



การปรับปรุงอัตราการไหลในเครือข่ายด้วยโปรแกรมแบบ GUI

Flow Improvement in Network with GUI

นายอภิวัฒน์ กันทวงศ์ รหัส 47380359

ห้องเรียน คณะวิศวกรรมศาสตร์  
วันที่รับ..... 25 พ.ค. 2553  
เลขทะเบียน..... 14๓ ๗๗๗  
เลขประจำตัว..... ๘/๕.  
เลขจดหมายประจำตัว ๐๒๖๖ ๗ ๒๕๕๓

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

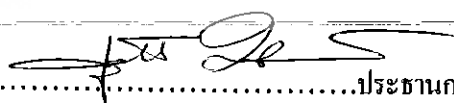
ปีการศึกษา 2551




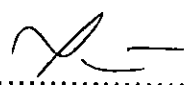
## ใบรับรองโครงการงานวิศวกรรม

หัวข้อโครงการ	การปรับปรุงอัตราการใช้ไฟในเครือข่ายด้วย โปรแกรมแบบ GUI
ผู้ดำเนินโครงการ	นายอภิวัฒน์ กันทวงศ์ รหัส 47380359
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ชนิต มาลากร
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2551

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตาม  
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์  
คณะกรรมการสอบ โครงการงานวิศวกรรม

  
.....ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.ชนิต มาลากร)

  
.....กรรมการ  
(ดร.พนมขวัญ ริษะมงคล)

  
.....กรรมการ  
(ดร.ไพศาล มุณีสว่าง)

## กิตติกรรมประกาศ

โครงการนี้จะสำเร็จลงไม่ได้หากมิได้รับความอนุเคราะห์จากผู้ช่วยศาสตราจารย์ ดร.ชนิด มาลากร ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการนี้ ผู้ช่วยศาสตราจารย์ ดร.ชนิด มาลากร ได้ให้ความช่วยเหลือและคอยให้คำปรึกษาด้านทฤษฎีกราฟ และอธิบายถึงการใช้งานโปรแกรม Matlab 7 เนื่องจากเป็นโปรแกรมที่ข้าพเจ้าไม่มีความรู้เลย จึงขอขอบพระคุณอาจารย์เป็นอย่างสูง

ขอกราบขอบพระคุณ คร.พนมขวัญ ธิยะมงคลและคร.ไพศาล มณีสว่าง ผู้ทรงคุณวุฒิที่ตลอดเวลาเป็นกรรมการในการสอบและกรรมการตรวจสอบข้อบกพร่องต่างๆ ของโครงการฉบับนี้

ขอขอบคุณห้องสมุด มหาวิทยาลัยนเรศวร ที่เป็นแหล่งค้นคว้าหาข้อมูลหลัก ในการทำโครงการครั้งนี้

สุดท้ายนี้ กราบขอบพระคุณบุพการีผู้ให้ทุกสิ่งทุกอย่างกับข้าพเจ้า อีกทั้งให้กำลังใจมาตลอด



ผู้จัดทำโครงการ

นาย อภิวัฒน์ กันทองค์

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 .....	1
1.1 หลักการและเหตุผล.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบข่ายของโครงการ.....	2
1.4 ขั้นตอนการดำเนินงาน.....	3
1.5 งบประมาณ.....	3
บทที่ 2.....	4
2.1 กราฟและไดกราฟ.....	5
2.2 การวิเคราะห์หาอัตราการไหลสูงสุดในเครือข่ายเชิงเดียว.....	8
2.3 เมทริกซ์และกราฟ.....	31
บทที่ 3.....	35
3.1 ขั้นตอนวิธีของ Ford-Fulkerson.....	35
3.2 ขั้นตอนวิธีของ Dinic.....	41
3.3 ขั้นตอนวิธีของ Augmenting Flow.....	46
บทที่ 4 .....	50
4.1 การติดตั้งโปรแกรม.....	50
4.2 การใช้งานโปรแกรม.....	53
4.3 ตัวอย่างการใช้งานโปรแกรม.....	63
บทที่ 5 .....	65
5.1 การทดสอบขั้นตอนวิธีในการหาอัตราการไหลสูงสุดในเครือข่าย.....	65
5.2 ปัญหาและอุปสรรค.....	65

## สารบัญ (ต่อ)

5.3 ข้อเสนอแนะและแนวทางการพัฒนา.....	65
เอกสารอ้างอิง.....	66
ประวัติผู้แต่ง.....	67



## สารบัญตาราง

	หน้า
ตารางที่ 1 แสดงการปรับอัตราการใช้และค่าความจุตกค้างบนส่วนโค้งตามวิถี.....	12
ตารางที่ 2 แสดงการปรับอัตราการใช้และค่าความจุตกค้างบนส่วนโค้งย้อนวิถี.....	13



## สารบัญรูป

หน้า

รูปที่ 2.1 แสดงตัวอย่างของการแทนโรงงานน้ำมันด้วยเครือข่าย.....	4
รูปที่ 2.2 แสดง “หัว” และ “หาง” ของส่วนโค้ง.....	5
รูปที่ 2.3 แสดงตัวอย่างของนิยามต่าง ๆ ที่เกี่ยวข้องกับกราฟ.....	6
รูปที่ 2.4 แสดงตัวอย่างของนิยามต่าง ๆ ที่เกี่ยวข้องกับไดกราฟ.....	6
รูปที่ 2.5 แสดงตัวอย่างของแนวเดินประเภทต่าง ๆ ในไดกราฟ.....	8
รูปที่ 2.6 แสดงถึงค่าอัตราการไหล และค่าความจุบนส่วนโค้ง.....	8
รูปที่ 2.7 แสดงตัวอย่างของเครือข่ายเชิงเดียว.....	10
รูปที่ 2.8 แสดงตัวอย่างของเครือข่ายเชิงเดียวที่มีอัตราการไหลเสมือน.....	10
รูปที่ 2.9 แสดงตัวอย่างของเครือข่ายตกค้าง.....	11
รูปที่ 2.10 แสดงเครือข่ายเชิงเดียวของขั้นตอนวิธีของ Ford-Fulkerson.....	15
รูปที่ 2.11 แสดงเครือข่ายเชิงเดียวที่มีอัตราการไหลเสมือนของขั้นตอนวิธีของ Ford-Fulkerson.....	15
รูปที่ 2.12 แสดงเครือข่ายตกค้างของขั้นตอนวิธีของ Ford-Fulkerson.....	15
รูปที่ 2.13 แสดงเครือข่ายที่มีการปรับครั้งที่ 1 ของขั้นตอนวิธีของ Ford-Fulkerson.....	16
รูปที่ 2.14 แสดงเครือข่ายที่มีการปรับครั้งที่ 2 ของขั้นตอนวิธีของ Ford-Fulkerson.....	16
รูปที่ 2.15 แสดงเครือข่ายที่มีการปรับครั้งที่ 3 ของขั้นตอนวิธีของ Ford-Fulkerson.....	17
รูปที่ 2.16 แสดงเครือข่ายที่มีการปรับครั้งที่ 4 ของขั้นตอนวิธีของ Ford-Fulkerson.....	17
รูปที่ 2.17 แสดงเครือข่ายเชิงเดียวของขั้นตอนวิธีของ Dinic.....	19
รูปที่ 2.18 แสดงเครือข่ายเชิงเดียวที่มีอัตราการไหลเสมือนของขั้นตอนวิธีของ Dinic.....	19
รูปที่ 2.19 แสดงเครือข่ายตกค้างของขั้นตอนวิธีของ Dinic.....	19
รูปที่ 2.20 เครือข่ายระดับใช้ในการหาวิถีที่สั้นที่สุดครั้งที่ 1.....	20
รูปที่ 2.21 แสดงเครือข่ายที่มีการปรับครั้งที่ 1 ของขั้นตอนวิธีของ Dinic.....	20
รูปที่ 2.22 แสดงเครือข่ายที่มีการขีดขวางอัตราการไหลสุทธิครั้งที่ 1 ของขั้นตอนวิธีของ Dinic.....	21
รูปที่ 2.23 เครือข่ายระดับใช้ในการหาวิถีที่สั้นที่สุดครั้งที่ 2.....	21
รูปที่ 2.24 แสดงเครือข่ายที่มีการปรับครั้งที่ 2 ของขั้นตอนวิธีของ Dinic.....	22
รูปที่ 2.25 แสดงเครือข่ายที่มีการขีดขวางอัตราการไหลสุทธิครั้งที่ 2 ของขั้นตอนวิธีของ Dinic.....	22
รูปที่ 2.26 เครือข่ายระดับใช้ในการหาวิถีที่สั้นที่สุดครั้งที่ 3.....	23
รูปที่ 2.27 แสดงเครือข่ายที่มีการปรับครั้งที่ 3 ของขั้นตอนวิธีของ Dinic.....	23
รูปที่ 2.28 แสดงเครือข่ายที่มีการขีดขวางอัตราการไหลสุทธิครั้งที่ 3 ของขั้นตอนวิธีของ Dinic.....	24
รูปที่ 2.29 เครือข่ายระดับใช้ในการหาวิถีที่สั้นที่สุดครั้งที่ 3.....	24
รูปที่ 2.30 แสดงเครือข่ายที่มีการปรับครั้งที่ 4 ของขั้นตอนวิธีของ Dinic.....	25

## สารบัญรูป(ต่อ)

รูปที่ 2.31 แสดงเครือข่ายที่มีการขีดขวางอัตราการไหลสุทธิครั้งที่ 4 ของขั้นตอนวิธีของ Dinic....	25
รูปที่ 2.32 แสดงเครือข่ายที่มีการปรับอัตราการไหลสุทธิเสร็จสิ้นแล้ว ของขั้นตอนวิธีของ Dinic..	25
รูปที่ 2.33 แสดงตัวอย่างของเครือข่ายเชิงเดียวของขั้นตอนวิธี Augmenting Flow.....	28
รูปที่ 2.34 แสดงผลการปรับปรุงอัตราการไหลสุทธิครั้งที่ 1 ของขั้นตอนวิธี Augmenting Flow....	29
รูปที่ 2.35 แสดงผลการปรับปรุงอัตราการไหลสุทธิครั้งที่ 2 ของขั้นตอนวิธี Augmenting Flow....	30
รูปที่ 2.36 แสดงผลการปรับปรุงอัตราการไหลสุทธิครั้งที่ 3 ของขั้นตอนวิธี Augmenting Flow....	30
รูปที่ 2.37 กราฟไม่ระบุทิศทาง หรือ กราฟ.....	31
รูปที่ 2.38 กราฟไม่ระบุทิศทางที่มีวงวน.....	32
รูปที่ 2.39 กราฟหลายเชิง.....	32
รูปที่ 2.40 ไคกราฟเชิงเดียว.....	32
รูปที่ 2.41 ไคกราฟหลายเชิง.....	32
รูปที่ 2.42 ตัวอย่างกราฟที่สมนัยกัน.....	33
รูปที่ 2.43 ตัวอย่างไคกราฟที่สมนัยกัน.....	33
รูปที่ 2.44 ตัวอย่างการแทนเครือข่ายเชิงเดียวด้วยเมทริกซ์.....	34
รูปที่ 3.1 แผนภาพการทำงานของขั้นตอนวิธี Ford-Fulkerson.....	36
รูปที่ 3.2 เครือข่ายเชิงเดียวสำหรับตัวอย่างที่ 1.....	37
รูปที่ 3.3 เครือข่ายตกค้างสำหรับตัวอย่างที่ 1.....	38
รูปที่ 3.4 เครือข่ายตกค้างที่มีค่าความจุตกค้างสำหรับตัวอย่างที่ 1.....	39
รูปที่ 3.5 เครือข่ายตกค้างสำหรับตัวอย่างที่ 2.....	40
รูปที่ 3.6 แผนภาพการทำงานของขั้นตอนวิธี Dinic.....	42
รูปที่ 3.7 เครือข่ายตกค้างสำหรับตัวอย่างที่ 3.....	44
รูปที่ 3.8 เครือข่ายเชิงเดียวสำหรับตัวอย่างที่ 4.....	45
รูปที่ 3.9 เครือข่ายเชิงเดียวหลังจากการขีดขวางอัตราการไหลสำหรับตัวอย่างที่ 4.....	46
รูปที่ 3.10 แผนภาพการทำงานของขั้นตอนวิธี Augmenting Flow.....	47
รูปที่ 3.11 เครือข่ายเชิงเดียวสำหรับตัวอย่างที่ 5.....	48
รูปที่ 3.12 กราฟมูลฐานสำหรับตัวอย่างที่ 5.....	48
รูปที่ 3.13 กราฟมูลฐานสำหรับตัวอย่างที่ 6.....	49



# บทที่ 1

## บทนำ

### 1.1 หลักการและเหตุผล

ปัญหาอัตราการไหลสูงสุดได้เกิดขึ้นมานานหลายสิบปี มีเอกสารที่ระบุว่าปัญหาการไหลสูงสุดนี้เกิดขึ้นตั้งแต่ต้นคริสต์ศตวรรษ 1930 โดย A.N. Tolstoi ได้คิดค้นหาแนวทางที่ดีที่สุดในการขนส่งสินค้าและยุทธโปกรณ์ผ่านทางระบบรถไฟในสหภาพโซเวียต ซึ่งนับได้ว่าแนวความคิดของ Tolstoi เป็นต้นแบบในการพัฒนาหาวิธีแก้ปัญหอัตราการไหลสูงสุด ในราวปี ค.ศ. 1951 G.B. Dantzig ได้พัฒนาปัญหาดังกล่าวให้มีความเป็นนัยทั่วไปมากขึ้น จากนั้นขั้นตอนวิธี (Algorithm) ที่ใช้ในการแก้ปัญหอัตราการไหลสูงสุดนี้ได้ถูกพัฒนาขึ้นเป็นครั้งแรกโดย Ford และ Fulkerson ในปี 1955 หลังจากนั้น ขั้นตอนวิธีแบบต่าง ๆ ได้ถูกพัฒนาขึ้นมาอย่างมากมาย เพื่อรองรับปัญหาที่มีความยุ่งยากซับซ้อนมากขึ้น และต้องการลดเวลาในการคำนวณลง

ขั้นตอนวิธีของ Ford-Fulkerson ใช้ได้กับเครือข่ายที่มีอัตราการไหลมีค่าเป็นจำนวนเต็มบวกหรือจำนวนตรรกยะบวกเท่านั้น ไม่สามารถนำไปใช้กับกรณีที่อัตราการไหลเป็นจำนวนอตรรกยะ หรือ อัตราการไหลที่เป็นลบ ทั้งนี้หลักการพื้นฐานของขั้นตอนวิธีของ Ford-Fulkerson อาศัยเครือข่ายตกค้าง (Residue network) ในการคำนวณหาผลเฉลยของปัญหาอัตราการไหลสูงสุด ซึ่งขั้นตอนวิธีของ Ford-Fulkerson ไม่ได้คำนึงถึงเวลาในการประมวลผลมากนัก จึงทำให้เสียเวลาในการหาผลลัพธ์ ต่อมาในราวปี ค.ศ. 1970 Dinic ได้พัฒนาขั้นตอนวิธีในการแก้ปัญหอัตราการไหลสูงสุด โดยอาศัยแนวความคิดจากขั้นตอนวิธีของ Ford-Fulkerson ร่วมกับแนวความคิดของการขัดขวางอัตราการไหล (Blocking flow) คือ การขัดขวางอัตราการไหลในเครือข่ายในกรณีที่ค่าอัตราการไหลมีค่าเท่ากับค่าความจุเพื่อลดความซับซ้อนในเครือข่าย ร่วมกับแนวความคิดการขัดขวางอัตราการไหล (Blocking Flow) และเครือข่ายระดับ (Level network) โดยที่ การขัดขวางอัตราการไหลคือ การลบหรือตัดส่วนโค้งที่โถมตัวออกจากเครือข่ายและเครือข่ายระดับ คือเครือข่ายที่มีระยะทางสั้นที่สุด (Shortest path) ภายในเครือข่ายจากแหล่งต้นทาง (source) ไปยังแหล่งปลายทาง (sink) โดยการใช้การค้นหาในแนวกว้าง (Breadth-first search) ในการหาเส้นทาง หรือ วิธีดังนั้นขั้นตอนวิธีนี้จึงมีการประมวลผลที่เร็วขึ้น

นอกจากนี้ ยังมีนักวิจัยหลายท่านที่ได้พัฒนาขั้นตอนวิธีขึ้นมาในการคำนวณหาอัตราการไหลสูงสุดในเครือข่าย อาทิ เช่น ขั้นตอนวิธีของ Edmonds-Kard, ขั้นตอนวิธีของ Goldberg-Tarjan, ขั้นตอนวิธีของ King-Rao-Tarjan และ ขั้นตอนวิธีของ Goldberg-Rao เป็นต้น

แม้ว่าจุดเริ่มต้นของปัญหาการหาอัตราการไหลสูงสุดนี้มาจากความต้องการในการแก้ไข ปัญหาการขนส่งสินค้าและยุทธวิธี แต่แนวทางการแก้ไขปัญหานี้สามารถนำไปประยุกต์ใช้ได้ อย่างกว้างขวาง อาทิ เช่น ปัญหาการจัดวางท่อส่งน้ำมัน เพื่อให้ได้การส่งน้ำมันสูงสุด ปัญหาการ วางสายพานในอุตสาหกรรมการผลิต ปัญหาการวางผังจราจรเพื่อเพิ่มปริมาณรถในถนนสายที่ ว่างอยู่ ปัญหาการเพิ่มอัตราการส่งข้อมูลผ่านทาง Router ในระบบอินเทอร์เน็ต เป็นต้น

ในการหาผลเฉลยของปัญหาการหาอัตราการไหลสูงสุดด้วยขั้นตอนวิธีต่าง ๆ ที่กล่าว ข้างต้น จำเป็นต้องอาศัยการพัฒนาชุดคำสั่งหรือโปรแกรมคอมพิวเตอร์เพื่อใช้ในการคำนวณ ซึ่ง การพัฒนาชุดคำสั่งนั้น ต้องอาศัยองค์ความรู้ด้านทฤษฎีเขียนภาษาคอมพิวเตอร์-อีกทั้งต้องอาศัย ประสบการณ์ของผู้พัฒนา ดังนั้น เพื่อให้ผู้ใช้งานทั่วไปสามารถคำนวณหาอัตราการไหลสูงสุดใน เครื่องมือได้อย่างสะดวก ไม่ซับซ้อน ผู้จัดทำจึงได้พัฒนาชุดคำสั่งที่อยู่ในรูปแบบของ ส่วนประสาน กราฟฟิกกับผู้ใช้ หรือ Graphical User Interface: GUI โดยที่ผู้ใช้งานทั่วไปเพียงแค่เอาเมาส์ไปเลือก ปุ่มที่ต้องการ หรือ พิมพ์ข้อมูลที่จำเป็นลงในกล่อง แล้วโปรแกรมจะทำการคำนวณหาผลเฉลยให้ โปรแกรมที่มีการพัฒนารูปแบบส่วนประสานกราฟฟิกกับผู้ใช้ได้แก่ โปรแกรม visual basic2005 โปรแกรม MATLAB เป็นต้น

โครงการนี้มุ่งเน้นในการพัฒนาและสร้างโปรแกรมที่เป็นแบบ ส่วนประสานกราฟฟิกกับ ผู้ใช้ ด้วยโปรแกรม-MATLAB-7.0 เนื่องจากว่าเป็นโปรแกรมที่สามารถทำให้ผู้ใช้เข้าใจได้ง่าย อีก หนึ่งเพื่ออำนวยความสะดวกให้แก่ผู้ใช้ในการคำนวณหาผลลัพธ์

## 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อมีความเข้าใจถึงขั้นตอนวิธีต่าง ๆ ในการคำนวณหาอัตราการไหลสูงสุดใน เครื่องมือได้
2. สามารถจำลองการทำงานของระบบเครือข่าย เพื่อเพิ่มประสิทธิภาพของอัตราการไหล ให้มีปริมาณใกล้เคียงกับค่าความจุ (Capacity) ได้
3. เพื่อนำความรู้ที่ได้ไปประยุกต์ใช้กับงานทางด้านอื่นได้

## 1.3 ขอบข่ายของโครงการ

1. สร้างโปรแกรมในรูปแบบของส่วนต่อประสานกราฟฟิกกับผู้ใช้โดยโปรแกรม MATLAB
2. สามารถจำลองเครือข่ายอย่างง่าย โดยจะมีการกำหนดบัพ (Node) จากผู้ใช้ไม่เกิน 20 บัพ
3. ผู้ใช้งานสามารถเลือกใช้อย่างน้อย 3 ขั้นตอนวิธี ในการคำนวณ
4. โปรแกรมสามารถแสดงผล ทั้งอัตราการไหลสูงสุด และจุดตัด (Cut) ได้อย่างถูกต้อง

## 1.4 ขั้นตอนการดำเนินโครงการ

กิจกรรม	เดือน - ปี											
	ต.ค. 49 - พ.ย. 49	ธ.ค. 49 - ม.ค. 50	ก.พ. 50 - มี.ค. 50	เม.ย. 50 - พ.ค. 50	มิ.ย. 50 - ก.ค. 50	ส.ค. 50 - ก.ย. 50	ต.ค. 50 - พ.ย. 50	ธ.ค. 50 - ม.ค. 51	ก.พ. 51 - มี.ค. 51	เม.ย. 51 - พ.ค. 51	มิ.ย. 51 - ก.ค. 51	ส.ค. 51 - กย. 51
ศึกษาโปรแกรม MATLAB	←		→									
ออกแบบและพัฒนา โปรแกรม			←				→					
ทดสอบและแก้ไข						←			→			
จัดทำรายงานและ สรุปผลการทำงาน								←				→

## 1.5 ผลที่คาดว่าจะได้รับ

1. สามารถหาอัตราการใช้หน่วยสูงที่สุดในระบบได้
2. ความรู้ทางด้านกราฟเขียนโปรแกรม MATLAB ที่เป็นแบบ GUI ได้
3. รู้จักการนำโปรแกรมมาประยุกต์ใช้ในด้านต่างๆ

## 1.6 งบประมาณ

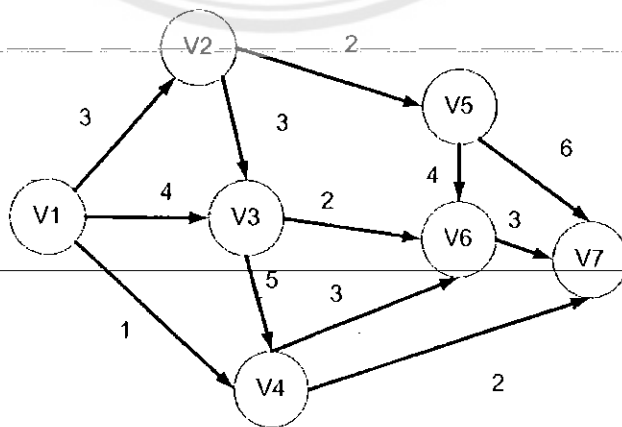
- |   |       |                       |
|---|-------|-----------------------|
| 1. ค่าวัสดุสำนักงาน                     | 700   | บาท                   |
| 2. ค่าถ่ายเอกสารและค่าเช่ารูปเล่มรายงาน | 300   | บาท                   |
| รวมเป็นเงินทั้งสิ้น                     | 1,000 | บาท (หนึ่งพันบาทถ้วน) |
- หมายเหตุ ถัวเฉลี่ยทุกรายการ

## บทที่ 2

### ทฤษฎีเครือข่าย

ก่อนที่จะกล่าวถึงนิยาม รวมทั้งทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับทฤษฎีเครือข่าย ให้พิจารณาตัวอย่างต่อไปนี้ โรงงานน้ำมันแห่งหนึ่งมีความต้องการจะส่งน้ำมันจากโรงขุดเจาะน้ำมัน ไปยังโรงกลั่น ซึ่งโรงงานน้ำมันดังกล่าวใช้ท่อส่งน้ำมันที่มีขนาด **ความจุ (Capacity)** แตกต่างกันไปในการส่งผ่านน้ำมันจากโรงขุดเจาะน้ำมัน ไปยัง โรงกลั่นต่าง ๆ โดยอาจจะผ่านสถานีพักน้ำมันเป็นระยะ ในกรณีนี้ หากกำหนดให้โรงกลั่นต่าง ๆ รวมทั้งสถานีพักน้ำมัน แทนด้วย **บัพ (Node)** หรือ **จุดยอด (Vertex)** และแทนท่อส่งน้ำมันด้วย **เส้นเชื่อม (Edge)** หรือ **ส่วนโค้ง (Arc)** โดยแต่ละเส้นเชื่อมมีการระบุอัตราการไหลของน้ำมันในเวลาต่าง ๆ กัน รวมทั้งความจุสูงสุดหรืออัตราการไหลสูงสุดของท่อที่น้ำมันสามารถไหลผ่านได้ แล้วผลที่ได้ ก็คือเครือข่ายของการส่งน้ำมัน ซึ่งเขียนแทนอยู่ในรูปของกราฟ หรือ ไดกราฟ นั่นเอง

ตัวอย่างของเครือข่ายของการส่งน้ำมันแสดงไว้ในรูปที่ 2.1 โดยที่ บัพ  $v_1$  คือ โรงขุดเจาะน้ำมัน บัพ  $v_7$  คือ โรงงานกลั่นน้ำมัน ในขณะที่บัพ  $v_2$  ถึง  $v_6$  แทนสถานีพักน้ำมัน ตัวเลขที่ระบุบนแต่ละเส้นเชื่อมบ่งบอกถึงค่าความจุสูงสุดที่น้ำมันสามารถไหลผ่านได้ภายในท่อ ปัญหาที่ผู้ออกแบบต้องการคือ ควรออกแบบให้น้ำมันไหลผ่านท่อด้วยอัตราเท่าไร ที่ทำให้น้ำมันไหลจากโรงขุดเจาะน้ำมัน  $v_1$  ไปยัง โรงงานกลั่นน้ำมัน  $v_7$  ด้วยอัตราการไหลที่สูงที่สุด ซึ่งปัญหานี้เป็นหนึ่งในหลาย ปัญหาที่สามารถอธิบายได้ด้วย กราฟ รวมถึง การคำนวณหาอัตราการไหลสูงสุดในเครือข่าย



รูปที่ 2.1 แสดงตัวอย่างของการแทน โรงงานน้ำมันด้วยเครือข่าย

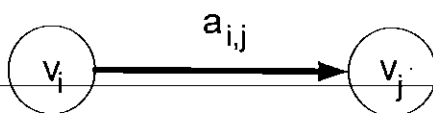
## 2.1 กราฟและไดกราฟ (Graph and Digraph)

หัวข้อนี้มุ่งศึกษานิยามของกราฟที่เกี่ยวข้องกับโครงงาน อันได้แก่ นิยามของกราฟและไดกราฟ ทางเดิน วิธี ในกราฟ รวมถึงระดับชั้นของบัพ เป็นต้น

**กราฟ (Graph) หรือ กราฟไม่ระบุทิศทาง (Undirected Graph)** คือคู่อันดับ  $G := (V, E)$  โดยที่  $V(G)$  คือ เซตที่ไม่ใช่เซตว่างอันมีสมาชิกเป็นจุดยอด (Vertices) หรือ บัพ (Nodes) ของกราฟ  $G$  และ  $E(G)$  คือ เซตของเส้นเชื่อมระหว่างบัพ  $v_i, v_j$  ใด ๆ ใน  $V(G)$  ซึ่งเขียนแทนด้วยสัญลักษณ์  $e_{i,j} = (v_i, v_j)$  หาก  $v_i = v_j$  แล้ว  $e_{i,i}$  ถูกเรียกว่า **วงวน (Loop)** นอกจากนี้ หากมีเส้นเชื่อมมากกว่าหนึ่งเส้นที่เชื่อมระหว่างบัพ  $v_i, v_j$  ใด ๆ ใน  $V(G)$  เส้นเชื่อมเหล่านั้นถูกเรียกว่า **เส้นเชื่อมขนาน (Parallel edges)** กราฟใด ๆ ที่ไม่มีวงวนและเส้นเชื่อมขนาน กราฟดังกล่าวนี้ถูกเรียกว่า **กราฟเชิงเดียว (Simple graph)** ในขณะที่กราฟที่มีวงวน หรือ เส้นเชื่อมขนาน จะถูกเรียกว่า **กราฟหลายเชิง (Multigraph)**

หากเส้นเชื่อมในกราฟมีการระบุทิศทาง เส้นเชื่อมดังกล่าวถูกเรียกว่า **ส่วนโค้ง (Arcs)** เพื่อแยกความแตกต่างจากเส้นเชื่อมในกราฟที่ไม่ระบุทิศทาง และนิยมเขียนแทนส่วนโค้งที่เชื่อมจากบัพ  $v_i$  ไปยังบัพ  $v_j$  ด้วย  $a_{i,j}$  สืบเนื่องจากที่ส่วนโค้งที่เกิดขึ้น มีการระบุทิศทางจากบัพหนึ่งไปยังอีกบัพหนึ่ง โดยทั่วไป หากมีส่วนโค้ง  $a_{i,j}$  ในกราฟแล้ว ไม่จำเป็นต้องมีส่วนโค้ง  $a_{j,i}$  ในกราฟนั้น และเพื่อแยกความแตกต่างระหว่างกราฟที่มีเส้นเชื่อม กับกราฟที่มีเส้นเชื่อมที่ระบุทิศทาง (หรือส่วนโค้ง) จึงนิยมเรียกกราฟที่มีส่วนโค้งว่า **ไดกราฟ (Digraph)** ซึ่งย่อมาจาก Directed graph หรือ กราฟที่มีการระบุทิศทาง

ในกราฟ  $G$  ใด ๆ หากมีเส้นเชื่อม  $e_{i,j} \in E(G)$  แล้ว  $v_i$  และ  $v_j$  ถูกกล่าวว่าเป็น **บัพประชิดกัน (Adjacent nodes)** โดยที่  $v_i, v_j$  เรียกว่าเป็น จุดปลายของเส้นเชื่อม แต่ในกรณีของไดกราฟ บัพ  $v_i$  และ  $v_j$  ถูกเรียกว่า จุดเริ่มต้น (Starting node) หรือ **หาง (Tail)** และจุดปลายทาง (Terminal node) หรือ **หัว (Head)** ของส่วนโค้ง  $a_{i,j}$  ตามลำดับ ตัวอย่างของส่วนโค้ง แสดงไว้ในรูปที่ 2.2

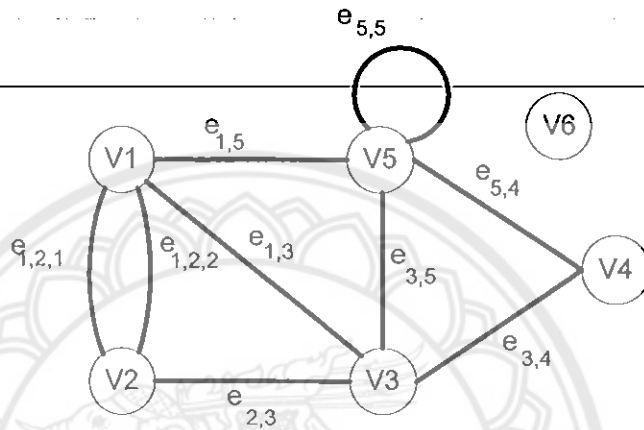


รูปที่ 2.2 แสดง “หัว” และ “หาง” ของส่วนโค้ง  $a_{i,j}$

ให้  $v_i$  เป็นบัพใด ๆ ในกราฟ  $G$  ระดับชั้นของ  $v_i$  คือจำนวนของเส้นเชื่อมที่มี  $v_i$  เป็นจุดปลายของเส้นเชื่อม ในขณะที่ ในกรณีของไดกราฟ ระดับชั้นของ  $v_i$  คือ ผลรวมของระดับชั้นเข้า

และระดับชั้นออก โดยที่ ระดับชั้นเข้า (In-degree) ของ  $v_i$  คือ จำนวนของส่วนโค้งที่มี  $v_i$  เป็น หัว (Head) และระดับชั้นออก (Out-degree) ของ  $v_i$  คือ จำนวนของส่วนโค้งที่มี  $v_i$  เป็น หาง (Tail) บัพใดที่ระดับชั้นเข้าเป็นศูนย์ บัพนั้นถูกกล่าวว่าเป็น **แหล่งกำเนิด (Source)** บัพใดที่ระดับชั้นออกเป็นศูนย์ บัพนั้นถูกเรียกว่าเป็น **แหล่งปลายทาง (Sink)** บัพใดที่ระดับชั้นเป็นศูนย์ บัพนั้นถูกเรียกว่า **บัพเอกเทศ (Isolated node)**

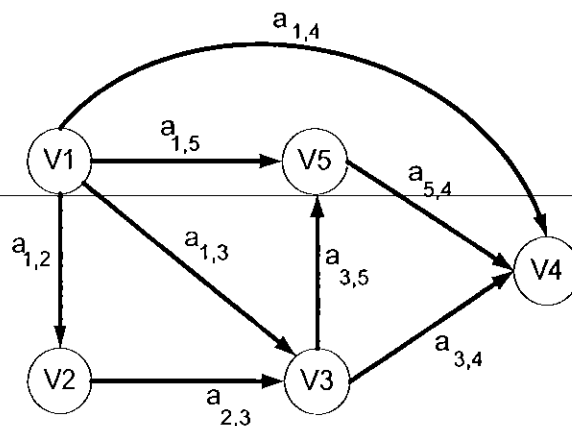
**ตัวอย่างที่ 1** ให้พิจารณากราฟต่อไปนี้



**รูปที่ 2.3** แสดงตัวอย่างของนิยามต่าง ๆ ที่เกี่ยวข้องกับกราฟ

จากรูปที่ 2.3 พบว่าเส้นเชื่อม  $e_{1,2,1}$  และ  $e_{1,2,2}$  เชื่อมระหว่างบัพคู่เดียวกัน นั่นคือ  $v_1$  และ  $v_2$  ดังนั้นเส้นเชื่อม  $e_{1,2,1}$  และ  $e_{1,2,2}$  คือ **เส้นเชื่อมขนาน (Parallel Edges)** ในขณะที่เส้นเชื่อม  $e_{5,5}$  คือ **วงวน** เนื่องจากกราฟในตัวอย่างนี้ มีทั้งเส้นเชื่อมขนานและวงวน ดังนั้น กราฟนี้จึงเป็น กราฟหลายเชิง นอกจากนี้ พบว่าบัพ  $v_1, v_2, v_3, v_4, v_5, v_6$  มีระดับชั้นเป็น 4, 3, 4, 2, 4, 0 และเนื่องจาก  $v_6$  เป็นบัพที่มีระดับชั้นเป็นศูนย์ ดังนั้น  $v_6$  จึงเป็น **บัพเอกเทศ**

**ตัวอย่างที่ 2** ให้พิจารณาไดกราฟต่อไปนี้



**รูปที่ 2.4** แสดงตัวอย่างของนิยามต่าง ๆ ที่เกี่ยวข้องกับไดกราฟ

จากรูปที่ 2.4 ส่วนโค้ง  $a_{i,j} = (v_i, v_j)$  เชื่อมต่อจาก  $v_i$  ไปยัง  $v_j$  ดังนั้น  $v_i$  คือ **หัว** และ  $v_j$  คือ **หาง** ของ  $a_{i,j}$  สำหรับระดับชั้นเข้า ระดับชั้นออก และระดับชั้น ของแต่ละบัพมีดังนี้

- $v_1$  มีระดับชั้นออกเท่ากับ 4 มีระดับชั้นเข้าเท่ากับ 0 ดังนั้นระดับชั้นของ  $v_1$  เท่ากับ 4
- $v_2$  มีระดับชั้นออกเท่ากับ 1 มีระดับชั้นเข้าเท่ากับ 1 ดังนั้นระดับชั้นของ  $v_2$  เท่ากับ 2
- $v_3$  มีระดับชั้นออกเท่ากับ 2 มีระดับชั้นเข้าเท่ากับ 2 ดังนั้นระดับชั้นของ  $v_3$  เท่ากับ 4
- $v_4$  มีระดับชั้นออกเท่ากับ 0 มีระดับชั้นเข้าเท่ากับ 3 ดังนั้นระดับชั้นของ  $v_4$  เท่ากับ 3
- $v_5$  มีระดับชั้นออกเท่ากับ 1 มีระดับชั้นเข้าเท่ากับ 2 ดังนั้นระดับชั้นของ  $v_5$  เท่ากับ 3

เนื่องจาก  $v_1$  มีระดับชั้นเข้าเป็นศูนย์ ดังนั้น  $v_1$  จึงเป็น **แหล่งต้นทาง** ในขณะที่  $v_4$  มีระดับชั้นออกเป็นศูนย์ ดังนั้น  $v_4$  เป็น **แหล่งปลายทาง**

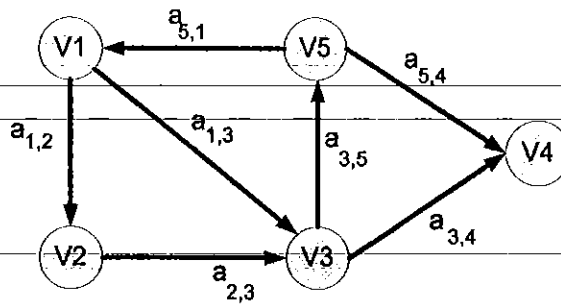
จากแนวความคิดของกราฟและไดกราฟข้างต้น หากพิจารณาให้กราฟ หรือไดกราฟนั้น คือแผนผังแสดงเส้นทางการไหลของน้ำมันจากโรงขุดเจาะน้ำมัน ไปยังโรงงานกลั่นน้ำมันผ่านตามสถานีพักน้ำมันต่างๆ จะได้ว่า โรงขุดเจาะน้ำมัน คือ แหล่งต้นทาง เนื่องจากระดับชั้นเข้าเป็นศูนย์ โรงงานกลั่นน้ำมัน คือ แหล่งปลายทาง เนื่องจากระดับชั้นออกเป็นศูนย์ ส่วนสถานีพักน้ำมันคือ บัพที่เหลือในกราฟ หากต้องการออกแบบให้อัตราการไหลรวมของน้ำมันในแต่ละท่อมีค่าสูงที่สุด จึงต้องพิจารณาแนวทางการไหลของน้ำมันทุกแนว จากโรงขุดเจาะน้ำมัน ไปยังโรงงานกลั่นน้ำมันด้วยเหตุนี้ จึงจำเป็นต้องสร้างนิยามที่เหมาะสมสำหรับ “แนวทางการไหลของน้ำมัน” ขึ้นมา เพื่อให้สามารถนำทฤษฎีของกราฟมาช่วยในการแก้ไขปัญหาดังกล่าว

ให้  $G := (V, E)$  เป็นไดกราฟ (หรือกราฟ) **แนวเดิน (Walk)** ในไดกราฟ คือ ลำดับจำกัดของบัพและส่วนโค้ง (หรือ เส้นเชื่อม) สลับกัน ดังนี้

$$W = (v_{i_1}, a_{i_1, i_2}, v_{i_2}, a_{i_2, i_3}, v_{i_3}, \dots, v_{i_{n-1}}, a_{i_{n-1}, i_n}, v_{i_n})$$

โดย  $a_{i_1, i_2}$  คือส่วนโค้งในไดกราฟ นอกจากนี้ บัพ  $v_{i_1}$  ถูกเรียกว่า **บัพเริ่มต้น** และบัพ  $v_{i_n}$  ถูกเรียกว่า **บัพจุดปลาย** ของแนวเดิน  $W$  หาก  $v_{i_1} = v_{i_n}$  แนวเดิน  $W$  ถูกกล่าวว่าเป็น **แนวเดินปิด (Closed walk)** แต่หากแนวเดินมีส่วนโค้งที่ไม่ซ้ำกัน แนวเดินดังกล่าวถูกเรียกว่า **รอยเดิน (Trail)** และหาก แนวเดินนั้น มีทั้งส่วนโค้งที่ไม่ซ้ำกันและบัพที่ไม่ซ้ำกันด้วย แนวเดินดังกล่าวถูกเรียกว่า **วิถี (Path)** สำหรับรอยเดินใด ๆ ที่เป็นรอยเดินปิด (Closed trail) รอยเดินลักษณะนั้นมีชื่อเรียกเฉพาะว่า **วงจร (Circuit)** และวิถีที่เป็นวิถีปิด (Closed path) มีชื่อเรียกว่า **วัฏจักร (Cycle)**

ตัวอย่างที่ 3 ให้พิจารณาไดกราฟต่อไปนี้



รูปที่ 2.5 แสดงตัวอย่างของแนวเดินประเภทต่าง ๆ ในไดกราฟ

จากกราฟในรูปที่ 2.5 ให้จำแนกแนวเดินต่อไปนี้

1.  $(v_5, a_{5,1}, v_1, a_{1,3}, v_3, a_{3,5}, v_5, a_{5,4}, v_4)$  เป็นรอยเดิน เนื่องจากไม่มีส่วนโค้งซ้ำ
2.  $(v_1, a_{1,2}, v_2, a_{2,3}, v_3, a_{3,5}, v_5, a_{5,1}, v_1)$  เป็นวัฏจักร หรือวัฏจักร
3.  $(v_1, a_{1,3}, v_3, a_{3,5}, v_5, a_{5,4}, v_4)$  เป็นวิถี

## 2.2 การวิเคราะห์หาอัตราการไหลสูงสุดในเครือข่ายเชิงเดียว

### (Maximum Flow Analysis in Simple Network)

คำว่า เครือข่ายเชิงเดียว หรือ Simple Network นั้น หมายถึง ไดกราฟเชิงเดียวที่เชื่อมโยกัน (Connected simple digraph) นั่นคือเป็น ไดกราฟที่ไม่มีบัพใดที่เป็นบัพเอกเทศ โดยที่แต่ละส่วนโค้งที่เชื่อมระหว่างบัพในเครือข่าย  $N$  มีฟังก์ชันถ่วงน้ำหนักสองฟังก์ชัน ดังนี้

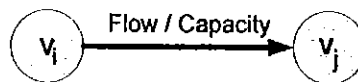
- ฟังก์ชันอัตราการไหลสุทธิ<sup>1</sup> (Net Flow function)

$$F: E(N) \rightarrow \mathbb{R} \text{ โดยที่ } F(a_{i,j}) := F_{i,j} \in \mathbb{R} \text{ สำหรับทุกส่วนโค้ง } a_{i,j} \in E(N)$$

- ฟังก์ชันค่าความจุ<sup>2</sup> (Capacity function)

$$C: E(N) \rightarrow \mathbb{R} \text{ โดยที่ } C(a_{i,j}) := C_{i,j} \in \mathbb{R} \text{ สำหรับทุกส่วนโค้ง } a_{i,j} \in E(N)$$

โดยทั่วไป นิยมเขียนฟังก์ชันถ่วงน้ำหนักทั้งสองไว้เหนือส่วนโค้ง ดังรูป



รูปที่ 2.6 แสดงถึงค่าอัตราการไหล และค่าความจุบนส่วนโค้ง  $a_{i,j}$

<sup>1</sup> คำว่า อัตราการไหลสุทธิจากบัพ  $u$  ไปยังบัพ  $v$  นั้น หมายความว่า หากในระบบเชิงกายภาพมีอัตราการไหลจากบัพ  $u$  ไปยังบัพ  $v$  เท่ากับ  $x$  และมีอัตราการไหลจากบัพ  $v$  ย้อนกลับไปยังบัพ  $u$  เท่ากับ  $y$  แล้ว อัตราการไหลสุทธิจากบัพ  $u$  ไปยังบัพ  $v$  มีค่าเท่ากับ  $x - y$  ในขณะที่อัตราการไหลสุทธิจากบัพ  $v$  ไปยังบัพ  $u$  มีค่าเป็น  $y - x$  ดังนั้น จึงมีความเป็นไปได้ที่อัตราการไหลสุทธิมีค่าเป็นลบ

<sup>2</sup> หากระหว่างบัพใด ๆ ในเครือข่ายไม่มีส่วนโค้งที่เชื่อมต่อกัน นั่นคือถ้า  $(v_i, v_j) \notin E(N)$  แล้วให้ถือว่า  $C_{i,j} = 0$



ฟังก์ชันอัตราการไหลสุทธิ  $F$  ที่นิยามบนเครือข่ายเชิงเดียวนั้น ต้องสอดคล้องตามเงื่อนไขต่อไปนี้

1. **เงื่อนไขค่าความจุ (Capacity constraints)**  
 $0 \leq F_{i,j} \leq C_{i,j}$  อัตราการไหลสุทธิข้อน้อยกว่าหรือเท่ากับค่าความจุ สำหรับทุกส่วนโค้ง  $a_{i,j} \in E(N)$  ในกรณีที่  $F_{i,j} = C_{i,j}$  เส้นโค้งดังกล่าวถูกเรียกว่า **ส่วนโค้งที่อิ่มตัว (Saturated arc)**
2. **ความเป็นสมมาตรเสมือน (Skew symmetry)**  
 $F_{i,j} = -F_{j,i}$  อัตราการไหลสุทธิจากบัพ  $v_i$  ไปยังบัพ  $v_j$  ย่อมมีทิศทางการไหลตรงข้ามกับอัตราการไหลสุทธิจากบัพ  $v_j$  ไปยังบัพ  $v_i$
3. **หลักอนุรักษนิยม (Conservatism)**
  - 3.1.  $F^-(v_i) = F^+(v_i)$  อัตราการไหลสุทธิที่ไหลเข้าเท่ากับอัตราการไหลสุทธิที่ไหลออก สำหรับทุกบัพที่ไม่ใช่แหล่งต้นทางและแหล่งปลายทาง
  - 3.2.  $F^-(s) = 0$  อัตราการไหลสุทธิที่ไหลเข้าสู่แหล่งต้นทาง (บัพ  $s := v_i$ ) เป็นศูนย์
  - 3.3.  $F^+(t) = 0$  อัตราการไหลสุทธิที่ไหลออกจากแหล่งปลายทาง (บัพ  $t := v_n$ ) เป็นศูนย์

**หมายเหตุ**  $F^-(v_i) = \sum_{j: a_{j,i} \in E(N)} F(a_{j,i})$  คือผลรวมของอัตราการไหลสุทธิที่ไหลเข้าสู่บัพ  $v_i$   
 $F^+(v_i) = \sum_{j: a_{i,j} \in E(N)} F(a_{i,j})$  คือผลรวมของอัตราการไหลสุทธิที่ไหลออกจากบัพ  $v_i$

นอกจากนี้ เซตของบัพในเครือข่ายเชิงเดียวถูกแบ่งออกเป็น 2 เซต เรียกว่า เซตของแหล่งต้นทาง ( $S$ ) และเซตของแหล่งปลายทาง ( $\bar{S}$ ) หากให้  $N := N(V, E)$  แทนเครือข่ายเชิงเดียว ดังนั้น  $V(N) = S \cup \bar{S}$  โดยที่  $s := v_i \in S$  และ  $t := v_n \in \bar{S}$  เมื่อ  $s$  และ  $t$  แทนแหล่งต้นทาง และแหล่งปลายทาง ตามลำดับ เมื่อเซตของบัพถูกแบ่งออกเป็นสองเซตดังกล่าวข้างต้น จึงกำหนดให้  $K$  แทนเซตของส่วนโค้ง  $a_{i,j}$  สำหรับทุกค่า  $i, j$  ที่ซึ่ง  $v_i \in S$  และ  $v_j \in \bar{S}$  นั่นคือ

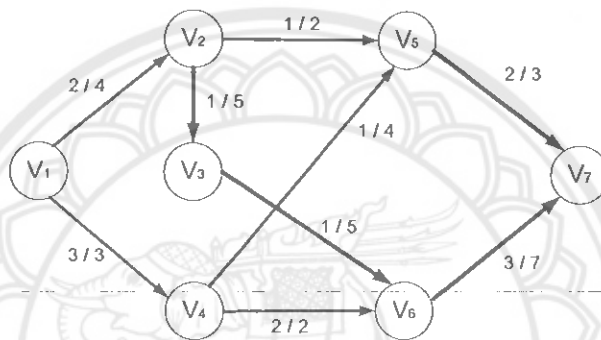
$$K = (S, \bar{S}) = \{a_{i,j} = (v_i, v_j) \in E(N) : v_i \in S, v_j \in \bar{S}\}$$

ในกรณีนี้  $K$  ถูกกล่าวว่าเป็น **จุดตัด (Cut)** ของเครือข่าย  $N$

อัตราการไหลสุทธิในเครือข่ายเชิงเดียว  $N$  ถูกกล่าวว่าเป็น**อัตราการไหลสูงสุด** ( $F_{\max}$ ) หาก  $Val(F_{\max}) \geq Val(F)$  สำหรับทุกอัตราการไหลสุทธิอื่น ๆ ที่เป็นไปได้ใน  $N$  โดยที่  $Val(F)$  คือ ผลรวมของอัตราการไหลสุทธิที่ไหลออกของแหล่งกำเนิด( $s$ ) หรือนั่นคือ  $Val(F) = F^+(s)$

จุดตัดถูกกล่าวว่าเป็น **จุดตัดที่ให้ค่าความจุที่น้อยที่สุด** ( $K_{\min}$ ) หาก  $C(K_{\min}) \leq C(K)$  สำหรับทุกจุดตัดอื่น ๆ ที่เป็นไปได้ในเครือข่าย โดยที่  $C(K)$  คือ ผลรวมของค่าความจุบนส่วนโค้ง ในบริเวณจุดตัด นั่นคือ  $C(K) = \sum_{a_{i,j} \in K} C(a_{i,j})$

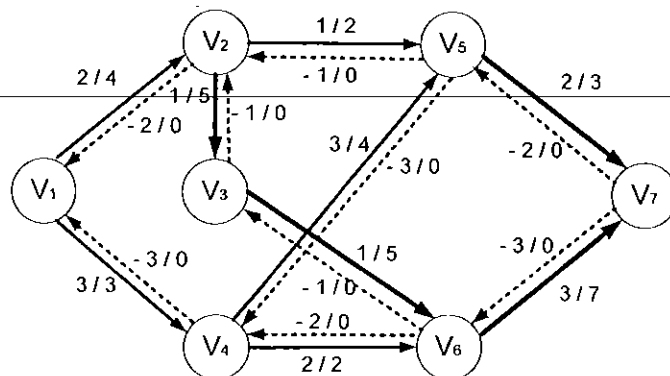
**ตัวอย่างที่ 4** ให้พิจารณาเครือข่ายเชิงเดียวดังแสดงในรูปที่ 2.7



**รูปที่ 2.7** แสดงตัวอย่างของเครือข่ายเชิงเดียว

จากรูป ผู้อ่านสามารถพิจารณาให้  $S = \{v_1\}$  และ  $\bar{S} = \{v_2, v_3, v_4, v_5, v_6, v_7\}$  หรือเลือกให้  $S = \{v_1, v_2\}$  และ  $\bar{S} = \{v_3, v_4, v_5, v_6, v_7\}$  แต่ต้องอยู่ภายใต้เงื่อนไขคือ  $v_1 \in S$  และ  $v_7 \in \bar{S}$  เนื่องจาก  $v_1$  เป็นแหล่งต้นทาง และ  $v_7$  เป็นแหล่งปลายทาง

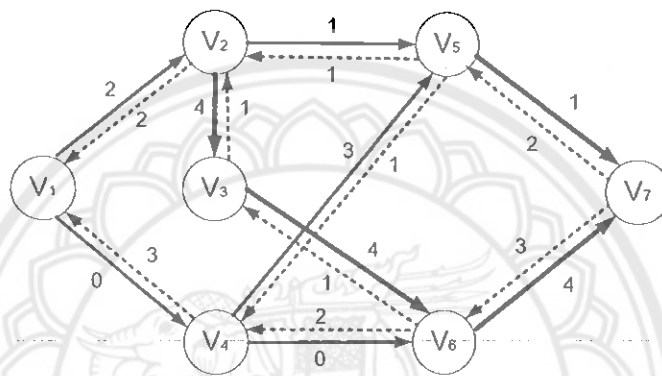
นอกจากนี้ เพื่อสะดวกในการวิเคราะห์หาอัตราการไหลสูงสุด จึงใช้เงื่อนไขของความเป็นสมมาตรเสมือนของฟังก์ชันอัตราการไหลสุทธิ มาเพิ่มเข้าไปในเครือข่ายที่โจทย์กำหนด ซึ่งแสดงไว้ในรูปที่ 2.8 ผู้อ่านสามารถตรวจสอบได้โดยไมยากว่า ฟังก์ชันอัตราการไหลสุทธิในเครือข่ายเชิงเดียวที่ได้ในรูปที่ 2.8 ยังคงรักษาเงื่อนไขทั้ง 3 ข้อของความเป็นฟังก์ชันอัตราการไหลสุทธิ



**รูปที่ 2.8** แสดงตัวอย่างของเครือข่ายเชิงเดียวที่มีอัตราการไหลเสมือน

หากกำหนดให้  $\Delta(a_{i,j}) := \Delta_{i,j} = C_{i,j} - F_{i,j}$  คือผลต่างระหว่างค่าความจุกับอัตราการไหลสุทธิของส่วนโค้งที่เชื่อมระหว่างบัพ  $v_i$  และ  $v_j$  ซึ่งโดยทั่วไป นิยมเรียกแทนผลต่างนี้ว่า **ค่าความจุตกค้าง หรือ Residue capacity** และไดกราฟหรือเครือข่ายที่ได้จากเครือข่ายเชิงเดียว  $N(V, E)$  แต่มีฟังก์ชันถ่วงน้ำหนักบนแต่ละส่วนโค้งเป็นค่าความจุตกค้าง เรียกว่า **เครือข่ายตกค้าง (Residue network)** ซึ่งเขียนแทนด้วย  $R_N(V, E_N)$  โดยเครือข่ายนี้ใช้บ่งบอกถึงความสามารถที่เครือข่ายสามารถเพิ่มอัตราการไหลได้อีกเป็นปริมาณมากหรือน้อยเท่าไร

เครือข่ายตกค้างของเครือข่ายในตัวอย่างที่ 4 แสดงไว้ในรูปที่ 2.9



รูปที่ 2.9 แสดงตัวอย่างของเครือข่ายตกค้าง

ผู้อ่านสามารถสังเกตได้ว่า ค่าความจุตกค้างบนส่วนโค้งที่เกิดจากเงื่อนไขความสมมาตรเสมือน คือ อัตราการไหลสุทธิ เช่น  $\Delta_{2,1} = 0 - (-2) = 2 = f_{1,2}$  ที่เป็นเช่นนี้ เนื่องจากบนส่วนโค้งที่มีอัตราการไหลเสมือนนั้น แท้จริงไม่มีอัตราการไหลดังกล่าว ดังนั้นค่าความจุบนส่วนโค้งนี้จึงมีค่าเป็นศูนย์ และอัตราการไหลเสมือนมีขนาดเท่ากับอัตราการไหลสุทธิเพียงแต่มีเครื่องหมายเป็นลบ ดังนั้น ค่าความจุตกค้างบนส่วนโค้งดังกล่าว จึงมีค่าเท่ากับ  $0 - (-f) = f$  นั่นเอง

### 2.2.1 แนวความคิดในการปรับปรุงอัตราการไหล

สืบเนื่องจากเงื่อนไขค่าความจุที่กล่าวไว้ว่า  $0 \leq F_{i,j} \leq C_{i,j}$  ดังนั้นอัตราการไหลสุทธิสูงสุดที่เป็นไปได้ต้องมีค่าไม่เกินค่าความจุบนแต่ละส่วนโค้ง ดังนั้น เมื่อกำหนดเครือข่ายใด ๆ มา การปรับปรุงอัตราการไหลเพื่อให้ได้อัตราการไหลสุทธิสูงสุดนั้น จึงต้องพยายามเพิ่มอัตราการไหลในแต่ละส่วนโค้งให้มีค่าเข้าใกล้ค่าความจุ แต่อย่างไรก็ตาม การเพิ่มอัตราการไหลสุทธินั้น ต้องรักษาตามหลักอนุรักษ์นิยมข้อ 3.1 นั่นคือ  $F^-(v_i) = F^+(v_i)$

ในเครือข่ายที่กำหนดให้ อาจมีอัตราการไหลสุทธิบนบางส่วนโค้งที่ไม่มีความจำเป็น หรืออัตราการไหลสุทธิที่โจทย์กำหนดมานั้นเกิดจากการแบ่งอัตราการไหลสุทธิจากบัพอื่น ซึ่งส่งผลให้อัตราการไหลสุทธิโดยรวมของทั้งเครือข่ายลดลง ในกรณีเช่นนี้ ผู้ออกแบบควรลดอัตราการไหล

สุทธิในส่วนโค้งดังกล่าวให้มากที่สุด กล่าวคือ พยายามทำให้อัศจรรย์ไหลสุทธิในส่วนโค้งนั้น ๆ มีค่าเข้าสู่ศูนย์นั่นเอง

เพื่อให้บรรลุวัตถุประสงค์ข้างต้น จึงต้องสร้างเครือข่ายตกค้างจากเครือข่ายที่โจทย์กำหนด เพื่อให้ทราบว่าแต่ละส่วนโค้งสามารถเพิ่มอัศจรรย์ไหลสุทธิได้มากเท่าไร จากนั้นทำการสร้างวิถี (path) ที่เริ่มจาก แหล่งต้นทาง ไปยังแหล่งปลายทาง โดยให้แต่ละวิถีที่สร้างได้เป็น  $P_k$  และให้  $\Delta_{i,j}$  เป็นค่าความจุตกค้างบนส่วนโค้ง  $a_{i,j}$  ตามวิถีของ  $P_k$  ที่สร้างขึ้น เนื่องจากค่าความจุตกค้างนั้น คือผลต่างของค่าความจุและอัศจรรย์ไหลสุทธิ ดังนั้นเพื่อเพิ่มอัศจรรย์ไหลให้สูงขึ้น จึงควรออกแบบให้อัศจรรย์ไหลสุทธิที่ไหลผ่านส่วนโค้ง  $a_{i,j}$  มีค่าเป็น  $F_{i,j} + \Delta_{i,j}$  ซึ่งนั่นคือค่าความจุบนเส้นโค้งนั่นเอง แต่อย่างไรก็ตาม การเพิ่มอัศจรรย์ไหลตามวิถีที่สร้างขึ้นนั้น ต้องเพิ่มด้วยอัตราเท่ากันทุกส่วนโค้งตลอดวิถี ดังนั้นจึงจำเป็นต้องเลือก ค่าความจุตกค้างบนวิถีให้มีค่าเท่ากับค่าต่ำที่สุดของค่าความจุบนแต่ละส่วนโค้ง หรือนั่นคือ  $\Delta(P_k) = \min_{a_{i,j} \in P_k} \Delta_{i,j}$  เพื่อไม่ให้มีอัศจรรย์ไหลสุทธิบนส่วนโค้งใดมีค่าเกินค่าความจุบนส่วนโค้งนั้น ๆ ได้ เนื่องจาก

$$F_{i,j} + \Delta(P_k) \leq F_{i,j} + \Delta_{i,j} = C_{i,j} \text{ สำหรับทุกส่วนโค้ง } a_{i,j} \text{ ตามวิถี } P_k$$

จากนั้นจึงทำการปรับเพิ่มอัศจรรย์ไหลบนวิถี  $P_k$  ด้วยค่าความจุตกค้างบนวิถี  $\Delta(P_k)$  นั่นคือ หากเดิมอัศจรรย์ไหลสุทธิ (โดยไม่พิจารณาว่าเป็นอัศจรรย์ไหลจริงหรืออัศจรรย์ไหลเสมือน) บนส่วนโค้ง  $a_{i,j}$  มีค่าเป็น  $F_{i,j}$  แล้วอัศจรรย์ไหลสุทธิหลังจากปรับแต่งแล้วมีค่าเป็น  $F_{i,j} + \Delta(P_k)$  หากพิจารณาในเครือข่ายตกค้าง พบว่า เมื่ออัศจรรย์ไหลเพิ่มขึ้น ทำให้ค่าความจุตกค้างลดลง ดังนั้น ในการปรับค่าความจุตกค้างในเครือข่ายตกค้างนั้น จึงต้องนำค่าความจุบนวิถีไปหักลบออกจากค่าความจุตกค้างบนแต่ละส่วนโค้งตามทิศทางของวิถี นั่นคือ

ก่อนปรับ	หลังปรับ
$F_{i,j}$	$F_{i,j} + \Delta(P_k)$
$C_{i,j} - F_{i,j} = \Delta_{i,j}$	$C_{i,j} - (F_{i,j} + \Delta(P_k)) = \Delta_{i,j} - \Delta(P_k)$

### ตารางที่ 1 แสดงการปรับอัศจรรย์ไหล และค่าความจุตกค้างบนส่วนโค้งตามวิถี

เนื่องจากการสร้างเครือข่ายตกค้างนั้น หากมีเส้นเชื่อม  $a_{i,j}$  แล้วย่อมมีเส้นเชื่อม  $a_{j,i}$  เสมอ อันเนื่องมาจากเงื่อนไขความสมมาตรเสมือนดังได้กล่าวมาข้างต้น ดังนั้น เมื่อไรก็ตามที่อัศจรรย์ไหลสุทธิ (โดยไม่พิจารณาว่าเป็นอัศจรรย์ไหลจริงหรืออัศจรรย์ไหลเสมือน)  $F_{i,j}$  มีการเพิ่มค่าเป็น  $F_{i,j} + \Delta(P_k)$  ดังนั้น อัศจรรย์ไหลสุทธิที่มีทิศทางย้อนกลับ นั่นคือ  $F_{j,i}$  (ซึ่งมีค่าเท่ากับ

$-F_{i,j}$ ) ย่อมต้องมีการปรับลดลงด้วยปริมาณเท่ากับ  $F_{j,i} - \Delta(P_k)$  เช่นกัน และหากมองในแง่ของค่าความจุตกค้างในเครือข่ายตกค้าง พบว่าค่าความจุตกค้างบนส่วนโค้งที่ย้อนทิศทางของวิถีมีค่าลดลง ดังแสดงในตารางต่อไปนี้

ก่อนปรับ	หลังปรับ
$F_{j,i}$	$F_{j,i} - \Delta(P_k)$
$C_{j,i} - F_{j,i} = \Delta_{j,i}$	$C_{j,i} - (F_{j,i} - \Delta(P_k)) = \Delta_{j,i} + \Delta(P_k)$

## ตารางที่ 2 แสดงการปรับอัตราการไหล และค่าความจุตกค้างบนส่วนโค้งย้อนวิถี

โดยกระบวนการปรับอัตราการไหล (หรือค่าความจุตกค้าง) ดังกล่าวนี้อ่านสามารถสังเกตได้ว่า จะต้องมียังน้อย 1 ส่วนโค้งบนวิถีนั้นที่ต้องเป็นส่วนโค้งอิมตัว นั่นคือ ส่วนโค้งที่มีอัตราการไหลสุทธิเท่ากับค่าความจุบนส่วนโค้ง (หรือค่าความจุตกค้างเป็นศูนย์) ซึ่งทำให้ไม่สามารถเพิ่มอัตราการไหลบนส่วนโค้งนี้ได้ จึงต้องทำการเลือกวิถีอื่นแล้วเริ่มกระบวนการเช่นนี้ไปเรื่อย ๆ จนไม่สามารถหาวิถีที่ไม่มีส่วนโค้งอิมตัวปรากฏอยู่บนวิถีนั้นได้ ซึ่งผลที่ได้คือเครือข่ายที่มีอัตราการไหลสุทธิสูงสุดนั่นเอง

### 2.2.2 ขั้นตอนวิธีของ Ford-Fulkerson

แนวความคิดที่กล่าวในหัวข้อ 2.2.1 เป็นแนวความคิดพื้นฐานของขั้นตอนวิธีของ Ford-Fulkerson ซึ่งสามารถสรุปได้ดังนี้

กำหนดให้  $N(V, E)$  เป็นเครือข่ายเชิงเดียวที่มี  $v_s$  เป็นแหล่งต้นทาง และ  $v_t$  เป็นแหล่งปลายทาง

#### 1. ขั้นเริ่มต้น (Initialization)

- ให้อัตราการไหลสุทธิบนทุกส่วนโค้งมีค่าเป็นศูนย์ ( $F_{i,j} = 0$  สำหรับทุกส่วนโค้ง  $a_{i,j} \in E(N)$ )
- สร้างเครือข่ายตกค้าง  $R_N(V, E_N)$

## 2. ขั้นทำซ้ำ (Iteration)

ในกรณีที่สามารหาวัด<sup>3</sup>  $P_k$  จาก  $v_s$  ไปยัง  $v_t$  ในเครือข่ายตกค้างที่ซึ่งมีค่า

ความจุตกค้าง  $\Delta_{i,j} > 0$  สำหรับทุกส่วนโค้ง  $a_{i,j}$  บนวิถี

- หาค่าความจุตกค้างบนวิถี  $\Delta(P_k) = \min_{a_{i,j} \in P_k} \Delta_{i,j} > 0$
- ปรับค่าความจุตกค้างบนแต่ละส่วนโค้ง  $a_{i,j}$  บนวิถี ดังนี้
  - $\Delta_{i,j} \rightarrow \Delta_{i,j} - \Delta(P_k)$
  - $\Delta_{j,i} \rightarrow \Delta_{j,i} + \Delta(P_k)$

## 3. ขั้นจบการทำงาน

เมื่อไม่สามารถหาวิถี  $P_k$  ที่ซึ่ง  $\Delta(P_k) > 0$  ได้

### หมายเหตุ

ขั้นตอนวิธีของ Ford-Fulkerson จบการทำงานเมื่อไม่สามารถหาวิถีที่ซึ่ง  $\Delta > 0$  ได้ นั้นย่อมแสดงว่า แหล่งต้นทาง  $v_s$  ไม่สามารถเชื่อมต่อกับแหล่งปลายทาง  $v_t$  ได้ในเครือข่ายตกค้าง (ค่าถ่วงน้ำหนักในเครือข่ายตกค้างเป็นศูนย์เสมือนว่าไม่มีเส้นเชื่อมต่อกัน) ดังนั้น หากให้  $S$  และ  $\bar{S}$  แทนเซตของบัพที่ซึ่งเชื่อมต่อกับ  $v_s$  และเชื่อมถึง  $v_t$  ตามลำดับ เครือข่ายที่โจทย์กำหนดมาจึงเกิดจุดตัด  $K = (S, \bar{S})$  โดยที่ค่าความจุสุทธิโดยรวมในเครือข่ายที่บริเวณจุดตัดดังกล่าวย่อมมีค่าเท่ากับอัตราการไหลสุทธิโดยรวมของเครือข่าย ซึ่งจากข้อสังเกตนี้ สามารถนำไปใช้พิสูจน์ทฤษฎีบทที่สำคัญในเรื่อง การหาอัตราการไหลสูงสุดในเครือข่าย ที่กล่าวไว้ ดังนี้

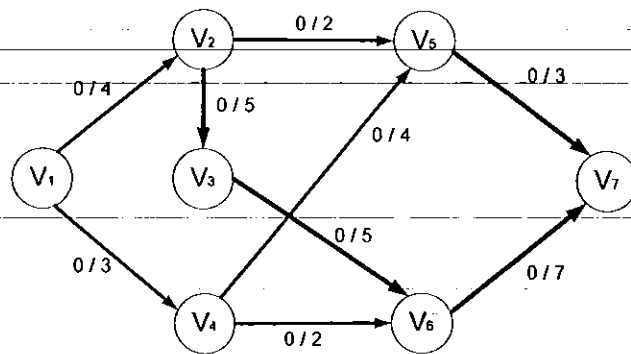
### ทฤษฎีบท (Maximum Flow/Minimum Cut)

ให้  $F$  คือ อัตราการไหลที่ออกจากแหล่งต้นทาง และ  $K = (S, \bar{S})$  คือจุดตัดในเครือข่าย ถ้า  $Val(F) = C(K)$  แล้ว  $F$  คือ อัตราการไหลสูงสุด และ  $K$  คือ จุดตัดที่มีค่าความจุน้อยที่สุด โดยที่

$$Val(F) = C(K) \Leftrightarrow \begin{cases} F(a_{i,j}) = 0; & \forall a_{i,j} \in (\bar{S}, S) \\ F(a_{i,j}) = C(a_{i,j}); & \forall a_{i,j} \in (S, \bar{S}) \end{cases}$$

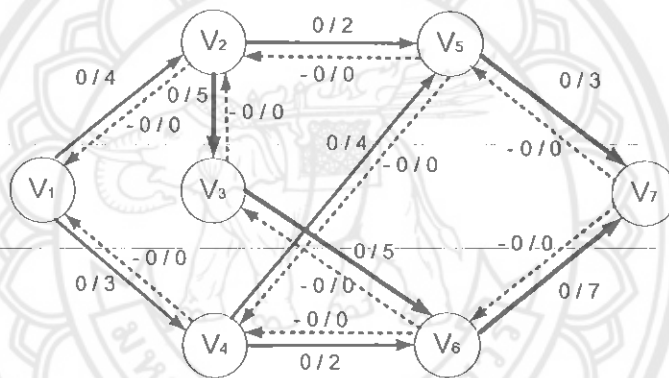
<sup>3</sup> การหาวิถีในเครือข่ายนั้น สามารถเลือกใช้การค้นหาในแนวกว้าง (Breadth-first-search) หรือการค้นหาในแนวลึก (Depth-first-search) หากเลือกใช้การค้นหาในแนวกว้าง ขั้นตอนวิธีนี้ถูกเรียกว่า ขั้นตอนวิธีของ Edmonds-Karp

**ตัวอย่างที่ 5** ให้พิจารณาเครือข่ายเชิงเดียวที่วาดไว้ในรูปที่ 2.10



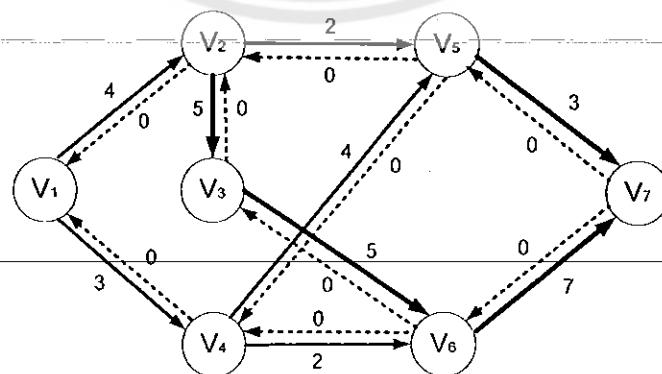
**รูปที่ 2.10** แสดงเครือข่ายเชิงเดียวของขั้นตอนวิธีของ Ford-Fulkerson

จากเครือข่ายในรูปที่ 2.10 นำมาวาดเครือข่ายเชิงเดียวที่มีอัตราการไหลเสมือนได้ดังนี้



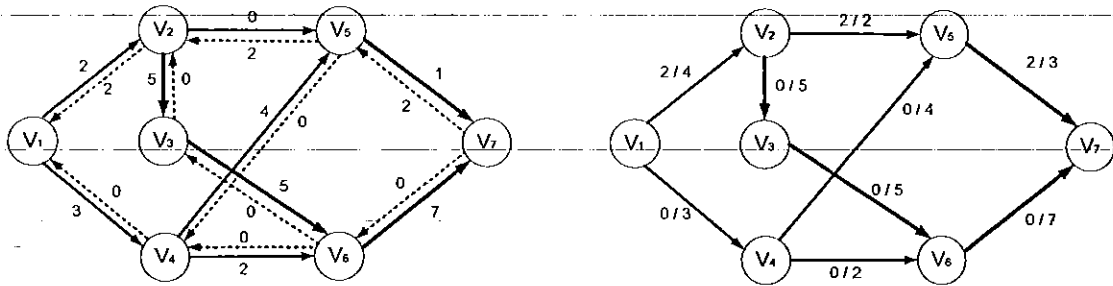
**รูปที่ 2.11** แสดงเครือข่ายเชิงเดียวที่มีอัตราการไหลเสมือนของขั้นตอนวิธีของ Ford-Fulkerson

จากเครือข่ายที่ได้ นำมาวาดเครือข่ายตกค้าง ได้ดังนี้



**รูปที่ 2.12** แสดงเครือข่ายตกค้างของขั้นตอนวิธีของ Ford-Fulkerson

ให้วิถีแรก  $P_1$  คือลำดับของ  $v_1, v_2, v_5, v_7$  ซึ่งพบว่า  $\Delta(P_1) = 2$  ซึ่งหลังจากมีการปรับอัตราการไหล หรือ ค่าความจุตกค้าง จะได้เครือข่ายตามรูปที่ 2.13

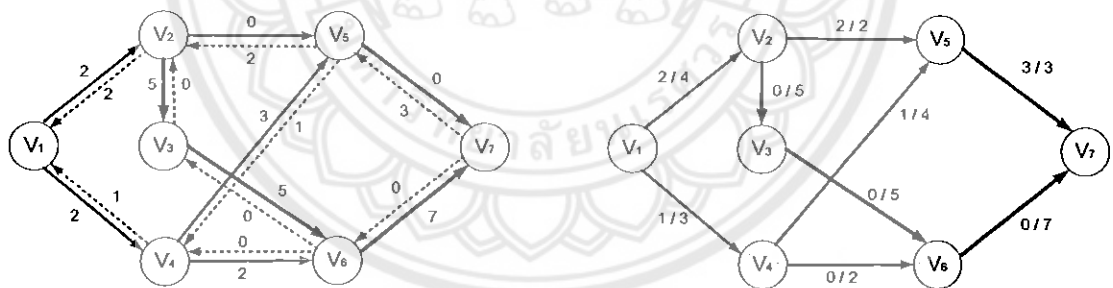


**รูปที่ 2.13** แสดงเครือข่ายที่มีการปรับครั้งที่ 1 ของขั้นตอนวิธีของ Ford-Fulkerson

(รูปซ้ายมือ) แสดงเครือข่ายตกค้างที่ปรับค่าความจุตกค้าง

(รูปขวามือ) แสดงเครือข่ายเชิงเดียวที่ปรับอัตราการไหลสุทธิ

เนื่องจากยังสามารถหาวิถีอื่นได้ที่ซึ่ง  $\Delta > 0$  จึงทำการหาวิถีใหม่ เลือกวิถีถัดไปเป็นลำดับของ  $v_1, v_4, v_5, v_7$  ซึ่งแทนวิถีนี้ด้วย  $P_2$  โดยมี  $\Delta(P_2) = 1$  และเมื่อทำการปรับอัตราการไหล หรือ ค่าความจุตกค้าง จะได้เครือข่ายตามรูปที่ 2.14



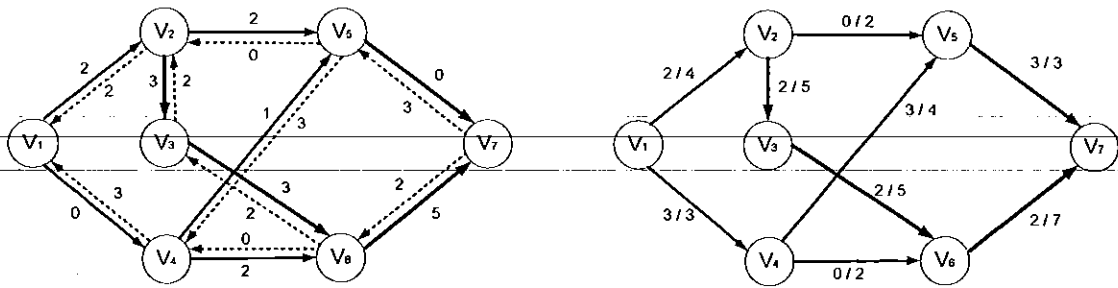
**รูปที่ 2.14** แสดงเครือข่ายที่มีการปรับครั้งที่ 2 ของขั้นตอนวิธีของ Ford-Fulkerson

(รูปซ้ายมือ) แสดงเครือข่ายตกค้างที่ปรับค่าความจุตกค้าง

(รูปขวามือ) แสดงเครือข่ายเชิงเดียวที่ปรับอัตราการไหลสุทธิ

เลือกวิถีถัดไปเป็นลำดับของ  $v_1, v_4, v_5, v_2, v_3, v_6, v_7$  ซึ่งแทนวิถีนี้ด้วย  $P_3$  โดยมี  $\Delta(P_3) = 2$  และเมื่อทำการปรับอัตราการไหล หรือ ค่าความจุตกค้าง จะได้เครือข่ายตามรูปที่ 2.15



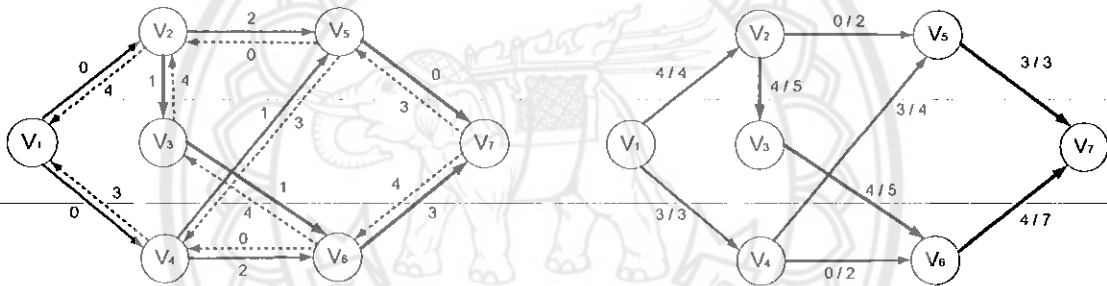


รูปที่ 2.15 แสดงเครือข่ายที่มีการปรับครั้งที่ 3 ของขั้นตอนวิธีของ Ford-Fulkerson

(รูปซ้ายมือ) แสดงเครือข่ายตกค้างที่ปรับค่าความจุตกค้าง

(รูปขวามือ) แสดงเครือข่ายเชิงเดียวที่ปรับอัตราการไหลสุทธิ

เนื่องจากยังสามารถหาวิถีที่ซึ่ง  $\Delta > 0$  กล่าวคือวิถีที่เป็นลำดับของ  $v_1, v_2, v_3, v_6, v_7$  ซึ่งแทนวิถีนี้ด้วย  $P_4$  โดยมี  $\Delta(P_4) = 2$  และเมื่อทำการปรับอัตราการไหล หรือ ค่าความจุตกค้าง จะได้เครือข่ายตามรูปที่ 2.16



รูปที่ 2.16 แสดงเครือข่ายที่มีการปรับครั้งที่ 4 ของขั้นตอนวิธีของ Ford-Fulkerson

(รูปซ้ายมือ) แสดงเครือข่ายตกค้างที่ปรับค่าความจุตกค้าง

(รูปขวามือ) แสดงเครือข่ายเชิงเดียวที่ปรับอัตราการไหลสุทธิ

เนื่องจากไม่สามารถหาวิถีใดในเครือข่ายตกค้างที่ซึ่ง  $\Delta > 0$  ได้ จึงจบการทำงาน โดยมีอัตราการไหลสูงสุดเท่ากับ 7 ซึ่งมีค่าเท่ากับค่าความจุต่ำสุดบริเวณจุดตัด  $K$  โดยที่จุดตัด  $K$  คือ

$$K = (S, \bar{S}) = \{(v_1, v_2), (v_1, v_4)\}$$

### 2.2.3 ขั้นตอนวิธีของ Dinic

ขั้นตอนวิธีของ Dinic มีความคล้ายคลึงกับขั้นตอนวิธีของ Ford-Fulkerson กล่าวคืออาศัยกราฟชนิดพิเศษ นั่นคือ เครือข่ายตกค้าง(Residue network) แต่ต่างกันตรงที่ในกรณีของ Dinic นั้นใช้การค้นหาวิถีจากแหล่งต้นทางไปยังแหล่งปลายทาง ด้วยการค้นหาแบบแนวกว้าง(Breadth-first search)ในเครือข่ายระดับ และอาศัยหลักการการขัดขวางอัตราการไหล หรือ Blocking Flow เพื่อลดเวลาในการคำนวณหาวิถีถึง อีกทั้งยังเป็นการเพิ่มประสิทธิภาพของการหาอัตราการไหลสูงสุด ซึ่งหลักการดังกล่าวนี้ทำการลบส่วนโค้งในเครือข่ายที่มีอัตราการไหลสุทธิเท่ากับค่าความจุ หรือ

นั่นคือ ค่าความจุตกค้างเป็นศูนย์บนส่วนโค้งนั้น ๆ ซึ่งทำให้ส่วนโค้งลดลง การเชื่อมต่อระหว่างบัพลดลงเรื่อย ๆ ทำให้ขั้นตอนการคำนวณหาวิถีลดลง และขั้นตอนวิธีจะหยุดเมื่อไม่สามารถหาวิถีที่ซึ่งมีค่าความจุตกค้าง  $\Delta > 0$  ได้ หลักการคำนวณโดยขั้นตอนวิธีของ Dinic แสดงไว้ดังนี้

กำหนดให้  $N(V, E)$  เป็นเครือข่ายเชิงเดียวที่มี  $v_s$  เป็นแหล่งต้นทาง และ  $v_t$  เป็นแหล่งปลายทาง

1. ขั้นเริ่มต้น (Initialization)

สร้างฟังก์ชันค่าถ่วงน้ำหนักบนแต่ละส่วนโค้งในเครือข่ายเชิงเดียว

$\Phi: E(N) \rightarrow \{1\}$  โดยที่  $\Phi(a_{i,j}) := \Phi_{i,j} = 1$  สำหรับทุกส่วนโค้ง  $a_{i,j} \in E(N)$

2. ขั้นทำซ้ำ (Iteration)

- สร้างเครือข่ายระดับจากแหล่งต้นทาง  $v_s$  ไปยังแหล่งปลายทาง  $v_t$  ด้วยการค้นหาแบบแนวกว้าง (Breadth-first search)
- คำนวณหาวิถีที่สั้นที่สุดจาก  $v_s$  ไปยัง  $v_t$  จากเครือข่ายระดับที่ได้
- ใช้ขั้นตอนวิธีของ Ford-Fulkerson ในการปรับปรุงอัตราการไหลสุทธิ
- ใช้หลักการขัดขวางการไหลสุทธิ (Blocking Flow) ในการลบหรือตัดส่วนโค้งที่อึดตัวจากเครือข่าย

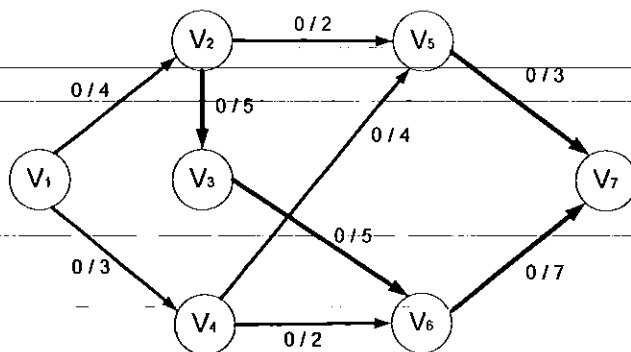
3. ขั้นจบการทำงาน

เมื่อไม่สามารถหาวิถีในเครือข่ายระดับที่เชื่อมต่อจาก  $v_s$  ไปยัง  $v_t$  ได้หรือไม่สามารถหาวิถีที่ทำให้  $\Delta(P_k) > 0$  ได้

**หมายเหตุ**

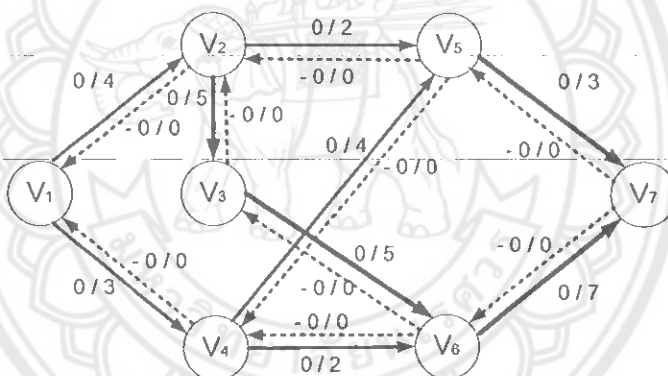
ในการค้นหาแบบแนวกว้าง (Breadth-first search) จะทำการหยุดค้นหาที่ต่อเมื่อในแขนง (Branch) ของต้นไม้ (Tree) มีใบ (leaf) ที่ซ้ำกับราก (root) ในแขนงนั้นๆ

ตัวอย่างที่ 6 ให้พิจารณาเครือข่ายเชิงเดียวที่วาดไว้ในรูปที่ 2.17

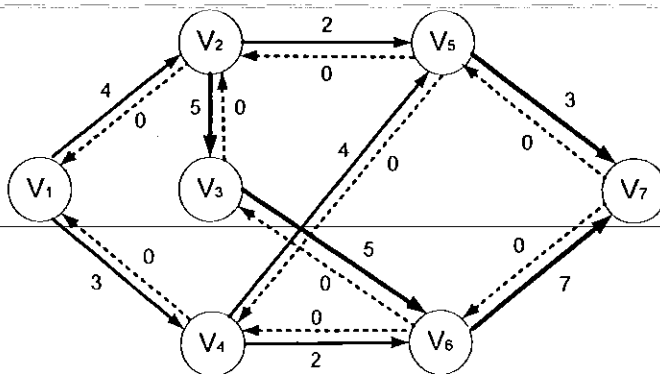


รูปที่ 2.17 แสดงเครือข่ายเชิงเดียวของขั้นตอนวิธีของ Dinic

จากเครือข่ายในรูปที่ 2.17 สามารถสร้างเครือข่ายเชิงเดียวที่มีอัตราการไหลเสมือนและเครือข่ายตกค้าง ให้ดังรูปที่ 2.18 และ รูปที่ 2.19 ตามลำดับ

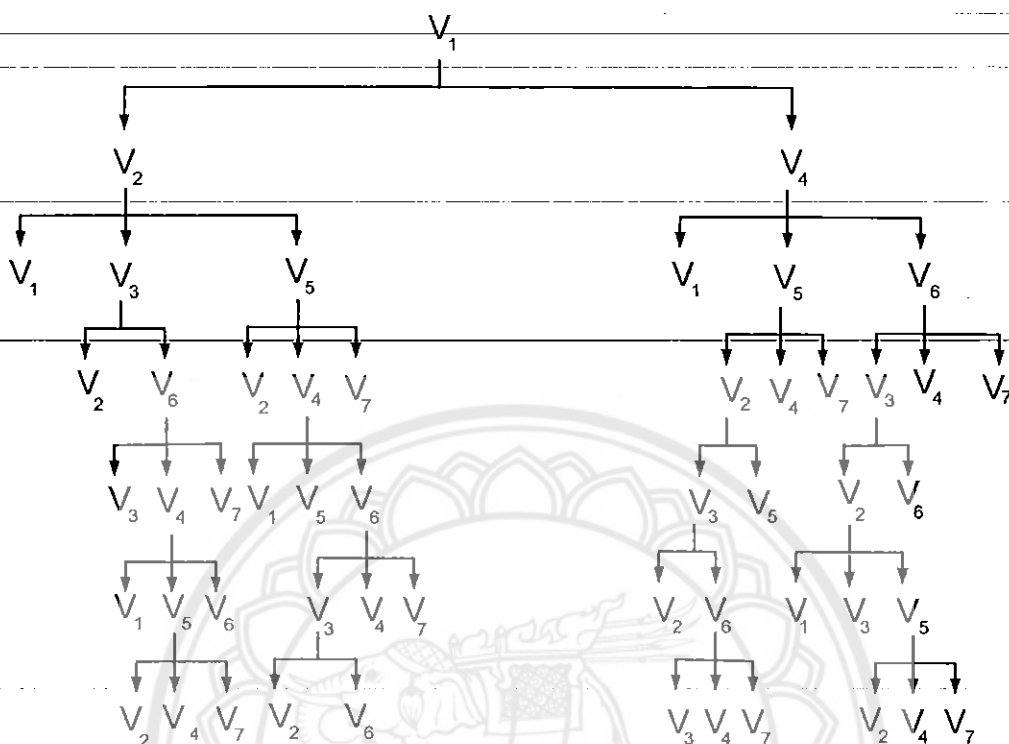


รูปที่ 2.18 แสดงเครือข่ายเชิงเดียวที่มีอัตราการไหลเสมือนของขั้นตอนวิธีของ Dinic



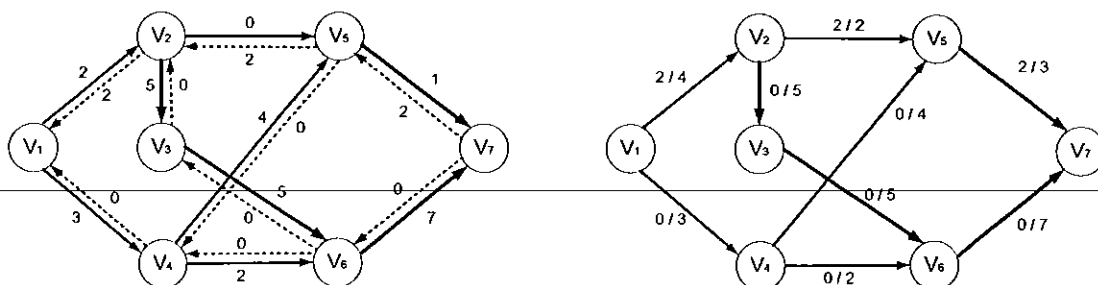
รูปที่ 2.19 แสดงเครือข่ายตกค้างของขั้นตอนวิธีของ Dinic

จากเครือข่ายตกค้างในรูปที่ 2.19 นำมาสร้างเครือข่ายระดับได้ดังนี้



รูปที่ 2.20 เครือข่ายระดับใช้ในการหาวิถีที่สั้นที่สุดครั้งที่ 1

เนื่องจากวิถี  $v_1, v_2, v_5, v_7$  เป็นวิถีที่สั้นที่สุดในเครือข่ายระดับ ดังนั้นจึงเลือกให้  $P_1$  คือวิถี  $v_1, v_2, v_5, v_7$  ซึ่งพบว่า  $\Delta(P_1) = 2$  ซึ่งหลังจากขั้นตอนวิธีของFord-Fulkerson ปรับอัตราการไหลสุทธิ หรือ ค่าความจุตกค้าง จะได้เครือข่ายตามรูปที่ 2.21

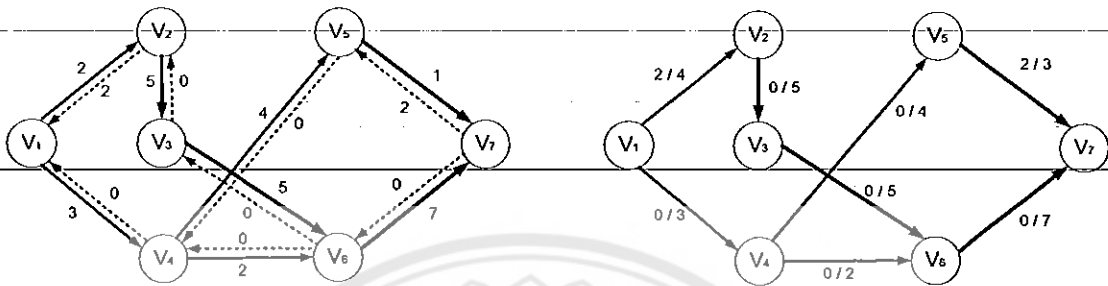


รูปที่ 2.21 แสดงเครือข่ายที่มีการปรับครั้งที่ 1 ของขั้นตอนวิธีของ Dinic

(รูปซ้ายมือ) แสดงเครือข่ายตกค้างที่ปรับค่าความจุตกค้าง

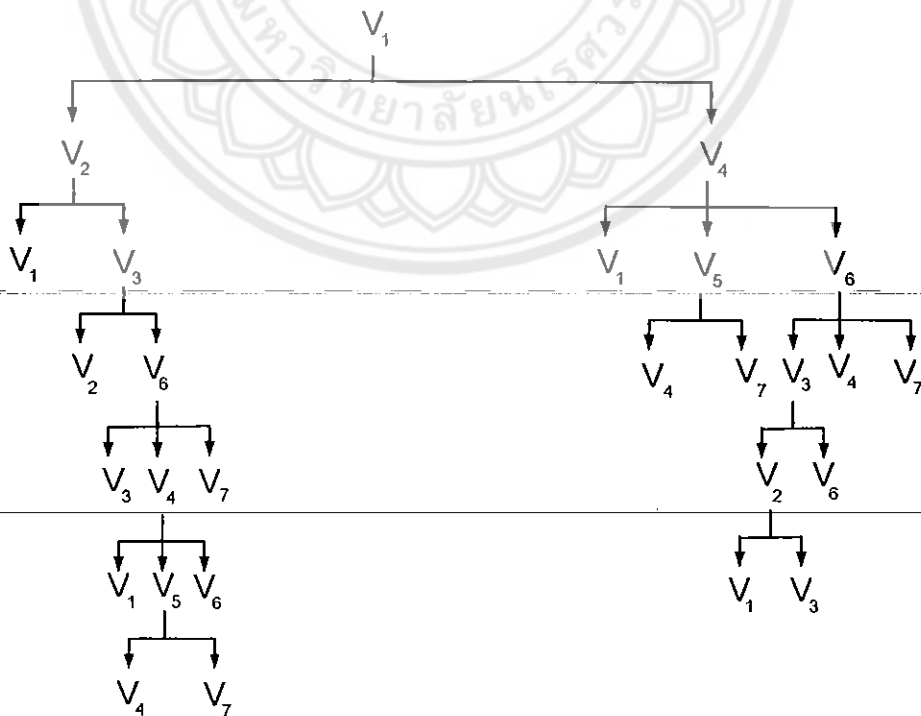
(รูปขวามือ) แสดงเครือข่ายเชิงเดียวที่ปรับอัตราการไหลสุทธิ

หลังจากที่ทำการปรับอัตราการไหลสุทธิแล้ว พบว่าอัตราการไหลสุทธิมีค่าเท่ากับค่าความจุบนส่วนโค้งจากบัพ  $v_2$  ไปยัง  $v_5$  ดังนั้นขั้นตอนวิธีจึงทำการขีดขวางอัตราการไหล (Blocking Flow) ระหว่างสองบัพดังกล่าวด้วยการลบส่วนโค้ง  $a_{2,5}$  ออกจากเครือข่ายดังแสดงในรูปที่ 2.22



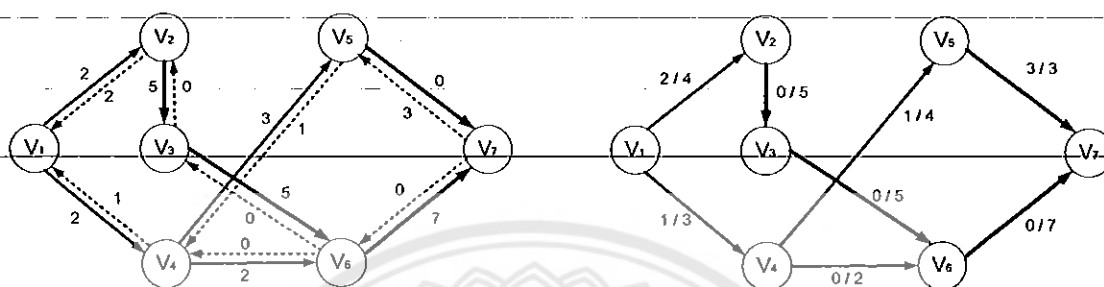
รูปที่ 2.22 แสดงเครือข่ายที่มีการขีดขวางอัตราการไหลสุทธิครั้งที่ 1 ของขั้นตอนวิธีของ Dinic (รูปซ้ายมือ) แสดงเครือข่ายตกค้าง (รูปขวามือ) แสดงเครือข่ายเชิงเดียว

จากเครือข่ายตกค้างที่ได้หลังทำการขีดขวางอัตราการไหลสุทธิแล้วนำมาคำนวณหาเครือข่ายระดับซึ่งจะได้ตามรูปที่ 2.23



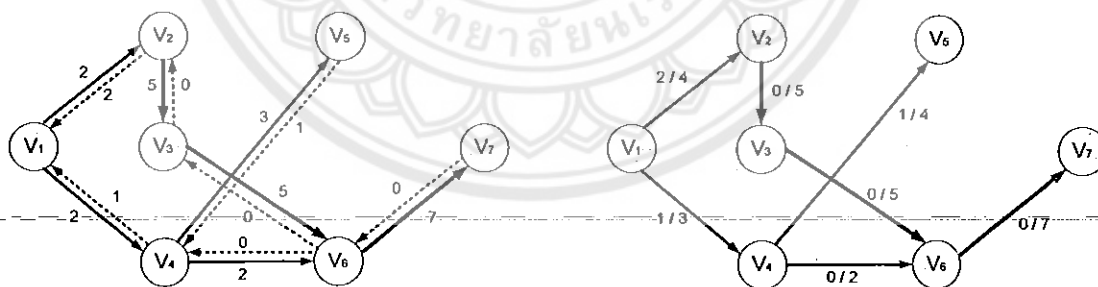
รูปที่ 2.23 เครือข่ายระดับใช้ในการหาวิถีที่สั้นที่สุดครั้งที่ 2

เลือก  $P_2$  แทนวิถี  $v_1, v_4, v_3, v_7$  เนื่องจากเป็นวิถีที่สั้นที่สุด ซึ่งมี  $\Delta(P_2) = 1$  และเมื่อทำการปรับอัตราการไหลสุทธิหรือค่าความจุตกค้างด้วยขั้นตอนวิธีของFord-Fulkersonแล้ว จะได้เครือข่ายตามรูปที่ 2.24



รูปที่ 2.24 แสดงเครือข่ายที่มีการปรับครั้งที่ 2 ของขั้นตอนวิธีของ Dinic (รูปซ้ายมือ) แสดงเครือข่ายตกค้างที่ปรับค่าความจุตกค้าง (รูปขวามือ) แสดงเครือข่ายเชิงเดี่ยวที่ปรับอัตราการไหลสุทธิ

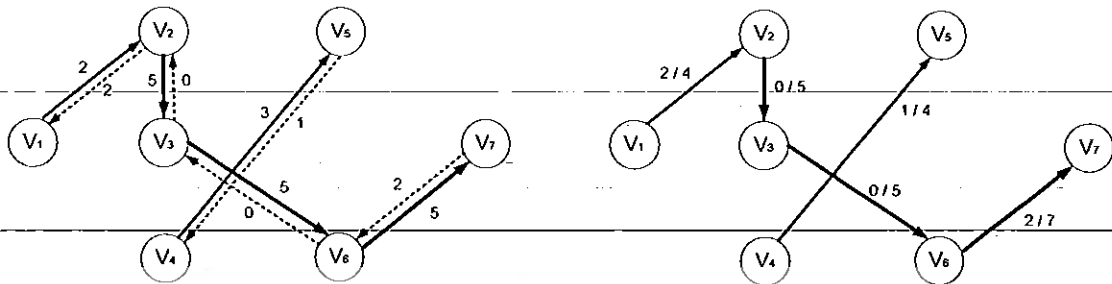
เมื่อทำการปรับอัตราการไหลสุทธิตามวิถี  $P_2$  เห็นได้ว่าค่าอัตราการไหลสุทธิมีค่าเท่ากับค่าความจุจากบัพ  $v_5$  ไปยังบัพ  $v_7$  ดังนั้นจึงทำการลบส่วนโค้ง  $a_{5,7}$  ออกจากเครือข่ายดังแสดงในรูปที่ 2.25



รูปที่ 2.25 แสดงเครือข่ายที่มีการตัดขวางอัตราการไหลสุทธิครั้งที่ 2 ของขั้นตอนวิธีของ Dinic (รูปซ้ายมือ) แสดงเครือข่ายตกค้าง (รูปขวามือ) แสดงเครือข่ายเชิงเดี่ยว

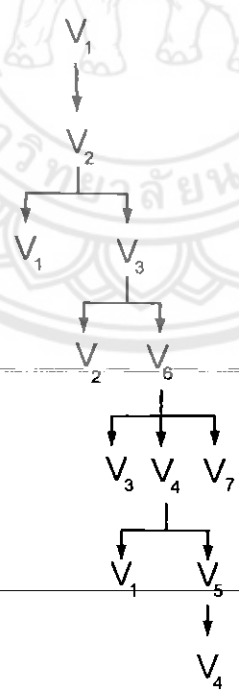


หลังจากที่ทำการปรับอัตราการไหลสุทธิแล้ว พบว่าส่วนโค้ง  $a_{4,6}$  และ  $a_{1,4}$  เป็นส่วนโค้งที่อิ่มตัว  
ขั้นตอนวิธีจึงทำการลบส่วนโค้งทั้งสองออกจากเครือข่าย ดังแสดงในรูปที่ 2.28



**รูปที่ 2.28** แสดงเครือข่ายที่มีการขัดขวางอัตราการไหลสุทธิครั้งที่ 3 ของขั้นตอนวิธีของ Dinic  
(รูปซ้ายมือ) แสดงเครือข่ายตกร้าง  
(รูปขวามือ) แสดงเครือข่ายเชิงเดียว

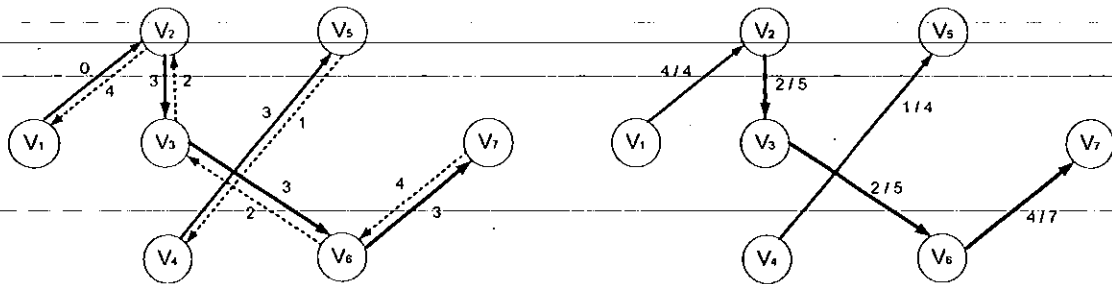
นำเครือข่ายตกร้างที่ได้มาหาเครือข่ายระดับได้ดังนี้



**รูปที่ 2.29** เครือข่ายระดับใช้ในการหาวิถีที่สั้นที่สุดครั้งที่ 3



สุดท้ายทำการเลือกวิถี  $P_4$  ซึ่งเป็นลำดับของ  $v_1, v_2, v_3, v_6, v_7$  โดยมี  $\Delta(P_4) = 2$  และเมื่อทำการปรับอัตราการไหลสุทธิ จะได้เครือข่ายตามรูปที่ 2.30



รูปที่ 2.30 แสดงเครือข่ายที่มีการปรับครั้งที่ 4 ของขั้นตอนวิธีของ Dinic

(รูปซ้ายมือ) แสดงเครือข่ายตกค้างที่รับค่าความจุตกค้าง

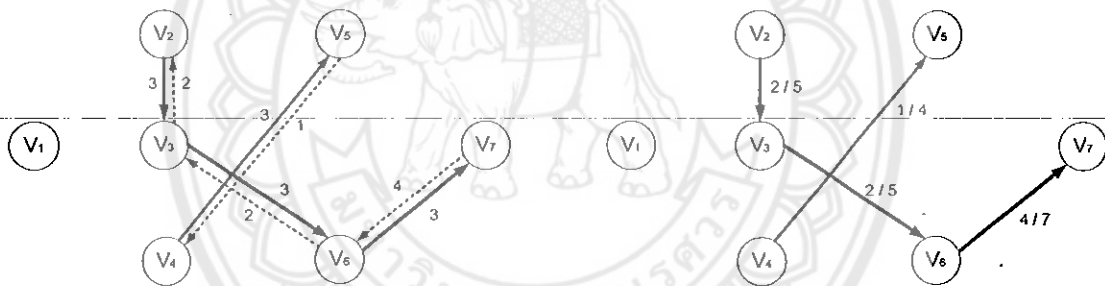
(รูปขวามือ) แสดงเครือข่ายเชิงเดียวที่ปรับอัตราการไหลสุทธิ

ผ/ร.

02667

2551

เนื่องจากส่วนโค้ง  $a_{1,2}$  เป็นส่วนโค้งอึดตัว ขั้นตอนวิธีของ Dinic ทำการขีดขวางอัตราการไหลสุทธิ โดยการลบส่วนโค้ง  $a_{1,2}$  ออกจากเครือข่าย ดังแสดงในรูปที่ 2.31

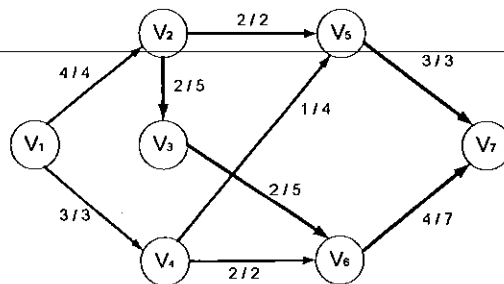


รูปที่ 2.31 แสดงเครือข่ายที่มีการขีดขวางอัตราการไหลสุทธิครั้งที่ 4 ของขั้นตอนวิธีของ Dinic

(รูปซ้ายมือ) แสดงเครือข่ายตกค้าง

(รูปขวามือ) แสดงเครือข่ายเชิงเดียว

เนื่องจากไม่สามารถหาวิถีที่เริ่มจากแหล่งต้นทาง  $v_1$  ไปยังแหล่งปลายทาง  $v_7$  ที่ซึ่ง  $\Delta > 0$  ดังนั้นจึงจบการทำงานและได้เครือข่ายดังแสดงในรูปที่ 2.32



รูปที่ 2.32 แสดงเครือข่ายที่มีการปรับอัตราการไหลสุทธิเสร็จสิ้นแล้ว ของขั้นตอนวิธีของ Dinic

จากเครือข่ายที่ได้หลังจากมีการปรับอัตราการไหลสุทธิพบว่าอัตราการไหลสูงสุดเท่ากับ 7 ซึ่งมีค่าเท่ากับค่าความจุต่ำสุดบริเวณจุดตัด  $K$  โดยที่จุดตัด  $K$  คือ

$$K = (S, \bar{S}) = \{(v_1, v_2), (v_1, v_4)\}$$

ซึ่งสอดคล้องกับการคำนวณโดยใช้ขั้นตอนวิธี Ford-Fulkerson

### 2.2.4 ขั้นตอนวิธีของ Augmenting Flow Improvement

แนวความคิดของขั้นตอนวิธีของ Augmenting Flow Improvement นั้นมีรากฐานมาจากขั้นตอนวิธีของ Ford-Fulkerson แต่ไม่จำเป็นต้องอาศัยเครือข่ายตกค้าง หากแต่เปลี่ยนจากพิจารณาเครือข่ายในลักษณะของไดกราฟมาเป็นกราฟ เพื่อที่สามารถหาวิถีที่เป็นไปได้ในทุกทิศทางโดยไม่ต้องคำนึงถึงทิศทางของส่วนโค้ง โดยกราฟที่ใช้ในขั้นตอนวิธีนี้เรียกว่า **กราฟฐานมูล** ซึ่งมีนิยามดังนี้

ให้  $N(V, E)$  เป็นเครือข่ายเชิงเดียว แล้ว **กราฟฐานมูล (Underlying graph)** ของเครือข่าย  $N$  คือกราฟที่มีบัพเดียวกับบัพใน  $N$  หากแต่เปลี่ยนจากส่วนโค้ง  $a_{i,j}$  มาเป็นเส้นเชื่อม  $e_{i,j}$  แทน ในที่นี้ ใช้  $G_N$  แทนกราฟมูลฐานของ  $N$

ดังนั้นเมื่อกำหนดเครือข่าย  $N$  มาให้ ขั้นตอนแรกจึงต้องคำนวณหากราฟมูลฐาน  $G_N$  ซึ่งเป็นกราฟที่ไม่มีการระบุทิศทาง จากนั้น ทำการคำนวณหาวิถีที่เป็นไปได้ทั้งหมดในกราฟมูลฐาน โดยวิถีดังกล่าวมีบัพเริ่มต้นที่ แหล่งต้นทาง (บัพ  $v_1$ ) และมีบัพสุดท้ายที่ แหล่งปลายทาง (บัพ  $v_n$ ) วิถีที่หาจากกราฟมูลฐานนี้ เรียกว่า **วิถีคล้าย (Quasi-path)** ซึ่งใช้แทนด้วย  $P_k$  จากวิถีคล้ายที่ได้ ให้พิจารณาส่วนโค้ง  $a_{i,j}$  บนเครือข่าย  $N$  หากแนวทางเดินบนวิถีคล้าย สอดคล้องกับทิศทางของส่วนโค้ง นั่นคือ หากมีส่วนโค้ง  $a_{i,j} \in E(N)$  แล้ววิถีคล้าย มีการเดินจาก บัพ  $v_i$  ไปยังบัพ  $v_j$  จะเรียกแนวทางเดินของวิถีคล้ายบนส่วนโค้ง  $a_{i,j}$  นี้ว่า **วิถีไปข้างหน้า (Forward path)** แต่ถ้าหากแนวทางเดินบนวิถีคล้ายสวนทางกับทิศทางของส่วนโค้ง นั่นคือ หากมีส่วนโค้ง  $a_{i,j} \in E(N)$  แล้ววิถีคล้าย มีการเดินจาก บัพ  $v_j$  ไปยังบัพ  $v_i$  จะเรียกแนวทางเดินของวิถีคล้ายบนส่วนโค้ง  $a_{i,j}$  นี้ว่า **วิถีย้อนกลับ (Backward path)** ทำการคำนวณหาค่าความจุตกค้างของแต่ละส่วนโค้งตามแนววิถีคล้าย  $P_k$  จากเงื่อนไขต่อไปนี้

$$\Delta(a_{i,j}) := \Delta_{i,j} = \begin{cases} C_{i,j} - F_{i,j}; & \text{ถ้า } a_{i,j} \text{ อยู่บนวิถีไปข้างหน้า} \\ F_{i,j}; & \text{ถ้า } a_{i,j} \text{ อยู่บนวิถีย้อนกลับ} \end{cases}$$

และเลือกค่าความจุตกค้างบนวิถีคล้ายทั้งวิถีจาก  $\Delta(P_k) = \min_{a_{i,j} \in P_k} \Delta_{i,j}$

หาก  $\Delta(P_k) = 0$  นั้นย่อมแสดงว่า อัตราการไหลสุทธิมีค่าเท่ากับค่าความจุบนบางส่วน ใ้คงตามแนวของวิถีคล้าย (นั่นคือตามวิถีไปข้างหน้า) หรือ อัตราการไหลสุทธิมีค่าเป็นศูนย์บนบางส่วน ใ้คงที่มีแนวของวิถีคล้ายทวนทิศทางของส่วน ใ้คง (นั่นคือตามวิถีย้อนกลับ) ซึ่งนั่นหมายถึงว่า วิถีคล้ายที่สร้างขึ้นมา ไม่สามารถปรับปรุงอัตราการไหลได้ ในกรณีดังกล่าวนี้ วิถีคล้ายที่สร้างขึ้นมา จะเรียกว่า **วิถีคล้ายที่อิ่มตัว (Saturated quasi-path)**

สำหรับวิถีคล้ายที่ไม่อิ่มตัว (Unsaturated quasi-path) คือวิถีที่สามารถปรับปรุงอัตราการไหลสุทธิได้ และถ้าให้  $P_{k,unsat}$  แทนวิถีคล้ายที่ไม่อิ่มตัว แล้ว อัตราการไหลสามารถปรับปรุงใหม่ได้จาก

$$\hat{F}_{i,j} = \begin{cases} F_{i,j} + \Delta(P_{k,unsat}) & ; \text{ ถ้า } a_{i,j} \text{ อยู่บนวิถีไปข้างหน้า ของ } P_{k,unsat} \\ F_{i,j} - \Delta(P_{k,unsat}) & ; \text{ ถ้า } a_{i,j} \text{ อยู่บนวิถีย้อนกลับ ของ } P_{k,unsat} \\ F_{i,j} & ; \text{ ถ้า } a_{i,j} \text{ ไม่อยู่บน } P_{k,unsat} \end{cases}$$

การปรับปรุงอัตราการไหลสุทธิจะดำเนินไปในทุก ๆ วิถีคล้ายที่เป็นไปได้ทั้งหมดในกราฟฐานมูลของเครือข่าย  $N$  จนกระทั่งไม่มีวิถีคล้ายใด ๆ ที่เป็น วิถีคล้ายที่ไม่อิ่มตัว แล้ว อัตราการไหลสุทธิที่ได้จะเป็นอัตราการไหลสุทธิสูงสุด หลักการคำนวณโดยขั้นตอนวิธีของ Augmenting Flow Improvement แสดงไว้ดังนี้

กำหนดให้  $N(V, E)$  เป็นเครือข่ายเชิงเดียวที่มี  $v_s$  เป็นแหล่งต้นทาง และ  $v_n$  เป็นแหล่งปลายทาง

#### 1. ขั้นเริ่มต้น (Initialization)

ทำการสร้างกราฟฐานมูลโดยเปลี่ยนจากส่วน ใ้คง  $a_{i,j}$  มาเป็นเส้นเชื่อม  $e_{i,j}$

#### 2. ขั้นทำซ้ำ(Iteration)

- ทำการหาวิถีคล้ายจากแหล่งต้นทาง  $v_s$  ไปยังแหล่งปลายทาง  $v_n$  จากกราฟฐานมูล
- ทำการหา  $\Delta_{i,j}$  ของแต่ละส่วน ใ้คงโดยที่

$$\Delta_{i,j} = \begin{cases} C_{i,j} - F_{i,j}; & \text{ ถ้า } a_{i,j} \text{ อยู่บนวิถีไปข้างหน้า} \\ F_{i,j}; & \text{ ถ้า } a_{i,j} \text{ อยู่บนวิถีย้อนกลับ} \end{cases}$$

- ทำการหาค่าต่ำสุดบนวิถีคล้าย  $\Delta(P_k) = \min_{a_{i,j} \in P_k} \Delta_{i,j}$

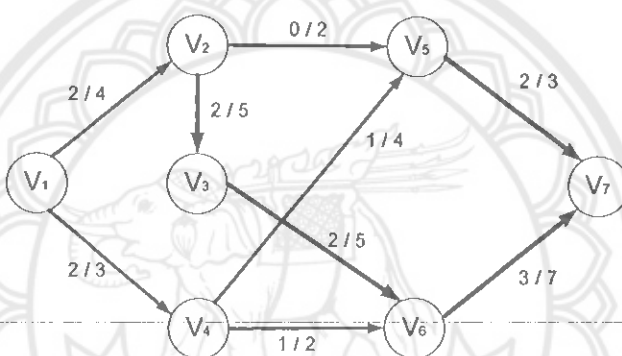
▪ ปรับปรุงอัตราการไหลสุทธิ

$$\hat{F}_{i,j} = \begin{cases} F_{i,j} + \Delta(P_{k,unsat}) & ; \text{ ถ้า } a_{i,j} \text{ อยู่บนวิถีไปข้างหน้า ของ } P_{k,unsat} \\ F_{i,j} - \Delta(P_{k,unsat}) & ; \text{ ถ้า } a_{i,j} \text{ อยู่บนวิถีย้อนกลับ ของ } P_{k,unsat} \\ F_{i,j} & ; \text{ ถ้า } a_{i,j} \text{ ไม่อยู่บน } P_{k,unsat} \end{cases}$$

3. ขั้นตอนการทำงาน

เมื่อไม่สามารถหาวิถีคล้ายใด ๆ ที่เป็น วิถีคล้ายที่ไม่อ้อมตัว

**ตัวอย่างที่ 7** ให้พิจารณาเครือข่ายอย่างง่ายในรูปที่ 2.33



**รูปที่ 2.33** แสดงตัวอย่างของเครือข่ายเชิงเดียวของขั้นตอนวิธี Augmenting Flow

จากรูปที่ 2.33 มีบัพที่เป็นแหล่งต้นทางคือ  $v_1$  และมีบัพที่เป็นแหล่งปลายทางคือ  $v_7$  หากเลือกวิถีคล้าย  $P_1$  ดังนี้  $P_1: v_1, v_2, v_5, v_7$  พบว่า แต่ละส่วนโค้งบนวิถีคล้ายดังกล่าวอยู่ในทิศทางเดียวกับทางเดินของวิถีคล้าย

$$\Delta_{1,2} = C_{1,2} - F_{1,2} = 4 - 2 = 2 \quad (\text{วิถีไปข้างหน้า})$$

$$\Delta_{2,5} = C_{2,5} - F_{2,5} = 2 - 0 = 2 \quad (\text{วิถีไปข้างหน้า})$$

$$\Delta_{5,7} = C_{5,7} - F_{5,7} = 3 - 2 = 1 \quad (\text{วิถีไปข้างหน้า})$$

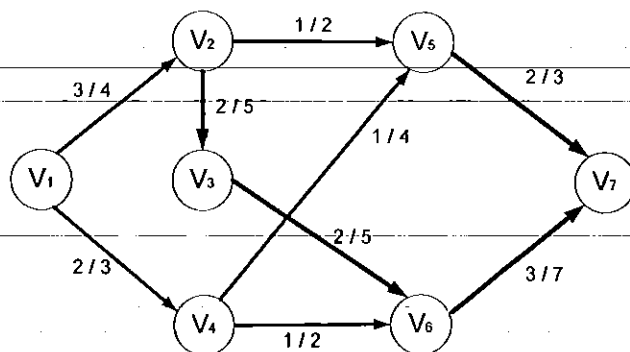
ในกรณีนี้ เลือก  $\Delta(P_1) = \min\{2, 2, 1\} = 1 \neq 0$  ดังนั้นวิถีคล้าย  $P_1$  เป็นวิถีที่ไม่อ้อมตัว ซึ่งสามารถทำการปรับปรุงได้ ดังนี้

$$\hat{F}_{1,2} = F_{1,2} + \Delta(P_1) = 2 + 1 = 3 \quad (\text{วิถีไปข้างหน้า})$$

$$\hat{F}_{2,5} = F_{2,5} + \Delta(P_1) = 0 + 1 = 1 \quad (\text{วิถีไปข้างหน้า})$$

$$\hat{F}_{5,7} = F_{5,7} + \Delta(P_1) = 2 + 1 = 3 \quad (\text{วิถีไปข้างหน้า})$$

ซึ่งผลจากการปรับปรุงอัตราการไหลสุทธิครั้งที่ 1 ตามแนววิถีคล้าย  $P_1$  แสดงไว้ในรูปที่ 2.34



รูปที่ 2.34 แสดงผลการปรับปรุงอัตราการไหลสุทธิครั้งที่ 1 ของขั้นตอนวิธี Augmenting Flow

ทำการเลือกวิถีคล้าย  $P_2$  ดังนี้  $P_2 : v_1, v_4, v_6, v_3, v_2, v_5, v_7$  พบว่า ส่วนโค้ง  $a_{1,4}$   $a_{4,6}$   $a_{2,5}$  และ  $a_{5,7}$  มีทิศทางสอดคล้องกับแนวทางการเดินทางของวิถีคล้าย ดังนั้น วิถีที่ผ่านตามส่วนโค้งคือ วิถีไปข้างหน้า ในขณะที่ แนวทางการเดินทางของวิถีคล้ายผ่านส่วนโค้ง  $a_{3,6}$  และ  $a_{2,3}$  เป็นวิถีย้อนกลับ เนื่องจากมีทิศทางตรงข้ามกับทิศทางของส่วนโค้ง ดังนั้นค่าความจุตกค้างบนแต่ละส่วนโค้งในวิถี  $P_2$  มีดังนี้

$$\Delta_{1,4} = C_{1,4} - F_{1,4} = 3 - 2 = 1 \quad (\text{วิถีไปข้างหน้า})$$

$$\Delta_{4,6} = C_{4,6} - F_{4,6} = 2 - 1 = 1 \quad (\text{วิถีไปข้างหน้า})$$

$$\Delta_{3,6} = F_{3,6} = 2 \quad (\text{วิถีย้อนกลับ})$$

$$\Delta_{2,3} = F_{2,3} = 2 \quad (\text{วิถีย้อนกลับ})$$

$$\Delta_{2,5} = C_{2,5} - F_{2,5} = 2 - 1 = 1 \quad (\text{วิถีไปข้างหน้า})$$

$$\Delta_{5,7} = C_{5,7} - F_{5,7} = 3 - 2 = 1 \quad (\text{วิถีไปข้างหน้า})$$

เนื่องจาก  $\Delta(P_2) = \min \{1, 1, 2, 2, 1, 1\} = 1 \neq 0$  ดังนั้นวิถีคล้าย  $P_2$  จึงเป็นวิถีที่ไม่อิ่มตัว ซึ่งสามารถทำการปรับปรุงได้ ดังนี้

$$\hat{F}_{1,4} = F_{1,4} + \Delta(P_2) = 2 + 1 = 3 \quad (\text{วิถีไปข้างหน้า})$$

$$\hat{F}_{4,6} = F_{4,6} + \Delta(P_2) = 1 + 1 = 2 \quad (\text{วิถีไปข้างหน้า})$$

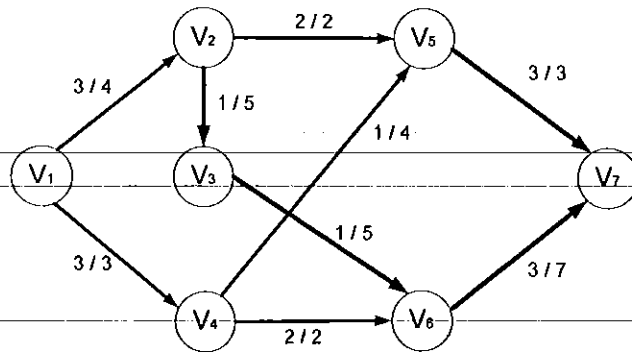
$$\hat{F}_{3,6} = F_{3,6} - \Delta(P_2) = 2 - 1 = 1 \quad (\text{วิถีย้อนกลับ})$$

$$\hat{F}_{2,3} = F_{2,3} - \Delta(P_2) = 2 - 1 = 1 \quad (\text{วิถีย้อนกลับ})$$

$$\hat{F}_{2,5} = F_{2,5} + \Delta(P_2) = 1 + 1 = 2 \quad (\text{วิถีไปข้างหน้า})$$

$$\hat{F}_{5,7} = F_{5,7} + \Delta(P_2) = 2 + 1 = 3 \quad (\text{วิถีไปข้างหน้า})$$

ซึ่งผลจากการปรับปรุงอัตราการไหลสุทธิครั้งที่ 2 ตามแนววิถีคล้าย  $P_2$  แสดงไว้ในรูปที่ 2.35



รูปที่ 2.35 แสดงผลการปรับปรุงอัตราการไหลสุทธิครั้งที่ 2 ของขั้นตอนวิธี Augmenting Flow

เลือกวิถีคล้าย  $P_3$  ดังนี้  $P_3 : v_1, v_2, v_3, v_6, v_7$  พบว่า แต่ละส่วนโค้งบนวิถีคล้ายดังกล่าวอยู่ในทิศทางเดียวกับทางเดินของวิถีคล้าย ดังนั้นจึงพบว่า

$$\Delta_{1,2} = C_{1,2} - F_{1,2} = 4 - 3 = 1 \quad (\text{วิถีไปข้างหน้า})$$

$$\Delta_{2,3} = C_{2,3} - F_{2,3} = 5 - 1 = 4 \quad (\text{วิถีไปข้างหน้า})$$

$$\Delta_{3,6} = C_{3,6} - F_{3,6} = 5 - 1 = 4 \quad (\text{วิถีไปข้างหน้า})$$

$$\Delta_{6,7} = C_{6,7} - F_{6,7} = 7 - 3 = 4 \quad (\text{วิถีไปข้างหน้า})$$

ในกรณีนี้ เลือก  $\Delta(P_3) = \min\{1, 4, 4, 4\} = 1 \neq 0$  ดังนั้นวิถีคล้าย  $P_3$  จึงเป็นวิถีที่ไม่อิ่มตัว จึงสามารถทำการปรับปรุงได้ ดังนี้

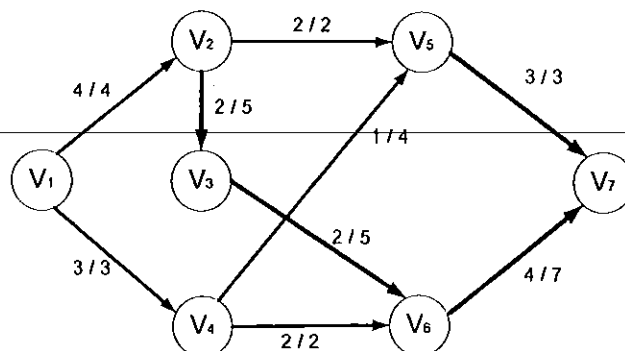
$$\hat{F}_{1,2} = F_{1,2} + \Delta(P_3) = 3 + 1 = 4 \quad (\text{วิถีไปข้างหน้า})$$

$$\hat{F}_{2,3} = F_{2,3} + \Delta(P_3) = 1 + 1 = 2 \quad (\text{วิถีไปข้างหน้า})$$

$$\hat{F}_{3,6} = F_{3,6} + \Delta(P_3) = 1 + 1 = 2 \quad (\text{วิถีไปข้างหน้า})$$

$$\hat{F}_{6,7} = F_{6,7} + \Delta(P_3) = 3 + 1 = 4 \quad (\text{วิถีไปข้างหน้า})$$

ซึ่งผลจากการปรับปรุงอัตราการไหลสุทธิครั้งที่ 2 ตามแนววิถีคล้าย  $P_3$  แสดงไว้ในรูปที่ 2.36



รูปที่ 2.36 แสดงผลการปรับปรุงอัตราการไหลสุทธิครั้งที่ 3 ของขั้นตอนวิธี Augmenting Flow

เนื่องจาก  $\Delta_{1,2} = 0 = \Delta_{1,4}$  จึงทำให้ไม่สามารถหาวิถีคล้ายใดในเครือข่ายที่ทำให้  $\Delta(P_k) > 0$  ได้ ดังนั้นจึงไม่สามารถปรับปรุงอัตราการไหลสุทธิได้อีก และอัตราการไหลสุทธิที่ได้จึงเป็นอัตราการไหลสูงสุด นั่นคือ

$$Val(F) = F^+(v_1) = 4 + 3 = 7$$

จากตัวอย่างที่ 7 หากกำหนดให้เซตของแหล่งต้นทาง  $S = \{v_1\}$  และเซตของแหล่งปลายทาง  $\bar{S} = \{v_2, v_3, v_4, v_5, v_6, v_7\}$  พบว่า  $F_{1,2} = C_{1,2} = 4$  และ  $F_{1,4} = C_{1,4} = 3$  ซึ่งสอดคล้องกับเงื่อนไขของทฤษฎีบท Maximum Flow/Minimum Cut ในหน้า 14 นั่นคือ

$$Val(F) = C(K) = 4 + 3 = 7$$

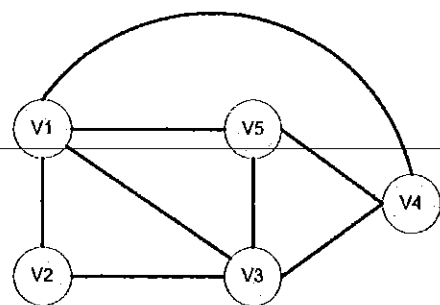
ดังนั้นอัตราการไหลที่ได้จึงเป็นอัตราการไหลสูงสุด

## 2.3 เมทริกซ์และกราฟ

วิธีที่นิยมใช้ในการแก้ปัญหากลศาสตร์ คือการแทนกราฟให้อยู่ในรูปของเมทริกซ์ ซึ่งมีความสะดวกต่อการคำนวณ วิธีการแทนกราฟด้วยเมทริกซ์นั้นมีหลายวิธี ซึ่งในหัวข้อนี้จะนำเสนอบางวิธีที่เกี่ยวข้องกับโครงการ

### 2.3.1 การแทนกราฟหรือไดกราฟด้วยเมทริกซ์

ให้  $G = (V, E)$  เป็นกราฟจำกัดใด ๆ ที่มีจำนวน  $n$  บัพ กราฟดังกล่าวสามารถเขียนแทนด้วยเมทริกซ์ขนาด  $n \times n$  โดยที่ดัชนี (Index) ของเมทริกซ์ที่ระบุในแนวตั้งและแนวนอน คือตำแหน่งของบัพแต่ละบัพ สมาชิกในเมทริกซ์ดังกล่าวที่ตำแหน่ง  $(i, j)$  ใดๆ คือ จำนวนเส้นเชื่อมหรือส่วนโค้งที่เชื่อมจากบัพ  $v_i$  ไปยังบัพ  $v_j$  ดังแสดงในตัวอย่างต่อไปนี้

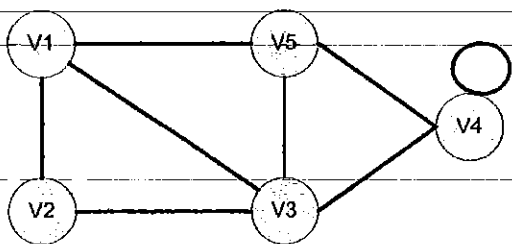


เขียนแทนด้วยเมทริกซ์ได้เป็น

$$\begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

รูปที่ 2.37 กราฟไม่ระบุทิศทาง หรือ กราฟ

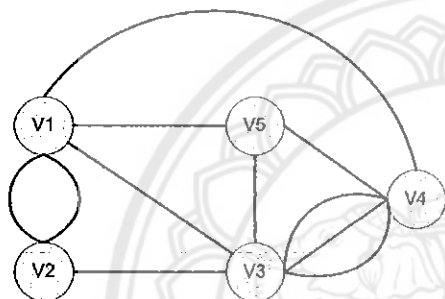
เขียนแทนด้วยเมทริกซ์ได้เป็น



$$\begin{matrix}
 & v_1 & v_2 & v_3 & v_4 & v_5 \\
 v_1 & \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \end{bmatrix} \\
 v_2 & \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix} \\
 v_3 & \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \end{bmatrix} \\
 v_4 & \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \end{bmatrix} \\
 v_5 & \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \end{bmatrix}
 \end{matrix}$$

รูปที่ 2.38 กราฟไม่ระบุทิศทางที่มีวงวน

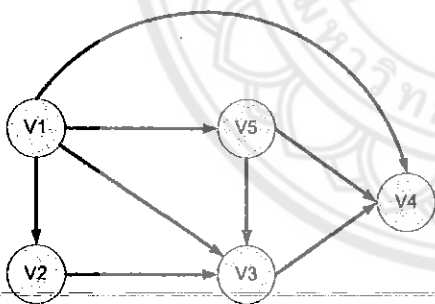
เขียนแทนด้วย เมทริกซ์ได้เป็น



$$\begin{matrix}
 & v_1 & v_2 & v_3 & v_4 & v_5 \\
 v_1 & \begin{bmatrix} 0 & 2 & 1 & 1 & 1 \end{bmatrix} \\
 v_2 & \begin{bmatrix} 2 & 0 & 1 & 0 & 0 \end{bmatrix} \\
 v_3 & \begin{bmatrix} 1 & 1 & 0 & 3 & 1 \end{bmatrix} \\
 v_4 & \begin{bmatrix} 1 & 0 & 3 & 0 & 1 \end{bmatrix} \\
 v_5 & \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \end{bmatrix}
 \end{matrix}$$

รูปที่ 2.39 กราฟหลายเชิง

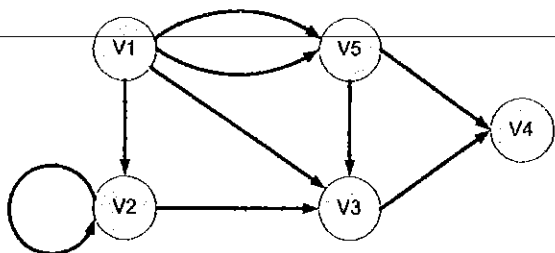
เขียนแทนด้วยเมทริกซ์ได้เป็น



$$\begin{matrix}
 & v_1 & v_2 & v_3 & v_4 & v_5 \\
 v_1 & \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \end{bmatrix} \\
 v_2 & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \\
 v_3 & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\
 v_4 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 v_5 & \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \end{bmatrix}
 \end{matrix}$$

รูปที่ 2.40 ไคกราฟเชิงเดียว

เขียนแทนด้วยเมทริกซ์ได้เป็น



$$\begin{matrix}
 & v_1 & v_2 & v_3 & v_4 & v_5 \\
 v_1 & \begin{bmatrix} 0 & 1 & 1 & 0 & 2 \end{bmatrix} \\
 v_2 & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \\
 v_3 & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\
 v_4 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 v_5 & \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \end{bmatrix}
 \end{matrix}$$

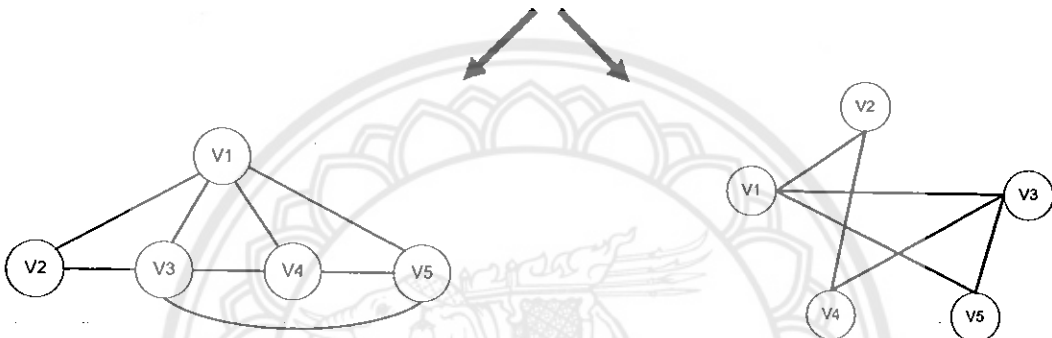
รูปที่ 2.41 ไคกราฟหลายเชิง



จากตัวอย่างข้างต้น เมื่อกำหนดกราฟหรือไดกราฟมาให้จะสามารถนำมาเขียนแทนได้ด้วยเมทริกซ์ ซึ่งเมทริกซ์ที่ได้นี้เรียกว่า **Adjacency Matrix** หรือ **เมทริกซ์ประชิด**

อย่างไรก็ตามหากกำหนดเมทริกซ์ประชิดมาให้สามารถนำมาเขียนกราฟได้หลายรูปแบบ ยกตัวอย่างเช่น

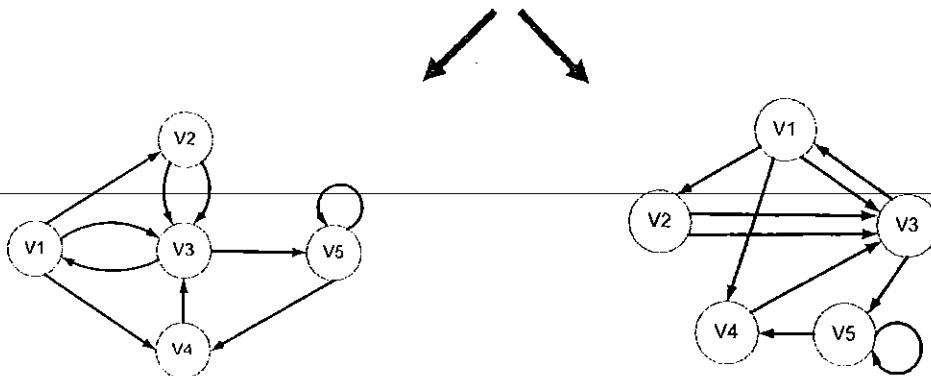
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$



รูปที่ 2.42 ตัวอย่างกราฟที่สมนัยกัน

หรือกรณีของไดกราฟเช่น

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$



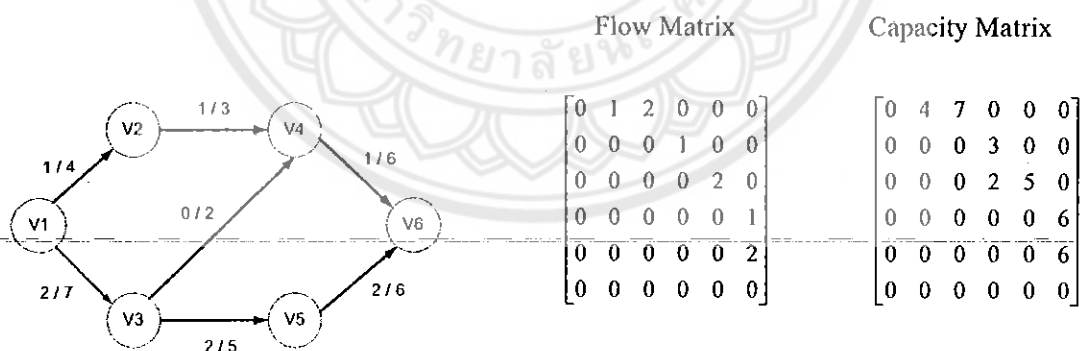
รูปที่ 2.43 ตัวอย่างไดกราฟที่สมนัยกัน

แม้ว่ากราฟที่ได้จะมีหลายรูปแบบก็ตามแต่ทุกกราฟจัดว่าเป็นกราฟสมสัณฐาน (Graph Isomorphism) นั่นคือ กราฟทุกกราฟเป็นกราฟเดียวกัน หากแต่ต่างกันตรงตำแหน่งการจัดวางหรือทอพอโลยี (topology) เท่านั้น

### 2.3.2 การแทนกราฟหรือไดกราฟที่มีฟังก์ชันค่าถ่วงน้ำหนักด้วยเมทริกซ์

ในกรณีของกราฟ(หรือไดกราฟ)เชิงเดียวที่มีฟังก์ชันค่าถ่วงน้ำหนักระบุบนเส้นเชื่อม (หรือส่วนโค้ง) สามารถเขียนแทนได้ด้วยเมทริกซ์เช่นกัน โดยที่สมาชิกของเมทริกซ์ที่ตำแหน่ง  $(i, j)$  คือ ค่าน้ำหนักถ่วงจากบัพ  $v_i$  ไปยังบัพ  $v_j$  และหากที่บัพ  $v_k$  และบัพ  $v_l$  ใดๆที่ไม่มีเส้นเชื่อม (หรือส่วนโค้ง) นั่นคือ  $a_{k,l} \notin E(G)$  ให้เขียนแทนสมาชิกที่ตำแหน่ง  $(k, l)$  ของเมทริกซ์ด้วย 0

สำหรับเครือข่ายเชิงเดียวมีการระบุฟังก์ชันค่าถ่วงน้ำหนัก 2 ฟังก์ชัน กล่าวคือ ฟังก์ชันอัตราการไหลสุทธิ ( $F_{i,j}$ ) และฟังก์ชันค่าความจุ ( $C_{i,j}$ ) ดังนั้นการแทนเครือข่ายเชิงเดียวด้วยเมทริกซ์นั้นจำเป็นต้องใช้เมทริกซ์ 2 ตัว โดยที่เมทริกซ์ตัวแรกเรียกว่า **เมทริกซ์อัตราการไหลสุทธิ (Flow Matrix)** ที่ใช้เก็บค่าอัตราการไหลสุทธิ และ **เมทริกซ์ค่าความจุ (Capacity Matrix)** ที่ใช้เก็บค่าความจุ ยกตัวอย่างเช่น



รูปที่ 2.44 ตัวอย่างการแทนเครือข่ายเชิงเดียวด้วยเมทริกซ์

### บทที่ 3

## ขั้นตอนวิธีการหาอัตราการไหลสูงสุด

เนื้อหาในบทที่ 2 มุ่งเน้นการศึกษาทฤษฎีพื้นฐานของเครือข่าย หลักการวิเคราะห์การหาอัตราการไหลสูงสุดในเครือข่าย รวมถึงการแทนเครือข่ายด้วยเมทริกซ์ ในบทนี้จะมุ่งศึกษาถึงขั้นตอนวิธีการหาอัตราการไหลสูงสุด (Maximum Flow Algorithm) ซึ่งจะนำเสนอในรูปแบบของรหัสเทียม (Pseudo-code) และแผนภาพการทำงาน (Flow Chart) โดยมุ่งหวังให้ผู้อ่านเกิดความเข้าใจในหลักการทำงานของแต่ละขั้นตอนวิธีอย่างลึกซึ้ง และสามารถนำไปประยุกต์ใช้ในการเขียนโปรแกรมได้อย่างถูกต้อง

กำหนดให้  $F_{i,j}$ ,  $C_{i,j}$ ,  $\Delta_{i,j}$  แทน สมาชิกตำแหน่งที่  $i, j$  ของเมทริกซ์อัตราการไหล ( $M(F)$ ) เมทริกซ์ค่าความจุ ( $M(C)$ ) และเมทริกซ์ค่าความจุตกค้าง ( $M(\Delta)$ ) ตามลำดับ และให้  $N_{i,j}$ ,  $R_{N,i,j}$ ,  $G_{N,i,j}$  แทน สมาชิกตำแหน่งที่  $i, j$  ของเมทริกซ์ประชิดของเครือข่ายเชิงเดียว ( $M(N)$ ) เมทริกซ์ประชิดของเครือข่ายตกค้าง ( $M(R_N)$ ) และเมทริกซ์ประชิดของกราฟมูลฐาน ( $M(G_N)$ ) ตามลำดับ โดยที่ให้  $n$  แทนจำนวนบัพทั้งหมดในเครือข่าย นอกจากนี้ทำการเปลี่ยนชื่อของแต่ละบัพในเครือข่ายซึ่งให้แหล่งต้นทาง (Source) =  $v_1$  และแหล่งปลายทาง (Sink) =  $v_n$

### 3.1 ขั้นตอนวิธีของ Ford-Fulkerson

ในหัวข้อนี้จะกล่าวถึงหลักการการทำงานของ Ford-Fulkerson ซึ่งเป็นแนวคิดที่ใช้เครือข่ายคงเหลือ (Residual network) ในการแก้ปัญหา โดยจะนำเสนอในรูปแบบของรหัสเทียม (Pseudo-code) และแผนผังการทำงาน (Flow chart) ดังนี้

**Initialization:** Set  $F_{i,j} = 0$ ;  $Stop = False$ ,  
Create the residual matrices ( $M(R_N)$ ,  $M(\Delta)$ ) by passing matrix  $M(N)$  to function "Residual" (See in section 3.1.1)

**Iteration:** While  $Stop = False$   
Find  $Path$  in residual graph  $R_N$  using function Findpath (See in section 3.1.2)

If  $Path \neq [v_1]$  Then  $\Delta(P_k) = \min_{a_{i,j} \in P_k} \Delta_{i,j}$

If  $\Delta(P_k) \neq 0$

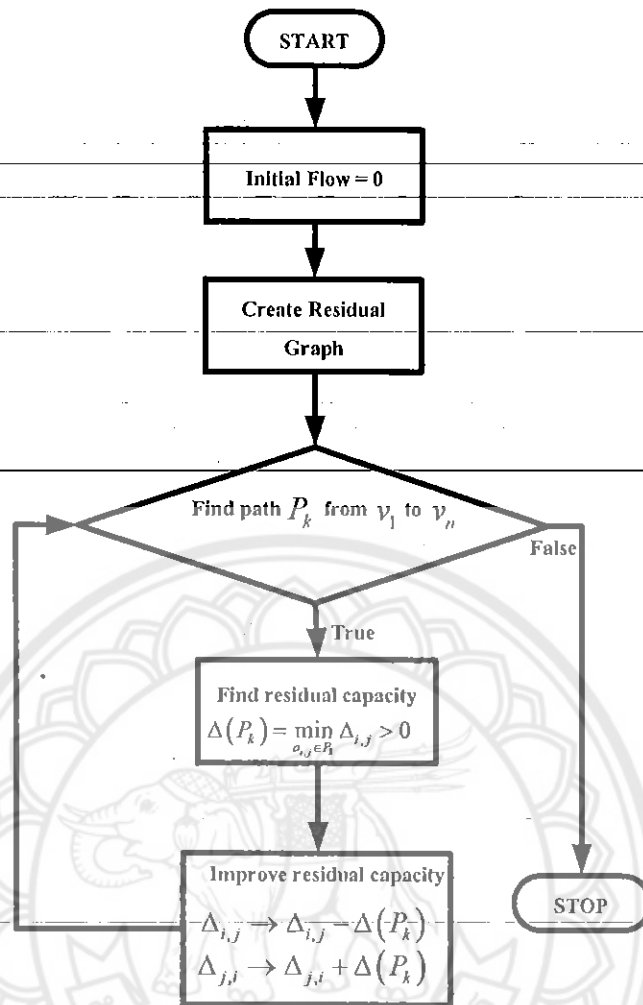
Improve residual network

$\Delta_{i,j} = \Delta_{i,j} - \Delta(P_k)$  % Decrement  $\Delta_{i,j}$  by  $\Delta(P_k)$

$\Delta_{j,i} = \Delta_{j,i} + \Delta(P_k)$  % Increment  $\Delta_{j,i}$  by  $\Delta(P_k)$  End

Else  $Stop = True$  End

End



รูปที่ 3.1 แผนภาพการทำงานของขั้นตอนวิธี Ford-Fulkerson

### 3.1.1 ขั้นตอนวิธีการหาเครือข่ายตกค้าง

จากในหัวข้อที่ 2.2 ได้กล่าวถึงหลักการสร้างเครือข่ายตกค้าง เพื่อช่วยในการคำนวณหาปริมาณอัตราการไหลสุทธิสูงสุดที่สามารถเพิ่มได้ในแต่ละวิถีโดยใช้ค่าความจุตกค้าง ในการสร้างเครือข่ายตกค้างนั้นอาศัยหลักการดังนี้

ถ้า  $N_{i,j} = 1$  แล้วกำหนดให้  $R_{N_{i,j}} = 1 = R_{N_{j,i}}$

หรือนั่นคือ  $M(R_N) = M(N) + M^T(N)$  นั่นเอง

เมื่อทำการสร้างเครือข่ายเป็นที่เรียบร้อยแล้วขั้นตอนต่อมาจะทำการสร้างความจุตกค้าง  $(\Delta_{i,j})$  บนส่วนโค้งให้กับเครือข่ายตกค้าง ซึ่งหลักการหาความจุตกค้างสามารถหาได้ดังนี้

1. ถ้า  $R_{N_{i,j}} = 1$  และ  $N_{i,j} = 1$  แล้ว  $\Delta_{i,j} = C_{i,j} - F_{i,j}$
2. ถ้า  $R_{N_{i,j}} = 1$  และ  $N_{i,j} = 0$  แล้ว  $\Delta_{i,j} = F_{j,i}$
3. ถ้า  $R_{N_{i,j}} = 0$  แล้ว  $\Delta_{i,j} = 0$

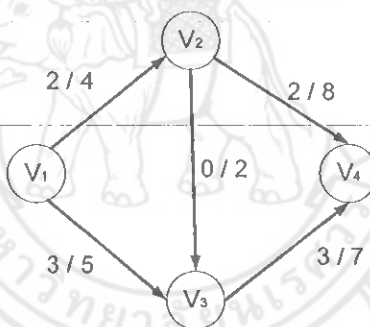
โดยการสร้างเครือข่ายดกค้างของขั้นตอน Ford-Fulkerson สามารถศึกษาได้จากรหัสเทียม (Pseudo-code) ดังนี้

```

Function [ $M(R_N), M(\Delta)$ ] = Residual ( $M(N)$ )
  For  $i = 1$  to  $n$ 
    For  $j = 1$  to  $n$ 
      If  $N_{i,j} = 1$  Then  $R_{N,i,j} = 1 = R_{N,j,i}$  End %Create residual matrix
      If  $R_{N,i,j} = 1$  and  $N_{i,j} = 1$  Then  $\Delta_{i,j} = C_{i,j} - F_{i,j}$ 
      Else If  $R_{N,i,j} = 1$  and  $N_{i,j} = 0$  Then  $\Delta_{i,j} = F_{j,i}$ 
      Else  $\Delta_{i,j} = 0$  End
    End
  End

```

**ตัวอย่างที่ 1** การสร้างเมทริกซ์ค่าความจุตกค้าง



รูปที่ 3.2 เครือข่ายเชิงเดียวสำหรับตัวอย่างที่ 1

พิจารณาเครือข่ายเชิงเดียว ( $N$ ) ในรูปที่ 3.2 ซึ่งมีเมทริกซ์ประชิด คือ

$$M(N) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

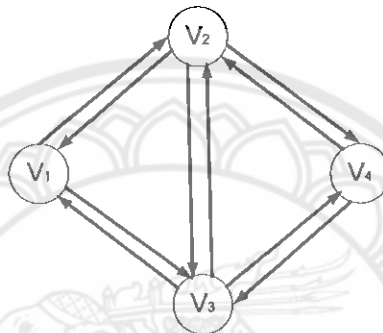
โดยที่เมทริกซ์อัตราการไหลสุทธิและเมทริกซ์ค่าความจุ สามารถหาได้ดังนี้

$$M(F) = \begin{bmatrix} 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad M(C) = \begin{bmatrix} 0 & 4 & 5 & 0 \\ 0 & 0 & 2 & 8 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

โดยที่ เมทริกซ์ประชิดของเครือข่ายตกค้าง  $M(R_N)$  สามารถคำนวณจากสูตร  $M(R_N) = M(N) + M^T(N)$  ซึ่งจะได้

$$M(R_N) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

จากเมทริกซ์ประชิดของเครือข่ายตกค้าง  $M(R_N)$  สามารถนำมาวาดเครือข่ายตกค้างได้ดังนี้



รูปที่ 3.3 เครือข่ายตกค้างสำหรับตัวอย่างที่ 1

โดยที่เมทริกซ์ค่าความจุตกค้าง  $M(\Delta)$  สามารถหาได้จากการพิจารณาในทีละตำแหน่ง  $(i, j)$  ในเมทริกซ์ประชิดของเครือข่ายตกค้าง  $M(R_N)$  ที่มีสมาชิกเป็น 1

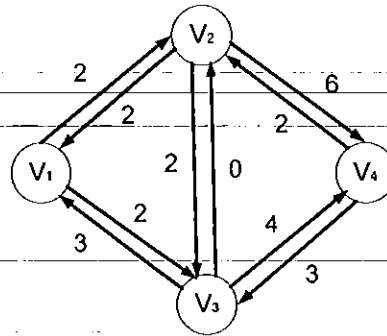
ยกตัวอย่างเช่น

- ที่ตำแหน่ง  $R_{N,1,1} = 0$  ดังนั้น  $\Delta_{1,1} = 0$
- ที่ตำแหน่ง  $R_{N,1,2} = 1$  และ  $N_{1,2} = 1$  ดังนั้น  $\Delta_{1,2} = C_{1,2} - F_{1,2} = 4 - 2 = 2$
- ที่ตำแหน่ง  $R_{N,2,1} = 1$  และ  $R_{N,2,1} - N_{2,1} = 1$  ดังนั้น  $\Delta_{2,1} = F_{1,2} = 2$

หลังจากกระทำงานเสร็จสิ้นกระบวนการแล้วสามารถสร้างเมทริกซ์ค่าความจุตกค้าง คือ

$$M(\Delta) = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 2 & 0 & 2 & 6 \\ 3 & 0 & 0 & 4 \\ 0 & 2 & 3 & 0 \end{bmatrix}$$

จากเมทริกซ์ค่าความจุตกค้างนำมาวาดเครือข่ายตกค้างได้ดังนี้



รูปที่ 3.4 เครือข่ายตกค้างที่มีค่าความจุตกค้างสำหรับตัวอย่างที่ 1

### 3.1.2 ขั้นตอนวิธีการหาวิถีบนเครือข่ายตกค้าง

จากเมทริกซ์  $M(R_N)$  ที่ได้ สามารถนำมาใช้หาวิถีจากแหล่งต้นทางไปยังแหล่งปลายทาง โดยอาศัยเมทริกซ์ประสิทธิของเครือข่ายตกค้าง โดยการหาวิถีดังกล่าวนี้สามารถหาได้หลายแบบ ขึ้นอยู่กับแนวทางการแก้ไขปัญหาของแต่ละผู้ใช้ ในโครงการจะเลือกวิธีการค้นหาโดยการพิจารณาบัพที่เชื่อมต่อกับแหล่งต้นทาง โดยคำนึงถึงค่าความจุตกค้างบนส่วนโค้งซึ่งต้องมีค่าความจุตกค้างไม่เท่ากับศูนย์ แล้วจึงทำการหาบัพที่เชื่อมต่อกับบัพนั้นไปเรื่อยๆ กรณีที่มีบัพที่เชื่อมต่อกว่าหนึ่งจะทำการเลือกบัพที่ตำแหน่งที่มีค่าน้อยกว่าก่อน หรือบัพที่สามารถเชื่อมต่อได้และไม่ทำการเลือกบัพที่ซ้ำกันในวิถี โดยจะทำการหยุดถัดเมื่อถึงบัพที่เชื่อมเป็นแหล่งปลายทาง

Function  $Path = \text{Findpath}(R_N, v_1, v_n, \Delta)$

**Initialization:**  $Path = [v_1], k = 1$

**Iteration:** While  $Stop = \text{False}$

While  $i_k \neq n$

$Count = 0;$

For  $j = 1$  to  $n$

If  $R(i_k, j) = 1$

If  $v_j \notin Path$

If  $\Delta \neq 0$  Then Push  $v_j$  in  $Path$

Increment  $k$  by 1

Set  $i_k = j$

Break;

Else Increment  $Count$  by 1    **End**

Else Increment  $Count$  by 1    **End**

**Else** Increment *Count* by 1 **End**

**End**

**If** *Count* = *n*

**If**  $i_k = S$

*Stop* = True

**Else**  $R_N(i_{k-1}, i_k) = 0$

*Path* =  $[v_1]$  ,  $k = 1$  **End**

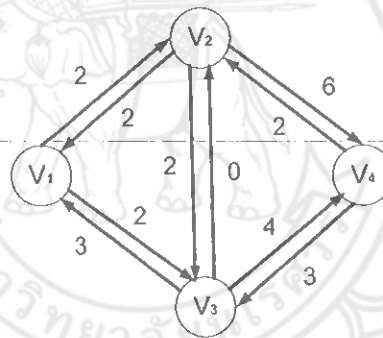
**End**

**End**

*Stop* = True

**End**

**ตัวอย่างที่ 2** วิธีการหาวิถีบนเครือข่ายตกค้าง



รูปที่ 3.5 เครือข่ายตกค้างสำหรับตัวอย่างที่ 2

จากรูปที่ 3.5 สามารถหาเมทริกซ์ประชิดเครือข่ายตกค้างได้ดังนี้

$$M(R_N) = \begin{matrix} & v_1 & v_2 & v_3 & v_4 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

ในการหาวิถีให้ดำเนินการดังนี้

เริ่มต้นทำการตรวจสอบบัพที่เชื่อมต่อกับแหล่งต้นทาง  $v_1$  โดยการตรวจสอบสมาชิกแถว 1 ว่ามีการเชื่อมกับบัพใดบ้าง จะเห็นได้ว่ามี 2 บัพที่เชื่อมต่อกับ  $v_1$  คือ  $v_2$  และ  $v_3$  ซึ่งในที่นี้เลือกบัพ  $v_2$  จากนั้นทำการตรวจสอบในแถว 2 เพื่อหาบัพที่เชื่อมต่อกับ  $v_2$  คือ  $v_1$ ,  $v_3$  และ  $v_4$  แต่โปรแกรมจะไม่ทำการเลือก  $v_1$  เนื่องจากเป็นบัพซ้ำกันซึ่งทำให้เกิดวงวน(loop) โปรแกรมจะทำการเลือก  $v_3$



จากนั้น โปรแกรมจะทำการค้นหาไปเรื่อยๆจนถึงแหล่งปลายทาง ซึ่งขั้นตอนวิธีนี้จะจบการทำงานเมื่อหาวิถีจนถึงแหล่งปลายทางในตัวอย่างนี้วิถีที่ได้คือ  $v_1, v_2, v_3, v_4$

### 3.2 ขั้นตอนวิธีของ Dinic

ในหัวข้อนี้จะกล่าวถึงหลักการการทำงานของ Dinic ซึ่งอาศัยแนวคิดที่ใช้วิธีแก้ปัญหาด้วยการใช้วิธีการขัดขวางอัตราการไหล (Blocking flow) ที่มีลักษณะพิเศษและเครือข่ายตกค้าง (Residual networks) ซึ่งมีวิธีการวิเคราะห์ที่คล้ายคลึงกับขั้นตอนวิธีของ Ford Fulkerson แต่ใช้เวลาในการคำนวณน้อยกว่ามาก เนื่องจากวิธีในการหาวิถีอาศัยหลักการที่เรียกว่าการค้นหาแบบแนวกว้าง (breadth first search : BFS) ดังแสดงในหัวข้อ 3.2.1

**Initialization:** Set  $F_{i,j} = 0$ ;  $Stop = False$

Create the residual matrices  $(M(R_N), M(\Delta))$  by passing matrix  $M(N)$  to function "Residual" (See in section 3.1.1)

**Iteration:** While  $Stop = False$

Find  $Path$  in residual graph  $R_N$  using function "BFS"  
(See in section 3.2.1)

If  $Path \neq [v_1]$

Find minimum  $(\Delta(P_k))$  in Path

$$\Delta(P_k) = \min_{a_{i,j} \in P_k} \Delta_{i,j} > 0$$

Improvement residual network

$$\Delta_{i,j} = \Delta_{i,j} - \Delta(P_k) \quad \% \text{ Decrement } \Delta_{i,j} \text{ by } \Delta(P_k)$$

$$\Delta_{j,i} = \Delta_{j,i} + \Delta(P_k) \quad \% \text{ Increment } \Delta_{j,i} \text{ by } \Delta(P_k)$$

In network

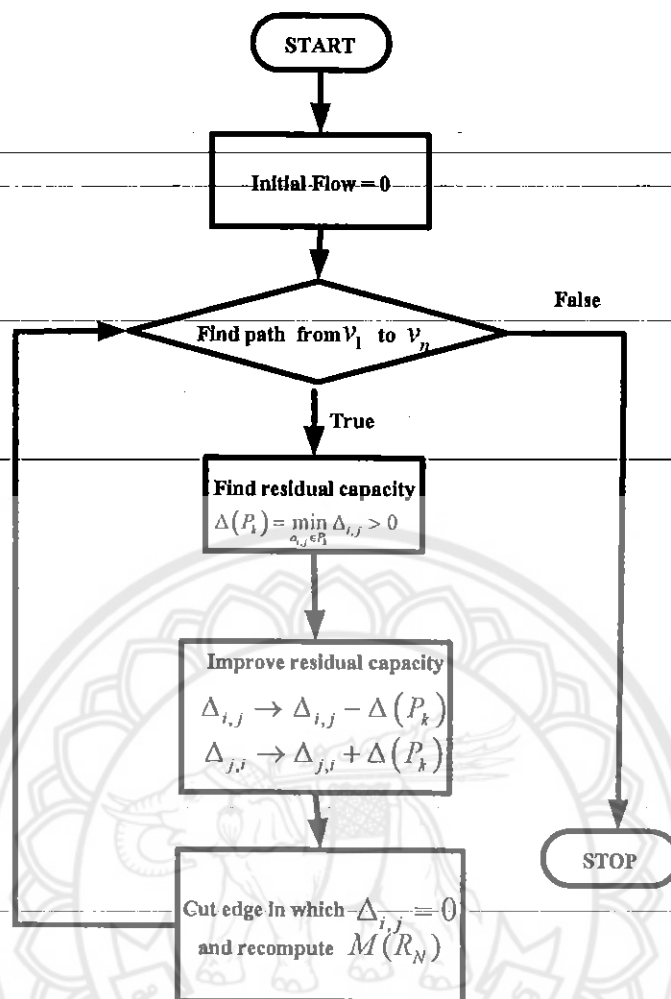
$$F_{i,j} = F_{i,j} + \Delta_{j,i} \quad \% \text{ Increment flow by } \Delta_{j,i}$$

If  $C_{i,j} = F_{i,j}$

Then Cut edge  $(i, j)$  by setting  $N_{i,j}, R_{N,i,j}, R_{N,j,i} = 0$  End

Else  $Stop = True$  End

End



รูปที่ 3.6 แผนภาพการทำงานของขั้นตอนวิธี Dinic

### 3.2.1 ขั้นตอนวิธีการหาวิถีบนเครือข่ายระดับแบบแนวกว้าง

เริ่มต้นโปรแกรมจะทำการสร้างเครือข่ายตักค้าง—(สามารถศึกษาวิธีการสร้างเครือข่ายตักค้างได้จากหัวข้อ 3.1.1) ซึ่งมีการเลือกวิถีที่สั้นที่สุดในการค้นหาแบบแนวกว้าง (breadth first search: BFS) จากแหล่งต้นทางไปยังแหล่งปลายทาง ซึ่งอาศัยหลักการดังนี้

**Function**  $[R_N, Path] = \text{BFS}(R_N, v_1, v_n, \Delta)$

**Initialization:**  $Path = [v_1]$ ,  $k = 1$ ,  $Travel = []$ ,  $Next = []$ ,  $Row = []$ ,  $Find = []$

**Iteration:** **While**  $Stop = \text{False}$

**While**  $k \neq n$

**For**  $j = 1$  to  $n$

**If**  $R_N(k, j) = 1$  and  $v_j \notin Path$

**Then** Enqueue  $v_j$  in *Next* **End**

**End**

Let  $l_N$  be the length of *Next*

**If**  $l_N \neq []$

**For**  $j = 1$  to  $l_N$

**If**  $Next(j) \neq v_n$

**If**  $\Delta \neq 0$  **Then** Push  $v_j$  in *Path*

Enqueue *Path* in *Find*

Pop  $v_j$  from *Path*

**Else**  $R_N(k, Next(j)) = 0$

**Else** Push  $v_j$  in *Path*

Enqueue *Path* in *Travel*

Break; **End**

**End**

**Else**  $l_N = []$

**If**  $Find = []$  **Then**  $Stop = True$  **End**

**End**

**If**  $Travel = []$  **Then** Dequeue the first vector from *Find*

And set *Path* to such a vector

$v_k =$  last element in *Path*

$Next = []$

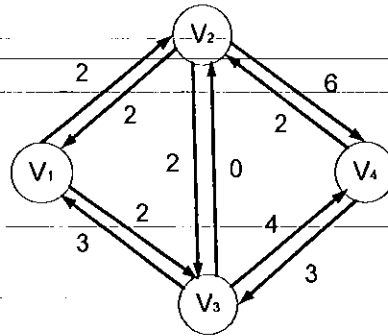
**Else**  $Path = Travel$

$Stop = True$  **End**

**End**

**End**

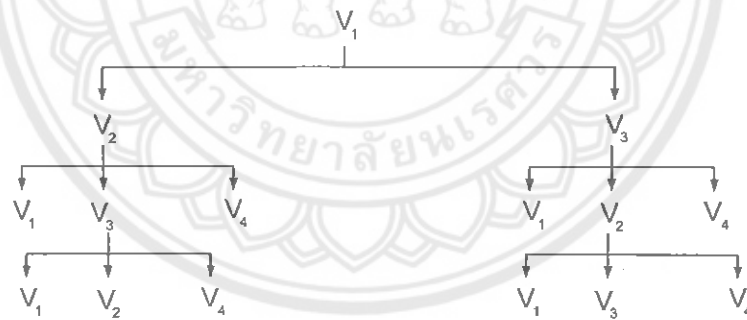
**ตัวอย่างที่ 3** การหาวิธีบนเครือข่ายระดับแบบแนวกว้าง



**รูปที่ 3.7** เครือข่ายตกล้างสำหรับตัวอย่างที่ 3

จากรูปที่ 3.7 สามารถหาเมทริกซ์ประชิดเครือข่ายตกล้างและกราฟต้นไม้แสดงการหาวิธีทั้งหมดจากแหล่งต้นทางไปยังแหล่งปลายทางได้ดังนี้

$$M(R_N) = \begin{matrix} & v_1 & v_2 & v_3 & v_4 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$



จากกราฟต้นไม้ที่ได้พบว่ามี 4 วิธีที่เป็นไปได้ คือ  $P_1 : v_1, v_2, v_4$

$$P_2 : v_1, v_3, v_4$$

$$P_3 : v_1, v_2, v_3, v_4$$

$$P_4 : v_1, v_3, v_2, v_4$$

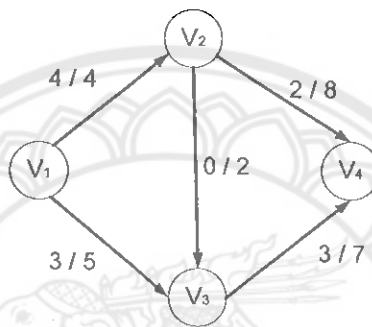
โดยโปรแกรมจะทำการเลือก  $P_1$  เป็นเส้นทางที่สั้นที่สุดจากแหล่งต้นทาง ( $v_1$ ) ไปยังแหล่งปลายทาง ( $v_4$ )

### 3.2.2 ขั้นตอนการขัดขวางอัตราการไหล

หลักการการขัดขวางอัตราการไหลจะทำการตรวจสอบส่วนโค้งในเครือข่ายที่ซึ่งมีอัตราการไหลสุทธิมีค่าเท่ากับค่าความจุ (Saturate) หรือ ค่าความจุตกค้างเป็นศูนย์ จากนั้นทำการขัดขวางอัตราการไหลของส่วนโค้ง โดยการลบส่วนโค้งนั้นๆ โดยอาศัยหลักการดังนี้

ที่ตำแหน่ง  $(i, j)$  ใดๆ ถ้า  $N_{i,j} = 1$  และ  $C_{i,j} = F_{i,j}$  แล้วให้ทำการปรับ  $N_{i,j} = 0$

#### - ตัวอย่างที่ 4 การขัดขวางอัตราการไหล



รูปที่ 3.8 เครือข่ายเชิงเดียวสำหรับตัวอย่างที่ 4

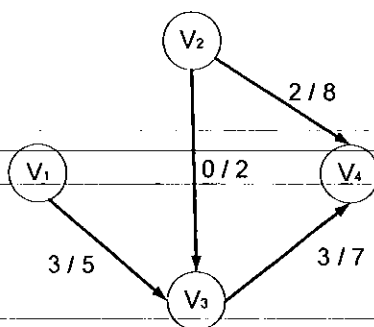
จากเครือข่ายเชิงเดียวในรูปที่ 3.8 พบว่าเมทริกซ์อัตราการไหลสุทธิ  $M(F)$  เมทริกซ์ค่าความจุ  $M(C)$  และเมทริกซ์ประชิด  $M(N)$  คือ

$$M(C) = \begin{bmatrix} 0 & 4 & 5 & 0 \\ 0 & 0 & 2 & 8 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad M(F) = \begin{bmatrix} 0 & 4 & 3 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad M(N) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

เนื่องจากที่ตำแหน่ง  $(1,2)$  พบว่า  $F_{1,2} = C_{1,2}$  ดังนั้นจึงทำการปรับให้  $N_{1,2} = 0$  ซึ่งจะได้เมทริกซ์ประชิด  $M(N)$  หลังจกการขัดขวางอัตราการไหล คือ

$$M(N) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

หรือมองในแง่เครือข่ายได้ตามรูปที่ 3.9



รูปที่ 3.9 เครือข่ายเชิงเดียวหลังจากการจัดขวางอัตราการใช้สำหรับตัวอย่างที่ 4

### 3.3 ขั้นตอนวิธีของ Augmenting Flow

ในหัวข้อนี้จะกล่าวถึงหลักการการทำงานของ Augmenting Flow เป็นแนวคิดที่ใช้กราฟมูลฐาน (Underlying graph) ในการแก้ปัญหา โดยจะนำเสนอในรูปแบบของรหัสเทียม (Pseudo-code) และแผนผังการทำงาน (Flow chart) ดังนี้

**Initialization:**  $Stop = False$ ,  $M = 1$ ,  $Travel = []$

Create the underlying matrix ( $M(G_N)$ ) and flow the residual matrix ( $M(\Delta)$ ) by passing matrix  $M(N)$  to function "Underlying" (See in section 3.3.1)

**Iteration:** **While**  $Stop = False$

Find  $Path$  in underlying graph  $G_N$  using function "Findpath"

(See in section 3.1.2)

**If**  $Path \neq [v_1]$

**If**  $G_{N,i,j} - N_{i,j} = 0$  **Then**  $\Delta_{i,j} = C_{i,j} - F_{i,j}$  **%Forward arc**

**Else**  $G_{N,i,j} - N_{i,j} = 1$  **Then**  $\Delta_{i,j} = F_{i,j}$  **End** **%Backward arc**

$\Delta(P_k) = \min_{a_{i,j} \in P_k} \Delta_{i,j}$  **%Find minimum ( $\Delta(P_k)$ ) in Path**

**If**  $\Delta(P_k) \neq 0$

Improve flow in network

**If**  $G_{N,i,j} - N_{i,j} = 0$  **%Forward arc**

**Then**  $\hat{F}_{i,j} = F_{i,j} + \Delta(P_k)$

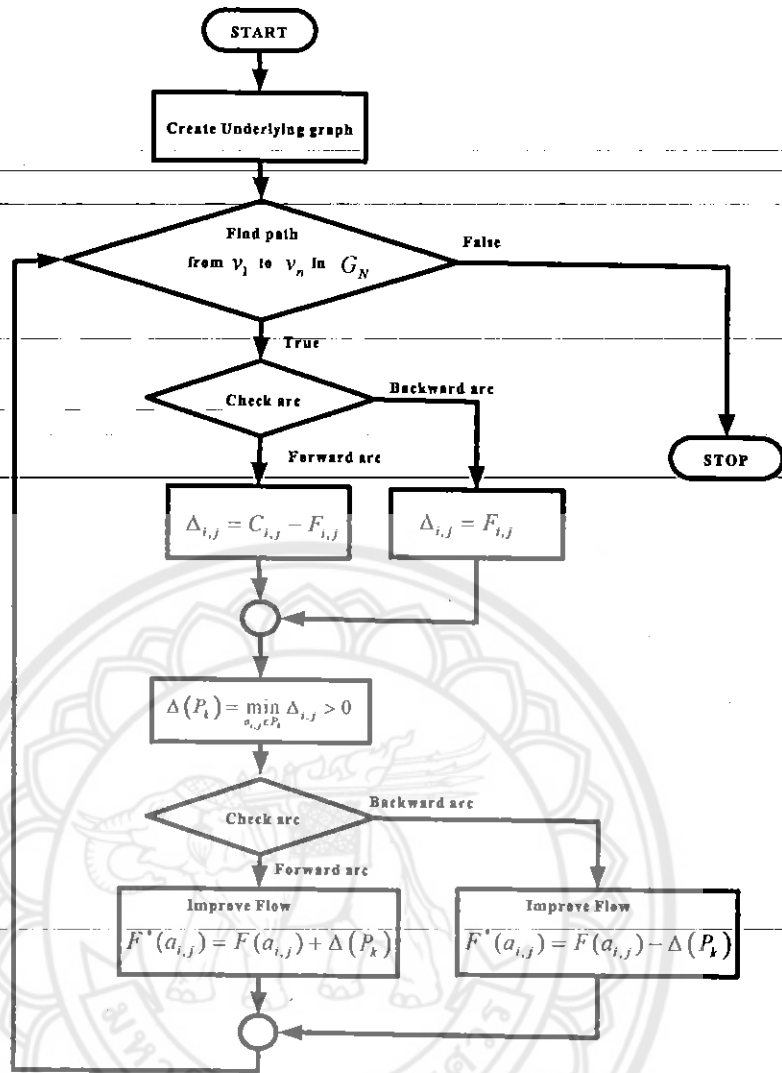
**Else**  $G_{N,i,j} - N_{i,j} = 1$  **%Backward arc**

**Then**  $\hat{F}_{i,j} = F_{i,j} - \Delta(P_k)$  **End**

**End**

**Else**  $Stop = True$  **End**

**End**



รูปที่ 3.10 แผนภาพการทำงานของขั้นตอนวิธี Augmenting Flow

### 3.3.1 ขั้นตอนวิธีการสร้างกราฟมูลฐาน

กราฟมูลฐานของเครือข่ายคือ กราฟที่มีบัพเดียวกับบัพใน  $N$  หากแต่เปลี่ยนจากส่วนโค้ง  $a_{i,j}$  มาเป็นเส้นเชื่อม  $e_{i,j}$  ซึ่งเขียนเป็นรหัสเทียม (Pseudo-code) ได้ดังต่อไปนี้

**Function**  $M(G_N) = \text{Underlying}(M(N))$

**For**  $i = 1$  to  $n$

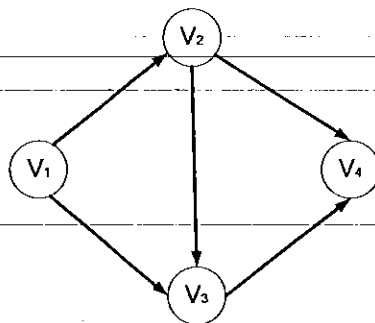
**For**  $j = 1$  to  $n$

**If**  $N_{i,j} = 1$  **then**  $G_{N,i,j} = 1 = G_{N,j,i}$  **End**

**End**

**End**

### ตัวอย่างที่ 5 การสร้างกราฟมูลฐาน



รูปที่ 3.11 เครื่องข่ายเชิงเดียวสำหรับตัวอย่างที่ 5

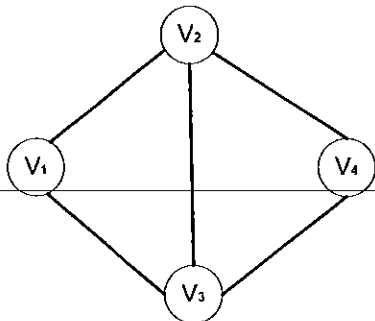
จากรูปที่ 3.11 สามารถหาเมทริกซ์ประชิด ได้ดังนี้

$$M(N) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

สร้างเมทริกซ์ประชิดของกราฟมูลฐาน ( $M(G_N)$ ) ได้เป็น

$$M(G_N) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

ซึ่งสมนัยกับกราฟในรูปที่ 3.12



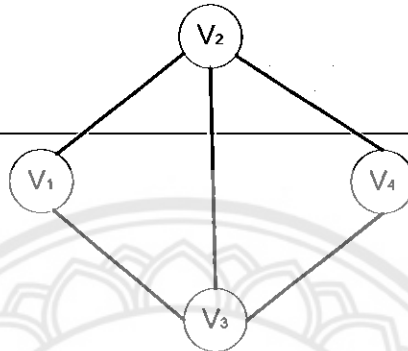
รูปที่ 3.12 กราฟมูลฐานสำหรับตัวอย่างที่ 5



### 3.3.2 ขั้นตอนการหาวิถีคล้ายบนกราฟมูลฐาน

เนื่องจากเมทริกซ์ประชิดของกราฟมูลฐาน ( $M(G_N)$ ) มีลักษณะเช่นเดียวกับเมทริกซ์ประชิดของเครือข่ายตกค้าง ( $M(R_N)$ ) ดังนั้นจึงสามารถใช้ขั้นตอนวิธีในหัวข้อที่ 3.1.2 ในการหาวิถีคล้ายบนกราฟมูลฐานได้

**ตัวอย่างที่ 6** การหาวิถีคล้ายบนกราฟมูลฐาน



รูปที่ 3.13 กราฟมูลฐานสำหรับตัวอย่างที่ 6

จากรูปที่ 3.14 สามารถนำมาเขียนเมทริกซ์ประชิดของกราฟมูลฐาน ได้ดังนี้

$$M(G_N) = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

จากเมทริกซ์จะได้

วิถีคล้าย  $P_1$  คือ  $v_1, v_2, v_3, v_4$

$P_2$  คือ  $v_1, v_2, v_4$

$P_3$  คือ  $v_1, v_3, v_2, v_4$

$P_4$  คือ  $v_1, v_3, v_4$

### 3.4 บทสรุป

ในบทนี้ได้กล่าวถึงโครงสร้างและขั้นตอนการทำงานของแต่ละขั้นตอนวิธี ซึ่งได้อธิบายการทำงานแบบรหัสเทียม(Pseudo-code) และแบบแผนภาพการทำงาน (Flow chart) ตลอดจนตัวอย่างของขั้นตอนต่างๆแต่ละขั้นตอนวิธี อาทิเช่น การสร้างเครือข่ายตกค้าง การสร้างความจุตกค้าง การจัดขวางอัตรการไหล การหาวิถี การปรับปรุงเครือข่าย เป็นต้น

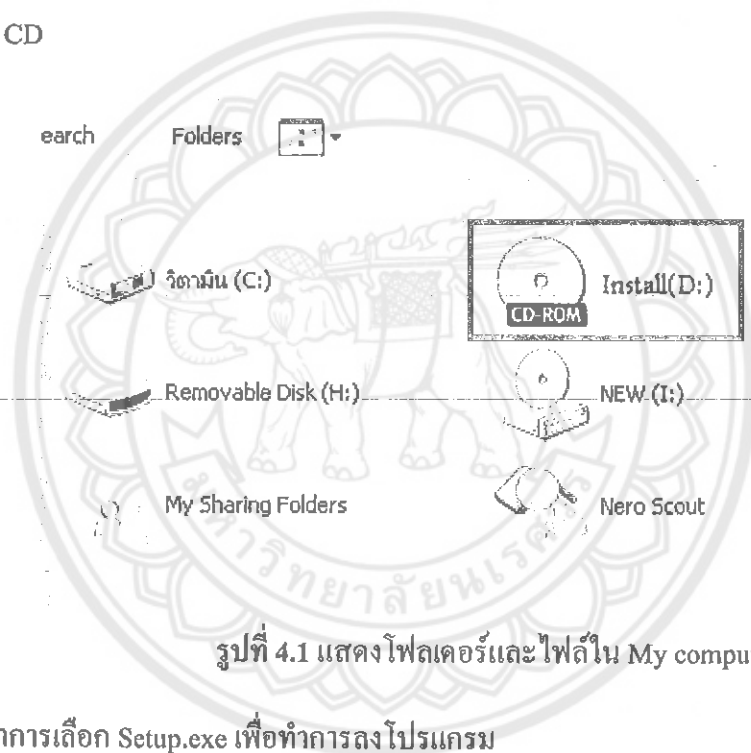
## บทที่ 4

# การใช้งานโปรแกรม

ในบทนี้เป็นการนำเสนอการประยุกต์ใช้งานของโปรแกรมที่พัฒนาขึ้น โดยเริ่มอธิบายจาก การติดตั้งโปรแกรมและวิธีการใช้งาน โปรแกรม อาทิ เช่น การสร้างบัพ การสร้างส่วนโค้ง การลบส่วนโค้ง การเลือกขั้นตอนวิธีในการคำนวณ เป็นต้น

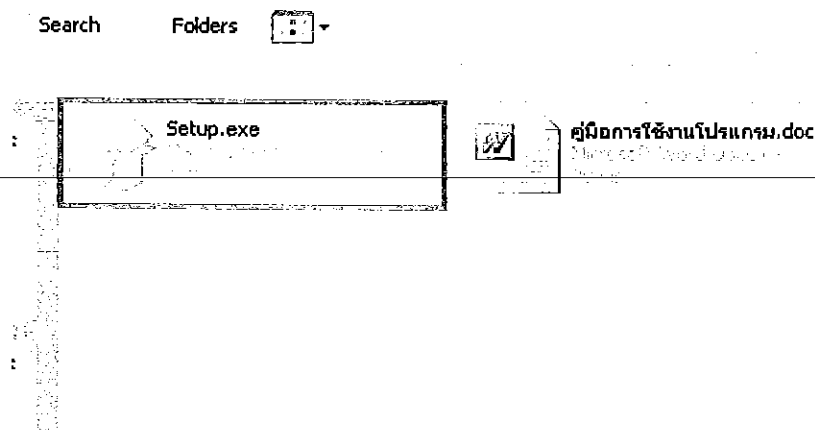
### 4.1 การติดตั้งโปรแกรม

นำแผ่น CD ใส่ในช่องสำหรับอ่าน CD จากนั้นให้ผู้ใช้งานเข้าไปที่ My computer แล้วทำการเปิดอ่านแผ่น CD



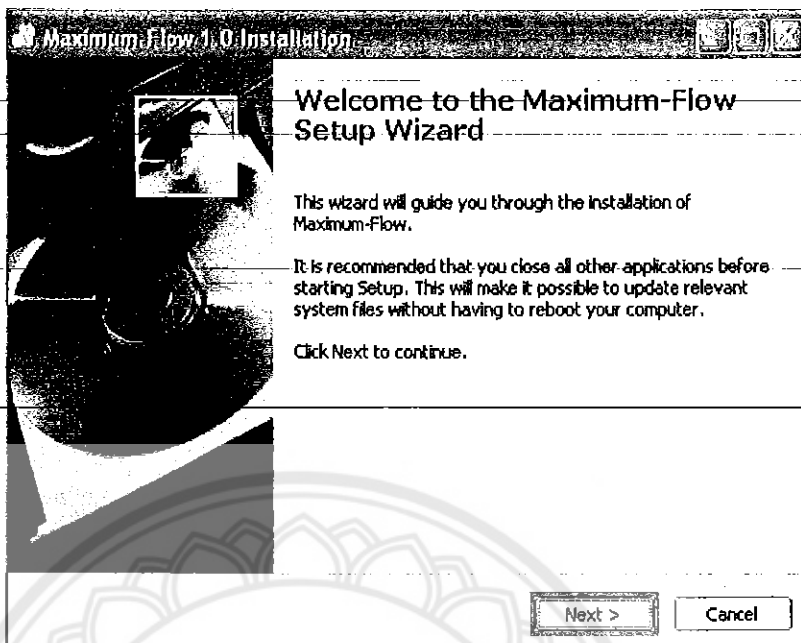
รูปที่ 4.1 แสดงโฟลเดอร์และไฟล์ใน My computer

จากนั้นทำการเลือก Setup.exe เพื่อทำการลงโปรแกรม



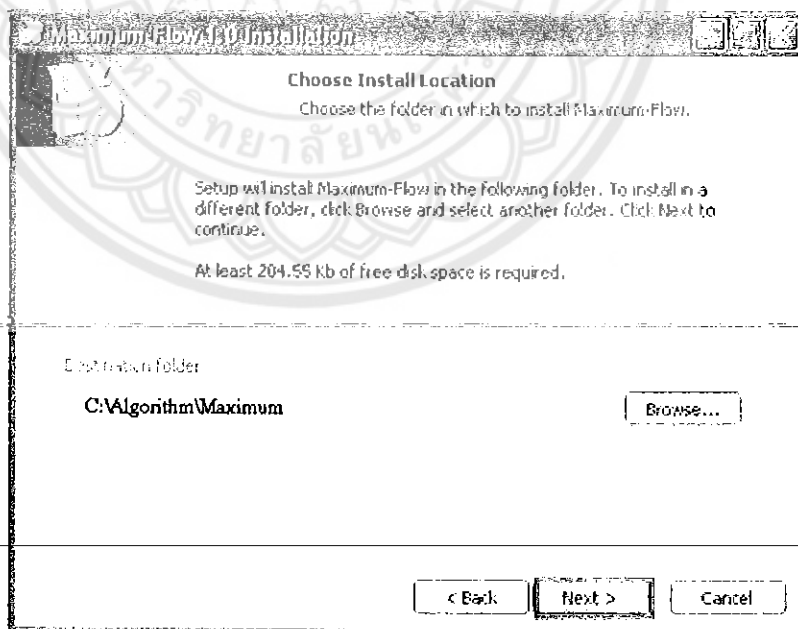
รูปที่ 4.2 แสดงไฟล์ที่ใช้ในการติดตั้ง

เมื่อปรากฏหน้าต่างรูปที่ 4.3 ให้กด Next เพื่อทำการติดตั้งโปรแกรม



รูปที่ 4.3 แสดงหน้าต่างการติดตั้งโปรแกรม

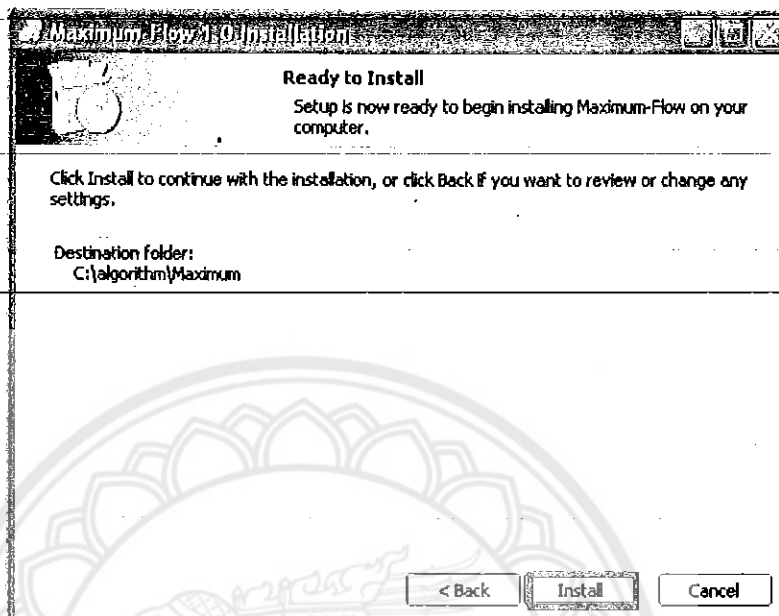
หลังจากนั้นให้ กด Next โดยไม่ต้องปรับค่าใดๆ เนื่องจาก โปรแกรมจะทำการสร้างโฟลเดอร์ให้โดยอัตโนมัติ<sup>1</sup>



รูปที่ 4.4 แสดงหน้าต่างเพื่อกำหนด Directory

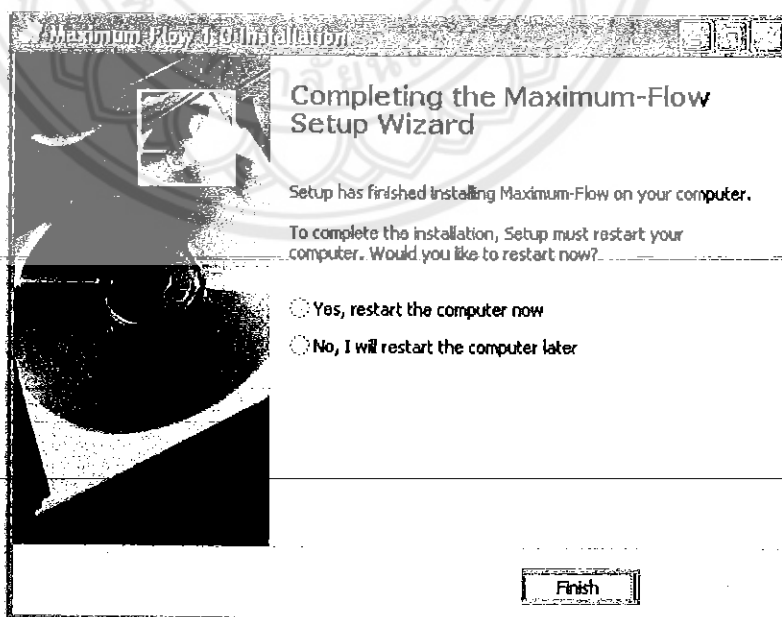
<sup>1</sup> ผู้ใช้งานสามารถเปลี่ยนแปลงได้ กรณีที่ต้องการสร้าง Directory ไว้ในตำแหน่งอื่น

เมื่อทำการกำหนด Directory เป็นที่เรียบร้อยแล้ว โปรแกรมจะทำการสร้างไฟล์เดสก์ทอปไอคอนตามที่โปรแกรมได้กำหนดไว้ หลังจากนั้นให้ กด Install เพื่อติดตั้ง โปรแกรม



รูปที่ 4.5 แสดงการขึ้นขั้นการติดตั้ง

เมื่อติดตั้ง โปรแกรมเป็นที่เรียบร้อยแล้ว ทำการเลือกปุ่ม Finish คอมพิวเตอร์จะทำการ Restart



รูปที่ 4.6 แสดงหน้าต่างเมื่อติดตั้งโปรแกรมเสร็จสมบูรณ์

กรณีที่ไม่ต้องการให้คอมพิวเตอร์ Restart ให้ทำการเลือกปุ่ม No, I will restart computer later หลังจากนั้นให้ กด Finish เพื่อสิ้นสุดการติดตั้งโปรแกรม



รูปที่ 4.7 แสดงหน้าต่างที่ไม่ต้องการ restart

## 4.2 การใช้งานโปรแกรม

เปิดโปรแกรม MATLAB ขึ้นมาไปที่หน้าต่าง command แล้วพิมพ์คำสั่ง guide เพื่อเรียกหน้าต่าง guide Quick start

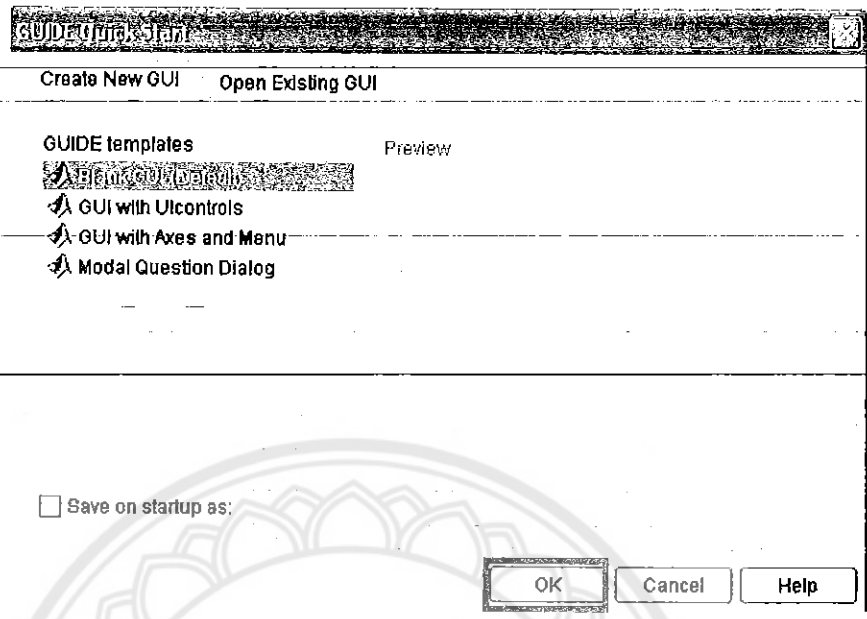
คำสั่งที่ใช้คือ

To get started, select **Help** or **Demos** from the Help menu.

```
>> guide
>> |
```

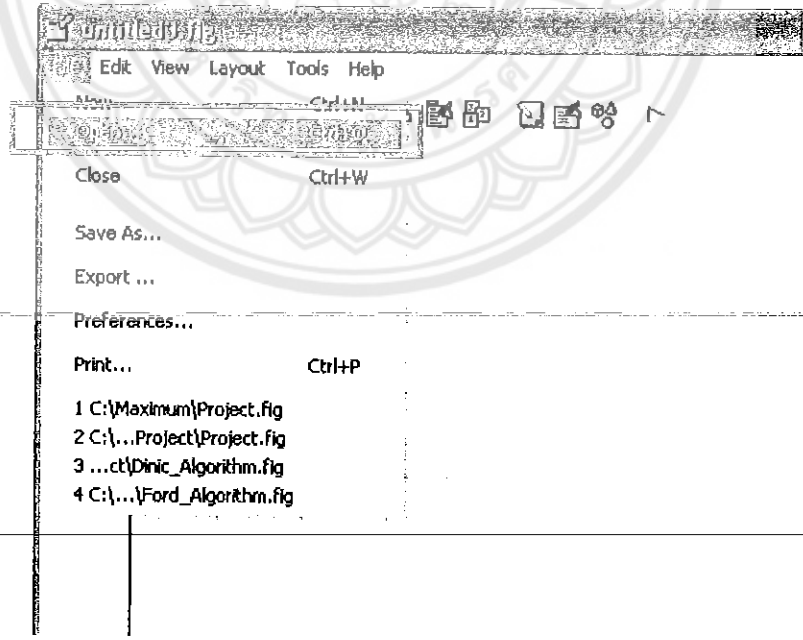
รูปที่ 4.8 การใช้คำสั่ง guide

หลังจากนั้นจะปรากฏหน้าต่าง Guide Quick start ทำการเลือกตัวเลือกที่ 1 แล้วเลือกปุ่ม OK



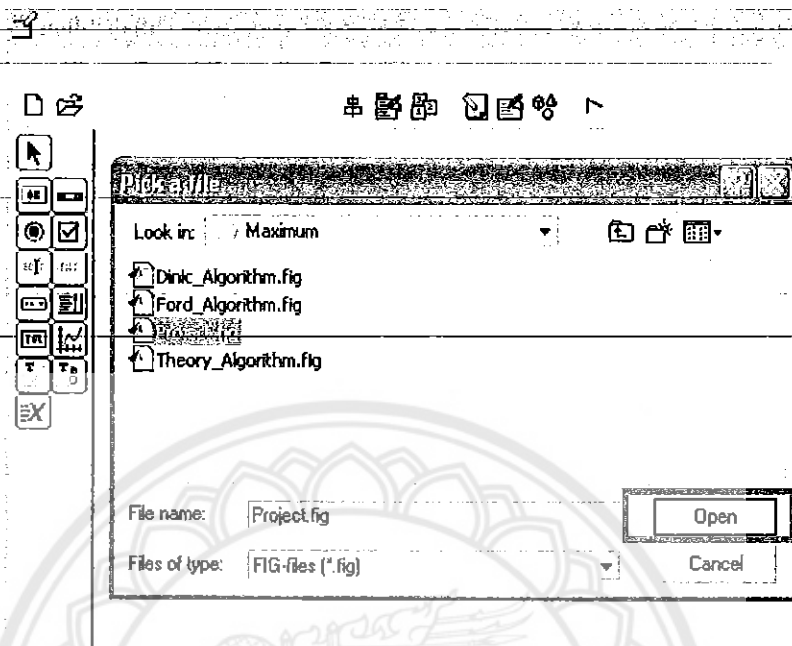
รูปที่ 4.9 แสดงหน้าต่าง การเรียกใช้งาน โปรแกรม

เลือกเปิด ไฟล์ Project.fig ภายใต้เมนู File->open



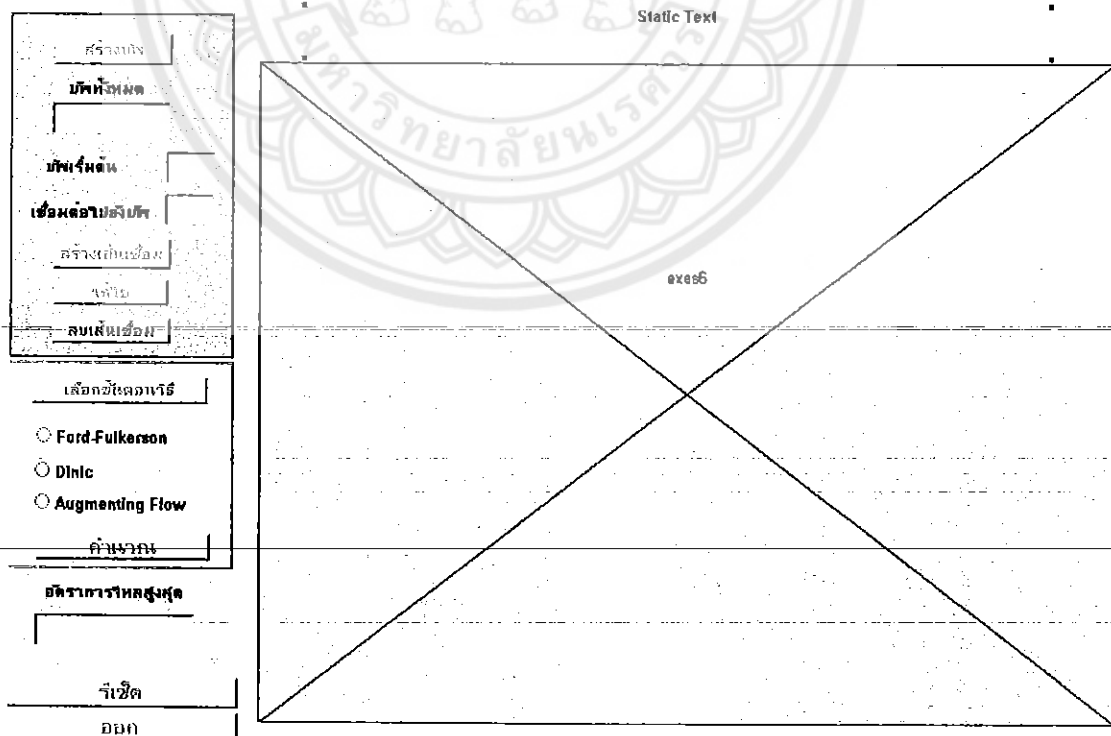
รูปที่ 4.10 แสดงการเปิดไฟล์โปรแกรม

โดยที่ไฟล์ Project.fig อยู่ใน Directory C:\Algorithm\maximum (กรณีที่ไม่มีการเปลี่ยนแปลง จาก Directory ที่ตั้งไว้ในโปรแกรม) เมื่อเลือกไฟล์ที่ได้แล้วให้กดปุ่ม Open



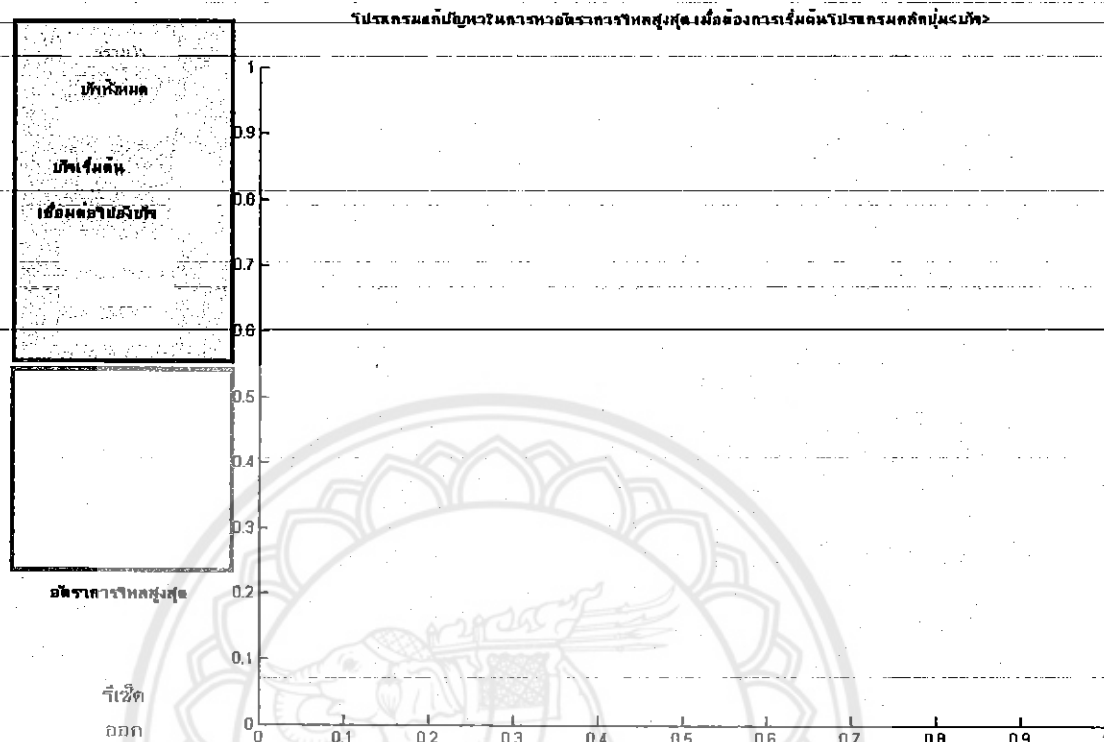
รูปที่ 4.11 เปิดไฟล์โปรแกรม

ทำการประมวลผลไฟล์ที่ได้โดยกดฟังก์ชันสามเหลี่ยมสีเขียว ที่อยู่ในเมนูด้านบน



รูปที่ 4.12 แสดงหน้าต่าง GUI

เมื่อโปรแกรมถูกประมวลผลเรียบร้อยแล้วจะได้หน้าต่างของโปรแกรมออกมา ดังแสดงในรูปที่ 4.13



รูปที่ 4.13 แสดงส่วนต่อประสานกราฟิก

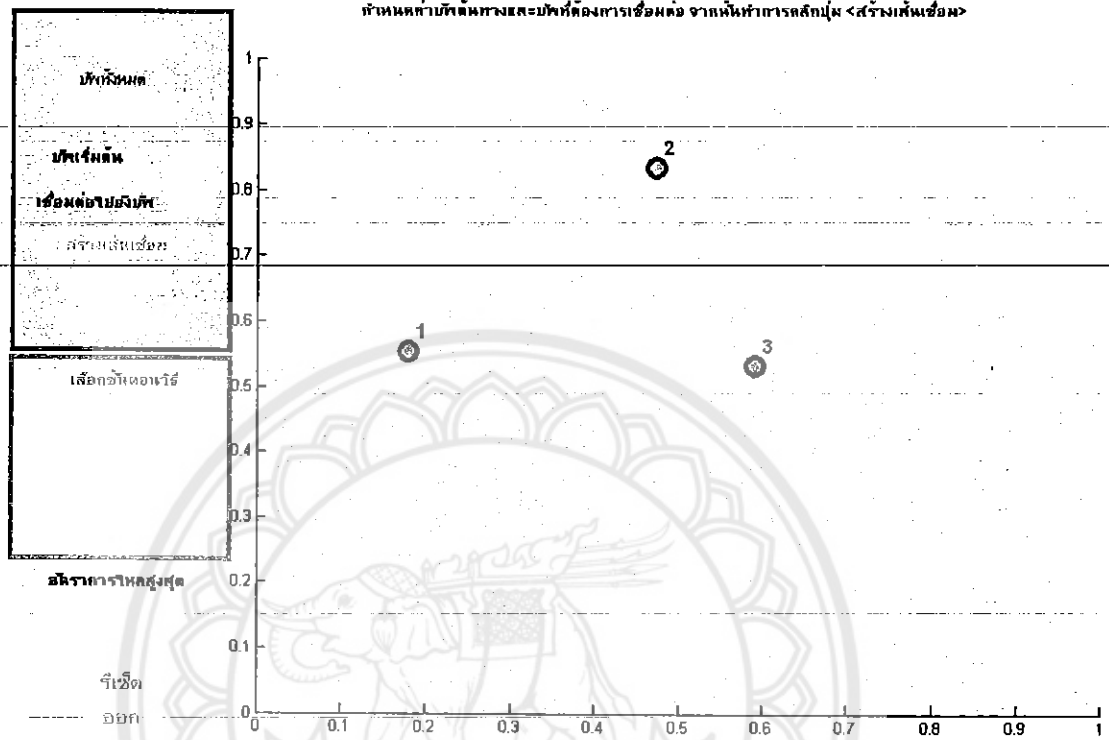
ในหน้าต่างของ โปรแกรมที่ได้มีปุ่มให้เลือก รวมทั้งมีกล่องข้อความสำหรับให้ผู้ใช้กรอก ซึ่งมีรายละเอียด ดังนี้

- ปุ่มรับ ใช้ในการสร้างรับ โดยให้ผู้ใช้งานกดเม้าท์ด้านซ้ายในตำแหน่งที่ต้องการสร้างรับและให้กดเม้าท์ด้านขวาเพื่อบ่งบอกว่าเป็นรับสุดท้ายที่ต้องการสร้าง
- ปุ่มสร้างเส้นเชื่อม ใช้ในการสร้างส่วนโค้งระหว่างรับ รวมทั้งกำหนดค่าอัตราการไหลสุทธิและค่าความจุ
- ปุ่มแก้ไข ใช้ในการแก้ไขค่าอัตราการไหล หรือค่าความจุระหว่างเส้นเชื่อม
- ปุ่มลบเส้นเชื่อม ใช้ในการลบเส้นเชื่อม
- ปุ่มเลือกขั้นตอนวิธี ใช้ในการเลือกขั้นตอนวิธีที่ใช้ในการคำนวณ
- ปุ่มคำนวณ ใช้ในการคำนวณเพื่อหาอัตราการไหลสูงสุด
- ปุ่มรีเซ็ต ใช้ในการลบค่าของตัวแปรต่างๆในโปรแกรม และ ล้างหน้าจอ
- ปุ่มออก ใช้ในการออกจากโปรแกรมเมื่อเลิกใช้โปรแกรมแล้ว



4.2.1 การสร้างบัพ

กลุ่มสร้างบัพ เพื่อทำการสร้างบัพ โดยให้ผู้ใช้งานกดเมาท์ด้านซ้ายในตำแหน่งที่ต้องการสร้างบัพ และให้กดเมาท์ด้านขวาเพื่อบ่งบอกว่า เป็นบัพสุดท้ายที่ต้องการสร้าง



รูปที่ 4.14 การสร้างบัพ

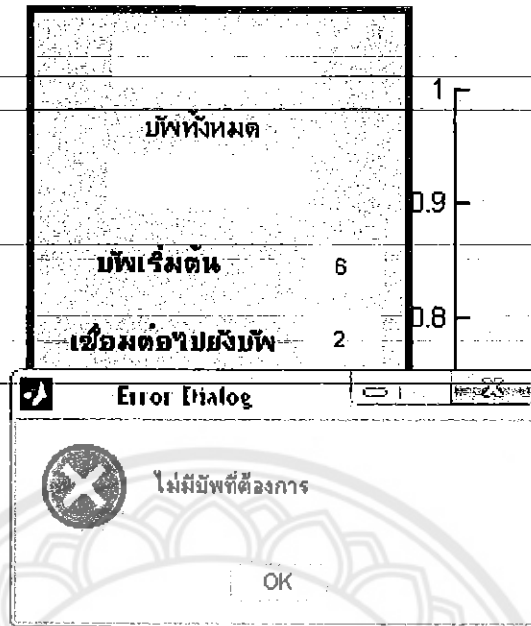
4.2.2 การสร้างเส้นเชื่อม การกำหนดอัตราการไหล และค่าความจุ

ให้ผู้ใช้งานระบุบัพเริ่มต้นและบัพที่ต้องการเชื่อมต่อในกล่องข้อความ โดยใส่เป็นหมายเลขตามบัพที่สร้าง จากนั้นกดปุ่ม สร้างเส้นเชื่อม

บัพทั้งหมด	
บัพเริ่มต้น	1
เชื่อมต่อไปยังบัพ	2
สร้างเส้นเชื่อม	

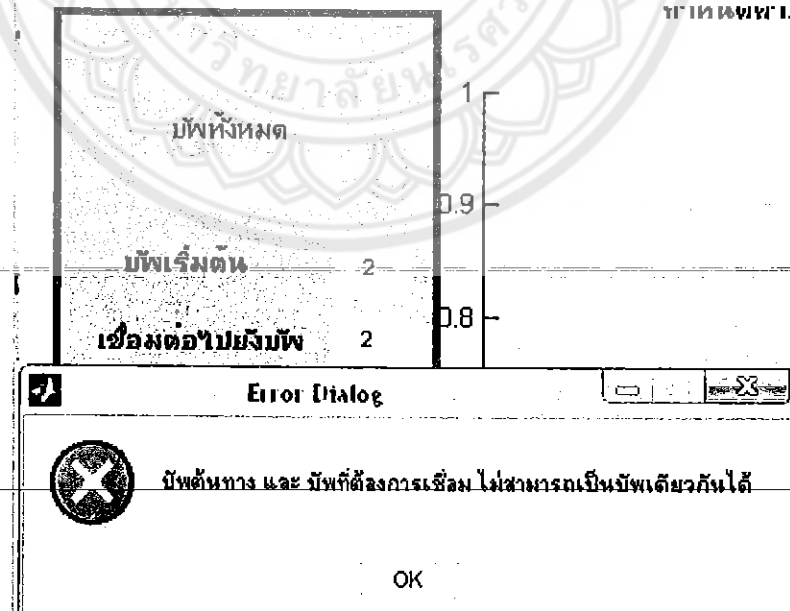
รูปที่ 4.15 ทำการสร้างเครือข่าย

- กรณีที่ผู้ใช้ ทำการใส่บัพต้นทางที่ไม่มีในหน้าจอ ระบบจะแสดงหน้าต่างแจ้งเตือนดังนี้



รูปที่ 4.16 หน้าต่างแสดงข้อผิดพลาดในกรณีที่ผู้ใช้ทำการใส่บัพที่ไม่มีในหน้าจอ

- กรณีที่ผู้ใช้ใส่บัพเริ่มต้นและบัพที่เชื่อมต่อเป็นบัพเดียวกัน ระบบจะแสดงหน้าต่างแจ้งเตือนดังนี้



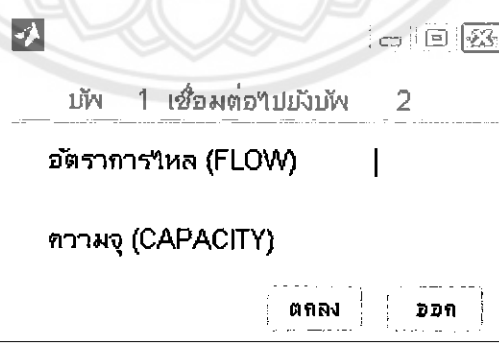
รูปที่ 4.17 หน้าต่างแสดงข้อผิดพลาดในกรณีที่ผู้ใช้ใส่บัพเริ่มต้นและบัพที่เชื่อมต่อเป็นบัพเดียวกัน

- กรณีที่ผู้ใช้ได้ใส่บัพเริ่มต้นและบัพที่เชื่อมต่อเข้ากับบัพเดิมที่เคยกำหนดไว้แล้ว ระบบจะแสดงหน้าต่างแจ้งเตือนดังนี้



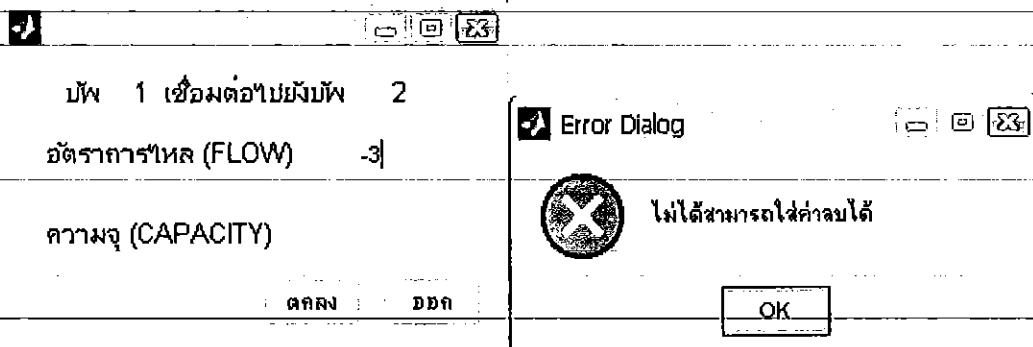
รูปที่ 4.18 หน้าต่างแสดงข้อผิดพลาดในกรณีที่ผู้ใช้ระบุบัพเริ่มต้นและบัพที่เชื่อมต่อซ้ำ

เมื่อกำหนดค่าบัพเริ่มต้นกับบัพที่ต้องการเชื่อมเป็นที่เรียบร้อยแล้ว ให้กดปุ่มสร้างเส้นเชื่อม ระบบจะแสดงหน้าต่างเพื่อให้ผู้ใช้งานกำหนดค่าอัตราการไหลและค่าความจุในแต่ละส่วนโค้ง โดยที่ค่าอัตราการไหลต้องมีค่าน้อยกว่าหรือเท่ากับค่าความจุและต้องมีค่ามากกว่าหรือเท่ากับศูนย์เท่านั้น จากนั้นให้ผู้ใช้งานกดปุ่มตกลงหรือกดปุ่มออกถ้าไม่ต้องการกำหนดค่าบนเส้นเชื่อมนั้น



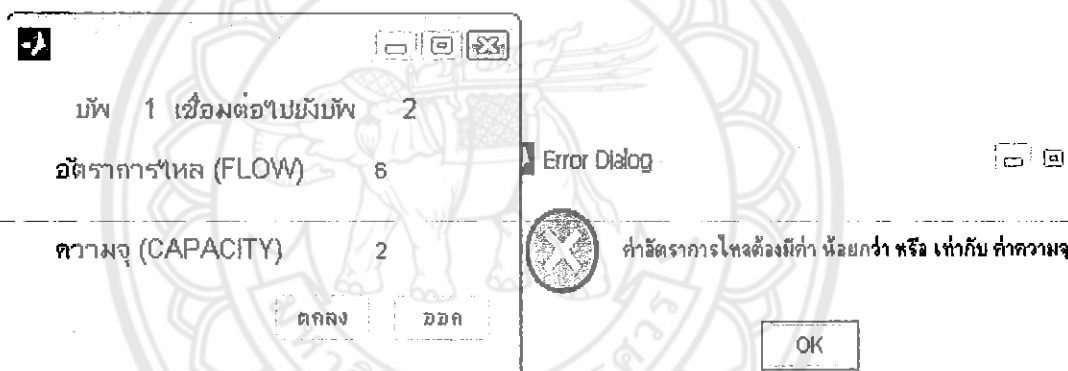
รูปที่ 4.19 หน้าต่างแสดงการกำหนดค่าอัตราการไหล และค่าความจุในแต่ละส่วนโค้ง

- กรณีที่ผู้ใช้ใส่ค่าอัตราการไหล หรือค่าความจุเป็นค่าลบ ระบบจะแสดงหน้าต่างแจ้งเตือนดังนี้



รูปที่ 4.20 หน้าต่างแสดงข้อผิดพลาดในกรณีผู้ใช้งานป้อนค่าเป็นลบ

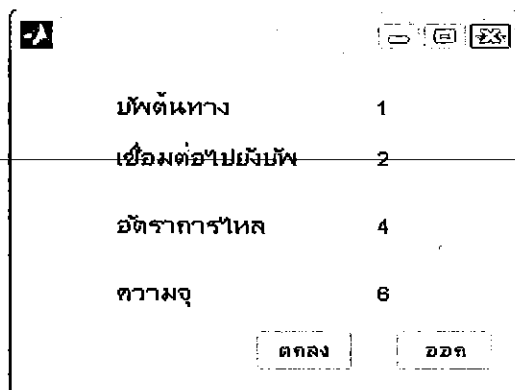
- กรณีที่ผู้ใช้ใส่ค่าอัตราการไหลมากกว่าค่าความจุ ระบบจะแสดงหน้าต่างแจ้งเตือนดังนี้



รูปที่ 4.21 หน้าต่างแสดงข้อผิดพลาดในกรณีผู้ใช้งานกำหนดค่าอัตราการไหลมากกว่าความจุ

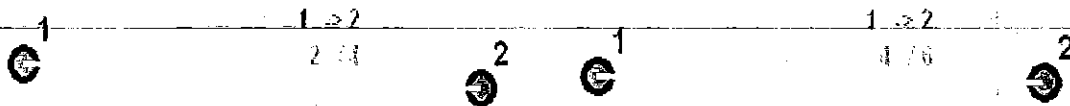
#### 4.2.3 การแก้ไขค่าอัตราการไหล ค่าความจุและการลบเส้นเชื่อม

เมื่อผู้ใช้ต้องการแก้ไขค่าอัตราการไหลหรือค่าความจุบนแต่ละส่วนโค้ง ให้คลิกปุ่มแก้ไขในเมนูด้านซ้ายมือซึ่งจะปรากฏหน้าต่างดังนี้



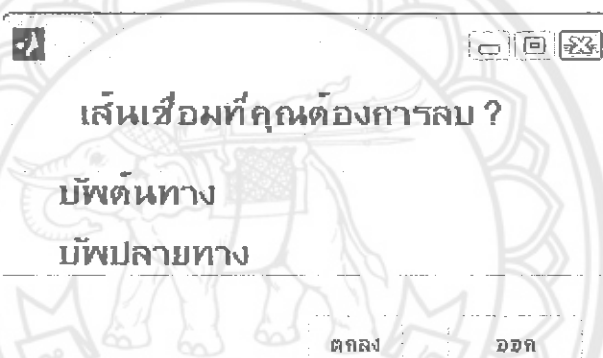
รูปที่ 4.22 แสดงหน้าต่างเมื่อต้องการแก้ไขค่าอัตราการไหลสุทธิ และค่าความจุ

ให้ผู้ใช้งานระบุทิศทาง บัพที่เชื่อมต่อ ค่าอัตราการไหล และค่าความจุที่ต้องการ ทำการกดปุ่มตกลง ซึ่งโปรแกรมจะทำการไขดังรูปที่ 4.23



รูปที่ 4.23 แสดงการแก้ไขค่าอัตราการไหลสุทธิและค่าความจุ  
(รูปซ้ายมือ) แสดงกราฟก่อนแก้ไขอัตราการไหลสุทธิและค่าความจุ  
(รูปขวามือ) แสดงกราฟหลังแก้ไขอัตราการไหลสุทธิและค่าความจุ

เมื่อผู้ใช้ต้องการลบเส้นเชื่อมที่ไม่ต้องการ ให้ผู้ใช้กดปุ่มลบเส้นเชื่อมดังแสดงในรูปที่ 4.24



รูปที่ 4.24 แสดงหน้าต่างเมื่อผู้ใช้งานต้องการลบเส้นเชื่อม



รูปที่ 4.25 แสดงการลบเส้นเชื่อม

รูปที่ 4.25 แสดงการลบเส้นเชื่อม  
(รูปซ้ายมือ) แสดงกราฟก่อนลบเส้นเชื่อม  
(รูปขวามือ) แสดงกราฟหลังลบเส้นเชื่อม

#### 4.2.4 การคิดคำนวณ

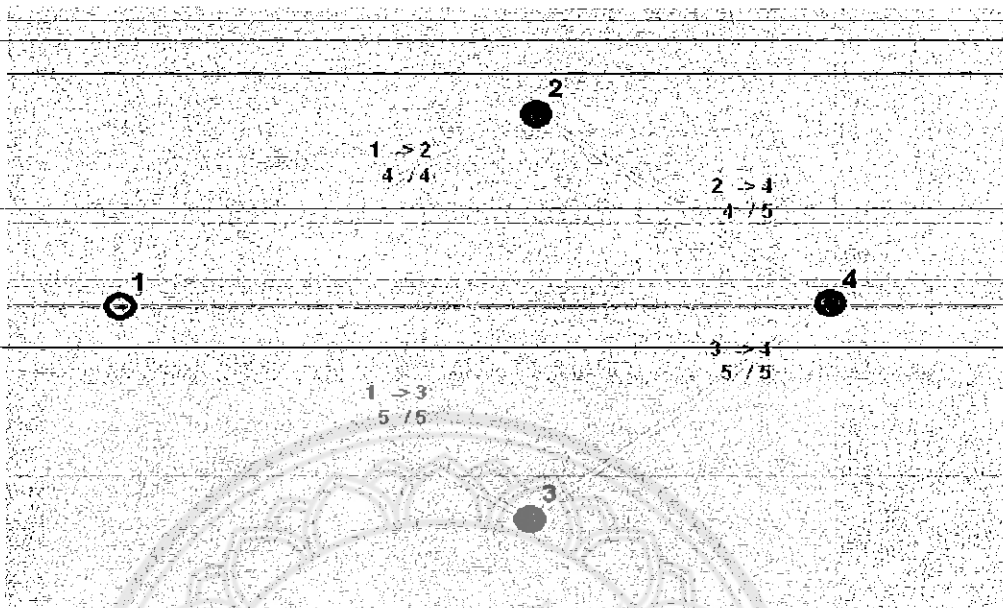
เมื่อผู้ใช้ทำการระบุค่าต่างๆจนเป็นที่เรียบร้อยแล้ว ให้ผู้ใช้งานกดปุ่มเลือกขั้นตอนวิธีเพื่อเลือกขั้นตอนวิธีที่ต้องการใช้ในการคำนวณในที่นี้มี 3 ขั้นตอนวิธีให้เลือกดังแสดงในรูปที่ 4.26 จากนั้นให้กดปุ่มคำนวณ

รูปที่ 4.26 แสดงการเลือกขั้นตอนวิธี

หลังจากกดปุ่มคำนวณแล้ว ระบบจะมีหน้าต่างแจ้งให้ผู้ใช้ถึงแหล่งต้นทางและแหล่งปลายทางของกราฟ จากนั้นให้ผู้ใช้กรอกแหล่งต้นทางและระบุแหล่งปลายทาง เพื่อความถูกต้อง

รูปที่ 4.27 แสดงหน้าต่างในการกำหนดแหล่งต้นทางและแหล่งปลายทาง

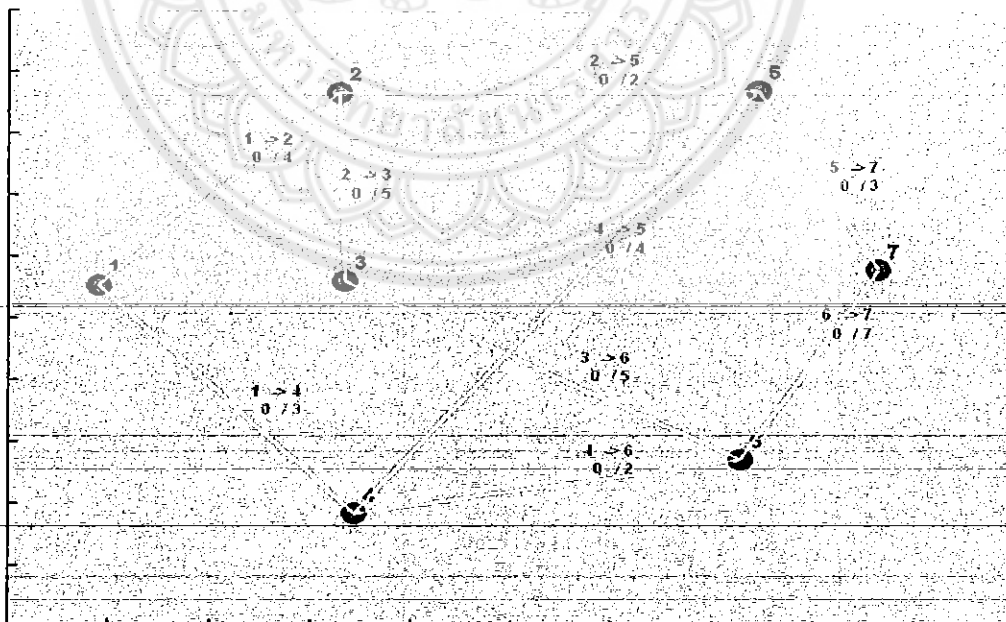
โปรแกรมจะทำการคำนวณหาอัตราการใช้พลังงานสูงสุดซึ่งได้ผลลัพธ์ตามรูปที่ 4.28 โดยเส้นสีเขียวจะระบุถึงจุดตัดต่ำสุด (min cut)



รูปที่ 4.28 แสดงหน้าต่างที่มีการปรับปรุงจนได้อัตราการใช้พลังงานสูงสุด

### 4.3 ตัวอย่างการใช้งานโปรแกรม

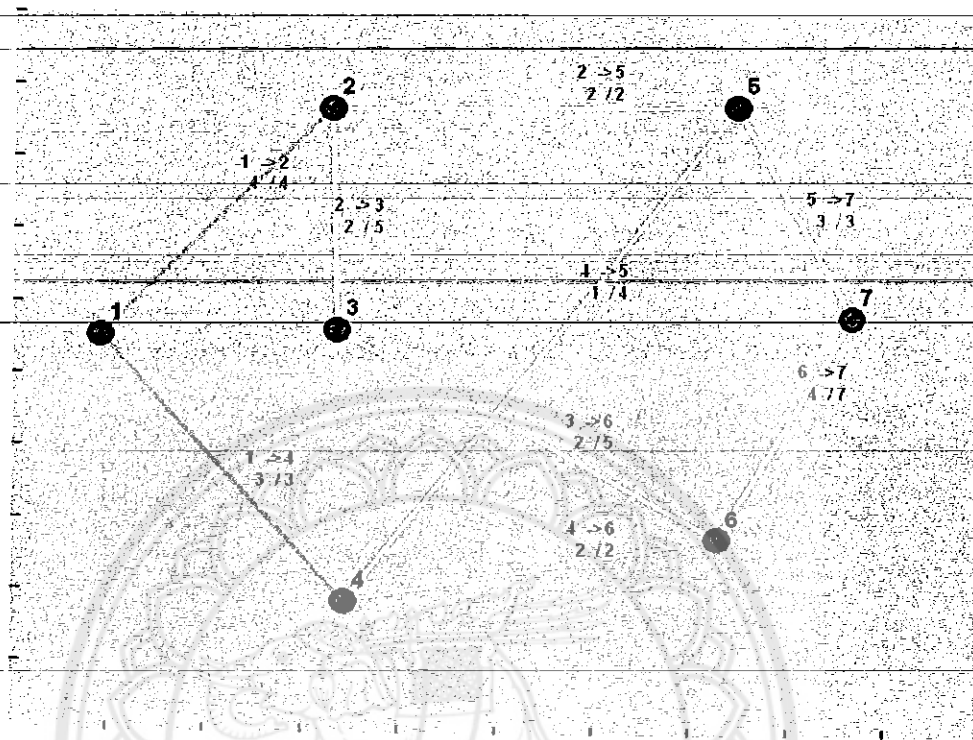
พิจารณาเครือข่ายเชิงเดียวในรูปที่ 4.29



รูปที่ 4.29 เครือข่ายเชิงเดียวก่อนการปรับปรุง

หลังจากใช้โปรแกรมที่พัฒนาขึ้นคำนวณพบว่าอัตราการไหลสูงสุด คือ 7 และมีจุดตัดบริเวณ

$K = (S, \bar{S}) = \{(v_1, v_2), (v_1, v_4)\}$  ซึ่งสอดคล้องกับส่วนโค้งที่มีสีเขียวในรูปที่ 4.30



รูปที่ 4.30 เครือข่ายเชิงเดียวหลังจากปรับปรุง



## บทที่ 5

### บทสรุป

โครงการได้ทำการประยุกต์ใช้ขั้นตอนวิธีที่ใช้สำหรับการหาอัตราการไหลสูงสุดใ  
เครือข่าย โครงการนี้ยังสามารถประยุกต์ใช้กับปัญหาอื่น ๆ ได้อีกหลายอย่างเช่น การหาอัตราการ  
ไหลของน้ำในท่อน้ำ ความหนาแน่นของรถบนถนนและจำนวนสินค้าที่เลื่อนบนสายพานใน  
โรงงานอุตสาหกรรม ซึ่งโครงการนี้ได้มีการมีการออกแบบเป็นแบบกราฟิก เพื่อให้ผู้ใช้ได้ใช้งาน  
อย่างสะดวก และสามารถเข้าใจ โปรแกรมได้ง่ายยิ่งขึ้น โดย โปรแกรมจะทำการจำลองระบบ  
เครือข่าย แสดงผลในรูปแบบกราฟ ตลอดจนผลลัพธ์ที่เกิดจากการใช้โปรแกรมนี้จะแสดงผล  
ออกมาเป็นแบบกราฟิก

#### 5.1 การทดสอบขั้นตอนวิธีในการหาอัตราการไหลสูงสุด

โปรแกรมนี้สร้างขึ้นเพื่อใช้ในการหาอัตราการไหลสูงสุดจากแหล่งต้นทางไปยังแหล่ง  
ปลายทาง โดยโปรแกรมที่พัฒนาขึ้นนี้มุ่งเน้นให้สะดวกต่อผู้ใช้งานทั่วไปซึ่งการแสดงผลเป็นการ  
แสดงผลเป็นแบบรูปภาพ มีใช้ปุ่มกด และมีคำอธิบายในส่วนหัวของโปรแกรมเป็นลำดับ จาก  
การทดสอบ พบว่าขั้นตอนวิธีที่เลือกใช้ให้ผลการคำนวณที่ถูกต้อง เข้าใจง่าย มีความสอดคล้องกับทฤษฎี  
ทุกประการ

#### 5.2 ปัญหาและอุปสรรค

5.2.1 เนื่องจาก โปรแกรม ภาษา MATLAB เป็นภาษาที่ใหม่ซึ่งผู้จัดทำโครงการไม่มีความเข้าใจ  
โดยเฉพาะด้านการออกแบบ ส่วนประสานกราฟิกกับผู้ใช้ (GUI) จึงทำให้การออกแบบและพัฒนา  
เป็นไปด้วยความล่าช้า

5.2.2 ขั้นตอนวิธีที่ใช้ในการหาอัตราการไหลสูงสุดยังไม่เป็นที่แพร่หลายเท่าที่ควร ทำให้  
การศึกษาค้นคว้าแต่ละขั้นตอนวิธีเกิดความล่าช้า

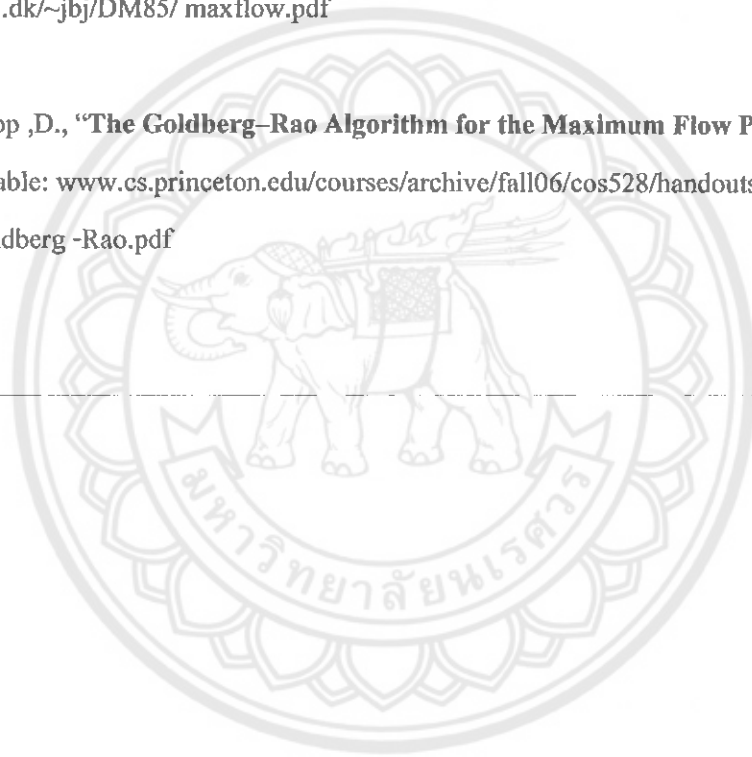
#### 5.3 ข้อเสนอแนะและแนวทางการพัฒนา

5.3.1 โปรแกรมที่พัฒนาสามารถประยุกต์ใช้หรือพัฒนากับปัญหาอื่น ๆ ได้อีกหลายอย่างเช่น  
การไหลของน้ำในท่อน้ำ ความหนาแน่นของรถบนถนน เป็นต้น

5.3.2 โปรแกรมที่พัฒนาขึ้นสามารถพัฒนากับระบบสัมผัส เพื่อเพิ่มความสะดวกกับผู้ใช้งาน

## เอกสารอ้างอิง

- [1] Hochbaum, D., "Graph Algorithms and Network Flows", IEOR 266, Fall 2003
- [2] Karger, D., "Blocking Flows", [Online]. Available : [courses.csail.mit.edu/6.854/06/ ...](http://courses.csail.mit.edu/6.854/06/...)  
[scribe/scribe11.pdf](#)
- [3] Larsen, J., and Clausen, J., "The Max Flow Problem", [Online]. Available : [www.imada.sdu.dk/~jbj/DM85/maxflow.pdf](http://www.imada.sdu.dk/~jbj/DM85/maxflow.pdf)
- [4] Papp, D., "The Goldberg-Rao Algorithm for the Maximum Flow Problem" [Online].  
Available: [www.cs.princeton.edu/courses/archive/fall06/cos528/handouts/ ...](http://www.cs.princeton.edu/courses/archive/fall06/cos528/handouts/...)  
[Goldberg -Rao.pdf](#)



## ประวัติผู้เขียนโครงการ



ชื่อ นายอภิวัฒน์ กันทวงศ์

ภูมิลำเนา 35/1 ถ.ไทยลานนา ต.หัวเวียง อ.เมือง จ.ลำปาง

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนบุญวาทย์วิทยาลัย

- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail: james\_13\_1@hotmail.com

