

ソフトウェア自動チューニングにおける反復次元探索による複数性能パラメータ推定法の実装と評価

関直人[†] 望月大義[†] 范谷瑛[†] 藤井昭宏[†] 田中輝雄[†] 片桐孝洋^{††}

工学院大学[†] 名古屋大学^{††}

1. はじめに

ユーザがプログラムのチューニングを手動で行うことは難しい。そこで、自動チューニングの研究が進められてきた。これまでは、性能の優劣を決める性能パラメータが複数になると、チューニングに必要なメモリ量と計算量が增大するという課題があった。チューニングにおけるメモリや処理時間は極力減らすことが重要である。本論文ではそれを意識しチューニング対象となるユーザプログラムへの影響が軽微な実装を実現することを目的とする。

本論文ではメモリ量、計算量が少ない反復次元探索による複数性能パラメータ推定法[1]の実装を行った。実装は、自動的にチューニング機能付きのプログラムを生成する自動チューニング基盤 ppOpen-AT[2]に対して行った。評価は、全性能パラメータのそれぞれの値とそのときに計測された実行時間のデータを用いて行った。

2. 標本点逐次追加型性能パラメータ推定法

標本点逐次追加型性能パラメータ推定法は、幾つかの標本点（計測を行った性能パラメータの組）から推定を始め、推定値が安定するまで標本点を選択し追加する。標本点が追加されるたび、近似関数を更新し、この更新された関数を使用して最適なパラメータを推定する。

2.1 一次元 d-Spline

推定に用いる近似関数は実測データに柔軟に追随し、少ない標本点でも解が得られ、計算量が少ないことが要求される。それらを満たす近似関数 d-Spline について説明する。

d-Spline は、離散的な近似関数 $f(x)$ であり、 n 個の離散点 x_j 上の値 $f_j = f(x_j), 1 \leq j \leq n$ で表現する。次に、実測データを $y_i (1 \leq i \leq N)$ とする。 N は実測したデータの個数である。そして、評価

関数 $\min(\|y - Ef\|^2 + \alpha^2 \|Df\|^2)$ から f を決定する。 E は y と f の距離を、 D は近似関数の滑らかさを表す。 α は実測データへの追従の度合いを表す。

2.2 複数性能パラメータ推定法

今回実装を行った反復次元探索による複数性能パラメータ推定法の手順を以下に示す。

1. 探索を開始する初期標本点を選択
2. 2a, 2b を最適値が連続するまで反復
 - 2a 探索方向の決定（各軸の方向のみ）
 - 2b 一次元 d-Spline を用いて最適値推定
3. 3a, 3b を最適値が連続するまで反復
 - 3a 探索方向の決定（斜め方向を含む全方向）
 - 3b（上記と同様）

手順 2a, 3a について図 1 を用いて説明する。基準となる標本点の周囲の点を計測する。それらの計測結果より最小値となる点と標本点を結ぶ直線を選択する。2a, 3a と分けるのは、まず 2a で探索方向を制限して計測を行う点を減らすためであり、次元が増えるとさらに効果が出る。

2b は、選択した直線上で一次元探索を行う。

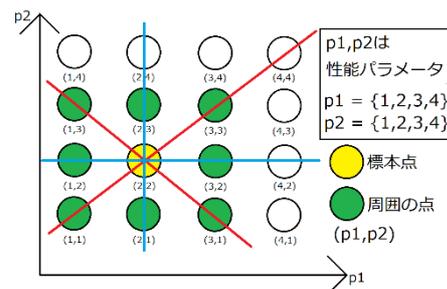


図 1 探索を行う直線とその決定

3. 自動チューニング基盤 ppOpen-AT

ppOpen-AT は、ユーザがプログラムにディレクティブを記述するだけで自動チューニング機能付きのプログラムを生成する自動チューニング基盤である。

3.1 反復次元探索の実装の特徴

プログラム実行時にチューニングをする場合、チューニングに必要なメモリや処理時間は最小限に抑えることが重要である。そこで、我々の

Implementation and Evaluation of Multi-performance Parameter Estimation Method by Iteration of One-dimensional Search in Software Automatic Tuning

Naoto Seki[†], Masayoshi Mochizuki[†], Guuing Fan[†], Akihiro Fujii[†], Teruo Tanaka[†], Takahiro Katagiri^{††}

[†]Kogakuin University, ^{††}Nagoya University

実装では、チューニングの際必要になるデータを、実測を行った標本点の分のみ確保することでメモリを軽微に抑えた。さらに、処理時間を短くするため次元 d-Spline を利用した。以上の取り組みによりチューニングに必要なメモリ量、計算量はともに性能パラメータの組み合わせの数を N として $O(N)$ となり、ユーザプログラムへの影響がなるべく少ない実装となっている。

3.2 ディレクティブの追加

ppOpen-AT を利用するときのディレクティブの記述イメージを図 2 に示す。

```
#pragma OAT dynamic variable (para1,para2) region start
#pragma OAT name MyFunc
#pragma OAT fitting 2stepspline2
#pragma OAT varied(para1) from 0.1 to 1.0 step 0.1
#pragma OAT varied(para2) from 1.0 to 5.0 step 0.2
Function(para1,para2);//チューニング対象
#pragma OAT dynamic variable (para1,para2) region end
```

図 2 ディレクティブの記述イメージ

図中 3 行目の“fitting”で推定方法を選択する。二次元パラメータ推定の“fitting”に 2stepspline2、三次元パラメータ推定の“fitting”に 2stepspline3 を新規のディレクティブとして追加した。

4. 評価実験

二次元データ a, b[3]と三次元データに対し評価実験を行った。チューニングに必要なメモリ量は二次元データがそれぞれ約 132[KB]、三次元データが約 9[KB]であった。チューニング時間は、いずれのデータについても 1[ms]程度であった。

4.1 二次元データに対する評価

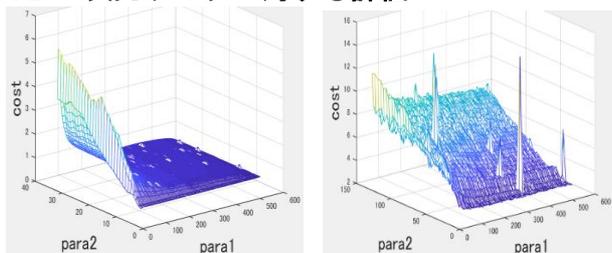


図 3 データ a

図 4 データ b

表 1 二次元データの推定結果

		データ a	データ b
内数	最小値を推定した個数	2083	4083
	相対誤差が 10%以内の個数	2013	13
上 2 項目の合計		4096	4096
平均標本点数		31.3	41.4

データの等高線図を図 3 と図 4 に示す。性能パラメータはともに 4096(32×128)通りの値を取る。データ a, b で 4096 通りすべての点を初期点としてそれぞれ推定を行った。結果を表 1 に示す。

データ a, b は 4096 通りの実行時間のうち最小値を推定した個数が半分以上で、すべての推定は最小値との相対誤差が 10% 以内に収まっている。

この結果より、二次元の実装が正しく行えていると言える。

4.2 三次元データに対する評価

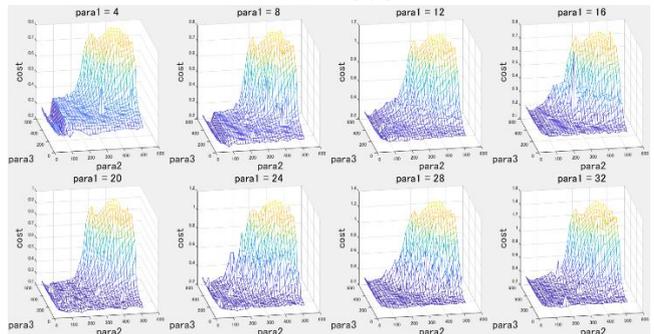


図 5 x(para1)軸で区切った等高線図

表 2 三次元データの推定結果

推定実行時間[s]	0.19	0.20	0.21	0.22	0.23	0.24
推定度数 (合計 8192)	805	1627	4838	48	1	873
平均標本点数	81.49					

性能パラメータは 8192(8×32×32)通りの値を取る。8 通りの性能パラメータ para1 で区切ったときのそれぞれの等高線図を図 5 に示す。この 8192 通りすべての点を初期点としてそれぞれ推定を行った。結果を表 2 に示す。

最小値 0.19 を推定した個数は 8192 個のうち 9.83%であった。一方 8192 個のうち 88.7%は上位 3 つの値(0.19 から 0.21)を推定できており、三次元についても問題なく動作していると言える。

5. おわりに

本研究では、ユーザプログラムへの影響が軽微なチューニングを実現するため、反復次元探索の手法を自動チューニング基盤の ppOpen-AT に実装を行った。評価実験では、推定に必要なメモリは約 100[KB]、処理時間も約 1[ms]と、チューニングで必要になるメモリ、処理時間がともに抑えられていることを確認した。

謝辞 本研究の一部は JSPS 科学研究費 16H02823 の助成を受けて行われた。評価に用いるデータは東京工業大学の佐藤幸紀特任講師から頂いた。

参考文献

- [1] M. Mochizuki, A. Fujii, T. Tanaka, “Fast Multidimensional Performance Parameter Estimation with Multiple One-dimensional d-Spline Parameter Search”, in Proc. IPDPSW2017, pp. 1426-1433, 2017.
- [2] T. Katagiri, S. Ohshima, M. Matsumoto, “Auto-Tuning on NUMA and Many-Core Environments with an FDM Code”, in Proc. IPDPSW2017, pp. 1399-1407, 2017.
- [3] 幸朋矢, 佐藤幸紀, 遠藤敏夫, Polyhedral コンパイラを用いたタイリングパラメータ自動調整ツールのメニーコア環境での評価, SWoPP2017, Vol. 160-34, pp. 1-8, 2017.