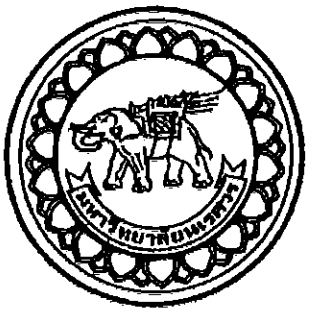


การจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

SOLVING THE QUAY CRANE SCHEDULING PROBLEM BY SIMULATED ANNEALING

นางสาวหทัยรัตน์ ธีระกาญจน์ รหัส 48210630

นางสาวศิริพร ชารีพร รหัส 50383004

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 24 มิ.ย. 2554
เลขทะเบียน 1'5513996
เลขเรียกหนังสือ ม5.
มหาวิทยาลัยนเรศวร ม1361ก
2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2553

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	การจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง
ผู้ดำเนินโครงงาน	นางสาวหทัยรัตน์ ธีระกาญจน์ รหัส 48210630
	นางสาวศิริพร ชารีพร รหัส 50383004
ที่ปรึกษาโครงงาน	ดร.ขวัญนิธิ คำเมือง
สาขาวิชา	วิศวกรรมอุตสาหการ
ภาควิชา	วิศวกรรมอุตสาหการ
ปีการศึกษา	2553

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ

.....ที่ปรึกษาโครงงาน
(ดร.ขวัญนิธิ คำเมือง)

.....ประธานกรรมการ
(ผศ.ดร.ภูพงษ์ พงษ์เจริญ)

.....กรรมการ
(อาจารย์ศรีสัจจา วิทยศักดิ์)

.....กรรมการ
(ดร.สุธนิตย์ พุทธพนม)

ชื่อหัวข้อโครงงาน	การจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง
ผู้ดำเนินโครงงาน	นางสาวหทัยรัตน์ ธีระกาญจน์ รหัส 48210630
	นางสาวศิริพร ชารีพร รหัส 50383004
ที่ปรึกษาโครงงาน	ดร.ขวัญนิธิ คำเมือง
สาขาวิชา	วิศวกรรมอุตสาหการ
ภาควิชา	วิศวกรรมอุตสาหการ
ปีการศึกษา	2553

.....

บทคัดย่อ

โครงงานวิจัยนี้จะเป็นการสร้างวิธีการในการจัดลำดับการทำงานของเครน โดยใช้วิธีการอบอ่อนจำลองเข้ามาช่วยในการจัดลำดับการทำงาน รวมทั้งได้มีการพัฒนาและสร้างโปรแกรมเพื่อใช้ในการแก้ไขปัญหา ซึ่งจะใช้การเขียนโปรแกรม โดยใช้ภาษา Visual Basic for Applications บน Microsoft Excel ทั้งนี้เพื่อกำหนดรูปแบบการทำงานของเครนที่จะทำให้ใช้เวลาในการทำงานทั้งหมดน้อยที่สุดหรือค่า Makespan

ผู้วิจัยได้ทำการศึกษาถึงผลจากการทำ Neighbourhood search หรือการหาคำตอบใหม่ในย่านคำตอบเดิมตามวิธีการอบอ่อนจำลองในหลายๆหลักการ นอกจากนี้ยังได้ศึกษาถึงการเปลี่ยนแปลงตัวแปรต่างๆในการอบอ่อนจำลองอีกด้วย ซึ่งในการทดสอบประสิทธิภาพของการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองนี้ ได้ทำการทดสอบกับปัญหาขนาดต่างๆ และมีการเปรียบเทียบผลลัพธ์ที่ได้กับงานวิจัยของผู้วิจัยเองและงานวิจัยของผู้อื่น จากการศึกษาสามารถสรุปได้ว่า ความแตกต่างกันของ Neighbourhood search มีผลต่อค่า Makespan ของปัญหาการจัดลำดับการทำงานของเครน และมีผลโดยตรงต่อค่าเวลาที่ใช้ในการประมวลผล

กิตติกรรมประกาศ

ไม่มีงานใดที่สำเร็จได้โดยที่ไม่มีอุปสรรค และอุปสรรคนั้นก็นำมาซึ่งความรู้และประสบการณ์ โครงงานวิจัยนี้ก็เช่นกันที่มีอุปสรรคและความยากลำบากเกิดขึ้นมากมายระหว่างการดำเนินงาน แต่กระนั้น งานวิจัยฉบับนี้ก็สามารถสำเร็จลุล่วงลงได้ด้วยดี จากคำปรึกษาที่ดีตลอดมาจาก ดร.ขวัญนิธิ คำเมือง ผู้เป็นอาจารย์ที่ปรึกษาโครงงาน จึงต้องขอขอบพระคุณที่กรุณาให้คำปรึกษา ชี้แนะแก้ไข และตรวจสอบ โครงงานวิจัยครั้งนี้ให้ผ่านไปอย่างราบรื่น นอกจากนั้นยังต้องขอขอบพระคุณคณะกรรมการทั้งสามท่าน อันประกอบด้วย ผศ.ดร ภูพงษ์ พงษ์เจริญ อ.ศรีสัจจา วิทยศักดิ์ และ ดร.สุธนิตย์ พุทธพนม ที่กรุณาให้ความช่วยเหลือในการแก้ไขและให้คำแนะนำที่มีประโยชน์ ที่มีส่วนทำให้งานวิจัยครั้งนี้มีคุณค่ามากยิ่งขึ้น จนสามารถออกมาเป็นปริญญานิพนธ์ฉบับนี้

อีกสิ่งหนึ่งที่จะลืมไม่ได้ก็คือ ความกรุณาจาก บิดา มารดา พี่และน้อง ที่ได้สนับสนุนในเรื่องทุนการศึกษา และ ให้กำลังใจที่ดีในทุกๆเรื่อง จนทำให้ผู้วิจัยมีกำลังใจและกำลังกายที่ดีตลอดมา ทางผู้วิจัยจึงขอขอบพระคุณเป็นอย่างสูง ณ ที่นี้ด้วย

คณะผู้ดำเนินโครงงานวิศวกรรม
นางสาวหทัยรัตน์ ธีระกาญจน์
นางสาวศิริพร ชารีพร

มีนาคม 2554

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช

บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 เกณฑ์ชี้วัดผลงาน 2
1.4 เกณฑ์ชี้วัดผลสำเร็จ 2
1.5 ขอบเขตในการดำเนินโครงงาน 2
1.6 สถานที่ในการดำเนินโครงงาน 2
1.7 ระยะเวลาในการดำเนินโครงงาน 2
1.8 ขั้นตอนและแผนการดำเนินโครงงาน 3

บทที่ 2 หลักการและทฤษฎีเบื้องต้น 4
2.1 เรือ ท่าเรือ เครนและการปัญหาในการจัดลำดับการทำงานของเครน 4
2.2 วิธีการอบอ่อนจำลอง (Simulated Annealing) 6
2.3 โปรแกรม Visual Basic for Applications บน Microsoft Excel 8

บทที่ 3 วิธีการดำเนินโครงงาน 10
3.1 การกำหนดปัญหาและศึกษาปัญหาการจัดลำดับการทำงานของเครน 10
3.2 การศึกษาแบบจำลองเชิงคณิตศาสตร์ 14
3.3 ออกแบบวิธีการอบอ่อนจำลองโดยเขียนโปรแกรมแบบจำลองเชิงคณิตศาสตร์บน โปรแกรม Visual Basic for Applications บน Microsoft Excel 15
3.4 การทดสอบโปรแกรมการปัญหาโดยวิธีการอบอ่อนจำลอง 58
3.5 จัดทำรายงานและสรุปผล 58
3.6 การนำเสนอโครงงาน 58

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลองและการวิเคราะห์ 59
4.1 โปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 59
4.2 ผลการทดสอบโปรแกรมการแก้ปัญหาด้วยวิธีการอบอ่อน 66
4.3 การเปรียบเทียบผลลัพธ์ที่ได้จากการทดสอบโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 79
4.4 การวิเคราะห์ผลการทดลอง 102

บทที่ 5 บทสรุปและข้อเสนอแนะ 106
5.1 บทสรุป 106
5.2 ข้อเสนอแนะ 106

เอกสารอ้างอิง 107

ภาคผนวก ก คู่มือโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 108
ภาคผนวก ข Source Code ของโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 122

ประวัติผู้ดำเนินโครงงาน 181

สารบัญตาราง

ตารางที่ หน้า

1.1 แสดงขั้นตอนและแผนการดำเนินโครงงาน 3
3.1 แสดงข้อมูลการทำงานของเครนแต่ละงาน 11
4.1 แสดงปัญหาแต่ละปัญหาเพื่อการทดสอบประสิทธิภาพของโปรแกรม 67
4.2 แสดงผลลัพธ์ตามลักษณะที่เคยมีผู้วิจัยมาแล้ว 68
4.3 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 1 71
4.4 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 2 71
4.5 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 3 72
4.6 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 4 72
4.7 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 5 73
4.8 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 6 73
4.9 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 7 74
4.10 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 8 74
4.11 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 9 75
4.12 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 10 75
4.13 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 11 76
4.14 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 12 76
4.15 แสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเองเมื่อไม่มีการเปลี่ยนแปลงตัวแปร 77
4.16 แสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเองเมื่อมีการเปลี่ยนแปลงตัวแปรครั้งที่ 1 78
4.17 แสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเองเมื่อมีการเปลี่ยนแปลงตัวแปรครั้งที่ 2 78
4.18 แสดงการเปรียบเทียบรายละเอียดของเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผลการจัดลำดับการทำงานของเครน ของผู้วิจัยเองและของผู้วิจัยอื่น 82
4.19 แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่เครนทำงานเสร็จสิ้นกับงานวิจัยอื่น 83
4.20 แสดงการเปรียบเทียบค่า Makespan ที่ดีที่สุดกับผลลัพธ์ของผู้วิจัยเป็นเปอร์เซ็นต์ 84
4.21 แสดงการเปรียบเทียบผลลัพธ์ของเวลาในการประมวลผลกับงานวิจัยอื่น 85
4.22 แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่เครนทำงานเสร็จสิ้นเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน 87
4.23 แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่ใช้ในการประมวลผลเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน 91
4.24 แสดงการเปรียบเทียบระหว่างผลลัพธ์ของเวลาที่เครนทำการงานเสร็จสิ้นของโปรแกรมที่ผู้วิจัยออกแบบเองกับโปรแกรมที่มีผู้วิจัยอื่นสร้างขึ้น 99
4.25 แสดงการเปรียบเทียบระหว่างผลลัพธ์ของเวลาที่ใช้ในการประมวลผล ของโปรแกรมที่ผู้วิจัยออกแบบเองกับโปรแกรมที่มีผู้วิจัยอื่นสร้างขึ้น 100
4.26 แสดงการเปรียบเทียบระหว่างผลลัพธ์ของเวลาที่เครนทำการงานเสร็จสิ้นของโปรแกรมที่ผู้วิจัยออกแบบเองกับโปรแกรมที่มีผู้วิจัยอื่นสร้างขึ้นเป็นเปอร์เซ็นต์ 101

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

4.27 แสดงการเปรียบเทียบระหว่างผลลัพธ์ของเวลาในการประมวลผล ของโปรแกรมที่ผู้วิจัยออกแบบเองกับโปรแกรมที่มีผู้วิจัยอื่นสร้างขึ้นเป็นเปอร์เซ็นต์ 102

มหาวิทยาลัยนเรศวร

สารบัญรูป

รูปที่ หน้า

2.1 แสดงความสัมพันธ์ระหว่างการเทียบท่าของท่าเรือ การกำหนดการทำงานของ เครนเข้ากับเรือและการจัดลำดับการทำงานของเครน 4

2.2 แสดงความสัมพันธ์ระหว่างปัญหาการเทียบท่าของท่าเรือ ปัญหาการกำหนดการทำงานของเครนเข้ากับเรือและปัญหาการจัดลำดับการทำงานของเครน 5

2.3 แสดงการสุ่มค่าค้นหาคำตอบเพื่อกระโดดและหลุดออกจาก Local Optimum 7

3.1 การทำงานของเครนที่ T = 0 ณ ตำแหน่งต่างๆ 11

3.2 การทำงานของเครนที่ T = 2 ณ ตำแหน่งต่างๆ 12

3.3 การทำงานของเครนที่ T = 6 ณ ตำแหน่งต่างๆ 12

3.4 การทำงานของเครนที่ T = 8 ณ ตำแหน่งต่างๆ 13

3.5 การทำงานของเครนที่ T = 9 ณ ตำแหน่งต่างๆ 13

3.6 การทำงานของเครนที่ T = 10 ณ ตำแหน่งต่างๆ 13

3.7 การทำงานของเครนที่ T = 12 ณ ตำแหน่งต่างๆ 14

3.8 การทำงานของเครนที่ T = 15 ณ ตำแหน่งต่างๆ 14

3.9 ตัวอย่างการสุ่มคำตอบแรกในกระบวนการอบอ่อนจำลอง 15

3.10 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบที่ได้จากการสุ่มครั้งแรก โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 16

3.11 ตัวอย่างการหาคำตอบใหม่โดยหลักการเลื่อนตำแหน่ง 17

3.12 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งแรกของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 45 นาที 18

3.13 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งสุดท้ายของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 57 นาที 19

3.14 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางขวามือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 45 นาที 20

3.15 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางซ้ายมือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 21

3.16 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางขวามือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 40 นาที 22

3.17 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางซ้ายมือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 23

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.18 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางขวามือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 38 นาที 24

3.19 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางซ้ายมือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 50 นาที 25

3.20 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางขวามือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 26

3.21 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางซ้ายมือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 27

3.22 ตัวอย่างการหาคำตอบใหม่โดยหลักการคงที่หนึ่งตำแหน่ง 28

3.23 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งแรกของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 45 นาที 29

3.24 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งสุดท้ายของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 48 นาที 30

3.25 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางขวามือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 62 นาที 31

3.26 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางซ้ายมือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 32

3.27 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางขวามือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 40 นาที 33

3.28 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางซ้ายมือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 57 นาที 34

3.29 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางขวามือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 38 นาที 35

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.30 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางซ้ายมือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 47 นาที 36

3.31 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางขวามือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 37

3.32 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางซ้ายมือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 60 นาที 38

3.33 ตัวอย่างการหาคำตอบใหม่โดยหลักการสลับที่ 39

3.34 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งแรกของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 40

3.35 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งสุดท้ายของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 41

3.36 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางขวามือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 60 นาที 42

3.37 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางซ้ายมือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 43

3.38 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางขวามือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 55 นาที 44

3.39 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางซ้ายมือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 45

3.40 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางขวามือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 47 นาที 46

3.41 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางซ้ายมือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 50 นาที 47

3.42 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางขวามือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 48

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.43 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางซ้ายมือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 60 นาที 49

3.44 ตัวอย่างการหาคำตอบใหม่โดยหลักการพิจารณาที่ 2 ตำแหน่ง 50

3.45 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งแรกและตำแหน่งที่สอง ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 62 นาที 51

3.46 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งสุดท้ายและตำแหน่งก่อนสุดท้าย ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที 52

3.47 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 และ 3 ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 40 นาที 53

3.48 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 และ 4 ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 38 นาที 54

3.49 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 และ 5 ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 50 นาที 55

3.50 Flow Chart แสดงวิธีการและขั้นตอนการอบอ่อนจำลอง 57

4.1 แสดงส่วนต้อนรับเข้าสู่โปรแกรมของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 60

4.2 แสดงขั้นตอนที่ 1 ระบุค่าข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 61

4.3 แสดงขั้นตอนที่ 2 เลือกวิธีการอบอ่อน ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 62

4.4 แสดงขั้นตอนที่ 3 ระบุเวลางาน ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 63

4.5 แสดงขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 64

4.6 แสดงการแก้ไขข้อมูลใน ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 64

4.7 แสดงการแก้ไขเวลางานใน ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 65

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.8 แสดงขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 65

4.9 แสดงแถบความก้าวหน้าเมื่อกดประมวลผลใน ขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 66

4.10 แผนภูมิแสดงการเปรียบเทียบระหว่างหลักการในการทำ Neighbourhood search ของเวลาที่เครนทำงานเสร็จสิ้นโดยเฉลี่ย 80

4.11 แผนภูมิแสดงการเปรียบเทียบระหว่างหลักการในการทำ Neighbourhood search ของเวลาที่เครนทำงานเสร็จสิ้นน้อยที่สุด 80

4.12 แผนภูมิแสดงการเปรียบเทียบระหว่างหลักการในการทำ Neighbourhood search ของเวลาที่ใช้ในการประมวลผลโดยเฉลี่ย 81

4.13 แผนภูมิแสดงการเปรียบเทียบระหว่างหลักการในการทำ Neighbourhood search ของเวลาที่ใช้ในการประมวลผลที่น้อยที่สุด 81

4.14 แสดงแผนภูมิการเปรียบเทียบเวลาที่เครนทำงานเสร็จสิ้นโดยเฉลี่ย ของปัญหาที่ได้มีการกำหนดขึ้นเอง 96

4.15 แสดงแผนภูมิการเปรียบเทียบเวลาที่เครนทำงานเสร็จสิ้นน้อยที่สุด ของปัญหาที่ได้มีการกำหนดขึ้นเอง 96

4.16 แสดงแผนภูมิการเปรียบเทียบเวลาที่ใช้ประมวลผลโดยเฉลี่ย ของปัญหาที่ได้มีการกำหนดขึ้นเอง 97

4.17 แสดงแผนภูมิการเปรียบเทียบเวลาที่ใช้ประมวลผลที่น้อยที่สุด ของปัญหาที่ได้มีการกำหนดขึ้นเอง 98

ก.1 แสดงขั้นตอนการตั้งค่าความปลอดภัยแมโคร 109

ก.2 แสดงหน้าต่างการตั้งระดับความปลอดภัยของแมโคร 110

ก.3 แสดงหน้าต่างเพื่อเปิดการใช้แมโคร 110

ก.4 แสดงหน้าต้อนรับเข้าสู่โปรแกรมของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 111

ก.5 แสดงหน้าต่าง ABOUT ที่บอกลักษณะของโปรแกรม 111

ก.6 แสดงขั้นตอนที่ 1 ระบุค่าข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 112

ก.7 แสดงหน้าต่างแจ้งเตือนให้ระบุข้อมูลที่ถูกต้อง 113

ก.8 แสดงขั้นตอนที่ 2 เลือกวิธีการอบอ่อน ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 114

ก.9 แสดงขั้นตอนที่ 3 ระบุเวลางาน ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 115

ก.10 แสดงขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 116

สารบัญรูป (ต่อ)

รูปที่ หน้า

ก.11 แสดงการแก้ไขข้อมูลใน ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 117

ก.12 แสดงการแก้ไขเวลางานใน ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 118

ก.13 แสดงขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 118

ก.14 แสดงแถบความก้าวหน้าเมื่อกดประมวลผลใน ขั้นตอนที่ 5 ประมวลผลของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 119

ก.15 แสดงหน้าต่างยืนยันการหยุดการประมวลผล 120

ก.16 แสดงผลจากการจัดลำดับการทำงานเมื่อประมวลผลเสร็จสิ้น ในขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 120

ก.17 แสดงหน้าต่างของ Microsoft Excel เพื่อใช้ในการประมวลผลจากการจัดลำดับการทำงานจากการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง 121

ข.1 แสดงคำสั่งเข้าสู่โปรแกรม 123

ข.2 แสดงคำสั่งเมื่อกด START ในหน้าแรกของโปรแกรม 123

ข.3 แสดงคำสั่งเมื่อกด ABOUT ในหน้าแรกของโปรแกรม 123

ข.4 แสดงคำสั่งเมื่อกด EXIT ในหน้าแรกของโปรแกรม 123

ข.5 แสดงคำสั่งเมื่อกด OK ใน ABOUT ของโปรแกรม 123

ข.6 แสดงคำสั่งประกาศตัวแปรในหน้าระบุค่าข้อมูล 124

ข.7 แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 1 ปรากฏ 124

ข.8 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนงาน 124

ข.9 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนเครน 124

ข.10 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอุณหภูมิเริ่มต้น 124

ข.11 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอุณหภูมิสุดท้าย 124

ข.12 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนรอบการอบอ่อน 125

ข.13 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอัตราการเย็นตัว 125

ข.14 แสดงคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าของหน้าขั้นตอนที่ 1 ระบุค่าข้อมูล 125

ข.15 แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไปของหน้าขั้นตอนที่ 1 ระบุค่าข้อมูล 125

ข.16 แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 2 ปรากฏ 127

ข.17 แสดงคำสั่งเมื่อเลือกหลักการเลื่อนตำแหน่ง 127

ข.18 แสดงคำสั่งเมื่อเลือกหลักการคงที่หนึ่งตำแหน่ง 127

ข.19 แสดงคำสั่งเมื่อเลือกหลักการสลับที่ 127

ข.20 แสดงคำสั่งเมื่อเลือกหลักการพิจารณาที่ 2 ตำแหน่ง 127

ข.21 แสดงคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าของหน้าขั้นตอนที่ 2 เลือกวิธีการอบอ่อน 128

ข.22 แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไปของหน้าขั้นตอนที่ 2 เลือกวิธีการอบอ่อน 128

สารบัญรูป (ต่อ)

รูปที่		หน้า
ข.23	แสดงคำสั่งประกาศตัวแปรในหน้าระบุเวลางาน	128
ข.24	แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 3 ปรากฏ	129
ข.25	แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของเวลางาน	129
ข.26	แสดงคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าหรือกลับไปเวลางานก่อนหน้าของขั้นตอนที่ 3 ระบุเวลางาน	129
ข.27	แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไปหรือระบุเวลางานถัดไปของขั้นตอนที่ 3 ระบุเวลางาน	130
ข.28	แสดงคำสั่งประกาศตัวแปรในหน้ายืนยันและแก้ไขข้อมูล	130
ข.29	แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 4 ปรากฏ	130
ข.30	แสดงคำสั่งเมื่อเลือกแก้ไขข้อมูลในหน้ายืนยันและแก้ไขข้อมูล	131
ข.31	แสดงคำสั่งเมื่อหน้าต่างแก้ไขข้อมูล ปรากฏ	131
ข.32	แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนเครน	131
ข.33	แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอุณหภูมิเริ่มต้น	132
ข.34	แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอุณหภูมิสุดท้าย	132
ข.35	แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนรอบการหาคำตอบ	132
ข.36	แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอัตราการเย็นตัว	132
ข.37	แสดงคำสั่งเมื่อกดบันทึกในหน้าแก้ไขข้อมูล	132
ข.38	แสดงคำสั่งเมื่อกดดับเบิ้ลคลิกที่งานที่ต้องการแก้ไขเวลาทำงานในหน้ายืนยันและแก้ไขข้อมูล	133
ข.39	แสดงคำสั่งเมื่อกดแก้ไขเวลาทำงานในหน้ายืนยันและแก้ไขข้อมูล	133
ข.40	แสดงคำสั่งเมื่อหน้าต่างแก้ไขเวลางาน ปรากฏ	133
ข.41	แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของเวลางาน	133
ข.42	แสดงคำสั่งเมื่อกดบันทึกในหน้าแก้ไขเวลางาน	133
ข.43	แสดงคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้า ของขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล	134
ข.44	แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไป ของขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล	134
ข.45	แสดงคำสั่งประกาศตัวแปรในหน้าประมวลผล	135
ข.46	แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 5 ปรากฏ	136
ข.47	แสดงคำสั่งเมื่อกดกลับไปแก้ไขข้อมูล ของขั้นตอนที่ 5 ประมวลผล	136
ข.48	แสดงคำสั่งเมื่อกดเริ่มระบุข้อมูลใหม่ทั้งหมด ของขั้นตอนที่ 5 ประมวลผล	136
ข.49	แสดงคำสั่งเมื่อกดเริ่มการประมวลผล ของขั้นตอนที่ 5 ประมวลผล	136
ข.50	แสดงคำสั่งประกาศตัวแปรในหน้าต่างแสดงแถบความก้าวหน้า	137
ข.51	แสดงคำสั่งเมื่อหน้าต่างแถบความก้าวหน้า ปรากฏ ซึ่งจะเป็นการดำเนินการประมวลผลการจัดลำดับการทำงาน	137
ข.52	แสดงคำสั่งเมื่อกด X เพื่อหยุดการประมวลผล ของหน้าแสดงแถบความก้าวหน้า	142
ข.53	แสดงคำสั่งเมื่อกดไปยังหน้า EXCEL ของขั้นตอนที่ 5 ประมวลผล	142
ข.54	แสดงคำสั่งประกาศตัวแปรในหน้าต่างของ Microsoft Excel ในชีท Solve	142

สารบัญรูป (ต่อ)

รูปที่		หน้า
ข.55	แสดงคำสั่งเมื่อกดกลับไปยังโปรแกรม ของหน้าต่าง Microsoft Excel ในซีท Solve	143
ข.56	แสดงคำสั่งเมื่อกดเริ่มการประมวลผล ของหน้าต่าง Microsoft Excel ในซีท Solve	143
ข.57	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง checkNumeric	147
ข.58	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง checkNumericForInt	147
ข.59	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง ClearSolve	147
ข.60	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง ClearAllTimeOfJob	147
ข.61	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง ClearData	148
ข.62	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง PreparationToInputTime	148
ข.63	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง Newns	148
ข.64	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง SetFormatProcessing123	149
ข.65	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง SetFormatProcessing4	150
ข.66	แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns	151
ข.67	แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง ReadPTime	158
ข.68	แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS1	158
ข.69	แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS2	161
ข.70	แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS3	167
ข.71	แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS4	170
ข.72	แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง MakespanInterval	173
ข.73	แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง ShowDialog	179
ข.74	แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง CancelConfirm	180

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

ในปัจจุบันนี้การขนส่งทางทะเลนับว่ามีความสำคัญเป็นอย่างมาก และมีการใช้กันมากที่สุดเมื่อเทียบกับระบบการขนส่งแบบอื่นๆ เนื่องจากมีต้นทุนที่ต่ำและสามารถขนส่งได้ในปริมาณที่มากในครั้งเดียว เรือและท่าเรือจึงเป็นปัจจัยสำคัญในการขนส่งทางทะเล โดยมีการใช้ตู้คอนเทนเนอร์ในการบรรจุสินค้าต่างๆเหล่านั้น ซึ่งในการเคลื่อนย้ายสินค้าทางเรือโดยผ่านตู้คอนเทนเนอร์นั้น จะต้องมีการใช้เครื่องมือหรืออุปกรณ์ในการโยกย้ายถ่ายเท เครนจึงเป็นทางเลือกหนึ่งในการใช้ขนย้ายสินค้าหรือตู้คอนเทนเนอร์จากท่าเรือไปสู่เรือ จากเรือลงท่าเรือและจากท่าเรือสู่รถบรรทุกเพื่อขนส่งสินค้าสู่เป้าหมายต่อไป

ในการขนส่งสินค้าผ่านตู้คอนเทนเนอร์โดยใช้เครนยกนั้น จะต้องมีการบริหารจัดการที่ดีในเรื่องของพื้นที่ ลำดับการทำงานและเวลา เนื่องจากท่าเรือและเรือที่ใช้ในการขนส่งนั้น จะต้องมีสินค้าอีกมากในการใช้บริการของท่าเรือ ท่าเทียบเรือและตัวเรือ เพราะการขนส่งสินค้าในแต่ละครั้งนั้นนับเป็นต้นทุนทั้งสิ้น ไม่ว่าจะเป็นต้นทุนของผู้บริการท่าเรือ ต้นทุนของผู้ให้บริการเรือหรือผู้ใช้บริการขนส่งสินค้า ซึ่งจะเห็นได้ว่ามีเรือต่างๆมาต่อแถวในการเทียบเรือที่ท่าเรือ และมีการใช้เวลาในการเคลื่อนย้ายสินค้าหรือตู้คอนเทนเนอร์โดยใช้เครนช่วยยกอย่างรวดเร็ว ซึ่งในการเคลื่อนย้ายสินค้านั้นจะมีการรีบเร่งและนับเวลากันเป็นนาทีเลยทีเดียว

จากปัญหาในการบริหารจัดการการขนส่งทางเรือดังกล่าว การจัดลำดับการทำงานของเครนจึงนับเป็นเรื่องที่สำคัญเป็นอย่างยิ่ง ว่าทำอย่างไรเครนจึงจะสามารถเคลื่อนย้ายสินค้ามากมายเหล่านั้นโดยใช้เวลาน้อยที่สุด และใช้ประโยชน์ของเครนทุกตัวให้ได้มากที่สุด เพราะตัวเครนก็มีข้อจำกัดหลายประการ เช่น เครนไม่สามารถทำงานในเส้นทางที่ชัดกันได้ เครนจะไม่หยุดทำงานจนกว่างานจะเสร็จและไม่ทำงานอื่นจนกว่างานของตนจะเสร็จสิ้น เป็นต้น การบริหารจัดการการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง หรือ Simulated Annealing จึงนับเป็นอีกทางเลือกหนึ่งที่ผู้จัดทำโครงงานได้ใช้ศึกษาเพื่อจัดลำดับการทำงานของเครน ซึ่งวิธีการอบอ่อนจำลองนี้แม้จะไม่ใช่วิธีการที่ดีที่สุดในการจัดลำดับการทำงานของเครน แต่ก็เป็นวิธีที่ง่าย สะดวกและรวดเร็วที่สุด โดยเครนใช้เวลาในการเคลื่อนย้ายสินค้าน้อยที่สุด ซึ่งผู้จัดทำโครงงานได้ศึกษาการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง โดยใช้โปรแกรม Visual Basic for Applications บน Microsoft Excel มาใช้ในการบริหารจัดการเหล่านั้น เพื่อให้เครนสามารถเคลื่อนย้ายสินค้าโดยใช้เวลาน้อยที่สุด และใช้ประโยชน์ของเครนทุกตัวให้ได้มากที่สุด โดยใช้วิธีการที่ง่าย สะดวกและรวดเร็วที่สุดนั่นเอง

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อสร้างวิธีการในการจัดลำดับการทำงานของเครนในอุตสาหกรรมท่าเรือ โดยวิธีการอบอ่อนจำลอง

1.2.2 เพื่อพัฒนาโปรแกรมที่ใช้ในการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

1.3 เกณฑ์ชี้วัดผลงาน (Output)

1.3.1 โปรแกรมช่วยจัดลำดับการทำงานของเครน โดยใช้วิธีการอบอ่อนจำลองในการแก้ไขปัญหา ซึ่งจะใช้การเขียนโปรแกรม โดยใช้ภาษา Visual Basic for Applications บน Microsoft Excel ในการช่วยจัดลำดับการทำงาน

1.3.2 วิธีการอบอ่อนจำลองที่ใช้ในการจัดลำดับการทำงานของเครนในการขนถ่ายสินค้าที่ท่าเรือ

1.3.3 การเปรียบเทียบคำตอบที่ได้จากการจัดลำดับการทำงานของเครน โดยใช้วิธีการอบอ่อนจำลอง กับคำตอบที่ได้จากการจัดลำดับการทำงานของเครนโดยวิธีที่เคยมีผู้วิจัยมาแล้ว (Mathematical Programming Model)

1.4 เกณฑ์ชี้วัดผลสำเร็จ (Outcome)

วิธีการจัดลำดับการทำงานของเครนและโปรแกรมที่ช่วยในการจัดลำดับการทำงานของเครนสามารถนำไปช่วยในการจัดลำดับการทำงานของเครนในอุตสาหกรรมท่าเรือได้

1.5 ขอบเขตในการดำเนินโครงงาน

1.5.1 ศึกษาปัญหาการทำงานของเครนโดยทำการกำหนดขนาดของโจทย์เอง คือ

1.5.1.1 ปัญหาขนาดเล็ก มีเครนตั้งแต่ 1-3 เครน และมีงานตั้งแต่ 1-5 งาน

1.5.1.2 ปัญหาขนาดกลาง มีเครนตั้งแต่ 1-3 เครน และมีงานตั้งแต่ 1-10 งาน

1.5.1.3 ปัญหาขนาดใหญ่ มีเครนตั้งแต่ 1-6 เครน และมีงานตั้งแต่ 1-15 งาน

1.5.2 ศึกษาปัญหากรณีที่เครนห้ามทำงานไขว้กัน

1.5.3 ศึกษาปัญหาการทำงานของเครนโดยขณะที่เครนทำงาน เครนจะไม่ทำงานอื่นก่อนที่จะทำงานของตัวเองเสร็จ และเครนจะไม่หยุดทำงานจนกว่าจะทำงานทั้งหมดเสร็จสิ้น

1.5.4 ศึกษาเครนโดยไม่มีข้อจำกัดในเรื่องน้ำหนักของงาน

1.5.5 ศึกษาการจัดลำดับการทำงานของเครนที่มีประสิทธิภาพโดยการลดเวลาในการทำงานของเครนให้น้อยที่สุด

1.6 สถานที่ในการดำเนินโครงงาน

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

1.7 ระยะเวลาในการดำเนินโครงงาน

เดือนพฤศจิกายน พ.ศ.2553 ถึงเดือนมีนาคม พ.ศ.2554

1.8 ขั้นตอนและแผนการดำเนินโครงงาน

ตารางที่ 1.1 แสดงขั้นตอนและแผนการดำเนินโครงงาน

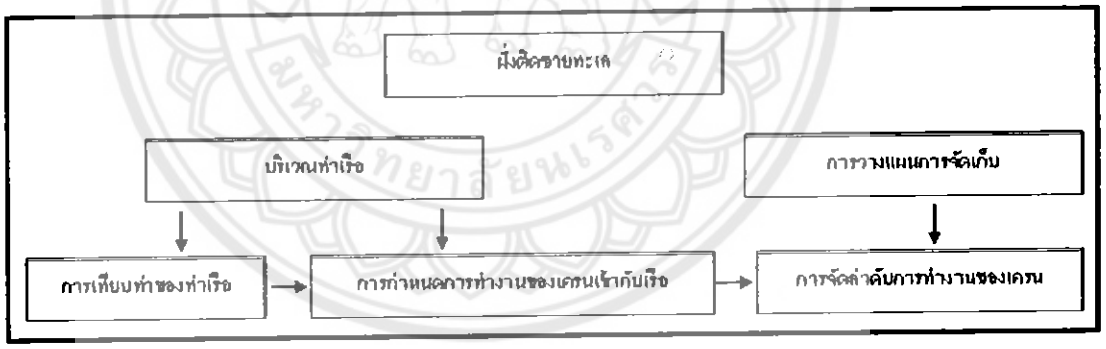
การดำเนินงาน	2553						2554					
	ต.ค.		พ.ย.		ธ.ค.		ม.ค.		ก.พ.		มี.ค.	
1. กำหนดปัญหาและศึกษาปัญหาการจัดลำดับการทำงานของเครน	←	→										
2. ศึกษาแบบจำลองเชิงคณิตศาสตร์		←	→									
3. ออกแบบวิธีการอบอ่อนจำลองโดยเขียนโปรแกรมแบบจำลองเชิงคณิตศาสตร์บนโปรแกรม Visual Basic for Applications บน Microsoft Excel			←	→	←	→						
4. การทดสอบโปรแกรมการแก้ปัญหาโดยวิธีการอบอ่อนจำลอง					←	→						
5. จัดทำรายงานและสรุปผล							←	→	→			
6. นำเสนอโครงงาน											←	→

บทที่ 2
หลักการและทฤษฎีเบื้องต้น

2.1 เรือ ท่าเรือ เครนและการปัญหาในการจัดลำดับการทำงานของเครน

ในการขนส่งทางทะเลนั้นจะมีการขนส่งด้วยระบบตู้คอนเทนเนอร์ ซึ่งตู้คอนเทนเนอร์นั้นจะมีโครงสร้างที่แข็งแรง อาจทำมาจากเหล็กหรืออลูมิเนียม สามารถวางเรียงซ้อนกันไม่น้อยกว่า 10 ชั้น ภายในตู้จะมีพื้นที่สำหรับใช้ในการวางและบรรจุสินค้า โดยสินค้าที่จะส่งจะต้องนำมาบรรจุใส่ตู้ และมีการขนย้ายตู้ขึ้นไว้บนเรือ ซึ่งออกแบบมาเป็นพิเศษสำหรับใช้ในการขนส่งสินค้าด้วยตู้คอนเทนเนอร์ โดยเรือสินค้าแต่ละลำจะมีที่ยกตู้ที่เรียกว่า เครน ประมาณ 1-4 ตัว โดยเครนแต่ละตัวจะลำเลียงตู้คอนเทนเนอร์ ซึ่งวางอยู่ตามความลึกของเรือ ซึ่งจะมีการเรียงกันเป็นคอลัมน์ ซึ่งท่าเรือที่จะมารองรับเรือประเภทนี้ จะต้องมีการออกแบบ เพื่อให้มีความเหมาะสมทั้งในเชิงวิศวกรรมและสิ่งแวดล้อม โดยจะต้องประกอบด้วย ท่าเทียบเรือ เขื่อนกั้นคลื่น รวมถึงสิ่งอำนวยความสะดวกต่างๆ

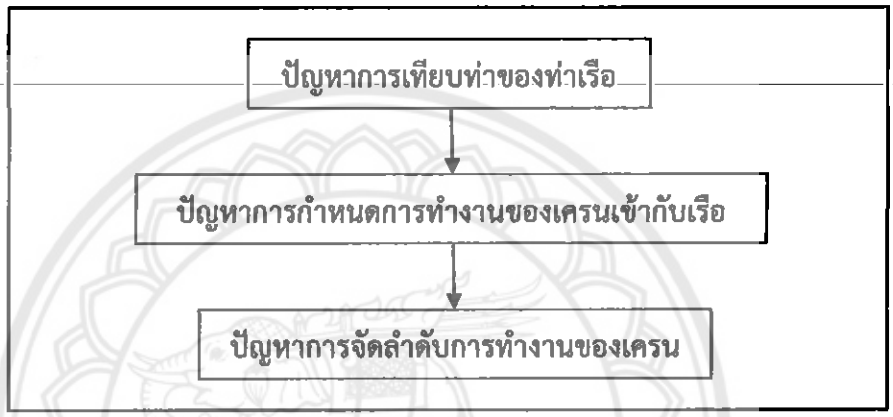
ในปัจจุบันนี้การวิจัยดำเนินงานมีบทบาทสำคัญในการจัดการด้านการเคลื่อนย้ายตู้คอนเทนเนอร์ ซึ่งมีงานวิจัยมากมายที่กล่าวถึงการวางแผนการดำเนินงานการขนถ่ายตู้คอนเทนเนอร์ จากสถานที่หนึ่งไปยังอีกสถานที่หนึ่ง จากรูปที่ 2.1 จะเห็นได้ว่าการจัดการเคลื่อนย้ายตู้คอนเทนเนอร์ในบริเวณท่าเรือที่ติดชายฝั่งทะเลมีความเกี่ยวข้องกันกับการเทียบท่าของท่าเรือ การกำหนดการทำงานของเครนเข้ากับเรือ และการจัดลำดับการทำงานของเครน



รูปที่ 2.1 แสดงความสัมพันธ์ระหว่างการเทียบท่าของท่าเรือ การกำหนดการทำงานของเครนเข้ากับเรือและการจัดลำดับการทำงานของเครน
ที่มา: Christian and Frank (2553)

ปัญหาของการวางแผนการดำเนินงานบริเวณชายฝั่งทะเลจะเกี่ยวข้องกับปัญหาต่อไปนี้ คือปัญหาการเทียบท่าของท่าเรือ จะเกี่ยวกับพื้นที่ของท่าเรือและเวลาในการขนถ่ายสินค้าลงจากเรือ ปัญหาการกำหนดการทำงานของเครนเข้ากับเรือ และปัญหาการจัดลำดับการทำงานของเครนจะเกี่ยวกับการขนถ่ายตู้คอนเทนเนอร์ระหว่างเรือและท่าเรือ ซึ่งปัญหาทั้งหมดเหล่านั้นจะมีความสัมพันธ์และความเชื่อมโยงซึ่งกันและกัน ดังจะแสดงได้ในรูปที่ 2 จะเห็นได้ว่าเมื่อเรือมาเทียบท่าที่ท่าเรือ จะเกิดปัญหาการเทียบท่าของท่าเรือ โดยจะต้องแก้ไขปัญหาว่า เรือแต่ละลำจะเข้าเทียบท่าเรืออย่างไรภายใต้ข้อจำกัดต่างๆ เช่น รูปแบบและขนาดของตัวเรือ หรือขนาดและลักษณะของท่าเรือ เป็นต้น

เมื่อกำหนดการเทียบท่าของท่าเรือได้แล้ว จะเริ่มการขนย้ายสินค้าของเรือไปสู่ท่าเรือ จะเกิดปัญหาการกำหนดการทำงานของเครนเข้ากับเรือ ว่าเรือลำใดควรจะใช้เครนตัวไหน และใช้เครนกี่ตัวในการขนย้ายสินค้าและตู้คอนเทนเนอร์เหล่านั้น โดยใช้เวลาในการขนย้ายสินค้าของเรือแต่ละลำที่มาเทียบท่าได้น้อยที่สุด เมื่อกำหนดการทำงานของเครนให้กับเรือได้แล้ว จะเริ่มดำเนินการขนย้ายสินค้าและตู้คอนเทนเนอร์ จะเกิดปัญหาขึ้นว่าจะจัดลำดับการทำงานของเครนในเรืออย่างไร ให้ใช้เวลาในการขนถ่ายสินค้าน้อยที่สุด ภายใต้ข้อกำจัดของเครนนั้นๆ เช่น เครนไม่สามารถทำงานไขว้กันหรือขัดกันได้ หรือในขณะที่เครนทำงาน เครนจะไม่ทำงานอื่นก่อนที่จะทำงานของตัวเองเสร็จ และเครนจะไม่หยุดทำงานจนกว่าจะทำงานทั้งหมดเสร็จสิ้น เป็นต้น



รูปที่ 2.2 แสดงความสัมพันธ์ระหว่างปัญหาการเทียบท่าของท่าเรือ ปัญหาการกำหนดการทำงานของเครนเข้ากับเรือและปัญหาการจัดลำดับการทำงานของเครน
ที่มา: Christian and Frank (2553)

ในการศึกษาปัญหาการจัดลำดับการทำงานของเครน (Quay crane scheduling problems : QCSP) จะสามารถพิจารณาได้จากงานที่ทำในแต่ละงาน ว่างานหนึ่งๆนั้นควรทำงานก่อนหรือหลังอีกงานหนึ่ง ซึ่งถ้างานใดถูกกำหนดว่าจะต้องทำงานนั้นก่อน จึงจะทำงานอีกงานหนึ่งได้ ก็จำเป็นต้องทำงานนั้นก่อนให้สำเร็จเช่นกัน เช่น มีงาน 3 งาน คือ งาน ก งาน ข และงาน ค โดยมีข้อกำหนดว่า งาน ข และงาน ค สามารถทำงานพร้อมกันได้ แต่ถ้าหากจะทำงาน ข หรืองาน ค แล้ว จะต้องทำงาน ก ก่อน เช่นนั้นแล้ว ถ้าต้องการทำงานทั้งหมด จะต้องทำงาน ก ก่อน จึงจะทำงานต่อไปได้ เป็นต้น ข้อพิจารณาอีกข้อหนึ่งคือ ณ เวลาหนึ่งๆนั้น เครนสามารถทำงานได้เพียงหนึ่งงานเท่านั้น และไม่สามารถทำงานอื่นพร้อมกันได้ นอกจากนั้นยังมีข้อพิจารณาอีกว่า ในขณะที่เครนตัวหนึ่งทำงานกับงานงานหนึ่งนั้น เครนตัวนั้นจะต้องทำงานนั้นให้เสร็จสิ้นเสียก่อน จึงจะทำงานอื่นต่อไปได้ ซึ่งก่อนที่งานนั้นจะสำเร็จเสร็จสิ้น เครนจะไม่สามารถข้ามไปทำงานอื่นได้ แล้วย้อนกลับมาทำงานต่อ หรือไม่อาจให้เครนตัวอื่นมาทำงานเดิมต่อจนเสร็จได้

จากทั้งหมดที่กล่าวมานี้จะสามารถหาคำตอบของปัญหานี้ คือ ตารางการทำงานของเครน (Quay crane schedule) ซึ่งสามารถบอกได้ว่า งานแต่ละงานนั้นถูกเริ่มทำงานในเวลาเท่าไร โดยเครนตัวใด และ ณ เวลาไหน โดยทั่วไปแล้วการหาค่าต่ำสุดของการแก้ปัญหาของเครนเพื่อค้นหาเวลางานทั้งหมดที่ทำให้เสร็จสิ้นน้อยที่สุด (Makespan) มักจะถูกใช้เป็นฟังก์ชันเป้าประสงค์ (Objective function) เนื่องจากเป็นการแสดงถึงเวลาในการขนถ่ายสินค้าของเรือทั้งหมด จากปัญหาทั้งหมด

เหล่านี้จะถูกเรียกกันว่า NP-hard ซึ่ง NP-hard จะหมายความได้ว่า เมื่อขนาดของปัญหาเพิ่มขึ้น (เช่นตัวแปรที่เกี่ยวข้องเพิ่มมากขึ้น) เวลาที่จะใช้ในการแก้ปัญหา เพื่อให้ได้คำตอบเป็นค่าที่เหมาะสมที่สุด (Optimal Solution) นั้นจะเพิ่มมากยิ่งขึ้นแบบ Exponential และเพื่อหลีกเลี่ยงข้อจำกัดจากการห้ามข้ามกัน ไขว้กัน หรือชัดกันของเครนนั้น ในการศึกษาปัญหาการจัดลำดับการทำงานของเครนจะต้องมีการกำหนดเงื่อนไขการห้ามข้ามกันของเครนด้วย

จากการศึกษาปัญหาการจัดลำดับการทำงานของเครน จะแสดงได้ว่าการเคลื่อนที่ของเครนจะมีรายละเอียดต่างๆ ด้านเวลาการทำงานของเครน ดังนี้

2.1.1 เวลาก่อนการทำงานของเครน (Ready times): การกำหนดเวลาการทำงานของเครนขึ้นเอง โดยใช้กำหนดความเป็นไปได้ก่อนการทำงานของเครน ว่าเครนตัวไหนควรทำงานใดก่อนหรือหลังเพื่อให้เกิดเวลาในการทำงานน้อยที่สุด

2.1.2 การวางขอบเขตเวลาการทำงานของเครน (Time windows): เป็นการวางกรอบหรือขอบเขตของเวลาในการทำงานของเครนแต่ละครั้งให้ครอบคลุมที่สุด ซึ่งจะถูกพิจารณาว่าใช้ได้กับเรือลำใดก่อน เพื่อให้เกิดเวลาในการทำงานน้อยที่สุด นอกจากนี้ การวางขอบเขตเวลาการทำงานของเครนยังเป็นผลสำเร็จของการกำหนดการทำงานของเครนเข้ากับเรืออีกด้วย

2.1.3 เวลาการทำงานของเครน (Positions): พิจารณาเวลาในการทำงานของเครนตั้งแต่เริ่มการทำงานจนกระทั่งทำงานเสร็จสิ้น

2.1.4 ความเร็วการเคลื่อนที่ของเครน (Travel times): ความเร็วของการเคลื่อนที่ของเครน ซึ่งจะมีผลต่อระยะเวลาในการเคลื่อนย้ายสินค้าของเครนในแต่ละครั้ง

ซึ่งในเรื่องของการจัดลำดับการทำงานของเครนนั้น ได้เคยมีผู้วิจัยมาแล้วมากมาย ทั้งในระดับนักวิจัยชำนาญการ ซึ่งได้มีผลงานวิจัยในวารสารทางวิชาการทั้งในและต่างประเทศ และผู้วิจัยระดับปริญญาในแขนงต่างๆ ซึ่งผลงานวิจัยเหล่านั้นได้มีบทบาทในการดำเนินงานวิจัยนี้เป็นอย่างมาก เช่น งานวิจัยระดับปริญญาวิศวกรรมศาสตร์บัณฑิต มหาวิทยาลัยนเรศวร ของ กัญญารัตน์ คุ้มคลอง และสายฝน ช่างเหล่า ในเรื่อง การจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง ปีการศึกษา 2550 หรือ งานวิจัยระดับปริญญาวิศวกรรมศาสตร์บัณฑิต มหาวิทยาลัยนเรศวร ของ เอราวิล ถาวร , หนึ่งฤทัย ทัพใหญ่ และกนกพร อารยิกานนท์ ในเรื่อง การแก้ปัญหาการจัดลำดับการทำงานของเครนโดยใช้โปรแกรมเชิงคณิตศาสตร์ ปีการศึกษา 2550 เป็นต้น ซึ่งผู้วิจัยจะนำเอาความรู้ที่ได้จากงานวิจัยอื่นดังที่กล่าวมา มามีส่วนในปรับปรุงและพัฒนางานวิจัยทางด้านการจัดลำดับการทำงานของเครนของตัวผู้วิจัยเองต่อไป (Christian and Frank, 2553), (เอราวิลและคณะ, 2550)

2.2 วิธีการอบอ่อนจำลอง (Simulated Annealing)

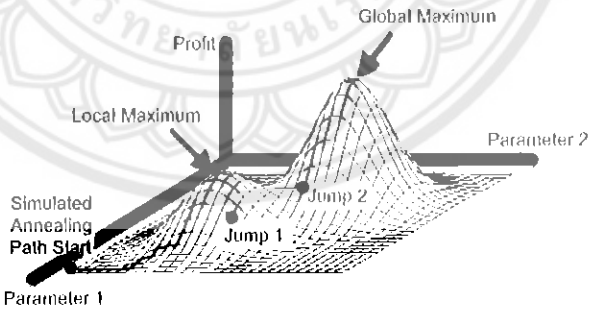
วิธีการอบอ่อนจำลอง หรือ Simulated Annealing เป็นวิธีการที่จะนำมาใช้ในการค้นหาคำตอบของปัญหาหนึ่งๆในการจัดลำดับการทำงานของเครน ซึ่งวิธีการอบอ่อนจำลองนี้เป็นวิธีการหนึ่งในการค้นหาคำตอบของปัญหาแบบฮิวริสติก (Heuristic) กล่าวคือการค้นหาคำตอบแบบฮิวริสติก คือ วิธีการหรือขั้นตอนวิธีที่สามารถหาคำตอบที่ดี หรืออาจได้คำตอบที่เหมาะสมที่สุด และใช้เวลาในการค้นหาคำตอบอย่างรวดเร็ว (กรุง)

การอบอ่อนจำลอง (Simulated Annealing หรือ SA) เป็นเทคนิคการค้นหาคำตอบแบบสุ่ม ซึ่งเลียนแบบกรรมวิธีการควบคุมอุณหภูมิของการอบโลหะให้ร้อนแล้วปล่อยให้เย็นลงช้าๆ เพื่อให้เกิดการเปลี่ยนแปลงทางโครงสร้างผลึกที่มีพลังงานภายในน้อยที่สุด คุณลักษณะดังกล่าวจะทำให้โลหะมี

คุณสมบัติทางกายภาพที่แข็งแกร่งและทนทาน ในทางตรงกันข้าม ถ้าไม่ทำการควบคุมการทำให้เย็นของโลหะร้อนแล้ว โครงสร้างที่ได้จะมีจุดตำหนิหรือบกพร่อง เป็นโลหะมีความแข็ง แต่เปราะ กรรมวิธีการควบคุมอุณหภูมิของการอบโลหะดังกล่าวเรียกว่าการอบอ่อน (annealing process) การอบอ่อนจำลองจึงเป็นเทคนิคการค้นหาคำตอบที่เหมาะสมที่สุด โดยมีการควบคุมอุณหภูมิของการอบอ่อนเป็นการปรับค่าตัวแปรของการค้นหาคำตอบของระบบนั่นเอง

การอบอ่อนจำลองได้รับการพัฒนาขึ้นในปี 1983 โดย Kirkpatrick และคณะ เพื่อใช้ในการแก้ปัญหาที่ไม่เป็นเชิงเส้นและมีความซับซ้อน และการค้นหาคำตอบมักติดอยู่ใน Local Optimum ซึ่งนั่นคือ การที่คำตอบที่ค้นพบถูกจำกัดอยู่ในวงแคบที่เกิดจากการที่กระบวนการค้นหาคำตอบยอมรับเฉพาะคำตอบที่ดีขึ้นเท่านั้น ทำให้โอกาสที่จะค้นหาคำตอบแบบอื่นๆ น้อยลง คำตอบสุดท้ายที่ได้จึงเป็นคำตอบที่ต่ำที่สุดเฉพาะในกลุ่มของคำตอบเพียงกลุ่มเดียว การอบอ่อนจำลองจะเข้าไปแก้ไขปัญหาในจุดนี้ โดยการยอมรับคำตอบที่มีการยืดหยุ่น กล่าวคือ นอกจากจะยอมรับคำตอบที่ดีขึ้นแล้วยังยอมรับคำตอบที่ด้อยลงด้วย ซึ่งจะถูกกำหนดด้วยค่าความน่าจะเป็นค่าหนึ่ง (p) ทำให้โอกาสที่จะค้นหาคำตอบที่ดีซึ่งอาจจะอยู่ในบริเวณอื่นเป็นไปได้มากขึ้น นั่นคือทำให้ได้รับคำตอบที่เหมาะที่สุดแบบวงกว้างหรือ Global optimum นั่นเอง (สมชัย, 2544)

การอบอ่อนจำลองจะมีการใส่ค่าสุ่มเข้าไปในการค้นหาคำตอบ การสุ่มค่าดังกล่าวทำให้สามารถ *"กระโดด"* หรือหลุดออกจากคำตอบที่เหมาะที่สุดแบบ Local Optimum ได้ ดังรูปที่ 2.3 การกระโดดดังกล่าวคือ การปรับอุณหภูมิในกระบวนการอบอ่อนข้างต้นนั่นเอง การปรับอุณหภูมิของการอบอ่อนจำลองยังสามารถพิจารณาได้อีกสองแง่มุม กล่าวคือที่อุณหภูมิสูง ระบบจะทำการค้นหาแบบหยาบๆ (การกระโดดไปมาอย่างแรง) ในขณะที่การค้นหาแบบละเอียดจะได้จากระบบที่มีอุณหภูมิต่ำๆ ด้วยลักษณะการทำงานแบบสุ่มดังกล่าว เราอาจเรียกการอบอ่อนจำลองว่าเป็นอัลกอริทึมเชิงความน่าจะเป็น (probabilistic algorithm)



รูปที่ 2.3 แสดงการสุ่มค่าค้นหาคำตอบเพื่อกระโดดและหลุดออกจาก Local Optimum
ที่มา: Max (2551)

การอบอ่อนจำลองมีองค์ประกอบที่สำคัญ 4 ประการ คือ รูปแบบของคำตอบที่ต้องการค้นหา ตัวกำเนิดสุ่ม (random generator) หรือตัวปฏิบัติการค้นหาสำหรับทำการ "เดิน" หรือ "move" คำตอบของระบบไปยังคำตอบใหม่ ฟังก์ชันเป้าประสงค์ (objective function) ของระบบ และตารางจัดการการอบอ่อน (annealing schedule) ซึ่งเป็นส่วนที่ใช้ปรับค่าอุณหภูมิและกำหนดเวลาที่จะให้ระบบเกิดการเปลี่ยนแปลงอุณหภูมิ ซึ่งรายละเอียดและขั้นตอนของการอบอ่อนจำลอง มีดังต่อไปนี้

2.2.1 การสุ่มค่าเริ่มต้น ซึ่งค่าเริ่มต้นที่ได้จะกำหนดให้เป็นค่าที่ยอมรับเริ่มต้น และเป็นค่าคำตอบที่ดีที่สุดเริ่มต้น

2.2.2 กำหนดตารางจัดการการอบอ่อนสำหรับตัวแปรอุณหภูมิ T และ ตั้งค่าเริ่มต้นของ T ไว้ที่ค่าสูงอย่างเพียงพอ

2.2.3 การค้นหาคำตอบใหม่ $x_p = x + \Delta x$ โดยที่ Δx เป็นการเปลี่ยนแปลงคำตอบที่ถูกนำเสนอต่อระบบ

2.2.4 คำนวณหาการเปลี่ยนแปลงของฟังก์ชันเป้าประสงค์ $\Delta f = f(x_p) - f(x)$

2.2.5 ทำการคำนวณหาว่าควรจะใช้ x_p เป็นคำตอบหรือสถานะใหม่ของระบบหรือไม่ กล่าวคือใช้ความน่าจะเป็นในการตัดสินใจที่จะเปลี่ยนสถานะตามเงื่อนไขต่อไปนี้

$$Pr(x \rightarrow x_p) = \begin{cases} 1 & \text{ถ้า } \Delta f < 0 \\ e^{-\Delta f/T} & \text{ถ้า } \Delta f \geq 0 \end{cases} \qquad (2.1)$$

โดยที่ T แทนอุณหภูมิของระบบ ในกรณีที่ $\Delta f \geq 0$ ค่าตัวเลขสุ่มที่มีการกระจายแบบสม่ำเสมอ η จะถูกเลือกจากในช่วง [0, 1] ถ้า $Pr(x \longrightarrow x_p) > \eta$ แล้ว คำตอบ x_p จะถูกใช้เป็นคำตอบหรือสถานะใหม่ของระบบ มิฉะนั้นแล้วให้คงคำตอบของระบบไว้ที่ x เหมือนเดิม

2.2.6 ขั้นตอนข้างต้นจะถูกทำซ้ำจนกระทั่งระบบเข้าสู่จุดสมดุล ที่ซึ่งรู้ได้จากจำนวนครั้งของการเปลี่ยนสถานะไม่มีนัยสำคัญเพียงพอ สภาวะดังกล่าวนี้จะเกิดขึ้นเมื่อระบบมีคำตอบหรือสถานะเข้าใกล้ค่าเหมาะที่สุดแบบ Local Optimum ซึ่งจำนวนครั้งของการทำซ้ำจะถูกกำหนดไว้ก่อนล่วงหน้า

2.2.7 ทำการปรับค่าอุณหภูมิ T ตามตารางจัดการการอบอ่อน แล้วเริ่มทำขั้นตอนทำซ้ำทั้งหมดใหม่อีกครั้ง กระบวนการทั้งหมดสามารถยุติได้เมื่อค่าอุณหภูมิ T มีค่าเป็นศูนย์ (อาทิตย์, 2552)

2.3 โปรแกรม Visual Basic for Applications บน Microsoft Excel

Microsoft Excel เป็นโปรแกรมประเภทแผ่นตารางการทำงาน ที่มีความสามารถในการทำงานที่เกี่ยวข้องกับคณิตศาสตร์ ข้อมูลทางธุรกิจ สถิติ เป็นต้น ที่ได้รับความนิยมเป็นอย่างมาก นอกจากการทำงานในลักษณะของโปรแกรมประเภทแผ่นตารางแล้ว เรายังสามารถพัฒนาขีดความสามารถของ Microsoft Excel ได้อีกหลากหลายเพื่อตอบสนองความต้องการของผู้ใช้ได้มากขึ้น นั่นก็คือการเขียนโปรแกรมให้เหมาะสมกับความต้องการ จากการที่ Microsoft Excel ที่เป็นโปรแกรมผู้ใช้งานคอมพิวเตอร์ทุกคนนิยมและคุ้นเคย ทำให้การเขียนโปรแกรมบน Microsoft Excel ทำได้ง่าย ไม่ยุ่งยาก แก้ไขปัญหาได้ง่าย ปรับปรุงเปลี่ยนแปลงได้ง่าย และโปรแกรมที่ใช้มีความยืดหยุ่นสูง นั่นเป็นอีกเหตุผลหนึ่งที่มีคนเป็นจำนวนมากนิยมเขียนโปรแกรมบน Microsoft Excel เช่น การเขียนโปรแกรมกับงานที่ต้องการสั่งทำอัตโนมัติตามเวลาที่ตั้งไว้ งานที่เปิดโอกาสให้ผู้ใช้ข้อมูลเลือก Option ได้ตามต้องการ งานที่ต้องการใช้งานร่วมกับแหล่งข้อมูลสารพัดรูปแบบ หรืองานที่ต้องเชื่อมประสานข้อมูลระหว่างโปรแกรมต่างๆใน Microsoft Office อย่างรวดเร็ว ซึ่งโปรแกรมที่ถูกเขียนขึ้นและใช้งานบน Excel จะเรียกว่า Visual Basic for Application (VBA)

Visual Basic for Application คือ โปรแกรมวิชวลเบสิกที่ถูกฝังไว้ใน Microsoft Excel เป็นส่วนที่สามารถใช้งานได้โดยการเขียนโปรแกรม ซึ่งรูปแบบการเขียนโปรแกรมนั้นไม่แตกต่างจากการเขียนโปรแกรม Visual Basic เท่าใดนัก เพียงแต่สามารถเข้าใจและเขียนโปรแกรมได้ง่ายกว่า ภาษาไม่ซับซ้อนมากนัก นอกจากการเขียนโปรแกรมโดยผู้ใช้งานแล้ว ในส่วนนี้ยังเก็บคำสั่งที่เกิดจากการ

บันทึกขั้นตอนต่างๆในการเขียนโปรแกรมด้วย VBA ซึ่งการบันทึกนี้จะเรียกว่า macro นอกจากนี้ VBA ยังสามารถเขียนโปรแกรมต่างๆได้ด้วยคำสั่งหรือ code ต่างๆ และมีหน้าต่างการทำงานการเขียนและแก้ไขคำสั่ง VBA คือ Visual Basic Editor (VBE) อีกด้วย (เวชยันต์, 2551) , (กิตตินันท์, 2552)

บทที่ 3
วิธีการดำเนินโครงงาน

3.1 การกำหนดปัญหาและศึกษาปัญหาการจัดลำดับการทำงานของเครน

ปัญหาที่เกิดขึ้นในแวดวงอุตสาหกรรมส่วนมากมีความซับซ้อนมาก การกำหนดปัญหาให้ตรงกับเป้าหมายจึงมีความสำคัญอย่างยิ่ง เพื่อที่จะหาผลลัพธ์แล้วนำไปปฏิบัติได้จริง การจัดตั้งปัญหาและศึกษาปัญหา มีหลักพอสังเขปดังนี้

จุดประสงค์ในการจัดลำดับการทำงานของเครนก็เพื่อต้องการที่จะลดเวลาในการทำงานของเครน และสิ่งที่จะกล่าวถึง คือ การแก้ปัญหาของเครนเพื่อค้นหาเวลางานทั้งหมดที่ทำให้เสร็จสิ้นน้อยที่สุด (makespan) โดยใช้วิธีการอบอ่อนจำลอง (Simulated Annealing) การกำหนดขนาดของปัญหาที่ใช้ในการทดสอบวิธีการแก้ปัญหา หรือ วิธีการหาคำตอบโดยใช้วิธีการอบอ่อนจำลอง โดยผู้จัดทำโครงงานจะทำการศึกษาความสัมพันธ์ที่เกี่ยวข้องกัน โดยจะกำหนดขนาดของปัญหาที่ใช้ในการทดสอบวิธีการแก้ปัญหาที่จะพิจารณา กำหนดจุดประสงค์และวิธีการวัดผลการดำเนินงาน กำหนดขอบเขตและสมมติฐานของปัญหา และกำหนดแนวทางการดำเนินงานที่เป็นไปได้ในการแก้ปัญหา รวมถึงการกำหนดช่วงเวลาในการแก้ปัญหา เพื่อค้นหาเวลาที่งานทั้งหมดเสร็จสิ้นน้อยที่สุดโดยใช้วิธีการอบอ่อนจำลอง

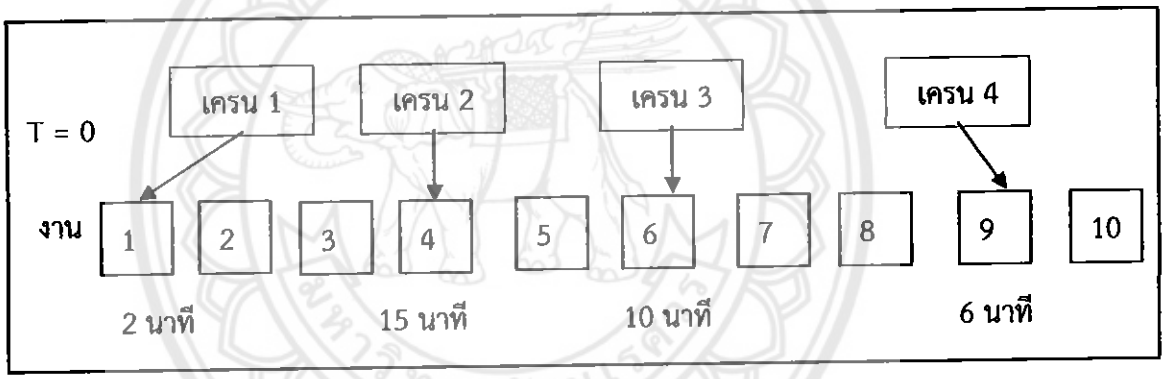
<u>ตัวอย่างเรียงการจัดลำดับการทำงานของเครน</u>

เป็นการแสดงรูปแบบการทำงานของเครน เมื่อจำนวนงานที่กำหนดให้ 10 งาน โดยแต่ละงานมีเวลาที่กำหนดดังตารางที่ 3.1 การจัดลำดับการทำงานของเครนได้แสดงไว้ในรูป 3.1 โดยแต่ละรูปแสดงรูปแบบการทำงานของเครนแต่ละตัวที่กระทำภายใต้ข้อจำกัด 3 ข้อ โดยในที่นี้งานที่ 1 คือ ตำแหน่งที่ 1 , งานที่ 2 คือ ตำแหน่งที่ 2 ,..., งานที่ n ในตำแหน่งที่ n โดยการจัดลำดับงานที่นำมาแสดงนี้เป็นเพียงคำตอบที่เป็นไปได้คำตอบหนึ่งเท่านั้น

ตารางที่ 3.1 แสดงข้อมูลการทำงานของเครนแต่ละงาน

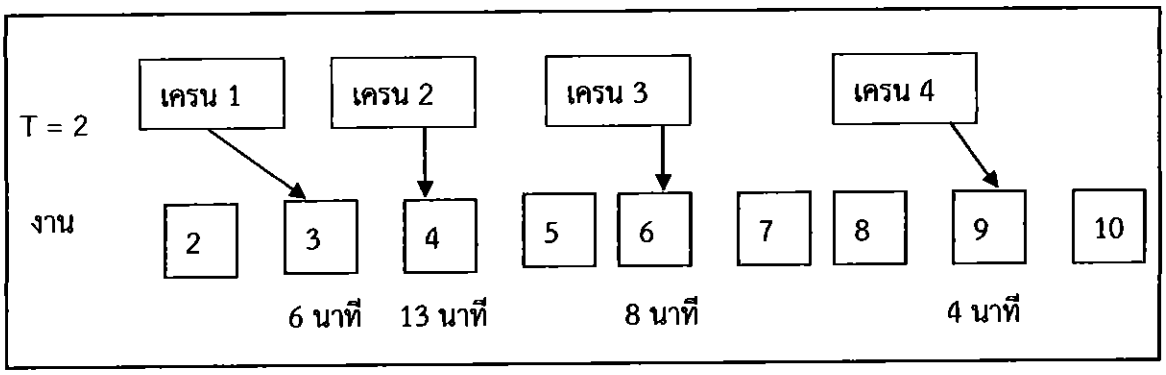
ตำแหน่งงาน	เวลาการทำงาน (นาที)
1	2
2	1
3	6
4	15
5	2
6	10
7	6
8	5
9	6
10	3

เมื่อกำหนดให้ T เป็นเวลา



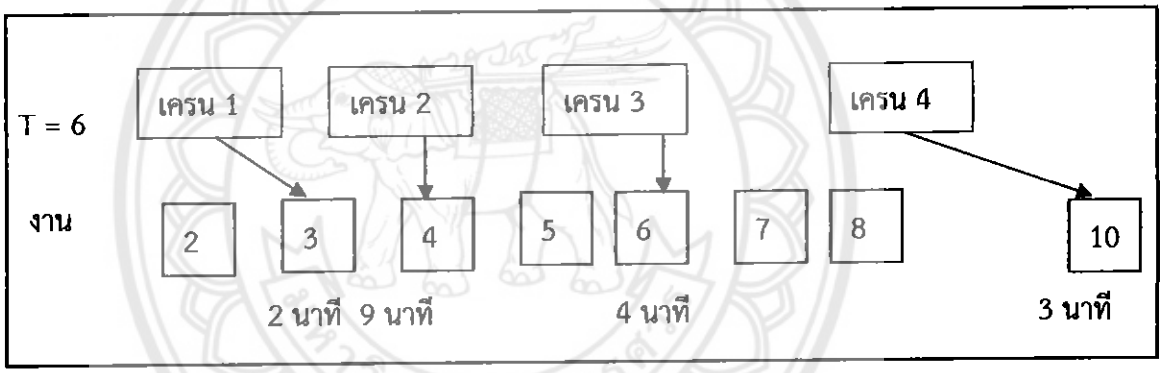
รูปที่ 3.1 การทำงานของเครนที่ T = 0 ณ ตำแหน่งต่างๆ

จากรูปที่ 3.1 จะเห็นได้ว่า ณ เวลาที่ T = 0 เครนแต่ละตัวเลือกทำงานตัวละหนึ่งงาน โดยกำหนดให้เครน 1 เลือกทำงานที่ 1 เครน 2 เลือกทำงานที่ 4, เครน 3 เลือกทำงานที่ 6 และเครน 4 เลือกทำงานที่ 9 ซึ่งเครน 1 จะใช้เวลาในการปฏิบัติงานเท่ากับ 2 นาที เครน 2 ใช้เวลาในการปฏิบัติงาน 15 นาที เครน 3 ใช้เวลาในการปฏิบัติงาน 10 นาที และเครน 4 ใช้เวลาในการปฏิบัติงาน 6 นาที



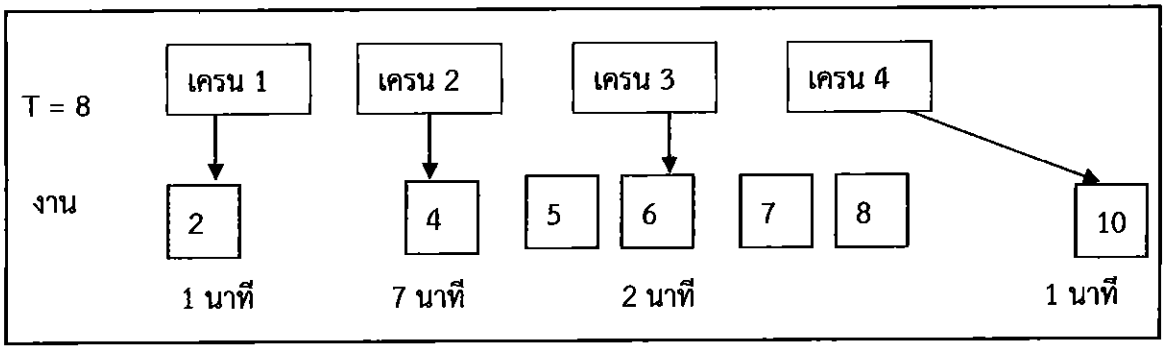
รูปที่ 3.2 การทำงานของเครนที่ T = 2 ณ ตำแหน่งต่างๆ

จากรูปที่ 3.2 จะเห็นได้ว่า เมื่อเครน 1 ทำงานที่ 1 เสร็จ ณ เวลา T = 2 เครน 1 สามารถเลือกทำงานที่ 2 และ 3 ได้ ดังนั้นกำหนดให้เครน 1 เลือกทำงานที่ 3 และเครน 1 จะใช้เวลาในการทำงานเท่ากับ 6 นาที เครน 2 จะเหลือเวลาในการทำงานเท่ากับ 13 นาที เครน 3 เหลือเวลาการทำงาน 8 นาที เครน 4 เหลือเวลาการทำงาน 4 นาที



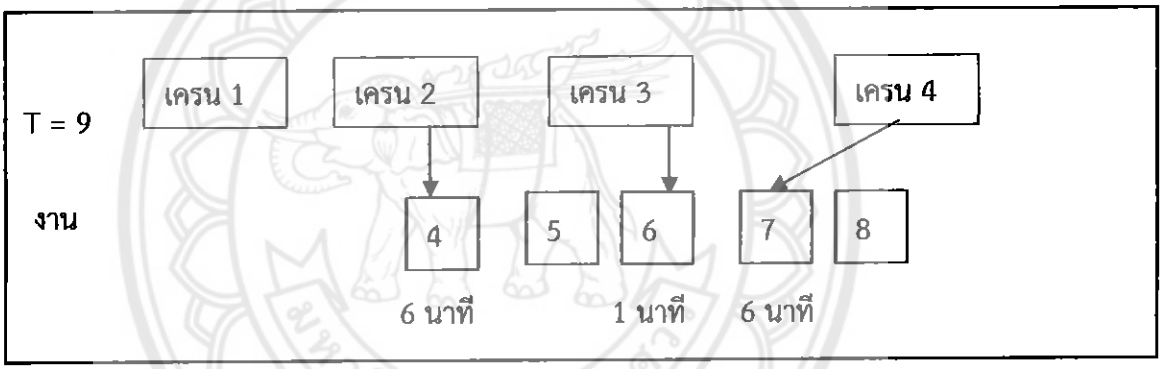
รูปที่ 3.3 การทำงานของเครนที่ T = 6 ณ ตำแหน่งต่างๆ

จากรูปที่ 3.3 จะเห็นได้ว่า ณ เวลา 4 นาทีถัดไป เครน 4 ทำงานที่ 9 เสร็จ แล้วเครน 4 สามารถเลือกทำงานที่ 7,8 และ 10 ได้ กำหนดให้เครน 4 เลือกทำงานที่ 10 และใช้เวลาในการทำงานเท่ากับ 3 นาที เครน 1 จะเหลือเวลาในการทำงานเท่ากับ 2 นาที เครน 2 เหลือเวลาการทำงาน 9 นาที เครน 4 เหลือเวลาการทำงาน 3 นาที



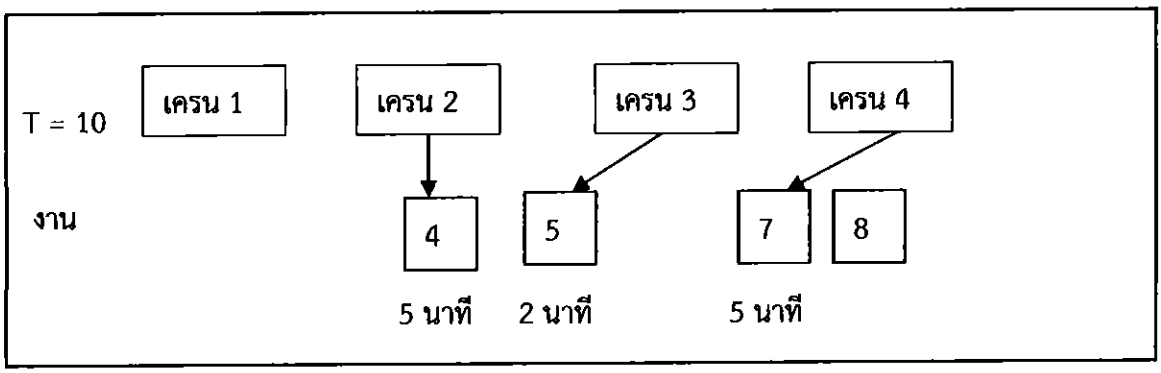
รูปที่ 3.4 การทำงานของเครนที่ T = 8 ณ ตำแหน่งต่างๆ

จากรูปที่ 3.4 จะเห็นได้ว่า ที่เวลา T = 8 เครน 1 ทำงานที่ 3 เสร็จแล้วกำหนดให้เครน 1 ทำงานที่ 2 ใช้เวลาในการทำงานเท่ากับ 1 นาที เครน 2 ยังคงทำงานในเครนที่ 4 ต่อไป โดยเหลือเวลาการทำงาน 7 นาที เครน 3 เหลือเวลาการทำงานที่ 6 เท่ากับ 2 นาที และเครน 4 เหลือเวลาในการทำงานที่ 10 เท่ากับ 1 นาที



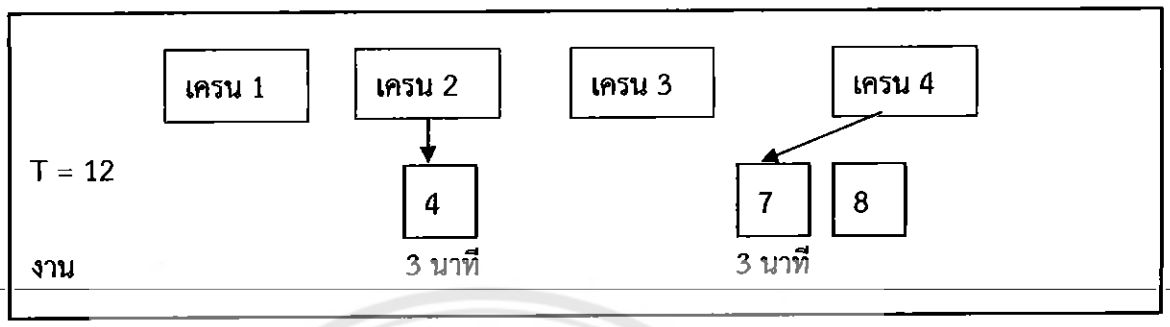
รูปที่ 3.5 การทำงานของเครนที่ T = 9 ณ ตำแหน่งต่างๆ

จากรูปที่ 3.5 จะเห็นได้ว่า เครน 1 ไม่สามารถเลือกทำงานได้ เนื่องจากเครนไม่สามารถทำงานข้ามตัดกันได้ เครน 1 จึงไม่สามารถเลือกทำงานที่ 5, 7 และ 8 ได้ แต่เครน 4 สามารถเลือกทำงานที่ 7 และ 8 ได้ โดยกำหนดให้เครน 4 เลือกทำงานที่ 7



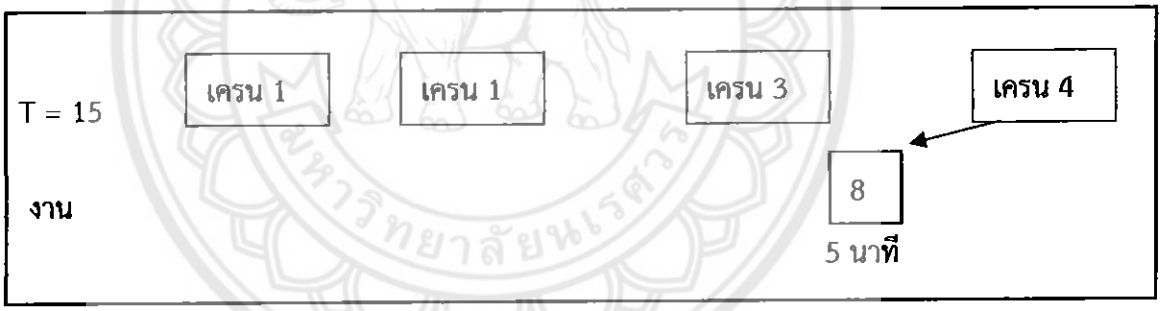
รูปที่ 3.6 การทำงานของเครนที่ T = 10 ณ ตำแหน่งต่างๆ

จากรูปที่ 3.6 จะเห็นได้ว่า ที่เวลา T = 10 เครน 3 ทำงานที่ 6 เสร็จแล้วสามารถเลือกทำงานที่ 5 ได้เท่านั้น จึงกำหนดให้เลือกทำงานที่ 5 โดยใช้เวลาในการทำงานเท่ากับ 2 นาที เครน 2 ซึ่งยังคงทำงานที่ 4 ไม่เสร็จเหลือเวลาในการทำงานอีก 5 นาที และเครน 4 ยังทำงานที่ 7 ที่ยังเหลือเวลา 5 นาที



รูปที่ 3.7 การทำงานของเครนที่ T = 12 ณ ตำแหน่งต่างๆ

จากรูปที่ 3.7 จะเห็นได้ว่า ที่เวลา T = 12 เมื่อเครน 3 ทำงานที่ 5 เสร็จแล้ว จะไม่สามารถกำหนดเลือกทำงานต่อได้ เนื่องจากเงื่อนไขที่ว่าการทำงานของเครนห้ามทำงานข้ามตัดกัน ดังนั้น เครน 1 และ 3 จึงเกิดการรอให้เครน 2 ทำงานที่ 4 เครน 4 ทำงานที่ 7 เสร็จเป็นเวลาทั้งสิ้น 3 นาที



รูปที่ 3.8 การทำงานของเครนที่ T = 15 ณ ตำแหน่งต่างๆ

จากรูปที่ 3.8 จะเห็นได้ว่า ที่เวลา T = 15 เมื่อเครน 2 และ 4 ทำงานเสร็จ ณ เวลา 3 นาทีต่อมา เครนทุกตัวจึงสามารถเลือกทำงานที่ 8 ได้ กำหนดให้เครน 4 ทำงานที่ 8 เป็นเวลาทั้งสิ้น 5 นาที ดังนั้นที่เวลา T = 18 งานทั้งหมดจึงแล้วเสร็จ ซึ่งจะใช้เวลาในการทำงานทั้งหมด 20 นาที

3.2 การศึกษาแบบจำลองเชิงคณิตศาสตร์

เมื่อกำหนดและเข้าใจถึงปัญหาอย่างถูกต้องแล้ว ในทางการวิจัยดำเนินงานนิยมใช้ตัวแบบคณิตศาสตร์สำหรับแทนระบบของปัญหา โดยปัญหานี้ได้กล่าวมาแล้ว ว่าได้มีผู้สร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหานี้แล้ว

3.3 ออกแบบวิธีการอบอ่อนจำลองโดยเขียนโปรแกรมแบบจำลองเชิงคณิตศาสตร์บนโปรแกรม Visual Basic for Applications บน Microsoft Excel

ในขั้นตอนนี้ผู้วิจัยจะทำการออกแบบวิธีการแก้ปัญหาโดยใช้วิธีการอบอ่อนจำลอง ซึ่งเมื่อออกแบบวิธีการได้แล้ว จึงจะนำไปเขียนโปรแกรมคอมพิวเตอร์ด้วยโปรแกรม Visual Basic for Applications บน Microsoft Excel

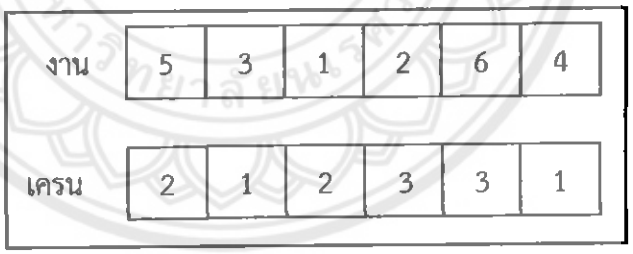
ซึ่งในการดำเนินการออกแบบวิธีการอบอ่อนจำลองนี้ จะมีกระบวนการหาคำตอบอยู่ 2 ลักษณะ คือ การหาคำตอบแรกที่ได้จากการสุ่ม และการหาคำตอบใหม่ในย่านคำตอบเดิมหรือการทำ Neighbourhood search ซึ่งจะเห็นได้ว่า การหาคำตอบใหม่ เป็นส่วนหนึ่งของกระบวนการอบอ่อนจำลอง ซึ่งมีจุดประสงค์ คือ ต้องการหาเวลาที่งานทั้งหมดถูกกระทำเสร็จสิ้นให้มีค่าน้อยที่สุด ภายใต้เงื่อนไขการห้ามทำงานข้ามกัน หรือไขว้กันของเครน ทั้งนี้ผู้วิจัยได้มีการออกแบบวิธีการหาคำตอบใหม่ในย่านคำตอบเดิม ทั้งหมด 4 แบบ ที่มีความแตกต่างกันออกไป ได้แก่

1. วิธีหาคำตอบใหม่โดยใช้หลักการเลื่อนตำแหน่ง
2. วิธีหาคำตอบใหม่โดยใช้หลักการคงที่ 1 ตำแหน่ง
3. วิธีหาคำตอบใหม่โดยใช้หลักการสลับที่
4. วิธีหาคำตอบใหม่โดยใช้หลักการพิจารณาที่ 2 ตำแหน่ง

ซึ่งในการหาคำตอบใหม่โดยใช้หลักการต่างๆข้างต้น สามารถอธิบายได้โดยใช้ตัวอย่างการทำงานดังต่อไปนี้

3.3.1 การหาคำตอบแรกที่ได้จากการสุ่ม

ในขั้นตอนแรกจะทำการสุ่มหาคำตอบ ที่จะเป็นคำตอบแรกออกมา ดังรูปที่ 3.9



รูปที่ 3.9 ตัวอย่างการสุ่มคำตอบแรกในกระบวนการอบอ่อนจำลอง

เมื่อ

ลำดับการทำงานของงาน แทนด้วยตำแหน่งที่เรียงจากซ้ายไปขวา

จำนวนงาน แทนด้วยจำนวนตำแหน่ง

จากตัวอย่างนี้ มีจำนวนเครน 3 เครน และงานจำนวน 6 งาน จะเห็นว่าคำตอบที่ได้จากการสุ่มคำตอบแรก คือ

ลำดับ 1 คือ งาน 5 ถูกทำโดยเครน 2

ลำดับ 2 คือ งาน 3 ถูกทำโดยเครน 1

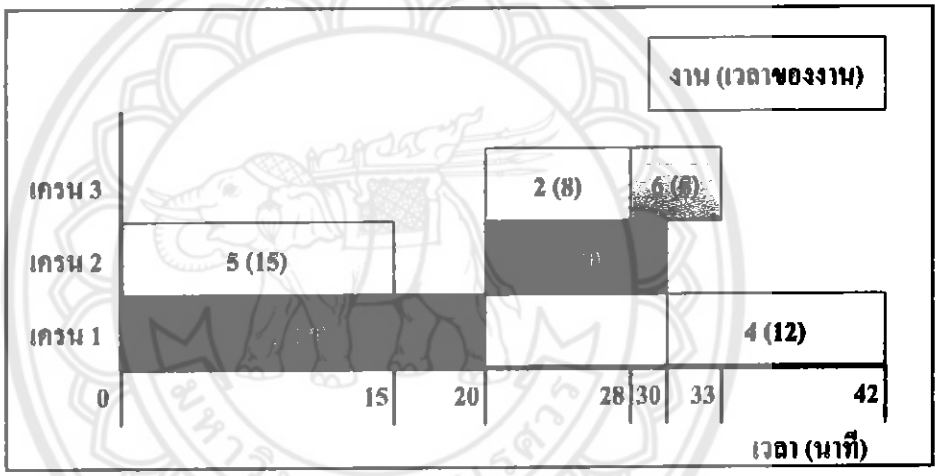
ลำดับ 3 คือ งาน 1 ถูกทำโดยเครน 2

ลำดับ 4 คือ งาน 2 ถูกทำโดยเครน 3
ลำดับ 5 คือ งาน 6 ถูกทำโดยเครน 3
ลำดับ 6 คือ งาน 4 ถูกทำโดยเครน 1

เมื่อกำหนดให้เวลาที่ใช้ในการทำงานแต่ละงาน มีดังนี้

งานที่ 1 ใช้เวลาในการทำงานทั้งสิ้น 10 นาที
งานที่ 2 ใช้เวลาในการทำงานทั้งสิ้น 8 นาที
งานที่ 3 ใช้เวลาในการทำงานทั้งสิ้น 20 นาที
งานที่ 4 ใช้เวลาในการทำงานทั้งสิ้น 12 นาที
งานที่ 5 ใช้เวลาในการทำงานทั้งสิ้น 15 นาที
งานที่ 6 ใช้เวลาในการทำงานทั้งสิ้น 5 นาที

จากตัวอย่างจะสามารถแสดงแผนภาพลำดับการทำงานของเครนได้ ดังรูปที่ 3.10



รูปที่ 3.10 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบที่ได้จากการสุ่มครั้งแรก โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

3.3.2 การหาคำตอบใหม่

เมื่อทำการหาคำตอบแรกได้แล้ว จะเริ่มมีการหาคำตอบใหม่ ซึ่งจะหาคำตอบใหม่ได้โดยการทำ Neighborhoods Search โดยจะทำการสุ่มหาคำตอบของตำแหน่งของงานและตำแหน่งของเครนใหม่ โดยเรียงตามลำดับการทำงานจากซ้ายไปขวา และจะทำการสุ่มจากคำตอบแรกเสมอ โดยในการทำการสุ่มหาคำตอบใหม่โดยการเปลี่ยนตำแหน่งของงานและตำแหน่งของเครนใหม่นั้น จะมีหลักการพิจารณาได้หลายหลักการ ซึ่งในที่นี้จะขอใช้หลักการพิจารณา 4 หลักการ ซึ่งจะขอทำการอธิบายดังตัวอย่างต่อไปนี้

3.3.2.1 วิธีหาคำตอบใหม่โดยใช้หลักการเลื่อนตำแหน่ง

ในการหาคำตอบใหม่โดยหลักการเลื่อนตำแหน่งนั้น จะเป็นการสุ่มตำแหน่งต่างๆ ในงานหรือเครนนั้นๆ แล้วให้นำตำแหน่งที่พิจารณาไปแทนที่ตำแหน่งสุ่ม หลังจากนั้นจึงทำการเลื่อนตำแหน่งไปลำดับถัดไป ดังเช่นตัวอย่างเดิม มีงานจำนวน 6 งาน และเครนจำนวน 3 เครน จะสามารถหาคำตอบใหม่ได้ ดังต่อไปนี้

งาน	5	3	1	2	6	4
เครน	2	1	2	3	3	1

รูปที่ 3.11 ตัวอย่างการหาคำตอบใหม่โดยหลักการเลื่อนตำแหน่ง

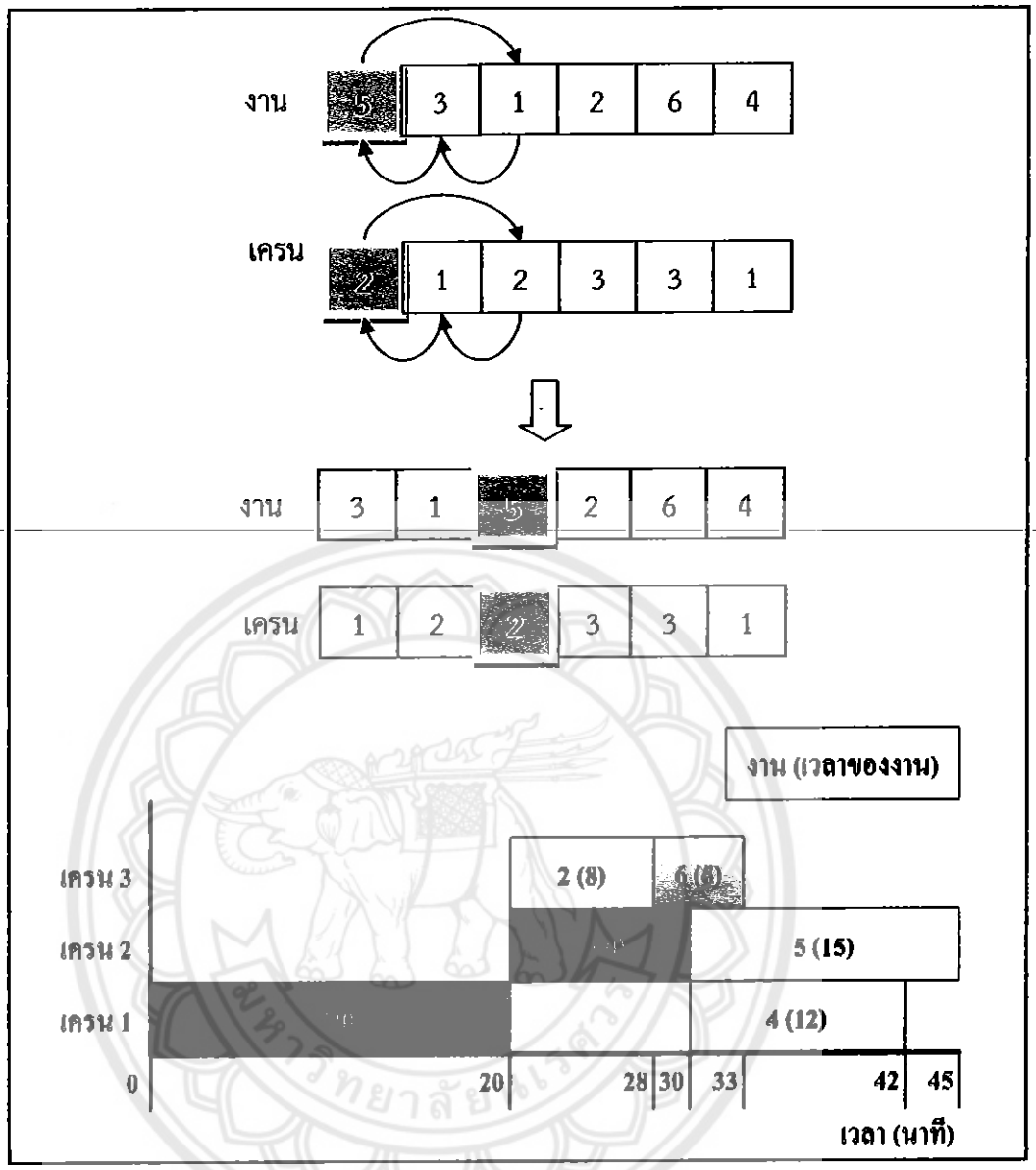
เราสามารถหาค่า X_{new} ได้ทั้งหมด 2 (n-1) คำตอบ โดยที่

n = จำนวนงาน

ดังนั้น จากตัวอย่างนี้ เราสามารถหา X_{new} ได้ทั้งหมด 10 คำตอบ ซึ่งสามารถพิจารณาตำแหน่งการสุ่มได้เป็น 3 ส่วน ดังต่อไปนี้

ก. พิจารณาที่ตำแหน่งแรก

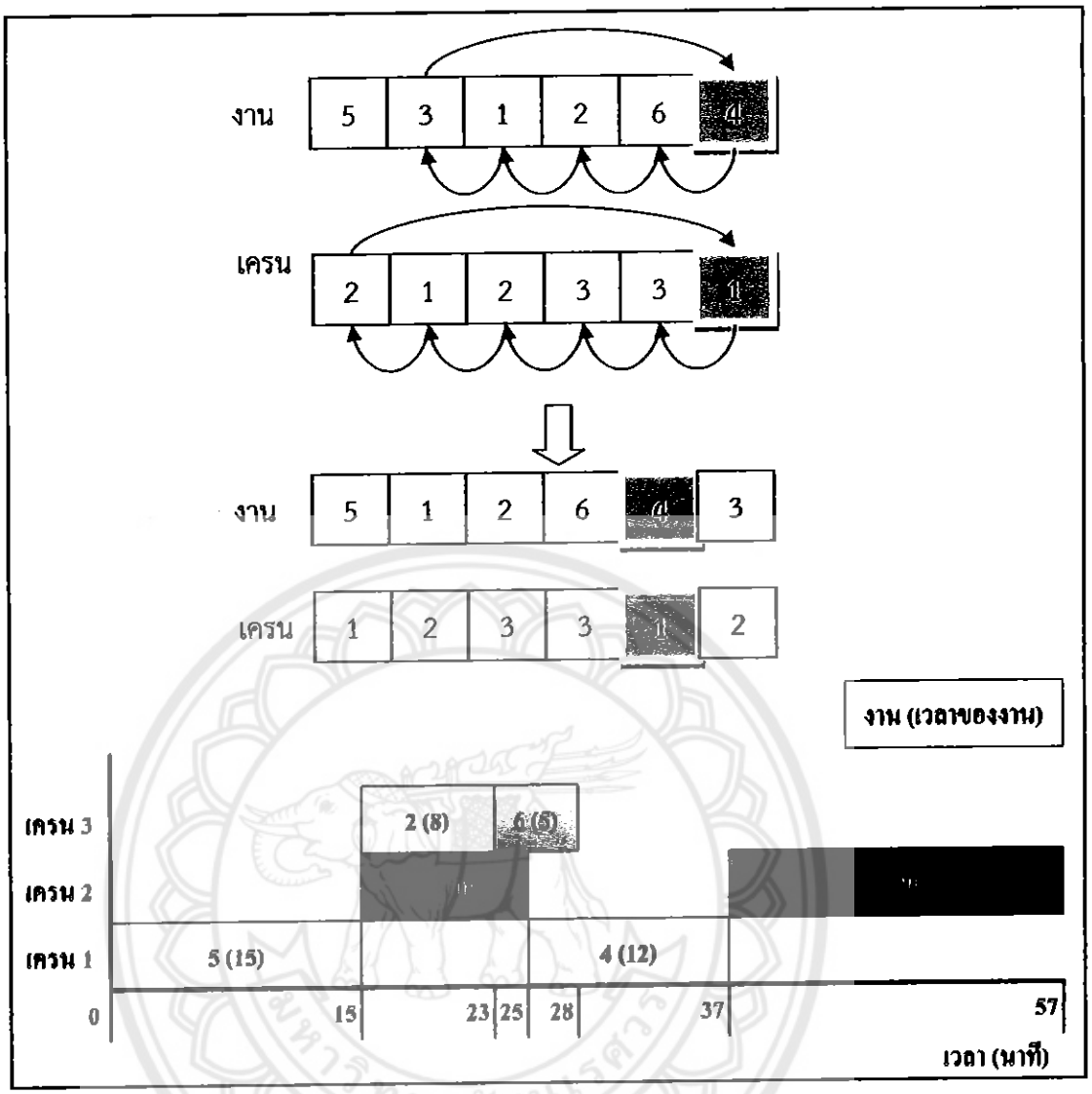
ในการพิจารณาที่ตำแหน่งแรกนั้น จากตัวอย่าง ตำแหน่งแรกจะไปแทนที่ได้ 5 ตำแหน่ง ไปทางขวามือ โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป โดยผลของการสุ่มจากตัวอย่างนี้คือ ที่ตำแหน่งของงาน สุ่มได้ตำแหน่งที่ 3 คืองานที่ 1 ทำย้ายงาน 5 จากตำแหน่งที่ 1 ไปแทนที่งาน 1 ที่ตำแหน่งที่ 3 และเลื่อนงานในตำแหน่งที่ 2 และ 3 ไปทางซ้ายมือทีละตำแหน่ง และที่ตำแหน่งของเครน สุ่มได้ตำแหน่งที่ 3 คือเครนที่ 2 ก็ย้ายตำแหน่งลักษณะเช่นเดียวกันกับตำแหน่งของงานเช่นเดียวกัน ดังรูป และจะได้คำตอบใหม่ คำตอบที่ 1 จากการทำ Neighborhoods Search คือ



รูปที่ 3.12 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งแรกของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 45 นาที

ข. พิจารณาที่ตำแหน่งสุดท้าย

ในการพิจารณาที่ตำแหน่งสุดท้ายนั้น จากตัวอย่างเดิม ตำแหน่งสุดท้ายจะไปแทนที่ได้ 5 ตำแหน่ง ไปทางซ้ายมือ โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป เช่นเดียวกับที่พิจารณาที่ตำแหน่งแรก จากตัวอย่างนี้ สุ่มได้ตำแหน่งที่ 2 คือ งานที่ 3 จะทำการย้ายงานที่ 3 ที่สุ่มได้จากตำแหน่งที่ 2 ไปแทนที่ตำแหน่งที่เราพิจารณาคือตำแหน่งสุดท้าย และเลื่อนงานที่เราพิจารณาคือตำแหน่งสุดท้าย นั่นคือ งานที่ 4 ไปทางซ้าย ทีละ 1 ตำแหน่ง และที่ตำแหน่งของเครนเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 2 ดังรูป



รูปที่ 3.13 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งสุดท้ายของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 57 นาที

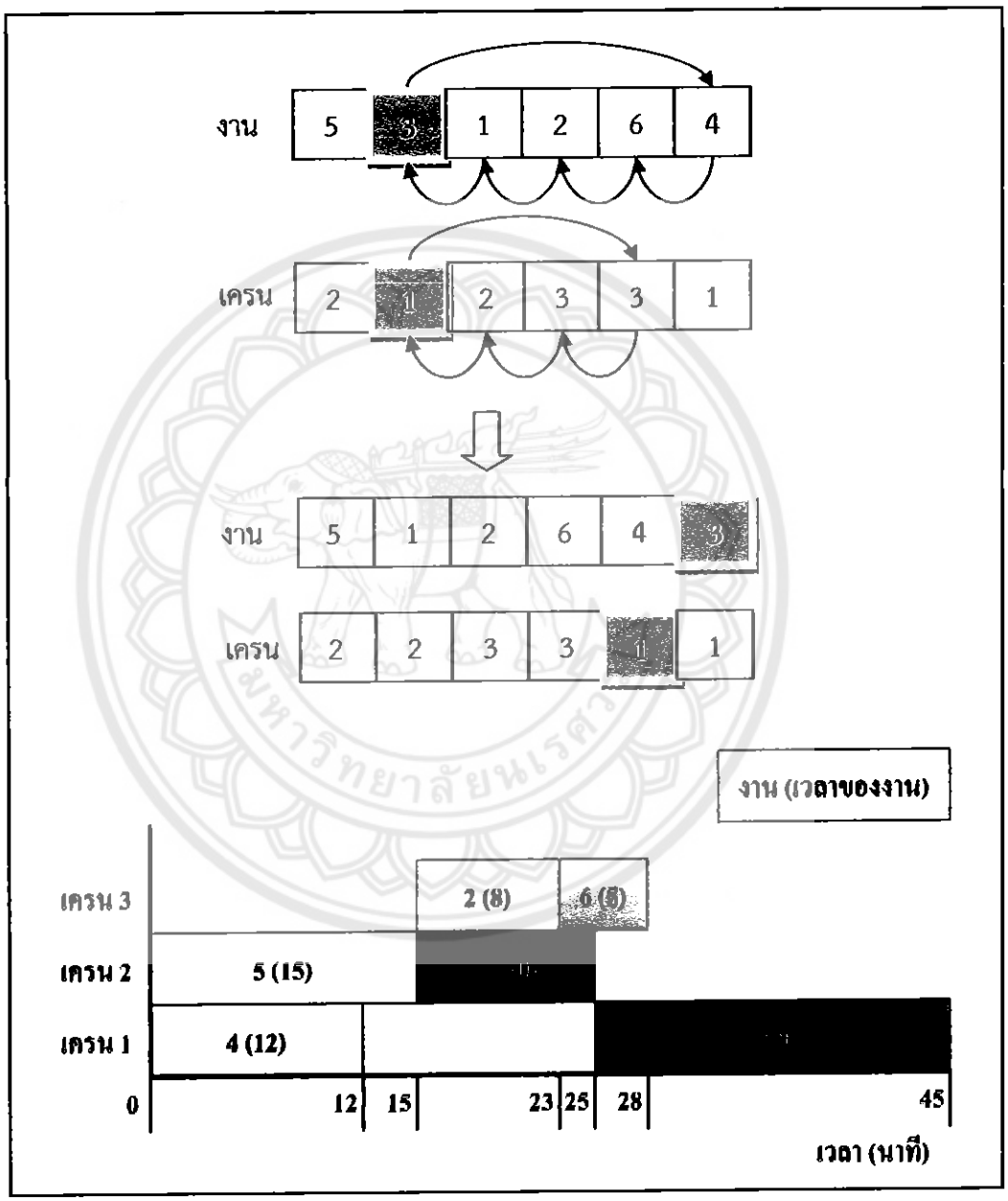
ค. พิจารณาที่ตำแหน่งกลาง

ในการพิจารณาในตำแหน่งกลางนั้น จะสามารถพิจารณาได้จาก ตำแหน่งทั้งหมดที่อยู่ระหว่างตำแหน่งแรกกับตำแหน่งสุดท้าย ซึ่งในแต่ละตำแหน่งนั้นเราสามารถหาคำตอบได้ 2 คำตอบ โดยการเลื่อนงานและเครนในตำแหน่งที่เราสนใจไปทางขวา และทางซ้ายมือ โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป

จากตัวอย่างเดิม ตำแหน่งกลางที่อยู่ระหว่างตำแหน่งแรกและตำแหน่งสุดท้าย มี 4 ตำแหน่ง คือ ตำแหน่งที่ 2, 3, 4 และ 5 ซึ่งในแต่ละตำแหน่งนั้น สามารถหาคำตอบได้ 2 คำตอบคือ การเลื่อนงานและเครนไปทางซ้ายและขวามือ ดังนั้นจะพิจารณาได้ในแต่ละตำแหน่ง คือ

ค.1 เลือกพิจารณาที่ตำแหน่งที่ 2 ไปทางขวามือ

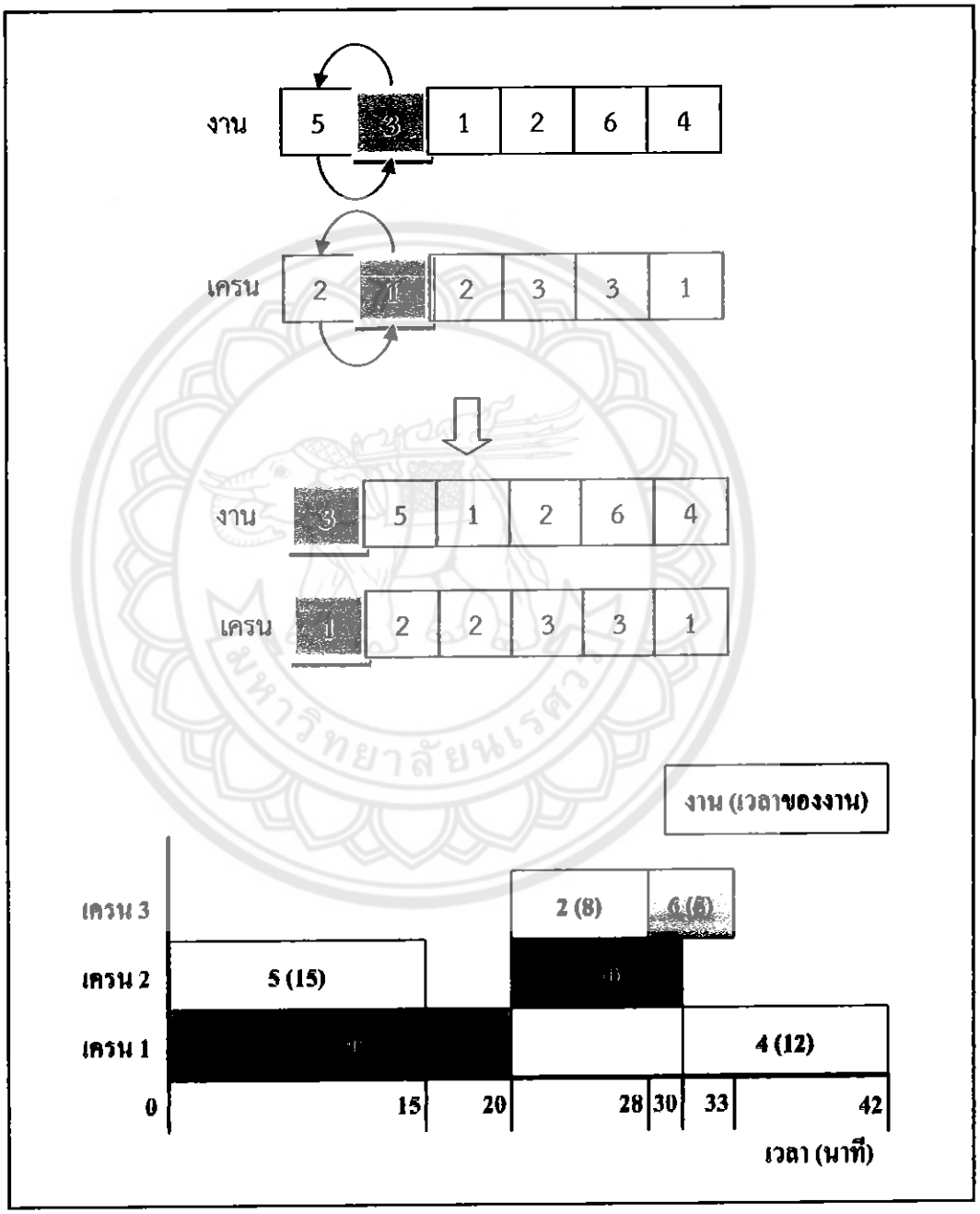
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 2 และสุ่มตำแหน่งงานที่ย้ายไป คือ ตำแหน่งสุดท้าย นั่นคือ งานที่ 3 ที่ตำแหน่งที่ 2 จะย้ายไปแทนที่งานที่ 4 ที่ตำแหน่งสุดท้าย และเลื่อนงานในตำแหน่งที่ 6, 5, 4, 3 ไปทางซ้ายมือ ทีละ 1 ตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 3 ดังรูป



รูปที่ 3.14 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางขวามือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 45 นาที

ค.2 เลือกพิจารณาที่ตำแหน่งที่ 2 ไปทางซ้ายมือ

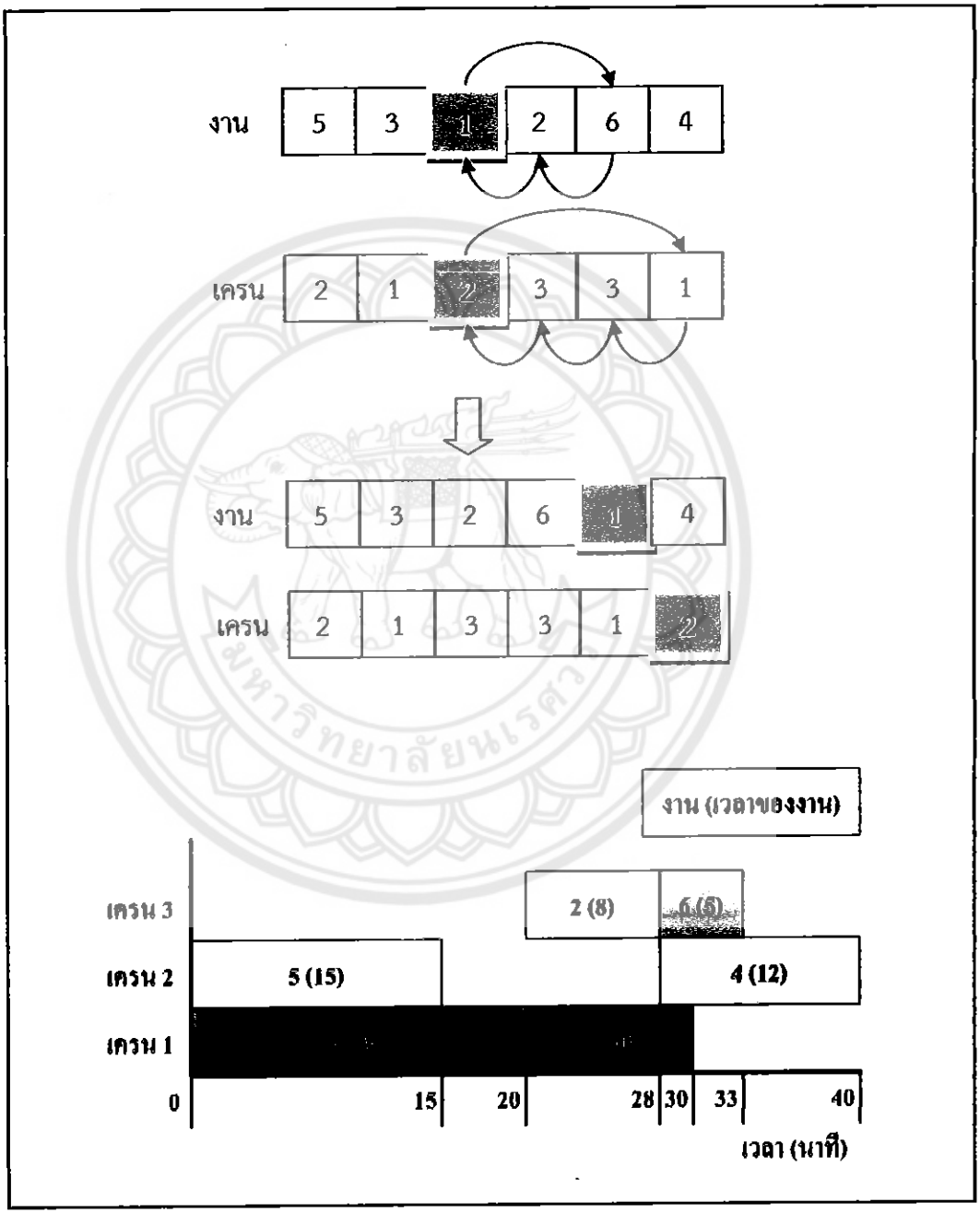
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 2 และสุ่มตำแหน่งงานที่ย้ายไปทางซ้ายมือ ในที่นี้มีงานเดียว คือ ที่งานที่ตำแหน่งแรก นั่นคือ งานที่ 3 ที่ตำแหน่งที่ 2 จะย้ายไปแทนที่งานที่ 5 ที่ตำแหน่งแรก และเลื่อนงานในตำแหน่งแรก ไปทางขวามือ ทีละ 1 ตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 4 ได้ ดังรูป



รูปที่ 3.15 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางซ้ายมือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ค.3 เลือกพิจารณาที่ตำแหน่งที่ 3 ไปทางขวามือ

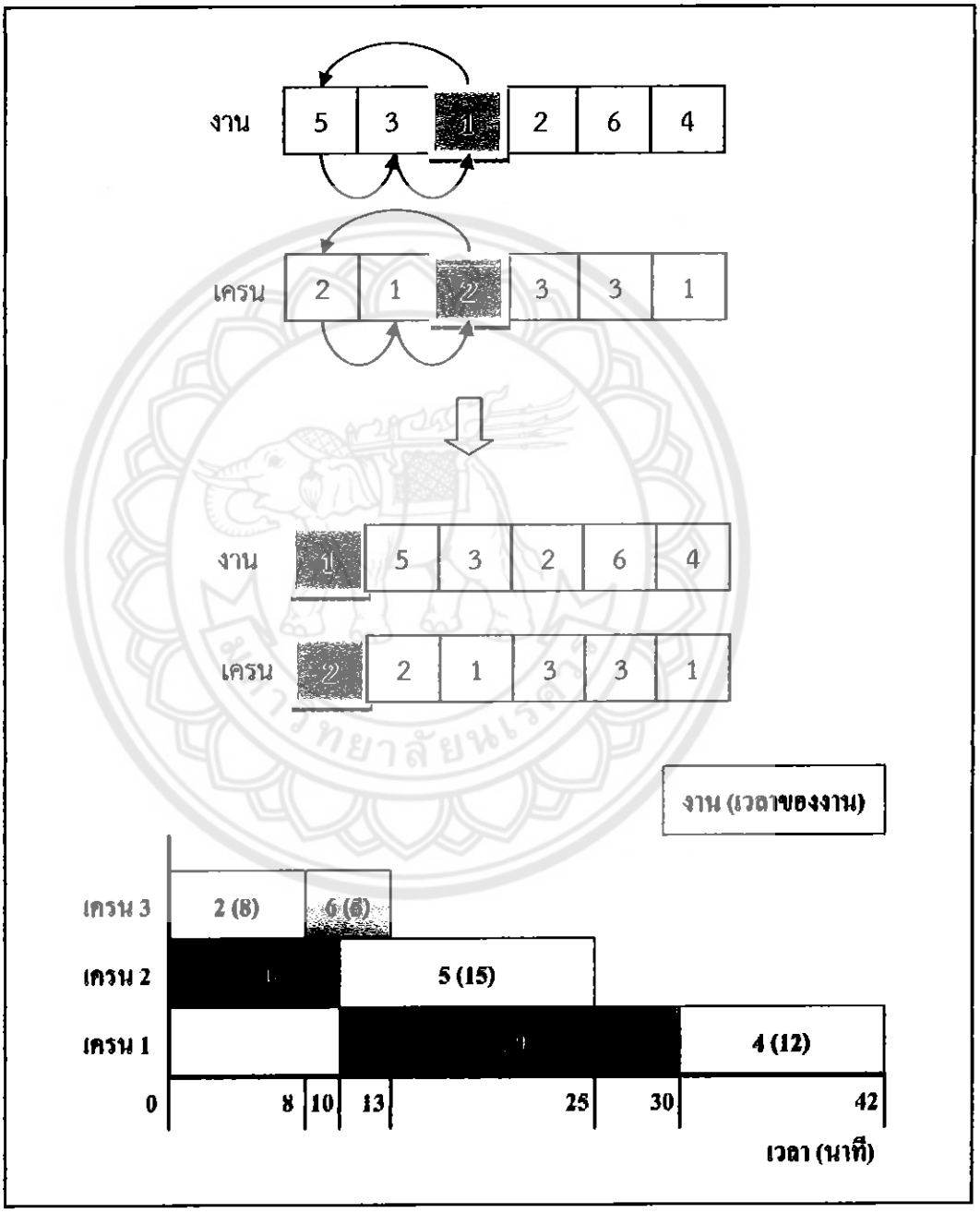
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 3 และสุ่มตำแหน่งงานที่ย้ายไป คือ ตำแหน่งที่ 5 นั่นคือ งานที่ 1 ที่ตำแหน่งที่ 3 จะย้ายไปแทนที่งานที่ 6 ที่ตำแหน่งที่ 5 และเลื่อนงานในตำแหน่งที่ 5 และ 4 ไปทางซ้ายมือ ทีละ 1 ตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 5 ดังรูป



รูปที่ 3.16 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางขวามือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 40 นาที

ค.4 เลือกพิจารณาที่ตำแหน่งที่ 3 ไปทางซ้ายมือ

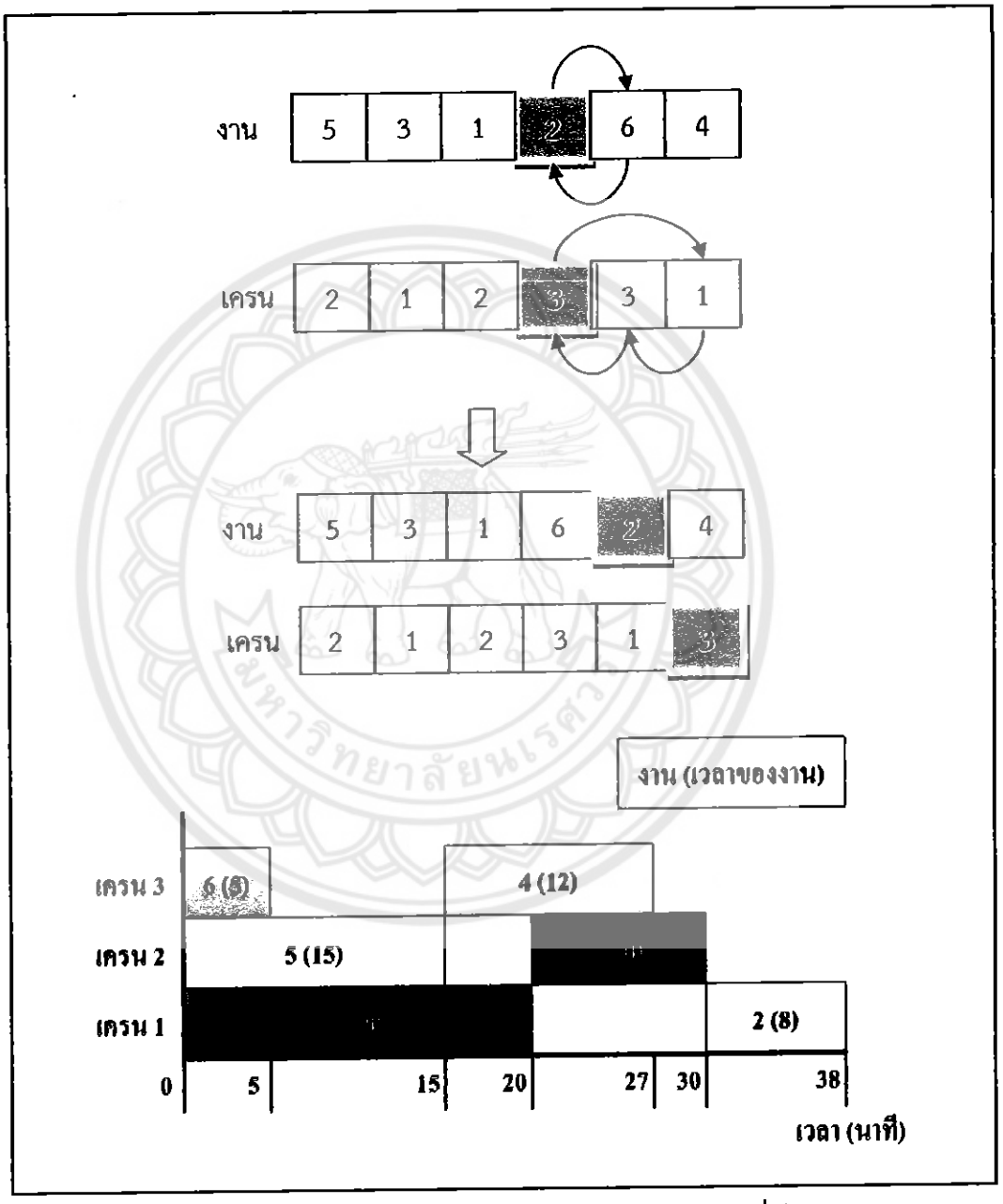
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 3 และสุ่มตำแหน่งงานที่ย้ายไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 1 นั่นคือ งานที่ 1 ที่ตำแหน่งที่ 3 จะย้ายไปแทนที่งานที่ 5 ที่ตำแหน่งแรก และเลื่อนงานในตำแหน่งแรก ไปทางขวามือ ทีละ 1 ตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 6 ดังรูป



รูปที่ 3.17 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางซ้ายมือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ค.5 เลือกพิจารณาที่ตำแหน่งที่ 4 ไปทางขวามือ

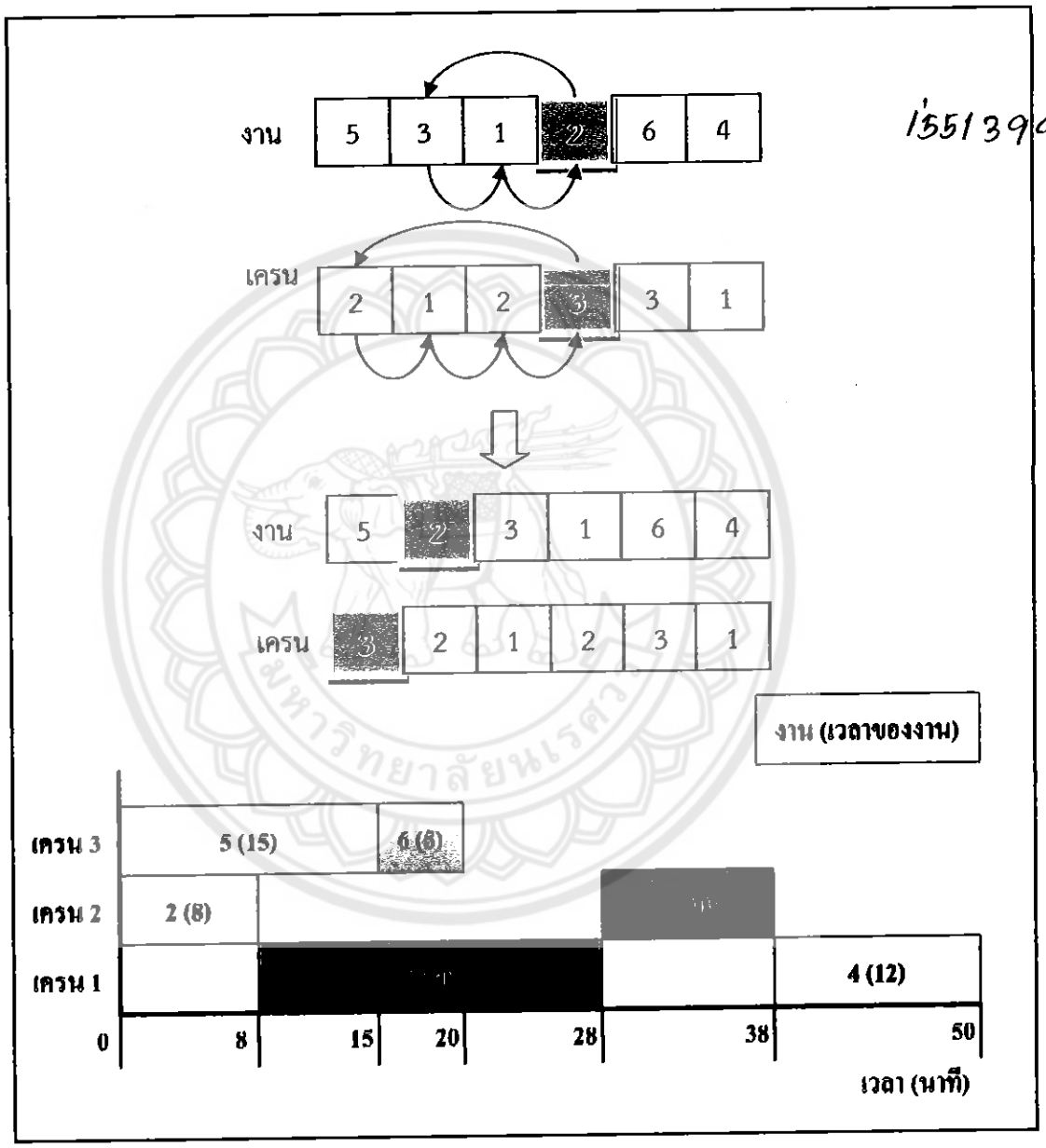
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 4 และสุ่มตำแหน่งงานที่ย้ายไป คือ ตำแหน่งที่ 5 นั่นคือ งานที่ 2 ที่ตำแหน่งที่ 4 จะย้ายไปแทนที่งานที่ 6 ที่ตำแหน่งที่ 5 และเลื่อนงานในตำแหน่งที่ 5 ไปทางซ้ายมือ 1 ตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 7 ดังรูป



รูปที่ 3.18 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางขวามือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 38 นาที

ค.6 เลือกพิจารณาที่ตำแหน่งที่ 4 ไปทางซ้ายมือ

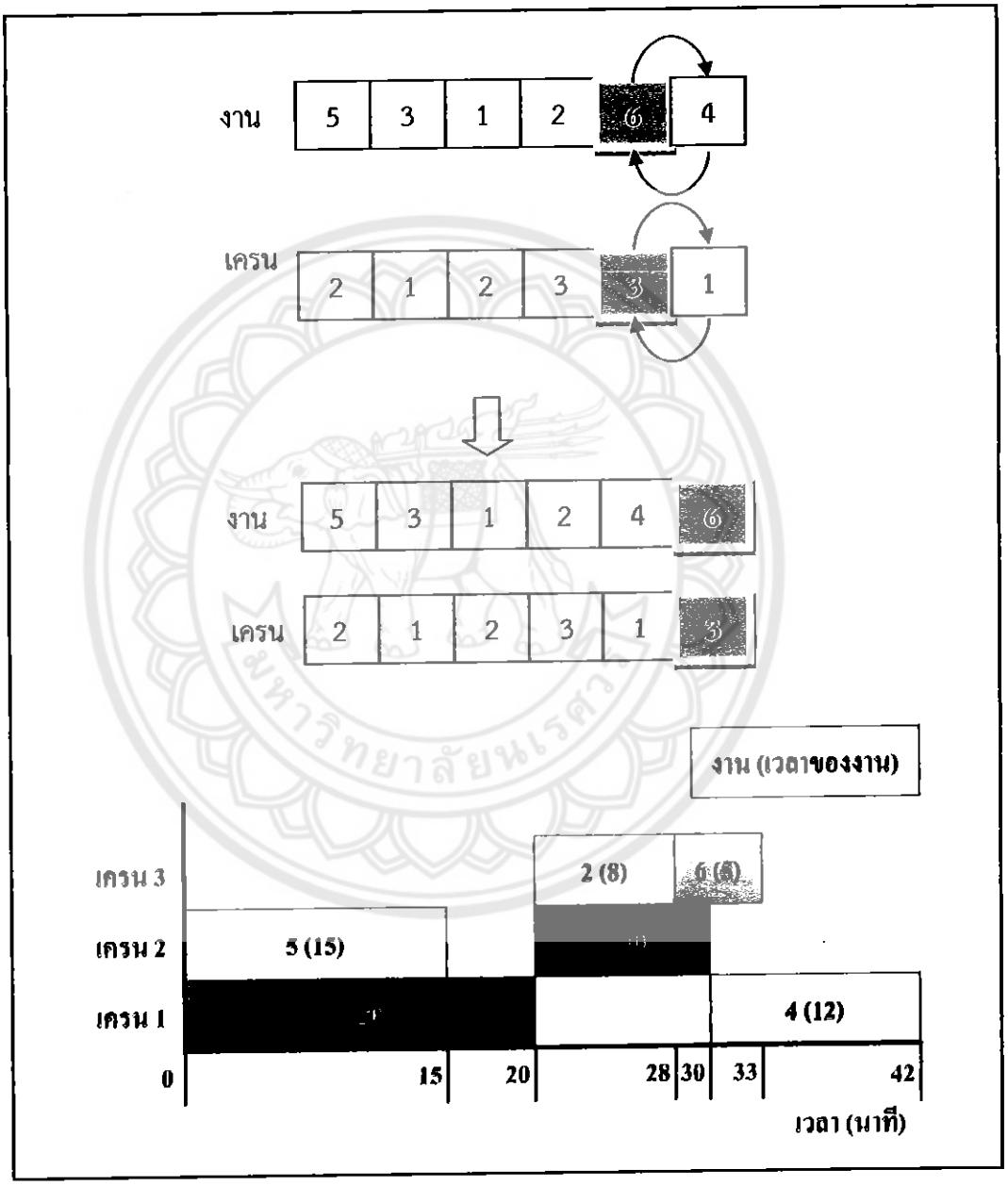
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 4 และสุ่มตำแหน่งงานที่ย้ายไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 2 นั่นคือ งานที่ 3 ที่ตำแหน่งที่ 4 จะย้ายไปแทนที่งานที่ 3 ที่ตำแหน่งที่ 2 และเลื่อนงานในตำแหน่งที่ 2 ไปทางขวามือ ทีละ 1 ตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 8 ได้ ดังรูป



รูปที่ 3.19 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางซ้ายมือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 50 นาที

ค.7 เลือกพิจารณาที่ตำแหน่งที่ 5 ไปทางขวามือ

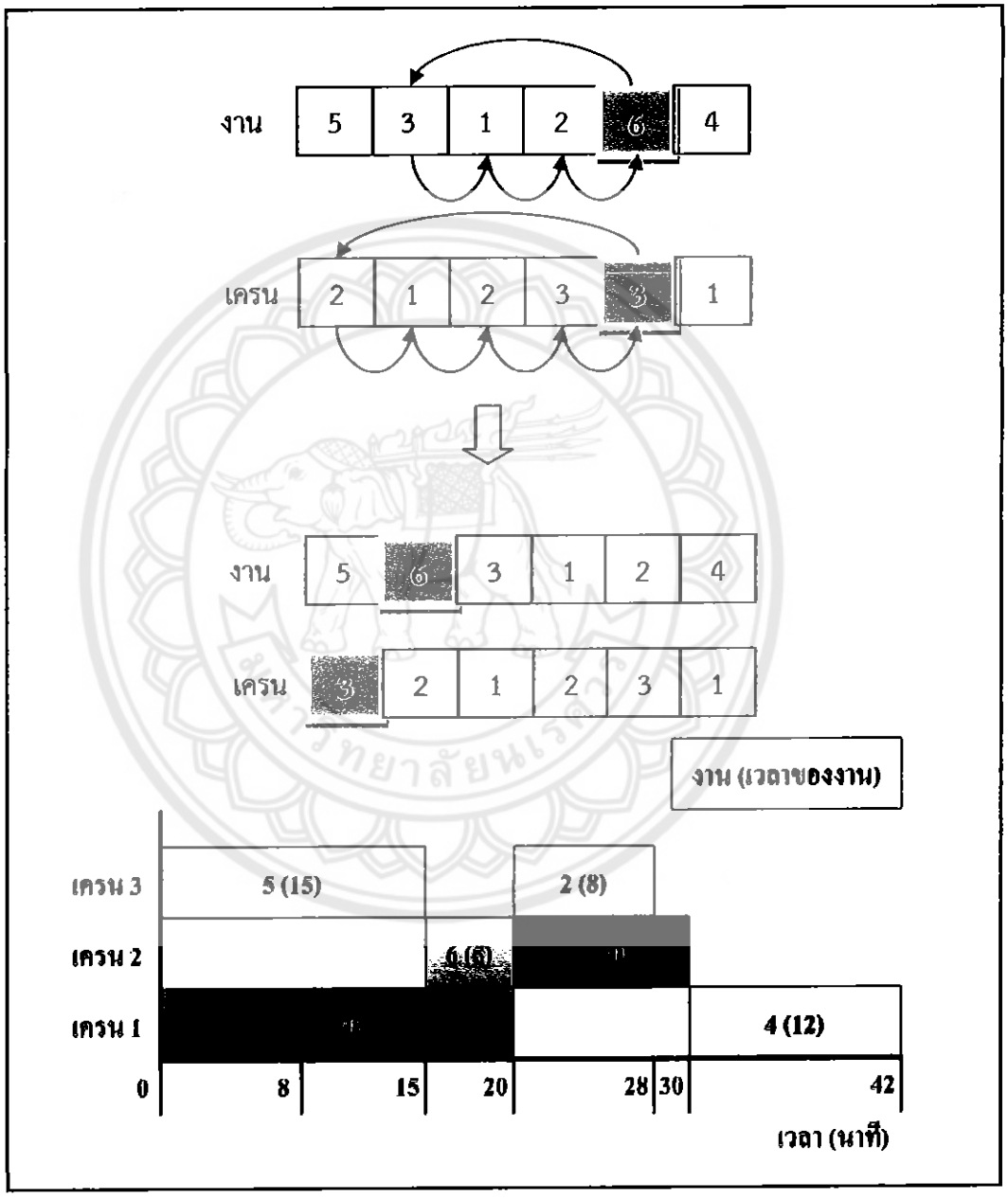
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 5 และสุ่มตำแหน่งงานที่ย้ายไป ในที่นี้มีตำแหน่งเดียว คือ ตำแหน่งที่ 6 นั่นคือ งานที่ 6 ที่ตำแหน่งที่ 5 จะย้ายไปแทนที่งานที่ 4 ที่ตำแหน่งที่ 6 และเลื่อนงานในตำแหน่งที่ 6 ไปทางซ้ายมือ 1 ตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 9 ดังรูป



รูปที่ 3.20 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางขวามือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ค.8 เลือกพิจารณาที่ตำแหน่งที่ 5 ไปทางซ้ายมือ

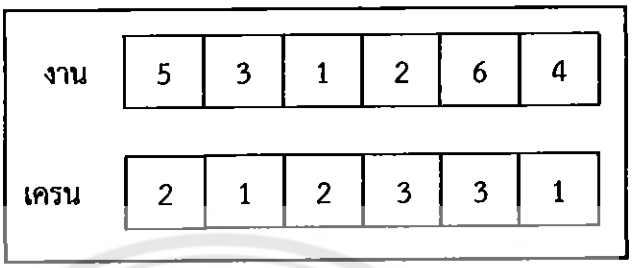
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 5 และสุ่มตำแหน่งงานที่ย้ายไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 2 นั่นคือ งานที่ 6 ที่ตำแหน่งที่ 5 จะย้ายไปแทนที่งานที่ 3 ที่ตำแหน่งที่ 2 และเลื่อนงานในตำแหน่งที่ 2 ไปทางขวามือ ทีละ 1 ตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 10 ได้ ดังรูป



รูปที่ 3.21 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางซ้ายมือ ของหลักการเลื่อนตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

3.3.2.2 วิธีหาคำตอบใหม่โดยใช้หลักการคงที่หนึ่งตำแหน่ง

ในการหาคำตอบใหม่โดยหลักการคงที่ 1 ตำแหน่งนั้น จะเป็นการสุ่มตำแหน่งต่างๆ ในงานหรือเครนนั้นๆ แล้วให้คงที่ตำแหน่งที่สุ่มไว้ แล้วให้นำตำแหน่งที่พิจารณาไปแทนที่ตำแหน่งสุดท้ายหรือตำแหน่งแรก หลังจากนั้นจึงทำการเลื่อนตำแหน่งไปลำดับถัดไป ดังเช่นตัวอย่างเดิม มีงานจำนวน 6 งาน และเครนจำนวน 3 เครน จะสามารถหาคำตอบใหม่ได้ ดังต่อไปนี้



รูปที่ 3.22 ตัวอย่างการหาคำตอบใหม่โดยหลักการคงที่หนึ่งตำแหน่ง

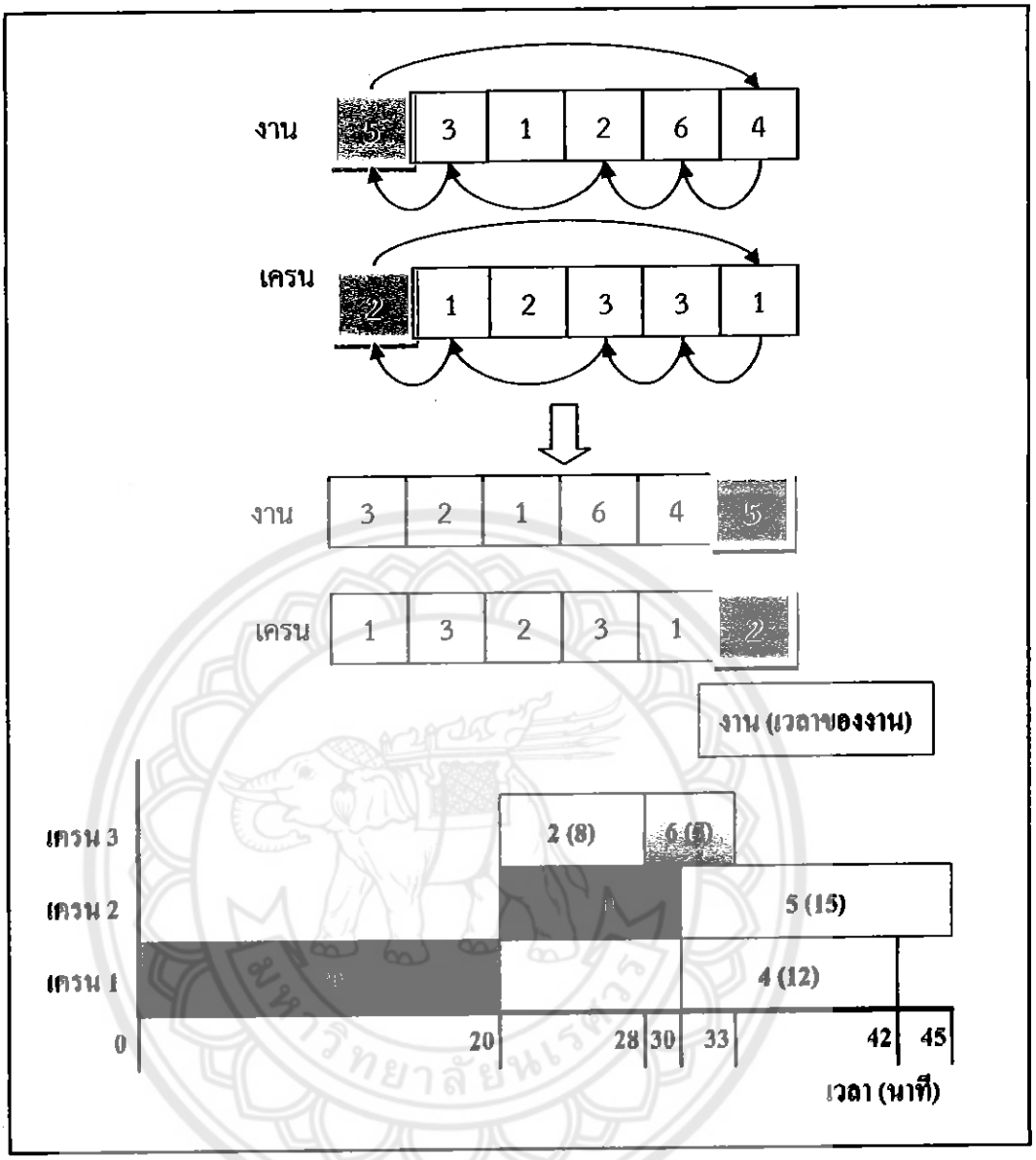
เราสามารถหาค่า X_{new} ได้ทั้งหมด 2 (n-1) คำตอบ โดยที่

n = จำนวนงาน

ดังนั้น จากตัวอย่างนี้ เราสามารถหา X_{new} ได้ทั้งหมด 10 คำตอบ ซึ่งสามารถพิจารณาตำแหน่งการสุ่มได้เป็น 3 ส่วน ดังต่อไปนี้

ก. พิจารณาที่ตำแหน่งแรก

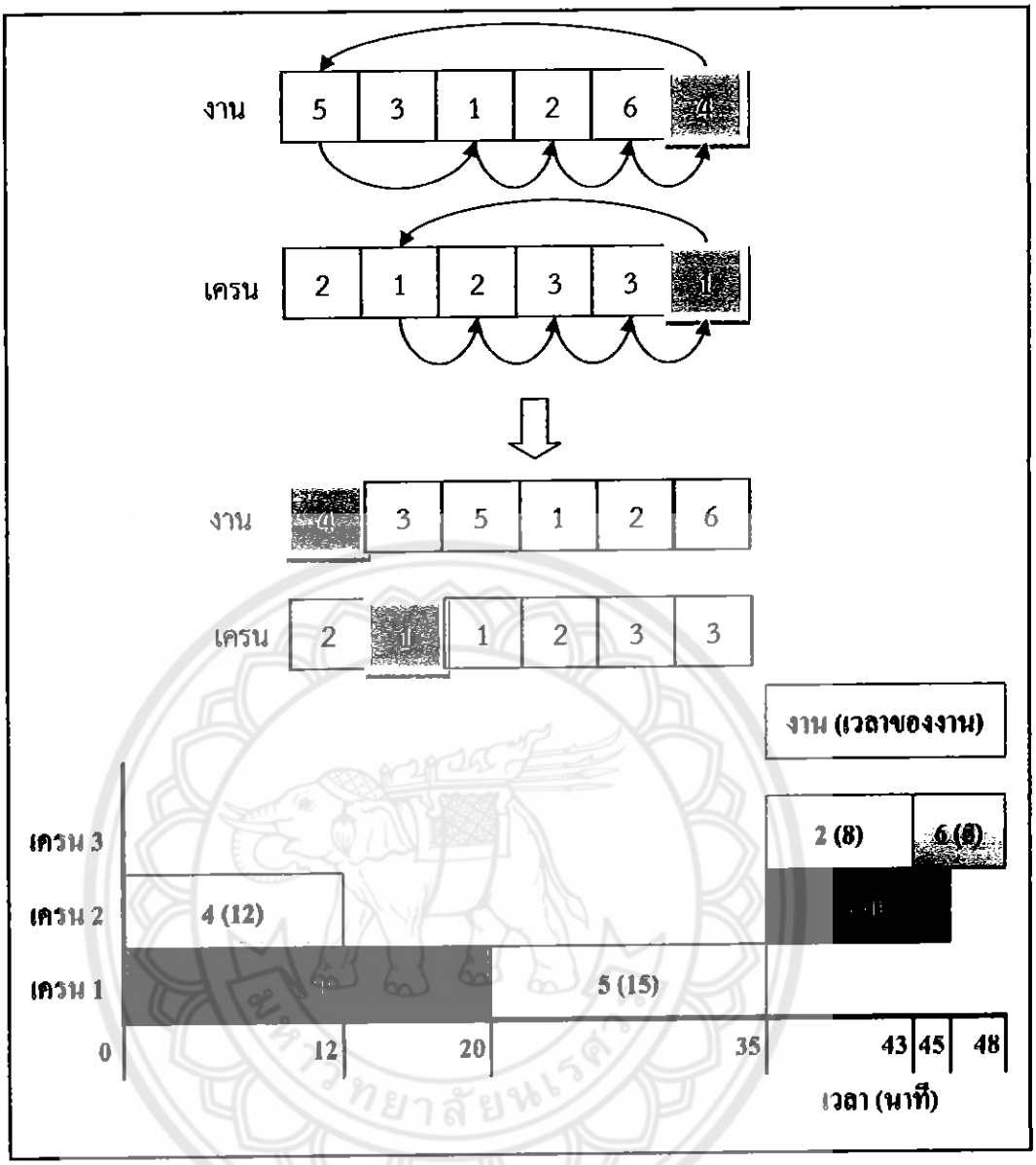
ในการพิจารณาที่ตำแหน่งแรกนั้น จากตัวอย่าง เมื่อพิจารณาที่ตำแหน่งแรก จะสามารถทำการสุ่มได้ 5 ตำแหน่ง ไปทางขวามือ โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป โดยผลของการสุ่มจากตัวอย่างนี้คือ ที่ตำแหน่งของงาน สุ่มได้ตำแหน่งที่ 3 คืองานที่ 1 จะทำการคงที่ตำแหน่งนี้ไว้ แล้วทำการย้ายงาน 5 จากตำแหน่งที่ 1 ไปแทนที่งานที่ 4 ที่ตำแหน่งสุดท้าย และเลื่อนงานในตำแหน่งที่ 5, 4 และ 2 ไปทางซ้ายมือทีละตำแหน่ง โดยไม่คำนึงถึงตำแหน่งที่ทำการคงที่ไว้ และที่ตำแหน่งของเครน สุ่มได้ตำแหน่งที่ 3 คือเครนที่ 2 ก็ย้ายตำแหน่งลักษณะเช่นเดียวกันกับตำแหน่งของงาน เช่นเดียวกัน ดังรูป และจะได้คำตอบใหม่ คำตอบที่ 1 คือ



รูปที่ 3.23 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งแรกของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 45 นาที

ข. พิจารณาที่ตำแหน่งสุดท้าย

ในการพิจารณาที่ตำแหน่งสุดท้ายนั้น จากตัวอย่างเดิม เมื่อพิจารณาที่ตำแหน่งสุดท้าย จะทำการสุ่มตำแหน่งได้ 5 ตำแหน่ง ไปทางซ้ายมือ โดยพิจารณาทั้งตำแหน่งของงานและตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป เช่นเดียวกับที่พิจารณาที่ตำแหน่งแรก จากตัวอย่างนี้ สุ่มได้ตำแหน่งที่ 2 คือ งานที่ 3 จะทำการคงที่งานที่ 3 ที่ตำแหน่งที่ 2 ไว้ และจะทำการย้ายงานที่ 4 ที่ตำแหน่งสุดท้ายไปยังตำแหน่งแรก หลังจากนั้นเลื่อนงานที่ตำแหน่งที่ 1, 3, 4 และ 5 ไปทางซ้าย ทีละ 1 ตำแหน่ง โดยไม่คำนึงถึงงานที่ทำการคงที่ไว้ และที่ตำแหน่งของเครนเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 2 ได้ ดังรูป



รูปที่ 3.24 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งสุดท้ายของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 48 นาที

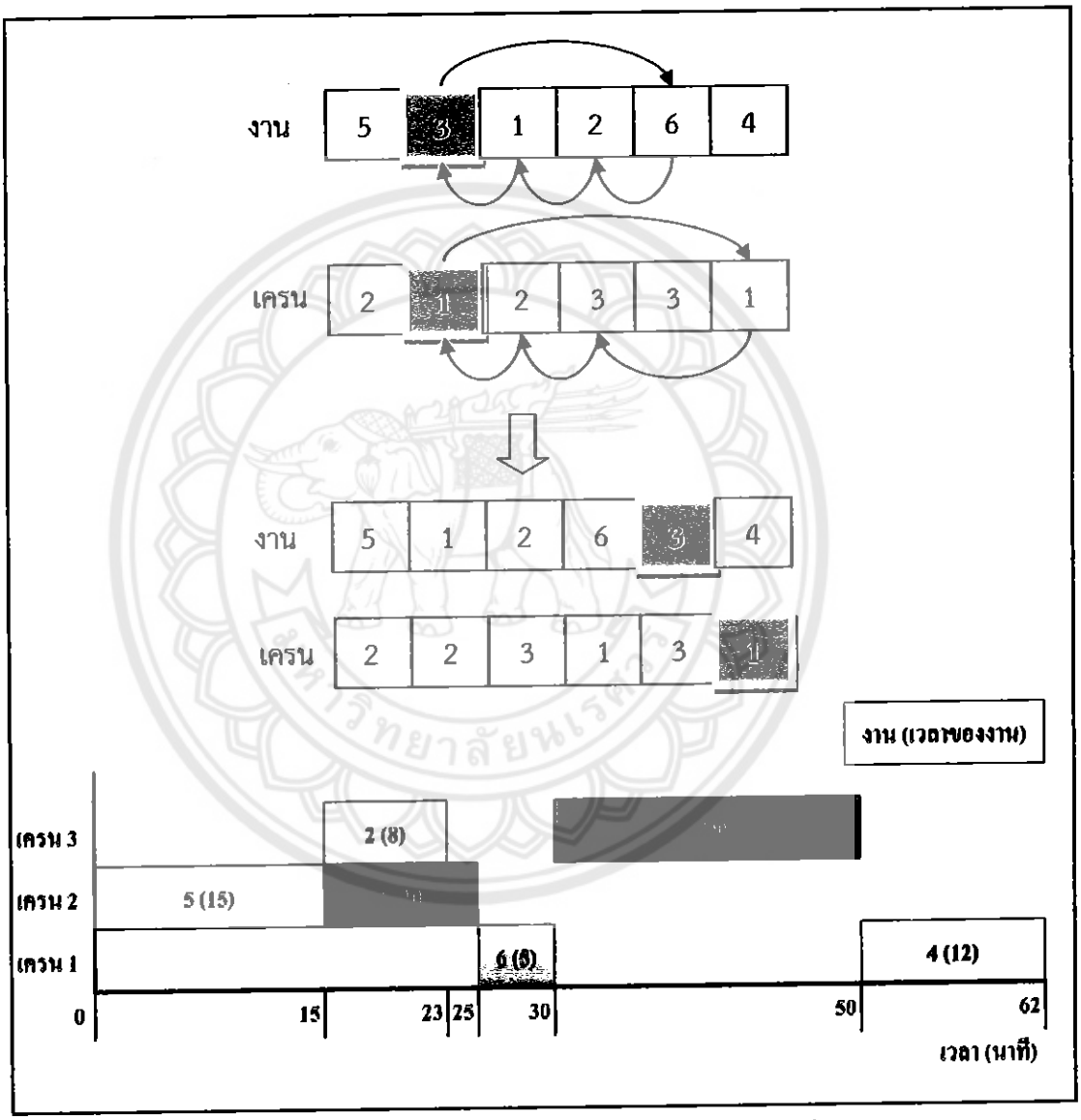
ค. พิจารณาที่ตำแหน่งกลาง

ในการพิจารณาในตำแหน่งกลางนั้น จะสามารถพิจารณาได้จาก ตำแหน่งทั้งหมดที่อยู่ระหว่างตำแหน่งแรกกับตำแหน่งสุดท้าย ซึ่งในแต่ละตำแหน่งนั้นเราสามารถหาคำตอบได้ 2 คำตอบ โดยการเลื่อนงานและเครนในตำแหน่งที่เราสนใจไปทางขวา และทางซ้ายมือ โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป

จากตัวอย่างเดิม ตำแหน่งกลางที่อยู่ระหว่างตำแหน่งแรกและตำแหน่งสุดท้ายมี 4 ตำแหน่ง คือ ตำแหน่งที่ 2, 3, 4 และ 5 ซึ่งในแต่ละตำแหน่งนั้น สามารถหาคำตอบได้ 2 คำตอบ คือ การเลื่อนงานและเครนไปทางซ้ายและขวามือ ดังนั้นจะสามารถพิจารณาในแต่ละตำแหน่ง คือ

ค.1 เลือกพิจารณาที่ตำแหน่งที่ 2 ไปทางขวามือ

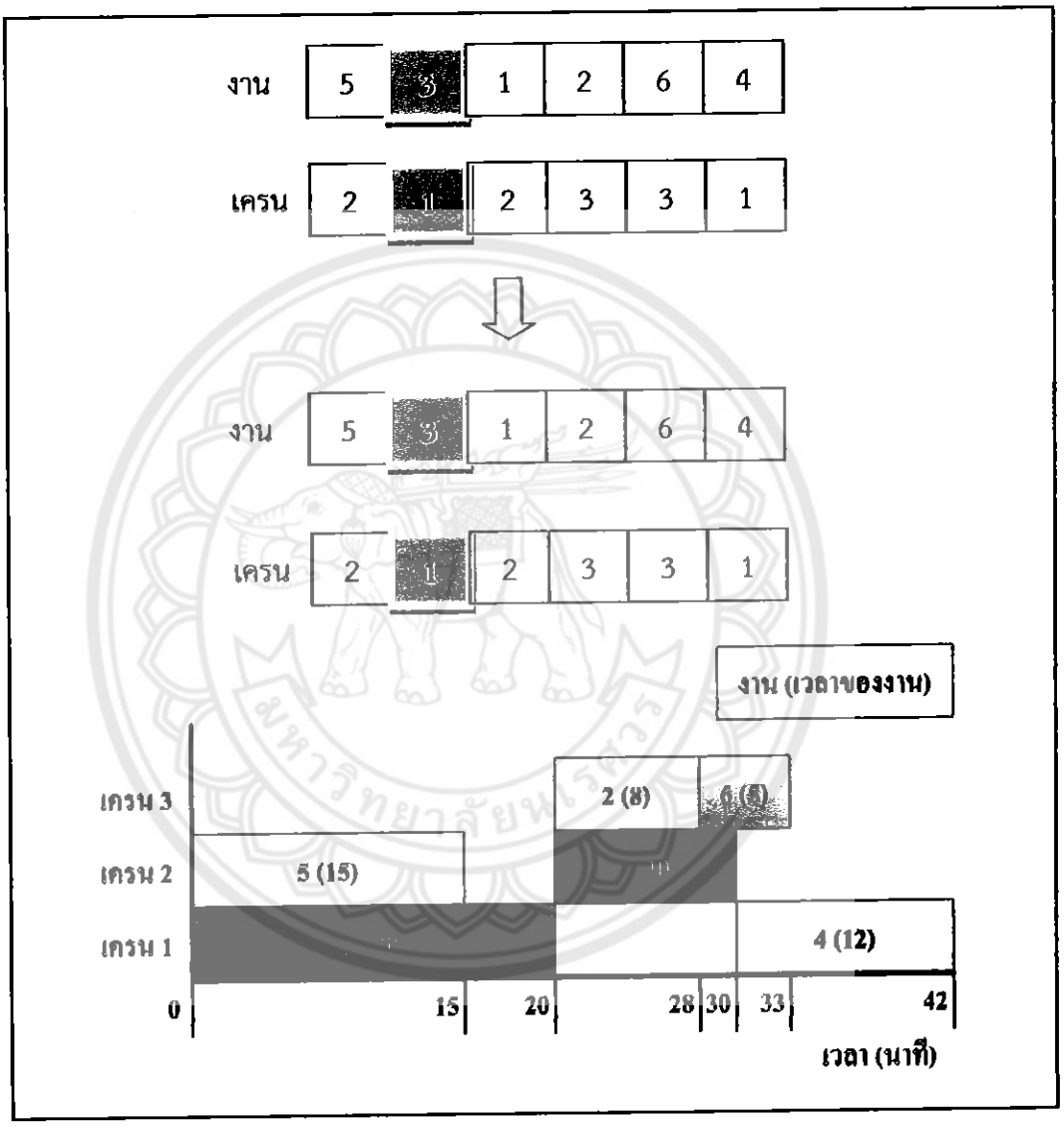
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 2 และสุ่มตำแหน่งงานคงที่ คือ ตำแหน่งสุดท้าย นั่นคือ งานที่ 4 ตำแหน่งสุดท้ายจะคงที่ และงานที่ 3 ที่ตำแหน่งที่ 2 จะย้ายไปแทนที่งานที่ 6 ที่ตำแหน่งที่ 5 และเลื่อนงานในตำแหน่งที่ 5, 4, 3 ไปทางซ้ายมือทีละ 1 ตำแหน่ง โดยไม่คำนึงถึงงานที่ทำการคงที่ไว้ และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 3 ดังรูป



รูปที่ 3.25 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางขวามือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 62 นาที

ค.2 เลือกพิจารณาที่ตำแหน่งที่ 2 ไปทางซ้ายมือ

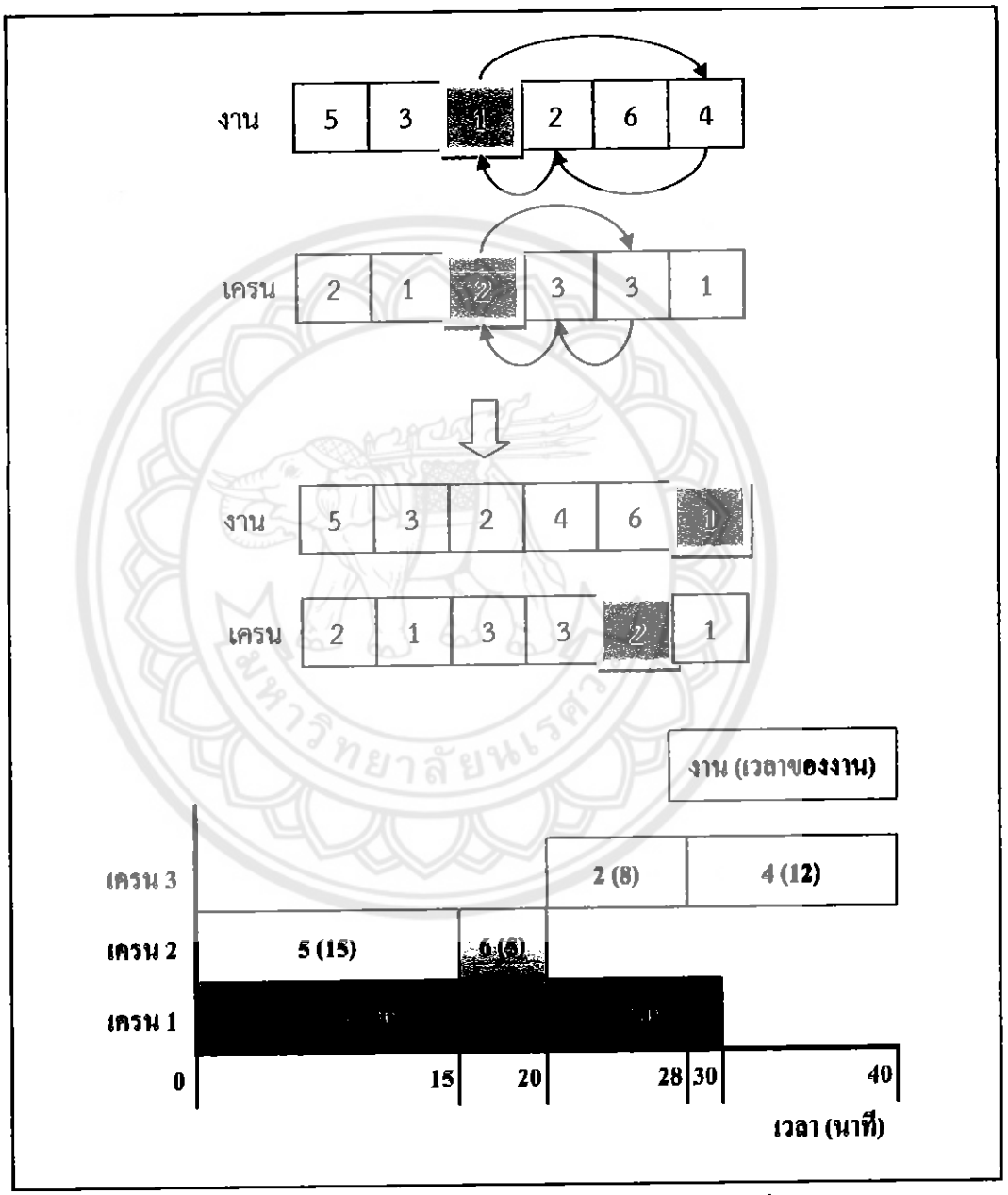
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 2 และสุ่มตำแหน่งงานที่จะคงที่ไปทางซ้ายมือ ในที่นี้มีงานเดียว คือ ที่งานที่ตำแหน่งแรก นั่นคือ งานที่ 3 ที่ตำแหน่งที่ 2 จะไม่สามารถย้ายไปแทนที่ตำแหน่งใดได้ จึงทำให้คำตอบใหม่คือคำตอบเดิม ซึ่งเป็นคำตอบที่ 4 ดังรูป



รูปที่ 3.26 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางซ้ายมือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ค.3 เลือกพิจารณาที่ตำแหน่งที่ 3 ไปทางขวามือ

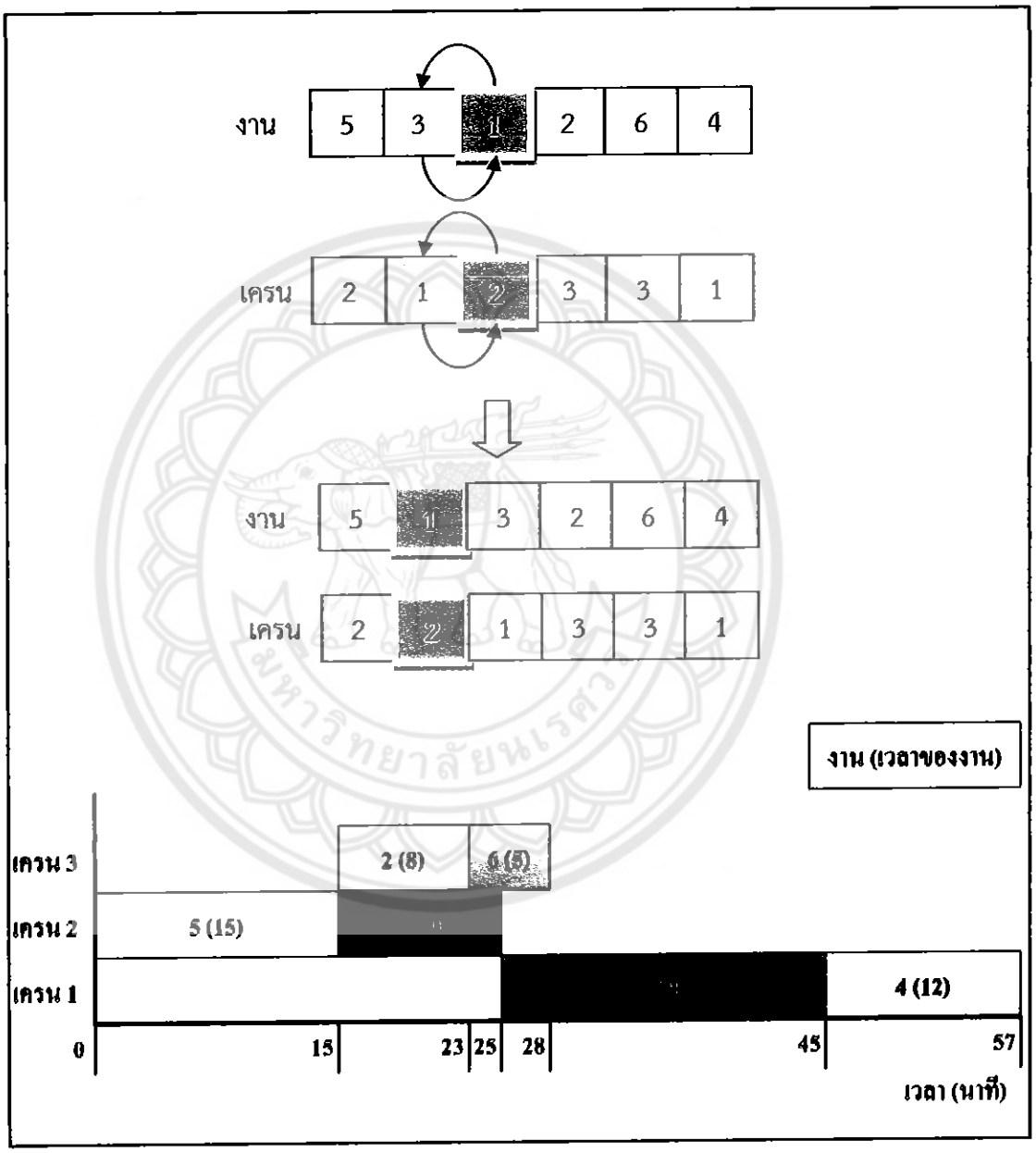
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 3 และสุ่มตำแหน่งงานที่ทำการคงที่ คือ ตำแหน่งที่ 5 นั่นคือ งานที่ 1 ที่ตำแหน่งที่ 3 จะย้ายไปแทนที่งานที่ 4 ที่ตำแหน่งสุดท้าย และเลื่อนงานในตำแหน่งที่ 6 และ 5 ไปทางซ้ายมือ ทีละ 1 ตำแหน่ง โดยไม่คำนึงถึงงานที่ทำการคงที่ไว้ และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 5 ดังรูป



รูปที่ 3.27 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางขวามือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 40 นาที

ค.4 เลือกพิจารณาที่ตำแหน่งที่ 3 ไปทางซ้ายมือ

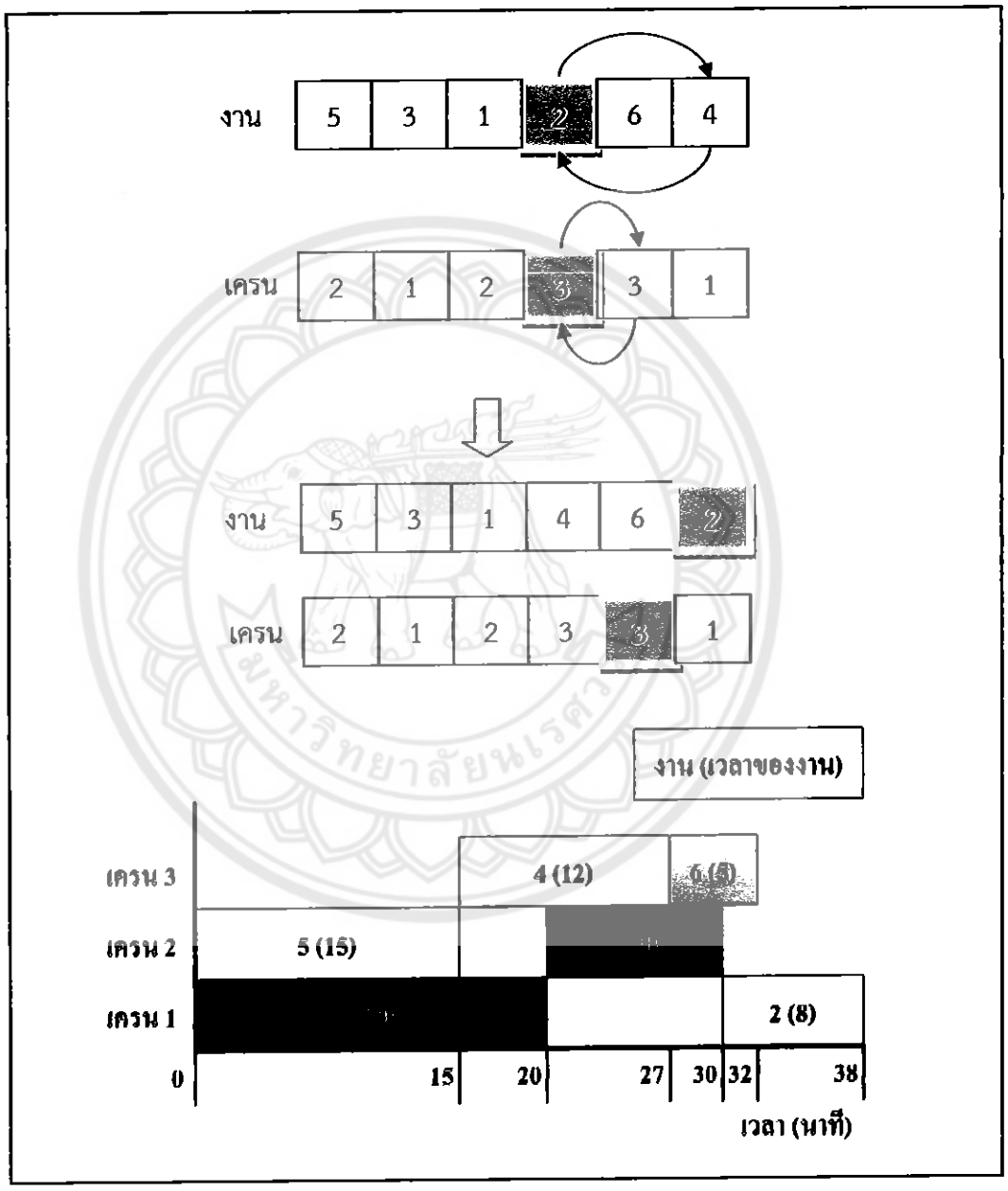
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 3 และสุ่มตำแหน่งงานที่จะทำการคงที่ไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 1 นั่นคือ งานที่ 1 ที่ตำแหน่งที่ 3 จะย้ายไปแทนที่งานที่ 3 ที่ตำแหน่งที่ 2 และเลื่อนงานในตำแหน่งที่ 2 ไปทางขวามือ ทีละ 1 ตำแหน่ง โดยไม่คำนึงถึงงานที่ทำการคงที่ไว้และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 6 ดังรูป



รูปที่ 3.28 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางซ้ายมือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 57 นาที

ค.5 เลือกพิจารณาที่ตำแหน่งที่ 4 ไปทางขวามือ

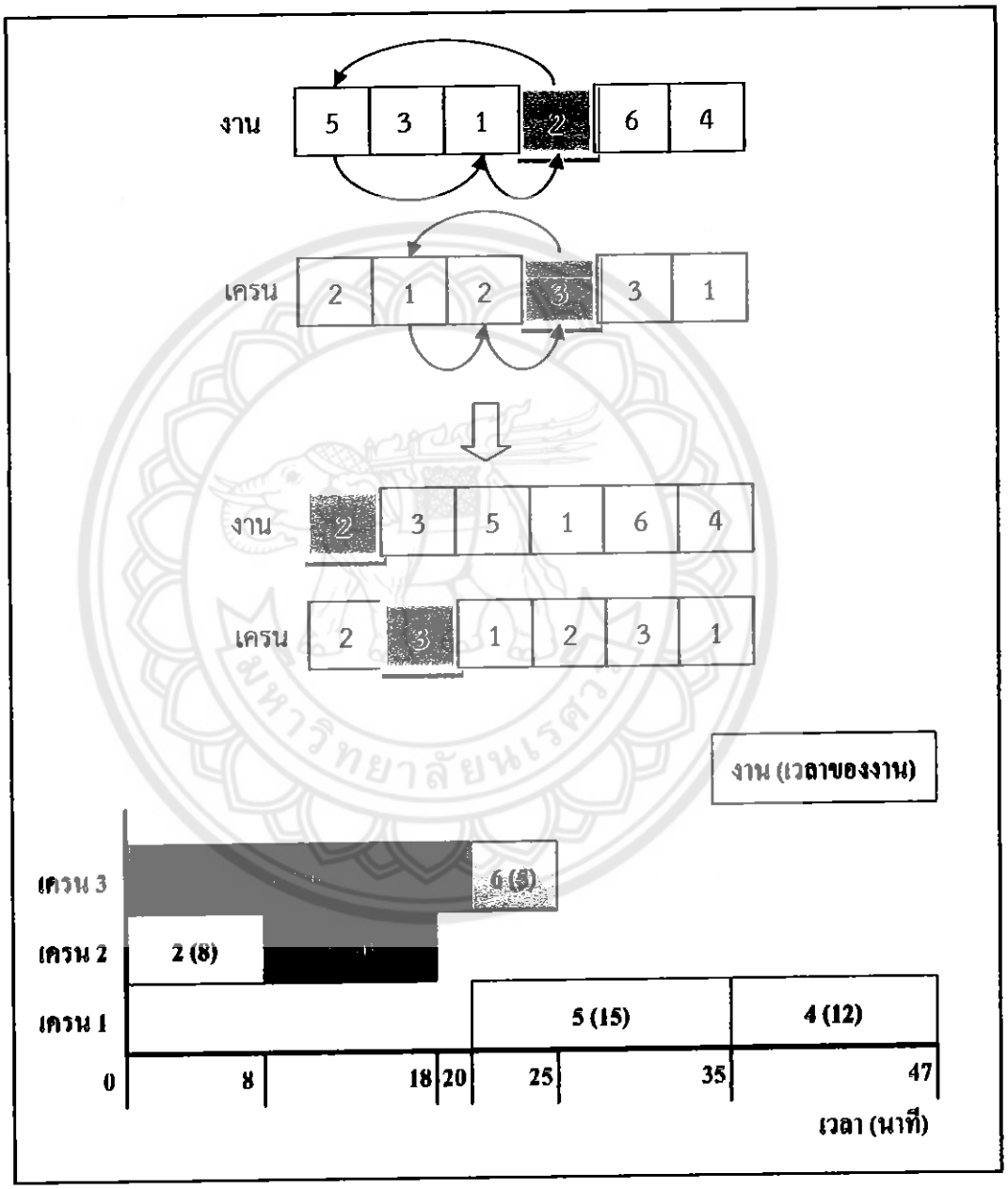
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 4 และสุ่มตำแหน่งงานที่ทำการคงที่ คือ ตำแหน่งที่ 5 นั่นคือ งานที่ 2 ที่ตำแหน่งที่ 4 จะย้ายไปแทนที่งานที่ 4 ที่ตำแหน่งสุดท้าย และเลื่อนงานในตำแหน่งที่ 6 ไปทางซ้ายมือ 1 ตำแหน่ง โดยไม่คำนึงถึงงานที่ทำการคงที่ไว้ และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 7 ดังรูป



รูปที่ 3.29 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางขวามือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 38 นาที

ค.6 เลือกพิจารณาที่ตำแหน่งที่ 4 ไปทางซ้ายมือ

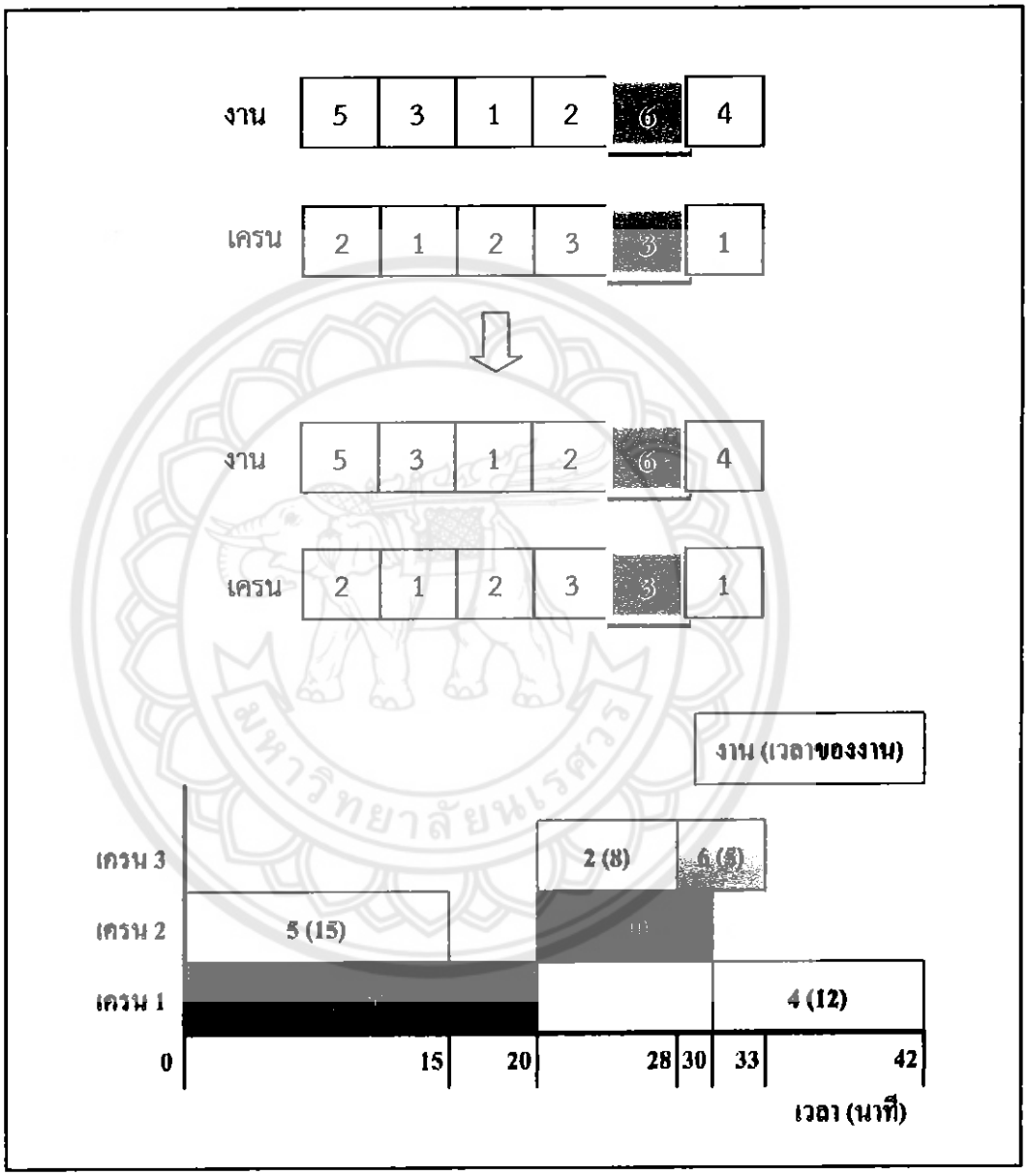
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 4 และสุ่มตำแหน่งงานที่ทำการคงที่ไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 2 นั่นคือ งานที่ 3 ที่ตำแหน่งที่ 4 จะย้ายไปแทนที่งานที่ 5 ที่ตำแหน่งแรก และเลื่อนงานในตำแหน่งที่ 1, 3 ไปทางขวามือ ทีละ 1 ตำแหน่ง โดยไม่คำนึงถึงงานที่ทำการคงที่ไว้ และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 8 ได้ ดังรูป



รูปที่ 3.30 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางซ้ายมือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 47 นาที

ค.7 เลือกพิจารณาที่ตำแหน่งที่ 5 ไปทางขวามือ

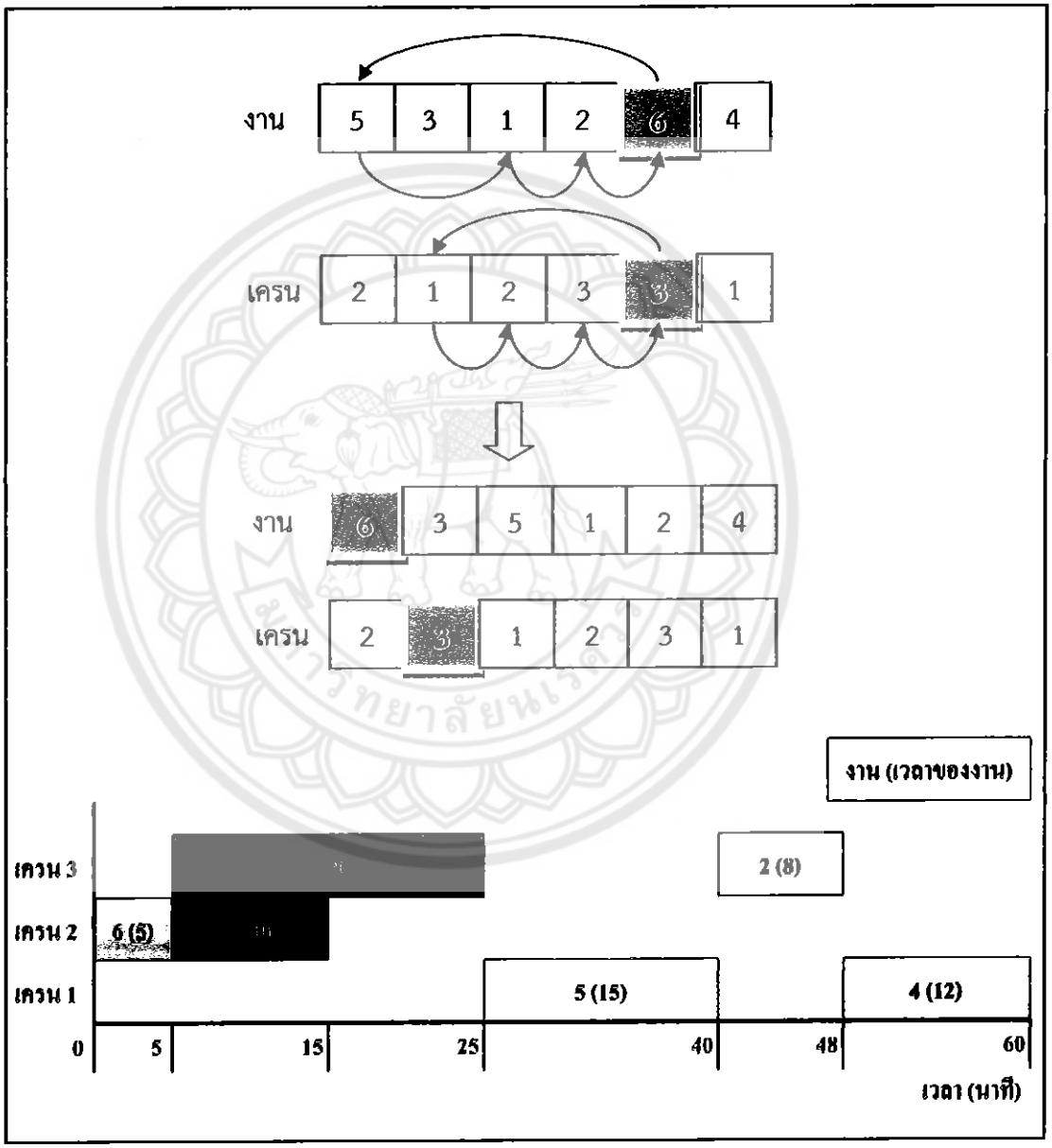
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 5 และสุ่มตำแหน่งงานที่ทำการคงที่ ในที่นี้มีตำแหน่งเดียว คือ ตำแหน่งที่ 6 นั่นคือ งานที่ 6 ที่ตำแหน่งที่ 5 จะไม่สามารถย้ายไปแทนที่ที่ใดได้คำตอบใหม่ที่ได้ คือคำตอบเดิม ซึ่งเป็นคำตอบที่ 9 ดังรูป



รูปที่ 3.31 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางขวามือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ค.8 เลือกพิจารณาที่ตำแหน่งที่ 5 ไปทางซ้ายมือ

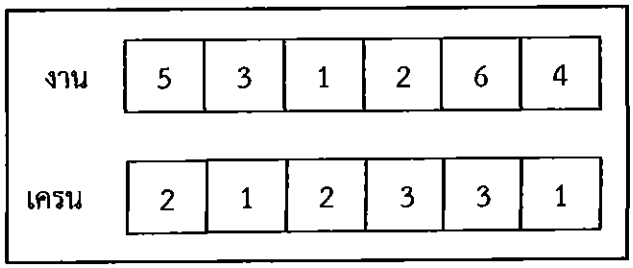
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 5 และสุ่มตำแหน่งงานที่จะทำการคงที่ไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 2 นั่นคือ งานที่ 6 ที่ตำแหน่งที่ 5 จะย้ายไปแทนที่งานที่ 5 ที่ตำแหน่งแรก และเลื่อนงานในตำแหน่งที่ 1, 3, 4 ไปทางขวามือ ทีละ 1 ตำแหน่ง โดยไม่คำนึงถึงงานที่ทำการคงที่ไว้ และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 10 ได้ ดังรูป



รูปที่ 3.32 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางซ้ายมือ ของหลักการคงที่หนึ่งตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 60 นาที

3.3.2.3 วิธีหาคำตอบใหม่โดยใช้หลักการสลับที่

ในการหาคำตอบใหม่โดยหลักการสลับที่นั้น จะเป็นการสุ่มตำแหน่งต่างๆ ในงานหรือเครนนั้นๆ แล้วให้สลับที่กับตำแหน่งที่พิจารณาไป ดังเช่นตัวอย่างเดิม มีงานจำนวน 6 งาน และเครนจำนวน 3 เครน จะสามารถหาคำตอบใหม่ได้ ดังต่อไปนี้



รูปที่ 3.33 ตัวอย่างการหาคำตอบใหม่โดยหลักการสลับที่

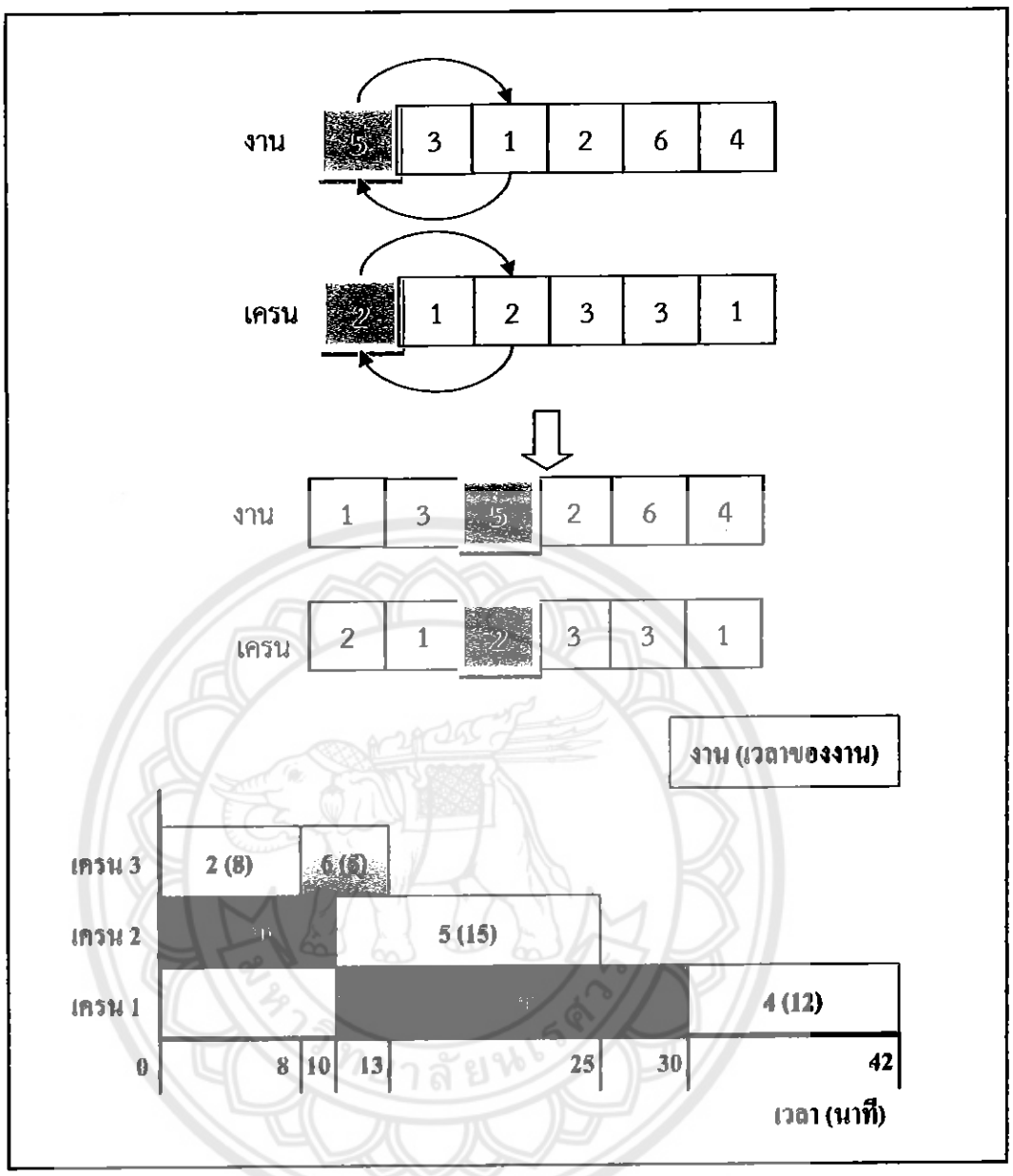
เราสามารถหาค่า X_{new} ได้ทั้งหมด 2 (n-1) คำตอบ โดยที่

n = จำนวนงาน

ดังนั้น จากตัวอย่างนี้ เราสามารถหา X_{new} ได้ทั้งหมด 10 คำตอบ ซึ่งสามารถพิจารณาตำแหน่งการสุ่มได้เป็น 3 ส่วน ดังต่อไปนี้

ก. พิจารณาที่ตำแหน่งแรก

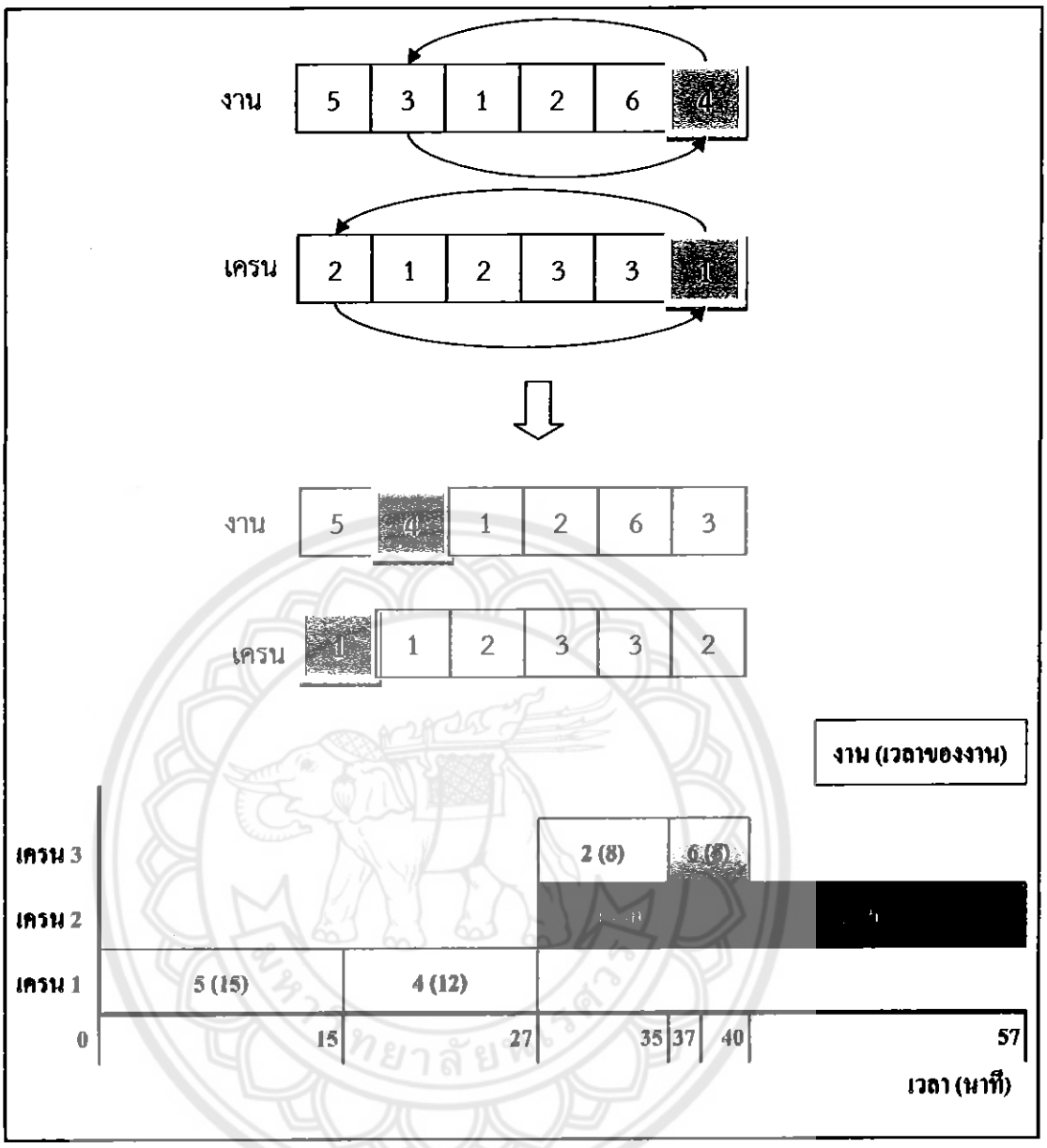
ในการพิจารณาที่ตำแหน่งแรกนั้น จากตัวอย่าง เมื่อพิจารณาที่ตำแหน่งแรกจะสามารถทำการสุ่มได้ 5 ตำแหน่ง ไปทางขวามือ โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป โดยผลของการสุ่มจากตัวอย่างนี้คือ ที่ตำแหน่งของงาน สุ่มได้ตำแหน่งที่ 3 คืองานที่ 1 จะทำการสลับที่งานที่ 5 ตำแหน่งแรก กับงานที่ 1 ตำแหน่งที่ 3 และที่ตำแหน่งของเครน สุ่มได้ตำแหน่งที่ 3 คือเครนที่ 2 ก็สลับที่ตำแหน่งลักษณะเช่นเดียวกันกับตำแหน่งของงาน เช่นเดียวกัน ดังรูป และจะได้คำตอบใหม่คำตอบที่ 1 คือ



รูปที่ 3.34 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งแรกของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ข. พิจารณาที่ตำแหน่งสุดท้าย

ในการพิจารณาที่ตำแหน่งสุดท้ายนั้น จากตัวอย่างเดิม เมื่อพิจารณาที่ตำแหน่งสุดท้าย จะทำการสุ่มตำแหน่งได้ 5 ตำแหน่ง ไปทางซ้ายมือ โดยพิจารณาทั้งตำแหน่งของงานและตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป เช่นเดียวกับที่พิจารณาที่ตำแหน่งแรก จากตัวอย่างนี้ สุ่มได้ตำแหน่งที่ 2 คือ งานที่ 3 จะทำการคงที่งานที่ 3 ที่ตำแหน่งที่ 2 ไว้ และจะทำการย้ายงานที่ 4 ที่ตำแหน่งสุดท้ายไปยังตำแหน่งแรก หลังจากนั้นเลื่อนงานที่ตำแหน่งที่ 1, 3, 4 และ 5 ไปทางซ้าย ทีละ 1 ตำแหน่ง โดยไม่คำนึงถึงงานที่ทำการคงที่ไว้ และที่ตำแหน่งของเครนเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 2 ได้ ดังรูป



รูปที่ 3.35 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งสุดท้ายของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

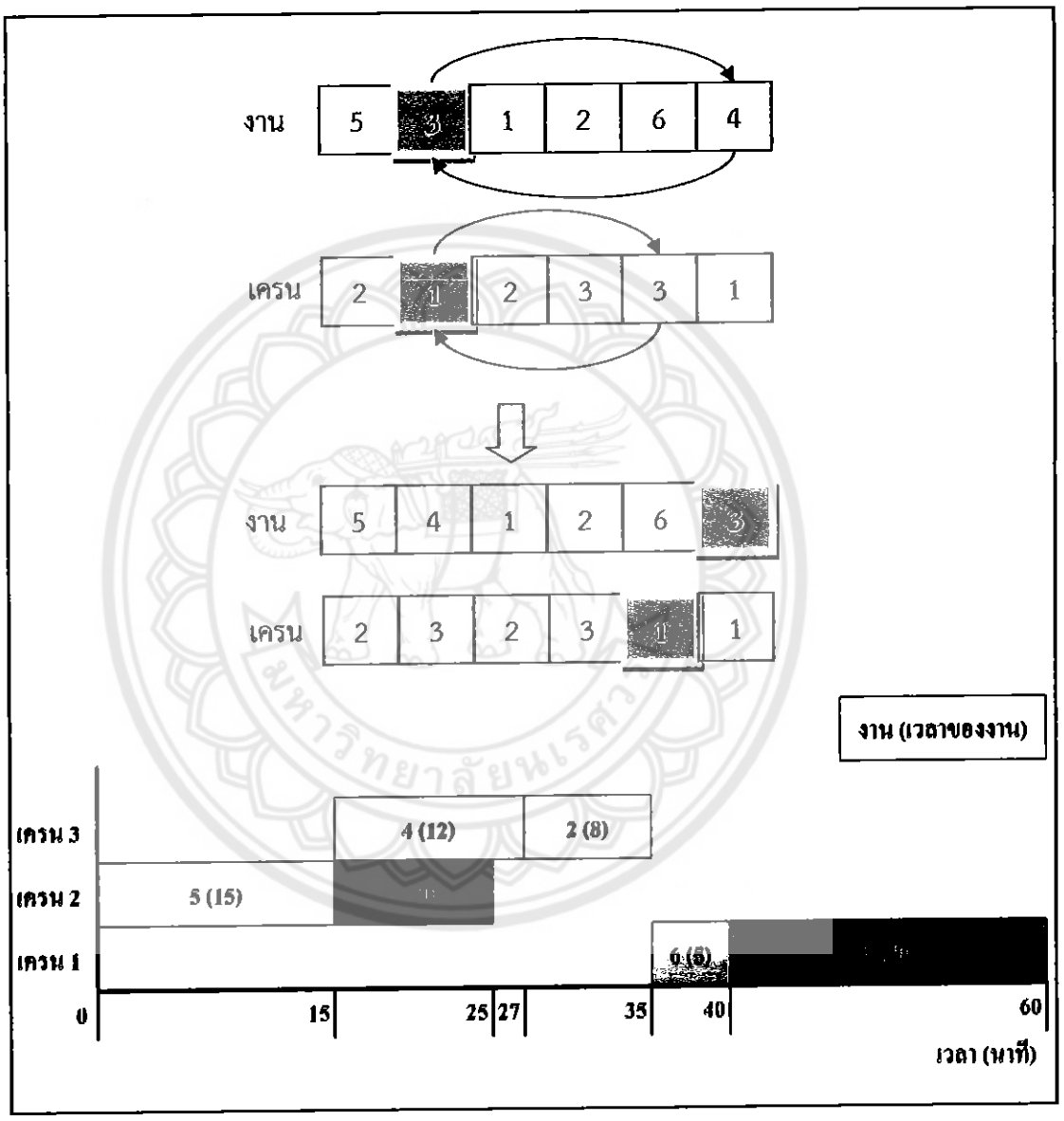
ค. พิจารณาที่ตำแหน่งกลาง

ในการพิจารณาในตำแหน่งกลางนั้น จะสามารถพิจารณาได้จาก ตำแหน่งทั้งหมดที่อยู่ระหว่างตำแหน่งแรกกับตำแหน่งสุดท้าย ซึ่งในแต่ละตำแหน่งนั้นเราสามารถหาคำตอบได้ 2 คำตอบ โดยการสลับงานและเครนในตำแหน่งที่เราสนใจไปทางขวา และทางซ้ายมือกับตำแหน่งที่สุ่มได้ โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป

จากตัวอย่างเดิม ตำแหน่งกลางที่อยู่ระหว่างตำแหน่งแรกและตำแหน่งสุดท้าย มี 4 ตำแหน่ง คือ ตำแหน่งที่ 2, 3, 4 และ 5 ซึ่งในแต่ละตำแหน่งนั้น สามารถหาคำตอบได้ 2 คำตอบคือ การสลับงานและเครนไปทางซ้ายและขวามือ ดังนั้นจะสามารถพิจารณาได้ในแต่ละตำแหน่ง คือ

ค.1 เลือกพิจารณาที่ตำแหน่งที่ 2 ไปทางขวามือ

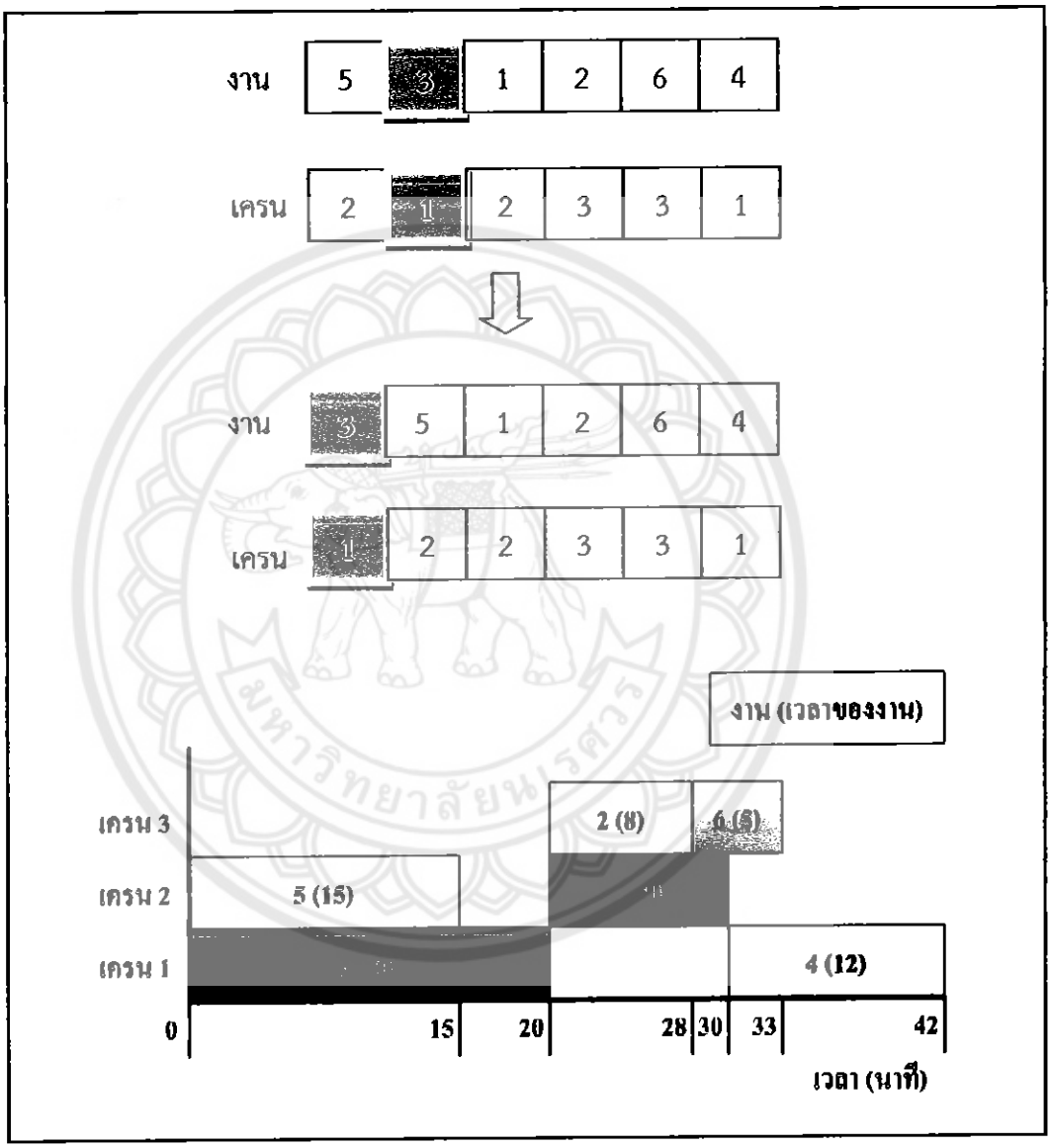
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 2 และสุ่มตำแหน่งงานที่ต้องการสลับ คือ ตำแหน่งสุดท้าย นั่นคือ งานที่ 3 ที่ตำแหน่งที่ 2 จะย้ายไปสลับที่กับงานที่ 4 ที่ตำแหน่งสุดท้าย และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 3 ดังรูป



รูปที่ 3.36 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางขวามือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 60 นาที

ค.2 เลือกพิจารณาที่ตำแหน่งที่ 2 ไปทางซ้ายมือ

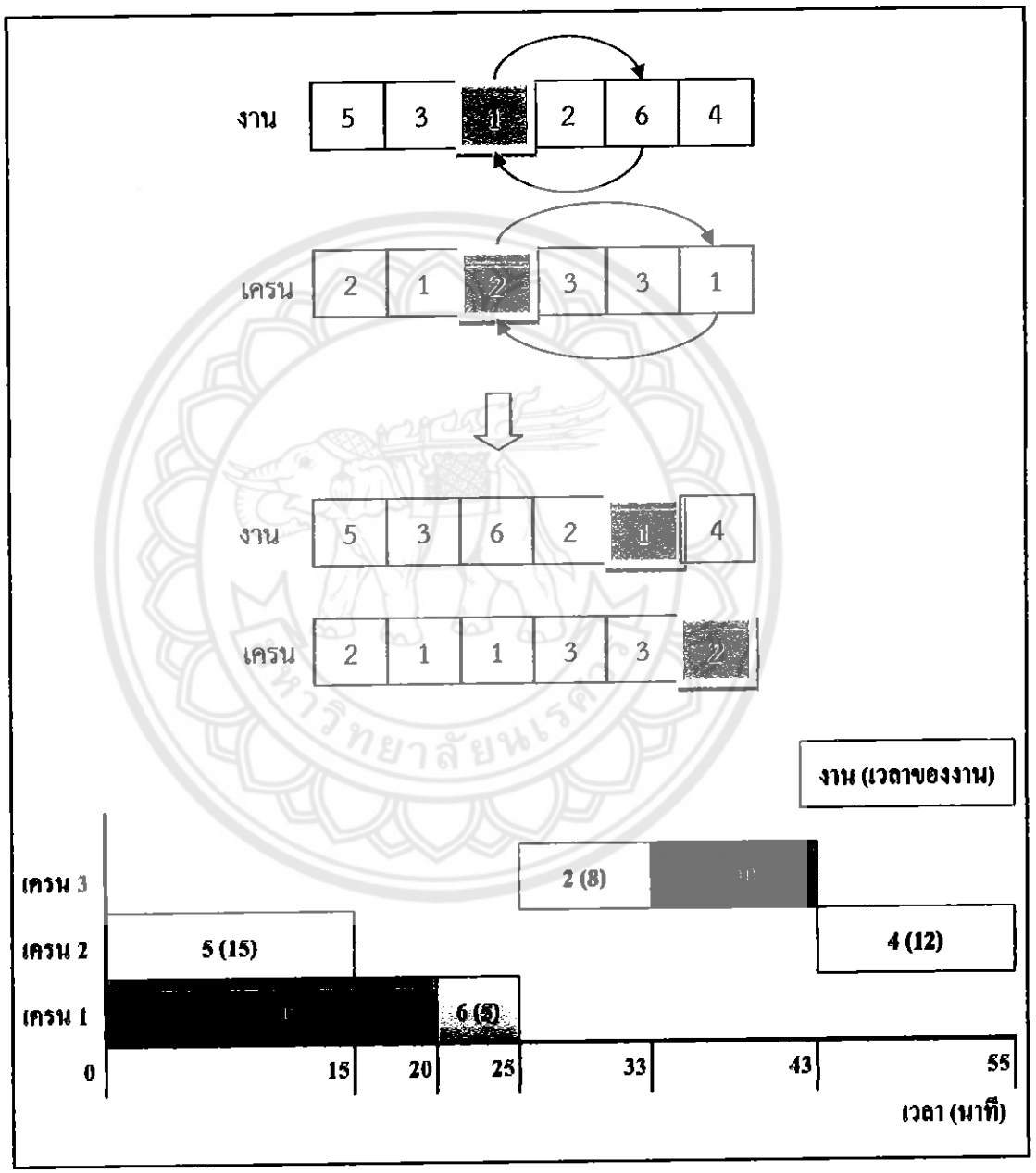
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 2 และสุ่มตำแหน่งงานที่ต้องการสลับไปทางซ้ายมือ ในที่นี้มีงานเดียว คือ ที่งานที่ตำแหน่งแรก นั่นคือ งานที่ 3 ที่ตำแหน่งที่ 2 จะสลับที่กับงานที่ 5 ที่ตำแหน่งแรก ละที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 4 ดังรูป



รูปที่ 3.37 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 ไปทางซ้ายมือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ค.3 เลือกพิจารณาที่ตำแหน่งที่ 3 ไปทางขวามือ

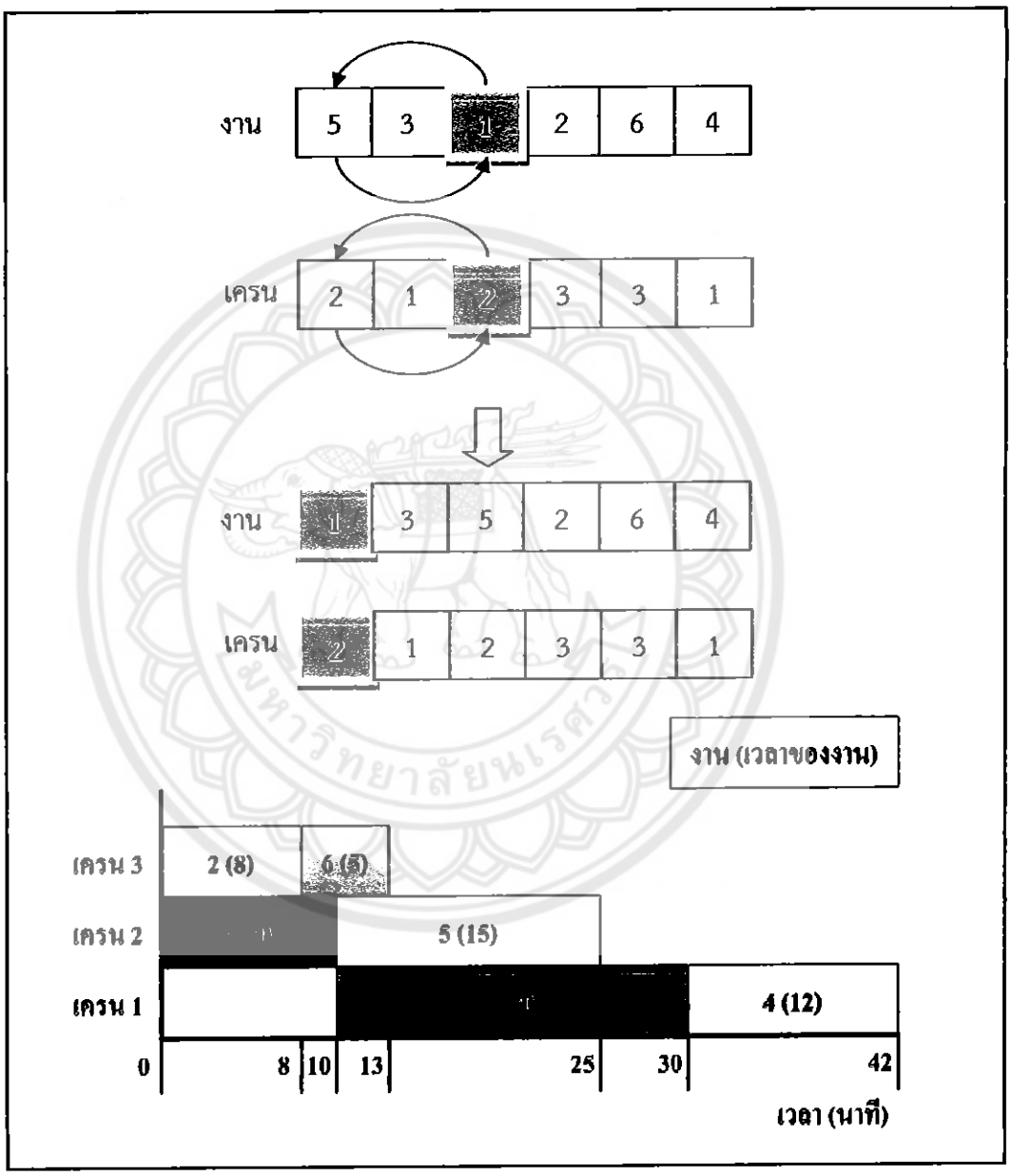
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 3 และสุ่มตำแหน่งงานที่ต้องการสลับที่ คือ ตำแหน่งที่ 5 นั่นคือ งานที่ 1 ที่ตำแหน่งที่ 3 จะย้ายไปสลับที่กับงานที่ 6 ที่ตำแหน่งที่ 5 และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 5 ดังรูป



รูปที่ 3.38 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางขวามือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 55 นาที

ค.4 เลือกพิจารณาที่ตำแหน่งที่ 3 ไปทางซ้ายมือ

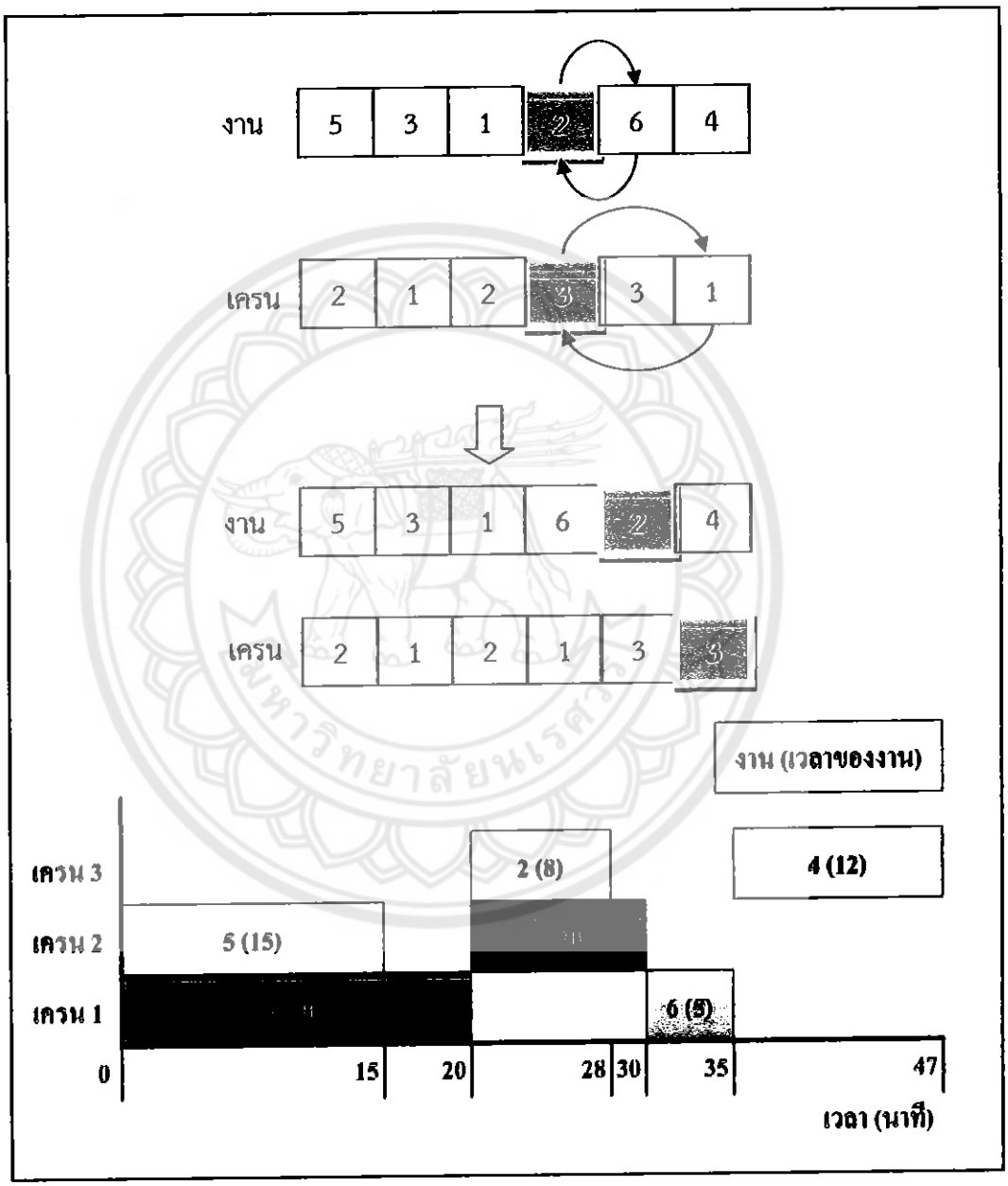
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 3 และสุ่มตำแหน่งงานที่จะทำการสลับที่ไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 1 นั่นคือ งานที่ 1 ที่ตำแหน่งที่ 3 จะย้ายไปสลับที่กับงานที่ 5 ที่ตำแหน่งที่ 1 และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 6 ดังรูป



รูปที่ 3.39 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 ไปทางซ้ายมือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ค.5 เลือกพิจารณาที่ตำแหน่งที่ 4 ไปทางขวามือ

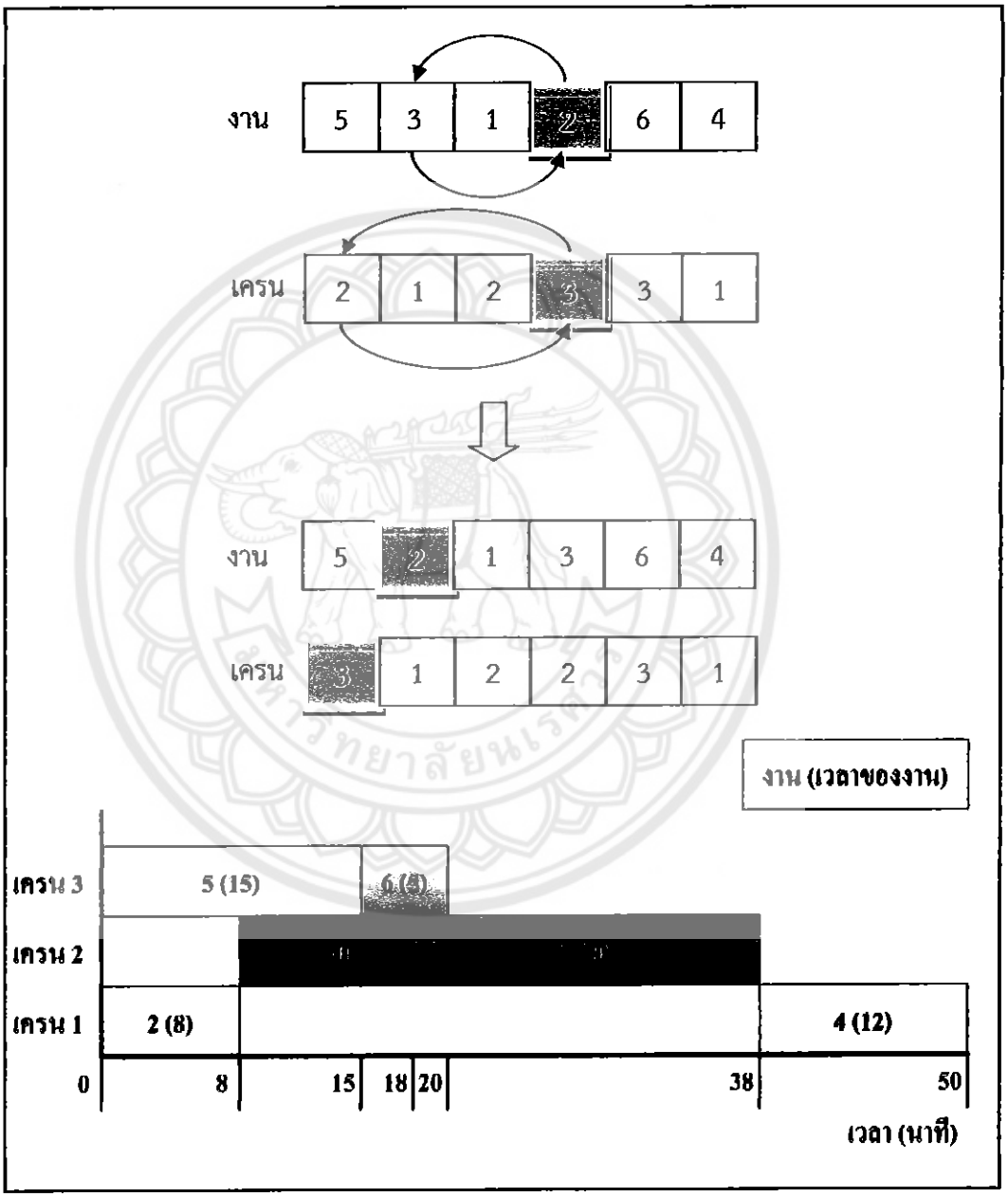
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 4 และสุ่มตำแหน่งงานที่ต้องการสลับที่ คือ ตำแหน่งที่ 5 นั่นคือ งานที่ 2 ที่ตำแหน่งที่ 4 จะย้ายไปสลับที่กับงานที่ 6 ที่ตำแหน่งที่ 5 และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 7 ดังรูป



รูปที่ 3.40 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางขวามือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 47 นาที

ค.6 เลือกพิจารณาที่ตำแหน่งที่ 4 ไปทางซ้ายมือ

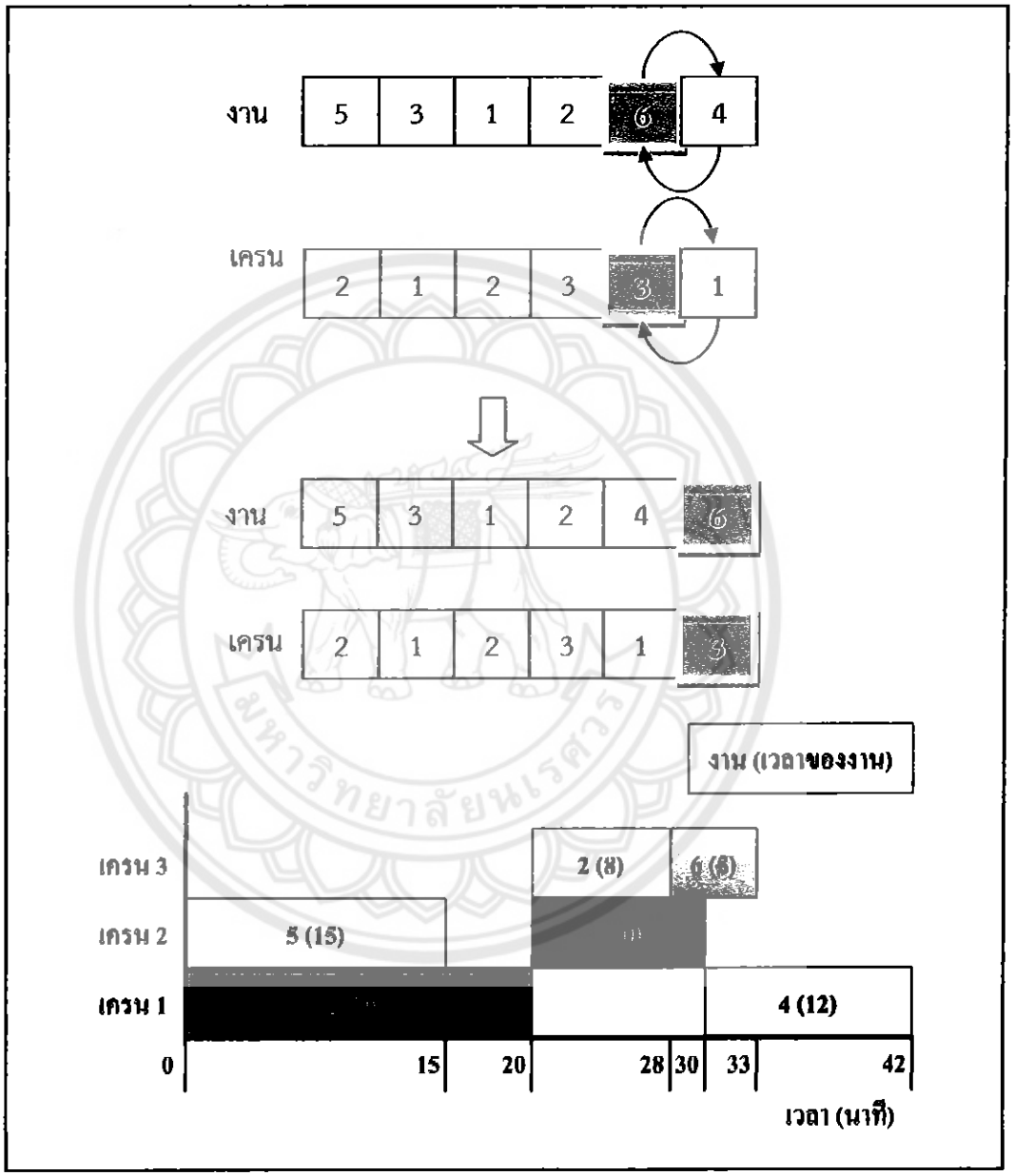
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 4 และสุ่มตำแหน่งงานที่ทำการสลับที่ไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 2 นั่นคือ งานที่ 3 ที่ตำแหน่งที่ 4 จะย้ายไปสลับที่กับงานที่ 3 ที่ตำแหน่งที่ 2 และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 8 ได้ ดังรูป



รูปที่ 3.41 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 ไปทางซ้ายมือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 50 นาที

ค.7 เลือกพิจารณาที่ตำแหน่งที่ 5 ไปทางขวามือ

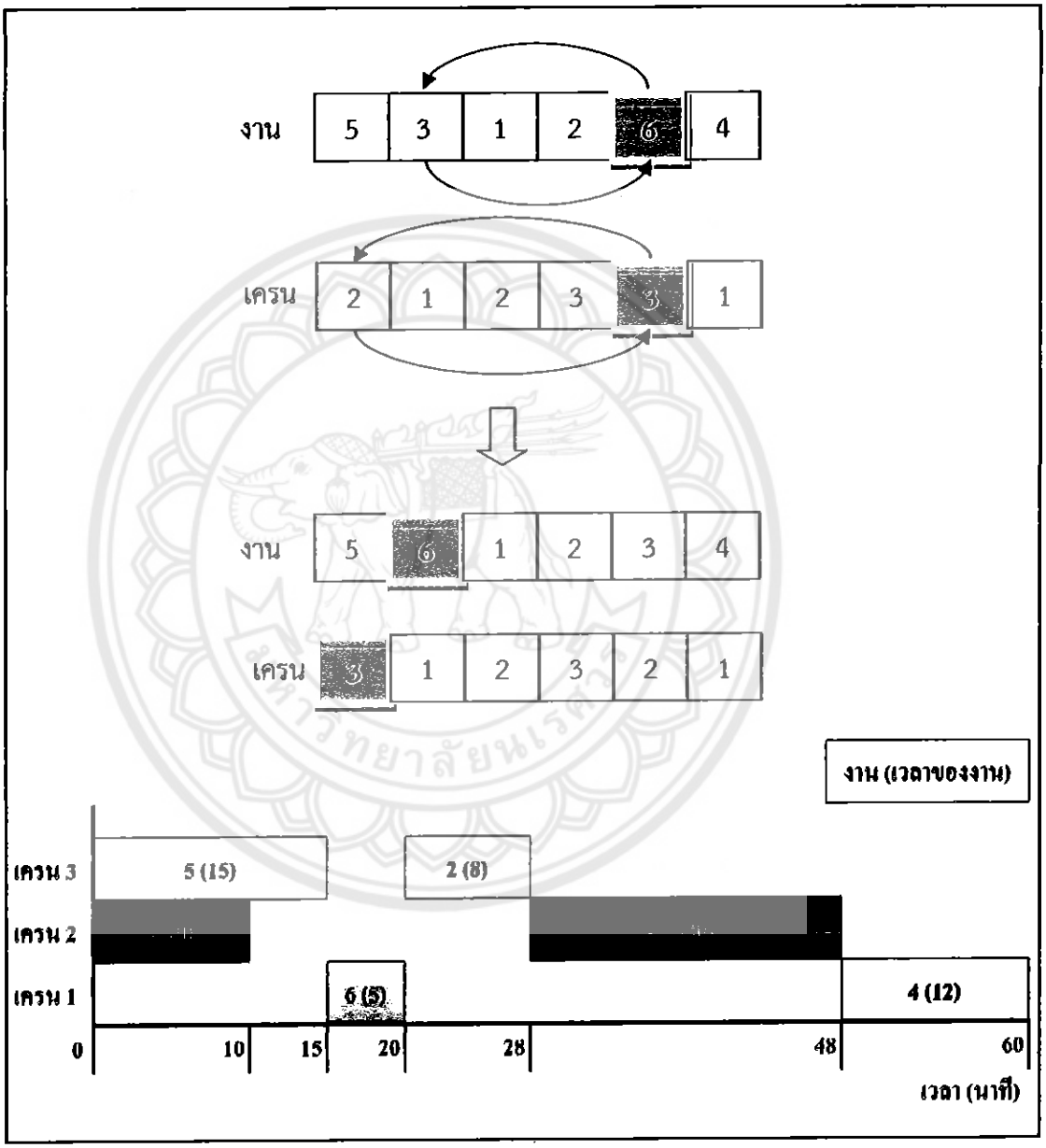
สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 5 และสุ่มตำแหน่งงานที่ทำการสลับที่ ในที่นี้มีตำแหน่งเดียว คือ ตำแหน่งที่ 6 นั่นคือ งานที่ 6 ที่ตำแหน่งที่ 5 จะสลับที่กับงานที่ 4 ตำแหน่งที่ 6และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 9 ได้ ดังรูป



รูปที่ 3.42 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางขวามือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

ค.8 เลือกพิจารณาที่ตำแหน่งที่ 5 ไปทางซ้ายมือ

สามารถอธิบายได้คือ พิจารณางานที่ตำแหน่งที่ 5 และสุ่มตำแหน่งงานที่จะทำการสลับที่ไปทางซ้ายมือ สุ่มได้ตำแหน่งงานที่ 2 นั่นคือ งานที่ 6 ที่ตำแหน่งที่ 5 จะย้ายไปสลับที่กับงานที่ 2 ที่ตำแหน่งที่ 2 และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบที่ 10 ได้ดังรูป



รูปที่ 3.43 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 5 ไปทางซ้ายมือ ของหลักการสลับที่ โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 60 นาที

3.3.2.4 วิธีหาคำตอบใหม่โดยใช้หลักการพิจารณาที่ 2 ตำแหน่ง

ในการหาคำตอบใหม่โดยหลักการพิจารณาที่ 2 ตำแหน่งนั้น จะเป็นการพิจาณาที่ 2 ตำแหน่งใดๆ ที่อยู่ติดกัน และทำการสุ่มตำแหน่งต่างๆ ในงานหรือเครนนั้นๆ ทั้ง 2 ตำแหน่งเช่นกัน แล้วให้นำตำแหน่งที่พิจารณาไปแทนที่ตำแหน่งสุ่ม ทั้งซ้ายมือและขวามือของตำแหน่งที่สุ่ม หลังจากนั้นจึงทำการเลื่อนตำแหน่งไปลำดับถัดไป ดังเช่นตัวอย่างเดิม มีงานจำนวน 6 งาน และเครนจำนวน 3 เครน จะสามารถหาคำตอบใหม่ได้ ดังต่อไปนี้

งาน	5	3	1	2	6	4
เครน	2	1	2	3	3	1

รูปที่ 3.44 ตัวอย่างการหาคำตอบใหม่โดยหลักการพิจารณาที่ 2 ตำแหน่ง

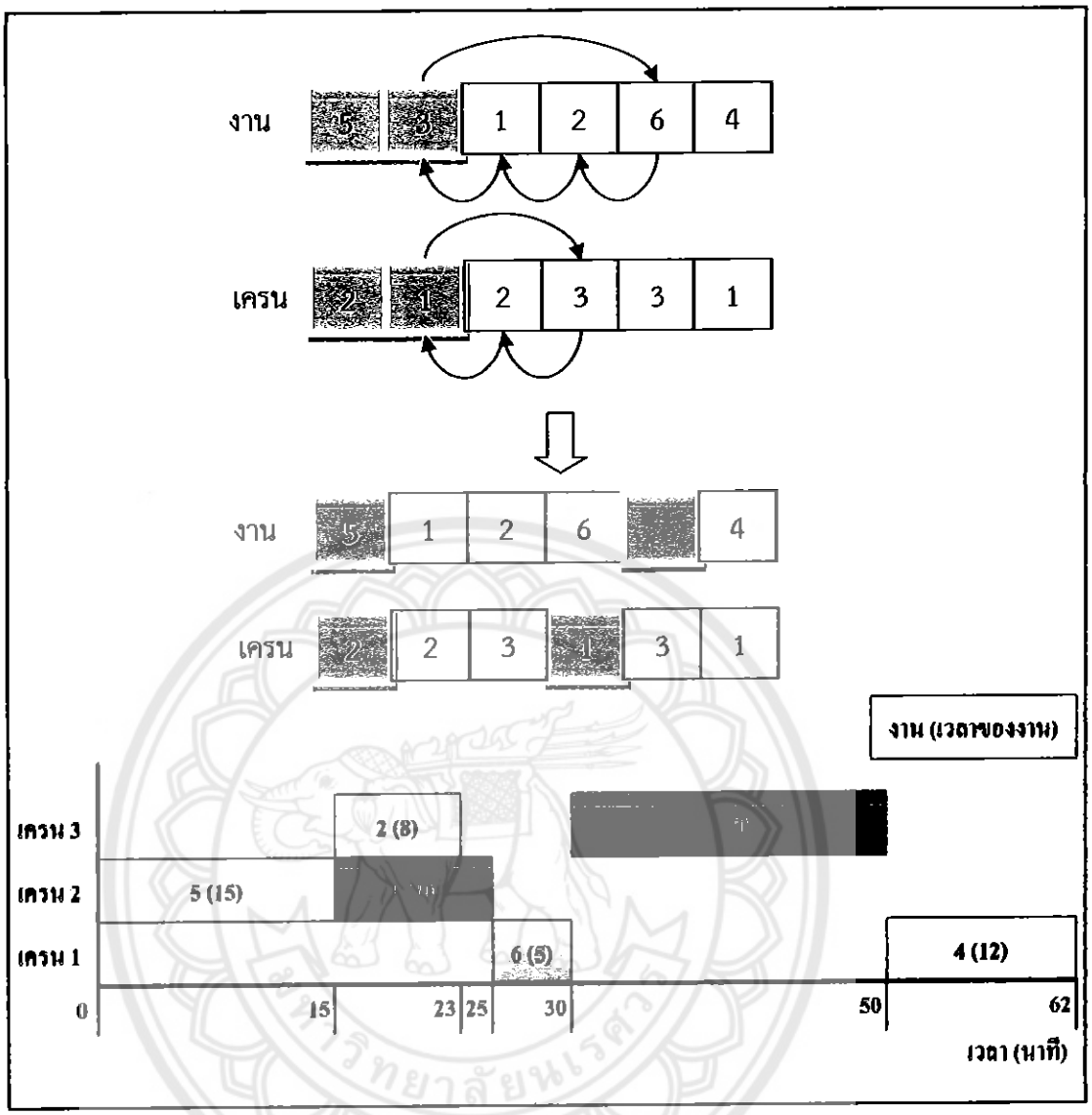
เราสามารถหาค่า X_{new} ได้ทั้งหมด $n-1$ คำตอบ โดยที่

n = จำนวนงาน

ดังนั้น จากตัวอย่างนี้ เราสามารถหา X_{new} ได้ทั้งหมด 5 คำตอบ ซึ่งสามารถพิจารณาตำแหน่งการสุ่มได้เป็น 3 ส่วน ดังต่อไปนี้

ก. พิจารณาที่ตำแหน่งแรกและตำแหน่งที่สอง

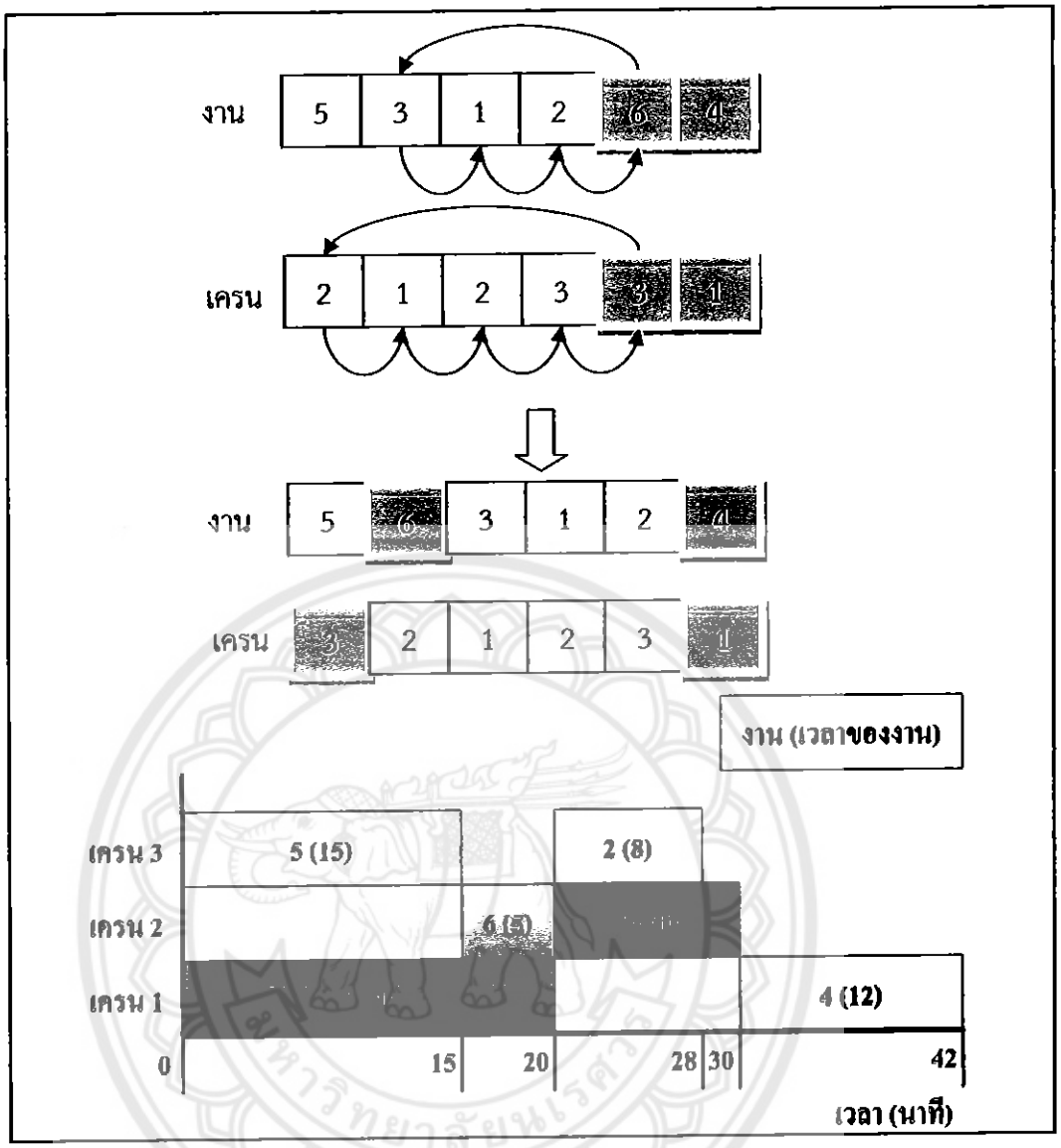
ในการพิจารณาที่ตำแหน่งแรกและตำแหน่งที่ 2 นั้น จากตัวอย่าง ตำแหน่งแรกจะไม่มีตำแหน่งว่างที่สามารถสุ่มได้ทางซ้ายมือ และในตำแหน่งที่ 2 มีตำแหน่งที่สามารถสุ่มได้ 4 ตำแหน่ง คือ ตำแหน่งที่ 3, 4, 5 และ 6 โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป โดยผลของการสุ่มจากตัวอย่างนี้คือ ที่ตำแหน่งของงานที่ตำแหน่งที่ 2 ที่พิจารณาทางขวามือ สุ่มได้ตำแหน่งที่ 5 คืองานที่ 6 จะทำย้ายงาน 3 จากตำแหน่งที่ 2 ไปแทนที่งาน 6 ที่ตำแหน่งที่ 5 และเลื่อนงานในตำแหน่งที่ 3, 4 และ 5 ไปทางซ้ายมือทีละตำแหน่ง และที่ตำแหน่งของเครนที่ตำแหน่งที่ 2 ที่พิจารณาทางขวามือ สุ่มได้ตำแหน่งที่ 4 คือเครนที่ 3 ก็ย้ายตำแหน่งลักษณะเช่นเดียวกันกับตำแหน่งของงาน เช่นเดียวกัน ดังรูป และจะได้คำตอบใหม่ คำตอบที่ 1 จากการทำ Neighborhoods Search คือ



รูปที่ 3.45 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งแรกและตำแหน่งที่สอง ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 62 นาที

ข. พิจารณาที่ตำแหน่งสุดท้ายและตำแหน่งก่อนสุดท้าย

ในการพิจารณาที่ตำแหน่งสุดท้ายและตำแหน่งก่อนสุดท้ายนั้น จากตัวอย่างเดิม ตำแหน่งสุดท้ายจะไม่สามารถไปแทนที่ที่ตำแหน่งใดทางขวามือได้ ส่วนตำแหน่งก่อนสุดท้ายที่พิจารณานั้น จะสามารถไปแทนที่ตำแหน่งต่างๆไปทางซ้ายมือได้ 4 ตำแหน่ง โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป เช่นเดียวกับที่พิจารณาที่ตำแหน่งแรกและตำแหน่งที่ 2 จากตัวอย่างนี้ ในการพิจารณาตำแหน่งก่อนสุดท้าย สุ่มได้ตำแหน่งที่ 2 คือ งานที่ 3 ทางซ้ายมือของตำแหน่งที่พิจารณา จะทำการย้ายงานที่ 4 ที่ตำแหน่งก่อนสุดท้าย นั่นคือ ตำแหน่งที่ 5 ไปแทนที่ตำแหน่งที่สุ่มได้ทางซ้ายมือคือ งานที่ 3 ตำแหน่งที่ 2 และเลื่อนงานตำแหน่งที่ 2, 3, 4 ไปทางขวามือทีละ 1 ตำแหน่ง และที่ตำแหน่งของเครนเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 2 ดังรูป



รูปที่ 3.46 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งสุดท้ายและตำแหน่งก่อนสุดท้าย ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 42 นาที

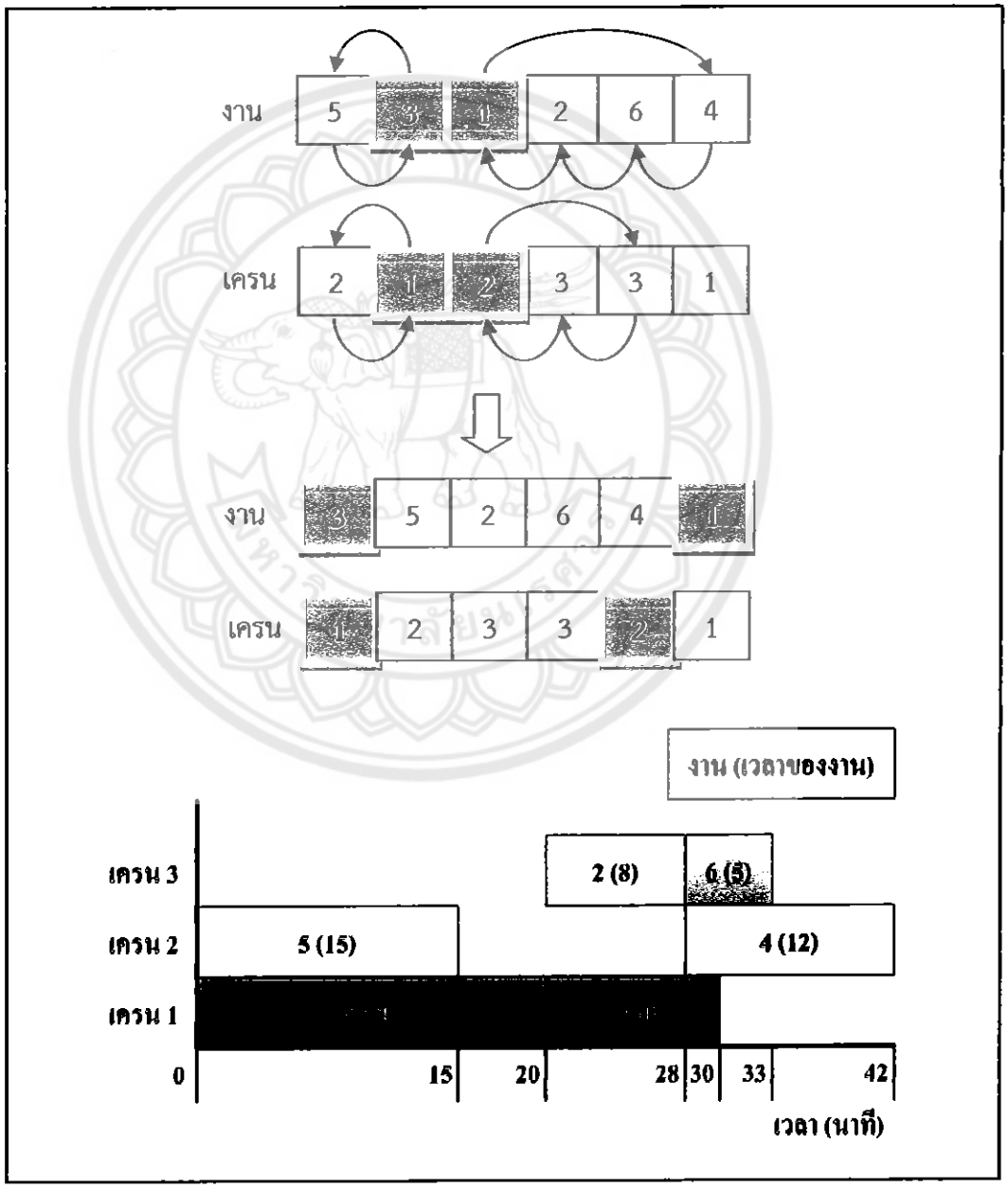
ค. พิจารณาที่ตำแหน่งกลาง 2 ตำแหน่ง

ในการพิจารณาในตำแหน่งกลาง 2 ตำแหน่งนั้น จะสามารถพิจารณาได้จาก ตำแหน่ง 2 ตำแหน่ง ทั้งหมดที่อยู่ระหว่างตำแหน่งแรกกับตำแหน่งสุดท้าย ซึ่งในแต่ละตำแหน่งนั้นเราสามารถหาคำตอบได้ 2 คำตอบ โดยการเลื่อนงานและเครนในตำแหน่งที่เราพิจารณาตำแหน่งที่ 1 ไปทางซ้ายมือ และเลื่อนงานและเครนในตำแหน่งที่เราพิจารณาตำแหน่งที่ 2 ไปทางขวามือ โดยพิจารณาทั้งตำแหน่งของงาน และตำแหน่งของเครน โดยไม่จำเป็นว่าตำแหน่งของงานและเครนที่สุ่มได้นั้นจะต้องตรงกันเสมอไป

จากตัวอย่างเดิม ตำแหน่งกลางที่อยู่ระหว่างตำแหน่งแรกและตำแหน่งสุดท้าย มี 4 ตำแหน่ง คือ ตำแหน่งที่ 2, 3, 4 และ 5 ซึ่งในแต่ละ 2 ตำแหน่งใดๆ สามารถหาคำตอบได้ 2 คำตอบคือ การเลื่อนงานและเครนไปทางซ้ายและขวามือ ดังนั้นจะสามารถพิจารณาในแต่ละตำแหน่ง คือ

ค.1 เลือกพิจารณาที่ตำแหน่งที่ 2 และ 3

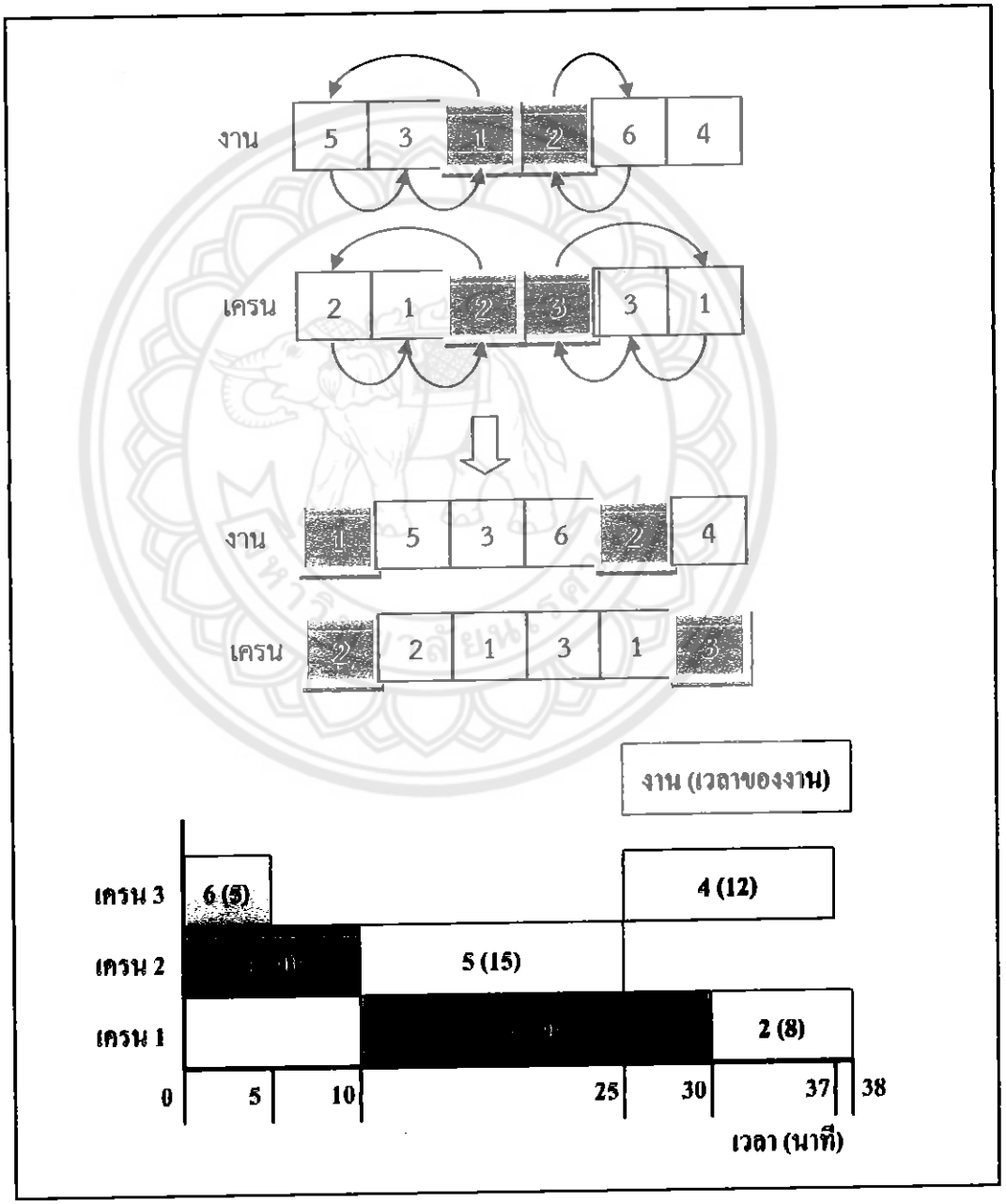
สามารถอธิบายได้คือ ลำดับแรกพิจารณางานที่ตำแหน่งที่ 2 แล้วทำการสุ่มตำแหน่งทางซ้ายมือของตำแหน่งที่ 2 ในที่นี้มีเพียงตำแหน่งเดียว คือตำแหน่งแรก ทำให้งานที่ 3 ที่ตำแหน่งที่ 2 จะไปแทนที่งานที่ 5 ตำแหน่งที่ 1 และตำแหน่งที่ 1 จะถูกเลื่อนไปทางขวามือทีละตำแหน่ง ลำดับที่ 2 พิจารณางานที่ตำแหน่งที่ 3 ทำการสุ่มตำแหน่งทางขวามือของตำแหน่งที่ 3 ซึ่งมี 3 ตำแหน่งคือ 4, 5, 6 ถ้าสุ่มได้ตำแหน่งที่ 6 ทำให้งานที่ 1 ที่ตำแหน่งที่ 3 จะไปแทนที่งานที่ 4 ตำแหน่งที่ 6 และตำแหน่งที่ 4, 5, 6 จะถูกเลื่อนไปทางซ้ายมือทีละตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 3 ดังรูป



รูปที่ 3.47 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 2 และ 3 ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 40 นาที

ค.2 เลือกพิจารณาที่ตำแหน่งที่ 3 และ 4

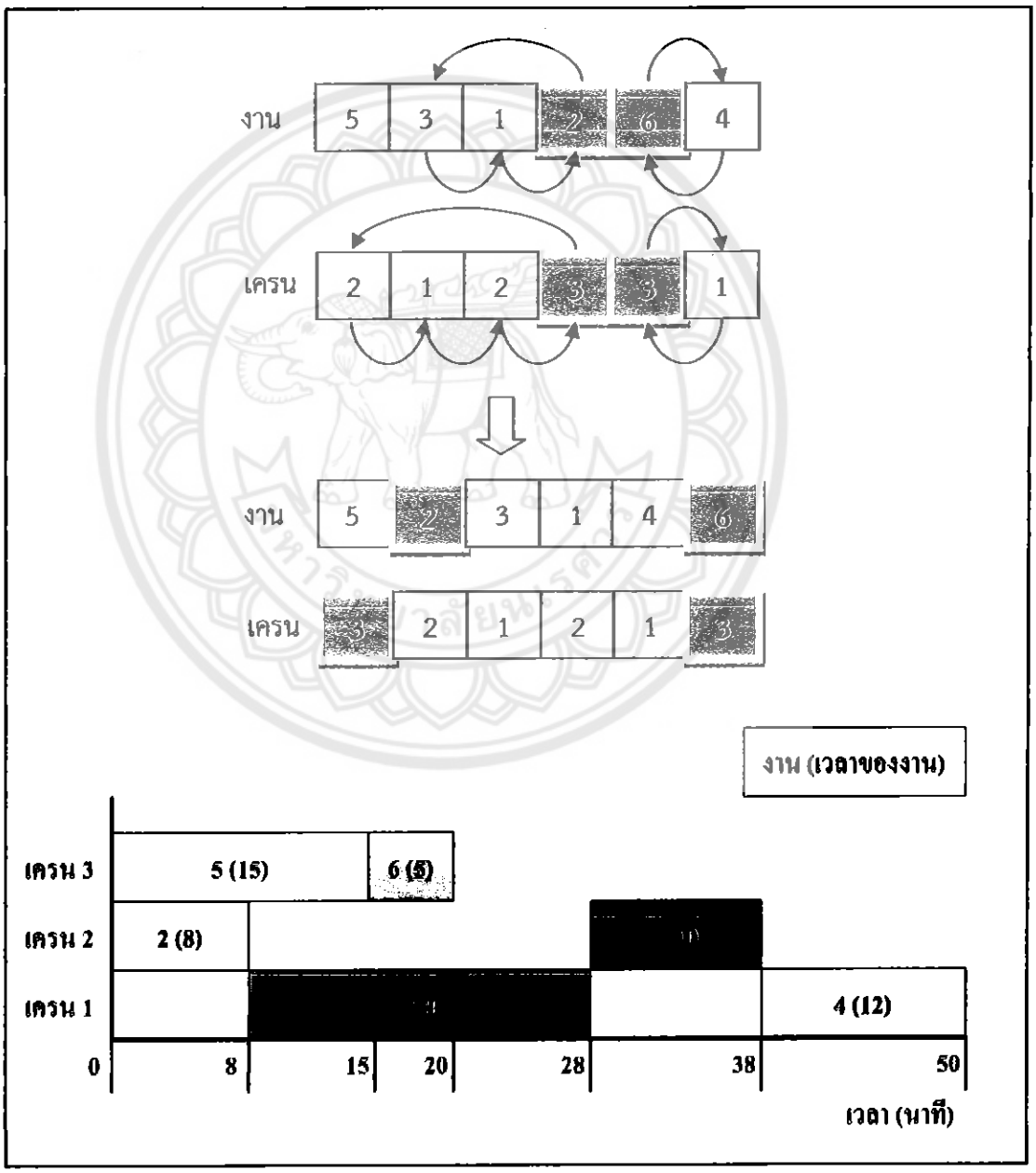
สามารถอธิบายได้คือ ลำดับแรกพิจารณางานที่ตำแหน่งที่ 3 แล้วทำการสุ่มตำแหน่งทางซ้ายมือของตำแหน่งที่ 3 ซึ่งมี 2 ตำแหน่งคือ 1, 2 ถ้าสุ่มได้ตำแหน่งที่ 1 ทำให้งานที่ 1 ที่ตำแหน่งที่ 3 จะไปแทนที่งานที่ 5 ตำแหน่งที่ 1 และตำแหน่งที่ 1, 2 จะถูกเลื่อนไปทางขวามือทีละตำแหน่ง ลำดับที่ 2 พิจารณางานที่ตำแหน่งที่ 4 ทำการสุ่มตำแหน่งทางขวามือของตำแหน่งที่ 4 ซึ่งมี 2 ตำแหน่งคือ 5, 6 ถ้าสุ่มได้ตำแหน่งที่ 5 ทำให้งานที่ 2 ที่ตำแหน่งที่ 4 จะไปแทนที่งานที่ 6 ตำแหน่งที่ 5 และตำแหน่งที่ 5 จะถูกเลื่อนไปทางซ้ายมือทีละตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 4 ดังรูป



รูปที่ 3.48 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 3 และ 4 ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 38 นาที

ค.3 เลือกพิจารณาที่ตำแหน่งที่ 4 และ 5

สามารถอธิบายได้คือ ลำดับแรกพิจารณางานที่ตำแหน่งที่ 4 แล้วทำการสุ่มตำแหน่งทางซ้ายมือของตำแหน่งที่ 4 ซึ่งมี 3 ตำแหน่งคือ 1, 2, 3 ถ้าสุ่มได้ตำแหน่งที่ 2 ทำให้งานที่ 2 ที่ตำแหน่งที่ 4 จะไปแทนที่งานที่ 3 ตำแหน่งที่ 2 และตำแหน่งที่ 2, 3 จะถูกเลื่อนไปทางขวามือทีละตำแหน่ง ลำดับที่ 2 พิจารณางานที่ตำแหน่งที่ 5 ทำการสุ่มตำแหน่งทางขวามือของตำแหน่งที่ 4 ซึ่งในที่นี้มีตำแหน่งเดียวคือ ตำแหน่งที่ 6 ทำให้งานที่ 6 ที่ตำแหน่งที่ 5 จะไปแทนที่งานที่ 4 ตำแหน่งที่ 6 และตำแหน่งที่ 6 จะถูกเลื่อนไปทางซ้ายมือทีละตำแหน่ง และที่ตำแหน่งของเครนก็ทำเช่นเดียวกัน และสามารถหาคำตอบใหม่คือคำตอบที่ 5 ดังรูป



รูปที่ 3.49 แผนภาพจัดเรียงลำดับการทำงานของเครนของคำตอบใหม่ที่ได้จากการพิจารณาตำแหน่งที่ 4 และ 5 ของหลักการพิจารณาที่ 2 ตำแหน่ง โดยมีเวลาที่เครนทำงานเสร็จสิ้นคือ 50 นาที

3.3.3 วิธีการและขั้นตอนการอบอ่อนจำลอง

หลังจากที่ได้ทำการศึกษาปัญหาการจัดลำดับการทำงานของเครน และได้ทำการออกแบบวิธีการสุ่มค่าเริ่มต้นและหลักการหาค่าคำตอบใหม่ในกระบวนการอบอ่อนจำลองแล้ว ต่อไปจะเป็นการออกแบบวิธีการอบอ่อนจำลอง ซึ่งมีขั้นตอนและกระบวนการดังต่อไปนี้

3.3.3.1 กำหนดค่าและตัวแปรที่ใช้ในการอบอ่อนจำลอง อันได้แก่

ก. NumOfJob คือ จำนวนงาน

ข. NumOfCrane คือ จำนวนเครน

ค. Tmax คือ อุณหภูมิเริ่มต้น

ง. Tmin คือ อุณหภูมิสุดท้าย

จ. Eq คือ จำนวนรอบในการอบอ่อน

ฉ. Trate คือ อัตราการเย็นตัว

ช. NS คือ หลักการหรือวิธีการคำตอบใหม่

3.3.3.2 กำหนดให้ค่าอุณหภูมิปัจจุบัน (T) เท่ากับอุณหภูมิเริ่มต้น (Tmax)

3.3.3.3 สุ่มค่าเริ่มต้น และให้ค่าเริ่มต้นเป็นคำตอบปัจจุบัน (Scur) และคำตอบที่ดีที่สุด (Sbest)

3.3.3.4 หาค่าคำตอบใหม่ (Snew) โดยวิธีการทำ Neighbouthood search

3.3.3.5 ทำการตัดสินใจว่า ค่าคำตอบใหม่ที่ได้ ดีกว่าคำตอบปัจจุบันหรือไม่

ก. ถ้าคำตอบใหม่ดีกว่าคำตอบปัจจุบัน (Snew <= Scur) ให้ คำตอบปัจจุบันเท่ากับคำตอบใหม่ (Scur = Snew) และให้ตัดสินใจต่อว่า คำตอบปัจจุบันดีกว่าคำตอบที่ดีที่สุดหรือไม่ (Scur <= Sbest) ซึ่งถ้าคำตอบปัจจุบันดีกว่าคำตอบที่ดีที่สุด จะให้คำตอบที่ดีที่สุดเท่ากับคำตอบปัจจุบันทันที (Sbest = Scur)

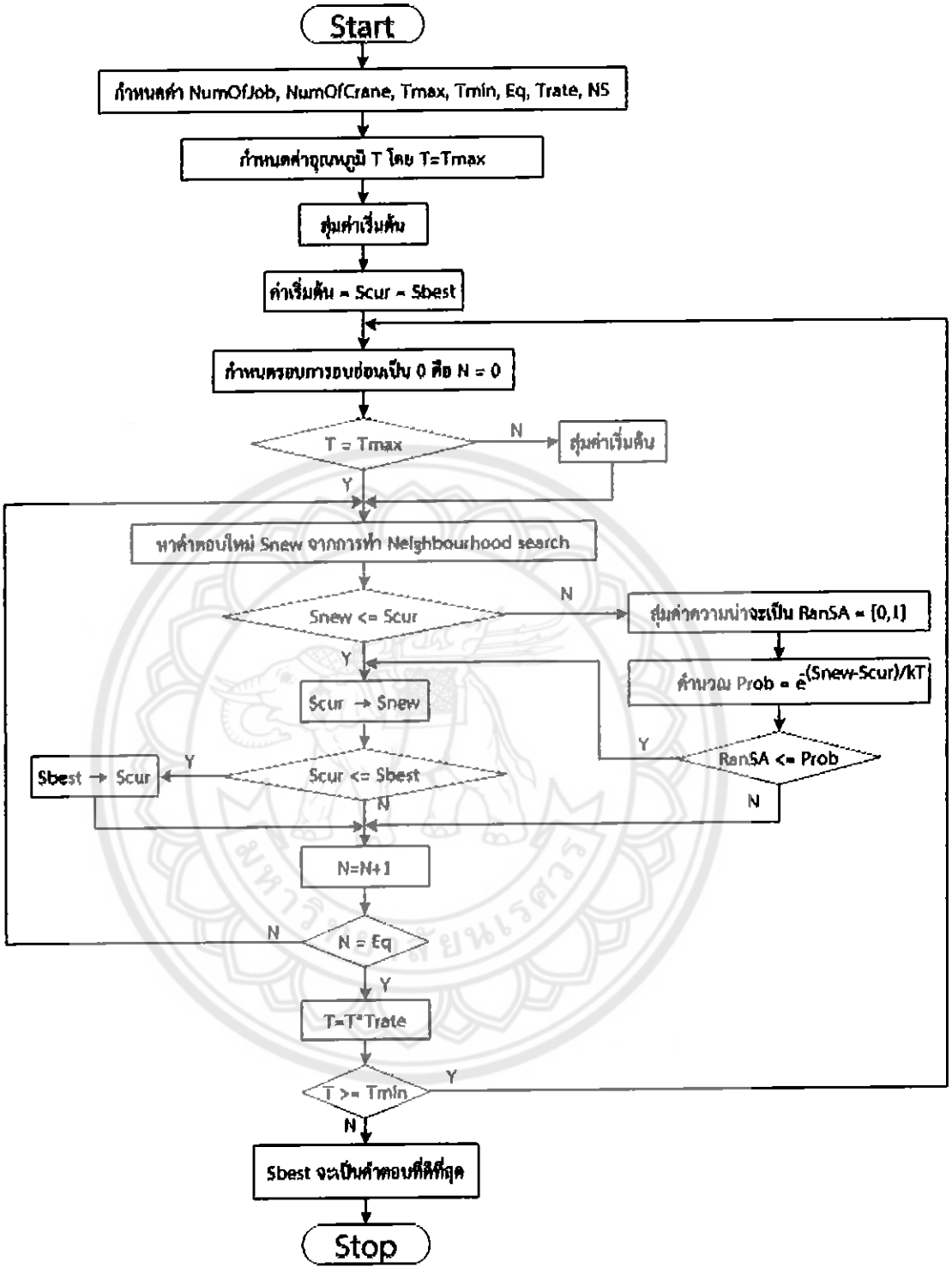
ข. ถ้าคำตอบใหม่ด้อยกว่าคำตอบปัจจุบัน (Snew > Scur) ให้ทำการสุ่มค่าตั้งแต่ 0 ถึง 1 ซึ่งแทนด้วยตัวแปร RanSA และทำการคำนวณความน่าจะเป็น (Prob) ด้วยสมการหนึ่ง ซึ่งก็คือ $Prob = e^{-(Snew-Scur)/kT}$ ซึ่งถ้าค่าที่ทำการสุ่มมามีค่าดีกว่าความน่าจะเป็นที่ได้ (RanSA <= Prob) จะทำให้เกิดการยอมรับค่าคำตอบใหม่ให้เป็นคำตอบปัจจุบัน (Scur = Snew)

3.3.3.6 ให้ทำตามข้อที่ 3.3.3.4 และ 3.3.3.5 ใหม่ จนกระทั่งครบตามจำนวนรอบในการอบอ่อน (Eq)

3.3.3.7 ทำการลดอุณหภูมิ T ลง ด้วยสมการ T = T*Trate และกลับไปเริ่มทำข้อที่ 3.3.3.4 ใหม่ และทำซ้ำไปเรื่อยๆ จนกระทั่งค่าอุณหภูมิ T เข้าใกล้หรือเท่ากับอุณหภูมิสุดท้ายมากที่สุด แต่ต้องไม่น้อยกว่าอุณหภูมิสุดท้าย

3.3.3.8 คำตอบที่ดีที่สุดที่ได้ (Sbest) จะเป็นคำตอบที่ดีที่สุดในการจัดลำดับการทำงานของเครน

ซึ่งวิธีการและขั้นตอนการอบอ่อนจำลอง สามารถแสดงได้ดังแผนภาพในหน้าถัดไป



รูปที่ 3.50 Flow Chart แสดงวิธีการและขั้นตอนการอบอ่อนจำลอง

3.4 การทดสอบโปรแกรมการปัญหาโดยวิธีการอบอ่อนจำลอง

หลังจากที่ได้ทำการศึกษาและออกแบบวิธีการอบอ่อนจำลอง รวมทั้งเขียนโปรแกรมคอมพิวเตอร์ด้วยโปรแกรม Visual Basic for Applications บน Microsoft Excel แล้ว ต่อไปจะเป็นการทดสอบประสิทธิภาพในการหาคำตอบของการจัดลำดับการทำงานของเครนในปัญหาต่างๆบ้าง โดยจะทำการแบ่งปัญหาออกเป็น 3 ขนาด คือ

3.4.1 ปัญหาขนาดเล็ก มีเครนตั้งแต่ 1-3 เครน และมีงานตั้งแต่ 1-5 งาน จำนวน 2 ปัญหา

3.4.2 ปัญหาขนาดกลาง มีเครนตั้งแต่ 1-3 เครน และมีงานตั้งแต่ 1-10 งาน จำนวน 4 ปัญหา

3.4.3 ปัญหาขนาดใหญ่ มีเครนตั้งแต่ 1-6 เครน และมีงานตั้งแต่ 1-15 งาน จำนวน 6 ปัญหา

ซึ่งปัญหาที่นำมาทดสอบประสิทธิภาพการหาคำตอบการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองนั้น จะเป็นปัญหาเดียวกันกับงานวิจัยระดับปริญญาวิศวกรรมศาสตร์บัณฑิต มหาวิทยาลัยนเรศวร ของ กัญญารัตน์ คุ้มคลอง และ สายฝน ช่างเหล่า ในเรื่อง การจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง ปีการศึกษา 2550 รวมทั้งงานวิจัยระดับปริญญาวิศวกรรมศาสตร์บัณฑิต มหาวิทยาลัยนเรศวร ของ เอราวิล ถาวร , หนึ่งฤทัย ทัพใหญ่ และกนกพร อารยิกานนท์ ในเรื่อง การแก้ปัญหาการจัดลำดับการทำงานของเครนโดยใช้โปรแกรมเชิงคณิตศาสตร์ ปีการศึกษา 2550 และเป็นปัญหาเดียวกันกับแบบจำลอง ของ Zhu กับ Lim ที่ใช้โปรแกรม LINGO ประมวลผลของปัญหา ทั้งนี้เพื่อเปรียบเทียบความสามารถของการหาคำตอบการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองที่ออกแบบโดยผู้วิจัย ว่ามีประสิทธิภาพ ความเหมือนกันและความแตกต่างกันอย่างไร และวิธีการหาคำตอบใหม่ในกระบวนการอบอ่อนจำลองมีผลต่อคำตอบที่ได้อย่างไร กับวิธีการที่เคยมีผู้วิจัยมาแล้วข้างต้น

นอกจากนี้แล้ว ผู้วิจัยยังได้ทำการกำหนดปัญหาการจัดลำดับการทำงานของเครนเอง อีก 2 ปัญหา เพื่อทำการทดสอบประสิทธิภาพกับงานวิจัยระดับปริญญาวิศวกรรมศาสตร์บัณฑิต มหาวิทยาลัยนเรศวร ของ กัญญารัตน์ คุ้มคลอง และ สายฝน ช่างเหล่า ในเรื่อง การจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง ปีการศึกษา 2550 โดยใช้เครื่องทดสอบประสิทธิภาพเดียวกันอีกด้วย

ทั้งนี้ผู้วิจัยยังได้ทำการทดลองอีกด้วยว่า ตัวแปรหรือพารามิเตอร์ที่เปลี่ยนแปลงไป ส่งผลอย่างไรต่อคำตอบของปัญหาในการจัดลำดับการทำงานของเครน เช่น อุณหภูมิเริ่มต้น อุณหภูมิสุดท้าย จำนวนรอบของการอบอ่อน และอัตราการเย็นตัว

3.5 จัดทำรายงานและสรุปผล

สรุปผลลัพธ์ที่ได้จากการทดสอบกับปัญหาขนาดต่างๆ แล้วจัดทำรูปเล่มรายงาน

3.6 การนำเสนอโครงงาน

นำผลลัพธ์ที่ได้ทั้งหมดจากการทำโครงงานมานำเสนอต่อคณะกรรมการในการสอบโครงงานวิศวกรรม

บทที่ 4
ผลการทดลองและการวิเคราะห์

ในบทนี้จะกล่าวถึงผลการออกแบบการเขียนโปรแกรมคอมพิวเตอร์ด้วยโปรแกรม Visual Basic for Applications บน Microsoft Excel และผลการทดสอบโปรแกรมการแก้ปัญหาด้วยวิธีการอบอ่อน รวมทั้งทดสอบประสิทธิภาพของโปรแกรมที่ทำการออกแบบ นอกจากนี้ยังเป็นการเปรียบเทียบและวิเคราะห์ผลที่ได้กับงานวิจัยอื่นที่กล่าวมาในบทที่ 3

ซึ่งจะสามารถแบ่งหัวข้อของผลการทดลองและการวิเคราะห์ได้ ดังต่อไปนี้

4.1 โปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

4.2 ผลการทดสอบโปรแกรมการแก้ปัญหาด้วยวิธีการอบอ่อน

4.3 การเปรียบเทียบผลลัพธ์ที่ได้จากการทดสอบโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

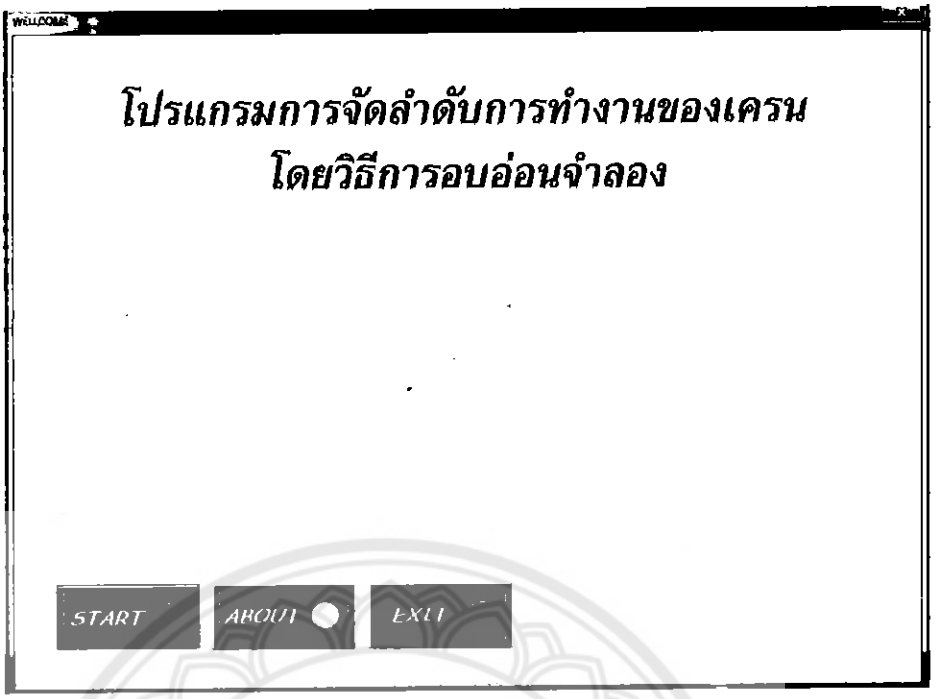
4.4 การวิเคราะห์ผลการทดลอง

4.1 โปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

ในส่วนของโปรแกรมการจัดลำดับการทำงานของเครนนั้น ผู้วิจัยได้ทำการออกแบบให้โปรแกรมมีการทำงานใน 4 ส่วน คือ ส่วนต้อนรับเข้าสู่โปรแกรม ส่วนรับข้อมูล ส่วนยืนยันและแก้ไขข้อมูล และส่วนประมวลผล ซึ่งสามารถแบ่งได้เป็น 6 ขั้นตอน ในกระบวนการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง ซึ่งในแต่ละส่วนของโปรแกรมนั้นสามารถทำงานได้ทั้งในหน้าต่างโปรแกรมและหน้าต่างของ Microsoft Excel ซึ่งสามารถดูรายละเอียดทั้งหมดได้ในภาคผนวก แต่ในที่นี้จะขอกล่าวคร่าวๆ ดังต่อไปนี้

4.1.1 ส่วนต้อนรับเข้าสู่โปรแกรม

ในส่วนต้อนรับเข้าสู่โปรแกรม จะเป็นส่วนที่เมื่อเข้าโปรแกรมมาแล้ว หน้าต่างนี้จะแสดงมาก่อน ดังรูปที่ 4.1



รูปที่ 4.1 แสดงส่วนต้อนรับเข้าสู่โปรแกรมของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

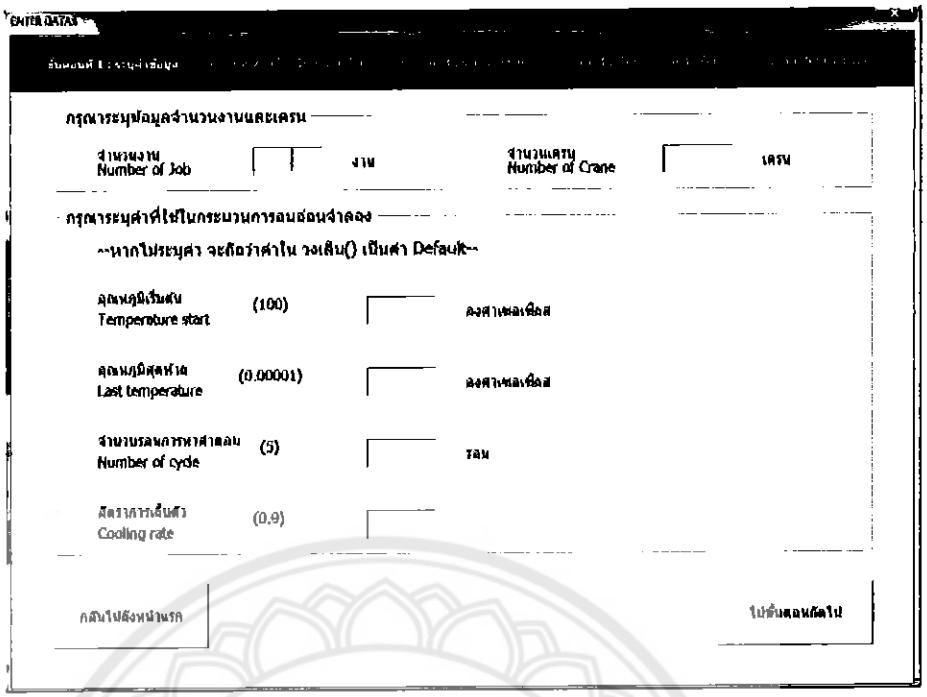
เมื่อกด START จะเริ่มเข้าสู่กระบวนการถัดไป คือ เข้าสู่ส่วนรับข้อมูล แต่เมื่อกด EXIT จะเป็นการปิดหน้าต่างโปรแกรม ซึ่งในส่วนนี้ผู้ใช้อาจเข้าไปสู่หน้าต่างของ Microsoft Excel หรือจะปิดโปรแกรมได้เลย และเมื่อกด ABOUT จะเป็นการให้ความช่วยเหลือ อธิบายโปรแกรม และกล่าวถึงผู้จัดทำโปรแกรม

4.1.2 ส่วนรับข้อมูล

ในส่วนรับข้อมูลนี้ จะเป็นการเริ่มต้นเข้าสู่การจัดลำดับการทำงานของเครน ซึ่งในส่วนรับข้อมูลนี้จะมี 3 ขั้นตอน คือ ขั้นตอนที่ 1 ระบุค่าข้อมูล ขั้นตอนที่ 2 เลือกวิธีการอบอ่อน และขั้นตอนที่ 3 ระบุเวลางาน ดังจะกล่าวดังต่อไปนี้

4.1.2.1 ขั้นตอนที่ 1 ระบุค่าข้อมูล

จะเป็นการให้ผู้ใช้ระบุข้อมูลทั้งหมด อันได้แก่ จำนวนงาน จำนวนเครน อุณหภูมิเริ่มต้นของการอบอ่อน อุณหภูมิสุดท้ายของการอบอ่อน จำนวนรอบในการอบอ่อนแต่ละอุณหภูมิ และอัตราการเย็นตัว ซึ่งผู้ใช้จำเป็นต้องระบุค่าของจำนวนงาน และจำนวนเครน นอกจากนี้แล้วอาจใช้ค่า Default ที่ระบุไว้ในวงเล็บแทนได้ ซึ่งไม่ต้องระบุค่า ดังรูป 4.2

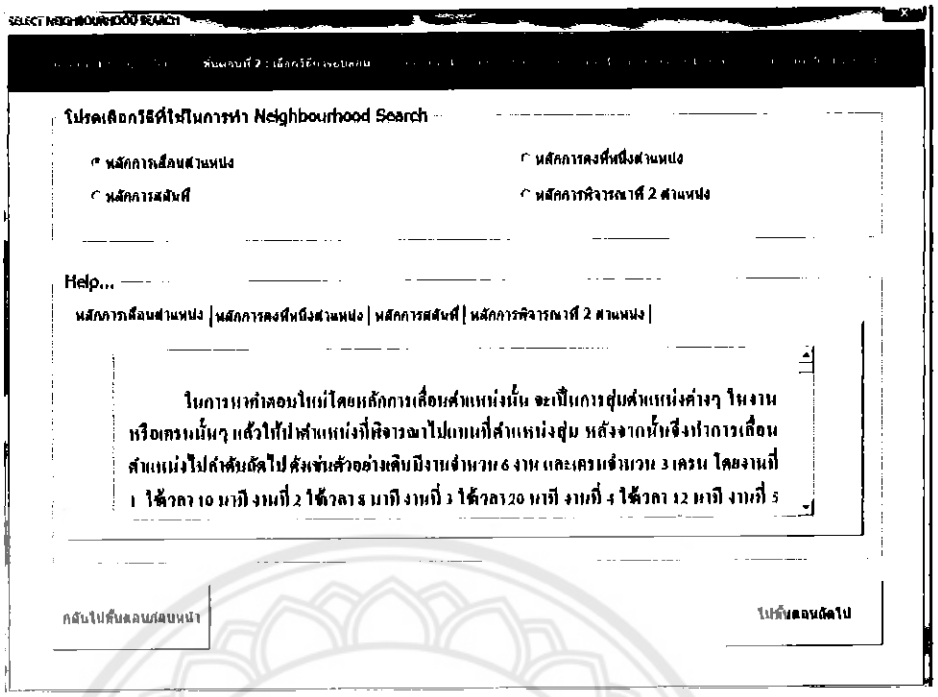


รูปที่ 4.2 แสดงขั้นตอนที่ 1 ระบุค่าข้อมูล ของโปรแกรมจัดลำดับการทำงาน**ของเครนโดย** วิธีการอบอ่อนจำลอง

เมื่อกดไปขั้นตอนถัดไป จะเข้าสู่ขั้นตอนที่ 2 เลือกวิธีการ**อบอ่อน แต่หากกด กลับไป**ยังหน้าแรก โปรแกรมจะกลับไปในส่วนต้อนรับเข้าสู่โปรแกรมซึ่งเป็นหน้า**แรกของโปรแกรม**

4.1.2.2 **ขั้นตอนที่** 2 เลือกวิธีการอบอ่อน

จะเป็นการให้ผู้ใช้คลิกเลือกวิธีการทำ Neighbourhood search **ซึ่งมีทั้งหมด** 4 **วิธี อัน**ได้แก่ หลักการเลื่อนตำแหน่ง หลักการคงที่หนึ่งตำแหน่ง หลักการ**สลับที่ และหลักการ พิจารณา**ที่ 2 ตำแหน่ง ซึ่งเมื่อคลิกเลือกหลักการใด จะมีหน้าต่างช่วยเหลือปรา**กฏ ซึ่งจะบอกวิธีการ และตัว**อย่างในการใช้หลักการนั้นๆ ดังรูปที่ 4.3

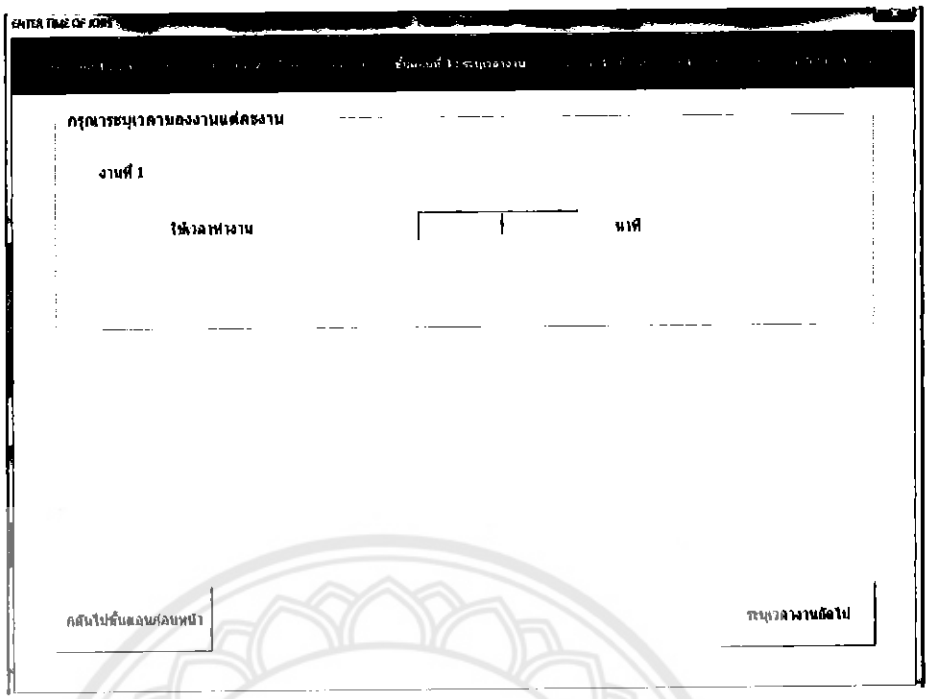


รูปที่ 4.3 แสดงขั้นตอนที่ 2 เลือกวิธีการอบอ่อน ของโปรแกรมจัดลำดับการทำงานของเครน โดยวิธีการอบอ่อนจำลอง

เมื่อกดไปขั้นตอนถัดไป จะเข้าสู่ขั้นตอนที่ 3 ระบุเวลางาน แต่หากกดกลับไปขั้นตอนก่อนหน้า โปรแกรมจะกลับไปในขั้นตอนที่ 1 ระบุค่าข้อมูล

4.1.2.3 ขั้นตอนที่ 3 ระบุเวลางาน

จะเป็นการให้ผู้ใช้ระบุค่าเวลาที่ใช้ในการทำงานแต่ละงาน ซึ่งใน 1 หน้าต่าง จะสามารถระบุเวลางานได้เพียง 1 งาน และจะปรากฏหน้าต่างนี้เท่าจำนวนของงานที่ระบุไว้ในขั้นตอนที่ 1 กล่าวคือ กระบวนการระบุเวลางานจะวนไปเรื่อยๆ จนกระทั่งครบตามจำนวนงานที่ระบุไว้



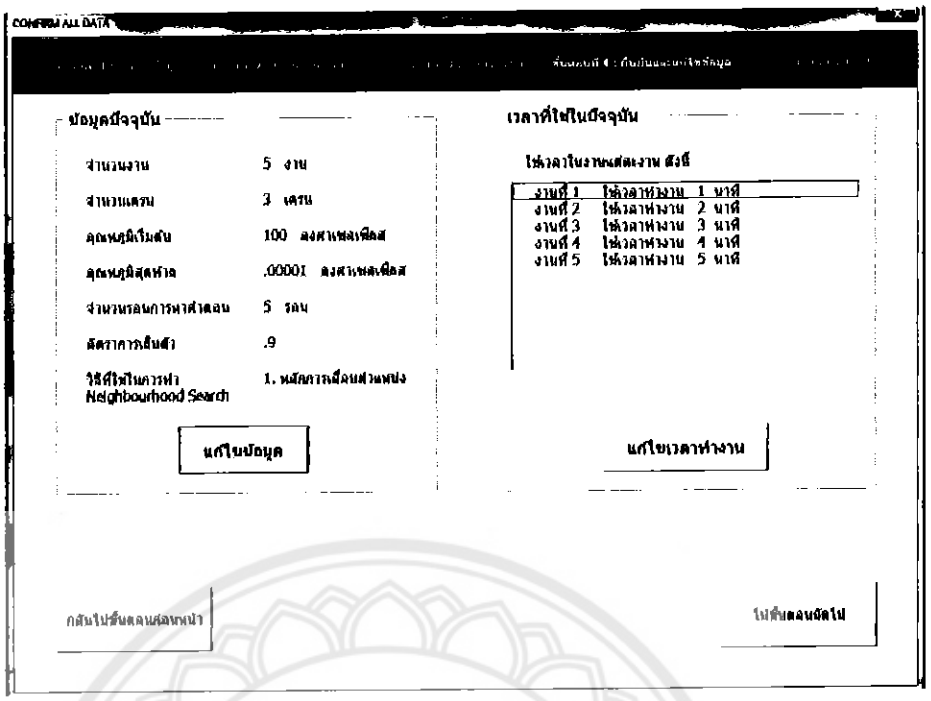
รูปที่ 4.4 แสดงขั้นตอนที่ 3 ระบุเวลางาน ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

หากระบุเวลางานไม่ครบตามจำนวนงาน เมื่อกดระบุเวลางานถัดไป จะเป็นการเริ่มให้ระบุเวลาของงานถัดไป แต่เมื่อกดกลับไปขั้นตอนก่อนหน้า (จะปรากฏเฉพาะในงานที่ 1 เท่านั้น) จะเป็นการกลับไปยังขั้นตอนที่ 2 เลือกวิธีการอบอ่อน เมื่อกดกลับไปเวลางานก่อนหน้า (จะปรากฏในทุกงานที่ไม่ใช่งานที่ 1) จะเป็นการกลับไปยังการระบุเวลางานก่อนหน้านี้

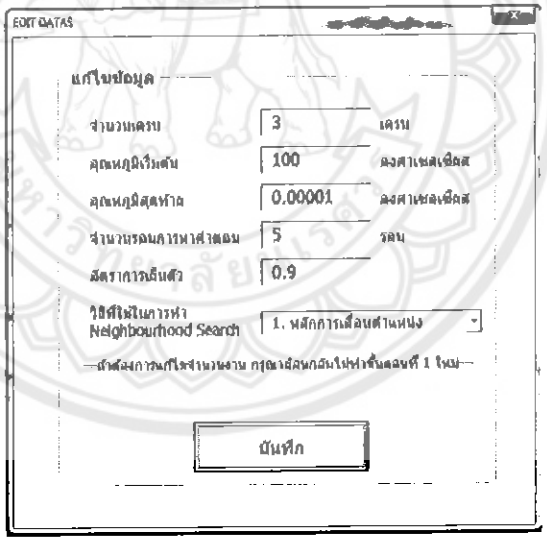
หากระบุเวลางานครบแล้วตามจำนวนงาน เมื่อกดไปขั้นตอนถัดไป จะเป็นการเข้าสู่ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ซึ่งอยู่ในส่วนยืนยันและแก้ไขข้อมูล

4.1.3 ส่วนยืนยันและแก้ไขข้อมูล

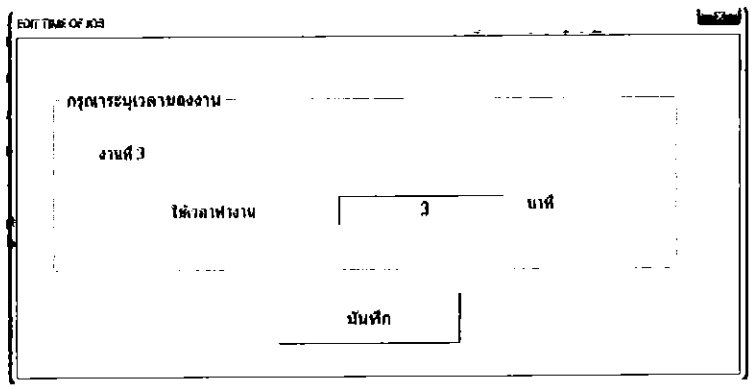
ในส่วนนี้จะเป็นขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ซึ่งจะแสดงให้เห็นถึงทุกข้อมูลที่ได้ระบุมาในขั้นตอนที่ 1-3 ซึ่งสามารถคลิกเพื่อแก้ไขข้อมูลได้ ดังรูป



รูปที่ 4.5 แสดงขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง



รูปที่ 4.6 แสดงการแก้ไขข้อมูลใน ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

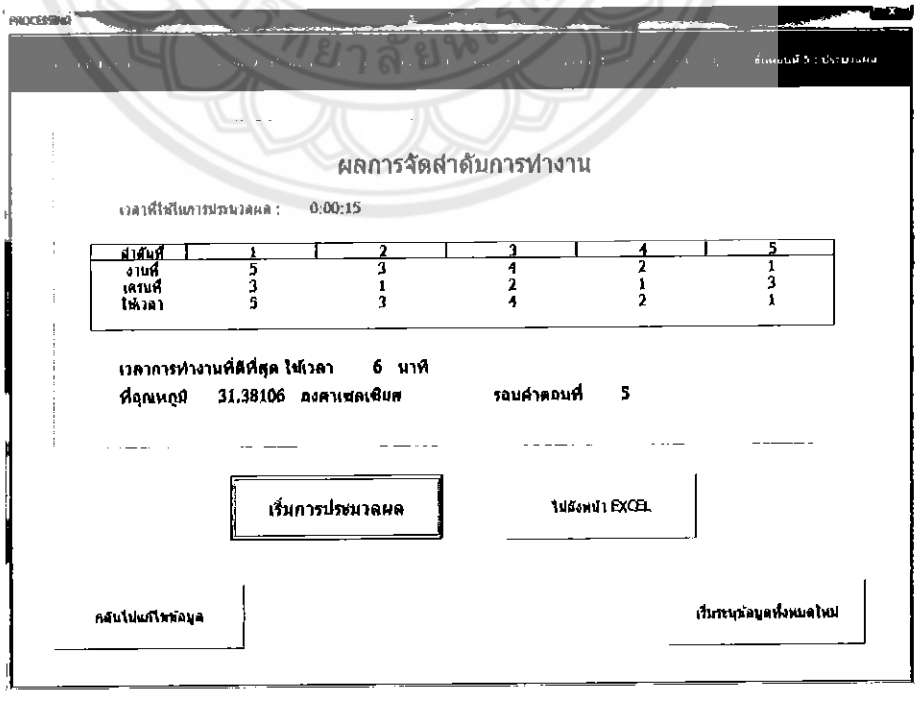


รูปที่ 4.7 แสดงการแก้ไขเวลางานใน ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

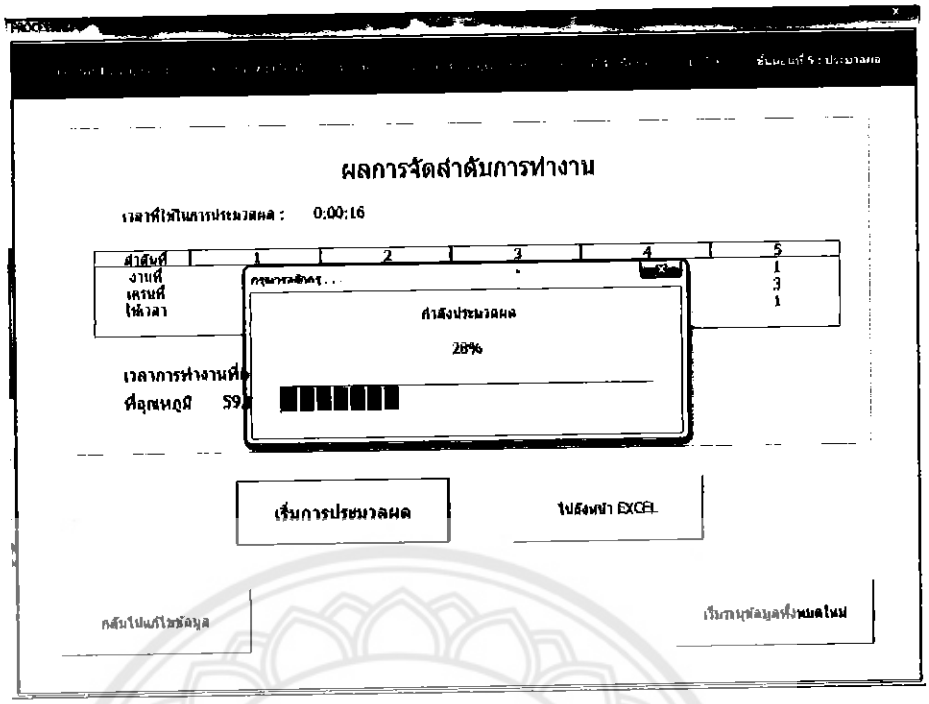
เมื่อตรวจสอบดูแล้วว่าข้อมูลถูกต้อง แล้วจะทำการยืนยันให้ข้อมูล เมื่อกดไปขั้นตอนถัดไปจะเป็นการเข้าสู่ขั้นตอนที่ 5 ประมวลผล หากกดกลับไปขั้นตอนก่อนหน้า โปรแกรมจะกลับไปขั้นตอนที่ 3 ระบุเวลางาน ในงานที่ 1 ใหม่

4.1.4 ส่วนประมวลผล

ในส่วนนี้จะเป็นขั้นตอนที่ 5 ประมวลผล ซึ่งจะเป็นการให้ผู้ใช้สั่งประมวลผลการจัดลำดับการทำงานของเครน และแสดงผลให้ผู้ใช้ทราบถึงผลการจัดลำดับการทำงานของเครน โดยที่จะแสดงเวลาที่ใช้ในการประมวลผล ลักษณะการจัดลำดับการทำงานของเครนที่ดีที่สุด เวลาการทำงานที่น้อยที่สุด อุณหภูมิและรอบคำตอบขณะที่จัดลำดับได้ ซึ่งในการประมวลผลนั้น จะมีแถบความก้าวหน้าบ่งบอกว่า ขณะนั้นโปรแกรมได้ทำการประมวลผลไปสักเท่าไรแล้วเป็นจำนวนเปอร์เซ็นต์



รูปที่ 4.8 แสดงขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง



รูปที่ 4.9 แสดงแถบความก้าวหน้าเมื่อกดประมวลผลใน ขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

ในหน้าต่างโปรแกรมในส่วนนี้ จะมีปุ่มให้เลือกกด 4 ปุ่ม เมื่อกดเริ่มการประมวลผล จะเป็นการสั่งให้โปรแกรมเริ่มการประมวลผลการจัดลำดับการทำงาน เมื่อกดกลับไปหน้า EXCEL จะเป็นทางเลือกให้ผู้ใช้เข้าไปสั่งประมวลผลในหน้าต่างของ Microsoft Excel ซึ่งจะแสดงรายละเอียดการมวลผลการจัดลำดับการทำงานแบบ real time ซึ่งในหน้าต่างของ Microsoft Excel นั้นก็จะมีปุ่มให้เลือกกด 2 ปุ่ม คือ เริ่มการประมวลผลและกลับไปยังโปรแกรม ซึ่งเป็นทางเลือกสำหรับผู้ใช้เช่นเดียวกัน เมื่อกดกลับไปแก้ไขข้อมูล จะเป็นการให้ผู้ใช้สามารถแก้ไขข้อมูลต่างๆได้ ยกเว้นจำนวนงาน ซึ่งหากผู้ใช้ต้องการแก้ไขจำนวนงานจะต้องกดเริ่มระบุข้อมูลทั้งหมดใหม่ ซึ่งโปรแกรมจะย้อนกลับไปยังขั้นตอนที่ 1 ระบุข้อมูล เพื่อให้เริ่มต้นใหม่อีกครั้ง

4.2 ผลการทดสอบโปรแกรมการแก้ปัญหาด้วยวิธีการอบอ่อน

หลังจากได้ทดสอบโปรแกรมแล้ว ต่อไปจะเป็นการแสดงผลลัพธ์ที่ได้จากการทดสอบประสิทธิภาพของโปรแกรมตามลักษณะของปัญหาดังที่ได้กล่าวมาแล้วในบทที่ 3 ซึ่งจะมีในลักษณะของปัญหาขนาดเล็ก ขนาดกลาง และขนาดใหญ่ รวมทั้งปัญหาที่ได้มีการกำหนดขึ้นเอง ทั้งหมด 14 ปัญหา โดยสามารถแบ่งการแสดงผลการทดสอบเป็น 3 ส่วน คือ การแสดงผลลัพธ์ตามลักษณะที่เคยมีผู้วิจัยมาแล้ว การแสดงผลลัพธ์เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อนจำลอง และการแสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเอง

โดยคอมพิวเตอร์ที่ใช้ในการประมวลผลโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองคือ Windows 7 Untimate Pantium(R) Dual-Core CPU Processor E5200, 2.50 GHz, 2.00 GB of RAM (1.75 GB usable), Hard disk 320 GB

ตารางที่ 4.1 แสดงปัญหาแต่ละปัญหาเพื่อการทดสอบประสิทธิภาพของโปรแกรม

ปัญหาที่	จำนวนเครน	จำนวนงาน	เวลาที่ใช้ในการทำงานแต่ละงาน (นาที)
ปัญหาขนาดเล็ก			
1	3	5	2,3,4,5,3
2	3	5	15,17,23,12,10
ปัญหาขนาดกลาง			
3	3	6	5,1,2,8,1,6
4	3	6	13,32,22,17,25,19
5	3	7	36,24,24,41,35,33,15
6	3	8	15,12,27,28,11,22,41,35
ปัญหาขนาดใหญ่			
7	5	6	11,22,10,10,10,6
8	5	7	1,2,4,1,2,1,1
9	5	8	25,25,35,25,25,35,25,25
10	5	9	4,4,4,3,3,7,2,2,7
11	5	10	26,18,26,30,18,55,30,15,30
12	5	12	20,22,25,20,29,21,25,29,25,23,24,27
ปัญหาที่ได้มีการกำหนดขึ้นเอง			
13	10	30	22,17,30,7,28,55,31,14,27,33,22,44,15,28,39, 41, 11,18,15,9,32,35,15,26,35,30,21,22,22,32
14	12	40	22,35,12,30,11,7,28,17,17,23,24,31,15,6,24, 32,18,22,17,36,24,10,3,18,40,31,16,22,27,18, 30,34, 21,14,17,32,9,40,32,14

4.2.1 การแสดงผลลัพธ์ตามลักษณะที่เคยมีผู้วิจัยมาแล้ว

เป็นการแสดงผลลัพธ์ที่ได้จากการทดสอบประสิทธิภาพของโปรแกรมที่ผู้วิจัยเขียนขึ้น โดยจะใช้หลักการในการหา Nieghbourhood search ทั้งหมด 4 หลักการ ซึ่งผลลัพธ์ที่ได้จะมีลักษณะของปัญหาและค่าตัวแปรในการอบอ่อนจำลองเหมือนกับที่เคยมีผู้วิจัยมาแล้ว คือ มี 12 ปัญหา และมีค่าตัวแปรในการอบอ่อน ดังนี้

อุณหภูมิเริ่มต้น	100	องศาเซลเซียส
อุณหภูมิสุดท้าย	0.00001	องศาเซลเซียส
จำนวนรอบในการอบอ่อน	5	รอบ
อัตราการเย็นตัว	0.90	

ซึ่งในการทำการทดลองมีจำนวนครั้งในการหาคำตอบหลักการละ 10 ครั้ง ต่อ 1 ปัญหา ซึ่งผลที่ได้จะมีลักษณะ ดังนี้

ตารางที่ 4.2 แสดงผลลัพธ์ตามลักษณะที่เคยมีผู้วิจัยมาแล้ว

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
ปัญหาขนาดเล็ก : ปัญหาที่ 1						
1.หลักการเลื่อนตำแหน่ง	8	8	0	18.3	18	0.48
2.หลักการคงที่หนึ่งตำแหน่ง	8	8	0	18.4	18	0.51
3.หลักการสลับที่	8	8	0	18.5	18	0.53
4.หลักการพิจารณาที่ 2 ตำแหน่ง	8	8	0	12.8	12	0.42
ปัญหาขนาดเล็ก : ปัญหาที่ 2						
1.หลักการเลื่อนตำแหน่ง	32	32	0	17.9	17	0.32
2.หลักการคงที่หนึ่งตำแหน่ง	32	32	0	18.1	18	0.32
3.หลักการสลับที่	32	32	0	18.3	18	0.48
4.หลักการพิจารณาที่ 2 ตำแหน่ง	32	32	0	12.8	12	0.42
ปัญหาขนาดกลาง : ปัญหาที่ 3						
1.หลักการเลื่อนตำแหน่ง	8	8	0	23.8	23	0.63
2.หลักการคงที่หนึ่งตำแหน่ง	8	8	0	24.2	24	0.42
3.หลักการสลับที่	8	8	0	24.2	23	0.63
4.หลักการพิจารณาที่ 2 ตำแหน่ง	8	8	0	15.9	15	0.32
ปัญหาขนาดกลาง : ปัญหาที่ 4						
1.หลักการเลื่อนตำแหน่ง	45	45	0	24.3	24	0.48
2.หลักการคงที่หนึ่งตำแหน่ง	45	45	0	24.2	24	0.42
3.หลักการสลับที่	45	45	0	23.8	23	0.63
4.หลักการพิจารณาที่ 2 ตำแหน่ง	45	45	0	15.7	12	1.34
ปัญหาขนาดกลาง : ปัญหาที่ 5						
1.หลักการเลื่อนตำแหน่ง	77	77	0	31.1	31	0.32
2.หลักการคงที่หนึ่งตำแหน่ง	77	77	0	30.7	27	1.34
3.หลักการสลับที่	77	77	0	31	31	0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	78.5	77	2.55	20.1	20	0.32
ปัญหาขนาดกลาง : ปัญหาที่ 6						
1.หลักการเลื่อนตำแหน่ง	70	68	2.16	39.2	39	0.42
2.หลักการคงที่หนึ่งตำแหน่ง	70.6	68	2.63	37.9	37	0.57
3.หลักการสลับที่	68.6	68	0.70	37.8	37	0.42
4.หลักการพิจารณาที่ 2 ตำแหน่ง	70.2	68	2.70	22.9	22	0.32

ตารางที่ 4.2 (ต่อ) แสดงผลลัพธ์ตามลักษณะที่เคยมีผู้วิจัยมาแล้ว

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
ปัญหาขนาดใหญ่ : ปัญหาที่ 7						
1.หลักการเลื่อนตำแหน่ง	22	22	0	23.8	23	0.42
2.หลักการคงที่หนึ่งตำแหน่ง	22	22	0	23.7	23	0.48
3.หลักการสลับที่	22	22	0	23.8	23	0.42
4.หลักการพิจารณาที่ 2 ตำแหน่ง	22	22	0	15.6	15	0.53
ปัญหาขนาดใหญ่ : ปัญหาที่ 8						
1.หลักการเลื่อนตำแหน่ง	4	4	0	30.8	30	0.42
2.หลักการคงที่หนึ่งตำแหน่ง	4	4	0	31.3	31	0.48
3.หลักการสลับที่	4	4	0	31	30	0.47
4.หลักการพิจารณาที่ 2 ตำแหน่ง	4	4	0	17.6	17	0.52
ปัญหาขนาดใหญ่ : ปัญหาที่ 9						
1.หลักการเลื่อนตำแหน่ง	56	50	5.16	32.6	32	0.52
2.หลักการคงที่หนึ่งตำแหน่ง	58	50	4.21	32.8	32	0.42
3.หลักการสลับที่	59	50	3.16	32.8	32	0.42
4.หลักการพิจารณาที่ 2 ตำแหน่ง	59	50	3.16	17.2	17	0.42
ปัญหาขนาดใหญ่ : ปัญหาที่ 10						
1.หลักการเลื่อนตำแหน่ง	9	9	0	41.3	41	0.48
2.หลักการคงที่หนึ่งตำแหน่ง	9	9	0	42.1	41	0.57
3.หลักการสลับที่	9	9	0	42.7	42	0.48
4.หลักการพิจารณาที่ 2 ตำแหน่ง	9.5	9	0.71	24	23	0.47
ปัญหาขนาดใหญ่ : ปัญหาที่ 11						
1.หลักการเลื่อนตำแหน่ง	73.4	70	1.43	54	54	0
2.หลักการคงที่หนึ่งตำแหน่ง	73.3	70	1.34	54	53	0.47
3.หลักการสลับที่	73.3	70	1.34	53.4	53	0.52
4.หลักการพิจารณาที่ 2 ตำแหน่ง	74.8	70	2.66	29.1	29	0.32
ปัญหาขนาดใหญ่ : ปัญหาที่ 12						
1.หลักการเลื่อนตำแหน่ง	74.8	72	1.55	83.7	76	5.38
2.หลักการคงที่หนึ่งตำแหน่ง	75.8	75	1.03	87.2	87	0.42
3.หลักการสลับที่	74.2	70	1.62	87.2	87	0.42
4.หลักการพิจารณาที่ 2 ตำแหน่ง	76.9	71	2.64	49.5	49	0.71

จะเห็นได้ว่า จากตารางที่ 4.2 จะเป็นผลลัพธ์ที่ได้จากการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองโดยใช้โปรแกรมที่ผู้วิจัยได้ออกแบบและสร้างขึ้น ซึ่งผลลัพธ์ที่ได้จะแสดงการหาค่า Makespan หรือเวลาที่เครนทำงานเสร็จสิ้นทั้งโดยเฉลี่ยและค่าน้อยที่สุด รวมทั้งเวลาที่ใช้ในการประมวลผลของโปรแกรมทั้งโดยเฉลี่ยและค่าที่น้อยที่สุดเช่นเดียวกัน ซึ่งจะเห็นได้ว่าเมื่อขนาดของปัญหาเพิ่มมากขึ้น จะส่งผลให้เวลาในการประมวลผลเพิ่มขึ้นเช่นเดียวกัน นอกจากนี้ยังแสดงถึงค่า SD หรือส่วนเบี่ยงเบนมาตรฐาน ซึ่งบ่งบอกว่าผลลัพธ์ที่ได้นั้น มีการกระจายตัวอย่างไร

ทั้งนี้การเปรียบเทียบผลลัพธ์ที่ได้นี้จะขอกล่าวต่อไปในหัวข้อการเปรียบเทียบผลลัพธ์ในข้อ 4.3.1 การเปรียบเทียบผลลัพธ์ที่ได้ตามลักษณะที่เคยมีผู้วิจัยมาแล้ว ต่อไป

4.2.2 การแสดงผลลัพธ์เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อนจำลอง

เป็นการแสดงผลลัพธ์ของโปรแกรมกับปัญหาขนาดต่างๆทั้ง 12 ปัญหา ซึ่งผู้วิจัยได้ทำการเปลี่ยนแปลงค่าตัวแปรต่างๆ ในการอบอ่อนทั้งหมด 2 ครั้ง ซึ่งในการเปลี่ยนแปลงค่าตัวแปรนี้ จะมีจำนวนครั้งในการหาคำตอบทั้งหมดเท่ากัน กล่าวคือ ในการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองโดยไม่มีการเปลี่ยนแปลงค่าตัวแปรนั้น จะมีจำนวนครั้งในการหาคำตอบตามหลักการอบอ่อนจำลองจำนวน 765 ครั้ง ซึ่งในการเปลี่ยนแปลงค่าตัวแปรทั้ง 2 ครั้งนั้น จะมีจำนวนครั้งในการหาคำตอบเท่ากันคือ 765 ครั้ง

ซึ่งตัวแปรในการอบอ่อนจำลองเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1 คือ

อุณหภูมิเริ่มต้น	500	องศาเซลเซียส
อุณหภูมิสุดท้าย	0.00001	องศาเซลเซียส
จำนวนรอบในการอบอ่อน	5	รอบ
อัตราการเย็นตัว	0.89	

ซึ่งตัวแปรในการอบอ่อนจำลองเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2 คือ

อุณหภูมิเริ่มต้น	50	องศาเซลเซียส
อุณหภูมิสุดท้าย	0.00001	องศาเซลเซียส
จำนวนรอบในการอบอ่อน	9	รอบ
อัตราการเย็นตัว	0.834	

ซึ่งในการทำการทดลองมีจำนวนครั้งในการหาคำตอบหลักการละ 10 ครั้ง ต่อ 1 ปัญหา และต่อไปจะเป็นการแสดงผลลัพธ์ของโปรแกรมเมื่อมีการเปลี่ยนแปลงตัวแปร ดังต่อไปนี้

ตารางที่ 4.3 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 1

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	8	8	0	18.4	18	0.52
2.หลักการคงที่หนึ่งตำแหน่ง	8	8	0	18.4	18	0.52
3.หลักการสลับที่	8	8	0	18.4	18	0.52
4.หลักการพิจารณาที่ 2 ตำแหน่ง	8	8	0	13.1	13	0.32
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	8	8	0	18	18	0
2.หลักการคงที่หนึ่งตำแหน่ง	8	8	0	17.9	17	0.32
3.หลักการสลับที่	8	8	0	17.8	17	0.42
4.หลักการพิจารณาที่ 2 ตำแหน่ง	8	8	0	12.7	12	0.48

ตารางที่ 4.4 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 2

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	32	32	0	18.5	18	0.53
2.หลักการคงที่หนึ่งตำแหน่ง	32	32	0	18.4	18	0.52
3.หลักการสลับที่	32	32	0	18.5	18	0.53
4.หลักการพิจารณาที่ 2 ตำแหน่ง	32	32	0	13	13	0
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	32	32	0	18	18	0
2.หลักการคงที่หนึ่งตำแหน่ง	32	32	0	18.2	18	0.42
3.หลักการสลับที่	32	32	0	18.1	18	0.35
4.หลักการพิจารณาที่ 2 ตำแหน่ง	32	32	0	12.4	12	0.52

ตารางที่ 4.5 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 3

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	8	8	0	24.8	24	0.42
2.หลักการคงที่หนึ่งตำแหน่ง	8	8	0	24.6	24	0.52
3.หลักการสลับที่	8	8	0	24.6	24	0.52
4.หลักการพิจารณาที่ 2 ตำแหน่ง	8	8	0	16	16	0
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	8	8	0	23.9	23	0.32
2.หลักการคงที่หนึ่งตำแหน่ง	8	8	0	23.9	23	0.32
3.หลักการสลับที่	8	8	0	24	24	0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	8	8	0	15.7	15	0.48

ตารางที่ 4.6 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 4

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	45	45	0	24.8	24	0.42
2.หลักการคงที่หนึ่งตำแหน่ง	45	45	0	25	25	0
3.หลักการสลับที่	45	45	0	25	25	0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	45	45	0	16.2	16	0.42
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	45	45	0	24	24	0
2.หลักการคงที่หนึ่งตำแหน่ง	45	45	0	24.2	24	0.42
3.หลักการสลับที่	45	45	0	23.9	23	0.32
4.หลักการพิจารณาที่ 2 ตำแหน่ง	45	45	0	15.9	15	0.32

ตารางที่ 4.7 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 5

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	77	77	0	32	32	0
2.หลักการคงที่หนึ่งตำแหน่ง	77	77	0	32	32	0
3.หลักการสลับที่	77	77	0	32	32	0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	77.9	77	2.02	20.5	20	0.53
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	77	77	0	31	31	0
2.หลักการคงที่หนึ่งตำแหน่ง	77	77	0	31	31	0
3.หลักการสลับที่	77	77	0	31	30	0.47
4.หลักการพิจารณาที่ 2 ตำแหน่ง	77.3	77	0.95	19.6	19	0.52

ตารางที่ 4.8 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 6

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	68.8	68	1.87	49.8	39	0.63
2.หลักการคงที่หนึ่งตำแหน่ง	69.6	68	2.37	40.2	40	0.42
3.หลักการสลับที่	68.7	68	0.67	39.7	39	0.48
4.หลักการพิจารณาที่ 2 ตำแหน่ง	70.3	68	2.58	24	24	0
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	68.4	68	0.70	39	38	0.47
2.หลักการคงที่หนึ่งตำแหน่ง	68.8	68	0.63	39	38	0.47
3.หลักการสลับที่	68.3	68	0.48	39	38	0.47
4.หลักการพิจารณาที่ 2 ตำแหน่ง	70	68	2.45	24	23.9	0.32

ตารางที่ 4.9 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 7

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	22	22	0	24.3	23	0.82
2.หลักการคงที่หนึ่งตำแหน่ง	22	22	0	24	24	0
3.หลักการสลับที่	22	22	0	24.5	23	0.71
4.หลักการพิจารณาที่ 2 ตำแหน่ง	22	22	0	15.6	15	0.52
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	22	22	0	23.5	23	0.53
2.หลักการคงที่หนึ่งตำแหน่ง	22	22	0	23.7	23	0.48
3.หลักการสลับที่	22	22	0	24.1	24	0.32
4.หลักการพิจารณาที่ 2 ตำแหน่ง	22	22	0	15.2	15	0.42

ตารางที่ 4.10 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 8

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	4	4	0	31.2	30	0.63
2.หลักการคงที่หนึ่งตำแหน่ง	4	4	0	31.3	31	0.95
3.หลักการสลับที่	4	4	0	31.5	30	0.71
4.หลักการพิจารณาที่ 2 ตำแหน่ง	4	4	0	18.1	17	0.57
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	4	4	0	32	32	0
2.หลักการคงที่หนึ่งตำแหน่ง	4	4	0	32	32	0
3.หลักการสลับที่	4	4	0	31.9	31	0.32
4.หลักการพิจารณาที่ 2 ตำแหน่ง	4	4	0	17.9	17	0.32

ตารางที่ 4.11 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 9

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	56	50	5.16	32.7	31	1.16
2.หลักการคงที่หนึ่งตำแหน่ง	58	50	4.22	32.8	31	0.79
3.หลักการสลับที่	58	50	4.22	32.2	31	0.79
4.หลักการพิจารณาที่ 2 ตำแหน่ง	59	50	3.16	17.6	17	0.97
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	56	50	5.16	33.2	31	1.23
2.หลักการคงที่หนึ่งตำแหน่ง	57	50	4.83	33.4	31	1.51
3.หลักการสลับที่	57	50	4.83	33.8	33	0.79
4.หลักการพิจารณาที่ 2 ตำแหน่ง	58	50	4.22	17.4	17	0.52

ตารางที่ 4.12 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 10

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	9	9	0	41.5	40	1.43
2.หลักการคงที่หนึ่งตำแหน่ง	9	9	0	41.9	41	1.45
3.หลักการสลับที่	9	9	0	42.4	40	1.90
4.หลักการพิจารณาที่ 2 ตำแหน่ง	9.2	9	0.42	23.9	23	0.74
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	9	9	0	41.9	40	1.37
2.หลักการคงที่หนึ่งตำแหน่ง	9	9	0	41.8	40	1.81
3.หลักการสลับที่	9	9	0	42.1	41	1.60
4.หลักการพิจารณาที่ 2 ตำแหน่ง	9.1	9	0.32	24.8	23	1.40

ตารางที่ 4.13 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 11

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	72.1	70	2.23	53.8	50	1.75
2.หลักการคงที่หนึ่งตำแหน่ง	72.5	66	0.95	53.8	50	2.57
3.หลักการสลับที่	70.7	66	3.62	55.1	51	2.13
4.หลักการพิจารณาที่ 2 ตำแหน่ง	72.9	66	2.88	30.5	27	2.22
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	71.9	70	1.66	53.7	50	2.95
2.หลักการคงที่หนึ่งตำแหน่ง	71.9	66	2.51	53.7	51	1.77
3.หลักการสลับที่	69.5	66	2.80	54.3	51	2.91
4.หลักการพิจารณาที่ 2 ตำแหน่ง	72.6	70	1.90	29.6	28	0.84

ตารางที่ 4.14 แสดงผลลัพธ์ของปัญหาเมื่อมีการเปลี่ยนแปลงตัวแปรของปัญหาที่ 12

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
เปลี่ยนแปลงตัวแปร ครั้งที่ 1						
1.หลักการเลื่อนตำแหน่ง	74.8	72	1.55	86.3	85	1.34
2.หลักการคงที่หนึ่งตำแหน่ง	76	75	12.5	86.9	85	1.52
3.หลักการสลับที่	75	74	1.05	94	86	4.08
4.หลักการพิจารณาที่ 2 ตำแหน่ง	77.7	74	2.45	51.9	49	2.18
เปลี่ยนแปลงตัวแปร ครั้งที่ 2						
1.หลักการเลื่อนตำแหน่ง	73.7	70	1.64	90.7	89	1.42
2.หลักการคงที่หนึ่งตำแหน่ง	74.4	72	1.58	91	88	1.94
3.หลักการสลับที่	74.4	72	1.07	89.9	83	3.03
4.หลักการพิจารณาที่ 2 ตำแหน่ง	76	75	1.15	54.5	49	3.31

จะเห็นได้ว่าจากตารางที่ 4.3-4.14 เป็นการแสดงผลลัพธ์ของการจัดลำดับการทำงานของเครนในรูปแบบของการหาค่า Makespan หรือเวลาที่เครนทำงานเสร็จสิ้นทั้งโดยเฉลี่ยและค่าน้อยที่สุด รวมทั้งเวลาที่ใช้ในการประมวลผลของโปรแกรมทั้งโดยเฉลี่ยและค่าที่น้อยที่สุดเช่นเดียวกัน เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อนในครั้งที่ 1 และครั้งที่ 2 ซึ่งจะแสดงตามหลักการที่ใช้ในการทำ Neighbourhood search ทั้ง 4 หลักการ ซึ่งจะเห็นได้ว่า การเปลี่ยนแปลงตัวแปร ครั้งที่ 2 จะให้

ค่า Makespan ที่ดีกว่าการเปลี่ยนแปลงตัวแปร ครั้งที่ 1 นอกจากนี้ยังแสดงถึงค่า SD หรือส่วนเบี่ยงเบนมาตรฐาน ซึ่งบ่งบอกว่าผลลัพธ์ที่ได้นั้น มีการกระจายตัวอย่างไร

ทั้งนี้การเปรียบเทียบผลลัพธ์ที่ได้นี้จะขอกล่าวต่อไปในหัวข้อการเปรียบเทียบผลลัพธ์ในข้อ 4.3.2 การเปรียบเทียบผลลัพธ์ที่ได้เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ต่อไป

4.2.3 การแสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเอง

จะเป็นการแสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเองทั้งหมด 2 ปัญหา คือ ปัญหาที่ 13 และปัญหาที่ 14 ซึ่งการแสดงผลลัพธ์นี้จะแสดงทั้งผลลัพธ์ที่ได้จากการประมวลผลโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองที่ผู้วิจัยได้ออกแบบและเขียนโปรแกรมขึ้นเอง และผลลัพธ์ที่ได้จากการประมวลผลโปรแกรมที่เคยมีวิจัยอื่นได้ออกแบบและเขียนโดย กัญญารัตน์ และสายฝน ทั้งนี้ผู้วิจัยได้ทำการประมวลผลของปัญหาที่กำหนดบนคอมพิวเตอร์เครื่องเดียวกัน เพื่อการเทียบเคียง

โดยในการประมวลผลนั้น ได้ทำการประมวลผลทั้งเมื่อไม่มีการเปลี่ยนแปลงตัวแปร มีการเปลี่ยนแปลงตัวแปรครั้งที่ 1 และมีการเปลี่ยนแปลงตัวแปรครั้งที่ 2

ซึ่งในการทำการทดลองมีจำนวนครั้งในการหาคำตอบหลักการละ 10 ครั้ง ต่อ 1 ปัญหา

ตารางที่ 4.15 แสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเองเมื่อไม่มีการเปลี่ยนแปลงตัวแปร

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
ปัญหาที่ได้มีการกำหนดขึ้นเอง : ปัญหาที่ 13						
1.หลักการเลื่อนตำแหน่ง	192	182	5.06	615.8	604	9.45
2.หลักการคงที่หนึ่งตำแหน่ง	197.4	187	7.79	641.5	600	21.28
3.หลักการสลับที่	191.2	175	11.16	647	612	19.87
4.หลักการพิจารณาที่ 2 ตำแหน่ง	203.3	175	14.70	323.2	317	11.15
โปรแกรมของกัญญารัตน์และสายฝน	195.4	172	12.21	399.6	370	14.34
ปัญหาที่ได้มีการกำหนดขึ้นเอง : ปัญหาที่ 14						
1.หลักการเลื่อนตำแหน่ง	219.5	197	9.69	1264.2	1261	3.99
2.หลักการคงที่หนึ่งตำแหน่ง	226.2	215	6.07	1248	1221	15.14
3.หลักการสลับที่	217.2	202	7.81	1241.4	1231	5.06
4.หลักการพิจารณาที่ 2 ตำแหน่ง	233	217	10.14	630.9	628	5.70
โปรแกรมของกัญญารัตน์และสายฝน	227.6	211	10.31	752.5	735	14.25

หมายเหตุ: โปรแกรมของกัญญารัตน์และสายฝนและโปรแกรมของผู้วิจัยจะถูกประมวลผลบนเครื่องคอมพิวเตอร์เครื่องเดียวกันในปัญหาที่ 13 และ 14 โดยผู้วิจัย

ตารางที่ 4.16 แสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเองเมื่อมีการเปลี่ยนแปลงตัวแปรครั้งที่ 1

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
ปัญหาที่ได้มีการกำหนดขึ้นเอง : ปัญหาที่ 13						
1.หลักการเลื่อนตำแหน่ง	189.9	182	6.45	641.1	628	9.56
2.หลักการคงที่หนึ่งตำแหน่ง	197.8	190	8.02	636.8	622	11.67
3.หลักการสลับที่	195.4	181	8.38	633.8	615	13.67
4.หลักการพิจารณาที่ 2 ตำแหน่ง	201.3	181	12.70	329.2	312	8.39
โปรแกรมของกัญญารัตน์และสายฝน	204.4	191	8.22	388.8	376	14.88
ปัญหาที่ได้มีการกำหนดขึ้นเอง : ปัญหาที่ 14						
1.หลักการเลื่อนตำแหน่ง	220.6	212	7.20	1257.8	1253	2.53
2.หลักการคงที่หนึ่งตำแหน่ง	223.6	207	8.26	1256.2	1241	8.29
3.หลักการสลับที่	220.9	212	7.89	1250.7	1221	11.80
4.หลักการพิจารณาที่ 2 ตำแหน่ง	226.8	206	13.20	621.4	610	6.72
โปรแกรมของกัญญารัตน์และสายฝน	236.1	222	9.69	750.5	727	15.86

หมายเหตุ: โปรแกรมของกัญญารัตน์และสายฝนและโปรแกรมของผู้วิจัยจะถูกประมวลผลบนเครื่องคอมพิวเตอร์เครื่องเดียวกันในปัญหาที่ 13 และ 14 โดยผู้วิจัย

ตารางที่ 4.17 แสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเองเมื่อมีการเปลี่ยนแปลงตัวแปรครั้งที่ 2

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
ปัญหาที่ได้มีการกำหนดขึ้นเอง : ปัญหาที่ 13						
1.หลักการเลื่อนตำแหน่ง	179.7	171	4.95	634.3	621	9.76
2.หลักการคงที่หนึ่งตำแหน่ง	180.2	160	11.53	630	619	7.38
3.หลักการสลับที่	172.1	158	8.18	628.4	616	10.12
4.หลักการพิจารณาที่ 2 ตำแหน่ง	190.2	171	10.54	325.6	318	8.02
โปรแกรมของกัญญารัตน์และสายฝน	192.8	191	10.83	388.7	372	15.86

ตารางที่ 4.17 (ต่อ) แสดงผลลัพธ์ของปัญหาที่ได้มีการกำหนดขึ้นเองเมื่อมีการเปลี่ยนแปลงตัวแปรครั้งที่ 2

หลักการพิจารณา	เวลาที่เครนทำงานเสร็จสิ้น (นาที)			เวลาที่ใช้ในการประมวลผล (วินาที)		
	ค่าเฉลี่ย	น้อยที่สุด	SD	ค่าเฉลี่ย	น้อยที่สุด	SD
ปัญหาที่ได้มีการกำหนดขึ้นเอง : ปัญหาที่ 14						
1.หลักการเลื่อนตำแหน่ง	203.8	195	5.75	1254.9	1248	4.53
2.หลักการคงที่หนึ่งตำแหน่ง	211	201	8.07	1241.2	1215	10.63
3.หลักการสลับที่	198.8	175	10.50	1251.1	1241	6.17
4.หลักการพิจารณาที่ 2 ตำแหน่ง	216.1	197	12.06	629.7	612	12.13
โปรแกรมของกัญญารัตน์และสายฝน	225.3	203	14.41	760.1	738	16.62

หมายเหตุ: โปรแกรมของกัญญารัตน์และสายฝนและโปรแกรมของผู้วิจัยจะถูกประมวลผลบนเครื่องคอมพิวเตอร์เครื่องเดียวกันในปัญหาที่ 13 และ 14 โดยผู้วิจัย

จะเห็นได้ว่าจากตารางที่ 4.15-4.17 จะเป็นการแสดงผลลัพธ์ของปัญหาที่ได้มีกำหนดขึ้นเอง คือ ปัญหาที่ 13-14 ตามหลักการในการทำ Neighbourhood search ทั้ง 4 หลักการ ซึ่งจะแสดงผลลัพธ์ของการจัดลำดับการทำงานของเครนในรูปแบบของการหาค่า Makespan หรือเวลาที่เครนทำงานเสร็จสิ้นทั้งโดยเฉลี่ยและค่าน้อยที่สุด รวมทั้งเวลาที่ใช้ในการประมวลผลของโปรแกรมทั้งโดยเฉลี่ยและค่าที่น้อยที่สุดเช่นเดียวกัน เมื่อไม่มีการเปลี่ยนแปลงตัวแปร และมีการเปลี่ยนแปลงตัวแปรในการอบอ่อนในครั้งที่ 1 และครั้งที่ 2 ตามลำดับ นอกจากนี้ยังแสดงถึงค่า SD หรือส่วนเบี่ยงเบนมาตรฐาน ซึ่งบ่งบอกว่าผลลัพธ์ที่ได้นั้น มีการกระจายตัวอย่างไร ซึ่งจะเห็นได้ว่าหลักการพิจารณาในหลักการสลับที่ จะให้ค่า Makespan ที่ดีกว่าหลักพิจารณาอื่นและโปรแกรมของกัญญารัตน์และสายฝน

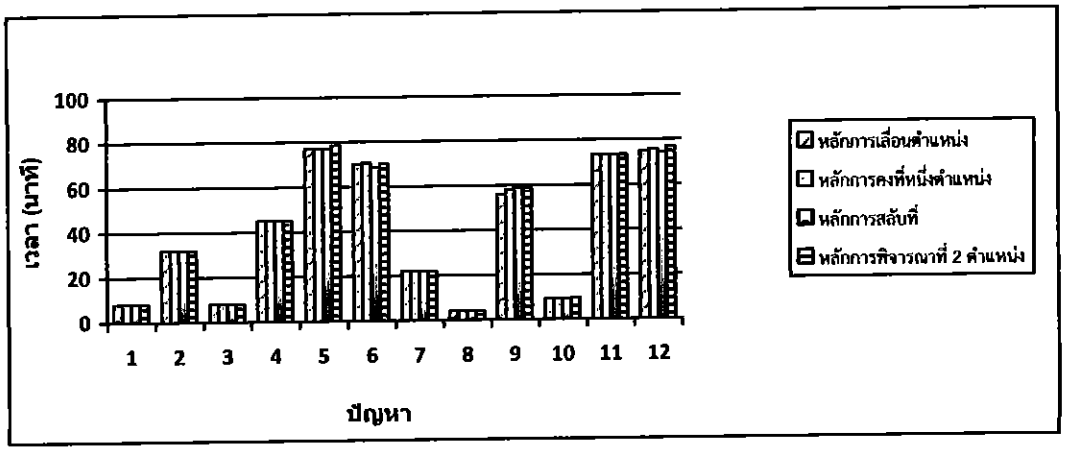
ทั้งนี้การเปรียบเทียบผลลัพธ์ที่ได้นี้จะขอกล่าวต่อไปในหัวข้อการเปรียบเทียบผลลัพธ์ในข้อ 4.3.3 การเปรียบเทียบผลลัพธ์ที่ได้จากปัญหาที่ได้มีการกำหนดขึ้นเอง ต่อไป

4.3 การเปรียบเทียบผลลัพธ์ที่ได้จากการทดสอบโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

การเปรียบเทียบผลลัพธ์นี้จะเป็นการเปรียบเทียบผลลัพธ์ที่ได้ดังข้อ 4.2 กับงานวิจัยอื่น รวมทั้งเป็นการเปรียบเทียบผลลัพธ์ที่ได้ระหว่างผลลัพธ์ด้วยกันของผู้วิจัยเอง ว่ามีความแตกต่างกันอย่างไร ให้ประสิทธิภาพให้การหาคำตอบมากน้อยเพียงใด ซึ่งจะแสดงในรูปแบบทั้งตาราง กราฟ และแผนภูมิเพื่อง่ายต่อความเข้าใจ

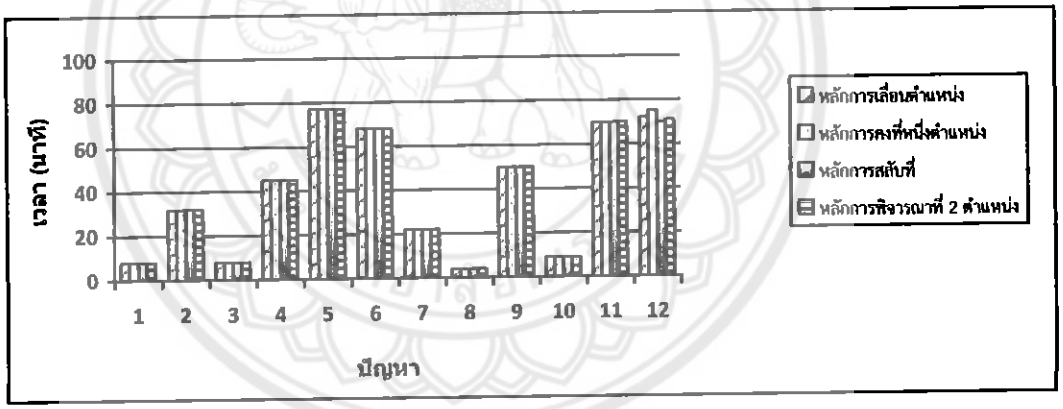
4.3.1 การเปรียบเทียบผลลัพธ์ที่ได้ตามลักษณะที่เคยมีผู้วิจัยมาแล้ว

หลังจากที่ได้แสดงผลลัพธ์ตามลักษณะที่เคยมีผู้วิจัยมาแล้ว ในข้อ 4.2.1 ต่อไปจะเป็นการเปรียบเทียบระหว่างผลลัพธ์ที่ได้ว่ามีความแตกต่างกันอย่างไร ระหว่างวิธีการหา Neighbourhood search ทั้ง 4 หลักการ ดังจะแสดงได้ดังต่อไปนี้



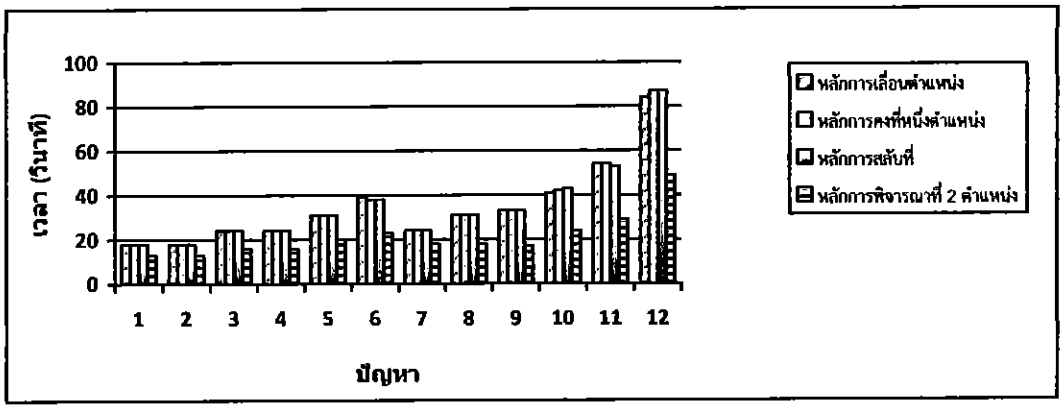
รูปที่ 4.10 แผนภูมิแสดงการเปรียบเทียบระหว่างหลักการในการทำ Neighbourhood search ของเวลาที่เครนทำงานเสร็จสิ้นโดยเฉลี่ย

จากรูปที่ 4.10 เป็นการเปรียบเทียบกันระหว่างหลักการในการทำ Neighbourhood search ทั้ง 4 หลักการ ว่าให้ผลลัพธ์ของ Makespan โดยเฉลี่ย นั้นมีความแตกต่างกันอย่างไรในแต่ละปัญหาที่ทำการทดสอบ ซึ่งจะเห็นได้ว่า ในทุกหลักการในการทำ Neighbourhood search จะให้ผลลัพธ์ที่มีความใกล้เคียงกันสูง



รูปที่ 4.11 แผนภูมิแสดงการเปรียบเทียบระหว่างหลักการในการทำ Neighbourhood search ของเวลาที่เครนทำงานเสร็จสิ้นน้อยที่สุด

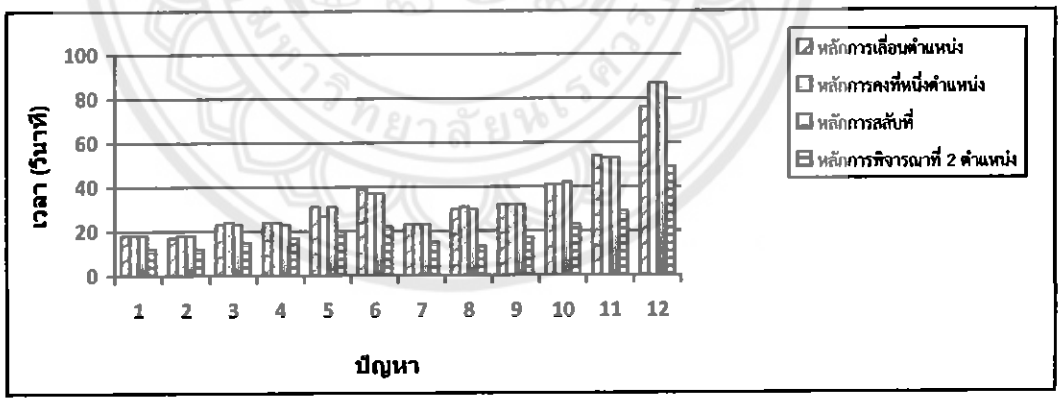
จากรูปที่ 4.11 เป็นการเปรียบเทียบกันระหว่างหลักการในการทำ Neighbourhood search ทั้ง 4 หลักการ ว่าให้ผลลัพธ์ของ Makespan ที่น้อยที่สุด นั้นมีความแตกต่างกันอย่างไรในแต่ละปัญหาที่ทำการทดสอบ ซึ่งจะเห็นได้ว่า ในทุกหลักการในการทำ Neighbourhood search จะให้ผลลัพธ์ที่มีความใกล้เคียงกัน



รูปที่ 4.12 แผนภูมิแสดงการเปรียบเทียบระหว่างหลักการในการทำ Neighbourhood search ของเวลาที่ใช้ในการประมวลผลโดยเฉลี่ย

จากรูปที่ 4.12 เป็นการเปรียบเทียบกันระหว่างหลักการในการทำ Neighbourhood search ทั้ง 4 หลักการ ว่าให้ผลลัพธ์ของเวลาในการประมวลผลโดยเฉลี่ย นั้นมีความแตกต่างกันอย่างไรในแต่ละปัญหาที่ทำการทดสอบ ซึ่งจะเห็นได้ว่า หลักการที่ 4 หลักการพิจารณาที่ 2 ตำแหน่ง ในการทำ Neighbourhood search นั้นจะให้ผลลัพธ์ที่มีแนวโน้มที่น้อยกว่าหลักการอื่นๆประมาณเกือบครึ่งหนึ่ง กล่าวคือ หลักการพิจารณาที่ 2 ตำแหน่ง จะใช้เวลาในการประมวลผลน้อยกว่าหลักการอื่นๆประมาณครึ่งหนึ่งนั่นเอง

ซึ่งในอีก 3 หลักการ ที่เหลือนั่นคือ หลักการเลื่อนตำแหน่ง หลักการคงที่หนึ่งตำแหน่ง และหลักการสลับที่นั้น ให้ผลลัพธ์ที่มีความใกล้เคียงกัน



รูปที่ 4.13 แผนภูมิแสดงการเปรียบเทียบระหว่างหลักการในการทำ Neighbourhood search ของเวลาที่ใช้ในการประมวลผลที่น้อยที่สุด

จากรูปที่ 4.13 เป็นการเปรียบเทียบกันระหว่างหลักการในการทำ Neighbourhood search ทั้ง 4 หลักการ ว่าให้ผลลัพธ์ของเวลาในการประมวลผลที่น้อยที่สุด นั้นมีความแตกต่างกันอย่างไรในแต่ละปัญหาที่ทำการทดสอบ ซึ่งจะเห็นได้ว่า หลักการที่ 4 หลักการพิจารณาที่ 2 ตำแหน่ง ในการทำ Neighbourhood search นั้นจะให้ผลลัพธ์ที่มีแนวโน้มที่น้อยกว่าหลักการอื่นๆประมาณเกือบครึ่งหนึ่ง กล่าวคือ หลักการพิจารณาที่ 2 ตำแหน่ง จะใช้เวลาในการประมวลผลน้อยกว่าหลักการอื่นๆประมาณครึ่งหนึ่งนั่นเอง

ซึ่งในอีก 3 หลักการ ที่เหลือนั่นคือ หลักการเลื่อนตำแหน่ง หลักการคงที่หนึ่งตำแหน่ง และหลักการสลับที่นั้น ให้ผลลัพธ์ที่มีความใกล้เคียงกัน

ต่อไปจะเป็นการเปรียบเทียบผลลัพธ์ที่ได้กับงานวิจัยอื่นที่เคยมีผู้วิจัยอื่นศึกษามาแล้ว อันได้แก่ งานวิจัยระดับปริญญาวิศวกรรมศาสตร์บัณฑิต มหาวิทยาลัยนเรศวร ของ กัญญารัตน์ คุ้มคลอง และ สายฝน ช่างเหล่า ซึ่งเป็นงานวิจัยที่ใช้วิธีการอบอ่อนจำลองเหมือนกัน รวมทั้งงานวิจัยระดับปริญญาวิศวกรรมศาสตร์บัณฑิต มหาวิทยาลัยนเรศวร ของ เอราวิล ถาวร และคณะ ซึ่งใช้โปรแกรมกำหนดการเชิงคณิตศาสตร์ในการแก้ไขปัญหา สุดท้ายคือ งานวิจัยของ Zhu กับ Lim ซึ่งเป็นการสร้างแบบจำลองเชิงคณิตศาสตร์เช่นเดียวกัน คือ โปรแกรม LINGO

นอกจากนี้ ยังเป็นการเทียบผลลัพธ์ระหว่างวิธีการหา Neighbourhood search ทั้ง 4 หลักการ ของวิจัยเองอีกด้วย ว่ามีความแตกต่างกันอย่างไร ซึ่งคอมพิวเตอร์ที่ใช้ในการประมวลผลการจัดลำดับการทำงานของเครนทั้งของผู้วิจัยเองและของผู้วิจัยอื่น มีรายละเอียดดังตารางที่ 4.18

ตารางที่ 4.18 แสดงการเปรียบเทียบรายละเอียดของเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผลการจัดลำดับการทำงานของเครน ของผู้วิจัยเองและของผู้วิจัยอื่น

การประมวลผลการจัดลำดับการทำงานของเครนของ	รายละเอียด
แบบจำลองของ Zhu และ Lim (เอราวิลและคณะ, 2550)	Microsoft Windows XP Professional AMD Turion(tm) 64 Mobile Technology ML-28 1.60 GHz, 512 MB of RAM, Hard disk 60 GB
แบบจำลองของเอราวิลและคณะ (เอราวิลและคณะ, 2550)	
โปรแกรมของกัญญารัตน์และสายฝน (กัญญารัตน์และสายฝน, 2550)	Microsoft Windows XP Professional AMD Sempron(tm) Processor 3100+, 1.80 GHz, 224 MB of RAM, Hard disk 80 GB
โปรแกรมของผู้วิจัย	Windows 7 Untimate Pantium(R) Dual-Core CPU Processor E5200, 2.50 GHz, 2.00 GB of RAM (1.75 GB usable), Hard disk 320 GB

หมายเหตุ: แบบจำลองของ Zhu และ Lim และแบบจำลองของเอราวิลและคณะ ถูกนำมาประมวลผลหาคำตอบการจัดลำดับการทำงานของเครนบนเครื่องคอมพิวเตอร์เครื่องเดียวกัน โดยการหาคำตอบและเปรียบเทียบคำตอบของเอราวิลและคณะ

ต่อไปจะเป็นการแสดงผลลัพธ์ที่ได้จากงานวิจัยอื่นทั้ง 3 งานวิจัยเมื่อเทียบกับผลลัพธ์ของผู้วิจัยเอง ดังที่ได้กล่าวมาแล้วข้างต้น

ตารางที่ 4.19 แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่เครนทำงานเสร็จสิ้นกับงานวิจัยอื่น

งานที่	แบบจำลองของ Zhu และ Lim	แบบจำลองของ เอราวิลและคณะ	โปรแกรมของกัญญารัตน์และสายฝน		โปรแกรมจัดลำดับการทำงานของเครนของผู้วิจัย							
					หลักการเลื่อนตำแหน่ง		หลักการคงที่หนึ่งตำแหน่ง		หลักการสลับที่		หลักการพิจารณาที่ 2 ตำแหน่ง	
	เวลา	เวลา	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด
ปัญหาขนาดเล็ก												
1	8	8	8	8	8	8	8	8	8	8	8	8
2	32	32	32	32	32	32	32	32	32	32	32	32
ปัญหาขนาดกลาง												
3	8	8	8	8	8	8	8	8	8	8	8	8
4	45	45	45	45	45	45	45	45	45	45	45	45
5	77	77	77	77	77	77	77	77	77	77	78.5	77
6	68	68	70.5	68	70	68	70.6	68	68.6	68	70.2	68
ปัญหาขนาดใหญ่												
7	22	22	22	22	22	22	22	22	22	22	22	22
8	4	4	4	4	4	4	4	4	4	4	4	4
9	50	50	56	50	56	50	58	50	59	50	59	50
10	9*	9	9	9	9	9	9	9	9	9	9.5	9
11	74*	66	73	70	73.4	70	73.3	70	73.3	70	74.8	70
12	[91,41]**	[69,49]***	75.5	74	74.8	72	75.8	75	74.2	70	76.9	71

หมายเหตุ : * คือ คำตอบยังคงเป็น Feasible

** คือ คำตอบยังคงเป็น Feasible และอยู่ในช่วง 41-91

*** คือ คำตอบยังคงเป็น Feasible และอยู่ในช่วง 49-69

รายละเอียดเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผลแสดงดังตารางที่ 4.18

จากตารางที่ 4.19 จะเห็นได้ว่าเป็นการแสดงการเปรียบเทียบค่า Makespan ระหว่างงานวิจัยต่างๆดังที่กล่าวมาข้างต้นกับงานของผู้วิจัยเองว่า มีความแตกต่างหรือใกล้เคียงกันอย่างไร ในทั้ง 12 ปัญหาการทดสอบ

ต่อไปจะเป็นการเปรียบเทียบงานของผู้วิจัยเองว่า เมื่อเทียบเป็นเปอร์เซ็นต์แล้ว ผลลัพธ์ของผู้วิจัยให้ค่าที่แตกต่างกันมากน้อยเพียงใดเมื่อเทียบกับค่า Makespan ที่น้อยที่สุด ซึ่งจะพิจารณาในทุกหลักการในการทำ Neighbourhood search

ตารางที่ 4.20 แสดงการเปรียบเทียบค่า Makespan ที่ดีที่สุดกับผลลัพธ์ของผู้วิจัยเป็นเปอร์เซ็นต์

งานที่	ค่า Makespan ที่น้อยที่สุด (นาที)	หลักการเลื่อนตำแหน่ง		หลักการคงที่หนึ่งตำแหน่ง		หลักการสลับที่		หลักการพิจารณาที่ 2 ตำแหน่ง	
		เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด
ปัญหาขนาดเล็ก									
1	8	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
2	32	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
ปัญหาขนาดกลาง									
3	8	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
4	45	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
5	77	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.0%	0.0%
6	68	3%	0.0%	3.8%	0.0%	0.9%	0.0%	3.2%	0.0%
ปัญหาขนาดใหญ่									
7	22	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
8	4	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
9	50	12%	0.0%	16%	0.0%	18%	0.0%	18%	0.0%
10	9	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	5.6%	0.0%
11	66	11.2%	6.1%	11.1%	6.1%	11.1%	6.1%	13.3%	6.1%
12	49*	52.7%	46.9%	54.7%	53%	51.4%	42.9%	56.9%	44.9%

หมายเหตุ : * คือ ค่า Makespan ต่ำสุดที่สามารถเป็นไปได้ จากช่วง [69,49]

จากตารางที่ 4.20 จะเห็นได้ว่าเป็นการเปรียบเทียบค่า Makespan ที่น้อยที่สุดกับผลลัพธ์ของวิจัยตามแต่ละหลักการเป็นเปอร์เซ็นต์ จะเห็นได้ว่า ค่าเวลาการทำงานที่ได้ไม่ค่อยมีความแตกต่างจากค่าเวลาการทำงานที่ดีที่สุดในทุกๆหลักการการทำ Neighbourhood search ยกเว้นในปัญหาที่ 6, 9, 11 และ 12 ที่มีความแตกต่างจากค่าที่ดีที่สุดมาก ทั้งนี้ในปัญหาที่ 12 ได้มีการใช้ค่าต่ำสุดของค่า Makespan ที่สามารถเป็นไปได้มาใช้ในการเปรียบเทียบ

ต่อไปจะเป็นการแสดงการเปรียบเทียบผลลัพธ์ด้านเวลาในการประมวลผลระหว่างผลลัพธ์ของผู้วิจัยเองและผลลัพธ์ของผู้วิจัยอื่น ดังแสดงได้ดังต่อไปนี้

ตารางที่ 4.21 แสดงการเปรียบเทียบผลลัพธ์ของเวลาในการประมวลผลกับงานวิจัยอื่น

งานที่	แบบจำลองของ Zhu และ Lim	แบบจำลองของ เอราวิล และคณะ	โปรแกรมของ กัญญารัตน์และสายฝน		โปรแกรมจัดลำดับการทำงานของเครนของผู้วิจัย							
					หลักการเลื่อนตำแหน่ง		หลักการคงที่หนึ่งตำแหน่ง		หลักการสลับที่		หลักการพิจารณาที่ 2 ตำแหน่ง	
	เวลา	เวลา	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด	เวลาโดยเฉลี่ย	เวลาที่น้อยที่สุด
ปัญหาขนาดเล็ก												
1	4 s.	1 s.	21.5 s.	19 s.	18.3 s.	18 s.	18.4 s.	18 s.	18.5 s.	18 s.	12.8 s.	12 s.
2	3 s.	1 s.	18.9 s.	18 s.	17.9 s.	17 s.	18.1 s.	18 s.	18.3 s.	18 s.	12.8 s.	12 s.
ปัญหาขนาดกลาง												
3	4 s.	1 s.	23.8 s.	23 s.	23.8 s.	23 s.	24.2 s.	24 s.	24.2 s.	23 s.	15.9 s.	15 s.
4	7min.43s.	2 s.	24.3 s.	23 s.	24.3 s.	24 s.	24.2 s.	24 s.	23.8 s.	23 s.	15.7 s.	12 s.
5	7min.43s.	1min.11s.	29.3 s.	29 s.	31.1 s.	31 s.	30.7 s.	27 s.	31 s.	31 s.	20.1 s.	20 s.
6	16min.19s.	1 min.	36.5 s.	35 s.	39.2 s.	39 s.	37.9 s.	37 s.	37.8 s.	37 s.	22.9 s.	22 s.
ปัญหาขนาดใหญ่												
7	55 s.	1 s.	26.8 s.	25 s.	23.8 s.	23 s.	23.7 s.	23 s.	23.8 s.	23 s.	15.6 s.	15 s.
8	1min43s.	1 s.	30.9 s.	30 s.	30.8 s.	30 s.	31.3 s.	31 s.	31 s.	30 s.	17.6 s.	17 s.
9	39min.25s.	4 s.	37.9 s.	37 s.	32.6 s.	32 s.	32.8 s.	32 s.	32.8 s.	32 s.	17.2 s.	17 s.
10	71hr.9min.*	24 s.	45.5 s	45 s.	41.3 s.	41 s.	42.1 s.	41 s.	42.7 s.	42 s.	24 s.	23 s.
11	8hr.45min.*	40min.34s.	54 s.	53 s.	54 s.	54 s.	54 s.	53 s.	53.4 s.	53 s.	29.1 s.	29 s.
12	71hr.59min.**	72 hr.***	80.1 s.	73 s.	83.7 s.	76 s.	87.2 s.	87 s.	87.2 s.	87 s.	49.5 s.	49 s.

หมายเหตุ : * คือ คำตอบยังคงเป็น Feasible

** คือ คำตอบยังคงเป็น Feasible และอยู่ในช่วง 41-91

*** คือ คำตอบยังคงเป็น Feasible และอยู่ในช่วง 49-69

รายละเอียดเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผลแสดงดังตารางที่ 4.18

จากตารางที่ 4.21 จะเห็นได้ว่าเป็นการแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลระหว่างงานวิจัยต่างๆดังที่กล่าวมาข้างต้นกับงานของผู้วิจัยเองว่า มีความแตกต่างหรือใกล้เคียงกันอย่างไร ในทั้ง 12 ปัญหาการทดสอบ ซึ่งแบบจำลองของ Zhu และ Lim จะใช้เวลาในการประมวลผลมากที่สุด รองลงมาคือ แบบจำลองของเอราวิลและคณะ และโปรแกรมของกัญญารัตน์และสายฝนตามลำดับ ซึ่งเวลาในการประมวลของของผู้วิจัยใน 3 หลักการแรก คือ หลักการเลื่อนตำแหน่ง หลักการคงที่หนึ่งตำแหน่ง และหลักการสลับที่ จะใช้เวลาในการประมวลผลใกล้เคียงกัน แต่หลักการพิจารณาที่ 2 ตำแหน่ง จะให้ผลลัพธ์น้อยกว่าประมาณครึ่งหนึ่ง

ซึ่งผลการเปรียบเทียบทั้งหมดที่กล่าวมาจะทำการวิเคราะห์ผลในหัวข้อ 4.4 การวิเคราะห์ผลการทดลอง

4.3.2 การเปรียบเทียบผลลัพธ์ที่ได้เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลังจากที่ได้แสดงผลลัพธ์ที่ได้เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อนในหัวข้อ 4.2.2 แล้ว ต่อไปจะเป็นการเปรียบเทียบผลลัพธ์ที่ได้จากการเปลี่ยนแปลงตัวแปรในรูปแบบของตาราง ในหน้าถัดไป

ตารางที่ 4.22 แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่เครนทำงานเสร็จสิ้นเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลักการในการทำ Neighbourhood search	เมื่อไม่มีการเปลี่ยนแปลงตัวแปร		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2	
	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)
ปัญหาขนาดเล็ก : ปัญหาที่ 1										
1.หลักการเลื่อนตำแหน่ง	8	8	8	8	8	8	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	8	8	8	8	8	8	0.0	0.0	0.0	0.0
3.หลักการสลับที่	8	8	8	8	8	8	0.0	0.0	0.0	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	8	8	8	8	8	8	0.0	0.0	0.0	0.0
ปัญหาขนาดเล็ก : ปัญหาที่ 2										
1.หลักการเลื่อนตำแหน่ง	32	32	32	32	32	32	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	32	32	32	32	32	32	0.0	0.0	0.0	0.0
3.หลักการสลับที่	32	32	32	32	32	32	0.0	0.0	0.0	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	32	32	32	32	32	32	0.0	0.0	0.0	0.0
ปัญหาขนาดกลาง : ปัญหาที่ 3										
1.หลักการเลื่อนตำแหน่ง	8	8	8	8	8	8	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	8	8	8	8	8	8	0.0	0.0	0.0	0.0
3.หลักการสลับที่	8	8	8	8	8	8	0.0	0.0	0.0	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	8	8	8	8	8	8	0.0	0.0	0.0	0.0

ตารางที่ 4.22 (ต่อ) แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่เครนทำงานเสร็จสิ้นเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลักการในการทำ Neighbourhood search	เมื่อไม่มีการเปลี่ยนแปลงตัวแปร		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2	
	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)
ปัญหาขนาดกลาง : ปัญหาที่ 4										
1.หลักการเลื่อนตำแหน่ง	45	45	45	45	45	45	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	45	45	45	45	45	45	0.0	0.0	0.0	0.0
3.หลักการสลับที่	45	45	45	45	45	45	0.0	0.0	0.0	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	45	45	45	45	45	45	0.0	0.0	0.0	0.0
ปัญหาขนาดกลาง : ปัญหาที่ 5										
1.หลักการเลื่อนตำแหน่ง	77	77	77	77	77	77	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	77	77	77	77	77	77	0.0	0.0	0.0	0.0
3.หลักการสลับที่	77	77	77	77	77	77	0.0	0.0	0.0	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	78.5	77	77.9	77	77.3	77	-0.7	0.0	-1.5	0.0
ปัญหาขนาดกลาง : ปัญหาที่ 6										
1.หลักการเลื่อนตำแหน่ง	70	68	68.8	68	68.4	68	-1.7	0.0	-2.3	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	70.6	68	69.6	68	68.8	68	-1.4	0.0	-2.5	0.0
3.หลักการสลับที่	68.6	68	68.7	68	68.3	68	0.1	0.0	-0.4	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	70.2	68	70.3	68	70	68	0.1	0.0	-0.3	0.0

ตารางที่ 4.22 (ต่อ) แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่เครนทำงานเสร็จสิ้นเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลักการในการทำ Neighbourhood search	เมื่อไม่มีการเปลี่ยนแปลงตัวแปร		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2	
	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)
ปัญหาขนาดใหญ่ : ปัญหาที่ 7										
1.หลักการเลื่อนตำแหน่ง	22	22	22	22	22	22	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	22	22	22	22	22	22	0.0	0.0	0.0	0.0
3.หลักการสลับที่	22	22	22	22	22	22	0.0	0.0	0.0	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	22	22	22	22	22	22	0.0	0.0	0.0	0.0
ปัญหาขนาดใหญ่ : ปัญหาที่ 8										
1.หลักการเลื่อนตำแหน่ง	4	4	4	4	4	4	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	4	4	4	4	4	4	0.0	0.0	0.0	0.0
3.หลักการสลับที่	4	4	4	4	4	4	0.0	0.0	0.0	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	4	4	4	4	4	4	0.0	0.0	0.0	0.0
ปัญหาขนาดใหญ่ : ปัญหาที่ 9										
1.หลักการเลื่อนตำแหน่ง	56	50	56	50	56	50	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	58	50	58	50	57	50	0.0	0.0	-1.7	0.0
3.หลักการสลับที่	59	50	58	50	57	50	-1.7	0.0	-3.4	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	59	50	59	50	58	50	0.0	0.0	-1.7	0.0

ตารางที่ 4.22 (ต่อ) แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่เครนทำงานเสร็จสิ้นเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลักการในการทำ Neighbourhood search	เมื่อไม่มีการเปลี่ยนแปลงตัวแปร		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2	
	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (นาที)	เวลาที่น้อยที่สุด (นาที)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)
ปัญหาขนาดใหญ่ : ปัญหาที่ 10										
1.หลักการเลื่อนตำแหน่ง	9	9	9	9	9	9	0.0	0.0	0.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	9	9	9	9	9	9	0.0	0.0	0.0	0.0
3.หลักการสลับที่	9	9	9	9	9	9	0.0	0.0	0.0	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	9.5	9	9.2	9	9.1	9	-3.2	0.0	-4.2	0.0
ปัญหาขนาดใหญ่ : ปัญหาที่ 11										
1.หลักการเลื่อนตำแหน่ง	73.4	70	72.1	70	71.9	70	-1.8	0.0	-2.0	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	73.3	70	72.5	66	71.9	66	-1.1	-5.7	-1.9	-5.7
3.หลักการสลับที่	73.3	70	70.7	66	69.5	66	-3.5	-5.7	-5.2	-5.7
4.หลักการพิจารณาที่ 2 ตำแหน่ง	74.8	70	72.9	66	72.6	70	-2.5	-5.7	-2.9	0.0
ปัญหาขนาดใหญ่ : ปัญหาที่ 12										
1.หลักการเลื่อนตำแหน่ง	74.8	72	74.8	72	73.7	70	0.0	0.0	-1.5	-2.8
2.หลักการคงที่หนึ่งตำแหน่ง	75.8	75	76	75	74.4	72	0.3	0.0	-1.8	-4.0
3.หลักการสลับที่	74.2	70	75	74	74.4	72	1.1	5.7	0.3	2.9
4.หลักการพิจารณาที่ 2 ตำแหน่ง	76.9	71	77.7	74	76	75	1.0	4.2	-1.2	1.4

หมายเหตุ : ค่าเปอร์เซ็นต์ที่เป็น ลบ แสดงให้เห็นว่า เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อนจะทำให้มีค่า Makespan ที่น้อยกว่าเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ในขณะที่ค่าเปอร์เซ็นต์ที่เป็น บวก แสดงให้เห็นว่า เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน จะทำให้มีค่า Makespan ที่มากกว่าเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

ตารางที่ 4.23 แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่ใช้ในการประมวลผลเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลักการในการทำ Neighbourhood search	เมื่อไม่มีการเปลี่ยนแปลงตัวแปร		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2	
	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)
ปัญหาขนาดเล็ก : ปัญหาที่ 1										
1.หลักการเลื่อนตำแหน่ง	18.3	18	18.4	18	18	18	0.5	0.0	-1.6	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	18.4	18	18.4	18	17.9	17	0.0	0.0	-2.7	-5.6
3.หลักการสลับที่	18.5	18	18.4	18	17.8	17	-0.5	0.0	-3.8	-5.6
4.หลักการพิจารณาที่ 2 ตำแหน่ง	12.8	12	13.1	13	12.7	12	2.3	8.3	-0.8	0.0
ปัญหาขนาดเล็ก : ปัญหาที่ 2										
1.หลักการเลื่อนตำแหน่ง	17.9	17	18.5	18	18	18	3.4	5.9	0.6	5.9
2.หลักการคงที่หนึ่งตำแหน่ง	18.1	18	18.4	18	18.2	18	1.7	0.0	0.6	0.0
3.หลักการสลับที่	18.3	18	18.5	18	18.1	18	1.1	0.0	-1.1	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	12.8	12	13	13	12.4	12	1.6	8.3	-3.1	0.0
ปัญหาขนาดกลาง : ปัญหาที่ 3										
1.หลักการเลื่อนตำแหน่ง	23.8	23	24.8	24	23.9	23	4.2	4.3	0.4	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	24.2	24	24.6	24	23.9	23	1.7	0.0	-1.2	-4.2
3.หลักการสลับที่	24.2	23	24.6	24	24	24	1.7	4.3	-0.8	4.3
4.หลักการพิจารณาที่ 2 ตำแหน่ง	15.9	15	16	16	15.7	15	0.6	6.7	-1.3	0.0

ตารางที่ 4.23 (ต่อ) แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่ใช้ในการประมวลผลเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลักการในการทำ Neighbourhood search	เมื่อไม่มีการเปลี่ยนแปลงตัวแปร		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2	
	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)
ปัญหาขนาดกลาง : ปัญหาที่ 4										
1.หลักการเลื่อนตำแหน่ง	24.3	24	24.8	24	24	24	2.1	0.0	-1.2	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	24.2	24	25	25	24.2	24	3.3	4.2	0.0	0.0
3.หลักการสลับที่	23.8	23	25	25	23.9	23	5.0	8.7	0.4	0.0
4.หลักการพิจารณาที่ 2 ตำแหน่ง	15.7	12	16.2	16	15.9	15	3.2	33.3	1.3	25
ปัญหาขนาดกลาง : ปัญหาที่ 5										
1.หลักการเลื่อนตำแหน่ง	31.1	31	32	32	31	31	2.9	3.2	-0.3	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	30.7	27	32	32	31	31	4.2	18.5	1.0	14.8
3.หลักการสลับที่	31	31	32	32	31	30	3.2	3.2	0.0	-3.2
4.หลักการพิจารณาที่ 2 ตำแหน่ง	20.1	20	20.5	20	19.6	19	2.0	0.0	-2.5	-5.0
ปัญหาขนาดกลาง : ปัญหาที่ 6										
1.หลักการเลื่อนตำแหน่ง	39.2	39	49.8	39	39	38	27.0	0.0	-0.5	-2.6
2.หลักการคงที่หนึ่งตำแหน่ง	37.9	37	40.2	40	39	38	6.1	8.1	2.9	2.7
3.หลักการสลับที่	37.8	37	39.7	39	39	38	5.0	5.4	3.2	2.7
4.หลักการพิจารณาที่ 2 ตำแหน่ง	22.9	22	24	24	24	23.9	4.8	9.1	4.8	8.6

ตารางที่ 4.23 (ต่อ) แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่ใช้ในการประมวลผลเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลักการในการทำ Neighbourhood search	เมื่อไม่มีการเปลี่ยนแปลงตัวแปร		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2	
	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)
ปัญหาขนาดใหญ่ : ปัญหาที่ 7										
1.หลักการเลื่อนตำแหน่ง	23.8	23	24.3	23	23.5	23	2.1	0.0	-1.3	0.0
2.หลักการคงที่หนึ่งตำแหน่ง	23.7	23	24	24	23.7	23	1.3	4.3	0.0	0.0
3.หลักการสลับที่	23.8	23	24.5	23	24.1	24	2.9	0.0	1.3	4.3
4.หลักการพิจารณาที่ 2 ตำแหน่ง	15.6	15	15.6	15	15.2	15	0.0	0.0	-2.6	0.0
ปัญหาขนาดใหญ่ : ปัญหาที่ 8										
1.หลักการเลื่อนตำแหน่ง	30.8	30	31.2	30	32	32	1.3	0.0	3.9	6.7
2.หลักการคงที่หนึ่งตำแหน่ง	31.3	31	31.3	31	32	32	0.0	0.0	2.2	3.2
3.หลักการสลับที่	31	30	31.5	30	31.9	31	1.6	0.0	2.9	3.3
4.หลักการพิจารณาที่ 2 ตำแหน่ง	17.6	17	18.1	17	17.9	17	2.8	0.0	1.7	0.0
ปัญหาขนาดใหญ่ : ปัญหาที่ 9										
1.หลักการเลื่อนตำแหน่ง	32.6	32	32.7	31	33.2	31	0.3	-3.1	1.8	-3.1
2.หลักการคงที่หนึ่งตำแหน่ง	32.8	32	32.8	31	33.4	31	0.0	-3.1	1.8	-3.1
3.หลักการสลับที่	32.8	32	32.2	31	33.8	33	-1.8	-3.1	3.0	3.1
4.หลักการพิจารณาที่ 2 ตำแหน่ง	17.2	17	17.6	17	17.4	17	2.3	0.0	1.2	0.0

ตารางที่ 4.23 (ต่อ) แสดงการเปรียบเทียบผลลัพธ์ของเวลาที่ใช้ในการประมวลผลเมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

หลักการในการทำ Neighbourhood search	เมื่อไม่มีการเปลี่ยนแปลงตัวแปร		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1		ผลต่างระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปร กับเมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2	
	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (วินาที)	เวลาที่น้อยที่สุด (วินาที)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)	เวลาเฉลี่ย (%)	เวลาที่น้อยที่สุด (%)
ปัญหาขนาดใหญ่ : ปัญหาที่ 10										
1.หลักการเลื่อนตำแหน่ง	41.3	41	41.5	40	41.9	40	0.5	-2.4	1.5	-2.4
2.หลักการคงที่หนึ่งตำแหน่ง	42.1	41	41.9	41	41.8	40	-0.5	0.0	-0.7	-2.4
3.หลักการสลับที่	42.7	42	42.4	40	42.1	41	-0.7	-4.8	-1.4	-2.3
4.หลักการพิจารณาที่ 2 ตำแหน่ง	24	23	23.9	23	24.8	23	-0.4	0.0	3.3	0.0
ปัญหาขนาดใหญ่ : ปัญหาที่ 11										
1.หลักการเลื่อนตำแหน่ง	54	54	53.8	50	53.7	50	-0.4	-7.4	-0.6	-7.4
2.หลักการคงที่หนึ่งตำแหน่ง	54	53	53.8	50	53.7	51	-0.4	-5.7	-0.6	-3.8
3.หลักการสลับที่	53.4	53	55.1	51	54.3	51	3.2	-3.8	1.7	-3.8
4.หลักการพิจารณาที่ 2 ตำแหน่ง	29.1	29	30.5	27	29.6	28	4.8	-6.9	1.7	-3.4
ปัญหาขนาดใหญ่ : ปัญหาที่ 12										
1.หลักการเลื่อนตำแหน่ง	83.7	76	86.3	85	90.7	89	3.1	11.8	8.4	17.1
2.หลักการคงที่หนึ่งตำแหน่ง	87.2	87	86.9	85	91	88	-0.3	-2.3	4.4	1.1
3.หลักการสลับที่	87.2	87	94	86	89.9	83	7.8	-1.1	3.1	-4.6
4.หลักการพิจารณาที่ 2 ตำแหน่ง	49.5	49	51.9	49	54.5	49	4.8	0.0	10.1	0.0

หมายเหตุ : ค่าเปอร์เซ็นต์ที่เป็น ลบ แสดงให้เห็นว่า เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อนจะทำให้มีเวลาในการประมวลผลที่น้อยกว่าเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ในขณะที่ค่าเปอร์เซ็นต์ที่เป็น บวก แสดงให้เห็นว่า เมื่อมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน จะทำให้มีเวลาในการประมวลผลที่มากกว่าเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน

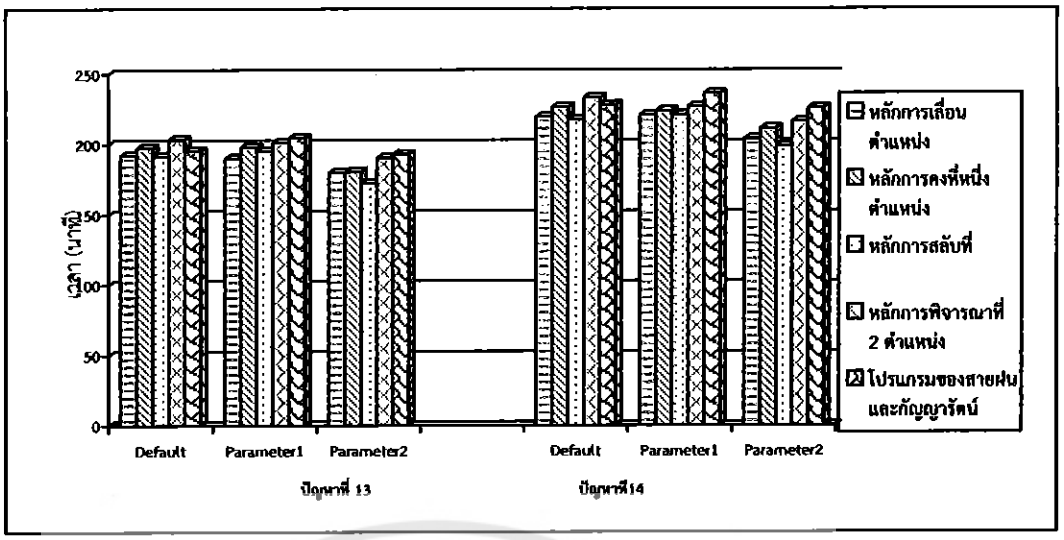
จากตารางที่ 4.22 เป็นการเปรียบเทียบค่า Makespan ระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ครั้งที่ 1 และมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ครั้งที่ 2 โดยแสดงทั้งเวลาในการเปรียบเทียบ และแสดงการเปรียบเทียบความแตกต่างเป็นเปอร์เซ็นต์ระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน กับมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ครั้งที่ 1 และเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน กับมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ครั้งที่ 2 จะเห็นได้ว่า ค่า Makespan เมื่อเปลี่ยนแปลงตัวแปรครั้งที่ 2 จะให้ค่าที่ต่ำที่สุด และเมื่อเทียบเป็นเปอร์เซ็นต์กับเมื่อไม่มีการเปลี่ยนแปลงตัวแปรแล้วพบว่าจะมีเปอร์เซ็นต์ความแตกต่างค่อนข้างมาก

จากตารางที่ 4.23 เป็นการเปรียบเทียบเวลาในการประมวลผล ระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ครั้งที่ 1 และมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ครั้งที่ 2 โดยแสดงทั้งเวลาในการเปรียบเทียบ และแสดงการเปรียบเทียบความแตกต่างเป็นเปอร์เซ็นต์ระหว่างเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน กับมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ครั้งที่ 1 และเมื่อไม่มีการเปลี่ยนแปลงตัวแปรในการอบอ่อน กับมีการเปลี่ยนแปลงตัวแปรในการอบอ่อน ครั้งที่ 2 จะเห็นได้ว่า การเปลี่ยนแปลงตัวแปรนั้น ไม่มีผลต่อเวลาในการประมวลผล แต่เวลาในการประมวลผลจะขึ้นอยู่กับหลักการในการทำ Neighbourhood search ซึ่งหลักการพิจารณาที่ 2 ตำแหน่งจะให้ค่าเวลาในการประมวลผลต่ำที่สุด คือ คิดเป็นครึ่งหนึ่งของหลักการที่เหลือ

ซึ่งผลการเปรียบเทียบทั้งหมดที่กล่าวมาจะทำการวิเคราะห์ผลในหัวข้อ 4.4 การวิเคราะห์ผลการทดลอง

4.3.3 การเปรียบเทียบผลลัพธ์ที่ได้จากปัญหาที่ได้มีการกำหนดขึ้นเอง

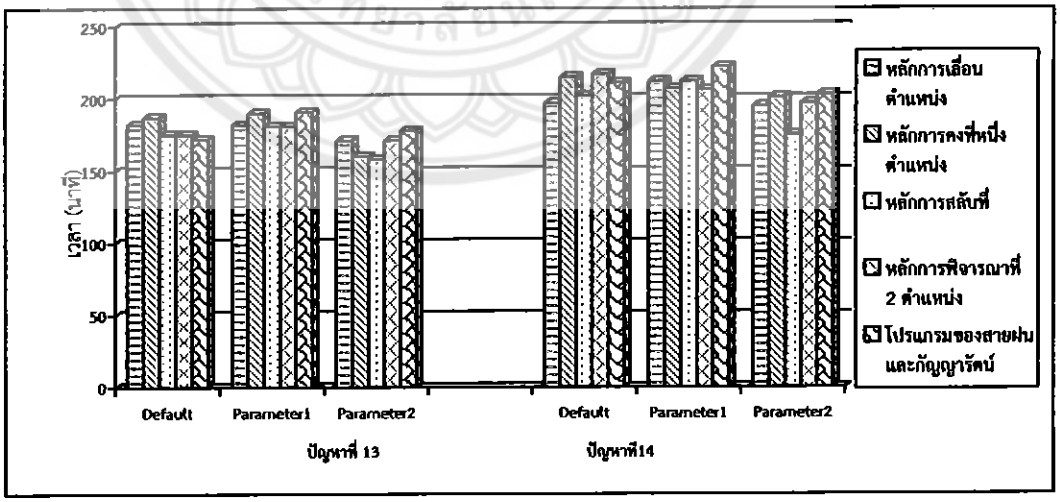
หลังจากที่ได้แสดงผลลัพธ์ที่ได้จากปัญหาที่ได้มีการกำหนดขึ้นเองในหัวข้อ 4.2.2 แล้วต่อไปจะเป็นการแสดงแผนภูมิการเปรียบเทียบระหว่างหลักการในการอบอ่อนของผู้วิจัยและโปรแกรมของสายฝนและกัญญารัตน์ในด้านเวลาการที่เครนทำงานเสร็จสิ้นและเวลาในการประมวลผล



รูปที่ 4.14 แสดงแผนภูมิการเปรียบเทียบเวลาที่เครนทำงานเสร็จสิ้นโดยเฉลี่ย ของปัญหาที่ได้มีการกำหนดขึ้นเอง

หมายเหตุ : Default คือ เมื่อไม่มีการเปลี่ยนแปลงตัวแปร,
Parameter 1 คือ เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1,
Parameter 2 คือ เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2

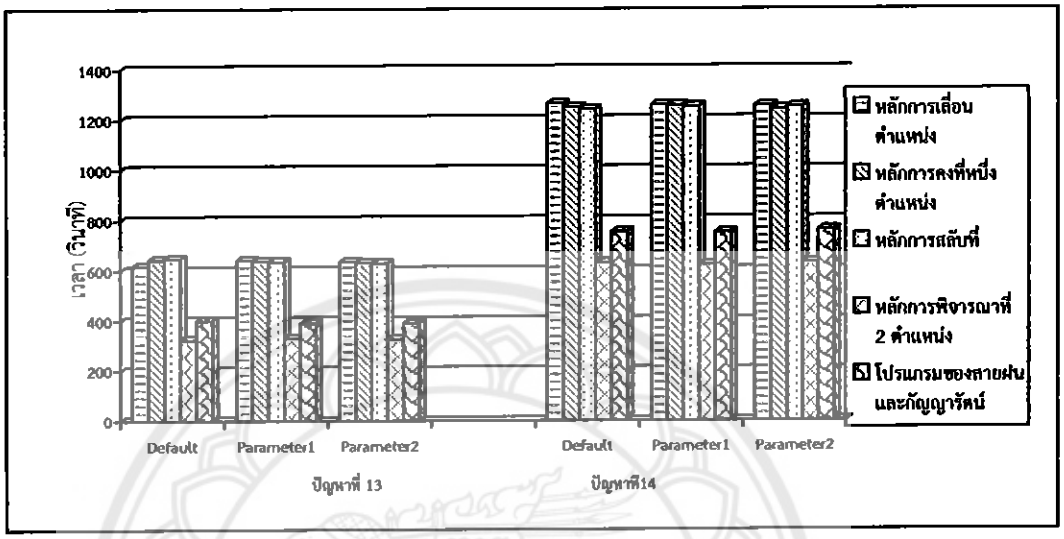
จากรูปที่ 4.14 เป็นการเปรียบเทียบค่า Makespan โดยเฉลี่ย ระหว่างหลักการในการพิจารณา ซึ่งรวมทั้งโปรแกรมของกัญญารัตน์และสายฝนด้วย และการเปลี่ยนแปลงตัวแปรในการอบอ่อนจำลอง ซึ่งจะเห็นได้ว่าหลักการพิจารณาในหลักการสลับที่ และการเปลี่ยนแปลงตัวแปร ครั้งที่ 2 จะส่งผลให้ค่า Makespan น้อยกว่าหลักการและการเปลี่ยนแปลงตัวแปรอื่นๆ



รูปที่ 4.15 แสดงแผนภูมิการเปรียบเทียบเวลาที่เครนทำงานเสร็จสิ้นน้อยที่สุด ของปัญหาที่ได้มีการกำหนดขึ้นเอง

หมายเหตุ : Default คือ เมื่อไม่มีการเปลี่ยนแปลงตัวแปร,
Parameter 1 คือ เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1,
Parameter 2 คือ เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2

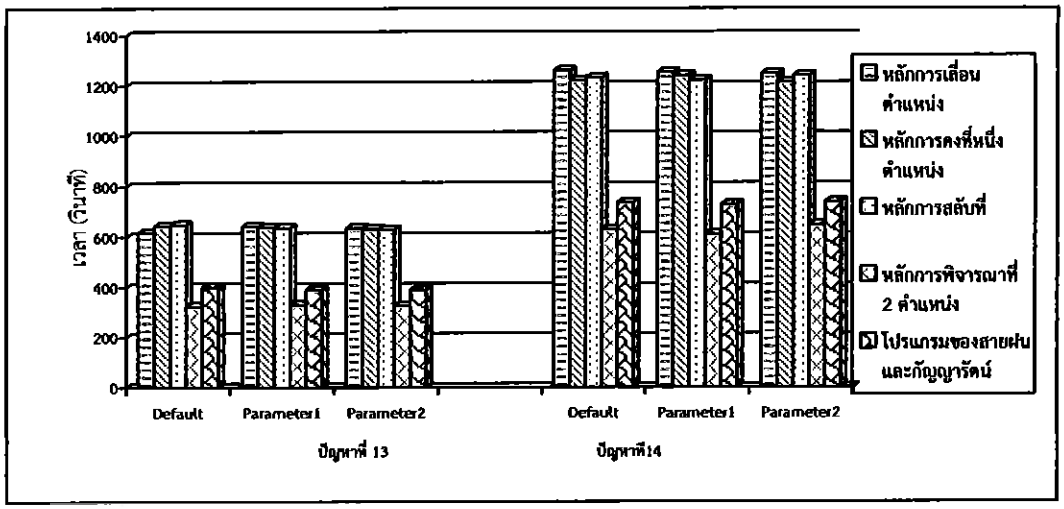
จากรูปที่ 4.15 เป็นการเปรียบเทียบค่า Makespan ที่น้อยที่สุด ระหว่างหลักการในการพิจารณา ซึ่งรวมทั้งโปรแกรมของกัญญารัตน์และสายฝนด้วย และการเปลี่ยนแปลงตัวแปรในการอบอ่อนจำลอง ซึ่งจะเห็นได้ว่าหลักการพิจารณาในหลักการสลับที่ และการเปลี่ยนแปลงตัวแปร ครั้งที่ 2 จะส่งผลให้ค่า Makespan น้อยกว่าหลักการและการเปลี่ยนแปลงตัวแปรอื่นๆ



รูปที่ 4.16 แสดงแผนภูมิการเปรียบเทียบเวลาที่ใช้ประมวลผลโดยเฉลี่ย ของปัญหาที่ได้มีการกำหนดขึ้นเอง

หมายเหตุ : Default คือ เมื่อไม่มีการเปลี่ยนแปลงตัวแปร,
Parameter 1 คือ เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1,
Parameter 2 คือ เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2

จากรูปที่ 4.16 เป็นการเปรียบเทียบเวลาที่ใช้ในการประมวลผลโดยเฉลี่ย ระหว่างหลักการในการพิจารณา ซึ่งรวมทั้งโปรแกรมของกัญญารัตน์และสายฝนด้วย และการเปลี่ยนแปลงตัวแปรในการอบอ่อนจำลอง ซึ่งจะเห็นได้ว่าหลักการพิจารณาใน 2 จะส่งผลให้เวลาที่ใช้ในการประมวลผลโดยเฉลี่ยน้อยกว่าหลักการอื่นที่พิจารณาครึ่งหนึ่ง และโปรแกรมของกัญญารัตน์และสายฝนจะใช้เวลาในการประมวลผลน้อยกว่าหลักการเลื่อนตำแหน่ง หลักการคงที่หนึ่งตำแหน่ง และหลักการสลับที่ แต่ใช้เวลาในการประมวลผลมากกว่าหลักการพิจารณาที่ 2 ตำแหน่ง เล็กน้อย



รูปที่ 4.17 แสดงแผนภูมิการเปรียบเทียบเวลาที่ใช้ประมวลผลที่น้อยที่สุด ของปัญหาที่ได้มีการกำหนดขึ้นเอง

หมายเหตุ : Default คือ เมื่อไม่มีการเปลี่ยนแปลงตัวแปร,
Parameter 1 คือ เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1,
Parameter 2 คือ เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2

จากรูปที่ 4.17 เป็นการเปรียบเทียบเวลาที่ใช้ในการประมวลผลโดยที่น้อยที่สุด ระหว่างหลักการในการพิจารณา ซึ่งรวมทั้งโปรแกรมของกัญญารัตน์และสายฝนด้วย และการเปลี่ยนแปลงตัวแปรในการอบอ่อนจำลอง ซึ่งจะเห็นได้ว่าหลักการพิจารณาใน 2 จะส่งผลให้เวลาที่ใช้ในการประมวลผลน้อยกว่าหลักการอื่นที่พิจารณาครั้งหนึ่ง และโปรแกรมของกัญญารัตน์และสายฝนจะใช้เวลาในการประมวลผลน้อยกว่าหลักการเลื่อนตำแหน่ง หลักการคงที่หนึ่งตำแหน่ง และหลักการสลับที่ แต่ใช้เวลาในการประมวลผลมากกว่าหลักการพิจารณาที่ 2 ตำแหน่ง เล็กน้อย

ต่อไปเป็นการเปรียบเทียบผลลัพธ์ระหว่างโปรแกรมที่ผู้วิจัยได้ออกแบบและสร้างขึ้นเองกับโปรแกรมที่ผู้วิจัยอื่นคือ กัญญารัตน์และสายฝนออกแบบและสร้างขึ้น ซึ่งปัญหาที่ใช้ทดสอบนั้นเป็นปัญหาที่กำหนดขึ้นเองจำนวน 2 ปัญหา โดยจะทำการเปรียบเทียบในแง่ของเวลาที่เครนทำงานเสร็จสิ้นและเวลาในการประมวลผล

ตารางที่ 4.24 แสดงการเปรียบเทียบระหว่างผลลัพธ์ของเวลาที่เครนทำการงานเสร็จสิ้นของโปรแกรมที่ผู้วิจัยออกแบบเองกับโปรแกรมที่มีผู้วิจัยอื่นสร้างขึ้น

ปัญหาที่	เวลาเฉลี่ยที่เครนทำงานเสร็จสิ้น (นาที)					เวลาที่น้อยที่สุดที่เครนทำงานเสร็จสิ้น (นาที)				
	โปรแกรมของกัญญารัตน์และสายฝน	โปรแกรมของผู้วิจัยในหลักการที่				โปรแกรมของกัญญารัตน์และสายฝน	โปรแกรมของผู้วิจัยในหลักการที่			
		1	2	3	4		1	2	3	4
เมื่อไม่มีการเปลี่ยนแปลงตัวแปร										
13	195.4	192	197.4	191.2	203.3	172	182	187	175	175
14	227.6	219.5	226.2	217.2	233	211	197	215	202	217
เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1										
13	204.4	189.9	197.8	195.4	201.3	191	182	190	181	181
14	236.1	220.6	223.6	220.9	226.8	222	212	207	212	206
เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2										
13	192.8	179.7	180.2	172.1	190.2	191	171	160	158	171
14	225.3	203.8	211	198.8	216.1	203	195	201	175	197

จากตารางที่ 4.24 เป็นการเปรียบเทียบค่า Makespan ระหว่างโปรแกรมของกัญญารัตน์และสายฝน กับโปรแกรมของผู้วิจัยเองในหลักการทำ Neighbourhood search ต่างๆ เมื่อไม่มีการเปลี่ยนแปลงตัวแปร มีการเปลี่ยนแปลงตัวแปรครั้งที่ 1 และมีการเปลี่ยนแปลงตัวแปรครั้งที่ 2 ของปัญหาที่ได้มีการกำหนดขึ้นเอง จะเห็นได้ว่า ค่า Makespan มีความแตกต่างกันค่อนข้างมาก และส่วนใหญ่แล้ว ค่า Makespan ของหลักการสลับที่ของผู้วิจัยจะมีค่าที่น้อยที่สุด ในขณะเดียวกัน เมื่อมีการเปลี่ยนแปลงตัวแปรครั้งที่ 2 ก็จะส่งผลให้ ค่า Makespan น้อยลงเช่นเดียวกัน

ตารางที่ 4.25 แสดงการเปรียบเทียบระหว่างผลลัพธ์ของเวลาที่ใช้ในการประมวลผลของโปรแกรมที่ผู้วิจัยออกแบบเองกับโปรแกรมที่มีผู้วิจัยอื่นสร้างขึ้น

ปัญหาที่	เวลาเฉลี่ยที่ใช้ในการประมวลผล (วินาที)					เวลาที่น้อยที่ใช้ในการประมวลผล (วินาที)				
	โปรแกรมของกัญญารัตน์และสายฝน	โปรแกรมของผู้วิจัยในหลักการที่				โปรแกรมของกัญญารัตน์และสายฝน	โปรแกรมของผู้วิจัยในหลักการที่			
		1	2	3	4		1	2	3	4
เมื่อไม่มีการเปลี่ยนแปลงตัวแปร										
13	399.6	615.8	641.5	647	323.2	370	604	600	612	317
14	752.5	1264.2	1248	1241.4	630.9	735	1261	1221	1231	628
เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1										
13	388.8	641.1	636.8	633.8	329.2	376	628	622	615	312
14	750.5	1257.8	1256.2	1250.7	621	727	1253	1241	1221	610
เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2										
13	388.7	634.3	630	628.4	325.6	372	621	619	616	318
14	760.1	1254.9	1241.2	1251.1	629.7	738	1248	1215	1241	612

จากตารางที่ 4.25 เป็นการเปรียบเทียบเวลาที่ใช้ในการประมวลผล ระหว่างโปรแกรมของกัญญารัตน์และสายฝน กับโปรแกรมของผู้วิจัยเองในหลักการทำ Neighbourhood search ต่างๆ เมื่อไม่มีการเปลี่ยนแปลงตัวแปร มีการเปลี่ยนแปลงตัวแปรครั้งที่ 1 และมีการเปลี่ยนแปลงตัวแปรครั้งที่ 2 ของปัญหาที่ได้มีการกำหนดขึ้นเอง จะเห็นได้ว่า ค่า Makespan มีความแตกต่างกันค่อนข้างมาก และส่วนใหญ่แล้ว ค่า Makespan ของหลักการสลับที่ของผู้วิจัยจะมีค่าที่น้อยที่สุด ในขณะเดียวกัน เมื่อมีการเปลี่ยนแปลงตัวแปรครั้งที่ 2 ก็จะส่งผลให้ ค่า Makespan น้อยลงเช่นเดียวกัน

ต่อไปจะเป็นการนำตารางที่ 4.24 และ 4.25 มาเปรียบเทียบเป็นเปอร์เซ็นต์ ระหว่างโปรแกรมของกัญญารัตน์และสายฝนกับโปรแกรมที่ผู้วิจัยสร้างขึ้นในหลักการต่างๆ ดังแสดงได้ดังต่อไปนี้

ตารางที่ 4.26 แสดงการเปรียบเทียบระหว่างผลลัพธ์ของเวลาที่เครนทำการงานเสร็จสิ้นของโปรแกรมที่ผู้วิจัยออกแบบเองกับโปรแกรมที่มีผู้วิจัยอื่นสร้างขึ้นเป็นเปอร์เซ็นต์

ปัญหาที่	เวลาเฉลี่ยที่เครนทำงานเสร็จสิ้น(%) ผลต่างของโปรแกรมของกัญญารัตน์และสายฝนเทียบกับของผู้วิจัยในหลักการที่				เวลาที่น้อยที่สุดที่เครนทำงานเสร็จสิ้น(%) ผลต่างของโปรแกรมของกัญญารัตน์และสายฝนเทียบกับของผู้วิจัยในหลักการที่			
	1	2	3	4	1	2	3	4
เมื่อไม่มีการเปลี่ยนแปลงตัวแปร								
13	-1.7	1	-2.1	4	5.8	8.7	1.7	1.7
14	-3.6	-0.6	-4.6	2.4	-6.6	1.9	-4.3	2.8
เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1								
13	-7.1	-3.2	-4.4	-1.5	-4.7	-0.5	-5.2	-5.2
14	-6.6	-5.3	-6.4	-3.9	-4.5	-6.8	-4.5	-7.2
เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2								
13	-6.8	-6.5	-10.7	-1.3	-10.5	-16.2	-17.3	-10.5
14	-9.5	-6.3	-11.8	-4.1	-3.9	-1	-13.8	-3

หมายเหตุ : ค่าเปอร์เซ็นต์ที่เป็น ลบ แสดงให้เห็นว่าโปรแกรมของผู้วิจัยเองมีค่า Makespan ที่น้อยกว่าโปรแกรมของกัญญารัตน์และสายฝน ในขณะที่ค่าเปอร์เซ็นต์ที่เป็นบวก แสดงให้เห็นว่าโปรแกรมของผู้วิจัยเองมีค่า Makespan ที่มากกว่าโปรแกรมของกัญญารัตน์และสายฝน

จากตารางที่ 4.26 เป็นการนำการเปรียบเทียบจากตารางที่ 4.24 มาเปรียบเทียบเป็นเปอร์เซ็นต์ของผลต่างของค่า Makespan ระหว่างโปรแกรมของกัญญารัตน์และสายฝนและโปรแกรมของผู้วิจัยเอง จะเห็นได้ว่าความแตกต่างเป็นเปอร์เซ็นต์ระหว่างระหว่างโปรแกรมของกัญญารัตน์และสายฝนและโปรแกรมของผู้วิจัยในหลักการที่ 3 หลักการสลับที่จะให้ผลความแตกต่างมากที่สุดและเนื่องจากมีค่าเป็นลบ นั่นแสดงว่า หลักการที่ 3 หลักการสลับที่นั้น ให้ค่า Makespan ที่ดีที่สุด และให้ค่าดีกว่าโปรแกรมของกัญญารัตน์และสายฝนนั่นเอง

ตารางที่ 4.27 แสดงการเปรียบเทียบระหว่างผลลัพธ์ของเวลาในการประมวลผลของโปรแกรมที่ผู้วิจัยออกแบบเองกับโปรแกรมที่มีผู้วิจัยอื่นสร้างขึ้นเป็นเปอร์เซ็นต์

ปัญหาที่	เวลาเฉลี่ยในการประมวลผล(%) ผลต่างของโปรแกรมของกัญญารัตน์และสายฝนเทียบกับของผู้วิจัยในหลักการที่				เวลาที่น้อยที่สุดในการประมวลผล(%) ผลต่างของโปรแกรมของกัญญารัตน์และสายฝนเทียบกับของผู้วิจัยในหลักการที่			
	1	2	3	4	1	2	3	4
เมื่อไม่มีการเปลี่ยนแปลงตัวแปร								
13	54	60.5	61.8	-19.2	63.2	62.2	65.4	-14.3
14	67.9	65.7	64.8	-16.2	71.6	66.1	67.5	-14.6
เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 1								
13	64.8	63.8	63	-15.4	67	65.4	63.6	-17
14	67.7	67.5	66.8	-17.2	72.4	70.7	68	-16
เมื่อมีการเปลี่ยนแปลงตัวแปร ครั้งที่ 2								
13	63	62	61.4	-16.2	67	66.4	65.6	-14.5
14	65	63.3	64.6	-17.1	69.1	64.6	68.2	-17.1

หมายเหตุ : ค่าเปอร์เซ็นต์ที่เป็น ลบ แสดงให้เห็นว่าโปรแกรมของผู้วิจัยเองมีค่าเวลาในการประมวลผลที่น้อยกว่าโปรแกรมของกัญญารัตน์และสายฝน ในขณะที่ค่าเปอร์เซ็นต์ที่เป็น บวก แสดงให้เห็นว่าโปรแกรมของผู้วิจัยเองมีค่าเวลาในการประมวลผลที่มากกว่าโปรแกรมของกัญญารัตน์และสายฝน

จากตารางที่ 4.27 เป็นการนำการเปรียบเทียบจากตารางที่ 4.25 มาเปรียบเทียบเป็นเปอร์เซ็นต์ของผลต่างของค่าเวลาที่ใช้ในการประมวลผล ระหว่างโปรแกรมของกัญญารัตน์และสายฝนและโปรแกรมของผู้วิจัยเอง จะเห็นได้ว่าผลต่างเมื่อเทียบเป็นเปอร์เซ็นต์ระหว่างโปรแกรมของกัญญารัตน์และสายฝนกับโปรแกรมของผู้วิจัยเองในหลักการที่ 1 หลักการเลื่อนตำแหน่ง หลักการที่ 2 หลักการคงที่หนึ่งตำแหน่งและหลักการที่ 3 หลักการสลับที่ จะให้ผลที่แตกต่างกันมาก มีความไล่เลี่ยกัน และให้ผลเป็นค่าบวก นั่นแสดงว่าโปรแกรมของผู้วิจัยเองมีค่าเวลาในการประมวลผลที่มากกว่า ในขณะที่ ผลต่างระหว่างโปรแกรมของกัญญารัตน์และสายฝนกับโปรแกรมของผู้วิจัยเองในหลักการที่ 4 หลักการพิจารณาที่ 2 ตำแหน่ง จะมีความแตกต่างกันไม่มากนัก และมีค่าเป็นลบ นั่นแสดงว่าโปรแกรมของผู้วิจัยเองมีค่าเวลาในการประมวลผลที่น้อยกว่าโปรแกรมของกัญญารัตน์และสายฝน

ซึ่งผลการเปรียบเทียบทั้งหมดที่กล่าวมาจะทำการวิเคราะห์ผลในหัวข้อ 4.4 การวิเคราะห์ผลการทดลองต่อไป

4.4 การวิเคราะห์ผลการทดลอง

ในการวิเคราะห์ผลการทดลองนี้ จะทำการวิเคราะห์และเปรียบเทียบผลการทดลองที่ได้จากการประมวลผลจากโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองและจากการเปรียบเทียบผลนั้น ซึ่งจะได้จากข้อ 4.2 และ 4.3 ซึ่งสามารถวิเคราะห์และเปรียบเทียบผลการทดลองได้เป็น 3 ประเด็น คือ การเปรียบเทียบผลจากหลักการในการทำ Neighbourhood search ทั้ง 4

หลักการ ขนาดของปัญหากับค่า Makespan และเวลาในการประมวลผล และการเปรียบเทียบกับงานวิจัยอื่น และการเปรียบเทียบผลเนื่องจากการเปลี่ยนแปลงตัวแปรในการอบอ่อน

4.4.1 การเปรียบเทียบผลจากหลักการในการทำ Neighbourhood search ทั้ง 4 หลักการ

จากหลักการในการทำ Neighbourhood search ทั้ง 4 หลักการ อันได้แก่ หลักการเลื่อนตำแหน่ง หลักการคงที่หนึ่งตำแหน่ง หลักการสลับที่ และหลักการพิจารณาที่ 2 ตำแหน่ง นั้น เมื่อทำการประมวลผลโปรแกรมด้วยปัญหาต่างๆแล้วพบว่า

หลักการสลับที่ จะให้ค่าเวลาที่เครนทำงานเสร็จสิ้น หรือค่า Makespan น้อยที่สุด นั่นคือ เวลาการทำงานที่ดีที่สุด จะเห็นได้จาก ตารางแสดงผลลัพธ์ ตารางแสดงผลการเปรียบเทียบต่างๆ และแผนภูมิต่างๆ ซึ่งสามารถพิจารณาได้อย่างชัดเจนจาก ตารางที่ 4.20 และ 4.26 ที่แสดงผลการเปรียบเทียบเป็นเปอร์เซ็นต์ ซึ่งสังเกตได้ว่า หลักการสลับที่นั้นจะให้ค่าเปอร์เซ็นต์ความแตกต่างที่ดีสูงกว่าหลักการอื่นๆ และรูปที่ 4.10, 4.11, 4.15 และ 4.15 ซึ่งสังเกตได้ว่า หลักการสลับที่นั้นจะใช้เวลาที่เครนทำงานเสร็จสิ้นน้อยกว่าหลักการอื่นๆ

หลักการพิจารณาที่ 2 ตำแหน่ง จะให้ค่าเวลาในการประมวลผลที่น้อยกว่าหลักการอื่นๆ จะเห็นได้จาก ตารางแสดงผลลัพธ์ ตารางแสดงผลการเปรียบเทียบต่างๆ และแผนภูมิต่างๆ ซึ่งสามารถพิจารณาได้อย่างชัดเจนจาก ตารางที่ 4.21, 4.25 และ 4.27 ซึ่งจะแสดงผลการเปรียบเทียบเป็นเวลาในการประมวลผลและเป็นเปอร์เซ็นต์เปรียบเทียบ ซึ่งสังเกตได้ว่า หลักการสลับที่นั้นจะให้ค่าเวลาและเปอร์เซ็นต์ความแตกต่างที่ดีสูงกว่าหลักการอื่นๆ และรูปที่ 4.12, 4.13, 4.16 และ 4.17 ซึ่งสังเกตได้ว่า หลักการพิจารณาที่ 2 ตำแหน่ง จะให้กราฟที่แสดงเวลาน้อยกว่าหลักการอื่นๆเกือบครึ่งหนึ่ง ซึ่งเหตุที่เป็นเช่นนั้น อันเนื่องมาจาก ในหลักการพิจารณาที่ 2 ตำแหน่งนั้น จะมีคำตอบจากการหาคำตอบใหม่ในกระบวนการอบอ่อนจำลองจำนวนน้อยกว่าหลักการอื่น นั่นคือ จะได้คำตอบ n-1 คำตอบ เมื่อ n เป็นจำนวนงาน ในขณะที่อีก 3 หลักการที่เหลือ จะให้คำตอบใหม่จำนวน 2(n-1) คำตอบ และนั่นจะทำให้เวลาในการประมวลผลหาคำตอบของหลักการพิจารณาที่ 2 ตำแหน่ง น้อยกว่าหลักการอื่นๆ

และจากสาเหตุที่คำตอบในการหาคำตอบใหม่ของกระบวนการอบอ่อนจำลองของหลักการพิจารณาที่ 2 ตำแหน่ง น้อยกว่าหลักการอื่นๆ ทำให้ค่า Makespan เฉลี่ยแล้ว มากกว่าหลักการอื่นด้วยเช่นกัน เพราะมีคำตอบให้เลือกน้อยกว่านั่นเอง จะเห็นได้จาก ตารางแสดงผลลัพธ์ ตารางแสดงผลการเปรียบเทียบต่างๆ และแผนภูมิต่างๆ ซึ่งสามารถพิจารณาได้อย่างชัดเจนจาก ตารางที่ 4.20 และ 4.26 ที่แสดงผลการเปรียบเทียบเป็นเปอร์เซ็นต์ และรูปที่ 4.10, 4.11, 4.15 และ 4.15 เช่นเดียวกัน

4.2.2 การเปรียบเทียบผลเนื่องจากการเปลี่ยนแปลงตัวแปรในการอบอ่อน

ในการเปลี่ยนแปลงตัวแปรในการอบอ่อนนั้น มีความเป็นจริงที่ปฏิเสธไม่ได้ว่า หากทำการเปลี่ยนแปลงตัวแปรแล้ว ทำให้จำนวนรอบทั้งหมดในการอบอ่อนมากขึ้น ก็ย่อมจะทำให้มีโอกาสในการที่จะได้ค่า Makespan ที่น้อยที่สุดเพิ่มขึ้นเช่นเดียวกัน แต่ในกรณีนี้เป็นการเปรียบเทียบว่า หากเปลี่ยนแปลงตัวแปรในการอบอ่อนแล้ว จะส่งผลอย่างไร โดยที่จำนวนรอบทั้งหมดในการอบอ่อนยังคงเท่าเดิม จากการทดลอง เมื่อทำการประมวลผลโปรแกรมด้วยปัญหาต่างๆแล้ว จะเห็นได้ว่า

จากการเปลี่ยนแปลงตัวแปรในการอบอ่อนครั้งที่ 1 ซึ่งเป็นการเพิ่มอุณหภูมิเริ่มต้นจาก 100 องศาเซลเซียส เป็น 500 องศาเซลเซียส และลดอัตราการเย็นตัวจาก 0.90 เป็น 0.89 แล้วพบว่าการเปลี่ยนแปลงที่เกิดขึ้นไม่อาจส่งผลให้ค่าของ Makespan และเวลาในการประมวลผลเปลี่ยนแปลงไปเมื่อเทียบกับการไม่เปลี่ยนแปลงตัวแปรในการอบอ่อน จะเห็นได้จาก ตารางแสดงผลลัพธ์ ตารางแสดงผลการเปรียบเทียบต่างๆ และแผนภูมิต่างๆ ซึ่งสามารถพิจารณาได้อย่างชัดเจนจาก ตารางที่ 4.22 และ 4.23 ที่แสดงผลการเปรียบเทียบเป็นเปอร์เซ็นต์ และรูปที่ 4.14-4.17 ที่แสดงแผนภูมิเปรียบเทียบให้เห็นอย่างชัดเจน ซึ่งเหตุที่เป็นเช่นนั้น อันเนื่องมาจาก การเปลี่ยนแปลงที่เกิดขึ้น เป็นการเพียงแค่การเปลี่ยนอุณหภูมิเริ่มต้นและอัตราการเย็นตัว ซึ่งส่งผลต่อค่าอุณหภูมิขณะทำการอบอ่อนจำลอง ซึ่งอาจเป็นเพียงแค่การเพิ่มค่าความน่าจะเป็นในการยอมรับคำตอบที่ด้อยกว่าในกระบวนการอบอ่อนจำลองเท่านั้น

จากการเปลี่ยนแปลงตัวแปรในการอบอ่อนครั้งที่ 2 ซึ่งเป็นการลดอุณหภูมิเริ่มต้นจาก 100 องศาเซลเซียส เป็น 50 องศาเซลเซียส ลดอัตราการเย็นตัวจาก 0.90 เป็น 0.834 และเพิ่มจำนวนรอบในการอบอ่อนจาก 5 รอบ เป็น 9 รอบ แล้วพบว่าการเปลี่ยนแปลงที่เกิดขึ้นไม่ส่งผลต่อค่าเวลาในการประมวลผล แต่จะส่งผลต่อค่า Makespan ที่ได้ ซึ่งทำให้ได้ค่า Makespan ที่น้อยลง เมื่อเทียบกับการไม่เปลี่ยนแปลงตัวแปรในการอบอ่อน ซึ่งสามารถพิจารณาได้อย่างชัดเจนจาก ตารางที่ 4.22 และ 4.23 ที่แสดงผลการเปรียบเทียบเป็นเปอร์เซ็นต์ และรูปที่ 4.14-4.17 ที่แสดงแผนภูมิเปรียบเทียบให้เห็นอย่างชัดเจน ซึ่งเหตุที่เป็นเช่นนั้น อันเนื่องมาจาก การเปลี่ยนแปลงที่เกิดขึ้น เป็นการเพิ่มจำนวนรอบในการหาคำตอบในการหาคำตอบที่อุณหภูมิหนึ่งๆในการอบอ่อนจำลอง ซึ่งเป็นการเพิ่มโอกาสในการที่จะได้คำตอบที่น้อยลง หรือได้ค่า Makespan ที่น้อยลงได้มากขึ้น ในขณะที่จำนวนทั้งหมดในการหาคำตอบ และจำนวนการหาคำตอบใหม่ในกระบวนการอบอ่อนจำลองยังคงเท่าเดิม และนั่นก็ส่งผลให้เวลาในการประมวลผลเท่าเดิมเช่นเดียวกันเมื่อเทียบกับการไม่เปลี่ยนแปลงตัวแปรในการอบอ่อนจำลอง

4.4.3 ขนาดของปัญหากับค่า Makespan และเวลาในการประมวลผล และการเปรียบเทียบกับงานวิจัยอื่น

จากการที่ผู้วิจัยได้ทำการแบ่งขนาดของปัญหาเป็นขนาดเล็ก ขนาดกลางและขนาดใหญ่ตามขนาดของจำนวนงานและจำนวนเครนนั้น เมื่อทำการประมวลผลโปรแกรมแล้วพบว่า

ขนาดของปัญหามีผลโดยตรงต่อเวลาในการประมวลผล ยิ่งขนาดและความซับซ้อนของปัญหามีมากเท่าใด จะต้องยิ่งใช้เวลาในการประมวลผลเพิ่มมากขึ้นเท่านั้น ซึ่งสามารถพิจารณาได้อย่างชัดเจนจาก ตารางที่ 4.2, 4.19 และ 4.25 ที่แสดงผลการเปรียบเทียบเป็นตัวเลข และรูปที่ 4.12, 4.13, 4.16 และ 4.17 ที่แสดงแผนภูมิเปรียบเทียบให้เห็นอย่างชัดเจน ในขณะที่ขนาดของปัญหาก็มีส่วนให้ผู้ใช้ต้องพิจารณาถึงว่า ปัญหาขนาดใด จะต้องกำหนดตัวแปรในการอบอ่อนมากน้อยเพียงใดจึงจะได้ค่า Makespan ที่น้อยที่สุดตามที่ต้องการ และเวลาที่ใช้ในการประมวลผลน้อยที่สุด เช่น ปัญหาขนาดใหญ่ หรือปัญหามีความซับซ้อนมาก จะต้องกำหนดให้มีจำนวนรอบทั้งหมดในกระบวนการอบอ่อนจำลองมาก ใช้หลักการสลับที่ในการทำ Neighbourhood search หรือปัญหามีขนาดเล็ก ไม่ซับซ้อน ก็สามารถกำหนดจำนวนรอบทั้งหมดในการอบอ่อนน้อยได้ เพื่อประหยัดเวลาในการประมวลผลโปรแกรม

เมื่อมีการนำเอาผลลัพธ์จากการประมวลไปเทียบกับงานวิจัยอื่น อันได้แก่ แบบจำลองของ Zhu และ Lim แบบจำลองของเอราวิลและคณะ และโปรแกรมของกัญญารัตน์และสายฝน จะเห็นได้ว่า โปรแกรมของผู้วิจัยสามารถประมวลผลได้ค่า Makespan ที่ใช่หรือใกล้เคียงค่า Makespan ที่น้อยที่สุดที่เป็นไปได้ตามแบบจำลองเชิงคณิตศาสตร์ของ Zhu และ Lim แบบจำลองเชิงคณิตศาสตร์ของเอราวิลและคณะ ในขณะที่ใช้เวลาในการประมวลที่น้อยกว่ามาก เมื่อปัญหาใหญ่ขึ้นและมีความซับซ้อนขึ้น ซึ่งบางปัญหาแล้วแบบจำลองทั้งสอง ยังไม่สามารถหาค่าคำตอบที่เป็น Optimum ได้เลย ดังจะเห็นได้ชัดเจนจากตารางที่ 4.19-4.21 ซึ่งจะเป็นการเปรียบเทียบทั้งเป็นค่าตัวเลขและเป็นเปอร์เซ็นต์

ในขณะที่เมื่อเปรียบเทียบจากโปรแกรมของกัญญารัตน์และสายฝนแล้วพบว่า โปรแกรมของกัญญารัตน์และสายฝนมีความใกล้เคียงกับโปรแกรมของผู้วิจัยมาก ทั้งผลลัพธ์ของค่า Makespan และเวลาในการประมวลผล เนื่องจากเป็นโปรแกรมการจัดลำดับการทำงานของเครนด้วยวิธีการอบอ่อนจำลองเหมือนกัน ซึ่งจะมีความแตกต่างกันที่โปรแกรมของผู้วิจัยนั้น มีการทำ Neighbourhood search ที่หลากหลายกว่า ทำให้มีผลลัพธ์บางส่วนที่มีความแตกต่างกัน เช่น ผู้วิจัยใช้หลักการสลับที่ในการประมวลผล ทำให้ได้ค่า Makespan ที่น้อยกว่าผลลัพธ์จากโปรแกรมของกัญญารัตน์และสายฝน หรือเมื่อผู้วิจัยใช้หลักการพิจารณาที่ 2 ตำแหน่งในการประมวลผล ทำให้ใช้เวลาในการประมวลผลน้อยกว่า แต่ให้ให้ค่า Makespan ที่มากกว่าผลจากโปรแกรมของกัญญารัตน์และสายฝน เป็นต้น ซึ่งสามารถสังเกตได้อย่างชัดเจนในตารางที่ 4.19, 4.21 และ 4.24-4.27 ซึ่งแสดงการเปรียบเทียบทั้งเป็นค่าตัวเลขและเป็นเปอร์เซ็นต์ และรูปที่ 4.14-4.17 ที่แสดงถึงแผนภูมิการเปรียบเทียบอย่างชัดเจน

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

5.1.1 จากการวิจัยในครั้งนี้ ทำให้เกิดโปรแกรมสำเร็จรูปที่ช่วยในการจัดลำดับการทำงานของเครน โดยใช้วิธีการอบอ่อนจำลอง ซึ่งสามารถนำไปใช้ในการจัดลำดับการทำงานของเครนในการขนถ่ายสินค้าในอุตสาหกรรมท่าเรือได้

5.1.2 โปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองจะทำการประมวลผลให้ได้คำตอบในการจัดลำดับการทำงานของเครนในเวลาที่รวดเร็ว และได้คำตอบที่น่าพึงพอใจ แม้คำตอบที่ได้นั้นจะไม่ใช่คำตอบที่ดีที่สุดก็ตาม

5.1.3 จากการทดสอบประสิทธิภาพของโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองนี้ ด้วยปัญหาต่างๆพบว่า

5.1.3.1 ความหลากหลายของการทำ Neighbourhood search มีผลต่อค่า Makespan หรือเวลาที่เครนทำงานเสร็จสิ้นน้อยที่สุด และเวลาในการประมวลผล ซึ่งหลักการสลับที่ในการทำ Neighbourhood search จะทำให้ได้ค่า Makespan ที่น้อยที่สุด และหลักการพิจารณาที่ 2 ตำแหน่ง จะทำให้ใช้เวลาในการประมวลผลน้อยที่สุด

5.1.3.2 เมื่อนำโปรแกรมของผู้วิจัยไปเปรียบเทียบกับโปรแกรมหรือแบบจำลองของผู้วิจัยอื่นพบว่า ค่า Makespan ที่ได้จากผลลัพธ์จากโปรแกรมของผู้วิจัย มีความใกล้เคียงค่า Makespan ที่น้อยที่สุดจากแบบจำลองกำหนดการเชิงคณิตศาสตร์ และสามารถใช้เวลาในการประมวลผลที่รวดเร็วยิ่งกว่า

5.1.3.3 การเปลี่ยนแปลงตัวแปรในการอบอ่อนจำลองจะมีบทบาทสำคัญ ในการที่จะได้คำตอบที่มีค่า Makespan ที่ดีหรือไม่ ยิ่งจำนวนรอบในการหาคำตอบแต่ละอุณหภูมิในกระบวนการอบอ่อนจำลองมีมากเท่าใด ก็ยิ่งมีโอกาสที่จะคำตอบในบริเวณใกล้เคียงที่มีค่าที่ดีกว่าได้เช่นกัน

5.2 ข้อเสนอแนะ

5.2.1 ในการดำเนินการทำ Neighbourhood search นั้น อาจจะมีความหลากหลายใดที่ให้คำตอบที่ดีกว่า สะดวกกว่า และรวดเร็วกว่าในการประมวลผล ซึ่งนั่นอาจจะเป็นเหตุให้จะต้องมีการศึกษาในการทำ Neighbourhood search หลายๆแบบต่อไป

5.2.2 ในการเปลี่ยนแปลงตัวแปรในการอบอ่อน อาจจะต้องมีการศึกษาในหลายๆรูปแบบ ว่าการเปลี่ยนแปลงตัวแปรใดมีผลอย่างไร และสามารถเลือกใช้กับปัญหาขนาดใดจึงจะมีความเหมาะสมเพื่อให้ได้คำตอบจากการจัดลำดับการทำงานที่ดีที่สุดในเวลาอันรวดเร็ว

เอกสารอ้างอิง

กรุง สินอภิรมย์สราญ. **ขั้นตอนวิธีประเภทศึกษาสำนึก Heuristic.** สืบค้นเมื่อ 8 สิงหาคม 2553, จาก http://pioneer.netserv.chula.ac.th /~skrung/2301365/Lecture012.html

กัญญารัตน์ คุ้มคลอง และ สายฝน ช่างเหลา. (2550). **การจัดลำดับการทำงานของเครนโดยใช้วิธีอบอ่อนจำลอง.** วิทยานิพนธ์ วศ.บ., มหาวิทยาลัยนเรศวร, พิษณุโลก.

กิตินันท์ พลสวัสดิ์. (2552). **Excel VBA สำหรับคนทำงาน.** กรุงเทพฯ: บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด.

มารีนเนอร์ไทยดอทคอม. **การขนส่งทางเรือด้วยระบบตู้คอนเทนเนอร์.** สืบค้นเมื่อ 8 สิงหาคม 2553, จาก http://www.marinerthai.com /sara/view.php?No=1006

เวชยันต์ สังข์จุ้ย. (2551). ล้วงลึก การเขียนโปรแกรมด้วย Excel. กรุงเทพฯ: บริษัท วี.พริ้น(1991) จำกัด.

สมชัย แสงทองสกุลเลิศ. (2544). **อัลกอริทึมสำหรับการจัดสรรโมบายล์เอเจนต์ในระบบการสืบค้นข้อมูล.** วิทยานิพนธ์ วศ.ม., จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพฯ.

เอราวิล ถาวร, หนึ่งฤทัย ทัพใหญ่ และ กนกพร อารยิกานนท์. (2550). **การแก้ปัญหาการจัดลำดับการทำงานของเครน.** วิทยานิพนธ์วศ.บ., มหาวิทยาลัยนเรศวร, พิษณุโลก.

Christian Bierwirth, Frank Meisel. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. In **European Journal of Operational Research 202** (202, 615–627).

Max Dama. (July 24, 2008). **Trading Optimization: Simulated Annealing.** Retrieved August 22, 2010, from http://www.maxdama.com/2008/07/trading-optimization-simulated.html

ภาคผนวก ก

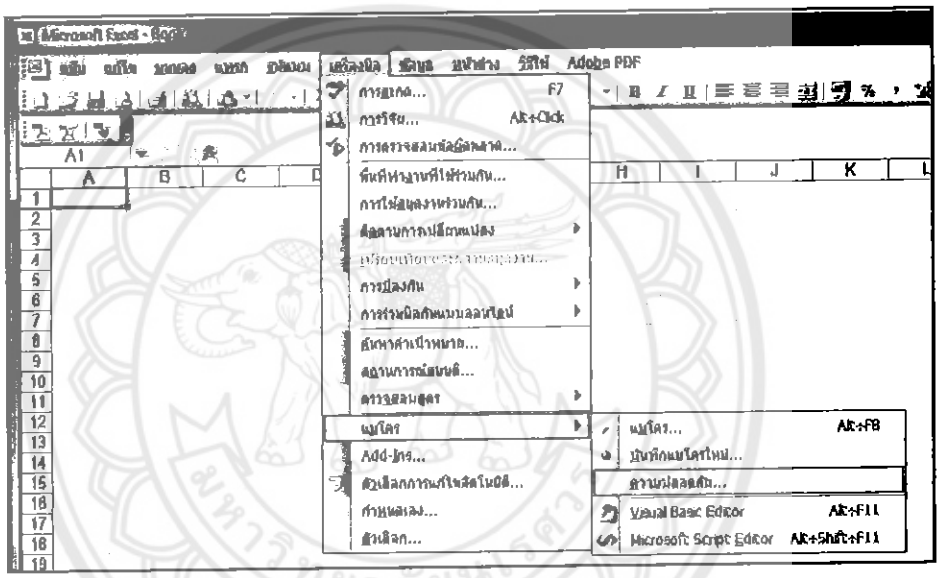
คู่มือโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

จากที่ทราบมาแล้วในบทที่ 4 ในหัวข้อ 4.1 โปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง ว่าในโปรแกรมมีส่วนประกอบอะไรบ้าง ซึ่งต่อไปนี้จะขอแนะนำโปรแกรมการทำงาน ดังต่อไปนี้

1. การเปิดใช้งานโปรแกรม

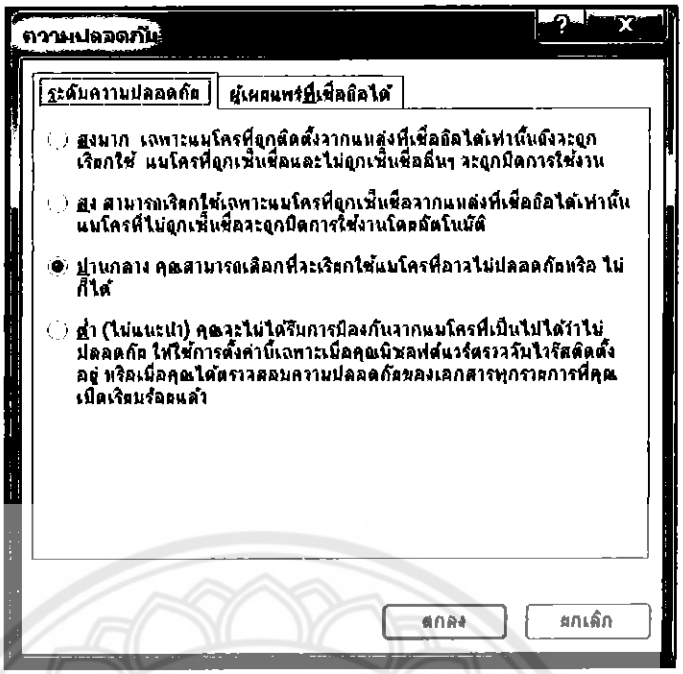
ก่อนที่จะเปิดโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองนั้น จะต้องทำการตั้งค่าแมโครใน Microsoft Excel ก่อน ซึ่งสามารถทำได้ดังนี้

1.1 เริ่มจากการเปิด Microsoft Excel ขึ้นมา ทำการตั้งค่าความปลอดภัย โดยเลือกที่แทบเครื่องมือ เลือกแมโคร และเลือกความปลอดภัย ดังรูปที่ ก.1



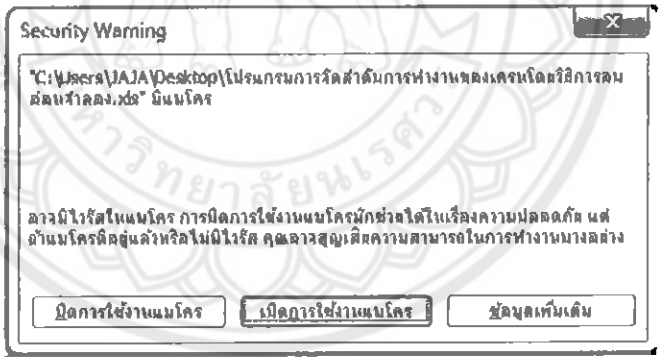
รูปที่ ก.1 แสดงขั้นตอนการตั้งค่าความปลอดภัยแมโคร

จากนั้นทำการเลือกทำการเลือกระดับความปลอดภัยที่ระดับปานกลาง แล้วคลิกปุ่ม ตกลง ดังรูปที่ ก.2 แล้วทำการปิด File Microsoft Excel นี้ก่อน



รูปที่ ก.2 แสดงหน้าต่างการตั้งระดับความปลอดภัยของแมโคร

1.2 เปิด File โปรแกรมชื่อ SA Crane Scheduling นี้ขึ้นมา จะปรากฏหน้าต่าง Security Warning เช่นนี้ขึ้นมา ให้เราเลือก ที่ปุ่ม เปิดการใช้งานแมโคร ดังรูปที่ ก.3



รูปที่ ก.3 แสดงหน้าต่างเพื่อเปิดการใช้แมโคร

2. การเข้าสู่โปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

จากที่ทราบมาแล้วว่าโปรแกรมมีการทำงานใน 4 ส่วน คือ ส่วนต้อนรับเข้าสู่โปรแกรม ส่วนรับข้อมูล ส่วนยืนยันและแก้ไขข้อมูล และส่วนประมวลผล ซึ่งสามารถแบ่งได้เป็น 6 ขั้นตอน ดังจะกล่าวได้ดังต่อไปนี้

2.1 หน้าต้อนรับเข้าสู่โปรแกรม

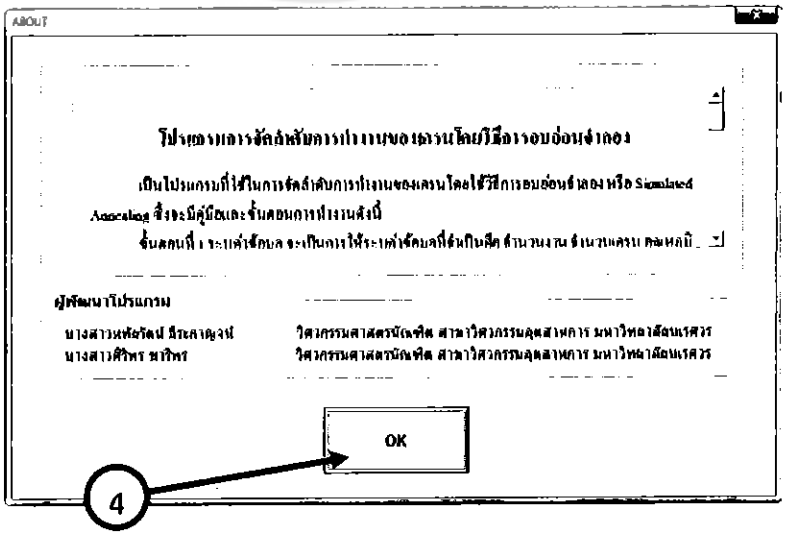
จะเป็นหน้าแรกของโปรแกรมซึ่งเมื่อเข้าโปรแกรมมาแล้ว หน้าต่างนี้จะแสดงมาก่อนทันทีโดยอัตโนมัติ ดังรูปที่ ก.4

รูปที่ ก.4 แสดงหน้าต้อนรับเข้าสู่โปรแกรมของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

เมื่อกดหมายเลข 1 คือกด START จะเป็นการเข้าสู่หน้าถัดไปของโปรแกรม เป็นการเริ่มระบุข้อมูล ดังข้อที่ 2.2 และรูปที่ ก.6

เมื่อกดหมายเลข 2 คือกด ABOUT จะเป็นการให้ความช่วยเหลือ อธิบายโปรแกรม และกล่าวถึงผู้จัดทำโปรแกรม ดังรูปที่ ก.5

เมื่อกดหมายเลข 3 คือกด EXIT จะเป็นการปิดหน้าต่างโปรแกรม ซึ่งในส่วนนี้ผู้ใช้อาจเข้าไปสู่หน้าต่างของ Microsoft Excel หรือจะปิดโปรแกรมได้เลย

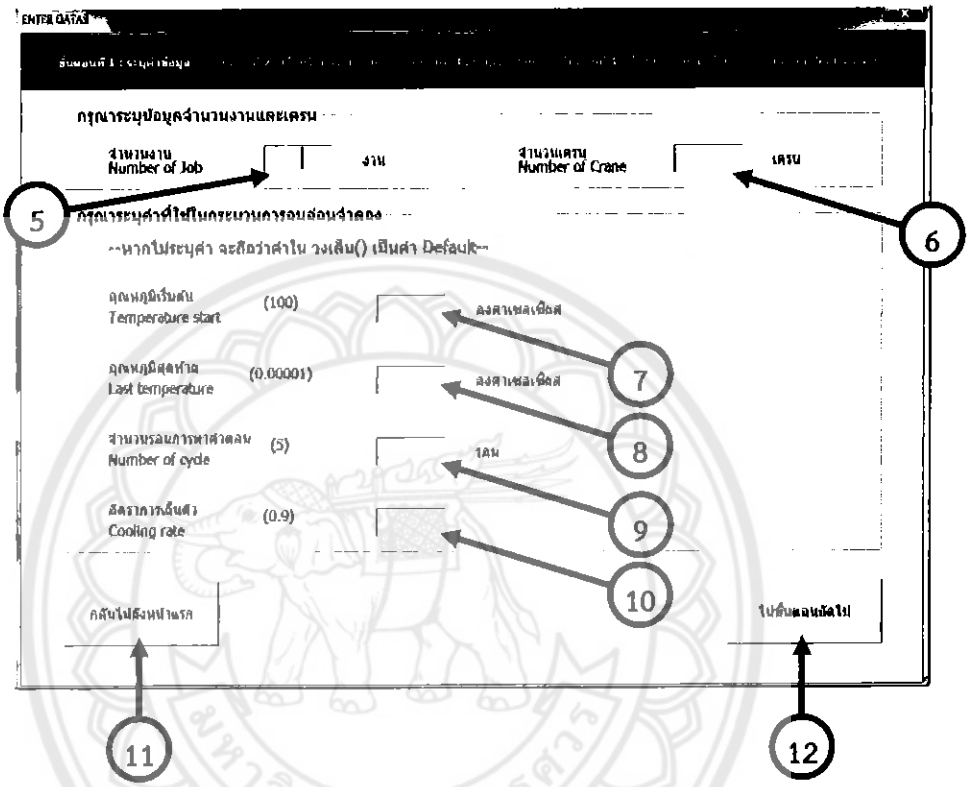


รูปที่ ก.5 แสดงหน้าต่าง ABOUT ที่บอกลักษณะของโปรแกรม

เมื่อกดหมายเลข 4 คือกด OK จะเป็นการปิดหน้าต่างนี้

2.2 ขั้นตอนที่ 1 ระบุค่าข้อมูล

จะเป็นการเริ่มต้นเข้าสู่การจัดลำดับการทำงานของเครน ซึ่งเป็นการให้ผู้ใช้ระบุข้อมูลทั้งหมด ดังรูป ก.6



รูปที่ ก.6 แสดงขั้นตอนที่ 1 ระบุค่าข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

สิ่งที่ผู้ใช้จะต้องทำการระบุค่าข้อมูล คือ

2.2.1 จำนวนงาน ซึ่งสามารถระบุได้ที่หมายเลข 5 จะเป็นการให้ผู้ใช้ระบุจำนวนงานตั้งแต่ 2 งานขึ้นไป ซึ่งสามารถระบุได้เป็นตัวเลขเท่านั้น หากไม่ระบุ จะมีหน้าต่างแจ้งเตือนดังรูป

2.2.2 จำนวนเครน ซึ่งสามารถระบุได้ที่หมายเลข 6 จะเป็นการให้ผู้ใช้ระบุจำนวนเครนที่มี ซึ่งจะต้องมีจำนวนมากกว่าจำนวนงาน หากไม่ระบุ จะมีหน้าต่างแจ้งเตือนดังรูป ก.7

2.2.3 อุณหภูมิเริ่มต้นของการอบอ่อน ซึ่งสามารถระบุได้ที่หมายเลข 7 จะเป็นการให้ผู้ใช้ระบุอุณหภูมิเริ่มต้นตามที่ต้องการ ซึ่งอาจต้องไม่ระบุก็ได้ โปรแกรมจะถือว่าหากไม่ระบุข้อมูลนี้ จะใช้ค่า Default ตามที่วงเล็บไว้เป็นค่าข้อมูล ในที่นี้คือ 100 องศาเซลเซียส

2.2.4 อุณหภูมิสุดท้ายของการอบอ่อน ซึ่งสามารถระบุได้ที่หมายเลข 8 จะเป็นการให้ผู้ใช้ระบุอุณหภูมิสุดท้ายตามที่ต้องการ ซึ่งจะต้องมีค่าน้อยกว่าอุณหภูมิเริ่มต้น และถ้าหากผู้ใช้ระบุค่าอุณหภูมิสุดท้ายมากกว่าอุณหภูมิเริ่มต้น จะมีหน้าต่างแจ้งเตือนดังรูป ก.7 ซึ่งข้อมูลของอุณหภูมิสุดท้ายนั้นอาจต้องไม่ระบุก็ได้ โปรแกรมจะถือว่าหากไม่ระบุข้อมูลนี้ จะใช้ค่า Default ตามที่วงเล็บไว้เป็นค่าข้อมูล ในที่นี้คือ 0.00001 องศาเซลเซียส

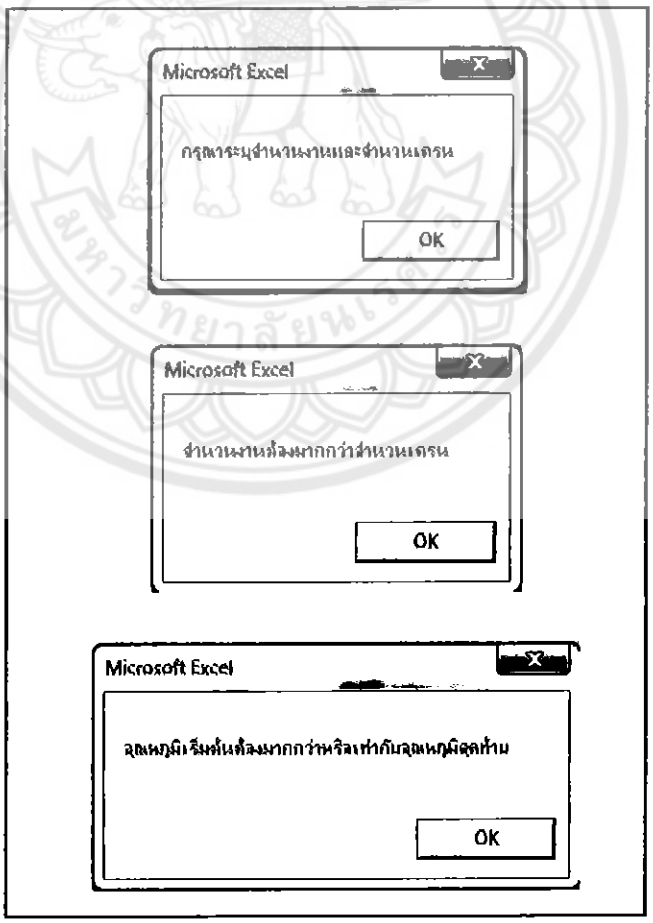
2.2.5 จำนวนรอบในการอบอ่อนแต่ละอุณหภูมิ ซึ่งสามารถระบุได้ที่หมายเลข 9 จะเป็นการให้ผู้ใช้ระบุจำนวนรอบตามที่ต้องการ ซึ่งอาจต้องไม่ระบุก็ได้ โปรแกรมจะถือว่าหากไม่ระบุข้อมูลนี้ จะใช้ค่า Default ตามที่วงเล็บไว้เป็นค่าข้อมูล ในที่นี้คือ 5 รอบ

2.2.6 อัตราการเย็นตัว ซึ่งสามารถระบุได้ที่หมายเลข 10 จะเป็นการให้ผู้ใช้ระบุอัตราการเย็นตัว ตามที่ต้องการ ซึ่งอาจต้องไม่ระบุก็ได้ โปรแกรมจะถือว่าหากไม่ระบุข้อมูลนี้ จะใช้ค่า Default ตามที่วงเล็บไว้เป็นค่าข้อมูล ในที่นี้คือ 0.90

เมื่อระบุค่าข้อมูลเสร็จแล้ว ผู้ใช้จะต้องดำเนินการต่อคือ

เมื่อกดหมายเลข 11 คือกดกลับไปยังหน้าแรก จะเป็นการให้โปรแกรมย้อนกลับไปยังหน้าต้อนรับ ดังข้อ 2.1 และรูปที่ ก.4

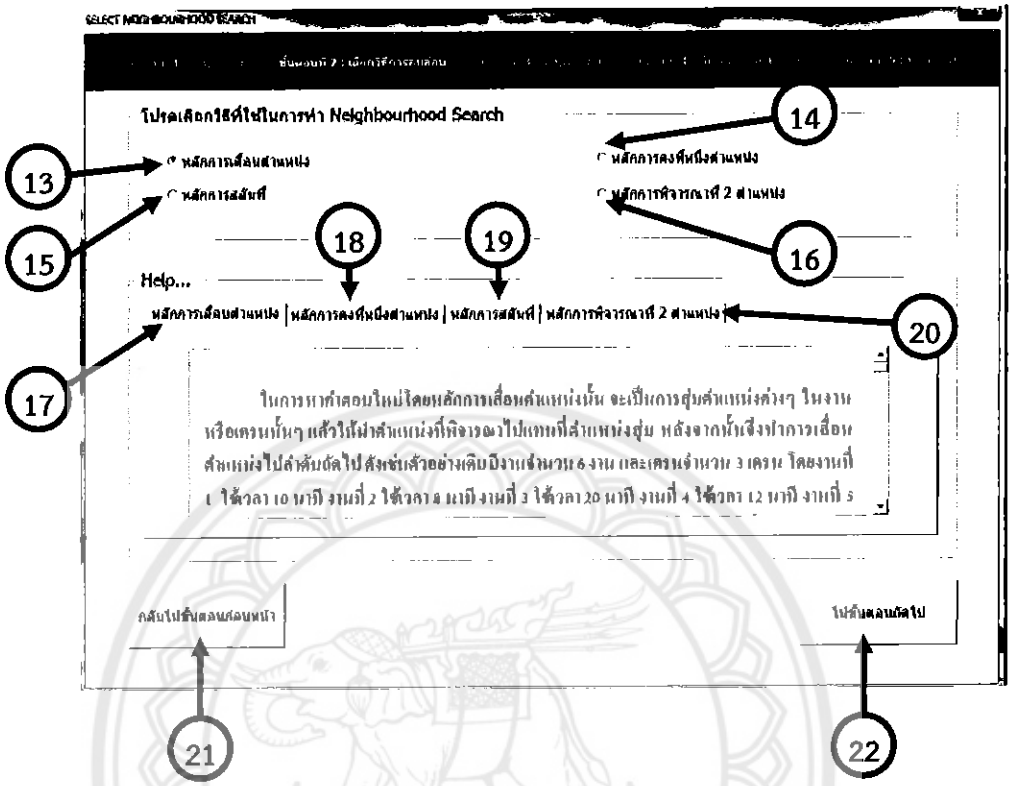
เมื่อกดหมายเลข 12 คือกดไปขั้นตอนถัดไป ซึ่งโปรแกรมจะตรวจสอบความถูกต้องของข้อมูลที่ระบุว่า ตรงตามข้อกำหนดหรือ คือ จะต้องระบุจำนวนงานและจำนวนเครน และจำนวนงานต้องมากกว่าจำนวนเครน รวมทั้งอุณหภูมิเริ่มต้นจะต้องมากกว่าอุณหภูมิสุดท้าย ซึ่งหากไม่ตรงตามข้อกำหนด โปรแกรมจะมีหน้าต่างแจ้งเตือนให้ระบุข้อมูลที่ถูกต้อง ดังรูปที่ ก.8 และหากว่าโปรแกรมตรวจสอบแล้วว่าข้อมูลถูกต้อง โปรแกรมจะบันทึกข้อมูลและจะดำเนินการขั้นถัดไป คือไปยังขั้นตอนที่ 2 เลือกวิธีการอบอ่อน ดังข้อ 2.3 และรูปที่ ก.8



รูปที่ ก.7 แสดงหน้าต่างแจ้งเตือนให้ระบุข้อมูลที่ถูกต้อง

2.3 ขั้นตอนที่ 2 เลือกวิธีการอบอ่อน

จะเป็นการให้ผู้ใช้คลิกเลือกวิธีการทำ Neighbourhood search ซึ่งมีทั้งหมด 4 วิธี ดังรูป



รูปที่ ก.8 แสดงขั้นตอนที่ 2 เลือกวิธีการอบอ่อน ของโปรแกรมจัดลำดับการทำงานของเครน โดยวิธีการอบอ่อนจำลอง

เมื่อกดหมายเลข 13 คือกดเลือกใช้หลักการเลื่อนตำแหน่ง และจะทำให้หน้าต่างช่วยเหลือแสดงคำอธิบายในหลักการ ดังหมายเลข 17

เมื่อกดหมายเลข 14 คือกดเลือกใช้หลักการคงที่หนึ่งตำแหน่ง และจะทำให้หน้าต่างช่วยเหลือ แสดงคำอธิบายในหลักการ ดังหมายเลข 18

เมื่อกดหมายเลข 15 คือกดเลือกใช้หลักการสลับที่ และจะทำให้หน้าต่างช่วยเหลือ แสดงคำอธิบายในหลักการ ดังหมายเลข 19

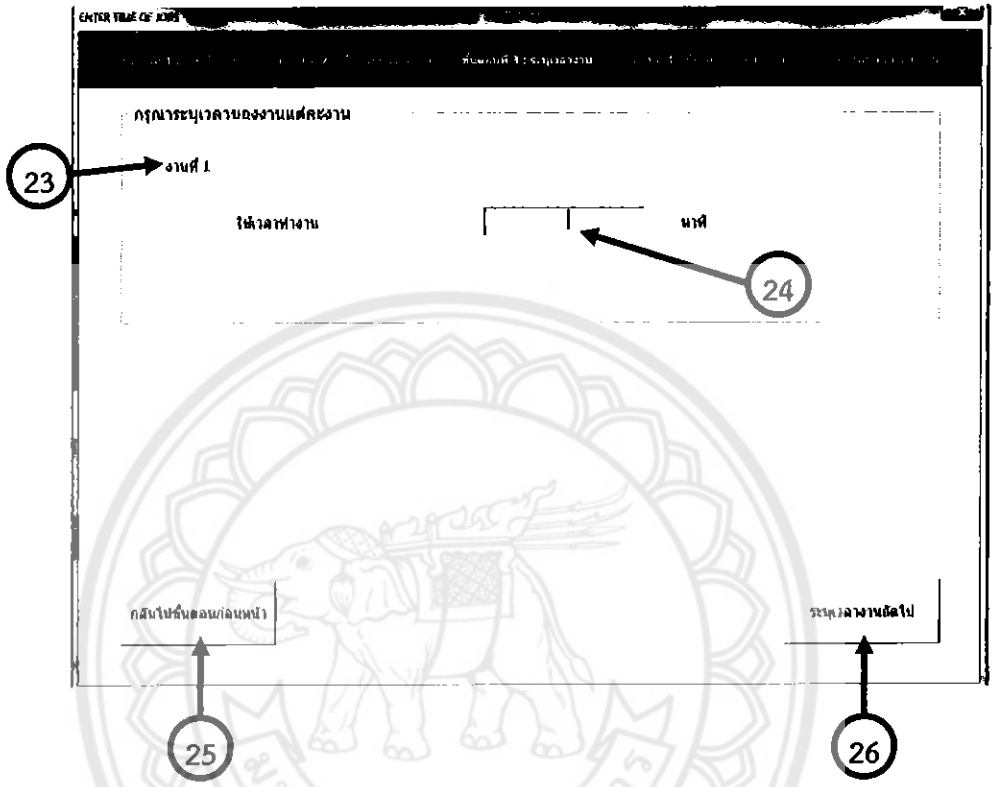
เมื่อกดหมายเลข 16 คือกดเลือกใช้หลักการพิจารณาที่ 2 ตำแหน่งและจะทำให้หน้าต่างช่วยเหลือ แสดงคำอธิบายในหลักการ ดังหมายเลข 20

เมื่อกดหมายเลข 21 คือกดกลับไปยังขั้นตอนก่อนหน้า โปรแกรมจะกลับไปยังขั้นตอนที่ 1 ระบุค่าข้อมูล และจะให้ผู้ใช้ระบุค่าทั้งหมดใหม่ เสมือนเป็นการเริ่มต้นใหม่ ดังข้อที่ 2.2 และรูปที่ ก.6

เมื่อกดหมายเลข 22 คือกดไปขั้นตอนถัดไป โปรแกรมจะบันทึกข้อมูลและจะดำเนินการขั้นถัดไป คือไปยังขั้นตอนที่ 3 ระบุเวลางาน ดังข้อ 2.3 และรูปที่ ก.8

2.4 ขั้นตอนที่ 3 ระบุเวลางาน

จะเป็นการให้ผู้ใช้ระบุค่าเวลาที่ใช้ในการทำงานแต่ละงาน ซึ่งใน 1 หน้าต่าง จะสามารถระบุเวลางานได้เพียง 1 งาน และจะปรากฏหน้าต่างนี้เท่าจำนวนของงานที่ระบุไว้ในขั้นตอนที่ 1 กล่าวคือ กระบวนการระบุเวลางานจะวนไปเรื่อยๆ จนกระทั่งครบตามจำนวนงานที่ระบุไว้ ดังรูป ก.9



รูปที่ ก.9 แสดงขั้นตอนที่ 3 ระบุเวลางาน ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

โดยหมายเลข 23 จะแสดงหมายเลขหรือลำดับของงานที่กำลังระบุข้อมูลอยู่ ผู้ใช้จะระบุเวลาการทำงานของงานนั้นๆที่หมายเลข 24 โดย

2.4.1 เมื่องานที่กำลังระบุเวลาเป็นงานที่ 1

2.4.1.1 จะทำให้ หมายเลข 25 จะแสดงข้อความกลับไปขั้นตอนก่อนหน้า และเมื่อกดหมายเลข 25 คือกดกลับไปยังขั้นตอนก่อนหน้า โปรแกรมจะกลับไปยังขั้นตอนที่ 2 เลือกวิธีการอบอ่อน และจะให้ผู้ใช้เลือกวิธีการทำ Neighbourhood search ใหม่ ดังข้อที่ 2.3 และรูปที่ ก.8

2.4.1.2 จะทำให้ หมายเลข 26 จะแสดงข้อความระบุเวลางานถัดไป และเมื่อกดหมายเลข 25 คือกดระบุเวลางานถัดไป โปรแกรมจะบันทึกข้อมูลเวลางานที่ระบุ และจะดำเนินการถัดไป คือ ให้ระบุเวลางานถัดไป

2.4.2 เมื่องานที่กำลังระบุเวลาเป็นงานสุดท้าย

2.4.2.1 จะทำให้ หมายเลข 25 จะแสดงข้อความกลับไปเวลางานก่อนหน้า และเมื่อกดหมายเลข 25 คือกดกลับไปเวลางานก่อนหน้า โปรแกรมจะกลับไปยังงานก่อนหน้า เพื่อให้ผู้ใช้ได้ทำการระบุหรือแก้ไขเวลางานก่อนหน้า

2.4.2.2 จะทำให้ หมายเลข 26 จะแสดงข้อความไปขั้นตอนถัดไป และเมื่อกดหมายเลข 25 คือกดไปขั้นตอนถัดไป โปรแกรมจะบันทึกข้อมูลเวลางานที่ระบุ และจะดำเนินการถัดไป คือ ไปยังขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ดังข้อที่ 2.5 รูปที่ ก.10

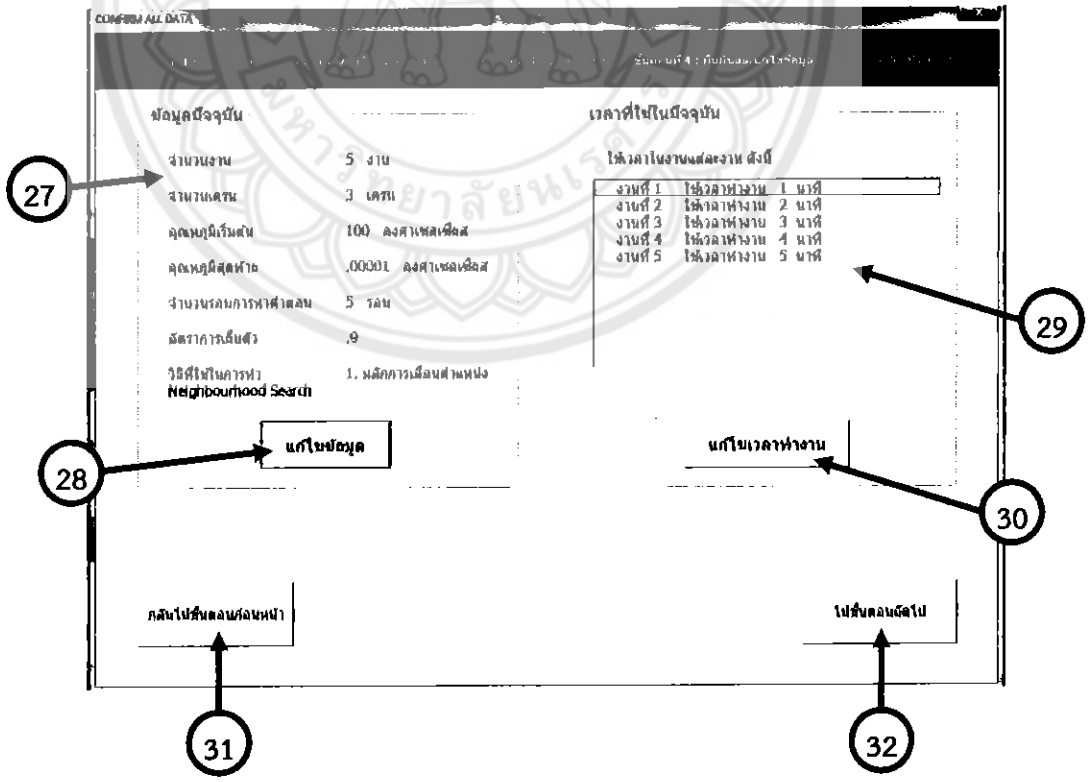
2.4.3 เมื่องานที่กำลังระบุเวลาเป็นงานระหว่างงานแรกและงานสุดท้าย

2.4.3.1 จะทำให้ หมายเลข 25 จะแสดงข้อความกลับไปเวลางานก่อนหน้า และเมื่อกดหมายเลข 25 คือกดกลับไปเวลางานก่อนหน้า โปรแกรมจะกลับไปยังงานก่อนหน้า เพื่อให้ผู้ใช้ได้ทำการระบุหรือแก้ไขเวลางานก่อนหน้า

2.4.3.2 จะทำให้ หมายเลข 26 จะแสดงข้อความระบุเวลางานถัดไป และเมื่อกดหมายเลข 25 คือกดระบุเวลางานถัดไป โปรแกรมจะบันทึกข้อมูลเวลางานที่ระบุ และจะดำเนินการถัดไป คือ ให้ระบุเวลางานถัดไป

2.5 ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล

จะแสดงให้เห็นถึงทุกข้อมูลที่ได้ระบุมาในขั้นตอนที่ 1-3 ซึ่งสามารถคลิกเพื่อแก้ไขข้อมูลได้ดังรูป ก.10



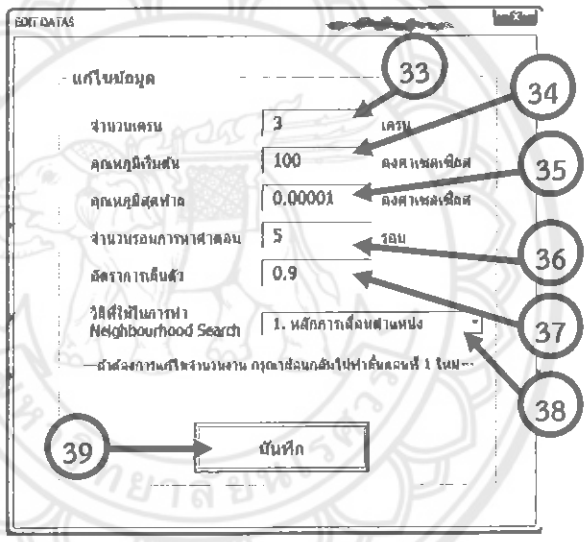
รูปที่ ก.10 แสดงขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

ที่หมายเลข 27 จะเป็นการแสดงข้อมูลทั้งหมดที่ผู้ใช้ระบุ ตั้งแต่ขั้นตอนที่ 1-3 ซึ่งผู้ใช้สามารถแก้ไขข้อมูลได้ โดยกดที่หมายเลข 28 ซึ่งจะแสดงหน้าต่างเพื่อการแก้ไขข้อมูล ดังรูป ก.11

ที่หมายเลข 29 จะเป็นการแสดงเวลาในการทำงานของแต่ละงานที่ผู้ใช้ระบุไปในขั้นตอนที่ 4 ซึ่งผู้ใช้สามารถแก้ไขข้อมูลได้ โดยกดเลือกงานที่ต้องการแก้ไขเวลา และกดที่หมายเลข 30 ซึ่งจะแสดงหน้าต่างเพื่อการแก้ไขข้อมูลของเวลาในการทำงานของงานที่ทำการเลือกไว้ ดังรูป ก.12

เมื่อกดหมายเลข 31 คือกดกลับไปยังขั้นตอนก่อนหน้า โปรแกรมจะกลับไปยังขั้นตอนที่ 3 ระบุเวลางาน และจะให้ผู้ใช้ระบุเวลาของงานทั้งหมดใหม่ ซึ่งเริ่มตั้งแต่งานที่ 1 ดังข้อที่ 2.3 และรูปที่ ก.9

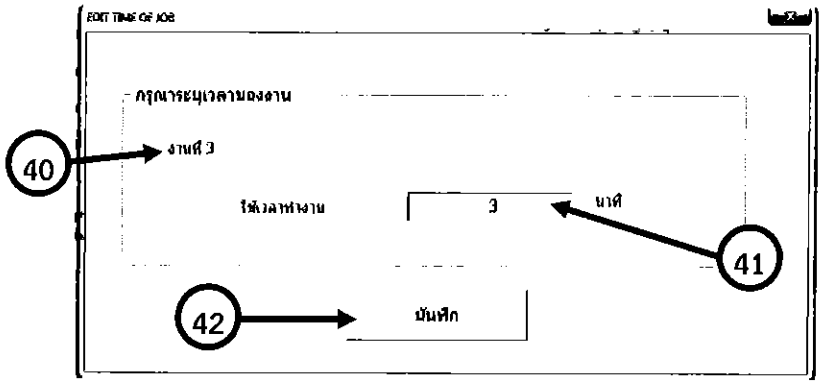
เมื่อกดหมายเลข 32 คือกดไปขั้นตอนถัดไป โปรแกรมจะทำการตรวจสอบข้อมูลว่า ตรงตามข้อกำหนดหรือไม่ หากไม่ตรงตามข้อกำหนดก็จะมีหน้าต่างแจ้งเตือนดังเช่นข้อที่ 2.2 เช่นเดียวกัน และหากว่าข้อมูลที่ระบุและได้รับการยืนยันจากผู้ใช้แล้ว โปรแกรมจะทำการบันทึกข้อมูลและจะดำเนินการขั้นถัดไป คือไปยังขั้นตอนที่ 5 ประมวลผล ดังข้อ 2.6 และรูปที่ ก.13



รูปที่ ก.11 แสดงการแก้ไขข้อมูลใน ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรม จัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

เมื่อเลือกที่จะแก้ไขข้อมูลแล้ว ผู้ใช้สามารถแก้ไขข้อมูลได้คือ จำนวนเครน ที่หมายเลข 33 อุณหภูมิเริ่มต้น ที่หมายเลข 34 อุณหภูมิสุดท้าย ที่หมายเลข 35 จำนวนรอบการหาคำตอบ ที่หมายเลข 36 และวิธีที่ใช้ในการทำ Neighbourhood search ที่หมายเลข 38

เมื่อผู้ใช้แก้ไขข้อมูลแล้ว จะกดที่หมายเลข 39 ซึ่งโปรแกรมจะทำการบันทึกข้อมูล และปิดหน้าต่างนี้ลง



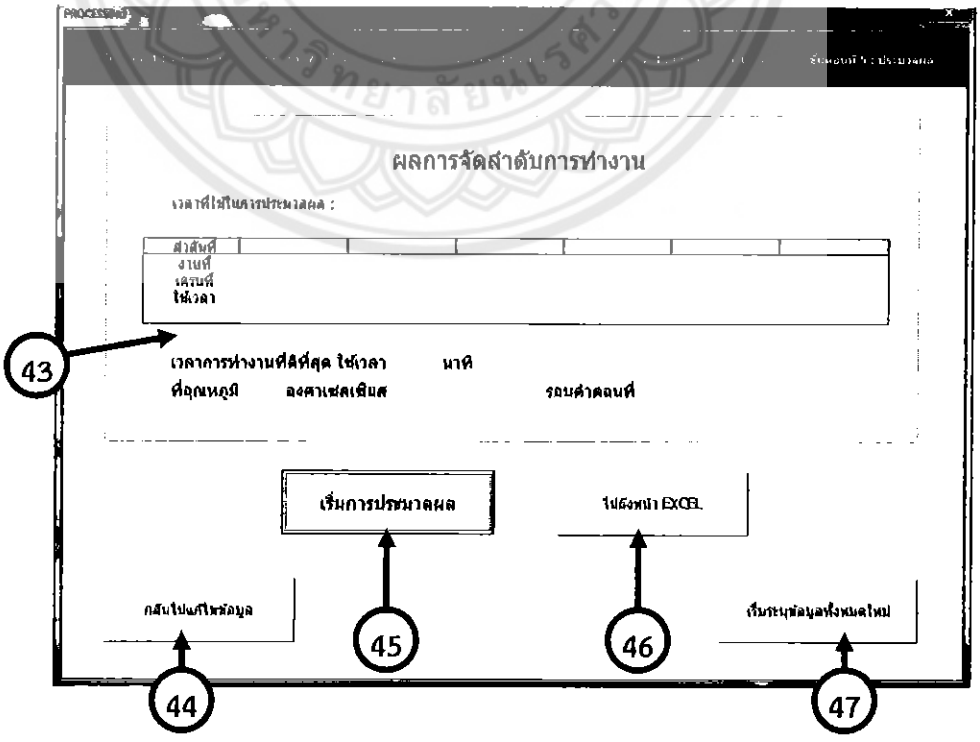
รูปที่ ก.12 แสดงการแก้ไขเวลางานใน ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

เมื่อผู้ใช้ต้องการแก้ไขเวลาของงานใดงานหนึ่งและเลือกแก้ไขงานมา ทำให้หน้าต่างการแก้ไขเวลางานปรากฏ

ซึ่งที่หมายเลข 40 จะแสดงข้อความงานที่ต้องการแก้ไข ซึ่งผู้ใช้สามารถแก้ไขเวลางานได้ที่หมายเลข 41 และทำการบันทึกเวลางานที่หมายเลข 42 เมื่อกดบันทึกแล้ว โปรแกรมจะบันทึกข้อมูลและปิดหน้าต่างนี้ลง

2.6 ขั้นตอนที่ 5 ประมวลผล

จะเป็นการให้ผู้ใช้สั่งประมวลผลการจัดลำดับการทำงานของเครน และแสดงผลให้ผู้ใช้ทราบถึงผลการจัดลำดับการทำงานของเครน ดังรูป ก.13



รูปที่ ก.13 แสดงขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

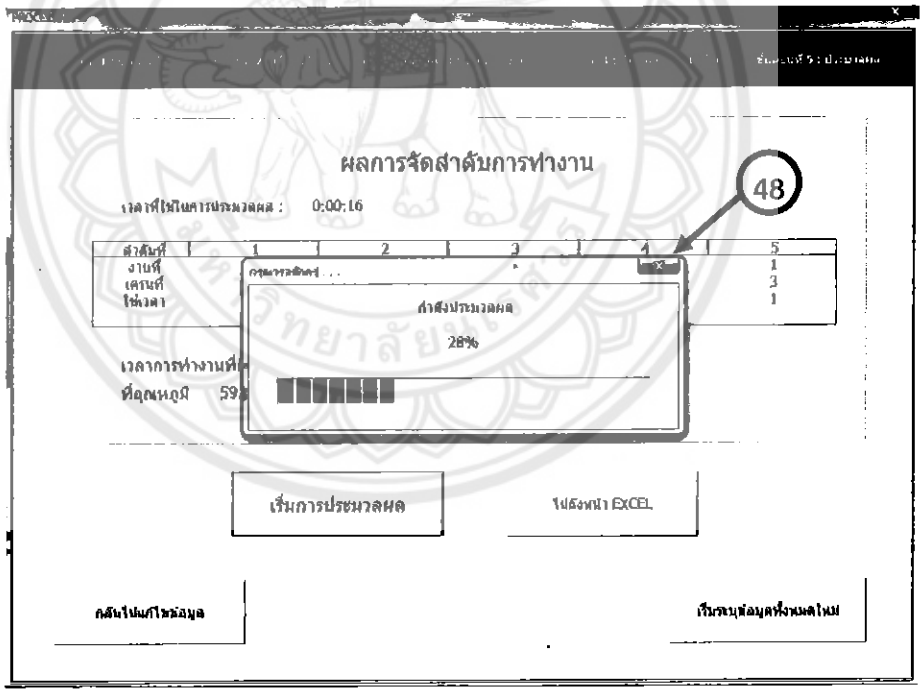
ที่หมายเลข 43 จะแสดงเวลาที่ใช้ในการประมวลผล ลักษณะการจัดลำดับการทำงานของเครนที่ดีที่สุด เวลาการทำงานที่น้อยที่สุด อุณหภูมิและรอบคำตอบขณะที่จัดลำดับได้

เมื่อกดหมายเลข 44 คือกดกลับไปแก้ไขข้อมูล ซึ่งโปรแกรมจะกลับไปยังขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล เพื่อให้ผู้ใช้สามารถแก้ไขข้อมูลได้ตามต้องการ

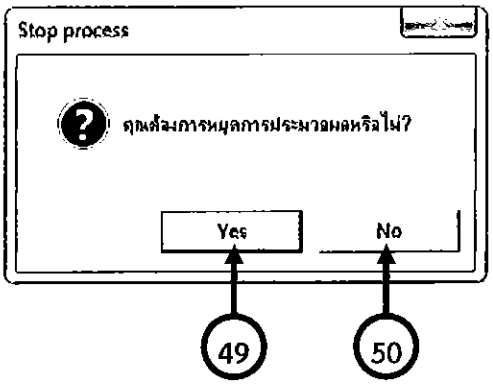
เมื่อกดหมายเลข 45 คือกดเริ่มการประมวลผล ซึ่งโปรแกรมจะเริ่มดำเนินการประมวลผลการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองตามข้อมูลที่ผู้ใช้ระบุไว้ ซึ่งโปรแกรมจะมีแถบความก้าวหน้าแสดงความก้าวหน้าในการประมวลผลเป็นเปอร์เซ็นต์ ดังรูปที่ ก.14 ซึ่งเมื่อประมวลผลเสร็จสิ้น หน้าต่างแถบความก้าวหน้านี้จะปิดลง และโปรแกรมจะทำการ Update ค่าที่แสดงผลจากการจัดลำดับการทำงานใหม่ ดังรูปที่ ก.16

เมื่อกดหมายเลข 46 คือกดไปยังหน้า EXCEL จะทำให้หน้าต่างของโปรแกรมปิดตัวลงและแสดงหน้าต่างของ Microsoft Excel ขึ้น เพื่อเป็นทางเลือกให้ผู้ใช้ได้ทำการประมวลผลโดยตรงทาง Microsoft Excel และเพื่อการดูและสังเกตการณ์ขณะประมวลผลโปรแกรม

เมื่อกดหมายเลข 47 คือกดเริ่มระบุข้อมูลทั้งหมดใหม่ โปรแกรมจะกลับไปยังขั้นตอนที่ 1 ระบุค่าข้อมูล และจะให้ผู้ใช้ระบุค่าทั้งหมดใหม่ เสมือนเป็นการเริ่มต้นใหม่ ดังข้อที่ 2.2 และรูปที่ ก.6



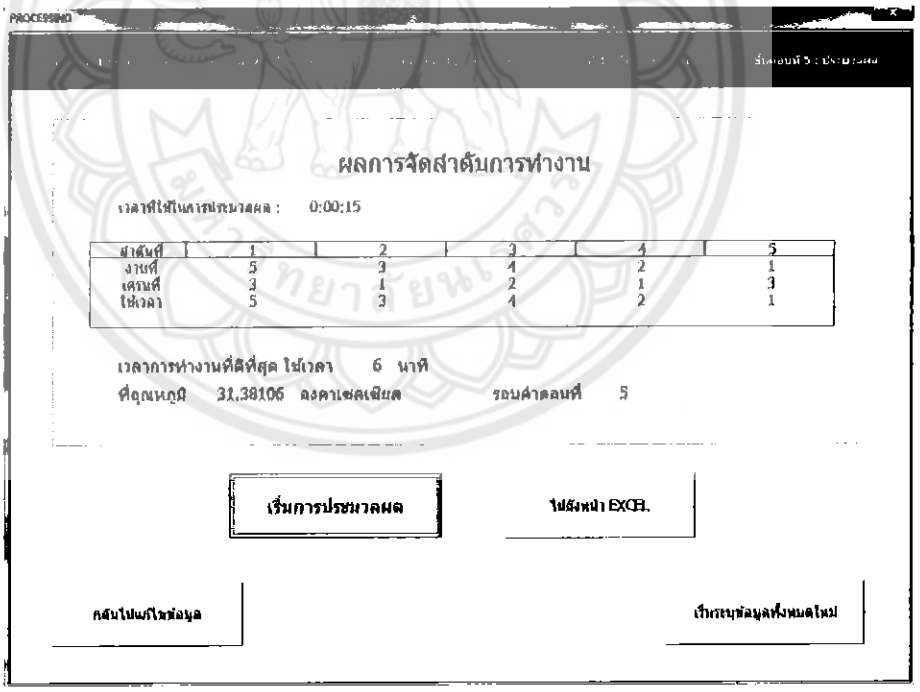
รูปที่ ก.14 แสดงแถบความก้าวหน้าเมื่อกดประมวลผลใน ขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง



รูปที่ ก.15 แสดงหน้าต่างยืนยันการหยุดการประมวลผล

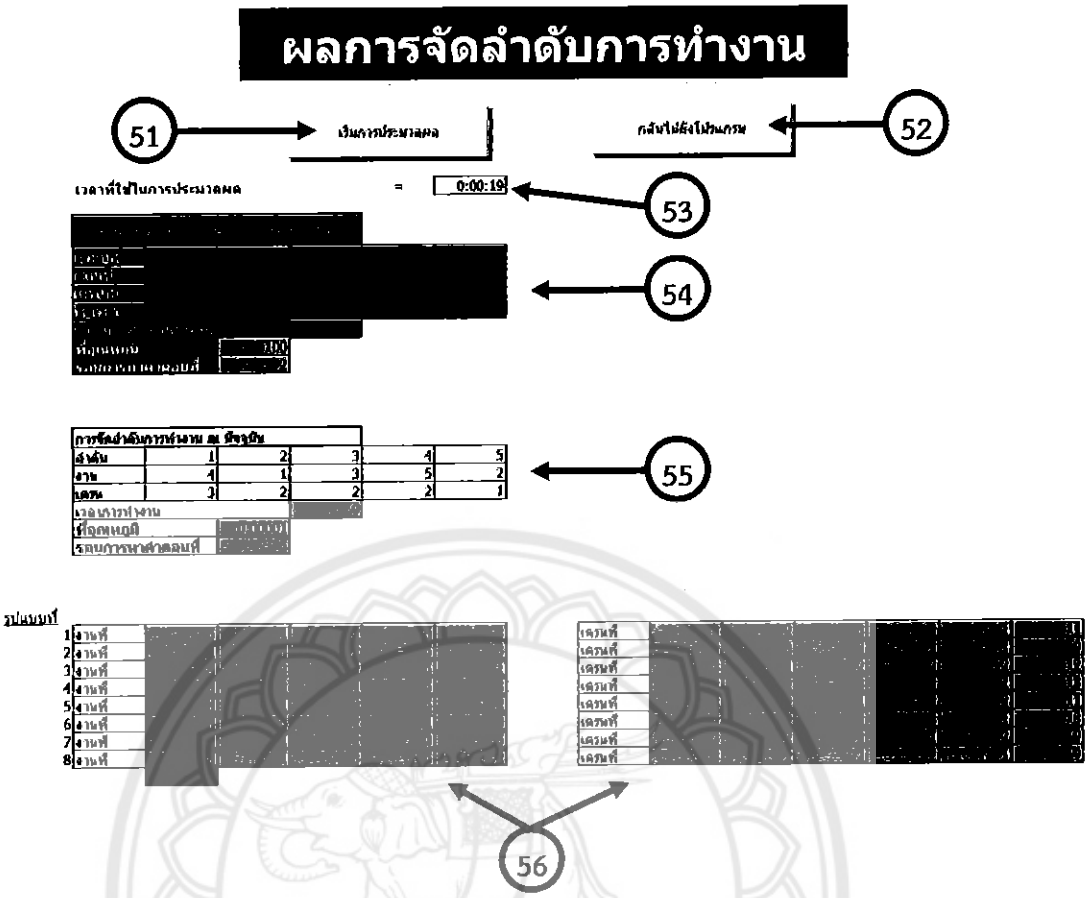
จากรูปที่ ก.14 เป็นหน้าต่างแสดงแถบความก้าวหน้าในการประมวลผล **ซึ่งหากต้องการหยุดการประมวลผล** ให้กดหมายเลข 48 คือกด X **เพื่อทำการหยุดการประมวลผล และจะมีหน้าต่างให้ผู้ใช้ได้**ทำการยืนยันความต้องการในการหยุดการประมวลผล

จากรูปที่ ก.15 หากกดหมายเลข 49 คือกด Yes **จะเป็นการหยุดการประมวลผลโปรแกรมเพียงเท่านั้น** และหน้าต่างแถบความก้าวหน้าจะปิดลง และหากกดหมายเลข 50 **คือกด** No **จะเป็นการปฏิเสธการหยุดการประมวลผล ทำให้โปรแกรมยังคงประมวลผลต่อไป**



รูปที่ ก.16 แสดงผลจากการจัดลำดับการทำงานเมื่อประมวลผลเสร็จสิ้น ในขั้นตอนที่ 5 ประมวลผล ของโปรแกรมจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

ต่อไปจะเป็นการแสดงรูปการประมวลผลทางหน้าต่างของ Microsoft Excel **ดังรูป** ก.17



รูปที่ ก.17 แสดงหน้าต่างของ Microsoft Excel เพื่อใช้ในการประมวลผลจากการจัดลำดับการทำงานจากการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

เมื่อกดหมายเลข 51 คือกดเริ่มการประมวลผล ซึ่งโปรแกรมจะเริ่มดำเนินการประมวลผลการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลองตามข้อมูลที่ผู้ใช้ระบุไว้

เมื่อกดหมายเลข 52 คือกดกลับไปยังโปรแกรม จะปรากฏหน้าต่างของโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง ในขั้นตอนที่ 5 ประมวลผล ดังรูปที่ ก.16 เพื่อให้ผู้ใช้ได้เลือกทำการประมวลผลทางโปรแกรม แก้ไขข้อมูล หรือเริ่มระบุข้อมูลใหม่

ในการแสดงผลบนหน้าต่างของ Microsoft Excel นี้ จะมีรายละเอียด ดังนี้

หมายเลข 53 แสดงเวลาในการประมวลผลในลักษณะ HH:MM:SS

หมายเลข 54 แสดงผลการจัดลำดับการทำงานที่ดีที่สุดตามลำดับการทำงาน เวลาของแต่ละงานในแต่ละลำดับ เวลาการทำงานที่ดีที่สุด อุณหภูมิและรอบคำตอบที่ใช้ในการจัดลำดับการทำงานที่ดีที่สุดนี้ ซึ่งเป็นคำตอบของการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

หมายเลข 55 แสดงผลการจัดลำดับการทำงาน ณ ปัจจุบันตามลำดับการทำงาน เวลาของแต่ละงานในแต่ละลำดับ เวลาการทำงาน ณ ปัจจุบัน อุณหภูมิและรอบคำตอบที่ใช้ในการจัดลำดับการทำงาน ณ ปัจจุบันนี้

หมายเลข 56 แสดงผลการจัดลำดับการทำงานจากการทำ Neighbourhood search ซึ่งแสดงคำตอบใหม่ทั้งหมด และเวลาการทำงานที่คำตอบใหม่นั้นด้วย

ภาคผนวก ข

Source Code

ของโปรแกรมการจัดลำดับการทำงานของเครนโดยวิธีการอบอ่อนจำลอง

เพื่อสะดวกแก่ความเข้าใจ จะขอแบ่งการแสดงคำสั่งหรือ Source Code ตามลักษณะของโปรแกรม ซึ่งจะแบ่งได้ดังนี้

1. การเปิดโปรแกรมและหน้าแรกที่เป็นส่วนต้อนรับโปรแกรม
2. ขั้นตอนที่ 1 ระบุค่าข้อมูล
3. ขั้นตอนที่ 2 เลือกวิธีการอบอ่อน
4. ขั้นตอนที่ 3 ระบุเวลางาน
5. ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล
6. ขั้นตอนที่ 5 ประมวลผล
7. คำสั่งเรียกใช้และคำสั่งใน Module

1. การเปิดโปรแกรมและหน้าแรกที่เป็นส่วนต้อนรับโปรแกรม

เมื่อเปิดโปรแกรมจะเข้าสู่หน้าแรกที่เป็นหน้าต้อนรับของโปรแกรมโดยอัตโนมัติ

```
Private Sub Workbook_Activate()
Page1First.Show
End Sub
```

รูปที่ ข.1 แสดงคำสั่งเข้าสู่โปรแกรม

จากรูปที่ ก.4 ในส่วนหน้าแรกของโปรแกรมจะมีปุ่มให้กด 3 ปุ่ม ซึ่ง รูปที่ ข.2-ข.4 จะแสดงคำสั่งเมื่อกดปุ่มหมายเลข 1, 2, 3 ตามลำดับ

```
Private Sub CommandButton1_Click()
Unload Me
Page2EnterData.Show
End Sub
```

รูปที่ ข.2 แสดงคำสั่งเมื่อกด START ในหน้าแรกของโปรแกรม

```
Private Sub CommandButton3_Click()
Page1FirstAbout.Show
End Sub
```

รูปที่ ข.3 แสดงคำสั่งเมื่อกด ABOUT ในหน้าแรกของโปรแกรม

```
Private Sub CommandButton2_Click()
Unload Me
End Sub
```

รูปที่ ข.4 แสดงคำสั่งเมื่อกด EXIT ในหน้าแรกของโปรแกรม

หากกด ABOUT ในหน้าแรกแล้ว จะเข้าสู่หน้าต่าง ABOUT ซึ่งในหน้าต่างจะมีคำสั่งที่ปุ่มหมายเลข 4 จากรูปที่ ก.5 ซึ่งจะแสดงได้ดังรูปที่ ข.5

```
Private Sub CommandButton1_Click()
Unload Me
End Sub
```

รูปที่ ข.5 แสดงคำสั่งเมื่อกด OK ใน ABOUT ของโปรแกรม

2. ขั้นตอนที่ 1 ระบุค่าข้อมูล

```
Dim NumberOfJob As Integer          'ประกาศตัวแปรของจำนวนงานที่ระบุมา
Dim DeTempStart As Double           'ประกาศตัวแปรค่า default ของอุณหภูมิเริ่มต้น
Dim DeTempLast As Double            'ประกาศตัวแปรค่า default ของอุณหภูมิสุดท้าย
Dim DeNumCycle As Double            'ประกาศตัวแปรค่า default ของจำนวนรอบการอบอ่อน
Dim DeCoolingRate As Double         'ประกาศตัวแปรค่า default ของอัตราการเย็นตัว
Dim NumJob As Integer
Dim NumCrane As Integer
Dim TempStart As Double
Dim TempLast As Double
Dim NumCycle As Integer
Dim CoolingRate As Double
```

รูปที่ ข.6 แสดงคำสั่งประกาศตัวแปรในหน้าระบุค่าข้อมูล

เมื่อหน้าต่างโปรแกรมของขั้นตอนที่ 1 ระบุค่าข้อมูล ปรากฏขึ้น จะมีคำสั่งดังรูปที่ ข.7

```
Private Sub UserForm_activate()
    Call ClearData          'ฟังก์ชันอยู่ใน Module1
    Call ClearAllTimeOfJob  'ฟังก์ชันอยู่ใน Module1
    Call ClearSolve         'ฟังก์ชันอยู่ใน Module1
End Sub
```

รูปที่ ข.7 แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 1 ปรากฏ

ต่อไปจะเป็นคำสั่งที่กำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข จากรูปที่ ก.6 ซึ่งรูปที่ ข.8-ข.13 จะแสดงคำสั่งของหมายเลข 5-10 ตามลำดับ

```
Private Sub boxNumJob_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumericForInt(KeyAscii)       'ฟังก์ชันอยู่ใน Module1
End Sub
```

รูปที่ ข.8 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนงาน

```
Private Sub boxNumCrane_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumericForInt(KeyAscii)       'ฟังก์ชันอยู่ใน Module1
End Sub
```

รูปที่ ข.9 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนเครน

```
Private Sub boxTempStart _KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumeric(KeyAscii)       'ฟังก์ชันอยู่ใน Module1
End Sub
```

รูปที่ ข.10 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอุณหภูมิเริ่มต้น

```
Private Sub boxTempLast_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumeric(KeyAscii)       'ฟังก์ชันอยู่ใน Module1
End Sub
```

รูปที่ ข.11 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอุณหภูมิสุดท้าย

```
Private Sub boxNumcycle_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
   Call checkNumeric(KeyAscii)        'ฟังก์ชันอยู่ใน Module1
End Sub
```

รูปที่ ข.12 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนรอบการอบอ่อน

```
Private Sub boxCoolingRate_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
   Call checkNumeric(KeyAscii)        'ฟังก์ชันอยู่ใน Module1
End Sub
```

รูปที่ ข.13 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอัตราการเย็นตัว

ต่อไปจะเป็นคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าและไปขั้นตอนถัดไปของหน้าระบุค่าข้อมูล ซึ่งเป็นหมายเลข 11 และ 12 จากรูปที่ ก.6 และสามารถแสดงคำสั่งได้ดังรูปที่ ข.14 และ ข.15 ตามลำดับ

```
Private Sub CommandButton2_Click()
'เมื่อคลิก Back จะกลับไปขั้นตอนก่อนหน้า
   Unload Me
   Page1First.Show
End Sub
```

รูปที่ ข.14 แสดงคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าของหน้าขั้นตอนที่ 1 ระบุค่าข้อมูล

```
Private Sub CommandButton1_Click()
'การไปยังขั้นตอนถัดไป
   'การตั้งค่า default ของค่าต่างๆ
   DeTempStart = 100
   DeTempLast = 0.00001
   DeNumCycle = 5
   DeCoolingRate = 0.9

   'ตรวจสอบว่าได้ระบุจำนวนงานหรือเครนหรือยัง ถ้ายังจะมีหน้าต่างเตือนให้ระบุจำนวน
   If boxNumJob = "" Or boxNumCrane = "" Then
      If boxNumJob = "" And boxNumCrane = "" Then
         MsgBox "กรุณาระบุจำนวนงานและจำนวนเครน"
      ElseIf boxNumJob = "" Then
         MsgBox "กรุณาระบุจำนวนงาน"
      ElseIf boxNumCrane = "" Then
         MsgBox "กรุณาระบุจำนวนเครน"
      End If

   Else
      'สั่งให้นำค่าที่ระบุมาแสดงใน Worksheets("EnterData") ถ้าไม่ใส่ค่าจะใช้ค่า default
      If boxTempStart = "" Then
         boxTempStart = DeTempStart
      End If
      If boxTempLast = "" Then
         boxTempLast = DeTempLast
      End If
      If boxNumCycle = "" Then
         boxNumCycle = DeNumCycle
      End If
```

รูปที่ ข.15 แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไปของหน้าขั้นตอนที่ 1 ระบุค่าข้อมูล

```
    If boxCoolingRate = "" Then
        boxCoolingRate = DeCoolingRate
    End If

    NumJob = boxNumJob
    NumCrane = boxNumCrane
    TempStart = boxTempStart
    TempLast = boxTempLast
    NumCycle = boxNumCycle
    CoolingRate = boxCoolingRate

    'ตรวจสอบว่าตรงตามข้อกำหนดหรือไม่
    If NumJob <= NumCrane Or TempStart < TempLast Or CoolingRate <= 0 Or CoolingRate >= 1 Then
        If NumJob <= NumCrane Then
            MsgBox "จำนวนงานต้องมากกว่าจำนวนเครน"
        End If
        If TempStart < TempLast Then
            MsgBox "อุณหภูมิเริ่มต้นต้องมากกว่าหรือเท่ากับอุณหภูมิสุดท้าย"
        End If
        If CoolingRate <= 0 Or CoolingRate >= 1 Then
            MsgBox "อัตราการเย็นตัวจะต้องมากกว่า 0 และน้อยกว่า 1"
        End If

    Else
        'การบันทึกข้อมูลใน Worksheets("EnterData")
        Worksheets("EnterData").Cells(9, 7).Value = NumJob
        Worksheets("EnterData").Cells(10, 7).Value = NumCrane
        Worksheets("EnterData").Cells(9, 15).Value = TempStart
        Worksheets("EnterData").Cells(10, 15).Value = TempLast
        Worksheets("EnterData").Cells(11, 15).Value = NumCycle
        Worksheets("EnterData").Cells(12, 15).Value = CoolingRate

        ให้ลบค่าเวลาเดิมทิ้ง แล้วเตรียมเพื่อระบุค่าใหม่
        Call ClearAllTimeOfJob          'ฟังก์ชันอยู่ใน Module1
        Call PreparationToInputTime     'ฟังก์ชันอยู่ใน Module1

        ให้แสดงหน้าถัดไป
        Unload Me
        Page3Search.Show
    End If
  End If
End Sub
```

รูปที่ ข.15 (ต่อ) แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไปของหน้าขั้นตอนที่ 1 ระบุค่าข้อมูล

3. ขั้นตอนที่ 2 เลือกวิธีการอบอ่อน

เมื่อหน้าต่างโปรแกรมของขั้นตอนที่ 2 เลือกวิธีการอบอ่อน ปรากฏขึ้น จะมีคำสั่งดังรูปที่ ข.16

```
Private Sub UserForm_activate()
   If Worksheets("EnterData").Cells(13, 15) = 1 Then
      OptionButton1.Value = True
      MultiPage1.Value = 0
   ElseIf Worksheets("EnterData").Cells(13, 15) = 2 Then
      OptionButton2.Value = True
      MultiPage1.Value = 1
   ElseIf Worksheets("EnterData").Cells(13, 15) = 3 Then
      OptionButton3.Value = True
      MultiPage1.Value = 2
   ElseIf Worksheets("EnterData").Cells(13, 15) = 4 Then
      OptionButton4.Value = True
      MultiPage1.Value = 3
   Else
      OptionButton1.Value = True
      MultiPage1.Value = 0
   End If
End Sub
```

รูปที่ ข.16 แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 2 ปรากฏ

เมื่อผู้ใช้ได้ทำการเลือกวิธีการอบอ่อนโดยการคลิกที่เลือกหลักการในการอบอ่อนตามหมายเลข 13-16 จากรูปที่ ก.8 จะมีคำสั่งดังรูปที่ ข.17-ข.20

```
Private Sub OptionButton1_Change()
   If OptionButton1.Value = True Then
      MultiPage1.Value = 0
   End If
End Sub
```

รูปที่ ข.17 แสดงคำสั่งเมื่อเลือกหลักการเลื่อนตำแหน่ง

```
Private Sub OptionButton2_Change()
   If OptionButton2.Value = True Then
      MultiPage1.Value = 1
   End If
End Sub
```

รูปที่ ข.18 แสดงคำสั่งเมื่อเลือกหลักการคงที่หนึ่งตำแหน่ง

```
Private Sub OptionButton3_Change()
   If OptionButton3.Value = True Then
      MultiPage1.Value = 2
   End If
End Sub
```

รูปที่ ข.19 แสดงคำสั่งเมื่อเลือกหลักการสลับที่

```
Private Sub OptionButton4_Change()
   If OptionButton4.Value = True Then
      MultiPage1.Value = 3
   End If
End Sub
```

รูปที่ ข.20 แสดงคำสั่งเมื่อเลือกหลักการพิจารณาที่ 2 ตำแหน่ง

ต่อไปจะเป็นคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าและไปขั้นตอนถัดไปของหน้าเลือกวิธีการอบอ่อน ซึ่งเป็นหมายเลข 21 และ 22 จากรูปที่ ก.8 และสามารถแสดงคำสั่งได้ดังรูปที่ ข.21 และ ข.22 ตามลำดับ

```
Private Sub CommandButton2_Click()
    Unload Me
    Page2EnterData.Show
End Sub
```

รูปที่ ข.21 แสดงคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าของหน้าขั้นตอนที่ 2 เลือกวิธีการอบอ่อน

```
Private Sub CommandButton1_Click()
'เป็นคำสั่งให้แสดงค่าการเลือวิธีการทำ Neighbourhood search ใน Worksheets("EnterData")

    If OptionButton1.Value = True Then
        Worksheets("EnterData").Cells(13, 15) = 1
        Worksheets("EnterData").Cells(1, 1) = 1
        Unload Me
        Page4EnterTime.Show

    ElseIf OptionButton2.Value = True Then
        Worksheets("EnterData").Cells(13, 15) = 2
        Worksheets("EnterData").Cells(1, 1) = 1
        Unload Me
        Page4EnterTime.Show

    ElseIf OptionButton3.Value = True Then
        Worksheets("EnterData").Cells(13, 15) = 3
        Worksheets("EnterData").Cells(1, 1) = 1
        Unload Me
        Page4EnterTime.Show

    ElseIf OptionButton4.Value = True Then
        Worksheets("EnterData").Cells(13, 15) = 4
        Worksheets("EnterData").Cells(1, 1) = 1
        Unload Me
        Page4EnterTime.Show

    Else
        MsgBox "กรุณาเลือกวิธีการทำ Neighbourhood Search"
    End If

End Sub
```

รูปที่ ข.22 แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไปของหน้าขั้นตอนที่ 2 เลือกวิธีการอบอ่อน

4. ขั้นตอนที่ 3 ระบุเวลางาน

```
Dim NumberOfJob As Integer          'ประกาศตัวแปรจำนวนงานที่ระบุเวลาแล้ว
Dim CountNumberOfJob As Integer     'ประกาศตัวแปรให้นับจำนวนงานที่ระบุเวลาแล้ว
```

รูปที่ ข.23 แสดงคำสั่งประกาศตัวแปรในหน้าระบุเวลางาน

เมื่อหน้าต่างโปรแกรมของขั้นตอนที่ 3 ระบุเวลางาน ปรากฏขึ้น จะมีคำสั่งดังรูปที่ ข.24

```
Private Sub UserForm_activate()
    CountNumberOfJob = Worksheets("EnterData").Cells(1, 1).Value
    NumberOfJob = Worksheets("EnterData").Cells(9, 7).Value

    Label1.Caption = "งานที่ " & CountNumberOfJob
    boxTimeJob = Worksheets("EnterData").Cells(24, 3 + CountNumberOfJob).Value

    If CountNumberOfJob < NumberOfJob Then
        CommandButton1.Caption = "ระบุเวลางานถัดไป"
    ElseIf CountNumberOfJob = NumberOfJob Then
        CommandButton1.Caption = "ไปขั้นตอนถัดไป"
    End If

    If CountNumberOfJob = 1 Then
        CommandButton2.Caption = "กลับไปขั้นตอนก่อนหน้า"
    ElseIf CountNumberOfJob > 1 Then
        CommandButton2.Caption = "ระบุเวลางานถัดไป"
    End If

End Sub
```

รูปที่ ข.24 แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 3 ปรากฏ

ต่อไปจะเป็นคำสั่งที่กำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข จากรูปที่ ก.9 ซึ่งรูปที่ ข.25 จะแสดงคำสั่งของหมายเลข 24

```
Private Sub boxTimeJob_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumeric(KeyAscii)
End Sub
```

รูปที่ ข.25 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของเวลางาน

ต่อไปจะเป็นคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าหรือกลับไปเวลางานก่อนหน้า และไปขั้นตอนถัดไปหรือระบุเวลางานถัดไป ของหน้าระบุเวลางาน ซึ่งเป็นหมายเลข 25 และ 26 จากรูปที่ ก.9 และสามารถแสดงคำสั่งได้ดังรูปที่ ข.26 และ ข.27 ตามลำดับ

```
Private Sub CommandButton2_Click()
    NumberOfJob = Worksheets("EnterData").Cells(9, 7).Value

    If CountNumberOfJob > 1 Then
        Worksheets("EnterData").Cells(24, 3 + CountNumberOfJob) = boxTimeJob
        CountNumberOfJob = CountNumberOfJob - 1
        Worksheets("EnterData").Cells(1, 1).Value = CountNumberOfJob
        Unload Me
        Page4EnterTime.Show

    ElseIf CountNumberOfJob = 1 Then
        Worksheets("EnterData").Cells(24, 3 + CountNumberOfJob) = boxTimeJob
        Unload Me
        Page3Search.Show
    End If
End Sub
```

รูปที่ ข.26 แสดงคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้าหรือกลับไปเวลางานก่อนหน้าของขั้นตอนที่ 3 ระบุเวลางาน

```
Private Sub CommandButton1_Click()
    If CountNumberOfJob < NumberOfJob Then
        Worksheets("EnterData").Cells(24, 3 + CountNumberOfJob) = boxTimeJob
        CountNumberOfJob = CountNumberOfJob + 1
        Worksheets("EnterData").Cells(1, 1).Value = CountNumberOfJob
        Unload Me
        Page4EnterTime.Show
    ElseIf CountNumberOfJob = NumberOfJob Then
        Worksheets("EnterData").Cells(24, 3 + CountNumberOfJob) = boxTimeJob
        Unload Me
        Page5ConfirmData.Show
    Else
        MsgBox "Error"
    End If

End Sub
```

รูปที่ ข.27 แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไปหรือระบุเวลางานถัดไปของขั้น**ตอนที่ 3** ระบุเวลางาน

5. ขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล

```
Dim NumberOfJob As Integer              'ประกาศตัวแปรจำนวนงานที่ระบุเวลามา
Dim CountNumberOfJob As Integer         'ประกาศตัวแปรให้นับจำนวนงานที่ระบุเวลา หรือต้องการแก้ไข
Dim NumJob As Integer
Dim NumCrane As Integer
Dim TempStart As Double
Dim TempLast As Double
Dim NumCycle As Integer
Dim CoolingRate As Double
Dim Search As Integer
```

รูปที่ ข.28 แสดงคำสั่งประกาศตัวแปรในหน้ายืนยันและแก้ไขข้อมูล

เมื่อหน้าต่างโปรแกรมของขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูลปรากฏขึ้น จะมีคำสั่งดังรูปที่ ข.29

```
Private Sub UserForm_activate()
    NumberOfJob = Worksheets("EnterData").Cells(9, 7).Value

    Label9.Caption = Worksheets("EnterData").Cells(9, 7).Value & "   งาน"
    Label10.Caption = Worksheets("EnterData").Cells(10, 7).Value & "   เครน"
    Label11.Caption = Worksheets("EnterData").Cells(9, 15).Value & "   องศาเซลเซียส "
    Label12.Caption = Worksheets("EnterData").Cells(10, 15).Value & "   องศาเซลเซียส"
    Label13.Caption = Worksheets("EnterData").Cells(11, 15).Value & "   รอบ "
    Label14.Caption = Worksheets("EnterData").Cells(12, 15).Value

    If Worksheets("EnterData").Cells(13, 15).Value = 1 Then
        Label15.Caption = Worksheets("EnterData").Cells(13, 15).Value & ". หลักการเลื่อนตำแหน่ง"
    ElseIf Worksheets("EnterData").Cells(13, 15).Value = 2 Then
        Label15.Caption = Worksheets("EnterData").Cells(13, 15).Value & ". หลักการคงที่หนึ่งตำแหน่ง"
    ElseIf Worksheets("EnterData").Cells(13, 15).Value = 3 Then
        Label15.Caption = Worksheets("EnterData").Cells(13, 15).Value & ". หลักการสลับที่"
```

รูปที่ ข.29 แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 4 ปรากฏ

```
    Elself Worksheets("EnterData").Cells(13, 15).Value = 4 Then
        Label15.Caption = Worksheets("EnterData").Cells(13, 15).Value & ". หลักการพิจารณาที่ 2 ตำแหน่ง"
    Else
        Label15.Caption = "โปรดเลือกวิธีการใหม่"
    End If

    For i = 1 To NumberOfJob
        ListBox1.AddItem ("   งานที่ " & i & "     ใช้เวลาทำงาน   " & _ Worksheets("EnterData").Cells(24, 3 + i).Value & "  นาที ")
    Next i

End Sub
```

รูปที่ ข.29 (ต่อ) แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 4 ปรากฏ

เมื่อผู้ใช้ต้องการแก้ไขข้อมูล จะเลือกแก้ไขข้อมูลที่หมายเลข 28 จากรูปที่ ก.10 ซึ่งมีคำสั่งดังรูปที่ ข.30

```
Private Sub CommandButton2_Click()
    Page5EditData.Show
End Sub
```

รูปที่ ข.30 แสดงคำสั่งเมื่อเลือกแก้ไขข้อมูลในหน้ายืนยันและแก้ไขข้อมูล

ซึ่งเมื่อเลือกแก้ไขข้อมูลแล้ว หน้าต่างแก้ไขข้อมูลจะปรากฏ ดังรูปที่ ก.11 ซึ่งมีคำสั่งในการปรากฏของหน้าต่าง ดังรูปที่ ข.31

```
Private Sub UserForm_activate()
    boxEditCrane = Worksheets("EnterData").Cells(10, 7).Value
    boxEditTempStart = Worksheets("EnterData").Cells(9, 15).Value
    boxEditTempLast = Worksheets("EnterData").Cells(10, 15).Value
    boxEditNumCycle = Worksheets("EnterData").Cells(11, 15).Value
    boxEditCoolingRate = Worksheets("EnterData").Cells(12, 15).Value
    ComboBox1.AddItem "1. หลักการเลื่อนตำแหน่ง"
    ComboBox1.AddItem "2. หลักการคงที่หนึ่งตำแหน่ง"
    ComboBox1.AddItem "3. หลักการสลับที่"
    ComboBox1.AddItem "4. หลักการพิจารณาที่ 2 ตำแหน่ง"
    ComboBox1.ListIndex = Worksheets("EnterData").Cells(13, 15).Value - 1
End Sub
```

รูปที่ ข.31 แสดงคำสั่งเมื่อหน้าต่างแก้ไขข้อมูล ปรากฏ

ในการเลือกแก้ไขข้อมูลใด จะมีคำสั่งที่กำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข จากรูปที่ ก.11 ซึ่งรูปที่ ข.32-ข.36 จะแสดงคำสั่งของหมายเลข 33-37 ตามลำดับ

```
Private Sub boxEditCrane_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumericForInt(KeyAscii)
End Sub
```

รูปที่ ข.32 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนเครน

```
Private Sub boxEditTempStart_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumeric(KeyAscii)
End Sub
```

รูปที่ ข.33 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอุณหภูมิเริ่มต้น

```
Private Sub boxEditTempLast_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumeric(KeyAscii)
End Sub
```

รูปที่ ข.34 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอุณหภูมิสุดท้าย

```
Private Sub boxEditNumCycle_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumeric(KeyAscii)
End Sub
```

รูปที่ ข.35 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของจำนวนรอบการหาคำตอบ

```
Private Sub boxEditCoolingRate_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumeric(KeyAscii)
End Sub
```

รูปที่ ข.36 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของอัตราการเย็นตัว

เมื่อแก้ไขข้อมูลเสร็จสิ้นแล้วทำการบันทึกข้อมูล ต่อไปเป็นคำสั่งให้บันทึกข้อมูลเมื่อกดหมายเลข 36 จากรูปที่ ก.11 ดังรูปที่ ข.37

```
Private Sub CommandButton2_Click()

    Worksheets("EnterData").Cells(10, 7) = boxEditCrane
    Worksheets("EnterData").Cells(9, 15) = boxEditTempStart
    Worksheets("EnterData").Cells(10, 15) = boxEditTempLast
    Worksheets("EnterData").Cells(11, 15) = boxEditNumCycle
    Worksheets("EnterData").Cells(12, 15) = boxEditCoolingRate
    Worksheets("EnterData").Cells(13, 15) = ComboBox1.ListIndex + 1
    Unload Me
    Page5ConfirmData.ListBox1.Clear
    Page5ConfirmData.Hide
    Page5ConfirmData.Show           'เพื่อการอัปเดตข้อมูล
End Sub
```

รูปที่ ข.37 แสดงคำสั่งเมื่อกดบันทึกในหน้าแก้ไขข้อมูล

เมื่อแก้ไขข้อมูลเสร็จสิ้น หน้าต่างขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูลจะปรากฏ และหากผู้ใช้ต้องการแก้ไขเวลางาน จากรูปที่ ก.10 สามารถเลือกแก้ไขเวลางานโดยการดับเบิ้ลคลิกที่งานที่ต้องการแก้ไข ซึ่งมีคำสั่งตามรูปที่ ข.38 หรือสามารถเลือกแก้ไขได้โดยการเลือกงานที่ต้องการแก้ไขเวลา และกดแก้ไขเวลาทำงานที่หมายเลข 30 ซึ่งมีคำสั่งตามรูปที่ ข.39

```
Private Sub ListBox1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    Worksheets("EnterData").Cells(1, 1).Value = ListBox1.ListIndex + 1
    Page5EditTime.Show
    ListBox1.Clear
End Sub
```

รูปที่ ข.38 แสดงคำสั่งเมื่อกดดับเบิ้ลคลิกที่งานที่ต้องการแก้ไขเวลาทำงานในหน้ายืนยันและแก้ไขข้อมูล

```
Private Sub CommandButton3_Click()
    Worksheets("EnterData").Cells(1, 1).Value = ListBox1.ListIndex + 1
    If Worksheets("EnterData").Cells(1, 1).Value = 0 Then
        MsgBox "กรุณาเลือกงานและเวลาที่ต้องการแก้ไข"
    Else
        Page5EditTime.Show
    End If
End Sub
```

รูปที่ ข.39 แสดงคำสั่งเมื่อกดแก้ไขเวลาทำงานในหน้ายืนยันและแก้ไขข้อมูล

ซึ่งเมื่อเลือกแก้ไขเวลางานแล้ว หน้าต่างแก้ไขเวลางานจะปรากฏ ดังรูปที่ ก.12 ซึ่งมีคำสั่งในการปรากฏของหน้าต่าง ดังรูปที่ ข.40

```
Private Sub UserForm_activate()

    Label1.Caption = "งานที่ " & Worksheets("EnterData").Cells(1, 1).Value
    boxEditTime = _
      Worksheets("EnterData").Cells(24, 3 + Worksheets("EnterData").Cells(1, 1)).Value
End Sub
```

รูปที่ ข.40 แสดงคำสั่งเมื่อหน้าต่างแก้ไขเวลางาน ปรากฏ

ในการเลือกแก้ไขเวลางานใด จะมีคำสั่งที่กำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข จากรูปที่ ก.12 ซึ่งรูปที่ ข.41 จะแสดงคำสั่งของหมายเลข 41

```
Private Sub boxEditTime_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call checkNumeric(KeyAscii)
End Sub
```

รูปที่ ข.41 แสดงคำสั่งกำหนดให้สามารถระบุค่าได้เพียงแค่ตัวเลข ของเวลางาน

เมื่อแก้ไขเวลางานเสร็จสิ้นแล้วทำการบันทึกข้อมูล ต่อไปเป็นคำสั่งให้บันทึกข้อมูลเมื่อกดหมายเลข 42 จากรูปที่ ก.12 ดังรูปที่ ข.42

```
Private Sub CommandButton1_Click()
    Worksheets("EnterData").Cells(24, 3 + Worksheets("EnterData").Cells(1, 1)).Value = boxEditTime
    Unload Me
End Sub
```

รูปที่ ข.42 แสดงคำสั่งเมื่อกดบันทึกในหน้าแก้ไขเวลางาน

เมื่อแก้ไขเวลางานเสร็จสิ้น หน้าต่างขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูลจะปรากฏ ต่อไปจะเป็นคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้า และไปขั้นตอนถัดไป ของหน้ายืนยันและแก้ไขข้อมูล ซึ่งเป็นหมายเลข 31 และ 32 จากรูปที่ ก.10 และสามารถแสดงคำสั่งได้ดังรูปที่ ข.43 และ ข.44 ตามลำดับ

```
Private Sub CommandButton5_Click()
    Worksheets("EnterData").Cells(1, 1) = 1
    Unload Me
    Page4EnterTime.Show
End Sub
```

รูปที่ ข.43 แสดงคำสั่งเมื่อกดกลับไปขั้นตอนก่อนหน้า ของขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล

```
Private Sub CommandButton1_Click()
Dim cellNumJob As Integer
Dim cellNumCrane As Integer
Dim cellTempStart As Double
Dim cellTempLast As Double
Dim cellCoolingRate As Double
Dim cellNumCycle As Integer
Dim cellSearch As Integer
Dim Check As Integer

If Worksheets("EnterData").Cells(9, 7) = "" Or Worksheets("EnterData").Cells(10, 7) = "" Or Worksheets("EnterData").Cells(9, 15) = "" Or _
    Worksheets("EnterData").Cells(10, 15) = "" Or Worksheets("EnterData").Cells(11, 15) = "" Or Worksheets("EnterData").Cells(12, 15) = "" Or _
    Worksheets("EnterData").Cells(13, 15) = "" Then

    MsgBox "กรุณาระบุข้อมูลให้ถูกต้อง ครบถ้วน"

Else
    cellNumJob = Worksheets("EnterData").Cells(9, 7).Value
    cellNumCrane = Worksheets("EnterData").Cells(10, 7).Value
    cellTempStart = Worksheets("EnterData").Cells(9, 15).Value
    cellTempLast = Worksheets("EnterData").Cells(10, 15).Value
    cellNumCycle = Worksheets("EnterData").Cells(11, 15).Value
    cellCoolingRate = Worksheets("EnterData").Cells(12, 15).Value
    cellSearch = Worksheets("EnterData").Cells(13, 15).Value

    Check = 1
    For i = 1 To cellNumJob
        If Worksheets("EnterData").Cells(24, 3 + i) = "" Then
            Check = Check + 1
        End If
    Next i

    If Check <> 1 Then
        MsgBox "โปรดระบุเวลาการทำงานของแต่ละงานให้ถูกต้องครบถ้วน"

    ElseIf cellNumJob <= cellNumCrane Or cellTempStart < cellTempLast Or cellTempLast <= 0 Or cellNumCycle <= 0 Or _
            cellCoolingRate <= 0 Or cellCoolingRate >= 1 Or cellSearch <= 0 Or cellSearch > 4 Then

        If cellNumJob <= cellNumCrane Then
            MsgBox "จำนวนงานต้องมากกว่าจำนวนเครน"
        End If

                If cellTempStart < cellTempLast Then
            MsgBox "อุณหภูมิเริ่มต้นต้องมากกว่าหรือเท่าอุณหภูมิสุดท้าย"
        End If
```

รูปที่ ข.44 แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไป ของขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล

```
        If cellTempLast <= 0 Then
            MsgBox "อุณหภูมิสุดท้ายต้องมากกว่า 0"
        End If

        If cellNumCycle <= 0 Then
            MsgBox "กรุณาระบุจำนวนรอบการหาคำตอบให้ถูกต้อง"
        End If

        If cellCoolingRate <= 0 Or cellCoolingRate >= 1 Then
            MsgBox "ค่าอัตราการเย็นตัวต้องมากกว่า 0 และน้อยกว่า 1"
        End If

        If cellSearch <= 0 Or cellSearch > 4 Then
            MsgBox "กรุณาเลือกวิธีการทำ Neighbourhood Search ให้ถูกต้อง"
        End If
    Else
        Unload Me
        Worksheets("Solve").Activate
        Page6Processing.Show
    End If
End If
End Sub
```

รูปที่ ข.44 (ต่อ) แสดงคำสั่งเมื่อกดไปขั้นตอนถัดไป ของขั้นตอนที่ 4 ยืนยันและแก้ไขข้อมูล

6. ขั้นตอนที่ 5 ประมวลผล

```
Dim TotalNS As Integer
Dim NumOfJob As Integer
Dim NumOfCrane As Integer
Dim PTime() As Integer
Dim BestRow As Integer
Dim CurSolJob() As Integer
Dim CurSolCrane() As Integer
Dim A As Double
Dim z As Integer
Dim NumOfAns As Integer
Dim K As Double
Dim StartTime As Date
Dim EndTime As Date
Dim RunningTime
```

รูปที่ ข.45 แสดงคำสั่งประกาศตัวแปรในหน้าประมวลผล

เมื่อหน้าต่างโปรแกรมของขั้นตอนที่ 5 ประมวลผลปรากฏขึ้น จะมีคำสั่งดังรูปที่ ข.46

```
Private Sub UserForm_activate()
Dim NumberOfJob As Integer
NumberOfJob = Worksheets("EnterData").Cells(9, 7).Value
ListBox1.ColumnCount = NumberOfJob + 1
ListBox1.ColumnWidths = 80
ListBox1.RowSource = "B15:AZ17"
Label33.Caption = Format(Worksheets("Solve").Cells(11, 7), "h.mm:ss")
Label35.Caption = Worksheets("Solve").Cells(18, 5) & "  นาที"
Label38.Caption = Worksheets("Solve").Cells(19, 4).Value & "  องศาเซลเซียส"
Label40.Caption = Worksheets("Solve").Cells(20, 4).Value
End Sub
```

รูปที่ ข.46 แสดงคำสั่งเมื่อหน้าต่างขั้นตอนที่ 5 ปรากฏ

หลังจากนี้ผู้ใช้สามารถเลือกกดเพื่อกลับไปแก้ไขข้อมูล เริ่มการประมวลผล ไปยังหน้า EXCEL **และเริ่มระบุข้อมูลใหม่ทั้งหมด**

ต่อไปจะเป็นคำสั่งเมื่อกดกลับไปแก้ไขข้อมูล คือกดหมายเลข 44 จากรูปที่ ก.13 **ดังรูปที่ ข.47**

```
Private Sub CommandButton2_Click()
    Unload Me
    Page5ConfirmData.Show
End Sub
```

รูปที่ ข.47 แสดงคำสั่งเมื่อกดกลับไปแก้ไขข้อมูล ของขั้นตอนที่ 5 ประมวลผล

ต่อไปจะเป็นคำสั่งเมื่อกดเริ่มระบุข้อมูลใหม่ทั้งหมด คือกดหมายเลข 47 จาก**รูปที่ ก.13 ดัง รูปที่ ข.48**

```
Private Sub CommandButton5_Click()
    Unload Me
    Page2EnterData.Show
End Sub
```

รูปที่ ข.48 แสดงคำสั่งเมื่อกดเริ่มระบุข้อมูลใหม่ทั้งหมด ของขั้นตอนที่ 5 ประมวลผล

ต่อไปจะเป็นคำสั่งเมื่อกดเริ่มการประมวลผล คือกดหมายเลข 45 จากรูปที่ ก.13 **ดังรูปที่ ข.49**

```
Private Sub CommandButton1_Click()
    Page6ProgressBar.Show
End Sub
```

รูปที่ ข.49 แสดงคำสั่งเมื่อกดเริ่มการประมวลผล ของขั้นตอนที่ 5 ประมวลผล

เมื่อกดเริ่มการประมวลผลแล้ว จะมีหน้าต่างที่แสดงแถบความก้าวหน้าปรากฏ ซึ่งมาคำสั่งในการประกาศตัวแปร ดังรูปที่ ข.50

```
Dim TotalNS As Integer
Dim NumOfJob As Integer
Dim NumOfCrane As Integer
Dim PTime() As Integer
Dim BestRow As Integer
Dim CurSolJob() As Integer
Dim CurSolCrane() As Integer
Dim A As Double
Dim z As Integer
Dim NumOfAns As Integer
Dim K As Double
Dim StartTime As Date
Dim EndTime As Date
Dim RunningTime
Dim can As Integer
```

รูปที่ ข.50 แสดงคำสั่งประกาศตัวแปรในหน้าต่างแสดงแถบความก้าวหน้า

เมื่อแถบความก้าวหน้าปรากฏ จะเป็นการดำเนินการประมวลผลการจัดลำดับการทำงานของเครน จากรูปที่ ก.14 ซึ่งมีคำสั่งดังรูปที่ ข.51

```
Private Sub UserForm_activate()
'เริ่มการประมวลผล

Dim Eq As Integer, No As Integer
Dim countProcess As Double, countMax As Double, countNow As Double
Dim T As Double, Tmax As Double, Tmin As Double, Trate As Double, TT As Double
Dim Sbest As Double, Scur As Double, Snew As Double
Dim RbestOfJob() As Integer, RcurOfJob() As Integer, RnewOfJob() As Integer
Dim RbestOfCrane() As Integer, RcurOfCrane() As Integer, RnewOfCrane() As Integer
Dim Prob As Double, RanSA As Double, BestRow As Integer
Dim Snews() As Integer

Worksheets("EnterData").Cells(1, 1) = 0

' อธิบายตัวแปร
   'Eq              = จำนวนรอบที่ใช้ในการหาคำตอบเพื่อเข้าสู่จุดสมดุล
   'T               = อุณหภูมิปัจจุบัน
   'Tmax            = อุณหภูมิเริ่มต้น
   'Tmin            = อุณหภูมิสิ้นสุด
   'Trate           = อัตราการเย็นตัว
   'Sbest           = คำตอบที่ดีที่สุด ตั้งแต่ต้นจนถึงปัจจุบัน
   'Scur            = คำตอบ ณ ปัจจุบัน
   'Snew            = คำตอบที่ได้จาก Neighbourhood search
   'RbestOfJob()    = ลำดับการทำงานที่ดีที่สุดของงาน ตั้งแต่ต้นจนถึงปัจจุบัน
   'RcurOfJob()     = ลำดับการทำงาน ณ ปัจจุบัน ของงาน
   'RnewOfJob()     = ลำดับการทำงานที่ได้จาก Neighbourhood search ของงาน
   'RbestOfCrane()  = ลำดับการทำงานที่ดีที่สุดของเครน ตั้งแต่ต้นจนถึงปัจจุบัน
   'RcurOfCrane()   = ลำดับการทำงาน ณ ปัจจุบัน ของเครน
   'RnewOfCrane()   = ลำดับการทำงานที่ได้จาก Neighbourhood search ของเครน
   'Snews()         = คำตอบที่ได้จากรูปแบบการทำ neighbourhood search แต่ละรูปแบบ
   'BestRow         = แถวของรูปแบบคำตอบจากการทำ neighbourhood search ที่ดีที่สุดในแต่ละรอบ
   'Prob            = ค่าความน่าจะเป็นด้วยสมการ exponential
   'RanSA           = ค่าการสุ่มเพื่อเทียบกับค่าความน่าจะเป็น prob
```

รูปที่ ข.51 แสดงคำสั่งเมื่อหน้าต่างแถบความก้าวหน้า ปรากฏ ซึ่งจะเป็นการดำเนินการประมวล ผลการจัดลำดับการทำงาน

```
    'NumOfJob          = จำนวนของงาน
    'NumOfCrane        = จำนวนของเครน

'เป็นส่วนของ Process bar
Tmax = Worksheets("EnterData").Cells(9, 15).Value
Tmin = Worksheets("EnterData").Cells(10, 15).Value
Eq = Worksheets("EnterData").Cells(11, 15).Value
Trate = Worksheets("EnterData").Cells(12, 15).Value
T = Tmax
No = 0
StartTime = Time

Do While T >= Tmin
    No = No + 1
    T = T * Trate
Loop

countMax = No * Eq

Page6ProgressBar.ProgressBar1.Value = 0
Label1.Caption = Format(0, "0%")

'เริ่มการประมวลผล
NumOfJob = Worksheets("EnterData").Cells(9, 7)

Application.ScreenUpdating = False

Call ClearSolve                 'ล้างการประมวลผลเดิม 'ฟังก์ชันอยู่ใน Module1
Call Newns                      'สุ่มคำตอบเริ่มต้น  'ฟังก์ชันอยู่ใน Module1
Call MakespanIntervalFirstAns   'หา Makespan ของคำตอบเริ่มต้น   'ฟังก์ชันอยู่ใน Module1

'จัดรูปแบบของ cell ของคำตอบโดยใช้หลักการที่ 1,2,3,4
If Worksheets("EnterData").Cells(13, 15) = 1 Or Worksheets("EnterData").Cells(13, 15) = 2 Or _
Worksheets("EnterData").Cells(13, 15) = 3 Then
    Call SetFormatProcessing123         'ฟังก์ชันอยู่ใน Module1
    NumOfAns = 2 * (NumOfJob - 1)
ElseIf Worksheets("EnterData").Cells(13, 15) = 4 Then
    Call SetFormatProcessing4           'ฟังก์ชันอยู่ใน Module1
    NumOfAns = NumOfJob - 1
End If

'วนรอบ SA
ReDim RbestOfJob(1 To NumOfJob)
ReDim RcurOfJob(1 To NumOfJob)
ReDim RnewOfJob(1 To NumOfJob)
ReDim RbestOfCrane(1 To NumOfJob)
ReDim RcurOfCrane(1 To NumOfJob)
ReDim RnewOfCrane(1 To NumOfJob)
ReDim Snews(1 To NumOfAns)

'จะเริ่มการทำ SA โดยการให้คำตอบเริ่มต้นเป็นคำตอบที่ดีที่สุดก่อน

For i = 1 To NumOfJob
    Worksheets("Solve").Cells(15, 2 + i) = Worksheets("Solve").Cells(26, 2 + i)
    Worksheets("Solve").Cells(16, 2 + i) = Worksheets("Solve").Cells(27, 2 + i)
Next i
```

รูปที่ ข.51 (ต่อ) แสดงคำสั่งเมื่อหน้าต่างแถบความก้าวหน้า ปรากฏ ซึ่งจะเป็นการดำเนินการประมวลผลการจัดลำดับการทำงาน

```
Worksheets("Solve").Cells(18, 5) = Worksheets("Solve").Cells(28, 5)

For i = 1 To NumOfJob
    RcurOfJob(i) = Worksheets("Solve").Cells(26, 2 + i).Value
    RbestOfJob(i) = Worksheets("Solve").Cells(15, 2 + i).Value
    RcurOfCrane(i) = Worksheets("Solve").Cells(27, 2 + i).Value
    RbestOfCrane(i) = Worksheets("Solve").Cells(16, 2 + I).Value
Next i

Scur = Worksheets("Solve").Cells(28, 5).Value
Sbest = Worksheets("Solve").Cells(18, 5).Value

'แสดงเวลางานของแต่ละงานที่ใช้ ในลำดับการทำงานที่ดีที่สุด
For i = 1 To NumOfJob
    For j = 1 To NumOfJob
        If Worksheets("EnterData").Cells(23, 3 + j).Value = Worksheets("Solve").Cells(15, 2 + i).Value Then
            Worksheets("Solve").Cells(17, 2 + i).Value = Worksheets("EnterData").Cells(24, 3 + j).Value
        End If
    Next j
Next i

'กำหนดค่าอุณหภูมิเริ่มต้นและสิ้นสุด
Tmax = Worksheets("EnterData").Cells(9, 15).Value
Tmin = Worksheets("EnterData").Cells(10, 15).Value
Eq = Worksheets("EnterData").Cells(11, 15).Value
Trate = Worksheets("EnterData").Cells(12, 15).Value

countProcess = 0
T = Tmax
Worksheets("Solve").Cells(19, 4).Value = Tmax
Worksheets("Solve").Cells(20, 4).Value = "เริ่มต้น"
Do While T >= Tmin

    If T <> Tmax Then
        Call Newns                          'สุ่มคำตอบเริ่มต้น  ฟังก์ชันอยู่ใน Module1
        Call MakespanIntervalFirstAns       'หา Makespan ของคำตอบเริ่มต้น ฟังก์ชันอยู่ใน Module1
    End If

For v = 1 To Eq

        countProcess = countProcess + 1

        If Worksheets("EnterData").Cells(1, 1).Value = 1 Then Exit Sub

        'การทำ Neighbourhood search ตามหลักการต่างๆ
        If Worksheets("EnterData").Cells(13, 15).Value = 1 Then
            Call NewNS1         ฟังก์ชันอยู่ใน Module2

        ElseIf Worksheets("EnterData").Cells(13, 15).Value = 2 Then
            Call NewNS2         ฟังก์ชันอยู่ใน Module2

        ElseIf Worksheets("EnterData").Cells(13, 15).Value = 3 Then
            Call NewNS3         ฟังก์ชันอยู่ใน Module2
```

รูปที่ ข.51 (ต่อ) แสดงคำสั่งเมื่อหน้าต่างแถบความก้าวหน้า ปรากฏ ซึ่งจะเป็นการดำเนินการประมวลผลการจัดลำดับการทำงาน

```
ElseIf Worksheets("EnterData").Cells(13, 15).Value = 4 Then
    Call NewNS4         'ฟังก์ชันอยู่ใน Module2
End If

'การประเมินค่าจากคำตอบที่ได้จากการทำ NS
For z = 1 To NumOfAns
    Call MakespanInterval(z)          'ฟังก์ชันอยู่ใน Module2
Next z

For i = 1 To NumOfAns
    Snews(i) = Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + NumOfJob + 1).Value
Next i
Snew = Snews(1)

For i = 1 To NumOfAns
    If Snews(i) <= Snew Then
        Snew = Snews(i)
        BestRow = i
    End If
Next i

Worksheets("Solve").Cells(34 + NumOfAns + 1, 3).Value = Snew

For i = 1 To NumOfJob
    RnewOfJob(i) = Worksheets("Solve").Cells(34 + BestRow, 2 + i).Value
    RnewOfCrane(i) = Worksheets("Solve").Cells(34 + BestRow, 2 + NumOfJob + 2 + i).Value
Next i

'ทำการประเมินว่าคำตอบที่ได้ มีค่าดีกว่า (ต่ำกว่า) คำตอบปัจจุบัน และคำตอบที่ดีที่สุดหรือไม่
Scur = Worksheets("Solve").Cells(28, 5).Value
Sbest = Worksheets("Solve").Cells(18, 5).Value

If Snew <= Scur Then
    Scur = Snew
    RcurOfJob() = RnewOfJob()
    RcurOfCrane() = RnewOfCrane()

    Worksheets("Solve").Cells(28, 5).Value = Scur

    For i = 1 To NumOfJob
        Worksheets("Solve").Cells(26, 2 + i).Value = RcurOfJob(i)
        Worksheets("Solve").Cells(27, 2 + i).Value = RcurOfCrane(i)
    Next i

    TT = Round(T, 5)

    Worksheets("Solve").Cells(29, 4).Value = TT
    Worksheets("Solve").Cells(30, 4).Value = v

    If Scur < Sbest Then
        Sbest = Scur
        RbestOfJob() = RcurOfJob()
        RbestOfCrane() = RcurOfCrane()

        Worksheets("Solve").Cells(18, 5).Value = Sbest
```

รูปที่ ข.51 (ต่อ) แสดงคำสั่งเมื่อหน้าต่างแถบความก้าวหน้า ปรากฏ ซึ่งจะเป็นการดำเนินการประมวลผลการจัดลำดับการทำงาน

```
            For i = 1 To NumOfJob
               Worksheets("Solve").Cells(15, 2 + i).Value = RbestOfJob(i)
               Worksheets("Solve").Cells(16, 2 + i).Value = RbestOfCrane(i)
            Next i
            'แสดงเวลางานของแต่ละงานที่ใช้ ในลำดับการทำงานที่ดีที่สุด
            For i = 1 To NumOfJob
               For j = 1 To NumOfJob
                  If Worksheets("EnterData").Cells(23, 3 + j).Value = Worksheets("Solve").Cells(15, 2 + i).Value Then
                     Worksheets("Solve").Cells(17, 2 + i).Value = Worksheets("EnterData").Cells(24, 3 + j).Value
                  End If
               Next j
            Next i

            TT = Round(T, 5)
            Worksheets("Solve").Cells(19, 4).Value = TT
            Worksheets("Solve").Cells(20, 4).Value = v

         End If

      ElseIf Snew > Scur Then
         K = 1.38E-23
         RanSA = Rnd
         Prob = Exp(-(Snew - Scur)) / (K * T)

         If RanSA <= Prob Then
            Scur = Snew
            RcurOfJob() = RnewOfJob()
            RcurOfCrane() = RnewOfCrane()

            Worksheets("Solve").Cells(28, 5).Value = Scur

            For i = 1 To NumOfJob
               Worksheets("Solve").Cells(26, 2 + i).Value = RcurOfJob(i)
               Worksheets("Solve").Cells(27, 2 + i).Value = RcurOfCrane(i)
            Next i

            TT = Round(T, 5)
            Worksheets("Solve").Cells(29, 4).Value = TT
            Worksheets("Solve").Cells(30, 4).Value = v

         End If
      End If

      countNow = (countProcess * 100) / countMax
      Page6ProgressBar.ProgressBar1.Value = countNow
      Label1.Caption = Format(countNow / 100, "0%")

      DoEvents
   Next v

   'ลดค่าอุณหภูมิด้วยค่าอัตราการเย็นตัว
   T = T * Trate

Loop
```

รูปที่ ข.51 (ต่อ) แสดงคำสั่งเมื่อหน้าต่างแถบความก้าวหน้า ปรากฏ ซึ่งจะเป็นการดำเนินการประมวลผลการจัดลำดับการทำงาน

```
If Worksheets("EnterData").Cells(1, 1) = 0 Then
    Application.ScreenUpdating = True

    EndTime = Time
    RunningTime = EndTime – StartTime

    Worksheets("Solve").Cells(11, 7) = RunningTime

    Unload Me
    Page6Processing.Hide
    Page6Processing.Show
End If

End Sub
```

รูปที่ ข.51 (ต่อ) แสดงคำสั่งเมื่อหน้าต่างแถบความก้าวหน้า ปรากฏ ซึ่งจะเป็นการดำเนินการประมวลผลการจัดลำดับการทำงาน

แต่หากในขณะที่ทำการประมวลและแถบความก้าวหน้ายังแสดงผลอยู่ **ผู้ใช้ต้องการหยุดการประมวล**ผล โดยกดหมายเลข 48 จากรูปที่ ก.14 มีคำสั่งคำสั่งดังรูปที่ ข.52

```
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
  If CloseMode = vbFormControlMenu Then
    If Not CancelConfirm Then
       Cancel = True                    'ฟังก์ชันอยู่ใน Module2
    End If
  End If
End Sub
```

รูปที่ ข.52 แสดงคำสั่งเมื่อกด X เพื่อหยุดการประมวลผล ของหน้าแสดงแถบความก้าวหน้า

กลับมายังหน้าต่างของขั้นตอนที่ 5 ประมวลผล เมื่อผู้ใช้กดไปยังหน้า EXCEL **ที่หมายเลข 46 จาก**รูปที่ ก.13 จะมีคำสั่งดังรูปที่ ข.53

```
Private Sub CommandButton3_Click()
    Unload Me
    Worksheets("Solve").Activate
End Sub
```

รูปที่ ข.53 แสดงคำสั่งเมื่อกดไปยังหน้า EXCEL **ของขั้นตอนที่ 5 ประมวลผล**

เมื่อผู้ใช้อยู่ในหน้าต่างของ Microsoft Excel **แล้ว จะมีคำสั่งประกาศตัวแปรดังรูปที่ ข.54**

```
Dim TotalNS As Integer
Dim NumOfJob As Integer, NumOfCrane As Integer, PTime() As Integer
Dim BestRow As Integer
Dim CurSolJob() As Integer, CurSolCrane() As Integer
Dim A As Double, z As Integer
Dim NumOfAns As Integer
Dim K As Double
Dim StartTime As Date
Dim EndTime As Date
Dim RunningTime
```

รูปที่ ข.54 แสดงคำสั่งประกาศตัวแปรในหน้าต่างของ Microsoft Excel **ในชีท** Solve

จากรูปที่ ก.17 เมื่อกดหมายเลข 52 คือกลับไปยังโปรแกรม จะมีคำสั่งดังรูปที่ ข.55

```
Private Sub CommandButton4_Click()
    Page6Processing.Show
End Sub
```

รูปที่ ข.55 แสดงคำสั่งเมื่อกดกลับไปยังโปรแกรม ของหน้าต่าง Microsoft Excel ในชีท Solve

จากรูปที่ ก.17 เมื่อกดหมายเลข 51 คือเริ่มการประมวลผล จะเป็นการเริ่มประมวลผลโปรแกรม จะมีคำสั่งดังรูปที่ ข.56

```
Private Sub CommandButton1_Click()

StartTime = Time

NumOfJob = Worksheets("EnterData").Cells(9, 7)

Call ClearSolve                 'ล้างการประมวลผลเดิม 'ฟังก์ชันอยู่ใน Module1
Call NewIns                     'สุ่มคำตอบเริ่มต้น  'ฟังก์ชันอยู่ใน Module1
Call MakespanIntervalFirstAns   'หา Makespan ของคำตอบเริ่มต้น   'ฟังก์ชันอยู่ใน Module1

'จัดรูปแบบของ cell ของคำตอบโดยใช้หลักการที่ 1,2,3,4
If Worksheets("EnterData").Cells(13, 15) = 1 Or Worksheets("EnterData").Cells(13, 15) =_
2 Or Worksheets("EnterData").Cells(13, 15) = 3 Then
    Call SetFormatProcessing123        'ฟังก์ชันอยู่ใน Module1
    NumOfAns = 2 * (NumOfJob - 1)

ElseIf Worksheets("EnterData").Cells(13, 15) = 4 Then
    Call SetFormatProcessing4          'ฟังก์ชันอยู่ใน Module1
    NumOfAns = NumOfJob - 1
End If

'วนรอบ SA
Dim Eq As Integer
Dim T As Double, Tmax As Double, Tmin As Double, Trate As Double, TT As Double
Dim Sbest As Double, Scur As Double, Snew As Double
Dim RbestOfJob() As Integer, RcurOfJob() As Integer, RnewOfJob() As Integer
Dim RbestOfCrane() As Integer, RcurOfCrane() As Integer, RnewOfCrane() As Integer
Dim Prob As Double, RanSA As Double, BestRow As Integer
Dim Snews() As Double

' อธิบายตัวแปร
    'Eq             = จำนวนรอบที่ใช้ในการหาคำตอบเพื่อเข้าสู่จุดสมดุล
    'T              = อุณหภูมิปัจจุบัน
    'Tmax           = อุณหภูมิเริ่มต้น
    'Tmin           = อุณหภูมิสิ้นสุด
    'Trate          = อัตราการเย็นตัว
    'Sbest          = คำตอบที่ดีที่สุด ตั้งแต่ต้นจนถึงปัจจุบัน
    'Scur           = คำตอบ ณ ปัจจุบัน
    'Snew           = คำตอบที่ได้จาก Neighbourhood search
    'RbestOfJob()   = ลำดับการทำงานที่ดีที่สุดของงาน ตั้งแต่ต้นจนถึงปัจจุบัน
    'RcurOfJob()    = ลำดับการทำงาน ณ ปัจจุบัน ของงาน
    'RnewOfJob()    = ลำดับการทำงานที่ได้จาก Neighbourhood search ของงาน
    'RbestOfCrane() = ลำดับการทำงานที่ดีที่สุดของเครน ตั้งแต่ต้นจนถึงปัจจุบัน
    'RcurOfCrane()  = ลำดับการทำงาน ณ ปัจจุบัน ของเครน
    'RnewOfCrane()  = ลำดับการทำงานที่ได้จาก Neighbourhood search ของเครน
```

รูปที่ ข.56 แสดงคำสั่งเมื่อกดเริ่มการประมวลผล ของหน้าต่าง Microsoft Excel ในชีท Solve

```
    'Snews()          = คำตอบที่ได้จากรูปแบบการทำ neighbourhood search แต่ละรูปแบบ
    'BestRow          = แถวของรูปแบบคำตอบจากการทำ neighbourhood search ที่ดีที่สุดในแต่ละรอบ
    'Prob             = ค่าความน่าจะเป็นด้วยสมการ exponential
    'RanSA            = ค่าการสุ่มเพื่อเทียบกับค่าความน่าจะเป็น prob
    'NumOfJob         = จำนวนของงาน
    'NumOfCrane       = จำนวนของเครน

ReDim RbestOfJob(1 To NumOfJob)
ReDim RcurOfJob(1 To NumOfJob)
ReDim RnewOfJob(1 To NumOfJob)
ReDim RbestOfCrane(1 To NumOfJob)
ReDim RcurOfCrane(1 To NumOfJob)
ReDim RnewOfCrane(1 To NumOfJob)
ReDim Snews(1 To NumOfAns)

'จะเริ่มการทำ SA โดยการให้คำตอบเริ่มต้นเป็นคำตอบที่ดีที่สุดก่อน
For i = 1 To NumOfJob
    Worksheets("Solve").Cells(15, 2 + i) = Worksheets("Solve").Cells(26, 2 + i)
    Worksheets("Solve").Cells(16, 2 + i) = Worksheets("Solve").Cells(27, 2 + i)
Next i

Worksheets("Solve").Cells(18, 5) = Worksheets("Solve").Cells(28, 5)

For i = 1 To NumOfJob
    RcurOfJob(i) = Worksheets("Solve").Cells(26, 2 + i).Value
    RbestOfJob(i) = Worksheets("Solve").Cells(15, 2 + i).Value
    RcurOfCrane(i) = Worksheets("Solve").Cells(27, 2 + i).Value
    RbestOfCrane(i) = Worksheets("Solve").Cells(16, 2 + i).Value
Next i

Scur = Worksheets("Solve").Cells(28, 5).Value
Sbest = Worksheets("Solve").Cells(18, 5).Value
'แสดงเวลางานของแต่ละงานที่ใช้ ในลำดับการทำงานที่ดีที่สุด
For i = 1 To NumOfJob
    For J = 1 To NumOfJob
        If Worksheets("EnterData").Cells(23, 3 + J).Value = Worksheets("Solve").Cells(15, 2 + i).Value Then
            Worksheets("Solve").Cells(17, 2 + i).Value = Worksheets("EnterData").Cells(24, 3 + J).Value
        End If
    Next J
Next i

'กำหนดค่าอุณหภูมิเริ่มต้นและสิ้นสุด
Tmax = Worksheets("EnterData").Cells(9, 15).Value
Tmin = Worksheets("EnterData").Cells(10, 15).Value
Eq = Worksheets("EnterData").Cells(11, 15).Value
Trate = Worksheets("EnterData").Cells(12, 15).Value

T = Tmax

Worksheets("Solve").Cells(19, 4).Value = Tmax
Worksheets("Solve").Cells(20, 4).Value = "เริ่มต้น"

Do While T >= Tmin

    If T <> Tmax Then
        Call Newns                    'สุ่มคำตอบเริ่มต้น  ฟังก์ชันอยู่ใน Module1
```

รูปที่ ข.56 (ต่อ) แสดงคำสั่งเมื่อกดเริ่มการประมวลผล ของหน้าต่าง Microsoft Excel ในชีท Solve

```
    Call MakespanIntervalFirstAns       'หา Makespan ของคำตอบเริ่มต้น ฟังก์ชันอยู่ใน Module1
End If

For v = 1 To Eq

    'การทำ Neighbourhood search ตามหลักการต่างๆ
    If Worksheets("EnterData").Cells(13, 15).Value = 1 Then
        Call NewNS1         'ฟังก์ชันอยู่ใน Module2

    Elself Worksheets("EnterData").Cells(13, 15).Value = 2 Then
        Call NewNS2         'ฟังก์ชันอยู่ใน Module2
    Elself Worksheets("EnterData").Cells(13, 15).Value = 3 Then
        Call NewNS3         'ฟังก์ชันอยู่ใน Module2

    Elself Worksheets("EnterData").Cells(13, 15).Value = 4 Then
        Call NewNS4         'ฟังก์ชันอยู่ใน Module2
    End If

    'การประเมินค่าจากคำตอบที่ได้จากการทำ NS
    For z = 1 To NumOfAns
        Call MakespanInterval(z)            'ฟังก์ชันอยู่ใน Module2
    Next z

    For i = 1 To NumOfAns
        Snews(i) = Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + NumOfJob + 1).Value
    Next i

    Snew = Snews(1)

    For i = 1 To NumOfAns
        If Snews(i) <= Snew Then
            Snew = Snews(i)
            BestRow = i
        End If
    Next i

    Worksheets("Solve").Cells(34 + NumOfAns + 1, 3).Value = Snew

    For i = 1 To NumOfJob
        RnewOfJob(i) = Worksheets("Solve").Cells(34 + BestRow, 2 + i).Value
        RnewOfCrane(i) = Worksheets("Solve").Cells(34 + BestRow, 2 + NumOfJob + 2 + i).Value
    Next i

    'ทำการประเมินว่าคำตอบที่ได้ มีค่าดีกว่า (ต่ำกว่า) คำตอบปัจจุบัน และคำตอบที่ดีที่สุดหรือไม่
    Scur = Worksheets("Solve").Cells(28, 5).Value
    Sbest = Worksheets("Solve").Cells(18, 5).Value

    If Snew <= Scur Then
        Scur = Snew
        RcurOfJob() = RnewOfJob()
        RcurOfCrane() = RnewOfCrane()

        Worksheets("Solve").Cells(28, 5).Value = Scur

        For i = 1 To NumOfJob
            Worksheets("Solve").Cells(26, 2 + i).Value = RcurOfJob(i)
            Worksheets("Solve").Cells(27, 2 + i).Value = RcurOfCrane(i)
        Next i
```

รูปที่ ข.56 (ต่อ) แสดงคำสั่งเมื่อกดเริ่มการประมวลผล ของหน้าต่าง Microsoft Excel ในชีท Solve

```
        TT = Round(T, 5)

        Worksheets("Solve").Cells(29, 4).Value = TT
        Worksheets("Solve").Cells(30, 4).Value = v
        Worksheets("Solve").Cells(18, 5).Value = Sbest

        If Scur < Sbest Then
            Sbest = Scur
            RbestOfJob() = RcurOfJob()
            RbestOfCrane() = RcurOfCrane()
            For i = 1 To NumOfJob
                Worksheets("Solve").Cells(15, 2 + i).Value = RbestOfJob(i)
                Worksheets("Solve").Cells(16, 2 + i).Value = RbestOfCrane(i)
            Next i

            'แสดงเวลางานของแต่ละงานที่ใช้ ในลำดับการทำงานที่ดีที่สุด
            For i = 1 To NumOfJob
                For j = 1 To NumOfJob
                    If Worksheets("EnterData").Cells(23, 3 + j).Value = Worksheets("Solve").Cells(15, 2 + i).Value Then
                        Worksheets("Solve").Cells(17, 2 + i).Value = Worksheets("EnterData").Cells(24, 3 + j).Value
                    End If
                Next j
            Next i

            TT = Round(T, 5)
            Worksheets("Solve").Cells(19, 4).Value = TT
            Worksheets("Solve").Cells(20, 4).Value = v
        End If
    ElseIf Snew > Scur Then
        K = 1.38E-23
        RanSA = Rnd
        Prob = Exp(-(Snew - Scur)) / (K * T)

        If RanSA <= Prob Then
            Scur = Snew
            RcurOfJob() = RnewOfJob()
            RcurOfCrane() = RnewOfCrane()

            Worksheets("Solve").Cells(28, 5).Value = Scur

            For i = 1 To NumOfJob
                Worksheets("Solve").Cells(26, 2 + i).Value = RcurOfJob(i)
                Worksheets("Solve").Cells(27, 2 + i).Value = RcurOfCrane(i)
            Next i
            TT = Round(T, 5)
            Worksheets("Solve").Cells(29, 4).Value = TT
            Worksheets("Solve").Cells(30, 4).Value = v
        End If
    End If
  Next v
  'ลดค่าอุณหภูมิด้วยค่าอัตราการเย็นตัว
  T = T * Trate
Loop

EndTime = Time
RunningTime = EndTime - StartTime
Worksheets("Solve").Cells(11, 7) = RunningTime
End Sub
```

รูปที่ ข.56 (ต่อ) แสดงคำสั่งเมื่อกดเริ่มการประมวลผล ของหน้าต่าง Microsoft Excel ในชีท Solve

7. คำสั่งเรียกใช้และคำสั่งใน Module

7.1 คำสั่งใน Module1

```
Function checkNumeric(KeyAscii)
'เป็นฟังก์ชันเรียกใช้ เพื่อเป็นการป้องกันไม่ให้กรอกข้อมูลที่ควรจะเป็นตัวเลขให้เป็นตัวอักษร
'เป็นการบังคับให้กรอกข้อมูลที่เป็นตัวเลขเท่านั้น

   If KeyAscii < 46 Or KeyAscii = 47 Or KeyAscii > 57 Then
      KeyAscii = 0
   End If

End Function
```

รูปที่ ข.57 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง checkNumeric

```
Function checkNumericForInt(KeyAscii)
'เป็นฟังก์ชันเรียกใช้ เพื่อเป็นการป้องกันไม่ให้กรอกข้อมูลที่ควรจะเป็นตัวเลขให้เป็นตัวอักษร
'เป็นการบังคับให้กรอกข้อมูลที่เป็นตัวเลขเท่านั้น และให้เป็นจำนวนเต็ม

   If KeyAscii < 48 Or KeyAscii > 57 Then
      KeyAscii = 0
   End If

End Function
```

รูปที่ ข.58 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง checkNumericForInt

```
Sub ClearSolve()
'เป็นฟังก์ชั่นเรียกใช้ เพื่อลบข้อมูลที่จำเป็น เพื่อการประมวลผลข้อมูลใหม่

   Worksheets("Solve").Cells(11, 7).ClearContents      'ลบเวลาที่ใช้ประมวลผล
   Worksheets("Solve").Cells(18, 5).Clear              'ลบเวลาการทำงานที่ดีที่สุด
   Worksheets("Solve").Cells(19, 4).Clear              'ลบอุณหภูมิการทำงานที่ดีที่สุด
   Worksheets("Solve").Cells(20, 4).Clear              'ลบรอบการหาคำตอบที่ดีที่สุด
   Worksheets("Solve").Cells(28, 5).Clear              'ลบเวลาการทำงาน ณ ปัจจุบัน
   Worksheets("Solve").Cells(29, 4).Clear              'ลบอุณหภูมิการทำงาน ณ ปัจจุบัน
   Worksheets("Solve").Cells(30, 4).Clear              'ลบรอบการหาคำตอบ ณ ปัจจุบัน
   Worksheets("Solve").Range("C14:AZ17").Clear         'ลบลำดับการทำงานที่ดีที่สุด
   Worksheets("Solve").Range("C25:AZ27").Clear         'ลบลำดับการทำงาน ณ ปัจจุบัน
   Worksheets("Solve").Range("A35:AZ500").Clear        'ลบรูปแบบทั้งหมดที่ใช้ในการหาคำตอบ

End Sub
```

รูปที่ ข.59 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง ClearSolve

```
Sub ClearAllTimeOfJob()
'เป็นฟังก์ชั่นเรียกใช้ เพื่อลบเวลาการทำงานของแต่ละงาน

   Worksheets("EnterData").Range("D23:AZ24").Clear

End Sub
```

รูปที่ ข.60 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง ClearAllTimeOfJob

```
Sub ClearData()
'เป็นฟังก์ชั่นเรียกใช้ เพื่อลบข้อมูลพื้นฐาน

    Worksheets("EnterData").Cells(1, 1).Clear              'ลบการนับที่ทดไว้
    Worksheets("EnterData").Cells(9, 7).Clear              'ลบจำนวนงาน
    Worksheets("EnterData").Cells(10, 7).Clear             'ลบจำนวนเครน
    Worksheets("EnterData").Cells(9, 15).Clear             'ลบอุณหภูมิเริ่มต้น
    Worksheets("EnterData").Cells(10, 15).Clear            'ลบอุณหภูมิสุดท้าย
    Worksheets("EnterData").Cells(11, 15).Clear            'ลบจำนวนรอบการอบอ่อน
    Worksheets("EnterData").Cells(12, 15).Clear            'ลบอัตราการเย็นตัว
    Worksheets("EnterData").Cells(13, 15).Clear            'ลบวิธีการอบอ่อน

End Sub
```

รูปที่ ข.61 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง ClearData

```
Sub PreparationToInputTime()
'เป็นฟังก์ชั่นเรียกใช้ เพื่อเตรียมตัวกรอกเวลาทำงานแต่ละงาน

    NumOfJob = Worksheets("EnterData").Cells(9, 7)

    For i = 1 To NumOfJob
        Worksheets("EnterData").Cells(23, 3 + i).Interior.Color = vbYellow
        Worksheets("EnterData").Cells(23, 3 + i).Borders.Weight = xlThin
        Worksheets("EnterData").Cells(23, 3 + i) = i
    Next i

    For i = 1 To NumOfJob
        Worksheets("EnterData").Cells(24, 3 + i).Interior.Color = vbYellow
        Worksheets("EnterData").Cells(24, 3 + i).Borders.Weight = xlThin
    Next i

End Sub
```

รูปที่ ข.62 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง PreparationToInputTime

```
Sub Newns()
'เป็นฟังก์ชั่นเรียกใช้ ในการสุ่มคำตอบแรก

NumOfJob = Worksheets("EnterData").Cells(9, 7)
NumOfCrane = Worksheets("EnterData").Cells(10, 7)

Dim iArr() As Integer
ReDim iArr(1 To NumOfJob)

    For i = 1 To NumOfJob
        iArr(i) = i
    Next i

    For i = NumOfJob To 1 Step -1
        R = Int(Rnd() * (i - 1)) + 1
        Temp = iArr(R)
        iArr(R) = iArr(i)
        iArr(i) = Temp
    Next i
```

รูปที่ ข.63 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง Newns

```
    For i = 1 To NumOfJob
        Worksheets("Solve").Cells(26, 2 + i) = iArr(i)
    Next i

    For i = 1 To NumOfJob
        Worksheets("Solve").Cells(27, 2 + i) = 0
    Next i

        R1 = Int((NumOfJob - 1) * Rnd() + 1)
        Worksheets("Solve").Cells(27, 2 + R1) = 1

      For i = 2 To NumOfCrane
        Do
          R1 = Int((NumOfJob - 1) * Rnd() + 1)
        Loop Until Worksheets("Solve").Cells(27, 2 + R1) = 0
        Worksheets("Solve").Cells(27, 2 + R1) = i
      Next i
      For i = 1 To NumOfJob
        If Worksheets("Solve").Cells(27, 2 + i) = 0 Then
            Worksheets("Solve").Cells(27, 2 + i) = Int((NumOfCrane * Rnd()) + 1)
        End If
      Next i
End Sub
```

รูปที่ ข.63 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง Newns

```
Sub SetFormatProcessing123()
'จัดรูปแบบของ cell ของคำตอบโดยใช้หลักการที่ 1,2,3
'คือ หลักการเลื่อนตำแหน่ง หลักการคงที่ 1 ตำแหน่ง และหลักการสลับที่

NumOfJob = Worksheets("EnterData").Cells(9, 7)

For i = 1 To NumOfJob
  'จัดสีและกรอบของคำตอบ ณ ปัจจุบัน และคำตอบที่ดีที่สุด
  Worksheets("Solve").Cells(14, 2 + i).Interior.Color = RGB(0, 255, 0)
  Worksheets("Solve").Cells(14, 2 + i).Borders.Weight = xlThin
  Worksheets("Solve").Cells(15, 2 + i).Interior.Color = RGB(0, 255, 0)
  Worksheets("Solve").Cells(15, 2 + i).Borders.Weight = xlThin
  Worksheets("Solve").Cells(16, 2 + i).Interior.Color = RGB(0, 255, 0)
  Worksheets("Solve").Cells(16, 2 + i).Borders.Weight = xlThin
  Worksheets("Solve").Cells(17, 2 + i).Interior.Color = RGB(0, 255, 0)
  Worksheets("Solve").Cells(17, 2 + i).Borders.Weight = xlThin
  Worksheets("Solve").Cells(25, 2 + i).Interior.Color = RGB(255, 255, 153)
  Worksheets("Solve").Cells(25, 2 + i).Borders.Weight = xlThin
  Worksheets("Solve").Cells(26, 2 + i).Interior.Color = RGB(255, 255, 153)
  Worksheets("Solve").Cells(26, 2 + i).Borders.Weight = xlThin
  Worksheets("Solve").Cells(27, 2 + i).Interior.Color = RGB(255, 255, 153)
  Worksheets("Solve").Cells(27, 2 + i).Borders.Weight = xlThin

  Worksheets("Solve").Cells(14, 2 + i) = i
  Worksheets("Solve").Cells(25, 2 + i) = i
Next I

  Worksheets("Solve").Cells(18, 5).Interior.Color = RGB(255, 0, 255)
  Worksheets("Solve").Cells(18, 5).Borders.Weight = xlThin
  Worksheets("Solve").Cells(19, 4).Interior.Color = RGB(255, 153, 204)
  Worksheets("Solve").Cells(19, 4).Borders.Weight = xlThin
```

รูปที่ ข.64 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง SetFormatProcessing123

```
    Worksheets("Solve").Cells(20, 4).Interior.Color = RGB(255, 153, 204)
    Worksheets("Solve").Cells(20, 4).Borders.Weight = xlThin
    Worksheets("Solve").Cells(28, 5).Interior.Color = RGB(255, 153, 0)
    Worksheets("Solve").Cells(28, 5).Borders.Weight = xlThin
    Worksheets("Solve").Cells(29, 4).Interior.Color = RGB(204, 153, 255)
    Worksheets("Solve").Cells(29, 4).Borders.Weight = xlThin
    Worksheets("Solve").Cells(30, 4).Interior.Color = RGB(204, 153, 255)
    Worksheets("Solve").Cells(30, 4).Borders.Weight = xlThin

For i = 1 To 2 * (NumOfJob - 1)
    For j = 1 To NumOfJob
        'จัดสีและกรอบของคำรูปแบบคำตอบ ตามหลักการที่ 1,2,3
        'ซึ่งมี คำตอบ 2*(NumofJob - 1) คำตอบ

        Worksheets("Solve").Cells(34 + i, 1) = i
        Worksheets("Solve").Cells(34 + i, 2) = "งานที่"
        Worksheets("Solve").Cells(34 + i, 2).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + i, 2 + j).Interior.Color = RGB(204, 153, 255)
        Worksheets("Solve").Cells(34 + i, 2 + j).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2) = "เครื่องที่"
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + j).Interior.Color = RGB(204, 153, 255)
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + j).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + 1 + 2 * (NumOfJob - 1), 3).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + 1 + 2 * (NumOfJob - 1), 3).Interior.Color = RGB(23, 200, 238)
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + NumOfJob + 1).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + NumOfJob + 1).Interior.Color = RGB(250, 150, 80)
    Next j
Next i

End Sub
```

รูปที่ ข.64 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง SetFormatProcessing123

```
Sub SetFormatProcessing4()
'จัดรูปแบบของ cell ของคำตอบโดยใช้หลักการที่ 4 หลักการพิจารณาที่ 2 ตำแหน่ง
NumOfJob = Worksheets("EnterData").Cells(9, 7)

For i = 1 To NumOfJob
    'จัดสีและกรอบของคำตอบ ณ ปัจจุบัน และคำตอบที่ดีที่สุด
    Worksheets("Solve").Cells(14, 2 + i).Interior.Color = RGB(0, 255, 0)
    Worksheets("Solve").Cells(14, 2 + i).Borders.Weight = xlThin
    Worksheets("Solve").Cells(15, 2 + i).Interior.Color = RGB(0, 255, 0)
    Worksheets("Solve").Cells(15, 2 + i).Borders.Weight = xlThin
    Worksheets("Solve").Cells(16, 2 + i).Interior.Color = RGB(0, 255, 0)
    Worksheets("Solve").Cells(16, 2 + i).Borders.Weight = xlThin
    Worksheets("Solve").Cells(17, 2 + i).Interior.Color = RGB(0, 255, 0)
    Worksheets("Solve").Cells(17, 2 + i).Borders.Weight = xlThin
    Worksheets("Solve").Cells(25, 2 + i).Interior.Color = RGB(255, 255, 153)
    Worksheets("Solve").Cells(25, 2 + i).Borders.Weight = xlThin
    Worksheets("Solve").Cells(26, 2 + i).Interior.Color = RGB(255, 255, 153)
    Worksheets("Solve").Cells(26, 2 + i).Borders.Weight = xlThin
    Worksheets("Solve").Cells(27, 2 + i).Interior.Color = RGB(255, 255, 153)
    Worksheets("Solve").Cells(27, 2 + i).Borders.Weight = xlThin

    Worksheets("Solve").Cells(14, 2 + i) = i
    Worksheets("Solve").Cells(25, 2 + i) = i
Next i
```

รูปที่ ข.65 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง SetFormatProcessing4

```
    Worksheets("Solve").Cells(18, 5).Interior.Color = RGB(255, 0, 255)
    Worksheets("Solve").Cells(18, 5).Borders.Weight = xlThin
    Worksheets("Solve").Cells(19, 4).Interior.Color = RGB(255, 153, 204)
    Worksheets("Solve").Cells(19, 4).Borders.Weight = xlThin
    Worksheets("Solve").Cells(20, 4).Interior.Color = RGB(255, 153, 204)
    Worksheets("Solve").Cells(20, 4).Borders.Weight = xlThin
    Worksheets("Solve").Cells(28, 5).Interior.Color = RGB(255, 153, 0)
    Worksheets("Solve").Cells(28, 5).Borders.Weight = xlThin
    Worksheets("Solve").Cells(29, 4).Interior.Color = RGB(204, 153, 255)
    Worksheets("Solve").Cells(29, 4).Borders.Weight = xlThin
    Worksheets("Solve").Cells(30, 4).Interior.Color = RGB(204, 153, 255)
    Worksheets("Solve").Cells(30, 4).Borders.Weight = xlThin

For i = 1 To (NumOfJob - 1)
    For j = 1 To NumOfJob
        'จัดสีและกรอบของคำรูปแบบคำตอบ ตามหลักการที่ 4
        'ซึ่งมี คำตอบ (NumofJob - 1) คำตอบ

        Worksheets("Solve").Cells(34 + i, 1) = i
        Worksheets("Solve").Cells(34 + i, 2) = "งานที่"
        Worksheets("Solve").Cells(34 + i, 2).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + i, 2 + j).Interior.Color = RGB(204, 153, 255)
        Worksheets("Solve").Cells(34 + i, 2 + j).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2) = "เครนที่"
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + j).Interior.Color = RGB(204, 153, 255)
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + j).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + 1 + (NumOfJob - 1), 3).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + 1 + (NumOfJob - 1), 3).Interior.Color = RGB(23, 200, 238)
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + NumOfJob + 1).Borders.Weight = xlThin
        Worksheets("Solve").Cells(34 + i, 2 + NumOfJob + 2 + NumOfJob + 1).Interior.Color = RGB(250, 150, 80)
    Next j

Next i

End Sub
```

รูปที่ ข.65 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง SetFormatProcessing4

```
Public Sub MakespanIntervalFirstAns()
'เป็นฟังก์ชันเรียกใช้ เพื่อหาค่า Makespan ของคำตอบเริ่มต้น

Dim STI() As Double
Dim FTI() As Double
Dim STJ() As Double
Dim FTJ() As Double
Dim OrderInCrane() As Integer
Dim Now() As Integer
Dim PTime() As Double
Dim Order() As Integer
Dim Crane() As Integer

NumOfJob = Worksheets("EnterData").Cells(9, 7)
NumOfCrane = Worksheets("EnterData").Cells(10, 7)

ReDim STJ(0 To NumOfCrane, 0 To NumOfJob)
ReDim FTJ(0 To NumOfCrane, 0 To NumOfJob)
```

รูปที่ ข.66 แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns

```
ReDim STJ(0 To NumOfCrane, 0 To NumOfJob)
ReDim FTJ(0 To NumOfCrane, 0 To NumOfJob)
ReDim OrderInCrane(1 To NumOfCrane, 1 To NumOfJob)
ReDim Now(1 To NumOfCrane)
ReDim PTime(1 To NumOfJob)
ReDim Order(1 To NumOfJob)
ReDim Crane(1 To NumOfJob)

For i = 1 To NumOfJob
    PTime(i) = Worksheets("EnterData").Cells(24, 3 + i)
Next i

For i = 1 To NumOfJob
   Order(i) = Worksheets("Solve").Cells(26, 2 + i)
   Crane(i) = Worksheets("Solve").Cells(27, 2 + i)

Next i

Dim CountCrane() As Integer 'Count How many job is assigned to a crane
ReDim CountCrane(1 To NumOfCrane)
Dim ContInt As Integer 'Count how many interval for a crane at present
Dim Interval() As Integer
ReDim Interval(1 To NumOfJob)

Dim IntTime() As Integer
ReDim IntTime(0 To NumOfCrane, 0 To NumOfJob)
Const UpBound = 10000

For i = 0 To NumOfCrane
   For j = 0 To NumOfJob
      STJ(i, j) = 0
      FTJ(i, j) = 0
   Next j
Next i

For i = 1 To NumOfCrane
'   Interval(i) = 0
   For j = 1 To NumOfJob
      IntTime(i, j) = 0
   Next j
Next i

For i = 1 To NumOfCrane
   CountCrane(i) = 0
Next i
For i = 1 To NumOfJob
   CountCrane(Worksheets("Solve").Cells(27, 2 + i)) = CountCrane(Worksheets("Solve").Cells(27, 2 + i)) + 1
   OrderInCrane(Worksheets("Solve").Cells(27, 2 + i), CountCrane(Worksheets("Solve").Cells(27, 2 + i))) = _
     Worksheets("Solve").Cells(26, 2 + i)
Next i

For i = 1 To NumOfJob
```

รูปที่ ข.66 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns

```
For j = 1 To NumOfCrane
   For K = 1 To NumOfJob
      STJ(j, K) = 0
      FTJ(j, K) = 0
   Next K
Next j

For j = 1 To NumOfCrane
   Interval(j) = 0
   For K = 1 To NumOfJob
      IntTime(j, K) = 0
   Next K
Next j

'For each job we set the starting time at the finished time of the last job
'that finish in the crane
Now(Crane(i)) = Now(Crane(i)) + 1
STJ(Crane(i), Now(Crane(i))) = FTJ(Crane(i), Now(Crane(i)) - 1)
FTJ(Crane(i), Now(Crane(i))) = STJ(Crane(i), Now(Crane(i))) + PTime(Order(i))

For j = 1 To NumOfCrane
   If Crane(i) < j Then
      ContInt = 0
      For K = 1 To Now(j)

         If FTJ(j, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
            If STJ(j, K) < FTJ(Crane(i), Now(Crane(i)) - 1) Then
               If OrderInCrane(j, K) > Order(i) Then
                  Interval(j) = Interval(j) + 1
                  STJ(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
                  ContInt = 1
                  IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(Crane(i), Now(Crane(i)) - 1) + FTJ(j, K)
                  If K = Now(j) Then
                     FTJ(j, Interval(j)) = UpBound
                     IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
                  End If
               ElseIf OrderInCrane(j, K) < Order(i) Then
                  If K = Now(j) Then
                     Interval(j) = Interval(j) + 1
                     STJ(j, Interval(j)) = FTJ(j, K)
                     FTJ(j, Interval(j)) = UpBound
                     IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
                  End If
               End If
            ' End If
            ElseIf STJ(j, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
               If FTJ(j, K - 1) < FTJ(Crane(i), Now(Crane(i)) - 1) Then
                  If OrderInCrane(j, K) > Order(i) Then
                     Interval(j) = Interval(j) + 1
                     STJ(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
                     ContInt = 1
                     IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(Crane(i), Now(Crane(i)) - 1) + FTJ(j, K)
                     If K = Now(j) Then
                        FTJ(j, Interval(j)) = UpBound
                        IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
                     End If
```

รูปที่ ข.66 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns

```
        ElseIf OrderInCrane(j, K) < Order(i) Then
            If STJ(j, K) > FTJ(Crane(i), Now(Crane(i)) - 1) Then
                Interval(j) = Interval(j) + 1
                STK(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
                IntTime(j, Interval(j)) = _
                                STJ(j, K) - FTJ(Crane(i), Now(Crane(i)) - 1)
                FTK(j, Interval(j)) = STK(j, Interval(j)) + IntTime(j, Interval(j))
            End If
            ContInt = 0
            If K = Now(j) Then
                Interval(j) = Interval(j) + 1
                STK(j, Interval(j)) = FTJ(j, K)
                FTK(j, Interval(j)) = UpBound
                IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
            End If
        End If
    ElseIf FTJ(j, K - 1) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
        If OrderInCrane(j, K) > Order(i) Then
            If ContInt = 0 Then
                Interval(j) = Interval(j) + 1
                STK(j, Interval(j)) = FTJ(j, K - 1)
                ContInt = 1
                IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + FTJ(j, K)
                If K = Now(j) Then
                    FTK(j, Interval(j)) = UpBound
                    IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
                End If
            ElseIf ContInt = 1 Then
                IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + FTJ(j, K)
                If K = Now(j) Then
                    FTK(j, Interval(j)) = UpBound
                    IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
                End If
            End If
        ElseIf OrderInCrane(j, K) < Order(i) Then
            If STJ(j, K) > FTJ(j, K - 1) Then
                If ContInt = 0 Then
                    Interval(j) = Interval(j) + 1
                    STK(j, Interval(j)) = FTJ(j, K - 1)
                    FTK(j, Interval(j)) = STJ(j, K)
                    IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
                ElseIf ContInt = 1 Then
                    IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + STJ(j, K)
                    FTK(j, Interval(j)) = STK(j, Interval(j)) + IntTime(j, Interval(j))
                    ContInt = 0
                End If
                If K = Now(j) Then
                    Interval(j) = Interval(j) + 1
                    STK(j, Interval(j)) = FTJ(j, K)
                    FTK(j, Interval(j)) = UpBound
                    IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
                End If
            ElseIf STJ(j, K) = FTJ(j, K - 1) Then
                If ContInt = 1 Then
                    FTK(j, Interval(j)) = STJ(j, K)
                    IntTime(j, Interval(j)) = FTJ(j, K) - STJ(j, K)
```

รูปที่ ข.66 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns

```
                    ContInt = 0
                End If
                If K = Now(j) Then
                    Interval(j) = Interval(j) + 1
                    STK(j, Interval(j)) = FTJ(j, K)
                    FTK(j, Interval(j)) = UpBound
                    IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
                End If
            End If
        End If
    End If
End If
Next K

ElseIf Crane(i) > J Then
    ContInt = 0
    For K = 1 To Now(J)
        If FTJ(J, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
            If STJ(J, K) < FTJ(Crane(i), Now(Crane(i)) - 1) Then
                If OrderInCrane(j, K) < Order(i) Then
                    Interval(j) = Interval(j) + 1
                    STK(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
                    ContInt = 1
                    IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(Crane(i), Now(Crane(i)) - 1) + FTJ(J, K)
                    If K = Now(j) Then
                        FTK(j, Interval(j)) = UpBound
                        IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
                    End If
                ElseIf OrderInCrane(j, K) > Order(i) Then
                    If K = Now(j) Then
                        Interval(j) = Interval(j) + 1
                        STK(j, Interval(j)) = FTJ(j, K)
                        FTK(j, Interval(j)) = UpBound
                        IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
                    End If
                End If

            ElseIf STJ(j, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
                If FTJ(j, K - 1) < FTJ(Crane(i), Now(Crane(i)) - 1) Then
                    If OrderInCrane(j, K) < Order(i) Then
                        Interval(j) = Interval(j) + 1
                        STK(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
                        ContInt = 1
                        IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(Crane(i), Now(Crane(i)) - 1) + FTJ(J, K)
                        If K = Now(j) Then
                            FTK(j, Interval(j)) = UpBound
                            IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
                        End If
                    ElseIf OrderInCrane(j, K) > Order(i) Then
                        If STJ(j, K) > FTJ(Crane(i), Now(Crane(i)) - 1) Then
                            Interval(j) = Interval(j) + 1
                            STK(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
                            IntTime(j, Interval(j)) = STJ(J, K) - FTJ(Crane(i), Now(Crane(i)) - 1)
                            FTK(j, Interval(j)) = STK(j, Interval(j)) + IntTime(j, Interval(j))
                        End If
                        ContInt = 0
```

รูปที่ ข.66 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns

```
            If K = Now(j) Then
               Interval(j) = Interval(j) + 1
               STJ(j, Interval(j)) = FTJ(j, K)
               FTJ(j, Interval(j)) = UpBound
               IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
            End If
         End If
      ElseIf FTJ(j, K - 1) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
         If OrderInCrane(j, K) < Order(i) Then
            If ContInt = 0 Then
               Interval(j) = Interval(j) + 1
               STJ(j, Interval(j)) = FTJ(j, K - 1)
               ContInt = 1
               IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + FTJ(j, K)
               If K = Now(j) Then
                  FTJ(j, Interval(j)) = UpBound
                  IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
               End If
            ElseIf ContInt = 1 Then
               IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + FTJ(j, K)
               If K = Now(j) Then
                  FTJ(j, Interval(j)) = UpBound
                  IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
               End If
            End If
         ElseIf OrderInCrane(j, K) > Order(i) Then
            If STJ(j, K) > FTJ(j, K - 1) Then
               If ContInt = 0 Then
                  Interval(j) = Interval(j) + 1
                  STJ(j, Interval(j)) = FTJ(j, K - 1)
                  FTJ(j, Interval(j)) = STJ(j, K)
                  IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
               ElseIf ContInt = 1 Then
                  IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + STJ(j, K)
                  FTJ(j, Interval(j)) = STJ(j, Interval(j)) + IntTime(j, Interval(j))
                  ContInt = 0
               End If
               If K = Now(j) Then
                  Interval(j) = Interval(j) + 1
                  STJ(j, Interval(j)) = FTJ(j, K)
                  FTJ(j, Interval(j)) = UpBound
                  IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
               End If
            ElseIf STJ(j, K) = FTJ(j, K - 1) Then
               If ContInt = 1 Then
                  FTJ(j, Interval(j)) = STJ(j, K)
                  IntTime(j, Interval(j)) = FTJ(j, K) - STJ(j, K)
                  ContInt = 0
               End If
               If K = Now(j) Then
                  Interval(j) = Interval(j) + 1
                  STJ(j, Interval(j)) = FTJ(j, K)
                  FTJ(j, Interval(j)) = UpBound
                  IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
               End If
```

รูปที่ ข.66 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns

```
                    End If
                End If
            End If
        End If
    End If
  Next K
 End If
Next j
'Now we take care of those cranes where there is no job at all
For j = 1 To NumOfCrane

    If Interval(j) = 0 Then
        Interval(j) = 1
        STJ(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
        FTJ(j, Interval(j)) = UpBound
        IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
    End If
Next j
'Now we check for the starting time of the job
Dim STCur As Integer
Dim FTCur As Integer
Dim Ok As Integer

For j = 1 To NumOfCrane
  If j <> Crane(i) Then
    For K = 1 To Interval(j)
      Ok = 0
      If FTJ(j, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
        If STJ(j, K) <= FTJ(Crane(i), Now(Crane(i)) - 1) Then
          STCur = FTJ(Crane(i), Now(Crane(i)) - 1)
          FTCur = STCur + PTime(Order(i))
        Else
          STCur = STJ(j, K)
          FTCur = STCur + PTime(Order(i))
        End If
        If FTCur <= FTJ(j, K) Then
          For m = 1 To NumOfCrane
            If m <> j And m <> Crane(i) Then
              Ok = 0

                For n = 1 To Interval(m)
                  If STJ(m, n) <= STCur And FTJ(m, n) >= FTCur Then
                    Ok = 1
                    Exit For
                  End If
                Next n
                If Ok = 0 Then Exit For
              End If
            Next m
          End If
          If Ok = 1 Then
            STJ(Crane(i), Now(Crane(i))) = STCur
            FTJ(Crane(i), Now(Crane(i))) = STJ(Crane(i), Now(Crane(i))) + PTime(Order(i))
            Exit For
          End If
        End If
      Next K
```

รูปที่ ข.66 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns

```
            If Ok = 1 Then Exit For
        End If
    Next J
Next i
'Check for makespan
Dim Makespan As Double
For i = 1 To NumOfCrane
    For J = 1 To Now(i)
        If FTJ(I, j) >= Makespan Then
            Makespan = FTJ(I, j)
        End If
    Next J
Next i
Worksheets("Solve").Cells(28, 5) = Makespan

End Sub
```

รูปที่ ข.66 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module1 คำสั่ง MakespanIntervalFirstAns

7.2 คำสั่งใน Module2

```
Public Sub ReadPTime()
'เป็นฟังก์ชันเรียกใช้ในการอ่านและจดจำเวลาของงานแต่ละงาน

Dim NumOfCrane As Integer
Dim NumOfJob As Integer
Dim PTime() As Double

NumOfJob = Worksheets("EnterData").Cells(9, 7)
NumOfCrane = Worksheets("EnterData").Cells(10, 7)
ReDim PTime(1 To NumOfJob)

For i = 1 To NumOfJob
    PTime(i) = Worksheets("EnterData").Cells(24, 3 + i)
Next i

End Sub
```

รูปที่ ข.67 แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง ReadPTime

```
Sub NewNS1()
'เป็นฟังก์ชันเรียกใช้การทำ Neighbourhood search ในการประมวลผล โดยใช้หลักการ 1.หลักการเลื่อนตำแหน่ง

Dim NumOfJob As Integer, NumOfCrane As Integer, Temp As Integer, Seq As Integer
Dim RanOfJob As Integer, RanOfCrane As Integer
Dim StOfJob As Integer, LastlOfJob As Integer, MidOfJob As Integer
Dim StOfCrane As Integer, LastOfCrane As Integer, MidOfCrane As Integer
Dim ArrOfJob() As Integer
Dim ArrOfCrane() As Integer

' อธิบายตัวแปร
    'numofjob    แสดง จำนวนของงาน        numofcrane  แสดง จำนวนของเครน
    'temp        แสดง ตัวแปรที่เก็บค่าได้  seq         แสดง ตำแหน่งที่ของงานหรือเครน
    'ranofjob    แสดง งานที่สุ่มได้        ranofcrane  แสดง เครนที่สุ่มได้
    'stofjob     แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งแรก
```

รูปที่ ข.68 แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS1

```
    'stofcrane      แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งแรก
    'lastofjob      แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งสุดท้าย
    'lastofcrane    แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งสุดท้าย
    'midofjob       แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลาง
    'midofcrane     แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลาง
    'arrofjob()     แสดง ตำแหน่งของงาน
    'arrofcrane()   แสดง ตำแหน่งของเครน

'รับจำนวนงานและจำนวนเครน
NumOfJob = Worksheets("EnterData").Cells(9, 7).Value
NumOfCrane = Worksheets("EnterData").Cells(10, 7).Value

ReDim ArrOfJob(1 To NumOfJob) As Integer
ReDim ArrOfCrane(1 To NumOfJob) As Integer

'ใส่งานและเครนในรูปแบบคำตอบทั้งหมด เพื่อเตรียมการเลื่อนตำแหน่ง
For i = 1 To NumOfJob
    ArrOfJob(i) = Worksheets("Solve").Cells(26, 2 + i)
    ArrOfCrane(i) = Worksheets("solve").Cells(27, 2 + i)
Next i

For i = 1 To 2 * (NumOfJob - 1)
    For j = 1 To NumOfJob
        Worksheets("solve").Cells(34 + i, 2 + j) = ArrOfJob(j)
        Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j) = ArrOfCrane(j)
    Next j
Next i

'การจัดลำดับใหม่ของงาน โดยการเลื่อนตำแหน่ง
For i = 1 To 2 * (NumOfJob - 1)

    For j = 1 To NumOfJob
        ArrOfJob(j) = Worksheets("solve").Cells(34 + i, 2 + j)
    Next j

    'งาน พิจารณาที่ตำแหน่งแรก
    If i = 1 Then
        StOfJob = Int((NumOfJob - 2 + 1) * Rnd + 2)
        Temp = ArrOfJob(i)

        For j = 1 To StOfJob - 1
            ArrOfJob(j) = ArrOfJob(j + 1)
        Next j

        ArrOfJob(StOfJob) = Temp

    'งาน พิจารณาที่ตำแหน่งสุดท้าย
    ElseIf i = (2 * (NumOfJob - 1)) Then
        LastOfJob = Int(((NumOfJob - 1) - 1 + 1) * Rnd + 1)
        Temp = ArrOfJob(LastOfJob)

        For j = LastOfJob To NumOfJob - 1
            ArrOfJob(j) = ArrOfJob(j + 1)
        Next j

        ArrOfJob(NumOfJob) = Temp
```

รูปที่ ข.68 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS1

```
    'งาน พิจารณาที่ตำแหน่งตรงกลาง
    Else
        Seq = (i \ 2) + 1

        'งานตำแหน่งที่พิจารณาสลับที่ไปทางด้านซ้าย
        If (i Mod 2) = 0 Then
            MidOfJob = Int(((Seq - 1) - 1 + 1) * Rnd + 1)
            Temp = ArrOfJob(Seq)

            For J = MidOfJob To Seq - 1
                ArrOfJob(Seq + MidOfJob - j) = ArrOfJob(Seq + MidOfJob - J - 1)
            Next J

            ArrOfJob(MidOfJob) = Temp

        'งานตำแหน่งที่พิจารณาสลับที่ไปทางด้านขวา
        Else
            MidOfJob = Int((NumOfJob - (Seq + 1) + 1) * Rnd + (Seq + 1))
            Temp = ArrOfJob(Seq)

            For j = Seq To MidOfJob - 1
                ArrOfJob(j) = ArrOfJob(J + 1)
            Next j

            ArrOfJob(MidOfJob) = Temp

        End If

    End If

    For j = 1 To NumOfJob
        Worksheets("solve").Cells(34 + i, 2 + j) = ArrOfJob(j)
    Next J

Next i

'จัดลำดับใหม่ของเครน โดยการเลื่อนตำแหน่ง

For i = 1 To 2 * (NumOfJob - 1)

    For j = 1 To NumOfJob
        ArrOfCrane(j) = Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j)
    Next J

    'เครน พิจารณาที่ตำแหน่งแรก
    If i = 1 Then
        StOfCrane = Int((NumOfJob - 2 + 1) * Rnd + 2)
        Temp = ArrOfCrane(i)

        For j = 1 To StOfCrane - 1
            ArrOfCrane(j) = ArrOfCrane(j + 1)
        Next j

        ArrOfCrane(StOfCrane) = Temp
```

รูปที่ ข.68 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS1

```
        'เครนตำแหน่งที่พิจารณาสลับที่ไปทางด้านขวา
        Else
            MidOfCrane = Int((NumOfJob - (Seq + 1) + 1) * Rnd + (Seq + 1))
            Temp = ArrOfCrane(Seq)

            For J = Seq To MidOfCrane - 1
                ArrOfCrane(j) = ArrOfCrane(J + 1)
            Next J

            ArrOfCrane(MidOfCrane) = Temp

        End If

    End If

    For j = 1 To NumOfJob
        Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j) = ArrOfCrane(j)
    Next j

Next i
End Sub
```

รูปที่ ข.68 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS1

```
Sub NewNS2()
'เป็นฟังก์ชันเรียกใช้การทำ Neighbourhood search ในการประมวลผล โดยใช้หลักการ 2.หลักการลงที่ 1 ตำแหน่ง

Dim NumOfJob As Integer, NumOfCrane As Integer, Temp As Integer, Seq As Integer
Dim RanOfJob As Integer, RanOfCrane As Integer
Dim StOfJob As Integer, LastOfJob As Integer, MidOfJob As Integer
Dim StOfCrane As Integer, LastOfCrane As Integer, MidOfCrane As Integer
Dim ArrOfJob() As Integer
Dim ArrOfCrane() As Integer
' อธิบายตัวแปร
    'numofjob    แสดง จำนวนของงาน        numofcrane  แสดง จำนวนของเครน
    'temp        แสดง ตัวแปรที่เก็บค่าได้  seq         แสดง ตำแหน่งที่ของงานหรือเครน
    'ranofjob    แสดง งานที่สุ่มได้        ranofcrane  แสดง เครนที่สุ่มได้
    'stofjob     แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งแรก
    'stofcrane   แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งแรก
    'lastofjob   แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งสุดท้าย
    'lastofcrane แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งสุดท้าย
    'midofjob    แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลาง
    'midofcrane  แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลาง
    'arrofjob()  แสดง ตำแหน่งของงาน
    'arrofcrane() แสดง ตำแหน่งของเครน

'รับจำนวนงานและจำนวนเครน
NumOfJob = Worksheets("EnterData").Cells(9, 7).Value
NumOfCrane = Worksheets("EnterData").Cells(10, 7).Value

ReDim ArrOfJob(1 To NumOfJob) As Integer
ReDim ArrOfCrane(1 To NumOfJob) As Integer

'ใส่งานและเครนในรูปแบบคำตอบทั้งหมด เพื่อเตรียมการเลื่อนตำแหน่ง โดยการลงที่ 1 ตำแหน่ง
For i = 1 To NumOfJob
    ArrOfJob(i) = Worksheets("Solve").Cells(26, 2 + i)
    ArrOfCrane(i) = Worksheets('solve").Cells(27, 2 + i)
Next i
```

รูปที่ ข.69 แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS2

```
For i = 1 To 2 * (NumOfJob - 1)
   For j = 1 To NumOfJob
      Worksheets("solve").Cells(34 + i, 2 + j) = ArrOfJob(j)
      Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j) = ArrOfCrane(j)
   Next j
Next i

'การจัดลำดับใหม่ของงาน เพื่อเลื่อนตำแหน่ง โดยการคงที่ 1 ตำแหน่ง
For i = 1 To 2 * (NumOfJob - 1)

   For j = 1 To NumOfJob
      ArrOfJob(j) = Worksheets("solve").Cells(34 + i, 2 + j)
   Next j

   'งาน พิจารณาที่ตำแหน่งแรก
   If i = 1 Then
      StOfJob = Int((NumOfJob - 2 + 1) * Rnd + 2)
      '== นำเอาตำแหน่งที่สุ่มไปสลับกับตำแหน่งสุดท้ายแล้วเลื่อน ==
      If StOfJob <> NumOfJob Then
         Temp = ArrOfJob(StOfJob)

         For j = StOfJob To NumOfJob - 1
            ArrOfJob(j) = ArrOfJob(j + 1)
         Next j

         ArrOfJob(NumOfJob) = Temp
      End If

      '== สลับและเลื่อนตำแหน่งจากตำแหน่งแรกไปตำแหน่งก่อนสุดท้าย ==
      Temp = ArrOfJob(1)

      For j = 1 To NumOfJob - 2
         ArrOfJob(j) = ArrOfJob(j + 1)
      Next j

      ArrOfJob(NumOfJob - 1) = Temp

      '== นำเอาตำแหน่งที่สุดท้ายไปสลับกับตำแหน่งที่สุ่มได้แล้วเลื่อน ==
      If StOfJob <> NumOfJob Then
         Temp = ArrOfJob(NumOfJob)

         For j = StOfJob To NumOfJob - 1
            ArrOfJob(NumOfJob + StOfJob - j) = ArrOfJob(NumOfJob + StOfJob - j - 1)
         Next j

         ArrOfJob(StOfJob) = Temp
      End If

   'งาน พิจารณาที่ตำแหน่งสุดท้าย
   ElseIf i = (2 * (NumOfJob - 1)) Then
      LastOfJob = Int(((NumOfJob - 1) - 1 + 1) * Rnd + 1)

      '== นำเอาตำแหน่งที่สุ่มไปสลับกับตำแหน่งสุดท้ายแล้วเลื่อน ==
      Temp = ArrOfJob(LastOfJob)

      For j = LastOfJob To NumOfJob - 1
         ArrOfJob(j) = ArrOfJob(j + 1)
      Next j
```

รูปที่ ข.69 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS2

```
    ArrOfJob(NumOfJob) = Temp

    '== สลับและเลื่อนตำแหน่งจากตำแหน่งแรกไปตำแหน่งก่อนสุดท้าย ==
    Temp = ArrOfJob(NumOfJob - 1)

    For J = 1 To NumOfJob - 1 - 1
       ArrOfJob(NumOfJob - 1 + 1 - j) = ArrOfJob(NumOfJob - 1 + 1 - J - 1)
    Next J

    ArrOfJob(1) = Temp

    '== นำเอาตำแหน่งที่สุดท้ายไปสลับกับตำแหน่งที่สุ่มได้แล้วเลื่อน ==
    Temp = ArrOfJob(NumOfJob)

    For j = LastOfJob To NumOfJob - 1
      ArrOfJob(NumOfJob + LastOfJob - j) = ArrOfJob(NumOfJob + LastOfJob - j - 1)
    Next j

    ArrOfJob(LastOfJob) = Temp

'งาน พิจารณาที่ตำแหน่งตรงกลาง
Else
    Seq = (i \ 2) + 1

    If (i <> 2) Or (i <> (2 * (NumOfJob - 1)) - 1) Then
    'งานตำแหน่งที่พิจารณาสลับที่ไปทางด้านซ้าย
       If (i Mod 2) = 0 Then
          MidOfJob = Int(((Seq - 1) - 1 + 1) * Rnd + 1)

          '== นำเอาตำแหน่งที่สุ่มไปสลับกับตำแหน่งสุดท้ายแล้วเลื่อน ==
          Temp = ArrOfJob(MidOfJob)

          For J = MidOfJob To NumOfJob - 1
             ArrOfJob(j) = ArrOfJob(J + 1)
          Next J

          ArrOfJob(NumOfJob) = Temp

          '== สลับและเลื่อนตำแหน่งจากตำแหน่งแรกไปตำแหน่งที่พิจารณา ==
          Temp = ArrOfJob(Seq - 1)

          For J = 1 To Seq - 2
             ArrOfJob(Seq - 1 + 1 - j) = ArrOfJob(Seq - 1 + 1 - J - 1)
          Next J

          ArrOfJob(1) = Temp

          '== นำเอาตำแหน่งที่สุดท้ายไปสลับกับตำแหน่งที่สุ่มได้แล้วเลื่อน ==
          Temp = ArrOfJob(NumOfJob)

          For j = MidOfJob To NumOfJob - 1
             ArrOfJob(NumOfJob + MidOfJob - j) = ArrOfJob(NumOfJob + MidOfJob - J - 1)
          Next J

          ArrOfJob(MidOfJob) = Temp

       'งานตำแหน่งที่พิจารณาสลับที่ไปทางด้านขวา
```

รูปที่ ข.69 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS2

```
        Else
            MidOfJob = Int((NumOfJob - (Seq + 1) + 1) * Rnd + (Seq + 1))

            '== นำเอาตำแหน่งที่สุ่มไปสลับกับตำแหน่งสุดท้ายแล้วเลื่อน ==
            Temp = ArrOfJob(MidOfJob)

            For J = MidOfJob To NumOfJob - 1
                ArrOfJob(j) = ArrOfJob(J + 1)
            Next j

            ArrOfJob(NumOfJob) = Temp

            '== สลับและเลื่อนตำแหน่งจากตำแหน่งแรกไปตำแหน่งที่พิจารณา ==

            Temp = ArrOfJob(Seq)

            For J = Seq To NumOfJob - 2
                ArrOfJob(j) = ArrOfJob(J + 1)
            Next J
            ArrOfJob(NumOfJob - 1) = Temp

            '== นำเอาตำแหน่งที่สุดท้ายไปสลับกับตำแหน่งที่สุ่มได้แล้วเลื่อน ==
            Temp = ArrOfJob(NumOfJob)

            For j = MidOfJob To NumOfJob - 1
                ArrOfJob(NumOfJob + MidOfJob - j) = ArrOfJob(NumOfJob + MidOfJob - j - 1)
            Next j

            ArrOfJob(MidOfJob) = Temp

        End If

    Else
        If i = 2 Then
            MidOfJob = 1
        Else
            MidOfJob = NumOfJob - 1
        End If
    End If

    For j = 1 To NumOfJob
        Worksheets("solve").Cells(34 + i, 2 + j) = ArrOfJob(j)
    Next j

Next i

'จัดลำดับใหม่ของเครน

For i = 1 To 2 * (NumOfJob - 1) '

    For j = 1 To NumOfJob
        ArrOfCrane(j) = Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j)
    Next j

    'เครน พิจารณาที่ตำแหน่งแรก
    If i = 1 Then
        StOfCrane = Int((NumOfJob - 2 + 1) * Rnd + 2)
```

รูปที่ ข.69 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS2

```
    '== นำเอาตำแหน่งที่สุ่มไปสลับกับตำแหน่งสุดท้ายแล้วเลื่อน ==
    If StOfCrane <> NumOfJob Then
       Temp = ArrOfCrane(StOfCrane)

       For j = StOfCrane To NumOfJob - 1
          ArrOfCrane(j) = ArrOfCrane(j + 1)
       Next j

       ArrOfCrane(NumOfJob) = Temp
    End If

    '== สลับและเลื่อนตำแหน่งจากตำแหน่งแรกไปตำแหน่งก่อนสุดท้าย ==
    Temp = ArrOfCrane(1)

    For j = 1 To NumOfJob - 2
       ArrOfCrane(j) = ArrOfCrane(j + 1)
    Next j

    ArrOfCrane(NumOfJob - 1) = Temp

    '== นำเอาตำแหน่งที่สุดท้ายไปสลับกับตำแหน่งที่สุ่มได้แล้วเลื่อน ==
    If StOfCrane <> NumOfJob Then
       Temp = ArrOfCrane(NumOfJob)

       For j = StOfCrane To NumOfJob - 1
          ArrOfCrane(NumOfJob + StOfCrane - j) = ArrOfCrane(NumOfJob + StOfCrane - j - 1)
       Next j

       ArrOfCrane(StOfCrane) = Temp
    End If

'เครน พิจารณาที่ตำแหน่งสุดท้าย
ElseIf i = (2 * (NumOfJob - 1)) Then
    LastOfCrane = Int(((NumOfJob - 1) - 1 + 1) * Rnd + 1)

    '== นำเอาตำแหน่งที่สุ่มไปสลับกับตำแหน่งสุดท้ายแล้วเลื่อน ==
    Temp = ArrOfCrane(LastOfCrane)

    For j = LastOfCrane To NumOfJob - 1
       ArrOfCrane(j) = ArrOfCrane(j + 1)
    Next j

    ArrOfCrane(NumOfJob) = Temp

    '== สลับและเลื่อนตำแหน่งจากตำแหน่งแรกไปตำแหน่งก่อนสุดท้าย ==
    Temp = ArrOfCrane(NumOfJob - 1)

    For j = 1 To NumOfJob - 1 - 1
       ArrOfCrane(NumOfJob - 1 + 1 - j) = ArrOfCrane(NumOfJob - 1 + 1 - j - 1)
    Next j

    ArrOfCrane(1) = Temp

    '== นำเอาตำแหน่งที่สุดท้ายไปสลับกับตำแหน่งที่สุ่มได้แล้วเลื่อน ==
    Temp = ArrOfCrane(NumOfJob)

    For j = LastOfCrane To NumOfJob - 1
       ArrOfCrane(NumOfJob + LastOfCrane - j) = ArrOfCrane(NumOfJob + LastOfCrane - j - 1)
    Next j
```

รูปที่ ข.69 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS2

```
    ArrOfCrane(LastOfCrane) = Temp

'เครน พิจารณาที่ตำแหน่งตรงกลาง
Else
    Seq = (i \ 2) + 1

    If (i <> 2) Or (i <> (2 * (NumOfJob - 1)) - 1) Then

    'เครน ตำแหน่งที่พิจารณาสลับที่ไปทางด้านซ้าย
        If (i Mod 2) = 0 Then
            MidOfCrane = Int(((Seq - 1) - 1 + 1) * Rnd + 1)

            '== นำเอาตำแหน่งที่สุ่มไปสลับกับตำแหน่งสุดท้ายแล้วเลื่อน ==
            Temp = ArrOfCrane(MidOfCrane)

            For j = MidOfCrane To NumOfJob - 1
                ArrOfCrane(j) = ArrOfCrane(j + 1)
            Next j

            ArrOfCrane(NumOfJob) = Temp

            '== สลับและเลื่อนตำแหน่งจากตำแหน่งแรกไปตำแหน่งที่พิจารณา ==
            Temp = ArrOfCrane(Seq - 1)

            For J = 1 To Seq - 2
                ArrOfCrane(Seq - 1 + 1 - j) = ArrOfCrane(Seq - 1 + 1 - j - 1)
            Next J

            ArrOfCrane(1) = Temp

            '== นำเอาตำแหน่งที่สุดท้ายไปสลับกับตำแหน่งที่สุ่มได้แล้วเลื่อน ==
            Temp = ArrOfCrane(NumOfJob)

            For j = MidOfCrane To NumOfJob - 1
                ArrOfCrane(NumOfJob + MidOfCrane - j) = ArrOfCrane(NumOfJob + MidOfCrane - j - 1)
            Next J

            ArrOfCrane(MidOfCrane) = Temp

        'เครน ตำแหน่งที่พิจารณาสลับที่ไปทางด้านขวา
        Else
            MidOfCrane = Int((NumOfJob - (Seq + 1) + 1) * Rnd + (Seq + 1))

            '== นำเอาตำแหน่งที่สุ่มไปสลับกับตำแหน่งสุดท้ายแล้วเลื่อน ==
            Temp = ArrOfCrane(MidOfCrane)

            For j = MidOfCrane To NumOfJob - 1
                ArrOfCrane(j) = ArrOfCrane(j + 1)
            Next j

            ArrOfCrane(NumOfJob) = Temp

            '== สลับและเลื่อนตำแหน่งจากตำแหน่งแรกไปตำแหน่งที่พิจารณา ==
            Temp = ArrOfCrane(Seq)

            For J = Seq To NumOfJob - 2
                ArrOfCrane(j) = ArrOfCrane(j + 1)
            Next J
```

รูปที่ ข.69 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS2

```
            ArrOfCrane(NumOfJob - 1) = Temp

            '== นำเอาตำแหน่งที่สุดท้ายไปสลับกับตำแหน่งที่สุ่มได้แล้วเลื่อน ==
            Temp = ArrOfCrane(NumOfJob)

            For j = MidOfCrane To NumOfJob - 1
               ArrOfCrane(NumOfJob + MidOfCrane - j) = ArrOfCrane(NumOfJob + MidOfCrane - j - 1)
            Next j

            ArrOfCrane(MidOfCrane) = Temp

         End If

      Else
         If i = 2 Then
            MidOfCrane = 1
         Else
            MidOfCrane = NumOfJob - 1
         End If

      End If

   End If

   For j = 1 To NumOfJob
      Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j) = ArrOfCrane(j)
   Next j

Next i

End Sub
```

รูปที่ ข.69 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS2

```
Sub NewNS3()
'เป็นฟังก์ชันเรียกใช้การทำ Neighbourhood search ในการประมวลผล โดยใช้หลักการ 3.หลักการสลับที่

Dim NumOfJob As Integer, NumOfCrane As Integer, Temp As Integer, Seq As Integer
Dim RanOfJob As Integer, RanOfCrane As Integer
Dim StOfJob As Integer, LastOfJob As Integer, MidOfJob As Integer
Dim StOfCrane As Integer, LastOfCrane As Integer, MidOfCrane As Integer
Dim ArrOfJob() As Integer
Dim ArrOfCrane() As Integer
' อธิบายตัวแปร
   'numofjob    แสดง จำนวนของงาน        numofcrane  แสดง จำนวนของเครน
   'temp        แสดง ตัวแปรที่เก็บค่าได้  seq         แสดง ตำแหน่งที่ของงานหรือเครน
   'ranofjob    แสดง งานที่สุ่มได้        ranofcrane  แสดง เครนที่สุ่มได้
   'stofjob     แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งแรก
   'stofcrane   แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งแรก
   'lastofjob   แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งสุดท้าย
   'lastofcrane แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งสุดท้าย
   'midofjob    แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลาง
   'midofcrane  แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลาง
   'arrofjob()  แสดง ตำแหน่งของงาน
   'arrofcrane() แสดง ตำแหน่งของเครน
```

รูปที่ ข.70 แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS3

```
'รับจำนวนงานและจำนวนเครน
NumOfJob = Worksheets("EnterData").Cells(9, 7).Value
NumOfCrane = Worksheets("EnterData").Cells(10, 7).Value

ReDim ArrOfJob(1 To NumOfJob) As Integer
ReDim ArrOfCrane(1 To NumOfJob) As Integer

'ใส่งานและเครนในรูปแบบคำตอบทั้งหมด เพื่อเตรียมการสลับที่
For i = 1 To NumOfJob
   ArrOfJob(i) = Worksheets("Solve").Cells(26, 2 + i)
   ArrOfCrane(i) = Worksheets("solve").Cells(27, 2 + i)
Next i

For i = 1 To 2 * (NumOfJob - 1)
   For j = 1 To NumOfJob
      Worksheets("solve").Cells(34 + i, 2 + j) = ArrOfJob(j)
      Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j) = ArrOfCrane(j)
   Next j
Next i

'การจัดลำดับงาน โดยการสลับที่
For i = 1 To 2 * (NumOfJob - 1)

   For j = 1 To NumOfJob
      ArrOfJob(j) = Worksheets("solve").Cells(34 + i, 2 + j)
   Next j

   'งาน พิจารณาที่ตำแหน่งแรก
   If i = 1 Then
      StOfJob = Int((NumOfJob - 2 + 1) * Rnd + 2)
      Temp = ArrOfJob(i)
      ArrOfJob(i) = ArrOfJob(StOfJob)
      ArrOfJob(StOfJob) = Temp

   'งาน พิจารณาที่ตำแหน่งสุดท้าย
   ElseIf i = (2 * (NumOfJob - 1)) Then
      LastOfJob = Int(((NumOfJob - 1) - 1 + 1) * Rnd + 1)
      Temp = ArrOfJob(NumOfJob)
      ArrOfJob(NumOfJob) = ArrOfJob(LastOfJob)
      ArrOfJob(LastOfJob) = Temp

   'งาน พิจารณาที่ตำแหน่งตรงกลาง
   Else
      Seq = (i \ 2) + 1

      'งานตำแหน่งที่พิจารณาสลับที่ไปทางด้านซ้าย
      If (i Mod 2) = 0 Then
         MidOfJob = Int(((Seq - 1) - 1 + 1) * Rnd + 1)
         Temp = ArrOfJob(MidOfJob)
         ArrOfJob(MidOfJob) = ArrOfJob(Seq)
         ArrOfJob(Seq) = Temp

      'งานตำแหน่งที่พิจารณาสลับที่ไปทางด้านขวา
      Else
         MidOfJob = Int((NumOfJob - (Seq + 1) + 1) * Rnd + (Seq + 1))
         Temp = ArrOfJob(Seq)
```

รูปที่ ข.70 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS3

```
            ArrOfJob(Seq) = ArrOfJob(MidOfJob)
            ArrOfJob(MidOfJob) = Temp

        End If
    End If

    For j = 1 To NumOfJob
        Worksheets("solve").Cells(34 + i, 2 + j) = ArrOfJob(j)
    Next j

Next i

'จัดลำดับใหม่ของเครน
For i = 1 To 2 * (NumOfJob - 1)

    For j = 1 To NumOfJob
        ArrOfCrane(j) = Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j)
    Next j

    'เครน พิจารณาที่ตำแหน่งแรก
    If i = 1 Then
        StOfCrane = Int((NumOfJob - 2 + 1) * Rnd + 2)
        Temp = ArrOfCrane(i)
        ArrOfCrane(i) = ArrOfCrane(StOfCrane)
        ArrOfCrane(StOfCrane) = Temp

    'เครน พิจารณาที่ตำแหน่งสุดท้าย
    ElseIf i = (2 * (NumOfJob - 1)) Then
        LastOfCrane = Int(((NumOfJob - 1) - 1 + 1) * Rnd + 1)
        Temp = ArrOfCrane(LastOfCrane)
        ArrOfCrane(LastOfCrane) = ArrOfCrane(NumOfCrane)
        ArrOfCrane(NumOfCrane) = Temp

    'เครน พิจารณาที่ตำแหน่งตรงกลาง
    Else
        Seq = (i \ 2) + 1

        'เครนตำแหน่งที่พิจารณาสลับที่ไปทางด้านซ้าย
        If (i Mod 2) = 0 Then
            MidOfCrane = Int(((Seq - 1) - 1 + 1) * Rnd + 1)
            Temp = ArrOfCrane(MidOfCrane)
            ArrOfCrane(MidOfCrane) = ArrOfCrane(Seq)
            ArrOfCrane(Seq) = Temp

        'เครนตำแหน่งที่พิจารณาสลับที่ไปทางด้านขวา
        Else
            MidOfCrane = Int((NumOfJob - (Seq + 1) + 1) * Rnd + (Seq + 1))
            Temp = ArrOfCrane(Seq)
            ArrOfCrane(Seq) = ArrOfCrane(MidOfCrane)
            ArrOfCrane(MidOfCrane) = Temp
        End If
    End If

    For j = 1 To NumOfJob
        Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j) = ArrOfCrane(j)
    Next j
Next i
End Sub
```

รูปที่ ข.70 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS3

```
Sub NewNS4()
'เป็นฟังก์ชันเรียกใช้การทำ Neighbourhood search ในการประมวลผล โดยใช้หลักการ 4.หลักการพิจารณาที่ 2 ตำแหน่ง

Dim NumOfJob As Integer, NumOfCrane As Integer, Temp As Integer, Seq As Integer
Dim RanOfJob As Integer, RanOfCrane As Integer
Dim StOfJob As Integer, LastOfJob As Integer, MidOfJobDoo1 As Integer
Dim MidOfJobDoo2 As Integer
Dim StOfCrane As Integer, LastOfCrane As Integer, MidOfCraneDoo1 As Integer
Dim MidOfCraneDoo2 As Integer
Dim ArrOfJob() As Integer
Dim ArrOfCrane() As Integer

' อธิบายตัวแปร
   'numofjob          แสดง จำนวนของงาน
   'numofcrane        แสดง จำนวนของเครน
   'temp              แสดง ตัวแปรที่เก็บค่าได้
   'seq               แสดง ตำแหน่งที่ของงานหรือเครน
   'ranofjob          แสดง งานที่สุ่มได้
   'ranofcrane        แสดง เครนที่สุ่มได้
   'stofjob           แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งแรกและตำแหน่งที่สอง
   'stofcrane         แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งแรกและตำแหน่งที่สอง
   'lastofjob         แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งสุดท้ายและก่อนสุดท้าย
   'lastofcrane       แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งสุดท้ายและก่อนสุดท้าย
   'midofjobdoo1      แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลางตำแหน่งที่ 1
   'midofjobdoo2      แสดง งานที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลางตำแหน่งที่ 2
   'midofcranedoo1    แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลางตำแหน่งที่ 1
   'midofcranedoo2    แสดง เครนที่สุ่มได้เมื่อพิจารณาที่ตำแหน่งกลางตำแหน่งที่ 2
   'arrofjob()        แสดง ตำแหน่งของงาน
   'arrofcrane()      แสดง ตำแหน่งของเครน

'รับจำนวนงานและจำนวนเครน
NumOfJob = Worksheets("EnterData").Cells(9, 7).Value
NumOfCrane = Worksheets("EnterData").Cells(10, 7).Value

ReDim ArrOfJob(1 To NumOfJob) As Integer
ReDim ArrOfCrane(1 To NumOfJob) As Integer

ใส่งานและเครนในรูปแบบคำตอบทั้งหมด เพื่อเตรียมการสลับที่
For i = 1 To NumOfJob
   ArrOfJob(i) = Worksheets("Solve").Cells(26, 2 + i)
   ArrOfCrane(i) = Worksheets("solve").Cells(27, 2 + i)
Next i

For i = 1 To (NumOfJob - 1)
   For j = 1 To NumOfJob
      Worksheets("solve").Cells(34 + i, 2 + j) = ArrOfJob(j)
      Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j) = ArrOfCrane(j)
   Next j
Next i

'การจัดลำดับงาน โดยการสลับที่

For i = 1 To NumOfJob - 1

   For j = 1 To NumOfJob
      ArrOfJob(j) = Worksheets("solve").Cells(34 + i, 2 + j)
   Next j
```

รูปที่ ข.71 แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS4

```
    'งาน พิจารณาที่ตำแหน่งแรกและตำแหน่งที่สอง
    If i = 1 Then
        StOfJob = Int((NumOfJob - 3 + 1) * Rnd + 3)
        Temp = ArrOfJob(i + 1)

        For j = 2 To StOfJob - 1
            ArrOfJob(j) = ArrOfJob(j + 1)
        Next j

        ArrOfJob(StOfJob) = Temp

    'งาน พิจารณาที่ตำแหน่งสุดท้ายและก่อนสุดท้าย
    ElseIf i = NumOfJob - 1 Then
        LastOfJob = Int(((NumOfJob - 2) - 1 + 1) * Rnd + 1)
        Temp = ArrOfJob(NumOfJob - 1)

        For j = LastOfJob To NumOfJob - 2
            ArrOfJob(NumOfJob - 1 + LastOfJob - j) = _
                ArrOfJob(NumOfJob - 1 + LastOfJob - j - 1)
        Next j

        ArrOfJob(LastOfJob) = Temp

    'งาน พิจารณาที่ตำแหน่งตรงกลาง
    Else
        '2 ตำแหน่งที่พิจารณา คู่ที่ตำแหน่งแรก
        MidOfJobDoo1 = Int(((i - 1) - 1 + 1) * Rnd + 1)
        Temp = ArrOfJob(i)

        For j = MidOfJobDoo1 To i - 1
            ArrOfJob(i + MidOfJobDoo1 - j) = ArrOfJob(i + MidOfJobDoo1 - j - 1)
        Next j

        ArrOfJob(MidOfJobDoo1) = Temp

        '2 ตำแหน่งที่พิจารณา คู่ที่ตำแหน่งที่สอง
        MidOfJobDoo2 = Int((NumOfJob - (i + 1) + 1) * Rnd + (i + 1))
        Temp = ArrOfJob(i + 1)

        For j = i + 1 To MidOfJobDoo2 - 1
            ArrOfJob(j) = ArrOfJob(j + 1)
        Next j

        ArrOfJob(MidOfJobDoo2) = Temp

    End If

    For j = 1 To NumOfJob
        Worksheets("solve").Cells(34 + i, 2 + j) = ArrOfJob(j)
    Next j
Next i

'จัดลำดับใหม่ของเครน
For i = 1 To (NumOfJob - 1)

    For j = 1 To NumOfJob
        ArrOfCrane(j) = Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j)
    Next j
```

รูปที่ ข.71 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS4

```
'เครน  พิจารณาที่ตำแหน่งแรกและตำแหน่งที่สอง
If i = 1 Then
   StOfCrane = Int((NumOfJob - 3 + 1) * Rnd + 3)
   Temp = ArrOfCrane(i + 1)

   For j = 2 To StOfJob - 1
      ArrOfCrane(j) = ArrOfCrane(j + 1)
   Next j

   ArrOfCrane(StOfCrane) = Temp

'เครน พิจารณาที่ตำแหน่งสุดท้ายและก่อนสุดท้าย
ElseIf i = NumOfJob - 1 Then
   LastOfCrane = Int(((NumOfJob - 2) - 1 + 1) * Rnd + 1)
   Temp = ArrOfCrane(NumOfJob - 1)

   For j = LastOfCrane To NumOfCrane - 2
      ArrOfCrane(NumOfJob - 1 + LastOfCrane - j) = _
            ArrOfCrane(NumOfJob - 1 + LastOfCrane - j - 1)
   Next j
   ArrOfCrane(LastOfCrane) = Temp

'เครน พิจารณาที่ตำแหน่งตรงกลาง
 Else
   '2 ตำแหน่งที่พิจารณา คูที่ตำแหน่งแรก
   MidOfCraneDoo1 = Int(((i - 1) - 1 + 1) * Rnd + 1)
   Temp = ArrOfCrane(i)

   For j = MidOfCraneDoo1 To i - 1
      ArrOfCrane(i + MidOfCraneDoo1 - j) = ArrOfCrane(i + MidOfCraneDoo1 - j - 1)
   Next j
   ArrOfCrane(MidOfCraneDoo1) = Temp

   '2 ตำแหน่งที่พิจารณา คูที่ตำแหน่งที่สอง
   MidOfCraneDoo2 = Int((NumOfJob - (i + 1) + 1) * Rnd + (i + 1))
   Temp = ArrOfCrane(i + 1)

   For j = i + 1 To MidOfCraneDoo2 - 1
      ArrOfCrane(j) = ArrOfCrane(j + 1)
   Next j

   ArrOfCrane(MidOfCraneDoo2) = Temp

End If

For j = 1 To NumOfJob
   Worksheets("solve").Cells(34 + i, 2 + NumOfJob + 2 + j) = ArrOfCrane(j)
Next j
Next i
End Sub
```

รูปที่ ข.71 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง NewNS4

```
Public Sub MakespanInterval(z As Integer)

Dim STI() As Double
Dim FTI() As Double
Dim STJ() As Double
Dim FTJ() As Double
Dim OrderInCrane() As Integer
Dim Now() As Integer

Dim Order() As Integer
Dim Crane() As Integer
Dim PTime() As Double


NumOfJob = Worksheets("EnterData").Cells(9, 7).Value
NumOfCrane = Worksheets("EnterData").Cells(10, 7).Value

ReDim STJ(0 To NumOfCrane, 0 To NumOfJob)
ReDim FTJ(0 To NumOfCrane, 0 To NumOfJob)
ReDim STI(0 To NumOfCrane, 0 To NumOfJob)
ReDim FTI(0 To NumOfCrane, 0 To NumOfJob)
ReDim OrderInCrane(1 To NumOfCrane, 1 To NumOfJob)
ReDim Now(1 To NumOfCrane)
ReDim PTime(1 To NumOfJob)
ReDim Order(1 To NumOfJob)
ReDim Crane(1 To NumOfJob)

For i = 1 To NumOfJob
   PTime(i) = Worksheets("EnterData").Cells(24, 3 + i)
Next i

For i = 1 To NumOfJob
   Order(i) = Worksheets("solve").Cells(34 + z, 2 + i)
   Crane(i) = Worksheets("solve").Cells(34 + z, 2 + NumOfJob + 2 + i)
Next i

Dim CountCrane()   As Integer                    'Count How many job is assigned to a crane
ReDim CountCrane(1 To NumOfCrane)
Dim ContInt As Integer                           'Count how many interval for a crane at present
Dim Interval() As Integer
ReDim Interval(1 To NumOfJob)

Dim IntTime() As Integer
ReDim IntTime(0 To NumOfCrane, 0 To NumOfJob)
Const UpBound = 10000

For i = 0 To NumOfCrane
   For j = 0 To NumOfJob
      STJ(i, j) = 0
      FTJ(i, j) = 0
'     STI(i, j) = 0
'     FTI(I, j) = 0
   Next j
Next i
```

รูปที่ ข.72 แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง MakespanInterval

```
For i = 1 To NumOfCrane
'   Interval(i) = 0
    For j = 1 To NumOfJob
        IntTime(i, j) = 0
    Next j
Next i

For i = 1 To NumOfCrane
    CountCrane(i) = 0
Next i

For i = 1 To NumOfJob
    CountCrane(Worksheets("solve").Cells(34 + z, 2 + NumOfJob + 2 + i)) = _
      CountCrane(Worksheets("solve").Cells(34 + z, 2 + NumOfJob + 2 + i)) + 1

    OrderInCrane(Worksheets("solve").Cells(34 + z, 2 + NumOfJob + 2 + i), _
      CountCrane(Worksheets("solve").Cells(34 + z, 2 + NumOfJob + 2 + i))) = _
      Worksheets("solve").Cells(34 + z, 2 + i)
Next i
For i = 1 To NumOfJob
    For j = 1 To NumOfCrane
        For K = 1 To NumOfJob
            STJ(j, K) = 0
            FTJ(j, K) = 0
        Next K
    Next j
    For j = 1 To NumOfCrane
        Interval(j) = 0
        For K = 1 To NumOfJob
            IntTime(j, K) = 0
        Next K
    Next j
    'For each job we set the starting time at the finished time of the last job
    'that finish in the crane
    Now(Crane(i)) = Now(Crane(i)) + 1
    STJ(Crane(i), Now(Crane(i))) = FTJ(Crane(i), Now(Crane(i)) - 1)
    FTJ(Crane(i), Now(Crane(i))) = STJ(Crane(i), Now(Crane(i))) + PTime(Order(i))

    For j = 1 To NumOfCrane
        If Crane(i) < j Then
            ContInt = 0
            For K = 1 To Now(j)
                If FTJ(j, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
                    If STJ(j, K) < FTJ(Crane(i), Now(Crane(i)) - 1) Then
                        If OrderInCrane(j, K) > Order(i) Then
                            Interval(j) = Interval(j) + 1
                            STJ(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
                            ContInt = 1
                            IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(Crane(i), Now(Crane(i)) - 1) + FTJ(j, K)
                            If K = Now(j) Then
                              FTJ(j, Interval(j)) = UpBound
                              IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
                            End If
                        ElseIf OrderInCrane(j, K) < Order(i) Then
                            If K = Now(j) Then
                              Interval(j) = Interval(j) + 1
```

รูปที่ ข.72 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง MakespanInterval

```
            STK(j, Interval(j)) = FTJ(J, K)
            FTK(j, Interval(j)) = UpBound
            IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
          End If
        End If
      ' End If
      ElseIf STJ(J, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
        If FTJ(J, K - 1) < FTJ(Crane(i), Now(Crane(i)) - 1) Then
          If OrderInCrane(j, K) > Order(i) Then
            Interval(j) = Interval(j) + 1
            STK(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
            ContInt = 1
            IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(Crane(i), Now(Crane(i)) - 1) + FTJ(j, K)
            If K = Now(j) Then
              FTK(j, Interval(j)) = UpBound
              IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
            End If
          ElseIf OrderInCrane(j, K) < Order(i) Then
            If STJ(j, K) > FTJ(Crane(i), Now(Crane(i)) - 1) Then
              Interval(j) = Interval(j) + 1
              STK(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
              IntTime(j, Interval(j)) = STJ(j, K) - FTJ(Crane(i), Now(Crane(i)) - 1)
              FTK(j, Interval(j)) = STK(j, Interval(j)) + IntTime(j, Interval(j))
            End If
            ContInt = 0
            If K = Now(j) Then
              Interval(j) = Interval(j) + 1
              STK(j, Interval(j)) = FTJ(j, K)
              FTK(j, Interval(j)) = UpBound
              IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
            End If
          End If
        ElseIf FTJ(j, K - 1) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
          If OrderInCrane(j, K) > Order(i) Then
            If ContInt = 0 Then
              Interval(j) = Interval(j) + 1
              STK(j, Interval(j)) = FTJ(j, K - 1)
              ContInt = 1
              IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + FTJ(j, K)
              If K = Now(j) Then
                FTK(j, Interval(j)) = UpBound
                IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
              End If
            ElseIf ContInt = 1 Then
              IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + FTJ(j, K)
              If K = Now(j) Then
                FTK(j, Interval(j)) = UpBound
                IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
              End If
            End If

          ElseIf OrderInCrane(j, K) < Order(i) Then
            If STJ(j, K) > FTJ(j, K - 1) Then
              If ContInt = 0 Then
                Interval(j) = Interval(j) + 1
                STK(j, Interval(j)) = FTJ(j, K - 1)
                FTK(j, Interval(j)) = STJ(j, K)
                IntTime(j, Interval(j)) = FTK(j, Interval(j)) - STK(j, Interval(j))
```

รูปที่ ข.72 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง MakespanInterval

```
                    ElseIf ContInt = 1 Then
                        IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + STJ(j, K)
                        FTI(j, Interval(j)) = STI(j, Interval(j)) + IntTime(j, Interval(j))
                        ContInt = 0
                    End If
                    If K = Now(j) Then
                        Interval(j) = Interval(j) + 1
                        STI(j, Interval(j)) = FTJ(j, K)
                        FTI(j, Interval(j)) = UpBound
                        IntTime(j, Interval(j)) = FTI(j, Interval(j)) - STI(j, Interval(j))
                    End If
                ElseIf STJ(j, K) = FTJ(j, K - 1) Then
                    If ContInt = 1 Then
                        FTI(j, Interval(j)) = STJ(j, K)
                        IntTime(j, Interval(j)) = FTJ(j, K) - STJ(j, K)
                        ContInt = 0
                    End If
                    If K = Now(j) Then
                        Interval(j) = Interval(j) + 1
                        STI(j, Interval(j)) = FTJ(j, K)
                        FTI(j, Interval(j)) = UpBound
                        IntTime(j, Interval(j)) = FTI(j, Interval(j)) - STI(j, Interval(j))
                    End If
                End If
            End If
        End If
    End If
  End If
Next K

ElseIf Crane(i) > j Then
  ContInt = 0
  For K = 1 To Now(j)

    If FTJ(j, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
      If STJ(j, K) < FTJ(Crane(i), Now(Crane(i)) - 1) Then
        If OrderInCrane(j, K) < Order(i) Then
          Interval(j) = Interval(j) + 1
          STI(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
          ContInt = 1
          IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(Crane(i), Now(Crane(i)) - 1) + FTJ(j, K)
          If K = Now(j) Then
            FTI(j, Interval(j)) = UpBound
            IntTime(j, Interval(j)) = FTI(j, Interval(j)) - STI(j, Interval(j))
          End If
        ElseIf OrderInCrane(j, K) > Order(i) Then
          If K = Now(j) Then
            Interval(j) = Interval(j) + 1
            STI(j, Interval(j)) = FTJ(j, K)
            FTI(j, Interval(j)) = UpBound
            IntTime(j, Interval(j)) = FTI(j, Interval(j)) - STI(j, Interval(j))
          End If
        End If

'
'       End If
      ElseIf STJ(j, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
        If FTJ(j, K - 1) < FTJ(Crane(i), Now(Crane(i)) - 1) Then
          If OrderInCrane(j, K) < Order(i) Then
```

รูปที่ ข.72 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 **คำสั่ง** MakespanInterval

```
            Interval(j) = Interval(j) + 1
            STJ(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
            ContInt = 1
            IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(Crane(i), Now(Crane(i)) - 1) + FTJ(j, K)
            If K = Now(j) Then
              FTJ(j, Interval(j)) = UpBound
              IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
            End If
          ElseIf OrderInCrane(j, K) > Order(i) Then
            If STJ(j, K) > FTJ(Crane(i), Now(Crane(i)) - 1) Then
              Interval(j) = Interval(j) + 1
              STJ(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
              IntTime(j, Interval(j)) = STJ(j, K) - FTJ(Crane(i), Now(Crane(i)) - 1)
              FTJ(j, Interval(j)) = STJ(j, Interval(j)) + IntTime(j, Interval(j))
            End If
            ContInt = 0
            If K = Now(j) Then
              Interval(j) = Interval(j) + 1
              STJ(j, Interval(j)) = FTJ(j, K)
              FTJ(j, Interval(j)) = UpBound
              IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
            End If
          End If
        ElseIf FTJ(j, K - 1) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
          If OrderInCrane(j, K) < Order(i) Then
            If ContInt = 0 Then
              Interval(j) = Interval(j) + 1
              STJ(j, Interval(j)) = FTJ(j, K - 1)
              ContInt = 1
              IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + FTJ(j, K)
              If K = Now(j) Then
                FTJ(j, Interval(j)) = UpBound
                IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
              End If
            ElseIf ContInt = 1 Then
              IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + FTJ(j, K)
              If K = Now(j) Then
                FTJ(j, Interval(j)) = UpBound
                IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
              End If
            End If
          ElseIf OrderInCrane(j, K) > Order(i) Then
            If STJ(j, K) > FTJ(j, K - 1) Then
              If ContInt = 0 Then
                Interval(j) = Interval(j) + 1
                STJ(j, Interval(j)) = FTJ(j, K - 1)
                FTJ(j, Interval(j)) = STJ(j, K)
                IntTime(j, Interval(j)) = FTJ(j, Interval(j)) - STJ(j, Interval(j))
              ElseIf ContInt = 1 Then
                IntTime(j, Interval(j)) = IntTime(j, Interval(j)) - FTJ(j, K - 1) + STJ(j, K)
                FTJ(j, Interval(j)) = STJ(j, Interval(j)) + IntTime(j, Interval(j))
                ContInt = 0
              End If
```

รูปที่ ข.72 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง MakespanInterval

```
                        If K = Now(j) Then
                            Interval(j) = Interval(j) + 1
                            STI(j, Interval(j)) = FTJ(j, K)
                            FTI(j, Interval(j)) = UpBound
                            IntTime(j, Interval(j)) = FTI(j, Interval(j)) - STI(j, Interval(j))
                        End If
                    ElseIf STJ(j, K) = FTJ(j, K - 1) Then
                        If ContInt = 1 Then
                            FTI(j, Interval(j)) = STJ(j, K)
                            IntTime(j, Interval(j)) = FTJ(j, K) - STJ(j, K)
                            ContInt = 0
                        End If
                        If K = Now(j) Then
                            Interval(j) = Interval(j) + 1
                            STI(j, Interval(j)) = FTJ(j, K)
                            FTI(j, Interval(j)) = UpBound
                            IntTime(j, Interval(j)) = FTI(j, Interval(j)) - STI(j, Interval(j))
                        End If
'
                    End If
                End If
            End If
        End If
    End If
  Next K
 End If
Next j

'Now we take care of those cranes where there is no job at all
 For j = 1 To NumOfCrane

    If Interval(j) = 0 Then
        Interval(j) = 1
        STI(j, Interval(j)) = FTJ(Crane(i), Now(Crane(i)) - 1)
        FTI(j, Interval(j)) = UpBound
        IntTime(j, Interval(j)) = FTI(j, Interval(j)) - STI(j, Interval(j))
    End If

 Next j
'Now we check for the starting time of the job
Dim STCur As Integer
Dim FTCur As Integer
Dim Ok As Integer

 For j = 1 To NumOfCrane
  If J <> Crane(i) Then
    For K = 1 To Interval(j)
      Ok = 0
      If FTI(j, K) >= FTJ(Crane(i), Now(Crane(i)) - 1) Then
        If STI(j, K) <= FTJ(Crane(i), Now(Crane(i)) - 1) Then
          STCur = FTJ(Crane(i), Now(Crane(i)) - 1)
          FTCur = STCur + PTime(Order(i))
        Else
          STCur = STI(j, K)
          FTCur = STCur + PTime(Order(i))
        End If
```

รูปที่ ข.72 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง MakespanInterval

```
            If FTCur <= FTJ(J, K) Then
               For m = 1 To NumOfCrane
                  If m <> J And m <> Crane(i) Then
                   Ok = 0

                     For n = 1 To Interval(m)
                        If STJ(m, n) <= STCur And FTJ(m, n) >= FTCur Then
                           Ok = 1
                           Exit For
                        End If
                     Next n
                     If Ok = 0 Then Exit For
                  End If
               Next m
            End If
            If Ok = 1 Then
               STJ(Crane(i), Now(Crane(i))) = STCur
               FTJ(Crane(i), Now(Crane(i))) = STJ(Crane(i), Now(Crane(i))) + PTime(Order(i))
               Exit For
            End If

         End If
      Next K
      If Ok = 1 Then Exit For

   End If
Next j
Next i

'Check for makespan
Dim Makespan As Double
For i = 1 To NumOfCrane
   For J = 1 To Now(i)
      If FTJ(i, J) >= Makespan Then
         Makespan = FTJ(i, J)
      End If
   Next J
Next i

Worksheets("Solve").Cells(34 + z, 2 + NumOfJob + 2 + NumOfJob + 1) = Makespan

End Sub
```

รูปที่ ข.72 (ต่อ) แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง MakespanInterval

```
Sub ShowDialog()

   Page6ProgressBar.Show

End Sub
```

รูปที่ ข.73 แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง ShowDialog

```
Function CancelConfirm() As Boolean
If MsgBox("คุณต้องการหยุดการประมวลผลหรือไม่?", vbQuestion + vbYesNo, "Stop process") = vbYes Then
  {Cancel = True
  CancelConfirm = True
  Worksheets("EnterData").Cells(1, 1).Value = 1
Else
  CancelConfirm = False
  Worksheets("EnterData").Cells(1, 1).Value = 0
End If
End Function
```

รูปที่ ข.74 แสดงคำสั่งเรียกใช้ใน Module2 คำสั่ง CancelConfirm

ประวัติผู้ดำเนินโครงงาน

ชื่อ นางสาวหทัยรัตน์ ธีระกาญจน์
ภูมิลำเนา 231 หมู่ 1 ต.วงฆ้อง อ.พรหมพิราม จ.พิษณุโลก
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนจุฬาภรณ์ฯ พิษณุโลก จ.พิษณุโลก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 6 สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: jahathairat@hotmail.com

ชื่อ นางสาวศิริพร ชารีพร
ภูมิลำเนา 2/1 หมู่ที่ 1 ต.อ่างทอง อ.เชียงคำ จ.พะเยา
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนเชียงคำวิทยาคม จ.พะเยา
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: kukik_ie@winsdowslive.com