

# グラフニューラルネットワークを用いた 高次元関数の次元分割による最適化手法の提案

2121057 清 恵人

コンピュータシステム研究室 指導教員: 中野 秀洋 教授

## 1 研究背景と研究目的

### 1.1 研究背景

#### 1.1.1 最適化

現実世界には大規模な環境におけるロボット制御, ニューラルネットワークアーキテクチャの探索など多くの多次元な関数の問題がある. しかしニューラルネットワークにおける最適化などでは計算コストがとて高コストで, メタヒューリスティックなどの進化計算では評価値が得るのに時間がかかりすぎるなどの問題がある.[1]

#### 1.1.2 Surrogate Model

目的関数がブラックボックスになっており定式化できないときに, Surrogate model (代理モデル) を用意することで目的関数の近似解を予測する. Surrogate model の導入は, 近似の代理モデルを提供し, 候補解が目的関数の評価値を効率良く得ることを可能にする. Surrogate model は Neural Network や RBF などのネットワーク構成から近似の目的関数を得る.[2]

#### 1.1.3 Graph Neural Network

グラフニューラルネットワークはグラフ構造を学習するニューラルネットワークでノード分類, グラフ分類などのタスクに用いられる. 従来の深層学習手法をグラフニューラルネットワークとして扱う研究がなされており, 画像や音声自然言語など実世界における様々なグラフデータに適用することができ幅広い応用が期待されている.[3] [4] これらのタスクには Graph Convolutional Network が用いられる.[5] 通常グラフニューラルネットワークはノード間の関係が似ているもの同士が近くに集まりノードの特徴量が異なるものは遠くへ配置される.

#### 1.1.4 従来研究の課題点

Surrogate model を用いても高次元な関数を探索するのは困難である. そこで高次元な関数を低次元ごとに分けて探索する方が探索はより探索性能は向上する. しかし変数の依存関係があるものを分割すると性能低下が低下する. よって高次元な関数を低次元ごとに正しく分割する指標がない.

### 1.2 研究目的

ブラックボックスになっているような(高次元かつ複雑で定式化できない) 関数をグラフニューラルネットワークを用いて低次元に正しく分割することで効率的な最適解への探索を行う.

## 2 目的達成の手法

### 2.1 提案手法

目的達成の全体の手法を以下に図 1 として示す. Training data の生成, Surrogate model, Graph Neural Network, 遺伝的アルゴリズムの順に行っていく.

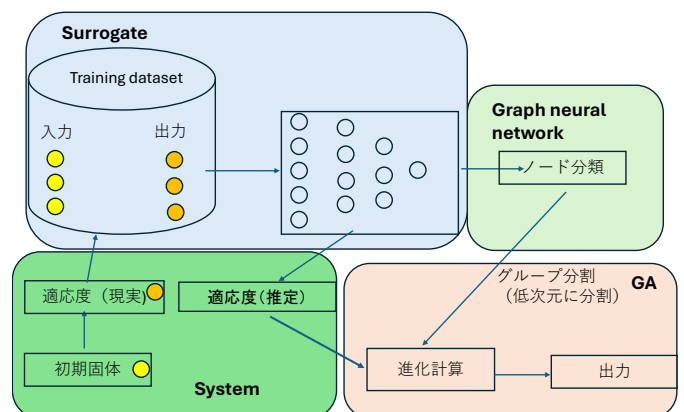


図 1: Graph Neural Network を用いた次元削減の流れ

### 2.1.1 Training data の生成

遺伝的アルゴリズムの初期個体を入力, 適合度を出力として訓練データを生成する. 訓練データの出力は実際の適応度として算出する.

### 2.1.2 Surrogate Model

定式化できない目的関数の近似解を得る代理モデルを作成する. Training data を訓練データとして Neural Network のモデルを構築する. ここでニューラルネットワークをグラフニューラルネットワークへの適用しやすくするために枝刈りを行う. 枝刈りはエポックごとに重みの削減する割合を高めていく. これにより重要な重みのみだけがエッジとして保存される.[6]

### 2.1.3 Graph Neural Network

高次元な関数を低次元に分割して探索を行うには, 変数間同士の依存関係を考慮して分割するべきである. ここで変数間同士の依存関係を見ながら分割する手法として Graph Neural Network を用いる. 代理モデルの Neural Network 構造を Graph Neural Network として扱う. 具体的には, ニューラルネットワークの入力を変数になっているので入力層のノードの分類を行う.

### 2.1.4 遺伝的アルゴリズム

分類されたノードから同じクラスのもの同士で遺伝的アルゴリズムを用いて探索を行う. これにより高次元な関数を低次元で探索することができる. ここでの適応度は Surrogate model からの近似解で最適化を行なう.

## 2.2 研究の進捗

目的関数は Rosenbrock(50 次元)+Dixon(50 次元)+Rosenbrock(50 次元) の計 150 次元の関数を用いた. 分割には 3 分割, 50 次元ずつ分割される方が望ましくなる. 式 (1)Rosenbrock function, 式 (2)Dixon function として以下に示す.

$$f(x) = \sum_{i=1}^d [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (1)$$

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^d i(2x_i^2 - x_{i-1})^2 \quad (2)$$

式 (1) は  $x_{i+1}$  と  $x_i$  が乗算されるので 1 つ次の変数に依存関係がある. 式 (2) は 1 つ前の変数に依存がある.

### 2.2.1 実験結果

表 1: 実験結果 (分割の比較)

| 分割なし     | 分割あり     |
|----------|----------|
| 2.19e+09 | 1.63e+09 |

分割する方が精度はよりよくなることがわかる. また本来ニューラルネットワークにグラフニューラルネットワークを適用するのは難しいが枝刈りを加えることによってノードの分類が可能になり正しい分割を行っている.

## 3 研究計画

入力層のノード分類は, 教師あり学習でノード分類を行っている. しかし実際は依存関係がわからないものをノード分類したいので Graph Neural Network を教師なし学習のクラスタリングを用いる. Graph Neural Network の clustering はノード間の重みの平均がそれぞれ等しくなるように分類する.[7] しかし Neural Network の構造をグラフ構造として適用しているため, うまく分類ができない. そこで attention を導入した clustering を用いてより認識精度向上を目指す.[8] またより高次元な目的関数への適用.

## 参考文献

- [1] Y. Liu, et, "Infill Criterion Ensemble in Multi-Objective Evolutionary Algorithm for Mixed-Variable Problems." 2024
- [2] X. Wang et. "A Graph Neural Network Assisted Evolutionary Algorithm for Expensive Multi-Objective Optimization." 2024
- [3] Wu, et. "A comprehensive survey on graph neural networks." 2020
- [4] Michaël Defferrard, et. "Convolutional neural networks on graphs with fast localized spectral filtering." 2016
- [5] Kipf, et. "Semi-Supervised Classification with Graph Convolutional Networks." 2016
- [6] 中野 秀洋, 土岐 健太郎 "中間層の逐次的追加による効率的な深層学習手法" 2024
- [7] Bianchi, F. M. . "Simplifying clustering with graph neural networks." 2023
- [8] Mao, Q et. "Clustering driven multi-hop graph attention network for speaker Diarization." 2023