

ソフトウェア自動チューニングにおける 反復 1 次元 d-Spline 探索法の複数か所同時適用

矢島 雄河 森戸 建太郎 藤家 空太郎 楊 暄 藤井 昭宏 田中 輝雄

工学院大学

1. はじめに

ソフトウェア自動チューニング (AT : software auto-tuning) とは, ソフトウェアの性能を自動的にチューニングする手法である[1][2]. 実行時 AT を用いることで, 実行環境による違いや実行時まで確定しない要素などを踏まえ, 最適化を行うことができる. 実行時にチューニングを行う場合, チューニング自体のコストが対象プログラムの性能に影響を与える. また, チューニング中に悪い性能パラメタで実行されてしまうこともある. これらの悪影響は, チューニングが長引くほど大きくなる. したがって, 最適値の探索を高速化することは, AT の研究目的のひとつとなっている.

AT では単一の性能パラメタに限らず, 複数の性能パラメタを同時にチューニングすることもある. 性能パラメタを複数にすることは, より効果的なチューニングを可能とする. 一方で, 複数の性能パラメタ探索はそれらの組合せに対して行われるため, 探索され得る性能パラメタの組合せの総数は, 性能パラメタの数に応じて指数的に増加する. これはチューニングを長引かせる原因となり, プログラムの性能低下につながる. これらに対処するため, 対象プログラムのチューニング範囲を複数か所に分割し, それぞれのチューニングを同時並行で行う方法を検討する. この方法では, 探索され得る性能パラメタの組合せ総数が大幅に減少し, 実際に探索される数も減少する可能性がある.

本研究では, 複数の性能パラメタを持つプログラムに対し, AT 手法を複数か所同時に適用することを目的とし, それにより最適値をより高速に特定することが可能であることを示す.

2. 実行時自動チューニング

2.1 実行時 AT の基本構造

図 1 に, 実行時 AT の基本構造を示す. 実行時 AT では, ユーザプログラム内の反復処理をチューニングの対象とする. チューニング処理は, ユーザプログラムとは別に用意された AT 機構が行う.

AT 機構は, チューニング対象部の 1 反復ごとに, 性能パラメタの取り得る値から実行パラメタを選択する. チューニング対象部の 1 反復が終わると, 選択された実行パラメタでの性能指標が AT 機構に渡される. 性能指標の例には, チューニング対象部の実行時間などがある. AT 機構は渡された性能指標を用いて評価を行い, 次の実行パラメタを選択する.

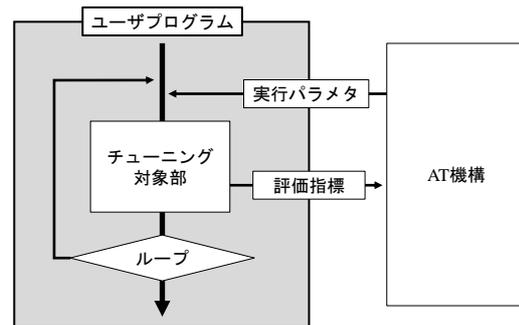


図 1 実行時 AT の基本構造

2.2 反復 1 次元 d-Spline 探索法

性能パラメタが取り得る値の集合から最適値を探索するアルゴリズムとして, 複数の実測値から多項式近似を用いて調べる標本点推定がある[2]. 性能パラメタが複数ある場合, 最適値は各性能パラメタの値の組合せによって決まる. したがって, チューニング時に探索され得る値の数は, 性能パラメタの数に応じて指数的に増加する.

実測の回数を抑えるため, 我々は d-Spline と呼ぶスプライン関数による, 標本点推定を用いている. d-Spline は離散的な近似関数であり, 計算量が少なく, 実測値の動きに柔軟に追随することができる[3]. この d-Spline を用いた複数性能パラメタのチューニングとして, 我々は反復 1 次元 d-Spline 探索法による性能パラメタ推定を提案している[4]. これは複数の性能パラメタが取り得る値から成る多次元空間内で, 以下の手順を繰り返す探索法である.

- (1) 方向探索
- (2) その方向での 1 次元 d-Spline 探索

反復 1 次元 d-Spline 探索法では, multi-steps search を用いている. この手法では, まず 1 つの性能パラメタごと独立に変化する方向に探索する. その後, 各性能パラメタの組合せである方向を順次加えていき, さらなる探索を行う. 同時に変化させる性能パラメタを制限することで探索標本点数を減らすことができ, 探索の効率が向上する. また複数の性能パラメタを同時に変化させることで, 性能パラメタ同士に相関がある場合に対応している.

3. 探索法の複数か所同時適用

3.1 探索法の同時適用方法

ここでは複数の性能パラメタを持つソフトウェアが, いくつかの範囲に分割できる場合を考える. ここで前提として, 分割されたそれぞれの範囲に必要な性能パラメタが, 他の範囲の性能パラメタとの間に相関を持たず, 独立している状況を想定する. 分割した範囲ごと, 複数か所を同時並行にチューニングするモデルとして, チュ

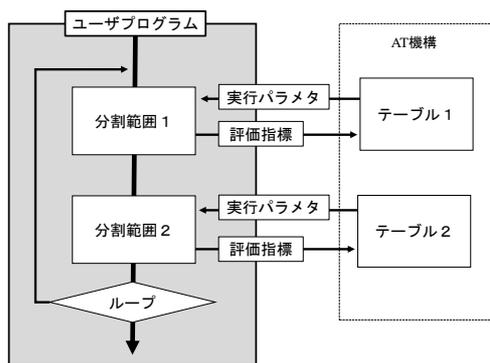


図2 分割した範囲ごとにチューニングするモデル

チューニング対象部を2つに分割した例を図2に示す。このモデルでは、複数のチューニングテーブルを用意する。テーブルはそれぞれ1つのチューニング対象部に対応し、各対象部の実行パラメータと性能評価結果の組を保持する。性能パラメータの決定と性能評価は、各対象部の実行ごとに行われる。このようにすることで、対象部それぞれの反復ごとにチューニングが個別に行うことが可能となる。

それぞれの対象部は独立しているため、異なる対象部の性能パラメータの最適値探索に影響を与えない。したがって、ある性能パラメータ集合の探索が終了した後に他の対象部の性能パラメータを変更しても、最初の対象部の性能パラメータは最適値であり続ける。

3.2 分割による効果

前節のように分割することにより、探索が必要なパラメータの組合せ総数が減少する。

例として n 種類の性能パラメータそれぞれが取り得る値が m 通りあるとする。このとき、これらをまとめて探索する場合はすべての組合せを調べることとなり、 m^n 通りとなる。ここで性能パラメータが $n/2$ 種類のパラメータからなる独立性能パラメータ集合2つから成る場合、それぞれの性能パラメータ集合に必要な最大探索数は $m^{n/2}$ であり、全体でも $2m^{n/2}$ 通りである。したがって、少ない探索回数で探索を完了することができる。

このことから、各探索部における性能パラメータの種類を減らすことは、multi-steps search の効果をさらに向上させる。あらかじめ関連の無い性能パラメータ同士の組合せを除外することで、同時に変化させる性能パラメータを増加させた際の探索における無駄を減少させることができる。

4. 実験

4.1 実験環境

使用したコンピュータは、CPUがIntel製 Xeon E5-2650 v3 (2.30GHz)、OSがUbuntu (20.04.2 LTS)である。

また、計測対象のプログラムとAT機構はC言語によって作成し、コンパイラはgcc(9.3.0)、コンパイル時の最適化オプションには-O3を指定した。

4.2 対象プログラムと性能パラメータ

対象となるプログラムは行数と列数がともに1000要素である密行列同士の、行列行列積である。

ここでは独立3種類ずつの部分集合が2つ、合計6種類

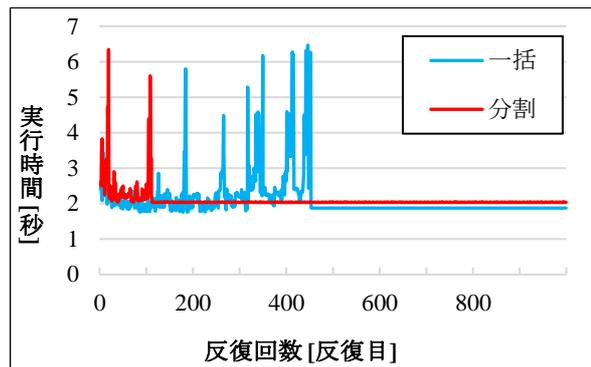


図3 行列行列積の実行時間

の性能パラメータを用意し、一括でチューニングを行った場合と分割してチューニングを行った場合の結果を比較した。独立したチューニング箇所のうち、一方にはブロッキング、もう一方にはループアンローリングを適用した。ブロッキングの性能パラメータは10から200まで10間隔のブロッキングサイズ、ループアンローリングは1, 2, 4のアンローリング段数である。1反復あたり合計2回の行列行列積を、2000反復行った。

4.3 実験結果と考察

1反復ごとの実行時間を、図3に示す。

ここで横軸は反復回数、縦軸が実行時間である。一括の場合は探索の終了までに640反復かかっているのに対し、分割の場合は127反復で探索を完了することができた。以上のことから、性能への影響が独立している性能パラメータを分割してチューニングすることにより、効率よく探索を行うことができることが確かめられた。

5. おわりに

本研究では、性能パラメータが独立した範囲ごとに分割してチューニングするモデルを示し、探索を行った。分割することは、探索を必要とする組合せ総数を減少させ、multi-steps search の無駄を減少させることで、効率の良いチューニングを可能にする。

今後の課題としては、性能への影響が独立した性能パラメータを判別する手法の確立があげられる。また、性能への影響のみが独立しており、性能評価を個別に行えない場合の適用手法も必要である。これらの確立により、ATのさらなる高効率化が可能であると考えられる。

謝辞 本研究は JHPCN 拠点の支援(課題番号:jh210019-NAH)及び JSPS 科研費 JP18K11340 の助成により実施した。

参考文献

- [1] 今村俊幸, 荻田武史, 尾崎克久, 片桐孝洋, 須田礼仁, 高橋大介, 滝沢寛之, 中島研吾. ソフトウェア自動チューニング—科学技術計算のためのコード最適化技術. 第1版, 森北出版, 2021.
- [2] 片桐孝洋. ソフトウェア自動チューニング—数値計算ソフトウェアへの適用とその可能性. 慧文社, 2004.
- [3] Teruo Tanaka, Ryo Otsuka, Akihiro Fujii, Takahiro Katagiri, Toshiyuki Imamura. Implementation of d-Spline based Incremental Performance Parameter Estimation Method with ppOpen-AT. Scientific Programming 2014, IOS Press, 22 4, 2014, pp.299-307.
- [4] 望月大義, 藤井昭宏, 田中輝雄. ソフトウェア自動チューニングにおける複数同時性能パラメータ探索手法の提案と評価. 情報処理学会論文誌 ACS 第11巻2号, 2018, pp.1-16.