บทที่ 7

การประมวลผลแบบขนานสำหรับปัญหาการสังเคราะห์เครื่องสถานะจำกัด

ในบทนี้นำเสนอการทดลองการประมวลผลแบบขนานของปัญหาที่สาม ปัญหานี้มีการศึกษาในงานวิจัย [3] ซึ่งเกี่ยวข้องกับการสังเคราะห์เครื่องสถานะจำกัดโดยขั้นตอนวิธีเชิงพันธุกรรม การนำเสนอประกอบด้วยลักษณะและค่ากำหนดของปัญหาจากงานวิจัยก่อน การออกแบบการประมวลผลแบบขนานและผลการทดลอง จากนั้นเป็นการวิเคราะห์ผล และส่วนสุดท้ายเป็นการสรุป

7.1 ลักษณะปัญหาการสังเคราะห์เครื่องสถานะจำกัด

เครื่องสถานะจำกัด (Finite State Machine หรือ FSM) สามารถสร้างได้ด้วยวิธีที่เป็นระบบ ถ้าเข้าใจการทำงานในลักษณะของแผนภาพสถานะ (state diagram) จากงานวิจัย [3] เสนอวิธีการสร้างเครื่องสถานะจำกัดที่ต่างออกไป คือ สร้างเครื่องสถานะจำกัดเมื่อไม่สามารถรู้ลักษณะการทำงาน โดยจะใช้การสังเกตจากสัญญาณขาเข้า และสัญญาณขาออก วิธีการนี้เหมาะกับบางกรณี เช่น เครื่องสถานะจำกัดที่ไม่สามารถสังเกตสถานะภายในจากผู้สังเกตภายนอก

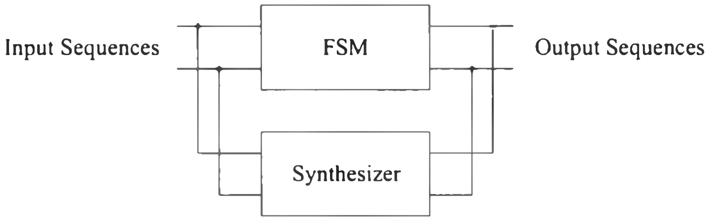
แนวความคิดของงานวิจัยนี้คือ การสร้างเครื่องสังเคราะห์ที่ต่อเข้ากับสัญญาณขาเข้า และสัญญาณขาออกของเครื่องสถานะจำกัดที่เราต้องการ ดังรูปที่ 7.1 จากนั้นเมื่อปล่อยให้เครื่องสถานะจำกัดทำงาน เครื่องสังเคราะห์จะทำการสังเกตสัญญาณขาเข้า และสัญญาณขาออก และเปลี่ยนตัวเองให้สามารถทำงานได้เหมือนกับเครื่องสถานะจำกัดที่ใช้เป็นตัวอย่าง

7.1.1 การใช้ขั้นตอนวิธีเชิงพันธุกรรมกับปัญหาการสังเคราะห์เครื่องสถานะจำกัด

การสังเคราะห์จะใช้ขั้นตอนวิธีเชิงพันธุกรรมในลักษณะประชากรขนาดเล็ก และมีจำนวนรุ่นเป็นจำนวนมาก โดยขั้นตอนการเลือกประชากรรุ่นใหม่จำเป็นต้องรักษาความหลากหลายด้วย ประชากรแต่ละตัวจะแทนเครื่องสถานะจำกัดในลักษณะตารางการเปลี่ยนสถานะ (state transition table) การเก็บข้อมูลในโครโมโซมจะเก็บสถานะถัดไป และสัญญาณขาออก ดังตัวอย่างในตาราง 7.1 ของวงจรบวกแบบเชิงลำดับ (serial adder) โครโมโซมจะเก็บข้อมูลเป็น “0001011001101011”

ในการใช้งานจริงจำนวนสถานะของเครื่องสถานะจำกัดเป็นสิ่งที่ไม่สามารถทราบได้ ดังนั้นในการทดลอง จำนวนสถานะของประชากรจะกำหนดให้มีค่ามากกว่าจำนวนสถานะของเครื่องสถานะจำกัดที่เป็นเป้าหมาย การกำหนดจำนวนสถานะที่เกินมานี้ทำให้คำตอบที่ได้มีจำนวนสถานะที่มากกว่าปกติ สถานะที่เกินมานี้เป็นไปได้สองแบบดังนี้ แบบแรกเป็นสถานะที่ไปไม่ถึงคือไม่มีการเชื่อมต่อมาจากสถานะอื่นๆ ในการทำงานสถานะนี้จึงไม่ได้ถูกใช้ และสามารถตัดทิ้งได้ แบบที่สองเป็นสถานะที่ซ้ำซ้อนคือมีสถานะตั้งแต่ 2 สถานะที่ทำงานเหมือนกัน และสามารถยุบรวมกันได้หลังจากที่ได้คำตอบ สถานะที่เกินเหล่านี้สามารถเอาออกจากคำตอบได้

การสร้างชุดสัญญาณขาเข้า และสัญญาณขาออกเพื่อใช้ในการหาค่าความเหมาะสมจะทำโดยกำหนดให้สถานะของเครื่องสถานะจำกัดที่ใช้เป็นเป้าหมายเป็นสถานะเริ่มต้น จากนั้นทำการสุ่มชุดสัญญาณขาเข้า แล้วป้อนสัญญาณขาเข้าไปยังเครื่องสถานะจำกัด และทำการเก็บชุดสัญญาณขาออกที่ได้



รูปที่ 7.1 การสังเคราะห์เครื่องสถานะจำกัด

ในการวัดค่าความเหมาะสมจะทำการกำหนดสถานะของประชากรแต่ละตัวให้เป็นสถานะเริ่มต้น จากนั้นนำชุดสัญญาณขาเข้าที่สร้างไว้ ป้อนเข้าไปประชากรแต่ละตัว แล้วนำสัญญาณขาออกมาเทียบกับสัญญาณขาออกของเครื่องสถานะจำกัดที่ใช้เป็นเป้าหมาย ค่าความเหมาะสมจะเป็นจำนวนบิตที่เหมือนกัน ประชากรที่เป็นคำตอบจะเป็นประชากรที่ให้สัญญาณขาออกเหมือนกับเครื่องสถานะจำกัดที่เป็นเป้าหมายทุกชุดสัญญาณขาเข้าที่สร้างเอาไว้

จากงานวิจัย [32] พบว่าคำตอบที่ได้สามารถแบ่งได้เป็นสองประเภท คือ

1. **คำตอบสมบูรณ์** เป็นคำตอบที่ทำงานได้ถูกต้องกับชุดสัญญาณขาเข้า และสัญญาณขาออกทุกแบบที่เป็นไปได้

2. **คำตอบที่ไม่สมบูรณ์** เป็นคำตอบที่ทำงานได้ถูกต้องกับชุดสัญญาณขาเข้า และสัญญาณขาออกที่ใช้ในขั้นตอนวิธีเชิงพันธุกรรม แต่ไม่สามารถทำงานได้ถูกต้องกับชุดสัญญาณขาเข้า และสัญญาณขาออกทุกแบบที่เป็นไปได้

งานวิจัย [32] จะใช้ชุดสัญญาณขาเข้าและสัญญาณขาออกเพียงชุดเดียวในการทดลองพบว่า "จำนวน" ของคำตอบสมบูรณ์เพิ่มขึ้นตาม "ความยาว" ของชุดสัญญาณขาเข้าและสัญญาณขาออก แต่ก็ไม่สามารถทำให้ได้จำนวนคำตอบสมบูรณ์เป็น 100 เปอร์เซ็นต์จากจำนวนคำตอบทั้งหมดเมื่อใช้สังเคราะห์เครื่องสถานะจำกัดที่มีความซับซ้อนมาก ในงานวิจัย [3] จึงเสนอการปรับปรุงโดยใช้ชุดสัญญาณขาเข้าและสัญญาณขาออก "หลายชุด" เพื่อเพิ่มจำนวนคำตอบสมบูรณ์ ผลการทดลองแสดงให้เห็นว่า เมื่อใช้จำนวนชุดสัญญาณขาเข้าและสัญญาณขาออกหลายชุดเพียงพอ สามารถเพิ่มจำนวนคำตอบสมบูรณ์เป็น 100 เปอร์เซ็นต์

7.2 การออกแบบการประมวลผลแบบขนานและผลการทดลอง

การแก้ปัญหาการสังเคราะห์เครื่องสถานะจำกัดจะแตกต่างจากปัญหาการนำร่องหุ่นยนต์ นอกจากประชากรที่ใช้จะมีขนาดเล็ก และใช้จำนวนรุ่นในการหาคำตอบเป็นจำนวนมาก ปัญหาการนำร่องหุ่นยนต์จำนวนรุ่นจะคงที่ ในขณะที่ปัญหาการสังเคราะห์เครื่องสถานะจำกัดจะดำเนินจนกระทั่งสามารถหาเครื่องสถานะจำกัดที่ถูกต้องตามชุดสัญญาณขาเข้า และสัญญาณขาออกได้ ทำให้จำนวนรุ่นไม่คงที่

เนื่องจากประชากรมีขนาดเล็ก โดยในงานวิจัย [3] ขนาดประชากรเท่ากับ 400 ตัว ดังนั้นถ้าใช้การแบ่งประชากรให้เล็กลงเมื่อประมวลผลแบบขนานจะทำให้เกิดปัญหาได้ เนื่องจากขนาดประชากรไม่เพียงพอสำหรับการทำงานของขั้นตอนวิธีเชิงพันธุกรรม ในการทดลองแบบแรกจึงเป็นการใช้ขนาดประชากรในแต่หน่วยประมวลผลเท่ากับในงานวิจัย [3] เช่น เมื่อประมวลผลแบบขนานโดยมี 4 หน่วยประมวลผล แต่ละ

ตารางที่ 7.1 ตารางการเปลี่ยนสถานะของวงจรบวกแบบเชิงลำดับ

State	Input	Next State	Output
0	00	0	0
0	01	0	1
0	10	0	1
0	11	1	0
1	00	0	1
1	01	1	0
1	10	1	0
1	11	1	1

หน่วยประมวลผลจะมีประชากรเท่ากับ 400 ดังนั้นประชากรรวมสำหรับการประมวลผลแบบขนานโดยใช้ 4 หน่วยประมวลผล จะเท่ากับ 1600

ในการทำงานแบบนี้ความเร็วที่เพิ่มขึ้นไม่ได้มาจากขนาดประชากรที่เล็กลง แต่จะอาศัยการเพิ่มหน่วยประมวลผลช่วยเพิ่มโอกาสที่จะพบคำตอบในจำนวนรุ่นที่น้อยกว่า และนอกจากนี้ประสิทธิภาพในแง่โอกาสที่จะพบคำตอบน่าจะเพิ่มขึ้นด้วย ซึ่งในการทำงานของขั้นตอนวิธีเชิงลำดับ เมื่อสังเคราะห์วงจรที่ซับซ้อนจะไม่ได้คำตอบทุกครั้ง

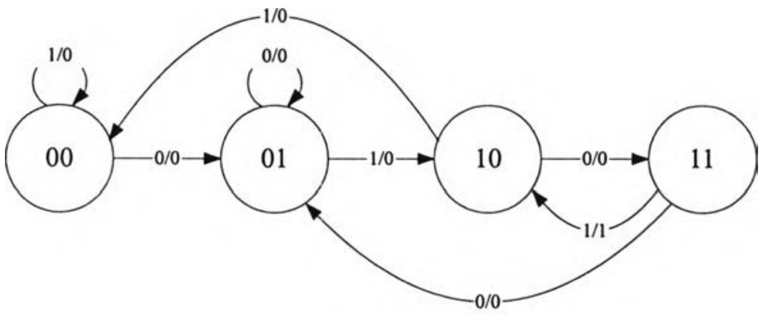
นอกจากนี้ในการทดลองได้ลดจำนวนประชากรในแต่ละหน่วยประมวลผลลงเป็น 200 และ 100 เพื่อทดสอบว่าเมื่อขนาดประชากรเล็กลง ประสิทธิภาพในการประมวลผลแบบขนานเทียบกับการประมวลผลแบบเชิงลำดับจะเป็นอย่างไร

7.2.1 การออกแบบการทดลอง

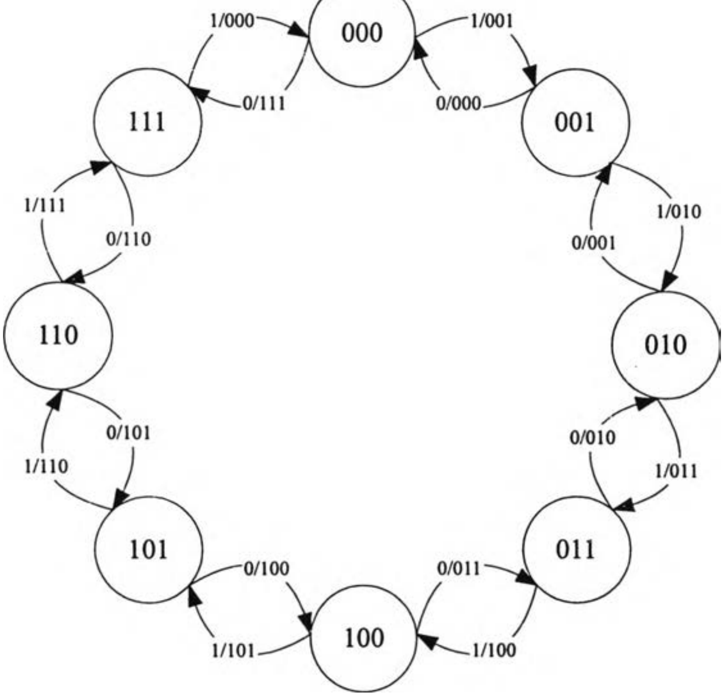
โปรแกรมที่ใช้ในการทดลองดัดแปลงมาจากโปรแกรมที่ใช้ในงานวิจัย [3] การอพยพจะดำเนินการในลักษณะวงแหวนแบบไม่ประสานเวลา เหมือนในการทดลองส่วนที่สองของปัญหาการนำร่องหุ่นยนต์โดยการเพิ่มจำนวนสิ่งแวดล้อม เนื่องจากการทำงานจะใช้จำนวนรุ่นเป็นจำนวนมาก ดังนั้นในการทดลองนี้จึงลดความถี่ของการอพยพลงเหลือทุก 10 รุ่นจะทำการอพยพหนึ่งครั้ง โดยเมื่อหน่วยประมวลผลใดทำงานจนครบ 10 รุ่นก่อน จะทำการส่งข้อความเพื่อขอทำการอพยพให้กับทุกหน่วยประมวลผล ในแต่ละครั้ง จำนวนประชากรในการอพยพจะเท่ากับ 5

วงจรที่ใช้ในการทดลองเลือกจากวงจรที่ใช้ทดลองในงานวิจัย [3] มา 2 วงจร โดยมีวงจร Reversible 8-Counter ซึ่งแทนวงจรขนาดใหญ่ และวงจร 0101 Detector แทนวงจรขนาดเล็ก แผนภาพสถานะของทั้งสองวงจรแสดงในรูป 7.2 สำหรับพารามิเตอร์ของวงจร และจำนวนของประชากรที่สร้างโดยการเปลี่ยนไขว้, การกลายพันธุ์, การทำซ้ำ กำหนดเช่นเดียวกับงานวิจัย [3] ดังตาราง 7.2 แสดงพารามิเตอร์ของวงจร และตาราง 7.3 แสดงจำนวนของประชากรที่สร้างตัวดำเนินการต่างๆ

จำนวนชุดสัญญาณขาเข้าและสัญญาณขาออกเท่ากับ 100 เพราะต้องการในทุกคำตอบที่ได้เป็นคำตอบ



I. 0101 Detector



II. Reversible 8-Counter

รูปที่ 7.2 แผนภาพสถานะของวงจร Reversible 8-Counter และวงจร 0101 Detector

ตารางที่ 7.2 พารามิเตอร์ของวงจรที่ใช้ในการทดลอง

วงจร	สัญญาณขาเข้า (บิต)	สัญญาณขาออก (บิต)	จำนวนสถานะภายใน	จำนวนสถานะภายในสูงสุดที่มีได้
Reversible 8-Counter	1	3	8	32
0101 Detector	1	1	4	8

ตารางที่ 7.3 จำนวนของประชากรที่สร้างตัวดำเนินการ

จำนวนประชากร	การเปลี่ยนไขว้	การกลายพันธุ์	การทำซ้ำ
400	200	100	100
200	100	50	50
100	50	25	25

ที่สมบูรณ์ เพื่อให้การแปลความหมายของผลการทดลองทำได้ง่ายขึ้น เนื่องจากจำนวนของคำตอบสมบูรณ์และไม่สมบูรณ์ไม่ได้เกี่ยวข้องกับประสิทธิภาพการทำงานในแง่ที่ศึกษาอยู่ในวิทยานิพนธ์ แต่จะเกี่ยวข้องกับจำนวนชุดสัญญาณขาเข้าและสัญญาณขาออกที่ต้องใช้เพื่อให้คำตอบที่ได้ถูกต้อง

7.2.2 ผลการทดลอง

การทดลองแต่ละชุดทำซ้ำ 10 ครั้งเท่ากับที่ใช้ในงานวิจัย [3] จำนวนครั้งของการให้คำตอบเป็นค่าหนึ่งที่ใช้ในการพิจารณาประสิทธิภาพของขั้นตอนวิธี ในตาราง 7.4 และ 7.5 แสดงจำนวนครั้งของการให้คำตอบสำหรับวงจรทั้งสอง สำหรับการประมวลผลแบบเชิงลำดับจำนวนครั้งของการให้คำตอบจะลดลงเมื่อลดขนาดของประชากรลง โดยเห็นได้ชัดเมื่อใช้วงจร Reversible 8-Counter ซึ่งเป็นวงจรขนาดใหญ่ เนื่องจากประชากรมีขนาดเล็กเกินไปเมื่อเทียบกับความยากของปัญหา ทำให้โอกาสประสบความสำเร็จในการหาคำตอบลดลง

สำหรับการประมวลผลแบบขนาน การลดขนาดประชากรไม่มีผลกับจำนวนครั้งของการให้คำตอบเมื่อใช้กับวงจรขนาดเล็กคือวงจร 0101 Detector และเมื่อใช้กับวงจร Reversible 8-Counter การใช้ประชากร 400 สามารถหาคำตอบได้ทุกครั้ง เมื่อลดประชากรลงเหลือ 200 ก็ยังให้คำตอบได้ทุกครั้ง ยกเว้นเมื่อใช้หน่วยประมวลผลเท่ากับ 6 หน่วยเท่านั้นที่ทำให้จำนวนครั้งของการให้คำตอบเป็น 9 ครั้ง ซึ่งก็ยังมากกว่าการใช้ประชากร 400 ของประมวลผลแบบเชิงลำดับ และเมื่อลดประชากรลงเหลือ 100 การประมวลผลแบบขนานเริ่มแสดงให้เห็นว่าขนาดประชากรไม่เพียงพอต่อการหาคำตอบ

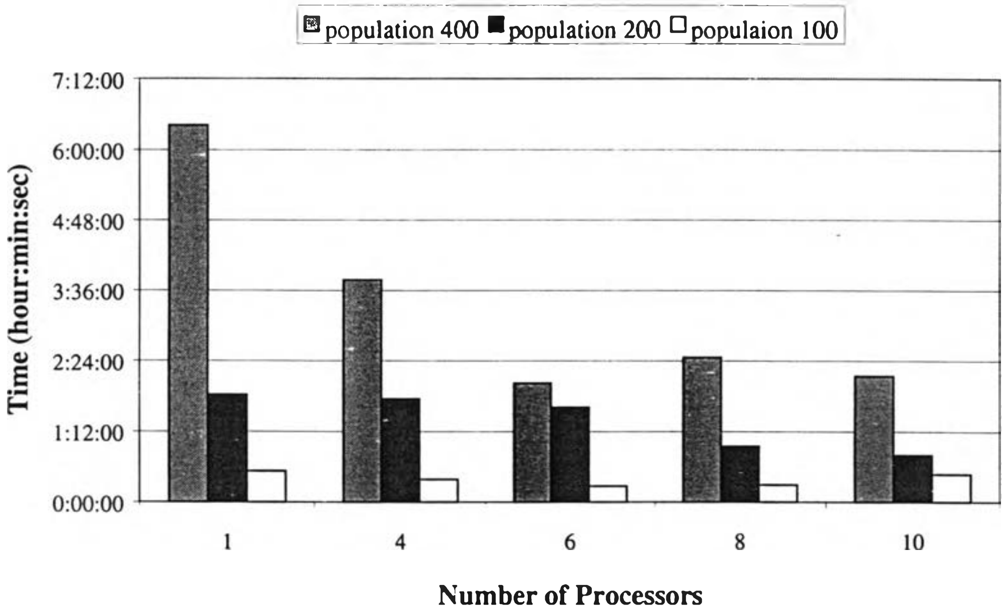
เวลาในการหาคำตอบสำหรับการทำงานที่พบคำตอบแสดงในรูป 7.3 และ 7.4 เมื่อทำการเปรียบเทียบเวลาในการหาคำตอบของการประมวลผลแบบขนาน กับการประมวลผลแบบเชิงลำดับ จะเห็นได้ว่าสำหรับวงจร 0101 Detector เวลาจะลดลงอย่างเห็นได้ชัด และเวลาในการหาคำตอบลดลงเมื่อเพิ่มหน่วยประมวลผลสำหรับวงจร Reversible 8-Counter เวลาของการประมวลผลแบบขนานน้อยกว่าแบบเชิงลำดับ แต่การเพิ่มหน่วยประมวลผลไม่ได้ลดเวลาในการหาคำตอบอย่างชัดเจน โดยเฉพาะในช่วงจาก 6 หน่วยประมวลผลไปจน

ตารางที่ 7.4 จำนวนครั้งของการให้คำตอบในการสังเคราะห์วงจร Reversible 8-Counter

จำนวนหน่วยประมวลผล	จำนวนครั้งของการให้คำตอบเมื่อใช้จำนวนประชากร		
	400	200	100
1	8	6	1
4	10	10	7
6	10	9	5
8	10	10	2
10	10	10	6

ตารางที่ 7.5 จำนวนครั้งของการให้คำตอบในการสังเคราะห์วงจร 0101 Detector

จำนวนหน่วยประมวลผล	จำนวนครั้งของการให้คำตอบเมื่อใช้จำนวนประชากร		
	400	200	100
1	10	9	9
4	10	10	10
6	10	10	10
8	10	10	10
10	10	10	10



รูปที่ 7.3 เวลาในการหาคำตอบในการสังเคราะห์วงจร Reversible 8-Counter

ถึง 10 หน่วยประมวลผล เมื่อใช้ประชากร 400 และ 100 ตัว เวลาที่ใช้ไม่ได้ลดลง ทั้งนี้เป็นไปได้ว่าวงจร Reversible 8-Counter เป็นปัญหาที่มีขนาดใหญ่มาก เมื่อเทียบกับการเพิ่มหน่วยประมวลผลเพียงเล็กน้อย จึงไม่สามารถทำให้เกิดความแตกต่าง

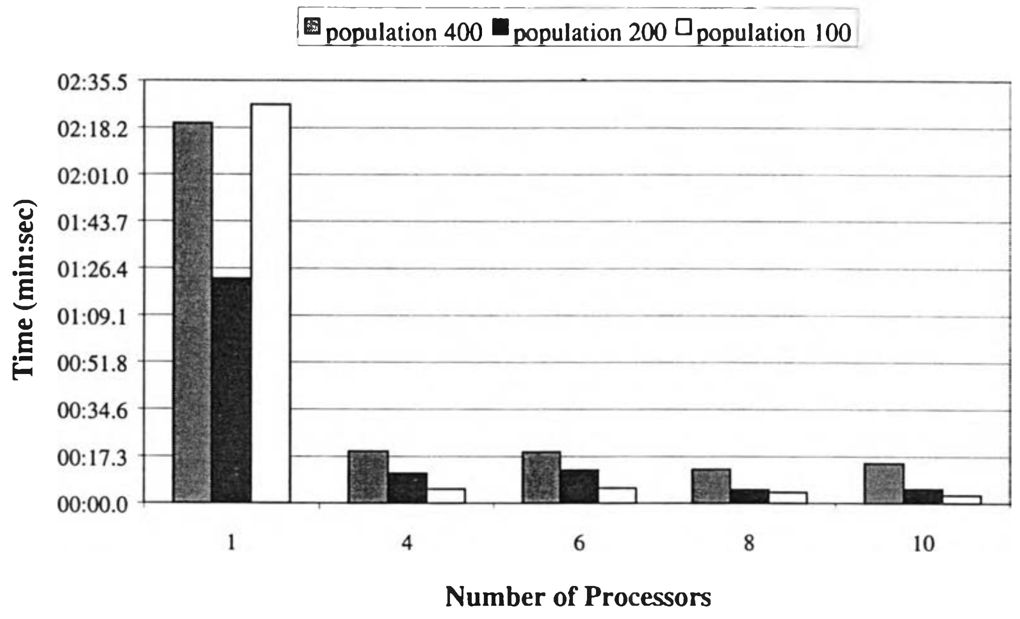
กราฟของจำนวนรุ่นที่พบคำตอบกับเวลาที่ใช้ในการหาคำตอบของการสังเคราะห์วงจร Reversible 8-Counter แสดงในรูป 7.5 และของการสังเคราะห์วงจร 0101 Detector แสดงในรูป 7.6 โดยเส้นที่แสดงในกราฟเป็นเส้นแนวโน้มแบบเชิงเส้น จากกราฟจะเห็นได้ว่าจำนวนรุ่นที่พบคำตอบมีความสัมพันธ์กับเวลาที่ใช้ในการหาคำตอบ การเพิ่มขึ้นของจำนวนรุ่นทำให้เวลาในการหาคำตอบเพิ่มขึ้นตาม นอกจากนี้การลดขนาดประชากรทำให้เวลาที่ใช้มีการลดลงเป็นอย่างมาก ซึ่งอธิบายเพิ่มเติมในหัวข้อ 7.3.1

กราฟ 7.7 และ 7.8 แสดงจำนวนรุ่นที่พบคำตอบสำหรับวงจร Reversible 8-Counter และ 0101 Detector ในวงจร Reversible 8-Counter เมื่อใช้ประชากร 400 และ 100 ตัว ช่วงที่เพิ่มหน่วยประมวลผลจาก 6 หน่วยประมวลผลไปจนถึง 10 หน่วยประมวลผล เวลาที่ใช้ในการหาคำตอบไม่ได้ลดลง เนื่องจากจำนวนรุ่นที่พบคำตอบไม่ได้ลดลง

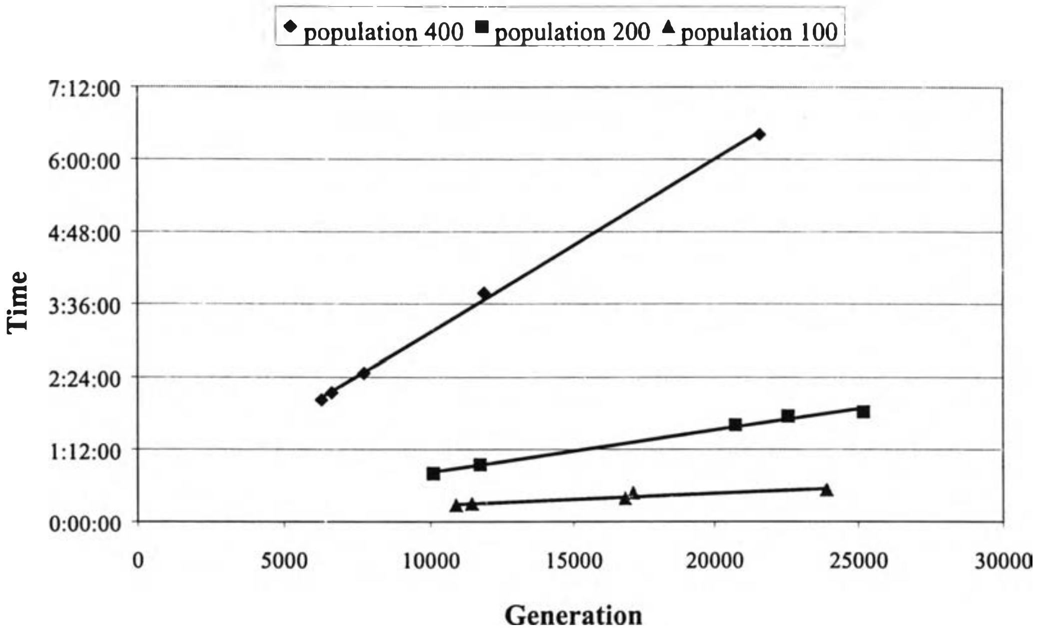
7.3 การวิเคราะห์ผล

7.3.1 ขนาดของประชากรกับเวลาในการหาคำตอบ

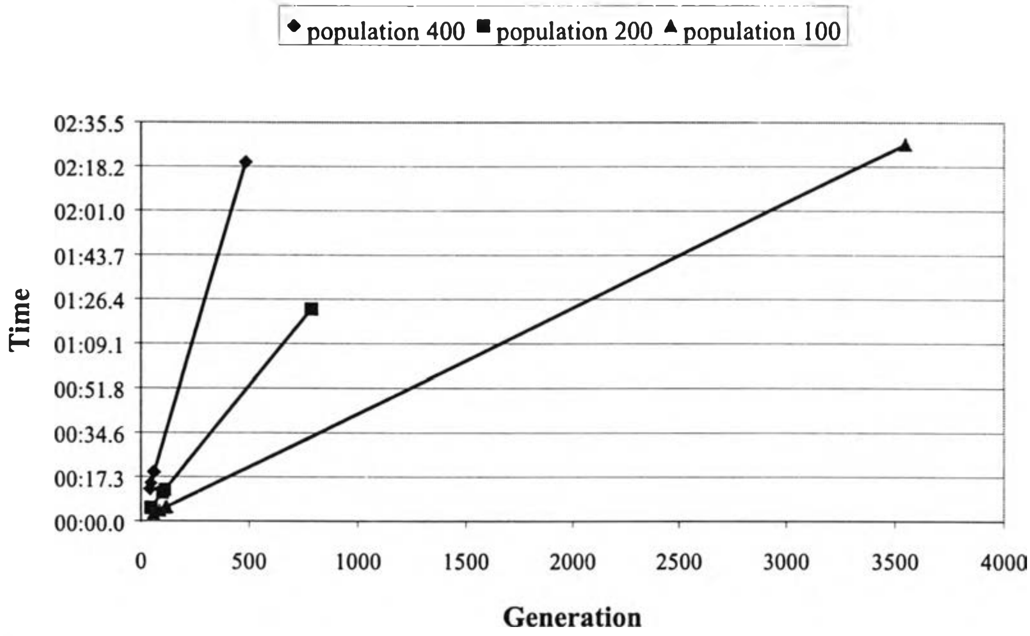
ในการลดขนาดของประชากรสำหรับการสังเคราะห์วงจร Reversible 8-Counter เมื่อใช้การประมวลผลแบบเชิงลำดับ จำนวนรุ่นที่พบคำตอบมีการเปลี่ยนแปลงไม่มากนัก แต่เวลาในการหาคำตอบลดลงอย่างรวดเร็ว จากกราฟ 7.3 เมื่อลดประชากรจาก 400 เป็น 200 ตัว เวลาในการหาคำตอบในหน่วย (ชั่วโมง:นาที:วินาที) จาก 6:25:01 ลดลงเป็น 1:49:31 การหาคำตอบเร็วขึ้น 3.52 เท่า และเมื่อลดประชากรจาก 400 เป็น 100 ตัว เวลาในการหาคำตอบลดลงเป็น 0:31:47 การหาคำตอบเร็วขึ้น 12.11 เท่า



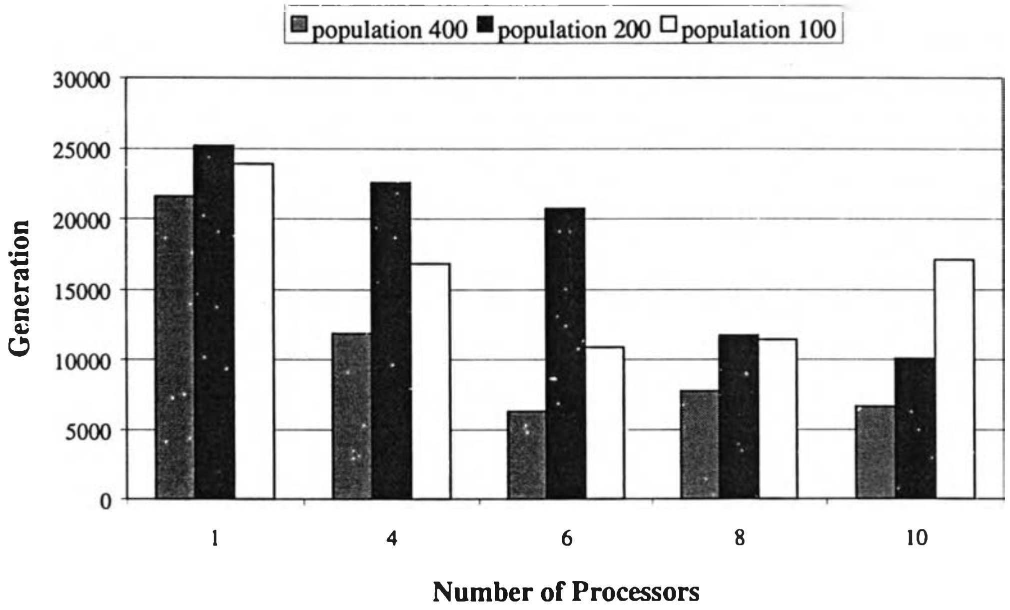
รูปที่ 7.4 เวลาในการหาคำตอบในการสังเคราะห์วงจร 0101 Detector



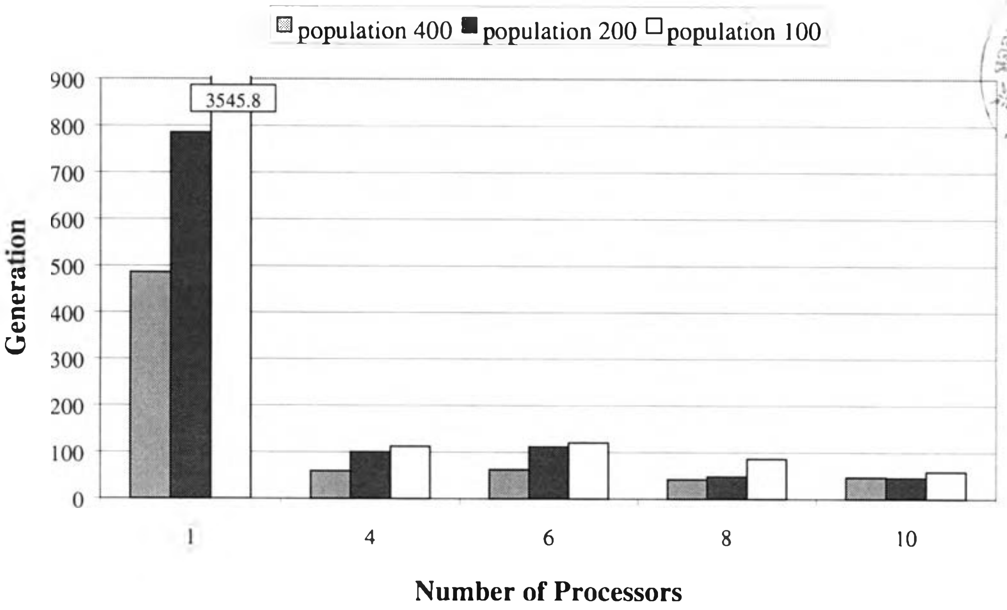
รูปที่ 7.5 ความสัมพันธ์ระหว่างจำนวนรุ่น และเวลาในการหาคำตอบของการสังเคราะห์วงจร Reversible 8-Counter



รูปที่ 7.6 ความสัมพันธ์ระหว่างจำนวนรุ่น และเวลาในการหาคำตอบของการสังเคราะห์วงจร 0101 Detector



รูปที่ 7.7 จำนวนรุ่นที่พบคำตอบในการสังเคราะห์วงจร Reversible 8-Counter



รูปที่ 7.8 จำนวนรุ่นที่พบคำตอบในการสังเคราะห์วงจร 0101 Detector

การลดลงของเวลาในการหาคำตอบเป็นอัตราส่วนที่สูง เมื่อเทียบกับขนาดประชากรที่ลดลง เนื่องมาจากลักษณะของกระบวนการคัดเลือกประชากรที่มีการรักษาความหลากหลาย โดยการคัดเลือกที่ใช้ในงานวิจัย [3] ดัดแปลงมาจากหนังสือ [33] การคัดเลือกประชากรจะพิจารณาในแง่คุณภาพของประชากรโดยใช้ค่าความเหมาะสม และความแตกต่างจากประชากรที่ได้เลือกไว้แล้ว ในแต่ละรอบของการคัดเลือก หลังจากทำการคัดเลือกประชากรไปครั้งละ 1 ตัว ต้องมีการจัดลำดับประชากรที่ยังไม่ได้ถูกเลือก และจะใช้ในการคัดเลือกในรอบถัดไปใหม่ทุกครั้ง การทำงานในลักษณะนี้เสียเวลาเป็น $O(n^2 \log n)$ โดย n คือ ขนาดประชากร ดังนั้นเวลาที่ใช้จึงมีการลดลงอย่างรวดเร็วเมื่อขนาดประชากรลดลง

สำหรับการสังเคราะห์วงจร 0101 Detector ด้วยการประมวลผลแบบเชิงลำดับ การลดขนาดประชากรทำให้เวลาลดลงไม่มากนักเมื่อเทียบกับการสังเคราะห์วงจร Reversible 8-Counter จากกราฟ 7.4 เมื่อลดประชากรจาก 400 เป็น 200 ตัว เวลาในการหาคำตอบในหน่วย (นาที:วินาที) จาก 02:19.9 ลดลงเป็น 01:22.6 การหาคำตอบเร็วขึ้น 1.69 เท่า และเมื่อลดประชากรจาก 400 เป็น 100 ตัว การหาคำตอบกลับช้าลงทั้งนี้เนื่องมาจากจำนวนรุ่นที่พบคำตอบมีการเพิ่มขึ้นเป็นอย่างมาก

ในการประมวลผลแบบขนานสำหรับการสังเคราะห์วงจรที่ใช้ในการทดลองทั้งสองวงจร จำนวนรุ่นที่ใช้ในการหาคำตอบเพิ่มขึ้นไม่มากนักเมื่อมีการลดขนาดประชากร นอกจากนี้จากกราฟ 7.3 และ 7.4 การลดขนาดประชากรทำให้เวลาในการหาคำตอบลดลงอย่างชัดเจน แสดงให้เห็นว่าเวลาในการคัดเลือกมีการลดลงเป็นจำนวนมาก เมื่อเทียบกับเวลาที่เพิ่มเนื่องจากจำนวนรุ่นที่เพิ่มขึ้น ดังนั้นการประมวลผลแบบขนานจึงสามารถใช้ประโยชน์จากการลดขนาดประชากรลงเพื่อเพิ่มความเร็วในการทำงาน ถ้าใช้จำนวนครั้งของการให้คำตอบเป็นเกณฑ์ ขนาดประชากรต่อหน่วยประมวลผลที่เหมาะสมของการทำงานแบบขนานสำหรับการสังเคราะห์วงจร Reversible 8-Counter คือ 200 ตัว และสำหรับการสังเคราะห์วงจร 0101 Detector คือ 100 ตัว

7.3.2 ค่าเวลาสัมพัทธ์ของการประมวลผลแบบขนาน

จากสองบทที่ผ่านมา การเปรียบเทียบประสิทธิภาพของการประมวลผลแบบเชิงลำดับ และการประมวลผลแบบขนานใช้เงื่อนไขที่คุณภาพคำตอบต้องเท่ากัน โดยคุณภาพของคำตอบพิจารณาจากความทนทานของโปรแกรมควบคุมหุ่นยนต์ ในการทดลองชุดนี้ไม่สามารถเปรียบเทียบคุณภาพคำตอบได้ เนื่องจากคำตอบที่ได้เป็นคำตอบสมบูรณ์ทุกครั้ง ดังนั้นในการทดลองชุดนี้ใช้ความสามารถในการให้คำตอบเป็นเงื่อนไขในการเปรียบเทียบประสิทธิภาพระหว่างการประมวลผลแบบเชิงลำดับ และการประมวลผลแบบขนาน

ค่าเวลาสัมพัทธ์เป็นการเปรียบเทียบกับเวลาของการประมวลผลแบบเชิงลำดับ เมื่อใช้ประชากร 400 ตัว โดยจำนวนการให้คำตอบของการประมวลผลแบบขนานที่เลือกมาเพื่อทำการคำนวณต้องไม่น้อยกว่าแบบเชิงลำดับ ในการสังเคราะห์วงจร Reversible 8-Counter ด้วยการประมวลผลแบบขนาน ขนาดประชากรที่จำนวนการให้คำตอบไม่น้อยกว่าการประมวลผลแบบเชิงลำดับ คือ ประชากร 200 ตัวต่อหน่วยประมวลผล ค่าเวลาสัมพัทธ์ที่คำนวณได้แสดงในรูป 7.9

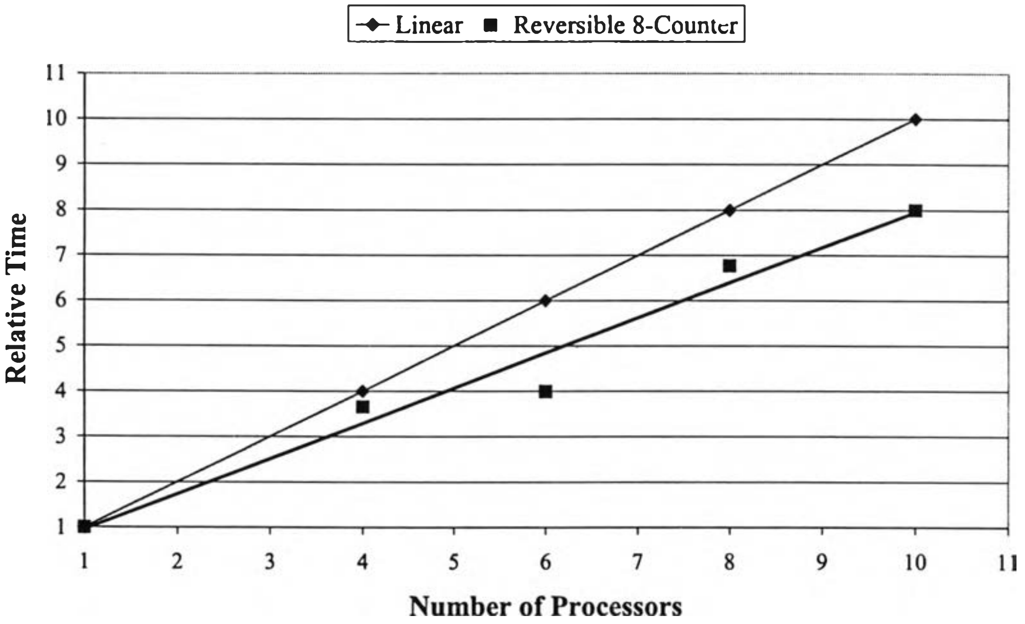
สำหรับการสังเคราะห์วงจร 0101 Detector ด้วยการประมวลผลแบบขนาน ขนาดประชากรที่จำนวนการให้คำตอบไม่น้อยกว่าการประมวลผลแบบเชิงลำดับ คือ ประชากร 100 ตัวต่อหน่วยประมวลผล รูป 7.10 แสดงค่าเวลาสัมพัทธ์ที่คำนวณได้ในการสังเคราะห์วงจร 0101 Detector

จากกราฟของค่าเวลาสัมพัทธ์ของการสังเคราะห์วงจร เมื่อใช้การประมวลผลแบบขนานกับการสังเคราะห์วงจรขนาดเล็ก คือ วงจร 0101 Detector ค่าเวลาสัมพัทธ์ที่ได้มีค่ามากกว่าเมื่อใช้กับวงจรขนาดใหญ่ คือ วงจร Reversible 8-Counter นอกจากนี้ในวงจรขนาดเล็กค่าเวลาสัมพัทธ์ที่ได้สูงกว่าจำนวนหน่วยประมวลผลเป็นจำนวนมาก ประสิทธิภาพที่ได้นี้เนื่องมาจาก 2 สาเหตุ คือการเพิ่มหน่วยประมวลผลทำให้มีโอกาสที่เจอคำตอบในจำนวนรุ่นที่น้อยกว่า และการประมวลผลแบบขนานทำให้สามารถใช้ประชากรที่มีขนาดเล็กลง โดยที่จำนวนรุ่นที่ใช้ในการหาคำตอบไม่เพิ่มขึ้นมากนัก การลดขนาดประชากรลงนี้ทำให้การหาคำตอบเป็นไปอย่างรวดเร็ว ดังที่ได้อธิบายในส่วน 7.3.1

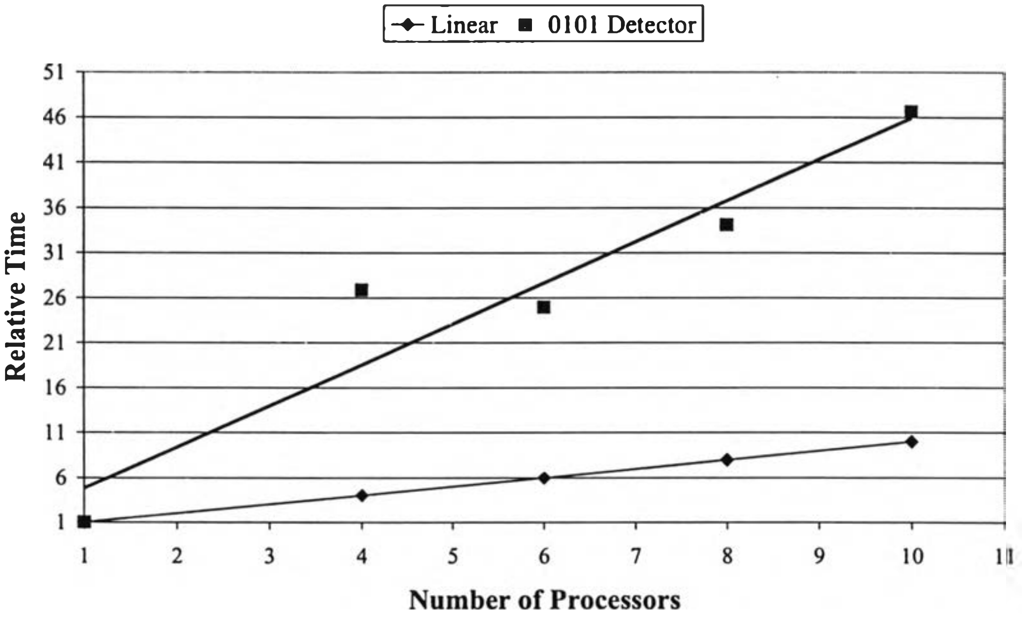
7.4 สรุปท้ายบท

การทดลองชุดนี้มีความแตกต่างจากการทดลองชุดอื่น โดยใน 2 ปัญหาที่ผ่านมา ประสิทธิภาพการประมวลผลแบบขนานที่ได้ส่วนหนึ่งมาจากการแบ่งประชากรให้กับแต่ละหน่วยประมวลผล และเมื่อเพิ่มจำนวนหน่วยประมวลผลจะทำให้ประชากรในแต่ละหน่วยประมวลผลมีจำนวนลดลง แต่การทดลองนี้ ประชากรแต่ละกลุ่มมีขนาดเล็ก การแบ่งประชากรเพื่อประมวลผลแบบหน่วยหยาบอาจทำให้จำนวนประชากรต่อหน่วยประมวลผลมีขนาดเล็กเกินไป ไม่สามารถทำงานได้ ดังนั้นในการทดลองนี้จึงไม่ใช้วิธีแบ่งประชากรที่ทำให้เกิดการลดจำนวนประชากรตามการเพิ่มของหน่วยประมวลผล ในช่วงแรกแต่ละหน่วยประมวลผลใช้ขนาดประชากรเหมือนกับการทำงานแบบเชิงลำดับ โดยประสิทธิภาพการประมวลผลแบบขนานที่ได้เนื่องมาความได้เปรียบในเชิงของความน่าจะเป็นที่จะได้คำตอบจำนวนรุ่นที่ต่ำกว่า ต่อมาได้ทำการทดลองเพื่อเพิ่มประสิทธิภาพการทำงานแบบขนานโดยลดจำนวนประชากรต่อหน่วยประมวลผลจาก 400 ตัวเป็น 200 และ 100 ตัว

การทดลองของปัญหาการนำร่องหุ่นยนต์ใช้ความทนทานแทนคุณภาพของคำตอบ และทำการเปรียบเทียบคุณภาพคำตอบก่อนที่จะมีการเปรียบเทียบประสิทธิภาพการประมวลผลแบบขนาน เนื่องจากการทดลองชุดนี้ไม่สามารถนิยามคุณภาพคำตอบได้ ดังนั้นจึงใช้ความสามารถในการให้คำตอบเป็นตัวเปรียบเทียบ



รูปที่ 7.9 ค่าเวลาสัมพัทธ์ของการสังเคราะห์วงจร Reversible 8-Counter



รูปที่ 7.10 ค่าเวลาสัมพัทธ์ของการสังเคราะห์วงจร 0101 Detector

ว่างานที่ทำในการประมวลผลแบบเชิงลำดับ และแบบขนานเท่ากัน จากผลการทดลองแสดงให้เห็นว่าในการสังเคราะห์วงจร Reversible 8-Counter ความสามารถในการให้คำตอบของการประมวลผลแบบขนานเมื่อใช้ประชากร 200 ตัวต่อหน่วยประมวลผลเทียบได้กับการประมวลผลแบบเชิงลำดับเมื่อใช้ประชากร 400 ตัว และในการสังเคราะห์วงจร 0101 Detector ความสามารถในการให้คำตอบของการประมวลผลแบบขนานเมื่อใช้ประชากร 100 ตัวต่อหน่วยประมวลผลเทียบได้กับการประมวลผลแบบเชิงลำดับเมื่อใช้ประชากร 400 ตัว

เวลาในการทำงานของการประมวลผลแบบขนานมีการลดลงเมื่อเพิ่มจำนวนหน่วยประมวลผล แต่ในบางช่วงของปัญหาขนาดใหญ่อย่าง Reversible 8-Counter การเพิ่มจำนวนหน่วยประมวลผลไม่ทำให้เวลาในการทำงานลดลง ทั้งนี้เป็นไปได้ว่าปัญหามีขนาดใหญ่มากเมื่อเพิ่มประชากรเพียงเล็กน้อยทำให้ไม่เห็นความแตกต่าง นอกจากนี้ในผลการทดลองมีการคำนวณค่าเวลาสัมพัทธ์ โดยการคำนวณเลือกจากจำนวนประชากรต่อหน่วยประมวลผลที่ความสามารถในการให้คำตอบของการประมวลผลแบบเชิงลำดับ และแบบขนานเท่ากัน ค่าเวลาสัมพัทธ์ที่ได้ของการสังเคราะห์วงจร 0101 Detector มีค่าสูงกว่าค่าเวลาสัมพัทธ์ของการสังเคราะห์วงจร Reversible 8-Counter