

ベイズ最適化を用いた非線形有限要素法の 解析パラメータ決定とその分散メモリ型並列化

Determination of Analysis Parameters for Nonlinear Finite Element Methods Using Bayesian Optimization and Its Distributed-Memory Parallelization

二村 匠音¹⁾ 柴田 怜²⁾ 遊佐 泰紀³⁾
Takuto Nimura, Rei Shibata, and Yasunori Yusa

¹⁾学 (工) 電気通信大学大学院情報理工学研究科 (〒 182-8585 東京都調布市調布ヶ丘 1-5-1, E-mail: t.nimura@uec.ac.jp)

²⁾学 (工) 電気通信大学大学院情報理工学研究科 (〒 182-8585 東京都調布市調布ヶ丘 1-5-1, E-mail: rei.shibata@uec.ac.jp)

³⁾博 (工) 電気通信大学大学院情報理工学研究科 助教 (〒 182-8585 東京都調布市調布ヶ丘 1-5-1, E-mail: y.yusa@uec.ac.jp)

In nonlinear finite element methods for large-deformation elastic-plastic problems, solutions are often not obtained depending on the setting of analysis parameters. Therefore, in this study, we proposed two parallel search methods with distributed memory computers for analysis parameters that can achieve nonlinear finite element analysis: the synchronous parallel search algorithm and the asynchronous parallel search algorithm using Bayesian optimization. The proposed method was applied to a simple rectangular body torsion analysis and achieved a 54.5% computation time reduction compared to random sampling in 32 parallel executions.

Key Words : Large-deformation Elastic-Plastic Analysis, Bayesian Optimization, Parallel Computing

1. 緒言

構造物や機械の設計に際し、大きな応力や変形を再現する場合には、塑性変形に伴う材料特性の非線形性や、大変形に伴う幾何学的な非線形性を考慮した有限要素法解析 (FEA: Finite Element Analysis)[1] を行う必要がある。しかし、この非線形有限要素法解析では反復解法の収束性が入力パラメータの僅かな違いに大きく影響を受けるため、必ず解析結果が得られる訳ではない。加えて、非線形有限要素法解析では問題に応じて長大な計算時間が必要となるため、解を収束させることのできる入力パラメータの特定に、大きな労力や経験が必要となる場合がある。

また、非線形有限要素法におけるパラメトリックスタディでは、逐次的な解析を行うと非常に長大な時間を要する。そのため、柴田ら [2] や Suwa et al.[3] は、分散型メモリ計算機を用いた静的負荷分散による並列計算手法を提案した。この手法は、入力パラメータが解析結果に与える影響の調査を目的としており、どのパラメータの組み合わせであっても必ず解が得られることを前提とした上で、網羅的な数値解析を行っている。そのため、解析不可能な場合については注目されていない。

非線形有限要素法解析での解析パラメータに対する応答のような、評価するコストの高いブラックボックス関数の最適化において、実験候補となるパラメータを選ぶ手法の一つに、ベイズ最適化 [4] が存在する。ベイズ最適化と非線形有限要素法解析を組み合わせた手法として、Son et al.[5] は設計最適化手法を、Raviolo et al.[6] は、構造物の非破壊検査を目的とした逆解析手法を提案した。また、その中でベイズ最適化は、その他の

最適化手法 (ディープニューラルネットワーク、一般パターン探索、焼きなまし法、遺伝的アルゴリズム) と比較され、関数の評価回数と最適化結果の 2 点で優れた結果を示している。また、ベイズ最適化の欠点として回帰のオーダーが訓練データ数の 3 乗と大きいことが挙げられるが、目的関数評価に長時間を要する問題ではその影響が小さい。そのため、評価するコストの低い目的関数を多数評価する場合に高い探索効率を発揮する他手法に対し、ベイズ最適化は非線形有限要素法解析におけるパラメータ探索に適性の高い手法であると考えられる。

本研究では、大変形弾塑性有限要素法解析における解析可能な入力パラメータの、ベイズ最適化を用いた探索手法を提案する。また、探索の高速化を図り、目的関数の評価において同期的並列と非同期的並列の 2 種類の分散メモリ型並列化を提案し、その並列性能調査を行う。

2. 数値解析手法

(1) 解くべき問題

図-1で定義される境界値問題を考える。物体の領域は Ω で、 Ω の境界は Γ とする。 Γ は変位境界 Γ_u と荷重境界 Γ_t からなり、 $\Gamma_u \cup \Gamma_t = \Gamma$ と $\Gamma_u \cap \Gamma_t = \emptyset$ の関係がある。このとき、仮想仕事式は

$$\int_{\Omega} \delta \tilde{\epsilon}^T \sigma d\Omega - \int_{\Gamma_t} \delta \tilde{u}^T t d\Gamma - \int_{\Omega} \delta \tilde{u}^T b d\Omega = 0 \quad (1)$$

となる。式 (1) において、 σ はコーシー応力、 t は Γ_t 上での面積力、 b は Ω 内での体積力、 $\delta \tilde{u}$ は仮想変位、 $\delta \tilde{\epsilon}$ は $\frac{1}{2}(\partial \delta \tilde{u} / \partial x + (\partial \delta \tilde{u} / \partial x)^T)$ である。仮想変位 $\delta \tilde{u}$ は形状

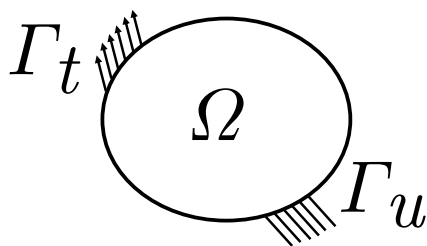


図-1 領域と境界

関数 N と節点仮想変位 δu を用いて,

$$\delta \tilde{u} \approx N \delta u \quad (2)$$

のように近似される. 式 (2) を式 (1) に代入すると,

$$r = 0 \quad (3)$$

となる. ここで,

$$r = f^{\text{int}} - f^{\text{ext}} \quad (4)$$

$$f^{\text{int}} = \int_{\Omega} B^T \sigma d\Omega \quad (5)$$

$$f^{\text{ext}} = \int_{\Gamma_t} N^T t d\Gamma + \int_{\Omega} N^T b d\Omega \quad (6)$$

であり, f^{ext} , f^{int} はそれぞれ全体外力ベクトルと全体内力ベクトルである. また, 式 (5) における B はひずみと節点変位を変換する行列である.

本論文での数値解析において, 大変形モデルとしては Hencky モデル [7] を, 弾塑性モデルとしては von Mises の降伏条件 [7] と等方硬化則 [7] を使用した. Hencky モデルを用いると, 式 (5) に示されるコーシー応力 σ はテンソル表記で次のようにあらわされる.

$$\sigma = \frac{\tau}{\det F} \quad (7)$$

$$\tau = \mathbb{D} : (\varepsilon - \varepsilon^p) \quad (8)$$

$$\varepsilon = \frac{1}{2} \ln(F \cdot F^T) \quad (9)$$

$$F = I + \frac{\partial \tilde{u}}{\partial x} \quad (10)$$

ここで, τ はキルヒホッフ応力, \mathbb{D} は弾性テンソル, ε は Hencky ひずみ, F は変形勾配テンソル, I は恒等テンソルである. Hencky ひずみの塑性部 ε^p は von Mises の降伏条件の下で, 等方硬化則に従う.

(2) 増分法と非線形方程式解法

大変形や弾塑性等の非線形を考慮した有限要素法解析では, 初期状態から目標とする状態までを複数のステップに分割して解析を行う. 各ステップ内では, 非線形方程式の反復解法を用いて連立非線形方程式 (3) を求解することで, 力のつり合いを満たす変位を求める. これを最後のステップまで繰り返すことで, 最終的な変位や応力の分布を求めることができる. そのため, 途中のステップで連立一次方程式が解けない等で計算が止まってしまう場合, あるいは反復しても力の残差が小さくならない場合, 最終的な解を得ることができない.

本論文では非線形方程式の反復解法として, Newton Raphson 法 [7], 修正 Newton 法 [1], Broyden 法 [8], BFGS 法 [1] の 4 つを使用している. 使用する手法により解析可能なステップ数に差が生じるため, この内のどれを使用するかは, 探索パラメータとした.

各ステップにおいて, 式 (3) が解けたとする条件は次式 (11) に示す力の残差

$$\frac{\|f^{\text{ext}} - f^{\text{int}}\|}{\|f^{\text{ext}}\|} \quad (11)$$

が一定値を下回ることとしている.

3. パラメータ探索手法

(1) 探索における目的関数

本論文では, 変位制御 [7] による FEA を行っている. 増分ステップ数が n とした際に, i ステップ目以降の解が得られない場合, 解析の進行の割合は, i/n となる. そこで, 最適化における目的関数は, i/n を出力とする. 探索では $i/n = 1$ を目的とし, 入力パラメータを設定する.

(2) ベイズ最適化

ベイズ最適化では, 訓練データのガウス過程回帰による統計モデルの構築, パラメータ空間内において統計モデルに基づく獲得関数の値が最大となる点の選択, その点での目的関数の評価, 結果を訓練データに追加, を繰り返すことにより最適化を行う.

ガウス過程回帰におけるカーネル関数 $k(\mathbf{x}, \mathbf{x}')$ には, ARD (Automatic Relevance Determination) 二乗指数カーネル,

$$k(\mathbf{x}, \mathbf{x}') = \exp \left(- \sum_{i=1}^d \frac{|\mathbf{x}^{(i)} - \mathbf{x}'^{(i)}|^2}{l_i^2} \right) \quad (12)$$

を使用した. ここで, \mathbf{x} は入力パラメータをまとめたベクトル, d は入力パラメータの次元, l はハイパーパラメータである. ハイパーパラメータは, 回帰を 10 回実行するごとに, タイプ II 最尤推定に基づき BFGS 法によって最適化を行っている.

獲得関数には, 期待改善量獲得関数 EI (Expected of Improvement),

$$\alpha_{\text{EI}}(\mathbf{x}) = (\mu_n(\mathbf{x}) - y_n^*) \Phi \left(\frac{y_n^* - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})} \right) + \sigma_n(\mathbf{x}) \phi \left(\frac{y_n^* - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})} \right) \quad (13)$$

を使用した. ここで, y_n^* は訓練データにおける出力の最大値, $\mu_n(\mathbf{x})$, $\sigma_n(\mathbf{x})$ はそれぞれ, 回帰された正規分布の平均と分散, Φ , ϕ はそれぞれ, 標準正規分布の上側累積確率関数と確率密度関数である.

目的関数評価の並列化のためには, 複数のパラメータによる FEA を並列実行する必要がある. これには, 先行研究 [3] を参考にし, 分散メモリ間の通信規格である, MPI (Message Passing Interface) を使用した. 加えて, 他プロセスで実行中のパラメータの扱いに関して, 嘘つき法 [4] を使用し, ガウス過程回帰による予測値 $\mu_n(\mathbf{x})$ を仮の出力値として, 一時的に訓練データに加えている.

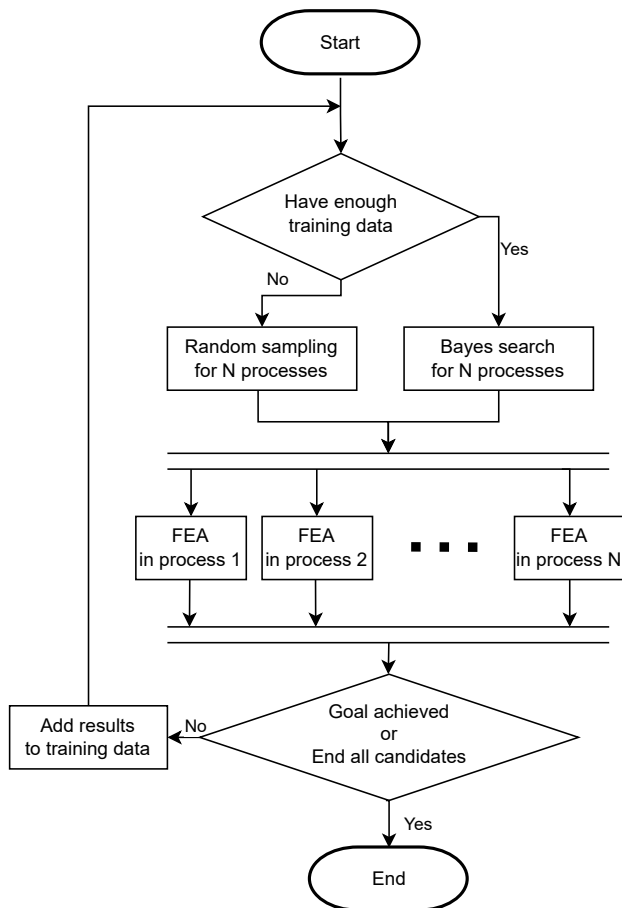


図-2 同期的並列探索アルゴリズムのフローチャート

また、非線形方程式解法の種類や要素形状の種類など、数値で表すことのできないパラメータについては、 2^n 個の種類に対して n 個の 0 または 1 をとる探索パラメータを追加し、その組み合わせで表現している。これは、本来存在しない大小関係の設定を避けるためである。

(3) 同期的並列探索アルゴリズム

同期的並列探索アルゴリズムのフローチャートを図-2に示す。同期型並列探索アルゴリズムは、探索と FEA の両方を行う 1 つの代表プロセスと、FEA のみを行うその他のプロセスで構成される。まず、代表プロセスが並列プロセス数分の実行タスクを探索する。次に、得られた実行タスクを各プロセスに割り振り、それに基づく FEA を並列に実施する。そして、全プロセスの FEA が終了後に代表プロセスがその結果を確認し、目標を達成してない場合は再度並列プロセス数分の実行タスクを探索する。

(4) 非同期型並列探索アルゴリズム

非同期的並列探索アルゴリズムは、探索のみを行うマスタープロセスと、目的関数の評価である FEA のみを行うワーカープロセスから構成される、マスター・ワーカー方式で実装されている。分散メモリ型計算機におけるマスターとワーカーの関係を、図-3に示す。非同期的

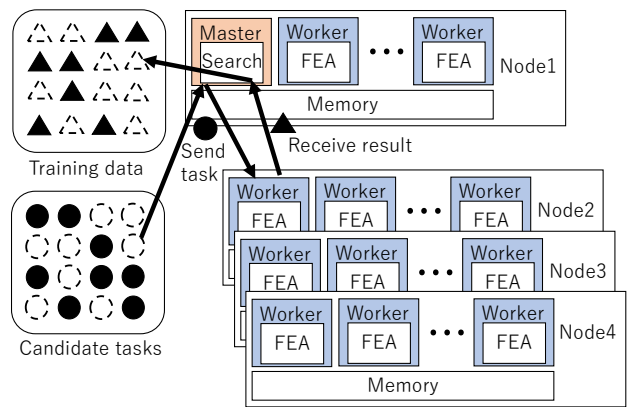


図-3 分散メモリ型計算機における非同期的並列探索アルゴリズムの模式図

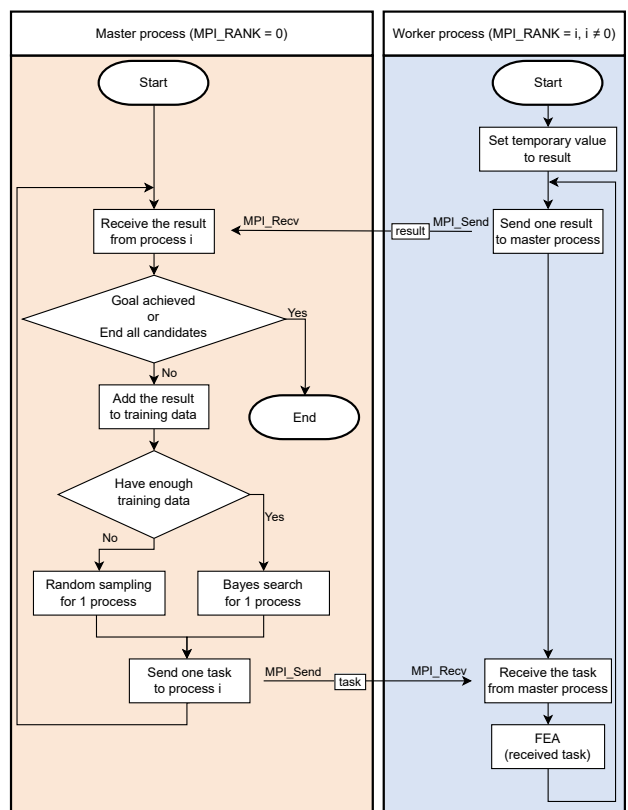


図-4 非同期的並列探索アルゴリズムにおけるマスタープロセスとワーカープロセスのフローチャート

並列探索アルゴリズムにおけるマスタープロセスとワーカープロセスのフローチャートを図-4に示す。マスターが次に実行するタスクを探索し、それをワーカーに送信 (MPI.Send) することで割り当てている。そして、その実行結果をマスターがワーカーから受信 (MPI.Recv) し、終了判定をしたのちに訓練データに追加する。そして、再びマスターが次のタスクを探索し、ワーカーに送信 (MPI.Send) するということを繰り返すことで、目的関数の最適化を行っている。

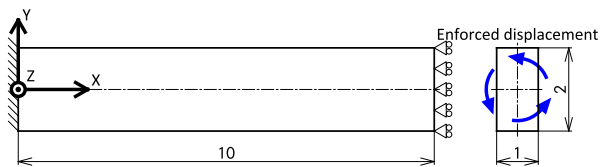


図-5 寸法と境界条件 (mm)

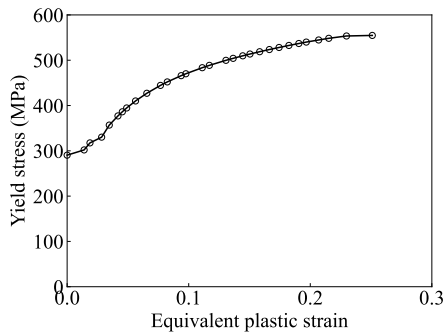


図-6 降伏応力-相当塑性ひずみ線図

4. 数値例の解析条件

提案手法について並列数の影響を調査した。本節ではその条件を示す。試行回数を増やすため、平均 10^4 程度の節点で構成される単純な形状であり、かつ増分ステップ数が数 10 程度の、1 回当たりの FEA に要する計算時間の短い解析を対象としている。

(1) FEA における収束条件

各ステップでの非線形方程式解法における収束判定は、式 (11) で定義される力の残差が 10^{-3} を下回ることとし、最大反復回数は 1,000 回としている。また、テンソルの指数・対数は級数を用いて計算している。級数の収束判定は項の値が 10^{-4} を下回ることとし、最大反復回数は 1,000 回としている。

(2) 寸法・境界条件

寸法と境界条件を図-5に示す。直方体の一方の端面を完全拘束し、もう一方の端面の長手方向変位を拘束した上で、軸にそってねじるように強制変位を与えている。目標とするねじり角度は 60° とした。材料は SGV410 を想定し、Young 率は 203 GPa、Poisson 比は 0.3 とし、解析には実験結果 [9] に基づく図-6に示す降伏応力-相当塑性ひずみ線図を入力している。入力パラメータとしては、各軸方向の要素数、非線形方程式解法、要素形状、増分ステップ数を対象とした。それぞれの概要を表-1に示す。組み合わせは総計 11,520 通りであり、その内の 0.45% が最後まで解析が可能なパラメータであった。最後まで解析できた場合の解析結果の一例を図-7、図-8に示す。図-7は変形後メッシュに相当応力を示すコンターを重ねたものであり、図-8は横軸をねじり角、縦軸をトルクとしたグラフである。非線形性がある解析となっていることが確認できる。

(3) 使用する計算機

本研究で使⽤した分散メモリ型計算機の性能を表-2に示す。計算機は 16 コアの CPU を搭載したノード 4 台

表-1 探索パラメータ

パラメータの種類	ケース数	備考
軸方向要素数	5	1,716 ~ 113,246 節点
厚さ方向要素数	6	
幅方向要素数	6	
非線形方程式解法	4	Newton-Raphson 法, 修正 Newton 法, Broyden 法, BFGS 法
要素形状	2	六面体一次要素, 四面体二次要素
増分ステップ数	8	3 ~ 10 deg/step

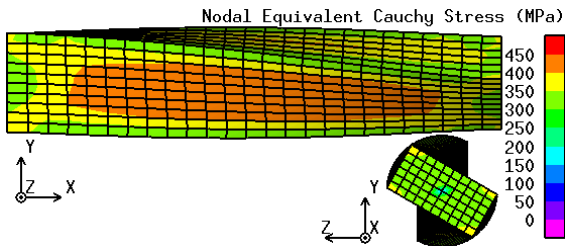


図-7 変形後形状・相当応力分布の一例

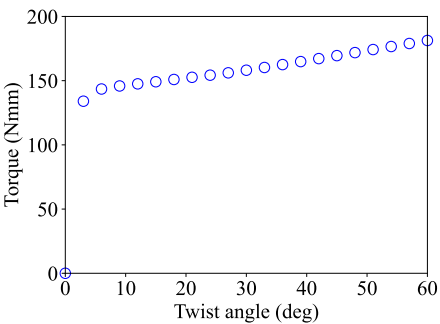


図-8 トルク-ねじり角度線図の一例

表-2 使用した分散メモリ型計算機の性能

CPU	16 cores (Intel Core i9-12900) × 4 nodes = 64 cores
Memory	64 GB (DDR4-3200) × 4 nodes = 256 GB
Storage	2 TB (SSD) × 4 nodes = 8 TB
Network	10 GbE

で構成されており、最大 64 並列での解析が可能である。

5. ストロングスケーリングテスト

(1) テストにおける設定

並列数は 2, 4, 8, 16, 32, 64 の 6 通りで実施した。なお、 $16 \times n$ 並列までは n 個のノードを使用して探索が行われる。また、非同期的並列探索アルゴリズムにおいて、2 並列 (1 マスター, 1 ワーカー) で実施した際の目標達成までの計算時間を、逐次での目標達成まで

の計算時間として処理している。初期の訓練データとしてランダムサンプリングによる 50 ケースの結果を与えており、実行中での次のタスクの選択はベイズ最適化によってのみ行っている。また、初期訓練データは、50 ケースの内の全てで目標を達成していなかったもののみを採用している。初期訓練データは 4 通りを用意し、そのそれぞれでストロングスケーリングテストを実施した。高速化率・並列化効率については、それぞれの初期訓練データを用いた場合で計算した。つまり、 i 番目の初期訓練データによる並列数 p での目標達成までの計算時間を T_i^p とすると、その高速化率 r_i^p と並列化効率 e_i^p は、

$$r_i^p = \frac{T_i^2}{T_i^p} \quad (14)$$

$$e_i^p = \frac{T_i^2}{p \times T_i^p} \times 100\% \quad (15)$$

となる。そのため以降の評価では、目標達成までの計算時間の平均を用いて計算した高速化率・並列化効率ではなく、各高速化率・並列化効率の平均 ($\sum_{i=1}^4 r_i^p / 4$, $\sum_{i=1}^4 e_i^p / 4$) を示している。

(2) 結果

横軸をマスターを含む並列数、縦軸を目標達成までの計算時間の高速化率としたグラフを作成すると、図-9となる。また、横軸をマスターを含む並列数、縦軸を並列化効率としたグラフを作成すると、図-10となる。なお、両グラフともにプロットは平均値を、エラーバーは最大値と最小値を示している。

図-9から、非同期並列探索では、高速化率の上昇は 32 並列で止まり、64 並列では 27% 低下している。また最大値は 32 並列での 22.2 であった。同期並列探索では、高速化率の上昇が 16 並列で止まり、最大値は並列での 10.8 であった。また、並列数が増加すると非同期並列探索の方が同期並列探索よりも高速化率が高くなる傾向にあることが確認できる。これは、並列数が増えたと非同期並列探索においてマスターが 1 プロセス使用することの影響が小さくなるとともに、同期並列探索で同期待ちの時間の影響が大きくなるためであると考えられる。

図-10から、非同期並列探索では、8 から 32 並列の間では平均 67.0% と比較的高く、64 並列では 25.3% と大きく下がった。原因としては、嘘つき法によって仮に設定される訓練データが多いことにより、探索に悪影響を与えていることが考えられる。また、プログラムの逐次実行時間の内、並列化ができる割合を α 、並列台数を p とすると、アムダールの法則 [10] に基づき理想的な並列化効率 e^p は、

$$e^p = \frac{1}{\alpha \left(\frac{1}{p} - 1 \right) + 1} \frac{1}{p} \quad (16)$$

となる。目的関数評価は並列化をしているが、ベイズ最適化によるタスクの選択は並列化ができないため、 $\alpha < 1$ である。そのため、マスタープロセスのアイドル状態がなくなる程度にまで並列数が増加すると、並列数の増加に伴って並列化効率が低下すると考えられる。

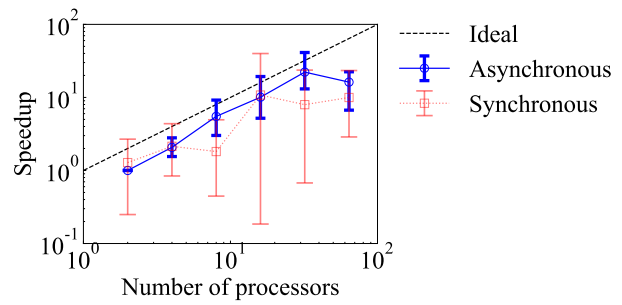


図-9 並列数に対する高速化率

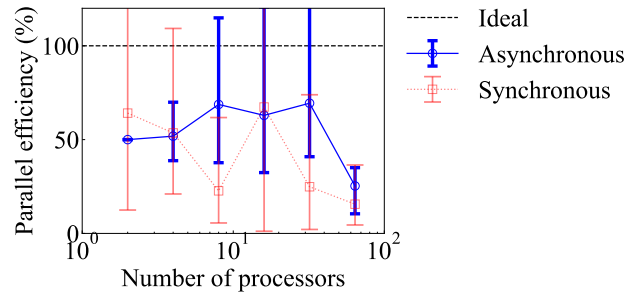


図-10 並列数に対する並列化効率

表-3 非同期並列探索における、ランダムサンプリングとの比較

Number of processors	Method	Number of evaluations	Time (s)
32	Bayes	93.3	270.0
	Random	150.5	593.7
64	Bayes	184.4	331.1
	Random	195.1	359.1

6. ランダムサンプリングとの比較

比較的優れていた非同期並列探索について、32 並列と 64 並列でランダムサンプリングとの比較を実施した。初期の訓練データとしてランダムサンプリングによる 50 ケースの結果を与え、それを 20 通りに変化させて実施した。目標達成までに必要であった計算時間と目的関数の評価回数の、それぞれの平均を表-3に示す。目的関数の評価回数とは、最初に最後まで解析を達成したパラメータが、何番目に選ばれたのかを示している。32 並列では、ベイズ最適化を使用した方が、ランダムサンプリングよりも必要な目的関数の評価回数が 38.0% 減少し、計算時間は 54.5% 減少した。64 並列では、ベイズ最適化を使用した方が、ランダムサンプリングよりも必要な目的関数の評価回数が 5.48% 減少し、計算時間は 7.80% 減少した。ベイズ最適化を使用した場合に、64 並列よりも 32 並列の方が優れた結果を示した原因としては、前述のような嘘つき法による探索への悪影響が考えられる。

7. 結言

本研究では、非線形有限要素法解析において解析可能な入力パラメータの探索を、分散メモリ型計算機を用いて並列に行う手法として、並列ベイズ最適化を用いた同期的並列探索アルゴリズム、非同期的並列探索アルゴリズムの2手法を提案した。

また、単純な直方体のねじり解析に対して提案手法を適用し、ストロングスケーリングテストを行った。その結果、非同期的並列では32並列までは高速化率が上昇したが、64並列では低下した。また、同期的並列では8並列まで高速化率が上昇し、その後は大きく変化しなかった。加えて、非同期的並列について、32並列と64並列でランダムサンプリングと比較した。その結果、ベイズ最適化を用いた場合の方がランダムサンプリングを用いた場合と比較し、目標達成までの目的関数の評価回数と計算時間が減少した。特に32並列ではランダムサンプリングに対し、平均して45.5%の計算時間で最後まで解析可能なパラメータを探索可能であった。しかし、64並列では、ランダムサンプリングに対し計算時間の削減が7.80%となった。この原因としては、嘘つき法により仮に設定される訓練データが多いことによる、探索への悪影響が考えられる。

今後の展望は、より多くの並列数でも並列化効率の低下を防ぐため、獲得関数の吟味や、他の最適化手法との比較を行うことである。また、途中まで同一となるような解析結果を共有することによる、目的関数評価の効率化も検討している。

謝辞: 本研究は日本学術振興会科研費 JP22K14167 の助成を受けたものである。

参考文献

[1] Yusa, Y., Miyauchi, S., Okada, H. (2021). Performance investigation of quasi-newton-based parallel nonlinear fem for large-deformation elastic-plastic analysis over 100 thousand degrees of freedom. *Mechanical Engineering Journal*, Vol. 8, Issue 3, Article 21-00053. doi:10.1299/mej.21-00053

[2] 柴田 怜, 諏訪 浩貴, 遊佐 泰紀 (2023). 大変形弾塑性問題の並列パラメトリック解析に向けた検討. 日本機械学会関東支部総会講演会講演論文集, Vol. 2023.29, Article 17H01. doi:10.1299/jsmekanto.2023.29.17H01

[3] Suwa, H., Yusa, Y., Kuboki, T. (2023). Parallel parametric analysis approach based on an s-version FEM for fracture mechanics analysis in designing hole positions. *Mechanical Engineering Journal*, Vol. 10, Issue 3, Article 22-00462. doi:10.1299/mej.22-00462

[4] 今村秀明, 松井孝太 (2023). ベイズ最適化 適応の実験計画の基礎と実践. 近代科学社.

[5] Son, D., Liimatainen, V., Sitti, M. (2021). Machine learning-based and experimentally validated optimal adhesive fibril designs. *Small*, Vol. 17, Issue 39, Article 2102867. doi:10.1002/smll.202102867

[6] Raviolo, D., Civera, M., Zanotti Fragonara, L. (2023). A comparative analysis of optimization algorithms for finite element model updating on numerical and experimental benchmarks. *Buildings*, Vol. 13, Issue 12, Article 3010. doi:10.3390/buildings13123010

[7] de Souza Neto, E., Perić, D., Owen, D. R. J., 著, 寺田賢二郎監訳 (2012). 非線形有限要素法 弾塑性解析の理論と実践. 森北出版.

[8] Yusa, Y., Okada, H., Yamada, T., Yoshimura, S. (2018). Scalable parallel elastic-plastic finite element analysis using a quasi-newton method with a balancing domain decomposition preconditioner. *Computational Mechanics*, Vol. 62, Issue 6, pp. 1563–1581. doi:10.1007/s00466-018-1579-4

[9] 日本溶接協会 原子力研究委員会 MFD 小委員会 (2017, (参照 2022-12-2)). 多軸応力下の疲労および延性破壊データベース公開資料. <http://www-it.jwes.or.jp/fatigue_db/>.

[10] 片桐孝洋 (2013). スパコンプログラミング入門 並列処理と MPI の基礎. 東京大学出版会.