ขั้นตอนวิธีการค้นหาความบรรสานสำหรับปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์: ทฤษฎี การประยุกต์ใช้ และกรณีศึกษา

โดย
นายธนพัทธ์ ทองชั้น

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาการจัดการงานวิศวกรรม แผน ก แบบ ก 2 ปริญญามหาบัณฑิต
ภาควิชาวิศวกรรมอุตสาหการและการจัดการ
บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร
ปีการศึกษา 2560
ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ขั้นตอนวิธีการค้นหาความบรรสานสำหรับปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์: ทฤษฎี การประยุกต์ใช้ และกรณีศึกษา

โดย
นายธนพัทธ์ ทองชั้น

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาการจัดการงานวิศวกรรม แผน ก แบบ ก 2 ปริญญามหาบัณฑิต
ภาควิชาวิศวกรรมอุตสาหการและการจัดการ
บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร
ปีการศึกษา 2560
ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

HARMONY SEARCH ALGORITHM FOR MULTI-OBJECTIVE OPTIMIZATION PROBLEM: THEORY, APPLICATION AND CASE STUDIES

By

MR. Tanapat TONGCHAN

A Thesis Submitted in partial Fulfillment of Requirements

for Master of Engineering (ENGINEERING MANAGEMENT)

Department of INDUSTRIAL ENGINEERING AND MANAGEMENT

Graduate School, Silpakorn University

Academic Year 2017

Copyright of Graduate School, Silpakorn University

หัวข้อ	ขั้นตอนวิธีการค้นหาความบรรสานสำหรับปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์: ทฤษฎี การประยุกต์ใช้ และกรณีศึกษา
โดย	ธนพัทธ์ ทองชั้น
สาขาวิชา	การจัดการงานวิศวกรรม แผน ก แบบ ก 2 ปริญญามหาบัณฑิต
อาจารย์ที่ปรึกษาหลัก	ผู้ช่วยศาสตราจารย์ ดร. ชูศักดิ์ พรสิงห์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร ได้รับพิจารณาอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร.ปานใจ ธารทัศนวงศ์)

พิจารณาเห็นชอบโดย

..... ประธานกรรมการ

(อาจารย์ ดร. คเณศ พันธุ์สวาสดิ์)

..... อาจารย์ที่ปรึกษาหลัก

(ผู้ช่วยศาสตราจารย์ ดร. ชูศักดิ์ พรสิงห์)

..... ผู้ทรงคุณวุฒิภายนอก

(ผู้ช่วยศาสตราจารย์ ดร. ทองแท่ง ทองลิ่ม)

58405201 : การจัดการงานวิศวกรรม แผน ก แบบ ก 2 ปริญญามหาบัณฑิต
คำสำคัญ : เมตาฮิวริสติก, ค่าเหมาะที่สุด, การค้นหาความบรรสาน

นาย ธนพัทธ์ ทองชั้น: ขั้นตอนวิธีการค้นหาความบรรสานสำหรับปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์: ทฤษฎี การประยุกต์ใช้ และกรณีศึกษา อาจารย์ที่ปรึกษาวิทยานิพนธ์ : ผู้ช่วยศาสตราจารย์ ดร. ชูศักดิ์ พรสิงห์

การค้นหาความบรรสานเป็นหนึ่งในขั้นตอนวิธีเมตาฮิวริสติกที่ถูกพัฒนาขึ้นจากพฤติกรรมของนักดนตรีที่พยายามค้นหาเสียงที่ไพเราะที่สุดสำหรับการแสดง โดยความสามารถที่สำคัญอย่างหนึ่งของขั้นตอนวิธีเมตาฮิวริสติก คือการหาคำตอบที่เหมาะสมในเวลาที่เหมาะที่สุด แต่อย่างไรก็ตามการจะได้มาซึ่งความสามารถดังกล่าวจำเป็นที่จะต้องมีการดำเนินการอย่างเหมาะสม ดังนั้นผู้วิจัยจึงศึกษาขั้นตอนวิธีการค้นหาความบรรสานกับปัญหาการเดินทางของพนักงานขายและการวางแผนการประกอบแผงวงจรอิเล็คทรอนิกส์ อีกทั้งแสดงแนวทางการหาคำตอบเหมาะที่สุด

เริ่มแรกทำการทดสอบการค้นหาความบรรสานกับปัญหาการเดินทางของพนักงานขายที่เป็นปัญหาลักษณะ NP-complete โดยเลือกชุดปัญหา UK12 WG59 และ SGB128 มาใช้สำหรับการทดสอบ จากนั้นดำเนินการสร้างวิธีการแปลงรหัส โดยทำการการสุ่มค่าตามจำนวนเมืองที่นำมาทดสอบ ซึ่งค่าที่ถูกสุ่มขึ้นจะถูกแทนรหัสเป็นเมืองแต่ละเมือง จากนั้นนำมาจัดลำดับจากน้อยไปหามาก เพื่อใช้เป็นค่าเริ่มต้นสำหรับการค้นหาความบรรสาน ต่อมาทางผู้วิจัยได้ดำเนินการหาพารามิเตอร์ที่เหมาะสมสำหรับการทดสอบในครั้งนี้ ด้วยวิธีการทางสถิติ พบว่าพารามิเตอร์ความน่าจะเป็นการใช้หน่วยความจำ (HMCR) และจำนวนการทำซ้ำ (T) ส่งผลต่อคำตอบอย่างมีนัยสำคัญ อีกทั้งพบว่าเมื่อจำนวนการทำซ้ำที่เพิ่มขึ้นมากเกินไปจะไม่ส่งผลในการพัฒนาคำตอบ

ต่อมาทำการทดสอบการค้นหาความบรรสานกับการวางแผนการประกอบแผงวงจรอิเล็คทรอนิกส์ซึ่งเป็นปัญหาหลายวัตถุประสงค์ โดยที่นี้จากผู้วิจัยได้ใช้ชุดข้อมูลที่มีจำนวนชิ้นส่วนอิเล็คทรอนิกส์ทั้งหมด 71 ชิ้นส่วน อีกทั้งนำลักษณะปัญหาการเดินทางของพนักงานขายมาประยุกต์กำหนดให้ชิ้นส่วนอิเล็คทรอนิกส์แต่ละชนิดแทนเมืองแต่ละเมืองที่จะต้องเดินทาง จากนั้นทำการสร้างวิธีการแปลงรหัส ซึ่งในที่นี้จะใช้วิธีการเดียวกับปัญหาการเดินทางของพนักงานขาย จากนั้นทำการทดสอบหาพารามิเตอร์ที่ส่งผลต่อคำตอบด้วยวิธีการทางสถิติ พบว่ามีเพียงพารามิเตอร์ความน่าจะเป็นการใช้หน่วยความจำ (HMCR) และจำนวนการทำซ้ำ (T) ที่ส่งผลต่อคำตอบเช่นเดียวกับการทดสอบปัญหาการเดินทางของพนักงานขาย แต่อย่างไรก็ตามเนื่องจากการวางแผนการประกอบแผงวงจรอิเล็คทรอนิกส์เป็นปัญหาหลายวัตถุประสงค์ การสร้างคำตอบจะสร้างออกมาในลักษณะพาเรโต้ฟร้อน

สำหรับการค้นหาความบรรสานในครั้งนี้ พบว่าเมื่อทำการทดสอบ 20,000 ครั้งจะมีคำตอบอยู่บนพาเรโต้ฟร้อน 5 คำตอบ หรือคิดเป็น 0.025 เปอร์เซ็น

58405201 : Major (ENGINEERING MANAGEMENT)
Keyword : meta-heuristic, optimization, Harmony search

MR. Tanapat TONGCHAN: Harmony Search Algorithm for Multi-objective Optimization Problem: Theory, Application and Case Studies Thesis advisor : Assistant Professor Choosak Pornsing, Ph.D.

Harmony search (HS) is one of the metaheuristic algorithms, which is developed from the behavior of musicians who trying to find the best aesthetic sound. The important attribute to metaheuristic is good solutions in a reasonable computational time. However, if we want a good answer, we have to find a proper conditions. Therefore, we studied the Harmony Search algorithm for traveling salesman problems (TSP) and Harmony Search algorithm for systematic planning for Surface Mount PCB Assembly.

In the first part, Harmony Search algorithm was applied to travelling salesman problem (TSP). TSP which is an NP-complete problem. It was solved by selecting the benchmark data sets (UK12, WG59 and SGB128) for the evaluations. Then, the encoding and decoding schemes are presented which is based on the amount of cities tested. Random values are represented by city codes. After that, the value is rearranged in ascending order. Subsequently, we has to find a parameter setting. In this study, we made use of statistical methods to find the appropriate parameter setting for travelling salesman problem. We found that the harmony memory consideration rate (HMCR) and iteration number (T) affect to the quality of results. In addition, we also confirmed that the large iteration number would not affect the development of the result.

In the second part, Harmony Search algorithm was applied to systematic planning for the Surface Mount PCB Assembly. The problem was formulated as multi-objective problem. We used a set of data with 71 electronic components. Each electronic component is applied to each city as same as traveling salesman problem. Then, the encoding and decoding methods are the same as we used in part 1. The

parameter setting was found by using statistical methods. We found that the harmony memory consideration rate (HMCR) and iteration number (T) parameters had affected to the response, with the same results as shown in the first part. Furthermore, due to the planning of the Surface Mount PCB Assembly is a multi-objective problem, we have shown the performance of Harmony Search algorithm in order to construct Pareto front line. We showed that within 20,000 replicates it yields 5 solutions on the Pareto front. It accounted to 0.025 percent

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับคำแนะนำและคำปรึกษารวมถึงการอนุเคราะห์ในเรื่องวิธีการดำเนินงาน ตลอดจนแนวทางในการจัดทำวิทยานิพนธ์นี้เป็นอย่างดี จากท่านอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.ชูศักดิ์ พรสิงห์ ที่ควบคุมการดำเนินงานศึกษา และจัดทำวิทยานิพนธ์ในครั้งนี้ รวมทั้งคณาจารย์ที่ทำการตรวจสอบ และชี้แนะข้อบกพร่องเพื่อความสมบูรณ์ถูกต้องของวิทยานิพนธ์อันประกอบด้วย อาจารย์ ดร.คเณศ พันธุ์สวาสดิ์ และผู้ช่วยศาสตราจารย์ ดร.ทองแท่ง ทองลิ่ม

ขอขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้การสนับสนุนเป็นอย่างดีเสมอมาสุดท้ายนี้ใคร่ขอขอบพระคุณอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ตั้งแต่ต้นจนถึงปัจจุบัน รวมทั้งบุคคลท่านอื่นๆ ที่มิได้กล่าวถึง ที่ให้ความช่วยเหลือในการจัดทำวิทยานิพนธ์นี้ จนสำเร็จลุล่วงได้ด้วยดี

ธนพัทธ์ ทองชั้น

มหาวิทยาลัยศิลปากร

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... ฉ
กิตติกรรมประกาศ ..... ซ
สารบัญ ..... ฌ
สารบัญตาราง ..... ฎ
สารบัญรูปภาพ ..... ฐ
บทที่ 1 บทนำ 5
1.1 ความเป็นมาและความสำคัญของปัญหา 5
1.2 วัตถุประสงค์ของการศึกษา 7
1.3 กรอบแนวการศึกษา 7
1.4 ประโยชน์ที่คาดว่าจะได้รับ 7
1.5 นิยามคำศัพท์ 8
บทที่ 2 ทบทวนทฤษฎีและงานวิจัยที่เกี่ยวข้อง 9
2.1 รูปแบบทางคณิตศาสตร์ 9
2.2 ค่าเหมาะที่สุดวัตถุประสงค์เดียว 11
2.3 ค่าเหมาะที่สุดหลายวัตถุประสงค์ 12
2.4 การค้นหาความบรรสาน 20
2.4.1 ประวัติความเป็นมาของการค้นหาความบรรสาน 20
2.4.2 การพัฒนาการค้นหาความบรรสาน 21
2.4.3 การเพิ่มขึ้นของความสนใจการใช้งานของการค้นหาความบรรสาน 23
2.4.4 การประยุกต์ใช้ที่มีการนำการค้นหาความบรรสานไปใช้งาน 24

2.2.5 หลักการของการค้นหาความบรรสานแบบดั้งเดิม 28
2.2.6 การใช้งานวิธีค้นหาความบรรสานแบบดั้งเดิมสำหรับการทดสอบฟังก์ชั่น 32
2.2.7 การปรับปรุงการค้นหาความบรรสาน 33
2.2.8 การค้นหาความบรรสานหลายวัตถุประสงค์ 35
บทที่ 3 วิธีดำเนินการวิจัย 38
3.1 การศึกษางานวิจัย ทฤษฎี หลักการและแนวคิดพื้นฐานที่เกี่ยวข้อง 38
3.2 ขั้นตอนออกแบบและพัฒนาการค้นหาความบรรสาน 39
3.3 การทดสอบกับปัญหาการเดินทางของพนักงานขาย และการวางแผนการประกอบชิ้นส่วนอุปกรณ์อิเล็คทรอนิกส์ 39
3.4 สรุปผลงานวิจัยและจัดทำรูปเล่มวิทยานิพนธ์ 39
บทที่ 4 การค้นหาความบรรสานสำหรับปัญหาการเดินทางของพนังานขาย 40
4.1 ลักษณะความเป็นมาของปัญหาการเดินทางของพนักงานขาย 40
4.2 การค้นหาความบรรสาน สำหรับปัญหาการเดินทางของพนักงานขาย 41
4.2.1 การค้นหาความบรรสาน 41
4.2.2 การค้นหาความบรรสานสำหรับปัญหาการเดินทางของพนักงานขาย 45
4.3 การหาค่าพารามิเตอร์ที่เหมาะสม 46
4.3.1 การตั้งค่าพารามิเตอร์การค้นหาความบรรสาน 46
4.3.2 การปรับแต่งพารามิเตอร์การค้นหาความบรรสาน 46
4.4 สรุปผลการทดลอง 52
บทที่ 5 การค้นหาความบรรสานสำหรับระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ 53
5.1 ลักษณะความเป็นมาของระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ 53
5.2 การค้นหาความบรรสาน สำหรับการวางแผนการประกอบวงจรพิมพ์ 54
5.2.1 การค้นหาความบรรสาน 54
5.2.3 รูปแบบของฟังก์ชั่นหลายวัตถุประสงค์ 58

5.2.4 การสร้างรูปแบบระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ เข้ากับปัญหาการเดินทางของพนักงานขาย 60
5.3 การหาค่าพารามิเตอร์ที่เหมาะสม 61
5.4 ผลการทดลองและการวิเคราะห์ผลการทดลอง 62
5.4.1 การทดสอบประสิทธิภาพการหาคำตอบของการค้นหาความบรรสาน 62
5.4.2 ผลของพารามิเตอร์สำหรับการทดลอง 64
5.4 สรุปผลการทดลอง 69
บทที่ 6 สรุปผลการวิจัย 70
6.1 สรุปผลการวิจัย 70
6.2 ปัญหาและอุปสรรคที่พบในการวิจัย 72
6.3 ข้อเสนอแนะ 72
ภาคผนวก 73
ภาคผนวก ก การพัฒนาตนเอง 74
รายการอ้างอิง 77
ประวัติผู้เขียน 82

มหาวิทยาลัยศิลปากร

สารบัญตาราง

ตารางที่ 1 ข้อมูลการผลิตผลิตภัณฑ์และกำไรต่อหน่วยของผลิตภัณฑ์ 10
ตารางที่ 2 การปรับตั้งพารามิเตอร์ 46
ตารางที่ 3 ค่า p-value จากการทดสอบด้วยการค้นหาความบรรสาน สำหรับชุดปัญหา UK12 47
ตารางที่ 4 ค่า p-value จากการทดสอบด้วยการค้นหาความบรรสาน สำหรับชุดปัญหา WG59 47
ตารางที่ 5 ค่า p-value จากการทดสอบด้วยการค้นหาความบรรสาน สำหรับชุดปัญหา SGB128 .. 48
ตารางที่ 6 แสดงเวลาการทดสอบ 50
ตารางที่ 7 ผลการวิเคราะห์ความแปรปรวน (ANOVA) สำหรับปัญหา WG59 50
ตารางที่ 8 การปรับตั้งพารามิเตอร์ 62
ตารางที่ 9 ค่า P-Value จากการทดสอบด้วยการค้นหาความบรรสาน สำหรับชุดข้อมูล BOM..... 64
ตารางที่ 10 แสดงผลการวิเคราะห์ความแปรปรวน (ANOVA) สำหรับการวางแผนประกอบแผงวงจรอิเล็คทรอนิกส์ 65

สารบัญรูปภาพ

รูปที่ 1 ขั้นตอนการแก้ไขปัญหา 9
รูปที่ 2 การหาคำตอบด้วยวิธีการกราฟ 11
รูปที่ 3 การครอบงำแบบพาเรโต (Pareto Dominance) คำตอบแบบพาเรโต้ (Pareto solution). 14
รูปที่ 4 ลักษณะของการครอบงำ 15
รูปที่ 5 วิธีหาค่าเหมาะที่สุดของปัญหาหลายวัตถุประสงค์ 18
รูปที่ 6 ตัวอย่างในการปรับปรุงและการหาค่าเหมาะที่สุด 21
รูปที่ 7 การเติบโตของการค้นหาความบรรสาน 22
รูปที่ 8 การเพิ่มขึ้นเพิ่มขึ้นของความสนใจการใช้งานของการค้นหาความบรรสาน 23
รูปที่ 9 การประยุกต์ใช้งานการค้นหาความบรรสาน 24
รูปที่ 10 ขั้นตอนการค้นหาความบรรสาน 31
รูปที่ 11 ฟังก์ชั่น Rosenbrock 32
รูปที่ 12 การทำงานของขั้นตอนวิธีการค้นหาความบรรสาน 44
รูปที่ 13 แผนภาพการเข้ารหัส (Encoding) และถอดรหัส (Decoding) 45
รูปที่ 14 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดปัญหาทดสอบ UK12 48
รูปที่ 15 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดปัญหาทดสอบ WG59 49
รูปที่ 16 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดปัญหาทดสอบ SBG128 49
รูปที่ 17 ผลการทดสอบด้วยวิธีตูกี กับจำนวนการทำซ้ำที่แตกต่างกัน 51
รูปที่ 18 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดปัญหาทดสอบ WG59 51
รูปที่ 19 แผนภาพการทำงานของอัลกอริทึมการค้นหาความบรรสาน สำหรับระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ 56
รูปที่ 20 แผนผังตำแหน่งเครื่อง Fuji FCP-IV 58
รูปที่ 21 การจัดวางส่วนประกอบอิเล็กทรอนิกส์ 60

รูปที่ 22 การเข้ารหัส (Encoding) และถอดรหัส (Decoding)..... 61

รูปที่ 23 ประสิทธิภาพการทำงานของการค้นหาความบรรสาน สำหรับการประกอบ แผงวงจรอิเล็คทรอนิกส์ 63

รูปที่ 24 การเคลื่อนที่ในการประกอบชิ้นส่วนอิเล็คทรอนิกส์..... 63

รูปที่ 25 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดข้อมูล BOM 65

รูปที่ 26 ผลการทดสอบด้วยวิธีตูกี กับจำนวนการทำซ้ำที่แตกต่างกัน 66

รูปที่ 27 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดข้อมูล BOM 67

รูปที่ 28 คำตอบของปัญหาการประกอบแผงวงจรอิเล็คทรอนิกส์ 68

มหาวิทยาลัยศิลปากร

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การหาค่าเหมาะที่สุด (Optimization) นั้นมีอยู่ทั่วไปทุกหนแห่ง โดยธรรมชาติของมนุษย์นั้นพยายามที่จะค้นหาทางเลือกที่ดีที่สุดอยู่เสมอ ในปีคริสตศักราช 1629 ได้มีนักคณิตศาสตร์ได้ทำการคิดค้นหลักการของเฟอร์แมท (Fermat's Principle) ขึ้นมา โดยเกี่ยวกับเส้นการเดินทางของคลื่นจากจุดหนึ่งไปยังจุดหนึ่งในแนวตามขวางที่มีระยะสั้นที่สุด ซึ่งจากการคิดค้นดังกล่าว ถือเป็นส่วนหนึ่งที่ทำให้เกิดการพัฒนาเรื่อยมา จนมาถึงในช่วงศตวรรษที่ 19 ซึ่งถือได้ว่าเป็นศตวรรษแห่งการเปลี่ยนแปลงไม่ว่าจะเป็นในด้านการเมือง เศรษฐกิจ และสังคม เช่น ในปีคริสตศักราช 1940 ได้มีการเกิดขึ้นของสิ่งประดิษฐ์ที่เรียกว่า คอมพิวเตอร์ ขึ้นมาจากการเปลี่ยนแปลงและพัฒนาในช่วงศตวรรษที่ 19 ซึ่งการเปลี่ยนแปลงดังกล่าวนั้นส่งผลต่อในด้านอุตสาหกรรม และการศึกษา โดยทางคณิตศาสตร์มีการค้นพบวิธีอย่างง่ายสำหรับโปรแกรมเส้นตรง (Simplex Method for Linear Programming) จากการค้นพบดังกล่าว ทำให้เกิดการพัฒนาการทางด้านโปรแกรมทางด้านคณิตศาสตร์ (Mathematical Programming) และทฤษฎีการควบคุมค่าเหมาะที่สุด (Optimal Control) ในปีคริสตศักราช 1950 หลังจากนั้นก็ยังคงมีการพัฒนาเรื่อยมา จนปัจจุบันศาสตร์ในการหาค่าเหมาะที่สุดเป็นสาขาวิชาที่หลากหลายสามารถใช้ได้ในหลากหลายวงการ [1]

การหาค่าเหมาะที่สุดวัตถุประสงค์เดียว (Single-objective Optimization) เป็นการค้นหาค่าเหมาะที่สุดเพื่อตอบสนองต่อวัตถุประสงค์เพียงวัตถุประสงค์เดียว ซึ่งจะได้คำตอบที่ดีที่สุดเพียงคำตอบเดียว โดยสามารถเรียกคำตอบนั้นได้ว่า ค่าเหมาะที่สุดแบบวงกว้าง (Global Optimum) โดยทั่วไปตัวอย่างของปัญหาวัตถุประสงค์เดียวมีหลากหลายปัญหา เช่น การออกแบบผลิตภัณฑ์ให้มีต้นทุนน้อยที่สุด แต่อย่างไรก็ตามปัญหาดังกล่าวยังคงไม่ครอบคลุมสำหรับการทำงานในปัจจุบัน ดังนั้นจึงมีการหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (Multi-objective Optimization: MOO)

การหาค่าเหมาะที่สุดหลายวัตถุประสงค์เกิดขึ้นในช่วงศตวรรษที่ 19 จากเอ็ดเวิดท์ (Francis Ysidro Edgeworth) และพาเรโต (Villfredo Pareto) ซึ่งทั้งสองคนเป็นนักเศรษฐศาสตร์และนักคณิตศาสตร์ โดยแรกเริ่มได้มีการนำหลักการหาค่าเหมาะที่สุดหลายวัตถุประสงค์มาใช้ในด้านเศรษฐศาสตร์และวิทยาศาสตร์การจัดการ ต่อมาจึงได้เริ่มมีการนำมาใช้ในด้านการจัดการงานวิศวกรรม โดยรูปแบบของปัญหาการหาค่าเหมาะที่สุดที่มีหลายวัตถุประสงค์ (Multi-objective Optimization Problem: MOP) โดยวัตถุประสงค์แต่ละวัตถุประสงค์นั้นมีความขัดแย้งกัน หรือเมื่อมีวัตถุประสงค์ค่าหนึ่งเพิ่มขึ้น วัตถุประสงค์อีกค่าหนึ่งจะต้องลดลง ตัวอย่างเช่น อุตสาหกรรมการผลิต

สินค้านั้นต้องการต้นทุนที่ถูก แต่จะต้องได้คุณภาพของสินค้าที่ดี ซึ่งการหาค่าเหมาะที่สุดที่มีหลายวัตถุประสงค์ นั้นสามารถที่จะแก้ปัญหาข้างต้นได้ แต่อย่างไรก็ตามเมื่อมีจำนวนวัตถุประสงค์เพิ่มมากขึ้น จะทำให้เวลาในการค้นหาค่าเหมาะที่สุดนั้นใช้เวลาค่อนข้างมาก ดังนั้นในทศวรรษที่ 1980 จึงมีการพัฒนาการแก้ไขปัญหาที่มีหลายวัตถุประสงค์ โดยการนำขั้นตอนวิธีเมตาฮิวริสติกเข้ามาช่วยแก้ไขปัญหา

ขั้นตอนวิธีเมตาฮิวริสติก (Meta-heuristic Algorithm) แนวคิดนี้เกิดมาจากการเลียนแบบปรากฏการณ์ทางธรรมชาติทั้งที่เป็นไปตามแบบแผน (Rules) และไม่เป็นไปตามแบบแผน (Randomness) โดยจะแบ่งการเกิดวิธีแบบเมตาฮิวริสติกออกเป็น 3 ลักษณะ คือ การวิวัฒนาการทางชีววิทยา (Biological Evolutionary Process) เช่น ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm: GA) พฤติกรรมของสัตว์ (Animal Behavior) เช่น อาณาจักรมด (Ant Colony Optimization) และกระบวนการทางกายภาพทั่วไป (Physical Process) เช่น การจำลองการอบอ่อน (Simulated Annealing: SA) ซึ่งขั้นตอนวิธีการที่แตกต่างกันจะมีลักษณะในการค้นหาค่าเหมาะที่สุดแตกต่างกัน แต่อย่างไรก็ตามถึงแม้ว่าขั้นตอนวิธีการดังกล่าวจะมีลักษณะที่แตกต่างกัน แต่เมื่อเปรียบเทียบความสามารถในการนำไปใช้ค้นหาค่าเหมาะที่สุดวัตถุประสงค์เดียวและหลายวัตถุประสงค์ กับวิธีการหาค่าเหมาะที่สุดแบบดั้งเดิม (Conventional Method) เช่น กำหนดการเชิงเส้น (Linear programming) พบว่าขั้นตอนวิธีการเมตาฮิวริสติกนั้นมีประสิทธิภาพที่ดีกว่าวิธีการดังกล่าว แต่อย่างไรก็ตามในปัญหาที่มีความซับซ้อนเป็นอย่างมาก ขั้นตอนวิธีแบบเมตาฮิวริสติกบางขั้นตอนที่นำมาใช้ค้นหาค่าเหมาะที่สุด จะไม่สามารถพบค่าที่เหมาะที่สุดแบบวงกว้างได้ (Global Optimum) แต่จะพบเพียงแต่ค่าเหมาะที่สุดแบบวงแคบ (Local Optimum) ดังนั้นการศึกษาความสามารถในการค้นหาค่าเหมาะที่สุดของขั้นตอนวิธีการเมตาฮิวริสติกชนิดต่าง ๆ นั้นมีความสำคัญอย่างมาก

ในปีคริสตศักราช 2001 หรือประมาณ 15 ปีที่ผ่านมา ขั้นตอนวิธีเมตาฮิวริสติกชนิดใหม่ที่มีชื่อว่า การค้นหาความบรรสาน (Harmony Search: HS) ได้ถูกคิดค้นขึ้นโดย Zong Woo Geem ซึ่งเกิดจากความสนใจในทางดนตรี และมีความต้องการหาเสียงที่ไพเราะที่สุดในสถานะนั้น โดยเป็นการเลียนแบบพฤติกรรมของนักดนตรีที่จะทำการปรับปรุงและแก้ไขตัวโน้ตของเครื่องดนตรีที่เล่น เพื่อการหาเสียงของโน้ตดนตรีนั้นให้เหมาะสมที่สุดในการเล่นประสานกับเครื่องดนตรีอื่น อีกทั้งยังต้องการให้มีระดับเสียงที่สอดคล้องกันมากที่สุด แต่อย่างไรก็ตามจากการทบทวนวรรณกรรมในช่วง 15 ปีที่การค้นหาความบรรสานนั้นได้ถือกำเนิดขึ้นมา พบว่าการประยุกต์ใช้กับปัญหาการหาค่าเหมาะที่สุดที่มีหลายวัตถุประสงค์ยังไม่แพร่หลาย เมื่อเทียบกับขั้นตอนวิธีเมตาฮิวริสติกชนิดอื่น

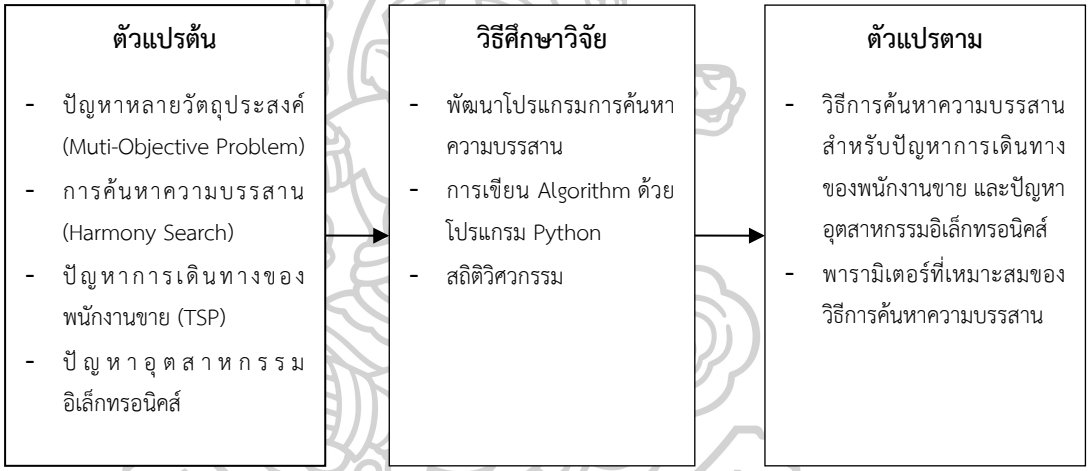
ดังนั้นในการศึกษาครั้งนี้จึงมีวัตถุประสงค์ที่จะพัฒนาวิธีการค้นหาความบรรสาน สำหรับการหาค่าเหมาะที่สุดที่มีหลายวัตถุประสงค์ เพื่อให้การค้นหาค่าเหมาะที่สุดเป็นไปในแบบวงกว้าง (Global Optimum) นอกจากนี้วิธีการที่คิดค้นใหม่นี้จะถูกนำไปประยุกต์ใช้กับปัญหาทางอุตสาหกรรมในเชิงกรณีศึกษาอีกด้วย

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อประยุกต์ใช้วิธีการค้นหาความบรรสานกับปัญหาการเดินทางของพนักงานขาย

1.2.2 เพื่อประยุกต์ใช้วิธีการค้นหาความบรรสานกับปัญหาหลายวัตถุประสงค์ กรณีศึกษาอุตสาหกรรมอิเล็คทรอนิกส์

1.3 กรอบแนวการศึกษา



1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 เกิดองค์ความรู้ในการหาค่าเหมาะที่สุดสำหรับปัญหาหลายวัตถุประสงค์ (MOP) ด้วยการค้นหาความบรรสาน (HS)

1.4.2 เป็นต้นแบบในการใช้การค้นหาความบรรสาน (HS) เพื่อหาค่าเหมาะที่สุดในอุตสาหกรรม

1.5 นิยามคำศัพท์

1.5.1 ค่าเหมาะที่สุด (Optimization) หมายถึง วิธีหาคำตอบที่ดี ในเวลาที่เหมาะสม “good solution in a reasonable time” ซึ่งมีความเหมาะสมสำหรับปัญหาบางปัญหาเท่านั้น

1.5.2 ขั้นตอนวิธีเมตาฮิวริสติก (Meta-heuristic Algorithm) หมายถึง วิธีที่ได้จากการพัฒนาและปรับปรุงวิธีฮิวริสติกให้มีความสามารถในการหาค่าเหมาะที่สุดของปัญหาที่ซับซ้อนและมีตัวแปรตัดสินใจจำนวนมาก ได้อย่างมีประสิทธิภาพและภายในเวลาที่เหมาะสม

1.5.3 ค่าเหมาะที่สุดวงกว้าง (Global Optimal) หมายถึง ตำแหน่งที่ไม่มีคำตอบอื่นในปริภูมิการค้นหาที่เป็นไปได้ ซึ่งให้ค่าฟังก์ชันจุดประสงค์ต่ำกว่าค่าฟังก์ชันจุดประสงค์ที่หาได้ (กรณีปัญหาการหาค่าต่ำที่สุด)

1.5.4 ค่าเหมาะที่สุดเฉพาะที่ (Local Optimal) หมายถึง ตำแหน่งที่ไม่มีผลเฉลยอื่นในบริเวณข้างเคียง (Neighborhood) ซึ่งจะให้ค่าฟังก์ชั่นจุดประสงค์ต่ำกว่าค่าฟังก์ชั่นจุดประสงค์ที่หาได้ (กรณีปัญหาการหาค่าต่ำที่สุด)

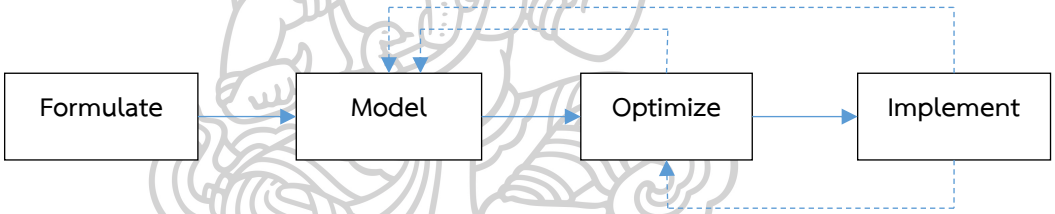
มหาวิทยาลัยศิลปากร

บทที่ 2
ทบทวนทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การศึกษาการหาค่าเหมาะที่สุดหลายวัตถุประสงค์โดยใช้แนวคิดการค้นหาความบรรสาน (Harmony Search: HS) เพื่อเป็นแนวทางในการศึกษาปัญหาทางด้านวิศวกรรม ทางผู้วิจัยได้ทำการศึกษาค้นคว้าและทำการรวบรวมเอกสารเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวข้อง เพื่อนำมากำหนดเป็นกรอบและแนวทางสำหรับการวิจัย โดยมีเนื้อหาที่เกี่ยวข้องในการศึกษา ดังนี้

2.1 รูปแบบทางคณิตศาสตร์

ในงานทางด้านวิศวกรรมโดยทั่วไปเมื่อพบปัญหาที่เกิดขึ้นภายในโรงงาน ไม่ว่าจะเป็นปัญหาการผลิตสินค้าหลากหลายประเภท การออกแบบผังโรงงานและอื่น ๆ รูปแบบทางคณิตศาสตร์เป็นวิธีหนึ่งที่สามารถที่สามารถช่วยแก้ไขปัญหาได้ โดยมีขั้นตอนการตัดสินใจดำเนินการ คือ กำหนดปัญหา กำหนดรูปแบบที่เหมาะสม กำหนดวิธีแก้ไขปัญหาที่เหมาะสม และสุดท้ายคือการได้มาซึ่งคำตอบที่เหมาะสม แสดงดังรูปที่ 1



รูปที่ 1 ขั้นตอนการแก้ไขปัญหา [2]

ขั้นตอนแรกเริ่มจากการกำหนดปัญหา (Formulate) ซึ่งเป็นขั้นตอนการกำหนดปัญหาโดยพิจารณาปัจจัยภายนอก ปัจจัยภายใน และวัตถุประสงค์ของปัญหา ขั้นตอนต่อมาซึ่งถือว่าสำคัญมากขั้นตอนหนึ่ง คือขั้นตอนการสร้างรูปแบบที่เหมาะสม (Model) โดยในขั้นตอนนี้จะมีการสร้างรูปแบบทางคณิตศาสตร์ที่สอดคล้องกับปัญหาที่ถูกกำหนดขึ้น ซึ่งในการสร้างรูปแบบให้เหมาะกับปัญหาค่อนข้างเป็นเรื่องที่ยากลำบาก ดังนั้นจำเป็นที่ต้องดำเนินการในขั้นตอนนี้อย่างละเอียดรอบคอบ หลังจากปัญหาได้ถูกกำหนดรูปแบบที่เหมาะสมแล้ว ต่อมาจะเข้าสู่ขั้นตอนการหาค่าเหมาะที่สุด (Optimize) โดยขั้นตอนนี้เปรียบเสมือนการเลือกหาวิธีที่จะใช้แก้ไขปัญหา แต่ละรูปแบบจะมีวิธีแก้ไขปัญหาที่เหมาะสมแตกต่างกันไป จึงจำเป็นที่จะต้องเลือกวิธีให้เหมาะสมกับรูปแบบที่กำหนด ต่อมาเมื่อทำการกำหนดปัญหา โดยสร้างรูปแบบที่เหมาะสม และการหาค่าเหมาะที่สุด จะเข้าสู่ขั้นตอนสุดท้ายคือขั้นตอนการนำค่าเหมาะที่สุดไปใช้ (Implement)ขั้นตอนนี้จะเป็นการนำค่าเหมาะที่สุดมา

แสดงผล หรือนำมาใช้งาน เช่น จากปัญหาการขนส่งพบว่าจำนวนรอบในการขนส่งที่เหมาะที่สุดเท่ากับ 5 รอบ ดังนั้นในปัญหาการขนส่งนี้จะมีการขนส่งจำนวน 5 รอบ

แต่อย่างไรก็ตามอาจจะมีบางกรณีที่การนำค่าเหมาะที่สุดไปใช้งาน พบว่าไม่ตรงตามความต้องการจริง ซึ่งอาจจะเกิดจากการเลือกวิธีการแก้ปัญหาที่ไม่ถูกต้อง หรือกำหนดรูปแบบนั้นไม่เหมาะสมกับปัญหา จึงจำเป็นที่ต้องมีการเลือกวิธีการ หรือกำหนดรูปแบบใหม่ ให้สอดคล้องกับปัญหา เพื่อความเข้าใจในขั้นตอนการแก้ไขปัญหา ทางผู้วิจัยจึงทำการยกตัวอย่างดังนี้

ให้บริษัทหนึ่งมีผลิตภัณฑ์จำนวน 2 ผลิตภัณฑ์ ผลิตภัณฑ์ที่ 1 ($Prod_1$) และ ผลิตภัณฑ์ที่ 2 ($Prod_2$) โดยใช้วัตถุดิบ 2 ชนิดในการผลิต วัตถุดิบที่ 1 (M_1) วัตถุดิบที่ 2 (M_2) โดยมีวัตถุประสงค์ที่จะใช้ส่วนผสมให้คุ้มค่ามากที่สุด ตารางที่ 1 แสดงข้อมูลการใช้วัตถุดิบในการผลิตผลิตภัณฑ์และกำไรต่อหน่วยของผลิตภัณฑ์แต่ละชนิด

ตารางที่ 1 ข้อมูลการผลิตผลิตภัณฑ์และกำไรต่อหน่วยของผลิตภัณฑ์

	จำนวนที่ใช้สำหรับ $Prod_1$	จำนวนที่ใช้สำหรับ $Prod_2$	วัตถุดิบที่ใช้ได้
M_1	6	4	24
M_2	1	2	6
กำไรต่อหน่วย (ยูโร)	€5	€4	

ขั้นตอนแรกกำหนดปัญหา คือการต้องการกำไรสูงสุด $\text{Max profit} = 5X + 4Y$

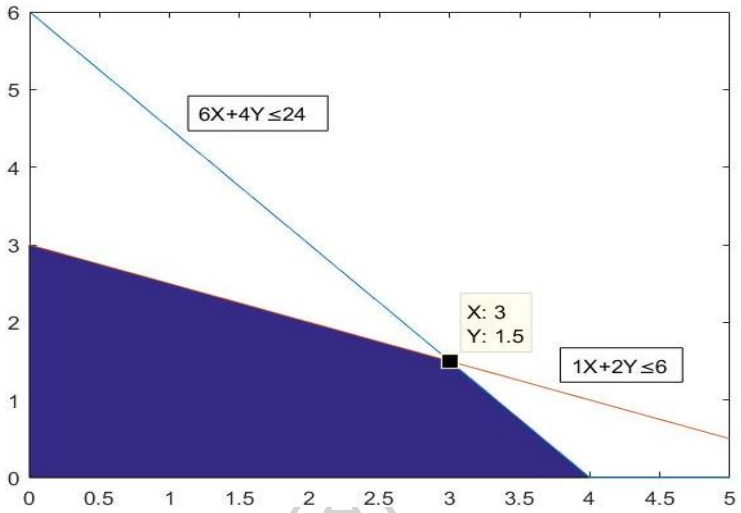
โดยมีเงื่อนไขดังนี้ คือ

$6X + 4Y \leq 24$

$1X + 2Y \leq 6$

$X, Y \geq 0$

ต่อมาทำการปรับรูปแบบปัญหาให้เป็นสมการทางคณิตศาสตร์ ซึ่งในตัวอย่างนี้เป็น รูปแบบโปรแกรมเชิงเส้น (Linear programming model) ซึ่งใช้วิธีการกราฟ (Graphical Method) เพื่อทำการหาค่าเหมาะที่สุด โดยค่าเหมาะที่สุด คือ $x_1 = 3$ $x_2 = 1.5$ ซึ่งจะได้กำไร เท่ากับ 21 ยูโร



รูปที่ 2 การหาคำตอบด้วยวิธีการกราฟ

จากปัญหาข้างต้นเป็นปัญหาที่ไม่ซับซ้อน ดังนั้นในการกำหนดรูปแบบ เลือกใช้วิธีการหาคำตอบที่สามารถหาคำตอบได้อย่างแม่นยำและใช้เวลาได้อย่างรวดเร็วสามารถทำได้ง่ายกว่าปัญหาที่มีความซับซ้อนมากกว่า เช่น ปัญหาที่มีจำนวนฟังก์ชั่นหลายวัตถุประสงค์ หรือ ปัญหาคอมบิเนทอเรียล (Combinatorial Problem)

แต่อย่างไรก็ตามเพื่อให้ง่ายต่อการศึกษาต่อของผู้ที่สนใจ งานวิจัยนี้ทางผู้วิจัยได้ทำการแบ่งประเภทของปัญหาเป็น 2 ประเภท คือ 1) ปัญหาค่าเหมาะที่สุดวัตถุประสงค์เดียว (Single Objective Optimization Problem) และ 2) ปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (Multi Objective Optimization Problem) โดยแต่ละประเภทของปัญหาจะมีรูปแบบของปัญหาที่แตกต่างกัน คือ ปัญหาค่าเหมาะที่สุดวัตถุประสงค์เดียวจะมุ่งเน้นไปในการหาค่าของสิ่งที่สนใจเพียงอย่างเดียว เช่น หากำไรที่มากที่สุด หรือ หาต้นทุนที่น้อยที่สุด ส่วนปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์จะมีการหาค่าที่สนใจหลายวัตถุประสงค์หลายอย่างไปพร้อมกัน เช่น การสร้างตึกโดยที่คานมีน้ำหนักน้อยที่สุดแต่สามารถรับภาระงานได้มากที่สุด

2.2 ค่าเหมาะที่สุดวัตถุประสงค์เดียว

ค่าเหมาะที่สุดวัตถุประสงค์เดียวนั้นเป็นการแสดงคำตอบออกมาสำหรับวัตถุประสงค์เพียงด้านเดียวเท่านั้น เช่น การแสดงต้นทุนที่น้อยที่สุด หรือ กำไรที่มากที่สุด โดยรูปแบบทั่วไปของปัญหาค่าเหมาะที่สุดวัตถุประสงค์เดียว (Single Objective Optimization Problem) คือ

$$f(x) = (x_1, \dots, x_n)$$

โดยมีสมการข้อจำกัด $g_i(x) \leq 0\,, i = \{1, \dots, m\}$ และ $h_j(x) = 0\,, j = \{1, \dots, p\}$

สุภาภรณ์ พวงชมภู [3] ได้นำเสนอปัญหาในการหาส่วนผสมปุ๋ยหมักจากฟางข้าว ที่มีวัตถุประสงค์ทำให้ต้นทุนในการผลิตต่ำที่สุดโดยใช้ส่วนประกอบได้แก่ ปริมาณฟางข้าว (x_1) ปริมาณกากถั่วเหลือง (x_2) ปริมาณมูลสัตว์ (x_3) และปริมาณมูลค้างคาว (x_4) โดยปุ๋ยที่ผลิตได้จะต้องผ่านมาตรฐาน ซึ่งพบว่าจะใช้ปริมาณฟางข้าว 0.6245 กรัม ปริมาณกากถั่วเหลือง 0.0534 กรัม ปริมาณมูลสัตว์ 0.0940 กรัม และปริมาณมูลค้างคาว 0.2280 กรัม เป็นปริมาณที่ให้ราคาต้นทุนต่ำที่สุดเท่ากับ 4.63 บาทต่อกิโลกรัม

แต่อย่างไรก็ตามในความเป็นจริงนั้นเห็นได้ว่าการผลิตสินค้าที่มีต้นทุนต่ำสุดจำเป็นที่ต้องได้สินค้าที่มีคุณภาพสูงที่สุดด้วยถึงจะสามารถแข่งขันในตลาดได้ ดังนั้นในการหาค่าเหมาะที่สุดเพียงแค่วัตถุประสงค์เดียวนั้นไม่เพียงพอต่อความต้องการในความเป็นจริง ดังนั้น การหาค่าเหมาะที่สุดหลายวัตถุประสงค์จึงได้รับความสนใจอย่างแพร่หลายมากขึ้น

2.3 ค่าเหมาะที่สุดหลายวัตถุประสงค์

ค่าเหมาะที่สุดหลายวัตถุประสงค์นั้นเติบโตมาจาก 3 แนวคิดหลัก 1. เศรษฐศาสตร์ดุลยภาพและทฤษฎีสวัสดิการ (Economic equilibrium and Welfare theories) 2. ทฤษฎีเกม (Game theory) และ 3. คณิตศาสตร์บริสุทธิ์ (Pure Mathematics) กรอบแนวคิดสำคัญของการหาค่าเหมาะที่สุดหลายวัตถุประสงค์ ได้แก่ การครอบงำ (Dominance), กลุ่มคำตอบที่เหมาะสมแบบพาเรโต้ (Pareto optimality), เซตของกลุ่มคำตอบที่เหมาะสมแบบพาเรโต้ (Pareto optimal set) และ พาเรโตฟร้อน (Pareto front) โดยนิยามดังกล่าวข้างต้น เป็นสิ่งที่จำเป็นในการค้นหาคำตอบของค่าเหมาะที่สุดหลายวัตถุประสงค์ [4]

บทนิยามที่ 1 ปัญหาการหาค่าเหมาะที่สุดหลายวัตถุประสงค์

ปัญหาการหาค่าเหมาะที่สุดหลายวัตถุประสงค์กำหนดได้ดังสมการ

$$MOP = \begin{cases} min\, F(x)=(f_1(x),f_2(x),\ldots,f_n(x)) \\ s.c.\ x \in S \end{cases}$$

โดยที่ n $(n \geq 2)$ คือจำนวนของวัตถุประสงค์ $x = (x_1, \ldots, x_k)$ คือ เวกเตอร์ที่แสดงตัวแปรตัดสินใจ (decision variables) และ S แสดงการประยุกต์ใช้ของคำตอบ (set of feasible solutions) ที่มีข้อจำกัดชนิดเท่ากัน และข้อจำกัดชนิดไม่เท่ากัน โดยที่ $F(x) = (f_1(x), f_2(x), \ldots, f_n(x))$ คือเวกเตอร์ของวัตถุประสงค์ที่หาค่าเหมาะที่สุด

บทนิยามที่ 2 การครอบงำแบบพาเรโต้ (Pareto Dominance)

เวกเตอร์วัตถุประสงค์ $u = (u_1, \dots, u_n)$ ที่ครอบงำ $v = (v_1, \dots, v_n)$ (แสดงโดย $u \prec v$) ก็ต่อเมื่อไม่มีส่วนประกอบของ v ที่มีค่าน้อยกว่า u และอย่างน้อยหนึ่งกรณี ส่วนประกอบของ u ต้องมีค่าน้อยกว่า v นั่นคือ $\forall i \in \{1, \dots, n\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, n\}: u_i \leq v_i$

บทนิยามที่ 3 กลุ่มคำตอบเหมาะที่สุดแบบพาเรโต้ (Pareto Optimality)

กำหนดให้ $x^* \in S$ คือ กลุ่มคำตอบเหมาะที่สุด (Pareto optimal) ถ้าสำหรับทุก$x \in S$ และ $F(x)$ ไม่ทำการครอบงำ $F(x^*)$ นั่นคือ $F(x) \nprec F(x^*)$

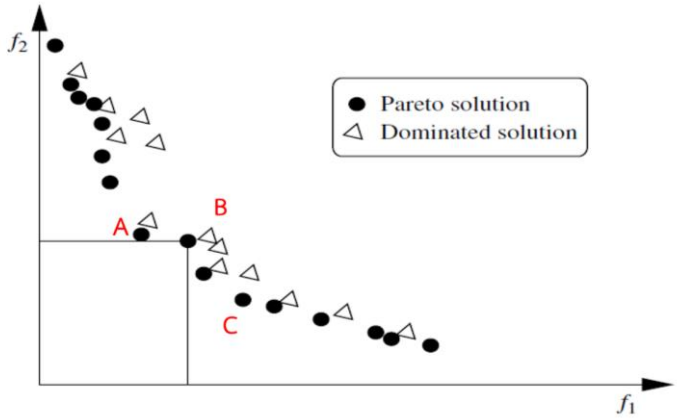
บทนิยามที่ 4 เซตของกลุ่มคำตอบเหมาะที่สุดแบบพาเรโต้ (Pareto Optimal Set)

กำหนดให้ ปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (MOP) (F,S) เซตของกลุ่มคำตอบที่เหมาะสม (Pareto optimal set) ถูกกำหนดให้เป็น $\mathcal{P}^* = \{x \in S / \nexists x' \in S, F(x') \prec F(x)\}$

บทนิยามที่ 5 พาเรโตฟร้อน (Pareto front)

กำหนดให้ ปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (MOP) (F,S) และเซตของกลุ่มคำตอบที่เหมาะสม (Pareto optimal set) คือ $\mathcal{P}^*$ พาเรโต้ฟร้อน (Pareto Front) ถูกกำหนดให้เป็น $\mathcal{P}F^* = \{F(x), x \in \mathcal{P}^*\}$

เพื่อให้เข้าใจมากขึ้นในบทนิยามที่ 2 3 4 และ 5 จึงขอแสดงดังรูปที่ 3 เมื่อพิจารณาจุด (•) ที่ตำแหน่ง A พบว่าค่าของ จุด (•) จะมีค่าน้อยกว่าหรือเท่ากับค่าของ จุด (△) ดังนั้นจึงเป็น การครอบงำแบบพาเรโต (Pareto Dominance) เมื่อพิจารณาตำแหน่ง B พบว่าค่าของ จุด (•) จะมีค่าน้อยกว่าค่าของ จุด (△) เพียงเท่านั้น ดังนั้นจึงเป็น กลุ่มคำตอบที่เหมาะที่สุดแบบพาเรโต้ (Pareto Optimal) และพิจารณาบริเวณแนวตำแหน่ง C พบว่า เซตของกลุ่มคำตอบที่เหมาะที่สุดแบบพาเรโต้ (Pareto Optimal) รวมตัวกันเรียกว่า พาเรโตฟร้อน (Pareto front)



รูปที่ 3 การครอบงำแบบพาเรโต (Pareto Dominance) คำตอบแบบพาเรโต้ (Pareto solution) [2]

บทนิยามที่ 6 เวกเตอร์อุดมคติ (Ideal Vector)

ที่จุด $y^* = (y_1^*, y_2^*, \dots, y_n^*)$ คือ ideal vector ถ้า เป็นค่าต่ำสุดของแต่ละฟังก์ชั่นวัตถุประสงค์ f_i ใน $F(x)$ นั้นคือ $y_i^* = min(f_i(x)), x \in S, i \in [1, n]$.

บทนิยามที่ 7 จุดอ้างอิง (Reference point)

ที่จุดอ้างอิง (Reference point) $z^* = [\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n]$ คือ vector ที่กำหนดระดับ (หรือเป้าหมาย) $\bar{z}_i$ จะไปยังแต่ละ ฟังก์ชั่นวัตถุประสงค์ f_i

บทนิยามที่ 8 จุดนาเดีย (Nadir point)

ที่จุด $y^* = (y_1^*, y_2^*, \dots, y_n^*)$ คือ nadir point ถ้าเป็นค่ามากสุดของแต่ละฟังก์ชั่นวัตถุประสงค์ f_iของ F ที่มากกว่า Pareto set นั่นคือ $y^* = \max\left(f_i(x)\right), x \in \mathcal{P}^*, i \in [1, n]$.

บทนิยามที่ 9 ฟังก์ชั่นอรรถประโยชน์ (Utility function)

ค่าฟังก์ชั่นอรรถประโยชน์ คือ v แสดงความพึงพอใจของผู้ตัดสินใจ , $v: \mathcal{R}^* \longrightarrow \mathcal{R}$

บทนิยามที่ 10 Locally Pareto optimal solution

วิธีแก้ปัญหา x คือ locally Pareto Optimal ก็ต่อเมื่อ $\forall w \in N(x)$,$F(w)$ ไม่ครอบงำ $F(x)$ และ $N(x)$ แสดงถึงบริเวณของวิธีแก้ปัญหา x

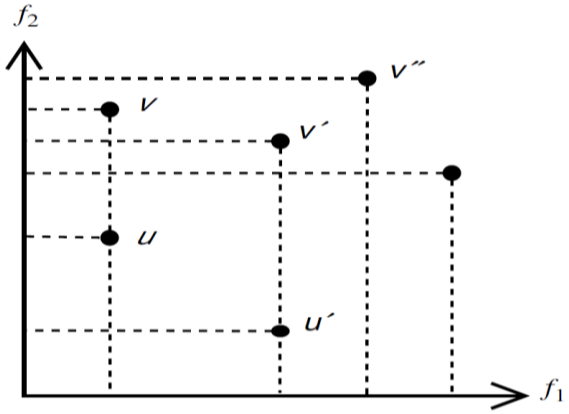
บทนิยามที่ 11 Weak Pareto Optimality

เวกเตอร์วัตถุประสงค์ $u = (u_1, \dots, u_n)$ กล่าวได้ว่า weakly dominate $v = (v_1, \dots, v_n)$ แสดงโดย $u \preccurlyeq v$ ถ้าองค์ประกอบของ u นั้นมีค่าน้อยกว่า หรือเท่ากับ v นั่นคือ $\forall i \in \{1, \dots, n\}, u_i \leq v_i$

บทนิยามที่ 12 Strict Pareto Optimality

เวกเตอร์วัตถุประสงค์ $u = (u_1, \dots, u_n)$ กล่าวได้ว่า strictly dominate $v = (v_1, \dots, v_n)$ แสดงโดย $u \prec\prec v$ ถ้าองค์ประกอบของ u มีค่าน้อยกว่า v นั่นคือ $\forall i \in \{1, \dots, n\}, u_i < v_i$

เพื่อความเข้าใจที่มากขึ้นในบทนิยามที่ 11 และ 12 ทางผู้วิจัยจึงแสดงดังรูปที่ 4 การที่ u เป็น strictly dominate ระหว่าง v' และ v'' เนื่องจากเมื่อพิจารณาในทะแยงมุม พบว่า ค่าของ u นั้นครอบงำ v' และ v'' และ u เป็น weakly dominate ของ v เนื่องจากเมื่อพิจารณา u ในแนวแกน f_2 นั้นพบว่ามีค่าครอบงำค่าของ v



รูปที่ 4 ลักษณะของการครอบงำ [2]

บทนิยามที่ 13 ϵ-Dominance

เวกเตอร์วัตถุประสงค์ $u = (u_1, \dots, u_n)$ กล่าวได้ว่า ϵ-Dominance $v = (v_1, \dots, v_n)$ แสดงโดย $u \prec_\epsilon v$ ก็ต่อเมื่อ ไม่มีองค์ประกอบของ v มีค่าน้อยกว่า $u - \epsilon$ และอย่างน้อย หนึ่งองค์ประกอบ ของ $u - \epsilon$ เป็น Strictly better นั่นคือ $\forall i \in \{1, \dots, n\} : u_i - \epsilon_i \leq v_i \wedge \exists i \in \{1, \dots, n\} : u_i - \epsilon_i < v_i$

จากบทนิยามดังกล่าว จะนำมาใช้ทำการพิสูจน์การแก้ไขปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์โดยรูปแบบทั่วไปของปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (Multi Objective Optimization Problem) คือ

$$F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{bmatrix}, F(x) = (f_1(x), \dots, f_k(x))$$

โดยมีสมการข้อจำกัด $g_i(x) \leq 0$, $i = \{1, \dots, m\}$ และ $h_j(x) = 0$, $j = \{1, \dots, p\}$

หลังจากการทำการระบุปัญหา และสร้างรูปแบบของปัญหาไม่ว่าจะเป็นปัญหาค่าเหมาะที่สุดวัตถุประสงค์เดียวหรือปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์ ในลำดับต่อมาจะทำการกำหนดวิธีการแก้ไขปัญหาเพื่อให้ได้ค่าเหมาะที่สุด โดยวิธีที่นิยมในการใช้แก้ไขปัญหาสามารถแบ่งออกได้เป็น 2 วิธี คือ 1) วิธีแม่นยำ (Exact Methods) และ 2) วิธีหาค่าโดยประมาณ (Approximate Methods) หรือวิธีฮิวริสติกส์ (Heuristic Methods)

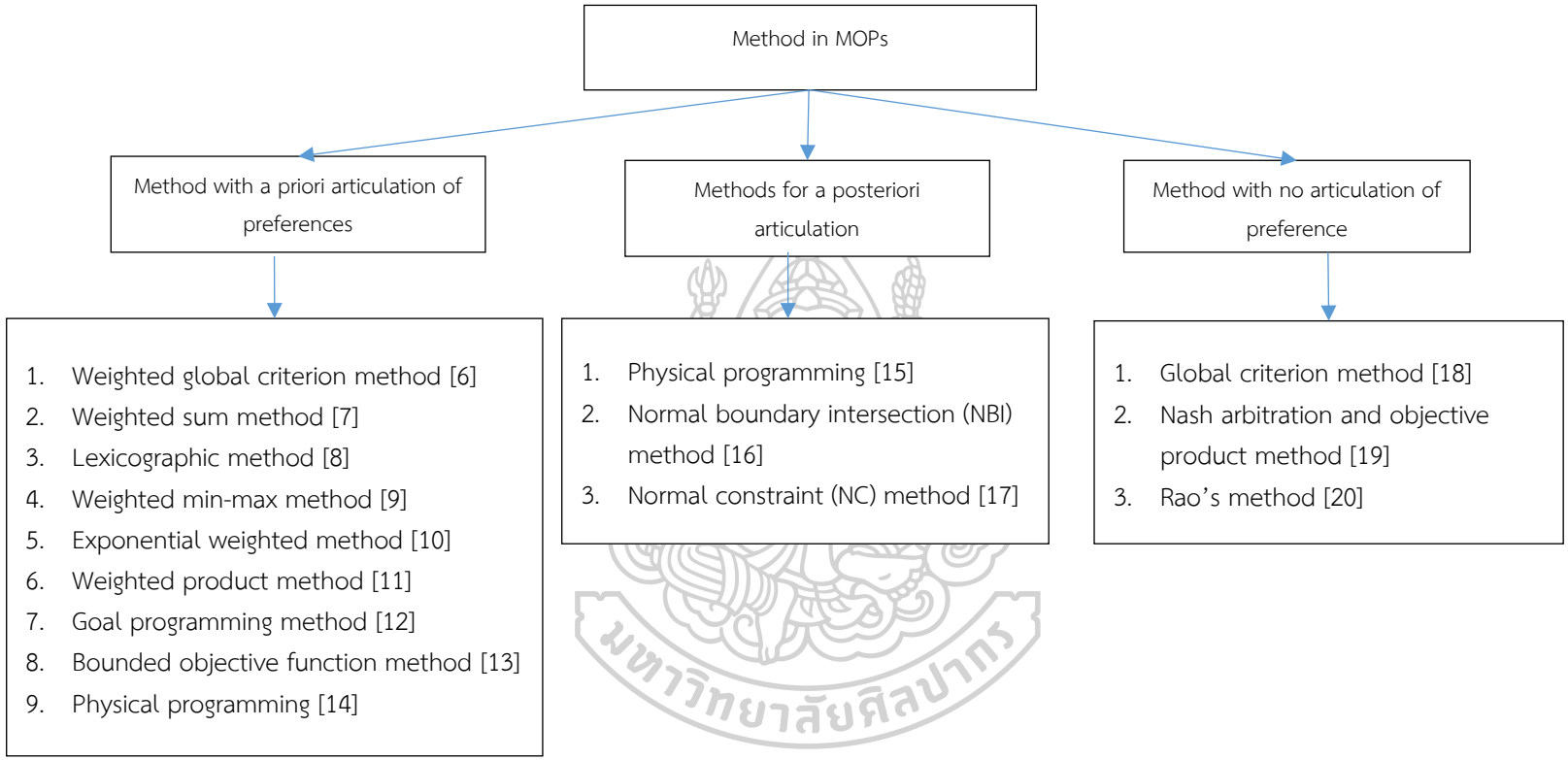
วิธีแม่นยำ (Exact Methods) เป็นวิธีที่สามารถรับประกันได้ว่าสามารถหาคำตอบเหมาะที่สุดจากปัญหาได้ เช่น วิธีขยายและจำกัดเขต (Branch and Bound) วิธีโปรแกรมเชิงพลวัต (Dynamic Programming) เป็นต้น โดยในที่นี้เพื่อให้สอดคล้องกับการกำหนดรูปแบบของปัญหา ทางผู้วิจัยได้กำหนดวิธีแก้ไขปัญหาดังนี้ วิธีแก้ไขปัญหาค่าเหมาะที่สุดวัตถุประสงค์เดียว (Single-Objective Optimization Methods: SOO Methods) และวิธีแก้ไขปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (Multi-Objective Optimization Methods: MOO Methods) แต่อย่างไรก็ตามในปัจจุบันปัญหาที่เกิดขึ้นนั้นมีหลากหลายวัตถุประสงค์ ดังนั้นทางผู้วิจัยจะไม่ขอกล่าวถึงวิธีการแก้ไขปัญหาค่าเหมาะที่สุดของปัญหาวัตถุประสงค์เดียว โดยถ้าหากมีผู้สนใจที่จะศึกษาสามารถศึกษาได้หนังสือด้านการหาค่าเหมาะที่สุด เช่น Linear and Nonlinear Optimization [4] เป็นต้น ลำดับต่อมาในส่วนของ

วิธีแก้ไขปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (Multi-Objective Optimization Method: MOO Methods) [5] ได้ทำการแบ่งกลุ่มวิธีการแก้ไขปัญหาออกเป็น 3 กลุ่ม ดังแสดงรูปที่ 5

กลุ่มแรก “Method with a priori articulation of preferences” คือ วิธีนี้จะอนุญาตให้ผู้ใช้งานกำหนดความชื่นชอบ อาจจะพิจารณาจากเป้าหมาย หรือความสัมพันธ์ที่สำคัญของวัตถุประสงค์ที่ต่างกัน ได้แก่ Weighted global criterion method [6] Weighted sum method [7] Lexicographic method [8] Weighted min-max method [9] Exponential weighted method [10] Weighted product method [11] Goal programming method [12] Bounded objective function method [13] Physical programming [14]Goal programming method [12]Bounded objective function method [13] และ Physical programming [14]

กลุ่มที่สอง “Methods for a posteriori articulation of preference” คือ ในบางกรณีเป็นเรื่องยากสำหรับผู้ตัดสินใจ (Decistion-maker) ที่จะบอกออกมาอย่างชัดเจนเกี่ยวกับฟังก์ชั่นที่ชื่นชอบ ดังนั้นเพื่อจะเพิ่มความสามารถให้ผู้ตัดสินใจเลือกจาก คำตอบที่ดีที่สุดของวิธีแก้ปัญหา ได้แก่ Physical programming [15] Normal boundary intersection (NBI) method [16] และ Normal constraint (NC) method [17]

กลุ่มที่สาม “Method with no articulation of preference” คือ บ่อยครั้งที่ผู้ตัดสินใจไม่สามารถกำหนดให้เป็นรูปธรรมได้ว่าจะเลือกให้ความสำคัญอะไรมากที่สุด จึงมีการพัฒนาวิธีมาจากกลุ่มแรก เพื่อช่วยในการแก้ปัญหาดังกล่าว ได้แก่ Global criterion method [18] Nash arbitration and objective product method [19] และ Rao’s method [20]



รูปที่ 5 วิธีหาค่าเหมาะที่สุดของปัญหาหลายวัตถุประสงค์

แต่อย่างไรก็ตาม เมื่อรูปแบบของปัญหามีความซับซ้อนและขนาดของปัญหาที่มากขึ้น การคำนวณหาคำตอบเหมาะที่สุดบางปัญหามีการใช้เวลาในการคำนวณนานถึง 2 เดือน หรือบางปัญหาอาจจะใช้เวลามากกว่า 1 ปีถึงจะได้คำตอบเหมาะที่สุดออกมา [2] จากปัญหาดังกล่าวจึงเห็นได้ว่าการใช้วิธีการแม่นยำ ไม่เหมาะสมอีกต่อไป ดังนั้นวิธีหาค่าโดยประมาณ (Approximate Methods) หรือวิธีฮิวริสติกส์ (Heuristic Methods) เริ่มเข้ามามีส่วนช่วยในการหาคำตอบของปัญหาดังกล่าว

วิธีฮิวริสติกส์ (Heuristics Methods) เป็นวิธีหาคำตอบที่ดี ในเวลาที่เหมาะสม “good solution in a reasonable time” แต่วิธีฮิวริสติกนั้นถูกสร้างขึ้นเพื่อหาคำตอบที่ดีเฉพาะปัญหาเท่านั้น เมื่อสามารถหาคำตอบที่ดีของปัญหาหนึ่งได้ อาจจะไม่สามารถนำไปใช้หาคำตอบที่ดีของปัญหาอื่นได้ Thomas และคณะ [21] ได้ทำการศึกษาการเปรียบเทียบระหว่างวิธีฮิวริสติกส์หลากหลายวิธีได้แก่ ERF EDF BLOCK และ Potts สำหรับการวางแผนงาน ซึ่งพบว่าวิธีฮิวริสติกแบบ Potts เหมาะสมสำหรับการทดสอบนี้มากที่สุดเมื่อเปรียบเทียบกับวิธีการฮิวริสติกแบบอื่น โดยสามารถหาคำตอบเหมาะที่สุดได้ในเวลา 10 วินาที จากบทความดังกล่าวข้างต้น จะเห็นได้ว่าถึงแม้ปัญหาในการทดสอบชนิดเดียวกับ แต่ใช้วิธีฮิวริสติกส์ที่แตกต่างกัน จะส่งผลสำหรับการหาคำตอบเหมาะที่สุด ด้วยเหตุนี้เองวิธีฮิวริสติกจึงไม่เหมาะสมในการแก้ไขปัญหาที่มีความหลากหลาย ดังนั้น เมตาฮิวริสติก (Meta-Heuristic) จึงเกิดการพัฒนาขึ้นมาเพื่อที่จะสามารถประยุกต์ในการแก้ไขปัญหาการหาค่าเหมาะที่สุดได้หลากหลายมากขึ้น

วิธีเมตาฮิวริสติก (Meta-Heuristics Methods) เป็นวิธีการหนึ่งที่เกิดขึ้นมาจากการสังเกตุพฤติกรรมของธรรมชาติ เช่น การศึกษาลักษณะการดำรงชีวิตของสัตว์ ทั้งนี้วิธีเมตาฮิวริสติกมีจุดประสงค์เพื่อที่จะสามารถหาคำตอบของปัญหาที่มีความซับซ้อนได้อย่างมีประสิทธิภาพ และใช้เวลาการหาคำตอบที่เหมาะสม ถึงแม้ว่าคำตอบที่ได้จะไม่ใช่คำตอบที่ดีที่สุดก็ตาม จากการศึกษาการประยุกต์ใช้ที่นำวิธีเมตาฮิวริสติกไปใช้งาน พบว่ามีความหลากหลายในการใช้งานอย่างมาก พีระวัฒน์ โชคณัติ และคณะ [22] ได้นำเสนอวิธีการเมตาฮิวริสติกสำหรับปัญหาการจัดเส้นทางยานพาหนะในการขนส่งสำหรับโรงงานผลิตน้ำดื่ม ด้วยวิธีการวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution Algorithm: DE) ซึ่งในการวิจัยได้แบ่งการทดสอบออกเป็น 2 กรณี คือ กรณีแรก มีลูกค้า 19 รายที่มีจุดกระจายสินค้าเพียงจุดเดียว กรณีที่สอง มีลูกค้า 73 รายที่มีจุดกระจายสินค้าเพียงจุดเดียวและมีเงื่อนไขความต้องการสินค้าของลูกค้าแต่ละรายไม่แน่นอน เมื่อทำการทดสอบระยะทางขนส่ง พบว่าวิธีการวิวัฒนาการโดยใช้ผลต่าง สามารถลดระยะทางจากกรณีแรกได้ 1 กิโลเมตรต่อวัน และสามารถลดระยะทางจากกรณีที่สองได้ถึง 34.8 กิโลเมตรต่อวัน

วิธีเมตาฮิวริสติกได้ถูกพัฒนาขึ้นมาอย่างต่อเนื่องตลอดระยะเวลาที่ผ่านมา เช่น ขั้นตอนวิธีการเชิงพันธุกรรม (Genetic Algorithm: GA) การหาค่าเหมาะสมที่สุดแบบอนุภาคกลุ่ม (Particle Swarm Optimization: PSO) และอื่น ๆ แต่อย่างไรก็ตามวิธีการที่ถูกพัฒนาขึ้น เมื่อนำไปหาคำตอบที่ต้องการพบว่ายังมีช่องว่างในการดำเนินการ จึงทำให้บางปัญหาไม่สามารถหาค่าเหมาะที่สุดได้ ดังนั้นงานวิจัยนี้ทางผู้วิจัยได้จึงได้สนใจทำการศึกษา วิธีเมตาฮิวริสติกที่ถือกำเนิดขึ้นมาในปี ค.ศ. 2001 หรือประมาณ 15 ปีที่ผ่านมา นั้นคือ การค้นหาความบรรสาน (Harmony Search) ซึ่งถือเป็นวิธีการเมตาฮิวริสติกที่เพิ่งเกิดขึ้นมาไม่นาน อีกทั้งยังมีช่องว่างของการนำไปใช้แก้ไขปัญหาการหาค่าเหมาะที่สุดหลายวัตถุประสงค์

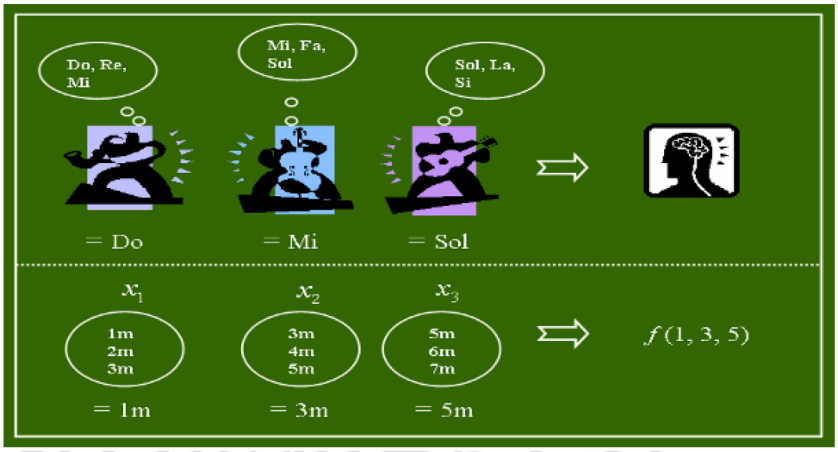
2.4 การค้นหาความบรรสาน

2.4.1 ประวัติความเป็นมาของการค้นหาความบรรสาน

เริ่มแรกในการพัฒนาเกิดขึ้นในระหว่างการศึกษาในระดับปริญญาเอกของ Zong Woo Geem ที่มหาวิทยาลัยเกาหลี โดยได้พยายามศึกษาเกี่ยวกับการออกแบบการขนถ่ายน้ำ (Design of water distribution network) [23] ซึ่งหัวข้อในการศึกษานี้ได้กลายเป็นจุดเริ่มต้นในการคิดค้น เมตาฮิวริสติกตัวใหม่ที่มีชื่อว่า “การค้นหาความบรรสาน (Hamony Search)” ขึ้นมา

การค้นหาความบรรสานมีแนวคิดมาจาก ความต้องการเลียนแบบพฤติกรรมของนักดนตรีที่ต้องการแก้ไขปัญหาในการประสานเสียงของเครื่องดนตรี ซึ่งนักดนตรีจะทำการปรับปรุงและแก้ไขตัวโน้ต เพื่อทำให้เกิดการประสานเสียงที่ไพเราะมากที่สุด โดยจะสามารถแสดงพฤติกรรมออกมาได้ 3 รูปแบบ คือ 1) นักดนตรีจะเลือกตัวโน้ตซึ่งเป็นเสียงการบรรเลงไพเราะที่สุดจากความทรงจำได้อย่างแม่นยำ 2) นักดนตรีจะสามารถเลือกบรรเลงตัวโน้ตที่เคยเล่นมาก่อนหน้านี้จากความทรงจำได้ แต่มีการเปลี่ยนแปลงของระดับเสียงเล็กน้อย 3) บรรเลงทำนองเพลงรูปแบบใหม่จากการสุ่มตัวโน้ตขึ้นมา โดยวิธีค้นหาความบรรสานได้ทำการปรับปรุงขึ้นจากรูปแบบดังกล่าว ซึ่งมีขั้นตอนดังนี้ 1) การใช้งานหน่วยความจำของฮาร์โมนี (Harmony memory: HM) 2) การปรับเสียง (Pitch adjustment) และ 3) การสุ่มค่า (Randomization) [24]

ต่อมาทางผู้วิจัยได้นำเสนอตัวอย่างแนวคิดการค้นหาความบรรสานอย่างง่าย แสดงดังรูปที่ 6 ตัวอย่างในการปรับปรุงและการหาค่าเหมาะที่สุด นักดนตรี (แซคโซโฟน, ดับเบิ้ลเบส และ กีต้าร์) ได้มีการบรรเลงเสียงของแต่ละเครื่องดนตรีออกมา (แซคโซโฟน = (Do, Re, Mi), ดับเบิ้ลเบส = (Mi, Fa, Sol), และกีต้าร์ = (Sol, La, Si) ซึ่งเมื่อทำการเลือกเสียงออกมา โดย แซคโซโฟน เลือกเสียง Do ดับเบิ้ลเบส เลือกเสียง Mi และกีต้าร์ เลือกเสียง Sol จะได้เสียงใหม่ออกมา คือ (Do,Mi,Sol) ถ้าเสียงที่ได้ใหม่นั้นมีคุณภาพที่ดี จะเก็บไว้ในความทรงจำของนักดนตรี เช่นเดียวกันกับ ฟังก์ชั่นทางคณิตศาสตร์ กำหนดช่วงของค่า $x_1 = \{1,2,3\}$ $x_2 = \{3,4,5\}$ และ $x_3 = \{5,6,7\}$ โดยทำการเลือกค่าออกมา $x_1 = 1$ $x_2 = 3$ และ $x_3 = 5$ จะได้เวกเตอร์คำตอบใหม่ออกมา คือ (1, 3, 5) ถ้าค่าที่ได้เป็นค่าเหมาะที่สุดในการค้นหานั้น จะถูกเก็บไว้ในหน่วยความจำ ทั้งนี้รายละเอียดขั้นตอนวิธีการ (Algorithm) จะอธิบายอีกครั้งในหัวข้อ 2.2.4 ต่อไป [25]

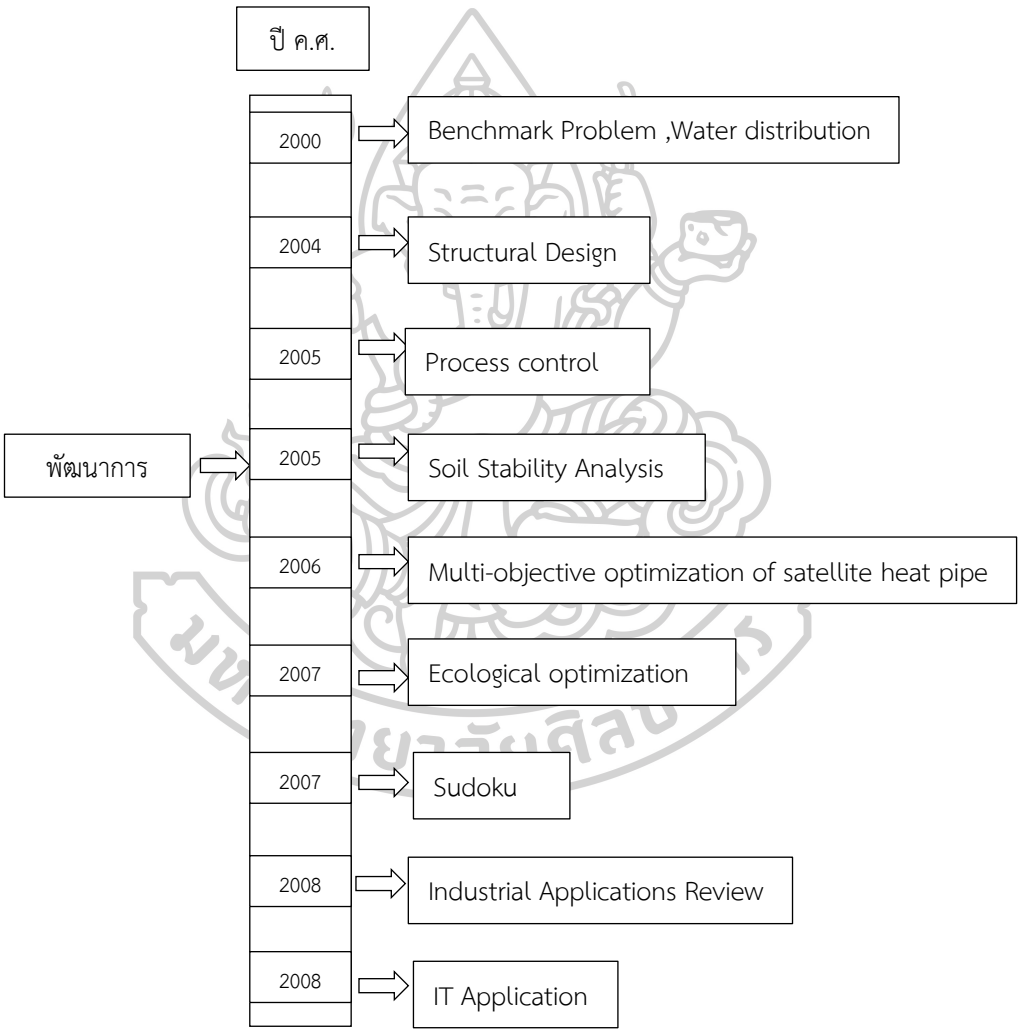


รูปที่ 6 ตัวอย่างในการปรับปรุงและการหาค่าเหมาะที่สุด [25]

2.4.2 การพัฒนาการค้นหาความบรรสาน

หลังจากที่ Zong Woo Geem ได้ทำการคิดค้นการค้นหาความบรรสานขึ้นมา วิธีการดังกล่าวได้ถูกนำไปใช้งานมากขึ้น ดังแสดงในรูปที่ 7 โดยในปี ค.ศ. 2000 ซึ่งถือได้ว่าเป็นปีแรกเริ่มในการค้นคว้าวิจัยการค้นหาความบรรสาน ได้มีการนำไปทำการทดสอบกับชุดปัญหามาตรฐาน และการขนถ่ายน้ำ ต่อมาในปี ค.ศ. 2004 Zong Woo Geem ได้ทำการค้นคว้าร่วมมือกับ Kang Seok Lee ในการนำไปใช้งานทางด้านการออกแบบโครงสร้าง ซึ่งเป็นการประยุกต์ใช้งานแรกที่มีการนำไปใช้งาน หลังจากนั้นการค้นหาความบรรสานได้เริ่มมีการพัฒนาไปใช้ในงานด้านอื่น อาทิเช่น ในปี ค.ศ. 2006 ได้นำไปใช้ในงานออกแบบท่อที่ติดต้นในดาวเทียม [26] ซึ่งเป็นรูปแบบปัญหาหลายวัตถุประสงค์ หรือแม้แต่นำไปทดสอบในการแก้ไขเกมปัญหาซูโดะคุ [27] ในปี ค.ศ. 2007 เป็นต้น

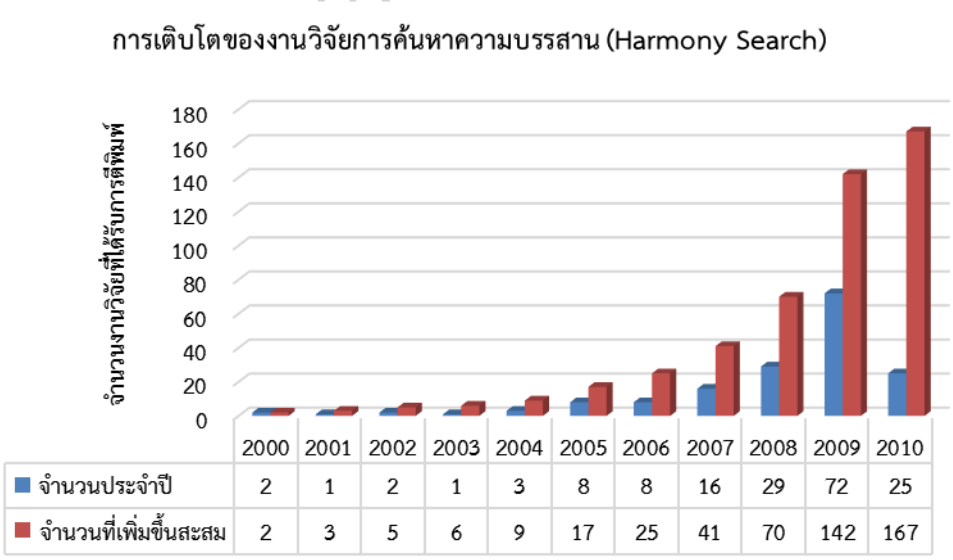
จากข้อความดังกล่าวข้างต้นจะเห็นได้ว่าเมื่อเวลาผ่านไป การนำการค้นหาความบรรสารไปใช้งานนั้นมีความหลากหลายมากขึ้น ไม่เพียงแต่งานทางด้านวิศวกรรมเท่านั้น ยังรวมไปถึงงานทางด้านสาธารณะสุข ทางด้านเทคโนโลยีสารสนเทศ และอีกมากมาย โดยในที่นี้ทางผู้วิจัยจะขอกล่าวเพิ่มเติมในหัวข้อ 2.2.4



รูปที่ 7 การเติบโตของการค้นหาความบรรสาน (ปรับปรุงจาก Ingram and Zhang [28])

2.4.3 การเพิ่มขึ้นของความสนใจการใช้งานของการค้นหาความบรรสาน

การตีพิมพ์งานวิจัยในเรื่อง การค้นหาความบรรสาน ในระดับนานาชาตินั้นเริ่มขึ้นในปี ค.ศ. 2001 จากการค้นหาข้อมูลในฐานข้อมูลทางวิทยาศาสตร์ (Google Scholar, Heal Link, Microsoft Academic Search) เมื่อนำข้อมูลมาเปรียบเทียบกันระหว่างปี ค.ศ. 2005 กับปี ค.ศ. 2008 พบว่ามีปริมาณการตีพิมพ์งานวิจัยเพิ่มขึ้น 4 เท่าและเมื่อนำมาเปรียบเทียบกับปี ค.ศ. 2009 มีการเพิ่มขึ้นจากเดิมสูงถึง 9 เท่า [29] ดังนั้นจึงเห็นได้ว่าปริมาณงานวิจัยที่เกี่ยวข้องกับ การค้นหาความบรรสานนั้นมีปริมาณเพิ่มขึ้นอย่างเห็นได้ชัด ดังแสดงในรูปที่ 8



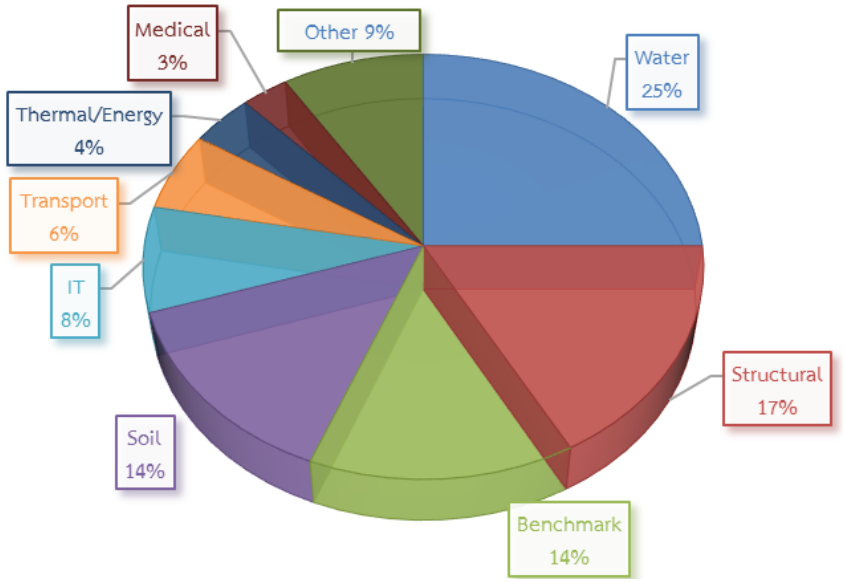
รูปที่ 8 การเพิ่มขึ้นเพิ่มขึ้นของความสนใจการใช้งานของการค้นหาความบรรสาน

จากหัวข้อที่ 2.4.2 และ 2.4.3 จะเห็นได้ว่าความหลากหลายที่เพิ่มมากขึ้นของการค้นคว้าวิจัยของการค้นหาความบรรสาน จากเดิมในปี ค.ศ. 2000 ถึง ค.ศ. 2004 มีการนำการค้นหาความบรรสานไปใช้กับการทดสอบกับฟังก์ชั่นมาตรฐาน และการออกแบบการขนส่งน้ำเท่านั้น จึงทำให้วิจัยที่ตีพิมพ์ออกมาไม่มากนัก แต่ในปี ค.ศ. 2008 เป็นต้นไป จะเห็นได้ว่างานวิจัยที่ได้รับการตีพิมพ์นั้นมากขึ้นอย่างต่อเนื่อง ซึ่งเป็นผลมาจาก การนำขั้นตอนวิธีการค้นหาความบรรสานไปใช้ในด้านอุตสาหกรรม ด้วยเหตุนี้เองขั้นตอนวิธีการค้นหาความบรรสานจึงมีความน่าสนใจในการศึกษาและพัฒนาต่อไป

2.4.4 การประยุกต์ใช้ที่มีการนำการค้นหาความบรรสานไปใช้งาน

จากความสัมพันธ์ระหว่างพัฒนาการของการค้นหาความบรรสาร กับปริมาณงานวิจัยของการค้นหาความบรรสานที่ในช่วง 10 ปี ที่ผ่านมา พบว่าการประยุกต์ใช้งานที่มีการนำการค้นหาความบรรสานไปประยุกต์ใช้งานนั้นมีความหลากหลายเป็นอย่างมาก ไม่ว่าจะเป็นในงานทางด้านออกแบบโครงสร้าง การจัดการข้อมูล และในอุตสาหกรรมต่าง ๆ ได้มีการนำการค้นหาความบรรสานไปใช้งาน

ดังนั้นเพื่อให้ง่ายต่อการศึกษา Ingram และ Zhang [28] ได้ทำการแบ่งประเภทการประยุกต์ใช้งานออกเป็น 9 ประเภท ดังแสดงในรูปที่ 9 ได้แก่ ปัญหาด้านการจัดการน้ำ (Water-Related Applications) การออกแบบโครงสร้าง (Structural Design) การแก้ไขชุดปัญหามาตรฐาน (Benchmark Optimization) การวิเคราะห์เสถียรภาพของพื้นดิน (Soil Stability Analysis) เทคโนโลยีสารสนเทศ (Information Technology Applications) ปัญหาด้านการขนส่ง (Transport-Related Problems) ด้านพลังงาน (Thermal and Energy Application) ทางด้านสาธารณะสุข (Medical Studies) และ อื่นๆ (Other Applications)



รูปที่ 9 การประยุกต์ใช้งานการค้นหาความบรรสาน [28]

โดยทั่วไปสิ่งที่สนใจในด้านการจัดการทรัพยากรน้ำ คือการบริหารจัดการน้ำที่มีอยู่ให้เพียงพอต่อความต่อต้องการ ในการลงทุนติดตั้งระบบการจ่ายน้ำจะทำการติดตั้งอย่างไรที่จะไม่ทำให้การใช้น้ำมีปัญหา ซึ่งในการค้นหาความบรรสานได้มีการศึกษาด้านการจัดการน้ำ โดยมีสัดส่วนงานวิจัยถึงร้อยละ 25 ซึ่งเป็นสัดส่วนที่มากที่สุดเป็นลำดับแรก จากงานวิจัยทั้งหมดที่นำวิธีการค้นหาความบรรสานไปใช้งาน

Geem [30] ได้ศึกษาการออกแบบเครือข่ายการจ่ายน้ำของเทศบาล ซึ่งพบว่าขนาดของท่อ (Pipe diameter) นั้นมีทั้งขนาดที่กว้างเกินไป และแคบเกินไป การให้ความดันแต่ละท่อจ่าย (Supply node) ไปยัง ท่อรับ (Demand node) นั้นไม่เหมาะสม ดังนั้นในการศึกษาจึงทำการพิจารณาจากค่าเหมาะที่สุดของขนาดท่อ การกำหนดความดันในแต่ละท่อจ่าย และโครงข่ายในการวางท่อจ่าย โดยพิจารณาทั้งระดับความสูงและระยะทางระหว่างจุด ซึ่งวิธีการค้นหาความบรรสาน จะไปทำการเลือกขนาดของท่อในแต่ละส่วนเพื่อให้มีค่าใช้จ่ายน้อยที่สุด และเพียงพอต่อความต้องการของลูกค้าในแต่ละจุดสำหรับการใช้แรงดันในการจ่ายน้ำที่ต่ำที่สุด

ลำดับต่อมาโดยทั่วไปด้านการออกแบบโครงสร้างส่วนใหญ่มีความต้องการ ค่าใช้จ่ายในการปลูกสร้างน้อยที่สุด น้ำหนักโครงสร้างน้อยที่สุด และโครงสร้างมีความแข็งแรงมากที่สุด ดังนั้นเพื่อให้ตอบสนองต่อความต้องการข้างต้น จึงมีความจำเป็นที่ต้องกำหนดพื้นที่หน้าตัดที่เหมาะที่สุด จุดบนคานรับน้ำหนัก รูปแบบการวางโครงสร้าง และความยาวของคานรับน้ำหนัก แต่อย่างไรก็ตามการออกแบบให้ตรงตามความต้องการนั้นเป็นไปได้ยาก จึงได้มีการนำการค้นหาความบรรสานมาช่วยแก้ไขปัญหา ซึ่งได้มีการศึกษาทั้งปัญหาตัวแปรแบบต่อเนื่อง (Continuous variable problem) และปัญหาตัวแปรไม่ต่อเนื่อง (Discrete variable problem)

Lee และ Geem [31] ได้นำเสนอการหาขนาดโครงถัก (truss) แบ่งการทดสอบเป็น 3 แบบ 1.เงื่อนไขการรับแรงเงื่อนไขเดียว ทำการทดสอบกับโครงถัก 2 มิติ (planar truss) และโครงสร้างถักแบบโดม (Dome space truss) ที่มีแท่งเหล็ก (bar) 10 18 และ 120 แท่งตามลำดับ 2. เงื่อนไขการรับแรงสองเงื่อนไข ทำการทดสอบกับโครงสร้างถักแบบ (Space truss) ที่มีแท่งเหล็ก 25 และ 72 แท่งตามลำดับ 3. เงื่อนไขการรับแรงสามเงื่อนไข ทำการทดสอบกับโครงสร้างถักแบบ และโครงถัก 2 มิติ ที่มีแท่งเหล็ก 22 และ 200 แท่งตามลำดับ ซึ่งพบว่าในการใช้การค้นหาความบรรสานนั้นจะมีค่าที่เหมาะสมกว่าวิธีทางคณิตศาสตร์แบบดั้งเดิม (Conventional Mathematical) และเมื่อเปรียบเทียบกับขั้นตอนวิธีเชิงพันธุกรรม (GA) พบว่าประสิทธิภาพการลู่เข้า (Convergence capability) ของการค้นหาความบรรสานนั้นมีประสิทธิภาพที่ดีกว่า

การทดสอบขั้นตอนวิธีที่คิดค้นขึ้นมาใหม่ หรือขั้นตอนวิธีที่มีการปรับปรุงพารามิเตอร์กับชุดปัญหา (Benchmark Problems) ถือได้ว่าเป็นการทดสอบความสามารถในการค้นหาคำตอบของขั้นตอนวิธี เนื่องจากชุดปัญหาที่นำมาทดสอบจะมีหลากหลายลักษณะ ไม่ว่าจะเป็น ปัญหาการหาค่าเหมาะที่สุดแบบต่อเนื่อง (Continuous Optimization Problems) ปัญหาการหาค่าเหมาะที่สุดแบบไม่ต่อเนื่อง (Discreat Optimization Problems) อีกทั้งยังมีรูปแบบที่หลากหลาย อาทิเช่น รูปแบบ

ไม่มีข้อจำกัด (Unconstrain) รูปแบบมีข้อจำกัด (Constrains) ที่ฟังก์ชั่นมีหลายจุดเฉพาะที่ (local) ทั้งนี้ในการนำการค้นหาความบรรสานมาทำการทดสอบแก้ไขชุดปัญหานั้น ซึ่งการค้นหาความบรรสานที่นำมาทดสอบประกอบไปด้วยวิธีการค้นหาความบรรสานแบบดั้งเดิม และวิธีการค้นหาความบรรสานที่มีการปรับปรุงพารามิเตอร์

Lee และ Geem [32] นำเสนอประสิทธิภาพการค้นหาความบรรสาน โดยทำการทดสอบชุดปัญหามาตรฐานทั้งหมด 12 ฟังก์ชั่นซึ่งประกอบไปด้วยรูปแบบไม่มีข้อจำกัดจำนวน 6 ฟังก์ชั่น และรูปแบบมีข้อจำกัดจำนวน 6 ฟังก์ชั่น จากการทดสอบพบว่าการค้นหาความบรรสานเป็นขั้นตอนวิธีการที่สามารถหาคำตอบแบบวงกว้าง (Global search algorithm) นอกจากนี้ผลของการทดสอบในบางฟังก์ชั่นนั้นให้ค่าที่ดีกว่าขั้นตอนวิธีการแบบเดิม ดังนั้นปัญหาที่ซับซ้อนในงานวิศวกรรมสามารถนำขั้นตอนวิธีการการค้นหาความบรรสานมาแก้ไขได้

การก่อสร้างสิ่งปลูกสร้างโครงสร้างของพื้นดินถือเป็นส่วนสำคัญ ถ้าโครงสร้างของพื้นดินมีลักษณะพื้นผิวที่ลาดเอียง จะทำให้เกิดความไม่เสถียรและการเคลื่อนตัวของพื้นดิน ดังนั้นจึงจำเป็นต้องมีการวิเคราะห์เสถียรภาพของพื้นดิน โดยมีวัตถุประสงค์เพื่อที่จะทำนายตำแหน่งที่จะเกิดการเคลื่อนตัว และประเมินถึงความปลอดภัยให้แก่โรงงาน ซึ่งพิจารณาจากอัตราส่วนของความสามารถของมวลดินในการรับแรงเฉือน (Inherent shear strength) ต่อแรงเฉือน (Shear Stress) โดยทั่วไปพื้นผิวของดินในแต่ละพื้นผิวจะให้ค่าการเคลื่อนตัวของพื้นผิวที่แตกต่างกัน จากสาเหตุดังกล่าวข้างต้นจึงมีการนำวิธีการค้นหาความบรรสานไปใช้งาน

Liang Li และคณะ [33] ได้นำเสนอการหาค่าความปลอดภัยของพื้นดินที่ลาดเอียง โดยใช้การค้นหาผสมระหว่าง การหาค่าเหมาะสมที่สุดแบบฝูงอนุภาค กับ การค้นหาความบรรสาน แต่อย่างไรก็ตามเมื่อทดสอบแล้วพบว่าในการใช้งานจริงยังคงหาคำตอบได้ดีกว่าจึงมีการพัฒนาการค้นหาความบรรสานต่อไป

จากความก้าวหน้าด้านเทคโนโลยีในปัจจุบัน ทำให้รูปแบบของข้อมูลอยู่ในรูปแบบออนไลน์มากขึ้น ดังนั้นการจัดการข้อมูล การแบ่งประเภทข้อมูล และกำหนดกลุ่มของข้อมูลเข้ามามีบทบาทสำคัญ จึงได้มีการนำเทคนิคการค้นหาความบรรสานมาประยุกต์ใช้งาน สำหรับงานด้านเทคโนโลยีสารสนเทศ

Forsati และคณะ [34] ได้แสดงการจัดกลุ่มข้อมูลเอกสารออนไลน์ ซึ่งมีนำเอา การค้นหาความบรรสาน และการค้นหาความบรรสานที่ผสมผสาน กับ K-mean (K-means Clustering Algorithm) มาทดสอบ พบว่าขั้นตอนวิธีการแบบผสมผสานนั้นสามารถหาคำตอบได้ดีกว่าเมื่อเปรียบเทียบกับวิธีการอื่น

การขนส่งหรือโลจิสติกส์ ถือว่ามีความสำคัญอย่างมาก ไม่ว่าจะเป็นอุตสาหกรรม หรือการบริการ ดังนั้นการนำวิธีการค้นหาความบรรสานไปใช้งาน ด้านของปัญหาการขนส่ง พบว่ามีสัดส่วนร้อยละ 6

Geem, Kim และ Loganathan [23] ได้นำเสนอการใช้การค้นหาความบรรสานแก้ไขปัญหาการเดินทางของพนักงานขาย (Travelling Sell-man Problem, TPS) สำหรับจำนวนเมือง 20 เมือง โดยมีวัตถุประสงค์การหาเส้นทางที่สั้นที่สุดสำหรับพนักงานขาย สามารถพบค่าเหมาะที่สุดแบบวงกว้าง (Global optimum) หลังจากการทำซ้ำจำนวน 5,000 ครั้ง

แต่อย่างไรก็ตามเมื่อพิจารณาการประยุกต์ใช้ทางด้านพลังงาน และทางด้านสาธารณะสุข พบว่าการนำค้นหาความบรรสานไปใช้งานนั้นยังไม่เป็นแพร่หลายนัก โดยมีสัดส่วนเพียงร้อยละ 4 และ ร้อยละ 3 ตามลำดับ

Geem และ Hwangbo [35] ได้ทำการออกแบบท่อส่งความร้อนในดาวเทียม ปัญหาจะประกอบด้วย การหาขนาดท่อ การทำงานในการส่งผ่านความร้อนโดยต้องการน้ำหนักของท่อที่น้อยที่สุด ในขณะที่ค่าการต้านทานความร้อนจะมีค่ามากที่สุด

Aditya และ Panchal [36] ได้แสดงการนำการค้นหาความบรรสานไปใช้ในการรักษามะเร็งด้วยรังสีเป็นการยับยั้งบริเวณเนื้อร้ายที่เกิดมะเร็ง โดยจุดประสงค์ของการวิจัยนี้เพื่อทำการศึกษาการหาค่าเหมาะที่สุดของเวลาระหว่างในการฉายรังสีที่มีผลกระทบต่อเนื้อเยื่อข้างเคียง เพื่อลดผลกระทบของการฉายรังสีที่จะเกิดขึ้น

นอกเหนือจากการประยุกต์ใช้งานดังกล่าวข้างต้น ซึ่งพบว่ามีสัดส่วนร้อยละ 9 ที่การค้นหาความบรรสานยังสามารถนำไปใช้งานในด้านอื่น

Geem [27] ได้นำการค้นหาความบรรสานมาใช้ในวิธีการแก้ปัญหาซูโดะคุ (Sudoku Puzzle) โดยมีหลักการในการเล่นเกมคือ ทุกแถวในแนวนอน ต้องมีตัวเลข 1 ถึง 9 ไม่ซ้ำกัน ทุกแถวในแนวตั้ง ต้องมีตัวเลข 1 ถึง 9 ไม่ซ้ำกัน และทุกตารางย่อย 3 x 3 ต้องมีตัวเลข 1 ถึง 9 ไม่ซ้ำกัน จาก

ทำการทดสอบพบว่าสามารถหาคำตอบเหมาะที่สุดวงกว้าง (Global Optimum) ได้ถึง 33 ครั้งจาก 36 มีช่วงเวลาในการค้นหาคำตอบเท่ากับ 3 ถึง 38 วินาที หรือโดยเฉลี่ยประมาณ 8 วินาที

แต่อย่างไรก็ตามการใช้การค้นหาความบรรสานมาแก้ไขปัญหานั้นยังไม่มีประสิทธิภาพที่ดี เนื่องจากพบว่าในบางกรณีคำตอบที่พบจะเป็นแบบวงแคบ (Local Optimum) ซึ่งเกิดจากการค้นหาเฉพาะที่ (Local Search)

จากข้างต้นจะเห็นได้ว่าขั้นตอนวิธีการค้นหาความบรรสาน (Harmony Search: HS) นั้นสามารถที่จะนำไปประยุกต์ใช้กับงานได้หลากหลายประเภท ถึงแม้ว่างานดังกล่าวนั้นจะไม่เกี่ยวข้องกันเลยก็ตาม ดังนั้นการพัฒนาและปรับปรุงให้ขั้นตอนวิธีการดังกล่าว ให้มีประสิทธิภาพการค้นหาคำตอบได้ดีมากยิ่งขึ้นนั้นเป็นประโยชน์อย่างมาก โดยรายละเอียดในการพัฒนาและปรับปรุงขั้นตอนวิธีการการค้นหาความบรรสานจะขอกล่าวในหัวข้อที่ 2.2.5 ต่อไป

2.2.5 หลักการของการค้นหาความบรรสานแบบดั้งเดิม

จากการที่กำหนดปัญหาที่มาทำการหาค่าเหมาะที่สุด และค่าตัวแปร (Parameter) แล้วจะมาทำการหาคำตอบเหมาะที่สุดด้วยวิธีค้นหาความบรรสาน ซึ่งแสดงขั้นตอนดั้งนี้

ขั้นตอนที่ 1 เริ่มต้นสร้างหน่วยความจำความบรรสาน (Harmony memory: HM)

การเริ่มต้นของหน่วยความจำความบรรสานประกอบด้วย ตัวเลขจำนวนหนึ่งของวิธีแก้ปัญหาที่ทำการสุ่มขึ้นมา สำหรับปัญหาค่าเหมาะที่สุดภายใต้การพิจารณา สำหรับปัญหา n-dimension โดยหน่วยความจำบรรสาน กับ ขนาดหน่วยความจำบรรสาน (Hamony memory size: HMS) สามารถแสดงได้ดังต่อไปนี้

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_n^{HMS} \end{bmatrix} = \begin{bmatrix} x^1 \\ \vdots \\ x^{HMS} \end{bmatrix} \in \Re^{HMS \times n}$$

โดยที่ $(x_1^i, x_2^i, \ldots, x_n^i)$ และ $(i = 1,2, \ldots, HMS)$ คือ การแก้ไขคำตอบแบบใหม่ (Candidate solution) ซึ่งขนาดหน่วยความจำบรรสาน จะมีค่าอยู่ระหว่าง 10 ถึง 100

ขั้นตอนที่ 2 การปรับปรุงวิธีแก้ปัญหาใหม่

การปรับปรุง $(x_1', x_2', \dots, x_n')$ จาก หน่วยความจำความบรรสาน (Harmony memory: HM) นั้นคือการสร้างเวกเตอร์ในการแก้คำตอบใหม่ขึ้นมา โดยการปรับปรุงค่าความบรรสาน (Harmony) ใหม่ขึ้นมานั้นจะขึ้นอยู่กับ 3 การดำเนินการ

1. การใช้งานหน่วยความจำความบรรสาน (Harmony memory: HM)

การใช้หน่วยความจำ ค่าตัวแปรแรก x_1' สำหรับเวกเตอร์ใหม่สามารถเลือกได้จากค่าใดๆ ในช่วงหน่วยความจำบรรสานที่ระบุ $(x_1' - x_1^{HMS})$ ค่าของตัวแปรอื่น (x_n') สามารถเลือกได้โดยวิธีเดียวกัน โดยในการเลือกเป็นไปได้ที่จะเลือกค่าใหม่โดยใช้ พารามิเตอร์ความน่าจะเป็นในการใช้หน่วยความจำ (Harmony Memory Considering Rate, HMCR) ที่แตกต่างกันช่วงระหว่าง 0 ถึง 1

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} & with\ probability\ HMCR \\ x_i' \in X_i & with\ probability\ (1 - HMCR) \end{cases}$$

โดยที่ HMCR คือความน่าจะเป็นของการเลือกตัวแปรค่าหนึ่งจากค่าที่เก็บไว้ใน HM และ (1-HMCR) คือความน่าจะเป็นของการสุ่มเลือกตัวแปรค่าหนึ่งที่ไม่ได้เก็บไว้ใน HM เช่น ค่า *HMCR* มีค่าเท่ากับ 0.95 หมายความว่ามีโอกาส 95 เปอร์เซ็น ที่จะเลือกตัวแปรจากค่าที่เก็บไว้ และมีโอกาส 5 เปอร์เซ็นที่จะไปเลือกตัวแปรจากค่าที่ไม่ได้เก็บไว้ *(*เลือกจากช่วงที่เป็นไปได้ทั้งหมด*)* ดังนั้นการเลือกใช้ค่า HMCR นั้นไม่ควรมีค่าเท่ากับ 1 เพราะจะไม่สามารถเลือกค่าตัวแปรจากที่ไม่ได้เก็บไว้ใน HM ได้เลย ซึ่งค่าตัวแปรที่ไม่ได้เก็บไว้ใน HM อาจจะมีค่าที่ให้คำตอบที่เหมาะสมกว่า

2. การปรับระดับเสียง (Pitch adjustment)

ทุกองค์ประกอบของเวกเตอร์ใหม่ $x' = (x_1', x_2', \dots, x_N')$ จะถูกตรวจสอบ เพื่อพิจารณาว่าควรจะปรับระดับเสียง โดยในขั้นตอนการปรับระดับเสียงจะใช้ พารามิเตอร์ความน่าจะเป็นในการปรับระดับเสียง (Pitch Adjustment Rate, PAR) ในการกำหนดอัตราการเปลี่ยนแปลงของการปรับสำหรับเสียงที่ถูกเลือกมาจาก HM

$$Pitch\ adjusting\ decision\ for\ x_i' \leftarrow \begin{cases} Yes & with\ probability\ PAR \\ No\ with\ probability\ (1 - PAR) \end{cases}$$

กระบวนการปรับเสียงจะเริ่มดำเนินการก็ต่อเมื่อมีค่าที่ถูกเลือกมาจากหน่วยความจำบรรสาน (HM) โดยค่า PAR คือ โอกาสในการปรับค่าที่ได้จาก HM และ (1-PAR) คือ โอกาสที่ค่าจะไม่มีการปรับค่าใดๆทั้งสิ้น เช่น ค่า PAR เท่ากับ 0.3 หมายความว่าจะมีโอกาสที่ทำการเลือกค่าใกล้เคียงกับ

$30\% \times HMCR$ ที่จะทำการปรับค่า ถ้าตัดสินใจที่จะทำการปรับระดับเสียง $x_i' = Yes$ และค่าของ x_i' ถูกกำหนดให้เป็น $x_i(k)$ โดย k เป็นองค์ประกอบใน X_i ดังนั้น ค่าที่จะทำการปรับระดับเสียงของ $x_i(k)$ คือ $x_i' \leftarrow x_i' + \alpha$ โดย α คือ ค่าของ $bw \times u(-1,1)$ bw คือ arbitrary distance bandwidth สำหรับตัวแปรต่อเนื่อง และ $u(-1,1)$ คือ จำนวนที่สุ่มจากรูปแบบการกระจายตัวที่อยู่ในช่วง -1 ถึง 1

3. การสุ่มค่า (Randomization)

การสุ่มค่าเป็นการเพิ่มความหลากหลายของวิธีแก้ไขปัญหา โดยค่าความน่าจะเป็นของการสุ่ม (Probability of randomization) มีค่า $P_{random} = 1 - HMCR$ และค่าความน่าจะเป็นของการปรับเสียง (Probability of the pitch adjustment) มีค่า $P_{pitch} = HMCR \times PAR$

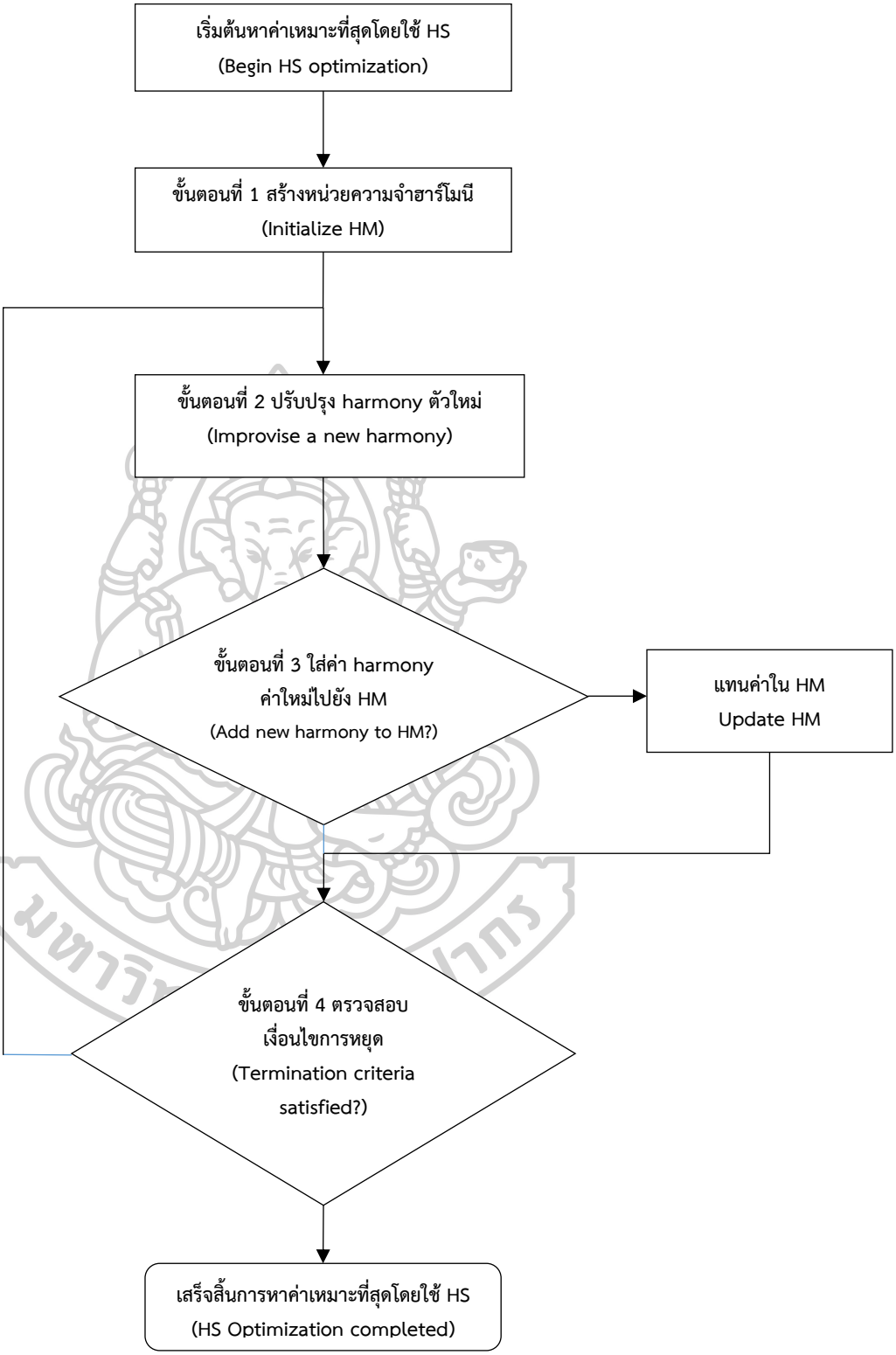
ขั้นตอนที่ 3 ใส่ค่า Harmony ค่าใหม่ใน หน่วยความจำความบรรสาน

เริ่มแรก วิธีแก้ปัญหาใหม่ จาก ขั้นตอนที่ 2 ที่ถูกคำนวณแล้ว ถ้าผลลัพธ์ค่า fitness ดีกว่าค่าที่ถูกเก็บใน หน่วยความจำความบรรสาน จะถูกแทนที่เข้าไปแทนค่าในหน่วยความจำความบรรสาน และค่าที่แย่กว่าจะถูกกำจัดออก

ขั้นตอนที่ 4 ตรวจสอบเงื่อนไขการหยุด

กลับไปสู่ ขั้นตอนที่ 2 จนกระทั่งพบเจอเงื่อนไขที่เป็นไปได้ (เช่น ค่ามากที่สุดของการทำซ้ำ)

มหาวิทยาลัยศิลปากร



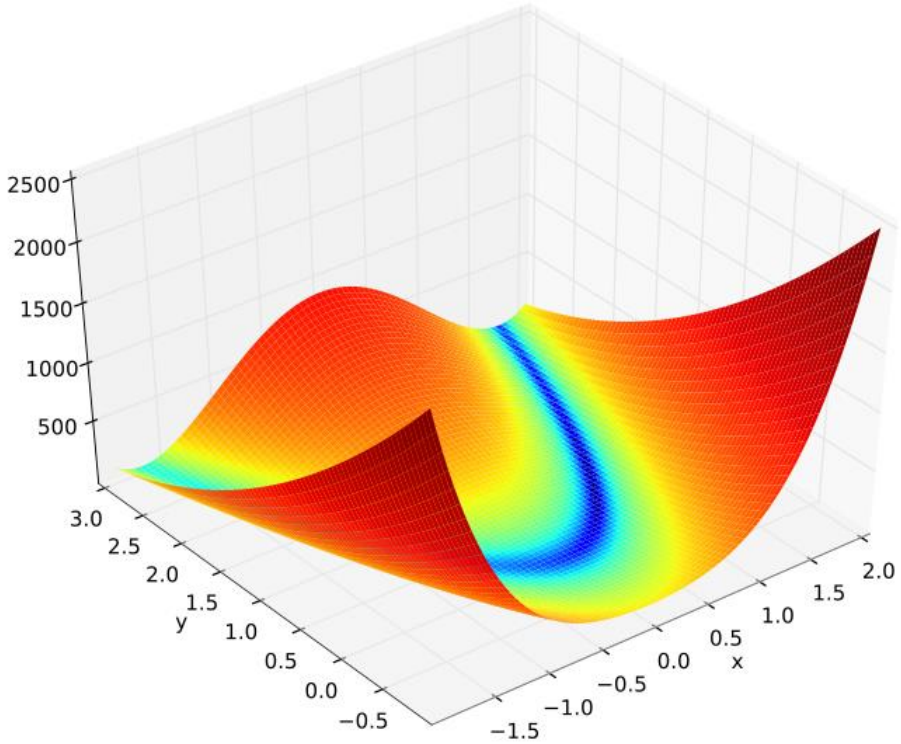
รูปที่ 10 ขั้นตอนการค้นหาความบรรสาน [28]

2.2.6 การใช้งานวิธีค้นหาความบรรสานแบบดั้งเดิมสำหรับการทดสอบฟังก์ชั่น

การทดสอบปัญหามาตรฐาน (Benchmark function) นั้นสามารถที่จะแสดงถึงความสามารถในการหาคำตอบของขั้นตอนวิธีการแต่ละชนิดได้ มีการนำวิธีค้นหาความบรรสานมาทำการทดสอบกับปัญหามาตรฐาน โดยประกอบไปด้วยฟังก์ชั่นน้อยที่สุดชนิดไม่มีข้อจำกัด (Unconstrained function minimization) ฟังก์ชั่นน้อยที่สุดชนิดมีข้อจำกัด (Constrained function minimization) โดยในที่นี้ทางผู้วิจัยจะทำการยกตัวอย่างการใช้การค้นหาความบรรสานทดสอบ Rosenbrock function ซึ่งเป็นฟังก์ชั่นน้อยที่สุดชนิดไม่มีข้อจำกัด (Unconstrained function minimization) ที่มีลักษณะ long narrow และ curved valley โดยแสดงดังนี้

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

ฟังก์ชั่น Rosenbrock มีค่าต่ำสุด เท่ากับ $x^* = (1.0,1.0)$ และให้ค่าฟังก์ชั่นวัตถุประสงค์ $f^*(x)$เท่ากับ 0.00



รูปที่ 11 ฟังก์ชั่น Rosenbrock (Rosenbrock function) [37]

จากการใช้ การค้นหาความบรรสาน ทดสอบฟังก์ชั่น Rosenbrock โดยกำหนด x_1และx_2 อยู่ในช่วงระหว่าง -0.5 ถึง 3.0 และค่าพารามิเตอร์ (Harmony memory size: HMS) = 20 (Harmony memory consideration rate: HMCR) = 0.90 (Pitch adjusting rate: PAR) = 0.35

หลังจากการค้นหา 50,000 ครั้ง ได้ชุดคำตอบที่ดีที่สุด คือ $x = (1.000000000, 1.000002384)$ ให้ค่าฟังก์ชั่นวัตถุประสงค์ $f(x)$ เท่ากับ $5.68434188 \times 10^{-10}$ ซึ่งมีค่าใกล้เคียงกับค่าเหมาะที่สุดเป็นอย่างมาก [38]

แต่อย่างไรก็ตามหลังจากการทดสอบค้นหาความบรรสานกับปัญหามาตรฐาน พบว่าบางปัญหานั้นยังไม่ได้ค่าที่เหมาะที่สุด แสดงว่าขั้นตอนในการวนหาค่าเหมาะที่สุดนั้นไม่มีประสิทธิภาพเพียงพอที่จะแก้ไขได้ จากจุดด้อยของการค้นหาความบรรสานดังกล่าว นักวิจัยจึงพยายามที่จะการพัฒนาปรับปรุงการค้นหาความบรรสานให้ดียิ่งขึ้น

2.2.7 การปรับปรุงการค้นหาความบรรสาน

จากข้อความดังกล่าวข้างต้น เห็นได้ว่าการค้นหาความบรรสานแบบดั้งเดิม นั้นยังไม่ดีเพียงพอ ดังนั้นจึงมีนักวิจัยหลายท่านได้ทำการศึกษาและพัฒนาต่อยอดจากวิธีค้นหาความบรรสานแบบดั้งเดิม ในที่นี้ทางผู้วิจัยได้ทำการแบ่งประเภทของการพัฒนาและการปรับปรุงออกเป็น 2 ประเภท ได้แก่ ขั้นตอนวิธีการที่ไม่ผสมผสาน (Non-Hybrid Algorithm) และขั้นตอนวิธีการแบบผสมผสาน (Hybrid Algorithm)

ขั้นตอนวิธีการที่ไม่ผสมผสาน (Non-Hybrid Algorithm) คือ ขั้นตอนวิธีการที่เปลี่ยนแปลงการค้นหาแต่ไม่ได้เปลี่ยนแปลงรูปแบบ หรือเป็นเพียงการปรับพารามิเตอร์จากรูปแบบดั้งเดิม

Mahdavi, Fesanghary, และ Damangir [39] ได้พัฒนาการค้นหาความบรรสานแบบปรับปรุง (Improved Harmony Search: IHS) ขึ้นโดยพบว่าการค้นหาความบรรสานแบบดั้งเดิม ค่าของ PAR และ BW มีความสำคัญอย่างมากในการค้นหาเวกเตอร์ของคำตอบเหมาะที่สุด อีกทั้งค่าดังกล่าวจึงมีประสิทธิภาพที่จะช่วยในการกำหนดอัตราการลู่เข้าสู่คำตอบ (Convergence rate) แต่พบว่าการค้นหาความบรรสานแบบดั้งเดิม มีการกำหนดค่า PAR และค่า BW เป็นค่าคงที่ไม่สามารถปรับเปลี่ยนได้

จากเหตุนี้เองจึงได้มีการคิดค้นการปรับค่าของความน่าจะเป็นในการปรับระดับเสียง (PAR) โดยแต่เดิมค่า PAR จะเป็นค่าคงที่ไม่เปลี่ยนแปลงตลอดการทดสอบ แต่ในที่นี้ค่าดังกล่าวจะถูกปรับค่าแบบไดนามิก อีกทั้งค่า BW จะถูกปรับให้มีการเพิ่มขึ้นอย่างต่อเนื่องแบบเอ็กโปเนนเชียล แสดงดังสมการต่อไปนี้

$$PAR(gn) = PAR_{min} + \left(\frac{PAR_{max} - PAR_{min}}{NI}\right) \times gn \text{ และ } BW(gn) = BW_{max}\exp(c.gn)$$

โดยพบว่า ในช่วงแรกของการทำซ้ำ ค่า PAR มีค่าต่ำ ค่า BW มีค่าที่สูง จึงทำให้ diversification ที่ดี และเมื่อทำการค้นหาคำตอบไประยะเวลาหนึ่ง ค่า PAR จะมีค่าสูงขึ้น ค่า BW จะมีค่าต่ำลง โดยสามารถให้ intensification ที่ดี จากนั้นทำการทดสอบด้วยฟังก์ชั่นข้อจำกัดของการออกแบบความดันท่อ เมื่อนำมาเปรียบเทียบกับการค้นหาความบรรสานแบบดั้งเดิม ขั้นตอนวิธีการเชิงพันธุกรรม และวิธีบรานซ์แอนด์บาวนด์ พบว่าวิธีการที่ได้พัฒนาขึ้นนั้นสามารถหาค่าได้ดีที่สุดเมื่อเทียบกับวิธีการข้างต้น

Omran และ Mahdavi [40] ได้พัฒนาการค้นหาความบรรสานแบบโกลบอลเบส (Global-best Harmony Search: GHS) ขึ้นมาโดยได้แนวคิดมาจากการหาค่าเหมาะที่สุดแบบอนุภาคกลุ่ม โดยอนุภาคกลุ่มจะบินผ่านพื้นที่ค้นหา โดยแต่ละอนุภาคจะแสดงคำตอบของผลเฉลย ซึ่งตำแหน่งของคำตอบที่ดีที่สุดในเซตการค้นหาคำตอบนั้นจะดึงให้ตำแหน่งอื่นเข้าไปหา วิธีการนี้จะช่วยลดความยุ่งยากในขั้นตอนการปรับระดับเสียง (Pitch adjustment) โดยขจัดความยุ่งยากในการเลือก BW และจากนั้นทำการทดสอบเปรียบเทียบการหาค่าเหมาะที่สุดโดยใช้ฟังก์ชั่นทดสอบที่มีลักษณะแตกต่างกัน ได้แก่ Sphere function, Schwefel's Problem, Step function, Rosenbrock function, Rotated hyper-ellipsoid function, Generalized Swefel's Problem, Rastrigin function, Ackley's function, Griewank function และ Six-Hump Camel-Back function ระหว่างการค้นหาความบรรสานแบบดั้งเดิม การค้นหาความบรรสานแบบปรับปรุง และวิธีการค้นหาความบรรสานแบบโกลบอลเบส ซึ่งพบว่าวิธีดังกล่าวสามารถหาคำตอบได้ดีที่สุด อีกทั้งยังสามารถใช้ได้ดีทั้งปัญหาต่อเนื่อง (Continuous problem) และปัญหาไม่ต่อเนื่อง (Discrete problem)

Kaveh และ M.Ahangaran [41] จากการที่การค้นหาความบรรสานแบบวงกว้าง (global search) นั้นไม่สามารถที่จะทำให้เกิดความสมดุลระหว่าง diversification และ intensification ของขั้นตอนการปรับระดับเสียงได้ ได้มีการพัฒนาการค้นหาความบรรสานแบบโซเชียล (Social harmony Search) โดยใช้หลักการกระจายตัวแบบปกติมาตรฐาน (Normal distribution) ในการเลือกค่าเหมาะที่สุด $N(x_i', \sigma_i')$ $\sigma_i' = \xi \times \sum_{j=1}^{HMS} \frac{|x_i' - x_i^j|}{HMS-1}$ ซึ่งพบว่าในช่วงแรกของการทำซ้ำ จะมีการค้นหาแบบวงกว้าง และในช่วงสุดท้ายของการทำซ้ำ จะมีการค้นหาเฉพาะที่ (local search) ซึ่งจะทำให้การลู่เข้าเป็นไปได้อย่างรวดเร็วอีกทั้งยังให้คำตอบที่ดีขึ้น หลังจากนั้นทำการทดสอบกับฟังก์ชั่นรูปแบบมีข้อจำกัด และรูปแบบไม่มีข้อจำกัด เมื่อนำมาเปรียบเทียบกับการค้นหาความบรรสานแบบดั้งเดิม และการค้นหาความบรรสานแบบปรับปรุง พบว่ามีการลู่เข้าสู่คำตอบได้รวดเร็วที่สุด

ขั้นตอนวิธีการแบบผสมผสาน (Hybrid Algorithms) คือ การนำรูปแบบขั้นตอนวิธีการที่แตกต่างกัน มาผสมผสานกันโดยเลือกจุดเด่น หรือ จุดด้อย ของขั้นตอนวิธีการต่าง ๆ มาใช้งาน

Nguyen, Khang, Nguyen, Phuc และTran [42] ทำการวิจัยขั้นตอนวิธีการแบบผสมผสานระหว่างการค้นหาความบรรสาน (HS) และอาณานิคมผึ้งเทียม (Artificial Colony Bees Algorithm: ABC) เพื่อแก้ไขปัญหาตารางเวลาเรียนของมหาวิทยาลัยในประเทศเวียดนาม จากข้อเสียของขั้นตอนวิธีอาณานิคมผึ้งที่จะลู่เข้าสู่คำตอบเร็วจนเกินไป แต่ขั้นตอนวิธีการค้นหาความบรรสานมีความสามารถในการควบคุมการ intensification และ diversification โดยขั้นตอนของการปรับปรุงเสียง และการสุ่มค่า ดังนั้นจึงมีการผสมผสานของขันตอนวิธีการเกิดขึ้น จากนั้นทำการเปรียบเทียบกับ การค้นหาคำตอบในย่านใกล้เคียงแบบแปรผัน (Variable Neighborhood Search) วิธีค้นหาทาบู (Tabu Search) พบว่าขั้นตอนวิธีการแบบผสมผสานให้ค่าที่ดีกว่าขั้นตอนวิธีการทั้งสอง

Seyedhosseini, Esfahani และ Ghaffari [43] ได้นำเสนอการแก้ปัญหา Portfolio (Portfolio optimization problem) ด้วยขั้นตอนวิธีการแบบผสมผสานระหว่างการค้นหาความบรรสาน (HS) กับอาณานิคมผึ้งเทียม (ABC) ซึ่งทำการปรับปรุงขั้นตอนการค้นหาคำตอบใหม่โดยใช้รูปแบบอาณานิคมผึ้งเทียมในการเริ่มหาแหล่งอาหารจากความทรงจำ ดังสมการ $v_{ij} = x_{ij} + r(x_{index,j} - x_{ij})$ จากนั้นเปรียบเทียบด้านประสิทธิภาพและความแม่นยำกับ การค้นหาความบรรสานแบบดั้งเดิม และขั้นตอนวิธีการเชิงพันธุกรรม (GA) โดยพบว่าขั้นตอนวิธีการแบบผสมผสานให้ค่าที่ดีกว่าที่ยกตัวอย่างมาข้างต้น

แต่อย่างไรก็ตามการปรับปรุงการค้นหาความบรรสานทั้งวิธีการแบบผสมผสานและไม่ผสมผสานส่วนใหญ่จะมีการนำไปประยุกต์ใช้กับวัตถุประสงค์เพียงอย่างเดียวเท่านั้น ซึ่งในการนำการค้นหาความบรรสานไปประยุกต์ใช้งานกับปัญหาหลายวัตถุประสงค์ทางผู้วิจัยจะอธิบายในหัวข้อ 2.2.8

2.2.8 การค้นหาความบรรสานหลายวัตถุประสงค์

ปัจจุบันจะเห็นได้ว่าความต้องการที่หลากหลายมากขึ้น ไม่ว่าจะเป็นทางด้านอุตสาหกรรมที่ต้องต้นทุนในการผลิตน้อยที่สุด แต่ต้องการจำนวนสินค้าที่ผลิตมากที่สุดและมีคุณภาพสูงที่สุด ด้านการออกแบบต้องการใช้วัสดุให้น้อยและมีน้ำหนักเบาที่สุด แต่ให้โครงสร้างที่แข็งแรงที่สุด หรือแม้แต่ทางด้านสาธารณสุขในการจัดตั้งหน่วยกู้ภัยฉุกเฉิน โดยมีความต้องการระยะทางในการช่วยเหลือที่ครอบคลุม แต่มีค่าใช้จ่ายในการดำเนินการน้อยที่สุด และทางด้านอื่น ๆ อีกมากมาย ซึ่งจะเห็นได้ว่าความต้องการดังกล่าวข้างต้นนั้นอยู่ในรูปแบบที่มีหลายวัตถุประสงค์ ดังนั้นเพื่อตอบสนองต่อการใช้

งานจริงทางผู้วิจัยจึงได้ทำการยกตัวอย่างการนำการค้นหาความบรรสานไปใช้งานในฟังก์ชั่นหลายวัตถุประสงค์

Geem และ Han Hwangbo [35] ได้ทำการศึกษาการค้นหาความบรรสาน (HS) ในการออกแบบดาวเทียม งานวิจัยนี้ถือได้ว่าเป็นงานวิจัยชิ้นแรกของ Zong Woo Geem ที่มีการนำมาใช้ในการหาค่าเหมาะที่สุดหลายวัตถุประสงค์ ซึ่งหาค่าการนำความร้อนที่สูงที่สุด และมีน้ำหนักรวมของท่อที่ต่ำที่สุด จากนั้นนำมาทดสอบเปรียบเทียบกับขั้นตอนวิธีการแบบดั้งเดิม (Traditional calculus-based algorithm) ซึ่งพบว่าการค้นหาความบรรสาน สามารถหาค่าเหมาะที่สุดได้ดีกว่าวิธีการดังกล่าวข้างต้น

Swarup และ Sivasubramani [44] ได้นำการค้นหาความบรรสานหลายวัตถุประสงค์ไปใช้แก้ปัญหาการไหลของกำลังไฟฟ้า (Optimal power flow problem) ทดสอบกับระบบมาตรฐาน 30 บัสของ IEEE จากนั้นทำการเปรียบเทียบกับวิธีการ NSGA-II (Non-dominated Sorting Genetic Algorithm : NSGA-II) โดยพบว่าการค้นหาความบรรสานหลายวัตถุประสงค์ จะมีการกระจายตัวของคำตอบเหมาะที่สุดแบบพาเรโต้ (Pareto Optimal) ที่ดีกว่าวิธีการ NSGA-II

Ricart และคณะ [45] ได้นำเสนองานวิจัยที่มีชื่อว่า “Multiobjective Harmony Search Algorithm Proposals” ซึ่งเป็นงานที่นำเสนอการค้นหาความบรรสานสำหรับการหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (MOSH) ซึ่งทำการเปรียบเทียบกับ NSGA-II โดยทดสอบด้วยฟังก์ชั่น ZDT (Zitzler Deb Thiele: ZDT) จำนวน 6 ฟังก์ชั่น ได้แก่ ZDT1 ZDT2 ZDT3 ZDT4 ZDT5 และ ZDT6 ซึ่งแต่ละฟังก์ชั่นจะมีรูปแบบที่แตกต่างกันออกไป convexity, nonconvexity, discreteness and non-uniformity, multimodal และ deceptive characteristics ซึ่งทำการวัดผลโดยใช้ ระยะทางระหว่างพาเรโต้พร้อน และค่าเหมาะที่สุดของพาเรโต้พร้อน (M_1^*) การกระจายตัวของพาเรโต้พร้อน (M_2^*) การขยายออกของพาเรโต้พร้อน (M_3^*) พบว่า MOSH นั้นให้ค่าที่ดีกว่า NSGA-II

Landa-Torres และคณะ [46] ได้นำเสนอการตั้งศูนย์ฉุกเฉิน 24 ชั่วโมงที่เมือง Guadalajara และเมือง Cuenca ประเทศสเปนโดยนำการค้นหาความบรรสานสำหรับค่าเหมาะที่สุดหลายวัตถุประสงค์มาใช้ในการค้นหาคำตอบ อีกทั้งยังมีการผสมผสานเทคนิคในการจัดกลุ่ม (Grouping encoding) และการจัดเรียงลำดับ (Sorting) มาช่วยในการค้นหาคำตอบ โดยพบว่าการนำเทคนิคดังกล่าวมาใช้นั้นสามารถช่วยลดพื้นที่ในการค้นหาลงได้ และยังทำให้การหาค่าเหมาะที่สุดทำได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

ในปัจจุบันการนำการค้นหาความบรรสานหลายวัตถุประสงค์ไปใช้งานนั้นยังคงไม่แพร่หลายเท่าที่ควร โดยอาจจะมีสาเหตุมาจากช่วงเวลาในการเกิดขึ้นของขั้นตอนวิธีการดังกล่าว ที่เกิดขึ้นไม่นานนัก อีกทั้งการวิจัยและพัฒนาส่วนใหญ่ยังคงมุ่งไปในการปรับปรุงขั้นตอนวิธีการ เพื่อค้นหาขีดความสามารถ ในการนำไปใช้งานกับปัญหาวัตถุประสงค์เดียว มากกว่าการนำไปทดสอบกับปัญหาหลายวัตถุประสงค์ แต่อย่างไรก็ตามจากสถานการณ์ในปัจจุบัน ปัญหาเพียงวัตถุประสงค์เดียวนั้นไม่เพียงพออีกต่อไป ดังนั้นทางผู้วิจัยจึงได้ทำการศึกษาการแก้ไขปัญหาหลายวัตถุประสงค์ด้วยการค้นหาความบรรสาน

บทที่ 3
วิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึงลำดับและรายละเอียดขั้นตอนการทดสอบของวิธีดำเนินงานวิจัย การค้นหาความบรรสานสำหรับค่าเหมาะที่สุดหลายวัตถุประสงค์: ทฤษฎี, การประยุกต์ใช้งาน และกรณีศึกษา ซึ่งผู้ทำการวิจัยแบ่งการทดสอบออกเป็น 2 กรณี ได้แก่ การค้นหาความบรรสานสำหรับปัญหาการเดินทางของพนักงานขาย และ การค้นหาความบรรสานสำหรับระบบการวางแผนการประกอบอุปกรณ์อิเล็คทรอนิกส์ โดยจะอธิบายดังหัวข้อต่อไปนี้

3.1 การศึกษางานวิจัย ทฤษฎี หลักการและแนวคิดพื้นฐานที่เกี่ยวข้อง
3.2 ขั้นตอนออกแบบและพัฒนาการค้นหาความบรรสาน
3.3 การทดสอบกับปัญหาการเดินทางของพนักงานขายและการวางแผนการประกอบอุปกรณ์อิเล็คทรอนิกส์
3.5 สรุปผลงานวิจัยและจัดทำรูปเล่มวิทยานิพนธ์

3.1 การศึกษางานวิจัย ทฤษฎี หลักการและแนวคิดพื้นฐานที่เกี่ยวข้อง

ผู้ทำการวิจัยดำทำการศึกษาและค้นคว้างานวิจัย หลักการ ทฤษฎี และแนวคิดพื้นฐานที่เกี่ยวข้อง พร้อมทั้งศึกษาปัญหาสำหรับการทดสอบ โดยมีแนวทางการศึกษาหัวข้อที่เกี่ยวข้องดังนี้

ลำดับแรกทางผู้วิจัยจะทำการศึกษาและค้นคว้าเกี่ยวกับรูปแบบทางคณิตศาสตร์ (Mathematical Model) ซึ่งแสดงให้เห็นถึงการสร้างรูปแบบให้เหมาะสมกับปัญหา ไม่ว่าจะเป็นสำหรับปัญหาวัตถุประสงค์เดียว หรือปัญหาหลายวัตถุประสงค์ ซึ่งมีลักษณะและทฤษฎีที่แตกต่างกัน อีกทั้งยังศึกษาต่อถึง วิธีการหาค่าเหมาะที่สุด เพื่อใช้สำหรับการแก้ปัญหา ไม่ว่าจะเป็นวิธีแม่นยำ (Exact Method) วิธีฮิวริสติกส์ (Heuristic Method) และวิธีเมตาฮิวริสติก (Meta-Heuristics Method)

ต่อมาได้ทำการศึกษาเกี่ยวกับทฤษฎี ขั้นตอนวิธีการค้นหาความบรรสาน (Harmony Search Algorithm: HSA) ซึ่งเป็นหนึ่งในวิธีเมตาฮิวริสติก ที่ถูกพัฒนาขึ้นโดย Zong Woo Geem โดยมีจุดกำเนิดมาจากพฤติกรรมของนักดนตรีที่ต้องการหาเสียงที่ไพเราะที่สุด ซึ่งขั้นตอนวิธีดังกล่าวได้ถูกศึกษาอย่างแพร่หลายทั้งในด้านวิศวกรรม ด้านเทคโนโลยีสารสนเทศ และอีกมากมาย

ลำดับสุดท้ายสำหรับการดำเนินการวิจัยในครั้งนี้ ทางผู้วิจัยได้ศึกษาเกี่ยวกับปัญหาที่นำมาทดสอบ และหาพารามิเตอร์ที่เหมาะที่สุด โดยทำการเลือกปัญหาการเดินทางของพนักงานขาย

(Travelling Salesman Problem: TSP) ซึ่งเป็นหนึ่งในปัญหาที่แพร่หลายอย่างมากในทางวิศวกรรม อีกทั้งทางผู้วิจัยยังนำลักษณะของปัญหาดังกล่าวมาเป็นส่วนหนึ่งสำหรับการวางแผนการประกอบชิ้นส่วนอิเล็คทรอนิกส์ (Planning System for Surface Mount PCB Assembly) ซึ่งจะทำการทดสอบในลำดับถัดไป

3.2 ขั้นตอนออกแบบและพัฒนาการค้นหาความบรรสาน

ในขั้นตอนนี้ จะเป็นส่วนของการออกแบบและพัฒนาตัวโปรแกรม โดยในที่นี้ทางผู้วิจัยได้เลือกใช้โปรแกรม Python 3.5.2 บนเครื่อง Intel® Core™ i7-4720HQ CPU 2.60 GHz processor, with 4.00 GB of RAM ในการออกแบบและพัฒนา ซึ่งมีขั้นตอนดังต่อไปนี้ 1. สร้างวิธีการเข้ารหัส (Encoding) และถอดรหัส (Decoding) 2. สร้างการค้นหาความบรรสาน สำหรับปัญหาที่จะนำเข้ามาทดสอบแต่ละปัญหา 3. ตรวจสอบความถูกต้อง และขั้นตอนสุดท้าย คือ 4. ทำการทดสอบ

3.3 การทดสอบกับปัญหาการเดินทางของพนักงานขาย และการวางแผนการประกอบชิ้นส่วนอุปกรณ์อิเล็คทรอนิกส์

สำหรับงานวิจัยในครั้งนี้ทางผู้วิจัยได้เลือกปัญหาการเดินทางของพนักงานขายมาทำการทดสอบด้วยวิธีการค้นหาความบรรสาน ซึ่งปัญหาดังกล่าวมีลักษณะปัญหา NP-complete อีกทั้งยังเป็นที่นิยมอย่างมากสำหรับการทดสอบกับวิธีเมตาฮิวริสติก และทำการหาค่าพารามิเตอร์ที่เหมาะสมด้วยวิธีการทางสถิติ โดยในที่นี้ทางผู้วิจัยเลือกใช้เทคนิคการออกแบบการทดลอง (Design of Experiment: DOE) และการทดสอบสมมุติฐาน (Hypothesis test) สำหรับการหาค่าพารามิเตอร์ดังกล่าว

ลำดับต่อมาหลังจากการทดสอบด้วยปัญหาการเดินทางของพนักงานขายด้วยวิธีการค้นหาความบรรสาน ทางผู้วิจัยได้นำเอาลักษณะของปัญหาดังกล่าวเข้ามาประยุกต์ใช้ในการวางแผนการประกอบชิ้นส่วนอุปกรณ์อิเล็คทรอนิกส์ ซึ่งเป็นปัญหาหลายวัตถุประสงค์ โดยกำหนดชิ้นส่วนที่ต้องการประกอบ เปรียบเสมือนเมืองแต่ละเมืองที่พนักงานขายจะต้องเดินทาง และเครื่องป้อนส่วนประกอบชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์แทนตัวพนักงานขาย จากนั้นทำการทดสอบเพื่อหาค่าพารามิเตอร์ที่เหมาะสม ด้วยวิธีทางสถิติเช่นเดียวกับการทดสอบปัญหาการเดินทางของพนักงานขาย และพิจารณาคำตอบเหมาะที่สุดด้วยทฤษฎีหลายวัตถุประสงค์

3.4 สรุปผลงานวิจัยและจัดทำรูปเล่มวิทยานิพนธ์

ในส่วนสุดท้ายจะเป็นการวิเคราะห์และสรุปผลที่จากการศึกษางานวิจัยในครั้งนี้ รวมถึงการเสนอแนะแนวทางการปรับปรุงและพัฒนางานวิจัยนี้ต่อไป

บทที่ 4
การค้นหาความบรรสานสำหรับปัญหาการเดินทางของพนังานขาย

จากการศึกษา ความบรรสานสำหรับค่าเหมาะที่สุดหลายวัตถุประสงค์: ทฤษฎี, การประยุกต์ใช้งาน และกรณีศึกษา ในส่วนแรก ผู้วิจัยได้ทำการทดลองหาค่าพารามิเตอร์ที่เหมาะที่สุดสำหรับปัญหาการเดินทางของพนักงานขาย ด้วยวิธีการค้นหาความบรรสาน โดยข้อมูลการทดลองดังกล่าว แสดงในหัวข้อดังต่อไปนี้

4.1 ลักษณะความเป็นมาของปัญหาการเดินทางของพนักงานขาย

4.2 การค้นหาความบรรสาน สำหรับปัญหาการเดินทางของพนักงานขาย

4.3 การหาค่าพารามิเตอร์ที่เหมาะสม

4.4 ผลการทดลองและการวิเคราะห์ผลการทดลอง

4.5 สรุปผลการทดลอง

4.1 ลักษณะความเป็นมาของปัญหาการเดินทางของพนักงานขาย

ปัญหาการเดินทางของพนักงานขาย (The traveling salesman problem: TSP) เป็นปัญหาหนึ่งในการวิจัยดำเนินงาน (Operation Research) ซึ่งเป็นที่สนใจทั้งในทางคณิตศาสตร์ วิทยาศาสตร์ และวิศวกรรมเป็นอย่างมาก [47] โดยปัญหาการเดินทางของพนักงานขายนี้จะสามารถอธิบายอย่างง่าย การที่คนขับรถต้องการเส้นทางการเดินทางที่สั้นที่สุด ที่ผ่านเมืองที่ต้องเดินทางจำนวน m เมือง โดยจะต้องเดินทางผ่านเมืองทุกเมือง และกลับไปยังเมืองเริ่มต้น หรือกล่าวได้ว่า เป็นลักษณะการเดินวนรอบ

จากลักษณะดังกล่าว จะเห็นได้ว่าปัญหาการเดินทางของพนักงานขาย มีลักษณะของตัวปัญหาที่ค่อนข้างง่าย แต่อย่างไรก็ตาม ปัญหานี้กลับยากต่อการหาคำตอบ เนื่องจากจำนวนของคำตอบที่เป็นไปได้ ซึ่งจำนวนเมือง m เมือง จะมีคำตอบที่เป็นไปได้ถึง $(m-1)!/2$ ดังนั้นถ้า มีจำนวนเมือง 15 เมือง จะมีจำนวนคำตอบถึง $(15-1)!/2$ หรือเท่ากับ 6.53×10^{11} คำตอบ ด้วยเหตุนี้เอง ปัญหาการเดินทางของพนักงานขายจึงถูกจัดให้อยู่ในปัญหาลักษณะ NP-complete โดยปัญหาลักษณะนี้เป็นที่นิยมในการใช้ วิธีการแบบเมตาฮิวริสติก ในการหาคำตอบ อาทิเช่น ขั้นตอนวิธีเชิงพันธุกรรม (Genetic algorithm: GA) [48] การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization: PSO) [49] เป็นต้น

แม้ว่าวิธีการเมตาฮิวริสติกข้างต้น จะมีการทำงานที่อย่างมีประสิทธิภาพ แต่ในบางสถานการณ์มีความจำเป็นในการเพื่อลดความยุ่งยาก สำหรับการทำงานสถานการณ์ในชีวิตจริง ดังนั้น การกำหนด พารามิเตอร์ (Parameters) ที่เหมาะสมด้วยวิธีการทางสถิติ จึงถือเป็นสิ่งเป็นอย่างยิ่ง โดยจะกล่าวต่อไปในหัวข้อถัดไปตามลำดับ

4.2 การค้นหาความบรรสาน สำหรับปัญหาการเดินทางของพนักงานขาย

ในลำดับต่อมา ทางผู้วิจัยจะทำการอธิบายถึงลักษณะของขั้นตอนวิธีการค้นหาความบรรสาน และการนำขั้นตอนวิธีดังกล่าวมาใช้ในปัญหาการเดินทางของพนักงานขาย ซึ่งแบ่งการอธิบายออกเป็น 2 ส่วน คือ 4.2.1 การค้นหาความบรรสาน (Conventional Harmony Search) และ 4.2.2 การค้นหาความบรรสานสำหรับปัญหาการเดินทางของพนักงานขาย (Harmony Search for Traveling Salesman Problems)

4.2.1 การค้นหาความบรรสาน

แนวคิดการค้นหาความบรรสาน เกิดขึ้นมาจากปรากฏการณ์ที่พบในการแสดงดนตรี ซึ่งปกติแล้วนักดนตรีจะพยายามค้นหาเสียงที่ไพเราะที่สุดสำหรับการแสดง ด้วยพฤติกรรมดังกล่าวจะเปรียบเทียบได้กับการหาค่าเหมาะที่สุด ดังนั้น Zoo Woo Geem จึงกำหนดขั้นตอนการทำงานการค้นหาความบรรสาน สำหรับปัญหาไม่ต่อเนื่อง (Discrete Problem) โดยประกอบไปด้วย 5 ขั้นตอนสำคัญ ดังนี้

ขั้นตอนที่ 1 การกำหนดปัญหาและขั้นตอนวิธีการ

ขั้นตอนแรกจะกำหนดปัญหาและข้อกำหนด ดังสมการที่ (1) และกำหนดค่าพารามิเตอร์ของขั้นตอนวิธีการค้นหาความบรรสาน ซึ่งประกอบไปด้วย ความน่าจะเป็นในการใช้หน่วยความจำ (HMCR), ความน่าจะเป็นในการปรับระดับเสียง (PAR), ขนาดของหน่วยความจำความบรรสาน (HMS) และจำนวนการทำซ้ำ (T)

$$\begin{aligned} &\text{Minimize } f(x) \\ &\text{subject to } Lx_i \le x_i \le Ux_i, \quad \forall i \in \{1,2,\dots,N\} \end{aligned} \tag{1}$$

ขั้นตอนที่ 2 เริ่มต้นสร้างหน่วยความจำความบรรสาน

ขั้นตอนต่อมา หลังจากทำการกำหนดค่าของพารามิเตอร์ต่าง ๆ จะทำการสร้างหน่วยความจำความบรรสาน (HM) โดยทำการสุ่มค่าขึ้นมาในลักษณะเวกเตอร์ $HM = [x_1\ x_2\ \dots\ x_N]$ ซึ่งค่าดังกล่าวจะถูกนำไปคำนวณหาค่าฟังก์ชั่นวัตถุประสงค์ และเก็บไว้ในหน่วยความจำความบรรสาน

$$HM = \left[\begin{array}{ccccc|c} x_1^1 & x_2^1 & \dots & x_{N-1}^1 & x_N^1 & f(\vec{x}^1) \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 & f(\vec{x}^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} & f(\vec{x}^{HMS-1}) \\ x_1^{HMS} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} & f(\vec{x}^{HMS}) \end{array}\right]$$

ขั้นตอนที่ 3 การปรับปรุงหน่วยความจำความบรรสาน

หลังจากขั้นตอนก่อนหน้ามีการสร้างหน่วยความจำบรรสาน ซึ่งหน่วยความจำดังกล่าว จะถูกปรับปรุงโดยเวกเตอร์ความบรรสาน $x' = [x_1'\ x_2' \dots x_{N-1}'\ x_N']$ ที่ได้ปรับปรุงจากกฎ 3 ข้อ คือ 1.กฎความน่าจะเป็นในการใช้หน่วยความจำ (HMCR Rule) 2.กฎความน่าจะเป็นในการปรับระดับเสียง (PAR Rule) 3.กฎการสุ่มค่า (Randomization Rule) ซึ่งกฎดังกล่าวมีความคล้ายคลึงกับการปรับปรุงเสียงของนักดนตรี ที่มีลักษณะ

สำหรับกฎความน่าจะเป็นในการใช้หน่วยความจำ และกฎการสุ่มค่า จะแสดงโดยค่าความน่าจะเป็นในการใช้หน่วยความจำ (HMCR) แสดงดังสมการ (2)

$$\begin{array}{l} \text{If } rand(0,1) < HMCR \text{ then} \\ x_i' \leftarrow x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} \text{ Else} \\ x_i' \leftarrow x_i' \in \{x_{i,\min}, x_{i,\max}\} \text{ End If} \end{array} \quad (2)$$

จากนั้นกฎความน่าจะเป็นในการปรับระดับเสียงจะถูกใช้ในลำดับต่อมา โดยการเลือกการปรับปรุงเวกเตอร์ความบรรสานหรือไม่ ซึ่งแสดงจากค่าความน่าจะเป็นในการปรับระดับเสียง (PAR) แสดงดังสมการ (3)

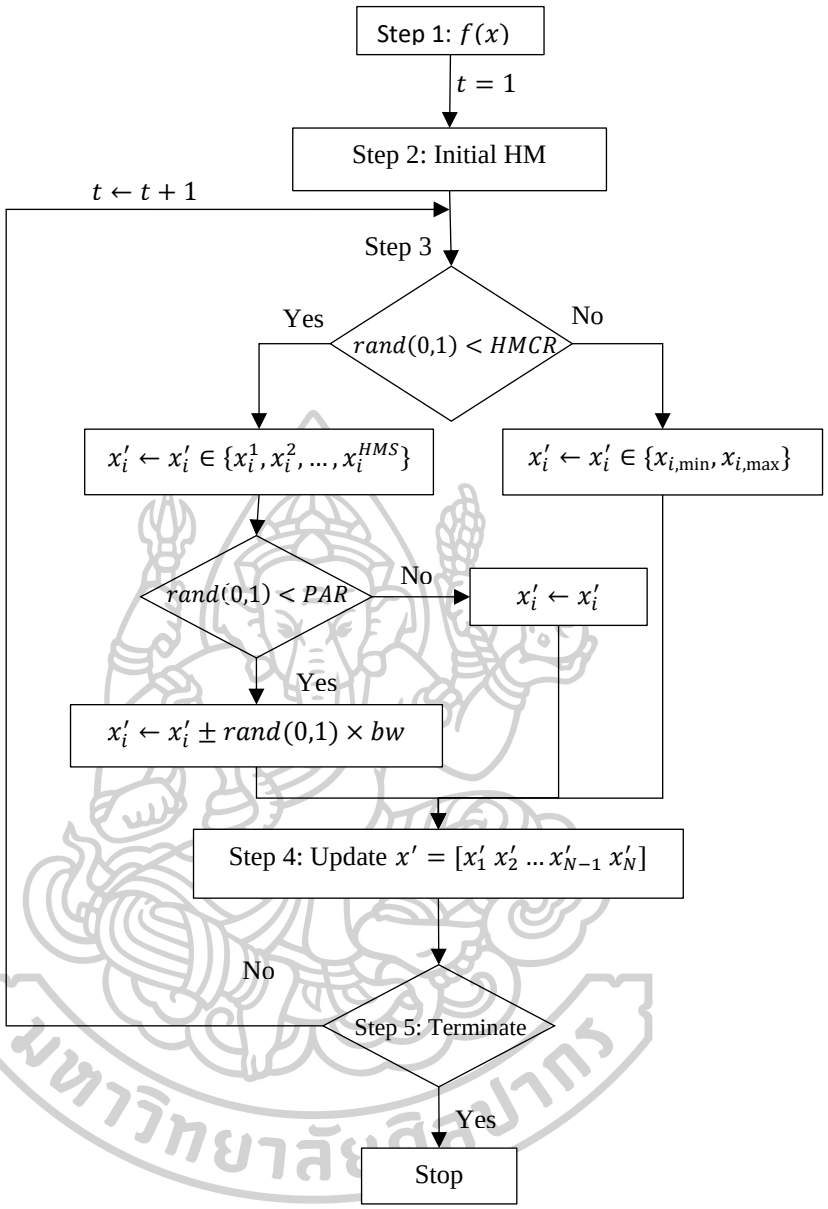
$$\begin{array}{l} \text{If } rand(0,1) < PAR \text{ then} \\ x_i' \leftarrow x_i' \pm rand(0,1) \times bw \text{ Else} \\ \quad x_i' \leftarrow x_i' \text{ End If} \end{array} \quad (3)$$

ขั้นตอนที่ 4 แทนที่คำตอบลงในหน่วยความจำความบรรสาน

เวกเตอร์ความบรรสานใหม่ $x' = [x'_1\ x'_2 \ldots x'_{N-1}\ x'_N]$ นั้นจะถูกนำมาพิจารณาค่าของคำตอบเพื่อทำการเปรียบเทียบกับค่าเดิมที่อยู่ในหน่วยความจำบรรสาน (HM) โดยถ้าค่าที่ได้ มีค่าที่ดีกว่าจะถูกแทนที่ลงใน หน่วยความจำบรรสาน ในขณะที่ค่าที่แย่กว่าจะถูกนำออกไป

ขั้นตอนที่ 5 ตรวจสอบเงื่อนไขการหยุด

ลำดับสุดท้ายจะเป็นการทำซ้ำตั้งแต่ ขั้นตอนที่ 3 จนถึงขั้นตอนที่ 5 ตามจำนวนค่าพารามิเตอร์จำนวนการทำซ้ำ อย่างไรก็ตาม มีบางกรณีที่จะมีเงื่อนไขการหยุดที่แตกต่างออกไป ตัวอย่างเช่น เงื่อนไขการหยุด เมื่อไม่มีการพัฒนาของคำตอบสุดท้าย เป็นต้น



รูปที่ 12 การทำงานของขั้นตอนวิธีการค้นหาความบรรสาน

4.2.2 การค้นหาความบรรสานสำหรับปัญหาการเดินทางของพนักงานขาย

ดังที่ได้กล่าวมาในหัวข้อ 4.1 จะเห็นได้ว่า ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem) เป็นปัญหาที่ง่ายต่อความเข้าใจ แต่ยากต่อการหาคำตอบ ซึ่งมีรูปแบบทางคณิตศาสตร์ของปัญหาการเดินทางของพนักงานขาย [48] ดังนี้

$$\min z = \sum_{j=1}^{m} \sum_{i=1}^{m} c_{ij} x_{ij} \tag{4}$$

s.t.

$$\sum_{j=1}^{m} x_{ij} = 1, \ \ \forall i \in \{1,..,m\} \tag{5}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \ \ \forall j \in \{1,..,m\} \tag{6}$$

$$\sum_{i \in K} \sum_{j \in K} x_{ij} \leq |K| - 1, \ \ \forall K \subset \{1,..,m\} \tag{7}$$

โดยที่ K คือ จำนวนเมืองที่อยู่ในเส้นทาง c_{ij} คือ ค่าใช้จ่ายในการเดินทางจาก เมือง i ไปสู่เมือง j และ x_{ij} คือ ตัวแปรตัดสินใจ โดยมีค่า เท่ากับ 1 เมื่อมีการเดินทางจาก เมือง i ไปสู่ เมือง j ($\overrightarrow{ij}$) และเท่ากับ 0 เมื่อไม่มีการเดินทางจากเมือง i ไปสู่ เมือง j

อย่างไรก็ตาม ทางผู้วิจัยต้องการที่จะประยุกต์ใช้ การค้นหาความบรรสาน สำหรับการหาคำตอบ ปัญหาการเดินทางของพนักงานขาย ดังนั้นวิธีการเข้ารหัส (Encoding) และถอดรหัส (Decoding) ได้ถูกนำมาใช้ โดยอธิบายได้ดังรูปที่ 13

(a)

1	2	3	4	5	6
.127	.159	.251	.010	.012	.570

⇩

(b)

4	5	1	2	3	6
.010	.012	.127	.159	.251	.570

รูปที่ 13 แผนภาพการเข้ารหัส (Encoding) และถอดรหัส (Decoding)

จากรูปที่ 13 จะเห็นได้ว่ามีจำนวนเมืองทั้งหมด 6 เมืองที่จะต้องเดินทาง (m = 6) เริ่มจากเวกเตอร์ (a) แถวด้านบน คือ เมืองแต่ละเมืองตั้งแต่ 1 ถึง 6 และแถวด้านล่างคือ ลำดับเลข ซึ่งลำดับเลขจะถูกกำหนดขึ้นมา โดยมีค่าอยู่ระหว่าง 0 และ 1 จากนั้น เวกเตอร์ (b) จะแสดงการจัดเรียงใหม่ตามลำดับจากน้อยไปมาก ซึ่งจะเห็นได้ว่า เมืองแรกที่เดินทางผ่านคือ เมือง 4 ซึ่งถูกเรียงลำดับมาแทนเมือง 1 จากเวกเตอร์ (a) ดังนั้นในการเดินทางจะผ่านเมือง 4-5-1-2-3-6 ตามลำดับ

4.3 การหาค่าพารามิเตอร์ที่เหมาะสม

สำหรับการหาค่าพารามิเตอร์ที่เหมาะสม ทางผู้วิจัยได้แบ่งออกเป็น 2 ส่วน คือ 4.3.1 การตั้งค่าพารามิเตอร์การค้นหาความบรรสาน (Harmony Search Parameter Setting) และ 4.3.2 การปรับแต่งพารามิเตอร์การค้นหาความบรรสาน (Harmony Search Parameter Tuning)

4.3.1 การตั้งค่าพารามิเตอร์การค้นหาความบรรสาน

จากที่ได้กล่าวมาข้างต้น ทางผู้วิจัยจะทำการใช้วิธีการทางสถิติเพื่อหา พารามิเตอร์ที่เหมาะสม ทั้งนี้เพื่อทำการตั้งค่าพารามิเตอร์ดังกล่าว ได้อย่างเหมาะสมสำหรับการแก้ปัญหาครั้งนี้ ดังนั้นทางผู้วิจัยได้ทำการออกแบบการทดลอง (General Full Factorial Design) สำหรับพารามิเตอร์ทั้ง 4 ชนิด ได้แก่ พารามิเตอร์ความน่าจะเป็นในการใช้หน่วยความจำ (Harmony Memory Considering Rate: HMCR), พารามิเตอร์ความน่าจะเป็นในการปรับระดับเสียง (Pitch Adjustment Rate: PAR), พารามิเตอร์ขนาดของหน่วยความจำความบรรสาน (Harmony Memory Size: HMS) และพารามิเตอร์จำนวนการทำซ้ำ (Iteration number: T) โดยแต่ละพารามิเตอร์จะมี 3 ระดับ ดังแสดงในตารางที่ 2

ตารางที่ 2 การปรับตั้งพารามิเตอร์

Parameter	Number of levels	Setting
$HMCR$	3	0.3, 0.6, 0.9
PAR	3	0.3, 0.6, 0.9
HMS	3	30, 60, 90
$Iteration\ number$	3	1000, 3000, 5000

จากการปรับตั้งพารามิเตอร์ดังกล่าวข้างต้น ทางผู้วิจัยจึงจำเป็นที่จะต้องทดสอบทั้งหมด 81 $(3 \times 3 \times 3 \times 3)$ ชุดการทดสอบ และแต่ละชุดการทดสอบจะมีการทำซ้ำ ชุดละ 20 ครั้ง ดังนั้นสำหรับปัญหาการเดินทางของพนักงานขาย จะมีการทดสอบการทดสอบทั้งหมด 1,620 ครั้ง

4.3.2 การปรับแต่งพารามิเตอร์การค้นหาความบรรสาน

ในการทดสอบปัญหาการเดินทางของพนักงานขาย ด้วยการค้นหาความบรรสาน จะถูกทำการทดสอบลงบนโปรแกรม Python 3.5.2 64 bit บนเครื่อง Intel® Core™ i7-4720HQ CPU 2.60 GHz processor, with 4.00 GB of RAM อีกทั้งทางผู้วิจัยได้ทำการเลือกชุดปัญหาทดสอบจำนวน 3 ชุดทดสอบ [50] ได้แก่ ชุดปัญหาทดสอบแรก UK12 โดยชุดปัญหานี้จะอธิบายเกี่ยวกับ จำนวนเมือง 12 เมืองในประเทศอังกฤษ ชุดปัญหาต่อมา WG59 ชุดปัญหานี้จะกล่าวถึงเมืองจำนวน 59 เมือง ใน

สหพันธ์สาธารณรัฐเยอรมนี และชุดปัญหาสุดท้าย SGB128 ซึ่งเป็นชุดทดสอบที่มีจำนวนของข้อมูลจำนวนมาก โดยในชุดทดสอบนี้จะประกอบไปด้วยจำนวนเมือง 128 เมือง ในทวีปอเมริกาเหนือ

4.4 ผลการทดลองและการวิเคราะห์ผลการทดลอง

จากการทดลอง กำหนดค่าของพารามิเตอร์ สำหรับชุดปัญหาทดสอบ 3 ชุดปัญหา พบว่า มีค่าของพารามิเตอร์ที่มีผลต่อกันอย่างมีนัยสำคัญ (significant parameter) ซึ่งค่าดังกล่าวจะแสดงในตารางต่อไปนี้

ตารางที่ 3 ค่า p-value จากการทดสอบด้วยการค้นหาความบรรสาน สำหรับชุดปัญหา UK12

Source	p-Value
$HMCR$	0.000
PAR	0.329
HMS	0.064
$iteration$	0.000
$HMCR \times PAR$	0.340
$HMCR \times HMS$	0.535
$HMCR \times iteration$	0.934
$PAR \times HMS$	0.412
$PAR \times iteration$	0.448
$HMS \times iteration$	0.442
$HMCR \times PAR \times HMS$	0.344
$HMCR \times PAR \times iteration$	0.710
$HMCR \times HMS \times iteration$	0.490
$PAR \times HMS \times iteration$	0.404
$HMCR \times PAR \times HMS \times iteration$	0.425

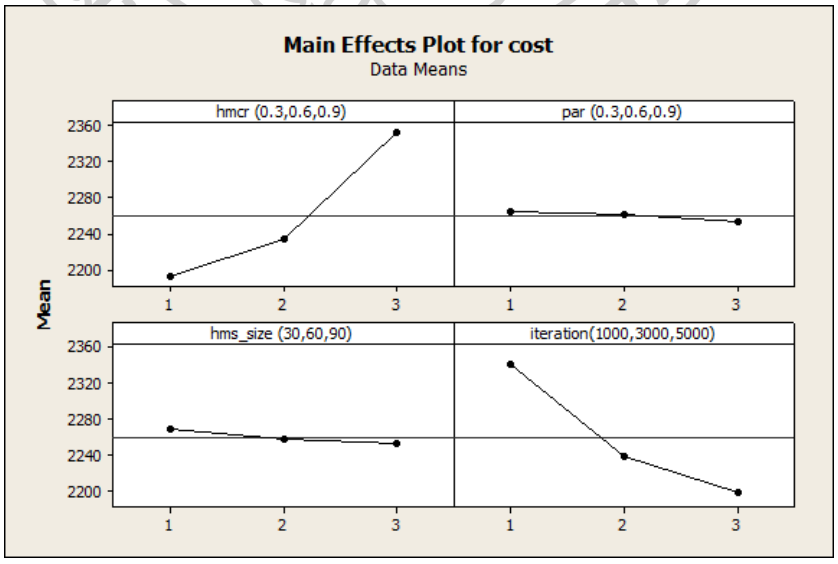
ตารางที่ 4 ค่า p-value จากการทดสอบด้วยการค้นหาความบรรสาน สำหรับชุดปัญหา WG59

Source	p-Value
$HMCR$	0.000
PAR	0.091
HMS	0.138
$iteration$	0.000
$HMCR \times PAR$	0.487
$HMCR \times HMS$	0.794
$HMCR \times iteration$	0.705
$PAR \times HMS$	0.971
$PAR \times iteration$	0.759
$HMS \times iteration$	0.192
$HMCR \times PAR \times HMS$	0.413
$HMCR \times PAR \times iteration$	0.558
$HMCR \times HMS \times iteration$	0.240
$PAR \times HMS \times iteration$	0.067
$HMCR \times PAR \times HMS \times iteration$	0.279

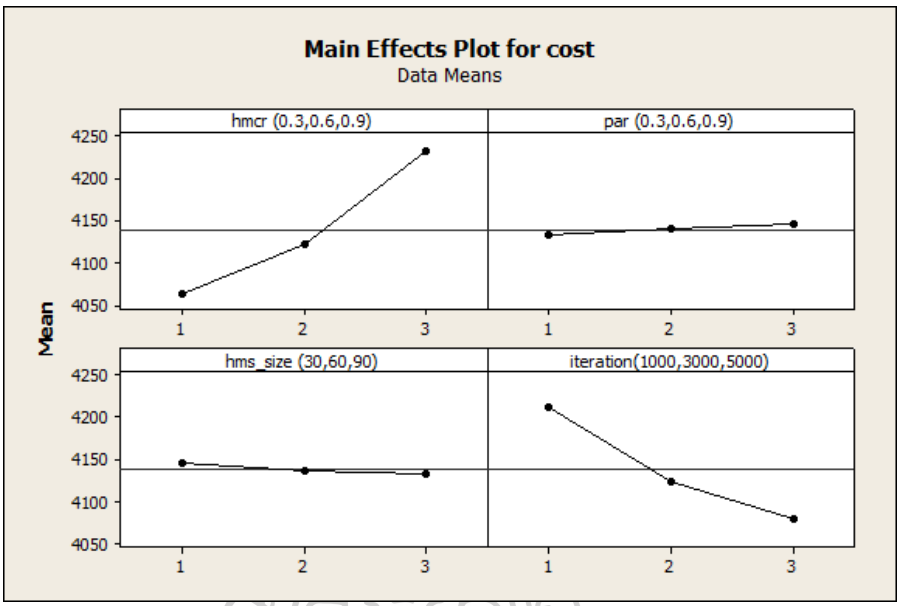
ตารางที่ 5 ค่า p-value จากการทดสอบด้วยการค้นหาความบรรสาน สำหรับชุดปัญหา SGB128

Source	p-Value
$HMCR$	0.000
PAR	0.157
HMS	0.829
$iteration$	0.000
$HMCR \times PAR$	0.202
$HMCR \times HMS$	0.855
$HMCR \times iteration$	0.627
$PAR \times HMS$	0.893
$PAR \times iteration$	0.389
$HMS \times iteration$	0.500
$HMCR \times PAR \times HMS$	0.356
$HMCR \times PAR \times iteration$	0.893
$HMCR \times HMS \times iteration$	0.617
$PAR \times HMS \times iteration$	0.529
$HMCR \times PAR \times HMS \times iteration$	0.754

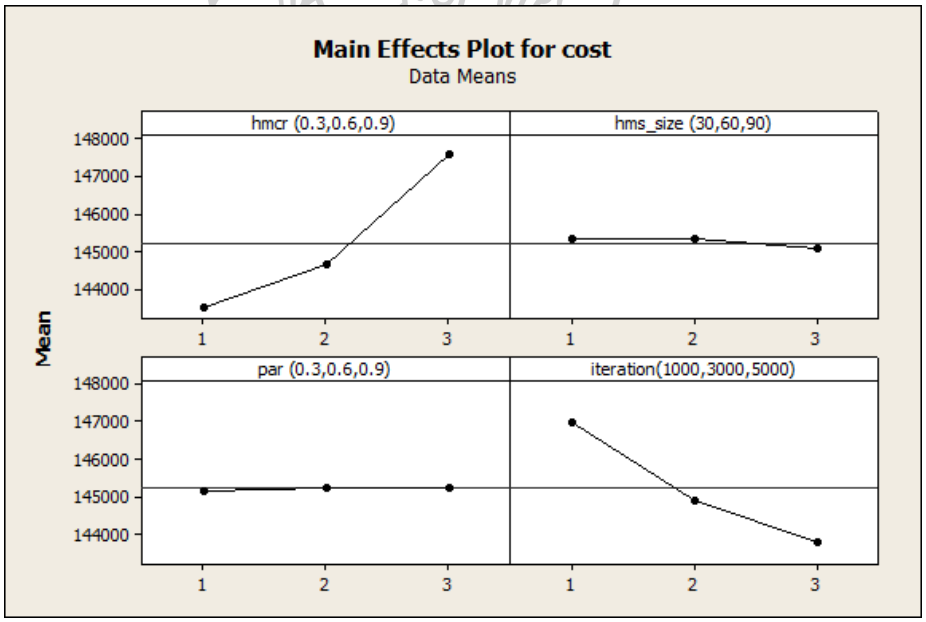
จากตารางที่ 3 4 และ 5 ที่ระดับนัยสำคัญ เท่ากับ 0.05 ($\alpha = 0.05$) พบว่าพารามิเตอร์ที่มีผลกันอย่างมีนัยสำคัญ คือ พารามิเตอร์ความน่าจะเป็นในการใช้หน่วยความจำ (Harmony Memory Considering Rate: HMCR) และพารามิเตอร์จำนวนการทำซ้ำ (Iteration number) ด้วยเหตุนี้เองทางผู้วิจัยจึงมุ่งเน้นไปในการปรับตั้งค่าที่เหมาะสมสำหรับพารามิเตอร์ดังกล่าว โดยทำการพิจารณาจากผลกระทบปัจจัยหลัก (Main effect) ผลปรากฏว่า ค่าที่เหมาะสมสำหรับพารามิเตอร์ความน่าจะเป็นในการใช้หน่วยความจำ มีค่าเท่ากับ 0.3 และ 5,000 รอบ สำหรับพารามิเตอร์จำนวนการทำซ้ำ ดังแสดงในรูปที่ 14 15 และ 16 ตามลำดับ



รูปที่ 14 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดปัญหาทดสอบ UK12



รูปที่ 15 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดปัญหาทดสอบ WG59



รูปที่ 16 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดปัญหาทดสอบ SBG128

ส่วนต่อมาทางผู้วิจัยได้ทำการแสดงถึงเวลาในการทดสอบ (Computational time) สำหรับชุดปัญหาทดสอบทั้ง 3 ชุดปัญหา แสดงดังตารางที่ 6

ตารางที่ 6 แสดงเวลาการทดสอบ

	1000 iterations	3000 iterations	5000 iterations
UK12	0.56	1.72	2.91
WG59	1.56	4.48	7.57
SGB128	5.43	14.83	24.81

จากตารางที่ 5 จะเห็นได้ว่าเวลาในการทดสอบจะขึ้นอยู่กับ ขนาดของชุดปัญหาที่นำมาทดสอบ และพารามิเตอร์จำนวนการทำซ้ำ โดยในความเป็นจริงแล้วขนาดที่ใหญ่ขึ้นของชุดปัญหาทดสอบส่งผลต่อเวลาในการทดสอบที่มากขึ้น ในทำนองเดียวกันกับการที่มีจำนวนรอบการทำซ้ำที่มากขึ้นจะส่งผลต่อเวลาในการทดสอบที่มากขึ้นเช่นกัน

ขั้นตอนต่อมาทางผู้วิจัยมีความสนใจ ในสมมุติฐาน จำนวนการทำซ้ำ (Iteration number) ที่มากขึ้น ค่าจากการทดสอบจะมีค่าที่ดีขึ้น ในที่นี้ทางผู้วิจัยจะทำการทดสอบสมมุติฐานนี้ด้วย การวิเคราะห์ความแปรปรวน. (Analysis of Variance: ANOVA) และเลือกใช้ชุดปัญหา WG59 สำหรับการทดสอบ โดยพารามิเตอร์สำหรับการวิเคราะห์ ทางผู้วิจัยได้ทำการกำหนดค่าต่าง ๆ ดังนี้ $HMCR = 0.3$, $PAR = 0.3$, $HMS = 60$ โดยในการทดลองนี้จะกำหนด จำนวนการทำซ้ำที่แตกต่างกัน คือ 2500, 5000, 7500 และ 10000 ตามลำดับ แต่ละการทดลองจะมีการทำซ้ำ จำนวน 20 ครั้ง ดังแสดงในตารางที่ 7

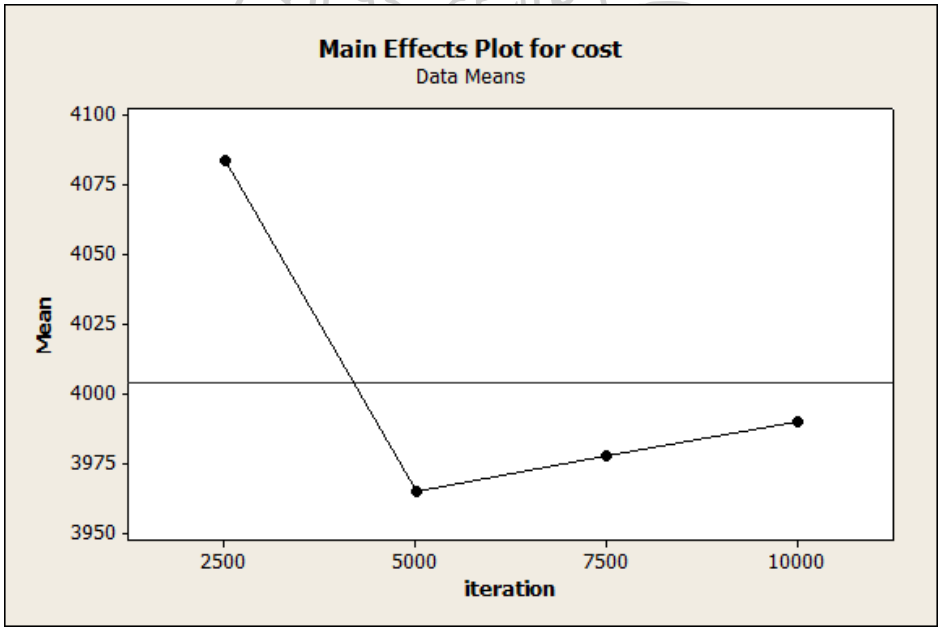
ตารางที่ 7 ผลการวิเคราะห์ความแปรปรวน (ANOVA) สำหรับปัญหา WG59

Source of Variance	Degree of Freedom	Sum of Squares	Mean Square	F_0	p-value
iterations	3	174553	58184	565	0.002
Error	76	783070	10304		
Total	79	957623			

จากตารางที่ 7 จะเห็นได้ว่าจำนวนการทำซ้ำมีผลต่อคำตอบที่ได้ อย่างมีนัยสำคัญ แต่อย่างไรก็ตาม ทางผู้วิจัยได้ทำการทดสอบต่อไป ด้วยวิธีการตูกี (Turkey test) เพื่อแสดงให้เห็นถึงความแตกต่างระหว่างจำนวนการทำซ้ำที่ต่างกัน (2500, 5000, 7500, และ 10000) ซึ่งจากผลการทดสอบในรูปที่ 17 ทางผู้วิจัยพบว่า สำหรับจำนวนการทำซ้ำ 5000, 7500 และ 10000 ถูกจัดให้อยู่ในกลุ่มเดียวกัน ขณะที่จำนวนการทำซ้ำ 2500 ถูกจัดอยู่ในกลุ่มที่แตกต่างกัน อีกทั้งจำนวนการทำซ้ำที่มากขึ้นเกินกว่า 5000 นั้นไม่ส่งผลให้ได้คำตอบที่ดีขึ้น ดังนั้นสำหรับการทดสอบปัญหาการเดินทางของพนักงานขายในครั้งนี้ ค่าพารามิเตอร์จำนวนการทำซ้ำ (Iteration number) เท่ากับ 5000 ถือว่ามีค่าเหมาะสมที่สุด ดังรูปที่ 18 แสดงการยืนยันว่าการตั้งค่าที่เหมาะสมของจำนวนการซ้ำคือ 5000

```
 Individual 95% CIs For Mean Based on
                        Pooled StDev
Level   N    Mean  StDev  -------+---------+---------+---------+--
 2500  20  4083.6   86.9                       (-------*------)
 5000  20  3965.2  126.8  (-------*------)
 7500  20  3977.6   89.6    (-------*------)
10000  20  3990.1   97.7      (-------*-------)
                           -------+---------+---------+---------+--
                               3960      4020      4080      4140

Pooled StDev = 101.5

Grouping Information Using Tukey Method

iteration   N    Mean  Grouping
 2500      20  4083.6  A
10000      20  3990.1    B
 7500      20  3977.6    B
 5000      20  3965.2    B

Means that do not share a letter are significantly different.
```

รูปที่ 17 ผลการทดสอบด้วยวิธีตูกี กับจำนวนการทำซ้ำที่แตกต่างกัน



รูปที่ 18 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดปัญหาทดสอบ WG59

4.4 สรุปผลการทดลอง

วิธีการเมตาฮิวริสติก เป็นวิธีการที่มีประสิทธิภาพวิธีการหนึ่งสำหรับการแก้ปัญหา NP-Complete อย่างไรก็ตาม ข้อเสียสำหรับวิธีการดังกล่าว คือการหาค่าของ พารามิเตอร์ที่เหมาะสมสำหรับการแก้ปัญหา ดังนั้นสำหรับการศึกษาในครั้งนี้ ทางผู้วิจัยได้ทำการศึกษาวิธีการทางสถิติเพื่อหาค่าพารามิเตอร์ที่เหมาะสมสำหรับปัญหาการเดินทางของพนักงานขาย ซึ่งชุดปัญหาในการทดสอบจะถูกแก้ไขด้วยวิธีการค้นหาความบรรสานโดยทำการเปลี่ยนพารามิเตอร์ที่แตกต่างกันออกไปด้วยวิธีการทางสถิติ

ทางผู้วิจัยพบว่า พารามิเตอร์ความน่าจะเป็นในการใช้หน่วยความจำ (HMCR) และพารามิเตอร์จำนวนการทำซ้ำ (T) มีผลต่อคำตอบที่ได้รับอย่างมีนัยยะสำคัญ นอกจากนี้ค่าที่ได้จากผลกระทบจากปัจจัยหลัก (Main effect plot) แสดงให้เห็นว่า ค่าสำหรับพารามิเตอร์ความน่าจะเป็นการใช้หน่วยความจำ และพารามิเตอร์จำนวนการทำซ้ำ ควรจะมีค่าเท่ากับ 0.3 และ 5000 ตามลำดับ ลำดับต่อมา การปรับตั้งค่าพารามิเตอร์จำนวนการทำซ้ำที่มากเกิน 5000 จะไม่สงผลให้คำตอบดีขึ้น โดยยืนยันได้จากการทดสอบด้วยวิธีการตูกี (Tukey’s test)

มหาวิทยาลัยศิลปากร

บทที่ 5
การค้นหาความบรรสานสำหรับระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์

จากบทก่อนหน้า ทางผู้วิจัยได้มีการศึกษาการค้นหาความบรรสานสำหรับปัญหาการเดินทางของพนักงานขาย ซึ่งจากลักษณะปัญหาดังกล่าวข้างต้น ทางผู้วิจัยได้นำมาประยุกต์ใช้กับ ระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ เพื่อหาค่าเหมาะที่สุดด้วยวิธีการค้นหาความบรรสาน อีกทั้งทำการทดลองปรับค่าพารามิเตอร์ด้วยวิธีการทางสถิติ โดยผลการทดลองจะแสดงดังต่อนี้

5.1 ลักษณะความเป็นมาของระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์

5.2 การค้นหาความบรรสาน สำหรับระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์

5.3 การหาค่าพารามิเตอร์ที่เหมาะสม

5.4 ผลการทดลองและการวิเคราะห์ผลการทดลอง

5.5 สรุปผลการทดลอง

5.1 ลักษณะความเป็นมาของระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์

การประกอบแผงวงจรอิเล็กทรอนิกส์ (Printed Circuit Board Assembly: PCBA) ถือได้ว่าเป็นหัวใจหลักของการผลิตสำหรับอุตสาหกรรมอุปกรณ์อิเลคทรอนิคส์ ในปัจจุบันการประกอบแผงวงจรอิเล็กทรอนิกส์ จะต้องมีปริมาณของส่วนประกอบที่เพิ่มขึ้น มีการทำงานที่ดีขึ้น อีกทั้งต้องมีขนาดที่เล็กลง และมีความเชื่อมั่น (Reliability) มากขึ้น ดังนั้นกระบวนการผลิต จึงมุ่งเน้นในการประกอบแผงวงจรอิเล็กทรอนิกส์แบบอัตโนมัติ ด้วยเหตุนี้เองการประกอบแผงวงจรอิเล็กทรอนิกส์จะมีความซับซ้อนที่เพิ่มขึ้น อีกทั้งยังส่งผลต่อการวางแผนลำดับการประกอบอุปกรณ์อิเล็กทรอนิกส์จะทำได้ยากยิ่งขึ้น [51]

การวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ (PCB Assembly Planning) เป็นการทำงานโดยมีความสัมพันธ์ระหว่าง การออกแบบ (Design) และการผลิต (Manufacturing) โดยการแปลงข้อมูลการออกแบบแผงวงจรไปสู่การจัดลำดับการประกอบ ในช่วงที่ผ่านมามีการใช้วิธีการต่าง ๆ เพื่อมาช่วยในการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ เช่น การสร้างรูปแบบระบบการวางแผน เข้ากับปัญหา “Rural Postman” (The Rural Postman Problem: RPP) โดยใช้ขั้นตอนวิธีที่ถูกพัฒนาขึ้นโดยพวกเขาทั้งสองคนในการหาคำตอบ [52] หรือการนำปัญหาการเดินทางของพนักงานขาย (TSP) มาประยุกต์ใช้ โดยใช้ขั้นตอนวิธีเชิงพันธุกรรม (GA) ในการหาคำตอบ [53]

สำหรับในบทนี้ทางผู้วิจัยดำเนินการพัฒนาการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์เข้ากับปัญหาการเดินทางของพนักงานขาย โดยทำการทดสอบด้วยขั้นตอนวิธีการค้นหาความบรรสานและสุดท้ายทำการทดสอบด้วยวิธีการทางสถิติ ซึ่งจะกล่าวต่อไปในหัวข้อถัดไปตามลำดับ

5.2 การค้นหาความบรรสาน สำหรับการวางแผนการประกอบวงจรพิมพ์

ในหัวข้อนี้ทางผู้วิจัยจะกล่าวถึง ขั้นตอนวิธีการค้นหาความบรรสาน และลักษณะของการวางแผนประกอบวงจรพิมพ์ อีกทั้งยังแสดงถึงการนำปัญหาการเดินทางของพนักงานขายเข้ามาประยุกต์ใช้ โดยแบ่งหัวข้อในการอธิบายออกเป็น 4 ส่วน ได้แก่ 5.2.1 การค้นหาความบรรสาน 5.2.2 กระบวนการประกอบแผงวงจรอิเล็กทรอนิกส์ 5.2.3 รูปแบบของฟังก์ชั่นหลายวัตถุประสงค์ และ 5.2.4 การสร้างรูปแบบระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ เข้ากับปัญหาการเดินทางของพนักงานขาย

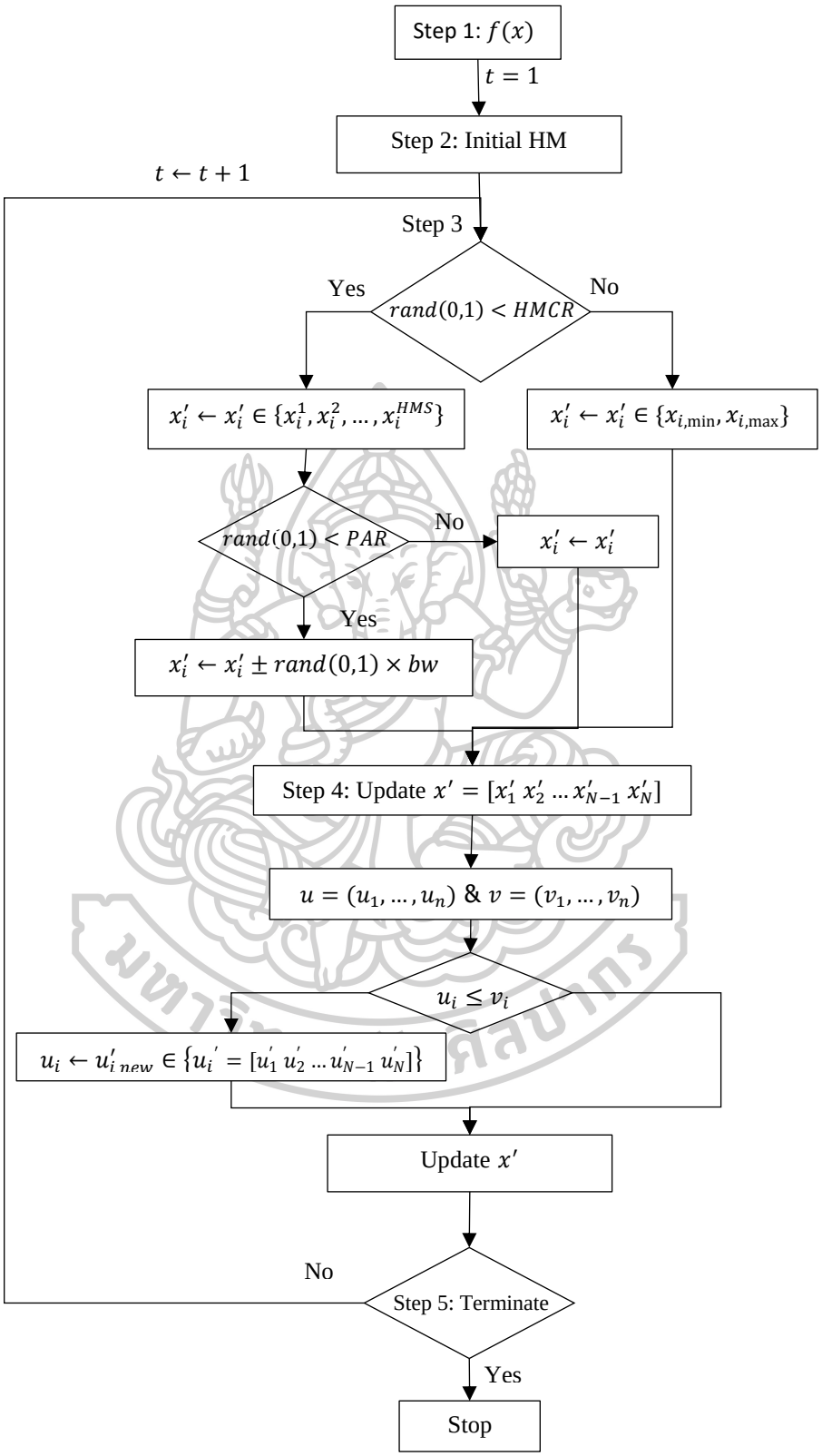
5.2.1 การค้นหาความบรรสาน

สืบเนื่องจากในหัวข้อที่ 4.2.1 ได้กล่าวถึงขั้นตอนการทำงานของ การค้นหาความบรรสาน ซึ่งประกอบไปด้วย 5 ขั้นตอนสำคัญ ได้แก่ 1. การกำหนดปัญหาและขั้นตอนวิธีการ 2. เริ่มต้นสร้างหน่วยความจำความบรรสาน 3. การปรับปรุงหน่วยความจำความบรรสาน 4. แทนที่คำตอบลงในหน่วยความจำความบรรสาน และ5. ตรวจสอบเงื่อนไขการหยุด ซึ่งขั้นตอนดังกล่าวถือได้ว่าเป็นวิธีการพื้นฐานสำหรับการค้นหาความบรรสาน

แต่อย่างไรก็ตาม จากลักษณะของการวางแผนประกอบวงจรพิมพ์ที่ต้องการลดเวลาการจัดเรียงส่วนประกอบ อีกทั้งยังต้องการลดเวลาสำหรับเครื่องป้อนส่วนประกอบ (Feeder Carrige) ซึ่งเป็นลักษณะของฟังก์ชั่นหลายวัตถุประสงค์ ดังนั้นการนำขั้นตอนวิธีการค้นหาความบรรสานมาประยุกต์ใช้นั้น จำเป็นต้องใช้หลักวิธีการหาค่าเหมาะที่สุดหลายวัตถุประสงค์ นั่นคือ การครอบงำ (Dominance) ของคำตอบ เพื่อทำการหาเซตของกลุ่มคำตอบที่เหมาะสม หรือ พาเรโต้ฟร้อน (Pareto Front) โดยในที่นี้ทางผู้วิจัยได้ทำการใส่หลักการดังกล่าว (1) ลงในขั้นตอนที่ 4 แทนที่คำตอบลงในหน่วยความจำบรรสาน

$$\begin{aligned}&\text{If } u_i \le v_i \text{ then}\\ &u_i \leftarrow u'_{i\,new} \in \{u_i{}' = [u'_1\, u'_2 \ldots u'_{N-1}\, u'_N]\} \text{ Else}\\ &u_i \nleftarrow u'_{i\,new} \text{ End If}\end{aligned} \tag{8}$$

จากสมการดังกล่าว จะเห็นได้ว่าถ้าค่าเวกเตอร์ความบรรสานใหม่ ($x'_{i\,new}$) มีค่าน้อยกว่า ค่าเวกเตอร์ความบรรสานเดิม (x'_i) จะถูกนำไปแทนที่ในหน่วยความจำบรรสาน โดยขั้นตอนการทำงานของ การค้นหาความบรรสานสำหรับการวางแผนการประกอบวงจรพิมพ์ จะแสดงในรูปที่ 19



รูปที่ 19 แผนภาพการทำงานของอัลกอริทึมการค้นหาความบรรสาน สำหรับระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์

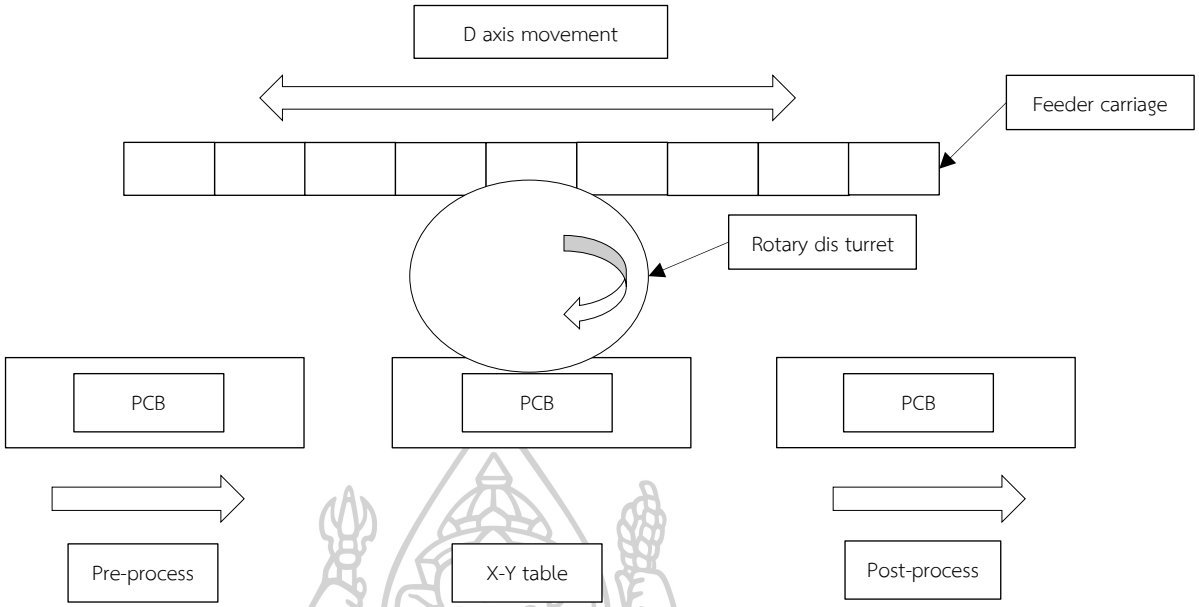
5.2.2 กระบวนการประกอบแผงวงจรอิเล็กทรอนิกส์

การประกอบแผงวงจรอิเล็กทรอนิกส์ จะเป็นกระบวนการวางชิ้นส่วนอิเล็กทรอนิกส์ลงบนแผงวงจรอิเล็กทรอนิกส์ (Printed Circuit Board: PCB) โดยทั่วไปการประกอบแผงวงจรอิเล็กทรอนิกส์จะมีอยู่ 2 ประเภท ได้แก่ PTH (Plated Through Hole: PTH) และ SMT (Surface Mount Technology: SMT) โดยการประกอบแผงวงจรอิเล็กทรอนิกส์แบบ PTH เป็นการประกอบแผงวงจรอิเล็กทรอนิกส์โดยต้องมีการเจาะรูลงไปในแผงวงจร ซึ่งอุปกรณ์ที่นำมาประกอบ จำเป็นที่ต้องมีขาเพื่อเสียบลงไปในรูของแผงวงจร แต่ในทางกลับกันเทคโนโลยี SMT จะเป็นการนำเอาชิ้นส่วนอุปกรณ์ที่ไม่ต้องเสียบขาเข้าไปในรูปของวงจร (Surface Mount Device: SMD) มาเชื่อมติดกับแผงวงจรอิเล็กทรอนิกส์ [51] สำหรับการทดลองครั้งนี้ทางผู้วิจัยได้เลือกเทคโนโลยี SMT ในการทดสอบ ซึ่งมีกระบวนการทำงาน ดังต่อไปนี้

1. เคลื่อนแผงวงจรอิเล็กทรอนิกส์ไปยังแผ่นจับประกอบ (Assembly Machine’s X-Y table)
2. นำเครื่องป้อนส่วนประกอบ (Feeder Carriage) หยิบชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์ (SMD)
3. เรียกเครื่องป้อนส่วนประกอบ (Feeder Carriage) กลับมาสู่ตำแหน่งเดิม
4. เปลี่ยนหาชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์ (SMD) ที่ต้องประกอบในตำแหน่งแรก จากเครื่องป้อนส่วนประกอบ (Feeder Carriage) และเตรียมประกอบลงบนแผงวงจรอิเล็กทรอนิกส์ (PCB)
5. ระบุตำแหน่งการวางอุปกรณ์อิเล็กทรอนิกส์ (SMD) ลงบนแผงวงจรอิเล็กทรอนิกส์ (PCB)
6. การวางอุปกรณ์อิเล็กทรอนิกส์ (SMD) ลงบนแผงวงจรอิเล็กทรอนิกส์ (PCB)
7. ทำซ้ำขั้นตอนที่ 2 ถึง 6 จนกระทั่งงานประกอบเสร็จสิ้น
8. การถอดแผงวงจรอิเล็กทรอนิกส์ (PCB) ออกจากแผ่นจับประกอบประกอบ

สำหรับการประกอบแผงวงจรอิเล็กทรอนิกส์ประเภทSMT จะมีเครื่องประกอบอยู่หลากหลายชนิด ไม่ว่าจะเป็น Mass placement machine, The coordinate-driven, Sequential pick-and-place machine, The rotary disk turret หรือ Concurrent pick and place machine ซึ่งเครื่องประกอบแผงวงจรอิเล็คทรอนิคส์ที่แตกต่างกัน จะมีข้อดีข้อเสียที่แตกต่างกัน [51]

แต่อย่างไรก็ตาม สำหรับการทดลองในครั้งนี้ทางผู้วิจัยได้เลือกเครื่อง Fuji FCP-IV ซึ่งจัดอยู่ในประเภท The rotary disk turret และ Concurrent pick and place machine โดยเครื่องดังกล่าวถือได้ว่าเป็นเครื่องที่ได้รับความนิยมมากที่สุดสำหรับผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ โดยลักษณะของเครื่องดังกล่าว แสดงในรูปที่ 20



รูปที่ 20 แผนผังตำแหน่งเครื่อง Fuji FCP-IV [51]

5.2.3 รูปแบบของฟังก์ชั่นหลายวัตถุประสงค์

จากรูปที่ จะเห็นได้ว่าเครื่อง Fuji FCP-IV มีการหลักการทำงาน 3 องค์ประกอบ ได้แก่ 1. การเคลื่อนที่ของเครื่องป้อนส่วนประกอบชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์ (Moving feeder carriage for SMDs) 2. การเคลื่อนที่ของแผ่นจับประกอบสำหรับการหาตำแหน่งแผงวงจรอิเล็กทรอนิกส์ (Moving X-Y table for positioning the PCB) และ 3. การหมุนของหัวประกอบชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์ (Rotary disk turret) จะเห็นได้ว่า ลำดับการวางชิ้นส่วนลงบนแผงวงจรอิเล็กทรอนิกส์ การจัดลำดับของอุปกรณ์อิเล็กทรอนิกส์ และความเร็วของหัวประกอบชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์ เป็นพารามิเตอร์ที่สำคัญ อีกทั้งการวางแผนการประกอบแผงวงจร ประกอบด้วยกฎพื้นฐาน 2 ข้อ ได้แก่

กฎข้อที่ 1: จัดตำแหน่งในการวางชิ้นส่วนอุปกรณ์อิเล็คทรอนิคส์ เพื่อให้ใช้เวลาให้น้อยที่สุด
กฎข้อที่ 2: จัดตำแหน่งเครื่องป้อนส่วนประกอบชิ้นส่วนอุปกรณ์อิเล็คทรอนิคส์ เพื่อลดเวลาในการหยิบ (Pick-up time)

จากกฎข้องบังคับดังกล่าว จะเห็นได้ว่า กฎข้อที่ 1 และกฎข้อที่ 2 จะเป็นตัวกำหนดฟังก์ชั่นหลายวัตถุประสงค์ (Multi-Objective function) ตามลำดับ

เครื่อง Fuji FCP-IV มีการเคลื่อนไหวอย่างอิสระ ทั้งแนวแกน X และแนวแกน Y ดังนั้นเวลา (C_t) ที่ได้จะขึ้นอยู่กับการเคลื่อนที่ทั้ง 2 แนวแกน แสดงดังสมการ (8) และ (9)

$$\begin{aligned} C_{x-time} &= Time\ for\ the\ X-Y\ table\ to\ travel\ in\ the\ X\ direction \\ &= \frac{|X_{n+1}-X_n|}{Speed\ of\ the\ X-Y\ table\ in\ the\ X-direction} \end{aligned} \tag{8}$$

$$\begin{aligned} C_{y-time} &= Time\ for\ the\ X-Y\ table\ to\ travel\ in\ the\ X\ direction \\ &= \frac{|Y_{n+1}-Y_n|}{Speed\ of\ the\ X-Y\ table\ in\ the\ Y-direction} \end{aligned} \tag{9}$$

จากสมการข้างต้น จะเห็นได้ว่าเวลาการเคลื่อนที่ (C_t) จะถูกเลือกจากค่ามากที่สุดของในแนวแกน X และ แนวแกน Y ดังแสดงในสมการที่ (10)

$$C_t = max(\ C_{x-time}, C_{y-time}) \tag{10}$$

ในทำนองเดียวกัน เวลาการเคลื่อนย้าย (F_t) ของ Cartridge จากตำแหน่ง n ไปสู่ ตำแหน่ง m สามารถคำนวนได้ดังสมการที่ (11)

$$F_t = |n-m| \times time\ travelled\ per\ cartridge \tag{11}$$

ดังนั้นฟังก์ชั่นของเวลาจะขึ้นอยู่กับค่าที่มากที่สุดของ เวลาการเคลื่อนที่ (C_t) และ เวลาการเคลื่อนย้าย (F_t) ดังแสดงในสมการที่ (12)

$$Cycle\ time = max(C_t, F_t) \tag{12}$$

5.2.4 การสร้างรูปแบบระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ เข้ากับปัญหาการเดินทางของพนักงานขาย

จากหัวข้อที่ 5.2.3 จะเห็นได้ว่า ระยะเวลาการประกอบแผงวงจร จะขึ้นอยู่กับปัจจัย 2 ประการ ได้แก่ ลำดับตำแหน่งชิ้นส่วน และการจัดเรียงชิ้นส่วนอุปกรณ์อิเล็คทรอนิกส์ (SMD) บนเครื่องป้อน (Feeder) โดยหนึ่งในวิธีการเพิ่มประสิทธิภาพลำดับการจัดเรียงส่วนประกอบ คือการแปลงให้เข้ากับปัญหาการเดินทางของพนักงานขาย (Pseudo-TSP) โดยกำหนดให้ (a) แทนลำดับการวางส่วนประกอบลงไปในแผงวงจรอิเล็คทรอนิคส์ที่อยู่บนแผ่นจับประกอบ และ (b) แทนลำดับการจัดเรียงบนเครื่องป้อนส่วนประกอบ ดังแสดงในรูปที่ 21

(a) Component Insertion Sequence (X-Y table)

3	4	2	1	8	7

(b) Feeder Arrangement Sequence

1	3	2

Component Reference Number	X-coordinate	Y-coordinate	Component Type
1	3.00	12.95	1
2	3.00	12.05	1
3	3.00	11.25	1
4	3.00	10.35	1
5	3.00	9.50	2
6	3.00	8.70	2
7	3.00	7.90	2
8	9.65	12.55	3
9	17.20	9.85	4
10	17.20	9.60	4
...	...	...	...
...	...	...	...
71	5.55	0.10	20

รูปที่ 21 การจัดวางส่วนประกอบอิเล็กทรอนิกส์ [51]

จากลักษณะดังกล่าวข้างต้น เพื่อให้เป็นการเข้าใจมากขึ้น ทางผู้วิจัยทำการอธิบายถึงวิธีการเข้ารหัส (Encoding) และถอดรหัส (Decoding) สำหรับการการทดลองในครั้งนี้ โดยแสดงดังรูปที่ 22

(a)

1			2				3		4	5		
1.1	1.2	1.3	2.1	2.2	2.3	2.4	3.1	3.2	4.1	5.1	5.2	5.3
0.354	0.974	0.128	0.148	0.276	0.639	0.784	0.248	0.534	0.671	0.846	0.371	0.932

(b)

1	2	3	2	1	5	3	2	4	2	5	5	1
1.3	2.1	3.1	2.2	1.1	5.2	3.2	2.3	4.1	2.4	5.1	5.3	1.2
0.128	0.148	0.248	0.276	0.354	0.371	0.534	0.639	0.671	0.784	0.846	0.932	0.974

รูปที่ 22 การเข้ารหัส (Encoding) และถอดรหัส (Decoding)

จากรูปที่ 22 จะเห็นได้ว่า กำหนดให้แถวด้านบนสุดของเวกเตอร์ (a) แทนชนิดของชิ้นส่วนอิเล็คทรอนิกส์ที่จะทำการประกอบลงบนแผนวงจร (PCB) ตั้งแต่ 1 ถึง 5 ซึ่งแต่ละชนิดของชิ้นส่วนอิเล็คทรอนิกส์จะมีจำนวนชิ้นส่วนย่อยที่แตกต่างกันดังแสดงในแถวกลาง และส่วนประกอบทุกชิ้นจะทำการสุ่มลำดับเลข ตั้งแต่ 0 ถึง 1 ดังแสดงในแถวล่างสุด หลังจากสุ่มค่าได้นั้นลำดับเลขดังกล่าวจะถูกนำมาเรียงตามค่าน้อยไปสู่ค่ามากดังแสดงเวกเตอร์ (b) ซึ่งค่าดังกล่าวจะถูกนำไปเป็นค่าเริ่มต้นในการค้นหาความบรรสาน

ดังนั้นด้วยวิธีการเข้ารหัสและถอดรหัส ดังกล่าวข้างต้น ค่าเริ่มต้นสำหรับชนิดชิ้นส่วนอิเล็คทรอนิกส์ในเครื่องป้อนส่วนประกอบ คือ 1>2>3>2>1>5>3>2>4>2>5>5>1 และค่าเริ่มต้นในส่วนของตำแหน่งการวางชิ้นส่วนอิเล็คทรอนิกส์ลงบนแผงวงจร คือ 1.3>2.1>3.1>2.2>1.1>5.2>3.2>2.3>4.1>2.4>5.1>5.3>1.2

5.3 การหาค่าพารามิเตอร์ที่เหมาะสม

สำหรับการออกแบบการทดลอง และการตั้งค่าพารามิเตอร์สำหรับระบบการวางแผนประกอบแผงวงจรอิเล็กทรอนิกส์ ทางผู้วิจัยจะทำการตั้งค่าพารามิเตอร์ทั้งหมด 4 ชนิด ได้แก่ พารามิเตอร์ความน่าจะเป็นในการใช้หน่วยความจำ (Harmony Memory Considering Rate: HMCR), พารามิเตอร์ความน่าจะเป็นในการปรับระดับเสียง (Pitch Adjustment Rate: PAR), พารามิเตอร์ขนาดของหน่วยความจำความบรรสาน (Harmony Memory Size: HMS) และพารามิเตอร์จำนวนการทำซ้ำ (Iteration number: T) อีกทั้งกำหนด ระดับการปรับตั้งพารามิเตอร์ 3 ระดับ ดังแสดงในตารางที่ 8

ตารางที่ 8 การปรับตั้งพารามิเตอร์

Parameter	Number of levels	Setting
$HMCR$	3	0.3, 0.6, 0.9
PAR	3	0.3, 0.6, 0.9
HMS	3	30, 60, 90
$Iteration\ number$	3	1000, 3000, 5000

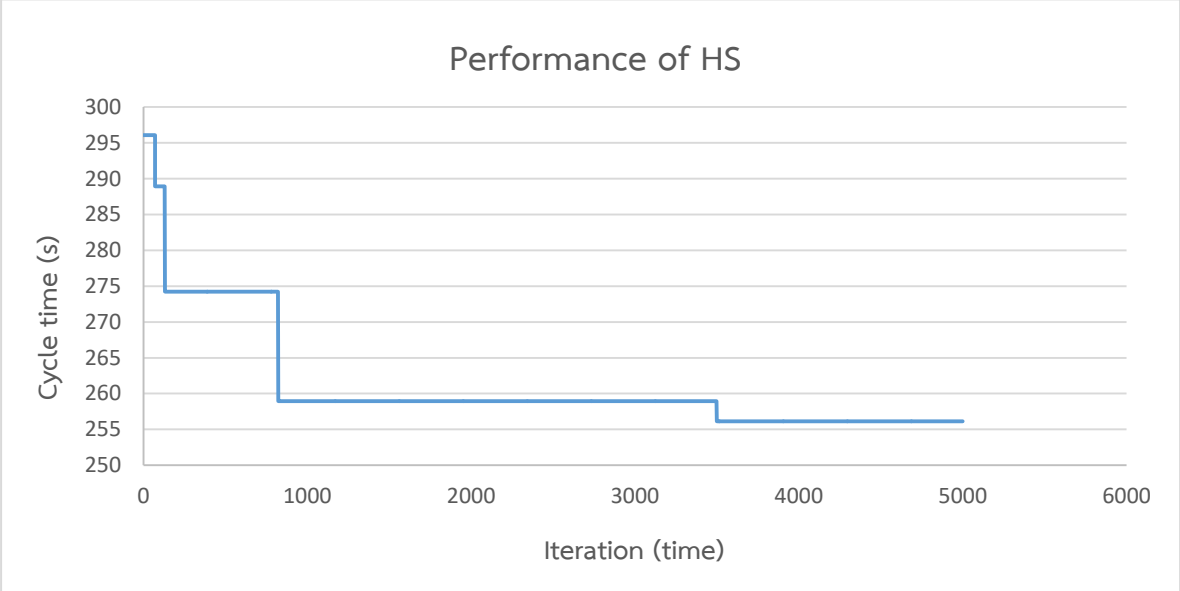
สำหรับการทดลองครั้งนี้ ทางผู้วิจัยได้เลือกใช้ชุดข้อมูลจาก (Sanchez and Priest) โดยประกอบไปด้วย 71 ส่วนประกอบสำหรับแผงวงจร และทำการทดลองด้วยโปรแกรม Python 3.5.2 64 bit บนเครื่อง Intel® Core™ i7-4720HQ CPU 2.60 GHz processor, with 4.00 GB of RAM ซึ่งมีชุดการทดลองทั้งหมด 81 $(3 \times 3 \times 3 \times 3)$ ชุดการทดลอง โดยแต่ละชุดการทดลองมีการทำซ้ำชุดละ 20 ครั้ง ดังนั้นในการทดลองการค้นหาความบรรสานสำหรับระบบการวางแผนประกอบแผงวงจรอิเล็กทรอนิกส์จะมีการทดลองทั้งหมด 1,620 ครั้ง

5.4 ผลการทดลองและการวิเคราะห์ผลการทดลอง

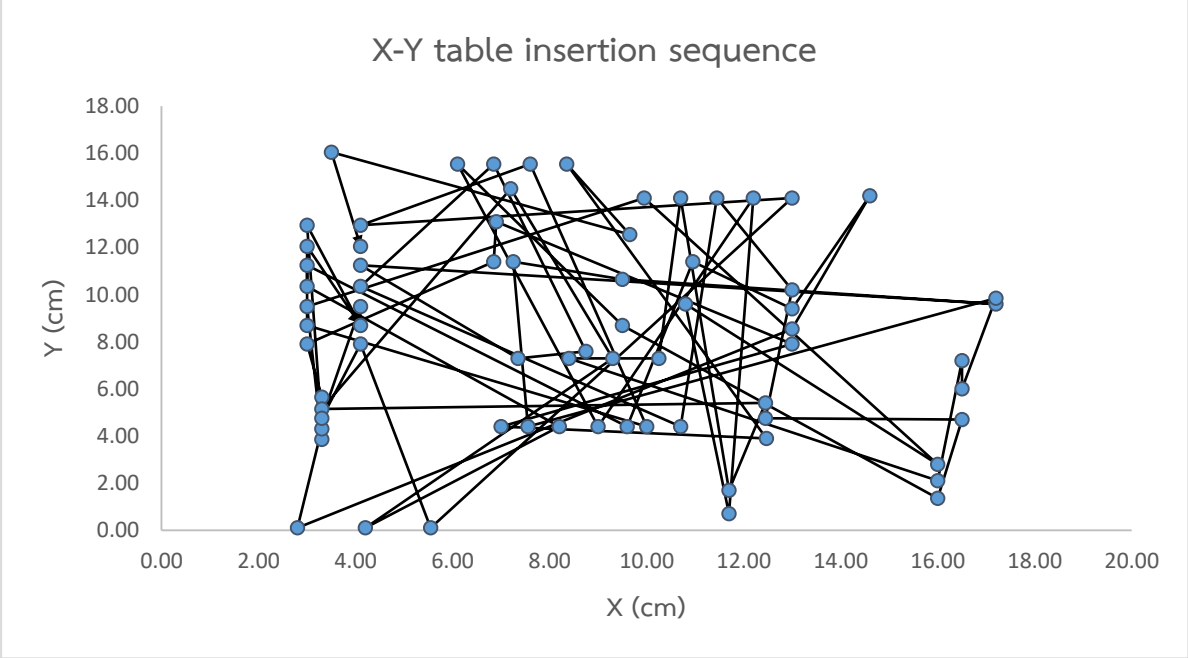
ในส่วนของผลการทดลองและการวิเคราะห์ผลการทดลอง ทางผู้วิจัยได้แบ่งการอธิบายเป็น 3 หัวข้อหลัก ได้แก่ การทดสอบประสิทธิภาพการหาคำตอบของการค้นหาความบรรสาน การแสดงผลของพารามิเตอร์สำหรับการทดลอง และปัญหาหลายวัตถุประสงค์สำหรับการทดลอง

5.4.1 การทดสอบประสิทธิภาพการหาคำตอบของการค้นหาความบรรสาน

ในส่วนแรกการค้นหาความบรรสานจะถูกนำมาทดสอบเข้ากับระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ ด้วยชุดข้อมูล Bill of materials ที่มีจำนวนชิ้นส่วน 71 ชิ้นส่วน [54] โดยมีวัตถุประสงค์เพื่อ แสดงถึงประสิทธิภาพการทำงานของการค้นหาความบรรสาน และการเคลื่อนที่ในการประกอบชิ้นส่วนอิเล็คทรอนิกส์ ดังแสดงในรูปที่ 23 และ 24 ตามลำดับ



รูปที่ 23 ประสิทธิภาพการทำงานของการค้นหาความบรรสาน สำหรับการประกอบแผงวงจรอิเล็คทรอนิกส์



รูปที่ 24การเคลื่อนที่ในการประกอบชิ้นส่วนอิเล็คทรอนิกส์

จากรูปที่ 23 จะเห็นได้ว่าในช่วงแรกของการทดสอบ (Iteration < 1000) จะเกิดการลู่เข้าหาคำตอบขึ้นอย่างต่อเนื่องเมื่อเปรียบเทียบกับ ช่วงหลังของการทดสอบ (Iteration > 1000) ซึ่งอาจจะสรุปได้ว่า จำนวนการทำซ้ำมากขึ้น ส่งผลให้คำตอบมีค่าที่ดีขึ้น แต่อย่างไรก็ตามผลการทดสอบ

ดังกล่าวเป็นเพียงการทดสอบอย่างง่าย จึงไม่เพียงพอต่อการยืนยันถึงข้อสันนิฐานดังกล่าว ดังนั้นผู้วิจัยจึงดำเนินการทดสอบด้วยวิธีการทางสถิติต่อไป

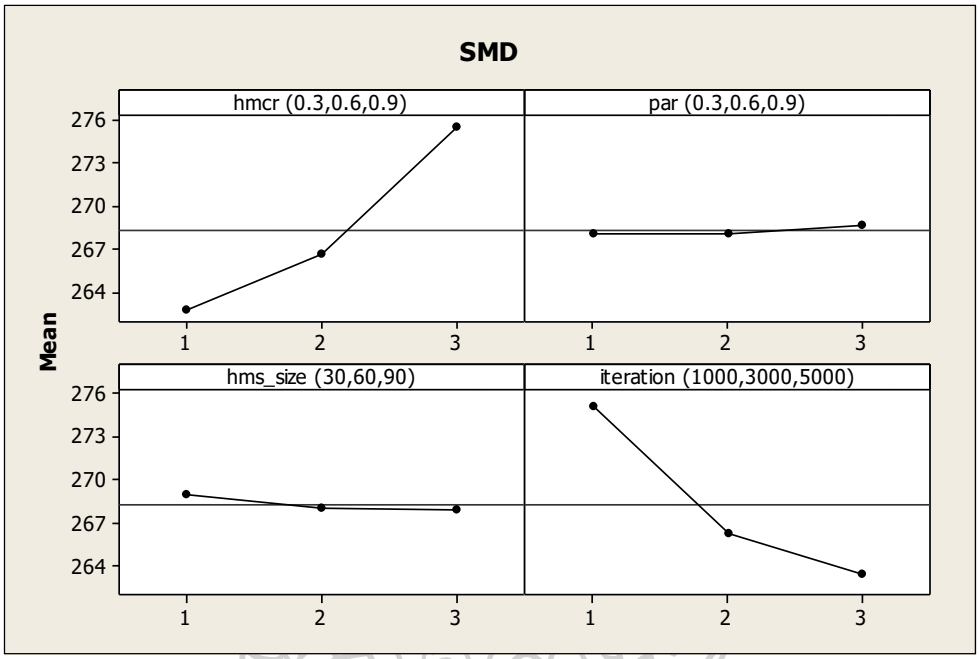
5.4.2 ผลของพารามิเตอร์สำหรับการทดลอง

ต่อมาทางผู้วิจัยได้ใช้วิธีการทางสถิติ เพื่อหาค่าของพารามิเตอร์ที่ส่งผลอย่างมีนัยสำคัญ (significant parameter) โดยผลดังกล่าวจะแสดงใน ตารางที่ 9

ตารางที่ 9 ค่า P-Value จากการทดสอบด้วยการค้นหาความบรรสาน สำหรับชุดข้อมูล BOM

Source	p-Value
$HMCR$	0.000
PAR	0.530
HMS	0.190
$iteration$	0.000
$HMCR \times PAR$	0.237
$HMCR \times HMS$	0.659
$HMCR \times iteration$	0.605
$PAR \times HMS$	0.086
$PAR \times iteration$	0.289
$HMS \times iteration$	0.021
$HMCR \times PAR \times HMS$	0.091
$HMCR \times PAR \times iteration$	0.893
$HMCR \times HMS \times iteration$	0.661
$PAR \times HMS \times iteration$	0.934
$HMCR \times PAR \times HMS \times iteration$	0.022

จากตารางที่ 9 ระดับนัยสำคัญ เท่ากับ 0.05 ($\alpha = 0.05$) พบว่าความน่าจะเป็นในการใช้หน่วยความจำ (Harmony Memory Considering Rate: HMCR) และจำนวนการทำซ้ำ (Iteration number) มีผลต่อคำตอบอย่างมีนัยสำคัญ ด้วยเหตุนี้เองทางผู้วิจัยจึงมุ่งเน้นไปในการปรับตั้งค่าที่เหมาะสมสำหรับพารามิเตอร์ดังกล่าว โดยทำการพิจารณาจากปัจจัยหลัก (Main effect plot) ผลปรากฏว่า ค่าที่เหมาะสมสำหรับพารามิเตอร์ความน่าจะเป็นในการใช้หน่วยความจำ มีค่าเท่ากับ 0.3 และ 5,000 รอบ สำหรับพารามิเตอร์จำนวนการทำซ้ำ ดังแสดงในรูปที่ 25



รูปที่ 25 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดข้อมูล BOM

ขั้นตอนต่อมาทางผู้วิจัยมีความสนใจ ถึงสมมุติฐาน ถ้าจำนวนการทำซ้ำ (Iteration number) เพิ่มมากขึ้น ค่าที่ได้จากการทดสอบจะมีค่าที่ดีขึ้น ซึ่งในที่นี้ทางผู้วิจัยจะทำการทดสอบสมมุติฐานนี้ด้วยการวิเคราะห์ความแปรปรวน. (Analysis of Variance, ANOVA) โดยพารามิเตอร์สำหรับการวิเคราะห์ ทางผู้วิจัยได้ทำการกำหนดค่าต่าง ๆ ดังนี้ $HMCR = 0.3$, $PAR = 0.3$, $HMS = 60$ อีกทั้งกำหนด จำนวนการทำซ้ำที่แตกต่างกัน คือ 2500, 5000, 7500 และ 10000 ตามลำดับ ซึ่งแต่ละการทดลองจะมีการทำซ้ำ จำนวน 20 ครั้ง ดังแสดงในตารางที่ 10

ตารางที่ 10 แสดงผลการวิเคราะห์ความแปรปรวน (ANOVA) สำหรับการวางแผนประกอบแผงวงจรอิเล็คทรอนิกส์

Source of Variance	Degree of Freedom	Sum of Squares	Mean Square	F_0	p-value
iterations	3	816.8	272.3	4.11	0.009
Error	76	5031.5	66.2		
Total	79	5848.3			

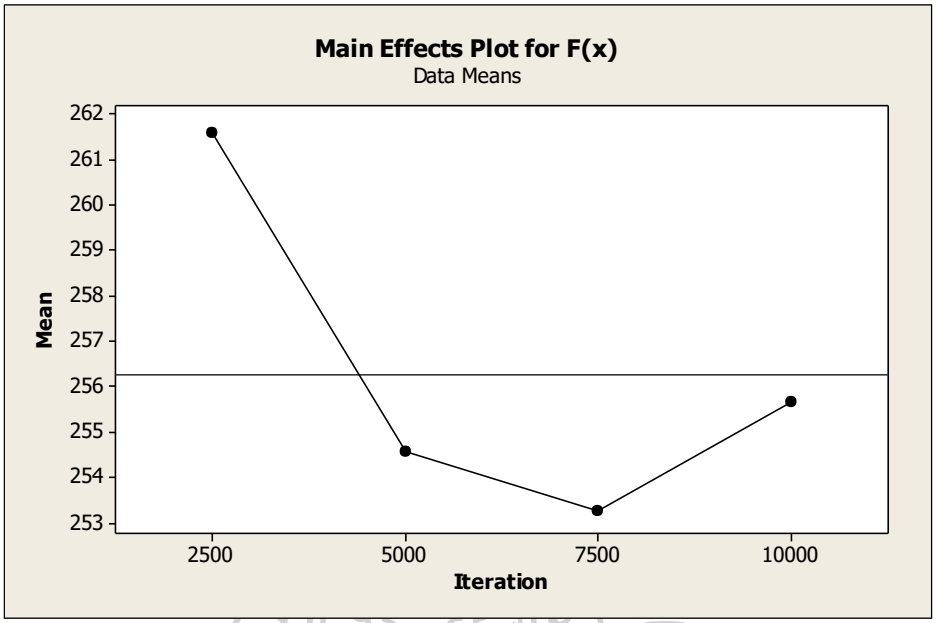
จากตารางที่ 10 จะเห็นได้ว่าจำนวนการทำซ้ำมีผลต่อคำตอบที่ได้ อย่างมีนัยสำคัญ แต่อย่างไรก็ตาม ทางผู้วิจัยได้ทำการทดสอบต่อไป ด้วยวิธีการตูกี (Tukey's test) เพื่อแสดงให้เห็นถึงความแตกต่างระหว่างจำนวนการทำซ้ำที่ต่างกัน (2500, 5000, 7500, และ 10000) นั้นจะมีผลต่อกันอย่างมีนัยสำคัญหรือไม่

จากผลการทดสอบในรูปที่ 26 ทางผู้วิจัยพบว่าสำหรับจำนวนการทำซ้ำ 5000, 7500 และ 10000 ถูกจัดให้อยู่ในกลุ่มเดียวกัน ขณะที่จำนวนการทำซ้ำ 2500 ถูกจัดอยู่ในกลุ่มที่แตกต่างกัน อีกทั้งจำนวนการทำซ้ำที่มากขึ้นเกินกว่า 7500 นั้นไม่ส่งผลให้ได้คำตอบที่ดีขึ้น ดังนั้นการค้นหาความบรรสาน สำหรับระบบการวางแผนการประกอบแผงวงจรอิเล็กทรอนิกส์ในครั้งนี้ ค่าพารามิเตอร์จำนวนการทำซ้ำ (Iteration number) เท่ากับ 7500 ถือว่ามีค่าเหมาะสมที่สุด

```
Individual 95% CIs For Mean Based on
                        Pooled StDev
Level   N    Mean  StDev  ------+---------+---------+---------+---
 2500  20  261.61   7.90                      (--------*--------)
 5000  20  254.57   7.78     (--------*--------)
 7500  20  253.26   9.78  (---------*--------)
10000  20  255.67   6.80     (--------*--------)
                        ------+---------+---------+---------+---
                            252.0     256.0     260.0     264.0
Pooled StDev = 8.14

Grouping Information Using Tukey Method

Iteration   N     Mean  Grouping
 2500      20  261.611  A
10000      20  255.669  A B
 5000      20  254.570    B
 7500      20  253.260    B

Means that do not share a letter are significantly different.
```

รูปที่ 26 ผลการทดสอบด้วยวิธีตูกี กับจำนวนการทำซ้ำที่แตกต่างกัน

และเพื่อยืนยันถึงค่าเหมาะที่สุดของพารามิเตอร์จำนวนการซ้ำ คือ 7500 ทางผู้วิจัยได้แสดงผลของ Main Effect Plot จะเห็นได้ว่าค่าจำนวนการซ้ำดังกล่าว จะให้ค่าที่ต่ำที่สุด ดังรูปที่ 27

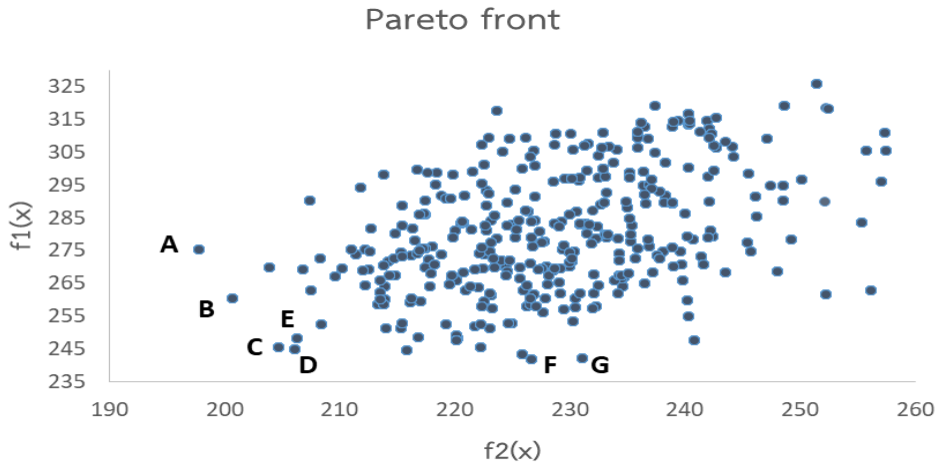


รูปที่ 27 ผลค่าเฉลี่ยของพารามิเตอร์หลัก สำหรับชุดข้อมูล BOM

5.4.3 ปัญหาหลายวัตถุประสงค์สำหรับการทดลอง

สำหรับการทดสอบครั้งนี้ทางผู้วิจัยเลือกใช้ชุดข้อมูล Bill of materials (BOM) มาทดสอบการประกอบแผงวงจรอิเล็กทรอนิกส์ ด้วยเครื่อง Fuji FCP-IV โดยมีตำแหน่งในการประกอบจำนวน 71 ตำแหน่ง และชิ้นส่วนสำหรับการประกอบ 71 ชิ้นส่วน ซึ่งประกอบไปด้วย สารกึ่งตัวนำ (Semi-conductor) ตัวต้านทาน (Resistor) ตัวต้านทานปรับค่าได้ (VAR Resistor) ไดโอด (Diode) ตัวเก็บประจุ (Capacitor) และ ทรานซิสเตอร์ (Transistor) ตามลำดับ

ตัวปัญหาสำหรับการทดสอบในครั้งนี้เป็นฟังก์ชั่น 2 วัตถุประสงค์ ซึ่งมีความซับซ้อนในการเลือกค่าเหมาะที่สุดของคำตอบ แต่อย่างไรก็ตามด้วยบทนิยามสำหรับการหาค่าเหมาะที่สุดหลายวัตถุประสงค์ ได้แก่ การครอบงำแบบพาเรโต้ (Pareto Dominance) พาเรโตฟร้อน (Pareto front) และ Weak-Strict Pareto Optimality จะเข้ามามีส่วนช่วยสำหรับการเลือกค่าของคำตอบ แสดงในรูปที่ 28



รูปที่ 28 คำตอบของปัญหาการประกอบแผงวงจรอิเล็คทรอนิกส์

จากรูปดังกล่าว จะเห็นได้ว่า เมื่อพิจารณาที่ตำแหน่ง A และตำแหน่ง B พบว่ามีลักษณะการครอบงำแบบพาเรโต้ คือ ค่าของคำตอบตำแหน่งดังกล่าวจะมีค่าน้อยกว่าคำตอบในตำแหน่งอื่น ๆ ต่อมาเมื่อพิจารณาตำแหน่ง C และตำแหน่ง E จะเห็นได้ว่าค่าของตำแหน่ง C นั้นครอบงำตำแหน่ง E ทั้งฟังก์ชั่นวัตถุประสงค์ที่ 1 และฟังก์ชั่นวัตถุประสงค์ที่ 2 หรือกล่าวได้ว่าเป็น strictly dominate แต่ในทางกลับกันเมื่อพิจารณาตำแหน่ง E กับตำแหน่ง D พบว่ามีการครอบงำสำหรับฟังก์ชั่นวัตถุประสงค์ที่ 1 เท่านั้น ซึ่งเป็นการครอบงำแบบ weakly dominate จากหลักการดังกล่าวกลุ่มคำตอบที่เหมาะสมของปัญหาการประกอบแผงวงจรอิเล็คทรอนิกส์ คือ ตำแหน่ง A-B-C-D-F-G ซึ่งเรียงตัวกันเกิดเป็น พาเรโตฟร้อน (Pareto front) อีกทั้งทางผู้วิจัยได้พิจารณาต่อถึงประสิทธิภาพการค้นหาความบรรสานในการได้มาซึ่งคำตอบที่อยู่บนพาเรโต้ฟร้อน โดยทำการทดสอบทั้งหมด 20,000 ครั้ง ซึ่งแต่ละครั้งจะให้ออกมา 1 คำตอบ จากการทดสอบพบว่า มีคำตอบอยู่บนพาเรโต้ฟร้อนทั้งหมด 5 คำตอบ หรือคิดเป็น 0.025 เปอร์เซ็น

5.4 สรุปผลการทดลอง

การออกแบบการค้นหาความบรรสานสำหรับการวางแผนการประกอบชิ้นส่วนอุปกรณ์อิเล็คทรอนิกส์ ดำเนินการจำลองเพื่อทดสอบกับเครื่อง Fuji FCP-IV ซึ่งมีลักษณะ Rotary disk turret และ Concurrent pick and place machine อีกทั้งยังมีรูปแบบการทำงานเชิงเส้น โดยกระบวนการดังกล่าวจะมีลักษณะหลายวัตถุประสงค์ ต่อมาชุดข้อมูล Bill of materials ได้ถูกนำมาทดสอบกับการออกแบบดังกล่าว โดยแสดงให้เห็นถึงความสามารถในการสร้างลำดับการจัดวางองค์ประกอบ และการจัดเรียงส่วนประกอบ

โดยส่วนแรกจะเป็นการทดสอบอย่างง่าย จะเห็นได้ว่าเมื่อเพิ่มจำนวนการทำซ้ำมากขึ้นจะส่งผลต่อการลู่เข้าหาคำตอบ อีกทั้งยังพบว่าค่าของคำตอบมีค่าที่ดีขึ้น แต่อย่างไรก็ตามผลการทดลองดังกล่าวเป็นเพียงผลการทดลองอย่างง่าย ดังนั้นลำดับต่อมาทางผู้วิจัยได้ทำการออกแบบการทดลองด้วยวิธีทางสถิติเพื่อหาค่าพารามิเตอร์ที่เหมาะสม พบว่าค่าพารามิเตอร์ความน่าจะเป็นในการใช้หน่วยความจำ (HMCR) และ พารามิเตอร์จำนวนการทำซ้ำ (T) นั่นส่งผลต่อคำตอบที่ได้รับอย่างมีนัยยะสำคัญ โดยมีค่าเท่ากับ 0.3 และ 5000 ตามลำดับ นอกจากนี้ทางผู้วิจัยได้ทำการทดสอบถึงสมมุติฐานจำนวนการทำซ้ำที่มากขึ้น ไม่ส่งผลให้คำตอบมีค่าที่ดีขึ้นด้วยวิธีการตูกี (Turkey test) ซึ่งพบว่าเมื่อจำนวนการทำซ้ำมีค่ามากเกินกว่า 7500 จะไม่ส่งผลสำหรับการพัฒนาของคำตอบ ส่วนสุดท้ายจะเป็นการแสดงถึงลักษณะและประสิทธิภาพสำหรับคำตอบปัญหาหลายวัตถุประสงค์ ซึ่งจะถูกสร้างออกมาในลักษณะพาเรโต้ฟร้อน โดยพบว่าสำหรับการค้นหาความบรรสานในครั้งนี้มีความสามารถในการสร้างคำตอบอยู่บนพาเรโต้ฟร้อน เท่ากับ 0.025 เปอร์เซ็น

บทที่ 6
สรุปผลการวิจัย

6.1 สรุปผลการวิจัย

ขั้นตอนวิธีเมตาฮิวริสติก เป็นการอาศัยพฤติกรรมของธรรมชาติเพื่อค้นหาคำตอบ ซึ่งความสามารถที่โดดเด่นอย่างหนึ่งของขั้นตอนวิธีเมตาฮิวริสติกคือการหาคำตอบเหมาะที่สุดของปัญหาที่มีความซับซ้อนได้ในเวลาที่เหมาะสม ด้วยเหตุนี้เองทำให้ขั้นตอนวิธีเมตาฮิวริสติกเกิดการพัฒนาขึ้นมาอย่างต่อเนื่อง โดยการค้นหาความบรรสานก็ถือเป็นหนึ่งในขั้นตอนวิธีดังกล่าวที่ถูกคิดค้นขึ้นในช่วงทศวรรตที่ผ่านมา แต่อย่างไรก็ตามการเกิดขึ้นมาไม่นานของขั้นตอนวิธีการค้นหาความบรรสาน ทำให้การนำไปใช้แก้ปัญหายังไม่หลากหลายและครอบคลุมเท่าที่ควร อีกทั้งเงื่อนไขที่เหมาะสม สำหรับแต่ละปัญหายังไม่มีการวิจัยอย่างแพร่หลาย จากเหตุผลดังกล่าวทางผู้วิจัยจึงทำการทดสอบกับการค้นหาความสานกับปัญหาพนักงานขายซึ่งเป็นปัญหาลักษณะ NP-complete โดยทำการทดสอบในชุดปัญหาที่แตกต่างกัน 3 ชุดปัญหาได้แก่ UK12 WG59 และ SGB128 อีกทั้งสร้างวิธีการแปลงรหัสสำหรับปัญหาการเดินทางของพนักงานขายขึ้นมาโดยเฉพาะ ซึ่งการแปลงรหัสดังกล่าว จะถูกนำมาเป็นค่าเริ่มต้นสำหรับขั้นตอนวิธีการค้นหาความบรรสาน อีกทั้งทางผู้วิจัยยังนำวิธีการออกแบบการทดลอง ซึ่งเป็นวิธีการทางสถิติที่ใช้อย่างแพร่หลาย เพื่อหาพารามิเตอร์ที่เหมาะสม ลำดับต่อมาทางผู้วิจัยได้นำการค้นหาความบรรสานมาทดสอบกับปัญหาหลายวัตถุประสงค์ โดยในที่นี้ได้เลือกการวางแผนประกอบแผงวงจรอิเล็คทรอนิกส์ซึ่งเป็นหนึ่งในกระบวนการที่สำคัญสำหรับอุตสาหกรรมอิเล็คทรอนิกส์ โดยดำเนินการทดสอบกับชุดข้อมูลชิ้นส่วนอิเล็คทรอนิกส์ 71 ชิ้นส่วน อีกทั้งยังทำการประยุกต์เอาลักษณะของปัญหาการเดินทางของพนักงานขายมาใช้กับกับการวางแผนดังกล่าว รวมถึงแสดงการแปลงรหัส และหาพารามิเตอร์ที่เหมาะสม และสุดท้ายเมื่อได้ถึงเงื่อนไข ที่เหมาะสมทางผู้วิจัยจะแสดงถึงขีดความสามารถในการสร้างคำตอบของปัญหาหลายวัตถุประสงค์ในรูปแบบพาเรโต้ฟร้อนอีกด้วย ซึ่งสรุปผลได้ดังต่อไปนี้

6.1.1 จากปัญหาการเดินทางของพนักงานขาย ทางผู้วิจัยเริ่มต้นจากการแปลงรหัสด้วยวิธีการสุ่มค่าออกเป็นเลขทศนิยม ตั้งแต่ 0-1 ซึ่งค่าดังกล่าวจะถูกนำไปแทนที่ในเมืองแต่ละเมือง จากนั้นเรียงค่าจากน้อยไปมาก จากนั้นค่าที่ได้จะถูกนำมาเป็นค่าเริ่มต้นสำหรับการค้นหาความบรรสาน ต่อมาทำการออกแบบการทดลองเพื่อหาพารามิเตอร์ที่เหมาะที่สุด พบว่าความน่าจะเป็นในการปรับระดับเสียง (PAR) และขนาดของหน่วยความจำความบรรสาน (HMS) นั้นไม่ส่งผลต่อคำตอบ

จะมีเพียงความน่าจะเป็นในการใช้หน่วยความจำ (HMCR) และจำนวนการทำซ้ำ (T) เท่านั้นที่ส่งผลต่อคำตอบอย่างมีนัยสำคัญ ต่อมาทำการทดสอบสมมุติฐานการเพิ่มขึ้นของจำนวนการทำซ้ำ ส่งผลต่อคำตอบที่ได้ ด้วยวิธีการตูกี พบว่าค่าจำนวนการทำซ้ำที่เพิ่มขึ้นเกินกว่า 5000 จะไม่ส่งผลให้ได้คำตอบที่ดีขึ้น แต่กลับทำให้เวลาในการหาคำตอบเพิ่มขึ้น ดังนั้นค่าพารามิเตอร์ที่เหมาะสมของการค้นหาความบรรสาน สำหรับปัญหาการเดินทางของพนักงานขาย คือ $HMCR = 0.3$ และ $T = 5000$ ตามลำดับ

6.1.2 สำหรับการวางแผนการประกอบแผงวงจรอิเล็คทรอนิกส์ ทางผู้วิจัยได้นำลักษณะปัญหาการเดินทางของพนักงานขายเข้ามาประยุกต์ใช้ โดยเปรียบเทียบชิ้นส่วนอิเล็กทรอนิกส์แต่ละชนิดแทนเมืองแต่ละเมืองที่พนักงานขายจะต้องเดินทาง จากนั้นทำการแปลงรหัสโดยวิธีการสุ่มค่า 0-1 เช่นเดียวกับการทดสอบปัญหาการเดินทางของพนักงานขาย แต่เนื่องจากปัญหาการวางแผนการประกอบแผงวงจรอิเล็คทรอนิกส์เป็นปัญหาหลายวัตถุประสงค์ ค่าที่ได้จากการแปลงรหัสจะถูกนำไปใช้เป็นค่าเริ่มต้น 2 ส่วนได้แก่ ส่วนเครื่องป้อนอุปกรณ์ และลำดับการประกอบชิ้นส่วนอิเล็คทรอนิกส์ลงบนแผงวงจร จากนั้นทำการทดสอบอย่างง่าย พบว่าช่วงแรกของการทดสอบ (Iteration < 1000) จะเกิดการลู่เข้าหาคำตอบขึ้นอย่างต่อเนื่องเมื่อเปรียบเทียบกับ ช่วงหลังของการทดสอบ (Iteration > 1000) ต่อมาทางผู้วิจัยได้ทำการออกแบบการทดลองเพื่อหาพารามิเตอร์ที่เหมาะสม พบว่ามีเพียงความน่าจะเป็นในการใช้หน่วยความจำ (HMCR) และจำนวนการทำซ้ำ (T) เท่านั้นที่ส่งผลต่อคำตอบอย่างมีนัยสำคัญ ซึ่งได้ผลเหมือนกับการทดสอบการค้นหาความบรรสาน สำหรับปัญหาการเดินทางของพนักงานขาย แต่เมื่อพิจารณาค่าจำนวนการทำซ้ำที่มากที่สุดที่ทำให้ผลของคำตอบดีขึ้น พบว่ามีค่าเท่ากับ 7500 ดังนั้นจากการออกแบบการทดลองค่าพารามิเตอร์ที่เหมาะสมสำหรับการวางแผนการประกอบแผงวงจรอิเล็คทรอนิกส์ มีค่าเท่ากับ $HMCR = 0.3$ และ $T = 7500$ ตามลำดับ ลำดับสุดท้ายทำการทดสอบการค้นหาความบรรสานจำนวน 100,000 รอบ โดยใช้ค่าพารามิเตอร์ที่เหมาะที่สุด พบว่ามีคำตอบอยู่บนพาเรโต้ฟร้อนจำนวน 5 คำตอบ หรือคิดเป็น 0.005 เปอร์เซ็น

6.2 ปัญหาและอุปสรรคที่พบในการวิจัย

6.2.1 เนื่องจากผู้วิจัยใช้คอมพิวเตอร์ในการทดสอบ รุ่น Lenovo Y50 Intel® Core™ i7-4720HQ CPU 2.60 GHz processor, with 4.00 GB of RAM ซึ่งยังถือว่าเป็นเพียงคอมพิวเตอร์ใช้งานเบื้องต้น อาจจะยังไม่เหมาะสมในการหาคำตอบของปัญหาในครั้งนี้

6.2.2 เนื่องจากปัญหาในการทดสอบครั้งนี้สามารถ เขียนชุดคำสั่งที่มีการทำงานแบบขนาด (Parallel) เพื่อลดระยะเวลาในการทดสอบ แต่อย่างไรก็ตามการเขียนสุดคำสั่งดังกล่าวไม่เหมาะสมสำหรับการทำงานของคอมพิวเตอร์ในการทดสอบครั้งนี้

6.3 ข้อเสนอแนะ

6.3.1 ขั้นตอนการค้นหาความบรรสานที่ผู้วิจัยได้นำมาวิจัยในครั้งนี้ เป็นขั้นตอนวิธีการค้นหาความบรรสานแบบดั้งเดิม ซึ่งในปัจจุบันขั้นตอนวิธีการค้นหาความบรรสานได้ถูกพัฒนาปรับปรุงไปอย่างหลากหลาย โดยแต่ละขั้นตอนวิธีจะมีประสิทธิภาพในการคำตอบที่แตกต่างกันขึ้นอยู่กับชนิดของปัญหา ด้วยเหตุนี้เองสำหรับงานวิจัยต่อไปควรจะมีการเปรียบเทียบประสิทธิภาพของแต่ละขั้นตอนวิธีเพื่อให้เกิดการเลือกใช้อย่างเหมาะสมสำหรับการทดสอบต่อไปในอนาคต

6.3.2 จากการทดสอบในครั้งนี้จะเห็นได้ว่า พารามิเตอร์ความน่าจะเป็นในการปรับระดับเสียง (PAR) นั้นไม่ส่งผลต่อคำตอบ ซึ่งอาจจะเป็นไปได้จากหลากหลายสาเหตุ เช่น ลักษณะของพารามิเตอร์ดังกล่าวที่พยายามจะเลือกค่าเดิมในหน่วยความจำบรรสาน (Harmony memory) เมื่อทำการปรับค่าของพารามิเตอร์สูงขึ้น และพยายามจะเลือกค่าใหม่นอกเหนือค่าเดิมในหน่วยความจำบรรสาน เมื่อทำการปรับค่าของตัวพารามิเตอร์ต่ำลง เป็นต้น ดังนั้นสำหรับการศึกษาต่อไปในอนาคตจึงควรมีการศึกษาพารามิเตอร์ดังกล่าว

ภาคผนวก

ภาคผนวก ก
การพัฒนาตนเอง

การเข้าร่วมการประชุมวิชาการข่ายงานวิศวกรรมอุตสาหการ ประจำปี 2559

วันที่ 6 - 8 กรกฎาคม 2559 ณ โรงแรมโฆษะ จังหวัดขอนแก่น

การเข้าร่วม การนำเสนอบทความวิจัยในการประชุมวิชาการระดับนานาชาติ

“2017 International Conference on Robotics and Automation Sciences (ICRAS 2017)”)

วันที่ 26 - 29 สิงหาคาม 2560 ณ เขตบริหารพิเศษฮ่องกง สาธารณรัฐประชาชนจีน

มหาวิทยาลัยศิลปากร

รายการอ้างอิง

1. Güler, O., *Foundations of optimization*. Vol. 258. 2010: Springer Science & Business Media.
2. Talbi, E.-G., *Metaheuristics: from design to implementation*. Vol. 74. 2009: John Wiley & Sons.
3. Poungchompu, S., *Application of Linear Programming to Find the Ingredients of Compost from Rice Straw in a Pattern of the Lowest Cost.* KKU RESEARCH JOURNAL, 2011. **16**(5): p. 485-492.
4. Griva, I., S.G. Nash, and A. Sofer, *Linear and nonlinear optimization*. 2009: Siam.
5. Marler, R.T. and J.S. Arora, *Survey of multi-objective optimization methods for engineering.* Structural and multidisciplinary optimization, 2004. **26**(6): p. 369-395.
6. Wierzbicki, A.P., *A methodological approach to comparing parametric characterizations of efficient solutions*, in *Large-scale modelling and interactive decision analysis*. 1986, Springer. p. 27-45.
7. Steuer, R.E. and E.-U. Choo, *An interactive weighted Tchebycheff procedure for multiple objective programming.* Mathematical programming, 1983. **26**(3): p. 326-344.
8. Stadler, W., *Fundamentals of multicriteria optimization*, in *Multicriteria Optimization in Engineering and in the Sciences*. 1988, Springer. p. 1-25.
9. Miettinen, K. and M. Mäkelä, *Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation.* Journal of the Operational Research Society, 1999. **50**(9): p. 949-959.
10. Athan, T.W. and P.Y. Papalambros, *A note on weighted criteria methods for compromise solutions in multi-objective optimization.* Engineering Optimization, 1996. **27**(2): p. 155-176.
11. Gerasimov, E. and V. Repko, *Multicriterial optimization.* International Applied Mechanics, 1978. **14**(11): p. 1179-1184.

12. Charnes, A. and W.W. Cooper, *Goal programming and multiple objective optimizations: Part 1.* European Journal of Operational Research, 1977. **1**(1): p. 39-54.
13. Hwang, C.-L. and A.S.M. Masud, *Methods for multiple objective decision making*, in *Multiple Objective Decision Making—Methods and Applications*. 1979, Springer. p. 21-283.
14. Messac, A. and P.D. Hattis, *Physical programming design optimization for high speed civil transport.* Journal of aircraft, 1996. **33**(2): p. 446-449.
15. Messac, A. and C.A. Mattson, *Generating well-distributed sets of Pareto points for engineering design using physical programming.* Optimization and Engineering, 2002. **3**(4): p. 431-450.
16. Das, I. and J.E. Dennis, *Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems.* SIAM Journal on Optimization, 1998. **8**(3): p. 631-657.
17. Messac, A., A. Ismail-Yahaya, and C.A. Mattson, *The normalized normal constraint method for generating the Pareto frontier.* Structural and multidisciplinary optimization, 2003. **25**(2): p. 86-98.
18. Eschenauer, H., J. Koski, and A. Osyczka, *Multicriteria optimization—Fundamentals and motivation*, in *Multicriteria Design Optimization*. 1990, Springer. p. 1-32.
19. Nash Jr, J.F., *The bargaining problem.* Econometrica: Journal of the Econometric Society, 1950: p. 155-162.
20. Rao, S., *Game theory approach for multiobjective structural optimization.* Computers & Structures, 1987. **25**(1): p. 119-127.
21. Kothmayr, T., et al., *Comparing heuristics and linear programming formulations for scheduling of in-tree tasksets.* RTAS WiP, 2014.
22. พีระวัฒน์ โชคณัติ, กนกวรรณ สุภักดี, and ร. ปิตาคะโส, วิธีการวิวัฒนาการโดยใช้ผลต่างเพื่อแก้ปัญหาการจัดเส้นทางยานพาหนะ. วารสารเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏอุบลราชธานี, 2015. **5**(1): p. 47-58.
23. Geem, Z.W., J.H. Kim, and G.V. Loganathan, *A new heuristic optimization algorithm: harmony search.* simulation, 2001. **76**(2): p. 60-68.

24. Yang, X.-S., *Harmony search as a metaheuristic algorithm*, in *Music-inspired harmony search algorithm*. 2009, Springer. p. 1-14.
25. Geem, Z.W. *Improved harmony search from ensemble of music players*. in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. 2006. Springer.
26. Geem, Z.W., *Artificial satellite heat pipe design using harmony search.* Harmony Search Algorithm. Springer, Verlag Berlin Heidelberg, 2016: p. 423-433.
27. Geem, Z.W. *Harmony search algorithm for solving sudoku*. in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. 2007. Springer.
28. Ingram, G. and T. Zhang, *Overview of applications and developments in the harmony search algorithm.* Music-inspired harmony search algorithm, 2009: p. 15-37.
29. Kougias, I. and N. Theodosiou. *A new music-inspired harmony based optimization algorithm. Theory and applications*. in *International Conference on Protection and Restoration of the Environment X*. 2010.
30. Geem, Z.W., *Optimal cost design of water distribution networks using harmony search.* Engineering Optimization, 2006. **38**(03): p. 259-277.
31. Lee, K.S. and Z.W. Geem, *A new structural optimization method based on the harmony search algorithm.* Computers & structures, 2004. **82**(9): p. 781-798.
32. Geem, Z., *State-of-the-art in the structure of harmony search algorithm.* Recent advances in harmony search algorithm, 2010: p. 1-10.
33. Li, L., et al. *The harmony search algorithm in combination with particle swarm optimization and its application in the slope stability analysis*. in *Computational Intelligence and Security, 2009. CIS'09. International Conference on*. 2009. IEEE.
34. Forsati, R., et al. *Web page clustering using harmony search optimization*. in *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on*. 2008. IEEE.
35. Geem, Z.W. and H. Hwangbo. *Application of harmony search to multi-objective optimization for satellite heat pipe design*. in *Proceedings of*. 2006.

36. Panchal, A., *Harmony search in therapeutic medical physics.* Music-inspired Harmony search algorithm, 2009: p. 189-203.
37. Wikipedia. *Rosenbrock function*. 2017 [cited 2017 2017 Sep 20]; Available from: https://en.wikipedia.org/wiki/Rosenbrock_function.
38. Lee, K.S. and Z.W. Geem, *A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice.* Computer methods in applied mechanics and engineering, 2005. **194**(36): p. 3902-3933.
39. Mahdavi, M., M. Fesanghary, and E. Damangir, *An improved harmony search algorithm for solving optimization problems.* Applied mathematics and computation, 2007. **188**(2): p. 1567-1579.
40. Omran, M.G. and M. Mahdavi, *Global-best harmony search.* Applied mathematics and computation, 2008. **198**(2): p. 643-656.
41. Kaveh, A. and M. Ahangaran, *Social harmony search algorithm for continuous optimization.* Iranian Journal of Science and Technology. Transactions of Civil Engineering, 2012. **36**(C2): p. 121.
42. Nguyen, K., P. Nguyen, and N. Tran, *A hybrid algorithm of harmony search and bees algorithm for a university course timetabling problem.* Int. J. Comput. Sci. Issues, 2012. **9**(1): p. 12-17.
43. Seyedhosseini, S.M., M.J. Esfahani, and M. Ghaffari, *A novel hybrid algorithm based on a harmony search and artificial bee colony for solving a portfolio optimization problem using a mean-semi variance approach.* Journal of Central South University, 2016. **23**(1): p. 181-188.
44. Sivasubramani, S. and K. Swarup, *Multi-objective harmony search algorithm for optimal power flow problem.* International Journal of Electrical Power & Energy Systems, 2011. **33**(3): p. 745-752.
45. Ricart, J., et al., *Multiobjective harmony search algorithm proposals.* Electronic Notes in Theoretical Computer Science, 2011. **281**: p. 51-67.
46. Landa-Torres, I., et al., *A multi-objective grouping harmony search algorithm for the optimal distribution of 24-hour medical emergency units.* Expert Systems with Applications, 2013. **40**(6): p. 2343-2349.

47. Hoffman, K.L., M. Padberg, and G. Rinaldi, *Traveling salesman problem*, in *Encyclopedia of operations research and management science*. 2013, Springer. p. 1573-1578.

48. Grefenstette, J., et al. *Genetic algorithms for the traveling salesman problem*. in *Proceedings of the first International Conference on Genetic Algorithms and their Applications*. 1985. Lawrence Erlbaum, New Jersey (160-168).

49. Wang, K.-P., et al. *Particle swarm optimization for traveling salesman problem*. in *Machine Learning and Cybernetics, 2003 International Conference on*. 2003. IEEE.

50. Burkardt, J., *Data for the travelling salesperson problem.*

51. Ball, M.O. and M.J. Magazine, *Sequencing of insertions in printed circuit board assembly.* Operations Research, 1988. **36**(2): p. 192-201.

52. Khoo, L. and K. Loh, *A genetic algorithms enhanced planning system for surface mount PCB assembly.* The International Journal of Advanced Manufacturing Technology, 2000. **16**(4): p. 289-296.

53. Wong, H. and M.C. Leu, *Adaptive genetic algorithm for optimal printed circuit board assembly planning.* CIRP Annals-Manufacturing Technology, 1993. **42**(1): p. 17-20.

54. Sanchez, J.M. and J.W. Priest, *Optimal component-insertion sequenceplanning methodology for the semiautomatic assembly of printed circuit boards.* Journal of Intelligent Manufacturing, 1991. **2**(3): p. 177-187.

ประวัติผู้เขียน

ชื่อ-สกุล	ธนพัทธ์ ทองชั้น
วัน เดือน ปี เกิด	30 เมษายน 2536
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	วศ.บ. (วิทยาการและวิศวกรรมวัสดุ) มหาวิทยาลัยศิลปากร (2554-2557) วศ.ม. (การจัดการงานวิศวกรรม) มหาวิทยาลัยศิลปากร (2558-ปัจจุบัน)
ที่อยู่ปัจจุบัน	131/1 หมู่ 14 ตำบล คลองจินดา อำเภอ สามพราน จังหวัด นครปฐม 73110
ผลงานตีพิมพ์	T. Tongchan, T. Suroengrit, A. Sawangsri, S. Lapnonkawow, P. Tharawetcharak and T. Karot, “Process Capability Analysis of Bias Tire Extrusion Process: A Case Study of Tire Factory”, Industrial Engineering Network Conference 2016. (IE-NETWORK 2016), 7th – 8th July 2016. (Oral Presentation) T. Tongchan, C. Pornsing and T. Tonglim, “Harmony Search Algorithm’s Parameter Tuning for Traveling Salesman Problem”, in Proc. 2017 IEEE International Conference on Robotics and Automation Sciences (ICRAS 2017), 26th - 29th August 2017, Hong Kong, China (Oral Presentation)

มหาวิทยาลัยศิลปากร