โปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธีนอร์ทาและคุณลักษณะของนอร์ทา

นายจตุพล ทะน้อม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาสถิติศาสตรมหาบัณฑิต

สาขาวิชาสถิติ ภาควิชาสถิติ

คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2548

ISBN 974-14-1809-4

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

PROGRAM FOR GENERATING MULTIVARIATE RANDOM VECTORS BY NORTA AND IT'S CHARACTERISTICS

Mr. Chatuphol Thanom

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Statistics
Department of Statistics
Faculty of Commerce and Accountancy
Chulalongkorn University
Academic Year 2005
ISBN 974-14-1809-4

หัวข้อวิทยานิพนธ์	โปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธีนอร์ทาและคุณลักษณะของนอร์ทา
โดย	นายจตุพล ทะน้อม
สาขาวิชา	สถิติ
อาจารย์ที่ปรึกษา	อาจารย์ ดร.เสกสรร เกียรติสุไพบูลย์

คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.... คณบดีคณะพาณิชยศาสตร์และการบัญชี

(ผู้ช่วยศาสตราจารย์ ดร.ดนุชา คุณพนิชกิจ)

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ

(รองศาสตราจารย์ ดร.สุพล ดุรงค์วัฒนา)

.... อาจารย์ที่ปรึกษา

(อาจารย์ ดร.เสกสรร เกียรติสุไพบูลย์)

.... กรรมการ

(อาจารย์ ดร.อรุณี กำลัง)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จตุพล ทะน้อม : โปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธีนอร์ทาและคุณลักษณะของนอร์ทา (PROGRAM FOR GENERATING MULTIVARIATE RANDOM VECTORS BY NORTA AND IT'S CHARACTERISTICS) อ. ที่ปรึกษา : อ.ดร.เสกสรร เกียรติสุไพบูลย์, 98 หน้า , ISBN 974-14-1809-4

การวิจัยในครั้งนี้มีวัตถุประสงค์เพื่อพัฒนาเครื่องมือที่ใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA (Normal To Anything) และศึกษาวิธีการแก้ปัญหา NORTA defective ที่อาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ซึ่งประกอบด้วย 2 วิธีสำคัญ คือวิธี Semidefinite Programming (SDP) โดยกำหนดฟังก์ชันระยะทาง (distance function) เป็น L_1, L_2 และ L_∞ และวิธี Eigenvalue Shifting (ES) ซึ่งผลการวิจัยได้ข้อสรุปดังนี้

ในการวิจัยนี้ได้พัฒนาโปรแกรม NORTA ซึ่งสามารถใช้งานได้บนไมโครซอฟต์ เอ็กเซล โดยโปรแกรม ดังกล่าวใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรที่มีการแจกแจงขอบแบบสม่ำเสมอและมีสหสัมพันธ์กัน สำหรับการทำงานของโปรแกรมเริ่มจากการกำหนดเมทริกซ์สหสัมพันธ์ จากนั้นโปรแกรมจะทำการตรวจสอบคุณสมบัติของเมทริกซ์สหสัมพันธ์ที่กำหนดว่าเกิดปัญหา NORTA defective หรือไม่ ในกรณีที่เกิดปัญหาดังกล่าวจะทำการแก้ปัญหา หลังจากนั้นโปรแกรมจะทำการสร้างเวกเตอร์สุ่มหลายตัวแปรตามที่ต้องการ

และจากผลการศึกษาวิธีการที่ใช้แก้ปัญหา NORTA defective ซึ่งอาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ผู้วิจัยพบว่าวิธี SDP โดยการกำหนดฟังก์ชันระยะทางเป็น L_2 มีความเหมาะสมที่สุดสำหรับใช้แก้ปัญหาดังกล่าว ในขณะที่วิธี ES มีประสิทธิภาพมากที่สุดในการแก้ปัญหา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา สถิติ	ลายมือชื่อนิสิต จตุพล ทะน้อม
สาขาวิชา สถิติ	ลายมือชื่ออาจารย์ที่ปรึกษา [signature]
ปีการศึกษา 2548	

4682184026 : MAJOR STATISTICS

KEY WORD : NORTA METHOD/NORTA DEFECTIVE/FESIBLE CORRELATION MATRIX/SEMIDEFINITE PROGRAMMING/EIGENVALUE SHIFTING

CHATUPHOL THANOM : PROGRAM FOR GENERATING MULTIVARIATE RANDOM VECTORS BY NORTA AND IT'S CHARACTERISTICS. THESIS ADVISOR : SEKSAN KIATSUPAIBUL, PH.D. 98 pp.

ISBN 974-14-1809-4

The purpose of this research is development the process of generating multivariate random vectors by NORTA (Normal To Anything) and study how to solve NORTA defective problem that happen in the generating multivariate random vectors, process. There are two methods : Semidefinite Programming (SDP) with distance functions are L_1, L_2 and L_∞ and Eigenvalue Shifting (ES). The result of this research are as fallow :

In this case has developed NORTA Program that can used in Microsoft Excel. This program can generate multivariate random vectors, it all so multivariate uniform random vectors and have correlation. This program will arrange the correlation matrix. Then, will check NORTA defective problem what happen in this correlation matrix.

In this case study NORTA defective problem that was happen in the generating multivariate random vectors process in NORTA program. The analysis is found that the SDP with L_2 distance method is the suitable way to improve this problem. But the ES method have most efficience.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department Statistics

Student's signature จตุพล ทะนอม

Field of study Statistics

Advisor's signature

Academic year 2005

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความช่วยเหลือ คำแนะนำและข้อเสนอแนะ รวมไปถึงการเอาใจใส่อย่างใกล้ชิดจากอาจารย์ ดร.เสกสรร เกียรติสุไพบูลย์ ผู้เขียนขอน้อมกราบขอบพระคุณต่อท่านอาจารย์เป็นอย่างสูง ตลอดทั้งอาจารย์ทุกท่านที่ได้สอนผู้เขียนมา ณ โอกาสนี้ด้วย

ผู้เขียนขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.สุพล ดุรงค์วัฒนา และอาจารย์ ดร.อรุณี กำลัง ในฐานะประธานกรรมการและกรรมการสอบวิทยานิพนธ์ที่กรุณาให้คำแนะนำอันเป็นประโยชน์ในการแก้ไขวิทยานิพนธ์ฉบับนี้ให้สมบูรณ์ยิ่งขึ้น

ผู้เขียนขอขอบคุณเจ้าหน้าที่ห้องสมุด จากทางคณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย รวมถึงเครือข่ายห้องสมุดของจุฬาลงกรณ์มหาวิทยาลัยทุกคณะ ที่ได้อำนวยความสะดวกในด้านตำราซึ่งใช้ค้นคว้าประกอบการทำวิทยานิพนธ์ และขอขอบคุณ พี่ ๆ เพื่อน ๆ ทุกคน ที่ให้ความช่วยเหลือ และเป็นกำลังใจให้ผู้เขียน

สุดท้ายนี้ ผู้เขียนใคร่ขอกราบขอบพระคุณบิดา มารดา ที่ช่วยส่งเสริมและสนับสนุนให้ผู้วิจัยได้มีโอกาสทางการศึกษาเสมอมาจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ฌ
สารบัญรูปและแผนภูมิ	ฎ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	3
1.3 ขอบเขตของการวิจัย	3
1.4 นิยามคำศัพท์	3
1.5 ประโยชน์ที่คิดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย	5
2.1 การจำลองตัวแปรสุ่มร่วมแบบปกติที่สัมพันธ์กัน	6
2.2 วิธีการแยกเมทริกซ์แบบโชเลซกี	7
2.3 ตัวแบบจำลองด้วยวิธี NORTA	8
2.4 คุณสมบัติของเวกเตอร์ NORTA	9
2.5 วิธี semidefinite program (SDP)	11
2.5.1 วิธี SDP- L_1	14
2.5.2 วิธี SDP- L_2	16
2.5.3 วิธี SDP- L_∞	18
2.6 วิธี Eigenvalue Shifting (ES)	20
2.7 การหาค่าเฉพาะที่ต่ำสุดด้วยวิธี Power Method	21
2.8 การตรวจสอบเมทริกซ์กึ่งบวกแน่นอนด้วยวิธี Superdiagonalization	25
บทที่ 3 วิธีดำเนินการวิจัย	27
3.1 การสร้างเครื่องมือที่ใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA	27
3.2 การศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective	31
3.2.1 แผนการศึกษาวิจัย	31

หน้า

3.2.2 ขั้นตอนในการศึกษาวิจัย.. 32
บทที่ 4 ผลการวิจัย.. 40
4.1 โปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA………………… 40
4.2 ผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective 41
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ... 64
5.1 สรุปผลการวิจัย.. 64
5.2 การเลือกใช้วิธีการที่เหมาะสมในทางปฏิบัติสำหรับการแก้ปัญหา NORTA defective... 66
5.3 ข้อเสนอแนะ.. 66
รายการอ้างอิง.. 67
ภาคผนวก.. 69
ภาคผนวก ก โปรแกรม NORTA ที่พัฒนาด้วยภาษา C/C++ กรณี 3 ตัวแปร ......… 70
ภาคผนวก ข โปรแกรม Visual Basic บนไมโครซอฟต์ เอ็กเซล 80
ภาคผนวก ค การใช้งานโปรแกรมสำหรับสร้างเวกเตอร์ด้วยวิธี NORTA…………… 90
ประวัติผู้เขียนวิทยานิพนธ์.. 98

สารบัญตาราง

ตารางที่ | หน้า

4.1 แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้จากวิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES กับเวกเตอร์ NORTA defective สำหรับแต่ละมิติของ $\mathbf{\Lambda_Z}$ โดยกำหนดมาตรวัดเป็น L_1- norm 44

4.2 แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้จากวิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES กับเวกเตอร์ NORTA defective สำหรับแต่ละมิติของ $\mathbf{\Lambda_Z}$ โดยกำหนดมาตรวัดเป็น L_2- norm 46

4.3 แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้จากวิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES กับเวกเตอร์ NORTA defective สำหรับแต่ละมิติของ $\mathbf{\Lambda_Z}$ โดยกำหนดมาตรวัดเป็น L_∞- norm 48

4.4 แสดงระยะทางโดยเฉลี่ยระหว่าง $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞- norm........ 51

4.5 แสดงระยะทางโดยเฉลี่ยระหว่าง $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{X}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_\mathbf{X}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞- norm........ 51

4.6 แสดงระยะทางระหว่าง $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_1- norm 54

4.7 แสดงระยะทางระหว่าง $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{X}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_\mathbf{X}$ โดยกำหนดมาตรวัดเป็น L_1- norm 54

4.8 แสดงระยะทางระหว่าง $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_2 - norm 56

4.9 แสดงระยะทางระหว่าง $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{X}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_\mathbf{X}$ โดยกำหนดมาตรวัดเป็น L_1- norm 57

4.10 แสดงระยะทางระหว่าง $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_2 - norm 58

4.11 แสดงระยะทางระหว่าง $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{X}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_\mathbf{X}$ โดยกำหนดมาตรวัดเป็น L_1- norm 59

4.12 แสดงเวลาโดยเฉลี่ยที่ใช้ในการแก้ปัญหา NORTA defective ของแต่ละวิธีการ โดยกำหนดเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาจำนวน 1,000 เมทริกซ์ สำหรับแต่ละมิติ (dimension) 60

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

4.13 แสดงผลการจำลองเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA จากโปรแกรม NOTRA ที่ได้พัฒนาขึ้น ซึ่งเป็นโปรแกรม Add-in โดยใช้งานบนไมโครซอฟต์ เอ็กเซล 63

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญรูปและแผนภูมิ

รูปที่ หน้า

4.1 แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{u}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ที่ได้จากวิธี SDP- L_1 ,SDP- L_2 ,SDP- L_∞ และ ES ตามลำดับ โดยกำหนดมาตรวัดเป็น L_1 - norm 45

4.2 แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{u}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ที่ได้จากวิธี SDP- L_1, SDP- L_2, SDP- L_∞ และ ES ตามลำดับ โดยกำหนดมาตรวัดเป็น L_2 - norm........ 47

4.3 แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{u}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ที่ได้จากวิธีSDP- L_1 ,SDP- L_2 ,SDP- L_∞ และ ES ตามลำดับ โดยกำหนดมาตรวัดเป็น L_∞ - norm 49

4.4 แสดงเวลาโดยเฉลี่ยที่ใช้ในการแก้ปัญหา NORTA defective ของแต่ละวิธีการ โดยกำหนดเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาจำนวน 1,000 เมทริกซ์ 61

ค1 แสดงส่วนประกอบของโปรแกรมสำหรับสร้างเวกเตอร์ด้วยวิธี NORTA ซึ่งถูกเก็บไว้ในโฟลเดอร์ ProgramNORTA 90

ค2 แสดงตัวอย่างคัดลอกโฟลเดอร์ ProgramNORTA วางไว้ในไดร์ฟ D 91

ค3 แสดง worksheet : InputData และ worksheet : OutputData ตามลำดับ 92

ค4 แสดงหน้าต่าง Project–VBAProject และหน้าต่างลงรหัส (coding window) ตามลำดับ 93

ค5 แสดงการปรับแก้รหัสในหน้าต่างลงรหัส (coding window) ในกรณีที่เก็บ โฟลเดอร์ NORTAprogram ไว้ที่ไดร์ฟ D 94

ค6 แสดงตัวอย่างการจำลองเวกเตอร์สุ่มด้วยวิธี NORTA กรณี 5 ตัวแปร 95

ค7 แสดงผลการจำลองเวกเตอร์สุ่มด้วยวิธี NORTA กรณี 5 ตัวแปร 96

ค8 แสดงกรอบโต้ตอบแจ้งเตือนในกรณี Correlation Matrix ที่ผู้ใช้กำหนดขึ้นไม่มีคุณสมบัติเป็นเมทริกซ์กึ่งบวกแน่นอนหรือเมริกซ์บวกแน่นอน 97

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

สำหรับการศึกษาวิเคราะห์ระบบ เพื่อให้เข้าใจความถึงความสัมพันธ์ระหว่างองค์ประกอบต่างๆ ของระบบ เพื่อที่จะทำการออกแบบระบบหรือพยากรณ์ผลการดำเนินงานของระบบจากการใช้นโยบายหรือกลยุทธ์ขององค์กรนั้น ๆ ในบางครั้งอาจจะเป็นการยากถ้าหากต้องทำการทดลองกับระบบจริงที่มีอยู่ ทั้งนี้อาจจะต้องเสียเวลาและค่าใช้มากเกินความจำเป็น ดังนั้นในการศึกษาวิเคราะห์หรือออกแบบระบบบ่อยครั้งจะทำโดยการศึกษาทดลองกับตัวแบบจำลอง (simulation model) ของระบบ

โดยทั่วไปแล้ว ในการศึกษาระบบจริง มักจะพบข้อเท็จจริงที่ว่า กระบวนการเข้า (input process) ของระบบหนึ่ง ๆ มักจะมีองค์ประกอบที่เกี่ยวเนื่องหรือสัมพันธ์กัน (dependent component) ตัวอย่างของระบบเช่น การผลิตรถยนต์ จะมีเครื่องจักรต่าง ๆ อุปกรณ์ต่าง ๆ วัตถุดิบ คนงาน เงินทุน ฯลฯ เป็นองค์ประกอบ (components) ซึ่งองค์ประกอบเหล่านี้มีกิจกรรมที่เกี่ยวเนื่องหรือสัมพันธ์กัน โดยมีวัตถุประสงค์คือผลิตรถยนต์ให้ได้ตามที่ต้องการ ดังนั้น ในการทดลองโดยใช้การจำลอง (simulation experiments) จึงจำเป็นที่จะต้องกำหนดความสัมพันธ์ขององค์ประกอบต่าง ๆ ซึ่งถือเป็นตัวแปรสุ่ม (random variable) ให้กับตัวแบบจำลองของระบบด้วย ด้วยเหตุผลดังกล่าวนี้ผู้วิจัยจึงมีความสนใจที่จะสร้างเครื่องมือสำหรับสร้างตัวแปรสุ่มหลายตัวแปรที่มีความสัมพันธ์กัน (correlated random variables) ให้กับตัวแบบจำลอง (simulation model) ของระบบ

ในปี ค.ศ. 1997 cario และ nelson ได้นำเสนอวิธีนอร์ทา (NORTA ,NORmal To Anything) ซึ่งเป็นวิธีการที่ใช้สร้างเวกเตอร์สุ่มหลายตัวแปร(multivariate random vector) ด้วยการกำหนดสองคุณลักษณะที่สำคัญของเวกเตอร์สุ่มหลายตัวแปรที่ต้องการ คือการแจกแจงขอบ (marginal distribution) $F_1,\ldots,F_n$ และเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ของเวกเตอร์สุ่ม (random vector) $\mathbf{X}=\left(X_1,\ldots,X_n\right)'$ ที่ต้องการ

อย่างไรก็ตาม ในปี ค.ศ. 2001 Ghosh และ Henderson ได้ชี้ให้เห็นถึงปัญหาสำคัญที่เกิดขึ้นจากวิธี NORTA โดยพบว่า ในเมทริกซ์สหสัมพันธ์บางเมทริกซ์จะนำไปสู่ความล้มเหลวสำหรับการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA **เมื่อมิติของเวกเตอร์สุ่มที่ต้องการสร้างด้วยวิธีนี้เพิ่มขึ้น** โดยเรียกปัญหาที่เกิดขึ้นนี้ว่า NORTA defective สิ่งที่ผู้วิจัยสนใจศึกษาคือการแก้ปัญหา

ในกรณีที่เกิด NORTA defective ซึ่งอาจเกิดขึ้นในการบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA โดย Ghosh และ Henderson นำเสนอวิธี Semidefinite Programming (SDP) โดยนิยามความใกล้ระหว่างสองเมทริกซ์ใด ๆ ด้วยฟังก์ชันระยะทาง (distance function) L_1 และ L_∞

ในการศึกษาวิจัยครั้งนี้ ผู้วิจัยได้แบ่งการดำเนินการวิจัยออกเป็น 2 ส่วนสำคัญ คือ

ส่วนที่ 1 พัฒนาโปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ให้สามารถนำไปใช้งานได้จริง

ส่วนที่ 2 ทำการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective ซึ่งเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ประกอบด้วย 2 วิธีการคือ

2.1) วิธี Semidefinite Programming (SDP) โดยทำการนิยามความใกล้เคียงระหว่างสองเมทริกซ์ใด ๆ ด้วยฟังก์ชันระยะทาง (distance function) L_1, L_2 และ L_∞ ซึ่ง L_1 และ L_∞ นำเสนอโดย Ghosh และ Henderson ในปี ค.ศ. 2001 ในขณะที่ L_2 ผู้วิจัยได้อ้างอิงจากงานวิจัยของ ดร.เสกสรร เกียรติสุไพบูลย์และ Dolchai La-ornual ในปี ค.ศ. 2004

2.2) วิธี Eigenvalue Shifting (ES) ซึ่งเป็นวิธีการที่ใช้แก้ปัญหากรณีที่เมทริกซ์สหสัมพันธ์ที่กำหนดขึ้นไม่เป็นเมทริกซ์กึ่งบวกแน่นอน (non-positive semidefinite matrix) ในโปรแกรม @RISK ซึ่งเป็นโปรแกรมสำหรับใช้งานบนไมโครซอฟต์ เอ็กเซล

ท้ายที่สุดแล้วในการวิจัยครั้งนี้ ต้องการที่จะพัฒนาโปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA รวมไปถึงการศึกษาถึงความแตกต่างในแต่ละวิธี ที่ใช้แก้ปัญหาที่เกิดขึ้นวิธี NORTA ทั้งนี้เพื่อที่จะพิจารณาเลือกวิธีการที่เหมาะสมที่สุดให้กับโปรแกรมที่ได้สร้างขึ้นโดยมุ่งหวังให้นำมาซึ่งคุณประโยชน์และการพัฒนาในด้านต่าง ๆ ต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.2 วัตถุประสงค์การวิจัย

เพื่อพัฒนาเครื่องมือที่ใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ให้สามารถใช้งานได้จริง และศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective จากสองวิธีการคือ วิธี SDP และวิธี ES ดังที่ได้กล่าวไว้แล้วข้างต้น

1.3 ขอบเขตการวิจัย

การศึกษาวิจัยนี้ ผู้วิจัยจะสร้างโปรแกรม NORTA ใช้งานบนไมโครซอฟฟ์ วินโดว์ (Microsoft Windows) โดยเฉพาะสามารถประยุกต์ให้ใช้งานกับไมโครซอฟฟ์ เอ็กเซลได้ และศึกษาความแตกต่างของการแก้ไขปัญหา NORTA defective โดยใช้วิธี SDP และวิธี ES โดยพิจารณาความใกล้เคียงระหว่างเมทริกซ์ที่ได้จากสองวิธีการดังกล่าวด้วยการกำหนดมาตรวัดระยะ (distance measure) 3 แบบ คือ L_1 - norm, L_2 - norm และ L_∞ - norm ตามลำดับ

1.4 นิยามคำศัพท์

NORTA (NORmal-To-Anything) เป็นวิธีการที่ใช้สร้างเวกเตอร์สุ่มหลายตัวแปร โดยทำการสร้างเวกเตอร์สุ่มแบบปกติมาตรฐาน (standard normal random vector) $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_d)'$ แล้วทำการแปลง (transformation) เวกเตอร์สุ่ม $\mathbf{Z}$ ไปเป็นเวกเตอร์สุ่ม $\mathbf{X}$ ที่มีการแจกแจงขอบ (marginal distribution) ตามที่ต้องการ

เมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) เป็นเมตริกซ์สหสัมพันธ์ที่มีคุณสมบัติของเมทริกซ์ไม่เป็นลบแน่นอน (non-negative definite matrix)

NORTA defective คือกรณีที่เมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ของเวกเตอร์สุ่ม $\mathbf{X}$ ที่ต้องการคือ $\mathbf{\Sigma_X}$ ซึ่งกำหนดขึ้นเพื่อสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA โดยหลังจากทำการแปลง (transformation) เมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma_X}$ ไปเป็นเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่มแบบปกติมาตรฐาน $\mathbf{\Sigma_Z}$ แล้ว พบว่า $\mathbf{\Sigma_Z}$ ไม่เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ทำให้ไม่สามารถสร้างเวกเตอร์สุ่มที่ต้องการด้วยวิธี NORTA ได้

Semidefinite Programming (SDP) เป็นวิธีการหาคำตอบของปัญหาที่เป็น convex optimization problem โดยที่ฟังก์ชันวัตถุประสงค์ (objective function) เป็นฟังก์ชันเชิงเส้น (linear function) และข้อจำกัด (constraint) ซึ่งมีคุณสมบัติของเมทริกซ์กึ่งบวกแน่นอน (positive semidifinite matrix)

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) โปรแกรม NORTA สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรที่มีประสิทธิภาพและมีความถูกต้องเหมาะสม สามารถนำไปใช้งานจริงได้

2) เพื่อเป็นแนวทางในการพัฒนาการสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA แก่ผู้ที่สนใจทั่วไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

ในการศึกษาวิจัยครั้งนี้ ผู้วิจัยได้กำหนดสัญลักษณ์ของเมทริกซ์สหสัมพันธ์ของตัวแปรสุ่มที่ใช้ทำการศึกษาเป็นดังนี้

1. กำหนดให้ $\mathbf{\Sigma_X}$ แทนเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X} = (X_1, X_2, \ldots, X_d)^T$ ที่มีการแจกแจงขอบแบบสม่ำเสมอ (uniform marginal distribution) โดยให้ $\Sigma_X(i,j)$ แทนสหสัมพันธ์ระหว่าง X_i กับ X_j สำหรับ $i \neq j$

2. กำหนดให้ $\mathbf{\Sigma_Z}$ แทนเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_d)^T$ ที่มีการแจกแจงขอบแบบปกติมาตรฐาน (standard normal marginal distribution) โดยให้ $\Sigma_Z(i,j)$ แทนสหสัมพันธ์ระหว่าง Z_i กับ Z_j สำหรับ $i \neq j$

3. กำหนดให้ $\mathbf{\Lambda_Z}$ แทนเมทริกซสหสัมพันธ์โดยประมาณของ $\mathbf{\Sigma_Z}$ โดยให้ $\Lambda_Z(i,j)$ แทนสหสัมพันธ์โดยประมาณของ $\Sigma_Z(i,j)$

โดยที่ $\mathbf{\Sigma_X}$, $\mathbf{\Sigma_Z}$ และ $\mathbf{\Lambda_Z}$ จะต้องเป็นเมทริกซ์ไม่เป็นลบแน่นอน (non-negative definite matrix)

สำหรับทฤษฎีที่เกี่ยวข้องกับงานวิจัย ได้นำเสนอเป็นลำดับดังต่อไปนี้

2.1 การจำลองตัวแปรสุ่มร่วมแบบปกติที่สัมพันธ์กัน (generating correlated normal random variables)

กำหนดให้ตัวแปรสุ่มร่วมแบบปกติ d ตัวแปร $Y_1, Y_2, \ldots, Y_d$ หรือเวกเตอร์สุ่ม $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_d)^T$ ที่มีการแจกแจงแบบปกติหลายตัวแปร(multivariate normal distribution) เขียนแทนด้วย $\mathbf{Y} \sim N_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ สามารถอธิบายคุณสมบัติด้วยเวกเตอร์ค่าเฉลี่ย (mean vector) $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_d)^T$ ซึ่ง $\mu_i = \mathrm{E}(Y_i),\ i = 1,2,\ldots,d$ และเมทริกซ์ความแปรปรวนร่วม (covariance matrix) $\boldsymbol{\Sigma}$ ขนาด $d \times d$ ซึ่งเป็นเมทริกซ์บวกแน่นอน (positive definite matrix) มีฟังก์ชันการแจกแจงความน่าจะเป็นร่วม (joint probability density function) เป็น

$$f(\mathbf{y}) = (2\pi|\boldsymbol{\Sigma}|)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right)$$

$$; -\infty < y_i < \infty,\ i = 1,2,\ldots,d$$

ซึ่ง $\mathbf{y} = (y_1, y_2, \ldots, y_d)^T$,

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dd} \end{bmatrix}$$

โดยที่ $\sigma_{ij} = \mathrm{Cov}(Y_i, Y_j) = \mathrm{Cov}(Y_j, Y_i) = \sigma_{ji}$ สำหรับ $i \neq j$ และ $\sigma_{ii} = \mathrm{Var}(Y_i)$; $i = 1,2,\ldots,d$, $j = 1,2,\ldots,d$, $|\boldsymbol{\Sigma}|$ คือดีเทอร์มิแนนต์ (determinant) ของ $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}^{-1}$ คือเมทริกซ์ผกผัน (inverse matrix) ของ $\boldsymbol{\Sigma}$ และ $(\mathbf{y} - \boldsymbol{\mu})^T$ คือเวกเตอร์ทราสโพส (transpose vector) ของเวกเตอร์ $(\mathbf{y} - \boldsymbol{\mu})$

เนื่องจาก $\boldsymbol{\Sigma}$ เป็นเมทริกซ์บวกแน่นอน ดังนั้น สามารถเขียน $\boldsymbol{\Sigma}$ ได้เป็น

$\boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^T$ โดย Cholesky decompositon

โดยที่ $\mathbf{C}$ เป็นเมทริกซ์สามเหลี่ยมล่าง (lower triangular matrix) ซึ่ง

$$\mathbf{C} = \begin{bmatrix} c_{11} & 0 & 0 & \cdots & 0 \\ c_{21} & c_{22} & 0 & \cdots & 0 \\ c_{31} & c_{32} & c_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{d1} & c_{d2} & c_{d3} & \cdots & c_{dd} \end{bmatrix}$$

กำหนดให้ $\mathbf{Z} = (\mathrm{Z}_1, \mathrm{Z}_2, \ldots, \mathrm{Z}_d)^T$ โดยที่ $Z_1, Z_2, \ldots, Z_d$ เป็นอิสระต่อกันและต่างมีการแจกแจงแบบปกติมาตรฐาน (standard normal distribution)

จากนั้น ทำการสร้างตัวแปรสุ่มใช้การแปลงเชิงเส้น (linear transformation) โดย

$$Y_i = \sum_{j=1}^{d} c_{ij} Z_j + \mu_i \quad , i = 1,2,\ldots,d$$

หรือเขียนอยู่ในรูปสัญลักษณ์เมทริกซ์ได้เป็น

$$\mathbf{Y} = \mathbf{CZ} + \boldsymbol{\mu}$$

จะได้ $\mathbf{Y}$ มีการแจกแจงแบบปกติหลายตัวแปร (multivariate normal distribution) ซึ่งมี $\mathrm{E}(\mathbf{Y}) = \boldsymbol{\mu}$ และ $\mathrm{Var}(\mathbf{Y}) = \mathbf{C}\mathrm{Var}(\mathbf{Z})\mathbf{C}^T = \mathbf{C}\mathbf{I}_d\mathbf{C}^T = \mathbf{C}\mathbf{C}^T = \boldsymbol{\Sigma}$ โดยที่ $\mathbf{I}_d$ เป็นเมทริกซ์เอกลักษณ์ (identity matrix) ขนาด $d \times d$

<u>หมายเหตุ</u> เนื่องจากเมทริกซ์ความแปรปรวนร่วม (covariance matrix) ของเวกเตอร์สุ่ม $\mathbf{Z} = (\mathrm{Z}_1, \mathrm{Z}_2, \ldots, \mathrm{Z}_d)^T$ คือ $\boldsymbol{\Sigma}_{\mathbf{Z}}$ ซึ่ง $Z_1, Z_2, \ldots, Z_d$ ต่างมีการแจกแจงแบบปกติมาตรฐาน (standard normal distribution) โดยที่ $\boldsymbol{\Sigma}_{\mathbf{Z}} = (\boldsymbol{\Sigma}_{\mathbf{Z}}(i,j) : 1 \le i, j \le d)$ และ $\boldsymbol{\Sigma}_{\mathbf{Z}}(i,i) = 1$ สำหรับ $i = 1,2,\ldots,d$ ซึ่งเป็นเมทริกซ์เดียวกันกับเมทริกซ์สหสัมพันธ์ (correlation matrix) ของเวกเตอร์สุ่ม $\mathbf{Z}$ ดังนั้นในงานวิจัยนี้ ผู้วิจัยจึงใช้สัญลักษณ์เมทริกซ์ $\boldsymbol{\Sigma}_{\mathbf{Z}}$ แทนเมทริกซ์สหสัมพันธ์ (correlation matrix) ของเวกตอร์สุ่ม $\mathbf{Z}$ ซึ่งอีกนัยหนึ่ง ก็คือเมทริกซ์ความแปรปรวนร่วมของเวกเตอร์สุ่ม $\mathbf{Z}$ นั่นเอง

2.2 **วิธีการแยกเมทริกซ์แบบโชเลซกี** (cholesky decomposition)

สำหรับการวิจัยครั้งนี้ ใช้วิธีการแยกเมทริกซ์แบบโชเลซกี เพื่อใช้ในกระบวนการสร้างตัวแปรสุ่มที่มีการแจกแจงแบบปกติมาตรฐานหลายตัวแปรที่สัมพันธ์กัน โดยมีทฤษฎีบทดังนี้

<u>ทฤษฎีบทที่ 1</u> การแยกเมทริกซ์แบบโชเลซกี (cholesky cecomposition)

ถ้าเมทริกซ์ $\mathbf{M}$ มีคุณสมบัติเป็นเมทริกซ์บวกแน่นอน (positive definite matrix) และเป็นเมทริกซ์สมมาตร (symmetric matrix) แล้ว $\mathbf{M}$ สามารถแยก (decompose) เป็นเมทริกซ์สามเหลี่ยมล่าง (lower triangular matrix) ที่องค์ประกอบเป็นเลขจำนวนจริง นั่นคือ $\mathbf{M} = \mathbf{C}\mathbf{C}^T$ โดยที่ $\mathbf{C}$ เป็นเมทริกซ์สามเหลี่ยมล่าง (lower triangular matrix)

ซึ่งสามารถสรุปสูตรในการคำนวณหา c_{ij} สำหรับเมทริกซ์สามเหลี่ยมล่าง $\mathbf{C}$ ได้ดังนี้

$$c_{ij} = \frac{\sigma_{ij} - \sum_{k=1}^{j-1} c_{ik} c_{jk}}{c_{jj}} \quad , 1 \le j < i \le d$$

โดยที่

$$c_{jj} = \sqrt{\sigma_{jj} - \sum_{k=1}^{j-1} c_{jk}^2} \quad , \ j = 1,2,\ldots,d$$

$$\sum_{k=1}^{j-1} c_{ik} c_{jk} = 0$$

2.3 ตัวแบบจำลองด้วยวิธี NORTA

โดยเป้าหมายของตัวแบบจำลองด้วยวิธี NORTA คือการสร้างเวกเตอร์สุ่ม $\mathbf{X} = (X_1, X_2, \ldots, X_d)^T$ ให้มีสองคุณสมบัติสำคัญที่ต้องการคือ

1. $X_i \sim F_{X_i}, i = 1,2,\ldots,d$ โดยแต่ละ F_{X_i} เป็นฟังก์ชันการแจกแจงสะสม (cumulative distribution function) ที่กำหนดขึ้น

2. เมทริกซ์สหสัมพันธ์ของ $\mathbf{X}$ เขียนแทนด้วย $\mathbf{\Sigma_X}$ ซึ่งเป็นเมทริกซ์สหสัมพันธ์ที่ต้องการ

โดย $\mathbf{X}$ ได้มาจากการแปลงเวกเตอร์ (vector transformation) d มิติ นั่นคือการแปลง $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_d)^T$ ซึ่งประกอบด้วยตัวแปรสุ่มแบบปกติมาตรฐานหลายตัวแปร (standard multivariate normal variables) และเมทริกซ์สหสัมพันธ์ของ $\mathbf{Z}$ เขียนแทนด้วย $\mathbf{\Sigma_Z}$ ดังนั้นเวกเตอร์ $\mathbf{X}$ ที่ได้จากวิธี NORTA คือ

$$\mathbf{X} = \begin{bmatrix} F_{X_1}^{-1}[\mathbf{\Phi}(Z_1)] \\ F_{X_2}^{-1}[\mathbf{\Phi}(Z_2)] \\ \vdots \\ F_{X_d}^{-1}[\mathbf{\Phi}(Z_d)] \end{bmatrix}$$

โดยที่ $\mathbf{\Phi}$ คือฟังก์ชันการแจกแจงสะสม (cumulative distribution function) ของตัวแปรสุ่มที่แบบปกติมาตรฐาน (standard normal random variable) และ $F_{\mathbf{X}}^{-1}(\mathbf{u}) \equiv \inf\{\mathbf{x} : F_{\mathbf{X}}(\mathbf{x}) \ge \mathbf{u}\}$ คือฟังก์ชันผกผันของการแจกแจงสะสม (inverse cumulative distribution function)

การแปลงด้วย $F_{X_i}^{-1}[\mathbf{\Phi}(\cdot)]$ จะได้ X_i ที่มีการแจกแจงขอบ (marginal distribution) F_{X_i} ตามที่ต้องการ เพราะฉะนั้น สิ่งที่สำคัญอย่างยิ่งสำหรับการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA คือการเลือก $\mathbf{\Sigma_Z}$ ที่ทำให้ได้ $\mathbf{\Sigma_X}$ ตามที่กำหนดไว้

2.4 **คุณสมบัติของเวกเตอร์** NORTA (properties of NORTA vectors)

สำหรับ $i \neq j$ กำหนดให้ $\Sigma_Z(i, j)$ เป็นสมาชิกแถวที่ i หลักที่ j ของ $\mathbf{\Sigma_Z}$ และ $\Sigma_X(i, j)$ เป็นสมาชิกแถวที่ i หลักที่ j ของ $\mathbf{\Sigma_X}$ ซึ่งเมทริกซ์สหสัมพันธ์ของ $\mathbf{X}$ จะถูกกำหนดโดยตรงจากเมทริกซ์สหสัมพันธ์ของ $\mathbf{Z}$ กล่าวคือ

$$\Sigma_X(i, j) = \mathrm{Corr}\left[X_i, X_j\right] = \mathrm{Corr}\left\{F_{X_i}^{-1}[\mathbf{\Phi}(Z_i)], F_{X_j}^{-1}\left[\mathbf{\Phi}\left(Z_j\right)\right]\right\}$$

ซึ่งสามารถเขียนให้อยู่ในรูปของ $\mathrm{E}[X_i X_j]$ ได้ดังนี้

$$\mathrm{Corr}\left[X_i, X_j\right] = \frac{\mathrm{E}[X_i X_j] - \mathrm{E}[X_i]\mathrm{E}[X_j]}{\sqrt{\mathrm{Var}[X_i]\mathrm{Var}[X_j]}}$$

โดย $\mathrm{E}[X_i], \mathrm{E}[X_i], \mathrm{Var}[X_i]$ และ $\mathrm{Var}[X_j]$ กำหนดได้โดยตรงจาก F_{X_i} และ F_{X_j} และให้ $\left(Z_i, Z_j\right)$ มีการแจกแจงแบบปกติมาตรฐานสองตัวแปร (standard bivariate normal distribution) ที่มีสหสัมพันธ์ $\mathrm{Corr}[Z_i, Z_j] = \Sigma_Z(i, j)$ ดังนั้น จะได้

$$\begin{aligned}\mathrm{E}[X_i, X_j] &= \mathrm{E}\left\{F_{X_i}^{-1}[\mathbf{\Phi}(Z_i)], F_{X_j}^{-1}\left[\mathbf{\Phi}\left(Z_j\right)\right]\right\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_{X_i}^{-1}[\mathbf{\Phi}(Z_i)], F_{X_j}^{-1}\left[\mathbf{\Phi}\left(Z_j\right)\right]\varphi_{\Sigma_Z(i,j)}\left(z_i, z_j\right)dz_i dz_j\end{aligned}$$

โดยที่ $\varphi_{\rho_Z(i,j)}$ คือฟังก์ชันความหนาแน่นของความน่าจะเป็นแบบปกติมาตรฐานสองตัวแปรที่มีสหสัมพันธ์เป็น $\Sigma_Z(i, j)$

ซึ่งในปี ค.ศ. 1985 Kruskal ได้แสดงฟังก์ชันความสัมพันธ์ระหว่าง $\mathbf{\Sigma_Z}$ กับ $\mathbf{\Sigma_X}$ เป็น

$$\Lambda_Z(i, j) = 2\sin\left(\frac{\pi}{6}\Sigma_X(i, j)\right) \qquad \ldots (2.1)$$

โดยที่ $\Lambda_Z(i, j)$ เป็นสหสัมพันธ์โดยประมาณของ $\Sigma_Z(i, j)$ และทำการประมาณเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma_Z}$ ด้วย $\mathbf{\Lambda_Z}$ โดยที่ $\mathbf{\Sigma_Z}$ และ $\mathbf{\Lambda_Z}$ จะต้องเป็นเมทริกซ์ไม่เป็นลบแน่นอน (non-negative definite matrix)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ในปี ค.ศ. 1972 Li และ Hammond แสดงให้เห็นว่าฟังก์ชันความสัมพันธ์ใน (2.1) ไม่อาจรับประกันได้ว่าสำหรับทุกเมทริกซ์ $\mathbf{\Lambda_Z}$ ที่ได้จะมีคุณสมบัติเป็นเมทริกซ์ที่ไม่เป็นลบแน่นอน ซึ่งได้ยกตัวอย่างกรณีเฉพาะที่เกิดปัญหา โดยสมมติให้เวกเตอร์สุ่ม $\mathbf{X} = (X_1, X_2, X_3)^T$ มีการแจกแจงขอบแบบสม่ำเสมอ (0,1] และมีเมทริกซ์สหสัมพันธ์เป็น

$$\mathbf{\Sigma_X} = \begin{bmatrix} 1 & -0.4 & 0.2 \\ -0.4 & 1 & 0.8 \\ 0.2 & 0.8 & 1 \end{bmatrix}$$

หลังจากนั้นทำการแปลงด้วย (2.1) จะได้

$$\mathbf{\Lambda_Z} = \begin{bmatrix} 1 & -0.41582338163552 & 0.20905692653531 \\ -0.41582338163552 & 1 & 0.8134732861516 \\ 0.20905692653531 & 0.8134732861516 & 1 \end{bmatrix}$$

ซึ่งพบว่าเมทริกซ์ $\mathbf{\Lambda_Z}$ ที่ได้ไม่มีคุณสมบัติเป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ โดยมีเวกเตอร์เฉพาะ (eigen vector) เป็น (-0.00920482, 1.16625071, 1.8429541) กล่าวคือ $\mathbf{\Lambda_Z}$ ไม่เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ทำให้ไม่สามารถใช้ $\mathbf{\Lambda_Z}$ เพื่อทำการประมาณเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma_Z}$ ได้ ซึ่งถือเป็นปัญหาสำคัญที่พบในกระบวนสร้างเวกเตอร์สุ่มด้วยวิธี NORTA โดยเรียกปัญหาที่เกิดขึ้นนี้ว่า NORTA defective และจะพบปัญหาดังกล่าวเพิ่มขึ้นเป็นทวีคูณเมื่อมิติของเวกเตอร์สุ่มที่ต้องการสร้างด้วยวิธี NORTA เพิ่มขึ้น

สำหรับในงานวิจัยนี้ ได้ทำการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective หรือกรณีที่ $\mathbf{\Lambda_Z}$ ไม่มีคุณสมบัติของเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ดังกล่าว ซึ่งประกอบด้วย 2 วิธีที่สำคัญ คือวิธี Semidefinite Programming (SDP) โดยอ้างอิงจากงานวิจัยของ Ghosh ในปี ค.ศ. 2001 และวิธี Eigenvalue Shifting (ES) ซึ่งเป็นวิธีที่ใช้ในโปรแกรม @RISK ซึ่งเป็นโปรแกรม Add-In สำหรับใช้งานบนไมโครซอฟต์ เอ็กเซล (Microsoft Excel) ดังนำเสนอรายละเอียดเป็นลำดับต่อไปนี้

2.5 วิธี semidefinite program (SDP)

จากงานวิจัยของ Vandenberghe และ Boyd ในปี ค.ศ. 1996 ได้พัฒนารูปแบบมาตรฐานสำหรับการแก้ปัญหาด้วยวิธี SDP ดังนี้

$$\begin{aligned} &\text{minimize} \quad \mathbf{c}^T\mathbf{x} \\ &\text{subject to} \quad \mathbf{F}(\mathbf{x}) \succeq 0 \end{aligned}$$

ซึ่ง $$\mathbf{F}(\mathbf{x}) \triangleq \mathbf{F}_0 + \sum_{i=1}^{n} x_i \mathbf{F}_i \qquad \ldots(2.2)$$

โดยที่ $\mathbf{c}$ และ $\mathbf{x} = [x_i]$ อยู่บน $\Re^n$ และ $\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_n$ เป็นเมทริกซ์ขนาด $d \times d$ ใน $\Re^{d\times d}$

สำหรับในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ถ้า $\mathbf{\Lambda_Z}$ ใน (2.1) ไม่เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (infeasible correlation matrix) สิ่งที่ต้องการคือการหา $\mathbf{\Sigma_Z}$ ที่ใกล้เคียงกับ $\mathbf{\Lambda_Z}$ มากที่สุด

ในปี ค.ศ. 2001 Ghosh และ Henderson ได้นำเสนอวิธี SDP ซึ่งมีรายละเอียดดังนี้

กำหนดให้ $r(\mathbf{\Sigma'_Z}, \mathbf{\Lambda_Z})$ เป็นฟังก์ชันระยะทางน้อยที่สุด ระหว่าง $\mathbf{\Lambda_Z}$ กับ $\mathbf{\Sigma'_Z}$ โดยที่ $\mathbf{\Sigma'_Z}$ เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ใด ๆ แล้ว ทำการหา $\mathbf{\Sigma'_Z}$ ที่ทำให้ $r(\mathbf{\Sigma'_Z}, \mathbf{\Lambda_Z})$ มีค่าน้อยที่สุดด้วยวิธี SDP ซึ่งรูปแบบทั่วไปดังนี้

$$\begin{aligned} &\text{minimize} \quad r(\mathbf{\Sigma'_Z}, \mathbf{\Lambda_Z}) \\ &\text{subject to} \quad \mathbf{\Sigma'_Z} \succeq \mathbf{0}\ , \\ &\qquad \Sigma'_Z(i,j) = \Sigma'_Z(j,i) \ \text{และ} \\ &\qquad \Sigma'_Z(i,i) = 1 \end{aligned} \qquad \ldots(2.3)$$

เมื่อกำหนดให้ $\mathbf{\Sigma'_Z} \succeq \mathbf{0}$ หมายความว่า $\mathbf{\Sigma'_Z}$ เป็นเมทริกซ์กึ่งบวกแน่นอน (positive semidefinite matrix)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

กำหนดให้ฟังก์ชันระยะทาง (distance function) สำหรับวิธี SDP 3 ฟังก์ชัน คือ

SDP-L_1 มีฟังก์ชันระยะทางเป็น $r_1(\mathbf{\Sigma}'_\mathbf{Z}, \mathbf{\Lambda}_\mathbf{Z}) = \sum_{i>j} \left| \Sigma'_Z(i,j) - \Lambda_Z(i,j) \right|$,

SDP-L_2 มีฟังก์ชันระยะทางเป็น $r_2(\mathbf{\Sigma}'_\mathbf{Z}, \mathbf{\Lambda}_\mathbf{Z}) = \sum_{i>j} (\Sigma'_Z(i,j) - \Lambda_Z(i,j))^2$

และ SDP-L_∞ มีฟังก์ชันระยะทางเป็น $r_\infty(\mathbf{\Sigma}'_\mathbf{Z}, \mathbf{\Lambda}_\mathbf{Z}) = \max_{i>j} \{ \left| \Sigma'_Z(i,j) - \Lambda_Z(i,j) \right| \}$

เมื่อ $1 \le i, j \le d$ โดยที่ SDP-L_1, SDP-L_∞ อ้างอิงจากงานวิจัยของ Ghosh และ Henderson ในปี ค.ศ. 2001 ส่วน SDP-L_2 อ้างอิงจากงานวิจัยของ ดร. เสกสรร เกียรติสุไพบูลย์ และ Dolchai La-ornual ในปี ค.ศ. 2004

จากที่กำหนดให้ฟังก์ชันการแจกแจงขอบ (marginal distribution function) เป็น $F_1, F_2, \ldots F_d$ ซึ่งจะมีเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma}_\mathbf{X}$ ขนาด $d \times d$ ดังนั้นปริภูมิเวกเตอร์สหสัมพันธ์ (correlation vector space) ของ $\mathbf{\Sigma}_\mathbf{X}$ ที่ใช้ทำการศึกษาจะมีขนาด $n = \frac{d(d-1)}{2}$ มิติ (dimension) ซึ่งก็คือจำนวนของสมาชิกที่อยู่เหนือ(หรือล่าง) ของสมาชิกที่อยู่ในแนวเส้นทแยงมุมหลัก (main diagonal) ของ $\mathbf{\Sigma}_\mathbf{X}$ นั่นเอง (เนื่องจาก $\mathbf{\Sigma}_\mathbf{X}$ เป็นเมทริกซ์สมมาตร (symmetric matrix) และมีสมาชิกในแนวเส้นทแยงมุมหลักเป็น 1)

กำหนดให้ $\mathbf{\Lambda}_\mathbf{Z}$ เป็นเมทริกซ์สหสัมพันธ์ที่เกิดปัญหา NORTA defective ซึ่ง

$$\mathbf{\Lambda}_\mathbf{Z} = \begin{bmatrix} 1 & u_1 & u_2 & \cdots & u_{n-(p-1)+1} \\ u_1 & 1 & u_3 & \cdots & u_{n-(p-1)+2} \\ u_2 & u_3 & 1 & \cdots & u_{n-(p-1)+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{n-(p-1)+1} & u_{n-(p-1)+2} & u_{n-(p-1)+3} & \cdots & 1 \end{bmatrix}_{d \times d}$$

โดยให้ $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T$ เป็นเวกเตอร์ NORTA defective (NORTA defective vector) บนเซตของจำนวนจริง $\Re^n$ และกำหนดให้ $\mathbf{\Sigma}'_\mathbf{Z}$ เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ซึ่ง

$$\mathbf{\Sigma}'_\mathbf{Z} = \begin{bmatrix} 1 & v_1 & v_2 & \cdots & v_{n-(p-1)+1} \\ v_1 & 1 & v_3 & \cdots & v_{n-(p-1)+2} \\ v_2 & v_3 & 1 & \cdots & v_{n-(p-1)+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{n-(p-1)+1} & v_{n-(p-1)+2} & v_{n-(p-1)+3} & \cdots & 1 \end{bmatrix}_{d \times d}$$

และให้ $\mathbf{v}=[v_1,v_2,\ldots,v_n]^T$ เป็นเวกเตอร์สหสัมพันธ์ที่เป็นไปได้(feasible correlation vector) ของ $\boldsymbol{\Sigma}'_{\mathbf{Z}}$ บนเซตของจำนวนจริง $\Re^n$ เมื่อ $-1\le u_i, v_i \le 1$

ดังนั้น สำหรับการวิจัยครั้งนี้ ผู้วิจัยได้กำหนดฟังก์ชันระยะทาง $r(\boldsymbol{\Sigma}'_{\mathbf{Z}},\boldsymbol{\Lambda}_{\mathbf{Z}})$ เป็นดังนี้

SDP-L_1 มีฟังก์ชันระยะทางเป็น $r_1(\boldsymbol{\Sigma}'_{\mathbf{Z}},\boldsymbol{\Lambda}_{\mathbf{Z}})=r_1(\mathbf{v},\mathbf{u})=\sum_{i=1}^{n}|v_i-u_i|$, ...(2.4)

SDP-L_2 มีฟังก์ชันระยะทางเป็น $r_2(\boldsymbol{\Sigma}'_{\mathbf{Z}},\boldsymbol{\Lambda}_{\mathbf{Z}})=r_2(\mathbf{v},\mathbf{u})=\sum_{i=1}^{n}(v_i-u_i)^2$ ...(2.5)

และ SDP-L_∞ มีฟังก์ชันระยะทางเป็น $r_\infty(\boldsymbol{\Sigma}'_{\mathbf{Z}},\boldsymbol{\Lambda}_{\mathbf{Z}})=r_\infty(\mathbf{v},\mathbf{u})=\max_i\{|v_i-u_i|\}$ ...(2.6)

ท้ายที่สุด จะกำหนดให้ $\boldsymbol{\Sigma}'_{\mathbf{Z}}$ ที่ทำให้ฟังก์ชันระยะทาง $r(\mathbf{v},\mathbf{u})$ สำหรับแต่ละฟังก์ชันของ SDP-L_1 , SDP-L_2 และ SDP-L_∞ มีค่าน้อยที่สุดเป็น $\boldsymbol{\Sigma}_{\mathbf{Z}}$

สำหรับรูปแบบฟังก์ชัน SDP-L_1 , SDP-L_2 และ SDP-L_∞ เมื่อกำหนดให้กับวิธี SDP ผู้วิจัยได้ยกกรณีศึกษาประกอบการอธิบายสำหรับตัวแปรสุ่ม 3 ตัวแปร $(d=3, n=3)$ คือ

$$\boldsymbol{\Sigma}'_{\mathbf{Z}}=\begin{bmatrix}1 & v_1 & v_2\\ v_1 & 1 & v_3\\ v_2 & v_3 & 1\end{bmatrix}, \mathbf{v}=[v_1,v_2,v_3]^T$$

และ ...(2.7)

$$\boldsymbol{\Lambda}_{\mathbf{Z}}=\begin{bmatrix}1 & u_1 & u_2\\ u_1 & 1 & u_3\\ u_2 & u_3 & 1\end{bmatrix}, \mathbf{u}=[u_1,u_2,u_3]^T$$

ส่วนการเขียนปัญหาให้อยู่ในรูปแบบของ SDP ซึ่งประกอบด้วยฟังก์ชัน L_1 , L_2 และ L_∞ อ้างอิงมาจากงานวิจัยของ ดร. เสกสรร เกียรติสุไพบูลย์ ในปี ค.ศ. 2006 ดังมีรายละเอียดนำเสนอเป็นลำดับต่อไปนี้

2.5.1 วิธี SDP-L_1

จาก (2.3) และ (2.4) สามารถเขียนปัญหาให้อยู่ในรูปแบบทั่วไป SDP ได้เป็น

$$\text{Minimize } \sum_{i=1}^{n} \left(|v_i - u_i| \right)$$

$$\text{subject to } \mathbf{\Sigma}'_{\mathbf{Z}} \succeq \mathbf{0} \qquad \ldots (2.8)$$

พิจารณาปัญหากรณีศึกษา (2.7) เขียนปัญหาด้วย (2.8) ได้ดังนี้

$$\text{Minimize } |v_1 - u_1| + |v_2 - u_2| + |v_3 - u_3|$$

$$\text{subject to } \mathbf{\Sigma}'_{\mathbf{Z}} \succeq \mathbf{0} \qquad \ldots (2.9)$$

จากงานวิจัยของ ดร.เสกสรร เกียรติสุไพบูลย์ ในปี ค.ศ. 2006 สามารถเขียนปัญหาในรูปแบบของ SDP โดยประยุกต์ปัญหาให้อยู่ในรูปแบบมาตรฐานของการกำหนดการเชิงเส้น (linear programming) กำหนดให้ $|v_i - u_i| = \left(x_i^+ + x_i^-\right)$ และ $\left(v_i - u_i\right) = \left(x_i^+ - x_i^-\right)$ ซึ่ง $x_i^+ \geq 0,\ x_i^- \geq 0$ เมื่อจัดปัญหาให้อยู่รูปแบบมาตรฐานของ SDP (2.2) ซึ่ง $\mathbf{x} \in \Re^9$ จะได้

$$\mathbf{x} = \begin{bmatrix} x_1^+ & x_1^- & x_2^+ & x_2^- & x_3^+ & x_3^- & v_1 & v_2 & v_3 \end{bmatrix}^T$$

โดยที่เวกเตอร์วัตถุประสงค์ (objective vector) คือ

$$\mathbf{c} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T$$

และเขียนรูปแบบของ constraint matrices ได้เป็น

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{F}_{i1} & & & & \\ & \mathbf{F}_{i2} & & & \\ & & \mathbf{F}_{i3} & & \\ & & & \mathbf{F}_{i4} & \\ & & & & \mathbf{F}_{i5} \end{bmatrix}$$

ซึ่งในแต่ละเมทริกซ์ $\mathbf{F}_{ij}$ มีขนาด 3×3 สำหรับ $i = 1, \ldots, 9$ และ $j = 1, \ldots, 5$ ในขณะที่บล็อคของ $\mathbf{F}_i$ ที่ไม่อยู่ในแนวเส้นทแยงมุม (off diagonal blocks) เป็นเมทริกซ์ศูนย์ (zero matrix) ทั้งหมด สำหรับ $\mathbf{F}_{ij}$ ทั้งหมดเขียนได้ดังนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$$\mathbf{F}_{01}=\begin{bmatrix} u_1 & & \\ & u_2 & \\ & & u_3 \end{bmatrix},\quad \mathbf{F}_{02}=\begin{bmatrix} -u_1 & & \\ & -u_2 & \\ & & -u_3 \end{bmatrix},\quad \mathbf{F}_{05}=\begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix},$$

$$\mathbf{F}_{11}=\begin{bmatrix} -1 & & \\ & 0 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{12}=\begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{13}=\begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix},$$

$$\mathbf{F}_{21}=\begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{22}=\begin{bmatrix} -1 & & \\ & 0 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{24}=\begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix},$$

$$\mathbf{F}_{31}=\begin{bmatrix} 0 & & \\ & -1 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{32}=\begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{33}=\begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix},$$

$$\mathbf{F}_{41}=\begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{42}=\begin{bmatrix} 0 & & \\ & -1 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{44}=\begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix},$$

$$\mathbf{F}_{51}=\begin{bmatrix} 0 & & \\ & 0 & \\ & & -1 \end{bmatrix},\quad \mathbf{F}_{52}=\begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix},\quad \mathbf{F}_{53}=\begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix},$$

$$\mathbf{F}_{61}=\begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix},\quad \mathbf{F}_{62}=\begin{bmatrix} 0 & & \\ & 0 & \\ & & -1 \end{bmatrix},\quad \mathbf{F}_{64}=\begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix},$$

$$\mathbf{F}_{71}=\begin{bmatrix} -1 & & \\ & 0 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{72}=\begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{75}=\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{F}_{81}=\begin{bmatrix} 0 & & \\ & -1 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{82}=\begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix},\quad \mathbf{F}_{85}=\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

$$\mathbf{F}_{91}=\begin{bmatrix} 0 & & \\ & 0 & \\ & & -1 \end{bmatrix},\quad \mathbf{F}_{92}=\begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix},\quad \mathbf{F}_{95}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

<u>หมายเหตุ</u> สมาชิกที่ไม่ปรากฏในแต่ละ $\mathbf{F}_{ij}$ มีค่าเป็น 0 และให้ $\mathbf{F}_{ij}$ ที่ไม่ได้กล่าวถึงจากข้างต้นเป็นเมทริกซ์ศูนย์ (zero matrix)

2.5.2 วิธี SDP- L_2

จาก (2.3) และ (2.5) สามารถเขียนปัญหาให้อยู่ในรูปแบบทั่วไปของ SDP ได้เป็น

$$\text{Minimize } \sum_{i=1}^{n}\left(\left|v_i - u_i\right|^2\right)$$
$$\text{subject to } \boldsymbol{\Sigma}'_{\mathbf{Z}} \succeq \mathbf{0} \qquad \ldots(2.10)$$

พิจารณาปัญหากรณีศึกษา (2.7) เขียนปัญหาด้วย (2.10) ได้ดังนี้

$$\text{Minimize } \left|v_1 - u_1\right|^2 + \left|v_2 - u_2\right|^2 + \left|v_3 - u_3\right|^2$$
$$\text{subject to } \boldsymbol{\Sigma}'_{\mathbf{Z}} \succeq \mathbf{0} \qquad \ldots(2.11)$$

<u>ข้อสังเกต</u> ฟังก์ชันวัตถุประสงค์ (objective function) ใน (2.11) ไม่เป็นฟังก์ชันเชิงเส้น (non-linear function)

จากงานวิจัยของ ดร. เสกสรร เกียรติสุไพบูลย์ ในปี ค.ศ. 2006 ได้พัฒนารูปแบบสำหรับการแก้ปัญหาด้วยวิธี SDP $-L_2$ ซึ่งในขั้นแรกทำการแปลงฟังก์ชันวัตถุประสงค์ใน (2.11) ให้เป็นฟังก์ชันเชิงเส้น (linear function) กำหนดให้ t เป็นตัวแปรสเกลาร์ (scalar variable) ใด ๆ ซึ่ง $t \in \Re$ และ

$$\mathbf{M} = \begin{bmatrix} \mathbf{I} & (\mathbf{v}-\mathbf{u}) \\ (\mathbf{v}-\mathbf{u})^T & t \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ (\mathbf{v}-\mathbf{u})^T & t \end{bmatrix} \begin{bmatrix} \mathbf{I} & (\mathbf{v}-\mathbf{u}) \\ \mathbf{0} & 1-(\mathbf{v}-\mathbf{u})^T(\mathbf{v}-\mathbf{u})/t \end{bmatrix}$$

โดยที่ $\mathbf{I}$ เป็นเมทริกซ์เอกลักษณ์ (identity matrix) และดีเทอร์มิแนนต์ (determinant) ของ $\mathbf{M}$ คือ

$$\det(\mathbf{M}) = t - (\mathbf{v}-\mathbf{u})^T(\mathbf{v}-\mathbf{u})$$

จากนั้นพิจารณาเมทริกซ์ $\mathbf{M}$ ซึ่งถ้าดีเทอร์มิแนนต์ (determinant) ของ $\mathbf{M}$ มีค่ามากกว่าหรือเท่ากับ 0 $(\det(\mathbf{M}) \geq 0)$ แล้ว $\mathbf{M}$ จะเป็นเมทริกซ์ไม่เป็นลบแน่นอน (non-negative definite matrix)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ดังนั้น สามารถเขียนปัญหาจาก (2.11) ในรูปแบบของ SDP ได้เป็น

Minimize t

subject to $\boldsymbol{\Sigma}'_{\mathbf{Z}} \succeq \mathbf{0}$,

$$\begin{bmatrix} \mathbf{I} & (\mathbf{v}-\mathbf{u}) \\ (\mathbf{v}-\mathbf{u})^T & t \end{bmatrix} \succeq 0 \qquad \dots (2.12)$$

เมื่อจัดปัญหาให้อยู่รูปแบบมาตรฐานของ SDP (2.2) ซึ่ง $\mathbf{x} \in \Re^4$ จะได้

$$\mathbf{x} = [t \quad v_1 \quad v_2 \quad v_3]^T$$

โดยที่เวกเตอร์วัตถุประสงค์ (objective vector) คือ

$$\mathbf{c} = [1 \quad 0 \quad 0 \quad 0]^T$$

และเขียนรูปแบบของ constraint matrices ได้เป็น

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{F}_{i1} & \\ & \mathbf{F}_{i2} \end{bmatrix}$$

โดยที่เมทริกซ์ $\mathbf{F}_{i1}$ มีขนาด 4×4 ในขณะที่เมทริกซ์ $\mathbf{F}_{i2}$ มีขนาด 3×3 สำหรับ $i = 1, \dots, 4$ และ $\mathbf{F}_{ij}$ ทั้งหมดเขียนได้ดังนี้

$$\mathbf{F}_{01} = \begin{bmatrix} 1 & 0 & 0 & -u_1 \\ 0 & 1 & 0 & -u_2 \\ 0 & 0 & 1 & -u_3 \\ u_1 & u_2 & u_3 & 0 \end{bmatrix}, \quad \mathbf{F}_{02} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{F}_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t \end{bmatrix}, \mathbf{F}_{12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{F}_{21} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{F}_{22} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{F}_{31} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \mathbf{F}_{32} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$$\mathbf{F}_{41} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{F}_{42} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

2.5.3 วิธี SDP-L_∞

จาก (2.3) และ (2.6) สามารถเขียนปัญหาให้อยู่ในรูปแบบทั่วไปของ SDP ได้เป็น

$$\begin{aligned} &\text{Minimize } \max\{|v_1 - u_1|, |v_2 - u_2|, \ldots, |v_n - u_n|\} \\ &\qquad \text{subject to } \boldsymbol{\Sigma}'_{\mathbf{Z}} \succeq \mathbf{0} \end{aligned} \qquad \ldots (2.13)$$

พิจารณาปัญหากรณีศึกษา (2.7) เขียนปัญหาด้วย (2.13) ได้ดังนี้

$$\begin{aligned} &\text{Minimize } \max\{|v_1 - u_1|, |v_2 - u_2|, |v_3 - u_3|\} \\ &\qquad \text{subject to } \boldsymbol{\Sigma}'_{\mathbf{Z}} \succeq \mathbf{0} \end{aligned} \qquad \ldots (2.14)$$

เขียนปัญหา (2.14) ใหม่ โดยอาศัยเทคนิคของการเขียนปัญหากำหนดการเชิงเส้น จะได้เป็น

$$\begin{aligned} &\text{Minimize } t \\ &\qquad \text{subject to } \boldsymbol{\Sigma}'_{\mathbf{Z}} \succeq \mathbf{0}, \\ &\qquad\qquad t \ge |v_1 - u_1|, \\ &\qquad\qquad t \ge |v_2 - u_2|, \\ &\qquad\qquad t \ge |v_3 - u_3| \end{aligned} \qquad \ldots (2.15)$$

โดยที่ t เป็นตัวแปรสเกลาร์ (scalar variable) ใด ๆ ซึ่ง $t \in \Re$

เช่นเดียวกันกับการเขียนปัญหาด้วย SDP-L_∞ กำหนดให้ $|v_i - u_i| = (x_i^+ + x_i^-)$ และ $(v_i - u_i) = (x_i^+ - x_i^-)$ ซึ่ง $x_i^+ \ge 0,\ x_i^- \ge 0$ เมื่อจัดปัญหาให้อยู่รูปแบบมาตรฐานของ SDP (2.2) โดยให้ $\mathbf{x} \in \Re^{10}$ จะได้

$$\mathbf{x} = \begin{bmatrix} x_1^+ & x_1^- & x_2^+ & x_2^- & x_3^+ & x_3^- & v_1 & v_2 & v_3 & t \end{bmatrix}^T$$

โดยที่เวกเตอร์วัตถุประสงค์ (objective vector) คือ

$$\mathbf{c} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

และเขียนรูปแบบของ constraint matrices ได้เป็น

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{F}_{i1} & & & & & \\ & \mathbf{F}_{i2} & & & & \\ & & \mathbf{F}_{i3} & & & \\ & & & \mathbf{F}_{i4} & & \\ & & & & \mathbf{F}_{i5} & \\ & & & & & \mathbf{F}_{i6} \end{bmatrix}$$

ซึ่งในแต่ละเมทริกซ์ $\mathbf{F}_{ij}$ มีขนาด 3×3 สำหรับ $i = 1,\ldots,10$ และ $j = 1,\ldots,6$ ในขณะที่บล็อคของ $\mathbf{F}_i$ ที่ไม่อยู่ในแนวเส้นทแยงมุม (off diagonal blocks) เป็นเมทริกซ์ศูนย์ (zero matrix) สำหรับ $\mathbf{F}_{ij}$ ทั้งหมดเขียนได้ดังนี้

$$\mathbf{F}_{01} = \begin{bmatrix} u_1 & & \\ & u_2 & \\ & & u_3 \end{bmatrix}, \quad \mathbf{F}_{02} = \begin{bmatrix} -u_1 & & \\ & -u_2 & \\ & & -u_3 \end{bmatrix}, \quad \mathbf{F}_{05} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix},$$

$$\mathbf{F}_{11} = \begin{bmatrix} -1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{12} = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{13} = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{16} = \begin{bmatrix} -1 & & \\ & 0 & \\ & & 0 \end{bmatrix},$$

$$\mathbf{F}_{21} = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{22} = \begin{bmatrix} -1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{24} = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{26} = \begin{bmatrix} -1 & & \\ & 0 & \\ & & 0 \end{bmatrix},$$

$$\mathbf{F}_{31} = \begin{bmatrix} 0 & & \\ & -1 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{32} = \begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{33} = \begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{36} = \begin{bmatrix} 0 & & \\ & -1 & \\ & & 0 \end{bmatrix},$$

$$\mathbf{F}_{41} = \begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{42} = \begin{bmatrix} 0 & & \\ & -1 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{44} = \begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{46} = \begin{bmatrix} 0 & & \\ & -1 & \\ & & 0 \end{bmatrix},$$

$$\mathbf{F}_{51} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & -1 \end{bmatrix}, \quad \mathbf{F}_{52} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix}, \quad \mathbf{F}_{53} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix}, \quad \mathbf{F}_{56} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & -1 \end{bmatrix},$$

$$\mathbf{F}_{61} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix}, \quad \mathbf{F}_{62} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & -1 \end{bmatrix}, \quad \mathbf{F}_{64} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix}, \quad \mathbf{F}_{66} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & -1 \end{bmatrix},$$

$$\mathbf{F}_{71} = \begin{bmatrix} -1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{72} = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathbf{F}_{75} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$$\mathbf{F}_{81} = \begin{bmatrix} 0 & & \\ & -1 & \\ & & 0 \end{bmatrix}, \qquad \mathbf{F}_{82} = \begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix}, \qquad \mathbf{F}_{85} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

$$\mathbf{F}_{91} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & -1 \end{bmatrix}, \qquad \mathbf{F}_{92} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix}, \qquad \mathbf{F}_{95} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{F}_{10,6} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$$

<u>หมายเหตุ</u> สมาชิกที่ไม่ปรากฏในแต่ละ $\mathbf{F}_{ij}$ มีค่าเป็น 0 และให้ $\mathbf{F}_{ij}$ ที่ไม่ได้กล่าวถึงจากข้างต้นเป็นเมทริกซ์ศูนย์ (zero matrix)

2.6 วิธี Eigenvalue Shifting (ES)

จากข้อ 2.4 ในกระบวนการสร้างเวกเตอร์ด้วยวิธี NORTA ถ้า $\mathbf{\Lambda_Z}$ ที่ได้จากความสัมพันธ์ในสมการ (2.1) ไม่เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (infeasible correlation matrix) จะทำการประมาณ $\mathbf{\Lambda_Z}$ ด้วย $\mathbf{\Sigma'_Z}$ ซึ่งมีขั้นตอนดังต่อไปนี้

1. หาค่าเฉพาะที่ต่ำสุด (lowest eigenvalue) ของ $\mathbf{\Lambda_Z}$ (λ_{lowest}) ซึ่ง $\lambda_{\text{lowest}} \in \Re^{-}$
2. ให้ $\mathbf{\Lambda'_Z} = \mathbf{\Lambda_Z} - \lambda_{\text{lowest}} * \mathbf{I}$ ซึ่ง $\mathbf{I}$ เป็นเมทริกซ์เอกลักษณ์ (identity matrix) ขนาด $d \times d$
3. คำนวณ $\mathbf{\Sigma'_Z} = \left(\dfrac{1}{1-\lambda_{\text{lowest}}}\right)\mathbf{\Lambda'_Z}$
4.

ซึ่ง $\mathbf{\Sigma'_Z}$ ที่ได้จากวิธีการนี้จะเป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) จากนั้นกำหนดให้ $\mathbf{\Sigma'_Z} = \mathbf{\Sigma_Z}$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.7 การหาค่าเฉพาะที่ต่ำสุดด้วยวิธี Power Method

<u>ทฤษฎีบทที่ 2</u> การหาค่าเฉพาะ (eigenvalue) ด้วยวิธี Power Method

กำหนดให้ $\mathbf{M}$ เป็นเมทริกซ์จัตุรัส (square matrix) ขนาด $n \times n$,ให้เวกเตอร์ $\mathbf{x_0} = (1,1,\ldots,1)_{1\times n}{}^T$ และกำหนดให้ $a_i\mathbf{x_i} = \mathbf{Mx_{i-1}}$ โดยที่องค์ประกอบของ $\mathbf{x_i}$ มีค่าสูงสุดเท่ากับ 1 ถ้า $\mathbf{x_i}$ ลู่เข้าสู่ $\mathbf{x}$ ($\mathbf{x_i}$ converge to $\mathbf{x}$) แล้ว โดยทั่วไป a_i จะลู่เข้าสู่ a ด้วย (a_i converge to a) ซึ่ง a ที่ได้คือค่าเฉพาะที่ใหญ่ที่สุด (largest eigenvalue) ของ $\mathbf{M}$

เนื่องจากวิธี Eigenvalue Shifting (ES) จำเป็นต้องใช้ค่าเฉพาะที่ต่ำสุด (lowest eigenvalue) λ_{lowest} ของเมทริกซ์ $\mathbf{\Lambda_Z}$ ซึ่ง λ_{lowest} มีค่าน้อยกว่าศูนย์ สำหรับงานวิจัยนี้ผู้วิจัยเลือกใช้วิธี Power Method เพื่อทำการประมาณค่าดังกล่าว โดยมีขั้นตอนดังแสดงด้วยตัวอย่างการหาค่าเฉพาะที่ต่ำสุดสำหรับเมทริกซ์ $\mathbf{\Lambda_Z}$ ขนาด 3×3 $(d = 3)$ ตามลำดับต่อไปนี้

1) สมมติให้

$$\mathbf{\Lambda_Z} = \begin{bmatrix} 1 & 0.9 & 0.6 \\ 0.9 & 1 & 0.1 \\ 0.6 & 0.1 & 1 \end{bmatrix}$$

2) นำค่ารอยเมทริกซ์ $\mathbf{\Lambda_Z}$ ($\text{trace}(\mathbf{\Lambda_Z})$) ซึ่งในที่นี้ $\text{trace}(\mathbf{\Lambda_Z}) = 3$ มาบวกเข้ากับทุกสมาชิกในแนวเส้นทแยงมุมของเมทริกซ์ $\mathbf{\Lambda_Z}$ จะได้

$$\mathbf{\Lambda_Z^*} = \begin{bmatrix} 1+3 & 0.9 & 0.6 \\ 0.9 & 1+3 & 0.1 \\ 0.6 & 0.1 & 1+3 \end{bmatrix} = \begin{bmatrix} 4 & 0.9 & 0.6 \\ 0.9 & 4 & 0.1 \\ 0.6 & 0.1 & 4 \end{bmatrix}$$

3) คำนวณหาเมทริกซ์ผกผันของ $\mathbf{\Lambda_Z^*}$ นั่นคือ

$$\left(\mathbf{\Lambda_Z^*}\right)^{-1} = \begin{bmatrix} 0.2692 & -0.0596 & -0.0389 \\ -0.0596 & 0.2634 & 0.0024 \\ -0.0389 & 0.0024 & 0.2258 \end{bmatrix} = \mathbf{B}$$

4) คำนวณหาค่าเฉพาะที่สูงสุด (highest eigenvalue) λ_{highest} ของ $\mathbf{B}$ ในรูปแบบ (form) ของความสัมพันธ์ระหว่างเมทริกซ์ $\mathbf{B}$ กับค่าเฉพาะที่สูงสุด λ_{highest} ดังนี้

$$\mathbf{Bx} = \lambda_{\text{highest}}\mathbf{x} \qquad \ldots (2.16)$$

โดยที่ $\mathbf{x} = (x_1, ..., x_d)^T$

ซึ่งวิธีการคำนวณหาค่า λ_{highest} มีขั้นตอนดังต่อไปนี้

ขั้นที่ 1 เขียนระบบของสมการจาก (2.16) จะได้

$$\begin{aligned}
2.2692x_1 \quad -0.0596x_2 \quad -0.0389x_3 &= \lambda_{\text{highest}}x_1 \\
-0.0596x_1 \quad +0.2634x_2 \quad +0.0024x_3 &= \lambda_{\text{highest}}x_2 \\
-0.0389x_1 \quad +0.0024x_2 \quad +0.2258x_3 &= \lambda_{\text{highest}}x_3
\end{aligned}$$

กำหนดให้ $\mathbf{x} = (x_1, x_2, x_3)^T = (1,1,1)^T$

$$\begin{aligned}
2.2692(1) \quad -0.0596(1) \quad -0.0389(1) &= 0.1707 \\
-0.0596(1) \quad +0.2634(1) \quad +0.0024(1) &= 0.2061 \\
-0.0389(1) \quad +0.0024(1) \quad +0.2258(1) &= 0.2192
\end{aligned}$$

จากนั้นเลือกค่ามากที่สุดทางด้านขวาของระบบสมการ (right-hand side) ซึ่งในขั้นนี้คือ 0.2192 แล้วทำให้ค่านี้มีค่าเท่ากับ 1 ดังแสดงต่อไปนี้

$$\begin{bmatrix} 2.2693 & -0.0596 & -0.0389 \\ -0.0596 & 2.2634 & 0.0024 \\ -0.0389 & 0.0024 & 0.2258 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.1707 \\ 0.2061 \\ 0.2192 \end{bmatrix} = 0.2192\begin{bmatrix} 0.7788 \\ 0.9401 \\ 1 \end{bmatrix}$$

ดังนั้นในขั้นที่ 1 ค่าประมาณของ λ_{highest} คือ ${}^{(1)}\lambda_{\text{highest}}$ มีค่าเท่ากับ 0.2192 และให้ $\mathbf{x} = (0.7788, 0.9401, 1)^T$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นที่ 2 คูณ $\mathbf{B}$ ด้วย $\mathbf{x} = (0.7788, 0.9401, 1)^T$ จาก (2.16) จะได้

$$\begin{bmatrix} 2.2693 & -0.0596 & -0.0389 \\ -0.0596 & 2.2634 & 0.0024 \\ -0.0389 & 0.0024 & 0.2258 \end{bmatrix} \begin{bmatrix} 0.7788 \\ 0.9401 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.1147 \\ 0.2035 \\ 0.2277 \end{bmatrix} = 0.2277 \begin{bmatrix} 0.5040 \\ 0.8938 \\ 1 \end{bmatrix}$$

โดยในขั้นที่ 2 ค่าประมาณของ λ_{highest} คือ ${}^{(2)}\lambda_{\text{highest}}$ มีค่าเท่ากับ 0.2277 และให้ $\mathbf{x} = (0.5040, 0.8938, 1)^T$ และกำหนดให้เปอร์เซ็นต์ค่าคลาดเคลื่อนของการประมาณค่า λ_{highest} (error estimate) คือ $|\varepsilon_0|$ มีค่าเป็นดังนี้

$$|\varepsilon_0| = \left| \frac{{}^{(1)}\lambda_{\text{highest}} - {}^{(2)}\lambda_{\text{highest}}}{{}^{(1)}\lambda_{\text{highest}}} \right| \times 100 = \left| \frac{0.2192 - 0.2277}{0.2192} \right| \times 100 = 3.86\%$$

<u>หมายเหตุ</u> สำหรับการวิจัยนี้ ผู้วิจัยได้กำหนดให้ $|\varepsilon_0| = (1 \times 10^{-16})\%$ เพื่อทำการคำนวณหาค่า λ_{highest}

ขั้นที่ 3 คูณ $\mathbf{B}$ ด้วย $\mathbf{x} = (0.5040, 0.8938, 1)^T$ จาก (2.16) จะได้

$$\begin{bmatrix} 2.2693 & -0.0596 & -0.0389 \\ -0.0596 & 2.2634 & 0.0024 \\ -0.0389 & 0.0024 & 0.2258 \end{bmatrix} \begin{bmatrix} 0.5040 \\ 0.8938 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.0435 \\ 0.2077 \\ 0.2383 \end{bmatrix} = 0.2383 \begin{bmatrix} 0.1827 \\ 0.8716 \\ 1 \end{bmatrix}$$

โดยในขั้นที่ 3 ค่าประมาณของ λ_{highest} คือ ${}^{(3)}\lambda_{\text{highest}}$ มีค่าเท่ากับ 0.2277 และให้ $\mathbf{x} = (0.1827, 0.8716, 1)^T$ และกำหนดให้เปอร์เซ็นต์ค่าคลาดเคลื่อนของการประมาณค่า λ_{highest} (error estimate) คือ $|\varepsilon_0|$ มีค่าเป็นดังนี้

$$|\varepsilon_0| = \left| \frac{{}^{(2)}\lambda_{\text{highest}} - {}^{(3)}\lambda_{\text{highest}}}{{}^{(2)}\lambda_{\text{highest}}} \right| = \left| \frac{0.2277 - 0.2383}{0.2277} \right| \times 100 = 4.6467\%$$

.

.

.

ทำเช่นต่อไปเรื่อย ๆ (ขั้นที่ 3,4 ,5,...) ต่อไปจนกระทั่งได้ $|\varepsilon_0| \le (1\times10^{-16})\%$ ซึ่งในตัวอย่างนี้จะหยุดกระทำในขั้นที่ 96 ดังนี้

ขั้นที่ 107 คูณ $\mathbf{B}$ ด้วย $\mathbf{x} = (1,-0.8195,-0.4994)^T$ จาก (2.16) จะได้

$$\begin{bmatrix} 2.2693 & -0.0596 & -0.0389 \\ -0.0596 & 2.2634 & 0.0024 \\ -0.0389 & 0.0024 & 0.2258 \end{bmatrix} \begin{bmatrix} 1 \\ -0.8195 \\ -0.4994 \end{bmatrix} = \begin{bmatrix} 0.3375 \\ -0.2766 \\ -0.1686 \end{bmatrix} = 0.3375 \begin{bmatrix} 1 \\ -0.8195 \\ -0.4994 \end{bmatrix}$$

โดยในขั้นที่ 106 ค่าประมาณของ λ_{highest} คือ ${}^{(107)}\lambda_{\text{highest}}$ มีค่าเท่ากับ 0.3375 และให้ $\mathbf{x} = (1,-0.8195,-0.4994)^T$ และกำหนดให้เปอร์เซ็นต์ค่าคลาดเคลื่อนของการประมาณค่า λ_{highest} (error estimate) คือ $|\varepsilon_0|$ มีค่าเป็นดังนี้

$$|\varepsilon_0| = \left| \frac{{}^{(89)}\lambda_{\text{highest}} - {}^{(90)}\lambda_{\text{highest}}}{{}^{(89)}\lambda_{\text{highest}}} \right| = \left| \frac{0.3375 - 0.3375}{0.3375} \right| \times 100 < (1\times10^{-16})\%$$

เพราะฉะนั้น จะได้ว่า λ_{highest} ของ $\mathbf{B}$ หรือ $(\mathbf{\Lambda}_{\mathbf{Z}}^*)^{-1}$ มีค่าโดยประมาณเป็น 0.3375

5) คำนวณหาค่าเฉพาะที่ต่ำสุด (lowest eigenvalue) λ_{lowest} ของ $\mathbf{\Lambda}_{\mathbf{Z}}$ ได้จาก λ_{highest} ของ $\mathbf{B}$ หรือ $(\mathbf{\Lambda}_{\mathbf{Z}}^*)^{-1}$ ดังนี้

$$\begin{aligned} \lambda_{\text{lowest}} &= \frac{1}{\lambda_{\text{highest}}} - \text{trace}(\mathbf{\Lambda}_{\mathbf{Z}}) \\ &= \frac{1}{0.3375} - 3 \\ &\approx -0.037245 \end{aligned}$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.8 การตรวจสอบเมทริกซ์กึ่งบวกแน่นอนด้วยวิธี Superdiagonalization

สำหรับงานวิจัยนี้ ในส่วนของการพัฒนาโปรแกรมสร้างเวกเตอร์สุ่มด้วยวิธี NORTA โดยใช้ภาษา C/C++ มีความจำเป็นที่ต้องทำการตรวจสอบเมทริกซ์ $\mathbf{\Lambda_Z}$ ในกระบวนการ NORTA ว่าเป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) หรือไม่ นั่นคือการตรวจสอบว่า $\mathbf{\Lambda_Z}$ เป็นเมทริกซ์กึ่งบวกแน่นอน (positive semidefinite matrix) หรือไม่ ซึ่งถ้าหากพบว่า $\mathbf{\Lambda_Z}$ ไม่เป็นเมทริกซ์กึ่งบวกแน่นอนก็จะทำการแก้ปัญหาด้วยวิธีการดังกล่าวไปแล้วข้างต้น (วิธี SDP และวิธี ES) สำหรับการตรวจสอบนี้ ผู้วิจัยได้เลือกใช้วิธี Superdiagonalization โดยมีรายละเอียดดังนี้

ให้ $\mathbf{M}$ เป็นเมทริกซ์จัตุรัส (square matrix) ขนาด $n \times n$ ซึ่ง $\mathbf{M}$ เป็นเมทริกซ์สมมาตร (symmetric matrix) หรือไม่เป็นเมทริกซ์สมมาตรก็ได้ โดยที่ $\mathbf{M}$ จะเป็นเมทริกซ์บวกแน่นอน (positive definite matrix) หรือเป็นเมทริกซ์กึ่งบวกแน่นอน (positive semidefinite matrix) ถ้า $\mathbf{D} = \mathbf{M} + \mathbf{M}^T$ เป็นเมทริกซ์บวกแน่นอน หรือเมทริกซ์กึ่งบวกแน่นอน ตามลำดับ

<u>บทแทรกที่ 1</u>

ถ้า $\mathbf{D}$ เป็นเมทริกซ์บวกแน่นอนแล้ว ทุกสมาชิกที่อยู่ในแนวทแยงมุมหลักของ $\mathbf{D}$ นั่นคือ $d_{ii}, i = 1,\ldots,n$ จะต้องมีค่ามากกว่า 0 หรือถ้า $\mathbf{D}$ เป็นไม่เป็นเมทริกซ์บวกแน่นอน แต่เป็นเมทริกซ์กึ่งบวกแน่นอน เมื่อ $d_{ii}, i = 1,\ldots,n$ มีค่ามากกว่าหรือเท่ากับ 0

<u>บทแทรกที่ 2</u>

ถ้าเมทริกซ์สมมาตร $\mathbf{D}$ มีสมาชิกเป็น d_{ij} ซึ่ง $i, j = 1,\ldots,n$ โดยที่ $\mathbf{D}$ เป็นเมทริกซ์กึ่งบวกแน่นอนและ $d_{ii} = 0$ แล้วสมาชิกทุกตัวที่อยู่ในแถว (row) เดียวกันกับ d_{ii} และหลัก (column) เดียวกันกับ d_{ii} จะต้องเป็น 0 ทั้งหมด

<u>ทฤษฎีบทที่ 3</u>

กำหนดให้ $\mathbf{D}$ เป็นเมทริกซ์สมมาตร (symmetric matrix) โดยที่

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}$$

ซึ่งมีสมาชิกตัวแรกในแนวเส้นทแยงมุมหลัก (first diagonal entry) ไม่เป็น 0 ($d_{11} \neq 0$) ทำการแปลง(transformation) เมทริกซ์ $\mathbf{D}$ ไปเป็น $\mathbf{D}_1$ ดังนี้

$$\mathbf{D}_1 = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} - \left(\frac{d_{21}}{d_{11}}\right)(d_{11}) & d_{22} - \left(\frac{d_{21}}{d_{11}}\right)(d_{12}) & \cdots & d_{2n} - \left(\frac{d_{21}}{d_{11}}\right)(d_{1n}) \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} - \left(\frac{d_{n1}}{d_{11}}\right)(d_{11}) & d_{n2} - \left(\frac{d_{n1}}{d_{11}}\right)(d_{12}) & \cdots & d_{nn} - \left(\frac{d_{n1}}{d_{11}}\right)(d_{1n}) \end{bmatrix}$$

$$= \begin{bmatrix} d_{11}^{(1)} & d_{12}^{(1)} & \cdots & d_{1n}^{(1)} \\ 0 & d_{22}^{(1)} & \cdots & d_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & d_{n1}^{(1)} & \cdots & d_{nn}^{(1)} \end{bmatrix}$$

ให้ $\mathbf{E_1}$ เป็นเมทริกซ์สมมาตรขนาด $(n-1)\times(n-1)$ ซึ่งประกอบด้วยสมาชิกของ $\mathbf{D}_1$ ที่ตัดแถวที่ 1 และหลักที่ 1 ออก นั่นคือ

$$\mathbf{E_1} = \begin{bmatrix} d_{22}^{(1)} & \cdots & d_{2n}^{(1)} \\ \vdots & \ddots & \vdots \\ d_{n1}^{(1)} & \cdots & d_{nn}^{(1)} \end{bmatrix}$$

สามารถสรุปได้ว่า $\mathbf{D}$ เป็นเมทริกซ์บวกแน่นอน (เมทริกซ์กึ่งบวกแน่นอน) ถ้า $d_{11} > 0$ และ $\mathbf{E_1}$ เป็นเมทริกซ์บวกแน่นอน (เมทริกซ์กึ่งบวกแน่นอน)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

วิธีดำเนินการวิจัย

การศึกษาวิจัยครั้งนี้ ผู้วิจัยได้แบ่งวิธีการดำเนินการวิจัยออกเป็น 2 ส่วนสำคัญ คือการสร้างเครื่องมือที่ใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA และการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective จากสองวิธีการคือ วิธี Semidefinite Programming (SDP) และวิธี Eigenvalue Shifting (ES) ซึ่งมีรายละเอียดดังนำเสนอตามลำดับดังนี้

3.1 การสร้างเครื่องมือที่ใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA

สำหรับการวิจัยในส่วนนี้ มีวัตถุประสงค์เพื่อพัฒนาโปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ให้สามารถใช้งานได้จริงบนโปรแกรมไมโครซอฟต์ เอ็กเซล (Microsoft Excel) ซึ่งผู้วิจัยได้นำเสนอขั้นตอนการวิจัยตามลำดับดังนี้

3.1.1 เขียนโปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA โดยผู้วิจัยเลือกใช้ภาษา C/C++ ซึ่งถือภาษาที่สนับสนุนการคำนวณฟังก์ชันทางคณิตศาสตร์ที่มีความซับซ้อนได้อย่างมีประสิทธิภาพและเป็นภาษาหลักสำหรับใช้พัฒนาซอฟต์แวร์ที่สำคัญ ๆ ทั้งในอดีตและปัจจุบัน สำหรับขั้นตอนในการเขียนโปรแกรมมีดังต่อไปนี้

กำหนดให้ $\mathbf{\Sigma_X}$ เป็นเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X} = (X_1, X_2, \ldots, X_d)^T$ ซึ่งมีการแจกแจงขอบเป็นแบบสม่ำเสมอ (uniform marginal) ตามที่ต้องการ และให้ $\mathbf{\Sigma_Z}$ เป็นเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_d)^T$ ซึ่งประกอบตัวแปรสุ่มที่มีการแจกแจงแบบปกติมาตรฐานหลายตัวแปร และเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{Z}$ โดยประมาณ คือ $\mathbf{\Lambda_Z}$ ซึ่งมีองค์ประกอบที่ i, j เป็น $\Lambda_Z(i,j) = 2\sin\left(\frac{\pi}{6}\Sigma_X(i,j)\right)$

จากนั้นทำการตรวจสอบว่า $\mathbf{\Lambda_Z}$ เป็นเมทริกซ์เมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) หรือไม่ โดยใช้วิธี Superdiagonalization ซึ่งมีรายละเอียดอยู่ในบทที่ 2 หัวข้อที่ 2.8

ถ้า $\mathbf{\Lambda_Z}$ เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้แล้วจะกำหนดให้ $\mathbf{\Lambda_Z} = \mathbf{\Sigma_Z}$ และถ้าพบว่า $\mathbf{\Lambda_Z}$ ไม่เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ จะเลือกใช้วิธี ES เพื่อหาเมทริกซ์ $\mathbf{\Sigma'_Z}$ ซึ่งเป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ที่ใกล้เคียงกับ $\mathbf{\Lambda_Z}$ มากที่สุด (สำหรับ

งานวิจัยนี้ได้นิยามความใกล้เคียงเป็นฟังก์ชันระยะทาง (distance function)) จากนั้นกำหนดให้ $\mathbf{\Sigma}'_{\mathbf{Z}} = \mathbf{\Sigma}_{\mathbf{Z}}$ แล้วทำการสร้างเวกเตอร์สุ่ม $\mathbf{X}$ โดยมีขั้นตอนดังต่อไปนี้

(1) ให้ $\mathbf{C}$ เป็นเมทริกซ์สามเหลี่ยมล่าง (lower-triangular matrix) และเป็นเมทริกซ์ไม่เอกฐาน (nonsingular matrix) ซึ่ง $\mathbf{CC}^T = \mathbf{\Sigma}_{\mathbf{Z}}$

(2) สร้างเวกเตอร์ $\mathbf{W} = (W_1, W_2, \ldots, W_d)^T$ ขนาด $d \times 1$ ซึ่งประกอบด้วยตัวแปรสุ่มที่มีการแจกแจงแบบปกติมาตรฐานเหมือนกันและเป็นอิสระต่อกัน

(3) ให้ $\mathbf{Z} = \mathbf{CW}$

(4) สร้าง

$$\mathbf{X} = \begin{bmatrix} F_{X_1}^{-1}[\mathbf{\Phi}(Z_1)] \\ F_{X_2}^{-1}[\mathbf{\Phi}(Z_2)] \\ \vdots \\ F_{X_d}^{-1}[\mathbf{\Phi}(Z_d)] \end{bmatrix}$$

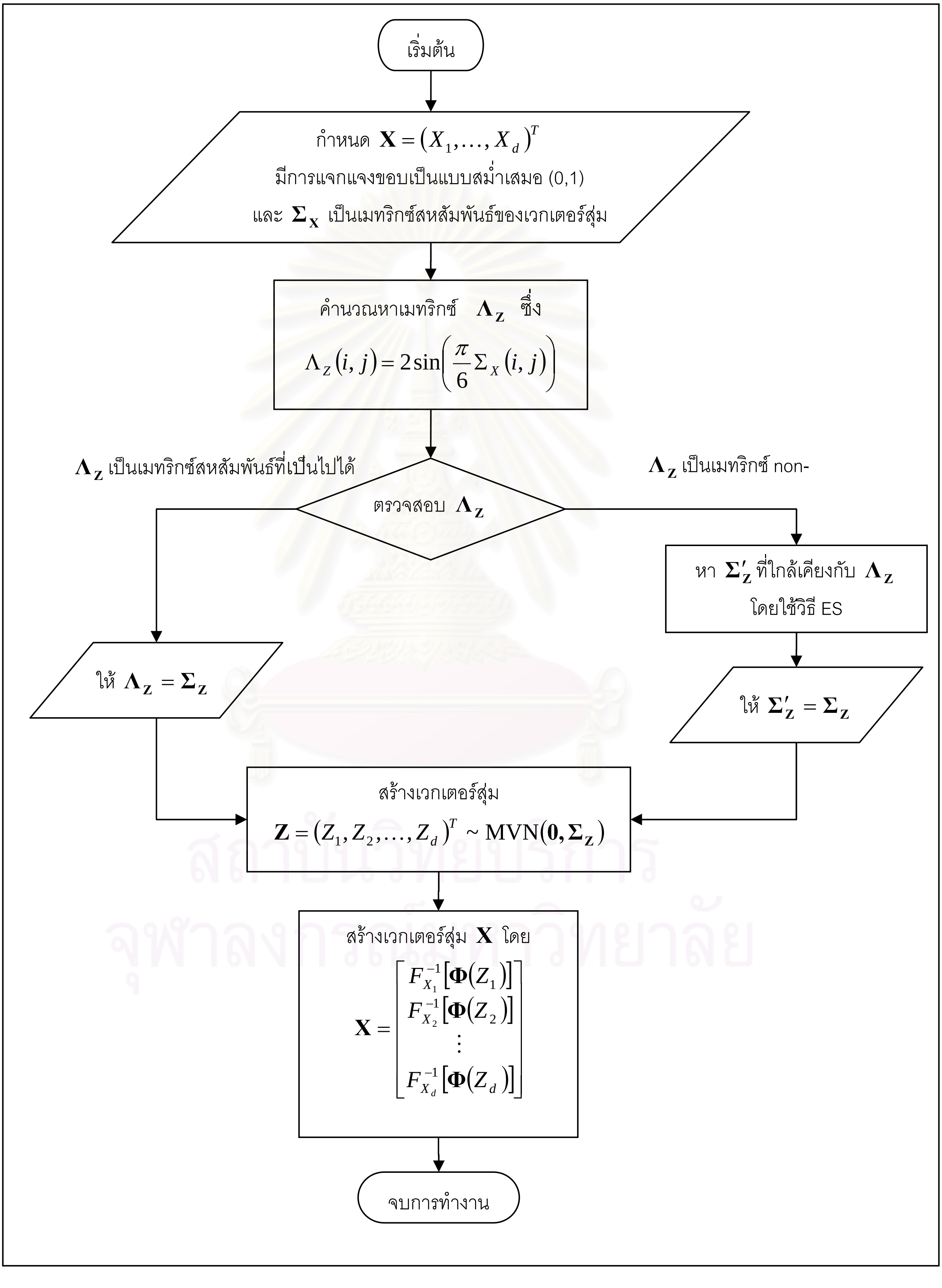
โดยที่ $i = 1, \ldots, d$

(5) กลับไปขั้นตอนที่ 2

ในขั้นตอนที่ 1-3 เป็นวิธีมาตรฐานสำหรับการสร้างเวกเตอร์สุ่มแบบปกติหลายตัวแปร (multivariate normal (MVN) random vector) ซึ่งมีรายละเอียดอยู่ในบทที่ 2 หัวข้อที่ 2.1

จากขั้นตอน 3.1.1 สามารถเขียนผังงาน (flow chart) ของการสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ได้ดังแสดงในแผนภูมิที่ 3.1

<u>แผนภูมิที่ 3.1</u> แสดงผังงานการเขียนโปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA



สำหรับการสร้างตัวเลขสุ่ม (random number) ที่มีการแจกแจงแบบสม่ำเสมอ (uniform distribution) ในช่วง [0,1] ที่ใช้ในการสร้างโปรแกรม NORTA ผู้วิจัยได้อ้างอิงจากงานวิจัยของ Pierre L'Ecuyer และ Richard Simard ในปี ค.ศ. 2000 ส่วนการประมาณค่าสำหรับการแจกแจงสะสม (cumulative distribution) ของตัวแปรสุ่มแบบปกติมาตรฐาน (standard normal random variable) ผู้วิจัยได้อ้างอิงจากงานวิจัยของ George Marsaglia ในปี ค.ศ. 2004

โดยในขั้นนี้จะได้ไฟล์ NORTA.cpp ซึ่งคอมไพล์(compile)ด้วยโปรแกรม Visual C++ 6.0

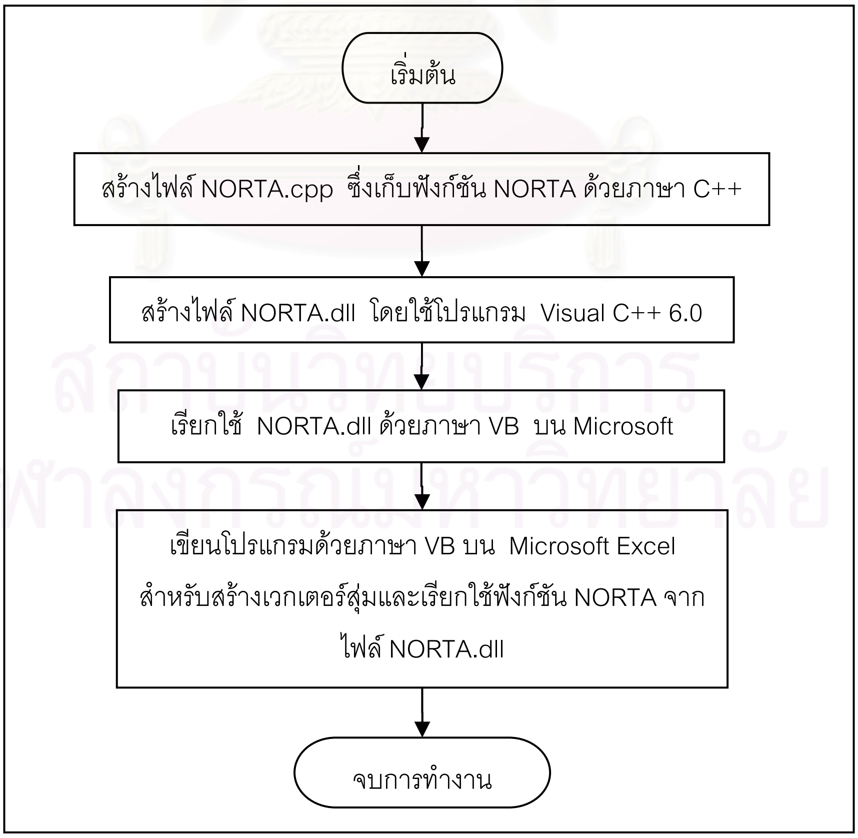
3.1.2 สร้าง (building) ไฟล์ที่มีนามสกุลเป็น dll (NORTA.dll) ซึ่งบรรจุฟังก์ชัน NORTA จากไฟล์ NORTA.cpp ด้วยโปรแกรม Visual C++ 6.0

3.1.3 เรียกใช้ไฟล์ NORTA.dll โดยเขียนคำสั่งด้วยภาษาวิชวล เบสิก (Visual Basic) จากโปรแกรมไมโครซอฟต์ เอ็กเซล (Visual Basic for application on Microsoft Excel)

3.1.4 เรียกใช้ฟังก์ชัน NORTA ผ่านทางโปรแกรม Microsoft Excel โดยใช้ Visual Basic Editor (VBE) ในโปรแกรม Microsoft Excel

จากขั้นตอน 3.1.1 - 3.1.4 สามารถเขียนผังงาน (flow chart) ได้ดังในแผนภูมิที่ 3.2

แผนภูมิที่ 3.2 แสดงผังงานการสร้างโปรแกรม NORTA สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA



3.2 การศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective

การวิจัยในส่วนนี้ มีวัตถุประสงค์เพื่อศึกษาความแตกต่างของวิธีการที่ใช้แก้ปัญหา NORTA defective ซึ่งอาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ประกอบด้วย 2 วิธีการหลัก คือ

1. วิธี Semidefinite Programming (SDP)

การศึกษาในวิธีการนี้ ได้กำหนดฟังก์ชันระยะทาง (distance function) สำหรับวิธี SDP 3 ฟังก์ชัน ทำให้แบ่งการศึกษาวิธี SDP ออกเป็น 3 วิธีย่อย ดังนี้

1.1 วิธี SDP- L_1

1.2 วิธี SDP- L_2

1.3 วิธี SDP- L_∞

2. วิธี Eigenvalue Shifting (ES)

โดยรายละเอียดของแต่ละวิธีได้นำเสนอไว้บทที่ 2

3.2.1 แผนการศึกษาวิจัย

ผู้วิจัยได้กำหนดสถานการณ์ต่าง ๆ สำหรับการวิจัยครั้งนี้ไว้ดังนี้

1) สร้างเมทริกซ์สหสัมพันธ์โดยที่เป็น NORTA defective

2) กำหนดจำนวนตัวแปรสุ่ม d สำหรับใช้ทำการศึกษาปัญหา NORTA defective เป็น $d = 3,4,\ldots,15$ ตัวแปร ตามลำดับ

3) กำหนดขนาดตัวอย่างเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาเป็น 5,000 เมทริกซ์ สำหรับทุกจำนวนตัวแปรสุ่ม $d = 3,4,\ldots,15$

4) เลือกใช้วิธีการเพื่อศึกษาการแก้ปัญหาเมทริกซ์ NORTA defective 4 วิธีการ คือ วิธี SDP- L_1 , วิธี SDP- L_2 , วิธี SDP- L_∞ และวิธี ES

5) กำหนดมาตรวัดระยะทาง (distance measure) เป็น L_1- norm , L_2- norm และ L_∞- norm ซึ่งมีรายละเอียดดังนี้

$$L_1\text{- norm} : \|\mathbf{v}-\mathbf{u}\|_1 = \sum_{i=1}^{n} |v_i - u_i| ,$$

$$L_2\text{- norm} : \|\mathbf{v}-\mathbf{u}\|_2 = \sqrt{\sum_{i=1}^{n} (v_i - u_i)^2}$$

และ

$$L_\infty\text{- norm} : \|\mathbf{v}-\mathbf{u}\|_\infty = \max_i \{|v_i - u_i|\}$$

โดยที่ $\mathbf{v}$ คือเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ของเมทริกซ์ $\mathbf{\Sigma}'_\mathbf{Z}$, $\mathbf{u}$ คือเวกเตอร์ NORTA defective ของเมทริกซ์ $\mathbf{\Lambda}_\mathbf{Z}$ และ $n = \dfrac{d(d-1)}{2}$

3.2.2 ขั้นตอนในการศึกษาวิจัย

ขั้นตอนในการดำเนินการศึกษาวิจัยมีดังนี้

1) สร้างเมทริกซ์สหสัมพันธ์ที่เป็นเมทริกซ์ NORTA defective ($\mathbf{\Lambda_Z}$) สำหรับทุก d ตัวแปรโดยสุ่ม โดยกำหนดให้ $d = 3,4,\ldots,15$

2) แก้ปัญหาเมทริกซ์ NORTA defective ด้วย

2.1) วิธี SDP- L_1 ,ให้เมทริกซ์สหสัมพันธ์ที่ได้จากวิธีนี้เป็น $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$

2.2) วิธี SDP- L_2 ,ให้เมทริกซ์สหสัมพันธ์ที่ได้จากวิธีนี้เป็น $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$

2.3) วิธี SDP- L_∞ ,ให้เมทริกซ์สหสัมพันธ์ที่ได้จากวิธีนี้เป็น $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$

2.4) วิธี ES ,ให้เมทริกซ์สหสัมพันธ์ที่ได้จากวิธีนี้เป็น $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{ES})}$

โดยที่เมทริกซ์ $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$, $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$, $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{ES})}$ เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้

<u>หมายเหตุ</u> สำหรับการแก้ปัญหาเมทริกซ์ NORTA defective ด้วยวิธี SDP- L_1, SDP- L_2 และวิธี SDP- L_∞ ผู้วิจัยได้โปรแกรม SDPA(SemiDefinite Programming Algorithm) เวอร์ชัน 6.20 ซึ่งพัฒนาโดย M. Kojima K. Fujisawa, K. Nakata และ M. Yamashita ในปี ค.ศ. 2005 ซึ่งเป็นโปรแกรมที่ดาวน์โหลดได้ฟรี (free download) ทางอินเตอร์เน็ท

3) คำนวณระยะทางระหว่างเมทริกซ์ NORTA defective ($\mathbf{\Lambda_Z}$) กับเมทริกซ์สหสัมพันธ์ที่ได้จากวิธี SDP- L_1 ,SDP- L_2 ,SDP- L_∞ และวิธี ES โดยกำหนดมาตรวัดเป็น L_1- norm, L_2- norm และ L_∞- norm ตามลำดับ หลังจากนั้นทำการศึกษาเปรียบเทียบระยะทางโดยเฉลี่ย (average distance) ของแต่ละมาตรวัด โดยสรุปผลในรูปของตารางและรูปภาพ

<u>หมายเหตุ</u> สำหรับขั้นตอนที่ 1-3 ผู้วิจัยกำหนดตัวอย่างเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาจำนวน 5,000 เมทริกซ์ สำหรับทุก d ตัวแปร โดยที่ $d = 3,4,\ldots,15$

4) ทำการศึกษาเมทริกซ์สหสัมพันธ์ที่ได้จากวิธี SDP โดยเปรียบเทียบระยะทาง (distance) ระหว่าง $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$, $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$, $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_\mathbf{Z}^{(\mathrm{ES})}$ กับเมทริกซ์ $\mathbf{\Lambda_Z}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞-norm ตามลำดับ (กำหนดตัวอย่างเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาในขั้นตอนนี้คือ 50,000 เมทริกซ์ สำหรับทุก d ตัวแปร)

จากนั้นทำการแปลง (transformation) เมทริกซ์ $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}, \mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{ES})}$ ด้วยสมการ (2.1) ดังมีรายละเอียดต่อไปนี้

4.1) แปลง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$ ด้วย (2.1) จะได้เมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$

4.2) แปลง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$ ด้วย (2.1) จะได้เมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$

4.3) แปลง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ ด้วย (2.1) จะได้เมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$

4.4) แปลง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{ES})}$ ด้วย (2.1) จะได้เมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{ES})}$

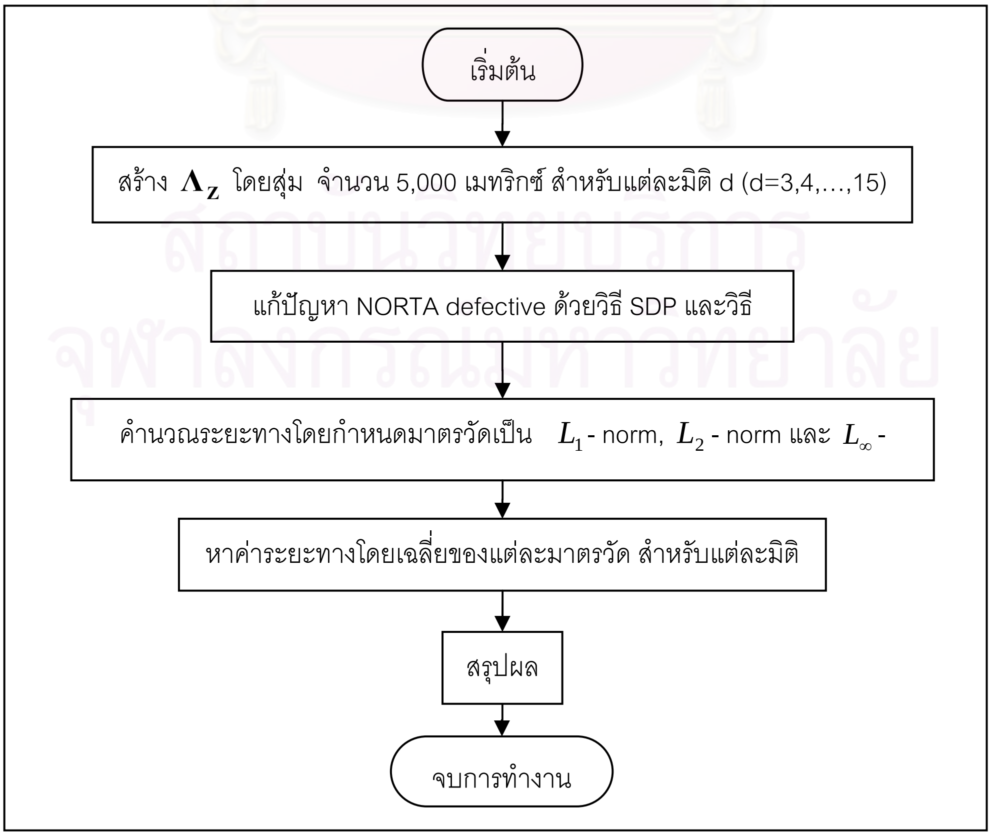
โดยที่เมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{ES})}$ เป็นเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ แล้วทำการศึกษาระยะทางระหว่าง $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma}_{\mathbf{X}}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞- norm ตามลำดับ นำเสนอผลการศึกษาในรูปของตารางโดยยกตัวอย่างประกอบ

5) ทำการศึกษาเปรียบเทียบประสิทธิภาพ (ความเร็ว) ของวิธีการแก้ปัญหา NORTA defective จากทั้ง 4 วิธีการคือ วิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES

6) ทำการจำลองเวกเตอร์สุ่มหลายตัวแปรจากโปรแกรม NORTA ที่พัฒนาขึ้นแล้วนำเสนอผลในรูปของตาราง

ผังงานแสดงขั้นตอนในการดำเนินการศึกษาวิจัย ดังแสดงในแผนภูมิที่ 3.3

แผนภูมิที่ 3.3 แสดงผังงานสำหรับศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สำหรับในแต่ละขั้นตอนมีรายละเอียดดังนำเสนอเป็นลำดับต่อไปนี้

ขั้นตอนที่ 1

สร้างเมทริกซ์สหสัมพันธ์ที่เป็นเมทริกซ์ NORTA defective ($\mathbf{\Lambda_Z}$) จำนวน 5,000 เมทริกซ์ สำหรับตัวแปรสุ่ม d ตัวแปรโดยสุ่ม ซึ่งกำหนดให้ $d = 3,4,\ldots,15$ โดยมีรายละเอียดดังนี้

1. สร้างเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) $\mathbf{\Sigma}_d$ สำหรับตัวแปรสุ่ม d ตัวแปร โดยใช้วิธี Onion method จากงานวิจัยของ Ghosh ในปี ค.ศ. 2004

สำหรับวิธี Onion method มีขั้นตอนดังนี้

1) ให้ $\mathbf{\Sigma}_1$ เป็นเมทริกซ์ขนาด 1×1

2) สำหรับ $k = 2,\ldots,d$

2.1) ให้ $\mathbf{q}$ เป็นคอลัมน์เวกเตอร์ (column vector) ใน $\Re^{k-1}$ ซึ่งมีการแจกแจงเป็น $\varphi_k(\cdot;\mathbf{\Sigma}_{k-1})$ และเป็นอิสระต่อกัน

2.2) กำหนดให้ $\mathbf{\Sigma}_k = \begin{bmatrix} \mathbf{\Sigma}_{k-1} & \mathbf{q} \\ \mathbf{q}^T & \mathbf{1} \end{bmatrix}$

2.3) ไปที่ค่า k ถัดไป

ซึ่งการจำลอง q สำหรับ $\mathbf{q}$ จาก φ_k ให้กับ $\mathbf{\Sigma}_{k-1}$ ทำได้ดังนี้

- จำลองตัวแปรสุ่ม Y ที่มีการแจกแจงเบตา (beta distribution) ด้วยพารามิเตอร์ $\alpha_1 = (k-1)/2$ และ $\alpha_2 = (d-k)/2+1$ โดยกำหนดให้ค่าตัวแปรสุ่ม $Y = y$
- ให้ $r = \sqrt{y}$
- สุ่มเวกเตอร์หนึ่งหน่วย (unit vector) $\mathbf{\theta}$ บนพื้นผิว (surface) ของ B^{k-1} โดยที่ B^{k-1} เป็นทรงกลมรัศมี 1 หน่วย (unit ball) บนปริภูมิ (space) $k-1$ มิติ
- ให้ $\mathbf{w} = r\mathbf{\theta}$
- กำหนดให้ $\mathbf{q} = \mathbf{\Sigma}_{k-1}^{\frac{1}{2}}\mathbf{w}$

<u>หมายเหตุ</u> สำหรับ $\mathbf{\Sigma}_{k-1}^{\frac{1}{2}}$ หาได้จากขั้นตอนต่อไปนี้

1) หาเมทริกซ์อินเวอร์ส (inverse matrix) ของ $\mathbf{\Sigma}_{k-1}$ นั่นคือ $\mathbf{\Sigma}_{k-1}^{-1}$

2) ทำการแยกเมทริกซ์ $\mathbf{\Sigma}_{k-1}^{-1}$ แบบโชเลซกี (cholesky decomposition) จะได้ $\mathbf{\Sigma}_{k-1}^{-1} = \mathbf{C}\mathbf{C}^T$ โดย Cholesky Decompositon โดยที่ $\mathbf{C}$ เป็นเมทริกซ์สามเหลี่ยมล่าง (lower triangular matrix)

3) กำหนดให้ $\mathbf{\Sigma}_{k-1}^{\frac{1}{2}} = \mathbf{C}^{-1}$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ซึ่งในขั้นตอนนี้จะได้เมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) $\mathbf{\Sigma}_d$ สำหรับ $d = 3,4,\ldots,15$

2. ทำการแปลง(transform) $\mathbf{\Sigma}_d$ ที่ได้จากขั้นตอนที่ 1. ด้วยฟังก์ชันความสัมพันธ์

$$\Lambda_Z(i,j) = 2\sin\left(\frac{\pi}{6}\Sigma_d(i,j)\right)$$

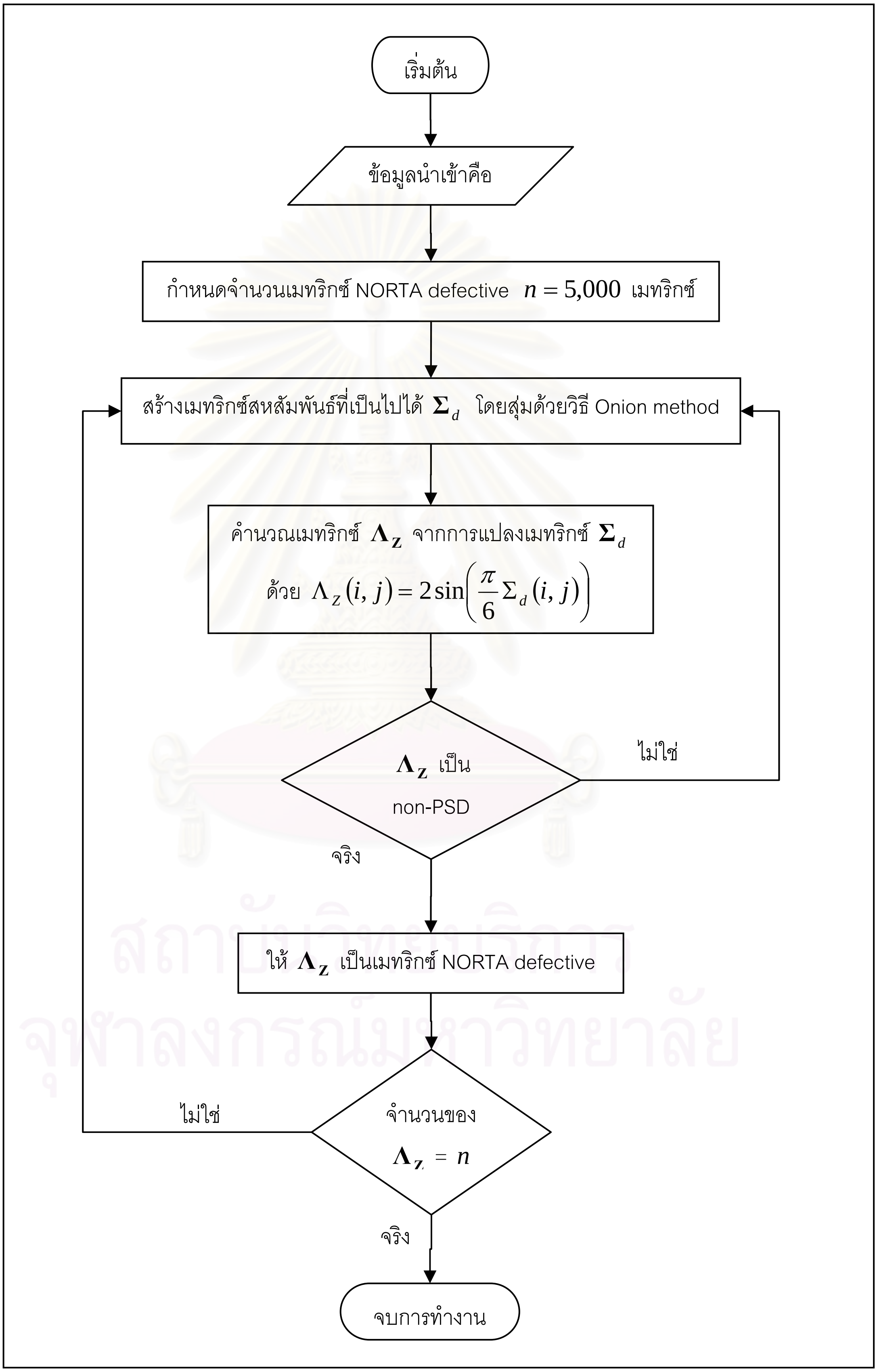
สำหรับ $i \neq j$ และ $\Lambda_Z(i,j) = 1$ เมื่อ $i = j$ โดย $i,j = 1,\ldots,d$ ในขั้นตอนนี้จะได้เมทริกซ์ $\mathbf{\Lambda_Z}$

3. จะกำหนดให้ $\mathbf{\Lambda_Z}$ เมทริกซ์ NORTA defective ถ้า $\mathbf{\Lambda_Z}$ ไม่เป็นเมทริกซ์กึ่งบวกแน่นอน (non-positive semidefinite matrix)

ผังงานแสดงขั้นตอนสำหรับการสร้างเมทริกซ์ NORTA defective ($\mathbf{\Lambda_Z}$) ดังแสดงในแผนภูมิที่ 3.4

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แผนภูมิที่ 3.4 แสดงผังงานสำหรับการสร้างเมทริกซ์ NORTA defective

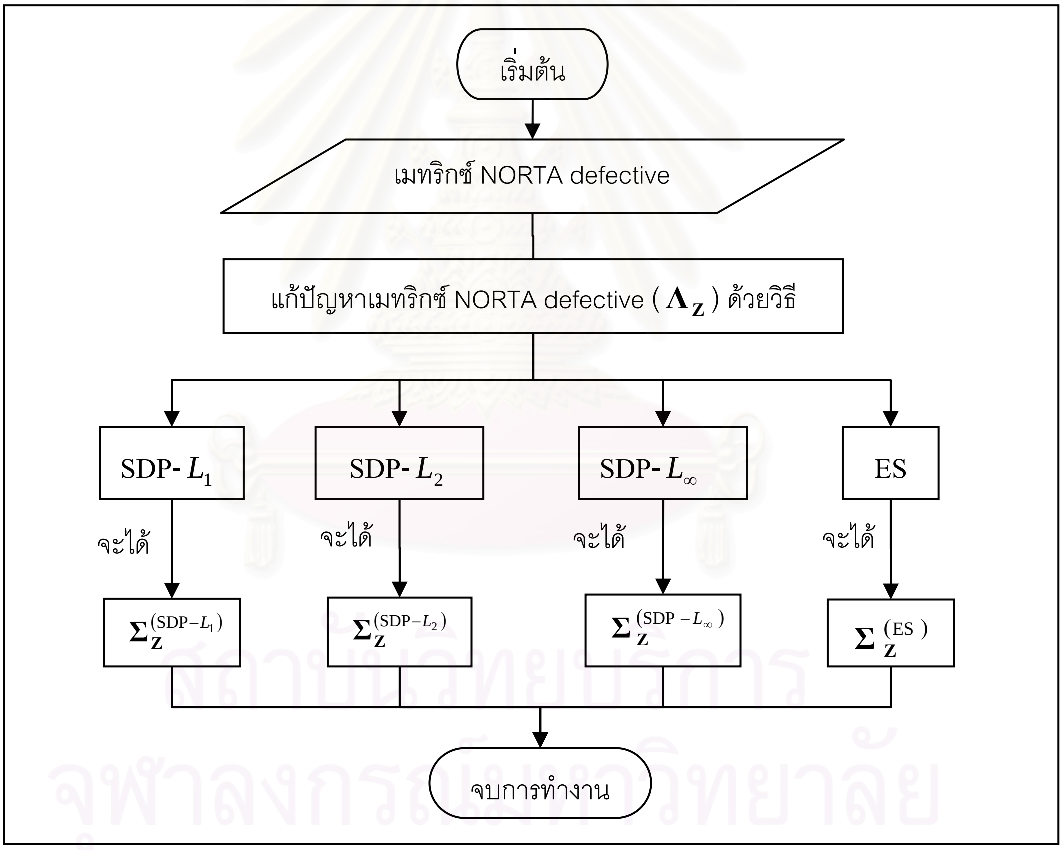


ขั้นตอนที่ 2

แก้ปัญหาเมทริกซ์ NORTA defective ($\mathbf{\Lambda_Z}$) ด้วยวิธี SDP- L_1 ,SDP- L_2 ,SDP- L_∞ และวิธี ES ซึ่งแต่ละวิธีจะทำการค้นหาเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ $\mathbf{\Sigma'_Z}$ ที่ใกล้เคียงกับ $\mathbf{\Lambda_Z}$ มากที่สุด

ซึ่งผังงานแสดงขั้นตอนสำหรับการแก้ปัญหาเมทริกซ์ NORTA defective ($\mathbf{\Lambda_Z}$) ดังแสดงในแผนภูมิที่ 3.5

แผนภูมิที่ 3.5 แสดงผังงานสำหรับ สำหรับการแก้ปัญหาเมทริกซ์ NORTA defectiveด้วยวิธี SDP- L_1 ,SDP- L_2 ,SDP- L_∞ และวิธี ES

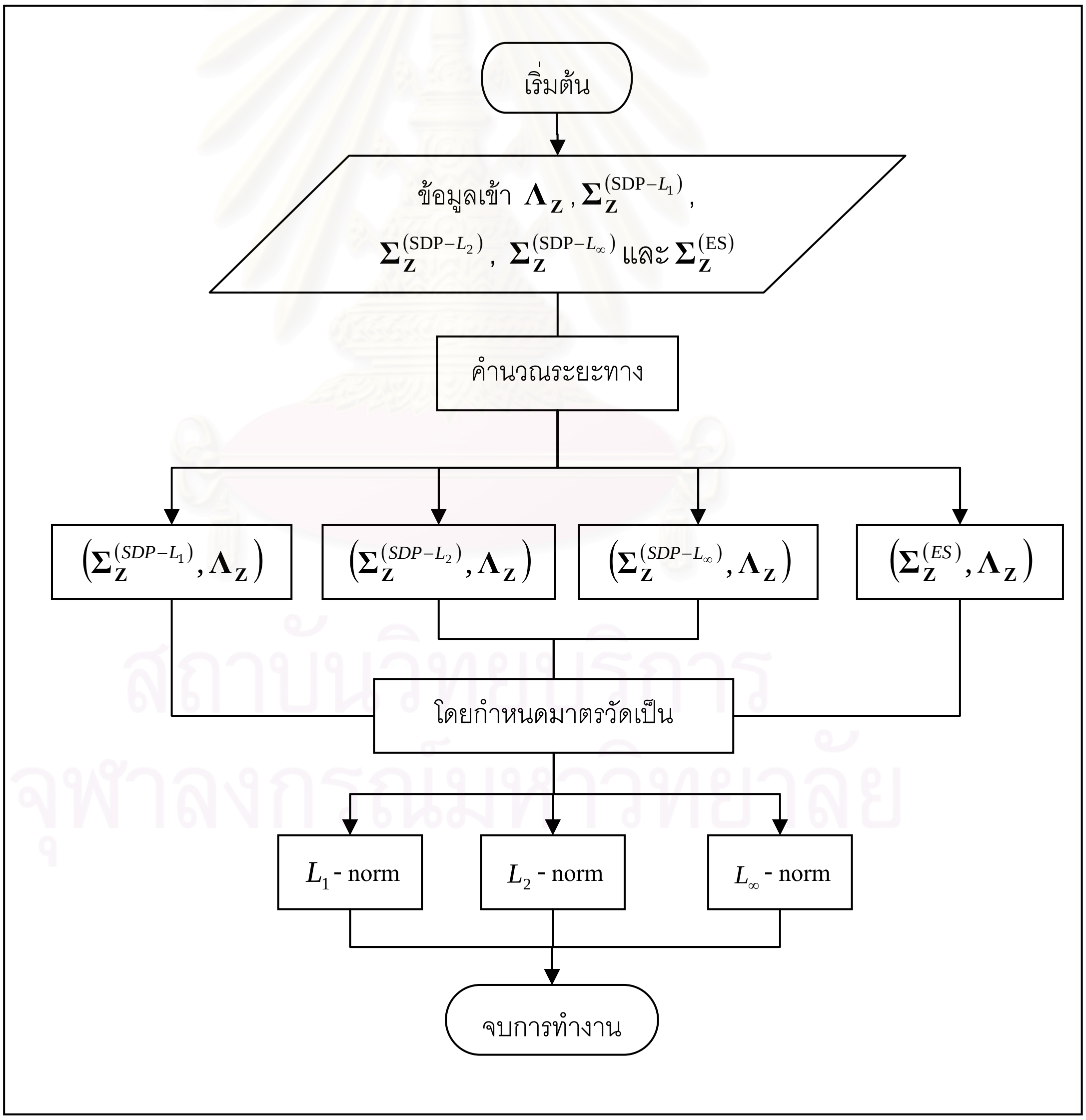


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนที่ 3

คำนวณระยะทางระหว่างเมทริกซ์ NORTA defective ($\mathbf{\Lambda_Z}$) กับเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ $\mathbf{\Sigma'_Z}$ จากวิธี SDP-L_1 , SDP-L_2 , SDP-L_∞ และวิธี ES ซึ่งก็คือ $\mathbf{\Sigma_Z^{(SDP-L_1)}}$, $\mathbf{\Sigma_Z^{(SDP-L_2)}}$, $\mathbf{\Sigma_Z^{(SDP-L_\infty)}}$ และ $\mathbf{\Sigma_Z^{(ES)}}$ โดยใช้มาตรวัดเป็น L_1 - norm , L_2 - norm และ L_∞ - norm ตามลำดับ ดังแสดงผังงานในแผนภูมิที่ 3.6

<u>แผนภูมิที่ 3.6</u> ผังงานแสดงขั้นตอนการคำนวณระยะทาง (distance) ระหว่างเมทริกซ์ NORTA defective ($\mathbf{\Lambda_Z}$) กับเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ $\mathbf{\Sigma'_Z}$ จากวิธี SDP-L_1 , SDP-L_2 , SDP-L_∞ และวิธี ES



หลังจากนั้นทำการศึกษาเปรียบเทียบระยะทางโดยเฉลี่ย (average distance) สำหรับแต่ละมาตรวัดในทุก d ตัวแปร

ขั้นตอนที่ 4

ทำการศึกษาระยะทาง (distance) ระหว่าง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเมทริกซ์ $\mathbf{\Lambda}_{\mathbf{Z}}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞-norm ตามลำดับ จากนั้นทำการแปลงเมทริกซ์ $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}, \mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}, \mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{ES})}$ ด้วยสมการ (2.1) จะได้เมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{ES})}$ ตามลำดับ แล้วทำการวัดระยะทางระหว่าง $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma}_{\mathbf{X}}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞- norm ตามลำดับ ในขั้นตอนนี้ผู้วิจัยกำหนดตัวอย่างเมทริกซ์ NORTA defective ($\mathbf{\Lambda}_{\mathbf{Z}}$) จำนวน 50,000 เมทริกซ์ สำหรับทุกมิติของเวกเตอร์สุ่มที่ทำการศึกษาและนำเสนอผลการศึกษาในรูปของตารางโดยยกตัวอย่างประกอบ

ขั้นตอนที่ 5

ทำการศึกษาเปรียบเทียบประสิทธิภาพ (ความเร็ว) ของวิธีการแก้ปัญหา NORTA defective จากทั้ง 4 วิธีการคือ วิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES โดยกำหนดตัวอย่างเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาในขั้นตอนนี้คือ 1,000 เมทริกซ์ และเครื่องคอมพิวเตอร์ที่ใช้ทำการศึกษามีความเร็วของ CPU เท่ากับ 800 เมกะเฮิร์ทซ์ (megahertz) และมีหน่วยความจำหลัก RAM ขนาด 256 เมกะไบต์ (megabyte)

ขั้นตอนที่ 6

ทำการจำลองเวกเตอร์สุ่มหลายตัวแปรจากโปรแกรม NORTA ที่พัฒนาขึ้น แล้วนำเสนอผลในรูปของตารางพร้อมอภิปรายผล โดยยกตัวอย่างประกอบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

ผลการวิจัย

ในการศึกษาวิจัยครั้งนี้มีวัตถุประสงค์เพื่อพัฒนาเครื่องมือที่ใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ให้สามารถใช้งานบนไมโครซอฟต์ วินโดว์ (Microsoft Windows) ได้ โดยผู้วิจัยได้พัฒนาเป็นไฟล์ Add-in ให้สามารถใช้งานกับไมโครซอฟต์ เอ็กเซล (Microsoft Excel) สำหรับภาษาที่ใช้ในการพัฒนาโปรแกรม NORTA เป็นภาษา C/C++ ในขณะที่วิธีการแก้ปัญหา NOTRA defective ที่อาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ผู้วิจัยได้เลือกใช้วิธี Eigenvalue Shifting (ES) โดยมีรายละเอียดดังนำเสนอต่อไปนี้

4.1 โปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA

สำหรับโปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ที่ผู้วิจัยได้พัฒนาขึ้นครั้งนี้ มีวิธีการใช้งานและข้อจำกัดดังต่อไปนี้

1) สนับสนุนการใช้งานบนระบบปฏิบัติการ Microsoft Windows เวอร์ชัน 2000, XP ขึ้นไป

2) สามารถใช้งานได้บน Microsoft Excel เวอร์ชัน 2002 , 2003 ขึ้นไป

3) สามารถสร้างเวกเตอร์สุ่มหลายตัวแปรที่มีการแจกแจงขอบแบบสม่ำเสมอ (uniform marginal distribution) ในช่วง (0,1)

4) ข้อจำกัดสำหรับการใช้โปรแกรมที่สร้างขึ้นในครั้งนี้คือโปรแกรมสามารถสร้างเวกเตอร์สุ่มแบบสม่ำเสมอเท่านั้นและสร้างเวกเตอร์สุ่มได้ไม่เกิน 10 ตัวแปร และจะต้องปิดโปรแกรมแล้วเปิดใหม่ทุกครั้งหากจำเป็นต้องกำหนดค่าของเมทริกซ์สหสัมพันธ์ใหม่

5) สำหรับการเรียกใช้ฟังก์ชัน NORTA หลังจากติดตั้งโปรแกรม NORTA ลงบน Microsoft Excel แล้ว ทำได้โดยกำหนดขนาดของเวกเตอร์สุ่มที่ต้องการสร้าง ซึ่งเป็นค่าเริ่มต้นให้กับฟังก์ชัน NORTA ในที่นี้ก็คือ จำนวนตัวแปรสุ่มที่ต้องการสร้างและเวกเตอร์สหสัมพันธ์ของตัวแปรสุ่มที่ต้องการนั่นเอง สำหรับการติดตั้งและตัวอย่างการใช้งานโปรแกรมแสดงไว้โดยละเอียดในภาคผนวก ค

4.2 ผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective

พร้อมกันนี้ผู้วิจัยได้ทำการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective ที่อาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA จากสองวิธีการคือ วิธี Semidefinite Programming (SDP) และวิธี Eigenvalue Shifting (ES) โดยเกณฑ์ที่ใช้ในพิจารณาคือระยะทางน้อยที่สุดระหว่างเมทริกซ์ NORTA defective กับเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ที่ได้จากวิธีการทั้งสองดังกล่าว โดยกำหนดมาตรวัดระยะทางเป็น L_1- norm , L_2- norm และ L_∞- norm

ผู้วิจัยทำการศึกษาความแตกต่างของวิธีการแก้ปัญหาเมทริกซ์ NORTA defective ซึ่งได้เสนอผลการวิจัยในลักษณะของตารางและรูปภาพโดยใช้สัญลักษณ์แทนความหมายต่าง ๆ ดังนี้

$\mathbf{X}$ คือเวกเตอร์สุ่มแบบสม่ำเสมอหลายตัวแปร

$\mathbf{\Sigma_X}$ คือเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X}$ ที่ต้องการ

$\mathbf{v_X}$ คือเวกเตอร์สหสัมพันธ์ของเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma_X}$

$\mathbf{Z}$ คือเวกเตอร์สุ่มแบบปกติหลายตัวแปร

$\mathbf{\Sigma_Z}$ คือเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{Z}$

$\mathbf{v_Z}$ คือเวกเตอร์สหสัมพันธ์ของเมทริกซ์ $\mathbf{\Sigma_Z}$

$\mathbf{\Lambda_Z}$ คือเมทริกซ์ NORTA defective

$\mathbf{u}$ คือเวกเตอร์ NORTA defective ของเมทริกซ์ $\mathbf{\Lambda_Z}$

SDP-L_1 คือการแก้ปัญหาเมทริกซ์ NORTA defective $\mathbf{\Lambda_Z}$ ด้วยวิธี SDP โดยกำหนดฟังก์ชันระยะทาง (distance function) เป็น L_1

SDP-L_2 คือการแก้ปัญหาเมทริกซ์ NORTA defective $\mathbf{\Lambda_Z}$ ด้วยวิธี SDP โดยกำหนดฟังก์ชันระยะทาง (distance function) เป็น L_2

SDP-L_∞ คือการแก้ปัญหาเมทริกซ์ NORTA defective $\mathbf{\Lambda_Z}$ ด้วยวิธี SDP โดยกำหนดฟังก์ชันระยะทาง (distance function) เป็น L_∞

ES คือการแก้ปัญหา NORTA defective $\mathbf{\Lambda_Z}$ ด้วยวิธี Eigenvelue Shifting

$\mathbf{\Sigma_Z}^{(\mathrm{SDP}-L_1)}$ คือเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma_Z}$ ที่ได้จากวิธี SDP-L_1

$\mathbf{v_Z}^{(\mathrm{SDP}-L_1)}$ คือเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ของเมทริกซ์ $\mathbf{\Sigma_Z}^{(\mathrm{SDP}-L_1)}$

$\mathbf{\Sigma_Z}^{(\mathrm{SDP}-L_2)}$ คือเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma_Z}$ ที่ได้จากวิธี SDP-L_2

$\mathbf{v_Z}^{(\mathrm{SDP}-L_2)}$ คือเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ของเมทริกซ์ $\mathbf{\Sigma_Z}^{(\mathrm{SDP}-L_2)}$

$\mathbf{\Sigma_Z}^{(\mathrm{SDP}-L_\infty)}$ คือเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma_Z}$ ที่ได้จากวิธี SDP-L_∞

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ คือเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ของเมทริกซ์ $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$

$\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{ES})}$ คือเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma}_{\mathbf{Z}}$ ที่ได้จากวิธี ES

$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ คือเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ของเมทริกซ์ $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{ES})}$

$\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$ คือเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X}$ ที่ได้จากการแปลง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$

$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$ คือเวกเตอร์สหสัมพันธ์ของเมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$

$\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$ คือเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X}$ ที่ได้จากการแปลง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$

$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$ คือเวกเตอร์สหสัมพันธ์ของเมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$

$\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ คือเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X}$ ที่ได้จากการแปลง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$

$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ คือเวกเตอร์สหสัมพันธ์ของเมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$

$\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{ES})}$ คือเมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X}$ ที่ได้จากการแปลง $\mathbf{\Sigma}_{\mathbf{Z}}^{(\mathrm{ES})}$

$\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ คือเวกเตอร์สหสัมพันธ์ของเมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}^{(\mathrm{ES})}$

$d(\mathbf{u},\mathbf{v})$ คือระยะทางระหว่างเวกเตอร์ $\mathbf{u}$ กับ $\mathbf{v}$

<u>หมายเหตุ</u>

สำหรับการวิจัยครั้งนี้ เวกเตอร์ $\mathbf{u}, \mathbf{v}_{\mathbf{Z}}, \mathbf{v}_{\mathbf{X}}$ เป็นเวกเตอร์ที่ประกอบสมาชิกที่เหนือสมาชิกที่อยู่ในแนวเส้นทแยงมุมหลักของเมทริกซ์ $\mathbf{\Lambda}_{\mathbf{Z}}$,$\mathbf{\Sigma}_{\mathbf{Z}}$, $\mathbf{\Sigma}_{\mathbf{X}}$ ตามลำดับ ยกตัวอย่างเช่น สมมติให้

$$\mathbf{\Lambda}_{\mathbf{Z}} = \begin{bmatrix} 1 & u_1 & u_2 \\ u_1 & 1 & u_3 \\ u_2 & u_3 & 1 \end{bmatrix}$$

ดังนั้นเวกเตอร์ $\mathbf{u}$ คือ $\mathbf{u} = (u_1, u_2, u_3)^T$ หรือ สมมติให้

$$\mathbf{\Sigma}_{\mathbf{Z}} = \begin{bmatrix} 1 & v_1 & v_2 & v_3 \\ v_1 & 1 & v_4 & v_5 \\ v_2 & v_4 & 1 & v_6 \\ v_3 & v_5 & v_6 & 1 \end{bmatrix}$$

จะได้ $\mathbf{v}_{\mathbf{Z}} = (v_1, v_2, v_3, v_4, v_5, v_6)^T$ เป็นต้น

การนำเสนอผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective ได้แบ่งการนำเสนอผลการศึกษาออกเป็น 5 ส่วน โดยที่ 3 ส่วนแรกแบ่งการศึกษาตามมาตรวัดระยะทาง (distance measure) ซึ่งผู้วิจัยได้กำหนดจำนวนของเมทริกซ์ $\mathbf{\Lambda}_{\mathbf{Z}}$ ที่ใช้ทำการศึกษาจำนวน 5,000 เมทริกซ์ สำหรับทุก d มิติ โดยที่ $d = 3,4,\ldots,15$ ขณะที่ผลการศึกษาความแตกต่างของวิธี SDPในส่วนที่ 4 ผู้วิจัยได้กำหนดจำนวนของเมทริกซ์ $\mathbf{\Lambda}_{\mathbf{Z}}$ ที่ใช้ทำการศึกษาจำนวน 50,000 เมทริกซ์ สำหรับทุก d มิติ โดยที่ $d = 3,4$ และในส่วนที่ 5 เป็นแสดงผลลัพธ์จากการใช้โปรแกรมที่พัฒนาขึ้น สำหรับรายละเอียดของผลการศึกษาในแต่ละส่วนมีดังนี้

ส่วนที่ 1 ผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยพิจารณาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยใช้มาตรวัด L_1 - norm

ส่วนที่ 2 ผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยพิจารณาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยใช้มาตรวัด L_2 - norm

ส่วนที่ 3 ผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยพิจารณาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยใช้มาตรวัด L_∞ - norm

ส่วนที่ 4 ผลการศึกษาจากการวัดระยะทางระหว่าง $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_1 - norm , L_2 - norm และ L_∞ -norm ตามลำดับ จากนั้นทำการแปลงเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ด้วยสมการ (2.1) จะได้ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ ตามลำดับ แล้ววัดระยะทางระหว่าง $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ โดยกำหนดมาตรวัดเป็น L_1 - norm , L_2 - norm และ L_∞ - norm

ส่วนที่ 5 ผลการศึกษาประสิทธิภาพ (ความเร็ว) ของวิธีการแก้ปัญหา NORTA defective จากทั้ง 4 วิธีการคือ วิธี SDP- L_1 , SDP- L_2 , SDP- L_∞ และวิธี ES โดยกำหนดตัวอย่างเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาในขั้นตอนนี้คือ 1,000 เมทริกซ์ และเครื่องคอมพิวเตอร์ที่ใช้ทำการศึกษามีความเร็วของ CPU เท่ากับ 800 เมกะเฮิร์ทซ์ (megahertz) และมีหน่วยความจำหลัก RAM ขนาด 256 เมกะไบต์ (megabyte)

ส่วนที่ 6 ผลการจำลองเวกเตอร์สุ่มหลายตัวแปรจากโปรแกรม NORTA ที่พัฒนาขึ้น

ต่อไปนี้จะกล่าวถึงรายละเอียดของผลการวิจัยในแต่ละส่วน

<u>ส่วนที่ 1</u> ผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยพิจารณาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยใช้มาตรวัด L_1- norm

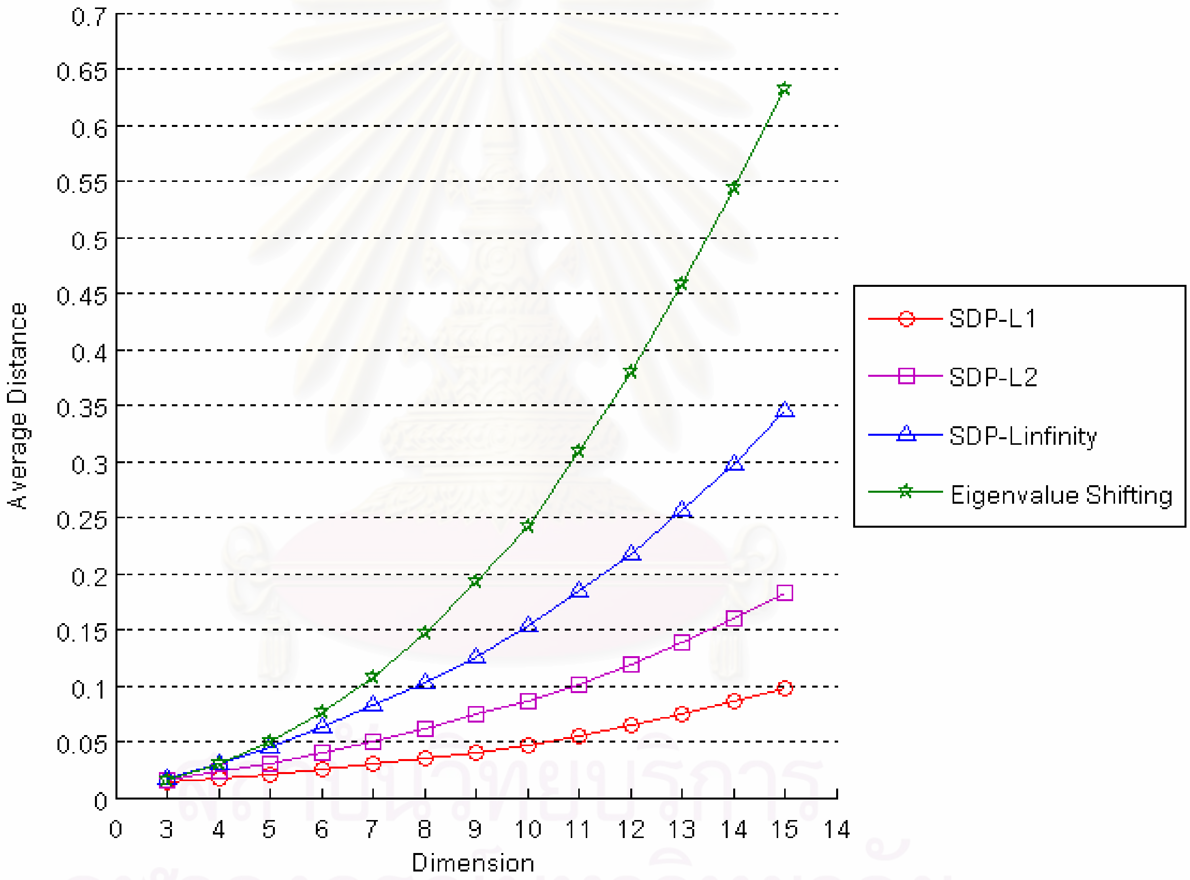
สำหรับการศึกษาระยะทางโดยเฉลี่ย (average distance) ระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ ($\mathbf{v}_{\mathbf{Z}}$) ที่ได้จากการแก้ปัญหา NORTA defective ด้วยวิธี SDP-L_1 , SDP-L_2 , SDP-L_∞ และวิธี ES คือ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ตามลำดับ กับเวกเตอร์ NORTA defective ($\mathbf{u}$) โดยกำหนดมาตรวัดเป็น L_1- norm ได้ระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}$ กับ $\mathbf{u}$ สำหรับแต่ละวิธีดังแสดงผลในตารางที่ 4.1

<u>ตารางที่ 4.1</u> แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้จากวิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES กับเวกเตอร์ NORTA defective สำหรับแต่ละมิติ (dimension) ของ $\mathbf{\Lambda}_{\mathbf{Z}}$ โดยกำหนดมาตรวัดเป็น L_1- norm

มิติ	**ระยะทางโดยเฉลี่ย (average distance) ระหว่าง**			
	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$ **กับ u**	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$ **กับ u**	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ **กับ u**	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ **กับ u**
3	**0.01336851**	0.01567182	0.01648898	0.01542814
4	**0.01693522**	0.02275397	0.02965510	0.03002324
5	**0.02133829**	0.03153256	0.04602560	0.05149511
6	**0.02571162**	0.04041634	0.06374470	0.07727912
7	**0.03008835**	0.04983118	0.08259394	0.10879305
8	**0.03485090**	0.06014011	0.10339699	0.14658118
9	**0.04121700**	0.07297411	0.12852537	0.19430936
10	**0.04830935**	0.08725008	0.15560603	0.24788286
11	**0.05527101**	0.10159042	0.18512909	0.30902130
12	**0.06457444**	0.11968895	0.21993564	0.38003880
13	**0.07439652**	0.13888371	0.25712917	0.45973457
14	**0.08592684**	0.16054367	0.29911511	0.54445389
15	**0.09748156**	0.18321961	0.34413466	0.64126286

จากผลการศึกษา โดยพิจารณาระยะทางโดยเฉลี่ย (average distance) ระหว่างเวกเตอร์ $\mathbf{v_Z}$ กับ $\mathbf{u}$ ด้วยมาตรวัด L_1- norm เมื่อมิติ (dimension) ของ $\mathbf{\Lambda_Z}$ สูงขึ้น พบว่าเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ $\mathbf{v_Z}$ ที่ได้จากวิธี SDP-L_1 , SDP-L_2 , SDP-L_∞ และ ES มีระยะทางโดยเฉลี่ยเรียงจากน้อยไปมากตามลำดับ ดังแสดงในรูปที่ 4.1

<u>รูปที่ 4.1</u> แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{u}$ กับเวกเตอร์ $\mathbf{v_Z}^{(\text{SDP}-L_1)}$, $\mathbf{v_Z}^{(\text{SDP}-L_2)}$, $\mathbf{v_Z}^{(\text{SDP}-L_\infty)}$ และ $\mathbf{v_Z}^{(\text{ES})}$ ที่ได้จากวิธี SDP-L_1 ,SDP-L_2 ,SDP-L_∞ และ ES ตามลำดับ โดยกำหนดมาตรวัดเป็น L_1- norm



ส่วนที่ 2 ผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยพิจารณาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยใช้มาตรวัด L_2- norm

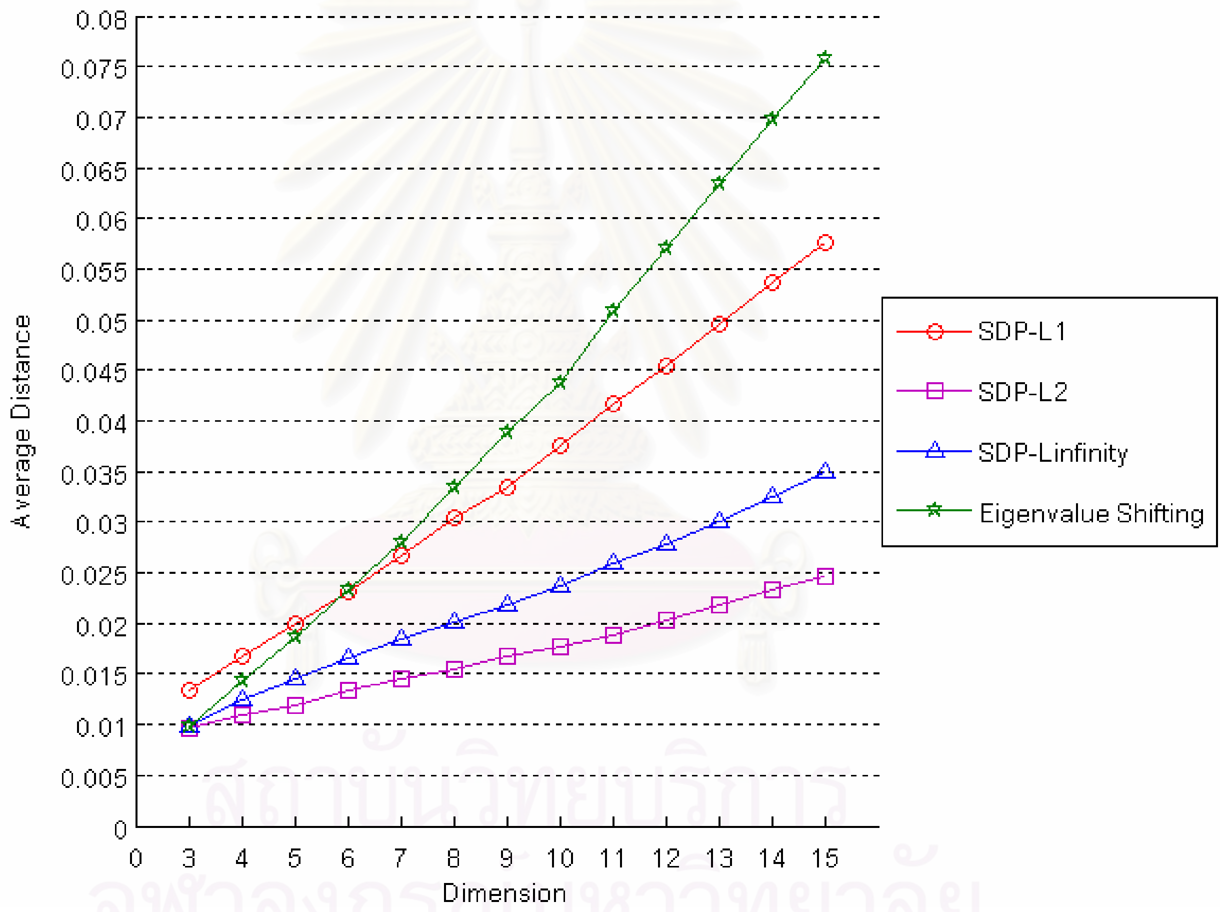
สำหรับการศึกษาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ ($\mathbf{v}_{\mathbf{Z}}$) ที่ได้จากการแก้ปัญหา NORTA defective ด้วยวิธี SDP-L_1 , SDP-L_2 , SDP-L_∞ และวิธี ES คือ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ตามลำดับ กับเวกเตอร์ NORTA defective ($\mathbf{u}$) โดยกำหนดมาตรวัดเป็น L_2- norm ได้ระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}$ กับ $\mathbf{u}$ สำหรับแต่ละวิธีดังแสดงผลในตารางที่ 4.2

ตารางที่ 4.2 แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้จากวิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES กับเวกเตอร์ NORTA defective สำหรับแต่ละมิติของ $\mathbf{\Lambda}_{\mathbf{Z}}$ โดยกำหนดมาตรวัดเป็น L_2- norm

มิติ	**ระยะทางโดยเฉลี่ย (average distance) ระหว่าง**			
	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$ **กับ** $\mathbf{u}$	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$ **กับ** $\mathbf{u}$	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ **กับ** $\mathbf{u}$	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ **กับ** $\mathbf{u}$
3	0.01313784	**0.00927594**	0.00951992	0.00989792
4	0.01648326	**0.01063119**	0.01217308	0.01404923
5	0.02039556	**0.01210463**	0.01470520	0.01900544
6	0.02387517	**0.01328404**	0.01678178	0.02368547
7	0.02716003	**0.01423564**	0.01846809	0.02837412
8	0.03016204	**0.01521947**	0.02013938	0.03333411
9	0.03424833	**0.01654504**	0.02218783	0.03919301
10	0.03806435	**0.01778966**	0.02406787	0.04487136
11	0.04134975	**0.01897599**	0.02598838	0.05072907
12	0.04539174	**0.02044292**	0.02819301	0.05711717
13	0.04939449	**0.02183559**	0.03027779	0.06368960
14	0.05353891	**0.02335786**	0.03258600	0.06997980
15	0.05752355	**0.02484392**	0.03485508	0.07683825

จากผลการศึกษา โดยพิจารณาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}$ กับ $\mathbf{u}$ ด้วยมาตรวัด L_2 - norm เมื่อมิติของ $\mathbf{\Lambda}_{\mathbf{Z}}$ สูงขึ้น พบว่าเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ $\mathbf{v}_{\mathbf{Z}}$ ที่ได้จากวิธี SDP- L_2 , SDP- L_∞ , SDP- L_1 และ ES มีระยะทางโดยเฉลี่ยเรียงจากน้อยไปมากตามลำดับ ดังแสดงในรูปที่ 4.2

รูปที่ 4.2 แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{u}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ที่ได้จากวิธี SDP- L_1, SDP- L_2 , SDP- L_∞ และ ES ตามลำดับ โดยกำหนดมาตรวัดเป็น L_2 - norm



<u>ส่วนที่ 3</u> ผลการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยพิจารณาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยใช้มาตรวัด L_∞ - norm

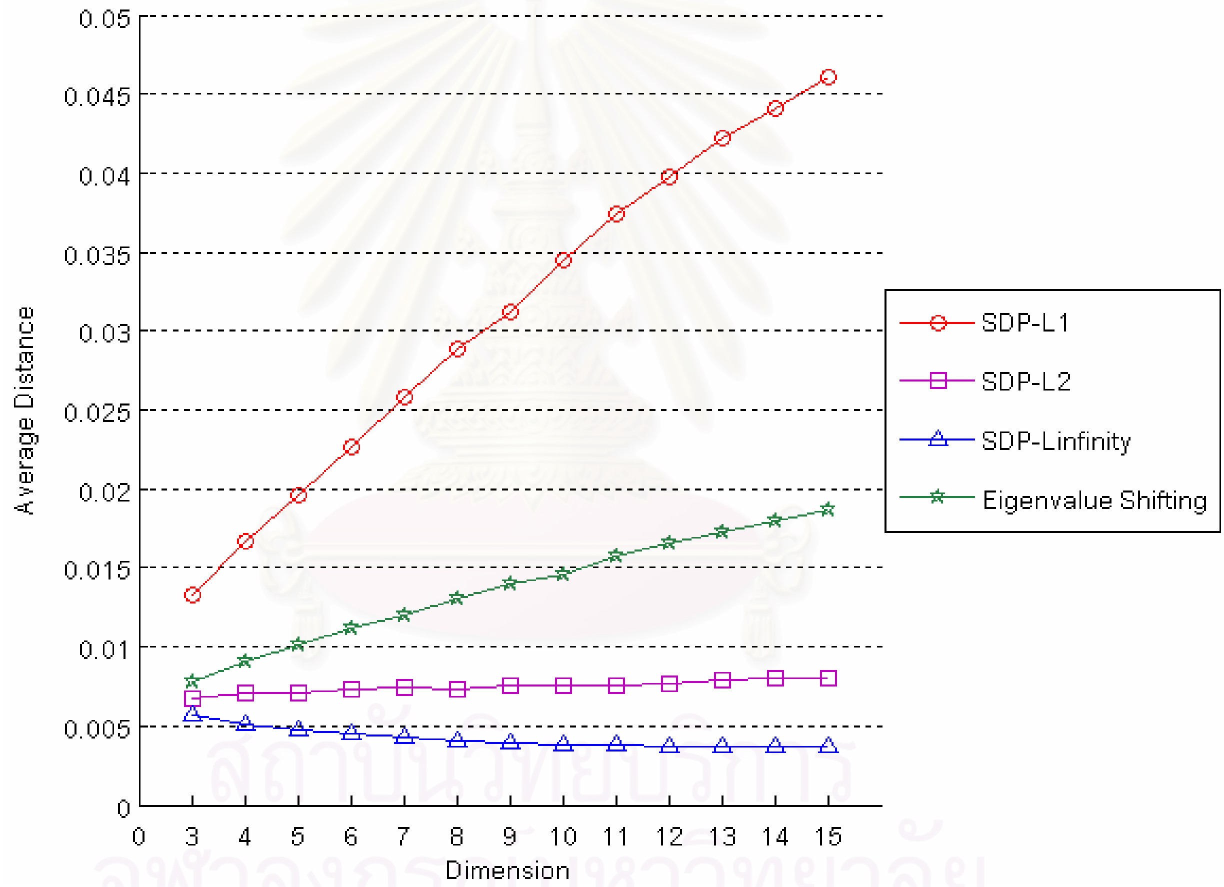
สำหรับการศึกษาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ $(\mathbf{v}_{\mathbf{Z}})$ ที่ได้จากการแก้ปัญหา NORTA defective ด้วยวิธี SDP-L_1 , SDP-L_2 , SDP-L_∞ และวิธี ES คือ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ตามลำดับ กับเวกเตอร์ NORTA defective ($\mathbf{u}$) โดยกำหนดมาตรวัดเป็น L_∞ - norm ได้ระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}$ กับ $\mathbf{u}$ สำหรับแต่ละวิธีดังแสดงผลในตารางที่ 4.3

<u>ตารางที่ 4.3</u> แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์สหสัมพันธ์ที่เป็นไปได้จากวิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES กับเวกเตอร์ NORTA defective สำหรับแต่ละมิติของ $\mathbf{\Lambda}_{\mathbf{Z}}$ โดยกำหนดมาตรวัดเป็น L_∞ - norm

มิติ	**ระยะทางโดยเฉลี่ย** (average distance) **ระหว่าง**			
	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$ **กับ** $\mathbf{u}$	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$ **กับ** $\mathbf{u}$	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ **กับ** $\mathbf{u}$	$\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ **กับ** $\mathbf{u}$
3	0.01306429	0.00654677	**0.00549667**	0.00774197
4	0.01634319	0.00686763	**0.00505428**	0.00897107
5	0.02008198	0.00716758	**0.00477519**	0.01033263
6	0.02328077	0.00728414	**0.00452828**	0.01138844
7	0.02623921	0.00726202	**0.00424704**	0.01217934
8	0.02866126	0.00729319	**0.00404934**	0.01300918
9	0.03200827	0.00744234	**0.00396904**	0.01407982
10	0.03484962	0.00752838	**0.00386213**	0.01487389
11	0.03709613	0.00765827	**0.00379759**	0.01565049
12	0.03964082	0.00780627	**0.00376430**	0.01654062
13	0.04197671	0.00788911	**0.00371260**	0.01735929
14	0.04415246	0.00804664	**0.00370009**	0.01809084
15	0.04605556	0.00818418	**0.00367860**	0.01880602

จากผลการศึกษา โดยพิจารณาระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}$ กับ $\mathbf{u}$ ด้วยมาตรวัดเป็น L_∞ - norm เมื่อมิติของ $\mathbf{\Lambda}_{\mathbf{Z}}$ สูงขึ้น พบว่าเวกเตอร์สหสัมพันธ์ที่เป็นไปได้ $\mathbf{v}_{\mathbf{Z}}$ ที่ได้จากวิธี SDP- L_∞ , SDP- L_2 , ES และ SDP- L_1 มีระยะทางโดยเฉลี่ยของเรียงจากน้อยไปมากตามลำดับ ดังแสดงในรูปที่ 4.3

<u>รูปที่ 4.3</u> แสดงระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{u}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ที่ได้จากวิธีSDP- L_1 ,SDP- L_2 ,SDP- L_∞ และ ES ตามลำดับ โดยกำหนดมาตรวัดเป็น L_∞ - norm



ส่วนที่ 4 ผลการศึกษาจากการวัดระยะทางระหว่าง $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞-norm ตามลำดับ จากนั้นทำการแปลงเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ จะได้ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ ตามลำดับ แล้วทำการวัดระยะทางระหว่าง $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞- norm

สำหรับการศึกษาวิจัยในส่วนนี้ มีวัตถุประสงค์เพื่อศึกษาเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma}_{\mathbf{X}}$ ที่ต้องการ ซึ่งจะต้องกำหนดให้กับกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ในกรณีที่เกิดปัญหา NORTA defective หลังจากทำการแก้ปัญหาด้วยวิธี SDP- L_1 , SDP- L_2 , SDP- L_∞ และ วิธี ES ผลที่ได้คือเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma}_{\mathbf{X}}$ โดยประมาณ ซึ่งเมทริกซ์สหสัมพันธ์ดังกล่าวนี้จะมีความใกล้เคียงกับเมทริกซ์สหสัมพันธ์ $\mathbf{\Sigma}_{\mathbf{X}}$ ที่ต้องการมากน้อยเพียงใด

ผู้วิจัยได้กำหนดมิติของเมทริกซ์ $\mathbf{\Lambda}_{\mathbf{Z}}$ ที่ใช้ทำการศึกษาเป็น 3 มิติ โดยทำการสุ่มตัวอย่างเมทริกซ์ $\mathbf{\Lambda}_{\mathbf{Z}}$ จำนวน 50,000 เมทริกซ์ โดยทำการศึกษาระยะทางโดยเฉลี่ยระหว่าง $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ จากนั้นทำการแปลงเวกเตอร์ด้วยความสัมพันธ์ใน (2.1) แล้วศึกษาระยะทางโดยเฉลี่ยระหว่าง $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ จากผลการศึกษาพบว่าระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ซึ่งเป็นเวกเตอร์ที่ได้จากการแก้ปัญหาด้วยวิธี SDP- L_1 , SDP- L_2 , SDP- L_∞ และ วิธี ES กับเวกเตอร์ $\mathbf{u}$ โดยส่วนใหญ่แล้วให้ผลสอดคล้องกันกับระยะทางโดยเฉลี่ยระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ ที่ต้องการ กล่าวคือ ถ้า $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$ มีระยะทางใกล้กับ $\mathbf{u}$ มากที่สุด จะส่งผลให้ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$ มีระยะทางใกล้กับ $\mathbf{v}_{\mathbf{X}}$ มากที่สุด ด้วย เป็นต้น ทั้งนี้ขึ้นอยู่มาตรวัดระยะทางที่กำหนดขึ้น ดังนำเสนอรายละเอียดในตารางที่ 4.4 และตารางที่ 4.5

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.4 แสดงระยะทางโดยเฉลี่ยระหว่าง $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞- norm

มาตรวัดระยะทาง	ระยะทางโดยเฉลี่ยระหว่าง			
	$\mathbf{u}$ กับ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$	$\mathbf{u}$ กับ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$	$\mathbf{u}$ กับ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$	$\mathbf{u}$ กับ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$
L_1- norm	**0.01353565**	0.01584189	0.01665163	0.01559755
L_2- norm	0.01329999	**0.00937212**	0.00961383	0.00999403
L_∞- norm	0.01322835	0.00660022	**0.00555089**	0.00780408

ตารางที่ 4.5 แสดงระยะทางโดยเฉลี่ยระหว่าง $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ โดยกำหนดมาตรวัดเป็น L_1- norm , L_2- norm และ L_∞- norm

มาตรวัดระยะทาง	ระยะทางโดยเฉลี่ยระหว่าง			
	$\mathbf{v}_{\mathbf{X}}$ กับ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$	$\mathbf{v}_{\mathbf{X}}$ กับ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$	$\mathbf{v}_{\mathbf{X}}$ กับ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$	$\mathbf{v}_{\mathbf{X}}$ กับ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$
L_1- norm	**0.01394093**	0.01589174	0.01659226	0.01578001
L_2- norm	0.01370419	**0.00947284**	0.00958485	0.01020173
L_∞- norm	0.01363225	0.00683823	**0.00574180**	0.00808541

นอกจากนี้ จากผลการศึกษายังพบว่ามีเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ บางเวกเตอร์ เมื่อทำการวัดระยะทางระหว่าง $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ กับเวกเตอร์ $\mathbf{u}$ แล้วให้ผลลัพธ์ไม่สอดคล้องกันกับระยะทางระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ กล่าวคือถ้า $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$ มีระยะทางใกล้กับ $\mathbf{u}$ มากที่สุด เมื่อทำการแปลงด้วยความสัมพันธ์ใน (2.1) แล้วพบว่า $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$ ไม่ได้มีระยะทางใกล้กับ $\mathbf{v}_{\mathbf{X}}$ มากที่สุดด้วย เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สำหรับที่มิติของเมทริกซ์ $\mathbf{\Lambda_Z}$ ที่ใช้ทำการศึกษาเป็น 3 มิติ โดยทำการสุ่มตัวอย่าง $\mathbf{\Lambda_Z}$ จำนวน 50,000 พบว่า

1) เมื่อกำหนดมาตรวัดระยะทางเป็น L_1- norm มีเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$ จำนวน 3 เวกเตอร์หรือคิดเป็นร้อยละ 0.006 ซึ่งมีระยะทางใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด แต่เมื่อทำการแปลงเป็น $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}$ กลับพบว่า $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}$ ไม่ได้มีระยะทางใกล้กับ $\mathbf{v}_\mathbf{X}$ มากที่สุด

2) เมื่อกำหนดมาตรวัดระยะทางเป็น L_2- norm มีเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$ จำนวน 745 เวกเตอร์หรือคิดเป็นร้อยละ 1.49 ซึ่งมีระยะทางใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด แต่เมื่อทำการแปลงเป็น $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}$ กลับพบว่า $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}$ ไม่ได้มีระยะทางใกล้กับ $\mathbf{v}_\mathbf{X}$ มากที่สุด

ในขณะที่เมื่อกำหนดมาตรวัดระยะทางเป็น L_∞- norm ไม่พบเวกเตอร์ที่มีลักษณะดังกล่าวข้างต้น แต่เมื่อเพิ่มมิติของเมทริกซ์ $\mathbf{\Lambda_Z}$ ที่ใช้ทำการศึกษาเป็น 4 มิติ และทำการสุ่มตัวอย่าง $\mathbf{\Lambda_Z}$ จำนวน 50,000 พบว่าเมื่อกำหนดมาตรวัดระยะทางเป็น L_∞- norm มีเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ จำนวน 1 เวกเตอร์หรือคิดเป็นร้อยละ 0.002 ซึ่งมีระยะทางใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด แต่เมื่อทำการแปลงเป็น $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ กลับพบว่า $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ ไม่ได้มีระยะทางใกล้กับ $\mathbf{v}_\mathbf{X}$ มากที่สุด โดยที่กำหนดมาตรวัดระยะทางเป็น L_1- norm และ L_2- norm พบเวกเตอร์ที่มีลักษณะดังกล่าวคิดเป็นร้อยละ 0.002 และ 0.042 ตามลำดับ

ผู้วิจัยได้ยกกรณีตัวอย่างของเวกเตอร์ลักษณะดังกล่าวข้างต้น 3 กรณี โดยแบ่งตามมาตรวัดระยะทาง L_1- norm , L_2- norm และ L_∞- norm ซึ่งมีรายละเอียดดังต่อไปนี้

<u>กรณีที่ 1</u> ผลการศึกษาการแก้ปัญหา NORTA defective โดยกำหนดมาตรวัดระยะทางเป็น L_1- norm

เมื่อกำหนดให้เมทริกซ์ $\mathbf{\Lambda_Z}$ ที่ใช้ทำการศึกษาเป็น 3 มิติ และทำศึกษาระยะทางระหว่าง $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ ภายหลังการแปลงเวกเตอร์ ทำการวัดระยะทางระหว่าง $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{X}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_\mathbf{X}$ พบว่ามีเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$ ที่มีระยะทางใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด แต่เมื่อทำการแปลงเวกเตอร์ดังกล่าวด้วยความสัมพันธ์ใน (2.1) ซึ่งจะได้เวกเตอร์ $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}$ แล้วปรากฏว่าเวกเตอร์ $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}$ ไม่ได้มีระยะทางใกล้กับเวกเตอร์ $\mathbf{v}_\mathbf{X}$ มากที่สุดเมื่อเปรียบเทียบกับเวกเตอร์ $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{X}^{(\mathrm{ES})}$ ดังนำเสนอรายละเอียดโดยยกกรณีตัวอย่างต่อไปนี้

ให้เมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X}$ เป็น

$$\mathbf{\Sigma_X} = \begin{bmatrix} 1 & -0.294078450 & -0.512600215 \\ -0.294078450 & 1 & -0.626379447 \\ -0.512600215 & -0.626379447 & 1 \end{bmatrix}$$

และเวกเตอร์ $\mathbf{v_X} = (-0.294078450, -0.512600215, -0.626379447)$ จากนั้นแปลง $\mathbf{\Sigma_X}$ ด้วย (2.1) จะได้เมทริกซ์ NORTA defective คือ

$$\mathbf{\Lambda_Z} = \begin{bmatrix} 1 & -0.306742748 & -0.530372040 \\ -0.306742748 & 1 & -0.644246654 \\ -0.530372040 & -0.644246654 & 1 \end{bmatrix}$$

และเวกเตอร์ $\mathbf{u} = (-0.306742748, -0.530372040, -0.644246654)$ ทำการแก้ปัญหาเวกเตอร์ NORTA defective ด้วยวิธี SDP- L_1 , SDP- L_2 ,SDP- L_∞ และวิธี ES ตามลำดับได้ผลดังนี้

$$\mathbf{v_Z}^{(\mathrm{SDP}-L_1)} = (-0.306737126, -0.530360831, -0.644222384)$$
$$\mathbf{v_Z}^{(\mathrm{SDP}-L_2)} = (-0.306688781, -0.530314889, -0.644184596)$$
$$\mathbf{v_Z}^{(\mathrm{SDP}-L_\infty)} = (-0.306728782, -0.530357323, -0.644231358)$$
$$\mathbf{v_Z}^{(\mathrm{ES})} = (-0.306734214, -0.530357284, -0.644228730)$$

เมื่อทำการวัดระยะทางระหว่างเวกเตอร์ $\mathbf{v_Z}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v_Z}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v_Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v_Z}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ พบว่าเวกเตอร์ $\mathbf{v_Z}^{(\mathrm{SDP}-L_1)}$ ใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด รองลงมาคือ $\mathbf{v_Z}^{(\mathrm{ES})}$, $\mathbf{v_Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v_Z}^{(\mathrm{SDP}-L_2)}$ ตามลำดับ ดังนำเสนอผลในตารางที่ 4.6

ตารางที่ 4.6 แสดงระยะทางระหว่าง $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_1- norm

	มีระยะทางเป็น
$d\left(\mathbf{u}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}\right)$	**0.0000411011**
$d\left(\mathbf{u}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}\right)$	0.0001731764
$d\left(\mathbf{u}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}\right)$	0.0000439805
$d\left(\mathbf{u}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}\right)$	0.0000412147

จากนั้นทำการแปลงเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ด้วยความสัมพันธ์ใน (2.1) จะได้ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ ตามลำดับ โดยมีผลดังนี้

$$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)} = (-0.294073017, -0.512589113, -0.626354966)$$
$$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)} = (-0.294026298, -0.512543613, -0.626316850)$$
$$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)} = (-0.294064953, -0.512585638, -0.626364018)$$
$$\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})} = (-0.294070203, -0.512585600, -0.626361367)$$

เมื่อทำการวัดระยะทางระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ พบว่าเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ ใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด รองลงมาคือ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$ ตามลำดับ ดังนำเสนอผลในตารางที่ 4.7

ตารางที่ 4.7 แสดงระยะทางระหว่าง $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ โดยกำหนดมาตรวัดเป็น L_1- norm

	มีระยะทางเป็น
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}\right)$	0.0000410153
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}\right)$	0.0001713507
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}\right)$	0.0000435022
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}\right)$	**0.0000409416**

กรณีที่ 2 ผลการศึกษาการแก้ปัญหา NORTA defective โดยกำหนดมาตรวัดระยะทางเป็น L_2 - norm

เมื่อกำหนดให้เมทริกซ์ $\mathbf{\Lambda_Z}$ ที่ใช้ทำการศึกษาเป็น 3 มิติ และทำการศึกษาระยะทางระหว่าง $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ ภายหลังการแปลงเวกเตอร์ ทำการวัดระยะทางระหว่าง $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{X}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v_X}$ พบว่ามีเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$ ที่มีระยะทางใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด แต่เมื่อทำการแปลงเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)}$ ด้วยความสัมพันธ์ใน (2.1) ซึ่งจะได้เวกเตอร์ $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}$ แล้วปรากฏว่าเวกเตอร์ $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_2)}$ ไม่ได้มีระยะทางใกล้กับเวกเตอร์ $\mathbf{v_X}$ มากที่สุดเมื่อเปรียบเทียบกับเวกเตอร์ $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_\mathbf{X}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{X}^{(\mathrm{ES})}$ ดังนำเสนอรายละเอียดโดยยกกรณีตัวอย่างต่อไปนี้

ให้เมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X}$ เป็น

$$\mathbf{\Sigma_X} = \begin{bmatrix} 1 & 0.145817457 & 0.686029799 \\ 0.145817457 & 1 & -0.579339252 \\ 0.686029799 & -0.579339252 & 1 \end{bmatrix}$$

และเวกเตอร์ $\mathbf{v_X} = (0.145817457, 0.686029799, -0.579339252)$ จากนั้นแปลง $\mathbf{\Sigma_X}$ ด้วย (2.1) จะได้เมทริกซ์ NORTA defective คือ

$$\mathbf{\Lambda_Z} = \begin{bmatrix} 1 & 0.152551372 & 0.703058975 \\ 0.152551372 & 1 & -0.597421277 \\ 0.703058975 & -0.597421277 & 1 \end{bmatrix}$$

และเวกเตอร์ $\mathbf{u} = (0.152551372, 0.703058975, -0.597421277)$ ทำการแก้ปัญหาเวกเตอร์ NORTA defective ด้วยวิธี SDP- L_1 , SDP- L_2 ,SDP- L_∞ และวิธี ES ตามลำดับ ได้ผลดังนี้

$$\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_1)} = (0.152548723, 0.701411280, -0.597414209)$$
$$\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_2)} = (0.152030223, 0.702340795, -0.596784411)$$
$$\mathbf{v}_\mathbf{Z}^{(\mathrm{SDP}-L_\infty)} = (0.151916968, 0.702424366, -0.596786736)$$
$$\mathbf{v}_\mathbf{Z}^{(\mathrm{ES})} = (0.152363459, 0.702192949, -0.596685375)$$

เมื่อทำการวัดระยะทางระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ พบว่าเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$ ใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด รองลงมาคือ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$ ตามลำดับ ดังนำเสนอผลในตารางที่ 4.8

ตารางที่ 4.8 แสดงระยะทางระหว่าง $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_2 - norm

	มีระยะทางเป็น
$d\left(\mathbf{u}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}\right)$	0.00164771268
$d\left(\mathbf{u}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}\right)$	**0.00109223516**
$d\left(\mathbf{u}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}\right)$	0.00109901746
$d\left(\mathbf{u}, \mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}\right)$	0.00115189637

จากนั้นทำการแปลงเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ ด้วยความสัมพันธ์ใน (2.1) จะได้ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ ตามลำดับ โดยมีผลดังนี้

$$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)} = (0.145814920, 0.684349376, -0.579332179)$$
$$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)} = (0.145318347, 0.685297285, -0.578702028)$$
$$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)} = (0.145209884, 0.685382520, -0.578704353)$$
$$\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})} = (0.145637490, 0.685146502, -0.578602942)$$

เมื่อทำการวัดระยะทางระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ พบว่าเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ ใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด รองลงมาคือ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$ ตามลำดับ ดังนำเสนอผลในตารางที่ 4.9

ตารางที่ 4.9 แสดงระยะทางระหว่าง $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ โดยกำหนดมาตรวัดเป็น L_1- norm

	มีระยะทางเป็น
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}\right)$	0.00168044031
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}\right)$	0.00109166890
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}\right)$	**0.00109142669**
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}\right)$	0.00116393923

กรณีที่ 3 ผลการศึกษาการแก้ปัญหา NORTA defective โดยกำหนดมาตรวัดระยะทางเป็น L_∞ - norm

เมื่อกำหนดให้เมทริกซ์ $\mathbf{\Lambda}_{\mathbf{Z}}$ ที่ใช้ทำการศึกษาเป็น 4 มิติ และทำการศึกษาระยะทางระหว่าง $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{u}$ ภายหลังการแปลงเวกเตอร์ ทำการวัดระยะทางระหว่าง $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ พบว่ามีเวกเตอร์ $\mathbf{v}_{\mathbf{Z}}^{(\mathrm{SDP}-L_\infty)}$ ที่ใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด แต่เมื่อทำการแปลงด้วย (2.1) ซึ่งจะได้เวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$ แล้วปรากฏว่าเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ ไม่ได้ใกล้กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ มากที่สุดเมื่อเปรียบเทียบกับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ ดังนำเสนอรายละเอียดโดยยกกรณีตัวอย่างต่อไปนี้

ให้เมทริกซ์สหสัมพันธ์ของเวกเตอร์สุ่ม $\mathbf{X}$ เป็น

$$\mathbf{\Sigma}_{\mathbf{X}} = \begin{bmatrix} 1 & -0.8597619173 & 0.0282032263 & -0.2194718061 \\ -0.8597619173 & 1 & -0.2813064271 & 0.2179772694 \\ 0.0282032263 & -0.2813064271 & 1 & -0.8663391477 \\ -0.2194718061 & 0.2179772694 & -0.8663391477 & 1 \end{bmatrix}$$

และ $\mathbf{v}_{\mathbf{X}} = (-0.8597619, 0.02820322, -0.2947181, -0.2813064, 0.2179773, -0.8663391)$

จากนั้นแปลง $\mathbf{\Sigma}_{\mathbf{X}}$ ด้วย (2.1) จะได้เมทริกซ์ NORTA defective คือ

$$\mathbf{\Lambda_Z} = \begin{bmatrix} 1 & -0.8702377248 & 0.0295332761 & -0.2293248345 \\ -0.8702377248 & 1 & -0.2935193995 & 0.2277700117 \\ 0.0295332761 & -0.2935193995 & 1 & -0.8764340149 \\ -0.2293248345 & 0.2277700117 & -0.8764340149 & 1 \end{bmatrix}$$

และ $\mathbf{u} = (-0.8702377, 0.0295333, -0.2293248, -0.2935194, 0.2277700, -0.8764340)$
ทำการแก้ปัญหาเวกเตอร์ NORTA defective ด้วยวิธี SDP- L_1 , SDP- L_2 ,SDP- L_∞ และวิธี ES ตามลำดับ ได้ผลดังนี้

$$\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_1)} = (-0.8702363, 0.0500710, -0.2293239, -0.2935164, 0.2277707, -0.8764327)$$
$$\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_2)} = (-0.8665216, 0.0335681, -0.2257834, -0.2896568, 0.2311564, -0.8727559)$$
$$\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_\infty)} = (-0.8665225, 0.0032486, -0.2256097, -0.2898042, 0.2314851, -0.8727188)$$
$$\mathbf{v}_\mathbf{Z}^{(\text{ES})} = (-0.8606756, 0.0292088, -0.2268050, -0.2902942, 0.2252673, -0.8668038)$$

เมื่อทำการวัดระยะทางระหว่างเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_1)}$, $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_2)}$, $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\text{ES})}$ กับเวกเตอร์ $\mathbf{u}$ พบว่าเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_2)}$ ใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด รองลงมาคือ $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_\infty)}$, $\mathbf{v}_\mathbf{Z}^{(\text{ES})}$ และ $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_1)}$ ตามลำดับ ดังนำเสนอผลในตารางที่ 4.10

<u>ตารางที่ 4.10</u> แสดงระยะทางระหว่าง $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_1)}$, $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_2)}$, $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\text{ES})}$ กับเวกเตอร์ $\mathbf{u}$ โดยกำหนดมาตรวัดเป็น L_2 - norm

	มีระยะทางเป็น
$d(\mathbf{u}, \mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_1)})$	0.020537737039
$d(\mathbf{u}, \mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_2)})$	0.004034835270
$d(\mathbf{u}, \mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_\infty)})$	0.003715285278
$d(\mathbf{u}, \mathbf{v}_\mathbf{Z}^{(\text{ES})})$	0.009630180154

จากนั้นทำการแปลงเวกเตอร์ $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_1)}$, $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_2)}$, $\mathbf{v}_\mathbf{Z}^{(\text{SDP}-L_\infty)}$ และ $\mathbf{v}_\mathbf{Z}^{(\text{ES})}$ ด้วยความสัมพันธ์ใน (2.1) จะได้ $\mathbf{v}_\mathbf{X}^{(\text{SDP}-L_1)}$, $\mathbf{v}_\mathbf{X}^{(\text{SDP}-L_2)}$, $\mathbf{v}_\mathbf{X}^{(\text{SDP}-L_\infty)}$, $\mathbf{v}_\mathbf{X}^{(\text{ES})}$ ตามลำดับ โดยมีผลดังนี้

$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)} = (-0.8597604, 0.0478193, -0.2194709, -0.2813035, 0.2179779, -0.8663377)$

$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)} = (-0.8558226, 0.0320567, -0.2160679, -0.2775780, 0.2212326, -0.8624335)$

$\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)} = (-0.8558236, 0.0317515, -0.2159009, -0.2777203, 0.2215485, -0.8623942)$

$\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})} = (-0.8496334, 0.0278933, -0.2170498, -0.2781933, 0.2155719, -0.8561216)$

เมื่อทำการวัดระยะทางระหว่างเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ พบว่าเวกเตอร์ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ ใกล้กับเวกเตอร์ $\mathbf{u}$ มากที่สุด รองลงมาคือ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}$ ตามลำดับ ดังนำเสนอผลในตารางที่ 4.11

<u>ตารางที่ 4.11</u> แสดงระยะทางระหว่าง $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}$, $\mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}$ และ $\mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}$ กับเวกเตอร์ $\mathbf{v}_{\mathbf{X}}$ โดยกำหนดมาตรวัดเป็น L_1- norm

	มีระยะทางเป็น
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_1)}\right)$	0.019616065408
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_2)}\right)$	0.003939291678
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{SDP}-L_\infty)}\right)$	0.003944904097
$d\left(\mathbf{v}_{\mathbf{X}}, \mathbf{v}_{\mathbf{X}}^{(\mathrm{ES})}\right)$	0.010217517055

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ส่วนที่ 5 ผลการศึกษาประสิทธิภาพของวิธีการแก้ปัญหา NORTA defective จากทั้ง 4 วิธีการคือ วิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES

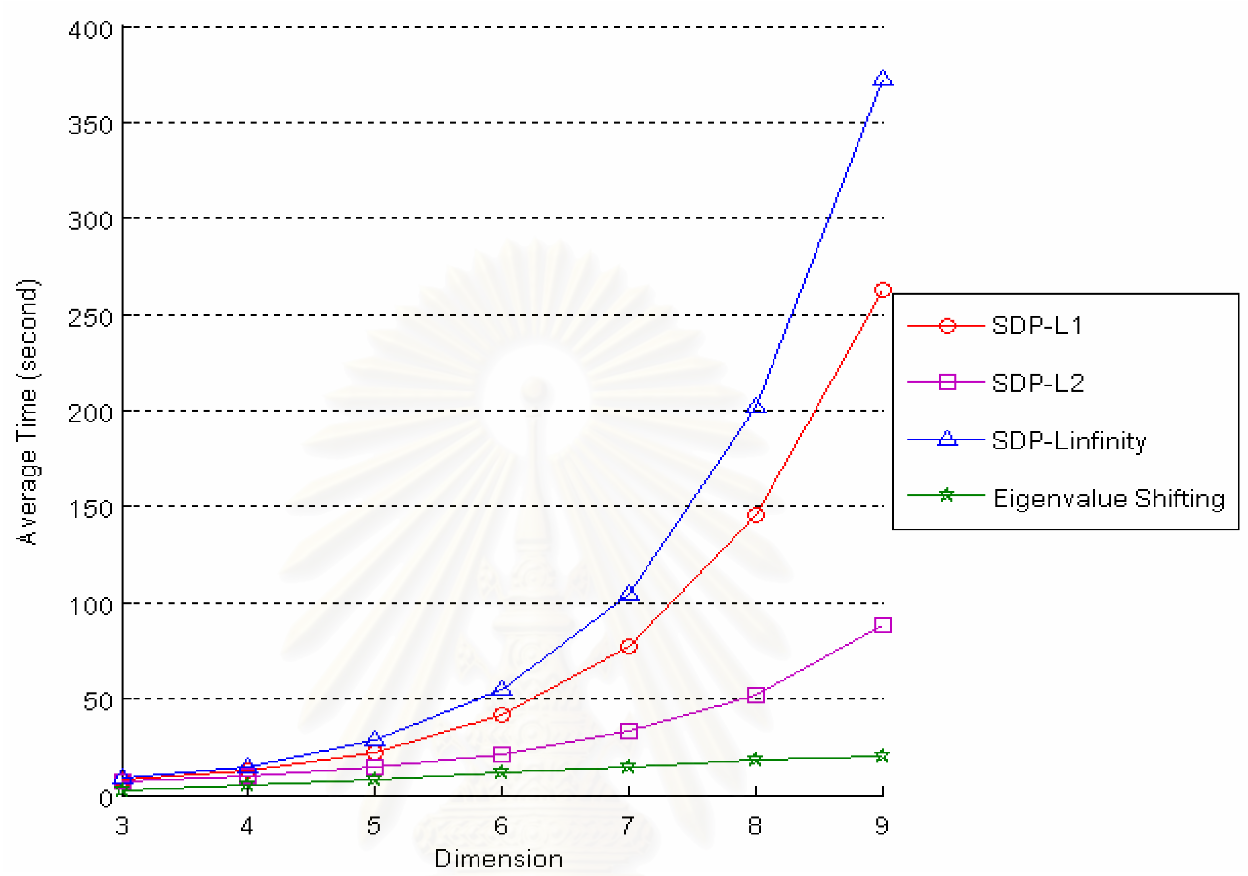
สำหรับการศึกษาประสิทธิภาพ (ความเร็ว) ของวิธีการแก้ปัญหา NORTA defective ได้แก่ วิธี SDP-L_1, SDP-L_2, SDP-L_∞ และวิธี ES โดยกำหนดตัวอย่างเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาในขั้นตอนนี้คือ 1,000 เมทริกซ์ ให้เครื่องคอมพิวเตอร์ที่ใช้ทำการศึกษามีความเร็วของ CPU เท่ากับ 800 เมกะเฮิร์ทซ์ (megahertz) และมีหน่วยความจำหลัก RAM ขนาด 256 เมกะไบต์ (megabyte) ดังแสดงผลในตารางที่ 4.12

ตารางที่ 4.12 แสดงเวลาโดยเฉลี่ยที่ใช้ในการแก้ปัญหา NORTA defective ของแต่ละวิธีการ โดยกำหนดเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาจำนวน 1,000 เมทริกซ์ สำหรับแต่ละมิติ (dimension)

มิติ	**เวลาโดยเฉลี่ยที่ใช้ในการแก้ปัญหา NORTA defective (วินาที)**			
	SDP-L_1	SDP-L_2	SDP-L_∞	ES
3	7.8212464	7.0601520	8.7626000	**2.1030240**
4	12.1875248	10.2547456	14.8012832	**4.9070560**
5	22.4322560	14.5909808	28.2606368	**8.2017936**
6	42.1005376	21.3006288	54.8288400	**11.6467472**
7	77.3512256	33.2578224	104.6504800	**14.6610816**
8	145.6694624	52.2551392	201.9203472	**17.9558192**
9	263.2084752	88.4972528	371.9348160	**20.1389584**

จากผลการศึกษา โดยพิจารณาเวลาโดยเฉลี่ยที่ใช้ในการแก้ปัญหา NORTA defective ของแต่ละวิธีการ พบว่าวิธี ES ใช้เวลาโดยเฉลี่ยน้อยที่สุดในทุกมิติ รองลงมาคือ SDP-L_2, SDP-L_1 และ SDP-L_∞ ใช้เวลาโดยเฉลี่ยเรียงจากน้อยไปมากตามลำดับ ดังแสดงในรูปที่ 4.4

รูปที่ 4.4 แสดงเวลาโดยเฉลี่ย (วินาที) ที่ใช้ในการแก้ปัญหา NORTA defective ของแต่ละวิธีการ โดยกำหนดเมทริกซ์ NORTA defective ที่ใช้ทำการศึกษาจำนวน 1,000 เมทริกซ์



หมายเหตุ ในขั้นนี้ ผู้วิจัยได้กำหนดให้เปอร์เซ็นต์ค่าคลาดเคลื่อนของการประมาณค่าเพื่อทำการคำนวณหาค่า $\lambda_{highest}$ ด้วยวิธี Power Method เท่ากับ $(1\times10^{-7})\%$ สำหรับใช้ในการแก้ปัญหา NORTA defective ด้วยวิธี Eigenvalue Shifting

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

<u>ส่วนที่ 6</u> ผลการจำลองเวกเตอร์สุ่มหลายตัวแปรจากโปรแกรม NORTA ที่พัฒนาขึ้น

ในเบื้องต้นโปรแกรมที่พัฒนาขึ้นในครั้งนี้ มีข้อจำกัดคือสามารถจำลองได้เพียงเวกเตอร์สุ่มหลายตัวแปรที่มีการแจกแจงขอบแบบสม่ำเสมอ (uniform marginal distribution) เท่านั้น สำหรับผลของการจำลองเวกเตอร์สุ่มหลายตัวแปรจากโปรแกรมที่ได้พัฒนาขึ้น ในส่วนนี้ ผู้วิจัยได้ยกตัวอย่างการสร้างเวกเตอร์สุ่มหลายตัวแปรที่มีการแจกแจงขอบแบบสม่ำเสมอ โดยมีเมทริกซ์สหสัมพันธ์ตามที่กำหนด ดังมีรายละเอียดนำเสนอดังนี้

ยกตัวอย่างเช่น ต้องการสร้างเวกเตอร์สุ่ม $\mathbf{X} = (X_1, X_2, X_3)^T$ ซึ่ง $X_i \sim \text{Uniform}(0,1), i = 1,2,3$ จำนวน 40 ชุด และกำหนดให้เมทริกซ์สหสัมพันธ์ของตัวแปรสุ่มเป็น

$$\mathbf{\Sigma_X} = \begin{bmatrix} 1 & 0.3 & 0.4 \\ 0.3 & 1 & 0.5 \\ 0.4 & 0.5 & 1 \end{bmatrix}$$

สำหรับผลลัพธ์ที่ได้จากการจำลองเวกเตอร์สุ่มดังกล่าวข้างต้นโดยใช้โปรแกรม NOTRA ที่ได้พัฒนาขึ้น ซึ่งมีคุณลักษณะเป็นโปรแกรม Add-in โดยใช้งานบนไมโครซอฟต์ เอ็กเซล มีผลการจำลองดังแสดงในตารางที่ 4.13 ซึ่งได้ $X_i \sim \text{Uniform}(0,1), i = 1,2,3$ (ทดสอบการแจกแจงของ X_i ว่ามีการแจกแจงแบบสม่ำเสมอหรือไม่ โดยใช้สถิติทดสอบโคลโมโกรอฟ-สไมร์นอฟ (Kolmogorov-Smirnov Test)) ในขณะที่เมทริกซ์สหสัมพันธ์ของตัวแปรสุ่มที่ได้ที่จากการจำลอง ซึ่งทำการวัดโดยใช้สัมประสิทธิ์สหสัมพันธ์ลำดับที่ของสเปียร์แมน (Spearman Rank Correlation Coefficient) คือ

$$\mathbf{\Sigma_X} = \begin{bmatrix} 1 & 0.356 & 0.462 \\ 0.356 & 1 & 0.512 \\ 0.462 & 0.512 & 1 \end{bmatrix}$$

ผลลัพธ์ที่ได้จากใช้โปรแกรมที่พัฒนาขึ้น พบว่า ได้เวกเตอร์สุ่มหลายตัวแปร ซึ่งมีคุณสมบัติสอดคล้องกับที่กำหนดไว้ในเบื้องต้น

ตารางที่ 4.13 แสดงผลการจำลองเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA จากโปรแกรม NOTRA ที่ได้พัฒนาขึ้น ซึ่งเป็นโปรแกรม Add-in โดยใช้งานบนไมโครซอฟต์ เอ็กเซล

ลำดับที่	ตัวแปรสุ่ม			ลำดับที่	ตัวแปรสุ่ม		
	X_1	X_2	X_3		X_1	X_2	X_3
1	0.895909	0.4871767	0.6319227	21	0.024749	0.7473251	0.3905439
2	0.086321	0.2409905	0.0461101	22	0.194001	0.1286708	0.2127792
3	0.237927	0.0828096	0.0928234	23	0.472680	0.1890967	0.6304841
4	0.575617	0.5378011	0.2224160	24	0.127118	0.2613186	0.5156464
5	0.237843	0.9358914	0.6002674	25	0.632937	0.5177645	0.7647444
6	0.048711	0.6890374	0.4546676	26	0.075759	0.0294038	0.4230466
7	0.949491	0.6555785	0.5034968	27	0.419425	0.0052382	0.0672602
8	0.096471	0.4106428	0.6512717	28	0.804107	0.8910437	0.8987647
9	0.358089	0.5423982	0.0965719	29	0.210034	0.4615712	0.4036434
10	0.584169	0.8391793	0.3220608	30	0.316756	0.0187362	0.1009511
11	0.955972	0.6460590	0.8319773	31	0.900714	0.6457400	0.5182254
12	0.794327	0.7725165	0.4040998	32	0.360690	0.0353851	0.0023524
13	0.262050	0.9098597	0.5806982	33	0.074920	0.3166363	0.3174046
14	0.461725	0.1118856	0.1412359	34	0.917681	0.7492253	0.8622307
15	0.219080	0.2391023	0.0104826	35	0.584083	0.5284120	0.2078923
16	0.609777	0.4756022	0.4576747	36	0.316979	0.6556826	0.5445653
17	0.268653	0.4847291	0.5051981	37	0.858037	0.2690799	0.7593943
18	0.499543	0.5872587	0.9413656	38	0.951110	0.5457686	0.5569359
19	0.726144	0.8561319	0.6258324	39	0.690135	0.6781150	0.7403643
20	0.112797	0.4404548	0.2363551	40	0.295503	0.3987849	0.6102692

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

ในการศึกษาวิจัยครั้งนี้ต้องการพัฒนาเครื่องมือที่ใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปร ด้วยวิธี NORTA และทำการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective ที่อาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA จากสองวิธีการคือ วิธี Semidefinite Programming (SDP) และวิธี Eigenvalue Shifting (ES)

5.1 สรุปผลการวิจัย

5.1.1 โปรแกรมสำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA

ในการพัฒนาเครื่องมือที่ใช้สำหรับสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ผู้วิจัยได้พัฒนาเป็นไฟล์ Add-in ที่สามารถเรียกใช้งานกับไมโครซอฟต์ เอ็กเซล (Microsoft Excel) ทั้งนี้ทำให้โปรแกรมที่พัฒนาขึ้นมีขนาดเล็ก ไม่ซับซ้อน และเนื่องจากมีข้อจำกัดในการนำใช้ค่อนข้างน้อย ทำให้สามารถใช้งานได้ง่ายและสะดวก สำหรับภาษาที่ใช้ในการพัฒนาโปรแกรม NORTA เป็นภาษา C/C++ ในขณะที่วิธีการแก้ปัญหา NORTA defective ที่อาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ผู้วิจัยได้เลือกใช้วิธี Eigenvalue Shifting (ES) โดยที่โปรแกรม NORTA ซึ่งผู้วิจัยได้พัฒนาขึ้นครั้งนี้ ในเบื้องต้นจะสามารถสร้างได้เพียงเวกเตอร์สุ่มหลายตัวแปรที่มีการแจกแจงขอบแบบสม่ำเสมอ (uniform marginal distribution) เท่านั้น

5.1.2 การศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective

สำหรับการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective ที่อาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA จากสองวิธีการคือ 1.วิธี Semidefinite Programming (SDP) ซึ่งประกอบด้วย 3 วิธีย่อยคือ SDP-L_1 , SDP-L_2 และ SDP-L_∞ และ 2.วิธี Eigenvalue Shifting (ES) โดยเกณฑ์ที่ใช้ในพิจารณาคือระยะทางน้อยที่สุดระหว่างเมทริกซ์ NORTA defective กับเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ที่ได้จากแต่ละวิธีการ โดยกำหนดมาตรวัดระยะทางเป็น L_1- norm , L_2- norm และ L_∞- norm

เนื่องจากการวัดระยะทางระหว่างเมทริกซ์สหสัมพันธ์ที่เป็นไปได้ $\mathbf{\Sigma}'_{\mathbf{Z}}$ กับเมทริกซ์ NORTA defective $\mathbf{\Lambda}_{\mathbf{Z}}$ เรารู้ว่า ถ้าใช้มาตรวัด L_1- norm จะทำให้ $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธี SDP-L_1 มีระยะทางน้อยที่สุดเมื่อเทียบกับ $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธีอื่น ในขณะเดียวกัน ถ้าใช้มาตรวัด L_2- norm จะทำให้ $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธี SDP-L_2 มีระยะทางน้อยที่สุดเมื่อเทียบกับ $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธีอื่นเช่นเดียวกัน ถ้าใช้มาตรวัด L_∞- norm จะทำให้ $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธี SDP-L_∞ มีระยะทางน้อยที่สุดเมื่อเทียบกับ $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธีอื่น

ดังนั้นสำหรับการศึกษาวิจัยครั้งนี้จึงสนใจที่ศึกษาระยะเฉลี่ยน้อยสุดระหว่าง $\mathbf{\Sigma}'_{\mathbf{Z}}$ กับ $\mathbf{\Lambda}_{\mathbf{Z}}$ ลำดับที่ 2 สำหรับแต่ละมาตรวัด ซึ่งได้แก่ L_1- norm , L_2- norm และ L_∞- norm ซึ่งผลสรุปการศึกษานำเสนอเป็นลำดับดังต่อไปนี้

1) จากการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยการพิจารณาระยะทางเฉลี่ยของ $d(\mathbf{\Sigma}'_{\mathbf{Z}},\mathbf{\Lambda}_{\mathbf{Z}})$ โดยใช้มาตรวัด L_1- norm พบว่า $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธี SDP-L_2 มีระยะทางเฉลี่ยน้อยที่สุดเป็นลำดับ 2 ในขณะที่วิธี SDP-L_∞ และวิธี ES ให้ระยะทางเฉลี่ยน้อยสุดเป็นลำดับที่ 3 และ 4 ตามลำดับ นั่นคือถ้าเรานิยามความใกล้ด้วยมาตรวัดระยะทาง (distance measure) เป็น L_1- norm วิธี SDP-L_1 และวิธี SDP-L_2 มีความเหมาะสมสำหรับใช้แก้ปัญหาเมทริกซ์ NORTA defective

2) จากการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยการพิจารณาระยะทางเฉลี่ยของ $d(\mathbf{\Sigma}'_{\mathbf{Z}},\mathbf{\Lambda}_{\mathbf{Z}})$ โดยใช้มาตรวัด L_2- norm พบว่า $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธี SDP-L_∞ มีระยะทางเฉลี่ยน้อยที่สุดเป็นลำดับ 2 ในขณะที่วิธี SDP-L_1 และวิธี ES ให้ระยะทางเฉลี่ยน้อยสุดเป็นลำดับที่ 3 และ 4 ตามลำดับ นั่นคือถ้าเรานิยามความใกล้ด้วยมาตรวัดระยะทาง (distance measure) เป็น L_2- norm วิธี SDP-L_2 และวิธี SDP-L_∞ มีความเหมาะสมสำหรับใช้แก้ปัญหาเมทริกซ์ NORTA defective

3) จากการศึกษาความแตกต่างของวิธีการแก้ปัญหา NORTA defective โดยการพิจารณาระยะทางเฉลี่ยของ $d(\mathbf{\Sigma}'_{\mathbf{Z}},\mathbf{\Lambda}_{\mathbf{Z}})$ โดยใช้มาตรวัด L_∞- norm พบว่า $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่ได้จากวิธี SDP-L_2 มีระยะทางเฉลี่ยน้อยที่สุดเป็นลำดับ 2 ในขณะที่วิธี ES และวิธี SDP-L_1 ให้ระยะทางเฉลี่ยน้อยสุดเป็นลำดับที่ 3 และ 4 ตามลำดับ นั่นคือถ้าเรานิยามความใกล้ด้วยมาตรวัดระยะทาง (distance measure) เป็น L_∞- norm วิธี SDP-L_∞ และวิธี SDP-L_2 มีความเหมาะสมสำหรับใช้แก้ปัญหาเมทริกซ์ NORTA defective

สำหรับการแก้ปัญหา NORTA Defective ด้วยวิธี Semidefinite Programming (SDP) ซึ่งประกอบด้วยวิธี SDP-L_1 , SDP-L_2 และ SDP-L_∞ เมื่อพิจารณาระยะทางที่น้อยที่สุดไม่อาจรับประกันได้ว่าทุกเมทริกซ์ $\mathbf{\Sigma}'_{\mathbf{Z}}$ ที่มีระยะทางใกล้กับเมทริกซ์ $\mathbf{\Lambda}_{\mathbf{Z}}$ มากที่สุด หลังจากทำการแปลงไปเป็นเมทริกซ์ $\mathbf{\Sigma}'_{\mathbf{X}}$ แล้วจะทำให้ $\mathbf{\Sigma}'_{\mathbf{X}}$ มีระยะทางใกล้กับเมทริกซ์ $\mathbf{\Sigma}_{\mathbf{X}}$ มากที่สุด ทั้งนี้อาจมีสาเหตุ

จากปัญหาเชิงตัวเลข (numerical problem) ซึ่งเกิดขึ้นในขั้นตอนของการคำนวณจากโปรแกรมที่ใช้ศึกษา แต่โดยส่วนใหญ่แล้วก่อนและหลังการแปลงจะให้ผลลัพธ์ที่สอดคล้องกัน

5.2 การเลือกใช้วิธีการที่เหมาะสมในทางปฏิบัติสำหรับการแก้ปัญหา NORTA defective

สำหรับการศึกษาวิจัยครั้งนี้ เมื่อพิจารณาความใกล้โดยใช้มาตรวัดระยะทาง (distance measure) ทั้ง 3 แบบคือ L_1- norm , L_2- norm และ L_∞- norm ผู้วิจัยมีความเห็นว่าแก้ปัญหา NORTA defective ด้วยวิธี Semidefinite Programming โดยกำหนดฟังก์ชันระยะทางด้วย L_2 หรือ SDP-L_2 มีความเหมาะสมที่สุดสำหรับใช้แก้ปัญหาในกรณีที่ต้องการเมทริกซ์สหสมพันธ์ที่เป็นไปได้ (feasible correlation matrix) ที่ใกล้เคียงกับเมทริกซ์ NORTA defective แต่ในขณะเดียวกันหากพิจารณาถึงจำนวนขั้นตอน(algorithm)และความเร็วของวิธีการแก้ปัญหาแล้ววิธี Eigenvalue Shifting (ES) มีความเหมาะสมมากที่สุดรวมไปถึงการมีขั้นตอนที่ไม่ซับซ้อนและมีความสะดวกต่อการนำไปใช้

5.3 ข้อเสนอแนะ

สำหรับโปรแกรมสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ซึ่งได้พัฒนาขึ้นในงานวิจัยนี้ มีข้อจำกัดคือสร้างได้เพียงเวกเตอร์สุ่มหลายตัวแปรที่มีการแจกแจงขอบแบบสม่ำเสมอ (uniform marginal distribution) เท่านั้น ในขณะที่เมื่อเกิดปัญหา NOTRA defective ซึ่งอาจเกิดขึ้นในกระบวนการสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ผู้วิจัยได้แก้ปัญหาด้วยวิธี Eigenvalue Shifting (ES)

ดังนั้น หากมีผู้สนใจทำวิจัยและต้องพัฒนาโปรแกรมที่เกี่ยวกับการสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA นี้ในโอกาสต่อไป ผู้วิจัยขอเสนอแนะประเด็นน่าสนใจดังนี้

1) การศึกษาวิธีการสร้างเวกเตอร์สุ่มหลายตัวแปรด้วยวิธี NORTA ที่มีการแจกแจงขอบ (marginal distribution) แบบอื่น ๆ เช่น การแจกแจงแบบแกมมา (gamma distribution) การแจกแจงแบบเบตา (beta distribution) หรือการแจกแจงแบบผสม (mixed distribution) เป็นต้น

2) การศึกษาวิธีการแก้ปัญหา NORTA defective ซึ่งถือเป็นปัญหาสำคัญเมื่อมิติ (dimension) ของเวกเตอร์สุ่มที่ต้องการสร้างด้วยวิธี NORTA เพิ่มสูงขึ้น โดยมุ่งเน้นวิธีการที่มีประสิทธิภาพ สะดวกต่อการนำไปใช้ และให้ผลลัพธ์ไม่แตกต่างไปจากวิธีการอื่น ๆ นอกเหนือจากวิธีการที่ผู้วิจัยได้นำเสนอไว้ในงานวิจัยนี้

รายการอ้างอิง

ภาษาไทย

ณัฐศิระ เยาวสุต. 2547. มือใหม่เริ่มเรียน หัดเขียน Macro และ VBA บน Microsoft Excel. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น.

มานพ วราภักดิ์. 2547. การจำลองเบื้องต้น : Introduction to Simulation. กรุงเทพมหานคร : ศูนย์ผลิตตำราเรียน สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

ยุทธนา ลีลาศวัฒนกุล. 2547. เริ่มต้นการเขียนโปรแกรมด้วยภาษา C++. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร : หจก.ไทยเจริญการพิมพ์.

ภาษาอังกฤษ

Cario, M. C. and Nelson, B. L. 1997. Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois.

Dalton, S. 2005. Excel Add-in Development in C/C++ : Applications in Finance. New York : John Wiley & Sons.

Ghosh, S. and Henderson, S. G. 2002b. Properties of the NORTA method in higher dimensions. In Proceedings of the 2002Winter Simulation Conference, E. Yucesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, Eds. IEEE, Piscataway, N.J., 263–269.

Ghosh, S. and Henderson, S. G. 2003. Behaviour of the NORTA method for correlated random vector generation as the dimension increases. ACM Transactions on Modeling and Computer Simulation 13, 276–294.

Ghosh, S. 2004. Dependence in Stochastic Simulation Models, PhD thesis, Cornell University.

Higham, N. J. 1990. Analysis of the Cholesky decomposition of a semi-definite matrix. In Cox, M. G. and Hammarling, S. J. editors, Reliable Numerical Computation, pages 161-185. Oxford University Press.

Kiatsupaibul, S. 2006. Mending flaws of an estimated variance-covariance matrix. Technical Research. Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University.

Kiatsupaibul, S. and La-Ornual, D. 2004. On variance-Covariance matrices for the optimal portfolio model. Operation Research. CRN conference, Chulalongkorn University.

L'Ecuyer, P., Simard, R., Chen, E. J., and Kelton, W. D. 2002. An object-oriented random number package with many long streams and substreams. Operations Research 50, 6, 1073–1075.

Marsaglia, G. 2004. Evaluating the Normal Distribution. Journal of Statistical Software. volume 11, issue 4.

Vandenberghe, L. and Boyd, S. 1996. Semidefinite programming. SIAM Review, 38(1) : 49–95.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก
โปรแกรม NORTA ที่พัฒนาด้วยภาษา C/C++ กรณี 3 ตัวแปร

```
#include <iostream>
#include <iomanip>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "RngStream.h"
using namespace std;
int MatrixChecking(double* LambdaZ,int n);
int Superdiagonalization(double* LambdaZ,int n,int index);
void InverseMatrix(double* LambdaZ,double*InverseLambdaZ,int n);
void SumRow(double* S1,double* S2,int n);
double MaxValue(double* EigenVal,int n);
void MultipM(double* Matrix,double* VectorOld,double* VectorNew,int n);
double FindLowestEigen(double* LambdaZ,int n);
void SigmaUtoLambdaZ(double* SigmaU,double* LambdaZ, int n);
void CholeskyDecomp(double* Co,double* Co2,int n);
void GenerateMVN(double* Cholesky,double* MVN,int n);
double GenerateMVUniform(double Zij);
const int nVar=3,Amount=1;
const double Pi = 3.1415926535897932;

double __stdcall nortaFunction(double a1,double a2,double a3,double* op1,double*
op2,double* op3)
{        static double VectorInput[nVar]={a1,a2,a3};
         static double MSigmaUniform[nVar][nVar]={0};
         static double SigmaZprime[nVar][nVar]={0};
         static double CholeskyLambdaZ[nVar][nVar]={0};
         static double MVNs[Amount][nVar]={0};
```

```
double Uvar[Amount][nVar];
double SmallestEigen;
double* SigmaUniform;
double* LamZ;
double* CholLamZ;
double* MVNvar;
int i,j,k=0;
for(i=0;i<nVar;i++)
        for(j=i;j<nVar;j++)
        {       if(i==j)
                        MSigmaUniform[i][j]=1;
                else
                {MSigmaUniform[i][j]= VectorInput[k]; k++;
                }
                MSigmaUniform[j][i]= MSigmaUniform[i][j];
        }
SigmaUniform = &MSigmaUniform[0][0];
LamZ = &SigmaZprime[0][0];
CholLamZ = &CholeskyLambdaZ[0][0];
MVNvar = &MVNs[0][0];
SigmaUtoLambdaZ(SigmaUniform,LamZ,nVar);
if (MatrixChecking(LamZ,nVar)==0)
{       SmallestEigen=FindLowestEigen(LamZ,nVar);
        for (i=0;i<nVar;i++)
                for(j=0;j<i+1;j++)
                {       if(i!=j)
                        {  SigmaZprime[i][j]= SigmaZprime[i][j]/(1-SmallestEigen);
                           SigmaZprime[j][i]= SigmaZprime[i][j];
                        }
                }
}
```

```
        CholeskyDecomp(LamZ,CholLamZ,nVar);
        GenerateMVN(CholLamZ,MVNvar,nVar);
        for (i=0;i<Amount;i++)
                for (j=0;j<nVar;j++)
                        Uvar[i][j] = GenerateMVUniform(MVNs[i][j]);
        op1[0]=Uvar[0][0];
        op2[0]=Uvar[0][1];
        op3[0]=Uvar[0][2];
        return 1;
}

int MatrixChecking(double* LambdaZ,int n)
{       int i,j,k,flag =1;
        static double LambZ[nVar][nVar]={0};
        double Coef;
        double* LambZmatrix;
        LambZmatrix = &LambZ[0][0];
        for (i=0;i<n;i++)
                for(j=0;j<n;j++)
                        LambZ[i][j] = *(LambdaZ +(i*n+j));
        for(k=0;k<n;k++)
        {       if (Superdiagonalization(LambZmatrix,n,k))
                {       if (LambZ[k][k]>0)
                                for (i=k+1;i<n;i++)
                                {       Coef = LambZ[i][k]/LambZ[k][k];
                                        for (j=k+1;j<n;j++)
                                                LambZ[i][j]-=Coef*LambZ[k][j];
                                        LambZ[i][k]=0;
                                }
                        else
                                ;
```

```
                }
                Else
                {  flag=0;   break;
                }
        } return flag;
}

int Superdiagonalization(double* LambdaZ,int n,int index)
{       int i,j,status=1;
        double resid;
        static double LambZ1[nVar][nVar]={0};
        for (i=0;i<n;i++)
                for(j=0;j<n;j++)
                        LambZ1[i][j] = *(LambdaZ +(i*n+j));
        for (i=index;i<n;i++)
        {       if (LambZ1[i][i]<0)
                {       status=0; break;
                }
                else if (LambZ1[i][i]==0)
                {       for (j=index,resid=0;j<n;j++)
                                resid += fabs(LambZ1[i][j])+fabs(LambZ1[j][i]);
                        if (resid>0)
                        {  status=0;  break;
                        }
                        else status = 1;
                }
                else status = 1;
        }
        return status;
}
```

```
void SigmaUtoLambdaZ(double* SigmaU,double* LambdaZ,int n)
{	int i,j;
	double U[nVar][nVar],V[nVar][nVar];
	double* CoU;
	CoU = &V[0][0];
	for (i=0;i<n;i++)
	{	for (j=0;j<n;j++)
		{	U[i][j]= *(SigmaU + (i*n +j));
			if (i==j)
				V[i][j]=1;
			else
				V[i][j]=2*sin((Pi/6)*U[i][j]);
			*(LambdaZ + (i*n + j)) = *(CoU + (i*n + j));
		}
	}
}

double FindLowestEigen(double* LambdaZ,int n)
{	static double MLambZinverse[nVar][nVar]= {0};
	static double VectorM[nVar]={0};
	static double SubMax[nVar]={0};
	static  double LambdaZ1[nVar][nVar]={0};
	double* SubM;
	double* LambZinverse;
	double* VecM;
	double* LambdaZ0;
	double MaxOld,MaxNew;
	int i,j;
	LambdaZ0=&LambdaZ1[0][0];
	VecM = &VectorM[0];
	SubM = &SubMax[0];
```

```
        LambZinverse=&MLambZinverse[0][0];
        for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                { if (i==j)
                        LambdaZ1[i][j] = 1+n ;
                  else
                        LambdaZ1[i][j] = *(LambdaZ+(i*n +j));
                }
        InverseMatrix(LambdaZ0,LambZinverse,nVar);
        SumRow(LambZinverse,VecM,nVar);
        do
        {       MaxOld = MaxValue(VecM,nVar);
                for (i=0;i<nVar;i++)
                        SubMax[i]=VectorM[i]/MaxOld;
                MultipM(LambZinverse,SubM,VecM,nVar);
                MaxNew = MaxValue(VecM,nVar);
        }
        while (fabs((MaxOld-MaxNew)/MaxOld) > 0.00000000000001);
        return (1/MaxNew)-n;
}

void SumRow(double* S1,double* S2,int n)
{       int i,j;
        static double R0[nVar][nVar]={0};
        double* R1;
        R1 = &R0[0][0];
        for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                {       *(R1 + (i*n + j)) = *(S1+ (i*n + j));       *(S2 + i) += R0[i][j];
                }
}
```

```
void InverseMatrix(double* LambdaZ,double*InverseLambdaZ,int n)
{	static double d[nVar][2*nVar]={0};
	double Temp;
	double* mCorrZ;
	int i,j,k;
	mCorrZ=&d[0][0];
	for(i=0;i<n;i++)
		d[i][i+n]=1;
	for(i=0;i<n;i++)
		for(j=0;j<n;j++)
			d[i][j]=*(LambdaZ+(i*n +j));
	for(j=0;j<n;j++)
	{	Temp = d[j][j];
		for(k=0;k<(n*2);k++)
			d[j][k]=d[j][k]/Temp;
		for(i=0;i<n;i++)
			if (j!=i)
			{	Temp=d[i][j];
				for(k=0;k<(n*2);k++)
					d[i][k]=d[i][k]-(d[j][k]*Temp);
			}
	}
	for(i=0;i<n;i++)
		for(j=0;j<n;j++)
			*(InverseLambdaZ+(i*n +j))=d[i][j+n];
}
```

```
double MaxValue(double* EigenVal,int n)
{       static double x[nVar]={0};
        int i;
        double max;
        for(i=0;i<n;i++)
                *(x + i) = *(EigenVal + i);
        for(i=0;i<n;i++)
        {       if (fabs(x[i]) > fabs(x[i+1]))
                {       max=x[i];  x[i+1]=max;
                }
                else if (fabs(x[i])== fabs(x[i+1]))
                {       if (x[i] > x[i+1])
                        {       max=x[i];   x[i+1]=max;
                        }
                        else
                                max=x[i+1];
                }
                else
                        max=x[i];
        } return max;
}

void MultipM(double* Matrix,double* VectorOld,double* VectorNew,int n)
{       int i,j;    static double sum = 0.0;        j=n;
        for(i=0;i<n;i++)
        {       sum = 0.0;
                for(j=0;j<n;j++)
                        sum += (*(Matrix+(i*n + j)))*(*(VectorOld+j));
                *(VectorNew + i) = sum;
        }
}
```

```
void CholeskyDecomp(double* Co,double* Co2,int n)
{       int i,j,k; double sum;
        static double c[nVar][nVar]={0};
        static double d[nVar][nVar]={0};
        for (i=0;i<n;i++)
        {       for (j=0;j<n;j++)
                        d[i+1][j+1] = *(Co + (i*n +j));
                c[i+1][1]=d[i+1][1];
        }
        for(k=1;k<=n;k++)
        {       for(i=1;i<=k-1;i++)
                {       sum = 0;
                        for(j=1;j<=i-1;j++)
                                sum += c[i][j]*c[k][j];
                        if (c[i][i]==0)
                                c[k][j]=0;
                        else
                                c[k][j] = (d[k][i]- sum)/c[i][i];
                }
                sum = 0;
                for (j=1;j<=k-1;j++)
                        sum += c[k][j]*c[k][j];
                if (d[k][k] - sum < 0.0000000000000001)
                        c[k][k]=0;
                else
                        c[k][k] = sqrt(d[k][k] - sum);
        }
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        *(Co2 +(i*n + j)) = c[i+1][j+1];
}
```

```
void GenerateMVN(double* Cholesky,double* MVN,int n)
{       int i,j,k;
        double v1,v2,w,s,sum,Z[Amount][nVar],X[Amount][nVar],q[nVar][nVar];
        double* p;
        RngStream RngObj;
        p = &X[0][0];
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        q[i][j] = *(Cholesky + (i*n +j));
        for (i=0;i<Amount;i++)
        {       for (j=0;j<nVar;j++)
                {       do
                        {       v1=2 * RngObj.RandU01() -1;
                                v2=2 * RngObj.RandU01() -1;          s =v1*v1+v2*v2;
                        }
                        while (s >= 1); w = sqrt(-2*log(s)/s);
                        Z[i][j]=v1*w;    sum = 0;
                        for (k=0;k<j+1;k++)
                                sum += q[j][k]*Z[i][k];
                        X[i][j]= sum;
                        *(MVN +(i*n +j)) = *(p +(i*n +j));
                }
        }
}

double GenerateMVUniform(double Zij)
{       double s=Zij,b=Zij,q=Zij*Zij,i=1,t=0;
        while(s != t)
                s=(t=s)+(b*=q/(i+=2));
        return (0.5+s*exp(-0.5*q-(0.91893853320467274178)));
}
```

ภาคผนวก ข
โปรแกรม Visual Basic บนไมโครซอฟต์ เอ็กเซล
สำหรับเรียกใช้ฟังก์ชัน NORTA จากไฟล์นามสกุล dll
ที่สร้างขึ้นโดยโปรแกรม Visual C++ 6.0

```
Option Explicit
Declare Function nortaFunction02var Lib "D:\ProgramNORTA\DLLs\nortaSim02var.dll" _
    (ByVal a1 As Double, ByRef op1 As Double, ByRef op2 As Double) As Double
Declare Function nortaFunction03var Lib "D:\ProgramNORTA\DLLs\nortaSim03var.dll" _
    (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, _
     ByRef op1 As Double, ByRef op2 As Double, ByRef op3 As Double) As Double
Declare Function nortaFunction04var Lib "D:\ProgramNORTA\DLLs\nortaSim04var.dll" _
    (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double,_
     ByVal a5 As Double, ByVal a6 As Double, ByRef op1 As Double, _
     ByRef op2 As Double, ByRef op3 As Double, ByRef op4 As Double) As Double
Declare Function nortaFunction05var Lib "D:\ProgramNORTA\DLLs\nortaSim05var.dll" _
   (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double, _
    ByVal a5 As Double, ByVal a6 As Double, ByVal a7 As Double, ByVal a8 As Double, _
    ByVal a9 As Double, ByVal a10 As Double,ByRef op1 As Double, _
    ByRef op2 As Double, ByRef op3 As Double, ByRef op4 As Double, _
    ByRef op5 As Double) As Double
Declare Function nortaFunction06var Lib "D:\ProgramNORTA\DLLs\nortaSim06var.dll" _
   (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double, _
    ByVal a5 As Double,ByVal a6 As Double, ByVal a7 As Double, ByVal a8 As Double, _
    ByVal a9 As Double, ByVal a10 As Double,ByVal a11 As Double, _
    ByVal a12 As Double, ByVal a13 As Double, ByVal a14 As Double, _
    ByVal a15 As Double,ByRef op1 As Double, ByRef op2 As Double, _
    ByRef op3 As Double, ByRef op4 As Double, ByRef op5 As Double, _
    ByRef op6 As Double) As Double
Declare Function nortaFunction07var Lib "D:\ProgramNORTA\DLLs\nortaSim07var.dll" _
   (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double, _
    ByVal a5 As Double,ByVal a6 As Double, ByVal a7 As Double, ByVal a8 As Double, _
```

```
    ByVal a9 As Double, ByVal a10 As Double,ByVal a11 As Double, _
    ByVal a12 As Double, ByVal a13 As Double, ByVal a14 As Double, _
    ByVal a15 As Double, ByVal a16 As Double, ByVal a17 As Double, _
    ByVal a18 As Double, ByVal a19 As Double, ByVal a20 As Double, _
    ByVal a21 As Double, ByRef op1 As Double, ByRef op2 As Double, _
    ByRef op3 As Double, ByRef op4 As Double, ByRef op5 As Double, _
    ByRef op6 As Double, ByRef op7 As Double) As Double
Declare Function nortaFunction08var Lib "D:\ProgramNORTA\DLLs\nortaSim08var.dll" _
   (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double, _
    ByVal a5 As Double, ByVal a6 As Double, ByVal a7 As Double, ByVal a8 As Double, _
    ByVal a9 As Double, ByVal a10 As Double, ByVal a11 As Double, _
    ByVal a12 As Double, ByVal a13 As Double, ByVal a14 As Double, _
    ByVal a15 As Double, ByVal a16 As Double, ByVal a17 As Double, _
    ByVal a18 As Double, ByVal a19 As Double, ByVal a20 As Double, _
    ByVal a21 As Double, ByVal a22 As Double, ByVal a23 As Double, _
    ByVal a24 As Double, ByVal a25 As Double, ByVal a26 As Double, _
    ByVal a27 As Double, ByVal a28 As Double, ByRef op1 As Double, _
    ByRef op2 As Double, ByRef op3 As Double, ByRef op4 As Double, _
    ByRef op5 As Double, ByRef op6 As Double, ByRef op7 As Double, _
    ByRef op8 As Double) As Double
Declare Function nortaFunction09var Lib "D:\ProgramNORTA\DLLs\nortaSim09var.dll" _
   (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double, _
    ByVal a5 As Double,ByVal a6 As Double, ByVal a7 As Double, ByVal a8 As Double, _
    ByVal a9 As Double, ByVal a10 As Double, ByVal a11 As Double, _
    ByVal a12 As Double, ByVal a13 As Double, ByVal a14 As Double, _
    ByVal a15 As Double, ByVal a16 As Double, ByVal a17 As Double, _
    ByVal a18 As Double, ByVal a19 As Double, ByVal a20 As Double, _
    ByVal a21 As Double, ByVal a22 As Double, ByVal a23 As Double, _
    ByVal a24 As Double, ByVal a25 As Double, ByVal a26 As Double, _
    ByVal a27 As Double, ByVal a28 As Double, ByVal a29 As Double, _
    ByVal a30 As Double, ByVal a31 As Double, ByVal a32 As Double, _
```

```
    ByVal a33 As Double, ByVal a34 As Double, ByVal a35 As Double, _
    ByVal a36 As Double, ByRef op1 As Double, ByRef op2 As Double, _
    ByRef op3 As Double, ByRef op4 As Double, ByRef op5 As Double, _
    ByRef op6 As Double, ByRef op7 As Double, ByRef op8 As Double, _
    ByRef op9 As Double) As Double
Declare Function nortaFunction10var Lib "D:\ProgramNORTA\DLLs\nortaSim10var.dll" _
   (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double, _
    ByVal a5 As Double, ByVal a6 As Double, ByVal a7 As Double, ByVal a8 As Double, _
    ByVal a9 As Double, ByVal a10 As Double, ByVal a11 As Double, _
    ByVal a12 As Double, ByVal a13 As Double, ByVal a14 As Double, _
    ByVal a15 As Double, ByVal a16 As Double, ByVal a17 As Double, _
    ByVal a18 As Double, ByVal a19 As Double, ByVal a20 As Double, _
    ByVal a21 As Double, ByVal a22 As Double, ByVal a23 As Double, _
    ByVal a24 As Double, ByVal a25 As Double, ByVal a26 As Double, _
    ByVal a27 As Double, ByVal a28 As Double, ByVal a29 As Double, _
    ByVal a30 As Double, ByVal a31 As Double, ByVal a32 As Double, _
    ByVal a33 As Double, ByVal a34 As Double, ByVal a35 As Double, _
    ByVal a36 As Double, ByVal a37 As Double, ByVal a38 As Double, _
    ByVal a39 As Double, ByVal a40 As Double, ByVal a41 As Double, _
    ByVal a42 As Double, ByVal a43 As Double, ByVal a44 As Double, _
    ByVal a45 As Double, ByRef op1 As Double, ByRef op2 As Double, _
    ByRef op3 As Double, ByRef op4 As Double, ByRef op5 As Double, _
    ByRef op6 As Double, ByRef op7 As Double, ByRef op8 As Double, _
    ByRef op9 As Double, ByRef op10 As Double) As Double
```

```
Sub clearCorr()
  Dim i As Integer, j As Integer, amount As Integer
  amount = Sheets("InputData").Range("f2").Value
  Sheets("InputData").Range(Range("C6"), Range("L15")).Interior.ColorIndex = Null
  For i = 2 To 10
      For j = 1 To i - 1
        Sheets("InputData").Cells(i + 5, j + 2).ClearContents
      Next j
  Next i
  For i = 1 To amount
    Sheets("InputData").Cells(i + 5, 3).Interior.ColorIndex = 36
      For j = 1 To i
        If i = j Then
          Sheets("InputData").Cells(i + 5, j + 2).Interior.ColorIndex = 20
        Else
          Sheets("InputData").Cells(i + 5, j + 2).Interior.ColorIndex = 36
        End If
      Next j
  Next i
End Sub

Sub NTsimulation()
  Dim Met  As Double, nVar As Integer, nSim As Long, nDimv As Long
  Dim i As Integer, j As Integer, k As Integer, status As Integer
  Dim reside As Double, coeff As Double, tempM() As Double, cc As Double
  Dim a() As Double, Vectors() As Double, matrixCorr() As Double
    Sheets("OutputData").Range("A5:P50000").ClearContents
    nSim = Sheets("InputData").Range("K2").Value
    nVar = Sheets("InputData").Range("nVAr").Value
    nDimv = nVar * (nVar - 1) / 2
```

```
ReDim a(0 To nDimv - 1), Vectors(0 To nVar - 1), matrixCorr(nVar, nVar) As Double, _
      tempM(nVar, nVar)
'Check Correlation Matrix by Superdiagonalization Method
For i = 1 To nVar
  For j = i To nVar
    matrixCorr(j, i) = Sheets("InputData").Cells(5 + j, 2 + i).Value
    matrixCorr(i, j) = matrixCorr(j, i)
  Next j
Next i
For k = 1 To nVar
  For i = k To nVar
    If matrixCorr(i, i) < 0 Then
      status = 0
      If status = 0 Then
      MsgBox " The Correlation Matrix that you filled is not positive definite matrix. " _
        & vbCrLf & "" & vbCrLf & "                    You must enter new data ! ", _
        vbOKOnly + vbExclamation, "Invalid Correlation Matrix"
        End If
      Exit Sub
    ElseIf matrixCorr(i, i) = 0 Then
      reside = 0
      For j = k To nVar
        reside = reside + Abs(matrixCorr(i, j)) + Abs(matrixCorr(j, i))
        If reside > 0 Then
          status = 0
          If status = 0 Then
      MsgBox " The Correlation Matrix that you filled is not positive definite matrix. " _
            & vbCrLf & "" & vbCrLf & "                    You must enter new data ! ", _
            vbOKOnly + vbExclamation, "Invalid Correlation Matrix"
            End If
          Exit Sub
```

```
            Else
                status = 1
            End If
        Next j
    Else
        status = 1
    End If
Next i
If status = 1 Then
    For i = k + 1 To nVar
        coeff = matrixCorr(i, k) / matrixCorr(k, k)
        For j = k + 1 To nVar
            matrixCorr(i, j) = matrixCorr(i, j) - coeff * matrixCorr(k, j)
        Next j
        matrixCorr(i, k) = 0
    Next i
End If
Next k

'Start Generate Multivariate Uniform Random Vectors by NORTA Technique
For i = 1 To nDimv
    a(i - 1) = Sheets("InputData").Cells(5 + i, nVar + 13).Value
Next i
For i = 1 To nVar
    Vectors(i - 1) = 0
Next i
    If nVar = 2 Then
        For i = 1 To nSim
            cc = nortaFunction02var(a(0), Vectors(0), Vectors(1))
            Sheets("OutputData").Cells(i + 4, 1) = i
```

```
        For j = 1 To nVar
            Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
        Next j
        Sheets("InputData").Range("i24") = i
    Next i
ElseIf nVar = 3 Then
    For i = 1 To nSim
        cc = nortaFunction03var(a(0), a(1), a(2), Vectors(0), Vectors(1), Vectors(2))
        Sheets("OutputData").Cells(i + 4, 1) = i
        For j = 1 To nVar
            Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
        Next j
        Sheets("InputData").Range("i24") = i
    Next i
ElseIf nVar = 4 Then
    For i = 1 To nSim
        cc = nortaFunction04var(a(0), a(1), a(2), a(3), a(4), a(5), _
                 Vectors(0), Vectors(1), Vectors(2), Vectors(3))
        Sheets("OutputData").Cells(i + 4, 1) = i
        For j = 1 To nVar
            Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
        Next j
        Sheets("InputData").Range("i24") = i
    Next i
ElseIf nVar = 5 Then
    For i = 1 To nSim
        cc = nortaFunction05var(a(0), a(1), a(2), a(3), a(4), a(5), a(6), a(7), a(8), a(9), _
        Vectors(0), Vectors(1), Vectors(2), Vectors(3), Vectors(4))
        Sheets("OutputData").Cells(i + 4, 1) = i
        For j = 1 To nVar
            Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
            Next j
            Sheets("InputData").Range("i24") = i
        Next i
    ElseIf nVar = 6 Then
        For i = 1 To nSim
            cc = nortaFunction06var(a(0), a(1), a(2), a(3), a(4), a(5), a(6), a(7), a(8), a(9), _
            a(10), a(11), a(12), a(13), a(14), _
            Vectors(0), Vectors(1), Vectors(2), Vectors(3), Vectors(4), Vectors(5))
            Sheets("OutputData").Cells(i + 4, 1) = i
            For j = 1 To nVar
                Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
            Next j
            Sheets("InputData").Range("i24") = i
        Next i
    ElseIf nVar = 7 Then
        For i = 1 To nSim
            cc = nortaFunction07var(a(0), a(1), a(2), a(3), a(4), a(5), a(6), a(7), a(8), a(9), _
            a(10), a(11), a(12), a(13), a(14), a(15), a(16), a(17), a(18), a(19), a(20), _
            Vectors(0), Vectors(1), Vectors(2), Vectors(3), Vectors(4), Vectors(5),
Vectors(6))
            Sheets("OutputData").Cells(i + 4, 1) = i
            For j = 1 To nVar
                Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
            Next j
            Sheets("InputData").Range("i24") = i
        Next i
    ElseIf nVar = 8 Then
        For i = 1 To nSim
            cc = nortaFunction08var(a(0), a(1), a(2), a(3), a(4), a(5), a(6), a(7), a(8), a(9), _
            a(10), a(11), a(12), a(13), a(14), a(15), a(16), a(17), a(18), a(19), a(20), _
            a(21), a(22), a(23), a(24), a(25), a(26), a(27), _
```

```
            Vectors(0), Vectors(1), Vectors(2), Vectors(3), Vectors(4), Vectors(5),
Vectors(6), _
            Vectors(7))
            Sheets("OutputData").Cells(i + 4, 1) = i
            For j = 1 To nVar
              Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
            Next j
            Sheets("InputData").Range("i24") = i
          Next i
        ElseIf nVar = 9 Then
          For i = 1 To nSim
            cc = nortaFunction09var(a(0), a(1), a(2), a(3), a(4), a(5), a(6), a(7), a(8), a(9), _
            a(10), a(11), a(12), a(13), a(14), a(15), a(16), a(17), a(18), a(19), a(20), _
            a(21), a(22), a(23), a(24), a(25), a(26), a(27), a(28), a(29), a(30), a(31), _
            a(32), a(33), a(34), a(35), _
            Vectors(0), Vectors(1), Vectors(2), Vectors(3), Vectors(4), Vectors(5),
Vectors(6), _
            Vectors(7), Vectors(8))
            Sheets("OutputData").Cells(i + 4, 1) = i
            For j = 1 To nVar
              Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
            Next j
            Sheets("InputData").Range("i24") = i
          Next i
        Else: nVar = 10
          For i = 1 To nSim
            cc = nortaFunction10var(a(0), a(1), a(2), a(3), a(4), a(5), a(6), a(7), a(8), a(9), _
            a(10), a(11), a(12), a(13), a(14), a(15), a(16), a(17), a(18), a(19), a(20), _
            a(21), a(22), a(23), a(24), a(25), a(26), a(27), a(28), a(29), a(30), a(31), _
            a(32), a(33), a(34), a(35), a(36), a(37), a(38), a(39), a(40), a(41), a(42), _
            a(43), a(44), _
```

```
        Vectors(0), Vectors(1), Vectors(2), Vectors(3), Vectors(4), Vectors(5), Vectors(6), _
        Vectors(7), Vectors(8), Vectors(9))
            Sheets("OutputData").Cells(i + 4, 1) = i
            For j = 1 To nVar
                Sheets("OutputData").Cells(i + 4, j + 1) = Vectors(j - 1)
            Next j
            Sheets("InputData").Range("i24") = i
        Next i
    End If
    Worksheets("OutputData").Activate
End Sub
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

การใช้งานโปรแกรมสำหรับสร้างเวกเตอร์ด้วยวิธี NORTA

ส่วนประกอบของโปรแกรมสำหรับสร้างเวกเตอร์ด้วยวิธี NORTA

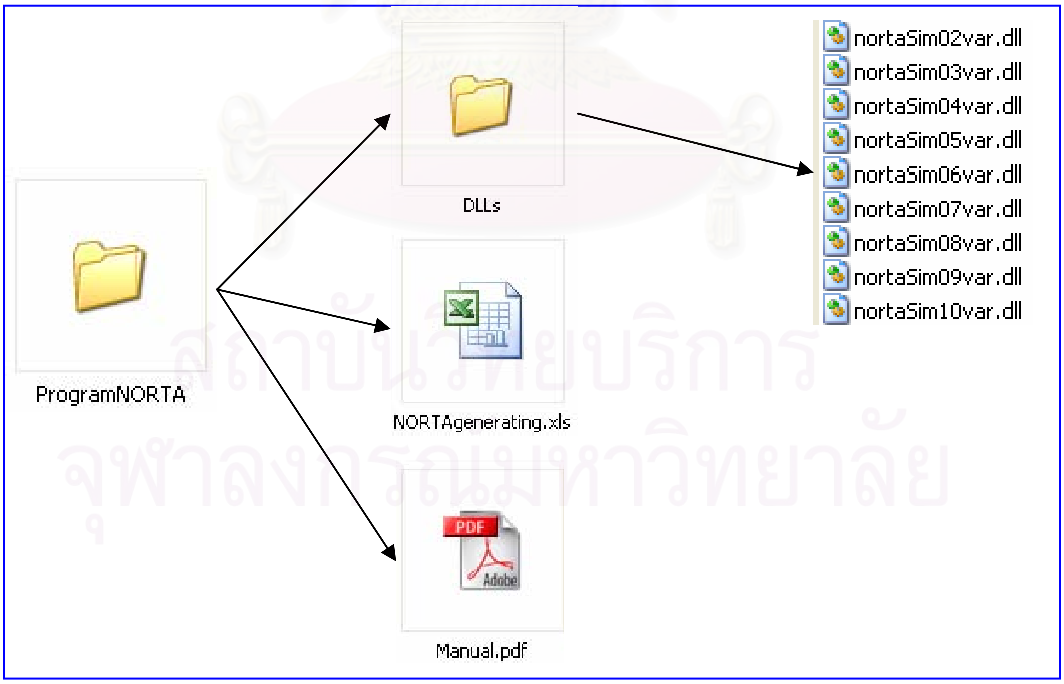
โปรแกรม NORTA ที่พัฒนาขึ้นในครั้งนี้ มีส่วนประกอบที่สำคัญ 3 ส่วนคือ

1) โฟลเดอร์ที่บรรจุไฟล์ที่มีนามสกุลเป็น dll จำนวน 9 ไฟล์ ซึ่งเก็บฟังก์ชันสำหรับสร้างเวกเตอร์สุ่มด้วยวิธี NORTA ขนาด 2–10 มิติ

2) ไฟล์ NORTAgenerating.xls ซึ่งเป็นไฟล์ของไมโครซอฟต์เอ็กเซล โดยที่ไฟล์นี้จะบรรจุโปรแกรมสำหรับสร้างเวกเตอร์สุ่มด้วยวิธี NORTA

3) ไฟล์ Manual.pdf ซึ่งเป็นไฟล์ที่อธิบายรายละเอียด คำแนะนำ ขั้นตอน รวมไปถึงข้อจำกัดในการใช้งานของโปรแกรม NORTA

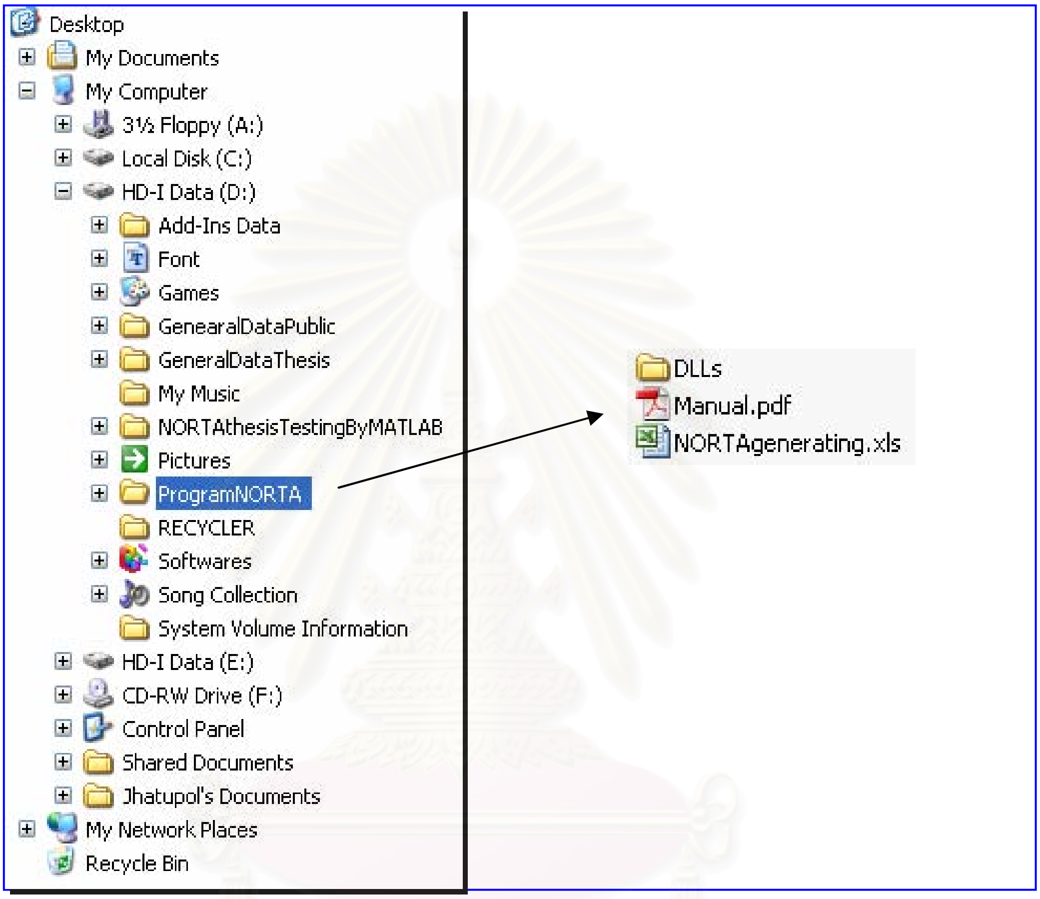
<u>รูปที่ ค1</u> แสดงส่วนประกอบของโปรแกรมสำหรับสร้างเวกเตอร์ด้วยวิธี NORTA ซึ่งถูกเก็บไว้ในโฟลเดอร์ ProgramNORTA



<u>ขั้นตอนการติดตั้งและการใช้งานโปรแกรม</u>

1) ทำการคัดลอก (coppy) โฟลเดอร์ ProgramNORTA แล้วนำไปวางไว้ในไดร์ฟ (drive) ที่ต้องการยกตัวอย่างเช่น คัดลอกโฟลเดอร์ ProgramNORTA วางไว้ในไดร์ฟ D

<u>รูปที่ ค2</u> แสดงตัวอย่างคัดลอกโฟลเดอร์ ProgramNORTA วางไว้ในไดร์ฟ D



2) จากนั้นเปิดโปรแกรมไมโครซอฟต์เอ็กเซล แล้วทำตามขั้นตอนดังแสดงต่อไปนี้

2.1 คลิกเมนู Tools ➢ Macro ➢ Security

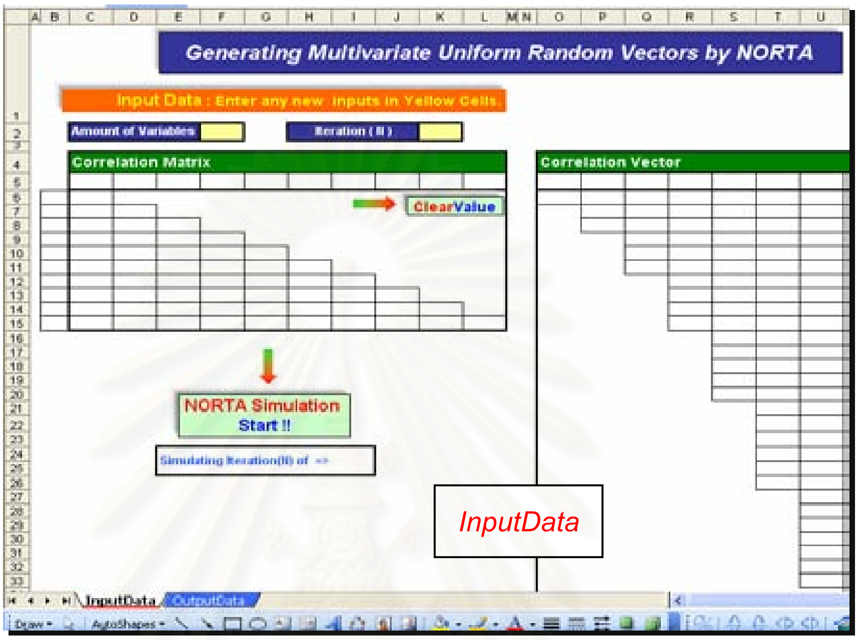
2.2 ที่กรอบโต้ตอบ Security คลิกแท็บ Security Level แล้วเลือก Medium จากนั้นคลิกปุ่ม OK

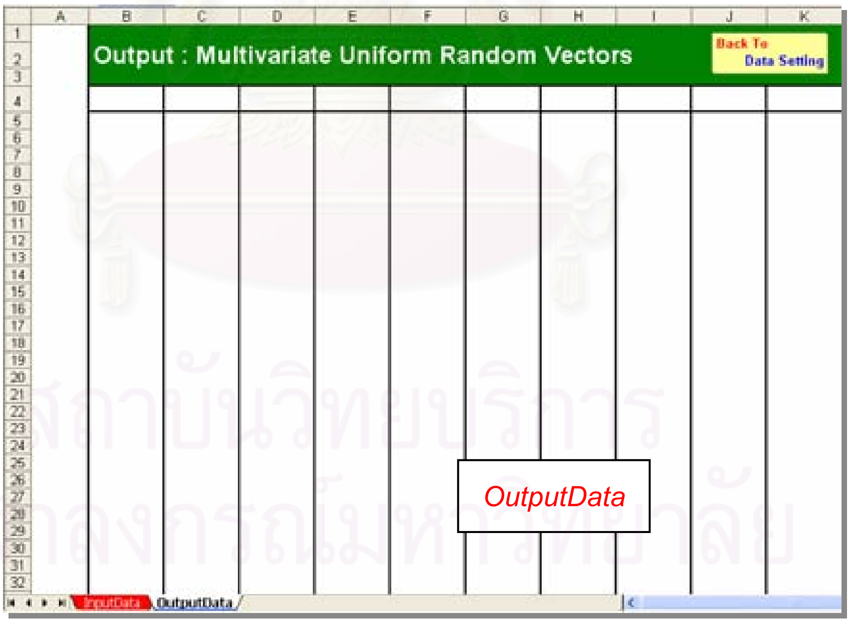
3) เปิดไฟล์ NORTAgenerating.xls โดยคลิกเมนู File ➢ Open … แล้วคลิกที่ช่อง Look in : เลือกไดร์ฟที่เก็บโฟลเดอร์ NORTAprogram ในที่นี้คือไดร์ฟ D จากนั้นดับเบิลคลิกที่โฟลเดอร์ NORTAprogram จะปรากฏ โฟลเดอร์ DLLs และไฟล์ NORTAgenerating.xls เลือกไฟล์ NORTAgenerating.xls แล้วคลิกปุ่ม Open

4) หลังจากเปิดไฟล์ NORTAgenerating.xls จะปรากฏกรอบโต้ตอบ Security Warning ให้คลิกแท็บ Enable Macros

5) ไฟล์ NORTAgenerating.xls ประกอบด้วย 2 worksheet คือ InputData และ OutputData ดังแสดงในรูปที่ ค3

รูปที่ ค3 แสดง worksheet : InputData และ worksheet : OutputData ตามลำดับ





5.1) worksheet : InputData มีวัตถุประสงค์เพื่อให้ผู้ใช้งานระบุคุณลักษณะของเวกเตอร์สุ่มที่ต้องการสร้างโดยแบ่งออกเป็น 3 ส่วนด้วยกันคือ

- ให้ผู้ใช้กำหนดขนาด (dimension) ของเวกเตอร์สุ่มหรือจำนวนตัวแปรสุ่มที่ต้องการสร้าง

- ให้ผู้ใช้กำหนดจำนวนชุด (N) ของเวกเตอร์สุ่มที่ต้องการสร้าง
- ให้ผู้ใช้กำหนดความสัมพันธ์ของตัวแปรสุ่มที่ต้องการในรูปของเมทริกซ์สหสัมพันธ์

5.2) worksheet : OutputData จะแสดงผลการจำลองเวกเตอร์สุ่มที่มีคุณลักษณะตามที่ผู้ใช้ ได้กำหนดไว้ใน worksheet : InputData

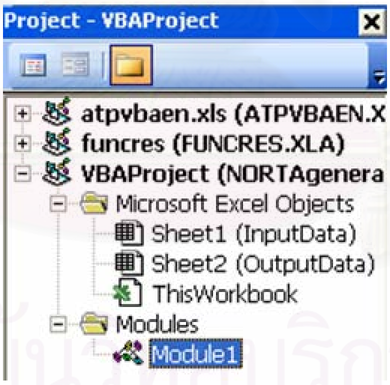
6) ระบุไดร์ฟที่เก็บไฟล์นามสกุล dll เพื่อเรียกใช้ฟังก์ชัน NORTA ซึ่งเก็บไว้ในไฟล์ที่มีนามสกุลดังกล่าว โดยมีขั้นตอนดังต่อไปนี้

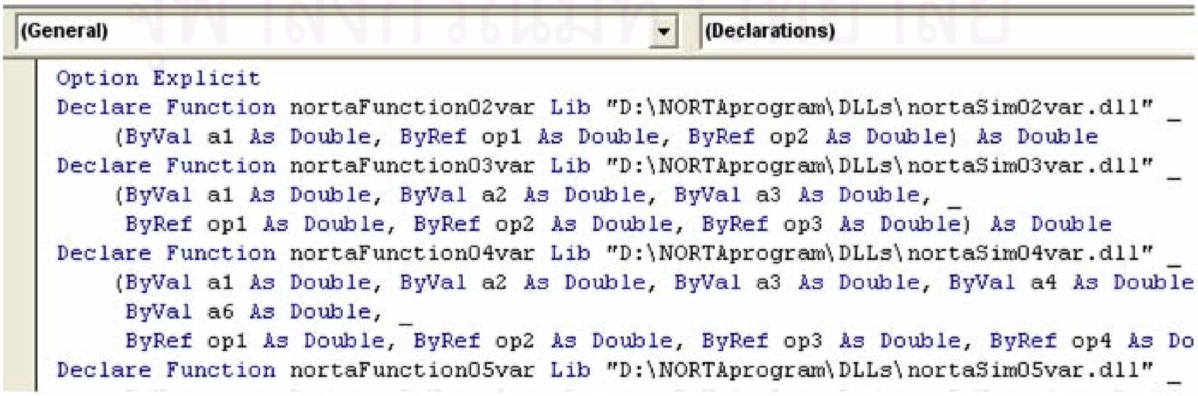
6.1) เปิดหน้าต่าง Visual Basic Editor โดยคลิกเมนู Tools ➢ Macro ➢ Visual Basic Editor หรือกดแป้น Alt + F11

6.2) เมื่อปรากฏหน้าต่าง Visual Basic Editor ให้ทำตามขั้นตอนย่อยต่อไปนี้

6.2.1) ไปที่หน้าต่าง Project – VBAProject เปิดโฟลเดอร์ Modules แล้วดับเบิลคลิกที่ Module1 จะปรากฏรหัส (code) ที่เขียนขึ้นด้วยภาษาวิชวลเบสิก ในหน้าต่างลงรหัส (coding window) ดังแสดงในรูปที่ ค4

รูปที่ ค4 แสดงหน้าต่าง Project – VBAProject และหน้าต่างลงรหัส (coding window) ตามลำดับ






```
Option Explicit
Declare Function nortaFunction02var Lib "D:\NORTAprogram\DLLs\nortaSim02var.dll" _
    (ByVal a1 As Double, ByRef op1 As Double, ByRef op2 As Double) As Double
Declare Function nortaFunction03var Lib "D:\NORTAprogram\DLLs\nortaSim03var.dll" _
    (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, _
     ByRef op1 As Double, ByRef op2 As Double, ByRef op3 As Double) As Double
Declare Function nortaFunction04var Lib "D:\NORTAprogram\DLLs\nortaSim04var.dll" _
    (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double
     ByVal a6 As Double, _
     ByRef op1 As Double, ByRef op2 As Double, ByRef op3 As Double, ByRef op4 As Do
Declare Function nortaFunction05var Lib "D:\NORTAprogram\DLLs\nortaSim05var.dll" _
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.2.2) ในหน้าต่างลงรหัส (Coding Window) ให้ปรับแก้รหัส (Code) โดยแก้ไขเพียงชื่อไดร์ฟที่เก็บโฟล์เดอร์ NORTAprogram ไว้ ดังนั้นในทุกบรรทัดที่ขึ้นต้นด้วย *Declare Function* ซึ่งมีทั้งหมด 9 บรรทัด ให้ปรับแก้รหัส (Code) ดังแสดงในตัวอย่างต่อไปนี้

- ถ้าเก็บโฟลเดอร์ไว้ในไดร์ฟ C ปรับแก้ดังนี้

Declare Function nortaFunction02var Lib “C:\NORTAprogram\DLLs\nortaSim02var.dll”

- ถ้าเก็บโฟลเดอร์ไว้ในไดร์ฟ D ปรับแก้ดังนี้

Declare Function nortaFunction02var Lib “D:\NORTAprogram\DLLs\nortaSim02var.dll”

- ถ้าเก็บโฟลเดอร์ไว้ในไดร์ฟ E ที่มีโฟลเดอร์ชื่อ MyProgram ปรับแก้ดังนี้

Declare Function nortaFunction02var Lib “E:\MyProgram\NORTAprogram\DLLs\nortaSim02var.dll”

โดยในตัวอย่างนี้ได้คัดลอกโฟลเดอร์ NORTAprogram เก็บไว้ที่ไดร์ฟ D จึงทำการปรับแก้รหัส (Code) ทั้ง 9 บรรทัด ดังแสดงในรูปที่ ค5

<u>รูปที่ ค5</u> แสดงการปรับแก้รหัสในหน้าต่างลงรหัส (coding window) ในกรณีที่เก็บ โฟลเดอร์ NORTAprogram ไว้ที่ไดร์ฟ D

```
Declare Function nortaFunction02var Lib "D:\NORTAprogram\DLLs\nortaSim02var.dll" _
    (ByVal a1 As Double, ByRef op1 As Double, ByRef op2 As Double) As Double
Declare Function nortaFunction03var Lib "D:\NORTAprogram\DLLs\nortaSim03var.dll" _
    (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, _
     ByRef op1 As Double, ByRef op2 As Double, ByRef op3 As Double) As Double
Declare Function nortaFunction04var Lib "D:\NORTAprogram\DLLs\nortaSim04var.dll" _
    (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double,
     ByVal a6 As Double, _
     ByRef op1 As Double, ByRef op2 As Double, ByRef op3 As Double, ByRef op4 As Doub
Declare Function nortaFunction05var Lib "D:\NORTAprogram\DLLs\nortaSim05var.dll" _
    (ByVal a1 As Double, ByVal a2 As Double, ByVal a3 As Double, ByVal a4 As Double,
     ByVal a6 As Double, ByVal a7 As Double, ByVal a8 As Double, ByVal a9 As Double,
     ByRef op1 As Double, ByRef op2 As Double, ByRef op3 As Double, ByRef op4 As Doub
Declare Function nortaFunction06var Lib "D:\NORTAprogram\DLLs\nortaSim06var.dll" _
```

6.2.3) ทำการบันทึกแฟ้ม โดยคลิกเมนู File➢Save NORTAgenerating.xls หรือกดแป้น Ctrl+S จากนั้นปิดหน้าต่างลงรหัส

<u>ตัวอย่างการจำลองเวกเตอร์สุ่มแบบสม่ำเสมอ (0,1) ด้วยวิธี NORTA กรณี 5 ตัวแปร</u>

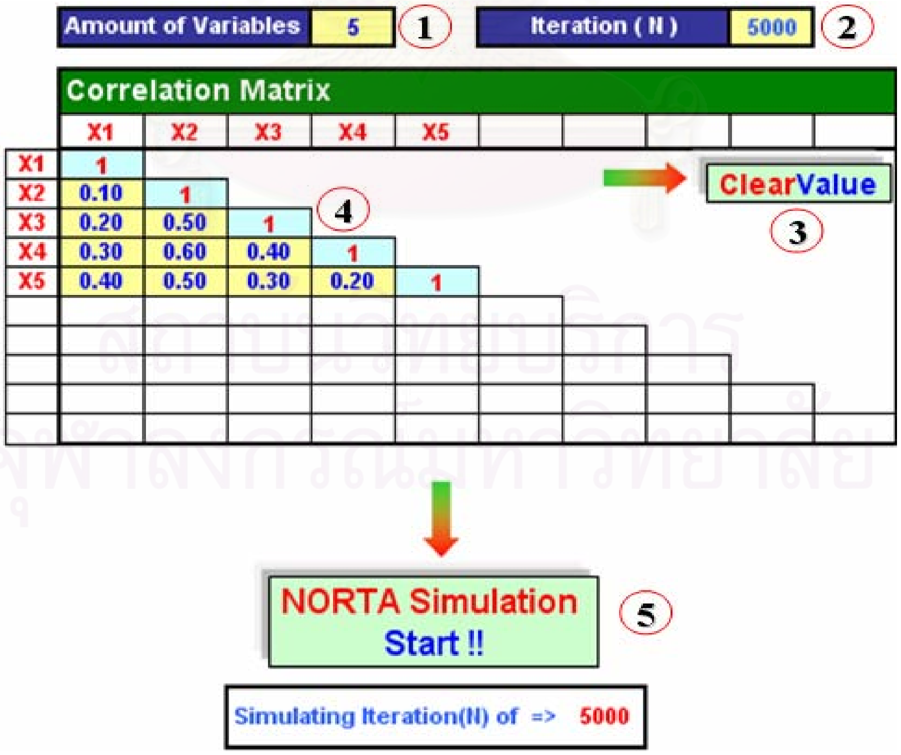
สมมติให้ ผู้ใช้ต้องการเวกเตอร์สุ่มที่มีขนาด 5 มิติ (ประกอบด้วย 5 ตัวแปรสุ่ม) จำนวน 5000 ชุด ซึ่งกำหนดความสัมพันธ์ของตัวแปรสุ่มในรูปของเมทริกซ์สหสัมพันธ์ดังนี้

$$\Sigma_X = \begin{bmatrix} 1 & 0.1 & 0.2 & 0.3 & 0.4 \\ 0.1 & 1 & 0.5 & 0.6 & 0.5 \\ 0.2 & 0.5 & 1 & 0.4 & 0.3 \\ 0.3 & 0.6 & 0.4 & 1 & 0.2 \\ 0.4 & 0.5 & 0.3 & 0.2 & 1 \end{bmatrix}$$

กลับไปที่ worksheet : InputData แล้วทำตามขั้นตอนต่อไปนี้

1) ที่ Amount of Variables ใส่ค่า 5 ในเซลล์สีเหลือง
2) ที่ Iteration (N) ใส่ค่า 5000 ในเซลล์สีเหลือง
3) คลิกที่กล่องข้อความ (Textbox) "ClearValue"
4) ใส่ค่าสหสัมพันธ์ในตาราง Correlation Matrix ในเซลล์สีเหลืองดังแสดงในรูปที่ ค6
5) คลิกที่กล่องข้อความ (Textbox) "NORTA Simulation Start!!"

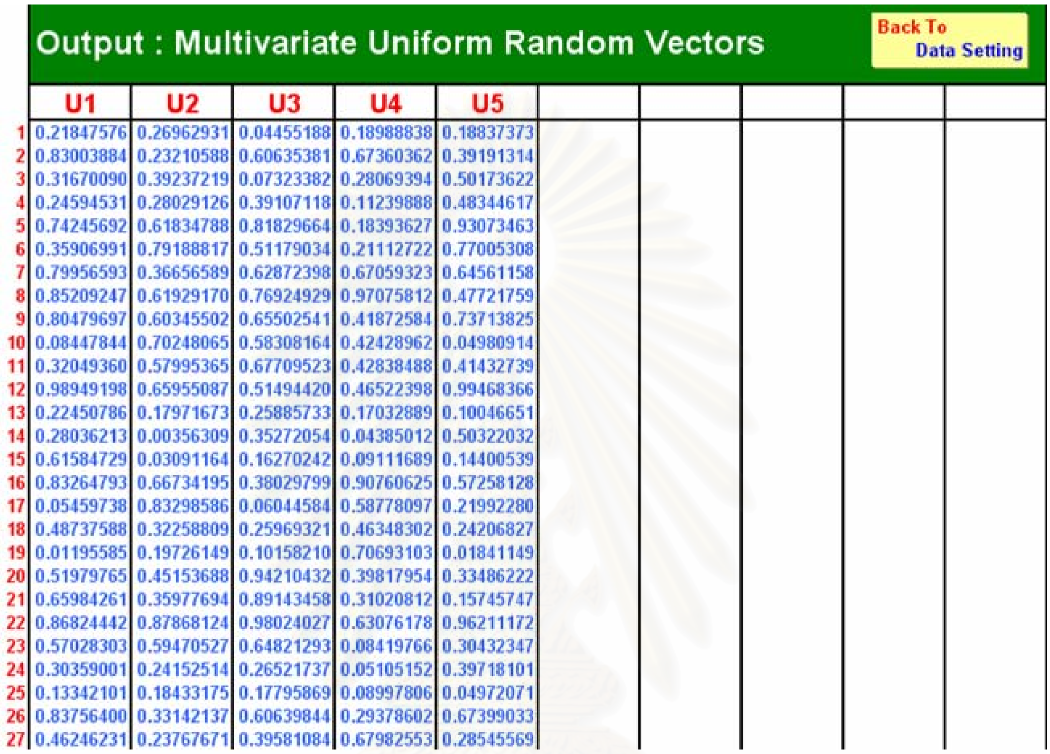
<u>รูปที่ ค6</u> แสดงตัวอย่างการจำลองเวกเตอร์สุ่มด้วยวิธี NORTA กรณี 5 ตัวแปร



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ที่ worksheet : OutputData จะแสดงผลการจำลองเวกเตอร์สุ่มด้วยวิธี NORTA ซึ่งมีคุณลักษณะตามที่ได้กำหนดไว้ใน worksheet : InputData ดังแสดงในรูปที่ ค7

<u>รูปที่ ค7</u> แสดงผลการจำลองเวกเตอร์สุ่มด้วยวิธี NORTA กรณี 5 ตัวแปร



Output : Multivariate Uniform Random Vectors

Back To Data Setting

	U1	U2	U3	U4	U5
1	0.21847576	0.26962931	0.04455188	0.18988838	0.18837373
2	0.83003884	0.23210588	0.60635381	0.67360362	0.39191314
3	0.31670090	0.39237219	0.07323382	0.28069394	0.50173622
4	0.24594531	0.28029126	0.39107118	0.11239888	0.48344617
5	0.74245692	0.61834788	0.81829664	0.18393627	0.93073463
6	0.35906991	0.79188817	0.51179034	0.21112722	0.77005308
7	0.79956593	0.36656589	0.62872398	0.67059323	0.64561158
8	0.85209247	0.61929170	0.76924929	0.97075812	0.47721759
9	0.80479697	0.60345502	0.65502541	0.41872584	0.73713825
10	0.08447844	0.70248065	0.58308164	0.42428962	0.04980914
11	0.32049360	0.57995365	0.67709523	0.42838488	0.41432739
12	0.98949198	0.65955087	0.51494420	0.46522398	0.99468366
13	0.22450786	0.17971673	0.25885733	0.17032889	0.10046651
14	0.28036213	0.00356309	0.35272054	0.04385012	0.50322032
15	0.61584729	0.03091164	0.16270242	0.09111689	0.14400539
16	0.83264793	0.66734195	0.38029799	0.90760625	0.57258128
17	0.05459738	0.83298586	0.06044584	0.58778097	0.21992280
18	0.48737588	0.32258809	0.25969321	0.46348302	0.24206827
19	0.01195585	0.19726149	0.10158210	0.70693103	0.01841149
20	0.51979765	0.45153688	0.94210432	0.39817954	0.33486222
21	0.65984261	0.35977694	0.89143458	0.31020812	0.15745747
22	0.86824442	0.87868124	0.98024027	0.63076178	0.96211172
23	0.57028303	0.59470527	0.64821293	0.08419766	0.30432347
24	0.30359001	0.24152514	0.26521737	0.05105152	0.39718101
25	0.13342101	0.18433175	0.17795869	0.08997806	0.04972071
26	0.83756400	0.33142137	0.60639844	0.29378602	0.67399033
27	0.46246231	0.23767671	0.39581084	0.67982553	0.28545569

<u>ข้อจำกัดและข้อพึงระวังสำหรับการใช้งานโปรแกรม NORTA</u>

1) โปรแกรมสามารถสร้างได้เพียงเวกเตอร์สุ่มแบบสม่ำเสมอ (0,1) เท่านั้น

2) มิติของเวกเตอร์ที่สร้างได้คือ 2-10 มิติเท่านั้น หรือกล่าวอีกนัยหนึ่ง คือ จำนวนตัวแปรสุ่มที่สร้างได้คือ 2 – 10 ตัวแปรเท่านั้น

3) ในการเปิดโปรแกรมครั้งหนึ่ง (NORTAgenerating.xls) เมื่อกำหนดจำนวนตัวแปรสุ่มค่าใดค่าหนึ่ง (2-10) พร้อมกับกำหนดค่าให้กับ Correlation Matrix แล้ว หลังจากทำการรันโปรแกรม จะ *<u>ไม่สามารถ</u>* ปรับเปลี่ยนค่าใน Correlation Matrix ได้อีก *<u>แต่สามารถเพิ่มหรือลดจำนวนของเวกเตอร์สุ่มได้ตามที่ต้องการ</u>*

ยกตัวอย่างเช่น ต้องการสร้างเวกเตอร์สุ่มขนาด 3 มิติ (ตัวแปรสุ่ม 3 ตัวแปร) จำนวน 1000 ชุด โดยกำหนดให้ Correlation Matrix เป็น

$$\mathbf{\Sigma}_{\mathbf{X}} = \begin{bmatrix} 1 & 0.3 & 0.4 \\ 0.3 & 1 & 0.5 \\ 0.4 & 0.5 & 1 \end{bmatrix}$$

โปรแกรมจะจดจำค่าที่อยู่ใน Correlation Matrix สำหรับเวกเตอร์สุ่มขนาด 3 มิตินี้ไว้จนกว่าจะปิดโปรแกรม (ซึ่งถือเป็นข้อด้อยของโปรแกรมที่พัฒนาขึ้นในครั้งนี้)

วิธีแก้ไข หากต้องการปรับเปลี่ยนค่าใน Correlation Matrix ใหม่สำหรับเวกเตอร์ขนาด 3 มิติให้ปิดโปรแกรม (NORTAgenerating.xls) แล้วเปิดขึ้นมาใหม่ จากนั้นให้ปรับเปลี่ยนค่า Correlation Matrix ที่ต้องการแล้วทำการรันโปรแกรมอีกครั้ง

4) Correlation Matrix ที่กำหนดให้กับโปรแกรมจะต้องมีคุณสมบัติเป็นเมทริกซ์กึ่งบวกแน่นอนหรือเมริกซ์บวกแน่นอน (Positive Semidefinite Matrix or Positive Definite Matrix) เท่านั้น ถ้าไม่มีคุณสมบัติดังกล่าว โปรแกรมจะแจ้งเตือนให้ผู้ใช้กำหนดค่าใน Correlation Matrix ใหม่

ยกตัวอย่างเช่น ต้องการสร้างเวกเตอร์สุ่มขนาด 3 มิติ (ตัวแปรสุ่ม 3 ตัวแปร) โดยกำหนดให้ Correlation Matrix เป็น

$$\mathbf{\Sigma}_{\mathbf{X}} = \begin{bmatrix} 1 & 0.9 & 0.6 \\ 0.9 & 1 & 0.1 \\ 0.6 & 0.1 & 1 \end{bmatrix}$$

ซึ่งไม่เป็นเมทริกซ์บวกแน่นอน (non-positive definite matrix) โปรแกรมจะแจ้งเตือนพร้อมกับให้ผู้ใช้กำหนดค่าให้กับ Correlation Matrix ใหม่ โดยจะแสดงกรอบโต้ตอบบังคับให้คลิกเลือก OK ดังแสดงตัวอย่างในรูปที่ ค8

รูปที่ ค8 แสดงกรอบโต้ตอบแจ้งเตือนในกรณี Correlation Matrix ที่ผู้ใช้กำหนดขึ้นไม่มีคุณสมบัติเป็นเมทริกซ์กึ่งบวกแน่นอนหรือเมทริกซ์บวกแน่นอน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายจตุพล ทะน้อม เกิดวันพฤหัสบดีที่ 27 ธันวาคม พ.ศ. 2522 สำเร็จการศึกษาปริญญาวิทยาศาสตร์บัณฑิต (วท.บ.) สาขาวิชาสถิติ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยมหาสารคาม ในปีการศึกษา 2545 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต (สต.ม.) สาขาวิชาสถิติ ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ.2546

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย