

3.2 第12回情報シナジー研究会

平成26年2月24日に開催された「第12回情報シナジー研究会」における講演の内容を掲載する。

第12回 情報シナジー研究会

東北大学サイバーサイエンスセンターでは、先端的情報技術の研究発表と情報交換を目的として情報シナジー研究会を企画・開催しております。今般、第12回研究会を下記の要領で開催致しますので、奮ってご参加下さい（参加申し込み・予約等は必要ありません）。

日時：平成26年2月24日（月） 10:00～12:00

会場：サイバーサイエンスセンター・本館5階・講義室

プログラム：

10:00～10:10

平成25年度 東北大学サイバーサイエンスセンター顕彰 表彰式

10:10～10:15

開会

10:15～10:35

OpenMP Parallelization using Compile Log of Automatic Parallelization

Azmir Ridzuan bin Azlan ⁽¹⁾, Kazuhiko Komatsu ⁽²⁾, Ryusuke Egawa ⁽²⁾, Hiroyuki Takizawa ⁽³⁾, Hiroaki Kobayashi ⁽²⁾

⁽¹⁾ Faculty of Engineering, Tohoku University, ⁽²⁾ Cyberscience Center, Tohoku University,

⁽³⁾ Graduate School of Information Sciences, Tohoku University

10:35～10:55

アクセラレータのためのプログラム最適化とその性能評価

平井亮太 ⁽¹⁾, 平澤将一 ⁽²⁾, 滝沢寛之 ⁽²⁾, 小林広明 ⁽³⁾

⁽¹⁾ 東北大学工学部, ⁽²⁾ 東北大学大学院情報科学研究科, ⁽³⁾ 東北大学サイバーサイエンスセンター

10:55～11:15

TOPIC参加組織におけるHTTPSサーバの運用状況

水木敬明 ⁽¹⁾, 森倫子 ⁽²⁾, 曽根秀昭 ⁽¹⁾

⁽¹⁾ 東北大学サイバーサイエンスセンター, ⁽²⁾ 東北大学情報部情報基盤課

11:15～11:35

フリーアドレスオフィスにおけるネットワークシステムの省電力化のための自律的制御手法

栗原孝太 ⁽¹⁾, 和泉諭 ⁽²⁾, 阿部亨 ⁽³⁾, 菅沼拓夫 ⁽³⁾

⁽¹⁾ 東北大学大学院情報科学研究科, ⁽²⁾ 東北大学電気通信研究所, ⁽³⁾ 東北大学サイバーサイエンスセンター

11:35～11:55

方向統計学に基づく位相限定相関関数の統計的解析

八巻俊輔 ⁽¹⁾, 阿部正英 ⁽²⁾, 川又政征 ⁽²⁾

⁽¹⁾ 東北大学サイバーサイエンスセンター, ⁽²⁾ 東北大学大学院工学研究科

11:55～12:00

閉会

問合せ・連絡先： 東北大学情報部情報基盤課

Tel. 022-795-3407

OpenMP Parallelization using Compile Log of Automatic Parallelization

Azmir Ridzuan bin Azlan¹, Kazuhiko Komatsu², Ryusuke Egawa², Hiroyuki Takizawa³,
Hiroaki Kobayashi²

Abstract. Recently, HPC applications play important roles in various fields of science and engineering. To achieve more complex and accurate simulation, exploiting the full potential of supercomputing systems is essential. The performance of recent supercomputing systems has been drastically improved by adding a lot of processing cores in a supercomputing system. Thus, the exploiting of the potential of such cores is crucial. Automatic parallelization and OpenMP parallelization are examples of the methods to utilize the multiple cores in recent supercomputing systems. However, utilizing the multiple cores using these methods alone is difficult, due to the low portability of the existing applications, and the time consuming procedures of OpenMP parallelization. Therefore, this study proposes a low cost OpenMP parallelization method that combines the function of automatic parallelization, and the high portability of OpenMP codes. The performance of the proposed OpenMP parallelization is evaluated on a Multi-Scale Simulator for the Geoenvironment (MSSG) using the NEC SX-9, the Nehalem EX Cluster, the Fujitsu FX10 and the Hitachi SR16000 supercomputing systems. The evaluations show that the proposed OpenMP parallelization is performed effectively on multiple supercomputing systems.

¹ Faculty of Engineering, Tohoku University

² Cyberscience Center, Tohoku University

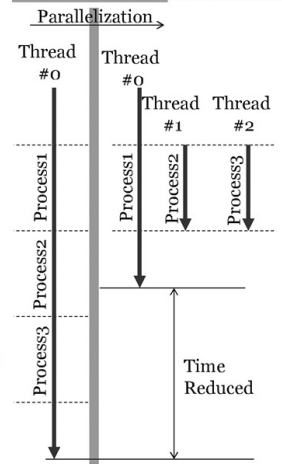
³ Graduate School of Information Science, Tohoku University

OpenMP Parallelization using Compile Log of Automatic Parallelization

Azmir Ridzuan bin Azlan , Kazuhiko Komatsu,
Ryusuke Egawa , Hiroyuki Takizawa ,
Hiroaki Kobayashi
Faculty of Engineering, Tohoku University

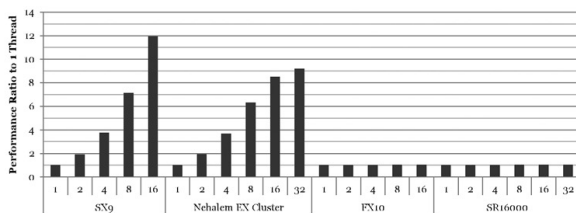
Background

- HPC applications
 - Sequential process
 - NOT FULL POTENTIAL
- Exploit supercomputing systems by optimizations
 - Vector Supercomputer
 - SX9
 - Scalar Supercomputer
 - Nehalem Ex Cluster
 - FX10
 - SR16000
 - Automatic parallelizations
 - Parallel process
 - Performed automatically by compilers
 - Reduce execution time
 - Increase performance



Preliminary Evaluation

- Kernel parallelized automatically on 4 different supercomputing systems
 - SX9, Nehalem-EX Cluster, FX10, SR16000



- Automatic parallelization is not always effective
 - Manual parallelizations required (ex : OpenMP)

OpenMP

- Language extension for parallelization process
- Insert OpenMP directives on parallelizable regions
- PROBLEM
 - Manual insertion
 - Identify parallelizable loop
 - Time consuming
 - Error-prone

```
! test.f90
...
!$omp parallel do
do j=1, n
do i=1, m
a(i,j) = b(i,j) + c(i,j)
...
enddo
enddo
!$omp end parallel do

do j = 1, n
e(i,j) = e(i,j-1) + 2
...
enddo
...
```

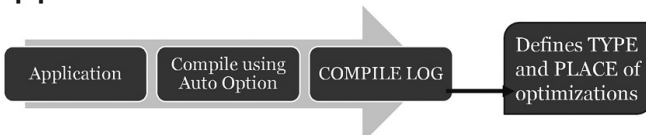
Parallel execution

Loop dependency

Objective

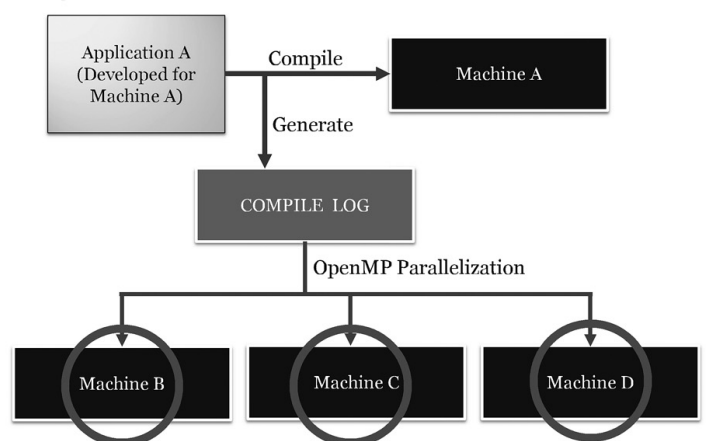
- Realize a low cost OpenMP parallelization for improving the performance of applications in various systems

Approach



- Proposes a method to easily insert OpenMP directives by using the compile logs
 - Easily determine which part of the application can be parallelized using OpenMP directives

OpenMP Parallelization



The Proposed Method



```

1  ! test.f90
...
200 do j=1, n
201 do i=1, m
202   a(i,j) = b(i,j) + c(i,j)
...
207   sum = sum + 1
208 enddo
209 enddo
...

```

Compile Log

```

sxf90 -R2 -Pauto -c test.f90 -o test.exe
...
f90: mul(10) : test.f90, line 200:
      Parallel routine generated : main_$1
f90: mul(1) : test.f90, line 200:
      Parallelized by PARDO
f90: vec(4) : test.f90, line 201:
      Vectorized loop
f90: mul(4) : test.f90, line 207 :
      CRITICAL section
...

```

Performance Evaluations

OBJECTIVE

- To confirm the effectiveness of proposed method
- To examine the performance of the proposed method on multiple supercomputing systems

MATRIX

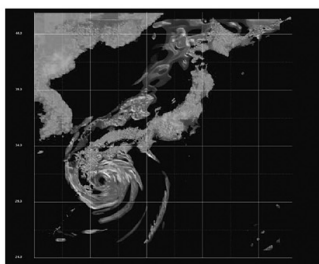
- Performance Ratio

$$\text{Performance Ratio} = \frac{\text{Performance of Automatic or OpenMP Parallelization on Multiple Threads}}{\text{Performance of Automatic Parallelization on 1 Thread}}$$

Target Application

Multi-Scale Simulator for the Geoenvironment (MSSG)

- A simulator for coupled non-hydrostatic atmosphere-ocean-land global circulation model
 - Predict the weather through the changes of atmosphere and ocean
- REISNER
 - Total Lines : 6044
 - Total Loops : 58
 - Max Nested Loop : 3



www.jamstec.go.jp/

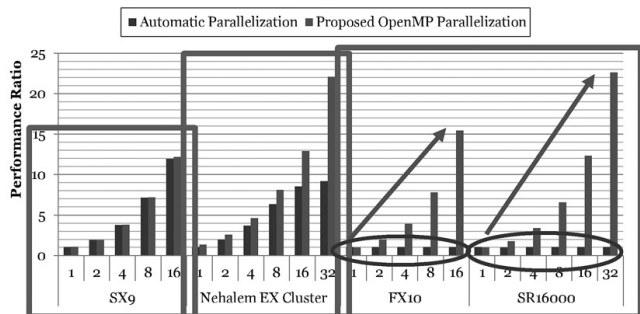
Experimental Environment

REISNER

- Developed and optimized on SX9
 - Proposed method using the compile log by SX compiler

System	Max Cores	Compiler Option
NEC SX9	16	sxf90 -R2 -Cvopt -Pauto/-Popenmp
Nehalem EX	32	ifort -xHost -O3 -parallel/-openmp
Fujitsu FX10	16	frtpx -Kfast -Kparallel,noopenmp/-Kopenmp,noparallel
Hitachi SR16K	32	f90 -Oss -parallel/-omp

Evaluation Results



SX9

- Similar Performance
- Proposed method is a SUCCESS in
- Multi core : No parallelization

FX10 & SR16000

- OpenMP : Performance increase
- Auto : Remains the same
- No auto. parallelization by compilers
- Different optimization prioritized

Conclusion

Objective

- Perform OpenMP parallelization that reduce cost of programmer and human errors

Proposed Method

- Perform parallelization using information on compile log of automatic parallelizations

Evaluation Results

- OpenMP parallelization is performed effectively
- Superior performance compared to codes parallelized automatically

Future Work

- Develop an automatic OpenMP parallelization tool that automatically performs OpenMP parallelization based on the information obtained by compile logs of automatic parallelization

アクセラレータのためのプログラム最適化に関する研究

平井 亮太* 平澤 将一† 滝沢 寛之† 小林 広明‡

概要

高性能計算分野において計算機の演算性能の向上を目的として、汎用プロセッサと高い演算性能を持つアクセラレータを組み合わせた複合型計算機の利用が注目されている。複合型計算機の実行環境として、各種のアクセラレータを同一のプログラムを用いて制御可能な **OpenCL** が普及している。しかしながら、各アクセラレータの性能を十分に引き出すにはプログラムの最適化が必要である。また、有効な最適化手法はアクセラレータによって異なる。そこで本研究においては、各種のアクセラレータで有効な最適化手法を明らかにすることを目的とし、現在用いられている一般的な最適化手法とアクセラレータ独自の最適化手法を同一プログラムに適用し、実行性能を測定することで効果を比較した。その結果、アクセラレータによって有効な最適化手法や最適な実行パラメータが異なることが明らかとなった。

* 東北大学工学部機械知能航空工学科

† 東北大学大学院情報科学研究科/JST CREST

‡ 東北大学サイバーサイエンスセンター

アクセラレータのための プログラム最適化に関する研究

情報シナジー研究会 2/24

滝沢 研
BOTB1206 平井 亮太
指導教員 滝沢 寛之 准教授

背景

- 多様なアクセラレータの登場
 - GPU, MICコプロセッサ, FPGAなど
 - 同じプログラムを共通環境OpenCL (Open Computing Language)で実行可能
- アクセラレータの性能を十分に発揮できるとは限らない
 - プログラムの高速化を実現するためには、そのプログラムの最適化が必要
 - 有効な最適化方法はアクセラレータにより異なる

各アクセラレータに対して有効な最適化手法を明らかにする必要

目的・アプローチ

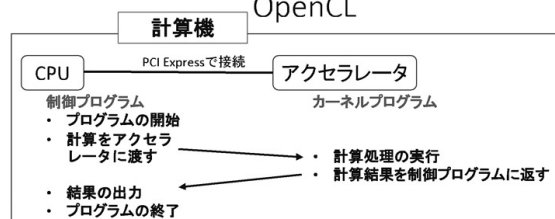
目的: 各アクセラレータに対して有効な最適化方法を明らかにする



アプローチ:

- OpenCLプログラムに対して各種の最適化をする
 - 従来から用いられている一般的な最適化
 - アクセラレータ特有の最適化方法
- 各アクセラレータで最適化したOpenCLプログラムの性能測定実験を行い、最適化が有効か評価する

アクセラレータ向け開発環境 OpenCL



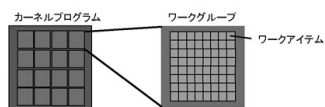
- 制御プログラムとカーネルプログラム
 - 制御プログラムはホストデバイス(CPU等)で実行
 - 主要な計算処理を実行するカーネルプログラムはアクセラレータ(GPU、MIC、FPGA)で実行

アクセラレータ向け開発環境 OpenCL

- カーネルプログラムの処理はワークグループ、ワークアイテムから構成
 - 各ワークアイテムは別々のデータに対して同じ計算を行う

例
ワークアイテム#1 $A \times B + C$ の計算
ワークアイテム#2 $D \times E + G$ の計算
ワークアイテム#N $X \times Y + Z$ の計算

ワークグループ、ワークアイテム数は任意に設定できる

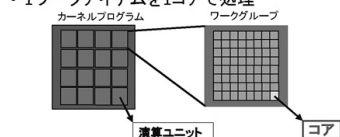


- 2種類のメモリを指定できる
 - グローバルメモリ 全てのワークグループからアクセス可能
 - ローカルメモリ 1ワークグループ内で共有

論理的なもので実際に割り当てられるメモリはアクセラレータにより異なる

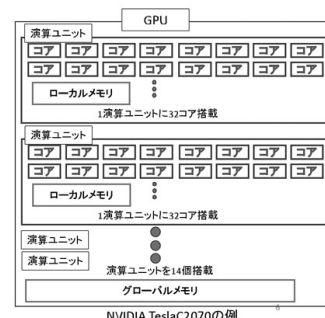
アクセラレータの特徴 GPU

- 1ワークグループを1演算ユニットで処理
- 1ワークアイテムを1コアで処理



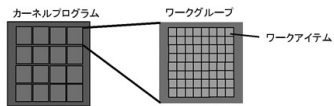
メモリ構成

- グローバルメモリを指定した場合
 - GPU全体からアクセス可能な、大容量、低速メモリ
- ローカルメモリを指定した場合
 - 演算ユニットで共有する、小容量、高速メモリ



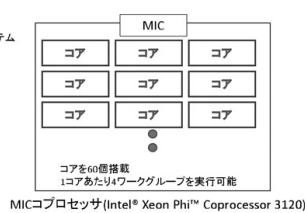
アクセラレータの特徴 MIC

- 4ワークグループを1コアで処理



メモリ構成

- グローバルメモリを指定しても、ローカルメモリを指定しても、物理的には同じメモリが割り当てられる

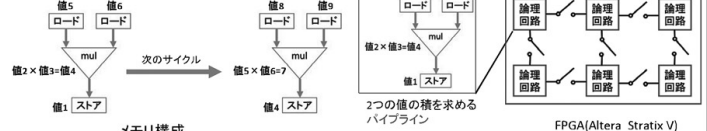


MICプロセッサ(Intel® Xeon Phi™ Coprocessor 3120)

7

アクセラレータの特徴 FPGA

- 書き換え可能な論理回路を多数有し、カーネルに合わせて処理の一連の流れを実行する回路(パイプライン)を生成
- パイプライン処理では別データに対する処理を順次実行可能



メモリ構成

- グローバルメモリを指定した場合
 - 大容量、低速メモリ
- ローカルメモリを指定した場合
 - 小容量、高速メモリ

FPGA(Altera Stratix V)

8

パイプライン多重化・ベクトル化(FPGAのみ)

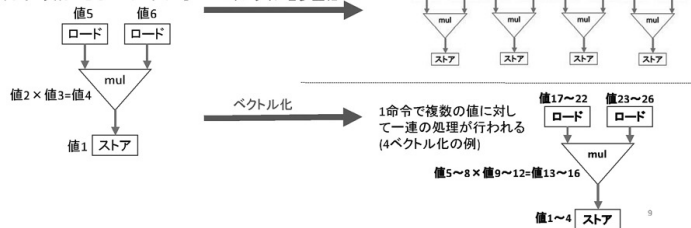
異なるデータのメモリへのストア、積計算、メモリからのロードを同時に実行できる「パイプライン」

積計算を実行するパイプラインが4重化(パイプライン4重化の例)

パイプラインを多重化

ベクトル化

1命令で複数の値に対して一連の処理が行われる(4ベクトル化の例)



9

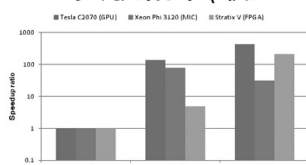
性能測定実験

アクセラレータの特徴を考慮し最適化したプログラムと最適化前のプログラムの性能を比較

- 評価するデバイス
 - GPU (NVIDIA Tesla C2070)
 - MICプロセッサ(Intel® Xeon Phi™ Coprocessor 3120)
 - FPGA(Altera Stratix V)
- 評価に使うプログラム
 - 行列積計算プログラム
 - 行列の大きさは4096 × 4096

10

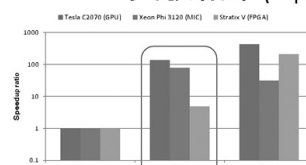
実験結果(最適化内容)



- Baseline 最適化前
- Opt1 実行パラメータ(ワークアイテム、ワークグループ数)の調整
- Opt2 実行パラメータ(ワークアイテム、ワークグループ数)の調整+ローカルメモリの使用
 - GPU向けの一般的な最適化

11

実験結果(Opt1)



最適化実行パラメータ

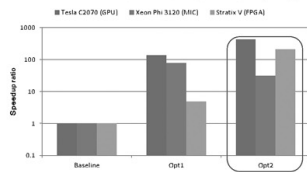
	GPU	MIC	FPGA
ワークグループ数	16384	65536	4096
1ワークグループあたりのワークアイテム数	1024	256	4096

Opt1 → 実行パラメータの調整

- 各デバイスで高速化
- 最適化実行パラメータはアクセラレータによって異なる
 - アクセラレータによってアーキテクチャやワークアイテムの処理が異なるため

12

実験結果(Opt2)



最適な実行パラメータ

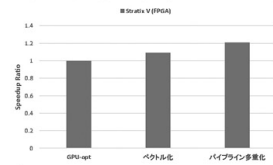
	GPU	MIC	FPGA
ワークグループ数	16384	65536	4096
1ワークグループあたりのワークアイテム数	1024	256	4096

Opt2 → 実行パラメータの調整+ローカルメモリの使用

- Opt1に比べ GPU、FPGAで高速化 MICでは低速化
- GPU、FPGAではローカルメモリを指定した場合、小容量、高速メモリが割り当てられる
 - 高速化に寄与
- MICではローカルメモリを指定した場合、グローバルメモリを指定した場合と同じメモリが割り当てられる
 - ローカルメモリの確保に時間がかかるため低速化

13

実験結果(FPGA-opt)



FPGAに対する最適化

- 演算のベクトル化 → 約1.09倍高速化
- パイプライン多重化 → 約1.21倍高速化

FPGAのOpenCLコンパイラによってコードの修正なく最適化可能

14

まとめ

目的

- 各アクセラレータに対して有効な最適化方法を明らかにする

アプローチ

- GPU、MIC、FPGAにおいて汎用的な最適化手法およびアクセラレータ独自の最適化手法をOpenCLプログラムに適用し効果を測定

結果

- 各最適化の効果が明らかになった
- 同じ最適化手法でもアクセラレータによって効果が異なることや、アクセラレータによっては性能が低下する可能性があることが明らかとなった

今後の課題

- 実アプリケーションでの性能測定・評価

15

TOPIC参加組織におけるHTTPSサーバの運用状況

水木敬明^{†1} 森倫子^{†2} 曾根秀昭^{†1}

^{†1} 東北大学 サイバーサイエンスセンター

^{†2} 東北大学 情報部情報基盤課

東北学術研究インターネットコミュニティ (TOPIC) は、東北地区の大学・高専・学術研究機関が参加することによりコミュニティを形成しており、東北大学サイバーサイエンスセンターに事務局が置かれている。本発表では、各TOPIC 参加組織が運用するHTTPSサーバに関して、その運用状況を調査する。具体的には、証明書の発行者、署名アルゴリズム、公開鍵、及び Record Protocol で利用される暗号化アルゴリズムについて調査及び考察する。なお、本発表の内容は、2013年11月に実施した調査（「大学ICT推進協議会 2013年度 年次大会」にて発表）を基に、2014年2月22日に実施した調査との比較を加えたものである。

TOPIC参加組織における HTTPSサーバの運用状況

水木敬明, 森 倫子, 曾根秀昭

東北大学サイバーサイエンスセンター
東北大学 情報部情報基盤課

第12回 情報シナジー研究会@本館5階・講義室
(2014年2月24日10:55—11:15)

1

はじめに

東北学術研究インターネットコ
ミュニティ (TOPIC)

TOPIC参加組織における HTTPSサーバの運用状況

2013年11月に実施した調査(「大学ICT推進協
議会 2013年度 年次大会」にて発表)

+

2014年2月22日の調査との比較

2

東北学術研究インターネット コミュニティ TOPIC

<http://www.topic.ad.jp/>

東北地区において、大学・高専・学術研究機関等
が学術研究・教育活動を支援するコンピュータ
ネットワーク環境の発展に貢献することを目的に
コミュニティを形成。

現在57機関。

事務局が東北大学サイバーサイエンスセンター
に置かれている。

3

やってみたこと

TOPICの参加組織のウェブサーバを調査

調査対象HTTPSサーバの列挙

サーバ証明書の発行者の調査

署名アルゴリズムの調査

公開鍵の調査

Record Protocolにおける暗号化通信

4

調査対象となったサーバ(2013年11月) : 39機関、171サーバ

HTTPSサーバの台数; 3台以上の組織(その1)

TOPIC参加機関	台数
東北大学 (tohoku.ac.jp)	64
秋田大学 (akita-u.ac.jp)	10
北里大学 (kitasato-u.ac.jp)	8
弘前大学 (hirosaki-u.ac.jp)	7
東北学院大学 (tohoku-gakuin.ac.jp)	7
会津大学 (u-aizu.ac.jp)	5
岩手県立大学 (iwate-pu.ac.jp)	5

5

調査対象となったサーバ(2013年11月) : HTTPSサーバの台数; 3台以上の組織(その2)

TOPIC参加機関	台数
福島大学 (fukushima-u.ac.jp)	5
仙台高等専門学校 (sendai-nct.ac.jp)	4
岩手医科大学 (iwate-med.ac.jp)	4
岩手大学 (iwate-u.ac.jp)	4
石巻専修大学 (isenshu-u.ac.jp)	4
八戸工業大学 (hi-tech.ac.jp)	4
東北福祉大学 (tfu.ac.jp)	3
福島県立医科大学 (fmu.ac.jp)	3

6

ホスト名の抽出

ホスト名	個数
www	58
webmail	9
wm	4
portal	4
ia	3
www3	2
wmail	2
ssl	2
menkyo	2
mail	2
lms	2

7

2014年2月22日に,

39機関、171サーバ(2013年11月調査時)

に対して, 再度SSL接続を試み, そのうち応答のあったものは

38機関、168サーバ(2014年2月調査)

8

発行者の調査の結果;上位6つ

発行者	サーバ数
NII Open Domain CA - G2	69
VeriSign Class 3 Secure Server CA - G3	23
GlobalSign Domain Validation CA - G2	13
RapidSSL CA	9
Cybertrust Japan Public CA G2	7
AlphaSSL CA - G2	5

9

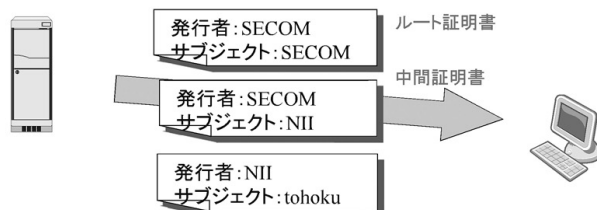
発行者の調査の結果; 2014年2月との比較

発行者	サーバ数
NII Open Domain CA - G2	69→69
VeriSign Class 3 Secure Server CA - G3	23→23
GlobalSign Domain Validation CA - G2	13→13
RapidSSL CA	9→9
Cybertrust Japan Public CA G2	7→7
AlphaSSL CA - G2	5→5

上位6つに
変化なし

10

各サーバから送られてくる証明書の数



送られてくる証明書数	サーバ数
1	27
2	105
3	32
4	7

11

各サーバから送られてくる証明書の数; 2014年2月との比較

送られてくる証明書数	サーバ数
1	27→24
2	105→105
3	32→32
4	7→7

12

署名アルゴリズムの調査の結果

署名アルゴリズム	サーバ数
sha1WithRSAEncryption	166
md5WithRSAEncryption	4
sha256WithRSAEncryption	1

***WithRSAは、すべてRSASSA-PKCS1-v1_5

鍵長は1つを除きすべて2048ビット

13

署名アルゴリズムの調査の結果; 2014年2月との比較

署名アルゴリズム	サーバ数
sha1WithRSAEncryption	166→163
md5WithRSAEncryption	4→4
sha256WithRSAEncryption	1→1

14

公開鍵の調査

公開鍵	サーバ数
rsaEncryption (2048bit)	149
rsaEncryption (1024bit)	20
rsaEncryption (512bit)	2

すべてrsaEncryption

15

公開鍵の調査; 2014年2月との比較

公開鍵	サーバ数
rsaEncryption (2048bit)	149→150
rsaEncryption (1024bit)	20→16
rsaEncryption (512bit)	2→2

16

どの暗号アルゴリズムが受け入れられるか？

CAMELLIA	AES256	AES128	RC4-SHA	RC4-MD5	サーバ数
	✓	✓	✓	✓	109
✓	✓	✓	✓	✓	39
			✓	✓	8
✓	✓	✓	✓		5
✓	✓	✓			4
	✓	✓	✓		3
		✓	✓	✓	2
	✓	✓			1

17

どの暗号アルゴリズムが受け入れられるか？ 2014年2月との比較

CAMELLIA	AES256	AES128	RC4-SHA	RC4-MD5	サーバ数
	✓	✓	✓	✓	109→106
✓	✓	✓	✓	✓	39→41
			✓	✓	8→7
✓	✓	✓	✓		5→5
✓	✓	✓			4→4
	✓	✓	✓		3→2
		✓	✓	✓	2→2
	✓	✓			1→1

18

フリーアドレスオフィスにおける ネットワークシステムの省電力化のための自律的制御手法

粟原 孝太^{†1}

和泉 諭^{†2}

阿部 亨^{†3}

菅沼 拓夫^{†3}

^{†1} 東北大学 大学院情報科学研究科

^{†2} 東北大学 電気通信研究所

^{†3} 東北大学 サイバーサイエンスセンター / 大学院情報科学研究科

本研究では、オフィス環境におけるネットワークシステムの省電力化を目指している。本論文では、オフィスで働く作業員の座席を固定しない「フリーアドレス」という形態のオフィスに焦点を当て、作業員の作業効率を維持しつつネットワークシステムを省電力化する手法を提案する。具体的には、人間の作業効率やオフィス全体の消費電力を考慮して最適化を行うための目的関数と、得られた最適状態へ適切に移行するためのプランニングモデルの設計を行い、シミュレーション実験を通して本手法が有効に働くことを確認する。

Autonomous Control Method for Energy-efficient Network Systems in Non-territorial Offices

Kouta Awahara^{†1}

Satoru Izumi^{†2}

Toru Abe^{†3}

Takuo Suganuma^{†3}

^{†1} Graduate School of Information Sciences, Tohoku University

^{†2} Research Institute of Electrical Communication, Tohoku University

^{†3} Cyberscience Center/Graduate School of Information Sciences, Tohoku University

In this study, we are aiming for reductions in energy consumption of network systems in office environments. This paper describes an autonomous control method for reducing energy consumption of network systems in non-territorial offices where members' desks are not fixed. Specifically, we propose objective functions for optimization considering system-wide energy consumption and members' working efficiency. Furthermore, we design planning model for making a network system migrate to its optimal state obtained. Finally, we confirm the effectiveness of our method through simulation experiments.

フリーアドレスオフィスにおけるネットワークシステムの省電力化のための自律的制御手法

栗原 孝太^{*1} 和泉 諭^{*2} 阿部 亨^{*3} 菅沼 拓夫^{*3}

^{*1} 東北大学 大学院情報科学研究科

^{*2} 東北大学 電気通信研究所

^{*3} 東北大学 サイバーサイエンスセンター/大学院情報科学研究科

目次

1. 序論
 1. 本研究の背景
 2. 本研究の概要
2. 関連研究とその課題
 1. 関連研究
 2. 関連研究の課題
3. 提案: ネットワークシステムの省電力化のための自律的制御手法
 1. 提案手法の概要
 2. 想定する環境
4. 設計・実装
 1. Optimizer
 2. Planner
5. 評価実験
 1. 実験の概要
 2. 実験1: Optimizerの評価
 3. 実験2: Plannerの評価
6. 結論
 1. 結論
 2. 今後の課題

2014/2/24

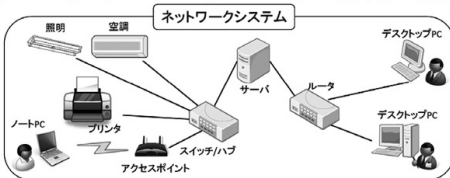
第12回情報シナジー研究会

2

1. 序論

1.1 本研究の背景

- オフィスにおける消費電力は年々増加傾向にあり^[1]、その約20%がICT機器による電力消費^[2]
- ➡ オフィスにおけるICT機器の省電力化が必要
- オフィスでは様々なICT機器(PC、プリンタ、ネットワーク機器etc.)が連動してネットワークシステムを構成
- 様々な機器を含むネットワークシステム全体を省電力化する技術は未確立
- 消費電力の低減だけでなく、作業者の作業効率の考慮も必要
- フリーアドレスオフィスが増加^[3]、メンバ配置の効率化が要検討



- [1] 資源エネルギー庁, <http://www.enecho.meti.go.jp/>
[2] 省エネルギーセンター, http://www.ecsi.or.jp/office_bldg/
[3] 社団法人日本オフィス家具協会, "フリーアドレスオフィスの効果," 『月刊経緯』2007年8月号, 2014/2/24

3

1. 序論

1.2 本研究の概要

- 目的
 - ・ オフィス環境におけるネットワークシステムの省電力化
- 課題
 - ・ 人の作業効率を維持しつつ、ICT機器の設定やネットワークシステムの構成の導出が困難
 - ・ 複雑なシステムに対するシステム再構成の自律化が困難
- 提案
 - ・ オフィス環境におけるネットワークシステムの省電力化のための自律的制御手法
 - (F1) 目的関数に基づくネットワークシステムの最適化
 - (F2) プランニングを用いた最適状態への移行プランの自動生成

2014/2/24

第12回情報シナジー研究会

4

2. 関連研究とその課題

2.1 関連研究

- オフィス環境における個々のICT機器の省電力化
 - ・ デスクトップPC^{[4][5]}
 - － オフィス環境を対象とし、リモートで使用されているPC内の処理を他のPCにマイグレーション、集約
 - ・ PCモニター^[6]
 - － 超音波センサを用いて人の在席状況を観測、離席またはモニターを使わない作業中にはモニターの電源off
 - ・ プリンタ^{[7][8]}
 - － オフィスにおける出力状況を分析、複合機の最適配置
 - ・ その他照明^[9]、空調^[10]など個別の機器に関する様々な研究が存在
 - － 個々のICT機器単独の制御のみ

- [4] Andreas Berl et al. "Network virtualization in energy-efficient office environments," Computer Networks, Volume 54, Issue 16, 15 November 2010, pp. 2856-2868.
[5] Andreas Berl et al. "An energy consumption model for virtualized office environments," Future Generation Computer Systems, Volume 27, Issue 8, October 2011, pp. 1047-1055.
[6] Paula Jaramillo et al. "Improving Energy Efficiency through Activity-Aware Control of Office Appliances using Proximity Sensing - A Real-Life Study," Proceedings of the 5th International Workshop on Smart Environments and Ambient Intelligence, pp. 664-669, 2013.
[7] 株式会社リコー, "マネージド・プリント・サービス (MPS)," <http://www.rioh.co.jp/outsourcing/mps/>
[8] 富士ゼロックス株式会社, "出力環境最適化," <http://www.fujixerox.co.jp/product/multifunction/promotion/optimize/>
[9] 小野 晋子他, "知能照明システムのための自律分散型最適化アルゴリズム," 電気学会論文誌-Vol.130, No.5, 2010.
[10] 高木 康夫 他, "快適性と省エネを両立させる連携型エネ空調制御技術," 計測自動制御学会論文集 Vol.46, No.8, pp.430-438, 2010.

5

2. 関連研究とその課題

2.2 関連研究の課題

- 従来研究の課題
 - (P1) 人の作業効率を維持しつつ、作業メンバの配置やネットワークシステムの構成の導出が困難
 - ➡ 人の作業効率を考慮しながら、ネットワークシステムを構成する各機器を統合的に制御する必要
 - (P2) 複雑なシステムに対するシステム再構成の自律化が困難
 - ➡ システム再構成の適切な手順を導出する手法が必要
- 本提案手法
 - オフィス環境におけるネットワークシステムの省電力化のための自律的制御手法
 - (F1) 目的関数に基づくネットワークシステムの最適化 ➡ (P1)の解決
 - (F2) プランニング*を用いた最適状態への移行プランの自動生成 ➡ (P2)の解決

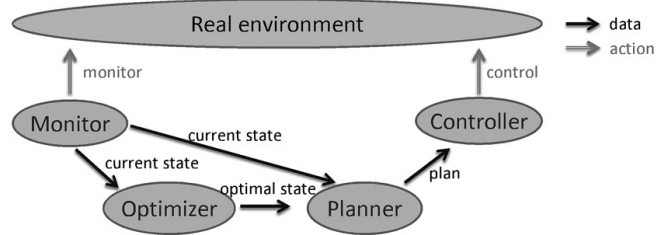
*プランニング: 与えられた目標を達成するために必要なアクションの系列を自動生成すること

2014/2/24

第12回情報シナジー研究会

6

3.1 提案手法の概要



Monitor	実環境の状態を観測
Optimizer	消費電力と人間の作業効率を考慮し最適状態を計算(F1)
Planner	実環境の状態と最適状態を受け取り、実環境の状態から最適状態へ至るためのプラン(動作や操作の系列)を生成(F2)
Controller	プランに基づき、実環境の状態を制御

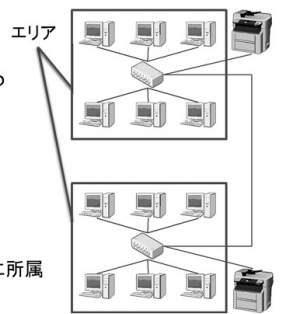
2014/2/24

第12回情報シナジー研究会

7

3.2 想定する環境

- エリアについて
 - ・ フリーアドレスオフィス環境を想定
 - ・ オフィスは複数のエリアに区切られている
 - ・ 1つのエリアにはスイッチ、プリンタが1つずつ設置
- メンバについて
 - ・ メンバは座席に設置されたデスクトップPCを用いて作業
 - ・ メンバがエリアにおいて作業を行う際、そのエリアのスイッチ、プリンタをON
 - ・ 全てのメンバはそれぞれ1つのプロジェクトに所属
 - ・ メンバがオフィスに到着する時間は未知

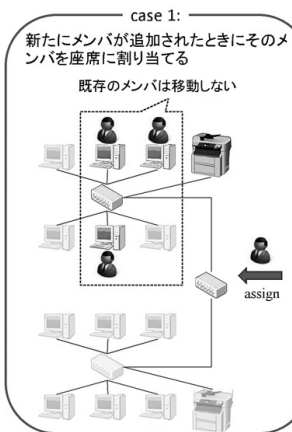


2014/2/24

第12回情報シナジー研究会

8

3.2 想定する環境



- ① 新たなメンバの到着を観測(Monitor)
- ② 目的関数が最小となる座席の割り当てを決定、その際のシステムの状態を最適状態(ゴール)とする(Optimizer)
- ③ ゴールへ移行するためのプランを生成(Planner)
- ④ 実際に各機器を制御(Controller)

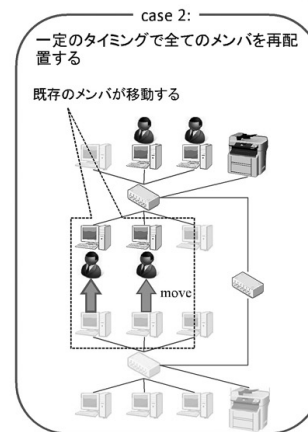
本研究における対象

2014/2/24

第12回情報シナジー研究会

9

3.2 想定する環境



- ① 現在の各機器の使用状態やメンバの配置を観測 (Monitor)
- ② 目的関数が最小となるメンバの配置を決定、その際のシステムの状態を最適状態(ゴール)とする(Optimizer)
- ③ ゴールへ移行するためのプランを生成(Planner)
- ④ 実際に各機器を制御(Controller)

本研究における対象

2014/2/24

第12回情報シナジー研究会

10

4.1 Optimizer

- ・ メンバが追加される際、各席に割り当てた場合の目的関数 $F(s)$ をそれぞれ計算し、 $F(s)$ が最小となる席を決定

$$F(s(t)) = \alpha \frac{E(s(t))}{E(s_e(t))} - (1 - \alpha) \frac{W(s(t))}{W(s_w(t))}$$

- ・ $s(t)$...時刻 t における状態ベクトル:
ある時刻におけるメンバの配置やシステムを構成する機器の状態を表す
- ・ $E(s(t))$...システム全体の消費電力 (PC、プリンタ、スイッチの消費電力の総和)
- ・ $s_e(t)$...割り当て方の中で消費電力が最大となる状態ベクトル
- ・ $W(s(t))$...メンバ全体の作業効率 (同じエリアにいる同一プロジェクトや異なるプロジェクトのペア数、PCのスペックによる作業効率を数値化)
- ・ $s_w(t)$...割り当て方の中で作業効率が最大となる状態ベクトル
- ・ α ...目的関数における消費電力の重み ($0 \leq \alpha \leq 1$)
(消費電力や作業効率の詳細は付録参照)

2014/2/24

第12回情報シナジー研究会

11

4.2 Planner

- プランナとして Graphplan のオープンソース実装である JPlan^[11] を用いる
- Graphplan^[12]: 高速性を重視したプランニングアルゴリズム

- プランナにおける記述の例 (詳細は付録参照)

- ・ オブジェクト
 - Member (メンバ)、PC (デスクトップPC)、etc.
- ・ ステート
 - InArea(Member ?m, Area ?a) ...メンバ?m がエリア?a にいる
 - PrinterOn(Printer ?pr) ...プリンタ?pr がONになっている
 - etc.
- ・ アクション
 - MoveArea(Area ?from, Area ?to, Member ?m) ...メンバ?m がエリア?from からエリア?to へ移動する
 - TurnPrinterOn(Printer ?pr, Switch ?sw) ...スイッチ?sw に繋がれたプリンタ?pr をONにする
 - etc.

[11] Yasser El-Manzawy, "JPlan", <http://jplan.sourceforge.net/>.

[12] Avrim Blum and Merrick Furst, "Fast Planning Through Planning Graph Analysis", Artificial Intelligence Vol. 90, pp.281-300, 1997.

2014/2/24

第12回情報シナジー研究会

12

5.1 実験の概要

➤ 実験の概要

OptimizerおよびPlannerを評価するためのシミュレーション実験を行った

- 実験1: Optimizerの評価実験
- 実験2: Plannerの評価実験

➤ 実験環境

PC	Dell Optiplex 980
OS	Windows 7 Professional 64bit
CPU	Intel® Core™ i7 870 @ 2.93GHz
RAM	4.00GB
開発環境	Eclipse IDE for Java EE Developers 1.5.2
実装言語	Java

5.2 実験1: Optimizerの評価

➤ 評価手法

- メンバを一人ずつ逐次座席に割り当てる場合を想定し、消費電力・作業効率・目的関数の出力の時間変化、およびその総和を比較

パラメータ・メンバの到着パターン等を変更し他アルゴリズムと比較

- 比較アルゴリズム
 - In order ...空いている席に順番に割り当てる
 - Random ...空いている席にランダムに割り当てる

実験1.1: メンバがバラバラに到着し、その後離席や復帰が起こる場合の評価

実験1.2: 目的関数のパラメータ α を変化させた場合の評価

5.2 実験1: Optimizerの評価

➤ 実験1.1: メンバの離席や復帰が起こる場合の評価

- 機器の詳細: 付録参照

- あるプロジェクトのメンバが会議で一時離席し、一定時間経過後に復帰する場合を想定
- 消費電力、作業効率、目的関数の出力の時間変化を比較

オフィス環境

Area	5
PC	30 (6×5)
Printer	5 (1×5)
Switch	5 (1×5)
Project	4

パラメータ

parameter	value
α	0.5
β	100
γ	50
δ	0.5

メンバ

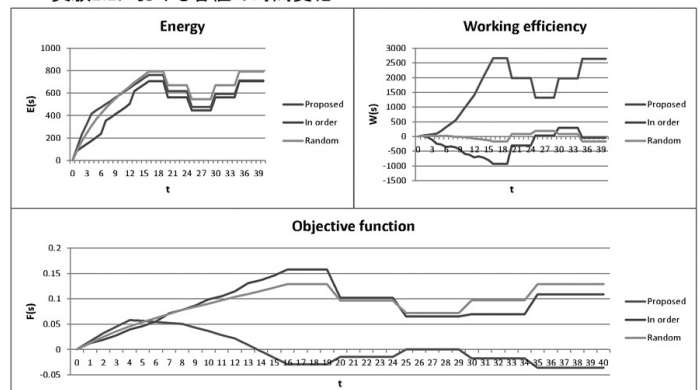
Project	Member
1	1,2,3,4
2	5,6,7,8
3	9,10,11,12
4	13,14,15,16

メンバの到着および離席

Time	Arrival	Leaving	Time	Arrival	Leaving
1	1	21			
2	5	22			
3	9	23			
4	13	24			
5	2	25	5,6,7,8		
6	6	26			
7	10	27			
8	14	28			
9	3	29			
10	7	30	1,2,3,4		
11	11	31			
12	15	32			
13	4	33			
14	8	34			
15	12	35	5,6,7,8		
16	16	36			
17		37			
18		38			
19		39			
20	1,2,3,4	40			

5.2 実験1: Optimizerの評価

- 実験1.1における各値の時間変化



5.2 実験1: Optimizerの評価

➤ 実験1.1の考察

- 消費電力について
 - In orderはメンバのプロジェクトにかかわらず部屋にメンバを詰め込むので、消費電力は他のアルゴリズムより小さくなる
 - メンバの離席や復帰などが発生する場合、提案手法との差は小さくなる
- 作業効率について
 - 提案手法では各プロジェクトのメンバが適切に各部屋に配置され、他のアルゴリズムより作業効率は非常に高くなる
 - 同じプロジェクトのメンバが同時に離席・復帰するため、In Orderでは離席前より作業効率が高くなる
- 目的関数について
 - 提案手法では目的関数の出力は他アルゴリズムより非常に小さい値となった

5.2 実験1: Optimizerの評価

➤ 実験1.2: 目的関数のパラメータ α を変化させた場合の評価

- メンバの到着時間はランダム (1000回の平均をとる)
- α を0~1.0まで0.1刻みで変化
- 機器の詳細: 付録F参照

オフィス環境

Area	5
PC	30 (6×5)
Printer	5 (1×5)
Switch	5 (1×5)
Project	4

パラメータ

parameter	value
α	0 ~ 1.0
β	100
γ	50
δ	0.5

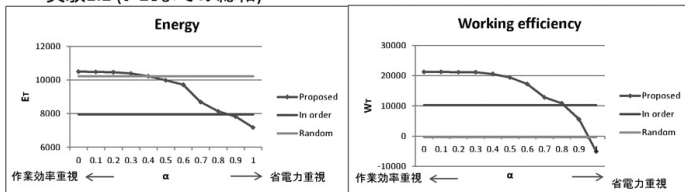
- 消費電力、作業効率、目的関数の出力の総和 ($t=20$ まで) を比較

Member	Project	Arrival Time
1	1	random
2	1	random
3	1	random
4	1	random
5	2	random
6	2	random
7	2	random
8	2	random
9	3	random
10	3	random
11	3	random
12	3	random
13	4	random
14	4	random
15	4	random
16	4	random

5. 評価実験

5.2 実験1: Optimizerの評価

・ 実験1.2 (t=20までの総和)



Objective function (Fr)

α	Proposed	In order	Random	Difference(In order)	Difference(Random)
0	-5.3018	1.1542	0.1007	-6.4560	-5.4025
0.1	-4.8583	1.2733	0.3998	-5.7318	-4.8583
0.2	-3.9951	1.4175	0.6982	-5.0125	-4.2932
0.3	-2.7533	1.5465	1.0029	-4.2998	-3.7562
0.4	-1.8471	1.6700	1.2928	-3.5171	-3.1399
0.5	-0.9070	1.7867	1.6053	-2.6937	-2.5123
0.6	0.0428	1.9001	1.9020	-1.8573	-1.8593
0.7	0.8817	2.0350	2.1962	-1.1533	-1.3145
0.8	1.4268	2.1899	2.5187	-0.7631	-1.0920
0.9	1.9966	2.3182	2.8204	-0.3216	-0.8238
1	2.1725	2.4179	3.0984	-0.2454	-0.9259

2014/2/24

第12回情報シナジー研究会

19

5. 評価実験

5.2 実験1: Optimizerの評価

➤ 実験1.2の考察

- 提案手法では、 α を変化させることにより消費電力と作業効率を任意のバランスで制御できることを確認
- 任意の α において、目的関数の出力は比較アルゴリズムより小さくなった

➤ 実験1を通した考察

- 提案手法では、メンバの到着順番やパラメータに関わらず、比較アルゴリズムより目的関数の出力は小さくなった
- 計算時間について
 - PC30台の環境において、メンバ16人を連続で割り当て際の計算時間は10msオーダーであり、充分小さい
 - メンバ1人を座席に割り当て際の計算時間は $O(\text{PCの数})$ であり、現実的なオフィスにおいては充分実用的

➡ P1(人の作業効率を維持しつつ、ICT機器の設定やネットワークシステムの構成の導出が困難)の解決

2014/2/24

第12回情報シナジー研究会

20

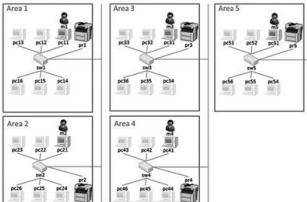
5. 評価実験

5.3 実験2: Plannerの評価

➤ 評価手法

- 既存メンバを再配置する場合を想定
- 初期状態とゴールを与え、プランニングにより適切なプランが出力されるかを確認

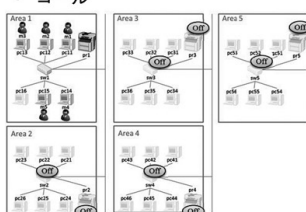
・ 初期状態



スタートに基づく記述は付録参照

・ ゴール

Area	5
PC	30 (6 × 5)
Printer	5 (1 × 5)
Switch	5 (1 × 5)
Member	5



2014/2/24

第12回情報シナジー研究会

21

5. 評価実験

5.3 実験2: Plannerの評価

➤ 結果

- 初期状態からゴールに至る適切なプランの出力を確認
- 計算時間について
 - 今回の実験では計算時間は2943ms
 - エリアの数が計算時間に最も影響を与える
 - エリア5つ程度の環境では充分実用的

➡ P2(複雑なシステムに対するシステム再構成の自律化が困難)の解決

出力されたプラン

Plan:

- UnusePC (m2, pc21, sw2)
- UnusePC (m3, pc31, sw3)
- UnusePC (m4, pc41, sw4)
- UnusePC (m5, pc51, sw5)
- TurnPrinterOff (pr2, sw2)
- TurnPrinterOff (pr3, sw3)
- TurnPrinterOff (pr4, sw4)
- TurnPrinterOff (pr5, sw5)
- MoveArea (a2, a1, m2)
- MoveArea (a3, a1, m3)
- MoveArea (a4, a1, m4)
- MoveArea (a5, a1, m5)
- TurnSwitchOff (sw2)
- TurnSwitchOff (sw3)
- TurnSwitchOff (sw4)
- TurnSwitchOff (sw5)
- UsePC (m2, pc12, sw1)
- UsePC (m3, pc13, sw1)
- UsePC (m4, pc14, sw1)
- UsePC (m5, pc15, sw1)

2014/2/24

第12回情報シナジー研究会

22

6. 結論

6.1 結論

➤ 目的

- オフィス環境におけるネットワークシステムの省電力化

➤ 課題

- (P1) 人の作業効率を維持しつつ、ICT機器の設定やネットワークシステムの構成を導出するのが困難
- (P2) 複雑なシステムに対し、システムの再構成を自律化するのが困難

➤ 提案

- オフィス環境におけるネットワークシステムの省電力化のための自律的制御手法
 - (F1)目的関数に基づくネットワークシステムの最適化
 - (F2)プランニングを用いた最適状態への移行プランの自動生成

➤ 成果

- (F1), (F2)により、作業効率を維持しつつネットワークシステムの省電力化を実用的な性能で実現できることを示した

2014/2/24

第12回情報シナジー研究会

23

6. 結論

6.2 今後の課題

➤ 今後の課題

- case 2におけるOptimizerの検討
 - アルゴリズム
 - 再配置を行うタイミング
 - 移動そのものによる作業効率変化のモデル化
 - etc.
- より複雑な環境でのプランニングの検討
- 環境の観測(Monitor)、制御(Controller)に関する検討
- 実環境における実験

2014/2/24

第12回情報シナジー研究会

24

付録: Optimizerの設計

- 消費電力 $E(s)$ の定義

$$E(s) = \sum_{pc \in PC} P(pc, s) + \sum_{pr \in Printer} P(pr, s) + \sum_{sw \in Switch} P(sw, s)$$

$P(pc, s) = S(pc, s) \times P_{pc}$...pcの消費電力

$S(pc, s)$...pcの状況(利用率)に応じて変化

※本実装の中では全てのPCにおいて $S(pc, s) = 0.28$ と設定

P_{pc} ...pcの最大消費電力、各PCで異なる値

$$P(pr, s) = \begin{cases} 0, & \text{if } pr \text{ is off} \\ P_{pr}, & \text{if } pr \text{ is on} \end{cases} \quad \dots \text{プリンタ } pr \text{ の消費電力}$$

P_{pr} ...プリンタON時の消費電力

$$P(sw, s) = \begin{cases} 0, & \text{if } sw \text{ is off} \\ P_{sw}, & \text{if } sw \text{ is on} \end{cases} \quad \dots \text{ネットワークスイッチ } sw \text{ の消費電力}$$

P_{sw} ...ネットワークスイッチON時の消費電力

2014/2/24

第12回情報シナジー研究会

25

付録: Optimizerの設計

- 作業効率 $W(s)$ の定義

$$W(s) = \sum_{a \in Area} W_{\text{same_pro}}(a, s) - \sum_{a \in Area} W_{\text{diff_pro}}(a, s) + \sum_{m \in Member} W_{PC}(m, s)$$

$$W_{\text{same_pro}}(a, s) = \beta \times N_{\text{same_pair}}(a, s)$$

$N_{\text{same_pair}}(a, s)$...同じエリアにいる同一プロジェクトのペアの数

同一プロジェクトのメンバが同じエリアにいる時、メンバ間でのコミュニケーションが促進され、作業効率が增加することを示す値

$$W_{\text{diff_pro}}(a, s) = \gamma \times N_{\text{diff_pair}}(a, s)$$

$N_{\text{diff_pair}}(a, s)$...同じエリアにいる異なるプロジェクトのペアの数

異なるプロジェクトのメンバが同じエリアにいる時、部外秘に気を遣わなければならないために作業効率が低下することを示す値

$$W_{PC}(m, s) = \delta \times P(pc_m, s)$$

pc_m ...メンバ m が使っているpc

メンバが使用するPCのスペックにより変化する作業効率を示す値

β, γ, δ ...重み係数

第12回情報シナジー研究会

26

付録: Plannerの設計

➤ プランナにおけるステートの記述

- InArea(Member ?m, Area ?a) ...メンバ?m がエリア?a にいる
- Using(Member ?m, PC?p) ...メンバ?m がPC?p を使っている
- NotUsing(Member ?m) ...メンバ?m がPCを使っていない
- PrinterOn(Printer ?pr) ...プリンタ?pr がONになっている
- PrinterOff(Printer ?pr) ...プリンタ?pr がOFFになっている
- SwitchOn(Switch ?sw) ...スイッチ?sw がONになっている
- SwitchOff(Switch ?sw) ...スイッチ?sw がOFFになっている
- HasOrder(PC?p, Switch ?sw) / HasOrder(Printer ?pr, Switch ?sw)
...PC?p / プリンタ ?pr がスイッチ?swに が繋がっている

2014/2/24

第12回情報シナジー研究会

27

付録: Plannerの設計

➤ プランナにおけるアクションの記述

- (アクションの名前) ((変数))
[[前提条件]]
[[追加するステート]]
[[削除するステート]]
- MoveArea (Area ?from, Area ?to, Member ?m) [エリアを移動する]
[InArea (?m, ?from) & NotUsing (?m)]
[InArea (?m, ?to)]
[InArea (?m, ?from)]

2014/2/24

第12回情報シナジー研究会

28

付録: Plannerの設計

➤ プランナにおけるアクションの記述 (Cont.)

- UsePC (Member ?m, PC?p, Switch ?sw) [PCを使い始める]
[NotUsing (?m) & HasOrder (?p, ?sw) & SwitchOn (?sw)]
[Using (?m, ?p)]
[NotUsing (?m)]
- UnusePC (Member ?m, PC?p, Switch ?sw) [PCを使うのをやめる]
[Using (?m, ?p) & HasOrder (?p, ?sw) & SwitchOn (?sw)]
[NotUsing (?m)]
[Using (?m, ?p)]
- TurnPrinterOn (Printer ?pr, Switch ?sw) [プリンタをオン]
[PrinterOff (?pr) & HasOrder (?pr, ?sw) & SwitchOn (?sw)]
[PrinterOn (?pr)]
[PrinterOff (?pr)]

2014/2/24

第12回情報シナジー研究会

29

付録: Plannerの設計

➤ プランナにおけるアクションの記述 (Cont.)

- TurnPrinterOff (Printer ?pr, Switch ?sw) [プリンタをオフ]
[PrinterOn (?pr) & HasOrder (?pr, ?sw) & SwitchOn (?sw)]
[PrinterOff (?pr)]
[PrinterOn (?pr)]
- TurnSwitchOn (Switch ?sw) [スイッチをオン]
[SwitchOff (?sw)]
[SwitchOn (?sw)]
[SwitchOff (?sw)]
- TurnSwitchOff (Switch ?sw) [スイッチをオフ]
[SwitchOn (?sw)]
[SwitchOff (?sw)]
[SwitchOn (?sw)]

2014/2/24

第12回情報シナジー研究会

30

付録:実験1における機器のパラメータ

Switch	Area	P_{SW}
1	1	20
2	2	20
3	3	20
4	4	20
5	5	20

Printer	Area	P_{pr}
1	1	40
2	2	40
3	3	40
4	4	40
5	5	40

ϕ C	area	ν_{dec}
11	1	100
12	1	102
13	1	104
14	1	106
15	1	108
16	1	110
21	2	200
22	2	103
23	2	105
24	2	107
25	2	109
26	2	111
31	3	200
32	3	102
33	3	104
34	3	106
35	3	108
36	3	110

pC	area	$\ln p_C$
41	4	101
42	4	103
43	4	105
44	4	107
45	4	109
46	4	111
51	5	90
52	5	92
53	5	94
54	5	96
55	5	98
56	5	100

31

付録:実験2における初期状態とゴール

objects:	PC (pc43):	* initial state:	HasOrder(pc25, sw2)8	* goal state:
Member (m1):	PC (pc44):	InArea(m1, a1)8	HasOrder(pc26, sw3)8	InArea(m1, a1)8
Member (m2):	PC (pc45):	InArea(m2, a2)8	HasOrder(pc31, sw3)8	InArea(m2, a1)8
Member (m3):	PC (pc46):	InArea(m3, a3)8	HasOrder(pc32, sw3)8	InArea(m3, a1)8
Member (m5):	PC (pc52):	InArea(m5, a5)8	HasOrder(pc34, sw3)8	InArea(m5, a1)8
Area (a1):	PC (pc53):	Using(m1, pc11)8	HasOrder(pc35, sw3)8	Using(m1, pc11)8
Area (a2):	PC (pc54):	Using(m2, pc12)8	HasOrder(pc36, sw3)8	Using(m2, pc12)8
Area (a3):	PC (pc55):	Using(m3, pc13)8	HasOrder(pc41, sw4)8	Using(m3, pc13)8
Area (a4):	PC (pc65):	Using(m4, pc14)8	HasOrder(pc42, sw4)8	Using(m4, pc14)8
Area (a5):	Printer (pr1):	Using(m5, pc15)8	HasOrder(pc43, sw4)8	Using(m5, pc15)8
PC (pc11):	Printer (pr2):	PrinterOn(pr1)8	HasOrder(pc44, sw4)8	PrinterOn(pr1)8
PC (pc12):	Printer (pr3):	PrinterOn(pr2)8	HasOrder(pc45, sw4)8	PrinterOn(pr2)8
PC (pc13):	Printer (pr4):	PrinterOn(pr3)8	HasOrder(pc46, sw4)8	PrinterOff(pr1)8
PC (pc14):	Printer (pr5):	PrinterOn(pr4)8	HasOrder(pn51, sw5)8	PrinterOff(pr4)8
PC (pc15):	Switch (sw1):	SwitchOn(sw5)8	HasOrder(pn52, sw5)8	PrinterOff(pn55)8
PC (pc16):	Switch (sw2):	SwitchOn(sw6)8	HasOrder(pn53, sw5)8	SwitchOn(sw1)8
PC (pc21):	Switch (sw3):	SwitchOn(sw2)8	HasOrder(pn54, sw5)8	SwitchOff(sw2)8
PC (pc22):	Switch (sw4):	SwitchOn(sw3)8	HasOrder(pn55, sw5)8	SwitchOff(sw3)8
PC (pc23):	Switch (sw5):	SwitchOn(sw4)8	HasOrder(pn56, sw5)8	SwitchOff(sw4)8
PC (pc24):		SwitchOn(sw5)8	HasOrder(pr1, sw1)8	SwitchOff(sw5)8
PC (pc25):		HasOrder(pn1, sw1)8	HasOrder(pr2, sw2)8	
PC (pc26):		HasOrder(pn2, sw1)8	HasOrder(pn3, sw1)8	
PC (pc31):		HasOrder(pn3, sw1)8	HasOrder(pn4, sw1)8	
PC (pc32):		HasOrder(pn4, sw1)8	HasOrder(pn5, sw5)8	
PC (pc33):		HasOrder(pn5, sw5)8		
PC (pc34):		HasOrder(pn6, sw1)8		
PC (pc35):		HasOrder(pn7, sw1)8		
PC (pc36):		HasOrder(pn2, sw2)8		
PC (pc41):		HasOrder(pn3, sw2)8		
PC (pc42):		HasOrder(pn4, sw2)8		

32

方向統計学に基づく位相限定相関関数の統計的解析

八巻 俊輔^{†1}

阿部 正英^{†2}

川又 政征^{†2}

^{†1} 東北大学 サイバーサイエンスセンター

^{†2} 東北大学 大学院工学研究科

本研究では、方向統計学に基づく位相限定相関関数の統計的性質について論じる。まず、2つの信号の位相スペクトル差を確率変数と仮定し、位相限定相関関数の期待値と分散を導出する。次に、位相スペクトル差を角度データと考え、その確率分布として円周確率分布を仮定する。その結果、位相限定相関関数の期待値および分散はそれぞれ、位相スペクトル差の円周分散の1次関数および2次関数として表せることを示す。

Statistical Analysis of Phase-Only Correlation Functions Based on Directional Statistics

Shunsuke Yamaki^{†1}

Masahide Abe^{†2}

Masayuki Kawamata^{†2}

^{†1} Cyberscience Center, Tohoku University

^{†2} Graduate School of Engineering, Tohoku University

In this study, we discuss statistical analysis of phase-only correlation functions based on directional statistics. We first derive the expectation and variance of the phase-only correlation functions assuming phase-spectrum differences of two input signals to be probability variables. We next assume circular probability distributions for the phase-spectrum differences, considering phase-spectrum differences to be circular data. As a result, we can express the expectation and variance of phase-only correlation functions as linear and quadratic functions of circular variance of phase-spectrum differences, respectively.

方向統計学に基づく 位相限定相関関数の統計的解析

○八巻 俊輔¹, 阿部 正英², 川又 政征²

¹ 東北大学サイバーサイエンスセンター

² 東北大学大学院工学研究科

2014年2月24日

第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ.

位相限定相関(POC)関数 (1/4)

■ POC関数を用いた信号マッチング技術

■ 応用例: パターン認識, 生体認証, 通信など

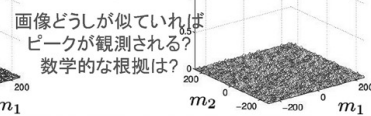
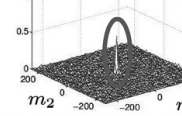
似ている画像どうし



似ていない画像どうし



POC関数
計算



画像どうしが似ていれば
ピークが観測される?
数学的な根拠は?

第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 1

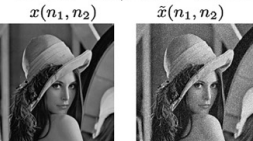
位相限定相関(POC)関数 (2/4)

■ 2信号間の位相スペクトル差とPOC関数との関係

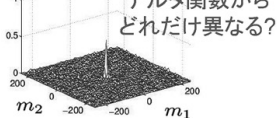
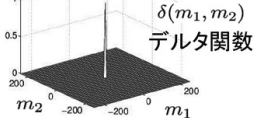
2つの信号の位相スペクトルが
等しいとき(位相スペクトル差=0)



2つの信号の位相スペクトルが
等しくないとき(位相スペクトル差≠0)



POC関数
計算



デルタ関数から
どれだけ異なる?

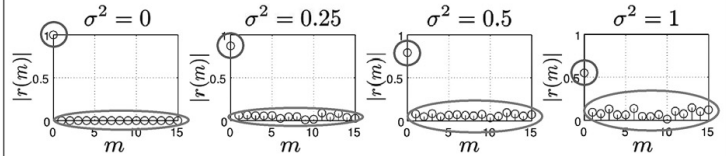
第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 2

位相限定相関(POC)関数 (3/4)

■ 位相スペクトル差の変動に対するPOC関数 $r(m)$ の挙動の変化

例: 位相スペクトル差 α_k がガウス分布 $N(0, \sigma^2)$ にしたがう
確率変数であると仮定



分散 σ^2 の増加に伴い,

○ $|r(0)|$ は減少する傾向

○ $|r(m \neq 0)|$ は増加する傾向

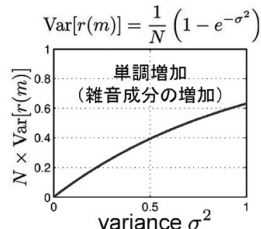
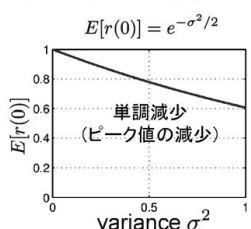
これらの挙動を説明する
数学的な根拠は?

第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 3

位相限定相関(POC)関数 (4/4)

■ 位相スペクトル差の分散 σ^2 の変動に対するPOC関数の期待値と分散の変化^[1]



疑問点: 位相スペクトル差の確率分布として、
ガウス分布は適さないのではないか? ➡ 方向統計学で
解決する

[1] Yamaki, Odagiri, Abe, and Kawamata, Proc. IEEE IC-NIDC, Sept. 2012.

第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 4

発表の概要

■ 研究の目的

■ 2つの信号間の位相スペクトル差の変化に対するPOC関数の
変化を統計的に解析する

■ 一般統計学に基づく統計的解析

■ 位相スペクトル差を線形確率分布に従う確率変数と仮定^[1]
■ 線形確率分布は、位相スペクトル差の分布には適さない?

■ 本発表の内容

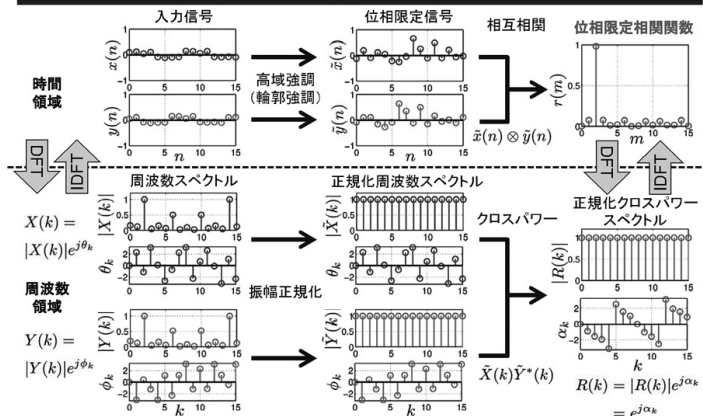
■ 方向統計学に基づき、位相スペクトル差を円周確率分布に従う
確率変数と仮定し、POC関数の挙動を統計的に解析する
■ 位相スペクトル差の確率分布として、代表的な円周確率分布
のひとつであるvon-Mises分布を仮定した計算例を示す

[1] Yamaki, Odagiri, Abe, and Kawamata, Proc. IEEE IC-NIDC, Sept. 2012.

第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 5

位相限定相関 (POC) 関数



第12回 情報シナジー研究会

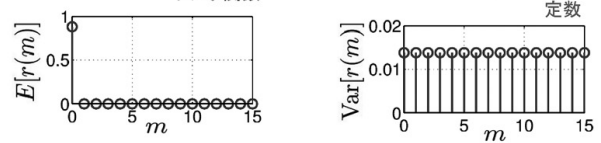
Kawamata Lab., Tohoku Univ. 6

一般統計学に基づく統計的解析 (1/2)

■ 複素信号のPOC関数の期待値と分散の導出^[1]

$$\text{POC関数 } r(m) = \frac{1}{N} \sum_{k=0}^{N-1} e^{j\alpha_k} W_N^{-mk} \quad \alpha_k: \text{位相スペクトル差 (確率変数と仮定)}$$

$$\text{期待値 } E[r(m)] = \frac{A\delta(m)}{\text{デルタ関数}} \quad \text{分散 } \text{Var}[r(m)] = \frac{1}{N}(1 - |A|^2)$$



ただし、 $A = E[e^{j\alpha_k}]$ (位相因子 $e^{j\alpha_k}$ の期待値)

[1] Yamaki, Odagiri, Abe, and Kawamata, Proc. IEEE IC-NIDC, Sept. 2012.

第12回 情報シナジー研究会

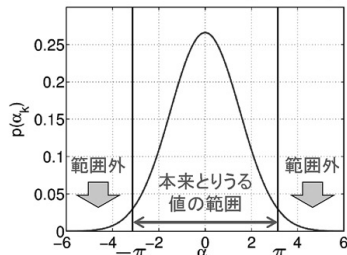
Kawamata Lab., Tohoku Univ. 7

一般統計学に基づく統計的解析 (2/2)

■ 確率分布の仮定における問題点

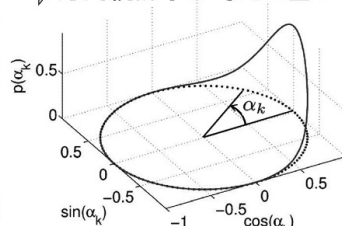
■ 位相スペクトル差 α_k の確率分布として、ガウス分布は適用できないのではないか?

ガウス分布 $\alpha_k \in (-\infty, \infty)$



円周確率分布 $\alpha_k \in [-\pi, \pi)$

➡ 方向統計学の考えに基づく



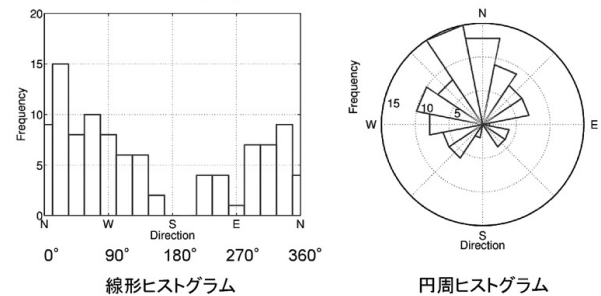
第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 8

方向統計学とは? (1/2)

■ 角度データ(円周上のデータ)を統計的に扱う学問

■ 例1: ある地点で定期的に観測した風向きのデータ



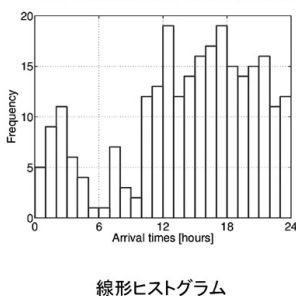
第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 9

方向統計学とは? (2/2)

■ 角度データ(円周上のデータ)を統計的に扱う学問

■ 例2: ある病院に搬送された時刻ごとの患者数のデータ※



線形ヒストグラム

円周ヒストグラム

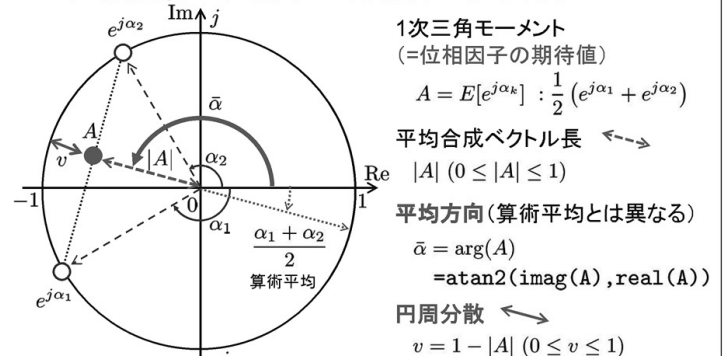
※ 出典: N. I. Fisher, "Statistical Analysis of Circular Data," Cambridge University Press, 1993.

第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 10

方向統計学: 平均方向と円周分散

■ 平均と分散の考え方(2つの角度データでの例)



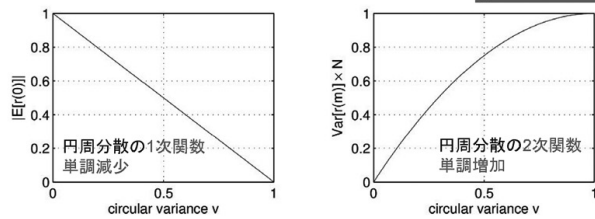
第12回 情報シナジー研究会

Kawamata Lab., Tohoku Univ. 11

POC関数の期待値と分散の一般式

■ 円周分散 v の変化に対するPOC関数 $r(m)$ の挙動

期待値 $|E[r(0)]| = (1-v)\delta(m)$ 分散 $\text{Var}[r(m)] = \frac{1}{N} (1 - (1-v)^2)$



2信号間の位相スペクトル差の円周分散を用いて、POC関数の性質を記述できる

von-Mises分布 (1/2)

■ von-Mises分布 $\text{VM}(\mu, \beta)$ (円周正規分布)

確率密度関数

$$p(\alpha_k) = \frac{1}{2\pi I_0(\beta)} e^{\beta \cos(\alpha_k - \mu)} \quad (-\pi \leq \alpha < \pi)$$

μ : 平均方向 β : 集中度

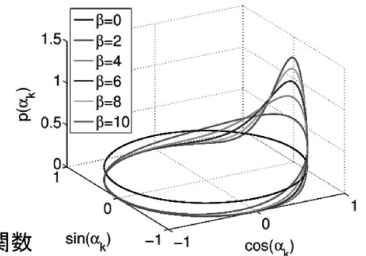
特性関数

$$\psi_\alpha(t) = \frac{I_{|t|}(\beta)}{I_0(\beta)} e^{j\mu t}$$

円周分散

$$v = 1 - \frac{I_1(\beta)}{I_0(\beta)}$$

$I_\nu(\beta)$: 第1種 ν 次修正ベッセル関数



von-Mises分布 (2/2)

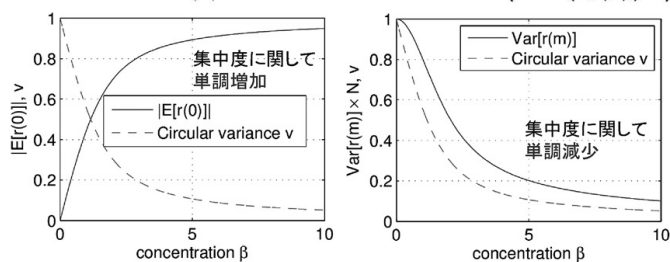
■ 集中度 β の変化に対するPOC関数 $r(m)$ の挙動

期待値

$$|E[r(m)]| = \frac{I_1(\beta)}{I_0(\beta)} \delta(m)$$

分散

$$\text{Var}[r(m)] = \frac{1}{N} \left(1 - \left(\frac{I_1(\beta)}{I_0(\beta)} \right)^2 \right)$$



まとめ

- 方向統計学に基づき、位相スペクトル差に円周確率分布を仮定し、POC関数の挙動を統計的に解析した。
- 位相スペクトル差に円周確率分布を仮定した場合、POC関数の期待値と分散はそれぞれ、位相スペクトル差の円周分散の1次関数と2次関数を用いて表せる。
- 位相スペクトル差の確率分布として、代表的な円周確率分布のひとつであるvon-Mises分布を仮定した計算例を示した。