บทที่ 3
วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงวิธีการดำเนินงานตั้งแต่หลักการทำงานของ CMS ขั้นตอนการทำงานของ ROC1 และ ROC2 การออกแบบโปรแกรม และขั้นตอนการเขียนโปรแกรมซึ่งเป็น แสดงเป็น Flow Chart เพื่อใช้ในการแก้ปัญหาหน่วยการผลิต

3.1 ศึกษาหลักการและทฤษฎี เรื่อง CMS

CMS เป็นกลยุทธ์ที่แบ่งแยกระบบการผลิต ออกเป็นกลุ่มระบบเล็กๆ หรือระบบหน่วยการผลิต โดยแต่ละหน่วยการผลิตจะสามารถผลิตชิ้นส่วน / ส่วนประกอบให้เสร็จสมบูรณ์ ซึ่งเป็นการประยุกต์ใช้ปรัชญาการผลิตที่รู้จักกันดีคือ GT ได้ถูกอธิบายไว้ว่า สามารถที่จะแบ่งปัญหาใหญ่ ๆ นั้นลงเป็นปัญหาที่สามารถจัดการได้ให้เล็กลงเป็นกลุ่ม และแก้ไขปัญหาได้อย่างมีประสิทธิภาพ

3.2 ขั้นตอนการทำงานของ ROC1 และ ROC2

ขอบเขตของ ROC1 และ ROC2

ในการทำ ROC1 และ ROC2 จะคำนึงถึงชนิดของเครื่องจักรเท่านั้นไม่คำนึงถึงจำนวนเครื่องจักร ไม่ว่าเครื่องจักรจะมีจำนวนเท่าไหร่ก็ให้คิดเป็น 1 เครื่อง

3.2.1 Rank Order Clustering 1 (ROC1)

วิธีนี้ถูกพัฒนาขึ้นโดย King (1980 a, b) สำหรับการจัดกลุ่มชิ้นส่วน – เครื่องจักร การจัดง่าย ๆ มีประสิทธิภาพและเทคนิคการวิเคราะห์มีประสิทธิภาพ ที่สามารถคำนวณได้ง่าย นอกจากนี้ยังหาจุดบรรจบได้เร็วและใช้เวลาคำนวณน้อย แต่ละแถว (คอลัมน์) ในเมตริก ชิ้นส่วน – เครื่องจักร ที่ถูกอ่านจาก Binary Word กระบวนการในการเปลี่ยน Binary Word สำหรับแต่ละแถว (คอลัมน์) ให้ได้ผลลัพธ์ (Decimal Equivalent)

วิธีในการจัดแถว (คอลัมน์) ใหม่อีกครั้งไปเรื่อย ๆ ในการลดค่าจนกว่าไม่มีการเปลี่ยน ดังข้างล่าง

ROC 1 Algorithm

Step 1 แถว m = 1, 2..., M, คำนวณคำตอบ Cm โดยการอ่านค่า Binary Word

$$c_m = \sum_{p=1}^{P} 2^{P-p} \cdot a_{pm} \quad (a_{pm} = 0 \text{ or } 1) \quad \text{... สมการที่ 3.1}$$

ทำซ้ำในแถวที่ c_m ลดลง ในกรณีที่คงที่ให้คงลำดับแบบเดิมไว้

Step 2 คอลัมน์ p = 1, 2..., P, คำนวณคำตอบ r_p, โดยอ่านค่า Binary Word

$$r_p = \sum_{m=1}^{M} 2^{M-m} \cdot a_{pm} \quad (a_{pm} = 0 \text{ or } 1) \quad \text{... สมการที่ 3.2}$$

ทำซ้ำในคอลัมน์ที่ r_p, ลดลง ในกรณีที่คงที่ให้คงลำดับแบบเดิมไว้

Step 3 ถ้าเมตริกซ์ชิ้นส่วน – เครื่องจักรใหม่ไม่เปลี่ยน ให้หยุดแล้วไปเริ่มที่ step1 ใหม่

ตัวอย่างที่ 3.1

ประยุกต์ ROC 1 ในเมตริกซ์ ชิ้นส่วน – เครื่องจักร ดังตางรางที่ 3.1

Step 1 ผลลัพธ์ของ Binary Number สำหรับแถวอยู่ทางขวามือในเมตริกซ์ ดังตารางที่ 3.2

การจัดลำดับ (Rank Order) ของแถวถูกแสดงในวงเล็บ

บนแถวที่มีการจัดเรียงใหม่ ใน Order ของลำดับที่ลดลง

แถวที่ถูกสลับในเมตริกซ์ ดังตารางที่ 3.3

Step 2 การจัดลำดับของคอลัมน์ แสดงในตารางที่ 3.3 ด้วย โดยการจัดอันดับคอลัมน์ใหม่ใน Order ของ Rank ที่ลดลง คอลัมน์ที่ถูกสลับในเมตริกซ์ดังตารางที่ 3.4

Step 1 จัดเรียงแถวใหม่โดยใช้พื้นฐานจาก Rank Order 1 ที่แสดงในตารางที่ 3.4

เมตริกซ์ที่หามาได้จะแสดงในตารางที่ 3.5

Step 2 นอกจากนี้ การจัดลำดับคอลัมน์ใหม่ จะไม่เกิดขึ้นบนพื้นฐานของการจัดเรียงที่แสดงในตารางที่ 3.5

Step 3 จากการแสดง Step 1 และ Step 2 เมตริกซ์ยังคงไม่ถูกแปลง ดังนั้นให้หยุด

จาก Block Diagonal Matrix ที่แสดงในตารางที่ 3.5

ที่จะมีสักอันที่สามารถระบุกลุ่มของชิ้นส่วนและกลุ่มของเครื่องจักรซึ่งมีความเป็นไปได้ 2 แบบ ดังตารางที่ 3.6 และ 3.7

ทุกการจัดเรียง 2 Cell (ตารางที่ 3.6) นำจำนวนที่น้อยที่สุดของส่วนประกอบที่มีน้อยและ Voids และ จากส่วนนี้ไปที่ถูกเลือก)

ตารางที่ 3.1 เมตริกซ์ชิ้นส่วน – เครื่องจักรแบบดั้งเดิม สำหรับตัวอย่างที่ 3.1

		Parts							
		1	2	3	4	5	6	7	8
Machines	1	1	1			1			1
	2			1	1		1	1	
	3	1	1	1		1	1		1
	4				1			1	1
	5			1	1			1	
	6	1	1				1		

		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
		Parts								
		1	2	3	4	5	6	7	8	
Machines	1	1	1			1			1	2^5
	2			1	1		1	1		2^4
	3	1	1	1		1	1		1	2^3
	4				1			1	1	2^2
	5			1	1			1		2^1
	6	1	1				1			2^0

ตารางที่ 3.2 ขั้นตอนที่ 1, การคำนวณแบบ Decimal ที่มีค่าเท่ากัน

		Binary weight									
		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
		Parts								Decimal equivalent	Rank
		1	2	3	4	5	6	7	8		
Machines	1	1	1			1			1	201	(2)
	2			1	1		1	1		54	(4)
	3	1	1	1		1	1		1	237	(1)
	4				1			1		19	(6)
	5			1	1			1		50	(5)
	6	1	1				1			196	(3)

ตารางที่ 3.3 ขั้นตอนที่ 1, การสลับลำดับเมตริกซ์ในแนวตั้ง

		Parts								Binary weight
		1	2	3	4	5	6	7	8	
	3	1	1	1		1	1		1	2^5
	1	1	1			1			1	2^4
Machines	6	1	1				1			2^3
	2			1	1		1	1		2^2
	5			1	1			1		2^1
	4				1			1	1	2^0
Decimal equivalent		56	56	38	7	48	44	7	49	
Rank		(1)	(2)	(6)	(7)	(4)	(5)	(8)	(3)	

ตารางที่ 3.4 ขั้นตอนที่ 2, การสลับลำดับเมตริกซ์ในแนวนอน

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Decimal equivalent	Rank
Parts	1	2	8	5	6	3	4	7		
3	1	1	1	1	1	1			252	(1)
1	1	1	1	1	1				240	(2)
Machines 6	1	1			1				200	(3)
2					1	1	1	1	15	(5)
5						1	1	1	7	(6)
4			1				1	1	37	(4)

ตารางที่ 3.5 ขั้นตอนที่ 1 ทำซ้ำขั้นตอนที่ 2

		Parts								Binary weight
		1	2	8	5	6	3	4	7	
	3	1	1	1	1	1	1			2^5
	1	1	1	1	1					2^4
Machines	6	1	1			1				2^3
	4			1				1	1	2^2
	2					1	1	1	1	2^1
	5						1	1	1	2^0
Decimal equivalent		56	56	52	48	42	35	7	7	
Rank		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	

ตารางที่ 3.6 กลุ่มของชิ้นส่วนและเครื่องจักรที่มีความคลายคลึงกัน (จัดไว้เป็นลำดับที่ 1)

Parts

Machines	1	2	8	5	6	3	4	7
3	1	1	1	1	1	1		
1	1	1	1	1	0			
6	1	1	0	1	1			
4			1			0	1	1
2					1	1	1	1
5						1	1	1

Number of exceptional elements = 3
Number of voids = 3
Total = 6

ตารางที่ 3.7 กลุ่มของชิ้นส่วนและเครื่องจักรที่มีความคลึงกัน (จัดไว้เป็นลำดับที่ 2)

Parts

Machines	1	2	8	5	6	3	4	7
3	1	1	1	1	1	1		
1	1	1	1	1				
6	1	1		1	1			
4			1			0	1	1
2					1	1	1	1
5						1	1	1

Number of exceptional elements = 7
Number of voids = 1
Total = 8

ข้อจำกัดของ ROC1

1. การอ่านค่าของการเข้าในฐานะที่เป็น Binary words เป็นการเสนอการคำนวณที่ยาก ตั้งแต่จำนวนเต็มที่ใหญ่ที่สุด ที่เป็นตัวอย่างในคอมพิวเตอร์ คือ $2^{48} - 1$ หรือน้อยกว่า จำนวนที่มากที่สุดของแถวและคอลัมน์จำกัดที่ 47
2. ผลลัพธ์ที่ขึ้นอยู่กับเมตริกซ์เริ่มแรก ดังนั้นการแก้ไขสุดท้ายไม่จำเป็นต้องเป็นการแก้ไขที่ดีที่สุด เป็นการรักษาส่วนประกอบที่น้อยมากอย่างไม่เจาะจง
3. มีแนวโน้มในการจัดกลุ่ม 1 ในมุมบนซ้ายมือในขณะที่ส่วนที่เหลือของเมตริกซ์อาจจะไม่มีระเบียบ
4. แม้ในโครงสร้างเมตริกซ์ที่ดี มันก็ไม่ใช่ ROC1 ที่แน่นอนที่ระบุโครงสร้าง Block Diagonal
5. การระบุเครื่องจักรที่ทำให้การผลิตช้าและชิ้นส่วนน้อยมาก เป็นการไม่เจาะจงและเป็นสิ่งสำคัญที่จะระบุการจัดกลุ่มในภายหลัง

3.2.2 Rank Order Clustering (ROC2)

ROC 2 ถูกพัฒนาโดย King และ Nakornchai (1982) เป็นการเอาชนะข้อจำกัดในการคำนวณ ROC 1 วิธีนี้เริ่มโดยการระบุในคอลัมน์ขวาสุดทุกแถวที่มีการเข้าเลข 1

แถวเหล่านี้จะถูกย้ายเป็นคอลัมน์สูงสุด การรักษาความสัมพันธ์ของ Order ระหว่างแถว วิธีนี้ถูกประยุกต์ที่แถวโดยเริ่มที่แถวสุดท้าย การใช้ Binary เป็นการขจัด ในวิธีการนี้ แม้ความคิดของการจัด Ordering ยังคงมีอยู่กับข้อจำกัดอื่น ๆ วิธีการนี้ถูกดำเนินการใน Program ที่มีระบบการโต้ตอบกับเครื่องมือที่อำนวยความสะดวกอย่างหลากหลาย ในการจัดข้อมูลในการกระทำที่ต้องการ ดังนั้น แม้สำหรับเมตริกซ์ที่ซับซ้อน งานที่พิสูจน์หลากหลายของส่วนประกอบที่น้อยมาก และการเปลี่ยนของชิ้นส่วนของชนิดเดียวกัน สามารถทำและผลที่ได้รับการตัดสินใจที่รวดเร็ว ถ้าผลไม่เป็นอย่างที่คาดหวังในตอนต้น สามารถทำให้สำเร็จได้รวดเร็วและการดำเนินการพิสูจน์อื่น ๆ วิธีมีดังนี้

ROC 2 Algorithm

1. จัดเรียงแถว จาก p = P (คอลัมน์สุดท้าย) ถึง 1 กำหนดแถวกับการ Entry 1 ย้ายแถวกับการเข้าส่วนหัวของแถว คง order ก่อนหน้านี้ของ entries
2. จัดเรียงคอลัมน์จาก m = M (แถวสุดท้าย) ถึง 1 กำหนดคอลัมน์กับการ Entry 1 ย้ายคอลัมน์กับการเข้าส่วนหัวของคอลัมน์ คง Order ก่อนหน้านี้ของ Entries
3. ทำ ข้อ 1 และ 2 ซ้ำ จนกว่าจะไม่มีการเปลี่ยนแปลง หรือได้รับการตรวจสอบ

ตัวอย่างที่ 3.2

พิจารณาเมตริกซ์ 8 ชิ้นส่วน และ 6 เครื่องจักร ดังตารางที่ 3.8

ข้อ 1 จัดแถวเลือกคอลัมน์สุดท้ายคือ p = 8 Order แรกของแถวคือ 1, 2, 3, 4, 5, 6 ภายใต้คะแนนของแถวที่ประกอบด้วย 1 ในคอลัมน์ได้ 1, 2, 3, 4, 5, 6 ย้าย 1, 3 และ 4 เป็นหัวของรายการ ตามด้วย 2, 5 และ 6 ใน Order เดียวกัน ที่อ่านจากซ้ายไปขวา ทำอย่างนี้ทุกคอลัมน์ (ตาราง 3.1) สลับแถวเมตริกซ์ที่แสดงในตารางที่ 3.9 สำหรับความสะดวกในการ Renumber แถวเหล่านี้ จาก 1 ถึง 6 เริ่มจากบนสุด ดังนั้นแถวใหม่ 1 ที่สอดคล้องอย่างที่เป็นอยู่กับแถวเก่า 3 และอื่น ๆ

ข้อ 2 จัดคอลัมน์ กระบวนการข้างบนให้ทำซ้ำโดยเลือกแถวสุดท้าย m = 6 (แถวเก่า4) จากการเปลี่ยนตำแหน่ง แสดงใน ตาราง 3.10 (เลขแถวเก่าแสดงในวงเล็บ) การสลับคอลัมน์ใน matrix แสดงในตารางที่ 3.11

ข้อ 1 จัดแถว จัดแถวใหม่เพื่อสังเกตถ้ามีการเปลี่ยนแปลงใน Order (ตาราง 3.12) ปรับปรุงแถวที่สลับในเมตริกซ์

ข้อ 2 จัดคอลัมน์ จัดคอลัมน์จนไม่เปลี่ยนแปลง

ข้อ 3 การพัฒนาทำต่อเป็นไปไม่ได้ ให้หยุด
เมตริกซ์สุดท้ายที่หามาได้เป็นอันเดียวกับที่ใช้ ROC1

ตารางที่ 3.8 ขั้นตอนที่ 1, การจัดการแนวตั้ง

Columns	Rows					
8	1	2	3	4	5	6
7	1	3	4	2	5	6
6	4	2	5	1	3	6
5	2	3	6	4	5	1
4	3	1	2	6	4	5
3	2	4	5	3	1	6
2	2	5	3	4	1	6
1	3	1	6	2	5	4
p	3	1	6	2	5	4

ตารางที่ 3.9 การจัดลำดับเมตริกซ์แนวตั้ง

Parts

Machines	1	2	3	4	5	6	7	8	New row numbers
3	1	1	1		1	1		1	1
1	1	1			1			1	2
6	1	1				1			3
2			1	1		1	1		4
5			1	1			1		5
4				1			1	1	6

ตารางที่ 3.10 ขั้นตอนที่ 2, การจัดการแนวนอน

Rows	Columns							
6(4)	1	2	3	4	5	6	7	8
5(5)	4	7	8	1	2	3	5	6
4(2)	4	7	3	8	1	2	5	6
3(6)	4	7	3	6	8	1	2	5
2(1)	6	1	2	4	7	3	8	5
1(3)	1	2	8	5	6	4	7	3
Column arrangement	1	2	8	5	6	3	4	7

ตารางที่ 3.11 การจัดลำดับเมตริกซ์ทั้งแนวนอนและแนวตั้ง

Parts

Machines	1	2	8	5	6	3	4	7	New row numbers
3	1	1	1	1	1	1			1
1	1	1	1	1					2
6	1	1			1				3
2					1	1	1	1	4
5						1	1	1	5
4			1				1	1	6
	1	2	3	4	5	6	7	8	

New column numbers

ตารางที่ 3.12 ขั้นตอนที่ 1 (ทบทวน), การจัดการแนวตั้ง

Columns	Rows					
8(7)	1(3)	2(1)	3(6)	4(2)	5(5)	6(4)
7(4)	4	5	6	1	2	3
6(3)	4	5	6	1	2	3
5(6)	4	5	1	6	2	3
4(5)	4	1	3	5	6	2
3(8)	1	2	4	3	5	6
2(2)	1	2	6	4	3	5
1(1)	1	2	3	6	4	5
Row Order	1(3)	2(1)	3(6)	6(4)	4(2)	5(5)

ตารางที่ 3.13 การจัดลำดับเมตริกซ์แนวตั้ง (ทบทวน)

Part

Machine	1	2	8	5	6	3	4	7	New row numbers
3	1	1	1	1	1	1			1
1	1	1	1	1					2
6	1	1			1				3
4			1				1	1	4
2					1	1	1	1	5
5						1	1	1	6
	1	2	3	4	5	6	7	8	

New column numbers

ตัวอย่างเพิ่มเติมของ ROC1 และ ROC2

สมมุติให้กระบวนการผลิตเป็นดังนี้

ตารางที่ 3.14 ตารางเครื่องจักรและชิ้นส่วนสำหรับตัวอย่างเพิ่มเติม

Machines \ Parts	1	2	3	4	5	6	7
1			1		1		1
2	1	1		1			1
3	1	1		1			
4			1		1	1	
5			1		1	1	1
6		1		1			1
7	1			1			1

ROC 1

ขั้นตอนที่ 1 ให้ค่า Binary Number เพื่อคำนวณค่า Decimal Equivalent และจัดลำดับคอลัมน์ใหม่

ตารางที่ 3.15 ขั้นตอนที่ 1 การให้ค่า Binary Number สำหรับจัดคอลัมน์

Machines \ Parts	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Decimal equivalent	Rank
	1	2	3	4	5	6	7		
1			1		1		1	21	(7)
2	1	1		1			1	105	(1)
3	1	1		1				104	(2)
4			1		1	1		22	(6)
5			1		1	1	1	23	(5)
6		1		1			1	41	(4)
7	1			1			1	73	(3)

ขั้นตอนที่ 2 นำเมตริกซ์ที่ได้จากขั้นตอนที่ 1 มาให้ค่า Binary Number เพื่อคำนวณค่า Decimal Equivalent และจัดลำดับแถวใหม่

ตารางที่ 3.16 ขั้นตอนที่ 2 การให้ค่า Binary Number สำหรับจัดแถว

		Parts						Binary weight
		1	2	3	4	5	6	7	
Machines	2	1	1		1			1	2^6
	3	1	1		1				2^5
	7	1			1			1	2^4
	6		1		1			1	2^3
	5			1		1	1	1	2^2
	4			1		1	1		2^1
	1			1		1		1	2^0
Decimal equivalent		112	107	7	120	7	6	93	
Rank		(2)	(3)	(5)	(1)	(6)	(7)	(4)	

ขั้นตอนที่ 3 นำเมตริกซ์ที่ได้จากขั้นตอนที่ 2 มาทำซ้ำตาม ขั้นตอนที่ 1 และ 2 จนเมตริกซ์ไม่มีการเปลี่ยนแปลงให้หยุด

ตารางที่ 3.17 การทำซ้ำจากขั้นตอนที่ 1 และ 2 สำหรับจัดคอลัมน์

		2^6	2^5	2^4	2^3	2^2	2^1	2^0	Decimal equivalent	Rank
		Parts								
		4	1	2	7	3	5	6		
Machines	2	1	1	1	1				120	(1)
	3	1	1	1					112	(2)
	7	1	1		1				104	(3)
	6	1		1	1				88	(4)
	5				1	1	1	1	15	(5)
	4					1	1	1	7	(7)
	1				1	1	1		14	(6)

ตารางที่ 3.18 การทำซ้ำจากขั้นตอนที่ 1 และ 2 สำหรับจัดแถว

		Parts							Binary weight
		4	1	2	7	3	5	6	
Machines	2	1	1	1	1				2^6
	3	1	1	1					2^5
	7	1	1		1				2^4
	6	1		1	1				2^3
	5				1	1	1	1	2^2
	1				1	1	1		2^1
	4					1	1	1	2^0
Decimal equivalent		120	112	104	94	7	7	5	
Rank		(1)	(2)	(3)	(4)	(5)	(6)	(7)	

ตารางที่ 3.19 ผลลัพธ์

		Parts						
		4	1	2	7	3	5	6
Machines	2	1	1	1	1			
	3	1	1	1				
	7	1	1		1			
	6	1		1	1			
	5				1	1	1	1
	1				1	1	1	
	4					1	1	1

ROC2

สมมติให้ใช้กระบวนการผลิตเดิม

ขั้นตอนที่ 1 จัดเรียงแถวจากคอลัมน์สุดท้าย ถึงคอลัมน์ที่ 1 และถ้าพบ 1 ในแถวนั้นให้นำไปไว้ด้านหน้าตามลำดับ ส่วนแถวที่ไม่มี 1 ให้เรียงลำดับต่อท้ายจากแถวที่มี และให้คงลำดับดังเดิม

ตารางที่ 3.20 ขั้นตอนที่ 1, จากจัดการแนวตั้ง

Columns	Rows						
7	1	2	3	4	5	6	7
6	1	2	5	6	7	3	4
5	5	4	1	2	6	7	3
4	5	4	1	2	6	7	3
3	2	6	7	3	5	4	1
2	5	4	1	2	6	7	3
1	2	6	3	5	4	1	7
p	2	3	7	6	5	4	1

ตารางที่ 3.21 การจัดลำดับเมตริกซ์แนวตั้ง

		Parts							
		1	2	3	4	5	6	7	
	2	1	1		1			1	
	3	1	1		1				
	7	1			1			1	
Machine	6		1		1			1	New row numbers
	5			1		1	1	1	
	4			1		1	1		
	1			1		1		1	

ขั้นตอนที่ 2 จัดเรียงคอลัมน์จากแถวสุดท้าย ถึงแถวที่ 1 และถ้าพบ 1 ในคอลัมน์นั้นให้นำไปไว้ด้านหน้าตามลำดับ ส่วนแถวที่ไม่มี 1 ให้เรียงลำดับต่อท้ายจากแถวที่มี และให้คงลำดับดังเดิม

ตารางที่ 3.22 ขั้นตอนที่ 2, การจัดการแนวนอน

Rows	Columns						
1	1	2	3	4	5	6	7
4	3	5	7	1	2	4	6
5	3	5	6	7	1	2	4
6	3	5	6	7	1	2	4
7	7	2	4	3	5	6	1
3	7	4	1	2	3	5	6
2	4	1	2	7	3	5	6
Columns arrangement	4	1	2	7	3	5	6

ตารางที่ 3.23 การจัดลำดับเมตริกซ์แนวนอน

		Parts						
		4	1	2	7	3	5	6
Machines	2	1	1	1	1			
	3	1	1	1	0			
	7	1	1	0	1			
	6	1	0	1	1			
	5				1	1	1	1
	4					1	1	1
	1				1	1	1	0

New column numbers

ขั้นตอนที่ 3 นำเมตริกซ์ที่ได้จากขั้นตอนที่ 2 มาทำซ้ำตาม ขั้นตอนที่ 1 และ 2 จนเมตริกซ์ไม่มีการเปลี่ยนแปลงให้หยุด

ตารางที่ 3.24 การทำซ้ำจนไม่มีการเปลี่ยนแปลง

Columns	Rows						
7(6)	1(2)	2(3)	3(7)	4(6)	5(5)	6(4)	7(1)
6(5)	5	6	1	2	3	4	7
5(3)	5	6	7	1	2	3	4
4(7)	5	6	7	1	2	3	4
3(2)	5	7	1	3	4	6	2
2(1)	1	4	2	5	7	3	6
1(4)	1	2	3	4	5	7	6
Row Order	1(2)	2(3)	3(7)	4(6)	5(5)	7(1)	6(4)

ตารางที่ 3.25 ผลลัพธ์

		Parts						
		4	1	2	7	3	5	6
Machines	2	1	1	1	1			
	3	1	1	1	0			
	7	1	1	0	1			
	6	1	0	1	1			
	5				1	1	1	1
	1				1	1	1	0
	4				0	1	1	1

3.2.3 สาเหตุที่ผลลัพธ์ของ ROC1 และ ROC2 จึงเหมือนกัน

ผู้จัดทำจะทำการเปรียบเทียบวิธีการทำงานของ ROC 1 และ ROC 2 ในแต่ละขั้นตอนไปพร้อม ๆ กันเพื่อเปรียบเทียบ ดังนี้

สมมุติว่ากระบวนการผลิตเป็นดังนี้

ตารางที่ 3.26 ตารางเครื่องจักรและชิ้นส่วน

		Parts			
		1	2	3	4
Machines	1	1		1	1
	2		1	1	
	3	1		1	
	4		1		1

ตารางที่ 3.27 ตารางเปรียบเทียบ ROC1 กับ ROC2 ขั้นตอนที่ 1

ROC1

		2^3	2^2	2^1	2^0	Decimal equivalent	Rank
		Parts					
		1	2	3	4		
Machines	1	1		1	1	11	(1)
	2		1	1		6	(3)
	3	1		1		10	(2)
	4		1		1	5	(4)

ROC2

Columns	Rows			
4	1	2	3	4
3	1	4	2	3
2	1	2	3	4
1	2	4	1	3
p	1	3	2	4

จะเห็นได้ว่าใน ROC1 จะให้น้ำหนักของแถวทางซ้ายมือมากที่สุด คือ 2^3 จึงทำให้ในคอลัมน์ที่มีเลข 1 อยู่ในนั้นจะมีลำดับอยู่ลำดับต้น ๆ ส่วนใน ROC2 นั้นจะเน้นไปที่คอลัมน์สุดท้าย ถ้าพบว่ามีเลข 1 อยู่ในแถวแรกก็จะทำให้มีลำดับอยู่ลำดับต้น ๆ

ตารางที่ 3.28 ตารางเปรียบเทียบ ROC1 กับ ROC2 ขั้นตอนที่ 2

		Parts				Binary weight
		1	2	3	4	
	1	1		1	1	2^3
Machines	3	1		1		2^2
	2		1	1		2^1
	4		1		1	2^0
Decimal equivalent		12	3	14	9	
Rank		(2)	(4)	(1)	(3)	

Rows	Columns			
4	1	2	3	4
2	2	4	1	3
3	2	3	4	1
1	3	1	2	4
Columns	3	1	4	2

จะเห็นได้ว่าใน ROC1 จะให้น้ำหนักของคอลัมน์ด้านบนมากที่สุด คือ 2^3 จึงทำให้ในแถวที่มีเลข 1 อยู่นั้นจะมีลำดับอยู่ในลำดับต้น ๆ ส่วนใน ROC2 นั้นจะเน้นไปที่ แถวสุดท้ายถ้าพบเลข 1 อยู่ในคอลัมน์นั้นก็จะทำให้มีลำดับอยู่ลำดับต้น ๆ

ผลลัพธ์ที่ได้

ตารางที่ 3.29 ผลลัพธ์

		Parts			
		3	1	4	2
Machines	1	1	1	1	
	3	1	1		
	2	1		0	1
	4			1	1

3.3 **ศึกษาขั้นตอนและวิธีการเขียนโปรแกรมโดยใช้** Microsoft Visual Basic.Net

สำหรับ Microsoft Visual Basic.Net นั้นมีความสามารถในการใช้งานมากมายหลากหลาย ทั้งความสามารถในการเขียนโปรแกรมติดต่อกับเครือข่ายอินเตอร์เน็ต การเชื่อมต่อกับระบบฐานข้อมูล รวมทั้งปรับปรุงเครื่องมือและการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ให้สมบูรณ์ยิ่งขึ้น พร้อมทั้งเพิ่มเครื่องมือต่าง ๆ อีกมากมายที่ทำให้ใช้งานได้ง่ายและสะดวก โดยเราจะศึกษาขั้นตอนและวิธีการเขียนโปรแกรมโดยใช้ Microsoft Visual Basic ดังนี้

1) ศึกษาการใช้งานแถบเครื่องมือต่าง ๆ บน Microsoft Visual Basic.Net

2) ศึกษาคำศัพท์ต่าง ๆ ที่นำมาใช้ในการเขียนโปรแกรม

3) ขอคำแนะนำจากผู้ที่มีความเชี่ยวชาญด้านการเขียนโปรแกรม Microsoft Visual Basic.Net

4) ค้นหาคำสั่งที่ใช้ในการเชื่อมโยงฐานข้อมูลกับ Microsoft Visual Basic.Net จากอินเตอร์เน็ต

สาเหตุที่เลือกศึกษาและเขียนโปรแกรมด้วย Microsoft Visual Basic.Net นั้นเนื่องมากจาก Microsoft Visual Basic.Net มีข้อดีหลายประการดังนี้

1) ง่ายต่อการเรียนรู้และเหมาะสำหรับผู้เริ่มต้น ทั้งในเรื่องไวยากรณ์ของภาษาเองและเครื่องมือการใช้งาน ดังชื่อที่บอกอยู่แล้วว่า Basic ซึ่งเหมาะสำหรับผู้เริ่มต้น

2) ความนิยมของตัวภาษา โดยอาจกล่าวได้ว่าภาษา Basic นั้นเป็นภาษาที่มีคนเรียนรู้และใช้งานมากที่สุดในประวัติศาสตร์คอมพิวเตอร์

3) การพัฒนาอย่างต่อเนื่อง การปรับปรุงประสิทธิภาพในด้านของตัวภาษาและความเร็วของการประมวลผล และในเรื่องของความสามารถใหม่ ๆ เช่น การติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต

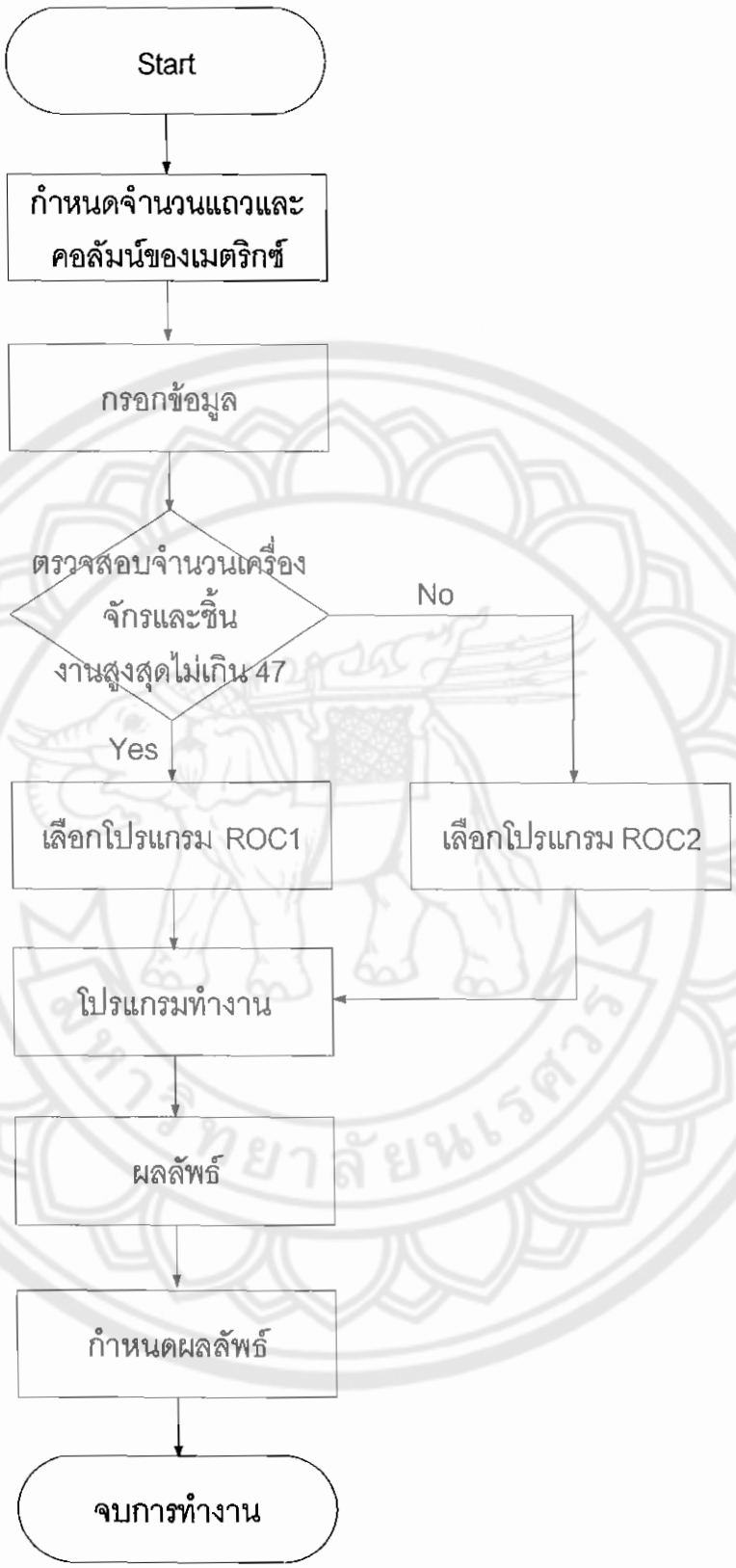
4) ผู้พัฒนาสำคัญของ Microsoft Visual Basic.Net คือบริษัท Microsoft ซึ่งจัดว่าเป็นยักษ์ใหญ่ ของวงการคอมพิวเตอร์ในปัจจุบัน เราจึงมั่นใจได้ว่า Microsoft Visual Basic.Net จะยังมีการพัฒนา ปรับปรุงและคงอยู่ไปอีกนาน

3.4 การออกแบบโปรแกรม

การออกแบบโปรแกรมนั้นเริ่มจากนำข้อมูลที่เกี่ยวข้องทั้งหมดมาพิจารณาและให้โปรแกรมทำการคำนวณผลลัพธ์ออกมา ซึ่งผลลัพธ์ที่ได้ออกมาอาจมีหลายผลลัพธ์ จึงออกแบบให้ผู้ใช้โปรแกรมสามารถกำหนดผลลัพธ์ที่ได้ออกมาด้วยตนเอง

ในการเขียนโปรแกรมจะทำการแบ่งขั้นตอนการเขียนออกเป็นแต่ละขั้นตอน การแบ่งการทำงานออกเป็นขั้นตอนและเป็นระบบจะช่วยให้ผู้ใช้โปรแกรมใช้งานได้โดยง่าย และสามารถเห็นภาพของโปรแกรมได้อย่างชัดเจน

นอกจากนี้การที่แบ่งงานแยกออกเป็นขั้นตอนย่อย ๆ จะช่วยให้ทราบว่าขณะนั้นเรากำลังอยู่ ณ จุดใดของเป้าหมาย ทำให้เราทราบว่าเราจะไปถึงจุดหมายเมื่อใด ทันเวลาและทันต่อความต้องการของผู้ใช้หรือไม่ ซึ่งจะแบ่งการออกแบบโปรแกรมเป็นขั้นตอนหลัก ๆ ตาม Flow Chart ดังรูปที่ 3.1 และคำอธิบายต่อไปนี้



รูปที่ 3.1 การออกแบบโปรแกรม

ขั้นตอนที่ 1 ให้ผู้ใช้งานกำหนดจำนวนแถวและคอลัมน์ของเมตริกซ์ที่จะใช้งาน

ขั้นตอนที่ 2 ใส่ข้อมูลของระบบการผลิต

ขั้นตอนที่ 3 ทำการตรวจสอบดูว่าจำนวนเครื่องจักรและชิ้นงานสูงที่ใช้ในการคำนวณไม่เกิน 47 เครื่องจักรหรือ 47 ชิ้นงาน โดยถ้าไม่เกินข้อจำกัดดังกล่าวผู้ใช้สามารถเลือกใช้โปรแกรม ROC1 ในการคำนวณแต่ถ้าเกินข้อจำกัดนั้นผู้ใช้ต้องเลือกใช้โปรแกรม ROC2 ในการคำนวณ

ขั้นตอนที่ 4 โปรแกรมจะทำงาน

ขั้นตอนที่ 5 แสดงผลลัพธ์

ขั้นตอนที่ 6 ผู้ใช้งานสามารถกำหนดผลลัพธ์และจัดกลุ่มหน่วยการผลิตได้เอง

3.5 **การพัฒนาและแก้ไขโปรแกรม**

ในการพัฒนาและแก้ไขโปรแกรมนั้นจะใช้คำแนะนำจากอาจารย์ที่ปรึกษาเป็นหลัก เนื่องมาจาก ROC1 และ ROC2 นั้นเป็นความรู้ที่ยังไม่แพร่หลาย ต้องค้นคว้าเท่านั้น หาผู้ที่มีประสบการณ์ได้น้อย จึงให้อาจารย์ที่ปรึกษาเป็นผู้ให้คำติชมและนำไปปรับปรุงโปรแกรมและนำเสนอใหม่