เทคนิคคอลัมน์เจเนอเรชันสำหรับปัญหาการจับคู่เที่ยวบิน

นางสาวสมิหลา คีรีศรี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

COLUMN GENERATION TECHNIQUE FOR CREW PAIRING PROBLEM

Miss Samila Kirisri

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Applied Mathematics and Computational Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2012

หัวข้อวิทยานิพนธ์	เทคนิคคอลัมน์เจอเนอเรชันสำหรับปัญหาการจับคู่เที่ยวบิน
โดย	นางสาวสมิหลา คีรีศรี
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	อาจารย์ ดร.บุญฤทธิ์ อินทิยศ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผู้ช่วยศาสตราจารย์ ดร.ชวลิต จีนอนันต์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

....คณบดีคณะวิทยาศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ หารหนองบัว)

คณะกรรมการสอบวิทยานิพนธ์

....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.กรุง สินอภิรมย์สราญ)

....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(อาจารย์ ดร.บุญฤทธิ์ อินทิยศ)

....อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(ผู้ช่วยศาสตราจารย์ ดร.ชวลิต จีนอนันต์)

....กรรมการ
(อาจารย์ ดร.พันทิพา ทิพย์วิวัฒน์พจนา)

....กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร.บุณฑริกา เกษมสันติธรรม)

สมิหลา คีรีศรี : เทคนิคคอลัมน์เจเนอเรชันสำหรับปัญหาการจับคู่เที่ยวบิน. (COLUMN GENERATION TECHNIQUE FOR CREW PAIRING PROBLEM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : อ.ดร.บุญฤทธิ์ อินทิยศ, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ผศ.ดร.ชวลิต จีนอนันต์, 50 หน้า.

ปัญหาการจับคู่เที่ยวบินเป็นปัญหาของการสร้างคู่เที่ยวบิน โดยมีลำดับเที่ยวบินที่เริ่มและจบที่ฐานการบินเดียวกัน และเป็นไปตามเงื่อนไขบังคับ โดยให้มีค่าใช้จ่ายน้อยที่สุด ปัญหาดังกล่าวเป็นปัญหาการหาค่าเหมาะที่สุดเชิงการจัดที่มีขนาดใหญ่ ซับซ้อนและยากในการหาคำตอบ ในบทความวิจัยนี้ได้นำเสนอตัวแบบทางคณิตศาสตร์สำหรับปัญหาการจับคู่เที่ยวบินโดยใช้เงื่อนไขบังคับเกี่ยวกับระยะเวลาการทำงานและการพักของพนักงานสายการบินจากบริษัทการบินไทย ในการหาคำตอบเราเสนอการใช้เทคนิคคอลัมน์เจเนอเรชันซึ่งเป็นเทคนิคที่มีประสิทธิภาพในการแก้ปัญหาที่มีขนาดใหญ่โดยการแก้ปัญหาย่อยอย่างซ้ำๆจนกระทั่งได้ผลเฉลยของปัญหาเดิม นอกจากนี้เรายังนำเสนอผลการคำนวณเชิงตัวเลขและการวิเคราะห์ผลโดยใช้ตัวอย่างข้อมูลจากการบินไทยเป็นกรณีศึกษา

ภาควิชา คณิตศาสตร์ประยุกต์และวิทยาการคณนา	ลายมือชื่อนิสิต
สาขาวิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก
ปีการศึกษา 2555	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม

5373829723 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : CREW PAIRING / PAIRING / COLUMN GENERATION

SAMILA KIRISRI : COLUMN GENERATION TECHNIQUE FOR CREW PAIRING PROBLEM. ADVISOR : BOONYARIT INTIYOT, Ph.D., CO-ADVISOR : ASST. PROF. CHAWALIT JEENANUNTA, Ph.D., 50 pp.

A crew pairing problem is a problem of creating the sequences of flights that start and end at the same crew base under some constraints while minimizing the cost. This is a large-scaled combinatorial optimization problem and it is complex and difficult to solve. In this paper, a mathematical model for a crew pairing problem is presented using the working time and rest time constraints from Thai Airways. We propose a column generation technique, which is a powerful technique for solving large-scale problems by iteratively solving smaller parts of the problem until the whole problem is solved. Some numerical results are also presented and analyzed using the data sample from Thai Airways as a case study.

Department : Mathematics and Computer Science Student's Signature ....

Field of Study : Applied Mathematics and Computational Science Advisor's Signature ....

Co-advisor's Signature ....

Academic Year : 2012

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลืออย่างดียิ่งของบุคคลหลายฝ่าย ผู้วิจัยขอกราบขอบพระคุณ อาจารย์ ดร. บุญฤทธิ์ อินทิยศ และ ผู้ช่วยศาสตราจารย์ ดร. ชวลิต จีนอนันต์ ที่ได้กรุณาให้ความรู้ คำแนะนำ รวมทั้งได้สละเวลาในการตรวจ และคำปรึกษาต่างๆที่เป็นประโยชน์ต่อการทำวิทยานิพนธ์

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. กรุง สินอภิรมย์สราญ ประธานกรรมการ อาจารย์ ดร.พันทิพา ทิพย์วิวัฒน์พจนา กรรมการ และ อาจารย์ ดร.บุญฑริกา เกษมสันติธรรม กรรมการภายนอกมหาวิทยาลัย เป็นอย่างสูงที่ได้กรุณาให้คำแนะนำแง่คิด และแก้ไขข้อบกพร่องต่างๆในงานวิจัยนี้ ทำให้วิทยานิพนธ์เล่มนี้มีความสมบูรณ์มากขึ้น

นอกจากนี้ผู้วิจัยขอขอบพระคุณโครงการมหาวิทยาลัยวิจัยแห่งชาติของสำนักงานคณะกรรมการการอุดมศึกษาและบริษัทการบินไทยจากัด (มหาชน) ที่สละเวลาให้ความร่วมมือและข้อมูลที่เป็นประโยชน์สาหรับการทำวิจัย

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา ครอบครัว และเพื่อนๆที่ให้การสนับสนุนช่วยเหลือ และให้กำลังใจในการทำวิทยานิพนธ์เล่มนี้ให้สำเร็จลุล่วงไปได้ด้วยดี

สารบัญ

สารบัญตาราง

ตารางที่ หน้า

สารบัญภาพ

สารบัญภาพ

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อุตสาหกรรมสายการบินเป็นอุตสาหกรรมขนาดใหญ่ เนื่องจากอุตสาหกรรมการบินประกอบไปด้วยหลายส่วนการทำงาน ปัญหาการจัดตารางเป็นปัญหาหนึ่งที่มักจะพบในอุตสาหกรรมสายการบิน โดยปัญหาการจัดตารางภายในสายการบินแบ่งออกเป็น 4 ปัญหา ได้แก่ 1. ปัญหาการจัดตารางการใช้เครื่องบิน (aircraft scheduling) 2. ปัญหาการจัดตารางปฏิบัติงานของพนักงานสายการบิน (crew scheduling) 3. ปัญหาการจัดตารางกรณีที่สายการบินเกิดปัญหาเฉพาะหน้า (disruption management) ทำให้ต้องมีการปรับเปลี่ยนตารางการบิน และ 4. ปัญหาการจัดตารางการใช้งานอุปกรณ์เทคนิคต่างๆ (machine scheduling) เช่น ปัญหาลำดับการลงจอดของเครื่องบิน และปัญหาการลำเลียงสัมภาระ เป็นต้น ปัญหาการจัดตารางปฏิบัติงานของพนักงานสายการบินเป็นปัญหาที่เราสนใจศึกษา เนื่องจากปัญหาดังกล่าวมีขนาดใหญ่ ซับซ้อนและยาก อีกทั้งในแต่ละสายการบินมีพนักงานจำนวนมาก ดังนั้นเรื่องค่าใช้จ่ายในส่วนของพนักงานจึงมีความสำคัญ การจัดตารางพนักงานสายการบินที่ดีจะช่วยให้สายการบินเสียค่าใช้จ่ายลดลง เพื่อแก้ปัญหาการจัดตารางพนักงานสายการบินจึงมีการแบ่งปัญหาออกเป็น 2 ส่วน คือ ปัญหาการจับคู่เที่ยวบินและปัญหาการมอบหมายงานให้กับพนักงาน โดยปัญหาการจับคู่เที่ยวบิน คือ การหาเส้นทางการบินโดยพิจารณาฐานที่เครื่องบินออกและเครื่องบินกลับจะต้องเป็นฐานเดียวกัน ส่วนปัญหาการมอบหมายงานให้กับพนักงาน คือ การมอบหมายคู่เที่ยวบินที่ได้จากปัญหาการจับคู่เที่ยวบินให้กับพนักงานแต่ละคน ซึ่งปัญหาทั้งสองนี้ต้องสอดคล้องทั้งกฎและระเบียบต่างๆของสายการบิน และต้องการเสียค่าใช้จ่ายให้น้อยที่สุด

ปัญหาการจับคู่เที่ยวบิน คือ การหาเส้นทางการบินโดยพิจารณาฐานที่เครื่องบินออกและเครื่องบินกลับจะต้องเป็นฐานเดียวกัน ปัญหาการจับคู่เที่ยวบินเป็นปัญหาที่มีนักวิจัยสนใจศึกษามาเป็นระยะเวลาหลายปีแล้ว เนื่องจากการจับคู่เที่ยวบินที่มีความเหมาะสมจะส่งผลให้ค่าใช้จ่ายของสายการบินลดลง วิธีที่นิยมใช้ในการแก้ปัญหาการจับคู่เที่ยวบินคือ การสร้างตัวแบบทางคณิตศาสตร์ในรูปแบบเซตคาฟเวอริ่ง แต่วิธีนี้มีข้อเสียคือ เมื่อทางสายการบินมีเที่ยวบินและเส้นทาง

สำหรับให้บริการจำนวนมาก การแก้ปัญหาโดยนำวิธีซิมเพล็กซ์มาใช้ร่วมกับรูปแบบเซตคาฟเวอริ่ง จะใช้เวลาในการคำนวณหาคำตอบนานจนไม่เหมาะสำหรับการนำไปใช้ประโยชน์จริง

เทคนิคคอลัมน์เจเนอเรชันเป็นวิธีการหาคำตอบที่นิยมใช้กับปัญหาที่มีขนาดใหญ่ โดยเทคนิคนี้จะแบ่งปัญหาออกเป็นสองส่วน คือ ปัญหาหลัก และปัญหาย่อย เทคนิคดังกล่าวจะสามารถลดเวลาในการคำนวณผลเฉลยให้เร็วยิ่งขึ้น ซึ่งงานวิจัยนี้เป็นการประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชันกับปัญหาการจับคู่เที่ยวบินโดยมีจุดประสงค์เพื่อจับคู่เที่ยวบินซึ่งมีผลรวมของช่วงเวลาพักระหว่างเที่ยวบินน้อยที่สุด

โดยทั่วไปงานวิจัยที่เกี่ยวข้องกับปัญหาการจับคู่เที่ยวบินมักแก้ปัญหาโดยการจับคู่เที่ยวบินในลักษณะของคู่เที่ยวบินสั้นๆ (pairing) ในกรณีที่สายการบินมีเที่ยวบินให้บริการจำนวนมาก การจับคู่เที่ยวบินในลักษณะดังกล่าวจะได้คู่เที่ยวบินจำนวนมาก ซึ่งคู่เที่ยวบินที่ได้จะถูกนำไปใช้ในขั้นตอนการแก้ปัญหาการมอบหมายงานให้แก่พนักงานทำให้ได้คำตอบที่เป็นไปได้จำนวนมากเช่นกัน เป็นผลให้ปัญหาดังกล่าวมีขนาดใหญ่ ทำให้ต้องใช้เวลาและทรัพยากรมากในการแก้ปัญหา ดังนั้นเพื่อที่จะลดจำนวนคำตอบที่เป็นไปได้ในขั้นตอนการมอบหมายงานให้แก่พนักงาน ในงานวิจัยนี้นำเสนอตัวแบบทางคณิตศาสตร์สำหรับแก้ปัญหาการจับคู่เที่ยวบินซึ่งเราแก้ปัญหาโดยการจับคู่เที่ยวบินในลักษณะของคู่เที่ยวบินที่ยาวขึ้น ในงานวิจัยนี้เรียกคู่เที่ยวบินที่ยาวขึ้นว่า ทัวร์ (tour) ซึ่งการจับคู่เที่ยวบินจะต้องเป็นไปตามเงื่อนไขการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย โดยจะประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชัน และเราจะใช้โปรแกรมซีเพล็กซ์เพื่อช่วยในการคำนวณภายในเวลาที่เหมาะสม

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อสร้างตัวแบบทางคณิตศาสตร์สำหรับปัญหาการจับคู่เที่ยวบิน ให้มีความสอดคล้องตามเงื่อนไขระยะเวลาการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย โดยประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชันในการหาคำตอบ

1.3 ขอบเขตของการวิจัย

ศึกษาการจับคู่เที่ยวบินในรูปแบบของทัวร์ที่สอดคล้องกับเงื่อนไขระยะเวลาการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย เฉพาะเที่ยวบินภายในประเทศจำนวน 76 เที่ยวบิน และสร้างตัวแบบกำหนดการเชิงจำนวนเต็มสำหรับปัญหาการจับคู่เที่ยวบินและจับคู่เที่ยวบินในรูปแบบทัวร์ที่ครอบคลุมระยะเวลา 2 วัน โดยใช้เทคนิคคอลัมน์เจเนอเรชันในการหาคำตอบ

1.4 ลำดับขั้นตอนในการนำเสนอผลการวิจัย

วิทยานิพนธ์เล่มนี้จัดเรียงหัวข้อต่างๆ ดังนี้

บทที่ 2 กล่าวถึงแนวคิดและทฤษฎีของขั้นตอนวิธีที่นำมาใช้แก้ปัญหาในงานวิจัยนี้ และงานวิจัยที่เกี่ยวข้องกับปัญหาการจับคู่เที่ยวบินและปัญหาการจัดตารางงานพนักงานสายการบิน

บทที่ 3 กล่าวถึงข้อมูลทั่วไปของบริษัทการบินไทยรวมไปถึงแนวคิดในการหาคำตอบของปัญหาการจับคู่เที่ยวบินโดยประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชัน

บทที่ 4 แสดงผลเฉลยที่ได้จากปัญหาดังกล่าว และวิเคราะห์ประสิทธิภาพของผลเฉลยในแต่ละตัวอย่างทดสอบ

บทที่ 5 กล่าวถึงข้อสรุปงานวิจัยและข้อเสนอแนะของงานวิจัยนี้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

สำหรับงานวิจัยเทคนิคคอลัมน์เจเนอเรชันสำหรับปัญหาการจับคู่เที่ยวบิน เพื่อความเข้าใจทฤษฎีที่เกี่ยวข้อง ผู้อ่านควรมีความรู้ในเรื่องวิธีการซิมเพล็กซ์ (Simplex method) ซึ่งสามารถศึกษาได้จากตำราการวิจัยดำเนินการทั่วไป

2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง

คู่เที่ยวบิน คือ ลำดับของเที่ยวบินที่เริ่มต้นและสิ้นสุดที่ฐาน (crew base) เดียวกัน โดยฐานในงานวิจัยนี้ หมายถึง สนามบินที่ใช้เป็นจุดเริ่มต้นและสิ้นสุดการทำงานของพนักงานสายการบิน ดังนั้นปัญหาการจับคู่เที่ยวบิน (crew paring problem) คือ ปัญหาการจัดลำดับของเที่ยวบินที่เริ่มต้นและสิ้นสุดที่ฐานเดียวกัน โดยการจับคู่เที่ยวบินจะต้องสอดคล้องกับเงื่อนไขของสายการบินนั้นๆอีกด้วย วัตถุประสงค์หลักของการจับคู่เที่ยวบิน คือ เพื่อหาลำดับเที่ยวบินที่ครอบคลุมทุกเที่ยวบิน (flights) และช่วยให้สายการบินเสียค่าใช้จ่ายที่น้อยที่สุด

โดยทั่วไปปัญหาการจับคู่เที่ยวบินมักจะนำมาสร้างเป็นปัญหาในลักษณะตัวแบบทางคณิตศาสตร์เซตคาฟเวอริ่ง (set covering) ดังนี้

Minimize	$\mathbf{c}^T\mathbf{x}$	; โดย $\mathbf{c}$ แทนค่าใช้จ่าย
Subject to	$A\mathbf{x} \geq 1$	; เงื่อนไขของการจับคู่เที่ยวบิน คือ แต่ละเที่ยวบินจะต้องถูกนำมาจัดในคู่เที่ยวบินอย่างน้อย 1 ครั้ง
	$\mathbf{x} = \{x_j \mid j = 1,2,3,\ldots,n\}, x_j \in \{0,1\}$	

โดยเมทริกซ์ A ประกอบไปด้วย คอลัมน์ ซึ่งแต่ละคอลัมน์แสดงเที่ยวบินที่ถูกนำมาจัดคู่เที่ยวบิน ซึ่งเมทริกซ์ $A = [a_{ij}]$ เป็นเมทริกซ์สัมประสิทธิ์ที่มี $a_{ij} = 1$ ถ้าเที่ยวบิน i อยู่ในคู่เที่ยวบิน j และ $a_{ij} = 0$ ในกรณีอื่นๆ และเงื่อนไขของตัวแบบทางคณิตศาสตร์เซตคาฟเวอรริ่งที่ว่าแต่ละเที่ยวบินจะต้องถูกนำมาจัดในคู่เที่ยวบินอย่างน้อย 1 ครั้ง เป็นเงื่อนไขเพื่อให้เกิดความหยืด

หยุ่นในการโยกย้ายพนักงานจากเที่ยวบินหนึ่งไปยังอีกเที่ยวบินหนึ่ง ซึ่งปัญหาดังกล่าวเป็นปัญหาขนาดใหญ่และมีความซับซ้อนเมื่อมีเที่ยวบินจำนวนมากและมีความซับซ้อนขึ้นไปอีกเมื่อมีเงื่อนไขอื่นๆเพิ่มเติม เมื่อนำวิธีซิมเพล็กซ์มาใช้ร่วมกับรูปแบบเซตคาฟเวอริ่งจะใช้เวลาในการคำนวณคำตอบนาน งานวิจัยของเราประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชันซึ่งเป็นเทคนิคการหาคำตอบสำหรับปัญหาที่มีขนาดใหญ่ ด้วยเทคนิคนี้จะสามารถลดเวลาในการคำนวณผลเฉลยให้เร็วยิ่งขึ้น

2.1.1 เทคนิคคอลัมน์เจเนอเรชัน (Column Generation Technique)

เทคนิคคอลัมน์เจเนอเรชันเป็นเทคนิคการหาคำตอบสำหรับปัญหาที่มีขนาดใหญ่โดยใช้ควบคู่ไปกับวิธีการ branch and bound เทคนิคดังกล่าวจะแก้ปัญหา LP relaxation โดยจะแบ่งปัญหาออกเป็นสองส่วน คือ

ปัญหาหลัก (master problem) คือ ปัญหาที่ได้มาจากโจทย์ของเราโดยตรง ซึ่งจากโจทย์มักพบว่ามีคอลัมน์ที่เป็นไปได้จำนวนมาก หากแก้ปัญหาด้วยวิธีการซิมเพล็กจะใช้เวลาในการคำนวณคำตอบนานโดยเริ่มต้นปัญหาหลักจะถูกจำกัดเซตคำตอบหรือคอลัมน์ที่เป็นไปได้ให้มีขนาดเล็กลงเรียกว่า ปัญหาหลักที่ถูกจำกัด (restricted master problem) เพื่อให้ง่ายสำหรับการหาคำตอบด้วยวิธีการซิมเพล็กซ์ ซึ่งจะทำให้ได้ค่าตัวแปรคู่ควบ (dual variable) ที่จะถูกนำไปสร้างปัญหาย่อย

ปัญหาย่อย (sub-problem) คือ ปัญหาสำหรับตรวจสอบคำตอบของปัญหาหลักที่ถูกจำกัดว่าเหมาะที่สุดแล้วหรือไม่ โดยพิจารณาจากค่า reduced cost ของตัวแปรทั้งหมด ในลักษณะเดียวกันกับที่ใช้ในวิธีการซิมเพล็กซ์ โดยปัญหาย่อยประกอบด้วยคอลัมน์ที่เป็นไปได้ทั้งหมดซึ่งรวมถึงคอลัมน์นอกเหนือจากคอลัมน์ในปัญหาหลักที่ถูกจำกัด คอลัมน์ที่เป็นไปได้ทั้งหมดไม่ได้ถูกแจงออกมาอย่างชัดแจ้ง แต่เขียนแบบเป็นนัยในรูปแบบเงื่อนไขข้อบังคับ ในกรณีที่ค่า reduced cost สอดคล้องตามเงื่อนไขที่ให้ค่าที่เหมาะสมที่สุด นั่นคือ ปัญหาหลักที่ถูกจำกัดได้ค่าที่เหมาะที่สุดแล้วจะได้ว่าปัญหาหลักได้ค่าที่เหมาะสมที่สุดเช่นเดียวกัน ในกรณีที่ตรวจสอบแล้วพบว่าปัญหาหลักยังไม่ได้ค่าที่เหมาะที่สุด คำตอบในปัญหาย่อยจะบ่งบอกถึงคอลัมน์ ที่จะถูกนำไปใช้ในการปรับปรุงคำตอบของปัญหาหลักได้ เทคนิคคอลัมน์เจเนอเรชั่นโดยทั่วไป มีวิธีการ ดังนี้ [1]

พิจารณาปัญหากำหนดการเชิงเส้น (linear programming) ในรูปแบบทั่วไป

โดยที่ x_j เป็นตัวแปรตัดสินใจ

c_j และ a_{ij} เป็นค่าคงที่

J เป็นเซตของดัชนีของตัวแปรตัดสินใจทั้งหมด

m เป็นจำนวนอสมการเงื่อนไขทั้งหมด

$$\begin{aligned} \text{Minimize} \quad & \sum_{j \in J} c_j x_j \\ \text{Subject to} \quad & \sum_{j \in J} a_{ij} x_j \geq b_i \quad , i = 1,2,3,\ldots,m \\ & x_j \geq 0, \quad j \in J \end{aligned}$$

เรียกปัญหากำหนดการเชิงเส้นข้างต้นว่า ปัญหาหลัก (master problem) ในกระบวนการหาคำตอบของปัญหากำหนดการเชิงเส้นด้วยวิธีซิมเพล็กซ์ จะทำการหาตัวแปรไม่พื้นฐาน (non-basic variable) เพื่อนำมาปรับปรุงคำตอบให้ดีขึ้นโดยการนำเข้าไปเป็นตัวแปรพื้นฐาน (basic variable) การหาตัวแปรไม่พื้นฐานดังกล่าวพิจารณาจากค่า reduced cost ซึ่งคำนวณได้จาก $c_j - \sum_{i=1}^{m} \pi_i a_{ij}$ โดยที่ π_i คือ ตัวแปรคู่ควบ (dual variable) ซึ่งสอดคล้องกับเงื่อนไขที่ i ในกรณีที่ปัญหากำหนดการเชิงเส้นมีขนาดใหญ่มาก เนื่องจากเซต J มีขนาดใหญ่มาก ปัญหาดังกล่าวจะถูกจำกัดขนาดของปัญหาให้เล็กลง เรียกปัญหาใหม่นี้ว่า ปัญหาหลักที่ถูกจำกัด (restricted master problem) โดยการกำหนดเซต J' ซึ่งเป็นเซตย่อยของเซต J มาแทนเซต J ซึ่งปัญหาหลักที่ถูกจำกัดที่ได้ เป็นดังนี้

$$\begin{aligned} \text{Minimize} \quad & \sum_{j \in J'} c_j x_j \\ \text{Subject to} \quad & \sum_{j \in J'} a_{ij} x_j \geq b_i \quad , i = 1,2,3,\ldots,m \\ & x_j \geq 0, \quad j \in J' \end{aligned}$$

ปัญหาหลักที่ถูกจำกัดสามารถหาคำตอบได้ง่ายกว่าปัญหาเดิมเนื่องจากมีขนาดเล็กกว่า แต่คำตอบที่ได้อาจไม่ใช่คำตอบที่เหมาะที่สุดสำหรับปัญหาหลัก เราทำการตรวจสอบคำตอบดังกล่าวว่าเหมาะ

ที่สุดสำหรับปัญหาหลักหรือไม่โดยการสร้างปัญหาย่อย (sub-problem) เพื่อตรวจสอบจากค่า reduced cost ของตัวแปรไม่พื้นฐาน ซึ่งปัญหาย่อยดังกล่าวมีลักษณะต่อไปนี้

$$\bar{c}^* := \mathrm{Min}\{c(\mathbf{a}) - \boldsymbol{\pi}^T \mathbf{a} \mid \mathbf{a} \in \mathrm{Col(A)}\}$$

โดยที่ Col(A) คือ เซตของคอลัมน์ทั้งหมดของปัญหาที่มีเมทริกซ์สัมประสิทธิ์ A

$c(\mathbf{a})$ คือ ค่าใช้จ่ายซึ่งคำนวณได้จากคอลัมน์ $\mathbf{a}$

$\boldsymbol{\pi}$ คือ เวกเตอร์ของตัวแปรคู่ควบ ซึ่งสอดคล้องกับคำตอบ ณ ปัจจุบัน ในปัญหาหลักที่ถูกจำกัด

ในทางปฏิบัติเซต Col(A) ไม่ได้ถูกใช้โดยตรงเนื่องจากจำนวนคอลัมน์ทั้งหมดของปัญหามีปริมาณมาก แต่เงื่อนไข $\mathrm{a} \in \mathrm{Col(A)}$ จะถูกใช้โดยถูกจัดอยู่ในรูปแบบของเงื่อนไขในการสร้างแต่ละคอลัมน์ ส่วนค่า $\bar{c}^*$ ที่ได้จะเป็นค่า reduced cost ที่น้อยที่สุด จากความรู้เรื่องวิธีการซิมเพล็กซ์สามารถเขียนค่าฟังก์ชันจุดประสงค์ในรูปทั่วไปสำหรับตัวแปรไม่พื้นฐาน (x_j) ใดๆในรูป $\mathrm{z} = \mathrm{z}_0 + \sum_{j \in \mathrm{NB}} (c_j - z_j) x_j$ โดยที่ z_0 คือ ค่าฟังก์ชันจุดประสงค์ที่คำนวณจากตัวแปรพื้นฐาน และ $(c_j - z_j)$ คือ ค่า reduced cost ของตัวแปรไม่พื้นฐาน เนื่องจากปัญหาหลักเป็นปัญหาการหาค่าที่น้อยที่สุด ถ้าค่า $(c_j - z_j) \geq 0$ สำหรับบาง j การนำ x_j ดังกล่าวมาเป็นตัวแปรพื้นฐานจะเป็นการเพิ่มค่า z ซึ่งการเพิ่มดังกล่าวทำให้ไม่ได้ค่าที่น้อยที่สุด ในทางกลับกันถ้าค่า $(c_j - z_j) < 0$ สำหรับบาง j การนำ x_j ดังกล่าวมาเป็นตัวแปรพื้นฐานจะช่วยปรับปรุงค่าฟังก์ชันจุดประสงค์ให้ดีขึ้น ดังนั้น หากค่า $\bar{c}^*$ ที่ได้มีค่ามากกว่าหรือเท่ากับศูนย์ แสดงว่าคำตอบ ณ ปัจจุบันของปัญหาหลักเหมาะสมที่สุดแล้วและในทางกลับกันหากค่า $\bar{c}^*$ ที่ได้มีค่าน้อยกว่าศูนย์ แสดงว่าปัญหาหลักยังสามารถปรับปรุงคำตอบให้เหมาะสมยิ่งขึ้นได้ โดยนำเข้าตัวแปรไม่พื้นฐานที่สอดคล้องกับคอลัมน์ $\mathbf{a}$ ที่ได้จากปัญหาย่อยเข้าไปในปัญหาหลักที่ถูกจำกัด แล้วทำการหาคำตอบของปัญหาหลักที่ถูกจำกัดอีกครั้งจนกระทั่งไม่สามารถหาคอลัมน์ที่ปรับปรุงคำตอบของปัญหาหลักได้

ตัวอย่างที่นิยมใช้เทคนิคคอลัมน์เจเนอเรชันในการหาคำตอบ คือ การแก้ปัญหา cutting stock พิจารณาตัวอย่างการแก้ปัญหาต่อไปนี้

โรงงานแห่งหนึ่งมีแผ่นอลูมิเนียมขนาดมาตรฐานยาว 218 เซนติเมตร ลูกค้าต้องการแผ่นอลูมิเนียม 3 แบบด้วยกัน คือ แบบที่ 1 ยาว 81 เซนติเมตร จำนวน 44 แผ่น แบบที่ 2 ยาว 70 เซนติเมตร จำนวน 3 แผ่น และแบบที่ 3 ยาว 68 เซนติเมตร จำนวน 48 แผ่น ทางโรงงานต้องการตัดแผ่นอลูมิเนียมตามขนาดที่ลูกค้าต้องการโดยใช้จำนวนแผ่นอลูมิเนียมขนาดมาตรฐานน้อยที่สุด

จากโจทย์ปัญหาข้างต้น สามารถเขียนเป็นตัวแบบทางคณิตศาสตร์ ได้ดังนี้

$$
\begin{aligned}
\text{Minimize} \quad & x_1 + x_2 + x_3 + \ldots + x_n \\
\text{Subject to} \quad & a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n \geq 44 \\
& a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n \geq 3 \qquad \ldots(1.1) \\
& a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n \geq 48 \\
& x_1, x_2, x_3, \ldots, x_n \geq 0 \text{ and integer}
\end{aligned}
$$

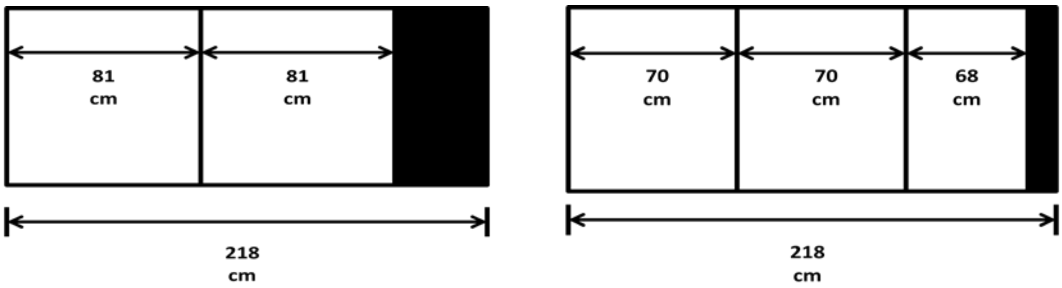
โดยที่ x_q คือ ตัวแปรตัดสินใจซึ่งแสดงจำนวนแผ่นอลูมิเนียมที่ถูกตัดตามรูปแบบที่ q บนแผ่นอลูมิเนียมมาตรฐาน โดยที่ $q \in \{1,2,3,\ldots,n\}$

n คือ จำนวนรูปแบบการตัดที่เป็นไปได้ทั้งหมด

a_{pq} คือ สัมประสิทธิ์ที่บอกจำนวนแผ่นที่ได้จากการตัดแผ่นอลูมิเนียม 1 แผ่น โดย a_{pq} แสดงจำนวนแผ่นที่มีความยาวแบบที่ p ซึ่งได้จากการตัดรูปแบบที่ q

จากตัวแบบคณิตศาสตร์ข้างต้นมีวัตถุประสงค์เพื่อหารูปแบบการตัดแผ่นอลูมิเนียมที่ทำให้โรงงานใช้แผ่นอลูมิเนียมขนาดมาตรฐานน้อยแผ่นที่สุด และการตัดแผ่นอลูมิเนียมจะต้องสอดคล้องกับเงื่อนไข คือ จำนวนแผ่นที่มีความยาวแบบที่ 1 (81 เซนติเมตร) จะต้องไม่น้อยกว่า 44 แผ่น จำนวนแผ่นที่มีความยาวแบบที่ 2 (70 เซนติเมตร) จะต้องไม่น้อยกว่า 3 แผ่น และจำนวนแผ่นที่มีความยาวแบบที่ 3 (68 เซนติเมตร) จะต้องไม่น้อยกว่า 48 แผ่น

ซึ่งรูปแบบการตัดที่เป็นไปได้มีหลายรูปแบบ ยกตัวอย่างการตัดแผ่นอลูมิเนียมดังภาพที่ 2.1



การตัดรูปแบบที่ 1 การตัดรูปแบบที่ 2

ภาพที่ 2.1 ตัวอย่างการตัดแผ่นอลูมิเนียมขนาดมาตรฐาน

ซึ่งแสดงรูปแบบในการตัดแผ่นอลูมิเนียมในลักษณะของเวกเตอร์ $\begin{pmatrix} a_{1q} \\ a_{2q} \\ a_{3q} \end{pmatrix}$ จากภาพที่ 2.1 ถ้าให้รูปที่หนึ่งและสองเป็นรูปแบบการตัดแบบที่ 1 และ 2 ตามลำดับ เราจะได้คอลัมน์ของ x_1 เป็น $\begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$ และ x_2 เป็น $\begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$ ซึ่งแสดงรูปแบบการตัดแผ่นอลูมิเนียมขนาดมาตรฐาน

โดยเวกเตอร์ $\begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$ แสดงให้เห็นว่าโรงงานตัดแผ่นอลูมิเนียมมาตรฐาน โดยตัดในความยาวแบบที่ 1 (81 cm) จำนวน 2 ชิ้นเท่านั้นและเวกเตอร์ $\begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$ แสดงให้เห็นว่าโรงงานตัดแผ่น

อลูมิเนียมมาตรฐาน โดยตัดในความยาวแบบที่ 2 (70 cm) จำนวน 2 ชิ้น และความยาวแบบที่ 3 (68 cm) จำนวน 1 ชิ้นเท่านั้น ซึ่งทางโรงงานสามารถตัดแผ่นอลูมิเนียมได้หลายรูปแบบด้วยกัน ทำให้ปัญหาดังกล่าวมีขนาดใหญ่เนื่องจากมีคอลัมน์ของรูปแบบการตัดเป็นจำนวนมาก การหาคำตอบด้วยวิธีการซิมเพล็กซ์จะใช้เวลาในการหาคำตอบนาน เทคนิคคอลัมน์เจเนอเรชันจึงถูกนำมาใช้เพื่อแก้ปัญหาดังกล่าว

สำหรับปัญหานี้ ปัญหาหลัก (master problem) สามารถเขียนอยู่ในรูปตัวแบบกำหนดการเชิงจำนวนเต็ม (1.1) ในการแก้ปัญหาด้วยเทคนิคคอลัมน์เจเนอเรชัน เราเริ่มด้วยการลดขนาดของปัญหาให้อยู่ในรูปแบบปัญหาหลักที่ถูกจำกัด ดังนี้

$$\begin{aligned}
\text{Minimize} \quad & x_1 + x_2 + x_3 \\
\text{Subject to} \quad & a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \geq 44 \\
& a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \geq 3 \\
& a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \geq 48 \\
& x_1, x_2, x_3 \geq 0 \text{ and integer}
\end{aligned}$$

โดยรูปแบบที่ 1 คือ $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ รูปแบบที่ 2 คือ $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ และ รูปแบบที่ 3 คือ $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ สำหรับตัวแปรพื้นฐานเริ่มต้น x_B สำหรับตัวอย่างนี้เราใช้ $x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ ดังนี้ เมทริกซ์พื้นฐานเริ่มต้น (initial basis matrix) สำหรับตัวอย่างนี้ เราใช้ $B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ เป็นเมทริกซ์พื้นฐานเริ่มต้น โดยในแต่ละคอลัมน์ของเมทริกซ์ B สอดคล้องกับรูปแบบในการตัดแผ่นอลูมิเนียม จากนั้นทำการหาคำตอบของปัญหาหลักที่ถูกจำกัด จะได้

$$x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = B^{-1}b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 44 \\ 3 \\ 48 \end{pmatrix} = \begin{pmatrix} 44 \\ 3 \\ 48 \end{pmatrix}$$

เพื่อตรวจสอบว่าคำตอบที่ได้เป็นคำตอบที่เหมาะสมที่สุดสำหรับปัญหาหลักหรือไม่ พิจารณาค่า reduced cost ซึ่งคำนวณจาก $c_j - \boldsymbol{\pi}^T \mathbf{a}_j$ โดย $\pi = \mathbf{c}_B^T B^{-1}$ คือ ค่าตัวแปรคู่ควบของคำตอบปัจจุบัน และ $\mathbf{a}_j$ คือ คอลัมน์ของตัวแปรไม่พื้นฐาน x_j สำหรับ $j = 1,2,3$ จะได้

$$c_j - \mathbf{c}_B^T B^{-1}\mathbf{a}_j = 1 - \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\mathbf{a}_j = 1 - \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} a_{1j} \\ a_{2j} \\ a_{3j} \end{pmatrix}$$

เนื่องจากเทคนิคคอลัมน์เจเนอเรชันส่วนของปัญหาย่อย คือ ส่วนที่ใช้ตรวจสอบว่าคำตอบ ณ ปัจจุบันที่ได้จากปัญหาหลักที่ถูกจำกัดนั้นเหมาะที่สุดหรือไม่ จากสมการดังกล่าว สามารถนำมาสร้างปัญหาย่อยได้ ดังนี้

$$\begin{array}{ll} \text{Minimize} & 1 - (\alpha_1 + \alpha_2 + \alpha_3) \\ \text{Subject to} & 81\alpha_1 + 70\alpha_2 + 68\alpha_3 \le 218 \\ & \alpha_1, \alpha_2, \alpha_3 \ge 0 \ \text{ and integer} \end{array}$$

โดย α_i เป็นตัวแปรตัดสินใจที่บอกถึงจำนวนแผ่นที่มความยาวแบบที่ i ที่จะถูกตัดจากแผ่นอลูมิเนียมมาตรฐาน สำหรับเงื่อนไขในปัญหาย่อย คือ รูปแบบการตัดแผ่นอลูมิเนียมจะต้องสอดคล้องกลับขนาดของแผ่นอลูมิเนียมมาตรฐาน ในที่นี้คือ การตัดแผ่นอลูมิเนียมด้วยความยาวต่างๆจะต้องมีความยาวไม่เกิน 218 เซนติเมตร ซึ่งพบว่าปัญหาย่อยที่ได้มีลักษณะของปัญหา knapsack problem เมื่อทำการแก้ปัญหาย่อยดังกล่าว จะได้ $\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}$ และค่าฟังก์ชันจุดประสงค์เท่ากับ -2 ซึ่งน้อยกว่า 0 แสดงว่าคำตอบในปัญหาหลักสามารถปรับปรุงค่าได้ จากปัญหาย่อยเราได้ตัวแปรไม่พื้นฐาน x_j (สมมติว่าเป็น x_4) ซึ่งสอดคล้องกับคอลัมน์ $\mathbf{a_4} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}$ โดยคอลัมน์ดังกล่าวแสดงรูปแบบการตัดแผ่นอลูมิเนียมสำหรับนำไปปรับปรุงค่าในปัญหาหลักต่อไป และเราหาตัวแปรนำออก (leaving variable) โดยการใช้ minimum ratio test ดังนี้

$$\frac{b_1}{\alpha_1} = \frac{44}{0} = \infty$$

$$\frac{b_2}{\alpha_2} = \frac{3}{0} = \infty$$

$$\frac{b_3}{\alpha_3} = \frac{48}{3} = 16$$

จากค่า minimum ratio test ที่ได้พบว่า ตัวแปร x_3 เป็นตัวแปรนำออก และคอลัมน์ใหม่สามารถสร้างได้จาก

$$B^{-1}\mathbf{a_4} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}$$

ดังนั้น เมทริกซ์พื้นฐาน (basis) ใหม่ คือ $B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix}$

ดั้งนั้นคำตอบใหม่ในปัญหาหลัก คือ $x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_4 \end{pmatrix} = B^{-1}b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{3} \end{pmatrix}\begin{pmatrix} 44 \\ 3 \\ 48 \end{pmatrix} = \begin{pmatrix} 44 \\ 3 \\ 16 \end{pmatrix}$

ทำซ้ำกระบวนการดังกล่าวไปจนกระทั่งไม่สามารถหาคอลัมน์ที่สามารถนำมาปรับปรุงคำตอบในปัญหาหลักได้ นั่นคือได้ค่าที่เหมาะที่สุดสำหรับปัญหาหลัก

2.2 ผลงานวิจัยที่เกี่ยวข้อง

ปัญหาการจัดตารางปฏิบัติงานของพนักงานสายการบิน เป็นปัญหาที่มีนักวิจัยสนใจมาหลายปีแล้ว จากการศึกษางานวิจัยพบว่ามีการนำเสนอการแก้ปัญหาดังกล่าวด้วยวิธีต่างๆ มากมาย ทั้งทางด้านที่ใช้การสร้างตัวแบบทางคณิตศาสตร์ในการแก้ปัญหา เช่น การสร้างตัวแบบทางคณิตศาสตร์ในรูปแบบเซตคาฟเวอริ่ง ([2], [3]) และใช้เทคนิคต่างๆในการแก้ปัญหา อย่างเช่น branch and bound, branch and cut, Lagragian lower bound [4], partially integrated approach [5], column generation ([6], [7], [8], [9], [10], [11], [12]), 0-1 multicommodity flow ([13], [14]) และอีกด้านหนึ่งคือการแก้ปัญหาโดยขั้นตอนวิธีทางฮิวริสติก เช่น scatter search heuristic [15], tabu search [16], simulated annealing [17], particle swarm optimization [18] ขั้นตอนวิธีทางพันธุกรรม (genetic algorithm) [19] ทั้งนี้โดยได้

รวบรวมและสรุปสาระสำคัญของงานวิจัยต่างๆที่เกี่ยวข้องกับการแก้ปัญหาการจัดตารางปฏิบัติงานของพนักงานสายการบิน เพื่อให้เห็นถึงแนวความคิดของปัญหา ดังต่อไปนี้

กชพร อ้นสวน, บุญฤทธิ์ อินทิยศ, ชวลิต จีนอนันต์ (2011) [19] ศึกษาปัญหาการมอบหมายงานให้กับพนักงานสายการบินโดยเสนอขั้นตอนวิธีทางพันธุกรรม (genetic algorithm) ในการแก้ปัญหาการมอบหมายงานให้กับพนักงานสายการบินของบริษัทการบินไทย โดยมีจุดประสงค์ของงานเพื่อกระจายรายได้และภาระงานของพนักงานแต่ละคนอย่างเท่าเทียมกัน งานวิจัยนี้ทำการทดลองโดยเปลี่ยนค่าพารามิเตอร์ซึ่งประกอบด้วยจำนวนโครโมโซมและจำนวนรุ่น เมื่อวิเคราะห์ความแปรปรวนพบว่าจำนวนรุ่นมากให้คำตอบที่ดีกว่า แต่จำนวนโครโมโซมมากไม่จำเป็นต้องให้คำตอบที่ดีกว่าเสมอไป ซึ่งจากผลการทดลองจำนวนโครโมโซมที่เหมาะสมคือ 300 และจำนวนรุ่นที่เหมาะสม คือ 30 วิธีการนี้ใช้เวลาคำนวณผลการทดลองอยู่ในช่วง 6 นาที-2 ชั่วโมง

Chawalit Jeenanunta, Boonyarit Intiyot, Wariya Puttapatimok, (2010) [14] ศึกษาปัญหาการมอบหมายงานให้กับพนักงานสายการบิน โดยสร้างแบบจำลองทางคณิตศาสตร์การไหลของโภคภัณฑ์หลายชนิด (multi-commodity flow) ในการแก้ปัญหาการมอบหมายงานให้กับพนักงานสายการบินของบริษัทการบินไทย โดยมีจุดประสงค์ของงานเพื่อกระจายรายได้และภาระงานของพนักงานแต่ละคนอย่างเท่าเทียมกัน งานวิจัยนี้พบว่า เวลาในการคำนวณคำตอบแต่ละชุดข้อมูลใช้เวลานานและเมื่อปัญหามีขนาดใหญ่จะมีปัญหาเรื่องหน่วยความจำคอมพิวเตอร์ไม่เพียงพอ

Goran Stojkovic, François Soumis, Jacques Desrosiers, Marius M. Solomon (2002) [2] ศึกษาปัญหาการจัดตารางการบิน โดยสร้างแบบจำลองทางคณิตศาสตร์ โดยมีจุดประสงค์เพื่อจับคู่เที่ยวบินที่ทำให้ช่วงเวลาตั้งแต่ออกจากฐานจนกระทั่งกลับฐานมีค่าน้อยที่สุด สำหรับการจัดตารางการบินใหม่เมื่อเกิดเหตุการณ์ไม่คาดคิดภายในสายการบิน โดยคำนึงถึงข้อจำกัดเกี่ยวกับเวลาที่จุดเริ่มต้น จุดสิ้นสุด และเวลาระหว่างการบิน และได้ทำการแปลงแบบจำลองทางคณิตศาสตร์ให้อยู่ในรูปแบบจำลองการไหลของโภคภัณฑ์ (network flow) ผลการศึกษาพบว่าเมื่อนำแบบจำลองทางคณิตศาสตร์เริ่มต้นมาหาผลลัพธ์ด้วยวิธีการซิมเพล็กซ์เวลาในการคำนวณจะเพิ่มขึ้นแบบโพลิโนเมียลตามขนาดของปัญหาที่ใหญ่ขึ้น และเมื่อนำแบบจำลองทางคณิตศาสตร์ในรูปแบบจำลองการไหลของโภคภัณฑ์มาแปลงเป็นรูปแบบคู่ควบแล้วเวลาที่ใช้คำนวณจะเพิ่มในลักษณะเส้นตรงตามขนาดของปัญหา

Shervin Ahmad Beygi, Amy Cohn, Marshall Weir (2009) [3] ศึกษาปัญหาการจับคู่เที่ยวบิน โดยอาศัยเทคนิคคอลัมน์เจเนอเรชันมาช่วยในการคำนวณ โดยสร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหาย่อยในรูปแบบกำหนดการเชิงจำนวนเต็ม (integer programming) โดยมีจุดประสงค์สำหรับปัญหาเพื่อหาคู่เที่ยวบินที่มีผลรวมของเวลาน้อยที่สุด แบบจำลองนี้อาศัยตัวแปรที่บอกว่าเที่ยวบินนั้นควรอยู่ตำแหน่งใดของคู่เที่ยวบิน (marker variables) กับตัวแปรบอกความต่อเนื่องของเที่ยวบิน (connection variables) มาช่วยในการสร้างแบบจำลอง งานวิจัยนี้ทำการทดลองกับข้อมูลของสายการบินหนึ่งในประเทศสหรัฐอเมริกา โดยใช้เที่ยวบินในการศึกษาจำนวน 143 เที่ยวบิน วิธีการนี้ใช้เวลาคำนวณผลการทดลองอยู่ในช่วง 194 วินาที-252 วินาที

Sylvie Lavoie, Michel Minoux, Edouard Odier (1988) [4] ศึกษาปัญหาการจับคู่เที่ยวบิน โดยสร้างแบบจำลองทางคณิตศาสตร์ในรูปแบบของเซตคาฟเวอริ่ง โดยมีจุดประสงค์เพื่อจับคู่เที่ยวบินที่ทำให้ช่วงเวลาตั้งแต่ออกจากฐานจนกระทั่งกลับฐานมีค่าน้อยที่สุด อีกทั้งสอดคล้องกับเงื่อนไขเกี่ยวกับชั่วโมงการบินและช่วงเวลาพักผ่อนของพนักงานสายการบินแอร์ฟราซน์ และประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชันในการแก้ปัญหา โดยแบ่งการทดลองออกเป็น 2 ชุด ชุดแรกในแต่ละข้อมูลมีความแตกต่างกันของจำนวนเที่ยวบิน เวลาที่เครื่องบินบินออกหรือเวลาที่เครื่องบินบินกลับ ผลการทดลองพบว่าประสิทธิของเวลาที่ใช้คำนวณในแต่ละข้อมูลอยู่ในวิสัยที่ดี ชุดที่สองใช้ข้อมูลชุดเดียวกัน ทำการทดลองซ้ำ 6 ครั้ง เพื่อทดสอบประสิทธิภาพของเวลาในการคำนวณตัวแบบ โดยใช้เที่ยวบินจำนวน 132 เที่ยวบิน พบว่า เวลาคำนวณผลการทดลองอยู่ในช่วง 52 วินาที-79 วินาที

สำหรับงานวิจัยนี้นำเสนอตัวแบบทางคณิตศาสตร์สำหรับปัญหาการจับคู่เที่ยวบินโดยใช้เงื่อนไขบังคับเกี่ยวกับระยะเวลาการทำงานและการพักของพนักงานสายการบินจากบริษัทการบินไทยในการหาคำตอบ โดยการสร้างตัวแบบทางคณิตศาสตร์ในรูปแบบเซตคาฟเวอริ่งและประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชันร่วมแก้ปัญหาดังกล่าวซึ่งเป็นเทคนิคที่นิยมใช้แก้ปัญหาที่มีขนาดใหญ่ เพราะช่วยหลีกเลี่ยงปัญหาเกี่ยวกับหน่วยความจำของคอมพิวเตอร์และช่วยลดเวลาที่ใช้ในการหาคำตอบ

บทที่ 3

ตัวแบบทางคณิตศาสตร์และวิธีการหาคำตอบ

3.1 ข้อมูลทั่วไปของบริษัทการบินไทย

บริษัทการบินไทย จำกัด (มหาชน) เป็นรัฐวิสาหกิจภายใต้สังกัดกระทรวงคมนาคม โดยให้บริการทางด้านการบินพาณิชย์ในประเทศ และระหว่างประเทศ ซึ่งบริษัทการบินไทยเป็นสายการบินแห่งชาติของประเทศไทย และเป็นรัฐวิสาหกิจที่สามารถทำกำไรให้กับประเทศไทยมาเป็นระยะเวลานาน อีกทั้งยังได้รับการยกย่องให้อยู่ในระดับสายการบินชั้นนำของโลกอีกด้วย ซึ่งการดำเนินงานของบริษัทการบินไทยแบ่งออกเป็นหลายส่วนงานด้วยกัน สำหรับขอบเขตของงานวิจัยนี้จะพิจารณาปัญหาการจัดตารางปฏิบัติงานของพนักงานสายการบิน โดยพิจารณาเฉพาะปัญหาการจับคู่เที่ยวบินของเที่ยวบินภายในประเทศเท่านั้น

การบินไทยใช้ข้อจำกัดเวลาทำการบินและเวลาปฏิบัติหน้าที่ตามประกาศกรมการบินพลเรือนในการจัดตารางปฏิบัติงานของพนักงานสายการบิน ตามประกาศกำหนดในเรื่องข้อจำกัดเวลาทำการบินและปฏิบัติหน้าที่ไว้ดังต่อไปนี้

- ชั่วโมงบิน (flight time/block time) หมายถึง เวลาตั้งแต่เครื่องบินบินขึ้นจากสนามบินต้นทางจนกระทั่งลงจอด ณ สนามบินปลายทาง
- ช่วงเวลาปฏิบัติหน้าที่การบิน (flight duty period) หมายถึง ช่วงเวลาต่อเนื่องที่พนักงานการบินปฏิบัติหน้าที่ โดยจะนับตั้งแต่หนึ่งชั่วโมงก่อนเครื่องบินบินขึ้น จนถึงสามสิบนาทีหลังเครื่องบินร่อนลงครั้งสุดท้าย
- ช่วงเวลาพักผ่อน (rest time) หมายถึง ช่วงเวลาที่พนักงานพ้นจากการปฏิบัติหน้าที่

บริษัทการบินไทยใช้กฎแรงงานประกอบในการจัดตารางงานปฏิบัติงานของพนักงาน โดยกฎแรงงานจะครอบคลุมเวลาการทำงานและเวลาพักดังนี้

- ภายในทุกๆ 7 วัน ต่อเนื่องกัน พนักงานจะมีชั่วโมงบินได้ไม่เกิน 34 ชั่วโมง
- ภายในทุกๆ 28 วัน ต่อเนื่องกัน พนักงานจะมีชั่วโมงบินได้ไม่เกิน 110 ชั่วโมง
- ภายในทุกๆ 365 วัน ต่อเนื่องกัน พนักงานจะมีชั่วโมงบินได้ไม่เกิน 1,000 ชั่วโมง

ตารางที่ 3.1 เวลาปฏิบัติหน้าที่การบินและเวลาพัก

เวลาปฏิบัติหน้าที่การบิน	เวลาพัก
< 8 ชั่วโมง	≥ 8 ชั่วโมง
8 - 10 ชั่วโมง	≥ 10 ชั่วโมง
10 - 12 ชั่วโมง	≥ 12 ชั่วโมง
12 - 14 ชั่วโมง	≥ 14 ชั่วโมง
14 - 16 ชั่วโมง	≥ 16 ชั่วโมง
16 - 20 ชั่วโมง	≥ 24 ชั่วโมง

ตารางที่ 3.1 กล่าวถึงเวลาปฏิบัติหน้าที่การบินและการพักของพนักงานสายการบิน คือ ช่วงเวลาปฏิบัติหน้าที่การบินไม่เกิน 8 ชั่วโมง ต้องมีช่วงเวลาพักผ่อนต่อเนื่องไม่น้อยกว่า 8 ชั่วโมง จึงสามารถให้ปฏิบัติหน้าที่การบินต่อได้ และช่วงเวลาปฏิบัติหน้าที่การบินตั้งแต่ 8 ถึง 10 ชั่วโมง ต้องมีช่วงเวลาพักผ่อนต่อเนื่องไม่น้อยกว่า 10 ชั่วโมง เป็นต้น

3.2 แนวคิดของปัญหา

จากการศึกษางานวิจัย A Multi-commodity Flow Approach to the Crew Rostering Problem (2010) [13] ซึ่งศึกษาปัญหาการมอบหมายงานให้กับพนักงานสายการบินพบว่า เวลาในการคำนวณคำตอบแต่ละชุดข้อมูลใช้เวลานานและเมื่อปัญหามีขนาดใหญ่จะมีปัญหาเรื่องหน่วยความจำคอมพิวเตอร์ไม่เพียงพอ โดยสาเหตุหนึ่งที่ทำให้ปัญหาการมอบหมายงานให้กับพนักงานสายการบินมีขนาดใหญ่และซับซ้อนก็คือการที่มีคู่เที่ยวบินที่ต้องคำนึงถึงในแต่ละวันมากเกินไปดังนั้นเราจึงมีแนวคิดว่าถ้าเราสามารถจับคู่เที่ยวบินให้ยาวขึ้นโดยจัดให้สอดคล้องกับข้อจำกัดในเรื่องกฎแรงงานและเวลาพักไว้ก่อน ขั้นตอนการมอบหมายงานให้กับพนักงานสายการบินจะทำได้ง่ายขึ้น

ในงานวิจัยนี้เรานำเสนอตัวแบบทางคณิตศาสตร์ในรูปแบบกำหนดการเชิงจำนวนเต็มสำหรับแก้ปัญหาการจับคู่เที่ยวบินตามเงื่อนไขการทำงานและการพักของพนักงานสายการบิน

บริษัทการบินไทย สำหรับงานวิจัยนี้ได้นำเสนอการจับคู่เที่ยวบินที่ยาวขึ้นในลักษณะของทัวร์เพื่อลดคำตอบที่เป็นไปได้ที่เกิดขึ้นในขั้นตอนการมอบหมายงานให้กับพนักงานสายการบิน ทัวร์ในงานวิจัยนี้นิยามไว้ดังนี้ คือ ลำดับของเที่ยวบินใน d วัน โดยพิจารณาฐานที่เครื่องบินออกและบินกลับจะต้องเป็นฐานเดียวกัน ซึ่งประกอบด้วยคู่เที่ยวบินหลายคู่ จุดประสงค์ของตัวแบบดังกล่าวคือการจัดทัวร์ที่ครอบคลุมทุกเที่ยวบินโดยให้มีผลรวมของช่วงเวลาพักระหว่างเที่ยวบิน (sit time) โดยรวมน้อยที่สุดและมีเวลาพักเป็นไปตามเงื่อนไขในตารางที่ 3.1 และในงานวิจัยนี้เราใช้เทคนิคคอลัมน์เจเนอเรชัน เพื่อการหาคำตอบที่เร็วขึ้นและช่วยแก้ปัญหาในส่วนของหน่วยความจำคอมพิวเตอร์

3.3 ตัวแบบทางคณิตศาสตร์

3.3.1 ปัญหาหลัก (Master problem)

ตัวแบบทางคณิตศาสตร์ในส่วนของปัญหาหลักเป็นดังนี้

$$\text{Minimize} \quad \sum_{j \in T} c_j t_j \tag{1}$$

$$\text{Subject to} \quad \sum_{j \in T} a_{ij} t_j \geq 1 \qquad \forall i \in F \tag{2}$$

$$t_j \in \{0,1\} \qquad \forall j \in T$$

โดยที่ F แสดง เซตของเที่ยวบินทั้งหมด

T แสดง เซตทัวร์ทั้งหมดที่เป็นไปได้

c_j แสดง ช่วงเวลาพักโดยรวมระหว่างเที่ยวบิน ของทัวร์ ที่ j

t_j แสดง ตัวแปรในการตัดสินใจของทัวร์ที่ j ซึ่ง t_j จะมีค่าเป็น 1 เมื่อทัวร์ที่ j ถูกเลือกและเป็น 0 ในกรณีอื่นๆ

a_{ij} แสดง สัมประสิทธิ์ที่บอกความสัมพันธ์ระหว่างเที่ยวบิน i และทัวร์ j ซึ่ง $a_{i,j}$ จะมีค่าเป็น 1 เมื่อเที่ยวบินที่ i ถูกใช้ในทัวร์ที่ j และเป็น 0 ในกรณีอื่นๆ

ตัวแบบนี้มีวัตถุประสงค์เพื่อจับคู่เที่ยวบินในรูปแบบของทัวร์ให้ได้ทัวร์ที่มีผลรวมของช่วงเวลาพักระหว่างเที่ยวบินน้อยที่สุด และสอดคล้องกับเงื่อนไขที่ (2) คือ ทุกเที่ยวบินจะต้องถูกนำมาจัดในทัวร์อย่างน้อย 1 ครั้ง

เริ่มต้นปัญหาดังกล่าวจะถูกจำกัดโดยพิจารณา t_j จากเซ็ต $\mathrm{T}' \subset \mathrm{T}$ ที่เล็กลงและแก้ปัญหา LP relaxation ซึ่งสามารถหาคำตอบคู่ควบ (dual solution) $\mathbf{w}$ จากคำตอบที่ได้ จากนั้นจะเข้าสู่การแก้ปัญหาย่อย โดยการนำคำตอบคู่ควบที่ได้ไปตรวจสอบว่าคำตอบในปัญหาหลักเหมาะที่สุดหรือไม่ ซึ่งดูจาก reduced cost ที่น้อยที่สุด ดังนี้

$$\underset{j \in T}{\mathrm{Min}}\,(c_j - \mathbf{w}^T \mathbf{a}_j) \tag{3}$$

เมื่อ $\mathbf{a}_j$ คือ คอลัมน์ของตัวแปรไม่พื้นฐาน t_j ซึ่งเป็นแบบรูป (pattern) ของเที่ยวบินที่ถูกใช้ในทัวร์ที่ j โดยจะต้องเป็นแบบรูปของเที่ยวบินหรือทัวร์ที่เป็นไปได้ (feasible tour) กล่าวคือเวลาที่ให้พักจะต้องสอดคล้องกับกฎแรงงานและเวลาพักของพนักงานสายการบิน และ c_j คือ ค่าใช้จ่ายของตัวแปร t_j ซึ่งในที่นี้คือ ช่วงเวลาพักระหว่างเที่ยวบินของทัวร์ที่ j

หากค่าที่ได้จากสมการที่ (3) มีค่ามากกว่าหรือเท่ากับศูนย์ แสดงว่าคำตอบ ณ ปัจจุบันของปัญหาหลักเหมาะที่สุดแล้ว ในทางกลับกันหากค่าที่ได้มีค่าน้อยกว่าศูนย์ แสดงว่าปัญหาหลักยังสามารถปรับปรุงคำตอบให้เหมาะสมยิ่งขึ้นได้ โดยนำคอลัมน์ที่ได้เพิ่มไปในคำตอบของปัญหาหลักที่ถูกจำกัด

ในการหาคำตอบเริ่มต้น (initial solution) โดยทั่วไปแล้วสามารถเลือกตัวแปรมาเป็นตัวแปรพื้นฐาน โดยจำนวนตัวแปรดังกล่าวต้องเท่ากับจำนวนแถว (หรือจำนวนเงื่อนไขข้อจำกัด) สำหรับปัญหาการจับคู่เที่ยวบิน นั่นหมายถึง เราต้องสร้างแบบรูปการบินของตัวแปรทัวร์เพื่อนำมาเป็นตัวแปรพื้นฐาน ซึ่งจะต้องใช้เวลาในการหาแบบรูปที่เป็นไปได้ ดังนั้นเพื่อความสะดวกรวดเร็วเราจะใช้ตัวแปรเทียม (artificial variable) ซึ่งมีแบบรูปการบินที่ครอบคลุมเพียงหนึ่งเที่ยวบินโดยแบบรูปดังกล่าวเป็นแบบรูปของทัวร์ที่เป็นไปไม่ได้ (infeasible tour) มาเป็นตัวแปรพื้นฐาน ดังนั้นเมทริกซ์สัมประสิทธิ์ของตัวแปรพื้นฐานเริ่มต้นจึงมีลักษณะเป็นเมทริกซ์เอกลักษณ์ (identity matrix) ดังแสดงในภาพที่ 3.1

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

ภาพที่ 3.1 ตัวอย่างเมทริกซ์สัมประสิทธิ์ของตัวแปรพื้นฐานในลักษณะของเมทริกซ์เอกลักษณ์ที่ใช้เป็นคำตอบเริ่มต้นของปัญหาหลักที่ถูกจำกัด

เนื่องจากเมทริกซ์ในภาพที่ 3.1 เป็นเมทริกซ์ที่มีตัวผกผัน เราสามารถหาค่า reduced cost ได้ โดยกำหนดค่าช่วงเวลาพักระหว่างเที่ยวบิน (cost) ของตัวแปรเทียมเป็นค่าที่มีค่ามากๆ ซึ่งจากกลไกของขั้นตอนวิธีจะเหนี่ยวนำให้ตัวแปรเทียมกลายเป็นตัวแปรไม่พื้นฐาน หรือกลายเป็นศูนย์

3.3.2 ปัญหาย่อย (Subproblem)

ปัญหาย่อยเป็นการสร้างทัวร์ที่เป็นไปได้ที่สามารถปรับปรุงผลเฉลยของปัญหาหลักเพื่อใช้เพิ่มเข้าไปในคอลัมน์ของปัญหาหลักที่ถูกจำกัด ในการแก้ปัญหาย่อยเราจะใช้ตัวแบบในลักษณะของโครงข่ายงานที่สอดคล้องกับเงื่อนไขระยะเวลาการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย

สำหรับการตรวจสอบคำตอบของปัญหาหลักว่าให้ค่าที่ดีแล้วหรือไม่ หรือการตรวจสอบค่า reduced cost ดังสมการ (3) เนื่องจาก c_j เป็นค่าที่ยังไม่ทราบค่าจนกว่าจะรู้แบบรูปของทัวร์ j ในเบื้องต้นเราจะพิจารณาเฉพาะปัญหา $\underset{j \in T}{\text{Min}}\,(-\mathbf{w}^T \mathbf{a}_j)$ หรือพิจารณาจากปัญหา

$$\underset{j \in T}{\text{Max}}\,(\mathbf{w}^T \mathbf{a}_j) \tag{4}$$

โดยที่ $\mathbf{w}$ คือ เวกเตอร์ของตัวแปรคู่ควบที่ได้จากปัญหาหลักที่ถูกจำกัด

$\mathbf{a}_j$ คือ คอลัมน์ของทัวร์ที่ j

ในการแก้ปัญหา (4) เราจะไม่แจง $\{\mathbf{a}_j \mid j \in T\}$ ออกมาตรงๆ แต่จะเขียนแทนเซตดังกล่าวโดยนัยโดยเขียนเป็นเงื่อนไขบังคับสำหรับแบบรูปทัวร์ที่เป็นไปได้เป็นเวกเตอร์ของตัวแปร $\mathbf{a} = \begin{bmatrix} x_{p_1q_1} \\ \vdots \\ x_{p_mq_m} \end{bmatrix}$ โดยที่ $x_{p_iq_i} \in \{0,1\}$ โดยเป็น 1 เมื่อเที่ยวบินที่บินออกจากจุดต่อ p_i ไปยังจุดต่อ q_i ถูกนำมาจัดอยู่ในทัวร์ที่ j และเป็น 0 ในกรณีที่ไม่ใช่ ยกตัวอย่างเช่น ถ้าเที่ยวบินที่ 3, 4, 7 และ 8 ถูกนำมาจัดเป็นทัวร์จะได้คอลัมน์ $\mathbf{a} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$ เนื่องจากเที่ยวบินแต่ละเที่ยวบินจะมีข้อมูลของเวลาที่เครื่องบินบินออก เวลาที่เครื่องบินบินกลับ และสนามบินซึ่งเป็นจุดเริ่มต้นและสิ้นสุดของเที่ยวบิน เมื่อเราทราบคอลัมน์ $\mathbf{a}$ และข้อมูลดังกล่าวของเที่ยวบิน จะสามารถหาลำดับการใช้เที่ยวบินภายในทัวร์นั้นๆได้ โดยแบบรูปทัวร์จะถือเป็นทัวร์ (หรือคอลัมน์) ที่เป็นไปได้ ก็ต่อเมื่อ ทัวร์ดังกล่าวสอดคล้องกับเงื่อนไขระยะเวลาการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย เรานำเงื่อนไขดังกล่าวมาเขียนในลักษณะของโครงข่ายงานซึ่งอธิบายไว้ในหัวข้อ 3.3.2.1 และ 3.3.2.2 ดังนั้นปัญหา (4) จึงถูกแปลงเป็นปัญหาย่อย ดังต่อไปนี้ กำหนดให้ $G = (N, A)$ เมื่อ N คือเซตของจุดต่อ และ A คือ เซตของอาร์ก

ตัวแบบทางคณิตศาสตร์ของปัญหาย่อยเป็นดังนี้

$$\text{Maximize} \quad \sum_{(s,t)\in A} w_{st} x_{st} \tag{5}$$

$$\text{Subject to} \quad \sum_{s:(s,t)\in A} x_{st} - \sum_{s:(t,s)\in A} x_{ts} = 0 \quad \forall t \in N \tag{6}$$

$$\sum_{k=d}^{d+6} \sum_{(s,t)\in A_d} p_{st} x_{st} \le 34 \quad \forall d \in D \tag{7}$$

$$x_{st} \in \{0,1\}, (s,t) \in A, d = 1,2,3$$

โดยที่ w_{st} แสดง ค่าของตัวแปรคู่ควบที่ได้จากปัญหาหลักหรือค่าใช้จ่าย (cost) สำหรับอาร์กจากจุดต่อ s ไปยังจุดต่อ t

x_{st} แสดง ตัวแปรในการตัดสินใจ เป็นค่าที่แสดงการไหลจากจุดต่อ s ไปจุดต่อ t โดยจะมีค่าเป็น 1 เมื่อมีการไหลจากจุดต่อ s ไปจุดต่อ t และเป็น 0 ในกรณีที่ไม่มีการไหล

p_{st} แสดง ช่วงเวลาปฏิบัติงานจากจุดต่อ s ไปจุดต่อ t

A_d แสดง เซตของอาร์กที่เริ่ม ณ วันที่ d

วัตถุประสงค์ของปัญหา (5) ต้องการหาการไหล จากจุดเริ่มต้น (source) ไปยังจุดสิ้นสุด (sink) ที่ทำให้มีค่าใช้จ่ายมากที่สุด ที่ต้องสอดคล้องกับเงื่อนไข (6) ซึ่งเป็นเงื่อนไขการอนุรักษ์การไหล (flow conservation) และเงื่อนไขที่ (7) ซึ่งเป็นเงื่อนไขเกี่ยวกับจำนวนชั่วโมงการปฏิบัติงานที่ว่าภายในทุกๆ 7 วัน ต่อเนื่องกัน พนักงานจะมีชั่วโมงบินได้ไม่เกิน 34 ชั่วโมง เนื่องจากในขั้นตอนการมอบหมายงานให้กับพนักงานสายการบิน พนักงานแต่ละคนจะถูกจับคู่กับทัวร์อย่างน้อยหนึ่งทัวร์ ดังนั้นทัวร์ที่เป็นไปได้จะต้องมีชั่วโมงบินไม่เกิน 34 ชั่วโมงภายใน 7 วันใดๆที่ติดกัน

จากตัวแบบทางคณิตศาสตร์สำหรับปัญหาย่อยเราจะได้ค่า $\underset{j\in T}{\text{Min}}(-\mathbf{w}^T\mathbf{a}_j)$ พร้อมทั้งคอลัมน์ $\mathbf{a}_j$ ที่ทำให้ค่าดังกล่าวน้อยที่สุด ดังนั้นจึงสามารถหาค่า c_j ได้ แล้วจึงนำค่าที่ได้ไปตรวจสอบค่า reduced cost หรือ $c_j - \mathbf{w}^T\mathbf{a}_j$ ด้วยวิธีการดังกล่าวทัวร์สุดท้ายที่ได้ไม่สามารถรับประกันได้ว่าเป็นทัวร์ที่มี reduced cost ต่ำที่สุด อย่างไรก็ตามสำหรับงานวิจัยนี้มีเงื่อนไขในการหาคำตอบสำหรับปัญหาการจับคู่เที่ยวบินดังนี้ คือ หากค่าที่ได้จากปัญหา (4) ให้ค่า reduced cost ที่มากกว่าหรือเท่ากับศูนย์ แสดงว่าคำตอบ ณ ปัจจุบันของปัญหาหลักได้ค่าที่เหมาะสำหรับปัญหาการจับคู่เที่ยวบินแล้ว ในทางกลับกันหากค่าที่ได้มีค่าน้อยกว่าศูนย์ คำตอบในปัญหาย่อยจะบ่งบอกถึงคอลัมน์ ที่จะถูกนำไปใช้ในการปรับปรุงคำตอบของปัญหาหลักได้

3.3.2.1 ลักษณะข้อมูลที่นำมาสร้างโครงข่ายงาน

ตัวอย่างข้อมูลของบริษัทการบินไทยที่นำมาใช้ในการสร้างโครงข่ายงาน มีดังนี้

ตารางที่ 3.2 ตัวอย่างข้อมูลที่นำมาสร้างโครงข่ายงาน

เที่ยวบิน	1	2	3	4	5	6	7	8
สนามบินต้นทาง	BKK	CNX	BKK	HDY	BKK	CNX	BKK	HDY
สนามบินปลายทาง	CNX	BKK	HDY	BKK	CNX	BKK	HDY	BKK
วันที่ออกเดินทาง	1	1	1	1	2	2	2	2
เวลาที่ออกเดินทาง	6:15	8:10	6:05	8:15	6:15	8:10	6:05	8:15
วันที่เดินทางกลับ	1	1	1	1	2	2	2	2
เวลาที่ถึงปลายทาง	7:25	9:20	7:35	9:40	7:25	9:20	7:35	9:20
ชั่วโมงบิน (ชั่วโมง:นาที)	1:10	1:10	1:30	1:25	1:10	1:10	1:30	1:25

ตารางที่ 3.2 แสดงตัวอย่างข้อมูลที่นำมาใช้ในการสร้างโครงข่ายงาน ยกตัวอย่างการนำข้อมูลจากตารางที่ 3.2 มาใช้ เช่น เที่ยวบินที่ 1 บินจากสนามบินกรุงเทพไปยังสนามบินเชียงใหม่ สามารถนำมาจัดทัวร์โดยจะต้องจับคู่กับเที่ยวบินที่เริ่มบินจากสนามบินเชียงใหม่ โดยเที่ยวดังกล่าวจะต้องมีเวลาการออกเดินทางจากเชียงใหม่มากกว่าเวลาที่เที่ยวบินที่ 1 บินมาถึงสถานีปลายทาง และต้องสอดคล้องกับเงื่อนไขการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย จะได้ทัวร์ที่เป็นไปได้สำหรับการเริ่มต้นจากเที่ยวบินที่ 1 ดังนี้

แบบที่ 1 เที่ยวบินที่ 1 ตามด้วย เที่ยวบินที่ 2

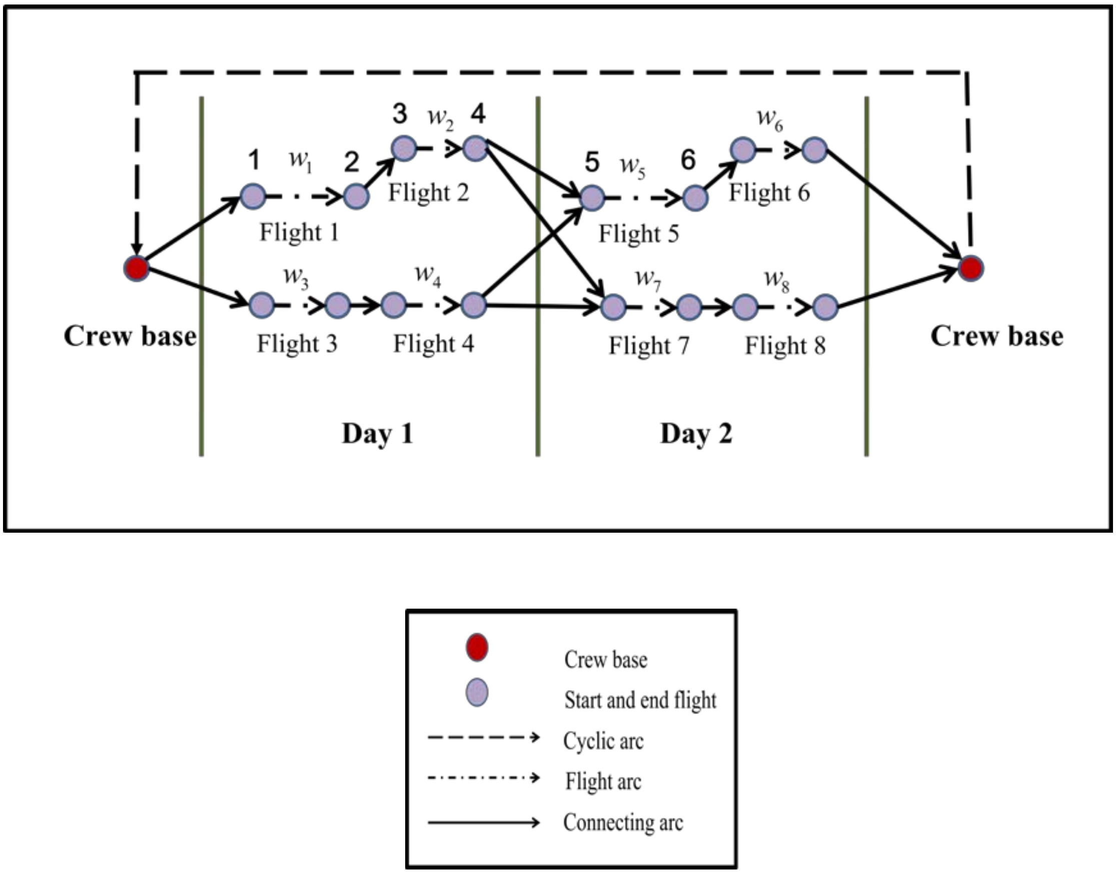
แบบที่ 2 เที่ยวบินที่ 1 ตามด้วยเที่ยวบินที่ 6

แบบที่ 3 เที่ยวบินที่ 1 ตามด้วย เที่ยวบินที่ 2 ตามด้วยเที่ยวบินที่ 5 ตามด้วยเที่ยวบินที่ 6

แบบที่ 4 เที่ยวบินที่ 1 ตามด้วย เที่ยวบินที่ 2 ตามด้วยเที่ยวบินที่ 7 ตามด้วยเที่ยวบินที่ 8

3.3.2.2 โครงข่ายงาน

งานวิจัยนี้ในขั้นตอนปัญหาย่อยใช้ตัวแบบในลักษณะของโครงข่ายงานที่สอดคล้องกับเงื่อนไขระยะเวลาการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย โดยตัวอย่างปัญหาย่อยในลักษณะของโครงข่ายงานแสดงในภาพที่ 3.2



ภาพที่ 3.2 ตัวอย่างปัญหาย่อยในลักษณะของโครงข่ายงาน

โครงข่ายงานที่สร้างประกอบไปด้วยจุดต่อ (node) และอาร์ก (arc) ดังนี้

crew base คือ จุดต่อที่แสดงจุดต่อเริ่มต้นการบิน (source) และจุดต่อสิ้นสุดการบิน (sink) ซึ่งจุดดังกล่าวจะต้องเป็นสนามบินที่เป็นฐานของพนักงานสายการบิน

start and end flight คือ จุดต่อที่ใช้แทนเวลาเริ่มต้นและสิ้นสุดของแต่ละเที่ยวบิน

cyclic arc คือ อาร์กที่เชื่อมจุดต่อสิ้นสุดการบินมายังจุดต่อเริ่มต้นการบิน

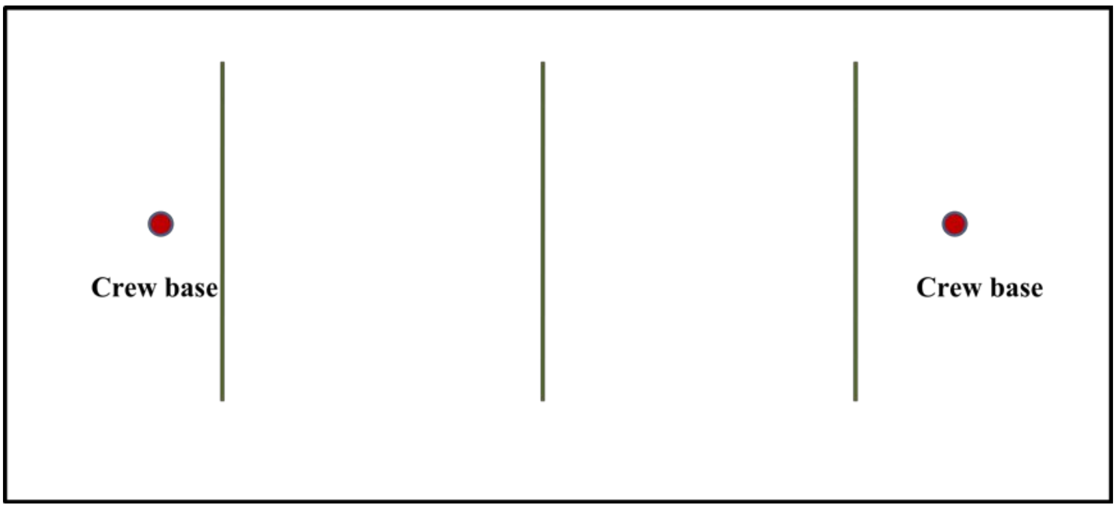
flight arc คือ อาร์กที่เชื่อมจุดต่อเริ่มต้นและจุดต่อสิ้นสุดของเที่ยวบิน

connecting arc คือ อาร์กที่แสดงความต่อเนื่องของแต่ละเที่ยวบิน

สำหรับค่าใช้จ่ายบน flight arc จะเป็นค่าจากตัวแปรคู่ควบจากปัญหาหลักที่ถูกจำกัด ส่วน connecting arc จะมีค่าใช้จ่ายเป็นศูนย์ จากโครงข่ายงานเราจะหาวิถีที่มีการไหลมากที่สุดและสอดคล้องกับเงื่อนไขต่างๆ จากภาพที่ 3.2 ตัวอย่างของช่วงเวลาพักระหว่างเที่ยวบิน (sit time) คือ เวลาระหว่างจุดต่อที่ 2 และ 3 และตัวอย่างช่วงเวลาพักที่ต้องเป็นไปตามตารางที่ 1 คือ เวลาระหว่างจุดต่อที่ 4 และ 5

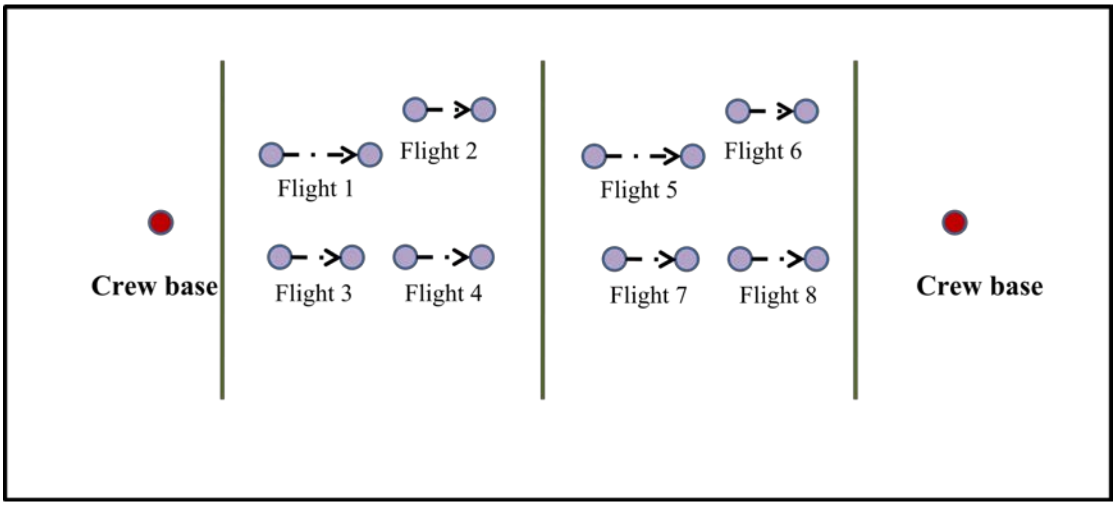
สำหรับขั้นตอนการสร้างโครงข่ายงานสำหรับการจัดทัวร์สำหรับ 2 วัน แสดงดังต่อไปนี้

1. กำหนด จุดต่อเริ่มต้นและจุดต่อสิ้นสุดของทัวร์ โดยจุดเริ่มต้นหรือสนามบินต้นทางและจุดสิ้นสุดหรือสนามบินปลายทางจะต้องเป็นสนามบินเดียวกัน โดยสนามบินดังกล่าวจะต้องเป็นสนามบินที่ใช้เป็นจุดเริ่มต้นและสิ้นสุดการทำงานของพนักงานสายการบิน (crew base) สำหรับเที่ยวบินภายในประเทศของบริษัทการบินไทย สนามบินที่ใช้เป็นจุดเริ่มต้นและสิ้นสุดการทำงานของพนักงานสายการบิน คือ สนามบินสุวรรณภูมิ ดังแสดงในภาพที่ 3.3



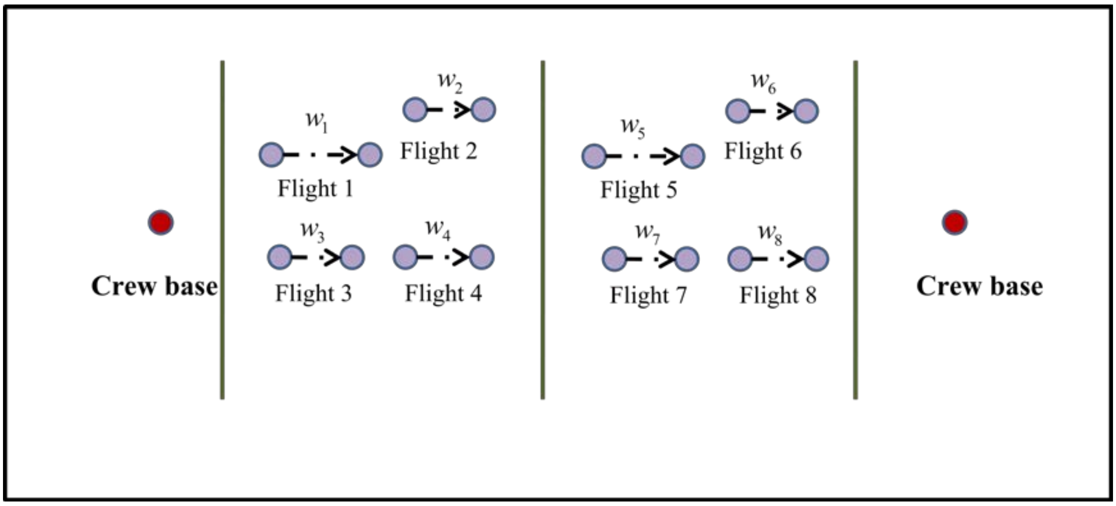
ภาพที่ 3.3 การกำหนดจุดเริ่มต้นและจุดสิ้นสุดของทัวร์

2. กำหนดจุดต่อ โดยจุดต่อแทนจุดเริ่มต้นของเวลาของเที่ยวบินและจุดสิ้นสุดของเวลาของเที่ยวบิน โดยแต่ละจุดต่อจะมีอาร์กเชื่อมต่อ ได้ภาพที่แสดงให้เห็นว่าแต่ละเที่ยวบินเริ่มออกเดินทางและสิ้นสุดการเดินทางจากเวลาใดถึงเวลาใด ดังภาพที่ 3.4



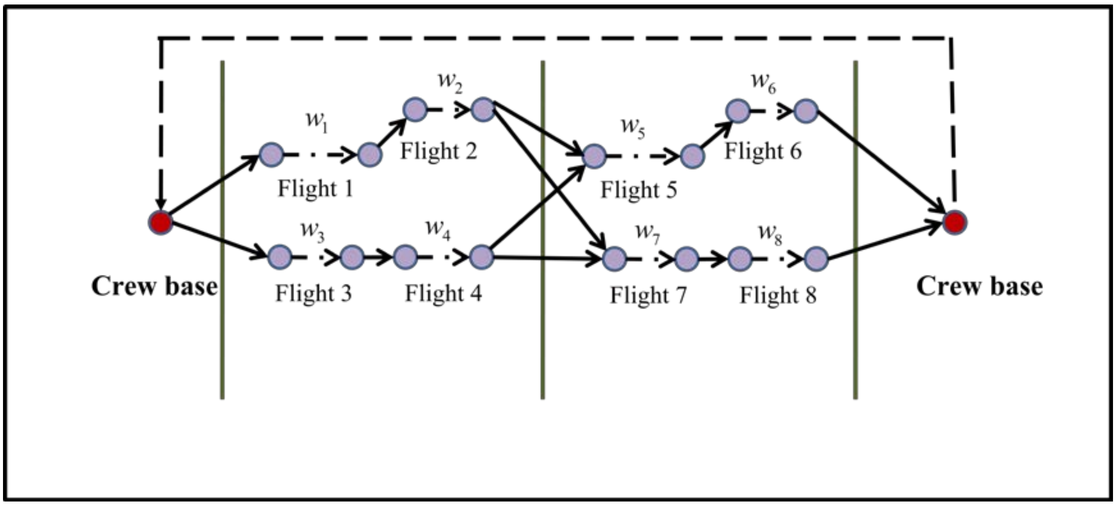
ภาพที่ 3.4 การกำหนดเที่ยวบินในโครงข่ายงาน

3. กำหนดค่าใช้จ่ายหรือตัวแปรคู่ควบซึ่งได้จากการแก้ปัญหาหลักจำกัดไปยังอาร์กที่แสดงเวลาเริ่มต้นและสิ้นสุดของแต่ละเที่ยวบิน ดังภาพที่ 3.5



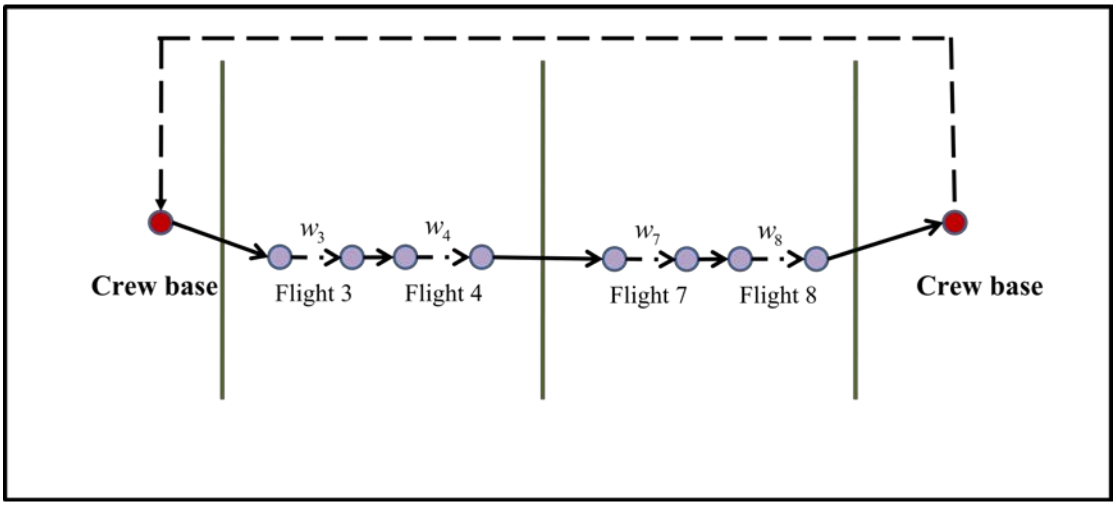
ภาพที่ 3.5 การกำหนดค่าใช้จ่ายให้แต่ละเที่ยวบิน

4. สร้างอาร์กเชื่อมต่อแต่ละจุดต่อที่เป็นไปได้ทั้งหมด โดยการเชื่อมต่อจะต้องสอดคล้องกับเงื่อนไขการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย โดยวิถีที่ได้จะแสดงทัวร์ที่เป็นไปได้ทั้งหมด ดังแสดงในภาพที่ 3.6



ภาพที่ 3.6 ทัวร์ที่เป็นไปได้

5. หาวิถีที่ให้ค่าใช้จ่ายมากที่สุด สมมติวิถีดังแสดงในภาพที่ 3.7 เป็นวิถีที่ให้ค่าใช้จ่ายมากที่สุด โดยวิถีดังกล่าวใช้ เที่ยวบินที่ 3, 4, 7 และ 8



ภาพที่ 3.7 ตัวอย่างวิถีที่ให้ค่าใช้จ่ายมากที่สุด

บทที่ 4
ผลการวิเคราะห์ข้อมูล

งานวิจัยนี้ทดสอบการประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชันสำหรับแก้ปัญหาการจับคู่เที่ยวบิน โดยใช้ข้อมูลเที่ยวบินภายในประเทศของบริษัทการบินไทยในการทดสอบ วัตถุประสงค์หลักของปัญหาคือ ต้องการจับคู่เที่ยวบินที่ยาวขึ้นในลักษณะของทัวร์เพื่อลดคำตอบที่เป็นไปได้ที่เกิดขึ้นในขั้นตอนการมอบหมายงานให้กับพนักงานสายการบิน โดยต้องการจัดทัวร์ให้ได้ทัวร์ที่มีผลรวมของช่วงเวลาพักระหว่างเที่ยวบินน้อยที่สุดและต้องเป็นทัวร์ที่สอดคล้องกับเงื่อนไขการทำงานและการพักของพนักงานสายการบินบริษัทการบินไทย โดยมีชุดข้อมูล 9 ชุด ดังรายละเอียดต่อไปนี้

4.1 รายละเอียดของชุดข้อมูล

งานวิจัยนี้ใช้ข้อมูลเที่ยวบินการบินไทย โดยเราพิจารณาเพียงเที่ยวบินภายในประเทศเท่านั้น ซึ่งจากข้อมูลของบริษัทการบินไทยมีเที่ยวบินให้บริการภายในประเทศจำนวน 80 เที่ยวบิน โดยมี 76 เที่ยวบินที่ให้บริการทุกวัน และอีก 4 เที่ยวบินเป็นเที่ยวบินที่ต้องบินต่อเนื่องกับเที่ยวบินในวันถัดไป ในส่วนของงานวิจัยนี้ต้องการศึกษาเที่ยวบินที่ให้บริการทุกวัน ดังนั้นเรามีข้อมูลเที่ยวบินสำหรับการศึกษาทั้งสิ้น 76 เที่ยวบินต่อวัน โดยแบ่งข้อมูลออกเป็น 9 ชุด ดังแสดงในตารางที่ 4.1 สำหรับใช้ในการทดสอบ

ตารางที่ 4.1 รายละเอียดของแต่ละชุดข้อมูล

ชุดข้อมูล	จำนวนวันที่ใช้ในทัวร์	จำนวนเที่ยวบินที่นำมาจัดทัวร์
1	2 วัน	8 เที่ยวบิน
2	2 วัน	20 เที่ยวบิน
3	2 วัน	40 เที่ยวบิน
4	2 วัน	60 เที่ยวบิน
5	2 วัน	80 เที่ยวบิน
6	2 วัน	100 เที่ยวบิน
7	2 วัน	120 เที่ยวบิน
8	2 วัน	140 เที่ยวบิน
9	2 วัน	152 เที่ยวบิน

4.2 การทดลอง

โปรแกรมที่ใช้ในงานวิจัยนี้เขียนขึ้นโดยใช้โปรแกรม IBM ILOG OPL CPLEX (version 12.1) รันบนเครื่องคอมพิวเตอร์ที่ใช้ Intel Core 2 Duo CPU 2.10 GHz Ram 1.92 GB โดยการหาคำตอบสำหรับงานวิจัยนี้ใช้คำตอบเริ่มต้นในลักษณะของเมทริกซ์เอกลักษณ์และกำหนดค่าช่วงเวลาพักระหว่างเที่ยวบิน (cost) ของตัวแปรเทียมเป็น 1,000,000 นาที โดยในแต่ละคอลัมน์ของเมทริกซ์แทนคอลัมน์ของตัวแปรเทียม ในส่วนผลการทดลองจะเปรียบเทียบประสิทธิภาพของเวลาที่ใช้ในการแก้ปัญหาแต่ละชุดตัวอย่างที่นำมาทดสอบ และแสดงแนวโน้มของค่าฟังก์ชันจุดประสงค์ในลักษณะของกราฟ

4.3 ผลการทดลอง

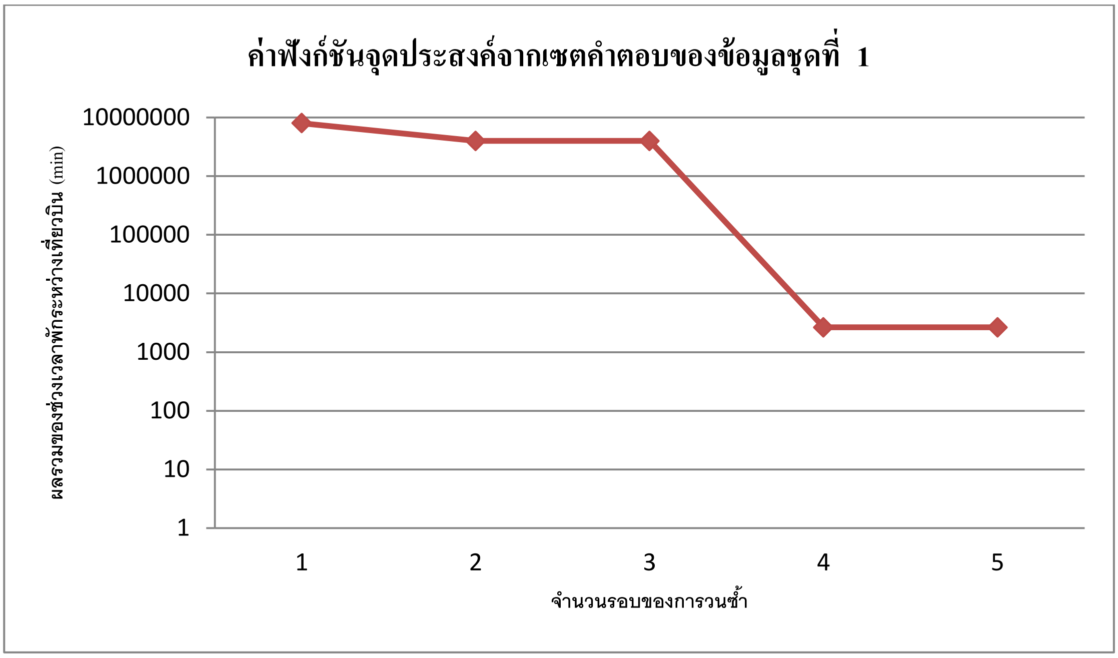
ผลลัพธ์ที่ได้จากการคำนวณด้วยเทคนิคคอลัมน์เจเนอเรชัน โดยโปรแกรม CPLEX จากข้อมูลตัวอย่าง 9 ชุดกับคำตอบเริ่มต้นซึ่งอาศัยตัวแปรเทียมในการหาคำตอบ ได้ผลดังตารางที่ 4.2

ตารางที่ 4.2 ผลการคำนวณ

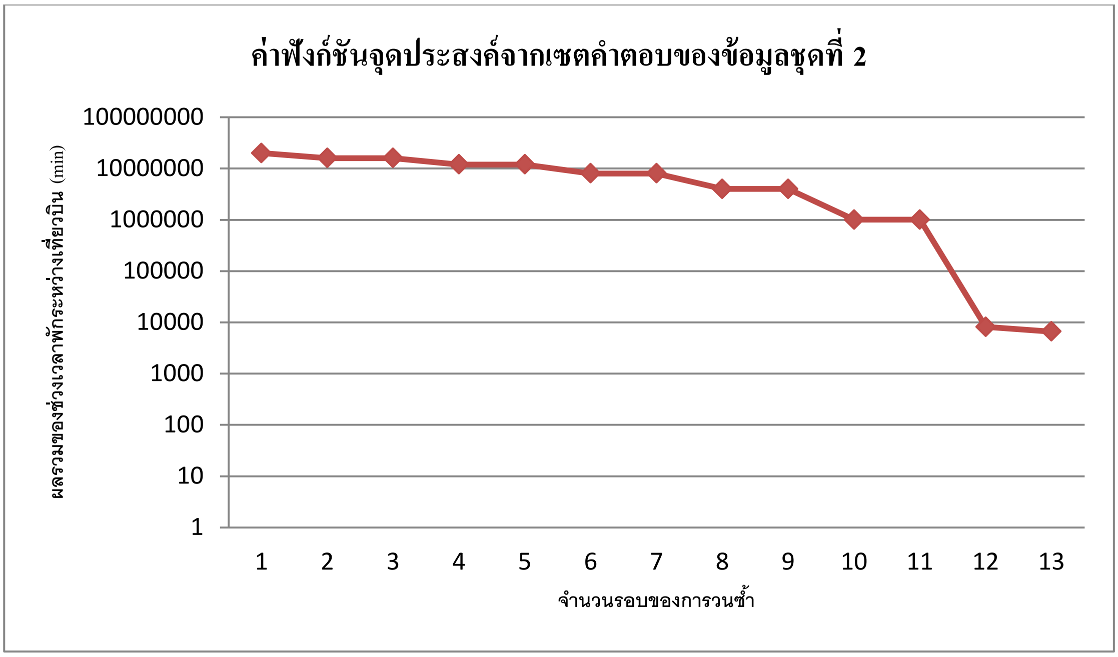
ข้อมูล	ผลรวมของช่วงเวลาพักระหว่างเที่ยวบิน	เวลาที่ใช้ในการหาคำตอบ (วินาที)	จำนวนปัญหาย่อย	จำนวนคอลัมน์ในเซตคำตอบเริ่มต้น	จำนวนคอลัมน์
1	2,650 นาที	3.37	5	8	12
2	6,625 นาที	4.13	13	20	32
3	7,415 นาที	6.18	32	40	71
4	7,880 นาที	9.43	57	60	116
5	11,780 นาที	9.09	43	80	122
6	16,935 นาที	15.74	74	100	173
7	17,535 นาที	21.35	77	120	196
8	17,170 นาที	32.54	104	140	243
9	17,405 นาที	41.13	114	152	265

จากตารางที่ 4.2 พบว่า โดยทั่วไปถ้าจำนวนเที่ยวบินมากขึ้นเวลาที่ใช้ในการคำนวณก็จะมากขึ้นด้วย แต่จำนวนเที่ยวบินมากไม่จำเป็นต้องใช้เวลาในการคำนวณผลมากกว่าเสมอไปกล่าวคือ ในข้อมูลชุดที่ 4 ใช้เที่ยวบินจำนวน 60 เที่ยวบิน ใช้เวลาในการคำนวณผลมากกว่าข้อมูลในชุดที่ 5 ซึ่งใช้เที่ยวบินมากกว่าคือ 80 เที่ยวบิน อย่างไรก็ตามจากผลการทดลองเวลาในการหาคำตอบจะขึ้นอยู่กับจำนวนครั้งของการวนซ้ำในขั้นตอนปัญหาย่อย คือ เมื่อมีการวนซ้ำของปัญหาย่อยน้อยครั้ง เวลาที่ใช้ในการคำนวณหาคำตอบจะน้อย และเมื่อมีการวนซ้ำของปัญหาย่อยมากครั้ง เวลาที่ใช้ในการคำนวณหาคำตอบก็จะมากขึ้นด้วย ซึ่งเวลาในการคำนวณคำตอบทั้ง 9 ชุด อยู่ในช่วงไม่เกิน 1 นาที ถือว่าอยู่ในวิสัยที่สามารถนำไปประยุกต์ใช้งานจริงได้ และพบว่าการเพิ่มขึ้นของค่าฟังก์ชันจุดประสงค์หรือผลรวมของช่วงเวลาพักระหว่างเที่ยวบินไม่ได้สอดคล้องตามจำนวนของเที่ยวบินที่เพิ่มขึ้นกล่าวคือ ในข้อมูลชุดที่ 7 ใช้เที่ยวบินจำนวน 120 เที่ยวบิน มีผลรวมของช่วงเวลาพักระหว่างเที่ยวบิน 17,535 นาที แต่ข้อมูลชุดที่ 8 ซึ่งใช้เที่ยวบินจำนวนมากกว่า คือ ใช้เที่ยวบินจำนวน 140 เที่ยวบิน มีผลรวมของช่วงเวลาพักระหว่างเที่ยวบิน 17,170 นาทีซึ่งน้อยกว่าข้อมูลชุดที่ 7 กรณีดังกล่าวแสดงให้เห็นว่า ถึงแม้ว่าเที่ยวบินจะมีจำนวนมากขึ้นแต่ถ้าหากมีการจับคู่เที่ยวบินที่เหมาะสมก็จะช่วยลดผลรวมของช่วงเวลาพักระหว่างเที่ยวบินได้

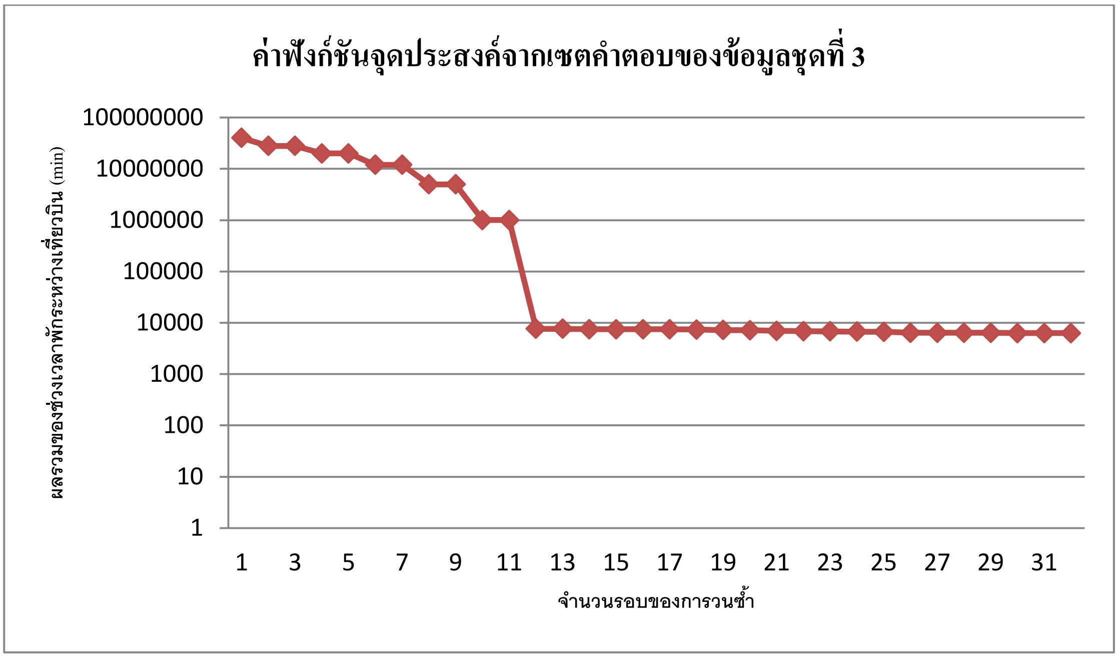
ภาพที่ 4.1-4.9 เป็นกราฟแสดงค่าฟังก์ชันจุดประสงค์หรือผลรวมของช่วงเวลาพักระหว่างเที่ยวบินที่ได้จากการแก้ปัญหาหลักที่ถูกจำกัดในแต่ละการวนซ้ำของขั้นตอนปัญหาย่อย โดยกราฟดังกล่าวใช้สเกลมาตราส่วนลอการิทึม



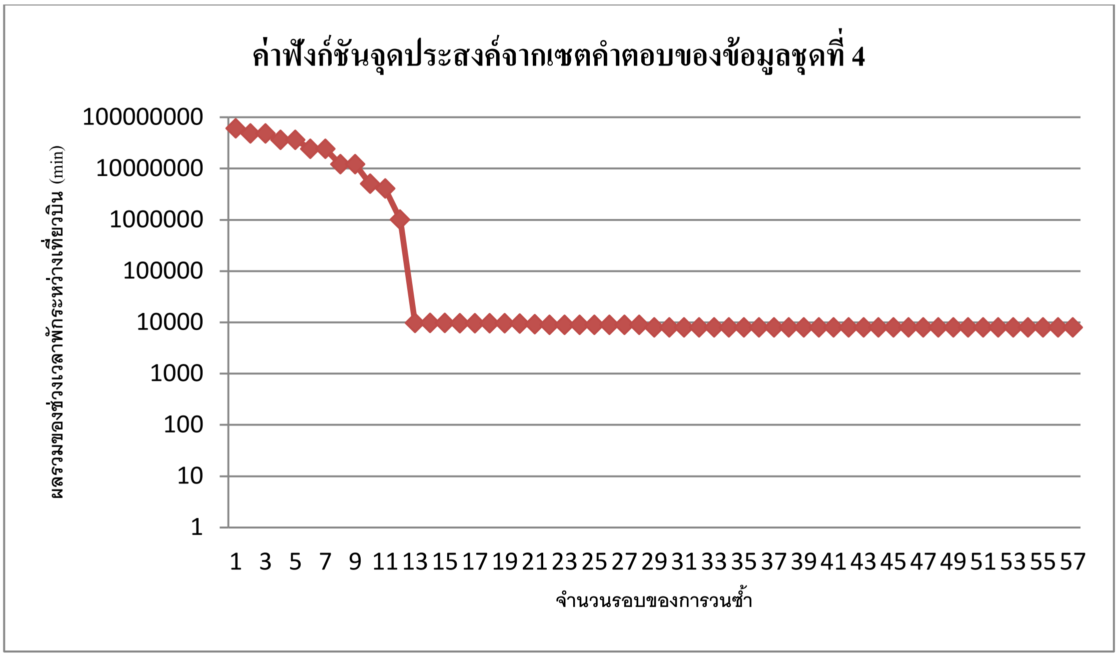
ภาพที่ 4.1 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 1



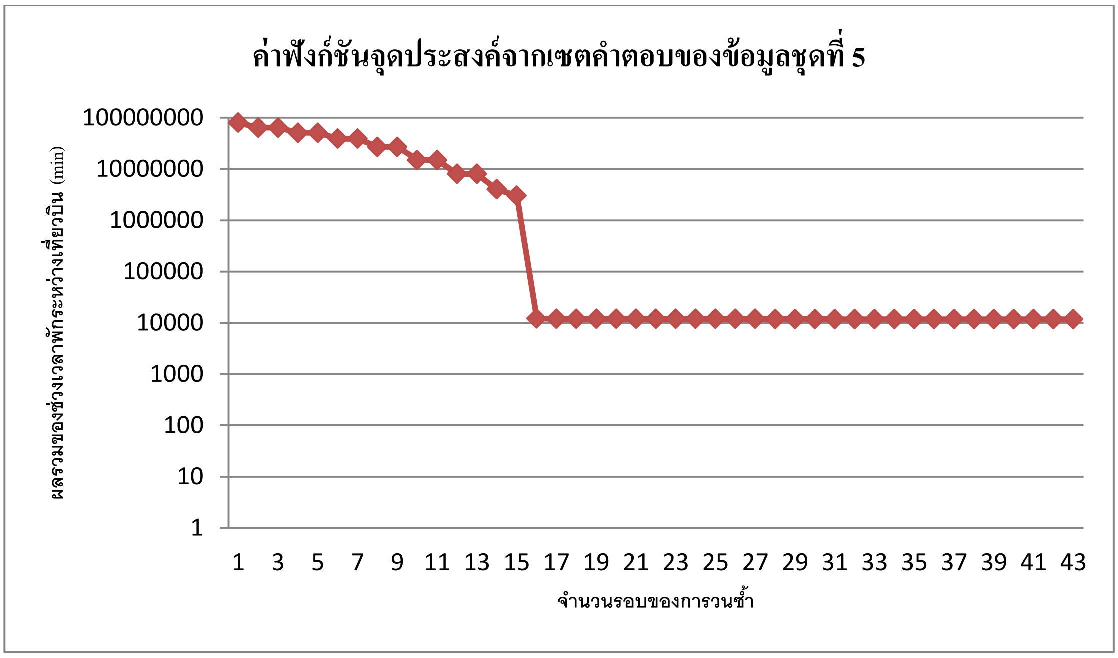
ภาพที่ 4.2 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 2



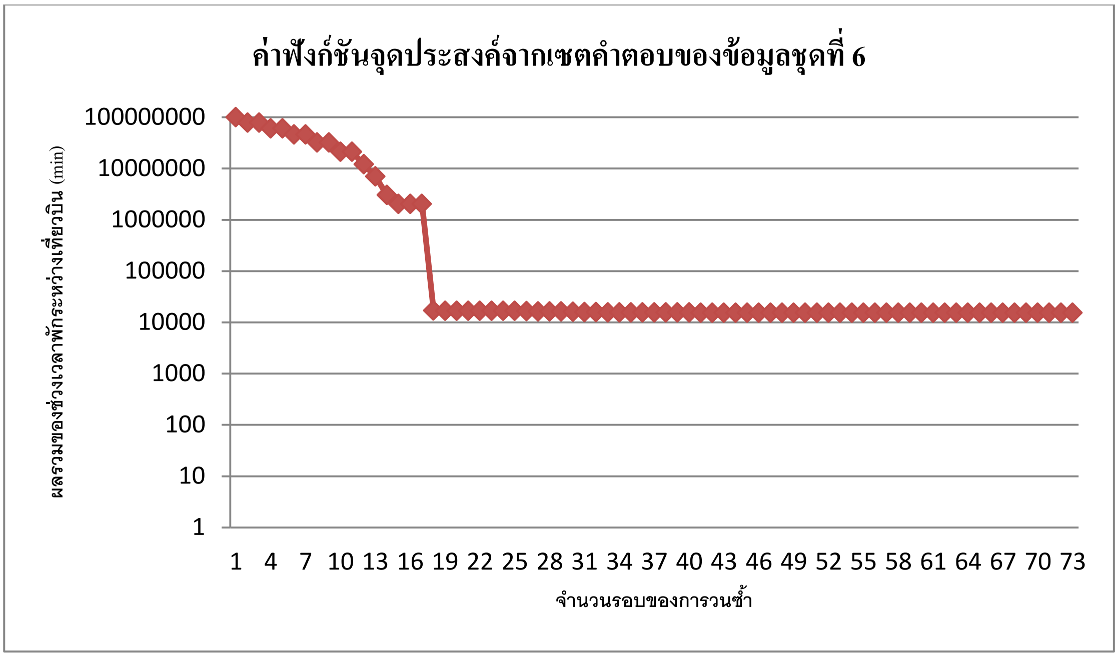
ภาพที่ 4.3 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 3



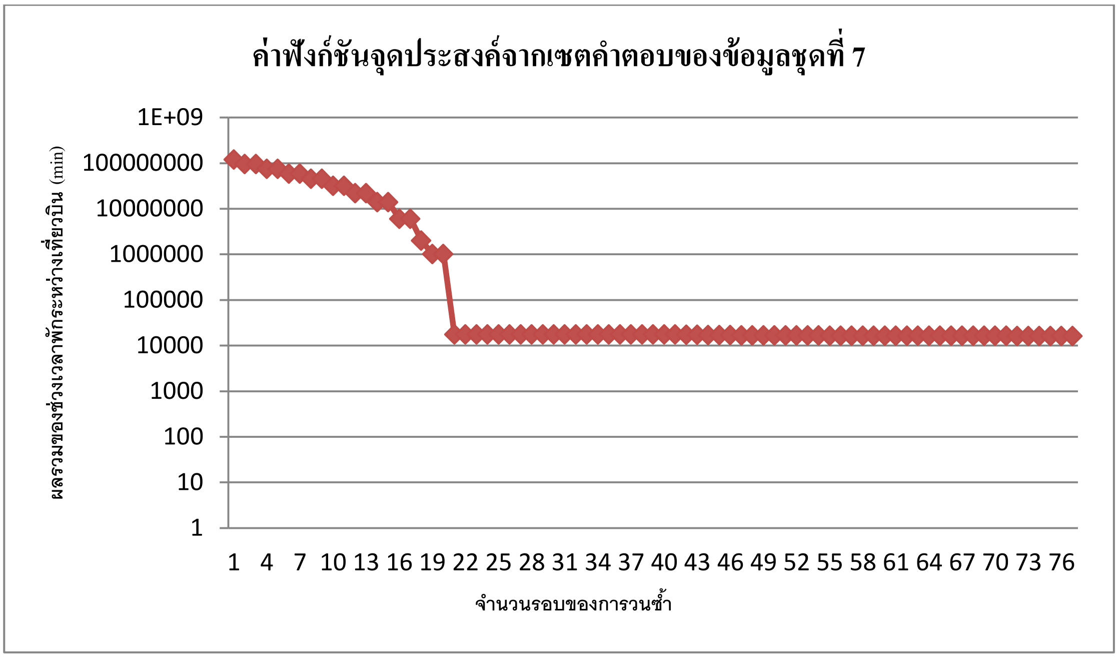
ภาพที่ 4.4 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 4



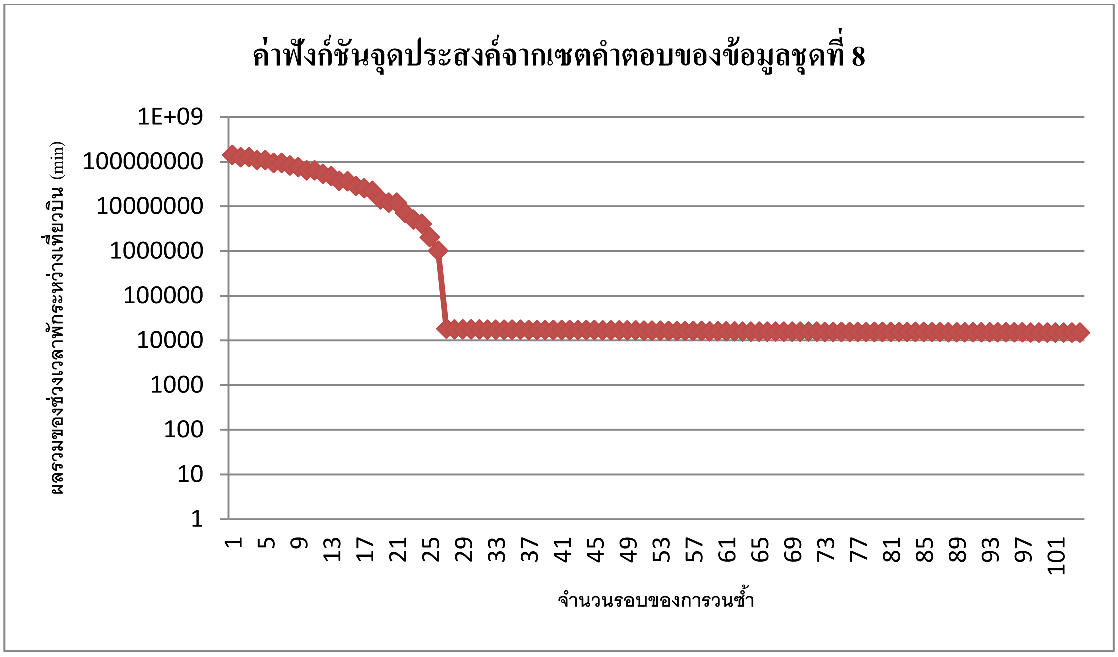
ภาพที่ 4.5 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 5



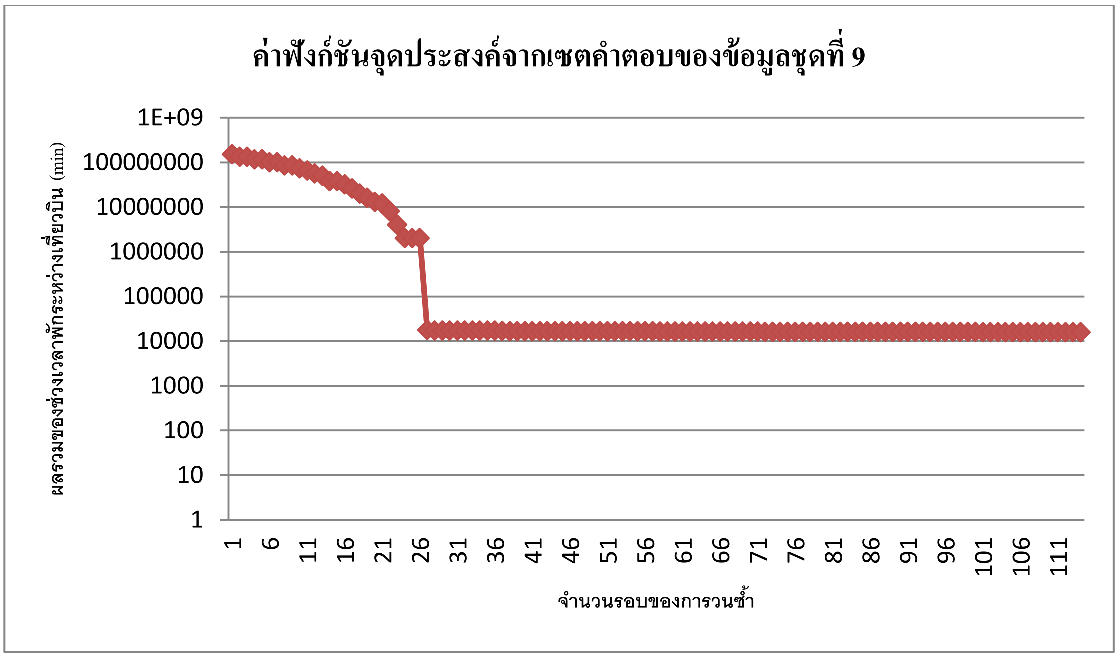
ภาพที่ 4.6 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 6



ภาพที่ 4.7 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 7



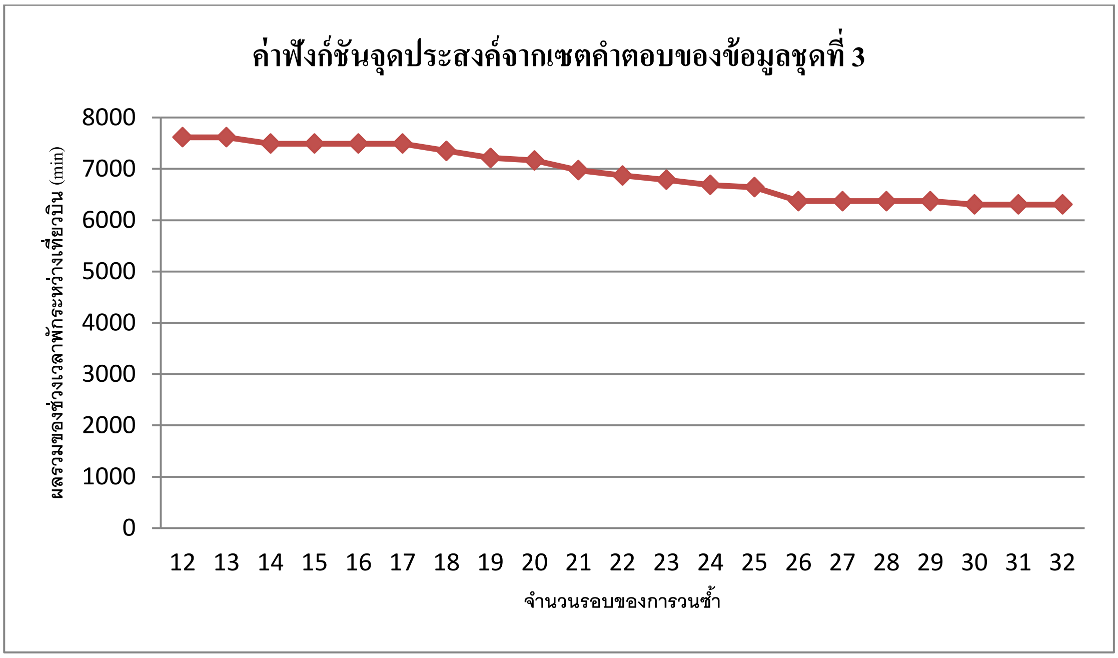
ภาพที่ 4.8 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 8



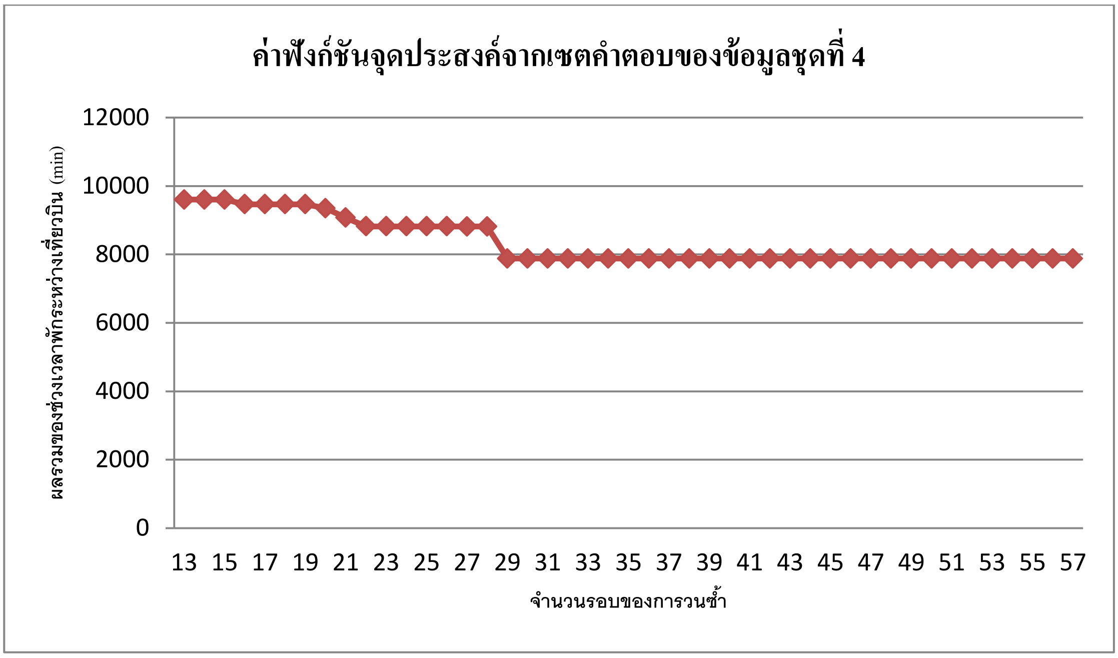
ภาพที่ 4.9 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยของข้อมูลชุดที่ 9

จากกราฟค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละปัญหาย่อยของข้อมูลทุกชุด พบว่า ในช่วงต้นค่าฟังก์ชันจุดประสงค์ของแต่ละชุดข้อมูลจะมีค่าสูงมาก เนื่องจากมีการเลือกใช้คอลัมน์ของตัวแปรเทียมเป็นตัวแปรพื้นฐานซึ่งสังเกตได้จากค่าฟังก์ชันจุดประสงค์ซึ่งมีค่าสูงกว่า 1,000,000 นาที โดยคอลัมน์ของตัวแปรเทียมจะถูกนำออกจากการวนซ้ำในแต่ละปัญหาย่อยตามกลไกขั้นตอนวิธีซึ่งจะเหนี่ยวนำให้ตัวแปรเทียมกลายเป็นตัวแปรไม่พื้นฐานหรือกลายเป็นศูนย์ ค่าฟังก์ชันจุดประสงค์จึงลดลงอย่างรวดเร็วในระยะแรก เมื่อคอลัมน์ของตัวแปรเทียมถูกนำออกจากปัญหาจนหมดซึ่งสังเกตได้จากค่าฟังก์ชันจุดประสงค์มีค่าต่ำกว่า 1,000,000 นาที ค่าฟังก์ชันจุดประสงค์ในแต่ละรอบของการวนซ้ำจะลดลงเล็กน้อยและค่อยๆคงที่ จนกระทั่งได้ค่าฟังก์ชันจุดประสงค์ที่เหมาะสำหรับปัญหาการจับคู่เที่ยวบิน

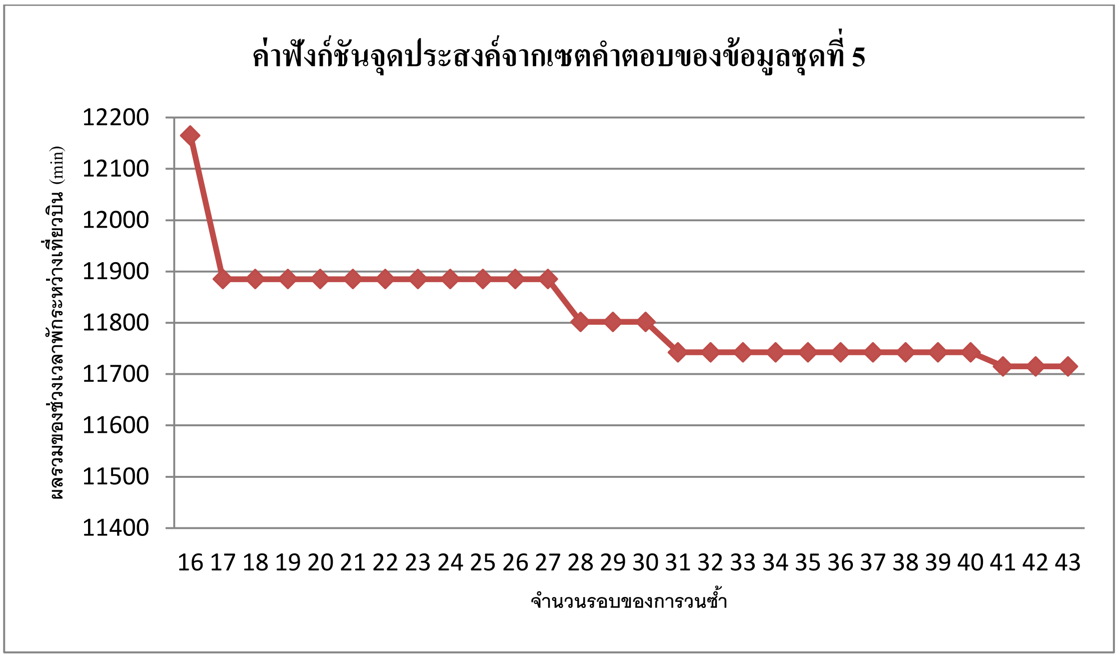
ภาพที่ 4.10-4.16 เป็นกราฟแสดงค่าฟังก์ชันจุดประสงค์หรือผลรวมของช่วงเวลาพักระหว่างเที่ยวบินที่ได้จากการแก้ปัญหาหลักที่ถูกจำกัดในแต่ละการวนซ้ำของขั้นตอนปัญหาย่อยหลังจากที่ตัวแปรเทียมถูกนำออกหมดของข้อมูลชุดที่ 3-9



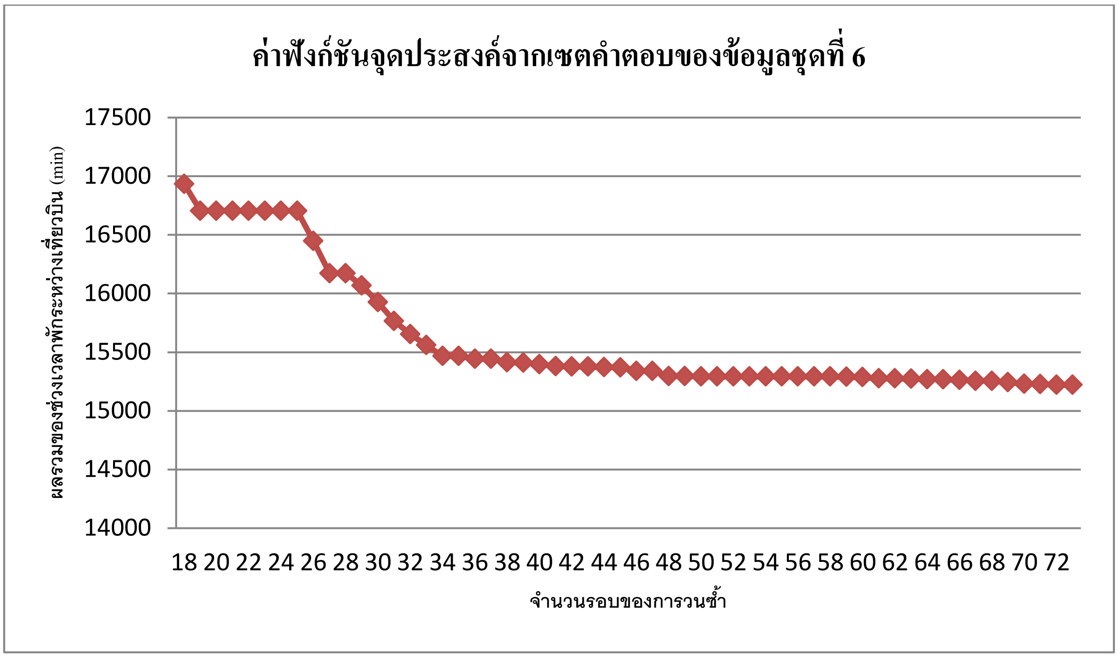
ภาพที่ 4.10 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยหลังจากที่ตัวแปรเทียมถูกนำออกหมดของข้อมูลชุดที่ 3



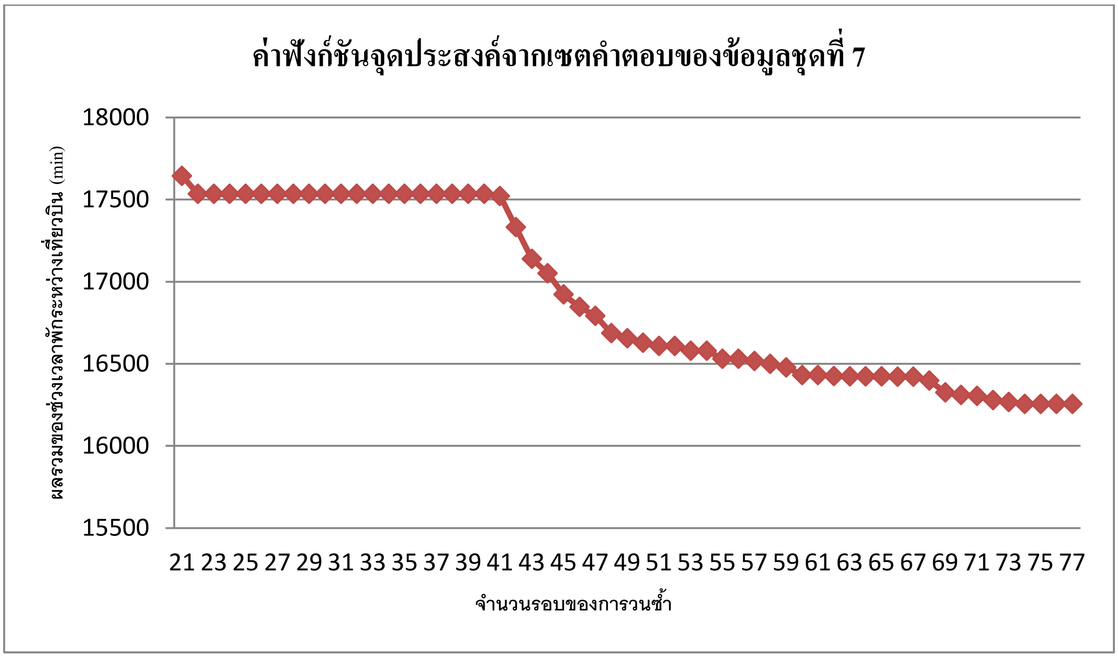
ภาพที่ 4.11 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยหลังจากที่ตัวแปรเทียมถูกนำออกหมดของข้อมูลชุดที่ 4



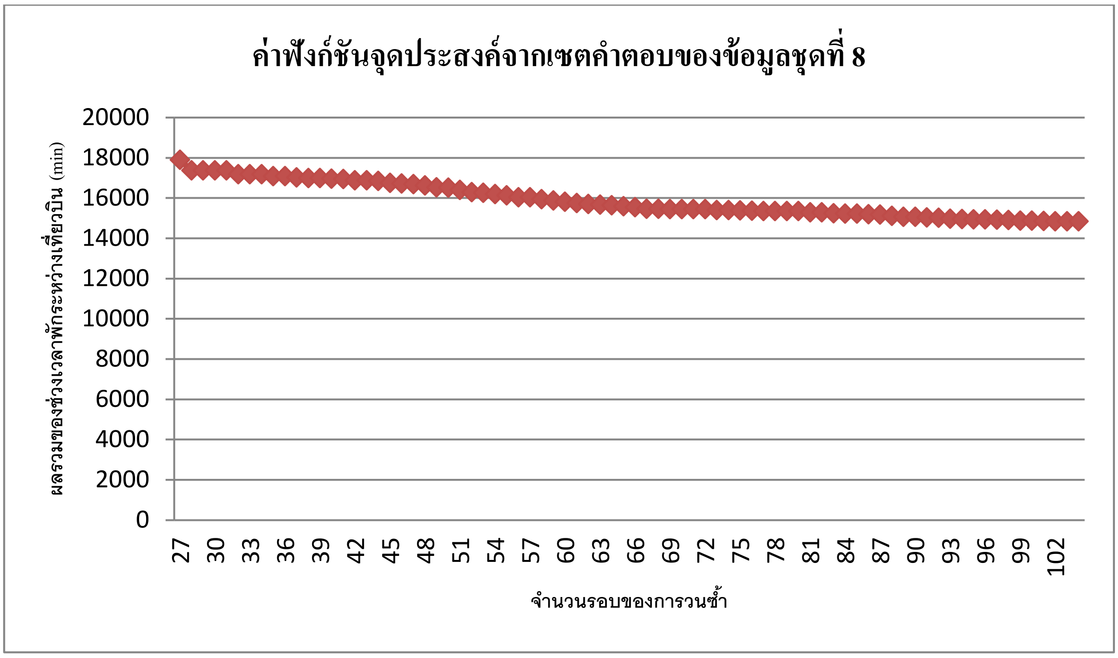
ภาพที่ 4.12 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยหลังจากที่ตัวแปรเทียมถูกนำออกหมดของข้อมูลชุดที่ 5



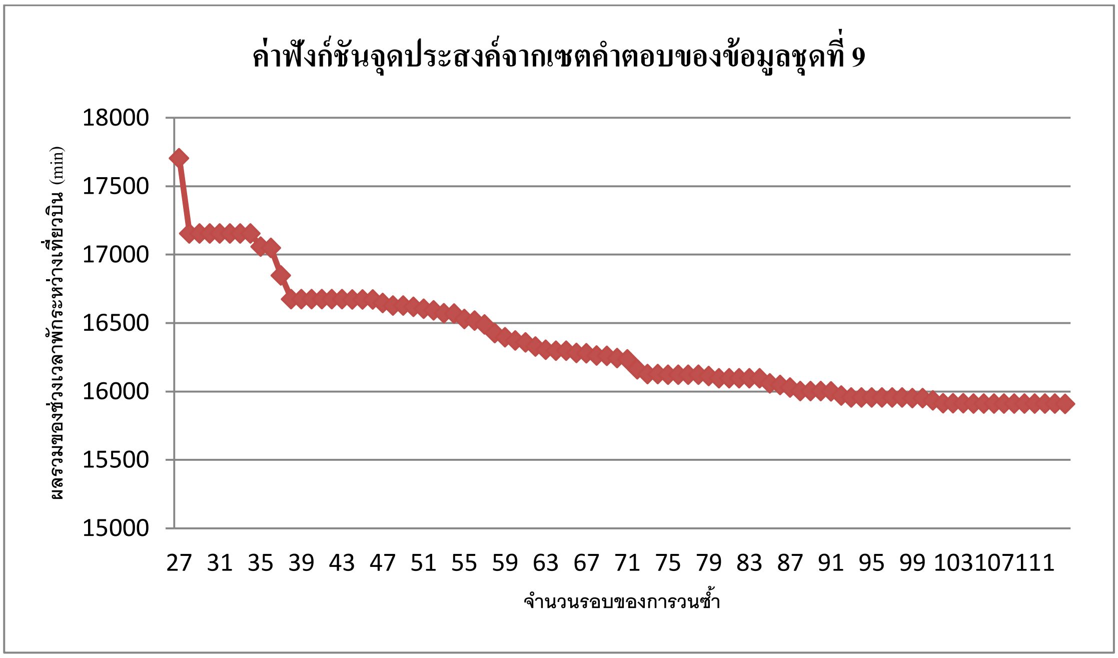
ภาพที่ 4.13 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยหลังจากที่ตัวแปรเทียมถูกนำออกหมดของข้อมูลชุดที่ 6



ภาพที่ 4.14 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยหลังจากที่ตัวแปรเทียม ถูกนำออกหมดของข้อมูลชุดที่ 7



ภาพที่ 4.15 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยหลังจากที่ตัวแปรเทียม ถูกนำออกหมดของข้อมูลชุดที่ 8



ภาพที่ 4.16 ค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละการวนซ้ำปัญหาย่อยหลังจากที่ตัวแปรเทียมถูกนำออกหมดของข้อมูลชุดที่ 9

กราฟค่าฟังก์ชันจุดประสงค์ที่ได้ในแต่ละปัญหาย่อยหลังจากที่ตัวแปรเทียมถูกนำออกหมดหรือกราฟค่าฟังก์ชันจุดประสงค์หลังจากที่ค่าฟังก์ชันจุดประสงค์มีค่าต่ำกว่า 1,000,000 นาที ของข้อมูลทุกชุด พบว่า ค่าฟังก์ชันจุดประสงค์ของแต่ละชุดข้อมูลจะลดลงในแต่ละรอบของการวนซ้ำเพียงเล็กน้อยและคงที่ในบางช่วง จนกระทั่งได้ค่าฟังก์ชันจุดประสงค์ที่เหมาะสำหรับปัญหาการจับคู่เที่ยวบินตามเงื่อนไขที่เราอธิบายไปแล้วในหน้า 20 และจากการพิจารณาแต่ละคอลัมน์ที่ถูกนำเข้าสู่ปัญหาหลักที่ถูกจำกัด พบว่า บางคอลัมน์ที่ถูกนำเข้าสู่ปัญหาหลักที่ถูกจำกัดจะถูกนำออกในขั้นตอนปัญหาย่อยในรอบของการวนซ้ำถัดไป

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยนี้นำเสนอเทคนิคคอลัมน์เจเนอเรชันสำหรับปัญหาการจับคู่เที่ยวบิน โดยมีวัตถุประสงค์หลัก คือ ต้องการจับคู่เที่ยวบินในรูปแบบของทัวร์ระยะเวลา 2 วัน ให้ได้ทัวร์ซึ่งมีผลรวมของช่วงเวลาพักระหว่างเที่ยวบินน้อยที่สุด และทำการทดสอบกับข้อมูลเที่ยวบินภายในประเทศของบริษัทการบินไทย โดยพิจารณาเฉพาะเที่ยวบินที่ให้บริการทุกวัน จำนวนทั้งสิ้น 76 เที่ยวบิน โดยแบ่งข้อมูลออกเป็น 9 ชุด ดังนี้ ชุดที่ 1 มี 8 เที่ยวบิน ชุดที่ 2 มี 20 เที่ยวบิน ชุดที่ 3 มี 40 เที่ยวบิน ชุดที่ 4 มี 60 เที่ยวบิน ชุดที่ 5 มี 80 เที่ยวบิน ชุดที่ 6 มี 100 เที่ยวบิน ชุดที่ 7 มี 120 เที่ยวบิน ชุดที่ 8 มี 140 เที่ยวบิน และชุดที่ 9 มี 152 เที่ยวบิน

การหาคำตอบเริ่มต้นสำหรับปัญหาแต่ละขนาด เนื่องจากการสุ่มเที่ยวบินเพื่อมาสร้างทัวร์คำตอบเริ่มต้นในบางครั้งจะได้เมทริกซ์เอกฐาน มีผลทำให้ไม่สามารถหาค่าตัวแปรคู่ควบที่จะนำมาใช้ในปัญหาย่อยได้ คำตอบในลักษณะดังกล่าวจึงไม่สามารถใช้เป็นคำตอบเริ่มต้นของปัญหาได้ ในงานวิจัยนี้จึงอาศัยตัวแปรเทียมในลักษณะของเมทริกซ์เอกลักษณ์เพื่อช่วยในการหาคำตอบที่เป็นไปได้

นอกจากนี้ปัญหาย่อยที่ใช้ไม่ใช่ปัญหา $\underset{j\in T}{\text{Min}}(c_j - \mathbf{w}^T\mathbf{a}_j)$ โดยตรงแต่เป็นปัญหา $\underset{j\in T}{\text{Min}}(-\mathbf{w}^T\mathbf{a}_j)$ เนื่องจากค่า c_j ของทัวร์ที่ j ไม่สามารถทราบค่าได้จนกว่าจะรู้แบบรูปของทัวร์ ค่าที่ได้จากปัญหาดังกล่าวจึงไม่รับประกันว่าเป็นค่าที่มี reduced cost ต่ำที่สุด ซึ่งเป็นผลให้คำตอบของปัญหาหลักที่ได้จากเทคนิคคอลัมน์เจเนอเรชันที่ใช้ปัญหาย่อยดังกล่าว อาจจะไม่ใช่คำตอบที่เหมาะที่สุด สำหรับงานวิจัยนี้คำตอบที่ได้จะถือว่าเป็นคำตอบที่ “เหมาะ” เท่านั้น

การหาคำตอบของปัญหาในงานวิจัยนี้เขียนขึ้นโดยใช้โปรแกรม IBM ILOG OPL CPLEX (version 12.1) รันบนเครื่องคอมพิวเตอร์ที่ใช้ Intel Core 2 Duo CPU 2.10 GHz Ram 1.92 GB โดยคำตอบเริ่มต้นสำหรับงานวิจัยนี้ใช้คำตอบเริ่มต้นในลักษณะของเมทริกซ์เอกลักษณ์ โดยในแต่ละคอลัมน์ของเมทริกซ์แทนคอลัมน์ของตัวแปรเทียม ผลการทดลองทางด้านประสิทธิภาพของเวลาที่ใช้ในการแก้ปัญหา พบว่า จำนวนเที่ยวบินมากไม่จำเป็นต้องใช้เวลาในการคำนวณผลมากกว่าเสมอไป โดยทั่วไปถ้าจำนวนเที่ยวบินมากขึ้นเวลาที่ใช้ในการคำนวณก็จะมากขึ้นด้วย อย่างไรก็ตามจากผลทดลองเวลาในการหาคำตอบจะขึ้นอยู่กับจำนวนครั้งของการวนซ้ำในขั้นตอนปัญหาย่อย คือ เมื่อ

มีการวนซ้ำของปัญหาย่อยน้อยครั้ง เวลาที่ใช้ในการคำนวณหาคำตอบจะน้อย และเมื่อมีการวนซ้ำของปัญหาย่อยมากครั้ง เวลาที่ใช้ในการคำนวณหาคำตอบก็จะมากขึ้นด้วย ซึ่งเวลาในการคำนวณคำตอบทั้ง 9 ชุด อยู่ในช่วงไม่เกิน 1 นาที ถือว่าอยู่ในวิสัยที่สามารถนำไปประยุกต์ใช้งานจริงได้ สำหรับแนวโน้มของค่าฟังก์ชันจุดประสงค์หรือแนวโน้มของผลรวมของช่วงเวลาพักระหว่างเที่ยวบินเมื่อเทียบกับจำนวนรอบของการวนซ้ำของข้อมูลทั้ง 9 ชุด เป็นไปในลักษณะเดียวกัน คือ ในช่วงต้น ค่าฟังก์ชันจุดประสงค์ของแต่ละชุดข้อมูลจะมีค่าสูงมาก เนื่องจากมีการเลือกใช้คอลัมน์ของตัวแปรเทียม โดยคอลัมน์ของตัวแปรเทียมจะถูกนำออกจากการวนซ้ำในแต่ละปัญหาย่อย ค่าฟังก์ชันจุดประสงค์จึงลดลงอย่างรวดเร็วในระยะแรก เมื่อคอลัมน์ของตัวแปรเทียมถูกนำออกจากปัญหาจนหมด ค่าฟังก์ชันจุดประสงค์ค่าลดลงเล็กน้อยและค่อยๆคงที่ จนกระทั่งได้ค่าฟังก์ชันจุดประสงค์ที่เหมาะสำหรับปัญหาการจับคู่เที่ยวบิน

การแก้ปัญหาการจับคู่เที่ยวบินโดยประยุกต์ใช้เทคนิคคอลัมน์เจเนอเรชัน ช่วยแก้ปัญหาทางด้านประสิทธิภาพในเรื่องเวลาคำนวณหาคำตอบ คือ ทำให้การคำนวณหาคำตอบเร็วขึ้น หากเกิดเหตุการณ์เฉพาะหน้าอันเป็นเหตุทำให้ต้องมีการปรับเปลี่ยนการจับคู่เที่ยวบิน การใช้เทคนิคคอลัมน์เจเนอเรชันจะช่วยให้สามารถแก้สถานการณ์ได้เร็ว สำหรับเทคนิคคอลัมน์เจเนอเรชันที่ได้นำเสนอในวิทยานิพนธ์นี้ สามารถนำไปพัฒนาเพิ่มเติมในส่วนต่างๆได้ เช่น การปรับเปลี่ยนเทคนิคการสร้างปัญหาย่อย หรือการปรับเปลี่ยนโปรแกรมที่ใช้ในการคำนวณหาคำตอบ เพื่อให้ประสิทธิภาพในการหาคำตอบดียิ่งขึ้นได้อีกด้วย

รายการอ้างอิง

[1] Guy Desaulniers, Jacques Desrosiers and Marius M. Solomon. Column Generation : 8-9

[2] Goran Stojkovic, François Soumis, Jacques Desrosiers and Marius M. Solomon. 2002. An optimization model for a real-time flight scheduling problem. Transportation Research Part A: Policy and Practice Vol.36: 779-788.

[3] Shervin Ahmad Beygi, Amy Cohn and Marshall Weir. 2009. An integer programming approach to generating airline crew pairings. Computers & Operations Research Vol.36: 1284-1298.

[4] Alberto Caprara, Paolo Toth, Daniele Vigo and Matteo Fischetti. 1998. Modeling and Solving the Crew Rostering Problem, Operations Research Vol.46: 820-830.

[5] Yufeng Guo, Taieb Mellouli, Leena Suhl and Markus P. Thiel. 2006. A partially integrated airline crew scheduling approach with time-dependent crew capacities and multiple home bases. European Journal of Operational Research Vol.171: 1169-1181.

[6] Sylvie Lavoie, Michel Minoux and Edouard Odier. 1988. A new approach for crew pairing problems by column generation to air transportation. European Journal of Operational Research Vol.35: 45-58.

[7] Ahmadbeygi, S. Cohn and A. M. .2006. Generating crew pairings for very large flight networks. Technical report TR06-03. Industrial and Operations Engineering Department, University of Michigan, Ann Arbor.

[8] Anbil, R., Forrest, J. and Pulleyblank, W. 1998. Column generation and the airline crew pairing problem. Documenta Mathematica Vol. ICM(III): 677–686.

[9] Yan S and Tu Y. 2001. A network model for airline cabin crew scheduling. European Journal of Operational Research Vol.140: 531–40.

[10] Barnhart C, Hatay L and Johnson E. 1995. Deadhead selection for long-haul crew pairing problem. Operations Research Vol.43: 491–9.

[11] Alberto Caprara, Paolo Toth, Daniele Vigo and Matteo Fischetti. 1998. Modeling and Solving the Crew Rostering Problem. Operations Research Vol.46: 820-830.

[12] Michel Gamache, François Soumis, Gérald Marquis and Jacques Desrosiers. 1999. A Column Generation Approach for Large-Scale Aircrew Rostering Problems. Operations Research Vol.47: 247-263.

[13] Paola Cappanera and Giorgio Gallo. 2004. A Multicommodity Flow Approach to the Crew Rostering Problem. Operations Research Vol.52: 583-596.

[14] Chawalit Jeenanunta, Boonyarit Intiyot and Wariya Puttapatimok. 2010. A Multi-commodity Flow Approach to the Crew Rostering Problem. The 2nd International Conference on Logistics and Transport, Queenstown, New Zealand, December 16-18 2010.

[15] B. Maebhout and Vanhoucke. 2010. A Hybrid Scatter Search Heuristic for Personalized Crew Rostering in the Airline Industry. European Journal of Operational Research Vol.206: 155-167.

[16] P. Lucic and D. Teodorovic. 2007. Metaheuristics approach to the aircrew rostering problem. Annals of Operations Research. Vol.155: 311-338.

[17] Michael J. Brusco and Larry W. Jacobs. 1993. A Simulated Annealing Approach to the Solution of Flexible Labour Scheduling Problems. The Journal of the Operational Research Society Vol.44: 1191-1200.

[18] V. Limlawan, B. Kasemsontitum and C. Jeenanunta. 2011. Airline Crew Rostering Problem Using Particle Swarm Optimization. The 2011 IEEE International Conference on Quality and Reliability, Bangkok, Thailand, Sep. 14-17 2011.

[19] K. Onsuan, B. Intiyot, and C. Jeenanunta. A Workload-Balance Crossover Operation in a Genetic Algorithm for solving an Airline Crew Rostering Problem. IE Network Journal 2011: 35-41.

ภาคผนวก

Source code สำหรับ IBM ILOG CPLEX Optimization

ในส่วนนี้จะแสดง Source code สำหรับ IBM ILOG CPLEX Optimization ที่ใช้ในวิทยานิพนธ์นี้ ซึ่งประกอบไปด้วยรายละเอียดต่างๆ ดังนี้

1. Main.mod

```
//initial value
 int NumFlights = ...;
 range Flights = 1..NumFlights;
 {float} AllObj=...;
 int Pattern[Flights];
 int SumSittime=0;
 float MaxDualFlight ;
 int MaxDualNumFlight;
float DualsFlights[Flights] = ...;

// tuple Variable
tuple arc {
      key int Fromnode;
      key int Tonode;
      string Arctype;
      int Flightnumber;
      int    Depday;
      int    Arrivalday;
      int   Deqtime;
      int   Arrivaltime;
      float Blocktime;
      float Resttime;
      float Sittime;
      string Origin;
      string Destination;
      }

tuple Tour
 {
    int TourNum;
      int TourPatterns[Flights];
      float Sittime;
 }


// Create Arcs list (Arc type) and read from excel
 {arc} Arcs = ...;

 float DualsAll[Arcs];

// Create Tours (Tour type) and read from Data.dat (initial solution)
 {Tour} FlightinTour = ...;
```

```
//Define Master Problem

//Decision Variable ( SelectTour  if = 1 -> selected : if = 0 -> NO
selected )

 dvar int+ GenTour[FlightinTour] in 0..1;

//Objective Function

  minimize  sum(t in FlightinTour)t.Sittime*GenTour[t];

  subject to {

// Flights Constraints

 forall (f in Flights )
      FlightsConstraints:
      sum(t in FlightinTour)t.TourPatterns[f]*GenTour[t]>=1;
 }
 tuple s {
   Tour t;
   int GenTour;
};

{s} Result = {<t,GenTour[t]>|t in FlightinTour:GenTour[t]!=0};

 execute RESULT
 {
      write(Result);
 }

//set dual values used to fill in the sub model

 execute FillDuals
 {  MaxDualFlight = -Infinity;
      MaxDualNumFlight = 0;

   for (var f in Flights)
      {
       DualsFlights[f] = FlightsConstraints[f].dual;
    }

    for (var k = 0 ; k < NumFlights ; k++)
   {
       DualsAll[Arcs.get(5+(2*k+1),5+(2*k+2))] = DualsFlights[k+1]

   }

main {

   var status = 0;

   // Master Model
   thisOplModel.generate();
   var masterDef = thisOplModel.modelDefinition;
   var masterCplex = cplex;
   var masterData = thisOplModel.dataElements;
   // Creating the master-model
```

```
var masterOpl = new IloOplModel(masterDef, masterCplex);
masterOpl.addDataSource(masterData);
masterOpl.generate();
write();

// Preparing sub-model source, definition and engine
var subSource = new IloOplModelSource("Subproblem.mod");
var subDef = new IloOplModelDefinition(subSource);
var subCplex = new IloCplex();

var best;
var curr = Infinity;
var ObjofInt=Infinity;
var count = 0;

for (var i = 1 ; i <= 120 ;i++)
{
   best = curr;
   masterOpl = new IloOplModel(masterDef,masterCplex);
   masterOpl.addDataSource(masterData);
   masterOpl.generate();
   masterOpl.convertAllIntVars();

   writeln("Solve master.");
   if ( masterCplex.solve() )
   {
         masterOpl.postProcess();
     curr = masterCplex.getObjValue();
     writeln();
     writeln("MASTER OBJECTIVE: ",curr);
     masterData.AllObj.add(curr);

   } else {
      writeln("No solution to master problem!");
      masterOpl.end();
      break;
   }

   // Ceating the sub model
   var subOpl = new IloOplModel(subDef,subCplex);

    //Using data elements from the master model.
   var subData = new IloOplDataElements();
   subData.Arcs = masterOpl.Arcs;
   subData.DualsAll = masterOpl.DualsAll;
   subData.NumFlights = masterOpl.NumFlights;
   subOpl.addDataSource(subData);
   subOpl.generate();


   // Previous master model is not needed anymore.
   writeln("Solve sub.");
   if ( subCplex.solve()){
        count++;
        writeln("iteration",count);
        writeln();
        writeln("SUB OBJECTIVE: ",subCplex.getObjValue());
             subOpl.postProcess();
```

```
                 writeln("Sum of Sittime",subOpl.SumSittime);

                 var CJ = (subOpl.SumSittime);
            writeln("reduce cost of path",CJ-
(subCplex.getObjValue()));
             if (CJ-(subCplex.getObjValue())>= 0)
             {

                   writeln("The current solution is optimal")
                   subData.end();
                   subOpl.end();
                   break;
             }

         } else {
         writeln("Don't new Tour, stop.");
         subData.end();
         subOpl.end();
         break;
       }
       // prepare next iteration
       writeln("new pattern tour",subOpl.pat);

masterData.FlightinTour.add(masterData.FlightinTour.size+1,subOpl.pat
,subOpl.SumSittime);
       masterOpl.end();

      // End sub model
       subData.end();
       subOpl.end();
       writeln("best ",best);
       writeln("curr ",curr);
    }

  //Check solution value
    masterOpl = new IloOplModel(masterDef,masterCplex);
    masterOpl.addDataSource(masterData);
    masterOpl.generate();
    writeln("Solve integer master.");

    if (masterCplex.solve()){
    writeln();
    writeln("OBJECTIVE:",masterCplex.getObjValue());
    masterOpl.postProcess().RESULT;
    masterOpl.printSolution();
    writeln("All objective solution is ");
    writeln(masterOpl.AllObj);
    writeln("iteration",count);

    }
    else {write("no solution")}

    subDef.end();
    subCplex.end();
    subSource.end();
//   status;
}
```

2. Sub-problem.mod

```
//initial value
 int NumFlights = ...;
 range Flights = 1..NumFlights;

 int Pattern[Flights];
 int SumSittime=0;


// tuple Variable
tuple arc {
      key int Fromnode;
      key int Tonode;
      string Arctype;
      int Flightnumber;
      int    Depday;
      int    Arrivalday;
      int   Deptime;
      int   Arrivaltime;
      float Blocktime;
      float Resttime;
      float Sittime;
      string Origin;
      string Destination;
      }

{arc} Arcs = ...;

float DualsAll[Arcs]=...;

int pat[Flights];

tuple Tour
 {
    int   TourNum;
      int TourPatterns[Flights];
      float Sittime;
 }


{int} Days = {a.Depday | a  in Arcs} union {a.Arrivalday| a in Arcs};
{int} Nodes = {a.Fromnode | a  in Arcs} union  {a.Tonode | a  in
Arcs};


dvar int Flow[Arcs]in 0..1;

maximize sum(<i,j,at,fnum,dday,aday,dtime,atime,bt,rt,st,ori,des> in
Arcs) Flow[<i,j>] * DualsAll[<i,j>];

 subject to{

    // Flow consevation constraints
    // For source node

    ctflowconservationsource:
    forall (i in Nodes:i==5)
```

```
            sum(<i,j,at,fnum,dday,aday,dtime,atime,bt,rt,st,ori,des>
in Arcs) Flow[<i,j>] == 1;

      // For sink node
      // 1.i==22 if we use Data 08F2D
      // 2.i==46 if we use Data 20F2D
      // 3.i==86 if we use Data 40F2D
      // 4.i==126 if we use Data 60F2D
      // 5.i==166 if we use Data 80F2D
      // 6.i==206 if we use Data 100F2D
      // 7.i==246 if we use Data 120F2D
      // 8.i==286 if we use Data 140F2D
      // 9.i==310 if we use Data 152F2D

      ctflowconservationsink:
      forall (i in Nodes:i ==22)
      -
sum(<k,i,at,fnum2,dday2,aday2,dtime2,atime2,bt2,rt2,st2,ori2,des2>in
Arcs)Flow[<k,i>] == -1;

    // For any nodes
    ctflowconservationanynode:
    forall (i in Nodes: i !=5 && i!=22)
    sum(<i,j,at,fnum,dday,aday,dtime,atime,bt,rt,st,ori,des> in Arcs)
Flow[<i,j>]
      -
sum(<k,i,at,fnum2,dday2,aday2,dtime2,atime2,bt2,rt2,st2,ori2,des2>in
Arcs)Flow[<k,i>] == 0;

    forall (d in Days : d-6>=1)
       ctHourDaily: sum (a in Arcs, i in d-7..d: a.Depday == i )
a.Blocktime*Flow[a] <= 2040;
}

execute test
 {
      for (var a in Arcs){
      SumSittime = SumSittime + Flow[a]*a.Sittime;

      if (a.Arctype == "flight arc")
      {
            pat[a.Flightnumber] = Flow[a];
      }

      }
 }
```

ประวัติผู้เขียนวิทยานิพนธ์

ชื่อ	สมิหลา คีรีศรี
วัน เดือน ปีที่เกิด	10 มกราคม 2531
สถานที่เกิด	จังหวัดสงขลา ประเทศไทย
ประวัติการศึกษา	ปริญญาตรี วิทยาศาสตรบัณฑิต (สาขาคณิตศาสตร์) จุฬาลงกรณ์มหาวิทยาลัย พ.ศ.2552

ผลงานตีพิมพ์

- สมิหลา คีรีศรี, บุญฤทธิ์ อินทิยศ และ ชวลิต จีนอนันต์. 2012. เทคนิคคอลัมน์เจเนอเรชันสำหรับปัญหาการจับคู่เที่ยวบิน. IE Network Conference 2012: 23-29.
- Samila Kirisri. 2012. Column generation technique for crew pairing problem. The 8th Mathematics and Physical Science Graduate Congress 2012: 118.