

การปรับปรุงจีเนติกอัลกอริทึม
สำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น

A MODIFIED GENETIC ALGORITHM
FOR FLEXIBLE JOB-SHOP SCHEDULING PROBLEMS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2558

KMITL-2015-IT-D-001-001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**A MODIFIED GENETIC ALGORITHM
FOR FLEXIBLE JOB-SHOP SCHEDULING PROBLEMS**

WANNAPORN TEEKENG

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2015

KMITL-2015-IT-D-001-001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2015

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การปรับปรุงขั้นตอนวิธีอัลกอริทึม สำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น
A modified genetic algorithm for flexible job-shop scheduling problems
นักศึกษา นางสาววรรณพร ทีเก่ง
รหัสประจำตัว 51612651
ปริญญา ปรัชญาคณะศึกษาศาสตร์
สาขาวิชา เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ รองศาสตราจารย์ ดร.อาริต ธรรมโน

คณะกรรมการสอบวิทยานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์ ดร.วราพงษ์ ศรีสุระเดช	
รองศาสตราจารย์ ดร.ไพโรจน์ ไสยสสถิตย์	
รองศาสตราจารย์ ดร.อาริต ธรรมโน	
รองศาสตราจารย์ ดร.พรฤดี เบตต์โสภากุล	
ผู้ช่วยศาสตราจารย์ ดร.กิติ์สุชาติ พงษ์ภา	

วัน / เดือน / ปี ที่สอบ วันพุธที่ 11 มีนาคม 2558 เวลา 09.30 น.

สถานที่สอบ ณ ห้อง 334 ชั้น 3 คณะเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศรับรองแล้ว

(รองศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์)

คณบดีคณะเทคโนโลยีสารสนเทศ

วันที่... 1 ...เดือน... พฤษภาคม พ.ศ. 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การปรับปรุงจีเนติกอัลกอริทึม สำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น
นักศึกษา	นางสาววรรณพร ทีแก่ง
รหัสนักศึกษา	51612651
ปริญญา	ปรัชญาคุษฎีบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2558
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร. อาริต ธรรมโน

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอการปรับปรุงจีเนติกอัลกอริทึมแบบดั้งเดิมสำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น โดยปรับปรุงส่วนกระบวนการทางพันธุกรรม ได้แก่ 1) การคัดเลือกโครโมโซมด้วยวิธีการวงล้อรูเล็ตแบบฟัซซี (Fuzzy Roulette wheel Selection) 2) การครอสโอเวอร์โดยการแบ่งกลุ่มโครโมโซมด้วยวิธี Cluster-Crossover operator ด้วยค่าความคล้ายของโครโมโซมแล้วทำการครอสโอเวอร์ และ 3) การมิวเตชันแบบโลคอล (Mutation-Local Search operator) เพื่อสร้างความหลากหลายของชุดคำตอบและนำไปสู่คำตอบที่เหมาะสมที่สุด

การประเมินประสิทธิภาพอัลกอริทึมที่นำเสนอด้วยทดสอบกับชุดข้อมูลมาตรฐาน จำนวน 3 ชุดข้อมูล และเปรียบเทียบกับอัลกอริทึมอื่น ผลการทดลองของอัลกอริทึมที่นำเสนอมีประสิทธิภาพในการค้นหาคำตอบที่เหมาะสมที่สุดสูงกว่าอัลกอริทึมที่นำมาเปรียบเทียบกับ

Thesis Title	A modified genetic algorithm for flexible job-shop scheduling problems
Student	Miss Wannaporn Teekeng
Student ID.	51612651
Degree	Doctor of Philosophy
Programme	Information Technology
Year	2015
Thesis Advisor	Assoc. Prof. Dr. Arit Thammano

ABSTRACT

This thesis proposes a modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems. The proposed algorithm is improved the genetic process. Firstly, the Fuzzy Roulette wheel Selection is used for the original Selection. Secondly, the Cluster-Crossover operator, the parent chromosomes are divided into two clusters with the similarity value then to the crossover operator. Thirdly, the Mutation-LS operator, the variety methods of the mutation are improved chromosomes to the optimal solution.

The proposed algorithm has been evaluated by three benchmark problems and compared the results with the other algorithms. The experimental results show that the proposed model is very effective in solving the optimization problems as compared with the other algorithms.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ.ดร.อาริต ธรรมโน ที่ให้ความรู้ ให้คำชี้แนะในการคิดและการแก้ไขปัญหา ให้ความช่วยเหลือตลอดจนให้กำลังใจที่ดีแก่ข้าพเจ้าตลอดมา

ขอกราบขอบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ประสิทธิ์ประสาทวิชาให้แก่ข้าพเจ้า

ขอขอบพระคุณกระทรวงวิทยาศาสตร์และเทคโนโลยีที่ได้มอบทุนรัฐบาลกระทรวงวิทยาศาสตร์และเทคโนโลยีให้แก่ข้าพเจ้าและขอขอบคุณเจ้าหน้าที่งานนักเรียนทุน สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติที่ให้การสนับสนุนและดูแลเอาใจใส่เป็นอย่างดีแก่ข้าพเจ้า

ขอขอบคุณเพื่อน พี่น้องทุกคนที่คอยให้กำลังใจแก่ข้าพเจ้ารวมทั้งเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และเจ้าหน้าที่มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ทุกท่านที่คอยประสานงานให้งานสำเร็จลุล่วงไปได้ด้วยดี
สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดาและขอขอบคุณครอบครัวที่คอยให้ความรัก ให้คำปลอบโยน ให้กำลังใจที่อบอุ่นแก่ข้าพเจ้า

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์นี้ ข้าพเจ้าขออุทิศเพื่อบูชาพระคุณบิดา มารดา อันเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่าน

วรรณพร ทีแก่ง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	2
1.3 ขอบเขตของการวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนของการดำเนินงานวิจัย	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ปัญหาการจัดตารางการผลิต	4
2.1.1 ลักษณะของปัญหาการจัดตารางการผลิต	4
2.1.2 ลักษณะของเครื่องจักรในการผลิต	5
2.1.3 การสร้างชุดคำตอบสำหรับปัญหาการจัดตารางการผลิต	7
2.1.4 เครื่องมือในการวัดประสิทธิภาพการจัดตารางการผลิต	8
2.2 ทฤษฎีที่เกี่ยวข้อง	9
2.2.1 การจัดกลุ่มแบบลำดับชั้น	9
2.2.2 จีเนติกอัลกอริทึม	10
2.2.3 ฟิชชีเซต	11
2.3 งานวิจัยที่เกี่ยวข้อง	12
2.3.1 A Simulated Annealing Algorithm for Flexible Job-Shop Scheduling Problem	12
2.3.2 An efficient Job-Shop Scheduling Algorithm base on Particle Swarm Optimization	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.3 An effective hybrid Particle Swarm Optimization Algorithm for multi-objective Flexible Job-Shop Scheduling Problem.....	20
2.3.4 Knowledge-Based Ant Colony Optimization for Flexible Job-Shop Scheduling Problem.....	23
2.3.5 An Artificial Immune Algorithm for the Flexible Job-Shop Scheduling Problem.....	27
2.3.6 An improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling Problem.....	30
2.3.7 An effective Hybridized Genetic Algorithm architecture for the Flexible Job-shop Scheduling Problem.....	32
2.3.8 An effective Genetic Algorithm for the Flexible Job-Shop Scheduling Problem.....	35
2.3.9 An efficient knowledge-based Algorithm for the Flexible Job-Shop Scheduling Problem.....	37
2.3.10 A new Biogeography-Based Optimization (BBO) Algorithm for the Flexible Job-Shop Scheduling problem.....	41
บทที่ 3 การปรับปรุงจีเนติกอัลกอริทึมสำหรับการจัดการตารางการผลิตแบบตามสั่งแบบยืดหยุ่น.....	46
3.1 การวิเคราะห์ปัญหาของจีเนติกอัลกอริทึมแบบดั้งเดิม.....	46
3.2 แนวทางการปรับปรุงจีเนติกอัลกอริทึมแบบดั้งเดิม.....	47
3.2.1 กระบวนการคัดเลือก.....	47
3.2.2 กระบวนการครอสโอเวอร์.....	49
3.2.3 กระบวนการมิวเตชัน.....	49
3.3 โครงสร้างการทำงานของอัลทึมที่นำเสนอ.....	51
3.4 อัลกอริทึมที่นำเสนอสำหรับปัญหาการจัดการตารางการผลิตแบบตามสั่งแบบยืดหยุ่น.....	53
3.4.1 การแทนค่าโครโมโซมเริ่มต้น.....	53
3.4.2 การคำนวณค่าฟิตเนส.....	54
3.4.3 การคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟิชชี.....	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และเผยแพร่ไปยังผู้อื่นโดยไม่ได้รับอนุญาต

สารบัญ (ต่อ)

	หน้า
3.4.4 การครอสโอเวอร์.....	58
3.4.5 การมีวเดชั่น.....	60
3.4.6 การสืบพันธุ์และสิ้นสุดการทำงาน.....	62
บทที่ 4 การทดสอบการปรับปรุงจีเนติกอัลกอริทึมสำหรับการจัดการการผลิต	
แบบตามสิ่งยึดหยุ่น.....	63
4.1 ข้อมูลที่ใช้ในการทดสอบ.....	64
4.2 เครื่องมือที่ใช้ในการวัดประสิทธิภาพ.....	66
4.3 การทดสอบเพื่อศึกษาปัจจัยที่ส่งผลต่ออัลกอริทึมที่นำเสนอ.....	67
4.3.1 การกำหนดค่าพารามิเตอร์.....	68
4.3.2 ผลการทดลอง.....	68
4.3.3 สรุปผลการทดลอง.....	74
4.4 การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Brandimarte's (BRdata) เปรียบเทียบกับผล การทดลองในงานวิจัยอื่น.....	80
4.4.1 การกำหนดค่าพารามิเตอร์.....	80
4.4.2 ผลการทดลอง.....	81
4.4.3 สรุปผลการทดลอง.....	81
4.5 การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Fattahi et al. (Fdata) เปรียบเทียบกับผล การทดลองในงานวิจัยอื่น.....	83
4.5.1 การกำหนดค่าพารามิเตอร์.....	83
4.5.2 ผลการทดลอง.....	83
4.5.3 สรุปผลการทดลอง.....	85
4.6 การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Barnes and Chamber library (BCdata) เปรียบเทียบกับผลการทดลองในงานวิจัยอื่น.....	86
4.6.1 การกำหนดค่าพารามิเตอร์.....	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.6.2 ผลการทดลอง.....	87
4.6.3 สรุปผลการทดลอง.....	89
4.7 สรุปผลการทดสอบอัลกอริทึมที่นำเสนอ.....	90
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	93
5.1 สรุปงานวิจัย.....	93
5.2 สรุปผลการทดสอบ.....	94
5.3 จุดเด่นของงานวิจัย.....	95
5.4 จุดด้อยของงานวิจัย.....	95
5.5 ปัญหาที่พบในงานวิจัย.....	95
5.6 แนวทางในการพัฒนาต่อไปในอนาคต.....	95
เอกสารอ้างอิง.....	97
ภาคผนวก.....	100
ภาคผนวก ก ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่.....	101
ประวัติผู้เขียน.....	137

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และตั้งชื่ออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างของปัญหาขนาด 4 งาน 4 เครื่องจักร	12
2.2 ค่าพารามิเตอร์ที่ใช้ในการทดสอบ	16
2.3 ชุดข้อมูลทดสอบและค่าพารามิเตอร์ที่ใช้ในการทดลอง	35
2.4 นิยามตามหลักการของอัลกอริทึมการกระจายของสิ่งมีชีวิตตามภูมิศาสตร์และ จินตนิมิตอัลกอริทึม	42
3.1 การเปรียบเทียบการคำตอบด้วยวิธีซิมูเลชันเทคนิคแอนนิลลิงอัลกอริทึมและการค้นหา แบบตาบอด	50
3.2 ตัวอย่างการคำนวณค่าฟิตเนสของโครโมโซมในวงล้อรูเล็ตแบบฟิชชี	55
4.1 ชุดข้อมูล BRdata	64
4.2 ชุดข้อมูล Fdata	65
4.3 ชุดข้อมูล BCdata	66
4.4 ค่าพารามิเตอร์เปรียบเทียบปัจจัยที่มีผลต่อประสิทธิภาพของอัลกอริทึม	68
4.5 ค่าพารามิเตอร์ของอัลกอริทึมที่นำเสนอสำหรับชุดข้อมูล BRdata	81
4.6 การเปรียบเทียบผลการทดลองสำหรับข้อมูล BRdata ระหว่างอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึมอื่น	82
4.7 ค่าพารามิเตอร์ของอัลกอริทึมที่นำเสนอสำหรับชุดข้อมูล Fdata	83
4.8 การเปรียบเทียบผลการทดลองสำหรับข้อมูล Fdata ระหว่างอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึมอื่น	84
4.9 ค่าพารามิเตอร์ของอัลกอริทึมที่นำเสนอสำหรับชุดข้อมูล BCdata	86
4.10 การเปรียบเทียบผลการทดลองสำหรับข้อมูล Fdata ระหว่างอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึมอื่น	87
4.11 ผลการทดลองเปรียบเทียบค่าคลาดเคลื่อนสัมพัทธ์ของ 8 อัลกอริทึม	91

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างของปัญหาแบบ JSSP	6
2.2 ตัวอย่างของปัญหาแบบ FSSP	7
2.3 ตัวอย่างการสร้างชุดคำตอบของปัญหาแบบ FJSP	8
2.4 แผนภูมิแกนต์ของชุดคำตอบ	8
2.5 อัลกอริทึมพื้นฐานของวิธีการรวมกลุ่มเป็นลำดับชั้น	9
2.6 แผนภาพการรวมกลุ่มเป็นลำดับชั้น	9
2.7 ตัวแทนชุดคำตอบ	13
2.8 การเข้ารหัสชุดคำตอบ A-String	21
2.9 การเข้ารหัสชุดคำตอบ B-String	21
2.10 รูปแบบของคำตอบ	26
2.11 ตัวแทนของแอนติบอดี	28
2.12 การเข้ารหัสชุดคำตอบ โครโมโซม	31
2.13 ลักษณะของโครโมโซม	36
2.14 โครงสร้างตัวแทนชุดคำตอบ	38
2.15 Operation Memory	39
2.16 การเปลี่ยนแปลงของอัตราการอพยพเข้า และอพยพออก	41
2.17 ชุดคำตอบสำหรับปัญหา 3 งาน 4 เครื่องจักรและ 8 โอเปอร์เรชัน	42
2.18 เซตชุดตัวแปรดัชนีความเหมาะสม (ฟิต) ของชุดคำตอบ	43
3.1 โครงสร้างการทำงานของอัลกอริทึมที่นำเสนอ	52
3.2 ตัวอย่างปัญหาการผลิตแบบตามงานแบบยืดหยุ่น ประเภท 2 งาน 3 เครื่องจักร	53
3.3 ตัวอย่างลักษณะโครโมโซมเริ่มต้น	53
3.4 ส่วนประกอบของโครโมโซม	54
3.5 การประเมินการจัดตารางการผลิตในหนึ่งโครโมโซม	54
3.6 กำหนดตัวแทนชุดคำตอบ (โครโมโซม)	55
3.7 กระบวนการจัดลำดับโครโมโซม	56
3.8 เมตริกซ์ค่าความคล้ายของโครโมโซม	57
3.9 ตัวอย่างโครโมโซมในวงล้อรูเล็ตแบบฟิชชี	57
3.10 ตัวอย่างโครโมโซมในวิธีวงล้อรูเล็ตแบบดั้งเดิม	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และดัดแปลงอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 ตัวอย่างการเลือกโครโมโซมด้วยค่าความเป็นสมาชิก.....	58
3.12 ตัวอย่างการแบ่งกลุ่มโครโมโซมต้นแบบ.....	59
3.13 ตัวอย่างการครอสโอเวอร์.....	60
3.14 การมีวเตชันแบบ MU1.....	61
3.15 การมีวเตชันแบบ MU2.....	61
3.16 การมีวเตชันแบบ MU3.....	61
3.17 การมีวเตชันแบบ MU4.....	62
3.18 กระบวนการมีวเตชันแบบโคลอด.....	62
4.1 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK01.....	69
4.2 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK02.....	69
4.3 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK03.....	70
4.4 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK04.....	70
4.5 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK05.....	71
4.6 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK06.....	71
4.7 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK07.....	72
4.8 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK08.....	72
4.9 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK09.....	73
4.10 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK10.....	73
4.11 กราฟแสดงการเปรียบเทียบผลการทดลองที่ระหว่างอัลกอริทึม IGA GA และ อัลกอริทึม BBO สำหรับชุดข้อมูล BCdata.....	89

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การนำหลักการของปัญหาประคิษฐ์มาใช้ในการสร้างฉลาดให้กับคอมพิวเตอร์ให้มีการเรียนรู้เกี่ยวกับกระบวนการคิด การกระทำ การใช้เหตุผล และการทำงานเหมือนกับสมองมนุษย์ เพื่อใช้ในการตัดสินใจของการแก้ปัญหาได้อย่างถูกต้อง จึงเป็นสิ่งที่ได้รับความสนใจ และมีการพัฒนาอย่างต่อเนื่องโดยนำมาประยุกต์ใช้กับงานด้านต่างๆ รวมทั้งงานทางด้านการวิจัยการดำเนินงาน (Operation Research) ซึ่งนิยมนำเอาหลักการของปัญหาประคิษฐ์มาใช้ในการแก้ปัญหา เช่น ปัญหาการจัดตารางการผลิต (Scheduling Problem) ปัญหาการมอบหมายงาน (Assignment Problem) และปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem) เป็นต้น ซึ่งล้วนแต่เป็นปัญหาที่ต้องการหาคำตอบที่อยู่ภายใต้ข้อจำกัดของทรัพยากร เช่น ค่าต่ำสุด ค่าสูงสุด และค่าที่เหมาะสมที่สุด เป็นต้น ในส่วนของปัญหาการจัดลำดับงานหรือปัญหาการจัดตารางการผลิตเป็นปัญหาประเภท NP-Hard แบบ Combinatorial Optimization หมายถึง ปัญหาที่ใช้เวลาหาคำตอบยาวนาน และเมื่อขนาดของปัญหาเพิ่มขึ้นแต่ต้องใช้เวลาในการแก้ปัญหามากขึ้นเป็นแบบเอ็กซ์โพเนนเชียล (Exponential) การแก้ปัญหขนาดเล็กละสามารถใช้หลักการคณิตศาสตร์ในการแก้ปัญหา เช่น การใช้วิธีโปรแกรมเชิงเส้น (Integer Linear Programming) หรือวิธีแตกกิ่งและขอบเขต (Branch and Bound) เป็นต้น ส่วนปัญหาที่มีขนาดใหญ่ และมีความซับซ้อนจะมีการใช้หลักการประมาณ (Approximation Optimization Algorithm) เพื่อหาคำตอบที่เหมาะสมที่สุด

วิธีการเมตา-ฮิวริสติก (Meta-heuristics) เป็นวิธีการหนึ่งที่สามารถหาคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณ และมีความรวดเร็วในการประมวลผลของการหาคำตอบที่ซับซ้อนได้ดี ตัวอย่างอัลกอริทึมของวิธีการนี้คือ จินเนติกอัลกอริทึม ซึ่งเป็นอัลกอริทึมที่เลียนแบบวิวัฒนาการทางพันธุกรรมของสิ่งมีชีวิต ซึ่งเป็นที่ยอมรับในประสิทธิภาพการทำงานที่สามารถหาค่าเหมาะสมที่สุดภายในเวลาที่กำหนด นอกจากนี้ยังมีอัลกอริทึมการเคลื่อนที่ของกลุ่มอนุภาค (Particle Swarm Optimization: PSO) เป็นอัลกอริทึมที่เลียนแบบการหาอาหารของฝูงนกหรือฝูงปลา, อัลกอริทึมอาณานิคมมด (Ant Colony Optimization: ACO) เป็นอัลกอริทึมที่เลียนแบบการหาอาหารของมด รวมทั้งในกลุ่มของการค้นหาแบบโลคอล (Local search) เช่น การค้นหาแบบตาบอด (Tabu Search: TS) เป็นการค้นหาแบบมีข้อห้าม และอัลกอริทึมซิมูเลทเทคแอนนิลลิง (Simulated Annealing: SA) เป็นการค้นหาคำตอบจากแนวคิดการใช้อุณหภูมิเพื่อหลอมโลหะให้อยู่ในสถานะที่เหมาะสมที่สุด เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาในครั้งนี้จะมุ่งเน้นการปรับปรุงขั้นตอนการทำงานของจินตริกอัลกอริทึมแบบดั้งเดิมในการค้นหาคำตอบที่เหมาะสมที่สุดสำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น (Flexible Job-shop Scheduling Problems) โดยปรับปรุงในส่วนกระบวนการทางพันธุกรรม ได้แก่ 1) การคัดเลือกประชากรต้นแบบ โดยนำเสนอ “วงล้อรูเล็ตแบบฟัซซี” (Fuzzy Roulette Wheel Selection) แทนวิธีวงล้อรูเล็ตแบบเดิมซึ่งจะพิจารณาความคล้ายระหว่างโครโมโซมสร้างวงล้อที่เหมาะสมกับลักษณะของโครโมโซม เพื่อเลือกประชากรต้นแบบ 2) การแบ่งกลุ่มครอสโอเวอร์ (Cluster-Crossover operator) กำหนดให้แบ่งโครโมโซมต้นแบบเป็น 2 กลุ่มโดยพิจารณาจากโครโมโซมที่มีความคล้ายกันอยู่กลุ่มเดียวกัน จากนั้นจึงจับคู่โครโมโซมเพื่อผสมพันธุ์ และ 3) การมิวเตชันแบบโลคอล (Mutation-LS operator) เพื่อสร้างความหลากหลายของชุดคำตอบโดยนำเสนอการแบ่งวิธีการมิวเตชัน 4 วิธี ได้แก่ MU1, MU2, MU3 และ MU4 ตามลำดับ การทดสอบประสิทธิภาพของอัลกอริทึมที่นำเสนอโดยทดสอบกับชุดข้อมูลมาตรฐาน และเปรียบเทียบผลการทดลองกับอัลกอริทึมจากงานวิจัยอื่น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อวิเคราะห์หาจุดเด่นและจุดด้อยของจินตริกอัลกอริทึมแบบดั้งเดิมสำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น
2. เพื่อวิเคราะห์ปัจจัยที่มีผลต่อกระบวนการทางพันธุกรรม ได้แก่ การคัดเลือก การครอสโอเวอร์ และการมิวเตชัน
3. เพื่อปรับปรุงและพัฒนาจินตริกอัลกอริทึมสำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่นให้มีประสิทธิภาพ

1.3 ขอบเขตของงานวิจัย

1. งานวิจัยนี้นำเสนออัลกอริทึมที่เกิดจากการปรับปรุงกระบวนการทางพันธุกรรมของจินตริกอัลกอริทึมแบบดั้งเดิมสำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น
2. งานวิจัยนี้ประเมินประสิทธิภาพโดยกำหนดฟังก์ชันวัตถุประสงค์ คือ เวลาในการผลิตรวมน้อยที่สุด (Minimize Makespan)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้ความเข้าใจอัลกอริทึมทางเมตา-ฮิวริสติก
2. มีความรู้ความเข้าใจเกี่ยวกับเครื่องมือที่ใช้วัดประสิทธิภาพการทำงานของอัลกอริทึมที่ใช้สำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ได้อัลกอริทึมที่มีประสิทธิภาพสำหรับปัญหาการจัดการตารางการผลิตแบบตามสั่งยืดหยุ่น
4. ได้ต้นแบบงานวิจัยที่เป็นต้นแบบสำหรับการศึกษาและพัฒนางานและอัลกอริทึมทางด้านปัญญาประดิษฐ์

1.5 ขั้นตอนของการดำเนินงานวิจัย

1. ศึกษางานวิจัยที่นำเสนออัลกอริทึมทางเมตา-ฮิวริสติก เพื่อใช้ในการแก้ปัญหาการจัดการตารางการผลิตแบบต่างๆ ปัญหาออฟติไมเซชัน และปัญหาอื่นๆ ที่เกี่ยวข้อง
2. ศึกษาทฤษฎีที่เกี่ยวข้องกับการพัฒนาอัลกอริทึมสำหรับแก้ปัญหาการจัดการตารางการผลิต เช่น ลักษณะของการจัดการตารางการผลิต เครื่องมือที่ใช้การวัดประสิทธิภาพของอัลกอริทึม และลักษณะของข้อมูลที่ใช้ในการทดสอบ เป็นต้น
3. วางแผนและออกแบบอัลกอริทึมใหม่
4. พัฒนาโปรแกรมตามรูปแบบอัลกอริทึมที่ได้ออกแบบไว้ โดยใช้โปรแกรม MATLAB เวอร์ชัน 7.6.0.324
5. ออกแบบการทดลองและจัดเตรียมข้อมูลทดสอบเพื่อใช้ในการวัดประสิทธิภาพอัลกอริทึมที่นำเสนอ จากนั้นทำการทดลองกับข้อมูลที่ใช้สำหรับทดสอบ
6. สรุปผลการทดลอง
7. จัดทำรายงานการวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะแบ่งเนื้อหาออกเป็น 3 ส่วน โดยส่วนแรกกล่าวถึงปัญหาการจัดตารางการผลิต ส่วนที่ 2 อธิบายถึงทฤษฎีพื้นฐานที่เกี่ยวข้องกับงานวิจัยนี้ และส่วนที่ 3 เป็นรายละเอียดของงานวิจัยที่เกี่ยวข้อง

2.1 ปัญหาการจัดตารางการผลิต

ในส่วนนี้จะอธิบายเกี่ยวกับลักษณะของปัญหาการจัดตารางการผลิต ลักษณะของเครื่องจักรในการผลิต และเครื่องมือในการวัดประสิทธิภาพการจัดตารางการผลิต โดยมีรายละเอียดดังนี้

2.1.1 ลักษณะของปัญหาการจัดตารางการผลิต

ในอุตสาหกรรมการผลิต เมื่อโรงงานมีรายการสั่งซื้อจากลูกค้า โรงงานจะดำเนินการวางแผนและกำหนดตารางการผลิตโดยมีวัตถุประสงค์เพื่อให้การผลิตแล้วเสร็จตามที่ได้วางแผน ซึ่งแต่ละโรงงานจะมีข้อจำกัดตามลักษณะกระบวนการผลิตที่แตกต่างกัน เช่น ประเภทงาน จำนวนเครื่องจักร, ประสิทธิภาพของเครื่องจักร และบุคลากร เป็นต้น ในด้านของประสิทธิภาพของการผลิตจะต้องใช้เวลาทำงานน้อยที่สุด และใช้เครื่องจักรในกระบวนการผลิตอย่างคุ้มค่าที่สุด ซึ่งการกำหนดตารางการผลิตจะต้องกำหนดวันและเวลาการเริ่มต้นและวันสิ้นสุดของงาน ซึ่งจะต้องสอดคล้องกับลำดับความสำคัญและเงื่อนไขของแต่ละงานในกระบวนการผลิตให้สามารถใช้เครื่องจักรให้มีประสิทธิภาพสูงสุดภายใต้ข้อจำกัดตามลักษณะของงาน ดังนั้นการจัดตารางการผลิต (Scheduling) จึงเป็นปัจจัยที่สำคัญและเป็นสิ่งที่ท้าทายสำหรับการวางแผนการจัดลำดับการผลิต (Sequencing and Scheduling) ซึ่งการแก้ปัญหานี้เบื้องต้นของการจัดตารางการผลิตโดยทั่วไปต้องอาศัยความชำนาญและเชี่ยวชาญของบุคลากรฝ่ายวางแผนการผลิตในการจัดลำดับการผลิตให้มีประสิทธิภาพสูงสุดและเหมาะสมกับทรัพยากร หากระบบงานมีขนาดใหญ่และมีความซับซ้อนมากหรือเครื่องจักรมีประสิทธิภาพที่หลากหลาย ก็จะมีปัญหามากยิ่งขึ้นทำให้การค้นหาคำตอบของวิธีการจัดตารางการผลิตอาจจะต้องใช้เวลานานในการคำนวณหาคำตอบ หรือบางปัญหาอาจไม่สามารถหาคำตอบได้เนื่องจากใช้เวลานานเกินไป ดังนั้นปัญหาการจัดตารางการผลิต (Sequencing Problem) ถือว่าเป็นปัญหาประเภทออปติไมเซชัน (Optimization Problem) คือ การหาจุดที่เหมาะสมที่สุดตามฟังก์ชันเป้าหมาย (Objective Function) มีค่าดีที่สุดในภายใต้ข้อจำกัด (Constraint) ของปัญหานั้นๆ ซึ่งค่าฟังก์ชันเป้าหมายที่ดีที่สุดอาจเป็นค่าต่ำสุดหรือค่าสูงสุดขึ้นอยู่กับปัญหาที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศึกษา [1] จึงทำให้ปัญหาดังกล่าวเป็นสิ่งที่ได้รับความสนใจเป็นอย่างมากทั้งในทางทฤษฎีและทางปฏิบัติ เนื่องจากสามารถนำแนวคิดดังกล่าวมาใช้แก้ปัญหาที่เกิดขึ้นในอุตสาหกรรมการผลิตได้จริง คำจำกัดความในการจัดตารางการผลิตในงานวิจัยนี้ประกอบด้วย

- งาน (Job): หมายถึงงานในตารางการผลิต กำหนดให้มีจำนวน n งาน ซึ่ง $J_i, i \in \{1, 2, \dots, n\}$
- โอเปอเรชัน (Operation): หมายถึงเซตของกิจกรรมในแต่ละงาน (J_i) ที่ต้องทำงานเรียงตามลำดับ เช่นงาน J_i ประกอบด้วยเซตของโอเปอเรชัน $O_{ij}, j \in \{1, 2, \dots, h_i\}$ งานที่ i โอเปอเรชันที่ j
- เครื่องจักร (Machine): หมายถึงเครื่องจักรที่สามารถทำงานได้ กำหนดให้มีเครื่องจักรจำนวน m เครื่องจักร ดังนั้นเซตของเครื่องจักร $M_k, k \in \{1, 2, \dots, m\}$
- เวลาทำงาน (Processing Time): หมายถึงระยะเวลาทำงานของแต่ละโอเปอเรชัน (O_{ij}) บนเครื่องจักร M_k ซึ่ง $(1 \leq k \leq m)$; แสดงด้วย P_{ijk} ซึ่งมากกว่า 0
- เวลาเสร็จของงาน (Completion Time): เวลาทำงานเสร็จสิ้นของทุกงาน (C_{max})

2.1.2 ลักษณะของเครื่องจักรในการผลิต แบ่งเป็น 5 ประเภท ได้แก่

- 1) เครื่องจักรเดี่ยว (Single Machine) ระบบนี้มีเครื่องจักรเพียงเครื่องเดียว ซึ่งทุกงานทำงานภายใต้เครื่องจักรเดียวกัน
- 2) เครื่องจักรแบบขนาน (Parallel Machines) เป็นระบบที่มีเครื่องจักรจำนวน m เครื่องที่เหมือนกัน โดยจะมีการทำงานแบบขนานกัน ซึ่งงานแต่ละงานสามารถเลือกเครื่องจักรใดก็ได้ใน m เครื่อง
- 3) เครื่องจักรแบบไหลเลื่อน (Flow Shop) เป็นระบบที่มีเครื่องจักรจำนวน m เครื่องและงานทั้งหมดมีเส้นทางการไหลของงานเป็นรูปแบบเดียวกัน เช่น งานที่ 1 ทำบนเครื่องจักรที่ 1 งาน 2 ทำบนเครื่องจักรที่ 2 ไปเรื่อยๆ จนถึงเครื่องจักรสุดท้าย
- 4) การผลิตแบบเปิด (Open Shop) เป็นระบบที่มีเครื่องจักรจำนวน m เครื่อง แต่ละงานจะต้องมีการดำเนินงานแบบเวียนซ้ำบนเครื่องจักรแต่ละเครื่อง ไม่มีข้อจำกัดเกี่ยวกับเส้นทางงานของแต่ละงาน และสามารถกำหนดเส้นทางการผลิตให้กับแต่ละงานได้
- 5) การผลิตแบบตามสั่ง (Job Shop) เป็นระบบที่มีเครื่องจักรจำนวน m เครื่อง แต่ละงานมีเส้นทางการทำงานของตัวเองตามที่ผู้ผลิตกำหนด โดยวิธีการจัดตารางหรือชุดคำตอบ (solution) มีขนาดเท่ากับ $(n!)^m$ วิธี ซึ่ง n หมายถึงจำนวนงานและ m หมายถึงจำนวนเครื่องจักร ยกตัวอย่างขนาดปัญหาและจำนวนวิธีของการแก้ปัญหาสำหรับ 3 ปัญหาคือ ปัญหา 10 งาน 5 เครื่องจักร, ปัญหา 10 งาน 10 เครื่องจักร และ ปัญหา 30 งาน 10 เครื่องจักร คือ 6.2924×10^{32} , 3.9594×10^{65} และ 1.7243×10^{328} ตามลำดับ จากปัญหาดังกล่าวจะเห็นว่าจำนวนวิธีของการแก้ปัญหามีจำนวนมากและไม่สามารถได้คำตอบได้ในเวลาที่จำกัดได้ ซึ่งการผลิตแบบตามสั่งสามารถแบ่งตามความซับซ้อนของงานได้ 2 ประเภท คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1) ปัญหาการจัดตารางการผลิตแบบตามสั่ง (Job-Shop Scheduling Problem: JSSP) หรือปัญหาการจัดตารางการผลิตแบบตามสั่งแบบดั้งเดิม (Classical Job-Shop Scheduling) โดยเครื่องจักรผลิตสินค้าตามความต้องการของลูกค้าเป็นการผลิตตามคำสั่งซื้อ เช่น การรับจ้างสกรีน ลวดลายบนเสื้อผ้า กระเป๋า เป็นต้น ตัวอย่างการผลิตแบบตามสั่งสามารถแสดงได้ตามรูปที่ 2.1 ที่มีงานจำนวน 3 งาน แต่ละงานมี 3 โอเปอเรชันจำนวนเท่าๆ กัน และแต่ละโอเปอเรชันทำงานบนเครื่องจักรตามที่กำหนดไว้ ตัวอย่าง เช่น งานที่ 1 โอเปอเรชันที่ 1 ทำงานบนเครื่องจักรที่ 1 ใช้เวลา 6 หน่วยเวลา เป็นต้น

Job	Operation (machine/Time)		
	Op1	Op2	Op3
J1	1 (6)	2 (6)	3 (4)
J2	1 (2)	3 (10)	2 (6)
J3	2 (6)	1 (4)	3 (6)

รูปที่ 2.1 ตัวอย่างของปัญหาแบบ JSSP

5.2) ปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น (Flexible Job-Shop Scheduling Problem: FJSP) ระบบการผลิตนี้เป็นเหมือนการผลิตแบบตามสั่งแต่มีความยืดหยุ่นมากกว่า กล่าวคือ ในแต่ละงานมีลำดับโอเปอเรชันที่กำหนดไว้ ซึ่งแต่ละโอเปอเรชันสามารถเลือกเครื่องจักรทำงานได้มากกว่าหรือเท่ากับ 1 เครื่อง เนื่องจากเครื่องจักรแต่ละเครื่องมีความสามารถทำงานได้หลากหลาย ตัวอย่างในรูปที่ 2.2 แสดงจำนวนงาน 2 งาน แต่ละงานสามารถมีเซตของโอเปอเรชันแตกต่างกันได้ แต่ละโอเปอเรชันสามารถเลือกเครื่องจักรทำงานจากเซตของเครื่องจักรที่สามารถทำงานได้ ตัวอย่างเช่น งานที่ 2 โอเปอเรชันที่ 1 สามารถเลือกเครื่องจักรทำงานได้ 2 เครื่อง คือ M_1 ใช้เวลา 12 หน่วยเวลา และ M_2 ใช้เวลา 14 หน่วยเวลา ส่วน M_3 แสดงด้วยสัญลักษณ์ “-” หมายถึงไม่สามารถทำงานได้ เป็นต้น

Job	Operation	Machines		
		M ₁	M ₂	M ₃
J ₁	O _{1,1}	-	10	11
	O _{1,2}	2	3	4
	O _{1,3}	-	4	5
J ₂	O _{2,1}	<u>12</u>	<u>14</u>	-
	O _{2,2}	5	7	6
	O _{2,3}	-	2	-

รูปที่ 2.2 ตัวอย่างของปัญหาแบบ FJSP

ในการศึกษาได้กำหนดสมมติฐานเกี่ยวกับงาน และสมมติฐานเกี่ยวกับเครื่องจักร ดังนี้

1) สมมติฐานเกี่ยวกับงาน

- งานทั้งหมดมีความพร้อมทำ ณ เวลาเริ่มต้นของการจัดตาราง
- ไม่สามารถแยกงานให้ทำบนเครื่องจักรมากกว่า 1 เครื่องในเวลาเดียวกันได้
- มีการกำหนดเวลาดำเนินการของแต่ละงานและให้มีค่าคงที่
- กำหนดให้เวลาปรับตั้งเครื่องจักรและเวลาขนย้ายเป็นอิสระจากงาน ลำดับงาน และเวลาเหล่านี้ถูกรวมไว้ในเวลาดำเนินการแล้ว
- ขณะทำงานไม่สามารถแทรกงานที่อยู่ระหว่างทำ
- ทุกงานมีความสำคัญเท่ากัน

2) สมมติฐานเกี่ยวกับเครื่องจักร

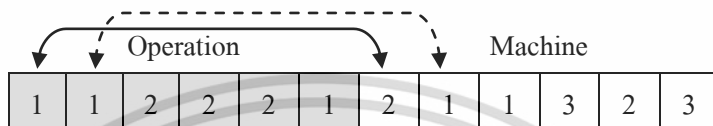
- เครื่องจักรทั้ง m เครื่องมีความพร้อมที่จะเริ่มต้นทำงานทันที ณ เวลาเริ่มต้นของการวางแผน
- ในแต่ละโอเปอเรชันสามารถทำบนเครื่องจักรเพียงเครื่องเดียว ในเวลาหนึ่งๆ
- เครื่องจักรทุกเครื่องอยู่ในสถานะพร้อมใช้งานและไม่มีการเสียในระหว่างทำงาน

2.1.3 การสร้างชุดคำตอบสำหรับปัญหาการจัดตารางการผลิต

จากตัวอย่างข้อมูลปัญหาการจัดตารางการผลิตแบบตามสิ่งยึดหยุ่นในรูปที่ 2.2 สามารถสร้างชุดคำตอบได้ดังรูปที่ 2.3 โดยกำหนดให้ความยาวของชุดคำตอบเท่ากับจำนวนโอเปอเรชันทั้งหมดของทุกงาน เท่ากับ 6 โอเปอเรชัน จากนั้นกำหนดลำดับการทำงานในชุดคำตอบซึ่งประกอบด้วย 2 ส่วน คือ ส่วนที่ 1) การจัดลำดับโอเปอเรชัน (operation) แสดงลำดับของหมายเลขงาน และส่วนที่ 2) การเลือกเครื่องจักร (machine) แสดงหมายเลขเครื่องจักรที่ถูกเลือกให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานสำหรับโอเปอร์เรชันนั้น ซึ่งการดำเนินการผลิตจะทำตามลำดับจากซ้ายไปขวา ดังตัวอย่างในรูป 2.3 คู่ลำดับแรก (เส้นทึบ) หมายถึงงานที่ 1 โอเปอร์เรชันที่ 1 (เนื่องจากเป็นลำดับแรกของงานที่ 1) กำหนดให้ทำงานบนเครื่องจักรที่ 2 ส่วนคู่ลำดับถัดไป (เส้นประ) หมายถึง งานที่ 1 โอเปอร์เรชันที่ 2 (เนื่องจากเป็นลำดับที่ 2 ของงานที่ 1) กำหนดให้ทำงานบนเครื่องจักรที่ 1 เป็นต้น จนครบทุกคู่ลำดับ ซึ่งวิธีการเลือกลำดับโอเปอร์เรชัน และเลือกเครื่องจักรมีหลายวิธี วิธีการสุ่มเป็นหนึ่งในวิธีที่สามารถเลือกโอเปอร์เรชันและเครื่องจักรเพื่อนำมาสร้างชุดคำตอบได้

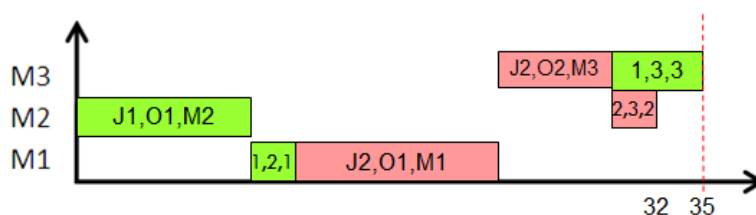


รูปที่ 2.3 ตัวอย่างการสร้างชุดคำตอบของปัญหาแบบ FJSP

2.1.4 เครื่องมือในการวัดประสิทธิภาพการจัดการการผลิต

แผนภูมิแกนต์ (Gantt Chart) ถูกพัฒนาขึ้นประมาณปี ค.ศ.1917 โดย Henry L. Gantt ซึ่งเป็นเครื่องมือช่วยทางควบคุมแผนงานและโครงการการบริหารที่นิยมใช้มานานเพื่อแสดงให้เห็นถึงความสัมพันธ์ของกิจกรรมต่างๆ ที่เกิดขึ้นภายใต้เวลาที่กำหนดให้ซึ่งแนวตั้งหมายถึงทรัพยากร และแนวนอนหมายถึงเวลาที่ใช้ของแต่ละกิจกรรม ซึ่งการวัดประเมินประสิทธิภาพของโมเดลการจัดการการผลิต สามารถประเมินได้หลายวิธีขึ้นอยู่กับลักษณะการผลิต เช่น เวลาในการผลิตรวม น้อยที่สุด (Minimize Makespan: C_{max}), จำนวนงานที่ล่าช้า (Number of Tardy Jobs) และเวลาล่าช้าสูงสุด (Maximum Lateness) เป็นต้น ในงานวิจัยนี้ได้ใช้การประเมินจากเวลาในการผลิตรวมที่น้อยที่สุด ตามสมการที่ 2.1 เมื่อ C_1, C_2, \dots, C_p คือ เซตของคำตอบที่เป็นไปได้ในการจัดการการผลิต ซึ่งจะเลือกคำตอบที่ใช้เวลาในการผลิตน้อยที่สุด และ Z หมายถึงวัตถุประสงค์ของการจัดการที่มีค่าต่ำสุด ดังสมการที่ 2.1 รูปที่ 2.4 แสดงตัวอย่างค่าคำตอบของชุดคำตอบจากรูปที่ 2.3 ใช้เวลาที่ใช้ในการผลิตรวมของทุกงาน ($C_{max} = 35$)

$$Z = f(C_1, C_2, \dots, C_p) \tag{2.1}$$



รูปที่ 2.4 แผนภูมิแกนต์ของชุดคำตอบ ($C_{max} = 35$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 การจัดกลุ่มเป็นลำดับชั้น

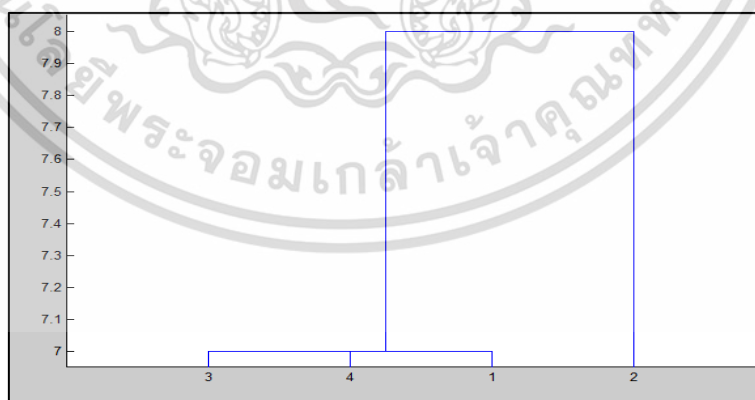
การจัดกลุ่มเป็นลำดับชั้น (Hierarchical Clustering) เป็นเครื่องมือที่นิยมใช้ในการวิเคราะห์ข้อมูล โดยมีแนวคิดมาจากวิธีการสร้างไบนารีทรี (Binary Tree) การรวมกลุ่มจะพิจารณาจากความคล้ายคลึงกันระหว่างข้อมูล ซึ่งสามารถแบ่งเป็น 2 วิธี ได้แก่

1) การรวมกลุ่มเป็นลำดับชั้น (Agglomerative Hierarchical method)

เป็นการแบ่งกลุ่ม โดยถือว่าทุกหน่วยข้อมูลเป็นหนึ่งกลุ่มย่อย จากนั้นพิจารณากลุ่มความคล้ายระหว่างหน่วยข้อมูลเพื่อรวมกลุ่มครั้งละ 2 กลุ่มจนกระทั่งรวมกลุ่มทั้งหมดเป็นกลุ่มเดียว สามารถสรุปเป็นขั้นตอนตามอัลกอริทึมและแผนภาพตามรูปที่ 2.5 และเมื่อจัดกลุ่มแล้วจะได้การจัดกลุ่ม 2 กลุ่ม กลุ่ม $G1 = \{1,3,4\}$ และ กลุ่ม $G2 = \{2\}$ ดังรูปที่ 2.6

- a) กำหนดค่าความคล้ายระหว่างตำแหน่งของข้อมูลด้วยตารางเมตริกซ์
- b) ทำซ้ำในขั้นตอนต่อไปนี้
 - รวมกลุ่มข้อมูล 2 กลุ่มที่อยู่ใกล้กัน
 - ปรับค่าความคล้าย ระหว่างตำแหน่งข้อมูลเดิมกับกลุ่มใหม่ที่ถูกรวม
- c) ทำจนกระทั่ง กลุ่มข้อมูลเป็นกลุ่มเดียวกัน

รูปที่ 2.5 อัลกอริทึมพื้นฐานของวิธีการรวมกลุ่มเป็นลำดับชั้น [3]



รูปที่ 2.6 แผนภาพการรวมกลุ่มเป็นลำดับชั้น

วิธีการรวมกลุ่มของการรวมกลุ่มแบบมีขั้นตอนมีหลายวิธี [4] ในที่นี้จะขอกกล่าว

เฉพาะวิธีนิยมใช้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1) วิธีการรวมกลุ่มแบบเดี่ยว (Single Linkage) พิจารณาจากค่าความคล้ายของคู่ที่อยู่ใกล้ที่สุด ดังสมการ 2.2

$$d_{SL}(G, H) = \min_{i \in G, j \in H} d_{i,j} \quad (2.2)$$

เมื่อ i, j คือ สมาชิกของกลุ่ม G และ H
 d คือ ระยะทางระหว่างตำแหน่ง i, j

1.2) วิธีการรวมกลุ่มแบบสมบูรณ์ (Complete Linkage) พิจารณาจากค่าความคล้ายของคู่ที่อยู่ใกล้ที่สุด ดังสมการ 2.3

$$d_{CL}(G, H) = \max_{i \in G, j \in H} d_{i,j} \quad (2.3)$$

เมื่อ i, j คือ สมาชิกของกลุ่ม G และ H
 d คือ ระยะทางระหว่างตำแหน่ง i, j

2) การแบ่งกลุ่มเป็นลำดับชั้น (Divisive hierarchical method) คือ เริ่มจากการแบ่งกลุ่มโดยถือว่าทุกหน่วยอยู่ในกลุ่มเดียวกันหรือมีเพียงกลุ่มเดียวเท่านั้น จากนั้นแบ่งออกเป็นกลุ่มย่อยๆ

2.2.2 จีเนติกอัลกอริทึม

จีเนติกอัลกอริทึม (Genetic Algorithm: GA) คิดค้นโดย John Holland ในปี ค.ศ. 1975 [5] โดยศึกษาทฤษฎีของการวิวัฒนาการทางธรรมชาติ และทฤษฎีของ Darwin ที่ศึกษาทฤษฎีวิวัฒนาการทางพันธุกรรมที่สามารถถ่ายทอดลักษณะต่างๆ ไปสู่รุ่นต่อไป เช่น การครอสโอเวอร์ (Crossover) และการมิวเตชัน (Mutation) เป็นต้น ผลลัพธ์ที่ได้คือโครโมโซมลูกที่ดีขึ้นหรือแย่งกระบวนการทางธรรมชาติจะทำการคัดเลือกโครโมโซมซึ่งโครโมโซมที่ดีมีโอกาสสูงที่จะถูกเลือก ดังนั้นโครโมโซมทายาทมีแนวโน้มจะเป็นโครโมโซมที่ดีขึ้น

หลักการทำงานของจีเนติกอัลกอริทึม มีขั้นตอนดังนี้

- 1) ขั้นตอนการเตรียมการ ประกอบด้วย การกำหนดฟังก์ชันเป้าหมาย (Objective Function) และการกำหนดรูปแบบโครโมโซมโดยการเข้ารหัสชุดคำตอบ (Encoding) ตามลักษณะของปัญหา
- 2) ขั้นตอนการดำเนินการ ประกอบด้วย
 - 2.1) สร้างประชากรหรือโครโมโซมเริ่มต้นจำนวน N โครโมโซม
 - 2.2) กำหนดค่าฟิตเนสของฟังก์ชันเป้าหมายในแต่ละโครโมโซม

เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3) คัดเลือกโครโมโซมเพื่อเป็นโครโมโซมต้นแบบ โดยเลือกคู่จากวงล้อรูเล็ตที่สร้างจากค่าฟิตเนสของแต่ละโครโมโซม

2.4) การครอสโอเวอร์ คือ การนำโครโมโซมต้นแบบ จำนวน 2 โครโมโซมมาแลกเปลี่ยนระหว่างกันเพื่อสร้างโครโมโซมทายาท (Offspring)

2.5) การมิวเตชัน คือ การปรับปรุ่ค่าในโครโมโซมลูก เพื่อสร้างความหลากหลายให้กับโครโมโซมในการปรับปรุ่โครโมโซม

2.6) ตรวจสอบเงื่อนไขการหยุด ถ้ายังไม่พบให้ทำการคัดเลือกโครโมโซมที่ดีที่สุดไปเป็นประชากรรุ่นถัดไปและไปทำขั้นตอนที่ 2.2

2.2.3 ฟัซซีเซต

ฟัซซีเซต (Fuzzy Sets) เป็นการคิดค้นโดย Lotfi A. Zadeh ในปี 1965 [6] จากแนวคิดทฤษฎีเซตแบบดั้งเดิม (Classical Set หรือ Crisp Set) ที่มีการแบ่งขอบเขตการเป็นสมาชิกของเซตไว้อย่างแน่นอนว่า “ใช่” หรือ “ไม่ใช่” ซึ่งง่ายต่อพิสูจน์ในความเป็นสมาชิกเนื่องจากการแบ่งแยกชัดเจน ตัวอย่างเช่น กำหนดให้ X เป็นเซตเอกภพสัมพัทธ์ (Universal Set) และมีองค์ประกอบคือ x สามารถเขียนนิยามเซตด้วยฟังก์ชัน $f_A(x)$ คือ ฟังก์ชันคุณลักษณะของเซต A (Characteristic Function) ดังสมการ 2.4

$$f_A(x) : X \rightarrow \{0,1\} \quad (2.4)$$

เมื่อมีค่าความเป็นสมาชิก

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

ในความเป็นจริงมีปัญหาหลายอย่างที่ก่อให้เกิดความขัดแย้งจากทฤษฎีเซตแบบดั้งเดิม จึงมีการนำทฤษฎีฟัซซีเซตมาใช้โดยกำหนดให้องค์ประกอบในฟัซซีเซตมีค่าระดับความเป็นสมาชิกของเซตอยู่ระหว่าง 0 และ 1 ดังนั้นองค์ประกอบดังกล่าวอาจจะไม่มีระดับเพียงแค่ใช่หรือไม่ใช่ แต่อาจจะใช่บางส่วน หรือไม่ใช่บางส่วน สำหรับทฤษฎีเซตแบบฟัซซีสามารถนิยามฟังก์ชันค่าความเป็นสมาชิกคือ $\mu_A(x)$ เรียกว่าค่าความเป็นสมาชิกของฟังก์ชัน (Membership Function) ของเซต A ดังสมการ 2.5

$$f_A(x) : X \rightarrow [0,1] \quad (2.5)$$

เมื่อมีค่าความเป็นสมาชิก

$$\mu_A(x) = \begin{cases} 1 & \text{ถ้า } x \text{ อยู่ในเซต } A \text{ ทั้งหมด} \\ 0 & \text{ถ้า } x \text{ ไม่อยู่ในเซต } A \text{ เลย} \\ (0,1) & \text{ถ้า } x \text{ อยู่ในเซต } A \text{ บางส่วน} \end{cases}$$

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 A Simulated Annealing Algorithm for Flexible Job-Shop Scheduling Problem

M. Yazdani, M. Gholami, M. Zandieh และ M. Mousakhani [7] ศึกษาวิธีการแก้ปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่นโดยใช้อัลกอริทึมซิมูเลทเทดแอนนิลลิง โดยมีฟังก์ชันเป้าหมาย คือ เวลาในการผลิตรวมน้อยที่สุด ซึ่งอัลกอริทึมซิมูเลทเทดแอนนิลลิงพัฒนาจาก Kirkpatrick et. al ในปี ค.ศ. 1984 สำหรับแก้ปัญหาการตัดสินใจที่ต้องการค่าต่ำสุดหรือค่าสูงสุด หลักการของอัลกอริทึมนี้คือการจำลองการควบคุมการเย็นตัวของวัตถุ ซึ่งวัตถุจะถูกความร้อนจนกระทั่งถึงจุดหลอมเหลวเมื่อเวลาผ่านไปอุณหภูมิของวัตถุจะค่อยๆ ลดลงและทำให้วัตถุมีความแข็งเมื่อเย็นตัวลง (Cooling Schedule) ซึ่งการเย็นตัวของวัตถุลดลงช้ามากเท่าไรจะได้วัตถุที่มีความแข็งแรงมากขึ้นเท่านั้น ในงานวิจัยนี้ได้เสนอแนวคิด ประกอบด้วย

ตารางที่ 2.1 ตัวอย่างของปัญหาขนาด 4 งาน 4 เครื่องจักร [7]

โอเปอร์เรชัน	เครื่องจักร			
	M1	M2	M3	M4
O _{1,1}	4	7	6	5
O _{1,2}	2	6	∞	5
O _{2,1}	4	5	7	∞
O _{2,2}	5	∞	6	3
O _{2,3}	∞	5	4	7
O _{3,1}	5	3	∞	6
O _{3,2}	∞	∞	4	∞
O _{4,1}	2	4	∞	5
O _{4,2}	∞	4	2	∞
O _{4,3}	5	4	6	3

1) กำหนดรูปแบบการเข้ารหัสชุดคำตอบ (Encoding)

กำหนดให้ชุดคำตอบ (solution) มีความยาวเท่ากับผลรวมของทุกโอเปอร์เรชันของทุกงาน

ในงานวิจัยนี้ใช้การเข้ารหัสชุดคำตอบแบบรวม (Integrated Approaches) [8] คือชุดคำตอบที่มีการ
 เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ วรวิทย์ วัฒนวิทย์ และคณะผู้วิจัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดเครื่องจักรให้ทำงานกับโอเปอร์เรชัน จากตัวอย่างปัญหาในตารางที่ 2.1 มี 4 งาน 4 เครื่องจักร แต่ละงานมีจำนวนโอเปอร์เรชันไม่เท่ากัน และใช้เวลาทำงานแตกต่างกันตามความสามารถของเครื่องจักร สัญลักษณ์ ∞ หมายถึงเครื่องจักรดังกล่าวไม่สามารถทำงานในโอเปอร์เรชันนั้นได้ ดังนั้นความยาวของชุดคำตอบเท่ากับ 10 โอเปอร์เรชัน แต่ละโอเปอร์เรชันประกอบด้วย 3 ค่า (i, j, k) โดยที่ i คือหมายเลขงาน j คือลำดับของโอเปอร์เรชัน และ k คือหมายเลขเครื่องจักร ในรูปที่ 2.7 แสดงชุดโครโมโซม ตำแหน่งโอเปอร์เรชันที่ 7 (1, 1, 4) หมายถึงงานที่ 1 และโอเปอร์เรชันที่ 1 ทำงานบนเครื่องจักร 4

S:	(3,1,2)	(2,1,1)	(3,2,3)	(4,1,1)	(2,2,4)	(4,2,3)	(1,1,4)	(1,2,1)	(4,3,2)	(2,3,3)
----	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

รูปที่ 2.7 ตัวแทนชุดคำตอบ [7]

2) การสร้างชุดคำตอบเริ่มต้น

ในงานวิจัยนี้อ้างอิงวิธีการกำหนดการสร้างชุดคำตอบจาก Pezzella et al. [9] มี 2 วิธีคือ 1) Assignment rule 1 คือการเลือกเครื่องจักรให้โอเปอร์เรชัน โดยเลือกเครื่องจักรที่ใช้เวลาทำงานน้อยที่สุดจากเวลาทำงานของเครื่องจักรทั้งหมด และปรับค่าเวลาทำงานของเครื่องจักรนั้นตามหน่วยเวลาที่เลือก จากนั้นพิจารณาเลือกโอเปอร์เรชันลำดับถัดไปด้วยหลักการเดิมคือพิจารณาเวลาทำงานของเครื่องจักรที่น้อยที่สุดจากขั้นตอนแรกและปรับค่าเวลาทำงานงานของเครื่องจักรนั้น ทำเช่นเดียวกันจนครบทุกโอเปอร์เรชัน และ 2) Assignment rule 2 คือเลือกเครื่องจักรให้โอเปอร์เรชันโดยพิจารณาจากเวลาทำงานน้อยที่สุด โดยพิจารณาเฉพาะเซตของเครื่องจักรที่สามารถทำบนโอเปอร์เรชันนั้น และปรับค่าเวลาของเครื่องจักรนั้นเช่นเดียวกับวิธีแรก ซึ่งงานวิจัยนี้กำหนดสัดส่วนการสร้างประชากรเริ่มต้นด้วยวิธี Assignment rule 1 และ Assignment rule 2 เท่ากับ 40% และ 60% ตามลำดับ

3) โครงสร้างการค้นหาคำตอบพื้นที่รอบข้าง (Neighborhood search)

3.1) NSS-Se คือ เปลี่ยนลำดับของโอเปอร์เรชัน โดยสุ่มโอเปอร์เรชันในชุดคำตอบและนำไปแทรกตำแหน่งที่เป็นไปได้ตามเงื่อนไขลำดับของงาน ส่วนของเครื่องจักรไม่เปลี่ยนแปลงและทำซ้ำเท่ากับจำนวน k_{\max} รอบ

3.2) NSS-As คือ เปลี่ยนเครื่องจักรให้กับโอเปอร์เรชัน โดยสุ่มโอเปอร์เรชันในชุดคำตอบจากนั้นเลือกเครื่องจักรใหม่จากเซตเครื่องจักรที่โอเปอร์เรชันนั้นสามารถทำได้ ให้ทำซ้ำเท่ากับจำนวน A_{\max} รอบ

4) ปรับระดับความเย็น (Cooling schedule) คือการควบคุมอุณหภูมิให้เย็นลงในแต่ละรอบการทำงานโดยคำนวณดังสมการ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_i = T_0 - i \frac{T_0 - T_f}{N} \quad (2.6)$$

- เมื่อ T_0 คือ อุณหภูมิเริ่มต้น
 T_f คือ อุณหภูมิที่ทำให้หยุดการไหล
 T_i คือ อุณหภูมิในแต่ละรอบการทำงาน
 i คือ จำนวนรอบ
 N คือ จำนวนรอบการทำงานของอัลกอริทึม

อัลกอริทึมที่นำเสนอในงานวิจัยมีขั้นตอนการทำงาน ดังนี้

1) กำหนดพารามิเตอร์เริ่มต้น

ประกอบด้วย รูปแบบการเข้ารหัสชุดคำตอบ กำหนดชุดคำตอบเริ่มต้น อุณหภูมิเริ่มต้น จำนวนรอบการค้นหาคำตอบพื้นที่รอบข้าง การลดลงของอุณหภูมิ ชุดของการค้นหาคำตอบพื้นที่รอบข้าง และเงื่อนไขการหยุดการทำงานของอัลกอริทึม

2) กำหนดเงื่อนไขการทำงาน N รอบ และกำหนดคำตอบเริ่มต้น คือ คำตอบของรอบปัจจุบัน ($S \leftarrow S_0$) และกำหนดคำตอบเริ่มต้นคือคำตอบแบบ โกลบอล ($S^* \leftarrow S_0$)

3) กำหนดรอบการทำงานการค้นหาแบบโลคอล (l), กำหนด $l = 1$ และสร้างชุดคำตอบรอบข้าง (S') ด้วยวิธีการดังนี้

- ถ้า $l = 1$ ให้สร้างคำตอบพื้นที่รอบข้างแบบ NSS-Se และกำหนดให้ $k_{\max} = 3$
- ถ้า $l = 2$ ให้สร้างคำตอบพื้นที่รอบข้างแบบ NSS-Ae และกำหนดให้ $A_{\max} = 1$
- ถ้า $l = 3$ ให้สร้างคำตอบพื้นที่รอบข้างแบบ NSS-Se และกำหนดให้ $k_{\max} = 4$
- ถ้า $l = 4$ ให้สร้างคำตอบพื้นที่รอบข้างแบบ NSS-Ae และกำหนดให้ $A_{\max} = 2$
- ถ้า $l = 5$ ให้สร้างคำตอบพื้นที่รอบข้างแบบ NSS-Se และกำหนดให้ $k_{\max} = 5$
- ถ้า $l = 6$ ให้สร้างคำตอบพื้นที่รอบข้างแบบ NSS-Ae และกำหนดให้ $A_{\max} = 3$
- ถ้า $l = 7$ ให้สร้างคำตอบพื้นที่รอบข้างแบบ NSS-Se และกำหนดให้ $k_{\max} = 6$

4) เปรียบเทียบคำตอบพื้นที่รอบข้าง (S') และคำตอบปัจจุบัน (S) ด้วยเงื่อนไขดังนี้

- ถ้าฟิตเนสของ (S') น้อยกว่า (S) ให้ทำการปรับค่าคำตอบปัจจุบันให้เป็นคำตอบพื้นที่รอบข้าง ($S \leftarrow S'$) และกำหนดวิธีสร้างคำตอบรอบข้างคือ $l = 1$

- ถ้าฟิตเนสของ (S') เท่ากับ (S) ให้สุ่มค่า $R[0,1]$ จากนั้นทำการเปรียบเทียบ ถ้า $R < 0.5$ แล้วให้ทำการปรับค่าคำตอบปัจจุบันให้เป็นคำตอบพื้นที่รอบข้าง ($S \leftarrow S'$) และกำหนดวิธีสร้างคำตอบพื้นที่รอบข้างคือ $l = 1$

- ถ้าฟิตเนสของ (S') มากกว่า (S) ให้ทำการสุ่มค่า $R[0,1]$ จากนั้นทำการเปรียบเทียบค่าที่สุ่มกับความน่าจะเป็นในการยอมรับการเปลี่ยนแปลงของพลังงาน (acceptance probabilities) คือ $R < \exp(-\Delta s, s'/T)$ แล้ว ให้ทำการปรับค่าคำตอบปัจจุบันให้เป็นคำตอบพื้นที่รอบข้าง ($S \leftarrow S'$) และกำหนดวิธีสร้างคำตอบพื้นที่รอบข้าง $l = l+1$

- ถ้าหาก $l = 8$ แล้วกำหนดให้ $l = 1$

5) ให้ทำซ้ำจนครบจำนวนรอบการค้นหาแบบ โลคอลกำหนดไว้ (nt)

6) ปรับค่าคำตอบเริ่มต้นของแต่ละรอบ

7) ถ้าค่าฟิตเนสของคำตอบที่ดีที่สุดในการค้นหาแบบ โลคอลดีกว่าคำตอบแบบ โกลบอลแล้ว ให้ปรับค่าคำตอบแบบ โกลบอล

8) ปรับลดอนุกรมวิธานด้วยสมการ 2.6

9) ทำซ้ำตั้งแต่ขั้นตอนที่ 2 – 9 จนครบจำนวนรอบที่กำหนดของอัลกอริทึม

งานวิจัยนี้ได้วัดประสิทธิภาพของโมเดล SA ที่นำเสนอ กับข้อมูลปัญหาจำนวน 20 ชุดของ Fattahi et al. (Fdata) [10] โดยแบ่งข้อมูลเป็น 2 ประเภท คือปัญหาขนาดเล็ก (SFJS1:10) และปัญหาระดับกลางและใหญ่ (MFJS1:10) โดยกำหนดค่าพารามิเตอร์ให้เหมาะสมกับข้อมูลแต่ละชุด ได้แก่ จำนวนประชากรเริ่มต้น ค่าอนุกรมวิธานเริ่มต้น จำนวนรอบการค้นหาคำตอบพื้นที่รอบข้าง และเงื่อนไขการหยุดการทำงานของอัลกอริทึม ดังตารางที่ 2.2 และเปรียบเทียบกับ 6 อัลกอริทึมซึ่งเป็นแบ่งเป็นการสร้างชุดคำตอบแบบลำดับชั้น ได้แก่ ISA และ ITS และการสร้างชุดคำตอบแบบรวม ได้แก่ HAS/SA, HAS/TS, HTS/TS และ HTS/SA ผลการทดลองพบว่าโมเดล SA ที่นำเสนอให้ผลลัพธ์ดีกว่าทั้ง 6 โมเดล ซึ่งพิจารณาจากค่าคลาดเคลื่อนสัมพัทธ์เฉลี่ย (Mean relative error: MRE) เท่ากับ 14.47 ซึ่งต่ำกว่าโมเดลที่นำมาเปรียบเทียบ

ข้อสังเกตของโมเดลนี้ จากการทดลองแสดงให้เห็นว่ารูปแบบการสร้างชุดคำตอบไม่มีผลต่อการค้นหาคำตอบที่เหมาะสม และโมเดลที่นำเสนอมีประสิทธิภาพสำหรับปัญหาขนาดเล็กมากกว่าปัญหาที่มีขนาดใหญ่

ตารางที่ 2.2 ค่าพารามิเตอร์ที่ใช้ในการทดสอบ

ปัญหา	จำนวนประชากร	อุณหภูมิเริ่มต้น (T_0)	จำนวนพื้นที่ที่รอบข้างในแต่ละอุณหภูมิ (nt)	จำนวนรอบสูงสุดของอัลกอริทึม (N)
SFJS1	20	35	30	20
SFJS2	20	32	30	20
SFJS3	20	76	50	50
SFJS4	30	84	50	50
SFJS5	30	38	50	50
SFJS6	30	75	50	50
SFJS7	30	102	50	50
SFJS8	30	62	100	50
SFJS9	50	55	200	50
SFJS10	50	132	200	100
MFJS1	100	120	200	200
MFJS2	100	112	200	300
MFJS3	100	125	200	400
MFJS4	200	136	200	500
MFJS5	200	130	200	500
MFJS6	300	145	300	1000
MFJS7	300	157	300	2000
MFJS8	500	152	300	3000
MFJS9	500	170	500	3000
MFJS10	500	150	500	3000

2.3.2 An efficient Job-Shop Scheduling Algorithm base on Particle Swarm

Optimization

Tsung-Lieh Lin, Shi-Jinn Horng, Tzong-Wann Kao, Yuan-Hsin Chen, Ray-Shine Ran, Rong-Jian Chen, Jui-Lin Lai และ I-Hong Kuo [11] งานวิจัยนี้นำเสนอโมเดล MPSO (Multi-type individual enhancement scheme into PSO) เป็นการแก้ปัญหาการจัดตารางการผลิตแบบตามสั่งที่ผสมผสานระหว่าง อัลกอริทึมพาร์ทิเคิลสวอร์ม อัลกอริทึมชิมูเลทเทดแอนนิลลิง และการเพิ่มความหลากหลายของพาร์ทิเคิล (Multi-type Individual Scheme : MIE) เพื่อนำมาใช้แก้ปัญหาการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบตามสั่ง เพื่อให้ได้ค่าที่เหมาะสมที่สุด ฟังก์ชันวัตถุประสงค์ของงานวิจัยนี้คือ เวลาในการผลิตรวมน้อยที่สุด ดังสมการ 2.7

$$\text{Minimize } O_{nm+1} \text{ หรือ } (C_{\max}) \quad (2.7)$$

เมื่อ O_{nm+1} หมายถึง เวลาทำงานเสร็จของ โอเปอร์เรชันตัวสุดท้าย ซึ่งกำหนดให้จำนวนโอเปอร์เรชันเท่ากับ $(nm) + 2$ (ตำแหน่งเริ่มต้นและสิ้นสุด)

ในงานวิจัยนี้ได้ศึกษาหลักการของอัลกอริทึมพาร์ทิเคิลสวอรัม การกำหนดตัวแทนของพาร์ทิเคิลด้วยวิธีการสุ่มสลับ, การปรับค่าพาร์ทิเคิลเพื่อเพิ่มความสามารถของการค้นหา และอัลกอริทึมซิมูเลทเทคแอนนิลลิง ซึ่งสามารถสรุปความสำคัญในงานวิจัยนี้ ดังนี้

1) หลักการของอัลกอริทึมพาร์ทิเคิลสวอรัมแบบดั้งเดิม

หลักการดั้งเดิมของอัลกอริทึมพาร์ทิเคิลสวอรัม เป็นการเคลื่อนที่ของกลุ่มพาร์ทิเคิลในการหาตำแหน่ง โดยจะมีการปรับค่าความเร็วและหาตำแหน่งใหม่ของแต่ละพาร์ทิเคิลทีละรอบ และมีการจดจำความเร็วและตำแหน่งที่ดีที่สุดของแต่ละพาร์ทิเคิล และค่าที่ดีที่สุดของกลุ่มในการปรับค่าความเร็วและตำแหน่ง ดังสมการ 2.8 และ 2.9

$$v_{id} = \omega \times v_{id} + c_1 \times \text{Rand}() \times (p_{id}^{\text{best}} - p_{id}) + c_2 \times \text{Rand}() \times (p_{gd}^{\text{best}} - p_{id}) \quad (2.8)$$

$$P_{id} = P_{id} + v_{id} \quad (2.9)$$

เมื่อ v_{id} คือ ความเร็วการบินของพาร์ทิเคิลตัวที่ id ซึ่งจะมีค่าอยู่ในช่วง $[-v_{\max}, v_{\max}]$

ω คือ ความเฉื่อยของค่าความเร็ว

C_1 คือ ค่าคงที่ที่แสดงถึงความเชื่อประสบการณ์ที่ดีที่สุดของตัวเอง (p_{id}^{best})

C_2 คือ ค่าคงที่ที่แสดงถึงความเชื่อตามฝูง (p_{gd}^{best})

$\text{Rand}()$ คือ ค่าคงที่จากการสุ่ม ซึ่งมีค่าอยู่ระหว่าง (0 - 1)

p_{id} คือ ตำแหน่งพาร์ทิเคิลตัวที่ id

p_{id}^{best} คือ ตำแหน่งที่ดีที่สุดของพาร์ทิเคิลตัวที่ id

p_{gd}^{best} คือ ตำแหน่งที่ดีที่สุดของพาร์ทิเคิลทุกตัวในฝูงนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) กำหนดตัวแทนของพาร์ทิเคิล (p) ด้วยวิธีสุ่มสลับ (Random-Key: RK) มีขั้นตอนดังนี้
 - 2.1) สุ่มเลขจำนวนจริง (Real Number) ของแต่ละโอเปอร์เรชันตามลำดับจนครบ
 - 2.2) แสดงลำดับของค่าจำนวนจริงจากน้อยไปมาก ($\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_{nm}$) ซึ่ง π_k เป็นค่าตัวเลขระหว่าง 1 และ nm ($1 \leq k \leq nm$) แสดงเลขลำดับตามตำแหน่งที่จัดเรียง
 - 2.3) มอดคูลุส n แล้ว $+1$ [$(\pi_k \bmod n) + 1$]
 - 2.4) แสดงลำดับโอเปอร์เรชันในแต่ละตำแหน่งของแต่ละพาร์ทิเคิล
- 3) ปรับค่าพาร์ทิเคิล (p') เพื่อเพิ่มความสามารถของการค้นหาแบบโลคอลให้ได้ชุดคำตอบที่ดีกว่าเดิม ด้วยการสุ่ม 2 ตำแหน่ง (p, q) ในพาร์ทิเคิลเพื่อทำการสลับ ซึ่งมี 4 วิธี คือ
 - 3.1) สลับตำแหน่ง (Prob_s) ระหว่าง p และ q
 - 3.2) แทรกตำแหน่ง (Prob_i) นำตำแหน่งของ p แทรกตำแหน่งของ q
 - 3.3) ย้อนกลับตำแหน่ง (Prob_{mv}) ย้อนกลับตำแหน่งตั้งแต่ p ถึง q
 - 3.4) กำหนดช่วงย้ายตำแหน่ง (Prob_{long}) ย้ายช่วง p ถึง q ไปตำแหน่ง r
- 4) อัลกอริทึมชิมูเลทเทคแอนนิลลิง มีขั้นตอน ดังนี้
 - 4.1) กำหนดคำตอบเริ่มต้น (s) และสุ่มหาคำตอบจากพื้นที่รอบข้าง (s')
 - 4.2) หาค่าความเหมาะสมของ $f(s)$ และ $f(s')$
 - 4.3) ถ้า $\Delta = f(s') - f(s)$, แล้ว Δ น้อยกว่า 0, ยอมรับคำตอบใหม่ (s')
 - 4.4) ถ้า $\Delta = f(s') - f(s)$, แล้ว Δ มากกว่า 0, ยอมรับคำตอบใหม่ (s') ถ้าการสุ่มค่าความน่าจะเป็นน้อยกว่า $\min\left\{1, \exp\left(\frac{-\Delta}{T}\right)\right\}$

อัลกอริทึม MPSO ที่ได้นำเสนอในงานวิจัยมีขั้นตอนการทำงาน ดังนี้

- 1) กำหนดค่าพารามิเตอร์

กำหนดค่า Prob_{MIE}, C1, C2, ω , MaxIter และสร้างพาร์ทิเคิล 1 กลุ่ม ด้วยการสุ่มค่าตำแหน่งของพาร์ทิเคิลจำนวน N พาร์ทิเคิล ใช้วิธีการสุ่มสลับ และกำหนดค่าความเร็วเริ่มต้นของแต่ละพาร์ทิเคิลด้วยการสุ่มค่าจำนวนจริง ซึ่งอยู่ระหว่าง $[-v_{max}, v_{max}]$
- 2) คำนวณค่าฟิตเนสของแต่ละพาร์ทิเคิล หากยังไม่พบคำตอบหรือครบจำนวนรอบที่กำหนดให้ทำขั้นตอนถัดไป
- 3) กำหนดให้แต่ละพาร์ทิเคิล มีขั้นตอนดำเนินการ ดังนี้
 - 3.1) สุ่มค่า (s) ถ้าน้อยกว่า Prob_{MIE} ให้ทำวิธีการ MIE ของแต่ละพาร์ทิเคิล มิฉะนั้นให้ทำขั้นตอนที่ 4
 - 3.2) ทำวิธี MIE เพื่อเพิ่มประสิทธิภาพของพาร์ทิเคิลด้วยอัลกอริทึมชิมูเลทเทคแอนนิลลิง มีขั้นตอนดังนี้

a) กำหนดอุณหภูมิเริ่มต้น (T) และปรับปรุงพาร์ทิเคิลด้วยวิธีการปรับค่า (p')

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- b) หาค่าความเหมาะสมของพาร์ทิเคิลตัวใหม่ (p')
- c) เปรียบเทียบค่าผลต่าง (Δ) ระหว่างค่าความเหมาะสมของพาร์ทิเคิลตัวเดิม (p) และพาร์ทิเคิลตัวใหม่ (p')

- ถ้า Δ มากกว่า 0 แสดงว่าพาร์ทิเคิลตัวใหม่ (p') แย่กว่าพาร์ทิเคิลตัวเดิม (p) ในการที่จะยอมรับพาร์ทิเคิลตัวใหม่ (p') จะต้องทำการพิจารณาค่าความน่าจะเป็น

โดยการสุ่มค่าเพื่อเปรียบเทียบกับ $\min\left\{1, \exp\left(\frac{-\Delta}{T}\right)\right\}$ หากค่าที่สุ่มได้น้อยกว่า ให้ยอมรับพาร์ทิเคิลตัวใหม่ (p') แทนที่ค่าพาร์ทิเคิลตัวเดิม (p)

- ถ้า Δ น้อยกว่าหรือเท่ากับ 0 ให้ปรับค่าพาร์ทิเคิล (p) ด้วยพาร์ทิเคิลตัวใหม่ (p') และปรับค่าอนุกรมวิธานให้ลดลง

- 4) ปรับค่าที่ดีที่สุดของแต่ละพาร์ทิเคิล (local best), ค่าที่ดีที่สุดของกลุ่ม (global best) และ ω
- 5) ปรับค่าความเร็วใหม่และตำแหน่งใหม่ของพาร์ทิเคิล
- 6) ทำซ้ำขั้นตอนที่ 2 – 5 จนครบรอบการทำงาน

การทดลองเพื่อประเมินประสิทธิภาพของโมเดล MPSO จากชุดข้อมูลมาตรฐานจำนวน 43 ชุดจาก OR-Library [12] เพื่อเปรียบเทียบกับอัลกอริทึมอื่น เช่น โมเดล HIA, โมเดล HEA และ โมเดล HGA-Param โดยสำหรับโมเดลนี้กำหนดพารามิเตอร์ดังนี้ $c1 = 2$, $c2 = 2$, $\omega_{max} = 1.4$, $\omega_{min} = 0.4$, $V_{max} = (nm) \times 0.1$, จำนวนพาร์ทิเคิล = 30 และ $Prob_{ME} = 0.01$ รันจำนวน 10 ครั้ง ครั้งละ 300 รอบ พบว่าจากข้อมูล 43 ชุด มีจำนวนข้อมูล 35 ชุดตามวิธีที่นำเสนอพบค่าที่เหมาะสมที่สุด โดยมีเปอร์เซ็นต์ค่าส่วนเบี่ยงเบนระหว่างคำตอบที่ดีที่สุด และค่าที่เหมาะสมที่สุดต่ำกว่าโมเดลอื่นๆ ดังนั้นโมเดล MPSO แสดงให้เห็นว่าการไม่ใช้วิธิตดในการกำหนดประชากรเริ่มต้นสามารถให้ค่าผลลัพธ์ที่ดีกว่าเนื่องจากไม่ต้องใช้เวลาในการคำนวณนาน ขณะที่สามารถกำหนดขนาดของประชากรเริ่มต้นและจำนวนรอบน้อยกว่าโมเดลอื่น

การทดลองเปรียบเทียบโมเดล MPSO ที่นำเสนอ กับอัลกอริทึมพาร์ทิเคิลสวอร์มแบบดั้งเดิมกับชุดข้อมูลจำนวน 11 ชุด พบว่าโมเดล MPSO สามารถพบจุดออปติ멈ได้ดีกว่า โดยพิจารณาจากค่าเฉลี่ยที่ต่ำกว่าอัลกอริทึมพาร์ทิเคิลสวอร์มดั้งเดิม จำนวน 9 ชุด และที่เหลือจำนวน 2 ชุด สามารถพบจุดที่เหมาะสมที่สุดได้เท่ากับอัลกอริทึมพาร์ทิเคิลสวอร์มดั้งเดิม ซึ่งแสดงให้เห็นว่าโมเดลที่นำเสนอสามารถเพิ่มประสิทธิภาพได้ดีกว่าอัลกอริทึมพาร์ทิเคิลสวอร์มดั้งเดิม

จากการศึกษาวิจัยนี้มีข้อสังเกต คือ โมเดลที่นำเสนอพยายามคงประสิทธิภาพการแก้ปัญหาตามอัลกอริทึมพาร์ทิเคิลสวอร์มแบบดั้งเดิม ด้วยการใส่ค่าจำนวนจริงแปลงเป็นตัวเลขไม่ต่อเนื่อง (Discrete) และสอดคล้องกับลักษณะของปัญหาการจัดตารางการผลิต ซึ่งทำให้ใช้เวลาในการแปลงค่าสำหรับการคำนวณในแต่ละตำแหน่งของพาร์ทิเคิล และในส่วนของ การปรับค่าพาร์ทิเคิลด้วยการใช้อัลกอริทึมซิมูเลทเทคแอนนิลลิงเพื่อหลีกเลี่ยงค่าตอบแบบโลคอลได้

2.3.3 An effective hybrid Particle Swarm Optimization Algorithm for multi-objective Flexible Job-Shop Scheduling Problem

Guahui Zhang, Xinyu Shao, Peigen Li และ Liang Gao [13] งานวิจัยนี้นำเสนอโมเดลที่ผสมผสานระหว่างอัลกอริทึมพาร์ทิเคิลสวอร์มและอัลกอริทึมตามู (Particle Swarm Optimization และ Tabu Search: PSO + TS) สำหรับการแก้ปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่นที่มีหลายวัตถุประสงค์ (Multi-objective) โดยอัลกอริทึมพาร์ทิเคิลสวอร์มใช้สำหรับเลือกเครื่องจักรให้กับแต่ละโอเปอเรชัน และส่วนของอัลกอริทึมตามูจะทำหน้าที่ในการค้นหาแบบโลคอล ซึ่งงานวิจัยนี้เป็นฟังก์ชันหลายวัตถุประสงค์ที่มีค่าน้อยที่สุด ซึ่งประกอบด้วย 1) เวลาในการผลิตรวมน้อยที่สุด ตามสมการ 2.10 2) เวลาทำงานของเครื่องจักรที่เสร็จช้าที่สุด (Workload of Machine) ตามสมการ 2.11 และ 3) เวลาทำงานของเครื่องจักรทั้งหมด (Total Workload of Machines) ตามสมการ 2.12 โดยการรวมวัตถุประสงค์ทั้ง 3 วัตถุประสงค์เข้าด้วยกัน จะทำให้เกิดสมการหลายวัตถุประสงค์ ตามสมการที่ 2.13

$$\min f_1 = \max_{1 \leq k \leq m} (C_k) \quad (2.10)$$

$$\min f_2 = \max_{1 \leq k \leq m} (W_k) \quad (2.11)$$

$$\min f_3 = \sum_{k=1}^m W_k \quad (2.12)$$

$$\text{Minimize}_{x \in \Omega} f(x) = (f_1(x), f_2(x), \dots, f_q(x), \dots, f_Q(x)) \quad (2.13)$$

เมื่อ Ω คือ คำตอบที่เป็นไปได้

$x = \{x_1, x_2, \dots, x_p, \dots, x_P\}$ คือ เซตของตำแหน่ง p โดย $(1 \leq p \leq P)$

โมเดล PSO + TS ที่ได้นำเสนอในงานวิจัยมีขั้นตอนการทำงาน ดังนี้

1) กำหนดค่าพารามิเตอร์

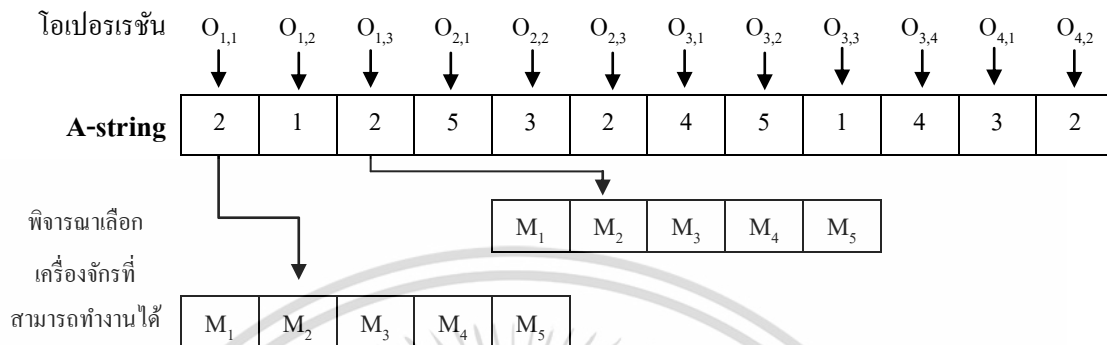
ค่าพารามิเตอร์ประกอบด้วย ขนาดของประชากร (กลุ่มพาร์ทิเคิล), จำนวนรุ่น (Gen), จำนวนรอบสูงสุดในการทำงาน (MaxIterNum), รอบปัจจุบัน (CurInerNum) และ $T = \phi$

2) สร้างประชากรเริ่มต้น

รูปแบบการเข้ารหัสพาร์ทิเคิลแบ่งได้เป็น 2 ส่วน คือ ส่วนแรก A-String แสดงเซตหมายเลขเครื่องจักรตามลำดับงานและโอเปอเรชัน ส่วนที่สอง B-String แสดงเซตโอเปอเรชัน

ตาม A-String ซึ่งความยาวของพาร์ทิเคิลเท่ากับ จำนวนโอเปอเรชันทั้งหมดของทุกงาน เช่น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ปัญหา มี 4 งาน 5 เครื่องจักร แต่ละงานมีจำนวนโอเปอเรชัน (3, 3, 4, 2) ตามลำดับ ดังนั้นขนาดความยาวของ A-String เท่ากับ 12 โอเปอเรชันและ B-String มีขนาดเท่ากับ A-String ดังรูปที่ 2.8 และ รูปที่ 2.9 ตามลำดับ



รูปที่ 2.8 การเข้ารหัสชุดคำตอบ A-String [13]



รูปที่ 2.9 การเข้ารหัสชุดคำตอบ B-String [13]

3) คำนวณค่าความเหมาะสม

สำหรับแต่ละฟังก์ชันจะคำนวณค่าความเหมาะสมจากฟังก์ชันหลายวัตถุประสงค์ ตามสมการ 2.14 ซึ่งประกอบด้วยจำนวน 3 ฟังก์ชันและในแต่ละฟังก์ชันจะกำหนดค่าน้ำหนัก (w) ซึ่งสามารถกำหนดน้ำหนักตามความสำคัญให้กับแต่ละฟังก์ชันวัตถุประสงค์

$$\text{Minimize } F = w_1 \times f_1 + w_2 \times f_2 + w_3 \times f_3 \tag{2.14}$$

เมื่อ w_1, w_2, w_3 คือ ค่าน้ำหนักตามฟังก์ชันวัตถุประสงค์แต่ละตัวซึ่งมีค่าระหว่าง 0 – 1

$$\text{Subject to: } \begin{cases} w_1 \in (0,1) \\ w_2 \in (0,1) \\ w_3 \in (0,1) \\ w_1 + w_2 + w_3 = 1 \end{cases}$$

4) ตรวจสอบเงื่อนไขการทำงานให้หยุดเมื่อพบคำตอบที่เหมาะสมที่สุด หรือครบจำนวนรอบสูงสุดในการทำงานให้หยุดการทำงาน แล้วไปขั้นตอนที่ 9 มิฉะนั้นให้ไปทำขั้นตอนที่ 5

5) เพิ่มรอบการทำงาน

6) ปรับปรุงข้อมูล (Information exchange) สำหรับครอสโอเวอร์และมิวเตชัน

การครอสโอเวอร์ A-String ด้วยการเปลี่ยนหมายเลขเครื่องจักร ด้วยการสุ่มเลือก 2 พาร์ทิเคิล (P_1 และ P_2) แล้วสุ่ม 2 ตำแหน่งบนพาร์ทิเคิลทั้งสอง จะสามารถแบ่งช่วงของพาร์ทิเคิลได้ 3 ช่วง แล้วให้คัดลอกช่วงที่ 1 และ 3 ของพาร์ทิเคิล P_1 ไปให้กับทายาท C_1 และคัดลอกช่วงที่ 2 ของพาร์ทิเคิล P_2 ไปให้กับ C_1 และสำหรับทายาท C_2 ทำลักษณะเดียวกันคือ คัดลอกช่วงที่ 1 และ 3 ของพาร์ทิเคิล P_2 ไปให้กับทายาท C_2 และคัดลอกช่วงที่ 2 ของพาร์ทิเคิล P_1 ไปให้กับ C_2

การครอสโอเวอร์ B-String ด้วยการเปลี่ยนลำดับโอเปอร์เรชันในการจัดตาราง ด้วยการแบ่งเป็น 2 กลุ่ม (JG_1 และ JG_2) แล้วสุ่มแบ่งงานให้แต่ละกลุ่มจำนวนเท่าๆ กัน เช่น $JG_1 = \{J_1, J_2\}$ และ $JG_2 = \{J_3, J_4\}$ จากนั้นสร้างทายาท (C_1) โดยการคัดลอกทุกโอเปอร์เรชันของงานในกลุ่ม JG_1 ไปให้กับ C_1 ในตำแหน่งเดิม ที่เหลือโอเปอร์เรชันของงานที่เหลือ (กลุ่ม JG_2) มาใส่ตามลำดับ

การมิวเตชัน B-String สุ่มตำแหน่งบนพาร์ทิเคิลแต่ละตัว 2 ตำแหน่ง แล้วเรียงลำดับพาร์ทิเคิลในช่วงที่สุ่มแบบย้อนกลับในจากนั้นนำพาร์ทิเคิลที่ปรับปรุงไปแทนที่พาร์ทิเคิลตัวเดิม

7) นำแต่ละพาร์ทิเคิลมาปรับปรุงด้วยการค้นหาแบบตามูซึ่งมีขั้นตอนดังนี้

7.1) กำหนดค่าพารามิเตอร์เริ่มต้นเพื่อใช้กับการค้นหาแบบตามูได้แก่

- s คือ คำตอบที่เป็นไปได้ของปัญหา
- $V(s)$ คือ ชุดคำตอบจากพื้นที่รอบข้างของ s
- $ObjFun(s)$ คือ ค่าฟังก์ชันวัตถุประสงค์ของคำตอบที่เป็นไปได้ (s)
- $ObjFun^*$ คือ ค่าฟังก์ชันวัตถุประสงค์ที่เลือกของรอบปัจจุบัน
- s^* คือ คำตอบที่ดีที่สุด
- $ObjFun(s^*)$ คือ ค่าฟังก์ชันวัตถุประสงค์ของคำตอบที่ดีที่สุด
- $MaxIterNum$ คือ จำนวนรอบการทำงาน
- $T = \emptyset$ คือ กำหนดค่าเริ่มต้นของหน่วยความจำของตามู (tabu list)
- $ObjFun^* = ObjFun(s^*)$ คือ ค่าฟังก์ชันวัตถุประสงค์ของรอบปัจจุบันเท่ากับค่าฟังก์ชันวัตถุประสงค์
- $s = s^*$ คือ ชุดคำตอบของรอบปัจจุบันมีค่าเท่ากับชุดคำตอบที่ดีที่สุด

7.2) เพิ่มรอบการทำงาน และสร้างชุดคำตอบจากพื้นที่รอบข้าง $V(s)$ จากการสลับโอเปอร์เรชันที่อยู่บนเส้นวิกฤตของชุดคำตอบที่เลือกในรอบนั้น

7.3) ประเมินค่าฟังก์ชันของคำตอบปัจจุบันจากการเลือกคำตอบที่ดีที่สุดของเซต

คำตอบจากพื้นที่รอบข้าง $v' \in V(s)$ ได้คำตอบใหม่ s' และ T' แล้วทำการปรับค่า $s = s'$ และ $T = T'$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4) ถ้าคำตอบ (s) น้อยกว่าคำตอบที่ดีที่สุดที่มีอยู่ (s*) ให้เลือกคำตอบในรอบนั้นปรับค่าผลลัพธ์คำตอบเป็นค่าที่ดีที่สุด (s*) มิฉะนั้นให้ไปทำข้อที่ 5

7.5) ให้ทำขั้นตอนที่ 7.2 จนครบเงื่อนไขการทำงาน และแสดงคำตอบที่ดีที่สุด (s*)

8) ประเมินค่าฟาร์ทิกิลของแต่ละรอบ เพื่อปรับค่าที่ดีที่สุดของแต่ละฟาร์ทิกิลและค่าที่ดีที่สุดของกลุ่ม และทำขั้นตอนที่ 4

9) แสดงผลลัพธ์

การทดลองเพื่อประเมินประสิทธิภาพของโมเดล PSO + TS ที่นำเสนอด้วยชุดข้อมูล 4 ชุด ซึ่งมีขนาด 4x5, 8x8, 10x10, 15x10 และกำหนดขนาดของฟาร์ทิกิล ($P_s = 100$), รอบการทำงาน ($Gen = 50$), จำนวนรอบสูงสุดของ TS ($MaxIterNum = 50$) โดยเปรียบเทียบกับอัลกอริทึมอื่นพบว่าโมเดลที่นำเสนอมีประสิทธิภาพดีกว่า

ข้อสังเกตของงานวิจัยนี้ โมเดลที่นำเสนอแบ่งเป็น 2 ส่วน คือ 1) ส่วนของอัลกอริทึมฟาร์ทิกิลสวอร์ม ซึ่งออกแบบการเข้ารหัสชุดคำตอบเป็นรูปแบบของตัวเลขแบบไม่ต่อเนื่อง และการนำโครอสโอเวอร์มาใช้แทนการปรับตำแหน่งและความเร็วของฟาร์ทิกิล ซึ่งเป็นการปรับปรุงจากอัลกอริทึมฟาร์ทิกิลสวอร์มแบบดั้งเดิมซึ่งคล้ายวิธีการของจินตออัลกอริทึมมากกว่า ส่วนที่ 2) นำเอาการค้นหาแบบตามาปรับปรุงเพื่อสร้างความหลากหลายในการค้นหาคำตอบที่ดีขึ้นจากพื้นที่รอบข้างซึ่งขึ้นอยู่กับค่าพารามิเตอร์ที่กำหนดเช่นกำหนดค่าความจำของตาบู่ (Tabu list)

2.3.4 Knowledge-Based Ant Colony Optimization for Flexible Job-Shop Scheduling

Problem

Li-Ning Xing, Ying-Wu Chen, Peng Wang, Qing-Song Zhao และ Jian Xiong [14] ได้นำอัลกอริทึมทางวิวัฒนาการ (Evolutionary Algorithm) เพื่อแก้ปัญหาร่วมกับอัลกอริทึมอาณานิคมมด (Ant Colony Optimization: ACO) ซึ่งถูกคิดค้นโดย Dorigo et al. [15] เป็นการเลียนแบบพฤติกรรมของการหาอาหารของมดโดยจะค้นหาเส้นทางที่สั้นที่สุดระหว่างรังกับแหล่งอาหารซึ่งอาศัยค่าฟีโรโมน (Pheromone) ในการสื่อสารกับมดตัวอื่น และเส้นทางที่ถูกเลือกซ้ำ ปริมาณฟีโรโมนจะสูงขึ้นขณะเดียวกันหากเส้นทางนั้นไม่ถูกเลือกหรือเลือกน้อย ค่าฟีโรโมนจะมีระเหย และเส้นทางที่มดเดินผ่านจะมีการปรับค่าปริมาณของฟีโรโมนทุกครั้งเพื่อเหลือเส้นทางที่ดีที่สุด

งานวิจัยนี้เป็นการรวมความรู้ และการค้นหาแบบฮิวริสติก เรียกว่า Knowledge-Based Ant Colony Optimization: KBACO) นำมาใช้ในการแก้ปัญหากับงานการผลิตที่เป็นการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น โดยมีฟังก์ชันวัตถุประสงค์คือ เวลาในการผลิตรวมน้อยที่สุด

โมเดล KBACO ที่ได้นำเสนอในงานวิจัยมีขั้นตอนการทำงาน ดังนี้

1) การเริ่มต้นความรู้ (Knowledge Initialization)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโมเดล KBACO เป็นวิธีการทำงานที่อาศัยความรู้และเรียนรู้ไปพร้อมกันระหว่างกระบวนการทำงานในแต่ละรอบการทำงาน ซึ่งความรู้เพื่อทำการเรียนรู้ประกอบด้วย 4 ส่วน คือ ความรู้จากการคัดเลือกกลุ่มที่ดี ความรู้จากการดำเนินการของเครื่องจักร ความรู้จากการดำเนินการตามลำดับที่เลือก และความรู้จากการกำหนดค่าพารามิเตอร์ ดังรูปที่ 2.15 สามารถอธิบายแต่ละส่วนดังนี้

1.1) ความรู้จากการคัดเลือกส่วนหรือกลุ่มที่ดี (Elite Solution Knowledge) ขั้นตอนนี้จะคัดเลือกมดที่ให้ค่าความเหมาะสมที่ดีที่สุดจำนวนหนึ่งจากมดทั้งหมดในแต่ละรอบเพื่อเก็บเป็นมดกลุ่มที่ดี (Elite Solution Set) โดยในแต่ละรอบจะมีการปรับค่าเมื่อค่าความเหมาะสมของรอบปัจจุบันดีกว่าค่าความเหมาะสมรอบที่ผ่านมา

1.2) ความรู้จากการเลือกเครื่องจักร (Operation Assignment Machine Knowledge: OAMK) เป็นการสะสมความรู้ของแต่ละโอเปอเรชันที่เลือกเครื่องจักรทำงาน การดำเนินการนี้จะมีปรับค่าเมื่อพบคำตอบที่ดีที่สุดในวงกว้าง (Global best Solution) การกำหนดค่าเริ่มต้นของค่าดำเนินการคำนวณได้จากสมการ 2.15 ดังนี้

$$OAMK(i,j,k) = \begin{cases} 0 & O_{ij} \text{ ไม่ได้ทำบนเครื่องจักร } k \\ 1/p_{i,j,k} & O_{ij} \text{ ทำบนเครื่องจักร } k \end{cases} \quad (2.15)$$

เมื่อ $p_{i,j,k}$ คือเวลาทำงานของงานที่ i โอเปอเรชัน j บนเครื่องจักร k

1.3) ความรู้จากการเลือกโอเปอเรชัน (Operation Assignment Priority Knowledge: OAPK) เป็นการสะสมความรู้ของโอเปอเรชันที่ถูกเลือกพิจารณาตามลำดับความสำคัญในการเลือก (สามารถกำหนดระดับความสำคัญได้) การดำเนินการนี้จะมีปรับค่าเมื่อพบคำตอบที่ดีที่สุด ในวงกว้าง กำหนดค่าเริ่มต้นสำหรับการเลือกของแต่ละโอเปอเรชันให้มีค่าน้อยๆ เท่ากันๆ กัน เช่นกำหนดให้ $\tau_0 = 0.01$ เป็นต้น

1.4) ความรู้จากการกำหนดค่าพารามิเตอร์ทำงาน (Parameter Knowledge) เป็นการกำหนดค่าพารามิเตอร์การทำงานในแต่ละรอบ ซึ่งใช้วิธีการสุ่มจากเซตของชุดพารามิเตอร์ที่สร้างจากวิธี Orthogonal Design Technology เป็นการพิจารณาค่าพารามิเตอร์ที่มีความเป็นไปได้เพื่อลดเวลาในการทดสอบ โดยกำหนดให้ค่าเริ่มต้นของพารามิเตอร์แต่ละชุดเท่ากับ 1 เพื่อให้มีโอกาสในการถูกเลือกเท่าๆ กัน เมื่อได้คำตอบที่ดีที่สุด ในวงกว้าง หรือได้คำตอบที่ดีกว่าเดิมจะปรับค่าจำนวนความถี่ของการเลือกชุดพารามิเตอร์ชุดนั้นๆ

2) กำหนดค่าโอเปอเรชัน (Operation Assignment)

คือ การกำหนดเครื่องจักรที่เป็นไปได้ให้กับแต่ละโอเปอเรชันมีขั้นตอน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1) กำหนดชุดคำตอบเริ่มต้นของมดแต่ละตัว เช่น [1 1 2 2 2] แสดงหมายเลขงาน

2.2) คำนวณความน่าจะเป็นของโอเปอร์เรชัน O_{ij} บนเครื่องจักร k โดยให้พิจารณาเครื่องจักรที่สามารถทำได้ตามสมการ 2.16

$$\Pr(i,j,k) = \frac{OAMK(i,j,k)}{\sum_{k=1}^m OAMK(i,j,k)} \quad (2.16)$$

เมื่อ $\Pr(i,j,k)$ คือ ความน่าจะเป็นของโอเปอร์เรชัน O_{ij} บนเครื่องจักร k
 $OAMK(i,j,k)$ คือ โอเปอร์เรชัน O_{ij} บนเครื่องจักร k

2.3) สร้างวงล้อจากความน่าจะเป็นสะสมของเครื่องจักรที่มีความเป็นไปได้ ทั้งหมดที่ได้จากสมการ 2.16 และทำการสุ่มตัวเลข $[0 - 1]$ เพื่อเลือกเครื่องจักร

2.4) ทำขั้นตอนที่ 2.1 – 2.3 จนครบทุกโอเปอร์เรชัน

3) จัดลำดับโอเปอร์เรชัน (Operation Sequencing)

พิจารณาโอเปอร์เรชันตามอันดับการเลือก (Priority) มีขั้นตอนดังนี้

3.1) เลือกโอเปอร์เรชันที่สามารถเริ่มทำได้โดยไม่รอโอเปอร์เรชันอื่นเสร็จ

3.2) กำหนดเซตของโอเปอร์เรชันที่สามารถทำงานต่อจากโอเปอร์เรชันที่ทำไปแล้ว

3.3) พิจารณาความน่าจะเป็นของโอเปอร์เรชันจากเซตโอเปอร์เรชันทั้งหมดที่เป็นไปได้ตามสมการ 2.17 และสร้างวงล้อจากความน่าจะเป็นสะสมของโอเปอร์เรชันที่เป็นไปได้ จากนั้นสุ่มค่าตัวเลขที่มีค่าระหว่าง $[0 - 1]$ เพื่อเลือกโอเปอร์เรชัน

$$\Pr(i, j, k, t) = \frac{[OAPK(i, j, m)]^\alpha \times [1/P_{i,j,m}]^\beta}{\sum_{O_{ij} \in \text{allow}(m,t)} \{[OAPK(i, j, m)]^\alpha \times [1/P_{i,j,m}]^\beta\}} \quad (2.17)$$

เมื่อ $\Pr(i,j,m,t)$ คือ ความน่าจะเป็นของโอเปอร์เรชัน O_{ij} บนเครื่องจักร m เวลา t

$OAPK(i,j,m)$ คือ โอเปอร์เรชัน O_{ij} บนเครื่องจักร m

α, β คือ ค่าน้ำหนักในการเรียนรู้ของ OAPK

$\text{allow}(m,t)$ คือ เซตของโอเปอร์เรชันที่เป็นไปได้บนเครื่องจักร m เวลา t

3.4) ทำขั้นตอนที่ 3.1 – 3.3 จนครบทุกโอเปอร์เรชัน จะได้คำตอบของแต่ละชุดคำตอบจำนวน N ชุด (มดทุกตัว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ปรับปรุงตารางการผลิต (Scheduling Improving) คือการปรับค่าชุดคำตอบเพื่อให้พบคำตอบที่ดีที่สุด ขั้นตอนนี้ทำเฉพาะกลุ่มคำตอบที่ดี (Elite set) ที่มีค่าความเหมาะสมที่ดีที่สุดจากคำตอบที่ทั้งหมดในรอบปัจจุบัน การปรับปรุงชุดคำตอบแบ่งเป็น 2 ส่วน คือ 1) ส่วนของลำดับโอเปอเรชัน (Operation Order) และ 2) ส่วนของเครื่องจักรที่เลือก (Machine Selection) สามารถแสดงตามรูปที่ 2.10

	Operation Order					Machine Selection				
ชุดคำตอบ	1	1	2	2	2	1	2	1	2	3

รูปที่ 2.10 รูปแบบของคำตอบ [14]

4.1) ปรับค่าส่วนของลำดับโอเปอเรชัน (Operation Order)

ใช้หลักการครอส 2 จุด (Two-point Crossover) ซึ่งเป็นการสุ่ม 2 ตำแหน่งในชุดคำตอบ A จากนั้นคัดลอกโอเปอเรชันที่อยู่ในช่วงตำแหน่งดังกล่าวไปยังชุดคำตอบใหม่ จากนั้นลบตำแหน่งโอเปอเรชันที่คัดลอกในชุดคำตอบ B แล้วคัดลอกโอเปอเรชันส่วนที่เหลือในชุดคำตอบ B ไปใส่ในชุดคำตอบใหม่ตามลำดับที่ว่างจากซ้ายไปขวา

4.2) ปรับค่าส่วนของการเลือกเครื่องจักร (Machine Selection)

ใช้การสุ่มค่า 2 ตำแหน่งในชุดคำตอบ A เพื่อกำหนดช่วงของโอเปอเรชัน และให้ทำการเปลี่ยนตำแหน่ง (exchanged) กับชุดคำตอบ B แทนชุดคำตอบเดิม

5) การเรียนรู้จากความรู้ (Knowledge learning) เป็นการปรับค่าตามที่พบคำตอบในแต่ละรอบ ดังนี้

5.1) ปรับค่าชุดคำตอบที่ดีในรอบปัจจุบันแทนที่ชุดคำตอบเดิมในรอบก่อนหน้า

5.2) ปรับค่าตามกฎการปล่อยและการระเหยของฟีโรโมนในแต่ละรอบจากคำตอบที่ดีที่สุดในช่วงกว้าง (Global best solution) จากหลักการของ Max-Min Ant System (MMAS) ที่มีกำหนดระดับช่วงของฟีโรโมน $[\tau_{\min}, \tau_{\max}]$ ตามสมการ ดังนี้

▪ กฎการปล่อยฟีโรโมน

$$OAMK(i, j, k) = OAMK(i, j, k) \times Q_1 \quad (2.18)$$

$$OAPK(i, j, k) = OAPK(i, j, k) \times Q_2 \quad (2.19)$$

เมื่อ Q_1 คือ สัมประสิทธิ์การเพิ่มขึ้นของ QAMK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Q₂ คือ การเพิ่มขึ้นของระดับ QAPK

■ กฎการระเหย

$$OAMK(i,j,k) = \min\{\tau_{\max}, \max\{\tau_{\min}, (1-\rho)OAMK(i,j,k)\}\} \quad (2.20)$$

$$OAPK(i,j,k) = \min\{\tau_{\max}, \max\{\tau_{\min}, (1-\rho)OAPK(i,j,k)\}\} \quad (2.21)$$

เมื่อ ρ คือ ความรู้ในการระเหย

5.3) ในแต่ละรอบถ้าพบคำตอบที่ดีที่สุดในช่วงกว้างให้เพิ่มความถี่ของพารามิเตอร์ชุดดังกล่าว +1 ดังนั้นชุดพารามิเตอร์ที่มีจำนวนความถี่มากที่สุดแสดงถึงพารามิเตอร์ชุดนั้นมีประสิทธิภาพ

การทดสอบโมเดล KBACO ที่นำเสนอถูกแบ่งเป็น 3 วิธีการ คือ วิธี 1) KBACO แบบไม่ใช้การปรับค่าในส่วนของการเลือกชุดคำตอบที่ดี และไม่มีการเรียนรู้ของพารามิเตอร์ วิธี 2) KBACO แบบไม่ใช้การปรับค่าในส่วนของการเลือกชุดคำตอบที่ดี และวิธีที่ 3) อัลกอริทึม KBACO ซึ่งทั้ง 3 วิธี พิจารณาเวลาทำงานที่เหมาะสมกับเวลาที่ใช้โดยเฉลี่ย พบว่าวิธีที่ 3 เป็นวิธีที่พบคำตอบที่เหมาะสมโดยใช้เวลาเฉลี่ยในการค้นหาที่น้อยที่สุด เมื่อเปรียบเทียบกับ 2 วิธีแรก และเมื่อนำไปเปรียบเทียบกับอัลกอริทึมอื่น พบว่าโมเดล KBACO ได้ผลดีกว่าเช่นกัน ซึ่งถือว่าโมเดลที่นำเสนอเป็นวิธีการที่มีการเรียนรู้จากค่าความเหมาะสมจาก ACO ไปพร้อมกัน แล้วปรับใช้ความรู้ในการค้นหา และเรียนรู้ จนสุดท้ายได้ผลลัพธ์ที่มีคุณภาพ และโมเดลดังกล่าวยังใช้หน่วยความจำเพียงเล็กน้อยในการทำงาน

จากการศึกษางานวิจัยนี้มีข้อสังเกต ดังนี้โมเดล KBACO ได้นำวิธีการเรียนรู้พฤติกรรมของมดโดยอาศัยฟีโรโมนในการเลือกเส้นทาง และใช้วิธีการเรียนรู้จากการได้คำตอบที่ดีขึ้นด้วยการปรับค่าพารามิเตอร์ที่เหมาะสมกับปัญหา ซึ่งวิธีการดังกล่าวทำให้การค้นหาคำตอบเข้าสู่เร็วขึ้น ทำให้พบกับคำตอบแบบโลคอลมากกว่าคำตอบแบบโกลบอลได้

2.3.5 An Artificial Immune Algorithm for the Flexible Job-Shop Scheduling

Problem

A. Bagheri, M. Zandieh, Iraj Mahdavi and M. Yazdani [16] งานวิจัยนี้นำเสนอโมเดล AIA สำหรับแก้ปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น ซึ่งใช้อัลกอริทึมภูมิคุ้มกันโรค (Artificial Immune Algorithm) เป็นการเลียนแบบกลไกการทำงานของร่างกายของมนุษย์ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ธรรมชาติที่มีระบบในการปกป้องร่างกายจากสิ่งแปลกปลอม (Antigen) หรือเชื้อโรคต่างๆ ที่เข้ามาในร่างกาย โดยสามารถจดจำและจำแนกกับความแตกต่างของสิ่งผิดปกติในลักษณะต่างๆ ได้อย่างถูกต้องและยังสามารถฆ่าเชื้อโรคเหล่านั้นเรียกว่าภูมิคุ้มกันหรือแอนติบอดี (Antibody) ดังนั้นการสร้างแอนติบอดีในร่างกายใช้หลักการของการคัดเลือกกลุ่มพันธุ์ (Clonal Selection Theory) ซึ่งถูกเสนอโดย Burnet, 1978 [17]

งานวิจัยนี้ได้ปรับการทำงานของอัลกอริทึมภูมิคุ้มกันโรคสำหรับแก้ปัญหาการจัดกาตารางการผลิตแบบตามสั่งยืดหยุ่น โดยเสนอวิธีสร้างประชากรเริ่มต้น การเลือกประชากรเพื่อสร้างเป็นรุ่นลูกและกระบวนการมิวเตชัน โดยมีฟังก์ชันวัตถุประสงค์คือเวลาในการผลิตรวมน้อยสุด

โมเดล AIA ที่ได้นำเสนอในงานวิจัยมีขั้นตอนการทำงาน ดังนี้

1) กำหนดค่าพารามิเตอร์และประชากรเริ่มต้น

ค่าพารามิเตอร์ประกอบด้วย จำนวนของประชากร(N) จำนวนการมิวเตชันในแต่ละรอบ จำนวนแอนติบอดีที่สามารถเปลี่ยนในแต่ละรอบ ความน่าจะเป็นของการมิวเตชัน จำนวนรอบการทำงานและสัดส่วนของการสร้างประชากรด้วยวิธี AssignmentRule1 และวิธี AssignmentRule2 และวิธีการจัดลำดับโอเปอร์เรชัน คือ วิธี (Most Work Remaining: MWR) และวิธี (Most Operation Remaining: MOR)

การกำหนดประชากรเริ่มต้น โมเดลนี้กำหนดวิธีการประชากรเริ่มต้นด้วยการแบ่งเป็น 2 ขั้นตอน คือ 1) กำหนดเครื่องจักรให้กับแต่ละโอเปอร์เรชัน ด้วยวิธีการ AssignmentRule1 และ AssignmentRule2 จากวิธีนำเสนอของ Pezzella et al.[9] 2) การจัดลำดับโอเปอร์เรชันจากข้อ 1 ด้วยวิธีการสุ่ม วิธีเลือกโอเปอร์เรชันที่มีเวลาในการผลิตเหลือมากที่สุด (MWR) และวิธีเลือกโอเปอร์เรชันที่เหลือมากที่สุด (MOR)

การกำหนดตัวแทนของแอนติบอดีในงานวิจัยนี้ประกอบด้วย (i, j, k) สำหรับแต่ละโอเปอร์เรชัน ซึ่ง i คืองาน ส่วน j คือลำดับของโอเปอร์เรชันของงาน i และ k คือเครื่องจักรทำงานบนโอเปอร์เรชันนั้นตามรูปที่ 2.11 ชุดคำตอบนี้สามารถประเมินโดยจัดลำดับด้วยแผนภาพแกนต์จะมีเวลาในการผลิตรวมเท่ากับ $23 (C_{max})$

S:	(3,1,2)	(2,1,1)	(3,2,3)	(4,1,1)	(2,2,4)	(4,2,3)	(1,1,4)	(1,2,1)	(4,3,2)	(2,3,3)
----	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

รูปที่ 2.11 ตัวแทนของแอนติบอดี [16]

2) คำนวณการหาค่าฟิตเนสของแต่ละแอนติบอดี (เวลาในการผลิตรวม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ประเมินความสัมพันธ (Affinity) ของแอนติบอดีด้วยสมการ 2.22 โดย Z หมายถึงชุดคำตอบของแต่ละแอนติบอดี

$$\text{Affinity}(Z) = \frac{1}{\text{makespan}(z)} \quad (2.22)$$

4) การคัดเลือกกลุ่มพันธุ์

4.1) เลือกแอนติบอดีจำนวน M ตัว จากประชากรทั้งหมดโดยสุ่มจากจากความสัมพันธของแอนติบอดีที่มีค่าฟิตเนสสูงสุด

4.2) สำเนาแอนติบอดีจากข้อ 4.1 ด้วยวิธีการของ ไบนารีทัวร์นาเมนต์ (Binary Tournament rule) คือเลือกมาทีละ 2 แอนติบอดีแล้วทำการเลือกตัวที่ดีกว่าจนครบจำนวน M ตัว

5) การคัดเลือกประชากรรุ่นต่อไป

5.1) สุ่มแอนติบอดีเพื่อทำการมิวเตจจากกลุ่มพันธุ์ที่สำเนา จำนวน (nm) ตัวเพื่อสร้างแอนติบอดีใหม่จากวิธีการมิวเตจชั้นกำหนด 4 วิธี คือ

a) Intelligent Mutation: เป็นการสุ่มเลือกโอเปอเรชันบนเครื่องจักรที่มีเวลาการทำงานมากที่สุด และทำการย้ายโอเปอเรชันนั้นไปทำงานบนเครื่องจักรที่มีเวลาทำงานน้อยสุด โดยอ้างอิงจาก [9]

b) Assignment Mutation: สุ่มเลือกโอเปอเรชัน 2 ตำแหน่งเพื่อแลกเปลี่ยนการทำงานโดยลำดับโอเปอเรชันยังคงเดิม

c) Precedence Preserving Shift Mutation (PPS): เป็นการจัดลำดับโอเปอเรชันคือสุ่มเลือก 1 โอเปอเรชัน และพิจารณาทุกตำแหน่งที่โอเปอเรชันนั้นสามารถแทรกได้ แล้วทำการสุ่มตำแหน่งที่เป็นไปได้มา 1 ตำแหน่งเพื่อทำการแทรก

d) Reordering mutation: เป็นการจัดลำดับโอเปอเรชันโดยสุ่มเลือกช่วงตำแหน่งโอเปอเรชันในแอนติบอดี $(p1, p2)$ แล้วเปลี่ยนจัดลำดับตำแหน่งใหม่ที่เป็นไปได้จากวิธีการสุ่ม

5.2) เพิ่มแอนติบอดีใหม่หลังจากมิวเตจจำนวน nm ตัวเข้าไปรวมในแอนติบอดีปัจจุบัน

5.3) สร้างแอนติบอดีใหม่ด้วยวิธีเดียวกับการสร้างประชากรเริ่มต้นจำนวน nea ตัวแทนที่แอนติบอดีที่มีความสัมพันธแย่งสุด

5.4) คัดลอกแอนติบอดีตัวที่ดีที่สุดไปเป็นประชากรรุ่นต่อไป

5.5) คัดเลือกแอนติบอดีที่เหมาะสมจำนวนเท่ากับ $N-1$ ไปเป็นประชากรรุ่นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) ให้ทำงานครบตามเงื่อนไข มิฉะนั้นให้ทำงานรอบต่อไป

งานวิจัยนี้ได้ทดสอบโมเดลที่นำเสนอกับข้อมูลจำนวน 4 ชุด คือ 1) Kacem [19] 2) Fdata [10] 3) BRdata [20] และ 4) HUdata [21] โดยกำหนดค่าพารามิเตอร์แตกต่างกันตามขนาดของข้อมูลในแต่ละชุด ได้แก่ จำนวนประชากร จำนวนรอบการทำงาน จำนวนแอนติบอดีที่ในการโคลน จำนวนแอนติบอดีที่ถูกมิวเตชัน และจำนวนแอนติบอดีที่จะถูกแทนที่ในแต่ละรอบ ผลการทดลองพบว่า โมเดล AIA ที่นำเสนอในงานวิจัยนี้มีประสิทธิภาพในการค้นหาคำตอบดีกว่าอัลกอริทึมที่นำมาเปรียบเทียบกับค่าความเบี่ยงเบนสัมพัทธ์ (Relative Deviation: Dev) มากกว่าและค่าความคลาดเคลื่อนสัมพัทธ์ (Relative Error: RE) ที่ต่ำกว่า

จากการศึกษาวิจัยนี้มีข้อสังเกตดังนี้ การคัดเลือกจำนวนแอนติบอดีตัวที่ดีจำนวน M ตัว เพื่อคัดลอกอาจได้แอนติบอดีตัวเดิมในรอบหลายๆ ของการทำงานตามอัลกอริทึมซึ่งทำให้เกิดความหลากหลายในการค้นหาคำตอบและอาจจะทำให้ได้คำตอบแบบโลคอล ส่วนการมิวเตชันที่มีหลายวิธีทำให้เกิดความหลากหลายในการมิวเตชันเพื่อปรับค่าในแอนติบอดี

2.3.6 An improved Genetic Algorithm for the Distributed and Flexible Job-shop

Scheduling Problem

L. De Giovanni และ F. Pezzella [22] ได้นำเสนอโมเดล IGA (Improved Genetic Algorithm) เพื่อปรับปรุงตารางการผลิตแบบตามสั่งยืดหยุ่นและมีการกระจาย โดยนำจีเนติกอัลกอริทึม และการค้นหาแบบโลคอลมาใช้ในการแก้ปัญหาในระบบการผลิตแบบ DFJS ซึ่งเป็นปัญหาที่มีความซับซ้อนกว่าการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น เนื่องจากมีกระบวนการผลิตขนาดใหญ่และมีจำนวนโรงงานหลายโรงงาน (Multi-Factory) ที่มีการผลิตสินค้าที่แตกต่างกันและเป็นการผลิตกึ่งสำเร็จ งานวิจัยมีวัตถุประสงค์ คือ เวลาในการผลิตรวมน้อยที่สุด

ลักษณะของปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่นและมีการกระจาย ที่เป็นกระบวนการผลิต ระบบสายพานจะดำเนินวัตถุดิบจากส่วนของ I/O Center ไปให้โรงงานที่ l ทำงาน เมื่อทำงานเสร็จจะส่งผลลัพธ์กลับไปให้ I/O Center โดยในแต่ละโรงงานมีเครื่องจักรจำนวนหลายเครื่องที่สามารถทำงานให้กับหนึ่งงาน และในแต่ละงานประกอบด้วยโอเปอเรชันจำนวนหนึ่งโดยในกระบวนการผลิตจะทำการผลิตหลายๆ งานไปพร้อมๆ กัน ดังนั้นจำนวนงานสำหรับการผลิตประกอบด้วย $J = \{j_1, j_2, \dots, j_i, \dots, j_n\}$ และในแต่ละงานจะทำงานเป็นอิสระจากกัน จำนวนโรงงานสำหรับการผลิตประกอบด้วย $U = \{U^1, U^2, \dots, U^i, \dots, U^q\}$ ซึ่งในแต่ละโรงงานที่ l ประกอบด้วยเซตของเครื่องจักร ประกอบด้วย $M^l = \{M^{l1}, M^{l2}, \dots, M^{lk}, \dots, M^{lm}\}$ และลำดับของโอเปอเรชันของงาน j_i ที่ทำงานบน โรงงานที่ U^l สามารถจัดลำดับโอเปอเรชันของงานประกอบด้วย $(O_{i1}^l, O_{i2}^l, \dots, O_{ij}^l, \dots, O_{in}^l)$ และการทำงานของงานที่ i และโรงงาน (FMU) ที่ l

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้มีระยะเวลาการส่งมอบแก่ลูกค้า d_i ซึ่งจะถูกรวมอยู่ในระยะเวลาการผลิตหลังจาก ระยะเวลาทำงานเสร็จสิ้นของเครื่องจักรในแต่ละโรงงาน

โมเดล IGA ที่ได้นำเสนอในงานวิจัยมีขั้นตอนการทำงาน ดังนี้

1) กำหนดรูปแบบการเข้ารหัสชุดคำตอบของโครโมโซม

กำหนดรูปแบบโครโมโซมตามลักษณะโอเปอร์เรชัน ซึ่งในแต่ละโครโมโซมประกอบด้วย จำนวนยีนเท่ากับจำนวนโอเปอร์เรชันของทุกงานรวมกัน ตัวอย่างเช่น มีงานจำนวน 5 งาน ในแต่ละ งานมีจำนวนโอเปอร์เรชันคือ [3 2 3 1 2] ตามลำดับ รวมจำนวนโอเปอร์เรชันเท่ากับ 11 โอเปอร์เรชัน เป็นต้น ดังนั้นหนึ่งโครโมโซมมีจำนวนยีนเท่ากับ 11 ยีน

2) กำหนดประชากรเริ่มต้นจำนวน N โครโมโซม

ในแต่ละยีนประกอบด้วย 2 พารามิเตอร์ (x,y) คือ หมายเลขโรงงาน (x) และหมายเลขงาน (y) ซึ่งได้จากกลุ่มตามเงื่อนไขที่เป็นไปได้ เช่นกำหนดให้ $U_1 = \{J_1, J_3\}$, $U_2 = \{J_2, J_4\}$, $U_3 = \{J_5\}$ หมายถึง โรงงานที่ 1 ทำงานที่ 1 และงานที่ 3 ส่วนโรงงานที่ 2 ทำงานที่ 2 และ 4 และ โรงงานที่ 3 ทำงาน 5 ดังนั้นในรูปที่ 2.12 ตำแหน่งแรกหมายถึง โรงงานที่ 1 ทำงานที่ 3 โอเปอร์เรชันที่ 1 เป็นต้น

1,3	2,2	2,2	1,1	3,5	2,4	1,1	3,5	1,1	1,3	1,3
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

รูปที่ 2.12 การเข้ารหัสชุดคำตอบโครโมโซม [22]

จากนั้นเลือกเครื่องจักรทำงานให้กับแต่ละ โอเปอร์เรชันหรือยีน กำหนดให้เลือกเครื่องจักร ที่ใช้เวลาการทำงานที่น้อยที่สุดเริ่มจาก โอเปอร์เรชันซ้ายสุดไปขวาสุด

3) คำนวณค่าความเหมาะสมของโครโมโซมจากแผนภาพแกนต์

4) คัดเลือกโครโมโซม

ใช้วิธี Linear ranking คือการเรียงโครโมโซมตามค่าความเหมาะสมหรือฟิตเนสจากแย่ที่สุด ไปดีที่สุด และกำหนดความน่าจะเป็นตามตำแหน่งการเรียงลำดับของโครโมโซมตามสมการที่ 2.23

$$P_i = \frac{2\sigma_i}{N(N+1)} \quad (2.23)$$

เมื่อ $i = 1, \dots, N$

σ_i หมายถึง ลำดับของโครโมโซมที่เรียงลำดับตามค่าความเหมาะสม

5) สร้างทายาท เพื่อหาโครโมโซมที่ดีกว่าเดิมสามารถทำได้ 3 ขั้นตอน ประกอบด้วย การครอสโอเวอร์ และการมิวเตชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5.1) คrossover โครโมโซมแบบชนิด crossover 2 จุด
 - 5.2) มิวเทชันแบบโลคอล คือ สุ่มตำแหน่งโอเปอร์เรชันในโครโมโซม 2 ตำแหน่ง แล้วสลับโอเปอร์เรชันจำนวน ρ_q^L ครั้ง
 - 5.3) มิวเทชันเครื่องจักร (Machine Mutation) คือ เลือกเครื่องจักรใหม่แทนเครื่องจักรเดิมโดยวิธีการสุ่มจำนวน ρ_q^M ครั้ง
 - 6) มิวเทชันแบบโกลบอล (Global Mutation) คือ เปลี่ยนโรงงานใหม่ให้กับงาน ทุกโอเปอร์เรชันของงานที่สุ่มจะถูกเปลี่ยนโรงงานสำหรับทำงานจำนวน ρ_q^G ครั้ง
 - 7) มิวเทชันแบบละเอียด (Refinement by Local Search) เลือกโอเปอร์เรชันในโรงงานที่มีเวลาทำงานมากที่สุดแล้วสลับตำแหน่งโอเปอร์เรชันและเลือกเครื่องจักรใหม่จำนวน ρ_q^R ครั้ง
 - 8) ทำการเลือกโครโมโซมที่ดีที่สุด N โครโมโซมไปเป็นรุ่นถัดไป
 - 9) หยุดการทำงานเมื่อครบจำนวนรอบที่กำหนดไว้ หรือพบคำตอบที่เป็นจุดออปติ멈และค่าผลลัพธ์ที่ได้ไม่เปลี่ยนแปลง หากไม่เป็นไปตามเงื่อนไขให้กลับไปทำขั้นตอนที่ 3 – 8
- ผลการทดลองเพื่อประเมินประสิทธิภาพของ โมเดล IGA โดยเปรียบเทียบระหว่างการใช้ Refinement และ ไม่ใช้ Refinement ด้วยการรัน 50 รอบ พบว่าการใช้ Refinement ให้ผลดีกว่า เนื่องจากมีการพบค่าออปติ멈ได้ใกล้เคียงกับคำตอบที่ถูกต้องมากกว่า
- จากการศึกษางานวิจัยนี้ มีปัจจัยด้านจำนวนโรงงานที่ทำการผลิตเข้ามาเกี่ยวข้องทำให้มีความซับซ้อนมากขึ้น ดังนั้นทุกโอเปอร์เรชันของงานหนึ่งต้องเลือกผลิตที่โรงงานเดียวกัน ซึ่งโมเดลที่นำมาใช้แก้ปัญหาเป็นตามกระบวนการของจินตออลกอริทึม แต่มีการเพิ่มวิธีการมิวเทชันให้มีความหลากหลายสามารถปรับเปลี่ยนในรายละเอียดของโอเปอร์เรชัน ซึ่งสามารถเปลี่ยนเครื่องจักร และ โรงงาน ได้และลำดับการจัดวางของโอเปอร์เรชันได้

2.3.7 An effective Hybridized Genetic Algorithm architecture for the Flexible Job-Shop Scheduling Problem

Nasr AI-Hinai and T.Y. ElMekkawy [23] งานวิจัยนี้นำเสนอการผสมผสานของจินตออลกอริทึมสำหรับแก้ปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น โดยนำเสนอในส่วนของโครงสร้างประชากรเริ่มต้น และวิธีการหาคำตอบแบบโลคอล โดยมีฟังก์ชันวัตถุประสงค์ คือ เวลาในการผลิตรวมน้อยที่สุด

โมเดลการผสมผสานของจินตออลกอริทึมในงานวิจัยมีขั้นตอนการทำงาน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) การเข้ารหัสชุดคำตอบโครโมโซม ในแต่ละโครโมโซมประกอบด้วยจำนวนโอเปอร์เรชันทั้งหมด ซึ่งแต่ละโอเปอร์เรชันหรือยีนประกอบด้วย (k, i, j) ซึ่ง k คือหมายเลขเครื่องจักร i คือหมายเลขงาน และ j คือลำดับของโอเปอร์เรชันของงาน i

2) ถอดรหัสชุดคำตอบโครโมโซม ใช้แผนภาพแกนต์หาค่าคำตอบของแต่ละโครโมโซม

3) กำหนดประชากรเริ่มต้น กำหนดให้แบ่งเป็น 2 ส่วน คือ ส่วนโอเปอร์เรชัน ได้จากการสุ่มหมายเลขงานตามจำนวนโอเปอร์เรชันของแต่ละงาน และส่วนเครื่องจักร โดยเลือกเครื่องจักรที่ทำให้งานเสร็จเร็วที่สุด ซึ่งอาจไม่ใช่เครื่องจักรที่มีเวลาทำงานสั้นที่สุดของโอเปอร์เรชันนั้น

4) การคัดเลือกโครโมโซม คัดเลือกโครโมโซมเพื่อเป็นโครโมโซมต้นแบบ โดยแบ่งเป็นส่วนผู้ให้และผู้รับซึ่งส่วนผู้ให้จะเลือกโดยวิธีสุ่มจากวงล้อที่แบ่งสัดส่วนของแต่ละโครโมโซม จากความน่าจะเป็นตามสมการที่ 2.24 โดยโครโมโซมจะไม่เรียงลำดับตามค่าฟิตเนสเพื่อเป็นการป้องกันไม่ให้ผู้เข้าหาคำตอบแบบโลคอล ส่วนผู้รับจะเลือกโครโมโซมด้วยวิธีการแข่งขัน (Tournament Method) คือการสุ่มโครโมโซมเพื่อเปรียบเทียบและเลือกโครโมโซมที่ดีที่สุด

$$P_s = \frac{F_{ind}}{F_{tot}} ; \quad (2.24)$$

เมื่อ F_{ind} = ค่าฟิตเนสของโครโมโซมตัวที่ ind

F_{tot} = ค่าฟิตเนสของโครโมโซมทั้งหมด

F_s = ความน่าจะเป็นของโครโมโซมตัวที่ ind

N = จำนวนประชากรในแต่ละรอบ

$ind = 1, \dots, N$

5) กระบวนการทางพันธุกรรม (Genetic)

5.1) วิธีการครอสโอเวอร์ ใช้วิธีการ POX (Precedence Operation Crossover) เพื่อสร้างโครโมโซมทายาทจากโครโมโซมต้นแบบ มีขั้นตอนดังนี้

a) สุ่มยีนหรือโอเปอร์เรชันในโครโมโซมผู้ให้ แล้วคัดลอกทุกโอเปอร์เรชันที่เป็นงานเดียวกันไปวางตำแหน่งเดิมในโครโมโซมทายาท

b) คัดลอกทุกโอเปอร์เรชันที่เหลือของทุกงานที่ไม่ถูกเลือกในข้อ a) จากโครโมโซมผู้รับไปยังโครโมโซมทายาท โดยวางในตำแหน่งที่ว่างตามลำดับจากซ้ายไปขวาจนครบทุกโอเปอร์เรชัน

5.2) วิธีมิวเตชันเฉพาะโครโมโซมทายาท ซึ่งมี 2 ขั้นตอนดังนี้

- a) ทำการสุ่มโอเปอร์เรชันในโครโมโซมจำนวน n_{rand} ครั้งเพื่อเปลี่ยนเครื่องจักร (Machine Based Mutation: MBM) ให้กับโครโมโซมทายาท
 - b) จัดลำดับโอเปอร์เรชัน (Position Based Mutation: PBM) เพื่อจัดลำดับตำแหน่งของโอเปอร์เรชันใหม่หลังจากวิธี MBM แล้ว โดยสุ่มเลือกโอเปอร์เรชันที่ต้องการแก้ไข 1 ตำแหน่งและสุ่มโอเปอร์เรชันใหม่อีก 1 ตำแหน่งเพื่อนำไปแทรก
- 6) การค้นหาแบบโลคอล เพื่อค้นหาคำตอบที่เหมาะสมที่สุด หรือใกล้เคียงงานวิจัยนี้ได้ เสนอการสร้างคำตอบจากพื้นที่รอบข้างของคำตอบปัจจุบัน มีขั้นตอนดังนี้
- 6.1) กำหนดให้ชุดคำตอบปัจจุบัน(S) เป็นค่าคำตอบเริ่มต้น และเป็นคำตอบที่ดีที่สุด (Sbest)
 - 6.2) ค้นหาพัฒนาจากแผนภาพแกนต์ และสร้างบล็อกของชุดคำตอบที่เป็นเส้นทางวิกฤตโดยพิจารณาจากโอเปอร์เรชันที่อยู่บนเครื่องจักรเดียวกันจากนั้นทำการสลับตำแหน่งของโอเปอร์เรชันภายในบล็อกภายใต้เงื่อนไขที่เป็นไปได้ (S')
 - 6.3) เปรียบเทียบค่าฟิตเนส ถ้า (S') ดีกว่า Sbest ให้ปรับค่า Sbest ← (S')
 - 6.4) ให้ทำขั้นตอนที่ 2 จนกระทั่งครบตามเงื่อนไข
 - 6.5) กระบวนการดังกล่าวใช้เฉพาะในรอบที่ d ในรอบการทำงานของอัลกอริทึมเท่านั้น
- 7) กลยุทธ์การคัดสรร ในแต่ละรอบการทำงาน โครโมโซมมีค่าฟิตเนสที่ดีกว่าจะถูกบันทึกในกลุ่มของโครโมโซมหิวกระหิและจะแทนที่โครโมโซมตัวที่แย่กว่า ซึ่งขั้นตอนนี้จะปรับปรุงเฉพาะรอบที่ k ในรอบการทำงานของอัลกอริทึมเท่านั้น

โมเดล hGA ที่นำเสนอประกอบด้วย 3 ส่วน คือ การกำหนดประชากรเริ่มต้น วิธีการค้นหาแบบโลคอล และจินตนาการอัลกอริทึม โดยวัดประสิทธิภาพของโมเดลกับข้อมูลปัญหาจำนวน 5 ชุด และกำหนดค่าพารามิเตอร์ ตามตารางที่ 2.3 ซึ่งเปรียบเทียบกับอัลกอริทึมที่เสนอกับโมเดลที่ได้เสนอโดย 1) Z.K.K ของ Zribi et al.[24] 2) P.M.C ของ Pezzella et al.[9] และ 3) G.S.G ของ Gao et al.[25] พบว่าโมเดล hGA มีประสิทธิภาพดีกว่าโดยมีค่าคลาดเคลื่อนสัมพัทธ์เฉลี่ยต่ำสุดเมื่อเปรียบเทียบกับโมเดลอื่น

ข้อที่นำสังเกตของโมเดลนี้การค้นหาแบบโลคอลที่งานวิจัยนี้ได้นำเสนอเป็นการค้นหาพื้นที่รอบข้างของคำตอบ ซึ่งหากคำตอบที่ค้นหาเป็นจุดที่ใกล้กับจุดออปติหมัการค้นหาแบบโลคอล จะทำให้พบกับจุดออปติหมัในที่สุด แต่หากคำตอบที่ทำการค้นหาพื้นที่รอบข้างนั้นห่างไกลจากจุดออปติหมัมากจะทำให้เสียเวลานานในการค้นหาคำตอบ

ตารางที่ 2.3 ชุดข้อมูลทดสอบและค่าพารามิเตอร์ในการทดลอง

ชุดข้อมูลทดสอบ	ตัวแทนข้อมูล		ค่า d	จำนวนประชากร
KMData	All		10	200
BRData	MK1/6/7/10		30	1200
	All remaining		10	200
BCData	All		30	1500
DPData	All		30	1200
HUData	EData	mt6/10	20	200
		mt20	30	800
		la01-la13	20	400
		La14-la25	20	500
		la26-la40	30	1200
RData		mt6/10/20	20	300
		la01-la25	20	400
		la26-la40	30	1000
VData		mt6/10/20	10	200
		La01-la40	10	300

2.3.8 An effective Genetic Algorithm for the Flexible Job-Shop Scheduling Problem

Guohui Zhang, Liang Gao และ Yang Shi [26] ได้นำเสนอโมเดล eGA เพื่อศึกษาวิธีการเพิ่มประสิทธิภาพของจีเนติกอัลกอริทึมสำหรับการแก้ปัญหาการจัดตารางการผลิตแบบตามสั่ง ยืดหยุ่น โดยได้นำเสนอการสร้างประชากรเริ่มต้น การครอสโอเวอร์ และการมิวเตชัน โดยมีฟังก์ชันเป้าหมายคือเวลาในการผลิตรวมน้อยที่สุด

โมเดล eGA ที่ได้นำเสนอในงานวิจัยมีขั้นตอนการทำงาน ดังนี้

1) การออกแบบโครโมโซม

ลักษณะของโครโมโซมแบ่งเป็น 2 ส่วน คือ ส่วนการเลือกเครื่องจักร (Machine Selection part: MS) และส่วนลำดับ โอเปอเรชัน (Operation Sequence part: OS) ตัวอย่างเช่น กำหนดให้มี 2 งาน งานที่ 1 มี 2 โอเปอเรชัน และงานที่ 2 มี 3 โอเปอเรชัน ดังนั้นจำนวนโอเปอเรชันจะมีความยาวเท่ากับ 5 x 2 โอเปอเรชัน ดังรูปที่ 2.13

ส่วนการเลือกเครื่องจักร					ส่วนลำดับโอเปอร์เรชัน				
4	1	2	2	4	2	2	1	1	2
O_{11}	O_{12}	O_{21}	O_{22}	O_{23}	O_{21}	O_{22}	O_{11}	O_{12}	O_{23}

รูปที่ 2.13 ลักษณะของโครโมโซม

1.1) ส่วนการเลือกเครื่องจักร กำหนดให้แต่ละโอเปอร์เรชัน คือหมายเลขเครื่องจักรที่เลือกให้กับโอเปอร์เรชันของงาน โดยกำหนดให้เรียงลำดับตามงานและตามโอเปอร์เรชัน เช่น ตำแหน่งแรก 4 หมายถึงเครื่องจักร ทำงานให้กับ O_{11} หมายถึงงานที่ 1 โอเปอร์เรชัน 1

1.2) ส่วนลำดับโอเปอร์เรชัน จะแสดงหมายเลขงานตามโอเปอร์เรชัน ตำแหน่งแรกคือ 2 หมายถึง งานที่ 2 โอเปอร์เรชัน 1 (O_{21})

2) กำหนดประชากรเริ่มต้น

2.1) ส่วนการเลือกเครื่องจักร แบ่งเป็น 3 วิธี คือ การเลือกแบบโกลบอล (Global Selection: GS) การเลือกแบบโลคอล (Local Selection: LS) และวิธีการสุ่ม (Random Selection: RS) มีรายละเอียดดังนี้

2.1.1) GS: เป็นการเลือกเครื่องจักร โดยเลือกเครื่องจักรที่ใช้เวลาการทำงานที่น้อยที่สุด ซึ่งเวลาทำงานรวมของเครื่องจักรจะรวมทุกงานเข้าด้วยกัน

2.1.2) LS: เป็นการเลือกเครื่องจักร โดยเลือกเครื่องจักรที่ใช้เวลาการทำงานที่น้อยที่สุด ซึ่งเวลาทำงานรวมของเครื่องจักรจะแยกตามงาน

2.1.3) RS: เป็นการเลือกโดยการสุ่มเครื่องจักรจากเซตของเครื่องจักรที่สามารถทำในโอเปอร์เรชันนั้นๆ

2.2) ส่วนลำดับโอเปอร์เรชัน ใช้วิธีการสุ่มหมายเลขงานตามจำนวนโอเปอร์เรชัน

3) การคัดเลือกประชากร

ใช้วิธีการแข่งขัน โดยสุ่มสมาชิกของประชากรจำนวน 3 โครโมโซม จากนั้นเลือกโครโมโซมที่มีค่าฟิตเนสที่ดีที่สุดเพื่อเป็นโครโมโซมต้นแบบจนครบจำนวนที่กำหนด

4) การครอสโอเวอร์ แบ่งเป็น 2 วิธี คือ

4.1) การครอสโอเวอร์ส่วนการเลือกเครื่องจักร ใช้ 2 วิธี คือ การสลับสายพันธุแบบ 2 ช่วง (Two-Point Crossover) จากนั้นใช้วิธีสลับสายพันธุแบบยูนิฟอร์ม (Uniform Crossover)

4.2) การครอสโอเวอร์ส่วนลำดับโอเปอร์เรชัน ใช้วิธีการ POX โดยสุ่มเลือกงานจำนวนครึ่งหนึ่งของจำนวนงานทั้งหมดแล้วทำการคัดลอกไปยังโครโมโซมทายาท

5) การมิวเตชันแบ่งเป็น 2 วิธี คือ

5.1) การมิวเตชันส่วนการเลือกเครื่องจักร โอเปอร์เรชันที่มีความน่าจะเป็นในการมิวเตชันจะเลือกเครื่องจักรที่ใช้เวลาทำงานน้อยสุดที่เป็นไปได้แทนเครื่องจักรเดิม

5.2) การมิวเตชันส่วนลำดับโอเปอร์เรชัน โอเปอร์เรชันที่มีความน่าจะเป็นในการมิวเตชันจะสุ่มเลือก 2 ตำแหน่ง แล้วสลับตำแหน่งเพื่อจัดลำดับโอเปอร์เรชันใหม่

6) การคัดเลือกโครโมโซมเพื่อเป็นประชากรรุ่นถัดไป โครโมโซมทายาทที่มีค่าฟิตเนสดีกว่าจะถูกนำไปแทนที่โครโมโซมที่มีค่าฟิตเนสที่แย่กว่า

การประเมินประสิทธิภาพของโมเดล eGA โดยค่าพารามิเตอร์ได้แก่ จำนวนรอบการทำงาน อัตราการสร้างประชากรเริ่มต้นด้วยวิธี GS, LS, RS, ความน่าจะเป็นของการครอสโอเวอร์ และความน่าจะเป็นในการมิวเตชัน ผลการทดลองพบว่ากำหนดประชากรเริ่มต้นด้วยวิธี GS LS และ RS พบว่าวิธีการ GS และ LS ได้ผลดีกว่าวิธีการสุ่ม และนำโมเดลที่นำเสนอทดสอบกับข้อมูลจำนวน 3 ชุด ได้แก่ 1) Brandimarte's (BRdata) [20] จำนวน 10 ปัญหา 2) Barnes และ Chambers (BCdata) [27] จำนวน 21 ปัญหา และ 3) Dauzere-Peres (DPdata) [28] จำนวน 18 ปัญหา โดยเปรียบเทียบกับอัลกอริทึมอื่น พบว่าโมเดลที่นำเสนอมีประสิทธิภาพดีกว่าโดยคำตอบที่ได้ใกล้เคียงกับจุดออปติ멈มากกว่าและใช้เวลาในการทำงานน้อยกว่า

ข้อที่น่าสังเกตของโมเดลนี้มีการกำหนดประชากรเริ่มต้นที่หลากหลายด้วยสัดส่วน GS:60% LS:30% และ RS:10% ของจำนวนประชกรนั้น ทำให้มีโอกาสพบคำตอบที่ดีในตอนเริ่มต้น ส่วนกระบวนการของทำงานของอัลกอริทึมยังคงใช้จินเนติกอัลกอริทึมที่มีจุดเด่นในการปรับค่าคำตอบด้วยวิธีการจินเนติกด้วยการครอสโอเวอร์และมิวเตชัน

2.3.9 An efficient knowledge-based Algorithm for the Flexible Job-Shop Scheduling Problem

Hamid Karimi, Seyed Habib A. Rahmti และ M. Zandieh [29] ผู้วิจัยได้นำเสนอโมเดล KBVNS (Knowledge-Base Variable Neighborhood Search) เพื่อปรับปรุงจุดด้อยของอัลกอริทึม VNS (Variable neighborhood search) ที่ใช้วิธีการสุ่มในการค้นหาคำตอบที่เหมาะสมที่สุด โดยโมเดลที่เสนอมีการพัฒนาวิธีการเรียนรู้ให้อัลกอริทึม VNS ให้สามารถค้นหาคำตอบได้ดีขึ้นกว่าอัลกอริทึม VNS แบบดั้งเดิม ซึ่งมีฟังก์ชันเป้าหมายคือ เวลาในการผลิตรวมน้อยที่สุด

อัลกอริทึม VNS แบบดั้งเดิมถูกสร้างขึ้นโดย Pierre Hansen และ Nenad Mladenovic ใน ปี 2001 [30] โดยแบ่งการทำงานเป็น 2 ส่วน คือ ส่วนการเขย่า (shaking) เป็นการสร้างความหลากหลายเพื่อหลีกเลี่ยงคำตอบแบบโลคอล และส่วนการค้นหาแบบโลคอลเป็นการปรับปรุงชุดคำตอบเพื่อให้เข้าใกล้คำตอบที่เป็นจุดออปติ멈

หลักการทํางานของ อัลกอริทึมสามารถสรุปได้ ดังนี้
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) สร้างชุดคำตอบจากพื้นที่รอบข้างจำนวน N_k ชุด เมื่อ $k = 1, \dots, k_{\max}$ เพื่อใช้ค้นหาคำตอบจากพื้นที่รอบข้างของชุดคำตอบเริ่มต้น (x) และกำหนดจำนวนรอบในการทำงาน

2) ให้ทำขั้นตอนดังกล่าวตามเงื่อนไข

2.1) กำหนดให้ $k \leftarrow 1$;

2.2) ทำจนกระทั่ง $k = k_{\max}$ ให้ทำขั้นตอนต่อไปนี

2.2.1) การเขย่า ด้วยการสร้างชุดคำตอบใหม่ (x') จากการสุ่มเลือกวิธีการสร้างชุดคำตอบจากพื้นที่รอบข้างชุดที่ k สำหรับ (x) เมื่อ ($x' \in N_k(x)$)

2.2.2) การค้นหาแบบโลคอล ใช้วิธีการค้นหาแบบโลคอลเพื่อปรับค่าชุดคำตอบ (x') เพื่อหาคำตอบที่เหมาะสมแบบโลคอล (x'')

2.2.3) ถ้า (x'') ดีกว่า (x) ให้ปรับค่าคำตอบเริ่มต้น ($x \leftarrow x''$) และกำหนดให้ชุดคำตอบจากพื้นที่รอบข้างชุดที่ 1 ($k \leftarrow 1$) แต่ถ้า (x'') แย่กว่า (x) จะกำหนดให้ชุดคำตอบจากพื้นที่รอบข้างเป็นชุดถัดไป ($k \leftarrow k+1$)

ส่วนที่ปรับปรุงและนำเสนอใน โมเดล KBVNS สามารถสรุปได้ ดังนี้

1) การกำหนดตัวแทนชุดคำตอบ (Solution Representation)

ประกอบด้วย 2 ส่วน คือชุดลำดับโอเปอร์เรชัน (Operation Sequence Vector) ซึ่งประกอบด้วยหมายเลขงานวางลำดับจากซ้ายไปขวา และชุดการเลือกเครื่องจักร (Machine Selection Vector) ประกอบด้วยเครื่องจักรที่ถูกเลือกให้กับแต่ละโอเปอร์เรชัน กำหนดให้เรียงลำดับตามตำแหน่งงานและโอเปอร์เรชันจากงาน 1 โอเปอร์เรชัน 1 จนถึง งาน 3 โอเปอร์เรชัน 3 ตามลำดับ ($O_{11}, O_{12}, \dots, O_{33}$) ดังรูปที่ 2.14

Operation sequence vector							Machine selection vector						
$O_{3,1}$	$O_{1,1}$	$O_{2,1}$	$O_{3,2}$	$O_{2,2}$	$O_{3,3}$	$O_{1,2}$	$O_{1,1}$	$O_{1,2}$	$O_{2,1}$	$O_{2,2}$	$O_{3,1}$	$O_{3,2}$	$O_{3,3}$
3	1	2	3	2	3	1	3	4	1	3	2	3	1

รูปที่ 2.14 โครงสร้างตัวแทนชุดคำตอบ [29]

2) การถอดรหัสชุดคำตอบ (Decoding)

ถอดรหัสชุดคำตอบด้วยวิธี Active Schedule กล่าวคือ จะพิจารณาทีละโอเปอร์เรชันของชุดลำดับโอเปอร์เรชันจากซ้ายไปขวาและเลือกเครื่องจักรจากชุดการเลือกเครื่องจักรบนแผนภูมิแกนต์

ซึ่งโอเปอร์เรชันสามารถเลื่อนข้ามมาอยู่หน้าก่อนโอเปอร์เรชันอื่นได้ โดยไม่ทำให้โอเปอร์เรชันอื่นที่ถูกข้ามต้องเริ่มต้นช้ากว่าเดิม (Global Left-Shift) เนื่องจากมีเวลาบนเครื่องจักรว่างอยู่

3) มอดูลความรู้ (Knowledge Module) จะจดจำลักษณะของคำตอบที่ดีและไม่ดีของชุดคำตอบก่อนหน้า เพื่อปรับปรุงคุณภาพของชุดคำตอบต่อไป

3.1) OM (Operation Memory) เป็นการจับเก็บลักษณะของคำตอบที่ดีจากรอบก่อนหน้านั้นด้วยการสร้างชุดการเลือกเครื่องจักรของคำตอบที่ดีที่สุด ที่ประกอบด้วยชุดของเครื่องจักรที่สามารถทำงานในโอเปอร์เรชันนั้น และเครื่องจักรของโอเปอร์เรชันนั้นเมื่อถูกเลือกจะมีค่าเท่ากับ 1 และไม่ถูกเลือกจะเท่ากับ 0 ดังรูป 2.15

O _{1,1}				O _{1,2}				O _{2,1}				O _{2,2}				O _{3,1}				O _{3,2}				O _{3,3}					
M ₁	M ₂	M ₃	M ₄	M ₁	M ₂	M ₃	M ₄	M ₁	M ₄	M ₁	M ₂	M ₃	M ₄	M ₁	M ₂	M ₃	M ₄	M ₁	M ₂	M ₃	M ₄	M ₁	M ₂	M ₃	M ₄	M ₁	M ₂	M ₃	M ₄
1	0	1	0	0	0	1	1	0	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1

รูปที่ 2.15 Operation Memory [29]

3.2) การปรับค่า OM เมื่อชุดคำตอบที่ได้ดีกว่าหรือเท่ากับชุดคำตอบเดิม ตำแหน่งของเครื่องจักรในชุดโอเปอร์เรชันที่ถูกเลือกจะเปลี่ยนจาก 0 เป็น 1 แต่หากมีการเลือกซ้ำจะไม่เปลี่ยนแปลง

4) การกำหนดโครงสร้างพื้นที่รอบข้าง (Neighborhood Structure)

แบ่งการทำงานเป็น 2 ส่วน คือการเขย่าเพื่อหลีกเลี่ยงคำตอบแบบโลคอลนำไปสู่คำตอบแบบโกลบอล ประกอบด้วย SN₁ และ SN₂ และการค้นหาแบบโลคอลเพื่อปรับค่าชุดคำตอบ ประกอบด้วย LN₁, LN₂, LN₃, LN₄, LN₅ มีรายละเอียด ดังนี้

LN₁: สุ่มเลือก 1 โอเปอร์เรชัน ในชุดของการเลือกเครื่องจักร จากนั้นทำการเปลี่ยนเครื่องจักรที่โอเปอร์เรชันนั้นได้เลือกไว้โดยเลือกจากชุดเครื่องจักรอื่นที่สามารถทำงานในโอเปอร์เรชันนั้นได้แทนเครื่องจักรตัวเดิม

LN₂: สุ่มเลือก 2 ตำแหน่งในชุดของลำดับโอเปอร์เรชัน (r1, r2) และทำการสลับตำแหน่งระหว่าง r1 และ r2

LN₃: สุ่ม 2 ตำแหน่งในชุดของลำดับโอเปอร์เรชัน (r1, r2) และทำการเรียงลำดับย้อนกลับ (invert) ในช่วง r1 และ r2 เช่น r1 =3 และ r2=5 ชุดคำตอบ คือ 1311322333 เป็น 1331122333 เป็นต้น

LN₄: สุ่มเลือก 1 โอเปอร์เรชัน ในชุดของลำดับโอเปอร์เรชัน จากนั้นพิจารณาเครื่องจักรที่สามารถทำงานในโอเปอร์เรชันที่สุ่ม โดยพิจารณาจาก OM เฉพาะเครื่องจักรที่มีค่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่ากับ 1 จากนั้นทำการสุ่มเลือกเครื่องจักรใหม่แทนตัวเดิม หากเครื่องจักรใน OM มีเพียงตัวเลือกเดียวให้สุ่มเลือกจากที่เหลือภายใต้ความเป็นไปได้ของโอเปอร์เรชันนั้น

LN_3 : ทำ LN_4 และ LN_2 รวมกัน

SN_1 : ใช้วิธี LN_1 และ LN_2 รวมกัน

SN_2 : เหมือน SN_1 และทำซ้ำ 2 ครั้ง

ขั้นตอนการทำงานของโมเดล KBVNS สรุปได้ ดังนี้

1) กำหนดชุดคำตอบเริ่มต้น (y) ด้วยวิธีการสุ่ม และกำหนดให้ชุดคำตอบเริ่มต้นเป็นชุดคำตอบที่ดีที่สุด

2) ใช้กระบวนการเขย่า (SN_k) กำหนดให้ $k = 1$ และได้ชุดคำตอบใหม่ (y')

3) ทำการค้นหาคำตอบแบบโลคอล (y'')

3.1) ทำงานตามเงื่อนไขที่กำหนด N รอบ

- สุ่มค่า r ในช่วง $[0,1]$

- ถ้า $r < P_1$ ใช้การค้นหาคำตอบแบบโลคอลชุดที่ LN_3

มิฉะนั้น ถ้า $r < P_2$ ใช้การค้นหาคำตอบแบบโลคอลชุดที่ LN_5

มิฉะนั้น ถ้า $r < P_3$ ใช้การค้นหาคำตอบแบบโลคอลชุดที่ LN_4

มิฉะนั้น ถ้า $r < P_4$ ใช้การค้นหาคำตอบแบบโลคอลชุดที่ LN_1

มิฉะนั้น ใช้การค้นหาคำตอบแบบโลคอลชุดที่ LN_2

- ถ้า (y'') ดีกว่า (y') ให้ปรับค่า ($y' \leftarrow y''$)

4) ถ้า (y') ดีกว่า ชุดคำตอบเริ่มต้น (y) ให้ปรับค่า ($y \leftarrow y'$), $k = 1$ และปรับค่า OM มิฉะนั้น $k = k+1$ และกลับไปทำขั้นตอนที่ 2

5) ทำงานจนครบตามเงื่อนไข

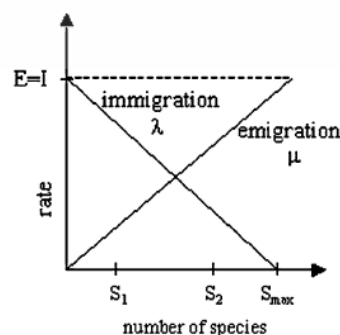
งานวิจัยนี้ได้ประเมินประสิทธิภาพของโมเดลที่นำเสนอ สำหรับชุดข้อมูลมาตรฐานของ Kacem et al.[19] Brandimarte [20] พบว่าโมเดลนี้สามารถพบค่าคำตอบที่ใกล้เคียงจุดออพติ멈มากกว่าอัลกอริทึมอื่น และใช้เวลาน้อยกว่า และยังได้ทำการสร้างชุดทดสอบโดยการสุ่มเพื่อเปรียบเทียบระหว่างโมเดล KBVNS ที่นำเสนอ กับ VNS แบบดั้งเดิม โดยรันจำนวน 5 ครั้ง พบว่าโมเดล KBVNS มีประสิทธิภาพดีกว่า VNS แบบดั้งเดิม

จากการศึกษางานวิจัยนี้สามารถวิเคราะห์จุดเด่นที่สำคัญดังนี้ อัลกอริทึม VNS แบบดั้งเดิมใช้วิธีการสุ่มสำหรับการค้นหาคำตอบซึ่งทำให้พื้นที่ในการค้นหาคำตอบมีขนาดใหญ่ ดังนั้นงานวิจัยจึงได้เสนอวิธีการเขย่า และวิธีการค้นหาแบบ โลคอลที่มีวิธีการที่หลากหลายเป็นการลดพื้นที่ในการค้นหา

2.3.10 A new Biogeography-Based Optimization (BBO) Algorithm for the Flexible Job-Shop Scheduling Problem

Seyed Habib A. Rahmati และ M. Zandieh [31] ศึกษาวิธีการแก้ปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น โดยใช้วิธีการหาค่าเหมาะสมแบบการกระจายของสิ่งมีชีวิตตามภูมิศาสตร์ (Biogeography-Based Optimization: BBO) พัฒนาโดย Simon ในปี 2008 [32] ซึ่งมีแนวคิดมาจากการเลียนแบบการอพยพย้ายถิ่นของสิ่งมีชีวิตหรือสปีชีส์ (Species) ตามภูมิศาสตร์ที่อยู่อาศัย ที่แสดงถึงการอพยพของสิ่งมีชีวิตจากที่อยู่อาศัยหนึ่งไปยังที่อยู่อาศัยอื่นๆ ทำให้เกิดสิ่งมีชีวิตใหม่และการสูญพันธุ์ของสิ่งมีชีวิต ความอุดมสมบูรณ์ของที่อยู่อาศัยสามารถแสดงด้วยดัชนีความเหมาะสม (Habitat Suitability Index: HSI) ดัชนีความเหมาะสมหมายถึงปัจจัยต่างๆ เช่น ปริมาณน้ำฝน ความหลากหลายของพืชพรรณ พื้นที่ และอุณหภูมิ ซึ่งจะเรียกว่าตัวแปรความเหมาะสม (Suitability Index Variables: SIVs) ที่อยู่อาศัยที่มีค่า HSI สูงจะมีสิ่งมีชีวิตอาศัยจำนวนมาก ทำให้มีอัตราการอพยพเข้าน้อย (Low Immigration rate) และมีอัตราการอพยพออกมาก (High Emigration rate) เนื่องจากมีความแออัดของที่อยู่อาศัย ในทางตรงกันข้ามที่อยู่อาศัยที่มีค่า HSI ต่ำจะมีสิ่งมีชีวิตอาศัยจำนวนน้อย มีอัตราการอพยพเข้ามาก (High Immigration rate) และมีอัตราการอพยพออกน้อย (Low Emigration rate) เนื่องจากสิ่งมีชีวิตอยู่น้อยและมีพื้นที่สำหรับสิ่งมีชีวิตที่จะอพยพเข้ามาได้มาก สามารถแสดงความสัมพันธ์ของการอพยพเข้าและอพยพออก ดังรูปที่ 2.16 เมื่อ E และ I คืออัตราการอพยพเข้าและอพยพออกสูงสุดซึ่งปกติมีค่าเท่ากับ 1 S_{max} คือจำนวนสูงสุดของสิ่งมีชีวิตในที่อยู่อาศัย และ S_0 คือจุดสมดุลระหว่างอัตราการอพยพเข้าและอพยพออก

ฟังก์ชันวัตถุประสงค์ในงานวิจัยนี้มี 3 ฟังก์ชัน คือ 1) เวลาในการผลิตรวมน้อยที่สุด 2) เวลาในการผลิตเครื่องจักรที่ทำงานมากที่สุด (Critical Machine Workload) และ 3) เวลาในการทำงานของทุกโอเปอเรชันของทุกงานบนเครื่องจักร (Total Work Load of Machine) อัลกอริทึมการกระจายของสิ่งมีชีวิตตามภูมิศาสตร์ มีลักษณะการทำงานใกล้เคียงกันกับจินเนติกอัลกอริทึม และพาร์ทิเคิลสวอร์มอัลกอริทึม สามารถเทียบนิยามของอัลกอริทึมการกระจายของสิ่งมีชีวิตตามภูมิศาสตร์กับจินเนติกอัลกอริทึมดังตารางที่ 2.4



รูปที่ 2.16 การเปลี่ยนแปลงของอัตราการอพยพเข้า และอพยพออก [31]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นใบเซอร์เวชันด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 นิยามตามหลักการของอัลกอริทึมการกระจายของสิ่งมีชีวิตตามภูมิศาสตร์ และจินตนาการ อัลกอริทึม

	BBO	GA
1	ประชากร	ประชากร
2	ที่อยู่อาศัย (สมาชิกของประชากร)	โครโมโซม (สมาชิกของประชากร)
3	ตัวแปรดัชนีความเหมาะสม (SIV)	ยีน
4	ที่อยู่อาศัย ประกอบด้วยตัวแปรดัชนีความเหมาะสม	โครโมโซม ประกอบด้วยยีน
5	มิวเทชัน	มิวเทชัน
6	การอพยพ (เข้า – ออก)	ครอสโอเวอร์
7	คำตอบที่ดีมีดัชนีความเหมาะสมของการอยู่อาศัยสูง	คำตอบที่ดีมีค่าฟิตเนสสูง
8	ที่อยู่อาศัยที่ดี สิ่งมีชีวิตมีความหลากหลาย	โครโมโซมที่ดีมีค่าฟิตเนสฟังก์ชันสูง
9	ประชากรเริ่มต้นไม่ถูกตัดออกในระหว่างกระบวนการ แต่จะมีการปรับปรุง	ประชากรเริ่มต้นสามารถถูกตัดออกในระหว่างกระบวนการของจินตนาการได้

ขั้นตอนอัลกอริทึมการกระจายของสิ่งมีชีวิตตามภูมิศาสตร์สำหรับแก้ปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น สามารถสรุปได้ดังนี้

1) กำหนดค่าพารามิเตอร์เริ่มต้น

ประกอบด้วย อัตราการอพยพออกสูงสุด (E), อัตราการอพยพเข้าสูงสุด(I), ความน่าจะเป็นการมิวเทชันสูงสุด (m_{max}), จำนวนสิ่งมีชีวิต (Pop_size) และจำนวนรอบการทำงาน

2) การสร้างชุดคำตอบเริ่มต้น

ใช้วิธีการของ Wang et al.[33] แบ่งชุดคำตอบหรือสิ่งมีชีวิตเป็น 2 ส่วน คือ ส่วนของลำดับโอเปอร์เรชัน แสดงหมายเลขงานตามจำนวนโอเปอร์เรชันของงานนั้นๆ และส่วนของเครื่องจักร แสดงหมายเลขเครื่องจักรตามตำแหน่งของงานและโอเปอร์เรชัน โดยกำหนดค่าสุ่มในแต่ละโอเปอร์เรชัน ($rand \in [0,1]$) ถ้าน้อยกว่า 0.8 เลือกเครื่องจักรที่ใช้เวลาทำงานน้อยที่สุด มิฉะนั้นเลือกเครื่องจักรที่ใช้เวลาทำงานมากที่สุด ดังรูปที่ 2.17

Operation Sequence	O ₃₁	O ₁₁	O ₂₁	O ₃₂	O ₁₂	O ₂₂	O ₃₃	O ₁₃
	3	1	3	3	1	2	3	1
Machine Assignment	O ₁₁	O ₁₂	O ₁₃	O ₂₁	O ₂₂	O ₃₁	O ₃₂	O ₃₃
	1	2	2	1	2	3	2	1

รูปที่ 2.17 ชุดคำตอบสำหรับปัญหา 3 งาน 4 เครื่องจักรและ 8 โอเปอร์เรชัน [31]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การแทนและการถอดรหัสชุดคำตอบ

เซตของชุดคำตอบประกอบด้วยตัวแปรดัชนีความเหมาะสมหรืออื่นสามารถแสดงตามรูปที่ 2.18 และถอดรหัสชุดคำตอบด้วยแผนภาพแกนต์

$$[(O_{31}, M_3), (O_{11}, M_1), (O_{21}, M_1), (O_{32}, M_2), (O_{12}, M_2), (O_{22}, M_2), (O_{33}, M_4), (O_{13}, M_2)]$$

รูปที่ 2.18 เซตชุดตัวแปรดัชนีความเหมาะสม (อื่น) ของชุดคำตอบ

4) ประเมินค่าดัชนีความเหมาะสมหรือค่าฟิตเนสของชุดคำตอบ และจัดอันดับตามดัชนีความเหมาะสม

5) คำนวณค่าพารามิเตอร์ ได้แก่ อัตราการอพยพเข้า (λ_i) อัตราการอพยพออก (μ_i) ความน่าจะเป็นของสิ่งมีชีวิตตัวที่ i (P_i) เพื่อสร้างเป็นวงล้อรูเล็ตสำหรับครอสโอเวอร์ และความน่าจะเป็นของการมิวเตชันของสิ่งมีชีวิตตัวที่ i (m_i) สำหรับการมิวเตชัน สามารถแสดงด้วยสมการ 2.25 - 2.29

$$\lambda_i = I \left(1 - \frac{k_i}{n} \right) \quad (2.25)$$

$$\mu_j = E \left(\frac{k_i}{n} \right) \quad (2.26)$$

$$P_i = \frac{v_i}{\sum_{i=1}^n v_i} \quad (2.27)$$

$$v_i = \begin{cases} \frac{n!}{(n+1-i)!(i-1)!} & i = 1, 2, 3, \dots, i' \\ v_n + 2 - i & i = i' + 1, \dots, n + 1 \end{cases} \quad (2.28)$$

$$m_i = m_{\max} \left(1 - \frac{P_i}{P_{\max}} \right) \quad (2.29)$$

เมื่อ λ_i คือ อัตราการอพยพเข้าของสิ่งมีชีวิตตัวที่ i

μ_i คือ อัตราการอพยพออกของสิ่งมีชีวิตตัวที่ i

k_i คือ ตำแหน่งของสิ่งมีชีวิตตัวที่ i

n คือ จำนวนของสิ่งมีชีวิตหรือชุดคำตอบเริ่มต้น

P_i คือ ความน่าจะเป็นของสิ่งมีชีวิตตัวที่ i

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

m_i คือ ความน่าจะเป็นของการมีเวชันของสิ่งมีชีวิตตัวที่ i

m_{\max} คือ อัตราการมีเวชันสูงสุด

6) การอพยพ

หลักการการอพยพตามอัลกอริทึมการกระจายของสิ่งมีชีวิตตามภูมิศาสตร์มีลักษณะคล้ายกับวิธีการครอสโอเวอร์ในจินเนติกอัลกอริทึม ถ้าสิ่งมีชีวิตตัวที่ (H_j) ที่มีอัตราการอพยพเข้า (λ_j) น้อยกว่าค่าสุ่ม $\text{rand} [0,1]$ (ถ้า $\text{rand} \leq \lambda_j$) จะทำการอพยพ โดยจะอพยพสิ่งมีชีวิตที่ดี (H_j) เข้าสู่สิ่งมีชีวิตที่ไม่ดีดังสมการ 2.30 ซึ่งการเลือกสิ่งมีชีวิตที่ดีได้จากสมการที่ 2.30

$$H_i(SIV) \leftarrow H_j(SIV) \quad (2.30)$$

สำหรับงานวิจัยนี้ใช้วิธี IPOX (Improved Precedence Operation Crossover) เพื่อครอสโอเวอร์ส่วนลำดับโอเปอร์เรชัน ให้แบ่งงานเป็น 2 กลุ่ม ซึ่ง $G1 = \{J1\}$ และ $G2 = \{J2, J3\}$ จากนั้นคัดลอกทุกโอเปอร์เรชันของงานใน $G1$ ไปที่ชุดคำตอบใหม่โดยคงตำแหน่งเดิม และคัดเลือกทุกโอเปอร์เรชัน ของงานใน $G2$ ไปชุดคำตอบใหม่ตามลำดับในตำแหน่งที่ว่างที่เหลือ

ส่วนของเครื่องจักรส่วนใช้วิธี MPX (Multipoint Preservative Crossover) สุ่มค่าทุกชิ้นระหว่าง 0 หรือ 1 และถ้าเป็น 0 ให้คัดลอกหมายเลขเครื่องจักรจากสิ่งมีชีวิตที่อพยพเข้า (Immigrating habitat) ไปชุดปรับปรุง (Modified Habitat) ตำแหน่งเดิมของชุดสิ่งมีชีวิตที่อพยพเข้า แต่ถ้าเป็น 1 คัดลอกหมายเลขเครื่องจักรจากสิ่งมีชีวิตที่อพยพออก (Emigrating habitat) ไปชุดปรับปรุง ตำแหน่งเดิมของชุดสิ่งมีชีวิตที่อพยพออก

7) การมีเวชัน

นำชุดคำตอบที่ได้จากกระบวนการอพยพสู่ขั้นตอนการมีเวชัน ซึ่งมีขั้นตอนมีเวชันดังนี้

7.1) คัดเลือกชุดคำตอบเข้าสู่กระบวนการมีเวชันด้วยวิธีวงล้อสุ่มด้วยค่าความน่าจะเป็นในการมีเวชัน (m_i) ซึ่งคำนวณจากสมการ 2.29 จากนั้นสุ่มค่า rand ($\text{rand} \in [0,1]$) ถ้า $\text{rand} \leq m_i$ ชุดคำตอบนั้นจะถูกเลือกให้มมีเวชัน

7.2) วิธีการมีเวชันแบ่งเป็น 2 วิธี คือ วิธีที่ 1) ส่วนลำดับโอเปอร์เรชัน ให้ทำการสุ่มตำแหน่งยีนในชุดคำตอบ จากนั้นสุ่มเลือกตำแหน่งก่อนหน้าเพื่อทำการแทรก วิธีที่ 2) ส่วนของเครื่องจักรให้สุ่มตำแหน่งยีน 2 ตำแหน่งแล้วทำการเปลี่ยนเครื่องจักรใหม่จากเซตเครื่องจักรที่เป็นไปได้ของโอเปอร์เรชันนั้น

8) ทำการวนซ้ำจนตั้งแต่ขั้นตอนที่ 4 – 7 จนครบตามจำนวนรอบการทำงานที่กำหนด

งานวิจัยนี้ได้แบ่งการทดลองกับชุดข้อมูลมาตรฐานจำนวน 3 ชุดข้อมูล ได้แก่ ชุดของ

และเปรียบเทียบผลการทดลองระหว่างโมเดลที่นำเสนอกับโมเดลในงานวิจัยอื่นๆ ซึ่งเป็นโมเดลที่พัฒนาจากจินตคติอัลกอริทึม พบว่าโมเดลที่นำเสนอมีประสิทธิภาพดีกว่าในการแก้ไขจุดออดิแม็กซ์ โดยแยกตาม 3 วัตถุประสงค์ คือ 1) เวลาในการผลิตรวมน้อยสุด 2) เวลาในการผลิตเครื่องจักรที่ทำงานมากที่สุด และ 3) เวลาในการทำงานของทุกโอเปอเรชั่นของทุกงานบนเครื่องจักร

งานวิจัยนี้มีข้อสังเกตคืออัตราการผลิตเข้าและออกของสิ่งมีชีวิตพิจารณาจากตำแหน่งการจัดเรียงโดยไม่ได้คำนึงถึงความเหมาะสมของสิ่งมีชีวิต วิธีการครอสโอเวอร์และวิธีการมิวเตชันเหมือนจินตคติอัลกอริทึม และการประเมินค่าฟังก์ชันวัตถุประสงค์ของงานวิจัยซึ่งมี 3 วัตถุประสงค์ไม่ได้ใช้หลักการของการประเมินแบบหลายวัตถุประสงค์ (Multi-Objective) ทำให้มีหลายชุดคำตอบที่เป็นคำตอบที่เหมาะสมที่สุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การปรับปรุงจีเนติกอัลกอริทึม

สำหรับปัญหาการจัดการตารางการผลิตแบบตามสั่งยืดหยุ่น

การศึกษาในครั้งนี้ได้นำเสนออัลกอริทึมจากการปรับปรุงขั้นตอนการทำงานของจีเนติกอัลกอริทึมแบบดั้งเดิมที่นำเสนอโดย Holland ในปี ค.ศ. 1975 สำหรับการแก้ปัญหาการจัดการตารางการผลิตแบบตามสั่งยืดหยุ่นให้มีประสิทธิภาพดียิ่งขึ้น

3.1 การวิเคราะห์ปัญหาของจีเนติกอัลกอริทึมแบบดั้งเดิม

จีเนติกอัลกอริทึมเป็นอัลกอริทึมเชิงวิวัฒนาการที่ใช้ในการค้นหาคำตอบ ซึ่งเป็นที่ยอมรับในด้านประสิทธิภาพการทำงาน และนิยมใช้อย่างแพร่หลายโดยเฉพาะปัญหาการหาค่าเหมาะสมที่สุด วิธีการของจีเนติกอัลกอริทึมมีความเหมาะสมกับปัญหาการจัดการตารางการผลิตที่เป็นตัวเลขแบบไม่ต่อเนื่อง อย่างไรก็ตามเมื่อทำการวิเคราะห์พบว่ามีสิ่งที่ควรปรับปรุงให้สอดคล้องกับลักษณะสภาพความเป็นจริงและเหมาะสมกับปัญหาการจัดการตารางการผลิต มีรายละเอียดดังนี้

3.1.1 กระบวนการคัดเลือก

ขั้นตอนการคัดเลือกโครโมโซมเพื่อเป็นโครโมโซมต้นแบบ (Parent chromosomes) โดยวิธีการของจีเนติกอัลกอริทึมจะกำหนดโอกาสในการถูกเลือก (Probability) โครโมโซมจากค่าฟิตเนสหรือค่าความเหมาะสม ซึ่งมีวิธีการคัดเลือกหลายวิธี เช่น วิธีการแบ่งสัดส่วน (Proportionate selection) วิธีการจัดอันดับ (Ranking selection) วิธีการสุ่ม (Random selection) วิธีการวงล้อรูเล็ต (Roulette Wheel selection) และวิธีการจัดการแข่งขัน (Tournament selection) เป็นต้น ซึ่งในการศึกษาครั้งนี้ได้เลือกปรับปรุงกระบวนการคัดเลือกด้วยวิธีการวงล้อรูเล็ต (Roulette wheel selection) เนื่องจากเป็นวิธีที่ใช้ในจีเนติกอัลกอริทึมแบบดั้งเดิม โดยเมื่อนำมาใช้กับปัญหาการจัดการตารางการผลิตพบว่ามีปัญหาไม่สอดคล้องกับลักษณะของปัญหาดังกล่าว กล่าวคือ 1) ในแต่ละโครโมโซมจะประกอบด้วยยีน ซึ่งยีนคือตัวแทนของส่วนประกอบย่อยของปัญหา ดังนั้นยีนในแต่ละโครโมโซมอาจมีความหมายเหมือนกัน เช่น การเลือกเครื่องจักรเหมือนกัน แสดงว่าโครโมโซมมีความคล้ายคลึงกัน หรือมีความคลุมเครือกันระหว่างโครโมโซม และ 2) โครโมโซมที่เหมือนกันจะมีโอกาสถูกเลือกซ้ำทำให้เกิดปัญหาการลู่เข้าสู่คำตอบที่เร็วเกินไป

3.1.2 กระบวนการครอสโอเวอร์

เป็นวิธีการสร้างโครโมโซมทายาทที่ได้จากการรวมส่วนประกอบระหว่างสองโครโมโซมต้นแบบที่ถูกเลือกสำหรับผสมพันธุ์ (Mating pool) ในวิธีการจีเนติกอัลกอริทึมแบบดั้งเดิมนั้นใช้การสุ่มเลือกโครโมโซม ซึ่งโครโมโซมทั้งสองอาจเป็นโครโมโซมตัวเดียวกัน ซึ่งการสุ่มเลือกใช้โครโมโซมต้นแบบที่เหมือนกันมากจะทำให้ได้โครโมโซมทายาทเหมือนกับโครโมโซมต้นแบบสูง และมีผลต่อการลดโอกาสที่จะได้รับการถ่ายทอดลักษณะใหม่ๆ ของรุ่นลูก และเพิ่มโอกาสให้ยีนที่มีลักษณะไม่ดีเกิดขึ้น ส่งผลให้โครโมโซมรุ่นลูกขาดความหลากหลายและการค้นหาคำตอบที่เหมาะสมเป็นคำตอบแบบโลคอล

3.1.3 กระบวนการมิวเทชัน

การมิวเทชันหรือการกลายพันธุ์คือการเปลี่ยนแปลงตามธรรมชาติของยีนทำให้สิ่งมีชีวิตเกิดขึ้นมาใหม่มีลักษณะแตกต่างจากเดิมเป็นวิวัฒนาการจากรุ่นสู่รุ่นของกระบวนการทางพันธุกรรม วิธีการจีเนติกอัลกอริทึมใช้กระบวนการมิวเทชันเพื่อสร้างการเปลี่ยนแปลงและเพิ่มความหลากหลายให้กับโครโมโซม เพื่อเป็นการหลีกเลี่ยงการลู่เข้าหาคำตอบที่เร็วเกินไปทำให้พบคำตอบแบบโลคอลได้ ซึ่งวิธีการจีเนติกแบบดั้งเดิมจะใช้วิธีการสุ่มโครโมโซมเพื่อสร้างกระบวนการมิวเทชัน โดยไม่คำนึงถึงค่าฟิตเนสของโครโมโซม ทำให้ไม่สามารถกำหนดระดับการเปลี่ยนแปลงที่เหมาะสมกับโครโมโซมนั้นได้ โดยเฉพาะในกรณีที่โครโมโซมอยู่ใกล้จุดออฟติมัมและถูกมิวเทชันทำให้ห่างไกลจากคำตอบ ทำให้ใช้เวลาในการค้นหาคำตอบนานและอาจไม่พบจุดออฟติมัมในที่สุด

3.2 แนวทางการปรับปรุงจีเนติกอัลกอริทึมแบบดั้งเดิม

อัลกอริทึมที่นำเสนอได้ปรับปรุงขั้นตอนการทำงานของจีเนติกอัลกอริทึมแบบดั้งเดิม เพื่อเพิ่มประสิทธิภาพในการค้นหาของอัลกอริทึม ซึ่งได้เสนอ 3 แนวคิดมีรายละเอียดดังนี้

3.2.1 กระบวนการคัดเลือก

วิธีการคัดเลือก อัลกอริทึมได้นำเสนอ “วงล้อรูเล็ตแบบฟัซซี” (Fuzzy roulette wheel selection) แทนวิธีวงล้อรูเล็ตแบบดั้งเดิม ซึ่งความหมายของยีนในแต่ละโครโมโซมคือค่าความคล้ายระหว่างโครโมโซมโดยประยุกต์เข้ากับแนวคิดของฟัซซีเซตที่อนุญาตให้โครโมโซมมีความคลุมเครือระหว่างกันและสามารถทับซ้อนกันได้ แบ่งเป็น 4 ขั้นตอน ดังนี้

- 1) การกำหนดสัดส่วนโครโมโซมในวงล้อรูเล็ตแบบฟัซซี โดยจะนำเวลาในการผลิตรวมของโครโมโซมมาทำการแปลงเป็นค่าฟิตเนสเพื่อกำหนดความกว้างของพื้นที่ที่โครโมโซมดังสมการที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$A_s = \frac{f_s}{\sum_{n=1}^N f_n} \times 100 \quad (3.1)$$

เมื่อ f_s คือค่าพิเศษของโครโมโซม s และ N คือจำนวนประชากร

2) การจัดลำดับของโครโมโซมในวงล้อรูเล็ท

การจัดลำดับของโครโมโซมในวงล้อรูเล็ท จะคำนวณจากค่าความคล้าย (Similarity value) ของโครโมโซม ซึ่งค่าความคล้ายคำนวณจากการเปรียบเทียบระหว่าง 2 โครโมโซม ซึ่งแบ่งเป็น 2 ส่วน คือ 1) ส่วนของโอเปอร์เรชัน คือโอเปอร์เรชันที่มีลำดับการทำงานตำแหน่งเดียวกันกับคู่ของโครโมโซมที่นำมาเปรียบเทียบ และ 2) ส่วนของเครื่องจักร คือ โอเปอร์เรชันของงานเดียวกันมีการเลือกเครื่องจักรเหมือนกัน ดังนั้นถ้าโครโมโซมที่นำมาเปรียบเทียบเป็นโครโมโซมตัวเดียวกันค่าความคล้ายจะมีค่าสูงสุด คือ จำนวนโอเปอร์เรชันทั้งหมดของหนึ่งโครโมโซม

3) การวางตำแหน่งของโครโมโซมในวงล้อรูเล็ท

การวางตำแหน่งโครโมโซมเริ่มจากโครโมโซมตัวแรกด้วยฟังก์ชันสมาชิกสี่เหลี่ยมคางหมูด้วยพื้นที่ดังสมการที่ 3.1 ถัดมาโครโมโซมลำดับที่ 2 เลือกจากโครโมโซมที่เหลือที่มีความคล้ายกับโครโมโซมแรกมากที่สุด และคำนวณตำแหน่งการวางซ้าย ขวา และกึ่งกลางด้วยฟังก์ชันสมาชิกสามเหลี่ยมในพีชชีเซตตามสมการที่ 3.2 – 3.4 ทำลักษณะเช่นเดียวกันจนครบทุกโครโมโซม

$$LC_s = RC_{s-1} - \left[A_{s-1} \times \frac{IG}{G} \right] \quad (3.2)$$

$$RC_s = LC_s + A_s \quad (3.3)$$

$$CP_s = \frac{LC_s + RC_s}{2} \quad (3.4)$$

เมื่อ LC_s คือ มุมซ้ายของฟังก์ชันสมาชิกสามเหลี่ยมในพีชชีเซต s
 RC_s คือ มุมขวาของฟังก์ชันสมาชิกสามเหลี่ยมในพีชชีเซต s
 CP_s คือ จุดศูนย์กลางของฟังก์ชันสมาชิกสามเหลี่ยมในพีชชีเซต s
 IG คือ จำนวนยีนที่เหมือนกันในสองโครโมโซมที่อยู่ติดกัน
 G คือ จำนวนยีนทั้งหมดในโครโมโซม

4) การคัดเลือกโครโมโซมจากวงล้อรูเล็ตแบบฟิชชี

ในการคัดเลือกโครโมโซมจำนวนจากโครโมโซมที่มีค่าความเป็นสมาชิกสูงสุด กรณีโครโมโซมที่มีค่าความคล้ายกันมากหรืออาจจะเป็นโครโมโซมตัวเดียวกัน โครโมโซมจะมีการซ้อนทับกันทำให้ลดโอกาสในการเลือกโครโมโซมตัวเดิม และพื้นที่ในสล็อตของฟิชชีรูเล็ตจะมีขนาดลดลงจากวิธีการวงล้อรูเล็ตแบบเดิม และส่งผลให้โครโมโซมที่มีค่าความคล้ายน้อย (โครโมโซมที่มีความแตกต่าง) มีโอกาสถูกเลือกมากขึ้นซึ่งเป็นการหลีกเลี่ยงที่จะพบคำตอบแบบโลคอลได้

3.2.2 กระบวนการครอสโอเวอร์

การครอสโอเวอร์ระหว่าง 2 โครโมโซมที่มีความคล้ายกันหรือมีโครโมโซมที่เหมือนกันมาก ทำให้ไม่สร้างความหลากหลายในการกำเนิดทายาทและส่งผลต่อโอกาสของรุ่นลูกที่ได้รับการถ่ายทอดลักษณะเด่นที่ดีได้ ดังนั้นจึงได้นำเสนอการครอสโอเวอร์แบบแบ่งกลุ่ม (Cluster-Crossover operator) สามารถสรุปขั้นตอนดังนี้

1) แบ่งโครโมโซมต้นแบบเป็น 2 กลุ่มโดยวิธีการแบ่งกลุ่มเป็นลำดับชั้น (Hierarchical clustering) เนื่องจากเป็นข้อมูลความถี่ และใช้การรวมกลุ่มแบบสมบูรณ์ (Complete linkage method) โดยคำนวณจากค่าความคล้ายระหว่างโครโมโซมดังสมการ 3.5 เมื่อ G, H หมายถึงกลุ่มข้อมูล และ $s(e, f)$ หมายถึงความคล้ายระหว่างโครโมโซม e และ f

$$S(G, H) = \text{MAX}_{e \in G, f \in H} [s(e, f)] \quad (3.5)$$

2) ทำการสลับสายพันธุของโครโมโซมด้วยโอเปอเรชันแบบ POX (Precedence Operation Crossover) โดยการสร้างโครโมโซมทายาทตัวที่ 1 ได้จากการสุ่ม 1 ตำแหน่ง (One-Point Crossover)

3.2.3 กระบวนการมิวเตชัน

ขั้นตอนนี้เป็นวิธีการปรับปรุงคุณภาพของชุดคำตอบ เพื่อให้มีความหลากหลายของประชากรหรือโครโมโซม และเป็นการหลีกเลี่ยงการลู่เข้าสู่คำตอบเร็วเกินไป โดยนำเสนอวิธีการมิวเตชันแบบโลคอล (Mutation-LS operator) คือ การแบ่งกลุ่มโครโมโซมเพื่อทำการมิวเตชันโดยใช้อัลกอริทึมชิมูเลทเทดแอนนิลลิงสำหรับค้นหาคำตอบรอบข้างของชุดคำตอบปัจจุบัน เนื่องจากเป็นอัลกอริทึมที่มีประสิทธิภาพในการค้นหาคำตอบได้เร็วและได้จุดออปติ멈ดีกว่าวิธีอื่น เช่น การค้นหาแบบตาม [34, 35] โดยได้ทดสอบการเปรียบเทียบการทำงานระหว่างการค้นหาแบบตาม และ

ชิมูเลทเทดแอนนิลลิงอัลกอริทึมกับปัญหาการจัดตารางการผลิตแบบตามสั่ง (Job-Shop Scheduling) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Problem) จาก OR-Library [36] ประกอบด้วย La01 – La20 จำนวน 20 ชุดข้อมูล โดยกำหนดให้ขนาดของประชากรเท่ากับ 100 และใช้วิธีการสุ่มสลับโอเปอร์เรชันในการค้นหาค่าความเหมาะสมที่สุด โดยทำการรัน 2 ครั้ง ผลการทดลองครั้งที่ 1 พบว่า ซิมูเลทเทคแอนนิลลิ่งอัลกอริทึมสามารถค้นหาคำตอบได้ดีกว่าการค้นหาแบบตามู จำนวน 8 ชุดข้อมูล จำนวน 11 ชุดข้อมูลที่ค่าเท่ากัน และมีจำนวน 1 ชุดข้อมูลที่แยกว่า ผลการทดลองครั้งที่ 2 ได้เปลี่ยนตำแหน่งค่าเริ่มต้นเพื่อทดสอบอีกครั้งพบว่าซิมูเลทเทคแอนนิลลิ่งอัลกอริทึมสามารถค้นหาคำตอบได้ดีกว่าการค้นหาแบบตามู จำนวน 7 ชุดข้อมูล จำนวน 12 ชุดข้อมูลที่ค่าเท่ากัน และมีจำนวน 1 ชุดข้อมูลที่แยกว่า แสดงรายละเอียดในตารางที่ 3.1 จากการทดลองจะเห็นได้ว่าซิมูเลทเทคแอนนิลลิ่งอัลกอริทึมมีประสิทธิภาพดีกว่าจึงเป็นเหตุผลในการนำมาใช้ในการวิเวชัน

ตารางที่ 3.1 การเปรียบเทียบการคำตอบด้วยวิธีซิมูเลทเทคแอนนิลลิ่งอัลกอริทึมและการค้นหาแบบตามู สัญลักษณ์ \checkmark แสดงการพบคำตอบที่เป็นจุดออฟติมัม

ชุดข้อมูล (JSSP)	ครั้งที่ 1		ครั้งที่ 2	
	SA	TS	SA	TS
La01	\checkmark	\checkmark	\checkmark	\checkmark
La02		\checkmark	\checkmark	
La03	\checkmark			\checkmark
La04	\checkmark		\checkmark	
La05	\checkmark	\checkmark	\checkmark	\checkmark
La06	\checkmark	\checkmark	\checkmark	\checkmark
La07	\checkmark	\checkmark	\checkmark	\checkmark
La08	\checkmark	\checkmark	\checkmark	\checkmark
La09	\checkmark	\checkmark	\checkmark	\checkmark
La10	\checkmark	\checkmark	\checkmark	\checkmark
La11	\checkmark	\checkmark	\checkmark	\checkmark
La12	\checkmark	\checkmark	\checkmark	\checkmark
La13	\checkmark	\checkmark	\checkmark	\checkmark
La14	\checkmark	\checkmark	\checkmark	\checkmark
La15	\checkmark		\checkmark	\checkmark
La16	\checkmark		\checkmark	
La17	\checkmark		\checkmark	
La18	\checkmark		\checkmark	
La19	\checkmark		\checkmark	
La20	\checkmark		\checkmark	

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้มีวิธีการมิวเตชัน 4 วิธี ประกอบด้วย MU1, MU2, MU3 และ MU4 ตามลำดับ มีรายละเอียด ดังนี้

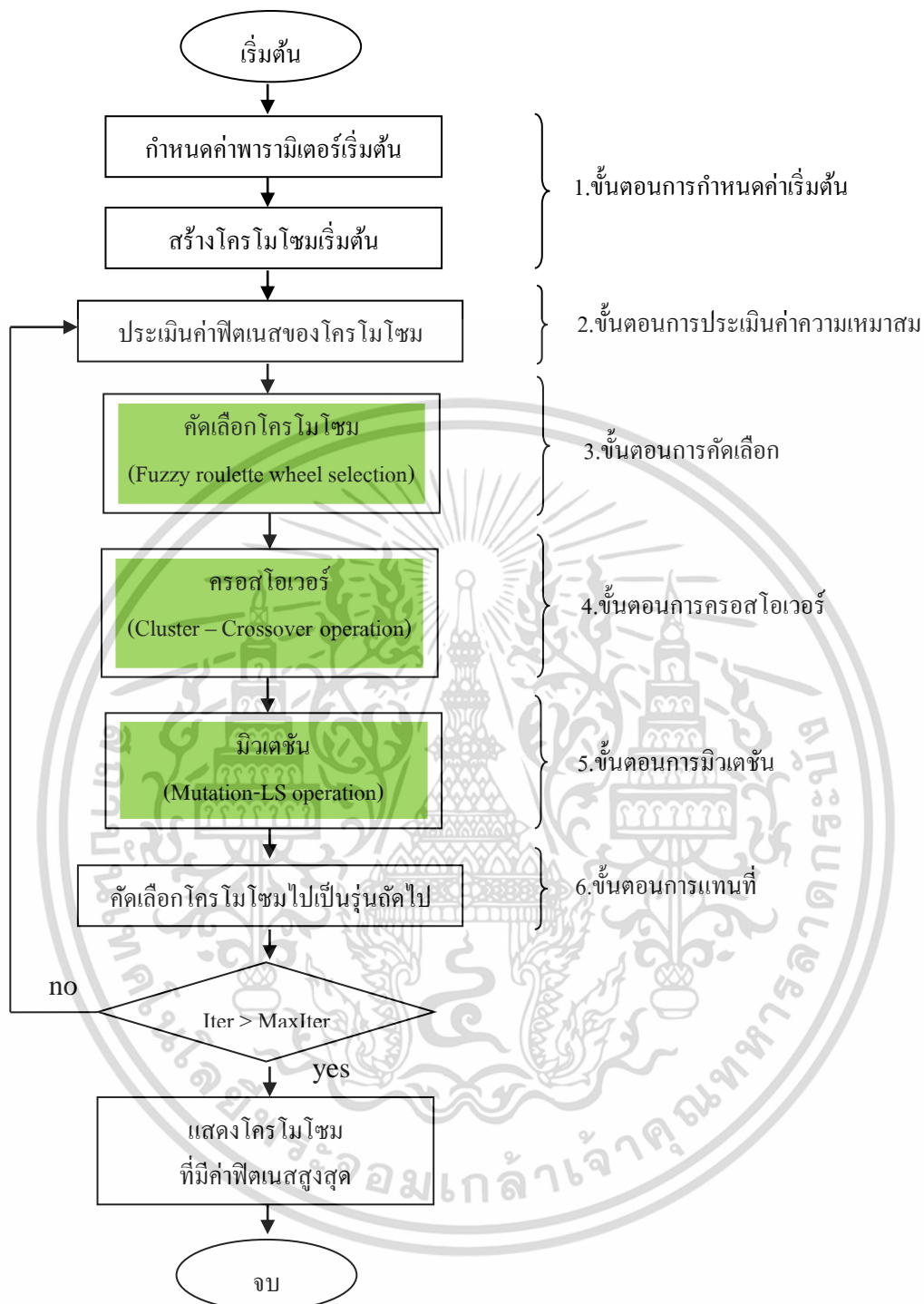
- MU1: เปลี่ยนลำดับโอเปอร์เรชันในส่วนโอเปอร์เรชันด้วยวิธีการสลับ ส่วนเครื่องจักรไม่เปลี่ยนแปลง
- MU2: เปลี่ยนลำดับโอเปอร์เรชันในส่วนโอเปอร์เรชันด้วยวิธีการสลับ และส่วนเครื่องจักรสุ่มเลือกเครื่องจักรใหม่แทนเครื่องจักรเดิม
- MU3: เปลี่ยนลำดับโอเปอร์เรชันในส่วนโอเปอร์เรชันแบบย้อนกลับ และส่วนเครื่องจักรสุ่มเลือกเครื่องจักรใหม่แทนเครื่องจักรเดิม
- MU4: สร้างชุดคำตอบใหม่โดยการสุ่มสำหรับส่วนของโอเปอร์เรชัน และส่วนของเครื่องจักรเลือกจากเครื่องจักรที่ใช้เวลาน้อยที่สุด (Shorted processing time: SPT)

3.3 โครงสร้างการทำงานของอัลกอริทึมที่นำเสนอ

อัลกอริทึมที่นำเสนอ MGA (Modified Genetic Algorithm) ได้ปรับปรุง 3 ส่วน คือ การคัดเลือกโครโมโซมต้นแบบด้วยวิธีวงล้อรูเล็ตแบบฟิชชี วิธีการครอสโอเวอร์แบบแบ่งกลุ่ม และวิธีการมิวเตชันแบบโลคอล ดังรูปที่ 3.1 โครงสร้างการทำงานของอัลกอริทึมที่นำเสนอ ประกอบด้วย 6 ขั้นตอนหลัก คือ

- 1) ขั้นตอนการกำหนดค่าเริ่มต้น ประกอบด้วยการกำหนดค่าพารามิเตอร์เริ่มต้น และสร้างโครโมโซมเริ่มต้น ซึ่งเป็นตัวแทนคำตอบที่ต้องการค้นหา
- 2) ขั้นตอนการประเมินค่าความเหมาะสมหรือค่าฟิตเนส เป็นการคำนวณหาความเหมาะสมของแต่ละโครโมโซมซึ่งในที่นี้กำหนดฟังก์ชันเป้าหมายคือ เวลารวมในการผลิตน้อยที่สุด
- 3) ขั้นตอนการคัดเลือกโครโมโซมเพื่อเป็นโครโมโซมต้นแบบ โดยพิจารณารายละเอียดของชุดคำตอบหรือโครโมโซมที่เหมือนกัน เพื่อนำมาสร้างวงล้อรูเล็ตแบบฟิชชี
- 4) ขั้นตอนการครอสโอเวอร์แบบแบ่งกลุ่ม โครโมโซมต้นแบบด้วยค่าความคล้าย เป็น 2 กลุ่มโดยใช้วิธีการแบ่งแบบขั้นตอน (Hierarchical Clustering) เพื่อให้โครโมโซมคล้ายกันอยู่ในกลุ่มเดียวกัน จากนั้นทำการจับคู่เพื่อทำการครอสโอเวอร์
- 5) ขั้นตอนการมิวเตชัน เพื่อกลายพันธุ์โครโมโซมทายาท (Offspring) โดยกำหนดให้มีวิธีการมิวเตชัน 4 วิธีเพื่อทำการค้นหาคำตอบแบบโอบอล และใช้ชิมูเลเทดแอนนิลลิงอัลกอริทึมมาค้นหาคำตอบแบบโลคอล
- 6) ขั้นตอนการแทนที่ จากการปรับปรุงโครโมโซมทายาทแล้วนำมารวมกับโครโมโซมเริ่มต้นจากนั้นคัดเลือกโครโมโซมที่ดีที่สุดไปเป็นประชากรเริ่มต้นรุ่นถัดไป ให้ทำซ้ำขั้นตอนที่ 2 – 6 จนครบจำนวนรอบที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 โครงสร้างการทำงานของอัลกอริทึมที่นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 อัลกอริทึมที่นำเสนอสำหรับปัญหาการจัดการตารางการผลิตแบบตามสั่งยืดหยุ่น

3.4.1 การแทนค่าโครโมโซมเริ่มต้น

ลักษณะข้อมูลสำหรับการจัดการตารางการผลิตแบบตามงานแบบยืดหยุ่นในวิธีการของ จีเนติก อัลกอริทึม ตัวแทนของชุดคำตอบจะจำลองโครโมโซมเป็นชุดคำตอบที่มีความเป็นไปได้ ซึ่งในแต่ละโครโมโซมประกอบด้วยองค์ประกอบที่เรียกว่า ยีนหรือโอเปอร์เรเตอร์ เช่น รูปที่ 3.2 เป็นตัวอย่างข้อมูลการจัดการตารางการผลิตแบบตามสั่งยืดหยุ่นที่มี 2 งาน 3 เครื่องจักร กำหนดให้หนึ่งโครโมโซมเป็นหนึ่งชุดคำตอบ ดังนั้นโครโมโซมจะประกอบด้วยโอเปอร์เรชันทั้งหมดของทุกงานเท่ากับ 6 คูณ 2 เท่ากับ 12 โอเปอร์เรชัน ซึ่งแบ่งเป็น 2 ส่วนคือ 1) ส่วนของโอเปอร์เรชัน และ 2) ส่วนของเครื่องจักร ตัวอย่างการสร้างโครโมโซมเริ่มต้น จำนวน 4 โครโมโซม ดังรูปที่ 3.3

Job	Operation	Machines		
		M ₁	M ₂	M ₃
J ₁	O _{1,1}	2	4	8
	O _{1,2}	5	4	7
	O _{1,3}	2	-	-
J ₂	O _{2,1}	10	11	10
	O _{2,2}	6	5	-
	O _{2,3}	2	-	6

รูปที่ 3.2 ตัวอย่างปัญหาการจัดการตารางการผลิตแบบตามงานแบบยืดหยุ่น ประเภท 2 งาน 3 เครื่องจักร

	ส่วนของโอเปอร์เรชัน						ส่วนของเครื่องจักร					
	1	2	3	4	5	6	1	2	3	4	5	6
โครโมโซม 1	1	2	2	2	1	1	2	2	2	1	1	1
โครโมโซม 2	1	2	1	1	2	2	1	1	1	1	1	3
โครโมโซม 3	2	2	1	1	2	1	1	1	2	2	1	1
โครโมโซม 4	2	2	2	1	1	1	3	2	3	2	2	1

รูปที่ 3.3 ตัวอย่างลักษณะ โครโมโซมเริ่มต้น

การสร้างตัวแทนคำตอบหรือโครโมโซม ดังรูปที่ 3.4 มีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ส่วนของโอเปอเรชัน (Operation part) ได้จากการสุ่มหมายเลขงานตามจำนวนโอเปอเรชันของแต่ละงาน เช่น [2 2 1 1 2 1] อธิบายดังตัวอย่างจากซ้ายไปขวาได้ดังนี้

- 2 หมายถึงงาน 2 โอเปอเรชันลำดับที่ 1 (O_{21})
- 2 หมายถึงงาน 2 โอเปอเรชันลำดับที่ 2 (O_{22})
- 1 หมายถึงงาน 1 โอเปอเรชันลำดับที่ 1 (O_{11})
- 1 หมายถึงงาน 1 โอเปอเรชันลำดับที่ 2 (O_{12})
- 2 หมายถึงงาน 2 โอเปอเรชันลำดับที่ 3 (O_{23})
- และ 1 หมายถึงงาน 1 โอเปอเรชันลำดับที่ 3 (O_{13}) ตามลำดับ

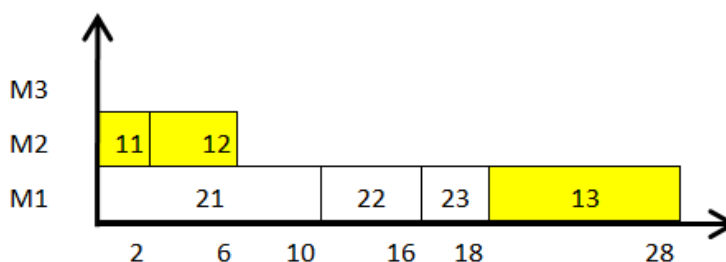
2) ส่วนของเครื่องจักร (Machine part) คือการเลือกเครื่องจักรให้กับโอเปอเรชันที่อ้างอิงจากข้อ 1 ซึ่งเลือกเครื่องจักรจากเซตของเครื่องจักรที่เป็นไปได้ที่สามารถทำงานในโอเปอเรชันนั้น

ส่วนของโอเปอเรชัน						ส่วนของเครื่องจักร					
2	2	1	1	2	1	1	1	2	2	1	1
O_{21}	O_{22}	O_{11}	O_{12}	O_{23}	O_{13}	M1	M1	M2	M2	M1	M1

รูปที่ 3.4 ส่วนประกอบของโครโมโซม

3.4.2 การคำนวณค่าฟิตเนส

เมื่อเข้ารหัสโครโมโซมตามลักษณะของปัญหาการจัดการตารางการผลิตได้ดังรูปที่ 3.4 จากนั้นนำมาประเมินประสิทธิภาพการจัดการตารางการผลิตในแผนภาพแกนต์ตามลำดับจากซ้ายไปขวาโดยเริ่มจาก O_{21} บนเครื่องจักรที่ 1 ด้วยเวลาเท่ากับ 10 และ O_{22} บนเครื่องจักรที่ 1 ด้วยเวลาเท่ากับ 6 ทำจนถึงโอเปอเรชันตัวสุดท้าย O_{13} บนเครื่องจักรที่ 1 ด้วยเวลาเท่ากับ 10 ตามลำดับได้เวลาในการผลิตรวม (C_{max}) ของโครโมโซมเท่ากับ 28 แสดงดังรูปที่ 3.5 ด้วยสมการ 3.6 จากนั้นนำค่าเวลาในการผลิตรวมมาคำนวณหาค่าฟิตเนสโดยค่าเวลาในการผลิตรวมน้อยจะมีค่าฟิตเนสสูงดังสมการ 3.7



รูปที่ 3.5 การประเมินการจัดการตารางการผลิตในหนึ่งโครโมโซม

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่ออยู่ภายใต้เงื่อนไขข้อนี้ ไม่สามารถนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์ได้

$$C_{\max} = \max_{1 \leq j \leq n} \{C_j\} \quad (3.6)$$

$$\text{Fitness} = \frac{1}{C_{\max}} \quad (3.7)$$

เมื่อ C_j คือ เวลาสิ้นสุดของโอเปอร์เรชันสุดท้ายของงาน j
 n คือ จำนวนงาน

3.4.3 การคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟิชชี มีขั้นตอนดังนี้

1) การกำหนดสัดส่วนโครโมโซมในวงล้อรูเล็ตแบบฟิชชี

กำหนดให้ตัวแทนชุดคำตอบเริ่มต้นกำหนดมี 4 โครโมโซม ดังรูปที่ 3.6 และค่าคำตอบของแต่ละโครโมโซมคือเวลาในการผลิตรวม จากนั้นคำนวณค่าฟิตเนสของโครโมโซมและพื้นที่ของวงล้อรูเล็ตตามสมการที่ 3.1, 3.6 และ 3.7 จะได้ค่าฟิตเนสของ 4 โครโมโซมและพื้นที่ของวงล้อรูเล็ตเท่ากับ 30.5% 27.4% 22% และ 20.1% ตามลำดับ ดังแสดงการคำนวณตารางที่ 3.2

โครโมโซม	ชุดคำตอบ
1	$O_{1,1}, O_{2,1}, O_{2,2}, O_{2,3}, O_{1,2}, O_{1,3}$
2	$O_{1,1}, O_{2,1}, O_{1,2}, O_{1,3}, O_{2,2}, O_{2,3}$
3	$O_{2,1}, O_{1,1}, O_{1,2}, O_{2,2}, O_{2,3}, O_{1,3}$
4	$O_{2,1}, O_{2,2}, O_{2,3}, O_{1,1}, O_{1,2}, O_{1,3}$

รูปที่ 3.6 กำหนดตัวแทนชุดคำตอบ (โครโมโซม)

ตารางที่ 3.2 ตัวอย่างการคำนวณค่าฟิตเนสของโครโมโซมในวงล้อรูเล็ตแบบฟิชชี

โครโมโซม	เวลาในการผลิต (c_{\max})	ค่าฟิตเนส	พื้นที่ในวงล้อรูเล็ต (ตามสมการ 3.1)
1	20	$1/20 = 0.050$	$(0.050 \cdot 100)/0.164 = 30.5$
2	22	$1/22 = 0.045$	$(0.045 \cdot 100)/0.164 = 27.4$
3	28	$1/28 = 0.036$	$(0.036 \cdot 100)/0.164 = 22.0$
4	30	$1/30 = 0.033$	$(0.033 \cdot 100)/0.164 = 20.1$
Sum	100	0.164	100.0

2) การจัดลำดับของโครโมโซมในวงล้อรูเล็ต

การวางลำดับจะเริ่มต้นจากโครโมโซมตำแหน่งแรก (จากรูปที่ 3.7(ก) เริ่มโครโมโซมที่ 1) จากนั้นเปรียบเทียบกับโครโมโซมที่เหลือ (โครโมโซมที่ 2, 3 และ 4) โครโมโซมตัวที่มีค่าความคล้ายกับโครโมโซมที่ 1 มากที่สุดจะถูกเลือกเป็นตำแหน่งที่ 2 จากรูปที่ 3.7(ก) โครโมโซมที่ 4 ถูก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกด้วยค่าความคล้ายสูงสุดเท่ากับ 5 จากนั้นพิจารณาจากโครโมโซมที่ 4 กับโครโมโซมที่เหลือ (โครโมโซมที่ 2 และ 3) ดังรูปที่ 3.7(ข) ทำเช่นเดียวกันจนครบทุกโครโมโซม ดังรูปที่ 3.7(ค) ดังนั้นการวางลำดับของโครโมโซมในวงล้อรูปที่ 3.7(ค) คือ โครโมโซม [1 – 4 – 3 – 2] ตามลำดับ และสามารถแสดงอยู่ในรูปของตารางค่าความคล้าย (Matrix similarity) ของโครโมโซมในรูปที่ 3.8

	โครโมโซม	ชุดคำตอบ
1	O _{1,1} , O _{2,1} , O _{2,2} , O _{2,3} , O _{1,2} , O _{1,3}	M ₂ , M ₂ , M ₂ , M ₁ , M ₁ , M ₁
2	O _{1,1} , O _{2,1} , O _{1,2} , O _{1,3} , O _{2,2} , O _{2,3}	M ₁ , M ₁ , M ₁ , M ₁ , M ₁ , M ₃
3	O _{2,1} , O _{1,1} , O _{1,2} , O _{2,2} , O _{2,3} , O _{1,3}	M ₂ , M ₁ , M ₂ , M ₂ , M ₂ , M ₁
4	O _{2,1} , O _{2,2} , O _{2,3} , O _{1,1} , O _{1,2} , O _{1,3}	M ₃ , M ₂ , M ₃ , M ₂ , M ₂ , M ₁

Chromosome	Chromosome	Similarity score (Max = 12)	Selected
1	2	2+2 = 4	[1,4]
1	3	1+3 = 4	
1	4	2+3 = 5	

(ก)

	โครโมโซม	ชุดคำตอบ
4	O _{2,1} , O _{2,2} , O _{2,3} , O _{1,1} , O _{1,2} , O _{1,3}	M ₃ , M ₂ , M ₃ , M ₂ , M ₂ , M ₁
2	O _{1,1} , O _{2,1} , O _{1,2} , O _{1,3} , O _{2,2} , O _{2,3}	M ₁ , M ₁ , M ₁ , M ₁ , M ₁ , M ₃
3	O _{2,1} , O _{1,1} , O _{1,2} , O _{2,2} , O _{2,3} , O _{1,3}	M ₂ , M ₁ , M ₂ , M ₂ , M ₂ , M ₁

Chromosome	Chromosome	Similarity score (Max = 12)	Selected
4	2	0+2 = 2	[1,4]
4	3	2+3 = 5	

(ข)

	โครโมโซม	ชุดคำตอบ
3	O _{2,1} , O _{1,1} , O _{1,2} , O _{2,2} , O _{2,3} , O _{1,3}	M ₂ , M ₁ , M ₂ , M ₂ , M ₂ , M ₁
2	O _{1,1} , O _{2,1} , O _{1,2} , O _{1,3} , O _{2,2} , O _{2,3}	M ₁ , M ₁ , M ₁ , M ₁ , M ₁ , M ₃

Chromosome	Chromosome	Similarity score (Max = 12)	Selected
3	2	1+2 = 3	[1,4]
			[4,3]
			[3,2]

(ค)

รูปที่ 3.7 กระบวนการจัดลำดับโครโมโซม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		โครโมโซม			
		1	2	3	4
โครโมโซม	1	-			
	2	4	-		
	3	4	3	-	
	4	5	2	5	-

รูปที่ 3.8 เมตริกซ์ค่าความคล้ายของโครโมโซม

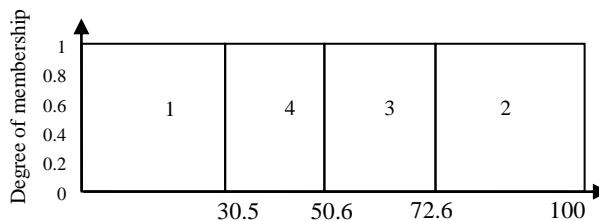
3) การวางตำแหน่งของโครโมโซมในวงล้อรูปสี่เหลี่ยม

การวางตำแหน่งโครโมโซมเริ่มจากโครโมโซมแรก (โครโมโซมที่ 1) ด้วยฟังก์ชันสมาชิกสี่เหลี่ยมคางหมู ด้วยตำแหน่ง 0, 0, 15.25 และ 30.5 ตามลำดับ จากนั้นตำแหน่งการวางของโครโมโซมถัดมา (โครโมโซมที่ 4) ที่มีความคล้ายกับโครโมโซมแรกแสดงถึงการมีพื้นที่ทับซ้อนกันระหว่างโครโมโซม การคำนวณโครโมโซมตำแหน่งที่ 2 ด้วยฟังก์ชันสมาชิกสามเหลี่ยมในพีชชีเซตตามสมการที่ 3.2–3.4 ด้วยค่าความคล้ายเท่ากับ 5 ยื่นจากทั้งหมด 12 ยื่น ดังนั้นตำแหน่งมุมซ้ายกลาง และมุมขวาของโครโมโซมนี้ประกอบด้วย LC_4 , RC_4 และ CP_4 คือ 17.8, 37.9 และ 20.1 ตามลำดับ ทำเช่นเดียวกันจนครบทุกโครโมโซมในพีชชีเซต ดังนั้นตำแหน่งการวางของโครโมโซมในวิธีวงล้อรูปสี่เหลี่ยมแบบพีชชีดังรูปที่ 3.9



รูปที่ 3.9 ตัวอย่างโครโมโซมในวงล้อรูปสี่เหลี่ยมแบบพีชชี

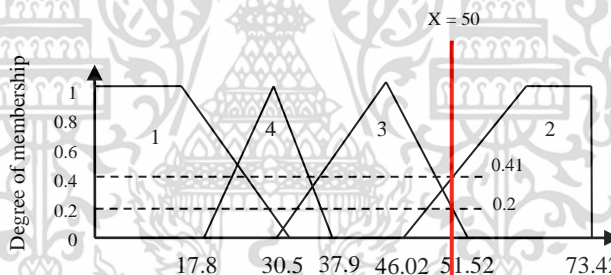
เมื่อนำมาเปรียบเทียบกับวิธีวงล้อรูปสี่เหลี่ยมแบบดั้งเดิมการกำหนดสัดส่วนและการวางตำแหน่งของโครโมโซมจะแบ่งแยกกันอย่างชัดเจนดังรูปที่ 3.10 ซึ่งเมื่อใช้วิธีการแบบวงล้อรูปสี่เหลี่ยมแบบพีชชีโครโมโซมสามารถทับซ้อนกันได้ในรูปแบบของพีชชีเซต ดังรูปที่ 3.9 โดยใช้ฟังก์ชันสี่เหลี่ยมคางหมู (Trapezoidal membership function) สำหรับพีชชีเซตที่อยู่ซ้ายสุด และขวาสุด และฟังก์ชันสามเหลี่ยม (Triangular membership function) สำหรับพีชชีเซตที่อยู่กลาง



รูปที่ 3.10 ตัวอย่างโครโมโซมในวิธีวงล้อรูปแบบดั้งเดิม

3) การคัดเลือกโครโมโซมจากวงล้อรูปแบบฟัซซี

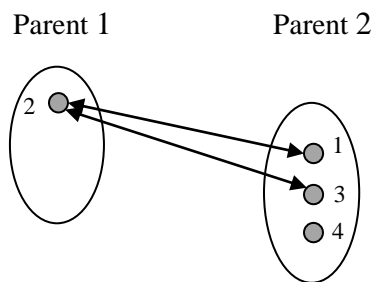
ในการคัดเลือกโครโมโซมแต่ละครั้งให้สุ่มค่าระหว่าง 0 ถึง 1 และนำมาคูณกับขอบเขตขงค่าของฟังก์ชันสมาชิกในฟัซซีรูเล่ด ดังในรูปที่ 3.10 คือ 73.42 จากนั้นคำนวณค่าความเป็นสมาชิกของแต่ละโครโมโซม โดยที่โครโมโซมที่มีค่าความเป็นสมาชิกสูงสุดจะถูกเลือก ดังรูปที่ 3.11 สุ่มค่า $x = 50$ มีความเป็นสมาชิกอยู่ 2 ฟัซซีเซต คือ $\mu_3(x) = 0.2$ และ $\mu_2(x) = 0.41$ ดังนั้นโครโมโซมที่ 2 จะถูกเลือกเป็นโครโมโซมต้นแบบด้วยค่าความเป็นสมาชิกที่สูงกว่า ให้ทำซ้ำจนกว่าครบจำนวนโครโมโซมต้นแบบที่กำหนดให้เท่ากับจำนวนประชากรเริ่มต้น



รูปที่ 3.11 ตัวอย่างการเลือกโครโมโซมด้วยค่าความเป็นสมาชิก

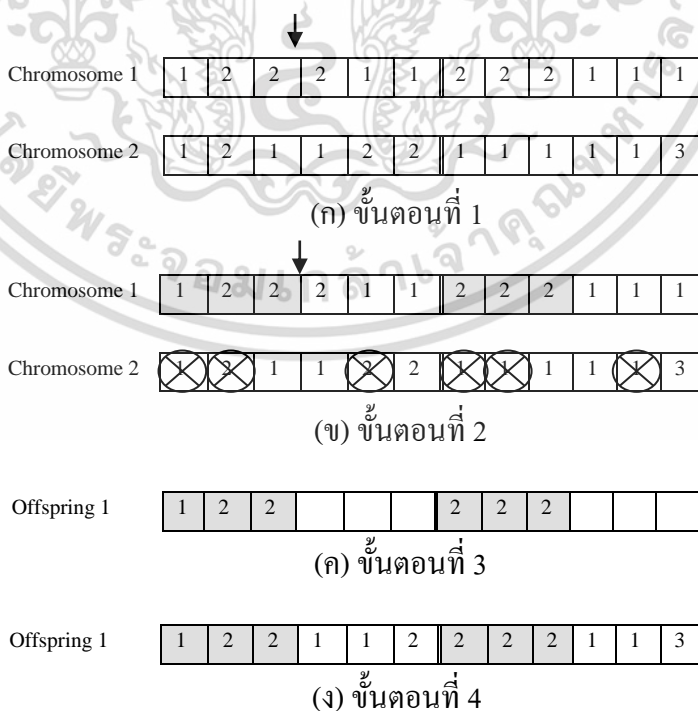
3.4.4 การครอสโอเวอร์แบบแบ่งกลุ่ม มีขั้นตอน ดังนี้

1) แบ่งโครโมโซมต้นแบบเป็น 2 กลุ่ม ด้วยวิธีการรวมกลุ่มแบบสมบูรณ์ (Complete linkage method) จากรูปที่ 3.12 แสดง 4 โครโมโซมที่แบ่งเป็น 2 กลุ่มซึ่งมีจำนวนสมาชิกไม่เท่ากัน สุ่มเลือก 1 โครโมโซมจากแต่ละกลุ่มเพื่อจับคู่ผสมพันธุ์สามารถจับคู่เพื่อผสมพันธุ์ได้ 2 คู่ คือ โครโมโซม 2-1 และ โครโมโซม 2-3



รูปที่ 3.12 ตัวอย่างการแบ่งกลุ่มโครโมโซมต้นแบบ

2) ทำการสลับสายพันธุของโครโมโซมด้วยโอเปอเรชันแบบ POX (Precedence Operation Crossover) การสร้างโครโมโซมทายาทตัวที่ 1 ได้จากการสุ่ม 1 ตำแหน่ง (One-Point Crossover) ในส่วนของโอเปอเรชันของโครโมโซมต้นแบบตัวที่ 1 และทำการคัดลอกทุกโอเปอเรชัน (ทั้งส่วนโอเปอเรชัน และส่วนเครื่องจักร) จากด้านซ้ายสุดจนถึงตำแหน่งที่สุ่มไปไว้โครโมโซมทายาทตัวที่ 1 จากนั้นลบโอเปอเรชันและเครื่องจักรที่ได้คัดลอกนั้นออกจากโครโมโซมต้นแบบตัวที่ 2 และทำการคัดลอกโอเปอเรชันและเครื่องจักรที่เหลือในโครโมโซมต้นแบบตัวที่ 2 ไปอยู่ในโครโมโซมทายาทตัวที่ 1 ตามลำดับจนครบทุกโอเปอเรชัน ส่วนโครโมโซมทายาทตัวที่ 2 ให้ทำเช่นเดียวกันกับโครโมโซมทายาทตัวที่ 1 แต่จะทำการสุ่มตำแหน่งครอสโอเวอร์ในโครโมโซมต้นแบบตัวที่ 2 แทน ดังรูปที่ 3.13 วิธีการนี้จะได้การจัดลำดับใหม่และเปลี่ยนเครื่องจักรทำงานใหม่ในโครโมโซมทายาทตัวที่ 1



รูปที่ 3.13 ตัวอย่างการครอสโอเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิใช่เผยแพร่ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 การมิมิตชันแบบโลคอล มีขั้นตอนดังนี้

1) คัดเลือกโครโมโซมสำหรับมิมิตชัน พิจารณาจากโครโมโซมทายาทที่ได้จากการครอสโอเวอร์ในขั้นตอนก่อนหน้า นำมาสุ่มค่า $r \in [0,1]$ เปรียบเทียบกับความน่าจะเป็นในการมิมิตชัน (P_m) ถ้า $r < P_m$ โครโมโซมนั้นจะถูกเลือกอยู่ในกลุ่มโครโมโซมเพื่อมิมิตชัน เมื่อได้กลุ่มโครโมโซมที่จะทำการมิมิตชันแล้วให้ทำการเรียงลำดับโครโมโซมดังกล่าวตามค่าฟิตเนสจากค่าสูงสุดไปหาค่าฟิตเนสต่ำสุด

2) แบ่งกลุ่มโครโมโซมที่จะทำการมิมิตชันเป็น 4 กลุ่มจำนวนเท่าๆ กัน คือ Fs1, Fs2, Fs3 และ Fs4 ตามลำดับ จากนั้นคำนวณหาค่าเฉลี่ยของแต่ละกลุ่มเพื่อกำหนดสัดส่วนของพื้นที่แบบวงล้อรูเล็ตแบบดั้งเดิม โดยกลุ่มที่มีค่าเฉลี่ยฟิตเนสสูงจะมีพื้นที่ในวงล้อมากกว่ากลุ่มที่มีค่าเฉลี่ยฟิตเนสต่ำ

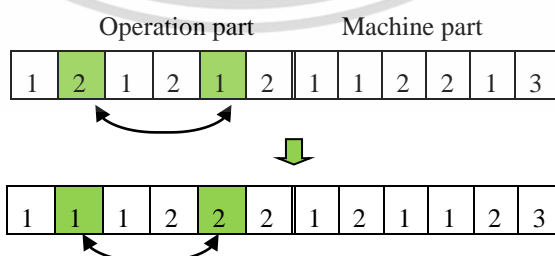
3) ใช้อัลกอริทึมชิมูเลทเทคแอนนิลลิงสำหรับค้นหาคำตอบแบบโลคอลของแต่ละกลุ่ม โดยกำหนดวิธีการสร้างชุดคำตอบรอบข้างจากชุดคำตอบปัจจุบัน 4 วิธีได้แก่

- MU1: สุ่ม 2 โอเปอร์เรชันหรือยีนเฉพาะในส่วนโอเปอร์เรชันแล้วสลับตำแหน่งเพื่อเปลี่ยนลำดับโอเปอร์เรชันใหม่ จาก $J_1O_1, J_2O_1, J_1O_2, J_2O_2, J_1O_3, J_2O_3$ เป็น $J_1O_1, J_1O_2, J_1O_3, J_2O_1, J_2O_2, J_2O_3$ ซึ่งในส่วนของเครื่องจักรไม่เปลี่ยนแปลง ดังรูปที่ 3.14

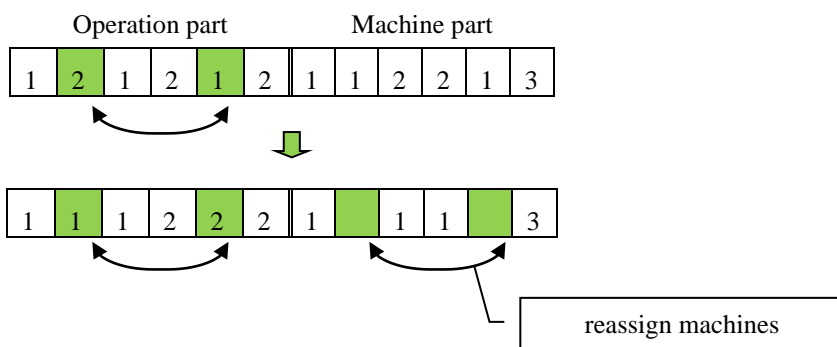
- MU2: สุ่ม 2 โอเปอร์เรชันหรือยีนในส่วนโอเปอร์เรชันแล้วสลับตำแหน่งเพื่อเปลี่ยนลำดับโอเปอร์เรชันใหม่ ส่วนของเครื่องจักรให้สุ่มเลือกเครื่องจักรใหม่เฉพาะโอเปอร์เรชันตำแหน่งที่สุ่ม ดังรูปที่ 3.15

- MU3: สุ่ม 2 โอเปอร์เรชันหรือยีนในส่วนโอเปอร์เรชันแล้วจัดลำดับใหม่แบบย้อนกลับ (reverses) ส่วนของเครื่องจักรให้สุ่มเลือกเครื่องจักรใหม่เฉพาะโอเปอร์เรชันตำแหน่งที่สุ่ม ดังรูปที่ 3.16

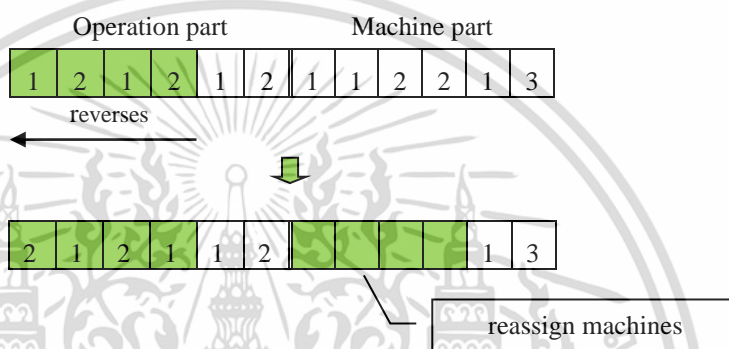
- MU4: สร้างโอเปอร์เรชันใหม่โดยการสุ่มสำหรับส่วนของโอเปอร์เรชัน และส่วนของเครื่องจักรเลือกจากเครื่องจักรที่ใช้เวลาน้อยที่สุด (Shorted processing time: SPT) ดังรูปที่ 3.17



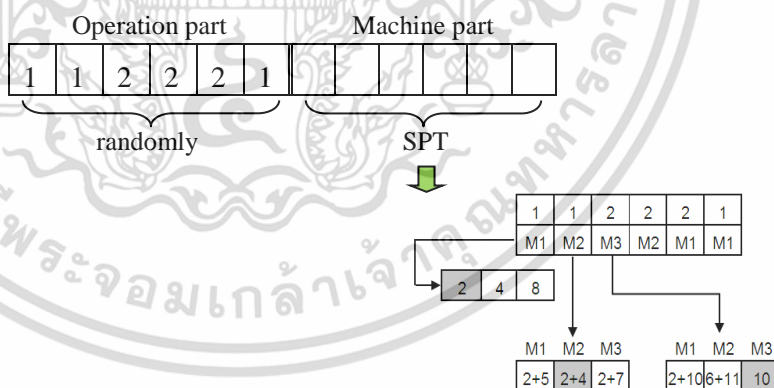
รูปที่ 3.14 การมิมิตชันแบบ MU1



รูปที่ 3.15 การมิวเตชันแบบ MU2



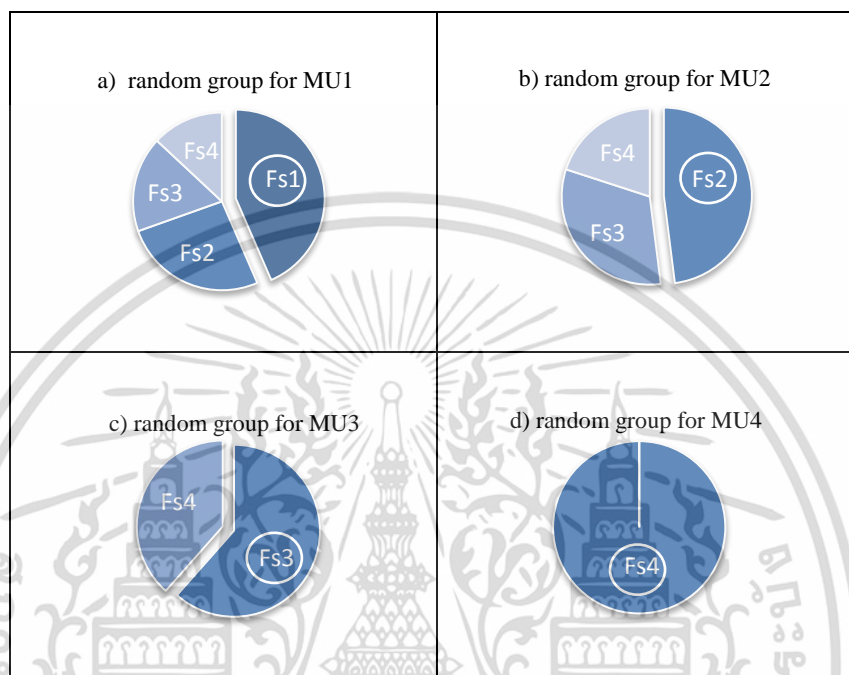
รูปที่ 3.16 การมิวเตชันแบบ MU3



รูปที่ 3.17 การมิวเตชันแบบ MU4

4) เลือกวิธีการค้นหาคำตอบของแต่ละกลุ่ม กำหนดให้เริ่มวิธีการมิวเตชันแบบ MU1, MU2, Mu3 และ MU4 ตามลำดับ การเลือกโครโมโซมตามที่แบ่งกลุ่มในขั้นตอนที่ 2 ด้วยการสุ่มค่าระหว่าง $r[0,1]$ ในวงล้อรูเล็ตเพื่อสำหรับกลุ่มที่จะถูกเลือก ดังตัวอย่างในรูปที่ 3.18(a) แบ่งกลุ่มโครโมโซมตามค่าเฉลี่ยฟิตเนสของกลุ่มซึ่งมี 4 กลุ่มเลือกสุ่มกลุ่มเพื่อใช้สำหรับการมิวเตชันแบบ MU1 ซึ่งสุ่มได้กลุ่ม Fs1 จากนั้นปรับพื้นที่ในวงล้อรูเล็ตสำหรับกลุ่มที่เหลือ ลำดับถัดไปสุ่มเลือกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มโครโมโซมที่เหลือสำหรับวิธีการมิวเทชัน MU2 ซึ่งสุ่มได้กลุ่ม Fs2 จากนั้นปรับพื้นที่ในวงล้อรู เล็ดสำหรับกลุ่มที่เหลือดังรูปที่ 3.18(b) ทำวิธีเดียวกันสำหรับการมิวเทชันแบบ MU3 และ MU4 ดัง รูปที่ 3.18(c) และ (d) ตามลำดับจนครบทุกกลุ่ม ถ้าพบคำตอบที่ดีกว่าเดิมให้ทำการปรับค่าชุด คำตอบของแต่ละโครโมโซม



รูปที่ 3.18 กระบวนการมิวเทชันแบบโคลอด

3.4.6 การสืบพันธุ์และสิ้นสุดการทำงาน

โครโมโซมที่ดีที่สุดจำนวนเท่ากับประชากรเริ่มต้น (Popsizе) จะถูกเลือกให้เป็นประชากร รุ่นถัดไป ขั้นตอนนี้จะทำซ้ำจนกว่าจำนวนรอบการทำงาน (Iter) ครบตามที่กำหนดไว้ (MaxIter) หรือตามเงื่อนไขที่กำหนดจะหยุดการทำงาน

บทที่ 4

การทดสอบการปรับปรุงจีเนติกอัลกอริทึม สำหรับปัญหาการจัดการตารางการผลิตแบบตามสั่งยืดหยุ่น

การศึกษาวิจัยครั้งนี้ได้นำเสนออัลกอริทึมที่เกิดจากการปรับปรุงขั้นตอนการทำงานของจีเนติกอัลกอริทึมแบบดั้งเดิมสำหรับปัญหาการจัดการตารางการผลิตแบบตามสั่งยืดหยุ่น 3 ส่วน ได้แก่ 1) การคัดเลือกประชากรหรือโครโมโซมเพื่อเป็นโครโมโซมต้นแบบด้วยวิธีวงล้อสุ่มแบบฟัซซี เนื่องจากในแต่ละโครโมโซมอาจมีส่วนคล้ายกันหรือมีความหมายเหมือนกัน ทำให้เกิดการวางซ้อนทับของโครโมโซมได้ 2) การครอสโอเวอร์แบบแบ่งกลุ่ม เพื่อเป็นการหลีกเลี่ยงการจับคู่โครโมโซมที่มีลักษณะคล้ายกันและเพื่อสร้างความหลากหลายให้กับโครโมโซมทายาท และ 3) การมีวเตชันแบบโลคอลสำหรับเพิ่มความหลากหลายของการเปลี่ยนแปลงโครโมโซมเพื่อเลี่ยงคำตอบแบบโลคอล ซึ่งได้กล่าวรายละเอียดในบทที่ผ่านมาแล้ว

การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA แบ่งเป็น 4 ส่วน ดังนี้

- 1) การทดสอบเพื่อศึกษาปัจจัยที่มีผลต่อประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอเปรียบเทียบกับจีเนติกอัลกอริทึมแบบดั้งเดิม โดยแยกทดสอบแต่ละปัจจัยประกอบด้วย
 - 1.1) จีเนติกอัลกอริทึมแบบดั้งเดิม (GA)
 - 1.2) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อสุ่มแบบฟัซซี (GA + Fuzzy Roulette Wheel Selection)
 - 1.3) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการครอสโอเวอร์แบบแบ่งกลุ่ม (GA + Cluster-Crossover operation)
 - 1.4) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการมีวเตชันแบบโลคอล (GA + Mutation Local Search operation)
 - 1.5) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อสุ่มแบบฟัซซีและการครอสโอเวอร์แบบแบ่งกลุ่ม (GA + Fuzzy Roulette Wheel Selection) + (GA + Cluster-Crossover operation)
 - 1.6) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อสุ่มแบบฟัซซีและการมีวเตชันแบบโลคอล (GA + Fuzzy Roulette Wheel Selection) + (GA + Mutation Local Search operation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.7) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการครอสโอเวอร์แบบแบ่งกลุ่มและการมิวเตชันแบบโลคอล (GA + Cluster-Crossover operation) + (GA + Mutation Local Search operation)
- 1.8) อัลกอริทึมที่นำเสนอ MGA (Fuzzy Roulette Wheel Selection + Cluster-Crossover operation + Mutation Local Search operation)
- 2) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Brandimarte's (BRdata) โดยเปรียบเทียบกับอัลกอริทึมอื่น
- 3) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Fattahi et al. (Fdata) โดยเปรียบเทียบกับอัลกอริทึมอื่น
- 4) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Barnes and Chamber library (BCdata) โดยเปรียบเทียบกับผลการทดลองในงานวิจัยของ Demir และ Isleyen [40]

4.1 ข้อมูลที่ใช้ในการทดสอบ

ชุดข้อมูลมาตรฐานที่ใช้ในการทดสอบสำหรับปัญหาการจัดการตารางการผลิตแบบตามสั่ง ยืดหยุ่น เป็นการหาค่าต่ำสุด (Minimization problem) ประกอบด้วย Brandimarte's (BRdata) [20] จำนวน 10 ชุด Fattahi et al. (Fdata) [10] จำนวน 20 ชุดและ Barnes and Chamber library (BCdata) [27] จำนวน 21 ชุดแสดงรายละเอียดแต่ละชุดข้อมูล จำนวนงาน จำนวนเครื่องจักร จำนวนโอเปอร์เรชัน และค่าคำตอบที่ดีที่สุด (Lower Bound: LB) ดังตารางที่ 4.1 4.2 และ 4.3 ตามลำดับ

ตารางที่ 4.1 ชุดข้อมูล BRdata

ลำดับ	ปัญหา	จำนวนงาน	จำนวนเครื่องจักร	จำนวนโอเปอร์เรชัน	LB
1	Mk01	10	6	5 – 7	36
2	Mk02	10	6	5 – 7	24
3	Mk03	15	8	10 – 10	204
4	Mk04	15	8	3 – 10	48
5	Mk05	15	4	5 – 10	168
6	Mk06	10	15	15 – 15	33
7	Mk07	20	5	5 – 5	133
8	Mk08	20	10	10 – 5	523
9	Mk09	20	10	10 – 15	299
10	Mk10	20	15	10 – 15	165

ตารางที่ 4.2 ชุดข้อมูล Fdata

ลำดับ	ปัญหา	จำนวนงาน	จำนวนเครื่องจักร	จำนวนโอเปอเรชั่น	LB
1	SFJS1	2	2	2	66
2	SFJS2	2	2	2	107
3	SFJS3	3	2	2	221
4	SFJS4	3	2	2	355
5	SFJS5	3	2	2	119
6	SFJS6	3	2	3	320
7	SFJS7	3	5	3	397
8	SFJS8	3	4	3	253
9	SFJS9	3	3	3	210
10	SFJS10	4	5	3	516
11	MFJS1	5	6	3	396
12	MFJS2	5	7	3	396
13	MFJS3	6	7	3	396
14	MFJS4	7	7	3	496
15	MFJS5	7	7	3	414
16	MFJS6	8	7	3	469
17	MFJS7	8	7	4	619
18	MFJS8	9	8	4	619
19	MFJS9	11	8	4	764
20	MFJS10	12	8	4	944

ตารางที่ 4.3 ชุดข้อมูล BCdata

ลำดับ	ปัญหา	จำนวนงาน	จำนวนเครื่องจักร	จำนวนโอเปอเรชั่น	LB
1	mt10c1	10	11	10	655
2	mt10cc	10	12	10	655
3	mt10x	10	11	10	655
4	mt10xx	10	12	10	655
5	mt10xxx	10	13	10	655
6	mt10xy	10	12	10	655

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานวิจัยเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ชุดข้อมูล BCdata (ต่อ)

ลำดับ	ปัญหา	จำนวนงาน	จำนวนเครื่องจักร	จำนวนโอเปอเรชั่น	LB
7	mt10xyz	10	13	10	655
8	setb4c9	15	11	10	857
9	setb4cc	15	12	10	857
10	setb4x	15	11	10	846
11	setb4xx	15	12	10	847
12	setb4xxx	15	13	10	846
13	setb4xy	15	12	10	845
14	setb4xyz	15	13	10	838
15	seti5c12	15	16	15	1027
16	seti5cc	15	17	15	955
17	seti5x	15	16	15	955
18	seti5xx	15	17	15	955
19	seti5xxx	15	18	15	955
20	seti5xy	15	17	15	955
21	seti5xyz	15	18	15	955

4.2 เครื่องมือที่ใช้ในการวัดประสิทธิภาพ

- 1) ค่าผลลัพธ์ คือ เวลาในการผลิตรวมต่ำสุดที่ได้จากการทดลอง โดยได้ทำการเปรียบเทียบหึ่งค่าคำตอบที่ดีที่สุดแทนด้วย C_{best}
- 2) ค่าเฉลี่ย (mean) คือ ผลรวมของค่าผลลัพธ์ที่ได้จากการทดลองแล้วหารด้วยจำนวนครั้งในการทดลอง เพื่อหาตัวแทนของค่าผลลัพธ์ที่ได้จากการทดลองซ้ำหลายๆ ครั้ง

$$\bar{C} = \frac{\sum_{i=1}^n C_i}{n} \quad (4.1)$$

- เมื่อ \bar{C} คือค่าเฉลี่ยของค่าผลลัพธ์ที่ได้จากการทดลอง
 C_i คือค่าผลลัพธ์ที่ได้จากการทดลองที่ i
 n คือจำนวนข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ค่าคลาดเคลื่อนสัมพัทธ์ (relative error: RE) คือ ค่าคลาดเคลื่อน

$$RE(\%) = \left[\frac{C_{\text{best}} - \text{BKS}}{\text{BKS}} \right] \times 100 \quad (4.2)$$

เมื่อ C_{best} คือค่าคำตอบที่ดีที่สุดของอัลกอริทึม
 BKS คือค่าที่ดีที่สุดของชุดคำตอบ (LB)

4.3 การทดสอบเพื่อศึกษาปัจจัย

การทดลองที่ 1 เป็นการทดลองเพื่อศึกษาปัจจัยที่ส่งผลต่อประสิทธิภาพการทำงานของ จีเนติกอัลกอริทึมแบบดั้งเดิม โดยการทดลองนี้ได้ทำการเปรียบเทียบการทำงานของ จีเนติกอัลกอริทึมแบบดั้งเดิม (GA) กับจีเนติกอัลกอริทึมที่ได้มีการปรับเปลี่ยนแต่ละส่วน แบ่งเป็น 8 อัลกอริทึม ดังนี้

- 1) จีเนติกอัลกอริทึมแบบดั้งเดิม (GA)
- 2) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟัซซี (GA + Fuzzy Roulette Wheel Selection)
- 3) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการครอสโอเวอร์แบบแบ่งกลุ่ม (GA + Cluster-Crossover operation)
- 4) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการมิวเตชันแบบโลคอล (GA + Mutation Local Search operation)
- 5) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟัซซีและการครอสโอเวอร์แบบแบ่งกลุ่ม (GA + Fuzzy Roulette Wheel Selection) + (GA + Cluster-Crossover operation)
- 6) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟัซซีและการมิวเตชันแบบโลคอล (GA + Fuzzy Roulette Wheel Selection) + (GA + Mutation Local Search operation)
- 7) จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการครอสโอเวอร์แบบแบ่งกลุ่มและการมิวเตชันแบบโลคอล (GA + Cluster-Crossover operation) + (GA + Mutation Local Search operation)
- 8) อัลกอริทึมที่นำเสนอ MGA (Fuzzy Roulette Wheel Selection + Cluster-Crossover operation + Mutation Local Search operation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 การกำหนดค่าพารามิเตอร์

การกำหนดค่าพารามิเตอร์สำหรับทดสอบ 8 อัลกอริทึมที่กล่าวข้างต้น สำหรับเปรียบเทียบประสิทธิภาพของแต่ละอัลกอริทึม โดยกำหนดค่าพารามิเตอร์ให้เหมือนกันทั้งหมด และมีการกำหนดจำนวนประชากรเริ่มต้นเป็นชุดเดียวกันและเพิ่มค่าขึ้นเพื่อให้ครอบคลุมกับขนาดของปัญหาที่ใช้ในการทดสอบดังตารางที่ 4.4

ตารางที่ 4.4 ค่าพารามิเตอร์เปรียบเทียบปัจจัยที่มีผลต่อประสิทธิภาพของอัลกอริทึม

พารามิเตอร์	อัลกอริทึม							
	1	2	3	4	5	6	7	8
จำนวนประชากรเริ่มต้น	50, 100, 150, 200							
ความน่าจะเป็นการครอสโอเวอร์	0.9							
ความน่าจะเป็นการมิวเทชัน	0.5							
เงื่อนไขในการหยุดการทำงาน	ค่าคำตอบไม่เปลี่ยนแปลงจำนวน 80% ของประชากรเริ่มต้น							

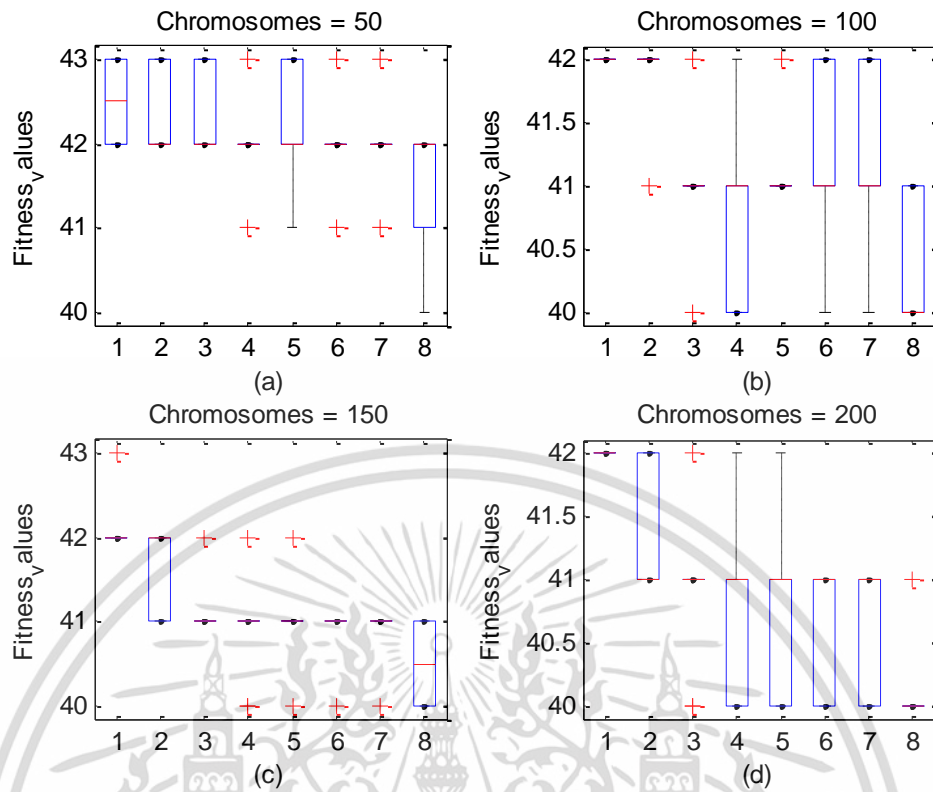
4.3.2 ผลการทดลอง

การทดลองที่ 1 ได้ทำการทดสอบจินตคณิตอัลกอริทึมแบบดั้งเดิม และจินตคณิตอัลกอริทึมที่ปรับเปลี่ยนทั้ง 8 อัลกอริทึม ทดสอบกับข้อมูลมาตรฐานของ Brandimarte's (BRdata) [20] จำนวน 10 ชุด คือ Mk01:Mk10 จากค่าพารามิเตอร์ที่กำหนดในตารางที่ 4.4 แต่ละอัลกอริทึมแต่ละตัวมีการรันจำนวน 10 ครั้งต่อหนึ่งชุดข้อมูล โดยกำหนดจำนวนโครโมโซมเริ่มต้น 50 โครโมโซม มีการรันจำนวน 10 ครั้ง โครโมโซมเริ่มต้น 100 โครโมโซม มีการรันจำนวน 10 ครั้ง โครโมโซมเริ่มต้น 150 โครโมโซม มีการรันจำนวน 10 ครั้ง และ โครโมโซมเริ่มต้น 200 โครโมโซม มีการรันจำนวน 10 ครั้ง ดังนั้น แต่ละอัลกอริทึมจึงมีการรันทั้งหมด 400 ครั้ง (40 ครั้ง x 10 ชุดข้อมูล)

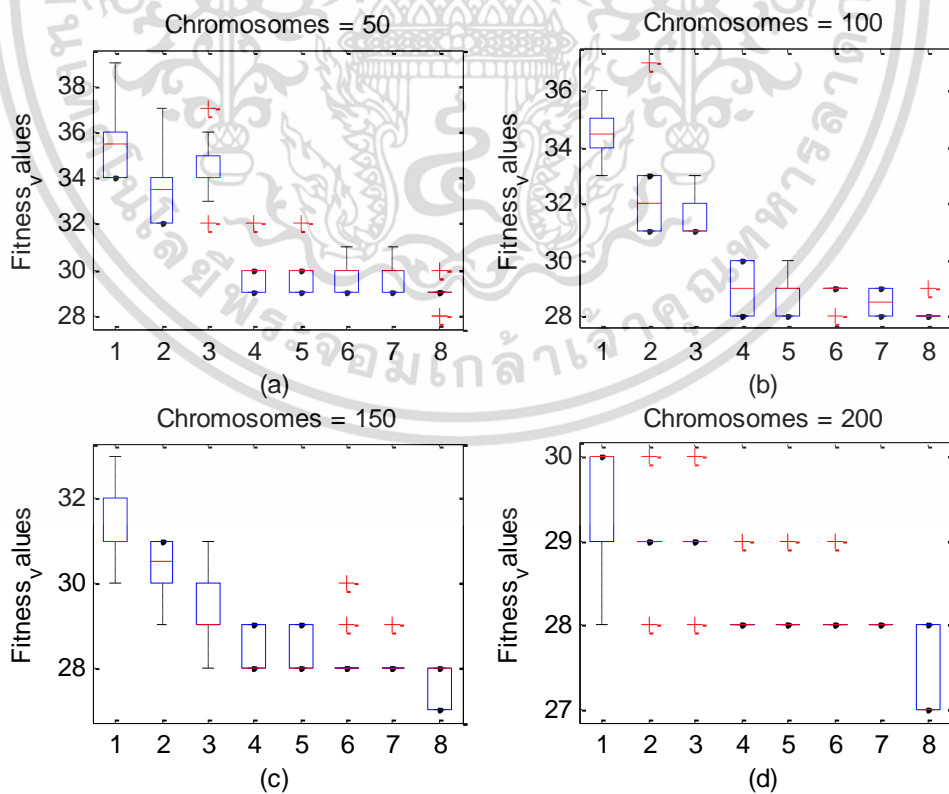
เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบคือ Intel @ core™ i7 2.8 GHZ (4 GB of RAM) และใช้โปรแกรม MATLAB เวอร์ชัน 7.6.0.324 ผลการทดลองที่ได้มีการเปรียบเทียบประสิทธิภาพของทั้งสองอัลกอริทึมด้วย 3 ตัวชี้วัด คือ 1) ค่าคำตอบที่ดีที่สุด (c_{best}) และ 2) ค่าเฉลี่ยของคำตอบทั้งหมด (\bar{c}) ดังรูปที่ 4.1 – 4.10 ของชุดข้อมูลจำนวน 10 ชุดแต่ละชุดแสดงผลการทดลองแบ่งเป็น 4 ส่วนคือ

- 1) ผลการทดลองการรันจำนวน 10 ครั้ง เมื่อกำหนดจำนวนโครโมโซม 50 โครโมโซม
- 2) ผลการทดลองการรันจำนวน 10 ครั้ง เมื่อกำหนดจำนวนโครโมโซม 100 โครโมโซม
- 3) ผลการทดลองการรันจำนวน 10 ครั้ง เมื่อกำหนดจำนวนโครโมโซม 150 โครโมโซม
- 4) ผลการทดลองการรันจำนวน 10 ครั้ง เมื่อกำหนดจำนวนโครโมโซม 200 โครโมโซม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

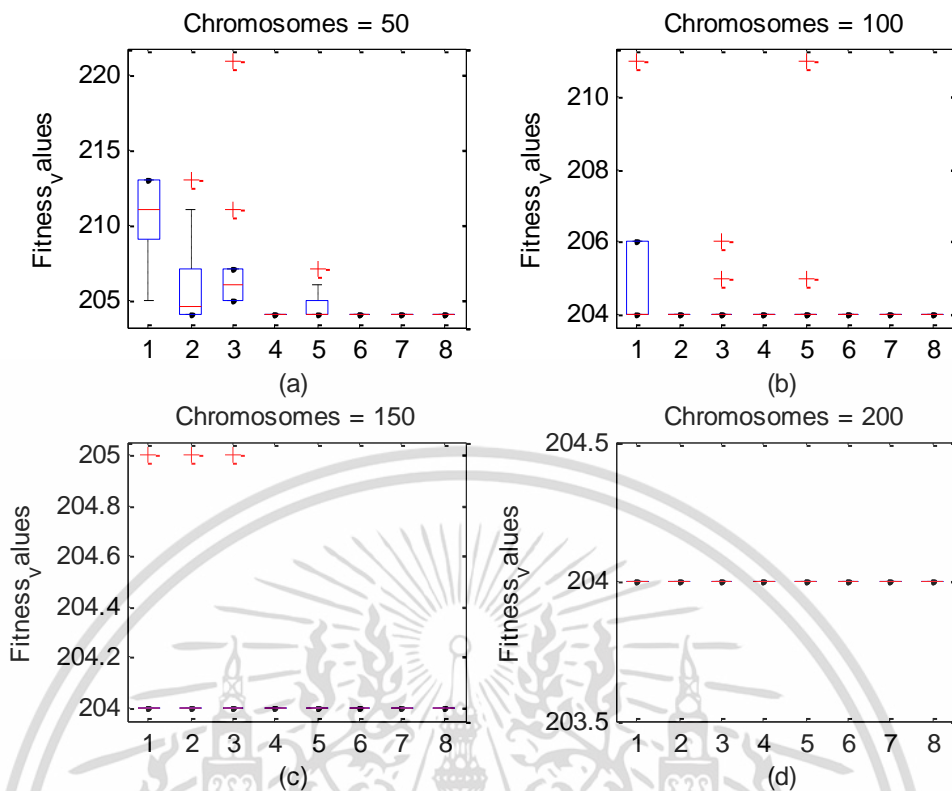


รูปที่ 4.1 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK01

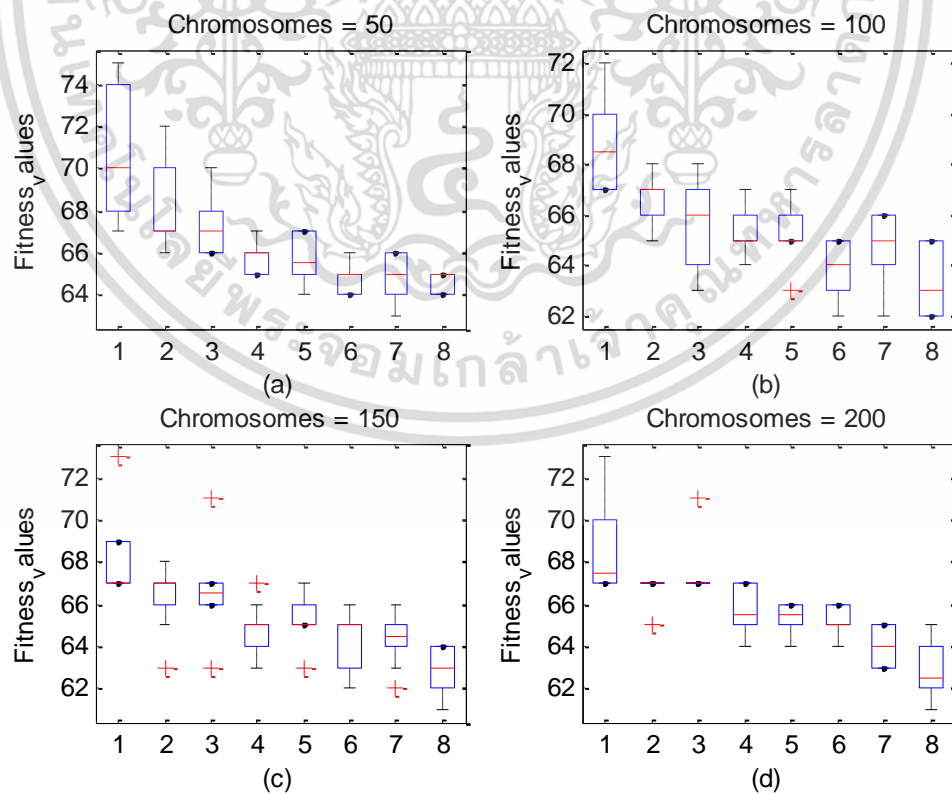


รูปที่ 4.2 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

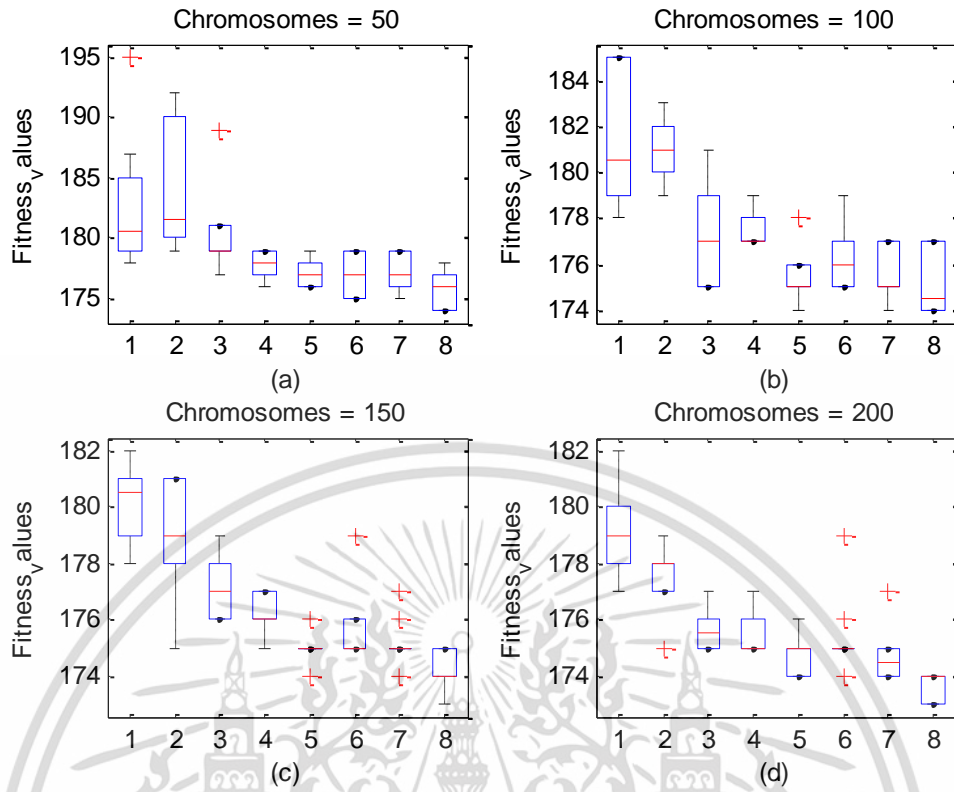


รูปที่ 4.3 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK03

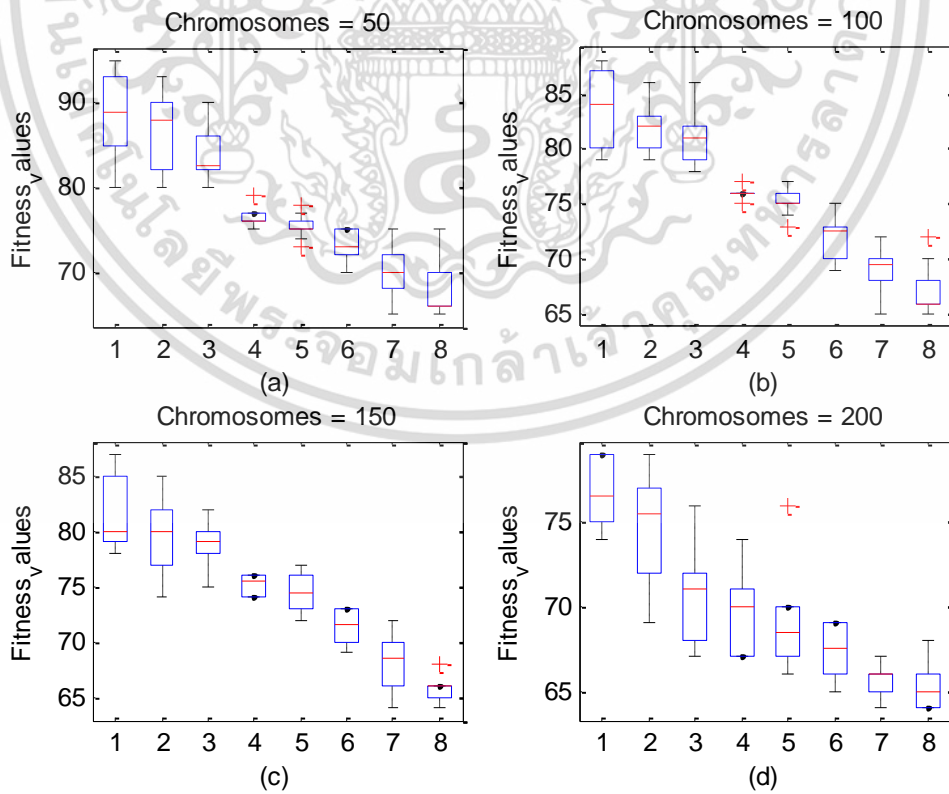


รูปที่ 4.4 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK04

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

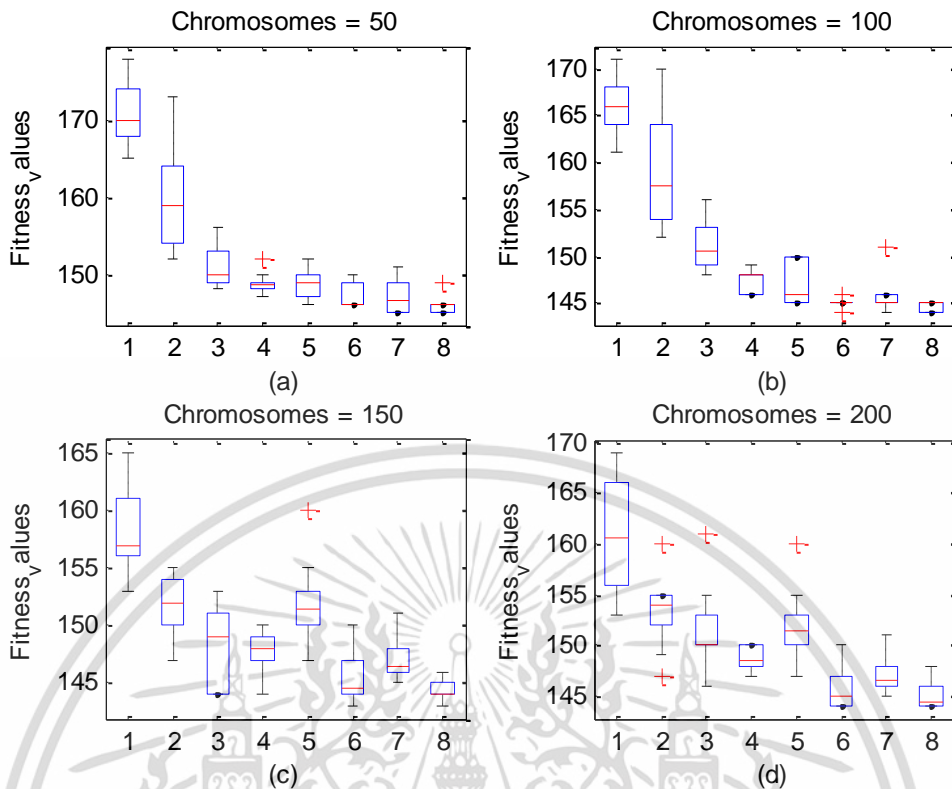


รูปที่ 4.5 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK05

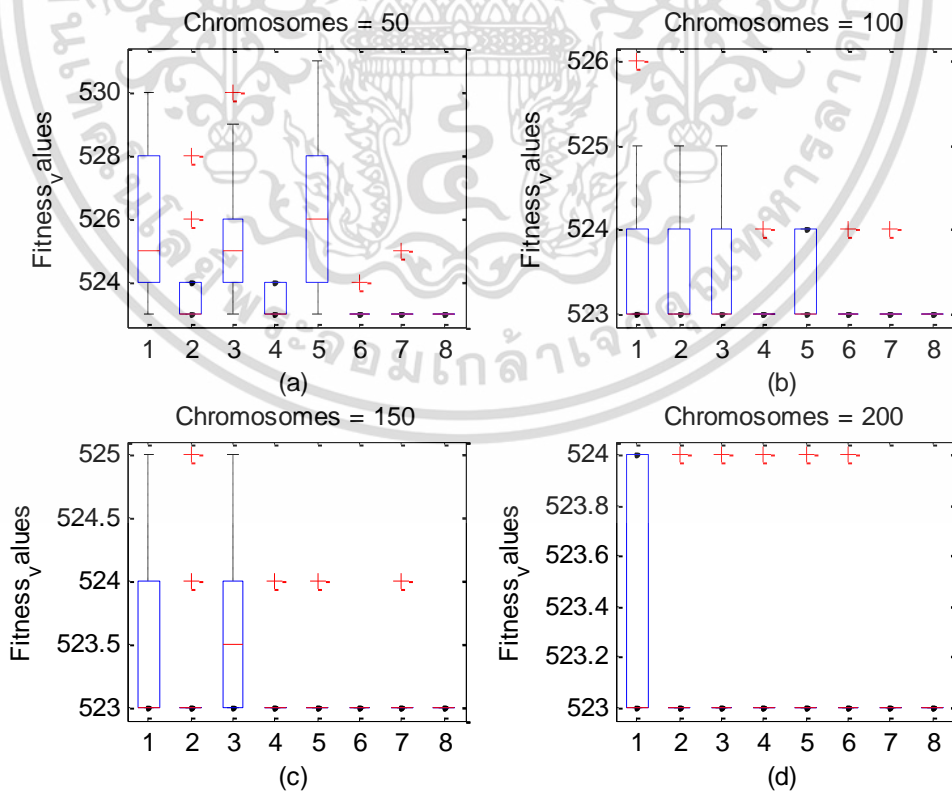


รูปที่ 4.6 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

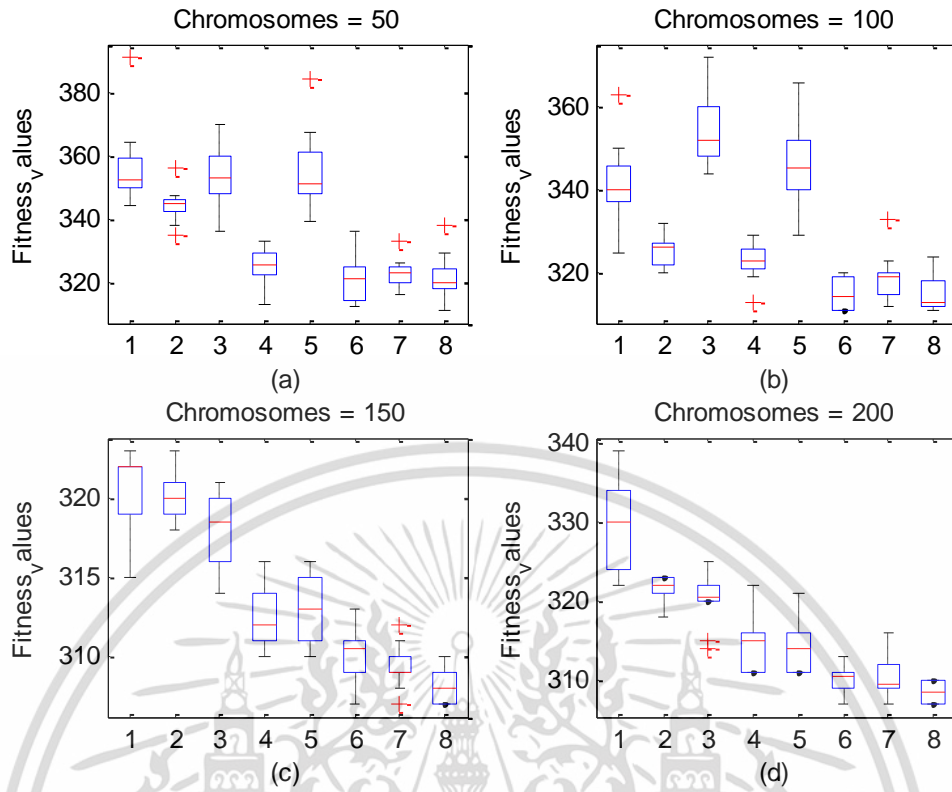


รูปที่ 4.7 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK07

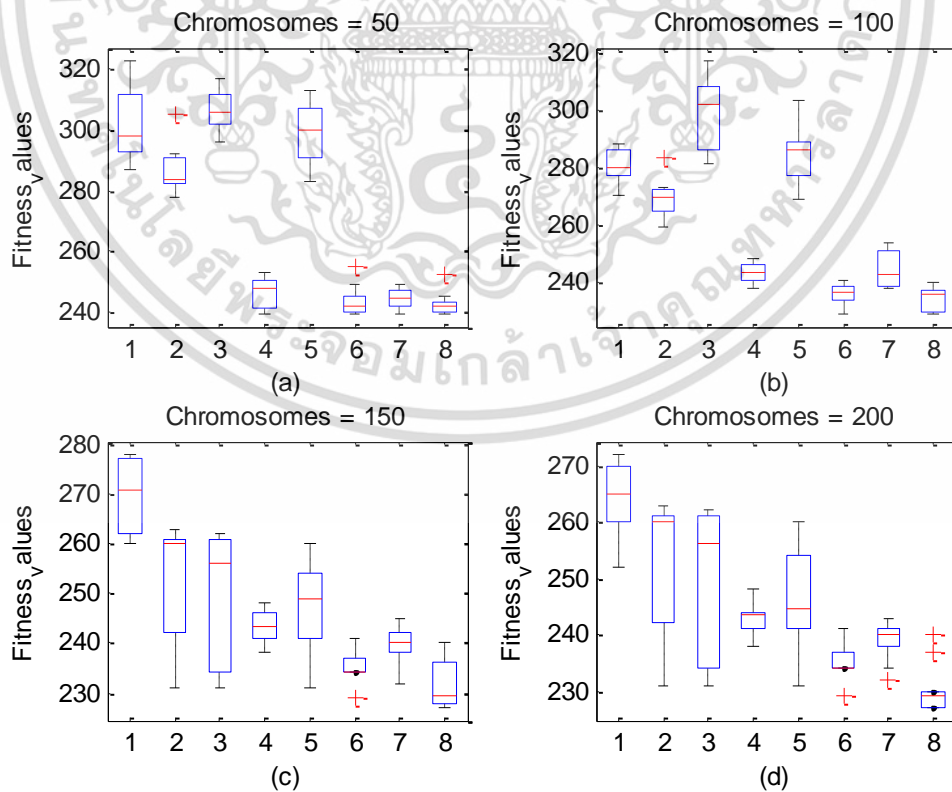


รูปที่ 4.8 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK08

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK09



รูปที่ 4.10 ผลการทดลองการรันจำนวน 10 ครั้ง สำหรับข้อมูล MK10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 สรุปผลการทดลอง

การทดลองที่ 1 ทดสอบปัจจัยที่มีผลต่อประสิทธิภาพของอัลกอริทึมที่นำเสนอ โดยเพิ่มการส่วนการปรับปรุงแต่ละส่วนใน 8 อัลกอริทึมดังรายละเอียดข้างต้น สามารถสรุปได้ดังนี้

1) ชุดข้อมูล MK01

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอ สามารถอธิบายได้ดังนี้

รูปที่ 4.1 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 4, 6 และ 7 มีค่าความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 4, 6 และ 7 มีค่าคำตอบเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นดีสุดเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 4, 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.1 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 6 และ 7 มีค่าคำตอบเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นดีสุดเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.1 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 4, 5, 6 และ 7 มีค่าความแปรปรวนน้อยที่สุด มีค่าคำตอบเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นดีสุดเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 4, 5, 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.1 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 3 มีค่าความแปรปรวนน้อยที่สุด สามารถสรุปว่า อัลกอริทึมที่ 3 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

2) ชุดข้อมูล MK02

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.2 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และมีค่าความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 และ 7 มีค่าคำตอบเฉลี่ยที่เท่ากันเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 และ 7 มีประสิทธิภาพต่อ

อัลกอริทึมที่นำเสนอ

รูปที่ 4.2 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และมีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 7 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.2 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 4, 5, 6, 7 และ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 6 และ 7 มีความแปรปรวนน้อยที่สุด สามารถสรุปว่า อัลกอริทึมที่ 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.2 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 4, 5, 6 และ 7 มีความแปรปรวนน้อยที่สุด มีค่าเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 4, 5, 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

3) ชุดข้อมูล MK03

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.3 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 4, 5, 6, 7 และ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุด (LB) และมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 4, 6, 7 และ 8 มีความแปรปรวนน้อยที่สุด สามารถสรุปว่า อัลกอริทึมที่ 4, 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.3 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า ทุกอัลกอริทึมสามารถค้นพบค่าคำตอบที่ดีที่สุด (LB) และอัลกอริทึมที่ 3, 4, 5, 6, 7 และ 8 มีความแปรปรวนน้อยที่สุด สามารถสรุปว่าชุดข้อมูลดังกล่าวเป็นชุดข้อมูลอย่างง่ายทำให้ทุกอัลกอริทึมสามารถค้นพบค่าคำตอบที่ดีที่สุดได้

รูปที่ 4.3 (c) และ (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 150 และ 200 ตามลำดับ ผลการทดสอบปรากฏว่า ทุกอัลกอริทึมสามารถค้นพบค่าคำตอบที่ดีที่สุด (LB) มีค่าเฉลี่ยของคำตอบดีที่สุด และมีความแปรปรวนน้อยที่สุด สามารถสรุปว่าชุดข้อมูลดังกล่าวเป็นชุดข้อมูลอย่างง่ายทำให้ทุกอัลกอริทึมสามารถค้นพบค่าคำตอบที่ดีที่สุดได้

4) ชุดข้อมูล MK04

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.4 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 6 และ 7 มีค่าเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.4 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด อัลกอริทึมที่ 6 มีค่าเฉลี่ยของคำตอบเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.4 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 7 มีความแปรปรวนน้อยที่สุด และมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.4 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยของคำตอบดีเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

5) ชุดข้อมูล MK05

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.5 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 6 และ 7 มีค่าเฉลี่ยที่เท่ากันเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.5 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 5 มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 5 และ 7 มีค่าเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 5 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.5 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด อัลกอริทึมที่ 5, 6 และ 7 มีค่าเฉลี่ยของคำตอบดีเป็นลำดับถัดมา และอัลกอริทึมที่ 5 และ 7 มีความแปรปรวนน้อยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 5, 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.5 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 7 มีความแปรปรวนน้อยที่สุด และมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

6) ชุดข้อมูล MK06

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.6 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 7 มีค่าเฉลี่ยของคำตอบดีเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.6 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด ส่วนอัลกอริทึมที่ 7 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.6 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 4 มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 7 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.6 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 7 มีความแปรปรวนน้อยที่สุด และมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

7) ชุดข้อมูล MK07

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.7 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 4 มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.7 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และมีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 และ 7 มีค่าเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.7 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และมีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.7 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และมีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

8) ชุดข้อมูล MK08

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.8 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 6, 7 และ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุด (LB) และมีค่าเฉลี่ยของคำตอบดีที่สุด มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 2 และ 4 มีค่าเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 2 และ 4 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.8 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 4, 6, 7 และ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุด (LB) และมีค่าเฉลี่ยของคำตอบดีที่สุด มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 1, 2, 3 และ 5 มีค่าเฉลี่ยที่เท่ากัน สรุปว่า อัลกอริทึมที่ 1, 2, 3 และ 5 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.8 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 6 และ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุด (LB) และมีค่าเฉลี่ยของคำตอบดีที่สุด มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 2, 4, 5 และ 7 มีค่าเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 2, 4, 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.8 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 7 และ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุด (LB) และมีค่าเฉลี่ยของคำตอบดีที่สุด มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 2, 3, 4, 5 และ 6 มีค่าเฉลี่ยที่เท่ากันและมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 2, 3, 4, 5 และ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

9) ชุดข้อมูล MK09

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.9 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 2 มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.9 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 2 มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.9 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 2 มีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 7 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.9 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 7 มีความแปรปรวนน้อยที่สุด และมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 7 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

10) ชุดข้อมูล MK10

ในการทดสอบ 8 อัลกอริทึมตามหัวข้อ 4.3 เพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังนี้

รูปที่ 4.10 (a) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบคือ 50 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และมีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.10 (b) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบ คือ 100 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และอัลกอริทึมที่ 6 มีความแปรปรวนน้อยที่สุด และมีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.10 (c) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบ คือ 150 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และมีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

รูปที่ 4.10 (d) แสดงกราฟผลการทดสอบ โดยกำหนดจำนวนโครโมโซมสำหรับการทดสอบ คือ 200 ผลการทดสอบปรากฏว่า อัลกอริทึมที่นำเสนอคืออัลกอริทึมที่ 8 สามารถค้นพบค่าคำตอบที่ดีที่สุดและมีค่าเฉลี่ยของคำตอบดีที่สุด และมีความแปรปรวนน้อยที่สุด ส่วนอัลกอริทึมที่ 6 มีค่าเฉลี่ยเป็นลำดับถัดมา สามารถสรุปว่า อัลกอริทึมที่ 6 มีประสิทธิภาพต่ออัลกอริทึมที่นำเสนอ

4.4 การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Brandimarte's (BRdata)

การทดลองที่ 2 ได้ทำการทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอโดยทดสอบกับชุดข้อมูลมาตรฐาน Brandimarte's (BRdata) [20] จำนวน 10 ชุด โดยได้เปรียบเทียบกับ 5 อัลกอริทึม ได้แก่ 1) LEGA (LEarnable Genetic Architecture) ที่นำเสนอโดย Ho et al. [37] 2) JSSANT (Job Shop Scheduling with ACO) ที่นำเสนอโดย Girish and Jawaliar [38] 3) PSO (Particle Swarm Optimization) ที่นำเสนอโดย Girish and Jawaliar [39] 4) KBACO (Knowledge-Based Ant Colony Optimization) ที่นำเสนอโดย Xing et al. [14] และ 5) BBO (Biogeography-Based Optimization) ที่นำเสนอ โดย Rahmati และ Zandieh [31] ด้วย

4.4.1 การกำหนดพารามิเตอร์

อัลกอริทึมที่นำเสนอสำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น มีการกำหนดค่าพารามิเตอร์ต่าง ๆ ดังตารางที่ 4.5 สำหรับชุดข้อมูลมาตรฐาน Brandimarte's (BRdata) ดังนี้

ตารางที่ 4.5 ค่าพารามิเตอร์ของอัลกอริทึมที่นำเสนอสำหรับชุดข้อมูล BRdata

พารามิเตอร์	MGA
จำนวนประชากรเริ่มต้น	

- BRdata (MK01:MK10)	100 - 400
ความน่าจะเป็นการครอสโอเวอร์	0.5
ความน่าจะเป็นการมิวเตชัน	0.5
- อุณหภูมิเริ่มต้น	100
- อัตราการลดลงของอุณหภูมิ	0.95
- จุดการเย็นตัว	0.98
เงื่อนไขในการหยุดการทำงาน	ค่าคำตอบไม่เปลี่ยนแปลงจำนวน 80% ของประชากรเริ่มต้น

4.4.2 ผลการทดลอง

การทดลองนี้ได้ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึมอื่นได้แก่ 1) LEGA 2) JSSANT 3) PSO 4) KBACO และ 5) BBO โดยได้ทดสอบกับปัญหาการจัดตารางการผลิตแบบตามสิ่งยึดหยุ่น คือ ข้อมูลชุด BRdata จำนวน 10 ชุดข้อมูลการทดลองเพื่อวัดประสิทธิภาพของอัลกอริทึมที่นำเสนอสำหรับปัญหาออปติไมเซชันได้ด้วยเครื่องคอมพิวเตอร์ Intel® core™ i7 2.8 GHz (4 GB of RAM) และใช้โปรแกรม MATLAB เวอร์ชัน 7.6.0.324 ผลการทดลองที่ได้ทำการเปรียบเทียบประสิทธิภาพของทั้ง 6 อัลกอริทึมด้วยตัวชี้วัดคือ เวลาในการผลิตรวมน้อยสุด (Minimize Makespan) และค่าคลาดเคลื่อนสัมพัทธ์ (Relative Error: RE) ดังตารางที่ 4.6

4.4.3 สรุปผลการทดลอง

การทดลองที่ 2 ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่นำเสนอ MGA กับ 5 อัลกอริทึม ได้แก่ 1) LEGA 2) JSSANT 3) PSO 4) KBACO และ 5) BBO สำหรับข้อมูลชุด BRdata ผลการทดลองสามารถสรุปผลได้ ดังนี้

อัลกอริทึมที่นำเสนอสามารถค้นหาค่าคำตอบที่เหมาะสมที่สุดได้จำนวน 2 ชุดข้อมูล (MK03 และ MK08) เช่นเดียวกับ 5 อัลกอริทึมที่เปรียบเทียบ ส่วนชุดข้อมูลที่เหลือ 8 ชุดข้อมูล (MK01, MK02, MK04, MK05, MK06, MK07, MK09, MK10) อัลกอริทึมที่นำเสนอไม่สามารถค้นหาค่าคำตอบที่เหมาะสมที่สุดได้ แต่สามารถค้นหาค่าคำตอบที่เหมาะสมได้ใกล้เคียงกับค่าคำตอบที่เหมาะสมที่สุด (lower bound) ได้ดีกว่า 5 อัลกอริทึม โดยภาพรวมการค้นหาค่าความเหมาะสมมีค่าคลาดเคลื่อนสัมพัทธ์เฉลี่ยของอัลกอริทึมที่นำเสนอ MGA ต่ำกว่า 5 อัลกอริทึม ได้แก่ 1) LEGA 2) JSSANT 3) PSO 4) KBACO และ 5) BBO เมื่อเปรียบเทียบกันค่าคลาดเคลื่อนสัมพัทธ์เฉลี่ยเท่ากับ 19.54%, 23.57%, 26.08%, 27.24%, 22.17% และ 21.35% ตามลำดับ

ตารางที่ 4.6 การเปรียบเทียบผลการทดลองสำหรับข้อมูล BRdata ระหว่างอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึมอื่น

Problems	n x m	LB	LEGA (2007) Ho et al. [37]	JSSANT (2009) Girish & Jawahar [38]	PSO (2009) Girish & Jawahar [39]	KBACO (2010) Xing et al. [14]	BBO (2012) Rahmati & Zandieh [31]	Proposed model: MGA
MK01	10 x 6	36	40 (11.11)	40 (11.11)	40 (11.11)	39 (8.33)	40 (11.11)	40 (11.11)
MK02	10 x 6	24	29 (20.83)	26 (8.33)	27 (12.50)	29 (20.83)	28 (16.67)	27 (12.50)
MK03	15 x 8	204	204 (0.00)	204 (0.00)	204 (0.00)	204 (0.00)	204 (0.00)	204 (0.00)
MK04	15 x 8	48	67 (39.58)	66 (37.50)	62 (29.17)	65 (35.42)	66 (37.50)	61 (27.08)
MK05	15 x 4	168	176 (4.76)	174 (3.57)	178 (5.95)	173 (2.98)	173 (2.98)	173 (2.98)
MK06	10 x 15	33	67 (103.03)	77 (133.33)	78 (136.36)	67 (103.03)	64 (93.94)	64 (93.94)
MK07	20 x 5	133	147 (10.53)	143 (7.52)	147 (10.53)	144 (8.27)	144 (8.27)	143 (7.52)
MK08	20 x 10	523	523 (0.00)	523 (0.00)	523 (0.00)	523 (0.00)	523 (0.00)	523 (0.00)
MK09	20 x 10	299	320 (7.02)	328 (9.79)	341 (14.05)	311 (4.01)	310 (3.68)	307 (2.68)
MK10	20 x 15	165	229 (38.79)	247 (49.70)	252 (52.73)	229 (38.79)	230 (39.39)	227 (37.58)
Mean Relative Error			23.57	26.08	27.24	22.17	21.35	19.54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Fattahi et al.(Fdata)

การทดลองที่ 3 ได้ทำการทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอโดยทดสอบกับชุดข้อมูลมาตรฐาน Fattahi et al.(Fdata) [10] จำนวน 20 ชุด โดยได้เปรียบเทียบกับ 4 อัลกอริทึม ได้แก่ 1) HTS/SA (Hierarchical approach and Simulated Annealing) ที่นำเสนอโดย Fattahi et al. [10] 2) SA(Simulated Annealing) ที่นำเสนอโดย Yazdani et al.[7] 3) AIA(Artificial Immune Algorithm) ที่นำเสนอโดย Bagheri et al.[16] และ 4) Mathematical model ที่นำเสนอโดย Demir & Isleyen [40] ด้วย

4.5.1 การกำหนดพารามิเตอร์

อัลกอริทึมที่นำเสนอสำหรับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น มีการกำหนดค่าพารามิเตอร์ต่าง ๆ ดังตารางที่ 4.7 สำหรับชุดข้อมูลมาตรฐาน Fdata ดังนี้

ตารางที่ 4.7 ค่าพารามิเตอร์ของอัลกอริทึมที่นำเสนอสำหรับชุดข้อมูล Fdata

พารามิเตอร์	MGA
จำนวนประชากรเริ่มต้น	
- Fdata (SFJS1:10)	100 - 400
- Fdata (MFJS1:10)	
ความน่าจะเป็นการครอสโอเวอร์	0.5
ความน่าจะเป็นการมิวเตชัน	0.5
- อุณหภูมิเริ่มต้น	100
- อัตราการลดลงของอุณหภูมิ	0.95
- จุดการเย็นตัว	0.98
เงื่อนไขในการหยุดการทำงาน	ค่าคำตอบไม่เปลี่ยนแปลงจำนวน 80% ของประชากรเริ่มต้น

4.5.2 ผลการทดลอง

การทดลองนี้ได้ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึมอื่น ได้แก่ 1) HTS/SA 2) SA และ 3) AIA โดยได้ทดสอบกับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น คือ ข้อมูลชุด Fdata จำนวน 20 ชุด (SFJS1:10 และ MFJS1:10) และเปรียบเทียบระหว่างอัลกอริทึมที่นำเสนอกับอัลกอริทึม Mathematical model สำหรับ 17 ชุดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(SFJS1:10 และ MFJS1:7) การทดลองเพื่อวัดประสิทธิภาพของอัลกอริทึมที่นำเสนอสำหรับปัญหา ออกพติไมเซชันได้ด้วยเครื่องคอมพิวเตอร์ Intel ® core™ i7 2.8 GHz (4 GB of RAM) และใช้ โปรแกรม MATLAB เวอร์ชัน 7.6.0.324 ผลการทดลองที่ได้ทำการเปรียบเทียบประสิทธิภาพของ ทั้ง 4 อัลกอริทึมด้วยตัวชี้วัด คือ เวลาในการผลิตรวมน้อยสุด (Minimize Makespan) และค่า คลาดเคลื่อนสัมพัทธ์ (relative error: RE) ดังตารางที่ 4.8

ตารางที่ 4.8 การเปรียบเทียบผลการทดลองสำหรับข้อมูล Fdata ระหว่างอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึมอื่น

Problems	n x m	LB	HTS/SA (2007) Fatahi et al. [10]	SA (2009) Yazdani et al. [7]	AIA (2010) Bagheri et al. [16]	Mathematical model (2013) Demir & İşleyen [40]	Proposed model: MGA
SFJS1	2 x 2	66	66 (0.00)	66 (0.00)	66 (0.00)	66 (0.00)	66 (0.00)
SFJS2	2 x 2	107	107 (0.00)	107 (0.00)	107 (0.00)	107 (0.00)	107 (0.00)
SFJS3	3 x 2	221	221 (0.00)	221 (0.00)	221 (0.00)	221 (0.00)	221 (0.00)
SFJS4	3 x 2	355	355 (0.00)	355 (0.00)	355 (0.00)	355 (0.00)	355 (0.00)
SFJS5	3 x 2	119	119 (0.00)	119 (0.00)	119 (0.00)	119 (0.00)	119 (0.00)
SFJS6	3 x 3	320	320 (0.00)	320 (0.00)	320 (0.00)	320 (0.00)	320 (0.00)
SFJS7	3 x 5	397	397 (0.00)	397 (0.00)	397 (0.00)	397 (0.00)	397 (0.00)
SFJS8	3 x 4	253	256 (1.19)	253 (0.00)	253 (0.00)	253 (0.00)	253 (0.00)
SFJS9	3 x 3	210	210 (0.00)	210 (0.00)	210 (0.00)	210 (0.00)	210 (0.00)
SFJS10	4 x 5	516	516 (0.00)	516 (0.00)	516 (0.00)	516 (0.00)	516 (0.00)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 การเปรียบเทียบผลการทดลองสำหรับข้อมูล Fdata ระหว่างอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึมอื่น (ต่อ)

Problems	n x m	LB	HTS/SA (2007) Fattahi et al. [10]	SA (2009) Yazdani et, al. [7]	AIA (2010) Bagheri et, al. [16]	Mathematical model (2013) Demir & İsteyen [40]	Proposed model: MGA
MFJS2	5 x 7	396	468 (18.18)	448 (13.13)	448 (13.13)	446 (12.63)	446 (12.63)
MFJS3	6 x 7	396	538 (35.86)	468 (18.18)	468 (18.18)	466 (17.68)	466 (17.68)
MFJS4	7 x 7	496	618 (24.60)	561 (13.1)	554 (11.69)	564 (11.69)	554 (11.69)
MFJS5	7 x 7	414	625 (50.97)	514 (24.15)	527 (27.29)	514 (24.15)	514 (24.15)
MFJS6	8 x 7	469	730 (55.65)	634 (35.18)	635 (35.39)	634 (35.18)	634 (35.18)
MFJS7	8 x 7	619	947 (52.99)	899 (45.23)	879 (42.00)	928 (49.92)	881 (42.33)
MFJS8	9 x 8	619	922 (48.95)	897 (44.91)	884 (42.81)	-	884 (42.81)
MFJS9	11 x 8	764	1105 (44.63)	1101 (44.11)	1088 (42.41)	-	1095 (43.32)
MFJS10	12 x 8	944	1384 (46.61)	1258 (33.26)	1267 (34.22)	-	1246 (31.99)
Mean Relative Error			19.90 -	14.47 -	14.27 -	- 10.11 *	13.99 9.52*

4.5.3 สรุปผลการทดลอง

การทดลองที่ 3 ได้ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่นำเสนอ MGA กับ 4 อัลกอริทึม ได้แก่ 1) HTS/SA 2) SA 3) AIA และ 4) Mathematical model สำหรับข้อมูลชุด Fdata ผลการทดลองสามารถสรุปผลได้ ดังนี้

สำหรับชุดข้อมูล Fdata 20 ชุดข้อมูล(SFJS1:10 และ MFJS1:10) เปรียบเทียบระหว่าง อัลกอริทึมที่นำเสนอ MGA กับ 3 อัลกอริทึม ได้แก่ 1) HTS/SA 2) SA และ 3) AIA พบว่า เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึมที่นำเสนอสามารถค้นหาค่าคำตอบที่เหมาะสมได้ใกล้เคียงกับค่าคำตอบที่เหมาะสมที่สุด (lower bound) ได้ดีกว่า 3 อัลกอริทึม โดยค่าคลาดเคลื่อนสัมพัทธ์เฉลี่ยของอัลกอริทึมที่นำเสนอ MGA ต่ำกว่า 3 อัลกอริทึมซึ่งมีค่าเท่ากับ 13.99%, 19.90%, 14.47 และ 14.27% ตามลำดับ

สำหรับชุดข้อมูล Fdata จำนวน 17 ชุดข้อมูล (SFJS1:10 และ MFJS1:7) เปรียบเทียบระหว่างอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึม Mathematical model พบว่าอัลกอริทึมที่นำเสนอ MGA มีค่าคลาดเคลื่อนสัมพัทธ์เฉลี่ยต่ำกว่าอัลกอริทึม Mathematical model ซึ่งมีค่าเท่ากับ 9.52% และ 10.11% ตามลำดับ

4.6 การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Barnes and Chamber library (BCdata) โดยเปรียบเทียบกับผลการทดลองในงานวิจัยของ Demir และ İşleyen

การทดลองที่ 4 ได้ทำการทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA เทียบกับอัลกอริทึม BBO ที่นำเสนอโดย Demir & İşleyen [40] ซึ่งเป็นอัลกอริทึมที่เกิดจากการนำเอาอัลกอริทึม Biogeography-Based Optimization มาแก้ปัญหา FJSP

4.6.1 การกำหนดพารามิเตอร์

อัลกอริทึมที่นำเสนอสำหรับปัญหาการจัดการวางแผนการผลิตแบบตามสั่งยืดหยุ่น มีการกำหนดค่าพารามิเตอร์ต่าง ๆ ดังตารางที่ 4.9 สำหรับชุดข้อมูลมาตรฐาน BCdata ดังนี้

ตารางที่ 4.9 ค่าพารามิเตอร์ของอัลกอริทึมที่นำเสนอสำหรับชุดข้อมูล BCdata

พารามิเตอร์	MGA
จำนวนประชากรเริ่มต้น	
- BCdata	100 - 400
ความน่าจะเป็นการครอสโอเวอร์	0.5
ความน่าจะเป็นการมิวเตชัน	0.5
- อุณหภูมิเริ่มต้น	100
- อัตราการลดลงของอุณหภูมิ	0.95
- จุดการเย็นตัว	0.98
เงื่อนไขในการหยุดการทำงาน	ค่าคำตอบไม่เปลี่ยนแปลงจำนวน 80% ของประชากรเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2 ผลการทดลอง

การทดลองนี้ได้ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึม BBO โดยได้ทดสอบกับปัญหาการจัดตารางการผลิตแบบตามสั่งยืดหยุ่น คือ ข้อมูลชุด BCdata จำนวน 21 ชุดข้อมูลการทดลองเพื่อวัดประสิทธิภาพของอัลกอริทึมที่นำเสนอสำหรับปัญหาออปติไมเซชันได้ด้วยเครื่องคอมพิวเตอร์ Intel® core™ i7 2.8 GHz (4 GB of RAM) และใช้โปรแกรม MATLAB เวอร์ชัน 7.6.0.324 ผลการทดลองที่ได้ทำการเปรียบเทียบประสิทธิภาพของทั้ง 2 อัลกอริทึมด้วยตัวชี้วัด คือเวลาในการผลิตรวมที่น้อยสุด (Minimize Makespan) และค่าคลาดเคลื่อนสัมพัทธ์ (Relative Error: RE) ดังตารางที่ 4.10

ตารางที่ 4.10 การเปรียบเทียบผลการทดลองของอัลกอริทึม MGA กับอัลกอริทึม BBO

ลำดับ	Problems	n x m	LB	BBO (2012) Rahmati & Zandieh [40]	Proposed model: MGA
1	mt10c1	10 x 11	655	946 (44.43)	927 (41.53)
2	mt10cc	10 x 12	655	946 (44.43)	911 (39.08)
3	mt10x	10 x 11	655	955 (45.80)	924 (41.07)
4	mt10xx	10 x 12	655	939 (43.36)	934 (42.60)
5	mt10xxx	10 x 13	655	954 (45.65)	918 (40.15)
6	mt10xy	10 x 12	655	951 (45.19)	913 (39.39)
7	mt10xyz	10 x 13	655	858 (30.99)	858 (30.99)
8	setb4c9	15 x 11	857	959 (11.90)	940 (9.68)
9	setb4cc	15 x 12	857	944 (10.15)	942 (9.80)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 การเปรียบเทียบผลการทดลองของอัลกอริทึม MGA กับอัลกอริทึม BBO (ต่อ)

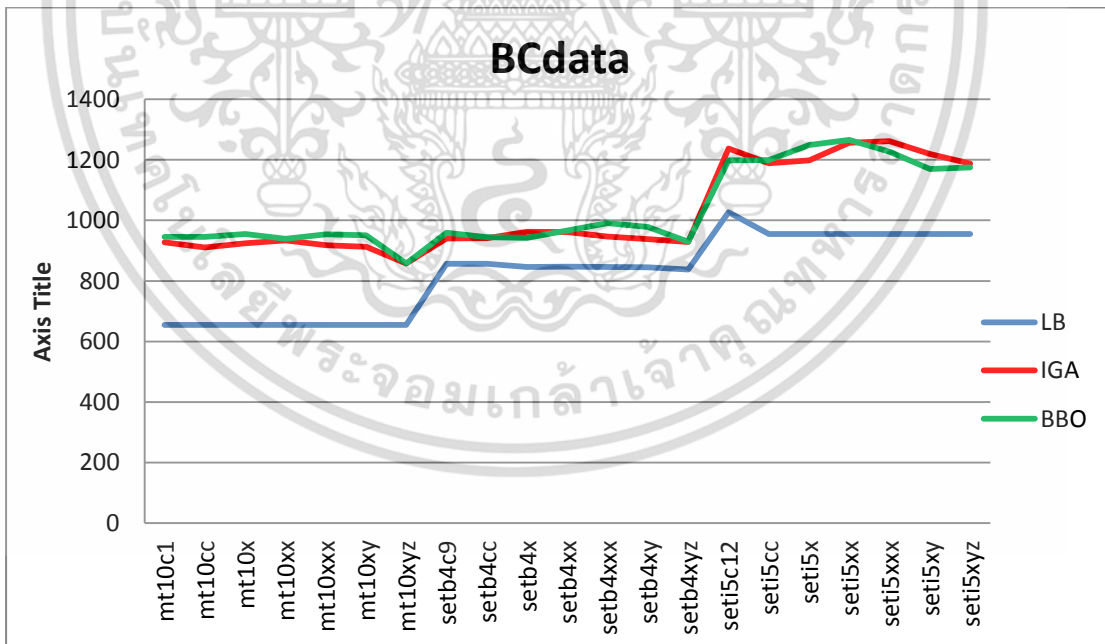
ลำดับ	Problems	n x m	LB	BBO (2012) Rahmati & Zandieh	Proposed model: MGA
10	setb4x	15 x 11	846	942 (11.35)	962 (13.71)
11	setb4xx	15 x 12	847	967 (14.17)	961 (13.46)
12	setb4xxx	15 x 13	846	991 (17.14)	947 (11.94)
13	setb4xy	15 x 12	845	978 (15.74)	938 (11.01)
14	setb4xyz	15 x 13	838	930 (10.98)	929 (10.86)
15	seti5c12	15 x 16	1027	1198 (16.65)	1237 (20.45)
16	seti5cc	15 x 17	955	1199 (25.55)	1189 (24.50)
17	seti5x	15 x 16	955	1249 (30.79)	1198 (25.45)
18	seti5xx	15 x 17	955	1266 (32.57)	1257 (31.62)
19	seti5xxx	15 x 18	955	1266 (32.57)	1262 (32.15)
20	seti5xy	15 x 17	955	1227 (28.48)	1220 (27.75)
21	seti5xyz	15 x 18	955	1175 (23.04)	1180 (24.40)
Mean Relative Error				21.60	21.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.3 สรุปผลการทดลอง

การทดลองที่ 4 ได้ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่นำเสนอ MGA กับอัลกอริทึม BBO ที่นำเสนอโดย Demir & İşleyen สำหรับข้อมูลชุด BCdata ผลการทดลองสามารถสรุปผลได้ ดังนี้

เมื่อเปรียบเทียบ 2 อัลกอริทึมพบว่าอัลกอริทึมที่นำเสนอ MGA สามารถค้นหาค่าความเหมาะสมได้ดีกว่า อัลกอริทึม BBO จำนวน 17 ชุดข้อมูล สามารถค้นหาค่าความเหมาะสมได้เท่ากับอัลกอริทึม BBO จำนวน 1 ชุดข้อมูล คือ 7) mt10xyz และอัลกอริทึม BBO สามารถค้นหาค่าความเหมาะสมได้ดีกว่าอัลกอริทึมที่นำเสนอ MGA จำนวน 3 ชุดข้อมูลคือ 10) setb4x1 5) seti5c12 และ 21) seti5xyz ดังนั้นในภาพรวมแล้วอัลกอริทึมที่นำเสนอ MGA สามารถค้นหาค่าคำตอบที่เหมาะสมได้ใกล้เคียงกับค่าคำตอบที่เหมาะสมที่สุด (lower bound) ได้ดีกว่าอัลกอริทึม BBO โดยมีค่าคลาดเคลื่อนสัมพัทธ์เฉลี่ยต่ำกว่าซึ่งอัลกอริทึมที่นำเสนอ MGA มีค่าคลาดเคลื่อนสัมพัทธ์เท่ากับ 21.23% ขณะที่อัลกอริทึม BBO ที่เสนอโดย Demir & İşleyen มีค่าคลาดเคลื่อนสัมพัทธ์เท่ากับ 21.60% กราฟรูปที่ 4.11 แสดงค่าความเหมาะสมที่ของอัลกอริทึม MGA และ BBO เทียบกับค่าความเหมาะสมที่สุด (LB) ของแต่ละชุดข้อมูลจะเห็นได้ว่าค่าความเหมาะสมของอัลกอริทึม MGA ใกล้เคียงค่าความเหมาะสมที่สุดมากกว่าอัลกอริทึม BBO



รูปที่ 4.11 กราฟแสดงการเปรียบเทียบผลการทดลองที่ 6 ระหว่างอัลกอริทึม MGA และอัลกอริทึม BBO สำหรับชุดข้อมูล BCdata

4.7 สรุปผลการทดสอบอัลกอริทึมที่นำเสนอ

1) สรุปผลการทดสอบเพื่อศึกษาปัจจัยการที่ส่งผลต่อประสิทธิภาพการทำงานของจีเนติกอัลกอริทึมแบบดั้งเดิมที่ทำการปรับปรุง 3 ส่วนและได้แยกส่วนทดสอบโดยแบ่งเป็น 8 อัลกอริทึมดังนี้

- a. จีเนติกอัลกอริทึมแบบดั้งเดิม (GA)
- b. จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟัซซี (GA + Fuzzy Roulette Wheel Selection)
- c. จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการครอสโอเวอร์แบบแบ่งกลุ่ม (GA + Cluster-Crossover operation)
- d. จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการมิวเตชันแบบโลคอล (GA + Mutation Local Search operation)
- e. จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟัซซีและการครอสโอเวอร์แบบแบ่งกลุ่ม (GA + Fuzzy Roulette Wheel Selection) + (GA + Cluster-Crossover operation)
- f. จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟัซซีและการมิวเตชันแบบโลคอล (GA + Fuzzy Roulette Wheel Selection) + (GA + Mutation Local Search operation)
- g. จีเนติกอัลกอริทึมแบบดั้งเดิมที่ปรับเปลี่ยนการครอสโอเวอร์แบบแบ่งกลุ่มและการมิวเตชันแบบโลคอล (GA + Cluster-Crossover operation) + (GA + Mutation Local Search operation)
- h. อัลกอริทึมที่นำเสนอ MGA (Fuzzy Roulette Wheel Selection + Cluster-Crossover operation + Mutation Local Search operation)

จีเนติกอัลกอริทึมแบบดั้งเดิมที่มีการปรับปรุง ได้แยกส่วนทดสอบโดยแบ่งเป็น 8 อัลกอริทึม พบว่าทั้ง 7 อัลกอริทึมที่ได้ปรับปรุงจากจีเนติกอัลกอริทึมแบบดั้งเดิมสามารถค้นหาคำตอบที่เหมาะสมได้ใกล้เคียงกับค่าคำตอบที่เหมาะสมที่สุดของแต่ละชุดข้อมูลได้ดีกว่า จากผลการทดลองที่ 1 นำค่าคำตอบที่ดีที่สุดของแต่ละอัลกอริทึมมาเปรียบเทียบค่าคลาดเคลื่อนสัมพัทธ์ ดังตารางที่ 4.11 สรุปได้ว่า ปัจจัยการคัดเลือกโครโมโซมโดยคัดเลือกด้วยวงล้อรูเล็ตแบบฟัซซี การครอสโอเวอร์แบบแบ่งกลุ่ม และการมิวเตชันแบบโลคอลทำให้ประสิทธิภาพในการค้นหาคำตอบที่เหมาะสมที่สุดดีขึ้น โดยมีค่าคลาดเคลื่อนสัมพัทธ์ต่ำกว่าจีเนติกอัลกอริทึมแบบดั้งเดิม

ได้แก่ 27.56%, 23.61%, 22.93%, 22.68%, 22.19%, 21.44%, 21.70% และ 20.12% ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 ผลการทดลองเปรียบเทียบค่าคลาดเคลื่อนสัมพัทธ์ของ 8 อัลกอริทึม

ชุดข้อมูล	LB	อัลกอริทึม							
		1	2	3	4	5	6	7	8
Mk01	36	42	41	40	40	40	40	40	40
Ml02	24	28	28	28	28	28	28	28	27
Mk03	204	204	204	204	204	204	204	204	204
Mk04	48	67	65	67	64	64	64	64	62
Mk05	168	177	175	175	175	174	174	174	173
Mk06	33	74	69	67	67	66	65	64	64
Mk07	133	153	147	146	147	147	144	145	144
Mk08	523	523	523	523	523	523	523	523	523
Mk09	299	315	318	314	310	310	307	307	307
Mk10	165	252	231	231	238	231	229	232	227
Mean Relative Error		27.56%	23.61%	22.93%	22.68%	22.19%	21.44%	21.70%	20.12%

2) สรุปการทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Brandimarte's (BRdata) โดยเปรียบเทียบกับอัลกอริทึมอื่น

การทดลองที่ 2 ได้ทำการทดสอบอัลกอริทึมที่นำเสนอซึ่งเป็นการรวมการทำงานของ 3 ปัจจัย คือ 1) การคัดเลือกโครโมโซมต้นแบบวงล้อรูเล็กแบบพีชชี 2) การครอสโอเวอร์แบบแบ่งกลุ่มและ 3) การมิวเตชันแบบโลคอลกับ 5 อัลกอริทึมที่นำมาเปรียบเทียบ ได้แก่ 1) LEGA 2) JSSANT 3) PSO 4) KBACO และ 5) BBO โดยทดสอบกับชุดข้อมูล BRdata ผลการทดสอบ พบว่าอัลกอริทึมที่นำเสนอสามารถค้นหาค่าคำตอบที่เหมาะสมได้ดีกว่า 5 อัลกอริทึมที่นำมาเปรียบเทียบ กล่าวคือมีค่าคลาดเคลื่อนต่ำกว่า 5 อัลกอริทึม 4.03% 6.54% 7.7% 2.63% และ 1.81% ตามลำดับ จากผลการทดลองนี้ สรุปได้ว่าอัลกอริทึมที่นำเสนอมีประสิทธิภาพในการค้นหาค่าคำตอบที่เหมาะสมที่สุดได้ดีกว่า 5 อัลกอริทึม ได้แก่ 1) LEGA 2) JSSANT 3) PSO 4) KBACO และ 5) BBO

3) สรุปการทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Fattahi et al. (Fdata) โดยเปรียบเทียบกับอัลกอริทึมอื่น

การทดลองที่ 3 ได้ทำการทดสอบอัลกอริทึมที่นำเสนอ ซึ่งเป็นการรวมการทำงานของ 3 ปัจจัย คือ 1) การคัดเลือกโครโมโซมต้นแบบวงล้อรูปสี่เหลี่ยมแบบพีชชี 2) การครอสโอเวอร์แบบแบ่งกลุ่ม และ 3) การมีเวชันแบบโลกคอลลกับ 3 อัลกอริทึมที่นำมาเปรียบเทียบ ได้แก่ 1) HTS/SA 2) SA และ 3) AIA โดยทดสอบกับชุดข้อมูล Fdata จำนวน 20 ข้อมูล (SFJS1:10 และ MFJS1:10) และชุดข้อมูล Fdata จำนวน 17 ข้อมูล (SFJS1:10 และ MFJS1:7) กับอัลกอริทึม Mathematical model ผลการทดสอบพบว่าอัลกอริทึมที่นำเสนอสามารถค้นหาคำคำตอบที่เหมาะสมได้ดีกว่า 4 อัลกอริทึมที่นำมาเปรียบเทียบ กล่าวคือมีค่าคลาดเคลื่อนต่ำกว่า 4 อัลกอริทึม 5.91% 0.48% 0.28% และ 0.59% ตามลำดับ จากผลการทดลองนี้ สรุปได้ว่าอัลกอริทึมที่นำเสนอมีประสิทธิภาพในการค้นหาคำตอบที่เหมาะสมที่สุดได้ดีกว่า 4 อัลกอริทึม ได้แก่ 1) HTS/SA 2) SA 3) AIA และ 4) Mathematical model

4) สรุปการทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Barnes and Chamber library (BCdata) โดยเปรียบเทียบกับอัลกอริทึม BBO

การทดลองที่ 4 ได้ทำการทดสอบอัลกอริทึมที่นำเสนอ ซึ่งเป็นการรวมการทำงานของ 3 ปัจจัย คือ 1) การคัดเลือกโครโมโซมต้นแบบวงล้อรูปสี่เหลี่ยมแบบพีชชี 2) การครอสโอเวอร์แบบแบ่งกลุ่ม และ 3) การมีเวชันแบบโลกคอลลกับอัลกอริทึม BBO โดยทดสอบกับชุดข้อมูล BCdata ผลการทดสอบ พบว่า อัลกอริทึมที่นำเสนอสามารถค้นหาคำคำตอบที่เหมาะสมได้ดีกว่า อัลกอริทึม BBO กล่าวคือมีค่าคลาดเคลื่อนต่ำกว่า 0.27% จากผลการทดลองนี้ สรุปได้ว่าอัลกอริทึมที่นำเสนอมีประสิทธิภาพในการค้นหาคำตอบที่เหมาะสมที่สุดได้ดีกว่าอัลกอริทึม BBO

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปงานวิจัย

การศึกษาวิจัยในครั้งนี้ได้พัฒนาอัลกอริทึมที่ปรับปรุงขั้นตอนการทำงานของจินตริก อัลกอริทึมแบบดั้งเดิม โดยทำการปรับปรุง 3 ส่วนคือ 1) วิธีการคัดเลือกโครโมโซมด้วยวงล้อรูเล็ตแบบฟิชซีโดยพิจารณาจากค่าความคล้ายระหว่างโครโมโซมทำให้โครโมโซมมีความสัมพันธ์กัน ซึ่งแตกต่างจากวงล้อรูเล็ตแบบดั้งเดิมที่แต่ละโครโมโซมแบ่งแยกกันไม่มีความสัมพันธ์กัน 2) วิธีการครอสโอเวอร์แบบแบ่งกลุ่ม เป็นการแบ่งกลุ่มโครโมโซมเป็น 2 กลุ่มก่อนแล้วจึงทำการจับคู่ผสมพันธุ์เพื่อหลีกเลี่ยงการลู่เข้าหาคำตอบเร็วเกินไปเนื่องจากการจับคู่ผสมพันธุ์จากคู่โครโมโซมที่เหมือนกันและให้กำเนิดทายาทที่ไม่แตกต่างจากโครโมโซมต้นแบบทำให้ไม่เกิดความหลากหลายของประชากร และ 3) วิธีการมิวเทชันแบบโลคอลทำให้เกิดความหลากหลายของโครโมโซม เป็นการหลีกเลี่ยงการลู่เข้าหาคำตอบที่เป็นแบบเฉพาะพื้นที่ที่ดั่งนั้นการปรับปรุงโครโมโซมด้วยวิธีการมิวเทชันแบบโลคอลทำโดยการแบ่งกลุ่มโครโมโซมทั้งหมดจากค่าฟิตเนส จากนั้นเลือกวิธีการมิวเทชันที่เหมาะสมกับแต่ละโครโมโซม

หลักการงาน โดยย่อของอัลกอริทึมที่นำเสนอเริ่มต้นจากการกำหนดค่าพารามิเตอร์และประชากรเริ่มต้นและประเมินค่าฟิตเนสของประชากรเริ่มต้นโดยจะเรียกว่าเป็นโครโมโซม จากนั้นทำการคัดเลือกโครโมโซมเพื่อเป็นโครโมโซมต้นแบบโดยใช้การคัดเลือกด้วยวงล้อรูเล็ตแบบฟิชซี โดยการหาค่าความเป็นสมาชิกของฟิชซีเซตจากฟังก์ชันสามเหลี่ยมและสี่เหลี่ยม การวางตำแหน่งของโครโมโซมในสล็อตวงล้อรูเล็ตพิจารณาจากค่าความคล้ายของแต่ละโครโมโซมซึ่งกำหนดให้โครโมโซมตัวแรกที่ของรุ่นเป็นตัวเริ่มต้นที่ในตำแหน่งแรกในสล็อต จากนั้นทำการพิจารณาโครโมโซมที่เหลือว่ามีรายละเอียดของยีนที่มีความหมายคล้ายกับโครโมโซมตัวอยู่ในสล็อตโครโมโซมที่มีความคล้ายสูงสุดจะถูกเลือกมาวางในสล็อตเป็นลำดับถัดมาและสามารถซ้อนทับกับโครโมโซมแรกด้วยค่าความคล้ายระหว่างคู่ของโครโมโซมนั้นๆ ทำขั้นตอนดังกล่าวจนครบทุกโครโมโซม การเลือกโครโมโซมใช้วิธีการสุ่มเลือก การเลือกจะทำจนครบจำนวนโครโมโซมที่กำหนดไว้ จากนั้นนำโครโมโซมมาทำการแบ่งกลุ่มด้วยหลักการของการแบ่งกลุ่มเป็นลำดับชั้นจากค่าความคล้ายของโครโมโซม เพื่อไม่ให้โครโมโซมเหมือนกันทำการจับคู่กันเพื่อเพิ่มความหลากหลายในการจับคู่จากนั้นเลือกวิธีการครอสโอเวอร์แบบ POX สำหรับกำเนิดทายาท คือการสลับตำแหน่งลำดับการทำงานของโอเปอเรชันของโครโมโซมตามโครโมโซมต้นแบบ ในขั้นตอนนี้มีได้ทายาทจากการผสมพันธุ์เท่ากับจำนวนประชากรเริ่มต้น จากนั้นนำโครโมโซมที่เป็นทายาทมา

ปรับปรุงด้วยวิธีการมิวเทชันแบบโลคอล ซึ่งมีขั้นตอนเริ่มจากนำโครโมโซมทายาทมาเรียงลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามค่าฟิตเนสและแบ่งกลุ่มเป็น 4 กลุ่ม และนำแต่ละกลุ่มไปปรับปรุงด้วยการมิวเตชันแบบโลคอล โดยจะกำหนดให้มี 4 วิธี ซึ่งกลุ่มโครโมโซมที่มีค่าฟิตเนสดีสุดจะถูกมิวเตชันด้วยวิธีที่มีความซับซ้อนน้อยกว่ากลุ่มโครโมโซมที่มีค่าฟิตเนสแย่สุด จากนั้นนำโครโมโซมทั้งหมดมาพิจารณาโครโมโซมที่มีค่าฟิตเนสดีสุดเท่ากับจำนวนประชากรเริ่มต้นจะถูกพิจารณาเลือกให้เป็นประชากรรุ่นถัดไป การทำงานจะทำซ้ำจนกระทั่งครบตามเงื่อนไขที่กำหนดไว้ โครโมโซมที่มีค่าฟิตเนสดีสุดถือเป็นคำตอบของปัญหา

การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ ได้แบ่งเป็น 4 ส่วน ดังนี้

- 1) เพื่อทดสอบปัจจัยที่มีผลต่อประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA เปรียบเทียบกับจีเนติกอัลกอริทึมแบบดั้งเดิม โดยแยกเป็น 8 อัลกอริทึม
- 2) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Brandimarte's (BRdata) เปรียบเทียบกับอัลกอริทึมอื่น
- 3) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Fattahi et al.(Fdata) เปรียบเทียบกับอัลกอริทึมอื่น
- 4) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Barnes and Chamber library (BCdata) เปรียบเทียบกับผลการทดลองในงานวิจัยของ Demir และ İşleyen [37]

5.2 สรุปผลการทดสอบ

1) ปัจจัยการคัดเลือกโครโมโซมโดยคัดเลือกด้วยวงล้อรูเล็ตแบบฟิชชี การครอสโอเวอร์แบบแบ่งกลุ่ม และการมิวเตชันแบบโลคอลทำให้ประสิทธิภาพในการค้นหาคำตอบที่เหมาะสมที่สุดดีขึ้น โดยมีค่าคลาดเคลื่อนสัมพัทธ์ต่ำกว่าจีเนติกอัลกอริทึมแบบดั้งเดิม ได้แก่ 27.56%, 23.61%, 22.93%, 22.68%, 22.19%, 21.44%, 21.70% และ 20.12% ตามลำดับ

2) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Brandimarte's (BRdata) โดยเปรียบเทียบกับอัลกอริทึมอื่นเปรียบเทียบ ได้แก่ 1) LEGA 2) JSSANT 3) PSO 4) KBACO และ 5) BBOพบว่า อัลกอริทึมที่นำเสนอสามารถค้นหาคำตอบที่เหมาะสมได้ดีกว่า 5 อัลกอริทึมที่นำมาเปรียบเทียบ กล่าวคือมีค่าคลาดเคลื่อนต่ำกว่า 4.03% 6.54% 7.7% 2.63% และ 1.81% ตามลำดับ

3) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Fattahi et al.(Fdata) โดยเปรียบเทียบกับอัลกอริทึมอื่น ได้แก่ 1) HTS/SA 2) SA 3) AIA และ 4) Mathematical modelพบว่า อัลกอริทึมที่นำเสนอสามารถค้นหาคำตอบที่เหมาะสมได้ดีกว่า 4 อัลกอริทึมที่นำมาเปรียบเทียบ กล่าวคือมีค่าคลาดเคลื่อนต่ำกว่า 5.91% 0.48% 0.28% และ

0.59% ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่นำเสนอ MGA โดยทดสอบกับชุดข้อมูลมาตรฐาน Barnes and Chamber library (BCdata) โดยเปรียบเทียบกับผลการทดลองในงานวิจัยของ Demir และ İşleyen [37] พบว่า อัลกอริทึมที่นำเสนอสามารถค้นหาค่าคำตอบที่เหมาะสมได้ดีกว่าอัลกอริทึม BBO ที่นำมาเปรียบเทียบ โดยมีค่าคลาดเคลื่อนต่ำกว่า 0.27%

5.3 จุดเด่นของงานวิจัย

- 1) การเลือกโครโมโซมเพื่อเป็นโครโมโซมต้นแบบในแต่ละรอบการทำงานมีโอกาสเจอโครโมโซมตัวเดิมซ้ำ ซึ่งเมื่อใช้วิธีการคัดเลือกด้วยวงล้อสุ่มแบบฟิชชีทำให้สุ่มเลือกมาเพียงหนึ่ง ทำให้มีโอกาสเลือกโครโมโซมอื่นเพื่อเพิ่มความหลากหลายได้มากยิ่งขึ้น
- 2) การจับคู่ระหว่างโครโมโซมที่เหมือนกันสำหรับกำเนิดทายาทจะทำให้มีการคู่เข้าหาคำตอบที่เร็วเกินไปและพบกับคำตอบที่เหมาะสมเฉพาะพื้นที่ เพื่อเพิ่มความหลากหลายในการจับคู่การแบ่งกลุ่มโครโมโซมสามารถเพิ่มความหลากหลายในการกำเนิดทายาทได้
- 3) การปรับปรุงโครโมโซมด้วยวิธีการมิวเตชันหลายวิธีเพิ่มความหลากหลายในการค้นหาคำตอบโดยการนำค่าฟิตเนสของแต่ละโครโมโซมมาพิจารณาสำหรับเลือกวิธีการมิวเตชัน เพื่อเพิ่มระดับความหลากหลายให้สอดคล้องกับโครโมโซมนั้นๆ

5.5 จุดด้อยของงานวิจัย

ในงานวิจัยได้ปรับปรุงเพิ่ม 3 ส่วนคือ ส่วนของการคัดเลือก การครอสโอเวอร์ และการมิวเตชัน ทำให้ขั้นตอนการทำงานของอัลกอริทึมเพิ่มขึ้นเพื่อได้คำตอบที่ดีขึ้นจึงมีผลให้เวลาในการค้นหาคำตอบเพิ่มขึ้นตาม

5.4 ปัญหาที่พบในงานวิจัย

ปัญหาสำหรับด้านออฟติไมเซชันเชิงการจัดใช้เวลาในการทดสอบนาน ซึ่งจำเป็นต้องใช้เครื่องคอมพิวเตอร์มีความสามารถในการประมวลผลสูงและสามารถทำงานได้อย่างต่อเนื่องสำหรับการทดสอบ

5.5 แนวทางการพัฒนาต่อในอนาคต

1. พัฒนาอัลกอริทึมที่นำเสนอสำหรับทดสอบกับวิธีการประมวลผลแบบคู่ขนาน เพื่อปรับปรุงประสิทธิภาพการ และเวลาที่ใช้ทดสอบได้เร็วยิ่งขึ้น

- พัฒนาอัลกอริทึมที่นำเสนอร่วมกับอัลกอริทึมอื่นๆ เช่น ปัญญาเชิงกลุ่ม (Swarm Intelligence) สำหรับปัญหาออฟติไมเซชันเชิงตัวเลขอื่นๆ เช่น ปัญหาฟังก์ชัน หรือปัญหาการจัดตารางการผลิตที่มีหลายเงื่อนไข เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Belegundu A. D., Chandrupatla T. R. **Optimization Concept and Applications in Engineering**. New York: Cambridge University Press. 2011.
- [2] Kleeman M. P., Lamont G. B. “Scheduling of Flow-Shop, Job-Shop, and Combined Scheduling Problems using MOEAs with Fixed and Variable Length chromosomes.” **Studies in Computational Intelligence (SCI)**, vol. 49, 2007. pp. 49–99.
- [3] Shalizi C. “Distances between Clustering, Hierarchical Cluster” [Online]. Available: <http://www.stat.cmu.edu/~cshalizi/350/lectures/08/lecture-08.pdf>. 2009
- [4] Kumar V. “Cluster Analysis: Basic Concepts and Algorithms” [Online]. Available: <http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf>. 2010.
- [5] Holland, J. H. **Adaptation in Natural and Artificial Systems**. Cambridge, MA: MIT Press. 1975.
- [6] Zadeh L. A. “Fuzzy Sets” **Information and Control**, vol. 8, 1965. pp. 338 – 353.
- [7] Yazdani M., Gholami M., Zandieh M., Mousakhani M. “A Simulated Annealing Algorithm for Flexible Job-Shop Scheduling Problem” **Journal of Applied Sciences**, 2009.
- [8] Jiang J., Wen M., Ma K., Long X., Li J. “Hybrid Genetic Algorithm for Flexible Job-Shop Scheduling with Multi-objective” **Journal of Information & Computational Science**, 2011. pp. 2197 – 2205.
- [9] Pezzella F., Morganti G., Giaschetti G. “A genetic algorithm for the Flexible Job-Shop Scheduling Problem” **Computers & Operations Research**, 2011. pp. 3202 – 3212.
- [10] Fattahi P., Mehrabad M. S., Jolai F. “Mathematical modeling and heuristic approaches to flexible job shop scheduling problems” **J Intell Manuf**, 2007. pp. 331 – 342.
- [11] Lin T.L, Horng S.J, Kao T.W., Chen Y.H., Ran R. S., Chen R. J., Lai J.L., Kuo I. H. “An efficient job-shop scheduling algorithm based on particle swarm optimization” **Expert Systems with Applications**, vol. 37, 2010. pp. 2629–2636.
- [12] Beasley J.E. “OR-Library: Distributing test problems by electronic mail” **Journal of the Operations Research Society**, vol. 41, 1990. pp. 1069-1072.
- [13] Zhang G., Shao X., Li P., Gao L. “An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem” **Computers & Industrial Engineering**, 2009. pp. 1309 – 1318.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [14] Xing, L.N., Chen, Y.W., Wang, P., Zhao Q. S., Xiong J. “A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems” **Applied Soft Computing**, 2010. pp. 888 – 896.
- [15] Dorigo M., Maniezzo V., Colomi A. “The Ant System: Optimization by a colony of cooperating agents” **IEEE Transactions on System**, vol. 27, 1996. pp. 1 – 13.
- [16] Bagheri A, Zandieh M., Mahdavi I., Yazdani M. “An artificial immune algorithm for the flexible job-shop scheduling problem” **Future Generation Computer Systems**, 2010.
- [17] Burnet, F. M. “Clonal Selection and After”, **Theoretical Immunology**, (Eds.) G. I. Bell, A. S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc. 1978. pp. 63 - 85.
- [18] Muhamad A. S., Deris S. “An artificial immune system for solving production scheduling problem: a review” **Artif Intell Rev**, 2011.
- [19] Kacem I., Hammadi S., Borne P. “Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems”. **IEEE Trans. Systems, Man, and Cybernetics**, 2002, 1–13
- [20] Brandimarte P. “Routing and scheduling in a flexible job shop by tabu search” **Annals of Operations Research**, 1993. pp. 157 – 183.
- [21] Hurink J., Jurisch B., Thole M. “Tabu search for the job-shop scheduling problem with multi-purpose machine” **OR Spektrum**, 1994. pp. 205 – 215.
- [22] Giovanni L. D., Pezzella F., “An Improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling problem” **European Journal of Operational Research**, 2010. pp. 395 – 408.
- [23] Hinai N. A., EIMekkawy T. Y. “An efficient hybridized genetic algorithm architecture for the flexible job shop scheduling problem” **Flex Serv Manuf J**, 2011. pp. 64 – 85.
- [24] Zribi N., Kacem I., Kamel A. E., Borne P. “Assignment and Scheduling in Flexible Job-Shops by Hierarchical Optimization” **IEEE Transactions on systems**, vol. 37, 2007.
- [25] Gao J., Sun L., Gen M. “A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems” **Computers & Operations research**, 2008.
- [26] Zhang G., Gao L., Shi Y. “An effective genetic algorithm for the flexible job-shop scheduling problem” **Expert Systems with Applications**, 2011.
- [27] Barnes J. W. และ Chambers J. B. “Flexible Job Shop Scheduling by Tabu Search” The University of Texas, Austin, TX, Technical Report Series ORP96-09, **Graduate**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program in Operations Research and Industrial Engineering, 1996.

- [28] Peres S. D. “An intergrated approach for modeling and solving the genetal multiprocessor job-shop scheduling problem using tabu search” **Annal of Operation Research**, 1997.
- [29] Karimi H., Rahmti S. H. A. และ Zandieh M. “An efficient knowledge-based algorithm for the Flexible Job-Shop Scheduling problem” **Knowledge-Based Systems**, 2012.
- [30] Hansen P. และ Mladenovic N. “Variable neighborhood search: Principles and applications” **European Journal of Operational Research**, 2001. pp. 449 – 467.
- [31] Rahmti S. H. A และ Zandieh M. “A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem” **Inter Journal Adv.Manuf Technol**, 2012. pp.1115 – 1129.
- [32] Simon D. “Biogeography-Based Optimization” **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, 2008.
- [33] Wang X., Gao L., Zhang C. และ Shao X. “A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem” **Inter Journal Adv.Manuf Technol**, 2010.
- [34] Paul G. “Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem” **Operations Research Letters**, 2010.
- [35] Kamboj M.S. และ Sengupta J. “Comparative analysis of Simulated Annealing and tabu search channel allocation algorithms” **International Journal of Computer Theory and Engineering**, 2009, pp.1793 – 8201.
- [36] OR Library, URL <http://mscmga.ms.ic.ac.uk>.
- [37] Ho N. B., Tay J. C. และ Lai E. M. K. “An effective architecture for learning and evolving flexible job-shop schedules” **European Journal of Operational Research**, 2007, vol.179, no. 2, 316-333.
- [38] Girish B. S. และ Jawahar N. “Scheduling job shop associated with multiple routings with genetic and ant colony heuristics” **International Journal of Production Research**, 2009, 3891-3917.
- [39] Girish B. S. และ Jawahar N. “A particle swarm optimization algorithm for flexible job shop scheduling problem” **Annual IEEE Conference on Automation Science and Engineering**, 2009, 298- 303.
- [40] Demir Y. และ Isleyen S. K. “Evaluation of mathematical models for flexible job-shop scheduling problems” **Applied Mathematical Modelling**, 2013, 977 - 988.



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

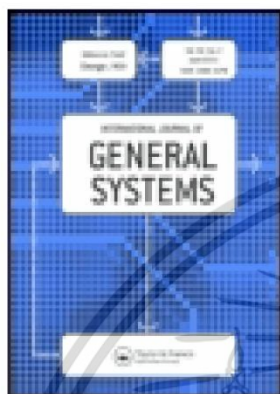
ภาคผนวก ก
ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

Thammano A., Teekeng W. “A modified genetic algorithm with fuzzy roulette wheel selection for job-shop scheduling problems” **International Journal of General Systems**, 2014.

Teekeng W., Thammano A. “A Combination of Shuffled Flog Leaping and Fuzzy Logic for Flexible Job-Shop Scheduling Problems” **Procedia Computer Science**, vol. 6, 2011. pp. 69–75.

Teekeng W., Thammano A. “Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems” **Procedia Computer Science**, vol. 12, 2012. pp. 122–128.

This article was downloaded by: [wannaporn teekeng]
 On: 07 January 2015, At: 05:18
 Publisher: Taylor & Francis
 Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered
 office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



CrossMark

[Click for updates](#)

International Journal of General Systems

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/ggen20>

A modified genetic algorithm with fuzzy roulette wheel selection for job-shop scheduling problems

Arit Thammano^a & Wannaporn Teekeng^a

^a Computational Intelligence Laboratory, Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

Published online: 03 Dec 2014.

To cite this article: Arit Thammano & Wannaporn Teekeng (2014): A modified genetic algorithm with fuzzy roulette wheel selection for job-shop scheduling problems, International Journal of General Systems, DOI: [10.1080/03081079.2014.969252](https://doi.org/10.1080/03081079.2014.969252)

To link to this article: <http://dx.doi.org/10.1080/03081079.2014.969252>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A modified genetic algorithm with fuzzy roulette wheel selection for job-shop scheduling problems

Arit Thammano* and Wannaporn Teekeng

Computational Intelligence Laboratory, Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

(Received 26 April 2013; accepted 22 September 2014)

The job-shop scheduling problem is one of the most difficult production planning problems. Since it is in the NP-hard class, a recent trend in solving the job-shop scheduling problem is shifting towards the use of heuristic and metaheuristic algorithms. This paper proposes a novel metaheuristic algorithm, which is a modification of the genetic algorithm. This proposed algorithm introduces two new concepts to the standard genetic algorithm: (1) fuzzy roulette wheel selection and (2) the mutation operation with tabu list. The proposed algorithm has been evaluated and compared with several state-of-the-art algorithms in the literature. The experimental results on 53 JSSPs show that the proposed algorithm is very effective in solving the combinatorial optimization problems. It outperforms all state-of-the-art algorithms on all benchmark problems in terms of the ability to achieve the optimal solution and the computational time.

Keywords: genetic algorithm; tabu search; fuzzy roulette wheel selection; job-shop scheduling problem

1. Introduction

Job-shop scheduling problems (JSSPs) occur in most of the manufacturing systems, in which each of the n jobs must be processed on all m machines in a given order. It is well known that the JSSP is in the NP-hard class; the JSSP is among the hardest combinatorial optimization problems. The total number of all possible schedules, including both feasible and infeasible solutions, is $(n!)^m$ for the problems with n jobs and m machines. For example, the number of possible solutions for a 10-job, 5-machine problem, 10-job, 10-machine problem and 30-job, 10-machine problem are 6.2924×10^{32} , 3.9594×10^{65} and 1.7243×10^{324} , respectively. It can be seen that the number of possible solutions grows at a much faster rate than the number of jobs and the number of machines. It is apparently impossible to evaluate all the alternatives to find the optimal solution even for a reasonably sized practical JSSP. Existing approaches used to solve the JSSP can be classified into three main categories: exact methods, heuristic methods and metaheuristic methods. The exact methods have been successfully applied to solve small-scale JSSPs (Haouari and Ladhari 2003; Pinedo 2008). Examples of commonly used exact methods are the branch and bound (Brucker, Jurisch, and Sievers 1994; Pinedo 2008), dynamic programming (Chen, Chu, and Proth 1998; Gromicho et al. 2012) and constraint logic programming (Varnier and Baptiste 1996; Paralić, Csontó, and Schmotzer 1997; Tan, Liu, and Wang 2010). However, for larger size problems, the exact

*Corresponding author. Email: arit@it.kmitl.ac.th

methods are computationally extremely expensive or even practically impossible (Binato et al. 2000). Therefore, the recent trend has shifted towards the other two categories, which sacrifice the guarantee of finding optimal solutions for the sake of getting good solutions in a significantly reduced amount of time. Examples for such approaches are tabu search (Glover 1989), simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983), genetic algorithm (Goldberg 1989), particle swarm optimization (PSO; Kennedy and Eberhart 1995) and many other approaches including fuzzy logic (Zadeh 1965).

A number of works involving the use of heuristic and metaheuristic algorithms to solve JSSPs can be found in the literature. Velmurugan and Selladurai (2007) applied the tabu search (TS) approach to minimize the makespan for the JSSP. In their version of TS, two modifications have been made to the original TS, which are (1) the priority dispatching rules are used to create the initial solution and (2) the first-last neighbourhood strategy with dynamic tabu length is used to search for new solutions. Zhang (2010) presented a new hybrid search algorithm, called TS&QQ. TS&QQ combines quasi-physics and quasi-human with the tabu search algorithm. The quasi-physics and quasi-human are used to trigger a jump for a stuck situation in order to get out of local minima. The experimental results show that TS&QQ is quite efficient for solving JSSPs. Yang et al. (2008) proposed a clonal selection-based memetic algorithm (CSMA) for JSSPs. This proposed algorithm consists of the clonal selection mechanism and the local search mechanism. The clonal selection-based algorithm explores different areas of the search space to find the most promising regions while the simulated annealing-based local search algorithm is presented to exploit the local regions around each solution for an even better solution. Shivasankaran et al. (2012) proposed an integrated approach based on hybridization of the bubble sorting algorithm and the simulated annealing algorithm. The bubble sorting algorithm is done first to confine the search space, and then the simulated annealing performs an extensive search over that search space. The authors claim that the proposed algorithm enhances convergence speed and reduces the computational time. Zhang, Rao, and Li (2008) presented a hybrid genetic algorithm which combines the local search to the genetic algorithm for the JSSP. In their hybrid algorithm, three novel features are proposed. First, a new full active schedule procedure based on the operation-based representation is introduced to construct schedule. Second, in order to preserve the meaningful characteristics of the previous generation, a new crossover operator, called the precedence operation crossover (POX), is proposed and adopted. Third, in order to reduce the disruptive effects of genetic operators, an improved generation alteration model is introduced. Hasan et al. (2009) introduced three priority rules, namely partial reordering, gap reduction and restricted swapping, for improving the performance of the traditional GA. These rules are applied in addition to the genetic operators. Their experimental results show that the proposed memetic algorithm outperforms the traditional GA and many state-of-the-art JSSP algorithms appearing in the literature. Asadzadeh and Zamanifar (2010) proposed an agent-based parallel genetic algorithm (PaGA) for the JSSP. In PaGA, four kinds of agents are developed: management agent, execute agent, processor agent and synchronization agent. First, management agent and execute agent are used to create the initial population. Then management agent divides the initial population into subpopulations of equal size and sends each of them to each processor agent. Each processor agent executes the genetic algorithm on its subpopulation. After the termination of genetic algorithm, synchronization agent coordinates migration among subpopulations of processor agents. In the migration phase, each processor agent exchanges some of its chromosomes with the neighbours. The genetic algorithm and the migration phase are repeated until an acceptable solution is obtained. The experimental results show that PaGA is more effective than the serial agent-based genetic algorithm. Gao et al. (2011) proposed a novel local search and combined it with the GA. In their proposed local

search, a systematic change of neighbourhood is carried out to avoid trapping in a local optimum, and to improve the quality of each solution. Raeesi and Kobti (2011) proposed a new memetic algorithm, which is a combination of GA and a local search heuristic. This memetic algorithm is based on a new chromosome representation, called Machine Operation Lists (MOL). MOL is a modification of the preference list-based representation. For the crossover operation, a linear version of order crossover is used. The experimental results show that MOL outperforms other existing representations. Moreover, the proposed MA is also very efficient when compared with another MA presented by Hasan et al. (2009). Ge et al. (2008) proposed a novel hybrid intelligent algorithm (HIA) based on the PSO and the artificial immune system (AIS). Since PSO is originally developed for continuous optimization problems, it cannot be directly applied to JSSPs. A new concept for the distance and velocity of a particle is presented. In addition, to improve the performance of AIS, a new vaccination operation and receptor editing are also proposed. The experimental results validate the effectiveness of the proposed HIA. Jamili, Shafia, and Tavakkoli-Moghaddam (2011) proposed a hybrid algorithm, namely PSO-SA, based on the PSO and the simulated annealing algorithm. The performance of PSO-SA is evaluated in comparison with that of SA, PSO and the electromagnetism like mechanism algorithm. The experimental results demonstrate that PSO-SA clearly outperforms those three algorithms.

Even though the above research studies show a very promising performance for JSSPs, many of them suffer from the problem of premature convergence to local optima. Premature convergence is one of the major reasons for poor performance in virtually all metaheuristic algorithms. It is due to the lack of diversity in the population. Most of the above researches use heuristics, such as TS and SA, to slow down the convergence speed of the search, thus helping in reducing the premature convergence problem. By doing this, the problem is alleviated but not completely solved. There is another way to deal with this problem by creating diversity in the population itself. To further prevent the premature convergence as well as accelerating the search process, Fuzzy roulette wheel (F-RW) selection is proposed in this paper to replace the original roulette wheel selection. Moreover, this research also applies the concept of tabu search to the mutation operation to prevent the occurrence of cycling and to avoid local optima.

The rest of the paper is organized as follows: Section 2 briefly discusses the background theories behind the proposed model. Section 3 describes the proposed F-RW selection. In Section 4, the proposed model with a detailed description of each step is presented. A brief description of the experiments and the experimental results is given in Section 5. Finally, Section 6 is the conclusions.

2. Background theories

2.1. Job-shop scheduling problem

A JSSP is an NP-hard optimization problem. That is, there is no known algorithm that guarantees to find an optimal solution of the JSSP within a polynomial time bound. The JSSP can be described as follows:

- There are a set of n jobs $J = \{j_1, j_2, \dots, j_n\}$ and a set of m machines $M = \{m_1, m_2, \dots, m_m\}$.
- Each job j_i consists of a sequence of m operations $O_i = \{o_{i1}, o_{i2}, \dots, o_{im}\}$.
- Each operation o_{ik} must be processed without interruption on a predefined machine for a duration of p_{ik} .

- Not more than one operation of any job can be executed simultaneously. In other words, the operations $o_{i1}, o_{i2}, \dots, o_{im}$ must be processed one after another in the given order.
- Each machine can process at most one operation at a time.
- Each job must pass through all machines exactly once to complete its work.
- The objective of the JSSP is to find a schedule of the operations on each machine that minimizes the makespan, the finishing time of the last operation in the schedule.

An example of a 3-job, 3-machine JSSP is given in Figure 1. The data in the body of the table consist of the routing of each job j_i through each machine (the number not in parentheses), and the processing time for each operation o_{ik} (the number in parentheses).

2.2. Genetic algorithm

Genetic algorithm (GA) is a method for finding the optimal solution to complex problems. A genetic algorithm is a heuristic search procedure which is based on the natural process of evolution as in biological sciences. GA starts with an initial population which contains a number of chromosomes; each chromosome represents a solution of the problem. Once the initial population has been created, the fitness of each chromosome will be evaluated and then the genetic operators will be applied to the population. The genetic operators typically consist of the following operators: selection, crossover and mutation. By applying genetic operators to the current population, a new set of population is created. Then the chromosomes of the current population whose fitness values are worse than the newly generated chromosomes are replaced. This evolution process is repeated until a predefined number of generations have been reached or the fitness of the best chromosome has reached a certain level.

2.3. Fuzzy set

Fuzzy set was introduced by Zadeh in 1965. It is an extension of the classical set theory for dealing with uncertainty, vagueness and ambiguity. In classical set, the boundary of the set is crisp, that is, an element in the universe of discourse either belongs or does not belong to the set. On the contrary, the boundary of a fuzzy set is not precise. Fuzzy set theory permits an element to have partial membership in a fuzzy set and/or multiple memberships in several different fuzzy sets. The degree of membership of an element e in a fuzzy set A is represented by a real number in the range $[0, 1]$.

$$\mu_A(e) \in [0, 1] \quad (1)$$

Job	Machine (Operation Time)		
	o_{i1}	o_{i2}	o_{i3}
J_1	1 (6)	2 (6)	3 (4)
J_2	1 (2)	3 (10)	2 (6)
J_3	2 (6)	1 (4)	3 (6)

Figure 1. Example of a 3-job, 3-machine JSSP.

The value 0 means that an element is not a member of the fuzzy set; the value 1 means that an element is fully a member of the fuzzy set; the values between 0 and 1 mean that an element is partially a member of the fuzzy set.

2.4. Tabu search

Tabu search (TS) was proposed by Glover in 1989. TS is an enhancement of the hill climbing heuristic. It uses a memory function to avoid being trapped at the local optimum. TS is based on two main ideas of (1) exploring the search space of all feasible solutions by a sequence of moves and (2) forbidding some search directions in order to avoid cycling and jump off the local optimum. The procedure of the TS algorithm is described as follows:

- (1) Create an initial solution and define this solution as both the current solution and the best solution.
- (2) Generate a set of neighbourhood solutions from the current solution, and then evaluate their fitnesses.
- (3) Designate the fittest admissible neighbour as the new current solution. By definition, an admissible neighbour is the one which is not in the tabu list or satisfies the aspiration criteria. The tabu list stores all the solutions visited in the last t iterations in order to forbid the moves which might cause the cycling. However, sometimes, the best possible move is a tabu move. Aspiration criteria are introduced in the tabu search to determine when tabu restrictions can be overridden.
- (4) If the new current solution is better than the best solution found to date, record it as the new best solution.
- (5) Update the tabu list and the aspiration criteria.
- (6) If the stopping criteria are met, stop and return the best solution; otherwise, go to step (2).

3. Fuzzy roulette wheel selection

There are several selection strategies, such as roulette wheel selection, tournament selection, rank selection and random selection. The most popular one is the roulette wheel selection. Roulette wheel selection is one of the oldest and simplest selection schemes in which p chromosomes are selected from the current population with a probability proportional to their fitness to be the parent chromosomes. The fitter the chromosome is the higher the probability of that chromosome being selected to be the parent chromosomes. The original roulette wheel selection suffers from the problem of premature convergence due to the lack of diversity in the population. To be more specific, the premature convergence occurs when most chromosomes in the population have similar gene content. The scenario is even worse as all the similar chromosomes have high fitness values while the remaining chromosomes with diverse solutions have lower fitness values. According to the original concept of the roulette wheel selection, the chromosomes with diverse solutions are given only small slices of the roulette wheel. Therefore, the system gets a very small chance of finding new superior members.

To overcome the problem of premature convergence, F-RW selection is proposed in this paper to replace the original roulette wheel selection. F-RW selection applies the concept of the fuzzy set to the original roulette wheel selection technique. The main concept of the proposed F-RW selection is that the chromosomes which are similar in their contents are placed next to each other with some degree of overlap in the roulette wheel. The size of the overlapping area is

positively correlated with the similarity between the two adjacent chromosomes. The more similar the two adjacent chromosomes is, the bigger the size of the overlapping area. By doing this, the ratio of the total area of all similar chromosomes to the whole area of the roulette wheel is smaller. This gives the chromosome with diverse content higher chance of being selected for the crossover operation, thus reducing the premature convergence problem. The F-RW selection is divided into three stages. The following paragraphs show an example of how to create the F-RW.

In Figure 2(a), there are four chromosomes in the population; each chromosome represents a different possible solution. The fitness values of the four chromosomes are 0.05, 0.045, 0.036 and 0.033, respectively. The purpose of the first stage is to assign a slice of the

Downloaded by [wannaporn teekeng] at 05:18 07 January 2015

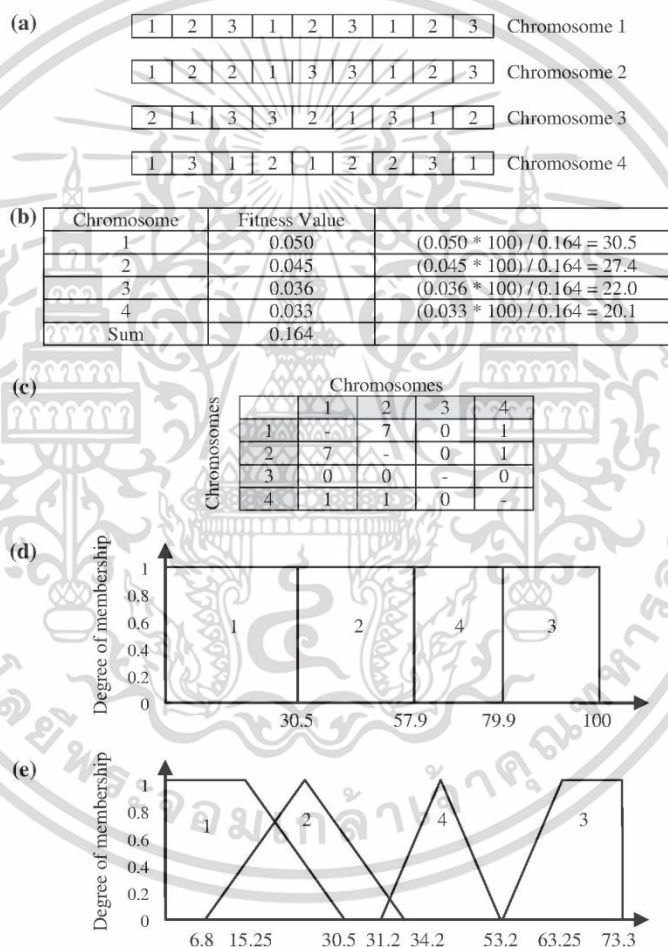


Figure 2. Example of how to create the F-RW.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

roulette wheel to each chromosome in the population. In this example, the roulette wheel is divided into four slices. Each chromosome will receive a slice of the roulette wheel proportional to its fitness. Since chromosome 1 has the highest fitness, it will receive the largest slice. The size of each slice can be calculated by the following equation:

$$A_s = \frac{f_s}{\sum_{n=1}^N f_n} \times 100 \quad (2)$$

where f_s is the fitness value of the chromosome s . N is the total number of chromosomes in the population. As shown in Figure 2(b), the slices which represent chromosomes 1, 2, 3 and 4 have size equal to 30.5, 27.4, 22 and 20.1% of the total area of the roulette wheel.

The next stage is to determine the position of each chromosome on the roulette wheel. In this paper, the first chromosome is always assigned to the first slot of the roulette wheel. Then the similarity between the first chromosome and the rest of the chromosomes (chromosomes 2, 3 and 4) are calculated. The similarity matrix in Figure 2(c) shows that chromosomes 1 and 2 are the most similar pair of the four chromosomes; therefore, they will be placed next to each other in the roulette wheel. That is, the second chromosome occupies the second slot. Next, the similarity between chromosome 2 and the remaining unassigned chromosomes (chromosomes 3 and 4) is determined. It can be seen from Figure 2(c) that chromosome 2 is more similar to chromosome 4 than to chromosome 3. Therefore, the third slot is assigned to the chromosome 4. This process is repeated until all chromosomes are allocated slots in the roulette wheel. In this example, at the end of this stage, the chromosomes are arranged in the following order: 1-2-4-3.

In contrast to the original roulette wheel selection, which has a clear-cut boundary between a chromosome and its neighbouring chromosomes as shown in Figure 2(d), F-RW selection allows overlapping between the chromosomes which are right next to each other. Each chromosome is represented by a fuzzy set with a triangular or a trapezoidal membership function. The leftmost and the rightmost fuzzy sets are represented by trapezoidal membership functions while other fuzzy sets in the middle are represented by triangular membership functions. Here is the last stage of the F-RW selection. The last stage is to determine the overlapping area between the fuzzy sets of the adjacent chromosomes. To be in line with the calculation in the first stage, the first chromosome, which occupies the first slot of the roulette wheel, is represented by a fuzzy set with a trapezoidal membership function with parameters (0, 0, 15.25, 30.5). Then the overlapping area between the fuzzy sets of chromosomes 1 and 2 is determined. The size of the overlapping area depends on how similar the two chromosomes are. The more similar two chromosomes are, the more they overlap. According to the above concept, the left, centre and right corners of the triangular membership function are calculated by using the following equations:

$$LC_s = RC_{s-1} - \left[A_{s-1} \times \frac{IG}{G} \right] \quad (3)$$

$$RC_s = LC_s + A_s \quad (4)$$

$$CP_s = \frac{LC_s + RC_s}{2} \tag{5}$$

where LC_s is the left corner of a triangular membership function of the fuzzy set s , RC_s is the right corner of a triangular membership function of the fuzzy set s , CP_s is the centre point of a triangular membership function of the fuzzy set s , IG is the number of identical genes in the two neighbouring chromosomes and G is the number of genes in the chromosome.

By comparing chromosomes 1 and 2, it can be seen that seven out of nine genes in the two chromosomes are the same. Therefore, LC_2 , RC_2 and CP_2 are 6.8, 34.2 and 20.5, respectively. Figure 2(e) shows the complete F-RW of this example.

In the selection of a chromosome for the crossover process, first, a random number between 0 and 1 is generated. Then the random number is multiplied by the right bound of the rightmost membership function, which is 73.3 in this example. Next, the membership degree to which a generated random number belongs to each chromosome is calculated. The chromosome with maximum membership value is chosen.

4. Methodology

4.1. Chromosome representation

In the literature, several different representations have been proposed for JSSPs (Gen, Cheng, and Lin 2008) such as operation-based representation, job-based representation, completion time-based representation, random key-based representation, preference list-based representation, priority rule-based representation, disjunctive graph-based representation and machine-based representation. The operation-based representation is used in this research for solution encoding. This representation encodes a schedule as a sequence of operations. Each gene represents one operation. Therefore, the length of each chromosome is equal to the total number of operations of all jobs. That is, for an n -job, m -machine JSSP, the chromosome will contain $n \times m$ genes. In the operation-based representation, all operations of a job are denoted by a job name; therefore, each job name will occur m times in the chromosome. For better understanding, consider a 3-job, 3-machine JSSP given in Figure 1, the chromosomes which encode candidate solutions to this problem will consist of nine genes as shown in Figure 3. In Figure 3, the numbers 1, 2 and 3 stand for jobs J_1 , J_2 and J_3 , respectively. Since each job has three operations, its name appears exactly three times in the chromosome. From left to right of the chromosome 2 in Figure 3, for example, the first gene 1 corresponds to the first operation of job J_1 , the second gene 2 corresponds to the first operation of job J_2 and the third gene 2 corresponds to the second operation of job J_2 , and so on. Therefore, the chromosome [1 2 2 1 3 3 1 2 3] can be converted into a list of ordered operations of [O_{11} , O_{21} , O_{22} , O_{12} , O_{31} , O_{32} , O_{13} , O_{23} , O_{33}].

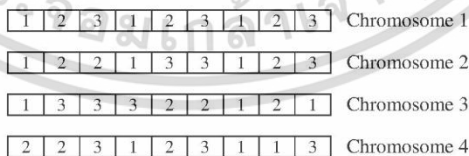


Figure 3. Initial chromosomes.

Downloaded by [wannaporn teekeng] at 05:18 07 January 2015

4.2. The proposed algorithm

The proposed algorithm is developed according to the concepts of genetic algorithm and fuzzy logic. The detailed procedure of the proposed algorithm is explained as follows:

4.2.1. Generate the initial population

In this first step, an initial population of C chromosomes $X = \{x_1, x_2, \dots, x_c, \dots, x_C\}$ is randomly created. Each chromosome represents a possible solution to a problem. The length of each chromosome is equal to the total number of operations of all jobs.

4.2.2. Evaluate the fitness of the chromosomes

The fitness is a measure of the quality of the solution represented by the chromosome. In general, the fitness function should correspond to the objective function of the problem in hand. Since the objective of the JSSP is to find a schedule of the operations on each machine that minimizes the makespan, in this paper; therefore, the reciprocal of the finishing time of the last operation in the schedule is used as the fitness value for each chromosome.

$$\text{Fitness} = \frac{1}{C_{max}} \quad (6)$$

$$C_{max} = \max_{1 \leq j \leq n} \{c_j\} \quad (7)$$

where c_j is the finishing time of the last operation of job j .

4.2.3. Fuzzy roulette wheel selection

The proposed F-RW selection, described in Section 3, is used here to select pairs of chromosomes for the crossover operation.

4.2.4. Crossover

Crossover is a basic operator of GA. It generates new chromosomes by combining sections of selected parent chromosomes. There are many different crossover operators. It is important to note that the crossover operator must be chosen according to how the chromosome represents the solution. As mentioned earlier in Section 4.1, the operation-based representation is adopted to encode the chromosomes in this paper. Therefore, the widely used POX, which can very well preserve the valuable characteristics of the parent chromosomes and always make the offspring feasible, is employed in this research. The procedure of the POX operator is as follows:

- (1) Randomly choose a particular job name from Parent 1 and Parent 2. For example, in Figure 4, job J_2 is chosen.
- (2) Copy the genes with the selected job name to the offspring, Child 1 and Child 2. Child 1 gets the genes from Parent 1 while Child 2 gets the genes from Parent 2.
- (3) Delete the copied genes from both parents.

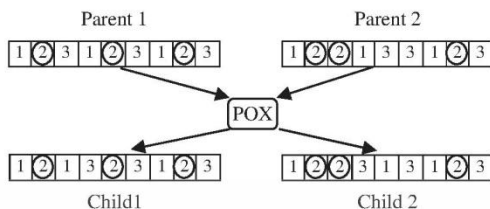


Figure 4. POX crossover.

- (4) According to the gene sequence in Parent 1, fulfil Child 2 with the remaining genes in Parent 1. At the same time, according to the gene sequence in Parent 2, fulfil Child 1 with the remaining genes in Parent 2.

4.2.5. Mutation operation with tabu list

In general, mutation is used to produce perturbations on chromosomes in order to maintain the diversity of population. In this paper, the exchange mutation (EM) operator proposed by Banzhaf (1990) is used. The EM operator randomly selects two genes in the chromosome and then swaps them as shown in Figure 5(a). The elements of both genes must not be the same. If the elements of the selected genes are the same as shown in Figure 5(b), the swapping pair will be reselected. To prevent the occurrence of cycling and to avoid local optimal, moreover, this paper also forbids the algorithm from selecting very recently swapped pairs by keeping them in the tabu list for the next *t* iterations.

4.2.6. Termination criteria

Steps 4.2.2–4.2.5 are repeated until any of the stopping criteria is met. The stopping criteria used in this research are as follows:

- The maximum number of iterations has been reached.
- Eighty percent of the population shares the same value.

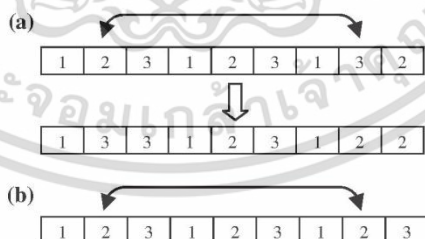


Figure 5. An example of the exchange mutation.

Downloaded by [wannaporn teekeng] at 05:18 07 January 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Experimental results

To evaluate the performance of the proposed algorithm, the proposed algorithm has been tested both in terms of the number of generations required to reach the best known solutions and the solution quality. Several experiments have been conducted on 53 test instances of various sizes and hardness levels. All test instances were downloaded from Beasley's OR-Library (Beasley 1990). The test instances can be classified into three groups as follows:

- The first group contains 40 instances of eight different sizes ($n \times m = 10 \times 5, 15 \times 5, 20 \times 5, 10 \times 10, 15 \times 10, 20 \times 10, 30 \times 10, 15 \times 15$) denoted as La01–La40 due to Lawrence (1984).
- The second group contains 10 instances denoted as Orb01–Orb10 due to Applegate and Cook (1991).
- The third group, contributed by Fisher and Thompson (1963), contains three instances symbolized as Ft06 (6×6), Ft10 (10×10) and Ft20 (20×5).

First, we examine the performance of the proposed model and the standard genetic algorithm in terms of the number of generations required to reach the best known solutions. In the standard GA, the standard genetic operators are used which include the roulette wheel selection, the one-point crossover and the exchange mutation. The following is the summary of the results. With the proposed model, the number of generations required to reach the best known solutions is always lower than its standard counterpart. Figures 6–11 are samples of the results obtained when applying both algorithms to solve La06, La11, La23, La31, Orb09 and Ft10. In Figures 6, 7 and 9, both algorithms can achieve the best known solutions; however, the proposed model is able to reach the best known solutions much faster than the standard GA. For La06, the proposed model reaches the best known solution in only 10 generations, compared to 21 for the standard GA. For La11, the proposed model converges in 29 generations, compared to 41 for the standard GA. For La31, the proposed model takes 305 generations to converge while the standard GA takes 444 generations. In Figures 8, 10 and 11, only the proposed model obtains the best known solutions.

Next, the proposed algorithm is compared with several state-of-the-art benchmark algorithms in terms of the solution quality. The benchmark algorithms are as follows:

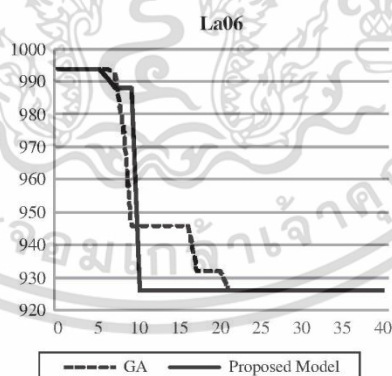


Figure 6. Convergence curves for La06.

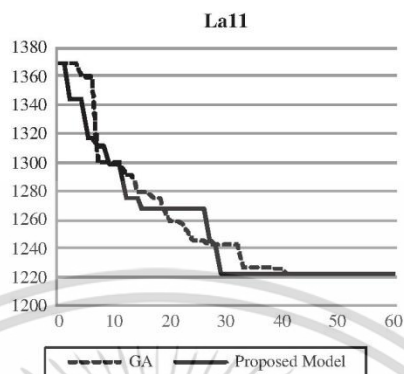


Figure 7. Convergence curves for La11.

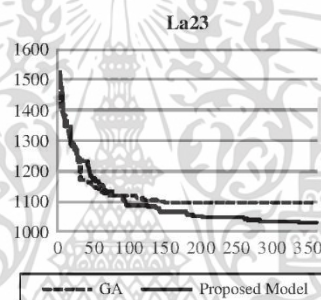


Figure 8. Convergence curves for La23.

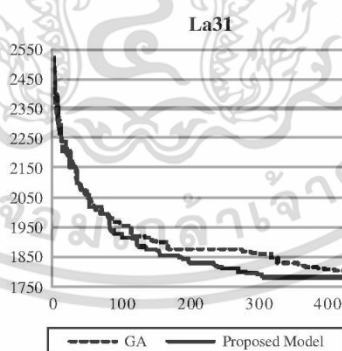


Figure 9. Convergence curves for La31.

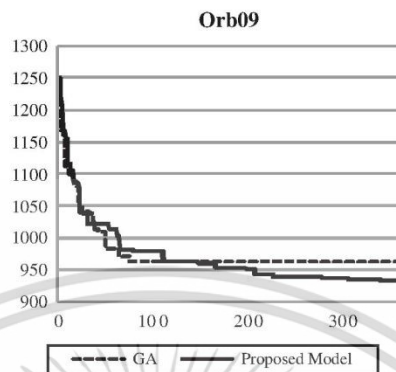


Figure 10. Convergence curves for Orb09.

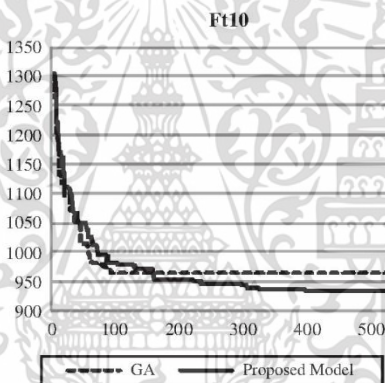


Figure 11. Convergence curves for Ft10.

- (1) Tabu search (TS): Velmurugan and Selladurai (2007).
- (2) Tabu search algorithm based on quasi-physics and quasi-human (TS&QQ): Zhang (2010).
- (3) CSMA: Yang et al. (2008).
- (4) Memetic algorithm with gap reduction and restricted swapping rules (MA(GR-RS)): Hasan et al. (2009).
- (5) Agent-based PaGA: Asadzadeh and Zamanifar (2010).
- (6) Memetic algorithm (MA 1): Gao et al. (2011).
- (7) Memetic algorithm (MA 2): Raeesi and Kobti (2011).
- (8) HIA: Ge et al. (2008).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The performance of these algorithms is compared both in terms of the best solution found and the average relative error over all test instances within each group. The relative error RE (%) is calculated for each instance as a percentage deviation of the obtained solution from the best known solution, as shown in the following equation:

$$RE(\%) = \frac{C_{\text{best}} - \text{BKS}}{\text{BKS}} \times 100 \quad (8)$$

where C_{best} is the best solution obtained from the algorithm. BKS is the best known solution.

The computational results are given in Tables 1–4. The figures in brackets indicate the relative error in percent. Table 1 presents the performance of the proposed method on 20 small- to medium-sized test instances of the first group (La01–La20), with reference to eight state-of-the-art approaches mentioned above. The average relative errors of the proposed model, TS, TS&QQ, CSMA, MA(GR-RS), PaGA, MA 1, MA 2 and HIA are 0.00, 0.01, 0.97, 0.04, 0.03, 0.94, 0.00, 0.03 and 0.00%, respectively. The proposed model, MA 1 and HIA are able to achieve the best known solutions for all 20 instances while TS, TS&QQ, CSMA, MA(GR-RS), PaGA and MA 2 fail to achieve the best known solutions on 1, 8, 2, 1, 7 and 1 instances (out of 20).

Table 2 lists the results obtained by applying the proposed method to 20 large-sized test instances of the first group (La21–La40); the obtained results are compared with those obtained by other six approaches: TS&QQ, MA(GR-RS), PaGA, MA 1, MA 2 and HIA. The symbol “–” denotes that the algorithm was not tested on that instance. As illustrated in Table 2, the average relative error of the proposed model (0.59) is the best among the compared approaches. The proposed model achieves the best known solutions for 10 out of 20 problem instances. For TS&QQ, seven of the best known solutions are obtained with the average relative error of 3.81; MA(GR-RS) achieves eight with the average relative error of 2.14. PaGA is only tested on 12 instances (La21–La32), and it is not able to find the best known solutions for any of the tested instances. For MA 1, 10 of the best known solutions are achieved with the average relative error of 0.71; MA 2 achieves eight with the average relative error of 0.84; HIA achieves nine with the average relative error of 0.70.

Table 3 shows the performance of the proposed method in finding the optimal solutions of all 10 instances of the second group. Among the compared approaches, the proposed model again has the smallest average relative error. Out of 10 instances, the proposed model obtains the best known solutions for seven instances. PaGA is not able to find the best known solution for any of the instances. It should be noted that TS is only applied to five instances (Orb01–Orb05), and it achieves the best known solutions for three instances.

Table 4 illustrates the comparative results of the proposed model and five of the above-mentioned state-of-the-art algorithms on all three instances of the third group. The proposed model, CSMA, MA 1 and HIA perform perfectly on all three instances while TS and PaGA only get the best known solution for Ft06, the small-sized test instance.

In summary, the performance comparisons with other state-of-the-art approaches support the validity of the proposed approach. First, the proposed approach is evaluated against the standard GA which uses a classical roulette wheel selection. The experimental results show that the proposed approach outperforms the standard GA both in terms of the number of generations required to reach the best known solutions and the ability to achieve the optimal solution. Second, the performance of the proposed approach is compared with that of eight state-of-the-art algorithms. The results in Tables 1–4 show that the proposed approach is superior to all other approaches in terms of the ability to find the optimal solution. A closer

Downloaded by [wannaporn teekeng] at 05:18 07 January 2015

Table 1. Experimental results for small- to medium-sized instances of the first group.

Size	BKS	TS	TS&QQ	CSMA	MA(GR-RS)	PaGA	MA 1	MA 2	HIA	Proposed model
La01	10 × 5	666 (0.00)	666 (0.00)	666 (0.00)	666 (0.00)	666 (0.00)	666 (0.00)	666 (0.00)	666 (0.00)	666 (0.00)
La02	10 × 5	655 (0.00)	667 (1.83)	655 (0.00)	655 (0.00)	655 (0.00)	655 (0.00)	655 (0.00)	655 (0.00)	655 (0.00)
La03	10 × 5	597 (0.00)	617 (3.35)	597 (0.00)	597 (0.00)	617 (3.35)	597 (0.00)	597 (0.00)	597 (0.00)	597 (0.00)
La04	10 × 5	590 (0.00)	607 (2.88)	590 (0.00)	590 (0.00)	607 (2.88)	590 (0.00)	590 (0.00)	590 (0.00)	590 (0.00)
La05	10 × 5	593 (0.00)	593 (0.00)	593 (0.00)	593 (0.00)	593 (0.00)	593 (0.00)	593 (0.00)	593 (0.00)	593 (0.00)
La06	15 × 5	926 (0.00)	926 (0.00)	926 (0.00)	926 (0.00)	926 (0.00)	926 (0.00)	926 (0.00)	926 (0.00)	926 (0.00)
La07	15 × 5	890 (0.00)	890 (0.00)	890 (0.00)	890 (0.00)	890 (0.00)	890 (0.00)	890 (0.00)	890 (0.00)	890 (0.00)
La08	15 × 5	863 (0.00)	863 (0.00)	863 (0.00)	863 (0.00)	863 (0.00)	863 (0.00)	863 (0.00)	863 (0.00)	863 (0.00)
La09	15 × 5	951 (0.00)	951 (0.00)	951 (0.00)	951 (0.00)	951 (0.00)	951 (0.00)	951 (0.00)	951 (0.00)	951 (0.00)
La10	15 × 5	958 (0.00)	958 (0.00)	958 (0.00)	958 (0.00)	958 (0.00)	958 (0.00)	958 (0.00)	958 (0.00)	958 (0.00)
La11	20 × 5	1222 (0.00)	1222 (0.00)	1222 (0.00)	1222 (0.00)	1222 (0.00)	1222 (0.00)	1222 (0.00)	1222 (0.00)	1222 (0.00)
La12	20 × 5	1039 (0.00)	1039 (0.00)	1039 (0.00)	1039 (0.00)	1039 (0.00)	1039 (0.00)	1039 (0.00)	1039 (0.00)	1039 (0.00)
La13	20 × 5	1150 (0.00)	1150 (0.00)	1150 (0.00)	1150 (0.00)	1150 (0.00)	1150 (0.00)	1150 (0.00)	1150 (0.00)	1150 (0.00)
La14	20 × 5	1292 (0.00)	1292 (0.00)	1292 (0.00)	1292 (0.00)	1292 (0.00)	1292 (0.00)	1292 (0.00)	1292 (0.00)	1292 (0.00)
La15	20 × 5	1207 (0.00)	1207 (0.00)	1207 (0.00)	1207 (0.00)	1207 (0.00)	1207 (0.00)	1207 (0.00)	1207 (0.00)	1207 (0.00)
La16	10 × 10	945 (0.00)	981 (3.81)	945 (0.00)	945 (0.00)	994 (5.19)	945 (0.00)	945 (0.00)	945 (0.00)	945 (0.00)
La17	10 × 10	784 (0.00)	793 (1.15)	784 (0.00)	784 (0.00)	793 (1.15)	784 (0.00)	784 (0.00)	784 (0.00)	784 (0.00)
La18	10 × 10	848 (0.00)	861 (1.53)	848 (0.00)	848 (0.00)	860 (1.42)	848 (0.00)	848 (0.00)	848 (0.00)	848 (0.00)
La19	10 × 10	842 (0.00)	852 (1.19)	844 (0.24)	842 (0.00)	873 (3.68)	842 (0.00)	842 (0.00)	842 (0.00)	842 (0.00)
La20	10 × 10	902 (0.00)	935 (3.66)	907 (0.55)	907 (0.55)	912 (1.11)	902 (0.00)	907 (0.55)	902 (0.00)	902 (0.00)
Average relative error		0.01	0.97	0.04	0.03	0.94	0.00	0.03	0.00	0.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. Experimental results for large-sized instances of the first group

Size	BKS	TS	TS&QQ	CSMA	MA(GR-RS)	PaGA	MA 1	MA 2	HIA	Proposed model
La21	1046	-	1114 (6.50)	-	1079 (3.15)	1146 (9.56)	1055 (0.86)	1057 (1.05)	1046 (0.00)	1053 (0.67)
La22	927	-	963 (3.88)	-	960 (3.56)	1007 (8.63)	927 (0.00)	935 (0.86)	932 (0.54)	927 (0.00)
La23	1032	-	1032 (0.00)	-	1032 (0.00)	1033 (0.10)	1032 (0.00)	1032 (0.00)	1032 (0.00)	1032 (0.00)
La24	935	-	977 (4.49)	-	959 (2.57)	1012 (8.24)	940 (0.53)	944 (0.96)	950 (1.60)	941 (0.64)
La25	977	-	1043 (6.76)	-	991 (1.43)	1067 (9.21)	984 (0.72)	983 (0.61)	979 (0.21)	978 (0.10)
La26	1218	-	1287 (5.67)	-	1218 (0.00)	1323 (8.62)	1218 (0.00)	1218 (0.00)	1218 (0.00)	1218 (0.00)
La27	1235	-	1309 (5.99)	-	1286 (4.13)	1359 (10.04)	1261 (2.11)	1269 (2.75)	1256 (1.70)	1262 (2.19)
La28	1216	-	1269 (4.36)	-	1286 (5.76)	1369 (12.58)	1216 (0.00)	1223 (0.58)	1227 (0.91)	1216 (0.00)
La29	1152	-	1270 (10.24)	-	1221 (5.99)	1322 (14.76)	1190 (3.30)	1191 (3.39)	1184 (2.78)	1163 (0.95)
La30	1355	-	1355 (0.00)	-	1355 (0.00)	1437 (6.05)	1355 (0.00)	1355 (0.00)	1355 (0.00)	1355 (0.00)
La31	1784	-	1784 (0.00)	-	1784 (0.00)	1844 (3.36)	1784 (0.00)	1784 (0.00)	1784 (0.00)	1784 (0.00)
La32	1850	-	1850 (0.00)	-	1850 (0.00)	1907 (3.08)	1850 (0.00)	1850 (0.00)	1850 (0.00)	1850 (0.00)
La33	1719	-	1719 (0.00)	-	1719 (0.00)	-	1719 (0.00)	1719 (0.00)	1719 (0.00)	1719 (0.00)
La34	1721	-	1721 (0.00)	-	1721 (0.00)	-	1721 (0.00)	1721 (0.00)	1721 (0.00)	1721 (0.00)
La35	1888	-	1888 (0.00)	-	1888 (0.00)	-	1888 (0.00)	1888 (0.00)	1888 (0.00)	1888 (0.00)
La36	1268	-	1334 (5.21)	-	1307 (3.08)	-	1281 (1.03)	1281 (1.03)	1281 (1.03)	1283 (1.18)
La37	1397	-	1502 (7.52)	-	1442 (3.22)	-	1431 (2.43)	1429 (2.29)	1415 (1.29)	1424 (1.93)
La38	1196	-	1240 (3.68)	-	1266 (5.85)	-	1216 (1.67)	1208 (1.00)	1213 (1.42)	1218 (1.84)
La39	1233	-	1310 (6.25)	-	1252 (1.54)	-	1241 (0.65)	1248 (1.22)	1246 (1.05)	1251 (1.46)
La40	1222	-	1290 (5.57)	-	1252 (2.45)	-	1233 (0.90)	1234 (0.98)	1240 (1.47)	1233 (0.90)
Average relative error	-	-	3.81	-	2.14	7.85	0.71	0.84	0.70	0.59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3. Experimental results for instances of the second group.

	Size	BKS	TS	TS&QQ	CSMA	MA(GR-RS)	PaGA	MA 1	MA 2	HIA	Proposed model
Orb01	10 × 10	1059	1064 (0.47)	—	—	—	1149 (8.50)	—	—	—	1064 (0.47)
Orb02	10 × 10	888	888 (0.00)	—	—	—	929 (4.62)	—	—	—	888 (0.00)
Orb03	10 × 10	1005	1005 (0.00)	—	—	—	1129 (12.34)	—	—	—	1005 (0.00)
Orb04	10 × 10	1005	1011 (0.60)	—	—	—	1062 (5.67)	—	—	—	1005 (0.00)
Orb05	10 × 10	887	887 (0.00)	—	—	—	936 (5.52)	—	—	—	889 (0.23)
Orb06	10 × 10	1010	—	—	—	—	1060 (4.95)	—	—	—	1012 (0.20)
Orb07	10 × 10	397	—	—	—	—	416 (4.79)	—	—	—	397 (0.00)
Orb08	10 × 10	899	—	—	—	—	1010 (12.35)	—	—	—	899 (0.00)
Orb09	10 × 10	934	—	—	—	—	994 (6.42)	—	—	—	934 (0.00)
Orb10	10 × 10	944	—	—	—	—	—	—	—	—	944 (0.00)
Average relative error			0.21	—	—	—	7.24	—	—	—	0.09

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4. Experimental results for instances of the third group.

Size	BKS	TS	TS&QQ	CSMA	MA(GR-RS)	FaGA	MA 1	MA 2	HIA	Proposed model
F06	55	55 (0.00)	—	55 (0.00)	—	55 (0.00)	55 (0.00)	—	55 (0.00)	55 (0.00)
F10	930	935 (0.54)	—	930 (0.00)	—	997 (7.20)	930 (0.00)	—	930 (0.00)	930 (0.00)
F20	1165	—	—	1165 (0.00)	—	1196 (2.66)	1165 (0.00)	—	1165 (0.00)	1165 (0.00)
Average relative error		0.27	—	0.00	—	3.29	0.00	—	0.00	0.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

look at the results reveals the following. First, all algorithms perform extremely well when applied to small- to medium-sized instances; there is no significant difference in the performance between the proposed algorithm and the benchmark algorithms. Second, for large-sized instances, the gaps between the performance of the proposed algorithm and that of the benchmark algorithms get wider.

6. Conclusions

This paper presents a modified genetic algorithm for the JSSP. The operation-based representation is adopted to encode the chromosome. To prevent the problem of premature convergence, a new selection method, named F-RW selection, is proposed in this paper to replace the original roulette wheel selection. The crossover and mutation operators used in this paper are the POX and the exchange mutation. This research also applies the concept of tabu search to the mutation operation to prevent the occurrence of cycling and to avoid local optimal. The proposed algorithm is tested on 53 benchmark instances taken from Beasley's OR-Library. The experimental results show that the proposed algorithm outperforms all state-of-the-art algorithms evaluated in this study in terms of the number of generations required to reach the best known solutions and the ability to achieve the optimal solution.

Notes on contributors



Arit Thammano is an associate professor at Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. His current research interests include evolutionary algorithms, memetic algorithms, swarm intelligence and optimization techniques.



Wannaporn Teekeng is a PhD candidate at Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. Her current research interests include evolutionary algorithms, memetic algorithms and optimization techniques.

References

- Applegate, D., and W. Cook. 1991. "A Computational Study of the Job-shop Scheduling Problem." *ORSA Journal on Computing* 3: 149–156.
- Asadzadeh, L., and K. Zamanifar, 2010. "An Agent-based Parallel Approach for the Job Shop Scheduling Problem with Genetic Algorithms." *Mathematical and Computer Modelling* 52: 1957–1965.
- Banzhaf, W. 1990. "The 'Molecular' Traveling Salesman." *Biological Cybernetics* 64: 7–14.
- Beasley, J. E. 1990. "OR-library: Distributing Test Problems by Electronic Mail." *Journal of the Operational Research Society* 41 (11): 1069–1072.
- Binato, S., W. J. Hery, D. M. Loewenstern, and M. G. C. Resende. 2000. "A GRASP for Job Shop Scheduling." *AT&T Labs Research Technical Report*, 00.6.1, 1–17.
- Brucker, P., B. Jurisch, and B. Sievers. 1994. "A Branch and Bound Algorithm for the Job-shop Scheduling Problem." *Discrete Applied Mathematics* 49: 107–127.

- Chen, H., C. Chu, and J. Proth. 1998. "An Improvement of the Lagrangean Relaxation Approach for Job Shop Scheduling: A Dynamic Programming Method." *IEEE Transactions on Robotics and Automation* 14 (5): 786–795.
- Fisher, H., and G. L. Thompson. 1963. "Probabilistic Learning Combinations of Local Job-shop Scheduling Rules." In *Industrial Scheduling*, edited by J. F. Muth and G. L. Thompson, 225–251. Eaglewood Cliffs, NJ: Prentice Hall.
- Gao, L., G. Zhang, L. Zhang, and X. Li. 2011. "An Efficient Memetic Algorithm for Solving the Job Shop Scheduling Problem." *Computers & Industrial Engineering* 60 (4): 699–705.
- Ge, H., L. Sun, Y. Liang, and F. Qian. 2008. "An Effective PSO and AIS-based Hybrid Intelligent Algorithm for Job-shop Scheduling." *IEEE Transactions on Systems, Man, and Cybernetics – Part A* 38 (2): 358–368.
- Gen, M., R. Cheng, and L. Lin. 2008. *Network Models and Optimization: Multiobjective Genetic Algorithm Approach*. London: Springer-Verlag.
- Glover, F. 1989. "Tabu Search – Part I." *ORSA Journal on Computing* 1 (3): 190–206.
- Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.
- Gromicho, J. A. S., J. J. Van Hoorn, F. Saldanha-da-Gama, and G. T. Timmer. 2012. "Solving the Job-shop Scheduling Problem Optimally by Dynamic Programming." *Computers & Operations Research* 39: 2968–2977.
- Haouari, M., and T. Ladhari. 2003. "A Branch-and-bound-based Local Search Method for the Flow Shop Problem." *Journal of the Operational Research Society* 54: 1076–1084.
- Hasan, S. M. K., R. Sarker, D. Essam, and D. Cornforth. 2009. "Memetic Algorithms for Solving Job-shop Scheduling Problems." *Memetic Computing* 1: 69–83.
- Jamili, A., M. A. Shafia, and R. Tavakkoli-Moghaddam. 2011. "A Hybrid Algorithm Based on Particle Swarm Optimization and Simulated Annealing for a Periodic Job Shop Scheduling Problem." *The International Journal of Advanced Manufacturing Technology* 54: 309–322.
- Kennedy, J., and R. Eberhart. 1995. "Particle Swarm Optimization." Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, 1942–1948.
- Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220 (4598): 671–680.
- Lawrence, S. 1984. *Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques*. Pittsburgh, PA: Graduate School of Industrial Administration.
- Paralić, J., J. Csontó, and M. Schmotzer. 1997. "Optimal Scheduling Using Constraint Logic Programming." Proceedings of the 8th Symposium on Information Systems, Varazdin, 65–72.
- Pinedo, M. L. 2008. *Scheduling: Theory, Algorithms, and Systems*. 3rd ed. New York: Springer.
- Raeesi, M. R. and Z. Kobi. 2011. "A Machine Operation Lists Based Memetic Algorithm for Job Shop Scheduling." Proceedings of the 2011 IEEE Congress on Evolutionary Computation, New Orleans, LA, USA, 2436–2443.
- Shivasankaran, N., P. S. Kumar, G. Nallakumarasamy, and K. V. Raja. 2012. "A Hybrid Bubble Sorting Simulated Annealing Algorithm for Job Shop Scheduling." Proceedings of the Third International Conference on Computing Communication & Networking Technologies, Coimbatore, India, 1–5.
- Tan, Y., S. Liu, and D. Wang. 2010. "A Constraint Programming-based Branch and Bound Algorithm for Job Shop Problems." Proceedings of the 2010 Chinese Control and Decision Conference, Xuzhou, China, 173–178.
- Varnier, C., and P. Baptiste. 1996. "Constraint Logic Programming and Scheduling Problems." Proceedings of the 1996 IEEE International Conference on Systems, Man, and Cybernetics, Beijing, China, 2942–2946.
- Velmungan, P. S., and V. Selladurai. 2007. "A Tabu Search Algorithm for Job Shop Scheduling Problem with Industrial Scheduling Case Study." *International Journal of Soft Computing* 2 (4): 531–537.
- Yang, J. H., L. Sun, H. P. Lee, Y. Qian, and Y. C. Liang. 2008. "Clonal Selection Based Memetic Algorithm for Job Shop Scheduling Problems." *Journal of Bionic Engineering* 5: 111–119.
- Zadeh, L. A. 1965. "Fuzzy Sets." *Information and Control* 8 (3): 338–353.
- Zhang, S. 2010. "A Tabu Search Algorithm Based on Quasi-physics and Quasi-human for Solving the Job Shop Scheduling Problem." Proceedings of the Second IITA International Conference on Geoscience and Remote Sensing, Qingdao, China, 112–115.
- Zhang, C., Y. Rao, and P. Li. 2008. "An Effective Hybrid Genetic Algorithm for the Job Shop Scheduling Problem." *The International Journal of Advanced Manufacturing Technology* 39: 965–974.

Available online at www.sciencedirect.com**SciVerse ScienceDirect**

Procedia Computer Science 6 (2011) 69–75

Procedia
 Computer Science

Complex Adaptive Systems, Volume 1
 Cihan H. Dagli, Editor in Chief
 Conference Organized by Missouri University of Science and Technology
 2011- Chicago, IL

A Combination of Shuffled Frog Leaping and Fuzzy Logic for Flexible Job-Shop Scheduling Problems

Wannaporn Teekeng*, Arit Thammano

*Computational Intelligence Laboratory, Faculty of Information Technology,
 King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 Thailand*

Abstract

Flexible Job-Shop Scheduling Problem (FJSP) is a well known NP-hard combinatorial optimization problem. Over the last decade, many algorithms have been proposed to tackle FJSP. Evolutionary algorithms, which solve problems by mimicking the process of natural evolution, are the most widely used techniques in solving FJSP. This paper proposes a novel evolutionary algorithm that integrates the concept of a fuzzy logic into Shuffled Frog Leaping Algorithm. In the proposed SFLA-FS model, the fuzzy roulette wheel selection is used in selecting frogs to form a sub-memplex. This selection method is proved to be better than the typically used rank selection. The objective of this research is to find a schedule for each of the 10 benchmark problems that minimize their makespan. The experimental results obtained from SFLA-FS show that the SFLA-FS is very efficient for all tested problems.

© 2011 Published by Elsevier B.V.

Keywords: Flexible Job-Shop Scheduling; Fuzzy Logic; Shuffled Frog Leaping Algorithm; Fuzzy Roulette Wheel

1. Introduction

Scheduling problems occur in many real-world applications, e.g. manufacturing, supply chain planning, and production line planning. Many artificial intelligence techniques have been proposed to find the optimal solutions of the problems. It is well known that the Job-Shop Scheduling Problem (JSSP) is one of the hardest combinatorial optimization problems. The Flexible Job-Shop Scheduling Problem (FJSP) is an extension of the Job-Shop Scheduling Problem (JSSP). FJSP allows each operation to be processed on more than one machine. For smaller

* Corresponding author. Tel.: +66(0)2723-4900; fax: +66(0)2723-4910.
 E-mail address: w_teekeng@hotmail.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

problems, mathematical programming, such as Integer linear programming and Branch-and-bound algorithms, can be used to find the optimal solutions; however, the above methods cannot guarantee optimal results for larger problems. Recently, a number of metaheuristic approaches, such as Genetic Algorithm (GA) [1], Tabu Search [2], Simulated Annealing (SA) [3], Particle Swarm Optimization (PSO) [4], and other approaches including Fuzzy Logic [5] have gained a lot of attention from researchers in the area.

Shuffled Frog-leaping algorithm (SFLA) is a new addition to the range of intelligent algorithms and a new member to the family of memetic algorithms. The SFLA uses memetic evolution in the form of infection of ideas from one individual to another in a local search. The local search is similar in concept to a PSO algorithm and can search for food based on a colony. In addition, SFLA can search for global exploration and is easy to understand. In recent years, several modifications have been proposed: Solving TSP with Shuffled Frog-leaping Algorithm [6]; Improved Shuffled Frog-leaping Algorithm for continuous optimization problem [7]; A Novel Memetic Algorithm for Global Optimization Based on PSO and SFLA [8].

In this paper, we propose a novel evolutionary algorithm that integrates the concept of fuzzy logic into Shuffled Frog-leaping Algorithm. The novel fuzzy roulette wheel selection is proposed and used in selecting frogs to form a sub-memplex. Additionally, the local search procedure is used to further improve the resulted solution.

This paper is organized as follows: Section 2 briefly describes the flexible job-shop scheduling problems and the original shuffled frog leaping algorithm. In Section 3, the proposed SFLA-FS algorithm is presented. The experimental results are given in Section 4. Section 5 provides the conclusion to the paper.

2. Related Theories

2.1. Flexible Job-Shop Scheduling Problems

The Flexible Job-Shop Scheduling Problem is an extension of the classic job shop scheduling problem which allows an operation to be processed by any machine out of a set of available machines. The objective of FJSP is to schedule a set of R jobs $J = \{J_1, J_2, \dots, J_R\}$ on a set of S machines $M = \{M_1, M_2, \dots, M_S\}$ so that the makespan (C_{max}) is minimized. Each job may have different number of operations. Each operation O_{ij} , the j^{th} operation of the i^{th} job, can be processed on any of the allowable machines. An example of FJSP is given in Table 1. Each row refers to an operation; each column refers to a machine and cells are processing times. As illustrated in Table 1, for example, the 3rd operation of the 2nd job is only allowed to be processed on M_1 and M_3 .

Table 1. Example of the Flexible Job-Shop Scheduling problem.

Job	Operation	Machines		
		M_1	M_2	M_3
J_1	$O_{1,1}$	2	4	8
	$O_{1,2}$	5	4	7
	$O_{1,3}$	2	-	-
J_2	$O_{2,1}$	10	11	10
	$O_{2,2}$	6	5	-
	$O_{2,3}$	2	-	6

2.2. The Shuffled Frog Leaping Algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Shuffled Frog Leaping Algorithm (SFLA) is proposed by Eusuff and Lansey [9, 10]. The algorithm is inspired by the behavior of frogs when seeking for the food. The algorithm begins by randomly selecting F frogs and sorting them according to their fitness value order. Next, the frogs are divided into m memeplexes. For each memeplex, q frogs are randomly selected to form a sub-memeplex. The chance of being selected is proportional to the frog fitness; fitter frogs have a higher chance of being included in the sub-memeplex.

$$P_j = 2(n+1-j) / [n(n+1)] \tag{1}$$

where n is the number of frogs in each memeplex.

In each sub-memeplex, the worst frog is selected. Then, the position of the worst frog in each sub-memeplex is updated according to Eq. (2), where the worst frog learns to move from the best frog in its sub-memeplex.

$$F_{worst}^{k+1} = F_{worst}^k + r(F_{sbest}^k - F_{worst}^k) \tag{2}$$

where F_{worst} is the position of the worst frog in the sub-memeplex.

F_{sbest} is the position of the best frog in the sub-memeplex.

r is a random number between 0 and 1.

k is the iteration number of the sub-memeplex.

If the new position is better than the previous position, the previous position is replaced by the new position. Otherwise Eq. (3) is used to update the position of the worst frog instead.

$$F_{worst}^{k+1} = F_{worst}^k + r(F_{gbest}^k - F_{worst}^k) \tag{3}$$

where F_{gbest} is the position of the best frog in the population.

3. The FSLA-FS algorithm

This paper proposes a novel evolutionary algorithm that integrates the concept of a fuzzy logic into Shuffled Frog Leaping Algorithm (SFLA-FS). The detail of the algorithm is given below:

3.1. Generate the Initial Population

In this step, an initial population of F frogs $X = (x_1, x_2, \dots, x_B, \dots, x_F)$ is created. Each frog, which in this paper is defined as shown in Fig 1, represents a possible solution to a problem. The length of each frog is equal to the total number of operations. As illustrated in Fig 1, each meme consists of two components: the operation to be processed and its corresponding machine. The first components of the memes are initialized by randomly permute the operations to be scheduled. Then, starting from the first meme, the corresponding machine is selected for the operation by using the shortest processing time (SPT) scheme as shown in Fig 2.

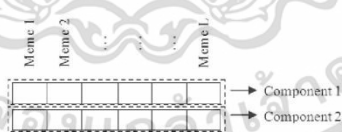


Fig. 1 The structure of a frog.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

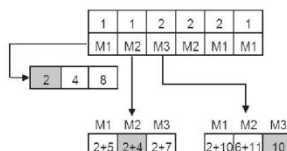


Fig. 2. Example of the process of assigning the machine to the operation.

3.2. Construct a sub-memplex using the fuzzy roulette wheel selection

The F frogs are ranked in descending order. Then they are partitioned into m memplexes. Each memplex contains n frogs. For each memplex, q frogs are selected to form a sub-memplex. This paper proposed a new approach in selecting frogs to form a sub-memplex. The main concept of the proposed approach is that the pair of frogs which are similar in their content are placed next to each other with some degree of overlap in the roulette wheel. For example, in Table 2, the population consists of 4 frogs; each frog represents a different possible solution. The following paragraphs show how to create the fuzzy roulette wheel.

Table 2. Some solutions of the example in Table 1.

Frog	Solutions
1	$O_{2,1}, O_{1,1}, O_{2,2}, O_{1,2}, O_{1,3}, O_{2,3}$
2	$O_{2,1}, O_{1,1}, O_{2,2}, O_{1,2}, O_{2,3}, O_{1,3}$
3	$O_{1,1}, O_{2,1}, O_{1,2}, O_{2,2}, O_{2,3}, O_{1,3}$
4	$O_{2,1}, O_{1,1}, O_{1,2}, O_{2,2}, O_{2,3}, O_{1,3}$

In this paper, the first frog is always assigned to the first slot of the roulette wheel. The slot size is proportional to its fitness value. Then the similarity between the first frog and the rest of the frogs (frogs 2, 3, and 4) are calculated. A pair of frogs with the highest similarity score is chosen; in this case, the second frog is selected with a similarity score of 4 (Fig 3(a)). Therefore, the second frog owns the second slot. The same process is repeated until all frogs are allocated slots in the roulette wheel. As illustrated in Figs 3(b) and 3(c), the fourth frog is assigned to the third slot while the third frog is assigned to the fourth slot of the roulette wheel.

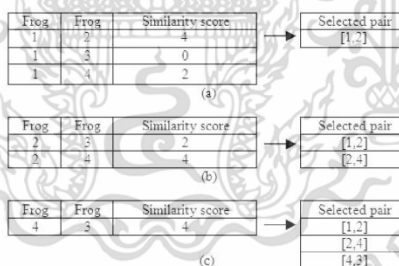


Fig. 3. Example of the process of assigning frogs to slots of a roulette wheel.

After all frogs are allocated slots in the roulette wheel, the overlapping area between the two adjacent frogs is determined. The size of the overlapping area is positively correlated with the similarity between the two adjacent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

frogs. The more similar the two adjacent frogs is, the bigger the size of the overlapping area. Fig 4 shows the fuzzy roulette wheel of the example in Table 2.

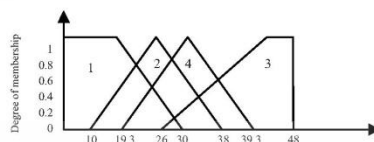


Fig. 4. Fuzzy roulette wheel

In the selection of frogs to form a sub-memeplex, first, a random number is generated. Then the membership degree to which a generated random number belongs to each frog is calculated. The one with the highest membership degree is chosen. This selection process continues until a total of q frogs has been selected.

3.3. Improve the performance of the worst frog of each memeplex

The proposed SLFA-FS model employs the crossover and the mutation operators to improve the quality of the solutions. Crossover operator plays an important part in improving the worst frog in each memeplex. After forming a sub-memeplex within each memeplex, the top two ranking frogs in a sub-memeplex are selected for crossover. Once the crossover points are defined, the Precedence Operation Crossover (POX) is adopted to develop a new offspring (Eq. (4)). If the new offspring is fitter than the worst frog in the corresponding memeplex, the worst frog will be replaced by this new offspring. If not, the crossover between the best frog in the population and the second rank frog in the sub-memeplex is performed (Eq. (5)).

$$Offspring = F_{sbest} \oplus F_{2s} \tag{4}$$

$$Offspring = F_{gbest} \oplus F_{2s} \tag{5}$$

where F_{2s} is the position of the second rank frog in the sub-memeplex.
 \oplus is the crossover denotation.
 F_{sbest} is the position of the best frog in the sub-memeplex.
 F_{gbest} is the position of the best frog in the population.

Best frog :	Job	2	1	1	2	1	2
	Machine	M1	M1	M2	M2	M1	M1
Second rank frog :	Job	2	2	1	1	1	2
	Machine	M3	M1	M2	M2	M1	M1
Offspring :	Job	2	1	1	2	1	2
	Machine	M3	M1	M2	M1	M1	M1

Fig. 5. Example of the crossover operation

Once again, if the new offspring is fitter than the worst frog in the corresponding memeplex, the worst frog will be replaced by this new offspring. Following the crossover operation, the local search, which will be described in section 3.5, is performed on the new frog. However, if the new offspring is still worse than the worst frog in the corresponding memeplex, the mutation operator is employed to perform the mutation on the second rank frog in the sub-memeplex. Mutation operator is used in order to maintain the diversity of the population and to overcome premature convergence. The mutation procedure is as follows: 1) Select the machine that processes the last operation of the schedule. 2) List all operations that are processed on this machine. 3) Randomly select one operation from the list. 4) Re-assign the new machine to the selected operation. 5) Determine the fitness value of the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mutated frog. If the mutated frog is fitter than the worst frog in the corresponding memplex, the worst frog will be replaced by this mutated frog. Then the mutated frog is further improved by the local search. However, if the mutated frog is not fitter than the worst frog in the corresponding memplex, a new frog is randomly generated to replace that worst frog. This process continues until all m memplexes are processed.

3.4. Shuffle the memplexes

All the memplexes are combined together. Then the population is sorted in order of decreasing performance value. Next, the termination criterion is checked. If it is met, the algorithm is terminated. If not, the population is repartitioned into memplexes and the loop continues.

3.5. Local search procedure

The local search procedure used in this paper further improves the solution by swapping the operations within a block on the critical path. The critical path is the longest route from source to sink in the disjunctive graph. It is possible to decompose the critical path into a number of blocks (a block is a maximal sequence of adjacent critical operations that is processed on the same machine). Fig 6(a) shows the Gantt chart of one of the resulted solutions $O_{1,1}, O_{2,1}, O_{1,2}, O_{2,2}, O_{2,3}, O_{1,3}$. In Fig 6(a), the critical path is $O_{2,1}-O_{2,2}-O_{2,3}-O_{1,3}$. There is only one block on this critical path, which is $O_{2,3}-O_{1,3}$. By swapping the sequence of operations $O_{2,3}$ and $O_{1,3}$, the makespan of the solution is decreased from 19 to 17 as shown in Fig 6(b).

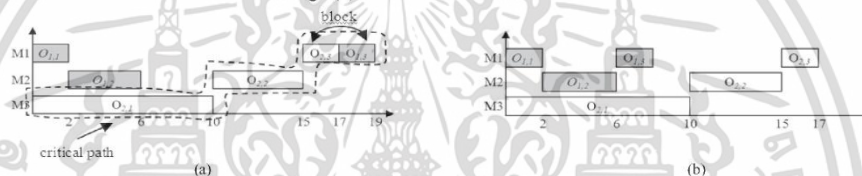


Fig. 6. Local search procedure

4. Experimental Results

The proposed algorithm was tested on 10 benchmark problems taken from [13]. The performance of the proposed algorithm is compared to that of the previous research work by Brandimarte [14]. The system parameters of SFLA-FS are set as follows: $F = 50, 100$; $N = 100$; $m = 10\%$ of F ; $n = F/m$; $G = 100$; and $q = 2/3$ of n . The percentage relative error from the best known solution (RE) is used as a measure to evaluate the performance of the proposed method.

Table 3 shows the experimental results of our SFLA-FS in comparison to the tabu search of Brandimarte [14]. The results indicate that the SFLA-FS algorithm outperforms Brandimarte's approach in 6 out of 9 problems.

5. Conclusion

This paper proposes a modification to the shuffled frog leaping algorithm for solving the flexible job-shop scheduling problem (FJSP). The proposed model is a combination of the shuffled frog leaping algorithm and the fuzzy logic concept. The experimental results show that the performance of the shuffled frog leaping algorithm is significantly improved when a new fuzzy roulette wheel selection is used to construct a sub-memplex, instead of the rank selection typically used in the original shuffled frog leaping algorithm. Our future work will deal with multi-objective FJSP, a much more complex version of the FJSP.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3. Experimental results

Problems	R x S	LB	Brandimarte [14]			SFLA-FS			
			Popsiz	Cmax	RE (%)	Popsiz	Cmax	RE (%)	
MK01	10 x 6	36	100	42	16.67	50	41*	13.89	
MK02	10 x 6	24	100	32	33.33	100	30*	25.00	
MK04	15 x 8	48	100	81	68.75	50	68*	41.67	
MK05	15 x 4	168	100	186	10.71	100	181*	7.74	
MK06	10 x 15	33	100	86	160.61	100	75*	127.27	
MK07	20 x 5	133	100	157*	18.05	100	159	19.55	
MK08	20 x 10	523	100	523*	0.00	100	523*	0.00	
MK09	20 x 10	299	100	369*	23.41	50	369*	23.41	
MK10	20 x 15	165	100	296	79.39	100	272*	64.85	
Average Relative Error.					45.66			35.93	

References

1. D.E. Goldberg, Genetic Algorithms in Search Optimisation and Machine Learning, Addison-Wesley, Reading, Massachusetts, 1989.
2. F. Glover, Tabu Search-Part I, Operations Research Society of America, J. Comput., (1989) 4–32.
3. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, Sci., vol. 220 (1983) 671-680.
4. J. Kennedy and R. Eberhard, Particle Swarm Optimization, Phys.Rev.,B.(1995) 13:5344-5348.
5. L. Zadeh, Fuzzy sets, Information and Control, 8(3) (1965) 338-353.
6. X.-H. Luo, Y. Yang and X. Li, Solving TSP with shuffled frog-leaping algorithm, Proc. ISDA (2008) 228 - 232
7. Ziyang Zhen, Daobo Wang, Yuanyuan Liu, Improved Shuffled Frog Leaping Algorithm for Continuous Optimization Problem, IEEE Congress on Evolutionary Computation, CEC, Norway, Trondheim, 2009.
8. Z. Y. Zhen, Z. S. Wang, Z. Gu, Y. Y. Liu, A novel memetic algorithm for global optimization based on PSO and SFLA, Lecture Notes in Computer Science, Springer (2007) 127-136.
9. M.M. Eusuff, K.E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, in J Water Resource Plan Manage (2003) 210–225.
10. M.M. Eusuff, K.E. Lansey, F. Pasha, Shuffled Frog-Leaping Algorithm: A Memetic Meta-heuristic for Discrete Optimization, Engineering and Technology, Mathematics and Optimization (2006) 129–154.
11. S.N. Sivanandam, S. Sumathi and S.N. Deepa, Introduction to Fuzzy Logic using Matlab, Springer-Verlag, Berlin Heidelberg, 2007.
12. Z. Chaoyong, R. Yunqing and Li. Peigen, An effective hybrid genetic algorithm for the job shop scheduling problem, Int J Adv Manuf Technol 39 (2008) 965–974.
13. M. Mastrolilli, Flexible Job Shop Problem, <http://www.idsia.ch/~monaldo/fjsp.html>.
14. P. Brandimarte, Routing and scheduling in a flexible job shop by tabu search, Annals of Operations Research 22 (1993) 158–183.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 12 (2012) 122 – 128

Procedia
Computer Science

Complex Adaptive Systems, Publication 2
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2012- Washington D.C.

Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems

Wannaporn Teekeng*, Arit Thammano

*Computational Intelligence Laboratory, Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 Thailand*

Abstract

This paper proposes a modified version of the genetic algorithm for flexible job-shop scheduling problems (FJSP). The genetic algorithm (GA), a class of stochastic search algorithms, is very effective at finding optimal solutions to a wide variety of problems. The proposed modified GA consists of 1) an effective selection method called "fuzzy roulette wheel selection," 2) a new crossover operator that uses a hierarchical clustering concept to cluster the population in each generation, and 3) a new mutation operator that helps in maintaining population diversity and overcoming premature convergence. The objective of this research is to find a schedule that minimizes the makespan of the FJSP. The experimental results on 10 well-known benchmark instances show that the proposed algorithm is quite efficient in solving flexible job-shop scheduling problems.

Keywords: Flexible Job-Shop Scheduling Problems; Genetic Algorithm; Fuzzy Roulette Wheel Selection; Hierarchical Clustering

1. Introduction

The flexible job-shop scheduling problem (FJSP) is one of the hardest combinatorial optimization problems. It is an extension of the classical job-shop scheduling problem (JSSP). The difference between the two is that FJSP allows each operation to be processed on more than one machine. However, different machines require different processing time for each operation. Comparing to the JSSP, FJSP is more complex and more difficult to solve. Recently, a number of metaheuristic approaches, such as Genetic Algorithms (GA) [1], Tabu Search [2], Simulated Annealing (SA) [3], Particle Swarm Optimization (PSO) [4], and other approaches including Fuzzy Logic [5] have gained a lot of attention from researchers in the area.

GA has been applied to many combinatorial problems such as job scheduling and the travelling salesman problem. Some relevant works are: Pezzella et al. [6] proposed an algorithm that integrates different strategies for

* Corresponding author. Tel.: +66-2-723-4964; fax: +66-2-723-4910.
E-mail address: w_teekeng@hotmail.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

generating the initial population, selecting the individuals for reproduction, and reproducing new individuals. Zhang et al. [7] developed new concepts for generating a high-quality initial population, called Global selection (GS) and Local selection (LS). Al-Hinai and ElMekkawy [8] proposed hybridized genetic algorithm architecture for the FJSP. In their algorithm, the efficiency of the GA is enhanced by integrating it with an initial population generation algorithm and a local search method. Ho et al. [9] proposed a new architecture based on the GA for learning and evolving FJSP called learnable genetic architecture (LEGA).

In this paper, we propose three new operations for the genetic algorithm. First, a new selection method, called fuzzy roulette wheel selection, is proposed and used in selecting chromosomes to form a mating pool. Second, a new crossover operator which employs the concept of a hierarchical clustering algorithm is used in crossover the mating pair. Third, a new mutation operator is applied to the offspring to maintain population diversity and overcome premature convergence.

This paper is organized as follows: Section 2 briefly describes the flexible job-shop scheduling problems. In Section 3, the proposed algorithm is presented. The experimental results are given in Section 4. Section 5 provides the conclusion to the paper.

2. Problem definition

The Flexible Job-Shop Scheduling Problem (FJSP) is an extension of the classical job shop scheduling problem. It allows an operation to be processed by any machine out of a set of available machines. The objective of FJSP is to schedule a set of R jobs $J = \{J_1, J_2, \dots, J_R\}$ on a set of S machines $M = \{M_1, M_2, \dots, M_S\}$ so that the makespan (C_{max}) is minimized. Each job may have a different number of operations. Each operation O_{ij} , the j^{th} operation of the i^{th} job, can be processed on any of the available machines. An example of FJSP is given in Table 1. Each row refers to an operation; each column refers to a machine and cells are processing times. As illustrated in Table 1, for example, the 3rd operation of the 2nd job is only allowed to be processed on M_1 and M_3 . The symbol “-” means that the machine cannot execute the corresponding operation.

Table 1. Example of the flexible job-shop scheduling problem.

Job	Operation	Machines		
		M_1	M_2	M_3
J_1	$O_{1,1}$	2	4	8
	$O_{1,2}$	5	4	7
	$O_{1,3}$	2	-	-
J_2	$O_{2,1}$	10	11	10
	$O_{2,2}$	6	5	-
	$O_{2,3}$	2	-	6

The following hypotheses for the FJSP, which are previously stated in [10], are made in this paper:

- All machines are available at time $t = 0$.
- All jobs are released at time $t = 0$.
- For each job, the sequence of operations is predetermined and cannot be modified.
- Each machine can only execute one operation at a time.
- Each operation O_{ij} must be processed without interruption on one of the S machines.
- The objective of the FJSP is to minimize the makespan.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. The proposed algorithm

The proposed algorithm is developed according to the concept of the genetic algorithm. The detailed procedure of the proposed algorithm is as follows:

3.1. Generate the Initial Population

As shown in Fig 1, the structure of the chromosome used in this paper consists of two components. The first component of the chromosome represents the sequence of operations to be processed while the second component contains the list of machines used in executing the operations in the first component.

For example, consider the 2-job, 3-machine problem given in Table 1. An initial population of 4 chromosomes is randomly created as shown in Fig 2. Each chromosome contains 12 genes. The numbers 1 and 2 which appear in the first component of the chromosome stand for jobs J_1 and J_2 respectively. Since each job has three operations, the numbers 1 and 2 therefore occur three times in the first component of the chromosome. On the other hand, the numbers 1, 2, and 3 which appear in the second component of the chromosome stand for machines M_1 , M_2 , and M_3 respectively. According to the chromosome 4 in Fig 2, the second component is [3 2 3 2 2 1]. The first gene in the second component 3 means that the 1st operation of the 2nd job is to be processed by the machine M_3 .

Component 1						Component 2					
1	2	1	2	1	2	1	2	2	1	3	

Fig. 1. Schematic representation of the chromosome structure

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

Fig. 2. Initial chromosomes

3.2. Fitness Evaluation

In this step, a fitness function is applied to evaluate the fitness of all chromosomes in the population. In line with the main objective of the FJSP, the fitness function used in this study is the reciprocal of the makespan of the chromosome.

3.3. Selection

In the genetic algorithm, parent chromosomes are selected with a probability related to their fitness. Highly fit chromosomes have a higher probability of being selected for mating than less fit chromosomes. In order to improve the performance of the genetic algorithm, the new selection method, called the fuzzy roulette wheel selection, is proposed in this paper. The concept of the fuzzy roulette wheel selection is that two chromosomes which are similar in content are placed next to each other with some degree of overlap in the roulette wheel [11]. The similarity between two chromosomes is measured based on the number of genes on one chromosome whose position and content are identical to those of the other chromosome. To be more specific, the similarity value of a pair of chromosomes is the summation of 1) the number of the same O_{ij} positions that both chromosomes have and 2) the number of the operations O_{ij} on both chromosomes that have chosen the same machine. For example, in Table 2, the population consists of 4 chromosomes; each chromosome represents a different possible solution. The following paragraphs show how to create the fuzzy roulette wheel selection.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. Some solutions of the example in Table 1

Chromosome	Solutions
1	O _{1,1} , O _{2,1} , O _{2,2} , O _{2,3} , O _{1,2} , O _{1,3} M ₂ , M ₃ , M ₂ , M ₁ , M ₁ , M ₁
2	O _{1,1} , O _{2,1} , O _{1,2} , O _{1,3} , O _{2,2} , O _{2,3} M ₁ , M ₁ , M ₁ , M ₁ , M ₁ , M ₃
3	O _{2,1} , O _{1,1} , O _{1,2} , O _{2,2} , O _{2,3} , O _{1,3} M ₂ , M ₁ , M ₂ , M ₂ , M ₂ , M ₁
4	O _{2,1} , O _{2,2} , O _{2,3} , O _{1,1} , O _{1,2} , O _{1,3} M ₃ , M ₂ , M ₃ , M ₂ , M ₂ , M ₁

In this proposed selection method, the first chromosome is always assigned to the first slot of the roulette wheel. The slot size is proportional to its fitness value. Then the similarity between the first chromosome and the rest of the chromosomes (chromosomes 2, 3, and 4) are calculated. A pair of chromosomes with the highest similarity score is chosen; in this case, the fourth chromosome is selected with a similarity score of 5 as shown in Fig 3. As a result, the fourth chromosome occupies the second slot. The above steps are repeated until all chromosomes are allocated slots in roulette wheel. Next, the overlapping area between the two adjacent chromosomes is determined. The size of the overlapping area is positively correlated with the similarity between the two adjacent chromosomes. Fig 4 shows the fuzzy roulette wheel of the example in Table 2.

In the selection of chromosomes to form a mating pool, first, a random number is generated. Then the membership degree to which a generated random number belongs to each chromosome is calculated. The one with the highest membership degree is chosen. This process continues until a total of Psize chromosomes has been selected.

Chromosome	Solutions
1	O _{1,1} , O _{2,1} , O _{2,2} , O _{2,3} , O _{1,2} , O _{1,3} M ₂ , M ₃ , M ₂ , M ₁ , M ₁ , M ₁
2	O _{1,1} , O _{2,1} , O _{1,2} , O _{1,3} , O _{2,2} , O _{2,3} M ₁ , M ₁ , M ₁ , M ₁ , M ₁ , M ₃
3	O _{2,1} , O _{1,1} , O _{1,2} , O _{2,2} , O _{2,3} , O _{1,3} M ₂ , M ₁ , M ₂ , M ₂ , M ₂ , M ₁
4	O _{2,1} , O _{2,2} , O _{2,3} , O _{1,1} , O _{1,2} , O _{1,3} M ₃ , M ₂ , M ₃ , M ₂ , M ₂ , M ₁

Chromosome	Chromosome	Similarity value (Max score = 12)			Selected pair
		Operation	Machine	Total	
1	2	2	2	4	→ [1,4]
1	3	1	3	4	
1	4	2	3	5	

Fig. 3. The process of assigning chromosomes to slots of a roulette wheel



Fig. 4. Fuzzy roulette wheel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4. Crossover operation

Next, the proposed model employs the crossover operation to improve the quality of the solutions. This paper proposes a new crossover operator, which employs the concept of the hierarchical clustering algorithm. The followings are steps of the proposed crossover operator:

- 1) By using the fuzzy roulette wheel selection, create a mating pool of size Psize, where Psize is the predefined population size.
- 2) Partition the chromosomes in the mating pool into 2 clusters by using the hierarchical clustering algorithm. In this hierarchical clustering algorithm, the single linkage method as defined in eq. (1) is employed to calculate the similarity between two clusters.

$$S(G, H) = \text{MAX}_{e \in G, f \in H} [s(e, f)] \tag{1}$$

where G and H are any two clusters.

s(e,f) denotes the similarity between chromosomes e and f.

- 3) Randomly select one chromosome from each cluster to form a mating pair. Once a crossover point is determined, a one point crossover technique as shown in Fig 6 is used to create new offspring. This third step is repeated (Psize/2) – 1 more times.

For example, in Fig 5, four chromosomes are divided into two clusters. Cluster 1 consists of 3 chromosomes while cluster 2 consists of 1 chromosome. Two mating pairs are formed: 1-2 and 3-2. Fig 6 shows an example of the one point crossover operation which is performed on the mating pair 1-2.

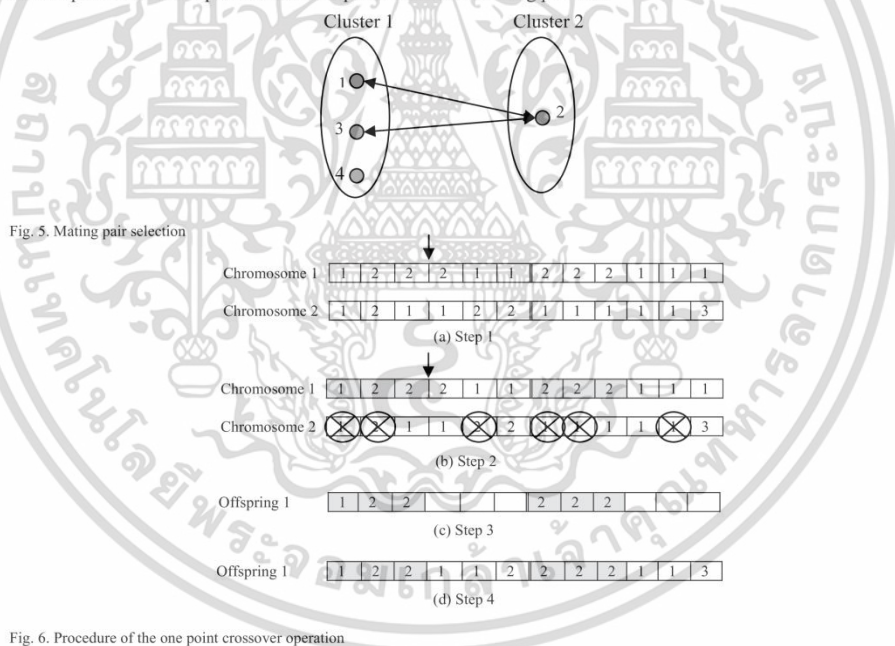


Fig. 6. Procedure of the one point crossover operation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5. Mutation operation

In this paper, the new mutation operator is employed to perform the mutation on the offspring. The mutation operation is used in order to maintain the diversity of the population and to overcome premature convergence. For each offspring q , the following mutation procedure is performed:

- 1) Randomly select two genes in the component l of the chromosome.
- 2) Swap the selected genes.
- 3) Evaluate the fitness of the mutated offspring. If the mutated offspring is fitter than its parent, it will replace its parent in the population. If not, the mutated offspring will be discarded.
- 4) This procedure is repeated T_q times, where T_q is the number of iterations in which the mutation operation is performed on the offspring q .

$$T_q = \max(1, r/x_b - x_q) \quad (2)$$

where r is a random number in the range $[-1,1]$.

x_b is the fitness value of the fittest chromosome in the current generation.

x_q is the fitness value of the chromosome q in the current generation.

3.6. Termination

The best $Psize$ chromosomes are selected from the combined list of current chromosomes and newly generated offspring. Next, the termination criterion is checked. If the criterion is met, the elite chromosome is returned as the best solution found so far. If not, the whole procedure is repeated. The termination criterion used this study is to terminate the iteration of the GA when eighty percent of the population shares the same value.

4. Experimental Results

The proposed algorithm was tested on 10 benchmark problems taken from [12]. The performance of the proposed algorithm is then compared to that of the previous research works by Ho et al. [9], Girish and Jawahar [13, 14], and Xing et al. [15]. The system parameters of the proposed algorithm are set as follows: $Psize = 200$; $Pc = 0.9$; $Pm = 0.3$. The percentage relative error from the best known solution (RE) is used as a measure to evaluate the performance of the proposed method. Table 3 shows the experimental results of the proposed algorithm in comparison to all of the above mentioned methods. The average relative error of the proposed algorithm is the best among the compared methods.

5. Conclusion

This paper proposes a modification to the genetic algorithm for solving the flexible job-shop scheduling problem (FJSP). Three main operations of the genetic algorithm are modified; selection, crossover, and mutation operations. The experimental results show that the performance of the proposed model is significantly improved when these three new operations are used in place of the original operations typically used in the genetic algorithm. A much more complex version of the FJSP, which contains multi-objective, will be studied in the future work.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3. Experimental results

Problems	n x m	LB	Ho et al. [9]	Girish and Jawahar [13]	Girish and Jawahar [14]	Xing et al. [15]	Proposed Model
MK01	10 x 6	36	40 (11.11)	40 (11.11)	40 (11.11)	39 (8.33)	40 (11.11)
MK02	10 x 6	24	29 (20.83)	26 (8.33)	27 (12.50)	29 (20.83)	27 (12.50)
MK03	15 x 8	204	-	204 (0.00)	204 (0.00)	204 (0.00)	204 (0.00)
MK04	15 x 8	48	67 (39.58)	66 (37.50)	62 (29.17)	65 (35.42)	64 (33.33)
MK05	15 x 4	168	176 (4.76)	174 (3.57)	178 (5.95)	173 (2.98)	175 (4.17)
MK06	10 x 15	33	67 (103.03)	77 (133.33)	78 (136.36)	67 (103.03)	65 (96.97)
MK07	20 x 5	133	147 (10.53)	143 (7.52)	147 (10.53)	144 (8.27)	144 (8.27)
MK08	20 x 10	523	523 (0.00)	523 (0.00)	523 (0.00)	523 (0.00)	523 (0.00)
MK09	20 x 10	299	320 (7.02)	328 (9.79)	341 (14.05)	311 (4.01)	309 (3.34)
MK10	20 x 15	165	229 (38.79)	247 (49.70)	252 (52.73)	229 (38.79)	234 (41.82)
Average Relative Error			23.57	26.08	27.24	22.17	21.15

References

1. D. E. Goldberg, Genetic Algorithms in Search Optimisation and Machine Learning, Addison-Wesley, Reading, Massachusetts, 1989.
2. F. Glover, Tabu Search-Part I, Operations Research Society of America, J. Comput., 1989.
3. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, Sci., vol. 220 (1983) 671-680.
4. J. Kennedy and R. Eberhard, Particle Swarm Optimization, Phys. Rev., B.(1995) 13:5344-5348.
5. L. Zadeh, Fuzzy sets, Information and Control, 8(3) (1965) 338-353.
6. F. Pezzella, G. Morganti, G. Ciaschetti, A genetic algorithm for the flexible job-shop scheduling problem, Computers and Operations Research, 35 (10) (2008), 3202-3212.
7. G. H. Zhang, L. Gao, Y. Shi, An effective genetic algorithm for the flexible job-shop scheduling problem, Expert Systems with Applications, 38 (4) (2011), pp. 3563-3573.
8. N. Al-Hinai and T. Y. ElMekkawy, An efficient hybridized genetic algorithm architecture for the flexible job-shop scheduling problem, Flexible Services and Manufacturing Journal, Vol. 23, (2011), pp. 64-85.
9. N. B. Ho, J. C. Tay, and E. M. K. Lai, An effective architecture for learning and evolving flexible job-shop schedules, European Journal of Operational Research, vol.179, no. 2, (2007) 316-333.
10. I. Kacem, S. Hammadi, and P. Borne, Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems, IEEE Transaction on Systems, Man and Cybernetics, (2002) 32(1), 1-13.
11. W. Teekeng and A. Thammano, A combination of Shuffled frog leaping algorithm and fuzzy logic for Flexible Job-Shop Scheduling Problems, Procedia Computer Science, Complex adaptive systems, vol.6, (2011) 69-75
12. M. Mastrolilli, Flexible Job Shop Problem, <http://www.idsia.ch/~monaldo/fjsp.html>.
13. B. S. Girish and N. Jawahar, Scheduling job shop associated with multiple routings with genetic and ant colony heuristics, International Journal of Production Research, (2009), 47(14): 3891-3917.
14. B. S. Girish and N. Jawahar, A particle swarm optimization algorithm for flexible job shop scheduling problem, IEEE International Conference on Automation Science and Engineering, (2009), 298-303.
15. L. N. Xing, Y. W. Chen, P. Wang, Q. S. Zhao, and J. Xiong, A knowledge-based ant colony optimization for flexible Job Shop scheduling problems. Applied Soft Computing, 10, (2010) 888-896.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ – นามสกุล	นางสาววรรณพร ทีเก่ง
วัน เดือน ปีเกิด	24 เมษายน 2515
ที่อยู่	41 ถนนวิเชตวัฒนา ต.แจ้ห่ม อ.แจ้ห่ม จ.ลำปาง 52120
ประวัติการศึกษา	บธ.บ.(ระบบสารสนเทศ) มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี บธ.ม.(บริหารธุรกิจ) มหาวิทยาลัยแม่โจ้ วท.ม.(วิทยาการคอมพิวเตอร์) มหาวิทยาลัยเชียงใหม่
ประสบการณ์ในการทำงาน	
พ.ศ.2537 - ปัจจุบัน	ตำแหน่งอาจารย์ สังกัดสาขาระบบสารสนเทศทางคอมพิวเตอร์ คณะบริหารธุรกิจและศิลปศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ตาก
พ.ศ.2553	ตำแหน่งผู้ช่วยศาสตราจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้