



ใบรับรองวิทยานิพนธ์  
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรดุษฎีบัณฑิต (วิศวกรรมคอมพิวเตอร์)

ปริญญา

วิศวกรรมคอมพิวเตอร์

วิศวกรรมคอมพิวเตอร์

สาขา

ภาควิชา

เรื่อง

ขั้นตอนวิธีสำหรับการค้นหาแบบช่วงและการวางแผนการส่งข้อมูลสำหรับการส่งข้อมูล  
แบบกระแสนเครือข่ายเพียร์ทูเพียร์

Algorithms for Range Queries and Content Delivery Planning for Peer-to-Peer Streaming

นามผู้วิจัย

นายจักริน ชวชาติ

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

( ผู้ช่วยศาสตราจารย์จักร์ทัศน์ ฝักเจริญผล, Ph.D. )

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

( ผู้ช่วยศาสตราจารย์ชัยพร ใจแก้ว, Ph.D. )

หัวหน้าภาควิชา

( ผู้ช่วยศาสตราจารย์ภูษงค์ อุทโยภาส, Ph.D. )

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

( รองศาสตราจารย์กัญญา ชีระกุล, D.Agr. )

คณบดีบัณฑิตวิทยาลัย

วันที่

เดือน

พ.ศ.

วิทยานิพนธ์

เรื่อง

ขั้นตอนวิธีสำหรับการค้นหาแบบช่วงและการวางแผนการส่งข้อมูลสำหรับการส่งข้อมูลแบบกระแสน  
เครือข่ายเพียร์ทูเพียร์

Algorithms for Range Queries and Content Delivery Planning for Peer-to-Peer Streaming

โดย

นายจักริน ขวชาติ

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต (วิศวกรรมคอมพิวเตอร์)

พ.ศ. 2556

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

จักริน ชวชาติ 2556: ขั้นตอนวิธีสำหรับการค้นหาแบบช่วงและการวางแผนการส่งข้อมูล  
สำหรับการส่งข้อมูลแบบกระแสนเครือข่ายเพียร์ทูเพียร์ ปรินญาวิศวกรรมศาสตรดุษฎีบัณฑิต  
(วิศวกรรมคอมพิวเตอร์) สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ อาจารย์ที่  
ปรึกษาวิทยานิพนธ์หลัก: ผู้ช่วยศาสตราจารย์จิรทัศน์ ฝักเจริญผล, Ph.D. 96 หน้า

วิทยานิพนธ์นี้ศึกษาปัญหาสองปัญหาที่เกี่ยวข้องกับการส่งข้อมูลแบบกระแสนเครือข่ายเพียร์  
ทูเพียร์ งานวิจัยส่วนแรกสนใจการวางแผนการส่งข้อมูลโดยมีเงื่อนไขว่าความสามารถในการส่งต่อ  
ข้อมูลของแต่ละเครื่องนั้นแตกต่างกันได้ งานวิจัยมีจุดประสงค์เพื่อออกแบบแผนการส่งข้อมูลที่ส่ง  
ข้อมูลจากเครื่องต้นทางไปยังทุกเครื่องได้รวดเร็วที่สุด เราพิจารณาปัญหาในกรณีที่แบนด์วิดท์ของเส้น  
เชื่อมทุกเส้นมีค่าเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการและพิสูจน์ว่าในกรณีดังกล่าวมีแผนการส่ง  
ข้อมูลที่ดีที่สุดที่เป็นต้นไม้อ่าง เราลดรูปปัญหานี้ไปยังปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่  
ต่ำที่สุดและดิกกรีมีขอบเขตในกรณีที่ดิกกรีไม่เท่ากัน และเสนอขั้นตอนวิธีซึ่งเป็นส่วนขยายของขั้นตอน  
วิธีของ Könemann *et al.* ซึ่งทำงานได้เฉพาะกรณีที่ขอบเขตของดิกกรีเท่ากันทั้งหมด เราได้พิสูจน์ว่าค่า  
ตอบที่ได้เป็นแผนการส่งข้อมูลที่มีความล่าช้าไม่เกิน  $O(\sqrt{\log n})$  เท่าของแผนการส่งข้อมูลที่ดีที่สุด เมื่อ  
 $n$  เป็นจำนวนโหนดในกราฟ

ในงานวิจัยส่วนที่สองเราพิจารณาการค้นหาแบบช่วง แม้ว่าในเครือข่ายเพียร์ทูเพียร์จะนิยม  
ใช้ตารางแฮชแบบกระจายเพื่อเก็บและค้นหากุญแจ แต่การแฮชทำลายคุณสมบัติท้องถิ่นของกุญแจ  
การค้นหาแบบช่วงจึงกระทำได้ลำบาก ในอีกแนวทางหนึ่งที่หลีกเลี่ยงการแฮชนั้น Ganesan *et al.*  
ได้นำเสนอวิธีการปรับดูแลการะของระบบที่เก็บและค้นหากุญแจบนระบบเพียร์ทูเพียร์ที่ทำงานบน  
ปริภูมิกุญแจโดยตรง ขั้นตอนวิธีที่เสนอรับประกันว่าอัตราส่วนความไม่สมดุลจะมีค่าไม่เกิน 4.237 เมื่อ  
อัตราส่วนความไม่สมดุลคืออัตราส่วนระหว่างภาระสูงสุดต่อต่ำสุดในเครือข่าย และมีค่าใช้จ่ายเฉลี่ย  
ในการดำเนินการเป็นค่าคงที่ อย่างไรก็ตามขั้นตอนวิธีดังกล่าวเป็นขั้นตอนวิธีแบบเรียกตัวเองซึ่งมีข้อ  
จำกัดหลายประการเมื่อนำไปประยุกต์ใช้งานจริง เราเสนอขั้นตอนวิธีการปรับดูแลการะที่ง่ายขึ้นและ  
ไม่มีการเรียกตัวเอง ที่รับประกันอัตราส่วนความไม่สมดุลไม่เกิน 7.464 และมีค่าใช้จ่ายเฉลี่ยในการ  
ดำเนินการเป็นค่าคงที่

Jakarin Chawachat 2013: Algorithms for Range Queries and Content Delivery Planning for Peer-to-Peer Streaming. Doctor of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Assistant Professor Jittat Fakcharoenphol, Ph.D. 96 pages.

This thesis considers two important problems in Peer-to-Peer streaming. The first part of this thesis considers content delivery planning where peers in the network have diverse uplink bandwidths. The objective of this problem is to find a streaming scheme that minimizes the maximum delivery time from the source to any peers. We consider the special case of this problem where all bandwidths are multiples of the required bandwidth and prove that there exists an optimal solution which is a tree. Using this fact, the problem in this case naturally reduces to the Bounded Degree Minimum Diameter Spanning Tree problem with Non-Uniform Degree Bounds. We propose an algorithm for this problem which is an extension of Könemann *et al.*'s algorithm that works only uniform degree bounds. The solution obtained from our algorithm is a streaming scheme whose delays are at most  $O(\sqrt{\log n})$  times the optimal delay where  $n$  is the number of nodes in the network.

In the second part, we consider range queries. Peer-to-Peer networks usually employ distributed hash tables to handle data queries. However, hashing destroys the locality property of object keys, the critical properties to range queries. Ganesan *et al.* avoid the hashing approach and introduce a load-balancing algorithm that works with a system that searches directly on the key space. Their algorithm guarantees an imbalance ratio of 4.237 where the imbalance ratio is the ratio between the maximum and minimum load in the network, while using only a constant amortized cost per operation. However, the proposed algorithm is recursive, which is undesirable in practical settings. We present a simpler and non-recursive load-balancing algorithm which guarantees an imbalance ratio of 7.464 with constant amortized costs per operation.

----- / ----- / -----  
Student's signature                      Thesis Advisor's signature

## กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.จิตรัทธน์ ฝักเจริญผล ประธานกรรมการที่ปรึกษาวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.ชัยพร ใจแก้ว กรรมการที่ปรึกษาสาขาวิชาเอกที่ให้คำปรึกษาในการเรียนและการค้นคว้าวิจัยตลอดจนการตรวจแก้วิทยานิพนธ์จนกระทั่งเสร็จสมบูรณ์

ขอกราบขอบพระคุณ อาจารย์ภาค วิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ได้อบรมสั่งสอนและมอบความรู้อันเป็นประโยชน์

ด้วยความดีหรือประโยชน์อันใดเนื่องจากวิทยานิพนธ์เล่มนี้ ขอมอบแด่คุณพ่อ คุณแม่ คุณอา ที่ได้อบรมและให้กำลังใจผู้วิจัยมาตลอดในทุกเรื่อง

จักริน ชวชาติ

กุมภาพันธ์ 2556

## สารบัญ

## หน้า

|                            |     |
|----------------------------|-----|
| สารบัญ                     | (1) |
| สารบัญภาพ                  | (2) |
| คำนำ                       | 1   |
| วัตถุประสงค์               | 9   |
| การตรวจเอกสาร              | 10  |
| อุปกรณ์และวิธีการ          | 26  |
| อุปกรณ์                    | 26  |
| วิธีการ                    | 26  |
| ผลและวิจารณ์               | 34  |
| สรุปและข้อเสนอแนะ          | 88  |
| เอกสารและสิ่งอ้างอิง       | 91  |
| ประวัติการศึกษาและการทำงาน | 96  |

## สารบัญภาพ

| ภาพที่                                                                     | หน้า |
|----------------------------------------------------------------------------|------|
| 1 ตัวอย่างแผนการส่งข้อมูล                                                  | 5    |
| 2 ตัวอย่างการแบ่งช่วงกุญแจของแต่ละ โหนดรับผิดชอบ                           | 7    |
| 3 ตัวอย่างเครือข่าย                                                        | 11   |
| 4 ตัวอย่างการทำงานของตารางแฮชแบบกระจาย                                     | 12   |
| 5 ตัวอย่างที่ขัดแย้งกับเงื่อนไขของ Huang <i>et al.</i> (2009)              | 16   |
| 6 ตัวอย่างการทำงานของขั้นตอนวิธี FHK                                       | 19   |
| 7 ตัวอย่างกุญแจเมื่อผ่านแฮชฟังก์ชัน                                        | 20   |
| 8 การดำเนินการ NbrAdjust และ Reorder                                       | 23   |
| 9 ขั้นตอนวิธี ADJUSTLOAD ในกรณีการใส่กุญแจ                                 | 25   |
| 10 ขั้นตอนวิธีสำหรับ BDST with non-uniform degree bound                    | 29   |
| 11 ตัวอย่างการทำงานของ BDST with non-uniform degree bound                  | 30   |
| 12 การดำเนินการ MINBALANCE                                                 | 32   |
| 13 การดำเนินการ SPLIT                                                      | 33   |
| 14 ตัวอย่างการลดรูป MDM problem                                            | 36   |
| 15 ตัวอย่างการแบ่งต้นไม้ออกเป็นเส้น 10 เส้น                                | 42   |
| 16 ตัวอย่างของการเชื่อม component $A$ และ $B$                              | 45   |
| 17 ตัวอย่างกรณีที่ทุกโหนดมีดีกรีไม่เกินสอง                                 | 50   |
| 18 ตัวอย่างกรณีที่มีบางโหนดมีดีกรีเกินสอง                                  | 51   |
| 19 ตัวอย่างเหตุการณ์ min-transfer ที่เกิดกับโหนด $z$ ทางด้านซ้ายและด้านขวา | 57   |
| 20 ตัวอย่างเหตุการณ์ min-transfer ครั้งที่ $(k + 1)$ ในรูปแบบต่างๆ         | 63   |
| 21 เซตของโหนด $P$ ที่อยู่ระหว่างโหนด $x$ และโหนด $v_{i+l+1}$               | 78   |

# ขั้นตอนวิธีสำหรับการค้นหาแบบช่วงและการวางแผนการส่งข้อมูล สำหรับการส่งข้อมูลแบบกระแสนเครือข่ายเพียร์ทูเพียร์

## Algorithms for Range Queries and Content Delivery Planning for Peer-to-Peer Streaming

### คำนำ

ในปัจจุบันมีการใช้งานอินเทอร์เน็ตอย่างกว้างขวาง ทำให้การติดต่อสื่อสารหรือการเผยแพร่ข้อมูลไปยังผู้ใช้งานที่อาศัยอยู่ในสถานที่ต่างกันทำได้สะดวกและรวดเร็ว สำหรับสื่อที่ถูกเผยแพร่ในอินเทอร์เน็ตนั้นมีมากมายหลายประเภทไม่ว่าจะเป็นภาพ เสียง วิดีทัศน์ ฯลฯ ในงานวิจัยฉบับนี้เราสนใจสื่อแบบกระแส (Streaming media) สื่อแบบกระแสเป็นสื่อประเภทหนึ่งที่ได้รับคามนิยมนอย่างมากในปัจจุบัน เนื่องจากเป็นสื่อประเภทที่เมื่อผู้ใช้งานได้รับข้อมูลแล้วสามารถแสดงผลได้ทันที แม้ว่าขณะนั้นผู้ใช้งานยังคงดาวน์โหลดข้อมูลจากผู้บริการอยู่ก็ตาม ตัวอย่างโปรแกรมประยุกต์ที่นำเอาสื่อแบบกระแสไปใช้งานเช่น โปรแกรมประยุกต์สำหรับการถ่ายทอดสดเหตุการณ์ต่างๆ โปรแกรมประยุกต์สำหรับการประชุมทางไกล (Teleconference) โปรแกรมประยุกต์บนเว็บ (Web application) ตัวอย่างเช่นเว็บไซต์ [www.youtube.com](http://www.youtube.com) ฯลฯ

ในการส่งข้อมูลแบบกระแสนี้มีการใช้สถาปัตยกรรมในการส่งข้อมูลหลากหลายรูปแบบ เริ่มแรกรูปแบบการเชื่อมต่อมีลักษณะเป็นเครือข่ายแบบรับ-ให้บริการ (Client-Server network) เครือข่ายแบบนี้ประกอบด้วยเครื่องบริการ (Server) และเครื่องรับบริการ (Client) เครื่องบริการเป็นเครื่องที่เก็บและส่งข้อมูลทุกอย่าง เมื่อเครื่องรับบริการต้องการข้อมูลสื่อแบบกระแส เครื่องรับบริการจะต้องติดต่อกับเครื่องบริการเพื่อขอข้อมูล จากนั้นเครื่องบริการจึงจะส่งข้อมูลให้กับเครื่องรับบริการที่ร้องขอข้อมูล ทั้งนี้รูปแบบการส่งข้อมูลแบบกระแสนี้ใช้สถาปัตยกรรมแบบนี้ มีข้อเสียหลายอย่างเช่น ระบบมี Single Point of Failure กล่าวคือถ้าเครื่องบริการเสียระบบจะไม่สามารถทำงานต่อไปได้ การติดตั้งระบบหรือเพิ่มเครื่องบริการมีค่าใช้จ่ายในการติดตั้งที่สูง เป็นต้น

สถาปัตยกรรมในการส่งข้อมูลแบบกระแสนี้ในงานวิจัยฉบับนี้สนใจได้แก่ รูปแบบการเชื่อมต่อที่มีลักษณะเป็นเครือข่ายแบบเพียร์ทูเพียร์ (Peer-to-Peer network) เครือข่ายแบบเพียร์ทูเพียร์นั้นไม่มีเครื่อง

บริการ แต่ละเครื่องในเครือข่ายมีสิทธิเท่าเทียมกันและสามารถเป็นได้ทั้งเครื่องที่รับและส่งต่อข้อมูล รูปแบบการเชื่อมต่อแบบนี้ทนทานต่อการเสียของฮาร์ดแวร์ นั่นคือแม้ว่าเครื่องต้นกำเนิดข้อมูลเสีย แต่ถ้ามีเครื่องในระบบบางเครื่องที่ได้รับข้อมูลแล้ว เครื่องที่มีข้อมูลเหล่านั้นสามารถส่งข้อมูลต่อแทนเครื่องต้นกำเนิดข้อมูลได้ ทำให้ระบบสามารถส่งข้อมูลนั้นต่อไปได้

เมื่อนำเอารูปแบบการเชื่อมต่อที่มีลักษณะเป็นเครือข่ายแบบแบบเพียร์ทูเพียร์มาใช้ในการส่งข้อมูลแบบกระจาย พบว่ามีปัญหาวิจัยที่น่าสนใจหลายปัญหาเช่น ปัญหาการค้นหาข้อมูล (Content searching) ปัญหาการออกแบบแผนการส่งข้อมูล (Content delivery) ปัญหาด้านความปลอดภัย (Security) ปัญหาการจัดการกำหนดการ (Scheduling) ปัญหาโหลดเข้าระบบออกระบบ (Dynamic) ปัญหาความเป็นธรรมในการรับส่งข้อมูล (Fairness) ฯลฯ

ในงานวิทยานิพนธ์ฉบับนี้เราสนใจปัญหาที่สำคัญที่เกี่ยวข้องกับการส่งข้อมูลแบบกระจายบนเครือข่ายเพียร์ทูเพียร์ 2 ปัญหาได้แก่ (1) ปัญหาการวางแผนการส่งข้อมูลแบบกระจายบนเครือข่ายเพียร์ทูเพียร์ และ (2) ปัญหาการออกแบบวิธีการปรับคุณภาพบนเครือข่ายเพียร์ทูเพียร์เพื่อรองรับการค้นหาแบบช่วง

เราจะกล่าวถึงวิทยานิพนธ์ในส่วนแรกซึ่งเกี่ยวกับปัญหาการวางแผนการส่งข้อมูลแบบกระจายบนเครือข่ายเพียร์ทูเพียร์ก่อน วิทยานิพนธ์ในส่วนแรกนี้มีจุดมุ่งหมายคือต้องการออกแบบแผนการส่งข้อมูลแบบกระจายที่ส่งข้อมูลจากเครื่องต้นทาง (Source) ไปยังทุกเครื่องในระบบได้รวดเร็วที่สุด วิทยานิพนธ์ส่วนแรกนี้มีเงื่อนไขว่าแต่ละเครื่องในเครือข่ายมีความสามารถในการส่งต่อข้อมูลได้แตกต่างกัน

เริ่มแรกการส่งข้อมูลแบบกระจายนั้นใช้การเชื่อมต่อแบบรับ-ให้บริการ การเชื่อมต่อแบบนี้มีโครงสร้างที่ง่าย นั่นคือเครื่องรับบริการทุกเครื่องเชื่อมต่อโดยตรงกับเครื่องบริการ แต่การเชื่อมต่อแบบนี้ไม่สามารถรองรับเครื่องรับบริการจำนวนมากได้ เนื่องจากโดยทั่วไปแล้วเครื่องบริการส่วนใหญ่มีความสามารถในการให้บริการที่จำกัด ในงานวิจัยในส่วนแรกเราจึงสนใจนำเอาการเชื่อมต่อแบบเครือข่ายเพียร์ทูเพียร์มาใช้ในการส่งข้อมูลแบบกระจาย เนื่องจากในเครือข่ายเพียร์ทูเพียร์ถ้าเครื่องต้นทางส่งข้อมูลให้กับเครื่องในระบบจำนวนหนึ่งแล้ว เครื่องที่ได้รับข้อมูลเหล่านั้นสามารถช่วยส่งข้อมูลต่อไปยังเครื่องอื่นๆ ในระบบต่อไปได้ การที่สามารถทำเช่นนี้ได้ทำให้มีประสิทธิภาพในการส่งข้อมูลมากกว่าการเชื่อมต่อแบบเครือข่ายแบบรับ-ให้บริการ

การส่งข้อมูลแบบกระแสนเครือข่ายเพียร์ทูเพียร์มีหลายปัจจัยที่ต้องคำนึงถึง สำหรับปัจจัยในการรับส่งข้อมูลที่วิทยานิพนธ์ส่วนแรกนี้สนใจมี 2 ปัจจัยได้แก่ ปัจจัยแรกคือความล่าช้าจากเครื่องบริการที่ต่ำ (Low delay) ซึ่งเป็นหนึ่งในเกณฑ์ทางคุณภาพที่นำมาใช้วัดคุณภาพของข้อมูลในโปรแกรมประยุกต์สำหรับถ่ายทอดสดสื่อแบบกระแส ปัจจัยที่สองคือความสามารถในการส่งข้อมูลต่อของเครื่องในระบบ เนื่องจากแต่ละเครื่องในเครือข่ายมีลักษณะที่แตกต่างกัน ดังนั้นการที่แต่ละเครื่องมีความสามารถในการส่งข้อมูลต่อไปยังเครื่องอื่นจึงไม่จำเป็นที่จะต้องเท่ากัน

จากปัจจัยความต้องการความล่าช้าจากเครื่องบริการที่ต่ำและแต่ละเครื่องมีความสามารถในการส่งข้อมูลต่อที่ต่างกัน Ren *et al.* (2008) ได้นิยามปัญหาการส่งข้อมูลที่มีโครงสร้างแบบเมชที่มีความล่าช้าต่ำที่สุด (Minimum Delay Mesh problem (MDM problem)) ขึ้น พวกเขาได้พิสูจน์ว่าปัญหานี้เป็นปัญหา NP-Complete และได้เสนอขั้นตอนวิธีแบบศึกษาสำนึก (Heuristic algorithm) นอกจากนี้พวกเขายังได้ทำการทดลองเพื่อเปรียบเทียบประสิทธิภาพการทำงานเทียบกับงานวิจัยของ Small *et al.* (2007) ซึ่งเป็นงานวิจัยก่อนหน้านี้ที่ใกล้เคียงกัน จากการทดลองพบว่าแผนการส่งข้อมูลจากงานของ Ren *et al.* (2008) มีความล่าช้าเฉลี่ยและความล่าช้าที่มากที่สุดจากเครื่องต้นทางไปยังโหนดใดๆ ต่ำกว่างานวิจัยของ Small *et al.* (2007) อีกทั้งยังพบว่าจำนวน hop เฉลี่ยและจำนวน hop สูงสุดมีจำนวนน้อยกว่าเช่นกัน

หลังจากนั้น Huang *et al.* (2009) ได้ศึกษากรณีพิเศษของ MDM problem ภายใต้อำนาจ The Minimum Delay Peer-to-Peer Streaming problem กรณีพิเศษที่พวกเขาสนใจคือกรณีที่แผนการส่งข้อมูลแบบกระแสนที่ดีที่สุดเป็นต้นไม้ พวกเขาได้ตั้งเงื่อนไขเกี่ยวกับแบนด์วิดท์ของเส้นเชื่อมโยงขึ้น (Uplink bandwidth) และอ้างว่าภายใต้เงื่อนไขดังกล่าวจะมีแผนสำหรับการส่งข้อมูลที่ดีที่สุดเป็นต้นไม้ หรือ เป็นโครงสร้างต้นไม้หลายต้นมารวมกัน (อ้างไว้ในงานวิจัยของพวกเขาในหัวข้อที่ 4.2.4) พวกเขาเสนอขั้นตอนวิธีแบบประมาณซึ่งเป็นการปรับแก้ขั้นตอนวิธีของ Könemann *et al.* (2005) ที่ให้ผลลัพธ์เป็นแผนการส่งข้อมูลที่มีความล่าช้าจากเครื่องต้นทางไปยังเครื่องใดๆ ไม่เกิน  $O(\sqrt{\log n})$  เท่าของคำตอบที่ดีที่สุด

อย่างไรก็ตามเราพบว่าเงื่อนไขที่เกี่ยวกับแบนด์วิดท์ของเส้นเชื่อมโยงขึ้นในงานวิจัยของ Huang *et al.* (2009) นั้นไม่จริง โดยเราจะแสดงตัวอย่างที่ไม่จริงในส่วนการตรวจเอกสาร

สำหรับงานวิจัยในส่วนแรกนี้ เราสนใจกรณีพิเศษของ MDM problem เช่นกัน แต่เราเสนอเงื่อนไขของปัญหาที่ต่างออกไป เงื่อนไขของวิทยานิพนธ์ส่วนแรกคือ

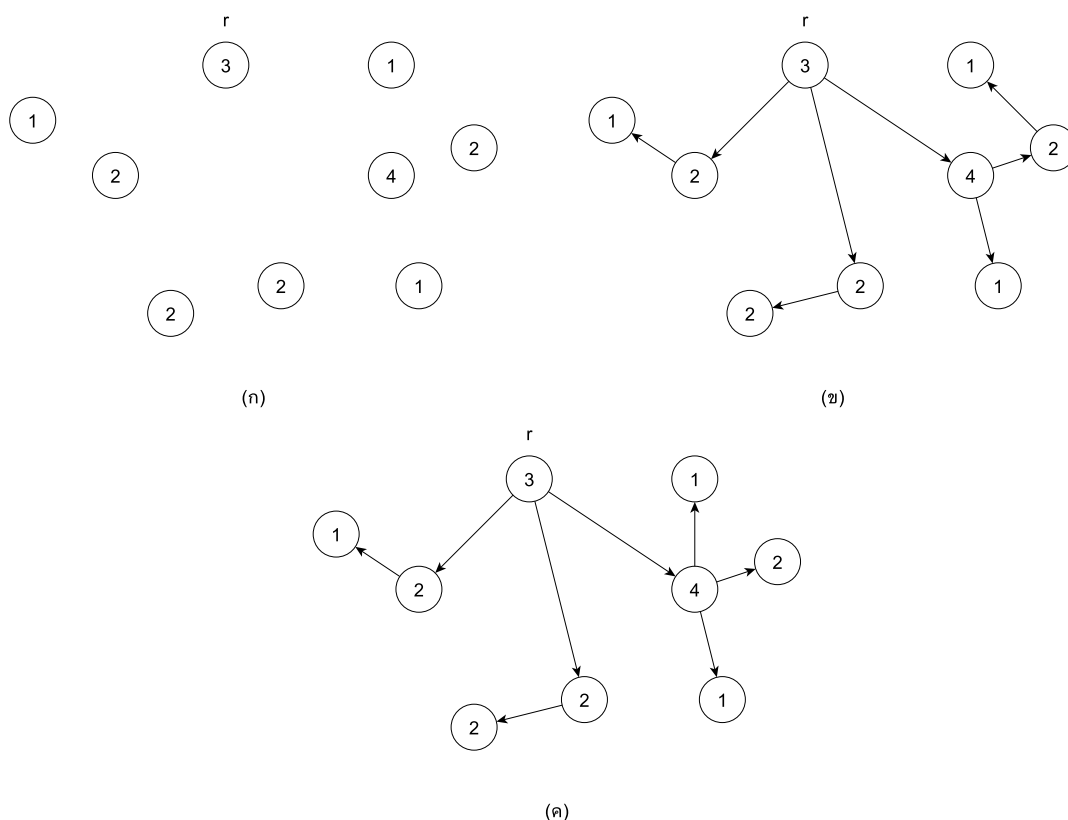
"แบนด์วิดท์ของเส้นเชื่อมทุกเส้นมีค่าเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการ"

เมื่อแบนด์วิดท์ที่ต้องการ (Required bandwidth) คือแบนด์วิดท์ขั้นต่ำที่ทำให้สามารถรับชมสื่อแบบกระแสได้อย่างมีประสิทธิภาพ ภายใต้เงื่อนไขดังกล่าวเราสามารถพิสูจน์ได้ว่ามีคำตอบที่ดีที่สุดของ MDM problem เป็นต้นไม้ ซึ่งเราจะพิสูจน์เงื่อนไขนี้เป็นจริงในบทตั้ง 1 ในบทผลและวิจารณ์

สำหรับโหนด  $v$  ใดๆ ดีกรีของโหนด  $v$  คือจำนวนครั้งทั้งหมดที่เส้นเชื่อมเกิดกับโหนด  $v$  เรานิยามขอบเขตดีกรีของโหนด  $v$  ว่าเป็นดีกรีที่มากที่สุดที่เป็นไปได้ของโหนด  $v$  โดยที่ใช้สัญลักษณ์ว่า  $B_v$  เมื่อมีคำตอบที่เป็นต้นไม้ใน MDM problem พิจารณาโหนดใดๆ จะพบว่าแบนด์วิดท์ของเส้นเชื่อมโยงขึ้นและเส้นเชื่อมโยงลง (Downlink bandwidth) สามารถพิจารณาเป็นจำนวนเส้นเชื่อมที่เกิดกับโหนดนั้นได้ ดังนั้นเมื่อพิจารณาโหนด  $u$  จะพบว่าดีกรีของโหนด  $u$  จะไม่เกิน  $U_u + 1$  เมื่อ  $U_u$  แทนแบนด์วิดท์ของเส้นเชื่อมโยงขึ้นของโหนด  $u$  ด้วยเหตุนี้ทำให้เราสนใจปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขตในกรณีที่ขอบเขตดีกรีไม่เท่ากัน โดยเส้นผ่านกลางคือความยาวของระยะทางที่สั้นที่สุดระหว่างโหนดสองโหนดใดๆ ที่ห่างกันที่สุดในต้นไม้

ปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขตในกรณีที่ขอบเขตดีกรีไม่เท่ากัน (The Bounded Degree Minimum Spanning Tree problem with non-uniform degree bound) เรียกว่า BDST problem with non-uniform degree bound ปัญหานี้กำหนด metric length function  $l$  บนเซตของโหนด  $V$  ที่มีจำนวน  $n$  โหนดและแต่ละโหนด  $v$  จะมีขอบเขตดีกรี  $B_v$  มาให้ จุดประสงค์ของปัญหานี้คือต้องการหาต้นไม้ทอดข้าม (Spanning tree) ที่มีเส้นผ่านกลางต่ำที่สุดโดยที่แต่ละโหนด  $v$  มีดีกรีในต้นไม้ไม่เกิน  $B_v$

ภาพที่ 1 แสดงตัวอย่าง instance ของปัญหา ตัวอย่างคำตอบและตัวอย่างคำตอบที่ดีที่สุด เมื่อกำหนดให้หมายเลขภายในแต่ละโหนดคือขอบเขตดีกรีของโหนดนั้น สมมติให้ค่าใช้ในการส่งข้อมูลระหว่างโหนดคู่ใดๆ เป็นหนึ่งหน่วย โดยภาพ (ก) แสดง instance ของปัญหา ภาพ (ข) แสดง



ภาพที่ 1 เมื่อหมายเลขในโหนดเป็นขอบเขตดีกรีของโหนด (ก) ตัวอย่าง instance ของปัญหา (ข) ตัวอย่างแผนการส่งข้อมูล (ค) ตัวอย่างแผนการส่งข้อมูลที่มีความล่าช้าต่ำที่สุด

ตัวอย่างแผนการส่งข้อมูลที่มีความล่าช้า 3 หน่วย และภาพ (ค) แสดงตัวอย่างแผนการส่งข้อมูลที่มีความล่าช้าต่ำที่สุดซึ่งมีความล่าช้า 2 หน่วย

เราจะลดรูป MDM problem ในกรณีที่แบนด์วิดท์ของเส้นเชื่อมทุกเส้นมีค่าเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการไปเป็น BDST problem with non-uniform degree bound เมื่อกำหนด instance ของ MDM problem มาให้ เราจะกำหนดดีกรีของแต่ละโหนดดังนี้ เรากำหนดให้ขอบเขตดีกรีของแต่ละโหนด  $v$  ให้มีค่าเป็น  $U_v + 1$  นั่นคือ  $B_v = U_v + 1$  เมื่อ  $U_v$  แทนแบนด์วิดท์ของเส้นเชื่อมโยงขึ้นของโหนด  $v$  หารด้วยแบนด์วิดท์ที่ต้องการ สำหรับโหนดรากกำหนดให้ขอบเขตดีกรีมีค่าเป็น  $U_r$  นั่นคือ  $B_r = U_r$

เรานำเอาขั้นตอนวิธีของ Könemann *et al.* (2005) ซึ่งจัดการปัญหานี้ในกรณีที่ขอบเขตดีกรีของทุกโหนดไม่เกิน  $B$  และขั้นตอนวิธีของ Frederickson *et al.* (1978) มาเป็นเครื่องมือในการจัดการ BDST problem with non-uniform degree bound คำตอบที่ได้จะเป็นต้นไม้ทอดข้ามที่มีเส้นผ่านกลาง

(Diameter) ยาวไม่เกิน  $O(\sqrt{\log n})$  เท่าของต้นไม้ที่เป็นคำตอบที่ดีที่สุด ตัวอย่างแสดงดังรูป 1

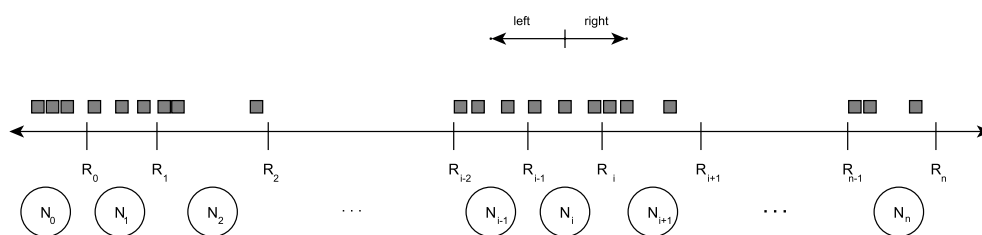
วิทยานิพนธ์ในส่วนที่สองสนใจปัญหาการค้นหาแบบช่วงบนเครือข่ายเพียร์ทูเพียร์ โดยทั่วไป การค้นหาวัตถุในเครือข่ายจะใช้กุญแจ (key) ซึ่งเป็นตัวแทนของวัตถุในระบบที่ต้องการค้นหา สำหรับการค้นหาในเครือข่ายแบบรับ-ให้บริการนั้น เมื่อเครื่องรับบริการต้องการค้นหาวัตถุ เครื่องรับบริการจะส่งคำร้องค้นหาพร้อมกุญแจให้กับเครื่องบริการ จากนั้นเครื่องบริการจะส่งคำตอบให้กับเครื่องรับบริการว่าวัตถุที่สัมพันธ์กับกุญแจนั้นเก็บไว้ตำแหน่งใดในระบบ

เนื่องจากในเครือข่ายเพียร์ทูเพียร์นั้นไม่มีเครื่องบริการ ส่วนใหญ่การค้นหาข้อมูลจึงมีการสร้างโครงสร้างข้อมูลแบบกระจายเพื่อใช้ในการค้นหา โครงสร้างข้อมูลที่เป็นที่นิยมได้แก่ตารางแฮชแบบกระจาย (Distributed hash tables (DHTs))

ตารางแฮชแบบกระจายเป็นกลุ่มของระบบแบบกระจายที่ไม่มีเครื่องศูนย์กลางและเป็นระบบที่ให้บริการการค้นหาคล้ายกับตารางแฮช โดยทั่วไปตารางแฮชแบบกระจายรองรับการดำเนินการ 2 อย่างได้แก่ การเก็บข้อมูลและการค้นหาข้อมูล การเก็บข้อมูลจะเป็นการเก็บคู่ลำดับของ (กุญแจ, ค่า) ไว้ที่โหนดที่สอดคล้องกับค่าที่ได้จากการแฮชกุญแจ ส่วนการค้นหาจะแฮชกุญแจเพื่อหาโหนดที่เก็บคู่ลำดับ ทำให้ได้ค่าจากคู่ลำดับที่โหนดนั้นเก็บไว้ การใช้ตารางแฮชแบบกระจายมีข้อดีคือ การแฮชทำให้คู่ลำดับของ (กุญแจ, ค่า) กระจายไปทั่วระบบจึงทำให้ภาระของแต่ละเครื่องที่เก็บคล้ายกัน อีกทั้งใช้เวลาในการค้นหาและเก็บข้อมูลที่ต่ำเป็น  $O(\log n)$  ข้อเสียคือเนื่องจากการใช้การแฮชนั้นทำลายคุณสมบัติท้องถิ่นของกุญแจ (Local properties) นั่นคือทำให้กุญแจที่ลำดับใกล้เคียงกัน เมื่อผ่านฟังก์ชันแฮชแล้วค่าแฮชที่ได้ไม่ใกล้เคียงกัน ทำให้การเก็บคล้ายกันไว้ต่างเครื่องกัน สิ่งนี้เองที่ทำให้เกิดความยากสำหรับโปรแกรมประยุกต์ที่ต้องการใช้การค้นหาแบบช่วง

ในวิทยานิพนธ์ส่วนที่สองนี้เราสนใจการค้นหาข้อมูลแบบช่วงที่ทำงานบนปริภูมิกุญแจโดยตรง ไม่ใช้การแฮช เราจะนำเอาเครื่องมาเรียงตามลำดับในปริภูมิของกุญแจ (Key space) จากนั้นแบ่งช่วงปริภูมิของกุญแจให้แต่ละเครื่องรับผิดชอบ เมื่อมีการใส่หรือลบกุญแจอาจจะทำให้ภาระของเครื่องนั้นมีความแตกต่างกับเครื่องอื่นมาก จึงจำเป็นต้องมีการปรับดุลภาระ (Load balancing)

Ganesan *et al.* (2004) ศึกษาปัญหาการปรับดุลภาระ (Load balancing) ที่รองรับการค้นหาแบบ



ภาพที่ 2 ตัวอย่างการแบ่งช่วงกุญแจที่แต่ละโหนดรับผิดชอบ

ช่วง Ganesan *et al.* (2004) ได้ตั้งปัญหาการปรับดุลภาระบนเครือข่ายเพียร์ทูเพียร์ไว้ดังนี้ ระบบประกอบด้วยเซต  $V$  ของโหนด  $n$  โหนดโดยที่แต่ละโหนดจะเก็บกลุ่มของกุญแจ ปริภูมิของกุญแจ (Key space) จะถูกแบ่งออกเป็นช่วง  $n$  ช่วงด้วยขอบเขต  $R_0 \leq R_1 \leq \dots \leq R_n$  กำหนดให้  $N_i$  แทนโหนดลำดับที่  $i$  ที่รับผิดชอบช่วง  $[R_{i-1}, R_i)$  สำหรับโหนด  $u$  ใดๆ กำหนดให้  $L(u)$  แทนภาระ (Load) ของโหนด  $u$  หรือจำนวนของกุญแจที่ถูกเก็บไว้ที่โหนด  $u$  นอกจากนี้เมื่อพิจารณาตามช่วงที่แต่ละโหนดรับผิดชอบ ณ เวลาใดๆ ก็ตามในระบบ โหนดในระบบจะมีลำดับซึ่งลำดับเหล่านี้จะใช้นิยามความสัมพันธ์ซ้าย (Left) และขวา (Right) ตัวอย่างแสดงดังภาพที่ 2

เมื่อกุญแจถูกใส่เข้าไปในระบบหรือถูกลบออกจากระบบ โหนดที่รับผิดชอบช่วงของกุญแจนั้นจะต้องมีการปรับปรุงข้อมูล หลังจากนั้นขั้นตอนวิธีการปรับดุลภาระจะถูกเรียก โดยมีจุดมุ่งหมายเพื่อรักษาอัตราส่วนของภาระของโหนดที่สูงสุดต่อภาระของโหนดที่ต่ำสุดให้มีค่าน้อย อัตราส่วนนี้เรียกว่า อัตราส่วนความไม่สมดุล (Imbalance ratio)

Ganesan *et al.* (2004) ได้เสนอขั้นตอนวิธีการปรับดุลภาระที่รองรับการค้นหาแบบช่วงบนเซตของโหนดที่เก็บกุญแจที่มีการเรียงของกุญแจตามลำดับ ซึ่งเรียกว่าขั้นตอนวิธี ADJUSTLOAD ในแต่ละครั้งที่มีการเรียกขั้นตอนวิธี ADJUSTLOAD ที่โหนดใดๆ จะมีการเรียกดูข้อมูลแบบครอบคลุม (Global information) ได้แก่ข้อมูลของภาระที่ต่ำที่สุดหรือสูงที่สุดในระบบ เพื่อนำมาเปรียบเทียบกับข้อมูลของภาระของโหนดนั้น จากนั้นขั้นตอนวิธีจึงจะตัดสินใจว่าจะดำเนินการอย่างไรต่อไป

พวกเขาได้พิสูจน์ว่าขั้นตอนวิธี ADJUSTLOAD รับประกันอัตราส่วนความไม่สมดุลว่าอย่างน้อยมีค่าประมาณ 4.237 อีกทั้งค่าใช้จ่ายในการดำเนินการในแต่ละการดำเนินการโดยเฉลี่ยยังเป็นค่าคงที่อีกด้วย

อย่างไรก็ตามขั้นตอนวิธีของ Ganesan *et al.* (2004) เป็นขั้นตอนวิธีแบบ recursive ซึ่งมีข้อจำกัดหลายประการเมื่อนำไปประยุกต์ใช้งานจริง ยิ่งไปกว่านั้นในการใส่หรือลบกุญแจหนึ่งครั้งต้องเรียกขั้นตอนวิธี ADJUSTLOAD ให้ทำงานกับทุกโหนดที่กระมีการเปลี่ยนแปลง ในกรณีที่แย่ที่สุดอาจจะมีการเรียกขั้นตอนวิธี ADJUSTLOAD ให้ทำงานกับทุกโหนด นอกจากนี้ในการเรียกตัวเองแต่ละครั้งจะมีการเรียกการดำเนินการพื้นฐานซึ่งต้องการข้อมูลแบบครอบคลุม นั่นหมายความว่ามีการค้นหาข้อมูลแบบครอบคลุมหลายครั้งต่อการใส่กุญแจหรือลบกุญแจหนึ่งครั้ง

ในวิทยานิพนธ์ส่วนที่สองนี้เราเสนอขั้นตอนวิธีอย่างง่ายในการปรับดุลภาระที่รองรับการค้นหาแบบช่วงและไม่เป็นขั้นตอนวิธีแบบ recursive ขั้นตอนวิธีที่เสนอนี้ใช้การดำเนินการพื้นฐานเช่นเดียวกับงานวิจัยของ Ganesan *et al.* (2004) อีกทั้งในการใส่กุญแจหรือลบกุญแจการดำเนินการเหล่านี้จะถูกเรียกใช้งานไม่เกินหนึ่งครั้ง นั่นหมายความว่าขั้นตอนวิธีที่เราเสนอนั้นจะมีการเรียกดูข้อมูลแบบครอบคลุมไม่เกินหนึ่งครั้งต่อการใส่กุญแจหรือลบกุญแจ

เราศึกษาการปรับดุลภาระในกรณีที่มิแต่การใส่กุญแจเพียงอย่างเดียวและกรณีที่มิทั้งการใส่กุญแจและลบกุญแจ ในกรณีที่มิแต่การใส่กุญแจอย่างเดียวเราแสดงว่าขั้นตอนวิธีที่เสนอนั้นมีอัตราส่วนความไม่สมดุลจะมีค่าไม่เกิน 7.464 โดยใช้ค่าใช้จ่ายถ่วงเฉลี่ยในแต่ละการดำเนินการเป็นค่าคงที่ ส่วนในกรณีที่มิทั้งการใส่กุญแจและลบกุญแจ เราแสดงว่าขั้นตอนวิธีที่เสนอนั้นมีอัตราส่วนความไม่สมดุลจะมีค่าไม่เกิน 7.464 เช่นกันโดยใช้ค่าใช้จ่ายถ่วงเฉลี่ยในแต่ละการดำเนินการเป็นค่าคงที่

## วัตถุประสงค์

1. เพื่อแก้ปัญหาการวางแผนการส่งข้อมูลแบบกระแสนเครือข่ายเพียร์ทูเพียร์ โดยมีเงื่อนไขว่าแต่ละเครื่องในเครือข่ายมีความสามารถในการส่งต่อข้อมูลได้แตกต่างกัน
2. เพื่อแก้ปัญหาการออกแบบวิธีการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วง

## ขอบเขตของงานวิจัย

1. ศึกษาปัญหาการวางแผนการส่งข้อมูลแบบกระแสในเครือข่ายเพียร์ทูเพียร์ โดยมีเงื่อนไขว่าแต่ละเครื่องในเครือข่ายมีความสามารถในการส่งต่อข้อมูลได้แตกต่างกัน โดยทำการวิจัยเฉพาะด้านทฤษฎี
2. สำหรับปัญหาการวางแผนการส่งข้อมูลแบบกระแสในเครือข่ายเพียร์ทูเพียร์ โดยมีเงื่อนไขว่าแต่ละเครื่องในเครือข่ายมีความสามารถในการส่งต่อข้อมูลได้แตกต่างกัน เราสนใจกรณีที่เป็นแบนด์วิดท์ของเส้นเชื่อมทุกเส้นเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการ
3. ศึกษาปัญหาการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วง โดยทำการวิจัยเฉพาะด้านทฤษฎี
4. สำหรับปัญหาการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วง เราสนใจการเพิ่มและลบคีย์ที่เป็นแบบลำดับ ไม่พิจารณากรณีที่เพิ่มหรือลบคีย์พร้อมกัน
5. สำหรับปัญหาการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วง เราจะสมมติว่าจำนวนโหนดในระบบคงที่ตลอดการทำงาน

## การตรวจเอกสาร

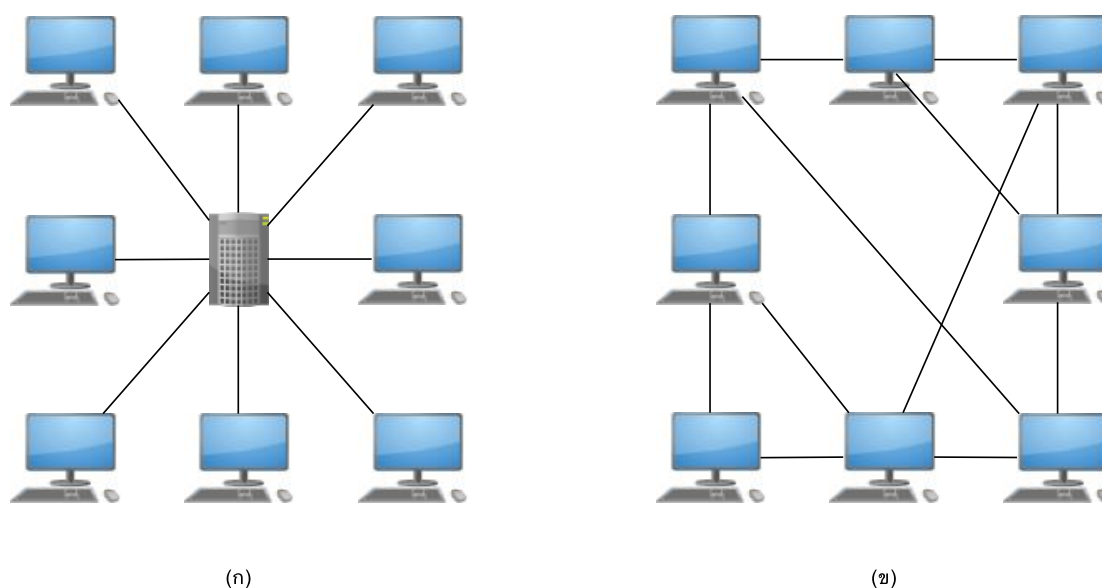
ในส่วนนี้เราจะตรวจเอกสารที่เกี่ยวข้องได้แก่ เครือข่ายเพียร์ทูเพียร์ ปัญหาการส่งข้อมูลแบบกระแส ปัญหาที่เกี่ยวข้องกับการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดิกรีมีขอบเขตและปัญหาของพนักงานขาย  $k$  คน หัวข้อดังกล่าวเกี่ยวข้องกับงานวิจัยการวางแผนการส่งข้อมูลแบบกระแสบนเครือข่ายเพียร์ทูเพียร์ จากนั้นเราจะตรวจเอกสารที่เกี่ยวข้องซึ่งเกี่ยวกับงานวิจัยการค้นหาแบบช่วงบนเครือข่ายเพียร์ทูเพียร์ได้แก่ ปัญหาการค้นหาแบบช่วง การปรับดูแลการในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วงและขั้นตอนวิธีของ Ganesan *et al.* (2004)

## เครือข่ายเพียร์ทูเพียร์

เครือข่ายเพียร์ทูเพียร์ (Peer-to-Peer network) เป็นเครือข่ายการติดต่อสื่อสารรูปแบบหนึ่งที่เครื่องคอมพิวเตอร์แต่ละเครื่องมีสิทธิเท่าเทียมกัน นั่นคือเครื่องคอมพิวเตอร์แต่ละเครื่องนั้นสามารถรับข้อมูลและส่งต่อข้อมูลได้ เครือข่ายเพียร์ทูเพียร์เป็นเครือข่ายที่แตกต่างกับเครือข่ายรับ-ให้บริการ (Client-Server network) ที่เมื่อเครื่องรับบริการต้องการข้อมูลจะต้องติดต่อกับเครื่องบริการเพียงอย่างเดียว สำหรับ โปรแกรมประยุกต์ที่ทำงานโดยใช้เครือข่ายเพียร์ทูเพียร์ที่ได้รับความนิยมตัวอย่างเช่น โปรแกรมประยุกต์สำหรับการแชร์ไฟล์ (File sharing) เช่น BitTorrent, eDonkey เป็นต้น โปรแกรมประยุกต์สำหรับโทรศัพท์ผ่านทางอินเทอร์เน็ต (Internet phone) เช่น Skype เป็นต้น และ โปรแกรมประยุกต์สำหรับชมสื่อแบบกระแส (Live streaming) เช่น Sopcast, CoolStreaming เป็นต้น

เนื่องจากเครือข่ายเพียร์ทูเพียร์นั้นไม่มีเครื่องบริการ การค้นหาข้อมูลส่วนใหญ่จึงมีการสร้างโครงสร้างข้อมูลเพื่อช่วยในการค้นหา สำหรับ โครงสร้างข้อมูลที่เป็นที่นิยมได้แก่ตารางแฮชแบบกระจาย (Distributed hash tables (DHTs))

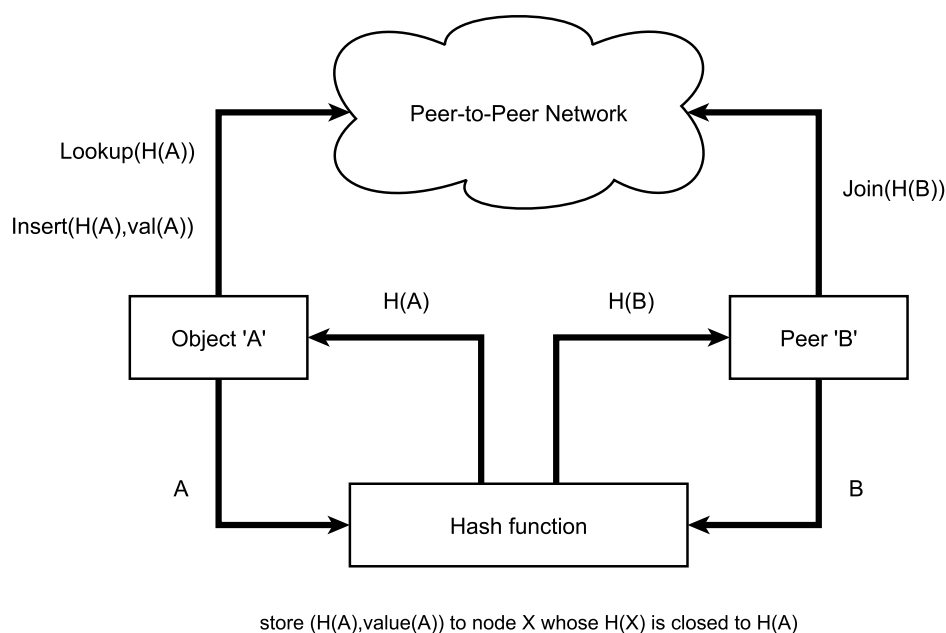
ตารางแฮชแบบกระจายเป็นกลุ่มของระบบแบบกระจายที่ไม่มีเครื่องศูนย์กลางและเป็นระบบที่ให้บริการการค้นหาคล้ายกับตารางแฮช โดยทั่วไปตารางแฮชแบบกระจายรองรับการดำเนินการพื้นฐาน 2 อย่างได้แก่ การเก็บข้อมูลและการค้นหาข้อมูล ภาพที่ 4 แสดงตัวอย่างการทำงานของตารางแฮชแบบกระจาย โดยอธิบายได้ดังนี้ เครื่องแต่ละเครื่องจะมีกุญแจที่สอดคล้องกันอยู่ซึ่งใช้เป็นตัวแทนของเครื่องนั้น ตัวอย่างเช่นเมื่อเครื่อง B จะเข้าใช้งานระบบ เครื่อง B จะนำเอากุญแจที่สอดคล้องนั้นมาผ่านแฮชฟังก์ชัน สมมติว่าได้ค่าเป็น  $H(B)$  จากนั้นจะเข้าใช้งานระบบด้วยคำสั่ง  $Join(H(B))$  สำหรับวัตถุใดๆ



ภาพที่ 3 ตัวอย่างเครือข่าย (ก) เครือข่ายรับ-ให้บริการและ (ข) เครือข่ายเพียร์ทูเพียร์

ก็เช่นกันจะมีกุญแจที่สอดคล้องกันอยู่ ตัวอย่างเช่นเมื่อต้องการเก็บค่าของวัตถุ A เริ่มต้นวัตถุ A จะนำเอากุญแจที่สอดคล้องกันไปผ่านแฮชฟังก์ชัน สมมติว่าได้ค่าเป็น  $H(A)$  จากนั้นการใส่วัตถุ A เข้าไปในระบบจะใช้คำสั่ง  $\text{Insert}(H(A), \text{val}(A))$  โดยที่คู่ลำดับ  $(H(A), \text{val}(A))$  จะถูกเก็บไว้ที่โหนด X ที่มีค่า  $H(X)$  ที่มีค่าใกล้เคียงกับค่า  $H(A)$  ในการค้นหาวัตถุ A จะใช้คำสั่ง  $\text{Lookup}(H(A))$  เพื่อค้นหาโหนด X ที่มีค่า  $H(X)$  ใกล้เคียงกับค่า  $H(A)$  สุดท้ายจะได้ค่า  $\text{val}(A)$

การใช้ตารางแฮชแบบกระจายมีข้อดีคือ การแฮชทำให้คู่ลำดับ  $(H(A), \text{val}(A))$  ใดๆ มีการกระจายไปทั่วระบบทำให้ปริมาณของกุญแจที่แต่ละโหนดนั้นเก็บไว้หรือเรียกอีกอย่างว่าภาระของแต่ละโหนดมีปริมาณที่ใกล้เคียงกัน นอกจากนี้การใช้ตารางแฮชแบบกระจายยังทำให้ระบบรองรับการปรับขนาดได้และมีความทนต่อการเสียของโหนด อีกทั้งใช้เวลาในการค้นหาข้อมูลต่ำเป็น  $O(\log n)$  ตัวอย่างงานวิจัยที่เกี่ยวกับตารางแฮชแบบกระจายเช่น Kademlia จากงานของ Maymounkov and Mazières (2002), CAN จากงานของ Ratnasamy *et al.* (2001), Pastry จากงานของ Rowstron and Druschel (2001), Chord จากงานของ Stoica *et al.* (2003) และ Tapestry จากงานของ Zhao *et al.* (2004)



ภาพที่ 4 ตัวอย่างการทำงานของตารางแฮชแบบกระจาย

### ปัญหาการออกแบบแผนการส่งข้อมูลแบบกระจาย

การส่งข้อมูลแบบกระจาย (Media streaming) ในเครือข่ายเพียร์ทูเพียร์เป็นหัวข้องานวิจัยที่ได้รับความสนใจเป็นอย่างมาก จุดประสงค์หลักของงานวิจัยทางด้านนี้คือการออกแบบวิธีการส่งสื่อแบบกระจายที่มีคุณภาพสูงไปให้กับผู้ใช้งานจำนวนมาก สำหรับวิทยานิพนธ์นี้ หนึ่งในเงื่อนไขสำหรับการการส่งข้อมูลแบบกระจายในเครือข่ายเพียร์ทูเพียร์ที่เราสนใจคือความล่าช้า (Delay) ในการส่งข้อมูลจากเครื่องต้นทางไปยังทุกเครื่องที่ต่ำที่สุด ทั้งนี้ความล่าช้าเป็นหนึ่งในเกณฑ์ที่นำมาใช้วัดคุณภาพของโปรแกรมประยุกต์แบบเวลาจริง (Real-Time applications) เพราะว่าความล่าช้าทำให้การโต้ตอบมีปัญหา

ปัญหาการส่งข้อมูลในเครือข่ายเพียร์ทูเพียร์ที่มีความล่าช้าที่ต่ำที่สุดนั้นมีการศึกษาค้นคว้ามากมาย หลายงานวิจัยเสนอการใช้โครงสร้างในการส่งข้อมูลแบบต่างๆ เพื่อที่จะทำให้การส่งข้อมูลมีความล่าช้าที่ต่ำ ตัวอย่างการใช้โครงสร้างต้นไม้ต้นเดียวในการส่งข้อมูลพบในงานวิจัย

Deshpande *et al.* (2002) และ Tran *et al.* (2006) เป็นต้น การใช้โครงสร้างต้นไม้หลายต้นในการส่งข้อมูลซึ่งเป็นการสร้างโครงสร้างต้นไม้แบบต้นเดียวหลายๆ แบบ โดยมีเงื่อนไขว่าเครื่องรับบริการแต่ละเครื่องจะต้องเป็นโหนดภายในของต้นไม้อย่างน้อยหนึ่งต้นจากหลายๆ ต้น พบในงานวิจัย

Padmanabhan *et al.* (2002), Castro *et al.* (2003), Venkataraman *et al.* (2006) และ Liao *et al.* (2007)

เป็นต้น นอกจากนี้ยังมีการใช้โครงสร้างอื่นๆ ในการส่งข้อมูลเช่น Jiang *et al.* (2003) เป็นการส่งข้อมูลบน โครงสร้างของเครือข่ายแบบ Gnutella , Zhang *et al.* (2005) เป็นการรับส่งข้อมูล โดยที่จะเลือกรับข้อมูลจากโหนดใกล้เคียงที่ทำการเลือกไว้ก่อน โดยใช้โพรโทคอลแบบ Gossip-Based เป็นต้น

นอกจากนี้ยังมีงานวิจัยหลายฉบับที่มีข้อสมมติฐานเกี่ยวกับสภาพแวดล้อมว่าสภาพแวดล้อมที่ศึกษามีลักษณะเหมือนกัน (Homogeneous environments) เช่น Nguyen and Cheung. (2005), Zhou *et al.* (2007) เป็นต้น แต่ข้อสมมติฐานเช่นนี้ไม่เป็นจริงในเครือข่ายจริง เนื่องจากในเครือข่ายจริงแต่ละเครื่องส่วนใหญ่มีลักษณะแตกต่างกัน โดยเฉพาะอย่างยิ่งความสามารถในการส่งต่อข้อมูลไปยังเครื่องอื่นมักแตกต่างกัน ดังนั้นในวิทยานิพนธ์ฉบับนี้เราจึงมีเงื่อนไขอีกอย่างที่น่าสนใจคือสิ่งแวดล้อมที่ศึกษามีลักษณะไม่เหมือนกัน (Heterogeneous environments) เมื่อกล่าวอย่างเจาะจงคือวิทยานิพนธ์ฉบับนี้สนใจเครือข่ายที่แต่ละโหนดมีแบนด์วิดท์ของเส้นเชื่อมโยงขึ้น (Uplink bandwidth) ที่แตกต่างกัน

ในวิทยานิพนธ์ส่วนแรกเราสนใจกรณีเฉพาะกรณีหนึ่งของปัญหาการส่งข้อมูลที่มี โครงสร้างแบบเมชที่มีความล่าช้าต่ำที่สุด (Minimum Delay Mesh problem) ซึ่งเรียกย่อว่า MDM problem ทั้งนี้ MDM problem เป็นปัญหาการออกแบบแผนการส่งข้อมูลแบบกระแสที่ได้รับความสนใจมากปัญหาหนึ่ง เริ่มต้น Ren *et al.* (2008) ได้นิยาม MDM problem ดังนี้

ใน MDM problem มีการนิยามส่วนประกอบ 3 ส่วน ส่วนแรกเป็นการนิยามเครือข่าย พิจารณาเครือข่ายเป็นโครงข่ายซ้อนทับ (Overlay network) ซึ่งโมเดลเป็นกราฟบริบูรณ์แบบมีทิศทาง (Complete directed graph)  $G = (V, E)$  โดยที่  $V$  เป็นเซตของเพียร์และ  $E$  เป็นเซตของเส้นเชื่อมของโครงข่ายซ้อนทับ เส้นเชื่อม  $(u, v)$  แต่ละเส้นจะมีค่าใช้จ่าย  $\ell(u, v)$  ที่สอดคล้องกับความล่าช้าในการส่งข้อมูลจากโหนด  $u$  ไปยังโหนด  $v$  ในเครือข่าย สำหรับความล่าช้าของเส้นทาง  $P$  คือผลรวมของความล่าช้าของแต่ละเส้นเชื่อมในเส้นทาง  $P$  ซึ่งมีค่าเป็น  $\sum_{(u,v) \in P} \ell(u, v)$

ส่วนที่สองเป็นการนิยามความต้องการแบนด์วิดท์ เนื่องจากเครือข่ายเพียร์ทูเพียร์ แต่ละโหนดสามารถส่งข้อมูลต่อให้กับโหนดอื่นๆ ได้ สำหรับแต่ละโหนด  $u$  จะกำหนดให้  $U_u$  แทนแบนด์วิดท์ของเส้นเชื่อมโยงขึ้น (Uplink bandwidth) ของโหนด  $u$  ให้หน่วย (Unit) แทนขนาดของแพ็คเกจ (Package) ที่เล็กที่สุดใน โปรแกรมประยุกต์ที่ส่งข้อมูลแบบกระแส ให้  $s$  เป็นอัตราในการส่งข้อมูลแบบกระแส สมมติว่ามีเครื่องต้นทาง  $r \in V$  ที่ต้องการส่งข้อมูลเพียงเครื่องเดียว เงื่อนไขความต้องการแบนด์วิดท์

ได้แก่  $r$  จะส่งข้อมูลด้วยอัตรา  $s$  หน่วยต่อวินาที และแต่ละเพียร์ต้องการรับข้อมูลอย่างน้อย  $s$  หน่วยต่อวินาที

ส่วนที่สามเป็นการนิยามแผนการส่งข้อมูลแบบกระแส (Streaming scheme) แผนการส่งข้อมูลแบบกระแสเป็นการอธิบายว่าข้อมูลแบบกระแสจะถูกส่งให้แต่ละเพียร์จากเครื่องต้นทางได้อย่างไร แผนการส่งนี้จะระบุเซตของโหนดพ่อ  $P_u$  ของแต่ละเพียร์  $u$  นั่นคือจะระบุว่าโหนดใดบ้างที่ส่งส่วนของข้อมูลแบบกระแสให้กับโหนด  $u$  นอกจากนี้ยังระบุปริมาณที่ส่งอีกด้วย โดยกำหนดให้  $s_u(v)$  แทนปริมาณของแบนด์วิธของข้อมูลแบบกระแสที่โหนด  $u$  รับมาจากโหนด  $v$  อย่างไรก็ตามแผนการส่งข้อมูลแบบกระแสนั้นแต่ละโหนดที่ส่งข้อมูลจะส่งได้ไม่เกินแบนด์วิธของเส้นเชื่อมโยงขึ้น นั่นคือเมื่อพิจารณาโหนดพ่อ  $v$  แต่ละโหนดนั้นมีเงื่อนไขว่า  $\sum_{u: P_u \ni v} s_u(v) \leq U_v$  อีกทั้งความสัมพันธ์พ่วงถูกที่นิยามโดย  $P_u$  สำหรับทุกโหนด  $u \in V$  จะต้องไม่เป็นวัฏจักร (Cycle)

จุดประสงค์ของ MDM problem เป็นการหาแผนการส่งข้อมูลแบบกระแสที่คำนึงถึงเงื่อนไขแบนด์วิธของเส้นเชื่อมโยงขึ้นของทุกโหนดขณะเดียวกันต้องการให้ความล่าช้าที่มากที่สุดจากเครื่องต้นทางไปยังโหนดใดๆ มีค่าต่ำที่สุด

Ren *et al.* (2008) ได้พิสูจน์ว่าปัญหานี้เป็นปัญหา NP-Complete และได้เสนอขั้นตอนวิธีแบบศึกษาสำนึก (Heuristic algorithm) พวกเขานิยาม  $P_i(j)$  ว่าเป็นอัตราส่วนที่โหนด  $j$  ให้แบนด์วิธบริการกับโหนด  $i$  หารด้วยความล่าช้าจากโหนด  $j$  ไปโหนด  $i$  ขั้นตอนวิธีของพวกเขาทำงานดังนี้ เริ่มต้นแต่ละโหนดจะถูกเรียงตามค่าแบนด์วิธของเส้นเชื่อมโยงขึ้นจากมากไปน้อย จากนั้นโหนดจะถูกเพิ่มเข้าไปในระบบตามลำดับนี้ เมื่อโหนด  $i$  ถูกเพิ่มเข้าไปในระบบ ขั้นตอนวิธีจะเลือกโหนด  $j$  ที่มี  $P_i(j)$  ที่มากที่สุด จากนั้นเชื่อมโหนด  $i$  ต่อกับโหนด  $j$  ถ้าโหนด  $i$  ยังรับข้อมูลไม่ถึง  $s$  หน่วย ขั้นตอนวิธีจะหาโหนดใหม่  $k$  ที่มี  $P_i(k)$  มากที่สุดเพื่อเชื่อมต่อกับโหนด  $i$  ขั้นตอนวิธีจะเชื่อมต่อโหนดใหม่กับโหนด  $i$  จนกระทั่งได้รับข้อมูลครบ  $s$  หน่วย จากนั้นขั้นตอนวิธีจะเพิ่มโหนดถัดไปเข้าสู่ระบบ ทำเช่นนี้จนครบทุกโหนด

นอกจากนี้พวกเขายังได้ทำการทดลองเพื่อเปรียบเทียบประสิทธิภาพการทำงานเทียบกับ วิธีการเพิ่มโหนดใหม่โดยเชื่อมเข้ากับโหนดในระบบที่ใกล้ที่สุดและเปรียบเทียบกับงานวิจัย Outreach ของ Small *et al.* (2007) จากการทดลองพบว่าผลลัพธ์ที่ได้มีความล่าช้าเฉลี่ยและความล่าช้าที่มากที่สุดไป

ยังโหนดใดๆ ต่ำกว่า 2 วิธีที่นำมาเปรียบเทียบ อีกทั้งพบว่าจำนวน hop เฉลี่ยและจำนวน hop สูงสุดมีจำนวนต่ำกว่าเช่นกัน

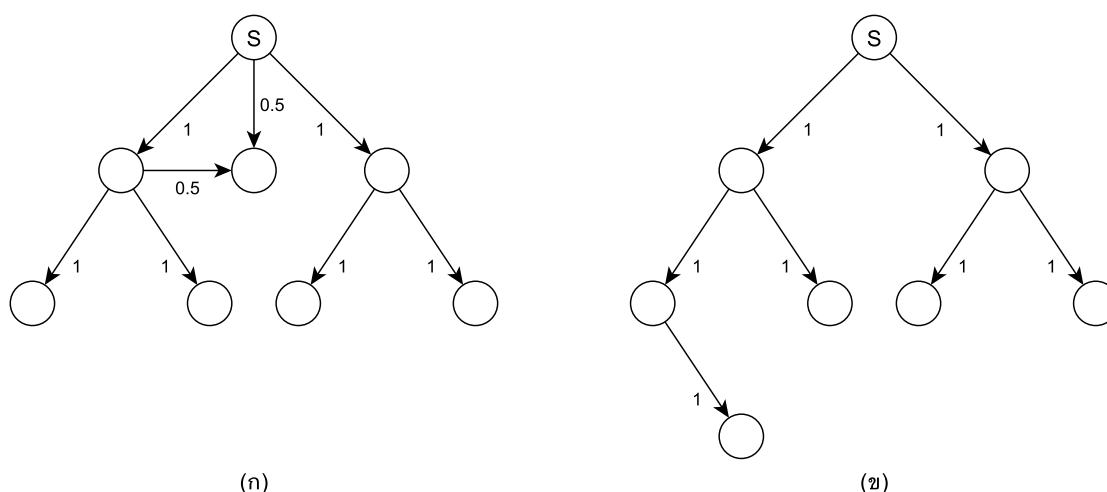
Huang *et al.* (2009) ได้ศึกษากรณีพิเศษกรณีหนึ่งของ MDM problem ในอีกชื่อหนึ่งคือปัญหาการส่งข้อมูลแบบกระแสในเครือข่ายเพียร์ทูเพียร์ที่มีความล่าช้าต่ำที่สุด (The Minimum Delay Peer-to-Peer streaming problem) ทั้งนี้พวกเขาได้อ้างว่าคำตอบที่ดีที่สุดในการกรณีพิเศษนี้จะเป็นต้นไม้ พวกเขาใช้เทคนิคในการแก้ปัญหาล้ำกับวิธีที่วิทยานิพนธ์นี้นำเสนอ นั่นคือการแปลงปัญหาให้อยู่ในรูปของปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางต่ำที่สุดและดิกกรีมีขอบเขตและใช้ขั้นตอนวิธีของ Könemann *et al.* (2005) ที่ปรับปรุงแล้วมาแก้ปัญหา ขั้นตอนวิธีที่พวกเขาเสนอนั้นให้ผลลัพธ์ที่มีความล่าช้าไม่เกิน  $O(\sqrt{\log n})$  เท่าของคำตอบที่ดีที่สุด

เมื่อกำหนดให้  $s$  คือปริมาณแบนด์วิดท์ที่ต้องการ Huang *et al.* (2009) ได้ตั้งเงื่อนไขสำหรับปัญหาการส่งข้อมูลแบบกระแสในเครือข่ายเพียร์ทูเพียร์ที่มีความล่าช้าต่ำที่สุดว่า

1. โหนดจำนวนอย่างน้อยครึ่งหนึ่งมีแบนด์วิดท์ของเส้นเชื่อมโยงขึ้นอย่างน้อย  $2s$  หน่วยต่อวินาที
2. ทุกโหนดจะต้องมีแบนด์วิดท์ของเส้นเชื่อมโยงขึ้นมากกว่าหรือเท่ากับ 1 หน่วยต่อวินาที

Huang *et al.* (2009) ได้อ้างว่าภายใต้เงื่อนไขดังกล่าวนี้ จะมีแผนสำหรับการส่งข้อมูลที่ดีที่สุดเป็นต้นไม้ในกรณีที่  $s = 1$  หรือ เป็นต้นไม้หลายต้นมารวมกันในกรณีที่  $s > 1$  โดย Huang *et al.* (2009) อ้างไว้ในงานของพวกเขาในหัวข้อที่ 4.2.4

เราพบว่าเงื่อนไขที่ Huang *et al.* (2009) ได้อ้างไว้ไม่เพียงพอต่อการรับประกันว่าแผนสำหรับการส่งข้อมูลที่ดีที่สุดเป็นต้นไม้ เราจะแสดงโดยการยกตัวอย่างที่ขัดแย้งกับเงื่อนไขเหล่านี้ พิจารณาเครือข่ายที่มี 8 โหนด โดยที่แต่ละโหนดมีแบนด์วิดท์ของเส้นเชื่อมโยงขึ้น 2.5 หน่วยต่อวินาที และกำหนดให้แบนด์วิดท์ที่ต้องการมีค่าเป็น 1 หน่วยต่อวินาที ( $s = 1$ ) และให้เส้นเชื่อมแต่ละเส้นมีความล่าช้า 1 หน่วย ด้วยการกำหนดค่าต่างๆ ดังที่กล่าวมาเราพบว่า ต้นไม้ใดๆ จะมีความล่าช้าที่ต่ำที่สุดมีค่าเป็น 3 หน่วย แต่เครือข่ายแบบเมชจะมีความล่าช้าที่ต่ำที่สุดมีค่าเป็น 2 หน่วย ตัวอย่างดังกล่าวขัดแย้งกับที่ Huang *et al.* (2009) ได้อ้างไว้ว่าจะมีแผนสำหรับการส่งข้อมูลที่ดีที่สุดเป็นต้นไม้ เราได้แสดงตัวอย่างดังภาพที่ 5



ภาพที่ 5 ตัวอย่างที่ขัดแย้งกับเงื่อนไขของ Huang *et al.* (2009) (ก) เครือข่ายแบบเมฆซึ่งมีความล่าช้า 2 หน่วย และ (ข) เครือข่ายแบบต้นไม้ ซึ่งมีความล่าช้า 3 หน่วย

### ปัญหาที่เกี่ยวข้องกับการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขต

Shi *et al.* (2001) ได้นิยามปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขตในกรณีที่มีขอบเขตดีกรีไม่เท่ากัน (The Bounded Degree Minimum Spanning Tree problem with non-uniform degree bound) ปัญหานี้เรียกย่อว่า BDST problem with non-uniform degree bound ปัญหานี้กำหนด metric length function  $l$  บนเซตของโหนด  $V$  ที่มีจำนวน  $n$  โหนดและแต่ละโหนด  $v$  จะมีขอบเขตดีกรี  $B_v$  มาให้ จุดประสงค์ของปัญหานี้คือต้องการหาต้นไม้ทอดข้าม (Spanning tree) ที่มีเส้นผ่านกลาง (Diameter) ต่ำที่สุดโดยที่แต่ละโหนด  $v$  มีดีกรีในต้นไม้ไม่เกิน  $B_v$  เมื่อเส้นผ่านกลางคือความยาวของระยะทางที่สั้นที่สุดระหว่างโหนดสองโหนดใดๆ ที่ห่างกันที่สุดในต้นไม้

Shi *et al.* (2001) ได้เสนอขั้นตอนวิธีแบบศึกษาสำนึกสำหรับ BDST problem with non-uniform degree bound ในปัญหานี้ขอบเขตดีกรีของแต่ละโหนดมีค่าแตกต่างกันได้ พวกเขาได้พิสูจน์ว่าปัญหานี้เป็นปัญหา NP-Complete และได้เสนอขั้นตอนวิธีเชิงละโมบ (Greedy algorithm) ที่ทำงานคล้ายกับ Prim's MST algorithm เริ่มต้นกำหนดให้โหนดราก  $r$  หนึ่งโหนดอยู่ในต้นไม้  $T$  จากนั้นเพิ่มโหนดใหม่  $u$  เข้าเชื่อมต่อกับโหนดที่อยู่ในต้นไม้  $T$  ขั้นตอนวิธีจะเลือกโหนดที่มีดีกรีที่ยังไม่ถูกใช้และเมื่อเชื่อมกับโหนด  $u$  แล้วทำให้ระยะทางจากโหนด  $u$  ไปยังโหนดใดๆ ใน  $T$  สั้นที่สุด Shi *et al.* (2001) ได้พิสูจน์ว่าผลลัพธ์ที่ได้จากขั้นตอนวิธีเป็นต้นไม้ทอดข้ามที่มีเส้นผ่านกลางไม่เกิน  $O(2\epsilon \log_{d_{\min}} d^{\max})$

เท่าของต้นไม้ที่ดีที่สุด เมื่อ  $\epsilon$  คือน้ำหนักของเส้นเชื่อมที่มากที่สุด  $d^{min}$  และ  $d^{max}$  แทนคิรีของโหนดที่น้อยที่สุดและมากที่สุดตามลำดับ

Könemann *et al.* (2005) ได้เสนอขั้นตอนวิธีแบบประมาณสำหรับปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางต่ำที่สุดและคิรีมีขอบเขต ในกรณีที่แต่ละโหนดมีขอบเขตคิรีจำกัดด้วยค่า  $B$  ค่าเดียวกันทุกโหนด ต่อไปเราจะเรียกว่าขั้นตอนวิธี KLS เริ่มต้นขั้นตอนวิธี KLS จะทำการแบ่งกราฟออกเป็นกลุ่มย่อยๆ ซึ่งภายในแต่ละกลุ่มมีเส้นผ่านกลางต่ำ แล้วสร้างต้นไม้ที่เป็นแกนหลักเชื่อมที่ระหว่างกลุ่มย่อย จากนั้นในแต่ละกลุ่มย่อยจะสร้าง Balanced  $(B - 1)$ -ary tree ขั้นตอนวิธี KLS ให้ผลลัพธ์เป็นต้นไม้ทอดข้ามที่มีเส้นผ่านกลางไม่เกิน  $O(\sqrt{\log n})$  เท่าของต้นไม้ที่ดีที่สุด เมื่อ  $B$  มีค่ามากกว่าหรือเท่ากับ 3

นอกจากนี้ Ravi (1994) ได้ศึกษาปัญหาที่ใกล้เคียงกันที่มีชื่อว่าปัญหาเวลาในการแพร่สัญญาณที่ต่ำที่สุด (The Minimum Broadcast Time problem) ปัญหานี้กำหนดกราฟที่เชื่อมกันแบบไม่มีทิศทางและโหนดราก (Root node) มาให้ จุดมุ่งหมายของปัญหานี้คือต้องการออกแบบแผนการส่งข้อมูล (Scheme) สำหรับการแพร่สัญญาณที่ใช้เวลาในการแพร่สัญญาณไปยังทุกโหนดต่ำที่สุด การทำงานจะทำงานเป็นรอบ ในแต่ละรอบการทำงานแต่ละโหนดที่ได้รับข้อมูลแล้วจะถูกอนุญาตให้ส่งข้อมูลให้กับโหนดอื่นที่ติดกันได้รอบละหนึ่งโหนดเท่านั้น จุดที่แตกต่างกันของปัญหานี้กับปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางต่ำที่สุดและคิรีมีขอบเขตคือ ปัญหานี้พิจารณากรณีที่เส้นเชื่อมที่เป็นไปได้ไม่จำเป็นต้องเป็นเมตริก (Non-metric case) แต่ว่าปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางต่ำที่สุดและคิรีมีขอบเขตพิจารณากรณีที่เส้นเชื่อมที่เป็นไปได้เป็นเมตริก (Metric case) Ravi ได้เสนอขั้นตอนวิธีแบบประมาณสำหรับปัญหาเวลาในการแพร่สัญญาณที่ต่ำที่สุดที่ได้อัตราส่วนต่อวิธีที่ดีที่สุดเป็น  $O(\frac{\log^2 n}{\log \log n})$  ต่อมา Elkin and Kortsarz (2006) ได้พัฒนาขั้นตอนวิธีที่ได้อัตราส่วนที่ดีขึ้นเป็น  $O(\frac{\log n}{\log \log n})$

### ปัญหาการเดินทางของพนักงานขาย $k$ คน

ปัญหานี้กำหนดกราฟบริบูรณ์แบบไม่มีทิศทาง (Undirected complete graph)  $G = (V, E)$  โหนดเริ่มต้น  $r \in V$  และฟังก์ชันค่าใช้จ่ายที่ไม่เป็นค่าลบบนแต่ละเส้นเชื่อม  $e$  ที่สอดคล้องกับอสมการอิงรูปสามเหลี่ยม (Triangle inequality) มาให้ ปัญหาการเดินทางของพนักงานขาย  $k$  คน The  $k$ -Traveling Salesperson Problem คือการหาหมู่ (Collection) ของทัวร์ย่อย (Subtour)  $k$  ทัวร์ที่แต่ละทัวร์ย่อยประกอบด้วยโหนดเริ่มต้น  $r$  และแต่ละโหนดจะอยู่ในทัวร์ย่อยอย่างน้อยหนึ่งทัวร์ ความยาวของ  $k$ -ทัวร์ถูกนิยาม

ด้วยความยาวที่มากที่สุดท่ามกลาง  $k$  ทัวรย้อยนี้ สำหรับ  $k$ -ทัวรที่ดีที่สุดคือ  $k$ -ทัวรที่มีความยาวน้อยที่สุด

Frederickson *et al.* (1978) ได้เสนอขั้นตอนวิธีสำหรับแก้ปัญหานี้ ซึ่งต่อไปเราจะเรียกว่าขั้นตอนวิธี FHK เริ่มต้นขั้นตอนวิธี FHK จะทำการหาทัวร  $R = (r = v_1, v_2, \dots, v_n, v_1)$  ที่ผ่านทุกโหนด โดยมีโหนด  $v_1$  เป็นโหนดเริ่มต้น ให้  $c$  แทนค่าใช้จ่ายที่ไม่เป็นค่าลบบนแต่ละเส้นเชื่อม  $e$  ที่สอดคล้องกับอสมการเชิงรูปสามเหลี่ยม สมมติว่าทัวร  $R$  มีค่าใช้จ่ายเป็น  $c(R) = L$  จากนั้นแต่ละ  $j$  ที่  $1 \leq j < k$  ให้ท่องไปตามทัวร  $R$  จนเจอโหนดสุดท้าย  $v_{l(j)}$  ที่มีค่าใช้จ่ายของเส้นทางเริ่มจากโหนด  $r$  ไปยังโหนด  $v_{l(j)}$  ตามทัวร  $R$  ไม่เกิน  $j/k(L - 2C_{max}) + C_{max}$  เมื่อ  $C_{max}$  แทนด้วย  $\max_{1 \leq i \leq n} c(v_1, v_i)$  จากนั้นเชื่อมแต่ละ  $v_{l(j)}$  และ  $v_{l(j)+1}$  กับ  $v_1$  จะได้  $k$ -ทัวรที่มาจาก  $k$  ทัวรย้อยดังนี้  $R_1 = v_1, \dots, v_{l(1)}, v_1$ ,  $R_2 = v_1, v_{l(1)+1}, \dots, v_{l(2)}, v_1, \dots, R_k = v_1, v_{l(k-1)+1}, \dots, v_n, v_1$  ตัวอย่างดังภาพที่ 6

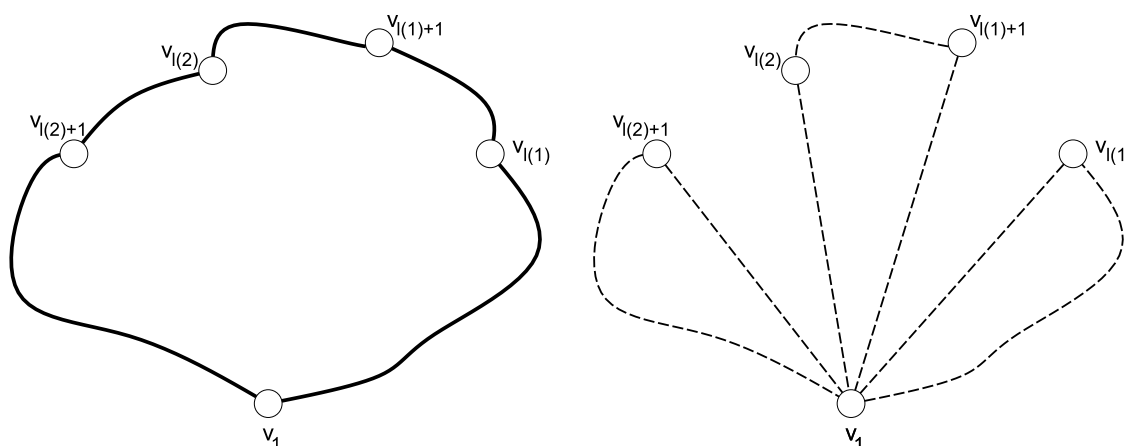
เมื่อกำหนดให้  $C_k$  แทนค่าใช้จ่ายของทัวรย้อยที่มากที่สุดที่ได้จากขั้นตอนวิธี FHK บนโหนดเซต  $V$  ของกราฟ  $G$  ที่มีโหนดราก  $r$  กำหนดให้  $C_k^*$  แทนค่าใช้จ่ายของทัวรย้อยที่มากที่สุดในคำตอบที่ดีที่สุดของปัญหาการเดินทางของพนักงานขาย  $k$  คนบนเมตริกในกราฟ  $G$  และกำหนดให้  $e$  แทนอัตราส่วนที่ดีที่สุดของขั้นตอนวิธีแบบประมาณของปัญหาการเดินทางของพนักงานขายหนึ่งคน ขั้นตอนวิธี FHK รับประกันว่า

$$C_k \leq (e + 1 - 1/k)C_k^*$$

เมื่อใช้ขั้นตอนวิธีของ Christofides (1976) ในการหาทัวรของปัญหาการเดินทางของพนักงานขายหนึ่งคน จะได้ค่า  $e = 1.5$  ดังนั้นขั้นตอนวิธี FHK จะให้อัตราส่วนการประมาณของปัญหาการเดินทางของพนักงานขาย  $k$  คนบนเมตริกเป็น  $5/2 - 1/k$

### ปัญหาการการค้นหาแบบช่วง

งานวิจัยที่เกี่ยวกับการค้นหาข้อมูลแบบซับซ้อนในเครือข่ายเพียร์ทูเพียร์เป็นปัญหาที่น่าสนใจมาเป็นเวลานาน เริ่มต้นเมื่อ Harren *et al.* (2002) ได้กล่าวในงานวิจัยของพวกเขาว่าการค้นหาข้อมูลแบบซับซ้อนเป็นปัญหาเปิดที่สำคัญในเครือข่ายเพียร์ทูเพียร์ หลังจากนั้นได้มีงานวิจัยที่เกี่ยวกับการค้นหาในเครือข่ายเพียร์ทูเพียร์เป็นจำนวนมาก นอกจากนี้ Risson and Moors (2006) ได้ทำการสำรวจเกี่ยวกับการค้นหาวิธีต่างๆ ในเครือข่ายเพียร์ทูเพียร์

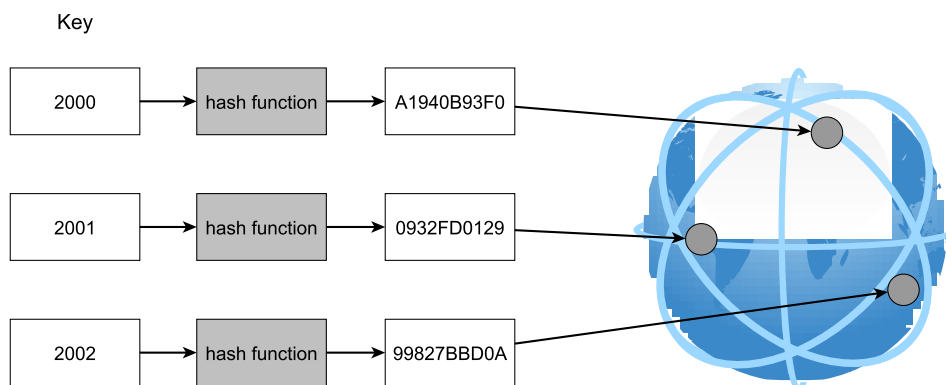


ภาพที่ 6 ตัวอย่างการทำงานของขั้นตอนวิธี FHK

ในวิทยานิพนธ์ฉบับนี้สนใจการค้นหาลักษณะช่วงซึ่งเป็นหนึ่งในวิธีการค้นหาที่ถูกใช้งานอย่างมากมายในหลายๆ โปรแกรมประยุกต์ รวมทั้งโปรแกรมประยุกต์ที่ทำงานบนเครือข่ายเพียร์ทูเพียร์เช่นกัน ตัวอย่างของการค้นหาลักษณะช่วงเช่นในระบบพีทูพีทีวี (P2PTV) ผู้ใช้งานต้องการค้นหาข้อมูลของรายการที่ผู้ชมอายุมากกว่า 13 ปีชมได้หรือในระบบบริการข้อมูล (Data management system) ผู้ใช้งานต้องการค้นหาข้อมูลของพนักงานที่มีรายได้ในช่วง 20000 บาทถึง 40000 บาท เป็นต้น

เนื่องจากในเครือข่ายเพียร์ทูเพียร์นั้นไม่มีเครื่องบริการทำให้มีการสร้างโครงสร้างข้อมูลมาช่วยในการค้นหาข้อมูล โครงสร้างข้อมูลที่ได้รับความนิยมได้แก่ตารางแฮชแบบกระจาย (DHTs) โดยตารางแฮชแบบกระจายจะนำเอาคุณสมบัติของวัตถุไปผ่านแฮชฟังก์ชัน จากนั้นจึงนำเอาค่าที่ได้จากการแฮชไปใช้ในการระบุตำแหน่งในเครือข่าย อย่างไรก็ตามการแฮชทำลายคุณสมบัติท้องถิ่น (Local property) นั่นคือคุณสมบัติที่อยู่ในลำดับติดกัน เมื่อผ่านแฮชฟังก์ชันแล้วค่าที่ได้ส่วนใหญ่ไม่อยู่ในลำดับติดกัน ตัวอย่างเช่นในภาพที่ 7 คุกกี้หมายเลข 2000 เมื่อผ่านแฮชฟังก์ชันแล้วได้ค่า A1940B93F0 แต่คุกกี้หมายเลข 2001 เมื่อผ่านแฮชฟังก์ชันแล้วได้ค่า 0932FD0129 เป็นต้น

งานวิจัยในช่วงแรกมุ่งไปที่การสร้างโครงสร้างข้อมูลที่ค้นหาลักษณะช่วงได้บนตารางแฮชแบบกระจายเช่น Prefix hash tree (PHT) จากงานของ Ramabhadran *et al.* (2004) เป็นการสร้าง Prefix tree บนตารางแฮชแบบกระจาย และ Distributed segment tree (DST) จากงานของ Zheng *et al.* (2006) เป็นการสร้าง Segment trees บนตารางแฮชแบบกระจาย ทั้งสองงานวิจัยมีข้อดีคือสามารถนำไปใช้งานกับตารางแฮชแบบกระจายใดๆ ได้โดยไม่ต้องปรับแก้ แต่การดำเนินการใดๆ ในโครงสร้างข้อมูลเหล่านี้



ภาพที่ 7 ตัวอย่างการค้นหาผ่านแฮชฟังก์ชัน

จะใช้เวลาในการทำงานเป็นสองชั้น นั่นคือชั้นแรกใช้เวลาในการค้นหาใน Binary search tree หรือ Segment tree จากนั้นในชั้นที่สองจะใช้เวลาในการค้นหาในตารางแฮชแบบกระจายอีกครั้งหนึ่ง

นอกจากนี้ยังมีโครงสร้างข้อมูลแบบอื่นที่รองรับการค้นหาแบบช่วงในเครือข่ายเพียร์ทูเพียร์ที่ไม่ได้ใช้ตารางแฮชแบบกระจาย ตัวอย่างเช่นโครงสร้างข้อมูล SkipNet จากงานของ Harvey *et al.* (2003) ซึ่งปรับปรุงมาจากโครงสร้างข้อมูล Skip Lists (Pugh (1990)) SkipNet รองรับการค้นหาแบบช่วงหรือรองรับการปรับลดการะอย่างใดอย่างหนึ่ง แต่ไม่สามารถทำทั้งสองอย่างพร้อมกันได้ โครงสร้างข้อมูล Skip Graphs จากงานของ Aspnes and Shah (2003) ได้ปรับปรุงมาจากโครงสร้างข้อมูล Skip Lists เช่นกัน Skip Graphs มีความสามารถคล้ายกับ Balanced tree ในเครือข่ายแบบกระจาย อีกทั้ง Skip Graphs รองรับการค้นหาแบบช่วงได้แต่ไม่รองรับการปรับลดการะบนจำนวนของวัตถุต่อโหนด

นอกจากนี้มีโครงสร้างข้อมูลหลายๆ แบบที่รองรับการค้นหาแบบช่วงอย่างมีประสิทธิภาพและได้แสดงประสิทธิภาพ โดยการทดลองว่ามีคุณสมบัติที่ดีในเรื่องของการปรับลดการะ ได้แก่ Mercury จากงานของ Bharambe *et al.* (2004), Baton จากงานของ Jagadish *et al.* (2005), Dak จากงานของ Yang and Hu (2006), Chordal graph จากงานของ Joungh (2008), Yarqs จากงานของ Zhang *et al.* (2009) เป็นต้น อย่างไรก็ตามงานวิจัยเหล่านี้ไม่ได้มีการรับประกันทางทฤษฎีเกี่ยวกับภาระของโหนดในระบบในโครงสร้างข้อมูลนั้นๆ เลย

## ปัญหาการปรับดุลภาระในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วง

ปัญหานี้มีสองกลุ่มนักวิจัยหลักที่ให้ความสนใจในทางทฤษฎีทั้งการค้นหาแบบช่วงและการปรับดุลภาระ กลุ่มแรกจะเป็นกลุ่มของ Karger and Ruhl (2004) กลุ่มที่สองเป็นกลุ่มของ Ganesan *et al.* (2004) ในส่วนนี้เราจะอธิบายงานวิจัยของ Karger and Ruhl (2004) ส่วนงานวิจัยของ Ganesan *et al.* (2004) เราจะอธิบายและทบทวนขั้นตอนวิธีในหัวข้อถัดไป เนื่องจากวิทยานิพนธ์ฉบับนี้ทำเปรียบเทียบงานวิจัยของ Ganesan *et al.* (2004)

ทั้งสองกลุ่มนักวิจัยใช้การดำเนินการพื้นฐาน 2 อย่างเหมือนกัน การดำเนินการอย่างแรกจะจัดการภาระของโหนดสองโหนดที่ติดกันให้สมดุล โดยการย้ายภาระของโหนดที่มีภาระมากกว่าไปยังโหนดที่มีภาระน้อยกว่า ในการดำเนินการอย่างที่สอง โหนด  $i$  จะย้ายภาระทั้งหมดของมันให้กับโหนดที่อยู่ติดกัน จากนั้นโหนด  $i$  จะย้ายตำแหน่งเพื่อไปแบ่งภาระของโหนด  $j$  ในตำแหน่งใหม่

งานวิจัย Karger and Ruhl (2004) ได้เสนอโพรโทคอลแบบสุ่ม ที่แต่ละโหนดจะสุ่มเลือกโหนดอื่นมาเพื่อเปรียบเทียบภาระกัน ถ้าภาระต่างกันมากจะมีการปรับดุลภาระ การสุ่มเรียกโหนดอื่นนี้จะเกิดขึ้นแม้ว่าจะไม่มีการใส่กุญแจหรือลบกุญแจที่โหนดนั้นก็ตาม พวกเขาได้แสดงว่าด้วยความน่าจะเป็นที่สูงถ้าแต่ละโหนดทำการติดต่อกับโหนดอื่นเพื่อทำการปรับดุลภาระเป็นจำนวน  $\Omega(\log n)$  ครั้งแล้วภาระของแต่ละโหนดจะอยู่ระหว่าง  $\frac{\epsilon}{16} \cdot L$  และ  $\frac{16}{\epsilon} \cdot L$  เมื่อ  $L$  แทนภาระเฉลี่ยของระบบและ  $\epsilon$  เป็นค่าคงที่ที่อยู่ระหว่าง  $0 < \epsilon < \frac{1}{4}$  ทั้งนี้ค่าคงที่ที่ซ่อนไว้ในสัญลักษณ์  $\Omega$  จะขึ้นอยู่กับค่า  $\epsilon$  นอกจากนี้พวกเขายังได้แสดงอีกว่าค่าใช้จ่ายของขั้นตอนการปรับดุลภาระสามารถถูกถ่วงเฉลี่ยไปกับค่าใช้จ่ายที่เป็นค่าคงที่ของการใส่กุญแจและลบกุญแจได้ โดยค่าคงที่ที่จะขึ้นอยู่กับค่า  $\epsilon$  เราพบว่าอัตราส่วนความระหว่างภาระสูงสุดต่อภาระต่ำสุดในงานของ Karger and Ruhl จะมีค่าเป็นอย่างน้อย 128

### ขั้นตอนวิธีของ Ganesan *et al.*

ในส่วนนี้เราจะอธิบายงานวิจัย Ganesan *et al.* (2004) โดยเราจะกล่าวถึงอัตราส่วนความไม่สมดุล ค่าใช้จ่ายในการปรับดุลภาระ การดำเนินการพื้นฐาน 2 อย่างและขั้นตอนวิธี ADJUSTLOAD

เมื่อกุญแจถูกใส่เข้าไปในระบบหรือถูกลบออกจากระบบ โหนดที่รับผิดชอบช่วงของกุญแจนั้นจำนวนกุญแจที่รับผิดชอบมีการเปลี่ยนแปลง ดังนั้นจะต้องปรับปรุงข้อมูล หลังจากนั้นขั้นตอนวิธีการ

ปรับดุลภาระจะถูกเรียกโดยมีจุดมุ่งหมายเพื่อรักษาอัตราส่วนของภาระของโหนดที่สูงสุดและภาระของโหนดที่ต่ำสุดให้มีค่าน้อย อัตราส่วนนี้ถูกเรียกว่าอัตราส่วนความไม่สมดุล (Imbalance ratio) ซึ่งจะใช้สัญลักษณ์ว่า  $\sigma$

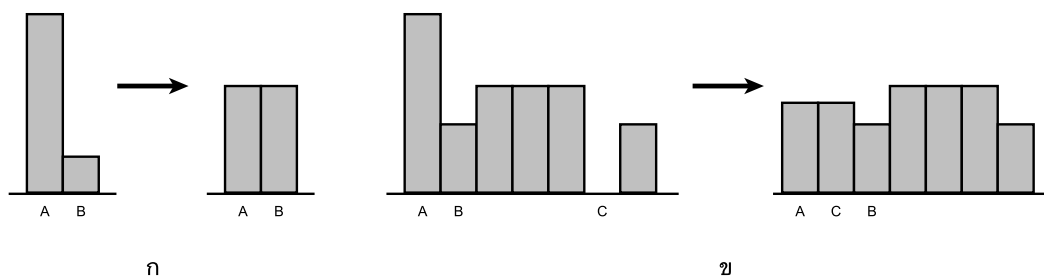
กำหนดให้  $L(u)$  แทนภาระของโหนด  $u$  หรือจำนวนของกุญแจที่ถูกเก็บไว้ที่โหนด  $u$  ณ เวลาใดๆ ก็ตามในระบบ เราจะกล่าวว่าขั้นตอนวิธีการปรับดุลภาระจะรับประกันอัตราส่วนความไม่สมดุลถ้าหลังจากการใส่กุญแจเข้าไปในระบบหรือลบกุญแจออกจากระบบในแต่ละครั้งและตามด้วยการทำงานของขั้นตอนวิธีการปรับดุลภาระแล้ว  $\max_u L(u) \leq \sigma \cdot \min_u L(u) + c_0$  สำหรับค่าคงที่  $c_0$  บางค่า

งานวิจัยของ Ganesan *et al.* (2004) มีจุดมุ่งหมายคือต้องการพัฒนาขั้นตอนวิธีการปรับดุลภาระที่รับประกันอัตราส่วนความไม่สมดุลที่มีค่าเป็นค่าคงที่และมีค่าใช้จ่ายต่ำเฉลี่ยต่อการใส่หรือลบกุญแจที่ต่ำ Ganesan *et al.* (2004) ได้พิจารณาค่าใช้จ่ายในการปรับดุลภาระ 3 ชนิดได้แก่

1. การย้ายข้อมูล ในการดำเนินการแต่ละครั้ง ถ้ามีการย้ายกุญแจจากโหนดหนึ่งไปเก็บไว้ที่อีกโหนดหนึ่งจะนับค่าใช้จ่ายเป็นหนึ่งหน่วย
2. การเปลี่ยนขอบเขต เมื่อขั้นตอนการปรับดุลภาระถูกเรียกใช้ ช่วงของกุญแจที่แต่ละโหนดรับผิดชอบอาจจะเปลี่ยนแปลง การเปลี่ยนแปลงนี้จะต้องถูกแพร่ข่าวออกไปในระบบ เพื่อที่ว่าการใส่กุญแจหรือลบกุญแจครั้งต่อไปจะได้เกิดขึ้นยังโหนดที่ถูกต้อง
3. การสอบถามข้อมูลของภาระ การทำงานนี้จะเกิดขึ้นเมื่อมีโหนดร้องขอข้อมูลแบบครอบคลุม (Global information) ได้แก่ ข้อมูลของภาระที่สูงที่สุดและข้อมูลของภาระต่ำที่สุด

ค่าใช้จ่ายของขั้นตอนวิธีการปรับดุลภาระจะวัดจากจำนวนของกุญแจที่ถูกย้ายต่อการใส่หรือลบกุญแจ โดยในงานวิจัยของ Ganesan *et al.* (2004) สนใจค่าใช้จ่ายแบบถัวเฉลี่ย ทั้งนี้ค่าใช้จ่ายแบบถัวเฉลี่ยของการใส่หรือลบกุญแจจะมีค่าเป็น  $c$  ถ้าสำหรับลำดับใดๆ ของการใส่หรือลบกุญแจ  $t$  ครั้ง จำนวนของกุญแจที่ถูกย้ายมีจำนวนไม่เกิน  $tc$

Ganesan *et al.* (2004) ได้เสนอขั้นตอนวิธี ADJUSTLOAD โดยขั้นตอนวิธี ADJUSTLOAD นี้



ภาพที่ 8 การดำเนินการ (ก) NbrAdjust (ข) Reorder

ใช้การดำเนินการพื้นฐาน 2 อย่างได้แก่ NBRADJUST และการดำเนินการ REORDER ทั้งนี้การดำเนินการพื้นฐานมีโหนดที่เกี่ยวข้องหลายโหนด เราจะนิยามโหนดเพื่อนบ้าน (Neighbor) ของโหนด  $u$  ใดๆ ว่าเป็นโหนดที่อยู่ในลำดับถัดกันทางด้านซ้ายหรือด้านขวาของโหนด  $u$  ในปริภูมิกุญแจ สำหรับการดำเนินการ NBRADJUST และการดำเนินการ REORDER นิยามได้ดังนี้

การดำเนินการ NbrAdjust: โหนด  $N_i$  จะย้ายภาระของมันให้กับโหนดเพื่อนบ้าน  $N_{i+1}$  การกระทำดังกล่าวอาจทำให้เกิดการเปลี่ยนแปลงของขอบเขต  $R_i$  ของโหนด  $N_i$  และโหนด  $N_{i+1}$

การดำเนินการ Reorder: พิจารณาโหนด  $N_i$  ที่มีช่วงรับผิดชอบเป็นช่วงว่าง  $[R_i, R_i]$  จะย้ายตำแหน่งของโหนด  $N_i$  เพื่อไปแบ่งช่วงของโหนด  $N_j$  หลังจากนั้นช่วง  $[R_j, X]$  จะถูกรับผิดชอบโดยโหนด  $N_j$  ในขณะที่ช่วง  $[X, R_{j+1}]$  จะถูกรับผิดชอบโดยโหนด  $N_i$  สำหรับค่า  $X, R_j \leq X \leq R_{j+1}$  บางค่า สุดท้ายแต่ละโหนดจะเปลี่ยนชื่อตามความเหมาะสม

สำหรับค่าคงที่  $c$  และ  $\delta$  บางค่า Ganesan *et al.* (2004) นิยามลำดับของระดับกั้น (Thresholds)

$$T_m = \lfloor c\delta^m \rfloor$$

สำหรับทุกๆ ค่า  $m \geq 1$  ระดับกั้นนี้จะถูกใช้เป็นเงื่อนไขในการเรียกขั้นตอนวิธี ADJUSTLOAD ให้ทำงาน เมื่อกำหนดให้ค่า  $\delta = 2$  พวกเขาจะเรียกขั้นตอนวิธีนี้ว่า Doubling Algorithm ทั้งนี้ขั้นตอนวิธี ADJUSTLOAD ยังคงทำงานได้เมื่อค่า  $\delta > \phi = \frac{(1+\sqrt{5})}{2} \approx 1.618$  ซึ่งเป็นค่า Golden ratio พวกเขาเรียกขั้นตอนวิธีที่ทำงานด้วยค่า Golden ratio ว่า Fibbing Algorithm พวกเขาได้พิสูจน์ว่าขั้นตอนวิธี ADJUSTLOAD ที่ทำงานด้วยอัตราเหล่านั้นสามารถรับประกันอัตราส่วนความไม่สมดุล  $\sigma$  ว่ามีค่า  $\delta^3$  (มีค่า

ประมาณ 4.237)

เมื่อกำหนดลำดับของระดับกันมาให้ ขั้นตอนวิธี ADJUSTLOAD จะทำงานดังนี้ เมื่อภาระของ โหนด  $N_i$  ข้ามผ่านระดับกัน  $T_m$  แล้วขั้นตอนวิธีการปรับดุลภาระจะถูกเรียกที่โหนด  $N_i$  กำหนดให้  $N_j$  เป็นโหนดเพื่อนบ้านของโหนด  $N_i$  ที่มีภาระน้อยที่สุด ถ้าภาระของโหนด  $N_j$  มีค่าไม่มากจะเรียก การดำเนินการ NBRADJUST ตามด้วยการเรียกขั้นตอนวิธี ADJUSTLOAD ซ้ำที่โหนด  $N_i$  และโหนด  $N_j$  แต่ถ้าภาระของโหนด  $N_j$  มีค่ามากจะพิจารณาภาระของโหนดที่มีภาระน้อยที่สุด  $N_k$  ถ้าอัตราส่วน ระหว่าง  $N_j$  และ  $N_k$  ต่างกันมากจะเรียกการดำเนินการ REORDER โดยเริ่มต้นโหนด  $N_k$  จะย้ายภาระ ทั้งหมดให้กับโหนดเพื่อนบ้านที่มีภาระน้อยที่สุด  $N$  จากนั้นการดำเนินการ REORDER จะถูกเรียกที่ โหนด  $N_k$  และโหนด  $N_i$  พายที่สุดจะมีการเรียกขั้นตอนวิธี ADJUSTLOAD อีกครั้งที่โหนด  $N$  ขั้นตอน วิธี ADJUSTLOAD สำหรับการใส่กุญแจถูกแสดงดังภาพที่ 9

ส่วนกรณีการลบกุญแจสามารถทำได้ในลักษณะคล้ายกัน เมื่อภาระของโหนด  $N_i$  ลดลงข้าม ระดับกัน  $T_j$  ถ้าภาระของโหนดเพื่อนบ้าน  $N_j$  มีค่ามากกว่า  $T_{j+1}$  แล้วโหนด  $N_i$  จะปรับดุลภาระกับ โหนดเพื่อนบ้านด้วยการดำเนินการ NBRADJUST ตามด้วยการเรียกขั้นตอนวิธี ADJUSTLOAD ซ้ำที่ โหนด  $N_i$  และโหนด  $N_j$  แต่ถ้าภาระของโหนดเพื่อนบ้านมีค่าน้อยจะพิจารณาโหนดที่มีภาระสูงสุด  $N_k$  ถ้า  $N_k$  มีภาระมากกว่า  $T_{j+2}$  โหนด  $N_i$  จะโอนภาระทั้งหมดให้โหนด  $N_j$  แล้วไปแบ่งครึ่งภาระของ โหนด  $N_k$  จากนั้นมีการเรียกขั้นตอนวิธี ADJUSTLOAD ซ้ำที่โหนด  $N_j$

อย่างไรก็ตามแม้ว่าการดำเนินการพื้นฐานทั้งสองอย่างได้แก่ NBRADJUST และ REORDER นั้น จะง่ายต่อการอธิบาย แต่ขั้นตอนวิธี ADJUSTLOAD เป็นขั้นตอนวิธีแบบ recursive ซึ่งมีข้อจำกัดหลาย ประการเมื่อนำไปประยุกต์ใช้งานจริง อีกทั้งในกรณีที่แย่ที่สุดในการเพิ่มหรือลบกุญแจ ขั้นตอนวิธี ADJUSTLOAD ไม่ได้รับประกันจำนวนครั้งของการดำเนินการที่ถูกเรียก ไม่ได้รับประกันจำนวนข้อมูลที่ ต้องถูกปรับให้ทันสมัย (การเปลี่ยนตัวแบ่งเขตและข้อมูลภาระ) และไม่ได้รับประกันจำนวนของโหนด ที่ได้รับผลกระทบ นอกจากนี้การดำเนินการปรับดุลภาระแต่ละครั้งมีการเรียกดูข้อมูลแบบครอบคลุม ดังนั้นจำนวนครั้งของการเรียกดูข้อมูลแบบครอบคลุมอาจจะมากกว่าจำนวนครั้งของการใส่กุญแจหรือ ลบกุญแจ

---

**Procedure** ADJUSTLOAD( $u$ ) FOR INSERTION

```

1: Let  $L(N_i) = x \in (T_m, T_{m+1}]$ .
2: Let  $N_j$  be the lighter loaded neighbor of  $N_i$ .
3: if  $L(N_j) \leq T_{m-1}$  then
4:   {Do NBRADJUST}
5:   Move tuples from  $N_i$  to  $N_j$  to equalize load.
6:   ADJUSTLOAD( $N_j$ ).
7:   ADJUSTLOAD( $N_i$ ).
8: else
9:   Find the least-loaded node  $N_k$ .
10:  if  $L(N_k) \leq T_{m-2}$  then
11:    {Do REORDER}
12:    Transfer all data from  $N_k$  to  $N = N_{k \pm 1}$ .
13:    Transfer data from  $N_i$  to  $N_k$ , s.t.,  $L(N_i) = \lceil x/2 \rceil$  and  $L(N_k) = \lfloor x/2 \rfloor$ 
14:    ADJUSTLOAD( $N$ ).
15:    {Rename nodes appropriately.}
16:  end if
17: end if

```

---

ภาพที่ 9 ขั้นตอนวิธี ADJUSTLOAD ในกรณีการใส่กุญแจ

## อุปกรณ์และวิธีการ

### อุปกรณ์

1. เครื่องคอมพิวเตอร์ Intel® Pentium® Core2 Duo CPU 2.26GHz 3 GB of RAM
2. ระบบปฏิบัติการ Microsoft Windows 7
3. ซอฟต์แวร์ MikTex
4. ซอฟต์แวร์ TeXworks
5. อุปกรณ์สำนักงาน

### วิธีการ

ในวิทยานิพนธ์ส่วนแรก เราจะแสดงการวางแผนการส่งข้อมูลแบบกระแสในเครือข่ายเพียร์ทูเพียร์ โดยมีเงื่อนไขว่าแต่ละเครื่องในเครือข่ายมีความสามารถในการส่งต่อข้อมูลได้แตกต่างกัน เราสนใจ MDM problem ในกรณีที่แบนด์วิธของเส้นเชื่อมทุกเส้นเป็นจำนวนเท่าของแบนด์วิธที่ต้องการ เราได้พิสูจน์ว่าในกรณีพิเศษนี้มีคำตอบที่ดีที่สุดเป็นต้นไม้อยู่ เราจึงลดรูปปัญหาไปเป็นการออกแบบวิธีการสร้างต้นไม้อย่างมีประสิทธิภาพที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขตในกรณีที่ขอบเขตดีกรีไม่เท่ากัน ดังนั้นเราจะอธิบายขั้นตอนวิธีสำหรับการสร้างต้นไม้อย่างมีประสิทธิภาพที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขตในกรณีที่ขอบเขตดีกรีไม่เท่ากัน ในวิทยานิพนธ์ส่วนที่สองเราจะแสดงการออกแบบวิธีการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วง โดยเราพิจารณากรณีที่เกิดขึ้นจากการใช้งานจริงสองกรณี กรณีแรกเป็นกรณีที่มีแต่การใส่กุญแจ เราจะแสดงการดำเนินการ MINBALANCE ที่จะถูกใช้งานเมื่อมีการใส่กุญแจ กรณีที่สองเป็นกรณีที่ทั้งการใส่กุญแจและลบกุญแจ เราจะแสดงการดำเนินการ SPLIT ที่จะถูกใช้งานเมื่อมีการลบกุญแจเกิดขึ้น

#### 1. การวางแผนการส่งข้อมูลแบบกระแสในเครือข่ายเพียร์ทูเพียร์

เนื่องจากเราได้พิสูจน์ในบทตั้ง 1 ในส่วนผลและวิจารณ์ว่า ถ้าแบนด์วิธของทุกเส้นเชื่อมมีค่าเป็นจำนวนเท่าของแบนด์วิธที่ต้องการ  $s$  และความล่าช้ามีคุณสมบัติเป็นเมตริกแล้วคำตอบของ MDM

problem จะสามารถเปลี่ยนให้เป็นต้นไม้ที่มีเส้นผ่านกลางไม่เพิ่มขึ้นจากคำตอบต้นฉบับ ด้วยความจริงนี้ทำให้เราสามารถลดรูปปัญหาไปเป็นปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขตในกรณีที่ขอบเขตดีกรีไม่เท่ากันได้ โดยเราจะกำหนดขอบเขตดีกรี  $B_v = U_v + 1$  ให้กับแต่ละโหนด  $v \in V - \{r\}$  และกำหนดขอบเขตดีกรี  $B_r = U_r$  สำหรับเครื่องต้นทาง  $r$

### 1.1 ขั้นตอนวิธีสำหรับ BDST with non-uniform degree bound

เราจะแสดงขั้นตอนวิธีสำหรับการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขตในกรณีที่ขอบเขตดีกรีไม่เท่ากัน สำหรับขอบเขตดีกรีของโหนด  $v$  ใดๆ เราจะใช้สัญลักษณ์เป็น  $B_v$  เราสมมติว่ามีโหนด  $v$  ที่มีขอบเขตดีกรี  $B_v > 2$  เราจะเลือกโหนดนี้เป็นโหนดราก  $r$  ซึ่งข้อสมมตินี้เราจะแสดงว่าสามารถยกเลิกได้ในหัวข้อย่อย 2.5 ในส่วนผลและวิจารณ์ กำหนดให้  $\Delta$  แทนค่าความยาวของเส้นผ่านกลางของคำตอบที่ดีที่สุด ซึ่งเราสมมติว่าขั้นตอนวิธีรู้ค่า  $\Delta$  ทั้งนี้ค่า  $\Delta$  จะอยู่ในช่วง  $[\max_{e \in E} \ell_e, n \cdot \max_{e \in E} \ell_e]$  เราสามารถใช้การค้นหาแบบทวิภาค (Binary search) เพื่อหาค่าที่เหมาะสมได้ นั่นคือค่าที่นำมาใช้จะมีค่ามากกว่าค่าที่ดีที่สุดไม่เกินสองเท่า

ขั้นตอนวิธีที่เสนอนี้จะหาต้นไม้ทอดข้าม  $T$  ที่มีโหนดราก  $r$  โดยที่ระยะทางจากโหนด  $r$  ไปยังโหนดใดๆ ใน  $T$  จะไม่เกิน  $O(\sqrt{\log n})\Delta$  โดยเราจะแสดงบทพิสูจน์ต่างๆ ในส่วนผลและวิจารณ์

ในการสร้างคำตอบเราจะแบ่งโหนด  $V$  ออกเป็น 3 เซต  $V_1, V_2$  และ  $V_3$  ดังนี้

1. สำหรับ  $i \in \{1, 2\}$ ,  $V_i$  เป็นเซตของโหนด  $v$  ที่  $B_v = i$
2.  $V_3$  เป็นเซตของโหนด  $v$  ที่  $B_v > 2$  (สังเกตว่าโหนด  $r \in V_3$  จากข้อสมมติ)

ขั้นตอนวิธีของเราจะจัดการโหนดแต่ละกลุ่มแยกกัน แสดงได้ดังภาพที่ 10

สำหรับโหนดในกลุ่ม  $V_3$  เราจะสร้างต้นไม้  $H$  โดยเราจะลดรูปปัญหาจากกรณีที่มีเงื่อนไขว่าแต่ละโหนดมีขอบเขตดีกรีไม่เท่ากันมาเป็นกรณีที่แต่ละโหนดมีขอบเขตดีกรีเป็น 3 การลดรูปทำดังนี้ สำหรับแต่ละโหนด  $v \in V$  เราจะสร้างโหนด  $v^1, v^2, \dots, v^{B_v-2}$  ซึ่งเปรียบเสมือนสำเนาของโหนด  $v$  จำนวน  $B_v - 2$  โหนดและแทนเซตของโหนดเหล่านี้ด้วย  $c(v)$  กำหนดให้  $V' = \cup_{v \in V} c(v)$  และนิยามฟังก์ชันการส่ง  $f : V' \rightarrow V$  ที่  $f(u) = v$  ก็ต่อเมื่อ  $u \in c(v)$  ส่วนเมตริก  $d'$  สามารถนิยามได้ด้วย

สำหรับ  $u, v \in V'$  ใดๆ ให้  $d'(u, v) = d(f(u), f(v))$  สังเกตได้ว่าสำหรับโหนด  $v$  ใดๆ ระยะทางระหว่างสองโหนดใดๆ ใน  $c(v)$  ภายใต้  $d'$  จะมีค่าเป็นศูนย์ สุดท้ายเราจะกำหนดค่าขอบเขตดีกรีของทุกโหนด  $v \in V'$  เป็น 3 จากนั้นเรียกใช้ขั้นตอนวิธี KLS ผลลัพธ์ที่ได้เป็นต้นไม้ทอดข้ามบนเซตของโหนด  $V'$  เราใช้บทตั้ง 3 ซึ่งพิสูจน์ในส่วนผลและวิจารณ์แปลงคำตอบจากเซตของโหนด  $V'$  กลับมาเป็น  $V$  โดยที่ความยาวของเส้นผ่านกลางไม่เพิ่มขึ้น ผลลัพธ์ที่ได้เราจะเรียกว่า ส่วนฐาน (base) ของคำตอบ ซึ่งเราพิสูจน์ในบทผลและวิจารณ์ว่าเส้นผ่านกลางของ  $H$  จะไม่เกิน  $O(\sqrt{\log n})\Delta$

สำหรับโหนดในกลุ่ม  $V_2$  เรากำหนดให้  $K$  คือจำนวนของดีกรีที่ยังไม่ถูกใช้ของส่วนฐาน เราจะเรียกขั้นตอนวิธี FHK เพื่อแบ่งเซตของโหนด  $V$  ออกเป็น  $K$  ทัวรย่อยที่มีโหนดราก  $r$  ในแต่ละทัวร เราจะสร้างเซตของเส้นทางโดยแต่ละทัวรย่อย เราจะท่องไปในทัวรย่อยเริ่มที่โหนดราก จากนั้นข้ามโหนดในเซต  $V_3 - \{r\}$  และโหนดในเซต  $V_1$  ในขั้นสุดท้ายเราจะนำเอาโหนด  $r$  ออกจากเส้นทาง ผลลัพธ์จะได้เซตของเส้นทาง  $K$  เส้น  $L_1, L_2, \dots, L_K$  ซึ่งเราจะเรียกว่าส่วนหาง (tails) ของคำตอบ สังเกตได้ว่าแต่ละเส้นทางจะมีเฉพาะโหนดในเซตของ  $V_2$  เราพิสูจน์ในบทผลและวิจารณ์ว่าความยาวของแต่ละเส้นนั้นไม่เกิน  $O(\Delta)$

สุดท้ายเราจะเชื่อมแต่ละ โหนดในเซต  $V_1$  ซึ่งเป็นโหนดใบต่อกับส่วนท้ายของส่วนหางแต่ละเส้นทาง  $L_i$  สำหรับค่า  $1 \leq i \leq K$  จากนั้นนำส่วนหางไปเชื่อมต่อกับโหนดในส่วนฐานที่มีดีกรีที่ไม่ถูกใช้

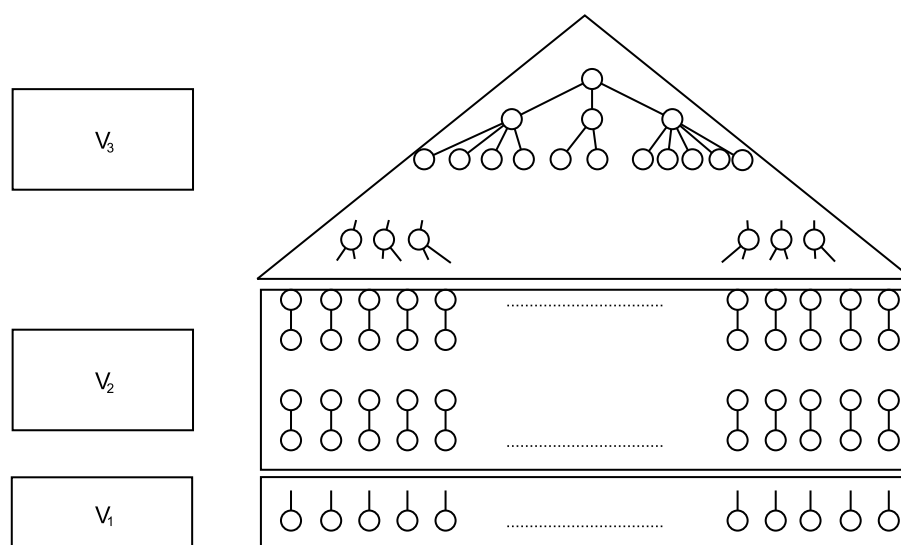
ภาพรวมของขั้นตอนวิธีในการสร้าง BDST with non-uniform degree bound แสดงดังภาพที่ 11 โดยโหนดเซต  $V_3$  จะถูกสร้างเป็นต้นไม้  $H$  โหนดเซต  $V_2$  จะถูกสร้างเซตของเส้นทางที่มีจำนวนเท่ากับผลรวมของดีกรีที่ไม่ถูกใช้ของต้นไม้  $H$  และ โหนดเซต  $V_1$  แต่ละ โหนดจะถูกนำมาเชื่อมส่วนท้ายของเส้นทางที่ได้จาก  $V_2$  แต่ละเส้น

---

**Procedure** BDST with non-uniform degree bound

- 1: Let  $V_1$  be the set of nodes  $v \in V$  with  $B_v = 1$ .
  - 2: Let  $V_2$  be the set of nodes  $v \in V$  with  $B_v = 2$ .
  - 3: Let  $V_3$  be the set of nodes  $v \in V$  with  $B_v > 2$ .
  - 4: //base construction
  - 5: For each node  $v \in V_3$ , create a set of nodes  $c(v) = \{v^1, v^2, \dots, v^{B_v-2}\}$ .
  - 6: Let  $V' = \cup_{v \in V_3} c(v)$ .
  - 7: Define a mapping function  $f : V' \rightarrow V$  to be such that  $f(u) = v$  iff  $u \in c(v)$ .
  - 8: Define the metric  $d'$ : for any  $u, v \in V'$ , let  $d'(u, v) = d(f(u), f(v))$ .
  - 9: Assign degree bound of every node  $v \in V'$  to 3.
  - 10: Using the *KLS* algorithm to construct a tree  $H'$  from  $V'$  using metric  $d'$ .
  - 11: Convert a tree  $H'$  to  $H$  which uses a node set  $V$  by Lemma 3
  - 12: //tails construction
  - 13: Let  $K$  be the total amount of unused degree in the base.
  - 14: Using the *FHK* algorithm to partition  $V$  into  $K$  tours.
  - 15: For each tour, traverse the tour starting at the root node  $r$ , and skipping over nodes in  $V_3 - \{r\}$  and  $V_1$ . Then remove  $r$  from the path. We got a set of paths,  $L_1, L_2, \dots, L_K$ .
  - 16: //combine
  - 17: Attach each path  $L_i$  to nodes in  $H$  which has unused degree for  $1 \leq i \leq K$ .
  - 18: Attach each node in  $V_1$  to the end of each tail.
- 

**ภาพที่ 10** ขั้นตอนวิธีสำหรับ BDST with non-uniform degree bound



ภาพที่ 11 ตัวอย่างการ สร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและคิรีมีขอบเขตในกรณีที่มีขอบเขตคิรีไม่เท่ากัน โดยแบ่งกลุ่มของโหนดออกเป็น 3 กลุ่ม

## 2. การออกแบบวิธีการปรับดูแลในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วง

สมมติว่ามีเครื่อง central directory ที่เก็บข้อมูลขอบเขตช่วง  $R_0 \leq R_1 \leq \dots \leq R_n$  และสามารถส่งคำร้องขอการเพิ่มกุญแจและการลบกุญแจไปยังโหนดที่ถูกต้องได้ การมีเครื่อง central directory ทำให้แต่ละโหนดสามารถเรียกขอข้อมูลหรือปรับปรุงข้อมูลใช้ค่าใช้จ่าย 1 หน่วย ทั้งนี้สามารถยกเลิกการใช้งาน central directory ได้โดยใช้โครงสร้างข้อมูล Skip graphs ซึ่งอธิบายในหัวข้อ 3.5 ของบทผลและวิจารณ์

เราจะพิจารณากรณีที่เกิดขึ้นในการใช้งานจริงสองกรณีด้วยกันคือ กรณีที่มีแต่การใส่กุญแจอย่างเดียวและกรณีทั่วไปที่มีทั้งการใส่และการลบกุญแจ เราสมมติว่าเริ่มต้นแต่ละโหนดมีภาระเก็บไว้จำนวน  $c_0$  โดยที่  $c_0$  เป็นค่าคงที่บางค่าซึ่งจำเป็นต่อการพิสูจน์ ไม่เช่นนั้นจะทำให้คำนวณอัตราส่วนความไม่สมดุลไม่ได้

เนื่องจากรูปแบบในการเก็บข้อมูลตามกุญแจมีลักษณะเป็นเชิงเส้น ดังนั้นเมื่อพิจารณาตามช่วงที่แต่ละโหนดรับผิดชอบ ณ เวลาใดๆ ก็ตามในระบบ โหนดในระบบจะมีลำดับซึ่งลำดับเหล่านี้จะใช้นิยามความสัมพันธ์ซ้าย (left) และขวา (right) ระหว่างโหนดได้ ซึ่งความสัมพันธ์นี้มีความสำคัญต่อการ

วิเคราะห์เป็นอย่างมาก

## 2.1 ขั้นตอนวิธีในกรณีที่มีแค่การใส่กุญแจ

กรณีที่มีแค่การใส่กุญแจนี้ง่ายต่อการวิเคราะห์และให้แนวคิดเกี่ยวกับการจัดการในกรณีทั่วไป อีกทั้งกรณีนี้มีความน่าสนใจเนื่องจากในหลายโปรแกรมประยุกต์การลบกุญแจแทบจะไม่เกิดขึ้นเลยเช่น การแชร์ไฟล์ (file sharing) เป็นต้น

ในกรณีนี้เราจะอธิบายการดำเนินการ MINBALANCE ซึ่งใช้การดำเนินการพื้นฐานสองอย่างได้แก่การดำเนินการ NBRADJUST และ REORDER เช่นเดียวกับงานวิจัยของ Ganesan *et al.* อย่างไรก็ตามขั้นตอนวิธีที่นำเสนอนี้ง่ายกว่าขั้นตอนวิธี ADJUSTLOAD ของ Ganesan *et al.*

มีข้อสังเกตว่าการดำเนินการ MINBALANCE ไม่ได้เป็นการดำเนินการเรียกตัวเองและจำนวนครั้งในการเรียกใช้งานการดำเนินการพื้นฐานคือการดำเนินการ NBRADJUST และ REORDER อย่างละหนึ่งครั้ง เราได้พิสูจน์อัตราส่วนความไม่สมดุลว่ามีค่าเป็น  $3 + \sqrt{5} \approx 5.237$  อีกทั้งยังแสดงว่าค่าใช้จ่ายเฉลี่ยของการใส่กุญแจเป็นค่าคงที่ ในหัวข้อย่อย 2.2 ของส่วนผลและวิจารณ์

### การดำเนินการ MINBALANCE

หลังจากที่มีการใส่กุญแจเกิดขึ้นที่โหนด  $u$  ใดๆ การดำเนินการ MINBALANCE จะถูกเรียกสมมติว่าโหนดที่มีภาระน้อยที่สุดคือโหนด  $v$  ถ้าภาระของโหนด  $u$  มีค่ามากกว่า  $\alpha$  เท่าของภาระของโหนด  $v$  โหนด  $v$  จะโอนภาระทั้งหมดของมันให้กับโหนดเพื่อนบ้านที่ติดกับมันที่มีภาระน้อยที่สุด หลังจากนั้นโหนด  $v$  จะย้ายตำแหน่งและแบ่งภาระครึ่งหนึ่งจากโหนด  $u$  เราจะเรียกขั้นตอนต่างๆ เหล่านี้ว่า การดำเนินการ MINBALANCE สังเกตได้ว่าการดำเนินการนี้จะต้องการข้อมูลของภาระที่ต่ำที่สุด ดังนั้นจะต้องมีวิธีในการเรียกค้นข้อมูลนี้จากระบบด้วย

---

**Procedure** MINBALANCE ( $u$ )

- 1: Let  $v$  be the minimum-loaded node in the system.
  - 2: **if**  $L(u) > \alpha \cdot L(v)$  **then**
  - 3:   //MINBALANCE steps
  - 4:   Let  $z$  be the lightly-loaded neighbor of  $v$ .
  - 5:   Transfer all keys of  $v$  to  $z$ .
  - 6:   Transfer a half-load of  $u$  to  $v$ , s.t.,  $L(u) = \left\lceil \frac{L(u)}{2} \right\rceil$  and  $L(v) = \left\lfloor \frac{L(u)}{2} \right\rfloor$ .
  - 7:   Rename nodes appropriately.
  - 8: **end if**
- 

ภาพที่ 12 การดำเนินการ MINBALANCE

## 2.2 ขั้นตอนวิธีในกรณีทั่วไป

สำหรับกรณีที่รองรับทั้งการใส่กุญแจและการลบกุญแจ หลังจากที่มีการใส่กุญแจหรือลบกุญแจแล้วขั้นตอนวิธีการปรับดุลภาระจะถูกเรียก สำหรับการใส่กุญแจจะเรียกการดำเนินการ MINBALANCE ที่ได้นำเสนอไว้ในหัวข้อก่อนหน้า ส่วนการลบกุญแจเราเสนอการดำเนินการ SPLIT ซึ่งบทพิสูจน์จะแสดงในหัวข้อย่อย 2.3 ของส่วนผลและวิจารณ์

### การดำเนินการ SPLIT

การดำเนินการ SPLIT จะถูกเรียกเมื่อมีการลบกุญแจเกิดขึ้นที่โหนด  $u$  ใดๆ กำหนดให้โหนด  $z$  เป็นโหนดเพื่อนบ้านที่มีภาระน้อยที่สุดของ  $u$  ขั้นตอนวิธีการปรับดุลภาระจะถูกเรียกเมื่อภาระของโหนด  $u$  มีค่าน้อยกว่า  $\frac{1}{\beta}$  เท่าของภาระที่สูงสุดในระบบ โดยมีขั้นตอนปรับดุลภาระอยู่สองชนิดขึ้นอยู่กับภาระของโหนด  $z$  ถ้าภาระของโหนด  $z$  มีค่ามากกว่า  $\frac{2}{\beta}$  เท่าของภาระที่สูงสุด โหนด  $u$  และโหนด  $z$  จะเฉลี่ยภาระกันทำให้ภาระของทั้งสองโหนดเท่ากัน เราเรียกขั้นตอนเหล่านี้ว่า SPLITNBR ในอีกกรณีหนึ่งถ้าภาระของโหนด  $z$  มีไม่เกิน  $\frac{2}{\beta}$  เท่าของภาระที่สูงสุด โหนด  $u$  โอนภาระของมันทั้งหมดให้กับโหนด  $z$  หลังจากนั้นโหนด  $u$  จะย้ายตำแหน่งไปแบ่งภาระครึ่งหนึ่งของโหนดที่มีภาระที่สูงสุด ขั้นตอนเหล่านี้จะเรียกว่า SPLITMAX ซึ่งแสดงได้ดังรูปที่ 13 สังเกตได้ว่า SPLITNBR จะเรียกการดำเนินการ NBRADJUST เพียงอย่างเดียว แต่ SPLITMAX จะเรียกทั้งการดำเนินการ NBRADJUST และการดำเนินการ REORDER สังเกตได้ว่าการดำเนินการนี้จะต้องการข้อมูลของภาระที่สูงที่สุด ดังนั้นจะต้องมีวิธีการเรียกค้นข้อมูลนี้จากในระบบด้วยเช่นกัน

---

**Procedure SPLIT ( $u$ )**

- 1: Let  $w$  be the maximum-loaded node in the system.
  - 2: **if**  $L(u) \leq \frac{L(w)}{\beta}$  **then**
  - 3:   Let  $z$  be the lightly-loaded neighbor of  $u$ .
  - 4:   **if**  $L(z) \leq 2 \cdot \frac{L(w)}{\beta}$  **then**
  - 5:     //SPLITMAX steps
  - 6:     Transfer all keys of  $u$  to  $z$ .
  - 7:     Transfer a half-load of  $w$  to  $u$ , s.t.,  $L(w) = \left\lceil \frac{L(w)}{2} \right\rceil$  and  $L(u) = \left\lfloor \frac{L(w)}{2} \right\rfloor$ .
  - 8:   **else**
  - 9:     //SPLITNBR steps
  - 10:    Move keys from  $z$  to  $u$  to equalize load, s.t.,  $L(u) = \left\lceil \frac{(L(u)+L(z))}{2} \right\rceil$  and  $L(z) = \left\lfloor \frac{(L(u)+L(z))}{2} \right\rfloor$ .
  - 11:   **end if**
  - 12:   {Rename nodes appropriately.}
  - 13: **end if**
- 

**ภาพที่ 13** การดำเนินการ SPLIT

## ผลและวิจารณ์

ในส่วนผลและวิจารณ์นี้จะเป็นส่วนที่แสดงบทพิสูจน์ต่างๆ ที่เกี่ยวข้องกับปัญหาวิจัยทั้งสองปัญหา เริ่มต้นเราจะพิสูจน์ว่าภายใต้เงื่อนไขที่ว่าแบนด์วิดท์ของเส้นเชื่อมทุกเส้นเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการนั้น MDM problem จะมีคำตอบที่ดีที่สุดเป็นต้นไม้ ต่อมาเราจะแสดงบทพิสูจน์ที่เกี่ยวกับปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดีกรีมีขอบเขตในกรณีที่ขอบเขตดีกรีไม่เท่ากัน จากนั้นในงานวิจัยส่วนที่สอง เราจะวิเคราะห์การออกแบบวิธีการปรับคุณภาพสำหรับวิธีการค้นหาแบบช่วงในเครือข่ายเพียร์ทูเพียร์ซึ่งพิจารณาทั้งกรณีที่มีแค่การใส่กุญแจและกรณีทั่วไปที่มีการใส่และลบกุญแจ

### 1. คำตอบที่เป็นต้นไม้สำหรับ MDM problem

ในส่วนนี้เราจะพิสูจน์ว่าภายใต้เงื่อนไขที่ว่าแบนด์วิดท์ของทุกๆ เส้นเชื่อมมีค่าเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการ  $s$  จะมีคำตอบที่ดีที่สุดที่มีการเชื่อมต่อกันเป็นต้นไม้สำหรับ MDM problem ในการพิสูจน์เราจะแสดงขั้นตอนวิธีในการแปลงคำตอบของ MDM problem ไปเป็นคำตอบที่มีการเชื่อมต่อกันเป็นต้นไม้

เราสมมติว่าความล่าช้ามีคุณสมบัติเป็นเมตริก นั่นคือ ความล่าช้ามีคุณสมบัติสมมาตรและสอดคล้องกับอสมการสามเหลี่ยม โดยไม่สูญเสียหลักการทั่วไปเราจะปรับแบนด์วิดท์ของเส้นเชื่อมทุกเส้นโดยหารด้วย  $s$  จากนั้นปัดเศษทิ้ง

บทตั้ง 1

ถ้าแบนด์วิดท์ของทุกเส้นเชื่อมมีค่าเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการ  $s$  และความล่าช้ามีคุณสมบัติเป็นเมตริกแล้วคำตอบของ MDM problem จะสามารถเปลี่ยนให้อยู่ในรูปของต้นไม้ที่มีความยาวของเส้นผ่านกลางไม่เพิ่มขึ้นจากคำตอบต้นฉบับ

## พิสูจน์

เราจะใช้ทฤษฎีบท max-flow min-cut (Ford and Fulkerson (1962)) ในการสร้างต้นไม้ เพื่อให้การไหลจากต้นทางไปปลายทางมีความชัดเจน เราจะสร้างกราฟสองส่วน (Bipartite graph) ที่แสดงแผนการส่งข้อมูลกระแส

เมื่อให้คำตอบของ MDM problem มา เราจะกำหนดให้ bipartite graph  $F = (V_1 \cup V_2, E_F)$  โดยที่  $V_1 = V_2 = V$  และแต่ละคู่ของโหนด  $u$  และ  $v$  ที่มี  $u \in P_v$  and  $s_v(u) > 0$  เราจะกำหนดให้มีเส้นเชื่อม  $(u, v) \in E_F$  โดยที่  $u \in V_1$  และ  $v \in V_2$  นั่นคือเราจะมีเส้นเชื่อม  $(u, v)$  ใน  $F$  ก็ต่อเมื่อ  $u$  มีการส่งข้อมูลกระแสให้  $v$

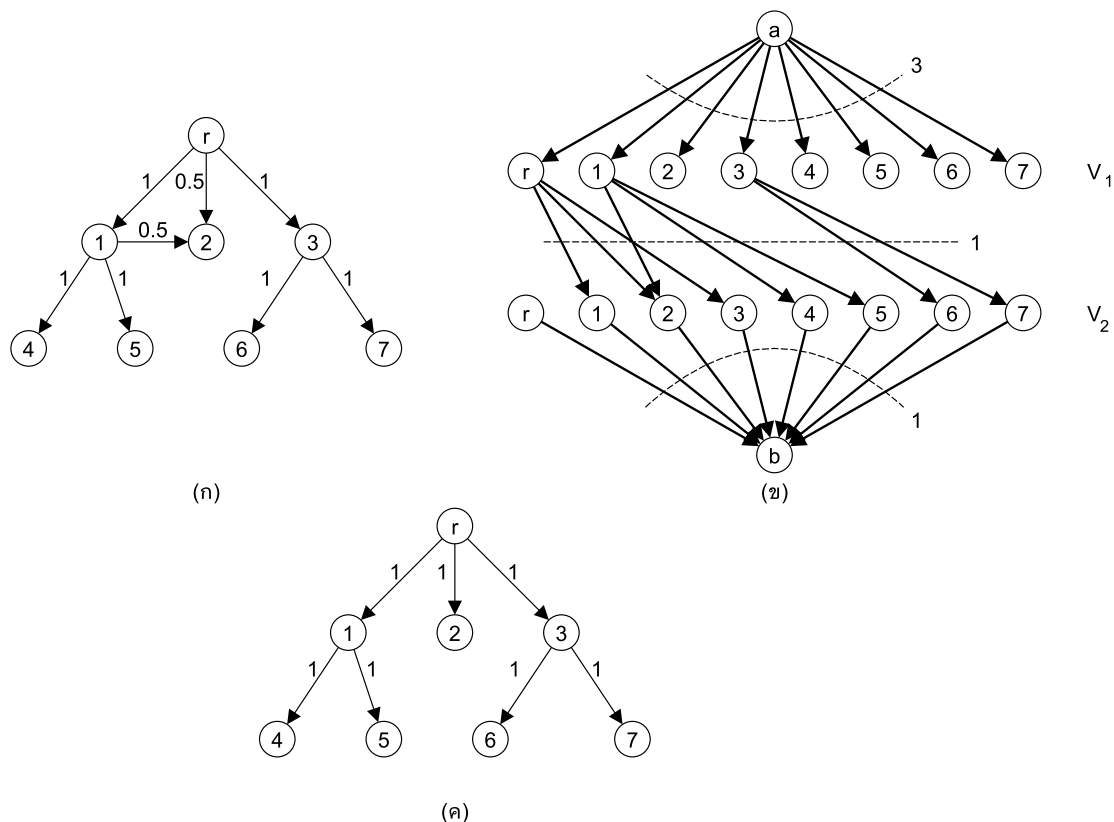
ในการสร้างคำตอบที่เป็นต้นไม้ มันเพียงพอถ้าเราสามารถแสดงได้ว่าเราสามารถหาโหนดพ่อ  $P_u$  ให้กับแต่ละโหนด  $u \neq r$  ที่สอดคล้องกับเงื่อนไขสองข้อได้แก่

1. แต่ละโหนด  $u$  นั้น เส้นเชื่อม  $(P_u, u) \in E_F$
2. แต่ละโหนด  $v$  จะมีโหนดลูกไม่เกิน  $U_v$  โหนด

เนื่องจากการที่สามารถกำหนดโหนดพ่อให้กับแต่ละโหนดนั้นทำให้ได้ต้นไม้แผ่ทั่ว เราจะแสดงขั้นตอนวิธีในการกำหนดโหนดพ่อให้กับโหนดลูก

เราจะลดรูปปัญหานี้ไปเป็นปัญหาการไหลสูงสุด (Max-flow problem) โดยใช้การลดรูปแบบมาตรฐาน นั่นคือการเพิ่มโหนดแหล่งกำเนิดพิเศษ  $a$  และโหนดแหล่งรับพิเศษ  $b$  แต่ละเส้นเชื่อม  $(u, v)$  ใน  $F$  เราจะกำหนดความจุเป็นจำนวนเต็มที่น้อยที่สุดที่ไม่น้อยไปกว่าปริมาณของไหลที่โหนด  $v$  ควบคุมโดยตรงมาจากโหนด  $u$  จากคำตอบที่ได้รับ จากนั้นเราจะกำหนดทิศทางให้กับแต่ละเส้นเชื่อมใน  $F$  จากเซต  $V_1$  ไป  $V_2$  สำหรับแต่ละโหนด  $u \in V_1$  เราจะเพิ่มเส้นเชื่อมที่มีทิศทาง  $(a, u)$  ที่มีความจุ  $U_u$  หน่วย สำหรับแต่ละโหนด  $v \in V_2$  เราจะเพิ่มเส้นเชื่อมที่มีทิศทาง  $(v, b)$  ที่มีความจุหนึ่งหน่วย

เราสามารถยืนยันได้ว่าเมื่อกำหนดคำตอบของ MDM problem มาให้จะมี minimum cut ระหว่าง  $ab$  ที่มีขนาดอย่างน้อย  $|V| - 1$  ดังนั้นจากทฤษฎีบท Max-Flow Min-Cut theorem เราจะได้ว่ามีการไหล



ภาพที่ 14 ตัวอย่างการสร้างคำตอบเป็นต้นไม้ออกจากการลดรูปคำตอบของ MDM problem สมมติว่าแต่ละโหนดมีแบนด์วิดท์ของเส้นเชื่อมโยงขึ้น 3 หน่วย

ขนาด  $|V| - 1$  เนื่องจากความจุของทุกเส้นเชื่อมเป็นจำนวนเต็ม ดังนั้นจะมีการไหลเต็มหน่วย (Integral flow) เมื่อกำหนดการไหลเต็มหน่วยมาให้ แต่ละโหนดใน  $x \in V_2 - \{r\}$  จะได้รับการไหลปริมาณหนึ่งหน่วยจากโหนดใน  $V_1$  ซึ่งจะกลายเป็นโหนดพ่อของโหนด  $x$  ดังนั้นเราสามารถหาการกำหนดความสัมพันธ์พ่อลูกได้

ภาพที่ 14 แสดงตัวอย่างการลดรูป โดยภาพที่ 14 (ก) แสดงตัวอย่างคำตอบของ MDM problem โดยสมมติว่าแต่ละโหนดมีแบนด์วิดท์ของเส้นเชื่อมโยงขึ้น 3 หน่วย ภาพที่ 14 (ข) แสดงการลดรูปไปเป็นปัญหาการไหลสูงสุด โดยมีการเพิ่มโหนดพิเศษ  $a$  และ  $b$  มีการเพิ่มเส้นเชื่อมระหว่าง  $a$  กับโหนดในเซต  $V_1$  ที่มีความจุเท่ากับค่าแบนด์วิดท์เชื่อมโยงขึ้นของแต่ละโหนดในเซต  $V_1$  เพิ่มเส้นเชื่อมระหว่างโหนดในเซต  $V_1$  กับโหนดในเซต  $V_2$  ที่มีความจุเป็นจำนวนเต็มที่มีค่าน้อยที่สุดที่ไม่น้อยไปกว่าปริมาณของไหลที่ถูกส่งจากโหนดในเซต  $V_1$  ไปโหนดในเซต  $V_2$  เพิ่มเส้นเชื่อมระหว่างโหนดในเซต  $V_2$  กับโหนด  $b$  ที่มีความจุเป็นหนึ่งหน่วย ภาพที่ 14 (ค) แสดงคำตอบเป็นต้นไม้อ

จากบทตั้ง 1 เมื่อพิจารณาแต่ละ โหนด  $u$  เราสามารถพิจารณาแบนด์วิดท์ของเส้นเชื่อมโยงขึ้นและแบนด์วิดท์ที่ต้องการเป็นเส้นเชื่อมที่เกิดกับโหนด  $u$  ได้ ปัญหานี้จึงลดรูปไปเป็น BDST problem with non-uniform degree bound ได้ด้วยการกำหนดขอบเขตดีกรี  $B_v = U_v + 1$  ให้กับแต่ละ โหนด  $v \in V - \{r\}$  และกำหนดขอบเขตดีกรี  $B_r = U_r$  สำหรับเครื่องต้นทาง  $r$  ดังนั้นเราจะได้บทแทรก

บทแทรก 1

ถ้าแบนด์วิดท์มีค่าเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการ  $s$  ถ้ามีขั้นตอนวิธีที่มีอัตราส่วนการประมาณเป็น  $O(\sqrt{\log n})$  สำหรับ BDST problem with non-uniform degree bound จะมีขั้นตอนวิธีการประมาณที่เป็น  $O(\sqrt{\log n})$  สำหรับ MDM problem

## 2. การวิเคราะห์ขั้นตอนวิธีสำหรับ BDST problem with non-uniform degree bound

ในส่วนนี้เราจะพิสูจน์ว่าขั้นตอนวิธีสำหรับ BDST problem with non-uniform degree bound ที่นำเสนอในบทอุปกรณและวิธีการ มีอัตราส่วนการประมาณเป็น  $O(\sqrt{\log n})$  เริ่มต้นเราจะทบทวนคุณสมบัติที่เกี่ยวข้องกับต้นไม้ก่อน ขั้นตอนวิธีที่เราแนะนำเสนอนั้น เราแบ่งโหนด  $V$  ออกเป็น 3 เซต  $V_1, V_2$  และ  $V_3$  เราจะวิเคราะห์แยกส่วนกันโดยโหนดในกลุ่ม  $V_3$  จะนำไปสร้างเป็นต้นไม้  $H$  ซึ่งเป็นส่วนฐานของคำตอบ เราจะพิสูจน์คุณสมบัติของต้นไม้  $H$  ในหัวข้อย่อย 2.2 สำหรับโหนดในกลุ่ม  $V_2$  เราจะสร้างเซตของเส้นทาง  $k$  เส้นซึ่งเป็นส่วนหางของคำตอบ เราจะพิสูจน์คุณสมบัติในส่วนหางในหัวข้อย่อย 2.3 สุดท้ายเมื่อนำเอาต้นไม้  $H$  มาต่อด้วยเซตของเส้นทาง  $k$  เส้นและต่อยด้วยเซตของโหนดในกลุ่ม  $V_1$  จะได้ทฤษฎีบท สุดท้ายเราแสดงวิธีการจัดการกับโหนดรากในกรณีที่โหนดรากมีดีกรีต่ำ

### 2.1 การวิเคราะห์คุณสมบัติของต้นไม้ (Properties of trees)

สำหรับต้นไม้  $T = (V, E)$  ใดๆ และโหนด  $u \in V$  เราจะนิยาม  $\deg_T(u)$  ว่าเป็นดีกรี (Degree) ของโหนด  $u$  สำหรับโหนดในต้นไม้  $T$  ที่มีดีกรีเท่ากับ 1 เราจะเรียกว่าโหนดใบ (Leaves) สำหรับโหนดที่ไม่ใช่โหนดใบของต้นไม้ (Non-leaf nodes) เราจะเรียกว่า โหนดภายใน (Internal nodes) เราจะนิยาม  $N(T)$  ว่าเป็นเซตของโหนดภายในของต้นไม้  $T$

ประพจน์ที่จะกล่าวต่อไปเป็นความจริงพื้นฐานของคุณสมบัติของต้นไม้

ประพจน์ 1

สำหรับต้นไม้  $T = (V, E)$  ใดๆ จำนวนของโหนดใบจะเท่ากับ

$$\sum_{v \in N(T)} (deg_T(v) - 2) + 2$$

พิสูจน์

จากคุณสมบัติพื้นฐานของต้นไม้ 2 ข้อได้แก่

$$2|E| = \sum_{v \in T} (deg_T(v)) \quad (1)$$

และ

$$|E| = |V| - 1 \quad (2)$$

เมื่อนำเอาสมการ (2) แทนค่าในสมการ (1) จะได้ว่า

$$2(|V| - 1) = \sum_{v \in T} (deg_T(v))$$

เนื่องจากเซต  $V$  ประกอบด้วย  $N(T)$  และ  $L$  เมื่อ  $L$  แทนเซตของโหนดใบไม้ในต้นไม้  $T$  จะได้ว่า

$$2|V| - 2 = \sum_{v \in N(T)} (deg_T(v)) + \sum_{v \in L} (deg_T(v))$$

โหนดใบของต้นไม้มีดีกรีเท่ากับ 1 ดังนั้น  $\sum_{v \in L} (deg_T(v)) = |L|$  ทำให้ได้ว่า

$$2|V| - 2 = \sum_{v \in N(T)} (deg_T(v)) + |L|$$

$$\begin{aligned}
2|V| - 2 - |L| &= \sum_{v \in N(T)} (deg_T(v)) \\
2|N(T) + L| - 2 - |L| &= \sum_{v \in N(T)} (deg_T(v)) \\
|L| - 2 &= \sum_{v \in N(T)} (deg_T(v)) - 2|N(T)| \\
|L| &= \sum_{v \in N(T)} (deg_T(v) - 2) + 2
\end{aligned}$$

จะได้ว่าจำนวนใบของต้นไม้  $T$  มีค่าเท่ากับ  $\sum_{v \in N(T)} (deg_T(v) - 2) + 2$  ■

เมื่อกำหนด instance ของ BDST problem with non-uniform degree bounds และต้นไม้  $T$  มาให้ สำหรับโหนด  $u$  ใน  $T$  กำหนดให้  $U_T(u)$  แทนจำนวนดีกรีที่ไม่ถูกใช้ (Unused degree) ของโหนด  $u$  ในต้นไม้  $T$  นั่นคือ  $U_T(u) = B_u - deg_T(u)$ , สำหรับ  $V'$  ที่เป็นซับเซตของ  $V$  จำนวนดีกรีที่ไม่ถูกใช้ของเซตของโหนด  $V'$  ของต้นไม้  $T$  จะแทนด้วย  $U_T(V')$  มีค่าเท่ากับผลรวมของดีกรีที่ไม่ถูกใช้ของโหนดในเซตของโหนด  $V'$  ในประพจน์ถัดไปจะเป็นการพิสูจน์เกี่ยวกับดีกรีที่ไม่ถูกใช้แล้วคำตอบที่เป็นไปได้ของ instance ของ BDST problem with non-uniform degree bounds

## ประพจน์ 2

กำหนดให้  $T'$  แทนต้นไม้ของเซตของโหนด  $V'$  ถ้าทุกๆ โหนด  $u$  ในเซตของโหนด  $V'$  ที่มี  $deg_{T'}(u) \leq B_u$  แล้วจะได้ว่าจำนวนดีกรีที่ไม่ถูกใช้มีค่าเป็น  $U_{T'}(V') = \sum_{u \in V'} (B_u - 2) + 2$

## พิสูจน์

พิจารณาต้นไม้  $T'$  ใดๆ สำหรับแต่ละโหนด  $u$  จำนวนดีกรีที่ไม่ถูกใช้ของ  $u$  จะเท่ากับ  $B_u$  ลบด้วยดีกรีของมันในต้นไม้  $T'$  เนื่องจากผลรวมของดีกรีของโหนดในต้นไม้ที่มี  $|V'|$  โหนดมีค่าเป็น  $2 \cdot (|V'| - 1) = 2|V'| - 2$  ดังนั้นผลรวมของดีกรีที่ไม่ถูกใช้ของทุกโหนดรวมกันเป็น  $\sum_{v \in V'} B_v - (2|V'| - 2) = \sum_{v \in V'} (B_v - 2) + 2$  ตามต้องการ ■

ประพจน์ถัดไปจะพิจารณาถึงความเป็นไปได้ของตัวอย่างของ BDST problem with non-uniform degree bounds ว่าตัวอย่างของปัญหาที่ได้รับมามีผลเฉลยหรือไม่

### ประพจน์ 3

ถ้า  $\sum_{u \in V} B_u < 2|V| - 2$  แล้ว instance ของ BDST problem with non-uniform degree bounds นั้นจะไม่มีผลเฉลย

จากประพจน์ 2 ทำให้เราได้เงื่อนไขของจำนวนโหนด  $u$  ที่มีดีกรีเป็น 1

### บทแทรก 2

กำหนดให้  $V_1$  แทนเซตของโหนด  $u$  ที่มี  $B_u = 1$  สำหรับ instance ที่มีคำตอบของ BDST problem with non-uniform degree bounds จะมี  $|V_1| \leq \sum_{w \in V - V_1} (B_w - 2) + 2$

ดังนั้นจากนี้ไปเราจะสมมติว่าเงื่อนไขของบทแทรก 2 เป็นจริง

ประพจน์ต่อไปจะแสดงว่ามีวิธีในการหาเซตของเส้นทางจำนวนเท่ากับ โหนดใบ โดยที่แต่ละเส้นทางไม่ใช่เส้นเชื่อมร่วมกันอีกทั้งยังแผ่ทั่วเซตของโหนดในต้นไม้อีกด้วย

### ประพจน์ 4

สำหรับต้นไม้  $T = (V, E)$  ใดๆ ที่มีโหนดใบเป็นจำนวนเท่ากับ  $k$  จะมีเซตของเส้นทาง  $k$  เส้นทางที่ไม่ใช่เส้นเชื่อมร่วมกัน (edge-disjoint paths) ใน  $T$  ที่แผ่ทั่ว  $T$

### พิสูจน์

เนื่องจาก  $T$  เป็นต้นไม้ที่มีโหนดราก จึงมีความสัมพันธ์ของโหนดในต้นไม้แบบพ่อลูก (Parent-child) ใน  $T$  เราสามารถกำหนดทิศทางของเส้นเชื่อมทุกๆ เส้นได้ โดยกำหนดให้แต่ละเส้นเชื่อมโหนดลูกชี้ไปยังโหนดพ่อ

เราจะแสดงวิธีการหาเซตของเส้นเชื่อม  $E'$  จากต้นไม้  $T$  ในรูปของกราฟใหม่ที่เป็นกราฟแบบมีทิศทาง (Directed graph)  $T' = (V, E')$  โดยมีเงื่อนไขว่า  $T'$  ประกอบด้วยเส้นทางที่ถูกระบุทิศทางจำนวน  $k$  เส้นทาง

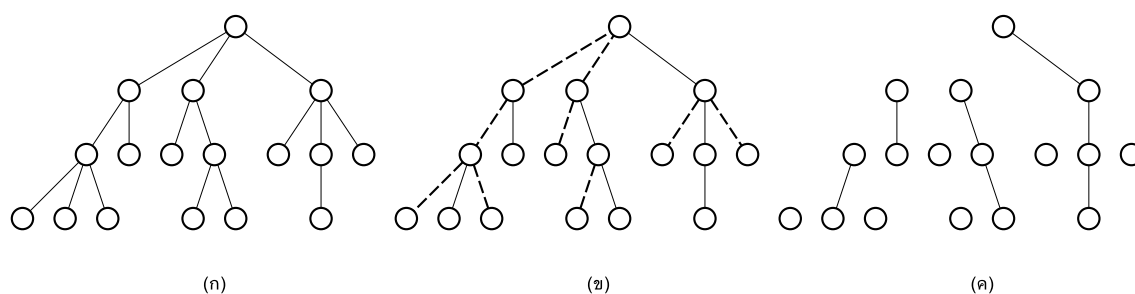
ในการหาเซตของเส้นเชื่อม  $E'$  โหนดภายในแต่ละโหนด  $u$  จะเลือกโหนดลูกของมัน  $v$  มาหนึ่งโหนดด้วยวิธีอะไรก็ได้ จากนั้นเราจะเก็บเส้นเชื่อม  $(v, u)$  ไว้ในเซต  $E'$

ผลลัพธ์ของกราฟ  $T'$  จะประกอบไปด้วยเซตของเส้นทางที่มีทิศทาง ทั้งนี้เพราะว่าดีกรีขาเข้า (Indegree) ของแต่ละโหนดใน  $T'$  มีไม่เกิน 1 และไม่มีวงจร (Cycle) ใน  $T'$  เนื่องจากมันเป็นซับกราฟแบบมีทิศทางในต้นไม้ที่เส้นเชื่อมมีทิศทางชี้ไปทางราก

ในการพิสูจน์ว่าส่วนประกอบ (Component) มีจำนวนเท่ากับ  $k$  เราจะแสดงว่าแต่ละส่วนประกอบจะประกอบด้วยโหนดใบใน  $T$  จำนวนหนึ่งโหนดเท่านั้น พิจารณาเส้นทาง  $P$  ใดๆ ใน  $T'$  สังเกตได้ว่า  $P$  จะประกอบด้วยโหนดใบไม่เกินหนึ่งโหนด เนื่องจากโหนดใบใดๆ จะมีดีกรีขาเข้าเป็นศูนย์ และใน  $P$  จะมีโหนดที่มีดีกรีขาเข้าเป็นศูนย์เพียงโหนดเดียว เนื่องจากโหนดภายในใดๆ ใน  $T$  มีดีกรีขาเข้าเป็นหนึ่งใน  $T'$  มันจึงไม่สามารถเป็นโหนดที่มีดีกรีขาเข้าเป็นศูนย์ใน  $P$  ได้ ดังนั้น  $P$  จะต้องมีโหนดใบของ  $T$  อย่างน้อยหนึ่งโหนด จะได้ว่าแต่ละเส้นทางที่มีทิศทางใน  $T'$  จะประกอบด้วยโหนดลูกเพียงโหนดเดียวเท่านั้น ทำให้พิสูจน์ได้ว่าเราจะมีเส้นทางแบบมีทิศทางเป็นจำนวน  $k$  เส้นทางที่แผ่ทั่วต้นไม้ที่เส้นเชื่อมมีทิศทางชี้ไปทางรากของ  $T$

สำหรับการที่จะได้เส้นทางที่ไม่ใช้เส้นเชื่อมร่วมกันใน  $T$  นั้นทำได้โดยเราจะเก็บแต่ละเส้นทางจาก  $T'$  และเอาทิศทางบนเส้นเชื่อมออก ■

ภาพที่ 15 เป็นตัวอย่างของการแบ่งต้นไม้ออกเป็นเส้น ภาพที่ 15 (ก) เป็นต้นไม้เริ่มต้น ในภาพที่ 15 (ข) แต่ละโหนดเลือกเส้นเชื่อมที่เชื่อมไปยังโหนดลูกหนึ่งเส้น โดยที่เส้นปะจะแทนเส้นเชื่อมที่จะถูกเอาออกและเส้นทึบจะเป็นเส้นที่เลือก ภาพที่ 15 (ค) แบ่งโหนดออกเป็น 10 กลุ่มตามเซตของโหนดที่ติดกัน



ภาพที่ 15 ตัวอย่างการแบ่งต้นไม้ออกเป็นเส้น 10 เส้น

ประพจน์ต่อไปสามารถถูกพิสูจน์ได้ด้วยอสมการอิงรูปสามเหลี่ยม (Triangle inequality) ซึ่งจะเป็นการสรุปคุณสมบัติที่มีประโยชน์ที่สัมพันธ์กับเส้นผ่านกลาง  $\Delta$

ประพจน์ 5

- (1) สำหรับโหนด  $v$  ใดๆ  $\ell(r, v) \leq \Delta$
- (2) สำหรับคู่ของโหนด  $v$  และ  $u$  ใดๆ  $\ell(u, v) \leq 2\Delta$

## 2.2 การวิเคราะห์ส่วนฐาน

เราจะสร้างต้นไม้  $H$  จากเซต  $V_3$  ถ้าทุกโหนดมีขอบเขตของดีกรีเท่ากัน Könnemann *et al.* (2005) ได้แสดงบทตั้งดังต่อไปนี้

บทตั้ง 2

(Könnemann *et al.*, 2005, ทฤษฎีบท 1) เมื่อกำหนดเซตของโหนด  $V$ , เมตริก  $d$  บน  $V$  และขอบเขตของดีกรีที่มากที่สุดเป็น  $B$  ขั้นตอนวิธีของ Könnemann *et al.* จะให้ผลลัพธ์เป็นต้นไม้ที่ดีกรีของโหนดมากที่สุดไม่เกิน  $B$  และเส้นผ่านกลางมีค่าไม่เกิน  $O(\sqrt{\log_B n} \cdot \Delta)$  เมื่อ  $\Delta$  คือเส้นผ่านกลางที่ต่ำที่สุดของต้นไม้ใดๆ ที่มีดีกรีไม่เกิน  $B$

เนื่องจากปัญหาที่วิทยานิพนธ์นี้สนใจเป็นกรณีที่แต่ละโหนดมีขอบเขตดีกรีไม่เท่ากัน เรา  
จะแสดงวิธีการลดรูปปัญหาไปยังปัญหาที่มีดีกรีเท่ากันก่อน เพื่อให้สามารถนำเอาขั้นตอนวิธีของ  
Könemann *et al.* มาใช้แก้ปัญหา

### การลดรูป

ในส่วนนี้เราจะเสนอวิธีการลดรูป เมื่อกำหนด instance  $\mathcal{I}$  ของ BDST problem  
with non-uniform degree bound มาให้ ซึ่งประกอบด้วยเซตของโหนด  $V$ , เมตริก  $d$  บน  $V$  และ ขอบเขต  
ดีกรีของแต่ละโหนด  $B_v$  โดยที่  $B_v \geq 3$  สำหรับแต่ละ  $v \in V$  เราจะทำการสร้าง instance  $\mathcal{I}'$  ของปัญหา  
ที่มีเงื่อนไขว่าทุกโหนดมีขอบเขตดีกรีเป็นค่าคงที่ค่าเดียวกันดังนี้

สำหรับแต่ละโหนด  $v \in V$  เราจะสร้างโหนด  $v^1, v^2, \dots, v^{B_v-2}$  ซึ่งเปรียบ  
เสมือนสำเนาของโหนด  $v$  จำนวน  $B_v - 2$  ชุดและแทนเซตของโหนดเหล่านี้ด้วย  $c(v)$  กำหนดให้  $V' =$   
 $\cup_{v \in V} c(v)$  และนิยามฟังก์ชันการส่ง  $f: V' \rightarrow V$  ที่  $f(u) = v$  ก็ต่อเมื่อ  $u \in c(v)$  สำหรับเมตริก  $d'$  สามารถ  
นิยามดังนี้ สำหรับโหนด  $u, v \in V'$  ใดๆ ให้  $d'(u, v) = d(f(u), f(v))$  สังเกตได้ว่าสำหรับโหนด  $v$  ใดๆ  
ระยะทางระหว่างสองโหนดใดๆ ใน  $c(v)$  ภายใต้  $d'$  จะมีค่าเป็นศูนย์ สุดท้ายเราจะกำหนดค่าขอบเขต  
ดีกรีของทุกโหนด  $v \in V'$  เป็น 3

ภายใต้การแปลงนี้ คำตอบใดๆ ในสำหรับกรณีที่ขอบเขตดีกรีไม่เท่ากัน  
สามารถเปลี่ยนไปเป็นคำตอบในกรณีที่ทุกโหนดมีขอบเขตดีกรีเป็นค่าคงที่ค่าเดียวกันได้ด้วยเส้นผ่าน  
กลางเดียวกัน ดังนั้นคำตอบที่ดีที่สุดของ instance  $\mathcal{I}$  สามารถถูกแปลงให้เป็นคำตอบของตัวอย่าง  $\mathcal{I}'$   
ของปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางน้อยที่สุดโดยทุกโหนดมีขอบเขตดีกรีค่าเดียวกันที่  
มีเส้นผ่านกลางที่เท่ากันได้ ดังนั้นทำให้สรุปได้ว่า คำตอบที่ดีที่สุดในทุกกรณีที่ทุกโหนดมีขอบเขตดีกรี  
เป็นค่าคงที่ค่าเดียวกันจะไม่แย่ไปกว่าคำตอบที่ดีที่สุดในทุกกรณีที่ขอบเขตดีกรีไม่เท่ากัน

กำหนดให้ต้นไม้  $T'$  แทนคำตอบของ  $\mathcal{I}'$  บทตั้งต่อไปจะเป็นการพิสูจน์จากกลับ

ต้นไม้  $T'$  สามารถถูกแปลงเป็นต้นไม้  $T$  ที่เป็นคำตอบของปัญหาในกรณีที่แต่ละโหนดมีขอบเขตคิรีไม่เท่ากันได้ ซึ่งเส้นผ่านกลางจะไม่เกินเส้นผ่านกลางของ  $T'$

### พิสูจน์

สำหรับกราฟ  $H$  ใดๆ และซับเซต  $U$  ของ  $H$  กำหนดให้  $H[U]$  แทนซับกราฟของ  $H$  ที่ถูกเหนี่ยวนำด้วย  $U$  เราจะกำหนดรากของต้นไม้  $T'$  ไว้ที่โหนด  $r \in T'$  โดยที่เป็นโหนดใดก็ได้ต่อไปเราจะแสดงว่า  $T'$  สามารถถูกปรับเปลี่ยนให้มีเงื่อนไขดังนี้

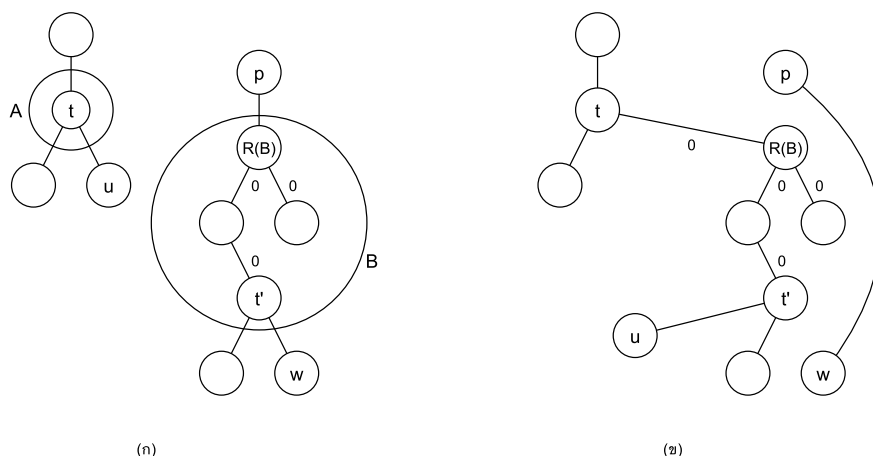
1. สำหรับแต่ละ  $v \in V$  ซับกราฟ  $T'[c(v)]$  จะเชื่อมกัน
2. ระยะทางจากโหนด  $u$  ไป  $r$  ในต้นไม้  $T'$  จะไม่เพิ่มขึ้น

เราจะอธิบายการดำเนินการบนโหนด  $v \in V$  ใดๆ เพื่อที่จะมั่นใจว่า  $T'[c(v)]$  จะสอดคล้องกับเงื่อนไข (1) ในขณะที่ทุกๆ ระยะทางยังคงสอดคล้องกับเงื่อนไข (2)

สำหรับส่วนประกอบที่เชื่อมกัน (Connected component)  $C$  ใดๆ เราจะนิยามโหนดรากของส่วนประกอบ  $R(C)$  ว่าเป็นโหนดที่ใกล้กับโหนดรากของ  $T'$  มากที่สุด และนิยามเซตของโหนดทางของส่วนประกอบ  $C$  ว่าเป็นเซตของโหนดใน  $C$  ที่อยู่ไกลจาก  $r$  มากที่สุดใช้สัญลักษณ์เป็น  $T(C)$  ดังนั้นเมื่อส่วนประกอบมีโหนดเดียวจะได้ว่า  $R(C) \in T(C)$

พิจารณาบางโหนด  $v \in V$  ที่  $T'[c(v)]$  ไม่เชื่อมกัน ให้  $A$  และ  $B$  แทนสองส่วนประกอบที่เชื่อมกันใน  $T'[c(v)]$  สมมติว่า  $R(A)$  อยู่ไม่ไกลจากโหนดรากไปกว่า  $R(B)$  เราจะทำการเชื่อม  $R(B)$  กับโหนดทางโหนดหนึ่งของ  $A$  กำหนดให้  $p$  เป็นโหนดพ่อของ  $R(B)$  ในต้นไม้  $T'$  สังเกตได้ว่าจะมีโหนด  $p$  เพราะว่ากรณีเดียวที่  $R(B)$  จะไม่มีโหนดพ่อคือเมื่อ  $R(B)$  นั้นเป็นโหนดราก แต่ในกรณีนั้น  $R(B)$  จะต้องใกล้โหนดรากมากกว่า  $R(A)$  ซึ่งขัดแย้งกับที่เราสมมติไว้

หยิบโหนด  $t \in T(A)$  ใดๆ มา เราจะเปลี่ยนโหนดพ่อของ  $R(B)$  เป็นโหนด  $t$  เริ่มต้นเราจะลบเส้นเชื่อม  $(p, R(B))$  และเพิ่มเส้นเชื่อม  $(t, R(B))$  สังเกตได้ว่าระยะทางจากโหนดราก  $r$  ไปยัง  $R(B)$  เท่ากับระยะทางจากโหนดราก  $r$  ไปยัง  $R(A)$



ภาพที่ 16 ตัวอย่างของการเชื่อม component  $A$  และ  $B$  (ก) ก่อนเชื่อม (ข) หลังเชื่อม

ถ้าดีกรีของ โหนด  $t$  ไม่เกินเจ็ด (ไม่เกินสาม) การแปลงถือว่าสำเร็จ ถ้าไม่ใช่ เนื่องจาก  $B_t = 3$  แสดงว่าขณะนี้ โหนด  $t$  มีโหนดลูกอยู่สามโหนดรวม  $R(B)$  พิจารณาโหนดลูกของ  $t$  ที่ไม่ใช่  $R(B)$  กำหนดให้  $u$  เป็นหนึ่งในโหนดเหล่านั้นและกำหนดให้  $t'$  เป็นบางโหนดใน  $T(B)$  ถ้า  $t'$  มีโหนดลูก  $w$  บางโหนด ให้ลบเส้นเชื่อม  $(t', w)$  และเพิ่มเส้นเชื่อม  $(p, w)$  เพื่อที่จะให้  $t'$  มีดีกรีไม่เกินสอง

สังเกตว่าก่อนหน้านี้ โหนด  $w$  ก่อนที่จะมีการดำเนินการ โหนด  $w$  จะไปถึง  $p$  จะผ่านทางส่วนประกอบใน  $T'[c(v)]$  เนื่องจากเส้นเชื่อม  $(p, w)$  เชื่อมโหนด  $w$  กับโหนด  $p$  โดยตรง จากอสมการสามเหลี่ยมจะได้ว่าระยะทางจากโหนด  $w$  ไปโหนด  $r$  ไม่เพิ่มขึ้น

เราจะเปลี่ยนโหนดพ่อของ  $u$  ไปเป็นโหนด  $t'$  โดยการลบเส้นเชื่อม  $(t, u)$  และเพิ่มเส้นเชื่อม  $(t', u)$  สังเกตได้ว่าระยะทางจากโหนด  $u$  ไปยังโหนดรากจะยังคงเท่าเดิมเนื่องจาก  $t'$  และ  $t$  ถูกเชื่อมกันด้วยเส้นทางที่มีค่าใช้จ่ายเป็นศูนย์และ  $d'(t, u) = d'(t', u)$  รูปที่ 16 แสดงตัวอย่างการลดรูป

เราสามารถทำการดำเนินการนี้ซ้ำจนกระทั่งทุก  $T'[c(v)]$  เชื่อมกันหมดสำหรับทุกๆ  $v \in V$  การดำเนินการเหล่านี้ทำให้  $T'$  ถูกแปลงเป็นคำตอบของ  $T$  ด้วยเส้นผ่านกลางค่าเดียวกัน ■

จากการลดรูปนี้แสดงว่าคำตอบใดๆ สำหรับกรณี ที่ทุก โหนดมีขอบเขตดีกรี

เป็นค่าคงที่ค่าเดียวกันสามารถเปลี่ยนไปเป็นคำตอบในกรณีที่มีขอบเขตดีกรีไม่เท่ากันได้ด้วยเส้นผ่านกลางเดียวกัน ดังนั้นสรุปได้ว่าคำตอบที่ดีที่สุดในกรณีที่มีขอบเขตดีกรีไม่เท่ากันจะไม่แย่ไปกว่าคำตอบที่ดีที่สุดในกรณีที่ทุกโหนดมีขอบเขตดีกรีเป็นค่าคงที่ค่าเดียวกัน

เมื่อพิจารณาการแปลงไปและแปลงกลับ ทำให้สรุปได้ว่าคำตอบที่ดีที่สุดในกรณีที่มีขอบเขตดีกรีไม่เท่ากันจะเท่ากับคำตอบที่ดีที่สุดในกรณีที่ทุกโหนดมีขอบเขตดีกรีเป็นค่าคงที่ค่าเดียวกัน ดังนั้นขั้นตอนวิธีแบบประมาณใดๆ สำหรับกรณีที่ทุกโหนดมีขอบเขตดีกรีเป็นค่าคงที่ค่าเดียวกันสามารถใช้ได้กับกรณีที่มีขอบเขตดีกรีไม่เท่ากันได้ด้วยอัตราส่วนการประมาณเดียวกัน เมื่อใช้บทตั้ง 2 เราจะได้ขอบเขตของเส้นผ่านกลางของส่วนฐานดังบทตั้ง 4

บทตั้ง 4

เส้นผ่านกลางของส่วนฐานจะไม่เกิน  $O(\sqrt{\log n})\Delta$

### 2.3 การวิเคราะห์ส่วนหาง

ในส่วนนี้เราจะพิจารณาโหนดใน  $V_2$  (โหนดที่มีดีกรีเท่ากับ 2) กำหนดให้  $K$  เป็นจำนวนของดีกรีที่ไม่ถูกใช้ในส่วนฐาน เราจะสร้างเส้นทาง  $K$  เส้นจากเซตของโหนด  $V_2$  จากนั้นจะเชื่อมเส้นทางเหล่านั้นเข้ากับโหนดในส่วนฐานที่มีดีกรีที่ไม่ถูกใช้

ในบทตั้งต่อไปจะแสดงว่า สำหรับต้นไม้ใดๆ ที่แต่ละโหนดมีดีกรีไม่เกินขอบเขตดีกรีของมัน จะมีจำนวนของใบจะไม่เกิน  $K$  นั่นคือ  $\sum_{v \in V_3} (B_v - 2) + 2$

บทตั้ง 5

กำหนดให้  $T = (V, E)$  เป็นต้นไม้ใดๆ ถ้าทุกโหนด  $u \in T$  ที่  $\deg_T(u) \leq B_u$  แล้วจำนวนของใบจะไม่เกิน  $K$

### พิสูจน์

กำหนดให้  $N$  แทนเซตของโหนดภายในในต้นไม้  $T$  จากประพจน์ 1 จะได้ว่าจำนวนของโหนดใบในต้นไม้  $T$  มีค่าเป็น  $\sum_{v \in N} (deg_T(v) - 2) + 2$  จากเงื่อนไขที่แต่ละโหนดมีดีกรีไม่เกินขอบเขตดีกรีของมัน จะได้ว่า

$$\sum_{v \in N} (deg_T(v) - 2) + 2 \leq \sum_{v \in N} (B_v - 2) + 2 = \sum_{v \in V_3} (B_v - 2) + 2.$$

ในส่วนสุดท้ายได้มาจาก  $N \subseteq V_2 \cup V_3$  และสำหรับแต่ละโหนด  $v \in V_2$  จะมี  $B_v - 2 = 0$  ■

พิจารณาคำตอบที่ดีที่สุด  $T^*$  กำหนดให้  $K'$  แทนจำนวนของโหนดใบใน  $T^*$ ,  $K$  แทนจำนวนของความจุของแบนด์วิธของเส้นเชื่อมโยงขึ้นที่ไม่ได้ใช้ของส่วนฐาน จากบทตั้ง 5 จะได้ว่า  $K' \leq K$  เมื่อเราใช้ประพจน์ 4 ในการแบ่งโหนดใน  $T^*$  ออกเป็นเส้นทาง  $K'$  เส้น จะเห็นได้ว่าความยาวของแต่ละเส้นทางจะไม่เกินค่า  $\Delta$  เพราะว่าแต่ละเส้นทางเป็นเส้นทางย่อยของต้นไม้  $T^*$  สังเกตได้ว่าหากเส้นทางที่ได้น้อยกว่า  $K$  เราสามารถเพิ่มจำนวนของเส้นทางให้เป็น  $K$  ได้โดยการตัดเส้นทางเดิม ซึ่งการตัดจะไม่เพิ่มความยาวของเส้นทาง ทำให้เราได้บทตั้งต่อไป

### บทตั้ง 6

จะมีเซตของเส้นทางที่ไม่ทับกัน  $K$  เส้นทางที่แผ่ทั่วเซตของโหนดใน  $T^*$  ที่ความยาวของแต่ละเส้นทางมีค่าไม่เกิน  $\Delta$

จากบทตั้งที่ผ่านมา เมื่อเราเพิ่มเส้นเชื่อมจาก  $r$  ไปยังทั้งสองจุดปลายของแต่ละเส้นทาง เราจะได้เซตของทัวร์ที่แผ่ทุกโหนดจำนวน  $K$  เส้น โดยที่แต่ละเส้นจะมีความยาวไม่เกิน  $3\Delta$  ทั้งนี้เนื่องจากประพจน์ 5 รับประกันว่า  $\ell(r, u) \leq \Delta$  สำหรับแต่ละโหนด  $u$  ดังนั้นเราจะได้อำตอบที่เป็นไปได้ของปัญหา The  $k$ -Traveling Salesperson Problem ที่มีค่าใช้จ่ายไม่เกิน  $3\Delta$

เราจะสร้างเซตของเส้นทาง  $K$  เส้นบนจากโหนดใน  $V_2$  ซึ่งแต่ละเส้นจะมีความยาวไม่เกิน  $3\alpha\Delta$  เมื่อ  $\alpha$  เป็นค่าการประมาณที่ได้รับการรับประกันของขั้นตอนวิธี FHK จากการใช้ขั้นตอนวิธี FHK เราจะแบ่งเซตของโหนด  $V$  ออกเป็น  $K$  ทัวร์ ที่มีโหนดราก  $r$  ในแต่ละทัวร์ เราจะสร้างเส้นทางโดยการท่องในทัวร์เริ่มที่โหนดราก จากนั้นข้ามโหนดในเซต  $V_3 - \{r\}$  และโหนดในเซต  $V_1$  ในขั้นสุดท้าย เราจะนำเอาโหนด  $r$  ออกจากเส้นทาง จากอสมการสามเหลี่ยม การที่เราข้ามโหนดในเซต  $V_3 - \{r\}$  และโหนดในเซต  $V_1$  จะไม่ทำให้ความยาวของเส้นทางเพิ่มขึ้น ดังนั้นแต่ละเส้นทางจะมีค่าใช้จ่ายแต่ละเส้นไม่เกิน  $3\alpha\Delta$  สังเกตได้ว่าแต่ละเส้นทางจะมีแต่โหนดในเซตของ  $V_2$  บทตั้งบทต่อไปจะแสดงว่าความยาวของแต่ละเส้นทางนั้นไม่ยาวจนเกินไป บทพิสูจน์ของมันได้มาจากค่าการประมาณที่รับประกันของขั้นตอนวิธี FHK และประพจน์ 6

#### บทตั้ง 7

ความยาวของแต่ละเส้นทางในส่วนหางจะ ไม่เกิน  $3\alpha\Delta$  เมื่อ  $\alpha$  แทนค่าการประมาณที่รับประกันของขั้นตอนวิธี FHK

#### พิสูจน์

เราจะสร้างคำตอบที่เป็นไปได้ของ The  $k$ -Traveling Salesperson Problem ที่ค่าใช้จ่ายไม่เกิน  $3\Delta$  จากประพจน์ 6 เราจะได้ว่ามีเซตของเส้นทาง  $K$  เส้นที่แผ่ทั่วโหนดทุกใน  $T^*$  สำหรับแต่ละเส้นทาง  $P$  เราจะสร้างทัวร์ที่มีจุดเริ่มต้นที่โหนด  $r$  โดยการเชื่อมจุดปลายทั้งสองข้างของเส้นทาง  $P$  เข้ากับโหนด  $r$  ดังนั้นเราจะได้ทัวร์  $K$  ทัวร์ที่มีจุดเริ่มต้นที่โหนด  $r$  ที่แผ่ทั่วโหนดในกราฟ สังเกตได้ว่าความยาวของแต่ละทัวร์ จะไม่เกิน  $3\Delta$  เพราะว่าความยาวของแต่ละเส้นทางไม่เกิน  $\Delta$  และความยาวของเส้นเชื่อม ที่เพิ่มเข้ามาในทัวร์แต่ละเส้นยาวไม่เกิน  $\Delta$

เนื่องจากขั้นตอนวิธี FHK จะให้เซตของทัวร์  $K$  ทัวร์ โดยที่ความยาวของทัวร์ที่ยาวที่สุดจะ ไม่เกิน  $\alpha$  เท่าของคำตอบที่ดีที่สุด ทำให้เราได้ว่าแต่ละทัวร์ที่เราสร้างขึ้นจะมีความยาวไม่เกิน  $\alpha(3\Delta)$  สังเกตได้ว่า การใช้ทางลัด (shortcutting) และการลบโหนด  $r$  นั้นไม่ทำให้ความยาวเพิ่มขึ้น ดังนั้นได้ประพจน์ตามที่ต้องการ ■

เมื่อนำเอาค่าประมาณที่รับประกันของขั้นตอนวิธี FHK และค่าประมาณของขั้นตอนวิธีแบบประมาณของปัญหาการเดินทางของพนักงานขายหนึ่งคนจากงานวิจัยของ Christofides (1976) เราจะได้บทแทรกบทต่อไป

### บทแทรก 3

เส้นทางแต่ละเส้นมีความยาวไม่เกิน  $(15/2 - 3/K)\Delta$

#### 2.4 การวิเคราะห์ทฤษฎีบทหลัก

เมื่อเรามีส่วนฐานและส่วนหางแล้ว เราจะนำเอาโหนดในเซต  $V_1$  มาเชื่อมต่อท้ายของแต่ละหางของส่วนหาง จากประพจน์ 5 ทำให้ได้ว่าการเชื่อมต่อนี้จะเพิ่มความยาวเส้นผ่านกลางไม่เกิน  $\Delta$  เมื่อนำเอาส่วนฐานและส่วนหางมารวมกัน จากประพจน์ 5 ทำให้ได้ว่าการเชื่อมต่อนี้จะเพิ่มความยาวเส้นผ่านกลางไม่เกิน  $\Delta$  เช่นกัน ดังนั้นเมื่อรวมความยาวของส่วนฐานโดยใช้บทตั้ง 4 ความยาวของส่วนหางโดยใช้บทแทรก 3 และประพจน์ 5 ประกอบกันจะได้ทฤษฎีบทหลัก

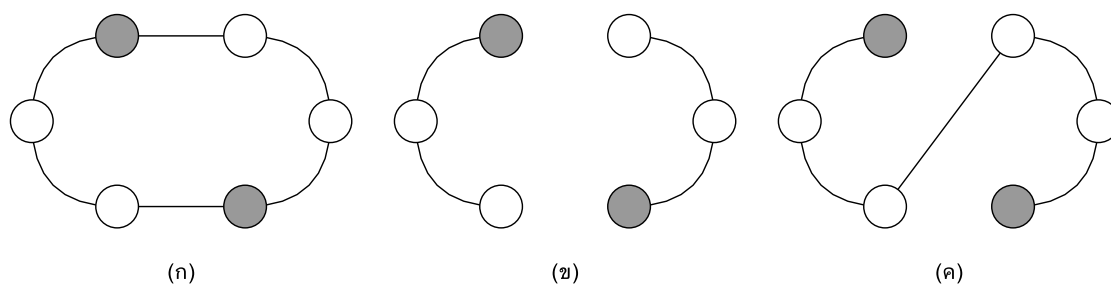
#### ทฤษฎีบท 1

มีขั้นตอนวิธีแบบประมาณที่ได้อัตราส่วน  $O(\sqrt{\log n})$  ที่ทำงานภายในเวลาเป็นฟังก์ชันพหุนามของ BDST with non-uniform degree bounds

#### 2.5 การจัดการกรณีที่โหนดรากมีดีกรีที่ต่ำ

ในโลกความเป็นจริงนั้น โหนดราก  $r$  อาจจะมีแบนด์วิดท์ของเส้นเชื่อมโยงขึ้นที่ต่ำ นั่นคือกรณีที่  $B_r \leq 2$  เราจะพิจารณาเป็น 2 กรณี

กรณีแรก ทุกโหนดมีดีกรีไม่เกินสอง



ภาพที่ 17 ตัวอย่างกรณีที่ทุกโหนดมีดีกรีไม่เกินสอง

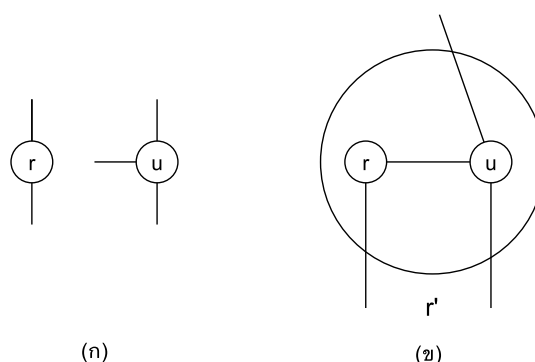
เราจะพิจารณาในกรณีที่ทุกโหนด  $v$  มีขอบเขตดีกรี  $B_v \leq 2$  ในกรณีนี้คำตอบที่ดีที่สุดจะมีลักษณะเป็นเส้นเท่านั้น ให้  $\Delta$  แทนเส้นผ่านกลางของมัน ถ้าเราเชื่อมสองโหนดในเส้นทางที่มีดีกรีเป็นหนึ่งเข้าด้วยกัน เราจะได้ทัวร์ที่มีความยาวไม่เกิน  $2\Delta$  ดังนั้นเราสามารถใช้ขั้นตอนวิธีการประมาณสำหรับปัญหาการเดินทางของพนักงานขายที่รับประกันว่าแย่กว่าทัวร์ที่ดีที่สุดไม่เกิน  $\beta$  เท่า ( $\beta$ -approximation algorithm) มาใช้ในการหาทัวร์  $T$  ผลลัพธ์ที่ได้จะมีความยาวไม่เกิน  $2\beta$  เท่าของเส้นผ่านกลางที่ดีที่สุด  $\Delta$  ตัวอย่างเช่นถ้าเราใช้ขั้นตอนวิธีของ Christofides (1976) ซึ่งมีค่า  $\beta = 1.5$  ความยาวของ  $T$  ก็จะไม่เกิน  $3\Delta$

เราจะทำการหาต้นไม้  $T$  ที่สอดคล้องกับข้อจำกัดของดีกรี สังเกตได้ว่าเงื่อนไขของดีกรีจะผิดเงื่อนไขที่โหนดที่มีขอบเขตดีกรีเป็นหนึ่ง แต่จะมีโหนดที่มีขอบเขตดีกรีเป็นหนึ่งได้ไม่เกินสองโหนด ในการหาต้นไม้ แต่ละโหนด  $u$  ที่  $B_u = 1$  เราจะลบเส้นเชื่อมหนึ่งเส้นออกจาก  $T$  ที่ติดกับมันและจะเพิ่มเส้นเชื่อมไม่เกินหนึ่งเส้นเพื่อเชื่อมเส้นทางให้ต่อกัน ดังนั้นเส้นผ่านกลางของต้นไม้คำตอบจะไม่เกิน  $6\Delta$

จากรูป 17 (ก) เป็นตัวอย่างที่สมมติให้โหนดที่มีสีเทาเป็นโหนดที่มีดีกรีจำกัดเป็นหนึ่งในทัวร์ รูป 17 (ข) แต่ละโหนดที่มีดีกรีเป็นหนึ่งลบเส้นเชื่อมเส้นหนึ่งที่ติดกับมัน รูป 17 (ค) เพิ่มเส้นเชื่อมเพื่อทำให้ส่วนประกอบสองส่วนเชื่อมต่อกัน

กรณีที่สอง มีบางโหนดที่มีดีกรีเกินสอง

ในกรณีนี้เราจะเลือกโหนด  $u$  ที่  $B_u > 2$  มาเชื่อมกับโหนด  $r$  หลังจากนั้นเราจะพิจารณาโหนด  $r$  และ  $u$  รวมเป็นโหนดเดียวกันจะเรียกว่าโหนด  $r'$  และกำหนดให้  $B_{r'} = B_r + B_u - 2$  (ถ้า



ภาพที่ 18 ตัวอย่างกรณีที่มีบางโหนดมีดีกรีเกินสอง

$B_{r'} < 3$  ให้เชื่อมกับโหนดใหม่  $v$  ที่มีดีกรีมากกว่าสอง) จากนั้นใช้โหนด  $r'$  เป็นโหนดรากของขั้นตอนวิธีของเรา คำตอบที่ได้จากขั้นตอนวิธีจะเป็นต้นไม้ที่มีความยาวของเส้นผ่านกลางเป็น  $O(\sqrt{\log_{B'} n})$  เท่าของคำตอบที่ดีที่สุดเมื่อ  $B' = \min\{B_v | B_v \geq 3\}$

จากรูป 18 (ก) เป็นตัวอย่างที่สมมติให้โหนด  $r$  มีดีกรีเป็นสอง และมีโหนด  $u$  ที่ดีกรีเป็นสามมาเชื่อมกับโหนด  $r$  รูป 18 (ข) แสดงการเชื่อมโหนด  $r$  เข้ากับโหนด  $u$  แล้วพิจารณาเป็นโหนดใหม่  $r'$

### 3. การวิเคราะห์การออกแบบวิธีการปรับคุณลักษณะสำหรับวิธีการค้นหาแบบช่วง

เราจะแบ่งการวิเคราะห์ออกเป็นสองส่วนใหญ่ๆคือ การวิเคราะห์อัตราส่วนความไม่สมดุลของกรณีที่มีแต่การใส่กุญแจอย่างเดียวและการวิเคราะห์อัตราส่วนความไม่สมดุลในกรณีทั่วไป เริ่มต้นเราจะแสดงการวิเคราะห์อัตราส่วนความไม่สมดุลในกรณีที่มีแต่การใส่กุญแจอย่างเดียว ทั้งนี้กรณีนี้ง่ายต่อการวิเคราะห์และให้แนวคิดเพื่อนำเอาไปใช้ในการจัดการกรณีทั่วไป กรณีที่มีแต่การใส่กุญแจอย่างเดียวนี้มีความน่าสนใจเพราะว่าในหลายๆ แอปพลิเคชันเช่นการแชร์ไฟล์ การลบนั้นเกิดขึ้นน้อยมาก จากนั้นเราจะแสดงค่าใช้จ่ายของขั้นตอนวิธีในกรณีที่มีแต่การใส่กุญแจ แล้วเราจะแสดงการวิเคราะห์อัตราส่วนความไม่สมดุลของกรณีทั่วไปพร้อมทั้งแสดงค่าใช้จ่ายของขั้นตอนวิธีในกรณีทั่วไป

เนื่องจากขั้นตอนวิธีที่นำเสนอต้องการข้อมูลแบบครอบคลุม นั่นคือข้อมูลภาระของโหนดที่มีภาระสูงที่สุดและภาระของโหนดที่ต่ำสุด เมื่อนำไปใช้ในเครือข่ายจริงต้องมีการจัดการเกี่ยวกับการให้

บริการข้อมูลเหล่านี้ ดังนั้นเราจะแสดงวิธีการจัดการกับข้อมูลเหล่านี้ในเครือข่ายจริงในหัวข้อย่อย 3.5

### 3.1 การวิเคราะห์อัตราส่วนความไม่สมดุลของกรณีที่มีแค่การใส่กุญแจอย่างเดียว

เพื่อความง่ายในการวิเคราะห์ เราจะสมมติว่าการดำเนินการแต่ละอย่างนั้นเมื่อถูกเรียกใช้งานจะทำงานเสร็จทันที

สำหรับเวลา  $t$  ใดๆ กำหนดให้  $L_t(u)$  แทนภาระของโหนด  $u$  หลังจากเวลา  $t$  ทันทีและให้  $L'_t(u)$  แทนภาระของโหนด  $u$  ก่อนเวลา  $t$  ทันที นั่นคือ  $t - \epsilon$  สำหรับ  $\epsilon > 0$  ให้  $Min_t$  และ  $Min'_t$  แทนภาระต่ำสุดในระบบหลังจากเวลา  $t$  ทันทีและก่อนเวลา  $t$  ทันทีตามลำดับนั้นคือ  $Min_t = \min_{u \in V} L_t(u)$  และ  $Min'_t = \min_{u \in V} L'_t(u)$

ในการวิเคราะห์อัตราส่วนความไม่สมดุลนั้น เราจะพิสูจน์คุณสมบัติที่ไม่เปลี่ยนแปลง (Invariant) ว่าจะคงอยู่ตลอดเวลา คุณสมบัติที่ไม่เปลี่ยนแปลงนั้นคือ

"สำหรับเวลา  $t$  ใดๆ ภาระของโหนดใดๆจะไม่เกิน  $(\alpha + 2)$  เท่าของภาระที่น้อยที่สุด"

สังเกตได้จากคุณสมบัตินี้ ทำให้ได้ว่าอัตราส่วนความไม่สมดุลจะมีค่าไม่เกิน  $\alpha + 2$  เนื่องจาก  $\frac{Max_t}{Min_t} \leq (\alpha + 2)$  เมื่อ  $Max_t$  เป็นภาระที่มากที่สุด ณ เวลา  $t$  เราจะพิสูจน์คุณสมบัติที่ไม่เปลี่ยนแปลงนี้ในทฤษฎีบท 2

ภาพรวมของการวิเคราะห์ของกรณีที่มีแค่การใส่กุญแจอย่างเดียว

แม้ว่าการวิเคราะห์จะค่อนข้างซับซ้อน แต่แนวคิดในการวิเคราะห์ไม่ได้ยากแก่การเข้าใจ ในส่วนนี้เราจะเกริ่นคร่าวๆ ถึงการวิเคราะห์

แนวคิดหลักของการวิเคราะห์คือ การพิสูจน์ว่าอัตราส่วนความไม่สมดุลยังคงเป็นค่าคงที่หลังจากการดำเนินการใดๆ เราจะสมมติว่าตอนเริ่มต้นระบบ แต่ละโหนดจะเก็บภาระไว้ด้วยค่าคงที่

$c_0$  ซึ่งเป็นค่าคงที่ที่มีค่ามากกว่า 0 และมีค่าน้อย เราจะเริ่มต้นด้วยการแสดงคุณสมบัติพื้นฐานสำหรับกรณีที่มีแค่การใส่กุญแจที่ว่าการะต่ำสุดจะไม่ลดลง จากนั้นเราจะวิเคราะห์ว่าหลังจากการใส่กุญแจเปลี่ยนแปลงการะของโหนดต่างๆ อย่างไร

เมื่อมีการใส่กุญแจเกิดขึ้นที่โหนด  $u$  จะมีเหตุการณ์ที่เกิดขึ้นได้สองอย่างขึ้นกับว่าโหนด  $u$  เรียกการดำเนินการ MINBALANCE หรือไม่ หลังจากการใส่กุญแจขอบเขตของการะของโหนด  $u$  สามารถวิเคราะห์ได้ไม่ยาก

อย่างไรก็ตามการใส่กุญแจมีผลกับโหนดอื่นอีกสองโหนดได้แก่ โหนด  $v$  ซึ่งเป็นโหนดที่มีการะต่ำสุดและโหนด  $z$  ซึ่งเป็นโหนดเพื่อนบ้านของโหนดที่มีการะต่ำสุด เมื่อการดำเนินการ MINBALANCE ถูกเรียกโหนด  $v$  จะโอนการะทั้งหมดให้กับโหนด  $z$  จากนั้นโหนด  $v$  จะไปแบ่งการะครึ่งหนึ่งของโหนด  $u$  ในขณะที่การวิเคราะห์ขอบเขตของการะของโหนด  $v$  จะตรงไปตรงมาแต่ขอบเขตของโหนด  $z$  นั้นต้องการการวิเคราะห์ที่ลงลึก เนื่องจากโหนด  $z$  อาจจะได้รับการะจากโหนดที่มีการะต่ำสุดหลายครั้งตลอดการทำงานของขั้นตอนวิธี

ส่วนที่ยากก็คือการวิเคราะห์ขอบเขตของการะของโหนด  $z$  เมื่อได้รับการะจากโหนดเพื่อนบ้านโดยที่ไม่มีการใส่กุญแจเกิดขึ้นที่โหนด  $z$  เลย อย่างไรก็ตามในกรณีนั้น เรารู้ว่าโหนดเพื่อนบ้านของโหนด  $z$  ณ บางเวลาจะกลายเป็นโหนดที่มีการะต่ำที่สุด ดังนั้นเราจึงวิเคราะห์ขอบเขตของการะของโหนด  $z$  ที่สัมพันธ์กับการะที่ต่ำที่สุดได้และสามารถพิสูจน์อัตราส่วนความไม่สมดุลได้

การวิเคราะห์อัตราส่วนความไม่สมดุล

เริ่มต้นเราจะแสดงคุณสมบัติพื้นฐานที่สำคัญของการะที่ต่ำที่สุดในระบบก่อน นั่นคือหลังจากการดำเนินการใดๆ ค่าของการะที่ต่ำที่สุดจะไม่ลดลง

บทตั้ง 8

สมมติว่า  $\alpha > 2$  แล้วการะที่ต่ำที่สุดในระบบจะไม่ลดลง

## พิสูจน์

การดำเนินการที่เกิดขึ้นในระบบมี 2 อย่าง ได้แก่ การใส่กุญแจและการดำเนินการ MIN-BALANCE การใส่กุญแจจะทำให้ภาระเพิ่มเพียงอย่างเดียว สิ่งที่ทำให้ภาระลดลงได้คือ การดำเนินการ MINBALANCE ดังนั้นเราจะพิจารณาการดำเนินการนี้ สำหรับแต่ละครั้งที่การดำเนินการ MINBALANCE ถูกเรียกนั้นจะมีโหนดที่เกี่ยวข้องอยู่ทั้งหมด 3 โหนดด้วยกัน ได้แก่ โหนด  $u$  ที่เป็นโหนดที่เริ่มการดำเนินการ MINBALANCE ณ เวลา  $t$  โหนด  $v$  ที่มีภาระต่ำที่สุดในขณะนั้นและโหนด  $z$  ที่เป็นโหนดเพื่อนบ้านของโหนด  $v$  ที่มีภาระน้อยที่สุด ณ เวลา  $t$  เริ่มต้นโหนด  $v$  จะโอนภาระทั้งหมดให้กับโหนด  $z$  ดังนั้นภาระของโหนด  $z$  จะเพิ่มขึ้น ส่วนภาระใหม่ของโหนด  $u$  และโหนด  $v$  คือครึ่งหนึ่งของภาระ ณ เวลา  $t$  ของโหนด  $u$  เนื่องจากโหนด  $u$  เรียกการดำเนินการ MINBALANCE แสดงว่าโหนด  $u$  จะต้องมีความมากกว่า  $\alpha Min_t > 2Min_t$  ดังนั้นหลังจากการดำเนินการเหล่านี้ภาระของโหนด  $u$  และโหนด  $v$  จะมีค่าอย่างน้อย  $Min_t$  ทำให้สรุปได้ว่าภาระที่ต่ำที่สุดในระบบจะไม่ลดลง ■

ต่อไปจะเป็นการวิเคราะห์ภาระของโหนด  $u$  ใดๆ หลังจากการใส่กุญแจเกิดขึ้นที่โหนด  $u$  ในบทต้งบทต่อไปจะเป็นการรับประกันว่าอัตราส่วนความไม่สมดุลของโหนด  $u$  จะมีค่าไม่มาก

## บทตั้ง 9

สมมติว่า  $\alpha \geq 3$  แล้วหลังจากการใส่กุญแจเกิดขึ้นที่โหนด  $u$  ณ เวลา  $t$  และตามด้วยขั้นตอนการปรับดุลภาระจะได้ว่า  $L_t(u) \leq \alpha Min_t$

## พิสูจน์

หลังจากมีการใส่กุญแจเกิดขึ้น เราจะพิจารณาได้ 2 กรณีขึ้นอยู่กับโหนด  $u$  คือ

กรณีแรก โหนด  $u$  เรียกการดำเนินการ MINBALANCE สังเกตได้ว่าภาระของโหนด  $u$  หลังจากการใส่กุญแจจะไม่เกิน  $(\alpha + 2)Min'_t + 1$  ซึ่งมาจากคุณสมบัติที่ไม่เปลี่ยนแปลงและการใส่กุญแจ 1 ดอก จากนั้นภาระของโหนด  $u$  จะถูกลดลงครึ่งหนึ่งจะได้ว่า

$$L_t(u) \leq \left\lceil \frac{(\alpha + 2)Min'_t + 1}{2} \right\rceil \leq \frac{(\alpha + 2)Min'_t + Min'_t}{2} \leq \frac{(\alpha + 3)Min'_t}{2}$$

สำหรับค่า  $\alpha \geq 3$  ใดๆ จะได้ความสัมพันธ์ว่า  $\frac{(\alpha+3)}{2} \leq \alpha$  ดังนั้น  $L_t(u) \leq \alpha Min'_t$  จากคุณสมบัติของภาระต่ำสุดที่ไม่ลดลงจะได้ว่า  $L_t(u) \leq \alpha Min_t$

กรณีที่สอง โหนด  $u$  ไม่เรียกการดำเนินการ MINBALANCE จากขั้นตอนวิธี แสดงว่าภาระของโหนด  $u$  ในกรณีนี้จะไม่เกิน  $\alpha Min'_t$  เนื่องจากภาระต่ำสุดไม่ลดลงทำให้เราได้ว่า  $L_t(u) \leq \alpha Min_t$  ■

สังเกตว่าการดำเนินการ MINBALANCE นั้นสนใจภาระของโหนดที่เพิ่งถูกใส่กุญแจเข้าไป และภาระของโหนดน้อยที่สุด แต่ไม่ได้สนใจภาระของโหนด  $z$  ที่เป็นโหนดเพื่อนบ้านของโหนดที่น้อยที่สุดซึ่งเป็นโหนดที่ได้รับการกระจาย เราจะนิยามเหตุการณ์ *min-transfer* โดยบอกว่าเหตุการณ์ *min-transfer* เกิดขึ้นที่โหนด  $z$  เมื่อโหนดที่มีภาระน้อยที่สุดโอนภาระของมันมาให้โหนด  $z$  ที่เป็นโหนดเพื่อนบ้านที่มีโหนดน้อยที่สุด ทั้งนี้การวิเคราะห์ส่วนใหญ่จะเกี่ยวข้องกับภาระของโหนดที่ได้รับการกระจายจากการเหตุการณ์โอนภาระชนิดนี้

พิจารณาลำดับของเหตุการณ์ *min-transfer* ที่เกิดขึ้นกับโหนด  $z$  กำหนดให้  $t_i$  แทนเวลาหลังจากที่เหตุการณ์ *min-transfer* ครั้งที่  $i$  เกิดขึ้นกับโหนด  $z$  สังเกตว่าก่อนเหตุการณ์ *min-transfer* จะเกิดขึ้นกับโหนด  $z$  อาจจะมีการใส่กุญแจเกิดขึ้นกับโหนด  $z$  ก่อนได้ ดังนั้นบทตั้งบทต่อไปจะเป็นการพิสูจน์ว่าภาระของโหนด  $z$  จะอยู่ในอัตราส่วนต่อภาระต่ำที่สุดที่ดีในกรณีนี้

บทตั้ง 10

ถ้ามีการใส่กุญแจเกิดขึ้นที่โหนด  $z$  ณ เวลา  $t$  ก่อนที่จะมีเหตุการณ์ *min-transfer* ครั้งที่  $i$  กับโหนด  $z$  แล้ว  $L_{t_i}(z) \leq (\alpha + 1)Min_{t_i}$

### พิสูจน์

จากบทตั้ง 9 หลังจากที่มีการใส่กุญแจที่โหนด  $z$  ที่บางเวลา  $t$  จะได้ว่า  $L_t(z) \leq \alpha \text{Min}_t$  หลังจากนั้นภาระของโหนด  $z$  จะเพิ่มขึ้นอีกครั้งเมื่อมีเหตุการณ์ min-transfer ครั้งที่  $i$  เกิดขึ้นที่เวลา  $t_i$  ดังนั้นภาระของโหนด  $z$  จะมีค่าเป็น

$$L_{t_i}(z) = L_t(z) + \text{Min}'_{t_i} \leq \alpha \text{Min}_t + \text{Min}'_{t_i}$$

จากคุณสมบัติของภาระที่ต่ำที่สุดจะไม่ลดลง เราจะได้ว่า  $L_{t_i}(z) \leq (\alpha + 1) \text{Min}_{t_i}$  ■

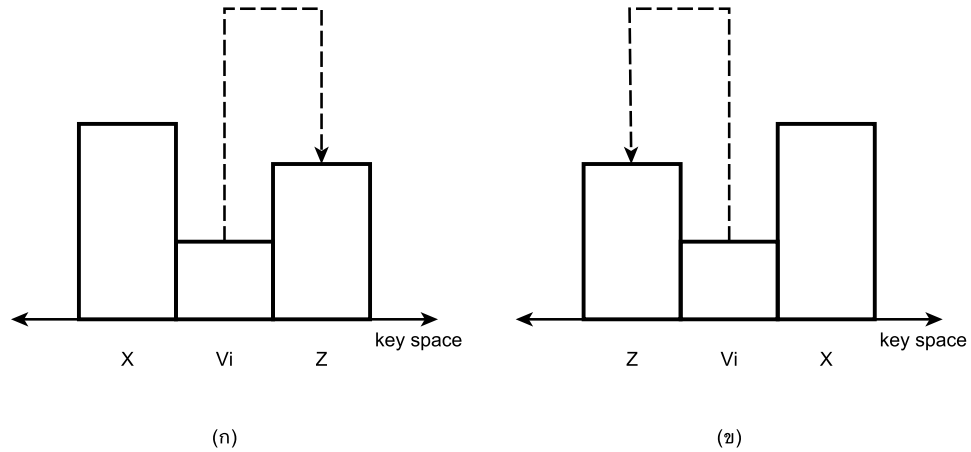
จากผลของบทตั้ง 10 ต่อไปเมื่อพิจารณาภาระของโหนด  $z$  เราจะสนใจแต่กรณีที่ไม่มีการใส่กุญแจที่โหนด  $z$  ก่อนเหตุการณ์ min-transfer จะเกิดขึ้นกับโหนด  $z$

ในการวิเคราะห์ของเรานั้นเราจะแบ่งเหตุการณ์ min-transfer ออกเป็น 2 ชนิดคือเหตุการณ์ left-transfer และเหตุการณ์ right-transfer โดยเหตุการณ์ left-transfer ที่เกิดกับโหนด  $z$  คือเหตุการณ์ min-transfer ที่โหนด  $z$  รับภาระมาจากโหนดที่อยู่ทางซ้ายของโหนด  $z$  (ตามรูป 19 (ก)) ส่วน right-transfer ที่เกิดกับโหนด  $z$  คือเหตุการณ์ min-transfer ที่โหนด  $z$  รับภาระมาจากโหนดที่อยู่ทางขวาของโหนด  $z$  (ตามรูป 19 (ข))

เมื่อเหตุการณ์ min-transfer ครั้งใหม่เกิดกับโหนด  $z$  จะมีสองสถานการณ์คือ เหตุการณ์ min-transfer ครั้งใหม่เป็นชนิดเดียวกับเหตุการณ์ min-transfer ครั้งก่อนหน้าและเหตุการณ์ min-transfer ครั้งใหม่เป็นคนละชนิดกันกับเหตุการณ์ min-transfer ครั้งก่อนหน้า ในบทตั้งต่อไปจะแสดงขอบเขตของภาระของโหนด  $z$  เมื่อมีเหตุการณ์ min-transfer ของชนิดเดียวกันเกิดขึ้นที่โหนด  $z$  เป็นจำนวน  $l$  ครั้งระหว่างเหตุการณ์ min-transfer ของอีกชนิด

### บทตั้ง 11

สมมติว่า  $\alpha \geq 1 + \sqrt{5} \approx 3.237$  และ  $l \geq 0$  พิจารณาเหตุการณ์ min-transfer ครั้งที่  $i, (i+1)$ ,



ภาพที่ 19 ตัวอย่างเหตุการณ์ min-transfer ที่เกิดกับ โหนด  $z$  ทางด้านซ้ายและด้านขวา (ก) เหตุการณ์ left-transfer ที่เกิดกับ โหนด  $z$  และ (ข) เหตุการณ์ right-transfer ที่เกิดกับ โหนด  $z$

...,  $(i + l + 1)$  ที่เกิดกับ โหนด  $z$  ถ้าเหตุการณ์ min-transfer ครั้งที่  $i$  และครั้งที่  $(i + l + 1)$  เป็นชนิดเดียวกัน ในขณะที่เหตุการณ์ min-transfer ครั้งที่  $(i + 1)$ ,  $(i + 2)$ , ...,  $(i + l)$  เป็นชนิดที่ต่างกับครั้งที่  $i$  และ  $(i + l + 1)$  แล้ว หลังจากเหตุการณ์ min-transfer ครั้งที่  $(i + l + 1)$  แล้วจะได้ว่า  $L_{t_{i+l+1}}(z) \leq (\alpha + l + 1)Min_{t_{i+l+1}}$

### พิสูจน์

โดยไม่เสียหลักการทั่วไปเราจะสมมติว่าเหตุการณ์ min-transfer ครั้งที่  $i$  และ  $(i + l + 1)$  ที่เกิดขึ้นที่ โหนด  $z$  เป็นเหตุการณ์ right-transfer ส่วนเหตุการณ์ min-transfer ครั้งที่  $(i + 1)$ ,  $(i + 2)$ , ...,  $(i + l)$  เป็นเหตุการณ์ left-transfer กำหนดให้  $v_i$  แทนโหนดที่มีภาระต่ำสุด ณ เวลา  $t_i$

เราจะเริ่มต้นด้วยกรณีที่มีการใส่กุญแจเกิดขึ้นที่ โหนด  $z$  ระหว่างเหตุการณ์ min-transfer ครั้งที่  $i$  และครั้งที่  $(i + l + 1)$  สมมติว่าการใส่กุญแจที่เกิดขึ้นที่ โหนด  $z$  นั้นเกิดขึ้นก่อนเหตุการณ์ min-transfer ครั้งที่  $k$  เมื่อ  $i < k \leq i + l + 1$  จากบทตั้ง 10 หลังจากเหตุการณ์ min-transfer ครั้งที่  $k$  จะได้ว่า

$$L_{t_k}(z) \leq (\alpha + 1)Min_{t_k}$$

เนื่องจากไม่มีการใส่กุญแจที่ โหนด  $z$  ดังนั้นหลังจากเหตุการณ์ min-transfer ครั้งที่  $k$  เรา

จะได้ว่า

$$L_{t_{i+l+1}}(z) \leq (\alpha + 1)Min_{t_k} + Min_{t_{k+1}} + \cdots + Min_{t_{i+l+1}}$$

จากคุณสมบัติของภาระที่ต่ำที่สุดจะไม่ลดลง ทำให้ได้

$$L_{t_{i+l+1}}(z) \leq (\alpha + 1)Min_{t_{i+l+1}} + (i + l - k)Min_{t_{i+l+1}}$$

เนื่องจาก  $k > i$  สุดท้ายเราจะได้ว่า  $L_{t_{i+l+1}}(z) \leq (\alpha + l + 1)Min_{t_{i+l+1}}$

เหลือกรณีที่ต้องพิจารณาคือ กรณีที่ไม่มีการใส่กุญแจที่โหนด  $z$  ระหว่างเหตุการณ์ min-transfer ครั้งที่  $i$  และครั้งที่  $(i + l + 1)$  ก่อนอื่นเราจะแสดงบทอ้าง

บทอ้าง 1

$$L_{t_{i+l+1}}(z) \leq L_{t_i}(z) + (l + 1)Min'_{t_{i+l+1}}$$

พิสูจน์

สังเกตได้ว่าสำหรับค่า  $j$  ใดๆ ที่  $i < j \leq i + l + 1$  ภาระของโหนด  $z$  ก่อนที่จะมีเหตุการณ์ min-transfer ครั้งที่  $j$  จะมีค่าเท่ากับภาระของมันหลังจากเหตุการณ์ min-transfer ครั้งที่  $(j - 1)$  นั่นคือ

$$L'_{t_j}(z) = L_{t_{j-1}}(z) \quad (3)$$

อีกทั้งสำหรับค่า  $j$  ใดๆ ที่  $i < j \leq i + l + 1$  ภาระของโหนด  $z$  หลังจากเหตุการณ์ min-transfer ครั้งที่  $j$  จะเพิ่มขึ้นจากภาระของโหนดที่ต่ำที่สุดก่อนเวลา  $t_j$  ซึ่งคือ  $Min'_{t_j}$  ดังนั้น

$$L_{t_j}(z) = L'_{t_j}(z) + \text{Min}'_{t_j} \quad (4)$$

เมื่อแทนค่า  $L'_{t_j}(z)$  ด้วยสมการ (3) จะได้ว่า

$$L_{t_j}(z) = L_{t_{j-1}}(z) + \text{Min}'_{t_j} \quad (5)$$

หลังจากนั้นทำการ Telescope เราจะได้

$$L_{t_{i+l+1}}(z) \leq L_{t_i}(z) + \text{Min}'_{t_{i+1}} + \text{Min}'_{t_{i+2}} + \cdots + \text{Min}'_{t_{i+l+1}} \leq L_{t_i}(z) + (l+1)\text{Min}'_{t_{i+l+1}}.$$

ขั้นสุดท้ายมาจากคุณสมบัติของภาระที่ต่ำที่สุดจะไม่ลดลง ■

ในการหาขอบเขตของภาระของโหนด  $z$  ณ เวลา  $t_{i+l+1}$  นั้นต้องการขอบเขตของภาระของโหนด  $z$  ณ เวลา  $t_i$  ดังนั้นเราจะพิจารณาเหตุการณ์ min-transfer ครั้งที่  $i$  ที่เกิดขึ้นกับโหนด  $z$  จากที่ได้สมมติไว้ก่อนหน้านี้เหตุการณ์ min-transfer ครั้งที่  $i$  เป็นเหตุการณ์ right-transfer กำหนดให้โหนด  $x$  เป็นโหนดที่อยู่ทางขวามือของโหนด  $v_i$  ณ เวลาก่อนที่จะมีเหตุการณ์ min-transfer ครั้งที่  $i$  เกิดขึ้นที่โหนด  $z$  (ดังรูป 19 (ข)) เนื่องจากเหตุการณ์ min-transfer ครั้งที่  $(i+l+1)$  เป็นเหตุการณ์ right-transfer โหนด  $x$  จะต้องกลายเป็นโหนดที่มีภาระต่ำที่สุด ณ บางเวลาหลังจาก  $t_i$  กำหนดให้  $t^*$  เป็นเวลาที่โหนด  $x$  กลายเป็นโหนดที่มีภาระต่ำสุดหลังจากเวลา  $t_i$  สังเกตได้ว่าโหนด  $x$  จะเป็นโหนดที่มีบทบาทมากในการวิเคราะห์ของเรา

เราจะพิจารณาเป็นสองกรณีขึ้นกับว่าโหนด  $x$  เรียกการดำเนินการ MINBALANCE หรือไม่

กรณีที่ 1 โหนด  $x$  ไม่ได้เรียกการดำเนินการ MINBALANCE ระหว่างช่วงเวลา  $t_i$  ถึง  $t_{i+l+1}$  ในกรณีนี้เหตุการณ์ min-transfer ครั้งที่  $i$  ที่เกิดขึ้นที่โหนด  $z$  โหนด  $v_i$  จะโอนภาระทั้งหมดของมันให้กับโหนด  $z$  ณ เวลา  $t_i$  แต่ไม่ได้โอนให้โหนด  $x$  นั้นหมายความว่าภาระของโหนด  $z$  ก่อนเหตุการณ์

min-transfer ครั้งที่  $i$  จะไม่เกินภาระของโหนด  $x$  นั่นคือ  $L'_{t_i}(z) \leq L'_{t_i}(x)$  จากนั้นหลังจากเวลา  $t_i$

$$L_{t_i}(z) = L'_{t_i}(z) + \text{Min}'_{t_i} \leq L'_{t_i}(x) + \text{Min}'_{t_i}, \quad (6)$$

เมื่อนำมารวมกับบทอย่าง 1 จะได้ว่า

$$L_{t_i+l+1}(z) \leq L'_{t_i}(x) + \text{Min}'_{t_i} + (l+1)\text{Min}'_{t_i+l+1}. \quad (7)$$

หลังจากเวลา  $t_i$  โหนด  $x$  จะกลายเป็นโหนดที่มีภาระต่ำสุดที่เวลา  $t^*$  สังเกตว่าโหนด  $x$  ไม่ได้เรียกการดำเนินการ MINBALANCE ดังนั้นเหตุการณ์ที่สามารถเกิดขึ้นกับโหนด  $x$  หลังจากเวลา  $t_i$  คือการใส่กุญแจหรือเหตุการณ์ min-transfer ดังนั้นภาระของโหนด  $x$  จะไม่ลดลง ทำให้ได้ว่า  $L'_{t_i}(x) \leq \text{Min}_{t^*}$  เมื่อแทนค่าลงในอสมการ (7)

$$L_{t_i+l+1}(z) \leq \text{Min}_{t^*} + \text{Min}'_{t_i} + (l+1)\text{Min}'_{t_i+l+1}.$$

จากคุณสมบัติของภาระที่ต่ำที่สุดจะไม่ลดลง เราจะได้ว่า  $L_{t_i+l+1}(z) \leq (l+3)\text{Min}_{t_i+l+1}$  และเมื่อ  $\alpha \geq 2$  แล้วจะได้ว่า  $L_{t_i+l+1}(z) \leq (\alpha + l + 1)\text{Min}_{t_i+l+1}$

กรณีที่ 2 โหนด  $x$  เรียกการดำเนินการ MINBALANCE ระหว่างช่วงเวลา  $t_i$  และ  $t_{i+l+1}$  ในกรณีนี้ กำหนดให้  $t'$  เป็นเวลาล่าสุดที่โหนด  $x$  เรียกการดำเนินการ MINBALANCE เนื่องจากการดำเนินการ MINBALANCE ถูกเรียกแสดงว่าต้องมีการใส่กุญแจที่โหนด  $x$  แล้วภาระของโหนด  $x$  มีค่าเกิน  $\alpha$  เท่าของภาระที่ต่ำที่สุด ณ เวลา  $t'$  หลังจากนั้นภาระของโหนด  $x$  จะถูกแบ่งออกเป็นสองส่วน ดังนั้นภาระของโหนด  $x$  จะมีค่า

$$L_{t'}(x) \geq \left\lceil \frac{\alpha \text{Min}_{t'}}{2} \right\rceil.$$

สังเกตว่าหลังจากเวลา  $t'$  โหนด  $x$  ไม่ได้เรียกการดำเนินการ MINBALANCE และมันกลายเป็นโหนดที่มีภาระต่ำสุด ณ เวลา  $t^*$  เหตุการณ์ที่อาจจะเกิดขึ้นได้กับโหนด  $x$  หลังจากเวลา  $t'$  คือการใส่กุญแจหรือเหตุการณ์ min-transfer ดังนั้นภาระของโหนด  $x$  หลังจากเวลา  $t'$  จะไม่ลดลง ทำให้

ได้ว่า  $L_{t'}(x) \leq L_{t^*}(x)$  ยิ่งไปกว่านั้นเรายังรู้ว่า  $L_{t^*}(x) \leq \text{Min}'_{t_{i+l+1}}$  จากคุณสมบัติของภาระที่ต่ำที่สุดจะไม่ลดลง ดังนั้น

$$\text{Min}'_{t_{i+l+1}} \geq \left\lceil \frac{\alpha \text{Min}_{t'}}{2} \right\rceil. \quad (8)$$

จากคุณสมบัติที่ไม่เปลี่ยนแปลง ทำให้รู้ว่าภาระของโหนด  $z$  ณ เวลา  $t_i$  จะไม่เกิน  $(\alpha + 2)$  เท่าของภาระต่ำสุด ณ เวลา  $t_i$  และเรารู้ว่าภาระต่ำสุดจะไม่ลดลง ทำให้ได้ว่า

$$L_{t_i}(z) \leq (\alpha + 2)\text{Min}_{t_i} \leq (\alpha + 2)\text{Min}_{t'}. \quad (9)$$

พิจารณาบทอ้าง 1 เราจะหารอสมการนั้นด้วยค่า  $\text{Min}_{i+l+1}$  ทั้งสองข้าง จะได้

$$\frac{L_{t_{i+l+1}}(z)}{\text{Min}_{i+l+1}} \leq \frac{L_{t_i}(z) + (l+1)\text{Min}'_{t_{i+l+1}}}{\text{Min}_{t_{i+l+1}}}$$

จากคุณสมบัติของภาระที่ต่ำที่สุดจะไม่ลดลง  $\text{Min}'_{t_{i+l+1}} \leq \text{Min}_{t_{i+l+1}}$  เราจะได้ว่า

$$\begin{aligned} \frac{L_{t_{i+l+1}}(z)}{\text{Min}_{i+l+1}} &\leq \frac{L_{t_i}(z) + (l+1)\text{Min}'_{t_{i+l+1}}}{\text{Min}_{t_{i+l+1}}} \\ &= (l+1) + \frac{L_{t_i}(z)}{\text{Min}'_{t_{i+l+1}}}, \end{aligned}$$

เมื่อนำเอาอสมการ (8) และอสมการ (9) มารวมเข้าไปได้

$$\begin{aligned} \frac{L_{t_{i+l+1}}(z)}{\text{Min}_{i+l+1}} &\leq (l+1) + \frac{(\alpha+2)\text{Min}_{t'}}{\frac{\alpha \text{Min}_{t'}}{2}} \\ &= (l+1) + \frac{2(\alpha+2)}{\alpha}. \end{aligned}$$

จากข้อสมมติที่ว่า  $\alpha \geq 1 + \sqrt{5}$  ทำให้ได้  $\frac{2(\alpha+2)}{\alpha} \leq \alpha$  ทำให้สรุปได้ว่า

$$\frac{L_{t_{i+l+1}}(z)}{Min_{i+l+1}} \leq (\alpha + l + 1)$$

■

เมื่อใช้บทตั้ง 11 เราสามารถสรุปขอบเขตของภาระของโหนด  $z$  หลังจากเหตุการณ์ min-transfer ที่เกิดขึ้นกับมันได้ดังนี้

บทตั้ง 12

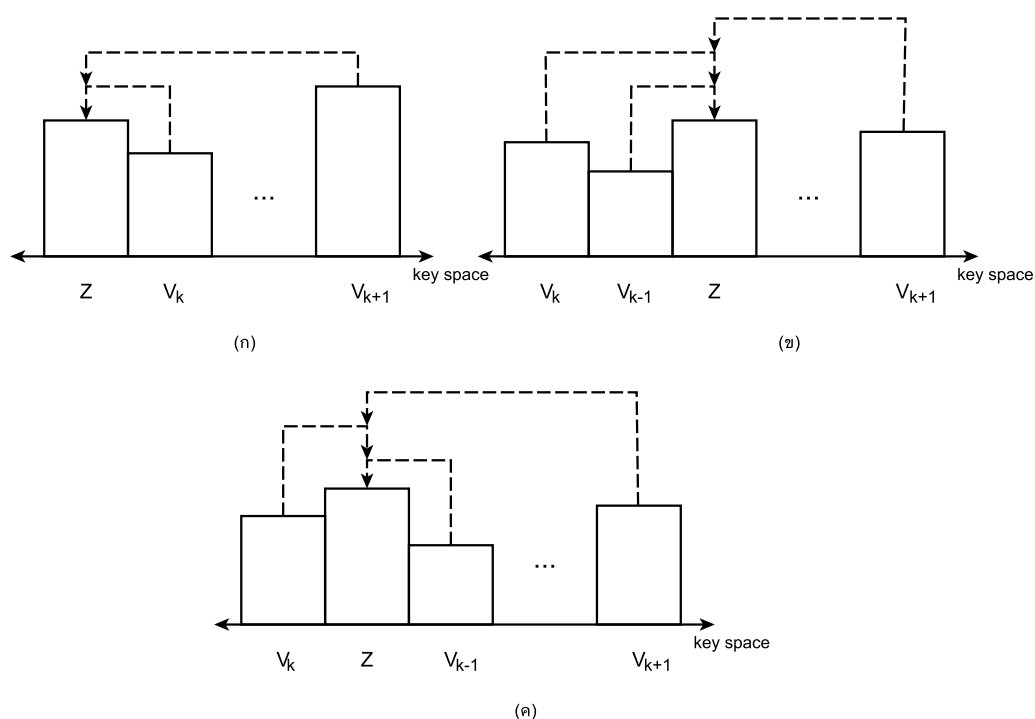
สมมติว่า  $\alpha \geq 1 + \sqrt{5} \approx 3.237$  หลังจากเหตุการณ์ min-transfer ครั้งที่  $i$  เกิดขึ้นกับโหนด  $z$  จะได้ว่า  $L_{t_i}(z) \leq (\alpha + 2)Min_{t_i}$

พิสูจน์

เราจะพิสูจน์โดยวิธีอุปนัยเชิงคณิตศาสตร์บนจำนวนครั้งของเหตุการณ์ min-transfer ที่เกิดขึ้นกับโหนด  $z$

ขั้นตอนฐานหลัก พิจารณาเหตุการณ์ min-transfer ที่เกิดขึ้นกับโหนด  $z$  ครั้งแรก ถ้ามีการใส่กุญแจเกิดขึ้นที่โหนด  $z$  ก่อนที่จะมีเหตุการณ์ min-transfer จากบทตั้ง 10 จะได้ว่า  $L_{t_1}(z) \leq (\alpha + 1)Min_{t_1}$  สำหรับกรณีที่ไม่มีการใส่กุญแจเกิดขึ้นที่โหนด  $z$  ก่อนที่จะมีเหตุการณ์ min-transfer ภาระของโหนด  $z$  ก่อนที่จะมีเหตุการณ์ min-transfer จะมีค่าเท่ากับภาระตอนเริ่มต้นของระบบซึ่งคือ  $c_0$  สังเกตว่าภาระที่น้อยที่สุด ณ เวลา  $t'_1$  ก็ยังคงเป็น  $c_0$  หลังจากเหตุการณ์ min-transfer ครั้งแรกจะได้ว่า  $L_{t_1}(z) = 2Min_{t_1}$  ดังนั้นภาระของโหนด  $z$  ทั้งสองกรณีจะไม่เกิน  $(\alpha + 2)Min_{t_1}$

ขั้นตอนอุปนัย สมมติว่าภาระของโหนดใดๆ ณ เวลา  $t_k$  มีค่าไม่เกิน  $(\alpha + 2)Min_{t_k}$  พิจารณาเหตุการณ์ min-transfer ครั้งที่  $(k + 1)$  ที่เกิดกับโหนด  $z$  พบว่าจะมี 2 กรณี



**ภาพที่ 20** ตัวอย่างเหตุการณ์ min-transfer ครั้งที่  $(k + 1)$  ในรูปแบบต่างๆ (ก) เหตุการณ์ min-transfer ครั้งที่  $k$  และครั้งที่  $(k + 1)$  เป็นชนิดเดียวกัน (ข) เหตุการณ์ min-transfer ครั้งที่  $(k - 1)$  และ  $k$  เป็นชนิดเดียวกันแต่เหตุการณ์ min-transfer ครั้งที่  $(k + 1)$  เป็นคนละชนิดกัน (ค) เหตุการณ์ min-transfer ครั้งที่  $(k - 1)$  และ  $(k + 1)$  เป็นชนิดเดียวกันแต่เหตุการณ์ min-transfer ครั้งที่  $k$  เป็นคนละชนิดกัน

กรณีที่ 1 เหตุการณ์ min-transfer ครั้งที่  $(k+1)$  และครั้งที่  $k$  เป็นชนิดเดียวกัน (ดูภาพที่ 20 (ก)) จากบทตั้ง 11 เมื่อกำหนดค่า  $l = 0$  หลังจากเหตุการณ์ min-transfer ครั้งที่  $(k+1)$  จะได้ว่า  $L_{t_{k+1}}(z) \leq (\alpha+1)Min_{t_{k+1}}$

กรณีที่ 2 เหตุการณ์ min-transfer ครั้งที่  $(k+1)$  เป็นคนละชนิดกับเหตุการณ์ min-transfer ครั้งที่  $k$  เราจะพิจารณาเหตุการณ์ min-transfer ครั้งที่  $(k-1)$  ซึ่งมีสามกรณีย่อยด้วยกัน

กรณีย่อยที่ 2.1 ไม่มีเหตุการณ์ min-transfer ครั้งที่  $(k-1)$  ดังนั้นเหตุการณ์ min-transfer ครั้งที่  $k$  เป็นเหตุการณ์ min-transfer ครั้งแรก เราสามารถหาขอบเขตของภาระของโหนด  $z$  หลังจากเวลา  $t_k$  ได้เช่นเดียวกับขั้นตอนฐานหลัก นั่นคือ  $L_{t_k}(z) \leq (\alpha+1)Min_{t_k}$  ภาระของโหนด  $z$  จะเพิ่มขึ้นอีกครั้งเมื่อเกิดเหตุการณ์ min-transfer ครั้งที่  $(k+1)$  เกิดขึ้นกับมัน ณ เวลา  $t_{k+1}$  นั่นคือ

$$L_{t_{k+1}}(z) = L_{t_k}(z) + Min'_{t_{k+1}} \leq (\alpha+1)Min_{t_k} + Min'_{t_{k+1}}.$$

จากคุณสมบัติของภาระที่ต่ำที่สุดจะไม่ลดลง เราจะได้ว่า  $L_{t_{k+1}}(z) \leq (\alpha+2)Min_{t_{k+1}}$  ดังนั้นบทตั้งเป็นจริงในกรณีย่อยนี้

กรณีย่อยที่ 2.2 เหตุการณ์ min-transfer ครั้งที่  $(k-1)$  เป็นชนิดเดียวกันกับเหตุการณ์ min-transfer ครั้งที่  $k$  (ดูภาพที่ 20 (ข)) จากบทตั้ง 11 เมื่อกำหนดค่า  $l = 0$  หลังจากเหตุการณ์ min-transfer ครั้งที่  $k$  แล้ว  $L_{t_k}(z) \leq (\alpha+1)Min_{t_k}$  หลังจากนั้นภาระของโหนด  $z$  จะเพิ่มขึ้นอีกครั้งจากเหตุการณ์ min-transfer ครั้งที่  $(k+1)$  นั่นคือ

$$L_{t_{k+1}}(z) = L_{t_k}(z) + Min'_{t_{k+1}} \leq (\alpha+1)Min_{t_k} + Min'_{t_{k+1}}.$$

จากคุณสมบัติที่ไม่ลดลงของภาระต่ำสุด เราจะได้ว่า  $L_{t_{k+1}}(z) \leq (\alpha+2)Min_{t_{k+1}}$  ดังนั้นบทตั้งเป็นจริงในกรณีย่อยนี้

กรณีย่อยที่ 2.3 เหตุการณ์ min-transfer ครั้งที่  $(k-1)$  เป็นคนละชนิดกับเหตุการณ์

min-transfer ครั้งที่  $k$  นั้นคือเหตุการณ์ min-transfer ครั้งที่  $(k - 1)$  เป็นชนิดเดียวกับเหตุการณ์ min-transfer ครั้งที่  $(k + 1)$  (ดูภาพที่ 20 (ค)) จากบทตั้ง 11 เมื่อกำหนดค่า  $l = 1$  หลังจากเหตุการณ์ min-transfer ครั้งที่  $(k + 1)$  แล้ว  $L_{t_{k+1}}(z) \leq (\alpha + 2)Min_{t_{k+1}}$  ซึ่งทำให้บทตั้งเป็นจริงในกรณีย่อยนี้ ■

สังเกตได้ว่าภาระของโหนดใดๆ สามารถเปลี่ยนแปลงได้หลังจากที่มีการใส่กุญแจหรือมีเหตุการณ์ min-transfer จากบทตั้ง 9 และบทตั้ง 12 ทำให้เราสามารถพิสูจน์คุณสมบัติที่ไม่เปลี่ยนแปลงได้ดังทฤษฎีบทต่อไป

## ทฤษฎีบท 2

พิจารณากรณีที่มีแต่การใส่กุญแจ สมมติว่า  $\alpha \geq 1 + \sqrt{5} \approx 3.237$  สำหรับโหนด  $u \in V$  ใดๆ หลังจากเหตุการณ์ใดๆ ณ เวลา  $t$  จะได้ว่า  $L_t(u) \leq (\alpha + 2)Min_t$

จากทฤษฎีบท 2 เราสามารถแสดงว่าอัตราส่วนความไม่สมดุลของขั้นตอนวิธีของเราในกรณีที่มีแต่การใส่กุญแจอย่างเดียวว่าเป็นค่าคงที่

## บทแทรก 4

พิจารณากรณีที่มีแต่การใส่กุญแจ สมมติว่า  $\alpha \geq 1 + \sqrt{5} \approx 3.237$  อัตราส่วนความไม่สมดุลของขั้นตอนวิธีจะมีค่าเป็นค่าคงที่

### 3.2 การวิเคราะห์ค่าใช้จ่ายของขั้นตอนวิธีในกรณีที่มีแต่การใส่กุญแจ

ขั้นตอนวิธีของเราในกรณีที่มีแต่การใส่กุญแจ ใช้การดำเนินการสองอย่างได้แก่การใส่กุญแจและการดำเนินการ MINBALANCE เราจะพิจารณาค่าใช้จ่ายในการการย้ายกุญแจหนึ่งดอกจากโหนดหนึ่งไปยังอีกโหนดหนึ่งจะคิดค่าใช้จ่ายหนึ่งหน่วย เราจะวิเคราะห์ค่าใช้จ่ายโดยใช้วิธีเดียวกับงานวิจัยของ Ganesan *et al.* (2004) ซึ่งใช้วิธีศักย์ (Potential method) และใช้ฟังก์ชันศักย์ (Potential function) ฟังก์ชันเดียวกันกับที่ Ganesan *et al.* ใช้ในการวิเคราะห์

### ทฤษฎีบท 3

ถ้า  $\alpha > 2(1 + \sqrt{3}) \approx 5.464$  แล้วค่าใช้จ่ายเฉลี่ยของขั้นตอนวิธีของเราในกรณีที่มีแต่การใส่กุญแจเป็นค่าคงที่

#### พิสูจน์

กำหนดให้  $\bar{L}_t$  แทนภาระเฉลี่ยของระบบ ณ เวลา  $t$  และกำหนดให้  $N_i$  เป็นโหนดที่  $i$  ในระบบที่รับผิดชอบช่วง  $[R_{i-1}, R_i)$  ฟังก์ชันศักย์ที่ใช้คือ  $\Phi = \frac{c(\sum_{i=1}^n (L_t(N_i))^2)}{\bar{L}_t}$  เมื่อ  $c$  เป็นค่าคงที่ที่ระบบค่าภายหลัง เราจะแสดงว่าพลังงานศักย์เพิ่มขึ้นเมื่อมีการใส่กุญแจจะไม่เกินค่าคงที่และพลังงานศักย์ลดลงเมื่อการดำเนินการ MINBALANCE เกิดขึ้น โดยพลังงานศักย์ที่ลดลงจะเพียงพอเป็นค่าใช้จ่ายของการดำเนินการ MINBALANCE

การใส่กุญแจ พิจารณาการใส่กุญแจที่โหนด  $N_j$  ณ เวลา  $t$  ก่อนที่ขั้นตอนการสมดุลภาระจะถูกเรียก สังเกตว่าภาระของทุกโหนดยกเว้นโหนด  $N_j$  จะไม่เปลี่ยนแปลงระหว่างการใส่กุญแจ ดังนั้นพลังงานศักย์ที่เพิ่มขึ้น  $\Delta\Phi$  มีค่าเป็น

$$\begin{aligned} \Delta\Phi &= \frac{c(\sum_{i=1}^n (L_t(N_i))^2) - c(L_t(N_j))^2 + c(L_t(N_j) + 1)^2}{\bar{L}_t + \frac{1}{n}} - \frac{c(\sum_{i=1}^n (L_t(N_i))^2)}{\bar{L}_t} \\ &\leq \frac{c((L_t(N_j) + 1)^2) - c(L_t(N_j))^2}{\bar{L}_t} \\ &= \frac{c(2L_t(N_j) + 1)}{\bar{L}_t}. \end{aligned}$$

จากคุณสมบัติที่ไม่เปลี่ยนแปลงจะได้ว่า  $L_t(N_j) \leq (\alpha + 2)Min_t$  จากความจริงที่ว่า  $\bar{L}_t \geq Min_t \geq 1$  แล้ว  $L_t(N_j) \leq (\alpha + 2)\bar{L}_t$  เมื่อแทนค่ากลับเข้าไปจะได้ว่า

$$\Delta\Phi \leq \frac{c(2(\alpha + 2)\bar{L}_t + 1)}{\bar{L}_t} \leq c(2\alpha + 5)$$

เนื่องจากการใส่กุญแจมีการย้ายกุญแจใหม่หนึ่งคอกให้กับบางโหนด ค่าใช้จ่ายจริงของการใส่กุญแจคือหนึ่งหน่วย ดังนั้นค่าใช้จ่ายถัวเฉลี่ยของการใส่กุญแจจะถูกจำกัดด้วยค่าคงที่

การดำเนินการ MinBalance มี 3 โหนดที่เกี่ยวข้องกับการดำเนินการนี้คือ โหนด  $N_j$  ที่เป็นโหนดที่เรียกการดำเนินการ MINBALANCE, โหนดที่มีภาระต่ำที่สุด  $N_k$  และโหนด  $N_l$  ที่เป็นโหนดเพื่อนบ้านของโหนด  $N_k$  ที่มีภาระต่ำที่สุด เมื่อโหนด  $N_j$  เรียกขั้นตอน MINBALANCE โหนด  $N_k$  จะโอนภาระทั้งหมดของมันให้กับโหนด  $N_l$  หลังจากนั้นโหนด  $N_k$  จะแบ่งภาระครึ่งหนึ่งของโหนด  $N_j$  พลังงานสัปดาห์จะลดลงเป็น

$$\begin{aligned}\Delta\Phi &= \frac{c \left( L_t(N_j)^2 + L_t(N_k)^2 + L_t(N_l)^2 - 2\left(\frac{L_t(N_j)}{2}\right)^2 - (L_t(N_k) + L_t(N_l))^2 \right)}{\bar{L}_t} \\ &= \frac{c \left( \frac{L_t(N_j)^2}{2} - 2L_t(N_k)L_t(N_l) \right)}{\bar{L}_t}.\end{aligned}$$

จากคุณสมบัติที่ไม่เปลี่ยนแปลงจะได้ว่า  $L_t(N_l) \leq (\alpha + 2)L_t(N_k)$  และจากเงื่อนไขที่โหนด  $N_j$  เรียกขั้นตอน MINBALANCE ถ้า  $L_t(N_k) \leq \frac{L_t(N_j)}{\alpha}$  ทำให้ได้ว่า  $L_t(N_l) \leq (\alpha + 2)\frac{L_t(N_j)}{\alpha}$  เมื่อแทนค่ากลับไปจะได้

$$\begin{aligned}\Delta\Phi &\geq \frac{c \left( \frac{L_t(N_j)^2}{2} - \frac{2(\alpha+2)L_t(N_j)^2}{\alpha^2} \right)}{\bar{L}_t} \\ &= \frac{c \left( L_t(N_j)^2 \left( \frac{1}{2} - \frac{2(\alpha+2)}{\alpha^2} \right) \right)}{\bar{L}_t}.\end{aligned}$$

เมื่อพิจารณาคุณสมบัติที่ไม่เปลี่ยนแปลง เราจะได้ว่า

$$\bar{L}_t \leq (\alpha + 2)L_t(N_k) \leq (\alpha + 2)L_t(N_j) \text{ ดังนั้นจะได้}$$

$$\Delta\Phi \geq cL_t(N_j) \left( \frac{1}{2(\alpha + 2)} - \frac{2}{\alpha^2} \right).$$

จำนวนของการย้ายข้อมูลของขั้นตอน MINBALANCE จะมีค่าเป็น  $\left\lfloor \frac{L_t(N_j)}{2} \right\rfloor + L_t(N_k) < L_t(N_j)$  สำหรับค่า  $c > \left( \frac{2\alpha^2(\alpha+2)}{\alpha^2-4(\alpha+2)} \right)$  ใดๆ โดยที่ค่า  $\alpha > 2(1+\sqrt{3}) \approx 5.464$  เราจะได้ความสัมพันธ์ว่า  $\Delta\Phi \geq L_t(N_j)$  ดังนั้นค่าใช้จ่ายของการย้ายข้อมูลจะถูกจ่ายด้วยการลดลงของพลังงานศักย์ ■

เนื่องจากเราต้องการให้อัตราส่วนความไม่สมดุลเป็นค่าคงที่และต้องการให้ค่าใช้จ่ายของขั้นตอนวิธีเป็นค่าคงที่ ดังนั้นเราต้องกำหนดค่า  $\alpha > 2(1+\sqrt{3}) \approx 5.464$  ทำให้ได้ทฤษฎีบท

#### ทฤษฎีบท 4

ถ้า  $\alpha > 2(1+\sqrt{3}) \approx 5.464$  แล้วเราจะได้ว่าอัตราส่วนความไม่สมดุลมีค่าเป็น  $\alpha + 2$  และค่าใช้จ่ายถัวเฉลี่ยของขั้นตอนวิธีของเราในกรณีที่มีแต่การใส่กุญแจเป็นค่าคงที่

### 3.3 การวิเคราะห์อัตราส่วนความไม่สมดุลในกรณีทั่วไป

สำหรับการวิเคราะห์กรณีทั่วไป เราใช้สัญลักษณ์ต่างๆ ที่เกี่ยวกับเวลาและคุณสมบัติที่ไม่เปลี่ยนแปลงเช่นเดียวกับการวิเคราะห์ในกรณีที่มีแต่การใส่กุญแจ เราจะวิเคราะห์อัตราส่วนความไม่สมดุลหลังจากเกิดเหตุการณ์ใดๆ สำหรับการใส่กุญแจนั้นสามารถพิสูจน์ได้เช่นเดียวกับกรณีที่มีแต่การใส่กุญแจ ดังนั้นเราจะวิเคราะห์การลบกุญแจ เหตุการณ์ min-transfer และอีกสองเหตุการณ์ได้แก่

1. เหตุการณ์ nbr-transfer ที่เกิดที่โหนด  $z$  เป็นเหตุการณ์ที่เกิดขึ้นเมื่อโหนด  $z$  รับภาระจากโหนดเพื่อนบ้านของมันที่เรียกขั้นตอน SPLITMAX

2. เหตุการณ์ nbr-share ที่เกิดที่โหนด  $z$  เป็นเหตุการณ์ที่เกิดขึ้นเมื่อโหนด  $z$  แบ่งภาระกับโหนดเพื่อนบ้านของมันที่เรียกขั้นตอน SPLITNBR

กำหนดให้  $Max_t$  และ  $Max'_t$  แทนภาระที่สูงที่สุดในระบบหลังจากเวลา  $t$  และก่อนเวลา  $t$  ตามลำดับนั้นคือ  $Max_t = \max_{u \in V} L_t(u)$  และ  $Max'_t = \max_{u \in V} L'_t(u)$

## ภาพรวมของการวิเคราะห์ของกรณีทั่วไป

ปัญหาหลักของการนำเอาบทพิสูจน์ในกรณีที่มีแค่การใส่กุญแจมาใช้ในกรณีทั่วไปคือ คุณสมบัติที่ว่าภาระต่ำสุดจะไม่ลดลง ในการที่จะทำให้คุณสมบัตินี้ใช้งานได้ เราจะวิเคราะห์การทำงานเป็นเฟส (Phase) โดยที่แต่ละเฟสจะครอบคลุมช่วงที่ภาระต่ำสุดไม่ลดลง การทำเช่นนี้ทำให้เราสามารถนำเอาเทคนิคส่วนใหญ่ที่ใช้กับกรณีที่มีแค่การใส่กุญแจมาใช้กับกรณีทั่วไปได้ ดังนั้นในการวิเคราะห์เหตุการณ์มีเงื่อนไขคือเหตุการณ์ต่างๆ ที่เกิดขึ้นไม่ทำให้เกิดการเปลี่ยนเฟส

ทางเดียวที่จะทำให้เกิดการเปลี่ยนเฟสคือ โหนดที่มีภาระต่ำสุดมีภาระลดลง เหตุการณ์นี้สามารถเกิดขึ้นได้เมื่อมีการลบกุญแจที่โหนดที่มีภาระต่ำสุด เราจะแสดงว่าถ้ามีการลบกุญแจเกิดขึ้นที่โหนดที่มีภาระต่ำสุดและเป็นจุดเริ่มการเปลี่ยนเฟสแล้ว อัตราส่วนความไม่สมดุลจะถูกจำกัดด้วยค่าคงที่ค่าหนึ่ง ทำให้ได้เงื่อนไขเริ่มต้นที่เพียงพอสำหรับการวิเคราะห์ในเฟสต่อไป

## สถานะระหว่างเฟส

เราจะวิเคราะห์สถานะระหว่างเฟสที่ติดกันซึ่งเกิดขึ้นเมื่อภาระต่ำสุดลดลง เมื่อพิจารณาการลบกุญแจที่เกิดขึ้นที่โหนดใดๆ บทตั้งบทต่อไปแสดงคุณสมบัติของภาระของโหนดหลังจากที่มีการลบกุญแจเกิดขึ้น

### บทตั้ง 13

สมมติว่า  $\beta > 3$  พิจารณากรณีที่มีการลบกุญแจเกิดขึ้นที่โหนด  $u$  ณ บางเวลา  $t$  กำหนดให้โหนด  $z$  เป็นโหนดเพื่อนบ้านที่มีภาระต่ำที่สุดของโหนด  $u$  ณ เวลา  $t$  หลังจากการลบกุญแจและขั้นตอนปรับดุลภาระจะได้ว่า  $L_t(u) > \frac{Max_t}{\beta}$  อีกทั้งภาระต่ำสุดจะสามารถลดลงได้ในกรณีที่ขั้นตอนการปรับดุลภาระไม่ถูกเรียก

## พิสูจน์

หลังจากการลบกุญแจเกิดขึ้นที่โหนด  $u$  มีกรณีที่จะต้องพิจารณา 2 กรณีคือ

กรณีที่หนึ่ง ขั้นตอนการปรับดุลภาระไม่ถูกเรียก ในกรณีที่ภาระของโหนด  $u$  หลังจากการลบกุญแจมีค่ามากกว่า  $\frac{Max'_t}{\beta}$  สังเกตได้ว่า ณ เวลา  $t$  เหตุการณ์เดียวที่เกิดขึ้นในระบบได้คือการลบที่โหนด  $u$  ดังนั้นภาระมากที่สุดจะไม่เพิ่มขึ้น นั่นคือ  $Max'_t \geq Max_t$  ทำให้ได้ความสัมพันธ์ว่า  $L_t(u) > \frac{Max'_t}{\beta} \geq \frac{Max_t}{\beta}$

กรณีที่สอง ขั้นตอนการปรับดุลภาระถูกเรียก ซึ่งในกรณีที่ภาระของโหนด  $u$  จะไม่เกิน  $\frac{2Max'_t}{\beta}$  กรณีนี้สามารถพิจารณาได้เป็นสองกรณีย่อยได้แก่

กรณีย่อย 1 โหนด  $u$  เรียกขั้นตอน SPLITMAX เริ่มต้นโหนด  $u$  จะโอนภาระทั้งหมดไปให้กับโหนด  $z$  หลังจากนั้นโหนด  $u$  จะไปแบ่งครึ่งภาระของโหนดที่มีภาระสูงสุด นั่นคือ

$$L_t(u) = \left\lfloor \frac{Max'_t}{2} \right\rfloor > \frac{Max'_t}{\beta}$$

ในกรณีย่อยนี้ โหนดที่มีภาระเพิ่มขึ้น ณ เวลา  $t$  นอกจากโหนด  $u$  คือ โหนด  $z$  สังเกตได้ว่าโหนด  $z$  จะรับภาระจากโหนด  $u$  จะได้ว่า

$$L_t(z) = L'_t(z) + L'_t(u) \leq \frac{2Max'_t}{\beta} + \frac{Max'_t}{\beta} \leq \frac{3Max'_t}{\beta}$$

จากข้อสมมติที่ว่า  $\beta > 3$  ทำให้ได้  $L_t(z) < Max'_t$  ดังนั้นภาระสูงสุดไม่เพิ่มขึ้น ณ เวลา  $t$  จะได้ว่า  $L_t(u) > \frac{Max_t}{\beta}$

กรณีย่อย 2 โหนด  $u$  เรียกขั้นตอน SPLITNBR กรณีย่อยนี้เกิดขึ้นเมื่อ  $L'_t(z) > \frac{2Max'_t}{\beta}$  ภาระของโหนด  $z$  จะถูกแบ่งให้กับโหนด  $u$  เพื่อปรับดุลภาระทั้งสองโหนด จะได้ว่า

$$L_t(u) \geq \frac{\frac{2Max'_t}{\beta} + L'_t(u)}{2} > \frac{Max'_t}{\beta}.$$

ในกรณีข้อย่อยนี้จะมีสองโหนดที่เกี่ยวข้องได้แก่โหนด  $u$  และโหนด  $z$  สังเกตได้ว่ามีโหนด  $u$  เพียงโหนดเดียวในระบบที่มีภาระเพิ่มขึ้น ณ เวลา  $t$  เราได้ว่า  $L'_t(u) \leq \frac{Max'_t}{\beta} < \frac{Max'_t}{2}$  ดังนั้นภาระของโหนด  $u$  หลังจากขั้นตอนปรับดุลภาระจะไม่เกิน  $Max'_t$  เพราะภาระของมันน้อยกว่า  $\frac{Max'_t}{2}$  และภาระที่มันรับมาจะไม่เกิน  $\frac{Max'_t}{2}$  ดังนั้นภาระสูงสุดจะไม่เพิ่มขึ้น ณ เวลา  $t$  เราจะได้ว่า  $L_t(u) > \frac{Max_t}{\beta}$

พิจารณาภาระของโหนดที่มีภาระต่ำสุดหลังจากมีการลบกุญแจเกิดขึ้นที่มัน ณ เวลา  $t$  มีกรณีที่สามารถเกิดขึ้นได้ 2 กรณี ขึ้นกับภาระของมัน กรณีแรกถ้าภาระของมันมีค่ามากกว่า  $\frac{Max'_t}{\beta}$  ขั้นตอนปรับดุลภาระจะไม่ถูกเรียก จากการพิสูจน์ในกรณีที่หนึ่งทำให้รู้ว่าภาระต่ำสุดสามารถลดลงได้ กรณีที่สองถ้าภาระของมันมีค่าไม่เกิน  $\frac{Max'_t}{\beta}$  ขั้นตอนปรับดุลภาระจะถูกเรียก จากการพิสูจน์ในกรณีที่สองทำให้รู้ว่าภาระหลังจากขั้นตอนปรับดุลภาระจะมากกว่า  $\frac{Max_t}{\beta}$  ซึ่งในกรณีนี้ภาระต่ำสุดจะไม่ลดลงหลังจากการลบกุญแจ ดังนั้นกรณีที่ขั้นตอนปรับดุลภาระจะไม่ถูกเรียกจะเป็นกรณีที่โหนดที่มีภาระต่ำสุดมีภาระลดลงได้ ■

จากบทตั้ง 13 ภาระต่ำสุด ณ เวลา  $t$  สามารถลดลงในกรณีที่มีการลบแล้วขั้นตอนปรับดุลภาระไม่ถูกเรียก เราจะได้ความสัมพันธ์ว่า  $Min_t > \frac{Max_t}{\beta}$  หลังจากนั้นเฟสจะเปลี่ยน อีกทั้งเงื่อนไขต่อไปนี้เป็นจริง

เมื่อเริ่มต้นแต่ละเฟส ณ เวลา  $t$  อัตราส่วนความไม่สมดุลรับประกันได้ด้วยค่า  $\beta$  นั่นคือ

$$\frac{Max_t}{Min_t} < \beta.$$

อัตราส่วนความไม่สมดุลภายในแต่ละเฟส

เราจะพิสูจน์คุณสมบัติที่ไม่เปลี่ยนแปลงเช่นเดียวกับกรณีที่มิแต่การใส่กุญแจ นั่นคือ

"สำหรับเวลา  $t$  ใดๆ ภาระของโหนดใดๆ จะไม่เกิน  $(\alpha + 2)$  เท่าของภาระที่ต่ำที่สุด"

ในตอนเริ่มต้นของแต่ละเฟส อัตราส่วนความไม่สมดุลจะมีค่าไม่เกิน  $\beta$  เมื่อเราเลือกค่า  $\beta$  โดยให้  $\beta \leq \alpha$  แล้วจะได้ว่าตอนเริ่มต้นของแต่ละเฟส  $Max_t \leq \beta Min_t \leq (\alpha + 2) Min_t$  ซึ่งสอดคล้องกับเงื่อนไขของคุณสมบัติที่ไม่เปลี่ยนแปลง

ต่อจากนี้ไปในการวิเคราะห์ของเรา เราจะไม่สนใจกรณีของการลบที่ไม่ได้ตามด้วยขั้นตอนปรับดุลภาระ เพราะว่าภาระของโหนดที่ได้รับผลกระทบจะไม่ไปเปลี่ยนค่าของอัตราส่วนความไม่สมดุล

ในการวิเคราะห์อัตราส่วนความไม่สมดุล เราพิจารณาว่าภาระของโหนดที่ได้รับผลกระทบเปลี่ยนแปลงอย่างไร ก่อนอื่นเราจะพิจารณาสองเหตุการณ์ที่ง่ายก่อน สำหรับการใส่กุญแจ เราจะใช้บทตั้ง 9 เพื่อรับประกันภาระของโหนดที่มีการใส่กุญแจ สำหรับการลบกุญแจเราจะใช้บทตั้ง 13 เพื่อรับประกันภาระของโหนดที่มีการลบ มีเหตุการณ์ที่ต้องพิจารณาเนื่องจากโหนดได้รับผลกระทบจากขั้นตอนปรับดุลภาระ มีเหตุการณ์สามชนิดได้แก่ nbr-transfer, nbr-share และ min-transfer

เราจะสรุปเหตุการณ์ต่างๆ และวิธีการจัดการดังต่อไปนี้

1. สำหรับเหตุการณ์ nbr-transfer เราสนใจโหนดเพื่อนบ้านที่มีภาระต่ำที่สุดของโหนดที่มีการลบกุญแจเกิดขึ้นที่มัน โหนดเพื่อนบ้านนี้จะรับภาระทั้งหมดจากโหนดที่มีการลบกุญแจ โดยบทตั้ง 14 จะจัดการเหตุการณ์นี้

2. สำหรับเหตุการณ์ nbr-share โหนดเพื่อนบ้านที่มีภาระต่ำของโหนดที่มีการลบกุญแจจะแบ่งภาระให้กับโหนดที่มีการลบกุญแจทำให้ทั้งสองโหนดมีภาระเท่ากัน เหตุการณ์นี้อัตราส่วนความไม่สมดุลจะถูกพิสูจน์ในบทตั้ง 14

3. สำหรับเหตุการณ์ min-transfer เราจะสนใจโหนดเพื่อนบ้านของโหนดที่มีภาระต่ำสุด ซึ่งมันจะได้รับภาระจากโหนดที่มีภาระต่ำสุด บทตั้ง 18 จะจัดการเหตุการณ์นี้

บทตั้งบทต่อไปจะจัดการกรณีเหตุการณ์ nbr-transfer และเหตุการณ์ nbr-share

## บทตั้ง 14

สมมติว่า  $\alpha \geq 3$  และ  $\beta \geq \frac{3(\alpha+2)}{\alpha}$  กำหนดให้  $z$  เป็นโหนดเพื่อนบ้านของโหนด  $u$  ที่มีภาระต่ำที่สุด ณ เวลา  $t$  พิจารณากรณีที่การลบกุญแจเกิดขึ้นที่โหนด  $u$  ณ เวลา  $t$  และโหนด  $u$  เรียกขั้นตอนปรับดุลภาระ จะได้ว่า  $L_t(u) \leq \alpha Min_t$  และ  $L_t(z) \leq \alpha Min_t$

## พิสูจน์

หลังจากการลบกุญแจมีขั้นตอนปรับดุลภาระสองชนิดที่เกิดขึ้นได้

กรณีที่ 1 โหนด  $u$  เรียกขั้นตอน SPLITMAX และเหตุการณ์ nbr-transfer เกิดขึ้นกับโหนด  $z$  ในกรณีนี้จะเกิดขึ้นเมื่อ  $L'_t(z) \leq \frac{2Max'_t}{\beta}$  และ  $L'_t(u) \leq \frac{Max'_t}{\beta}$  จากนั้นโหนด  $u$  จะดำเนินการต่อโดยโอนภาระทั้งหมดของมันให้โหนด  $z$  นั่นคือ

$$L_t(z) \leq \frac{2Max'_t}{\beta} + \frac{Max'_t}{\beta}.$$

จากคุณสมบัติที่ไม่เปลี่ยนแปลง เราจะได้ว่า  $L_t(z) \leq \frac{3((\alpha+2)Min'_t)}{\beta}$  จากข้อสมมติที่ว่า  $\beta \geq \frac{3(\alpha+2)}{\alpha}$  ทำให้ได้ความสัมพันธ์  $\frac{3(\alpha+2)}{\beta} \leq \alpha$  ดังนั้นภาระของโหนด  $z$  ณ เวลา  $t$  จะไม่เกิน  $\alpha Min'_t$  จากคุณสมบัติที่ไม่ลดลงของภาระต่ำสุด เราจะได้ว่า  $L_t(z) \leq \alpha Min_t$

หลังจากนั้นโหนด  $u$  จะแบ่งภาระครึ่งหนึ่งจากภาระของโหนดที่มีภาระสูงสุด จากคุณสมบัติที่ไม่ลดลงของภาระต่ำสุด เราจะได้ว่า  $L_t(u) \leq \frac{(\alpha+2)Min'_t}{2}$  เมื่อค่า  $\alpha \geq 2$  ทำให้ได้ความสัมพันธ์  $\frac{(\alpha+2)}{2} \leq \alpha$  ดังนั้นภาระของโหนด  $u$  หลังจากการลบกุญแจจะไม่เกิน  $\alpha Min'_t$  จากนั้นเราจะได้ว่า  $L_t(u) \leq \alpha Min_t$  เนื่องจากคุณสมบัติที่ไม่ลดลงของภาระต่ำสุด

กรณีที่ 2 โหนด  $u$  เรียกขั้นตอน SPLITNBR และเหตุการณ์ nbr-share เกิดขึ้นที่โหนด  $z$  ในกรณีนี้เกิดขึ้นเมื่อ  $L'_t(z) > \frac{2Max'_t}{\beta}$  และ  $L'_t(u) \leq \frac{Max'_t}{\beta}$  จากคุณสมบัติที่ไม่เปลี่ยนแปลง เราจะได้ว่า

$L_t'(z) \leq \text{Max}_t' \leq (\alpha + 2)\text{Min}_t'$  หลังจากนั้น โหนด  $u$  และ โหนด  $z$  จะแบ่งภาระให้เท่ากัน นั่นคือ

$$L_t(z) \leq \frac{(\alpha + 2)\text{Min}_t' + (\alpha + 2)\frac{\text{Min}_t'}{\beta}}{2} \leq \frac{(\beta + 1)(\alpha + 2)}{2\beta}\text{Min}_t'$$

จากที่สมมติว่า  $\alpha \geq 3$  และ  $\beta \geq \frac{3(\alpha+2)}{\alpha}$  เราจะได้ความสัมพันธ์ว่า  $\beta > \frac{(\alpha+2)}{(\alpha-2)}$  ยิ่งไปกว่านั้น เมื่อค่า  $\beta > \frac{(\alpha+2)}{(\alpha-2)}$  และ  $\alpha > 2$  ทำให้ได้ความสัมพันธ์ว่า  $\frac{(\beta+1)(\alpha+2)}{2\beta} \leq \alpha$  ดังนั้นภาระของโหนด  $z$  และ โหนด  $u$  หลังจากขั้นตอนปรับดุลภาระจะไม่เกิน  $\alpha\text{Min}_t'$  จากคุณสมบัติที่ไม่ลดลงของภาระที่ต่ำที่สุด เราจะได้ว่า  $L_t(u) \leq \alpha\text{Min}_t$  และ  $L_t(z) \leq \alpha\text{Min}_t$  ■

มีกรณีที่ต้องพิจารณาอีกคือ กรณีเหตุการณ์ min-transfer กำหนดให้  $e$  เป็นเหตุการณ์ min-transfer ที่เกิดขึ้นกับ โหนด  $z$  ในบทตั้งบทต่อไปจะเกี่ยวกับกรณีที่  $e$  เป็นเหตุการณ์แรกที่เกิดขึ้นกับ โหนด  $z$  ในเฟส  $d$  ใดๆ

บทตั้ง 15

พิจารณาเฟส  $d$  ใดๆ สมมติว่า  $\beta > 2$  ถ้าเหตุการณ์แรกที่เกิดขึ้นกับโหนด  $z$  ในเฟสนั้นเป็น เหตุการณ์ min-transfer ครั้งที่  $i$  จะได้ว่า  $L_{t_i}(z) \leq (\beta + 1)\text{Min}_{t_i}$

พิสูจน์

มีสองกรณีที่ต้องพิจารณา

กรณีแรก เป็นกรณีที่  $d = 1$  หรือกรณีที่เฟสแรก ในตอนเริ่มต้นของเฟสแรก ภาระของแต่ละโหนดจะมีค่าเท่ากับ  $c_0$  สังเกตว่าภาระของโหนด  $z$  จะไม่เปลี่ยนแปลงจนกระทั่ง เหตุการณ์ min-transfer ครั้งแรกเกิดกับมัน ยิ่งไปกว่านั้น ณ เวลา  $t_1$  ภาระต่ำสุดมีค่าเป็น  $c_0$  เพราะว่า มันไม่สามารถลดลงและมันไม่สามารถเพิ่มเกิน  $c_0$  ได้ เมื่อเหตุการณ์ min-transfer ที่เกิดขึ้นกับโหนด  $z$  ภาระของมันจะเพิ่มขึ้น นั่นคือ  $L_{t_1}(z) = c_0 + \text{Min}_{t_1}' = 2\text{Min}_{t_1}'$  จากคุณสมบัติที่ไม่ลดลงของภาระต่ำสุด เราจะได้ว่า  $L_{t_1}(z) \leq 2\text{Min}_{t_1}$  จากข้อสมมติ  $\beta > 2$  จะได้ว่า  $L_{t_1}(z) \leq (\beta + 1)\text{Min}_{t_1}$

กรณีที่สอง เป็นกรณีที่  $d > 1$  หรือกรณีที่ไม่ใช่เฟสแรก กำหนดให้  $t'$  เป็นเวลาเริ่มต้นของเฟส  $d$  จากเงื่อนไขของภาระตอนเริ่มต้นของแต่ละเฟส  $Max_{t'} < \beta Min_{t'}$  ดังนั้น  $L_{t'}(z) < \beta Min_{t'}$  หลังจากเหตุการณ์ min-transfer ครั้งที่  $i$  เกิดขึ้นกับโหนด  $z$  ภาระของมันจะเพิ่มขึ้นเป็น  $L_{t_i}(z) = L_{t'}(z) + Min'_{t_i} \leq \beta Min_{t'} + Min'_{t_i}$  จากที่ภาระที่ต่ำที่สุดจะไม่ลดลงภายในเฟสเราจะได้ว่า  $L_{t_i}(z) \leq (\beta + 1)Min_{t_i}$  ตามต้องการ ■

ต่อไปเราจะสมมติว่า  $e$  ไม่ใช่เหตุการณ์แรกที่เกิดขึ้นกับโหนด  $z$  ในเฟส  $d$

กำหนดให้  $e'$  เป็นเหตุการณ์ล่าสุดที่เกิดขึ้นกับโหนด  $z$  ก่อนเหตุการณ์  $e$  บทตั้งบทต่อไปจะพิจารณากรณีที่เหตุการณ์  $e'$  ไม่ใช่เหตุการณ์ min-transfer ขณะที่บทตั้ง 17 ซึ่งเป็นบทตั้งที่ซับซ้อนกว่าซึ่งจะพิจารณากรณีที่เหตุการณ์  $e'$  เป็นเหตุการณ์ min-transfer

บทตั้ง 16

พิจารณาเฟสใดๆ สมมติว่า  $\alpha \geq 3$  และ  $\beta \geq \frac{3(\alpha+2)}{\alpha}$  ถ้า  $e'$  เป็นเหตุการณ์ใดๆ ยกเว้นเหตุการณ์ min-transfer ที่เกิดขึ้นกับโหนด  $z$  ก่อนที่จะมีเหตุการณ์ min-transfer ครั้งที่  $i$  แล้วหลังจากเหตุการณ์ min-transfer ครั้งที่  $i$  เกิดขึ้นกับโหนด  $z$  จะได้ว่า  $L_{t_i}(z) \leq (\alpha + 1)Min_{t_i}$

พิสูจน์

เหตุการณ์  $e'$  สามารถเป็นได้คือ การใส่กุญแจ, การลบกุญแจที่ตามด้วยขั้นตอนปรับดุลภาระ, เหตุการณ์ nbr-transfer หรือเหตุการณ์ nbr-share หลังจากเหตุการณ์  $e'$  เกิดขึ้นกับโหนด  $z$  ณ เวลา  $t'$  สำหรับการใส่กุญแจพิจารณาจากบทตั้ง 9 และสำหรับเหตุการณ์อื่นๆ พิจารณาบทตั้ง 14 เราจะได้ว่า  $L_{t'}(z) \leq \alpha Min_{t'}$  หลังจากนั้นภาระของโหนด  $z$  จะเพิ่มขึ้นอีกครั้งจากเหตุการณ์ min-transfer ครั้งที่  $i$  นั่นคือ  $L_{t_i}(z) = L_{t'}(z) + Min'_{t_i} \leq \alpha Min_{t'} + Min'_{t_i}$  เนื่องจากภาระต่ำสุดไม่ลดลงในแต่ละเฟสเราจะได้ว่า  $L_{t_i}(z) \leq (\alpha + 1)Min_{t_i}$  ■

สุดท้ายเราจะพิจารณากรณีเมื่อเหตุการณ์ล่าสุด  $e'$  ก่อนเหตุการณ์  $e$  เป็นเหตุการณ์ min-transfer เช่นเดียวกับกรณีที่มีแต่การใส่กุญแจเราแบ่งเหตุการณ์ min-transfer ออกเป็นสองชนิดคือ เหตุการณ์ left-transfer และเหตุการณ์ right-transfer บทตั้งบทต่อไปจะเป็นกรณีทั่วไปของบทตั้ง 11 ที่จัดการการลบ SPLITMAX และ SPLITNBR ได้ด้วย

บทตั้ง 17

สมมติว่า  $\alpha \geq \frac{3+\sqrt{33}}{2} \approx 4.373$ ,  $\beta \geq \frac{3(\alpha+2)}{\alpha}$  และ  $l \geq 0$  พิจารณาเหตุการณ์ min-transfer ครั้งที่  $i, (i+1), \dots, (i+l+1)$  ที่เกิดกับโหนด  $z$  ในเฟสใดๆ ถ้าเหตุการณ์ min-transfer ครั้งที่  $i$  และ ครั้งที่  $(i+l+1)$  เป็นชนิดเดียวกัน ในขณะที่เหตุการณ์ min-transfer ครั้งที่  $(i+1), (i+2), \dots, (i+l)$  เป็นชนิดต่างกับครั้งที่  $i$  และ  $(i+l+1)$  แล้ว หลังจากเหตุการณ์ min-transfer ครั้งที่  $(i+l+1)$  จะได้  $L_{t_{i+l+1}}(z) \leq (\alpha + l + 1)Min_{t_{i+l+1}}$

พิสูจน์

โดยไม่เสียหลักการทั่วไปเราจะสมมติว่าเหตุการณ์ min-transfer ครั้งที่  $i$  และครั้งที่  $(i+l+1)$  ที่เกิดขึ้นกับโหนด  $z$  เป็นเหตุการณ์ right-transfer ส่วนเหตุการณ์ min-transfer ครั้งที่  $(i+1), (i+2), \dots, (i+l)$  เป็นเหตุการณ์ left-transfer กำหนดให้  $v_i$  แทนโหนดที่มีภาระต่ำสุด ณ เวลา  $t_i$

เราจะเริ่มต้นด้วยกรณีที่มีเหตุการณ์อื่นยกเว้นเหตุการณ์ min-transfer เกิดขึ้นที่โหนด  $z$  ระหว่างเหตุการณ์ min-transfer ครั้งที่  $i$  และครั้งที่  $(i+l+1)$  สมมติว่าเหตุการณ์ล่าสุดที่ไม่ใช่เหตุการณ์ min-transfer ที่เกิดที่โหนด  $z$  ก่อนเหตุการณ์ min-transfer ครั้งที่  $k$  เมื่อ  $i < k \leq i+l+1$  จากบทตั้ง 16 หลังจากเหตุการณ์ min-transfer ครั้งที่  $k$  เราจะได้ว่า

$$L_{t_k}(z) \leq (\alpha + 1)Min_{t_k}$$

หลังจากนั้นเหตุการณ์ถัดจาก min-transfer ครั้งที่  $k$  มีแต่เหตุการณ์ min-transfer ที่สามารถเกิดกับโหนด  $z$  สังเกตว่าหลังจากเวลา  $t_k$  จะมีเหตุการณ์ min-transfer เกิดขึ้นกับโหนด  $z$  ไม่เกิน  $i+l-k$

ครั้งประกอบกับภาระต่ำสุดไม่ลดลง ณ เวลา  $t_{i+l+1}$  เราจะได้ว่า

$$L_{t_{i+l+1}}(z) \leq (\alpha + 1)Min_{t_{i+l+1}} + (i + l - k)Min_{t_{i+l+1}},$$

และสุดท้ายเราจะได้  $L_{t_{i+l+1}}(z) \leq (\alpha + l + 1)Min_{t_{i+l+1}}$  เนื่องจาก  $k > i$

มีกรณีเหลือที่จะต้องพิจารณาคือกรณีที่ไม่มีเหตุการณ์อื่นเกิดขึ้นกับโหนด  $z$  ยกเว้นเหตุการณ์ min-transfer ระหว่างเหตุการณ์ min-transfer ครั้งที่  $i$  ถึง ครั้งที่  $(i + l + 1)$  เราจะดำเนินการตามบทอ้าง 1 ในบทพิสูจน์ของบทตั้ง 11 และใช้คุณสมบัติที่ว่าแต่ละเฟสภาระต่ำสุดจะไม่ลดลงดังนั้นเราจะได้ว่า

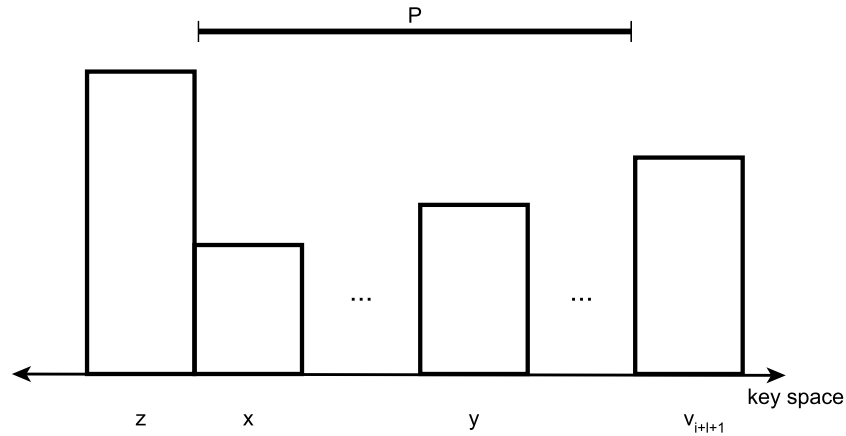
$$L_{t_{i+l+1}}(z) \leq L_{t_i}(z) + (l + 1)Min'_{t_{i+l+1}} \quad (10)$$

พิจารณาเหตุการณ์ min-transfer ครั้งที่  $i$  ที่เกิดกับโหนด  $z$  จากที่เราได้สมมติไว้ว่ามันเป็นเหตุการณ์ right-transfer กำหนดให้  $x$  เป็นโหนดทางขวามือของโหนด  $v_i$  ก่อนที่จะเกิดเหตุการณ์ min-transfer ครั้งที่  $i$  สังเกตได้ว่าเหตุการณ์ min-transfer ครั้งที่  $(i + l + 1)$  เป็นเหตุการณ์ right-transfer เช่นกัน ดังนั้นโหนด  $x$  จะต้องกลายเป็นโหนดที่มีภาระต่ำสุด ณ บางเวลาหลังจากเวลา  $t_i$  กำหนดให้  $t^*$  เป็นเวลาที่โหนด  $x$  กลายเป็นโหนดที่มีภาระต่ำสุดหลังจากเวลา  $t_i$

แทนที่เราจะพิจารณาเฉพาะโหนด  $x$  เราจะพิจารณาเซตของโหนด  $P$  ที่เรียงกันตามลำดับจากโหนด  $x$  ไปยังโหนด  $v_{i+l+1}$  ระหว่างเวลา  $t_i$  และ  $t_{i+l+1}$  (ดูรูป 21 (จ)) เราจะหาขอบเขตของภาระของโหนด  $z$  จากภาระของโหนดในเซตนี้ เราพิจารณาได้เป็นสองกรณี

กรณีที่ 1 ไม่มีโหนดใดๆ ในเซต  $P$  ที่เรียกขั้นตอน MINBALANCE, SPLITMAX และ SPLITNBR ระหว่างเวลา  $t_i$  และ  $t_{i+l+1}$  ในกรณีนี้เราจะพิจารณาโหนด  $x$  สังเกตได้ว่าอาจจะมีกรลบเกิดขึ้นกับโหนด  $x$  ดังนั้นมีสองกรณีย่อยที่ต้องพิจารณา

กรณีย่อย 1.1 ไม่มีการลบเกิดขึ้นที่โหนด  $x$  หลังจากเวลา  $t_i$  พิจารณาเหตุการณ์ min-transfer ครั้งที่  $i$  ที่เกิดขึ้นที่โหนด  $z$  โหนดที่มีภาระที่ต่ำสุด ณ เวลา  $t_i$  จะโอนภาระของมันทั้งหมด



ภาพที่ 21 เขตของโหนด  $P$  ที่อยู่ระหว่างโหนด  $x$  และโหนด  $v_{i+l+1}$

ให้กับโหนด  $z$  นั้นหมายความว่า  $L'_{t_i}(z) \leq L'_{t_i}(x)$  จากนั้นเราจะได้

$$L_{t_i}(z) = L'_{t_i}(z) + \text{Min}'_{t_i} \leq L'_{t_i}(x) + \text{Min}'_{t_i},$$

เมื่อรวมกับอสมการ (10) เราจะได้

$$L_{t_{i+l+1}}(z) \leq L'_{t_i}(x) + \text{Min}'_{t_i} + (l+1)\text{Min}'_{t_{i+l+1}}. \quad (11)$$

หลังจากเวลา  $t_i$  การลบกุญแจและขั้นตอนปรับดุลภาระไม่เกิดขึ้นกับโหนด  $x$  การดำเนินการที่เกิดขึ้นได้คือการใส่กุญแจ ดังนั้นภาระของโหนด  $x$  จะไม่ลดลง เมื่อเวลา  $t^*$  โหนด  $x$  จะกลายเป็นโหนดที่มีภาระต่ำสุด จะได้ว่า  $L'_{t_i}(x) \leq \text{Min}_{t^*}$  เมื่อรวมกับอสมการ (11) เราจะได้ว่า

$$L_{t_{i+l+1}}(z) \leq \text{Min}_{t^*} + \text{Min}'_{t_i} + (l+1)\text{Min}'_{t_{i+l+1}}.$$

เนื่องจากภาระต่ำสุดไม่ลดลง จะได้ว่า  $L_{t_{i+l+1}}(z) \leq (l+3)\text{Min}_{t_{i+l+1}}$  เมื่อ  $\alpha \geq 2$  เราจะได้ว่า  $L_{t_{i+l+1}}(z) \leq (\alpha + l + 1)\text{Min}_{t_{i+l+1}}$  ดังนั้นบทตั้งนี้เป็นจริงในกรณีข้อย่อยนี้

กรณีข้อย่อย 1.2 มีการลบเกิดขึ้นที่โหนด  $x$  หลังจากเวลา  $t_i$  แต่ขั้นตอน SPLIT-

MAX และ SPLITNBR ไม่ถูกเรียก กำหนดให้  $t'$  เป็นเวลาที่โหนด  $x$  มีภาระต่ำที่สุดหลังจากการลบกุญแจเกิดขึ้นที่มันระหว่างช่วงเวลา  $t_i$  และ  $t^*$  จากบทตั้ง 13 ภาระมากที่สุด ณ เวลา  $t'$  จะมีค่าไม่เกิน  $\beta$  เท่าของภาระ  $x$  ณ เวลา  $t'$  ดังนั้นเราจะได้ว่า

$$Max_{t'} \leq \beta L_{t'}(x) \leq \beta L_{t^*}(x). \quad (12)$$

สังเกตได้ว่าเหตุการณ์เดียวที่จะเกิดขึ้นกับโหนด  $z$  ระหว่างช่วงเวลา  $t_i$  และ  $t'$  คือเหตุการณ์ min-transfer ดังนั้นภาระของโหนด  $z$  หลังจากเวลา  $t_i$  จะไม่ลดลง เราจะได้ว่า  $L_{t_i}(z) \leq L_{t'}(z)$  เมื่อพิจารณาสมการ (10) จะได้ว่า

$$L_{t_{i+l+1}}(z) \leq L_{t'}(z) + (l+1)Min'_{t_{i+l+1}}.$$

จากสมการ (12) จะได้

$$L_{t_{i+l+1}}(z) \leq Max_{t'} + (l+1)Min'_{t_{i+l+1}} \leq \beta L_{t^*}(x) + (l+1)Min'_{t_{i+l+1}}.$$

เนื่องจากในแต่ละเฟสภาระต่ำสุดจะไม่ลดลง เราจะได้ว่า  $L_{t_{i+l+1}}(z) \leq (\beta + l + 1)Min_{t_{i+l+1}}$  จากที่สมมติว่า  $\alpha \geq \frac{3+\sqrt{33}}{2}$  ทำให้ได้ทำให้ได้ความสัมพันธ์  $\alpha \geq \beta$  ดังนั้นบทตั้งนี้จึงเป็นจริงในกรณีย่อยนี้

กรณีที่ 2 มีอย่างน้อยหนึ่งโหนดในเซต  $P$  ที่เรียกขั้นตอน MINBALANCE หรือ SPLITMAX หรือ SPLITNBR ระหว่างช่วงเวลา  $t_i$  และ  $t_{i+l+1}$  กำหนดให้  $y$  เป็นโหนดล่าสุดในเซตนี้ที่เรียกขั้นตอนปรับดุลภาระ ณ เวลา  $\hat{t}$  เราจะพิจารณากรณีที่มีการลบกุญแจเกิดขึ้นที่โหนด  $y$  หลังจากเวลา  $\hat{t}$  กรณีนี้สามารถพิสูจน์ได้เช่นเดียวกับกรณีย่อย 1.2 ทำให้ได้ว่า  $L_{t_{i+l+1}}(z) \leq (\alpha + l + 1)Min_{t_{i+l+1}}$

ดังนั้นเหลือกรณีที่ ไม่มีการลบกุญแจเกิดขึ้นที่โหนด  $y$  หลังจากขั้นตอน MINBALANCE หรือ SPLITNBR หรือ SPLITMAX เกิดขึ้นกับโหนด  $y$  หลังจากเวลา  $\hat{t}$

กรณีย่อย 2.1 โหนด  $y$  เรียกขั้นตอน MINBALANCE ณ เวลา  $\hat{t}$  กรณีย่อยนี้

สามารถถูกพิสูจน์ได้เช่นเดียวกันกับกรณีที่สองในบทตั้ง 11 เมื่อ  $\alpha \geq 1 + \sqrt{5}$  จะได้ว่า  $\frac{L_{t_i+l+1}(z)}{Min_{t_i+l+1}} \leq (\alpha + l + 1)$  ดังนั้นกรณีย่อยนี้เป็นจริง

กรณีย่อย 2.2 โหนด  $y$  เรียกขั้นตอน SPLITNBR ณ เวลา  $\hat{t}$  จากบทตั้ง 13 หลังจากขั้นตอน SPLITNBR เราจะได้ว่า  $Max_{\hat{t}} \leq \beta L_{\hat{t}}(y)$  สังเกตได้ว่าเหตุการณ์ที่เกิดขึ้นกับโหนด  $z$  หลังจากเวลา  $t_i$  คือเหตุการณ์ min-transfer ดังนั้นภาระของโหนด  $z$  จะไม่ลดลงหลังจากเวลา  $t_i$  เราจะได้ว่า  $L_{t_i}(z) \leq L_{\hat{t}}(z)$  เมื่อพิจารณาร่วมกับอสมการ (10) จะได้ว่า

$$L_{t_i+l+1}(z) \leq Max_{\hat{t}} + (l+1)Min'_{t_i+l+1} \leq \beta L_{\hat{t}}(y) + (l+1)Min'_{t_i+l+1}$$

หลังจากเวลา  $\hat{t}$  ไม่มีขั้นตอนปรับดุลภาระหรือการลบกุญแจเกิดขึ้นที่โหนด  $y$  ดังนั้นหลังจาก  $\hat{t}$  ภาระของโหนด  $y$  จะไม่ลดลง เนื่องจากเหตุการณ์ min-transfer ครั้งที่  $(i + l + 1)$  เป็นเหตุการณ์ right-transfer ดังนั้นโหนด  $y$  จะกลายเป็นโหนดที่มีภาระต่ำสุด ณ บางเวลา กำหนดให้  $t''$  เป็นเวลาที่โหนด  $y$  กลายเป็นโหนดที่มีภาระต่ำสุด จะได้ว่า  $L_{\hat{t}}(y) \leq L_{t''}(y)$  ดังนั้นจะได้ว่า  $L_{t_i+l+1}(z) \leq (\beta + l + 1)Min_{t_i+l+1}$  เมื่อ  $\alpha \geq \frac{3+\sqrt{33}}{2}$  จะได้ว่า  $\alpha \geq \beta$  ดังนั้นกรณีย่อยนี้เป็นจริง

กรณีย่อย 2.3 โหนด  $y$  เรียกขั้นตอน SPLITMAX ณ เวลา  $\hat{t}$  หลังจากโหนด  $y$  เรียกขั้นตอนเหล่านั้น มันจะโอนภาระของมันทั้งหมดให้กับโหนดเพื่อนบ้านโหนด  $w$  หลังจากนั้นเราจะได้ว่า  $L_{\hat{t}}(w) \geq 2Min_{\hat{t}}$  จากนั้นโหนด  $y$  จะถูกย้ายตำแหน่ง ดังนั้นเราจะพิจารณาโหนด  $w$  แทน

เราจะแสดงว่าโหนด  $w$  อยู่ในเซต  $P$  เราจะพิสูจน์ด้วยวิธีขัดแย้งโดยจะสมมติว่าโหนด  $w$  ไม่อยู่ในเซต  $P$  สังเกตว่าตำแหน่งของโหนด  $w$  สามารถอยู่ทางซ้ายหรือทางขวาของ  $y$  ก็ได้

พิจารณากรณีที่โหนด  $w$  อยู่ทางขวาของโหนด  $y$  โหนด  $w$  สามารถที่จะอยู่นอกเซต  $P$  เมื่อโหนด  $y$  เป็นโหนดทางขวามือสุดในเซต  $P$  เนื่องจากโหนด  $v_{i+l+1}$  เป็นโหนดทางขวามือสุดในเซต  $P$  ซึ่งขัดแย้ง

พิจารณากรณีที่โหนด  $w$  อยู่ทางซ้ายของโหนด  $y$  ในกรณีนี้โหนด  $y$  จะต้องเป็นโหนดทางซ้ายมือสุดในเซต  $P$  และโหนด  $y$  จะโอนภาระของมันให้กับโหนดนอกเซต  $P$  สังเกตว่า

โหนดที่โหนด  $y$  โอนภาระของมันให้คือโหนด  $z$  ซึ่งขัดแย้งกับที่ว่าไม่มีเหตุการณ์ใดๆ เกิดขึ้นกับโหนด  $z$  ยกเว้นเหตุการณ์ min-transfer

พิจารณากรณีที่มีการลบกุญแจเกิดขึ้นที่โหนด  $w$  หลังจาก  $\hat{t}$  กรณีนี้สามารถถูกพิสูจน์ได้เช่นเดียวกับกรณีย่อย 1.2

ต่อไปเราจะสมมติว่าไม่มีการลบกุญแจเกิดขึ้นที่โหนด  $w$  หลังจากเวลา  $\hat{t}$  ที่เวลา  $\hat{t}$  เราได้ว่า  $2Min_{\hat{t}} \leq L_{\hat{t}}(w)$  เนื่องจากเหตุการณ์ min-transfer ครั้งที่  $(i + l + 1)$  เป็นเหตุการณ์ right-transfer นั่นคือโหนด  $w$  จะกลายเป็นโหนดที่มีภาระต่ำสุด ณ บางเวลาหลังจากเวลา  $\hat{t}$  กำหนดให้  $t^*$  เป็นเวลาที่โหนด  $w$  กลายเป็นโหนดที่มีภาระน้อยที่สุดหลังจากเวลา  $\hat{t}$  หลังจากเวลา  $\hat{t}$  ขั้นตอนปรับดุลภาระและการลบกุญแจจะไม่เกิดขึ้นกับโหนด  $w$  ดังนั้นภาระของโหนด  $w$  จะไม่ลดลง เราจะได้ว่า  $L_{\hat{t}}(w) \leq L_{t^*}(w)$  ยิ่งไปกว่านั้นเราจะได้ว่า  $L_{\hat{t}}(w) \leq Min'_{t_{i+l+1}}$  จากคุณสมบัติที่ไม่ลดลงของภาระต่ำสุด ดังนั้น  $2Min_{\hat{t}} \leq Min'_{t_{i+l+1}}$

พิจารณาสมการ (10) เราจะหารมันด้วย  $Min_{i+l+1}$  นั้นจะได้

$$\begin{aligned} \frac{L_{t_{i+l+1}}(z)}{Min_{i+l+1}} &\leq \frac{L_{t_i}(z) + (l+1)Min'_{t_{i+l+1}}}{Min_{i+l+1}} \\ &\leq \frac{L_{t_i}(z) + (l+1)Min'_{t_{i+l+1}}}{Min'_{t_{i+l+1}}} \\ &= (l+1) + \frac{L_{t_i}(z)}{Min'_{t_{i+l+1}}} \end{aligned}$$

จากคุณสมบัติที่ไม่เปลี่ยนแปลงเราจะได้ว่า  $L_{t_i}(z) \leq (\alpha + 2)Min_{t_i} \leq (\alpha + 2)Min_{\hat{t}}$  เมื่อ  $t_i < \hat{t}$  จากที่เราได้ว่า  $2Min_{\hat{t}} \leq Min'_{t_{i+l+1}}$  ทำให้ได้

$$\frac{L_{t_{i+l+1}}(z)}{Min_{i+l+1}} \leq (l+1) + \frac{(\alpha + 2)Min_{\hat{t}}}{2Min_{\hat{t}}}$$

เมื่อ  $\alpha > 2$  จะได้ว่าความสัมพันธ์ว่า  $\frac{(\alpha+2)}{2} \leq \alpha$  เมื่อแทนกลับเข้าไปในสมการ เราจะได้ว่า  $\frac{L_{t_{i+l+1}}(z)}{Min_{i+l+1}} \leq (\alpha + l + 1)$  ดังนั้นกรณีย่อยนี้เป็นจริง ■

จากบทตั้ง 15, 16 และ 17 ทำให้เราสามารถสรุปผลกระทบของเหตุการณ์ min-transfer ที่เกิดขึ้นกับโหนด  $z$  ใดๆ ได้ ในบทตั้ง 18 เป็นการสรุปคุณสมบัติหลังเหตุการณ์ min-transfer เกิดขึ้น เราจะละบทพิสูจน์เนื่องจากว่าสามารถพิสูจน์ได้ในลักษณะเดียวกับบทตั้ง 12

#### บทตั้ง 18

พิจารณาเฟส  $d$  ใดๆ สมมติว่า  $\alpha \geq \frac{3+\sqrt{33}}{2} \approx 4.373$  และ  $\beta \geq \frac{3(\alpha+2)}{\alpha}$  หลังจากเหตุการณ์ min-transfer ครั้งที่  $i$  ที่เกิดขึ้นกับโหนด  $z$  แล้วจะได้  $L_{t_i}(z) \leq (\alpha + 2)Min_{t_i}$

สุดท้ายเราสามารถที่จะพิสูจน์การรับประกันค่าอัตราส่วนความไม่สมดุลได้ โดยในแต่ละเฟสภาระของโหนดใดๆ สามารถถูกเปลี่ยนแปลงได้จากการใส่กุญแจ, การลบกุญแจ, nbr-transfer, nbr-share และ min-transfer จากบทตั้ง 9, 13, 14 และ 18 ทำให้เราสามารถสรุปขอบเขตบนของภาระของโหนดใดๆ ในเฟสใดๆ หลังจากเหตุการณ์เหล่านี้

#### ทฤษฎีบท 5

พิจารณาเฟสใดๆ สมมติว่า  $\alpha \geq \frac{3+\sqrt{33}}{2} \approx 4.373$  และ  $\beta \geq \frac{3(\alpha+2)}{\alpha}$  สำหรับโหนด  $u \in V$  ใดๆ หลังจากเหตุการณ์ใดๆ ณ เวลา  $t$  จะได้ว่า  $L_t(u) \leq (\alpha + 2)Min_t$  ดังนั้นอัตราส่วนความไม่สมดุลของขั้นตอนวิธีของเราในกรณีทั่วไปมีค่าเป็นค่าคงที่

### 3.4 การวิเคราะห์ค่าใช้จ่ายของขั้นตอนวิธีในกรณีทั่วไป

เราจะวิเคราะห์ค่าใช้จ่ายของขั้นตอนวิธีของเราในกรณีทั่วไป เช่นเดียวกับค่าใช้จ่ายของขั้นตอนวิธีในกรณีที่มีแต่การใส่กุญแจ เรากำหนดให้ค่าใช้จ่ายในการย้ายกุญแจหนึ่งดอกจากโหนด  $u$  ไปยังโหนด  $v$  มีค่าใช้จ่ายเป็นหนึ่งหน่วย ในขั้นตอนวิธีของเรา มีการดำเนินการ 4 อย่างได้แก่ การใส่กุญแจ, การลบกุญแจ, MINBALANCE และ SPLIT เราจะพิสูจน์ค่าใช้จ่ายแบบถัวเฉลี่ยโดยใช้วิธีการศักย์ และใช้ฟังก์ชันศักย์เช่นเดียวกับงานของ Ganesan *et al.*

### ทฤษฎีบท 6

ถ้า  $\alpha > 2(1 + \sqrt{3}) \approx 5.464$  แล้ว ค่าใช้จ่ายถัวเฉลี่ยของขั้นตอนวิธีในกรณีทั่วไปเป็นค่าคงที่

#### พิสูจน์

ฟังก์ชันศักย์ที่เราใช้คือ  $\Phi = \frac{c(\sum_{i=1}^n (L_t(N_i))^2)}{\bar{L}_t}$  เมื่อ  $\bar{L}_t$  เป็นภาระเฉลี่ยที่เวลา  $t$  และ  $N_i$  เป็นโหนดที่  $i$  ที่จัดการช่วง  $[R_{i-1}, R_i)$

เราจะแสดงว่าเมื่อมีการใส่กุญแจหรือลบกุญแจ พลังงานศักย์จะเพิ่มขึ้นไม่เกินค่าคงที่ค่าหนึ่งและเมื่อการดำเนินการ MINBALANCE หรือ SPLIT เกิดขึ้นพลังงานศักย์จะลดลงเพื่อจ่ายเป็นค่าใช้จ่ายของการดำเนินการ ทั้งหมดนี้ทำให้แสดงได้ว่าค่าใช้จ่ายถัวเฉลี่ยของการใส่กุญแจและลบกุญแจเป็นค่าคงที่

การพิสูจน์ค่าใช้จ่ายของการใส่กุญแจและการดำเนินการ MINBALANCE สามารถพิสูจน์ได้เช่นเดียวกับที่พิสูจน์ไปแล้วในทฤษฎีบท 3 ทั้งนี้ต้องมีการกำหนดค่า  $\alpha > 2(1 + \sqrt{3}) \approx 5.464$  ดังนั้นต่อไปเราจะพิสูจน์ค่าใช้จ่ายของการลบกุญแจและการดำเนินการ SPLIT

การลบกุญแจ เมื่อมีการลบกุญแจเกิดขึ้นที่โหนด  $N_j$  ที่เวลา  $t$  พลังงานศักย์จะเพิ่มขึ้นไม่เกิน

$$\begin{aligned} \Delta\Phi &\leq c \frac{(\sum_{i \in V} L_t(N_i)^2)}{(\bar{L}_t - \frac{1}{n})} - c \frac{(\sum_{i \in V} L_t(N_i)^2)}{\bar{L}_t} \\ &= c \frac{(\sum_{i \in V} L_t(N_i)^2) (\frac{1}{n})}{\bar{L}_t (\bar{L}_t - \frac{1}{n})}. \end{aligned}$$

เมื่อใช้ความจริงที่ว่า  $\bar{L}_t \geq \text{Min}_t$  แล้วนำมารวมกับคุณสมบัติที่ไม่เปลี่ยนแปลงเราจะได้อสมการ

$$L_t(u) \leq (\alpha + 2)\text{Min}_t \leq (\alpha + 2)\bar{L}_t$$

นำสมการที่ได้ไปแทนค่าในฟังก์ชันพลังงานศักย์ สำหรับโหนด  $u$  ใดๆ เมื่อเราใช้ความจริงที่ว่า  $\bar{L}_t - \frac{1}{n} \geq \frac{\bar{L}_t}{2}$  เมื่อ  $n \geq 2$  และ  $\bar{L}_t \geq 1$  เราจะได้ว่า

$$\Delta\Phi \leq c \frac{((\alpha + 2) \cdot \bar{L}_t)^2}{\bar{L}_t \cdot \left(\frac{\bar{L}_t}{2}\right)} \leq 2c(\alpha + 2)^2.$$

เนื่องจากค่าใช้จ่ายจริงของการลบนั้นมีค่าหนึ่งหน่วย ดังนั้นค่าใช้จ่ายแบบถัวเฉลี่ยของการลบทุกๆ จะเป็นค่าคงที่

Split พิจารณาภาระของโหนด  $N_j$  ที่เรียกการดำเนินการ SPLIT ที่เวลา  $t$  เมื่อภาระของมันมีค่าน้อยกว่า  $\frac{Max_t}{\beta}$  ขั้นตอนการปรับดุลภาระจะถูกเรียก หลังจากนั้น 2 กรณีที่สามารถเกิดขึ้นได้นั้นคือ ขั้นตอน SPLITNBR และขั้นตอน SPLITMAX

กรณีขั้นตอน SPLITNBR เกิดขึ้น กำหนดให้  $N_k$  แทนโหนดเพื่อนบ้านที่มีภาระน้อยที่สุดของ  $N_j$  เมื่อโหนด  $N_j$  เรียกขั้นตอน SPLITNBR โหนด  $N_k$  จะย้ายกุญแจของมันให้กับโหนด  $N_j$  เพื่อให้ภาระของทั้งสองโหนดสมดุล พลังงานศักย์จะลดลงดังนี้

$$\begin{aligned} \Delta\Phi &= \frac{c \left( L_t(N_j)^2 + L_t(N_k)^2 - 2 \left( \frac{L_t(N_j) + L_t(N_k)}{2} \right)^2 \right)}{\bar{L}_t} \\ &= c \frac{\left( \frac{L_t(N_j)^2}{2} + \frac{L_t(N_k)^2}{2} - (L_t(N_j)L_t(N_k)) \right)}{\bar{L}_t} \\ &= c \frac{(L_t(N_k) - L_t(N_j))^2}{2\bar{L}_t} \\ &= c \frac{(L_t(N_k) - L_t(N_j))}{2} \frac{(L_t(N_k) - L_t(N_j))}{\bar{L}_t}. \end{aligned}$$

กรณีนี้เกิดขึ้นเมื่อ  $L_t(N_k) > \frac{2Max_t}{\beta}$  และ  $L_t(N_j) < \frac{Max_t}{\beta}$  ดังนั้นเมื่อแทนค่า  $L_t(N_k)$  และ  $L_t(N_j)$  จะได้ว่า

$$\begin{aligned} \Delta\Phi &\geq c \frac{(L_t(N_k) - L_t(N_j))}{2} \frac{\left( \frac{2Max_t}{\beta} - \frac{Max_t}{\beta} \right)}{\bar{L}_t} \\ &= c \frac{(L_t(N_k) - L_t(N_j))}{2} \frac{\left( \frac{Max_t}{\beta} \right)}{\bar{L}_t} \end{aligned}$$

$$\geq c \frac{(L_t(N_k) - L_t(N_j))}{2} \left( \frac{1}{\beta} \right).$$

จำนวนของกุญแจที่ถูกย้ายเมื่อขั้นตอน SPLITNBR ถูกเรียกใช้คือ  $\frac{(L_t(N_k) - L_t(N_j))}{2}$  สำหรับค่า  $c > \beta$  ใดๆ ทำให้ได้ว่า  $\Delta\Phi \geq \frac{(L_t(N_k) - L_t(N_j))}{2}$  ดังนั้นพลังงานศักย์ที่ลดลงจะถูกจ่ายเพียงพอกับการย้ายข้อมูล

กรณีขั้นตอน SPLITMAX เกิดขึ้น โหนด  $N_j$  ย้ายภาระของมันทั้งหมดให้โหนดเพื่อนบ้าน  $N_k$  จากนั้นโหนด  $N_j$  จะไปแบ่งภาระของโหนดที่มีภาระมากที่สุด  $N_l$  ครั้งหนึ่ง พลังงานศักย์จะลดลงเป็น

$$\begin{aligned} \Delta\Phi &= c \frac{\left( L_t(N_j)^2 + L_t(N_k)^2 + L_t(N_l)^2 - 2 \left( \frac{L_t(N_l)}{2} \right)^2 - (L_t(N_k) + L_t(N_j))^2 \right)}{\bar{L}_t} \\ &= c \frac{\left( \frac{L_t(N_l)^2}{2} - 2L_t(N_k)L_t(N_j) \right)}{\bar{L}_t} \\ &= \frac{c}{\bar{L}_t} L_t(N_l)^2 \left( \frac{1}{2} - \frac{2L_t(N_k)L_t(N_j)}{L_t(N_l)^2} \right). \end{aligned}$$

จากเงื่อนไขของขั้นตอนวิธี ขั้นตอน SPLITMAX จะเกิดขึ้นเมื่อ  $L_t(N_k) \leq \frac{2Max_t}{\beta}$  และ  $L_t(N_j) \leq \frac{Max_t}{\beta}$  ดังนั้นจะได้ว่า

$$\begin{aligned} \Delta\Phi &\geq cL_t(N_l) \left( \frac{1}{2} - \frac{2 \frac{2Max_t}{\beta} \cdot \frac{Max_t}{\beta}}{L_t(N_l)^2} \right) \\ &\geq c \cdot L_t(N_l) \left( \frac{1}{2} - \frac{4}{\beta^2} \right). \end{aligned}$$

เมื่อขั้นตอน SPLITMAX ถูกเรียก จำนวนกุญแจที่ถูกย้ายจะเป็น  $\frac{L_t(N_l)}{2} + L_t(N_j) \leq L_t(N_l)$  สำหรับค่า  $c > \left( \frac{2\beta^2}{\beta^2 - 8} \right)$  ใดๆ จะได้ว่า  $\Delta\Phi \geq L_t(N_l)$  ดังนั้นพลังงานศักย์ที่ลดลงจะถูกจ่ายเพียงพอให้การย้ายของกุญแจ

ดังนั้นค่าใช้จ่ายถ่วงเฉลี่ยของขั้นตอนวิธีของเราจะเป็นค่าคงที่ ■

เนื่องจากเราต้องการให้อัตราส่วนความไม่สมดุลเป็นค่าคงที่และต้องการให้ค่าใช้จ่ายของ

ขั้นตอนวิธีเป็นค่าคงที่ ดังนั้นเราต้องกำหนดค่า  $\alpha > 2(1 + \sqrt{3}) \approx 5.464$  ทำให้ได้ทฤษฎีบท

### ทฤษฎีบท 7

ถ้า  $\alpha > 2(1 + \sqrt{3}) \approx 5.464$  แล้วเราจะได้ว่าอัตราส่วนความไม่สมดุลมีค่าเป็น  $\alpha + 2$  และค่าใช้จ่ายเฉลี่ยของขั้นตอนวิธีของเราในกรณีทั่วไปเป็นค่าคงที่

### 3.5 การวิเคราะห์ค่าใช้จ่ายในเครือข่ายจริง

เนื่องจากในขั้นตอนวิธีที่ได้เสนอไปมีการเรียกใช้ข้อมูลแบบครอบคลุม (Global information) นั่นคือข้อมูลภาระที่ต่ำที่สุดและภาระที่สูงที่สุด ในหัวข้อย่อนี้เราจะอธิบายถึงวิธีการจัดการข้อมูลแบบครอบคลุมเหล่านี้ในเครือข่ายจริง เราใช้วิธีการจัดการเช่นเดียวกันกับงานของ Ganesan *et al.* (2004) นอกจากนี้เราจะอธิบายและเปรียบเทียบถึงงานของ Karger and Ruhl (2004) เมื่อพิจารณาในเครือข่ายจริง

เนื่องจากในเครือข่ายเพียร์ทูเพียร์ไม่มีเครื่องบริการที่คอยจัดการข้อมูลแบบครอบคลุม ดังนั้นเมื่อพิจารณาโหนด  $y$  ใดๆ ถ้าโหนด  $x$  ใดๆ ต้องการข้อมูลของโหนด  $y$  โหนด  $x$  จะต้องส่งข้อความร้องขอข้อมูลไปยังโหนด  $y$  เพื่อขอข้อมูล เมื่อพิจารณาค่าใช้จ่ายในการดำเนินการนอกจากค่าใช้จ่ายในการย้ายข้อมูลแล้วยังมีค่าใช้จ่ายอย่างอื่นที่ต้องพิจารณานั้นคือ ค่าใช้จ่ายในการติดต่อสื่อสาร โดยค่าใช้จ่ายในการติดต่อสื่อสารจะนิยามเป็นจำนวนของข้อความที่จำเป็นเพื่อที่จะทำให้การดำเนินการสำเร็จ

เนื่องจากเราจะจัดการข้อมูลแบบครอบคลุมเช่นเดียวกับงานของ Ganesan *et al.* (2004) เราจะอธิบายว่าในงานของ Ganesan *et al.* (2004) นั้นนำไปใช้ในเครือข่ายจริงได้อย่างไร ในการใช้งานจริงนั้น Ganesan *et al.* (2004) จะสร้าง Skip graphs (Aspnes and Shah (2003)) เพื่อเก็บข้อมูลแบบครอบคลุม โดย Skip graphs นั้นรองรับการดำเนินการค้นหา, การดำเนินการเพิ่มโหนด และการดำเนินการลบโหนดโดยใช้ข้อความจำนวน  $O(\log n)$  ด้วยความน่าจะเป็นที่สูง อีกทั้งโหนดข้างเคียงถ้าจะติดต่อกันใช้ข้อความเพียง  $O(1)$  Ganesan *et al.* (2004) ใช้ skip graphs 2 อัน สำหรับอันแรกโหนดจะถูกเรียงต่อกันด้วยค่ากุญแจที่น้อยที่สุดในช่วงที่โหนดนั้นดูแลอยู่ สำหรับอันที่สองนั้นโหนดจะถูกเรียง

ด้วยภาระของมัน ดังนั้นข้อมูลแบบครอบคลุมจะสามารถค้นหาได้โดยใช้  $O(\log n)$  ข้อความและเมื่อขอบเขตเปลี่ยนค่าใช้จ่ายก็จะใช้ข้อความไม่เกิน  $O(\log n)$  สังเกตได้ว่าค่าใช้จ่ายในการดำเนินการเหล่านี้มีค่ามากกว่าค่าคงที่ซึ่งมากกว่าค่าใช้จ่ายที่ใช้ในการย้ายข้อมูล แต่โดยทั่วไปแล้วในการหาโหนดที่รับผิดชอบกุญแจจะต้องใช้ข้อความ  $O(\log n)$  เช่นกัน ดังนั้นค่าใช้จ่ายเหล่านี้สามารถถูกลบไปกับค่าใช้จ่ายในการค้นหาได้

ในงานของ Karger and Ruhl (2004) นั้นเขาได้พิสูจน์ว่าค่าใช้จ่ายถูกลบในการเพิ่มกุญแจหรือลบกุญแจเป็นค่าคงที่ อย่างไรก็ตามพวกเขาไม่ได้พิจารณาถึงค่าใช้จ่ายในการหาโหนดที่ต้องการในระบบ ดังนั้นถ้าพวกเขาไม่เก็บข้อมูลสารบบส่วนกลาง (Central directory) ของโหนด พวกเขาก็ต้องใช้โครงสร้างข้อมูลในเครือข่ายเพียร์ทูเพียร์ในการค้นหาข้อมูล ดังนั้นในการดำเนินการต่างๆ จะใช้จำนวนข้อความเป็น  $O(\log n)$  เช่นกัน

## สรุปและข้อเสนอแนะ

ในวิทยานิพนธ์ฉบับนี้มีงานวิจัยที่เกี่ยวข้องกับการส่งข้อมูลแบบกระแสน์เครือข่ายเพียร์ทูเพียร์สองเรื่อง ได้แก่ การวางแผนการส่งข้อมูลแบบกระแสน์เครือข่ายเพียร์ทูเพียร์ โดยมีเงื่อนไขว่าแต่ละเครื่องในเครือข่ายมีความสามารถในการส่งต่อข้อมูลได้แตกต่างกัน และการออกแบบวิธีการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์เพื่อรองรับการค้นหาแบบช่วง ซึ่งสรุปผลได้ดังนี้

### การวางแผนการส่งข้อมูลแบบกระแสน์เครือข่ายเพียร์ทูเพียร์

ในงานวิจัยนี้เราสนใจการวางแผนการส่งข้อมูลแบบกระแสน์ที่มีความล่าช้าจากเครื่องต้นทางต่ำที่สุด เราสนใจกรณีพิเศษกรณีหนึ่งของ MDM problem ที่มีเงื่อนไขว่าแบนด์วิดท์ของเส้นเชื่อมทุกเส้นมีค่าเป็นจำนวนเท่าของแบนด์วิดท์ที่ต้องการ จากเงื่อนไขของแบนด์วิดท์ข้างต้น เราพิสูจน์ได้ว่ามีแผนการส่งข้อมูลที่ดีที่สุดที่อยู่ในรูปโครงสร้างต้นไม้ เราจึงลดรูปปัญหาไปเป็นปัญหาการสร้างต้นไม้ทอดข้ามที่มีเส้นผ่านกลางที่ต่ำที่สุดและดิกรีมีขอบเขตในกรณีที่ดีกรีไม่เท่ากันและเราได้เสนอขั้นตอนวิธีที่ทำงานในเวลาเป็นฟังก์ชันพหุนามสำหรับแก้ปัญหานี้ ขั้นตอนวิธีที่นำเสนอรับประกันว่าเส้นผ่านกลางของต้นไม้ที่ได้จากขั้นตอนวิธีมีความยาวไม่เกิน  $O(\sqrt{\log n})$  เท่าของเส้นผ่านกลางของต้นไม้ที่เป็นคำตอบที่ดีที่สุด เมื่อนำขั้นตอนวิธีของเราไปใช้ในการส่งข้อมูลแบบกระแสน์ จะได้แผนการส่งข้อมูลที่มีความล่าช้าไม่เกิน  $O(\sqrt{\log n})$  เท่าเมื่อเทียบกับแผนการส่งข้อมูลที่ดีที่สุด เราสามารถนำเอาขั้นตอนวิธีในงานวิจัยส่วนแรกไปประยุกต์ใช้กับโปรแกรมประยุกต์ที่ส่งข้อมูลแบบกระแสน์ไปยังผู้ใช้งานจำนวนมากได้เช่น โปรแกรมประยุกต์สำหรับถ่ายทอดสดเหตุการณ์ต่างๆ ทั้งนี้เครื่องคอมพิวเตอร์ของผู้ใช้งานนั้นไม่จำเป็นต้องมีความสามารถในการส่งต่อข้อมูลแบบกระแสน์ได้เท่ากัน ทำให้โปรแกรมประยุกต์มีความยืดหยุ่นสำหรับเครื่องที่จะนำมาใช้งาน อีกทั้งเราได้พิสูจน์ว่าความล่าช้าเมื่อนำเอาขั้นตอนวิธีในงานวิจัยส่วนแรกไปใช้จะช้ากว่าแผนการส่งข้อมูลที่ดีที่สุดไม่เกิน  $O(\sqrt{\log n})$  เท่าซึ่งเราคาดว่าเมื่อนำขั้นตอนวิธีที่เราได้นำเสนอไปใช้งานจริงจะมีประสิทธิภาพที่ดี

### การออกแบบวิธีการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์เพื่อรองรับการค้นหาแบบช่วง

ในงานวิจัยนี้เราได้นำเสนอขั้นตอนวิธีสำหรับแก้ปัญหการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์ที่รองรับการค้นหาแบบช่วงที่ทำงานในเวลาเป็นฟังก์ชันพหุนาม ขั้นตอนวิธีที่เราเสนอไม่เป็นขั้นตอนวิธีแบบเรียกตัวเองซึ่งต่างจากขั้นตอนวิธีของ Ganesan *et al.* (2004) เราได้สนใจกรณีที่เกิดขึ้นใน

การใช้งานจริงสองกรณีได้แก่ กรณีที่มีแต่การใส่กุญแจและกรณีที่มีทั้งการใส่และลบกุญแจ สำหรับกรณีที่มีแต่การใส่กุญแจเราได้เสนอการดำเนินการ MINBALANCE เราได้พิสูจน์อัตราส่วนความไม่สมดุลว่ามีค่าไม่เกิน 7.464 และพิสูจน์ว่าค่าใช้จ่ายเฉลี่ยของขั้นตอนวิธีเป็นค่าคงที่ สำหรับกรณีที่มีทั้งการใส่และลบกุญแจเราใช้การดำเนินการ MINBALANCE เมื่อมีการใส่กุญแจ เราเสนอการดำเนินการ SPLIT เมื่อมีการลบกุญแจเกิดขึ้น เราได้พิสูจน์ว่าอัตราส่วนความไม่สมดุลในกรณีนี้มีค่าไม่เกิน 7.464 เช่นกันและพิสูจน์ว่าค่าใช้จ่ายเฉลี่ยของขั้นตอนวิธีเป็นค่าคงที่ ถึงแม้ว่าอัตราส่วนความไม่สมดุลของขั้นตอนวิธีที่เรานำเสนอมีค่ามากกว่าขั้นตอนวิธีของ Ganesan *et al.* (2004) ซึ่งรับประกันอัตราส่วนความไม่สมดุลมีค่าอย่างต่ำเป็น 4.237 แต่ขั้นตอนวิธีของเรามีความง่ายในการนำไปใช้จริงมากกว่าของ Ganesan *et al.* (2004) เนื่องจากไม่ใช่ขั้นตอนวิธีแบบเรียกตัวเอง เราสามารถนำเอาขั้นตอนวิธีการปรับคุณภาพที่เสนอในงานวิจัยส่วนที่สองไปประยุกต์ใช้กับโปรแกรมประยุกต์ที่ต้องการการปรับคุณภาพพร้อมกับรองรับการค้นหาข้อมูลแบบช่วงตัวอย่างเช่น ในโปรแกรมประยุกต์พีทูพีทีวี (P2PTV) ผู้ใช้งานต้องการค้นหารายการที่มีการจัดเรตการชมอายุช่วงต่างๆ หรือในระบบประมูลสินค้าบนเครือข่ายเพียร์ทูเพียร์มีการค้นหาโทรศัพท์มือถือที่อยู่ในช่วงราคา 8000 บาทถึง 20000 บาท โดยขั้นตอนวิธีที่งานวิจัยนี้เสนอนั้นมีความง่ายในการนำไปใช้งานและยังมีอัตราส่วนความไม่สมดุลที่ต่ำอีกด้วย

### ปัญหาเปิดที่สามารถทำต่อจากงานวิจัยนี้ได้

ในส่วนนี้เราจะได้เสนอปัญหาเปิด (Open problem) ที่น่าสนใจของปัญหาการวางแผนการส่งข้อมูลแบบกระแสนเครือข่ายเพียร์ทูเพียร์ และการออกแบบวิธีการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์เพื่อรองรับการค้นหาแบบช่วง

ปัญหาเปิดของการวางแผนการส่งข้อมูลแบบกระแสนเครือข่ายเพียร์ทูเพียร์

1. การออกแบบแผนการส่งข้อมูลแบบกระจาย (Distributed version) เนื่องจากขั้นตอนวิธีที่ได้้นำเสนอไปเป็นขั้นตอนวิธีแบบรวมศูนย์
2. การออกแบบแผนการส่งข้อมูลที่มีการรองรับโหนดเข้าและออกในระบบเพื่อให้สอดคล้องกับเครือข่ายเพียร์ทูเพียร์ในโลกความจริง
3. การลดเงื่อนไขแบนด์วิธของเส้นเชื่อม

4. การออกแบบแผนการส่งข้อมูลที่รองรับ multi-channel streaming เนื่องจากขั้นตอนวิธีที่ได้นำเสนอไปเป็น single-channel streaming

ปัญหาเปิดของการออกแบบวิธีการปรับคุณภาพในเครือข่ายเพียร์ทูเพียร์เพื่อรองรับการค้นหาแบบช่วง

1. การจัดการข้อมูลแบบครอบคลุม หากข้อมูลแบบครอบคลุมที่ได้มีความผิดพลาด ความถูกต้องต่ำ หรือเป็นข้อมูลแบบครอบคลุมสำหรับบางช่วง จะจัดการอย่างไร หรือมีวิธีที่ไม่ต้องใช้ข้อมูลแบบครอบคลุมหรือไม่
2. การวิเคราะห์ต่อในกรณีที่มีโหนดเข้าและออกเนื่องจากในวิทยานิพนธ์นี้ไม่ได้วิเคราะห์กรณีที่มีโหนดเข้าและออกจากระบบ
3. การออกแบบวิธีการปรับคุณภาพที่รองรับการค้นหาแบบช่วงในหลายมิติ

## เอกสารและสิ่งอ้างอิง

- Aspnes, J. and G. Shah. 2003. Skip graphs, pp. 384-393. In **Proceedings of the 14<sup>th</sup> annual ACM-SIAM symposium on Discrete algorithms**. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Bharambe, A. R., M. Agrawal and S. Seshan. 2004. Mercury: supporting scalable multi-attribute range queries, pp. 353-366. In **Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications**. ACM, New York, NY, USA.
- Castro, M., P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron and A. Singh. 2003. Splitstream: high-bandwidth multicast in cooperative environments, pp. 298-313. In **Proceedings of the 19<sup>th</sup> ACM symposium on Operating systems principles**. ACM, New York, NY, USA.
- Christofides, N. 1976. **Worst-case analysis of a new heuristic for the travelling salesman problem**. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University. 10.
- Deshpande, H., M. Bawa and H. Garcia-Molina. 2002. **Streaming live media over peers**. Technical Report 2002-21, Stanford InfoLab. 14.
- Elkin, M. and G. Kortsarz. 2006. Sublogarithmic approximation for telephone multicast. **Journal of Computer and System Sciences** 72 (4): 648 - 659.
- Ford, D. R. and D. R. Fulkerson. 1962. **Flows in Networks**. Princeton University Press, Princeton, NJ, USA.
- Frederickson, G. N., M. S. Hecht and C. E. Kim. 1978. Approximation algorithms for some routing problems. **SIAM Journal on Computing** 7(2): 178-193.

- Ganesan, P., M. Bawa and H. Garcia-Molina. 2004. Online balancing of range-partitioned data with applications to peer-to-peer systems, pp. 444-455. In **Proceedings of the 30<sup>th</sup> international conference on Very large data bases - Volume 30**. VLDB Endowment, Toronto, Canada.
- Harren, M., J. M. Hellerstein, R. Huebsch, B. T. Loo, S. Shenker and I. Stoica. 2002. Complex queries in dht-based peer-to-peer networks, pp. 242-259. In **Revised Papers from the 1<sup>st</sup> International Workshop on Peer-to-Peer Systems**. Springer-Verlag, London, UK.
- Harvey, N. J. A., M. B. Jones, S. Saroiu, M. Theimer and A. Wolman. 2003. Skipnet: a scalable overlay network with practical locality properties, pp. 9. In **Proceedings of the 4<sup>th</sup> conference on USENIX Symposium on Internet Technologies and Systems - Volume 4**. USENIX Association, Berkeley, CA, USA.
- Huang, F., B. Ravindran and A. V. Kumar. 2009. An approximation algorithm for minimum-delay peer-to-peer streaming, pp. 71-80. In **IEEE 9<sup>th</sup> International Conference on Peer-to-Peer Computing**. IEEE Computer Society, Washington, DC, USA.
- Jagadish, H. V., B. C. Ooi and Q. H. Vu. 2005. Baton: a balanced tree structure for peer-to-peer networks, pp. 661-672. In **Proceedings of the 31<sup>th</sup> international conference on Very large data bases**. VLDB Endowment, Toronto, Canada.
- Jiang, X., Y. Dong, D. Xu and B. Bhargava. 2003. Gnustream: A p2p media streaming system prototype, pp. 325-328. In **Proceedings of the International Conference on Multimedia and Expo**. IEEE Computer Society, Washington, DC, USA.
- Joung, Y.-J. 2008. Approaching neighbor proximity and load balance for range query in p2p networks. **Computer Networks** 52: 1451-1472.
- Karger, D. R. and M. Ruhl. 2004. Simple efficient load balancing algorithms for peer-to-peer systems, pp. 36-43. In **Proceedings of the 16<sup>th</sup> annual ACM symposium on Parallelism in algorithms and architectures**. ACM, New York, NY, USA.

- Könemann, J., A. Levin and A. Sinha. 2005. Approximating the degree-bounded minimum diameter spanning tree problem. **Algorithmica** 41(2): 117-129.
- Liao, X., H. Jin, Y. Liu and L. M. Ni. 2007. Scalable live streaming service based on interoverlay optimization, pp. 1663-1674. **IEEE Transaction Parallel Distributed System** IEEE Computer Society, Washington, DC, USA.
- Maymounkov, P. and D. Mazières. 2002. Kademlia: A peer-to-peer information system based on the xor metric, pp. 53-65. In the **1<sup>st</sup> International Workshop on Peer-to-Peer Systems**. Springer-Verlag, London, UK.
- Nguyen, T., D. Tran and S. Cheung. 2005. Efficient p2p data dissemination in a homogeneous capacity network using structured mesh, pp. 73 - 77. In the **1<sup>st</sup> International Conference on Multimedia Services Access Networks**. IEEE Computer Society, Washington, DC, USA.
- Padmanabhan, V. N., H. J. Wang, P. A. Chou and K. Sripanidkulchai. 2002. Distributing streaming media content using cooperative networking, pp. 177-186. In the **12<sup>th</sup> International Workshop on Network and Operating System Support for Digital Audio and Video**. ACM, New York, NY, USA.
- Pugh, W. 1990. Skip lists: a probabilistic alternative to balanced trees. **Communications of the ACM** 33(6): 668-676.
- Ramabhadran, S., S. Ratnasamy, J. M. Hellerstein and S. Shenker. 2004. Brief announcement: prefix hash tree, pp. 368-368. In **Proceedings of the 23<sup>th</sup> annual ACM symposium on Principles of distributed computing**. ACM, New York, NY, USA.
- Ratnasamy, S., P. Francis, M. Handley, R. Karp and S. Shenker. 2001. A scalable content-addressable network, pp. 161-172. In **Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications**. ACM, New York, NY, USA.

- Ravi, R. 1994. Rapid rumor ramification: approximating the minimum broadcast time, pp. 202-213. In **Proceedings of the 35<sup>th</sup> Annual Symposium on Foundations of Computer Science**. IEEE Computer Society, Washington, DC, USA.
- Ren, D., Y.-T. Li and S.-H. Chan. 2008. On reducing mesh delay for peer-to-peer live streaming, pp. 1058-1066. In **the 27<sup>th</sup> IEEE Conference on Computer Communication**. IEEE Computer Society, Washington, DC, USA.
- Risson, J. and T. Moors. 2006. Survey of research towards robust peer-to-peer networks: Search methods. **Computer Networks** 50: 3485 - 3521.
- Rowstron, A. I. T. and P. Druschel. 2001. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems, pp. 329-350. In **Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg**. Springer-Verlag, London, UK.
- Shi, S. Y., J. S. Turner and M. Waldvogel. 2001. Dimensioning server access bandwidth and multicast routing in overlay networks, pp. 83-92. In **the 11<sup>th</sup> International Workshop on Network and Operating System Support for Digital Audio and Video**. ACM, New York, NY, USA.
- Small, T., B. Li, S. Member and B. Liang. 2007. Outreach: Peer-to-peer topology construction towards minimized server bandwidth costs, pp. 35-45. In **IEEE Journal on Selected Areas in Communications, Special Issue on Peer-to-Peer Communications and Applications, First Quarter**. IEEE Computer Society, Washington, DC, USA.
- Stoica, I., R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek and H. Balakrishnan. 2003. Chord: a scalable peer-to-peer lookup protocol for internet applications. **IEEE/ACM Transaction of Networking** 11: 17-32.
- Tran, D. A., K. A. Hua and T. T. Do. 2006. A peer-to-peer architecture for media streaming. **IEEE Journal on Selected Areas in Communications**, 22: 121-133.

- Venkataraman, V., K. Yoshida and P. Francis. 2006. Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast, pp. 2-11. In **Proceedings of the 14<sup>th</sup> IEEE International Conference on Network Protocols**. IEEE Computer Society, Washington, DC, USA.
- Yang, X. and Y. Hu. 2006. A scalable index architecture for supporting multidimensional range queries in peer-to-peer networks, pp. 1-10. In **Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing**. IEEE Computer Society, Washington, DC, USA.
- Zhang, H., H. Jin and Q. Zhang. 2009. Yarqs: Yet another range queries schema in dht based p2p network, pp. 51-56. In **Proceedings of the 9<sup>th</sup> IEEE International Conference on Computer and Information Technology**. IEEE Computer Society, Washington, DC, USA.
- Zhang, X., J. Liu, B. Li and Y.-S. Yum. 2005. Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming, pp. 2102-2111. In **the 24<sup>th</sup> IEEE Conference on Computer Communication**. IEEE Computer Society, Washington, DC, USA.
- Zhao, B. Y., L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph and J. D. Kubiatowicz. 2004. Tapestry: A resilient global-scale overlay for service deployment. **IEEE Journal on Selected Areas in Communications** 22: 41-53.
- Zheng, C., G. Shen, S. Li and S. Shenker. 2006. Distributed segment tree: Support of range query and cover query over dht. In **Electronic publications of the 5<sup>th</sup> International Workshop on Peer-to-Peer Systems**. Santa Barbara, CA, USA.
- Zhou, Y., D. M. Chiu and J. Lui. 2007. A simple model for analyzing p2p streaming protocols, pp. 226-235. In **IEEE International Conference on Network Protocols**. IEEE Computer Society, Washington, DC, USA.

## ประวัติการศึกษาและการทำงาน

|                      |                                                                                                      |
|----------------------|------------------------------------------------------------------------------------------------------|
| ชื่อ                 | นายจักริน ชวชาติ                                                                                     |
| เกิดวันที่           | 12 มกราคม 2524                                                                                       |
| สถานที่เกิด          | อำเภอเมือง จังหวัดเชียงใหม่                                                                          |
| ประวัติการศึกษา      | วท.บ. (วิทยาการคอมพิวเตอร์) มหาวิทยาลัยเชียงใหม่<br>วท.ม. (วิทยาการคอมพิวเตอร์) มหาวิทยาลัยเชียงใหม่ |
| ตำแหน่งปัจจุบัน      | -                                                                                                    |
| สถานที่ทำงานปัจจุบัน | -                                                                                                    |
| ทุนการศึกษาที่ได้รับ | ได้รับทุนพัฒนาอาจารย์สาขาขาดแคลน<br>จาก มหาวิทยาลัยเชียงใหม่ (พ.ศ.2546)                              |