

การจัดสมดุลสายงานการประกอบโดยใช้การค้นหาแบบตามเชิงพันทาง

ASSEMBLY LINE BALANCING BY HYBRID TABU SEARCH

สุภาภรณ์ สุวรรณรังษี

SUPAPORN SUWANNARONGSRI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหกรรม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2551

KMITL-2008-EN-M-217-402

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจัดสมุดสายงานการประกอบโดยใช้การค้นหาแบบตามูเชิงพันทาง

ASSEMBLY LINE BALANCING BY HYBRID TABU SEARCH

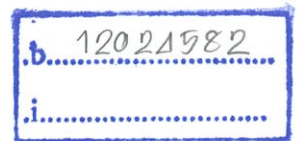


สุภาพรณ์ สุวรรณรังษี

SUPAPORN SUWANNARONGSRI

พ.
ศ ๒๕๕๑
๒๕๕๑

เลขหมู่.....
เลขทะเบียน 87094
วัน,เดือน,ปี 30 ส.ค. 2552



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหกรรม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2551

KMITL-2008-EN-M-217-402

ASSEMBLY LINE BALANCING BY HYBRID TABU SEARCH

SUPAPORN SUWANNARONGSRI

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN INDUSTRIAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2008

KMITL-2008-EN-M-217-402

COPYRIGHT 2008

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การจัดสมมูลสายงานการประกอบโดยใช้การค้นหาแบบตามเชิงพันทาง
นักศึกษา	นางสาวสุภาภรณ์ สุวรรณรัมย์
รหัสประจำตัว	49061806
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมอุตสาหการ
พ.ศ.	2551
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ. ดร.สรรพสิทธิ์ ลีมนรรัตน์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผศ. ดร.เดชา พวงดาวเรือง

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอวิธีการทางปัญญาประดิษฐ์แบบพันทางแนวใหม่เพื่อแก้ปัญหาการจัดสมมูลสายงานการประกอบ การค้นหาแบบตาม (tabu search: TS) ซึ่งเป็นเทคนิคการค้นหาแบบปัญญาประดิษฐ์ที่ทรงประสิทธิภาพ ได้รับการนำมาใช้งานร่วมกับเทคนิคการสลับที่แบบสุ่ม (random permutation technique) เพื่อแก้ปัญหาการจัดสมมูลสายงานการประกอบแบบผลิตภัณฑ์เดียว โดยให้ชื่อว่าการค้นหาแบบตามเชิงพันทาง (hybrid tabu search: HTS) ค่าความแปรปรวนของภาระงานถูกกำหนดให้เป็นฟังก์ชันวัตถุประสงค์ของกระบวนการค้นหา จากวิธีการที่นำเสนอในวิทยานิพนธ์ฉบับนี้ TS จะทำหน้าที่จัดสรรจำนวนชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน ในขณะที่เทคนิคการสลับที่แบบสุ่มจะทำหน้าที่จัดลำดับของชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน โดยสอดคล้องกับเงื่อนไขลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน อัลกอริทึมของวิธี HTS ที่นำเสนอจะถูกเขียนเป็นโปรแกรมการค้นหาโดย MATLAB ปัญหาการจัดสมมูลสายงานการประกอบแบบผลิตภัณฑ์เดียวจำนวน 4 ปัญหาที่ได้มาจากการสำรวจงานวิจัยที่เกี่ยวข้องจะถูกนำมาทดสอบกับวิธีที่นำเสนอ ผลการทดสอบถูกนำไปเปรียบเทียบกับวิธี COMSOAL ซึ่งเป็นเทคนิคการแก้ปัญหาการจัดสมมูลสายงานการประกอบแบบดั้งเดิมผ่านการจำลองด้วยคอมพิวเตอร์ จากผลการทดสอบพบว่าวิธี HTS สามารถค้นหาคำตอบของปัญหาการจัดสมมูลสายงานการประกอบที่ดีกว่าวิธี COMSOAL ในทุกกรณีปัญหา ค่าความแปรปรวนของภาระงานที่ลดลงมากที่สุดจากคำตอบที่ได้จากวิธี HTS เปรียบเทียบกับวิธี COMSOAL มีค่าเท่ากับ 96.73% ซึ่งสรุปได้ว่าวิธี HTS ที่นำเสนอ เป็นวิธีการที่มีศักยภาพ และเป็นทางเลือกหนึ่งในการแก้ปัญหาการจัดสมมูลสายงานการประกอบ

Thesis	Assembly Line Balancing by Hybrid Tabu Search
Student	Miss.Supaporn Suwannarongsri
Student ID.	49061806
Degree	Master of Engineering
Program	Industrial Engineering
Year	2008
Thesis Advisor	Asst.Prof. Dr.Sunpasit Limnarat
Thesis Co-advisor	Asst.Prof. Dr.Deacha Puangdownreong

ABSTRACT

This master thesis proposes a new hybrid intelligent method for solving assembly line balancing problems. The tabu search (TS) method, one of the most efficient AI search techniques, and the random permutation technique are combined to identify and provide solutions for single-model assembly line balancing problems. The proposed approach is named the hybrid tabu search (HTS) method. The workload variance is set as the objective function. With the proposed approach, the TS method will address the number of tasks assigned for each workstation, while the random permutation technique will assign the sequence of tasks for each workstation according to precedence constraints. The proposed HTS algorithm will be coded by MATLAB program. In this thesis, four single-model assembly line balancing problems from a survey of literature will be tested against the proposed approach. Results obtained from the proposed method are compared with those obtained from the COMSOAL method, one of the conventional methods for assembly line balancing problems. As obtained results, it was found that the proposed HTS method is capable of producing solutions superior to the COMSOAL method for all four problems. The maximum reduction of the workload variance is of 96.73% when comparing with the COMSOAL method. It can be concluded that the HTS is a potential and alternative method to solve assembly line balancing problems.

กิตติกรรมประกาศ

งานวิจัยวิทยานิพนธ์นี้สำเร็จลงได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ. ดร.สรพพิทธิ์ ลิ้มนรรตน์ อาจารย์ประจำภาควิชาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และอาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ผศ. ดร.เดชา พวงดาวเรือง อาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเอเชียอาคเนย์ ที่ให้ความช่วยเหลือและให้คำแนะนำในการแก้ปัญหา ตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณ ผศ. ดร.ฤดี มาสุจันทร์ ผศ. ดร.สิทธิพร พิมพัสกุล และอาจารย์ ดร.สกันธ์ คล่องบุญจิต กรรมการสอบวิทยานิพนธ์ที่ได้กรุณาให้คำแนะนำอันเป็นประโยชน์

ขอขอบคุณบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้ทุนสนับสนุนการจัดทำวิทยานิพนธ์ในครั้งนี้

ขอขอบคุณเพื่อนนักศึกษาบัณฑิตศึกษา ภาควิชาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ได้ให้ความช่วยเหลือและกำลังใจในการทำงานวิจัยนี้ด้วยดีเสมอมา

ท้ายนี้ ผู้ทำวิจัยขอกราบขอบพระคุณ บิดา มารดา พี่น้อง และญาติทุกท่านซึ่งคอยให้การสนับสนุนในทุกๆ ด้าน ทั้งความห่วงใย และกำลังใจเสมอมา จนข้าพเจ้าสำเร็จการศึกษา

สุภาภรณ์ สุวรรณรัมย์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 สมมุติฐานของการวิจัย.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 ขอบเขตของการวิจัย.....	3
1.6 ข้อตกลงเบื้องต้น.....	4
1.7 ขั้นตอนของการดำเนินงานวิจัย.....	4
1.8 การจัดรูปเล่มวิทยานิพนธ์.....	5
บทที่ 2 ปฏิทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 กล่าวนำ.....	6
2.2 การจัดสมดุลสายงานการประกอบ.....	6
2.3 การค้นหาแบบปัญญาประดิษฐ์.....	13
2.4 การประยุกต์การค้นหาแบบปัญญาประดิษฐ์เพื่อแก้ปัญหา ALB.....	15
2.5 สรุป.....	17
บทที่ 3 ปัญหาการจัดสมดุลสายงานการประกอบ.....	18
3.1 กล่าวนำ.....	18
3.2 ทฤษฎีเบื้องต้นของการจัดสมดุลสายงานการประกอบ.....	18
3.3 ลักษณะปัญหาการจัดสมดุลสายงานการประกอบแบบผลิตภัณฑ์เดียว.....	21
3.4 วิธี COMSOAL.....	22
3.5 สรุป.....	26

สารบัญ (ต่อ)

	หน้า
บทที่ 4 การค้นหาแบบตามเชิงพื้นที่.....	27
4.1 กล่าวนำ.....	27
4.2 การค้นหาแบบตาม.....	27
4.3 การค้นหาแบบตามเชิงพื้นที่.....	32
4.4 การทดสอบโปรแกรมการค้นหาแบบตามเชิงพื้นที่.....	38
4.5 สรุป.....	54
บทที่ 5 ผลการทดสอบ.....	55
5.1 กล่าวนำ.....	55
5.2 ปัญหา ALB ที่ใช้ในการทดสอบ.....	55
5.3 ขั้นตอนการทดสอบ.....	59
5.4 การกำหนดค่าพารามิเตอร์ของ HTS.....	60
5.5 ผลการทดสอบ.....	64
5.6 สรุป.....	85
บทที่ 6 บทสรุปและข้อเสนอแนะ.....	88
6.1 ข้อสรุป.....	88
6.2 อภิปรายผล.....	90
6.3 ข้อเสนอแนะ.....	91
เอกสารอ้างอิง.....	92
ภาคผนวก ก รายละเอียดปัญหา ALB ที่ใช้ในงานวิจัย.....	97
ภาคผนวก ข รายละเอียดโปรแกรมการค้นหาแบบตาม.....	104
ภาคผนวก ค รายละเอียดโปรแกรมการค้นหาแบบตามเชิงพื้นที่เพื่อแก้ปัญหา ALB.....	111
ภาคผนวก ง รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์.....	130
ประวัติผู้เขียน.....	131

สารบัญตาราง

ตารางที่	หน้า
3.1 รายละเอียดของสายงานการประกอบในรูปที่ 3.1	20
3.2 การจัดสมมูลสายงานการประกอบในรูปที่ 3.1.....	21
3.3 รายละเอียดของสายงานการประกอบในรูปที่ 3.3	24
3.4 ผลการจัดสมมูลสายงานการประกอบในรูปที่ 3.3 ด้วยวิธี COMSOAL	25
4.1 รายละเอียดของสายงานการประกอบอย่างง่าย.....	36
4.2 ผลการทดสอบ โปรแกรม TS กับ BF	40
4.3 รายละเอียดปัญหา ALB ที่ใช้ทดสอบ โปรแกรม HTS.....	46
4.4 ผลการทดสอบ โปรแกรม HTS เพื่อแก้ปัญหา ALB ที่มีชั้นงานจำนวน 11 ชั้นงาน	49
4.5 ผลการแก้ปัญหา ALB จำนวน 11 ชั้นงานที่ได้จากวิธี HTS	54
5.1 ปัญหา ALB ที่ใช้ในการทดสอบ	55
5.2 ปัญหา ALB ที่ใช้ในการทดสอบ	57
5.3 ผลการทดสอบการแก้ปัญหา Buxey ด้วยวิธี HTS.....	65
5.4 ผลการเปรียบเทียบการแก้ปัญหา Buxey ด้วยวิธี COMSOAL และวิธี HTS.....	69
5.5 ผลการทดสอบการแก้ปัญหา Gunther ด้วยวิธี HTS	70
5.6 ผลการเปรียบเทียบการแก้ปัญหา Gunther ด้วยวิธี COMSOAL และวิธี HTS	71
5.7 ผลการทดสอบการแก้ปัญหา Random ด้วยวิธี HTS.....	76
5.8 ผลการเปรียบเทียบการแก้ปัญหา Random ด้วยวิธี COMSOAL และวิธี HTS.....	80
5.9 ผลการทดสอบการแก้ปัญหา M-cycle ด้วยวิธี HTS.....	81
5.10 ผลการเปรียบเทียบการแก้ปัญหา M-cycle ด้วยวิธี COMSOAL และวิธี HTS.....	86
5.11 สรุปผลการเปรียบเทียบการแก้ปัญหา ALB ด้วยวิธี COMSOAL และวิธี HTS.....	87
ก.1 รายละเอียดของสายงานการประกอบปัญหา Buxey	97
ก.2 รายละเอียดของสายงานการประกอบปัญหา Gunther	98
ก.3 รายละเอียดของสายงานการประกอบปัญหา Random.....	100
ก.4 รายละเอียดของสายงานการประกอบปัญหา M-cycle	101

สารบัญรูป

รูปที่	หน้า
2.1 สายงานการประกอบ	7
2.2 ปัญหา ALB จำแนกตาม Ghosh และ Gagon [21]	9
2.3 ปัญหา ALB จำแนกตาม Scholl และ Becker [15],[30]	10
2.4 แผนภูมิลำดับงานก่อนหลัง	12
3.1 แผนภูมิลำดับงานก่อนหลัง	19
3.2 ผลการจัดสมมูลสายงานการประกอบในรูปที่ 3.1	21
3.3 แผนภูมิลำดับงานก่อนหลังของสายงานการประกอบที่มีชั้นงานจำนวน 11 ชั้นงาน	24
3.4 การจัดสมมูลสายงานการประกอบในรูปที่ 3.3 ด้วยวิธี COMSOAL	25
4.1 การสุ่มคำตอบเริ่มต้น x_0	29
4.2 การผลิตค่าใกล้เคียงแบบสุ่ม	29
4.3 การปรับค่า $x_0 = x_1$	29
4.4 ทิศทางการค้นหา	29
4.5 การใช้ x_0 ใหม่เป็นคำตอบเริ่มต้น	30
4.6 ทิศทางการค้นหาในรอบถัดมา	30
4.7 กลไกย้อนรอยการค้นหา	30
4.8 แผนภูมิการทำงานของการค้นหาแบบตาบ	31
4.9 แผนผังการแก้ปัญหา ALB ด้วย HTS	32
4.10 แผนภูมิการทำงานของ HTS เพื่อการแก้ปัญหา ALB	34
4.11 แผนภูมิการสลับที่แบบสุ่ม	35
4.12 การจัดกลุ่มชั้นงานเพื่อการสลับที่แบบสุ่มของปัญหา ALB ในตารางที่ 4.1	37
4.13 ผลการจัดกลุ่มชั้นงานเพื่อการสลับที่แบบสุ่มของปัญหา ALB ในตารางที่ 4.1	37
4.14 พื้นผิวฟังก์ชัน BF	38
4.15 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 25%$ และ $N = 15$	41
4.16 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 25%$ และ $N = 30$	41
4.17 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 25%$ และ $N = 45$	41
4.18 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 25%$ และ $N = 60$	42
4.19 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 50%$ และ $N = 15$	42

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.20 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 50%$ และ $N = 30$	42
4.21 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 50%$ และ $N = 45$	43
4.22 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 50%$ และ $N = 60$	43
4.23 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 75%$ และ $N = 15$	43
4.24 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 75%$ และ $N = 30$	44
4.25 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 75%$ และ $N = 45$	44
4.26 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 75%$ และ $N = 60$	44
4.27 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 100%$ และ $N = 15$	45
4.28 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 100%$ และ $N = 30$	45
4.29 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 100%$ และ $N = 45$	45
4.30 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ เมื่อ $R = 100%$ และ $N = 60$	46
4.31 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ในตารางที่ 4.3	47
4.32 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา ALB ในตารางที่ 4.3	48
4.33 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 20%$ และ $N = 15$	50
4.34 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 20%$ และ $N = 30$	50
4.35 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 20%$ และ $N = 45$	50
4.36 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 20%$ และ $N = 60$	51
4.37 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 40%$ และ $N = 15$	51
4.38 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 40%$ และ $N = 30$	51
4.39 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 40%$ และ $N = 45$	52
4.40 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 40%$ และ $N = 60$	52
4.41 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 60%$ และ $N = 15$	52
4.42 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 60%$ และ $N = 30$	53
4.43 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 60%$ และ $N = 45$	53
4.44 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชิ้นงาน เมื่อ $R = 60%$ และ $N = 60$	53
4.45 ผลการแก้ปัญห ALB 11 ชิ้นงานที่ได้จากวิธี HTS	54
5.1 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ปัญหาที่ 1 (Buxey)	56

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.2 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ปัญหาที่ 2 (Gunther).....	56
5.3 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ปัญหาที่ 3 (Random).....	57
5.4 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ปัญหาที่ 4 (M-cycle).....	58
5.5 แผนผังการแก้ปัญห ALB ด้วยวิธี HTS ที่นำเสนอ	60
5.6 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา Buxey	61
5.7 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา Gunther	62
5.8 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา Random.....	62
5.9 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา M-cycle	63
5.10 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 20\%$ และ $N = 30$	65
5.11 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 20\%$ และ $N = 45$	66
5.12 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 20\%$ และ $N = 60$	66
5.13 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 40\%$ และ $N = 30$	66
5.14 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 40\%$ และ $N = 45$	67
5.15 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 40\%$ และ $N = 60$	67
5.16 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 60\%$ และ $N = 30$	67
5.17 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 60\%$ และ $N = 45$	68
5.18 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 60\%$ และ $N = 60$	68
5.19 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey ที่ให้ผลดีที่สุด.....	68
5.20 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 20\%$ และ $N = 30$	72
5.21 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 20\%$ และ $N = 45$	72
5.22 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 20\%$ และ $N = 60$	72
5.23 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 40\%$ และ $N = 30$	73
5.24 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 40\%$ และ $N = 45$	73
5.25 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 40\%$ และ $N = 60$	73
5.26 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 60\%$ และ $N = 30$	74
5.27 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 60\%$ และ $N = 45$	74
5.28 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 60\%$ และ $N = 60$	74

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.29 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther ที่ให้ผลดีที่สุด.....	75
5.30 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 20\%$ และ $N = 30$	76
5.31 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 20\%$ และ $N = 45$	77
5.32 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 20\%$ และ $N = 60$	77
5.33 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 40\%$ และ $N = 30$	77
5.34 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 40\%$ และ $N = 45$	78
5.35 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 40\%$ และ $N = 60$	78
5.36 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 60\%$ และ $N = 30$	78
5.37 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 60\%$ และ $N = 45$	79
5.38 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 60\%$ และ $N = 60$	79
5.39 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random ที่ให้ผลดีที่สุด.....	79
5.40 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 20\%$ และ $N = 30$	82
5.41 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 20\%$ และ $N = 45$	82
5.42 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 20\%$ และ $N = 60$	82
5.43 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 40\%$ และ $N = 30$	83
5.44 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 40\%$ และ $N = 45$	83
5.45 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 40\%$ และ $N = 60$	83
5.46 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 60\%$ และ $N = 30$	84
5.47 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 60\%$ และ $N = 45$	84
5.48 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 60\%$ และ $N = 60$	84
5.49 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle ที่ให้ผลดีที่สุด.....	85

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

นับตั้งแต่อดีตจนถึงปัจจุบัน ปัญหาที่มีความสำคัญอย่างยิ่งปัญหาหนึ่งในระบบการผลิตคือการจัดสมดุลสายงานการประกอบ (assembly line balancing: ALB) สายงานการประกอบคือลำดับของสถานีงานที่เชื่อมต่อกันเพื่อการประกอบส่วนประกอบต่างๆ ขึ้นเป็นผลิตภัณฑ์ ปัญหาการจัดสมดุลสายงานการประกอบ คือการกำหนดชิ้นงาน (task) ให้กับแต่ละสถานีงาน (workstation) เพื่อให้เวลาว่างงาน (idle time) ของสายงานการประกอบมีค่าน้อยที่สุด และสอดคล้องกับเงื่อนไขที่สำคัญสองเงื่อนไข เงื่อนไขแรกคือเวลาการทำงานรวมของแต่ละสถานีงานจะต้องน้อยกว่าหรือเท่ากับรอบเวลาการผลิต (cycle time) และเงื่อนไขที่สองคือชิ้นงานที่ได้รับการจัดสรรให้กับแต่ละสถานีงานจะต้องสอดคล้องกับลำดับความสัมพันธ์ก่อนหลัง precedence relationship) [1] ปัญหา ALB อาจมีหลายรูปแบบเช่น ปัญหา ALB แบบผลิตภัณฑ์เดียว (single-model) และแบบผลิตภัณฑ์ผสม (mixed-model) เป็นต้น ตัวแปรที่เกี่ยวข้องกับการจัดสมดุลสายงานประกอบได้แก่จำนวนชิ้นงาน เวลาการทำงานของแต่ละชิ้นงาน ลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน และรอบเวลาการผลิต วัตถุประสงค์ของการแก้ปัญหา ALB คือการทำให้จำนวนสถานีงานน้อยที่สุด ความแปรปรวนของภาระงานน้อยที่สุด เวลาว่างงานน้อยที่สุด และประสิทธิภาพของสายงานการประกอบมากที่สุด

ปัญหาการจัดสมดุลสายงานการประกอบเป็นปัญหามตะ (classic problem) ปัญหาหนึ่งในบริบทของวิศวกรรมอุตสาหกรรม ซึ่งการจัดสมดุลของสายงานการประกอบในทางอุตสาหกรรมได้รับการนำเสนอเป็นครั้งแรกในช่วงต้นทศวรรษที่ 1900 [2] และได้รับการตีพิมพ์ในนามปัญหาการจัดสมดุลสายงานการประกอบ (assembly line balancing problems: ALBP) หรืออาจเรียกว่าปัญหา ALB เป็นครั้งแรกในปี ค.ศ. 1955 [3] นับจากนั้นเป็นต้นมาปัญหา ALB ได้รับความสนใจอย่างกว้างขวางในกลุ่มนักวิจัยและวิศวกร ปัญหา ALB นับเป็นปัญหาแบบ NP-Hard (nondeterministic polynomial-time hard) ในลักษณะปัญหาการหาค่าเหมาะที่สุดเชิงผสมผสาน (combinatorial optimization problem) [4] กล่าวคือเป็นปัญหาที่มีคำตอบวงแคบเฉพาะถิ่น (local solution) จำนวนมาก แต่มีคำตอบวงกว้าง (global solution) เพียงคำตอบเดียว ปัญหาเช่นนี้จะใช้เวลาในการค้นหาคำตอบที่เหมาะสมในลักษณะที่มีความแปรผันตรงแบบเอ็กโพเนนเชียลกับจำนวนตัวแปรที่เพิ่มขึ้น ซึ่งเหมาะสมต่อการนำเอาเทคนิคการค้นหาแบบปัญญาประดิษฐ์ที่ตั้งอยู่บนพื้นฐานของวิธีศึกษาสำนึก (heuristic approach) มาประยุกต์ใช้เพื่อแก้ปัญหา

การแก้ปัญหา ALB ที่ปรากฏในวิจัยก่อนหน้าเป็นการพัฒนาอัลกอริทึมที่สามารถให้คำตอบที่เหมาะสมที่สุด ทั้งวิธีแบบแม่นยำ [5-6] และอัลกอริทึมที่ใช้วิธีศึกษาสำนึก (heuristic method) [7-8] เป็นต้น แต่เนื่องจากปัญหา ALB เป็นปัญหาแบบ NP-Hard ในลักษณะปัญหาการหาค่าที่เหมาะสมที่สุดเชิงผสมผสานดังได้กล่าวมาแล้ว จึงพบว่าอัลกอริทึมที่ใช้วิธีศึกษาสำนึกจะมีความยืดหยุ่นมากกว่า และสามารถให้คำตอบที่ดีกว่า จากงานวิจัยที่ผ่านมาการประยุกต์เทคนิคการค้นหาแบบปัญญาประดิษฐ์ที่ตั้งอยู่บนวิธีศึกษาสำนึกสำหรับแก้ปัญหา ALB ได้แก่ การประยุกต์ใช้ GA (genetic algorithm) ในการแก้ปัญหา ALB แบบผลิตภัณฑ์เดียวดังปรากฏใน [9-10] การประยุกต์ใช้ TS ในการแก้ปัญหา ALB แบบผลิตภัณฑ์เดียวดังปรากฏใน [11] และการประยุกต์ใช้ GA ในการแก้ปัญหา ALB แบบผลิตภัณฑ์ผสมดังปรากฏใน [12] อย่างไรก็ตาม แม้ว่า GA จะมีวิวัฒนาการในการค้นหาคำตอบที่เหมาะสมที่สุดตามกลไกการวิวัฒนาการของสิ่งมีชีวิต แต่กระบวนการค้นหาคำตอบจะดำเนินไปอย่างค่อนข้างล่าช้า อีกทั้งอาจประสบปัญหาการกำหนดพารามิเตอร์ที่เหมาะสมให้กับกระบวนการจากการสำรวจงานวิจัยพบว่า TS สามารถค้นหาคำตอบที่เหมาะสมได้เร็วกว่า GA [13] แต่อย่างไรก็ตาม การประยุกต์ใช้ TS ในการแก้ปัญหา ALB เพียงอย่างเดียวโดยตรงนั้นอาจประสบปัญหาเกี่ยวกับการกำหนดฟังก์ชันวัตถุประสงค์ (objective function) สำหรับการค้นหา ซึ่งอาจมีความซับซ้อน และอาจอยู่ในรูปแบบหลายวัตถุประสงค์ (multiple objectives)

ดังนั้นในวิทยานิพนธ์นี้จึงได้นำเสนอวิธีการค้นหาแบบตามูเชิงพันทาง (hybrid tabu search: HTS) เพื่อแก้ปัญหา ALB หลักการของอัลกอริทึมที่นำเสนอคือการทำงานร่วมกันของ TS และเทคนิคการสลับที่แบบสุ่ม (random permutation technique) โดยที่ TS จะทำหน้าที่จัดสรรจำนวนชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน ในขณะที่การสลับที่แบบสุ่มจะทำหน้าที่จัดลำดับของชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน โดยสอดคล้องกับเงื่อนไขลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน อัลกอริทึมของ HTS จะถูกเขียนเป็นโปรแกรมการค้นหาโดย MATLAB เพื่อแก้ปัญหา ALB แบบผลิตภัณฑ์เดียวจำนวน 4 ปัญหาที่ได้มาจากการสำรวจงานวิจัยที่เกี่ยวข้อง ผลการทดสอบวิธีการที่นำเสนอจะถูกนำไปเปรียบเทียบกับวิธี COMSOAL [8] ซึ่งเป็นวิธีการแก้ปัญหา ALB แบบดั้งเดิมวิธีการหนึ่งที่ได้รับคามนิยมอย่างกว้างขวาง โดยผ่านการจำลองด้วยคอมพิวเตอร์ และการประเมินผลการทดสอบอาศัยการเปรียบเทียบความแปรปรวนของภาระงานของสายงานการประกอบของวิธีการที่นำเสนอกับวิธีการ COMSOAL

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อพัฒนาอัลกอริทึมการค้นหาแบบตามูเชิงพันทางสำหรับแก้ปัญหาการจัดสมดุลสายงานการประกอบ

2. เพื่อพัฒนาโปรแกรมคอมพิวเตอร์การค้นหาแบบตามูเชิงพันทางสำหรับแก้ปัญหาการจัดสมดุลสายงานการประกอบ
3. เพื่อแก้ปัญหาการจัดสมดุลสายงานการประกอบโดยใช้การค้นหาแบบตามูเชิงพันทาง

1.3 สมมติฐานของการวิจัย

การค้นหาแบบตามูเชิงพันทางที่น่าเสนอ สามารถแก้ปัญหาการจัดสมดุลสายงานการประกอบได้อย่างมีประสิทธิภาพ และได้ผลดีกว่าวิธี COMSOAL โดยการตรวจสอบสมมติฐานดังกล่าวจะอาศัยการเปรียบเทียบค่าความแปรปรวนของภาระงานที่ได้จากวิธีการที่น่าเสนอและวิธี COMSOAL

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถใช้การค้นหาแบบตามูเชิงพันทางแก้ปัญหาการจัดสมดุลสายงานการประกอบ
2. สามารถพัฒนาอัลกอริทึม และโปรแกรมการค้นหาแบบตามูเชิงพันทางสำหรับแก้ปัญหาการจัดสมดุลสายงานการประกอบ
3. ลดความยุ่งยากและลดเวลาในการแก้ปัญหาการจัดสมดุลสายงานการประกอบ
4. เป็นแนวทางในการตัดสินใจจัดสถานีงาน และจัดระบบสายงานการประกอบให้มีประสิทธิภาพมากยิ่งขึ้น

1.5 ขอบเขตของการวิจัย

1. ทำการศึกษาเฉพาะปัญหา ALB แบบผลิตภัณฑ์เดียวที่มีลักษณะของปัญหาเป็นแบบ SMD (single-model deterministic)
2. ปัญหา ALB ที่ศึกษาจะให้ความสนใจเฉพาะการจัดกลุ่มของชิ้นงานให้แต่ละสถานีงาน โดยมีการกำหนดรอบเวลาการผลิตมาให้
3. ปัญหา ALB แบบผลิตภัณฑ์เดียวที่ศึกษามีลักษณะดังนี้
 - 3.1 เวลาการทำงานของแต่ละชิ้นงานจะกำหนดเป็นเวลามาตรฐาน
 - 3.2 สายงานการประกอบแบบอนุกรม กล่าวคือแต่ละสถานีงานจะทำงานต่อเนื่องกันตามลำดับไม่มีสถานีงานที่ทำงานขนานกัน
 - 3.3 สายงานการประกอบที่ไม่มีข้อจำกัดด้านขนาดของบัฟเฟอร์ (buffer size) ของแต่ละสถานีทำงาน
4. การแก้ปัญหา ALB อาศัยการค้นหาแบบตามูเชิงพันทางที่พัฒนาขึ้น โดยโปรแกรม MATLAB เพื่อจัดสรรให้สายงานการประกอบมีความแปรปรวนของภาระงานน้อยที่สุด

5. ทำการแก้ปัญหา ALB แบบผลิตภัณฑ์เดียวจำนวน 4 ปัญหาที่ได้มาจากการสำรวจงานวิจัยที่เกี่ยวข้อง
6. ผลการแก้ปัญหา ALB ที่ได้จากการค้นหาแบบตามเชิงพันทางจะถูกนำไปเปรียบเทียบกับผลการแก้ปัญหา ALB ที่ได้จากวิธีการ COMSOAL โดยผ่านการจำลองด้วยคอมพิวเตอร์
7. การประเมินผลการทดสอบอาศัยค่าความแปรปรวนของภาระงานของสายงานการประกอบ

1.6 ข้อตกลงเบื้องต้น

1. ข้อมูลอินพุตจะประกอบด้วยชิ้นงาน เวลาการทำงานของแต่ละชิ้นงาน ลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน และรอบเวลาการผลิต
2. ทุกสถานีงานจะมีความสามารถในการทำงานที่เท่ากัน
3. เวลาการทำงานของแต่ละชิ้นงานจะมีค่าคงที่ ไม่ขึ้นกับลำดับการจัดชิ้นงาน และสถานีงาน
4. การจัดสมดุลสายงานการประกอบในวิทยานิพนธ์นี้ จะดำเนินการเพื่อให้ค่าความแปรปรวนของภาระงานน้อยที่สุด
5. การกำหนดชิ้นงานให้กับสถานีงานจะต้องไม่ขัดกับลำดับความสัมพันธ์ก่อนหลังของงาน
6. สถานีงานสามารถทำงานได้หลายชิ้นงาน แต่เวลารวมของการทำงานในแต่ละสถานีงานจะต้องไม่เกินรอบเวลาการผลิตที่กำหนดไว้
7. ชิ้นงานหนึ่งๆ จะต้องทำบนสถานีงานใดสถานีงานหนึ่งเท่านั้น ไม่สามารถแยกชิ้นงานไปทำในสถานีงานอื่นได้
8. จำนวนสถานีงานทั้งหมดในสายงานการประกอบจะต้องไม่น้อยกว่าจำนวนสถานีงานที่น้อยที่สุดที่เป็นไปได้ และจะต้องไม่เกินจำนวนสถานีงานสูงสุดที่ยอมรับได้
9. ชิ้นงานทุกชิ้นงานจะต้องถูกจัดสรรให้กับสถานีงานใดสถานีงานหนึ่งบนสายงานการประกอบ
10. ชิ้นงานแต่ละชิ้นงานสามารถถูกจัดสรรให้กับสถานีงานใดๆ โดยไม่มีข้อจำกัดในเรื่องของการจัดสรรพื้นที่ (zoning restriction)

1.7 ขั้นตอนของการดำเนินงานวิจัย

1. ทำการศึกษาและสำรวจวรรณกรรมและงานวิจัยที่เกี่ยวข้องกับทฤษฎีการจัดสมดุลสายงานการประกอบ เทคนิคการค้นหาแบบปัญญาประดิษฐ์ และการค้นหาแบบตาม
2. กำหนดปัญหา ALB เพื่อใช้เป็นปัญหาในวิทยานิพนธ์จำนวน 4 ปัญหาจากการสำรวจงานวิจัยที่เกี่ยวข้อง
3. ศึกษาวิธีการแก้ปัญหา ALB ด้วยวิธี COMSOAL

4. พัฒนาอัลกอริทึมและโปรแกรมการค้นหาแบบตามูเชิงพันทาง ที่มีการทำงานร่วมกันระหว่าง TS และเทคนิคการสลับที่แบบสุ่ม
5. ทำการทดสอบโปรแกรมการค้นหาแบบตามูเชิงพันทาง เพื่อกำหนดค่าพารามิเตอร์ของการค้นหาให้เหมาะสม
6. ดำเนินการแก้ปัญหา ALB ทั้ง 4 ปัญหาด้วยวิธี COMSOAL
7. ดำเนินการแก้ปัญหา ALB ทั้ง 4 ปัญหาด้วยการค้นหาแบบตามูเชิงพันทางที่นำเสนอ
8. เปรียบเทียบและอภิปรายผลการแก้ปัญหา ALB และเปรียบเทียบผลที่ได้จากวิธี COMSOAL และการค้นหาแบบตามูเชิงพันทาง
9. สรุปผลการแก้ปัญหา ALB ด้วยวิธีการที่นำเสนอ พร้อมให้ข้อเสนอแนะเพื่อเป็นแนวทางในการพัฒนางานวิจัยต่อไป

1.8 การจัดรูปเล่มวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วย 6 บท และ 4 ภาคผนวก นอกจากบทที่ 1 ซึ่งเป็นบทนำแล้วยังมีบทอื่นๆ ที่มีเนื้อหาโดยสรุปดังต่อไปนี้

บทที่ 2 การปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง กล่าวถึงการปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง เพื่อใช้เป็นข้อมูลอ้างอิง และเป็นแนวทางสำหรับการดำเนินงานวิจัยต่อไป

บทที่ 3 ปัญหาการจัดสมมูลสายงานการประกอบ กล่าวถึงทฤษฎีเกี่ยวกับการจัดสมมูลสายงานการประกอบ และการแก้ปัญหา ALB ด้วยวิธี COMSOAL

บทที่ 4 การค้นหาแบบตามูเชิงพันทาง กล่าวถึงการค้นหาแบบตามู การพัฒนาอัลกอริทึมของการค้นหาแบบตามูเชิงพันทางเพื่อการแก้ปัญหา ALB

บทที่ 5 ผลการทดสอบ กล่าวถึงการทดสอบการแก้ปัญหา ALB โดยการค้นหาแบบตามูเชิงพันทางที่พัฒนาขึ้น พร้อมให้การอภิปรายผลที่ได้เชิงเปรียบเทียบผลที่ได้จากวิธี COMSOAL

บทที่ 6 บทสรุปและข้อเสนอแนะ เป็นบทสรุปของงานวิจัย และข้อเสนอแนะ

ภาคผนวก ก. แสดงรายละเอียดปัญหา ALB ที่ใช้ในงานวิจัย

ภาคผนวก ข. แสดงรายละเอียดโปรแกรมการค้นหาแบบตามู

ภาคผนวก ค. แสดงรายละเอียดโปรแกรมการค้นหาแบบตามูเชิงพันทางเพื่อแก้ปัญหา ALB

ภาคผนวก ง. แสดงรายชื่อบทความวิจัยที่ได้รับการตีพิมพ์

บทที่ 2

ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 กล่าวนำ

ในบทที่ 2 นี้จะกล่าวถึงการปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องกับการจัดสมดุลสายงานการประกอบ การค้นหาแบบปัญญาประดิษฐ์ โดยให้ความสนใจการค้นหาแบบตาบู่เป็นหลัก และการประยุกต์การค้นหาแบบปัญญาประดิษฐ์เพื่อแก้ปัญหา ALB ผลจากการปริทัศน์วรรณกรรมจะถูกใช้เป็นข้อมูลอ้างอิง และเป็นแนวทางสำหรับการดำเนินงานวิจัยต่อไป

2.2 การจัดสมดุลสายงานการประกอบ

การจัดสมดุลของสายงานการประกอบในทางอุตสาหกรรมได้รับการนำเสนอเป็นครั้งแรกในช่วงต้นของปี ค.ศ. 1900 [2] และเป็นที่รู้จักกันในนามปัญหา ALB เป็นครั้งแรกในปี ค.ศ. 1955 [3] จากนั้นเป็นต้นมาการแก้ปัญหา ALB ได้รับการศึกษาวิจัยและแตกแขนงออกไปอย่างกว้างขวาง โดยอาจเริ่มต้นศึกษาจากบทความผลงานวิจัยเชิงทบทวนผลงานที่ผ่านมา อาทิเช่น [1],[3],[7],[14-22] แนวทางการแก้ปัญหา ALB ที่ปรากฏในวิจัยก่อนหน้าเป็นการพัฒนาอัลกอริทึมที่สามารถให้คำตอบที่เหมาะสมที่สุด ทั้งวิธีแบบแม่นยำ [5-6],[23-26] และวิธีการที่อาศัยอัลกอริทึมเชิงศึกษานานึก [7-8],[27-29] ดังรายละเอียดต่อไปนี้

2.2.1 สายงานการประกอบ

สายงานการประกอบเป็นการจัดรูปแบบของผังการประกอบ ซึ่งจะประกอบไปด้วยหน่วยการผลิตต่างๆ ที่เรียกว่าสถานีงาน ในระบบสายงานการประกอบแบบต่อเนื่อง ชิ้นส่วนผลิตภัณฑ์ที่จะทำการประกอบจะถูกลำเลียงหรือเคลื่อนย้ายมาตามสถานีงานต่างๆ เมื่อชิ้นส่วนหรือผลิตภัณฑ์ดังกล่าวแต่ละหน่วยเข้ามาสู่สถานีงานใดๆ แล้ว จะก่อให้เกิดชิ้นงานการประกอบ (assembly operation) ขึ้นในสถานีงานนั้นตามลำดับ และเมื่อหมดขั้นตอนการประกอบในสถานีงานนั้นแล้ว ชิ้นส่วนนั้นจะถูกเคลื่อนย้ายไปยังสถานีงานต่อไป ในขณะที่เดียวกัน ที่สถานีงานเดิม ก็จะมีชิ้นส่วนผลิตภัณฑ์หน่วยถัดไปเข้ามาแทน โดยทั่วไปสายงานการประกอบจะมีอยู่ด้วยกัน 3 รูปแบบคือ

1. สายงานการประกอบแบบผลิตภัณฑ์เดียว (single-model assembly line) เป็นสายงานการประกอบที่ใช้สำหรับการผลิตผลิตภัณฑ์เพียงชนิดเดียว และเป็นผลิตภัณฑ์ที่มีรูปแบบเดียว
2. สายงานการประกอบสำหรับผลิตภัณฑ์เป็นชุด (batch-model assembly line) เป็นสายงานการประกอบที่ใช้สำหรับการผลิตผลิตภัณฑ์ตั้งแต่ 2 ชนิดขึ้นไป ซึ่งผลิตภัณฑ์นั้นต้องมีความคล้ายกัน และสามารถผลิตบนสายงานการประกอบเดียวกันได้

3. สายงานการประกอบแบบผลิตภัณฑ์ผสม (mixed-model assembly line) เป็นสายงานการประกอบที่ใช้สำหรับการผลิตผลิตภัณฑ์ตั้งแต่ 2 ชนิดขึ้นไป โดยที่ผลิตภัณฑ์ต่างๆ จะเข้าสู่สายงานการประกอบปนกันโดยไม่มีการแบ่งว่าต้องดำเนินการผลิตผลิตภัณฑ์ชุดไหนก่อน

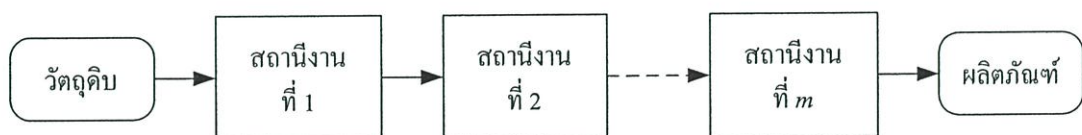
นอกจากนี้สายงานการประกอบยังสามารถแบ่งออกได้อีก 2 รูปแบบตามการดำเนินงานที่อ้างอิงกับรอบเวลาการผลิตดังนี้

1. Paced line เป็นสายงานการประกอบที่กำหนดให้เวลาดำเนินงานของสถานีงานต้องมีค่าเท่ากับรอบเวลาการผลิต ซึ่งหมายความว่าในทุกๆ รอบเวลาการผลิตนั้นชิ้นส่วนผลิตภัณฑ์จะต้องถูกส่งต่อไปยังสถานีงานต่อไปไม่ว่าจะทำงานในสถานีงานนั้นเสร็จทันหรือไม่ก็ตาม ถ้าหากงานในสถานีงานนั้นทำเสร็จก่อนรอบเวลาการผลิต ชิ้นส่วนผลิตภัณฑ์ก็ต้องคอยอยู่ในสถานีงานนั้นก่อนจนกว่าจะครบรอบเวลาการผลิต ซึ่งจะเห็นได้ว่าสายงานการประกอบแบบนี้อาจทำให้ได้ชิ้นส่วนผลิตภัณฑ์ที่ไม่สมบูรณ์ซึ่งจะต้องถูกนำไปซ่อมแซมและทำใหม่อีกครั้งหนึ่ง

2. Unpaced line (asynchronous line) เป็นสายงานการประกอบที่ในแต่ละสถานีงานจะทำงานตามชิ้นงานที่ถูกกำหนดให้จนกว่าจะเสร็จ แล้วจึงเคลื่อนย้ายงานไปทำยังสถานีทำงานถัดไป ดังนั้นเวลาดำเนินงานของแต่ละสถานีงานอาจมากกว่าหรือน้อยกว่ารอบเวลาการผลิต

สายงานการประกอบจะประกอบไปด้วยชิ้นงานต่างๆ มากมาย จึงจำเป็นต้องมีการจัดสรรหรือแบ่งงานให้กับสถานีงานต่างๆ เป็นกลุ่มๆ ไป การจัดสรรงานสามารถกระทำได้หลายวิธี เมื่องานถูกจัดสรรให้กับแต่ละสถานีงานอย่างสมดุล เวลาว่างงานในแต่ละสถานีงานจะมีค่าน้อย ส่งผลให้สายงานการประกอบมีประสิทธิภาพสูง

การจัดสมดุลของสายงานการประกอบ หรือที่เรียกอีกอย่างหนึ่งว่า การจัดสมดุลสายการผลิต เป็นปัญหาการกำหนดชิ้นงานให้กับหน่วยผลิตหนึ่ง ซึ่งเป็นลักษณะของการผลิตหรือประกอบผลิตภัณฑ์ในปริมาณมากๆ และค่อนข้างมีความสม่ำเสมอ สายการประกอบจะถูกแบ่งออกเป็นสถานีงานหลายๆ สถานีงานที่ทำงานต่อเนื่องกันดังแสดงในรูปที่ 2.1



รูปที่ 2.1 สายงานการประกอบ

การจัดสมดุลสายงานการประกอบเป็นงานที่อาจเกิดขึ้นในช่วงของการออกแบบการผลิตหรือเป็นงานในช่วงหลังการวางแผนการผลิตรวม ถ้าเกิดในช่วงของการออกแบบการผลิต หมายถึงกระบวนการผลิตนั้นเป็นแบบแน่นอน เครื่องจักรที่ใช้ส่วนมากเป็นเครื่องจักรชนิดพิเศษเพื่อทำการ

ผลิตสินค้าเฉพาะอย่าง ตำแหน่งของขั้นตอนการทำงานต่างๆ ส่วนใหญ่จะถูกกำหนดแน่นอนตามลำดับขั้นเป็นสายการผลิต ในกรณีของงานประกอบสายการผลิต หรือสายงานการประกอบงานในสถานีนงานต่างๆ อาจจะพอเปลี่ยนแปลงได้บ้างเพื่อให้สอดคล้องกับแผนการผลิต แต่ถ้าเป็นการผลิตทั่วไป ไม่ว่าจะแผนการผลิตจะมีการเปลี่ยนแปลงอย่างไร งานต่างๆ ในสถานีนงานจะไม่เปลี่ยนแปลง การจัดสายงานการประกอบในโรงงานที่มีการผลิตแบบต่อเนื่อง นับว่ามีความสำคัญมากในด้านการออกแบบโรงงาน โรงงานที่มีการจัดสายการประกอบอย่างมีประสิทธิภาพ จะต้องพยายามจัดสายการประกอบให้มีความสมดุล ซึ่งความหมายของการจัดสมดุลของสายงานการประกอบคือการพิจารณากำหนดงานหรือชิ้นงานต่างๆ ที่ใช้ในการประกอบผลิตภัณฑ์หรือสินค้าให้กับสถานีนงานหรือหน่วยผลิต โดยพยายามจัดสรรให้สถานีนงานต่างๆ มีภาระงานที่สมดุลกัน และจะต้องไม่ขัดกับลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน หรือข้อจำกัดในการผลิตที่มีอยู่ขณะเดียวกันก็ต้องสามารถผลิตผลิตภัณฑ์หรือสินค้าได้ตามอัตราที่ต้องการ

2.2.2 ประเภทของปัญหาการจัดสมดุลสายงานการประกอบ

ประเภทของปัญหา ALB สามารถถูกจำแนกและอธิบายได้หลายรูปแบบ ซึ่งในหัวข้อนี้จะกล่าวถึงประเภทของปัญหา ALB ที่ได้รับการจำแนกใน 2 รูปแบบใหญ่ๆ ดังนี้

(1) ประเภทของปัญหา ALB จำแนกตาม Ghosh และ Gagon [21]

ปัญหา ALB สามารถจำแนกตาม Ghosh และ Gagon [21] ได้ 4 ประเภทดังแสดงในรูปที่ 2.2 ซึ่งปัญหา ALB ทั้ง 4 ประเภทมีรายละเอียดดังนี้

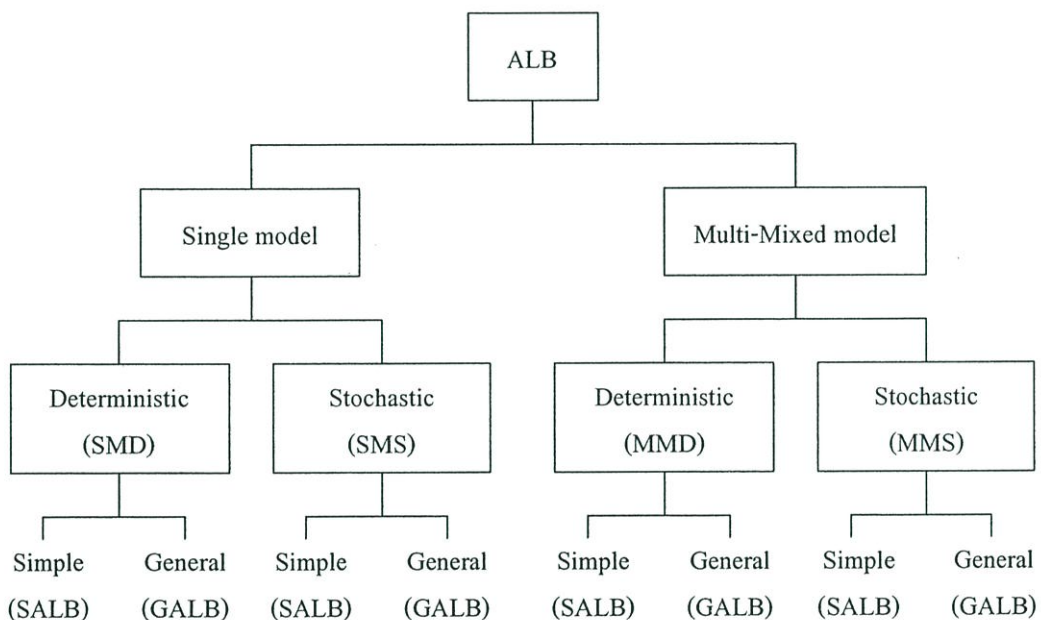
- (1) Single model deterministic (SMD)
- (2) Single model stochastic (SMS)
- (3) Multi/Mixed model deterministic (MMD)
- (4) Multi/Mixed model stochastic (MMS)

- **SMD** คือปัญหา ALB แบบดั้งเดิมที่มีรูปแบบปัญหาง่ายที่สุด และมีผู้สนใจนำมาศึกษาวิจัยมากที่สุด [21] ลักษณะเด่นของปัญหาประเภทนี้ คือเป็นการจัดสมดุลสายงานการประกอบสำหรับการผลิตภัณฑ์เดียว และเวลาดำเนินงานของชิ้นงานถูกกำหนดไว้อย่างแน่นอน

- **SMS** คือปัญหา ALB แบบผลิตภัณฑ์เดียวเช่นเดียวกับ SMD แต่ต่างกันตรงที่เวลาดำเนินงานของชิ้นงานนั้นมีความไม่คงที่ ซึ่งปัญหาประเภทนี้จะใกล้เคียงกับปัญหาจริงมากกว่า ในกรณีที่เวลาดำเนินงานของชิ้นงานมีความไม่คงที่ จำเป็นต้องพิจารณาถึงสิ่งเกี่ยวข้องอื่นๆ ที่ตามมา เช่น เวลาดำเนินงานของสถานีนงานมากกว่ารอบเวลาการผลิต การผลิตชิ้นส่วนที่ไม่สมบูรณ์ ขนาดพื้นที่ของวัสดุคงคลัง (inventory buffer) เป็นต้น

- **MMD** คือปัญหา ALB แบบผลิตภัณฑ์ผสม หรือแบบหลายผลิตภัณฑ์ (มากกว่าหนึ่งผลิตภัณฑ์) โดยมีการกำหนดเวลาดำเนินงานของชิ้นงานไว้อย่างแน่นอน โดยทั่วไปแล้วผลิตภัณฑ์ที่ถูกผลิตในสายงานการประกอบประเภทนี้มีก็มีชิ้นงานที่คล้ายกัน ดังนั้นจึงมักทำการออกแบบสายงานการประกอบเพียงแบบเดียวเพื่อรองรับผลิตภัณฑ์ดังกล่าว

- **MMS** คือปัญหา ALB แบบผลิตภัณฑ์ผสม หรือแบบหลายผลิตภัณฑ์ ที่ซึ่งเวลาดำเนินงานของชิ้นงานมีค่าไม่แน่นอน สายงานการประกอบประเภทนี้มีความยุ่งยากและซับซ้อนมากที่สุด เพราะต้องคำนึงถึงองค์ประกอบที่เกี่ยวข้องอื่นๆ เช่น ผลกระทบทางการเรียนรู้และระดับทักษะ ความชำนาญของคนงาน การออกแบบงาน นอกจากนั้นแล้วสายงานการประกอบประเภทนี้ยังต้องการการจัดสมดุลใหม่บ่อยครั้งตามการเปลี่ยนแปลงของผลิตภัณฑ์



รูปที่ 2.2 ปัญหา ALB จำแนกตาม Ghosh และ Gagon [21]

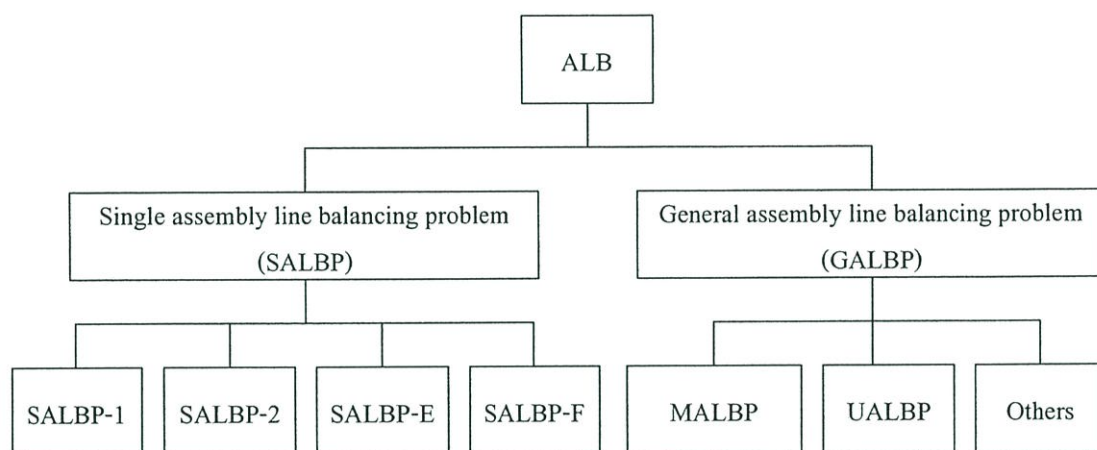
จากการจำแนกปัญหา ALB ตาม Ghosh และ Gagon [21] ทั้ง 4 ประเภทข้างต้น ยังสามารถแบ่งได้อีก 2 ลักษณะดังแสดงในรูปที่ 2.2 ซึ่งมีรายละเอียดดังนี้

- **Simple case (SALB)** คือปัญหา ALB แบบง่าย (simple assembly line balancing: SALB) เป็นปัญหาการกำหนดชิ้นงานต่างๆ จำนวน n ชิ้นงานให้กับสถานีงานจำนวน m สถานีงาน โดยที่เวลาดำเนินงานของชิ้นงานแต่ละชิ้นงานคือ t_i ในหน่วยเวลา และมีความสัมพันธ์ก่อนหลังของชิ้นงานต่างๆ รวมทั้งรอบเวลาการผลิตเป็นข้อจำกัดหรือเงื่อนไขที่จะต้องพิจารณาประกอบ ผลของการจัดสรรชิ้นงานให้กับแต่ละสถานีงานจะต้องไม่ขัดกับความสัมพันธ์ก่อนหลังของชิ้นงาน และเวลาดำเนินงานของสถานีงานในแต่ละสถานีงานจะต้องไม่มากเกินไปกว่ารอบเวลาการผลิตที่กำหนด

- **General case (GALB)** คือ ปัญหา ALB แบบทั่วไป (general assembly line balancing: GALB) เป็นปัญหาที่มีข้อจำกัดหรือปัจจัยอื่นๆ เพิ่มเติมมากกว่าปัญหาแบบ SALB เช่น มีสถานีงานแบบขนาน (parallel station) มีการพิจารณาขนาดของวัสดุคงคลังในแต่ละสถานีงาน และมีข้อจำกัดเกี่ยวกับการจัดรวมกลุ่มชิ้นงานไว้ในพื้นที่เฉพาะ เป็นต้น

(2) ประเภทของปัญหา ALB จำแนกตาม Scholl และ Becker [15],[30]

ปัญหา ALB สามารถจำแนกตาม Scholl และ Becker [15],[30] ได้ดังแสดงในรูปที่ 2.3 ซึ่งมีรายละเอียดดังนี้



รูปที่ 2.3 ปัญหา ALB จำแนกตาม Scholl และ Becker [15],[30]

- **SALBP** คือปัญหา ALB แบบง่าย (simple assembly line balancing problem) ซึ่งเป็นปัญหาแบบผลิตภัณฑ์เดียว ปัญหาประเภทนี้จะพิจารณาเฉพาะเงื่อนไขลำดับความสัมพันธ์ก่อนหลังของชิ้นงานระหว่างชิ้นงาน ซึ่งสามารถจำแนกรูปแบบปัญหาประเภทนี้ได้เป็น 4 รูปแบบด้วยกันดังนี้

- **Type 1 (SALBP-1)** ปัญหา ALB ในลักษณะนี้จะเป็นปัญหาการจัดสรรชิ้นงานเข้าสู่สถานีงานเพื่อให้จำนวนของสถานีงานมีจำนวนน้อยที่สุด โดยจะต้องกำหนดอัตราการผลิตหรือรอบเวลาการผลิตมาให้

- **Type 2 (SALBP-2)** ปัญหา ALB ในลักษณะนี้จะเป็นปัญหาการจัดสรรชิ้นงานเข้าสู่สถานีงานเพื่อให้รอบเวลาการผลิตมีค่าน้อยที่สุด (อัตราการผลิตมีค่ามากที่สุด) โดยจะต้องกำหนดจำนวนของสถานีงานมาให้

- **Type E (SALBP-E)** ปัญหา ALB ในลักษณะนี้จะเป็นปัญหาการจัดสรรชิ้นงานเข้าสู่สถานีงานเพื่อให้ประสิทธิภาพของสายงานการประกอบมีค่ามากที่สุด ในขณะที่เดียวกันจะต้องทำให้อัตราการผลิตและจำนวนสถานีงานมีค่าน้อยที่สุด

▪ **Type F (SALBP-F)** ปัญหา ALB ในลักษณะนี้จะเป็นปัญหาการจัดสรรชิ้นงานเข้าสู่สถานีงานเพื่อให้สายงานการประกอบเกิดความสมดุลของชิ้นงานในแต่ละสถานีงานขึ้นจริงๆ โดยจะต้องกำหนดจำนวนสถานีงานและรอบเวลาการผลิตมาให้ร่วมกัน

• **GALBP** คือปัญหา ALB แบบทั่วไป (general assembly line balancing problem) ซึ่งเป็นปัญหา ALB ที่มีขนาดใหญ่และมีเงื่อนไขเกี่ยวกับการปฏิบัติจริงเข้ามาเกี่ยวข้อง เช่น การเลือกอุปกรณ์ ข้อจำกัดในการกำหนดชิ้นงาน การเลือกเวลาในการดำเนินงาน เป็นต้น ซึ่งสามารถจำแนกรูปแบบปัญหาประเภทนี้ได้เป็น 2 รูปแบบด้วยกันดังนี้

▪ **MALBP** (multi/mixed model assembly line balancing problems) เป็นปัญหาการจัดสมดุลสายงานการประกอบแบบผลิตภัณฑ์ผสม หรือแบบหลายผลิตภัณฑ์ (มากกว่าหนึ่งผลิตภัณฑ์) เพื่อจัดสรรชิ้นงานเข้าสู่สถานีงานต่างๆ โดยการพิจารณาเวลาดำเนินงานของชิ้นงานที่แตกต่างกันสำหรับผลิตภัณฑ์ที่ต่างกัน ทั้งนี้เพื่อให้จำนวนสถานีงานและรอบเวลาการผลิตมีค่าน้อยที่สุดรวมทั้งทำให้เกิดสมดุลในสายงานการประกอบด้วย [14] ปัญหา ALB แบบนี้จะมีความซับซ้อนกว่าปัญหา ALB แบบผลิตภัณฑ์เดียวเนื่องจากเวลาดำเนินงานของสถานีงานแต่ละสถานีต้องมีความต่อเนื่องและราบรื่น ซึ่งผู้อ่านสามารถศึกษาเพิ่มเติมได้จาก [14] ,[31-32]

▪ **UALBP** (U-shaped assembly line balancing problems) เป็นปัญหาการจัดสมดุลสายงานการประกอบที่มีรูปทรงของสายงานเป็นตัวอักษร U ซึ่งสถานีงานต่างๆ จะถูกจัดเรียงตามรูปทรงตัวอักษร U แคบๆ ผู้ปฏิบัติงานหรือเครื่องจักรที่ใช้ประกอบชิ้นงานจะสามารถทำงานได้ทั้ง 2 ด้านของสายงานรูปทรงดังกล่าวได้พร้อมกัน ทั้งชิ้นงานที่มาก่อนหน้าและชิ้นงานที่มีทีหลัง ซึ่งเงื่อนไขบางอย่างในการจัดสมดุลสายงานประเภทนี้จะถูกตัดแปลงให้เหมาะสม ซึ่งผู้อ่านสามารถศึกษาเพิ่มเติมได้จาก [22],[33-34]

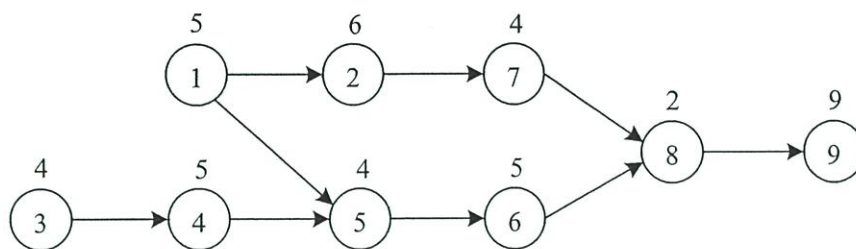
2.2.3 ข้อมูลพื้นฐานที่จำเป็นต้องทราบในการแก้ปัญหาการจัดสมดุลสายงานการประกอบ

ในการแก้ปัญหา ALB ข้อมูลพื้นฐานที่จำเป็นต้องทราบได้แก่

1. ข้อมูลแสดงขั้นตอนการทำงานต่างๆ ซึ่งบอกให้ทราบถึงลำดับก่อนหลังของชิ้นงานต่างๆ โดยอาจเขียนเป็นแผนภูมิลำดับงานก่อนหลัง (precedence diagram) ดังแสดงในรูปที่ 2.4 ซึ่งความหมายของโนดหรือวงกลมคือชิ้นงานแต่ละชิ้นงาน น้ำหนักของโนดแสดงเวลาดำเนินงานของชิ้นงาน และลูกศรแสดงลำดับชิ้นงานที่จะต้องดำเนินการตามลำดับทิศทางของลูกศร

2. ข้อมูลแสดงเวลาดำเนินงานของชิ้นงานต่างๆ ซึ่งควรเป็นเวลามาตรฐาน (standard time) ของชิ้นงานนั้นๆ ดังแสดงด้วยน้ำหนักของโนดดังรูปที่ 2.4 น้ำหนักดังกล่าวแสดงด้วยตัวเลขที่เขียนอยู่เหนือวงกลมแต่ละวงในหน่วยเวลา

3. อัตราการผลิตที่ต้องการ เพื่อนำไปใช้ในการคำนวณหารอบเวลาการผลิตที่เหมาะสม



รูปที่ 2.4 แผนภูมิลำดับงานก่อนหลัง

2.2.4 ขั้นตอนในการจัดสมดุลของสายงานการประกอบ

การจัดสมดุลของสายงานการประกอบ ประกอบด้วยขั้นตอนต่างๆ ดังต่อไปนี้

1. กำหนดชิ้นงานต่างๆ ซึ่งเป็นสิ่งที่บอกให้ทราบถึงลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน
2. กำหนดเวลาดำเนินงานของชิ้นงาน ซึ่งควรเป็นเวลามาตรฐานของชิ้นงานนั้นๆ
3. กำหนดรอบเวลาการผลิตจากอัตราการผลิตที่กำหนดหรือที่ต้องการ
4. กำหนดหาจำนวนสถานีงานที่น้อยที่สุดที่ต้องการจากรอบเวลาการผลิต
5. เลือกชิ้นงานมาจัดลงแต่ละสถานีงาน โดยพิจารณาถึงลำดับความสัมพันธ์ก่อนหลังของชิ้นงานต่างๆ สิ่งที่จะต้องพิจารณาคือเวลาดำเนินงานของสถานีงานแต่ละสถานีงานจะต้องไม่เกินรอบเวลาการผลิตที่กำหนดไว้

2.2.5 เทคนิควิธีและอัลกอริทึมในการแก้ปัญหาการจัดสมดุลสายงานการประกอบ

เทคนิควิธีและอัลกอริทึมในการแก้ปัญหา ALB สามารถแบ่งได้เป็นวิธีแบบแม่นยำ (exact method) และวิธีการที่อาศัยอัลกอริทึมเชิงศึกษาคำนึง (heuristic method) [21] ดังต่อไปนี้

(1) วิธีแม่นยำ ได้แก่ Linear programming (LP), Integer programming (IP), Dynamic programming (DP), Goal programming (GP), Shortest-path technique (SP), Maximal-path technique (MP) และ Branch and bound (BB) เป็นต้น

(2) วิธีศึกษาคำนึง ได้แก่ Priority ranking and assignment, Tree search (heuristic BB), Trade and transfer, Random sampling, Computer method of sequencing operations for assembly line (COMSOAL), Genetic algorithm (GA), Tabu search (TS), Artificial neural network (ANN), และ Ant colony optimization (ACO) เป็นต้น

2.2.6 เกณฑ์วัตถุประสงค์ในการจัดสมดุลสายงานการประกอบ

เกณฑ์วัตถุประสงค์ (objective criteria) ในการจัดสมดุลสายงานการประกอบถือเป็นเกณฑ์ที่ใช้ในการประเมินประสิทธิภาพสายงานการประกอบที่ได้รับการจัดสมดุล โดยสามารถแบ่งได้ 2

ลักษณะคือ เกณฑ์วัตถุประสงค์ทางเทคนิค (technical objective criteria) และเกณฑ์วัตถุประสงค์ทางเศรษฐศาสตร์ (economic objective criteria) [21] ดังรายละเอียดต่อไปนี้

(1) เกณฑ์วัตถุประสงค์ทางเทคนิค

1. การทำให้จำนวนสถานีงานมีจำนวนน้อยที่สุด เมื่อกำหนดรอบเวลาการผลิตมาให้
2. การทำให้รอบเวลาการผลิตมีค่าน้อยที่สุด เมื่อกำหนดจำนวนสถานีงานมาให้
3. การทำให้เวลารอรวมมีค่าน้อยที่สุด
4. การทำให้ความแปรปรวนของภาระงาน (workload variance) มีค่าน้อยที่สุด
5. การทำให้ประสิทธิภาพของสายงานการประกอบมีค่ามากที่สุด
6. การทำให้เวลาดังแต่นำชิ้นงานหรือวัสดุเข้าสู่สายงานการประกอบจนกระทั่งนำผลิตภัณฑ์ออกจากสายงานมีค่าน้อยที่สุด
7. การทำให้ความน่าจะเป็นที่เวลาดำเนินงานของสถานีงานมีค่ามากกว่ารอบเวลาการผลิตมีค่าน้อยที่สุด

(2) เกณฑ์วัตถุประสงค์ทางเศรษฐศาสตร์

1. การทำให้ค่าใช้จ่ายรวมที่เกิดขึ้นเนื่องจากแรงงาน สถานีทำงาน และงานที่ไม่เสร็จสมบูรณ์มีค่าน้อยที่สุด
2. การทำให้ค่าแรงงานต่อหน่วยการผลิตมีค่าน้อยที่สุด
3. การทำให้ค่าปรับรวมของงานที่ไม่มีประสิทธิภาพมีค่าน้อยที่สุด
4. การทำให้ค่าวัสดุคงคลัง ค่าปรับตั้งเครื่อง และค่าใช้จ่ายในการว่างงานมีค่าน้อยที่สุด
5. การทำให้ค่าวัสดุคงคลังในระหว่างการผลิตรวมมีค่าน้อยที่สุด
6. การทำให้กำไรสุทธิมีค่ามากที่สุด

ในการศึกษาวิจัยการแก้ปัญหา ALB ส่วนใหญ่จะใช้เกณฑ์วัตถุประสงค์ทางเทคนิค ซึ่งรูปแบบที่ถูกนำมาใช้มากที่สุดคือ การทำให้จำนวนสถานีงานมีจำนวนน้อยที่สุด เมื่อกำหนดรอบเวลาการผลิตมาให้ และการทำให้รอบเวลาการผลิตมีค่าน้อยที่สุด เมื่อกำหนดจำนวนสถานีงานมาให้ [21] สำหรับรายละเอียดของปัญหา ALB ที่ใช้ในวิทยานิพนธ์นี้จะกล่าวถึงในบทที่ 3

2.3 การค้นหาแบบปัญญาประดิษฐ์

การค้นหาแบบปัญญาประดิษฐ์ที่ได้รับการนำมาประยุกต์ใช้ในอุตสาหกรรม ได้แก่ จินเนติก อัลกอริทึม (GA) การหาค่าเหมาะที่สุดด้วยวิธีกลุ่มอนุภาค (particle swarm optimization: PSO) การค้นหาแบบมด (ant search) และการค้นหาแบบตาวู (TS) เป็นต้น แต่จากการสำรวจงานวิจัยที่เกี่ยวข้องพบว่า TS ได้รับการประยุกต์ใช้ในอุตสาหกรรมอย่างกว้างขวาง มีอัลกอริทึมที่ไม่ซับซ้อน อีกทั้งยังได้รับการพิสูจน์ว่ามีคุณสมบัติของการค้นหาค่าตอบวงกว้าง ในวิทยานิพนธ์นี้จะให้ความสำคัญที่จะประยุกต์ใช้ TS เพื่อแก้ปัญหา ALB ดังนั้นการปริทัศน์วรรณกรรมที่จะกล่าวถึงใน

หัวข้อนี้จึงเกี่ยวกับความเป็นมาและการประยุกต์ใช้งาน TS ดังรายละเอียดต่อไปนี้

Glover [35-37] ได้นำเสนอการค้นหาแบบตาบอด (tabu search: TS) ขึ้นเป็นครั้งแรก เมื่อปี ค.ศ. 1986 ในฐานะเทคนิคการค้นหาแบบปัญญาประดิษฐ์ เพื่อใช้ในการค้นหาคำตอบที่เหมาะสมที่สุดสำหรับปัญหาแบบไม่ต่อเนื่องเชิงผสมผสาน TS อาศัยกระบวนการค้นหาค่าใกล้เคียง (neighborhood search) และรายชื่อตาบอด (tabu list: TL) เพื่อทำหน้าที่เก็บคำตอบในอดีต คำตอบใน TL สามารถใช้ประโยชน์ในกรณีการค้นหาเกิดการล๊อคโดยคำตอบวงแคบเฉพาะถิ่น โดยอาศัยเกณฑ์ปรารถนา หรือเกณฑ์ทะเยอทะยาน (aspiration criteria: AC) กระบวนการค้นหาจะเคลื่อนที่อย่างมีระเบียบแบบแผน (deterministic) โดยทั่วไปจะถูกกำกับไว้ด้วยเงื่อนไขความคงอยู่ล่าสุด (recency) และเงื่อนไขความซ้ำ (frequency)

Glover และ Laguna [38] ได้เสนอการค้นหาแบบตาบอด หรือ TS โดยการปรับปรุงให้กระบวนการค้นหามีประสิทธิภาพมากยิ่งขึ้น โดยการเพิ่มกลไกอีกสองกลไก คือ กลไกความเข้มข้น (intensification) และกลไกความหลากหลาย (diversification) สำหรับกลไกความเข้มข้นจะเป็นการค้นหาคำตอบในกลุ่มคำตอบที่พบแล้วในอดีต ที่ได้รับการบันทึกไว้ใน TL ซึ่งกลุ่มคำตอบดังกล่าวได้รับการพิจารณาแล้วว่าเป็นกลุ่มคำตอบที่ดี กลไกความเข้มข้นจะกำหนดให้กลไกการค้นหาดำเนินการค้นหาคำตอบซ้ำในพื้นที่เดิมที่เคยพบกลุ่มคำตอบที่ดีอย่างละเอียดมากขึ้น ซึ่งอาจส่งผลทำให้การค้นหาพบคำตอบที่ดีกว่าได้ สำหรับกลไกความหลากหลาย จะเป็นการค้นหาคำตอบในปริภูมิหรือพื้นที่การค้นหาที่ยังไม่เคยได้รับการค้นหามาก่อน กลไกความหลากหลายจะกำหนดให้กลไกการค้นหาดำเนินการค้นหาคำตอบในพื้นที่อื่น หรือทิศทางอื่นที่แตกต่างไปจากเดิม ซึ่งอาจส่งผลทำให้การค้นหาสามารถพบคำตอบที่ดีกว่าได้เช่นกัน

Glover และ Hanafi [39-40] ได้นำเสนอผลการพิสูจน์การลู่เข้าหาคำตอบวงกว้าง หรือเรียกว่า การพิสูจน์คุณสมบัติการลู่เข้าหาผลเฉลยที่เหมาะสมของ TS โดยอาศัยทฤษฎีบทและการพิสูจน์ทางคณิตศาสตร์ ผวนกับลอจิกที่เหมาะสม ซึ่งพบว่า TS มีคุณสมบัติการลู่เข้าหาคำตอบวงกว้าง

Glover และคณะ [41] ได้ประยุกต์ใช้ TS เพื่อแก้ปัญหาการควบคุมอัตราการผลิตแปรผันของการจัดการด้านการเงินที่มีการผันแปรแบบสุ่ม (stochastic) ที่ระดับต่างๆ กัน ซึ่งพบว่า TS สามารถให้คำตอบต่อปัญหาดังกล่าวได้เป็นอย่างดี

Kulworawanichpong และ Sujitjorn [42] ได้ประยุกต์ใช้ TS เพื่อวางแผนการผลิตและการจ่ายไฟฟ้ากำลัง ซึ่งพบว่า TS สามารถดำเนินการตัดสินใจเพื่อการวางแผนผลิตและจำหน่าย รวมทั้งจัดสรรอัตราการผลิตของกำลังไฟฟ้าได้อย่างเหมาะสม

Yao และคณะ [43] ได้ประยุกต์ใช้ TS เพื่อทำการระบุเอกลักษณ์ระบบพลวัต (dynamic system identification) ซึ่งระบบที่ใช้ในงานวิจัยนี้ คือระบบกระตุ้นเครื่องกำเนิดไฟฟ้า จากผลการวิจัยพบว่า TS สามารถค้นหาค่าพารามิเตอร์ของระบบดังกล่าวได้อย่างถูกต้อง

Nowicki และ Smutnicki [44] ได้นำเอา TS ไปแก้ปัญหาลำโพง flow shop ซึ่งเป็นกระบวนการที่สำคัญกระบวนการหนึ่งในสายงานการผลิตทางอุตสาหกรรม ซึ่งผลการวิจัยพบว่า TS สามารถแก้ปัญหาลำโพง flow shop ได้อย่างน่าพอใจ

นอกจากนี้ยังมีคณะผู้วิจัยอีกกลุ่มหนึ่งที่ทำให้ความสนใจในเรื่องของการพัฒนาความสามารถในการค้นหาคำตอบของ TS ให้มีประสิทธิภาพมากยิ่งขึ้น เช่น การค้นหาแบบตามูแบบรีแอกทีฟ (reactive tabu search: RTS) ที่ได้รับการพัฒนาและนำเสนอโดย Battiti [45-46] การค้นหาแบบตามูเชิงน่าจะเป็น (probabilistic tabu search: PTS) ที่ได้รับการพัฒนาและนำเสนอโดย Kochetov และ Goncharov [47] และการค้นหาแบบตามูเชิงปรับตัว (adaptive tabu search: ATS) ที่ได้รับการพัฒนาและนำเสนอโดย Sujitjorn และคณะ [48-50] สำหรับรายละเอียดของ TS ที่ใช้ในงานวิจัยวิทยานิพนธ์นี้จะกล่าวถึงในบทที่ 4

2.4 การประยุกต์การค้นหาแบบปัญญาประดิษฐ์เพื่อแก้ปัญหาลำโพง ALB

จากการสำรวจงานวิจัยที่เกี่ยวข้อง การค้นหาแบบปัญญาประดิษฐ์ที่ถูกนำมาประยุกต์ใช้เพื่อแก้ปัญหาลำโพง ALB ได้แก่ GA TS การค้นหาแบบมด และเครือข่ายประสาทเทียม (neural network) นอกจากนี้ยังมีการประยุกต์การค้นหาแบบปัญญาประดิษฐ์ร่วมกับวิธีการแบบแม่นยำตรง ซึ่งอาจเรียกว่าวิธีการแบบพันทาง (hybrid approach) เพื่อแก้ปัญหาลำโพง ALB ดังรายละเอียดต่อไปนี้

Arcus [8] ได้ทำการวิจัยและคิดค้นอัลกอริทึมที่ใช้กับปัญหาลำโพง ALB และนำเสนออัลกอริทึมดังกล่าวภายใต้ชื่อวิธี COMSOAL ผลการวิจัยพบว่าวิธี COMSOAL สามารถนำมาใช้ในการแก้ปัญหาลำโพง ALB อย่างง่ายและปัญหาลำโพง ALB ที่มีสถานีนงานแบบขนาน หรือปัญหาลำโพง ALB ที่มีการกำหนดงานเฉพาะให้กับสถานีนงานบางสถานีนงานไว้ล่วงหน้าได้เป็นอย่างดี

กรรณิกา ศิลาพันธ์ [9] ได้นำเอา GA ไปประยุกต์ใช้แก้ปัญหาลำโพง ALB แบบผลิตภัณฑ์เดียว โดยการกำหนดฟังก์ชันวัตถุประสงค์เป็นแบบหลายวัตถุประสงค์ จากนั้นได้ทำการเปรียบเทียบผลการแก้ปัญหาลำโพง ALB กับวิธี COMSOAL ซึ่งจากผลการวิจัยพบว่า GA สามารถให้คำตอบต่อการแก้ปัญหาลำโพง ALB แบบผลิตภัณฑ์เดียวได้ดีกว่าวิธี COMSOAL

Rubinovitz และ Levitin [10] ได้นำเสนองานวิจัยที่พัฒนาและทดสอบเกี่ยวกับการนำเอา GA มาใช้สำหรับแก้ปัญหาลำโพง ALB โดยเปรียบเทียบกับผลที่ได้จากวิธีศึกษานักแบบดั้งเดิม จากการวิจัยพบว่าการนำเอา GA ไปทำงานร่วมกับเทคนิค local optimization crossover และ mutation อย่างง่ายจะให้ผลการแก้ปัญหาลำโพง ALB ที่ดี และสามารถแก้ปัญหาลำโพง ได้เร็วกว่าวิธีศึกษานักแบบดั้งเดิมเมื่อปัญหาลำโพง ALB มีจำนวนสถานีนงานมากและมีความยืดหยุ่นสูง

Chiang [11] ได้ประยุกต์ใช้ TS สำหรับแก้ปัญหาลำโพง ALB โดยที่ TS ได้รับการพัฒนาให้มีกลไกการค้นหาที่แตกต่างกัน 4 ลักษณะ ปัญหาลำโพง ALB ที่ใช้ในการทดสอบมาจากการสำรวจงานวิจัยที่

เกี่ยวข้อง ผลการคำนวณเชิงตัวเลขพบว่า TS สามารถแก้ปัญหา ALB ได้อย่างมีประสิทธิภาพ โดยที่ TS จะให้คำตอบที่เหมาะสมที่สุด

จงกล เอี่ยมมิ [12] ได้นำเอา GA ไปประยุกต์ใช้แก้ปัญหา ALB แบบผลิตภัณฑ์ผสม โดยทำการเปรียบเทียบผลการแก้ปัญหา ALB กับวิธี COMSOAL ซึ่งจากผลการวิจัยพบว่า GA สามารถให้คำตอบต่อการแก้ปัญหา ALB แบบผลิตภัณฑ์ผสมได้ดีกว่าวิธี COMSOAL

Yogathasan [51] ได้ทำการศึกษาความเป็นไปได้ในการประยุกต์เครือข่ายประสาทเทียมมาใช้ในการแก้ปัญหา ALB โดยได้ทำการพัฒนาอัลกอริทึมที่เรียกว่า cascaded Hopfield network ขึ้นมา และทำการทดสอบวิธีการดังกล่าวเปรียบเทียบกับวิธี Hopfield network ที่มีอยู่เดิม ผลปรากฏว่าวิธีที่นำเสนอให้คำตอบที่ดีกว่า

Mapfaira และ Byrne [52] ได้ทำการวิจัยเกี่ยวกับการแก้ปัญหา ALB โดยการประยุกต์ GA เพื่อให้มีจำนวนสถานีงานน้อยที่สุด โดยใช้ประชากรเริ่มต้นที่เป็นไปได้แทนการกำหนดประชากรแบบสุ่ม และทำการศึกษาเงื่อนไขที่ทำให้ได้คำตอบที่ดีที่สุดเมื่อนำเอา GA มาใช้ นอกจากนี้ยังทำการศึกษาผลของขนาดประชากรที่มีต่อสมรรถภาพของ GA อีกด้วย ผลการวิจัยพบว่าการใช้ประชากรเริ่มต้นที่เป็นไปได้แทนการกำหนดประชากรแบบสุ่มทำให้ได้คำตอบในการแก้ปัญหา ALB ที่ดีที่สุดได้ นอกจากนี้ในงานวิจัยยังชี้ให้เห็นว่าสมรรถภาพของ GA จะเพิ่มมากขึ้นอย่างสม่ำเสมอตามขนาดของประชากร ขณะเดียวกันจะใช้เวลาในการหาคำตอบนานขึ้น

Suresh และคณะ [53] ได้ทดลองนำเอา GA มาใช้แก้ปัญหา ALB ในงานวิจัยดังกล่าวได้เสนอวิธี GA 2 รูปแบบ กล่าวคือ รูปแบบที่ 1 กำหนดกลุ่มของประชากรกลุ่มเดียวซึ่งยอมให้มีคำตอบที่เป็นไปไม่ได้เกิดขึ้นจำนวนหนึ่ง และรูปแบบที่ 2 กำหนดกลุ่มประชากรคำตอบมีลักษณะแตกต่างกันโดยแบ่งเป็น 2 กลุ่ม กลุ่มแรกจะไม่ยอมให้มีคำตอบที่เป็นไปไม่ได้เกิดขึ้น ส่วนกลุ่มที่สองจะยอมให้มีคำตอบที่เป็นไปไม่ได้เกิดขึ้นจำนวนหนึ่ง ผลงานวิจัยสรุปว่าวิธีการแบบหลังสามารถให้คำตอบในการแก้ปัญหา ALB ได้ดีกว่าแบบแรกในกรณีที่ปัญหา ALB มีขนาดใหญ่

Leu และคณะ [54] ได้นำเอา GA ไปประยุกต์ใช้แก้ปัญหา ALB และได้ทำเปรียบเทียบคำตอบกับวิธีศึกษาสำนึกแบบดั้งเดิม ผลการวิจัยชี้ให้เห็นอีกครั้งหนึ่งว่า GA สามารถแก้ปัญหา ALB ได้อย่างมีประสิทธิภาพและมีความยืดหยุ่นสูง

Pastor และ Corominas [55] ได้ประยุกต์ใช้ TS เพื่อแก้ปัญหา ALB ที่มีเงื่อนไขของจำนวนสถานีงานไม่เท่ากัน รวมทั้งภาระของสถานีงานต่างๆ ถูกกำหนดไว้ล่วงหน้า ผลการทดสอบถูกนำไปเปรียบเทียบกับวิธีการศึกษาสำนึกแบบดั้งเดิม ซึ่งผลการทดสอบพบว่า TS สามารถให้คำตอบในการแก้ปัญหา ALB ได้ดีกว่า

Chen และคณะ [56] ได้นำเสนอวิธี GA แบบพันทางเพื่อแก้ปัญหา ALB โดยการพัฒนาวิธีการปรับตัวเองสำหรับ โครโมโซมหรือคำตอบในแต่ละรุ่น วิธีการที่นำเสนอจะช่วยให้พบคำตอบที่

ดีกว่าภายในเวลาที่รวดเร็ว โดยสามารถแก้ปัญหา ALB ที่กำหนดให้ฟังก์ชันวัตถุประสงค์เป็นแบบหลายวัตถุประสงค์ได้ดีกว่าวิธีศึกษาสำนักแบบดั้งเดิม

Bautista และ Pereira [57] ได้ประยุกต์ใช้การค้นหาแบบมดเพื่อแก้ปัญหา ALB ที่มีข้อกำหนดเงื่อนไขด้านเวลาและพื้นที่เข้ามาเกี่ยวข้อง ผลการแก้ปัญหาถูกนำไปเปรียบเทียบกับวิธีศึกษาสำนักแบบดั้งเดิม ซึ่งพบว่าวิธีการค้นหาแบบมดสามารถพบคำตอบสำหรับปัญหา ALB ที่สนใจได้ดีกว่าวิธีศึกษาสำนักแบบดั้งเดิม

จากผลการสำรวจงานวิจัยที่เกี่ยวข้องกับการนำเอาเทคนิคการค้นหาแบบปัญญาประดิษฐ์มาประยุกต์ใช้เพื่อแก้ปัญหา ALB ที่ผ่านมาจะพบว่างานวิจัยส่วนใหญ่ได้ประยุกต์ใช้ GA เพื่อการแก้ปัญหา สำหรับการประยุกต์ใช้ TS ในการแก้ปัญหา ALB ยังมีไม่ค่อนแพร่หลายมากนัก แต่จากการสำรวจงานวิจัยพบว่า TS สามารถค้นหาคำตอบที่เหมาะสมได้เร็วกว่า GA ส่วนการประยุกต์ใช้การค้นหาแบบมดและเครือข่ายประสาทเทียมในการแก้ปัญหา ALB ยังไม่ได้รับการสนใจจากนักวิจัยเท่าที่ควร

2.5 สรุป

บทที่ 2 นี้ได้รายงานผลการปรัทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องกับปัญหา ALB การค้นหาแบบปัญญาประดิษฐ์โดยเน้นที่การค้นหาแบบตามูหรือ TS เป็นหลัก และการประยุกต์การค้นหาแบบปัญญาประดิษฐ์เพื่อแก้ปัญหา ALB ผลการปรัทัศน์วรรณกรรมทำให้ทราบถึงหลักการและแนวทางการแก้ปัญหา ALB แบบดั้งเดิม ทราบถึงแนวคิด ประเภท เกณฑ์วัตถุประสงค์ ขั้นตอนการแก้ปัญหา เทคนิควิธีและอัลกอริทึมต่างๆ ในการแก้ปัญหา ALB ทราบถึงหลักการและอัลกอริทึมของการค้นหาแบบตามู และทำให้ทราบว่า การประยุกต์เทคนิคการค้นหาแบบปัญญาประดิษฐ์มาใช้แก้ปัญหา ALB ซึ่งส่วนใหญ่จะใช้จินเนติกอัลกอริทึมหรือ GA สำหรับการประยุกต์ใช้ TS ในการแก้ปัญหา ALB ยังไม่ค่อนแพร่หลายมากนัก ส่วนการประยุกต์ใช้การค้นหาแบบมดและเครือข่ายประสาทเทียมเพื่อแก้ปัญหา ALB ยังได้รับความสนใจน้อย

บทที่ 3

ปัญหาการจัดสมดุลสายงานการประกอบ

3.1 กล่าวนำ

ในบทที่ 3 นี้จะกล่าวถึงทฤษฎีเบื้องต้นของการจัดสมดุลสายงานการประกอบ กล่าวถึงปัญหาการจัดสมดุลสายงานการประกอบหรือปัญหา ALB ที่ใช้ในวิทยานิพนธ์นี้ ซึ่งจะเป็นปัญหา ALB แบบผลิตภัณฑ์เดียวที่มีการกำหนดเวลาดำเนินงานของชิ้นงานให้มีค่าคงที่ (SMD) นอกจากนี้ยังกล่าวถึงวิธีการแก้ปัญหา ALB แบบศึกษาสำนึกที่นิยมใช้อย่างแพร่หลาย และจะถูกนำมาประยุกต์ใช้ในวิทยานิพนธ์นี้ด้วย นั่นคือวิธี COMSOAL ดังรายละเอียดต่อไปนี้

3.2 ทฤษฎีเบื้องต้นของการจัดสมดุลสายงานการประกอบ

สายงานการประกอบจะประกอบไปด้วยสถานีงานที่ถูกจัดเรียงลำดับด้วยสายพานลำเลียงหรืออุปกรณ์ที่มีลักษณะคล้ายคลึงกัน งานต่างๆ จะถูกลำเลียงผ่านสายงานการประกอบ และส่งผ่านจากสถานีงานหนึ่งไปยังอีกสถานีงานหนึ่ง เวลาดำเนินงานของสถานีงานต่างๆ จะอ้างอิงกับรอบเวลาการผลิต โดยทั่วไปการจัดสมดุลสายงานการประกอบจะเป็นการจัดสรรกลุ่มงานหรือชิ้นงานให้กับสถานีงานตามวัตถุประสงค์เฉพาะ

เพื่อความสะดวกและความเข้าใจที่ตรงกันในการศึกษาวิจัยเกี่ยวกับการจัดสมดุลสายงานการประกอบ จึงได้มีการกำหนดคำจำกัดความของคำศัพท์ต่างๆ ที่ใช้ดังต่อไปนี้

- **สายงานการประกอบ** คือสายงานที่ใช้ในการประกอบชิ้นส่วนผลิตภัณฑ์ตั้งแต่เริ่มต้นจนเสร็จสิ้นกระบวนการ ในที่นี้สายงานการประกอบจะหมายถึงสายงานการประกอบสำหรับผลิตภัณฑ์เดียวเท่านั้น

- **สมดุลสายงานการประกอบ** หรืออาจจะเรียกอีกอย่างหนึ่งว่าสมดุลสายการผลิต (production line balancing) คือการกำหนดชิ้นงานต่างๆ ในการประกอบผลิตภัณฑ์หรือสินค้าที่ทำให้ภาระงานในสถานีงานต่างๆ ใกล้เคียงกัน

- **ชิ้นงาน** คือชิ้นการทำงานหนึ่งที่ไม่สามารถแบ่งแยกให้กับคนงานตั้งแต่ 2 คนขึ้นไป หรือเครื่องจักรตั้งแต่ 2 เครื่องขึ้นไป โดยไม่มีความขัดแย้งระหว่างกันได้

- **เวลาดำเนินงานของชิ้นงาน** คือระยะเวลาที่ใช้ดำเนินการของชิ้นงานจนสำเร็จ

- **สถานีงาน** คือหน่วยหรือบริเวณที่อยู่ตามสายงานการประกอบ ซึ่งเป็นตำแหน่งที่เกิดกิจกรรมการประกอบชิ้นส่วนผลิตภัณฑ์ตามการปฏิบัติงานที่ได้ถูกกำหนดไว้

- เวลาดำเนินงานของสถานีงาน คือระยะเวลาที่ใช้ดำเนินงานทุกชั้นงานในสถานีงานนั้นๆ จนสำเร็จ

- การดำเนินงานในแต่ละสถานีงาน คือกลุ่มของชั้นงานต่างๆ ที่มารวมกัน และเป็นกลุ่มงานที่ทำในสถานีงานหนึ่งๆ

- รอบเวลาการผลิต คือเวลาที่ชิ้นส่วนต่างๆ ถูกประกอบจนแล้วเสร็จบนสายการประกอบ โดยทั่วไปรอบเวลาการผลิตจะขึ้นอยู่กับอัตราการผลิต ยกตัวอย่าง เช่น ถ้าอัตราการผลิต คือ 10 หน่วยต่อชั่วโมง นั่นคือรอบเวลาการผลิตจะเท่ากับ 6 นาทีต่อหน่วย

- เวลาว่างงาน คือผลต่างของเวลาระหว่างเวลาดำเนินงานของสถานีงานและรอบเวลาการผลิต

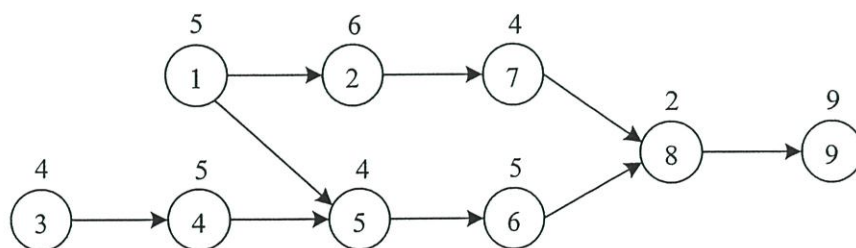
- ความสัมพันธ์ก่อนหลังของชั้นงาน คือความสัมพันธ์ของชั้นงานระหว่าง 2 ชั้นงาน โดยที่ชั้นงานแต่ละชั้นงานอาจมีความสัมพันธ์กับชั้นงานอื่นๆ มากกว่าหนึ่งชั้นงาน ลักษณะการกำหนดความสัมพันธ์ดังกล่าวจะมีด้วยกัน 2 แบบดังนี้คือ

- ชั้นงานหนึ่งจะต้องทำก่อนอีกชั้นงานหนึ่ง หมายความว่า ชั้นงานที่ถูกกำหนดให้ต้องทำก่อนจะถูกจัดให้กับสถานีงานที่อยู่ตามลำดับหลังสถานีงานของอีกชั้นงานหนึ่งไม่ได้

- ชั้นงานหนึ่งจะต้องทำหลังชั้นงานหนึ่ง หมายความว่า ชั้นงานที่ถูกกำหนดให้ต้องทำภายหลังจะถูกจัดให้กับสถานีงานที่อยู่ในลำดับก่อนหน้าสถานีงานของอีกชั้นงานหนึ่งไม่ได้

กำหนดให้สถานีงานของสายงานการประกอบเป็น $k = 1, \dots, m$ เซตของชั้นงานกำหนดให้เป็น $V = \{1, \dots, m\}$ เวลาดำเนินงานของชั้นงาน j กำหนดให้เป็น t_j ภาระงานรวมสำหรับการจัดสมดุลสายงานการประกอบจะคำนวณจากผลรวมของเวลาดำเนินงานของชั้นงานทุกชั้นงานในสายงาน นั่นคือ $\sum_{j=1}^n t_j$ ซึ่งลำดับของการดำเนินการประกอบชั้นงานสามารถอธิบายได้ด้วยแผนภูมิ

ลำดับงานก่อนหลังดังแสดงในรูปที่ 2.4 ซึ่งได้รับการนำมาแสดงอีกครั้งในรูปที่ 3.1 ซึ่งมีชั้นงานทั้งหมดจำนวน 9 ชั้นงาน และมีเวลาดำเนินงานของชั้นงานอยู่ในช่วง 2 – 9 หน่วยเวลา



รูปที่ 3.1 แผนภูมิลำดับงานก่อนหลัง

จากแผนภูมิลำดับงานก่อนหลังของสายงานการประกอบในรูปที่ 3.1 สามารถเขียนแสดงรายละเอียดของชิ้นงาน เวลาดำเนินงานของชิ้นงาน และลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน ดังแสดงในตารางที่ 3.1

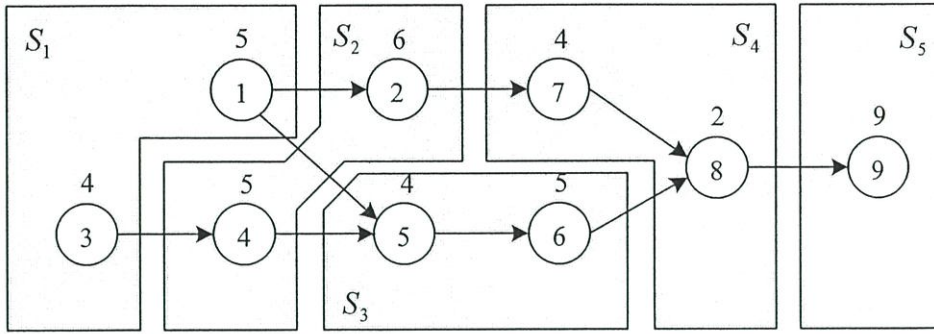
ตารางที่ 3.1 รายละเอียดของสายงานการประกอบในรูปที่ 3.1

ชิ้นงาน	เวลาดำเนินงานของชิ้นงาน (หน่วยเวลา)	ลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน (ชิ้นงานที่ต้องดำเนินการก่อน)
1	5	-
2	6	1
3	4	-
4	5	3
5	4	1,4
6	5	5
7	4	2
8	2	6,7
9	9	8

การจัดสมดุลสายงานการประกอบจะเป็นการจัดสรรชิ้นงานให้กับแต่ละสถานีงานโดยไม่ขัดกับลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน กำหนดให้ S_k คือเซตของชิ้นงานที่ถูกจัดสรรให้กับสถานีงาน k ดังนั้นเวลาดำเนินงานของสถานีงานคือ $t(S_k) = \sum_{j \in S_k} t_j$

เมื่อกำหนดให้รอบเวลาการผลิตเป็น c (สายงานการประกอบแบบ paced line) การจัดสมดุลสายงานการประกอบจะมีความเป็นไปได้ก็ต่อเมื่อเวลาดำเนินงานของสถานีงานทุกสถานีงานต้องมีค่าไม่เกิน c ในกรณีที่ $(S_k) < c$ สถานีงาน k จะปรากฏเวลาว่างงานซึ่งสามารถคำนวณได้จากผลต่างระหว่าง c และ $t(S_k)$ นั่นคือเวลาว่างงานจะมีค่าเท่ากับ $c - t(S_k)$

จากสายงานการประกอบในรูปที่ 3.1 และตารางที่ 3.1 เมื่อกำหนดให้ $c = 11$ หน่วยเวลา และกำหนดให้จำนวนสถานีงานทั้งหมดมีค่าเป็น $m = 5$ สถานีงาน คำตอบของการจัดสมดุลสายงานการประกอบที่เป็นไปได้คำตอบหนึ่งคือ $S_1 = \{1, 3\}, S_2 = \{2, 4\}, S_3 = \{5, 6\}, S_4 = \{7, 8\}$ และ $S_5 = \{9\}$ ดังแสดงในรูปที่ 3.2 และตารางที่ 3.2 ตามลำดับ วัตถุประสงค์ทั่วไปในการจัดสมดุลสายงานการประกอบคือต้องการให้ประสิทธิภาพของสายงานการประกอบ (assembly line efficiency, E) มีค่ามากที่สุด โดยที่ค่าประสิทธิภาพ $E = \sum_{j=1}^n t_j / (m \times c)$



รูปที่ 3.2 ผลการจัดสมดุลสายงานการประกอบในรูปที่ 3.1

ตารางที่ 3.2 การจัดสมดุลสายงานการประกอบในรูปที่ 3.1

สถานีงาน	ชิ้นงาน	เวลาดำเนินงานของสถานีงาน (หน่วยเวลา)	เวลาว่างงาน (หน่วยเวลา)
S_1	1,3	9	2
S_2	2,4	11	0
S_3	5,6	9	2
S_4	7,8	6	5
S_5	9	9	2

3.3 ลักษณะปัญหาการ จัดสมดุลสายงานการประกอบแบบผลิตภัณฑ์เดียว

ปัญหาการ จัดสมดุลสายงานการประกอบหรือปัญหา ALB ที่ใช้ในวิทยานิพนธ์นี้เป็น ปัญหา ALB แบบผลิตภัณฑ์เดียวที่มีการกำหนดเวลาดำเนินงานของชิ้นงานให้มีค่าคงที่ (SMD) ปัญหา ALB ดังกล่าวถือเป็นปัญหา ALB แบบดั้งเดิมที่มีรูปแบบปัญหาอย่างง่ายและมีผู้สนใจนำมา ศึกษาวิจัยมากที่สุด [21] ซึ่งมีการคำนวณสถานีงานดังสมการที่ (3.1) การคำนวณเวลาว่างงานรวม ดังสมการที่ (3.2) การคำนวณความแปรปรวนของภาระงานดังสมการที่ (3.3) และการคำนวณ ประสิทธิภาพของสายงานดังสมการที่ (3.4)

$$m = \frac{\sum_{j=1}^n t_j}{c} \quad (3.1)$$

$$T_{ID} = \sum_{k=1}^m t_{ID}^k = \sum_{k=1}^m (c - t(S_k)) \quad (3.2)$$

$$w_v = \frac{\sum_{k=1}^m \left[t(S_k) - \left(\frac{\sum_{j=1}^n t_j}{m} \right) \right]^2}{m} \quad (3.3)$$

$$E = 100 \frac{\sum_{k=1}^m t(S_k)}{m \times c_r} = 100 \frac{\sum_{j=1}^n t_j}{m \times c_r} \quad (3.4)$$

เมื่อ m	คือจำนวนสถานีงาน
t_j	คือเวลาดำเนินงานของชิ้นงาน j
c	คือรอบเวลาการผลิต
c_r	คือรอบเวลาการผลิตจริง
T_{ID}	คือเวลาว่างงานรวม
t_{ID}^k	คือเวลาว่างงานของสถานีงาน k
$t(S_k)$	คือเวลาดำเนินงานของสถานีงาน k
w_v	คือความแปรปรวนของภาระงาน
E	คือประสิทธิภาพของสายงานการประกอบ

ความแตกต่างระหว่าง c และ c_r คือ c เป็นรอบเวลาการผลิตที่ได้จากการคำนวณทางทฤษฎี แต่ c_r คือรอบเวลาการผลิตที่เกิดจากการปฏิบัติงานจริง

3.4 วิธี COMSOAL

วิธีการแก้ปัญหา ALB แบบศึกษาสำนึกที่นิยมใช้อย่างแพร่หลาย และจะถูกนำมาประยุกต์ใช้ในงานวิจัยวิทยานิพนธ์นี้ด้วยนั่นคือวิธี COMSOAL (a computer method of sequencing operations for assembly line) ที่นำเสนอโดย Arcus [8] วิธีการดังกล่าวมีขั้นตอนที่ไม่ยุ่งยากซับซ้อน และได้รับการนำมาใช้เปรียบเทียบกับวิธีการใหม่สำหรับการแก้ปัญหา ALB อยู่เสมอ การแก้ปัญหา ALB ด้วยวิธี COMSAOL มีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1. จำแนกชิ้นงานทุกชิ้นงานที่มีอยู่ในสายงานการประกอบ พร้อมทั้งระบุรายชื่อชิ้นงานอื่นทุกชิ้นงานที่ต้องดำเนินการต่อจากชิ้นงานนั้น โดยทันที

ขั้นตอนที่ 2. สร้าง LIST A ขึ้นเพื่อบรรจุงานทุกงานที่ยังไม่ได้รับการจัดสรรให้อยู่ในสถานงานใด และจำนวนงานที่ต้องดำเนินการทันทีก่อนหน้างานนั้น ถ้าใน LIST A ไม่มีงานใดอยู่เลย แสดงว่าการขั้นตอนการจัดดำเนินงานจนเสร็จเรียบร้อยแล้ว

ขั้นตอนที่ 3. สร้าง LIST B ขึ้นเพื่อบรรจุงานที่ไม่มีงานก่อนหน้าโดยเลือกมาจาก LIST A ดังนั้น LIST B จึงเปรียบเสมือนการรวบรวมงานที่พร้อมจะจัดสรรเข้าสู่สถานงานในทันที

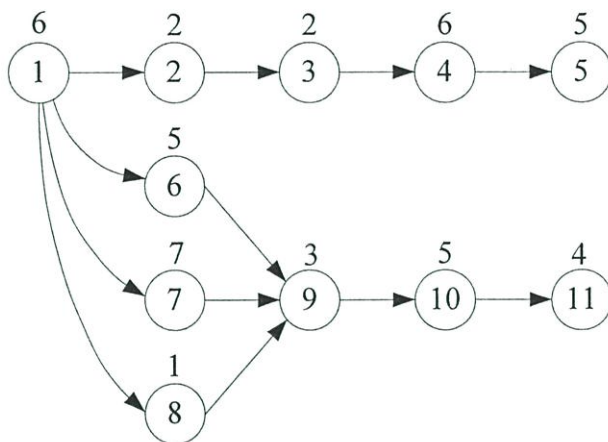
ขั้นตอนที่ 4. เลือกงานจาก LIST B มาเพียงงานเดียวโดยวิธีสุ่มแบบมีกฎเกณฑ์ ซึ่งจะได้กล่าวต่อไป งานที่เลือกมานี้จะถือว่าเป็นงานที่จัดเข้าในสถานงานอย่างถาวร และในการเลือกจะต้องตรวจสอบเวลาดำเนินงานของสถานงานที่เหลืออยู่กับเวลาดำเนินงานของงานที่กำลังพิจารณา ซึ่งเวลาดำเนินงานของงานที่เลือกมานั้นจะต้องไม่เกินเวลาดำเนินงานของสถานงานที่เหลืออยู่ ถ้าหากไม่มีงานที่ใช้เวลาดำเนินงานน้อยกว่าหรือเท่ากับเวลาดำเนินงานของสถานงานที่เหลืออยู่แล้ว ให้ทำการเพิ่มสถานงานขึ้นใหม่อีกสถานีหนึ่งในลำดับต่อจากสถานงานเดิม หลังจากนั้นให้ดำเนินงานซ้ำในขั้นตอนที่ 4 ใหม่ โดยเลือกงานลงในสถานงานใหม่นี้ งานที่ได้รับเลือกในขั้นตอนนี้จะบรรจุลงใน LIST C ซึ่งในแต่ละครั้งจะมีการเลือกเพียงงานเดียวเท่านั้น

ขั้นตอนที่ 5. ลบงานที่ปรากฏใน LIST C ออกจาก LIST A เนื่องจากงานนั้นได้ถูกกำหนดให้อยู่ในสถานงานอย่างถาวรแล้ว จากนั้นย้อนกลับไปดำเนินงานซ้ำในขั้นตอนที่ 2

การเลือกงานในขั้นตอนที่ 4 เป็นการเลือกงานแบบสุ่ม ดังนั้นประสิทธิภาพของสายงานการประกอบจึงมีความไม่แน่นอนเพราะไม่มีหลักเกณฑ์ในการเลือก ดังนั้นจึงนิยมนำหลักเกณฑ์ในการเลือกงานแบบต่างๆ มาใช้แทนดังนี้

- Longest operation time คือการเลือกงานที่ใช้เวลาดำเนินงานมากที่สุด เพื่อง่ายต่อการบรรจุงานขนาดเล็กที่เหลือเมื่อรอบเวลาการผลิตเหลือน้อย
- Shortest operation time คือการเลือกงานที่ใช้เวลาดำเนินงานน้อยที่สุด
- Least number of immediate follower คือการเลือกงานที่มีงานตามหลังทันทีน้อยที่สุด
- Most number of immediate predecessor คือการเลือกงานที่มีงานก่อนหน้ามากที่สุด
- Maximal-path technique คือการเลือกงานที่มีงานตามหลังต่อเป็นลูกโซ่หรือเป็นลำดับยาวที่สุด
- Most total time of remaining tasks คือการเลือกงานที่มีเวลาดำเนินงานรวมตั้งแต่ต้นจนงานที่พิจารณาจนถึงงานลำดับสุดท้ายที่มากที่สุด

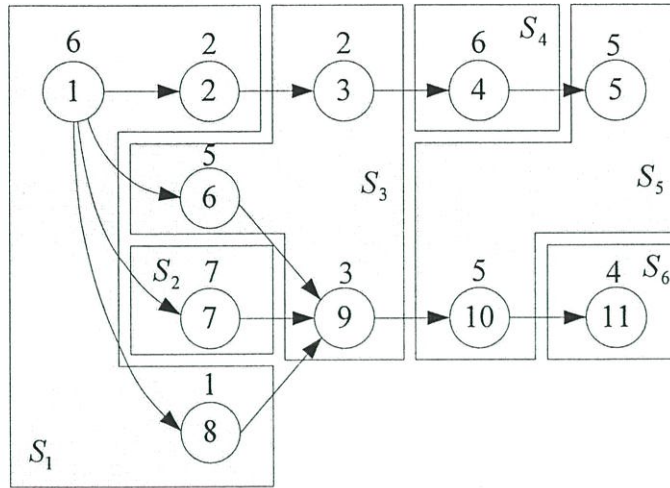
เพื่อให้เกิดความเข้าใจในการประยุกต์ใช้วิธี COMSOAL ในการแก้ปัญหา ALB มากยิ่งขึ้น จึงได้ยกตัวอย่างสายงานการประกอบแบบผลิตภัณฑ์เดียวที่มีชิ้นงานจำนวน 11 ชิ้นงาน ที่มีรายละเอียดดังแสดงด้วยแผนภูมิลำดับงานก่อนหลังดังรูปที่ 3.3



รูปที่ 3.3 แผนภูมิลำดับงานก่อนหลังของสายงานการประกอบที่มีชิ้นงานจำนวน 11 ชิ้นงาน

ตารางที่ 3.3 รายละเอียดของสายงานการประกอบในรูปที่ 3.3

ชิ้นงาน	เวลาดำเนินงานของชิ้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน (ชิ้นงานที่ต้องดำเนินการก่อน)
1	6	-
2	2	1
3	2	2
4	6	3
5	5	4
6	5	1
7	7	1
8	1	1
9	3	6,7,8
10	5	9
11	4	10



รูปที่ 3.4 การจัดสมดุลสายงานการประกอบในรูปที่ 3.3 ด้วยวิธี COMSOAL

ตารางที่ 3.4 ผลการจัดสมดุลสายงานการประกอบในรูปที่ 3.3 ด้วยวิธี COMSOAL

สถานีงาน	ชิ้นงาน	เวลาดำเนินงานของสถานีงาน (นาที)	เวลาว่างงาน (นาที)
S_1	1,2,8	9	1
S_2	7	7	3
S_3	6,3,9	10	0
S_4	4	6	4
S_5	5,10	10	0
S_6	11	4	6
เวลาว่างงานรวม = 14 นาที			
ค่าความแปรปรวนของภาระงาน = 4.89			
ประสิทธิภาพของสายงานการประกอบ = 76.67%			

จากแผนภูมิลำดับงานก่อนหลังของสายงานการประกอบในรูปที่ 3.3 สามารถเขียนแสดงรายละเอียดของชิ้นงาน เวลาดำเนินงานของชิ้นงาน และลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน ดังแสดงในตารางที่ 3.3 กำหนดให้รอบเวลาการผลิต $c = 10$ นาที การแก้ปัญหาการจัดสมดุลสายงานการประกอบนี้ด้วยวิธี COMSOAL ทำให้เกิดจำนวนสถานีงาน $m = 6$ สถานีงาน การคำนวณเวลาว่างงานรวม ความแปรปรวนของภาระงาน และประสิทธิภาพของสายงานการประกอบอาศัยสมการที่ (3.2) ถึง (3.4) ตามลำดับ ซึ่งคำตอบที่ได้จากการแก้ปัญหา ALB นี้ด้วยวิธี COMSOAL คือ

เวลาว่างงานรวม $T_D = 14$ นาที ค่าความแปรปรวนของภาระงาน $w_p = 4.89$ และประสิทธิภาพของสายงานการประกอบ $E = 76.67\%$ ดังแสดงในรูปที่ 3.4 และตารางที่ 3.4 ตามลำดับ

3.5 สรุป

บทที่ 3 นี้ได้กล่าวถึงทฤษฎีเบื้องต้นของการจัดสมดุลสายงานการประกอบ กล่าวถึงคำจำกัดความของคำศัพท์ต่างๆ ที่ใช้เกี่ยวกับการจัดสมดุลสายงานการประกอบ กล่าวถึงปัญหาการจัดสมดุลสายงานการประกอบหรือปัญหา ALB ที่ใช้ในวิทยานิพนธ์นี้ ซึ่งจะเป็นปัญหา ALB แบบ SMD การคำนวณสถานีงาน เวลาว่างงานรวม ความแปรปรวนของภาระงาน และประสิทธิภาพของสายงานที่ได้จากการจัดสมดุลสายงานการประกอบเรียบร้อยแล้ว ในบทที่ 3 ยังได้กล่าวถึงรายละเอียดขั้นตอนของวิธีการแก้ปัญห ALB แบบศึกษาสำนึกที่นิยมใช้อย่างแพร่หลายนั่นคือวิธี COMSOAL ซึ่งจะถูกนำมาประยุกต์ใช้ในวิทยานิพนธ์นี้

บทที่ 4

การค้นหาแบบตามูเชิงพันทาง

4.1 กล่าวนำ

การแก้ปัญหาการจัดสมมูลสายงานการประกอบในวิทยานิพนธ์นี้ จะประยุกต์ใช้การค้นหาแบบตามูเชิงพันทาง ซึ่งได้รับการพัฒนาขึ้นมาจากการค้นหาแบบตามูดั้งเดิม ดังนั้นเนื้อหาที่ปรากฏในบทที่ 4 นี้จะกล่าวถึง

- การค้นหาแบบตามู
- อัลกอริทึมของการค้นหาแบบตามู
- การค้นหาแบบตามูเชิงพันทางที่ได้รับการพัฒนาขึ้นเพื่อแก้ปัญหา ALB
- อัลกอริทึมของการค้นหาแบบตามูเชิงพันทาง
- การทดสอบโปรแกรมการค้นหาแบบตามู
- การทดสอบโปรแกรมการค้นหาแบบตามูเชิงพันทางที่พัฒนาขึ้น

4.2 การค้นหาแบบตามู

การค้นหาแบบตามู (tabu search: TS) [35-37] เป็นวิธีการค้นหาคำตอบที่เหมาะสมที่สุดสำหรับปัญหาแบบไม่ต่อเนื่องเชิงผสมผสาน การค้นหาแบบตามูอาศัยกระบวนการค้นหาใกล้เคียง (neighborhood search) และรายชื่อตามู (tabu list: TL) ซึ่งจะทำหน้าที่เก็บค่าคำตอบในอดีต คำตอบใน TL สามารถใช้ประโยชน์ในกรณีการค้นหาเกิดการล๊อคโดยคำตอบวงแคบเฉพาะถิ่น (local solution) โดยอาศัยเกณฑ์ปรารถนา (aspiration criteria: AC) กระบวนการค้นหาจะเคลื่อนที่แบบดีเทอร์มินิสติก ซึ่งจะถูกกำหนดไว้อย่างมีระเบียบแบบแผน โดยทั่วไปจะถูกกำกับไว้ด้วยเงื่อนไขความคงอยู่ล่าสุด (recency) และเงื่อนไขความซ้ำ (frequency) [38]

เงื่อนไขความคงอยู่ล่าสุด คือเงื่อนไขที่ใช้พิจารณาทิศทางการค้นหาคำตอบในช่วงเวลาที่ผ่านมา เมื่อกลไกการค้นหาพบคำตอบด้วยทิศทางการค้นหาอื่นๆ เงื่อนไขความคงอยู่ล่าสุดจะกำหนดให้ทิศทางดังกล่าวมีสถานะต้องห้าม (tabu) ในช่วงระยะเวลาหนึ่ง ต่อเมื่อเวลาผ่านไปมากเพียงพอจนมั่นใจได้ว่าการค้นหาจะไม่วนกลับไปพบคำตอบเดิมอีกแล้ว เงื่อนไขความคงอยู่ล่าสุดจะยกเลิกสถานะต้องห้ามของทิศทางการค้นหาดังกล่าว ด้วยเงื่อนไขความคงอยู่ล่าสุดนี้จะเป็นการประกันว่ากลไกการค้นหาจะไม่เข้าไปทำการค้นหาในทิศทางเดิมหรือพื้นที่เดิมที่ได้ดำเนินการค้นหาผ่านมาแล้ว เงื่อนไขนี้จะช่วยให้การค้นหาสามารถพบคำตอบที่ดีอื่นๆ แม้ว่าในจำนวนคำตอบเหล่านั้นอาจเป็นคำตอบที่ไม่ดีกว่าคำตอบเดิมที่เคยพบแล้วก็ตาม

สำหรับเงื่อนไขความซ้ำ คือเงื่อนไขที่ใช้พิจารณาจำนวนการเคลื่อนที่เข้าหาคำตอบของกลไกการค้นหาคำตอบด้วยทิศทางเดิม กล่าวคือเมื่อทิศทางการเคลื่อนที่เข้าหาคำตอบใดๆ ถูกเรียกใช้ด้วยจำนวนครั้งที่มากเกินไปจนเกินกำหนด เงื่อนไขความซ้ำจะกำหนดให้ทิศทางการเคลื่อนที่ดังกล่าวนั้นมีสถานะต้องห้ามในช่วงระยะเวลาหนึ่ง เพื่อเป็นการประกันว่ากลไกการค้นหาคำตอบจะไม่ดำเนินการค้นหาในลักษณะวนรอบอยู่กับที่ อันจะส่งผลทำให้เกิดการล้นของค่าคำตอบ

เงื่อนไขความคงอยู่ต่ำสุด และเงื่อนไขความซ้ำ จะถูกประยุกต์ใช้งานร่วมกันเสมอในลักษณะการดำเนินงานด้วยหน้าที่ตรงกันข้ามอย่างสัมพันธ์กัน กล่าวคือเมื่อเงื่อนไขหนึ่งทำหน้าที่กำหนดสถานะต้องห้ามให้กับกลไกการค้นหา อีกเงื่อนไขหนึ่งจะทำหน้าที่ยกเลิกสถานะต้องห้ามนั้นอย่างสอดคล้องและสัมพันธ์กัน อีกทั้งยังต้องสัมพันธ์กับ TL ด้วย

การค้นหาแบบตาบอดยังมีกลไกที่คอยทำหน้าที่เพิ่มสมรรถนะการค้นหาคำตอบ อีกสองกลไกด้วยกัน คือ กลไกความเข้มข้น (intensification) และกลไกความหลากหลาย (diversification) สำหรับกลไกความเข้มข้น จะเป็นการค้นหาคำตอบในกลุ่มคำตอบที่พบแล้วในอดีตที่ได้รับการบันทึกไว้ใน TL ซึ่งกลุ่มคำตอบดังกล่าวได้รับการพิจารณาแล้วว่าเป็นกลุ่มคำตอบที่ดี กลไกความเข้มข้นจะกำหนดให้กลไกการค้นหาคำตอบดำเนินการค้นหาคำตอบซ้ำในพื้นที่เดิมที่เคยพบกลุ่มคำตอบที่ดีอย่างละเอียดมากยิ่งขึ้น ซึ่งอาจส่งผลทำให้การค้นหาสามารถพบคำตอบที่ดีกว่าเดิมได้ สำหรับกลไกความหลากหลาย จะเป็นการค้นหาคำตอบในปริภูมิหรือพื้นที่การค้นหาที่ยังไม่เคยได้รับการสำรวจหรือค้นหามาก่อน กลไกความหลากหลายจะกำหนดให้กลไกการค้นหาคำตอบดำเนินการค้นหาคำตอบในพื้นที่อื่นหรือทิศทางการสำรวจอื่นที่แตกต่างไปจากเดิม ซึ่งอาจส่งผลทำให้การค้นหาสามารถพบคำตอบที่ดีกว่าเดิมได้เช่นเดียวกัน

ด้วยเพราะขั้นตอนที่ค่อนข้างง่าย อีกทั้งได้มีการพิสูจน์การลู่เข้าหาคำตอบวงกว้าง หรือเรียกว่า การพิสูจน์คุณสมบัติการลู่เข้าหาผลเฉลยที่เหมาะสมโดยอาศัยทฤษฎีบทและการพิสูจน์ทางคณิตศาสตร์ ผสมกับบล็อกที่ที่เหมาะสม ซึ่งพบว่าการค้นหาแบบตาบอดมีคุณสมบัติการลู่เข้าหาคำตอบวงกว้าง [39-40] ดังนั้นการค้นหาแบบตาบอดจึงได้รับความนิยมอย่างแพร่หลาย และถูกนำไปประยุกต์ใช้แก้ปัญหาต่างๆ มากมาย อาทิเช่น การแก้ปัญหาเกี่ยวกับการเงิน [41] การวางแผนการผลิตและการจ่ายไฟฟ้ากำลัง [42] การระบุเอกลักษณ์ระบบพลวัต [43] การแก้ปัญหา flow shop [44] และการแก้ปัญหา ALB [11],[55] เป็นต้น นอกจากนี้ยังมีบทความวิจัยที่ยืนยันว่าการค้นหาแบบตาบอดสามารถค้นหาคำตอบที่เหมาะสมได้เร็วกว่า GA [13]

อัลกอริทึมของการค้นหาแบบตาบอดแบบดั้งเดิม จะเป็นพื้นฐานในการพัฒนาไปสู่การค้นหาแบบตาบอดเชิงพันทาง ซึ่งมีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 1. กำหนดขอบเขตปริภูมิการค้นหา (search space), รัศมีการค้นหา (R), จำนวนค่าใกล้เคียง (N), จำนวนรอบการค้นหาเริ่มต้น ($count=0$), $TL=\emptyset$, และเกณฑ์ยุติการค้นหา (termination criteria: TC) (ซึ่งโดยทั่วไปกำหนดให้เป็นจำนวนรอบการค้นหาสูงสุด $count_max$)

ขั้นตอนที่ 2. สุ่มคำตอบเริ่มต้น x_0 ภายในปริภูมิการค้นหาดังแสดงในรูปที่ 4.1

ขั้นตอนที่ 3. ประเมินคำตอบเริ่มต้น x_0 ด้วยฟังก์ชันวัตถุประสงค์ (objective function) เพื่อหาค่าวัตถุประสงค์ (cost) โดยกำหนดให้เป็น $f(x_0)$

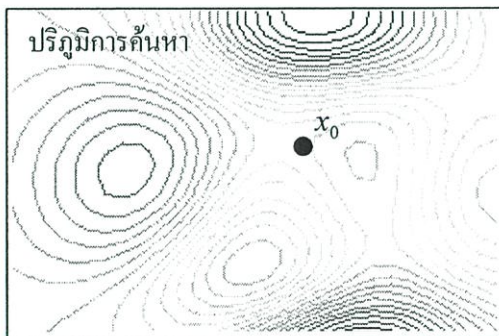
ขั้นตอนที่ 3. ผลิตค่าใกล้เคียง แบบสุ่มรอบคำตอบเริ่มต้น x_0 ภายในปริภูมิการค้นหารัศมี R จำนวน N ตัว แล้วเก็บไว้ในเซต X ดังแสดงในรูปที่ 4.2

ขั้นตอนที่ 4. ประเมินค่าใกล้เคียงด้วยฟังก์ชันวัตถุประสงค์ โดยกำหนดให้ x_1 คือค่าใกล้เคียงที่ทำให้ค่าฟังก์ชันวัตถุประสงค์ $f(x_1)$ มีค่าน้อยที่สุดในบรรดาค่าใกล้เคียงในเซต X

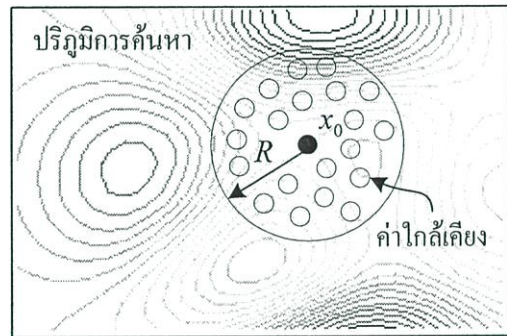
ขั้นตอนที่ 5. ถ้า $f(x_1) < f(x_0)$ แล้วทำการเก็บ x_0 ไว้ใน TL จากนั้นปรับค่า $x_0 = x_1$ ดังแสดงในรูปที่ 4.3 ไม่เช่นนั้นแล้วทำการเก็บ x_1 ไว้ใน TL แทน และเกิดทิศทางการค้นหาขึ้นอันจะก่อให้เกิดรอบการค้นหาดัดไปดังรูปที่ 4.4 ถึง 4.6

ขั้นตอนที่ 6. เรียกใช้กลไกย้อนรอยการค้นหา เมื่อเกิดการลือกของคำตอบวงแคบเฉพาะถิ่นขึ้น ดังแสดงในรูปที่ 4.7

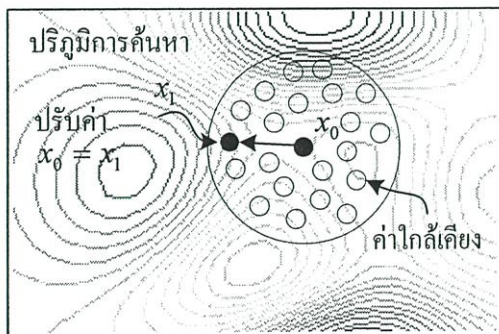
ขั้นตอนที่ 7. ตรวจสอบ TC ถ้า TC สอดคล้อง ให้ยุติการกระบวนการค้นหา ค่า x_0 คือคำตอบที่ดีที่สุด ไม่เช่นนั้นให้เพิ่มค่า *count* แล้ววนกลับไปยังขั้นตอนที่ 3 เพื่อดำเนินกระบวนการค้นหาในรอบต่อไป



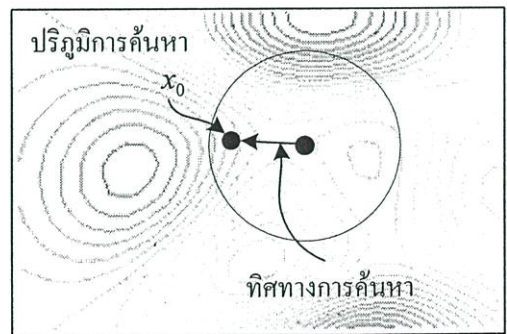
รูปที่ 4.1 การสุ่มคำตอบเริ่มต้น x_0



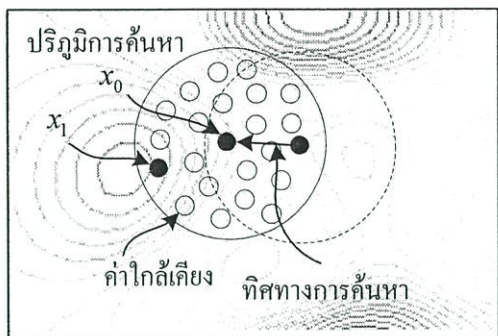
รูปที่ 4.2 การผลิตค่าใกล้เคียงแบบสุ่ม



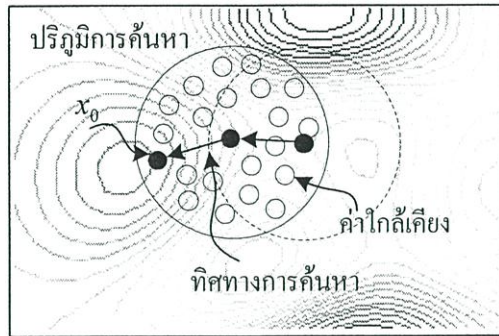
รูปที่ 4.3 การปรับค่า $x_0 = x_1$



รูปที่ 4.4 ทิศทางการค้นหา

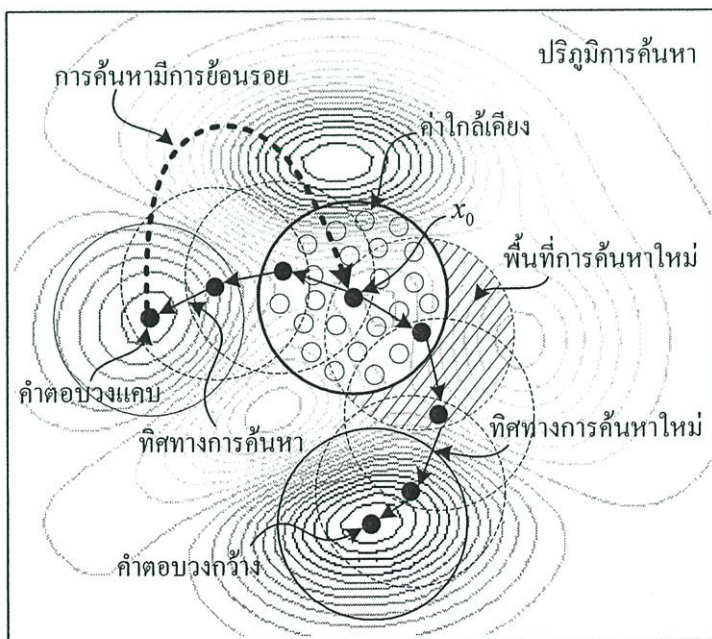


รูปที่ 4.5 การใช้ x_0 ใหม่เป็นคำตอบเริ่มต้น

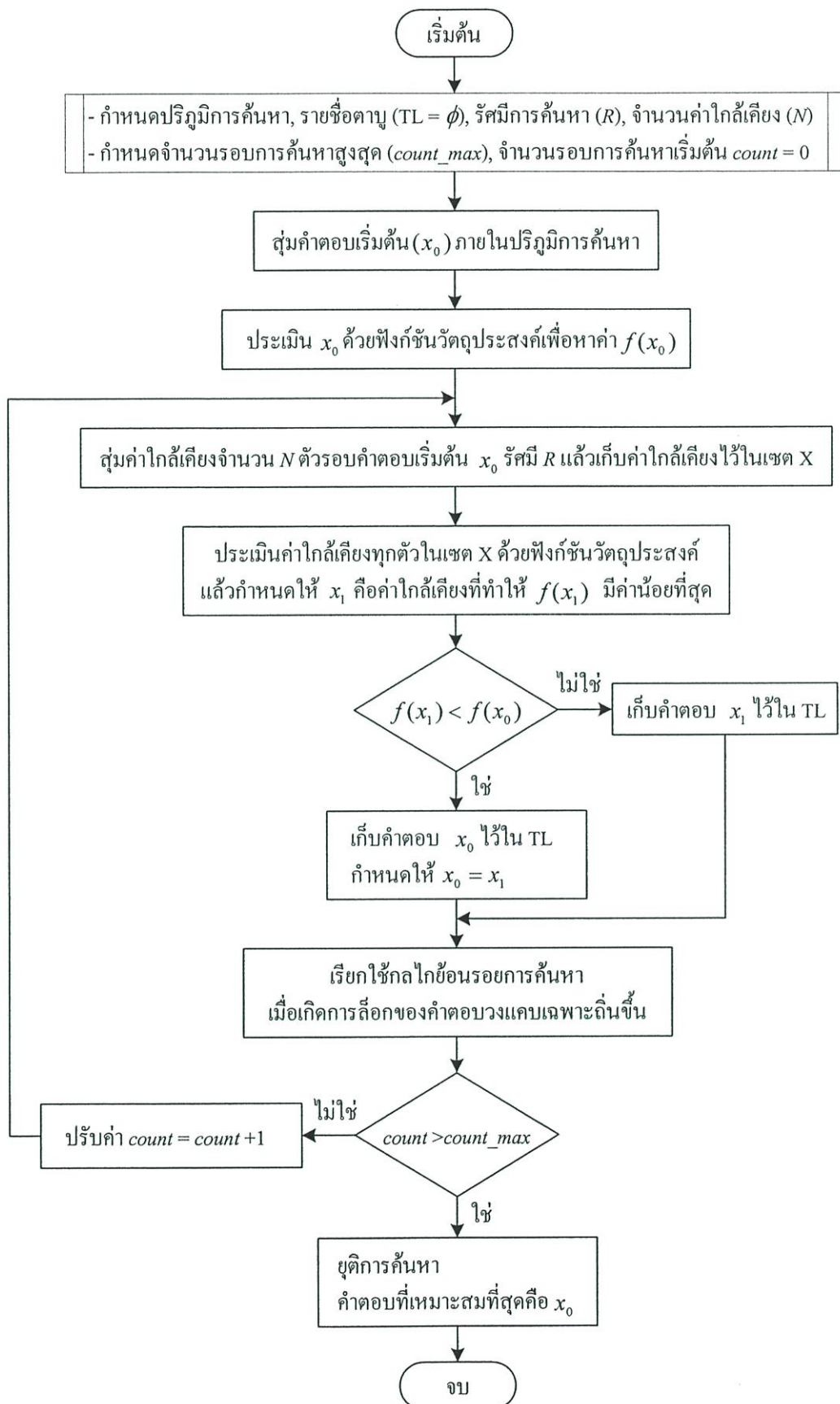


รูปที่ 4.6 ทิศทางการค้นหาในรูปอัตรา

กลไกย้อนรอยการค้นหา ถือเป็นเกณฑ์ปรารถนา (aspiration criteria: AC) รูปแบบหนึ่ง กลไกจะมีกระบวนการเรียกใช้คำตอบเก่าที่บันทึกไว้ใน TL เมื่อพบว่าคำตอบที่ได้จากการค้นหาเกิดการล๊อคโดยคำตอบวงแคบเฉพาะถิ่น ดังแสดงในรูปที่ 4.7 เมื่อกระบวนการค้นหาเกิดการล๊อคโดยคำตอบวงแคบเฉพาะถิ่นขึ้น กลไกการย้อนรอยค้นหาจะอนุญาตให้ใช้คำตอบที่ได้มีการค้นหาไว้แล้วซึ่งบรรจุอยู่ใน TL เพื่อใช้เป็นคำตอบเริ่มต้นสำหรับการค้นหาในรอบต่อไป ซึ่งจะทำให้เกิดพื้นที่การค้นหาใหม่ เป็นการเปิดโอกาสให้กระบวนการค้นหาเปลี่ยนทิศทาง และสามารถหลุดออกจากการล๊อคโดยคำตอบวงแคบเฉพาะถิ่นได้ จากนั้นกระบวนการค้นหาจะดำเนินต่อไปจนพบกับคำตอบวงแคบเฉพาะถิ่นอื่นๆ และในที่สุดก็จะพบกับคำตอบวงกว้าง และจากอัลกอริทึมที่กล่าวมาสามารถเขียนแสดงด้วยแผนภูมิการทำงานดังรูปที่ 4.8



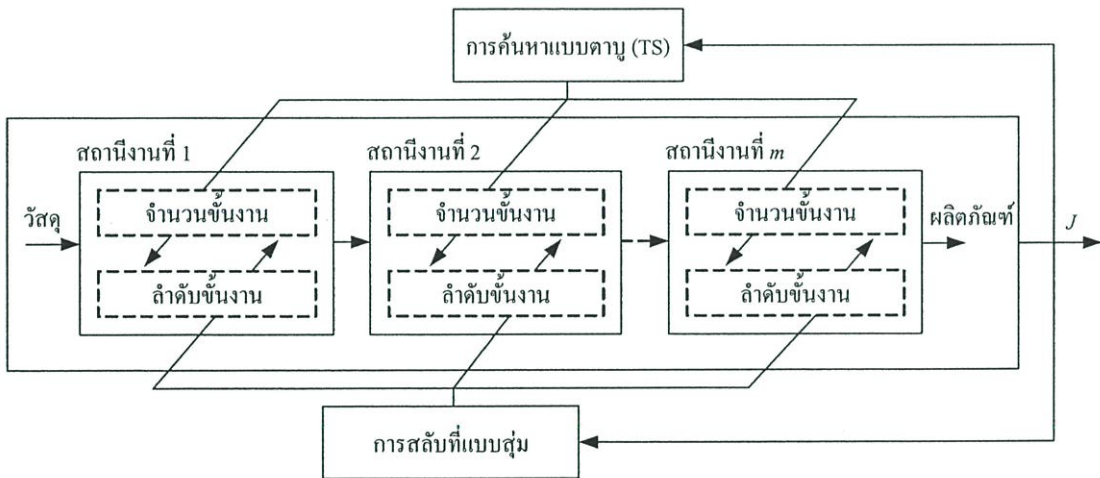
รูปที่ 4.7 กลไกย้อนรอยการค้นหา



รูปที่ 4.8 แผนภูมิการทำงานของการค้นหาแบบคาบ

4.3 การค้นหาแบบตามูเชิงพันทาง

การค้นหาแบบตามูเชิงพันทาง (hybrid tabu search: HTS) ได้รับการพัฒนาขึ้นจากการค้นหาแบบตามู (TS) เพื่อแก้ปัญหา ALB แบบผลิตภัณฑ์เดียว การค้นหาแบบตามูเชิงพันทางหรือ HTS จะเป็นกระบวนการทำงานร่วมกันระหว่าง TS และการสลับที่แบบสุ่ม (random permutation) กล่าวคือ TS จะทำหน้าที่จัดสรรจำนวนชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน ในขณะที่การสลับที่แบบสุ่มจะทำหน้าที่จัดลำดับของชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน ดังแสดงในรูปที่ 4.9 ในกรณีการแก้ปัญหา ALB ที่มีการกำหนดจำนวนสถานีงานที่เหมาะสมมาแล้วแต่ต้น ซึ่งอาจคำนวณได้จากความสัมพันธ์ดังสมการที่ (3.1) ในวิทยานิพนธ์นี้กำหนดให้ค่าความแปรปรวนของภาระงาน (w_v) เป็นค่าฟังก์ชันวัตถุประสงค์ (J) ของกระบวนการค้นหา จากรูปที่ 4.9 ค่า J จะถูกป้อนกลับไปยัง TS และการสลับที่แบบสุ่มเพื่อให้มีค่าน้อยที่สุดและสอดคล้องกับลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน (precedence constraints) ดังสมการที่ (4.1) และ (4.2) ตามลำดับ จำนวนรอบการค้นหาสูงสุดจะถูกกำหนดให้เป็นเกณฑ์ยุติการค้นหาของกระบวนการ (TC) อัลกอริทึมของ HTS ที่นำเสนอเพื่อการแก้ปัญหา ALB มีรายละเอียดดังต่อไปนี้



รูปที่ 4.9 แผนผังการแก้ปัญหา ALB ด้วย HTS

$$J = w_v = \frac{\sum_{k=1}^m \left[t(S_k) - \left(\frac{\sum_{j=1}^n t_j}{m} \right) \right]^2}{m} \quad (4.1)$$

$$\begin{aligned} \min \quad & J \\ \text{subject to} \quad & \text{precedence constraints} \end{aligned} \quad (4.2)$$

ขั้นตอนที่ 1. กำหนดปฏิภูมิการค้นหา (ขอบเขตของจำนวนชั้นงานในแต่ละสถานีงาน), รัศมีการค้นหา (R), จำนวนค่าใกล้เคียง (N), รายชื่อต่า (TL-I = ϕ) เพื่อเก็บจำนวนชั้นงานในแต่ละสถานีงาน, รายชื่อต่า (TL-II = ϕ) เพื่อเก็บลำดับของชั้นงาน, จำนวนรอบการค้นหาสูงสุด ($count_max$) จำนวนรอบเริ่มต้น ($count = 0$), และจำนวนสถานีงาน (m)

ขั้นตอนที่ 2. โหลดชั้นงาน (n) เวลาดำเนินงานของชั้นงาน และลำดับความสัมพันธ์ก่อนหลังของชั้นงาน

ขั้นตอนที่ 3. ผลิตชุดลำดับชั้นงานจำนวน q ครั้ง โดยการนำชั้นงานทั้งหมดมาทำการสลับที่แบบสุ่ม แล้วเก็บไว้ใน $\psi_{i,n}, i=1,2,\dots,q$ กำหนดให้ $\psi_{i,n} \in \Psi$

ขั้นตอนที่ 4. สุ่มค่าตอบเริ่มต้น ($x_0^i, i=1,2,\dots,m$) ภายในปฏิภูมิการค้นหา

ขั้นตอนที่ 5. ประเมินค่าตอบเริ่มต้น x_0^i ด้วยฟังก์ชันวัตถุประสงค์ (ค่าความแปรปรวนของภาระงาน, w_v) เพื่อหาค่าวัตถุประสงค์ $f(x_0^i)$ โดยนำชุดลำดับชั้นงาน $\psi_{i,n}$ ที่เตรียมไว้มาจัดสรรเข้าสู่สถานีงานต่างๆ

ขั้นตอนที่ 6. กำหนดให้ x_0^i ที่ทำให้ $f(x_0^i)$ มีค่าน้อยที่สุดเป็น x_1^i และค่า $f(x_0^i)$ ที่มีค่าน้อยที่สุดคือ $f(x_1^i)$ กำหนดให้ลำดับชั้นงานที่ทำให้เกิด x_1^i และ $f(x_1^i)$ เป็น $\psi_{1,n}$

ขั้นตอนที่ 7. สุ่มค่าใกล้เคียงจำนวน N ตัวรอบค่าตอบ x_1^i รัศมี R แล้วเก็บค่าใกล้เคียงไว้ในเซต X

ขั้นตอนที่ 8. ประเมินค่าใกล้เคียงทุกตัวในเซต X ด้วยฟังก์ชันวัตถุประสงค์ (ความแปรปรวนของภาระงาน, w_v) เพื่อหาค่า $f()$ โดยนำชุดลำดับชั้นงาน $\psi_{i,n}$ ที่เตรียมไว้มาจัดสรรเข้าสู่สถานีงานต่างๆ

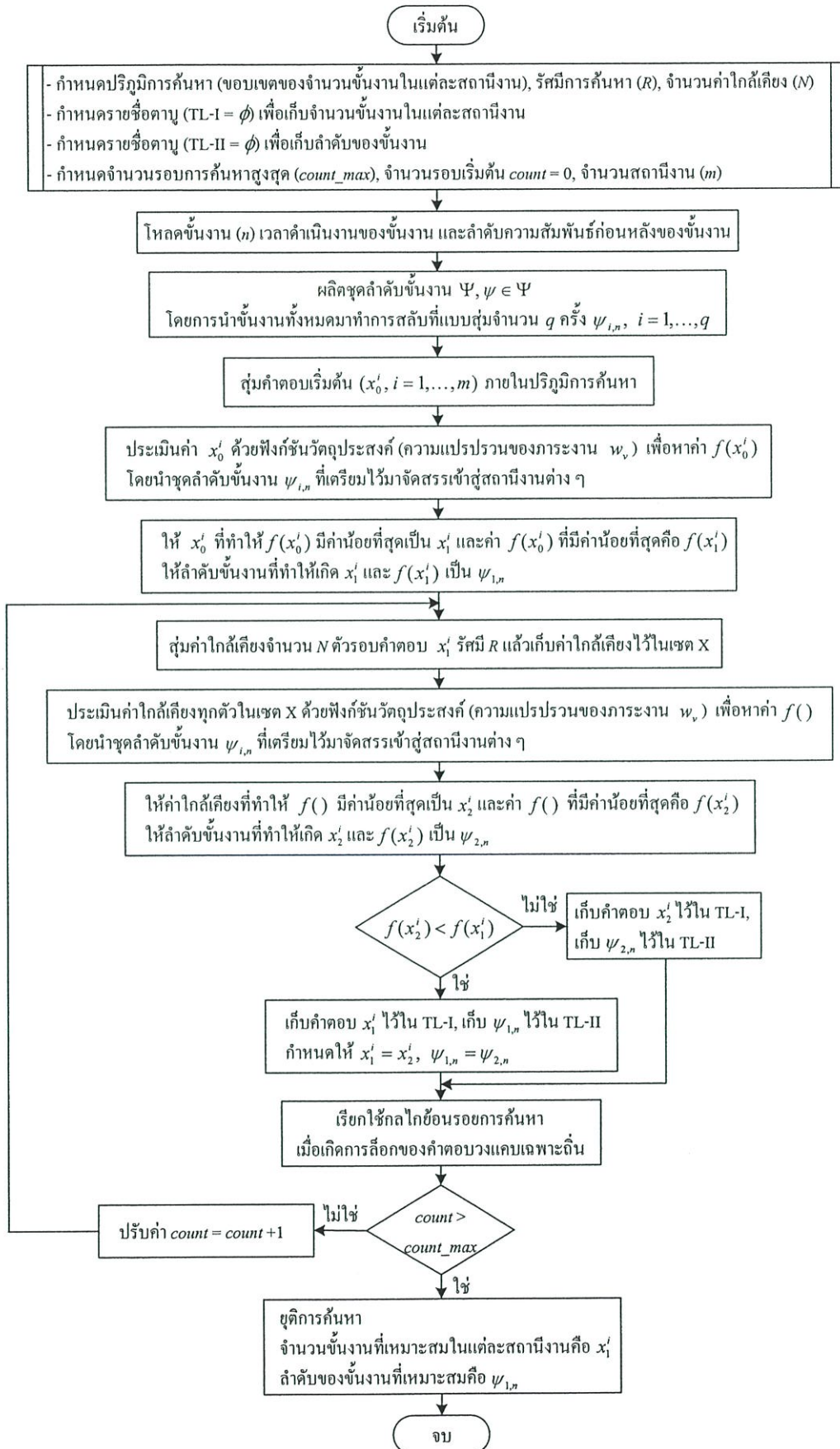
ขั้นตอนที่ 9. กำหนดให้ค่าใกล้เคียงที่ทำให้ $f()$ มีค่าน้อยที่สุดเป็น x_2^i และค่า $f()$ ที่มีค่าน้อยที่สุดคือ $f(x_2^i)$ กำหนดให้ลำดับชั้นงานที่ทำให้เกิด x_2^i และ $f(x_2^i)$ เป็น $\psi_{2,n}$

ขั้นตอนที่ 10. ถ้าการเปรียบเทียบ $f(x_2^i) < f(x_1^i)$ เป็นจริงแล้วให้ทำการเก็บค่าตอบ x_1^i ไว้ใน TL-I และเก็บ $\psi_{1,n}$ ไว้ใน TL-II จากนั้นกำหนดให้ $x_1^i = x_2^i$ และ $\psi_{1,n} = \psi_{2,n}$ แต่ถ้าไม่เป็นจริง ให้ทำการเก็บค่าตอบ x_2^i ไว้ใน TL-I และเก็บ $\psi_{2,n}$ ไว้ใน TL-II แทน

ขั้นตอนที่ 11. เรียกใช้กลไกย้อนรอยการค้นหาเมื่อเกิดการล็อกของค่าตอบวงแคบเฉพาะถิ่น

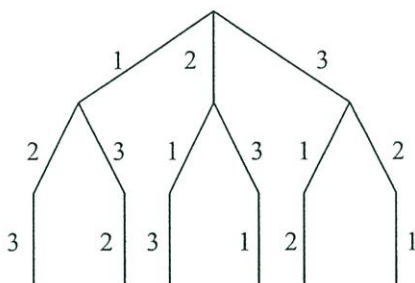
ขั้นตอนที่ 12. ตรวจสอบ TC ถ้า TC สอดคล้องให้ยุติกระบวนการค้นหา ค่า x_1^i และ $\psi_{1,n}$ คือคำตอบที่ดีที่สุด (x_1^i คือจำนวนชั้นงานที่เหมาะสมในแต่ละสถานีงาน และ $\psi_{1,n}$ คือลำดับชั้นงานที่เหมาะสมและสอดคล้องกับลำดับความสัมพันธ์ก่อนหลังของชั้นงาน) ถ้าไม่เช่นนั้นให้เพิ่มค่า $count$ แล้ววนกลับไปยังขั้นตอนที่ 7 เพื่อดำเนินกระบวนการค้นหาในรอบต่อไป

จากอัลกอริทึมที่กล่าวมาสามารถเขียนแสดงด้วยแผนภูมิการทำงานดังรูปที่ 4.10

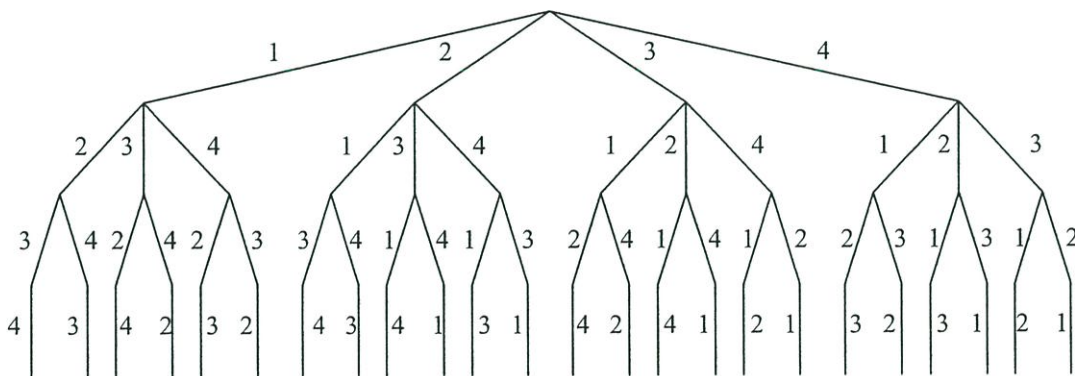


รูปที่ 4.10 แผนภูมิการทำงานของ HTS เพื่อการแก้ปัญหา ALB

จากขั้นตอนที่ 3 ของอัลกอริทึมที่นำเสนอ ได้มีการใช้การสลับที่แบบสุ่มเพื่อการผลิตชุดลำดับของชั้นงานขึ้น การสลับที่แบบสุ่มเป็นการสลับที่ของเวกเตอร์จำนวนจริงใดๆ เมื่อกำหนดให้เวกเตอร์จำนวนจริงมีสมาชิกเท่ากับ n การสลับที่แบบสุ่มของเวกเตอร์จำนวนจริงดังกล่าวจะมีจำนวนเท่ากับ $n!$ พิจารณาเวกเตอร์จำนวนจริง $A = \{1, 2, 3\}$ นั่นคือเวกเตอร์ A มีสมาชิก $n = 3$ การสลับที่แบบสุ่มของเวกเตอร์ A จะมีจำนวนเท่ากับ $n!$ นั่นคือ 6 ชุด (a_1 ถึง a_6) ดังนี้ $a_1 = \{1, 2, 3\}$, $a_2 = \{1, 3, 2\}$, $a_3 = \{2, 1, 3\}$, $a_4 = \{2, 3, 1\}$, $a_5 = \{3, 1, 2\}$, และ $a_6 = \{3, 2, 1\}$, เราสามารถเขียนแสดงแผนภูมิการสลับที่แบบสุ่มของเวกเตอร์ A ได้ดังรูปที่ 4.11(ก) ในทำนองเดียวกันเมื่อพิจารณาเวกเตอร์จำนวนจริง $B = \{1, 2, 3, 4\}$ นั่นคือเวกเตอร์ B มีสมาชิก $n = 4$ การสลับที่แบบสุ่มของเวกเตอร์ B จะมีจำนวนเท่ากับ $n!$ นั่นคือ 24 ชุด (b_1 ถึง b_{24}) ซึ่งสามารถเขียนแสดงแผนภูมิการสลับที่แบบสุ่มของเวกเตอร์ B ได้ดังรูปที่ 4.11(ข) จากรูปที่ 4.11(ข) เราสามารถเขียนได้เป็น $b_1 = \{1, 2, 3, 4\}$, $b_2 = \{1, 2, 4, 3\}$, ..., $b_{24} = \{4, 3, 2, 1\}$ นั่นเอง สำหรับอัลกอริทึมการสุ่มแบบสลับที่จะไม่ขอกล่าวถึงในที่นี้ ผู้อ่านที่สนใจสามารถศึกษาได้จาก [58-60] เป็นต้น



(ก) เวกเตอร์ $A = \{1, 2, 3\}$



(ข) เวกเตอร์ $B = \{1, 2, 3, 4\}$

รูปที่ 4.11 แผนภูมิการสลับที่แบบสุ่ม

ในกรณีการแก้ปัญหา ALB ที่มีจำนวนชั้นงานหลายๆ การสลับที่แบบสุ่มในชั้นตอนที่ 3 อาจมีความล่าช้าและอาจไม่มีประสิทธิภาพ ในกรณีดังกล่าวอัลกอริทึมในชั้นตอนที่ 3 จำเป็นจะต้องได้รับการปรับปรุง กล่าวคือจะต้องมีการจัดกลุ่มของชั้นงานเพื่อการสลับที่แบบสุ่ม โดยอาศัยหลักการจัดกลุ่มแบบศึกษาสำนึก โดยมีขั้นตอนดังนี้

1. กำหนดให้จำนวนชั้นงานทั้งหมดเท่ากับ n
2. กำหนดให้กลุ่มชั้นงานคือ $G_i, i = 1, 2, \dots, g$
3. กลุ่มชั้นงาน G_i จะบรรจุชั้นงานที่ไม่ชั้นงานก่อนหน้า
4. กลุ่มชั้นงาน $G_i, i > 1$ จะบรรจุชั้นงานที่มีชั้นงานก่อนหน้าในกลุ่มชั้นงาน G_{i-1}
5. จัดเรียงกลุ่มชั้นงาน $G_1 | G_2 | G_3 | \dots | G_g |$
6. ทำการสลับที่แบบสุ่มให้กับชั้นงานในแต่ละกลุ่มชั้นงาน ผลรวมของชั้นงานจากทุกกลุ่มชั้นงานจะเท่ากับ n

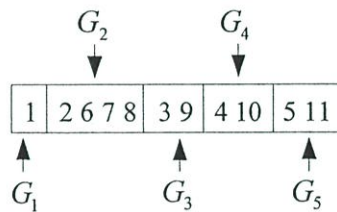
เพื่อให้เกิดความเข้าใจในการจัดกลุ่มของชั้นงานเพื่อการสลับที่แบบสุ่มเพื่อการแก้ปัญหา ALB มากยิ่งขึ้น จึงได้ยกตัวอย่างการจัดกลุ่มของชั้นงานเพื่อการสลับที่แบบสุ่มของสายงานการประกอบแบบผลิตภัณฑ์เดียวที่มีชั้นงานจำนวน 11 ชั้นงาน ที่มีรายละเอียดดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 รายละเอียดของสายงานการประกอบอย่างง่าย

ชั้นงาน	เวลาดำเนินงานของชั้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชั้นงาน (ชั้นงานที่ต้องดำเนินการก่อน)
1	6	-
2	2	1
3	2	2
4	6	3
5	5	4
6	5	1
7	7	1
8	1	1
9	3	6,7,8
10	5	9
11	4	10

จากขั้นตอนการจัดกลุ่มของชั้นงานเพื่อการสลับที่แบบสุ่ม เราสามารถจัดกลุ่มชั้นงานให้กับสายงานการประกอบในตารางที่ 4.1 ได้ดังนี้

1. จำนวนชั้นงาน $n = 11$
2. กลุ่มชั้นงาน G_1 บรรจุชั้นงานที่ 1
3. กลุ่มชั้นงาน G_2 บรรจุชั้นงานที่ 2, 6, 7, 8
4. กลุ่มชั้นงาน G_3 บรรจุชั้นงานที่ 3, 9
5. กลุ่มชั้นงาน G_4 บรรจุชั้นงานที่ 4, 10
6. กลุ่มชั้นงาน G_5 บรรจุชั้นงานที่ 5, 11
7. จัดเรียงกลุ่มชั้นงาน $G_1 | G_2 | G_3 | G_4 | G_5$ ดังแสดงในรูปที่ 4.12
8. ทำการสลับที่แบบสุ่มให้กับชั้นงานในแต่ละกลุ่มชั้นงาน ซึ่งจะได้จำนวนชุดลำดับของชั้นงานทั้งหมดเท่ากับ $4!+2!+2!+2! = 30$ ชุด ดังแสดงในรูปที่ 4.13



รูปที่ 4.12 การจัดกลุ่มชั้นงานเพื่อการสลับที่แบบสุ่มของปัญหา ALB ในตารางที่ 4.1

ชุดที่ 1 →	1	2 6 7 8	3 9	4 10	5 11
ชุดที่ 2 →	1	2 7 6 8	3 9	4 10	5 11
	1	2 6 8 7	3 9	4 10	5 11
⋮	⋮	⋮	⋮	⋮	⋮
	1	2 6 7 8	9 3	4 10	5 11
	⋮	⋮	⋮	⋮	⋮
	1	2 6 7 8	3 9	10 4	5 11
	⋮	⋮	⋮	⋮	⋮
	1	2 6 7 8	3 9	4 10	11 5
	⋮	⋮	⋮	⋮	⋮
ชุดที่ 30 →	1	8 7 6 2	9 3	10 4	11 5

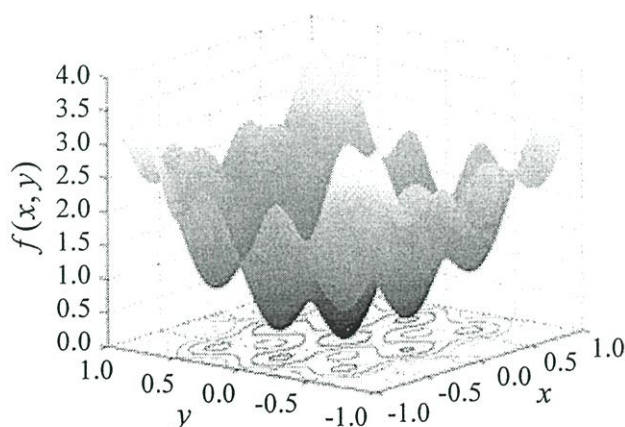
รูปที่ 4.13 ผลการจัดกลุ่มชั้นงานเพื่อการสลับที่แบบสุ่มของปัญหา ALB ในตารางที่ 4.1

4.4 การทดสอบโปรแกรมการค้นหาแบบตามูเชิงพันทาง

อัลกอริทึมของการค้นหาแบบตามู หรือ TS ที่ได้รับการนำเสนอในหัวข้อที่ 4.2 และอัลกอริทึมของการค้นหาแบบตามูเชิงพันทาง หรือ HTS ที่ได้รับการนำเสนอในหัวข้อที่ 4.3 จะถูกพัฒนาขึ้นเป็นโปรแกรมการค้นหา เพื่อการแก้ปัญหา ALB แบบผลิตภัณฑ์เดียวโดยอาศัย โปรแกรม MATLAB ดังรายละเอียดที่แสดงในภาคผนวก ข. และ ค. ตามลำดับ ในหัวข้อนี้จะเป็นการทดสอบโปรแกรมการค้นหาแบบตามู (โปรแกรม TS) และโปรแกรมการค้นหาแบบตามูเชิงพันทาง (โปรแกรม HTS) ดังรายละเอียดต่อไปนี้

4.4.1 การทดสอบโปรแกรมการค้นหาแบบตามู

การทดสอบโปรแกรมการค้นหาแบบตามู หรือโปรแกรม TS เพื่อตรวจสอบสมรรถนะในการค้นหาผลเฉลย หรือคำตอบวงกว้างของปัญหาที่มีคำตอบวงแคบเฉพาะถิ่นจำนวนมาก แต่มีคำตอบวงกว้างเพียงคำตอบเดียว และเนื่องจาก TS ตั้งอยู่บนพื้นฐานของกระบวนการสุ่ม ดังนั้นการทดสอบจะต้องกระทำด้วยจำนวนการค้นหาที่มากเพียงพอและด้วยคำตอบเริ่มต้นที่ไม่เท่ากัน เพื่อเป็นการยืนยันสมรรถนะในการค้นหาคำตอบวงกว้างของปัญหาที่สนใจ ในวิทยานิพนธ์นี้จะใช้โปรแกรม TS ที่พัฒนาขึ้นด้วยโปรแกรม MATLAB ที่ดำเนินการบนเครื่องคอมพิวเตอร์ Pentium 4 ความเร็ว 1.6 จิกกะเฮิร์ตซ์ สำหรับปัญหาที่ใช้ในการทดสอบจะเป็นการหาค่าต่ำสุดของฟังก์ชันโบฮาเชฟสกี (Bohachevsky's function) [61] หรือฟังก์ชัน BF ดังแสดงในสมการที่ (4.3) ซึ่งเป็นฟังก์ชันที่มีตัวแปรต้นสองตัวแปรคือ x และ y ค่าฟังก์ชัน $f(x, y)$ แสดงดังพื้นผิวในรูปที่ 4.14 ฟังก์ชัน BF มีคำตอบวงแคบเฉพาะถิ่นจำนวนมาก แต่จะมีคำตอบวงกว้าง หรือคำตอบ x และ y ที่ทำให้ $f(x, y)$ มีค่าน้อยที่สุดมีอยู่เพียงคำตอบเดียว คือ $x = y = 0$ ซึ่งจะทำให้ $f(x, y) = 0$



รูปที่ 4.14 พื้นผิวฟังก์ชัน BF

$$f(x, y) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7 \quad (4.3)$$

เมื่อพิจารณาตามบริบทของการหาค่าเหมาะที่สุด เราสามารถเขียนได้ดังสมการที่ (4.4)

$$\begin{aligned} \min \quad & f(x, y) \\ \text{subject to} \quad & -1 \leq x \leq 1 \\ & -1 \leq y \leq 1 \end{aligned} \quad (4.4)$$

การค้นหาแบบตาบอด หรือ TS มีตัวประกอบการค้นหา (search factor) หรือค่าพารามิเตอร์ของการค้นหา (search parameter) ที่ส่งผลกระทบต่อสมรรถนะการค้นหาอย่างชัดเจนดังนี้

1. R หรืออัตราี่การค้นหา
2. N หรือจำนวนสมาชิกค่าใกล้เคียง

การทดสอบโปรแกรม TS จะเป็นการทดสอบผลกระทบของตัวประกอบการค้นหา R และ N โดยที่การทดสอบจะกำหนดให้ตัวประกอบการค้นหาดังกล่าวมีค่าดังนี้

- $R = 25, 50, 75,$ และ 100% ของปริภูมิการค้นหา
- $N = 15, 30, 45,$ และ 60 ตัว

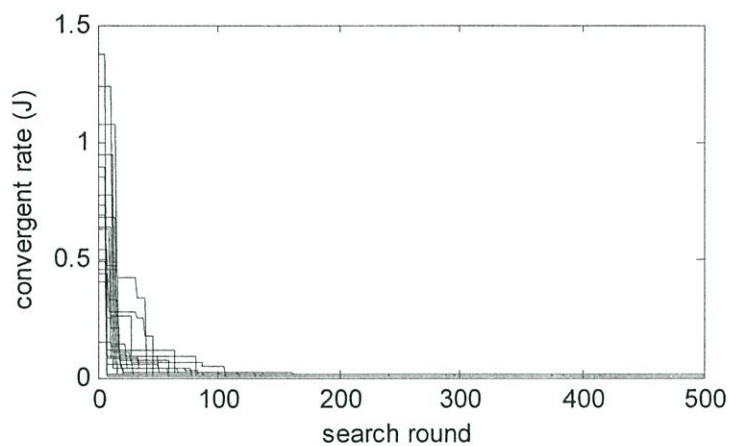
ตัวประกอบการค้นหาข้างต้นจะได้รับการปรับทีละตัว ขณะที่ตัวอื่นจะถูกคงค่าเอาไว้ การปรับค่าตัวประกอบการค้นหาแต่ละครั้ง จะทำการทดสอบจำนวน 20 ครั้ง ค่าคำตอบเริ่มต้นจะถูกกำหนดแบบสุ่มโดยโปรแกรม MATLAB ทั้งนี้เพื่อเป็นการประกันว่าค่าคำตอบเริ่มต้นในแต่ละครั้งของการค้นหาจะมีค่าไม่เท่ากัน และกระบวนการค้นหาจะยุติลงเมื่อ TC สอดคล้อง ซึ่ง TC ที่ใช้คือ จำนวนรอบของการค้นหามีค่าเท่ากับจำนวนรอบของการค้นหาสูงสุด (*count_max*) กำหนดให้มีค่าเท่ากับ 500 รอบ ผลการทดสอบโปรแกรม TS ที่พัฒนาขึ้นด้วยโปรแกรม MATLAB มีรายละเอียดดังแสดงในตารางที่ 4.2 ซึ่งพบว่ากระบวนการค้นหาของ TS สามารถหาค่าคำตอบ (ใกล้เคียง) วงกว้างได้อย่างมีประสิทธิภาพ ค่าตอบที่ดีที่สุดคือค่าฟังก์ชันวัตถุประสงค์ (*min-cost*) เท่ากับ 3.5333×10^{-6} ทั้งนี้เมื่อปรับให้ $R = 100\%$ และ $N = 60$ ตัว อย่างไรก็ตาม เวลาที่ใช้ในการค้นหาจะเพิ่มขึ้นตามจำนวนค่าใกล้เคียงที่กำหนดเพิ่มขึ้น รูปที่ 4.15 ถึง 4.30 แสดงการลู่เข้าของค่าฟังก์ชันวัตถุประสงค์เมื่อทำการปรับตัวประกอบการค้นหาทีละตัว เมื่อพิจารณาจากรูปที่ 4.15 ถึง 4.30 จะพบว่า TS จะมีอัตราการลู่เข้าหาค่าตอบของฟังก์ชัน BF อย่างมีประสิทธิภาพเมื่อปรับ $R = 50$ ถึง 100% และ $N = 45$ ถึง 60 ตัว

อย่างไรก็ตาม การปรับค่าตัวประกอบการค้นหาที่เหมาะสมจะขึ้นอยู่กับปัญหาที่สนใจ (problem dependent) นั่นหมายความว่าเมื่อนำ TS ไปประยุกต์ใช้กับปัญหาอื่นจำเป็นที่จะต้องทำการทดสอบ TS เพื่อหาค่าตัวประกอบการค้นหาที่เหมาะสมใหม่ โดยอาจใช้ค่าที่สรุปไว้เป็นแนวทางตั้งต้นได้เท่านั้น

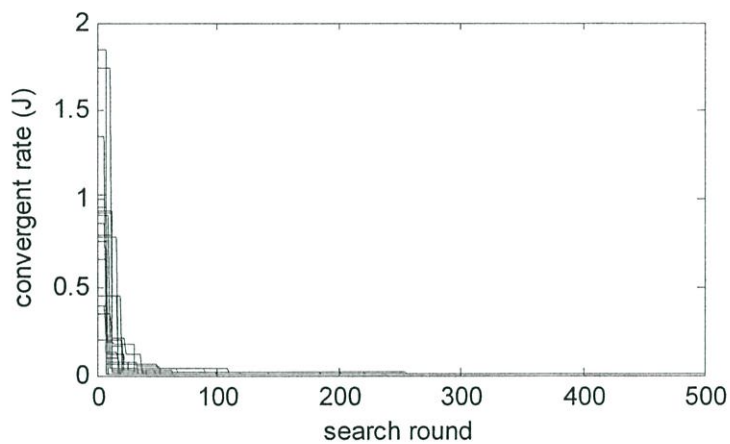
ตารางที่ 4.2 ผลการทดสอบโปรแกรม TS กับ BF

จำนวนค่าใกล้เคียง (<i>N</i>)	รัศมีการค้นหา (<i>R</i>) = 25% ของปริภูมิการค้นหา				
	min-cost	max-cost	mean-cost	std-cost	search time (sec)
15	2.3094e-004	0.0095	0.0032	0.0027	0.2820
30	8.8207e-005	0.0096	0.0016	0.0023	0.5310
45	5.6929e-005	0.0044	0.0012	0.0010	0.7660
60	1.0604e-005	0.0033	0.0010	9.7788e-004	1.0160
จำนวนค่าใกล้เคียง (<i>N</i>)	รัศมีการค้นหา (<i>R</i>) = 50% ของปริภูมิการค้นหา				
	min-cost	max-cost	mean-cost	std-cost	search time (sec)
15	5.1248e-004	0.0153	0.0036	0.0035	0.3910
30	1.7730e-005	0.0045	0.0020	0.0013	0.7350
45	8.8154e-006	0.0052	9.6737e-004	0.0012	1.0940
60	7.0588e-005	0.0043	0.0011	0.0011	1.4690
จำนวนค่าใกล้เคียง (<i>N</i>)	รัศมีการค้นหา (<i>R</i>) = 75% ของปริภูมิการค้นหา				
	min-cost	max-cost	mean-cost	std-cost	search time (sec)
15	4.6781e-006	0.0128	0.0038	0.0035	0.5000
30	4.9147e-005	0.0044	0.0016	0.0014	0.9530
45	3.1561e-004	0.0032	0.0012	7.3176e-004	1.4220
60	4.8586e-005	0.0034	9.7727e-004	9.3506e-004	1.8910
จำนวนค่าใกล้เคียง (<i>N</i>)	รัศมีการค้นหา (<i>R</i>) = 100% ของปริภูมิการค้นหา				
	min-cost	max-cost	mean-cost	std-cost	search time (sec)
15	5.0150e-004	0.0119	0.0029	0.0025	0.6090
30	3.3790e-005	0.0049	0.0017	0.0015	1.1720
45	4.8950e-005	0.0041	0.0011	0.0011	1.7970
60	3.5333e-006	0.0037	9.1290e-004	0.0010	2.3280

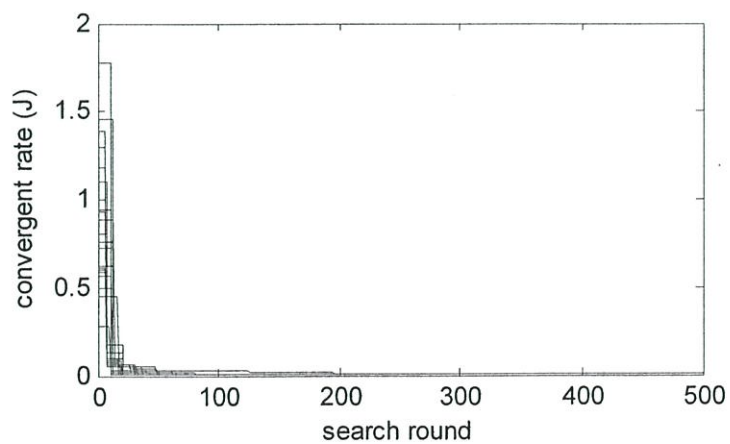
เมื่อ min-cost คือค่าฟังก์ชันวัตถุประสงค์ที่มีค่าน้อยที่สุด, max-cost คือค่าฟังก์ชันวัตถุประสงค์ที่มีค่ามากที่สุด, mean-cost คือค่าฟังก์ชันวัตถุประสงค์เฉลี่ย, std-cost คือค่าเบี่ยงเบนมาตรฐานของค่าฟังก์ชันวัตถุประสงค์, และ search time คือเวลาที่ใช้ในการค้นหา



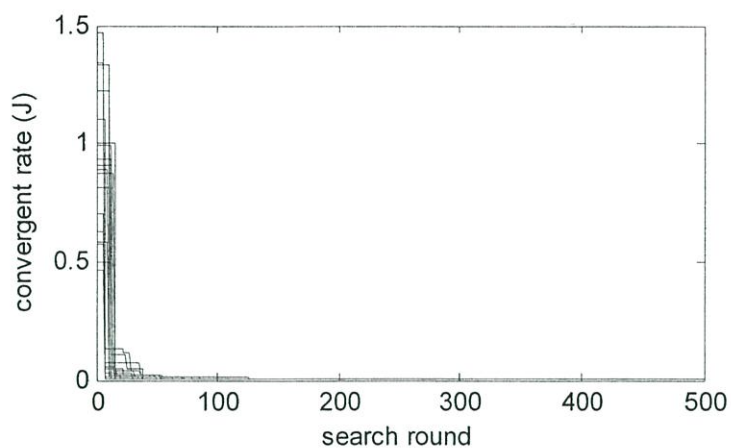
รูปที่ 4.15 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 25\%$ และ $N = 15$



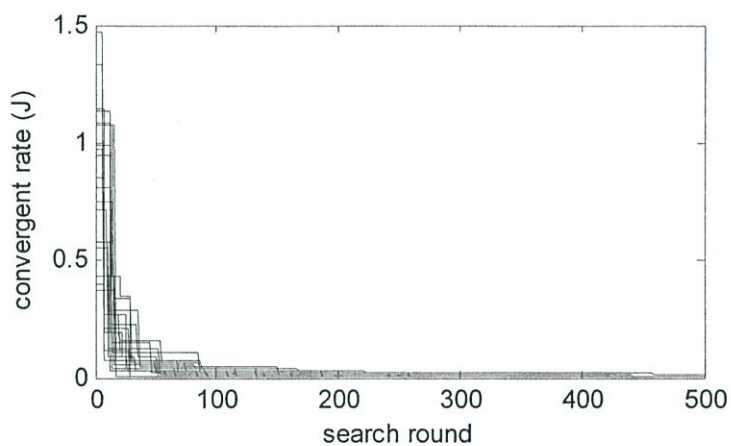
รูปที่ 4.16 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 25\%$ และ $N = 30$



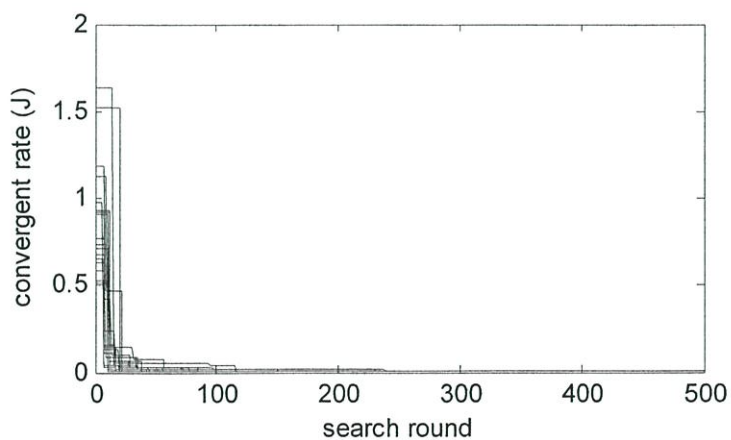
รูปที่ 4.17 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 25\%$ และ $N = 45$



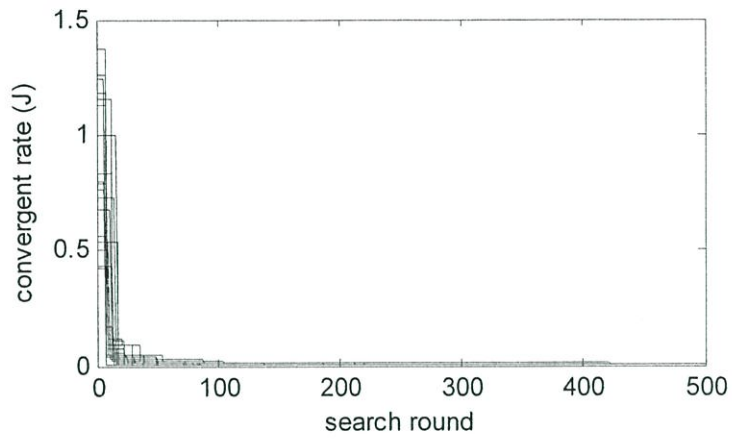
รูปที่ 4.18 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 25\%$ และ $N = 60$



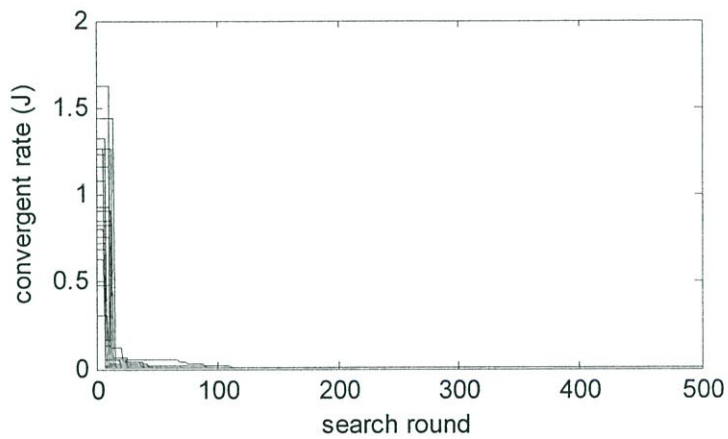
รูปที่ 4.19 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 50\%$ และ $N = 15$



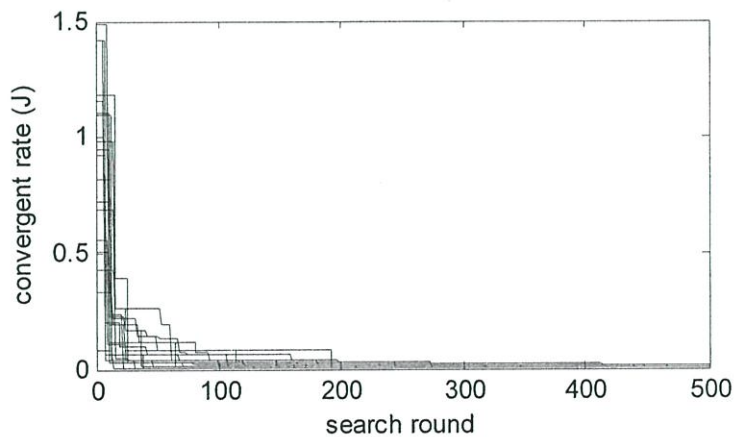
รูปที่ 4.20 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 50\%$ และ $N = 30$



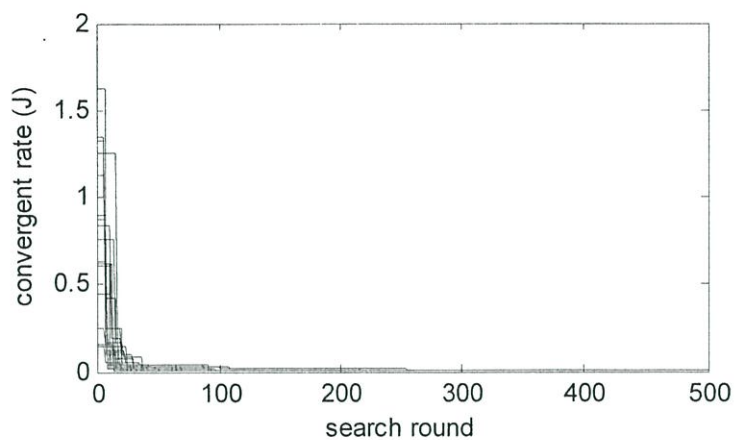
รูปที่ 4.21 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 50\%$ และ $N = 45$



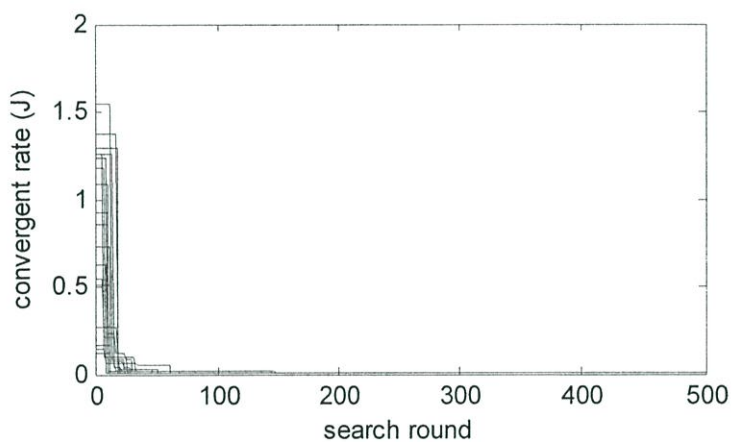
รูปที่ 4.22 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 50\%$ และ $N = 60$



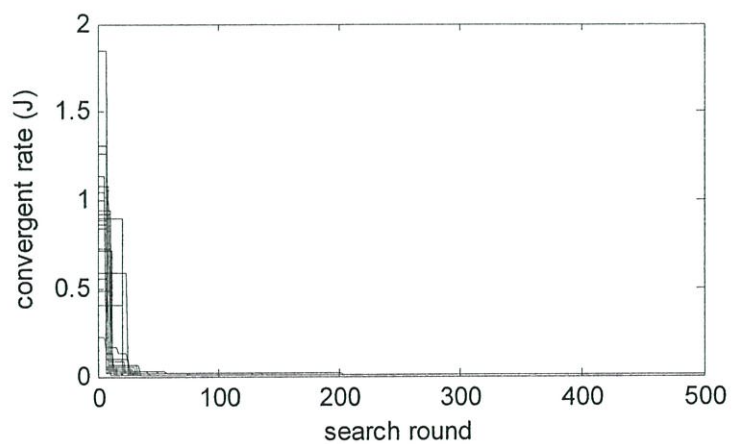
รูปที่ 4.23 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 75\%$ และ $N = 15$



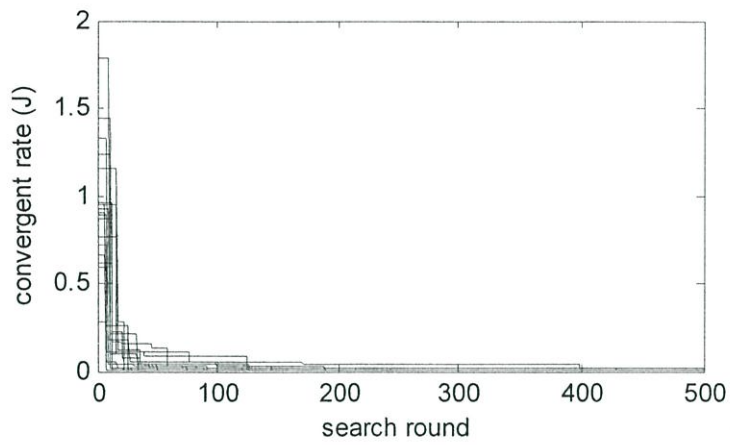
รูปที่ 4.24 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 75\%$ และ $N = 30$



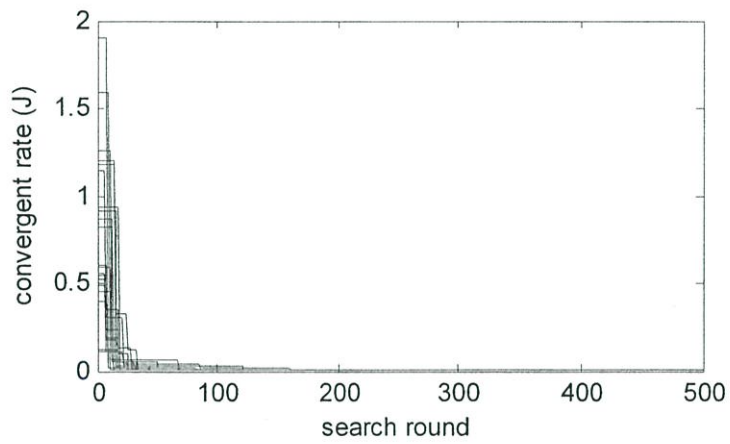
รูปที่ 4.25 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 75\%$ และ $N = 45$



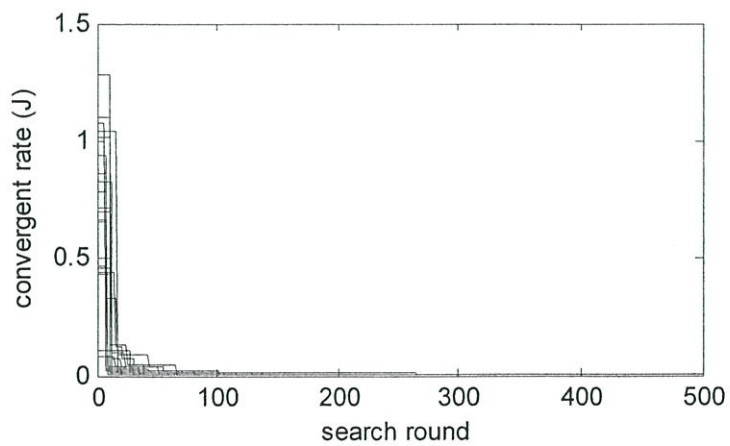
รูปที่ 4.26 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 75\%$ และ $N = 60$



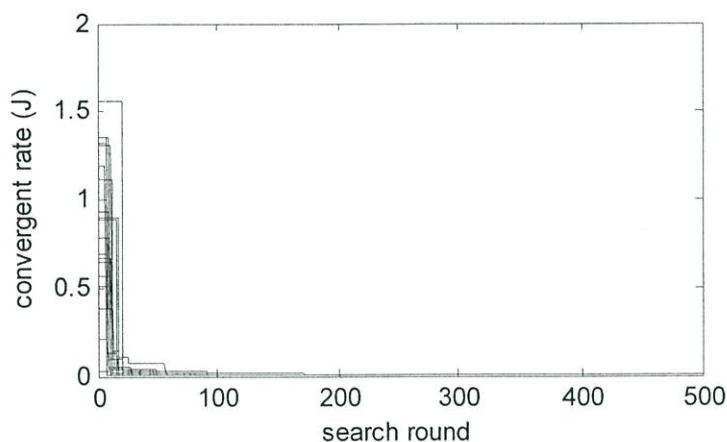
รูปที่ 4.27 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 100\%$ และ $N = 15$



รูปที่ 4.28 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 100\%$ และ $N = 30$



รูปที่ 4.29 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 100\%$ และ $N = 45$



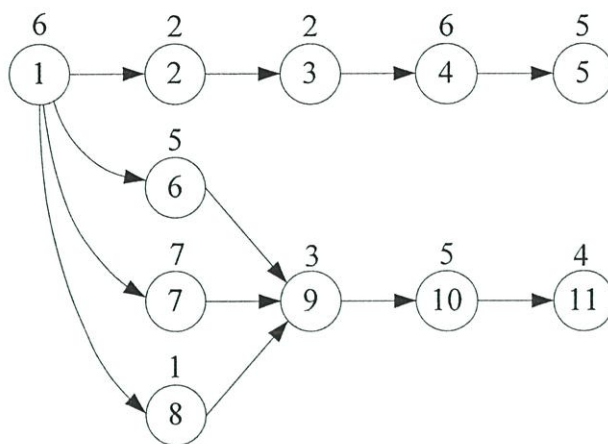
รูปที่ 4.30 อัตราการลู่เข้าหาคำตอบของ TS ทดสอบกับ BF เมื่อ $R = 100\%$ และ $N = 60$

4.4.2 การทดสอบโปรแกรมการค้นหาแบบตามเชิงพันทาง

การทดสอบโปรแกรมการค้นหาแบบตามเชิงพันทาง หรือโปรแกรม HTS เพื่อการแก้ปัญหา ALB แบบผลิตภัณฑ์เดียว โดยที่ปัญหา ALB ที่ใช้ในการทดสอบโปรแกรมในหัวข้อนี้เป็นปัญหาที่มีจำนวนชั้นงานเท่ากับ 11 ชั้นงาน [9] ดังรายละเอียดที่แสดงในตารางที่ 4.3 กำหนดให้รอบเวลาการผลิต (c) = 10 นาที, เวลาการประกอบรวม (W) = 46 นาที, สถานีงาน ($m = W/c$) = 5 สถานีงาน และสถานีงานสูงสุด (m_{\max}) = 6 สถานีงาน จากตารางที่ 4.3 สามารถเขียนแสดงด้วยแผนภูมิลำดับงานก่อนหลังได้ดังรูปที่ 4.31

ตารางที่ 4.3 รายละเอียดปัญหา ALB ที่ใช้ทดสอบโปรแกรม HTS

ชั้นงาน	เวลาดำเนินงานของชั้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชั้นงาน (ชั้นงานที่ต้องดำเนินการก่อน)
1	6	-
2	2	1
3	2	2
4	6	3
5	5	4
6	5	1
7	7	1
8	1	1
9	3	6,7,8
10	5	9
11	4	10



รูปที่ 4.31 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ในตารางที่ 4.3

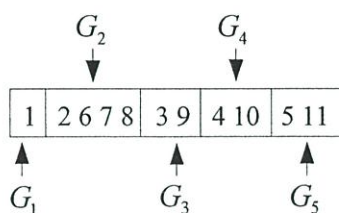
กำหนดให้ปริภูมิการค้นหาของสถานีงาน (S) เป็น $S_1 \in [1, 3], S_2 \in [1, 2], S_3 \in [1, 3], S_4 \in [1, 2], S_5 \in [1, 3]$ และ $S_6 \in [1, 2]$ และเนื่องจาก HTS ตั้งอยู่บนพื้นฐานของกระบวนการสุ่ม ดังนั้นการทดสอบจะต้องกระทำด้วยจำนวนการค้นหาที่มากเพียงพอและด้วยค่าคำตอบเริ่มต้นที่ไม่เท่ากัน เพื่อเป็นการยืนยันสมรรถนะในการแก้ปัญหา ALB ในวิทยานิพนธ์นี้จะใช้โปรแกรม HTS ที่พัฒนาขึ้นด้วยโปรแกรม MATLAB ดำเนินการบนเครื่องคอมพิวเตอร์ Pentium 4 ความเร็ว 1.6 จิกกะเฮิรตซ์ ฟังก์ชันวัตถุประสงค์ที่ใช้คือค่าความแปรปรวนของภาระงาน (w_v) ดังแสดงในสมการที่ (4.5)

$$w_v = \frac{\sum_{k=1}^{m_{\max}} \left[t(S_k) - \left(\frac{W}{m_{\max}} \right) \right]^2}{m_{\max}} \quad (4.5)$$

เมื่อพิจารณาตามบริบทของการหาค่าที่เหมาะสมที่สุด เราสามารถเขียนได้ดังสมการที่ (4.6)

$$\begin{aligned} \min \quad & w_v \\ \text{subject to} \quad & S_1 \in [1, 3], S_2 \in [1, 2], \\ & S_3 \in [1, 3], S_4 \in [1, 2], \\ & S_5 \in [1, 3], S_6 \in [1, 2], \\ & \text{precedence constraints} \end{aligned} \quad (4.6)$$

การจัดกลุ่มงานเพื่อการสลับที่แบบสุ่มสำหรับปัญหา ALB นี้ เมื่อพิจารณาจากตารางที่ 4.3 สามารถทำการจัดกลุ่มงานเพื่อการสลับที่แบบสุ่มได้ 5 กลุ่ม ดังแสดงในรูปที่ 4.32



รูปที่ 4.32 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา ALB ในตารางที่ 4.3

การค้นหาแบบตามูเชิงพันทาง หรือ HTS มีตัวประกอบการค้นหาหรือค่าพารามิเตอร์ของการค้นหาที่ส่งผลกระทบต่อสมรรถนะการแก้ปัญหา ALB อย่างชัดเจนดังนี้

1. R หรือรัศมีการค้นหา
2. N หรือจำนวนสมาชิกค่าใกล้เคียง

การทดสอบ โปรแกรม HTS จะเป็นการทดสอบผลกระทบของตัวประกอบการค้นหา R และ N โดยที่การทดสอบจะกำหนดให้ตัวประกอบการค้นหาดังกล่าวมีค่าดังนี้

- $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา
- $N = 15, 30, 45,$ และ 60 ตัว

ตัวประกอบการค้นหาข้างต้นจะได้รับการปรับทีละตัว ขณะที่ตัวอื่นจะถูกคงค่าเอาไว้ การปรับค่าตัวประกอบการค้นหาแต่ละครั้ง จะทำการทดสอบจำนวน 20 ครั้ง ค่าคำตอบเริ่มต้นจะถูกกำหนดแบบสุ่มโดยโปรแกรม MATLAB ทั้งนี้เพื่อเป็นการประกันว่าค่าคำตอบเริ่มต้นในแต่ละครั้งของการค้นหาจะมีค่าไม่เท่ากัน และกระบวนการค้นหาจะยุติลงเมื่อ TC สอดคล้อง ซึ่ง TC ที่ใช้คือ จำนวนรอบของการค้นหาของ HTS มีค่าเท่ากับจำนวนรอบของการค้นหาสูงสุด ($count_max$) ซึ่งกำหนดให้มีค่าเท่ากับ 30 รอบ สำหรับการสลับที่แบบสุ่มเพื่อจัดสรรลำดับของชิ้นงานที่เหมาะสมตามความสัมพันธ์ก่อนหลังของชิ้นงานได้ดำเนินการสลับที่แบบสุ่มจำนวน 500 ครั้ง ผลการทดสอบ โปรแกรม HTS ที่พัฒนาขึ้นด้วยโปรแกรม MATLAB เพื่อแก้ปัญหา ALB มีรายละเอียดดังแสดงในตารางที่ 4.4

จากตารางที่ 4.4 พบว่ากระบวนการค้นหาของ HTS สามารถหาคำตอบของปัญหา ALB ได้อย่างมีประสิทธิภาพ นั่นคือทำให้ $w_v = 0.8889$ ในทุกการปรับค่าตัวประกอบการค้นหา อย่างไรก็ตาม เวลาที่ใช้ในการค้นหาเฉลี่ยจะเพิ่มขึ้นตามจำนวนค่าใกล้เคียงที่กำหนดเพิ่มขึ้น รูปที่ 4.33 ถึง 4.44 แสดงการลู่เข้าของค่า w_v ซึ่งกำหนดให้เป็นค่าฟังก์ชันวัตถุประสงค์เมื่อทำการปรับ ตัวประกอบการค้นหาทีละตัว เมื่อพิจารณาจากรูปที่ 4.33 ถึง 4.44 จะพบว่า HTS จะมีอัตราการลู่เข้าหาคำตอบของปัญหา ALB อย่างมีประสิทธิภาพเมื่อปรับ $R = 20$ ถึง 60% ของปริภูมิการค้นหา และ $N = 45$ ถึง 60 ตัว คำตอบของปัญหา ALB ที่เหมาะสมที่ได้จาก HTS แสดงในตาราง

ที่ 4.5 และรูปที่ 4.45 ตามลำดับ ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 0.8889, เวลาว่างงานรวม = 14 นาที, และค่าประสิทธิภาพของสายงานการประกอบ = 85.1852%

อย่างไรก็ตาม ดังที่ได้กล่าวมาแล้วว่า การปรับค่าตัวประกอบการค้นหาที่เหมาะสมจะขึ้นอยู่กับปัญหาที่สนใจ นั่นหมายความว่าเมื่อนำ HTS ไปประยุกต์ใช้กับปัญหา ALB ปัญหาอื่น จำเป็นที่จะต้องทำการทดสอบ HTS เพื่อหาค่าตัวประกอบการค้นหาที่เหมาะสมใหม่ โดยอาจใช้ค่าที่สรุปไว้เป็นแนวทางตั้งต้นได้เท่านั้น

ตารางที่ 4.4 ผลการทดสอบโปรแกรม HTS เพื่อแก้ปัญหา ALB ที่มีชิ้นงานจำนวน 11 ชิ้นงาน

จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 20% ของปริภูมิการค้นหา				
	min-w _v	std-w _v	min-idlt (min)	max-eff (%)	average search time (sec)
15	0.8889	0.0000	14	85.1852	10.1734
30	0.8889	0.0000	14	85.1852	19.4586
45	0.8889	0.0000	14	85.1852	29.7446
60	0.8889	0.0000	14	85.1852	39.0868
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 40% ของปริภูมิการค้นหา				
	min-w _v	std-w _v	min-idlt (min)	max-eff (%)	average search time (sec)
15	0.8889	0.0000	14	85.1852	9.8718
30	0.8889	0.0000	14	85.1852	19.6603
45	0.8889	0.0000	14	85.1852	29.3984
60	0.8889	0.0000	14	85.1852	38.6047
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 60% ของปริภูมิการค้นหา				
	min-w _v	std-w _v	min-idlt (min)	max-eff (%)	average search time (sec)
15	0.8889	0.0000	14	85.1852	9.7922
30	0.8889	0.0000	14	85.1852	19.5772
45	0.8889	0.0000	14	85.1852	29.5547
60	0.8889	0.0000	14	85.1852	38.4602

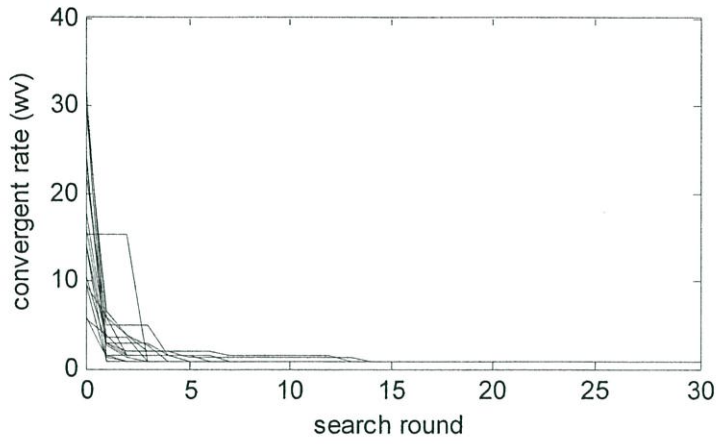
เมื่อ min-w_v คือค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุด,

std-w_v คือค่าเบี่ยงเบนมาตรฐานของค่าความแปรปรวนของภาระงาน,

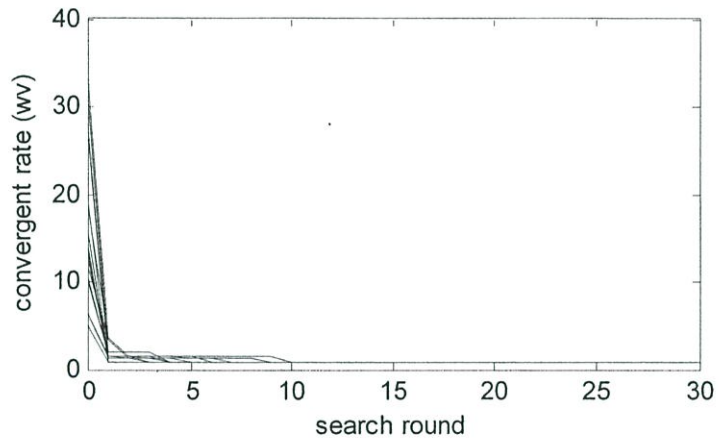
min-idlt คือค่าเวลาว่างงานที่มีค่าน้อยที่สุด,

max-eff คือค่าประสิทธิภาพของสายงานการประกอบที่มีค่ามากที่สุด,

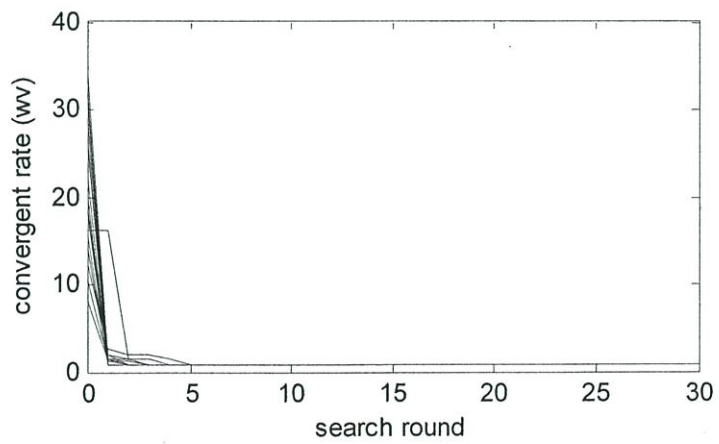
และ average search time คือเวลาที่ใช้ในการค้นหาเฉลี่ย



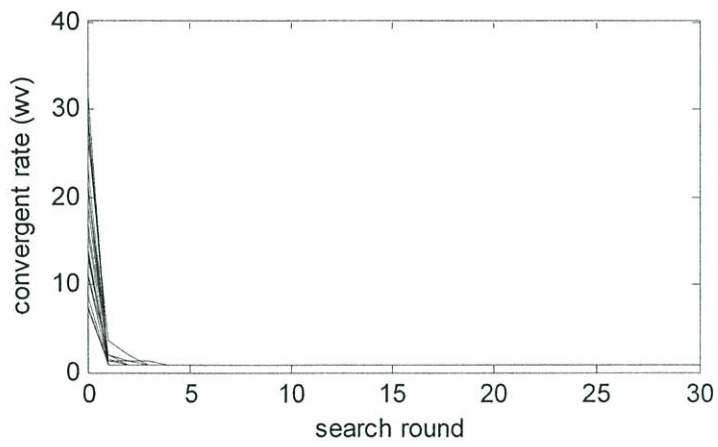
รูปที่ 4.33 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 20\%$ และ $N = 15$



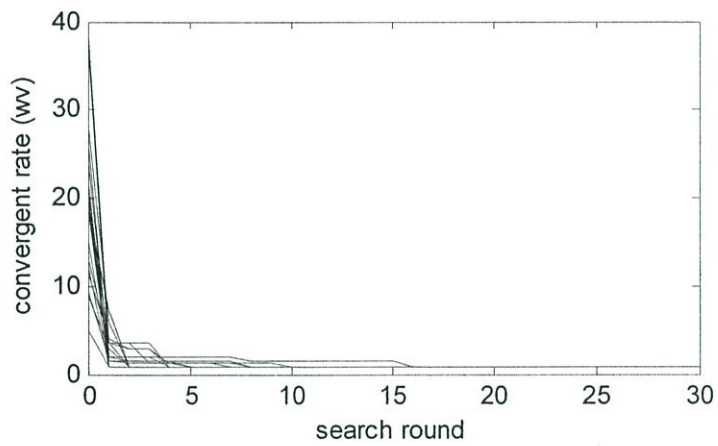
รูปที่ 4.34 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 20\%$ และ $N = 30$



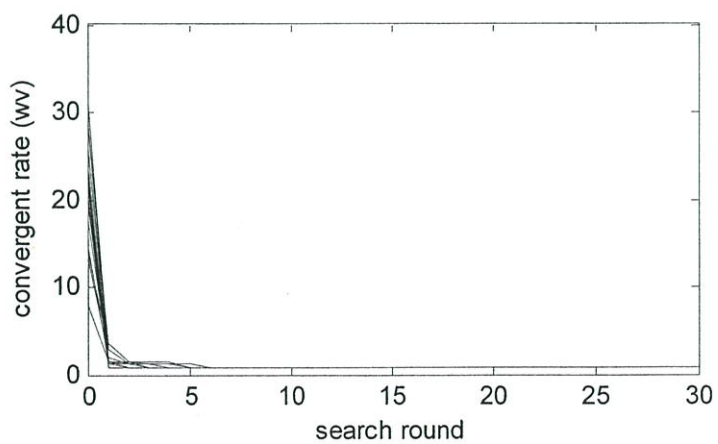
รูปที่ 4.35 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 20\%$ และ $N = 45$



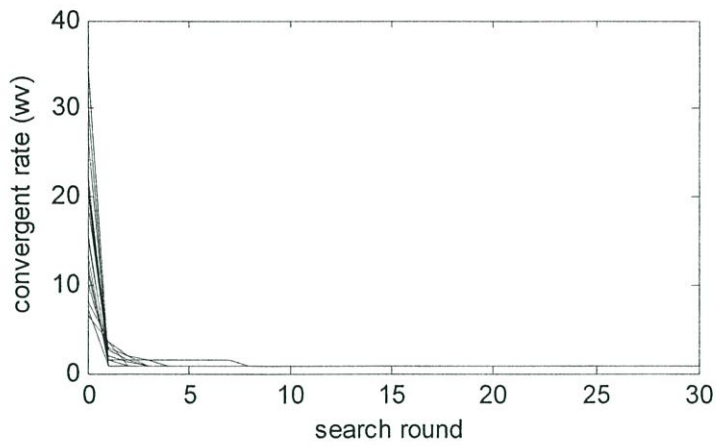
รูปที่ 4.36 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 20\%$ และ $N = 60$



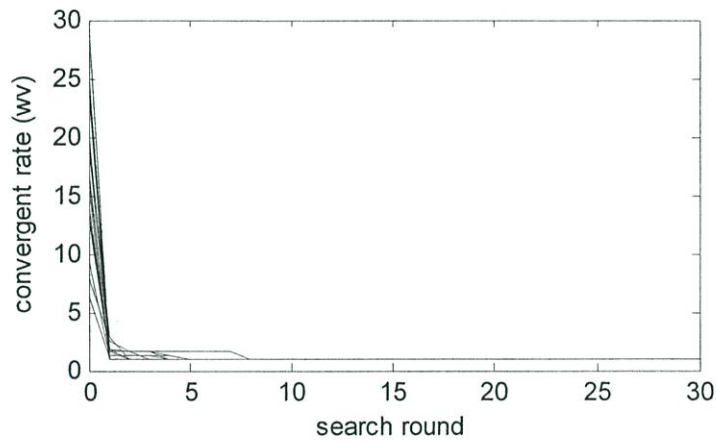
รูปที่ 4.37 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 40\%$ และ $N = 15$



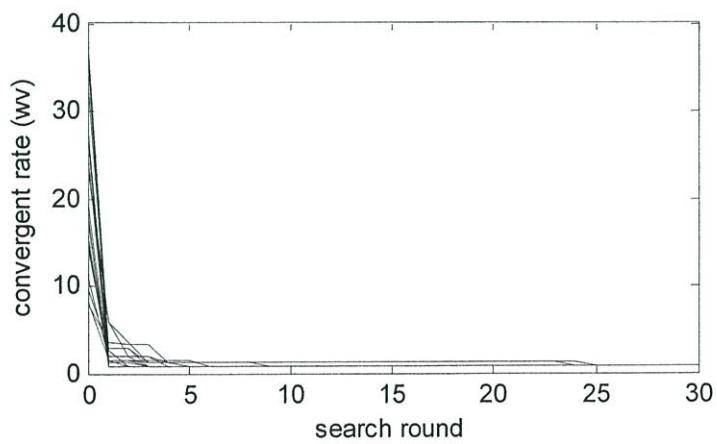
รูปที่ 4.38 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 40\%$ และ $N = 30$



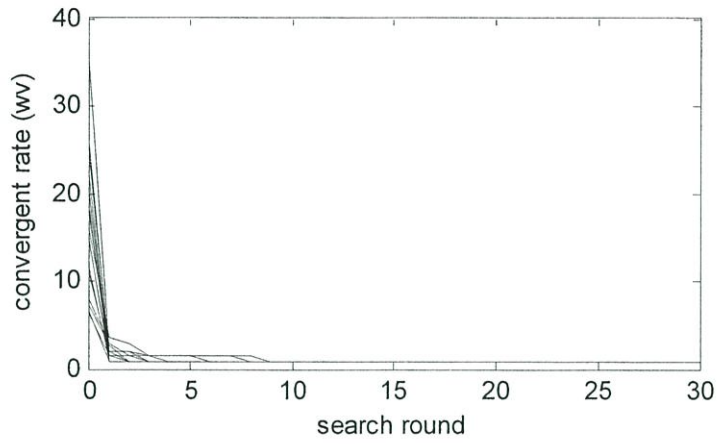
รูปที่ 4.39 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 40\%$ และ $N = 45$



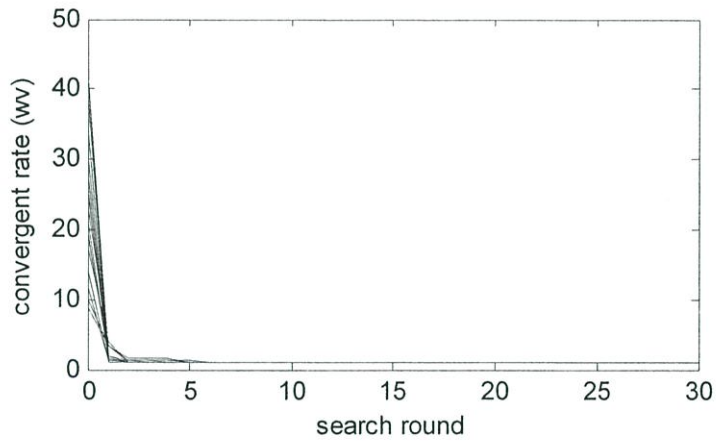
รูปที่ 4.40 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 40\%$ และ $N = 60$



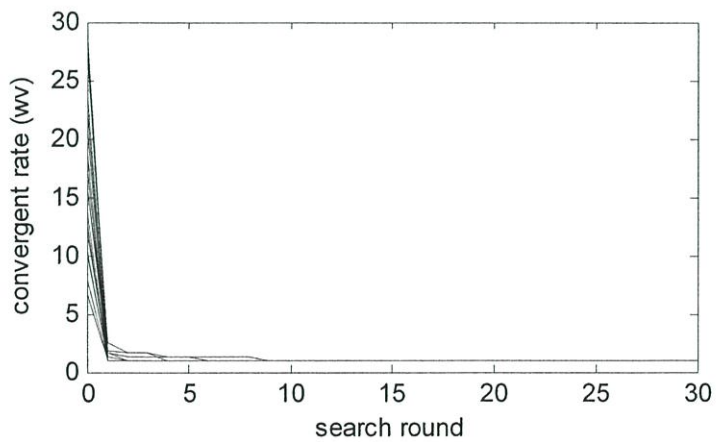
รูปที่ 4.41 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 60\%$ และ $N = 15$



รูปที่ 4.42 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 60\%$ และ $N = 30$



รูปที่ 4.43 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 60\%$ และ $N = 45$

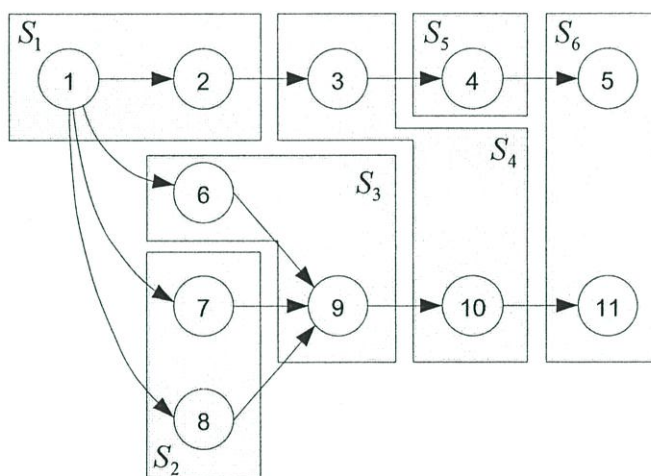


รูปที่ 4.44 อัตราการลู่เข้าหาคำตอบของปัญหา ALB 11 ชั้นงาน เมื่อ $R = 60\%$ และ $N = 60$

ตารางที่ 4.5 ผลการแก้ปัญหา ALB จำนวน 11 ชิ้นงานที่ได้จากวิธี HTS

สถานีงาน	ชิ้นงาน	เวลาดำเนินงานของสถานีงาน (นาที)	เวลาว่างงาน(นาที)
S_1	1,2	8	2
S_2	7,8	8	2
S_3	6,9	8	2
S_4	3,10	7	3
S_5	4	6	4
S_6	5,11	9	1

เวลาว่างงานรวม = 14 นาที
 ค่าความแปรปรวนของภาระงาน = 0.8889
 ประสิทธิภาพของสายงานการประกอบ = 85.1852%



รูปที่ 4.45 ผลการแก้ปัญหา ALB จำนวน 11 ชิ้นงานที่ได้จากวิธี HTS

4.5 สรุป

บทที่ 4 นี้ได้กล่าวถึงการค้นหาแบบตามู (TS) อัลกอริทึมของ TS การค้นหาแบบตามูเชิงพันทาง (HTS) ที่ได้รับการพัฒนาขึ้นเพื่อแก้ปัญหา ALB และอัลกอริทึมของ HTS จากอัลกอริทึมดังกล่าวได้ถูกพัฒนาเป็น โปรแกรม TS และ โปรแกรม HTS โดยอาศัยโปรแกรม MATLAB นอกจากนี้ในบทที่ 4 ยังได้แสดงผลการทดสอบโปรแกรม TS กับปัญหาการหาค่าต่ำสุดของฟังก์ชัน โบฮาเชฟสกี และผลการทดสอบ โปรแกรม HTS เพื่อแก้ปัญหา ALB อย่างง่ายที่มีชิ้นงานจำนวน 11 ชิ้นงาน จากผลการทดสอบโปรแกรม TS พบว่าสามารถค้นหาคำตอบ (ใกล้เคียง) วงกว้างของฟังก์ชัน โบฮาเชฟสกีได้อย่างมีประสิทธิภาพ และผลการทดสอบ โปรแกรม HTS พบว่าสามารถค้นหาคำตอบที่เหมาะสมของปัญหา ALB ได้อย่างน่าพึงพอใจ

บทที่ 5

ผลการทดสอบ

5.1 กล่าวนำ

ในบทที่ 5 นี้จะกล่าวถึงการทดสอบการแก้ปัญหา ALB โดยการค้นหาแบบตาบู่เชิงพันทาง ที่ได้รับการพัฒนาขึ้น โดยปัญหา ALB ที่ใช้ในวิทยานิพนธ์นี้จะเป็นปัญหา ALB แบบผลิตภัณฑ์เดียวที่มีการกำหนดเวลาดำเนินงานของชิ้นงานให้มีค่าคงที่ (SMD) จำนวน 4 ปัญหาซึ่งได้มาจากการสำรวจงานวิจัยที่เกี่ยวข้อง ผลการแก้ปัญหา ALB โดยการค้นหาแบบตาบู่เชิงพันทาง หรือ HTS จะได้รับการนำไปเปรียบเทียบกับวิธี COMSOAL ดังรายละเอียดต่อไปนี้

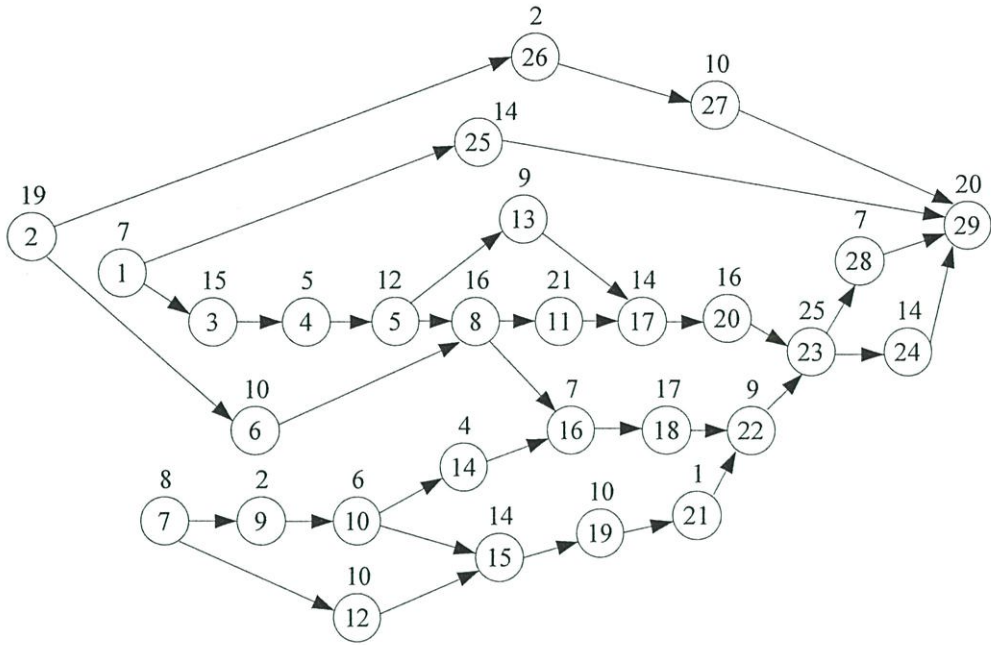
5.2 ปัญหา ALB ที่ใช้ในการทดสอบ

ปัญหาการจัดสมมูลสายงานการประกอบ หรือปัญหา ALB ที่ใช้ในการทดสอบเป็นปัญหา ALB แบบผลิตภัณฑ์เดียวแบบ SMD ซึ่งมีอยู่ด้วยกัน 4 ปัญหา ดังแสดงในตารางที่ 5.1

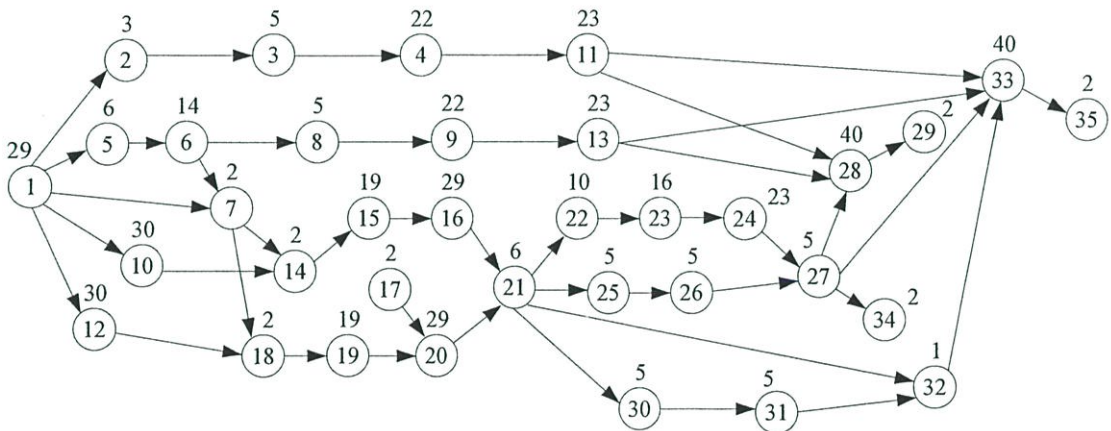
ตารางที่ 5.1 ปัญหา ALB ที่ใช้ในการทดสอบ

ปัญหาที่	ชื่อปัญหา	จำนวนชิ้นงาน	เวลาดำเนินงานของชิ้นงาน	ลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน	รอบเวลาการผลิต
1	Buxey	29	กำหนดมาให้พร้อมกับปัญหา	กำหนดมาให้พร้อมกับปัญหา	กำหนดแบบไม่เจาะจง
2	Gunther	35	กำหนดมาให้พร้อมกับปัญหา	กำหนดมาให้พร้อมกับปัญหา	กำหนดแบบไม่เจาะจง
3	Random	45	สุ่มแบบไม่เจาะจง	สุ่มแบบไม่เจาะจง	กำหนดแบบไม่เจาะจง
4	M-cycle	60	กำหนดมาให้พร้อมกับปัญหา	กำหนดมาให้พร้อมกับปัญหา	กำหนดแบบไม่เจาะจง

ปัญหา ALB ปัญหาที่ 1 คือปัญหา Buxey [14],[62-63] มีจำนวนชิ้นงานทั้งหมด 29 ชิ้นงาน เวลาการประกอบรวม (T) = 324 นาที สำหรับเวลาดำเนินงานของชิ้นงาน และลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน แสดงด้วยแผนภูมิลำดับงานก่อนหลังดังรูปที่ 5.1



รูปที่ 5.1 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ปัญหาที่ 1 (Buxey)



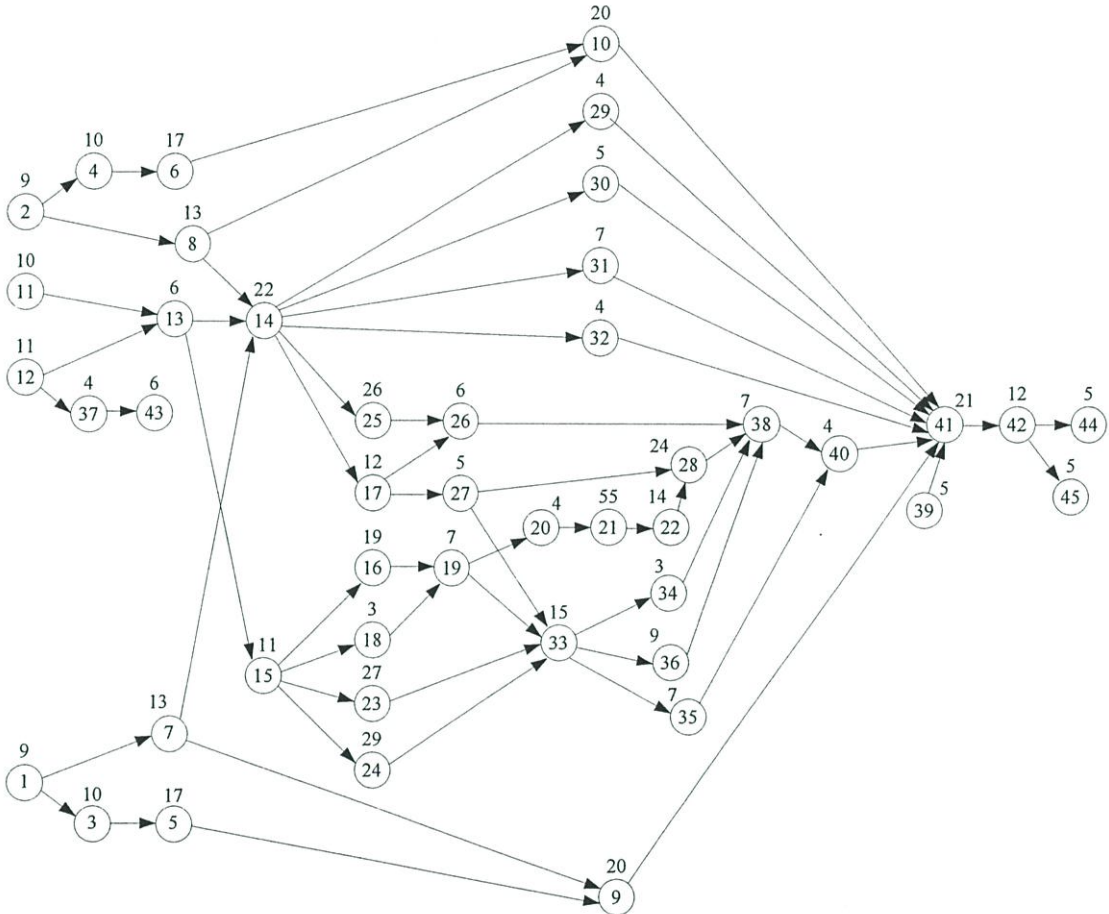
รูปที่ 5.2 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ปัญหาที่ 2 (Gunther)

ปัญหา ALB ปัญหาที่ 2 คือปัญหา Gunther [14],[62-63] มีจำนวนชิ้นงานทั้งหมด 35 ชิ้นงาน เวลาการประกอบรวม = 483 นาที สำหรับเวลาดำเนินงานของชิ้นงาน และลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน แสดงด้วยแผนภูมิลำดับงานก่อนหลังดังรูปที่ 5.2

ปัญหา ALB ปัญหาที่ 3 คือปัญหา Random ซึ่งเป็นปัญหาที่ถูกกำหนดขึ้นมาแบบสุ่ม ในลักษณะที่ไม่เจาะจง มีจำนวนชิ้นงานทั้งหมด 45 ชิ้นงาน เวลาการประกอบรวม = 552 นาที สำหรับเวลาดำเนินงานของชิ้นงาน และลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน แสดงด้วยแผนภูมิลำดับงานก่อนหลังดังรูปที่ 5.3

ปัญหา ALB ปัญหาที่ 4 คือปัญหาสายงานการประกอบมอเตอร์ไซค์ หรืออาจเรียกว่าปัญหา M-cycle ซึ่งเป็นปัญหาการจัดสมดุลสายงานการประกอบจริงในภาคอุตสาหกรรม โดยใช้

รายละเอียดของสายงานการประกอบรถยนต์ไฮด์ของ บริษัททวาซากิ มอเตอร์ เอ็นเตอร์ไพรส์ ประเทศไทย จำกัด [64] ซึ่งมีจำนวนชิ้นงานทั้งหมด 60 ชิ้นงาน เวลาการประกอบรวม = 2,475 วินาที สำหรับเวลาดำเนินงานของชิ้นงาน และลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน แสดงด้วยแผนภูมิลำดับงานก่อนหลังดังรูปที่ 5.4

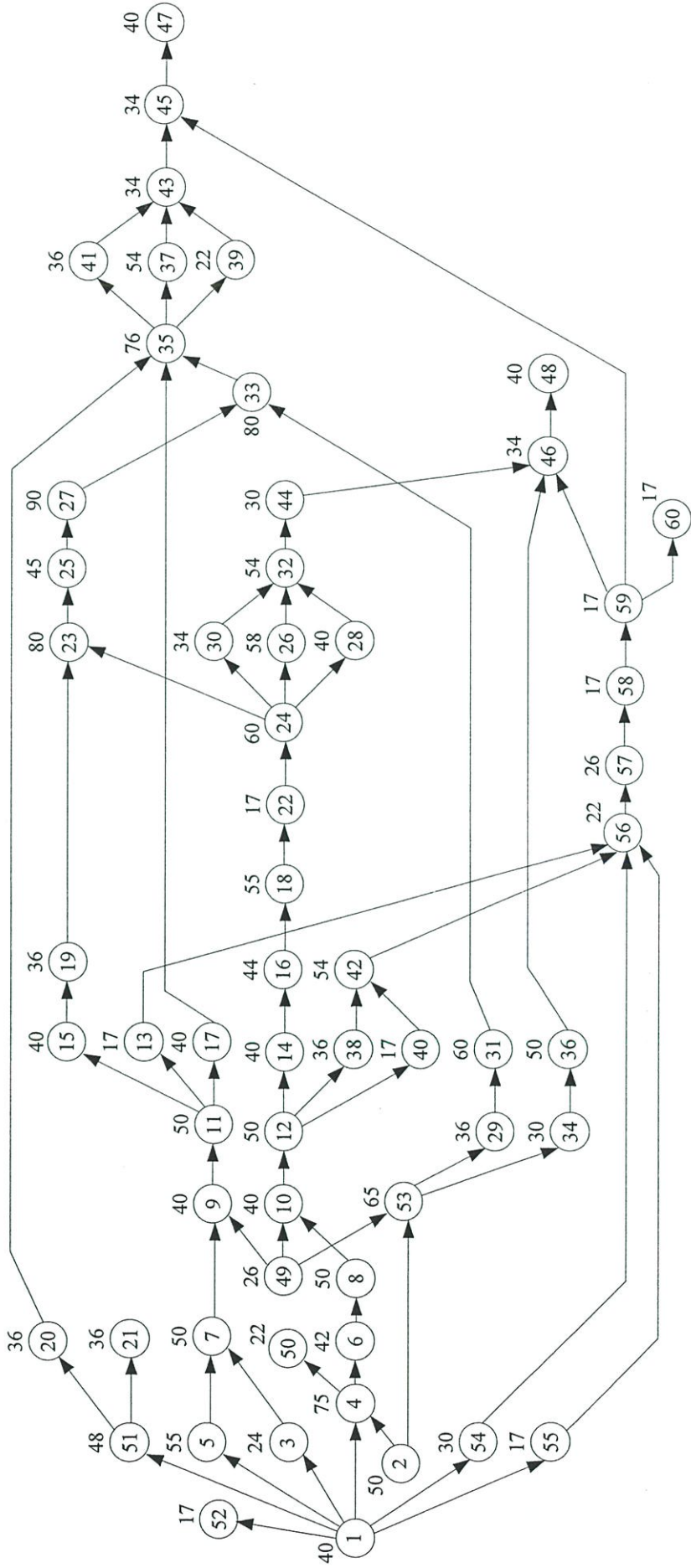


รูปที่ 5.3 แผนภูมิลำดับงานก่อนหลังของปัญหา ALB ปัญหาที่ 3 (Random)

รายละเอียดของปัญหา ALB ทั้ง 4 ปัญหาได้รับแสดงไว้ในภาคผนวก ก. สำหรับตารางที่ 5.2 แสดงข้อมูลที่จำเป็นในการทดสอบปัญหา ALB ทั้ง 4 ปัญหา

ตารางที่ 5.2 ปัญหา ALB ที่ใช้ในการทดสอบ

ปัญหา ที่	ชื่อปัญหา	จำนวน ชิ้นงาน	เวลาการ ประกอบรวม	รอบเวลา การผลิต	จำนวนสถานีงาน สูงสุดที่ยอมรับได้
1	Buxey	29	324 นาที	50 นาที	8
2	Gunther	35	483 นาที	60 นาที	10
3	Random	45	552 นาที	80 นาที	8
4	M-cycle	60	2,475 วินาที	360 วินาที	8



รูปที่ 5.4 แผนภูมิตำดับงานก่อนหลังของปัญหา ALB ปัญหาที่ 4 (M-cycle)

5.3 ขั้นตอนการทดสอบ

ขั้นตอนการทดสอบการแก้ปัญหา ALB โดยการค้นหาแบบตามูเชิงพันทาง หรือ HTS มีรายละเอียดดังต่อไปนี้

1. กำหนดรอบเวลาการผลิต (c) ให้กับปัญหา ALB ดังแสดงในตารางที่ 5.2

- ปัญหา Buxey : กำหนดรอบเวลาการผลิตเท่ากับ 50 นาที
- ปัญหา Gunther : กำหนดรอบเวลาการผลิตเท่ากับ 60 นาที
- ปัญหา Random : กำหนดรอบเวลาการผลิตเท่ากับ 80 นาที
- ปัญหา M-cycle : กำหนดรอบเวลาการผลิตเท่ากับ 360 วินาที

2. คำนวณจำนวนสถานีงาน (m) โดยอาศัยความสัมพันธ์จากสมการที่ (3.1) ซึ่งได้รับการแสดงไว้อีกครั้งในสมการที่ (5.1) เพื่อกำหนดจำนวนสถานีงานสูงสุดที่ยอมรับได้ (m_{\max}) ดังสมการที่ (5.2) ให้กับปัญหา ALB ดังแสดงในตารางที่ 5.2

$$m = \frac{\sum_{j=1}^n t_j}{c} = \frac{W}{c} \quad (5.1)$$

$$m_{\max} = m + 1 \quad (5.2)$$

- ปัญหา Buxey :

สถานีงาน $m = W/c = 324/50 = 6.48 \approx 7$ สถานีงาน, $m_{\max} = 8$ สถานีงาน

- ปัญหา Gunther :

สถานีงาน $m = W/c = 483/60 = 8.05 \approx 9$ สถานีงาน, $m_{\max} = 10$ สถานีงาน

- ปัญหา Random :

สถานีงาน $m = W/c = 552/80 = 6.90 \approx 7$ สถานีงาน, $m_{\max} = 8$ สถานีงาน

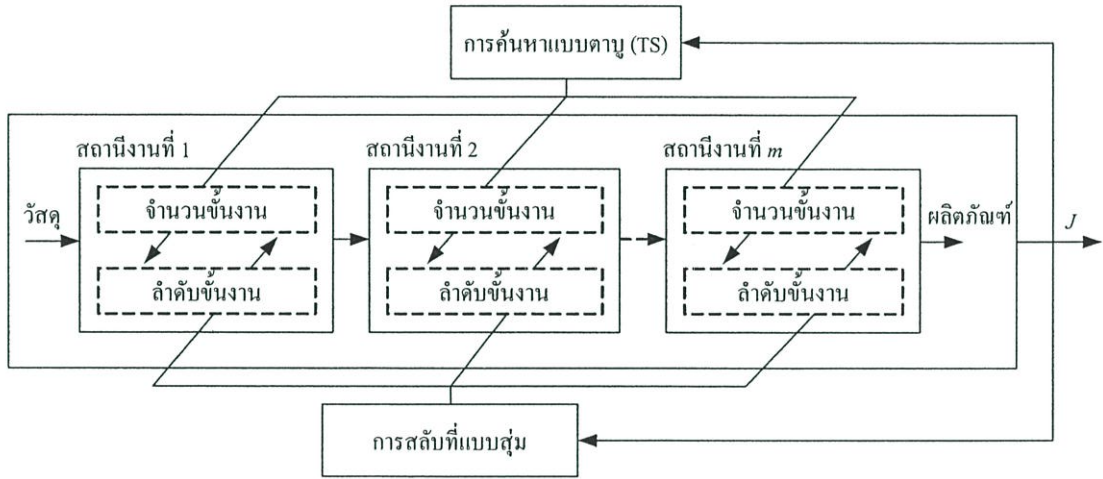
- ปัญหา M-cycle :

สถานีงาน $m = W/c = 2475/360 = 6.88 \approx 7$ สถานีงาน, $m_{\max} = 8$ สถานีงาน

3. ทำการแก้ปัญหาคำการจัดสมดุลสายงานการประกอบทั้ง 4 ปัญหาด้วยวิธี COMSOAL เพื่อให้ค่าความแปรปรวนของภาระงาน (w_v) ดังแสดงในสมการที่ (5.3) มีค่าน้อยที่สุด

$$w_v = \frac{\sum_{k=1}^{m_{\max}} \left[t(S_k) - \left(\frac{W}{m_{\max}} \right) \right]^2}{m_{\max}} \quad (5.3)$$

4. ทำการแก้ปัญหาการจัดสมดุลสายงานการประกอบทั้ง 4 ปัญหาด้วยวิธี HTS โดยกำหนดให้ค่าความแปรปรวนของภาระงานเป็นค่าฟังก์ชันวัตถุประสงค์ (J) ดังสมการที่ (5.4) จากรูปที่ 5.5 ซึ่งแสดงการแก้ปัญหา ALB ด้วย HTS พบว่า TS จะทำหน้าที่จัดสรรจำนวนชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน ในขณะที่การสลับที่แบบสุ่มจะทำหน้าที่จัดลำดับของชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน ค่า J จะถูกป้อนกลับไปยัง TS และการสลับที่แบบสุ่มเพื่อให้มีค่าน้อยที่สุดและสอดคล้องกับลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน



รูปที่ 5.5 แผนผังการแก้ปัญหา ALB ด้วยวิธี HTS ที่นำเสนอ

$$J = w_v = \frac{\sum_{k=1}^{m_{\max}} \left[t(S_k) - \left(\frac{W}{m_{\max}} \right) \right]^2}{m_{\max}} \quad (5.4)$$

5. เปรียบเทียบผลการแก้ปัญหาการจัดสมดุลสายงานการประกอบทั้ง 4 ปัญหา โดยอาศัยค่าความแปรปรวนของภาระงานที่ได้จากวิธี COMSOAL และวิธี HTS

5.4 การกำหนดค่าพารามิเตอร์ของ HTS

เนื่องจาก HTS ตั้งอยู่บนพื้นฐานของกระบวนการสุ่ม ดังนั้นการแก้ปัญหา ALB จะต้องกระทำด้วยจำนวนการค้นหาที่มากเพียงพอเพื่อเป็นการยืนยันคำตอบที่พบ ในการทดสอบนี้ได้กำหนดฟังก์ชันวัตถุประสงค์เป็นค่าความแปรปรวนของภาระงานดังแสดงในสมการที่ (5.4) วิธี HTS ที่พัฒนาขึ้นด้วยโปรแกรม MATLAB ได้รับการทดสอบบนเครื่องคอมพิวเตอร์ Pentium 4 ความเร็ว 1.6 จิกกะเฮิร์ตซ์ กำหนดให้จำนวนรอบการค้นหาสูงสุด ($count_max$) เป็นเกณฑ์ยุติการค้นหา (TC)

ซึ่งกำหนดให้มีค่าเท่ากับ 50 รอบ ในแต่ละปัญหา ALB ได้ทำการทดสอบทั้งหมดจำนวน 20 ครั้ง ค่าพารามิเตอร์ของ HTS ในแต่ละปัญหา ALB ได้มีการกำหนดดังต่อไปนี้

5.4.1 ปัญหา Buxey

- กำหนดรัศมีการค้นหา $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา
- กำหนดจำนวนสมาชิกค่าใกล้เคียง $N = 30, 45, 60$
- การจัดกลุ่มของชั้นงานเพื่อการสลับที่แบบสุ่มแสดงดังรูปที่ 5.6

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

รูปที่ 5.6 การจัดกลุ่มชั้นงานเพื่อการสลับที่แบบสุ่มของปัญหา Buxey

- ทำการสลับที่แบบสุ่มจำนวน 1,000 ครั้ง
- กำหนดให้ปริภูมิการค้นหาของสถานีงาน (S) เป็น $S_1 \in [2, 7], S_2 \in [3, 8], S_3 \in [2, 6], S_4 \in [3, 7], S_5 \in [1, 4], S_6 \in [1, 5], S_7 \in [1, 5],$ และ $S_8 \in [1, 3]$
- กำหนดให้ $J = w_v$ และเมื่อพิจารณาตามบริบทของการหาค่าเหมาะที่สุด เราสามารถเขียนได้ดังสมการที่ (5.5)

$$\begin{aligned}
 & \min && J \\
 & \text{subject to} && S_1 \in [2, 7], S_2 \in [3, 8], \\
 & && S_3 \in [2, 6], S_4 \in [3, 7], \\
 & && S_5 \in [1, 4], S_6 \in [1, 5], \\
 & && S_7 \in [1, 5], S_8 \in [1, 3], \\
 & && \text{precedence constraints}
 \end{aligned} \tag{5.5}$$

5.4.2 ปัญหา Gunther

- กำหนดรัศมีการค้นหา $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา
- กำหนดจำนวนสมาชิกค่าใกล้เคียง $N = 30, 45, 60$
- การจัดกลุ่มของชั้นงานเพื่อการสลับที่แบบสุ่มแสดงดังรูปที่ 5.7
- ทำการสลับที่แบบสุ่มจำนวน 1,000 ครั้ง

- กำหนดให้ปฏิทินการค้นหของสถานีงาน (S) เป็น $S_1 \in [3, 5], S_2 \in [1, 3], S_3 \in [6, 8], S_4 \in [1, 3], S_5 \in [1, 3], S_6 \in [1, 3], S_7 \in [4, 6], S_8 \in [6, 8], S_9 \in [1, 2]$ และ $S_{10} \in [2, 4]$
- กำหนดให้ $J = w$, และเมื่อพิจารณาตามบริบทของการหาค่าที่เหมาะสมที่สุด เราสามารถเขียนได้ดังสมการที่ (5.6)

1	17	2	5	10	12	3	6	4	7	8	9	11	14	18	13	15	19	16	20	21	22	25	30	...
...	23	26	31	24	32	27	28	33	34	29	35													

รูปที่ 5.7 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา Gunther

$$\begin{aligned}
 & \min \quad J \\
 & \text{subject to} \quad S_1 \in [3, 5], S_2 \in [1, 3], \\
 & \quad \quad \quad S_3 \in [6, 8], S_4 \in [1, 3], \\
 & \quad \quad \quad S_5 \in [1, 3], S_6 \in [1, 3], \\
 & \quad \quad \quad S_7 \in [4, 6], S_8 \in [6, 8], \\
 & \quad \quad \quad S_9 \in [1, 2], S_{10} \in [2, 4], \\
 & \quad \quad \quad \text{precedence constraints}
 \end{aligned} \tag{5.6}$$

5.4.3 ปัญหา Random

- กำหนดรัศมีการค้นหา $R = 20, 40$, และ 60% ของปฏิทินการค้นหา
- กำหนดจำนวนสมาชิกค่าใกล้เคียง $N = 30, 45, 60$
- การจัดกลุ่มของชิ้นงานเพื่อการสลับที่แบบสุ่มแสดงดังรูปที่ 5.8
- ทำการสลับที่แบบสุ่มจำนวน 1,000 ครั้ง

1	2	11	12	39	3	4	7	8	37	5	6	43	9	10	13	14	15	...
...	16	17	18	23	24	25	29	30	31	32	19	26	27	20	33	...		
...	21	34	35	36	22	28	38	40	41	42	44	45						

รูปที่ 5.8 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา Random

- กำหนดให้ปฏิทินการค้นหของสถานีงาน (S) เป็น $S_1 \in [7, 9], S_2 \in [5, 7], S_3 \in [3, 5], S_4 \in [6, 8], S_5 \in [5, 7], S_6 \in [3, 5], S_7 \in [5, 7]$, และ $S_8 \in [3, 5]$

- กำหนดให้ $J = w_v$ และเมื่อพิจารณาตามบริบทของการหาค่าที่เหมาะสมที่สุด เราสามารถเขียนได้ดังสมการที่ (5.7)

$$\begin{aligned}
 & \min && J \\
 & \text{subject to} && S_1 \in [7, 9], S_2 \in [5, 7], \\
 & && S_3 \in [3, 5], S_4 \in [6, 8], \\
 & && S_5 \in [5, 7], S_6 \in [3, 5], \\
 & && S_7 \in [5, 7], S_8 \in [3, 5], \\
 & && \text{precedence constraints}
 \end{aligned} \tag{5.7}$$

5.4.4 ปัญหา M-cycle

- กำหนดรัศมีการค้นหา $R = 20, 40$, และ 60% ของปริภูมิการค้นหา
- กำหนดจำนวนสมาชิกค่าใกล้เคียง $N = 30, 45, 60$
- การจัดกลุ่มของชิ้นงานเพื่อการสลับที่แบบสุ่มแสดงดังรูปที่ 5.9
- ทำการสลับที่แบบสุ่มจำนวน 1,000 ครั้ง

1 2 49	3 4 5 51 52 53 54 55	6 7 20 21 29 34 50	8 9 36
...	10 11	12 13 15 17	14 16 19 38 40
...	18 31 42	22 56	...
...	24 57	23 26 28 30 58	25 32 59
...	27 44 60	33	35
...	37 39 41	43	45 46
...	47 48		

รูปที่ 5.9 การจัดกลุ่มชิ้นงานเพื่อการสลับที่แบบสุ่มของปัญหา M-cycle

- กำหนดให้ปริภูมิการค้นหาของสถานีงาน (S) เป็น $S_1 \in [8, 10], S_2 \in [8, 10], S_3 \in [7, 9], S_4 \in [8, 10], S_5 \in [8, 10], S_6 \in [6, 8], S_7 \in [7, 9]$, และ $S_8 \in [1, 4]$
- กำหนดให้ $J = w_v$ และเมื่อพิจารณาตามบริบทของการหาค่าที่เหมาะสมที่สุด เราสามารถเขียนได้ดังสมการที่ (5.8)

$$\begin{aligned}
 & \min && J \\
 & \text{subject to} && S_1 \in [8, 10], S_2 \in [8, 10], \\
 & && S_3 \in [7, 9], S_4 \in [8, 10], \\
 & && S_5 \in [8, 10], S_6 \in [6, 8], \\
 & && S_7 \in [7, 9], S_8 \in [1, 4], \\
 & && \text{precedence constraints}
 \end{aligned} \tag{5.8}$$

5.5 ผลการทดสอบ

ผลการทดสอบการแก้ปัญหาคำตอบของการจัดสมดุลสายงานการประกอบทั้ง 4 ปัญหา ได้แก่ปัญหา Buxey ปัญหา Gunther ปัญหา Random และ ปัญหา M-cycle โดยวิธี HTS ที่นำเสนอในวิทยานิพนธ์ เปรียบเทียบกับผลที่ได้จากวิธี COMSOAL มีรายละเอียดและการอภิปรายผลดังต่อไปนี้

5.5.1 ผลการทดสอบปัญหา Buxey

ผลการทดสอบปัญหา Buxey ด้วยวิธี HTS ที่พัฒนาขึ้นแสดงดังตารางที่ 5.3 จากตารางที่ 5.3 พบว่ากระบวนการค้นหาของ HTS สามารถหาคำตอบของปัญหา Buxey ได้อย่างมีประสิทธิภาพ กล่าวคือ เมื่อปรับให้รัศมีการค้นหา $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา ค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุด ($\min-w_v$) ค่าความแปรปรวนของภาระงานที่มีค่ามากที่สุด ($\max-w_v$) ค่าความแปรปรวนของภาระงานเฉลี่ย ($\text{mean}-w_v$) และค่าเบี่ยงเบนมาตรฐานของค่าความแปรปรวนของภาระงาน ($\text{std}-w_v$) มีค่าใกล้เคียงกัน และเมื่อค่าใกล้เคียง (N) มีจำนวนมากขึ้นจะทำให้อัตราการลู่เข้าหาคำตอบของวิธี HTS มีความแปรปรวนน้อยลง อย่างไรก็ตาม เมื่อ N มีจำนวนมากจะทำให้เวลาที่ใช้ในการค้นหาเพิ่มมากขึ้น

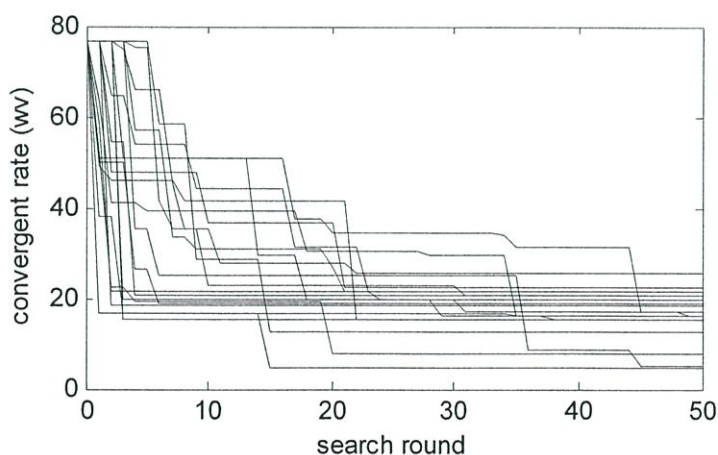
จากตารางที่ 5.3 พบว่าเมื่อปรับ $R = 20\%$ ของปริภูมิการค้นหา และ $N = 45$ ตัว ค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุด ($\min-w_v = 2.50$) คำตอบของปัญหา Buxey ที่เหมาะสมที่ได้จากวิธี HTS แสดงในตารางที่ 5.4 เพื่อเปรียบเทียบกับผลที่ได้จากวิธี COMSOAL และรูปที่ 5.10 ถึง 5.18 แสดงอัตราการลู่เข้าของคำตอบของปัญหา Buxey เมื่อปรับ $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา และ $N = 30, 45,$ และ 60 ตัว ตามลำดับ รูปที่ 5.19 แสดงอัตราการลู่เข้าของคำตอบของปัญหา Buxey ที่ให้ผลการทดสอบดีที่สุดเมื่อปรับ $R = 20\%$ ของปริภูมิการค้นหา และ $N = 45$ ตัว ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 2.50 เวลาว่างงานรวม = 76 นาที และค่าประสิทธิภาพของสายงานการประกอบ = 92.05%

จากปัญหา Buxey เมื่อทำการหาคำตอบโดยวิธี COMSOAL ผลที่ได้แสดงดังตารางที่ 5.4 ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 76.50, เวลาว่างงานรวม = 76 นาที, และค่าประสิทธิภาพของสายงานการประกอบ = 81.00% เมื่อทำการเปรียบเทียบผลการทดสอบการแก้ปัญหาคำตอบของปัญหา Buxey ด้วยวิธี COMSOAL และวิธี HTS ที่นำเสนอพบว่าวิธี HTS สามารถค้นหาคำตอบของปัญหา Buxey ได้ดีกว่าวิธี COMSOAL โดยที่ค่าความแปรปรวนของภาระงานที่ได้จากวิธี HTS มีค่าลดลง 96.73% (ลดลงจาก 76.50 มาเป็น 2.50) เมื่อเทียบกับวิธี COMSOAL และค่าประสิทธิภาพของสายงานการประกอบที่ได้จากวิธี HTS มีค่าเพิ่มขึ้น 13.64% เมื่อเทียบกับวิธี COMSOAL

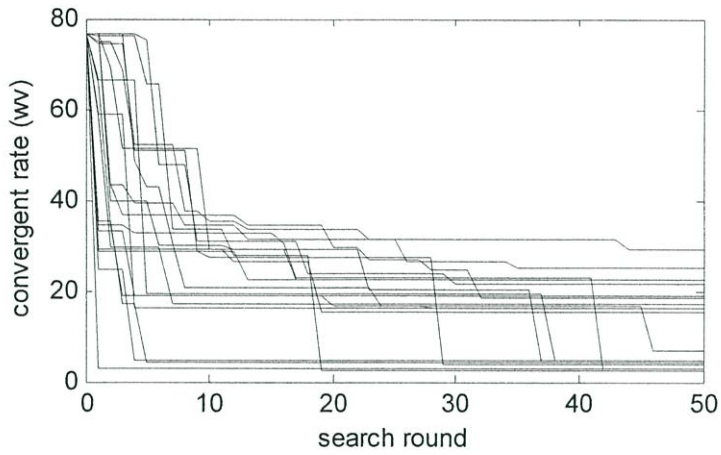
ตารางที่ 5.3 ผลการทดสอบการแก้ปัญหา Buxey ด้วยวิธี HTS

จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 20% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	4.5000	25.7500	16.5875	5.6009	62.2138
45	2.5000	29.0000	13.0125	8.8438	109.1875
60	3.5000	22.7500	12.7500	5.7543	142.8899
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 40% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	4.0000	26.2500	16.6875	5.4975	72.1726
45	3.0000	24.7500	16.3875	5.7115	105.5213
60	6.5000	21.5000	15.2250	3.8771	141.2000
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 60% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	4.0000	31.0000	19.2750	5.9393	72.1828
45	4.0000	24.5000	14.3125	5.9692	108.4999
60	4.0000	22.7500	16.2250	6.3923	129.7554

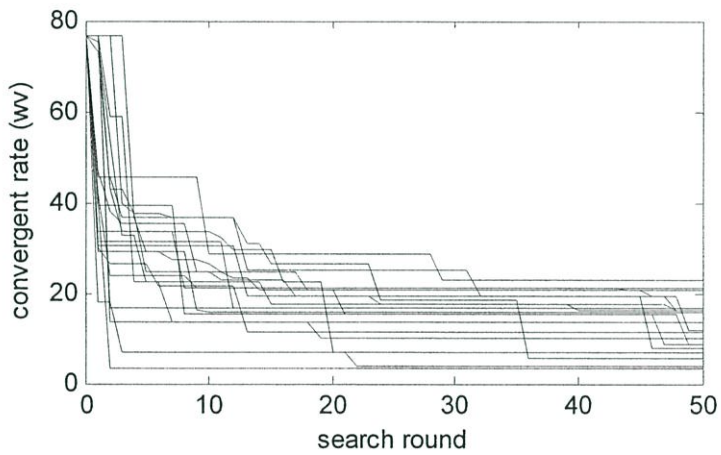
เมื่อ min- w_v คือค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุด,
max- w_v คือค่าความแปรปรวนของภาระงานที่มีค่ามากที่สุด,
mean- w_v คือค่าความแปรปรวนของภาระงานเฉลี่ย,
std- w_v คือค่าเบี่ยงเบนมาตรฐานของค่าความแปรปรวนของภาระงาน,
และ average search time คือเวลาที่ใช้ในการค้นหาเฉลี่ย



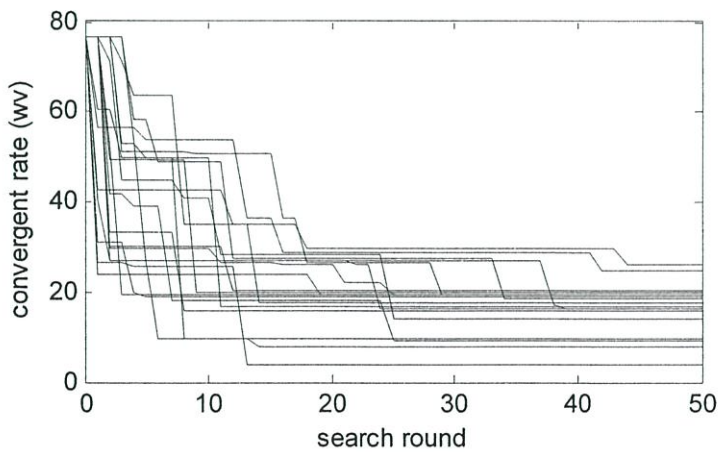
รูปที่ 5.10 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 20\%$ และ $N = 30$



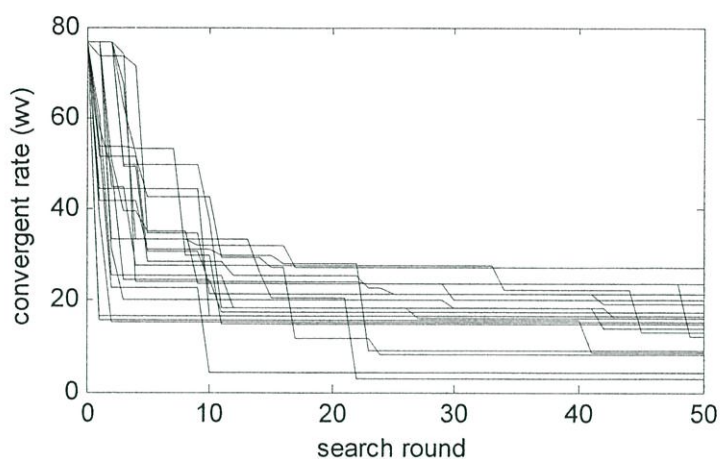
รูปที่ 5.11 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 20\%$ และ $N = 45$



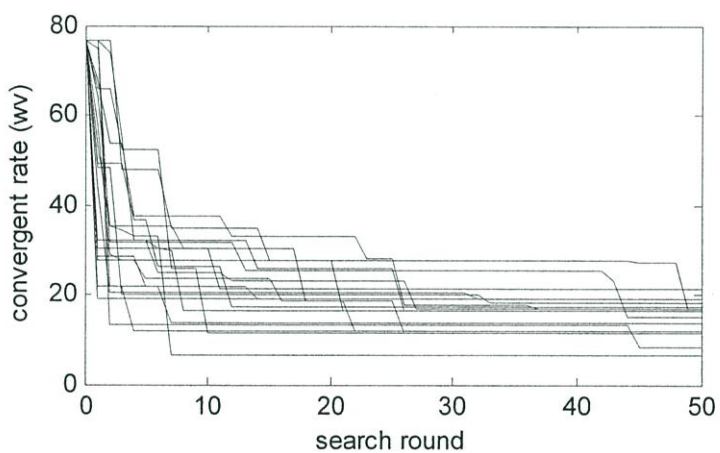
รูปที่ 5.12 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 20\%$ และ $N = 60$



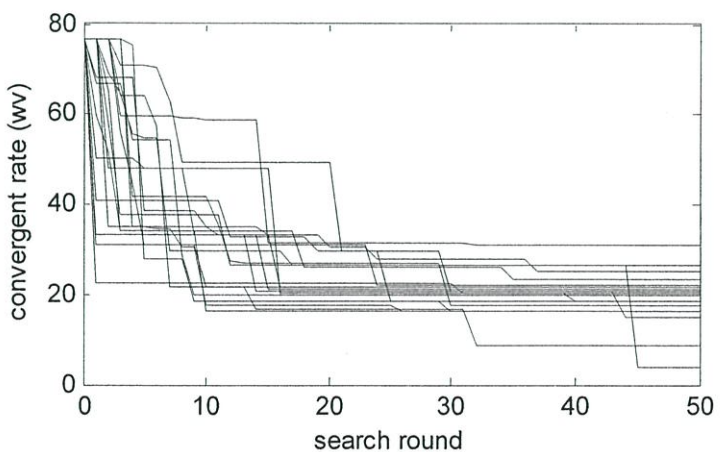
รูปที่ 5.13 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 40\%$ และ $N = 30$



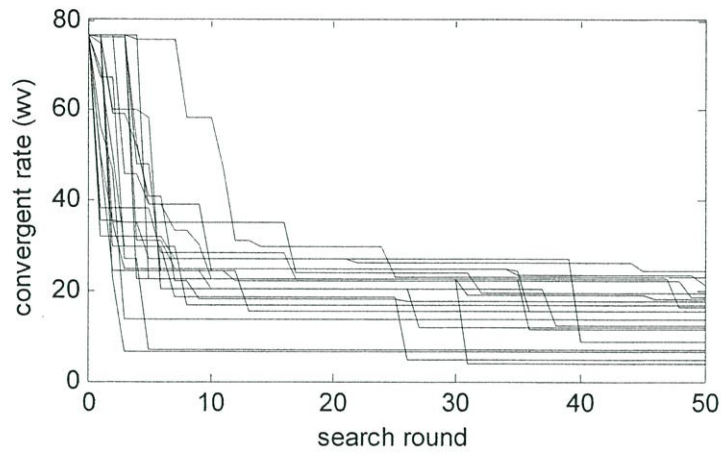
รูปที่ 5.14 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 40\%$ และ $N = 45$



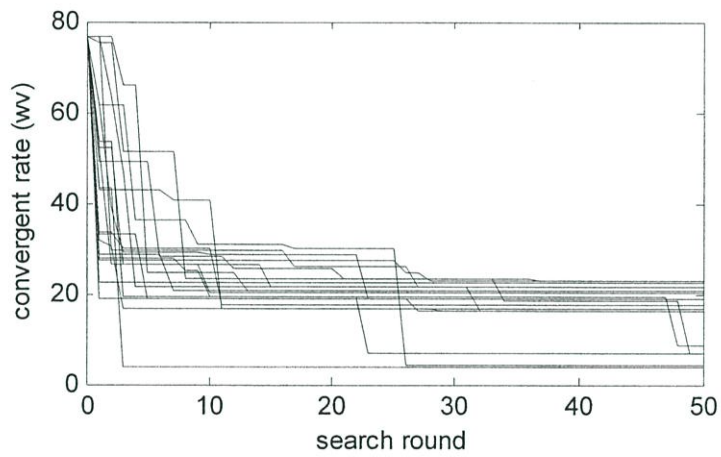
รูปที่ 5.15 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 40\%$ และ $N = 60$



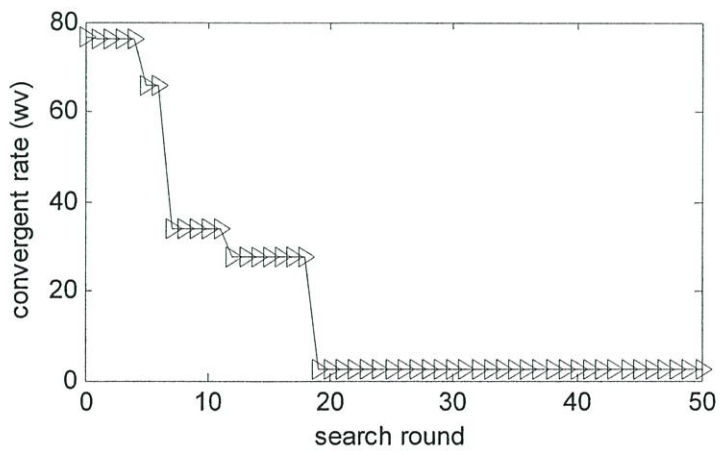
รูปที่ 5.16 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 60\%$ และ $N = 30$



รูปที่ 5.17 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 60\%$ และ $N = 45$



รูปที่ 5.18 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey เมื่อ $R = 60\%$ และ $N = 60$



รูปที่ 5.19 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Buxey ที่ให้ผลดีที่สุด

ตารางที่ 5.4 ผลการเปรียบเทียบการแก้ปัญหา Buxey ด้วยวิธี COMSOAL และวิธี HTS

ผลที่ได้จากวิธี COMSOAL			
สถานีงาน	ชิ้นงาน	เวลาดำเนินงาน ของสถานีงาน (นาที)	เวลาว่างงาน (นาที)
S_1	1,2,7,9,25	50	0
S_2	3,6,12,26,4,10	48	2
S_3	27,5,14,15	40	10
S_4	8,13,19,16,21	43	7
S_5	11,18	38	12
S_6	17,20,22	39	11
S_7	23,24,28	46	4
S_8	29	20	30
<p>เวลาว่างงานรวม = 76 นาที ค่าความแปรปรวนของภาระงาน = 76.50 ประสิทธิภาพของสายงานการประกอบ = 81.00%</p>			
ผลที่ได้จากวิธี HTS			
สถานีงาน	ชิ้นงาน	เวลาดำเนินงาน ของสถานีงาน (นาที)	เวลาว่างงาน (นาที)
S_1	1,7,2,12	44	6
S_2	3,25,6,26	41	9
S_3	9,4,10,27,5,14	39	11
S_4	15,13,8	39	11
S_5	19,11,16,21	39	11
S_6	18,17,22	40	10
S_7	20,23	41	9
S_8	24,28,29	41	9
<p>เวลาว่างงานรวม = 76 นาที ค่าความแปรปรวนของภาระงาน = 2.50 ประสิทธิภาพของสายงานการประกอบ = 92.05%</p>			

5.5.2 ผลการทดสอบปัญหา Gunther

ผลการทดสอบปัญหา Gunther ด้วยวิธี HTS ที่พัฒนาขึ้นแสดงดังตารางที่ 5.5 จากตารางพบว่า กระบวนการค้นหาของ HTS สามารถหาคำตอบของปัญหา Gunther ได้อย่างมีประสิทธิภาพ และเมื่อค่าใกล้เคียง (N) มีจำนวนมากขึ้นจะทำให้อัตราการรู้เข้าหาคำตอบของวิธี HTS มีความแปรปรวนน้อยลง อย่างไรก็ตามเมื่อ N มีจำนวนมากจะทำให้เวลาที่ใช้ในการค้นหาเพิ่มมากขึ้น

จากตารางที่ 5.5 ค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุดที่พบเกิดจากการปรับค่า R และ N หลายค่า จึงทำการพิจารณาจากค่าความแปรปรวนของภาระงานเฉลี่ยที่มีค่าน้อยที่สุด ซึ่งจากตารางที่ 5.5 พบว่าค่าความแปรปรวนของภาระงานเฉลี่ยที่มีค่าน้อยที่สุด ($\text{mean-}w_v = 35.25$) จะเกิดจากการปรับ $R = 20\%$ ของปริภูมิการค้นหา และ $N = 45$ ตัว คำตอบของปัญหา Gunther ที่เหมาะสมที่ได้จากวิธี HTS แสดงในตารางที่ 5.6 รูปที่ 5.20 ถึง 5.28 แสดงอัตราการลู่เข้าหาคำตอบของปัญหา Gunther เมื่อปรับ $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา และ $N = 30, 45,$ และ 60 ตัว ตามลำดับ และรูปที่ 5.29 แสดงอัตราการลู่เข้าหาคำตอบของปัญหา Gunther ที่ให้ผลการทดสอบที่ดีที่สุด ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 30.21, เวลาว่างงานรวม = 117 นาที, และค่าประสิทธิภาพของสายงานการประกอบ = 80.50%

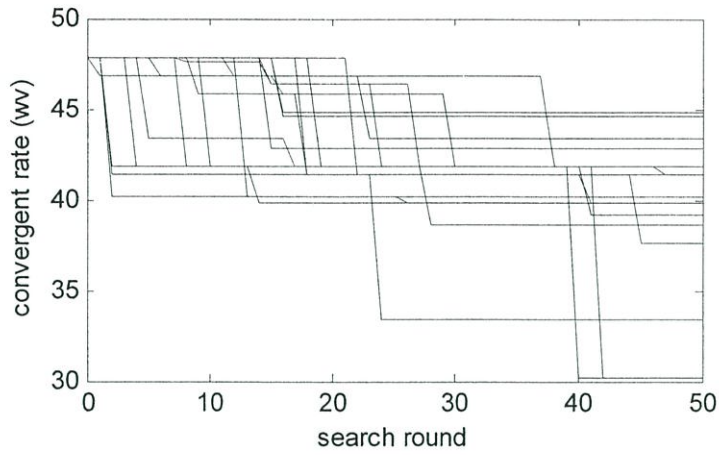
จากปัญหา Gunther เมื่อทำการหาคำตอบโดยวิธี COMSOAL ผลที่ได้แสดงดังตารางที่ 5.6 ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 47.81, เวลาว่างงานรวม = 117 นาที, และค่าประสิทธิภาพของสายงานการประกอบ = 80.50% เมื่อทำการเปรียบเทียบผลการทดสอบการแก้ปัญหา Gunther ด้วยวิธี COMSOAL และวิธี HTS ที่นำเสนอพบว่าวิธี HTS สามารถค้นหาคำตอบของปัญหา Gunther ได้ดีกว่าวิธี COMSOAL โดยที่ค่าความแปรปรวนของภาระงานที่ได้จากวิธี HTS มีค่าลดลง 36.81% (ลดลงจาก 47.81 มาเป็น 30.21) เมื่อเทียบกับวิธี COMSOAL และค่าประสิทธิภาพของสายงานการประกอบที่ได้จากวิธี COMSOAL และวิธี HTS มีค่าเท่ากัน

ตารางที่ 5.5 ผลการทดสอบการแก้ปัญหา Gunther ด้วยวิธี HTS

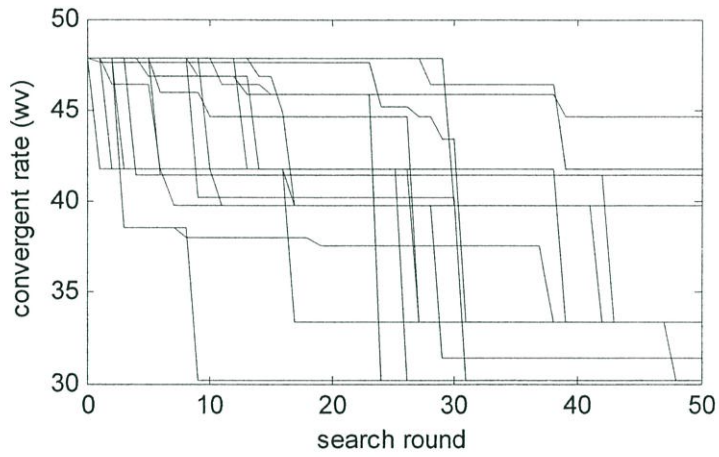
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 20% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	30.2100	44.8100	39.7300	4.1236	121.1756
45	30.2100	44.6100	35.2500	4.7814	206.5716
60	31.4100	41.8100	36.7200	3.9585	283.5734
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 40% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	30.2100	45.8100	38.5200	4.6991	138.7023
45	30.2100	41.8100	37.3600	4.0792	204.9827
60	30.2100	41.8100	36.2000	4.1302	271.5772
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 60% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	30.2100	45.8100	37.5200	4.3801	141.6688
45	30.2100	44.6100	38.3900	4.1997	213.1652
60	31.4100	41.8100	36.2300	3.2657	247.7094

ตารางที่ 5.6 ผลการเปรียบเทียบการแก้ปัญหา Gunther ด้วยวิธี COMSOAL และวิธี HTS

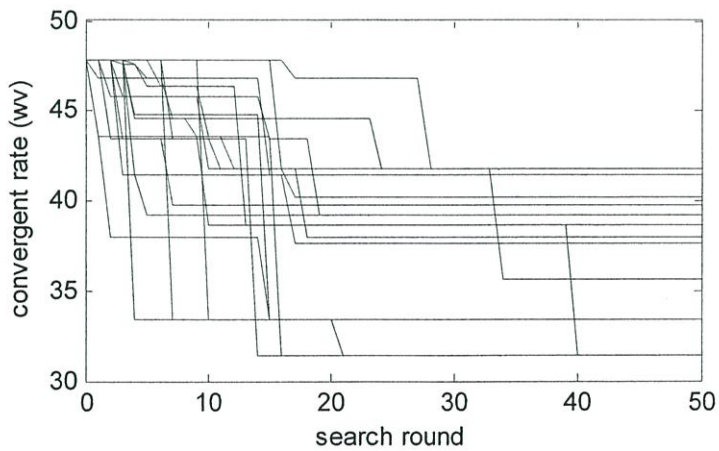
ผลที่ได้จากวิธี COMSOAL			
สถานีงาน	ชิ้นงาน	เวลาดำเนินงาน ของสถานีงาน (นาที)	เวลาว่างงาน (นาที)
S_1	1,17,2,5	40	20
S_2	10,12	60	0
S_3	3,6,4,7,8,14,18	52	8
S_4	9,11	45	15
S_5	13,15	42	18
S_6	19,16	48	12
S_7	20,21,22,25,30	55	5
S_8	23,26,31,24,27,32,34	57	3
S_9	28	40	20
S_{10}	33,29,35	44	16
<p>เวลาว่างงานรวม = 117 นาที ค่าความแปรปรวนของภาระงาน = 47.81 ประสิทธิภาพของสายงานการประกอบ = 80.50%</p>			
ผลที่ได้จากวิธี HTS			
สถานีงาน	ชิ้นงาน	เวลาดำเนินงาน ของสถานีงาน (นาที)	เวลาว่างงาน (นาที)
S_1	1,17,5,2	40	20
S_2	10,12	60	0
S_3	6,3,4,7,8,18	50	10
S_4	9,11	45	15
S_5	14,19,13	44	16
S_6	15,20	48	12
S_7	16,21,30,22	50	10
S_8	25,26,23,31,24,32	55	5
S_9	27,28	45	15
S_{10}	34,33,29,35	46	14
<p>เวลาว่างงานรวม = 117 นาที ค่าความแปรปรวนของภาระงาน = 30.21 ประสิทธิภาพของสายงานการประกอบ = 80.50%</p>			



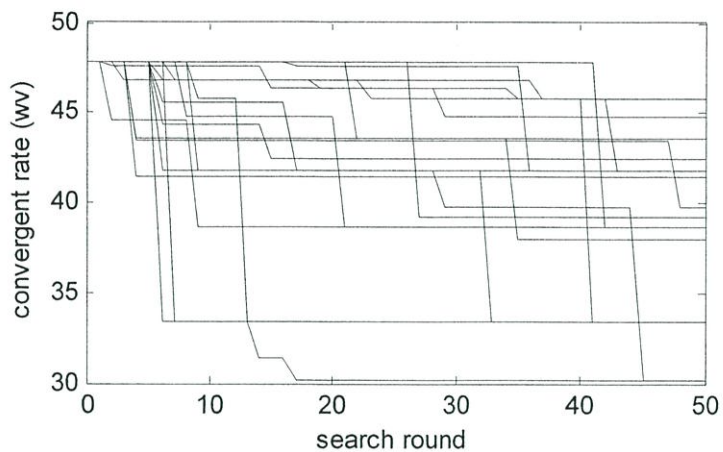
รูปที่ 5.20 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 20\%$ และ $N = 30$



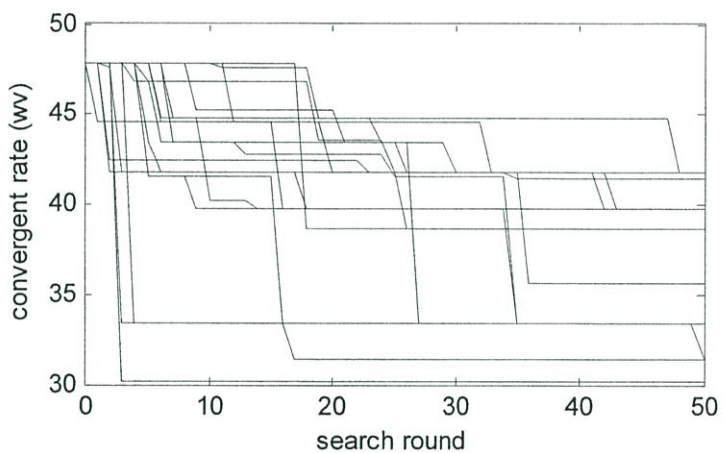
รูปที่ 5.21 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 20\%$ และ $N = 45$



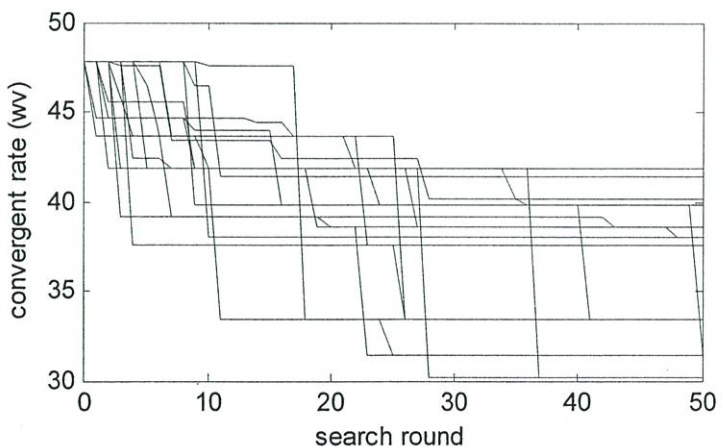
รูปที่ 5.22 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 20\%$ และ $N = 60$



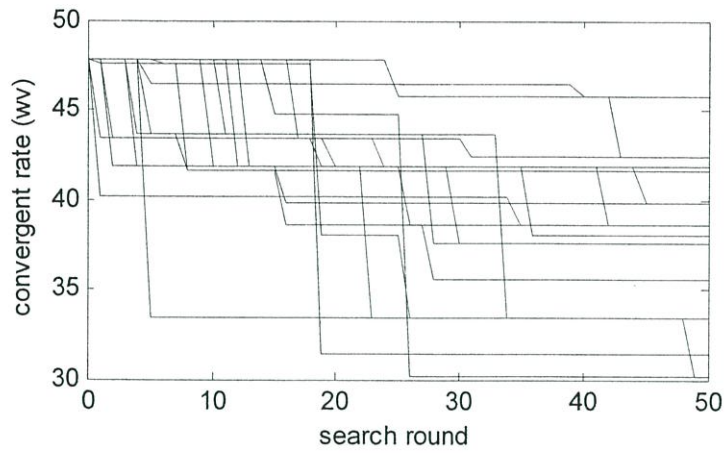
รูปที่ 5.23 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 40\%$ และ $N = 30$



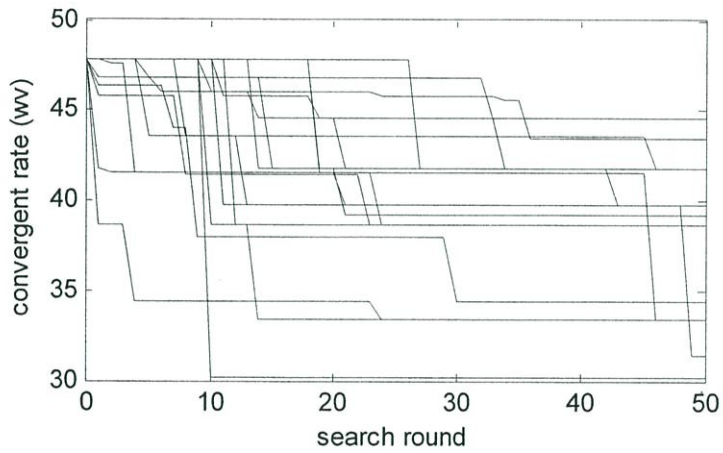
รูปที่ 5.24 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 40\%$ และ $N = 45$



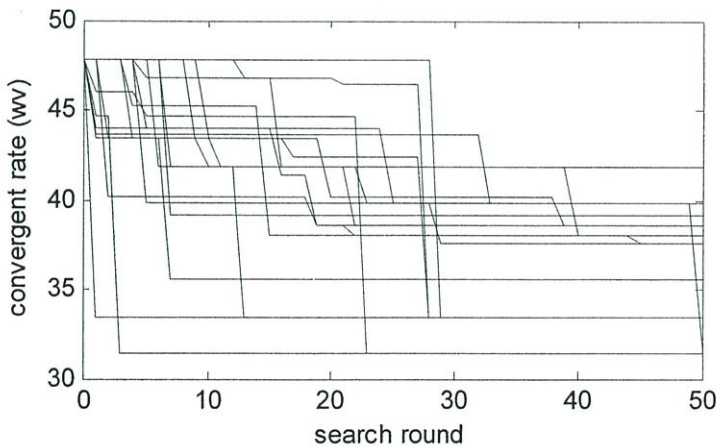
รูปที่ 5.25 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 40\%$ และ $N = 60$



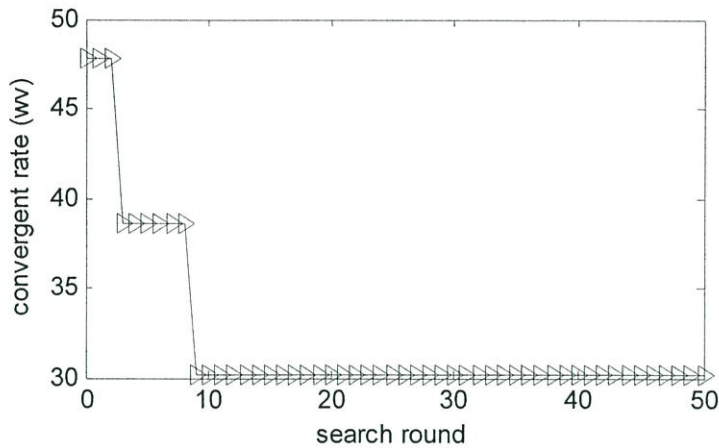
รูปที่ 5.26 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 60\%$ และ $N = 30$



รูปที่ 5.27 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 60\%$ และ $N = 45$



รูปที่ 5.28 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther เมื่อ $R = 60\%$ และ $N = 60$



รูปที่ 5.29 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Gunther ที่ให้ผลดีที่สุด

5.5.3 ผลการทดสอบปัญหา Random

ผลการทดสอบปัญหา Random ด้วยวิธี HTS ที่พัฒนาขึ้นแสดงดังตารางที่ 5.7 จากตารางพบว่า กระบวนการค้นหาของ HTS สามารถหาคำตอบของปัญหา Random ได้อย่างมีประสิทธิภาพ กล่าวคือ เมื่อปรับให้รัศมีการค้นหา $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา ค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุด ค่าความแปรปรวนของภาระงานที่มีค่ามากที่สุด ค่าความแปรปรวนของภาระงานเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของค่าความแปรปรวนของภาระงานมีค่าใกล้เคียงกัน และเมื่อค่าใกล้เคียง (N) มีจำนวนมากขึ้นจะทำให้อัตราการลู่เข้าหาคำตอบของวิธี HTS มีความแปรปรวนน้อยลง อย่างไรก็ตามเมื่อ N มีจำนวนมากจะทำให้เวลาที่ใช้ในการค้นหามีค่าเพิ่มมากขึ้น

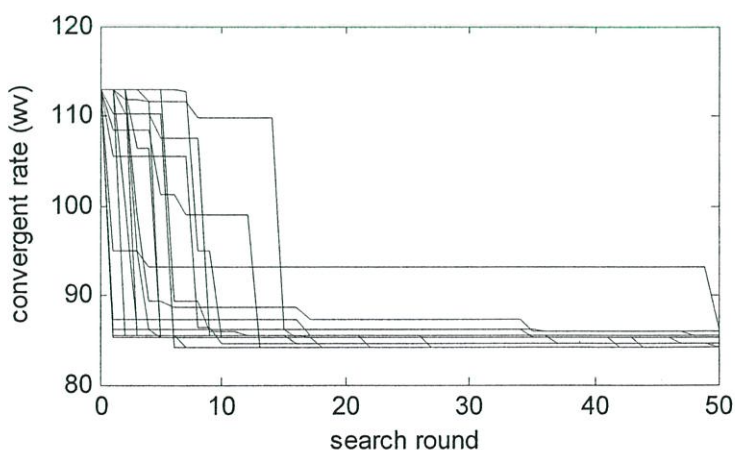
จากตารางที่ 5.7 ค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุดที่พบเกิดจากการปรับค่า R และ N หลายค่าจึงทำการพิจารณาจากค่าความแปรปรวนของภาระงานเฉลี่ยที่มีค่าน้อยที่สุดซึ่งจากตารางที่ 5.7 พบว่าค่าความแปรปรวนของภาระงานเฉลี่ยที่มีค่าน้อยที่สุด ($\text{mean-}w_v = 84.40$) จะเกิดจากการปรับ $R = 20\%$ ของปริภูมิการค้นหา และ $N = 60$ ตัว คำตอบของปัญหา Random ที่เหมาะสมที่ได้จากวิธี HTS แสดงในตารางที่ 5.8 รูปที่ 5.30 ถึง 5.38 แสดงอัตราการลู่เข้าหาคำตอบของปัญหา Random เมื่อปรับ $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา และ $N = 30, 45,$ และ 60 ตัว ตามลำดับ และรูปที่ 5.39 แสดงอัตราการลู่เข้าหาคำตอบของปัญหา Random ที่ให้ผลการทดสอบดีที่สุด ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 84.25, เวลาว่างงานรวม = 88 นาที, และค่าประสิทธิภาพของสายงานการประกอบ = 89.61%

จากปัญหา Random เมื่อทำการหาคำตอบโดยวิธี COMSOAL ผลที่ได้แสดงดังตารางที่ 5.8 ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 113.00, เวลาว่างงานรวม = 88 นาที, และค่าประสิทธิภาพของสายงานการประกอบ = 89.61% เมื่อทำการเปรียบเทียบผลการทดสอบการแก้ปัญหา Random

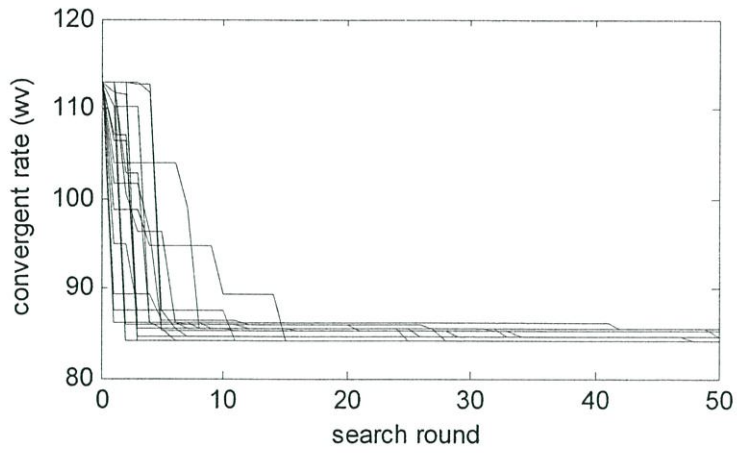
ด้วยวิธี COMSOAL และวิธี HTS ที่นำเสนอพบว่าวิธี HTS สามารถค้นหาคำตอบของปัญหา Random ได้ดีกว่าวิธี COMSOAL โดยที่ค่าความแปรปรวนของภาระงานที่ได้จากวิธี HTS มีค่าลดลง 25.44% (ลดลงจาก 113.00 มาเป็น 84.25) เมื่อเทียบกับวิธี COMSOAL และค่าประสิทธิภาพของสายงานการประกอบที่ได้จากวิธี COMSOAL และวิธี HTS มีค่าเท่ากัน

ตารางที่ 5.7 ผลการทดสอบการแก้ปัญหา Random ด้วยวิธี HTS

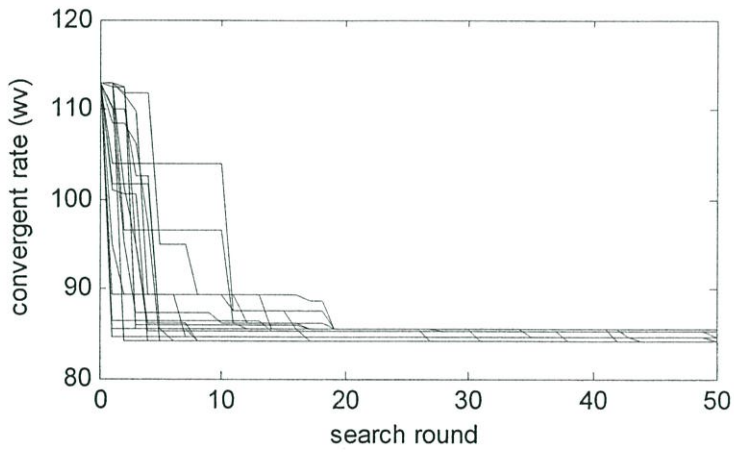
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 20% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	84.2500	86.2500	84.8125	0.6731	196.3767
45	84.2500	85.5000	84.6625	0.4952	300.5071
60	84.2500	85.5000	84.4000	0.3479	393.1483
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 40% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	84.2500	86.2500	84.7875	0.6452	195.2774
45	84.2500	85.5000	84.6000	0.4323	292.8819
60	84.2500	85.5000	84.5250	0.4792	398.3993
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 60% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	84.2500	85.5000	84.7250	0.5435	179.3656
45	84.2500	85.5000	84.7750	0.5555	303.2687
60	84.2500	85.5000	84.5250	0.4509	413.2844



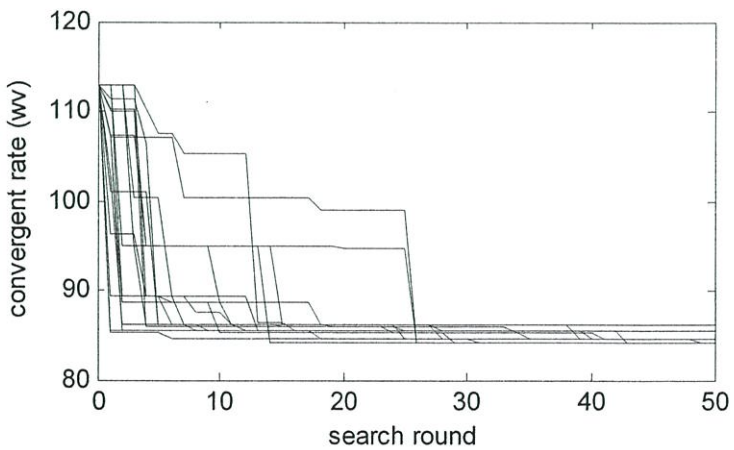
รูปที่ 5.30 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 20\%$ และ $N = 30$



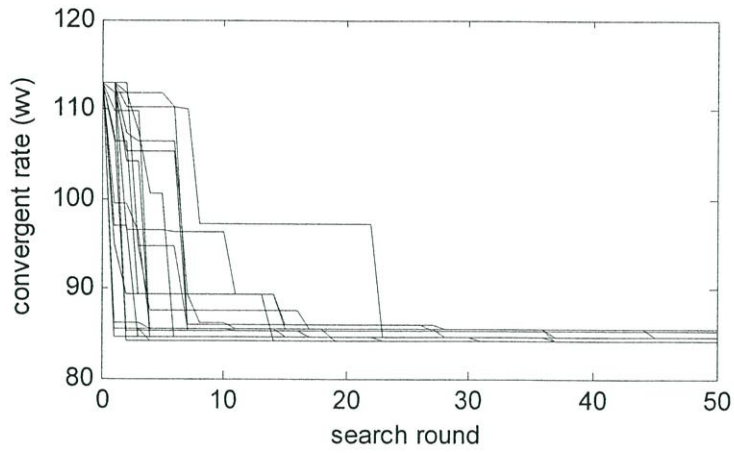
รูปที่ 5.31 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 20\%$ และ $N = 45$



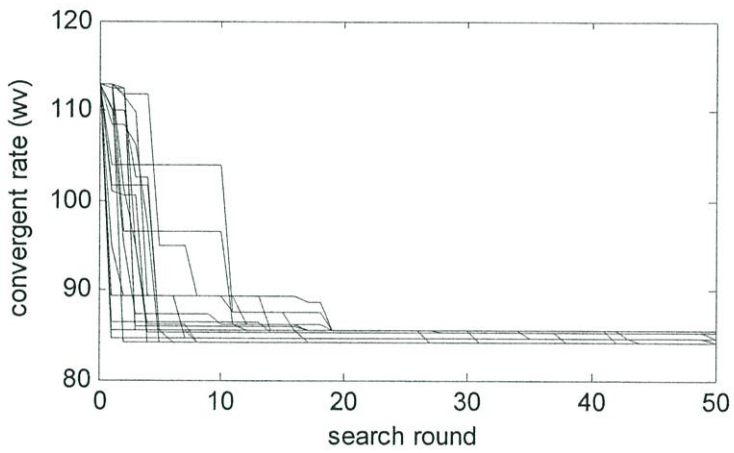
รูปที่ 5.32 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 20\%$ และ $N = 60$



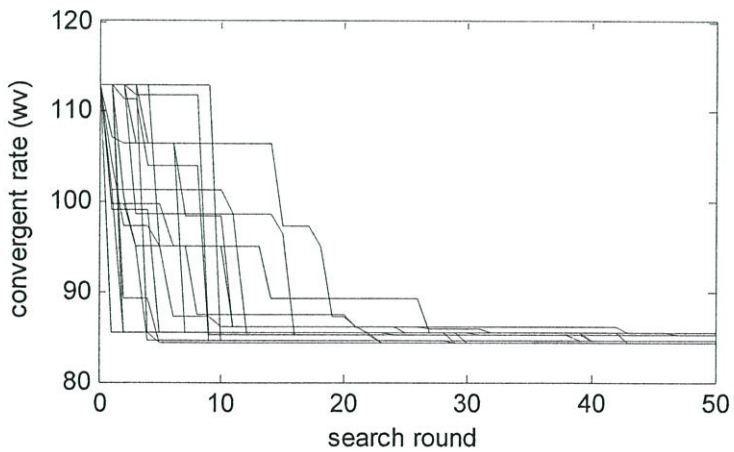
รูปที่ 5.33 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 40\%$ และ $N = 30$



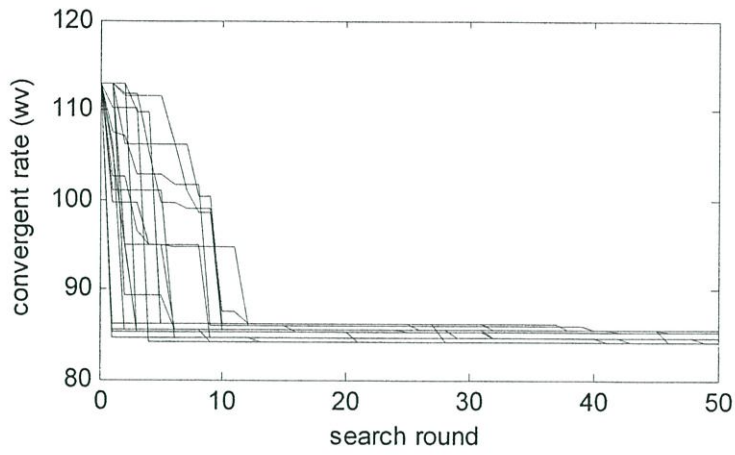
รูปที่ 5.34 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 40\%$ และ $N = 45$



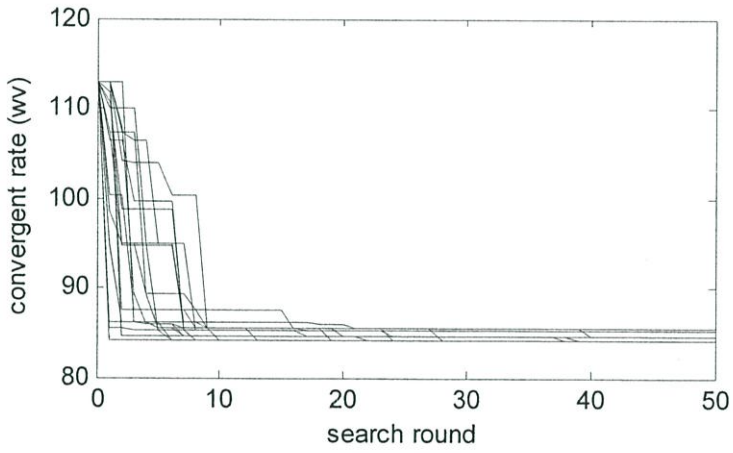
รูปที่ 5.35 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 40\%$ และ $N = 60$



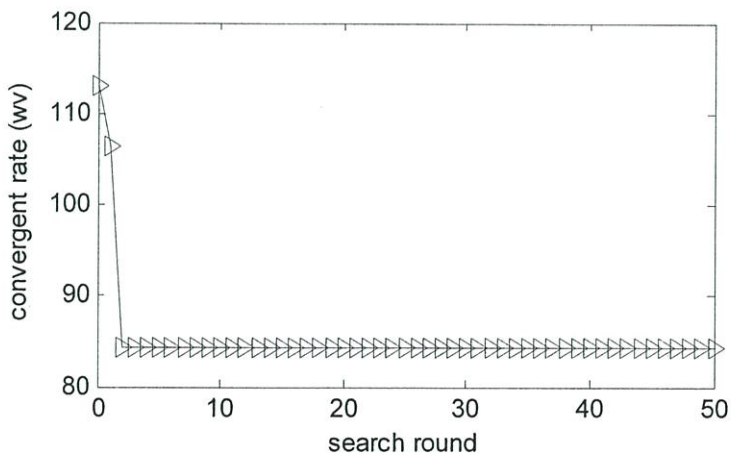
รูปที่ 5.36 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 60\%$ และ $N = 30$



รูปที่ 5.37 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 60\%$ และ $N = 45$



รูปที่ 5.38 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random เมื่อ $R = 60\%$ และ $N = 60$



รูปที่ 5.39 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา Random ที่ให้ผลดีที่สุด

ตารางที่ 5.8 ผลการเปรียบเทียบการแก้ปัญหา Random ด้วยวิธี COMSOAL และวิธี HTS

ผลที่ได้จากวิธี COMSOAL			
สถานีงาน	ชั้นงาน	เวลาดำเนินงาน ของสถานีงาน (นาที)	เวลาว่างงาน (นาที)
S_1	1,2,12,39,3,4,37,7	71	9
S_2	8,5,6,11,43,13	69	11
S_3	9,10,14,15	73	7
S_4	16,17,18,23,29,30,31	77	3
S_5	24,25,32,19,26,27	77	3
S_6	20,33,21,34	77	3
S_7	35,36,22,28,38,40	65	15
S_8	41,42,44,45	43	37
เวลาว่างงานรวม = 88 นาที ค่าความแปรปรวนของภาระงาน = 113.00 ประสิทธิภาพของสายงานการประกอบ = 89.61%			
ผลที่ได้จากวิธี HTS			
สถานีงาน	ชั้นงาน	เวลาดำเนินงาน ของสถานีงาน (นาที)	เวลาว่างงาน (นาที)
S_1	39,1,2,11,12,37,4,7	71	9
S_2	8,3,5,6,43,13	69	11
S_3	10,9,15,14,29	77	3
S_4	31,30,24,17,18,16	75	5
S_5	23,25,32,19,26,27	75	5
S_6	33,20,21	74	6
S_7	35,34,36,22,28,38	64	16
S_8	40,41,42,45,44	47	33
เวลาว่างงานรวม = 88 นาที ค่าความแปรปรวนของภาระงาน = 84.25 ประสิทธิภาพของสายงานการประกอบ = 89.61%			

5.5.4 ผลการทดสอบปัญหา M-cycle

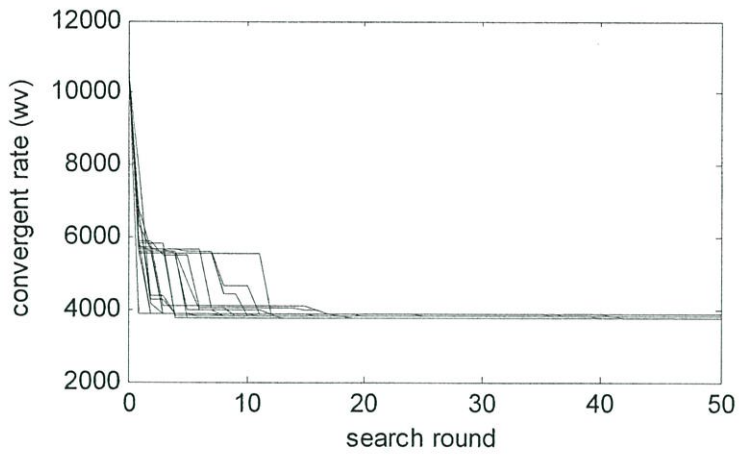
ผลการทดสอบปัญหา M-cycle ด้วยวิธี HTS ที่พัฒนาขึ้นแสดงดังตารางที่ 5.9 จากตารางพบว่า กระบวนการค้นหาของ HTS สามารถหาคำตอบของปัญหา M-cycle ได้อย่างมีประสิทธิภาพ และเมื่อค่าใกล้เคียง (N) มีจำนวนมากขึ้นจะทำให้อัตราการรู้เข้าหาคำตอบของวิธี HTS มีความแปรปรวนน้อยลง แต่จะทำให้เวลาที่ใช้ในการค้นหามีค่าเพิ่มมากขึ้น

จากตารางที่ 5.9 ค่าความแปรปรวนของภาระงานเฉลี่ยที่มีค่าน้อยที่สุด ($\text{mean-}w_v = 3.7936 \times 10^3$) เกิดจากการปรับ $R = 60\%$ ของปริภูมิการค้นหา และ $N = 60$ ตัว คำตอบของปัญหา M-cycle ที่เหมาะสมที่ได้จากวิธี HTS แสดงในตารางที่ 5.10 รูปที่ 5.40 ถึง 5.48 แสดงอัตราการลู่เข้าหาคำตอบของปัญหา M-cycle เมื่อปรับ $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา และ $N = 30, 45,$ และ 60 ตัว ตามลำดับ และรูปที่ 5.49 แสดงอัตราการลู่เข้าหาคำตอบของปัญหา M-cycle ที่ให้ผลการทดสอบดีที่สุด ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 3,784.50, เวลาว่างงานรวม = 405 วินาที, และค่าประสิทธิภาพของสายงานการประกอบ = 90.20%

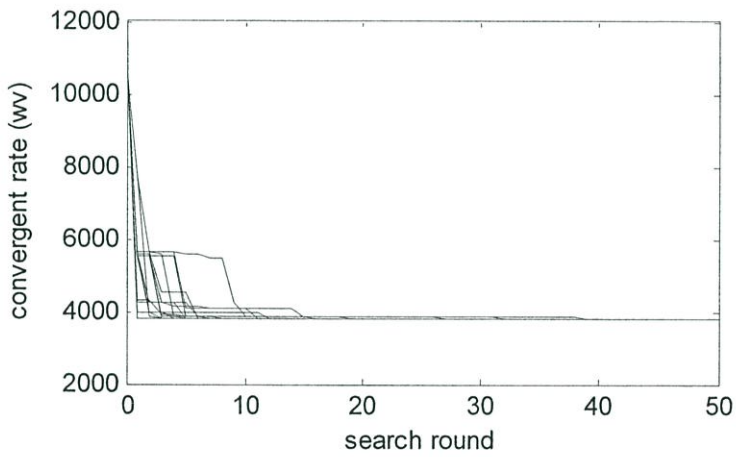
จากปัญหา M-cycle เมื่อทำการหาคำตอบโดยวิธี COMSOAL ผลที่ได้แสดงดังตารางที่ 5.10 ซึ่งพบว่าค่าความแปรปรวนของภาระงาน = 10,474, เวลาว่างงานรวม = 405 วินาที, และค่าประสิทธิภาพของสายงานการประกอบ = 85.94% เมื่อทำการเปรียบเทียบผลการทดสอบการแก้ปัญหา M-cycle ด้วยวิธี COMSOAL และวิธี HTS ที่นำเสนอพบว่าวิธี HTS สามารถค้นหาคำตอบของปัญหา M-cycle ได้ดีกว่าวิธี COMSOAL โดยที่ค่าความแปรปรวนของภาระงานที่ได้จากวิธี HTS มีค่าลดลง 63.87% (ลดลงจาก 10,474.00 มาเป็น 3,784.50) เมื่อเทียบกับวิธี COMSOAL และค่าประสิทธิภาพของสายงานการประกอบที่ได้จากวิธี HTS มีค่าเพิ่มขึ้น 4.96% เมื่อเทียบกับวิธี COMSOAL

ตารางที่ 5.9 ผลการทดสอบการแก้ปัญหา M-cycle ด้วยวิธี HTS

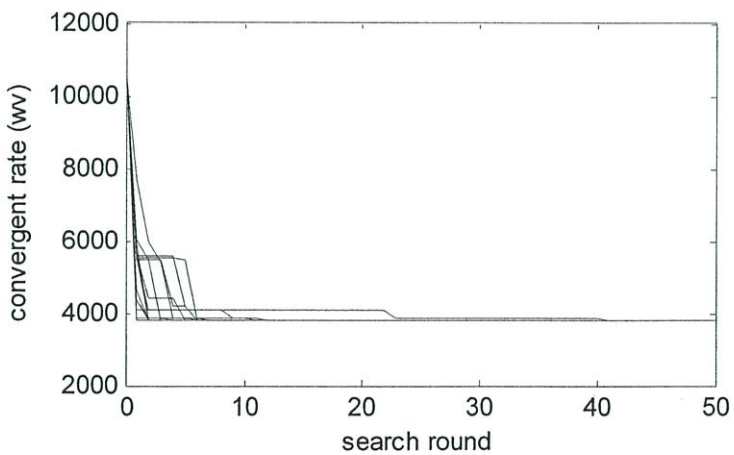
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 20% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	3.7845×10^3	3.8645×10^3	3.7995×10^3	17.4919	187.3336
45	3.7845×10^3	3.8135×10^3	3.7982×10^3	8.3183	293.1686
60	3.7845×10^3	3.8132×10^3	3.7919×10^3	8.9153	400.5328
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 40% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	3.7845×10^3	3.8322×10^3	3.8027×10^3	14.3137	175.1412
45	3.7845×10^3	3.8255×10^3	3.7963×10^3	10.7423	292.8625
60	3.7845×10^3	3.8135×10^3	3.7955×10^3	8.0580	385.4875
จำนวนค่าใกล้เคียง (N)	รัศมีการค้นหา (R) = 60% ของปริภูมิการค้นหา				
	min- w_v	max- w_v	mean- w_v	std- w_v	average search time (sec)
30	3.7845×10^3	3.8095×10^3	3.7952×10^3	8.4727	197.7978
45	3.7845×10^3	3.8320×10^3	3.7968×10^3	11.9457	250.2772
60	3.7845×10^3	3.8035×10^3	3.7936×10^3	7.7385	376.8358



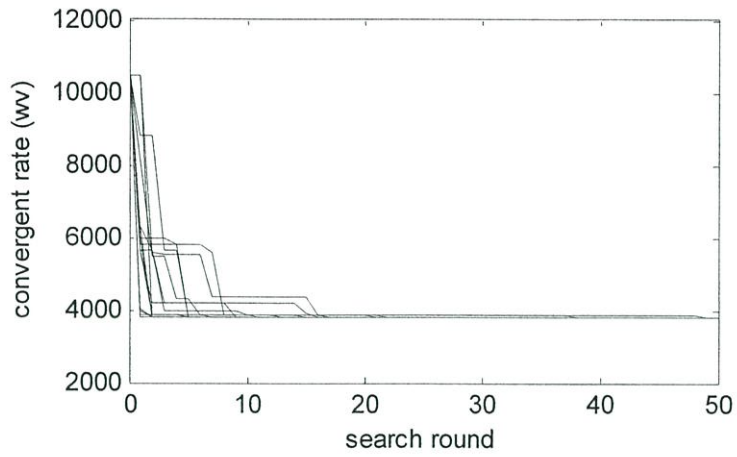
รูปที่ 5.40 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 20\%$ และ $N = 30$



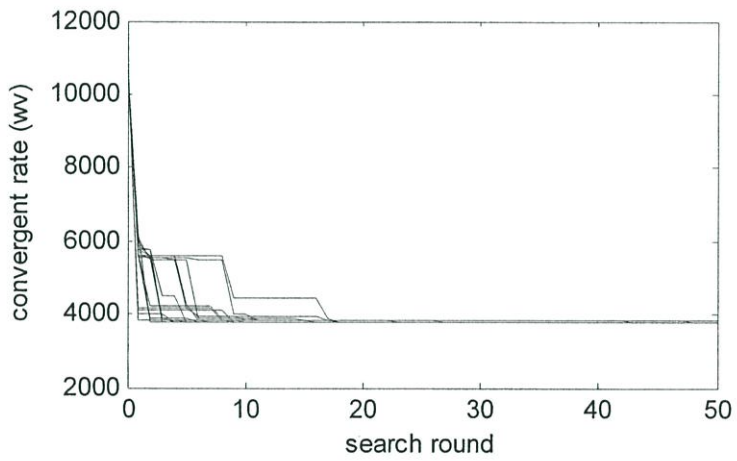
รูปที่ 5.41 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 20\%$ และ $N = 45$



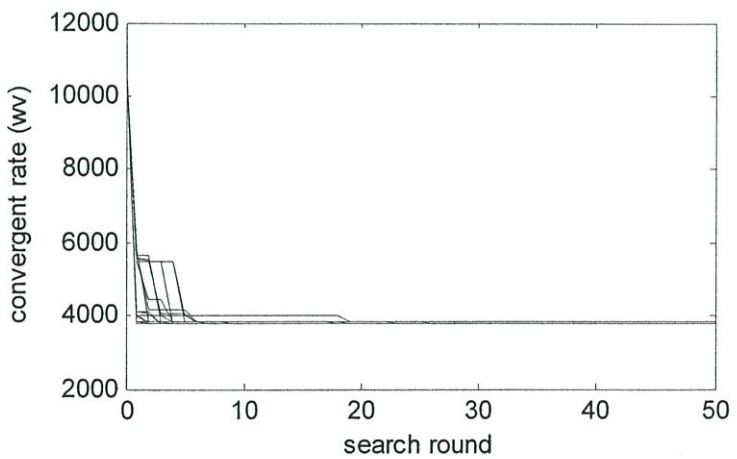
รูปที่ 5.42 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 20\%$ และ $N = 60$



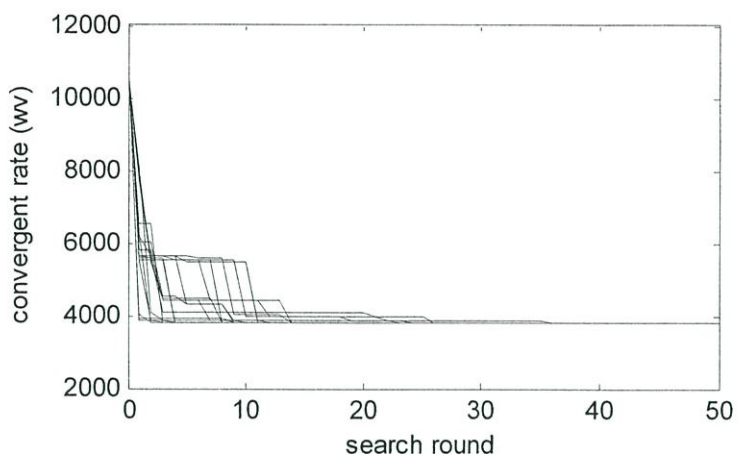
รูปที่ 5.43 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 40\%$ และ $N = 30$



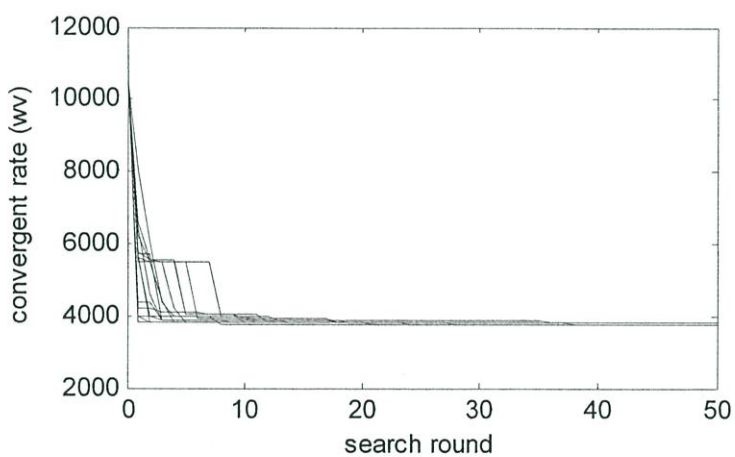
รูปที่ 5.44 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 40\%$ และ $N = 45$



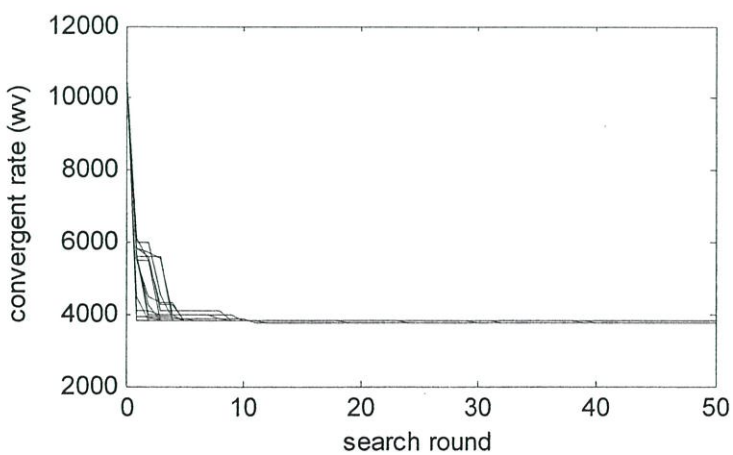
รูปที่ 5.45 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 40\%$ และ $N = 60$



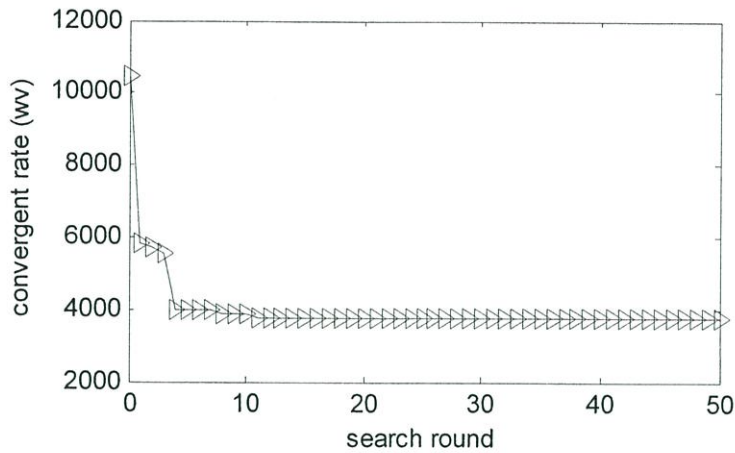
รูปที่ 5.46 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 60\%$ และ $N = 30$



รูปที่ 5.47 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 60\%$ และ $N = 45$



รูปที่ 5.48 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle เมื่อ $R = 60\%$ และ $N = 60$



รูปที่ 5.49 อัตราการลู่เข้าหาคำตอบของ HTS กับปัญหา M-cycle ที่ให้ผลดีที่สุด

5.6 สรุป

ในบทที่ 5 นี้ได้กล่าวถึงการทดสอบการแก้ปัญหา ALB โดยการค้นหาแบบตาบอดเชิงพันทาง หรือ HTS ที่ได้รับการพัฒนาขึ้น ปัญหา ALB แบบ SMD จำนวน 4 ปัญหาอันได้แก่ปัญหา Buxey ซึ่งมีจำนวนชั้นงานทั้งหมด 29 ชั้นงาน ปัญหา Gunther ซึ่งมีจำนวนชั้นงานทั้งหมด 35 ชั้นงาน ปัญหา Random ซึ่งมีจำนวนชั้นงานทั้งหมด 45 ชั้นงาน และปัญหา M-cycle ซึ่งมีจำนวนชั้นงานทั้งหมด 60 ชั้นงาน ได้รับการแก้ปัญหาโดยวิธี HTS ที่นำเสนอ

ขั้นตอนการทดสอบการแก้ปัญหา ALB ด้วยวิธี HTS ได้มีการกำหนดค่าพารามิเตอร์ที่เหมาะสม โดยกำหนดให้รัศมีการค้นหา $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา จำนวนสมาชิกค่าใกล้เคียง $N = 30, 45, 60$ และดำเนินการสลบที่แบบสุ่มจำนวน 1,000 ครั้ง การกำหนดปริภูมิการค้นหาของสถานีงานทำการกำหนดโดยอาศัยวิธีศึกษาสำนัก เกณฑ์ยุติการค้นหา (TC) ของวิธี HTS อาศัยจำนวนรอบการค้นหาสูงสุด ซึ่งกำหนดให้มีค่าเท่ากับ 50 รอบ ในการทดสอบการแก้ปัญหา ALB แต่ละปัญหา ได้ทำการทดสอบทั้งหมดจำนวน 20 ครั้ง เพื่อนำคำตอบที่ได้มาวิเคราะห์และเปรียบเทียบกับผลที่ได้จากวิธี COMSOAL

จากผลการทดสอบการกำหนดค่าพารามิเตอร์ของวิธี HTS ให้ $R = 20, 40,$ และ 60% ของปริภูมิการค้นหาและ $N = 30, 45, 60$ จากคำตอบที่ได้พบว่า ค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุด ค่าความแปรปรวนของภาระงานที่มีค่ามากที่สุด ค่าความแปรปรวนของภาระงานเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของค่าความแปรปรวนของภาระงานมีค่าใกล้เคียงกัน และเมื่อ N มีจำนวนมากขึ้นจะทำให้อัตราการลู่เข้าหาคำตอบของวิธี HTS มีความแปรปรวนน้อยลง อย่างไรก็ตาม เวลาที่ใช้ในการค้นหาคำตอบของวิธี HTS จะแปรผันตรงกับจำนวนของ N

ตารางที่ 5.10 ผลการเปรียบเทียบการแก้ปัญหา M-cycle ด้วยวิธี COMSOAL และวิธี HTS

ผลที่ได้จากวิธี COMSOAL			
สถานีงาน	ชั้นงาน	เวลาดำเนินงาน ของสถานีงาน (วินาที)	เวลาว่างงาน (วินาที)
S_1	1,2,49,5,55,3,4,51,52	352	8
S_2	53,54,6,7,20,21,29,34,50	347	13
S_3	8,9,36,10,11,12,15,17	360	0
S_4	13,14,16,19,38,40,18,42,31	359	1
S_5	22,56,24,57,23,26,28,30,58	354	6
S_6	25,32,59,27,44,60,33	333	27
S_7	35,37,39,41,43,45,46,47	330	30
S_8	48	40	320
เวลาว่างงานรวม = 405 วินาที ค่าความแปรปรวนของภาระงาน = 10,474.00 ประสิทธิภาพของสายงานการประกอบ = 85.94%			
ผลที่ได้จากวิธี HTS			
สถานีงาน	ชั้นงาน	เวลาดำเนินงาน ของสถานีงาน (วินาที)	เวลาว่างงาน (วินาที)
S_1	1,49,2,5,55,3,4,52,54	334	26
S_2	51,53,50,21,34,29,7,6	329	31
S_3	20,8,9,36,10,11,12	316	44
S_4	17,13,15,19,14,38,40,16,31	330	30
S_5	18,42,22,56,57,24,58,26,30	343	17
S_6	28,23,32,59,25,60,27	343	17
S_7	44,33,35,37,41,39,43	332	28
S_8	46,45,47,48	148	212
เวลาว่างงานรวม = 405 วินาที ค่าความแปรปรวนของภาระงาน = 3,784.50 ประสิทธิภาพของสายงานการประกอบ = 90.20%			

จากผลการทดสอบการแก้ปัญหา ALB ทั้ง 4 ปัญหาด้วยวิธี HTS ที่นำเสนอเปรียบเทียบกับผลที่ได้จากวิธี COMSOAL เมื่อพิจารณาจากค่าความแปรปรวนของภาระงาน พบว่าวิธี HTS สามารถค้นหาคำตอบของปัญหา ALB ทั้ง 4 ปัญหาได้ดีกว่าคำตอบที่ได้จากวิธี COMSOAL ซึ่งสามารถแสดงโดยสรุปได้ดังตารางที่ 5.11

ตารางที่ 5.11 สรุปผลการเปรียบเทียบการแก้ปัญหา ALB ด้วยวิธี COMSOAL และวิธี HTS

ปัญหา	ค่าความแปรปรวนของภาระงาน			ค่าประสิทธิภาพสายงานการประกอบ		
	COMSOAL	HTS	ค่าลดลง	COMSOAL	HTS	ค่าเพิ่มขึ้น
Buxey	76.50	2.50	96.73%	81.00%	92.05%	13.64%
Gunther	47.81	30.21	36.81%	80.50%	80.50%	-
Random	113.00	84.25	25.44%	89.61%	89.61%	-
M-cycle	10,474.00	3,784.50	63.87%	85.94%	90.20%	4.96%

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 ข้อสรุป

วิทยานิพนธ์นี้ได้นำเสนอวิธีการค้นหาแบบตามูเชิงพันทาง (hybrid tabu search: HTS) เพื่อแก้ปัญหาการจัดสมดุลสายงานการประกอบหรือปัญหา ALB แบบผลิตภัณฑ์เดียวที่มีการกำหนดเวลาดำเนินงานของชิ้นงานให้มีค่าคงที่ (SMD) โดยมีความมุ่งหมายและวัตถุประสงค์เพื่อพัฒนาอัลกอริทึมการค้นหาแบบตามูเชิงพันทางสำหรับแก้ปัญหาการจัดสมดุลสายงานการประกอบและเพื่อแก้ปัญหา ALB โดยใช้วิธี HTS หลักการของอัลกอริทึมที่นำเสนอคือการทำงานร่วมกันของการค้นหาแบบตามู (tabu search: TS) และเทคนิคการสลับที่แบบสุ่ม (random permutation technique) โดยที่ TS ทำหน้าที่จัดสรรจำนวนชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงาน ในขณะที่การสลับที่แบบสุ่มทำหน้าที่จัดลำดับของชิ้นงานที่เหมาะสมให้กับแต่ละสถานีงานโดยสอดคล้องกับเงื่อนไขลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน อัลกอริทึมของวิธี HTS ได้ถูกเขียนเป็นโปรแกรมการค้นหาโดย MATLAB ผลการทดสอบการแก้ปัญหา ALB แบบผลิตภัณฑ์เดียวจำนวน 4 ปัญหาที่ได้มาจากการสำรวจงานวิจัยที่เกี่ยวข้องด้วยวิธี HTS ที่นำเสนอถูกนำไปเปรียบเทียบกับวิธี COMSOAL ซึ่งเป็นวิธีการแก้ปัญหา ALB แบบแบบศึกษาสำนักวิชาการหนึ่ง ที่ได้รับความนิยมอย่างกว้างขวาง การดำเนินงานวิทยานิพนธ์ดังกล่าวสำเร็จลุล่วงตามความมุ่งหมายและวัตถุประสงค์ โดยสามารถสรุปผลการศึกษาวิจัยและพัฒนาทางวิศวกรรมเป็นข้อสรุปได้ดังต่อไปนี้

1. การปฏิสัมพันธ์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง ได้รับการรายงานไว้ในบทที่ 2 โดยให้ความสนใจเกี่ยวกับปัญหา ALB การค้นหาแบบปัญญาประดิษฐ์ โดยเน้นที่การค้นหาแบบตามูหรือ TS และการประยุกต์การค้นหาแบบปัญญาประดิษฐ์เพื่อแก้ปัญหา ALB ผลการปฏิสัมพันธ์วรรณกรรมทำให้ทราบหลักการและขั้นตอนการแก้ปัญหา ALB แบบดั้งเดิม ทราบถึงแนวคิด ประเภท เกณฑ์ วัตถุประสงค์ ขั้นตอนการแก้ปัญหา เทคนิควิธี และอัลกอริทึมต่างๆ ในการแก้ปัญหา ALB ทราบถึงหลักการและอัลกอริทึมของการค้นหาแบบตามู และยังพบว่าการประยุกต์เทคนิคการค้นหาแบบปัญญาประดิษฐ์เพื่อแก้ปัญหา ALB ส่วนใหญ่จะใช้จินเนติกอัลกอริทึมหรือ GA สำหรับการประยุกต์ใช้ TS ในการแก้ปัญหา ALB ยังไม่แพร่หลายมากนัก ส่วนการประยุกต์ใช้การค้นหาแบบมดและเครือข่ายประสาทเทียมเพื่อแก้ปัญหา ALB ยังคงได้รับความสนใจน้อย

2. รายละเอียดทฤษฎีการจัดสมดุลสายงานการประกอบได้รับการศึกษาและเรียบเรียงไว้ในบทที่ 3 ซึ่งได้รวบรวมคำจำกัดความของคำศัพท์ต่างๆ ที่ใช้เกี่ยวกับการจัดสมดุลสายงานการประกอบ กล่าวถึงปัญหาการจัดสมดุลสายงานการประกอบหรือปัญหา ALB ที่ใช้ในวิทยานิพนธ์

ซึ่งเป็นปัญหา ALB แบบ SMD การคำนวณสถานีงาน เวลาว่างงาน ความแปรปรวนของภาระงาน และประสิทธิภาพของสายงานการประกอบ นอกจากนี้ยังได้ให้รายละเอียดขั้นตอนของวิธีการแก้ปัญหา ALB ด้วยวิธี COMSOAL

3. อัลกอริทึมของการค้นหาแบบตาบอดหรือ TS และการพัฒนาอัลกอริทึมของการค้นหาแบบตาบอดเชิงพันทางหรือ HTS เพื่อแก้ปัญหา ALB ได้รับการนำเสนอในบทที่ 4 อัลกอริทึมของ TS และ HTS ถูกพัฒนาเป็นโปรแกรม TS และโปรแกรม HTS โดยอาศัยโปรแกรม MATLAB ดังรายละเอียดที่แสดงในภาคผนวก ข. และ ค. ตามลำดับ

4. การทดสอบสมรรถนะการค้นหาคำตอบของโปรแกรม TS ที่พัฒนาขึ้นได้รับการนำเสนอในบทที่ 4 การทดสอบดังกล่าวเป็นการทดสอบเพื่อหาค่าต่ำสุดของฟังก์ชัน โบฮาเซฟสกี ซึ่งเป็นฟังก์ชันที่มีคำตอบวงแคบเฉพาะถิ่นจำนวนมาก แต่มีคำตอบวงกว้างเพียงคำตอบเดียว จากผลการทดสอบโปรแกรม TS พบว่าสามารถค้นหาคำตอบ (ใกล้) วงกว้างของฟังก์ชัน โบฮาเซฟสกีได้อย่างมีประสิทธิภาพ

5. การทดสอบสมรรถนะการแก้ปัญหา ALB ของโปรแกรม HTS ที่พัฒนาขึ้นได้รับการนำเสนอในบทที่ 4 การทดสอบดังกล่าวอาศัยปัญหา ALB อย่างง่ายที่มีชั้นงานจำนวน 11 ชั้นงาน จากผลการทดสอบโปรแกรม HTS พบว่าสามารถค้นหาคำตอบที่เหมาะสมของปัญหา ALB ได้อย่างน่าพึงพอใจ

6. ปัญหา ALB แบบผลิตภัณฑ์เดียวจำนวน 4 ปัญหาที่ใช้ทดสอบวิธี HTS ที่พัฒนาขึ้นในวิทยานิพนธ์ได้รับการนำเสนอในบทที่ 5 และภาคผนวก ก. ซึ่งประกอบด้วยปัญหา Buxey ซึ่งมีจำนวนชั้นงานทั้งหมด 29 ชั้นงาน ปัญหา Gunther ซึ่งมีจำนวนชั้นงานทั้งหมด 35 ชั้นงาน ปัญหา Random ซึ่งมีจำนวนชั้นงานทั้งหมด 45 ชั้นงาน และปัญหาสุดท้ายคือปัญหาสายงานการประกอบมอเตอร์ไซค์ หรือปัญหา M-cycle ซึ่งเป็นปัญหาการจัดสมดุลสายงานการประกอบจริงในภาคอุตสาหกรรม โดยใช้รายละเอียดของสายงานการประกอบมอเตอร์ไซค์ของบริษัท คาวาซากิ มอเตอร์ เอ็นเตอร์ไพรส์ ประเทศไทย จำกัด ซึ่งมีจำนวนชั้นงานทั้งหมด 60 ชั้นงาน

7. การทดสอบการแก้ปัญหา ALB แบบผลิตภัณฑ์เดียวได้รับการนำเสนอในบทที่ 5 ซึ่งจากการทดสอบสมรรถนะการค้นหาคำตอบของวิธี HTS เมื่อกำหนดครีสมิการค้นหา $R = 20, 40,$ และ 60% ของปริภูมิการค้นหา กำหนดค่าใกล้เคียง $N = 30, 45, 60$ พบว่าค่าความแปรปรวนของภาระงานที่มีค่าน้อยที่สุด ค่าความแปรปรวนของภาระงานที่มีค่ามากที่สุด ค่าความแปรปรวนของภาระงานเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของค่าความแปรปรวนของภาระงานมีค่าใกล้เคียงกัน และเมื่อ N มีจำนวนมากขึ้นจะทำให้อัตราการลู่เข้าหาคำตอบของวิธี HTS มีความแปรปรวนน้อยลง แต่เวลาที่ใช้ในการค้นหาคำตอบของวิธี HTS จะเพิ่มมากขึ้น

8. การทดสอบแก้ปัญหา ALB ทั้ง 4 ปัญหาในหัวข้อที่ 6 ด้วยวิธี HTS ที่นำเสนอเปรียบเทียบกับผลที่ได้จากวิธี COMSOAL ได้รับการนำเสนอในบทที่ 5 ซึ่งจากการทดสอบ เมื่อพิจารณาจาก

ค่าความแปรปรวนของภาระงาน พบว่าวิธี HTS สามารถค้นหาคำตอบของปัญหา ALB ทั้ง 4 ปัญหาได้อย่างน่าพึงพอใจยิ่ง และให้ผลที่ดีกว่าวิธี COMSOAL ในทุกกรณีปัญหาดังสรุปได้ดังนี้

- ปัญหา Buxey พบว่าค่าความแปรปรวนของภาระงานที่ได้จากวิธี HTS มีค่าลดลง 96.73% เมื่อเทียบกับวิธี COMSOAL และค่าประสิทธิภาพของสายงานการประกอบที่ได้จากวิธี HTS มีค่าเพิ่มขึ้น 13.64% เมื่อเทียบกับวิธี COMSOAL

- ปัญหา Gunther พบว่าค่าความแปรปรวนของภาระงานที่ได้จากวิธี HTS มีค่าลดลง 36.81% เมื่อเทียบกับวิธี COMSOAL และค่าประสิทธิภาพของสายงานการประกอบที่ได้จากวิธี COMSOAL และวิธี HTS มีค่าเท่ากัน

- ปัญหา Random พบว่าค่าความแปรปรวนของภาระงานที่ได้จากวิธี HTS มีค่าลดลง 25.44% เมื่อเทียบกับวิธี COMSOAL และค่าประสิทธิภาพของสายงานการประกอบที่ได้จากวิธี COMSOAL และวิธี HTS มีค่าเท่ากัน

- ปัญหา M-cycle พบว่าค่าความแปรปรวนของภาระงานที่ได้จากวิธี HTS มีค่าลดลง 63.87% เมื่อเทียบกับวิธี COMSOAL และค่าประสิทธิภาพของสายงานการประกอบที่ได้จากวิธี HTS มีค่าเพิ่มขึ้น 4.96% เมื่อเทียบกับวิธี COMSOAL

จากผลการทดสอบการแก้ปัญหา ALB ทั้ง 4 ปัญหาสามารถสรุปได้ว่าวิธี HTS ที่นำเสนอสามารถค้นหาคำตอบของปัญหา ALB ได้ดีกว่าวิธี COMSOAL โดยสามารถทำให้ค่าความแปรปรวนของภาระงานมีค่าลดลงสูงสุดถึง 96.73%

9. จากผลการดำเนินงานวิทยานิพนธ์ที่ผ่านมา ก่อให้เกิดผลงานวิจัยตีพิมพ์และเผยแพร่ในการประชุมวิชาการระดับชาติจำนวน 1 บทความ และการประชุมวิชาการระดับนานาชาติจำนวน 2 บทความ ดังรายละเอียดที่ปรากฏในภาคผนวก ง.

6.2 อภิปรายผล

จากผลการทดสอบที่ผ่านมาพบว่า การแก้ปัญหา ALB โดยวิธี HTS ที่นำเสนอให้ค่าความแปรปรวนของภาระงานลดลงไม่เท่ากันในแต่ละกรณีปัญหา ทั้งนี้เนื่องจากปัญหา ALB แต่ละปัญหาจะมีเงื่อนไขที่เกี่ยวข้องกับลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน เวลาการทำงานของแต่ละชิ้นงาน และรอบเวลาการผลิตที่กำหนดให้ จากผลการทดสอบปัญหา ALB ทั้ง 4 ปัญหาพบว่า ปัญหา Buxey สามารถลดค่าความแปรปรวนของภาระงานได้สูงสุด เนื่องจากชิ้นงานส่วนใหญ่มีเวลาการทำงานน้อยกว่ารอบเวลาการผลิต อีกทั้งเมื่อพิจารณาจากลำดับความสัมพันธ์ก่อนหลังของชิ้นงานพบว่ามีจำนวนชิ้นงานที่ต้องดำเนินงานก่อนหน้าค่อนข้างน้อย จึงทำให้สามารถจัดสรรลำดับของชิ้นงานลงในแต่ละสถานีงานได้อย่างหลากหลาย และส่งผลให้ค่าความแปรปรวนของภาระงานลดลงได้มากกว่าปัญหาอื่น

6.3 ข้อเสนอแนะ

จากการดำเนินงานวิทยานิพนธ์ที่ผ่านมาและผลที่ได้รับ ทำให้เกิดแนวคิดและข้อเสนอแนะในการดำเนินงานวิจัยต่อไปในอนาคต ดังต่อไปนี้

1. ในการนำวิธี HTS ที่นำเสนอไปประยุกต์ใช้แก้ปัญหา ALB ปัญหาอื่นๆ จำเป็นที่จะต้องทำการทดสอบวิธี HTS เพื่อหาค่าตัวประกอบการค้นหาหรือค่าพารามิเตอร์ที่เหมาะสมใหม่ ไม่ว่าจะเป็นการรันการค้นหา หรือจำนวนค่าใกล้เคียง ทั้งนี้เนื่องมาจากค่าตัวประกอบการค้นหาหรือค่าพารามิเตอร์ที่เหมาะสมจะขึ้นอยู่กับปัญหาที่สนใจ (problem dependent)

2. สืบเนื่องจากข้อที่ 1 การวิเคราะห์หาค่าตัวประกอบการค้นหาหรือค่าพารามิเตอร์ที่เหมาะสมของวิธี HTS ที่นำเสนอควรได้รับการศึกษาเพิ่มเติม โดยอาจอาศัยแนวทางตามทฤษฎีการออกแบบการทดลอง (design of experiment: DOE) โดยอาจใช้วิธีเชิงสถิติอันได้แก่วิธี ANOVA หรือวิธี Duncan เพื่อวิเคราะห์หาค่าตัวประกอบการค้นหาหรือค่าพารามิเตอร์ที่มีนัยสำคัญและเหมาะสม

3. ควรมีการประยุกต์วิธี HTS ที่นำเสนอเพื่อการแก้ปัญหาการจัดสมดุลสายงานการประกอบรูปแบบอื่นๆ เช่นปัญหา ALB แบบผลิตภัณฑ์ผสมหรือแบบหลายผลิตภัณฑ์ (multi/mixed-model assembly line balancing problem) และปัญหา ALB ที่มีรูปทรงของสายงานการประกอบเป็นตัวอักษร U (U-shape assembly line balancing problem) เพื่อแสวงหาหนทางการใช้ประโยชน์วิธี HTS ที่นำเสนอให้กว้างขวางมากยิ่งขึ้น

4. สืบเนื่องจากข้อที่ 3 ในการประยุกต์วิธี HTS ที่นำเสนอเพื่อการแก้ปัญหาการจัดสมดุลสายงานการประกอบหรือปัญหา ALB ควรมีการศึกษาเพิ่มเติมเกี่ยวกับการกำหนดฟังก์ชันวัตถุประสงค์ในรูปแบบอื่นๆ ที่อาจส่งผลให้มีประสิทธิภาพมากยิ่งขึ้น เช่น การกำหนดฟังก์ชันวัตถุประสงค์แบบหลายวัตถุประสงค์ (multiple-objective function) โดยอาจเกิดจากการพิจารณาความแปรปรวนของภาระงาน เวลาว่างงานรวม จำนวนสถานีงาน และประสิทธิภาพของสายงานการประกอบร่วมกัน

5. ควรมีการศึกษาเพิ่มเติมเกี่ยวกับการค้นหาทางปัญญาประดิษฐ์หรือ AI ชนิดอื่นๆ เช่นการค้นหาแบบมด (ant search) หรือการหาค่าเหมาะที่สุดด้วยวิธีกลุ่มอนุภาค (particle swarm optimization: PSO) เพื่อประยุกต์ใช้ในการแก้ปัญหา ALB รูปแบบต่างๆ หรือการประยุกต์การค้นหาทางปัญญาประดิษฐ์แบบพันทางชนิดอื่นๆ เพื่อแก้ปัญหา ALB และทำการเปรียบเทียบผลหรือคำตอบที่ได้ในแง่มุมต่างๆ ให้ครอบคลุม อันจะนำมาซึ่งข้อสรุปว่าวิธีการใดมีประสิทธิภาพในการแก้ปัญหา ALB มากที่สุด

เอกสารอ้างอิง

- [1] Sury R.J. "Aspects of Assembly Line Balancing", **International Journal of Production Research**, vol. 9, 1971, pp.8-14.
- [2] Fonseca D.J., Guest C.L., Elam M., Karr C.L. "A Fuzzy Logic Approach to Assembly Line Balancing", **Mathware and Soft Computing**, vol. 12, 2005, pp.57-74.
- [3] Salveson M.E. "The Assembly Line Balancing Problem", **The Journal of Industrial Engineering**, vol. 40, no. 3, 1995, pp.18-25.
- [4] Gutjahr A.L., Nemhauser G.L. "An Algorithm for the Balancing Problem", **Management Science**, vol. 11, 1964, pp.23-35.
- [5] Jackson J.R. "A Computing Procedure for a Line Balancing Problem", **Management Science**, vol. 2, no. 3, 1956, pp.261-271.
- [6] Bowman E.H. "Assembly Line Balancing by Linear Programming", **Operation Research**, vol. 8, no. 3, 1960, pp.385-389.
- [7] Kilbridge M.D., Wester L., "A Heuristic Method of Assembly Line Balancing", **The Journal of Industrial Engineering**, vol. 12, no. 4, 1961, pp.292-298.
- [8] Arcus A.L., "COMSOAL: A Computer Method of Sequencing Operations for Assembly Line", **International Journal of Production Research**, vol. 4, 1966, pp.25-32.
- [9] กรรณิกา ศิลานนท์. "การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดสมดุลของสายงานการประกอบแบบหลายวัตถุประสงค์" วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต, จุฬาลงกรณ์มหาวิทยาลัย, 2542.
- [10] Rubinovitz J., Levitin G. "Genetic Algorithm for Assembly Line Balancing", **International Journal of Production Economics**, vol. 41, 1995, pp.343-354.
- [11] Chiang W.C. "The Application of the Tabu Search Metaheuristic to the Assembly Line Balancing Problem", **Journal of Operation Research**, vol. 77, 1998, pp.209-227.
- [12] จงกล เอี่ยมมิ. "การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดสมดุลสายการประกอบแบบผลิตภัณฑ์ผสม" วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต, จุฬาลงกรณ์มหาวิทยาลัย, 2543.
- [13] Areerak K-N., Sujitjorn S. "Performance Comparison between Genetic Algorithm and Tabu Search Methods", **Suranaree J. Sci. Technol**, vol. 9, no. 1, 2002, pp.61-68.
- [14] Scholl A. **Balancing and Sequencing of Assembly Lines**. Physica, Heidelberg, 1999.

- [15] Becker C., Scholl A. "A Survey on Problems and Methods in Generalized Assembly Line Balancing", **European Journal of Operational Research**, vol. 168, 2006, pp. 694-715.
- [16] Kriengkarakot N., Pianthong N. "The Assembly Line Balancing Problem: Review Articles", **KKU Engineering Journal**, vol. 34, no. 2, 2007, pp.133 - 140.
- [17] Mastor A.A. "An Experimental Investigation and Comparative Evaluation of Production Line Balancing Techniques", **Management Science**, vol. 16, no. 11, 1970, pp.728-746.
- [18] Johnson R.V. "Assembly Line Balancing Algorithms : Computation Comparisons", **International Journal of Production Research**, vol. 19, no. 3, 1981, pp.277-287.
- [19] Talbot F.B., Patterson J.H., Gehrlein W.V. "Comparative Evaluation of Heuristic Line Balancing Techniques", **Management Science**, vol. 32, no. 4, 1986, pp.430-454.
- [20] Baybars I. "A Survey of Exact Algorithms for the Simply Assembly Line Balancing Problem", **Management Science**, vol. 32, no. 8, 1986, pp.909-932.
- [21] Ghosh S., Gagon R.J. "A Comprehensive Literature Review and Analysis of the Design, Balancing and Scheduling of Assembly Systems", **International Journal of Production Research**, vol. 27, no. 4, 1989, pp.637-670.
- [22] Erel E., Sabuncuoglu I., Aksu B.A. "Balancing of U-type Assembly System Using Simulated Annealing", **International Journal of Production Research**, vol. 39, no. 13, 2001, pp.3003-3015.
- [23] Van Assche F., Herroelen W.S. "An Optimal Procedure for the Single-Model Deterministic Assembly Line Balancing Problem", **European Journal of Operations Research**, vol. 3, 1978, pp.142-149.
- [24] Mamoud K.I. "A Generalised Assembly Line Balancing Algorithms", Ph.D. Dissertation of University of Bradford, UK, 1989.
- [25] Hackman S.T., Magazine M.J., Wee T.S. "Fast, Effective Algorithms for Simple Assembly Line Balancing Problems", **Operations Research**, vol. 37, no. 6, 1989, pp.916-924.
- [26] Sarin S.C., Erel E., Dar-El. E.M. "A Methodology for Solving Single-Model, Stochastic Assembly Line Balancing Problem", **Omega**, vol. 27, 1999, pp.477-495.
- [27] Helgeson W.P., Birnie D.P. "Assembly Line Balancing using the Ranked Positional Weight Technique", **The Journal of Industrial Engineering**, vol. 12, no. 6, 1961, pp.394-398.

- [28] Hoffman T.R. "Assembly Line Balancing with a Precedence Matrix", **Management Science**, vol. 9, No. 4, 1963, pp.551-562.
- [29] Mansoor E.M. "Assembly Line Balancing an Improvement on the Ranked Positional Weight Technique", **The Journal of Industrial Engineering**, vol. 15, no. 2, 1964, pp.73-77.
- [30] Scholl A., Becker C. "State-of-the-art Exact and Heuristic Solution Procedures for Single Assembly Line Balancing", **European Journal of Operation Research**, vol. 168, 2007, pp. 666-693.
- [31] Bard J.F., Dar-El E., Shtub A. "An Analytic Framework for Sequencing Mixed Model Assembly Lines", **International Journal of Production Research**, vol. 30, 1992, pp.35-48.
- [32] Scholl A., Klein R., Domschke W. "Pattern Based Vocabulary Building for Effectively Sequencing Mixed Model Assembly Lines", **Journal of Heuristics**, vol. 4, 1998, pp.359-381.
- [33] Miltenberg G.J., Wijngaard J, "The U-line Line Balancing Problem", **Management Sciences**, vol. 40, no.10, 1994, pp.1378-1388.
- [34] Urban T.L. "Note. Optimal Balancing of U-Shaped Assembly Lines", **Management Sciences**, vol. 44, no. 5, 1998, pp.738-741.
- [35] Glover F. "Future Paths for Integer Programming and Links to Artificial Intelligence", **Computer and Operation Research**, vol. 13, 1986, pp.533-549.
- [36] Glover F. "Tabu Search-Part I", **ORSA Journal on Computing**, vol. 1, 1989, pp.190-206.
- [37] Glover F. "Tabu Search-Part II", **ORSA Journal on Computing**, vol. 2, 1990, pp.4-32.
- [38] Glover F., Laguna M. "**Tabu Search**", Kluwer Academic Publishers, Norwell, 1997.
- [39] Glover F., Hanafi S. "Finite Convergence of Tabu Search", **MIC'2001-4th Metaheuristic**, 2001, pp.333-336.
- [40] Hanafi S. "On the Convergence of Tabu Search", **Heuristic**, vol. 7, 2000, pp.47-58.
- [41] Glover F., Mulvey J.M., Hoyland K. "**Solving Dynamic Stochastic Control Problem in Finance using Tabu Search with Variable Scale**", Statistics and Operation Research Technical Report SOR-94-13, Princeton University, USA, 1994.
- [42] Kulwarawanichpong T., Sujitjorn S. "Optimal Power Flow using Tabu Search", **IEEE power Engineering Review**, vol. 22, no. 6, 2002, pp.37-40.

- [43] Yao X., Karady G.G., Farmer R.G., Agrawal B.L. "Estimation of Generator Excitation System Parameters by Tabu Search", **IEEE Power Engineering Society : Winter Meeting**, vol. 3, 2001, pp.160-175.
- [44] Nowicki E., Smutnicki C. "Fast Tabu Search Algorithm for the Flow Shop Problem", **European Journal of Operation Research**, vol. 91, 1996, pp160-175.
- [45] Battiti R., Techiolli G. "The Reactive Tabu Search", **ORSA Journal on computing**, vol. 6, no. 2, 1994, pp.126-140.
- [46] Battiti R. "**Reactive Search : Toward Self-Tuning Heuristics, Modern Heuristic Search Method**", John Wiley&Sons, 1996.
- [47] Kochetov Y.A., Goncharov E.N. "Probabilistic Tabu Search Algorithms for Multi-Stage Uncapacitated Facility Location Problem", **Operation Research Proceeding**, Springer, 2000, pp.65-70.
- [48] Puangdownreong D., Sujitjorn S., Kulworawanichpong T. "Convergence Analysis of Adaptive Tabu Search", **Science Society of Thailand**, vol. 30, no. 2, 2004, pp.183-190.
- [49] Puangdoenreong D., Kulworawanichpong T., Sujitjorn S. "Finite Convergence and Performance Evaluation of Adaptive Tabu Search", **LNCS (Lecture Notes in Computer Science)**, vol. 3, 2004, pp.710-717.
- [50] Sujitjorn S., Kulworawanichpong T., Puangdownreong D., Areerak K-N. "**Adaptive Tabu Search and Applications in Engineering Design**", in Integrated Intelligent Systems for Engineering Design, Edited by Xuan F.Z., Howlett R.J., IOS-Press., 2006, pp.233-257.
- [51] Yogathanan M. "**Application of Neural Network in Assembly Line Balancing**", AIT Thesis, NO.ISE-96-4, 1996.
- [52] Mapfaira H., Byrne M. "A Genetic Algorithm Approach for Assembly Line Balancing", **The 15th International Conference on Production Research**, 1999.
- [53] Suresh G., Vinod V.V., Sahu S. "A Genetic Algorithm for Assembly Line Balancing", **Production Planning and Control**, vol. 7, no. 1, 1996, pp.38-46.
- [54] Leu Y.Y., Matheson L.A., Rees L.P. "Assembly Line Balancing using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria", **Decision Sciences**, vol. 25, 1991, pp.581-606.

- [55] Pastor R., Corominas A. "Assembly Line Balancing with Work Zone and Station Load Timewindows Constraints", **The ISATP '99 Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning**, 1999, pp. 448-453
- [56] Chen R.S, Lu K.Y., Yu S.C. "A Hybrid Genetic Algorithm Approach on Multi-Objective of Assembly Planning Problem", **Artificial Intelligent**, vol. 15, 2002, pp.447-457.
- [57] Bautista J., Pereira J. "Ant Algorithms for a Time and Space Constrained Assembly Line Balancing Problem", **European Journal of Operation Research**, vol. 177, 2007, pp. 2016-2032.
- [58] Sedgewick R. "Permutation Generation Methods", **Computing Surveys**, vol. 9, 1977, pp.137-164.
- [59] Johnson S.M. "Generation of Permutations by Adjacent Transpositions", **Math. Computation**, vol. 17, 1963, pp.282-285.
- [60] Moses L.E., Oakford R.V. "**Tables of Random Permutations**", Stanford University Press, Stanford Calif., 1963.
- [61] Bohachevsky I.O., Johnson M.E., Stein M.L. "Generalized Simulated Annealing for Function Optimization", **Technometrics**, vol. 28, no. 3, 1986, pp. 209-218.
- [62] Scholl A., Klein R. "**Homepage for Assembly Line Balancing Research**", [Online]. Available : <http://www.assembly-line-balancing.de/>. 2007.
- [63] Scholl A., Klein R. "Balancing Assembly Lines Effectively - A Computational Comparison", **European Journal of Operational Research**, vol. 114, 1999, pp. 50-58.
- [64] Lila B., Jarukasetwit A., Kansunthia P. "A Flexible Line Balancing Program with Ranked Positional Weight Technique", **IE-Network Conference Proceeding**, 2004.

ภาคผนวก ก

รายละเอียดปัญหา ALB ที่ใช้ในงานวิจัย

ปัญหาการจัดสมมูลสายงานการประกอบ หรือปัญหา ALB ที่ใช้ในงานวิจัยวิทยานิพนธ์นี้เป็นปัญหา ALB แบบผลิตภัณฑ์เดี่ยวแบบ SMD ซึ่งมีอยู่ด้วยกัน 4 ปัญหา ดังรายละเอียดต่อไปนี้

ก.1 ปัญหา ALB ปัญหาที่ 1 (ปัญหา Buxey)

ปัญหา ALB ปัญหาที่ 1 คือปัญหา Buxey [14],[62-63] มีจำนวนชิ้นงานทั้งหมด 29 ชิ้นงาน เวลาการประกอบรวม (T) = 324 นาที สำหรับเวลาดำเนินงานของชิ้นงาน และลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน แสดงดังตารางที่ ก.1

ตารางที่ ก.1 รายละเอียดของสายงานการประกอบปัญหา Buxey

ชิ้นงาน	เวลาดำเนินงานของชิ้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน (ชิ้นงานที่ต้องดำเนินการก่อน)
1	7	-
2	19	-
3	15	1
4	5	3
5	12	4
6	10	2
7	8	-
8	16	5,6
9	2	7
10	6	9
11	21	8
12	10	7
13	9	5
14	4	10
15	14	10,12
16	7	8,14
17	14	11,13
18	17	16

ตารางที่ ก.1 (ต่อ)

ชั้นงาน	เวลาดำเนินงานของชั้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชั้นงาน (ชั้นงานที่ต้องดำเนินการก่อน)
19	10	15
20	16	17
21	1	19
22	9	18,21
23	25	20,22
24	14	23
25	14	1,7
26	2	2
27	10	26
28	7	23
29	20	24,25,27,28
$W = 324$ นาที		

ก.2 ปัญหา ALB ปัญหาที่ 2 (ปัญหา Gunther)

ปัญหา ALB ปัญหาที่ 2 คือปัญหา Gunther [14],[62-63] มีจำนวนชั้นงานทั้งหมด 35 ชั้นงาน เวลาการประกอบรวม = 483 นาที สำหรับเวลาดำเนินงานของชั้นงาน และลำดับความสัมพันธ์ก่อนหลังของชั้นงาน แสดงดังตารางที่ ก.2

ตารางที่ ก.2 รายละเอียดของสายงานการประกอบปัญหา Gunther

ชั้นงาน	เวลาดำเนินงานของชั้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชั้นงาน (ชั้นงานที่ต้องดำเนินการก่อน)
1	29	-
2	3	1
3	5	2
4	22	3
5	6	1
6	14	5
7	2	1,6
8	5	6
9	22	8
10	30	1
11	23	4

ตารางที่ ก.2 (ต่อ)

ชั้นงาน	เวลาดำเนินงานของชั้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชั้นงาน (ชั้นงานที่ต้องดำเนินการก่อน)
12	30	1
13	23	9
14	2	7,10
15	19	14
16	29	15
17	2	-
18	2	7,12
19	19	18
20	29	17,19
21	6	16,20
22	10	21
23	16	22
24	23	23
25	5	21
26	5	25
27	5	24,26
28	40	11,13,27
29	2	28
30	5	21
31	5	30
32	1	21,31
33	40	11,13,27,32
34	2	27
35	2	33
$W = 483$ นาที		

ก.3 ปัญหา ALB ปัญหาที่ 3 (ปัญหา Random)

ปัญหา ALB ปัญหาที่ 3 คือปัญหา Random ซึ่งเป็นปัญหาที่ถูกกำหนดขึ้นมาแบบสุ่ม ในลักษณะที่ไม่เจาะจง มีจำนวนชั้นงานทั้งหมด 45 ชั้นงาน เวลาการประกอบรวม = 552 นาที สำหรับเวลาดำเนินงานของชั้นงาน และลำดับความสัมพันธ์ก่อนหลังของชั้นงาน แสดงดังตารางที่ ก.3

ตารางที่ ก.3 รายละเอียดของสายงานการประกอบปัญหา Random

ชั้นงาน	เวลาดำเนินงานของชั้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชั้นงาน (ชั้นงานที่ต้องดำเนินการก่อน)
1	9	-
2	9	-
3	10	1
4	10	2
5	17	3
6	17	4
7	13	1
8	13	2
9	20	5,7
10	20	6,8
11	10	-
12	11	-
13	6	11,12
14	22	7,8,13
15	11	13
16	19	15
17	12	14
18	3	15
19	7	16,18
20	4	19
21	55	20
22	14	21
23	27	15
24	29	15
25	26	14
26	6	17,25
27	5	17
28	24	22,27
29	4	14
30	5	14
31	7	14
32	4	14
33	15	19,23,24,27

ตารางที่ ก.3 (ต่อ)

ชั้นงาน	เวลาดำเนินงานของชั้นงาน (นาที)	ลำดับความสัมพันธ์ก่อนหลังของชั้นงาน (ชั้นงานที่ต้องดำเนินการก่อน)
34	3	33
35	7	33
36	9	33
37	4	12
38	7	26,28,34,36
39	5	-
40	4	35,38
41	21	9,10,29,30,31,32,39,40
42	12	41
43	6	37
44	5	42
45	5	42
$W = 552$ นาที		

ก.4 ปัญหา ALB ปัญหาที่ 4 (ปัญหา M-cycle)

ปัญหา ALB ปัญหาที่ 4 คือปัญหาสายงานการประกอบมอเตอร์ไซค์ หรืออาจเรียกว่าปัญหา M-cycle ซึ่งเป็นปัญหาการจัดสมดุลสายงานการประกอบจริงในภาคอุตสาหกรรม โดยใช้รายละเอียดของสายงานการประกอบรถมอเตอร์ไซค์ของ บริษัทคาวาซากิ มอเตอร์ เอ็นเตอร์ไพรส์ ประเทศไทย จำกัด [64] ซึ่งมีจำนวนชั้นงานทั้งหมด 60 ชั้นงาน เวลาการประกอบรวม = 2,475 วินาที สำหรับรายละเอียดของชั้นงาน เวลาดำเนินงานของชั้นงาน และลำดับความสัมพันธ์ก่อนหลังของชั้นงาน แสดงดังตารางที่ ก.4

ตารางที่ ก.4 รายละเอียดของสายงานการประกอบปัญหา M-cycle

ชั้นงาน	เวลาดำเนินงานของ ชั้นงาน (วินาที)	รายละเอียดของชั้นงาน	ลำดับความสัมพันธ์ ก่อนหลังของชั้นงาน
1	40	Subassembly race (92048-009) to frame (32160-1773) and press number frame	-
2	50	Subassembly front fender (35004-1457) to front fork (44011-0033)	-
3	24	Assembly plug (92066-1081) to frame	1
4	75	Assembly front fork to frame	1,2
5	55	Assembly harness (92048-009) to frame	1
6	42	Assembly front panel brake (41035-1543) and front wheel (41001-0001) to frame	4

ตารางที่ ก.4 (ต่อ)

ชั้นงาน	เวลาดำเนินงานของ ชั้นงาน (วินาที)	รายละเอียดของชั้นงาน	ลำดับความสัมพันธ ก่อนหลังของชั้นงาน
7	50	Assembly engine (54105-1005) to frame	3,5
8	50	Assembly engine (54105-1005) to frame	6
9	40	Assembly guard engine and holder step (92015-1222) to frame	7,49
10	40	Assembly guard engine and holder step (92015-1222) to frame	8,49
11	50	Assembly right step (34028-1238) to frame	9
12	50	Assembly left step (34028-1237) to frame	10
13	17	Assembly bracket shroud (11052-1639) to frame	11
14	40	Subassembly bushing (29042-1088) to swing arm (33001-1622)	12
15	40	Assembly swing arm and shock (45014-0037) to frame	11
16	44	Subassembly rod torque (43007-1203) to swing arm	14
17	40	Assembly tube (92191-1700) to frame	11
18	55	Assembly swing arm and shock (45014-0037) to frame	16
19	36	Subassembly panel brake (41035-1542) and coupling (42033-1223) to rear wheel (41001-0002)	15
20	36	Assembly left bolt and rear fender (35022-1342) to frame	51
21	36	Assembly right bolt and rear fender (35022-1342) to frame	51
22	17	Assembly guard (12053-1459) to frame	18
23	80	Assembly rear wheel to frame	19,24
24	60	Assembly sprocket (41314-1323) and chain (92057-0090) to frame	22
25	45	Subassembly rear level brake (43001-1445) to rod brake (43011- 1186)	23
26	58	Assembly side stand (34024-1390) to frame	24
27	90	Assembly rear level brake to frame	25
28	40	Assembly guard chain (12053-1460) to frame	24
29	36	Assembly level front brake (46076-1107) to frame	53
30	34	Assembly case chain (36014-1258) to frame	24
31	60	Assembly grip throttle (46018-1434) to frame	29
32	54	Assembly cover chain (14026-1285) to frame	26,28,30
33	80	Subassembly cover (49107-1218) and cover (49107-1217) to muffler (18087-1168)	27,31
34	30	Assembly switch stop (27010-1332) to frame	53
35	76	Assembly muffler to frame	17,20,33
36	50	Assembly grip (46075-1024) to frame	34
37	54	Assembly cover (49107-1213) to frame	35
38	36	Assembly coil (21121-1331) to frame	12
39	22	Assembly damper (95161-1284) to frame	35
40	17	Assembly igniter (21119-1612) to frame	12

ตารางที่ ก.4 (ต่อ)

ชิ้นงาน	เวลาดำเนินงานของ ชิ้นงาน (วินาที)	รายละเอียดของชิ้นงาน	ลำดับความสัมพันธ ก่อนหลังของชิ้นงาน
41	36	Assembly cover (48107-1221) to frame	35
42	54	Assembly air filter (11010-1773) to frame	38,40
43	34	Assembly level kick (13064-0012) to frame	37,39,41
44	30	Assembly level change (13156-1667) to frame	32
45	34	Assembly right side cover (36001-1667) to frame	43,59
46	34	Assembly left side cover (36001-1639) to frame	36,44,59
47	40	Assembly right shroud (49089-1145) to frame	45
48	40	Assembly left shroud (49089-1144) to frame	46
49	26	Subassembly damper (92075-015) to guard engine (5020-1784)	-
50	22	Assembly nut holder upper (92015-1653) to frame	4
51	48	Assembly rear fender (35022-1342) to frame	1
52	17	Assembly regulator (21066-1086) to frame	1
53	65	Assembly handle (46003-0056) to frame	2,49
54	30	Assembly damper tank (92161-1077) to frame	1
55	17	Assembly damper tank (92075-1787) to frame	1
56	22	Assembly tank (51082-5165) to frame	13,42,54,55
57	26	Assembly bolt tank (132G0614) to frame	56
58	17	Assembly damper (92072-1229) to frame	57
59	17	Assembly seat (53066-0052) to frame	58
60	17	Assembly cap tank (51048-1124) to frame	59
$W = 2,475$ วินาที			

ภาคผนวก ข

รายละเอียดโปรแกรมการค้นหาแบบตาบอด

โปรแกรมการค้นหาแบบตาบอด หรือ โปรแกรม TS ที่ได้รับการพัฒนาขึ้นโดยโปรแกรม MATLAB 7.0 ประกอบด้วยโปรแกรมหลัก และ โปรแกรมย่อย 4 โปรแกรม ดังรายละเอียดต่อไปนี้

ข.1 โปรแกรมหลัก (TS.m)

```
%-----%
%
%                TS Program (TS.m)                %
%                by Supaporn Suwannarongsri        %
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007   %
%-----%

function [b_ne,b_er,time,CT,loc]=TS(N)                % สร้างฟังก์ชัน TS เพื่อเป็นโปรแกรมหลัก
% โดยมี N เป็นอินพุตของโปรแกรม
% N คือจำนวนพารามิเตอร์ที่ต้องการค้นหา
% ในที่นี้ให้ N = 2 (เพื่อเป็นตัวอย่าง)
% เอาต์พุตของโปรแกรมประกอบด้วยค่าพารามิเตอร์
% ที่ดีที่สุดหรือค่าคำตอบวงกว้าง (b_ne),
% ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด (b_er),
% ค่าเวลาที่ใช้ในการค้นหา (time),
% ค่าจำนวนรอบการค้นหาทั้งหมด (CT),
% และค่าคำตอบวงแคบเฉพาะถิ่น (loc)

xlim=[1 1;-1 -1];                                  % กำหนดปริภูมิการค้นหาของพารามิเตอร์
% แถวที่ 1 ของ xlim คือขอบเขตบนของพารามิเตอร์,
% แถวที่ 2 ของ xlim คือขอบเขตล่างของพารามิเตอร์

for r=1:5
    S(r,:)=((xlim(1,:)-xlim(2,:)).*rand(1,N))+xlim(2,:);
end                                                  % สุ่มคำตอบเริ่มต้นภายในปริภูมิการค้นหาของ
% พารามิเตอร์แต่ละตัวจำนวน 5 ชุด
% โดยอาศัยความสัมพันธ์
%  $(x_{n,i}-x_{i,i})rand + x_{i,i}$ 

for k=1:size(S,1)
    outsim=PreObj(S(k,1),S(k,2));                    % ประเมินคำตอบเริ่มต้นจากฟังก์ชันวัตถุประสงค์
```

```

CostV(k,1)=outsim; % ในโปรแกรม PreObj โดยที่อินพุตของโปรแกรม PreObj
end % คือค่าตอบเริ่มต้นที่สุ่มได้จำนวน 5 ชุด
[b_er,index]=min(CostV); % เอาต์พุตคือค่าฟังก์ชันวัตถุประสงค์ที่ได้จากคำตอบ
% แต่ละชุด จากนั้นเก็บค่าฟังก์ชันวัตถุประสงค์ทั้ง 5 ค่า
% ไว้ใน CostV
S0=S(index,:); % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์
% มีค่าน้อยที่สุดไว้ใน S0
max_CT=500; % กำหนดจำนวนรอบสูงสุดในการค้นหา
% เพื่อใช้เป็นเกณฑ์ยุติการค้นหา
b_ne=S0; % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์
% มีค่าน้อยที่สุดไว้ใน b_ne ปัจจุบัน
ne_list=zeros(5,N); % รีเซ็ตค่า ne_list
radius=0.2; % กำหนดครีสมีกการค้นหาเริ่มต้น
N_ne=30; % กำหนดจำนวนค่าใกล้เคียงเพื่อการสุ่ม
ovl_b_er=b_er; % ปรับค่า ovl_b_er = b_er
ovl_ne=b_ne; % ปรับค่า ovl_ne = b_ne
n=0; % ตั้งค่าเริ่มต้นของจำนวนการซ้ำของคำตอบ
t=0; % ตั้งค่าตัวเลื่อนสำหรับ loc
tt=0; % ตั้งค่าตัวเลื่อนสำหรับ TL
ttt=0; % ตั้งค่าตัวเลื่อนสำหรับ b_er_list
CT=0; % ตั้งค่าเริ่มต้นของรอบการค้นหา
n_BT=0; % ตั้งค่าเริ่มต้นของจำนวนการซ้ำของคำตอบ
% เพื่อเรียกใช้กลไกย้อนรอยการค้นหา
tic; % เริ่มต้นคำนวณเวลาการค้นหา
t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ loc
tt=tt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ TL
ttt=ttt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ b_er_list
disp([CT b_er ovl_b_er]) % แสดงค่า CT, b_er, และ ovl_b_er
loc(t,1)=CT; % เก็บค่า CT ไว้ในสดมภ์ที่ 1 ของ loc
loc(t,2:3)=b_ne; % เก็บค่า b_ne ไว้ในสดมภ์ที่ 2 และ 3 ของ loc
loc(t,4)=b_er; % เก็บค่า b_er ไว้ในสดมภ์ที่ 4 ของ loc
TL(tt,1)=CT; % เก็บค่า CT ไว้ในสดมภ์ที่ 1 ของ TL
TL(tt,2:3)=b_ne; % เก็บค่า b_ne ไว้ในสดมภ์ที่ 2 และ 3 ของ TL
TL(tt,4)=b_er; % เก็บค่า b_er ไว้ในสดมภ์ที่ 4 ของ TL
b_er_list(ttt,1)=CT; % เก็บค่า CT ไว้ในสดมภ์ที่ 1 ของ b_er_list
b_er_list(ttt,2:3)=b_ne; % เก็บค่า b_ne ไว้ในสดมภ์ที่ 2 และ 3 ของ b_er_list

```

```

b_er_list(ttt,4)=b_er; % เก็บค่า b_er ไว้ในสดมภ์ที่ 4 ของ b_er_list
for CT=1:max_CT % วนรอบการค้นหา
    S1=RAN_ne(N_ne,radius,xlim,S0); % เรียกใช้โปรแกรม RAN_ne เพื่อสุ่มค่าใกล้เคียง
    % รอบคำตอบ S0 ภายในปริภูมิการค้นหาที่ย่อยปัจจุบัน
    % อินพุตของโปรแกรมนี้คือ N_ne, radius, xlim, และ S0
    % เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ N_ne
    [b_er1,b_ne1,b_er,b_ne]=PostObj(S1,b_er,S0); % เรียกใช้โปรแกรม PostObj เพื่อประเมินค่าใกล้เคียง
    % อินพุตของโปรแกรมนี้คือ S1, b_er,
    % และ S0 เอาต์พุตคือ b_er1, b_ne1, b_er, และ b_ne
    ne_list(k,1:size(S0,2))=[b_ne1]; % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์
    % มีค่าน้อยที่สุดในปริภูมิการค้นหาที่ย่อยปัจจุบัน
    % ไว้ในสดมภ์ที่ 1 และ 2 ของ ne_list
    ne_list(k,size(S0,2)+1)=b_er1; % เก็บค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
    % ภายในปริภูมิการค้นหาที่ย่อยปัจจุบัน ไว้ใน
    % สดมภ์ที่ 3 ของ ne_list
    if (b_er1-b_er)>1e-18 % ตรวจสอบการเข้าค่าคำตอบ โดยการเปรียบเทียบระหว่าง
        n=n+1; % b_er1 และ b_er ถ้ามีการเข้าค่าคำตอบ ให้ปรับเพิ่มค่า n
    else
        n=0;
    end
    tt=tt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ TL
    TL(tt,1)=CT; % เก็บค่า CT ไว้ในสดมภ์ที่ 1 ของ TL
    TL(tt,2:3)=b_ne1; % เก็บค่า b_ne1 ไว้ในสดมภ์ที่ 2 และ 3 ของ TL
    TL(tt,4)=b_er1; % เก็บค่า b_er1 ไว้ในสดมภ์ที่ 4 ของ TL
    ttt= ttt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ b_er_list
    b_er_list(ttt,1)=CT; % เก็บค่า CT ไว้ในสดมภ์ที่ 1 ของ b_er_list
    b_er_list(ttt,2:3)=b_ne; % เก็บค่า b_ne ไว้ในสดมภ์ที่ 2 และ 3 ของ b_er_list
    b_er_list(ttt,4)=b_er; % เก็บค่า b_er ไว้ในสดมภ์ที่ 4 ของ b_er_list
    disp([CT b_er ovl_b_er]) % แสดงค่า CT, b_er, และ ovl_b_er
    if n>=5 % เรียกใช้กลไกย้อนรอยการค้นหา
        % เมื่อจำนวนการเข้าค่าคำตอบเท่ากับ 5
        n_BT=n_BT+1; % เพิ่มค่า n_BT
        TEMP=TL(CT-3:CT+1,:); % ทำการจัดลำดับ (ranking) ค่าฟังก์ชันวัตถุประสงค์
        % และค่าพารามิเตอร์ 5 ชุดสุดท้ายก่อนมีการเรียกใช้
        % กลไกย้อนรอยการค้นหา
        % TEMP จะเก็บค่าฟังก์ชันวัตถุประสงค์

```

% และค่าพารามิเตอร์ 5 ชุดสุดท้ายก่อนทำการจัดลำดับ

```
[MAX,INDEX] = max(TEMP(:,4));
RANK(5,:) = TEMP(INDEX,:);TEMP(INDEX,4)= 0;
[MAX,INDEX] = max(TEMP(:,4));
RANK(4,:) = TEMP(INDEX,:);TEMP(INDEX,4)= 0;
[MAX,INDEX] = max(TEMP(:,4));
RANK(3,:) = TEMP(INDEX,:);TEMP(INDEX,4)= 0;
[MAX,INDEX] = max(TEMP(:,4));
RANK(2,:) = TEMP(INDEX,:);TEMP(INDEX,4)= 0;
[MAX,INDEX] = max(TEMP(:,4));
RANK(1,:) = TEMP(INDEX,:);TEMP(INDEX,4)= 0;
ne=RANK(5,2:3);           % แทนค่า ne ด้วยค่าพารามิเตอร์ลำดับที่ 5 ใน RANK
S0=ne;                    % ปรับค่า S0 = ne
if b_er<ovl_b_er          % ถ้า b_er < ovl_b_er
    ovl_b_er=b_er;        % ให้แทนค่า ovl_b_er ด้วย b_er
    ovl_b_ne=b_ne;        % และแทนค่าพารามิเตอร์ ovl_b_ne ด้วย b_ne
    t=t+1;                % ปรับเพิ่มค่าตัวเลื่อนสำหรับ loc
    loc(t,1)=CT;          % เก็บค่า CT ไว้ในสดมภ์ที่ 1 ของ loc
    loc(t,2:3)=b_ne;      % เก็บค่า b_ne ไว้ในสดมภ์ที่ 2 และ 3 ของ loc
    loc(t,4)=b_er;        % เก็บค่า b_er ไว้ในสดมภ์ที่ 4 ของ loc
end
    b_er=RANK(5,4);        % แทนค่า b_er ด้วยค่าฟังก์ชันวัตถุประสงค์ลำดับที่ 5
                            % ใน RANK
    n=0;                   % ปรับค่า n = 0
else
    S0=b_ne;               % แทนค่า S0 ด้วย b_ne
    b_er=b_er;             % แทนค่า b_er ด้วย b_er อีกครั้ง
end
end
    end
if ovl_b_er<b_er          % ถ้า ovl_b_er < b_er
    b_er=ovl_b_er;        % ให้แทนค่า b_er ด้วย ovl_b_er
    b_ne=ovl_b_ne;        % และแทนค่าพารามิเตอร์ b_ne ด้วย ovl_b_ne
end
time=toc;                 % ยุติการคำนวณเวลาการค้นหา
return
%-----%
```

ข.2 โปรแกรม Pre-Objective (PreObj.m)

```

%-----%
%                               Pre-Objective Program (PreObj.m)                               %
%                               by Supaporn Suwannarongsri                               %
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007   %
%-----%

function outsim=PreObj(x,y)                % สร้างฟังก์ชัน PreObj เป็น โปรแกรมย่อย
                                           % เพื่อคำนวณฟังก์ชันวัตถุประสงค์
                                           % กำหนดให้เป็นฟังก์ชัน โบฮาเชฟสกี (เพื่อเป็นตัวอย่าง)

    outsim=(x.^2)+(2*y.^2)-(0.3*cos(3*pi*x))-(0.4*cos(4*pi*y))+0.7;

return
%-----%

```

ข.3 โปรแกรม Post-Objective (PostObj.m)

```

%-----%
%                               Post-Objective Program (PostObj.m)                               %
%                               by Supaporn Suwannarongsri                               %
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007   %
%-----%

function [b_er1,b_ne1,b_er,b_ne]=PostObj(S1,b_er,S0)

                                           % สร้างฟังก์ชัน PostObj เป็น โปรแกรมย่อย PostObj
                                           % เพื่อประเมินค่าใกล้เคียงที่สุ่ม ได้รอบ S0
                                           % อินพุตของ โปรแกรมนี้คือ S1, b_er, และ S0
                                           % เอาต์พุตคือ b_er1, b_ne1, b_er, และ b_ne

    er=[];                                % กำหนดตัวแปร er ไว้รองรับค่าฟังก์ชันวัตถุประสงค์
                                           % ของค่าใกล้เคียง

    for k=1:size(S1,1)                    % ประเมินค่าใกล้เคียงใน S1 ผ่านฟังก์ชันวัตถุประสงค์
        outsim=PreObj(S1(k,1),S1(k,2)); % ใน โปรแกรม PreObj
        er(k,1)=outsim;                  % เก็บค่าฟังก์ชันวัตถุประสงค์ของค่าใกล้เคียงทั้งหมด
    end

    [b_er1,index]=min(er);                % หาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุด
                                           % ในบรรดาค่าใกล้เคียง และเก็บไว้ใน b_er1

    b_ne1=S1(index,:);                    % เก็บค่าพารามิเตอร์ของค่าใกล้เคียงที่ทำให้
                                           % ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุดไว้ใน b_ne1

```

```

if b_er1 < b_er % ปรับค่า b_er
    b_er = b_er1; % ถ้า b_er1 < b_er ให้แทนค่า b_er ด้วย b_er1
    b_ne = S1(index,:); % และแทนค่าพารามิเตอร์ b_ne ด้วย S1(index,:)
else
    b_ne = S0; % แทนค่า b_ne ด้วย S0
end
return
%-----%

```

ข.4 โปรแกรม RAN (RAN.m)

```

%-----%
%                               Random Program (RAN.m) %
%                               by Supaporn Suwannarongsri %
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007 %
%-----%
function x=RAN(a,b) % สร้างฟังก์ชัน RAN เป็นโปรแกรมย่อย
                    % เพื่อสุ่มค่าพารามิเตอร์ในช่วงที่กำหนด
                    % อินพุตของโปรแกรมคือขอบเขตบนของพารามิเตอร์ (b)
                    % และขอบเขตล่างของพารามิเตอร์ (a)
                    % โดยอาศัยความสัมพันธ์ (b-a)rand+a
                    % เอาต์พุตของโปรแกรมคือผลของการสุ่มค่าพารามิเตอร์
                    % ในช่วง a ถึง b
    x = a + rand*(b-a);
return
%-----%

```

ข.5 โปรแกรม RAN_ne (RAN_ne.m)

```

%-----%
%                               Random Neighborhood Program (RAN_ne.m) %
%                               by Supaporn Suwannarongsri %
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007 %
%-----%
function S1=RAN_ne(N_ne,radius,xlim,S0) % สร้างฟังก์ชัน RAN_ne เป็นโปรแกรมย่อย
                                        % เพื่อสุ่มค่าใกล้เคียงรอบค่าตอบ S0
                                        % อินพุตของโปรแกรมนี้คือ N_ne, radius, xlim, และ S0
                                        % เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ N_ne

```

```

for u=1:N_ne
    for k=1:size(xlim,2)
        S1(u,k)=S0(1,k)+(radius*(xlim(1,k)-xlim(2,k))*RAN(-1,1));
        % ทำการสุ่มค่าใกล้เคียงโดยอาศัยความสัมพันธ์
        %  $S_{1,i} = S_0 + [\text{radius} \times \text{rand}(-1, 1) \times (x_{h,i} - x_{l,i})]$ 
        % เรียกใช้โปรแกรม RAN
        % โดยให้ทำการสุ่มค่าในช่วง -1 ถึง 1
        % นั้นหมายความว่าค่าสุ่มมีพื้นที่อยู่ในช่วง
        %  $S_0 \pm [\text{radius} \times (x_{h,i} - x_{l,i})]$ 
        while ( S1(u,k)>xlim(1,k) | S1(u,k)<xlim(2,k) )
            % ค่าใกล้เคียงที่สุ่มได้จะต้องอยู่ในขอบเขตบนและ
            % ขอบเขตล่างของค่าพารามิเตอร์
            % ถ้าค่าใกล้เคียงที่สุ่มได้ค่าใดมีค่าเกินขอบเขตบนและ
            % ขอบเขตล่างที่กำหนดให้ทำการสุ่มค่าใกล้เคียงใหม่
            S1(u,k)=S0(1,k)+(radius*(xlim(1,k)-xlim(2,k))*RAN(-1,1));
        end
    end
end
return
%-----%

```

ภาคผนวก ค

รายละเอียดโปรแกรมการค้นหาแบบตามูเชิงพัน ทางเพื่อแก้ปัญหา ALB

โปรแกรมการค้นหาแบบตามูเชิงพันทาง หรือโปรแกรม HTS ที่ได้รับการพัฒนาขึ้นเพื่อแก้ปัญหา ALB โดยโปรแกรม MATLAB 7.0 ประกอบด้วยโปรแกรมหลัก และโปรแกรมย่อย 8 โปรแกรม ดังรายละเอียดต่อไปนี้

ค.1 โปรแกรมหลัก (HTS.m)

```
%-----%
%
%           HTS Program (HTS.m)
%           by Supaporn Suwannarongsri
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007
%-----%

function [Best_Idle_Time,Best_Wv,Best_Leff,Best_StationTask,Best_TaskSeq,time]=Mcycle()
% สร้างฟังก์ชัน Mcycle เป็น โปรแกรมหลัก
% เพื่อแก้ปัญหา ALB
% ในที่นี้ใช้ปัญหา M-cycle (เพื่อเป็นตัวอย่าง)
% เอาต์พุตของโปรแกรมประกอบด้วย
% Best_Idle_Time คือเวลาว่างงานน้อยที่สุด
% Best_Wv คือค่าความแปรปรวนของภาระงานน้อยที่สุด
% Best_Leff คือค่าประสิทธิภาพของสายงานมากที่สุด
% Best_StationTask คือจำนวนชิ้นงานของ
% แต่ละสถานีงานที่เหมาะสมที่สุด
% Best_TaskSeq คือลำดับของชิ้นงานที่เหมาะสมที่สุด
% time คือเวลาในการค้นหา

global pn TaskSeq N ct Tw Prec We W
% กำหนดตัวแปรโกลบอลเพื่อใช้กับทุกโปรแกรมย่อย
% pn คือจำนวนชิ้นงานของแต่ละสถานีงาน
% TaskSeq คือลำดับชิ้นงานที่ได้จากการสลับที่แบบสุ่ม
% N คือจำนวนสถานีงานสูงสุด
% ct คือรอบเวลาการผลิต
% Tw คือเวกเตอร์เวลาการทำงานของแต่ละชิ้นงาน
```

```

for ii=1:1000
    a1=randperm(49);
    j=1;
    for i=1:49
        if a1(i)<3||a1(i)>48
            a(j)=a1(i);
            j=j+1;
        end
    end
    b1=randperm(55);
    j=1;
    for i=1:55
        if b1(i)>2
            b2(j)=b1(i);
            j=j+1;
        end
    end
    j=1;
    for i=1:53
        if b2(i)<6||b2(i)>50
            b(j)=b2(i);
            j=j+1;
        end
    end
    c1=randperm(50);
    j=1;
    for i=1:50
        if c1(i)>5
            c2(j)=c1(i);
            j=j+1;
        end
    end
end

```

% Prec คือเมตริกซ์ลำดับชั้นงานก่อนหลัง
 % We คือจำนวนชั้นงานทั้งหมด
 % W คือเวลาการทำงานรวมของสายงานการประกอบ
 % ทำการสลับที่แบบสุ่มจำนวน 1,000 ครั้ง
 % เพื่อจัดสรรลำดับของชั้นงานในแต่ละกลุ่ม
 % ชั้นงานในกลุ่มที่ 1 ประกอบด้วยชั้นงานที่ 1, 2 และ 49
 % ชั้นงานในกลุ่มที่ 2 ประกอบด้วยชั้นงานที่ 3, 4, 5, 51,
 % 52, 53, 54 และ 55
 % ชั้นงานในกลุ่มที่ 3 ประกอบด้วยชั้นงานที่ 6, 7, 20, 21,
 % 29, 34 และ 50

```

j=1;
for i=1:45
    if c2(i)<8||c2(i)>19
        c3(j)=c2(i);
        j=j+1;
    end
end
j=1;
for i=1:33
    if c3(i)<22||c3(i)>28
        c4(j)=c3(i);
        j=j+1;
    end
end
j=1;
for i=1:26
    if c4(i)<30||c4(i)>33
        c5(j)=c4(i);
        j=j+1;
    end
end
j=1;
for i=1:22
    if c5(i)<35||c5(i)>49
        c(j)=c5(i);
        j=j+1;
    end
end
d1=randperm(36);
j=1;
for i=1:36
    if d1(i)>7
        d2(j)=d1(i);
        j=j+1;
    end
end
end

```

% ชั้นงานในกลุ่มที่ 4 ประกอบด้วยชั้นงานที่ 8, 9 และ 36

```

j=1;
for i=1:29
    if d2(i)<10||d2(i)>35
        d(j)=d2(i);
        j=j+1;
    end
end
e1=randperm(11);
% ชั้นงานในกลุ่มที่ 5 ประกอบด้วยชั้นงานที่ 10 และ 11
j=1;
for i=1:11
    if e1(i)>9
        e(j)=e1(i);
        j=j+1;
    end
end
f1=randperm(17);
% ชั้นงานในกลุ่มที่ 6 ประกอบด้วยชั้นงานที่ 12, 13,
% 15 และ 17
j=1;
for i=1:17
    if f1(i)>11
        f2(j)=f1(i);
        j=j+1;
    end
end
j=1;
for i=1:6
    if f2(i)<14||f2(i)>14
        f3(j)=f2(i);
        j=j+1;
    end
end
j=1;
for i=1:5
    if f3(i)<16||f3(i)>16
        f(j)=f3(i);
        j=j+1;
    end
end

```

```

end

g1=randperm(40);
j=1;
for i=1:40
    if g1(i)>13
        g2(j)=g1(i);
        j=j+1;
    end
end
j=1;
for i=1:27
    if g2(i)<15||g2(i)>15
        g3(j)=g2(i);
        j=j+1;
    end
end
j=1;
for i=1:26
    if g3(i)<17||g3(i)>18
        g4(j)=g3(i);
        j=j+1;
    end
end
j=1;
for i=1:24
    if g4(i)<20||g4(i)>37
        g5(j)=g4(i);
        j=j+1;
    end
end
j=1;
for i=1:6
    if g5(i)<39||g5(i)>39
        g(j)=g5(i);
        j=j+1;

```

% ชั้นงานในกลุ่มที่ 7 ประกอบด้วยชั้นงานที่ 14, 16, 19,
% 38 และ 40

```

    end
end
h1=randperm(42);
j=1;
for i=1:42
    if h1(i)>17
        h2(j)=h1(i);
        j=j+1;
    end
end
j=1;
for i=1:25
    if h2(i)<19||h2(i)>30
        h3(j)=h2(i);
        j=j+1;
    end
end
j=1;
for i=1:13
    if h3(i)<32||h3(i)>41
        h(j)=h3(i);
        j=j+1;
    end
end
end
m1=randperm(56);
j=1;
for i=1:56
    if m1(i)>21
        m2(j)=m1(i);
        j=j+1;
    end
end
j=1;
for i=1:35
    if m2(i)<23||m2(i)>55
        m(j)=m2(i);

```

% ชั้นงานในกลุ่มที่ 8 ประกอบด้วยชั้นงานที่ 18, 21 และ 42

% ชั้นงานในกลุ่มที่ 9 ประกอบด้วยชั้นงานที่ 22 และ 56

```

        j=j+1;
    end
end
n1=randperm(57);
j=1;
for i=1:57
    if n1(i)>23
        n2(j)=n1(i);
        j=j+1;
    end
end
j=1;
for i=1:34
    if n2(i)<25||n2(i)>56
        n(j)=n2(i);
        j=j+1;
    end
end
o1=randperm(58);
j=1;
for i=1:58
    if o1(i)>22
        o2(j)=o1(i);
        j=j+1;
    end
end
j=1;
for i=1:36
    if o2(i)<24||o2(i)>25
        o3(j)=o2(i);
        j=j+1;
    end
end
j=1;
for i=1:34
    if o3(i)<27||o3(i)>27

```

% ชั้นงานในกลุ่มที่ 10 ประกอบด้วยชั้นงานที่ 24 และ 57

% ชั้นงานในกลุ่มที่ 11 ประกอบด้วยชั้นงานที่ 23, 26, 28,
% 30 และ 58

```

        o4(j)=o3(i);
        j=j+1;
    end
end
j=1;
for i=1:33
    if o4(i)<29||o4(i)>29
        o5(j)=o4(i);
        j=j+1;
    end
end
j=1;
for i=1:32
    if o5(i)<31||o5(i)>57
        o(j)=o5(i);
        j=j+1;
    end
end
p1=randperm(59);
j=1;
for i=1:59
    if p1(i)>24
        p2(j)=p1(i);
        j=j+1;
    end
end
j=1;
for i=1:35
    if p2(i)<26||p2(i)>31
        p3(j)=p2(i);
        j=j+1;
    end
end
j=1;
for i=1:29
    if p3(i)<33||p3(i)>58

```

% ชั้นงานในกลุ่มที่ 12 ประกอบด้วยชั้นงานที่ 25,
% 32 และ 59

```

        p(j)=p3(i);
        j=j+1;
    end
end
q1=randperm(60);
j=1;
for i=1:60
    if q1(i)>26
        q2(j)=q1(i);
        j=j+1;
    end
end
j=1;
for i=1:34
    if q2(i)<28||q2(i)>43
        q3(j)=q2(i);
        j=j+1;
    end
end
j=1;
for i=1:18
    if q3(i)<45||q3(i)>59
        q(j)=q3(i);
        j=j+1;
    end
end
r1=randperm(41);
j=1;
for i=1:41
    if r1(i)>36
        r2(j)=r1(i);
        j=j+1;
    end
end
j=1;
for i=1:5

```

% ชั้นงาน ในกลุ่มที่ 13 ประกอบด้วยชั้นงานที่ 27, 44,
% และ 60

% ชั้นงาน ในกลุ่มที่ 14 ประกอบด้วยชั้นงานที่ 37, 39,
% และ 41

```

    if r2(i)<38||r2(i)>38
        r3(j)=r2(i);
        j=j+1;
    end
end
j=1;
for i=1:4
    if r3(i)<40||r3(i)>40
        r(j)=r3(i);
        j=j+1;
    end
end
s1=randperm(46); % ชิ้นงานในกลุ่มที่ 15 ประกอบด้วยชิ้นงานที่ 45 และ 46
j=1;
for i=1:46
    if s1(i)>44
        s(j)=s1(i);
        j=j+1;
    end
end
u1=randperm(48); % ชิ้นงานในกลุ่มที่ 16 ประกอบด้วยชิ้นงานที่ 47 และ 48
j=1;
for i=1:48
    if u1(i)>46
        u(j)=u1(i);
        j=j+1;
    end
end
TaskSeq(ii,:)=['a b c d e f g h m n o p q 33 35 r 43 s u']; % เก็บชุดลำดับของชิ้นงานที่ได้จากการ
End % สลับที่แบบสุ่มไว้ใน TaskSeq
N=8; % กำหนดสถานีงานสูงสุดเท่ากับ 8 สถานีงาน
ct=360; % กำหนดรอบเวลาการผลิตเท่ากับ 360 วินาที
We=60; % กำหนดจำนวนชิ้นงานทั้งหมด
Tw=[40;50;24;75;55;42;50;50;40;40;50;50;17;40;40;44;40;55;36;36;36;17;80;60;45;58;90;40;36;34;
60;54;80;30;76;50;54;36;22;17;36;54;34;30;34;34;40;40;26;22;48;17;65;30;17;22;26;17;17;17];
% กำหนดเวกเตอร์เวลาการทำงานของแต่ละชิ้นงาน

```

```

Prec=[-1 0 0 0 0;-1 0 0 0 0;1 0 0 0 0;1 2 0 0 0;1 0 0 0 0;
4 0 0 0 0;3 5 0 0 0;6 0 0 0 0;7 4 9 0 0 0;8 4 9 0 0 0;9 0 0 0 0;
10 0 0 0 0;11 0 0 0 0;12 0 0 0 0;11 0 0 0 0;14 0 0 0 0;11 0 0 0 0;
16 0 0 0 0;15 0 0 0 0;51 0 0 0 0;51 0 0 0 0;18 0 0 0 0;19 24 0 0 0;22 0 0 0 0;
23 0 0 0 0;24 0 0 0 0;25 0 0 0 0;24 0 0 0 0;53 0 0 0 0;24 0 0 0 0;29 0 0 0 0;26 28 30 0 0;
27 31 0 0 0;53 0 0 0 0;17 20 33 0 0;34 0 0 0 0;35 0 0 0 0;12 0 0 0 0;35 0 0 0 0;12 0 0 0 0;
35 0 0 0 0;38 40 0 0 0;37 39 41 0 0;32 0 0 0 0;43 59 0 0 0;36 44 59 0 0;45 0 0 0 0;46 0 0 0 0;
-1 0 0 0 0;4 0 0 0 0;1 0 0 0 0;1 0 0 0 0;2 4 9 0 0 0;1 0 0 0 0;1 0 0 0 0;13 42 54 55 0;56 0 0 0 0;
57 0 0 0 0;58 0 0 0 0;59 0 0 0 0];

% กำหนดเมตริกซ์ลำดับชั้นงานก่อนหลัง
W=sum(Tw); % คำนวณเวลาการทำงานรวมของสายงานการประกอบ
max_CT=50; % กำหนดจำนวนรอบการค้นหาสูงสุด
xlim=[10 10 9 10 10 8 9 4;8 8 7 8 8 6 7 1]; % กำหนดปริภูมิการค้นหา
% นั่นคือขอบเขตของชั้นงานแต่ละสถานีงาน
[loc_CT,loc_b_ne,loc_b_er,loc_b_TS,time]=TS(N,max_CT,xlim); % เรียกใช้ฟังก์ชัน TS
Ps=loc_b_ne; % ให้ Ps คือจำนวนชั้นงานที่เหมาะสมที่สุดของแต่ละ
สถานีงาน
Ts=loc_b_TS; % ให้ Ts คือลำดับของชั้นงานที่เหมาะสมที่สุด
Er=loc_b_er; % ให้ Er คือค่าความแปรปรวนของภาระงานที่น้อยที่สุด
TT=0;
q=1;
for k=1:size(Ps,1)
    Station_task=Sta_Seq(Ps(k,:),Ts(k,:),N); % จัดสรรจำนวนชั้นงานที่เหมาะสมให้แต่ละสถานีงาน
    for i=1:size(Station_task,1) % และจัดสรรลำดับของชั้นงานเข้าสู่สถานีงานต่างๆ
        for j=1:size(Station_task,2)
            A=Station_task(i,j);
            if A==0
                break;
            end
            TT=TT+Tw(A);
            T(i)=TT;
        end
    end
    TT=0;
end
WWv=0;
for h=1:N

```

```

        WWv=WWv+(T(h)-(W/N)).^2;
    end
    ct_r=0;
    for h=1:N
        if T(h)>ct_r
            ct_r=T(h);
        else
            ct_r=ct_r;
        end
    end
    ct(q)=ct_r;
    Wv(q)=WWv/N; % คำนวณค่าความแปรปรวนของภาระงาน
    Leff(q)=100*W/(ct_r*N); % คำนวณประสิทธิภาพของสายงานการประกอบ
    q=q+1;
end
[best_Wv,index]=min(Wv) % หาค่าความแปรปรวนของภาระงานที่น้อยที่สุด
B_Idle_Time=ct(index) % หาค่าเวลาว่างงานรวมที่น้อยที่สุด
B_Wv=best_Wv % ให้ B_Wv คือค่าความแปรปรวนของภาระงานน้อยที่สุด
B_Leff=Leff(index) % ให้ B_Leff คือค่าประสิทธิภาพของสายงานมากที่สุด
B_StationTask=Ps(index,:) % ให้ B_StationTask คือจำนวนชิ้นงานของแต่ละ
% สถานีงานที่เหมาะสมที่สุด
B_TaskSeq=Ts(index,:) % ให้ B_TaskSeq คือลำดับของชิ้นงานที่เหมาะสมที่สุด
time % แสดงค่าเวลาที่ใช้ในการค้นหา
round=0:1:max_CT; % พล็อตค่าความแปรปรวนของภาระงานที่ดีที่สุดที่พบ
plot(round,Wv,'k') % ในแต่ละรอบการค้นหา
xlabel('search round');ylabel('wv')
hold on
return
%-----%

```

ก.2 โปรแกรม TS (TS.m)

```

%-----%
%
%                               TS Program (TS.m) %
%
%                               by Supaporn Suwannarongsri %
%
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007 %
%-----%

```

```

function [loc_CT,loc_b_ne,loc_b_er,loc_b_TS,time]=TS(N,max_CT,xlim)
% สร้างฟังก์ชัน TS เป็นโปรแกรมย่อย
% เพื่อค้นหาจำนวนชิ้นงานที่เหมาะสม
% ให้กับแต่ละสถานีงาน
% อินพุตของโปรแกรมย่อยนี้คือ
% N (จำนวนสถานีงานสูงสุด)
% max_CT (จำนวนรอบการค้นหาสูงสุด)
% และ xlim (ปริภูมิการค้นหา)
% เอาต์พุตของโปรแกรมย่อยนี้คือ
% loc_CT (รอบการค้นหาที่พบคำตอบวงแคบเฉพาะถิ่น)
% loc_b_ne (จำนวนชิ้นงานที่เหมาะสมแต่ละสถานีงาน)
% loc_b_er (ค่าความแปรปรวนของภาระงานที่น้อยที่สุด)
% loc_b_TS (ลำดับของชิ้นงานที่เหมาะสม)
% time (เวลาที่ใช้ในการค้นหา)
b_er=1.0474e4; % กำหนดค่าความแปรปรวนของภาระงานเริ่มต้น
S0=[9 9 8 9 9 7 8 1]; % กำหนดจำนวนชิ้นงานเริ่มต้นให้กับแต่ละสถานีงาน
TS_b0=[1 2 49 5 55 3 4 51 52 53 54 6 7 20 21 29 34 50 8 9
36 10 11 12 15 17 13 14 16 19 38 40 18 42 31 22 56 24 57 23
26 28 30 58 25 32 59 27 44 60 33 35 37 39 41 43 45 46 47 48];
% กำหนดลำดับของชิ้นงานเริ่มต้น
b_ne=S0; % ให้ b_ne=S0
radius=0.2; % กำหนดรัศมีการค้นหา
N_ne=30; % กำหนดจำนวนค่าใกล้เคียง
t=0;
CT=0; % กำหนดรอบการค้นหาเริ่มต้น
tic; % เริ่มต้นคำนวณเวลาการค้นหา
t=t+1;
disp([CT b_er]); % แสดงค่า CT และ b_er
loc_CT(t,1)=CT; % เก็บค่า CT ไว้ใน loc_CT
loc_b_ne(t,1:N)=b_ne; % เก็บค่า b_ne ไว้ใน loc_b_ne
loc_b_er(t,1)=b_er; % เก็บค่า b_er ไว้ใน loc_b_er
loc_b_TS(t,1:We)= TS_b0; % เก็บค่า TS_b0 ไว้ใน loc_b_TS
for CT=1:max_CT % วนรอบการค้นหา
S1=round(RAN_ne(N_ne,radius,xlim,S0)); % เรียกใช้โปรแกรม RAN_ne เพื่อสุ่มค่าใกล้เคียง
% รอบคำตอบ S0 ภายในปริภูมิการค้นหาย่อยปัจจุบัน
% อินพุตของโปรแกรมนี้คือ N_ne, radius, xlim, และ S0

```

```

% เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ N_ne
[b_er,b_ne,TS_b0]=Pre_OBJ_ALB(S1,b_er,b_ne,TS_b0); % เรียกใช้โปรแกรม Pre_OBJ_ALB
% เพื่อประเมินค่าใกล้เคียง
% อินพุตของโปรแกรมนี้คือ S1, b_er, b_ne, และ TS_b0
% เอาต์พุตคือ b_er, b_ne, และ TS_b0

t=t+1;
loc_CT(t,1)=CT; % เก็บค่า CT ไว้ใน loc_CT
loc_b_ne(t,1:N)=b_ne; % เก็บค่า b_ne ไว้ใน loc_b_ne
loc_b_er(t,1)=b_er; % เก็บค่า b_er ไว้ใน loc_b_er
loc_b_TS(t,1:We)= TS_b0; % เก็บค่า TS_b0 ไว้ใน loc_b_TS
disp([CT b_er]); % แสดงค่า CT และ b_er
end
time=toc; % ยุติการคำนวณเวลาการค้นหา
return
%-----%

```

ก.3 โปรแกรม RAN_ne (RAN_ne.m)

```

%-----%
%
% Random Neighborhood Program (RAN_ne.m) %
%
% by Supaporn Suwannarongsri %
% Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007 %
%-----%
function S1=RAN_ne(N_ne,radius,xlim,S0) % สร้างฟังก์ชัน RAN_ne เป็นโปรแกรมย่อย
% เพื่อสุ่มค่าใกล้เคียงรอบคำตอบ S0
% อินพุตของโปรแกรมนี้คือ N_ne, radius, xlim, และ S0
% เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ N_ne

for u=1:N_ne
    for k=1:size(xlim,2)
        S1(u,k)=S0(1,k)+(radius*(xlim(1,k)-xlim(2,k))*RAN(-1,1));
% ทำการสุ่มค่าใกล้เคียงโดยอาศัยความสัมพัทธ์
%  $S_{h,i} = S_0 + [\text{radius} \times \text{rand}(-1, 1) \times (x_{h,i} - x_{l,i})]$ 
% เรียกใช้โปรแกรม RAN
% โดยให้ทำการสุ่มค่าในช่วง -1 ถึง 1
% นั่นหมายความว่า การสุ่มมีพื้นที่อยู่ในช่วง
%  $S_0 \pm [\text{radius} \times (x_{h,i} - x_{l,i})]$ 

        while ( S1(u,k)>xlim(1,k) | S1(u,k)<xlim(2,k) )

```

```

% ค่าใกล้เคียงที่สุ่มได้จะต้องอยู่ในขอบเขตบนและ
% ขอบเขตล่างของค่าพารามิเตอร์
% ถ้าค่าใกล้เคียงที่สุ่มได้ค่าใดมีค่าเกินขอบเขตบนและ
% ขอบเขตล่างที่กำหนดให้ทำการสุ่มค่าใกล้เคียงใหม่

S1(u,k)=S0(1,k)+(radius*(xlim(1,k)-xlim(2,k))*RAN(-1,1));

end

end

end

return

%-----%

```

ค.4 โปรแกรม RAN (RAN.m)

```

%-----%
%
% Random Program (RAN.m) %
%
% by Supaporn Suwannarongsri %
%
% Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007 %
%-----%

function x=RAN(a,b) % สร้างฟังก์ชัน RAN เป็นโปรแกรมย่อย
% เพื่อสุ่มค่าพารามิเตอร์ในช่วงที่กำหนด
% อินพุตของโปรแกรมคือขอบเขตบนของพารามิเตอร์ (b)
% และขอบเขตล่างของพารามิเตอร์ (a)

x=a+rand*(b-a); % โดยอาศัยความสัมพันธ์ (b-a)rand+a
% เอาต์พุตของโปรแกรมคือผลของการสุ่มค่าพารามิเตอร์
% ในช่วง a ถึง b

return %

%-----%

```

ค.5 โปรแกรม Pre_OBJ_ALB (Pre_OBJ_ALB.m)

```

%-----%
%
% Pre-ALB Objective Program (Pre_OBJ_ALB.m) %
%
% by Supaporn Suwannarongsri %
%
% Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007 %
%-----%

function [b_er,b_ne,TS_b0]=Pre_OBJ_ALB(S1,b_er,b_ne,TS_b0)
% สร้างฟังก์ชัน Pre_OBJ_ALB เป็นโปรแกรมย่อย
% เพื่อส่งผ่านคำตอบที่ได้จากกระบวนการค้นหา

```

```

% อินพุตของโปรแกรมนี้คือ S1, b_er, b_ne, และ TS_b0
% เอาต์พุตของโปรแกรมนี้คือ b_er, b_ne, และ TS_b0
% กำหนดตัวแปร er ไว้รองรับค่าฟังก์ชันวัตถุประสงค์
% ของค่าใกล้เคียง
for k=1:size(S1,1)
    [y,seq]=OBJ_NEW(S1(k,:));
    er(k,1)=y;
    TS1(k,:)=seq;
end
[b_er1,index]=min(er);
b_ne1=S1(index,:);
TS_b1=TS1(index,:);
if b_er1<b_er
    b_er=b_er1;
    b_ne=b_ne1;
    TS_b0=TS_b1;
else
    b_ne=b_ne;
    TS_b0=TS_b0;
end
return
%-----%

```

ก.6 โปรแกรม OBJ_NEW (OBJ_NEW.m)

```

%-----%
%
%                               NEW Objective Program (OBJ_NEW.m)
%                               by Supaporn Suwannarongsri
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007
%-----%
function [y,seq]=OBJ_NEW(x)
% ฟังก์ชัน OBJ_NEW เป็นโปรแกรมย่อย
% ทำหน้าที่คัดเลือกจำนวนชิ้นงานของแต่ละสถานีงาน
for k=1:size(x,1)
    if sum(x(k,:))=We
        pn=x(1:length(x));
        [y,seq]=OBJ_ALB(pn,TaskSeq);
    else
        % ถ้าผลรวมของจำนวนชิ้นงานเท่ากับ We
        % ให้ pn คือจำนวนชิ้นงานของแต่ละสถานีงาน
        % เรียกใช้โปรแกรม OBJ_ALB
        % ถ้าผลรวมของจำนวนชิ้นงานไม่เท่ากับ We
    end
end

```

```

y = 10000;
seq=zeros(1,We);
end
end
return
%-----%

```

ค.7 โปรแกรม OBJ_ALB (OBJ_ALB.m)

```

%-----%
%
%                               ALB Objective Program (OBJ_ALB.m)                               %
%                               by Supaporn Suwannarongsri                               %
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007   %
%-----%
function [WWv,Sseq]=OBJ_ALB(pn,TaskSeq)
% ฟังก์ชัน OBJ_ALB คือ โปรแกรมย่อย
% ทำหน้าที่คำนวณค่าความแปรปรวนของภาระงาน
% ซึ่งกำหนดให้เป็นฟังก์ชันวัตถุประสงค์
% อินพุตของโปรแกรมย่อยนี้คือ
% pn, และ TaskSeq
% เอาต์พุตของโปรแกรมย่อยนี้คือ WWv และ Sseq

for rr=1:length(TaskSeq)
    p_work=TaskSeq(rr,:);
    u0=1;
    for u=1:length(pn)
        for v=u0:u0+pn(u)-1
            ST_task(u,v-u0+1)=p_work(v);
            % จัดเรียงลำดับของงานเข้าสู่สถานีงาน
        end
        u0=u0+pn(u);
    end
end
count_chk=0;
chk_st=0;
for i=1:size(ST_task,1)
    for j=1:size(ST_task,2)
        A=ST_task(i,j);
        if A~=0
            count_chk=count_chk+1;
            for k=1:size(Prec,2)
                % ตรวจสอบเงื่อนไขลำดับความสัมพันธ์ก่อนหลัง
            end
        end
    end
end

```



```

% เงื่อนไขลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน
Wv(rr)=10000; % ให้ค่าความแปรปรวนของภาระงานเท่ากับ 10,000
seq(rr,:)=zeros(1,We); % และให้ลำดับของชิ้นงานเป็นเวกเตอร์ศูนย์
end
end
[b_Wv,index]=min(Wv); % ค้นหาค่าความแปรปรวนของภาระงานที่น้อยที่สุด
b_TSQ1=seq(index,:); % ค้นหาลำดับของชิ้นงานที่เหมาะสมที่สุด
WWv=b_Wv; % เก็บค่าความแปรปรวนของภาระงานที่น้อยที่สุด
% ไว้ใน b_Wv
Sseq=b_TSQ1; % เก็บลำดับของชิ้นงานที่เหมาะสมที่สุดไว้ใน Sseq
return
%-----%

```

ค.8 โปรแกรม Station_Task (Sta_Seq.m)

```

%-----%
%                               Station Task Program (Sta_Seq.m)                               %
%                               by Supaporn Suwannarongsri                               %
%   Department of Industrial Engineering, King Mongkut's Institute of Technology Ladkrabang, 2007   %
%-----%
function Station_task=Sta_Seq(pn,p_work,N) % ฟังก์ชัน Sta_Seq คือ โปรแกรมย่อย
% ทำหน้าที่จัดเรียงลำดับของชิ้นงานเข้าสู่สถานีนงาน
% อินพุตของโปรแกรมย่อยนี้คือ
% pn, p_work, และ N
% เอาต์พุตของโปรแกรมย่อยนี้คือ Station_task
u0=1;
for u=1:N
    for v=u0:u0+pn(u)-1
        Station_task(u,v-u0+1)=p_work(v); % จัดเรียงลำดับของชิ้นงานเข้าสู่แต่ละสถานีนงาน
    end
    u0=u0+pn(u);
end
return
%-----%

```

ภาคผนวก ง

รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์

ผลงานวิจัยวิทยานิพนธ์ที่ผ่านมาได้รับการจัดทำเป็นบทความผลงานวิจัยเพื่อตีพิมพ์เผยแพร่ในการประชุมวิชาการจำนวน 3 บทความ โดยแบ่งเป็นการประชุมวิชาการระดับชาติจำนวน 1 บทความ และการประชุมวิชาการระดับนานาชาติจำนวน 2 บทความ รายชื่อบทความผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่มีรายละเอียดดังนี้

ง.1 การประชุมวิชาการระดับชาติ

- [1] สุภาภรณ์ สุวรรณรังษี, สรรพสิทธิ์ ลิ้มนรรรัตน์, เดชา พวงดาวเรือง, “การแก้ปัญหาการจัดสมดุลสายงานการประกอบ โดยวิธี TSGA”, การประชุมวิชาการช่างงานวิศวกรรมอุตสาหกรรม (IE-Network Conference 2007), 2550, หน้า 485-490.

ง.2 การประชุมวิชาการระดับนานาชาติ

- [1] Suwannarongsri S., Limnararat S., Puangdownreong D. “A Hybrid Tabu Search Method for Assembly Line Balancing”, **Proceeding of the 7th WSEAS International Conference on Simulation, Modelling and Optimization (SMO'07)**, Beijing, China, 2007, pp. 443-448.
- [2] Suwannarongsri S., Limnararat S., Puangdownreong D. “A New Hybrid Intelligent Method for Assembly Line Balancing”, **Proceeding of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM2007)**, Singapore, 2007, pp. 1115-1119.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวสุภาภรณ์ สุวรรณรังษี
วัน เดือน ปีเกิด	5 กุมภาพันธ์ 2526 ที่กรุงเทพมหานคร
ที่อยู่	91/3 หมู่ 1 ถนนเลียบคลองทวีวัฒนา แขวงทวีวัฒนา เขตทวีวัฒนา กรุงเทพฯ 10170 โทร 086-605-3750
ประวัติการศึกษา	2546 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ มหาวิทยาลัยธนบุรี
ความชำนาญเฉพาะด้าน	- การแก้ปัญหาการจัดสมดุลสายงานการประกอบ - การประยุกต์เทคนิคการวิจัยเชิงปฏิบัติการ - การประยุกต์ AI เพื่อแก้ปัญหาทางวิศวกรรมอุตสาหการ
ประสบการณ์การทำงานและผลงานวิจัย	
พ.ศ.2547-2549	ตำแหน่งอาจารย์ประจำสาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธนบุรี
พ.ศ.2549-ปัจจุบัน	ตำแหน่งอาจารย์ประจำสาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเอเชียอาคเนย์ - ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่ [1] การแก้ปัญหาการจัดสมดุลสายงานการประกอบโดยวิธี TSGA, IE-Network2007, (by Thesis and Research) [2] A Hybrid Tabu Search Method for Assembly Line Balancing, WSEAS (SMO'07), (by Thesis and Research). [3] A New Hybrid Intelligent Method for Assembly Line Balancing, IEEE (IEEM2007), (by Thesis and Research). [4] Balancing of U-Shape Assembly Lines using Tabu Search Method, ECTI-CON2008. [5] Multi-Objective Assembly Line Balancing via Adaptive Tabu Search Method with Partial Random Permutation Technique, IEEE (IEEM2008). [6] Optimal Balancing of Multi-Objective U-Shape Assembly Lines using the TSGA Method, IEEE (IEEM2008).