สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รายงานการวิจัยฉบับสมบูรณ์

อัลกอริทึมทางวิวัฒนาการกับการจัดตารางการผลิตแบบยืดหยุ่นตามงาน

Evolutionary Algorithm for Solving the Flexible Job Shop Scheduling Problem

รศ.ดร. อาริต ธรรมโน

RCH
ว6550
2557

ตรวจหมู่.....
เลขทะเบียน...142769
รับเมื่อ วัน 25 เม.ย. 2559

b.....12785726
i.....

ได้รับทุนสนับสนุนงานวิจัยจากเงินรายได้ ประจำปีงบประมาณ 2557

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

รายละเอียดโครงงานวิจัย

ชื่อโครงการ (ภาษาไทย) อัลกอริทึมทางวิวัฒนาการกับการจัดตารางงานการผลิตแบบยืดหยุ่นตามงาน

แหล่งเงิน เงินรายได้ คณะเทคโนโลยีสารสนเทศ สจล.

ประจำปีงบประมาณ 2557 จำนวนเงินที่ได้รับการสนับสนุน 50,000 บาท

ระยะเวลาทำการวิจัย 1 ปี ตั้งแต่ 1 ตุลาคม 2556 ถึง 30 กันยายน 2557

ชื่อ-สกุล หัวหน้าโครงการ และผู้ร่วมโครงการวิจัย พร้อมระบุ หน่วยงานต้นสังกัด

1. รศ.ดร.อาริต ธรรมโน คณะเทคโนโลยีสารสนเทศ สจล. Email: arit@it.kmitl.ac.th

บทคัดย่อ

ปัญหาการจัดตารางการผลิตแบบยืดหยุ่นตามงาน (Flexible Job Shop Scheduling Problem) แบ่งออกเป็นสองปัญหาย่อยคือ 1) จะทำการจับคู่งานกับเครื่องจักรกลอย่างไรจึงจะมีความเหมาะสม และ 2) จะทำการจัดลำดับงานบนแต่ละเครื่องจักรกลนั้นอย่างไรให้ระยะเวลาดำเนินงานการผลิตน้อยที่สุด ในงานวิจัยนี้ได้ประยุกต์อัลกอริทึมทางปัญญาประดิษฐ์ที่มีชื่อว่า Marriage in Honey Bee Optimization (MBO) algorithm และ Artificial Bee Colony (ABC) Algorithm มาแก้ปัญหาการจัดตารางการผลิตแบบยืดหยุ่นตามงาน ซึ่ง MBO เป็นอัลกอริทึมที่เลียนแบบพฤติกรรมการผสมพันธุ์ของผึ้ง ส่วน ABC เป็นอัลกอริทึมที่เลียนแบบพฤติกรรมการหาอาหารของผึ้งงานที่จะพยายามค้นหาแหล่งอาหารซึ่งมีปริมาณน้ำหวานมากที่สุดเพื่อเป็นแหล่งอาหารแห่งใหม่ และนำอาหารเหล่านั้นกลับมาหล่อเลี้ยงสมาชิกตัวอื่นๆ ในรัง โดยเป้าหมายในการประยุกต์ใช้อัลกอริทึม MBO และ ABC ร่วมกันคือ นำข้อดีของทั้ง 2 อัลกอริทึมมาทำให้การค้นหาคำตอบของปัญหามีความเหมาะสมมากยิ่งขึ้น

คำสำคัญ: Flexible Job Shop Scheduling, Swarm Intelligence, Marriage in Honey Bee Optimization algorithm, Artificial Bee Colony Algorithm, Optimization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

Research Title: Evolutionary Algorithm for Solving the Flexible Job Shop Scheduling Problem

Researcher: Arit Thammano

Faculty: Information Technology **Department:** Information Technology

Abstract

Flexible Job Shop Scheduling Problem can be divided into 2 sub-problems: 1) assigning operations to appropriate machines and 2) finding a sequence of operations on each machine that minimizes the makespan. This research employs two powerful artificial intelligence techniques, Marriage in Honey Bee Optimization (MBO) algorithm and Artificial Bee Colony (ABC) Algorithm, for solving the flexible job shop scheduling problem. MBO is inspired by the process of marriage in real bees' colonies while ABC is inspired by the foraging behavior of honey bee swarm. The aim of the proposed hybrid algorithm is to combine the advantages of the two techniques for searching the solution space.

Keywords: Flexible Job Shop Scheduling, Swarm Intelligence, Marriage in Honey Bee Optimization algorithm, Artificial Bee Colony Algorithm, Optimization

King Mongkut's Institute of Technology Ladkrabang

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

กิตติกรรมประกาศ

โครงการวิจัยนี้ได้รับทุนสนับสนุนเงินวิจัยจากเงินรายได้ ประจำปีงบประมาณ 2557 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญ

สารบัญ	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	2
1.3 ขอบเขตของการวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการศึกษา	2
บทที่ 2 ทฤษฎีพื้นฐานและแนวคิดในการวิจัย	3
2.1 ปัญหาออฟติไมเซชั่นเชิงการจัด	3
2.2 ปัญหาการจัดตารางแบบยืดหยุ่นตามงาน	4
2.3 อัลกอริทึมการผสมพันธุ์ของผึ้ง	6
2.4 อัลกอริทึมการค้นหาอาหารของผึ้งงาน	8
2.5 พฤติกรรมของผึ้งกาฝาก	10
2.6 งานวิจัยที่เกี่ยวข้อง	10
2.6.1 An effective artificial bee colony algorithm for the flexible job shop scheduling problem	10
2.6.2 An effective shuffled flog-leaping algorithm for multi-objective flexible job shop scheduling problem	15
2.6.3 Solving Multi-Objective Flexible Job Shop Scheduling Problem Using Immune Algorithm	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญ (ต่อ)

สารบัญ หน้า

2.6.4 Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem..... 24
2.6.5 A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem..... 29

บทที่ 3 การแก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงานโดยประยุกต์ใช้อัลกอริทึมการผสมพันธุ์ของผึ้ง..... 34
3.1 ขั้นตอนการสร้างโซลูชั่นเริ่มต้น..... 34
3.2 ขั้นตอนการหาค่าความเหมาะสมของโซลูชั่นเริ่มต้น..... 36
3.3 ขั้นตอนการคัดเลือกโซลูชั่นผึ้งเพศผู้..... 36
3.4 ขั้นตอนการสร้างโซลูชั่นลูกเพศเมีย..... 37
3.4.1 ขั้นตอนการครอสโอเวอร์..... 37
3.4.2 ขั้นตอนการมิวเทชั่น..... 39
3.5 ขั้นตอนการปรับปรุงโซลูชั่นผึ้งนางพญา..... 40
3.6 ขั้นตอนการหาอาหารของผึ้งงาน..... 41
3.7 ขั้นตอนการสร้างโซลูชั่นลูกเพศผู้..... 43
3.7.1 โซลูชั่นลูกเพศผู้ที่เกิดจากโซลูชั่นผึ้งนางพญา..... 43
3.7.2 โซลูชั่นลูกเพศผู้ที่เกิดจากโซลูชั่นผึ้งงาน..... 45
3.8 ขั้นตอนการสร้างโซลูชั่นผึ้งกาฝาก..... 45
3.9 ขั้นตอนการคัดเลือกโซลูชั่นเริ่มต้นกลุ่มใหม่..... 46
3.10 ขั้นตอนการพิจารณาหยุดการค้นหา..... 46

บทที่ 4 การทดลองและผลการทดลอง..... 47
4.1 ชุดข้อมูลที่ใช้ในการทดลอง..... 47
4.2 ผลการทดลอง..... 48
4.3 ผลการเปรียบเทียบประสิทธิภาพ..... 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 5 สรุปผลและข้อเสนอแนะ 52
5.1 สรุปผลการทดลอง 52
5.2 ข้อเสนอแนะ 52

เอกสารอ้างอิง 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตัวอย่างปัญหาการจัดตารางแบบยืดหยุ่นตามงาน..... 4
3.1 ตารางปัญหาการจัดตารางแบบยืดหยุ่นตามงาน..... 34
4.1 ผลการทดลองอัลกอริทึมกับชุดข้อมูล Brandimarte..... 48
4.2 ผลการเปรียบเทียบอัลกอริทึมที่นำเสนอกับอัลกอริทึมอื่นๆ..... 50

King Mongkut's Institute of Technology Ladkrabang

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญรูป

รูปที่		หน้า
2.1	การแทนค่าโซลูชั่นของปัญหาการจัดตารางแบบยืดหยุ่นตามงาน	5
2.2	การคำนวณหาระยะทางรวมที่สั้นที่สุด	5
2.3	การครอสโอเวอร์แบบสองจุด	8
2.4	วิธีการครอสโอเวอร์ (งานวิจัยที่เกี่ยวข้อง 2.6.1)	12
2.5	ตัวอย่างกราฟเส้นทางของโอเปอเรชั่น (งานวิจัยที่เกี่ยวข้อง 2.6.1)	13
2.6	วิธีการครอสโอเวอร์ (งานวิจัยที่เกี่ยวข้อง 2.6.2)	17
2.7	กระบวนการ Local search สำหรับส่วนโอเปอเรชั่น (งานวิจัยที่เกี่ยวข้อง 2.6.2)	19
2.8	กระบวนการ Clonal Selection เบื้องต้น (งานวิจัยที่เกี่ยวข้อง 2.6.3)	21
2.9	โครงสร้างของอัลกอริทึมที่นำเสนอ (งานวิจัยที่เกี่ยวข้อง 2.6.3)	23
2.10	กระบวนการ Mutation สำหรับส่วนโอเปอเรชั่น (งานวิจัยที่เกี่ยวข้อง 2.6.3)	23
2.11	วิธีการหา Hamming distance ส่วนโอเปอเรชั่น (งานวิจัยที่เกี่ยวข้อง 2.6.4)	26
2.12	วิธีการหา Hamming distance ส่วนเครื่องจักร (งานวิจัยที่เกี่ยวข้อง 2.6.4)	27
2.13	ความสัมพันธ์ของอัตราการอพยพเข้าและอพยพออกเมื่อมีจำนวนสิ่งมีชีวิตแตกต่างกัน (งานวิจัยที่เกี่ยวข้อง 2.6.5)	29
2.14	วิธีการ Migration ของส่วนโอเปอเรชั่น (งานวิจัยที่เกี่ยวข้อง 2.6.5)	31
2.15	วิธีการ Migration ของส่วนเครื่องจักร (งานวิจัยที่เกี่ยวข้อง 2.6.5)	31
2.16	วิธีการมิวเทชั่นของส่วนโอเปอเรชั่น (งานวิจัยที่เกี่ยวข้อง 2.6.5)	32
2.17	วิธีการมิวเทชั่นของส่วนเครื่องจักร (งานวิจัยที่เกี่ยวข้อง 2.6.5)	32
3.1	ตัวอย่างรูปแบบโซลูชั่นเริ่มต้น	35
3.2	การครอสโอเวอร์แบบ Three-Point สำหรับส่วนเครื่องจักร	37
3.3	การครอสโอเวอร์แบบ MPOX สำหรับส่วนลำดับงาน	38
3.4	การมิวเทชั่นแบบ Inserting Move สำหรับส่วนลำดับงาน	39
3.5	การมิวเทชั่นแบบ Swapping Move สำหรับส่วนลำดับงาน	39
3.6	การมิวเทชั่นแบบสุ่มเลือกเครื่องจักรใหม่ สำหรับส่วนเครื่องจักร	39
3.7	ตัวอย่างเส้นทางวิกฤติของโซลูชั่นเริ่มต้นในรูปที่ 3.1	41
3.8	Tournament selection method	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.9 การสร้างโซลูชั่นลูกเพศผู้จากโซลูชั่นผึ้งนางพญา..... 43
3.10 วิธีฮิวริสติกแบบที่ 1 44
3.11 วิธีฮิวริสติกแบบที่ 2 44
3.12 วิธีฮิวริสติกแบบที่ 3 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันอุตสาหกรรมการผลิตมีอัตราการแข่งขันที่เพิ่มสูงขึ้น เนื่องจากมีผู้ผลิตรายใหม่เข้าสู่ตลาดอยู่เสมอ ดังนั้นอุตสาหกรรมการผลิตจะต้องคำนึงถึงขั้นตอนการผลิตที่มีประสิทธิภาพ ซึ่งจะส่งผลให้อุตสาหกรรมการผลิตสามารถแข่งขันและสร้างผลกำไรเพิ่มขึ้นได้ ทั้งนี้การที่จะดำเนินการผลิตให้มีประสิทธิภาพได้นั้น การวางแผนจัดตารางงานให้ใช้ระยะเวลาในการผลิตน้อยที่สุดเพื่อให้ทันส่งมอบงาน รวมทั้งทำให้สินค้าออกสู่ตลาดได้อย่างรวดเร็วเป็นสิ่งสำคัญมาก ซึ่งปัจจัยที่จะสนับสนุนให้สามารถวางแผนจัดตารางงานได้อย่างมีประสิทธิภาพคือ การทำให้งานและเครื่องจักรกลมีความสอดคล้องกัน และการจัดสรรลำดับงานบนเครื่องจักรกลให้ลงตัว หรือแม้กระทั่งการจับคู่เครื่องจักรกลให้เหมาะสมกับงาน

จากปัญหาที่จะทำอย่างไรให้อุตสาหกรรมการผลิตสามารถวางแผนจัดตารางงานได้ประสบความสำเร็จ จึงมีนักวิจัยพยายามประยุกต์ศาสตร์ทางคอมพิวเตอร์มาช่วยพัฒนาเพื่อให้เกิดความรวดเร็วและมีประสิทธิภาพ ปัญญาประดิษฐ์เป็นศาสตร์หนึ่งทางคอมพิวเตอร์ที่มีหลักการคือการดึงความสามารถด้านการคำนวณที่ซับซ้อนของคอมพิวเตอร์มาใช้แก้ปัญหาต่างๆ ทั้งนี้มีนักวิจัยจำนวนหนึ่งที่นำอัลกอริทึมทางปัญญาประดิษฐ์มาประยุกต์ใช้แก้ปัญหาการจัดตารางงาน (Scheduling problem) โดยอัลกอริทึมที่ถูกกล่าวถึงคือ อัลกอริทึมในกลุ่มความฉลาดทางชีววิทยา (Biologically inspired algorithm) ซึ่งมีหลักการคือ การเลียนแบบวิธีทางธรรมชาติและการดำรงอยู่ของสิ่งมีชีวิต ตัวอย่างอัลกอริทึมในกลุ่มนี้เช่น อัลกอริทึมอาณานิคมมด (Ant colony optimization algorithm) อัลกอริทึมการหาอาหารของฝูงนกหรือฝูงปลา (Particle swarm optimization algorithm) และอัลกอริทึมวิวัฒนาการทางพันธุกรรมของสิ่งมีชีวิต (Genetic algorithm) เป็นต้น

ในงานวิจัยนี้ได้ประยุกต์อัลกอริทึมการผสมพันธุ์ของผึ้ง (Marriage in honey bee optimization algorithm) ซึ่งเป็นอัลกอริทึมหนึ่งในกลุ่มความฉลาดทางชีววิทยา มาใช้แก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงาน (Flexible Job Shop Scheduling Problem)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาปัญหาการจัดตารางแบบยืดหยุ่นตามงาน และศึกษางานวิจัยที่เกี่ยวข้อง

1.2.2 เพื่อศึกษาอัลกอริทึมและทฤษฎีอื่นๆ ที่นำมาใช้ในงานวิจัย

1.2.3 เพื่อพัฒนาอัลกอริทึมการผสมพันธุ์ของผึ้งร่วมกับอัลกอริทึมการหาอาหารของผึ้งมาใช้แก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงาน

1.3 ขอบเขตการวิจัย

งานวิจัยนี้ได้ทำการทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐาน Brandimarte data set

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 มีความรู้ความเข้าใจปัญหาการจัดตารางแบบยืดหยุ่นตามงาน

1.4.2 มีความรู้ความเข้าใจขั้นตอนการทำงานของอัลกอริทึม และทฤษฎีต่างๆ ที่นำมาประยุกต์ใช้

1.4.3 ได้อัลกอริทึมที่มีประสิทธิภาพในการแก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงาน

1.5 ขั้นตอนการศึกษา

1.5.1 ศึกษาปัญหาการจัดตารางแบบยืดหยุ่นตามงานชนิดวัตถุประสงค์เดียว และศึกษางานวิจัยอื่นๆ ที่เกี่ยวข้อง

1.5.2 ศึกษาอัลกอริทึมการผสมพันธุ์ของผึ้งและอัลกอริทึมการหาอาหารของผึ้ง

1.5.3 พัฒนาอัลกอริทึมการผสมพันธุ์ของผึ้งร่วมกับอัลกอริทึมการหาอาหารของผึ้ง

1.5.4 ดำเนินการทดลองและทดสอบประสิทธิภาพของอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐานใน Flexible job shop scheduling library (FJSPLIB) และเปรียบเทียบผลลัพธ์ที่ได้กับผลลัพธ์จากงานวิจัยอื่นที่ทดสอบกับข้อมูลชุดเดียวกัน

1.5.5 สรุปผลการทดลองและผลการเปรียบเทียบ

1.5.6 จัดทำเอกสารประกอบการทำเล่มงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะแบ่งเนื้อหาออกเป็น 2 ส่วนคือ ส่วนแรกจะกล่าวถึงปัญหาการจัดตารางแบบยืดหยุ่นตามงานและทฤษฎีอื่นๆ ที่นำมาประยุกต์ใช้ในงานวิจัย ในส่วนที่ 2 จะกล่าวถึงงานวิจัยที่เกี่ยวข้อง

2.1 ปัญหาออฟติไมเซชั่นเชิงการจัด

ปัญหาออฟติไมเซชั่นเชิงการจัด (Combinatorial Optimization Problems) เป็นปัญหาหนึ่งในฟิลด์ของปัญหาออฟติไมเซชั่น (Optimization problems) ซึ่งมีเป้าหมายเพื่อค้นหาโซลูชั่นที่เหมาะสมมากที่สุดในขอบเขตของเงื่อนไขที่กำหนด ลักษณะของปัญหาจะเป็นการจัดเรียง (arrangement), การจัดกลุ่ม (grouping), การทำตามสั่ง (ordering) หรือการเลือก (Selection) โซลูชั่น โดยโซลูชั่นจะมีรูปแบบที่ไม่ต่อเนื่อง (Discrete) หรือสามารถแปลงให้มีรูปแบบที่ไม่ต่อเนื่องได้ ส่วนการประเมินความเหมาะสมของโซลูชั่นอาจจะประเมินจากค่าต่ำที่สุดหรือค่าสูงที่สุด ขึ้นอยู่กับปัญหาที่ศึกษา ทั้งนี้ปัญหาที่เกี่ยวข้องกับการออฟติไมเซชั่นเชิงการจัด ยกตัวอย่างเช่น routing and shortest path problems, matching and assignment problems, traveling salesman problems และ scheduling problem เป็นต้น พื้นฐานของปัญหาออฟติไมเซชั่นเชิงการจัดจะมี 3 ส่วนประกอบ [1] ที่สำคัญดังนี้

- S แสดงถึงขอบเขตพื้นที่การค้นหาที่กำหนด (Search space) และ s คือโซลูชั่นที่เป็นไปได้ (Feasible solution) ที่อยู่ในขอบเขตพื้นที่การค้นหาที่กำหนด
- f ฟังค์ชั่นวัตถุประสงค์ (Objective function) ที่สามารถเป็นได้ทั้งค่ามากที่สุด (Maximize) หรือค่าน้อยที่สุด (Minimize)
- Ω คือเซตของข้อจำกัดที่จะต้องปฏิบัติตามเพื่อให้สามารถแก้ปัญหาได้

อธิบายตามเงื่อนไข $\{s^* \in S : f(s^*) \leq f(s), \forall s \in S\}$ [2] ได้ว่าโซลูชั่นที่มีความเหมาะสม (s*) จะถูกค้นหาภายในขอบเขตพื้นที่ที่กำหนดไว้ โดยฟังค์วัตถุประสงค์ของ s* จะต้องมีความเหมาะสมมากกว่าโซลูชั่น s ทุกตัวตามเงื่อนไข Ω และเป็นสมาชิกใน S โดยปัญหาออฟติไมเซชั่นเชิงการจัดที่จะกล่าวถึงมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.2 ปัญหาการจัดตารางแบบยืดหยุ่นตามงาน

ปัญหาการจัดตารางแบบยืดหยุ่นตามงาน (Flexible Job Shop Scheduling Problem: FJSP) เป็นปัญหาที่มีฟังก์ชั่นวัตถุประสงค์เดียว (Single Objective) แบบค่าที่น้อยที่สุด (Minimization) และเป็นปัญหาแบบ NP-Hard ที่เกี่ยวข้องกับการจัดลำดับงานในตาราง โดยมีเงื่อนไขคือ ในตารางจะประกอบด้วยงานหลายๆ งาน (Jobs) ซึ่งแต่ละงานจะมีสมาชิกเป็นงานย่อยๆ หรือเรียกว่าโอเปอเรชั่น (Operation) และแต่ละโอเปอเรชั่นสามารถดำเนินการได้กับเครื่องจักร 1 เครื่องหรือมากกว่า ดังนั้นปัญหาที่เกิดขึ้นในการจัดตารางให้สอดคล้องกับเงื่อนไขข้างต้นคือ 1) จะทำการจับคู่โอเปอเรชั่นกับเครื่องจักรใดจึงจะมีความเหมาะสมมากที่สุด และ 2) จะทำการจัดลำดับโอเปอเรชั่นบนแต่ละเครื่องจักรอย่างไรให้ระยะเวลารวมในการทำงานน้อยที่สุด (Minimize makespan) ยกตัวอย่างปัญหาการจัดตารางแบบยืดหยุ่นตามงานได้ในตารางที่ 2.1

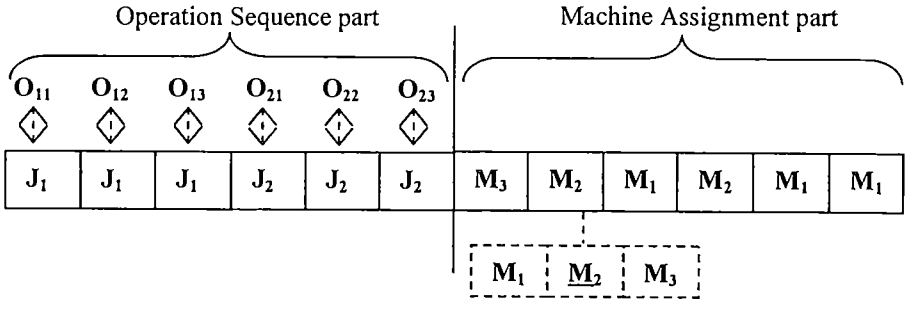
ตารางที่ 2.1 ตัวอย่างปัญหาการจัดตารางแบบยืดหยุ่นตามงาน

JOB	OPERATION	MACHINES DURATION (Days)		
		M1	M2	M3
Job_1	O_{11}	2	4	8
	O_{12}	5	4	7
	O_{13}	2	-	-
Job_2	O_{21}	5	4	3
	O_{22}	6	5	-
	O_{23}	2	-	6

จากตารางที่ 2.1 สามารถอธิบายได้ว่า ในตารางประกอบด้วย 2 Jobs คือ 1) Job_1 มีสมาชิกเป็นโอเปอเรชั่น O_{11}, O_{12} ซึ่งสามารถดำเนินการได้กับทุกเครื่องจักร และ O_{13} ซึ่งสามารถดำเนินการได้กับเครื่องจักร M_1 เท่านั้น โดยการดำเนินการของแต่ละโอเปอเรชั่นกับเครื่องจักรต่างชนิดกันก็จะใช้ระยะเวลาต่างกัน 2) Job_2 มีสมาชิกเป็นโอเปอเรชั่น O_{21} ซึ่งสามารถดำเนินการได้กับทุกเครื่องจักร โอเปอเรชั่น O_{22} สามารถดำเนินการได้กับเครื่องจักร M_1 และเครื่องจักร M_2 ส่วนโอเปอเรชั่น O_{23} สามารถดำเนินการได้กับเครื่องจักร M_1 และ M_3

รูปแบบการแทนค่าโซลูชั่นของปัญหาการจัดตารางแบบยืดหยุ่นตามงานนั้น จะแบ่งเป็น 2 ส่วนเสมอคือ 1) ส่วนของการเลือกเครื่องจักร (Machine assignment part) ให้กับแต่ละโอเปอเรชั่น และ 2) ส่วนของการจัดลำดับโอเปอเรชั่น (Operation sequence part) บนแต่ละเครื่องจักร ดังแสดงในรูปที่ 2.1 ซึ่งส่วนของการเลือกเครื่องจักรจะแสดงชื่อของเครื่องจักรที่ถูกเลือก และส่วนของการจัดลำดับโอเปอเรชั่นจะแสดงหมายเลขของงาน ที่สามารถบ่งบอกถึงลำดับของโอเปอเรชั่นได้ด้วย หมายความว่า J_1 ครั้งที่ 1 เป็นตัวแทนของโอเปอเรชั่น O_{11}, J_1 ครั้งที่ 2 เป็นตัวแทนของโอเปอเรชั่น O_{12} และ J_1 ครั้งที่ 3 เป็นตัวแทนของโอเปอเรชั่น O_{13} เป็นต้น ซึ่งจากรูปที่ 2.1 สามารถอธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

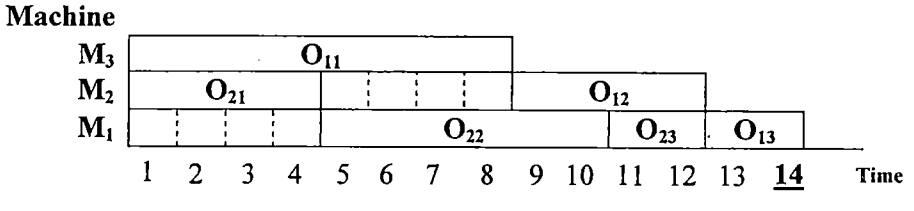


รูปที่ 2.1 การแทนค่าโซลูชั่นของปัญหาการจัดตารางแบบยืดหยุ่นตามงาน

โอเปอเรชั่น O_{11} ถูกกำหนดให้ดำเนินการกับเครื่องจักร M_3 โดยใช้ระยะเวลา 8 วัน, โอเปอเรชั่น O_{12} ถูกกำหนดให้ดำเนินการกับเครื่องจักร M_2 โดยใช้ระยะเวลา 4 วัน, โอเปอเรชั่น O_{13} ถูกกำหนดให้ดำเนินการกับเครื่องจักร M_1 โดยใช้ระยะเวลา 2 วัน, โอเปอเรชั่น O_{2i} ถูกกำหนดให้ดำเนินการกับเครื่องจักร M_2 โดยใช้ระยะเวลา 4 วัน, โอเปอเรชั่น O_{22} ถูกกำหนดให้ดำเนินการกับเครื่องจักร M_1 โดยใช้ระยะเวลา 6 วัน และโอเปอเรชั่น O_{23} ถูกกำหนดให้ดำเนินการกับเครื่องจักร M_1 โดยใช้ระยะเวลา 2 วัน

ทั้งนี้ปัญหาการจัดตารางแบบยืดหยุ่นตามงานจะมีเงื่อนไขที่เป็นข้อจำกัดดังต่อไปนี้

- แต่ละ Job จะไม่มีความสัมพันธ์กัน
- ลำดับโอเปอเรชั่นของแต่ละ Job จะต้องจัดเรียงตามเงื่อนไขลำดับก่อนหลัง เช่น โอเปอเรชั่น O_{12} ไม่สามารถเริ่มต้นได้หาก O_{11} ยังไม่เสร็จ เป็นต้น
- โอเปอเรชั่นแต่ละตัวสามารถทำได้บนเครื่องจักรหลายแบบ แต่จะเลือกได้เพียง 1 แบบเท่านั้น
- ในช่วงเวลา T ใดๆ เครื่องจักร M_k จะสามารถดำเนินการได้เพียง 1 โอเปอเรชั่นเท่านั้น



รูปที่ 2.2 การคำนวณหาระยะทางรวมที่สั้นที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

การคำนวนหาค่าความเหมาะสมตามฟังก์ชั่นวัตถุประสงค์ของปัญหาการจัดตารางแบบยืดหยุ่นตามงานจะสามารถทำได้ด้วยการหาระยะทางรวมที่สั้นที่สุด (Minimize Makespan) ตามสมการที่ 2.1

$$C_{\max} = \max_{1 \le i \le R} \{C_i\} \tag{2.1}$$

โดย C_{max} คือระยะเวลาสิ้นสุดรวมของทุกงาน และC_i คือระยะเวลาสิ้นสุดของ job_i จากรูปที่ 2.2 แสดงวิธีการหาระยะทางรวมที่สั้นสุดของโซลูชั่นในรูปที่ 2.1

2.3 อัลกอริทึมการผสมพันธุ์ของผึ้ง (Marriage in honey bee optimization algorithm)

ผึ้งตามธรรมชาติถูกแบ่งออกเป็น 3 ชนิดคือ ผึ้งนางพญา (Queen), ผึ้งเพศผู้ (Drone) และผึ้งงาน (Worker) ผึ้งแต่ละชนิดมีหน้าที่ในรังที่แตกต่างกัน ดังนี้ผึ้งนางพญาจะบินไปนอกรังเพื่อเลือกผสมพันธุ์กับผึ้งตัวผู้จากรังอื่นๆ โดยระหว่างการผสมพันธ์นั้นเสปิร์มของผึ้งตัวผู้จะถูกเก็บไว้ในถุงเก็บเสปิร์ม (Spermatheca) ของผึ้งนางพญา โดยผึ้งตัวผู้จะตายหลังจากการผสมพันธุ์ จากนั้นผึ้งนางพญาจะบินกลับรังและเริ่มวางไข่ โดยผึ้งนางพญาจะวางไข่ 2 ประเภทคือ ไข่ที่มาจากการปฏิสนธิกับเสปิร์ม (Fertilise Egg) ซึ่งจะเจริญเติบโตเป็นผึ้งเพศเมีย และไข่ที่ไม่ได้รับการปฏิสนธิ (Unfertilise Egg) ซึ่งจะเจริญเติบโตเป็นผึ้งเพศผู้ สำหรับผึ้งงานจะเป็นผึ้งเพศเมียที่มีหน้าที่หาอาหาร ปกป้องรัง ทำความสะอาดรัง เลี้ยงดูผึ้งนางพญา เลี้ยงดูผึ้งเพศผู้ รวมทั้งเลี้ยงดูตัวอ่อนผึ้ง และในบางครั้งผึ้งงานจะสามารถพัฒนารังไข่และวางไข่ได้ โดยไข่ของผึ้งงานจะพัฒนาเป็นผึ้งเพศผู้

อัลกอริทึมการผสมพันธุ์ของผึ้งมีขั้นตอนเลียนแบบพฤติกรรมการผสมพันธุ์ตามธรรมชาติของผึ้ง ซึ่งประกอบด้วยขั้นตอนดังต่อไปนี้

1. ขั้นตอนการกำหนดค่าตัวแปรเริ่มต้น

ในอัลกอริทึมการผสมพันธุ์ของผึ้งมีตัวแปรที่ต้องกำหนดคือ จำนวนของผึ้งนางพญา จำนวนของผึ้งตัวผู้ ขนาดถุงเก็บสเปิร์มของผึ้งนางพญา และจำนวนตัวอ่อนผึ้ง

2. ขั้นตอนการสร้างประชากรผึ้งเริ่มต้น

ทำการสุ่มสร้างประชากรผึ้งเริ่มต้นตามจำนวนที่กำหนดไว้ จากนั้นคำนวณหาค่าความเหมาะสมของผึ้งเริ่มต้น โดยผึ้งที่มีความเหมาะสมมากที่สุดจะถูกกำหนดให้เป็นผึ้งนางพญา และผึ้งเริ่มต้นส่วนที่เหลือจะถูกกำหนดให้เป็นผึ้งตัวผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3. ขั้นตอนการผสมพันธุ์ [2]

ผึ้งนางพญาเลือกผสมพันธุ์กับผึ้งตัวผู้โดยการพิจารณาค่าความน่าจะเป็นของการผสมพันธุ์ ผึ้งนางพญาจะเลือกผสมพันธุ์กับผึ้งตัวผู้ที่มีค่าความน่าจะเป็นในการผสมพันธุ์ที่เหมาะสม โดยผึ้งตัวผู้จะได้รับการผสมพันธุ์เมื่อค่าความน่าจะเป็นในการผสมพันธุ์มากกว่าค่าที่กำหนดไว้ ดังแสดงไว้ในสมการที่ 2.2

$$prob(Q,D) = e^{\frac{-difference}{speed(t)}} \tag{2.2}$$

เมื่อ prob(Q,D) คือ ค่าความน่าจะเป็นในการผสมพันธุ์ระหว่างผึ้งนางพญา Q และผึ้งตัวผู้ D

difference คือ ค่าผลต่างความเหมาะสมระหว่างผึ้งนางพญา Q และผึ้งตัวผู้ D

speed(t) คือ ค่าความเร็วของผึ้งนางพญา ณ เวลา t

จากสมการที่ 2.2 จะพบว่า ค่าความน่าจะเป็นในการผสมพันธุ์จะมีค่าสูงเมื่อความเร็วมีค่าสูงและความเหมาะสมระหว่างผึ้งราชินี Q และผึ้งตัวผู้ D ใกล้เคียงกัน ซึ่งหลังจากการผสมพันธุ์สเปิร์มของผึ้งตัวผู้จะถูกเก็บไว้ในถุงเก็บเสปิร์มของผึ้งนางพญา ทั้งนี้กระบวนการเลือกผึ้งตัวผู้เพื่อผสมพันธุ์จะดำเนินไปจนกระทั่งถุงเก็บเสปิร์มของผึ้งนางพญาเต็มหรือค่าพลังงานและความเร็วของผึ้งนางพญาลดลงถึงค่าที่กำหนดไว้ โดยอัตราการลดลงของค่าพลังงานและความเร็วสามารถคำนวณได้ตามสมการที่ 2.3 และ 2.4 ตามลำดับ โดย step คือ อัตราการลดลงของค่าพลังงาน และ α คือ อัตราการลดลงของค่าความเร็ว ซึ่งมีค่าอยู่ในช่วง [0,1]

$$energy(t+1) = energy(t) - step \tag{2.3}$$

$$speed(t+1) = \alpha * speed(t) \tag{2.4}$$

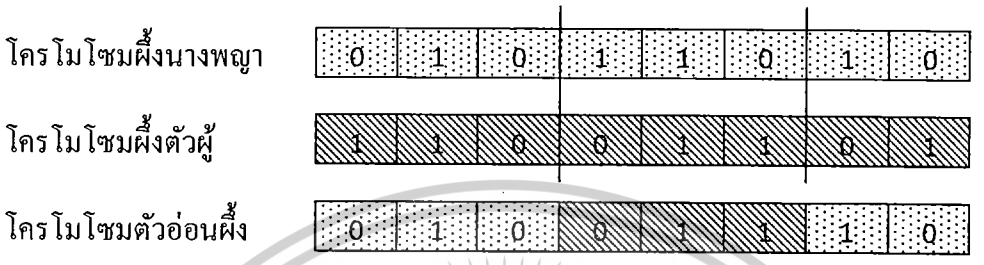
$$step(t+1) = (0.5 x energy(t)) / SpermathecaSize \tag{2.5}$$

4. ขั้นตอนการสร้างตัวอ่อนผึ้ง

ผึ้งนางพญาสุ่มเลือกเสปิร์มจากถุงเก็บเสปิร์มเพื่อครอสโอเวอร์ ซึ่งการครอสโอเวอร์เป็นการถ่ายทอดลักษณะทางพันธุกรรมจากผึ้งนางพญาและผึ้งตัวผู้ไปยังตัวอ่อนผึ้ง ยกตัวอย่างเช่น วิธีการครอสโอเวอร์แบบสองจุด (Two point crossover) ดังแสดงในรูปที่ 2.3 เป็นต้น หลังจากนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

จะทำการมิวเทชั่นตัวอ่อนที่ได้จากขั้นตอนการครอสโอเวอร์ ซึ่งกระบวนการมิวเตชั่นจะทำให้โครโมโซมของตัวอ่อนผึ้งเกิดความแตกต่างจากโครโมโซมของผึ้งนางพญาและผึ้งตัวผู้มากยิ่งขึ้น ยกตัวอย่างเช่น วิธีการมิวเทชั่นแบบสลับบิต สามารถทำได้โดยสุ่มสลับยีนส์ของโครโมโซมตัวอ่อนผึ้งจากบิต 0 เป็นบิต 1 หรือบิต 1 เป็นบิต 0 เป็นต้น



รูปที่ 2.3 การครอสโอเวอร์แบบสองจุด

ตัวอ่อนผึ้งจะได้รับการเลี้ยงดู (ปรับปรุง) โดยผึ้งงาน ซึ่งในอัลกอริทึมนี้ผึ้งงานแต่ละตัวจะถูกแทนด้วยฟังก์ชั่นฮิวริสติก ตัวอ่อนผึ้งแต่ละตัวจะได้รับการปรับปรุงโดยฟังก์ชั่นฮิวริสติกที่สุ่มเลือกขึ้นมาได้ด้วยวิธีวงล้อรูเล็ท ทั้งนี้ฟังก์ชั่นฮิวริสติกที่สามารถปรับปรุงตัวอ่อนผึ้งได้ดีที่สุดจะมีโอกาสถูกเลือกเพิ่มขึ้นในรอบถัดไป

5. ขั้นตอนการคัดเลือกประชากรใหม่

ในขั้นตอนสุดท้ายจะเป็นการเปรียบเทียบความเหมาะสมระหว่างผึ้งนางพญาและตัวอ่อนผึ้งที่มีดีที่สุด ผึ้งตัวที่มีความเหมาะสมมากกว่าจะถูกกำหนดให้เป็นผึ้งนางพญาตัวใหม่

6. ทำซ้ำในขั้นตอนที่ 3 - 5 จนกระทั่งครบตามเงื่อนไขที่กำหนดไว้

2.4 อัลกอริทึมการค้นหาอาหารของผึ้งงาน (Artificial bee colony algorithm)

การหาอาหารเป็นหน้าที่สำคัญของผึ้งงาน โดยผึ้งงานซึ่งทำหน้าที่หาอาหารถูกแบ่งเป็น 3 ชนิดคือ ผึ้งเอมพลอย (Employed bee) ผึ้งออนลุกเกอร์ (Onlooker bee) และผึ้งสเกาต์ (Scout bee) โดยกระบวนการหาอาหารเริ่มต้นเมื่อผึ้งเอมพลอยบินค้นหาแหล่งอาหารแห่งใหม่ให้กับรัง และจดจำแหล่งอาหารที่มีความเหมาะสมมากที่สุดไว้ จากนั้นผึ้งเอมพลอยจะบินกลับรังและแชร์ข้อมูลแหล่งอาหารที่พบให้กับผึ้งตัวอื่นในรัง โดยวิธีการเต้นในท่าทางที่แตกต่างกัน ผึ้งออนลุกเกอร์แต่ละตัวจะเลือกจดจำแหล่งอาหารที่เหมาะสมจากข้อมูลที่ผึ้งเอมพลอยแบ่งปันให้ และบินไปค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

แหล่งอาหารบริเวณใกล้เคียงแหล่งอาหารที่เลือกจดจำมา ส่วนผึ้งสเกาต์จะมีหน้าที่บินค้นหาแหล่งอาหารแห่งใหม่ในบริเวณรอบๆ รัง

อัลกอริทึมการหาอาหารของผึ้งมีขั้นตอนเลียนแบบพฤติกรรมการหาอาหารตามธรรมชาติของผึ้ง ซึ่งประกอบด้วยขั้นตอนดังต่อไปนี้

1. ขั้นตอนการกำหนดค่าตัวแปรเริ่มต้น

กำหนดจำนวนของผึ้งเอมพลอย และผึ้งออนลุกเกอร์ โดนในอัลกอริทึมนี้จะกำหนดให้ตำแหน่งของผึ้งเป็นตัวแทนของโซลูชั่นปัญหา และปริมาณอาหารของตำแหน่งนั้นๆ จะเป็นตัวแทนความเหมาะสมของโซลูชั่นปัญหา

2. ขั้นตอนของผึ้งเอมพลอย

2.1 ขั้นตอนการกำหนดตำแหน่งแหล่งอาหารเริ่มต้น

กำหนดตำแหน่งแหล่งอาหารเริ่มต้นให้ผึ้งเอมพลอยโดยวิธีการสุ่ม จากนั้นคำนวณหาค่าความเหมาะสมของตำแหน่งแหล่งอาหารนั้นๆ

2.2 ขั้นตอนการค้นหาแหล่งอาหารแห่งใหม่ของผึ้งเอมพลอย

ผึ้งเอมพลอยแต่ละตัวทำการค้นหาแหล่งอาหารแห่งใหม่ในบริเวณใกล้เคียงแหล่งอาหารเริ่มต้น ซึ่งผึ้งเอมพลอยจะเปรียบเทียบความเหมาะสมของแหล่งอาหารแห่งใหม่และแหล่งอาหารเริ่มต้น โดยผึ้งเอมพลอยจะเลือกจดจำเฉพาะแหล่งอาหารที่เหมาะสมกว่าเท่านั้น จากนั้นผึ้งเอมพลอยจะบินกลับไปที่รัง และกระจายข่าวสารของแหล่งอาหารที่พบให้กับผึ้งออนลุกเกอร์โดยวิธีการเต้นในท่าทางที่แตกต่างกัน ขึ้นอยู่กับความสมบูรณ์และระยะทางของแหล่งอาหาร

3. ขั้นตอนของผึ้งออนลุกเกอร์

3.1 ขั้นตอนการเลือกแหล่งอาหาร

ผึ้งออนลุกเกอร์จะรอรับข้อมูลแหล่งอาหารจากผึ้งเอมพลอย โดยผึ้งออนลุกเกอร์แต่ละตัวจะเลือกตำแหน่งของแหล่งอาหารที่มีความเหมาะสม ซึ่งแหล่งอาหารที่มีความเหมาะสมมากก็มีโอกาสถูกเลือกมากกว่า จากนั้นผึ้งออนลุกเกอร์จะบินไปยังตำแหน่งแหล่งอาหารที่เลือกไว้ ทั้งนี้อาจจะมีแหล่งอาหารของผึ้งเอมพลอยที่ผึ้งออนลุกเกอร์ไม่เลือก ซึ่งผึ้งเอมพลอยที่ถูกละเลยเหล่านี้จะถูกกำหนดให้ทำหน้าที่เป็นผึ้งสเกาต์

3.2 ขั้นตอนการค้นหาแหล่งอาหารแห่งใหม่ของผึ้งออนลุกเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ผึ้งออนลุกเกอร์แต่ละตัวจะค้นหาแหล่งอาหารแห่งใหม่ในบริเวณใกล้เคียงแหล่งอาหารเริ่มต้น ซึ่งผึ้งออนลุกเกอร์จะเปรียบเทียบความเหมาะสมของแหล่งอาหารแห่งใหม่และแหล่งอาหารเริ่มต้น โดยผึ้งออนลุกเกอร์จะเลือกจดจำเฉพาะแหล่งอาหารที่เหมาะสมกว่าเท่านั้น

4. ขั้นตอนของผึ้งสเกาต์

ผึ้งสเกาต์แต่ละตัวจะทำหน้าที่สุ่มค้นหาแหล่งอาหารแห่งใหม่

5. ขั้นตอนการคัดเลือกประชากรใหม่

ในขั้นตอนสุดท้ายจะเป็นการรวมและเปรียบเทียบแหล่งอาหารผึ้งเอมพลอย ผึ้งออนลุกเกอร์และผึ้งสเกาต์ แหล่งอาหารที่ดีที่สุด N ลำดับแรกจะถูกกำหนดให้เป็นแหล่งอาหารเริ่มต้นในรอบถัดไป

6. ทำซ้ำในขั้นตอนที่ 2 - 5 จนกระทั่งครบตามเงื่อนไขที่กำหนดไว้

2.5 พฤติกรรมของผึ้งกาฝาก

ในธรรมชาตินั้น ผึ้งงานจะสามารถวางไข่ได้เมื่อผึ้งราชินีสูญหายไปจากรัง หรือผึ้งราชินีอ่อนแอจนไม่สามารถหลั่งฟีโรโมนออกมาควบคุมการพัฒนารังไข่ของผึ้งงานได้ ซึ่งผึ้งงานเหล่านี้จะสามารถวางไข่ที่เจริญเติบโตเป็นผึ้งเพศผู้ แต่ผึ้งงานชนิด Cape honeybee (Apis mellifera capensis) หรือเรียกว่าผึ้งกาฝาก มีพฤติกรรมที่แตกต่างจากผึ้งงานชนิดอื่น เนื่องจากผึ้งกาฝากสามารถวางไข่ที่เจริญเติบโตเป็นผึ้งเพศเมียได้ ดังนั้นเมื่อผึ้งกาฝากบุกรุกหรือแฝงตัวเข้าไปยังรังของผึ้งชนิดอื่นๆที่กำลังอ่อนแออยู่ ผึ้งกาฝากจะสามารถวางไข่ที่สามารถเจริญเติบโตเป็นผึ้งงานหรือผึ้งราชินีได้ [3] นอกจากนั้นผึ้งกาฝากยังสามารถเปลี่ยนฟีโรโมนของตัวมันเองให้เหมือนกับฟีโรโมนของผึ้งราชินี ส่งผลให้ผึ้งกาฝากสามารถสถาปนาตัวเองเป็นผึ้งราชินีและครอบครองรังที่มันเข้าไปบุกรุกได้อย่างสมบูรณ์ [4]

2.6 งานวิจัยที่เกี่ยวข้อง

2.6.1 An effective artificial bee colony algorithm for the flexible job shop scheduling problem

Ling Wang, Gang Zhou และ Shengyao [5] ได้ประยุกต์ใช้อัลกอริทึมที่เลียนแบบพฤติกรรมอันชาญฉลาดในการดำรงชีวิตรวมกันเป็นสังคมของสัตว์ (Swarm Intelligence) ชนิดหนึ่งที่เรียกว่า อัลกอริทึมการหาอาหารของผึ้งงาน (Honey Bee Foraging) ตามหลักการทางชีววิทยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ผึ้งงานในรังจะถูกแบ่งออกเป็น 3 ชนิดตามหน้าที่ความรับผิดชอบในรัง คือ 1) ผึ้งเอมพลอย (Employed bees) มีหน้าที่ในการสำรวจค้นหาแหล่งอาหารใหม่บริเวณโดยรอบแหล่งอาหารปัจจุบันของรัง 2) ผึ้งออนลุกเกอร์ (Onlooker Bees) จะมีหน้าที่รออยู่ในรังเพื่อรับข้อมูลแหล่งอาหารจากผึ้งเอมพลอย 3) ผึ้งสเกาต์ (Scout Bees) จะทำหน้าที่ค้นหาแหล่งอาหารบริเวณรอบๆ รัง ดังนั้นธรรมชาติในการค้นหาแหล่งอาหารของผึ้งงานสามารถอธิบายเป็นลำดับได้ดังนี้ เริ่มต้นที่ผึ้งเอมพลอยจะบินไปยังแหล่งอาหารของรังและทำการค้นหาแหล่งอาหารแห่งใหม่ในบริเวณโดยรอบ ซึ่งผึ้งเอมพลอยจะเลือกจดจำตำแหน่งของแหล่งอาหารที่มีปริมาณอาหารมากกว่าและบินกลับมาที่รังพร้อมกับอาหารจำนวนหนึ่ง จากนั้นผึ้งเอมพลอยจะส่งสัญญาณบอกถึงปริมาณอาหารและระยะทางของแหล่งอาหารที่ไปค้นพบมาโดยการเต้นในท่าทางที่แตกต่างกันไป และในขณะเดียวกันผึ้งออนลุกเกอร์จะสังเกตท่าเต้นของผึ้งเอมพลอยแต่ละตัว และตัดสินใจเลือกจดจำแหล่งอาหารที่คิดว่ามีความเหมาะสม จากนั้นจะบินไปยังแหล่งอาหารของผึ้งเอมพลอยที่เลือก ผึ้งออนลุกเกอร์จะสำรวจและทำการค้นหาแหล่งอาหารใหม่ในบริเวณใกล้เคียง ผึ้งสเกาต์มีหน้าที่ค้นหาแหล่งอาหารใหม่ในบริเวณรอบๆ รัง ในกรณีที่ผึ้งออนลุกเกอร์และผึ้งสเกาต์สามารถค้นพบแหล่งอาหารที่มีความเหมาะสมมากกว่าผึ้งเอมพลอย ผึ้งออนลุกเกอร์และผึ้งสเกาต์จะเปลี่ยนหน้าที่ไปเป็นผึ้งเอมพลอย

งานวิจัยนี้จะประยุกต์ใช้อัลกอริทึมการหาอาหารของผึ้งงานเพื่อนำมาใช้แก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงาน (Flexible Job Shop Scheduling Problem) โดยแหล่งอาหารเป็นตัวแทนโซลูชั่นของปัญหา และความเหมาะสมของแหล่งอาหารเป็นตัวแทนแสดงความเหมาะสมของโซลูชั่นปัญหา ซึ่งอัลกอริทึมที่งานวิจัยนี้นำเสนอมีขั้นตอนดังต่อไปนี้

1. ขั้นตอนการกำหนดประชากรเริ่มต้น (Population initialization)

งานวิจัยนี้ประยุกต์ใช้กลยุทธ์ที่มีความแตกต่างกันเพื่อทำให้ประชากรเริ่มต้นเกิดความหลากหลาย โดยจะแบ่งการสร้างประชากรเริ่มต้นเป็น 2 ส่วนคือ 1) ส่วนการเลือกเครื่องจักร (Machine Assignment) ของโซลูชั่นเริ่มต้นจะถูกกำหนดจาก 3 กฎคือ 10% ของจำนวนโซลูชั่นเริ่มต้นจะถูกสร้างจากกฎการสุ่มเลือก (Random rule), 30% ของจำนวนโซลูชั่นเริ่มต้นถูกสร้างจากกฎการเลือกเครื่องจักรที่มีภาระงานน้อยที่สุดเมื่อพิจารณาจากโอเปอเรชั่นที่สัมพันธ์กัน (Local minimum processing time rule), ที่เหลือ 60% ของจำนวนโซลูชั่นเริ่มต้นถูกสร้างจากกฎการเลือกเครื่องจักรที่มีภาระงานน้อยที่สุดเมื่อพิจารณาจากโอเปอเรชั่นทั้งหมด (Global minimum processing time rule) และ 2) ส่วนการจัดเรียงโอเปอเรชั่น (Operation Sequence) ของโซลูชั่นเริ่มต้นจะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

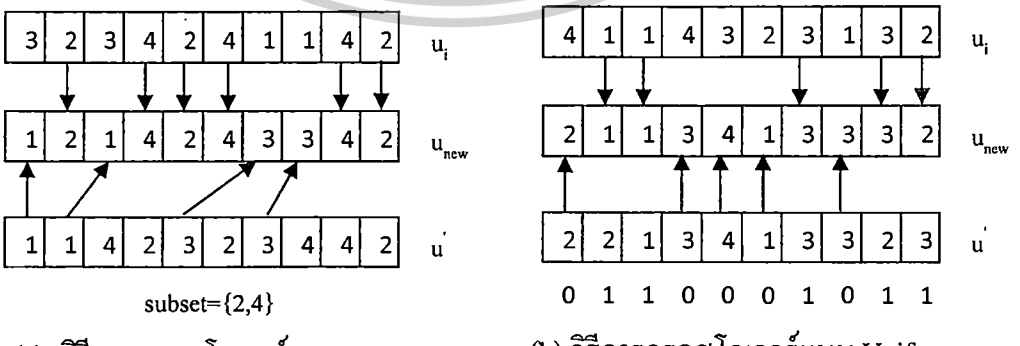
กำหนดจาก 3 กฎคือ 20% ของจำนวนโซลูชั่นเริ่มต้นถูกสร้างจากกฎการสุ่มจัดเรียง (Random rule), 40% ของจำนวนโซลูชั่นเริ่มต้นถูกสร้างจากกฎการเลือกพิจารณา job ที่ประกอบด้วยโอเปอเรชั่นซึ่งยังไม่ถูกเลือกเหลืออยู่มากที่สุด (Most number of operations remaining rule), และ 40% ของจำนวนโซลูชั่นเริ่มต้นถูกสร้างจากกฎการพิจารณา job ที่เวลาการดำเนินงานรวมยังคงเหลืออยู่มากที่สุด (Most time remaining rule) โซลูชั่นที่มีความเหมาะสมมากที่สุดจะถูกบันทึกให้เป็นโซลูชั่น G_{best}

2. ขั้นตอนของผึ้งเอมพลอย (Employed bee phase)

งานวิจัยนี้ได้นำกระบวนการครอสโอเวอร์และมิวเทชั่นมาใช้เพื่อค้นหาโซลูชั่นใหม่ที่อยู่ในบริเวณใกล้เคียงกับโซลูชั่นเริ่มต้น โดยวิธีการครอสโอเวอร์แบบ MPOX ถูกนำมาใช้สำหรับการสร้างลำดับโอเปอเรชั่นของโซลูชั่นใหม่ ดังแสดงในรูปที่ 2.4(a) และวิธีการครอสโอเวอร์แบบ Uniform ถูกนำมาใช้สำหรับการเลือกเครื่องจักรของโซลูชั่นใหม่ ดังแสดงในรูปที่ 2.4(b) หลังจากนั้นจะเป็นกระบวนการมิวเทชั่น ซึ่งจะช่วยทำให้เกิดความหลากหลาย โดยการสุ่มเลือกโอเปอเรชั่น O_{ij} จากนั้นทำการสุ่มเลือกเครื่องจักรใหม่ให้กับโอเปอเรชั่น O_{ij}

จากรูปที่ 2.4(a) และ 2.4(b) โซลูชั่น U_i คือโซลูชั่นเริ่มต้นที่กำลังพิจารณาอยู่ โซลูชั่น U' ได้มาจากการเลือกโซลูชั่นที่มีความเหมาะสมมากที่สุดจากโซลูชั่นเริ่มต้น 3 ตัวที่สุ่มเลือกมา และโซลูชั่น U_{new} คือโซลูชั่นใหม่ที่ได้จากการครอสโอเวอร์

โซลูชั่นใหม่ที่ได้จะถูกนำไปเปรียบเทียบกับโซลูชั่น U_i และโซลูชั่นที่มีความเหมาะสมมากกว่าจะถูกบันทึกให้เป็นโซลูชั่นผึ้งเอมพลอย จากนั้นอัพเดทโซลูชั่น G_{best}



(a) วิธีการครอสโอเวอร์แบบ MPOX (b) วิธีการครอสโอเวอร์แบบ Uniform (Uniform crossover)

รูปที่ 2.4 วิธีการครอสโอเวอร์ [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

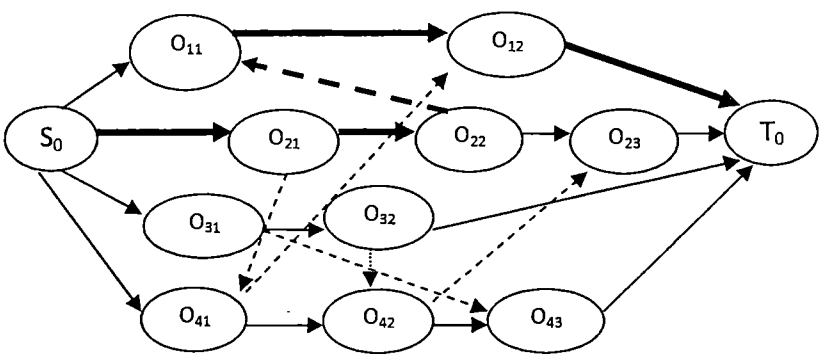
3. ขั้นตอนของผึ้งออนลุกเกอร์ (Onlooker bee phase)

งานวิจัยนี้จะเลือกโซลูชั่นผึ้งออนลุกเกอร์จากโซลูชั่นผึ้งเอมพลอย โดยโซลูชั่นผึ้งเอมพลอยที่มีความเหมาะสมมากกว่าจะมีโอกาสที่จะถูกเลือกเป็นโซลูชั่นเริ่มต้นของผึ้งออนลุกเกอร์มากเช่นกัน ทั้งนี้ความน่าจะเป็นที่โซลูชั่นผึ้งเอมพลอยแต่ละตัวจะถูกเลือกแสดงไว้ในสมการที่ 2.5 โดย p_i คือค่าความน่าจะเป็นที่โซลูชั่นผึ้งเอมพลอยตัวที่ i จะถูกเลือก, $f(s_i)$ คือค่าความเหมาะสมของโซลูชั่นผึ้งเอมพลอยตัวที่ i และ n_{fs} คือ จำนวนโซลูชั่นผึ้งเอมพลอยทั้งหมด ทั้งนี้โซลูชั่นผึ้งเอมพลอยที่ได้รับเลือกจะถูกกำหนดให้เป็นโซลูชั่นเริ่มต้นของผึ้งออนลุกเกอร์

$$p_i = \frac{[f(s_i)]^{-1}}{\sum_{j=1}^{n_{fs}} [f(s_j)]^{-1}} \tag{2.5}$$

ขั้นตอนนี้จะเป็นการค้นหาโซลูชั่นใหม่บริเวณโดยรอบโซลูชั่นเริ่มต้นของผึ้งออนลุกเกอร์ โดยกระบวนการค้นหาจะทำบนเส้นทางวิกฤติ (Local search on critical path) ทั้งนี้เส้นทางวิกฤติคือ ลำดับของโอเปเรชั่นที่เรียงต่อกันแล้วมีระยะทางมากที่สุด (Minimize Makespan) ยกตัวอย่างจากรูปที่ 2.5 เส้นทางวิกฤติคือ (S_G, O_{21}, O_{22}, O_{11}, O_{12}, T_G) โดยโอเปเรชั่นบนเส้นทางวิกฤติจะถูกเรียกว่า โอเปอเรชั่นวิกฤติ (Critical Operations)

วิธีการค้นหาโซลูชั่นใหม่ในพื้นที่โลคอลบนเส้นทางวิกฤติสามารถทำได้โดยการเปลี่ยนโอเปอเรชั่นวิกฤติทุกตัวไปยังตำแหน่งใหม่ที่มีความเป็นไปได้ เพื่อให้เส้นทางวิกฤติใหม่ที่ได้มีระยะทางสั้นลง ยกตัวอย่างเช่น การลบโอเปอเรชั่น O_{22} ออกจากเส้นทางวิกฤติ จากนั้นค้นหา



รูปที่ 2.5 ตัวอย่างกราฟเส้นทางของโอเปอเรชั่น [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ตำแหน่งใหม่ที่มีความเป็นไปได้เพื่อเพิ่มโอเปอเรชั่น O_{22} กลับเข้าไป ทั้งนี้การเปลี่ยนทุกโอเปอเรชั่นวิกฤติไปยังตำแหน่งอื่นจะทำให้เกิดเป็นโซลูชั่นใหม่ขึ้น โดยโซลูชั่นใหม่จะได้รับการยอมรับให้แทนที่โซลูชั่นเดิมเมื่อ 1) โซลูชั่นใหม่ให้คำตอบเป็นระยะทางที่สั้นกว่าเดิม 2) หากโซลูชั่นใหม่มีระยะทางที่เท่ากับโซลูชั่นเดิม จำนวนเส้นทางวิกฤติที่ได้จะต้องลดน้อยลง ดังนั้นโซลูชั่นที่มีความเหมาะสมมากกว่าจะถูกบันทึกให้เป็นโซลูชั่นของผึ้งออนลุกเกอร์ จากนั้นอัพเดทโซลูชั่น G_{best}

4. ขั้นตอนของผึ้งสเกาต์ (Scout bee phase)

ขั้นตอนนี้จะเป็นการสุ่มค้นหาโซลูชั่นของผึ้งสเกาต์ตามกระบวนการเช่นเดียวกับขั้นตอนที่ 1 โดยโซลูชั่นผึ้งสเกาต์จะถูกนำไปเปรียบเทียบกับโซลูชั่นผึ้งเอมพลอยที่มีความเหมาะสมน้อยที่สุด และโซลูชั่นผึ้งสเกาต์จะถูกนำไปแทนที่ในกรณีที่มีความเหมาะสมมากกว่า จากนั้นอัพเดทโซลูชั่น G_{best}

5. ขั้นตอนการพิจารณาเงื่อนไขการหยุดค้นหา (Stopping Criterion)

หากไม่เป็นไปตามเงื่อนไขการหยุด ให้กลับไปทำซ้ำในขั้นตอนที่ 2 ถึง 4

ในงานวิจัยนี้ได้ทำการทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐาน 2 ชุดคือ ชุดข้อมูล Kacem instances ซึ่งประกอบด้วย 5 ชุดข้อมูลย่อย และชุดข้อมูล BRdata instances ซึ่งประกอบด้วย 10 ชุดข้อมูลย่อย จากนั้นนำผลการทดสอบที่ได้ไปเปรียบเทียบกับอัลกอริทึมของงานวิจัยอื่นๆ ที่ทดสอบกับข้อมูลชุดเดียวกัน ซึ่งผลลัพธ์จากการเปรียบเทียบแสดงให้เห็นว่า ในชุดข้อมูล Kacem instances อัลกอริทึมที่งานวิจัยนี้นำเสนอให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมอื่นๆ ใน 3 ชุดข้อมูลย่อยจากทั้งหมด 5 ชุดข้อมูลย่อย และในชุดข้อมูล BRdata instances อัลกอริทึมที่งานวิจัยนี้นำเสนอให้ผลลัพธ์ที่เหมาะสมกว่าอัลกอริทึมอื่นๆ ในทุกชุดข้อมูลย่อย

ดังนั้นจากการศึกษางานวิจัย An effective artificial bee colony algorithm for the flexible job shop scheduling problem สามารถวิเคราะห์ให้เห็นจุดเด่นที่สำคัญคือ อัลกอริทึมที่นำเสนอมีขั้นตอนการทำงานหลายขั้นตอนคือ ขั้นตอนการค้นหาของผึ้งเอมพลอย ขั้นตอนการค้นหาของผึ้งออนลุกเกอร์ และขั้นตอนการค้นหาของผึ้งสเกาต์ ซึ่งขั้นตอนเหล่านี้ช่วยทำให้สามารถวางแผนเพื่อแก้ปัญหาได้อย่างหลากหลายวิธีและครบถ้วนทั้ง Global search และ Local search จึงทำให้ผลลัพธ์ที่ได้มีความเหมาะสมมากกว่าอัลกอริทึมอื่นๆ ที่นำมาเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

อย่างไรก็ตามอัลกอริทึมที่นำเสนอนี้ยังมีจุดด้อยที่สำคัญคือ กระบวนการในขั้นตอนของผึ้งสเกาต์ที่งานวิจัยนี้นำเสนอ ยังใช้วิธีการค้นหารูปแบบเดิมซึ่งเคยใช้ไปแล้วกับขั้นตอนการค้นหาโซลูชั่นเริ่มต้น ดังนั้นจึงทำให้มีโอกาสที่จะได้โซลูชั่นคำตอบเดิม ซึ่งทำให้ขั้นตอนการค้นหาของอัลกอริทึมไม่เกิดความหลากหลาย

2.6.2 An effective shuffled flog-leaping algorithm for multi-objective flexible job shop scheduling problem

Janquing Li, Quanke Pan และ Shengxian Xie [6] ได้ประยุกต์ใช้อัลกอริทึม Shuffled-flog leaping algorithm (SFLA) เพื่อแก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงานชนิดหลายวัตถุประสงค์ (Multi-objective flexible job shop scheduling problem) โดย SFLA เป็นอัลกอริทึมที่มีขั้นตอนเลียนแบบพฤติกรรมการรวมกลุ่มของกบเพื่อหาอาหาร ซึ่งกบแต่ละกลุ่มจะกระโดดไปยังโขดหินต่างๆ ที่โผล่อยู่กลางแหล่งน้ำแล้วค้นหาโขดหินที่มีอาหารอยู่มากที่สุด โดยกบเหล่านี้จะสื่อสารแลกเปลี่ยนข้อมูลอาหารระหว่างกัน ดังนั้น SFLA จะมีกระบวนการเริ่มต้นที่การสร้างโซลูชั่นกบเริ่มต้น จากนั้นทำการแบ่งโซลูชั่นกบเริ่มต้นเป็นกลุ่มย่อยๆ ซึ่งกลุ่มย่อยนี้จะถูกเรียกว่า มีมีเพล็ก (Memeplex) จำนวน m กลุ่ม โดยแต่ละมีมีเพล็กจะมีโซลูชั่นกบจำนวน n ตัว ในการแบ่งกลุ่มจะจัดให้โซลูชั่นกบที่มีความเหมาะสมสูงสุดอยู่ในมีมีเพล็กที่หนึ่ง โซลูชั่นกบตัวที่มีความเหมาะสมลำดับที่สองจัดให้อยู่ในมีมีเพล็กที่สอง โซลูชั่นกบตัวที่ m จัดให้อยู่ในมีมีเพล็กที่ m และโซลูชั่นกบตัวที่ m+1 จัดให้อยู่ในมีมีเพล็กที่หนึ่งอีกครั้ง จากนั้นจะเป็นการปรับปรุงโซลูชั่นกบที่แย่ที่สุดของแต่ละมีมีเพล็กโดยกระบวนการ Local search หลังจากกระบวนการปรับปรุงหากพบว่าโซลูชั่นกบตัวที่แย่ที่สุดยังไม่ดีขึ้น อัลกอริทึมจะทำการตัดโซลูชั่นกบตัวนี้ออกไป แล้วจะทำการสุ่มสร้างโซลูชั่นกบตัวใหม่ขึ้นมาแทน หลังจากนั้นเป็นขั้นตอนของการแลกเปลี่ยนข้อมูลระหว่างแต่ละมีมีเพล็กโดยกระบวนการสับกลุ่ม (Shuffled) โดยการรวมโซลูชั่นกบทุกตัวจากทุกมีมีเพล็กเข้าด้วยกัน แล้วทำการเรียงโซลูชั่นกบตามลำดับความเหมาะสม โซลูชั่นกบที่มีความเหมาะสมมากที่สุดจะถูกกำหนดให้เป็นโซลูชั่นคำตอบของอัลกอริทึม

ทั้งนี้ในงานวิจัยนี้จะกำหนดให้กบเป็นตัวแทนของโซลูชั่นปัญหา และความเหมาะสมของตำแหน่งที่กบอยู่แทนค่าความเหมาะสมของโซลูชั่น ทั้งนี้อัลกอริทึมที่งานวิจัยนี้นำเสนอมีขั้นตอนดังต่อไปนี้

1) การกำหนดโซลูชั่นเริ่มต้น (Population Initialization)

เพื่อให้โซลูชั่นเริ่มต้นเกิดความหลากหลาย งานวิจัยนี้จึงกำหนดให้โซลูชั่นเริ่มต้นถูกสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

มาจากวิธีการที่แตกต่างกัน โดยจะแบ่งโซลูชั่นออกเป็น 2 ส่วนคือ ส่วนของเครื่องจักร (Machine Selection) และส่วนของโอเปอเรชั่น (Operation Sequence) จากนั้นแยกพิจารณาเพื่อสร้างโซลูชั่นเริ่มต้นในแต่ละส่วน โดยโซลูชั่นเริ่มต้นในส่วนของเครื่องจักรจะถูกสุ่มสร้างจาก 2 กฎคือ 1) กฎการเลือกเครื่องจักรที่มีระยะเวลาการดำเนินงานน้อยที่สุด (Operation minimum processing time rule) และ 2) กฎการเลือกเครื่องจักรตามภาระงาน (Global machine workload balance rule) โดยเครื่องจักรที่มีภาระงานน้อยที่สุดจะถูกเลือก ทั้งนี้โซลูชั่นเริ่มต้นในส่วนของโอเปอเรชั่นจะถูกสุ่มสร้างจาก 3 กฎคือ 1) กฎการจัดลำดับโอเปอเรชั่นโดยการพิจารณาให้ความสำคัญกับ job ที่มีระยะเวลาเหลือมากที่สุดก่อน (Most work remaining rule) 2) กฎการจัดลำดับโอเปอเรชั่นโดยการพิจารณาให้ความสำคัญกับ job ที่เหลือจำนวนโอเปอเรชั่นที่ยังไม่ถูกดำเนินการอยู่มากที่สุดก่อน (Most number of operation remaining rule) และ 3) กฎการจัดลำดับโอเปอเรชั่นโดยวิธีการสุ่ม (Random Rule)

2) การแบ่งกลุ่มโซลูชั่น (Population Partition)

ในขั้นตอนนี้จะเป็นการจัดโซลูชั่นเริ่มต้นตามลำดับความเหมาะสม จากนั้นจะเป็นการแบ่งโซลูชั่นเริ่มต้นออกเป็นกลุ่ม (Memeplex) ทั้งหมด m กลุ่ม โดยโซลูชั่นเริ่มต้นลำดับที่ 1 (มีความเหมาะสมมากที่สุด) จะถูกจัดให้อยู่ $memeplex_1$, โซลูชั่นเริ่มต้นลำดับที่ 2 จะถูกจัดให้อยู่ $memeplex_2$, โซลูชั่นเริ่มต้นลำดับที่ m จะถูกจัดให้อยู่ $memeplex_m$, โซลูชั่นเริ่มต้นลำดับที่ m+1 จะถูกจัดให้อยู่ $memeplex_1$ อีกครั้ง ซึ่งขั้นตอนแบ่งกลุ่มจะดำเนินไปเรื่อยๆ จนโซลูชั่นเริ่มต้นทุกตัวถูกจัดเข้ากลุ่มทั้งหมด ทั้งนี้โซลูชั่นที่มีความเหมาะสมมากที่สุดในแต่ละ memeplex จะถูกกำหนดให้เป็นโซลูชั่น R(U(lb)), โซลูชั่นที่แย่ที่สุดในแต่ละ memeplex จะถูกกำหนดให้เป็นโซลูชั่น R(U(w)) และโซลูชั่นที่มีความเหมาะสมมากที่สุดจากทุก memeplex จะถูกกำหนดให้เป็นโซลูชั่น R(U(gb))

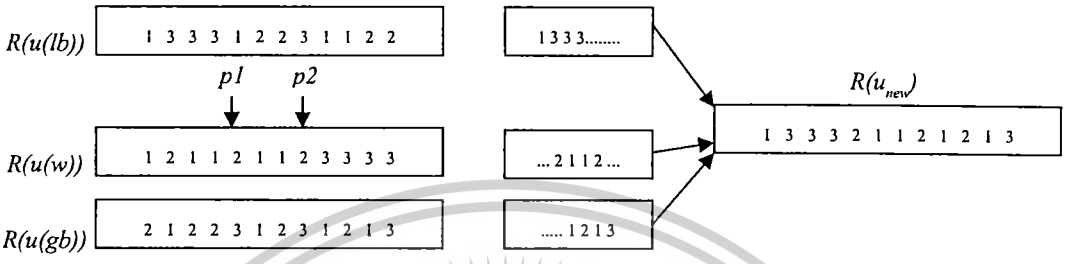
3) การค้นหาโซลูชั่นใหม่ในแต่ละ memeplex (Evolution process in each memeplex)

ในงานวิจัยนี้กำหนดวิธีการสร้างโซลูชั่นใหม่แยกเป็น 2 ส่วน คือ 1) ส่วนเครื่องจักร มีขั้นตอนคือ สุ่มเลือกตำแหน่งในส่วนเครื่องจักรขึ้นมา 2 ตำแหน่งคือ P1 และ P2 จากนั้นจะทำการคัดลอกเครื่องจักรจาก R(U(w)), R(U(LB)) และ R(U(gb)) ไปไว้ยังโซลูชั่นใหม่ ดังแสดงในรูปที่ 2.6(a) 2) ส่วนโอเปอเรชั่น มีขั้นตอนคือ สุ่มเลือกตำแหน่งในส่วนโอเปอเรชั่นขึ้นมา 2 ตำแหน่ง คือ R1 และ R2 จากนั้นจะทำการคัดลอกโอเปอเรชั่นจาก R(U(w)), R(U(LB)) และ R(U(gb)) ไปไว้ยัง

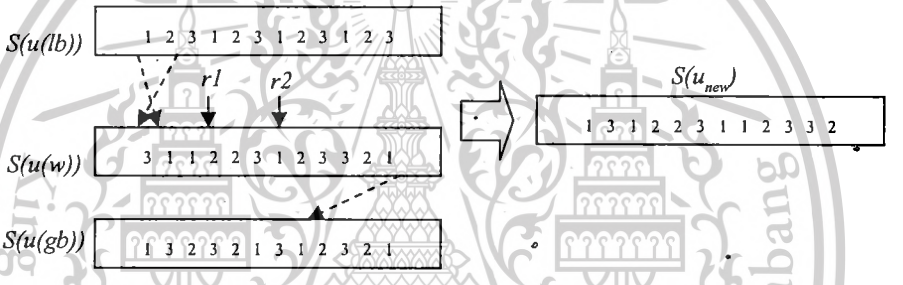
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โซลูชั่นใหม่ ดังแสดงในรูปที่ 2.6(b) ในกรณีที่โซลูชั่นใหม่มีความเหมาะสมมากกว่าโซลูชั่น R(U(w)), โซลูชั่น R(U(w)) จะถูกแทนที่ด้วยโซลูชั่นใหม่ ในทางตรงกันข้ามหากโซลูชั่น R(U(w)) มีความเหมาะสมมากกว่าโซลูชั่นใหม่ งานวิจัยนี้จะค้นหาโซลูชั่นใหม่อีกครั้งโดยใช้วิธี Local Search และโซลูชั่นใหม่ที่ได้จากขั้นตอน Local Search จะถูกนำไปแทนที่โซลูชั่น R(U(w))



(a) การครอสโอเวอร์สำหรับส่วนเครื่องจักร



(b) การครอสโอเวอร์สำหรับส่วนโอเปอเรชั่น

รูปที่ 2.6 วิธีการครอสโอเวอร์ [6]

สำหรับการปรับปรุงโซลูชั่นด้วยวิธี Local Search มีกระบวนการคือ ทำการสุ่มเลือกโซลูชั่น R(U(LB)) หรือ โซลูชั่น R(U(gb)) มาเพียง 1 โซลูชั่น จากนั้นจะทำการค้นหาโซลูชั่นใหม่โดยการปรับปรุงโซลูชั่น R(U(LB)) หรือ โซลูชั่น R(U(gb)) ที่สุ่มเลือกได้ด้วยกระบวนการคือ

3.1) การค้นหาโซลูชั่นใหม่ในส่วนเครื่องจักร โดยการสุ่มเลือกวิธี Local search 1 วิธีจาก 3 วิธี ดังต่อไปนี้

วิธีที่ 1) Random neighborhood มีกระบวนการคือ สุ่มเลือกโอเปอเรชั่น O_{ij} มา 1 ตัว จากนั้นทำการเลือกเครื่องจักรใหม่ที่ไม่ซ้ำกับเครื่องจักรเดิมให้กับโอเปอเรชั่น O_{ij}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

142769

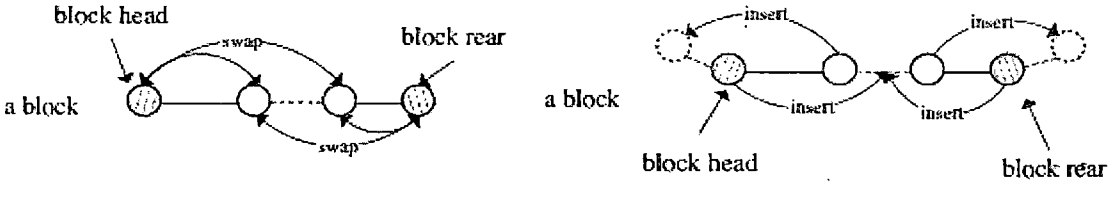
วิธีที่ 2) Top k-q most critical operation neighborhood มีกระบวนการคือ พิจารณาเครื่องจักรที่มีภาระงานมากที่สุด แล้วค้นหาว่ามีโอเปอเรชั่นวิกฤติตัวใดบ้างที่ดำเนินการอยู่กับเครื่องจักรข้างต้น จากนั้นทำการสุ่มเลือกเครื่องจักรตัวใหม่ให้กับโอเปอเรชั่นที่เลือกมาข้างต้น

วิธีที่ 3) Workload considered neighborhood มีกระบวนการคือ 1) สุ่มเลือกโอเปอเรชั่น O_{ij} มา 1 ตัว พร้อมกับสุ่มเลือกเครื่องจักรของโอเปอเรชั่น O_{ij} มาอย่างน้อย 2 ตัว 2) พิจารณาเครื่องจักร 2 ตัวที่สุ่มได้จากขั้นตอนที่ 1 แล้วเลือกเครื่องจักรที่ใช้ระยะเวลาน้อยที่สุด กำหนดให้เป็น M_s 3) พิจารณา workload ของเครื่องจักร M_s รวมกับระยะเวลาของโอเปอเรชั่น O_{ij} บนเครื่องจักร M_s, หาก workload ของเครื่องจักร M_s น้อยกว่า workload ของเครื่องจักรเดิม (M_k) ที่ใช้ ก็ทำการแทนที่เครื่องจักร M_k ด้วยเครื่องจักร M_s

3.2) การค้นหาโซลูชั่นใหม่ในส่วนโอเปอเรชั่น โดยการแบ่งโอเปอเรชั่นวิกฤติไว้เป็นกลุ่มเรียกว่า Critical block จากนั้นสุ่มเลือกวิธี Local search 1 วิธีจาก 2 วิธี ดังต่อไปนี้

วิธีที่ 1) Swap neighborhood มีกระบวนการคือ 1) สุ่มเลือก critical block ที่มีสมาชิกมากกว่า 2 โอเปอเรชั่น 2) สุ่มเลือกโอเปเรชั่น O_k ที่อยู่ตำแหน่งตรงกลาง critical block 3) เลือกโอเปอเรชั่นที่อยู่ในตำแหน่งแรก หรือ ตำแหน่งสุดท้ายของ critical block มา 1 โอเปอเรชั่น แล้วกำหนดให้เป็นโอเปอเรชั่น O_b 4) สลับตำแหน่งโอเปอเรชั่น O_k กับ โอเปอเรชั่น O_b ดังแสดงในรูปที่ 2.7(a)

วิธีที่ 2) Insert neighborhood มีกระบวนการคือ 1) สุ่มเลือก critical block ที่มีสมาชิกมากกว่า 2 โอเปอเรชั่น 2) สุ่มเลือกโอเปอเรชั่น O_k ใน critical block มา 1 ตัว 3) นำโอเปอเรชั่น O_k ไปวางไว้ในตำแหน่งหน้า block head หรือ block rare 4) นำโอเปอเรชั่นในตำแหน่ง block head หรือ block rare ไปวางไว้ที่ตำแหน่งอื่นๆใน critical block ดังแสดงในรูปที่ 2.7(b)



(a) Swap Neighborhood **(b)** Insert Neighborhood

รูปที่ 2.7 กระบวนการ Local search สำหรับส่วนโอเปอเรชั่น [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4) ทำการอัพเดทโซลูชั่นในแต่ละ memeplex

จากนั้นนำโซลูชั่นในแต่ละ memeplex รวมกันเพื่อจัดเรียงโซลูชั่นตามความเหมาะสม พร้อมทั้งอัพเดทโซลูชั่น S(U(gb)) ซึ่งโซลูชั่นที่มีความเหมาะสม m ลำดับแรกจะเป็นตัวแทนโซลูชั่นสำหรับอัลกอริทึมในรอบถัดไป

5) ตรวจสอบเงื่อนไขการหยุดค้นหา หากไม่เป็นไปตามเงื่อนไขให้ทำซ้ำขั้นตอนที่ 2 – 4

งานวิจัยนี้ได้ประเมินประสิทธิภาพของอัลกอริทึม HSFLA ที่นำเสนอกับชุดข้อมูลมาตรฐาน Kacem ซึ่งประกอบด้วย 3 ชุดข้อมูลย่อย จากนั้นนำผลการประเมินไปเปรียบเทียบกับงานวิจัยอื่นๆ ที่ทดสอบกับข้อมูลชุดเดียวกัน ซึ่งผลการเปรียบเทียบแสดงให้เห็นว่าอัลกอริทึม HSFLA มีความเหมาะสมมากกว่าทุกงานวิจัย และเมื่อนำอัลกอริทึม HSFLA เปรียบเทียบกับอัลกอริทึม MOEA-GLS ในชุดข้อมูล Kacem with release dates พบว่าอัลกอริทึม HSFLA ให้ผลลัพธ์ใกล้เคียงกับ อัลกอริทึม MOEA-GLS แต่อัลกอริทึม HSFLA ใช้ระยะเวลาในการประมวลผลน้อยกว่า สุดท้ายคือการประเมินประสิทธิภาพของอัลกอริทึม HSFLA กับชุดข้อมูล BR Data แล้วนำผลการประเมินที่ได้เปรียบเทียบกับงานวิจัยอื่นๆ ซึ่งผลลัพธ์การเปรียบเทียบแสดงให้เห็นว่า อัลกอริทึม HSFLA มีประสิทธิภาพที่ใกล้เคียงกับงานวิจัยอื่นๆ

จากการศึกษางานวิจัยที่นำเสนอนี้ จึงสามารถวิเคราะห์ให้เห็นจุดเด่นที่สำคัญได้คือ HSFLA เป็นอัลกอริทึมที่ไม่มีความซับซ้อน สามารถนำไปประยุกต์ได้ง่ายและใช้เวลาในการประมวลผลน้อย ซึ่งทำให้สามารถนำไปเพื่อใช้แก้ปัญหาได้ง่าย และให้ผลลัพธ์ที่รวดเร็ว แต่ทั้งนี้อัลกอริทึม HSFLA ยังมีจุดด้อยที่สำคัญคือ อัลกอริทึม HSFLA มีขั้นตอนในการทำงานน้อย และไม่ได้ประยุกต์ใช้วิธีการอื่นๆ ที่จะช่วยแก้ข้อด้อยของอัลกอริทึมมาร่วมด้วย จึงส่งผลให้อัลกอริทึมไม่เกิดความหลากหลาย อีกทั้งอัลกอริทึม HSFLA ยังไม่มีขั้นตอนที่จะช่วยแก้ปัญหาการตก Local optimum ได้อย่างชัดเจน ดังนั้นจึงทำให้ผลลัพธ์ของอัลกอริทึม HSFLA ไม่ดีที่สุดเมื่อเปรียบเทียบกับอัลกอริทึมของงานวิจัยอื่นๆ

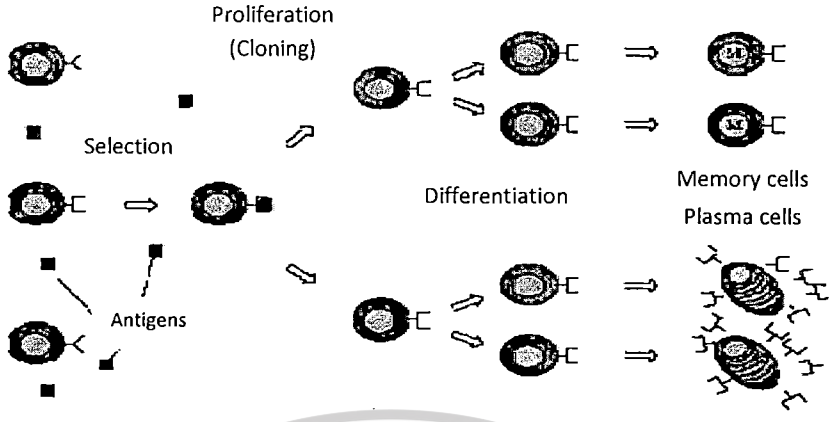
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.6.3 Solving Multi-Objective Flexible Job Shop Scheduling Problem Using Immune Algorithm

Hamid Davoudpour และ Noder Azad [7] ได้ประยุกต์ใช้อัลกอริทึมที่เลียนแบบระบบการทำงานของภูมิคุ้มกันภายในร่างกายสิ่งมีชีวิตที่มีชื่อเรียกว่า Artificial Immune System Algorithm มาใช้เพื่อแก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงานชนิดหลายวัตถุประสงค์ (Multi objective flexible job shop scheduling problem) โดยสามารถอธิบายระบบภูมิคุ้มกันตามหลักการทางชีววิทยาได้ดังนี้ ระบบภูมิคุ้มกันในร่างกายของสิ่งมีชีวิตจะประกอบด้วยเซลล์ 2 ชนิดที่ทำงานสัมพันธ์กันคือ B-Cells และ T-Cells ซึ่งเซลล์ 2 ชนิดนี้มีหน้าที่เดียวกันคือ การจำแนกเซลล์เชื้อโรค (Antigen) ออกจากเซลล์ของร่างกาย ซึ่งในการวิจัยนี้จะกล่าวถึงเฉพาะหน้าที่ของ B-cells ต่อระบบภูมิคุ้มกันของร่างกายเท่านั้น

ลักษณะทางกายภาพของ B-Cells จะมีส่วนที่เรียกว่า Receptor ซึ่งทำหน้าที่ดักจับเซลล์เชื้อโรค และทำการหลั่งสารแอนติบอดี (Antibody) เพื่อกำจัดเซลล์เชื้อโรคชนิดนั้นๆ ให้ออกไปจากร่างกาย ทั้งนี้ B-Cells ที่มีความจำเพาะกับเชื้อโรคจะใช้วิธีการแบ่งตัวเพื่อดักจับเซลล์เชื้อโรค และ B-Cells ที่ได้จากการแบ่งตัวจะหลั่งสาร Antibody ออกไปทำลายเซลล์เชื้อโรคนั้นๆ ดังแสดงในรูปที่ 2.8 ซึ่งวิธีการนี้ถูกเรียกว่า Clonal Selection อัตราการแบ่งตัวของ B-Cells จะมีสัดส่วนโดยตรงกับระดับความเข้ากันได้ของ B-Cells กับเชื้อโรค หมายความว่า Receptor ของ B-Cell ยิ่งสามารถจับกับเชื้อโรคได้ดีมากเท่าใด สัดส่วนการแบ่งตัวของ B-Cells ชนิดนั้นๆ ก็จะเพิ่มมากขึ้น โดย B-Cells ที่ถูกแพร่กระจายออกไปจะต้องผ่านขั้นตอนที่เรียกว่า Maturation (Mutation) ซึ่งเป็นขั้นตอนที่จะช่วยปรับปรุงให้ B-Cells สามารถจับกับเชื้อโรคได้ดียิ่งขึ้น โดยอัตราการมิวเทชั่นจะแปรผกผันกับความเหมาะสมของ B-Cells ซึ่งอธิบายเงื่อนไขข้อนี้ได้ว่า B-Cells ตัวใดที่มี Receptor ซึ่งสามารถจับกับเชื้อโรคได้ดีอยู่แล้ว ก็จะมีอัตราการถูก Mutation น้อย และในทางกลับกัน หาก B-Cells ตัวใดที่มี Receptor ที่ยังจับกับเชื้อโรคได้ไม่ดีเท่าที่ควร ก็จะมีอัตราการถูก Mutation มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 2.8 กระบวนการ Clonal Selection เบื้องต้น [7]

ในงานวิจัยนี้กำหนดให้ Antibody เป็นตัวแทนของโซลูชั่นปัญหา และ Antigen เป็นตัวแทนของฟังก์ชั่นวัตถุประสงค์ โดยขั้นตอนของอัลกอริทึมที่งานวิจัยนี้นำเสนอมีดังต่อไปนี้

1) การสร้าง Antibody เริ่มต้น (Antibody Colonization)

ในงานวิจัยนี้ได้แบ่ง Antibody ออกเป็น 2 ส่วนคือ ส่วน Machine order string และ Operation order string โดยกำหนดวิธีสร้าง Antibody ทั้ง 2 ส่วนไว้ดังต่อไปนี้ สำหรับส่วน Operation order string งานวิจัยนี้จะใช้วิธีสุ่มสลับตำแหน่ง 2 โอเปอเรชั่น เพื่อสร้างเป็น Antibody ที่มีความแตกต่างกัน สำหรับส่วน Machine order string งานวิจัยนี้จะใช้วิธี Bootstrapping Method ซึ่งเป็นวิธีการสุ่มเครื่องจักรออกและใส่กลับคืน อธิบายได้ว่า แม้เครื่องจักร M_k จะถูกเลือกให้ดำเนินการกับโอเปอเรชั่น O_{ij} ในช่วงเวลา T ไปแล้ว แต่เครื่องจักร M_k ก็ยังสามารถถูกเลือก (รูปแบบการใส่คืน) ให้ดำเนินการกับโอเปอเรชั่น O_{ij+1} ในช่วงเวลา T+1 ได้อีก เป็นต้น ทั้งนี้โอเปอเรชั่นแต่ละตัวจะถูก assign ไปยังเครื่องจักรที่มีระยะเวลาการดำเนินงานน้อยที่สุดเสมอ

2) การคัดเลือก Antibody ที่มีความเหมาะสม

Antibody ซึ่งมีความเหมาะสมมากที่สุด n ลำดับแรกจากขั้นตอนที่ 1 จะถูกเลือกให้เป็นตัวแทนของ Antibody ทั้งหมด และถูกกำหนดให้เป็น Parent Antibodies จากนั้น Parent Antibodies ทั้ง n ตัวจะถูก assign ให้เป็น Antibody เริ่มต้นของกลุ่ม Pool จำนวน n กลุ่ม แสดงในรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3) การแบ่งตัวของ Parent Antibody (Antibody Colonization)

ขั้นตอนนี้เป็นกระบวนการกำหนดจำนวนการแบ่งตัว (โคลน) ของ Parent Antibody แต่ละตัว โดย Parent Antibody ที่มีความเหมาะสมมากก็จะมีอัตราการโคลนที่สูงกว่า ซึ่งจำนวนการโคลนของ Parent Antibody แต่ละตัวสามารถคำนวณได้จากสมการที่ 2.6 (เมื่อ w_i แทนค่าน้ำหนักของฟังก์ชั่นวัตถุประสงค์ที่ i และ $f_i(\cdot)$ คือ ค่าความเหมาะสมของ Antibody ใดๆ บนฟังก์ชั่นวัตถุประสงค์ที่ i)

$$\text{clone rate} = j + \prod_{i=1}^{3} w_i[f_i(\text{worst antibody}) - f_i(\text{current antibody})] \qquad (2.6)$$

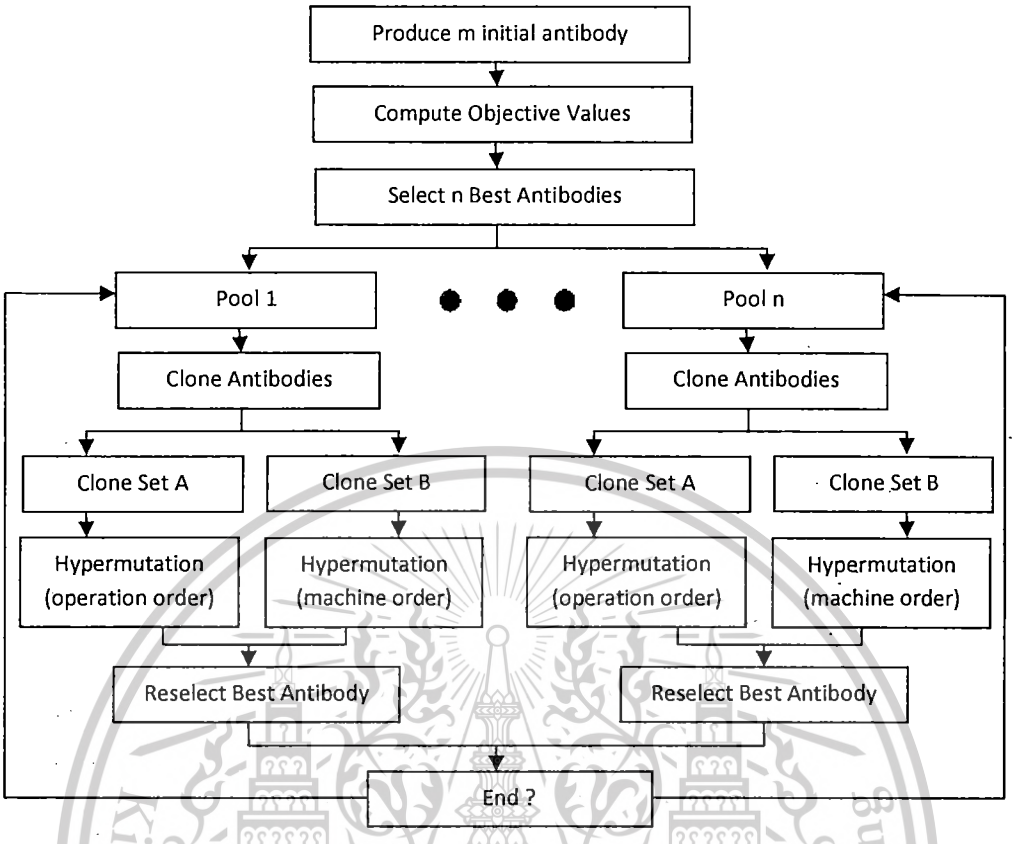
เนื่องจาก Antibody ที่มีความเหมาะสมมากจะมีอัตราการแบ่งตัวสูง ในทางตรงกันข้ามอัตราการ improvement จะต่ำ ดังนั้นเพื่อเพิ่มโอกาสให้ Parent Antibodies ได้รับการ improvement มากขึ้น งานวิจัยนี้จึงกำหนดให้มีการเพิ่มอัตราการแบ่งตัวให้กับ Parent Antibodies แต่ละตัวอีก 1 หน่วย ด้วยเหตุผลที่ว่า เมื่อมี Antibody จำนวนมากขึ้น โอกาสที่จะทำให้เกิดการปรับปรุงก็เพิ่มขึ้นด้วย ทั้งนี้ Antibody ที่ได้จากขั้นตอนการแบ่งตัว จะถูกกำหนดให้เป็น Cloned Antibodies

4) การมิวเทชั่น (Antibody Mutation)

ขั้นตอนนี้เป็นการปรับปรุง Cloned Antibodies ในแต่ละกลุ่ม pool ด้วยกระบวนการมิวเทชั่น โดยจะแบ่งวิธีการมิวเทชั่นออกเป็น 2 ส่วนคือ วิธีมิวเทชั่นสำหรับส่วนเครื่องจักร และวิธีการมิวเทชั่นสำหรับส่วนโอเปอเรชั่น ซึ่ง Cloned Antibody แต่ละตัวจะมีอัตราการมิวเทชั่น (mutation rate) ไม่เท่ากัน ขึ้นอยู่กับค่าความเหมาะสม โดยอัตรามิวเทชั่นของ Cloned Antibody แต่ละตัวสามารถคำนวณได้จากสมการที่ 2.7 ทั้งนี้ในงานวิจัยได้กำหนดให้มีการเพิ่มอัตราการมิวเทชั่นให้กับ Cloned Antibody แต่ละตัวขึ้นอีก 1 หน่วย เพื่อเพิ่มโอกาสในการเจอ Antibody ที่มีความเหมาะสมมากขึ้น ซึ่งสามารถอธิบายกระบวนการ Mutation ได้ต่อไปนี้

$$\text{mutation rate} = j + \prod_{i=1}^{3} w_i[f_i(\text{current antibody}) - f_i(\text{best antibody})] \qquad (2.7)$$

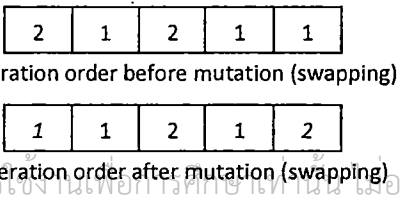
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 2.9 โครงสร้างของอัลกอริทึมที่นำเสนอ [7]

4.1) กระบวนการมิวเทชั่นสำหรับส่วนเครื่องจักร มีวิธีการคือ ทำการสุ่มเลือกโอเปอเรชั่น O_{ij} มา 1 ตัว จากนั้นทำการสุ่มเลือกเครื่องจักรใหม่ให้กับโอเปอเรชั่น O_{ij} ทั้งนี้จำนวนโอเปอเรชั่นที่จะถูกสุ่มเลือกขึ้นมา จะแปรผันตรงกับอัตราการมิวเทชั่นในสมการที่ 2.7

4.2) กระบวนการมิวเทชั่นสำหรับส่วนโอเปอเรชั่น มีวิธีการคือ ทำการสุ่มเลือกโอเปอเรชั่น O_{ij} มา 1 คู่ จากนั้นทำการสลับตำแหน่งของทั้ง 2 โอเปอเรชั่นข้างต้นนี้ ดังแสดงในรูปที่ 2.10 ทั้งนี้จำนวนโอเปอเรชั่นที่จะถูกสลับตำแหน่งกัน จะแปรผันตรงกับอัตราการมิวเทชั่นในสมการที่ 2.7 ซึ่งหมายความว่า อัตราการมิวเทชั่นยิ่งสูง จำนวนโอเปอเรชั่นที่จะถูกสลับตำแหน่งยิ่งมากตาม



รูปที่ 2.10 กระบวนการ Mutation สำหรับส่วนโอเปอเรชั่น [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

หลังจากผ่านขั้นตอนการมิวเทชั่น งานวิจัยนี้จะทำการจัดเรียง Antibody ในแต่ละ Pool ตามลำดับความเหมาะสม โดย Antibody ที่มีความเหมาะสมมากที่สุด N ลำดับแรก จะถูกกำหนดให้เป็น Parent Antibody เพื่อใช้ดำเนินการในรอบถัดไป

5) ตรวจสอบเงื่อนไขการหยุดค้นหา หากไม่เป็นไปตามเงื่อนไขการหยุดค้นหาให้ทำซ้ำขั้นตอนที่ 3-4

ในงานวิจัยนี้ได้ทำการทดสอบประสิทธิภาพอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐาน 3 ชุดข้อมูลคือ ชุดข้อมูล 8×8, ชุดข้อมูล 10×10 และชุดข้อมูล 15×10 โดยแต่ละชุดข้อมูลจะประกอบด้วยจำนวน job และจำนวน operation ที่ต่างกัน ซึ่งผลการทดสอบที่ได้จะถูกนำไปเปรียบเทียบกับอัลกอริทึมของงานวิจัยอื่นๆ ที่ทดสอบกับข้อมูลชุดเดียวกัน และผลการเปรียบเทียบแสดงให้เห็นว่า อัลกอริทึมที่งานวิจัยนี้นำเสนอให้ผลลัพธ์ที่ดีกว่า และใช้เวลาในการประมวลผลที่น้อยกว่าเช่นกัน

จากการศึกษางานวิจัยที่นำเสนอนี้ สามารถวิเคราะห์ให้เห็นจุดเด่นที่สำคัญคือ ขั้นตอนของงานวิจัยนี้ไม่มีความซับซ้อน ส่งผลให้ใช้เวลาในการประมวลผลค่อนข้างน้อยและยังให้ผลลัพธ์ที่มีประสิทธิภาพด้วย อีกทั้งงานวิจัยนี้ยังได้วิเคราะห์การทำงานของอัลกอริทึม จึงช่วยเพิ่มโอกาสให้อัลกอริทึมได้พบเจอคำตอบที่มีความเหมาะสมมากขึ้นในขั้นตอนการโคลนและการมิวเทชั่น แต่อย่างไรก็ตาม อัลกอริทึมที่งานวิจัยนี้นำเสนอยังมีจุดด้อยที่สำคัญคือ กระบวนการค้นหาคำตอบของอัลกอริทึมนี้มีเพียงขั้นตอนการมิวเทชั่นเพียงขั้นตอนเดียว ซึ่งทำให้กระบวนการค้นหาไม่เกิดความหลากหลาย และเนื่องจากงานวิจัยนี้ทดสอบกับชุดข้อมูลที่มีขนาดเล็ก จึงทำให้ผลการเปรียบเทียบยังไม่ชัดเจนมากนัก

2.6.4 Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem

Xinyu Shao, Weiqi Liu, Qiong Liu และ Chaoyong Zhang [8] ได้แก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงานชนิดหลายวัตถุประสงค์ (Multi-objective flexible job shop scheduling problem) โดยนำเสนออัลกอริทึมที่ผสมผสานหลักการของอัลกอริทึมพาติเคิลสวอม (Particle swarm optimization: PSO) และอัลกอริทึมการอบอ่อนจำลอง (Simulated Annealing Algorithm: SA) โดยอัลกอริทึม PSO ถูกนำมาประยุกต์ใช้เพื่อค้นหาคำตอบในพื้นที่ global และอัลกอริทึม SA ถูกนำมาประยุกต์ใช้เพื่อค้นหาคำตอบในพื้นที่ local ทั้งนี้ PSO เป็นอัลกอริทึมในกลุ่มของการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

เลียนแบบพฤติกรรมการอยู่ร่วมกันเป็นสังคมของสัตว์ (Social Behavior) โดย PSO เลียนแบบพฤติกรรมการเคลื่อนที่ของฝูงนก ซึ่งฝูงนกจะมีรูปแบบการบินเคลื่อนที่สอดคล้องกันในเวลาออกหาอาหาร โดยนกจะมีการสื่อสารกันในฝูง ข้อมูลที่สื่อสารกันจะเป็นข้อมูลแหล่งอาหารที่เหมาะสม ซึ่งนกแต่ละตัวไปค้นพบมา (Pbest) และข้อมูลแหล่งอาหารที่มีความเหมาะสมมากที่สุดจากข้อมูลของนกทุกตัว (Gbest) ทั้งนี้นกทุกตัวในฝูงจะค่อยๆ เคลื่อนที่ไปยังแหล่งอาหาร Gbest ส่งผลให้รูปแบบการเคลื่อนที่ของนกทั้งฝูงสอดคล้องไปในทิศทางเดียวกัน ส่วน SA เป็นอัลกอริทึมที่อยู่ในกลุ่มของเมตาฮิวริสติก ซึ่งถูกออกแบบมาเพื่อช่วยให้การค้นหาคำตอบหลุดพ้นจากพื้นที่ local โดย SA มีวิธีการเลียนแบบขั้นตอนการปรับลดอุณหภูมิในการหลอมละลายเหล็ก เพื่อให้ได้เหล็กที่มีความเหมาะสมและทนทาน โดยการลดอุณหภูมิแต่ละครั้งจะทำให้ได้เหล็กที่มีสถานะความเหมาะสมที่ต่างกัน เหล็กที่ได้หลังจากขั้นตอนการปรับลดอุณหภูมิจะถูกยอมรับเมื่อมีความเหมาะสมมากกว่าสถานะเหล็กก่อนปรับลดอุณหภูมิ โดยเหล็กที่ได้หลังจากขั้นตอนการปรับลดอุณหภูมิที่มีความเหมาะสมต่ำกว่าสถานะเหล็กก่อนปรับลดอุณหภูมิจะได้รับการยอมรับตามเงื่อนไขของสมการที่ 2.9

$$\text{Probability of acceptance } (P_b) > \varepsilon \tag{2.9}$$

$$\text{Probability of acceptance } (P_b) = \exp(-\Delta/T),\ \Delta = f(z^*) - f(z_n) \tag{2.10}$$

ในงานวิจัยนี้ได้ใช้อัลกอริทึม PSO เป็นโครงสร้างหลัก และประยุกต์ใช้อัลกอริทึม SA ในขั้นตอนการค้นหาคำตอบในพื้นที่ local โดยสามารถอธิบายอัลกอริทึมที่นำเสนอได้ดังนี้

1) ขั้นตอนการสร้างโซลูชั่นเริ่มต้น (Initialization)

การกำหนดโซลูชั่นเริ่มต้นจะถูกแยกพิจารณาเป็น 2 ส่วนคือ 1) ส่วนเครื่องจักร ในงานวิจัยนี้กำหนดวิธีสร้างส่วนเครื่องจักรของโซลูชั่นเริ่มต้นจากกฎการเลือกเครื่องจักรที่มีระยะเวลาการดำเนินงานน้อยที่สุด (Minimum processing time) และกฎการเลือกเครื่องจักรที่มีภาระงานน้อยที่สุด (Minimum workload) 2) ส่วนโอเปอเรชั่น ในงานวิจัยนี้กำหนดวิธีสร้างส่วนโอเปอเรชั่นของโซลูชั่นเริ่มต้นจาก 3 กฎคือ กฎการสุ่ม (Random rule), กฎการเลือกโอเปอเรชั่นของ job ที่มีระยะเวลาคงเหลือรวมมากที่สุด (Most working remaining rule), และกฎการเลือกโอเปอเรชั่นของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

job ที่มีโอเปอเรชั่นคงเหลือรวมมากที่สุด (Most number of operation remaining rule) ทั้งนี้โซลูชั่นเริ่มต้นของงานวิจัยนี้ถูกกำหนดให้เป็น Z_n เมื่อ n คือจำนวนโซลูชั่นเริ่มต้น

2) ขั้นตอนการคำนวณหาค่าความเหมาะสม (Fitness evaluation)

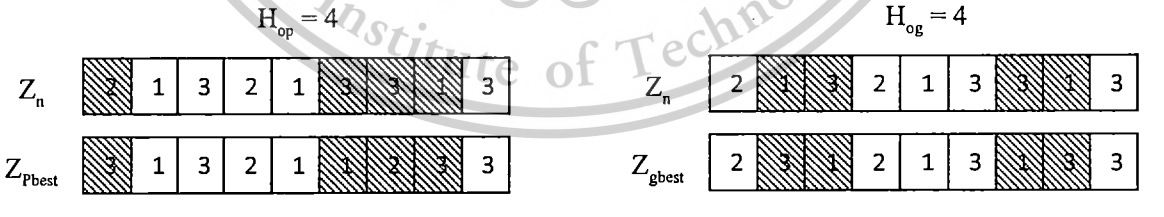
โซลูชั่นที่มีความเหมาะสม (Non-dominated solution) จะถูกบันทึกให้เป็นโซลูชั่น Pbest โดยโซลูชั่น Pbest แต่ละตัวจะถูกบันทึกไว้ใน Archive set และโซลูชั่น Gbest คือโซลูชั่นใดๆ ที่ถูกสุ่มเลือกมาจาก Archive set

3) ขั้นตอนการปรับตำแหน่งของโซลูชั่นเริ่มต้น (Displacement of particles)

3.1) การปรับตำแหน่งส่วนโอเปอเรชั่นของโซลูชั่น Z_n

สามารถอธิบายได้คือ เริ่มต้นที่การคำนวณหาค่าความเหมือน (S_{op}) ระหว่างโซลูชั่น Z_n กับโซลูชั่น Z_{pbest} และคำนวณหาค่าความเหมือน (S_{og}) ระหว่างโซลูชั่น Z_n กับโซลูชั่น Z_{gbest} จากสมการ $S_{op} = H_{op}/I$ และ $S_{og} = H_{og}/I$ ตามลำดับ ซึ่งตัวแปร H_{op} และ H_{og} แทนค่า Hamming distance ส่วนตัวแปร I แทนจำนวนโอเปอเรชั่นทั้งหมด ซึ่งจากตัวอย่างในรูปที่ 2.11(a) และ 2.11(b) แสดงให้เห็นว่าค่า $H_{op} = 4$, $H_{og} = 4$, $I = 9$, $S_{op} = 0.44$, $S_{og} = 0.44$

ทำการปรับตำแหน่งส่วนโอเปอเรชั่นของโซลูชั่น Z_n จากซ้ายไปขวาทีละตำแหน่ง โดยการสุ่มค่าที่อยู่ในช่วง [0,1] มา 1 ค่า หากค่าที่สุ่มได้น้อยกว่าหรือเท่ากับค่า S_0 ข้อมูลตำแหน่งที่



(a) Hamming distance ระหว่างโซลูชั่น Z_n และโซลูชั่น Z_{Pbest}

(b) Hamming distance ระหว่างโซลูชั่น Z_n และโซลูชั่น Z_{gbest}

รูปที่ 2.11 วิธีการหา Hamming distance ส่วนโอเปอเรชั่น [8]

พิจารณาอยู่จะถูกลบทิ้ง ในทางตรงกันข้ามหากค่าที่สุ่มได้มากกว่าค่า S_0 ข้อมูลตำแหน่งที่พิจารณาอยู่จะถูกสืบทอดไปยังโซลูชั่น Z_{n+1} ทั้งนี้โซลูชั่น Z_{pbest} และโซลูชั่น Z_{gbest} ก็จะถูกพิจารณาจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ด้านซ้ายไปขวาเช่นเดียวกัน โดยข้อมูลในโซลูชั่น Z_{pbest} และโซลูชั่น Z_{gbest} ที่มีความสัมพันธ์กับข้อมูลในโซลูชั่น Z_{n+1} จะถูกลบทิ้งไป แล้วคัดลอกข้อมูลที่ยังเหลือในโซลูชั่น Z_{pbest} และโซลูชั่น Z_{gbest} ไปไว้ยังโซลูชั่น Z_{n+1} โดยการพิจารณาจากซ้ายไปขวา

3.2) การปรับตำแหน่งส่วนเครื่องจักรของโซลูชั่น Z_n

สามารถอธิบายได้คือ ทำการคำนวณหาค่าความเหมือน (S_{mp}) ระหว่างโซลูชั่น Z_n กับโซลูชั่น Z_{pbest} และทำการคำนวณหาค่าความเหมือน (S_{mg}) ระหว่างโซลูชั่น Z_n กับโซลูชั่น Z_{gbest} จากสมการ $S_{mp} = H_{mp}/I$ และ $S_{mg} = H_{mg}/I$ ซึ่งตัวแปร H_{mp} และ H_{mg} แทนค่า Hamming distance ส่วนตัวแปร I แทนจำนวนโอเปอเรชั่นทั้งหมด ซึ่งจากตัวอย่างในรูปที่ 2.12(a) และ 2.12(b) แสดงให้เห็นว่าค่า $H_{mp} = 6$, $H_{mg} = 4$, $I = 9$, $S_{mp} = 0.66$, $S_{mg} = 0.55$

(a) Hamming distance ระหว่างโซลูชั่น Z_n และโซลูชั่น Z_{Pbest}

(b) Hamming distance ระหว่างโซลูชั่น Z_n และโซลูชั่น Z_{gbest}

รูปที่ 2.12 วิธีการหา Hamming distance ส่วนเครื่องจักร [8]

จากนั้นทำการสุ่มลบข้อมูลส่วนเครื่องจักรของโซลูชั่น Z_n ตามค่า S_m พร้อมกับคัดลอกข้อมูลในตำแหน่งที่ไม่ถูกลบไปยังโซลูชั่น Z_{n+1} แล้วสุ่มเลือกข้อมูลจากโซลูชั่น Z_{Pbest} หรือ Z_{gbest} ไปไว้ในตำแหน่งว่างของโซลูชั่น Z_{n+1}

เมื่อสร้างส่วนโอเปอเรชั่นและส่วนเครื่องจักรของโซลูชั่น Z_{n+1} ตามวิธีการข้างต้นเสร็จสิ้น จะทำการเปรียบเทียบความเหมาะสมของโซลูชั่น Z_{pbest} และโซลูชั่น Z_{n+1} โดยโซลูชั่น Z_{pbest} จะถูกแทนที่ด้วยโซลูชั่น Z_{n+1} ในกรณีที่โซลูชั่น Z_{n+1} มีความเหมาะสมมากกว่า

4) ขั้นตอนการค้นหาโซลูชั่นใหม่ในพื้นที่ Local ด้วยอัลกอริทึม SA

ขั้นตอนนี้จะเป็นการค้นหาโซลูชั่นใหม่จากโซลูชั่น Z_n ด้วยกระบวนการของอัลกอริทึม SA โดยหลักการของอัลกอริทึม SA จะเป็นการ Move โซลูชั่น Z_n ไปยังตำแหน่งต่างๆ ซึ่งการ Move แต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ละครั้งจะทำให้ได้โซลูชั่นใหม่ (Z*) เสมอ จากนั้นทำการอัพเดท Archive set ด้วยโซลูชั่น Z* ทั้งนี้ โซลูชั่น Z* จะได้รับการยอมรับให้แทนที่โซลูชั่น Z_n ใน 2 กรณีคือ

กรณีที่ 1: โซลูชั่น Z* มีความเหมาะสมมากกว่าโซลูชั่น Z_n กรณีนี้โซลูชั่น Z_n จะถูกแทนที่ด้วยโซลูชั่น Z*

กรณีที่ 2: โซลูชั่น Z* มีความเหมาะสมน้อยกว่าโซลูชั่น Z_n กรณีนี้โซลูชั่น Z_n จะถูกแทนที่ด้วยโซลูชั่น Z* ตามเงื่อนไขของสมการที่ 2.9

จากนั้นจะเป็นขั้นตอนการปรับอุณหภูมิ (T) ให้ลดลงตามสมการ $T = T*K_b$ หลังจากนั้นกระบวนการค้นหาด้วยอัลกอริทึม SA จะดำเนินไปจนกระทั่งค่าอุณหภูมิ (T) ลดลงจนถึงจุดที่กำหนดไว้

5) ตรวจสอบเงื่อนไขการหยุดค้นหา หากไม่เป็นไปตามเงื่อนไขจะทำซ้ำในขั้นตอน 2 ถึง 4 โดยโซลูชั่น Z_n ที่ได้จากขั้นตอนที่ 4 จะถูกใช้ในรอบถัดไป

งานวิจัยนี้ได้ทดสอบประสิทธิภาพของอัลกอริทึม DPSO กับชุดข้อมูลมาตรฐาน 2 ชุดข้อมูลคือ ชุดข้อมูลที่ 1 คือ Kacem data set ซึ่งประกอบด้วยชุดข้อมูลย่อย 3 ชุดคือ Problem 8×8, Problem 10×10 และ Problem 15×10 และชุดข้อมูล 2 คือ BR data set ซึ่งประกอบด้วยชุดข้อมูลย่อย 10 ชุดคือ MK01, MK02, ..., MK10 จากนั้นนำผลการทดสอบที่ได้ไปเปรียบเทียบกับอัลกอริทึมในงานวิจัยอื่นๆ พบว่า อัลกอริทึม DPSO ที่งานวิจัยนี้นำเสนอให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมของงานวิจัยอื่นๆ ในข้อมูลชุดที่ 1 อีกทั้งยังใช้ระยะเวลาในการประมวลผลที่น้อยกว่า ส่วนในการเปรียบเทียบกับชุดข้อมูลที่ 2 อัลกอริทึม DPSO ให้ผลลัพธ์ที่ใกล้เคียงกับงานวิจัยอื่นๆ แต่มีโซลูชั่นคำตอบที่เหมาะสมในจำนวนที่มากกว่า

เมื่อทำการศึกษาอัลกอริทึม DPSO จึงสามารถวิเคราะห์ให้เห็นจุดเด่นได้คือ งานวิจัยนี้พยายามแก้ปัญหาของอัลกอริทึม PSO ที่ไม่สามารถลู่เข้าพบคำตอบที่มีความเหมาะสมได้ โดยการประยุกต์ใช้อัลกอริทึม SA โดย SA จะช่วยแก้ปัญหาในกรณีที่คำตอบตกอยู่ในพื้นที่ local optimum และในกระบวนการ displacment ของ PSO เองยังช่วยทำให้การค้นหาเกิดความหลากหลาย แต่ทั้งนี้อัลกอริทึม DPSO ยังมีข้อด้อยที่สำคัญคือ ในขั้นตอน displacment จะต้องทำการคำนวณหาค่าความ

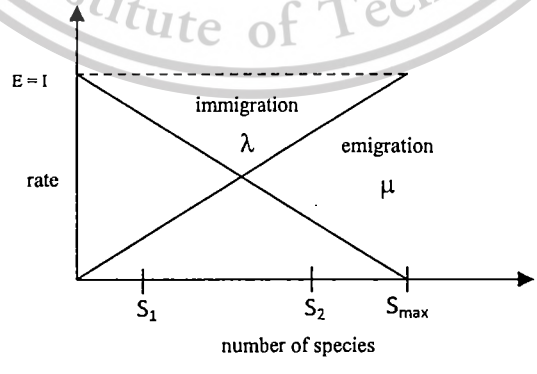
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

เหมือนของโซลูชั่น Z_n ทุกตัวกับโซลูชั่น Z_{pbest} และโซลูชั่น Z_{gbest} อีกทั้งยังต้องทำกระบวนการนี้ในทุกรอบของอัลกอริทึม ซึ่งอาจจะส่งผลให้อัลกอริทึมใช้เวลาในการประมวลผลนาน

2.6.5 A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem

Sayed Habib, A. Rahmati และ M. Zandieh [9] ได้ประยุกต์ใช้อัลกอริทึมชื่อ biogeography based optimization (BBO) algorithm มาแก้ปัญหา Flexible job shop scheduling โดย BBO เป็นอัลกอริทึมที่เลียนแบบหลักการอพยพย้ายถิ่นที่อยู่ของสิ่งมีชีวิต ซึ่งสิ่งมีชีวิตจะอพยพย้ายไปในถิ่นที่อยู่ที่เหมาะสมมากกว่า อธิบายจากรูปที่ 2.13 ได้ว่าเมื่อจำนวนของสิ่งมีชีวิตเพิ่มขึ้น ถิ่นที่อยู่จะเริ่มหนาแน่น มีเพียงสิ่งมีชีวิตจำนวนเล็กน้อยเท่านั้นที่อพยพเข้าไปแล้วสามารถดำรงชีวิตอยู่ได้ ดังนั้นอัตราการอพยพเข้า (Immigration rate: λ) จะเริ่มลดลง ซึ่งแต่ละถิ่นที่อยู่จะรองรับสิ่งมีชีวิตได้จำกัดที่ S_{max} และเช่นเดียวกันเมื่อจำนวนของสิ่งมีชีวิตเพิ่มขึ้น ถิ่นที่อยู่จะเริ่มหนาแน่น สิ่งมีชีวิตจำนวนมากจะเริ่มย้ายไปถิ่นที่อยู่ใหม่ที่สามารถดำรงชีวิตอยู่ได้ ดังนั้นอัตราการอพยพออก (Emigration rate: μ) จะเพิ่มขึ้น

ในอัลกอริทึม BBO จะออกแบบให้ถิ่นที่อยู่เป็นตัวแทนของโซลูชั่นปัญหา และดัชนีความเหมาะสมในการเป็นที่อยู่อาศัย (Habitat suitability index (HSI)) จะแสดงถึงค่าความเหมาะสมของโซลูชั่นปัญหา จากรูปที่ 2.13 S_1 สัมพันธ์กับโซลูชั่นที่มีความเหมาะสมน้อย และ S_2 สัมพันธ์กับโซลูชั่นที่มีความเหมาะสมมาก



รูปที่ 2.13 ความสัมพันธ์ของอัตราการอพยพเข้าและอพยพออกเมื่อมีจำนวนสิ่งมีชีวิตแตกต่างกัน [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ดังนั้นจึงอธิบายได้ว่า โซลูชั่นที่มีความเหมาะสมน้อย (สิ่งมีชีวิตอาศัยอยู่น้อย) จะมีอัตราการอพยพเข้าสูงและอัตราการอพยพออกต่ำ ในทางตรงกันข้ามโซลูชั่นที่มีความเหมาะสมมาก (สิ่งมีชีวิตอาศัยอยู่หนาแน่น) จะมีอัตราการอพยพเข้าต่ำและอัตราการอพยพออกสูง ทั้งนี้วัตถุประสงค์ของการประยุกต์ใช้อัลกอริทึม BBO คือ การถ่ายทอดคุณลักษณะของโซลูชั่นที่มีความเหมาะสมมากไปยังโซลูชั่นที่มีความเหมาะสมน้อยกว่า เพื่อปรับปรุงให้โซลูชั่นที่มีความเหมาะสมน้อยมีความเหมาะสมเพิ่มมากขึ้น ซึ่งอัลกอริทึม BBO ที่งานวิจัยนี้นำเสนอมีขั้นตอนดังต่อไปนี้

1) การสร้างโซลูชั่นเริ่มต้น (Initialization)

ในงานวิจัยนี้กำหนดวิธีการสร้างโซลูชั่นเริ่มต้นโดยแบ่งเป็น 2 ส่วนคือ 1) ส่วนโอเปอเรชั่นจะถูกสร้างด้วยวิธีการสุ่มจัดลำดับ และ 2) ส่วนเครื่องจักรจะพิจารณากับโอเปอเรชั่น O_{ij} ทีละตัว โดยการสุ่มเครื่องจักรที่เป็นไปได้สำหรับ โอเปอเรชั่น O_{ij} ขึ้นมา 2 ตัว และเครื่องจักรตัวที่ใช้เวลาในการดำเนินงานน้อยกว่าจะถูกเลือก

ซึ่งกระบวนการนี้จะถูกทำซ้ำจนได้โซลูชั่นเริ่มต้นตามจำนวนที่กำหนดไว้ จากนั้นจัดเรียงโซลูชั่นเริ่มต้นตามลำดับความเหมาะสมจากน้อยไปมาก

2) การปรับปรุงโซลูชั่นด้วยขั้นตอน migration (Migration operator)

ขั้นตอนนี้จะเป็นการปรับปรุงโซลูชั่นเริ่มต้น (H) ด้วยกระบวนการแลกเปลี่ยนข้อมูลกับโซลูชั่นอื่นๆ ตามหลักการของการอพยพย้ายถิ่นของสิ่งมีชีวิต โดยโซลูชั่น H_i ใดๆ จะถูกกำหนดให้เป็นโซลูชั่น immigration ได้นั้นจะต้องพิจารณาค่า λ_i ตามสมการที่ 2.15 และโซลูชั่น H_i ใดๆ จะถูกกำหนดให้เป็นโซลูชั่น emigration ได้นั้นจะต้องพิจารณาค่า μ_i ตามสมการที่ 2.16 โดย k_i คือลำดับความเหมาะสมของโซลูชั่น H_i, n แทนจำนวนของโซลูชั่น H ทั้งหมด, E และ I มีค่าเท่ากับ 1

$$\lambda_i = I(1-\frac{k_i}{n}) \tag{2.15}$$

$$\mu_i = E(\frac{k_i}{n}) \tag{2.16}$$

ซึ่งวิธีการเลือกโซลูชั่น immigration และโซลูชั่น emigration มีกระบวนการดังต่อไปนี้

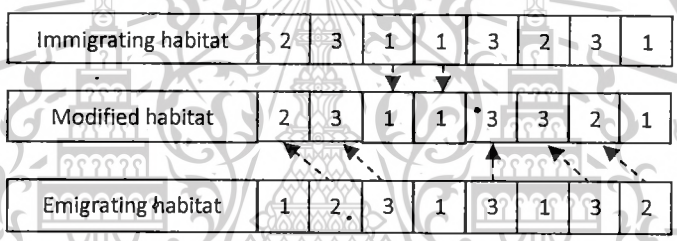
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.1) พิจารณาว่าโซลูชั่น H_i ควรที่จะถูกกำหนดให้เป็นโซลูชั่นสำหรับการถูกอพยพเข้า (immigration) หรือไม่

โดยการสุ่มค่าที่อยู่ในช่วง [0,1] ขึ้นมา 1 ค่า จากนั้นทำการเปรียบเทียบค่าที่สุ่มได้กับค่า λ_i ซึ่งโซลูชั่น H_i จะถูกเลือกให้เป็นโซลูชั่น immigration ในกรณีค่าที่สุ่มได้น้อยกว่า λ_i

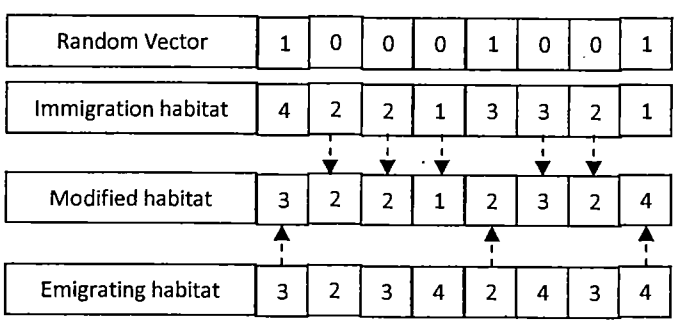
2.2) พิจารณาเลือกโซลูชั่นสำหรับการถูกอพยพออก (emigration)

โดยมีขั้นตอนดังนี้ 1) ทำการสุ่มเลือกจากโซลูชั่นเริ่มต้นด้วยวิธี Rolette wheel selection แล้วกำหนดให้เป็นโซลูชั่น n 2) สุ่มค่าที่อยู่ในช่วง [0,1] ขึ้นมา 1 ค่า แล้วเปรียบเทียบค่าที่สุ่มได้กับค่า μ ของโซลูชั่น n 3) หากค่าที่สุ่มได้น้อยกว่าค่า μ โซลูชั่น n จะถูกกำหนดให้เป็นโซลูชั่นสำหรับการถูกอพยพออก (emigration) หรือ โซลูชั่น H_j



รูปที่ 2.14 วิธีการ Migration ของส่วนโอเปอเรชั่น [9]

หลังจากเลือกโซลูชั่น immigration และโซลูชั่น emigration ได้แล้ว ในขั้นตอนถัดไปจะเป็นการปรับปรุงโซลูชั่น immigration ด้วยการถ่ายทอด SIV (คล้ายยีนส์ใน GA) ของโซลูชั่น emigration ไปยังโซลูชั่น immigration สำหรับทั้งส่วนโอเปอเรชั่นและเครื่องจักรดังแสดงในรูปที่ 2.14 และ 2.15 ตามลำดับ

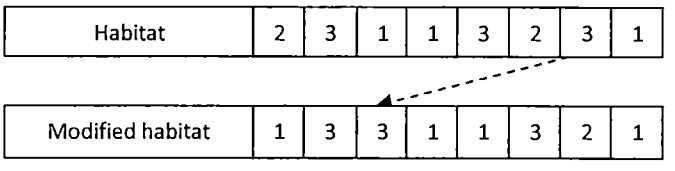


รูปที่ 2.15 วิธีการ Migration ของส่วนเครื่องจักร [9]

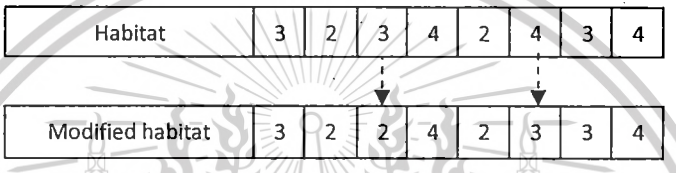
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3) การปรับปรุงโซลูชั่นด้วยขั้นตอน Mutation (Mutation Operator)

ในงานวิจัยนี้จะทำการปรับโซลูชั่น H_i ด้วยกระบวนการมิวเทชั่นตามกระบวนการที่แสดงในรูปที่ 2.16 และ 2.17 สำหรับส่วนโอเปอเรชั่นและเครื่องจักร ตามลำดับ



รูปที่ 2.16 วิธีการมิวเทชั่นของส่วนโอเปอเรชั่น [9]



รูปที่ 2.17 วิธีการมิวเทชั่นของส่วนเครื่องจักร [9]

โดยในการพิจารณาว่าจะมิวเทชั่นโซลูชั่น H_i หรือไม่นั้นจะพิจารณาที่ค่าความน่าจะเป็นในการมิวเทชั่น (Mutation probability: P_m) ซึ่งคำนวณได้จากสมการที่ 2.17 ค่า P_m จะถูกกำหนดด้วย Species count probabilities (P_s) โดยค่า P_s จะแสดงถึงจุดสมดุลของประชากรในแต่ละถิ่นที่อยู่ อธิบายได้ว่าโซลูชั่นที่มีค่า P_s ต่ำจะมีโอกาสถูกมิวเทชั่นมาก และในทางตรงกันข้ามโซลูชั่นที่มีค่า P_s สูงจะมีโอกาสถูกมิวเทชั่นน้อยตามสมการที่ 2.18

$$m_i = m_{\max}\left(1 - \frac{P_i}{p_{\max}}\right) \tag{2.17}$$

$$P_i = \frac{v_i}{\sum_{i=1}^{n} v_i} \tag{2.18}$$

$$v_i = \begin{cases} \dfrac{n!}{(n-1-i)!(i-1)!} & (i = 1,\ldots,i') \\ v_{n+2-i} & (i = i'+1,\ldots,n+1) \end{cases} \tag{2.19}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4) ตรวจสอบเงื่อนไขการหยุดค้นหา หากไม่เป็นไปตามเงื่อนไขให้ทำซ้ำขั้นตอนที่ 2-3 โดยโซลูชั่นที่มีความเหมาะสม n ลำดับแรกจะถูกใช้เป็นโซลูชั่นเริ่มต้นในรอบถัดไป

งานวิจัยนี้ได้ทำการทดสอบประสิทธิภาพของอัลกอริทึม BBO ที่นำเสนอกับชุดข้อมูล Brandimarte library (BRdata) และชุดข้อมูล Barnes and Chambers library (BCdata) จากนั้นนำผลการทดสอบไปเปรียบเทียบกับอัลกอริทึมอื่นๆ พบว่า อัลกอริทึม BBO ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมอื่นๆในชุดข้อมูล BRdata และเมื่อนำอัลกอริทึม BBO เปรียบเทียบกับ GA พบว่า BBO ให้ผลลัพธ์ที่ใกล้เคียงกับ GA แต่ใช้ระยะเวลาในการประมวลผลน้อยกว่า

เมื่อทำการศึกษาอัลกอริทึม BBO จึงสามารถวิเคราะห์ให้เห็นจุดเด่นได้คือ ในขั้นตอน Migration งานวิจัยนี้ได้กำหนดให้ความน่าจะเป็นที่แต่ละโซลูชั่นจะถูกเลือกขึ้นอยู่กับอัตราการ immigration และ emigration ที่คำนวณได้ ซึ่งทำให้กระบวนการปรับปรุงมีแนวทางมากขึ้น โซลูชั่นที่ไม่ดีจะได้รับการปรับปรุงจากโซลูชั่นที่ดี แต่ทั้งนี้อัลกอริทึม BBO ยังมีข้อด้อยที่สำคัญคือ ยังไม่มีขั้นตอนที่อัลกอริทึมสามารถค้นหาคำตอบได้อย่างละเอียดในพื้นที่ local ส่งผลให้อัลกอริทึมไม่สามารถแก้ปัญหาการติด local optimum ได้อย่างเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 3
การแก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงานโดยประยุกต์ใช้อัลกอริทึมการผสมพันธุ์ของผึ้ง

ในบทนี้จะกล่าวถึงการปรับปรุงอัลกอริทึมการผสมพันธุ์ของผึ้ง (MBO) เพื่อแก้ปัญหาการจัดตารางแบบยืดหยุ่นตามงาน โดยในงานวิจัยที่นำเสนอนี้จะใช้อัลกอริทึม MBO เป็นโครงสร้างหลัก และเพิ่มประสิทธิภาพการค้นหาคำตอบในพื้นที่โลคอลของอัลกอริทึม MBO โดยประยุกต์ใช้อัลกอริทึมการค้นหาอาหารของผึ้ง และเพื่อป้องกันไม่ให้อัลกอริทึม MBO ลู่เข้าสู่คำตอบที่ยังไม่เหมาะสมเร็วเกินไป จึงนำพฤติกรรมของผึ้งกาฝากมาใช้ร่วมกับอัลกอริทึม MBO

ในงานวิจัยที่นำเสนอนี้จะกำหนดให้ผึ้งเป็นตัวแทนของประชากร (โซลูชั่น) และความแข็งแรงของผึ้งเป็นตัวแทนความเหมาะสมของโซลูชั่น โดยสามารถอธิบายขั้นตอนของอัลกอริทึมที่นำเสนอได้ตามลำดับดังนี้

3.1 ขั้นตอนการสร้างโซลูชั่นเริ่มต้น

การสร้างโซลูชั่นเริ่มต้นถูกแยกพิจารณาเป็น 2 ส่วนคือ ส่วนลำดับงาน (Operation sequence part) และส่วนเครื่องจักร (Machine assignment part) โดยสามารถอธิบายรายละเอียดได้ดังนี้

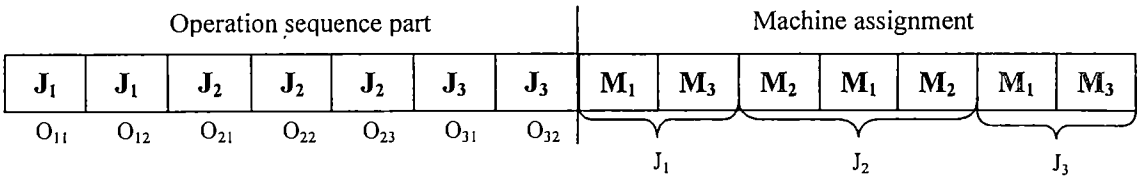
- การสร้างโซลูชั่นเริ่มต้นในส่วนเครื่องจักร

ในการสร้างส่วนเครื่องจักรของโซลูชั่นเริ่มต้นแต่ละตัวนั้นสามารถทำได้โดยการสุ่มเลือกกฎ 1 กฎ อธิบายได้ดังต่อไปนี้

ตารางที่ 3.1 ตารางปัญหาการจัดตารางแบบยืดหยุ่นตามงาน

JOB	OPERATION	DURATION		
		Machine 1	Machine 2	Machine 3
Job_1	O_{11}	2	4	8
	O_{12}	-	4	7
Job_2	O_{21}	2	8	9
	O_{22}	5	-	3
	O_{23}	6	5	-
Job_3	O_{31}	2	6	-
	O_{32}	7	6	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 3.1 ตัวอย่างรูปแบบโซลูชั่นเริ่มต้น (อ้างอิงตารางที่ 3.1)

1) Random rule คือ การสุ่มเลือกเครื่องจักรให้กับแต่ละโอเปอเรชั่น

2) Operation minimum processing time rule คือ การเลือกเครื่องจักรที่มีระยะเวลาในการดำเนินงานน้อยที่สุดให้กับแต่ละโอเปอเรชั่น ยกตัวอย่างเช่น เครื่องจักร M_1 จะถูกจับคู่กับโอเปอเรชั่น O_{11} เนื่องจากเครื่องจักร M_1 ใช้เวลาในการดำเนินการน้อยที่สุดคือ 2 วัน, เครื่องจักร M_2 จะถูกจับคู่กับโอเปอเรชั่น O_{12} เนื่องจากเครื่องจักร M_2 ใช้เวลาในการดำเนินการน้อยที่สุดคือ 4 วัน เป็นต้น

3) Local minimum processing time rule คือ การเลือกเครื่องจักรที่มีภาระงานน้อยที่สุดให้กับแต่ละโอเปอเรชั่น ทั้งนี้หากเครื่องจักรทุกตัวไม่มีภาระงานเลย เครื่องจักรที่มีระยะเวลางานน้อยที่สุดจะถูกเลือก ยกตัวอย่างจากตารางที่ 3.1 ในตอนเริ่มต้นทุกเครื่องจักรมีภาระงานเป็นศูนย์ โดยจะเริ่มต้นพิจารณาเลือกเครื่องจักรให้โอเปอเรชั่น O_{11} ซึ่งจะพบว่าเครื่องจักร M_1 ใช้ระยะเวลาน้อยที่สุด จึงถูกเลือกให้กับโอเปอเรชั่น O_{11} ดังนั้นในขณะนี้เครื่องจักร M_1 จึงมีภาระงาน 2 วัน จากนั้นพิจารณาเลือกเครื่องจักรให้กับโอเปอเรชั่น O_{12} โดยพิจารณาทีละเครื่องจักรจะพบว่า เมื่อรวมระยะเวลางานของโอเปอเรชั่น O_{12} กับภาระงานของแต่ละเครื่องจักรจะพบว่า หากทำโอเปอเรชั่น O_{12} บนเครื่องจักร M_1 จะทำให้ เครื่องจักร M_1 มีภาระงานรวม 2+5 = 7 วัน หากทำโอเปอเรชั่น O_{12} บนเครื่องจักร M_2 จะทำให้เครื่องจักร M_2 มีภาระงานรวม 4 วัน และหากทำโอเปอเรชั่น O_{12} บนเครื่องจักร M_3 จะทำให้เครื่องจักร M_3 มีภาระงานรวม 7 วัน ดังนั้นจึงเลือกจับคู่โอเปอเรชั่น O_{12} กับเครื่องจักร M_2 เพราะมีภาระงานน้อยที่สุด เป็นต้น

- การสร้างโซลูชั่นเริ่มต้นในส่วนลำดับงาน

หลังจากกำหนดเครื่องจักรให้กับแต่ละโอเปอเรชั่นเสร็จสิ้นแล้ว ขั้นตอนนี้จะเป็นการจัดลำดับโอเปอเรชั่น ซึ่งสามารถทำได้โดยการสุ่มเลือกกฎการจัดลำดับ 1 กฎดังต่อไปนี้

1) Random rule คือ การสุ่มจัดลำดับโอเปอเรชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2) Most work remaining rule คือ การเลือกโอเปอเรชั่นของงานที่มีระยะเวลารวมคงเหลือมากที่สุดมาจัดลำดับก่อน เช่น Job_1 มีระยะเวลารวมของทุกโอเปอเรชั่นเท่ากับ 7 วัน และ Job_2 มีระยะเวลารวมของทุกโอเปอเรชั่นเท่ากับ 10 วัน ดังนั้นโอเปอเรชั่นที่เหลืออยู่ในอันดับแรกของ Job_2 จะถูกเลือกมาจัดลำดับก่อน เป็นต้น

3) Most number of operation remaining rule คือ การเลือกพิจารณางานที่ประกอบด้วยจำนวนโอเปอเรชั่นซึ่งยังไม่ถูกดำเนินการอยู่มากที่สุดมาจัดลำดับก่อน

3.2 ขั้นตอนการหาค่าความเหมาะสมของโซลูชั่นเริ่มต้น

หลังจากสร้างโซลูชั่นเริ่มต้นเสร็จสิ้นแล้ว ในขั้นตอนนี้จะเป็นการคำนวณหาระยะทางรวมที่สั้นที่สุดของโซลูชั่นเริ่มต้นทุกตัว

โซลูชั่นเริ่มต้นที่มีความเหมาะสมมากที่สุดจะถูกกำหนดเป็น โซลูชั่นผึ้งนางพญา (Queen solution) และโซลูชั่นเริ่มต้นที่เหลือจะถูกกำหนดเป็น โซลูชั่นผึ้งเพศผู้ (Drone solution)

3.3 ขั้นตอนการคัดเลือกโซลูชั่นผึ้งเพศผู้

การเลือกโซลูชั่นผึ้งเพศผู้ใดๆ เพื่อผสมพันธุ์กับโซลูชั่นผึ้งนางพญานั้น จะพิจารณาที่ค่าความเหมาะสมสัมพันธ์ระหว่างโซลูชั่นผึ้งทั้งสอง โซลูชั่นผึ้งเพศผู้ที่มีความเหมาะสมมากจะมีโอกาสถูกเลือกมาก และโซลูชั่นผึ้งเพศผู้ที่มีความเหมาะสมน้อยจะมีโอกาสถูกเลือกน้อย

จากสมการที่ 3.2 ค่า Prob(Q,D) คือ ค่าความน่าจะเป็นที่โซลูชั่นผึ้งเพศผู้ D จะถูกเลือกให้ผสมพันธุ์กับโซลูชั่นผึ้งนางพญา Q ค่า $\Delta(f)$ คือผลต่างค่าความเหมาะสมของโซลูชั่นผึ้งนางพญา Q กับโซลูชั่นผึ้งเพศผู้ D และ speed คือความเร็วของโซลูชั่นผึ้งนางพญา Q โดยค่า speed จะค่อยๆ ลดลงในอัตราที่เท่ากันทุกรอบตามสมการที่ 3.3 ทั้งนี้หาก Prob(Q,D) มีค่ามากกว่าค่าตัวแปร ε ที่กำหนดไว้ สเปิร์มของโซลูชั่นผึ้งเพศผู้ D จะถูกเก็บไว้ในรังไข่ของผึ้งนางพญา ทั้งนี้โซลูชั่นผึ้งเพศผู้ D ที่มีความเหมาะสมมากจะมีโอกาสถูกเลือกมาก ส่วนโซลูชั่นผึ้งเพศผู้ที่มีความเหมาะสมน้อยจะมีโอกาสถูกเลือกน้อย กระบวนการผสมพันธุ์นี้จะดำเนินไปเรื่อยๆ จนกระทั่งรังไข่ที่ใช้เก็บสเปิร์มของผึ้งนางพญาเต็ม

วิธีการคัดเลือกโซลูชั่นผึ้งเพศผู้มีขั้นตอนดังนี้ 1) คำนวณหาค่า prob(Q,D) ของโซลูชั่นผึ้งนางพญา Q กับโซลูชั่นผึ้งเพศผู้ D 2) เปรียบเทียบค่า prob(Q,D) กับค่าตัวแปรที่กำหนดไว้ 3) สเปิร์มของโซลูชั่นผึ้งเพศผู้ D จะถูกบันทึกไว้เมื่อค่า prob(Q,D) มากกว่าค่าตัวแปรที่กำหนดไว้

$$prob(Q,D) > \varepsilon \tag{3.1}$$

$$prob(Q,D) = e^{\frac{-\Delta(f)}{speed(t)}} \tag{3.2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

$$speed(t+1) = \alpha * speed(t) \tag{3.3}$$

ในงานวิจัยที่นำเสนอนี้ จะแบ่งโซลูชั่นลูก (Offspring) เป็น 2 ประเภทคือ โซลูชั่นลูกเพศเมีย (Female Offspring) ซึ่งเกิดจากกระบวนการครอสโอเวอร์และมิวเทชั่น และโซลชั่นลูกเพศผู้ (Male offspring) ซึ่งเกิดจากการถ่ายทอดยีนส์โดยตรงจากโซลูชั่นผึ้งนางพญา โดยอธิบายรายละเอียดไว้ในหัวข้อที่ 3.4 และ 3.5 ตามลำดับ

3.4 ขั้นตอนการสร้างโซลูชั่นลูกเพศเมีย

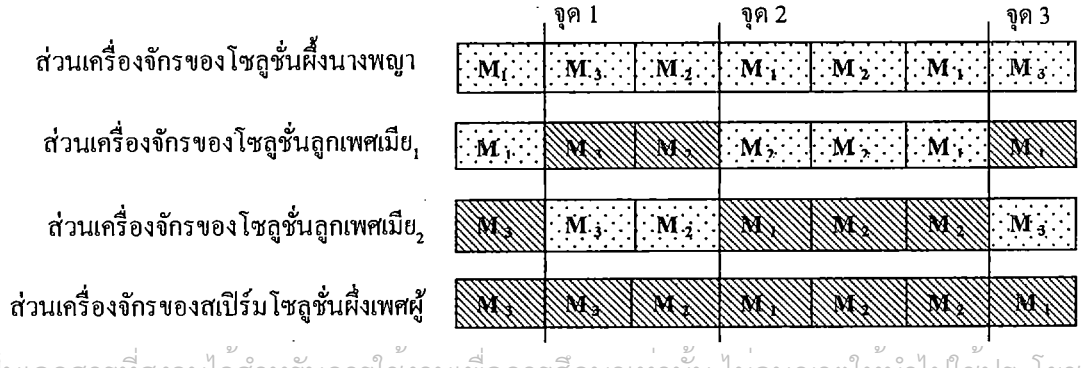
3.4.1 ขั้นตอนการครอสโอเวอร์

การครอสโอเวอร์เป็นวิธีการถ่ายทอดลักษณะทางพันธุกรรมของโซลูชั่นผึ้งนางพญาและสเปิร์มของโซลูชั่นผึ้งเพศผู้แต่ละตัวไปสู่โซลูชั่นลูกเพศเมีย จำนวนโซลูชั่นลูกเพศเมียที่ได้จากกระบวนการครอสโอเวอร์สามารถคำนวณได้ตามสมการที่ 3.4

$$2(\text{QueenN} \times \text{SpermN}) \tag{3.4}$$

เมื่อ QueenN คือจำนวนของโซลูชั่นนางพญา และ SpermN คือจำนวนสเปิร์มของโซลูชั่นผึ้งเพศผู้ ในการครอสโอเวอร์โซลูชั่นผึ้งนางพญาจะถูกจับคู่กับสเปิร์มโดยวิธีการสุ่มทีละตัว จากนั้นจะสุ่มค่าที่อยู่ในช่วง 0 ถึง 1 ขึ้นมาหนึ่งค่า หากค่าที่สุ่มได้น้อยกว่าค่าความน่าจะเป็นในการครอสโอเวอร์ (Crossover Probability: P_C) ในงานวิจัยนี้จะใช้วิธีการครอสโอเวอร์แบบ Three-Point Crossover กับส่วนเครื่องจักร และ Modified Precedence Operation Crossover (MPOX) กับส่วนลำดับงาน ดังแสดงตัวอย่างในรูปที่ 3.2 และ 3.3 ตามลำดับ แต่หากค่าที่สุ่มได้มากกว่าค่า P_C โซลูชั่นผึ้งนางพญาและสเปิร์มจะถูกถ่ายทอดโดยตรงไปยังโซลูชั่นลูกทั้ง 2 ตัว

ทั้งนี้ในงานวิจัยที่นำเสนอนี้ คู่ของโซลูชั่นผึ้งนางพญากับสเปิร์มที่ดีที่สุด จะถูกถ่ายทอดโดยตรงสู่โซลูชั่นลูก 1 คู่ เพื่อเพิ่มโอกาสให้โซลูชั่นที่ดีมีชีวิตอยู่รอดต่อไป

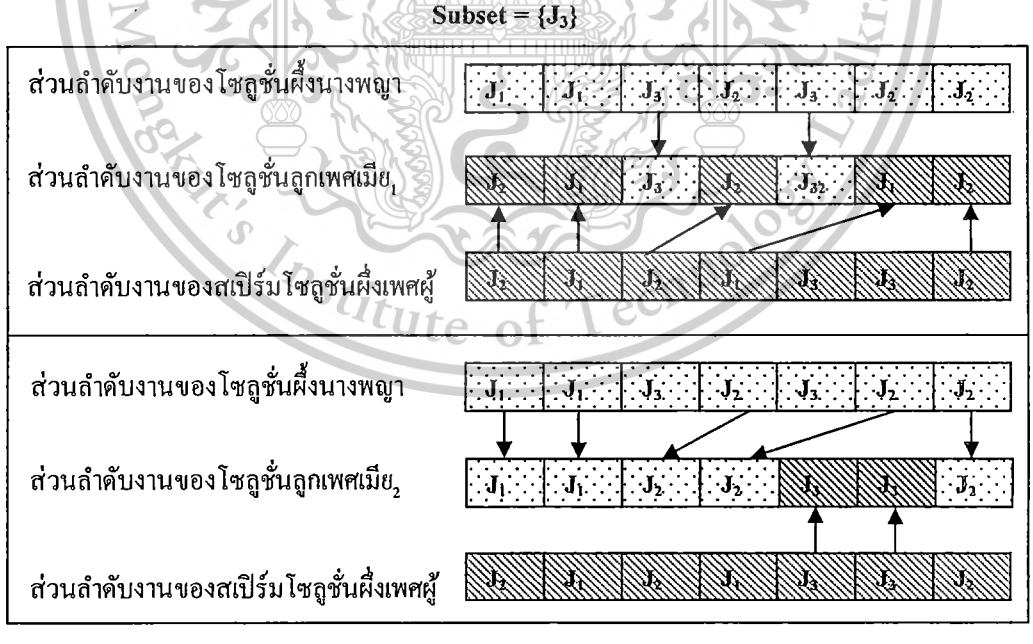


รูปที่ 3.2 การครอสโอเวอร์แบบ Three-Point สำหรับส่วนเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

การครอสโอเวอร์แบบ Three-Point Crossover มีวิธีการคือ ต้องทำการสุ่มเลือกตำแหน่งที่จะทำการครอสโอเวอร์ (Crossover point) จำนวน 3 จุด จากตัวอย่างในรูปที่ 3.2 พบว่าเมื่อใช้จุดครอสโอเวอร์ 3 จุด โซลูชั่นจะถูกแบ่งออกเป็น 4 ส่วน โดยโซลูชั่นลูกตัวที่ 1 จะได้รับการถ่ายทอดคุณลักษณะส่วนที่ 1 มาจากโซลูชั่นผึ้งนางพญา ได้รับการถ่ายทอดคุณลักษณะส่วนที่ 2 มาจากสเปิร์ม ได้รับการถ่ายทอดคุณลักษณะส่วนที่ 3 มาจากโซลูชั่นผึ้งนางพญา และได้รับการถ่ายทอดคุณลักษณะส่วนที่ 4 มาจากสเปิร์ม ในทางตรงกันข้ามกับโซลูชั่นลูกตัวที่ 2 จะได้รับการถ่ายทอดคุณลักษณะส่วนที่ 1 มาจากสเปิร์ม ได้รับการถ่ายทอดคุณลักษณะส่วนที่ 2 มาจากโซลูชั่นผึ้งนางพญา ได้รับการถ่ายทอดคุณลักษณะส่วนที่ 3 มาจากสเปิร์ม และได้รับการถ่ายทอดคุณลักษณะส่วนที่ 4 มาจากโซลูชั่นผึ้งนางพญา

การครอสโอเวอร์แบบ MPOX คือการสุ่มกำหนด subset ของ job ที่ต้องการถ่ายทอดจากโซลูชั่นผึ้งนางพญาและสเปิร์มไปยังโซลูชั่นลูก จากตัวอย่างในรูปที่ 3.3 subset ที่สุ่มกำหนดไว้คือ job หมายเลข 3 โดย job หมายเลข 3 จากโซลูชั่นผึ้งนางพญาจะถูกถ่ายทอดไปยังโซลูชั่นลูกตัวที่ 1 ในตำแหน่งเดิม ส่วนตำแหน่งที่เหลือของโซลูชั่นลูกตัวที่ 1 จะได้รับการถ่ายทอดจากสเปิร์มซึ่งต้องเป็นหมายเลข job ไม่ซ้ำกับ subset ตามลำดับจากซ้ายไปขวา



รูปที่ 3.3 การครอสโอเวอร์แบบ MPOX สำหรับส่วนลำดับงาน

และในทางตรงกันข้าม โซลูชั่นลูกตัวที่ 2 จะได้รับการถ่ายทอด job หมายเลข 3 ในตำแหน่งเดียวกันจากสเปิร์ม ส่วนในตำแหน่งอื่นๆของโซลูชั่นลูกตัวที่ 2 จะได้รับการถ่ายทอดจาก

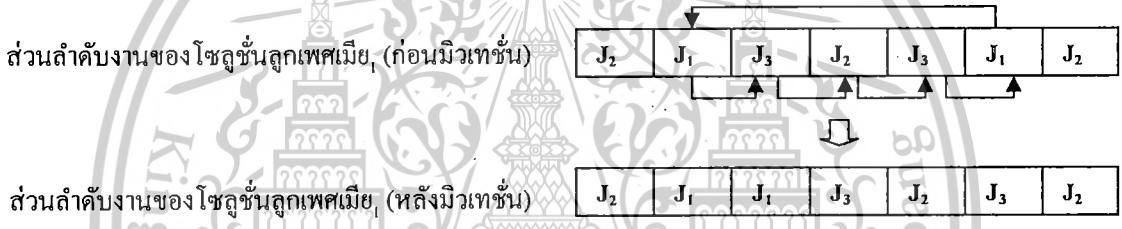
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โซลูชั่นผึ้งนางพญาซึ่งต้องเป็นหมายเลข job ที่ไม่ซ้ำกับ subset ตามลำดับจากซ้ายไปขวา หลังจากกระบวนการครอสโอเวอร์เสร็จสิ้น โซลูชั่นลูกเพศเมียทั้งหมดจะถูกส่งไปในขั้นตอนมิวเทชั่น

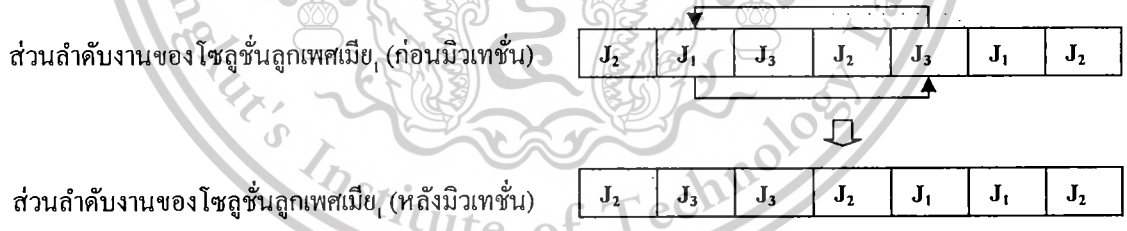
3.4.2 ขั้นตอนการมิวเทชั่น

การมิวเทชั่นดำเนินการโดยการสุ่มค่าที่อยู่ในช่วง 0 ถึง 1 จำนวนหนึ่งค่า จากนั้นเปรียบเทียบค่าที่สุ่มได้กับค่าความน่าจะเป็นในการมิวเทชั่น (Mutation probability: P_M) หากค่าที่สุ่มได้น้อยกว่าค่า P_M ให้ทำการมิวเทชั่นตำแหน่งดังกล่าวในโซลูชั่นลูก

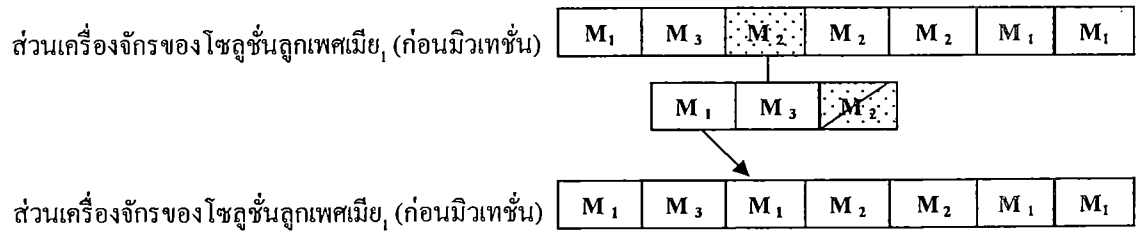
สำหรับการมิวเทชั่นในส่วนลำดับงานของโซลูชั่นลูกแต่ละโซลูชั่นนั้นจะมี 2 วิธีคือ inserting move และ swapping move ดังแสดงในรูป 3.4 และ 3.5 ตามลำดับ โดยจะสุ่มเลือกมา 1 วิธีเท่านั้น ส่วนการมิวเทชั่นในส่วนเครื่องจักรของโซลูชั่นลูก จะใช้วิธีการเลือกเครื่องจักรใหม่ ดังแสดงในรูปที่ 3.6



รูปที่ 3.4 การมิวเทชั่นแบบ Inserting Move สำหรับส่วนลำดับงาน



รูปที่ 3.5 การมิวเทชั่นแบบ Swapping Move สำหรับส่วนลำดับงาน



รูปที่ 3.6 การมิวเทชั่นแบบสุ่มเลือกเครื่องจักรใหม่ สำหรับส่วนเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

หลังจากกระบวนการมิวเทชั่นเสร็จสิ้น งานวิจัยนี้จะกำหนดให้โซลูชั่นลูกเพศเมียที่มีความเหมาะสมมากที่สุดเป็นโซลูชั่นผึ้งนางพญาตัวใหม่ (New queen solution) และโซลูชั่นลูกเพศเมียที่เหลือทั้งหมดถูกกำหนดให้เป็นโซลูชั่นผึ้งงาน (Worker solution) โดยโซลูชั่นผึ้งงานจะมีหน้าที่ในการหาอาหาร

3.5 ขั้นตอนการปรับปรุงโซลูชั่นผึ้งนางพญา

โซลูชั่นผึ้งนางพญาตัวใหม่จะได้รับการปรับปรุงด้วยวิธีการค้นหาบนเส้นทางวิกฤติ (Local search based on the critical path method) วิธีการค้นหาบนเส้นทางวิกฤติ เป็นวิธีการเปลี่ยนตำแหน่งของโอเปอเรชั่นที่อยู่บนเส้นทางวิกฤติ ไปยังตำแหน่งใหม่ โดยการเปลี่ยน 1 ตำแหน่งก็จะทำให้ได้โซลูชั่นใหม่ 1 โซลูชั่น ซึ่งโซลูชั่นใหม่นี้จะถูกนำมาเปรียบเทียบกับโซลูชั่นก่อนการเปลี่ยนตำแหน่งเสมอ

เส้นทางวิกฤติคือ เส้นทางที่ยาวที่สุด ซึ่งสามารถหาได้โดยการคำนวณหาค่าดังต่อไปนี้ 1. ระยะเวลาที่จะเริ่มต้นโอเปอเรชั่นได้เร็วที่สุด (early start time (ES)) 2. ระยะเวลาเสร็จสิ้นของโอเปอเรชั่นอย่างเร็วที่สุด (early finish time (EF)) 3. ระยะเวลาเริ่มต้นของโอเปอเรชั่นอย่างช้าที่สุด (late start time (LS)) 4. ระยะเวลาเสร็จสิ้นของโอเปอเรชั่นอย่างช้าที่สุด (late finish time (LF)) และ 5. ระยะเวลาที่กิจกรรมสามารถล่าช้าได้โดยไม่ทำให้โครงการล่าช้า (total slack (TS)) ดังแสดงในสมการที่ 3.5 ถึง 3.10 ตามลำดับ ทั้งนี้โอเปอเรชั่นที่มีค่า TS เท่ากับศูนย์จะถูกเรียกว่า "โอเปอเรชั่นวิกฤติ (Critical operation)"

ดังนั้นหากสามารถปรับตำแหน่งของโอเปอเรชั่นวิกฤติไปยังตำแหน่งใหม่ที่มีความเหมาะสมมากกว่า ก็จะส่งผลให้โซลูชั่นผึ้งนางพญาตัวใหม่มีเส้นทางสั้นลงได้ (มีความเหมาะสมมากขึ้น) โดย PM_{ij}^{k} คือโอเปอเรชั่นที่อยู่ในลำดับก่อนโอเปอเรชั่น O_{ij} บนเครื่องจักร k, PJ_{ij} คือโอเปอเรชั่นที่อยู่ในลำดับก่อนโอเปอเรชั่น O_{ij} บน job_i, O_{ij+1}^{k} คือโอเปอเรชั่นที่อยู่ในลำดับหลังโอเปอเรชั่น O_{ij} บนเครื่องจักร k, O_{ij+1} คือ โอเปอเรชั่นที่อยู่ในลำดับหลังโอเปอเรชั่น O_{ij} บน job_i, LS_{ij+1}^{k} คือระยะเวลาเริ่มต้นอย่างช้าที่สุดของโอเปอเรชั่นที่อยู่ในลำดับหลังโอเปอเรชั่น O_{ij} บนเครื่องจักร k และ LS_{ij+1} คือระยะเวลาเริ่มต้นอย่างช้าที่สุดของโอเปอเรชั่นที่อยู่ในลำดับหลังโอเปอเรชั่น O_{ij} บน job_i

รูปที่ 3.7 แสดงตัวอย่างโอเปอเรชั่นวิกฤติของโซลูชั่นเริ่มต้นในรูปที่ 3.1 ซึ่งโอเปอเรชั่นวิกฤติคือโอเปอเรชั่น O_{11} O_{22} O_{31} และ O_{32} เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

$$ES_{ij} = \max(PM_{ij}^{k}, PJ_{ij}) \tag{3.5}$$

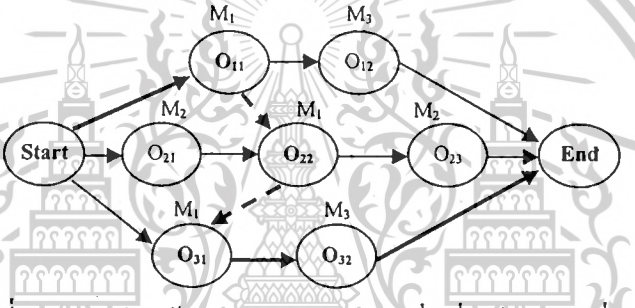
$$EF_{ij} = Duration(O_{ij}) + ES_{ij} \tag{3.6}$$

$$LS_{ij+1} = (LF(O_{ij+1}) - EF(O_{ij+1})) + ES(O_{ij+1}) \tag{3.7}$$

$$LS_{ij}^{k} = (LF(O_{ij+1}^{k}) - EF(O_{ij+1}^{k})) + ES(O_{ij+1}^{k}) \tag{3.8}$$

$$LF_{ij} = \min(LS_{ij+1}^{k}, LS_{ij+1}) \tag{3.9}$$

$$TS_{ij} = LF_{ij} - EF_{ij} \tag{3.10}$$



รูปที่ 3.7 ตัวอย่างเส้นทางวิกฤติของโซลูชั่นเริ่มต้นในรูปที่ 3.1

3.6 ขั้นตอนการหาอาหารของผึ้งงาน

โซลูชั่นผึ้งงานจะมีหน้าที่ในการหาอาหาร ซึ่งในงานวิจัยที่นำเสนอนี้ได้นำขั้นตอนของผึ้งเอมพลอยและผึ้งออนลุกเกอร์ของอัลกอริทึมการค้นหาแหล่งอาหารของผึ้ง (Artificial bee colony algorithm) มาประยุกต์ใช้ โดยกำหนดให้ผึ้งงานที่สามารถค้นพบแหล่งอาหารได้ดีที่สุด จะมีร่างกายที่แข็งแรง รวมทั้งมีรังไข่ที่สมบูรณ์ ส่งผลให้ผึ้งงานตัวนี้สามารถวางไข่ที่เจริญเติบโตไปเป็นผึ้งเพศผู้ได้ โดยสามารถอธิบายขั้นตอนการประยุกต์ใช้อัลกอริทึมการค้นหาแหล่งอาหารของผึ้งได้ดังนี้

(1) กำหนดให้ตำแหน่งของโซลูชั่นผึ้งงาน เป็นตำแหน่งแหล่งอาหารเริ่มต้นของผึ้งเอมพลอย

(2) ทำการค้นหาตำแหน่งแหล่งอาหารแห่งใหม่จากตำแหน่งแหล่งอาหารเริ่มต้นผึ้งเอมพลอยแต่ละตัวโดยการสุ่มเลือกเทคนิคที่จะใช้ในการค้นหาระหว่าง Iterated local search technique และ Scatter search technique

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

(3) เปรียบเทียบตำแหน่งแหล่งอาหารแห่งใหม่กับตำแหน่งแหล่งอาหารเริ่มต้นผึ้งเอมพลอยที่สัมพันธ์กัน และทำการบันทึกตำแหน่งที่มีความเหมาะสมมากกว่าเก็บไว้ โดยกลุ่มตำแหน่งที่มีความเหมาะสมมากกว่านี้จะถูกกำหนดเป็นตำแหน่งแหล่งอาหารของผึ้งเอมพลอย

(4) ใช้วิธี Tournament selection ดังแสดงในรูปที่ 3.8 ในการเลือกตำแหน่งแหล่งอาหารของผึ้งเอมพลอย ยกตัวอย่างเช่น สุ่มเลือกตำแหน่งแหล่งอาหารของผึ้งเอมพลอยมา 3 ตำแหน่ง จากนั้นตำแหน่งแหล่งอาหารที่ดีที่สุดจะถูกเลือก เป็นต้น โดยจะกำหนดตำแหน่งแหล่งอาหารที่ถูกเลือกนี้เป็นตำแหน่งแหล่งอาหารเริ่มต้นของผึ้งออนลุกเกอร์ (ในขั้นตอนนี้แหล่งอาหารของผึ้งเอมพลอยมีโอกาสจะถูกเลือกซ้ำ และมีโอกาสที่แหล่งอาหารของผึ้งเอมพลอยบางตัวจะไม่ถูกเลือก)

โซลูชั่นที่สุ่มเลือกมา	ค่าความเหมาะสม	
ตำแหน่งแหล่งอาหารของผึ้งเอมพลอย	14	
ตำแหน่งแหล่งอาหารของผึ้งเอมพลอย	***8***	⇦ โซลูชั่นที่ถูกเลือก
ตำแหน่งแหล่งอาหารของผึ้งเอมพลอย	10	

รูปที่ 3.8 Tournament selection method

(5) ทำการค้นหาตำแหน่งแหล่งอาหารแห่งใหม่จากแหล่งอาหารเริ่มต้นของผึ้งออนลุกเกอร์แต่ละตัว โดยวิธีการดังนี้ 1) หากแหล่งอาหารเริ่มต้นของผึ้งออนลุกเกอร์ติดอยู่ในตำแหน่ง Local optimum จะประยุกต์ใช้อัลกอริทึมการอบอ่อนจำลอง (Simulated annealing algorithm) เป็นวิธีการค้นหา ซึ่งอัลกอริทึมนี้มีคุณสมบัติที่สามารถเคลื่อนตำแหน่งเดิมให้หลุดออกจาก Local optimum 2) หากแหล่งอาหารเริ่มต้นของผึ้งออนลุกเกอร์ไม่ติดอยู่ในตำแหน่ง Local optimum จะประยุกต์ใช้ Filter and Fan algorithm มาเป็นวิธีการค้นหา

(6) เปรียบเทียบตำแหน่งแหล่งอาหารแห่งใหม่กับตำแหน่งแหล่งอาหารเริ่มต้นของผึ้งออนลุกเกอร์ที่สัมพันธ์กัน และทำการบันทึกตำแหน่งแหล่งอาหารที่มีความเหมาะสมมากกว่าเก็บไว้ โดยกลุ่มตำแหน่งที่มีความเหมาะสมมากกว่านี้จะถูกกำหนดเป็นตำแหน่งแหล่งอาหารของผึ้งออนลุกเกอร์

(7) นำตำแหน่งแหล่งอาหารของผึ้งเอมพลอยที่ไม่ถูกเลือกและตำแหน่งแหล่งอาหารของผึ้งออนลุกเกอร์มารวมกัน แล้วคัดเลือกตำแหน่งแหล่งอาหารที่ดีที่สุดมาจำนวน N ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

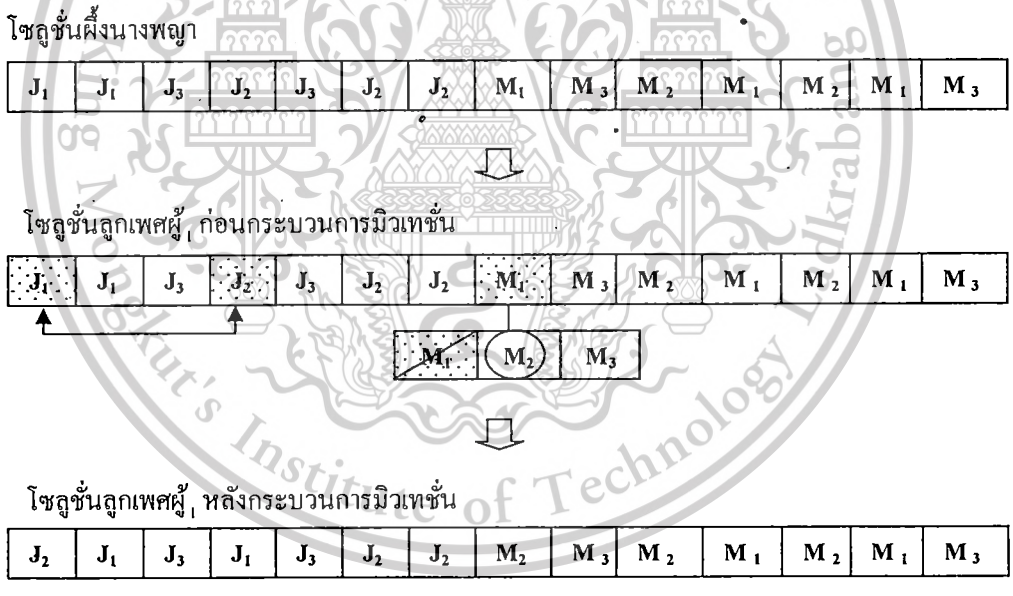
(8) บันทึกตำแหน่งแหล่งอาหารที่ดีที่สุดจำนวน N ตัวข้างต้น เป็นตำแหน่งของโซลูชั่นผึ้งงานกลุ่มที่ 3

3.7 ขั้นตอนการสร้างโซลูชั่นลูกเพศผู้

โซลูชั่นลูกเพศผู้เกิดโดยการถ่ายทอดยีนส์โดยตรงจากโซลูชั่นผึ้งนางพญาและโซลูชั่นผึ้งงาน ซึ่งสามารถอธิบายได้ดังต่อไปนี้

3.7.1 โซลูชั่นลูกเพศผู้ที่เกิดจากโซลูชั่นผึ้งนางพญา

โซลูชั่นลูกเพศผู้ไม่ได้เกิดจากการผสมพันธุ์ แต่เกิดจากการถ่ายทอดยีนส์โดยตรงจากโซลูชั่นผึ้งนางพญา ดังนั้นยีนส์ของโซลูชั่นลูกเพศผู้จะเหมือนกับยีนส์ของโซลูชั่นผึ้งนางพญาทุกประการ ทั้งนี้โซลูชั่นลูกเพศผู้ทุกตัวที่เกิดจากโซลูชั่นผึ้งนางพญา จะต้องผ่านกระบวนการมิวเทชั่น ซึ่งกระบวนการมิวเทชั่นนี้จะทำให้โซลูชั่นลูกเพศผู้แต่ละตัวแตกต่างจากโซลูชั่นนางพญา ดังแสดงในรูปที่ 3.9



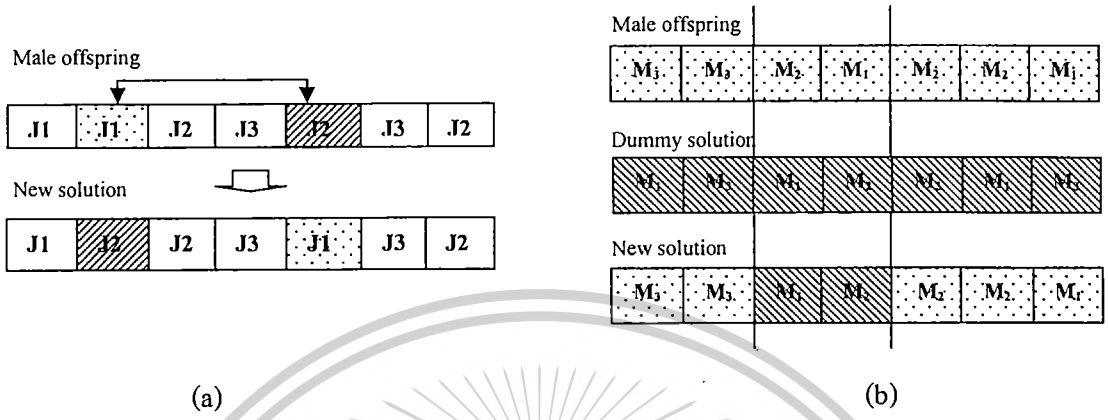
รูปที่ 3.9 การสร้างโซลูชั่นลูกเพศผู้จากโซลูชั่นผึ้งนางพญา

โซลูชั่นลูกเพศผู้ที่ผ่านกระบวนการมิวเทชั่นแล้ว จะได้รับการปรับปรุงโดยวิธีฮิวริสติก 3 วิธี ดังต่อไปนี้

3.7.1.1 วิธีฮิวริสติกแบบที่ 1 สำหรับส่วนลำดับงานนั้น จะใช้วิธีการสลับตำแหน่งโอเปอเรชั่นที่สุ่มเลือกได้ ดังแสดงในรูปที่ 3.10(a) สำหรับส่วนเครื่องจักรจะใช้วิธีการครอสโอเวอร์แบบสองจุดระหว่างส่วนเครื่องจักรของโซลูชั่นลูกเพศผู้และโซลูชั่นที่สุ่มสร้างขึ้นมา (โซลูชั่นดัมมี่)

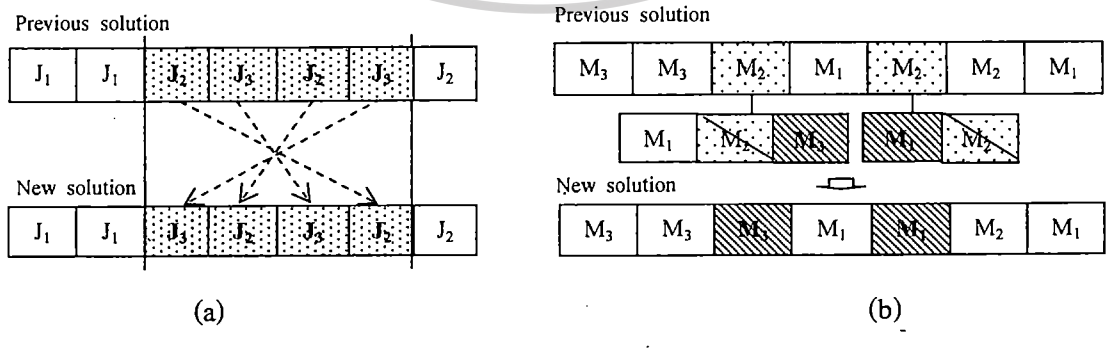
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ดังแสดงในรูปที่ 3.10(b) ซึ่งการเปลี่ยนตำแหน่งในแต่ละครั้งจะทำให้ได้โซลูชั่นใหม่เสมอ กระบวนค้นหาโซลูชั่นใหม่นี้จะดำเนินไปจนกระทั่งเจอโซลูชั่นใหม่ที่ดีกว่าโซลูชั่นลูกเพศผู้ หรือโซลูชั่นใหม่ถูกสร้างขึ้นครบตามจำนวนที่กำหนดไว้



รูปที่ 3.10 วิธีฮิวริสติกแบบที่ 1: (a) ส่วนลำดับงาน (b) ส่วนเครื่องจักร

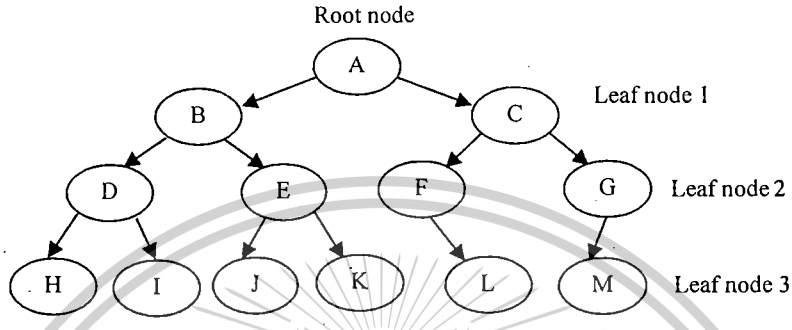
3.7.1.2 วิธีฮิวริสติกแบบที่ 2 โดยมีขั้นตอนในการทำเริ่มต้นจากการกำหนดให้โซลูชั่นลูกเพศผู้เป็น Previous solution หลังจากนั้นจะทำการสร้างโซลูชั่นใหม่จาก Previous solution โดยการสลับตำแหน่งโอเปอเรชั่นสำหรับส่วนลำดับงาน และการเลือกเครื่องจักรใหม่สำหรับส่วนเครื่องจักร ดังแสดงในรูปที่ 3.11 (a) และ (b) ตามลำดับ เมื่อทำการสร้างโซลูชั่นใหม่เรียบร้อยแล้ว เปรียบเทียบโซลูชั่นใหม่กับโซลูชั่นลูกเพศผู้ หากโซลูชั่นใหม่มีความเหมาะสมมากกว่า ให้แทนที่โซลูชั่นลูกเพศผู้ด้วยโซลูชั่นใหม่ แต่หากโซลูชั่นใหม่แย่กว่า ให้ทำการแทนที่ Previous solution ด้วยโซลูชั่นใหม่ และทำซ้ำในขั้นตอนข้างต้นจนกระทั่งพบโซลูชั่นใหม่ที่ดีกว่าโซลูชั่นลูกเพศผู้ หรือโซลูชั่นใหม่ถูกสร้างขึ้นเท่ากับจำนวนที่กำหนดไว้



รูปที่ 3.11 วิธีฮิวริสติกแบบที่ 2: (a) ส่วนลำดับงาน (b) ส่วนเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.7.1.3 วิธีฮิวริสติกแบบที่ 3 มีลำดับการค้นหาแบบแผนภูมิต้นไม้ (Tree search) โดยกำหนดให้โซลูชั่นลูกเพศผู้เป็น Rood node จากนั้นจะสุ่มเลือกวิธีการปรับปรุง Rood node ส่วนลำดับงานและส่วนเครื่องจักรจากฮิวริสติกแบบที่ 1 และแบบที่ 2 ซึ่งกระบวนการการค้นหาจะเป็นไปตามลำดับ A-B-D-H-I-E-J-K-C-F-L-G-M ดังแสดงในรูปที่ 3.12



รูปที่ 3.12 วิธีฮิวริสติกแบบที่ 3

โดยสามารถอธิบายได้ดังนี้ ให้ทำการสร้างโซลูชั่นใหม่ B จาก Root node จากนั้นเปรียบเทียบโซลูชั่นใหม่ B กับ Root node หากโซลูชั่นใหม่ B มีความเหมาะสมมากกว่า Root node จะแทนที่ Root node ด้วยโซลูชั่นใหม่ B และการค้นหาจะสิ้นสุดลง แต่หากโซลูชั่นใหม่ B แย่กว่า Root node จะทำการสร้างโซลูชั่นใหม่ D ขึ้นมาโดยการปรับเปลี่ยนตำแหน่งของโซลูชั่นใหม่ B ตามวิธีฮิวริสติกแบบที่ 1 หรือแบบที่ 2 จากนั้นจะนำโซลูชั่นใหม่ D ที่ได้ไปเปรียบเทียบความเหมาะสมกับ Root node ทั้งนี้กระบวนการค้นหาจะดำเนินไปเรื่อยๆ จนกระทั่งพบโซลูชั่นใหม่ที่ดีกว่า Root node หรือโซลูชั่นใหม่ตัวที่ M ถูกสร้างขึ้น

3.7.2 โซลูชั่นลูกเพศผู้ที่เกิดจากโซลูชั่นผึ้งงาน

จากที่กล่าวไปแล้วก่อนหน้าถึงคุณสมบัติที่โซลูชั่นผึ้งงานที่มีความสมบูรณ์และแข็งแรง M โซลูชั่นสามารถที่จะให้กำเนิดโซลูชั่นลูกเพศผู้ได้โดยกระบวนการถ่ายทอดยีนส์โดยตรง ดังนั้นโซลูชั่นลูกเพศผู้จำนวน M โซลูชั่นจะเกิดจากการได้รับถ่ายทอดยีนส์จากโซลูชั่นผึ้งงาน

3.8 ขั้นตอนการสร้างโซลูชั่นผึ้งกาฝาก

ขั้นตอนนี้ถูกนำเสนอขึ้นมาเพื่อเพิ่มประสิทธิภาพให้อัลกอริทึมที่นำเสนอไม่ติดอยู่ในตำแหน่ง Local optimum โดยงานวิจัยนี้จะใช้วิธีสุ่มสร้างโซลูชั่นผึ้งกาฝากขึ้นมา จากนั้นจะปรับปรุงโซลูชั่นผึ้งกาฝากด้วยวิธีการค้นหาบนเส้นทางวิกฤติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.9 ขั้นตอนการคัดเลือกโซลูชั่นเริ่มต้นกลุ่มใหม่

เปรียบเทียบความเหมาะสมของโซลูชั่นผึ้งนางพญากับโซลูชั่นผึ้งนางพญาตัวใหม่ และโซลูชั่นผึ้งกาฝาก โดยโซลูชั่นที่เหมาะสมมากที่สุดจะถูกเลือกให้เป็นโซลูชั่นผึ้งนางพญาสำหรับรอบถัดไปของอัลกอริทึม จากนั้นนำโซลูชั่นลูกเพศผู้ทุกตัวมาเรียงลำดับตามความเหมาะสม โซลูชั่นลูกเพศผู้ที่มีความเหมาะสมจำนวน (N-1)/2 โซลูชั่นจะถูกเลือกให้เป็นโซลูชั่นผึ้งเพศผู้สำหรับรอบถัดไปของอัลกอริทึม ส่วนโซลูชั่นผึ้งเพศผู้อีกครึ่งหนึ่งคือ (N-1)/2 จะถูกสุ่มสร้างขึ้น ทั้งนี้จุดประสงค์ในการสุ่มเพื่อทำให้โซลูชั้นผึ้งเพศผู้มีความหลากหลายมากขึ้น

3.10 ขั้นตอนการพิจารณาหยุดการค้นหา

ทำซ้ำในขั้นตอนที่ 3.3 ถึง 3.9 จนกระทั่งครบตามจำนวนรอบที่กำหนด

King Mongkut's Institute of Technology Ladkrabang

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 4
การทดลองและผลการทดลอง

4.1 ชุดข้อมูลที่ใช้ในการทดลอง

ในงานวิจัยนี้ได้ทำการทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐานจากฐานข้อมูลออนไลน์ "A Library of flexible job shop scheduling problem instances" โดยเลือกทดสอบกับชุดข้อมูล Brandimarte [10] ซึ่งประกอบไปด้วยชุดข้อมูลย่อยจำนวน 10 ชุด ดังต่อไปนี้

4.1.1 ชุดข้อมูลย่อย MK01 ประกอบด้วยงานหลักจำนวน 10 งาน แต่ละงานหลักประกอบด้วยจำนวนโอเปอเรชั่นที่แตกต่างกัน ยกตัวอย่างเช่น job 1 ประกอบด้วย 6 โอเปอเรชั่น และ job 2 ประกอบด้วย 5 โอเปอเรชั่น เป็นต้น ในข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 6 เครื่อง

4.1.2 ชุดข้อมูลย่อย MK02 ประกอบด้วยงานหลักจำนวน 10 งาน แต่ละงานหลักประกอบด้วยโอเปอเรชั่นจำนวน 5 ถึง 6 โอเปอเรชั่น และมีจำนวนเครื่องจักรจำกัดที่ 6 เครื่อง

4.1.3 ชุดข้อมูลย่อย MK03 ประกอบด้วยงานหลักจำนวน 15 งาน แต่ละงานหลักประกอบด้วยจำนวนโอเปอเรชั่น 10 โอเปอเรชั่น ข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 8 เครื่อง

4.1.4 ชุดข้อมูลย่อย MK04 ประกอบด้วยงานหลักจำนวน 15 งาน แต่ละงานหลักประกอบด้วยจำนวนโอเปอเรชั่นที่แตกต่างกันตั้งแต่ 5 ถึง 9 โอเปอเรชั่น ในข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 8 เครื่อง

4.1.5 ชุดข้อมูลย่อย MK05 ประกอบด้วยงานหลักจำนวน 15 งาน แต่ละงานหลักประกอบด้วยจำนวนโอเปอเรชั่นที่แตกต่างกันตั้งแต่ 5 ถึง 9 โอเปอเรชั่น ในข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 4 เครื่อง

4.1.6 ชุดข้อมูลย่อย MK06 ประกอบด้วยงานหลักจำนวน 10 งาน แต่ละงานหลักประกอบด้วย 15 โอเปอเรชั่น ในข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 15 เครื่อง

4.1.7 ชุดข้อมูลย่อย MK07 ประกอบด้วยงานหลักจำนวน 20 งาน แต่ละงานหลักประกอบด้วยจำนวนโอเปอเรชั่น 5 โอเปอเรชั่น ในข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 5 เครื่อง

4.1.8 ชุดข้อมูลย่อย MK08 ประกอบด้วยงานหลักจำนวน 20 งาน แต่ละงานหลักประกอบด้วยจำนวนโอเปอเรชั่นที่แตกต่างกันตั้งแต่ 10 ถึง 14 โอเปอเรชั่น ในข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 10 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.1.9 ชุดข้อมูลย่อย MK09 ประกอบด้วยงานหลักจำนวน 20 งาน แต่ละงานหลักประกอบด้วยจำนวนโอเปอเรชั่นที่แตกต่างกันตั้งแต่ 10 ถึง 14 โอเปอเรชั่น ในข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 10 เครื่อง

4.1.10 ชุดข้อมูลย่อย MK10 ประกอบด้วยงานหลักจำนวน 20 งาน แต่ละงานหลักประกอบด้วยจำนวนโอเปอเรชั่นตั้งแต่ 10 ถึง 14 โอเปอเรชั่น ในข้อมูลชุดนี้มีจำนวนเครื่องจักรจำกัดที่ 15 เครื่อง

4.2 ผลการทดลอง

เมื่อทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูล Brandimarte พบว่าอัลกอริทึมที่นำเสนอสามารถค้นพบคำตอบที่เท่ากับขอบเขตล่าง 2 ชุดข้อมูล คือชุดข้อมูล MK03 และ MK08 ส่วนชุดข้อมูลอื่นๆ อัลกอริทึมนี้สามารถค้นพบคำตอบที่ใกล้เคียงกับขอบเขตล่าง ดังแสดงในตารางที่ 4.1 โดยคอลัมภ์ Data แสดงถึงชุดข้อมูลที่ใช้ทดลอง คอลัมภ์ J × M แสดงถึงจำนวนของงานหลัก (Job) และ M แสดงถึงจำนวนเครื่องจักรของชุดข้อมูลนั้นๆ โดยที่คอลัมภ์ Lower bound แสดงถึงคำตอบที่เป็นขอบเขตล่าง หรือกล่าวได้ว่าเป็นคำตอบที่เหมาะสมมากที่สุด คอลัมภ์ Result แสดงผลลัพธ์จากการทดลองมีหน่วยเป็นวัน (Day) สุดท้ายคือคอลัมภ์ %Dev from LB แสดงค่าความคาดเคลื่อนของคำตอบจากขอบเขตล่าง

ตารางที่ 4.1 ผลการทดลองอัลกอริทึมกับชุดข้อมูล Brandimarte

Data	J × M	Lower Bound (LB)	Result (Day)	%Dev from LB
MK01	10x6	36	40	11.1%
MK02	10x6	24	26	8.3%
MK03	15x8	204	204	0.0%
MK04	15x8	48	59	22.9%
MK05	15x4	168	172	2.3%
MK06	10x15	33	58	75.7%
MK07	20x5	133	140	5.2%
MK08	20x10	523	523	0.00%
MK09	20x10	299	307	2.6%
MK10	20x10	165	201	21.8%

ตารางที่ 4.1 แสดงผลการทดลองประสิทธิภาพของอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐาน 10 ชุด ซึ่งสามารถอธิบายได้ดังนี้

ชุดข้อมูล MK01 ประกอบด้วยจำนวนงานหลัก 10 งาน และจำนวนเครื่องจักร 6 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 36 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ดำเนินงานได้ 40 วัน มากกว่าขอบเขตล่างอยู่ 4 วัน คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อนจากขอบเขตล่างคือ 11.1%

ชุดข้อมูล MK02 ประกอบด้วยจำนวนงานหลัก 10 งาน และจำนวนเครื่องจักร 6 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 24 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 26 วัน มากกว่าขอบเขตล่างอยู่ 2 วัน คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อนจากขอบเขตล่างคือ 8.3%

ชุดข้อมูล MK03 ประกอบด้วยจำนวนงานหลัก 15 งาน และจำนวนเครื่องจักร 8 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 204 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 204 วัน ซึ่งไม่คาดเคลื่อนจากคำตอบขอบเขตล่าง

ชุดข้อมูล MK04 ประกอบด้วยจำนวนงานหลัก 15 งาน และจำนวนเครื่องจักร 8 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 48 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 59 วัน มากกว่าขอบเขตล่างอยู่ 11วัน คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อนจากขอบเขตล่างคือ 22.9%

ชุดข้อมูล MK05 ประกอบด้วยจำนวนงานหลัก 15 งาน และจำนวนเครื่องจักร 4 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 168 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 172 วัน มากกว่าขอบเขตล่างอยู่ 4 วัน คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อนจากขอบเขตล่างคือ 2.3%

ชุดข้อมูล MK06 ประกอบด้วยจำนวนงานหลัก 10 งาน และจำนวนเครื่องจักร 15 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 33 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 58 วัน มากกว่าขอบเขตล่างอยู่ 25 วัน คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อนจากขอบเขตล่างคือ 75.7%

ชุดข้อมูล MK07 ประกอบด้วยจำนวนงานหลัก 20 งาน และจำนวนเครื่องจักร 5 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 133 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 140 วัน มากกว่าขอบเขตล่างอยู่ 7 วัน คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อนจากขอบเขตล่างคือ 5.2%

ชุดข้อมูล MK08 ประกอบด้วยจำนวนงานหลัก 20 งาน และจำนวนเครื่องจักร 10 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 523 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 523 วัน ไม่คาดเคลื่อนจากขอบเขตล่าง

เอกสารนี้มีลิขสิทธิ์ สามารถใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ใช้เพื่อการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ชุดข้อมูล MK09 ประกอบด้วยจำนวนงานหลัก 20 งาน และจำนวนเครื่องจักร 10 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 299 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 307 วัน มากกว่าขอบเขตล่างอยู่ 8 วัน คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อนจากขอบเขตล่างคือ 2.6%

ชุดข้อมูล MK10 ประกอบด้วยจำนวนงานหลัก 20 งาน และจำนวนเครื่องจักร 10 เครื่อง มีระยะเวลาการดำเนินงานที่เป็นขอบเขตล่างคือ 165 วัน โดยอัลกอริทึมที่นำเสนอค้นพบเส้นทางที่คำนวณระยะเวลาการดำเนินงานได้ 201 วัน มากกว่าขอบเขตล่างอยู่ 36 วัน คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อนจากขอบเขตล่างคือ 21.8%

4.3 ผลการเปรียบเทียบประสิทธิภาพ

ตารางที่ 4.2 แสดงการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่นำเสนอกับอัลกอริทึมของงานวิจัยอื่นที่ทดลองกับข้อมูลชุดเดียวกัน โดยคอลัมภ์ Data แสดงถึงชุดข้อมูลที่นำมาทดสอบ และ Average %Dev from LB แสดงค่าเฉลี่ยความคาดเคลื่อนจากขอบเขตล่าง

ในงานวิจัยนี้ได้เปรียบเทียบอัลกอริทึมที่นำเสนอกับงานวิจัยอื่นๆ จำนวน 3 งาน คือ

1) งานวิจัยชื่อ An effective artificial bee colony algorithm for the flexible job-shop scheduling problem หรือ ABC

2) งานวิจัยชื่อ A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems หรือ KBACO

3) งานวิจัยชื่อ A heuristic algorithm for solving flexible job shop scheduling problem หรือ Heuristic

ตารางที่ 4.2 ผลการเปรียบเทียบอัลกอริทึมที่นำเสนอกับอัลกอริทึมอื่นๆ

Data	LB	Proposed		ABC [5]		KBACO [11]		Heuristic [12]	
		Result (Day)	%Dev from LB	Result (Day)	%Dev from LB	Result (Day)	%Dev from LB	Result (Day)	%Dev from LB
MK01	36	40	11.1%	40	11.1%	39	8.3%	42	16.7%
MK02	24	26	8.3%	26	8.3%	29	20.8%	28	16.7%
MK03	204	204	0.0%	204	0.0%	204	0.0%	204	0.0%
MK04	48	59	22.9%	60	25.0%	65	35.4%	75	56.3%
MK05	168	172	2.3%	172	2.4%	173	3.0%	179	6.5%
MK06	33	58	75.7%	60	81.8%	67	103.0%	69	109.1%
MK07	133	140	5.2%	139	4.5%	144	8.3%	149	12.0%
MK08	523	523	0.00%	523	0.0%	523	0.0%	555	6.1%
MK09	299	307	2.6%	307	2.7%	311	4.0%	342	14.4%
MK10	165	201	21.8%	208	26.1%	229	38.8%	242	46.7%
Average %Dev from LB		**15.03**		**16.19**		**22.17**		**28.44**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

จากตารางที่ 4.2 พบว่า อัลกอริทึมที่นำเสนอดีกว่าอัลกอริทึมอื่นที่นำมาเปรียบเทียบในชุดข้อมูล MK04, MK06 และ MK10 เนื่องจากได้คำตอบที่คาดเคลื่อนจากขอบเขตล่างน้อยที่สุดคือ 22.9%, 75.7% และ 21.8% ตามลำดับ นอกจากนั้นอัลกอริทึมที่นำเสนอยังพบคำตอบที่เหมาะสมเช่นเดียวกับอัลกอริทึม ABC ในชุดข้อมูล MK02, MK05 และ MK09 คำตอบที่พบคือ 26, 172 และ 307 ตามลำดับ ในขณะที่อัลกอริทึม KBACO ให้คำตอบที่ดีที่สุดในชุดข้อมูล MK01 และอัลกอริทึม ABC ให้คำตอบที่ดีที่สุดในชุดข้อมูล MK07 ทั้งนี้ทุกอัลกอริทึมสามารถค้นพบคำตอบเท่ากับคำตอบขอบเขตล่างในชุดข้อมูล MK03 และทุกอัลกอริทึมสามารถค้นพบคำตอบเท่ากับคำตอบขอบเขตล่างสำหรับชุดข้อมูล MK08 ยกเว้นอัลกอริทึม Heuristic

ดังนั้นเมื่อพิจารณาค่าเฉลี่ยความคาดเคลื่อนจากขอบเขตล่าง จะพบว่าอัลกอริทึมที่นำเสนอให้ผลลัพธ์ที่เหมาะสมมากกว่างานวิจัยอื่นๆ ที่นำมาเปรียบเทียบ

King Mongkut's Institute of Technology Ladkrabang

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

งานวิจัยนี้นำเสนอการปรับปรุงประสิทธิภาพของอัลกอริทึมการผสมพันธุ์ของผึ้ง (Marriage in honey bees optimization: MBO) โดยนำเสนอวิธีในการเพิ่มความหลากหลายให้กับอัลกอริทึม MBO โดยการแบ่งโซลูชั่นลูกออกเป็น 2 กลุ่มคือ โซลูชั่นลูกเพศผู้ที่เกิดจากกระบวนการโคลนนิ่งและโซลูชั่นลูกเพศเมียที่เกิดจากกระบวนการครอสโอเวอร์ และเพิ่มความสามารถในการค้นหาคำตอบบริเวณ local โดยกำหนดให้โซลูชั่นลูกเพศเมียมีหน้าที่หาอาหารตามขั้นตอนของอัลกอริทึมการหาแหล่งอาหารของผึ้ง (Artificial bee colony algorithm: ABC) โซลูชั่นลูกเพศเมียที่สามารถค้นพบแหล่งอาหารได้ดีที่สุดจะสามารถให้กำเนิดโซลูชั่นลูกเพศผู้ได้ สุดท้ายคือการสุ่มสร้างโซลูชั่นผึ้งจากรังอื่นขึ้นมาเพื่อป้องกันไม่ให้อัลกอริทึม MBO ลู่เข้าสู่คำตอบที่ยังไม่เหมาะสมเร็วเกินไป

ในงานวิจัยนี้จะทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐานชื่อ Brandimarte ซึ่งประกอบด้วยชุดข้อมูลย่อยทั้งหมด 10 ชุดข้อมูลคือ ชุดข้อมูล MK01 ถึง MK10 ซึ่งผลการวัดประสิทธิภาพของอัลกอริทึมที่นำเสนอกับชุดข้อมูลที่นำมาทดสอบพบว่า อัลกอริทึมทีนำเสนอสามารถพบคำตอบที่ใกล้เคียงกับขอบเขตล่างมาก

โดยเมื่อนำค่าเฉลี่ยความคลาดเคลื่อนจากขอบเขตล่างของงานวิจัยนี้ไปเปรียบเทียบกับค่าเฉลี่ยความคลาดเคลื่อนจากขอบเขตล่างของงานวิจัยอื่นๆ พบว่า งานวิจัยที่นำเสนอคาดเคลื่อนจากขอบเขตล่างน้อยที่สุด ดังนั้นจึงสรุปได้ว่าอัลกอริทึมที่งานวิจัยนี้นำเสนอมีประสิทธิภาพสามารถค้นพบคำตอบที่มีความเหมาะสมได้

5.2 ข้อเสนอแนะ

อัลกอริทึมที่นำเสนอมีเป้าหมายเพื่อค้นหาคำตอบที่มีความเหมาะสม โดยไม่ได้ให้ความสำคัญกับเวลาในการประมวลผลข้อมูล ดังนั้นในขั้นต่อไป เพื่อเพิ่มประสิทธิภาพของอัลกอริทึมที่นำเสนอ จึงควรทำการปรับปรุงอัลกอริทึมที่นำเสนอให้ประมวลผลได้เร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

เอกสารอ้างอิง

[1] Frank Neumann and Carsten Witt, Bioinspired Computation in Combinatorial Optimization Algorithm and Their Computational Complexity, Springer, 2010.

[2] Geir Hasle, Discrete Optimization – Heuristics, University of Jyväskylä, Finland, 2009.

[3] James D. Ellis and Ashley N. Mortensen, "Cape honey bee," Featured Creatures, http://entnemdept.ufl.edu/creatures/misc/bees/cape_honey_bee.htm

[4] Robin F. A. Moritz, H. Michael G. Lattorff, Kendall L. Crous, and Randall H. Hepburn, "Social parasitism of queens and workers in the Cape honeybee (Apis mellifera capensis)," Behav Ecol Sociobiol, 65, pp. 735-740, 2011.

[5] Ling Wang, Gang Zhou, Ye Xu, Shengyao Wang and Min Liu, "An effective artificial bee colony algorithm for the flexible job-shop scheduling problem," Int J Adv Manuf Technol, 60, pp. 303–315, 2012.

[6] Junqing Li, Quanke Pan, and Shengxian Xie, "An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems," Applied Mathematics and Computation, 218, pp. 9353–9371, 2012.

[7] Hamid Davoudpour and Nader Azad, "Solving multi-objective flexible job shop scheduling problems using immune algorithm," International Journal of Modern Science and Technology, pp. 1-12, 2012.

[8] Xinyu Shao, Weiqi Liu, Qiong Liu, and Chaoyong Zhang, "Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem," Int J Adv Manuf Technol, 67, pp. 2885–2901, 2013.

[9] Seyed Habib A. Rahmati and M. Zandieh, "A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem," Int J Adv Manuf Technol, 58, pp. 1115–1129, 2012.

[10] Monaldo Mastrolilli, Flexible Job Shop Problem, http://www.idsia.ch/~monaldo/fjsp.html

[11] Li-Ning Xing, Ying-Wu Chen, Peng Wang, Qing-Song Zhao, and Jian Xiong, "A knowledge-based ant colony optimization for flexible job shop scheduling problems," Applied Soft Computing, 10, pp. 888-896, 2010.

[12] Mohsen Ziaee, "A heuristic algorithm for solving flexible job shop scheduling problem," Int J Adv Manuf Technol, 71, pp. 519-528, 2014.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.