

โปรแกรมเพื่อแก้ปัญหาการมอบหมายงานหลายรูปแบบ
ด้วยภาษาไพทอน

A PROGRAM FOR SOLVING VARIOUS ASSIGNMENT
PROBLEMS USING PYTHON

นางสาวรัชฎา พิสิทธิ์

นางสาวลีปาง สายสุข

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (คณิตศาสตร์ประยุกต์)
ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

A PROGRAM FOR SOLVING VARIOUS ASSIGNMENT
PROBLEMS USING PYTHON

MS. RATCHADA PHILUK

MS. SIPANG SAYSUK

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (APPLIED MATHEMATICS)
DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2018

หัวข้อโครงการพิเศษ โปรแกรมเพื่อแก้ปัญหาการมอบหมายงานหลายรูปแบบด้วยภาษาไพทอน
A PROGRAM FOR SOLVING VARIOUS ASSIGNMENT PROBLEMS
USING PYTHON

ชื่อนักศึกษา นางสาวรัชฎา พิสิทธิ์ รหัสนักศึกษา 58050135

นางสาวสิปang สายสุข รหัสนักศึกษา 58050167

ปริญญา วิทยาศาสตร์บัณฑิต (คณิตศาสตร์ประยุกต์)

ภาควิชา คณิตศาสตร์

คณะ วิทยาศาสตร์

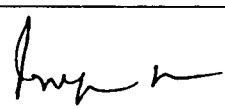

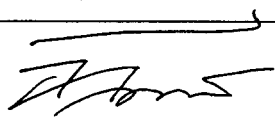
มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)

ปีการศึกษา 2561

อาจารย์ที่ปรึกษา รศ.ดร.ฉัฐไชย์ สีนาวงศ์

อาจารย์ที่ปรึกษาร่วม ดร.พุทธา สักกะพลางกูร

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติ
ให้โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชา
คณิตศาสตร์ประยุกต์ ประจำปีการศึกษา 2561

คณะกรรมการสอบ	ลายมือชื่อ
รศ. ไพโรบลย์ พันธรักษ์พงษ์ (ประธานกรรมการ)	
อ. ศิริกุล ศิริธีรากล (กรรมการ)	
รศ.ดร.ฉัฐไชย์ สีนาวงศ์ (กรรมการและอาจารย์ที่ปรึกษา)	
ดร.พุทธา สักกะพลางกูร (กรรมการและอาจารย์ที่ปรึกษาร่วม)	พธธ) สักกะพลางกูร

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงการพิเศษ	โปรแกรมเพื่อแก้ปัญหาการมอบหมายงานหลายรูปแบบด้วยภาษาไพทอน
ชื่อนักศึกษา	นางสาวรัชฎา พิสิทธิ์ รหัสนักศึกษา 58050135 นางสาวสิปง สายสุข รหัสนักศึกษา 58050167
ปริญญา	วิทยาศาสตรบัณฑิต (คณิตศาสตร์ประยุกต์)
ภาควิชา	คณิตศาสตร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2561
อาจารย์ที่ปรึกษา	รศ.ดร.ฉัฐไชย์ สีนาวงศ์
อาจารย์ที่ปรึกษาร่วม	ดร.พุทธา สักกะพลางกูร

บทคัดย่อ

ปัญหาการมอบหมายงาน (The Assignment Problem) หรือ การจับคู่ที่มีน้ำหนักรวมต่ำที่สุด (The Minimum – Weight Matching) คือ ปัญหาการหาค่าเหมาะที่สุดเชิงการจัด (Combinatorial Optimization Problem) ที่รู้จักกันดี เป็นปัญหาที่เกี่ยวข้องกับการทำงานทั้งหมดให้สำเร็จ โดยการกำหนดตัวแทนหนึ่งตัวแทนให้กับงานเพียงงานเดียว และในทางกลับกันเพื่อลดต้นทุนโดยรวมของการมอบหมายทั้งหมด ในงานวิจัยนี้ โปรแกรมใช้อัลกอริทึมของฮังการี (The Hungarian Algorithm) ในการแก้ปัญหา ได้รับการพัฒนาโดยใช้ภาษาโปรแกรมไพทอน (Python Programming Language) โปรแกรมจะลดความซับซ้อนของการคำนวณในการแก้ปัญหาการมอบหมายด้วยตนเอง โดยเฉพาะอย่างยิ่งสำหรับปัญหาขนาดใหญ่ ถูกออกแบบมาเพื่อจัดการกับปัญหาหลายรูปแบบ เช่น ปัญหาค่าสูงสุด (Maximization Problem) ปัญหาที่ไม่สามารถมอบหมายงานบางงานให้กับตัวแทนบางคน และตัวแทนกับงานจำนวนไม่เท่ากัน นอกจากนี้โปรแกรมนี้ มีจุดประสงค์เพื่อการใช้งานที่ง่ายโดยบุคคลทั่วไปที่มีพื้นฐานทางคณิตศาสตร์ที่จำกัด และสามารถนำไปใช้ในการตัดสินใจด้านการจัดการต่างๆ เช่น การผลิต โลจิสติกส์ หรือทรัพยากรมนุษย์ เป็นต้น

คำสำคัญ : การพัฒนาซอฟต์แวร์ การเพิ่มประสิทธิภาพ ปัญหาการมอบหมาย ฮังการีอัลกอริทึม

Title	A PROGRAM FOR SOLVING VARIOUS ASSIGNMENT PROBLEMS USING PYTHON		
Students	Ms. Ratchada Philuk	Student ID	58050135
	Ms. Sipang Saysuk	Student ID	58050167
Degree	Bachelor of Science (Applied Mathematics)		
Department	Mathematics		
Faculty	Science		
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)		
Academic Year	2018		
Advisor	Assoc. Prof. Dr. Chartchai Leenawong		
Co – Advisor	Dr. Puttha Sakkaplangkul		

ABSTRACT

The Assignment Problem or the minimum-weight matching is a well-known combinatorial optimization problem. The problem involves getting all tasks done by assigning exactly one agent to exactly one task and vice versa so as to minimize the total cost of all assignments. In this work, a program adopting the Hungarian algorithm for solving this problem is developed using the Python Programming Language. The program diminishes the computation complexity of solving the Assignment Problem manually, especially for the large-scale problems. It is designed to tackle several variations of the problem such as the maximization version of the problem, the problem having some prohibited agent-task assignments, and the unequal numbers of agents and tasks. Furthermore, this program is intended for easy use by general public with limited mathematical background and thus it can be used in various managerial decision contexts such as manufacturing, logistics, or human resource, to name just a few.

Keywords : Assignment Problem, Hungarian Algorithm, Optimization, Software Development

กิตติกรรมประกาศ

ในการจัดทำโครงการพิเศษนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์อย่างยิ่งจาก รศ.ดร.ฉัฐไชย์ สีนาวงศ์ และ ดร.พุทธา สักกะพลางกูร อาจารย์ที่ปรึกษา ซึ่งได้สละเวลาในการให้คำปรึกษา ข้อชี้แนะ คำแนะนำวิธีการในการศึกษา ให้คำปรึกษาในปัญหาที่เกิดขึ้น ช่วยแก้ไขข้อมูลที่ผิดพลาด ช่วยปรับปรุงเนื้อหา และเสนอแนะแนวทางในการจัดทำโครงการพิเศษฉบับนี้ ทั้งนี้ตลอดระยะเวลาการศึกษาวิจัย และความช่วยเหลือหลายสิ่งหลายอย่างจนกระทั่งลุล่วงไปได้ด้วยดี คณะผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

คณะผู้วิจัยขอกราบขอบพระคุณ รศ.ไพโรบลุย์ พันธรักษ์พงษ์ ประธานกรรมการสอบโครงการพิเศษ รวมไปถึง รศ.ดร.ฉัฐไชย์ สีนาวงศ์, ดร.พุทธา สักกะพลางกูร และอาจารย์ศิริกุล ศิริธีระกุล คณะกรรมการสอบโครงการพิเศษ ที่ตรวจและให้คำแนะนำในการแก้ไขข้อบกพร่องต่างๆ ทำให้งานวิจัยฉบับนี้สมบูรณ์มากยิ่งขึ้น

สุดท้ายนี้คณะผู้วิจัยหวังว่า โครงการพิเศษฉบับนี้จะเป็นประโยชน์ต่อวงการการศึกษา อุตสาหกรรมการขนส่ง อุตสาหกรรมต่างๆ และผู้ที่สนใจศึกษา หรือเป็นแนวทางในการศึกษาเพิ่มเติม

รัชฎา พิสิท

สิปาง สายสุข

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของปัญหา	2
1.3 ขอบเขตของปัญหา	2
1.4 ขั้นตอนในการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 แผนภาพการดำเนินงาน	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ทฤษฎีที่เกี่ยวข้อง	4
2.1.1 ปัญหาการมอบหมายงาน (The Assignment Problem)	4
2.1.2 วิธีแก้ปัญหาการมอบหมายงานแบบฮังการีเรียน (The Hungarian Method for Assignment Problem)	8
2.1.3 ภาษาไพทอน (Python Programming)	19
2.2 งานวิจัยที่เกี่ยวข้อง	20
2.2.1 การพัฒนาและประยุกต์ใช้ระบบการพยากรณ์ความต้องการ สินค้าและการจัดสรรงาน : กรณีศึกษาโรงงานผลชิ้นส่วน โทรศัพท์เคลื่อนที่	20
2.2.2 แอปพลิเคชันฮังการีเรียนในการคัดเลือกทีมเบสบอลและ การมอบหมาย (Application of the Hungarian Algorithm in Baseball Team Selection and Assignment)	22

สารบัญ(ต่อ)

	หน้า
บทที่ 3 วิธีการดำเนินงานวิจัย	23
3.1 การทำงานของโปรแกรมไพทอนสำหรับการมอบหมายงานในแต่ละรูปแบบ	23
3.1.1 โปรแกรมไพทอนปัญหาการมอบหมายงานสำหรับค่าต่ำสุด	23
3.1.2 โปรแกรมไพทอนปัญหาการมอบหมายงานสำหรับค่าสูงสุด	27
3.1.3 โปรแกรมไพทอนปัญหาการมอบหมายงานที่มีเส้นทางต้องห้าม	28
3.2 แผนภาพสรุปขั้นตอนในการทำงานของฮังกาเรียน	29
บทที่ 4 ผลการดำเนินงาน	30
4.1 วิธีการใช้งานโปรแกรมฮังกาเรียนเพื่อแก้ปัญหาการมอบหมายงาน	30
บทที่ 5 สรุปผลการดำเนินงาน	39
5.1 สรุปผลการดำเนินงาน	39
5.2 ข้อเสนอแนะ	40
เอกสารอ้างอิง	41
ภาคผนวก ก.	42
ก.1 ไพชาร์ม (Pycharm)	43
ก.2 ขั้นตอนการติดตั้ง Pycharm	43

สารบัญตาราง

ตารางที่	หน้า
2.1.1 ตารางแสดงต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร	10
2.1.2 ตารางแสดงผลลัพธ์การมอบหมายการผลิตอะไหล่รถยนต์โดยเครื่องจักร	13
2.1.3 ตารางแสดงต้นทุนการมอบหมายงานให้คนงานโดยเครื่องจักร	14
2.1.4 ตารางแสดงผลลัพธ์ต้นทุนการมอบหมายงานให้คนงานโดยเครื่องจักร	15
2.1.5 ตารางแสดงค่าธุรกิจสูงสุดต่อเดือน	15
2.1.6 ตารางแสดงผลลัพธ์การมอบหมายงานให้ได้ค่าธุรกิจสูงสุดต่อเดือน	16
2.1.7 ตารางแสดงต้นทุนการขนส่งสินค้าจากโรงงานผลิตสินค้าไปยังเมืองต่างๆ	17
2.1.8 ตารางแสดงผลลัพธ์การขนส่งสินค้าจากโรงงานผลิตสินค้าไปยังเมืองต่างๆ	18
2.2.1 ตารางการมอบหมายรายได้จากการใช้โปรแกรมเชิงเส้น	20
2.2.2 ตารางการเปรียบเทียบรอบเวลาการทำงานก่อนและหลังการมอบหมายงาน	21
2.2.3 ตารางแสดงผลลัพธ์ของการมอบหมายรายได้จากการใช้โปรแกรมเชิงเส้น	21
4.1.1 ตารางแสดงต้นทุนระหว่างศูนย์กระจายสินค้า 8 แห่ง ไปยังลูกค้า 8 ราย	31
4.1.2 ตารางแสดงค่าประสิทธิภาพการเข้าจอดของเครื่องบิน 8 ลำ เทียบประตู	36
ขึ้นเครื่อง 7 ประตู	

สารบัญรูป

รูปที่	หน้า
2.1.1 ตัวอย่างการมอบหมายงาน	4
3.1.1.1 การดำเนินการของโปรแกรมปัญหาการมอบหมายสำหรับค่าต่ำสุด	23
3.1.1.2 ดำเนินการนำค่าต่ำสุดในแต่ละแถวและคอลัมน์ไปลบทุกค่าที่อยู่ใน แต่ละแถวและคอลัมน์	24
3.1.1.3 การดำเนินการนับจำนวนเลขศูนย์ในแต่ละแถวและคอลัมน์เลือกแถวหรือคอลัมน์	25
3.1.1.4 การดำเนินการเช็คจำนวนแถวและคอลัมน์ที่เลือก และเปรียบเทียบ	26
กับจำนวนแถวของเมทริกซ์	
3.1.1.5 การดำเนินการเลือกคำตอบที่ดีที่สุด	27
3.1.2.1 การดำเนินการหาค่าที่มากที่สุด และนำไปลบออกจากทุกค่าในเมทริกซ์	28
3.1.3.1 การดำเนินการกำหนดเส้นทางต้องห้ามให้มีค่ามากที่สุด ในเมทริกซ์	28
3.2.1 แผนภาพแสดงการทำงานของฮังกาเรียน	29
4.1.1 โปรแกรมฮังกาเรียนเพื่อแก้ปัญหาการมอบหมายงาน	30
4.1.2 กำหนดข้อมูลลงในโปรแกรมฮังกาเรียน	31
4.1.3 ตารางสำหรับกรอกข้อมูลในโปรแกรมฮังกาเรียน	32
4.1.4 ตารางแสดงข้อมูลที่ผู้ใช้งานกรอกในโปรแกรมฮังกาเรียน	32
4.1.5 ตารางแสดงผลลัพธ์ในโปรแกรมฮังกาเรียน	33
4.1.6 ตารางแสดงข้อมูลเส้นทางต้องห้ามที่ผู้ใช้งานกรอกในโปรแกรมฮังกาเรียน	34
4.1.7 ตารางแสดงผลลัพธ์โปรแกรมฮังกาเรียน	35
4.1.8 กำหนดความต้องการลงในโปรแกรมฮังกาเรียน	37
4.1.9 ตารางสำหรับกรอกข้อมูลในโปรแกรมฮังกาเรียน	37
4.1.10 ตารางแสดงผลลัพธ์ในโปรแกรมฮังกาเรียน	38
ก. 2.1 เว็บไซต์สำหรับดาวน์โหลดโปรแกรม Pycharm	43
ก.2.2 หน้าต่าง Pycharm Community Edition Setup	43
ก.2.3 หน้าต่าง Installation Options	44
ก.2.4 หน้าต่าง Installing Pycharm	44
ก.2.5 หน้าต่าง Create New Project	45
ก.2.6 หน้าต่างกำหนด Location ของไฟล์	45

สารบัญรูป

รูปที่		หน้า
ก.2.7	หน้าต่างสร้างไฟล์ไพทอน (New Python File)	46
ก.2.8	หน้าต่างตั้งชื่อไฟล์ไพทอน (New Python File Name)	46
ก.2.9	หน้าต่างการใช้งานไพทอน	47

บทที่ 1

บทนำ

1.1. ที่มาและความสำคัญของปัญหา

สภาพเศรษฐกิจในปัจจุบันที่มีสถานะการแข่งขันสูง จึงมีระบบการจัดการต่างๆ เพื่อลดต้นทุนของกระบวนการการทำงาน ทำให้ผลประกอบการของบริษัทยังคงมีผลกำไรของส่วนต่างระหว่างต้นทุนและยอดขาย ในบริษัทมีต้นทุนที่ต้องใช้ทางด้านวัตถุดิบ พนักงาน และอื่นๆ จำนวนมาก ทำให้ต้นทุนเรื่องวัสดุและค่าแรงสูงมากตามไปด้วย ปัจจุบันผู้บริหารหันมาให้ความสำคัญกับการวางแผนการลงทุนโดยคำนึงถึงต้นทุนที่ต่ำที่สุดและหวังผลกำไรที่มากที่สุด จึงมีการวางแผนการดำเนินงานในธุรกิจอย่างรอบคอบ เพื่อประหยัดต้นทุนให้มากที่สุด ทำให้เกิดปัญหาการมอบหมายงาน (The Assignment Problem) หรือ การจับคู่ที่มีน้ำหนักรวมต่ำสุด (The Minimum – Weight Matching) ขึ้น ดังนั้นเราจึงใช้วิธีการแก้ปัญหาการมอบหมายงานแบบฮังการีเรียน (The Hungarian Algorithm) เข้ามาช่วยในการแก้ปัญหา

เนื่องจากปัญหาการมอบหมายงานจริงอาจมีขนาดใหญ่มาก ทำให้เกิดความซับซ้อนและใช้เวลาในการคำนวณด้วยตนเองมากเช่นกัน ซึ่งง่ายที่จะเกิดข้อผิดพลาดในการคำนวณ จึงมีการสร้างโปรแกรมสำหรับการแก้ปัญหาการมอบหมายงานที่ง่ายและสะดวกสำหรับการใช้งานจริงอย่างมีประสิทธิภาพ ช่วยในการตัดสินใจ ลดข้อผิดพลาดในการคำนวณและประหยัดเวลา โดยนำอัลกอริทึมของฮังการีเรียน มาใช้เพื่อแก้ปัญหาการมอบหมายงาน พัฒนาโดยใช้ภาษาไพทอน (Python Programming Language) ซึ่งเป็นภาษาที่อ่านง่าย ถูกออกแบบมาให้มีโครงสร้างที่มองเห็นได้โดยไม่ซับซ้อน โปรแกรมสำหรับการแก้ปัญหาการมอบหมายงานนี้มีการวางแผนทรัพยากรได้อย่างมีประสิทธิภาพ เพื่อลดต้นทุนรวมในส่วนของการจัดสรรทรัพยากรบุคคลกับงานนั้นๆ นอกจากนี้ตัวโปรแกรมยังสามารถแก้ไขปัญหาค่าสูงสุด ปัญหาที่ไม่สามารถมอบหมายงานบางงานให้กับคนทำงานบางคน และคนทำงานกับงานจำนวนไม่เท่ากัน สามารถนำไปประกอบการตัดสินใจในด้านการจัดการต่างๆ เช่น การขนส่งสินค้า การผลิตสินค้า ทรัพยากรมนุษย์ และอื่นๆ อีกมากมาย

1.2. วัตถุประสงค์ของงานวิจัย

1. เพื่อศึกษาและเข้าใจวิธีการแก้ปัญหาการมอบหมายงานโดยวิธีฮังการีเรียน
2. เพื่อสร้างโปรแกรมสำหรับการแก้ปัญหาการมอบหมายงาน โดยใช้วิธีการแก้ปัญหาโดยวิธีฮังการีเรียนได้อย่างมีประสิทธิภาพ และสามารถนำไปใช้งานได้จริง
3. เพื่อวางแผนการจัดสรร โดยวิธีที่เหมาะสมกับลักษณะปัญหาการมอบหมายงานในทุกรูปแบบ
4. เพื่อประยุกต์ความรู้ความเข้าใจ ในการวางแผนการจัดสรรงานให้เข้ากับปัญหาปัจจุบันได้อย่างถูกต้องและแม่นยำ

1.3. ขอบเขตของงานวิจัย

โปรแกรมคอมพิวเตอร์สำหรับพิจารณาปัญหาการมอบหมายงานในทุกรูปแบบที่อาจเป็นไปได้ วิเคราะห์ผลของปัญหาการมอบหมายงานแบบสมดุล (The Balanced Assignment Problem) และปัญหาการมอบหมายงานแบบไม่สมดุล (The Unbalanced Assignment Problem) รวมไปถึงปัญหาการมอบหมายงานต่ำสุด (The Minimization Assignment Problem) ปัญหาการมอบหมายงานสูงสุด (The Maximization Assignment Problem) และปัญหาการมอบหมายที่มีเส้นทางต้องห้าม (The Assignment Problem with Prohibited Route) โดยพัฒนาโปรแกรมคอมพิวเตอร์ด้วยภาษาไพทอน (Python Programming)

1.4. ขั้นตอนในการดำเนินงาน

1. ค้นคว้าเอกสารและข้อมูลที่เกี่ยวข้องวิธีการแก้ปัญหาการมอบหมายงานโดยวิธีฮังการีเรียน
2. เข้าใจแบบจำลองวิธีการแก้ปัญหาโดยวิธีฮังการีเรียน
3. เขียนโปรแกรมของแบบจำลองวิธีการแก้ปัญหาโดยวิธีฮังการีเรียน
4. นำเสนอโปรแกรมของแบบจำลองวิธีการแก้ปัญหาโดยวิธีฮังการีเรียน
5. แก้ไขข้อบกพร่อง พร้อมทำรูปเล่มโครงงานปัญหาพิเศษเรื่อง
6. สร้างตัวอย่างปัญหาที่จะนำมาทดสอบโปรแกรมการแก้ปัญหาการมอบหมายงาน

1.5. ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำความรู้ที่ได้ไปใช้ในการแก้ไขปัญหาการมอบหมายงานได้อย่างมีประสิทธิภาพ
2. สามารถคำนวณหาต้นทุนต่ำสุด หรือกำไรสูงสุด ของปัญหาการมอบหมายงานได้
3. ช่วยให้ผู้บริหารวางแผนการดำเนินงานในธุรกิจได้อย่างรอบคอบ และประหยัดต้นทุน
4. ทำให้ผู้บริหารหันมาให้ความสำคัญกับการวางแผนการลงทุนโดยคำนึงถึงต้นทุนที่ต่ำที่สุด และหวังผลกำไรที่มากที่สุด
5. ได้เครื่องมือที่ช่วยในการตัดสินใจมอบหมายงาน

บทที่ 2

ทฤษฎีบทที่เกี่ยวข้อง

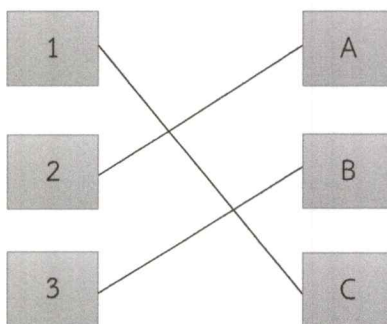
ในบทนี้จะแสดงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับปัญหาการมอบหมายงาน เพื่อทำความเข้าใจและเป็นแนวทางในการนำไปประยุกต์ใช้ในการแก้ปัญหาการมอบหมายงาน และสามารถเลือกวิธีการมอบหมายงานที่เหมาะสมที่สุดในการแก้ปัญหา

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องกับปัญหาพิเศษนี้ประกอบไปด้วย ปัญหาการมอบหมายงาน วิธีการแก้ปัญหาการมอบหมายงานแบบฮังกาเรียน และภาษาไพทอนที่นำมาพัฒนาโปรแกรมคอมพิวเตอร์

2.1.1 ปัญหาการมอบหมายงาน (The Assignment Problem)

ปัญหาการมอบหมายงาน (The Assignment Problem) เป็นปัญหาที่พัฒนามาจากรูปแบบปัญหาการขนส่ง (Transportation Problem) ได้แก่ ปัญหาที่เกี่ยวกับการจัดสรรบุคลากร อุปกรณ์ เครื่องจักร ให้เหมาะสมกับงาน หน้าที่ สถานที่ โดยคำนึงถึงค่าใช้จ่ายต่ำสุด หรือในบางปัญหาเพื่อให้ได้ประโยชน์สูงสุดหรือกำไรสูงสุด



รูปที่ 2.1.1 ตัวอย่างการมอบหมายงาน

ปัญหาการมอบหมายงานโดยทั่วไป มีลักษณะดังต่อไปนี้

1. จำนวนวัตถุของทั้งสองกลุ่มต้องเท่ากัน (Balanced) เช่น มีค่าเท่ากับ m
2. การจับคู่วัตถุทั้งสองกลุ่มต้องเป็นแบบฟังก์ชันหนึ่งต่อหนึ่ง (One – to – One Function) และทั่วถึง (Onto Function) กล่าวคือ วัตถุแต่ละชิ้นทั้งในกลุ่มแรกและกลุ่มสองต้องถูกจับคู่เพียงหนึ่งครั้งเท่านั้น และวัตถุทุกชิ้นในทั้งสองกลุ่มจะต้องถูกจับคู่ทั้งหมด
3. ต้องมีฟังก์ชันแสดงถึงความเข้ากันได้ (Fitness Function) ที่ต้องการทำให้สูงสุดหรือต่ำสุด เช่น ต้นทุน เวลา หรือความพึงพอใจ

ปัญหาการมอบหมายงานแบบสมมูลสามารถเขียนให้อยู่ในรูปแบบจำลองการตัดสินใจได้ ยกตัวอย่างการเขียนแบบจำลองสำหรับปัญหาการมอบหมายงานจำนวน m คน ให้กับคนทำงาน m งาน โดยมีวัตถุประสงค์เพื่อให้ได้ต้นทุนรวมต่ำที่สุด

กำหนดให้ข้อมูลและตัวแปรการตัดสินใจของปัญหาเป็นดังนี้

C_{ij} = ต้นทุนการมอบหมายงานที่ i ให้คนทำงานคนที่ j

X_{ij} = ตัวแปรฐานสอง มีค่าเป็น 0 หรือ 1 โดยที่

$$X_{ij} = \begin{cases} 1 & \text{ถ้าคนที่ } i \text{ ให้ทำงานที่ } j \\ 0 & \text{ถ้าคนที่ } i \text{ ไม่ได้ทำงานที่ } j \end{cases}$$

เมื่อ i เท่ากับ คนที่ $1, 2, 3, \dots, m$

j เท่ากับ งานที่ $1, 2, 3, \dots, m$

โดยวัตถุประสงค์ที่ต้องการให้ได้ต้นทุนรวมต่ำที่สุด สามารถแปลงเป็นฟังก์ชันวัตถุประสงค์ได้ดังต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & C_{11}X_{11} + C_{12}X_{12} + C_{13}X_{13} + \cdots + C_{1m}X_{1m} \\ & C_{21}X_{21} + C_{22}X_{22} + C_{23}X_{23} + \cdots + C_{2m}X_{2m} \\ & \cdot \\ & \cdot \\ & \cdot \\ & C_{m1}X_{m1} + C_{m2}X_{m2} + C_{m3}X_{m3} + \cdots + C_{mm}X_{mm} \end{aligned}$$

มีข้อจำกัด (Constraint) 2 ข้อ ดังต่อไปนี้

- ข้อจำกัดของวัตถุในกลุ่มที่ 1 คนทำงานหนึ่งคน จะได้รับมอบหมายให้ทำงานได้เพียงหนึ่งงานเท่านั้น นั่นคือ

$$\begin{aligned} (\text{คนที่ } 1) \quad & X_{11} + X_{12} + X_{13} + \cdots + X_{1m} = 1 \\ (\text{คนที่ } 2) \quad & X_{21} + X_{22} + X_{23} + \cdots + X_{2m} = 1 \\ & \cdot \\ & \cdot \\ & \cdot \\ (\text{คนที่ } m) \quad & X_{m1} + X_{m2} + X_{m3} + \cdots + X_{mm} = 1 \quad (2.1) \end{aligned}$$

- ข้อจำกัดของวัตถุในกลุ่มที่ 2 งานหนึ่งงานจะถูกมอบหมายให้กับคนทำงานได้เพียงหนึ่งคนเท่านั้น นั่นคือ

$$\begin{aligned} (\text{งานที่ } 1) \quad & X_{11} + X_{21} + X_{31} + \cdots + X_{m1} = 1 \\ (\text{งานที่ } 2) \quad & X_{12} + X_{22} + X_{32} + \cdots + X_{m2} = 1 \\ & \cdot \\ & \cdot \\ & \cdot \\ (\text{งานที่ } m) \quad & X_{1m} + X_{2m} + X_{3m} + \cdots + X_{mm} = 1 \quad (2.2) \end{aligned}$$

แบบจำลองการตัดสินใจสำหรับปัญหาการมอบหมายงาน สามารถเขียนใหม่ได้ในรูปแบบกระชับ (Compact Form) ได้ดังนี้

Minimize

$$\sum_{i=1}^m \sum_{j=1}^m C_{ij} X_{ij}$$

(2.3)

Constraint

$$\sum_{j=1}^m X_{ij} = 1 \quad i = 1, 2, 3, \dots, m$$

(2.4)

$$\sum_{i=1}^m X_{ij} = 1 \quad j = 1, 2, 3, \dots, m$$

(2.5)

$$X_{ij} \geq 0 \quad i, j = 1, 2, 3, \dots, m$$

$$X_{ij} = \begin{cases} 1 & \text{ถ้าคนที่ } i \text{ ให้ทำงานที่ } j \\ 0 & \text{ถ้าคนที่ } i \text{ ไม่ได้ทำงานที่ } j \end{cases}$$

โดยที่ m คือจำนวนคน หรือ จำนวนงาน

ปัญหาการมอบหมายงานแบ่งออกเป็น 3 แบบ ดังต่อไปนี้

1. ปัญหาการมอบหมายงานแบบสมดุล (The Balanced Assignment Problem)

ปัญหาการมอบหมายงานแบบสมดุลสามารถเขียนให้อยู่ในรูปแบบจำลองการตัดสินใจ เป็นแบบจำลองสำหรับปัญหาการมอบหมายงานคนจำนวน m คนให้ทำงาน m งาน โดยมีวัตถุประสงค์เพื่อให้ได้ต้นทุนต่ำสุด หรือกำไรสูงสุด ขึ้นอยู่กับปัญหาที่ได้รับมอบหมาย

2. ปัญหาการมอบหมายงานแบบไม่สมดุล (The Unbalanced Assignment Problem)

ปัญหาการมอบหมายงานแบบไม่สมดุลสามารถเขียนให้อยู่ในรูปแบบจำลองการตัดสินใจ เป็นแบบจำลองสำหรับปัญหาการมอบหมายงานคนจำนวน m คนให้ทำงาน n งาน กล่าวคือจำนวนงานที่มอบหมายไม่เท่ากับจำนวนคนที่ได้รับมอบหมายงาน โดยมีวัตถุประสงค์เพื่อให้ได้ต้นทุนต่ำสุด หรือกำไรสูงสุด ขึ้นอยู่กับปัญหาที่ได้รับมอบหมาย วิธีการเปลี่ยนให้เป็นปัญหาแบบสมดุลทำได้โดยการเพิ่มคนหรืองานสมมติ (Dummy Person or Job) เข้าไปเพื่อให้จำนวนคนเท่ากับจำนวนงาน

3. ปัญหาการมอบหมายที่มีเส้นทางต้องห้าม (The Assignment Problem with Prohibited Routes)

ปัญหาการขนส่งที่มีเส้นทางต้องห้ามนี้ จะสามารถใช้ตารางการขนส่งเพื่อหาคำตอบที่เหมาะสมที่สุดได้ โดยการกำหนดให้เส้นทางจาก i ไป j ที่เป็นเส้นทางต้องห้ามนั้น มีค่าขนส่งต่อหน่วย C_{ij} สูงมาก โดยเฉพาะเมื่อเทียบกับค่าขนส่งต่อหน่วยที่เหลือในช่องอื่น

จากปัญหาการมอบหมายงานในรูปของแบบจำลองการตัดสินใจ สามารถหาคำตอบที่ดีที่สุดได้หลายวิธี ในงานวิจัยนี้จะใช้วิธีแก้ปัญหาการมอบหมายงานแบบฮังการี (The Hungarian Method for Assignment Problem) มาใช้ในการแก้ปัญหา

2.1.2 วิธีแก้ปัญหาการมอบหมายงานแบบฮังการี (The Hungarian Method for the Assignment Problem)

วิธีแก้ปัญหาการมอบหมายงานแบบฮังการี เป็นวิธีการหาคำตอบที่ดีที่สุดสำหรับปัญหาการมอบหมายงานที่มีขนาดเล็ก พัฒนามาจากวิธีการซิมเพล็กซ์ (Simplex Algorithm) วิธีนี้สามารถหาผลลัพธ์ได้รวดเร็วและเป็นที่ยอมรับ

วิธีฮังการีถูกคิดค้นขึ้นมาในปี พ.ศ. 2549 โดยนักคณิตศาสตร์ชาวฮังการี ชื่อว่า D.Konig เรียกวิธีการแก้ปัญหานี้ว่า วิธีฮังการี (Hungarian Method) แบ่งออกเป็น 2 ประเภท ดังต่อไปนี้

1. วิธีฮังการีสำหรับปัญหาค่าต่ำสุด (Hungarian Method for Balanced Minimization Assignment Problem)
2. วิธีฮังการีสำหรับปัญหาค่าสูงสุด (Hungarian Method for Balanced Maximization Assignment Problem)

วิธีฮังกาเรียนมีหลักการใช้ทั่วไปคือการลดตัวเลขในตารางตามวิธีทางคณิตศาสตร์ ทำให้เกิดค่าศูนย์ในตาราง ขั้นตอนการคำนวณมีดังต่อไปนี้

ขั้นตอนที่ 1 จากปัญหาสำหรับค่าต่ำสุด กำหนดให้ต้นทุนการมอบหมายงานเท่ากับ

$$i (i = 1, 2, 3, \dots, m) \text{ ไปยังคนทำงาน } j (j = 1, 2, 3, \dots, m) \text{ เท่ากับ } C_{ij}$$

ขั้นตอนที่ 2 หาค่าต่ำสุดของ C_{ij} ในแถวและคอลัมน์ นำค่าต่ำสุดนั้นมาลบทุกค่าในแถวและคอลัมน์เดียวกัน

ขั้นตอนที่ 3 จากผลลัพธ์ที่ได้จากขั้นตอนที่ 2 ทำการลากเส้นตรงผ่านแถวหรือคอลัมน์ให้เส้นตรงผ่านแถวหรือคอลัมน์ที่มีเลขศูนย์มากที่สุด และให้ได้จำนวนเส้นที่ลากผ่านน้อยที่สุด

ขั้นตอนที่ 4 นับจำนวนเส้นตรงที่ลากผ่านแถวหรือคอลัมน์ ถ้าจำนวนเส้นตรงเท่ากับจำนวนแถวหรือจำนวนคอลัมน์ให้หยุดทำงาน และจับคู่งานกับคนทำงานโดยเริ่มจากจับคู่ในช่องที่มีเลขศูนย์ไม่ซ้ำกันเป็นอันดับแรก และทำการจับคู่งานกับคนทำงานจนครบทุกช่อง จะได้คำตอบการมอบหมายงานที่ดีที่สุด ถ้าจำนวนเส้นตรงไม่เท่ากับจำนวนแถวหรือคอลัมน์ให้กระทำในขั้นตอนต่อไป

ขั้นตอนที่ 5 พิจารณาค่าที่ไม่มีเส้นตรงลากผ่าน และนำค่าที่ต่ำที่สุดจากค่าที่ไม่มีเส้นตรงลากผ่านไปลบออกจากทุกค่าที่ไม่มีเส้นตรงลากผ่าน จากนั้นนำไปบวกกับค่าที่อยู่บนจุดตัดของเส้นตรงสองเส้นในตาราง และกลับไปทำขั้นตอนที่ 3 ทำซ้ำจนกว่าจะได้จำนวนเส้นตรงเท่ากับจำนวนแถวหรือคอลัมน์

หมายเหตุ สำหรับปัญหาค่าสูงสุด จะต้องทำการปรับตารางความพึงพอใจให้เป็นปัญหาค่าต่ำสุดในขั้นตอนแรก นั่นคือ เลือกค่าสูงสุดในตารางเป็นตัวตั้ง ค่าใหม่แต่ละช่องจะได้จากค่าสูงสุดลบด้วยค่าที่อยู่ในช่องนั้น จากนั้นทำการดำเนินการตามขั้นตอนที่ 1 ถึงขั้นตอนที่ 5

ตัวอย่างที่ 2.1 ตัวอย่างปัญหาการมอบหมายงานแบบสมมูลของปัญหาค่าต่ำสุด

บริษัทอุตสาหกรรมการผลิต จำกัด ทำธุรกิจเกี่ยวกับการผลิตอะไหล่รถยนต์ เพื่อขายให้กับบริษัทประกอบรถยนต์ ขณะนี้บริษัทมีงานอยู่ 3 ชิ้น คือ A , B และ C และมีเครื่องจักร 3 ชุด คือ X , Y และ Z ตามลำดับ อะไหล่ทั้ง 3 ชิ้นนั้น สามารถที่จะผลิตด้วยเครื่องจักรชุดใดๆ ก็ได้ แต่จะมีค่าใช้จ่ายแตกต่างกันออกไปซึ่งทางบริษัทได้คำนวณไว้แล้ว บริษัทมีนโยบายที่จะผลิตอะไหล่แต่ละชิ้นด้วยเครื่องจักรชุดใดชุดหนึ่งเพียงชุดเดียวเท่านั้น โดยให้ต้นทุนการผลิตรวมต่ำที่สุด ต้นทุนการผลิตอะไหล่แต่ละชิ้นโดยเครื่องจักรแต่ละชุด แสดงในตารางที่ 2.1.1 บริษัทจะตัดสินใจอย่างไร จึงจะเสียค่าใช้จ่ายในการผลิตรวมต่ำที่สุด

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$25	\$31	\$35
B	\$15	\$20	\$24
C	\$22	\$19	\$17

ตารางที่ 2.1.1 ตารางแสดงต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร

ขั้นตอนที่ 1 กำหนดให้ต้นทุนการมอบหมายงานเท่ากับ i ($i = 1, 2, 3, \dots, m$) ไปยัง คนทำงาน j ($j = 1, 2, 3, \dots, m$) เท่ากับ C_{ij}

ขั้นตอนที่ 2 หาค่าต่ำสุดของ C_{ij} ในแต่ละแถวและคอลัมน์ นำค่าต่ำสุดนั้นมาลบทุกค่าในแถวและคอลัมน์เดียวกัน

- แถวแนวนอน (Row) ให้เลือกค่าต่ำสุดในแต่ละแถว นำไปลบออกจากค่าอื่นๆ ที่อยู่ในแถวเดียวกัน ทำในลักษณะนี้ทุกแถวแนวนอน

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร			
	X	Y	Z	
A	\$25	\$31	\$35	25
B	\$15	\$20	\$24	15
C	\$22	\$19	\$17	17

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$0	\$6	\$10
B	\$0	\$5	\$9
C	\$5	\$2	\$0

- แถวแนวตั้ง (Column) ดำเนินการแถวแนวตั้งด้วยวิธีเดียวกับแถวแนวนอน นั่นคือ เลือกค่าต่ำสุดของแถวแนวตั้งในแต่ละแถวแนวตั้ง นำค่าต่ำสุดไปหักจากค่าอื่น ๆ ในแถวแนวตั้งนั้นๆ ทำในลักษณะนี้ทุกแถวแนวตั้ง

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$0	\$6	\$10
B	\$0	\$5	\$9
C	\$5	\$2	\$0
	0	2	0

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$0	\$4	\$10
B	\$0	\$3	\$9
C	\$5	\$0	\$0

ขั้นตอนที่ 3 จากผลลัพธ์ที่ได้จากขั้นตอนที่ 2 ทำการลากเส้นตรงผ่านแถวหรือคอลัมน์ ให้เส้นตรงผ่านแถวหรือคอลัมน์ที่มีเลขศูนย์มากที่สุดเป็นลำดับแรก

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$0	\$4	\$10
B	\$0	\$3	\$9
C	\$5	\$0	\$0

ขั้นตอนที่ 4 นับจำนวนเส้นตรงที่ลากผ่านแถวและคอลัมน์ ถ้าจำนวนเส้นตรงเท่ากับจำนวนแถวหรือจำนวนคอลัมน์ให้ทำการจับคู่ โดยเริ่มจากจับคู่ในช่องที่มีเลขศูนย์ไม่ซ้ำกันเป็นอันดับแรก จากนั้นทำการจับคู่อะไหล่รถยนต์กับชุดเครื่องจักรจนครบทุกช่อง จะได้ผลลัพธ์การมอบหมายงานที่ดีที่สุด ถ้าจำนวนเส้นตรงไม่เท่ากับจำนวนแถวหรือคอลัมน์ ให้กระทำในขั้นตอนต่อไป

เนื่องจากจำนวนเส้นตรงที่ลากผ่านแถวและคอลัมน์ไม่เท่ากับจำนวนแถวหรือจำนวนคอลัมน์ ดังนั้นจะต้องดำเนินการในขั้นตอนต่อไป

ขั้นตอนที่ 5 พิจารณาค่าที่ไม่มีเส้นตรงลากผ่าน และนำค่าที่ต่ำที่สุดจากค่าที่ไม่มีเส้นตรงลากผ่าน ไปลบออกจากทุกค่าที่ไม่มีเส้นตรงลากผ่าน และนำไปบวกกับค่าที่อยู่บนจุดตัดของเส้นตรงสองเส้นในตารางและกลับไปทำขั้นตอนที่ 3 ทำซ้ำจนกว่าจะได้จำนวนเส้นตรงเท่ากับจำนวนแถวหรือคอลัมน์

จากข้อมูลในตาราง ค่าที่ต่ำที่สุดที่ไม่มีเส้นตรงลากผ่าน คือ 3 ดังนั้นจึงนำ 3 นำไปลบกับทุกค่าที่ไม่มีเส้นตรงลากผ่าน และไปบวกกับทุกค่าที่อยู่บนจุดตัดของเส้นตรงสองเส้น จะได้ผลลัพธ์ดังตารางต่อไปนี้

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$0	\$1	\$7
B	\$0	\$0	\$6
C	\$8	\$0	\$0

จากนั้นย้อนกลับไปดำเนินการตามขั้นตอนที่ 3 ทำซ้ำจนกว่าจะได้จำนวนเส้นตรงเท่ากับจำนวนแถวหรือคอลัมน์

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$0	\$1	\$7
B	\$0	\$0	\$6
C	\$8	\$0	\$0

เนื่องจากเส้นตรงที่ลากผ่านแถวและคอลัมน์ มีจำนวนเท่ากับจำนวนแถวและคอลัมน์จึงสามารถทำการจับคู่อะไหล่รถยนต์กับชุดเครื่องจักรได้ดังต่อไปนี้

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$0	\$1	\$7
B	\$0	\$0	\$6
C	\$8	\$0	\$0

สามารถสรุปผลการมอบหมายงานสำหรับอะไหล่รถยนต์ 3 ชิ้นให้กับชุดเครื่องจักร 3 ชุดได้ดังต่อไปนี้

อะไหล่รถยนต์	ต้นทุนต่อหน่วยในการผลิตอะไหล่รถยนต์โดยเครื่องจักร		
	X	Y	Z
A	\$25	\$31	\$35
B	\$15	\$20	\$24
C	\$22	\$19	\$17

ตารางที่ 2.1.2 ตารางแสดงผลลัพธ์การมอบหมายการผลิตอะไหล่รถยนต์โดยเครื่องจักร

บริษัทควรมอบหมายงาน ดังต่อไปนี้

อะไหล่รถยนต์ A ผลิตโดยเครื่องจักร X ต้นทุนการผลิต มีค่าเท่ากับ \$25

อะไหล่รถยนต์ B ผลิตโดยเครื่องจักร Y ต้นทุนการผลิต มีค่าเท่ากับ \$20

อะไหล่รถยนต์ C ผลิตโดยเครื่องจักร Z ต้นทุนการผลิต มีค่าเท่ากับ \$17

ดังนั้น ต้นทุนการผลิตอะไหล่รถยนต์รวม เท่ากับ \$62

ตัวอย่างที่ 2.2 ปัญหาการมอบหมายงานแบบไม่สมดุลของปัญหาค่าต่ำสุด

บริษัทหนึ่งต้องการมอบหมายงานให้กับคนงานเพื่อควบคุมการทำงานของเครื่องจักรภายในบริษัท โดยบริษัทแห่งนี้ มีเครื่องจักรทั้งหมด 4 เครื่อง ได้แก่ เครื่องจักร A เครื่องจักร B เครื่องจักร C และเครื่องจักร D ต้องการมอบหมายงานให้กับคนงานทั้งหมด 3 คน ได้แก่ คนงานที่ 1 คนงานที่ 2 และ คนงานที่ 3 เนื่องจากการมอบหมายงานจะต้องมีต้นทุนในการมอบหมายงาน ดังนั้นทางบริษัทจึงต้องการให้มีการมอบหมายงานโดยใช้ต้นทุนรวมต่ำที่สุด ต้นทุนการมอบหมายงานแสดงในตารางที่ 2.1.3 บริษัทจะตัดสินใจอย่างไร จึงจะเสียค่าใช้จ่ายในการมอบหมายงานรวมต่ำที่สุด

คนงาน	ต้นทุนต่อหน่วยในการมอบหมายงานให้คนงานโดยเครื่องจักร			
	A	B	C	D
1	\$5	\$3	\$7	\$9
2	\$1	\$2	\$5	\$6
3	\$5	\$7	\$5	\$3

ตารางที่ 2.1.3 ตารางแสดงต้นทุนการมอบหมายงานให้คนงานโดยเครื่องจักร

เนื่องจากตัวอย่างนี้เป็นปัญหาการมอบหมายงานแบบไม่สมดุล จำนวนคนงานกับจำนวนเครื่องจักรไม่เท่ากัน ดังนั้นจะต้องทำการใส่ดัมมี่ (Dummy) เพื่อปรับให้เป็นปัญหาการมอบหมายงานแบบสมดุล จึงจะสามารถดำเนินการในขั้นตอนการดำเนินงานแบบการแก้ปัญหามอบหมายงานแบบสมดุลของปัญหาค่าต่ำสุดได้

คนงาน	ต้นทุนต่อหน่วยในการมอบหมายงานให้คนงานโดยเครื่องจักร			
	A	B	C	D
1	\$5	\$3	\$7	\$9
2	\$1	\$2	\$5	\$6
3	\$5	\$7	\$5	\$3
4	\$0	\$0	\$0	\$0

ผลการมอบหมายงานการควบคุมเครื่องจักร 4 เครื่อง ให้กับคนทำงาน 3 คน ดังต่อไปนี้

คนงาน	ต้นทุนต่อหน่วยในการมอบหมายงานให้คนงานโดยเครื่องจักร			
	A	B	C	D
1	\$5	\$3	\$7	\$9
2	\$1	\$2	\$5	\$6
3	\$5	\$7	\$5	\$3
4	\$0	\$0	\$0	\$0

ตารางที่ 2.1.4 ตารางแสดงผลลัพธ์ต้นทุนการมอบหมายงานให้คนงานโดยเครื่องจักร

บริษัทควรมอบหมายงาน ดังต่อไปนี้

คนงานคนที่ 1 ได้รับมอบหมายให้ควบคุมเครื่องจักร B ต้นทุนการมอบหมายงาน เท่ากับ \$3

คนงานคนที่ 2 ได้รับมอบหมายให้ควบคุมเครื่องจักร A ต้นทุนการมอบหมายงาน เท่ากับ \$1

คนงานคนที่ 3 ได้รับมอบหมายให้ควบคุมเครื่องจักร D ต้นทุนการมอบหมายงาน เท่ากับ \$3

ดังนั้น ต้นทุนการมอบหมายงานรวม เท่ากับ \$7

ตัวอย่างที่ 2.3 ตัวอย่างปัญหาการมอบหมายงานแบบสมดุลของปัญหาค่าสูงสุด

บริษัทขายยาแห่งหนึ่งมี 4 สาขา ในแต่ละเมือง ได้แก่ เมือง A เมือง B เมือง C เมือง D มีผู้จัดการสาขา 4 คน แต่ละคนจะได้รับแต่งตั้งในแต่ละเมือง ได้แก่ ผู้จัดการคนที่ 1 ผู้จัดการคนที่ 2 ผู้จัดการคนที่ 3 ผู้จัดการคนที่ 4 ธุรกิจรายเดือนขึ้นอยู่กับเมืองและประสิทธิภาพของผู้จัดการสาขาในเมืองนั้น โดยค่าธุรกิจรายเดือนแสดงในตารางที่ 2.1.5 ผู้จัดการคนใดควรได้รับการมอบหมายงานไปที่เมืองใด เพื่อให้ได้ธุรกิจสูงสุดต่อเดือนโดยรวม

ผู้จัดการสาขา	เมือง			
	A	B	C	D
	Monthly Business (USD)			
ผู้จัดการคนที่ 1	\$11	\$11	\$9	\$9
ผู้จัดการคนที่ 2	\$13	\$16	\$11	\$10
ผู้จัดการคนที่ 3	\$12	\$17	\$13	\$8
ผู้จัดการคนที่ 4	\$16	\$14	\$16	\$12

ตารางที่ 2.1.5 ตารางแสดงค่าธุรกิจสูงสุดต่อเดือน

ขั้นตอนแรก ทำการปรับตารางให้เป็นปัญหาค่าต่ำสุดโดยเลือกค่าสูงสุดในตารางเป็นตัวตั้งค่าใหม่แต่ละช่องจะได้จากค่าสูงสุดลบด้วยค่าที่อยู่ในช่องนั้น จากนั้นทำการดำเนินการตามขั้นตอนของปัญหาค่าต่ำสุด เนื่องจาก 17 คือค่าที่สูงที่สุดในตาราง จึงนำไปลบกับทุกค่าในตาราง จะได้ดังต่อไปนี้

ผู้จัดการสาขา	เมือง			
	A	B	C	D
	Monthly Business (USD)			
ผู้จัดการคนที่ 1	\$6	\$6	\$8	\$8
ผู้จัดการคนที่ 2	\$4	\$1	\$6	\$7
ผู้จัดการคนที่ 3	\$5	\$0	\$4	\$9
ผู้จัดการคนที่ 4	\$1	\$3	\$1	\$5

หลังจากทำการปรับค่าในตารางค่าสูงสุดให้เป็นค่าต่ำสุดแล้ว จะสามารถทำการดำเนินการตามขั้นตอนของปัญหาค่าต่ำสุดได้ จะได้ตารางผลลัพธ์ดังต่อไปนี้

ผู้จัดการสาขา	เมือง			
	A	B	C	D
	Monthly Business (USD)			
ผู้จัดการคนที่ 1	\$11	\$11	\$9	\$9
ผู้จัดการคนที่ 2	\$13	\$16	\$11	\$10
ผู้จัดการคนที่ 3	\$12	\$17	\$13	\$8
ผู้จัดการคนที่ 4	\$16	\$14	\$16	\$12

ตารางที่ 2.1.6 ตารางแสดงผลลัพธ์การมอบหมายงานให้ได้ค่าธุรกิจสูงสุดต่อเดือน

บริษัทควรมอบหมายงาน ดังต่อไปนี้

ผู้จัดการคนที่ 1 ประจำเมือง D จะได้ ธุรกิจรายเดือน เท่ากับ \$9

ผู้จัดการคนที่ 2 ประจำเมือง A จะได้ ธุรกิจรายเดือน เท่ากับ \$13

ผู้จัดการคนที่ 3 ประจำเมือง B จะได้ ธุรกิจรายเดือน เท่ากับ \$17

ผู้จัดการคนที่ 4 ประจำเมือง C จะได้ ธุรกิจรายเดือน เท่ากับ \$16

ดังนั้น ธุรกิจรายเดือนสูงที่สุด เท่ากับ \$55

ตัวอย่างที่ 2.4 การแก้ปัญหาของปัญหาการมอบหมายงานที่มีเส้นทางต้องห้าม

บริษัทผลิตคอมพิวเตอร์แห่งหนึ่งมีโรงงานการผลิตสินค้า 5 แห่ง โดยบริษัทต้องการขนส่งเครื่องคอมพิวเตอร์จากโรงงานผลิตสินค้าทั้งหมด 5 แห่ง ได้แก่ โรงงานผลิตสินค้า A , B , C , D และ E ไปยังเมืองต่างๆ 5 เมือง ได้แก่ นิวยอร์ก ลอนบีช ชิคาโก แจ็คสันวิลล์ และฮิวสตัน แต่ทางบริษัทไม่ต้องการขนส่งสินค้าจากโรงงานผลิตสินค้า D ไปยังเมืองชิคาโก ต้นทุนการขนส่งสินค้าจากโรงงานผลิตสินค้าไปยังเมืองต่างๆแสดงในตารางที่ 2.1.7 เนื่องจากเส้นทางการขนส่งนั้นมีปัญหาด้านการจราจรติดขัด บริษัทควรจัดการการขนส่งอย่างไรเพื่อใช้ต้นทุนในการขนส่งรวมต่ำที่สุด

โรงงานผลิต สินค้า	ต้นทุนต่อหน่วยในการขนส่งสินค้าในแต่ละเมือง				
	นิวยอร์ก	ลอนบีช	ชิคาโก	แจ็คสันวิลล์	ฮิวสตัน
A	\$11	\$17	\$8	\$16	\$20
B	\$9	\$7	\$12	\$6	\$15
C	\$13	\$16	\$15	\$12	\$16
D	\$21	\$24	-	\$28	\$26
E	\$14	\$10	\$12	\$11	\$13

ตารางที่ 2.1.7 ตารางแสดงต้นทุนการขนส่งสินค้าจากโรงงานผลิตสินค้าไปยังเมืองต่างๆ

เนื่องจากกำหนดให้ไม่มีการขนส่งสินค้าจากโรงงานผลิตสินค้า D ไปยังเมืองชิคาโก ดังนั้นจะต้องกำหนดให้เส้นทางต้องห้ามนั้นๆ ให้มีค่าขนส่งต่อหน่วยที่สูงมาก เมื่อเทียบกับค่าขนส่งต่อหน่วยทั้งหมด เราจึงสมมติค่าของเส้นทางต้องห้ามนี้คือ นำค่าที่มากที่สุดในการมอบหมายงาน คูณ 100

จากตัวอย่าง ต้นทุนการขนส่งที่มากที่สุดขนส่งจากโรงงานผลิตสินค้า D ไปยังแจ็คสันวิลล์ คือ \$28 ดังนั้น การขนส่งสินค้าจากโรงงานผลิตสินค้า D ไปยังเมืองชิคาโก จึงมีค่าเป็น \$2800 แสดงในตารางต่อไปนี้

โรงงานผลิต สินค้า	ต้นทุนต่อหน่วยในการขนส่งสินค้าในแต่ละเมือง				
	นิวยอร์ก	ลอนบีช	ชิคาโก	แจ็คสันวิลล์	ฮิวสตัน
A	\$11	\$17	\$8	\$16	\$20
B	\$9	\$7	\$12	\$6	\$15
C	\$13	\$16	\$15	\$12	\$16
D	\$21	\$24	\$2800	\$28	\$26
E	\$14	\$10	\$12	\$11	\$13

ผลของการขนส่งจากโรงงานผลิตสินค้าทั้ง 5 แห่ง ไปยังเมือง 5 เมือง ได้ดังต่อไปนี้

โรงงานผลิต สินค้า	ต้นทุนต่อหน่วยในการขนส่งสินค้าในแต่ละเมือง				
	นิวยอร์ก	ลองบีช	ชิคาโก	แจ๊คสันวิลล์	ฮิวสตัน
A	\$11	\$17	\$8	\$16	\$20
B	\$9	\$7	\$12	\$6	\$15
C	\$13	\$16	\$15	\$12	\$16
D	\$21	\$24	-	\$28	\$26
E	\$14	\$10	\$12	\$11	\$13

ตารางที่ 2.1.8 ตารางแสดงผลลัพธ์การขนส่งสินค้าจากโรงงานผลิตสินค้าไปยังเมืองต่างๆ

บริษัทผลิตคอมพิวเตอร์ควรกำหนดเส้นทางการขนส่ง ดังต่อไปนี้

- โรงงานผลิตสินค้า A ส่งสินค้าไปที่เมือง ชิคาโก ต้นทุนการขนส่ง เท่ากับ \$8
- โรงงานผลิตสินค้า B ส่งสินค้าไปที่เมือง ลองบีช ต้นทุนการขนส่ง เท่ากับ \$7
- โรงงานผลิตสินค้า C ส่งสินค้าไปที่เมือง แจ๊คสันวิลล์ ต้นทุนการขนส่ง เท่ากับ \$12
- โรงงานผลิตสินค้า D ส่งสินค้าไปที่เมือง นิวยอร์ก ต้นทุนการขนส่ง เท่ากับ \$21
- โรงงานผลิตสินค้า E ส่งสินค้าไปที่เมือง ฮิวสตัน ต้นทุนการขนส่ง เท่ากับ \$13

ดังนั้น ต้นทุนการขนส่งสินค้ารวมจากโรงงานผลิตสินค้าไปยังเมืองต่างๆ เท่ากับ \$61

2.1.3. ภาษาไพทอน (Python Programming)



ภาษาโปรแกรม Python คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลชุดคำสั่งที่เราเขียนให้เป็นภาษา Python มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่เราต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA เป็นต้น ภาษาไพทอน นั้นถูกพัฒนาขึ้นมาโดยมีความตั้งใจว่าจะให้เป็นภาษาที่อ่านง่าย มันถูกออกแบบมาให้มีโครงสร้างที่มองเห็นได้โดยไม่ซับซ้อน โดยมักจะใช้คำในภาษาอังกฤษในขณะที่ภาษาอื่นใช้เครื่องหมายวรรคตอน นอกจากนี้ ไพทอนมีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าภาษาซี

Python Interpreter นั้นเป็นตัวแปรภาษาของภาษาไพทอน เพื่อให้สามารถรันโค้ดไพทอนได้ ซึ่งได้มากับไลบรารีมาตรฐานที่สามารถใช้งานได้ฟรี เป็นโปรแกรมแบบ source และ binary สำหรับแพลตฟอร์มที่ได้รับความนิยม นอกจากนี้ Interpreter ยังสนับสนุนการเขียนโปรแกรมกับ Interactive shell ซึ่งเป็นการเขียนโค้ดของภาษาไพทอน ลงไปและเห็นผลลัพธ์การทำงานของคำสั่งได้ในทันที Python Interpreter นั้นยังสามารถนำเพิ่มความสามารถกับฟังก์ชันใหม่ที่ถูกพัฒนามาจากภาษา C และ C++ ไพทอนนั้นเหมาะสำหรับเป็นภาษาในการสร้างแอปพลิเคชันที่ปรับแต่งได้ ไพทอนเวอร์ชันแรก คือ เวอร์ชัน 0.9.0 ออกมาเมื่อปี พ.ศ. 2533 และเวอร์ชันปัจจุบันคือ 3.7

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 การพัฒนาและประยุกต์ใช้ระบบการพยากรณ์ความต้องการสินค้าและการจัดสรรงาน : กรณีศึกษาโรงงานผลิชิ้นส่วนโทรศัพท์เคลื่อนที่

สภาพเศรษฐกิจในปัจจุบันที่มีสภาวะการแข่งขันสูง เพื่อองค์กรยังสามารถดำเนินกิจการต่อไปได้ จึงมีระบบการจัดการต่างๆ เพื่อเพิ่มมูลค่าของสินค้าหรือลดต้นทุนของกระบวนการ เพื่อให้ผลประกอบการของบริษัทยังคงมีผลกำไรของส่วนต่างระหว่างต้นทุนและยอดขาย จึงมีการนำระบบต่างๆ มาประยุกต์ใช้ โดยระบบการวางแผนทรัพยากรประยุกต์ใช้เพื่อลดต้นทุนในการผลิตในส่วนของการจัดสรรทรัพยากรวัตถุดิบและระบบการมอบหมายเป็นการลดต้นทุนในการผลิตในส่วนของการจัดสรรทรัพยากรบุคคลกับงานนั้นๆ ในโรงงานมีต้นทุนที่ต้องใช้ทางด้านวัตถุดิบและพนักงานจำนวนมาก ทำให้ต้นทุนเรื่องวัสดุคงคลังและค่าแรงสูงมากตาม การเลือกวิธีการพยากรณ์ที่เหมาะสมและการมอบหมายงานกับคนในปริมาณวัตถุดิบปริมาณมาก พนักงานจำนวนมากและกระบวนการที่มากตามชนิดสินค้า

วิธียังกาเรียนอัลกอริทึม โดย รศ. บุญสม ศิริโสภณา (2553) อธิบายว่ายังกาเรียนอัลกอริทึมเป็นวิธีสำคัญและเหมาะสมกับสภาพปัญหาการกำหนดงานแบบหนึ่งต่อหนึ่ง (One – to – one Assignment) ที่ดีที่สุดวิธีหนึ่ง การวิเคราะห์ข้อมูลมีดังต่อไปนี้

หมายเลขพนักงาน	กระบวนการ (วินาที)									
	ขั้นรูปพลาสติก	ตัดชิ้นงาน	ประกอบชิ้นงาน 1	ประกอบชิ้นงาน 2	ตรวจชิ้นงาน 1	ตรวจชิ้นงาน 2	ตรวจชิ้นงาน 3	ตรวจชิ้นงาน 4	ตรวจชิ้นงาน 5	ตรวจชิ้นงาน 6
D10001	10	8	22	22	55	55	55	55	55	55
D10002	9	7	17	17	50	50	50	50	50	50
D10003	12	8	18	18	62	62	62	62	62	62
D10004	13	9	19	19	48	48	48	48	48	48
D10005	12	10	22	22	67	67	67	67	67	67
D10006	10	7	24	24	58	58	58	58	58	58
D10007	9	8	20	20	52	52	52	52	52	52
D10008	10	7	19	19	62	62	62	62	62	62
D10009	12	9	24	24	59	59	59	59	59	59
D10010	12	11	23	23	49	49	49	49	49	49

ตารางที่ 2.2.1 ตารางการมอบหมายงานที่ได้จากการใช้โปรแกรมเชิงเส้น

จากตารางข้างต้นจะได้ D10005 ทำกระบวนการขึ้นรูปพลาสติก ที่ 12 วินาที D10006 ทำกระบวนการตัดชิ้นงาน ที่ 7 วินาที D10003 และ D10008 ทำกระบวนการประกอบชิ้นงานที่ ที่ 18 และ 19 วินาที ตามลำดับ D10001 D10002 D10004 D10007 D10009 และ D10010 ทำกระบวนการตรวจชิ้นงาน ที่ 55, 50, 48, 52, 59 และ 49 วินาที ตามลำดับ โดยใช้เวลารวมในการทำงานทั้งหมด 369 วินาที

การมอบหมายงานโดยใช้โปรแกรมเชิงเส้น เพื่อหาระยะเวลารวมของการทำงานให้น้อยที่สุด เพื่อลดต้นทุนรวมของกระบวนการผลิต โดยการเก็บข้อมูลของสายการผลิตตัวอย่างเป็นระยะเวลาหนึ่งและเปรียบเทียบผลโดยทางทฤษฎีและเปรียบเทียบผลก่อนและหลังการมอบหมายงานโดยใช้โปรแกรมเชิงเส้น

	ก่อนใช้โปรแกรมเชิงเส้น	หลังใช้โปรแกรมเชิงเส้น
เวลารวม (วินาที)	401	369

ตารางที่ 2.2.2 ตารางการเปรียบเทียบรอบเวลาการทำงานก่อนและหลังการมอบหมายงาน

จากข้อมูลตารางข้างต้น พบว่า หลังจากการมอบหมายงานโดยใช้โปรแกรมเชิงเส้นแล้ว ผลต่างของเวลาที่ลดลงที่ 31 วินาที เทียบเป็น 92.0 % ของเวลาก่อนใช้โปรแกรมเชิงเส้นทำให้ผลต่าง 8 % ของรอบเวลาการทำงานลดลง

หมายเลขพนักงาน	กระบวนการ (วินาที)									
	ขึ้นรูปพลาสติก	ตัดชิ้นงาน	ประกอบชิ้นงาน 1	ประกอบชิ้นงาน 2	ตรวจชิ้นงาน 1	ตรวจชิ้นงาน 2	ตรวจชิ้นงาน 3	ตรวจชิ้นงาน 4	ตรวจชิ้นงาน 5	ตรวจชิ้นงาน 6
D10001	10	8	22	22	55	55	55	55	55	55
D10002	9	7	17	17	50	50	50	50	50	50
D10003	12	8	18	18	62	62	62	62	62	62
D10004	13	9	19	19	48	48	48	48	48	48
D10005	12	10	22	22	67	67	67	67	67	67
D10006	10	7	24	24	58	58	58	58	58	58
D10007	9	8	20	20	52	52	52	52	52	52
D10008	10	7	19	19	62	62	62	62	62	62
D10009	12	9	24	24	59	59	59	59	59	59
D10010	12	11	23	23	49	49	49	49	49	49

ตารางที่ 2.2.3 ตารางแสดงผลลัพธ์ของการมอบหมายงานที่ได้จากการใช้โปรแกรมเชิงเส้น

2.2.2 แอปพลิเคชันฮังการีเรียนในการคัดเลือกทีมเบสบอลและการมอบหมาย (Application of the Hungarian Algorithm in Baseball Team Selection and Assignment)

เพื่อพัฒนาระบบที่ใช้ประเมินความสามารถของผู้เล่นเบสบอลในทุกแง่มุมของการแข่งขันและสร้างทีมที่ผู้เล่นทุกคนถูกกำหนดบทบาทให้เหมาะสมกับทักษะการเล่นเบสบอล วิธีที่ใช้ในการแก้ปัญหาการมอบหมาย คือฮังการีเรียนอัลกอริทึม ซึ่งรู้จักกันในชื่ออัลกอริทึม Kuhn-Munkres เป็นอัลกอริทึมที่มีประสิทธิภาพช่วยให้มั่นใจในการได้รับมอบหมายบทบาทที่ดีที่สุดของผู้เล่น การวิเคราะห์ความไวจะทำในส่วนของแอปพลิเคชัน มีการพัฒนาระบบที่ใช้ประเมินระดับทักษะของผู้เล่นเกี่ยวกับแง่มุมต่างๆ ของเบสบอลและไม่เพียงเลือกทีมที่ดีที่สุดสำหรับกลยุทธ์ที่กำหนดไว้ล่วงหน้า แต่ยังกำหนดให้เหมาะสมกับบทบาทในทีม

บทที่ 3

วิธีดำเนินงานวิจัย

ในส่วนนี้จะนำเสนอการดำเนินงานวิจัย โดยจะอธิบายขั้นตอนวิธีการทำงานของอัลกอริทึมสำหรับปัญหาการมอบหมายงานสำหรับค่าต่ำสุด ปัญหาการมอบหมายงานสำหรับค่าสูงสุด ปัญหาการมอบหมายที่มีเส้นทางต้องห้าม และปัญหาการมอบหมายงานแบบไม่สมดุล

3.1 การดำเนินงานของโปรแกรมสำหรับการมอบหมายงานในแต่ละรูปแบบ

การทำงานของโปรแกรมสำหรับการมอบหมายงานในแต่ละรูปแบบจะมีวิธีการที่แตกต่างกันเล็กน้อย ดังต่อไปนี้

3.1.1 การดำเนินงานของโปรแกรมปัญหาการมอบหมายงานสำหรับค่าต่ำสุด

เริ่มต้นการทำงานของโปรแกรมด้วยการเช็คเมทริกซ์ที่รับจากหน้าจอจากผู้ใช้งานที่ทำการกรอกข้อมูล จากนั้นโปรแกรมจะทำการเช็คข้อมูลที่อยู่ในเมทริกซ์เป็นอันดับแรก ถ้ามีข้อมูลเพียงหนึ่งข้อมูล โปรแกรมจะทำการแสดงข้อมูลการมอบหมายงานออกจากหน้าจอแสดงผล แต่ถ้าข้อมูลที่อยู่ในเมทริกซ์มีค่ามากกว่าหนึ่งข้อมูล โปรแกรมจะทำการเช็คข้อมูลนั้นว่าเป็นเมทริกซ์จัตุรัส (Square Matrix) หรือไม่ โดยโปรแกรมจะทำการเพิ่มศูนย์ตามจำนวนผลต่างระหว่างแถวและคอลัมน์ เพื่อปรับให้เป็นเมทริกซ์จัตุรัส จากนั้นโปรแกรมจะดำเนินในขั้นตอนต่อไป

```
Matrix_Size_Check (Matrix)
```

```
input MatrixA
```

```
if Only One Element in the Matrix :
```

```
    print ( "Solution Matrix is " + MatrixA )
```

```
else if ColumnSize > RowSize :
```

```
    Add_Row_Zero = ColumnSize - RowSize
```

```
else if ColumnSize < RowSize :
```

```
    Add_Column_Zero = RowSize - ColumnSize
```

```
else ColumnSize == RowSize :
```

```
    continue
```

รูปที่ 3.1.1.1 การดำเนินการตรวจสอบและปรับขนาดของเมทริกซ์

เมื่อเช็คว่าเป็นเมทริกซ์จัตุรัสแล้ว โปรแกรมจะทำการหาค่าที่ต่ำที่สุดในแต่ละแถว และนำค่าที่ได้ไปลบออกจากทุกค่าที่อยู่ในแถวนั้นๆ จากนั้นโปรแกรมจะทำการหาค่าที่ต่ำที่สุดในแต่ละคอลัมน์ และนำค่าที่ได้ไปลบออกจากทุกค่าที่อยู่ในคอลัมน์นั้นๆ

```

Find_Min_Value (Matrix):
for Row[ i ] in MatrixA:
    minValue = The Minimum Value in Row[ i ]
    for All Element[ i,j ] in This Row[ i ] :
        Element[ i,j ] = Element[ i,j ] - minValue
    end for All Element[ i,j ] in This Row[ i ]
return MatrixA
end for Row[ i ] in MatrixA

for Column[ j ] in MatrixA:
    minValue = The Minimum Value in Column[ j ]
    for All Element[ i,j ] in This Column[ j ] :
        Element[ i,j ] = Element[ i,j ] - minValue
    end for All Element[ i,j ] in This Column[ j ]
return MatrixA
end for Column[ j ] in MatrixA

```

รูปที่ 3.1.1.2 ดำเนินการนำค่าต่ำสุดในแต่ละแถวและคอลัมน์
ไปลบทุกค่าที่อยู่ในแต่ละแถวและคอลัมน์

ต่อมาโปรแกรมจะทำการนับจำนวนเลขศูนย์ในแต่ละแถวและคอลัมน์ จากนั้นเลือกแถวหรือคอลัมน์ที่เลขศูนย์มากที่สุดเป็นอันดับแรก โดยเลขศูนย์ที่ถูกเลือกในรอบแรกนั้นจะไม่ถูกเลือกในรอบต่อไป จากนั้นทำการเลือกแถวหรือคอลัมน์ต่อมาที่มีเลขศูนย์มากที่สุด เมื่อทำการเลือกเลขศูนย์จนครบทุกตัวให้ดำเนินการในอันดับต่อไป

```
Find_Cover (Matrix):
```

```
Find_Row_Cover (Matrix):
```

```
rowNum = None
```

```
for Row[ i ] in MatrixA:
```

```
    count = 0
```

```
    buff = [ ]
```

```
    for Element[ i,j ] in Row[ i ]:
```

```
        if Element is 0:
```

```
            Count Increased by 1
```

```
    end for Element[ i,j ] in Row[ i ]
```

```
    Add Count into buff
```

```
end for Row[ i ] in MatrixA
```

```
maxValue = The Maximum Value in the buff
```

```
rowNum = The Row Index of the MaxValue
```

```
return rowNum
```

```
Find_Column_Cover (Matrix):
```

```
Matrix = Transpose Previous Matrix
```

```
columnNum = None
```

```
columnNum = The Return Value of Taking Transposed Matrix into the
```

```
Find_Row_Cover Function
```

```
return columnNum
```

รูปที่ 3.1.1.3 การดำเนินการนับจำนวนเลขศูนย์ในแต่ละแถวและคอลัมน์
เลือกแถวหรือคอลัมน์

หลังจากทำการเลือกเลขศูนย์ในแถวและคอลัมน์จนครบทุกตัว โปรแกรมจะทำการเช็คจำนวนแถวและคอลัมน์ที่เลือกนั้นมีค่าเท่ากับจำนวนแถวของเมทริกซ์หรือไม่ ถ้าจำนวนแถวที่เลือกนั้นมีค่าไม่เท่ากับจำนวนแถวของเมทริกซ์ โปรแกรมจะทำการนำค่าที่ต่ำที่สุดที่ไม่ถูกเลือกในเมทริกซ์ไปลบกับทุกค่าที่ไม่ได้อยู่บนแถวหรือคอลัมน์ที่เลือก จากนั้นนำค่าที่ต่ำที่สุดนี้ไปบวกกับค่าที่ถูกเลือกระหว่างแถวและคอลัมน์ ถ้าจำนวนแถวและคอลัมน์ที่เลือกนั้นมีค่าเท่ากับจำนวนแถวในเมทริกซ์ให้ดำเนินในขั้นตอนต่อไป

```
Check_NumberOfCover (Matrix):
```

```
for Matrix_Cover in MatrixA:
```

```
  if Matrix_Cover = Number of Row:
```

```
    continue
```

```
  else Matrix_Cover != Number of Row:
```

```
    MinValue = The Minimum Element is not in the Cover
```

```
    Element = Every Element is not on the Cover
```

```
    Intersection_Element = Every Element is in the Intersection Cover
```

```
    Element = Element - MinValue
```

```
    Intersection_Element = Intersection_Element + MinValue
```

```
  return The Value of Matrix into the Find_Cover (Matrix)
```

```
  end else Matrix_Cover != Number of Row:
```

```
return Matrix
```

```
end for Matrix_Cover in MatrixA
```

รูปที่ 3.1.1.4 การดำเนินการเช็คจำนวนแถวและคอลัมน์ที่เลือก และเปรียบเทียบกับจำนวนแถวของเมทริกซ์

โปรแกรมจะทำการเลือกเลขศูนย์ที่อยู่บนเมทริกซ์ โดยเลือกเลขศูนย์ที่อยู่บนแถวที่มีเลขศูนย์เพียงหนึ่งตัวเป็นอันดับแรก จากนั้นจะทำการเลือกเลขศูนย์ที่เหลือโดยแถวและคอลัมน์ที่ถูกเลือกแล้วนั้นจะไม่ถูกเลือกซ้ำในรอบต่อไป โปรแกรมจะทำซ้ำจนกว่าจะเลือกเลขศูนย์ครบทุกแถวและคอลัมน์โดยไม่มี การเลือกซ้ำกันในแต่ละแถวและคอลัมน์ จากนั้นโปรแกรมจะแสดงคำตอบออกทางหน้าจอแสดงผล

Pick_Items (Matrix):

Solution = a 2 – D Array Which Contains (The Amount of Rows) Sub Array

rowCondition = a 2 – D Array Which Contains (The Amount of Rows) [False]

Condition Sub Array

while Still Have Zero in a MatrixA:

targetRowIndex = Find The Row Having The Minimum Zero(s)

targetRow = Matrix [targetRowIndex]

for i in Target Row:

Find The Index of a 0 Column In The Solution Array,

Using Index as Row Number, Using Value as Column Number

Mark The Condition of This Row as [True] (Checked)

Change All The Value in This Row to be None

return Solution Array

end while Still Have Zero in a MatrixA

รูปที่ 3.1.1.5 การดำเนินการเลือกคำตอบที่ดีที่สุด

3.1.2 การดำเนินงานของโปรแกรมปัญหาการมอบหมายงานสำหรับค่าสูงสุด

การดำเนินงานของโปรแกรมปัญหาการมอบหมายงานสำหรับค่าสูงสุดนั้น จะมีขั้นตอนการดำเนินงานเหมือนกับขั้นตอนการดำเนินงานของโปรแกรมปัญหาการมอบหมายงานสำหรับค่าต่ำสุด โปรแกรมจะปรับเมทริกซ์ปัญหาการมอบหมายงานสำหรับค่าสูงสุดให้เป็นเมทริกซ์ปัญหาการมอบหมายงานสำหรับค่าต่ำสุดเป็นอันดับแรก โดยโปรแกรมจะนำค่าที่สูงที่สุดในเมทริกซ์ไปลบออกจากค่าทุกค่าที่อยู่ในเมทริกซ์ จากนั้นโปรแกรมจะดำเนินงานตามขั้นตอนการดำเนินงานของโปรแกรมปัญหาการมอบหมายงานสำหรับค่าต่ำสุด

```

Find_MaxValue (Matrix):
for maxValue in Matrix:
    maxValue = The Maximum Value in Matrix
    for Element in This Matrix :
        Element = Element – maxValue
    end for Element in This Matrix
return Matrix
end for maxValue in Matrix

```

รูปที่ 3.1.2.1 การดำเนินการหาค่าที่มากที่สุด และนำไปลบออกจากทุกค่าในเมทริกซ์

3.1.3 การดำเนินงานของโปรแกรมปัญหาการมอบหมายงานสำหรับเส้นทางต้องห้าม

การดำเนินงานของโปรแกรมปัญหาการมอบหมายงานสำหรับเส้นทางต้องห้าม โปรแกรมจะทำการรับค่า “ - ” จากหน้าจอแสดงผล โปรแกรมจะทำการนำค่าสูงสุดในเมทริกซ์คูณ 100 เพื่อปรับให้เส้นทางนั้นๆ มีค่ากว่าค่าอื่นๆ ในเมทริกซ์ จากนั้นโปรแกรมจะดำเนินงานตามขั้นตอนการดำเนินงานของโปรแกรมปัญหาการมอบหมายงานสำหรับค่าต่ำสุด

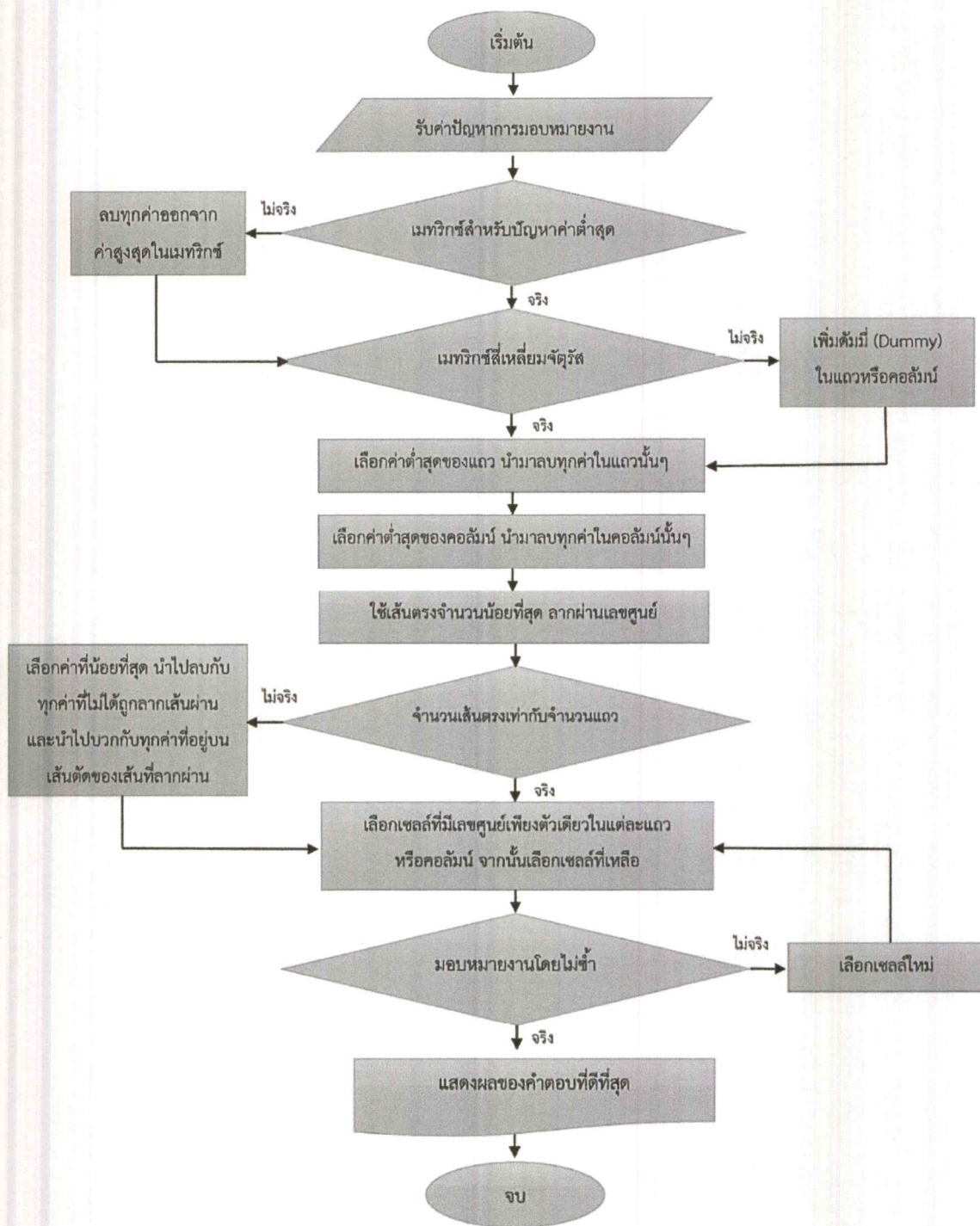
```

Prohibited_Routes (Matrix):
for ProhibitedRoutes in Matrix:
    ProhibitedRoutes == “ - “
    for Element in This Matrix :
        Element = maxValue * 100
    end for Element in This Matrix
return Matrix
end for ProhibitedRoutes in Matrix

```

รูปที่ 3.1.3.1 การดำเนินการกำหนดเส้นทางต้องห้ามให้มีค่ามากที่สุดในเมทริกซ์

3.2 แผนภาพสรุปขั้นตอนในการทำงานของฮังกาเรียน



ภาพที่ 3.2.1 แผนภาพแสดงการทำงานของฮังกาเรียน

บทที่ 4

ผลการดำเนินงาน

ในบทนี้จะอธิบายวิธีการทำงานของโปรแกรมฮังการีเรียนเพื่อแก้ปัญหาการมอบหมายงานทุกรูปแบบที่โปรแกรมสามารถดำเนินการแก้ปัญหาได้ อธิบายวิธีการใช้งานโปรแกรม รวมไปถึงอธิบายผลลัพธ์ที่ได้จากการประมวลผลของโปรแกรมฮังการีเรียน

4.1 วิธีการใช้งานโปรแกรมฮังการีเรียนเพื่อแก้ปัญหาการมอบหมายงาน

โปรแกรมฮังการีเรียนเพื่อแก้ปัญหาการมอบหมายงานนี้ ถูกออกแบบเพื่อให้ใช้งานได้ง่าย สะดวกต่อการใช้งานจริง

The screenshot shows a software interface titled "Hungarian Program". The main heading is "Data for the Assignment Problem". Below this, there are four input fields: "Row Name" and "Column Name" are text boxes, while "Number of Rows" and "Number of Columns" are dropdown menus. There are two radio buttons labeled "MINIMIZE" and "MAXIMIZE". To the right of these is a text box labeled "Objective Value". At the bottom of the interface, there are two buttons: "SUBMIT" and "QUIT".

รูปที่ 4.1.1 โปรแกรมฮังการีเรียนเพื่อแก้ปัญหาการมอบหมายงาน

โปรแกรมฮังการีเรียนเพื่อแก้ปัญหาการมอบหมายงาน มีวิธีการใช้งานโปรแกรมดังต่อไปนี้

จากข้อมูลของบริษัทขนส่งแห่งหนึ่งต้องการวางแผนการจัดการสินค้าระหว่างศูนย์กระจายสินค้า (Factory) จำนวน 8 แห่ง ให้กับลูกค้า (Customer) จำนวน 8 ราย โดยต้นทุนการขนส่งระหว่างศูนย์กระจายสินค้าไปยังลูกค้าแต่ละรายแสดงได้ดังตารางที่ 4.1.1 บริษัทต้องการทราบต้นทุนรวมที่ต่ำที่สุดของการขนส่งระหว่างศูนย์กระจายสินค้ากับลูกค้า เพื่อประกอบการตัดสินใจในการขนส่งสินค้า

Customer Factory	Cus1	Cus2	Cus3	Cus4	Cus5	Cus6	Cus7	Cus8
Fac1	\$13	\$13	\$24	\$33	\$20	\$50	\$48	\$37
Fac2	\$46	\$24	\$28	\$47	\$50	\$21	\$31	\$43
Fac3	\$11	\$15	\$13	\$35	\$41	\$47	\$22	\$15
Fac4	\$15	\$36	\$36	\$38	\$14	\$21	\$50	\$50
Fac5	\$41	\$12	\$26	\$50	\$11	\$10	\$41	\$33
Fac6	\$49	\$21	\$48	\$48	\$35	\$14	\$38	\$10
Fac7	\$36	\$27	\$24	\$33	\$17	\$17	\$19	\$24
Fac8	\$49	\$29	\$48	\$14	\$15	\$23	\$47	\$50

ตารางที่ 4.1.1 ตารางแสดงต้นทุนระหว่างศูนย์กระจายสินค้า 8 แห่ง ไปยังลูกค้า 8 ราย

จากข้อมูลในตารางที่ 4.1.1 กำหนดให้ชื่อแถว (Row Name) คือ ศูนย์กระจายสินค้า (Factory) จำนวนแถว (Number of Row) เท่ากับ 8 แถว จากนั้นทำการกำหนดให้ชื่อคอลัมน์ (Column Name) คือ ลูกค้า (Customer) จำนวนคอลัมน์ (Number of Column) เท่ากับ 8 คอลัมน์ จากนั้นผู้ใช้งานสามารถกำหนดว่าข้อมูลของผู้ใช้งานนั้นเป็นปัญหาค่าสูงสุด หรือปัญหาค่าต่ำสุด จากข้อมูลข้างต้นผู้ใช้งานต้องการทราบต้นทุนรวมของการขนส่งระหว่างศูนย์กระจายสินค้าซึ่งเป็นปัญหาค่าต่ำสุด ดังนั้นจึงทำการกำหนดค่าวัตถุประสงค์ (Objective Value) เป็นค่าต่ำสุด (Minimize)

The screenshot shows a window titled "Hungarian Program" with a subtitle "Data for the Assignment Problem". The interface includes the following fields and controls:

- Row Name:** A text box containing "Factory".
- Number of Rows:** A numeric input field containing "8".
- Column Name:** A text box containing "Customer".
- Number of Columns:** A numeric input field containing "8".
- Objective Value:** A dropdown menu currently set to "MINIMIZE".
- Buttons:** "SUBMIT" and "QUIT" buttons are located at the bottom.

รูปที่ 4.1.2 กำหนดความต้องการลงในโปรแกรมฮังการี

เมื่อทำการกดตกลง (Submit) โปรแกรมจะแสดงตารางตามจำนวนแถวและคอลัมน์ของข้อมูลที่ผู้ใช้งานกำหนดเพื่อให้ผู้ใช้งานทำการกรอกข้อมูลลงบนตาราง

Data Input	Customer1	Customer2	Customer3	Customer4	Customer5	Customer6	Customer7	Customer8
Factory1								
Factory2								
Factory3								
Factory4								
Factory5								
Factory6								
Factory7								
Factory8								

รูปที่ 4.1.3 ตารางสำหรับกรอกข้อมูลในโปรแกรมฮังการี

Data Input	Customer1	Customer2	Customer3	Customer4	Customer5	Customer6	Customer7	Customer8
Factory1	13	13	24	33	20	50	48	37
Factory2	46	24	28	47	50	21	31	43
Factory3	11	15	13	35	41	47	22	15
Factory4	15	36	36	38	14	21	50	50
Factory5	41	12	26	50	11	10	41	33
Factory6	49	21	48	48	35	14	38	10
Factory7	36	27	24	33	17	17	19	24
Factory8	49	29	48	14	15	23	47	50

รูปที่ 4.1.4 ตารางแสดงข้อมูลที่ผู้ใช้งานกรอกในโปรแกรมฮังการี

หลังจากที่ผู้ใช้งานกรอกข้อมูลครบทุกช่อง ให้ผู้ใช้งานทำการกดตกลง (Submit) โปรแกรมจะแสดงผลการมอบหมายงานออกมาในรูปแบบตาราง และแสดงผลลัพธ์รวมของค่าเป้าหมาย (Optimal Value) ออกทางหน้าจอแสดงผล

ผลลัพธ์ของปัญหาการมอบหมายงานเพื่อวางแผนการจัดการสินค้าระหว่างศูนย์กระจายสินค้าจำนวน 8 แห่ง ให้กับลูกค้าจำนวน 8 รายที่ได้จากโปรแกรมฮังกาเรียน มีดังต่อไปนี้

Hungarian Program								
Data Input	Customer1	Customer2	Customer3	Customer4	Customer5	Customer6	Customer7	Customer8
Factory1	13	*	*	*	*	*	*	*
Factory2	*	*	*	*	*	21	*	*
Factory3	*	*	13	*	*	*	*	*
Factory4	*	*	*	*	14	*	*	*
Factory5	*	12	*	*	*	*	*	*
Factory6	*	*	*	*	*	*	*	10
Factory7	*	*	*	*	*	*	19	*
Factory8	*	*	*	14	*	*	*	*

Optimal Value: 116

BACK

รูปที่ 4.1.5 ตารางแสดงผลลัพธ์ในโปรแกรมฮังกาเรียน

- ศูนย์กระจายสินค้าที่ 1 ส่งสินค้าไปยังลูกค้าคนที่ 1 ต้นทุนการขนส่งเท่ากับ \$13
 ศูนย์กระจายสินค้าที่ 2 ส่งสินค้าไปยังลูกค้าคนที่ 6 ต้นทุนการขนส่งเท่ากับ \$21
 ศูนย์กระจายสินค้าที่ 3 ส่งสินค้าไปยังลูกค้าคนที่ 3 ต้นทุนการขนส่งเท่ากับ \$13
 ศูนย์กระจายสินค้าที่ 4 ส่งสินค้าไปยังลูกค้าคนที่ 5 ต้นทุนการขนส่งเท่ากับ \$14
 ศูนย์กระจายสินค้าที่ 5 ส่งสินค้าไปยังลูกค้าคนที่ 2 ต้นทุนการขนส่งเท่ากับ \$12
 ศูนย์กระจายสินค้าที่ 6 ส่งสินค้าไปยังลูกค้าคนที่ 8 ต้นทุนการขนส่งเท่ากับ \$10
 ศูนย์กระจายสินค้าที่ 7 ส่งสินค้าไปยังลูกค้าคนที่ 7 ต้นทุนการขนส่งเท่ากับ \$19
 ศูนย์กระจายสินค้าที่ 8 ส่งสินค้าไปยังลูกค้าคนที่ 4 ต้นทุนการขนส่งเท่ากับ \$14

สรุป ต้นทุนรวมที่ต่ำที่สุดของการขนส่งระหว่างศูนย์กระจายสินค้ากับลูกค้ามีค่าเท่ากับ \$116

สำหรับปัญหาการมอบหมายงานที่มีเส้นทางต้องห้าม ผู้ใช้งานจะต้องทำการกรอกเครื่องหมาย “ - “ เพื่อป้องกันผู้ใช้งานไม่ต้องการมอบหมายงานไปในเส้นทางนั้นๆ

จากข้อมูลของบริษัทขนส่ง เนื่องจากเส้นทางการขนส่งสินค้าจากศูนย์กระจายสินค้าที่ 2 ไปยังลูกค้าคนที่ 6 มีสภาพการจราจรติดขัดอย่างหนัก ซึ่งทำให้ต้องใช้เวลาอย่างมากในการขนส่งสินค้าทางบริษัทจึงกำหนดให้ไม่มีการขนส่งสินค้าจากศูนย์กระจายสินค้าที่ 2 ไปยังลูกค้าคนที่ 6 เพื่อหลีกเลี่ยงสภาพการจราจรที่ติดขัดและประหยัดเวลาในการขนส่งสินค้า ดังรูปที่ 4.1.6

Hungarian Program								
Data Input	Customer1	Customer2	Customer3	Customer4	Customer5	Customer6	Customer7	Customer8
Factory1	13	13	24	33	20	50	48	37
Factory2	46	24	28	47	50	-	31	43
Factory3	11	15	13	35	41	47	22	15
Factory4	15	36	36	38	14	21	50	50
Factory5	41	12	26	50	11	10	41	33
Factory6	49	21	48	48	35	14	38	10
Factory7	36	27	24	33	17	17	19	24
Factory8	49	29	48	14	15	23	47	50

SUBMIT
BACK

รูปที่ 4.1.6 ตารางแสดงข้อมูลเส้นทางต้องห้ามที่ผู้ใช้งานกรอกในโปรแกรมฮังการี

ผลลัพธ์ของปัญหาการมอบหมายงานเพื่อวางแผนการจัดการสินค้าระหว่างศูนย์กระจายสินค้าจำนวน 8 แห่ง ให้กับลูกค้าจำนวน 8 ราย โดยไม่มีการขนส่งสินค้าจากศูนย์กระจายสินค้าที่ 2 ไปยังลูกค้าคนที่ 6 ที่ได้จากโปรแกรมฮังการี มีดังต่อไปนี้

Hungarian Program								
Data Input	Customer1	Customer2	Customer3	Customer4	Customer5	Customer6	Customer7	Customer8
Factory1	13	*	*	*	*	*	*	*
Factory2	*	24	*	*	*	*	*	*
Factory3	*	*	13	*	*	*	*	*
Factory4	*	*	*	*	14	*	*	*
Factory5	*	*	*	*	*	10	*	*
Factory6	*	*	*	*	*	*	*	10
Factory7	*	*	*	*	*	*	19	*
Factory8	*	*	*	14	*	*	*	*

Optimal Value: 117

BACK

รูปที่ 4.1.7 ตารางแสดงผลลัพธ์ในโปรแกรมฮังการี

- ศูนย์กระจายสินค้าที่ 1 ส่งสินค้าไปยังลูกค้าคนที่ 1 ต้นทุนการขนส่งเท่ากับ \$13
- ศูนย์กระจายสินค้าที่ 2 ส่งสินค้าไปยังลูกค้าคนที่ 2 ต้นทุนการขนส่งเท่ากับ \$24
- ศูนย์กระจายสินค้าที่ 3 ส่งสินค้าไปยังลูกค้าคนที่ 3 ต้นทุนการขนส่งเท่ากับ \$13
- ศูนย์กระจายสินค้าที่ 4 ส่งสินค้าไปยังลูกค้าคนที่ 5 ต้นทุนการขนส่งเท่ากับ \$14
- ศูนย์กระจายสินค้าที่ 5 ส่งสินค้าไปยังลูกค้าคนที่ 6 ต้นทุนการขนส่งเท่ากับ \$10
- ศูนย์กระจายสินค้าที่ 6 ส่งสินค้าไปยังลูกค้าคนที่ 8 ต้นทุนการขนส่งเท่ากับ \$10
- ศูนย์กระจายสินค้าที่ 7 ส่งสินค้าไปยังลูกค้าคนที่ 7 ต้นทุนการขนส่งเท่ากับ \$19
- ศูนย์กระจายสินค้าที่ 8 ส่งสินค้าไปยังลูกค้าคนที่ 4 ต้นทุนการขนส่งเท่ากับ \$14

สรุป ต้นทุนรวมที่ต่ำที่สุดของการขนส่งระหว่างศูนย์กระจายสินค้ากับลูกค้าโดยไม่มีการขนส่งสินค้าจากศูนย์กระจายสินค้าที่ 2 ไปยังลูกค้าคนที่ 6 มีค่าเท่ากับ \$117

สำหรับปัญหาค่าสูงสุด ผู้ใช้งานจะต้องกำหนดค่าเป้าหมายรวม (Objective Value) เป็นค่าสูงสุด (Maximize)

จากข้อมูลของสายการบินแห่งหนึ่ง ต้องการนำเครื่องบิน (Plane) 8 ลำ เข้าจอดที่สนามบินคุนหมิง (Kunming International Airport) ในประเทศจีน โดยจะต้องเข้าเทียบประตูขึ้นเครื่อง (Boarding Gate) 7 ประตู ค่าประสิทธิผลจากการเข้าจอดที่สนามบิน แสดงในตารางที่ 4.1.2 ทางสายการบินต้องการทราบว่าควรจะกำหนดให้เครื่องบินแต่ละลำเข้าเทียบประตูขึ้นเครื่องใด เพื่อให้ค่าประสิทธิผลรวมสูงสุด โดยเครื่องบินแต่ละลำสามารถเข้าเทียบประตูขึ้นเครื่องได้เพียงประตูเดียวเท่านั้น และประตูขึ้นเครื่องแต่ละประตูสามารถมีเครื่องบินเข้าเทียบได้เพียงหนึ่งลำเท่านั้น

Gate \ Plane	Gate1	Gate2	Gate3	Gate4	Gate5	Gate6	Gate7
TG 1	\$5000	\$3200	\$2500	\$4000	\$9500	\$2700	\$3000
TG 2	\$7000	\$9700	\$15000	\$10000	\$69000	\$9900	\$7800
TG 3	\$12000	\$12000	\$9400	\$20000	\$10000	\$19000	\$23000
TG 4	\$5500	\$3000	\$4000	\$7000	\$28000	\$7500	\$9200
TG 5	\$2000	\$6900	\$2500	\$5500	\$8800	\$6700	\$10000
TG 6	\$15000	\$9900	\$25000	\$22000	\$11000	\$21000	\$40000
TG 7	\$80000	\$10000	\$11000	\$32000	\$19000	\$25000	\$20000
TG 8	\$70000	\$3300	\$200	\$2500	\$30000	\$6600	\$8000

ตารางที่ 4.1.2 ตารางแสดงค่าประสิทธิผลการเข้าจอดของเครื่องบิน 8 ลำ
เทียบประตูขึ้นเครื่อง 7 ประตู

จากข้อมูลในตารางที่ 4.1.2 กำหนดให้ชื่อแถว (Row Name) คือ เครื่องบิน (Plane) จำนวนแถว (Number of Row) เท่ากับ 8 แถว จากนั้นทำการกำหนดให้ชื่อคอลัมน์ (Column Name) คือ ประตูเครื่องบิน (Boarding Gate) จำนวนคอลัมน์ (Number of Column) เท่ากับ 7 คอลัมน์ จากนั้นผู้ใช้งานสามารถกำหนดว่าข้อมูลของผู้ใช้งานนั้นเป็นปัญหาค่าสูงสุด หรือปัญหาค่าต่ำสุด จากข้อมูลข้างต้นผู้ใช้งานต้องการทราบค่าประสิทธิผลรวมของการเข้าจอดที่สนามบิน ซึ่งเป็นปัญหาค่าสูงสุด ดังนั้นจึงทำการกำหนดค่าเป้าหมายรวม (Objective Value) เป็นค่าสูงสุด (Maximize)

Hungarian Program

Data for the Assignment Problem

Row Name : Number of Rows :

Column Name : Number of Columns :

MINIMIZE MAXIMIZE Objective Value

รูปที่ 4.1.8 กำหนดความต้องการลงในโปรแกรมฮังการี

เมื่อทำการกดตกลง (Submit) โปรแกรมจะแสดงตารางตามจำนวนแถวและคอลัมน์ของข้อมูล
 ผู้ใช้งานกำหนดเพื่อให้ผู้ใช้งานทำการกรอกข้อมูลลงบนตาราง

Hungarian Program

Data Input	Gate1	Gate2	Gate3	Gate4	Gate5	Gate6	Gate7
TG1	5000	3200	2500	4000	9500	2700	3000
TG2	7000	9700	15000	10000	69000	9900	7800
TG3	12000	12000	9400	20000	10000	19000	23000
TG4	5500	3000	4000	7000	28000	7500	9200
TG5	2000	9600	2500	5500	8800	6700	10000
TG6	15000	9900	25000	22000	11000	21000	40000
TG7	80000	10000	11000	32000	19000	25000	20000
TG8	70000	3300	200	2500	30000	6600	8000

รูปที่ 4.1.9 ตารางสำหรับกรอกข้อมูลในโปรแกรมฮังการี

หลังจากที่ผู้ใช้งานกรอกข้อมูลครบทุกช่อง ให้ผู้ใช้งานทำการกดตกลง (Submit) โปรแกรมจะแสดงผลการมอบหมายงานออกมาในรูปแบบตาราง และแสดงผลลัพธ์รวมของค่าเป้าหมาย (Optimal Value) ออกทางหน้าจอแสดงผล

Data Input	Gate1	Gate2	Gate3	Gate4	Gate5	Gate6	Gate7	Gate8
TG1	*	*	*	*	*	*	*	0
TG2	*	*	*	*	69000	*	*	*
TG3	*	*	94000	*	*	*	*	*
TG4	*	*	*	*	*	7500	*	*
TG5	*	6900	*	*	*	*	*	*
TG6	*	*	*	*	*	*	40000	*
TG7	*	*	*	32000	*	*	*	*
TG8	70000	*	*	*	*	*	*	*

Optimal Value: 319400

BACK

รูปที่ 4.1.10 ตารางแสดงข้อมูลที่ผู้ใช้งานกรอกในโปรแกรมยังกาเรียน

ผลลัพธ์ของปัญหาการมอบหมายงานเพื่อวางแผนการจัดการนำเครื่องบิน 8 ลำ เข้าเทียบประตูขึ้นเครื่อง 7 ประตู ที่ได้จากโปรแกรมยังกาเรียน มีดังต่อไปนี้

เครื่องบินลำที่ TG 1	ไม่มีการเข้าเทียบประตูขึ้นเครื่องใดๆ	ค่าประสิทธิผล เท่ากับ \$0
เครื่องบินลำที่ TG 2	เข้าเทียบประตูขึ้นเครื่องที่ 5	ค่าประสิทธิผล เท่ากับ \$69,000
เครื่องบินลำที่ TG 3	เข้าเทียบประตูขึ้นเครื่องที่ 3	ค่าประสิทธิผล เท่ากับ \$94,000
เครื่องบินลำที่ TG 4	เข้าเทียบประตูขึ้นเครื่องที่ 6	ค่าประสิทธิผล เท่ากับ \$7,500
เครื่องบินลำที่ TG 5	เข้าเทียบประตูขึ้นเครื่องที่ 2	ค่าประสิทธิผล เท่ากับ \$6,900
เครื่องบินลำที่ TG 6	เข้าเทียบประตูขึ้นเครื่องที่ 7	ค่าประสิทธิผล เท่ากับ \$40,000
เครื่องบินลำที่ TG 7	เข้าเทียบประตูขึ้นเครื่องที่ 4	ค่าประสิทธิผล เท่ากับ \$32,000
เครื่องบินลำที่ TG 8	เข้าเทียบประตูขึ้นเครื่องที่ 1	ค่าประสิทธิผล เท่ากับ \$70,000

สรุป ค่าประสิทธิผลรวมของการเข้าจอดที่สนามบินของเครื่องบิน 8 ลำ เทียบประตูขึ้นเครื่อง 7 ประตู มีค่าเท่ากับ \$319,400

นอกจากนี้ถ้าผู้ใช้งานต้องการกรอกข้อมูลการมอบหมายงานใหม่ ให้ทำการกดปุ่มย้อนกลับ (Back) เพื่อทำการกรอกข้อมูลการมอบหมายงานใหม่ที่ผู้ใช้งานต้องการ

บทที่ 5

สรุปผลการดำเนินงาน

โครงการพิเศษนี้ได้ศึกษาปัญหาการมอบหมายงานและวิธีการแก้ปัญหาการมอบหมายงานในทุกรูปแบบ วางแผนจัดการการมอบหมายงานได้อย่างเหมาะสมและใช้ประโยชน์ได้อย่างมากที่สุด รวมทั้งเพื่อตอบสนองความต้องการลดต้นทุนและเพิ่มกำไรของการมอบหมายงานของบริษัท โดยมี การนำแนวคิดทางคณิตศาสตร์มาประยุกต์ใช้กับความรู้ทางด้านโปรแกรมคอมพิวเตอร์เพื่อใช้ในการแก้ปัญหา ซึ่งผลจากการแก้ปัญหาการวางแผนการมอบหมายงานนั้น สามารถสรุปได้ดังต่อไปนี้

5.1 สรุปผลการดำเนินงาน

ผลลัพธ์จากการนำอัลกอริทึมฮังการี (The Hungarian Algorithm) มาสร้างโปรแกรมสำหรับการแก้ปัญหาการมอบหมายงานโดยใช้ภาษาไพทอน (Python Programming) นั้น สามารถที่จะนำโปรแกรมนี้ไปใช้งานได้จริงอย่างมีประสิทธิภาพ โดยสามารถนำไปใช้กับปัญหาการมอบหมายงานได้ทุกรูปแบบ ไม่ว่าจะเป็นปัญหาการขนส่ง (The Transportation Problem) ปัญหาการผลิต (Manufacturing) ปัญหาทางด้านโลจิสติกส์ (Logistics) รวมไปถึงปัญหาการจัดการทรัพยากรมนุษย์ (Human Resource) สามารถเลือกฟังก์ชันการใช้งานได้สองรูปแบบ นั่นคือสามารถเลือกกว่าปัญหาของผู้ใช้งานนั้นเป็นปัญหาการมอบหมายงานแบบปัญหาค่าต่ำสุดที่ต้องการวางแผนการลงทุนอย่างไรให้ใช้ต้นทุนรวมการลงทุนที่น้อยที่สุด หรือปัญหาการมอบหมายงานแบบปัญหาค่าสูงสุดที่ต้องการวางแผนเพื่อให้ได้ผลกำไรรวมมากที่สุดให้กับทางบริษัท นอกจากนี้ยังสามารถกำหนดเส้นทางต้องห้ามให้กับเส้นทางที่ผู้ใช้งานไม่ต้องการมอบหมายงานไปในเส้นทางนั้นๆ สามารถช่วยลดระยะเวลาการทำงานให้กับทางบริษัท

การนำโปรแกรมมาช่วยในการตัดสินใจในการมอบหมายงาน พบว่าการมอบหมายงานที่มีความเหมาะสม มีต้นทุนการมอบหมายงานรวมที่ต่ำที่สุดหรือมีกำไรมากที่สุดนั้น อาจมีปัจจัยอื่นๆ เข้ามาเกี่ยวข้องทำให้ส่งผลต่อต้นทุนรวม ดังนั้นโปรแกรมนี้จึงเป็นทางเลือกที่ช่วยในการพิจารณาการมอบหมายงานเท่านั้น

โปรแกรมนี้เหมาะสำหรับผู้ประกอบการที่ไม่มีความรู้ในด้านของอัลกอริทึมแต่ต้องการเครื่องมือที่ช่วยในการประกอบการตัดสินใจ สำหรับตัวโปรแกรมผู้ใช้งานสามารถใช้งานได้อย่างสะดวกสบาย รวดเร็ว และแม่นยำ อย่างไรก็ตามโปรแกรมที่พัฒนาขึ้นมานี้เป็นเพียงตัวแบบเริ่มต้นที่สามารถนำไปพัฒนาโปรแกรมให้มีประสิทธิภาพมากยิ่งขึ้นในอนาคต

5.2 ข้อเสนอแนะ

จากการสร้างโปรแกรมสำหรับการแก้ปัญหาการมอบหมายงานโดยวิธีฮังกาเรียนนี้ อาจมีปัจจัยที่สามารถปรับปรุงหรือพัฒนาให้โปรแกรมมีประสิทธิภาพมากขึ้น ได้ดังต่อไปนี้

1. สามารถแสดงการมอบหมายงานที่มีผลลัพธ์เท่ากัน แต่มีการมอบหมายงานแตกต่างกัน ที่สามารถเป็นไปได้ทั้งหมด
2. สามารถรับข้อมูลการมอบหมายงานจริงจากโปรแกรมเอกเซล (Microsoft Excel) หรือโปรแกรมอื่นๆ ของทางบริษัทผู้ใช้งาน เข้ามาคำนวณในโปรแกรม
3. พัฒนาโปรแกรมให้ทันสมัยและง่ายต่อการใช้งานมากยิ่งขึ้น

เอกสารอ้างอิง

- [1] Chartchai Leenawong. (2017). The Assignment Problem. Operation Research. 4(170 – 200), Bangkok
- [2] MUNKRES. (1957). J. Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics 5, 32–38.
- [3] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2), 83-97.
- [4] Konig, D.: Uber graphen und ihre anwendung auf determinantentheorie und mengenlehre. Mathematische Annalen 77(4) (1916) 453-465
- [5] Britz, S. S., & Maltitz, M. (2011). Application of the Hungarian Algorithm in Baseball team selection and Assignment. Mathematical Statistics and Actuarial Science, University of the Free State.
- [6] Alvarado, C., Morrison, B. B., Ericson, B., Guzdial, M., Miller, B., & Ranum, D. L. (2012). Performance and use evaluation of an electronic book for introductory Python programming.
- [7] Sanner, M. F. (1999). Python: a programming language for software integration and development. J Mol Graph Model, 17(1), 57-61.
- [8] Lutz, M. (2013). Learning python: Powerful object-oriented programming. "O'Reilly Media, Inc."

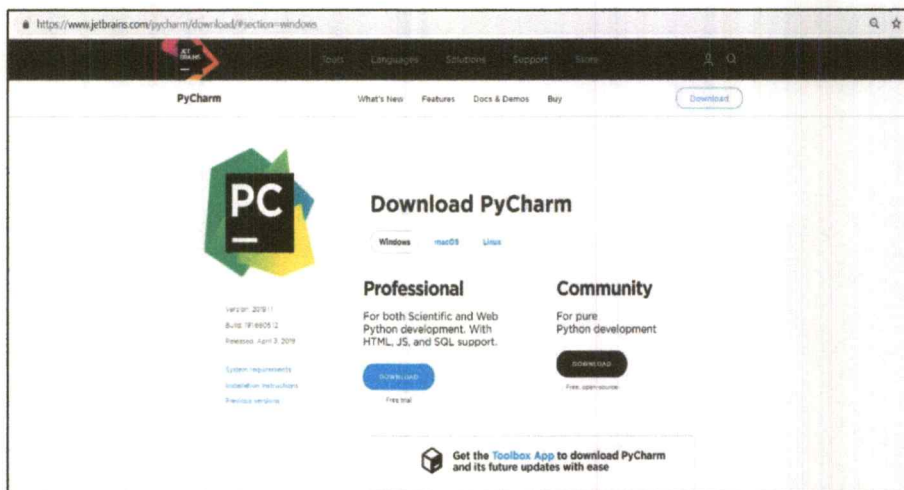
ภาคผนวก ก

ก. 1 ไพชาร์ม (Pycharm)

ไพชาร์ม (Pycharm) คือโปรแกรมที่ใช้สำหรับเขียนภาษาไพทอน เป็นโปรแกรมที่ใช้งานได้ง่าย และสามารถใช้งานได้ฟรี เหมาะสำหรับผู้ที่ต้องการจะฝึกเขียนภาษาไพทอน และนอกจากนี้ยังรองรับระบบปฏิบัติการ Windows , [Linux](#) และ MacOS

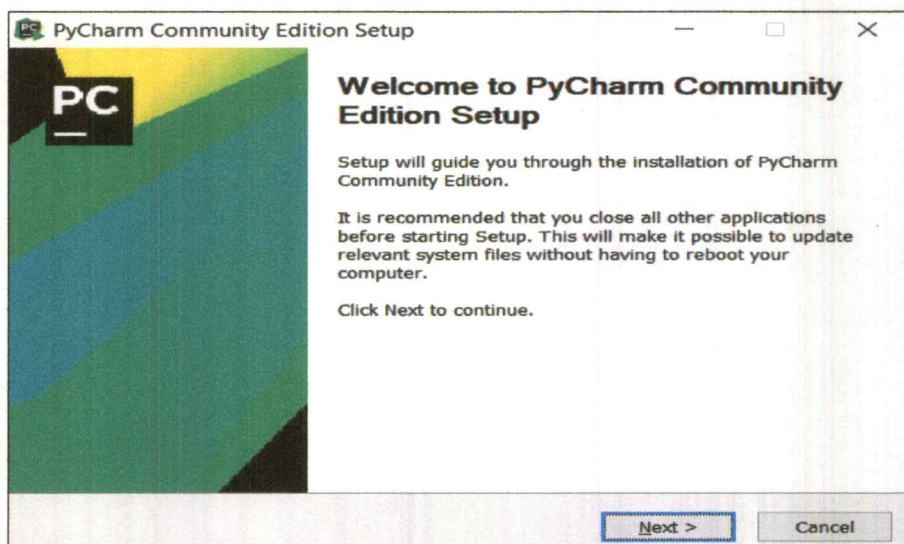
ก.2 ขั้นตอนการติดตั้ง Pycharm

1. ดาวน์โหลดโปรแกรม Pycharm จาก <https://www.jetbrains.com/pycharm/download> โดยเลือกโหมดที่จะดาวน์โหลดเป็น Community



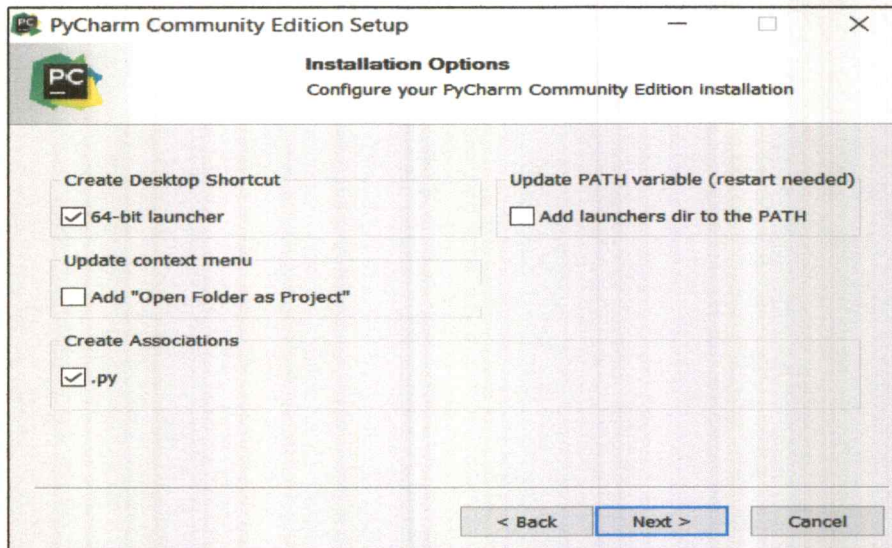
รูปที่ ก. 2.1 เว็บไซต์สำหรับดาวน์โหลดโปรแกรม Pycharm

2. เมื่อทำการดาวน์โหลดโปรแกรมเสร็จเรียบร้อยแล้ว จะมีหน้าต่าง "Pycharm Community Edition Setup" ขึ้นมา ให้ทำการกด "Next" ไปตามขั้นตอนการติดตั้ง



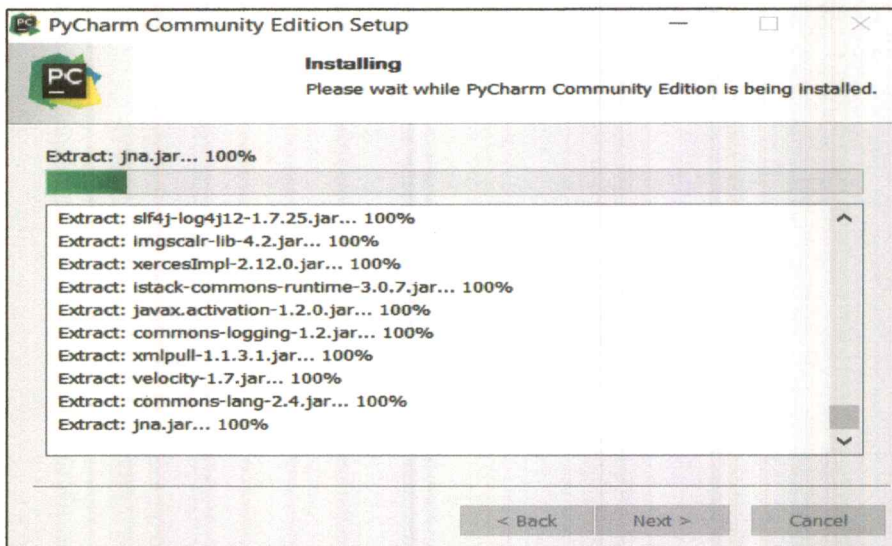
รูปที่ ก. 2.2 หน้าต่าง Pycharm Community Edition Setup

3. เมื่อมาถึงหน้าต่าง "Installation Options" สามารถกำหนดค่าเองได้ว่าผู้ใช้งานต้องการให้มี Shortcut บนหน้าจอเดสก์ท็อปหรือไม่



รูปที่ ก. 2.3 หน้าต่าง Installation Options

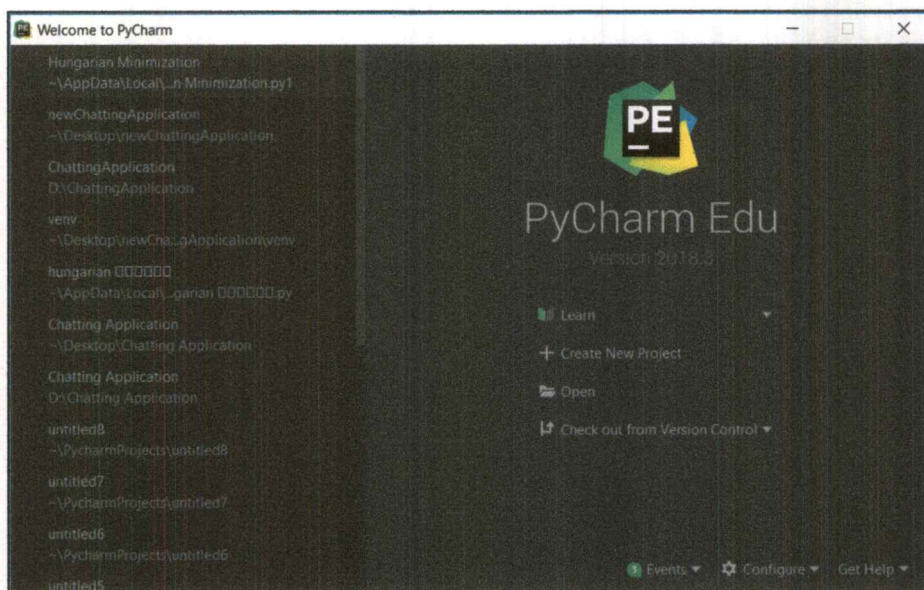
4. ทำการติดตั้ง Pycharm



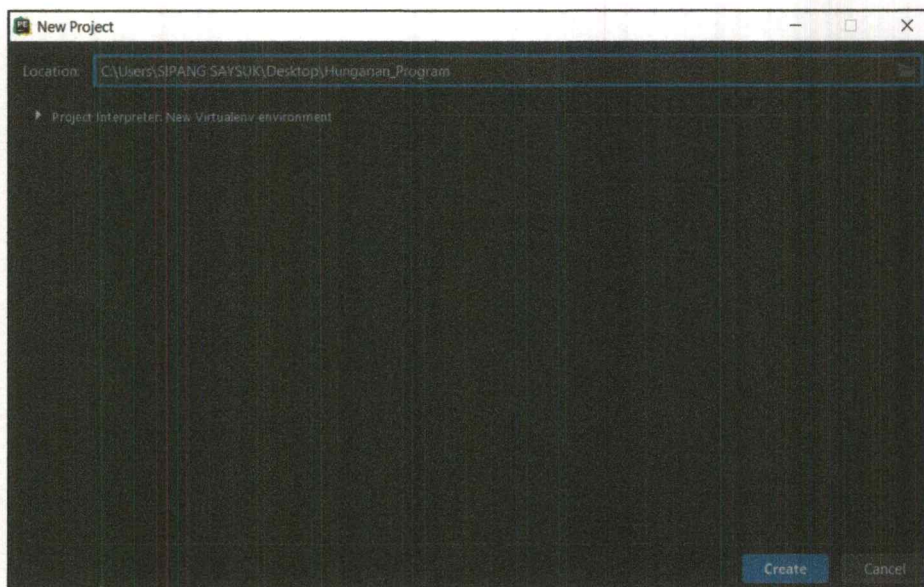
รูปที่ ก. 2.4 หน้าต่าง Installing Pycharm

เมื่อโปรแกรมติดตั้งเสร็จเรียบร้อย ผู้ใช้งานสามารถใช้งานโปรแกรมไพชามได้โดยใช้ภาษาไพทอนในการเขียนโปรแกรม โดยผู้ใช้งานจะต้องทำการสร้างโปรเจกใหม่ (Create New Project) แสดงดังรูปที่ ก.2.5 จากนั้นผู้ใช้งานจะต้องกำหนดโลเคชัน (Location) ของไฟล์ที่ผู้ใช้งานต้องการ แสดงดังรูปที่

ก.2.6

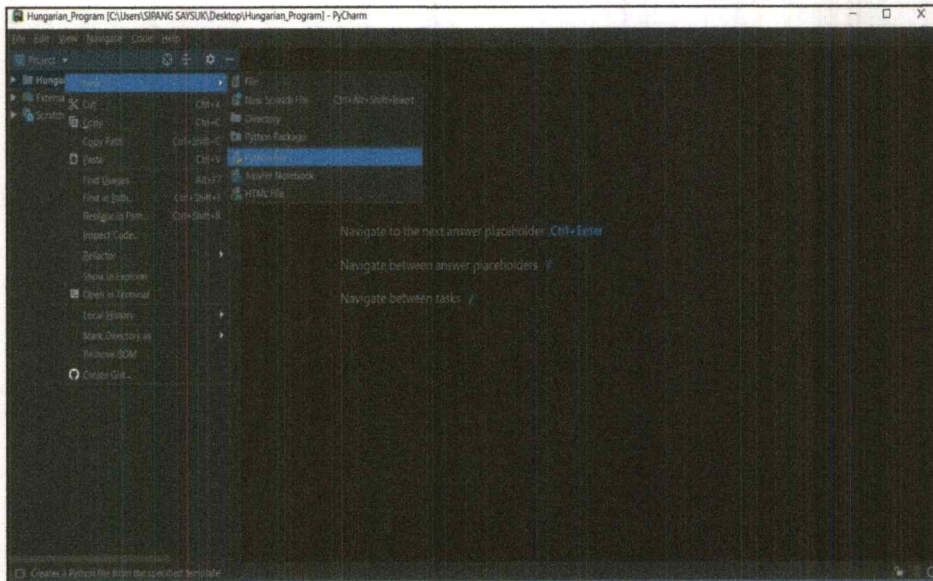


รูปที่ ก. 2.5 หน้าต่าง Create New Project

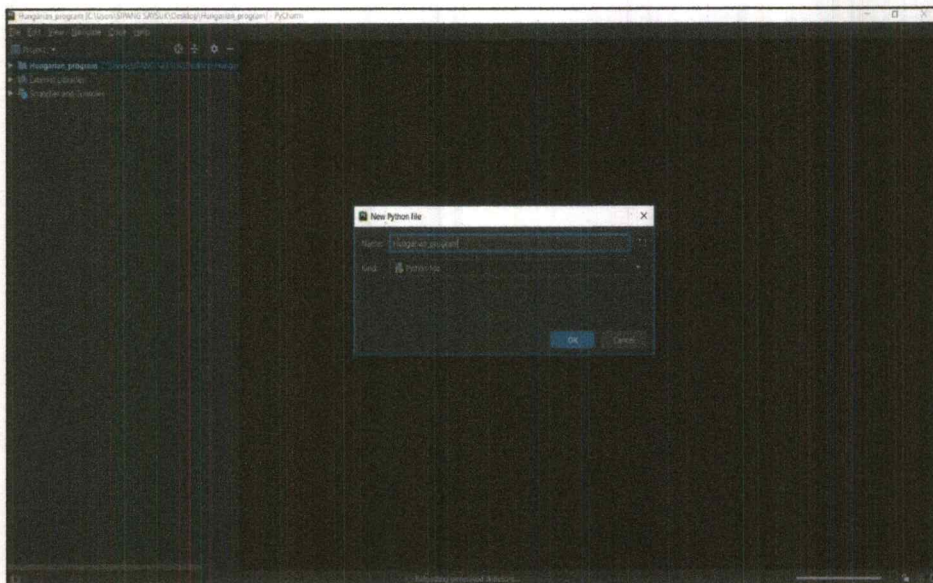


รูปที่ ก. 2.6 หน้าต่างกำหนด Location ของไฟล์

หลังจากที่ผู้ใช้งานทำการสร้างโปรเจกต์ใหม่เรียบร้อยแล้ว ผู้ใช้งานจะต้องสร้างไฟล์ไพทอน (Python File) โดยทำการคลิกซ้ายที่ชื่อไฟล์ของผู้ใช้งาน จากนั้นทำการเลือก New > Python File เพื่อเป็นการสร้างไพทอนไฟล์สำหรับใช้งาน แสดงดังรูปที่ ก. 2.7 และทำการตั้งชื่อไฟล์ไพทอน แสดงดังรูปที่ ก. 2.8



รูปที่ ก. 2.7 หน้าต่างสร้างไฟล์ไพทอน (New Python File)



รูปที่ ก. 2.8 หน้าต่างตั้งชื่อไฟล์ไพทอน (New Python File Name)

