

進捗報告

1 今週やったこと

- 論文読んでまとめた

2 A Survey of Automatic Parameter Tuning Methods for Metaheuristics

2.1 導入

最適化アプローチ

1. 厳密法

長所：最適性保証

短所：NP 困難では膨大な計算コスト

2. ヒューリスティック法

戦略やルールで問題に特化したアルゴリズム

長所：NP 困難な問題で顕著な成功

短所：問題依存性（適用の制限）・局所最適解

3. メタヒューリスティック法

高レベルの方法論・一般的なアルゴリズムのテンプレート・自然から着想を得た

長所：幅広い問題を解決

2.1.1 最適化のノー・フリー・ランチ（NFL）定理

全ての最適化問題に対応する万能アルゴリズムはない

2.1.2 パラメータ設定問題

問題ごとにパラメータを適切に設定する必要がある

1. パラメータチューニング（オフラインチューニング）

事前にチューニングで得たパラメータを利用・実行中は変更されない

利点：汎用性

欠点：多くの設定ごとに実行するので時間がかかる

2. パラメータ制御（オンラインチューニング）

戦略に従い実行中に変更

利点：ー

欠点：非普遍性・変更するパラメータの適切な理解が必要

2.2 自動パラメータ調整

開発動機

- 時間コストの削減・性能の向上
- アルゴリズムの評価、比較のためパラメータの最適性の影響を緩和
- パラメータが与える影響の知識の必要性を排除

2.2.1 分類

1. パラメータの種類

1) 数値パラメータ：実数や整数

2) カテゴリパラメータ：メカニズムや演算子

2. 1) ゼネラリスト：広い範囲の問題に対してよいパフォーマンスのパラメータ設定

2) スペシャリスト：1つの問題に対してのみ優れた性能のパラメータ設定

2.2.2 チューニング方法の分類

主要カテゴリ

1. 単純な生成・評価法

生成段階で候補となる設定を生成、評価段階で評価して最適な設定を見つける

2. 反復的生成・評価法

少数の設定を生成して、最も優れたものをみつけ、反復的に生成する新しい設定の指針とする。

3. 高レベルの生成・評価方法

高レベル生成機構では既存のチューナーや探索手法から少数精鋭の設定を生成。評価段階では慎重に評価する

サブカテゴリ

1. 評価の繰り返し

確率的な目的関数最適化問題では複数回評価した平均値が最も分かりやすい方法

2. F-Racing

評価した統計的に劣る設定を段階的に排除し、有望な候補に計算を集中。繰り返し評価より効率的

3. インテンシフィケーション（強化）

評価途中で候補が暫定設定より劣るなら排除し、そうでなければ次の問題で評価する。全ての問題で評価されると新しい暫定設定となる

4. シャープニング

少ないテスト数で評価が始まり、将来性のあるパラメータはテスト数が2倍になる。素早く探索できる

5. アダプティブキャッピング

有望でない設定の実行を中断して計算量を削減

2.3 単純生成評価法

2.3.1 ブルートフォースアプローチ

FFD や他の DOE 技術を用いてパラメータ設定を生成し、学習インスタンスに対して同じ回数の実行を行うことで、各設定候補の性能を推定

欠点：各設定に計算資源を均等に割り当てるので非効率的・実行回数を決める基準がない

2.3.2 F レース

構成候補のうち少なくとも1つが他の構成候補とパフォーマンス指標の点で有意に異なるかを調べる。ノンパラメトリック Friedman 検定で、違いがないという帰無仮説が棄却された場合、最高ランクの構成と各構成との1対比較が実行され、劣った性能の候補が排除される。残った2つの候補のうち良いほうを結果とする

1. FFD/F-Race

FFD によって候補を生成

欠点：パラメータ基準が必要、候補数がパラメータ数に指数関数的

2. RSD/F-Race

RSD はパラメータ空間に定義された確率モデル（一様分布）に従ってサンプリングされる

利点：数値パラメータ水準を定義する必要がない、任意の数の候補を生成

2.4 反復的生成評価法

2.4.1 実験設計に基づくチューニング

DOE：統計的手法で分析できるように実験を計画する方法

できること：構成候補の生成、探索空間の有望な領域を特定、パラメータ値が変化した場合の影響を分析、パラメータの重要度をスクリーニング、ランク付け

1. CALIBRA

DOE と局所探索を組み合わせた反復調整アルゴリズム。実験解析を用いて探索空間を狭くし、次の実験を開始。各反復について、狭くなった範囲の境界と中点を用いて、田口の分数実験計画を行い局所最適基準を満たすまで繰り返す。

欠点：田口の分数実験計画の制約でパラメータは5つまで、パラメータ間の相互作用効果を考慮できない

2.4.2 数値最適化に基づくチューニング

アルゴリズムのパラメータがすべて数値的なら、数値最適化手法と評価手法で解ける

1. 二次近似による境界最適化（BOBYQA）と繰り返し評価の組み合わせ

2. メッシュ適応的直接探索（MADS）と繰り返し評価と F レースの組み合わせ

欠点：カテゴリカルパラメータを扱うことができない

2.4.3 ヒューリスティック検索に基づく方法

1. 反復 F-Race

確率モデルに基づいて設定候補のセットを生成。標準的な F-Race を実行。生き残った候補で次の反復の確率モデルを更新

利点：候補の数を増やすことなく効率的に探索

欠点：計算時間を短縮できない、チューニング予算が少なすぎると解が貧弱になる

2. メタ EA

個体はパラメータ設定. 各設定の評価方法による性能尺度が, (メタ) フィットネスに対応.

利点: パラメータ空間内で大域的な最適値に到達する可能性がある. 任意の時点で停止し, 最良の設定を解として返すことができる

欠点: パラメータ空間が大きい場合には現実的ではない. カテゴリカルなパラメータを扱うことができない

3. ParamILS

まずデフォルト構成とランダム生成された設定に対して, 一度に 1 つのパラメータ値のみを変更 (一交換近傍探索) して, 最適なパラメータ設定を探索

1) パラメータ値をランダムに変更

2) 反復改善プロセスの実行

3) どの設定から探索を継続するか決定

利点: 探索プロセスのほぼ全ての時点で停止できる

欠点: パラメータごとに離散化して候補の近傍を定義する必要がある

4. HORA

任意の n 個の問題を選択し, 問題ごとに応答曲面法 (RSM) で最適なパラメータ設定を特定

1) 最も好ましい設定の近傍に新しい候補を動的に生成

2) 設定候補をレース法で評価し, 統計的証拠に基づいて悪い設定を破棄

利点: GA と SA より計算コストがはるかに低い

欠点: アルゴリズム性能が複雑な場合やマルチモーダルな場合は, 単純な RSM は誤った領域を探索する

2.4.4 モデルベースの最適化アプローチ

アルゴリズム性能のパラメータ設定への依存性を応答曲面でモデル化し, 良いパラメータ設定を探索. 探索と搾取の間のトレードオフに対処できる.

1. SPO

LHD で初期設定を生成し, 最適な設定が初期の暫定候補として選択. パラメータと対応する性能測定値に基づいて, Kriging モデルと呼ばれる応答曲面モデルを構築. 新しい設計が生成され, 構築されたモデルを使用してテスト. 最も高い EI を持つ設

定を現在の現職と比較. 新たに評価された点で, モデルを更新.

欠点: カテゴリカルなパラメータを扱うことができない. 単一の問題でのみアルゴリズムの性能を最適化

2. SMAC

Kriging の代わりに, ランダムフォレストを用いる. 単純なマルチスタート局所探索を行い, EI が最大となる設定を見つけ, 得られた設定を次の反復の有望な候補に反映.

利点: カテゴリカルな入力データに対して優れた性能を発揮. 予測の不確実性を定量化できる. 一連の問題に対してアルゴリズムの性能を最適化できる

SMAC, TB-SPO, GGA, ParamILS に対して有意に改善

2.5 高レベル生成評価法

2.5.1 ポストセレクションの仕組み

1. エリート認定フェーズ: 高品質な構成, あるいは精鋭化候補の構成をいくつか特定

2. エリート選択フェーズ: エリート構成を徹底的に評価し, 最適なものを慎重に選択

既存のチューナーをジェネレーターとして使用し, エリート候補の構成を作成・特定し, その中から慎重に評価した上で最適なものを選択することで, 良いパラメータ設定を見つけるという新しいアイデア

利点: 多数のエリート構成を容易に提供することができる

欠点: エリート認定の段階でエリート候補の多様性を維持できない

2.6 今後の研究展望

2.6.1 パラメータチューニング問題の解決効率を向上させる

1. テストする候補構成の総数を減らす: 先進的なサンプリング戦略

2. 構成の評価にかかる平均コストを削減する: 効率的にテストするための評価方法

2.6.2 ベンチマークテスト式と使いやすいアルゴリズムチューニングツールボックスの確立

標準化されたチューニング問題のベンチマークセット

新しいチューニング手法の簡単な利用と統合を可能にするオープンなアルゴリズムパラメータチューニングツールボックス

目的：チューニングアルゴリズムの実証研究を容易にする

2.6.3 多目的チューニングアプローチの研究

複数の性能指標を同時に最適化する

3 考察

Pytorch で使えそうなチューナーを探して, 実際にどの程度時間がかかるかを見たい.

4 今後の予定

- 5 クラス識別器を 2 つ用いた 10 クラス識別とクラスの分割