2006－HPC－107（33）
2006／8／1

汎用的なソフトウェア自動チューニング機構のための実験計画法の応用の検討

小　谷　和　正[†]　須　田　礼　仁[†]

本稿では行列の積計算におけるループアンローリングを実例にとり，品質工学で用いられる実験計画法の技術を汎用的な手法としてソフトウェアの自動チューニングへ応用することを検討する．すなわち，直交多項式モデルや直交表などをソフトウェアの性能評価関数の推定に導入することで，性能パラメータの効果ごとの成分分解や推定精度の向上，実験実行回数の縮小を目指す．また構成したモデルの評価方法について，自動チューニングの目的である最適パラメータ選択という視点に即した方法を検討する．

Consideration of Applicating Design of Experiments to General Purpose Automatic Tuning Facility for Softwares

Kazumasa Kotani [†] and Reiji Suda [†]

In this paper, with an example of an loop unrolling optimization in a matrix-matrix multiply function, we consider to applicate some tecniques of Design of Experiments as general-purpose methods to automatic performance tuning of softwares. Adopting orthogonal polynomial model and orthogonal array on estimation of performance evaluation function, we try to decompose the effects of tuning parameters, improve the accuracy of estimation, and reduce the number of test executions. And we consider evaluation method that is based on the demand in performance tuning as making choice of optimal parameter configuration.

1. はじめに

近年，ATLAS や FFTW などのように自動チューニング機構を備えたソフトウェアが開発されており，自動チューニング機構を直接組み込んだ個々のソフトウェアについては多く研究されてきた．しかしながら，そういった機構をプログラム中に手軽に組み込むことのできるような汎用的な枠組みについての研究は少ない．自動チューニング機構のプログラムへの自動的な組み込みを実現した例として片桐らの開発した ABCLibScript[1) が挙げられるが，チューニングパラメータに対する性能評価関数の推定には改善の余地が残されている．

自動チューニングによるパラメータ値の決定において，厳密な意味での最適値を求めるならばパラメータ値同士の全組合せを試すことになるが，実際にはより少ない実験により性能変化を測定した上で（最適ではなくとも）良いパラメータを決定するという程度で十分な場合もある．当然ながら推定の精度と実験実行の回数はトレードオフの関係にあるが，少ない実験回数でも必要な情報ができる限り精確に得られるように実験するパラメータの組合せを工夫することは可能である．

そこで本稿では，品質工学の分野において汎用的に用いられている実験計画法の技術を導入し，ソフトウェアのチューニングにおける可能性や問題点などについて検討を行う．以下 2 章では実験計画法の概要について述べ，3 章では行列積プログラムのループアンローリングによる性能変化を例にとり，実際に実験条件の設定や性能評価モデルのパラメータ推定を行う．4 章では構成したモデル関数の評価方法についての検討を行う．最後に，5 章で本稿をまとめる．

2. 実験計画法

実験計画法は初め 1920 年代に R.A. フィッシャーにより考案された手法で，その後様々な研究改良が行われてきた．現在では様々な参考書が出版されており，以下では文献 2)3)4) について本稿に必要な部分をまとめたものである．

† 東京大学大学院 情報理工学系研究科 コンピュータ科学専攻
Dept. of Computer Science, Graduate School of Information Science and Technology, University of Tokyo

2.1 概　　要

実験で得られる測定データは，実験において制御した要因と，測定誤差や背景環境など自由に制御できない要因の両方に影響されて変動しており，そのためデータの変動を各要因ごとに分割する必要がある．それについてのフィッシャーによる実験計画の基本的な考え方は次のようなものである．

（1）　反復：要因の効果かそれ以外の要因によるばらつきかを確かめる

（2）　局所管理：調べる要因以外の条件を可能な限り一定にする

（3）　無作為化：以上でも制御できない要因の影響を除くため実験順序などの条件を無作為化する

これらを実現するために用いられる技術について以下で説明する．

2.2 直交多項式

データの変動を各要因の効果ごとに分割するための数式モデルとして，直交多項式系によるモデルが用いられる．

要因 A の値（水準☆と呼ぶ）が等間隔に

$$A_i = A_1 + (i-1)h$$

と設定されている場合，ここでの直交多項式系は次のようなものである

$$P_0(A) = 1$$
$$P_1(A) = (A - \bar{A})$$
$$P_2(A) = (A - \bar{A})^2 - \frac{a^2-1}{12}h^2$$
$$\vdots$$
$$P_n(A) = P_{n-1}(A)P_1(A) - \frac{(n-1)^2\{a^2-(n-1)^2\}h^2P_{n-2}(A)}{4\{4(n-1)^2-1\}}$$

これらは以下の性質を満たす．

$$\sum_{i=1}^{a} P_s(A_i)P_t(A_i) = 0 \quad (s \neq t)$$

A の各水準 A_k で r 個のデータ $x_{k1}\ x_{k2} \ldots x_{kr}$ を得たとして，A_k 内の平均を $\bar{x_k}$，全体での平均を $\bar{x}$ とする．このデータを 3 次までの直交多項式を線形結合したモデル

$$x = b_0 + b_1P_1(A) + b_2P_2(A) + b_3P_3(B) + \varepsilon$$

として最小二乗推定を行うと，各係数は上記の性質から

$$b_0 = \bar{x}$$
$$b_k = \frac{\sum_{j=1}^{n} P_k(A_j)\bar{x}_j}{\sum_{j=1}^{n}(P_k(A_j))^2} \qquad (k = 1, 2, \ldots)$$

と計算できる．なお，これはモデルの次数を変化させても同じ係数は計算し直す必要がないという特徴もある．

また，ここで求めているのは要因の主効果と呼ばれ，他の要因条件については平均した変動である．これに対してある要因の条件によって別の要因の変動が異なる場合，それらの要因の間には交互作用があると言う．

要因が複数ある場合のモデルは，主効果については各要因ごとにデータを整理することで前述と同様にモデルを構成することができ，それらを加算したものが全体でのモデルとなる．交互作用の効果についても，効果ごとにモデルを構成し，項を追加することができる (詳細は文献を参照のこと).

例えば，要因 A の s 次と B の t 次との交互作用は以下の式でモデル化され，

$$I_{st} = b_{st}Ps(A)P_t(B)$$

最小二乗法により係数は

$$b_{st} = \frac{\sum_{i=1}^{n}\left\{P_t(B_j)\sum_{j=1}^{n}P_s(A_j)\bar{x}_{ij}\right\}}{\sum_{i=1}^{n}\left\{(P_t(B_j))^2\sum_{j=1}^{n}(P_s(A_j))^2\right\}}$$

となる．ただし $\bar{x}_{ij}$ は A の水準が A_i，B の水準が B_j であるデータの平均値である．

2.3 直　交　表

それぞれ要因の各水準について，その全ての組合せを調査することによって要因の効果を計算する方法を全因子計画（full factorial design）と呼ぶ．それに対して**表 1** のような直交表と呼ばれる表に従って条件を設定した実験を行う方法を直交表計画（orthogonal design）と呼ぶ．ここで N, i, j, k は要因として，$1 \ldots 16$ は実験番号．表の a,b,c,d は各要因についてそれぞれの水準値でおきかえる．

直交表とは，任意の 2 列についてその水準の全ての組合せが同数回ずつ出現するという条件を満たすよ

表 1　L16 直交表 （N, i, j, k は要因）
Table 1　L16 Orthogonal Array

	N	i	j	k		N	i	j	k
1	a	a	a	a	9	a	c	d	b
2	b	b	b	b	10	b	d	c	a
3	c	c	c	c	11	c	a	b	d
4	d	d	d	d	12	d	b	a	c
5	a	b	c	d	13	a	d	b	c
6	b	a	d	c	14	b	c	a	d
7	c	d	a	b	15	c	b	d	a
8	d	c	b	a	16	d	a	c	b

☆ 水準とは要因を質的に分類，もしくは量的に変化させた条件のこと

```
!ABCLib$ install unroll (I) region start
!ABCLib$ name MyMatMul
!ABCLib$ varied (I,J,K) from 1 to 5
      do I=1, N
        do J=1, N
          da1 =  A(I,J)
          do K=1, N
           dc = C(K, J)
           da1 = da1 + B(I,K) * dc
          enddo
          A(I,J) = da1
        enddo
      enddo
!ABCLib$ install unroll (I) region end
```

図 1　ABCLibScript によるアンロール指定つきの行列積プログラム

Fig. 1　The matarix-matrix multiplier with unroll directives of ABCLibScript

うな表で，各要因の影響を調べることができる配置になっている．

全因子計画は当然上の条件を満たすが，全ての主効果と交互作用を調べるのでなければより実験回数が少なくなるような直交表を用いればよい．表 1 は各効果の主効果を全て調べることができる配置で，全 $4^4 = 256$ 通りの組合せに対して実験数を 16 個に抑えている．また小さな直交表から始め，より多くの要因が解析できるように直交条件を満足するように表を拡張してゆくということも可能である．

3. 実　　験

3.1 実験条件と実行環境

図 1 に示す行列積プログラムに対して，以下の条件でループアンローリングを施し，実行時間の最適化を行った．ソースコードは Fortran90 と ABCLibScript により記述されており，アンロール後のコードはプリプロセッサにより自動生成されたものを用いた．

また，対象とする正方行列のサイズは $128 \leq N \leq 1024$，アンロール段数は各ループ $2 \leq i, j, k \leq 5$ とする．その上で，性能評価値が N, i, j, k の関数であるとして各効果が最大 3 次の直交多項式モデル（2.2 節）を構成することを目標とする．

なお，ここで行列サイズ N はチューニングパラメータとして自由に変更できる変数ではなく，ユーザからの入力から得られる情報であることに注意したい．この場合のチューニング目標は，入力の行列サイズ N に対して実行時間が最小になる i, j, k を求めることにある．

実行環境は以下に示す．

CPU:	Intel Pentium M 740
L1 キャッシュ:	64KBytes
L2 キャッシュ:	2MBytes
物理メモリ:	1024MBytes
OS:	FreeBSD 6.1-R
コンパイラ:	GNU Fortran 95(gcc4)
最適化オプション:	-O3

3.2 検討課題

本節では，試行の段階で確認されたいくつかの問題点について述べる．

まず，$N \times N$ 行列同士の積算一回の実行時間をそのまま性能評価値とすることを考えた場合，実行時間が N^3 にほぼ比例して増加するために，他の要因の影響を計るのが困難になる等の問題が発生する．N は自由に変更できる値ではないので，N の主効果は本質的に重要ではない．

そこで以下では，行列積計算の最内ループ内での処理一回にかかる平均実行時間，つまりを行列積全体の実行時間を N^3 で割った値を性能評価値とする．

モデルパラメータの推定において，アンロール段数 i, j, k についてはそれぞれ $2 \leq i, j, k \leq 5$ の全点を標本点にとるが，N については少ない標本点から補間することを考える．ここで以下のような性質が確認された．

- N, i, j, j の主効果による分散が全分散の大部分（9 割前後）占めている
- **図 2** で見る通り，$N = 256, 512$ 付近を境に何らかの性質変化があるように見える
- またそれ以外にも N による小さなばらつきは発生する
- 図 2 には載せていないが，N が 256 の倍数では実行時間が周辺の 2 倍程度に増加し，一種の特異点となっている

3.3 実験方針

前述の問題は既知とした上で 対策を施した実験を計画する．3.1 節の条件に加え，以下を設定する．

- プログラム性能の評価値は行列積一回の実行時間を N^3 で割った値とする
- 簡単のため，モデルには主効果の項のみ含める
- N =128〜256, 256〜512, 512〜1024 と三つのブロックに分け，別々にモデルを構成する
- モデルに含める各効果は 3 次までの直交多項式とし，次数はブロック毎に AIC を最小化するように決定する

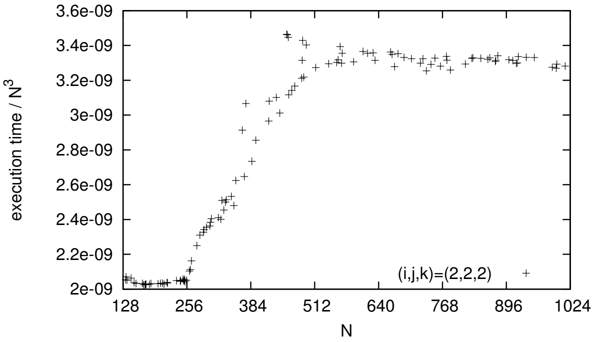

図 2　N による性能値変化の様子
Fig. 2　The appearance of the effect of N

- 256 の倍数となる N は標本点として避ける

なお線形モデルについては最小二乗法が最尤推定であり，データ数を n，自由パラメータ数を m，残差平方和を Q として AIC（赤池情報量規準）は以下の式で計算できる．詳細は文献 5) を参照のこと．

$$AIC = n\left\{1 + \log(2\pi\frac{Q}{n})\right\} + 2m$$

行列サイズ N による細かいばらつきを平均化するために，各ブロックでは以下の手順でモデルを構成する．

(1)　ブロックを 4 等分し，それぞれの区間の中央を N の水準とする
(2)　区間内の N についてランダムに点（ここでは 6 点）を標本点にとり，それらの平均値を各水準におけるデータとする
(3)　2.2 節に従ってモデルパラメータを計算する
(4)　AIC によりモデル次数の決定を行う
(5)　モデルの性格上区間の両端まではモデルに含まれないので，ブロックを 3 等分した上で両端を含む 4 点の推定値で改めて直交多項式フィッティングを行う．このとき両端の値は上で構成したモデルで外挿する

上記をまず全因子計画としてそれぞれの要因の各水準は全組合せについて実験と測定を行い，そのデータのうち表 1 で表される直交表に載る部分データを直交表計画を行った場合のデータとみなした．

3.4　実 験 結 果

まず直交計画の場合について，各効果の次数に対する AIC の変化をグラフとして**図 3** に示す．横軸は N, i, j, k の次数の組合せ (d_N, d_i, d_j, d_k) を一次元に射影したもので，$\{(d_N - 1) * 27 + (d_i - 1) * 9 + (d_j - 1) * 3 + (d_k - 1)\}$ の値になっている．

この場合 AIC が最小値となる次数は各ブロック順に (d_N, d_i, d_j, d_k) =$(1, 2, 1, 2)$，$(1, 1, 1, 1)$，$(1, 1, 1, 1)$

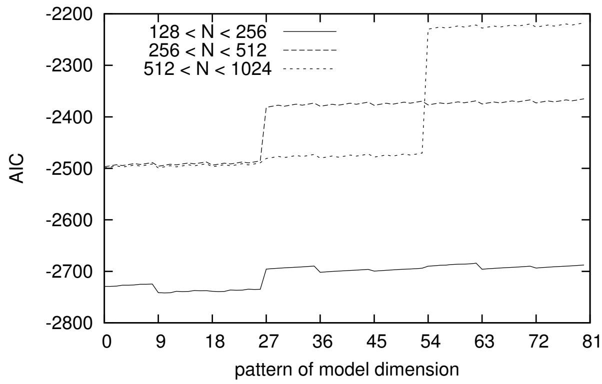

図 3　各モデル次数における AIC
Fig. 3　calculated AIC at each model dimension

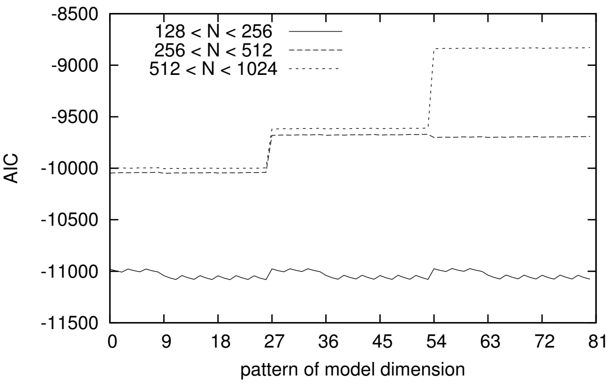

図 4　各モデル次数における AIC（全因子計画）
Fig. 4　calculated AIC at each model dimension (in full factorial design)

であった．また，全因子計画の実験では図 3 のようになり，AIC 最小の次数はそれぞれ順に $(1, 3, 3, 3)$，$(1, 2, 1, 1)$，$(1, 2, 1, 1)$ となった．

なお 3.3 節の計算式で分かるように，AIC の値は標本点に依存している点に注意したい．本実験では主に N と i の次数が AIC の値に影響を及ぼしていると見るべきである．

ここで決定した次数のモデルをによる推定の様子を，例として $N = 724$ におけるアンロール段数と性能値の関係で示すと**図 6**，アンロール段数 $(i, j, k) = (3, 4, 2)$ における N と性能値の関係で示すと図 6 のようになった．

上で構成した観測値のモデルによる再現度を評価するために，推定に用いた点集合について (観測値との残差 / 観測値の全分散) を計算した．またこれらのモデルの推定精度を評価するため，上記の次数を採用した場合について推定に用いた標本点とは別に無作為に

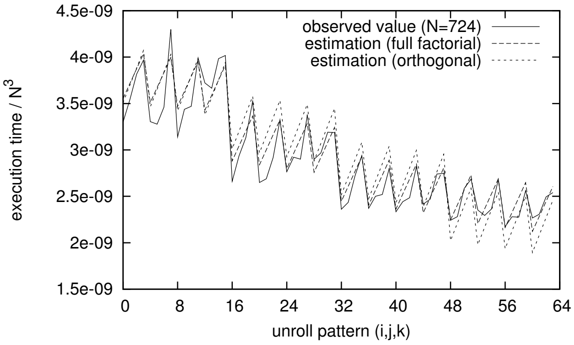

図 5　N=724 における実験値と推定値
Fig. 5　observed values and estimated values of each unroll patterns at N=724

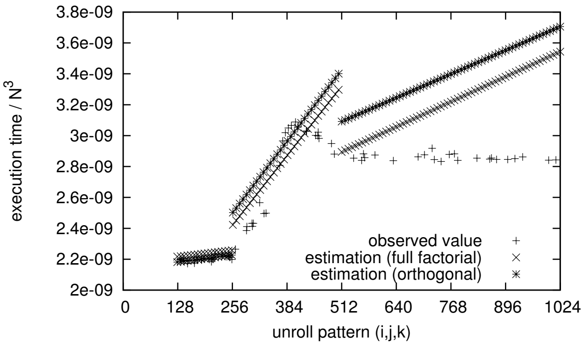

図 6　アンロール段数 $(i,j,k)=(3,4,2)$ における実験値と推定値
Fig. 6　observed values and estimated values at unroll levels $(i,j,k)=(3,4,2)$

点を選び，その実測値との残差平方和 / 全分散を計算した．対象の点は各ブロックごとに N について範囲内のランダムな 24 点，i,j,k は全点の全組合せとし，直交計画によるモデルと全因子計画によるモデルに対して同一の点集合を用いた．結果は**表 2** に示す．

$512<N<1024$ の部分が特に残差が大きくなっているが, アンロール段数によって推定の良さが大きく変化していた（図 6 はあまり一致しないパターンの例

表 2　残差平方和によるモデルの評価
Table 2　evaluation by residual sum of squares

		全因子計画	直交表計画
推定に用いた点			
$128<N$	<256	9.97 %	12.1 %
$256<N$	<512	13.6 %	18.4 %
$512<N$	<1024	54.5 %	57.3 %
新たに選んだ点			
$128<N$	<256	17.1 %	19.9 %
$256<N$	<512	35.7 %	38.9 %
$512<N$	<1024	40.6 %	47.6 %

である）ことから, N と i,j,k の交互作用が影響しているものと思われる．ただし，全因子計画においてモデルに交互作用を導入した場合でも，残差は高々全分散の 10%ほど減少する程度で, 表 2 における残差は N による細かい変動の影響も大きいのかもしれない．

3.5　考　　察

残差による評価では，全因子計画に比べ実験数の少ない直交表実験の方が総じて悪い評価になるものの，大きくは変わらないようである．明確な規準はないが，この場合には主効果を推定するのには十分であったと言ってもよいだろう．

また，直交表を自動チューニングにおけるサンプリングに用いるとすれば，モデルに含める効果を逐次的に増やしてゆき，その効果を推定できるように直交表を拡張し実験を行う，という手順を残差が満足の行く程度に減少するまで続けるという手順が予想できる．

4.　評価方法の検討

4.1　残差平方和による評価

前節で評価に用いた残差平方和では，プログラムの性能評価値についての推定精度を評価している．しかしながら，ここでのチューニング目標はあくまで最適なチューニングパラメータ（アンロール段数）を決定することにあり，直接的な評価ではないと考えられる．残差を基にした AIC によるモデルの評価ついても同様で，AIC の値が小さいことと，モデルが最適パラメータをより良く推定できることとは相関がない可能性がある．

以下では，最適パラメータ（ここでは最小値をとるアンロール段数）を選択するという視点でのモデル評価方法についていくつか案を挙げ検討を行う．

4.2　パラメータ主導の評価

4.2.1　最適パラメータの一致率による評価

まず単純な方法として，標本点における，実測値での最適パラメータ（以下 $(i_r,j_r,k_r)_N$）とモデルによる推定値での最適パラメータ（以下 $(i_e,j_e,k_e)_N$）の一致する割合を計算することを考える．最適なチューニングを行うという意味では，一致率が大きいほど良いモデルであると考えることはできるだろう．

しかし，推定して最適パラメータが実際の最適パラメータとは一致せずとも，性能値が最適に近い値であるならば悪い推定ではないとも言える．

4.2.2　最適性能値の相対誤差

前節をふまえ，推定最適パラメータにおける実測の性能値（$f((i_e,j_e,k_e)_N)$ とする）と実際の最適性能値（$f((i_r,j_r,k_r)_N)$ とする）の誤差を評価に用いること

を考える．つまり，各標本点 N での最適値の相対誤差

$$v_N = \frac{|f((i_r, j_r, k_r)_N) - f((i_e, j_e, k_e)_N)|}{|f((i_r, j_r, k_r)_N)|}$$

として，モデルの評価値を

$$\sum_N v_N$$

とする．最適パラメータが一致する場合は $v_N = 0$ となり，前節での評価（実際は全節とは逆に不正解率であるが）に対してある種の重み付けをしていることになる．

残差による場合と同様に，前節と本節の方法について評価を行ったところ，**表 3** のような結果になった．

4.2.3　性能値の変動を考慮した評価

前節での評価は最適値からのずれを計算しているが，他の，より悪い性能をとるパラメータでの性能値は考慮していない．

例えば，最適性能値で割る代わりにパラメータ変化に対する実測性能値の偏差や分散で割ってもよいだろう．もしくは性能が悪くなるパラメータを避けるという視点では，代わりに実測の |(最悪性能値 − 最適性能値)| で割るといったことも考えられる．

5．考　　察

参考のため直交表計画において，各次数のモデルでの残差平方和に対して 4.2.2 節の方法で計算した評価値（$256 < N < 512$ の部分）をグラフとして**図 7** に示す．縦軸が 4.2.2 節の評価値であるが，値がほぼ二値に分類されるように見えるのは，最小パラメータの推定に失敗する場合に，各モデル次数でその N や推定した最適パラメータがほぼ同一のパターンになっていることに起因している．

図 7 からわかるように，この二つの評価値の良し悪しはあまり相関がないようである．つまり，残差平方和のみでは評価の方法としては不適切である可能性があると言えるだろう．

ただし全節で提案した評価方法に共通するが，最適パラメータの誤差などはあくまで個々の N ごとのものであり，それを加算した場合に算出される値がどのような意味をもつかは明確ではない．

表 3　最適値をとるパラメータによる評価
（左が一致率，右が相対誤差の和）
Table 3　evaluation by optimal parameter configuration

		全因子計画	直交表計画
推定に用いた点			
$128 < N$	< 256	19/24 (0.024)	19/24 (0.024)
$256 < N$	< 512	17/24 (0.154)	17/24 (0.154)
$512 < N$	< 1024	1/24 (1.08)	1/24 (1.08)
新たに選んだ点			
$128 < N$	< 256	18/24 (0.025)	17/24 (0.027)
$256 < N$	< 512	20/24 (0.163)	20/24 (0.163)
$512 < N$	< 1024	2/24 (0.98)	2/24 (0.98)

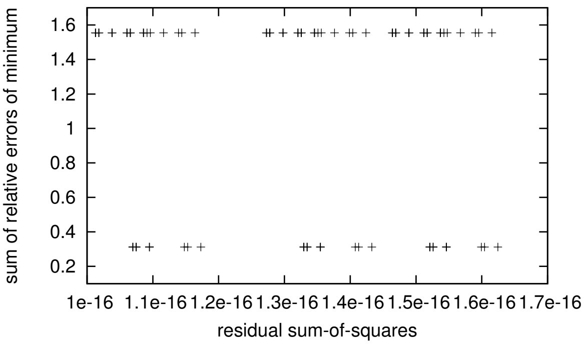

図 7　各モデル次数における，最小値の相対誤差の和と残差平方和
Fig. 7　The relation between residual sum-of-squares and sum of relative errors of minimum

また，評価方法はチューニング目的によって選択するべきであることは容易に想像される．本稿の実験は実行時間の最小化を目的としたチューニングであるが，例えば性能保証型のチューニングではまた評価方法を検討する必要がある．

6．ま と め

3 節では，ソフトウェアの性能評価関数を推定するにあたり遭遇し得る問題点と，実験計画的な方針に基づく対応策について実例を示した．そこでは実験計画法の技術がソフトウェアの自動チューニングにおいても有用となる可能性があることは確認された．

しかしながら 4 節で述べたように，性能評価値についての推定精度の向上と最適パラメータ選択精度の向上は直接的には結び付かず，自動チューニングの目的にあわせて実験計画の方針について検討することは今後の課題である．

参 考 文 献

1) 片桐孝洋：ソフトウエア自動チューニング–数値計算ソフトウエアへの適用とその可能性，慧文社 (2004).
2) 田口玄一：実験計画法 第三版 上下，丸善 (1976).
3) 柏村孝義，白鳥正樹，于　強：実験計画法による非線形問題の最適化–統計的設計支援システム，朝倉書店 (1998).
4) Montgomery, D. C.: *Design and Analysis of Experiments*, Probability and Statistics, John Wiley, 3rd edition (1991).
5) 坂元慶行，石黒真木夫，北川源四郎：情報量統計学，共立出版 (1982).