สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รายงานการวิจัย

การจัดตารางการผลิตแบบทำตามสั่งด้วยวิธีการทางวิวัฒนาการ

Job-Shop Scheduling Using Evolutionary Algorithm

รศ. ดร. อาริต ธรรมโน

RCH
TS
157.5
Q655T
สขหมู่.....
เลขทะเบียน.....115596
วัน,เดือน,ปี.....21 มิ.ย. 2554

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณรายได้ ประจำปีงบประมาณ 2553
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

b.12317555
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 พ.ค. 53

ชื่อโครงการ การจัดตารางการผลิตแบบทำตามสั่งด้วยวิธีการทางวิวัฒนาการ

Job-Shop Scheduling Using Evolutionary Algorithm

ได้รับทุนอุดหนุนการวิจัยจากเงินงบประมาณรายได้คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประจำปี 2553 จำนวนเงิน 50,000 บาท
ระยะเวลาทำการวิจัย 1 ปี ตั้งแต่ 1 ตุลาคม 2552 ถึง 30 กันยายน 2553

หน่วยงานและผู้ดำเนินการวิจัยพร้อมหน่วยงานที่สังกัดและเลขหมายโทรศัพท์
รศ. ดร. อาริต ธรรมโน คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โทร. 02-723-4964

บทคัดย่อ

ในงานวิจัยครั้งนี้ได้นำวิธีการที่ในกลุ่มของปัญญาประดิษฐ์ (Artificial Intelligence: AI) โดยได้นำวิธีการจีเนติกอัลกอริทึม (Genetic Algorithm: GA), อัลกอริทึมตาบู (Tabu Search) และฟัซซีลอจิก (Fuzzy Logic) มาผสมผสานเพื่อแก้ปัญหาการจัดตารางการผลิตแบบตามงาน (Job-Shop Scheduling Problem: JSSP) โดยมีวัตถุประสงค์คือเวลารวมในการผลิตน้อยที่สุด (minimize Makespan) ในโมเดลที่นำเสนอนี้ได้นำเอาฟัซซีลอจิกมาใช้ในการหาค่าความเป็นสมาชิกในการเลือกโครโมโซมที่แท้จริงแทนการใช้วงล้อรูเล็ท (roulette wheel) ส่วนวิธีการตาบูได้นำมาใช้ในขั้นตอนการมิวเตชันเพื่อเป็นการหลีกเลี่ยงคำตอบที่เหมาะสมที่สุดเฉพาะพื้นที่ (local optimum) โมเดลที่นำเสนอได้ทดลองใช้กับข้อมูลมาตรฐาน OR-Library และเปรียบเทียบกับโมเดลอื่น พบว่าว่าอัลกอริทึมที่นำเสนอมีประสิทธิภาพดีกว่า

This research presents the novel algorithm for solving a Job-Shop Scheduling Problem: JSSP. An new Artificial Intelligence algorithm based on Genetic Algorithm (GA), Tabu Search algorithm (TS) and Fuzzy Logic were developed. The objective of this scheduling problem is to minimize makespan. This paper proposes Hybrid GA (HGA) which uses a membership function concept in Fuzzy Logic to select chromosomes and uses tabu list in TS to avoid becoming trapped in a local optimum. The proposed model are tested and compared with other models using the standard instances in OR-Library. The superior results validate the effectiveness of the proposed algorithm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

สารบัญ II

สารบัญตาราง IV

สารบัญรูป V

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 2

1.3 ขอบเขตของการวิจัย 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

1.5 ขั้นตอนของการศึกษา 2

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 3

2.1 การจัดตารางการผลิต 3

2.1.1 การจัดตารางการผลิตแบบตามงาน 4

2.1.2 เครื่องมือการวัดประสิทธิภาพของโมเดลการจัดตารางการผลิต 5

2.2 ทฤษฎีพื้นฐานที่เกี่ยวข้องกับงานวิจัย 5

2.2.1 จีเนติกอัลกอริทึม 5

2.2.2 การค้นหาแบบตาบู 8

2.2.3 ฟัซซีลอจิก 8

2.3 งานวิจัยที่เกี่ยวข้อง 14

2.3.1 Local Search Genetic Algorithms for the Job Shop Scheduling Problem 14

2.3.2 A Tabu Search Algorithm for Job Shop Scheduling Problem with Industrial Scheduling Case Study 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 โมเดลที่นำเสนอ 21
3.1 การเตรียมข้อมูลอินพุท 21
3.2 การใช้ฟังก์ชันการเป็นสมาชิก 22
3.3 กระบวนการเรียนรู้ 23

บทที่ 4 การทดลองและผลการทดลอง 25
4.1 ข้อมูลของปัญหาการจัดตารางการผลิตแบบตามงานที่ใช้ทดสอบ 25
4.2 กำหนดค่าพารามิเตอร์ของระบบ 25
4.3 ผลการทดลอง 25

บทที่ 5 สรุปผลงานวิจัยและข้อเสนอแนะ 30
5.1 สรุปผลการวิจัย 30
5.2 ข้อเสนอแนะ 30

บรรณานุกรม 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตัวอย่างของปัญหาประเภท 3 งาน 3 เครื่องจักร 15
2.2 ลำดับงาน และเวลาทำงานของปัญหาประเภท 4 งาน 3 เครื่องจักร 17
3.1 ปัญหาการผลิตแบบตามงานประเภท 3 งาน 3 เครื่องจักร 21
4.1 ผลการทดลองของข้อมูล S. Lawrence (1984) กลุ่ม 1 26
4.2 ผลการทดลองของข้อมูล S. Lawrence (1984) กลุ่ม 2 27
4.3 ผลการทดลองของข้อมูล D. Applegate, W. Cook (1991) 28
4.4 ผลการทดลองของข้อมูล H. Fisher, G.L. Thompson (1963) 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 กระบวนการทำงานของการผลิตแบบตามงาน (JSSP) 4
2.2 ตัวอย่างเวลาในการผลิต....5
2.3 การเลือกโครโมโซมแบบวงล้อรูเล็ท....6
2.4 วิธีการครอสโอเวอร์แบบ POX....7
2.5 ฟังก์ชันการเป็นสมาชิก $\mu_{A\cup B}$ ของเซต $A\cup B$ 9
2.6 ฟังก์ชันการเป็นสมาชิก $\mu_{A\cap B}$ ของเซต $A\cap B$ 10
2.7 ฟังก์ชันการเป็นสมาชิก $\mu_{\overline{A}}$ ของเซต $\overline{A}$ 10
2.8 ฟังก์ชันการเป็นสมาชิกแบบสามเหลี่ยม 11
2.9 ฟังก์ชันการเป็นสมาชิกแบบสี่เหลี่ยมคางหมู 11
2.10 ฟังก์ชันการเป็นสมาชิกแบบเกาซ์เซียน....12
2.11 ฟังก์ชันการเป็นสมาชิกแบบระฆัง....12
2.12 ฟังก์ชันการเป็นสมาชิกแบบตัวเอส 13
2.13 ฟังก์ชันการเป็นสมาชิกแบบตัวแซท....14
2.14 รูปแบบของโครโมโซม ประเภท 3 งาน 3 เครื่องจักร 15
2.15 กราฟการเลือก (Disjunctive Graph) 17
2.16 อัลกอริทึมของวิธีการ first and last neighbours....18
2.17 บล็อกของโหนดที่เป็นชุดคำตอบข้างเคียง....19
2.18 ตาบูลิสแบบพลวัตร 19
2.19 ผลการทดสอบโมเดลที่เสนอกับข้อมูลทดสอบ 29 ชุด 20
3.1 ตัวอย่างโครโมโซมของปัญหา 3 งาน 3 เครื่องจักรของโมเดล HGA....21
3.2 ฟังก์ชันการเป็นสมาชิกของโครโมโซมในโมเดล HGA....22
3.3 ตัวอย่างความคล้ายของโครโมโซม 23
3.4 ตัวอย่างการมิวเตชันโครโมโซม....24
3.5 แผนผังแสดงขั้นตอนของโมเดลที่นำเสนอ....24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การนำหลักการของปัญญาประดิษฐ์มาใช้ในการสร้างความฉลาดให้กับสิ่งที่ไม่มีชีวิตให้มีการเรียนรู้เกี่ยวกับกระบวนการคิด การกระทำ การให้เหตุผล และการทำงานของสมองมนุษย์ เพื่อใช้แก้ปัญหา ตัดสินใจ ได้อย่างถูกต้อง เป็นสิ่งที่ได้รับความสนในและพัฒนามาอย่างต่อเนื่องโดยนำมาประยุกต์ใช้กับงานด้านต่างๆ รวมทั้งงานทางด้านการวิจัยการดำเนินงาน (Operation Research) ที่นิยมนำเอาหลักการของปัญญาประดิษฐ์มาใช้แก้ปัญหา ได้แก่ ปัญหาการจัดตารางผลิต (Scheduling problem) ปัญหาการมอบหมายงาน (Assignment problem) และปัญหาด้านพนักงานขาย (Traveling Salesman problem) เป็นต้น ซึ่งล้วนแต่เป็นปัญหาที่ต้องการหาคำตอบที่อยู่ภายใต้ข้อจำกัดของทรัพยากร เช่น ค่าต่ำสุด ค่าสูงสุด และค่าที่เหมาะสมที่สุด เป็นต้น ซึ่งปัญหาประเภทนี้เป็นปัญหาประเภท NP-Hard แบบ Combinatorial Optimization ซึ่งหมายถึง ปัญหาที่ใช้เวลาหาคำตอบยาวนานโดยเมื่อขนาดของปัญหาเพิ่มขึ้นเพียงเล็กน้อย ก็จะต้องใช้เวลาในการแก้ปัญหานานมากขึ้นหลายเท่าตัว ซึ่งการใช้หลักการคณิตศาสตร์ เช่น Integer linear programming หรือ Branch and bound algorithm ไม่สามารถแก้ปัญหาเพื่อให้ได้คำตอบที่เหมาะสมได้ ดังนั้นปัญหาในการหาค่าเหมาะสมที่สุด (Optimization Algorithm) ของปัญหาจึงนิยมใช้หลักการประมาณ (Approximation optimization algorithm) เพื่อหาคำตอบที่เหมาะสมแทน

วิธีการเมตา-ฮิวริสติก (Meta-heuristics) เป็นวิธีการที่สามารถหาคำตอบที่ดีที่สุดได้โดยอาศัยหลักการประมาณ และมีความรวดเร็วในการประมวลผลและหาคำตอบที่ซับซ้อนได้ดี เช่น จีเนติกอัลกอริทึม ซึ่งเป็นอัลกอริทึมที่เลียนแบบวิวัฒนาการทางพันธุกรรมของสิ่งมีชีวิต อัลกอริทึมตาบู (Tabu Search) เป็นอัลกอริทึมการค้นหาแบบมีข้อห้าม และอัลกอริทึมการอบอ่อนจำลอง (Simulated Annealing) เป็นอัลกอริทึมที่ใช้แนวคิดของการใช้อุณหภูมิเพื่อหลอมโลหะให้อยู่ในสภาวะเหมาะสมที่สุด เป็นต้น

การศึกษาในครั้งนี้จึงมุ่งเน้นการประยุกต์ใช้อัลกอริทึมทางเมตา-ฮิวริสติก ในการค้นหาคำตอบที่เหมาะสมที่สุด มาประยุกต์ใช้ในกับงานการจัดตารางการผลิตแบบตามงาน (Job-Shop Scheduling Problem: JSSP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

ศึกษาและพัฒนาโมเดลใหม่ในการจัดตารางการผลิตตามงาน โดยใช้จีเนติกอัลกอริทึม อัลกอริทึมตาบู และฟัซซีลอจิก

1.3 ขอบเขตของงานวิจัย

1. งานวิจัยนี้นำเสนอโมเดลการแก้ปัญหาการจัดตารางการผลิตแบบตามงาน
2. งานวิจัยนี้ทดลองในระบบแบบออฟไลน์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้ความเข้าใจในการผสมผสานโมเดลระหว่างจีเนติกอัลกอริทึม ร่วมอัลกอริทึมตาบู และฟัซซีลอจิก
2. ได้องค์ความรู้ใหม่สำหรับงานวิจัยด้านการจัดตารางการผลิตแบบตามงาน
3. เป็นพื้นฐานสำหรับการประยุกต์ใช้แนวคิดทางด้านปัญญาประดิษฐ์มาใช้ในการจัดตารางการผลิตแบบตามงาน

1.5 ขั้นตอนของการดำเนินงานวิจัย

1. ศึกษาเทคนิคการจัดตารางการทำงานที่ใช้ปัญญาประดิษฐ์
2. ศึกษาทฤษฎีที่เกี่ยวข้องกับการพัฒนาโมเดลการหาคำตอบแบบเมตา-ฮิวริสติก
3. ออกแบบโมเดลที่เกิดจากการผสมผสานระหว่างขั้นตอนวิธีเชิงพันธุกรรม ร่วมกับอัลกอริทึมตาบู และฟัซซีลอจิก
4. พัฒนาโปรแกรมตามรูปแบบโมเดลที่ออกแบบไว้ โดยใช้โปรแกรม MATLAB
5. จัดเตรียมข้อมูลสำหรับการทดลองวิจัยแล้วดำเนินการทดลองเพื่อวัดประสิทธิภาพของโมเดลที่นำเสนอ
6. สรุปผลการทดลอง
7. จัดทำรายงานการวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะแบ่งเนื้อหาออกเป็น 3 ส่วน โดยส่วนแรกจะกล่าวถึงการจัดตารางการผลิต ส่วนที่ 2 อธิบายถึงทฤษฎีพื้นฐานที่เกี่ยวข้องกับงานวิจัยนี้ และส่วนที่ 3 เป็นรายละเอียดของงานวิจัยที่เกี่ยวข้อง

2.1 การจัดตารางการผลิต

การจัดตารางการผลิต (Scheduling) [1] หมายถึง การจัดสรรทรัพยากร (Resource) ที่มีอยู่อย่างจำกัดให้กับภารกิจ (Task) จำนวนหนึ่งภายใต้ระยะเวลาที่กำหนดให้ เพื่อให้สามารถบรรลุเป้าหมาย (Goal) หรือวัตถุประสงค์ (Objective) สูงสุดตามที่กำหนดไว้ ในการจัดตารางการผลิตประกอบด้วยงาน (Job) ในแต่ละงานมีองค์ประกอบย่อยที่มีความสัมพันธ์เป็นลำดับก่อนหลังซึ่งเรียกว่าโอเปอร์เรชัน (Operation) ซึ่งมีคำจำกัดความในการจัดตารางการผลิต ดังนี้

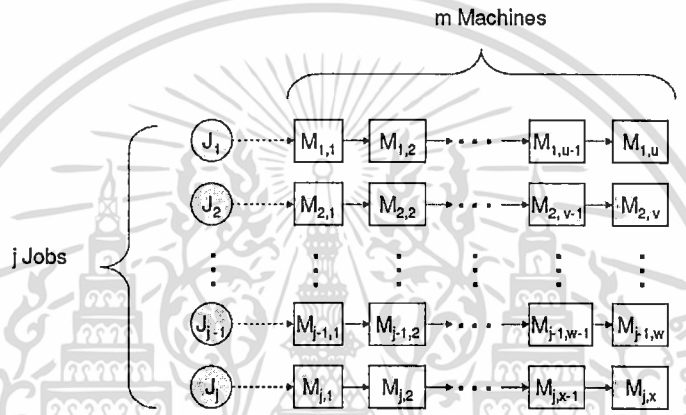
- งาน (Job): กำหนดให้งานมีจำนวน n งาน และในงาน 1 งานจะประกอบไปด้วยหลายๆ งานย่อยหรือที่เราเรียกกันว่าโอเปอร์เรชัน โดยเซตของงานจะประกอบด้วย $J = \{j_1, j_2, ..., j_n\}$
- โอเปอร์เรชัน (Operation): ในแต่ละงานประกอบด้วยลำดับของโอเปอร์เรชัน ดังนั้นเซตของโอเปอร์เรชันจะประกอบด้วย $O = \{O_1, O_2, ..., O_n\}$
- เครื่องจักร (Machine): กำหนดให้มีเครื่องจักรจำนวน m เครื่องจักร และเครื่องจักร 1 เครื่องจักรสามารถทำงานได้มากกว่า 1 งาน เซตของเครื่องจักรประกอบด้วย $M = \{m_1, m_2, ..., m_m\}$
- เวลาทำงาน (Processing Time): ระยะเวลาในการทำงานของแต่ละโอเปอร์เรชัน $\tau = \{\tau_{i1}, \tau_{i2}, ..., \tau_{im_i}\}$
- เวลาเสร็จของงาน (Completion Time): เวลาทำงานเสร็จของงานทุกงาน

ลักษณะของปัญหาที่นิยมนำมาจัดตารางการผลิต สามารถแบ่งตามลักษณะการทำงานของเครื่องจักร ได้เป็น 4 ประเภท คือ 1.การผลิตแบบเครื่องจักรเดี่ยว (Single Machine) 2. การผลิตด้วยเครื่องจักรแบบขนาน (Parallel Machines) 3. การผลิตด้วยเครื่องจักรแบบไหลเลื่อน (Flow Shop) และ 4. การผลิตด้วยเครื่องจักรแบบตามงาน (Job-Shop) ซึ่งในการศึกษาครั้งนี้มุ่งศึกษาปัญหาการจัดตารางการผลิตแบบตามงาน (Job-Shop Scheduling Problem) ซึ่งมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 การจัดตารางการผลิตแบบตามงาน (Job-Shop Scheduling Problem: JSSP)

ปัญหาการจัดตารางการผลิตแบบตามงาน หรือเรียกว่าปัญหาการจัดตารางผลิตแบบคลาสสิก (Classical Job-Shop Scheduling) เป็นลักษณะการผลิตที่ประกอบด้วยงานหลายงาน และมีการทำงานบนเครื่องจักรหลายเครื่อง โดยแต่ละงานมีเส้นทางการไหลของตัวเองตามที่ผู้ผลิตกำหนดให้เท่านั้น เครื่องจักรที่ทำงานในระบบการผลิตนี้มีหน้าที่ผลิตสินค้าที่มีลักษณะหลากหลายตามความต้องการของลูกค้า เช่น การรับจ้างสกรีนลวดลายบนเสื้อผ้า กระเป๋า เป็นต้น รูปที่ 2.1 แสดงตัวอย่างกระบวนการผลิตแบบตามงาน [2]



รูปที่ 2.1 กระบวนการทำงานของการผลิตแบบตามงาน (JSSP)

สมมติฐานเกี่ยวกับงาน

- งานทั้งหมดมีความพร้อมทำที่เวลาเริ่มต้นของการจัดตาราง
- ไม่สามารถแยกโอเปอร์เรชันของงานให้ทำบนเครื่องจักรมากกว่า 1 เครื่องได้ในเวลาเดียวกัน
- ทราบเวลาดำเนินการของแต่ละงานและให้มีค่าคงที่
- เวลาปรับตั้งเครื่องจักรและเวลาขนย้ายเป็นอิสระจากงาน ลำดับงาน และเวลาเหล่านี้ถูกรวมไว้ในเวลาดำเนินการแล้ว
- ไม่สามารถแทรกงานที่อยู่ระหว่างทำ และทุกงานมีความสำคัญเท่ากัน

สมมติฐานเกี่ยวกับเครื่องจักร

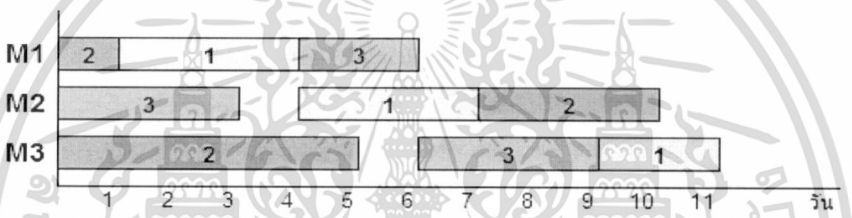
- เครื่องจักรทั้ง m เครื่องมีความพร้อมที่จะเริ่มต้นทำงานทันที ณ เวลาเริ่มต้นของการวางแผน
- โอเปอร์เรชันของงานแต่ละงานสามารถทำบนเครื่องจักรเพียงเครื่องเดียวในเวลาหนึ่งๆ
- เครื่องจักรไม่มีการเสียระหว่างทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 เครื่องมือในการวัดประสิทธิภาพของโมเดลการจัดตารางการผลิต

วิธีการวัดและประเมินประสิทธิภาพของโมเดลการจัดตารางการผลิต สามารถประเมินได้หลายวิธีขึ้นอยู่กับลักษณะการผลิต เช่น ด้านปริมาณการผลิต ด้านกำหนดวันส่งมอบ และด้านค่าใช้จ่าย เป็นต้น ในโมเดลงานวิจัยนี้ได้ใช้การประเมินจากเวลาที่ทำงานทั้งหมดเสร็จสิ้น หรือเวลาในการผลิตรวมที่น้อยที่สุด (minimize makespan) ตามสมการที่ 2.1 เมื่อ $C_1, C_2, ..., C_n$ คือ ชุดคำตอบที่เป็นไปได้ในการจัดตารางการผลิต ซึ่งจะเลือกคำตอบที่ใช้เวลาในการผลิตน้อยที่สุด ตัวอย่างคำตอบที่เป็นไปได้ของเวลาที่ใช้ในการผลิต สามารถแสดงได้ตามรูปที่ 2.2

$$Z = f(C_1, C_2, ..., C_n) \tag{2.1}$$



รูปที่ 2.2 ตัวอย่างเวลาในการผลิต ($C_i = 11$)

2.2 ทฤษฎีพื้นฐานที่เกี่ยวข้องกับงานวิจัย

ในงานวิจัยครั้งนี้ได้มีการศึกษาทฤษฎีพื้นฐานที่เกี่ยวข้อง 3 ส่วน คือ จีเนติกอัลกอริทึม อัลกอริทึมตาบู และฟัซซีลอจิก

2.2.1 จีเนติกอัลกอริทึม

จีเนติกอัลกอริทึม (Genetic Algorithm: GA) คิดค้นโดย John Holland ในปี ค.ศ. 1975 โดยศึกษาทฤษฎีของการวิวัฒนาการทางธรรมชาติ และทฤษฎีของ Darwin ที่ศึกษาทฤษฎีวิวัฒนาการทางพันธุกรรมที่สามารถถ่ายทอดลักษณะต่างๆ ไปสู่รุ่นต่อไป เช่น การสลับสายพันธุ์ (Crossover) หรือการกลายพันธุ์ (Mutation) เป็นต้น ผลลัพธ์ที่ได้คือโครโมโซมลูกซึ่งอาจจะดีขึ้นหรือแย่ลง หลังจากนั้นกระบวนการธรรมชาติจะทำการคัดเลือกโครโมโซม ซึ่งโครโมโซมที่ดีจะมีโอกาสสูงที่จะถูกเลือกให้มีชีวิตต่อไป ดังนั้นโครโมโซมลูกหลานจึงมีแนวโน้มที่จะเป็นโครโมโซมที่ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของจีเนติกอัลกอริทึม มีขั้นตอนดังนี้

1) ขั้นตอนการเตรียมการ ประกอบด้วย

1.1) กำหนดฟังก์ชันเป้าหมาย (Objective Function) เพื่อใช้ในการหาค่าความเหมาะสม (fitness) ของโครโมโซม

1.2) กำหนดรูปแบบโครโมโซมโดยการเข้ารหัส (Coding) และการแปลงตัวแปรเพื่อให้อยู่ในรูปแบบโครโมโซมตามที่กำหนด เช่น เลขจำนวนจริง ตัวอักษร หรือเลขฐานสอง เป็นต้น โดยในหนึ่งโครโมโซมจะประกอบด้วยหลายๆ ยีน ซึ่งจะประกอบกันเป็น วิธีการแก้ปัญหา 1 วิธีหรือหนึ่งคำตอบ

2) ขั้นตอนการดำเนินการ ประกอบด้วย

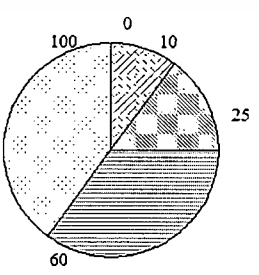
2.1) สร้างประชากรหรือโครโมโซมเริ่มต้นจำนวน N โครโมโซม โดยใช้วิธีการสุ่มค่าที่เป็นไปได้ให้กับแต่ละยีนของโครโมโซมทีละยีน จนครบจำนวนที่ต้องการ

2.2) คำนวณค่าความเหมาะสมของประชากรโครโมโซมทั้งหมดจากฟังก์ชันเป้าหมาย

2.3) คัดเลือกโครโมโซม (Selection) ในขั้นตอนที่จะทำการคัดเลือกโครโมโซมที่ดีโดยดูจากค่าความเหมาะสมของแต่ละโครโมโซม เพื่อนำมาเป็นโครโมโซมต้นฉบับ ในการสร้างโครโมโซมรุ่นต่อไป

วิธีการคัดเลือกที่ใช้ในเจเนติกอัลกอริทึมอย่างง่าย คือ การคัดเลือกแบบวงล้อรูเล็ท (Roulette wheel) ดังรูปที่ 2.3 ซึ่งวงล้อจะถูกสร้างจากค่าความเหมาะสมของแต่ละโครโมโซมมาคิดเป็นเปอร์เซ็นต์ และเปอร์เซ็นต์สะสม แล้วนำเปอร์เซ็นต์สะสมของโครโมโซมตั้งแต่ตัวแรกถึงตัวสุดท้ายมาเรียงต่อกันในวงล้อรูเล็ท การเลือกโครโมโซมทำได้โดยสุ่มตัวเลข มีค่าระหว่าง [0-1] เพื่อเลือกโครโมโซมในวงล้อเป็นโครโมโซมต้นแบบ (พ่อ-แม่)

โครโมโซม	ฟิตเนส(%)	% สะสม
1	10	10
2	15	25
3	35	60
4	40	100



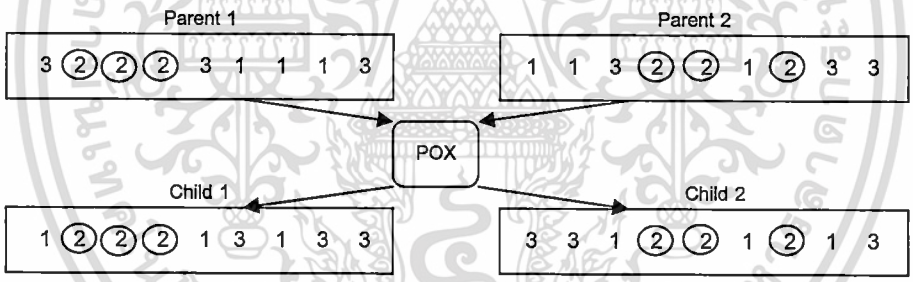
รูปที่ 2.3 การเลือกโครโมโซมแบบวงล้อรูเล็ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4) การครอสโอเวอร์ (Crossover) คือ การนำโครโมโซมที่ได้รับการคัดเลือกเป็นโครโมโซมต้นแบบและนำมาสลับค่าตามตำแหน่งที่สุ่มในโครโมโซม เพื่อสร้างโครโมโซมลูก (Offspring) จำนวน 2 โครโมโซมเพื่อเป็นประชากรในรุ่นต่อไป วิธีการครอสโอเวอร์มีหลายวิธี เช่น การครอสโอเวอร์หนึ่งตำแหน่ง (one point crossover), การครอสโอเวอร์สองตำแหน่ง (two point crossover), การครอสโอเวอร์หลายตำแหน่ง (multiple point crossover), และวิธีการครอสโอเวอร์แบบ POX เป็นต้น

วิธีการครอสโอเวอร์แบบ POX (Precedence Operation Crossover) มีขั้นตอนดังนี้

- สุ่มหมายเลขงาน (1, ..., n)
- คัดลอกหมายเลขงานที่สุ่มได้จาก P1 และ P2 ไปยังตำแหน่งเดียวกันใน C1 และ C2 ตามลำดับ
- คัดลอกหมายเลขงานที่เหลือจาก P1 ไป C2 และจาก P2 ไป C1 ตามลำดับ



รูปที่ 2.4 วิธีการครอสโอเวอร์แบบ POX

2.5) การมิวเตชัน (Mutation) คือ การเปลี่ยนค่ายีนในแต่ละโครโมโซมของลูกที่ได้จากการครอสโอเวอร์โครโมโซมต้นแบบ ซึ่งในการมิวเตชัน จะทำการสุ่มค่าที่อยู่ระหว่าง 0 ถึง 1 และนำค่าที่สุ่มเปรียบเทียบกับค่าความน่าจะเป็นในการมิวเตชัน (Pm) ที่ได้กำหนดไว้ หากค่าที่สุ่มต่ำกว่า Pm จะทำการมิวเตชันโครโมโซมนั้น

2.6) ตรวจสอบการจบทำงานของระบบ ถ้ายังไม่พบคำตอบที่พอใจให้กลับไปทำขั้นตอนที่ 2.2 โดยทั่วไปแล้วเงื่อนไขการหยุดการทำงานของจีเนติกอัลกอริทึมคือเมื่อพบคำตอบที่พอใจ หรือครบจำนวนรอบตามที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การค้นหาแบบตาบู

การค้นหาแบบตาบู หรืออัลกอริทึมตาบู (Tabu Search: TS) คิดค้นโดย Glover ที่เสนอแนวคิดของ TS ไว้ในปี พ.ศ. 2520 คำว่า "tabu" มีความหมายว่า "ต้องห้าม" เพื่อห้ามเลือกคำตอบเดิมที่เคยพบมาเมื่อเร็วๆ นี้ แล้วให้ทำการค้นหาพื้นที่อื่นที่อาจนำไปสู่การค้นหาคำตอบที่เหมาะสมที่สุด หลักการนี้จะช่วยให้สามารถหลีกเลี่ยงคำตอบที่เป็นค่าเหมาะสมที่สุดเฉพาะพื้นที่ (local optimum) และสามารถค้นหาคำตอบให้ใกล้เคียงกับคำตอบที่เหมาะที่สุดแบบวงกว้าง (global optimum) ได้

หลักการทำงานของอัลกอริทึมตาบู

1) กำหนดตัวแทนของคำตอบเริ่มต้น (initial solution) ด้วยวิธีการทางฮิวริสติก เช่น เลือกเส้นทางที่ใช้เวลาน้อยที่สุด (short processing time: SPT) เป็นต้น ซึ่งคำตอบดังกล่าวต้องมีความเป็นไปได้ หลังจากนั้นจะทำการหาค่าความเหมาะสม (fitness) ของคำตอบนั้น และกำหนดให้เป็นคำตอบของการทำงานในรอบแรก

2) ในรอบต่อไปให้ สร้างเซตชุดคำตอบข้างเคียงของคำตอบจากขั้นตอนที่ 1 และหาค่าความเหมาะสมของชุดคำตอบข้างเคียง (ชุดคำตอบข้างเคียง หมายถึง คำตอบข้างเคียงที่เป็นไปได้จากคำตอบรอบปัจจุบันซึ่งอาจใช้วิธีการสลับตำแหน่งคุณสมบัติในองค์ประกอบของคำตอบ ซึ่งจะขึ้นอยู่กับลักษณะของปัญหา)

3) เลือกคำตอบที่ดีที่สุดของชุดคำตอบข้างเคียงเป็นคำตอบในรอบปัจจุบัน และทำการเก็บคำตอบที่เลือกไว้ในตาบูลิส (tabu list) ซึ่งเป็นสถานะต้องห้าม ซึ่งจะทำให้คำตอบนี้ไม่ถูกเลือกอีกภายในระยะเวลาที่กำหนด (tabu tenure) แต่มีข้อยกเว้นถ้าค่าคำตอบในรอบนี้ดีกว่าที่เคยพบมา สามารถเลือกคุณสมบัตินี้ซ้ำได้ถึงแม้จะมีสถานะต้องห้ามก็ตาม หลักการนี้เรียกว่าเกณฑ์ความปรารถนา (Aspiration criterion)

4) เมื่อยังไม่พบคำตอบที่ต้องการให้กลับไปทำขั้นตอนที่ 2 และหยุดการทำงานเมื่อครบตามเงื่อนไขที่กำหนด เช่น พบคำตอบที่พอใจ หรือครบตามจำนวนรอบที่กำหนด เป็นต้น

2.2.3 ฟัซซีลอจิก

ฟัซซีลอจิก (Fuzzy Logic) หรือตรรกะแบบคลุมเครือ ซึ่งเป็นคณิตศาสตร์แขนงหนึ่งที่สนับสนุนการตัดสินใจภายใต้ข้อมูลที่มีความไม่แน่นอน คุณสมบัติพิเศษของฟัซซีลอกจิกที่แตกต่างจากบลูลีนลอจิก คือ ค่าความจริงจะไม่ได้ถูกจำกัดแค่จริงหรือเท็จ แต่จะมีค่าความจริงอยู่ระหว่างจริงและเท็จด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ฟัซซีเซต

ฟัซซีเซต (Fuzzy Set) ประกอบด้วยคู่ลำดับของ x และ $\mu_A(x)$ โดย x คือสมาชิกของเซต A และ $\mu_A(x)$ คือค่าระดับความเป็นสมาชิกของ x ในเซต A โดยค่าของ $\mu_A(x)$ จะหาได้จากฟังก์ชันสมาชิก (Membership function) ของเซต A ซึ่งจะเป็นค่าต่อเนื่องตั้งแต่ 0 ถึง 1 ดังสมการ (2.2)

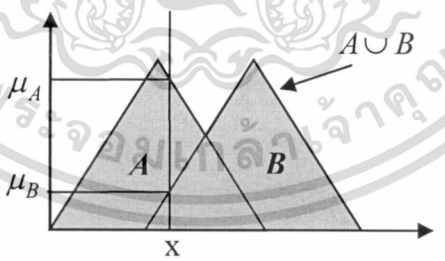
$$\mu_A(x) = X \rightarrow [0,1] \tag{2.2}$$

2) โอเปอร์เรชันของฟัซซีเซต

โอเปอร์เรชันพื้นฐานของฟัซซีเซต ได้แก่ ยูเนียน (union) อินเตอร์เซกชัน (intersection) และคอมพลีเมนต์ (complement) ซึ่งมีรายละเอียดดังนี้

2.1) ยูเนียน (Union) ของฟัซซีเซต จะเป็น OR operation ถ้ากำหนดให้ A และ B เป็นเซตใน X โดยมีค่าความเป็นสมาชิก μ_A และ μ_B ตามลำดับ การหาค่าความเป็นสมาชิกของฟัซซีเซต A ยูเนียนกับฟัซซีเซต B คำนวณได้จากการเลือกค่าความเป็นสมาชิกที่มากที่สุด ระหว่าง μ_A และ μ_B ดังสมการที่ 2.3

$$\begin{aligned}\mu_{A\cup B}(x) &= \mu_A(x) \vee \mu_A(x) \\ &= \max(\mu_A(x), \mu_A(x))\end{aligned} \tag{2.3}$$

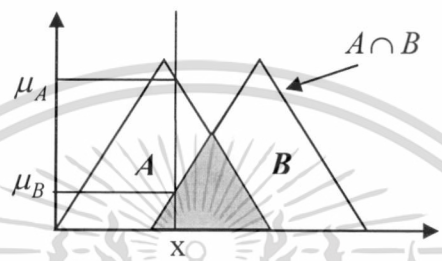


รูปที่ 2.5 ฟังก์ชันการเป็นสมาชิก $\mu_{A\cup B}$ ของเซต $A \cup B$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2) อินเตอร์เซกชัน (Intersection) ของฟัซซีเซต จะเป็น AND operation ดังสมการที่ 2.4

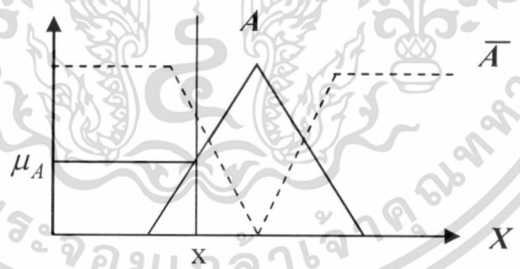
$$\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_A(x) \qquad (2.4)$$
$$= \min(\mu_A(x), \mu_A(x))$$



รูปที่ 2.6 ฟังก์ชันการเป็นสมาชิก $\mu_{A\cap B}$ ของเซต $A \cap B$

2.3) คอมพลีเมนต์ (Complement) ของฟัซซีเซต คือผลต่างของ 1 กับ μ_A ในสมการที่ 2.5

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \qquad (2.5)$$



รูปที่ 2.7 ฟังก์ชันการเป็นสมาชิก $\mu_{\overline{A}}$ ของเซต $\overline{A}$

3) ฟังก์ชันความเป็นสมาชิก

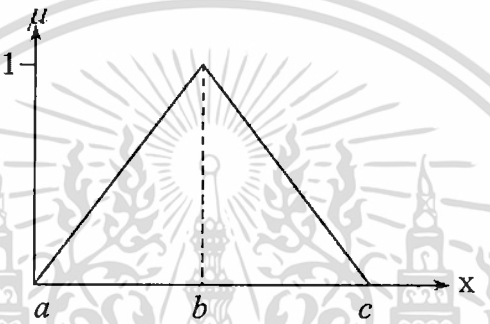
ฟังก์ชันความเป็นสมาชิก (membership function) เป็นฟังก์ชันที่มีการกำหนดระดับความเป็นสมาชิกของตัวแปรที่ต้องการใช้งาน ชนิดของฟังก์ชันความเป็นสมาชิกที่ใช้งานทั่วไปมีหลายชนิด แต่ในที่นี้จะกล่าวถึง 6 ชนิดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1) ฟังก์ชันสามเหลี่ยม (triangular membership function)

ฟังก์ชันสามเหลี่ยมมีทั้งหมด 3 พารามิเตอร์คือ $\{a, b, c\}$

$$triangular(x:a,b,c)=\begin{cases}0 & x<a\\ (x-a)/(b-a) & a\le x\le b\\ (c-x)/(c-b) & b\le x\le c\\ 0 & x>c\end{cases} \tag{2.6}$$

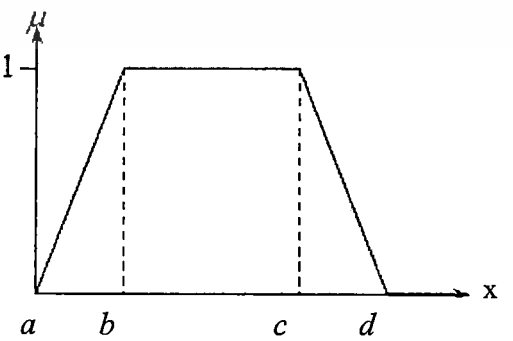


รูปที่ 2.8 ฟังก์ชันการเป็นสมาชิกแบบสามเหลี่ยม

3.2) ฟังก์ชันสี่เหลี่ยมคางหมู (trapezoidal membership function)

ฟังก์ชันสี่เหลี่ยมคางหมูมีทั้งหมด 4 พารามิเตอร์คือ $\{a, b, c, d\}$

$$trapezoidal(x:a,b,c,d)=\begin{cases}0 & x<a\\ (x-a)/(b-a) & a\le x<b\\ 1 & b\le x<c\\ (d-x)/(d-c) & c\le x<d\\ 0 & x\ge d\end{cases} \tag{2.7}$$



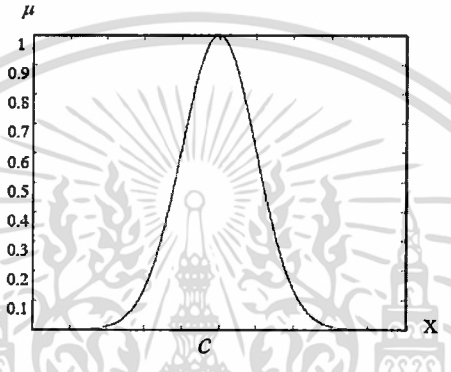
รูปที่ 2.9 ฟังก์ชันการเป็นสมาชิกแบบสี่เหลี่ยมคางหมู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3) ฟังก์ชันเกาส์เซียน (Gaussian membership function)

ฟังก์ชันเกาส์เซียนมีทั้งหมด 2 พารามิเตอร์คือ $\{m, \sigma\}$ ซึ่ง m หมายถึง ค่าเฉลี่ย และ σ หมายถึง ค่าเบี่ยงเบนมาตรฐาน

$$guassian(x:\sigma,c)=\exp\left(-\frac{1}{2}\left(\frac{(x-c)}{\sigma}\right)^2\right) \tag{2.8}$$

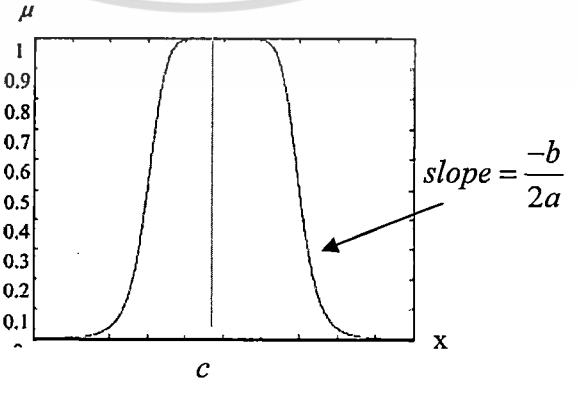


รูปที่ 2.10 ฟังก์ชันการเป็นสมาชิกแบบเกาซ์เซียน

3.4) ฟังก์ชันระฆังคว่ำ (Bell-shaped membership function)

ฟังก์ชันรูประฆังคว่ำมีพารามิเตอร์ทั้งหมด 3 ค่าคือ $\{a, b, c\}$

$$bell-shaped(x:a,b,c)=\left(\frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}\right) \tag{2.9}$$



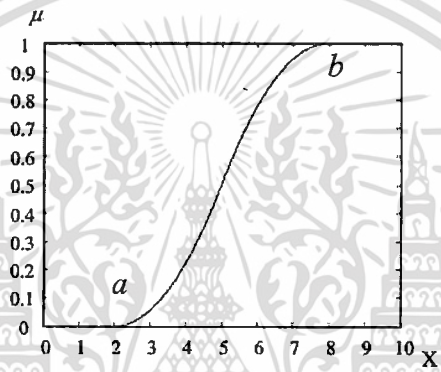
รูปที่ 2.11 ฟังก์ชันการเป็นสมาชิกแบบระฆัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5) ฟังก์ชันตัวเอส (Smooth Membership Function)

ฟังก์ชันรูปตัวเอสมีพารามิเตอร์ทั้งหมด 2 ค่าคือ $\{a, b\}$

$$S(x:a,b)=\begin{cases}0 & x<a\\ 2\left(\dfrac{x-b}{b-a}\right)^2 & a\le x\le\dfrac{a+b}{2}\\ 1-2\left(\dfrac{x-b}{b-a}\right)^2 & \dfrac{a+b}{2}\le x<b\\ 1 & x\ge b\end{cases} \tag{2.10}$$



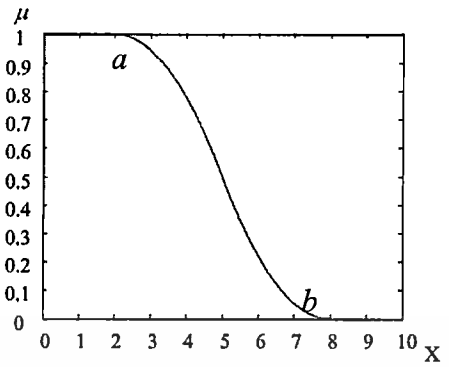
รูปที่ 2.12 ฟังก์ชันการเป็นสมาชิกแบบตัวเอส

3.6) ฟังก์ชันตัวแซด (Z-membership function)

ฟังก์ชันรูปตัวเอสมีพารามิเตอร์ทั้งหมด 2 ค่าคือ $\{a, b\}$

$$Z(x:a,b)=\begin{cases}1 & x<a\\ 1-2\left(\dfrac{x-b}{b-a}\right)^2 & a\le x\le\dfrac{a+b}{2}\\ 2\left(\dfrac{x-b}{b-a}\right)^2 & \dfrac{a+b}{2}\le x<b\\ 0 & x\ge b\end{cases} \tag{2.11}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ฟังก์ชันการเป็นสมาชิกแบบตัวแซท

การเลือกฟังก์ชันการเป็นสมาชิก จะต้องเลือกตามความเหมาะสมและความครอบคลุมของข้อมูลที่จะรับเข้ามา โดยสามารถที่จะทับซ้อนกันเพื่อให้การดำเนินงานราบเรียบ และฟังก์ชันความเป็นสมาชิกสามารถเปลี่ยนแปลงแก้ไขให้เหมาะกับงานที่กำลังปฏิบัติงานหรือตามความต้องการได้

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 Local Search Genetic Algorithms for the Job Shop Scheduling Problem เขียนโดย BEARICE M. OMBUKI AND MARIO VENTRESCA [3]

งานวิจัยนี้นำเสนอโมเดล LSGA และ Hybrid GA สำหรับปัญหาการจัดตารางการผลิตแบบตามงาน (JSSP) โดยมีวัตถุประสงค์ คือ เวลาในการผลิตรวมน้อยที่สุด (minimize makespan) โดยใช้หลักการของจีเนติกอัลกอริทึม ร่วมกับอัลกอริทึมตาบูมาเป็นโมเดลในการจัดตารางการผลิต ข้อมูลที่ใช้ทดสอบประสิทธิภาพของโมเดล คือ ข้อมูลมาตรฐานใน OR-Library [6] ซึ่งทั้ง 2 โมเดลที่เสนอนำใช้กับชุดข้อมูลที่มีลักษณะทั่วไปและข้อมูลที่มีลักษณะเฉพาะ เพื่อสามารถหาค่าความเหมาะสมดีที่สุด ซึ่งกระบวนการทำงานของโมเดล LSGA คือขั้นตอนที่ 1 – 7 และโมเดล Hybrid GA คือ กระบวนการตามโมเดล LSGA และเพิ่มวิธีการของตาบูในขั้นตอนที่ 8 มีรายละเอียดดังนี้

1) กำหนดพารามิเตอร์ของจีเนติกอัลกอริทึม ดังนี้

- จำนวนประชากร (Pop size = 200)
- จำนวนรอบทำงาน (Gen = 550)
- อัตราการครอสโอเวอร์ (Pm = 0.95)
- ความน่าจะเป็นในการมิวเตชัน (0.15)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) สร้างประชากรเริ่มต้น โดยโครโมโซมจะมีความยาวเท่ากับ $n \times m$ ตัวอย่างเช่นข้อมูลปัญหาประเภท 3 งาน 3 เครื่องจักร ดังตัวอย่างในตารางที่ 2.1 ในแต่ละโครโมโซมจะมี 9 ยีน ในแต่ละยีนจะประกอบด้วยหมายเลขงาน และหมายเลขเครื่องจักร ตามรูปที่ 2.14

ตารางที่ 2.1 ตัวอย่างของปัญหาประเภท 3 งาน 3 เครื่องจักร

Job	(m,t)	(m,t)	(m,t)
Job1	3,1	1,3	2,6
Job2	2,8	3,5	1,10
Job3	2,3	1,8	3,1

1,3	2,2	3,2	1,1	2,3	3,1	1,2	2,1	3,3

รูปที่ 2.14 รูปแบบของโครโมโซม ประเภท 3 งาน 3 เครื่องจักร

3) หาค่าความเหมาะสม (fitness) ของแต่ละโครโมโซม

$$F(x) = (\max_length - T(x)) + small_int \qquad (2.12)$$

โดยที่

max_length = ค่าความเหมาะสมของโครโมโซมที่มากที่สุดของรอบ

T(x) = ค่าความเหมาะสมของโครโมโซมปัจจุบัน

small_int = ค่าความเหมาะสมของโครโมโซมที่น้อยที่สุดของรอบ

4) เลือกโครโมโซมต้นแบบ (P1, P2) เพื่อสร้างทายาท (C1, C2)

5) ทำการครอสโอเวอร์ ด้วยวิธี POX method มีขั้นตอนดังนี้

5.1) เมื่อเลือกโครโมโซมต้นแบบ (P1, P2) ตามขั้นตอนที่ 4

5.2) ให้สุ่มหมายเลขงาน i ซึ่งประกอบด้วยงาน 1, 2, …, n สำหรับ P1

5.3) คัดลอกทุกโอเปอร์เรชันที่เป็นของงาน i บน P1 ไปที่ C1 ที่ตำแหน่งเดียวกัน และคัดลอกทุกโอเปอร์เรชันที่เหลือ (ยกเว้นงาน i) ของ P2 ไปที่ C1

5.4) สำหรับ C2 ให้ทำเช่นเดียวกันกับ C1 (ตามขั้นตอนที่ 5.2 – 5.3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) การมิวเตชันโครโมโซมทายาท โดยวิธีการปรับปรุงเฉพาะพื้นที่ (local search) ด้วยการกำหนดค่าความน่าจะเป็นในการมิวเตชันเท่ากับ 0.5 หากค่าที่สุ่มของแต่ละโครโมโซมมากกว่า 0.5 ให้ทำปรับปรุงโครโมโซม โดยสุ่มหมายเลขเครื่องจักร m_i เพื่อทำการสลับตำแหน่งงานบนเครื่องจักรนั้น (t_j และ t_{j+1}) โดยที่ j = 0, 1, ..., n หากการสลับทำให้ค่าความเหมาะสมดีกว่าเดิมให้ปรับค่าโครโมโซมนั้น แต่หากไม่ดีกว่าเดิมให้ใช้โครโมโซมเดิม และหยุดการทำงาน

7) คัดเลือกโครโมโซมที่ดีไปสู่รุ่นต่อไปและทำขั้นตอนที่ 3 จนครบจำนวนรอบที่กำหนด โครโมโซมที่มีค่าความเหมาะสมดีที่สุดของรอบสุดท้ายจะเป็นคำตอบของโมเดล LSGA

8) นำโครโมโซมที่ดีที่สุดจากโมเดล LSGA มาเพิ่มประสิทธิภาพด้วยอัลกอริทึมตาบูเป็นโมเดล Hybrid GA โดยกำหนดพารามิเตอร์ที่เกี่ยวข้อง คือ จำนวนรอบของการหาคำตอบเท่ากับ 750 รอบ และความยาวของตาบูลิส (TL) ตามสมการ (2.13)

$$TL = \frac{(numMachines \times numJobs)}{6} \tag{2.13}$$

โดยที่

numMachines คือ จำนวนเครื่องจักรทั้งหมด

numJobs คือ จำนวนงานทั้งหมด

ผลการทดลองเปรียบเทียบประสิทธิภาพของโมเดล LSGA กับอัลกอริทึม GA พบว่าโมเดล LSGA มีประสิทธิภาพดีกว่าอัลกอริทึม GA เนื่องจากสามารถพบจุดออพติมัมของข้อมูล หรือพบค่าที่ใกล้เคียงจุดออพติมัมมากกว่า และมีค่าความผิดพลาดน้อยกว่า

ผลการทดลองประสิทธิภาพของโมเดลที่นำเสนอระหว่างโมเดล LSGA กับโมเดล Hybrid GA พบว่าโมเดล Hybrid GA มีความสามารถในการพบจุดออพติมัมของข้อมูลได้มากกว่าหรือพบค่าที่ใกล้เคียงจุดออพติมัมมากกว่าโมเดล LSGA และสามารถใช้กับข้อมูลที่มีความซับซ้อนได้ดีกว่าโมเดล LSGA

2.3.2 A Tabu Search Algorithm for Job Shop Scheduling Problem with Industrial Scheduling Case Study เขียนโดย P. Senthil Velmurugan และ V. Selladurai [4]

งานวิจัยนี้นำเสนอโมเดลตาบูกับกลยุทธ์การหาชุดคำตอบข้างเคียงจากการใช้กราฟการเลือก (Disjunctive graph model) ซึ่งเป็นกราฟที่แสดงกิจกรรมของงานตามลำดับของงานและเครื่องจักร ซึ่งเส้นการเชื่อมระหว่างโอเปอร์เรชันที่เป็นงานเดียวกัน เรียกว่า เส้นเชื่อมโยง (Conjunctive arc) และเส้นเชื่อมระหว่างโอเปอร์เรชันที่ทำบนเครื่องจักรเดียวกัน เรียกว่า ส่วนโค้งการเลือก (Disjuctive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

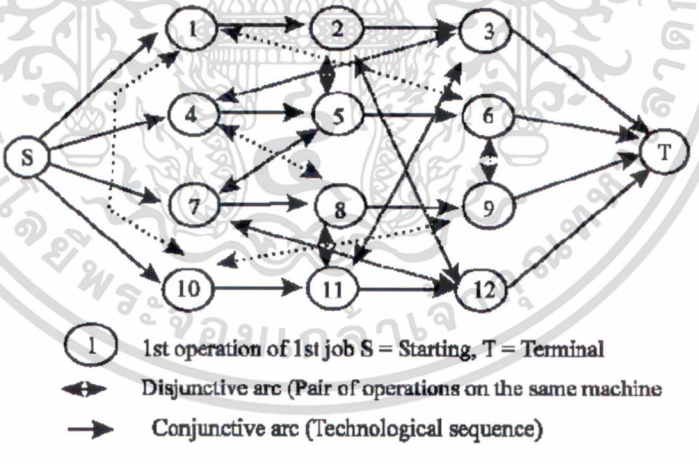
arc) โดยในงานวิจัยนี้มีวัตถุประสงค์ของการจัดตารางการผลิต คือ เวลาในการผลิตรวมน้อยที่สุด (makespan) ซึ่งมีขั้นตอนในการนำเสนอ ดังนี้

1) กำหนดชุดคำตอบเริ่มต้น (Initial solutions) โดยกำหนดรูปแบบของชุดคำตอบจากกราฟการเลือก ตัวอย่างเช่นข้อมูลจากตารางที่ 2.2 แสดงเวลาในการผลิตของปัญหาประเภท 4 งาน 3 เครื่องจักร ซึ่งมีจำนวนโอเปอร์เรชันของงานทั้งหมดเท่ากับ 12 โอเปอร์เรชัน โดยจัดลำดับตามงานและโอเปอร์เรชันของงาน สามารถเขียนความสัมพันธ์ด้วยกราฟการเลือก จำนวน 12 โหนด ตามรูปที่ 2.15

การเลือกชุดคำตอบเริ่มต้น สามารถเลือกจากโหนดที่ใช้เวลาทำงานของเครื่องจักรน้อยที่สุด (Shortest Processing Time : SPT) เช่น [1-7-8-9-10-11-12-2-3-4-5-6] เป็นต้น

ตารางที่ 2.2 ลำดับงาน และเวลาทำงานของปัญหาประเภท 4 งาน 3 เครื่องจักร

Job	Operation sequence (processing time)		
1	1 (2) [1]	2 (3) [2]	3 (4) [3]
2	3 (4) [4]	2 (4) [5]	1 (1) [6]
3	2 (2) [7]	3 (2) [8]	1 (3) [9]
4	1 (3) [10]	3 (3) [11]	2 (1) [12]



รูปที่ 2.15 กราฟการเลือก (Disjunctive Graph)

2) หาค่าความเหมาะสม (fitness) ของชุดคำตอบหากพบค่าออพติมัมของคำตอบให้หยุดการทำงาน แต่หากไม่พบให้เพิ่มรอบการทำงานและทำขั้นตอนที่ 3 ต่อไป

3) หาเส้นทางที่เป็นเส้นทางวิกฤต และสร้างชุดของคำตอบข้างเคียงจากการสร้างบล็อกบนเส้นวิกฤต (บล็อก หมายถึง ชุดของโอเปอร์เรชันหรือโหนดจำนวนอย่างน้อย 2 โอเปอร์เรชัน/โหนด ที่อยู่ติดกันและทำงานบนเครื่องจักรเดียวกัน โดยจะสร้างบนเส้นวิกฤต และอาจจะมีหลายบล็อกได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
115596

4) ใช้วิธีการของ first and last neighbours เพื่อหาตำแหน่งของโอเปอร์เรชันในบล็อกและทำการสลับตำแหน่งของโอเปอร์ชันคู่ที่อยู่ติดกันทุกคู่บนเส้นวิกฤตนั้น ดังอัลกอริทึมในรูปที่ 2.16 และเลือกสลับคู่โอเปอร์เรชันที่มีค่าความเหมาะสมต่ำสุด หรือเวลาในการผลิตรวมน้อยที่สุดตามรูปที่ 2.17

ในการเลือกคำตอบที่ดีสุดในแต่ละรอบจะต้องทำการตรวจสอบตำแหน่งคู่ที่สลับซึ่งเป็นสถานะต้องห้ามหรือไม่ (tabu list) ซึ่งหมายถึงว่าตำแหน่งคู่นั้นเคยถูกเลือกแล้ว และยังมีสถานะต้องห้าม จะไม่อนุญาตให้เลือกอีก ดังนั้น คำตอบที่ดีตัวถัดไปจะถูกนำมาพิจารณา แต่หากไม่เป็นสถานะต้องห้าม จะสามารถเลือกคำตอบดังกล่าวเป็นคำตอบในรอบนี้ และปรับค่าคำตอบรอบนี้แทนที่คำตอบเดิม ในโมเดลนี้ผู้วิจัยได้กำหนดให้ความยาวเริ่มต้นของตาบูลิสอยู่ในช่วง 8 - 14 ซึ่งความยาวของตาบูลิสในแต่ละรอบการทำงานไม่ได้เป็นค่าคงที่ มีการเปลี่ยนแปลงตามการสุ่ม ในบางช่วงจะมีค่ามาก และในบางช่วงจะมีค่าน้อย ในงานวิจัยนี้ได้กำหนดความยาวของตาบูลิสให้มีค่าตามรูปที่ 2.18 ซึ่งแสดงความยาวของตาบูลิสแบบพลวัตร (dynamic list) [5]

5) พิจารณาคำตอบที่ดีของแต่ละรอบเป็นคำตอบที่ดีที่สุด (best solution)

6) แสดงคำตอบที่ดีที่สุด (best solution)

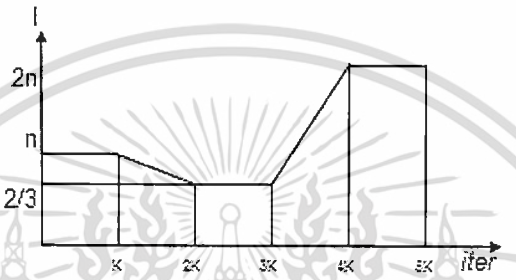
```
TS Algorithm  : Find neighbours
Parameters :   k                    = Points the last neighbour
               noofmach             = The number of available machines
               numblock[i]          = The number of blocks in ith machine
               block[i][j].node[k]  = Holds the posistion of kth  node in the
                                      jth block of ith  machine
               block[i][j].count    = Holds the total number of nodes in the
                                      jth  block of ith  machine
               neighbourlist[i].start = Holds the start node of ith  neighbour
               neighbourlist[i].end   = Holds the end node of ith neighbour
k=0;
for(m=0;m<noofmach;m++)
{
 for(l=0;l<=numblock[m];l++)
 {
      if(block[m][l].node[0]!=0)
      {
       assign block[m][l].node[0] to neighbourlist[k].start
       assign block[m][l].node[1] to neighbourlist[k].end
       k++;
       if(block[m][l].count>1)
       {
      assign block[m][l].nodes[block[m][l].count-1] to neighbourlist[k].start
      ssign block[m][l].nodes[block[m][l].count] to neighbourlist[k].end
      k++;
       }
      }
 }
}
```

รูปที่ 2.16 อัลกอริทึมของวิธีการ first and last neighbours

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Itcr. No.	Neighbours	C_{max}	Move	Tabu list
1	10-9, 3-11, 4-3	21, 22, 22	10-9	9-10
2	4-3, 3-11, 10-1	17, 21, 21	4-3	9-10, 3-4
3	10-1, 4-11	16,17	10-1	9-10, 3-4, 1-10
4	11-8, 4-11	17, 17	11-8	9-10, 3-4, 1-10, 8-11
5	8-11, 4-8	16, 16	4-8	9-10, 3-4, 1-10, 8-11, 8-4
6	4-11, 6-9, 3-8	14, 18, 20	4-11	9-10, 3-4, 1-10, 8-11, 8-4, 11-4
7	5-12, 1-10, 12-2	13, 14, 16	5-12	9-10, 3-4, 1-10, 8-11, 8-4, 11-4, 12-5

รูปที่ 2.17 บล็อกของโหนดที่เป็นชุดคำตอบข้างเคียง



รูปที่ 2.18 ตาบูลิสแบบพลวัตร

การทดลองได้นำไปทดสอบกับข้อมูลมาตรฐาน จาก OR-Library [6] จำนวน 29 ชุดพบว่าส่วนใหญ่พบค่าจุดออพติมัม (optimum) ของคำตอบ ส่วนข้อมูลที่ไม่สามารถหาจุดออพติมัมได้ มีความแตกต่างจากจุดออพติมัม (relative error : RE) โดยเฉลี่ยน้อยกว่า 1 เปอร์เซ็นต์ มีจำนวน 5 ชุด ประกอบด้วย LA16, ORB01, ORB04, ABZ05 และ FT10 ตามลำดับ ดังรูปที่ 2.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test instances	Size	Best solution known	Obtained solution	RE (%)
LA 01	10×5	666	666	0.00
LA 02	10×5	655	655	0.00
LA 03	10×5	597	597	0.00
LA 04	10×5	590	590	0.00
LA 05	10×5	593	593	0.00
LA 06	15×5	926	926	0.00
LA 07	15×5	890	890	0.00
LA 08	15×5	863	863	0.00
LA 09	15×5	951	951	0.00
LA 10	15×5	958	958	0.00
LA 11	20×5	1222	1222	0.00
LA 12	20×5	1039	1039	0.00
LA 13	20×5	1150	1150	0.00
LA 14	20×5	1292	1292	0.00
LA 15	20×5	1207	1207	0.00
LA16	10×10	945	946	0.10
LA17	10×10	784	784	0.00
LA18	10×10	848	848	0.00
LA19	10×10	842	842	0.00
LA20	10×10	902	902	0.00
ORB01	10×10	1059	1064	0.47
ORB02	10×10	888	888	0.00
ORB03	10×10	1005	1005	0.00
ORB04	10×10	1005	1011	0.59
ORB05	10×10	887	887	0.00
ABZ05	10×10	1234	1236	0.16
ABZ06	10×10	943	943	0.00
FT 06	06×06	55	55	0.00
FT10	10×10	930	935	0.53

รูปที่ 2.19 ผลการทดสอบโมเดลของ P. S. Velmurugan and V. Selladurai กับข้อมูลทดสอบ 29 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โมเดลที่นำเสนอ

งานวิจัยนี้นำเสนอวิธีการจัดตารางการผลิตแบบตามงานโดยใช้วิธีการเชิงพันธุกรรมร่วมกับอัลกอริทึมตาบู และฟัซซีลอจิก ซึ่งเนื้อหาในบทนี้จะกล่าวถึงขั้นตอนการดำเนินงานของโมเดลที่นำเสนอ ซึ่งประกอบด้วย 3 ส่วนสำคัญ คือ เตรียมข้อมูลอินพุท การคัดเลือกโครโมโซมด้วยฟังก์ชันการเป็นสมาชิก และกระบวนการเรียนของโมเดล HGA ที่นำเสนอในการจัดตารางการผลิตแบบตามงาน

3.1 การเตรียมข้อมูลอินพุท

ตัวแทนของชุดคำตอบที่มีความเป็นไปได้จะถูกจำลองเป็นโครโมโซม โดยภายในโครโมโซมจะประกอบด้วยองค์ประกอบที่เรียกว่า ยีน (genes) ยกตัวอย่างปัญหาประเภท 3 งาน 3 เครื่องจักร ตามตารางที่ 3.1 กำหนดให้หนึ่งโครโมโซมเป็นหนึ่งชุดคำตอบที่สามารถเป็นไปได้จากการสุ่มหมายเลขงานตามจำนวนเครื่องจักร เช่น [2 1 1 2 2 3 3 1 3] โดย 2 ตัวแรก คืองานที่ 2 โอเปอร์เรชันที่ 1 และ 1 ตัวที่สอง คืองานที่ 1 โอเปอร์เรชันที่ 1 เป็นต้น ทำการสร้างโครโมโซมเท่ากับจำนวนประชากรเริ่มต้น (N) ตามที่กำหนด ดังนั้นลักษณะของโครโมโซมแสดงได้ตามรูปที่ 3.1

ตารางที่ 3.1 ตัวอย่างปัญหาการผลิตแบบตามงานประเภท 3 งาน 3 เครื่องจักร

Job No.	Operation (machine / Process time)		
	Op1	Op2	Op3
1	1/3	2/3	3/2
2	1/1	3/5	2/3
3	2/3	1/2	3/3

2	1	3	3	2	1	3	2	1

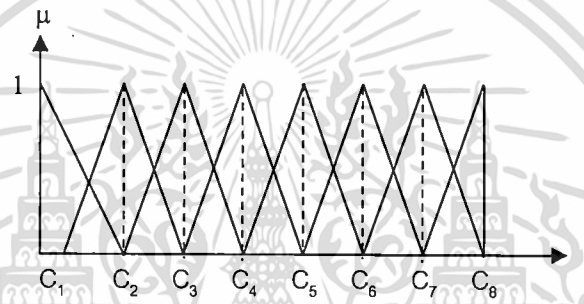
รูปที่ 3.1 ตัวอย่างโครโมโซมของปัญหา 3 งาน 3 เครื่องจักรของโมเดล HGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การใช้ฟังก์ชันการเป็นสมาชิก

การศึกษาในครั้งนี้เลือกใช้ฟังก์ชันการเป็นสมาชิกแบบสามเหลี่ยม (triangular membership function) จำนวน N ฟังก์ชัน (เท่ากับจำนวนโครโมโซมเริ่มต้น) เช่นถ้า N = 8 ฟังก์ชันการเป็นสมาชิกเริ่มต้นสามารถแสดงได้ดังรูปที่ 3.2

$$triangular(x:a,b,c)=\begin{cases}0 & x<a\\ (x-a)/(b-a) & a\le x\le b\\ (c-x)/(c-b) & b\le x\le c\\ 0 & x>c\end{cases} \tag{3.1}$$



รูปที่ 3.2 ฟังก์ชันการเป็นสมาชิกของโครโมโซมในโมเดล HGA

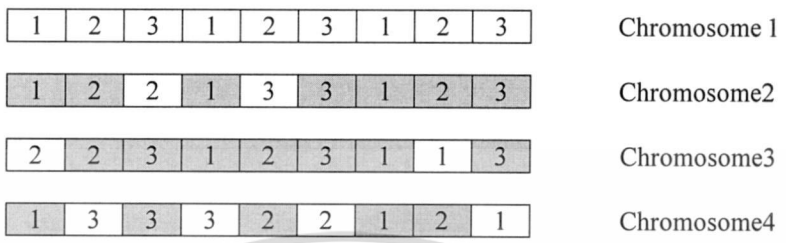
โดยแต่ละฟังก์ชันจะมีการกำหนดค่าพารามิเตอร์ 3 ค่าคือจุดเริ่มต้น (a) จุดศูนย์กลาง (c) และจุดสิ้นสุด (b) ดังสมการที่ 3.1 โดยมีการกำหนดจุดศูนย์กลาง C_1 คือโครโมโซมตัวแรกของรอบ, C_2 คือจุดศูนย์กลางของโครโมโซมที่มีความคล้ายกับโครโมโซมแรกมากที่สุด และ C_3 ถึง C_8 เป็นจุดศูนย์กลางของโครโมโซมที่ความคล้ายจากโครโมโซมก่อนหน้าตามลำดับจนครบ N โครโมโซม เมื่อ x คือค่าที่สุ่มเลือกโครโมโซมและใช้โอเปอร์เรชันของฟัซซีเซตเป็นยูเนียน (OR) ดังสมการที่ (3.2)

$$\mu_{A\cup B}(x)=\max[\mu_A(x),\mu_B(x)] \tag{3.2}$$

การแสดงความคล้ายของแต่ละโครโมโซมที่มีตำแหน่งงานและตำแหน่งยีนในโครโมโซมอยู่ตำแหน่งเดียวกัน ยกตัวอย่างรูปที่ 3.3 แสดงโครโมโซม 4 โครโมโซมจะเริ่มพิจารณาความคล้ายเริ่มจากโครโมโซมแรกและพิจารณาความคล้ายกับโครโมโซมที่เหลือ พบว่ามีความคล้ายกับโครโมโซมที่ 2 3 และ 4 คือ 7 7 และ 5 ตามลำดับ จึงเลือกโครโมโซมที่ 2 ดังนั้นการจัดวางฟังก์ชันความเป็นสมาชิกตัวแรกคือโครโมโซมที่ 1 ด้วยความกว้างของฟังก์ชันสามเหลี่ยมเท่ากับค่าความเหมาะสมของโครโมโซมที่ 1 และถัดไปคือโครโมโซมที่ 2 ด้วยความกว้างซ้อนทับฟังก์ชันความเป็นสมาชิกของโครโมโซมที่ 1 ด้วยความคล้าย 7 จากจำนวน 9 ยีนทั้งหมดโดยเทียบจากค่าความเหมาะสมของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครโมโซมก่อนหน้า และพิจารณาเลือกโครโมโซมลำดับถัดไปโดยพิจารณาโครโมโซมที่มีความคล้ายกับโครโมโซมที่ 2 มากที่สุด ทำลักษณะเดียวกันจนครบ N โครโมโซม



รูปที่ 3.3 ตัวอย่างความคล้ายของโครโมโซม

3.3 กระบวนการเรียนรู้

อัลกอริทึมสำหรับการเรียนรู้ตามรูปที่ 3.5 ซึ่งมีรายละเอียดดังนี้

ขั้นตอนที่ 1 จัดเตรียมข้อมูลอินพุทและฟังก์ชันวัตถุประสงค์

ขั้นตอนที่ 2 หาค่าความเหมาะสม (fitness) ของแต่ละโครโมโซม

ขั้นตอนที่ 3 สร้างฟังก์ชันความเป็นสมาชิกแบบสามเหลี่ยม

กำหนดให้เริ่มต้นจากโครโมโซมตัวแรกของรอบ และพิจารณาโครโมโซมที่มีความคล้ายกับโครโมโซมแรกมากที่สุดจะถูกเลือกจากสมการด้านล่าง

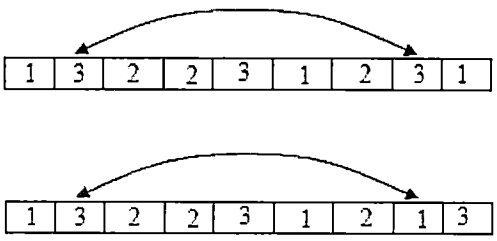
$$\mu_{A\cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

พิจารณาจนครบจำนวนโครโมโซมทั้งหมด จากนั้นทำการสุ่มค่าเพื่อเลือกฟังก์ชันที่มีความเป็นสมาชิกมากที่สุดเพื่อเป็นโครโมโซมต้นแบบ

ขั้นตอนที่ 4 สร้างทายาทด้วยการครอสโอเวอร์แบบ POX

ขั้นตอนที่ 5 สร้างความหลากหลายของโครโมโซมด้วยการมิเตชัน โดยใช้วิธีการตาบูในการจัดเก็บคุณสมบัติของตำแหน่งที่สุ่ม ในการมิวเตชันโดยการสุ่ม 2 ตำแหน่งในโครโมโซมที่เป็นไปได้ตามรูปที่ 3.4 (ซึ่งค่าที่สุ่มต้องไม่เป็นหมายเลขงานเดียวกัน) ให้ทำการสลับตำแหน่งและเก็บคุณสมบัติของการสลับในแต่ละโครโมโซมเป็นสถานะต้องห้าม โดยกำหนดให้ความยาวของตาบูลิสเป็นค่าคงที่ (จากการทดลองกำหนดให้ TL = 7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

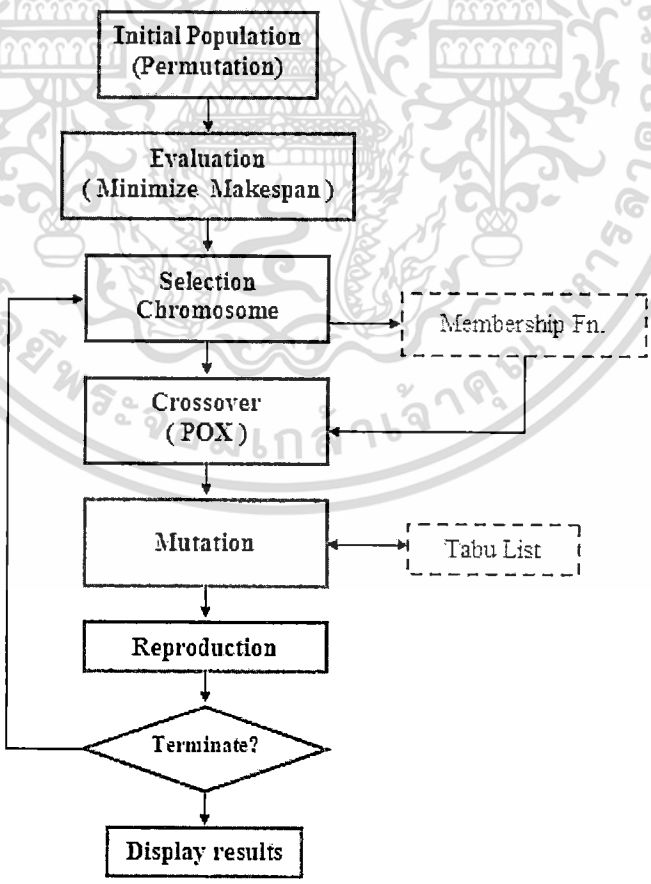


รูปที่ 3.4 ตัวอย่างการมิวเตชันโครโมโซม

ขั้นตอนที่ 6 การคัดเลือกโครโมโซม โดยโครโมโซมที่มีค่าความเหมาะสมสูงกว่าจะถูกเลือกให้เป็นโครโมโซมรุ่นต่อไป

ขั้นตอนที่ 7 หากพบคำตอบที่ได้ยังไม่พอใจให้ กลับไปทำขั้นตอนที่ 3 แต่ถ้าพบกับค่าที่เหมาะสมที่สุดให้หยุดการทำงานไปทำขั้นตอนต่อไป

ขั้นตอนที่ 8 แสดงคำตอบที่เหมาะสมที่สุด



รูปที่ 3.5 แผนผังแสดงขั้นตอนของโมเดลที่นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ข้อมูลของปัญหาการจัดตารางการผลิตแบบตามงานที่ใช้ทดสอบ

ข้อมูลที่ใช้ทดสอบครั้งนี้ใช้จากข้อมูลมาตรฐานจาก OR-Library [6] โดยแบ่งเป็น 3 กลุ่ม ดังนี้

1) ข้อมูลกลุ่มที่ 1 เป็นข้อมูลมาตรฐานของ S. Lawrence (1984) จำนวน 40 ชุด ได้แก่ la01 - la40
2) ข้อมูลกลุ่มที่ 2 เป็นข้อมูลมาตรฐานของ D. Applegate, W. Cook (1991) จำนวน 10 ชุดได้แก่ orb01 - orb10
3) ข้อมูลกลุ่มที่ 3 เป็นข้อมูลมาตรฐานของ H. Fisher, G.L. Thompson (1963) จำนวน 3 ชุด ได้แก่ ft06, ft10 และ ft20

4.2 กำหนดค่าพารามิเตอร์ของระบบ

ค่าพารามิเตอร์ที่กำหนดในการทดลองสำหรับโมเดล HGA มีดังต่อไปนี้

1) จำนวนโครโมโซม = 200
2) เงื่อนไขการหยุดเมื่อค่าความเหมาะสมไม่เปลี่ยนแปลง 80% ของโครโมโซม
3) อัตราการครอสโอเวอร์ = 0.95
4) อัตราการมิวเตชัน = 1
5) ความยาวของ TL = 7

การวัดประสิทธิภาพของโมเดล ได้จากการเปรียบเทียบค่าความผิดพลาด (relative error: RE) ระหว่างคำตอบที่ได้จากโมเดลที่นำเสนอกับผลเฉลยของชุดข้อมูล สามารถเขียนดังสมการ (4.1) ดังนี้

$$RE = \left(\frac{C_{best} - LB}{LB} \right) \times 100\% \qquad (4.1)$$

โดยที่ C_{best} หมายถึง คำตอบที่ได้จากโมเดล (makespan)

LB หมายถึง ค่าต่ำสุดของข้อมูล (lower bound)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดลอง

ข้อมูลที่ใช้ในการทดลองจะแบ่งเป็น 3 กลุ่มตามตารางที่ 4.1 – 4.4 โดยเทียบกับอัลกอริทึมอื่น เช่น Hybrid GA และ LSGA [3] และ TS [4]

ตารางที่ 4.1 ผลการทดลองของข้อมูล S. Lawrence (1984) กลุ่ม 1

ชุดข้อมูล	จำนวนงาน	จำนวนเครื่องจักร	LB	LSGA (2004)	TS (2007)	HGA	RE (%)
La01	10	5	666	666	666	666	0.0
La02	10	5	655	655	655	655	0.0
La03	10	5	597	597	597	597	0.0
La04	10	5	590	590	590	590	0.0
La05	10	5	593	593	593	593	0.0
La06	15	5	926	926	926	926	0.0
La07	15	5	890	890	890	890	0.0
La08	15	5	863	863	863	863	0.0
La09	15	5	951	951	951	951	0.0
La10	15	5	958	958	958	958	0.0
La11	20	5	1222	1222	1222	1222	0.0
La12	20	5	1039	1039	1039	1039	0.0
La13	20	5	1150	1150	1150	1150	0.0
La14	20	5	1292	1292	1292	1292	0.0
La15	20	5	1207	1207	1207	1207	0.0
La16	10	10	945	989	946	945	0.0
La17	10	10	784	793	784	784	0.0
La18	10	10	848	866	848	848	0.0
La19	10	10	842	867	842	842	0.0
La20	10	10	902	923	902	902	0.0
รวมจำนวนชุดที่พบจุดออพติมัมและความผิดพลาดเฉลี่ย				15	19	20	0.0

การทดสอบข้อมูลของ S. Lawrence กลุ่มที่ 1 จำนวน 20 ชุด (la01 – la20) พบว่าโมเดลที่นำเสนอมีความถูกต้องในการหาจุดออพติมัมได้อย่างถูกต้อง โดยค่าความผิดพลาดเท่ากับ 0.0% และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเทียบกับอัลกอริทึม LSGA และ TS พบว่าโมเดล HGA ที่นำเสนอสามารถพบจุดออพติมัมมากกว่า แสดงว่าโมเดลที่นำเสนอมีประสิทธิภาพดีกว่า

ตารางที่ 4.2 ผลการทดลองของข้อมูล S. Lawrence (1984) กลุ่ม 2

ชุดข้อมูล	จำนวนงาน	จำนวนเครื่องจักร	LB	LSGA (2004)	Hybrid GA (2004)	HGA	RE(%)
La16	10	10	945	959	959	945	0.0
La17	10	10	784	792	792	784	0.0
La18	10	10	848	875	875	848	0.0
La19	10	10	842	860	860	842	0.0
La20	10	10	902	907	907	902	0.0
La21	15	10	1053	1114	1097	1053	0.0
La22	15	10	927	989	980	930	0.32
La23	15	10	1032	1035	1032	1032	0.0
La24	15	10	935	1032	1001	941	0.64
La25	15	10	977	1047	1031	989	1.22
La26	20	10	1218	1307	1295	1218	0.0
La27	20	10	1235	1350	1306	1266	2.51
La28	20	10	1216	1312	1302	1234	1.48
La29	20	10	1152	1311	1280	1178	2.25
La30	20	10	1355	1451	1406	1355	0.0
La31	30	10	1784	1784	1784	1784	0.0
La32	30	10	1850	1850	1850	1850	0.0
La33	30	10	1719	1745	1719	1719	0.0
La34	30	10	1721	1784	1758	1721	0.0
La35	30	10	1888	1958	1888	1888	0.0
La36	15	15	1268	1358	1357	1301	2.60
La37	15	15	1397	1517	1494	1439	3.0
La38	15	15	1217	1362	1338	1227	0.82
La39	15	15	1233	1391	1343	1257	1.94
La40	15	15	1222	1323	1311	1242	1.63
รวมจำนวนชุดที่พบจุดออพติมัมและความผิดพลาดเฉลี่ย				2	5	14	0.74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบข้อมูลของ S. Lawrence กลุ่มที่ 2 จำนวน 25 ชุด (la16 - la40) พบว่าโมเดลที่นำเสนอมีความถูกต้องในการหาจุดออพติมัมได้อย่างถูกต้อง โดยมีค่าความผิดพลาด (RE) โดยเฉลี่ย 0.74 ซึ่งน้อยกว่า 1% และเมื่อเทียบกับอัลกอริทึม LSGA และ Hybrid GA พบว่าโมเดล HGA ที่นำเสนอสามารถพบจุดออพติมัมมากกว่าแสดงว่าโมเดลที่นำเสนอมีประสิทธิภาพดีกว่า

ตารางที่ 4.3 ผลการทดลองของข้อมูล D. Applegate, W. Cook (1991)

ชุดข้อมูล	จำนวนงาน	จำนวนเครื่องจักร	LB	LSGA (2004)	TS (2007)	HGA	RE(%)
Orb01	10	10	1059	1088	1064	1071	1.13
Orb02	10	10	888	921	888	888	0.0
Orb03	10	10	1005	1041	1005	1020	1.49
Orb04	10	10	1005	1052	1011	1005	0.0
Orb05	10	10	887	903	887	889	0.22
Orb06	10	10	1010	1062	-	1013	0.29
Orb07	10	10	397	408	-	397	0.0
Orb08	10	10	899	908	-	899	0.0
Orb09	10	10	934	980	-	934	0.0
Orb10	10	10	944	-	-	944	0.0
รวมจำนวนชุดที่พบจุดออพติมัมและความผิดพลาดเฉลี่ย				0	3	6	0.31

การทดสอบข้อมูลของ D. Applegate, W. Cook จำนวน 10 ชุด (Orb01 – Orb10) พบว่าโมเดลที่นำเสนอมีความถูกต้องในการหาจุดออพติมัมได้อย่างถูกต้อง โดยมีค่าความผิดพลาด (RE) โดยเฉลี่ย 0.31 ซึ่งน้อยกว่า 0.5% และเมื่อเทียบกับอัลกอริทึม LSGA และ TS พบว่าโมเดล HGA ที่นำเสนอสามารถพบจุดออพติมัมมากกว่า แสดงว่าโมเดลที่นำเสนอมีประสิทธิภาพดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ผลการทดลองของข้อมูล H. Fisher, G.L. Thompson (1963)

ชุดข้อมูล	จำนวนงาน	จำนวนเครื่องจักร	LB	Hybrid GA (2004)	TS (2007)	HGA	RE(%)
Ft06	6	6	55	55	55	55	0.0
Ft10	10	10	930	963	935	930	0.0
Ft20	20	5	1165	1222	-	1165	0.0
รวมจำนวนชุดที่พบจุดออพติมัมและความผิดพลาดเฉลี่ย				1	1	3	0.0

การทดสอบข้อมูลของ H. Fisher, G.L. Thompson จำนวน 3 ชุด (ft06, ft10 และ ft20) พบว่าโมเดลที่นำเสนอมีความถูกต้องในการหาจุดออพติมัมได้อย่างถูกต้อง โดยมีค่าความผิดพลาด (RE) เท่ากับ 0.0% และเมื่อเทียบกับอัลกอริทึม Hybrid GA และ TS พบว่าโมเดล HGA ที่นำเสนอ สามารถพบจุดออพติมัมมากกว่าแสดงว่าโมเดลที่นำเสนอมีประสิทธิภาพดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอโมเดล HGA ที่ประยุกต์ใช้ความรู้ทางปัญญาประดิษฐ์มาช่วยในการจัดตารางการผลิตแบบตามงาน โดยโมเดลที่นำเสนอได้จากการผสมผสานระหว่างจีเนติกอัลกอริทึม อัลกอริทึมตาบู และฟัซซีลอจิกเข้าด้วยกัน เพื่อหาจุดออพติมัมของข้อมูลโดยนำฟัซซีลอจิกในการหาค่าความเป็นสมาชิกของโครโมโซมเพื่อเลือกเป็นโครโมโซมต้นแบบในการสร้างทายาท และใช้เลือกอัลกอริทึมตาบูในการมิวเตชันเพื่อเพิ่มความหลากหลายของโครโมโซม

การทดสอบประสิทธิภาพของโมเดลที่นำเสนอ ได้ทำการทดลองกับข้อมูลมาตรฐานใน OR-Library [6] จำนวน 3 กลุ่ม คือ 1) ข้อมูลของ S. Lawrence (1984) จำนวน 40 ชุด ได้แก่ la01-la40 2) ข้อมูลของ D. Applegate, W. Cook (1991) จำนวน 10 ชุดได้แก่ orb01-orb10 3) ข้อมูลของ H. Fisher, G.L. Thompson (1963) จำนวน 3 ชุด ได้แก่ ft06, ft10 และ ft20 ตามลำดับ ซึ่งการทดลองของข้อมูลทั้ง 4 กลุ่มพบว่ามีความถูกต้องของการหาค่าออพติมัม โดยมีค่าความผิดพลาด (RE) โดยเฉลี่ย 0.0 0.74 0.31 และ 0.0 ตามลำดับ และเมื่อเทียบกับอัลกอริทึมอื่น [3] และ [4] พบว่าโมเดลที่นำเสนอมีความถูกต้องมากกว่า

จากการศึกษาในครั้งนี้แสดงให้เห็นว่าการนำความรู้ทางปัญญาประดิษฐ์มาประยุกต์ใช้ในการจัดตารางการผลิตแบบตามงานเป็นวิธีหนึ่งที่น่าสนใจ แต่ยังมีปัจจัยในการกำหนดความถูกต้องในการหาออพติมัมของข้อมูล คือ ค่าพารามิเตอร์ Pop size คือ จำนวนประชากร, TL ค่าหน่วยความจำในการเก็บคุณสมบัติของคำตอบ, Pc อัตราการครอสโอเวอร์ และ Pm ความน่าจะเป็นในการมิวเตชัน ซึ่งการกำหนดพารามิเตอร์ขึ้นอยู่กับข้อมูลเป็นสำคัญ ดังนั้นอุปสรรคข้อหนึ่งที่สำคัญคือการกำหนดค่าพารามิเตอร์ที่เหมาะสมเพื่อให้โมเดลสามารถพบจุดออพติมัมได้ถูกต้องมากยิ่งขึ้น

5.2 ข้อเสนอแนะ

การศึกษาในครั้งนี้ได้สนใจการพัฒนาการหาค่าเหมาะสมที่สุด (Optimization) ของปัญหาการจัดตารางการผลิตแบบตามงาน (JSSP) เท่านั้น ดังนั้นเพื่อให้สามารถนำไปประยุกต์ใช้งานได้จริง จึงควรพัฒนางานวิจัยนี้ต่อไป โดยให้สามารถหาค่าเหมาะสมที่สุดกับปัญหาการจัดตารางการผลิตที่มีเงื่อนไขที่มีความซับซ้อนตามลักษณะของอุตสาหกรรมการผลิตจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ปารเมศ ชุติมา, "เทคนิคการจัดตารางการดำเนินงาน", กรุงเทพฯ : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2546.

[2] M. P. Kleeman and G. B. Lamont, **"Scheduling of flow-shop, job-shop, and combined scheduling problems using moeas with fixed and variable length chromosomes"**, Studies in Computational Intelligence, 49:49–99, 2007.

[3] Ombuki BM, Ventresca M. **"Local search genetic algorithms for the job shop scheduling problem"**, Appl Intell 21:99–109, 2004.

[4] Senthil Velmurugan, P. and V. Selladurai, **"A Tabu search algorithm for job shop scheduling problem with industrial scheduling case study"**, Int. J. Soft Comput., 2(4): 531-537, 2007.

[5] Pezzella, F. and E. Merelli, **"A tabu search method guided by shifting bottleneck for the job-shop scheduling problem"**, European Journal of Operational Research, 120, 297–310 (2000).

[6] OR Library, URL http://mscmga.ms.ic.ac.uk.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้