รายงานวิจัย ฉบับสมบูรณ์

โครงการวิจัย เรื่อง

ลำดับของสตริงที่มีความยาวแปรเปลี่ยน และการประยุกต์

Variable-length String Sequences and Applications

ชื่อหัวหน้าโครงการผู้รับทุน

รองศาสตราจารย์ ดร. อำพล ธรรมเจริญ

โครงการวิจัยประเภทงบประมาณเงินรายได้

จากเงินอุดหนุนรัฐบาล (งบประมาณแผ่นดิน)

ประจำปีงบประมาณ พ.ศ. ๒๕๕๖

มหาวิทยาลัย บูรพา

รหัสโครงการ ๘๔๙๗๙
สัญญาเลขที่ ๓๔/๒๕๕๖

รายงานวิจัย ฉบับสมบูรณ์

โครงการวิจัย เรื่อง

ลำดับของสตริงที่มีความยาวแปรเปลี่ยน และการประยุกต์
Variable-length String Sequences and Applications

ชื่อหัวหน้าโครงการผู้รับทุนวิจัย

รองศาสตราจารย์ ดร. อำพล ธรรมเจริญ Ph.D.
ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์
มหาวิทยาลัยบูรพา ชลบุรี ๒๐๑๓๑

เดือน สิงหาคม พ.ศ. ๒๕๕๘

คำนำ

การวิจัยเรื่องลำดับของสตริงที่มีความยาวแปรเปลี่ยน และการประยุกต์ (Variable –length String Sequences and Applications) เป็นเรื่องที่คณะผู้วิจัยดำเนินการโดยมีจุดมุ่งหมายเพื่อศึกษาเกี่ยวกับสตริงและลำดับ โดยสร้างนิยาม ทฤษฎีบท ที่เกี่ยวข้องกับสตริง ซึ่งจะประกอบด้วย ลักษณะของสตริง การเปลี่ยนสตริง สตริงรูปแบบพิเศษ ทั้งนี้เพื่อนำมาประยุกต์ใช้ในการสร้างขั้นตอนวิธีที่มีประสิทธิภาพสำหรับการคำนวณที่ต้องมีการสับเปลี่ยนทุกกรณี โดยใช้สตริงรูปแบบพิเศษช่วย

ผลการวิจัย ผู้วิจัยได้สร้างสตริงรูปแบบพิเศษ ชื่อว่า e-k string sequence เป็นลำดับของสตริงที่มีความยาวเป็นกำลังของ k และ e-2 sequence #1 เป็นลำดับของสตริงที่มีความยาวเป็นกำลังของ 2 ลำดับพิเศษนี้มีการประยุกต์ใช้ โดยนำไปใช้เพื่อชี้นำการเปลี่ยน โดยบอกว่าหน่วยที่จะเปลี่ยนคือหน่วยใด เพื่อที่จะนำไปสร้างขั้นตอนวิธีเพื่อแก้ปัญหาต่าง ๆ ดังนี้

1. ปัญหาการหาค่าต่ำสุดของค่าใช้จ่ายในการเปลี่ยนหน่วยโดยให้มีการสับเปลี่ยนทุกกรณีครบถ้วน เช่น ปัญหาการถ่ายรูปชุดแต่งกายสับเปลี่ยนทุกกรณี ปัญหาการทดสอบซอฟต์แวร์ที่มีการสับเปลี่ยนทุกกรณี
2. ปัญหาการคำนวณหานอร์มของเมทริกซ์ ที่มีการแปลงเวกเตอร์ระหว่างปริภูมิที่มีนอร์มไม่เหมือนกัน

ผู้วิจัยได้เขียนผลงานวิจัย เป็นบทความที่พร้อมที่จะพิมพ์ในวารสารวิชาการสาขาคณิตศาสตร์สองบทความ คือ

1. การลำดับที่มีค่าใช้จ่ายน้อยสุดของตัวสับเปลี่ยนทุกกรณี (Minimum cost Sequencing of all Combinations
2. การคำนวณค่าแท้จริงของนอร์มของเมทริกซ์บางแบบ (Exact Solutions of Some Induced Matrix Norms)

ขณะนี้กำลังอยู่ในระหว่างส่งไปพิจารณาลงพิมพ์ในวารสาร

คณะผู้วิจัยขอขอบคุณคณะกรรมการการวิจัยที่อนุมัติงบประมาณวิจัยครั้งนี้ และขอขอบคุณคณะวิทยาศาสตร์ ที่ให้ใช้สถานที่ทำงาน และเปิดโอกาสให้ได้ทำงานจนสำเร็จ คณะผู้วิจัยคาดหวังว่าผลงานวิจัยจะเป็นประโยชน์ในทางวิชาการในด้านการคำนวณ เป็นประโยชน์ในสาขาวิทยาศาสตร์และวิศวกรรมศาสตร์ต่อไป

อำพล ธรรมเจริญ หัวหน้าคณะวิจัย

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับทุนสนับสนุนการวิจัยจากงบประมาณเงินรายได้จากเงินอุดหนุนรัฐบาล(งบประมาณแผ่นดิน) ประจำปีงบประมาณ พ.ศ. ๒๕๕๖ มหาวิทยาลัยบูรพา ผ่านสำนักงานคณะกรรมการวิจัยแห่งชาติ เลขที่สัญญา ๓๔/๒๕๕๖

Acknowledgment

This work was financially supported by the Research Grant of Burapha University through National Research Council of Thailand (Grant no. 34/2556).

รายงานการวิจัย

ชื่อโครงการวิจัย (ภาษาไทย) ลำดับของสตริงที่มีความยาวแปรเปลี่ยนและการประยุกต์

(ภาษาอังกฤษ) " Variable-length String Sequences and Applications"

บทคัดย่อ

การจัดเรียงเพื่อตอบสนองต่อความประสงค์บางประการเป็นเรื่องยาก ในการวิจัยนี้ได้นำเสนอลำดับซึ่งจะใช้เป็นตัวนำทางในการเปลี่ยนค่าเพื่อสร้างเซตลำดับของกรณีการสับเปลี่ยนทุกกรณี เพื่อที่จะทำให้ค่าใช้จ่ายของการเปลี่ยนมีค่าน้อยที่สุด ผลที่ได้เป็นเซตลำดับ ที่สามารถนำไปประยุกต์ใช้กับปัญหาการจัดลำดับที่มีค่าใช้จ่ายน้อยที่สุด เช่น การจัดลำดับการถ่ายรูปแสดงชุดแต่งกายต่าง ๆ การสร้างกรณีทดสอบซอฟต์แวร์แบบทุกกรณี การหาค่าแท้จริงของนอร์มของการเปลี่ยนเชิงเส้นระหว่างปริภูมิที่ต่างกัน

Abstract:

Generating the all combinations can be done in many ways, but it sometimes difficult to arrange that combination to serve some specific purpose. This paper introduces a special sequence that can be used to navigate the transition in the generation of all possible combination, in order to minimize that transition cost. The obtained ordered set can be applied to the minimum sequencing problem, such as the costume-photograph problems, the all combination test case generating problem, and the exact calculation of induced matrix norms problem, which requires a search of all the combinations.

สารบัญเรื่อง

Table of Contents

เรื่อง	หน้า
1. บทนำ	1
2. การดำเนินการวิจัยและผลการวิจัย	4
3. สรุป และอภิปรายผล	10
4. ข้อเสนอแนะ	10
บรรณานุกรม	10
ภาคผนวก	12
ภาคผนวก ก. (การพิสูจน์)	13
ภาคผนวก ข. (ผลผลิต ; Output)	17
ภาคผนวก ค. (ประวัตินักวิจัย)	42
รายงานการเงิน	44

บทสรุปสำหรับผู้บริหาร

(Executive Summary)

ข้าพเจ้า ร.ศ. ดร. อำพล ธรรมเจริญ ได้รับทุนสนับสนุนโครงการวิจัย จากมหาวิทยาลัยบูรพา ประเภทงบประมาณเงินรายได้ จากเงินอุดหนุนรัฐบาล (งบประมาณแผ่นดิน) มหาวิทยาลัยบูรพา

โครงการวิจัย เรื่อง ลำดับของสตริงที่มีความยาวแปรเปลี่ยน และการประยุกต์

ภาษาอังกฤษ Variable-length String Sequences and Applications

รหัสโครงการ ๘๔๙๗๙ สัญญาเลขที่ ๓๔/๒๕๕๖

ได้รับงบประมาณรวมทั้งสิ้น ๑๒๖,๔๐๐ บาท (หนึ่งแสนสองหมื่นหกพันสี่ร้อยบาทถ้วน) ระยะเวลาทำงาน ๑ ปี (ระหว่างวันที่ ๑๗ ตุลาคม พ.ศ. ๒๕๕๖ ถึงวันที่ ๓๐ กันยายน พ.ศ. ๒๕๕๖)

บทคัดย่อ

การสร้างเซตของกรณีการสับเปลี่ยนทุกกรณีสามารถกระทำได้หลายวิธี แต่บางครั้งการจัดเรียงเพื่อตอบสนองต่อความประสงค์บางประการเป็นเรื่องยาก ในการวิจัยนี้ได้นำเสนอลำดับซึ่งจะใช้เป็นตัวนำทางในการเปลี่ยนค่าเพื่อสร้างเซตลำดับของกรณีการสับเปลี่ยนทุกกรณี เพื่อที่จะทำให้ค่าใช้จ่ายของการเปลี่ยนมีค่าน้อยที่สุด

ผลที่ได้เป็นเซตลำดับ ที่สามารถนำไปประยุกต์ใช้กับปัญหาการจัดลำดับที่มีค่าใช้จ่ายน้อยที่สุด เช่น การจัดลำดับการถ่ายรูปแสดงชุดแต่งกายต่าง ๆ การสร้างกรณีทดสอบซอฟต์แวร์แบบทุกกรณี การหาค่าแท้จริงของนอร์มของการเปลี่ยนเชิงเส้นระหว่างปริภูมิที่ต่างกัน

Output/Outcome

ผลที่ได้จากการวิจัย ได้สร้างวิธีการจัดลำดับของการดำเนินการเพื่อให้การดำเนินการนั้นมีค่าใช้จ่ายต่ำสุด ซึ่งสามารถนำไปประยุกต์ใช้กับเรื่องต่าง ๆ ที่มีการจัดลำดับทุกกรณี เช่น การถ่ายรูปชุดแต่งกายทุก ๆ แบบ การทดสอบซอฟท์แวร์ ในสาขาวิศวกรรมซอฟท์แวร์ การหาค่าแท้จริงของนอร์มของการเปลี่ยนเชิงเส้นระหว่างปริภูมิที่ต่างกัน

ผลงานที่ได้ ได้เขียนเป็นบทความวิจัย สองบทความ คือ

1. Minimum cost Sequencing of all Combinations
2. Exact Calculation of Some Induced Matrix Norms

ซึ่งจะได้นำไปเผยแพร่ในวารสารวิชาการ ต่อไป

ข้อเสนอแนะ

วิธีการสร้างลำดับสตริงที่ผู้วิจัยสร้างขึ้น อาจนำไปใช้สำหรับแก้ปัญหาอื่น ๆ ได้ ปัญหาที่จะกล่าวต่อไปนี้เป็นปัญหาที่ยากขึ้น ผู้วิจัยเสนอไว้เพื่อเป็นปัญหาสำหรับหารวิจัยต่อไป

ปัญหาข้อ 1. ปัญหาการถ่ายรูปหมู่เครื่องแต่งกาย

มีคน m คน แต่ละคนมีชุดแต่งกายคนละหลายชุด จะถ่ายรูปหมู่ m คน โดยที่แต่ละคนเปลี่ยนชุด โดยการถ่ายทุก ๆ แบบของการเรียงสับเปลี่ยน ถ้าค่าเปลี่ยนชุดของแต่ละชุด ของแต่ละคนไม่เท่ากัน เราจะมีวิธีจัดเรียงสับเปลี่ยนอย่างไร เพื่อให้มีค่าใช้จ่ายในการเปลี่ยนชุดต่ำสุด (ปัญหาข้อนี้ต่างจากปัญหาที่ผู้วิจัยแก้ปัญหาที่ว่า ในปัญหาที่ผู้วิจัยทำ ราคาการเปลี่ยนชุดของคนหนึ่ง เท่ากันทุกชุด)

ปัญหาข้อ 2. ปัญหาการเฉลี่ยภาระเท่ากัน

ลักษณะเหมือนกับปัญหาการถ่ายรูปหมู่เครื่องแต่งกาย แต่ว่า ในกรณีนี้ ทุกคนมีจำนวนชุดเท่ากัน และราคาการเปลี่ยนชุดของทุกคนทุกชุดเท่ากันหมด เมื่อเป็นดังนี้แต่ละคนก็ไม่อยากเปลี่ยนชุดมากเกินจำเป็น ดังนั้น จะจัดอย่างไรให้มีจำนวนการเปลี่ยนชุดของแต่ละคนเท่ากันหรือใกล้เคียงกัน คือ ต่างกันไม่เกิน 1 ครั้ง

ปัญหาทำนองนี้มีปรากฏเป็นปัญหาการจัดให้ทำงาน หรือปัญหาการเปลี่ยนสภาพ เมื่อภาระหน้าที่คือชิ้นส่วนอีเล็กโทรนิกส์ ซึ่งจะต้องจัดให้มีภาระพอ ๆ กันเพื่อไม่ให้บางชิ้นส่วนมีภาระเกินกำลัง

1. บทนำ

1.1 ความสำคัญ และที่มาของปัญหาที่ทำการวิจัย

การออกแบบขั้นตอนวิธีในการคำนวณที่ดีจะต้องมุ่งเน้นถึงความสำเร็จของวิธีการและการประหยัดแรงงาน ปัญหาการหาค่าเหมาะสุดบางปัญหาอาจจำเป็นต้องมีการคำนวณทุกกรณีเพื่อแก้ปัญหาดังกล่าว จะยกตัวอย่างที่จะทำให้เข้าใจปัญหาง่าย ๆ ดังนี้

สามี ภรรยา และลูกสาว 1 คน ต่างคนต่างก็มีชุดประจำตัวคนละ 2 ชุด จะถ่ายรูปหมู่สามคน โดยแต่งกายชุดต่าง ๆ ที่ตนมีอยู่ โดยเปลี่ยนชุดและสับเปลี่ยนให้ครบทุกชุด ถ้าใช้ 0 และ 1 แทนชุดสองชุดที่แต่ละคนมี และเขียนแสดงชุดของสามี ภรรยา และลูกสาว เรียงตามลำดับ จะได้การเรียงสับเปลี่ยนดังนี้

000 001 010 011 100 101 110 111 (1)

รูปถ่ายทั้งหมดจะมี 8 รูป ครบทุกกรณีของชุดทั้งหมด

การเรียงลำดับกรณีต่าง ๆ ใน (1) เป็นการเรียงตามลำดับของจำนวนเลขฐานสอง จากน้อยไปมาก ถ้าพิจารณาชุดการแต่งกาย เริ่มต้นที่ชุดใดชุดหนึ่ง คือ 000 แล้วต่อมาลูกสาวเปลี่ยนชุด แต่คนอื่น ๆ คงเดิม จะเป็น 001 ต่อมาเป็น 010 หมายถึงภรรยา และลูกสาวเปลี่ยนชุด แต่สามีคงเดิม จาก 011 เป็น 100 แสดงว่าเปลี่ยนชุดทั้งสามคน ถ้าการเปลี่ยนชุดมีค่าใช้จ่าย จะเห็นว่า การเรียงแบบ (1) ไม่เหมาะ เพราะตั้งแต่ 000 ถึง 111 จะมีการเปลี่ยนชุดทั้งหมด 10 ครั้ง ถ้าเรียงลำดับเป็น

000 001 011 010 110 111 101 100 (2)

ทั้งหมดมี 8 กรณีเช่นเดียวกัน แต่ถ้าพิจารณาการเปลี่ยนชุด จะเห็นว่า มีการเปลี่ยนทั้งหมด 7 ครั้ง สามีเปลี่ยน 1 ครั้ง ภรรยาเปลี่ยน 2 ครั้ง และลูกสาวเปลี่ยน 4 ครั้ง ถ้าการเปลี่ยนชุดของลูกสาวถูกที่สุด ของภรรยาแพงขึ้น และของสามีแพงที่สุด ลำดับใน (2) ก็จะเป็นการลำดับที่มีค่าใช้จ่ายต่ำสุด แต่ถ้าการเปลี่ยนชุดของภรรยาแพงที่สุด ของลูกสาวรองลงมา และของสามีถูกที่สุด ลำดับใน (2) ก็ยังคงใช้ได้ ถ้าเขียนลำดับของการเปลี่ยนชุดเป็น ภรรยา ลูกสาว และสามี ตามลำดับ

ในกรณีทั่วไป ถ้ามีการถ่ายรูปหมู่ m คน แต่ละคนมีจำนวนชุดแต่งกาย n_i ชุด จำนวนการเรียงสับเปลี่ยนทั้งหมดจะเท่ากับ $n_1 \times x_2 \times \ldots \times n_m$ วิธี การจัดลำดับของกรณีต่าง ๆ เพื่อให้ค่าใช้จ่ายต่ำสุดจะทำได้อย่างไร อันนี้เป็นปัญหาที่การวิจัยนี้จะดำเนินการ

ปัญหาที่คล้ายกันคือ ปัญหาการสร้างกรณีทดสอบซอฟต์แวร์ ก่อนที่ผู้ผลิตซอฟต์แวร์จะนำไปใช้งานต้องมีการทดสอบว่าจะมีข้อผิดพลาดหรือไม่ การทดสอบกระทำโดยการใส่ข้อมูลแล้วดูว่าคอมพิวเตอร์จะทำงานตรงตามต้องการหรือไม่ [9, 16] หน่วยที่จะใส่ข้อมูลมี m หน่วย และข้อมูลทดสอบในแต่ละหน่วยมีหลายข้อมูล การทดสอบทุกกรณีก็จะต้องมีการสับเปลี่ยนเช่นเดียวกันกับเรื่องการถ่ายรูปหมู่เครื่องแต่งกาย จะจัดลำดับกรณีทดสอบอย่างไรเพื่อให้การเปลี่ยนข้อมูลมีค่าใช้จ่ายน้อยที่สุด

อีกตัวอย่างหนึ่งคือ การคำนวณหาค่านอร์มของการแปลงเชิงเส้น T จากปริภูมิ R_n ซึ่งมีนอร์ม $\|\cdot\|_\infty$ ไปยังปริภูมิ R_m ซึ่งมีนอร์ม $\|\cdot\|_1$ กำหนดโดย

$$T(\mathbf{x}) = A\mathbf{x}$$

เมื่อ A เป็นเมทริกซ์ $m \times n$ และ **x** เป็นเวกเตอร์ นอร์มของการแปลงกำหนดโดย

$$\|T\|_{\infty 1} = \sup_{\|x\|_\infty = 1} \|T(x)\|_1 = \sup_{\|x\|_\infty = 1} \|A\mathbf{x}\|_1 = \max_{\|x\|_\infty = 1} \{ \sum_{i=1}^{m} | \sum_{j=1}^{n} a_{ij} x_j | \}$$

[5] การหาค่านอร์มดังกล่าวไม่มีวิธีอื่นนอกจากจะต้องคำนวณทุกกรณีของ $x_j = 1$, หรือ -1, $j = 1, 2, \ldots, n$ ซึ่งจะมีจำนวนกรณีทั้งหมด 2^n กรณี การเปลี่ยนค่าครั้งหนึ่ง(กรณีหนึ่ง)มีการคำนวณ(คูณและบวก) $n \times m$ ครั้ง จะเห็นว่าจำนวนการคำนวณทั้งหมดมีค่าเป็นฟังก์ชันเอ็กโพเนนเชียล คูณกับ $n \times m$ แต่ถ้ามีการจัดการที่ดี ทำให้การคำนวณแต่ละครั้งกระทำในคอลัมน์เดียว คือ m ครั้ง ก็จะประหยัดแรงงานได้มาก

วิธีการจัดลำดับของตัวเรียงสับเปลี่ยนอาจกระทำได้โดยการใช้ลำดับของตัวเลขเป็นตัวชี้นำการบอกตำแหน่งที่จะเปลี่ยน ดังเช่น (2) เริ่มจาก 000 ต่อมาเปลี่ยนตำแหน่งขวาสุดจาก 0 เป็น 1 และต่อมาเปลี่ยนตำแหน่งที่สองจากขวา จาก 0 เป็น 1 ลำดับที่จะใช้ชี้นำการเปลี่ยนตำแหน่งจะเป็นดังนี้

1, 2, 1, 3, 1, 2, 1

(ตำแหน่งที่นับจากทางขวา) ลำดับนี้อาจเรียกว่า สตริง (String) (คือลำดับจำกัดของจำนวนหรือตัวอักษร เช่น 13101 หรือ acx12a)

ลำดับของขั้นตอนวิธีเป็นเรื่องสำคัญที่จะทำให้ขั้นตอนวิธีมีประสิทธิภาพ ดังนั้น การศึกษาเรื่องของสตริงจะทำให้ทราบลำดับของขั้นตอนวิธี โดยจะสร้างสตริงที่มีลักษณะพิเศษที่จะใช้นำขั้นตอนวิธีเพื่อทำให้การดำเนินการได้ผลสำเร็จ และประหยัดแรงงาน

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างนิยาม ทฤษฎีบท ที่เกี่ยวข้องกับสตริง ซึ่งจะประกอบด้วยลักษณะของสตริง การเปลี่ยนสตริง สตริงรูปแบบพิเศษ

2. เพื่อเสาะหาขั้นตอนวิธีที่มีประสิทธิภาพสำหรับการคำนวณที่ต้องมีการสลับเปลี่ยนทุกกรณี โดยใช้สตริงรูปแบบพิเศษช่วย

1.3 ทฤษฎี สมมุติฐาน หรือกรอบแนวความคิด

สตริง (String) คือลำดับจำกัดของจำนวนหรือตัวอักษร สตริงสามารถนำมาต่อกัน (Concatenation) ลำดับที่ประกอบด้วยสตริงต่อกันเราเรียกว่าลำดับสตริง (String sequence) พีเทรงโคและซิเมา [12] ได้กำหนดวิธีการสร้างลำดับ และได้ทำการตรวจสอบลำดับที่สร้างขึ้นโดยใช้ลำดับย่อยที่แยกไว้ ดีไลน์และบิมบอท [3] ได้สร้างตัวแบบภาษาโดยใช้ลำดับแปรเปลี่ยนความยาว เกรกอรีและรูสิโนวิช [7] ได้ทำการจับคู่

สตริงที่ไม่เกี่ยวข้องกับความยาวของสตริง โคฮาวี [10] ได้กล่าวถึงการแยกลำดับที่มีความยาวแปรปลี่ยนและการประยุกต์ใช้ในเรื่องการทดสอบซอฟต์แวร์ ที่กล่าวมานี้ไม่มีเรื่องที่เกี่ยวกับการคำนวณ แต่เป็นเรื่องที่เกี่ยวข้องกันกับเรื่องสตริง

ปัญหาพหุนามคือปัญหาที่มีจำนวนครั้งของการดำเนินการเป็นพหุนามของขนาดของปัญหา [8, 13, 14] ปัญหาเอ็กซ์โพเนนเชียลมีนิยามเช่นเดียวกัน และเป็นปัญหาแบบ NP-Complete ซึ่งใช้แรงงานสูงในการแก้ปัญหา ในงานวิจัยนี้จะหาเทคนิคที่จะลดแรงงานที่ใช้ในการแก้ปัญหาดังกล่าว

มีทฤษฎีพื้นฐานเกี่ยวกับเรื่องการจัดเรียงสับเปลี่ยน ดังต่อไปนี้ [2, 8, 15]

1. ถ้ามีการกระทำหลายอย่างเกิดขึ้นต่อเนื่องกันไปหรือกระทำพร้อมกัน การกระทำอย่างแรกมี n_1 วิธี และแต่ละวิธีของการกระทำอย่างแรกนั้น เกิดการกระทำอย่างที่สอง n_2 วิธี และแต่ละวิธีของการกระทำอย่างที่สอง จะกระทำอย่างที่สาม ได้ n_3 วิธี ดังนี้ต่อไปเรื่อย ๆ จนถึงการกระทำอย่างที่ k ซึ่งทำได้ n_k วิธี เพราะฉะนั้น การกระทำ k อย่าง ต่อเนื่องกันนี้ จะมีวิธีกระทำได้ถึง $n_1 n_2 \ldots n_k$ วิธี

2. มีการกระทำ k อย่าง ซึ่งกระทำต่อเนื่อง หรือกระทำพร้อม ๆ กัน ถ้าแต่ละอย่างมีจำนวนวิธีที่จะทำได้ n วิธี ดังนั้นการกระทำทั้ง k อย่างต่อเนื่องกัน หรือพร้อม ๆ กัน จะมีวิธีกระทำได้ n^k วิธี

3. ถ้ามีการกระทำ k อย่าง แต่ละอย่างเราเลือกกระทำได้ $n_1, n_2, \ldots, n_k$ วิธี ตามลำดับ ดังนั้น เรามีวิธีเลือกกระทำเพียงอย่างใดอย่างหนึ่งได้ $n_1 + n_2 + \ldots + n_k$ วิธี

1.4 ผลงานวิจัยที่เกี่ยวข้อง (literature review)

ปัญหาที่ผู้วิจัยเสนอนี้เป็นเรื่องใหม่ งานวิจัยที่ตรง หรือใกล้เคียงมีน้อย ซึ่งจะเป็นเรื่องของการหาค่าเหมาะสุดของการจัดเรียง (Combinatorial Optimization) ฟาง และคณะ [6] ได้จัดลำดับ DNA ในชีววิทยา อำพล [4] ได้เสนอวิธีการนำวิธีจัดการแข่งขันแบบพบกันหมดมาใช้ในการจัดกรณีทดสอบซอฟต์แวร์ และ อำพล [1] ได้เสนอวิธีการเขียนลำดับสองมิติ ซึ่งจะสามารถนำมาใช้ในงานวิจัยนี้

ในการประยุกต์ใช้ในเรื่องนอร์มของเมทริกซ์ที่นิยามจากตัวดำเนินการ (Induced matrix norms) รอน [14] ได้กล่าวถึงความยากของการคำนวณนอร์ม ดราคาคิส [5] ได้หาสูตรของการคำนวณนอร์มของเมทริกซ์ที่เป็นตัวดำเนินการจากปริภูมิหนึ่งไปยังปริภูมิหนึ่งซึ่งไม่เหมือนกัน แต่มีบางสูตรคลาดเคลื่อน ซึ่งผู้วิจัยจะเสนอสูตรแก้ไข

1.5 ประโยชน์ที่คาดว่าจะได้รับ เป็นการสร้างองค์ความรู้ใหม่ในสาขาวิชาคณิตศาสตร์ประยุกต์ มีประโยชน์ในการพัฒนาความรู้ทางคณิตศาสตร์ของประเทศ ผลการวิจัยนี้สามารถนำไปใช้ได้อย่างกว้างขวาง คือ 1. ความรู้ทฤษฎีบทต่าง ๆ ที่สร้างขึ้นเพื่อยืนยันความถูกต้องของวิธีการจะเป็นความรู้พื้นฐานสำหรับเรื่องอื่น ๆ และเป็นเนื้อหาทางคณิตศาสตร์ระดับสูง ที่จะเป็นองค์ความรู้ ใช้สอนในระดับปริญญาโท

(เอก) ได้ 2. ระเบียบวิธีการคำนวณที่สร้างขึ้นจะสามารถนำมาใช้สร้างซอฟต์แวร์คำนวณสำหรับปัญหาทางวิทยาศาสตร์และวิศวกรรมศาสตร์ที่เกี่ยวข้อง

ผลกระทบที่เกิดขึ้น หากผลงานเป็นไปดังคาดหมาย จะเป็นองค์ความรู้ใหม่ที่มีประโยชน์ใน การพัฒนาเทคโนโลยี มีผลกระทบต่อสังคมไทย ในแง่ของการสร้างความเชื่อมั่นในภูมิปัญญาของคนไทย สร้างความมั่นคงด้านวิชาการ สร้างศักยภาพในการเรียนการสอนระดับสูง (ปริญญาโท เอก) สร้างชื่อเสียงให้มหาวิทยาลัย และของประเทศโดยรวม

2. ผลการวิจัย

ได้ผลการวิจัยดังต่อไปนี้

1. ได้สร้างนิยาม ทฤษฎีบทเกี่ยวกับ ลำดับของสตริง (Sequence of string)
2. ได้ลำดับของสตริงชื่อว่า e-2 string sequence #1 ซึ่งสามารถนำไปประยุกต์ใช้เป็น "ลำดับนำ" (Leading sequence) สำหรับการกระทำที่มีการเรียงสับเปลี่ยนทุกกรณี ซึ่งโดยการดำเนินการตามลำดับนำนี้จะทำให้ลดปริมาณของการดำเนินการลงได้มากกว่าครึ่งของการดำเนินการแบบเดิม
3. ได้นำไปประยุกต์ใช้กับการคำนวณค่านอร์มของเมทริกซ์ โดยผู้วิจัยได้พิสูจน์สูตรการคำนวณนอร์มของเมทริกซ์บางแบบที่ยังไม่มีใครเสนอสูตรไว้ก่อน

มีรายละเอียดดังต่อไปนี้

2.1 กำหนดปัญหา

1. ปัญหาการถ่ายรูปเครื่องแต่งกาย

ถ้ามีคน m คน แต่ละคนมีจำนวนชุดแต่งกาย n_i ชุด ค่าใช้จ่ายในการเปลี่ยนชุดของคนที่ i เท่ากับ c_i ต้องการถ่ายรูปหมู่ m คน โดยแต่ละคนเปลี่ยนชุดให้ได้ครบทุกชุด และถ่ายรูปชุดที่แตกต่างกันทุกแบบ จะหาวิธีการการจัดลำดับของกรณีต่าง ๆ เพื่อให้ค่าใช้จ่ายต่ำสุด

2. ปัญหาสร้างกรณีทดสอบซอฟต์แวร์

มีหน่วยป้อนข้อมูล m หน่วย แต่ละหน่วยมีข้อมูล n ข้อมูล ค่าใช้จ่ายในการใส่ข้อมูล(เวลาที่ใช้) ของแต่ละหน่วย เป็น ทำการทดสอบทุกกรณีของข้อมูลที่แตกต่างกัน จะหาวิธีการจัดลำดับกรณีทดสอบต่าง ๆ เพื่อให้มีค่าใช้จ่ายต่ำสุด

3. ปัญหาการคำนวณหานอร์มของเมทริกซ์ 3 แบบ [5]

1. $$\|A\|_{\infty 1} = \max_{\|\mathbf{x}\|_\infty = 1} \|A\mathbf{x}\|_1 = \max_{x_j \in S} \{ \sum_{i=1}^{m} | \sum_{j=1}^{n} a_{ij} x_j | \} \text{ where } S = \{-1, 1\}.$$

2. $$\|A\|_{21} = \max_{\|\mathbf{x}\|_2 = 1} \|A\mathbf{x}\|_1 = \max_{s_i \in S} \{ (\sum_{j=1}^{n} (\sum_{i=1}^{m} s_i a_{ij})^2)^{1/2} \}, \text{ where } S = \{-1, 1\}.$$

3. $$\|A\|_{\infty 2} = \max_{\|\mathbf{x}\|_\infty = 1} \|A\mathbf{x}\|_2 = \max_{x_j \in S} [\sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j)^2]^{1/2} \text{ where } S = \{-1, 1\}.$$

2.2 กรณีจัดเรียงแบบ e-ni (e-ni combination)

นิยาม 1. ให้ m เป็นจำนวนเต็มบวก สำหรับ i = 1, 2, . . . , m ให้ Si เป็นเซตของแบบต่าง ๆ ของสิ่งของ ni ชิ้น ที่แตกต่างกัน ลำดับของ m ชิ้น เมื่อหน่วยที่ i เป็นสิ่งของจาก Si เราเรียกว่า ตัวจัดเรียงความยาว m เซตของตัวจัดเรียงความยาว m เขียนว่า S(ni, m) ถ้า ni = n ทุก ๆ i จะเรียกว่า ตัวจัดเรียงแบบ e-n (e-n combination) เซตของตัวจัดเรียงแบบ e-n เขียนว่า S(n, m)

จากนิยาม เราได้สมบัติดังนี้

สมบัติ 1. เซต S(ni, m) มีจำนวนสมาชิกเท่ากับ $n_1n_2 \ldots n_m$ สมาชิก และ เซต S(n, m)มีจำนวนสมาชิกเท่ากับ n^m สมาชิก

สมบัติ 2. เซต S(n, m) สมสัณฐานกับเซตของจำนวน m หลัก ในฐาน n

การแปลง 1 บิต (1-bit transformation)

นิยาม 2. ให้ C เป็นเซตย่อยของ S(ni, m) ให้ T เป็นการแปลงบน C ถ้า T(c) กับ c ต่างกันเพียงตำแหน่งเดียวแล้ว เราเรียกการแปลง T ว่าเป็นการแปลง 1 บิต

ทฤษฎีบท 1. ให้ S(ni, m) เป็นเซตของตัวเรียงสับเปลี่ยน เซตนี้จะกำหนดได้โดยการแปลง 1 บิตจากสมาชิกตัวใดตัวหนึ่งของเซต

ทฤษฎีนี้มีความสำคัญ จะแสดงการพิสูจน์ภายหลัง

ลำดับของสตริง

ถ้า S เป็นเซตของสิ่งของ สตริงจากเซต S คือลำดับจำกัดของสิ่งของจาก S ลำดับสตริงคือลำดับที่ประกอบด้วยพจน์ที่เป็นสตริง ตัวอย่างเช่น ลำดับ

$\langle 1 \rangle, \langle 1, 2 \rangle, \langle 1, 2, 3 \rangle, \langle 1, 2, 3, 4 \rangle, \ldots$

ถ้าถอดตัวแบ่งสตริงออก จะเป็นลำดับ

1, 1, 2, 1, 2, 3, 1, 2, 3, 4, . . .

นิยาม 3. เราเรียกชื่อลำดับสตริงดังต่อไปนี้

ลำดับสตริงแบบ c-k: เป็นลำดับเมื่อสตริงมีความยาวเป็นค่าคงตัว k

ลำดับสตริงแบบ n-k: เป็นลำดับเมื่อสตริงมีความยาวแปรตามค่าผลคูณของ k

ลำดับสตริงแบบ e-k: เป็นลำดับเมื่อสตริงมีความยาวแปรตามค่ากำลังของ k

ทฤษฎีบท 2. ให้ S เป็นลำดับสตริง ที่สตริง s_i มีความยาว k_i , i = 1, 2, 3, . . . ให้ $m_0 = 0$ และ $m_i = \sum_{q=1}^{i} k_q$ ให้ u_n เป็นพจน์ที่ n ของลำดับ ให้ a_{ij} เป็นสมาชิกตัวที่ j ของสตริง s_i แล้ว สำหรับ i = 1, 2, 3, . . . จะได้ว่า

$n = m_{i-1} + j$ เมื่อ $1 \le j \le k_i$ ก็ต่อเมื่อ $u_n = a_{ij}$

การพิสูจน์กระทำได้ไม่ยาก จะไม่แสดงในที่นี้

ลำดับสตริงแบบ e-2 (#1) (e-2 string sequence #1)

นิยาม 4. ลำดับสตริงแบบ e-2 (#1) ขนาด m เขียนสัญลักษณ์ว่า S_{m2} เป็นลำดับในแบบ

$$s_1, s_2, s_3, \ldots, s_m$$

เมื่อ m เป็นจำนวนเต็มบวก และ s_i เป็นสตริงกำหนดโดยกฎ

$$s_1 = \langle 1 \rangle \qquad a_1 = \langle 1 \rangle$$

และสำหรับ i = 2, 3, . . . , m;

$$s_i = \langle i, a_{i-1} \rangle \qquad a_i = \langle a_{i-1}, s_i \rangle$$

โดยที่สัญลักษณ์ $\langle a_{i-1}, s_i \rangle$ เป็นลำดับที่ประกอบด้วยสมาชิกของสตริง a_{i-1} กับ s_i

ลำดับแบบ e-2 (#1) เขียนสัญลักษณ์ S_{m2} เช่นเดียวกัน คือลำดับที่ประกอบด้วยสมาชิกของลำดับสตริง e-2 (#1)

ตัวอย่างของลำดับ e-2 (#1) เมื่อ m = 4

1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1

ซึ่งได้จากสมาชิกของลำดับสตริง e-2 (#1)

$$\langle 1 \rangle, \langle 2, 1 \rangle, \langle 3, 1, 2, 1 \rangle, \langle 4, 1, 2, 1, 3, 1, 2, 1 \rangle$$

ทฤษฎีบท 3. ลำดับสตริง e-2 (#1) ขนาด m มีสมบัติดังนี้

1. สตริง s_i มีความยาว 2^{i-1} สตริง a_i มีความยาว $2^i - 1$
2. จำนวนพจน์ของลำดับ S_{m2} คือ $2^m - 1$
3. ถ้า u เป็นค่าของพจน์ที่ j ของ s_i แล้ว u จะเป็นพจน์ที่ j ของ s_{i+1} ถ้า $j \neq 1$ และเป็นพจน์ที่ $2^{i-1} + j$ ของ s_{i+1}
4. ถ้า u_n เป็นพจน์ที่ n ของลำดับ และ $n = 2^{k-1}(2q - 1)$ เมื่อ k และ q เป็นจำนวนเต็มบวกแล้ว $u_n = k$

การพิสูจน์จะใส่ไว้ในภาคผนวก

นิยาม 5. ให้ $c_1 \in S(2, m)$ และ S_m เป็นลำดับสตริงแบบ e-2 (#1) ขนาด m T_k เป็นการแปลง 1 บิต $k \in S_m$ $N = 2^m$ แล้ว ลำดับ $\langle c_1, c_2, \ldots, c_N \rangle$ เมื่อ $c_{i+1} = T_{u_i}(c_i)$, $i = 1, 2, \ldots, N-1$ เรียกว่าตัวเรียงสับเปลี่ยนแบบ e-2 (#1) ที่ก่อกำเนิดโดย c_1

ตัวอย่าง ถ้า m = 4 เซตตัวเรียงสับเปลี่ยน #1 ก่อกำเนิดโดย 0000 คือ

{0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000}

บทแทรก: เซตตัวเรียงสับเปลี่ยน e=2 (#1) เป็นเซตอันดับของตัวเรียงสับเปลี่ยน ที่มีจำนวนการเปลี่ยนหน่วยน้อยที่สุด

ลำดับสตริงแบบ e-ni (#1) (e-ni string sequence #1)

นิยาม 6. ให้ $m, n_1, n_2, \ldots, n_m$ เป็นจำนวนเต็มบวก ลำดับสตริงแบบ e-ni (#1) ขนาด m เขียนสัญลักษณ์ว่า S_{mni} เป็นลำดับในแบบ

$$s_1, s_2, s_3, \ldots, s_m$$

เมื่อ s_i เป็นสตริงกำหนดโดยกฎ

$$s_1 = \langle 1_1, 1_2, \ldots, 1_{n_i-1} \rangle \qquad a_1 = \langle 1_1, 1_2, \ldots, 1_{n_i-1} \rangle$$

และสำหรับ $i = 2, 3, \ldots, m$;

$$s_i = \langle \langle i, a_{i-1} \rangle_1, \ldots, \langle i, a_{i-1} \rangle_{n_i-1} \rangle \qquad a_i = \langle a_{i-1}, s_i \rangle$$

โดยที่สัญลักษณ์ $\langle a_{i-1}, s_i \rangle$ เป็นลำดับที่ประกอบด้วยสมาชิกของสตริง a_{i-1} กับ s_i

ลำดับแบบ e-ni (#1) เขียนสัญลักษณ์ S_{mni} เช่นเดียวกัน คือลำดับที่ประกอบด้วยสมาชิกของลำดับสตริง e-ni (#1)

ทฤษฎีบท 4. ลำดับสตริง e-ni (#1) ขนาด m มีสมบัติดังนี้

4. สตริง s_i มีความยาว $n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i - 1)2^{i-1}$
5. จำนวนพจน์ของลำดับ S_{mni} คือ $n_1 \times n_2 \times \ldots \times n_m - 1$

นิยาม 7. ให้ $c_1 \in S(ni, m)$ และ S_{mni} เป็นลำดับสตริงแบบ e-ni (#1) ขนาด m T_{u_i} เป็นการแปลง 1 บิต $u_i \in S_{mni}$ $N = n_1 \times n_2 \times \ldots \times n_m$ แล้ว ลำดับ $\langle c_1, c_2, \ldots, c_N \rangle$ เมื่อ $c_{i+1} = T_{u_i}(c_i)$, $i = 1, 2, \ldots, N-1$ เรียกว่าตัวเรียงสับเปลี่ยนแบบ e-ni (#1) ที่ก่อกำเนิดโดย c_1

ตัวอย่าง: ให้ $m = 3, S_1 = \{1, 2, 3, 4\}, S_2 = \{1, 2, 3\}, S_3 = \{1, 2\}, c_1 = 111$

จะได้ว่า $N = 4 \times 3 \times 2 = 24$.

S_{mni} = 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1

และได้ เซตของตัวเรียงสับเปลี่ยนแบบ e-ni (#1) e-ni combination #1

{111, 112, 113, 114, 124, 121, 122, 123, 133, 134, 131, 132, 232, 233, 234, 231, 211, 212, 213, 214, 224, 221, 222, 223}

2.3 การประยุกต์:

ผลเฉลยของปัญหาการลำดับของการจัดเรียงสับเปลี่ยนทุกกรณี ที่มีค่าใช้จ่ายต่ำสุด

พิจารณาปัญหาการถ่ายรูปหมู่เครื่องแต่งกาย และปัญหาการจัดกรณีทดสอบซอฟต์แวร์ทุก ๆ กรณี จะหาค่าต่ำสุดของค่าใช้จ่าย $\sum_{k=1}^{N}\left(\sum_{i=1}^{m}\eta_{ki}c_i\right)_k$ เหนือเซตของเซตอันดับ S(ni, m) ถ้า $c_1 \le c_2 \le \ldots \le c_m$ แล้ว ปัญหามีผลเฉลยคือ เซตอันดับตัวเรียงสับเปลี่ยนแบบ e-ni (#1) $\langle e_1, e_2, \ldots, e_N\rangle$ ซึ่งได้กล่าวไว้แล้ว ค่าใช้จ่ายทั้งหมดคือ ค่าใช้จ่ายเริ่มต้นของ e_1 และค่าใช้จ่ายในการเปลี่ยนแต่ละหน่วย ซึ่งหาได้โดยการนับว่าแต่ละหน่วยมีการเปลี่ยนกี่ครั้ง (โดยนับจากลำดับแบบ e-ni (#1)) แล้วคูณด้วยค่าใช้จ่ายประจำหน่วย แล้วรวมกันทั้งหมด

จำนวนครั้งของการเปลี่ยนคือ

$$q_1 = (n_1-1) \times n_2 \times \ldots \times n_m$$

$$q_2 = (n_2-1) \times n_3 \times \ldots \times n_m$$

- - - - -

$$q_m = n_m - 1$$

ดังนั้นค่าใช้จ่ายทั้งหมดคือ $C + \sum_{i=1}^{m} q_i c_i$, เมื่อC เป็นค่าใช้จ่ายเริ่มต้น

หมายเหตุ จำนวนครั้งของการเปลี่ยนทั้งหมดคือ $\sum_{i=1}^{m} q_i c_i = n_1 \times n_2 \times \ldots \times n_m - 1$.

การคำนวณค่านอร์มของเมทริกซ์

จะแสดงการคำนวณค่านอร์ม $\|A\|_{\infty 1} = \max_{x_j \in S} \{\sum_{i=1}^{m} |\sum_{j=1}^{n} a_{ij}x_j|\}$ เมื่อ S = {-1, 1} ส่วนอีกสองค่ามีวิธีคำนวณทำนองเดียวกัน

ในการหาค่า $\sum_{i=1}^{m}|\sum_{j=1}^{n}a_{ij}x_j|$ เนื่องจากจำนวนเป็นค่าสัมบูรณ์ ดังนั้นเราคำนวณเพียงครึ่งเดียวของกรณีทั้งหมด ก็จะได้ผลตามต้องการ กล่าวคือ เราคำนวณโดยใช้ค่าตามรูปแลลของตัวเรียงสับเปลี่ยน e-2 (#1) เพียง 2^{n-1} ตัว แทนที่จะเป็น 2^n ตัว

ในการหาค่า เราหาผลรวมของทั้งหมดในกรณีแรกก่อน แล้วเมื่อเปลี่ยนเป็นกรณีอื่น เราลบด้วยค่าประจำหน่วยเดิมที่เปลี่ยนออก แล้วบวกด้วยค่าประจำหน่วยใหม่ กระทำดังนี้จนครบทุกกรณี ทำให้ประหยัดแรงงานการคำนวณได้มาก

โดยวิธีแบบเดิม สำหรับ $k = 1, 2, \ldots, 2^{n-1}$ เราคำนวณค่า $M_i = \sum_{j=1}^{n}a_{ij}x_j$, $i = 1, 2, \ldots, m$ แล้วจึงหาค่า $\sum_{i=1}^{m}|M_i|$ จะเห็นว่า มีการคูณ mn ครั้ง และมีการบวก mn – 1 ครั้ง ดังนั้นสำรับทุก ๆ k จะต้องมีการคูฯถึง mn 2^{n-1} ครั้ง และมีจำนวนการบวกถึง $(mn – 1)2^{n-1}$ ครั้ง

ถ้าคำนวณโดยใช้ตัวจัดเรียงสับเปลี่ยนแบบ e-2 (#1) เราให้ $M_i = \sum_{j=1}^{n}a_{ij}x_j$ ทุก ๆ i เมื่อ $x_j \in S$ เริ่มต้นเราคำนวณค่า $M_i = \sum_{j=1}^{n}a_{ij}$ ต่อไป สำหรับ $k = 1, 2, \ldots, 2^{n-1} – 1$ เราให้ j เป็นพจน์ที่ k ของลำดับ แทนที่ x_j ด้วย $-x_j$, แล้วคำนวณ:

$$M_i = M_i + 2x_ja_{ij}$$

$i = 1, 2, \ldots, m$ แล้วจึงหาค่า $\sum_{i=1}^{m}|M_i|$

โดยใช้วิธีนี้ จำนวนการคูณจะเท่ากับ $2mn + m(2^{n-1} – 1)$ ครั้ง และมีจำนวนการบวกเท่ากับ $mn – 1 + (2m – 1)(2^{n-1} – 1)$ ครั้ง

เปรียบเทียบกัน จะเห็นว่า วิธีที่เสนอจะประหยัดแรงงานได้ คือ ประหยัดจำนวนการคูณ $m((2^{n-1} – 1)(n – 1) – n)$ ครั้ง และประหยัดการบวกได้ถึง $m(n – 2) (2^{n-1} – 1)$ ครั้ง

3. สรุป และอภิปรายผล

เราได้สร้างลำดับสตริงในรูปแบบพิเศษที่จะทำให้ได้ลำดับที่สามารถเป็นลำดับนำในการชี้หน่วยที่จะเปลี่ยน ในการแปลงเพื่อที่จะทำให้ได้เซตอันดับของตัวเรียงสับเปลี่ยนทุกกรณี ซึ่งเซตนี้ก็จะเป็นผลเฉลยของปัญหาการจัดเรียงสับเปลี่ยนที่ทำให้มีค่าใช้จ่ายต่ำสุด ซึ่งสามารถจะนำไปประยุกต์ใช้ในปัญหาที่คล้ายกัน ได้แก่ปัญหาการจัดถ่ายรูปหมู่ของเครื่องแต่งกายทุก ๆ แบบ ปัญหาการจัดกรณีทดสอบซอฟต์แวร์ทุกกรณี และยังนำไปใช้ในการคำนวณค่านอร์มของเมทริกซ์ที่ทำให้ลดแรงงานการคำนวณลงได้มาก

ในการสร้างรูปแบบของลำดับ เราเริ่มพิจารณากรณีที่มีจำนวนน้อย ซึ่งอาจดูง่าย แต่ในกรณีทั่วไป หรือเมื่อจำนวนมากขึ้น การพิจารณากระทำได้ยาก ผลการวิจัยเราได้สูตรและได้วิธีการในกรณีทั่วไป มีการสร้างนิยาม สร้างทฤษฎีบทที่เกี่ยวข้อง มีการพิสูจน์ ทำให้แน่ใจได้ว่าวิธีการที่เราได้ใช้ได้ผลจริง

4. ข้อเสนอแนะ

วิธีการสร้างลำดับสตริงที่ผู้วิจัยสร้างขึ้น อาจนำไปใช้สำหรับแก้ปัญหาอื่น ๆ ได้ ปัญหาที่จะกล่าวต่อไปนี้เป็นปัญหาที่ยากขึ้น ผู้วิจัยเสนอไว้เพื่อเป็นปัญหาสำหรับหารวิจัยต่อไป

ปัญหาข้อ 1. ปัญหาการถ่ายรูปหมู่เครื่องแต่งกาย

มีคน m คน แต่ละคนมีชุดแต่งกายคนละหลายชุด จะถ่ายรูปหมู่ m คน โดยที่แต่ละคนเปลี่ยนชุดโดยการถ่ายทุก ๆ แบบของการเรียงสับเปลี่ยน ถ้าค่าเปลี่ยนชุดของแต่ละชุด ของแต่ละคนไม่เท่ากัน เราจะมีวิธีจัดเรียงสับเปลี่ยนอย่างไร เพื่อให้มีค่าใช้จ่ายในการเปลี่ยนชุดต่ำสุด (ปัญหาข้อนี้ต่างจากปัญหาที่ผู้วิจัยแก้ปัญหาที่ว่า ในปัญหาที่ผู้วิจัยทำ ราคาการเปลี่ยนชุดของคนหนึ่ง เท่ากันทุกชุด)

ปัญหาข้อ 2. ปัญหาการเฉลี่ยภาระเท่ากัน

ลักษณะเหมือนกับปัญหาการถ่ายรูปหมู่เครื่องแต่งกาย แต่ว่า ในกรณีนี้ ทุกคนมีจำนวนชุดเท่ากันและราคาการเปลี่ยนชุดของทุกคนทุกชุดเท่ากันหมด เมื่อเป็นดังนี้แต่ละคนก็ไม่อยากเปลี่ยนชุดมากเกินจำเป็น ดังนั้น จะจัดอย่างไรให้มีจำนวนการเปลี่ยนชุดของแต่ละคนเท่ากันหรือใกล้เคียงกัน คือ ต่างกันไม่เกิน 1 ครั้ง

ปัญหาทำนองนี้มีปรากฏเป็นปัญหาการจัดให้ทำงาน หรือปัญหาการเปลี่ยนสภาพ เมื่อภาระหน้าที่คือชิ้นส่วนอีเล็กโทรนิกส์ ซึ่งจะต้องจัดให้มีภาระพอ ๆ กันเพื่อไม่ให้บางชิ้นส่วนมีภาระเกินกำลัง

บรรณานุกรม

[1] อำพล ธรรมเจริญ *ลำดับสองมิติ* วารสารคณิตศาสตร์ ของสมาคมคณิตศาสตร์แห่งประเทศไทย ในพระบรมราชูปถัมภ์ ปริมา 55 ฉบับที่ 623-625 สิงหาคม- ตุลาคม 2553

[2] Anderson, J. A., "Discrete Mathematics with Combinatorics, 2nd edition", Pearson Prentice Hall, New Jersey, 2004.

[3] Deligne, S. and F. Bimbot, *Language Modelling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams,* ICASSP-95 i995 International Conference on Acoustics, Speech, and Signal Processing, May 1995.

[4] Dhamacharoen, A. *Round Robin Tournament Scheduling and Test Case Generating*, The 5 International Joint Conference on Computer Science and Software Engineering (JCSSE2008), Silapakorn University, Thailand 2008, p. 414 – 418.

[5] Drakakis, K., *On the Calculation of the* $l_2 \rightarrow l_1$ *Induced Matrix Norm*, International Journal of Algebra, Vol. 3, 2009, no. 5, 231 – 240.

[6] Fang, S-C, Yong Wang and Jie Zhong, *A Genetic Algorithm Approach to solving DNA Fragment Assembly Problem*, Journal of Computational and Theoretical Nanoscience, Vol. 2, 1-7, 2005.

[7] Gregory K. and M. Rusinowitch, *Matching a set of strings with variable length don't cares'*, Theoretical Computer Science, 178 (1997) 129-154

[8] Johnsonbaugh, R., "Discrete Mathematics", Macmillan Publishing Company, New York 1986.

[9] Jorgensen, Paul C., "Software Testing: Craftsman's Approach", 1995, CRC Press, ISBN 0849308097

[10] Kohavi, I. and Z. Kohavi, *Variable-Length Distinguishing Sequences and Their Application to the Design of Fault-Detection Experiments*, IEEE Transaction on Computer, August 1968.

[11] Lancaster, P. and M. Tismenetsky, "The Theory of Matrices, second edition with Applications", Academic Press (1985).

[12] Petrenko, A. and A. Simão, *Checking Sequence Generation Using State Distinguishing Subsequences*, ICSTW 09 Conference on Software Testing, April 2009.

[13] Weiss, M. A., "Data Structures and Algorithm Analysis, 2nd edition", The Benjamin/Cummings Publishing Company, Inc. California, 1995

[14] Rohn, J., *Computing the Norm* $||a||_{\infty,1}$ *is NP-Hard*. Linear and Multilinear Algebra, 47, 2000.

[15] Schrijver, A., "A course in Combinatorial Optimization", Department of Mathematics, University of Amsyerdam, Plantage Muidergradht 24, 2018 TV, Amsterdam, The Netherlands.

[16] Phadke, Madhav S., "Planning Efficient Software Tests", Phadke Associates, Inc. TTTThttp://www.stsc.hill.af.mil/crosstalk/1997/TTTT 10/planning.asp

ภาคผนวก

ก. การพิสูจน์ทฤษฎีบท

ข. แบบร่างบทความงานวิจัยที่พร้อมจะนำไปลงในวารสารวิชาการ

ค. ประวัตินักวิจัยและคณะ

ภาคผนวก ก.

การพิสูจน์ทฤษฎีบท

Theorem 2: Let S be a string sequence with string s_i, of length k_i, i = 1, 2, 3, . . ., and $m_i = \sum_{q=1}^{i} k_q$ and $m_0 = 0$. Let u_n be the n^{th} term of the sequence, a_{ij} the j^{th} element in the string s_i. Then, for i = 1, 2, 3, . . . ,

$$n = m_{i-1} + j, \text{ where } 1 \le j \le k_i \tag{10}$$

if, and only if,

$$u_n = a_{ij} \tag{11}$$

Proof: From (10) we have that $u_n \in s_i$ and u_n is the j^{th} term in s_i. Thus (11) holds.
From (11), u_n is the j^{th} term in s_i, then $u_n \in s_i$ and thus (10) holds. □

Theorem 3: The e-2 string sequence #1, of size m, has some special properties:

1. The string s_i has length 2^{i-1}. The string a_i has length 2^i - 1
2. The number of terms of the sequence S_{m2} is $2^m - 1$.
3. If u is the value of the j^{th} term of s_i then u is the value of the j^{th} term of s_{i+1} if i ≠ 1, and the $(2^{i-1}+ j)^{th}$ term of s_{i+1}.
4. Let u_n be the n^{th} term of the sequence, and $n = 2^{k-1}(2q - 1)$ for some positive integers k and q, then

$$u_n = k.$$

Proof 1.: We shall prove by mathematical induction. Let length(x) denote the length of x. We have length(s_1) = 1 = 2^{1-1}, and length(a_1) = 1 = $2^1 - 1$. Therefore, the proposition is true for i = 1. Suppose that it is true for i ≥ 1, i.e length(s_i) = 2^{i-1}. Then we have length(a_{i-1}) + 1 = length(s_i) = 2^{i-1}. Then,

$$\text{length}(a_i) = \text{length}(a_{i-1}) + \text{length}(s_i) = 2^{i-1} - 1 + 2^{i-1} = 2^i - 1.$$

Consequently we have

$$\text{length}(s_{i+1}) = \text{length}(a_i) + 1 = 2^i - 1 + 1 = 2^i \,;$$

the proposition is true for i +1. Thus 1. (in the above list) is true for all i. □

Proof 2.: The number of terms is the sum of the length of s_i, which is as follows:

$$\sum_{i=1}^{m} length(s_i) = \sum_{i=1}^{m} 2^{i-1} = 2^m - 1. \quad □$$

Proof 3.: From Definition 4, we have u_1 = 1, and for i > 1

$$s_i = \langle i, a_{i-1}\rangle \text{ and } a_i = \langle a_{i-1}, s_i\rangle.$$

Let s_{ij} denotes the j^{th} term of s_i, for the positive integers i, j and j ≤ 2^{i-1}. We have s_{i1} = i for i = 1, 2, ... Consider $s_{i+1} = \langle i+1, a_i\rangle$. For j ≠ 1, s_{ij} is the $(j-1)^{th}$ term of a_{i-1} and consequently the $(j\text{-}1)^{th}$ term a_j which is the j^{th} term of s_{i+1}. s_{ij} is also the $(2^{i-1}+j)^{th}$ term of s_{i+1}. The last expression is true for all the positive integers i and $j \le 2^{i-1}$. Therefore if $s_{ij} = u$, then $s_{i+1,j} = u$ for $j \neq 1$, and $s_{i+1,(2^{i-1}+j)} = u$ for $1 \le j \le 2^{i-1}$. □

Proof 4.: Let u_n be the n^{th} term of the sequence S_{m2}. Suppose that $n = 2^{k-1}(2q – 1)$ for some positive integers k and q. We will demonstrate that $u_n = k$. Note that there always exists a unique pair of positive integers k and q, such that the equality holds for each value of positive integer n.

If u_n is the j^{th} term of s_i, then $n = 2^{i-1} + j – 1$. From 3. (above), u_n has the same value for $n = 2^{i-1} + j – 1$, $n = 2^i + j – 1$ and $n = 2^i + 2^{i-1} + j – 1$ if $j \neq 1$, and it has the same value for $n = 2^{i-1}$and $n = 2^i + 2^{i-1}$.

For $i \ge 1$, if $j = 1$, we have $u_n = i$. Since $n = 2^{i-1} = 2^{k-1}(2q – 1)$, we have $q = 1$ and $k = i$. Therefore if $j = 1$, the expression is true for all $i \ge 1$..

If $j \neq 1$, we will prove the expression in 4. by way of induction on i.

For $i = 1$, we have $n = 1$ and $u_1 = 1$. Since $1 = 2^{1-1}(2(1) – 1)$, then by the formula $u_1 = 1$. The expression is true for $i = 1$.

Suppose that the expression in 4. (above) is true for all n, where $u_n \in s_i$.; that is, if $n = 2^{i-1} + j – 1 = 2^{k-1}(2q – 1)$ for some positive integers k and q, then $u_n = k$. Now let u_n be the j^{th} term of s_{i+1}, then $n = 2^i + j – 1$, $1 \le j \le 2^i$. Suppose that $n = 2^i + j – 1$
$= 2^{k-1}(2q – 1)$ for some positive integers k and q, if $1 < j \le 2^{i-1}$, then u_n has the same value as when $n = 2^{i-1} + j – 1$. Then u_n is the j^{th} term of s_i, and so we have the following:

$$\begin{aligned} 2^{i-1} + j – 1 &= 2^i + j – 1 + 2^{i-1} – 2^i \\ &= 2^{k-1}(2q – 1) – 2^{i-1} \\ &= 2^{k-1}(2q – 1 – 2^{i-k}) \\ &= 2^{k-1}(2(q\text{-}2^{i-k-1}) – 1) \end{aligned}$$

then $u_n = k$.

If $2^{i-1} < j \le 2^i$, then $n = 2^i + 2^{i-1} + j^* – 1$, $1 \le j^* \le 2^{i-1}$. By 3., u_n has the same value as when $n = 2^{i-1} + j^* – 1$. Then u_n is the j^{*th} term of s_i, and we have

$$\begin{aligned} 2^{i-1} + j^* – 1 &= 2^i + j – 1 + 2^{i-1} – 2^i – 2^{i-1} \\ &= 2^{k-1}(2q – 1) – 2^i \\ &= 2^{k-1}(2(q\text{-}2^{i-k}) – 1) \end{aligned}$$

then $u_n = k$.

Thus, the expression is true when $u_n \in s_{i+1}$. The proof is hereby complete. □

Theorem 4: The e-ni string sequence #1 has some special properties:

1. The string s_i has length $n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i – 1)$
2. The number of terms of the sequence is $n_1 \times n_2 \times \ldots \times n_m – 1$.

Proof 1.: The proof is by induction on i. Case $i = 1$ by (18a) we have $\text{length}(s_1) = n_1 – 1$. Therefore, the proposition is true for $i = 1$. Suppose the proposition is true for $i > 1$, then:

$$\text{length}(s_i) = n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i – 1). \qquad \ldots\ldots\ldots\ldots\ldots(A)$$

From (18b), we have

$$\text{length}(s_i) = (\text{length}(a_{i-1}) + 1) \times (n_i - 1)$$

Compare to (A), we have

$$\text{length}(a_{i-1}) = n_1 \times n_2 \times \ldots \times n_{i-1} - 1$$

Then

$$\begin{aligned}\text{length}(a_i) &= \text{length}(a_{i-1}) + \text{length}(s_i)\\ &= n_1 \times n_2 \times \ldots \times n_{i-1} - 1 + n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i - 1)\\ &= n_1 \times n_2 \times \ldots \times n_{i-1} \times (-1 + n_i - 1) - 1\\ &= n_1 \times n_2 \times \ldots \times n_{i-1} \times n_i - 1\end{aligned}$$

Also from (18b) $s_{i+1} = \langle\langle i, a_i\rangle_1, \ldots, \langle i, a_i\rangle_{n_{i+1}-1}\rangle$

We have

$$\begin{aligned}\text{length}(s_{i+1}) &= (n_{i+1} - 1) \times (\text{length}(a_i) + 1)\\ &= (n_{i+1} - 1) \times n_1 \times n_2 \times \ldots \times n_{i-1} \times n_i\end{aligned}$$

The case i+1 is true. Therefore

$$\text{length}(s_i) = n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i - 1)$$

is true for all i. □

Proof 2.: The number of terms is the sum of the length of s_i, which is as follows:

$$\begin{aligned}\sum_{i=1}^{m} length(s_i) &= n_1 - 1 + \sum_{i=2}^{m}(n_1 n_2 \ldots n_{i-1})(n_i - 1)\\ &= n_1 - 1 + \sum_{i=2}^{m}(n_1 n_2 \ldots n_{i-1} n_i - n_1 n_2 \ldots n_{i-1})\\ &= n_1 \times n_2 \times \ldots \times n_m - 1\end{aligned}$$ □

Theorem 1: Let S(ni, m) be defined as in Definition 1. The set S(ni, m) can be generated from an element in that set, by a sequence of $(n_1 \times n_2 \times \ldots \times n_m - 1)$ 1-bit transformations.

Proof: Let $e_1 \in S(ni, m)$ be given. The sequence e-ni combination #1 $\langle e_1, e_2, \ldots, e_N\rangle$ given in Definition 7 is obviously a subset of S(ni, m). If we can show that all e_i's are different, then we can conclude that these sets are the same. We will prove this by way of induction.
Let $N_0 = 1$, and for each $i \geq 1$, $N_i = n_1 \times n_2 \times \ldots \times n_i$. Assume that $N_i > 1$ since otherwise no change has been made.

Recall that from Definition 6:

$$s_1 = \langle 1_1, 1_2, \ldots, 1_{n_1-1}\rangle, \qquad a_1 = \langle 1_1, 1_2, \ldots, 1_{n_1-1}\rangle,$$

and for i = 2, 3, . . . , m;

$$s_i = \langle\langle i, a_{i-1}\rangle_1, \ldots, \langle i, a_{i-1}\rangle_{n_i-1}\rangle, \qquad a_i = \langle a_{i-1}, s_i\rangle.$$

And from Definition 7: $e_1 \in S(ni, m)$, and $u_i \in S_{mni,}$ then $e_{j+1} = T_{u_j}(e_j)$, j = 1, 2, . . .
Consider $\{e_1, e_2, \ldots, e_{n_1}\}$, we have $u_j = 1$ for $j = 1, \ldots, n_1 - 1$, since $u_j \in s_1$. Then all e_j's are different by the first component. The proposition is true for i = 1.

For i > 1, we make the assumption that for each $e_p \in S(ni, m)$, $e_p, e_{p+1}, \ldots, e_{p-1+N_{i-1}}$ produced by $e_{p+j} = T_{u_j}(e_{p+j-1})$, $j = 1, 2, \ldots, N_{i-1}-1$, are different. Therefore $e_1, e_2, \ldots, e_{N_{i-1}}$ are different. We will show that e_j's are different for all $j = 1, 2, \ldots, N_i$

First, consider $e_{j+1} = T_{u_j}(e_j)$ where $j = N_{i-1}$. We have $u_j = i$, which is the first element of s_i. Therefore e_{j+1} is different from e_k, for all $k = 1, 2, \ldots, N_{i-1}$, by the i^{th} component. This is also true for $j = N_{i-1} + 1, \ldots, N_i - 1$.

Note that $u_j \in s_i$, if and only if $N_{i-1} \le j < N_i$.

For $k = 1, 2, \ldots, n_i - 1$, we have the following:

$u_j \in \langle i, a_{i-1}\rangle_k$ if, and only if, $N_{i-1} + (k-1)N_{i-1} \le j < N_{i-1} + kN_{i-1}$.

For each k, let $p_k = N_{i-1} + (k-1)N_{i-1}$, then $u_{p_k} = i$. Then, for all k, $e_{p_{k+1}} = T_{u_{pk}}(e_{p_k})$, they are different by the i^{th} component.

Let $j = p_k + q$, $q = 1, 2, \ldots, N_{i-1}-1$, then $u_j \in a_{i-1}$. By assumption, all e_j's are different.

Thus, all e_j's are different for $j = N_{i-1}+1, N_{i-1}+2, \ldots, N_i$. Consequently, all e_j's are different for $j = 1, 2, \ldots, N_i$.

Therefore we can conclude that $s(ni, m) = \{e_1, e_2, \ldots, e_N\}$. The proof of Theorem 1 is complete. □

ภาคผนวก ข.

ผลผลิต (Output)

บทความที่เป็นส่วนหนึ่งของการวิจัย มี 2 บทความ คือ

1. Minimum cost Sequencing of all Combinations
2. Exact Calculation of Some Induced Matrix Norms

บทความที่จะนำไปลงพิมพ์ในวารสาร

เรื่องที่ 1

การลำดับที่มีค่าใช้จ่ายน้อยที่สุดของตัวสับเปลี่ยนทุกกรณี

Minimum Cost Sequencing of All Combinations

โดย

ร.ศ. ดร. อำพล ธรรมเจริญ

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา

Minimum Cost Sequencing of All Combinations

Ampon Dhamacharoen
Department of Mathematics, Burapha University
Bangsaen, Chonburi 20131, Thailand

Abstract: Generating all combinations can be done in many ways, but it is sometimes difficult to arrange those combinations to serve some specific purposes. This paper introduces a special sequence that can be used to navigate the changes in transition in generating all the possible combinations, in order to minimize the transition cost. The obtained ordered set can be applied to the minimum sequencing problem, such as the costume-photograph problems, the all combination test case generating problem, and the exact calculation of some induced matrix norms problem, which requires a search of all the combinations.

Keywords: All combinations, Minimum cost sequencing, 1-bit transformation, String sequence, e-2 sequence #1, Test case generating.

1. Introduction:

The sequencing problems involve problems relating to how to generate the ordered set to meet some specific purposes. Problems arise in many different types of fields of research, such as DNA sequencing in biology, tournament scheduling in games, and test case generating in software engineering. [1, 2, 3]. Here we introduce some mathematical sequencing problems which can be immediately applied to these interesting problems.

Consider the whole number in base 2 of three digits, running from 0 to 7 and which are written in order:

$$000 \quad 001 \quad 010 \quad 011 \quad 100 \quad 101 \quad 110 \quad 111. \qquad (1)$$

When considering the transition in generating these numbers from left to right, i.e. from 000 to 001, the last position is changed. However, from 001 to 010, there are two positions which are changed, and from 011 to 100, three positions are changed. Now, if we reorder them so that *the next number has one change from the previous one*, we may obtain the following:

$$000 \quad 001 \quad 011 \quad 010 \quad 110 \quad 111 \quad 101 \quad 100. \qquad (2)$$

From 000 to 001 the last digit is changed; from 001 to 011, the second digit is changed; and from 011 to 010, the last digit is changed. The position number (from right to left) where the digit changes in order is 1, 2, 1, respectively. Continue this process until all the combinations of three digits have been reached once. The positions of the changed digits are as follows:

$$1, 2, 1, 3, 1, 2, 1 \qquad (3)$$

The set of numbers in (1) and (2) are the same selection of "all combinations", from 2 digits into three fixed positions, but as ordered sets they are different. In transition from left to right, (1) has 11 changes, while (2) has only 7 changes. Note that 7 is the least number of changes, and there are many ordered sets which hold this same property. The following story may remind us of the importance of ordered sets.

The King, the Queen and a royal servant, are to be photographed together. Each one has two costumes to wear, and the photographs must be taken for all the different costumes. Therefore, a total of eight pictures are to be taken. Suppose the King, Queen and the servant are positioned as (King, Queen, Servant), and 0, 1 are used to indicate the different costumes, then the ordered set in (1) or (2) will serve as the sequence for this event. Since changing the costume is costly, it is therefore obvious that sequence (2) is more appropriate than sequence (1). If changing the King's clothes incurs the highest costs, and the servant incurs the lowest costs, then order set (2) becomes the minimum cost sequencing.

The above example illustrates the importance of the order of the elements in the set. The ordered set (1) will be called the natural order combination, and for some reason, the ordered set (2) will be called the "e-2 combination #1".

To generate the e-2 combination #1, firstly begin with the m-tuple of 0's, and then change 1 position each time, to obtain all the combinations. Selecting the position each time is important, since all the combinations must be achieved within the least number of operations. To do this systematically, we use the sequence of strings, which are called, "the leading sequence", to point out which position is to be changed. For n = 3, the leading sequence for the e-2 combinations #1 is (3) i.e. position 1, 2, 1, 3, 1, 2, 1 will be changed in order.

The aim of this research is to find the leading sequence that can generate the ordered set of all the combinations in a desired order.

The e-ni combination:

Definition 1: Let m be a positive integer. For i = 1, 2, . . . , m, let S_i be a set of n_i different types of objects. The m-tuple, in which the component i is an object taken from S_i, is called a combination of length m, or a string of length m. The set of all combinations of length m is denoted by S(ni, m). If n_i = n for all i, then the combination is called an e-n (exponential-n) combination, and the set of all e-n combinations is denoted by S(n, m). (The term ‘exponential’ is a reminder that the number of elements of S(n, m) is the power of n)

The following properties are obviously seen, and their proof is omitted.

Property 1: The set S(ni, m) has $n_1 \times n_2 \times . . . \times n_m$ elements. The set S(n, m) has n^m elements.

Property 2: The set S(n, m) is isomorphic to a set of m-digit numbers in base n, i.e. there is a one-to-one and onto function between them.

Definition 2: The set of m-digit numbers in base n, ordered by value from 0 to n^m, is called the natural order combination.

2. The Cost Sequencing of All-Combination Problems

Let m be a positive integer. For i = 1, 2, . . . , m, let S_i be a set of n_i different types of objects, S(ni, m) an ordered set of all combination $(s_1, s_2, . . . , s_m)$, $s_i \in S_i$. $N = n_1 \times n_2 \times . . . \times n_m$, and c_i the cost of replacing an element of S_i in a combination, η_{ki} =

$\begin{cases} 0, & \text{if no repacing} \\ 1, & \text{if the replacing is made} \end{cases}$. The problem is to minimize the total cost $\sum_{k=1}^{N}\left(\sum_{i=1}^{m}\eta_{ki}c_i\right)_k$ over the set of ordered sets S(ni, m).

The costume photograph problem: A group of m persons will be taking picture with different costumes belonging to each person, for all the different combinations of these costumes. Suppose that for each person, the cost for changing all his/her costumes are the same. Finding the sequence (i.e. who will change the costume the next time) that will minimize the costs for changing the costumes is the costume photograph problem.

The all-combination test case generating problem: In software engineering, all software has to be tested before it can be released to the users. Testing software can be performed by inputting the selected test data into the input components, and then executing. The all-combination test suit is the most efficient test suit that can be used to detect the error, but it also the most energy-consuming since changing the data each time may be associated with the costs (or time) [3, 4]. Therefore, the problem that arises is how to generate the ordered set of all-combination test suits in a manner which can minimize the costs for changing the data.

The induced matrix norm computation problems: Let A be a real m × n matrix. When considering A as a linear transformation between two finite-dimensional vector spaces, the induced norm of A can be defined as follows:

$$||A||_{pq} = \sup_{||\mathbf{x}||_p=1} ||A\mathbf{x}||_q \tag{4}$$

where $\mathbf{x} \in R^{n\times 1}$. The closed form formula for computing the value of norms for p, q = 1, 2 and ∞ are known [5, 6, 7]. Listed below are some norms whereby their computation concerns all the combinations of 2 elements.

$$||A||_{\infty 1} = \max_{x_j\in S} \{\sum_{i=1}^{m}|\sum_{j=1}^{n} a_{ij}x_j|\}, \quad \text{where } S = \{-1, 1\} \tag{5}$$

$$||A||_{21} = \max_{s_i\in S} \{(\sum_{j=1}^{n}(\sum_{i=1}^{m} s_i a_{ij})^2)^{1/2}\}, \text{ where } S = \{-1, 1\} \tag{6}$$

$$||A||_{\infty 2} = \max_{x_j\in S} [\sum_{i=1}^{m}(\sum_{j=1}^{n} a_{ij}x_j)^2]^{1/2}, \text{ where } S = \{-1, 1\} \tag{7}$$

To compute these norms, the value $\sum_{i=1}^{m}|\sum_{j=1}^{n} a_{ij}x_j|$ in (5), $\sum_{j=1}^{n}(\sum_{i=1}^{m} s_i a_{ij})^2$ in (6) or $\sum_{i=1}^{m}(\sum_{j=1}^{n} a_{ij}x_j)^2$ in (7) must be computed for each combination in S(2, m). The method for carrying out computations with minimum costs (number of calculations) will be described in the next section.

3. 1-bit Transformation:

Definition 3: Let C ⊆ S(ni, m). A transformation T on C is called a 1-bit transformation if T(c) and c are different by 1 position.

If S_i has two elements, the 1–bit transformation that changes the element in position i is denoted by T_i. If S_i has more than two elements, then the element of S_i needs to be specified. The 1-bit transformation which changes the element in position i to a is denoted by T_{ia}. For example, if $S_i = \{0, 1\}$ and the string has length 4, then $T_2(1100) = 1000$ and $T_3(1100) = 1110$. If $S_i = \{1, 2, 3\}$ then $T_{23}(1122) = 1322$.

Theorem 1: Let S(ni, m) be defined as in Definition 1. The set S(ni, m) can be generated from an element in that set, by a sequence of $(n_1 \times n_2 \times \ldots \times n_m - 1)$ 1-bit transformations.

We will postpone the proof until the end of this section.

The string sequence:

Let S be a set of objects. , A string from S is a finite sequence of objects from S. A string sequence is a sequence that is composed of elements of the strings. The strings in the sequence may have a constant or variable length. The discussion relating to these topics concerns the changing between the elements of the strings, and the terms of the sequence [8, 9]. Below, are examples of sequences of this type:

$$1, 2, 3, 2, 3, 4, 3, 4, 5, \ldots \tag{8}$$
$$1, 1, 2, 1, 2, 3, 1, 2, 3, 4, \ldots \tag{9}$$

(8) is the sequence of strings $\langle 1, 2, 3\rangle$, $\langle 2. 3. 4\rangle$, $\langle 3, 4, 5\rangle$, . . . which are of equal length. (9) is the sequence of strings $\langle 1\rangle$, $\langle 1, 2\rangle$, $\langle 1, 2, 3\rangle$, $\langle 1, 2, 3, 4\rangle$, . . . for which the length is varied. From a sequence, the strings can be set up in many ways depending on its meaning and usage. For convenience, let us define the type of string sequence as follows:

c-k string sequence: the sequence where the strings have a constant of length k.
n-k string sequence : the sequence where the length of string varies as a multiple of k.
e-k string sequence : the sequence where the length of string varies as a power of k.

Theorem 2: Let S be a string sequence with string s_i, of length k_i, i = 1, 2, 3, . . ., and $m_i = \sum_{q=1}^{i} k_q$ and $m_0 = 0$. Let u_n be the n^{th} term of the sequence, a_{ij} the j^{th} element in the string s_i. Then, for i = 1, 2, 3, . . . ,

$$n = m_{i-1} + j, \text{ where } 1 \le j \le k_i \tag{10}$$

if, and only if,

$$u_n = a_{ij} \tag{11}$$

Proof: From (10) we have that $u_n \in s_i$ and u_n is the j^{th} term in s_i. Thus (11) holds.
From (11), u_n is the j^{th} term in s_i, then $u_n \in s_i$ and thus (10) holds. □

Interchanging between the elements of the string and the term of sequence, can be carried out by using formula (10) and (11).

To illustrate, let us consider an exponential-length string sequence:

1, 1, 2, 1, 2, 3, 4, 1, 2, 3, 4, 5, 6, 7, 8, ...

in which the strings are ⟨1⟩, ⟨1, 2⟩, ⟨1, 2, 3, 4⟩, ⟨1, 2, 3, 4, 5, 6, 7, 8⟩, ...which are of length 2^{k-1}, the e-2 sequence. Such sequence can be written in two-dimensional form as follows:

1							
1	2						
1	2	3	4				
1	2	3	4	5	6	7	8
-	-	-	-	-	-	-	-

Using (6) and (7) we have that for i = 1, 2, 3, . . . , $k_i = 2^{i-1}$, $m_i = \sum_{q=1}^{i} 2^{q-1} = 2^i - 1$, $a_{ij} = j$, for $1 \le j \le 2^{i-1}$. Then $n = 2^{i-1} - 1 + j$ and we have

$$u_n = n - 2^{i-1} + 1, \quad 2^{i-1} \le n \le 2^i - 1, \tag{12}$$

i = 1, 2, 3, ...

The e-2 string sequence #1

Let us consider the set S(2, 4) of whole numbers in base 2 of four digits, which are ordered in such a way that the next number has one change from the previous one:

0000 0001 0011 0010 0110 0111 0101 0100 (13)
1100 1101 1111 1110 1010 1011 1001 1000

The positions of the changed digit are

$$1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1 \tag{14}$$

which can be viewed as an exponential-length string sequence:

$$\langle 1\rangle, \langle 2, 1\rangle, \langle 3, 1, 2, 1\rangle, \langle 4, 1, 2, 1, 3, 1, 2, 1\rangle. \tag{15}$$

This sequence (15) is called the *e-2 string sequence #1*, while (14) is called the *e-2 sequence #1*. We use (14) as a leading sequence to generate the e-2 combination #1 of length 4 in (13).

Definition 4: The e-2 string sequence #1 of size m, denoted by S_{m2}, is a sequence in the form

$$s_1, s_2, s_3, \dots, s_m \tag{16}$$

where m is a positive integer, and s_i is a string generated by the following rule:

$$s_1 = \langle 1\rangle \qquad a_1 = \langle 1\rangle \tag{16a}$$

For i = 2, 3, . . . , m;

$$s_i = \langle i, a_{i-1}\rangle \qquad a_i = \langle a_{i-1}, s_i\rangle \tag{16b}$$

while the notation $\langle a_{i-1}, s_i\rangle$ is the sequence composed of elements in the strings a_{i-1} and s_i (not the pair of strings a_{i-1} and s_i). The e-2 sequence #1, which is also denoted by S_{m2}, is a sequence whose terms are from elements of the e-2 string sequence #1.

The reader may easily verify that for m = 4, (16a) and (16b) produce (14). Note that from the definition, the first term of s_i is i, for all positive integer i.

Theorem 3: The e-2 string sequence #1, of size m, has some special properties:

1. The string s_i has length 2^{i-1}. The string a_i has length 2^i - 1

2. The number of terms of the sequence S_{m2} is $2^m - 1$.
3. If u is the value of the j^{th} term of s_i then u is the value of the j^{th} term of s_{i+1} if $j \neq 1$, and the $(2^{i-1}+j)^{th}$ term of s_{i+1}.
4. Let u_n be the n^{th} term of the sequence, and $n = 2^{k-1}(2q - 1)$ for some positive integers k and q, then

$$u_n = k. \tag{17}$$

Proof 1.: We shall prove by mathematical induction. Let length(x) denote the length of x. We have length(s_1) = 1 = 2^{1-1}, and length(a_1) = 1 = $2^1 - 1$. Therefore, the proposition is true for i = 1. Suppose that it is true for $i \geq 1$, i.e length(s_i) = 2^{i-1}. Then, from (12b), length(a_{i-1}) + 1 = length(s_i) = 2^{i-1}. Then,

$$\text{length}(a_i) = \text{length}(a_{i-1}) + \text{length}(s_i) = 2^{i-1} - 1 + 2^{i-1} = 2^i - 1.$$

Consequently we have

$$\text{length}(s_{i+1}) = \text{length}(a_i) + 1 = 2^i - 1 + 1 = 2^i;$$

the proposition is true for i +1. Thus 1. (in the above list) is true for all i. □

Proof 2.: The number of terms is the sum of the length of s_i, which is as follows:

$$\sum_{i=1}^{m} length(s_i) = \sum_{i=1}^{m} 2^{i-1} = 2^m - 1. \qquad \square$$

Proof 3.: From Definition 4, we have $u_1 = 1$, and for i > 1

$$s_i = \langle i, a_{i-1}\rangle \text{ and } a_i = \langle a_{i-1}, s_i\rangle.$$

Let s_{ij} denotes the j^{th} term of s_i, for the positive integers i, j and $j \leq 2^{i-1}$. We have s_{i1} = i for i = 1, 2, ... Consider $s_{i+1} = \langle i+1, a_i\rangle$. For $j \neq 1$, s_{ij} is the $(j-1)^{th}$ term of a_{i-1} and consequently the $(j-1)^{th}$ term a_i which is the j^{th} term of s_{i+1}. s_{ij} is also the $(2^{i-1}+ j)^{th}$ term of s_{i+1}. The last expression is true for all the positive integers i and $j \leq 2^{i-1}$. Therefore if s_{ij} = u, then $s_{i+1,j}$ = u for $j \neq 1$, and $s_{i+1,(2^{i-1}+j)}$ = u for $1 \leq j \leq 2^{i-1}$. □

Proof 4.: Let u_n be the n^{th} term of the sequence S_{m2}. Suppose that $n = 2^{k-1}(2q - 1)$ for some positive integers k and q. We will demonstrate that u_n = k. Note that there always exists a unique pair of positive integers k and q, such that the equality holds for each value of positive integer n.

If u_n is the j^{th} term of s_i, then $n = 2^{i-1} + j - 1$. From 3. (above), u_n has the same value for $n = 2^{i-1} + j - 1$, $n = 2^i + j - 1$ and $n = 2^i + 2^{i-1} + j - 1$ if $j \neq 1$, and it has the same value for $n = 2^{i-1}$and $n = 2^i + 2^{i-1}$.

For $i \geq 1$, if j = 1, we have u_n = i. Since $n = 2^{i-1} = 2^{k-1}(2q - 1)$, we have q = 1 and k = i. Therefore if j = 1, the expression is true for all $i \geq 1$..

If $j \neq 1$, we will prove the expression in 4. by way of induction on i.

For i = 1, we have n = 1 and u_1 = 1. Since $1 = 2^{1-1}(2(1) - 1)$, then by the formula u_1 = 1. The expression is true for i = 1.

Suppose that the expression in 4. (above) is true for all n, where $u_n \in s_i$.; that is, if $n = 2^{i-1} + j - 1 = 2^{k-1}(2q - 1)$ for some positive integers k and q, then $u_n = k$. Now let u_n be the j^{th} term of s_{i+1}, then $n = 2^i + j - 1$, $1 \le j \le 2^i$. Suppose that $n = 2^i + j - 1$
$= 2^{k-1}(2q - 1)$ for some positive integers k and q, if $1 < j \le 2^{i-1}$, then u_n has the same value as when $n = 2^{i-1} + j - 1$. Then u_n is the j^{th} term of s_i, and so we have the following:

$$\begin{aligned} 2^{i-1} + j - 1 &= 2^i + j - 1 + 2^{i-1} - 2^i \\ &= 2^{k-1}(2q - 1) - 2^{i-1} \\ &= 2^{k-1}(2q - 1 - 2^{i-k}) \\ &= 2^{k-1}(2(q-2^{i-k-1}) - 1) \end{aligned}$$

then $u_n = k$.

If $2^{i-1} < j \le 2^i$, then $n = 2^i + 2^{i-1} + j^* - 1$, $1 \le j^* \le 2^{i-1}$. By 3., u_n has the same value as when $n = 2^{i-1} + j^* - 1$. Then u_n is the j^{*th} term of s_i, and we have

$$\begin{aligned} 2^{i-1} + j^* - 1 &= 2^i + j - 1 + 2^{i-1} - 2^i - 2^{i-1} \\ &= 2^{k-1}(2q - 1) - 2^i \\ &= 2^{k-1}(2(q-2^{i-k}) - 1) \end{aligned}$$

then $u_n = k$.

Thus, the expression is true when $u_n \in s_{i+1}$. The proof is hereby complete. □

Definition 5: Let $c_1 \in S(2, m)$, and S_m be the e-2 string sequence #1 of size m, T_k, a 1-bit transformation, $k \in S_m$, $N = 2^m$. The sequence $\langle c_1, c_2, \ldots, c_N \rangle$ where $c_{i+1} = T_{u_i}(c_i)$, i = 1, 2, . . . , N-1, is called the e-2 combination #1 generated by c_1.

Corollary: The e-2 combination #1 is the ordered set of all the combinations, with the least number of transition changes.
Proof: Follow from Theorem 4 that $S(2, m) = \{c_1, c_2, \ldots, c_N\}$, which is all the combinations. The number of transitions from c_1 to c_N is $N - 1$, which is the minimum number of transitions required. □

The e-ni string sequence #1

Now we generalize the string sequence in order to generate the e-ni combination ordered set.

Definition 6: Let m, $n_1, n_2, \ldots, n_m$ be positive integers. The e-ni string sequence #1 of size m, denoted by S_{mni}, is a sequence in the form:

$$s_1, s_2, s_3, \ldots, s_m \qquad (18)$$

where m is a positive integer, and s_i is a string generated by the following rule:

$$s_1 = \langle 1_1, 1_2, \ldots, 1_{n_1\text{-}1} \rangle \qquad a_1 = \langle 1_1, 1_2, \ldots, 1_{n_1\text{-}1} \rangle \qquad (18a)$$

(s_1 and a_1 has $n_1 - 1$ terms of 1's)
For i = 2, 3, . . . , m;

$$s_i = \langle \langle i, a_{i-1} \rangle_1, \ldots, \langle i, a_{i-1} \rangle_{n_i\text{-}1} \rangle \qquad a_i = \langle a_{i-1}, s_i \rangle \qquad (18b)$$

where the notation $\langle a_{i-1}, s_i\rangle$ is the sequence composed of the elements in the strings a_{i-1} and s_i (not the pair of strings a_{i-1} and s_i). The e-ni sequence #1 is the sequence whose terms are from the e-ni string sequence #1, and which are also denoted by S_{mni}.

Theorem 4: The e-ni string sequence #1 has some special properties:

1. The string s_i has length $n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i - 1)$
2. The number of terms of the sequence is $n_1 \times n_2 \times \ldots \times n_m - 1$.

Proof 1.: The proof is by induction on i. Case i = 1 by (18a) we have $\text{length}(s_1) = n_1 - 1$. Therefore, the proposition is true for i = 1. Suppose the proposition is true for i > 1, then:

$$\text{length}(s_i) = n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i - 1). \quad \ldots\ldots\ldots\ldots\ldots(A)$$

From (18b), we have

$$\text{length}(s_i) = (\text{length}(a_{i-1}) + 1) \times (n_i - 1)$$

Compare to (A), we have

$$\text{length}(a_{i-1}) = n_1 \times n_2 \times \ldots \times n_{i-1} - 1$$

Then

$$\begin{aligned}\text{length}(a_i) &= \text{length}(a_{i-1}) + \text{length}(s_i)\\ &= n_1 \times n_2 \times \ldots \times n_{i-1} - 1 + n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i - 1)\\ &= n_1 \times n_2 \times \ldots \times n_{i-1} \times (-1 + n_i - 1) - 1\\ &= n_1 \times n_2 \times \ldots \times n_{i-1} \times n_i - 1\end{aligned}$$

Also from (18b) $\quad s_{i+1} = \langle\langle i, a_i\rangle_1, \ldots, \langle i, a_i\rangle_{n_{i+1}-1}\rangle$

We have

$$\begin{aligned}\text{length}(s_{i+1}) &= (n_{i+1} - 1) \times (\text{length}(a_i) + 1)\\ &= (n_{i+1} - 1) \times n_1 \times n_2 \times \ldots \times n_{i-1} \times n_i\end{aligned}$$

The case i+1 is true. Therefore

$$\text{length}(s_i) = n_1 \times n_2 \times \ldots \times n_{i-1} \times (n_i - 1)$$

is true for all i. □

Proof 2.: The number of terms is the sum of the length of s_i, which is as follows:

$$\begin{aligned}\sum_{i=1}^{m} length(s_i) &= n_1 - 1 + \sum_{i=2}^{m}(n_1 n_2 \ldots n_{i-1})(n_i - 1)\\ &= n_1 - 1 + \sum_{i=2}^{m}(n_1 n_2 \ldots n_{i-1} n_i - n_1 n_2 \ldots n_{i-1})\\ &= n_1 \times n_2 \times \ldots \times n_m - 1\end{aligned}$$ □

Definition 7: Let $e_1 \in S(ni, m)$, and S_{mni} the e-ni sequence #1 of size m, T_{u_i} a 1-bit transformation, $u_i \in S_{mni}$, $N = n_1 \times n_2 \times \ldots \times n_m$. The sequence $\langle e_1, e_2, \ldots, e_N\rangle$ where $e_{i+1} = T_{u_i}(e_i)$, i = 1, 2, . . . , N-1, is called the e-ni combination #1 generated by e_1.

Since for some i, S_i has more than two elements, therefore, in order to avoid ambiguity while performing the 1-bit transformation, the elements of S_i must be ordered so that the change at the position i can be carried out in a one directional circle..

Example: Let m = 3, $S_1 = \{1, 2, 3, 4\}$, $S_2 = \{1, 2, 3\}$, $S_3 = \{1, 2\}$, $e_1 = 111$

We have N = 4 × 3 × 2 = 24.

S_{mni} = 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1
(Component count from right to left)

e-ni combination #1 = 111,
112, 113, 114, 124, 121, 122, 123, 133, 134, 131, 132, 232, 233, 234, 231, 211, 212, 213, 214, 224, 221, 222, 223

Prove of Theorem 1:

Let $e_1 \in S(ni, m)$ be given. The sequence e-ni combination #1 $\langle e_1, e_2, \ldots, e_N\rangle$ given in Definition 7 is obviously a subset of S(ni, m). If we can show that all e_i's are different, then we can conclude that these sets are the same. We will prove this by way of induction.

Let $N_0 = 1$, and for each $i \geq 1$, $N_i = n_1 \times n_2 \times \ldots \times n_i$. Assume that $N_i > 1$ since otherwise no change has been made.

Recall that from Definition 6:

$$s_1 = \langle 1_1, 1_2, \ldots, 1_{n_1-1}\rangle, \qquad a_1 = \langle 1_1, 1_2, \ldots, 1_{n_1-1}\rangle,$$

and for i = 2, 3, . . . , m;

$$s_i = \langle\langle i, a_{i-1}\rangle_1, \ldots, \langle i, a_{i-1}\rangle_{n_i-1}\rangle, \qquad a_i = \langle a_{i-1}, s_i\rangle.$$

And from Definition 7: $e_1 \in S(ni, m)$, and $u_i \in S_{mni,}$ then $e_{j+1} = T_{u_j}(e_j)$, j = 1, 2, . . .

Consider $\{e_1, e_2, \ldots, e_{n_1}\}$, we have $u_j = 1$ for $j = 1, \ldots, n_1 -1$, since $u_j \in s_1$. Then all e_j's are different by the first component. The proposition is true for i = 1.

For i > 1, we make the assumption that for each $e_p \in S(ni, m)$, $e_p, e_{p+1}, \ldots, e_{p-1+N_{i-1}}$ produced by $e_{p+j} = T_{u_j}(e_{p+j-1})$, $j = 1, 2, \ldots, N_{i-1}-1$, are different. Therefore $e_1, e_2, \ldots, e_{N_{i-1}}$ are different. We will show that e_j's are different for all $j = 1, 2, \ldots, N_i$

First, consider $e_{j+1} = T_{u_j}(e_j)$ where $j = N_{i-1}$. We have $u_j = i$, which is the first element of s_i. Therefore e_{j+1} is different from e_k, for all $k = 1, 2, \ldots, N_{i-1}$, by the i^{th} component. This is also true for $j = N_{i-1} + 1, \ldots, N_i - 1$.

Note that $u_j \in s_{i,}$ if and only if $N_{i-1} \leq j < N_i$.

For $k = 1, 2, \ldots, n_i - 1$, we have the following:

$u_j \in \langle i, a_{i-1}\rangle_k$ if, and only if, $N_{i-1} + (k-1)N_{i-1} \leq j < N_{i-1} + kN_{i-1}$.

For each k, let $p_k = N_{i-1} + (k-1)N_{i-1}$, then $u_{p_k} = i$. Then, for all k, $e_{p_{k+1}} = T_{u_{pk}}(e_{p_k})$, they are different by the i^{th} component.

Let $j = p_k + q$, $q = 1, 2, \ldots, N_{i-1}-1$, then $u_j \in a_{i-1}$. By assumption, all e_j's are different.

Thus, all e_j's are different for $j = N_{i-1}+1, N_{i-1}+2, \ldots, N_i$. Consequently, all e_j's are different for $j = 1, 2, \ldots, N_i$.

Therefore we can conclude that $s(ni, m) = \{e_1, e_2, \ldots, e_N\}$. The proof of Theorem 1 is complete. □

4. Application:

Solution for the all-combination cost sequencing problems

Let us consider the costume-photograph problem, and the all-combination test case generating problem, as defined in Section 2:

Minimize the total cost $\sum_{k=1}^{N}\left(\sum_{i=1}^{m}\eta_{ki}c_i\right)_k$ over the set of ordered sets S(ni, m).

Suppose that $c_1 \le c_2 \le \ldots \le c_m$, then the problems already have a solution: the e-ni combination #1 $\langle e_1, e_2, \ldots, e_N\rangle$, as described in the earlier section. The total costs are the setting costs for e_1, and the operating (changing) costs in transition. Since there is only one change in each transition, the cost can be calculated by counting the total number of changes in each component, times the cost, and adding these totals up to obtain the total cost. Using the e-ni sequence #1 to navigate the change, the number of total changes in the i^{th} component are:

$$q_1 = (n_1-1) \times n_2 \times \ldots \times n_m$$
$$q_2 = (n_2-1) \times n_3 \times \ldots \times n_m$$
$$- - - - -$$
$$q_m = n_m - 1$$

Then, the total cost is $C + \sum_{i=1}^{m} q_i c_i$, where C is the setting cost.

Note that the total number of changes is $\sum_{i=1}^{m} q_i c_i = n_1 \times n_2 \times \ldots \times n_m - 1$.

Computation of the induced matrix norm problems:

We will describe the method to calculate the value of the norm in (5) by using the e-2 combination #1. The algorithm to compute (6) and (7) will be analogous to that of (5).

To evaluate $\sum_{i=1}^{m}|\sum_{j=1}^{n} a_{ij}x_j|$, since the values are in absolute value, we only need to compute half of all the combinations, i.e. the combination of (n-1)-tuple are needed. This can reduce the number of e-2 combination #1 to 2^{n-1}, instead of 2^n.

The number of operations in the computation can be reduced by firstly computing such values for a particular combination, and then for the next combination, we compute only at the changed component, and add or subtract from the old values.

In computing by using the conventional method, we proceed as follows: For each k = 1, 2, . . . , 2^{n-1}, we calculate the value of $M_i = \sum_{j=1}^{n} a_{ij}x_j$, i = 1, 2, . . . , m, and then find $\sum_{i=1}^{m}|M_i|$. Using this formula, the number of multiplications is mn, and the number of additions is mn – 1. Subsequently, for all k, it is necessary to carry out mn 2^{n-1} multiplications and $(mn - 1)2^{n-1}$ additions.

Using the e-2 combination #1, we can compute such value using less effort. In computing of $M_i = \sum_{j=1}^{n} a_{ij}x_j$ for all i where $x_j \in S$, we first generate the e-2 sequence #1 and start with the value of $M_i = \sum_{j=1}^{n} a_{ij}$. For k = 1, 2, . . . , $2^{n-1} - 1$, let j be the k^{th} term of the sequence, replace x_j by $-x_j$, then compute:

$$M_i = M_i + 2x_j a_{ij}$$

i = 1, 2, . . . , m. (The value $2a_{ij}$ can be set prior, in order to avoid repetition of the calculation.) Then, compute $\sum_{i=1}^{m} | M_i |$.

Using this method, the number of calculations will be $2mn + m(2^{n-1} - 1)$ multiplications, and $mn - 1 + (2m - 1)(2^{n-1} - 1)$ additions.

Compared to the conventional method, using the e-2 combination #1 can reduce the number of calculations by $m((2^{n-1} - 1)(n - 1) - n)$ multiplications and $m(n - 2)\ (2^{n-1} - 1)$ additions.

5. Conclusion:

The sequence of string be defined in a certain way, in order to obtain the sequence that can navigate the changes in transition when generating the ordered set of all combinations, which is the solution for the all combination minimum cost sequencing problem. The method of calculation concerning the all-combination constructed based on that special sequence can reduce the number of calculations as compared to using the conventional method. The sequence may look easy for the first few terms, but generalizing such a manner is not quite so easy. The method to construct such a sequence is given here, and its properties were proved in general. The problem may be viewed as being similar to the traveling salesman problem [10], and the proof of Theorem 1 could be carried out in another way. However, the proof given here can provide information on how to generate the path, which is useful in applications.

There are more interesting problems concerning all the combination searches. The examples proposed here, may be considered as open problems:

Problem 1. The costume-photograph problem: A group of m persons will take photos with different costumes belonging to each person, for all different combinations of costumes. The costs of changing the costume for each person are different, depending on the types of costumes. The problem is to minimize the cost of changing the costumes for all the combinations of pictures.

Problem 2. The duty-balancing problem: Given a situation which is the same as the costume-photograph problem. However, now the costs of changing the costumes are all equal, and each person does not want to change the costumes too many times. Therefore, the problem is to find the ordered set of all combinations in such a way that the number of changes in each component is different than the other by at most one.

This problem may occur in assignment problems, or switching problems, where the duties are devices in an electronic circuit, which should be balanced to avoid overloading some devices.

Note: The following are some solutions for Problem 2:

For S(2, 3): The solution set is 000, 001, 011, 111, 101, 100, 110, 010

The number of changes in component 1, 2, 3 is 2, 3, 2 respectively.

Note that the total number of changes is 2 + 3 + 2 = 7.

For S(2, 4): The solution set is 0000, 0001, 0011, 0111, 1111, 1101, 0101, 0100, 1100, 1000, 1001, 1011, 1010, 1110, 0110, 0010

The number of changes in component 1, 2, 3, 4 is 4, 3, 4, 4 respectively.

Note that the total number of changes is 4 + 3 + 4 + 4 = 15.

Acknowledgments
The author would like to thank the referees for their comments. This research was supported by the Research Fund from the National Research Council of Thailand

References

[1] Fang, S-C, Yong Wang and Jie Zhong, A Genetic Algorithm Approach to solving DNA Fragment Assembly Problem, *Journal of Computational and Theoretical Nanoscience*, Vol. 2, 1-7, 2005.
[2] Dhamacharoen, A. Round Robin Tournament Scheduling and Test Case Generating, *The 5 International Joint Conference on Computer Science and Software Engineering (JCSSE2008), Silapakorn University, Thailand 2008*, p. 414 – 418.
[3] Jorgensen, Paul C., "Software Testing: Craftsman's Approach", 1995, CRC Press, ISBN 0849308097
[4] Phadke, Madhav S., "Planning Efficient Software Tests", Phadke Associates, Inc. http://www.stsc.hill.af.mil/crosstalk/1997/. 10/planning.asp
[5] Dhamacharoen, A., Exact Calculation of Some Induced Matrix Norms, *will be appeared.*
[6] Drakakis, K., On the Calculation of the $l_2 \to l_1$ Induced Matrix Norm, *International Journal of Algebra*, Vol. 3, 2009, no. 5, 231 – 240.
[7] Rohn, J., Computing the Norm ||a||∞.1 is NP-Hard. *Linear and Multilinear Algebra*, 47, 2000.
[8] Petrenko, A. and A. Simão, Checking Sequence Generation Using State Distinguishing Subsequences, *ICSTW 09 Conference on Software Testing, April 2009*.
[9] Kohavi, I. and Z. Kohavi, *Variable-Length Distinguishing Sequences and Their Application to the Design of Fault-Detection Experiments*, IEEE Transaction on Computer, August 1968.
[10] Schrijver, A., "A course in Combinatorial Optimization", Department of Mathematics, University of Amsyerdam, Plantage Muidergradht 24, 2018 TV, Amsterdam, The Netherlands.

บทความที่จะนำไปลงพิมพ์ในวารสาร

เรื่องที่ 2

การคำนวณค่าแท้จริงของนอร์มของเมทริกซ์บางแบบ

Exact Calculation of Some Induced Matrix Norms

โดย

ร.ศ. ดร. อำพล ธรรมเจริญ

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา

Exact Calculation of Some Induced Matrix Norms

Ampon Dhamacharoen
Department of Mathematics, Faculty of Sciences
Burapha University, Bangsaen, Chonburi 20131, Thailand.
e-mail: ampon@buu.ac.th

Abstract: This paper will investigate the induced matrix norms of linear operators between finite dimensional vector spaces. In the past, the closed form formulas for $||A||_{pq}$ were known for some p, q = 1, 2, ∞, but not for $||A||_{2\infty}$ and $||A||_{\infty 2}$. The formulas for these two norms and for $||A||_{21}$ are derived in this paper. The computation of $||A||_{21}$, $||A||_{\infty 1}$ and $||A||_{\infty 2}$ are generally difficult, since they concern all the combination arrangements of objects. The algorithms developed here, can provide the exact numerical value, with the least amount of calculations.

Keywords: Matrix norm; Induced matrix norms; Norm computation.

1. Introduction:

Let R^n be the vector space of n-tuple of real numbers over a field of real numbers. $T:R^n \to R^m$ is a linear operator defined by $T(\mathbf{x}) = A\mathbf{x}$, where $A = [a_{ij}]_{m\times n}$ is a matrix with real entries, and $\mathbf{x} = [x_1 \quad x_2 \quad ... \ x_n]^T$ is a vector in R^n written in the matrix form. The matrix p-norm can be defined for p > 0 as follows:

$$||T||_p = \sup_{||x||_p=1} ||T(x)||_p \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

If p < 1, the function $|| \cdot ||_p$ lacks triangle inequality, and may be called a semi-norm. For convenience, we will write $||A||$ instead of $||T||$. The normed values, when p = 1, 2 or ∞, can be calculated through the given formulas below:

$$||A||_1 = \max_{1\le j\le n} \{\sum_{i=1}^{m} | a_{ij} |\} \qquad (2)$$

$$||A||_\infty = \max_{1\le i\le m} \{\sum_{j=1}^{n} | a_{ij} |\} \qquad (3)$$

$$||A||_2 = \sqrt{\lambda}\,, \qquad (4)$$

where λ is the largest eigenvalue of A^*A.

These formulas are well known and can be found in most matrix theory or numerical analysis text books(e.g. [1]). However, for the other value of p, the closed form formulas of the norms are not known.

Suppose the operator T has been defined according to different normed spaces, i.e., from l_p to l_q, then the induced matrix norm can be defined as follows:

$$||T||_{pq} = \sup_{||\mathbf{x}||_p=1} ||T(\mathbf{x})||_q \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

These norms are a generalization of (1), or (1) can be viewed as a special case of (2) if p = q, and therefore, we write $||T||_p$ instead of $||T||_{pp}$. In finite dimensional vector space, the supremum is the maximum. The closed form formula for computing the value of norms for p, q = 1, 2 and ∞ are known, and are listed below:

$$||A||_{1\infty} = \max_{||\mathbf{x}||_1=1} ||A\mathbf{x}||_\infty = \max_{1\le i\le m, 1\le j\le n} \{|a_{ij}|\} \quad (6)$$

$$||A||_{\infty 1} = \max_{||\mathbf{x}||_\infty=1} ||A\mathbf{x}||_1 = \max_{x_j\in S} \{\sum_{i=1}^{m} |\sum_{j=1}^{n} a_{ij}x_j|\} \text{ where } S = \{-1, 1\} \quad (7)$$

$$||A||_{12} = \max_{||\mathbf{x}||_1=1} ||A\mathbf{x}||_2 = \max_{1\le j\le n} (\sum_{i=1}^{m} a_{ij}^2)^{1/2} \quad (8)$$

$$||A||_{21} = \max_{||\mathbf{x}||_2=1} ||A\mathbf{x}||_1 = \max_{s_i\in S} \{(\sum_{j=1}^{n}(\sum_{i=1}^{m} s_i a_{ij})^2)^{1/2}\}, \text{ where } S = \{-1, 1\} \quad (9)$$

$$||A||_{\infty 2} = \max_{||\mathbf{x}||_\infty=1} ||A\mathbf{x}||_2 = \max_{x_j\in S} [\sum_{i=1}^{m}(\sum_{j=1}^{n} a_{ij}x_j)^2]^{1/2}, \text{ where } S = \{-1, 1\} \quad (10)$$

$$||A||_{2\infty} = \max_{||\mathbf{x}||_2=1} ||A\mathbf{x}||_\infty = \max_{1\le i\le m} (\sum_{j=1}^{n} a_{ij}^2)^{1/2} \quad (11)$$

(6) and (7) appeared in Rohn [2]. (8) and (9) appeared in Drakakis [2], with a different form:

$$||A||_{21} = \max_{s_i\in S} \{||\mathbf{s}^T A||_2\} \quad \text{where } \mathbf{s}^T = [s_1\ s_2\ \ldots\ s_m],\ S = \{-1, 1\}, \quad (12)$$

and also (8) was proved in general for $0 < p \le 1 \le q$:

$$||A||_{pq} = \max_{1\le j\le m} \{||\mathbf{a}_{.j}||_q\},$$

where $\mathbf{a}_{.j}$ is the j^{th} column of A. Drakakis also gave the formula (10) $||A||_{\infty 2} = ||A^T||_{12}$, which is incorrect. (10), (11) and (9) are derived here in the next section, using a geometric approach.

Although the formulas are in a closed form, the computation concerning all the combination searches is still difficult, since they require huge amount of work [3, 4]. The formula (7), (9) and (10) are of this type. In Section 3, we will introduce algorithms for computing such values, which will save some energy.

2. Derivation:

Although formula (9) was proved in [3], the derivation is somewhat difficult, but also motivating and challenging, and so we will redo the proof here using a geometric approach.

In a rectangular axis, the graph of the equation ax + by = c is called a line, a plane in 3-dimensional, and for more than 3-dimensional vector spaces, the graph of $\sum_{i=1}^{n} a_i x_i = c$ will be called a hyper-plane. The graph of $\sum_{i=1}^{n} x_i^2 = a^2$ (or $||\mathbf{x}||_2 = a$) is a hyper-sphere center at the origin with a radius a, the graph of $\sum_{i=1}^{n} |x_i| = a$ (or $||\mathbf{x}||_1 = a$) is a hypercube center at the origin and vertices on the axis, and the graph of $\max_{1\le i\le n} \{|x_i|\} = a$ (or $||\mathbf{x}||_\infty = a$) is a hypercube center at the origin, which faces perpendicular to the axis. The following properties are well known in three dimensional space, but these are also true for higher dimensional spaces.

1. An n-space hypercube has 2n faces, and 2^n corner points (vertices).
2. A vector perpendicular to the hyperplane $\sum_{i=1}^{n} a_i x_i = c$ is $(a_1, a_2, \ldots, a_n)$.
3. The minimum distance from the origin to the hyperplane $\sum_{i=1}^{n} a_i x_i = c$ is $\dfrac{|c|}{\sqrt{\sum_{i=1}^{n} a_i^2}}$. The minimum point is $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ when $x_j = \dfrac{a_j}{\sum_{i=1}^{n} a_i^2} c$, $j = 1, 2, \ldots, n$.

Now we are ready to show the proof of some of the formulas:

Proof of (9): $\|A\|_{21} = \max\limits_{s_i \in S} \{(\sum_{j=1}^{n} (\sum_{i=1}^{m} s_i a_{ij})^2)^{1/2}\}$, where $S = \{-1, 1\}$.

Proof : By Definition (5)

$$\|A\|_{21} = \max_{\|\mathbf{x}\|_2 = 1} \|A\mathbf{x}\|_1 = \max_{\|\mathbf{x}\|_2 = 1} \sum_{i=1}^{m} |\sum_{j=1}^{n} a_{ij} x_j|$$

Consider $\sum_{i=1}^{m} |\sum_{j=1}^{n} a_{ij} x_j| = \sum_{i=1}^{m} s_i \sum_{j=1}^{n} a_{ij} x_j$, where $s_i = \text{sign}(\sum_{j=1}^{n} a_{ij} x_j)$, ……………….(A)

$$= \sum_{j=1}^{n} x_j \sum_{i=1}^{m} s_i a_{ij} \quad \text{……………………..(B)}$$

Since we do not know the sign of $\sum_{j=1}^{n} a_{ij} x_j$ in advance, we therefore have to search for the maximum of (B) over the set $\{(s_1, s_2, \ldots, s_n)\}$, where $s_j \in \{-1, 1\}$ and $\|\mathbf{x}\|_2 = 1$. For each $(s_1, s_2, \ldots, s_n)$ the equation $\sum_{j=1}^{n} x_j \sum_{i=1}^{m} s_i a_{ij} = C$ can be viewed as a hyper-plane in the space $R^{n,}$ and $\|\mathbf{x}\|_2 = 1$ is a unit hyper-sphere center at the origin. The tangent point of the plane to the sphere is the maximum value of C, which occurs when vector $\mathbf{x}$ is perpendicular to the plane. Therefore at this point, we have $x_j = k \sum_{i=1}^{m} s_i a_{ij}$, $j = 1, 2, \ldots, n$. Demanding $\|\mathbf{x}\|_2 = 1$ will yield the value of k, so that

$$x_j = \frac{\sum_{i=1}^{m} s_i a_{ij}}{(\sum_{j=1}^{n} (\sum_{i=1}^{m} s_i a_{ij})^2)^{1/2}}, \; j = 1, 2, \ldots, n.$$

Substitute in (B), we have $C = (\sum_{j=1}^{n} (\sum_{i=1}^{m} s_i a_{ij})^2)^{1/2} = \|\mathbf{s}^T A\|_2$. Maximizing these values over the set $\{(s_1, s_2, \ldots, s_n)\}$, where $s_j \in \{-1, 1\}$ will yield the maximum, as desired. □□

Proof of (10): $\|A\|_{\infty 2} = \max\limits_{x_j \in S} [\sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j)^2]^{1/2}$ where $S = \{-1, 1\}$.

***Proof*:** By Definition (5)

$$\|A\|_{\infty 2} = \max_{\|\mathbf{x}\|_\infty = 1} \|A\mathbf{x}\|_2 = \max_{\|\mathbf{x}\|_\infty = 1} \left(\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2\right)^{1/2}.$$

We claim that the maximum point must lie on one corner of the hypercube $\|x\|_1 = 1$, i.e. all the components of **x** must be 1 or -1. To verify this claim, we will proceed to prove that at the point **z,** not on the corner of that hypercube, the maximum is not attained. Let **x** and **y** be the vectors in R^n, such that all the components are x_j, $y_j \in \{-1, 1\}$, so that $\|\mathbf{x}\|_\infty = \|\mathbf{y}\|_\infty = 1$.
Let $\mathbf{z} = \lambda\mathbf{x} + (1-\lambda)\mathbf{y}$ where $\lambda \in [0, 1]$. We have that $\|\mathbf{z}\|_\infty \le 1$, and

$$\begin{aligned}\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}z_j\right)^2 &= \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}(\lambda x_j + (1-\lambda)y_j\right)^2 \\ &= \sum_{i=1}^{m}\left(\lambda\sum_{j=1}^{n} a_{ij}x_j + (1-\lambda)\sum_{j=1}^{n} a_{ij}y_j\right)^2 \\ &\le \sum_{i=1}^{m}\left(\lambda\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2 + (1-\lambda)\left(\sum_{j=1}^{n} a_{ij}y_j\right)^2\right) \\ &= \lambda\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2 + (1-\lambda)\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}y_j\right)^2\end{aligned}$$

The inequality is valid since the square function is convex. Now, suppose that K = $\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2 \ge \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}y_j\right)^2$, then

$$\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}z_j\right)^2 \le \lambda K + (1-\lambda)K = K = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2.$$

That is, the maximum must be attained at the corner of the unit hypercube. Therefore (10) is valid. □□

Proof of (11): $\|A\|_{2\infty} = \max_{1\le i\le m} \left(\sum_{j=1}^{n} a_{ij}{}^2\right)^{1/2}.$

***Proof*:** By Definition (5)

$$\|A\|_{2\infty} = \max_{\|\mathbf{x}\|_2 = 1} \|A\mathbf{x}\|_\infty = \max_{\|\mathbf{x}\|_2 = 1} \max_{1\le i\le m} \left\{\left|\sum_{j=1}^{n} a_{ij}x_j\right|\right\}.$$

Consider the hyper-plane $\sum_{j=1}^{n} a_{ij}x_j = C_i$, i = 1, 2, . . . , m, $|C_i|$ has a maximum value over the unit sphere $\|x\|_2 = 1$ at the tangent point, which occurs when vector **x** is perpendicular to the plane. Let $x_j = \dfrac{a_{ij}}{\left(\sum_{j=1}^{n} a_{ij}^2\right)^{1/2}}$, j = 1, 2, . . . , n, then we have $\|\mathbf{x}\|_2 = 1$. If we substitute this value in the above expression, we have $\|A\|_{2\infty} = \max_{1\le i\le m} \left\{\left(\sum_{j=1}^{n} a_{ij}{}^2\right)^{1/2}\right\}$, as desired. □□

3. Calculation method:

This section introduces the method to compute the exact value of some matrix norms which required "all combination searches". (See Dhamacharoen [5]).

Firstly, we first introduce the all combination search of n-tuple on the set {0, 1}. For n = 3, we have $2^3 = 8$ combinations, which are listed below:

$$000 \quad 001 \quad 010 \quad 011 \quad 100 \quad 101 \quad 110 \quad 111 \tag{C}$$

This set is a list of the numbers in base two from 000 to 111, which are written in order, and which represent "all combinations" of selection of 2 digits into three fixed positions. The following set is the same set as shown above, but which have been arranged in a different order:

$$000 \quad 100 \quad 110 \quad 010 \quad 011 \quad 111 \quad 101 \quad 001 \tag{D}$$

In the process of digit changing from left to right, in (C) from 000 to 001, the first position is changed, but from 001 to 010 there are two positions changed, and 3 positions changed from 011 to 100. The total number of positions changed is 11, while in (D) the total number of positions changed is 7.

The above example illustrates the importance of ordering the elements in the set. The ordered set in (C) will be called the natural order combination, and for some reason, the ordered set in (D) will be called the "e-2 combination #1".

To generate the e-2 combination #1, begin with the n-tuple of 0's, and then change 1 position each time, in order to obtain all the combinations. Selecting the position each time is important, since all the combinations must be achieved within the least number of operations. In order to do this systematically, we will use a sequence called the "e-2 sequence #1", to point out which position is to be changed. This is sometimes called the leading sequence, since we use it to navigate in order to generate the all combination ordered set. The leading sequence for the e-2 combinations #1 is $\langle 1, 2, 1, 3, 1, 2, 1\rangle$; that is, the position 1, 2, 1, 3, 1, 2, 1 is to be changed in order.

Definition 1: The e-2 string sequence #1 is a sequence in the form:

$$P = \langle p_1, p_2, \ldots, p_n\rangle$$

where p_i is the string generated by the following rule:

$$p_1 = \langle 1\rangle \qquad\qquad b_1 = \langle 1\rangle$$

For i = 2, 3, . . . , n;

$$p_i = \langle i, b_{i-1}\rangle \qquad\qquad b_i = \langle b_{i-1}, p_i\rangle$$

where the notation $\langle b_{i-1}, p_i\rangle$ is the sequence composed of the elements in the strings b_{i-1} and p_i (not the pair of strings b_{i-1} and p_i).

The e-2 sequence #1 is the sequence of terms from the e-2 string sequence.

Theorem 3.1: The e-2 string sequence #1 has some special properties:

1. The string p_i has length 2^{i-1}.
2. Let u_k be the k^{th} term of the e-2 sequence #1. If $k = 2^{p-1}(2q-1)$ for some positive integers p and q, then

$$u_k = p. \tag{13}$$

3. If n is the number of strings in the e-2 string sequence #1, then the number of terms of the e-2 sequence #1 is $2^n - 1$.

The proof can be found in [4], and will be omitted here.

Definition 2: Let S = $\{t_1, t_2\}$. The ordered set of n-tuple $s_i = (s_{i1}, s_{i2}, \ldots, s_{in})$, i = 1, 2, . . . , m, where $s_{ij} \in S$ and $m = 2^n$, is called the e-2 combination. The e-2 combination #1 is the ordered set generated by the rule:

$$s_1 = (t_1, t_1, \ldots, t_1)$$
$$s_{i+1} = P_i(s_i),\ i = 1, \ldots, m\text{-}1.$$

where P_i is the function that changes the component s_{iu_i}, $u_i \in P$ in definition 1, so that it is the other element of S. We also say that the e-2 combination #1 is generated using P as a leading sequence.

Theorem 3.2: The e-2 combination #1 is an ordered set of all combinations, with the least number of transition changes.

The proof can be found in [5]. Note that the number of combinations for selecting the object from the set of two kinds of elements, which are to be put in n different position, is 2^n. From constructing the set, we find that the ordered set has 2^n elements, and the number of changes is $2^n - 1$, which is the least number of changes. Showing that all the elements are different will prove the theorem.

Computation of Norms:

In this section, we will describe the algorithm to calculate the value of the norms in (7), (9) and (10) using the e-2 combination #1. Since the values are in absolute value, we only need to compute half of all the combinations; i.e. the combination of (n-1)-tuple are needed. This reduces the number of e-2 combination #1 to be 2^{n-1}, instead of 2^n.

If, for each k = 1, 2, . . . , 2^{n-1}, we calculate the value of $M_i = \sum_{j=1}^{n} a_{ij}s_j$, i = 1, 2, . . . , m, using this formula, the number of multiplications is mn, and the number of additions is m(n – 1) + m - 1. Then for all k, it will do mn 2^{n-1} multiplications, and $(mn - 1)2^{n-1}$ additions.

Using the e-2 combination #1, we can compute such value using less effort. In computing $M_i = \sum_{j=1}^{n} a_{ij}s_j$ for all i, where $s_j \in S$, we first generate the e-2 sequence #1 and start with the value of $M_i = \sum_{j=1}^{n} a_{ij}$. Then for k = 1, 2, . . . , $2^{n-1} - 1$, let j be the k^{th} term of the sequence, then $s_j = -s_j$, and we compute as follows:

$$M_i = M_i + 2s_ja_{ij}$$

i = 1, 2, . . . , m. (The value $2a_{ij}$ can be set prior, in order to avoid repetition of calculation.)

Using this method, the number of calculations will be 2mn + $m(2^{n-1} - 1)$ multiplications, and mn – 1 + $(2m - 1)(2^{n-1} - 1)$ additions.

Now, we calculate (7), (9) and (10) using the e-2 combination #1.

Algorithm 1 (For (7)): $\|A\|_{\infty 1} = \max_{x_j \in S} \{ \sum_{i=1}^{m} | \sum_{j=1}^{n} a_{ij}x_j | \}$ where S = {-1, 1}.

Given m, n and a_{ij}, i = 1, 2, . . . , m; j = 1, 2, . . . , n. Let $N = 2^{n-1}$.

1. Generate the e-2 sequence #1 ⟨1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, . . . , 1⟩ of length $2^{n-1} - 1$.
2. For j = 1, 2, . . . , n, Let $x_j = 1$.
3. Let $b_{ij} = 2a_{ij}$, i = 1, 2, . . . , m; j = 1, 2, . . . , n.
4. For i = 1, 2, . . . , m, compute $M_{i1} = \sum_{j=1}^{n} a_{ij}$.
5. Compute $sum_1 = \sum_{i=1}^{m} |M_{i1}|$
6. For k = 1, 2, 3, . . . , N-1, let u_k be the k^{th} term of the sequence in 1; let $j = n - u_k + 1$.
   6.1 Let $x_j = -x_j$
   6.2 For i = 1, 2, . . . , m
   $M_{ik} = M_{ik} + x_j b_{ij}$,
   6.3 Compute $sum_{k+1} = \sum_{i=1}^{m} |M_{ik}|$.
7. Let Max = maximum$\{sum_1, sum_2, . . . , sum_N\}$.

Then $||A||_{\infty 1}$ = Max. □

Algorithm 2 (For (9)): $||A||_{21} = \max_{s_i \in S} \{(\sum_{j=1}^{n}(\sum_{i=1}^{m} s_i a_{ij})^2)^{1/2}\}$, where S = {-1, 1}.

Let $N = 2^{n-1}$. Given a_{ij}, i = 1, 2, . . . , m; j = 1, 2, . . . , n.

1. Generate the e-2 sequence #1 ⟨1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, . . . , 1⟩ of length $2^{m-1} - 1$.
2. For i = 1, 2, . . . , m, Let $x_i = 1$.
3. Let $b_{ij} = 2a_{ij}$, i = 1, 2, . . . , m; j = 1, 2, . . . , n.
4. For j = 1, 2, . . . , n, compute $M_{1j} = \sum_{i=1}^{m} a_{ij}$.
5. Compute $sum_1 = (\sum_{j=1}^{n} M_{1j}^2)^{1/2}$
6. For k = 1, 2, 3, . . . , N-1, let u_k be the k^{th} term of the sequence in 1; let $i = n - u_k + 1$.
   6.1 Let $x_i = -x_i$
   6.2 For j = 1, 2, . . . , n
   $M_{kj} = M_{kj} + x_i b_{ij}$,
   6.3 Compute $sum_{k+1} = (\sum_{j=1}^{n} M_{k+1,j}^2)^{1/2}$
7. Let Max = maximum$\{sum_1, sum_2, . . . , sum_N\}$.

Then $||A||_{21}$ = Max. □□

Algorithm 3 (For (10)): $\|A\|_{\infty 2} = \max_{x_j \in S} [\sum_{i=1}^{m}(\sum_{j=1}^{n} a_{ij}x_j)^2]^{1/2}$ where S = {-1, 1}.

Let $N = 2^{n-1}$. Given a_{ij}, i = 1, 2, . . . , m; j = 1, 2, . . . , n.

1. Generate the e-2 sequence #1 ⟨1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, . . . , 1 ⟩ of length $2^{n-1} - 1$.
2. For j = 1, 2, . . . , n, Let $x_j = 1$.
3. Let $b_{ij} = 2a_{ij}$, i = 1, 2, . . . , m; j = 1, 2, . . . , n.
4. For i = 1, 2, . . . , m, compute $M_{i1} = \sum_{j=1}^{n} a_{ij}$.
5. Compute $sum_1 = (\sum_{i=1}^{m} M_{i1}^2)^{1/2}$
6. For k = 1, 2, 3, . . . , N-1, let u_k be the k^{th} term of the sequence in 1; let $j = n - u_k + 1$.
   - 6.1 Let $x_j = -x_j$
   - 6.2 For i = 1, 2, . . . , m
     $M_{ik} = M_{ik} + x_j b_{ij}$,
   - 6.3 Compute $sum_{k+1} = (\sum_{i=1}^{m} M_{ik+1}^2)^{1/2}$
7. Let Max = maximum{sum_1, sum_2, . . . , sum_N}.

Then $\|A\|_{\infty 2}$ = Max. □

Example: Let A be a 3 × 2 matrix, B a 3×4 matrix, given by:

$$A = \begin{bmatrix} -1 & 3 \\ 4 & 1 \\ 2 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 2 & -1 & 0 & 3 \\ 1 & 1 & -2 & 2 \\ -3 & 1 & -2 & -2 \end{bmatrix}.$$

By Algorithm 1., $\|A\|_{\infty 1} = 10$ $\|B\|_{\infty 1} = 14$
By Algorithm 2., $\|A\|_{21} = 7.61577$ $\|B\|_{21} = 9.273619$
By Algorithm 3., $\|A\|_{\infty 2} = 5.83095$ $\|B\|_{\infty 2} = 10$

4. Conclusion:

The induced norms for a matrix of linear transformation between finite dimensional spaces were defined, and their computation formulas were derived. The formulas in (10) and (11) are believed to have first appeared here in this paper. The formulas (9), (10) and (11) can be derived using the concepts of cube, plane and sphere from analytic geometry. Computations of some norms are generally difficult, since they involve the all combination search. Here, we have generated an ordered set of all the combinations in a special way, so that the computation following that set can provide an exact solution, which will require the least amount of computation.

Acknowledgments

The author would like to thank the referees for their comments. This research was supported by the Research Fund from the National Research Council of Thailand

Reference

[1] Lancaster, P. and M. Tismenetsky, "The Theory of Matrices, second edition with Applications", Academic Press (1985).

[2] Drakakis, K., On the Calculation of the $l_2 \to l_1$ Induced Matrix Norm, *International Journal of Algebra*, Vol. 3, 2009, no. 5, 231 – 240.

[3] Rohn, J., Computing the Norm ||a||∞.1 is NP-Hard. *Linear and Multilinear Algebra*, 47, 2000.

[4] Higham, N., Estimating the matrix p-norm. Numerische Mathematik, 62(1):539-555, Dec 1992. doi: 10.1007/BF01396242.

[5] Dhamacharoen, A., The Minimum Cost Sequencing of All-Combination, *will be appeared.*

ภาคผนวก ค.

ประวัตินักวิจัยและคณะ

ส่วน ค ***: ประวัติคณะผู้วิจัย***

1. ชื่อ – นายอำพล ธรรมเจริญ (ภาษาอังกฤษ) Mr. Ampon Dhamacharoen
2. เลขหมายบัตรประจำตัวประชาชน 3 2001 00615 23 9
3. ตำแหน่งปัจจุบัน รองศาสตราจารย์
4. หน่วยงานและสถานที่อยู่ที่ติดต่อได้สะดวก พร้อมหมายเลขโทรศัพท์ โทรสาร และไปรษณีย์อิเล็กทรอนิกส์ (e-mail)

   ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา บางแสน ชลบุรี 20131

   โทรศัพท์ 038 10 3048 E-mail ampon@buu.ac.th
5. ประวัติการศึกษา

ระดับปริญญา	อักษรย่อปริญญาและชื่อเต็ม	ปีที่จบการศึกษา	ชื่อสถาบันการศึกษา	ประเทศ
ปริญญาตรี	กศ.บ. (คณิตศาสตร์)	2514	วศ. ประสานมิตร	ไทย
ปริญญาโท	กศ.ม. (คณิตศาสตร์)	2516	มศว. ประสานมิตร	ไทย
	M.S. (Mathematics)	1983	North Carolina State U.	USA
ปริญญาเอก	Ph.D. (Applied Math.)	1986	North Carolina State U.	USA

6. ประสบการณ์ที่เกี่ยวข้องกับงานวิจัยทั้งหมด

6.1 งานที่ทำเสร็จแล้ว

1. "การแก้ปัญหาการเปลี่ยนวงโคจร โดยใช้แรงเริ่มต้นครั้งเดียวโดยวิธีของนิวตัน" เสนอในการประชุมทางวิชาการคณิตศาสตร์ ปี 2530 ที่มหาวิทยาลัยสงขลานครินทร์ หาดใหญ่ จ.สงขลา
2 . "Controllability of Positive Discrete Linear Control Systems with Bounded Inputs", Srinakharinwirot University Science Journal, Volume 3 No. 2 December l987, Bangkok, Thailand.
3. "Solving Positive Discrete Linear Control Systems Using a Method of Nonlinear Programming", Proceedings, The Sixth Southeast Asian Statistics Seminar, 18-20 September 1987, Bangkok Thailand.
4. "ความยุติธรรมและความเหมาะสมของข้อสอบแบบเลือกตอบที่ใช้ในการสอบแข่งขัน" รายงานการวิจัย มหาวิทยาลัยบูรพา 2534

5. “The Distributions of Real-Score Obtained by Multiple-Choice Test when Guessing is Permissible”, Proceedings, The International Conference on Applied Mathematics, Institut Teknologi Bandung, Indonesia, 1992.
6. “Using Quadratic Programming to Solve the Real-Score Problem”, Proceedings, The International Conference on Applied Mathematics, Institute Teknologi Bandung, Indonesia, 1992.
7. “Commutation of the Inverse-Pairs Matrices”. Presented at the Thailand Annual Mathematical Conference, l994, Suranaree University of Technology, Nakhon-Rachasima, Thailand.
8. Dhamacharoen, A., “An Exact Line Search Algorithm for Linear Minimax Problems” Proceedings, การประชุมวิชาการด้านการวิจัยดำเนินงาน ประจำปี พ.ศ. 2549, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์ 2549.
9. Dhamacharoen, A., "Steepest Descent Method for the Linear Minimax Problems", Thai Journal of Mathematics, Special issue (Annual meeting in Mathematics, 2007) page 175-190.
10. Dhamacharoen, A., “Round Robin Tournament Scheduling and Test Case Generating”, Proceedings, The 5th International Joint Conference on Computer Science and Software Engineering (JCSSE2008), Silpakorn University, Thailand, 2008, Page 414 - 418.
11. Dhamacharoen, A., “Two-Stage Iteration Method in Solving Systems of Nonlinear Equations”, Proceedings, The 15th Annual Meeting in Mathematics, Bangkok, Annu. (2010) 6-8.
12. Dhamacharoen, A., and P. Kasempin “The Negative Inner Product Sets", East-West Journal of Mathematics Volume 12, Number 2, December 2010, Page 197 – 205.
13. Chompuvised, K., and A. Dhamacharoen "Solving Boundary Value Problems of Ordinary Differential Equations with Non-Separated Boundary Conditions", Applied Mathematics and Computation, Volume 217, Issue 24, 15 August 2011, Pages 10355-10360. (Impact factor 1.534)
14. Dhamacharoen, A., and K. Chompuvised, " An efficient method for solving multipoint equation boundary value problems", World Academy of Science, Engineering and Technology 75 2013
15. Dhamacharoen, A., "An Efficient Hybrid Method for Solving Systems of Nonlinear Equations" Journal of Computational and Applied Mathematics, 263 (2014) 59 – 68 (Impact factor 0.95)

16. Dhamacharoen, A., and P. Kasempin, "On Derivation of Rational Solutions of Babbage's Functional Equation" International Journal of Pure and Applied Mathematics, Volume 94, No. 1 2014, 9 – 21.

6.2 งานวิจัยที่กำลังทำ (ทำเสร็จแล้ว กำลังเสนอบทความเพื่อลงพิมพ์ในวารสารวิชาการ)

1. Dhamacharoen, A., “An Efficient Method for Solving Integro-Differential Equations Boundary Value Problems”
2. Dhamacharoen, A., " Minimum Cost Sequencing of All Combinations"
3. Dhamacharoen, A., " Exact Calculation of Some Induced Matrix Norms"
4. Dhamacharoen, A., "Some Numerical Methods for solving Differential Algebraic Equation."

รายงานสรุปการเงิน

เลขที่โครงการระบบบริหารงานวิจัย (NRMS 0000) สัญญาเลขที่ ๓๔/๒๕๕๖

โครงการวิจัยโครงการวิจัยประเภทงบประมาณเงินรายได้

จากเงินอุดหนุนรัฐบาล (งบประมาณแผ่นดิน)

ประจำปีงบประมาณ พ.ศ. ๒๕๕๖

มหาวิทยาลัยบูรพา

ชื่อโครงการวิจัย เรื่อง ลำดับของสตริงที่มีความยาวแปรเปลี่ยน และการประยุกต์

ภาษาอังกฤษ Variable-length String Sequences and Applications

ชื่อหัวหน้าโครงการวิจัยผู้รับทุน ร.ศ. ดร. อำพล ธรรมเจริญ

รายงานในช่วงตั้งแต่วันที่ ๑๗ ตุลาคม พ.ศ. ๒๕๕๕ ถึงวันที่ ๒๕ สิงหาคม พ.ศ. ๒๕๕๘

ระยะเวลาดำเนินการ ๑ ปี (ขยายเวลา ๒ ครั้ง) ตั้งแต่วันที่ ๑๗ ลาคม พ.ศ. ๒๕๕๕ ถึงวันที่ ๒๕ สิงหาคม พ.ศ. ๒๕๕๘

รหัสโครงการ ๓๔/๒๕๕๖ สัญญาเลขที่ ๓๔/๒๕๕๖

ได้รับงบประมาณรวมทั้งสิ้น ๑๒๖,๔๐๐ บาท (หนึ่งแสนสองหมื่นหกพันสี่ร้อยบาทถ้วน) ระยะเวลาทำงาน ๓ ปี (ระหว่างวันที่ ๑ ตุลาคม พ.ศ. ๒๕๕๖ ถึงวันที่ ๒๐ กันยายน พ.ศ. ๒๕๕๖)

<u>รายรับ</u>

จำนวนเงินที่ได้รับ

งวดที่ ๑ (๕๐%)		๖๓,๒๐๐ บาท เมื่อวันที่ ๒๑ พฤศจิกายน พ.ศ. ๒๕๕๕
งวดที่ ๒ (๔๐%)		๕๐,๕๖๐ บาท เมื่อวันที่ ๒๓ กันยายน พ.ศ. ๒๕๕๖
งวดที่ ๓ (๑๐%)		--- บาท เมื่อวันที่ ...
	รวม	๑๑๓,๗๖๐ บาท

<u>รายจ่าย</u>

รายการ	งบประมาณที่ตั้งไว้	งบประมาณที่ใช้จริง	จำนวนเงินคงเหลือ /เกิน
1. ค่าตอบแทน	25,000	25,000	-
2. ค่าจ้าง	40,000	40,000	-
3. ค่าวัสดุ	29,200	26,560	2,640
4. ค่าใช้สอย	32,200	22,200	10,000
5. ค่าครุภัณฑ์			
6. ค่าใช้จ่ายอื่น ๆ			
รวม	126,400	113,760	12,640

(..)

ลงนามหัวหน้าโครงการวิจัยผู้รับทุน

วันที่ 25 เดือนสิงหาคม พ.ศ. 2558