การประยุกต์ใช้โปรแกรมไมโครซอฟท์เอกซ์เซลเพื่อกำหนดระยะทางของคลังสินค้า

โดย

นายตฤณ ปฐมนิธิภิญโญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาการจัดการงานวิศวกรรม แผน ก แบบ ก 2 ปริญญามหาบัณฑิต

ภาควิชาวิศวกรรมอุตสาหการและการจัดการ

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2562

ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

การประยุกต์ใช้โปรแกรมไมโครซอฟท์เอกซ์เซลเพื่อกำหนดระยะทางของคลังสินค้า

โดย
นายตฤณ ปฐมนิธิภิญโญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาการจัดการงานวิศวกรรม แผน ก แบบ ก 2 ปริญญามหาบัณฑิต
ภาควิชาวิศวกรรมอุตสาหการและการจัดการ
บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร
ปีการศึกษา 2562
ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

AN APPLICATION TO DETERMINE WAREHOUSE DISTANCE ON MICROSOFT EXCEL

By

MR. Trin PATHOMNITHIPINYO

A Thesis Submitted in Partial Fulfillment of the Requirements

for Master of Engineering (ENGINEERING MANAGEMENT)

Department of INDUSTRIAL ENGINEERING AND MANAGEMENT

Graduate School, Silpakorn University

Academic Year 2019

Copyright of Graduate School, Silpakorn University

หัวข้อ	การประยุกต์ใช้โปรแกรมไมโครซอฟท์เอกซ์เซลเพื่อกำหนดระยะทางของคลังสินค้า
โดย	ตฤณ ปฐมนิธิภิญโญ
สาขาวิชา	การจัดการงานวิศวกรรม แผน ก แบบ ก 2 ปริญญามหาบัณฑิต
อาจารย์ที่ปรึกษาหลัก	ผู้ช่วยศาสตราจารย์ ดร. คเณศ พันธุ์สวาสดิ์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร ได้รับพิจารณาอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย
(รองศาสตราจารย์ ดร.จุไรรัตน์ นันทานิช)

พิจารณาเห็นชอบโดย

.....ประธานกรรมการ
(รองศาสตราจารย์ ดร.ประจวบ กล่อมจิตร)

.....อาจารย์ที่ปรึกษาหลัก
(ผู้ช่วยศาสตราจารย์ ดร.คเณศ พันธุ์สวาสดิ์)

.....ผู้ทรงคุณวุฒิภายนอก
(ผู้ช่วยศาสตราจารย์ ดร.กนกพร ศรีปฐมสวัสดิ์)

60405204 : การจัดการงานวิศวกรรม แผน ก แบบ ก 2 ปริญญามหาบัณฑิต
คำสำคัญ : คลังสินค้า, โปรแกรมไมโครซอฟท์ เอ็กซ์เซล, วิถีสั้นสุดทุกคู่โหนด, กิจกรรมการจ่ายสินค้า, การไหลในเครือข่าย

นาย ตฤณ ปฐมนิธิภิญโญ: การประยุกต์ใช้โปรแกรมไมโครซอฟท์เอกซ์เซลเพื่อกำหนดระยะทางของคลังสินค้า อาจารย์ที่ปรึกษาวิทยานิพนธ์ : ผู้ช่วยศาสตราจารย์ ดร. คเณศ พันธุ์สวาสดิ์

โดยทั่วไปคลังสินค้าหลาย ๆ แห่ง จะไม่มีข้อมูลระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า งานวิจัยนี้จึงมีแนวคิดในการหาระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า โดยใช้วิธีการร่างรูปคลังสินค้าลงบนโปรแกรมไมโครซอฟท์เอกซ์เซล (Microsoft Excel) หลังจากนั้นโปรแกรมจะเปลี่ยนช่องทางเดินภายในคลังสินค้าให้เป็นโหนด (Node) โดยใช้รูปแบบโครงข่าย (Network Model) ในการหาระยะทางแบบแมนแฮตตันระหว่างโหนดที่อยู่ติดกัน เมื่อได้ระยะทางระหว่างโหนดแล้วจะใช้ขั้นตอนวิธีของฟลอยด์-วอร์แชล (The Floyd-Warshall Algorithm) มาหาระยะทางวิถีสั้นสุดทุกคู่ (All-Pair Shortest Path) โดยผลลัพธ์ของโปรแกรมที่ได้สามารถประยุกต์ใช้ได้กับคลังสินค้าหลายรูปแบบ และเมื่อทดสอบจำนวนโหนดที่มีผลต่อระยะเวลาทำให้ทราบว่าที่จำนวนโหนด 20 50 100 150 200 และ 250 โหนด ใช้เวลาในการทำงานของโปรแกรม 0.02 0.04 0.11 0.23 0.43 และ 1.02 นาที ตามลำดับ

มหาวิทยาลัยศิลปากร

60405204 : Major (ENGINEERING MANAGEMENT)
Keyword : WAREHOUSE DISTANCE, MICROSOFT EXCEL, ALL-PAIR SHORTEST PATH, ORDER PICKING, NETWORK FLOW

MR. TRIN PATHOMNITHIPINYO : AN APPLICATION TO DETERMINE WAREHOUSE DISTANCE ON MICROSOFT EXCEL THESIS ADVISOR : ASSISTANT PROFESSOR KANATE PANSAWAT

In general, many warehouses lack distance information from every position to every position in the warehouse. This research, therefore, has the idea to determine the distance from every position to every position within the warehouse by drawing the warehouse image on Microsoft Excel. After that, the program will transform the aisle inside the warehouse into a node by using the Network Model to find the manhattan distance between the only adjacent nodes. When the distance has been obtained between nodes then the program uses the Floyd-Warshall algorithm to determine All-Pair Shortest Path. The program can be applied to a variety of warehouses. After, testing the number of nodes that affect the duration that at the number of nodes 20, 50, 100, 150 200 and 250, the program takes 0.02 0.04 0.11 0.23 0.43 and 1.02 minutes respectively.

มหาวิทยาลัยศิลปากร

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากผู้วิจัยได้รับความช่วยเหลือและดูแลเอาใจใส่เป็นอย่างดีจากหลายๆ ฝ่าย โดยเฉพาะอาจารย์ที่ปรึกษา ผศ.ดร. คเณศ พันธุ์สวาสดิ์ ในการให้คำแนะนำ ตรวจแก้ไข ให้ข้อเสนอแนะ รวมไปถึงติดตามความก้าวหน้าในการดำเนินงานวิจัย ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาของอาจารย์เป็นอย่างยิ่ง และขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้ ขอขอบคุณ รองศาสตราจารย์ ดร. ประจวบ กล่อมจิตร และ ผู้ช่วยศาสตราจารย์ ดร. กนกพร ศรีปฐมสวัสดิ์ กรรมการสอบปริญญานิพนธ์ ที่ได้สละเวลากรุณาให้ข้อเสนอแนะแก้ไขและให้แนวคิดต่าง ๆ ที่เป็นประโยชน์ ขอขอบคุณพี่พนักงานที่ให้คำปรึกษา และดำเนินการด้านเอกสารต่าง ๆ ที่เกี่ยวข้องกับการวิจัยในครั้งนี้ รวมไปถึงภาควิชาวิศวกรรมอุตสาหการและการจัดการ มหาวิทยาลัยศิลปากร วิทยาเขตพระราชวังสนามจันทร์ที่อนุเคราะห์สถานที่เพื่อทำการวิจัย และสถานที่ในการสอบวิจัย สุดท้ายนี้ ผู้วิจัยขอขอบพระคุณบิดามารดาและครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจผู้วิจัยเสมอมาจนสำเร็จการศึกษา

ตฤณ ปฐมนิธิภิญโญ

มหาวิทยาลัยศิลปากร

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฌ
สารบัญรูปภาพ ..... ญ
บทที่ 1 1
บทนำ 1
1.1 ที่มาและความสำคัญของงานวิจัย 1
1.2 วัตถุประสงค์ของการวิจัย 3
1.3 กรอบแนวคิดของการวิจัย 3
1.4 ขอบเขตของการวิจัย 3
1.5 ประโยชน์ที่คาดว่าจะได้รับ 4
บทที่ 2 5
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5
2.1 ตัวแบบโครงข่าย [5, 6] 5
2.2 ระยะทางแบบแมนแฮตตัน 6
2.3 วิถีสั้นสุดแบบทุกคู่ 6
2.3 งานวิจัยที่เกี่ยวข้อง 8
บทที่ 3 16
วิธีการดำเนินการ 16

3.1 สร้างรูปแบบสำหรับผู้ใช้เพื่อกรอกข้อมูลบนโปรแกรม Microsoft Excel 17
3.2 เปลี่ยนช่องทางเดินให้เป็นโหนด 18
3.3 หาระยะทางระหว่างโหนด 19
3.5 สร้างขั้นตอนสำหรับหาวิถีสั้นสุดทุกคู่โหนด 22
3.6 ขั้นตอนการตรวจสอบความถูกต้องของโปรแกรม 23
บทที่ 4 25
ผลและวิเคราะห์ผลการวิจัย 25
4.1 โปรแกรมคำนวณหาระยะทางเดินภายในคลังสินค้า 25
4.2 ผลการทดสอบโปรแกรม 27
4.3 ผลการตรวจสอบความถูกต้องของโปรแกรม 33
4.4 จำนวนโหนดที่ส่งผลต่อระยะเวลาการทำงานของโปรแกรม 39
บทที่ 5 49
สรุปผลการวิจัย 49
5.1 สรุปผลการวิจัย 49
5.2 ข้อเสนอแนะ 49
รายการอ้างอิง 50
ภาคผนวก ก 52
ผลลัพธ์ของโปรแกรมที่ใช้ทดสอบคลังสินค้า 53
ภาคผนวก ข 63
รูปแบบคลังสินค้าที่ใช้ในการทดสอบระยะเวลาในการทำงานของโปรแกรม 64
ภาคผนวก ค 68
คู่มือการใช้งานโปรแกรม 69
ประวัติผู้เขียน 73

มหาวิทยาลัยศิลปากร

สารบัญตาราง

หน้า

ตารางที่ 1 ตารางตัวอย่างแผนภูมิจากไป 3
ตารางที่ 2 ตารางขอบเขตข้อมูลคลังสินค้าที่ใช้ในการทดสอบเวลาในการทำงานของโปรแกรม 4
ตารางที่ 3 งานวิจัยที่เกี่ยวข้องที่เกี่ยวข้องกับขั้นตอนวิธีของ Floyd-Warshall 14
ตารางที่ 4 ผลการตรวจสอบค่าของระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า 34
ตารางที่ 5 แสดงข้อมูลเวลาในการทำงานของโปรแกรมที่จำนวนโหนดต่างกัน 39
ตารางที่ 6 ระยะทางระหว่างโหนดของคลังสินค้าที่มีรูปแบบสี่เหลี่ยมผืนผ้า 54
ตารางที่ 7 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่มีรูปแบบสี่เหลี่ยมผืนผ้า 54
ตารางที่ 8 ระยะทางระหว่างโหนดของคลังสินค้าที่มีรูปแบบกำแพงด้านหนึ่งเป็นแบบขั้นบันได 56
ตารางที่ 9 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่มีรูปแบบกำแพงด้านหนึ่งเป็นแบบขั้นบันได 56
ตารางที่ 10 ระยะทางระหว่างโหนดของคลังสินค้าที่มีรูปแบบเป็นเครื่องหมายบวก 58
ตารางที่ 11 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่มีรูปแบบเป็นเครื่องหมายบวก 58
ตารางที่ 12 ระยะทางระหว่างโหนดของคลังสินค้าที่มีออฟฟิศ หรือห้องอยู่ตรงกลางคลังสินค้า 60
ตารางที่ 13 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่มีออฟฟิศ หรือห้องอยู่ตรงกลางคลังสินค้า 60
ตารางที่ 14 ระยะทางระหว่างโหนดของคลังสินค้าที่เป็นสี่เหลี่ยมผืนผ้า และมีทางเข้าออกสองด้าน 61
ตารางที่ 15 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่เป็นสี่เหลี่ยมผืนผ้า และมีทางเข้าออกสองด้าน 62
ตารางที่ 16 แสดงขั้นตอนการทำงานของโปรแกรมสำหรับผู้ใช้งาน 69
ตารางที่ 17 ตารางตัวอย่างข้อมูลระยะทางกับชั้นวางสินค้า 71

สารบัญรูปภาพ

หน้า

ภาพที่ 1 เปอร์เซ็นต์ต้นทุนของกิจกรรมภาพในคลังสินค้า [2] 1

ภาพที่ 2 ความหลากหลายของเส้นทางเดินภายในคลังสินค้า 2

ภาพที่ 3 รูปแบบคลังสินค้าจริงที่ร่างลงบนโปรแกรม Microsoft Excel 2

ภาพที่ 4 กรอบแนวคิดงานวิจัย 3

ภาพที่ 5 รูปแบบตัวแบบโครงข่ายแบบกราฟ (a) แบบเส้นเชื่อมไม่มีทิศทาง (b) แบบเส้นเชื่อมมีทิศทาง 5

ภาพที่ 6 รูปแบบการระบุน้ำหนัก หรือระยะทางของรูปแบบโครงข่าย 6

ภาพที่ 7 รูปแบบการเดินแบบแมนแฮตตัน 6

ภาพที่ 8 ขั้นตอนวิธีของ Floyd-Warshall [8] 7

ภาพที่ 9 ตัวอย่างคลังสินค้า 8

ภาพที่ 10 (a) รูปแบบคลังสินค้า (b) เส้นทางเดินภายในคลังสินค้า 9

ภาพที่ 11 ขั้นตอนวิธีของ The Floyd-Warshall ที่ใช้ในการหาระยะทางทุกคู่โหนด ภายในกราฟที่มี negative cycle 10

ภาพที่ 12 Floyd Warshall Algorithm VS Rectangular Algorithm (Probability p = (1) 0.1 (2) 0.3 10

ภาพที่ 13 Floyd Warshall Algorithm VS Rectangular Algorithm (Probability p = (1) 0.5 (2) 0.8 11

ภาพที่ 14 รูปแบบ Dimension Chain 11

ภาพที่ 15 (a) เมทริกซ์สัญลักษณ์ที่ใช้กำหนดรูปทรง และ (b) เมทริกซ์ค่าความคลาดเคลื่อนเชิงตัวเลข 12

ภาพที่ 16 (a) เมทริกซ์ค่าความคลาดเคลื่อนขั้นตอนที่ 1 และ (b) เมทริกซ์ค่าความคลาดเคลื่อนขั้นตอนที่ 2 12

ภาพที่ 17 แสดงการทำงานของแอปพลิเคชันเพื่อแก้ปัญหาการจราจร 13

ภาพที่ 18 ขั้นตอนการดำเนินการ 16
ภาพที่ 19 ตารางข้อมูลที่ต้องกรอกในโปรแกรมจำลองคลังสินค้า 17
ภาพที่ 20 ระยะห่างระหว่างชั้นสินค้าในแนวแกน x และแกน y 17
ภาพที่ 21 รูปแบบคลังสินค้าที่วาดบนโปรแกรม Microsoft Excel 18
ภาพที่ 22 โหนดที่ได้จากการเปลี่ยนช่องทางเดินจากตัวอย่างคลังสินค้าในภาพที่ 3.4 18
ภาพที่ 23 ตัวอย่าง (a) การนับช่องทางเดินของคลังสินค้า (b) ระยะทางเดินระหว่างโหนดที่ได้ 20
ภาพที่ 24 รูปตารางแสดงผลของระยะที่ได้ภายในคลังสินค้า 21
ภาพที่ 25 ตัวเลขของตำแหน่ง หรือโหนดที่กำหนดบนช่องทางเดิน 23
ภาพที่ 26 (a) เส้นเชื่อมและระยะห่างระหว่างโหนด 24
ภาพที่ 27 รูปแบบโครง่ข่ายที่ใช้ในการตรวจสอบโปรแกรม 24
ภาพที่ 28 รูปแบบโปรแกรมคำนวณหาระยะทางภายในคลังสินค้า 25
ภาพที่ 29 รูปแบบแบบฟอร์มที่ใช้ในการกรอกข้อมูล k_x และ k_y 26
ภาพที่ 30 (a) รูปแบบคลังสินค้าจริง (b) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม Microsoft Excel 26
ภาพที่ 31 ภาพปุ่มที่ใช้สำหรับการเริ่มทำงานของโปรแกรม 26
ภาพที่ 32 (a) รูปแบบคลังสินค้าที่เป็นรูปสี่เหลี่ยม (b) รูปแบบคลังสินค้าที่เป็นรูปตัวแอล 27
ภาพที่ 33 รูปแบบแบบฟอร์มที่ใช้ในการกรอกข้อมูล k_x และ k_y 27
ภาพที่ 34 รูปแบบคลังสินค้าที่ร่างลงบนโปรแกรม Microsoft Excel 28
ภาพที่ 35 (a) รูปแบบโหนดที่ได้จากโปรแกรม Microsoft Excel (b) โหนดในรูปแบบโครงข่าย 28
ภาพที่ 36 ตารางระยะห่างระหว่างโหนดของคลังสินค้าที่มีรูปแบบสี่เหลี่ยม 29
ภาพที่ 37 ตารางระยะห่างระหว่างโหนดของคลังสินค้าที่มีรูปแบบสี่เหลี่ยม 29
ภาพที่ 38 รูปแบบคลังสินค้าที่ร่างลงบนโปรแกรม Microsoft Excel 30
ภาพที่ 39 (a) รูปแบบโหนดที่ได้จากโปรแกรม Microsoft Excel (b) แสดงโหนดในรูปแบบโครงข่าย 30

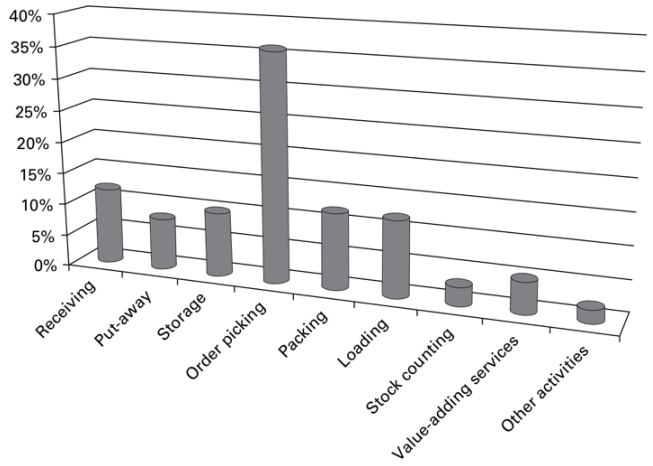
ภาพที่ 40 ตารางระยะห่างระหว่างโหนดของคลังสินค้าที่มีรูปแบบตัวแอล 31
ภาพที่ 41 ตารางระยะห่างระหว่างโหนดของคลังสินค้าที่มีรูปแบบตัวแอล 31
ภาพที่ 42 รูปแบบคลังสินค้าที่ใช้ในการทดสอบโปรแกรม 32
ภาพที่ 43 (a) รูปแบบคลังสินค้าที่ใช้ในการตรวจสอบโปรแกรม (c) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม (c) โหนดที่ได้จากการเปลี่ยนช่องทางเดิน และ (d) ระยะทางระหว่างโหนด 33
ภาพที่ 44 (a) แสดงเส้นทางเดินภายในคลังสินค้า และ 34
ภาพที่ 45 กราฟความสัมพันธ์ระหว่างเวลากับจำนวนโหนด 40
ภาพที่ 46 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ 53
ภาพที่ 47 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม 53
ภาพที่ 48 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ 55
ภาพที่ 49 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม (b) รูปแบบโครงข่ายที่ได้จากโปรแกรม 55
ภาพที่ 50 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ 57
ภาพที่ 51 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม (b) รูปแบบโครงข่ายที่ได้จากโปรแกรม..... 57
ภาพที่ 52 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ 59
ภาพที่ 53 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม (b) รูปแบบโครงข่ายที่ได้จากโปรแกรม..... 59
ภาพที่ 54 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ 61
ภาพที่ 55 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม (b) รูปแบบโครงข่ายที่ได้จากโปรแกรม..... 61
ภาพที่ 56 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 20 โหนด 64
ภาพที่ 57 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 50 โหนด 64
ภาพที่ 58 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 100 โหนด 65
ภาพที่ 59 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 150 โหนด 65
ภาพที่ 60 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 200 โหนด 66
ภาพที่ 61 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 250 โหนด 67
ภาพที่ 62 แสดงแบบฟอร์มสำหรับกรอกค่า k_x และ k_y 70

ภาพที่ 63 แสดงรูปแบบการวัดค่า k_x และ k_y 70
ภาพที่ 64 ภาพคลังสินค้าที่ระบุรหัสสินค้าระหว่างช่องทางเดิน 70
ภาพที่ 65 ภาพคลังสินค้าที่ระบุรหัสสินค้าระหว่างโหนดช่องทางเดิน 71

บทที่ 1
บทนำ

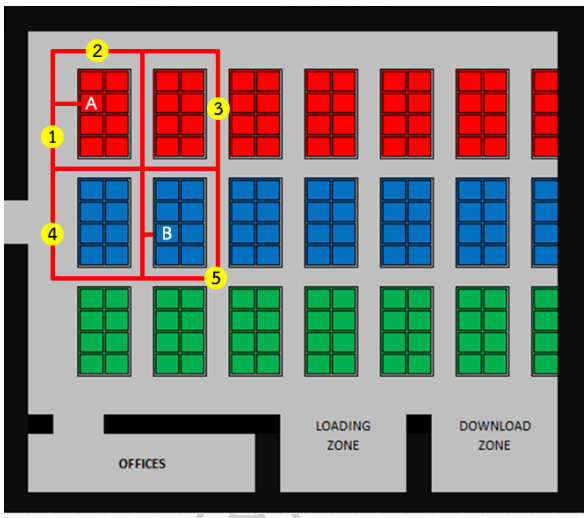
1.1 ที่มาและความสำคัญของงานวิจัย

ปัจจุบันมีการวิจัยเกี่ยวกับการจัดการคลังสินค้าเพิ่มมากขึ้น เพราะมีต้นทุนในการดำเนินการสูงมาก โดยเฉพาะการเลือกหยิบสินค้าตามคำสั่งซื้อ (Order Picking) แสดงดังภาพที่ 1 [1, 2] เป็นกิจกรรมที่มีค่าใช้จ่าย และต้นทุนสูงถึงร้อยละ 55 – 70 เปอร์เซ็นต์ [3] เนื่องจากเป็นกิจกรรมที่ใช้เวลาในการดำเนินการค่อนข้างมาก [4]



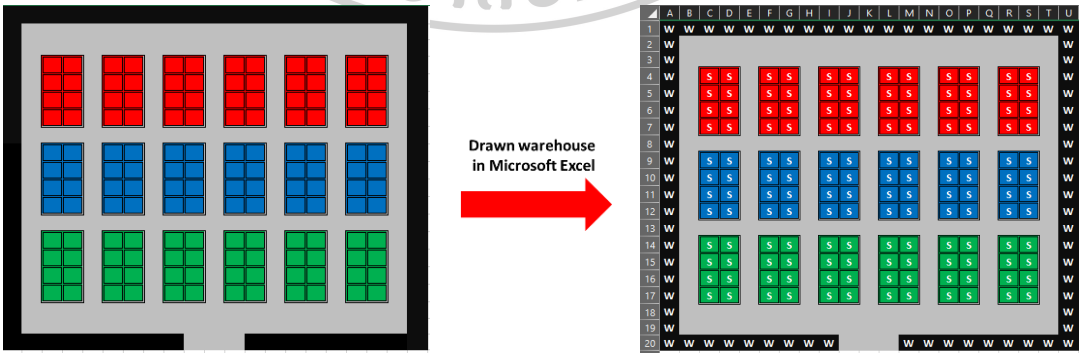
ภาพที่ 1 เปอร์เซ็นต์ต้นทุนของกิจกรรมภาพในคลังสินค้า [2]

ปัญหาที่ส่งผลต่อกิจกรรม Order Picking ก็คือ การหยิบสินค้าโดยไม่คำนึงถึงระยะทางเดินภายในคลังสินค้า ซึ่งธุรกิจที่มีคลังสินค้าส่วนใหญ่จะไม่ทราบระยะทางภายในคลังสินค้า และจะใช้พนักงานในการวัดระยะทางภายในคลังสินค้า เนื่องจากปัญหาการเลือกหยิบสินค้าภายในคลังสินค้าสามารถเดินหยิบได้หลายเส้นทาง แสดงดังภาพที่ 2 เป็นตัวอย่างการเดินจากจุด A ไปยังจุด B พบว่ามีเส้นทางการเดินหลายเส้นทางที่มีระยะทางแตกต่างกัน ซึ่งพนักงานไม่สามารถทราบว่าเส้นทางไหนมีระยะทางสั้นที่สุด และเป็นเรื่องยากที่จะใช้มนุษย์ในการวัดระยะทางภายในคลังสินค้า กรณีที่คลังสินค้ามีขนาดใหญ่นอกจากการวัดระยะทางจะเป็นเรื่องยากแล้วยังสูญเสียเวลาในการดำเนินงานดังกล่าว งานวิจัยส่วนใหญ่จะมุ้งเน้นไปในการพัฒนาขั้นตอนวิธีในการหาเส้นทางที่เหมาะสมที่สุด แต่ในการหาเส้นทางที่เหมาะสมที่สุด จำเป็นต้องทราบระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า เพื่อนำไปใช้ในการหาเส้นทางที่เหมาะสมในการเลือกหยิบสินค้าตามคำสั่งซื้อ ซึ่งยังไม่มีงานวิจัยที่ให้ความสำคัญในการหาระยะทาง จากการใช้รูปแบบของคลังสินค้าจริงในการคำนวณ



ภาพที่ 2 ความหลากหลายของเส้นทางเดินภายในคลังสินค้า

ดังนั้นผู้วิจัยได้พิจารณาและวิเคราะห์ถึงปัญหาดังกล่าว จึงมีแนวคิดในการจัดทำโปรแกรมในการคำนวณหาระยะทางจากทุกจุดไปยังทุกจุด และเพื่อให้ง่ายต่อการทำงานของโปรแกรม ผู้วิจัยจึงมีแนวคิดในการร่างรูปแบบของคลังสินค้าลงบนโปรแกรมไมโครซอฟท์เอ็กซ์เซล (Microsoft Excel) ดังภาพที่ 3 เพื่อคำนวณหาระยะทาง โดยใช้รูปแบบการเดินแบบแมนแฮตตัน (Manhattan Model) เมื่อทำการร่างรูปแบบคลังสินค้าลงบนโปรแกรม Microsoft Excel โปรแกรมจะทำการเปลี่ยนทางเดิน (Aisle) ภายในคลังสินค้าให้เป็นโหนด (Node) ที่อยู่ระหว่างชั้นวางสินค้า (Rack or Goods shelves) โดยจะนำข้อมูลที่ได้ไปคำนวณหาระยะทางที่มีความสัมพันธ์ระหว่างทุก ๆ ตำแหน่งภายในคลังสินค้า และแสดงผลออกมาในรูปแบบของแผนภูมิจากไป (From-To Chart) ดังตารางที่ 1 โดยสามารถนำโปรแกรมนี้ไปประยุกต์ใช้ในธุรกิจที่มีคลังสินค้าได้



ภาพที่ 3 รูปแบบคลังสินค้าจริงที่ร่างลงบนโปรแกรม Microsoft Excel

ตารางที่ 1 ตารางตัวอย่างแผนภูมิจากไป

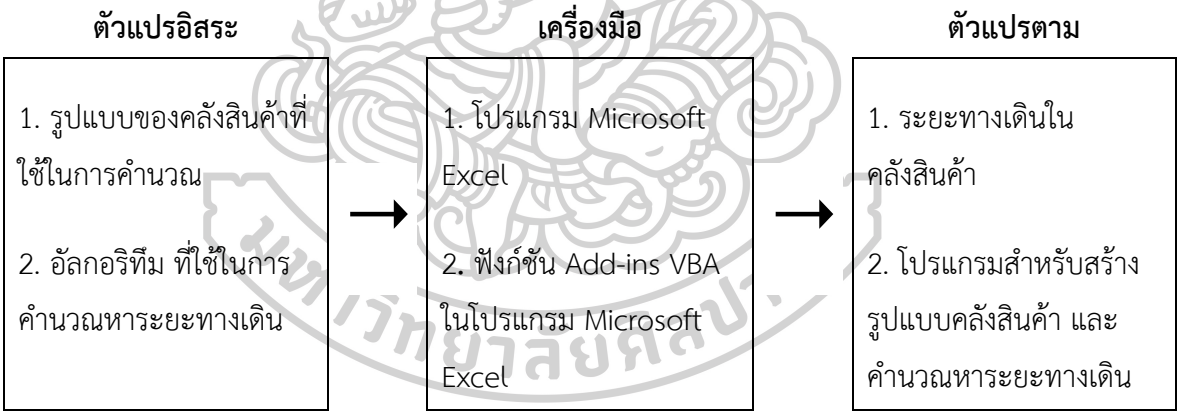
	1	2	3	4	5	.....	j
1	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	.....	D_{1j}
2	D_{21}	D_{22}	D_{23}	D_{24}	D_{25}	.....	D_{2j}
3	D_{31}	D_{32}	D_{33}	D_{34}	D_{35}	.....	D_{3j}
4	D_{41}	D_{42}	D_{43}	D_{44}	D_{45}	.....	D_{4j}
5	D_{51}	D_{52}	D_{53}	D_{54}	D_{55}	.....	D_{5j}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	D_{i1}	D_{i2}	D_{i3}	D_{i4}	D_{i5}	.....	D_{ij}

* D_{ij} แทนระยะทางจากตำแหน่ง i ไป j

1.2 วัตถุประสงค์ของการวิจัย

เพื่อสร้างโปรแกรมที่ใช้ในการคำนวณหาระยะทางจากทุกตำแหน่งภายในคลังสินค้าโดยการวาดภาพคลังสินค้าลงบนโปรแกรม Microsoft Excel

1.3 กรอบแนวคิดของการวิจัย



ภาพที่ 4 กรอบแนวคิดงานวิจัย

1.4 ขอบเขตของการวิจัย

1.4.1 ทดสอบโปรแกรมด้วยรูปแบบคลังสินค้าที่หลากกลาย เช่น คลังสินค้าที่มีรูปแบบสี่เหลี่ยม รูปตัวแอล รูปบวก คลังสินค้าที่มีทางออกสองทาง คลังสินค้าที่มีกำแพงด้านหนึ่งมีรูปแบบเป็นขั้นบันได และ คลังสินค้าที่มีออฟฟิศอยู่ตรงกลาง เป็นต้น

1.4.2 กำหนดคลังสินค้าที่มีจำนวนโหนด 20 50 100 150 200 และ 250 เพื่อทดสอบระยะเวลาที่ใช้ในการทำงานของโปรแกรม โดยขอบเขตของคลังสินค้าที่ใช้ทดสอบแสดงดังตารางที่ 2

ตารางที่ 2 ตารางขอบเขตข้อมูลคลังสินค้าที่ใช้ในการทดสอบเวลาในการทำงานของโปรแกรม

ขอบเขต	จำนวนโหนด					
	20	50	100	150	200	250
แถว	7	9	15	16	21	27
คอลัมน์	8	14	17	23	23	23
พื้นที่ (m^2)	56	126	255	368	483	621

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 สามารถหาระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้าได้จากการวาดภาพคลังสินค้าลงบนโปรแกรม Microsoft Excel

1.5.2 สามารถนำโปรแกรมนี้ไปประยุกต์ใช้ในธุรกิจที่มีคลังสินค้าได้

มหาวิทยาลัยศิลปากร

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะนำเสนอเกี่ยวกับตัวแบบโครงข่าย (Network Model) สำหรับการแก้ไขปัญหาโดยใช้กราฟ (Graph Problem) เพื่อสร้างแบบจำลองของคลังสินค้าให้อยู่ในรูปแบบโครงข่าย และคำนวณหาระยะทางระหว่างโหนดด้วยวิธีการหาระยะทางแบบแมนแฮตตัน (Manhattan Distance) รวมไปถึงการหาวิถีสั้นสุดแบบทุกคู่ (All-Pair Shortest Path) และงานวิจัยที่เกี่ยวข้อง

2.1 ตัวแบบโครงข่าย [5, 6]

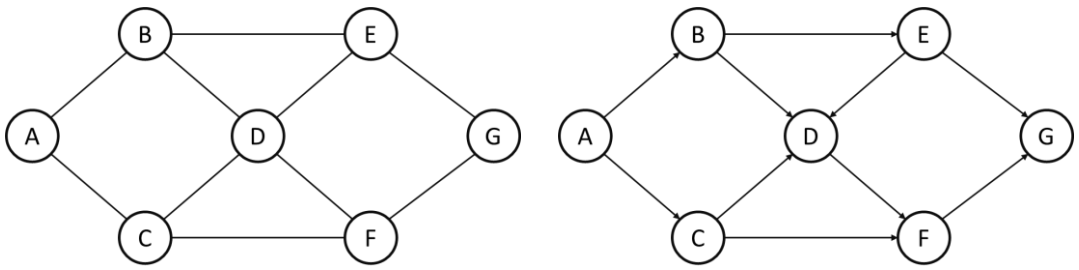
ตัวแบบโครงข่าย เป็นการแก้ไขปัญหาโดยใช้กราฟ ซึ่งจะแทนองค์ประกอบต่าง ๆ ของปัญหาด้วยการวาดกราฟ แสดงส่วนประกอบของตัวแบบโครงข่าย ดังนี้

2.1.1 โหนด หรือ จุดยอด (Vertex) แทนตำแหน่งต่าง ๆ ของปัญหาภายในกราฟ ยกตัวอย่าง เช่น เมือง หรือ ตำแหน่งของคลังสินค้า เป็นต้น

2.1.2 เส้นเชื่อมระหว่างโหนด (Arc) เป็นเส้นเชื่อมที่ใช้แสดงความสัมพันธ์ระหว่างโหนด แบ่งออกเป็น 2 ประเภท คือ

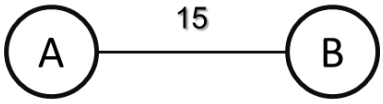
1. เส้นเชื่อมระหว่างโหนดที่ไม่มีทิศทาง (Indirect Arc) หมายถึง สามารถเดินทางไปกลับ ระหว่างโหนดได้ โดยมีน้ำหนักหรือระยะทางเท่ากัน แสดงดังภาพที่ 5 (a)
2. เส้นเชื่อมระหว่างโหนดที่มีทิศทาง (Direct Arc) หมายถึง เส้นทางเดินที่กำหนดทิศทางตามหัวของลูกศร โดยอาจจะมีน้ำหนักหรือระยะทางแตกต่างกันตามแต่ละเส้นทาง แสดงดังภาพที่ 5 (b)

2.1.3 น้ำหนัก (Weight) ใช้แสดงค่าของน้ำหนักที่ใช้ในการตัดสินใจ หรือระยะทางของปัญหา แสดงดังภาพที่ 6



(a) (b)

ภาพที่ 5 รูปแบบตัวแบบโครงข่ายแบบกราฟ
(a) แบบเส้นเชื่อมไม่มีทิศทาง (b) แบบเส้นเชื่อมมีทิศทาง

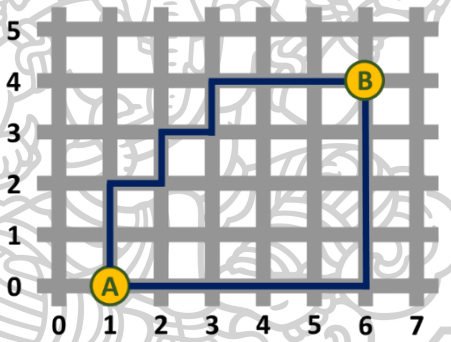


ภาพที่ 6 รูปแบบการระบุน้ำหนัก หรือระยะทางของรูปแบบโครงข่าย

2.2 ระยะทางแบบแมนแฮตตัน

รูปแบบการเดินแบบแมนแฮตตัน หรือ ระยะทางแบบแมนแฮตตัน เป็นการหาระยะทางระหว่างตำแหน่งสองตำแหน่ง ในแนวแกนที่ตั้งฉากกัน ดังภาพที่ 7 [7] ยกตัวอย่าง เช่น การหาระยะทางระหว่างตำแหน่ง A (X_1, Y_1) และตำแหน่ง B (Y_1, Y_2) สามารถหาระยะทางจาก A ไป B (D_{AB}) ได้ดังสมการที่ 1

$$D_{AB} = | X_1 - X_2 | + | Y_1 - Y_2 | \qquad (1)$$



ภาพที่ 7 รูปแบบการเดินแบบแมนแฮตตัน

2.3 วิถีสั้นสุดแบบทุกคู่

เนื่องจากต้องการทราบระยะทางเดินในทุกตำแหน่งภายในคลังสินค้า จำเป็นต้องใช้วิธีการหาวิถีสั้นสุดแบบทุกคู่ ซึ่งเป็นกำหนดการพลวัต (Dynamic Programming Algorithm) และขั้นตอนวิธีที่นิยมใช้กันมากที่สุดในการแก้ปัญหาวิถีสั้นสุดแบบทุกคู่ คือ ขั้นตอนวิธีของฟลอยด์-วอร์แชล (The Floyd-Warshall Algorithm) เนื่องจากเป็นวิธีที่ง่ายและไม่ซับซ้อน โดยมีเงื่อนไขขั้นตอนวิธีของฟลอยด์-วอร์แชล [8, 9] ดังนี้

1) เป็นกราฟที่มีทิศทาง (Directed graph)

2) เป็นกราฟที่ระบุน้ำหนักหรือระยะทาง (Weighted or Distance)

3) น้ำหนักหรือเส้นเชื่อมระหว่างโหนดเป็นบวกหรือสามารถติดลบได้

4) ต้องไม่มี (Negative Cycle) ภายในกราฟ

ขั้นตอนการดำเนินการอัลกอริทึมของฟลอยด์-วอร์แชล [10]

ขั้นตอนวิธีของฟลอยด์วอร์แชล เริ่มพิจารณาระยะทาง (Distance) โดยใช้อาร์เรย์สองมิติ ซึ่งจะได้ระยะทางระหว่างโหนดเริ่มต้น และโหนดเป้าหมาย และกำหนดให้ขั้นตอนการทำงานของ Floyd-Warshall Algorithm [8] แสดงดังภาพที่ 8 และมีขั้นตอนวิธีการดำเนินการ ดังนี้

(1) กำหนดให้ระยะทาง d (s, v) = ∞ และ pred (s, v) = 0

(2) กำหนดให้ระยะทางที่มีเส้นเชื่อมกลับเข้าหาตัวมันเอง (Self-loop) d (s, v) = 0

(3) กำหนดให้ d (s, v) คือ ระยะทางระหว่างโหนด s และโหนด v

และ pred (s, v) คือ โหนดก่อนหน้า

(4) พิจารณาเส้นทางจาก s ไป v เทียบกับ s ไป k และ k ไป v

ถ้า d (s, v) มากกว่า d (s, k) + d (k, v)

จะได้ d (s, v) เท่ากับ d (s, k) + d (k, v)

และ pred (s, v) เท่ากับ pred (k, v)

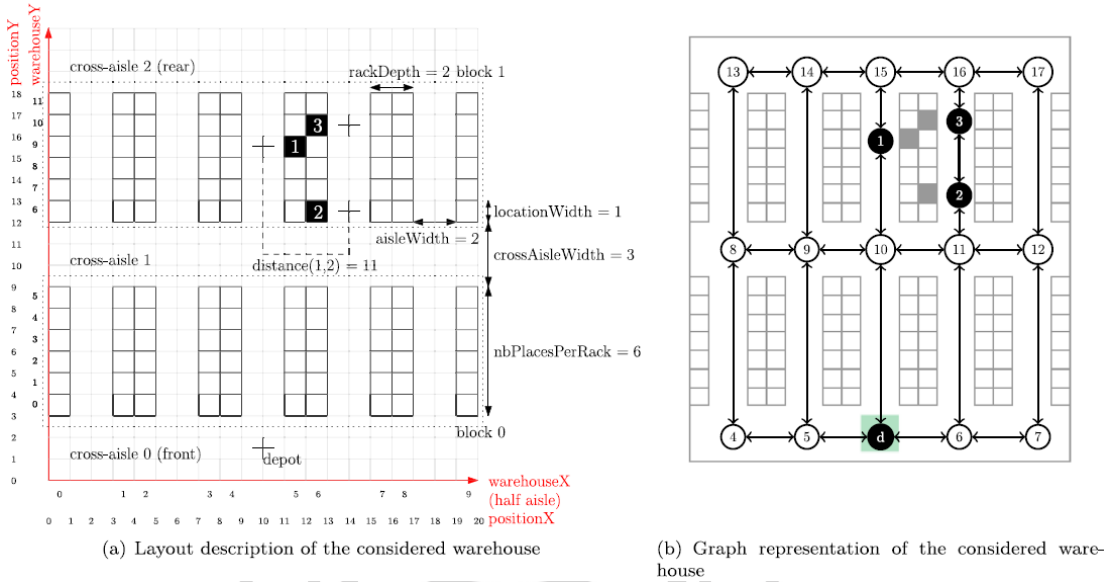
```
begin
    for all node pair(s, v) ∈ N x N do
        d(s, v) = ∞ and pred(s, v) = 0
    for all node s ∈ N do d(s, s) = 0
    for each arc (s, v) ∈ N do d(s, v) = c_sv and pred(s, v) = i
    for each k = 1 to n do
        for each(s, v) ∈ N x N do
            if d(s, v) > d(s, k) + d(k, v) then
            begin
                d(s, v) = d(s, k) + d(k, v)
                pred(s, v) = pred(k, v)
            end
end
```

ภาพที่ 8 ขั้นตอนวิธีของ Floyd-Warshall [8]

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 ตัวอย่างการออกแบบคลังสินค้าที่ใช้ในการหาวิถีสั้นสุด (Design Warehouse examples for Shortest path)

Pansart, L., N. Catusse, และ H. Cambazard (2018) [11] นำเสนอการแก้ปัญหาการหยิบสินค้า ด้วยโปรแกรมผสมจำนวนเต็ม (Mixed Integer Programming) โดยจะพิจารณาคลังสินค้าที่มีลักษณะเป็นสี่เหลี่ยมมุมฉาก แสดงดังภาพที่ 9 และกำหนดความกว้างของ Cross-aisle เท่ากับ 3 ความกว้างของ aisle และความลึกของชั้นวางของเท่ากับ 2 ยกเว้นที่ติดกับผนังจะมีความลึกเท่ากับ 1

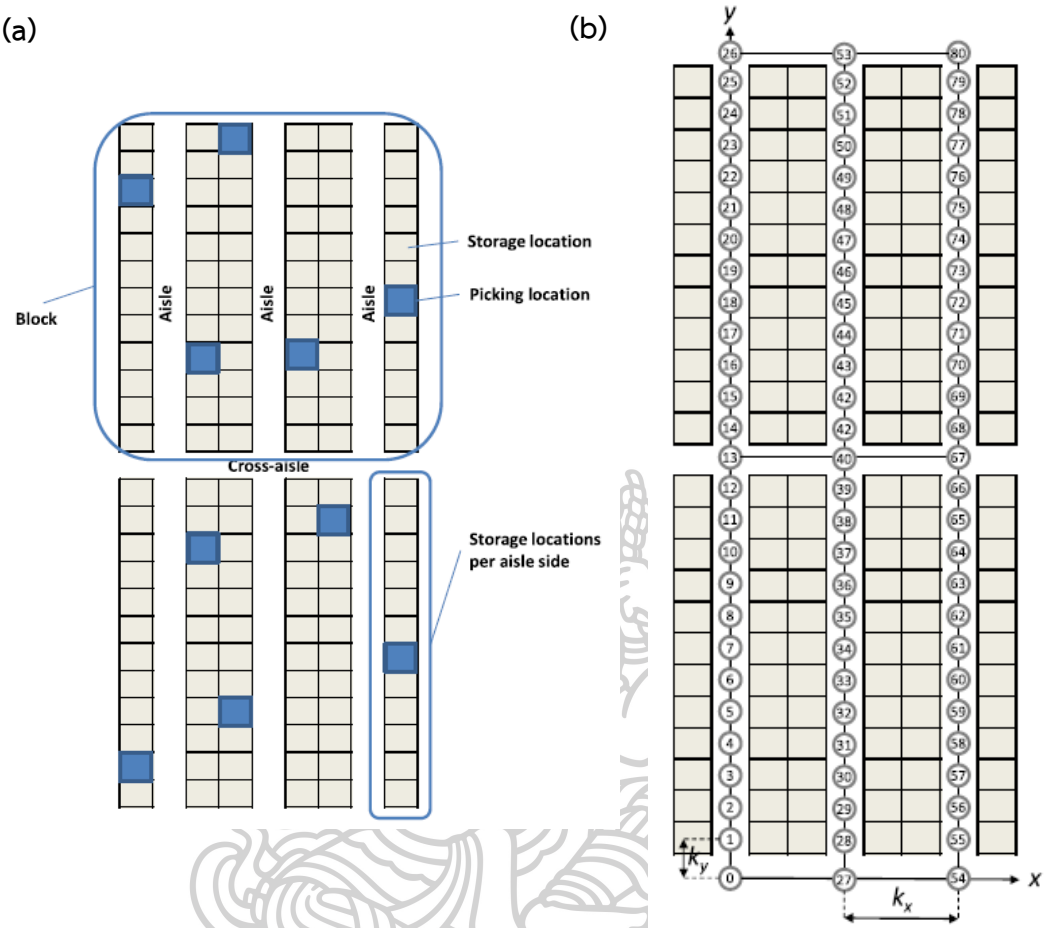


(a) Layout description of the considered warehouse

(b) Graph representation of the considered warehouse

ภาพที่ 9 ตัวอย่างคลังสินค้า

De Santis และคณะ (2018) [12] ได้อธิบายและกำหนดรูปแบบของคลังสินค้า ดังแสดงในภาพที่ 2.6 (a) โดยรูปแบบคลังสินค้าปกติ ประกอบด้วยทางเดินยาวขนานกันระหว่างที่เก็บสินค้าทั้งสองข้าง ซึ่งที่เก็บสินค้าจะแบ่งออกเป็นช่องๆ ตามชนิดของสินค้า ในส่วนของภาพที่ 10 (b) จะแสดงความสัมพันธ์เพื่อกำหนดเส้นทางเดินระหว่างที่เก็บสินค้าเป็นโหนด ทำให้สามารถระบุจำนวนช่องเพื่อง่ายต่อการประยุกต์ใช้ในการคำนวณด้วยขั้นตอนวิธีของ Floyd-Warshall โดยกำหนดให้ k_y คือ ระยะทางเดินระหว่างโหนดในแนวแกน y และ k_x คือ ช่องทางเดินระหว่างโหนดในแนวแกน x



ภาพที่ 10 (a) รูปแบบคลังสินค้า (b) เส้นทางเดินภายในคลังสินค้า

2.3.2 การประยุกต์ใช้วิถีสั้นสุดและขั้นตอนวิธีของฟลอยด์-วอร์แชล (Applications of Shortest path and Floyd-Warshall algorithm)

Hougardy, S. (2010) [13] ได้ศึกษาขั้นตอนวิธีของ Floyd-Warshall ในกรณีที่กราฟมี negative cycles เพื่อตรวจสอบความถูกต้องของขั้นตอนวิธี จะทำการพิจารณาจากจำนวนโหนดเมื่อมีจำนวนเพิ่มมากขึ้น หรือเป็นปัญหาที่มีข้อมูลขนาดใหญ่ ส่งผลให้การทำงานของขั้นตอนวิธีดังกล่าวเกิดความผิดพลาด โดยผู้วิจัยได้เสนอแนวทางการแก้ไขปัญหาที่เกิดขึ้น ด้วยการรวมบรรทัดที่ 8 และ 9 ไปไวแทนที่บรรทัดที่ 6 ก่อนจะเข้า for loop ในบรรทัดที่ 6 ดังแสดงในภาพที่ 11 หรือนำขั้นตอนวิธีของ Bellman-Ford เข้ามาช่วยในการตรวจสอบ negative cycle แล้วค่อยนำผลลัพธ์ที่ได้เข้าขั้นตอนวิธีของ Floyd-Warshall เพื่อคำนวณหาวิถีสั้นสุดแบบทุกคู่

Floyd–Warshall algorithm

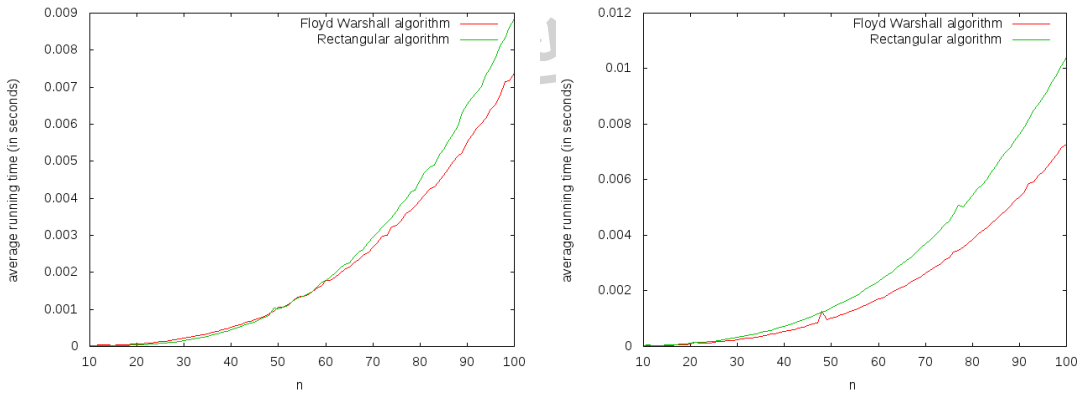
Input: A digraph G with $V(G) = \{1, \ldots, n\}$ and weights $c : E(G) \to \mathbb{R}$

Output: An $n \times n$ matrix M such that $M[i, j]$ contains the length of a shortest path from vertex i to vertex j.

```
1  M[i, j] := ∞ ∀i ≠ j
2  M[i, i] := 0 ∀i
3  M[i, j] := c((i, j)) ∀(i, j) ∈ E(G)
4  for i := 1 to n do
5      for j := 1 to n do
6          for k := 1 to n do
7              if M[j, k] > M[j, i] + M[i, k] then M[j, k] := M[j, i] + M[i, k]
8  for i := 1 to n do
9      if M[i, i] < 0 then return ('graph contains a negative cycle')
```

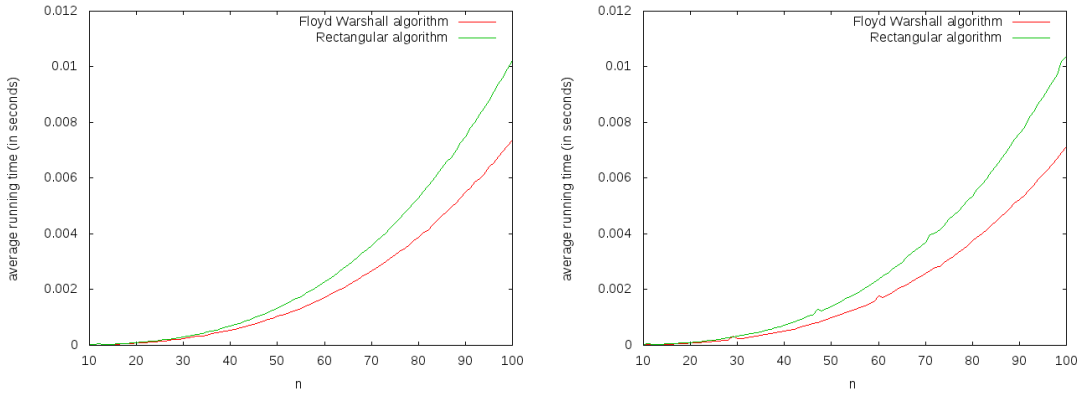
ภาพที่ 11 ขั้นตอนวิธีของ The Floyd-Warshall ที่ใช้ในการหาระยะทางทุกคู่โหนดภายในกราฟที่มี negative cycle

Singh, A. และ P.K. Mishra (2014) [14] ได้ศึกษาเปรียบเทียบขั้นตอนวิธีการหาวิถีสั้นสุดแบบทุกคู่ ระหว่างขั้นตอนวิธีของ Floyd-Warshall และ Rectangular โดยทดสอบด้วยการสร้างกราฟแบบสุ่ม (Random Graphs) ด้วยจำนวนโหนดตั้งแต่ 10 ถึง 100 ผ่านรูปแบบอีอาร์ (Erdös – Rényi (ER) model) ที่ความน่าจะเป็น (p) เท่ากับ 0.1 0.3 0.5 และ 0.8 แสดงผลการทดลองระหว่างจำนวนโหนดในแนวแกน x กับค่าเฉลี่ยของระยะเวลาในการทำงานของโปรแกรม ในแนวแกน y ดังภาพที่ 12 และ 13 สรุปได้ว่าประสิทธิภาพขั้นตอนวิธีของ Floyd-Warshall นั้นดีกว่าขั้นตอนวิธีแบบ Rectangular เมื่อจำนวนโหนด หรือขนาดของข้อมูลเพิ่มมากขึ้น



(1) (2)

ภาพที่ 12 Floyd Warshall Algorithm VS Rectangular Algorithm (Probability p = (1) 0.1 (2) 0.3

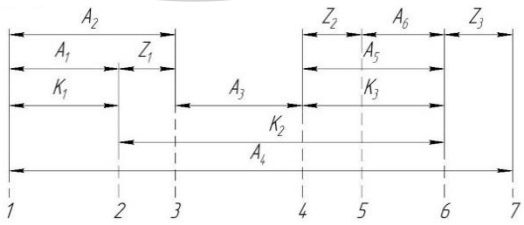


(1) (2)

ภาพที่ 13 Floyd Warshall Algorithm VS Rectangular Algorithm
(Probability p = (1) 0.5 (2) 0.8

Esuabana, I.M., I.N. Ikpang, และ E.-o.J. Okon (2015) [15] ได้นำขั้นตอนวิธีของ Floyd Warshall มาประยุกต์ใช้ในการหาเส้นทางการขนส่งของเมืองทั้ง 21 แห่ง ในประเทศ Nigeria โดยกำหนดเมือง Calabar กับ Cross River State เป็นโหนดต้นทาง และเมือง Kaduna กับ Kaduna โดยผลลัพธ์ที่ได้สามารถนำมาประยุกต์ใช้ในการแก้ปัญหาการขนส่งได้อย่างมีประสิทธิภาพ

Muholzoev, A. และ V. Masyagin (2016) [16] ได้นำขั้นตอนวิธีของฟลอยด์-วอร์แชลมาประยุกต์ใช้เพื่อคำนวณหาความน่าจะเป็นในการวิเคราะห์เกณฑ์ความคลาดเคลื่อน (Tolerance Analysis) ของขนาดชิ้นงาน (Dimension Chain) แสดงดังภาพที่ 14 ในงานวิศวกรรมเครื่องกลต้องใช้ระบบอัตโนมัติเข้ามาเกี่ยวข้องในกระบวนการผลิต โดยสามารถช่วยลดความซับซ้อนและระยะเวลาในการคำนวณ ของระบบคอมพิวเตอร์ได้ โดยผลลัพธ์ที่ได้จะแสดงออกมาในรูปของอาร์เลย์ (Array) ภาพรูปที่ 15 และ 16



ภาพที่ 14 รูปแบบ Dimension Chain

a

	1	2	3	4	5	6	7
1		A_1=K_1	A_2				A_4
2	A_1=K_1		Z_1			K_2	
3	A_2	Z_1		A_3			
4			A_3		Z_2	A_5=K_3	
5				Z_2		A_6	
6		K_2		A_5=K_3	A_6		Z_3
7	A_4					Z_3	

b

	1	2	3	4	5	6	7
1	0	0.05	0.3				0.5
2	0.05	0	Z_1			K_2	
3	0.3	Z_1	0	0.1			
4			0.1	0	Z_2	0.01	
5				Z_2	0	0.2	
6		K_2		0.01	0.2	0	Z_3
7	0.5					Z_3	0

ภาพที่ 15 (a) เมทริกซ์สัญลักษณ์ที่ใช้กำหนดรูปทรง และ
(b) เมทริกซ์ค่าความคลาดเคลื่อนเชิงตัวเลข

a

	1	2	3	4	5	6	7
1	0	0.05	0.3				0.5
2	0.05	0	Z_1			K_2	
3	0.3	Z_1	0	0.1			
4			0.1	0	Z_2	0.01	
5				Z_2	0	0.2	
6		K_2		0.01	0.2	0	Z_3
7	0.5					Z_3	0

b

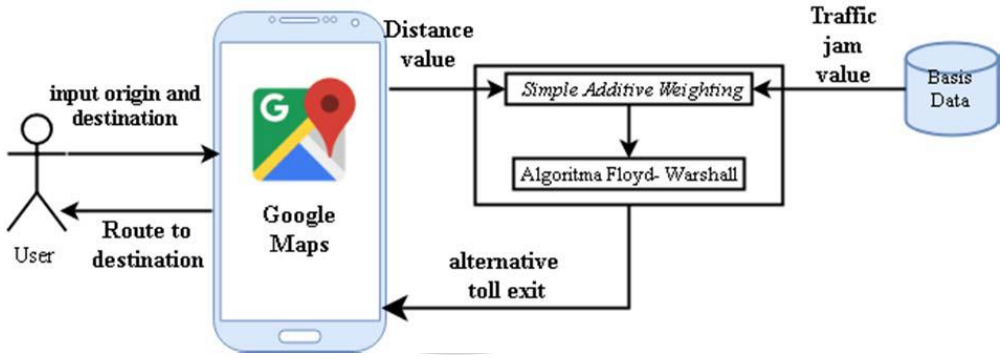
	1	2	3	4	5	6	7
1	0	0.05	0.3				0.5
2	0.05	0	Z_1			K_2	
3	0.3	Z_1	0	0.1			
4			0.1	0	Z_2	0.01	
5				Z_2	0	0.2	
6		K_2		0.01	0.2	0	Z_3
7	0.5					Z_3	0

ภาพที่ 16 (a) เมทริกซ์ค่าความคลาดเคลื่อนขั้นตอนที่ 1 และ
(b) เมทริกซ์คาความคลาดเคลื่อนขั้นตอนที่ 2

Aziz, A., M.M. Farid, และ E. Suryani (2017) [17] ได้นำขั้นตอนวิธีของ Floyd-Warshall มาใช้ในการคำนวณหาเส้นทางการอพยพของชุมชนที่อยู่ใกล้กับภูเขาไฟ เนื่องจากเหตุการณ์ในปี 2010 มีการระเบิดของภูเขาไฟ Merapi ในประเทศอินโดนีเซียทำให้มีผู้คนล้มตายจำนวนมาก ดังนั้นผู้วิจัยจึงได้ตระหนักถึงความสำคัญของเส้นทางการอพยพเพื่อให้สามารถกระจายผู้ที่อยู่ในบริเวณนั้นออกจากพื้นที่ได้อย่างปลอดภัย รวมไปถึงช่วยลดความวุ่นวาย โดยนำการตัดสินใจแบบตรรกศาสตร์คลุมเครือ (fuzzy logic) มาใช้ในการตัดสินใจรวมกับขั้นตอนวิธี Floyd-Warshall และสรุปได้ว่าขั้นตอนวิธีดังกล่าวเหมาะสมในการวางแผนเส้นทางการอพยพได้ดี

Ramadhan, Z., A.P.U. Siahaan, และ M. Mesran. (2018) [18] ศึกษาการแก้ปัญหาวิถีสั้นสุด โดยใช้ขั้นตอนวิธีของพริม (Prim Algorithm) และ Floyd-Warshall เพื่อเปรียบเทียบประสิทธิภาพการทำงานของทั้งสองวิธี ซึ่งขั้นตอนวิธีของ Prim เป็นวิธีที่ใช้กันอย่างแพร่หลายเพื่อแก้ไขปัญหาต้นไม้แบบทอดข้ามน้อยสุด (Minimum Spanning Tree) และได้พิสูจน์แล้วว่าวิธีทั้งสองสามารถแก้ไขปัญหาวิถีสั้นสุดได้ โดยขั้นตอนวิธีของ Floyd-Warshall มีประสิทธิภาพในการแก้ไขปัญหาได้ดีกว่า ต้นทุนน้อยกว่า และคำตอบที่ได้มีความเชื่อมั่นในการคำนวณหาเส้นทางที่สั้นที่สุดมากกว่าวิธีของ Prim

Pandika, I.K.L.D., B. Irawan, และ C. Setianingsih (2018) [19] ได้พัฒนาแอปพลิเคชันเพื่อแก้ปัญหาการจราจร ของนักท่องเที่ยวภายในเมือง Bandung ประเทศอินโดนีเซีย โดยนำค่าระยะทางจากแอปพลิเคชัน Google Maps ในการตัดสินใจแบบวิธีถ่วงน้ำหนัก (Simple Additive Weighting: SAW) ผนวกเข้ากับขั้นตอนวิธีของ Floyd-Warshall เพื่อแสดงเส้นทางบนแอปพลิคชัน โดยมีขั้นตอนการทำงานดังภาพที่ 17 ผลที่ได้ คือ การตัดสินใจแบบวิธีถ่วงน้ำหนักสามารถใช้งานได้ดีกับขั้นตอนวิธีของ Floyd-Warshall และแสดงเส้นทางบนแอปพลิเคชัน Google Maps ได้อย่างมีประสิทธิภาพ



ภาพที่ 17 แสดงการทำงานของแอปพลิเคชันเพื่อแก้ปัญหาการจราจร

Bukhori, D. and N. Dengen. (2018) [20] ได้นำขั้นตอนวิธีของ Floyd-Warshall มาประยุกต์ใช้ในการหาระยะทางที่สั้นที่สุด โดยใช้การปักหมุดบน Google Maps เพื่อหาระยะทางที่เหมาะสมของรถรับบริจาค ภายในเมือง Samarinda ทางตะวันออกของหมู่เกาะ Kalimantan ในประเทศอินโดนีเซีย โดยจะใช้ขั้นตอนวิธีของ Floyd-Warshall มาคำนวณหาระยะทาง ตามด้วยขั้นตอนวิธีเลือกเส้นทาง (Route Selection Algorithm) เพื่อหาเส้นทางที่เหมาะสมที่สุดจากทั้งหมด 20 แห่ง ในเมือง Samarinda

ตารางที่ 3 งานวิจัยที่เกี่ยวข้องที่เกี่ยวข้องกับขั้นตอนวิธีของ Floyd-Warshall

ผู้วิจัย	ปีที่ตีพิมพ์	วัตถุประสงค์	เครื่องมือที่ใช้ในการแก้ไขปัญหา	
			Floyd Warshall	อื่น ๆ
Hougardy, S.	2010	ศึกษาขั้นตอนวิธีของ Floyd-Warshall ในกรณีที่กราฟมี negative cycles	√	-
Singh, A. และ P.K. Mishra	2014	เปรียบเทียบวิธีการหาวิถีสั้นสุดแบบทุกคู่ของ Floyd-Warshall และ Rectangular	√	Rectangular และ Erdös – Rényi (ER) model
Esuabana, I.M., I.N. Ikpang, และ E.-o.J. Okon	2015	เพิ่มประสิทธิภาพการขนส่งของประเทศ Nigeria	√	-
Muholzoev,A. และ V. Masyagin	2016	เพื่อคำนวณหาความน่าจะเป็นในการวิเคราะห์เกณฑ์ความคลาดเคลื่อน ของขนาดชิ้นงาน	√	-
Aziz, A., M.M. Farid, และ E. Suryani	2017	คำนวณหาเส้นทางการอพยพของชุมชนที่อยู่ใกล้กับภูเขาไฟ	√	Fuzzy
Ramadhan, A.P.U. Siahaan, และ M. Mesran.	2018	ศึกษาการแก้ปัญหาวิถีสั้นสุด โดยใช้ขั้นตอนวิธีของ Prim และ Floyd-Warshall	√	Prim

ตารางที่ 3 (ต่อ) แสดงสรุปงานวิจัยที่เกี่ยวกับขั้นตอนวิธีของ Floyd-Warshall

ผู้วิจัย	ปีที่ตีพิมพ์	วัตถุประสงค์	เครื่องมือที่ใช้ในการแก้ไขปัญหา	
			Floyd Warshall	อื่น ๆ
Pandika, I.K.L.D., B. Irawan, และ C. Setianingsih	2018	พัฒนาแอปพลิเคชันเพื่อแก้ปัญหาการจราจร ของนักท่องเที่ยวภายในเมือง Bandung ประเทศอินโดนีเซีย	√	Simple Additive Weighting
Bukhori, D. และ N. Dengen.	2018	หาระยะทางที่สั้นที่สุดโดยใช้การปักหมุดบน Google Maps เพื่อหาระยะทางที่เหมาะสมในการเดินรถรับบริจาค	√	Route Selection

มหาวิทยาลัยศิลปากร

บทที่ 3
วิธีการดำเนินการ

งานวิจัยฉบับนี้เป็นการสร้างขั้นตอนวิธี ด้วยภาษา VBA ซึ่งเป็นฟังก์ชัน Add-ins ในโปรแกรมไมโครซอฟต์เอกซ์เซล โดยขั้นตอนวิธีการดำเนินงานวิจัยประกอบด้วย สร้างรูปแบบสำหรับผู้ใช้งานบนโปรแกรม Microsoft Excel เปลี่ยนค่าช่องทางเดินให้เป็นโหนด หาระยะทางระหว่างโหนด กำหนดตารางแสดงผล หาวิถีสั้นสุดแบบทุกคู่ และขั้นตอนการตรวจสอบความถูกต้องของโปรแกรม แสดงขั้นตอนการดำเนินการดังภาพที่ 18

ภาพที่ 18 ขั้นตอนการดำเนินการ

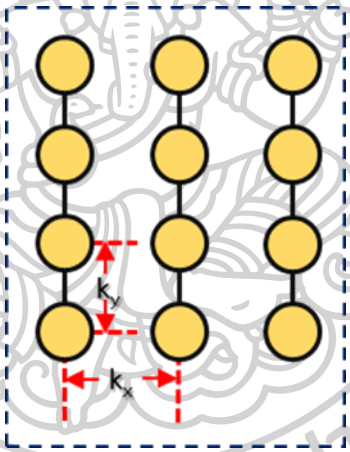
3.1 สร้างรูปแบบสำหรับผู้ใช้เพื่อกรอกข้อมูลบนโปรแกรม Microsoft Excel

3.1.1 สร้างแบบฟอร์มสำหรับการกรอกข้อมูลคลังสินค้า

สร้างแบบฟอร์มสำหรับกรอกข้อมูลของระยะห่างระหว่างชั้นวางสินค้าในแนวแกน x (k_x) และระยะห่างระหว่างชั้นวางสินค้าในแนวแกน y (k_y) แสดงดังภาพที่ 19 โดยตัวอย่างการวัดระยะห่างระหว่างชั้นว่างสินค้าของค่า k_x และ k_y แสดงดังภาพที่ 20

Variable	Values	Unit	Comment
k_x	1	m	ระยะห่างระหว่างชั้นสินค้าในแนวแกน x
k_y	1	m	ระยะห่างระหว่างชั้นสินค้าในแนวแกน y

ภาพที่ 19 ตารางข้อมูลที่ต้องกรอกในโปรแกรมจำลองคลังสินค้า



ภาพที่ 20 ระยะห่างระหว่างชั้นสินค้าในแนวแกน x และแกน y

3.1.2 กำหนดรูปแบบในการวาดภาพคลังสินค้า

กำหนดรูปแบบที่ใช้สำหรับวาดภาพคลังสินค้าในแผ่นงาน (Worksheet) บนโปรแกรม Microsoft Excel ภาพที่ 21 จะแสดงตัวอย่างรูปคลังสินค้าที่ทำการวาดบนโปรแกรม และกำหนดตัวแปรที่ใช้ในการพิจารณา ดังนี้

กำหนดให้ W คือ ขอบเขต หรือกำแพงในคลังสินค้า
S คือ ชั้นวางสินค้า
ช่องว่าง คือ ช่องทางเดินในคลังสินค้า

	A	B	C	D	E	F	G	H
1	W	W	W	W	W	W	W	W
2	W							
3	W	S		S	S		S	W
4	W	S		S	S		S	W
5	W	S		S	S		S	W
6	W							W
7	W		W	W	W	W	W	W

ภาพที่ 21 รูปแบบคลังสินค้าที่วาดบนโปรแกรม Microsoft Excel

3.2 เปลี่ยนช่องทางเดินให้เป็นโหนด

สร้างขั้นตอนวิธีสำหรับการอ่านข้อมูลของคลังสินค้า โดยจะเปลี่ยนช่องว่างให้เป็นโหนด พิจารณาจากขอบเขตของ Warehouse เริ่มนับหนึ่งจากด้านซ้ายมือ เมื่อขึ้นบรรทัดใหม่ให้นับต่อจากบรรทัดข้างต้น และเริ่มนับจากซ้ายไปขวาเช่นเดียวกัน แสดงตัวอย่างการเปลี่ยนช่องว่างให้เป็นโหนดจากตัวอย่างภาพคลังสินค้าที่ 21 จะได้โหนดดังภาพที่ 22

	A	B	C	D	E	F	G	H
1								
2		1	2	3	4	5	6	7
3			8			9		
4			10			11		
5			12			13		
6		14	15	16	17	18	19	
7		20						

ภาพที่ 22 โหนดที่ได้จากการเปลี่ยนช่องทางเดินจากตัวอย่างคลังสินค้าในภาพที่ 3.4

รหัสเทียม (Pseudo Code) ของการเปลี่ยนช่องทางเดินให้เป็นโหนด

```
Data: n, W_warehouse                                    (1)
Result: Node
begin
Node = 0                                                (2)
    For i = 1 to n do                                   (3)
        For j = 1 to n do                               (4)
            If W_warehouse [i, j] = “ ” then            (5)
                Node[i, j] = Node + 1
```

ขั้นตอนการทำงานของรหัสเทียม

(1) กำหนดตัวแปร

n	แทน	ขนาดของคลังสินค้าในแนวแกน x และ แกน y
$W_{warehouse}$	แทน	ขอบเขตรูปคลังสินค้าที่ใช้ในการพิจารณา
Node	แทน	โหนดที่ได้จากการเปลี่ยนช่องทางเดิน

(2) กำหนด Node เริ่มต้นเท่ากับศูนย์

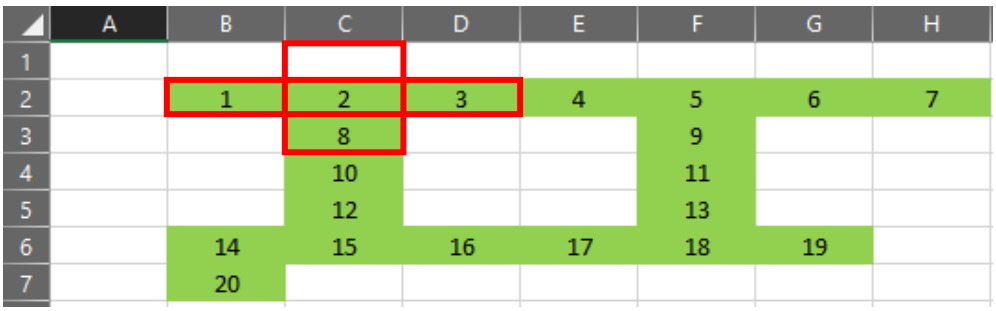
(3) ทำการวนซ้ำตามเงื่อนไขเมื่อ i เท่ากับ 1 ถึง n (พิจารณาแถว)

(4) ทำการวนซ้ำตามเงื่อนไขเมื่อ j เท่ากับ 1 ถึง n (พิจารณาคอลัมน์)

(5) พิจารณาเงื่อนไขในขอบเขตถ้า $W_{warehouse}$ [i, j] เท่ากับ ช่องว่าง “ ” ให้ Node[i, j] เท่ากับ Node + 1

3.3 หาระยะทางระหว่างโหนด

การหาระยะทางระหว่างโหนดจะพิจารณาจากโหนดที่อยู่ติดกับโหนดที่กำหนด ทางซ้ายมือ ขวามือ ด้านบน และด้านล่าง โดยกำหนดให้ระยะห่างระหว่างโหนดเท่ากับหนึ่งหน่วย ในกรณีที่ตำแหน่งใกล้เคียงโหนดที่กำหนดเป็นช่องว่างจะไม่ทำการนับ เนื่องจากไม่ได้รวมอยู่ในส่วนของเส้นทางเดิน แสดงตัวอย่างการนับช่องทางเดินดังภาพที่ 23 (a) จากภาพกำหนดให้โหนดที่ 2 เป็นโหนดที่สนใจ แล้วทำการนับช่องทางเดินทางซ้ายมือคือโหนดที่ 1 ช่องทางเดินด้านขวามือคือโหนด 3 และด้านล่างคือโหนด 1 ส่วนด้านบนจะไม่ทำการนับเนื่องจากเป็นช่องว่าง เมื่อได้ระยะทางเดินจากการพิจารณาข้างต้น แสดงดังภาพที่ 23 (b) จะนำค่าระยะทางที่ได้คูณกับค่าของ k_x และ k_y เพื่อให้ได้ระยะทางเดินจริง



	A	B	C	D	E	F	G	H
1								
2		1	2	3	4	5	6	7
3			8			9		
4			10			11		
5			12			13		
6		14	15	16	17	18	19	
7		20						

(a)

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

(b)

ภาพที่ 23 ตัวอย่าง (a) การนับช่องทางเดินของคลังสินค้า (b) ระยะทางเดินระหว่างโหนดที่ได้

รหัสเทียม (Pseudo Code) ของขั้นตอนการหาระยะทางระหว่างโหนด

Data: n, k_x, k_y, W_{Node} (1)

Result: D_{node}

begin

For i = 1 to n do (2)

For j = 1 to n do (3)

If $W_{Node}[i, j - 1] \neq$ “ ” then (3.1)

$D_{node} = 1 * k_x$

If $W_{Node}[i, j + 1] \neq$ “ ” then (3.2)

$D_{node} = 1 * k_x$

If $W_{Node}[i - 1, j] \neq$ “ ” then (3.3)

$D_{node} = 1 * k_y$

If $W_{Node}[i + 1, j] \neq$ “ ” then (3.4)

$D_{node} = 1 * k_y$

ขั้นตอนการทำงานของรหัสเทียม

(1) กำหนดตัวแปร

n แทน ขนาดของคลังสินค้าในแนวแกน x และ แกน y

W_{Node} แทน ขอบเขตรูปคลังสินค้าที่ใช้ในการพิจารณา

D_{Node} แทน ระยะทางระหว่างโหนด

k_x แทน ระยะห่างระหว่างชั้นวางสินค้ากับชั้นวางสินค้า หรือในแนวแกน x

(2) ทำการวนซ้ำตามเงื่อนไขเมื่อ i เท่ากับ 1 ถึง n (พิจารณาแถว)

(3) ทำการวนซ้ำตามเงื่อนไขเมื่อ j เท่ากับ 1 ถึง n (พิจารณาคอลัมน์)

(3.1) พิจารณาเงื่อนไขในขอบเขตถ้า $W_{Node}[i, j - 1]$ ไม่เท่ากับช่องว่าง “ ”

ให้ D_{node} เท่ากับ 1 คูณค่า k_x

(3.2) พิจารณาเงื่อนไขในขอบเขตถ้า $W_{Node}[i, j + 1]$ ไม่เท่ากับช่องว่าง “ ”

ให้ D_{node} เท่ากับ 1 คูณค่า k_x

(3.3) พิจารณาเงื่อนไขในขอบเขตถ้า $W_{Node}[i - 1, j]$ ไม่เท่ากับช่องว่าง “ ”

ให้ D_{node} เท่ากับ 1 คูณค่า k_y

(3.4) พิจารณาเงื่อนไขในขอบเขตถ้า $W_{Node}[I + 1, j]$ ไม่เท่ากับช่องว่าง “ ”

ให้ D_{node} เท่ากับ 1 คูณค่า k_y

3.4 กำหนดรูปแบบตารางแสดงผล

การแสดงผลจะแสดงออกมาในรูปของตารางภาพที่ 24 กำหนดให้คอลัมน์ซ้ายสุดเป็นโหนดเริ่มต้น และแถวบนสุดจะเป็นโหนดเป้าหมาย และภายในตารางจะแสดงค่าระยะทางที่หาได้จากขั้นตอนที่ 3.3.1 โดยจะกำหนดให้ระยะทางของโหนดเริ่มต้นที่มีโหนดปลายทางเป็นโหนดเดียวกันมีค่าเท่ากับศูนย์ และให้ระยะทางของโหนดที่ไม่อยู่ติดกันมีค่าเท่ากับอนันต์ (Infinity)

	1	**2**	**3**	**4**
1	0	1	∞	∞
2	1	0	1	∞
3	∞	1	0	1
4	∞	∞	1	0

ภาพที่ 24 รูปตารางแสดงผลของระยะที่ได้ภายในคลังสินค้า

3.5 สร้างขั้นตอนสำหรับหาวิถีสั้นสุดทุกคู่โหนด

3.5.1 หาวิถีสั้นสุดทุกคู่โหนด

ในขั้นตอนการหา All-Pair Short Path จะเลือกใช้ขั้นตอนวิธีของฟลอยด์-วอร์แชล (Floyd-Warshall Algorithm) ซึ่งเป็นวิธีที่นิยมและใช้กันอย่างแพร่หลายเนื่องจากเป็นขั้นตอนวิธีที่ง่ายและไม่ซับซ้อน โดยสามารถอธิบายขั้นตอนวิธีของ Floyd Warshall ได้ดังนี้

ขั้นตอนที่ 1 กำหนดตัวแปรที่ใช้ในการคำนวณ

pred (i, j)	คือ	ตำแหน่งของโหนดก่อนหน้า
d (i, j)	คือ	ระยะทางจากโหนด i ไป j
i	คือ	ตำแหน่งของโหนดเริ่มต้น
j	คือ	ตำแหน่งของโหนดปลายทาง
k	คือ	ตำแหน่งของโหนดที่ใช้ในการเปรียบเทียบ หรือ ตำแหน่งของทางผ่านที่อยู่ระหว่างโหนด i ไป j

ขั้นตอนที่ 2 นำเข้าข้อมูลระยะทางจากหัวข้อ 3.3

ขั้นตอนที่ 3 เรียกใช้ขั้นตอนวิธีของ Floyd-Warshall เพื่อหาระยะทางจากโหนด i ไป j

ขั้นตอนที่ 4 กำหนดระยะทางจากโหนด i ไป j เท่ากับอนันต์ (Infinity) และ ตำแหน่งของโหนดเริ่มต้นเท่ากับศูนย์

ขั้นตอนที่ 5 กำหนดระยะทางจากโหนด i ไป i เท่ากับศูนย์

ขั้นตอนที่ 6 อ่านค่าระยะทางจากโหนด i ไป j และตำแหน่งก่อนหน้าเท่ากับ i

ขั้นตอนที่ 7 พิจารณาเงื่อนไข ถ้าระยะทางจากโหนด i ไป j มากกว่าระยะทางจากโหนด i ไป k รวมกับระยะทางจากโหนด k ไป j ให้เปลี่ยนตำแหน่งของโหนดก่อนหน้าจาก i ให้เป็น k

ขั้นตอนที่ 8 ทำซ้ำขั้นตอนที่ 3 เพื่อหาระยะทางอื่น ๆ

ขั้นตอนที่ 9 เมื่อหาระยะทางครบแล้วสิ้นสุดการทำงาน

3.5.2 กำหนดรูปแบบตารางแสดงผล

3.4.2.1 ตารางแสดงระยะทาง

การแสดงผลจะแสดงออกมาในรูปของตาราง กำหนดให้คอลัมน์ซ้ายสุดเป็นโหนดเริ่มต้น และแถวบนสุดจะเป็นโหนดเป้าหมาย และภายในตารางจะแสดงค่าระยะทางวิถีสั้นสุดแบบทุกคู่

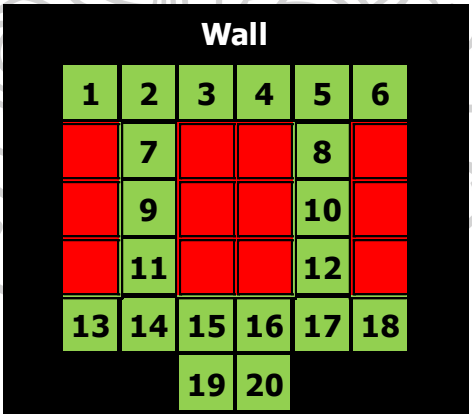
3.4.2.2 ตารางแสดงแผนภูมิจาก-ไป

การแสดงผลจะแสดงออกมาในรูปของตาราง กำหนดให้คอลัมน์ซ้ายสุดเป็นโหนดเริ่มต้น และแถวบนสุดจะเป็นโหนดเป้าหมาย และภายในตารางจะแสดงโหนดก่อนหน้า ในส่วนนี้จะเป็นตารางที่บอกเส้นทางเดิน เพื่อให้ได้มาซึ่งระยะทางที่สั้นที่สุด

3.6 ขั้นตอนการตรวจสอบความถูกต้องของโปรแกรม

3.6.1 การตรวจสอบความถูกต้องในการเปลี่ยนช่องทางเดินให้เป็นโหนด

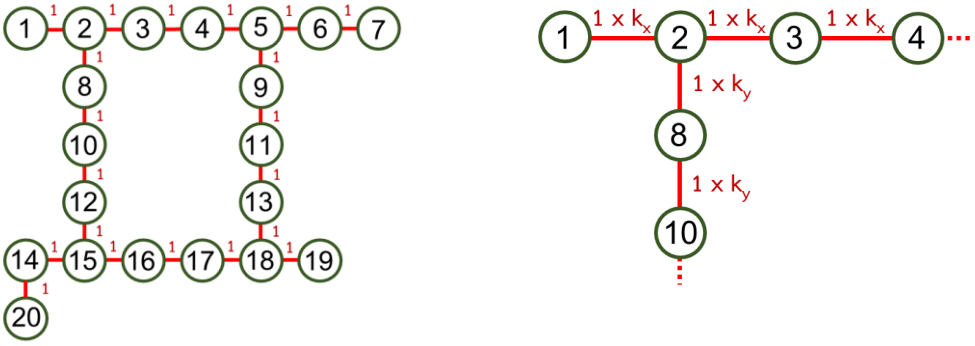
ในส่วนนี้จะตรวจสอบโหนดที่ได้จากการเปลี่ยนช่องทางเดิน และตัวเลขที่กำหนดบนช่องทางเดินดังภาพที่ 25



ภาพที่ 25 ตัวเลขของตำแหน่ง หรือโหนดที่กำหนดบนช่องทางเดิน

3.6.2 ตรวจสอบความถูกต้องของระยะทางระหว่างโหนด

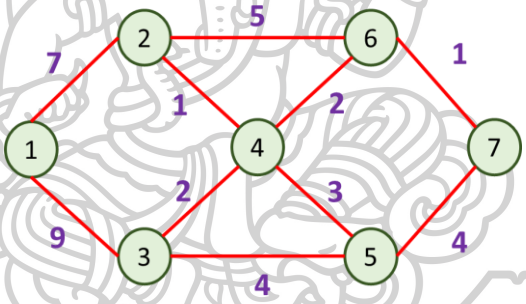
ในการตรวจสอบความถูกต้องของระยะทางระหว่างโหนดจะทำการตรวจสอบค่าของระยะทางระหว่างตำแหน่งที่อยู่ติดกันเท่านั้น และเป็นค่าของระยะทางสุดท้ายที่ได้จากการคูณค่าของตัวแปร k_x และ k_y แสดงดังภาพที่ 26 (a) และ (b)



(a) (b)

ภาพที่ 26 (a) เส้นเชื่อมและระยะห่างระหว่างโหนด
(b) แสดงน้ำหนักของเส้นเชื่อมระหว่างโหนด กรณีคูณกับ k_x และ k_y

3.6.3 ตรวจสอบความถูกต้องของระยะทางวิถีสั้นสุดทุกคู่โหนด และเส้นทางเดินภายในคลังสินค้า ยกตัวอย่างการตรวจสอบความถูกต้องดังภาพที่ 27



ภาพที่ 27 รูปแบบโครงข่ายที่ใช้ในการตรวจสอบโปรแกรม

การหาระยะทาง และเส้นทางที่สั้นที่สุด

ยกตัวอย่าง เช่น ต้องการหาระยะทาง จาก 1 ⟶ 4 จะเห็นได้ว่าเส้นทางจาก 1 ไป 4 สามารถหาได้หลายเส้นทาง อาทิ 1 ⟶ 2 ⟶ 4 และ 1 ⟶ 3 ⟶ 4

1 ⟶ 2 ⟶ 4 ; $D_{14} = D_{12} + D_{24}$

$= 7 + 1 = 8$

1 ⟶ 3 ⟶ 4 ; $D_{14} = D_{13} + D_{34}$

$= 9 + 2 = 11$

ดังนั้น คำตอบที่ถูกต้องก็คือ 1 ⟶ 2 ⟶ 4 ซึ่งมีระยะทางที่สั้นที่สุด เท่ากับ 8

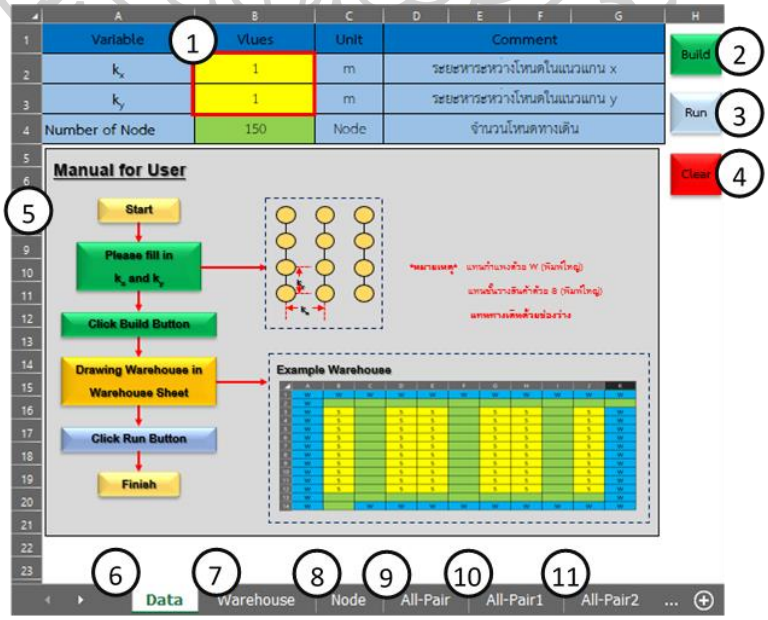
บทที่ 4
ผลและวิเคราะห์ผลการวิจัย

ผลและการวิเคราะห์ผลการวิจัย แบ่งออกเป็น 4 ส่วน โดยส่วนแรกเป็นส่วนของโปรแกรมคำนวณหาระยะทางเดินภายในคลังสินค้า ประกอบด้วย รูปแบบโปรแกรม และขั้นตอนการใช้งานโปรแกรม ส่วนที่สองเป็นผลการทดสอบโปรแกรมของคลังสินค้าที่รูปแบบเป็นสี่เหลี่ยม และไม่ใช้สี่เหลี่ยม เช่น คลังสินค้ารูปตัวแอล ฯลฯ ส่วนที่สามเป็นผลการตรวจสอบความถูกต้องของโปรแกรม และส่วนสุดท้ายเป็นการวิเคราะห์จำนวนโหนดที่ส่งผลต่อเวลาการทำงานของโปรแกรม

4.1 โปรแกรมคำนวณหาระยะทางเดินภายในคลังสินค้า

4.1.1 รูปแบบโปรแกรม

รูปแบบโปรแกรมในโปรแกรมแสดงดังภาพที่ 28 โดยจะระบุองค์ประกอบต่าง ๆ ของโปรแกรม ตัวโปรแกรมประกอบด้วย 1) ช่องสำหรับกรอกข้อมูล k_x และ k_y เพื่อใช้ในการคำนวณหาระยะทางจริง 2) ปุ่มสำหรับสร้างแผ่นงาน 3) ปุ่มสำหรับเริ่มทำงานโปรแกรม 4) ปุ่มสำหรับลบข้อมูลภายในแผ่นงาน Node All-Pair All-Pair1 และ All-Pair2 5) คู่มือการใช้งานโปรแกรม 6) แผ่นงาน Data ใช้สำหรับกรอกข้อมูลในหน้าต่างปัจจุบัน 7) แผ่นงาน Warehouse ใช้สำหรับวาดรูปคลังสินค้าบนโปรแกรม 8) แผ่นงาน Node สำหรับแสดงรูปแบบโครงข่ายของช่องทางเดิน 9) แผ่นงาน All-Pair แสดงระยะห่างระหว่างโหนด 10) และ 11) แผ่นงาน All-Pair1 และ 2 ใช้แสดงวิถีสั้นสุดทุกคู่โหนด



ภาพที่ 28 รูปแบบโปรแกรมคำนวณหาระยะทางภายในคลังสินค้า

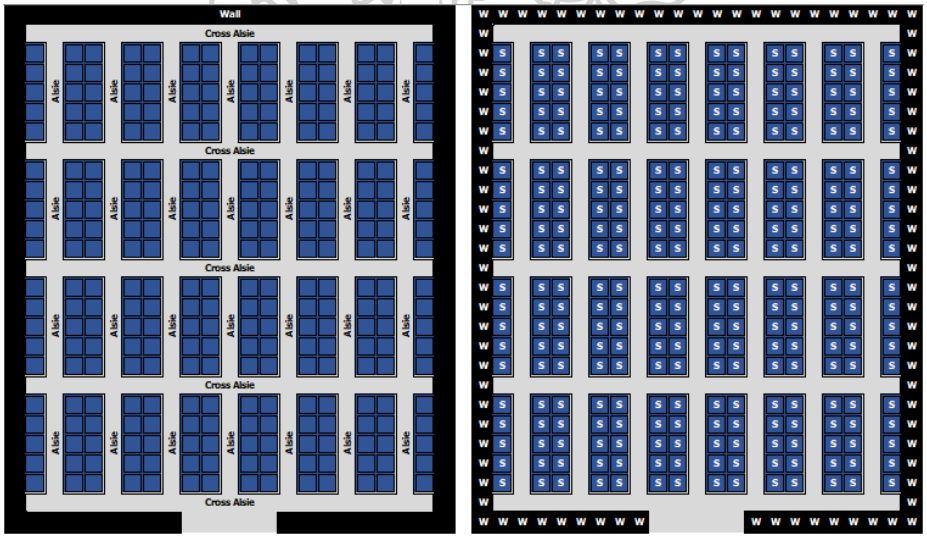
4.1.2 ขั้นตอนการใช้งานโปรแกรม

ขั้นตอนที่ 1 กำหนดตัวแปร k_x และ k_y เท่ากับ 1.5 และ 2 ตามลำดับ กรอกค่าของ k_x และ k_y ลงบนแบบฟอร์มของโปรแกรม Microsoft Excel ดังภาพที่ 29

	A	B	C	D	E	F	G	H
1	Variable	Vlues	Unit	Comment				Build
2	k_x	1	m	ระยะหาระหว่างโหนดในแนวแกน x				
3	k_y	1	m	ระยะหาระหว่างโหนดในแนวแกน y				Run
4	Number of Node	150	Node	จำนวนโหนดทางเดิน				

ภาพที่ 29 รูปแบบแบบฟอร์มที่ใช้ในการกรอกข้อมูล k_x และ k_y

ขั้นตอนที่ 2 ทำการร่างรูปแบบคลังสินค้าจากภาพที่ 30 (a) ลงบนโปรแกรม Microsoft Excel จะได้ดังภาพที่ 30 (b)



(a) (b)

ภาพที่ 30 (a) รูปแบบคลังสินค้าจริง (b) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม Microsoft Excel

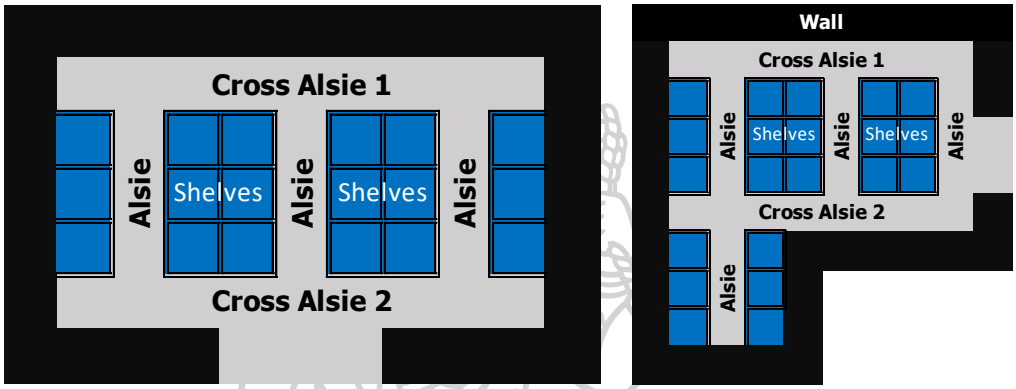
ขั้นตอนที่ 3 ทำการคลิกปุ่ม Run เพื่อเริ่มการทำงานของโปรแกรม ดังภาพที่ 31

	A	B	C	D	E	F	G	H
1	Variable	Vlues	Unit	Comment				Build
2	k_x	1	m	ระยะหาระหว่างโหนดในแนวแกน x				
3	k_y	1	m	ระยะหาระหว่างโหนดในแนวแกน y				Run
4	Number of Node	150	Node	จำนวนโหนดทางเดิน				

ภาพที่ 31 ภาพปุ่มที่ใช้สำหรับการเริ่มทำงานของโปรแกรม

4.2 ผลการทดสอบโปรแกรม

ในการทดสอบโปรแกรมจะทำการทดสอบโปรแกรมด้วยรูปแบบคลังสินค้าที่เป็นสี่เหลี่ยมและรูปแบบคลังสินค้าที่ไม่ใช่รูปสี่เหลี่ยม ดังภาพที่ 32 (a) และ (b) ตามลำดับ เพื่อให้ง่ายต่อการแสดงผลของตารางแผนภูมิจาก-ไป ทางผู้วิจัยได้กำหนดจำนวนโหนดของการทดสอบคลังสินค้าเท่ากับ 30 โหนด รวมไปถึงทำการทดสอบจำนวนโหนดที่ส่งผลกระทบต่อระยะเวลาในการทำงานของโปรแกรม



(a) (b)

ภาพที่ 32 (a) รูปแบบคลังสินค้าที่เป็นรูปสี่เหลี่ยม (b) รูปแบบคลังสินค้าที่เป็นรูปตัวแอล

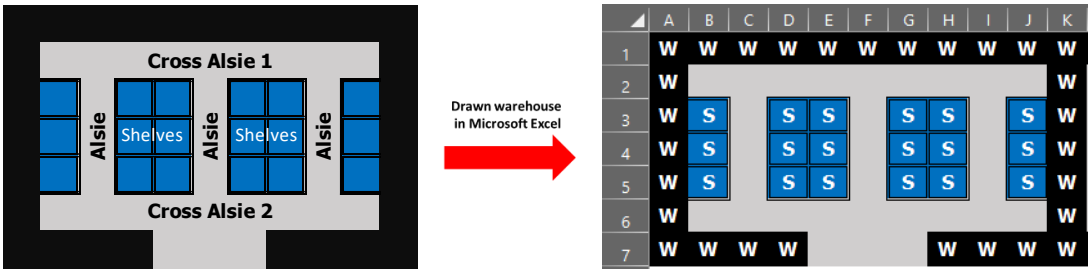
4.2.1 รูปแบบคลังสินค้าที่เป็นสี่เหลี่ยม

ขั้นตอนที่ 1 กำหนดตัวแปร k_x และ k_y เท่ากับ 1.5 และ 2 ตามลำดับ กรอกค่าของ k_x และ k_y ลงบนแบบฟอร์มของโปรแกรม Microsoft Excel ดังภาพที่ 33

Variable	Vlues	Unit	Comment
k_x	1.5	m	ระยะหาระหว่างโหนดในแนวแกน x
k_y	2	m	ระยะหาระหว่างโหนดในแนวแกน y
Number of Node	30	Node	จำนวนโหนดทางเดิน

ภาพที่ 33 รูปแบบแบบฟอร์มที่ใช้ในการกรอกข้อมูล k_x และ k_y

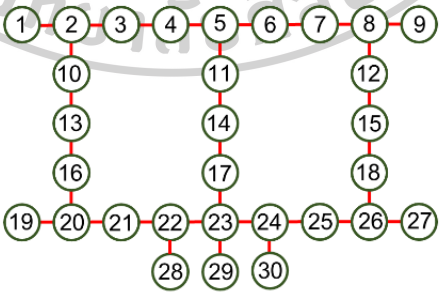
ขั้นตอนที่ 2 ทำการร่างรูปแบบสินค้าลงบนโปรแกรม Microsoft Excel จะได้ดังภาพที่ 34



ภาพที่ 34 รูปแบบคลังสินค้าที่ร่างลงบนโปรแกรม Microsoft Excel

ขั้นตอนที่ 3 ทำการคลิกปุ่ม Run เพื่อเริ่มการทำงานของโปรแกรม จะได้ผลลัพธ์ดังภาพที่ 35 (a) และ (b) ซึ่งแสดงรูปแบบของโหนดที่ได้จากการเปลี่ยนช่องทางเดินจากภาพที่ 34 โดยตัวโปรแกรมจะกำหนดตัวเลข เพื่อใช้ในการพิจารณาตำแหน่งเริ่มต้น และตำแหน่งเป้าหมายในการหาระยะทาง เมื่อคำนวณหาระยะทางเสร็จจะได้รูปแบบตารางดังภาพที่ 36 หลังจากนั้นตัวโปรแกรมจะนำตารางที่ได้มาคำนวณหาวิถีสั้นสุดทุกคู่โหนด และแสดงผลลัพธ์ที่ได้ในรูปแบบของตาราง From to Chart ดังภาพที่ 37

(a)



(b)

ภาพที่ 35 (a) รูปแบบโหนดที่ได้จากโปรแกรม Microsoft Excel
(b) โหนดในรูปแบบโครงข่าย

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	**0**	**1.5**																												
2	**1.5**	**0**	**1.5**							**2**																				
3		**1.5**	**0**	**1.5**																										
4			**1.5**	**0**	**1.5**																									
5				**1.5**	**0**	**1.5**					**2**																			
6					**1.5**	**0**	**1.5**																							
7						**1.5**	**0**	**1.5**																						
8							**1.5**	**0**	**1.5**			**2**																		
9								**1.5**	**0**																					
10		**2**								**0**			**2**																	
11					**2**						**0**			**2**																
12								**2**				**0**			**2**															
13										**2**			**0**			**2**														
14											**2**			**0**			**2**													
15												**2**			**0**			**2**												
16													**2**			**0**				**2**										
17														**2**			**0**						**2**							
18															**2**			**0**						**2**						
19																			**0**	**1.5**										
20																**2**			**1.5**	**0**	**1.5**									
21																				**1.5**	**0**	**1.5**								
22																					**1.5**	**0**	**1.5**					**2**		
23																	**2**					**1.5**	**0**	**1.5**					**2**	
24																							**1.5**	**0**	**1.5**					**2**
25																								**1.5**	**0**	**1.5**				
26																		**2**							**1.5**	**0**	**1.5**			
27																										**1.5**	**0**			
28																						**2**						**0**	**1.5**	
29																							**2**					**1.5**	**0**	**1.5**
30																								**2**					**1.5**	**0**

ภาพที่ 36 ตารางระยะห่างระหว่างโหนดของคลังสินค้าที่มีรูปแบบสี่เหลี่ยม

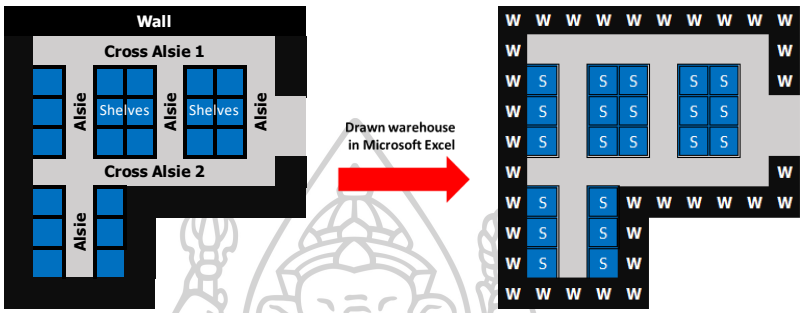
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	1.5	3	4.5	6	7.5	9	10.5	12	3.5	8	12.5	5.5	10	14.5	7.5	12	16.5	11	9.5	11	12.5	14	15.5	17	18.5	20	14.5	16	17.5
2	1.5	0	1.5	3	4.5	6	7.5	9	10.5	2	6.5	11	4	8.5	13	6	10.5	15	9.5	8	9.5	11	12.5	14	15.5	17	18.5	13	14.5	16
3	3	1.5	0	1.5	3	4.5	6	7.5	9	3.5	5	9.5	5.5	7	11.5	7.5	9	13.5	11	9.5	11	12.5	11	12.5	14	15.5	17	14.5	13	14.5
4	4.5	3	1.5	0	1.5	3	4.5	6	7.5	5	3.5	8	7	5.5	10	9	7.5	12	12.5	11	12.5	11	9.5	11	12.5	14	15.5	13	11.5	13
5	6	4.5	3	1.5	0	1.5	3	4.5	6	6.5	2	6.5	8.5	4	8.5	10.5	6	10.5	14	12.5	11	9.5	8	9.5	11	12.5	14	11.5	10	11.5
6	7.5	6	4.5	3	1.5	0	1.5	3	4.5	8	3.5	5	10	5.5	7	12	7.5	9	15.5	14	12.5	11	9.5	11	12.5	11	12.5	13	11.5	13
7	9	7.5	6	4.5	3	1.5	0	1.5	3	9.5	5	3.5	11.5	7	5.5	13.5	9	7.5	17	15.5	14	12.5	11	12.5	11	9.5	11	14.5	13	14.5
8	10.5	9	7.5	6	4.5	3	1.5	0	1.5	11	6.5	2	13	8.5	4	15	10.5	6	18.5	17	15.5	14	12.5	11	9.5	8	9.5	16	14.5	13
9	12	10.5	9	7.5	6	4.5	3	1.5	0	12.5	8	3.5	14.5	10	5.5	16.5	12	7.5	20	18.5	17	15.5	14	12.5	11	9.5	11	17.5	16	14.5
10	3.5	2	3.5	5	6.5	8	9.5	11	12.5	0	8.5	13	2	10.5	15	4	12.5	17	7.5	6	7.5	9	10.5	12	13.5	15	16.5	11	12.5	14
11	8	6.5	5	3.5	2	3.5	5	6.5	8	8.5	0	8.5	10.5	2	10.5	12.5	4	12.5	12	10.5	9	7.5	6	7.5	9	10.5	12	9.5	8	9.5
12	12.5	11	9.5	8	6.5	5	3.5	2	3.5	13	8.5	0	15	10.5	2	17	12.5	4	16.5	15	13.5	12	10.5	9	7.5	6	7.5	14	12.5	11
13	5.5	4	5.5	7	8.5	10	11.5	13	14.5	2	10.5	15	0	12.5	17	2	10.5	15	5.5	4	5.5	7	8.5	10	11.5	13	14.5	9	10.5	12
14	10	8.5	7	5.5	4	5.5	7	8.5	10	10.5	2	10.5	12.5	0	12.5	10.5	2	10.5	10	8.5	7	5.5	4	5.5	7	8.5	10	7.5	6	7.5
15	14.5	13	11.5	10	8.5	7	5.5	4	5.5	15	10.5	2	17	12.5	0	15	10.5	2	14.5	13	11.5	10	8.5	7	5.5	4	5.5	12	10.5	9
16	7.5	6	7.5	9	10.5	12	13.5	15	16.5	4	12.5	17	2	10.5	15	0	8.5	13	3.5	2	3.5	5	6.5	8	9.5	11	12.5	7	8.5	10
17	12	10.5	9	7.5	6	7.5	9	10.5	12	12.5	4	12.5	10.5	2	10.5	8.5	0	8.5	8	6.5	5	3.5	2	3.5	5	6.5	8	5.5	4	5.5
18	16.5	15	13.5	12	10.5	9	7.5	6	7.5	17	12.5	4	15	10.5	2	13	8.5	0	12.5	11	9.5	8	6.5	5	3.5	2	3.5	10	8.5	7
19	11	9.5	11	12.5	14	15.5	17	18.5	20	7.5	12	16.5	5.5	10	14.5	3.5	8	12.5	0	1.5	3	4.5	6	7.5	9	10.5	12	6.5	8	9.5
20	9.5	8	9.5	11	12.5	14	15.5	17	18.5	6	10.5	15	4	8.5	13	2	6.5	11	1.5	0	1.5	3	4.5	6	7.5	9	10.5	5	6.5	8
21	11	9.5	11	12.5	11	12.5	14	15.5	17	7.5	9	13.5	5.5	7	11.5	3.5	5	9.5	3	1.5	0	1.5	3	4.5	6	7.5	9	3.5	5	6.5
22	12.5	11	12.5	11	9.5	11	12.5	14	15.5	9	7.5	12	7	5.5	10	5	3.5	8	4.5	3	1.5	0	1.5	3	4.5	6	7.5	2	3.5	5
23	14	12.5	11	9.5	8	9.5	11	12.5	14	10.5	6	10.5	8.5	4	8.5	6.5	2	6.5	6	4.5	3	1.5	0	1.5	3	4.5	6	3.5	2	3.5
24	15.5	14	12.5	11	9.5	11	12.5	11	12.5	12	7.5	9	10	5.5	7	8	3.5	5	7.5	6	4.5	3	1.5	0	1.5	3	4.5	5	3.5	2
25	17	15.5	14	12.5	11	12.5	11	9.5	11	13.5	9	7.5	11.5	7	5.5	9.5	5	3.5	9	7.5	6	4.5	3	1.5	0	1.5	3	6.5	5	3.5
26	18.5	17	15.5	14	12.5	11	9.5	8	9.5	15	10.5	6	13	8.5	4	11	6.5	2	10.5	9	7.5	6	4.5	3	1.5	0	1.5	8	6.5	5
27	20	18.5	17	15.5	14	12.5	11	9.5	11	16.5	12	7.5	14.5	10	5.5	12.5	8	3.5	12	10.5	9	7.5	6	4.5	3	1.5	0	9.5	8	6.5
28	14.5	13	14.5	13	11.5	13	14.5	16	17.5	11	9.5	14	9	7.5	12	7	5.5	10	6.5	5	3.5	2	3.5	5	6.5	8	9.5	0	1.5	3
29	16	14.5	13	11.5	10	11.5	13	14.5	16	12.5	8	12.5	10.5	6	10.5	8.5	4	8.5	8	6.5	5	3.5	2	3.5	5	6.5	8	1.5	0	1.5
30	17.5	16	14.5	13	11.5	13	14.5	13	14.5	14	9.5	11	12	7.5	9	10	5.5	7	9.5	8	6.5	5	3.5	2	3.5	5	6.5	3	1.5	0

ภาพที่ 37 ตารางระยะห่างระหว่างโหนดของคลังสินค้าที่มีรูปแบบสี่เหลี่ยม

4.2.2 รูปแบบคลังสินค้าอื่น ๆ

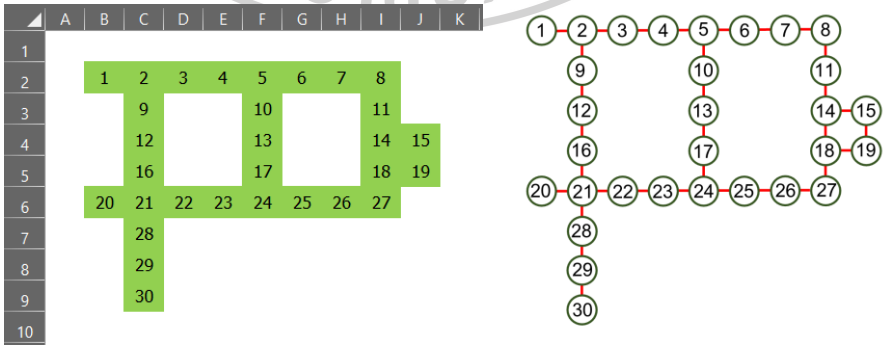
ขั้นตอนที่ 1 กำหนดตัวแปร kx และ ky เท่ากับ 1.5 และ 2 ตามลำดับ กรอกค่าของ kx และ ky ลงบนแบบฟอร์มของโปรแกรม Microsoft Excel ดังภาพที่ 33

ขั้นตอนที่ 2 ทำการร่างรูปแบบสินค้าลงบนโปรแกรม Microsoft Excel จะได้ดังภาพที่ 38



ภาพที่ 38 รูปแบบคลังสินค้าที่ร่างลงบนโปรแกรม Microsoft Excel

ขั้นตอนที่ 3 ทำการคลิกปุ่ม Run เพื่อเริ่มการทำงานของโปรแกรม จะได้ผลลัพธ์ที่ได้ดังภาพที่ 39 ซึ่งแสดงรูปแบบของโหนดที่ได้จากการเปลี่ยนช่องทางเดินจากภาพที่ 38 โดยตัวโปรแกรมจะกำหนดตัวเลข เพื่อใช้ในการพิจารณาตำแหน่งเริ่มต้น และตำแหน่งเป้าหมายในการหาระยะทาง เมื่อคำนวณหาระยะทางเสร็จจะได้รูปแบบตารางดังภาพที่ 40 หลังจากนั้นตัวโปรแกรมจะนำตารางที่ได้มาคำนวณหาวิถีสั้นสุดทุกคู่โหนด และแสดงผลลัพธ์ที่ได้ในรูปแบบของตาราง From to Chart ดังภาพที่ 41



(a) (b)

ภาพที่ 39 (a) รูปแบบโหนดที่ได้จากโปรแกรม Microsoft Excel
(b) แสดงโหนดในรูปแบบโครงข่าย

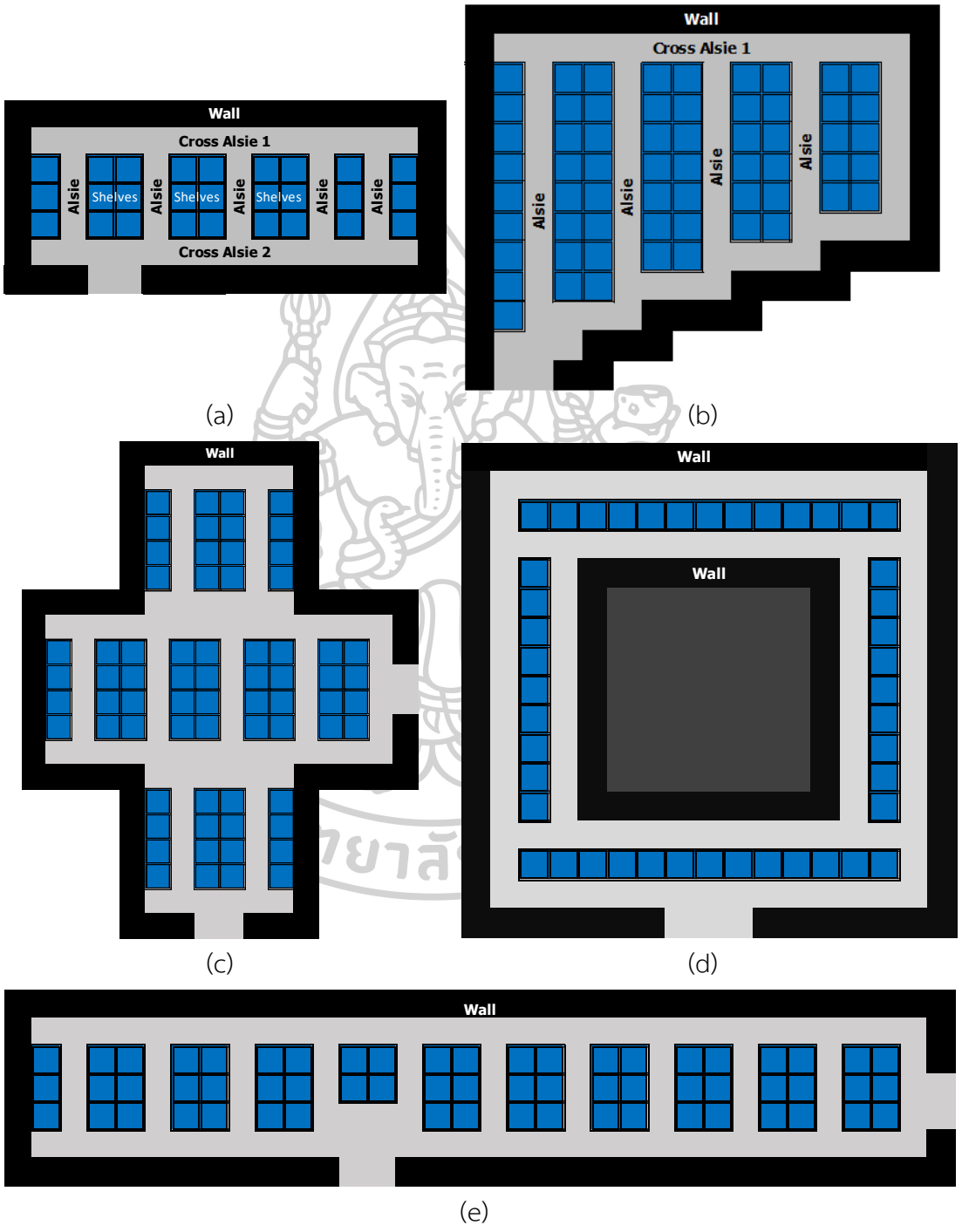
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	1.5																												
2	1.5	0	1.5						2																					
3		1.5	0	1.5																										
4			1.5	0	1.5																									
5				1.5	0	1.5				2																				
6					1.5	0	1.5																							
7						1.5	0	1.5																						
8							1.5	0			2																			
9		2							0			2																		
10					2					0			2																	
11								2			0			2																
12									2			0				2														
13										2			0				2													
14											2			0	1.5			2												
15														1.5	0				2											
16												2				0					2									
17													2				0							2						
18														2				0	1.5								2			
19															2			1.5	0											
20																				0	1.5									
21																2				1.5	0	1.5						2		
22																					1.5	0	1.5							
23																						1.5	0	1.5						
24																	2						1.5	0	1.5					
25																								1.5	0	1.5				
26																									1.5	0	1.5			
27																		2								1.5	0			
28																					2							0	2	
29																												2	0	2
30																													2	0

ภาพที่ 40 ตารางระยะห่างระหว่างโหนดของคลังสินค้าที่มีรูปแบบตัวแอล

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	1.5	3	4.5	6	7.5	9	10.5	3.5	8	12.5	5.5	10	14.5	16	7.5	12	16.5	18	11	9.5	11	12.5	14	15.5	17	18.5	11.5	13.5	15.5
2	1.5	0	1.5	3	4.5	6	7.5	9	2	6.5	11	4	8.5	13	14.5	6	10.5	15	16.5	9.5	8	9.5	11	12.5	14	15.5	17	10	12	14
3	3	1.5	0	1.5	3	4.5	6	7.5	3.5	5	9.5	5.5	7	11.5	13	7.5	9	13.5	15	11	9.5	11	12.5	11	12.5	14	15.5	11.5	13.5	15.5
4	4.5	3	1.5	0	1.5	3	4.5	6	5	3.5	8	7	5.5	10	11.5	9	7.5	12	13.5	12.5	11	12.5	11	9.5	11	12.5	14	13	15	17
5	6	4.5	3	1.5	0	1.5	3	4.5	6.5	2	6.5	8.5	4	8.5	10	10.5	6	10.5	12	14	12.5	11	9.5	8	9.5	11	12.5	14.5	16.5	18.5
6	7.5	6	4.5	3	1.5	0	1.5	3	8	3.5	5	10	5.5	7	8.5	12	7.5	9	10.5	15.5	14	12.5	11	9.5	11	12.5	11	16	18	20
7	9	7.5	6	4.5	3	1.5	0	1.5	9.5	5	3.5	11.5	7	5.5	7	13.5	9	7.5	9	17	15.5	14	12.5	11	12.5	11	9.5	17.5	19.5	21.5
8	10.5	9	7.5	6	4.5	3	1.5	0	11	6.5	2	13	8.5	4	5.5	15	10.5	6	7.5	18.5	17	15.5	14	12.5	11	9.5	8	19	21	23
9	3.5	2	3.5	5	6.5	8	9.5	11	0	8.5	13	2	10.5	15	16.5	4	12.5	17	18.5	7.5	6	7.5	9	10.5	12	13.5	15	8	10	12
10	8	6.5	5	3.5	2	3.5	5	6.5	8.5	0	8.5	10.5	2	10.5	12	12.5	4	12.5	14	12	10.5	9	7.5	6	7.5	9	10.5	12.5	14.5	16.5
11	12.5	11	9.5	8	6.5	5	3.5	2	13	8.5	0	15	10.5	2	3.5	17	12.5	4	5.5	16.5	15	13.5	12	10.5	9	7.5	6	17	19	21
12	5.5	4	5.5	7	8.5	10	11.5	13	2	10.5	15	0	12.5	17	18.5	2	10.5	15	16.5	5.5	4	5.5	7	8.5	10	11.5	13	6	8	10
13	10	8.5	7	5.5	4	5.5	7	8.5	10.5	2	10.5	12.5	0	12.5	14	10.5	2	10.5	12	10	8.5	7	5.5	4	5.5	7	8.5	10.5	12.5	14.5
14	14.5	13	11.5	10	8.5	7	5.5	4	15	10.5	2	17	12.5	0	1.5	15	10.5	2	3.5	14.5	13	11.5	10	8.5	7	5.5	4	15	17	19
15	16	14.5	13	11.5	10	8.5	7	5.5	16.5	12	3.5	18.5	14	1.5	0	16.5	12	3.5	2	16	14.5	13	11.5	10	8.5	7	5.5	16.5	18.5	20.5
16	7.5	6	7.5	9	10.5	12	13.5	15	4	12.5	17	2	10.5	15	16.5	0	8.5	13	14.5	3.5	2	3.5	5	6.5	8	9.5	11	4	6	8
17	12	10.5	9	7.5	6	7.5	9	10.5	12.5	4	12.5	10.5	2	10.5	12	8.5	0	8.5	10	8	6.5	5	3.5	2	3.5	5	6.5	8.5	10.5	12.5
18	16.5	15	13.5	12	10.5	9	7.5	6	17	12.5	4	15	10.5	2	3.5	13	8.5	0	1.5	12.5	11	9.5	8	6.5	5	3.5	2	13	15	17
19	18	16.5	15	13.5	12	10.5	9	7.5	18.5	14	5.5	16.5	12	3.5	2	14.5	10	1.5	0	14	12.5	11	9.5	8	6.5	5	3.5	14.5	16.5	18.5
20	11	9.5	11	12.5	14	15.5	17	18.5	7.5	12	16.5	5.5	10	14.5	16	3.5	8	12.5	14	0	1.5	3	4.5	6	7.5	9	10.5	3.5	5.5	7.5
21	9.5	8	9.5	11	12.5	14	15.5	17	6	10.5	15	4	8.5	13	14.5	2	6.5	11	12.5	1.5	0	1.5	3	4.5	6	7.5	9	2	4	6
22	11	9.5	11	12.5	11	12.5	14	15.5	7.5	9	13.5	5.5	7	11.5	13	3.5	5	9.5	11	3	1.5	0	1.5	3	4.5	6	7.5	3.5	5.5	7.5
23	12.5	11	12.5	11	9.5	11	12.5	14	9	7.5	12	7	5.5	10	11.5	5	3.5	8	9.5	4.5	3	1.5	0	1.5	3	4.5	6	5	7	9
24	14	12.5	11	9.5	8	9.5	11	12.5	10.5	6	10.5	8.5	4	8.5	10	6.5	2	6.5	8	6	4.5	3	1.5	0	1.5	3	4.5	6.5	8.5	10.5
25	15.5	14	12.5	11	9.5	11	12.5	11	12	7.5	9	10	5.5	7	8.5	8	3.5	5	6.5	7.5	6	4.5	3	1.5	0	1.5	3	8	10	12
26	17	15.5	14	12.5	11	12.5	11	9.5	13.5	9	7.5	11.5	7	5.5	7	9.5	5	3.5	5	9	7.5	6	4.5	3	1.5	0	1.5	9.5	11.5	13.5
27	18.5	17	15.5	14	12.5	11	9.5	8	15	10.5	6	13	8.5	4	5.5	11	6.5	2	3.5	10.5	9	7.5	6	4.5	3	1.5	0	11	13	15
28	11.5	10	11.5	13	14.5	16	17.5	19	8	12.5	17	6	10.5	15	16.5	4	8.5	13	14.5	3.5	2	3.5	5	6.5	8	9.5	11	0	2	4
29	13.5	12	13.5	15	16.5	18	19.5	21	10	14.5	19	8	12.5	17	18.5	6	10.5	15	16.5	5.5	4	5.5	7	8.5	10	11.5	13	2	0	2
30	15.5	14	15.5	17	18.5	20	21.5	23	12	16.5	21	10	14.5	19	20.5	8	12.5	17	18.5	7.5	6	7.5	9	10.5	12	13.5	15	4	2	0

ภาพที่ 41 ตารางระยะห่างระหว่างโหนดของคลังสินค้าที่มีรูปแบบตัวแอล

นอกจากรูปแบบคลังสินค้ารูปตัวแอลที่แสดงในข้างต้นแล้ว โปรแกรมยังสามารถหาระยะทางของคลังสินค้าได้อีกหลายรูปแบบ ดังแสดงในภาพที่ 42 โดยผลลัพธ์ของระยะทางระหว่างโหนด และระยะทางวิถีสั้นสุดทุกคู่ปม จะแสดงไว้ในภาคผนวก ก.



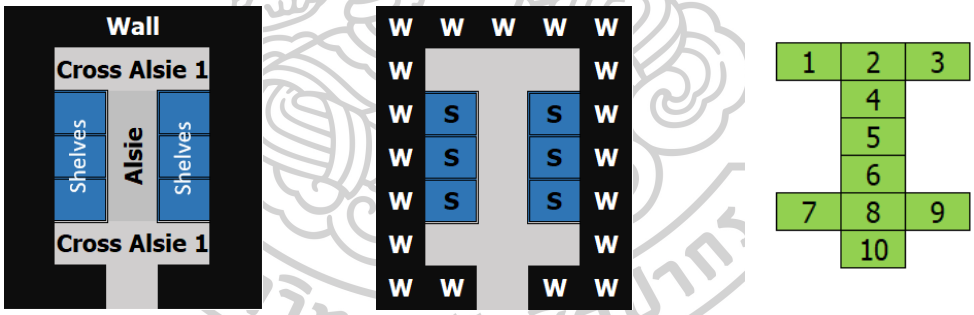
ภาพที่ 42 รูปแบบคลังสินค้าที่ใช้ในการทดสอบโปรแกรม

4.3 ผลการตรวจสอบความถูกต้องของโปรแกรม

ในการตรวจสอบความถูกต้องของโปรแกรมนั้นจะแบ่งออกเป็น 3 ส่วน คือ (1) การเปลี่ยนช่องทางเดินให้เป็นโหนด (2) ระยะทางระหว่างโหนดเมื่อคูณกับค่า k_x และ k_y (3) ระยะทางวิถีสั้นสุดทุกคู่ และเส้นทางการเดิน รูปแบบที่ใช้ในการตรวจสอบความถูกต้องของโปรแกรมแสดงดังภาพที่ 43 (a) โดยกำหนดค่าของ k_x และ k_y เท่ากับ 2 และ 1 ตามลำดับ

4.3.1 ตรวจสอบความถูกต้องของช่องทางเดินที่เปลี่ยนเป็นโหนด แสดงดังภาพที่ 43 (b) เปลี่ยนไปเป็น 43 (c) จะเห็นได้ว่าช่องทางเดินดังกล่าวจะกำหนดตัวเลขเพื่อแสดงถึงตำแหน่ง หรือโหนดที่ใช้ในการคำนวณหาระยะทาง

4.3.2 ระยะทางระหว่างโหนดแสดงดังภาพที่ 43 (d) จากการกำหนดระยะห่างในแนวแกน x จะมีค่าเท่ากับ 2 หน่วย และ ระยะหาในแนวแกน y จะมีค่าเท่ากับ 1 หน่วย ยกตัวอย่างการตรวจสอบในแนวแกน x จากตำแหน่ง 1 ⟶ 2 และ 7 ⟶ 8 มีระยะทางระหว่างโหนดถูกต้องตรงตามค่า k_x ที่กำหนดไว้ข้างต้น และทำการตรวจสอบซ้ำในแนวแกน y จากตำแหน่ง 2 ⟶ 4 และ 5 ⟶ 6 มีระยะทางระหว่างโหนดถูกต้องตรงตามค่า k_y ที่กำหนด



(a) (b) (c)

	1	2	3	4	5	6	7	8	9	10
1	0	2	9999	9999	9999	9999	9999	9999	9999	9999
2	2	0	2	1	9999	9999	9999	9999	9999	9999
3	9999	2	0	9999	9999	9999	9999	9999	9999	9999
4	9999	1	9999	0	1	9999	9999	9999	9999	9999
5	9999	9999	9999	1	0	1	9999	9999	9999	9999
6	9999	9999	9999	9999	1	0	9999	1	9999	9999
7	9999	9999	9999	9999	9999	9999	0	2	9999	9999
8	9999	9999	9999	9999	9999	1	2	0	2	1
9	9999	9999	9999	9999	9999	9999	9999	2	0	9999
10	9999	9999	9999	9999	9999	9999	9999	1	9999	0

(d)

ภาพที่ 43 (a) รูปแบบคลังสินค้าที่ใช้ในการตรวจสอบโปรแกรม (c) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม (c) โหนดที่ได้จากการเปลี่ยนช่องทางเดิน และ (d) ระยะทางระหว่างโหนด

4.3.3 เส้นทางวิถีสั้นสุด และระยะทางวิถีสั้นสุดของทุกตำแหน่งภายในคลังสินค้า แสดงดังภาพที่ 44 (a) และ 44 (b) ตามลำดับ ในการตรวจสอบความถูกต้องของโปรแกรมนั้นจะแสดงค่าของระยะทาง และเส้นทางของวิถีสั้นสุด ดังตารางที่ 4 ภายในตารางระบุตำแหน่งเริ่มต้น และตำแหน่งปลายทาง ระยะทาง และเส้นทางการเดิน ยกตัวอย่าง เช่น ต้องการทราบระยะทางเดิน และเส้นทางการเดิน จากตำแหน่งที่ 1 ไปยังตำแหน่งที่ 6 มีระยะทางที่สั้นที่สุดเท่ากับ 5 และมีเส้นทางการเดิน คือ 1-2-3-4-5-6 หมายถึง จาก 1 ไป 6 ต้องเดินผ่านตำแหน่งที่ 2 3 4 5 ตามลำดับ

	1	2	3	4	5	6	7	8	9	10
1	1	1	2	2	4	5	8	6	8	8
2	2	2	2	2	4	5	8	6	8	8
3	2	3	3	2	4	5	8	6	8	8
4	2	4	2	4	4	5	8	6	8	8
5	2	4	2	5	5	5	8	6	8	8
6	2	4	2	5	6	6	8	6	8	8
7	2	4	2	5	6	8	7	7	8	8
8	2	4	2	5	6	8	8	8	8	8
9	2	4	2	5	6	8	8	9	9	8
10	2	4	2	5	6	8	8	10	8	10

(a)

	1	2	3	4	5	6	7	8	9	10
1	0	2	4	3	4	5	8	6	8	7
2	2	0	2	1	2	3	6	4	6	5
3	4	2	0	3	4	5	8	6	8	7
4	3	1	3	0	1	2	5	3	5	4
5	4	2	4	1	0	1	4	2	4	3
6	5	3	5	2	1	0	3	1	3	2
7	8	6	8	5	4	3	0	2	4	3
8	6	4	6	3	2	1	2	0	2	1
9	8	6	8	5	4	3	4	2	0	3
10	7	5	7	4	3	2	3	1	3	0

(b)

ภาพที่ 44 (a) แสดงเส้นทางเดินภายในคลังสินค้า และ (b) แสดงระยะทางวิถีสั้นสุดทุกคู่โหนดภายในคลังสินค้า

ตารางที่ 4 ผลการตรวจสอบค่าของระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า

ตำแหน่งเริ่มต้น	ตำแหน่งปลายทาง	ระยะทาง	เส้นทางการเดิน
1	1	0	-
1	2	2	1-2
1	3	4	1-2-3
1	4	3	1-2-4
1	5	4	1-2-4-5

ตารางที่ 4 (ต่อ) ผลการตรวจสอบค่าของระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า

ตำแหน่งเริ่มต้น	ตำแหน่งปลายทาง	ระยะทาง	เส้นทางการเดิน
1	6	5	1-2-4-5-6
1	7	8	1-2-4-5-6-8-7
1	8	6	1-2-4-5-6-8
1	9	8	1-2-4-5-6-8-9
1	10	7	1-2-4-5-6-8-10
2	1	2	2-1
2	2	0	-
2	3	2	2-3
2	4	1	2-4
2	5	2	2-4-5
2	6	3	2-4-5-6
2	7	6	2-4-5-6-8-7
2	8	4	2-4-5-6-8
2	9	6	2-4-5-6-8-9
2	10	5	2-4-5-6-8-10
3	1	4	3-2-1
3	2	2	3-2
3	3	0	-
3	4	3	3-2-4
3	5	4	3-2-4-5
3	6	5	3-2-4-5-6
3	7	8	3-2-4-5-6-8-7
3	8	6	3-2-4-5-6-8

ตารางที่ 4 (ต่อ) ผลการตรวจสอบค่าของระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า

ตำแหน่งเริ่มต้น	ตำแหน่งปลายทาง	ระยะทาง	เส้นทางการเดิน
3	9	8	3-2-4-5-6-8-9
3	10	7	3-2-4-5-6-8-10
4	1	3	4-2-1
4	2	1	4-2
4	3	3	4-2-3
4	4	0	-
4	5	1	4-5
4	6	2	4-5-6
4	7	5	4-5-6-8-7
4	8	3	4-5-6-8
4	9	5	4-5-6-8-9
4	10	4	4-5-6-8-10
5	1	4	5-4-2-1
5	2	2	5-4-2
5	3	4	5-4-2-3
5	4	1	5-4
5	5	0	-
5	6	1	5-6
5	7	4	5-6-8-7
5	8	2	5-6-8
5	9	4	5-6-8-9
5	10	3	5-6-8-10
6	1	5	6-5-4-2-1

ตารางที่ 4 (ต่อ) ผลการตรวจสอบค่าของระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า

ตำแหน่งเริ่มต้น	ตำแหน่งปลายทาง	ระยะทาง	เส้นทางการเดิน
6	2	3	6-5-4-2
6	3	5	6-5-4-2-3
6	4	2	6-5-4
6	5	1	6-5
6	6	0	-
6	7	3	6-8-7
6	8	1	6-8
6	9	3	6-8-9
6	10	2	6-8-10
7	1	8	7-8-6-5-4-2-1
7	2	6	7-8-6-5-4-2
7	3	8	7-8-6-5-4-2-3
7	4	5	7-8-6-5-4
7	5	4	7-8-6-5
7	6	3	7-8-6
7	7	0	-
7	8	2	7-8
7	9	4	7-8-9
7	10	3	7-8-10
8	1	6	8-6-5-4-2-1
8	2	4	8-6-5-4-2
8	3	6	8-6-5-4-2-3
8	4	3	8-6-5-4

ตารางที่ 4 (ต่อ) ผลการตรวจสอบค่าของระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า

ตำแหน่งเริ่มต้น	ตำแหน่งปลายทาง	ระยะทาง	เส้นทางการเดิน
8	5	2	8-6-5
8	6	1	8-6
8	7	2	8-7
8	8	0	-
8	9	2	8-9
8	10	1	8-10
9	1	8	9-8-6-5-4-2-1
9	2	6	9-8-6-5-4-2
9	3	8	9-8-6-5-4-3
9	4	5	9-8-6-5-4
9	5	4	9-8-6-5
9	6	3	9-8-6
9	7	4	9-8-7
9	8	2	9-8
9	9	0	-
9	10	3	9-8-10
10	1	7	10-8-6-5-4-2-1
10	2	5	10-8-6-5-4-2
10	3	7	10-8-6-5-4-2-3
10	4	4	10-8-6-5-4
10	5	3	10-8-6-5
10	6	2	10-8-6
10	7	3	10-8-7

ตารางที่ 4 (ต่อ) ผลการตรวจสอบค่าของระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า

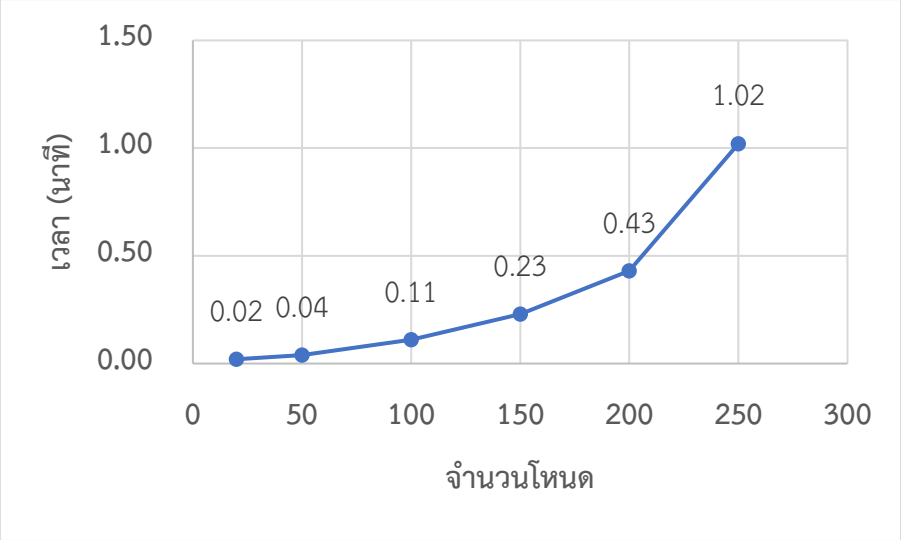
ตำแหน่งเริ่มต้น	ตำแหน่งปลายทาง	ระยะทาง	เส้นทางการเดิน
10	8	1	10-8
10	9	3	10-8-9
10	10	0	-

4.4 จำนวนโหนดที่ส่งผลต่อระยะเวลาการทำงานของโปรแกรม

ทดสอบจำนวนโหนดที่ส่งผลต่อระยะเวลาการทำงานของโปรแกรม ทางผู้วิจัยใช้คอมพิวเตอร์โน้ตบุ๊ค Lenovo รุ่น ideapad Z510 ซึ่งมีระบบประมวลผล Intel Core i7-4702MQ (2.20 GHz, 6 MB L3 Cache, up to 3.20 GHz) ทำให้ทราบว่าที่จำนวนโหนด 20 50 100 150 200 และ 250 โหนด ใช้เวลาในการทำงานของโปรแกรม 0.02 0.04 0.11 0.23 0.43 และ 1.02 นาที จะเห็นได้ว่าเมื่อเพิ่มจำนวนโหนดในการคำนวณหาระยะทางจะมีระยะเวลาเพิ่มขึ้น ดังตารางที่ 5 และภาพที่ 45 อย่างไรก็ตามระยะเวลาที่ใช้ในการคำนวณนั้นจะขึ้นอยู่กับประสิทธิภาพของอุปกรณ์ที่ใช้ในการทำงานของโปรแกรม

ตารางที่ 5 แสดงข้อมูลเวลาในการทำงานของโปรแกรมที่จำนวนโหนดต่างกัน

จำนวนโหนด	เวลา (วินาที)				เวลา (นาที)
	1	2	3	Avg.	
20	1	2	2	2	0.02
50	5	4	4	4	0.04
100	13	10	10	11	0.11
150	23	22	23	23	0.23
200	42	47	40	43	0.43
250	62	62	62	62	1.02



ภาพที่ 45 กราฟความสัมพันธ์ระหว่างเวลากับจำนวนโหนด

มหาวิทยาลัยศิลปากร

บทที่ 5
สรุปผลการวิจัย

5.1 สรุปผลการวิจัย

โปรแกรมคำนวณหาระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า สามารถคำนวณหาระยะทางจริงภายในคลังสินค้าได้ ทั้งรูปแบบคลังสินค้าที่เป็นสี่เหลี่ยม และรูปแบบอื่น ๆ อาทิ คลังสินค้ารูปตัวแอล คลังสินค้ารูปตัวที เป็นต้น โดยโปรแกรมนี้จะช่วยลดระยะเวลาในการวัดเส้นทางภายในคลังสินค้าจากการใช้มนุษย์ หรือวิธีการประมาณค่า (Euclidean distance) ในการทดสอบโปรแกรมโดยใช้ขนาดคลังสินค้าที่แตกต่างกันโดยการเพิ่มจำนวนโหนดของช่องทางเดินภายในคลังสินค้า ทำให้ทราบว่าที่จำนวนโหนด 20 50 100 150 200 และ 250 โหนด ใช้เวลาในการทำงานของโปรแกรม 0.02 0.04 0.11 0.23 0.43 และ 1.02 นาที ตามลำดับ และสรุปได้ว่าที่จำนวนโหนดเพิ่มมากขึ้นระยะเวลาในการทำงานของโปรแกรมจะเพิ่มตามไปด้วย อย่างไรก็ตามระยะเวลาในการทำงานของโปรแกรมนั้นจะขึ้นอยู่กับประสิทธิภาพของเครื่องมือ หรือคอมพิวเตอร์ที่ใช้ในการทำงานของโปรแกรมคำนวณหาระยะทาง

5.2 ข้อเสนอแนะ

5.2.1 ข้อมูลที่ได้จากโปรแกรม คือ ตารางแผนภูมิจาก-ไป ของระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า โดยสามารถนำข้อมูลที่ได้ไปประยุกต์ใช้ในการคำนวณหาเส้นทางเดินหรือ เส้นทางหยิบสินค้าที่เหมาะสมภายในคลังสินค้าได้

5.2.2 สามารถนำโปรแกรมที่ได้ไปประยุกต์ใช้ร่วมกับขั้นตอนวิธีการอบเหนี่ยว (Simulated Annealing Algorithm) ขั้นตอนวิธีการปีนเข้า (Hill Climbing Algorithm) ขั้นตอนวิธีหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด (Ant Colony Optimization) ฯลฯ เพื่อใช้ในการหาเส้นทางที่เหมาะสมในการหยิบสินค้าภายในคลังสินค้าได้

รายการอ้างอิง

1. Beroule, B., et al., *Order Picking Problem in a Warehouse Hospital Pharmacy.* IFAC-PapersOnLine, 2017. **50**(1): p. 5017-5022.
2. Richards, G., *Warehouse management: a complete guide to improving efficiency and minimizing costs in the modern warehouse*. 2017: Kogan Page Publishers.
3. Giannikas, V., et al., *An interventionist strategy for warehouse order picking: Evidence from two case studies.* International Journal of Production Economics, 2017. **189**: p. 63-76.
4. Lu, W., et al., *An algorithm for dynamic order-picking in warehouse operations.* European Journal of Operational Research, 2016. **248**(1): p. 107-122.
5. ไพฑูรย์, การวิเคราะห์เชิงปริมาณ (*Quantitative Analysis*). 2561, กรุงเทพฯ: บริษัท วี.พริ้นท์(1991) จำกัด.
6. ชิดชนก, *Analysis & Design of Algorithms.* 2543, กรุงเทพฯ: โรงพิมพ์ ดี แอล เอส กรุงเทพฯ.
7. Pandit, S. and S. Gupta, *A comparative study on distance measuring approaches for clustering.* International Journal of Research in Computer Science, 2011. **2**(1): p. 29-31.
8. Ahuja, R.K., et al., *Network flows: theory, algorithms and applications.* ZOR-Methods and Models of Operations Research, 1995. **41**(3): p. 252-254.
9. Skiena, S.S., *The algorithm design manual: Text*. Vol. 1. 1998: Springer Science & Business Media.
10. Heineman, G.T., G. Pollice, and S. Selkow, *Algorithms in a nutshell: A practical guide*. 2016: " O'Reilly Media, Inc.".
11. Pansart, L., N. Catusse, and H. Cambazard, *Exact algorithms for the order picking problem.* Computers & Operations Research, 2018. **100**: p. 117-127.
12. De Santis, R., et al., *An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses.* European Journal of Operational Research, 2018. **267**(1): p. 120-137.
13. Hougardy, S., *The Floyd–Warshall algorithm on graphs with negative cycles.* Information Processing Letters, 2010. **110**(8-9): p. 279-281.

14. Singh, A. and P.K. Mishra, *Performance Analysis of Floyd Warshall Algorithm vs Rectangular Algorithm.* International Journal of Computer Applications, 2014. **107**(16).

15. Esuabana, I.M., I.N. Ikpang, and E.-o.J. Okon, *Shortest Transportation Route Network in Nigeria Using Floyd-Warshall's Algorithm.*

16. Muholzoev, A. and V. Masyagin, *Probabilistic calculation of tolerances of the dimension chain based on the Floyd-Warshall algorithm.* Procedia Engineering, 2016. **150**: p. 959-962.

17. Aziz, A., M.M. Farid, and E. Suryani. *Floyd warshall algorithm with FIS Sugeno for search evacuation route optimization*. in *2017 International Seminar on Application for Technology of Information and Communication (iSemantic)*. 2017. IEEE.

18. Ramadhan, Z., A.P.U. Siahaan, and M. Mesran. *Prim and Floyd-Warshall Comparative Algorithms in Shortest Path Problem*. in *Proceedings of the Joint Workshop KO2PI and The 1st International Conference on Advance & Scientific Innovation*. 2018.

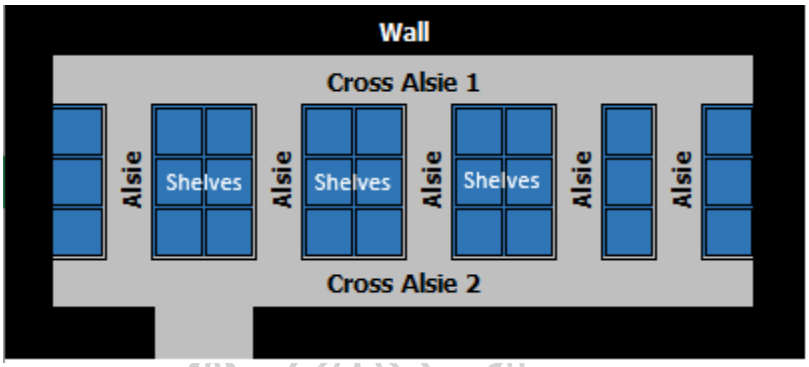
19. Pandika, I.K.L.D., B. Irawan, and C. Setianingsih. *Apllication of Optimization Heavy Traffic Path with Floyd-Warshall Algorithm*. in *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*. 2018. IEEE.

20. Bukhori, D. and N. Dengen. *Floyd-warshall algorithm to determine the shortest path based on android*. in *IOP Conference Series: Earth and Environmental Science*. 2018. IOP Publishing.

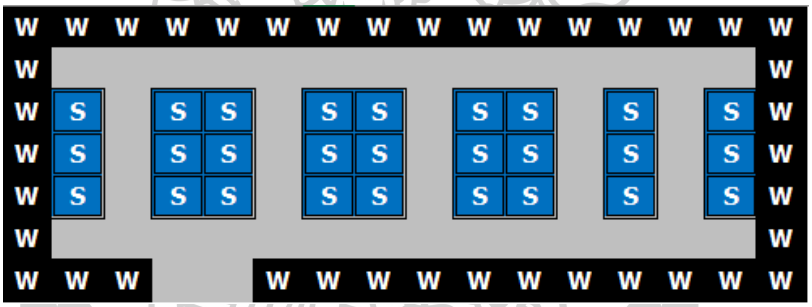
ภาคผนวก ก
ผลลัพธ์ของโปรแกรมที่ใช้ทดสอบคลังสินค้า

ผลลัพธ์ของโปรแกรมที่ใช้ทดสอบคลังสินค้า

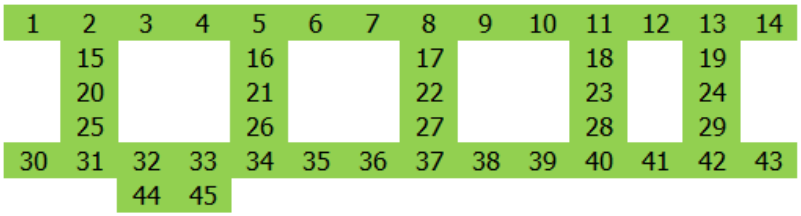
1. **คลังสินค้าที่เป็นสี่เหลี่ยมผืนผ้า**



ภาพที่ 46 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ



(a)



(b)

ภาพที่ 47 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม
(b) รูปแบบโครงข่ายที่ได้จากโปรแกรม

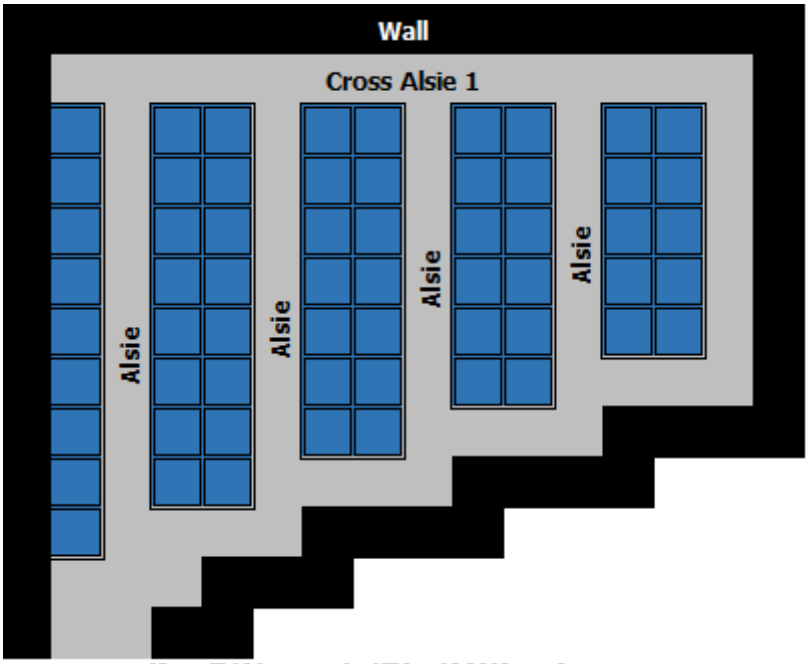
ตารางที่ 6 ระยะทางระหว่างโหนดของคลังสินค้าที่มีรูปแบบสี่เหลี่ยมผืนผ้า

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	....	**45**
1	0	2										**....**	
2	2	0	2									....	
3		2	0	2								....	
4			2	0	2							....	
5				2	0	2						....	
6					2	0	2					....	
7						2	0	2				....	
8							2	0	2			....	
9								2	0	2		....	
10									2	0	2	....	
11										2	0	....	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
45												....	0

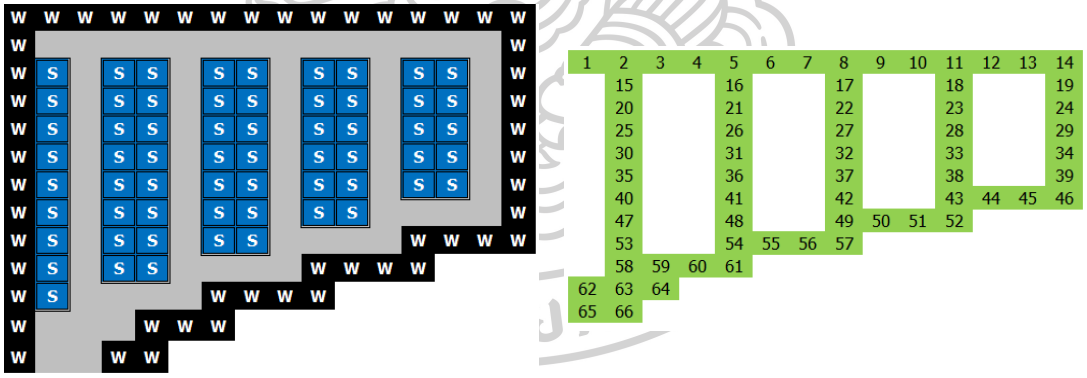
ตารางที่ 7 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่มีรูปแบบสี่เหลี่ยมผืนผ้า

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	....	**45**
1	0	2	4	6	8	10	12	14	16	18	20	**....**	11
2	2	0	2	4	6	8	10	12	14	16	18	....	9
3	4	2	0	2	4	6	8	10	12	14	16	....	11
4	6	4	2	0	2	4	6	8	10	12	14	....	9
5	8	6	4	2	0	2	4	6	8	10	12	....	7
6	10	8	6	4	2	0	2	4	6	8	10	....	9
7	12	10	8	6	4	2	0	2	4	6	8	....	11
8	14	12	10	8	6	4	2	0	2	4	6	....	13
9	16	14	12	10	8	6	4	2	0	2	4	....	15
10	18	16	14	12	10	8	6	4	2	0	2	....	17
11	20	18	16	14	12	10	8	6	4	2	0	....	19
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
45	11	9	11	9	7	9	11	13	15	17	19	....	0

2. คลังสินค้าที่มีกำแพงด้านหนึ่งเป็นแบบขั้นบันได



ภาพที่ 48 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ



(a) (b)

ภาพที่ 49 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม
(b) รูปแบบโครงข่ายที่ได้จากโปรแกรม

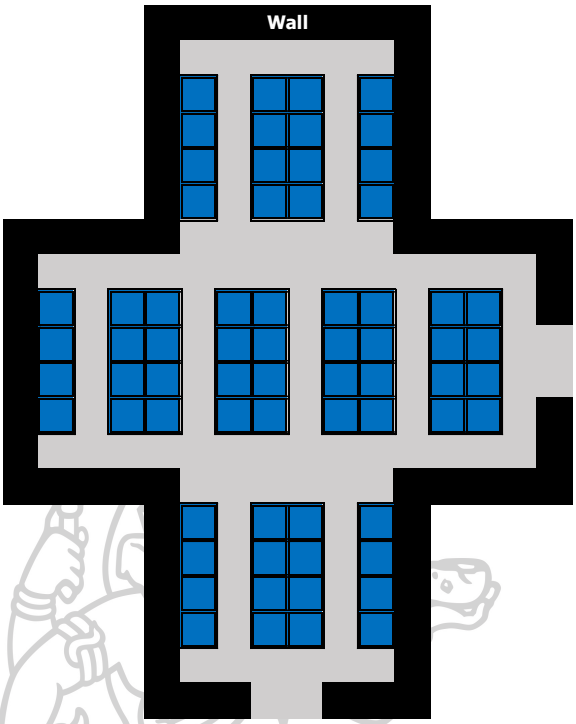
ตารางที่ 8 ระยะทางระหว่างโหนดของคลังสินค้าที่มีรูปแบบกำแพงด้านหนึ่งเป็นแบบขั้นบันได

	1	2	3	4	5	6	7	8	9	10	11	....	66
1	0	2										**....**	
2	2	0	2									....	
3		2	0	2								....	
4			2	0	2							....	
5				2	0	2						....	
6					2	0	2					....	
7						2	0	2				....	
8							2	0	2			....	
9								2	0	2		....	
10									2	0	2	....	
11										2	0	....	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
66												....	0

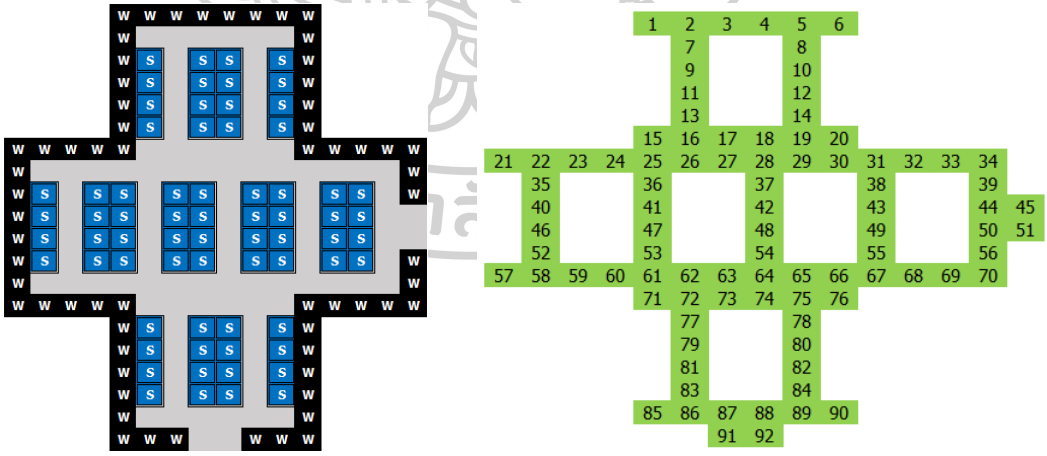
ตารางที่ 9 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่มีรูปแบบกำแพงด้านหนึ่งเป็นแบบขั้นบันได

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	....	**66**
1	0	2	4	6	8	10	12	14	16	18	20	**....**	13
2	2	0	2	4	6	8	10	12	14	16	18	....	11
3	4	2	0	2	4	6	8	10	12	14	16	....	13
4	6	4	2	0	2	4	6	8	10	12	14	....	15
5	8	6	4	2	0	2	4	6	8	10	12	....	17
6	10	8	6	4	2	0	2	4	6	8	10	....	19
7	12	10	8	6	4	2	0	2	4	6	8	....	21
8	14	12	10	8	6	4	2	0	2	4	6	....	23
9	16	14	12	10	8	6	4	2	0	2	4	....	25
10	18	16	14	12	10	8	6	4	2	0	2	....	27
11	20	18	16	14	12	10	8	6	4	2	0	....	29
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
66	13	11	13	15	17	19	21	23	25	27	29	....	0

3. **คลังสินค้าที่มีรูปแบบเป็นเครื่องหมายบวก**



ภาพที่ 50 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ



(a) (b)

ภาพที่ 51 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม

(b) รูปแบบโครงข่ายที่ได้จากโปรแกรม

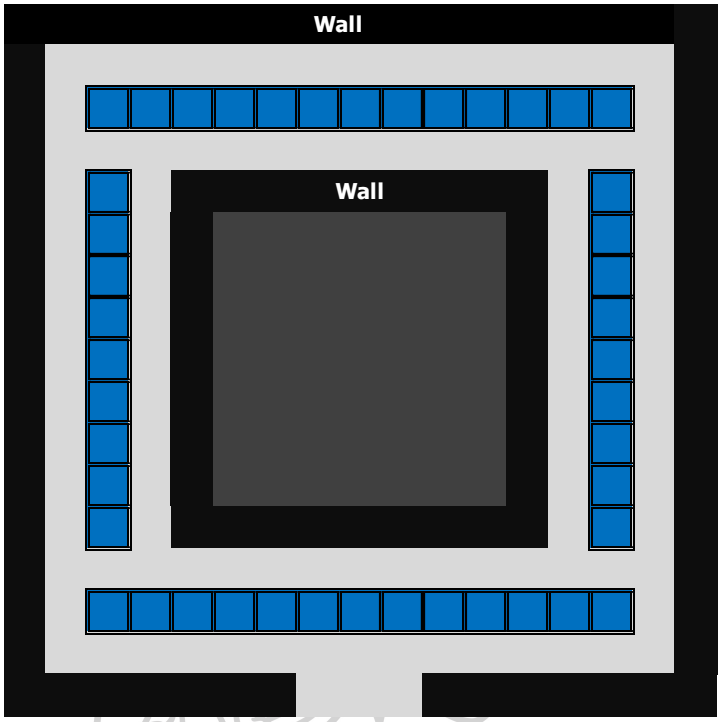
ตารางที่ 10 ระยะทางระหว่างโหนดของคลังสินค้าที่มีรูปแบบเป็นเครื่องหมายบวก

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	....	**92**
1	0	2										**....**	
2	2	0	2				1					....	
3		2	0	2								....	
4			2	0	2							....	
5				2	0	2		1				....	
6					2	0						....	
7		1					0		1			**....**	
8					1			0		1		....	
9							1		0		1	....	
10								1		0		....	
11									1		0	....	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
92												....	0

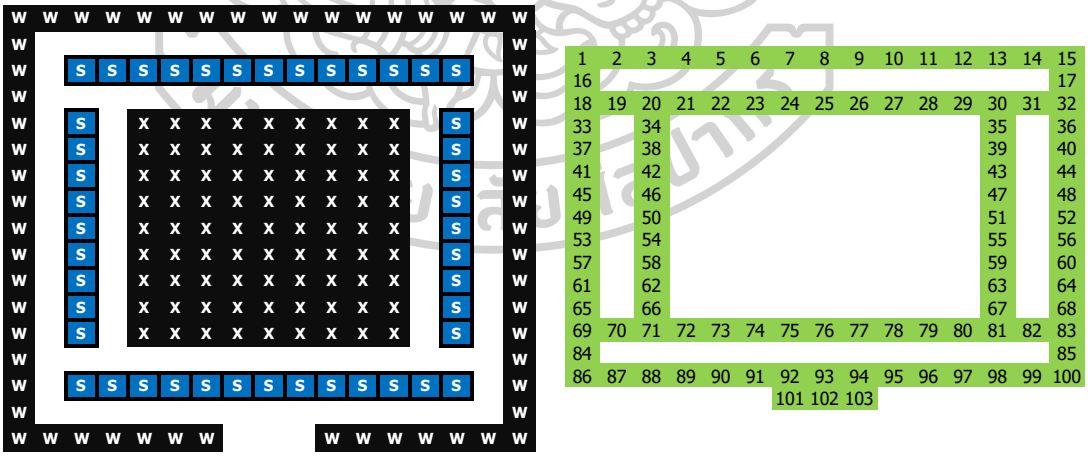
ตารางที่ 11 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่มีรูปแบบเป็นเครื่องหมายบวก

	1	2	3	4	5	6	7	8	9	10	11	....	103
1	0	2	4	6	8	10	12	14	16	18	20	**....**	29
2	2	0	2	4	6	8	10	12	14	16	18	....	27
3	4	2	0	2	4	6	8	10	12	14	16	....	25
4	6	4	2	0	2	4	6	8	10	12	14	....	23
5	8	6	4	2	0	2	4	6	8	10	12	....	21
6	10	8	6	4	2	0	2	4	6	8	10	....	19
7	12	10	8	6	4	2	0	2	4	6	8	....	17
8	14	12	10	8	6	4	2	0	2	4	6	....	15
9	16	14	12	10	8	6	4	2	0	2	4	....	13
10	18	16	14	12	10	8	6	4	2	0	2	....	11
11	20	18	16	14	12	10	8	6	4	2	0	....	9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
103	29	27	25	23	21	19	17	15	13	11	9	....	0

4. คลังสินค้าที่มีออฟฟิศ หรือห้องอยู่ตรงกลางคลังสินค้า



ภาพที่ 52 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ



ภาพที่ 53 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม
(b) รูปแบบโครงข่ายที่ได้จากโปรแกรม

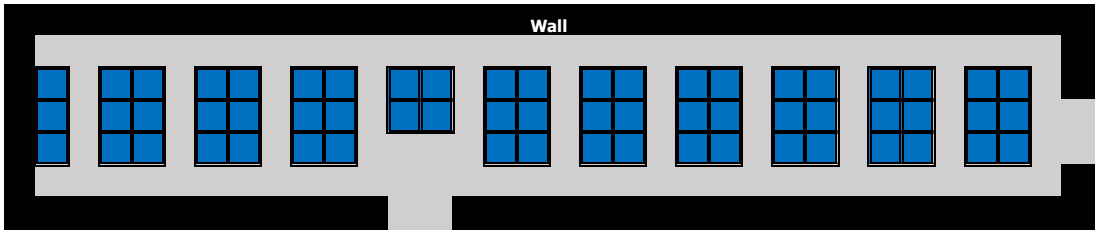
ตารางที่ 12 ระยะทางระหว่างโหนดของคลังสินค้าที่มีออฟฟิศ หรือห้องอยู่ตรงกลางคลังสินค้า

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**....**	**103**
1	0	2										....	
2	2	0	2									....	
3		2	0	2								....	
4			2	0	2							....	
5				2	0	2						....	
6					2	0	2					....	
7						2	0	2				....	
8							2	0	2			....	
9								2	0	2		....	
10									2	0	2	....	
11										2	0	....	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
103												2	0

ตารางที่ 13 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่มีออฟฟิศ หรือห้องอยู่ตรงกลางคลังสินค้า

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**....**	**103**
1	0	2	4	6	8	10	12	14	16	18	20	....	31
2	2	0	2	4	6	8	10	12	14	16	18	....	33
3	4	2	0	2	4	6	8	10	12	14	16	....	35
4	6	4	2	0	2	4	6	8	10	12	14	....	37
5	8	6	4	2	0	2	4	6	8	10	12	....	39
6	10	8	6	4	2	0	2	4	6	8	10	....	41
7	12	10	8	6	4	2	0	2	4	6	8	....	43
8	14	12	10	8	6	4	2	0	2	4	8	....	41
9	16	14	12	10	8	6	4	2	0	2	4	....	39
10	18	16	14	12	10	8	6	4	2	0	2	....	37
11	20	18	16	14	12	10	8	6	4	2	0	....	35
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
103	31	33	35	37	39	41	43	41	39	37	35	....	0

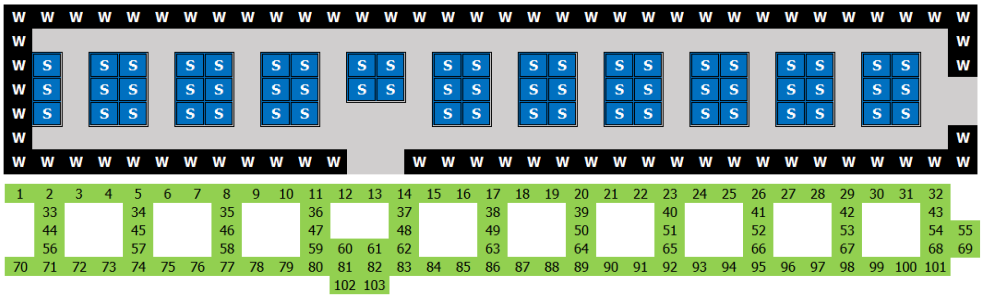
5. คลังสินค้าที่เป็นสี่เหลี่ยมผืนผ้า และมีทางเข้าออกสองด้าน



ภาพที่ 54 รูปแบบคลังสินค้าที่ใช้ในการทดสอบ

(a)

(b)



ภาพที่ 55 (a) รูปแบบคลังสินค้าที่วาดลงบนโปรแกรม

(b) รูปแบบโครงข่ายที่ได้จากโปรแกรม

ตารางที่ 14 ระยะทางระหว่างโหนดของคลังสินค้าที่เป็นสี่เหลี่ยมผืนผ้า และมีทางเข้าออกสองด้าน

	1	2	3	4	5	6	7	8	9	10	11	....	103
1	0	2										....	
2	2	0	2									....	
3		2	0	2								....	
4			2	0	2							....	
5				2	0	2						....	
6					2	0	2					....	
7						2	0	2				....	
8							2	0	2			....	
9								2	0	2		....	
10								2	0	2		....	
11										2	0	....	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
103												....	0

ตารางที่ 15 ระยะทางวิถีสั้นสุดทุกคู่โหนดของคลังสินค้าที่เป็นสี่เหลี่ยมผืนผ้า และมีทางเข้าออกสองด้าน

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**....**	**103**
1	0	2	4	6	8	10	12	14	16	18	20	....	29
2	2	0	2	4	6	8	10	12	14	16	18	....	27
3	4	2	0	2	4	6	8	10	12	14	16	....	25
4	6	4	2	0	2	4	6	8	10	12	14	....	23
5	8	6	4	2	0	2	4	6	8	10	12	....	21
6	10	8	6	4	2	0	2	4	6	8	10	....	19
7	12	10	8	6	4	2	0	2	4	6	8	....	17
8	14	12	10	8	6	4	2	0	2	4	6	....	15
9	16	14	12	10	8	6	4	2	0	2	4	....	13
10	18	16	14	12	10	8	6	4	2	0	2	....	11
11	20	18	16	14	12	10	8	6	4	2	0	....	9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
103	29	27	25	23	21	19	17	15	13	11	9	....	0

มหาวิทยาลัยศิลปากร

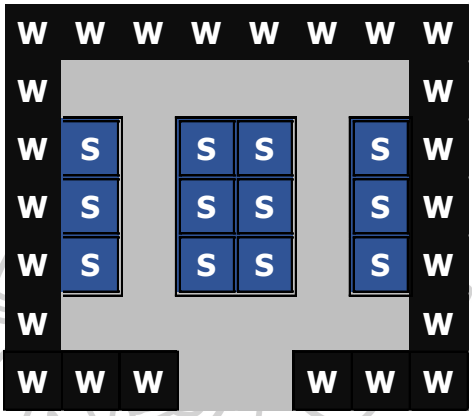
ภาคผนวก ข

รูปแบบคลังสินค้าที่ใช้ในการทดสอบระยะเวลาในการทำงานของโปรแกรม

รูปแบบคลังสินค้าที่ใช้ในการทดสอบระยะเวลาในการทำงานของโปรแกรม

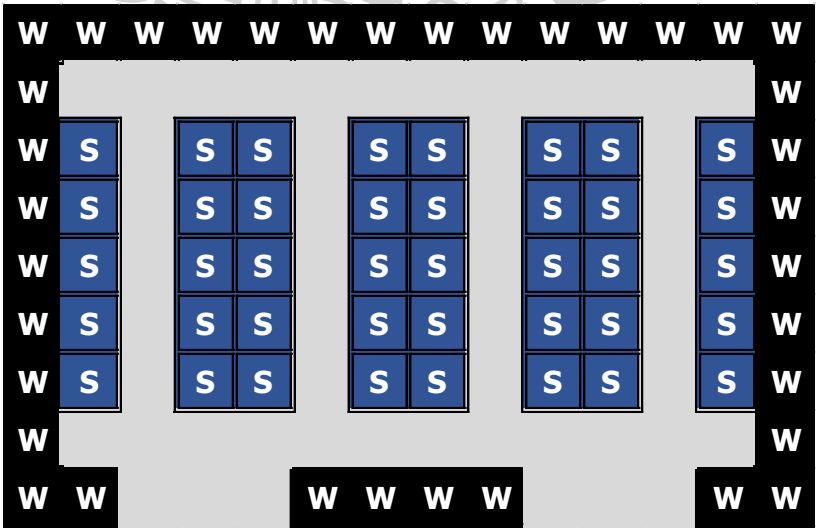
ในการทดสอบระยะเวลาในการทำงานของโปรแกรมจะทดสอบผ่านรูปแบบคลังสินค้าที่มีจำนวนโหนดของเส้นทางเดินแตกต่างกันโดยกำหนดขอบเขตของรูปแบบคลังสินค้าแสดงดังตารางที่ 2 และใช้ค่าตัวแปร k_x และ k_y

1. **รูปแบบคลังสินค้าที่มีจำนวนโหนด 20 โหนด**



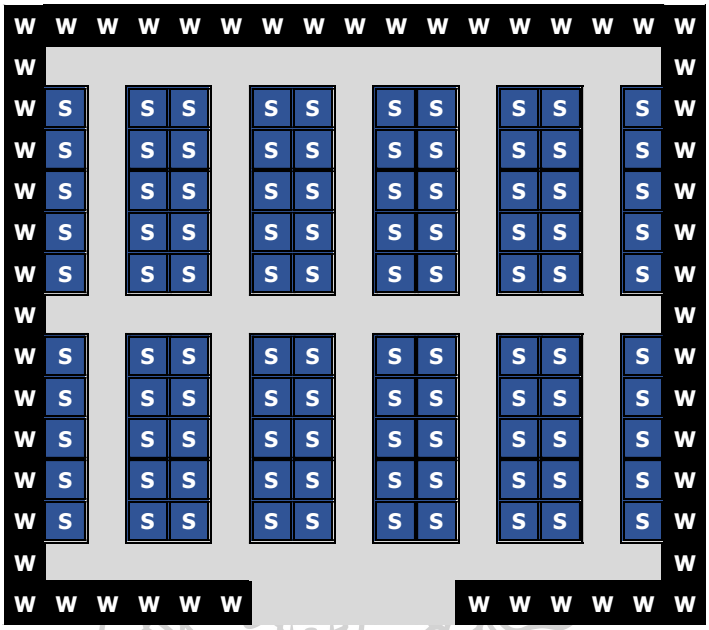
ภาพที่ 56 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 20 โหนด

2. **รูปแบบคลังสินค้าที่มีจำนวนโหนด 50 โหนด**



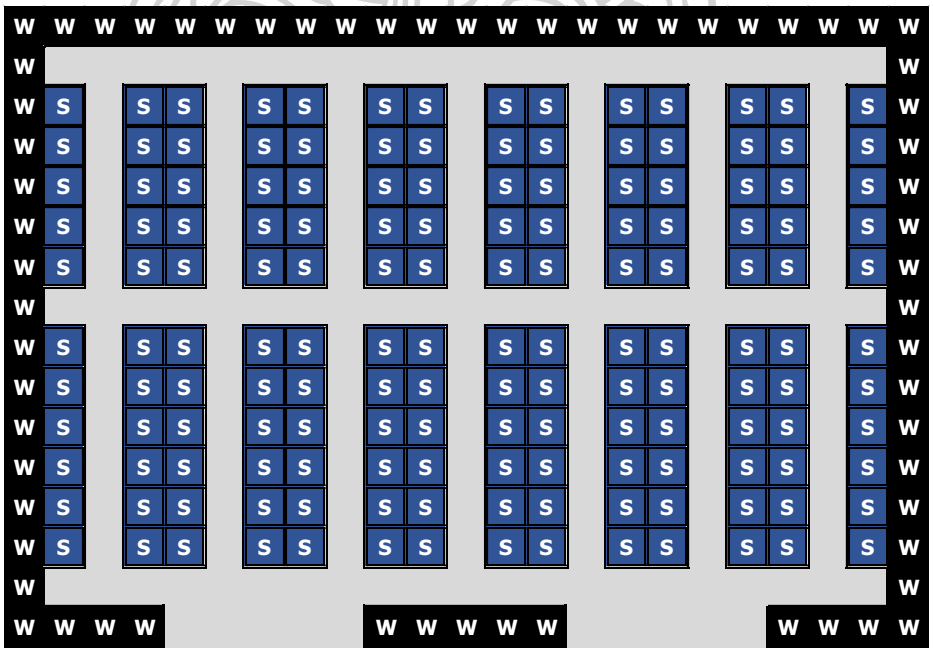
ภาพที่ 57 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 50 โหนด

3. **รูปแบบคลังสินค้าที่มีจำนวนโหนด 100 โหนด**



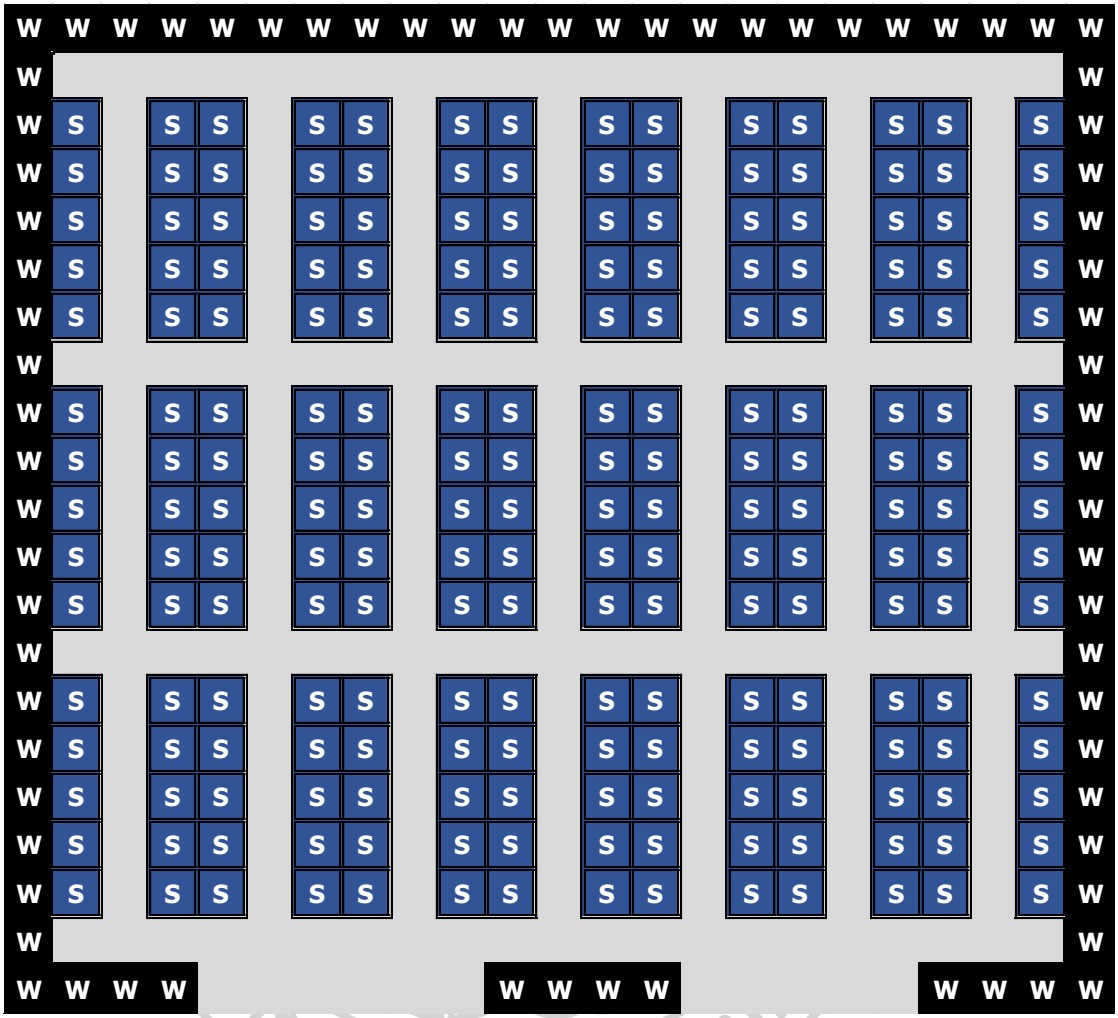
ภาพที่ 58 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 100 โหนด

4. **รูปแบบคลังสินค้าที่มีจำนวนโหนด 150 โหนด**



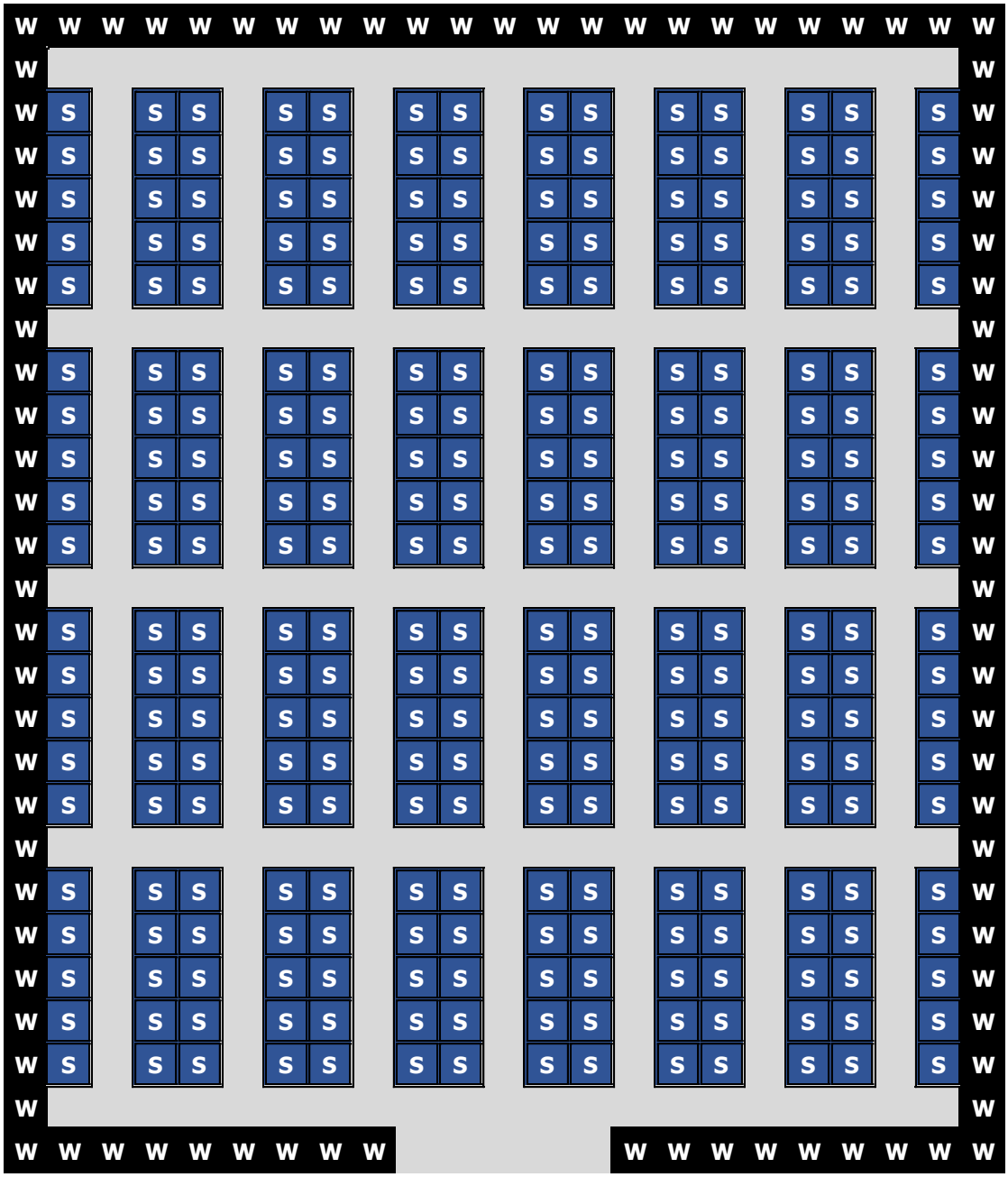
ภาพที่ 59 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 150 โหนด

5. รูปแบบคลังสินค้าที่มีจำนวนโหนด 200 โหนด



ภาพที่ 60 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 200 โหนด

6. **รูปแบบคลังสินค้าที่มีจำนวนโหนด 250 โหนด**

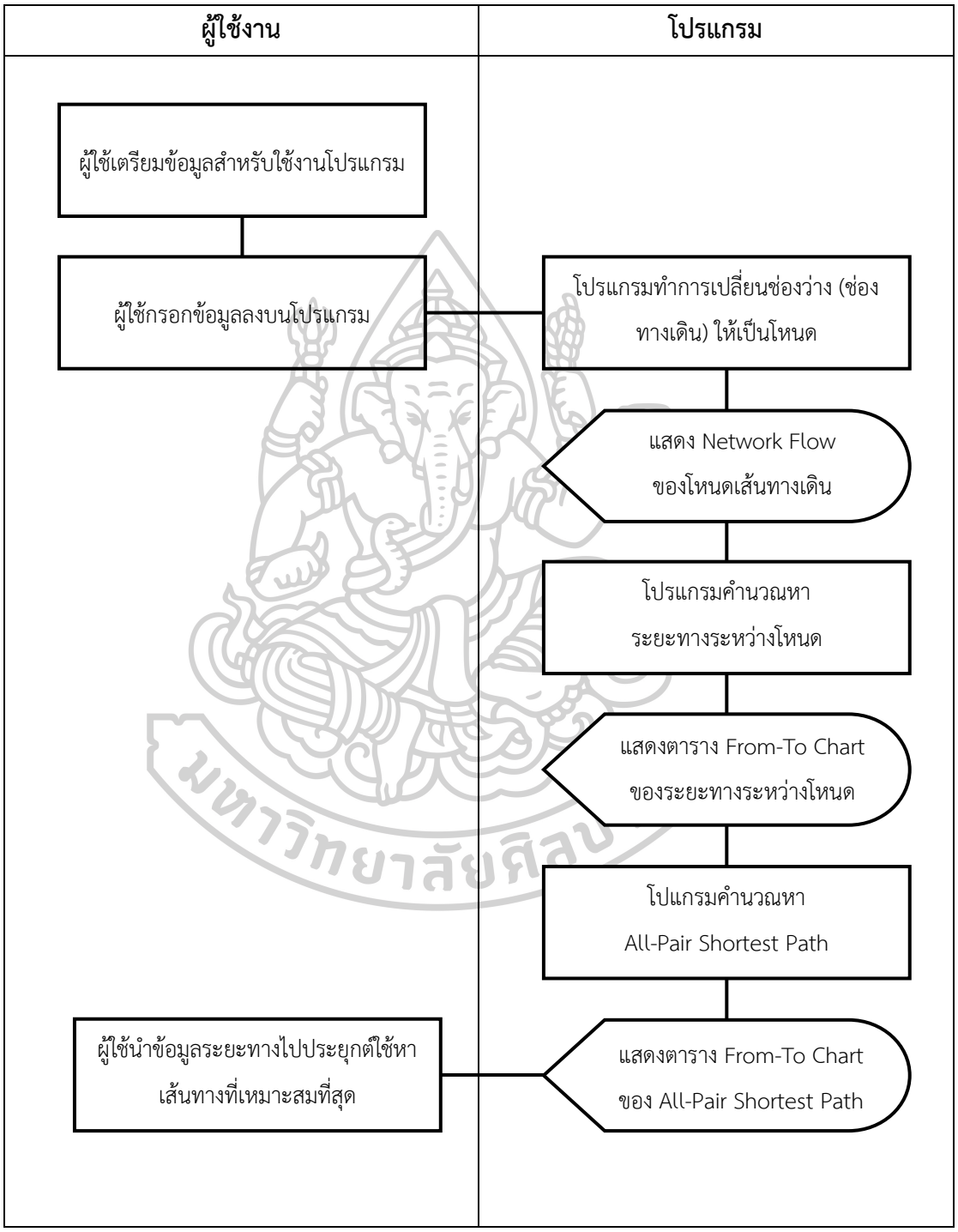


ภาพที่ 61 รูปแบบคลังสินค้าที่มีจำนวนช่องทางเดิน 250 โหนด

ภาคผนวก ค
คู่มือการใช้งานโปรแกรม

คู่มือการใช้งานโปรแกรม

ตารางที่ 16 แสดงขั้นตอนการทำงานของโปรแกรมสำหรับผู้ใช้งาน

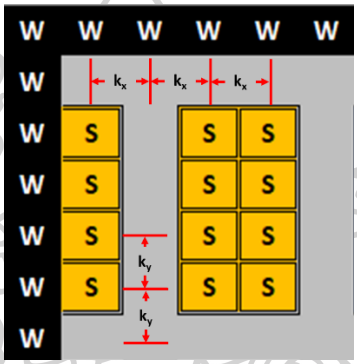


1. ผู้ใช้เตรียมข้อมูลสำหรับการใช้งานโปรแกรม

1) ผู้ใช้เตรียมข้อมูลของระยะทางจริงภายในคลังสินค้า k_x และ k_y

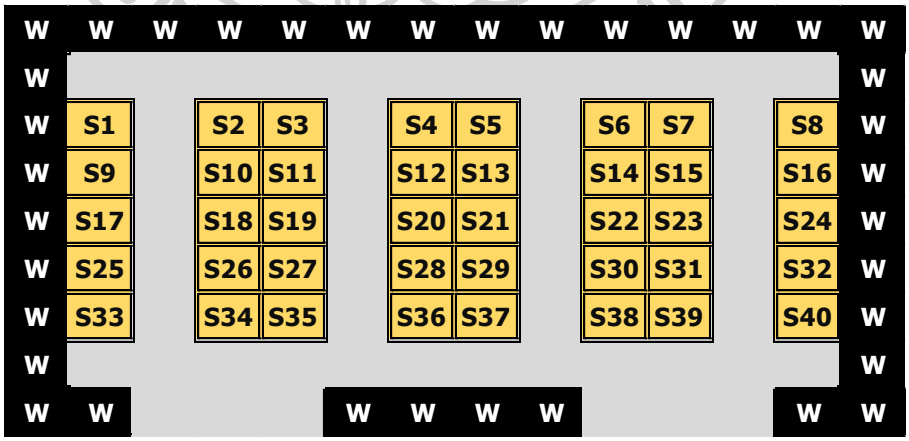
	A	B	C	D	E	F	G	H
1	Variable	Vlues	Unit	Comment				
2	k_x	1	m	ระยะหาระหว่างโหนดในแนวแกน x				
3	k_y	1	m	ระยะหาระหว่างโหนดในแนวแกน y				
4	Number of Node	150	Node	จำนวนโหนดทางเดิน				
5								
6	Build		Run			Clear		
7								

ภาพที่ 62 แสดงแบบฟอร์มสำหรับกรอกค่า k_x และ k_y



ภาพที่ 63 แสดงรูปแบบการวัดคา k_x และ k_y

2) ผู้ใช้เตรียมข้อมูลของสินค้าเพื่อกรอกข้อมูลให้ตรงกับโหนดที่ตั้ง โดยกำหนดรหัสให้กับสินค้าดังแสดงในรูปที่ 64 กำหนดให้รหัสแทนด้วย S1 ถึง S40



ภาพที่ 64 ภาพคลังสินค้าที่ระบุรหัสสินค้าระหว่างช่องทางเดิน

2. ผู้ใช้งานกรอกข้อมูลลงบนโปรแกรม

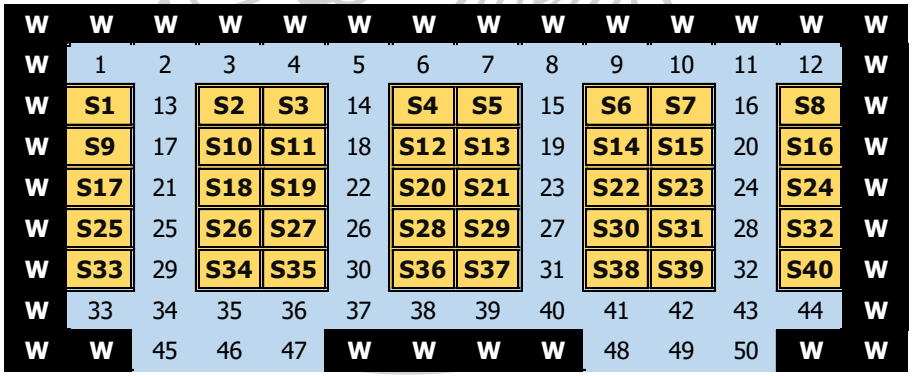
1) ผู้ใช้คลิกปุ่ม Build เพื่อสร้าง Worksheets สำหรับวาดรูปแบบคลังสินค้า
2) ผู้ใช้วาดรูปแบบคลังสินค้าลงบน Worksheet “Warehouse”
3) ผู้ใช้กรอกข้อมูลของ kx และ ky Worksheet “Data”
4) ผู้ใช้ทำการคลิกปุ่ม Run เพื่อเริ่มการทำงานของโปรแกรม

3. โปรแกรมทำงาน

1) โปรแกรมเปลี่ยนช่องทางเดินภายในคลังสินค้าให้เป็นโหนด (รูปแบบ Network Flow)
2) โปรแกรมคำนวณหาระยะทางระหว่างโหนด และแสดงผลออกมาในรูปแบบของตาราง From-To Chart ใน Worksheet “All-Pair”
3) โปรแกรมคำนวณหาระยะทางจากทุกตำแหน่งไปยังทุกตำแหน่งภายในคลังสินค้า และแสดงผลออกมาในรูปแบบของตาราง From-To Chart ใน Worksheet “All-Pair2”

4. ผู้ใช้นำข้อมูลระยะทางไปประยุกต์ใช้หาเส้นทางที่เหมาะสมที่สุด

1) ผู้ใช้สามารถเปรียบเทียบข้อมูลกับผลลัพธ์ที่ได้ดังภาพที่ 65 และตารางที่ 17



ภาพที่ 65 ภาพคลังสินค้าที่ระบุรหัสสินค้าระหว่างโหนดช่องทางเดิน

ตารางที่ 17 ตารางตัวอย่างข้อมูลระยะทางกับชั้นวางสินค้า

Node	ตำแหน่งสินค้า	
	ซ้ายมือ	**ขวามือ**
13	S1	S2
14	S3	S4
15	S5	S6

2) นำข้อมูลภายในตาราง From-To Chart ใน Worksheet “All-Pair” ไปประยุกต์ใช้ในการหาชั้นวางสินค้าระหว่างโหนดทางด้านซ้ายและด้านขวา ว่าเป็นสินค้าประเภทไหน
3) สามารถนำขั้นตอนวิธีดังกล่าวไปประยุกต์ใช้ในขั้นตอนวิธีการหาเส้นทางที่เหมาะสมที่สุด

ประวัติผู้เขียน

ชื่อ-สกุล	ตฤณ ปฐมนิธิภิญโญ
วัน เดือน ปี เกิด	7 สิงหาคม 2535
สถานที่เกิด	ลำปาง
วุฒิการศึกษา	มหาวิยาลัยศิลปากร
ที่อยู่ปัจจุบัน	689 หมู่ที่ 3 ตำบลต้นธงชัย อำเภอเมือง จังหวัดลำปาง 52000
ผลงานตีพิมพ์	1) Low acid bio-oil from Para rubber seeds produced via catalytic pyrolysis with V-modified silica catalyst (old name: Anucha Khadthiphong) 2) Platinum-Free Counter Electrodes Comprised of Polypyrrole-Graphene Composite (old name: Anucha Khadthiphong) 3) The fabrication of graphene-polypyrrole composite for application with dye-sensitized solar cells (old name: Anucha Khadthiphong) 4) Catalytic Pyrolysis of Water Hyacinth with Rice Husk-derived Silica Modified by Vanadium (old name: Anucha Khadthiphong)

มหาวิทยาลัยศิลปากร