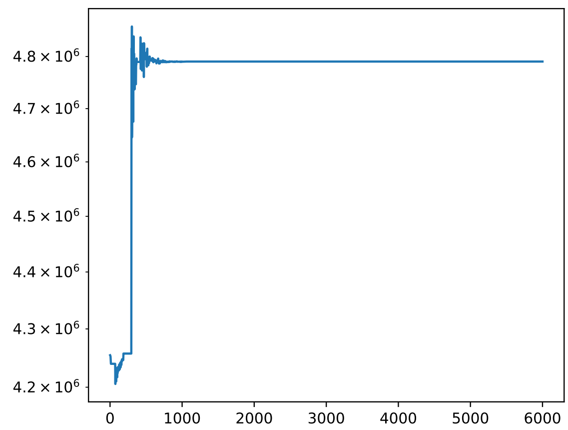

図 2: 目的関数値の推移

5 ネルダーミード法と CMA-ES

ネルダーミード法と CMA-ES を組み合わせた最適化を用いて, ベンチマーク問題について以下の数値実験を行った.

- 実験 1: CMA-ES を用いて最適化をし，制約違反を満たす最初の解をネルダーミード法の初期解として代入する.
- 実験 2: CMA-ES を用いて最適化をし，最終世代の解をネルダーミード法の初期解として代入する.

実験設定は，ネルダーミード法の変化しない場合の許容回数を 4000 とする以外は同じである.

5.1 実験 1

図 3 に結果として得られた目的関数の推移を示す．縦軸は目的関数値，横軸は世代数または操作回数である．3005 世代からネルダーミード法による最適化がされたが，図 3 を見る限り CMA-ES とネルダーミード法にはほとんど違いがない．ここで，図 4 に 3005 世代以降の目的関数値の推移を示す．縦軸は目的関数値，横軸は世代数または操作回数である．図 4 より，CMA-ES の方がネルダーミード法より目的関数を最適化できていることがわかる.

5.2 実験 2

図 5 に結果として得られた目的関数の推移を示す．縦軸は目的関数値，横軸は世代数と操作回数である．図

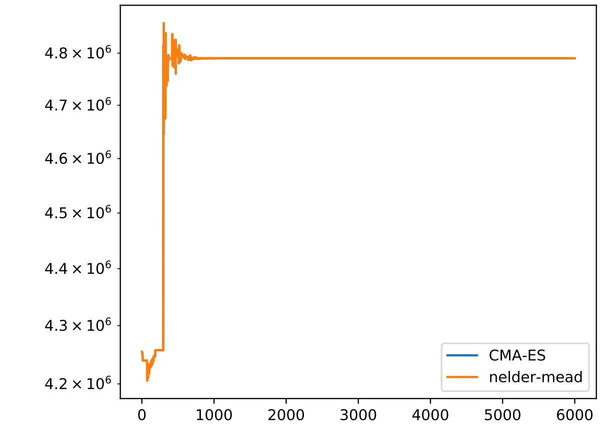

図 3: 目的関数値の推移

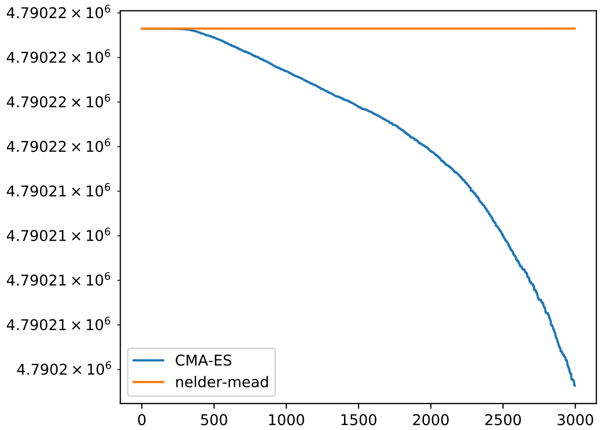

図 4: 目的関数値の推移

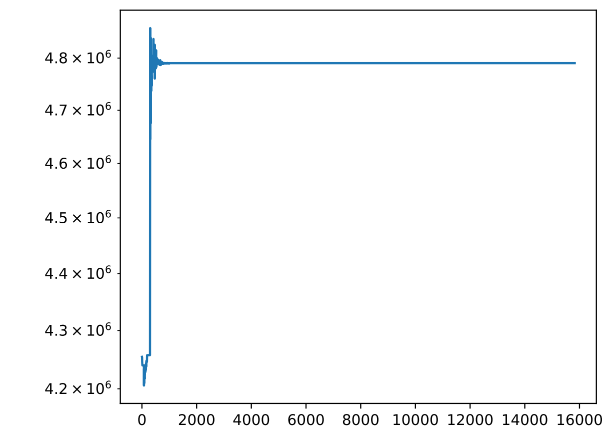

図 5: 目的関数値の推移

5 より，CMA-ES の結果を用いたネルダーミード法は目的関数をほとんど最適化できていないことがわかる．

6 考察と今後の展望

CMA-ES を用いてベンチマーク問題の制約違反を優先して最適化をすると，目的関数値が約 480 万に落ち着いた．既存の見つかっている解の目的関数値が約 400 万であることを考えると，一つの局所解にたどり着き抜け出せなくなっている可能性が高い．したがって，ある程度制約違反を許しながら目的関数の最適化をするという方法が考えられる．

また，ネルダーミード法と CMA-ES を組み合わせた最適化として，ネルダーミード法をした後に CMA-ES をするという最適化や，CMA-ES の数世代ごとにネルダーミード法をするという最適化も考えられる．

さらに CMA-ES のパラメータの調整や，他のバリエーションの CMA-ES も用いてアプローチしたいと考えている．

参考文献

[1] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, Vol. 7, No. 4, pp. 308–313, 01 1965.

[2] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 312–317, 1996.

[3] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, Vol. 13, pp. 2171–2175, jul 2012.