# บทที่ 5

# การประมวลผลแบบขนานสำหรับปัญหาการนำร่องหุ่นยนต์ โดยเพิ่มจำนวนสิ่งแวดล้อม

ปัญหาที่ใช้ในการทดลองเพื่อวัดประสิทธิภาพการประมวลผลแบบขนานมีทั้งหมด 3 ปัญหา โดยการทดลองของทั้ง 3 ปัญหาจะใช้การประมวลผลแบบขนานแบบหน่วยหยาบ ในสองปัญหาแรกจะเป็นปัญหาที่มีการศึกษาในงานวิจัย [1] และ [2] ซึ่งเกี่ยวข้องกับการนำร่องหุ่นยนต์ด้วยโปรแกรมที่สร้างจากกำหนดการเชิงพันธุกรรม งานวิจัยทั้งสองเป็นการศึกษาหาวิธีที่จะเพิ่มความทนทานของคำตอบที่ได้จากกำหนดการเชิงพันธุกรรม โดยความแตกต่างของงานวิจัยทั้งสองอยู่ที่วิธีที่ใช้ในการเพิ่มความทนทานของคำตอบ ในปัญหาที่สามจะเป็นปัญหาที่มีการศึกษาในงานวิจัย [3] ซึ่งเกี่ยวข้องกับการสังเคราะห์เครื่องสถานะจำกัดโดยขั้นตอนวิธีเชิงพันธุกรรม

การนำเสนอประกอบด้วยลักษณะและค่ากำหนดของปัญหาจากงานวิจัยก่อน การออกแบบการประมวลผลแบบขนานและผลการทดลอง จากนั้นเป็นการวิเคราะห์ผล และการสรุป โดยการนำเสนอจะแยกเป็นสามบทตามปัญหาที่ใช้ในการทดลอง ในบทนี้นำเสนอในส่วนของการประมวลผลแบบขนานสำหรับปัญหาการนำร่องหุ่นยนต์โดยเพิ่มจำนวนสิ่งแวดล้อม ตามวิธีที่ใช้ในงานวิจัย [1] ในบทที่ 6 นำเสนอในส่วนของการประมวลผลแบบขนานสำหรับปัญหาการนำร่องหุ่นยนต์โดยเพิ่มชนิดของฟังก์ชัน ตามวิธีที่ใช้ในงานวิจัย [2] และในบทที่ 7 นำเสนอในส่วนของการประมวลผลแบบขนานสำหรับปัญหาการสังเคราะห์เครื่องสถานะจำกัด ตามวิธีที่ใช้ในงานวิจัย [3]

## 5.1 ลักษณะปัญหาการนำร่องหุ่นยนต์

ปัญหาการนำร่องหุ่นยนต์นี้เป็นการศึกษาการสร้างโปรแกรมควบคุมหุ่นยนต์แบบอัตโนมัติ เพื่อควบคุมหุ่นยนต์ให้เดินจากจุดเริ่มต้นไปยังเป้าหมายในสิ่งแวดล้อมที่มีขอบเขตจำกัด และมีสิ่งกีดขวางกระจายอยู่ โดยระบบจะถูกจำลองอยู่บนเครื่องคอมพิวเตอร์ และใช้กำหนดการเชิงพันธุกรรมในการหาคำตอบ ซึ่งอยู่ในรูปของโปรแกรมควบคุมหุ่นยนต์

### 5.1.1 ค่ากำหนดของปัญหาการนำร่องหุ่นยนต์

ค่ากำหนดที่แสดงต่อไปนี้เป็นค่าที่ใช้ในงานวิจัย [1] โดยค่ากำหนดบางค่าจะแตกต่างจากค่าที่ใช้ในงานวิจัย [2] เล็กน้อย

1. **สิ่งแวดล้อม** ลักษณะของสิ่งแวดล้อมเป็นพื้นที่สี่เหลี่ยม ขนาด $600 \times 400$ ตารางหน่วย มีขอบทั้ง 4 ด้าน ความหนาของขอบเท่ากับ 5 หน่วย ตัวอย่างของสิ่งแวดล้อมที่ใช้ในการทดลองแสดงในรูปที่ 5.1

2. **หุ่นยนต์** มีลักษณะเป็นวงกลมรัศมี 5 หน่วย หุ่นยนต์มีความสามารถในการเคลื่อนที่คือ เดินหน้า เลี้ยวซ้ายและเลี้ยวขวา นอกจากนี้หุ่นยนต์สามารถรับรู้ได้ว่าการเคลื่อนไหวนั้นทำให้เข้าใกล้หรือออกห่างจากเป้าหมาย ระยะทางการเคลื่อนที่ไปข้างหน้าแต่ละครั้งคือ 1 หน่วย โดยตำแหน่งของหุ่นยนต์

อยู่บนพิกัดจำนวนจริง และสำหรับการเลี้ยวแต่ละครั้งหุ่นยนต์จะหมุนในลักษณะอยู่กับที่เป็นมุม 22.5 องศากับทิศทางเดิม

3. สิ่งกีดขวาง สิ่งแวดล้อมที่ใช้ในการทดลองจะมีสิ่งกีดขวางกระจายอยู่ทั่วไป ครอบคลุมพื้นที่ร้อยละ 20 ของพื้นที่สิ่งแวดล้อม ขนาดของสิ่งกีดขวางโดยเฉลี่ยคือ 20 × 20 ตารางหน่วย รูปร่างของสิ่งกีดขวางจะมี 4 ลักษณะคือ วงกลม สามเหลี่ยมหน้าจั่วมุมฉาก สี่เหลี่ยมจัตุรัส และ หกเหลี่ยม สิ่งกีดขวางสามารถวางซ้อนเหลื่อมกันได้ โดยการเลือกตำแหน่งของสิ่งกีดขวางจะใช้การสุ่ม แต่จะระวังไม่ให้ตำแหน่งนั้นใกล้จุดเริ่มต้นหรือเป้าหมายมากเกินไป

4. จุดเริ่มต้นและเป้าหมาย จุดเริ่มต้นจะอยู่ที่พิกัด (100.0,100.0) และเป้าหมายอยู่ที่พิกัด (500.0,300.0) เป้าหมายมีลักษณะเป็นพื้นที่วงกลมรัศมี 5 หน่วย ถ้าจุดศูนย์กลางของหุ่นยนต์อยู่ในพื้นที่ของเป้าหมาย แสดงว่าหุ่นยนต์เคลื่อนที่เข้าถึงเป้าหมายได้สำเร็จ

### 5.1.2 โครงสร้างของโปรแกรมควบคุมหุ่นยนต์

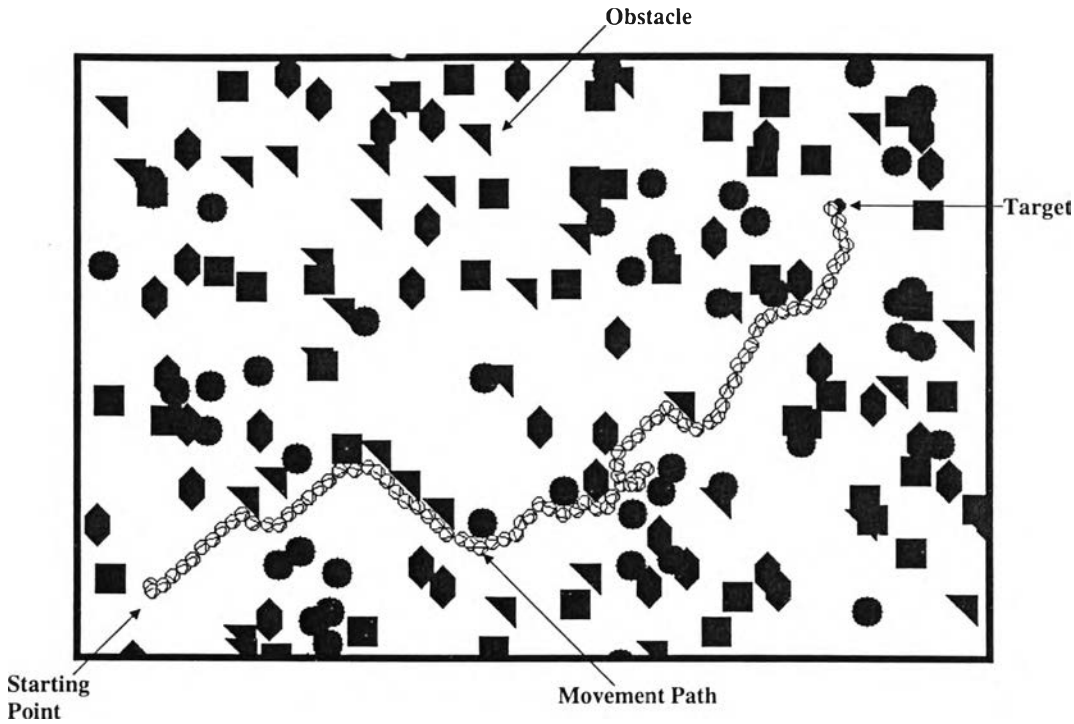
โปรแกรมควบคุมหุ่นยนต์จะมีลักษณะเป็นโครงสร้างต้นไม้ ประกอบไปด้วยฟังก์ชัน และเทอมินอล ดังนี้

1. ฟังก์ชัน ฟังก์ชันที่ใช้เป็นเงื่อนไขในการตัดสินใจประกอบด้วย 3 ฟังก์ชันดังนี้

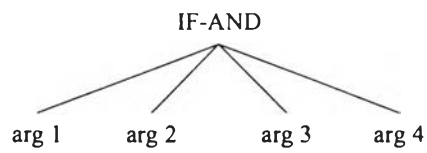
   - **ฟังก์ชัน if-and** เป็นฟังก์ชันที่มี 4 อาร์กิวเมนต์ดังรูป 5.2 สองอาร์กิวเมนต์แรกจะเป็นเงื่อนไข โดยจะดำเนินการตามเงื่อนไขทั้งสองก่อน และถ้าสองอาร์กิวเมนต์แรกให้ค่าคืนเป็นจริงทั้งคู่ จะดำเนินการตามอาร์กิวเมนต์ที่สามพร้อมทั้งส่งค่าคืนเป็นจริง มิฉะนั้นจะดำเนินการตามอาร์กิวเมนต์ที่สี่พร้อมทั้งส่งค่าคืนเป็นเท็จ
   - **ฟังก์ชัน if-or** เป็นฟังก์ชันที่มี 4 อาร์กิวเมนต์ดังรูป 5.3 สองอาร์กิวเมนต์แรกจะเป็นเงื่อนไข โดยจะดำเนินการตามเงื่อนไขทั้งสองก่อน หลังจากนั้นถ้าสองอาร์กิวเมนต์แรกมีอาร์กิวเมนต์ใดให้ค่าคืนเป็นจริง จะดำเนินการตามอาร์กิวเมนต์ที่สามพร้อมทั้งส่งค่าคืนเป็นจริง มิฉะนั้นจะดำเนินการตามอาร์กิวเมนต์ที่สี่พร้อมทั้งส่งค่าคืนเป็นเท็จ
   - **ฟังก์ชัน if-not** เป็นฟังก์ชันที่มี 3 อาร์กิวเมนต์ดังรูป 5.4 อาร์กิวเมนต์แรกจะเป็นเงื่อนไข โดยจะดำเนินการตามเงื่อนไขก่อน และหลังจากนั้นถ้าอาร์กิวเมนต์แรกให้ค่าคืนเป็นเท็จ จะดำเนินการตามอาร์กิวเมนต์ที่สองพร้อมทั้งส่งค่าคืนเป็นจริง มิฉะนั้นจะดำเนินการตามอาร์กิวเมนต์ที่สามพร้อมทั้งส่งค่าคืนเป็นเท็จ

   ความหมายของฟังก์ชันทั้งสามที่ใช้ในงานวิจัย [2] จะแตกต่างตรงค่าที่ส่งคืนจากฟังก์ชัน โดยค่าที่ส่งคืนจะไม่ใช่ค่าจากเงื่อนไข แต่จะเป็นค่าจากการดำเนินการตามสิ่งที่ถูกเลือก เช่น ในฟังก์ชัน if-and ถ้าสองอาร์กิวเมนต์แรกให้ค่าคืนเป็นจริงทั้งคู่ จะดำเนินการตามอาร์กิวเมนต์ที่สามพร้อมทั้งส่งค่าคืนเป็นค่าจากอาร์กิวเมนต์ที่สาม มิฉะนั้นจะดำเนินการตามอาร์กิวเมนต์ที่สี่พร้อมทั้งส่งค่าคืนเป็นค่าจากอาร์กิวเมนต์ที่สี่

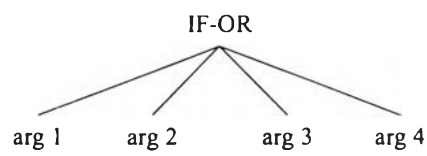
2. เทอมินอล เทอมินอลที่ใช้มีสองแบบคือ เทอมินอลที่ใช้เป็นคำสั่งควบคุมการเคลื่อนไหวของหุ่นยนต์ ได้แก่ forward, left และ right นอกจากนี้ยังมีเทอมินอลที่ให้ข้อมูลเกี่ยวกับระยะของหุ่นยนต์กับเป้าหมาย ได้แก่ isnearer



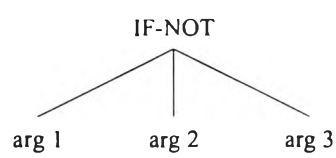

รูปที่ 5.1 ลักษณะของสิ่งแวดล้อมจำลองของปัญหาการนำร่องหุ่นยนต์



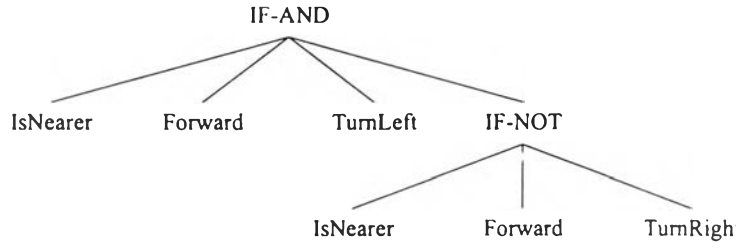

รูปที่ 5.2 ฟังก์ชัน if-and




รูปที่ 5.3 ฟังก์ชัน if-or




รูปที่ 5.4 ฟังก์ชัน if-not




รูปที่ 5.5 ตัวอย่างของโปรแกรมหุ่นยนต์

```
if (IsNearer and Forward) then
begin
    TurnLeft
end
else
begin
    if (not IsNearer) then
        Forward
    else
        TurnRight
end
```

รูปที่ 5.6 รหัสเทียมของโปรแกรมหุ่นยนต์

- เทอมินอล **forward** เป็นคำสั่งที่เคลื่อนหุ่นยนต์ไปข้างหน้า 1 หน่วยในทิศทางปัจจุบัน ถ้าหุ่นยนต์ประสบความสำเร็จในการเคลื่อนที่ไปข้างหน้า จะทำการส่งค่าคืนเป็นจริง แต่ถ้าหุ่นยนต์ไม่สามารถเคลื่อนที่ไปข้างหน้าได้ เนื่องจากมีสิ่งกีดขวางอยู่ข้างหน้า จะทำการส่งค่าคืนเป็นเท็จ
- เทอมินอล **left** ทำการหมุนหุ่นยนต์ไปในทิศทางซ้ายเป็นจำนวน 22.5 องศา ค่าที่ส่งคืนจะเป็นจริงเสมอ
- เทอมินอล **right** ทำการหมุนหุ่นยนต์ไปในทิศทางขวาเป็นจำนวน 22.5 องศา ค่าที่ส่งคืนจะเป็นจริงเสมอ
- เทอมินอล **isnearer** จะให้ข้อมูลเกี่ยวกับการเคลื่อนที่ครั้งสุดท้าย ว่าทำให้หุ่นยนต์เข้าใกล้เป้าหมายหรือไม่ ถ้าเข้าใกล้เป้าหมายมากขึ้นจะส่งค่าคืนเป็นจริง มิฉะนั้นค่าที่ส่งคืนเป็นเท็จ

ในการประมวลผลโปรแกรมจะเริ่มต้นจากรากของต้นไม้ ไปจนสิ้นสุดต้นไม้ จากนั้นการประมวลผลจะย้อนกลับไปเริ่มต้นอีกครั้งที่รากของต้นไม้ จนกว่าเงื่อนไขสิ้นสุดจะเป็นจริง ตัวอย่างของโปรแกรมหุ่นยนต์แสดงในรูป 5.5 และความหมายของโปรแกรมนี้เขียนเป็นรหัสเทียม ได้ดังรูป 5.6

### 5.1.3 การใช้กำหนดการเชิงพันธุกรรมกับปัญหาการนำร่องหุ่นยนต์

#### 5.1.3.1 ประชากรของโปรแกรมควบคุมหุ่นยนต์

ประชากรเริ่มต้นจะถูกสร้างโดยวิธีการสุ่ม โดยความสูงของโปรแกรมที่มีโครงสร้างแบบต้นไม้จะไม่เกิน 4 ชั้น จากนั้นในกระบวนการวิวัฒนาการ การสร้างประชากรรุ่นใหม่จะใช้ประชากรที่ได้รับการคัดเลือกจำนวนหนึ่ง

โดยใช้ตัวดำเนินการดังนี้คือ การทำซ้ำ จะใช้ประชากรที่ถูกเลือกทั้งหมดกำหนดให้เป็นประชากรรุ่นใหม่ โดยไม่มีการเปลี่ยนแปลง ส่วนที่เหลือของประชากรรุ่นใหม่จะสร้างโดย การเปลี่ยนไขว้ ร้อยละ 90 และการกลายพันธุ์ ร้อยละ 10 ค่ากำหนดนี้ต่างจากที่ใช้ใน [1] ซึ่งอธิบายเพิ่มเติมในหน้า 36

#### 5.1.3.2 เงื่อนไขการสิ้นสุดการประมวลผลโปรแกรม

การสิ้นสุดการประมวลผลโปรแกรมจะเกิดขึ้นเมื่อเป็นไปตามเงื่อนไขที่กำหนดไว้อย่างใดอย่างหนึ่ง ดังต่อไปนี้

- จำนวนบัพ (node) ของการประมวลผลของโปรแกรมหุ่นยนต์แต่ละโปรแกรมมีค่าเกินกว่า 10000 บัพ
- หุ่นยนต์ไปถึงเป้าหมาย โดยจุดศูนย์กลางของหุ่นยนต์อยู่ในพื้นที่เป้าหมาย

#### 5.1.3.3 การวัดประสิทธิภาพของโปรแกรมควบคุมหุ่นยนต์

การวัดประสิทธิภาพของโปรแกรมสามารถแบ่งได้เป็นสองช่วง คือ ในระหว่างกระบวนการของกำหนดการเชิงพันธุกรรมที่เป็นการวัดค่าความเหมาะสม และ การวัดประสิทธิภาพของโปรแกรมที่ได้จากกำหนดการเชิงพันธุกรรม

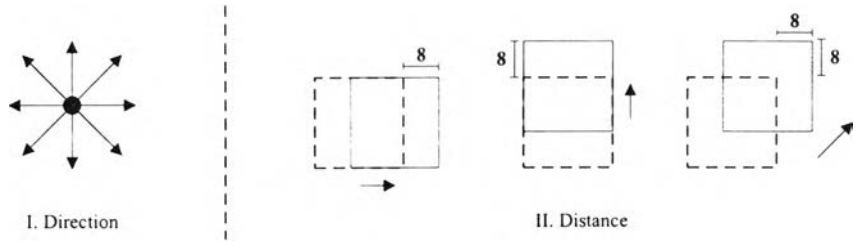
- **การวัดประสิทธิภาพระหว่างกระบวนการของกำหนดการเชิงพันธุกรรม** ในการคัดเลือกประชากรจากแต่ละรุ่นของกำหนดการเชิงพันธุกรรมเพื่อใช้สร้างประชากรรุ่นใหม่ จะใช้หลักการว่าประชากรที่มีคุณภาพ หรือประสิทธิภาพในการแก้ปัญหาดีกว่าจะมีโอกาสที่จะถูกเลือกมากกว่า การวัดประสิทธิภาพหรือค่าความเหมาะสมของประชากรสำหรับปัญหานี้จะทำโดย นำประชากรแต่ละตัวมาทำการควบคุมหุ่นยนต์ในระบบจำลอง แล้วให้คะแนนโปรแกรมแต่ละโปรแกรมเป็นตัวเลขตามความสามารถในการควบคุมหุ่นยนต์

  การให้ค่าความเหมาะสมของโปรแกรมแต่ละโปรแกรมจะขึ้นอยู่กับ จำนวนบัพที่ใช้ในการทำงาน ($m$) และระยะทางที่เข้าใกล้เป้าหมายมากที่สุด ($d$) โดยสมการที่ใช้ในการหาค่าความเหมาะสม ($f$) ของโปรแกรมเป็นดังนี้

$$f = 10000 \times d + m \tag{5.1}$$

  จากสมการจะเห็นได้ว่าจะให้ความสำคัญกับระยะทางที่เข้าใกล้เป้าหมายมากกว่าจำนวนบัพที่ใช้ และโปรแกรมที่มีค่าความเหมาะสมน้อยกว่าจะมีประสิทธิภาพที่ดีกว่า

- **การวัดประสิทธิภาพหลังกระบวนการของกำหนดการเชิงพันธุกรรม** เพื่อทดสอบคุณภาพของคำตอบที่ได้จากกำหนดการเชิงพันธุกรรมว่ามีความสามารถในการแก้ไขปัญหามากน้อยเพียงใด สามารถทำได้โดยนำคำตอบมาทำการวัดประสิทธิภาพ โดยนิยามค่าความสามารถของโปรแกรมเป็น ความทนทาน (robustness) ซึ่งจะแสดงถึงประสิทธิภาพการทำงานของโปรแกรม ในสิ่งแวดล้อมที่ต่างจากที่เรียนรู้ในกำหนดการเชิงพันธุกรรม การวัดความทนทานจะทำโดยการนำโปรแกรมที่ดีที่สุดโดยเลือกตามค่าความเหมาะสม มาทดลองในสิ่งแวดล้อมที่มีการใส่ การรบกวน (disturbance) และวัดจำนวนครั้งที่โปรแกรมประสบความสำเร็จในการควบคุมหุ่นยนต์ไปสู่เป้าหมาย




รูปที่ 5.7 ทิศทางและระยะของการเปลี่ยนตำแหน่งสิ่งกีดขวาง

การรบกวนที่ใส่เข้าไปจะกำหนดเป็นเปอร์เซ็นต์ของการรบกวน $(D)$ โดยสมการที่ใช้ในการวัดเปอร์เซ็นต์ของการรบกวนเป็นดังนี้

$$D = \frac{N_m}{N_o} \times 100 \tag{5.2}$$

$N_m$ เป็นจำนวนสิ่งกีดขวางที่ถูกเคลื่อนย้าย

$N_o$ เป็นจำนวนสิ่งกีดขวางทั้งหมดในสิ่งแวดล้อม

การใส่การรบกวนเข้าไปในสิ่งแวดล้อมจะทำโดยเลือกสิ่งกีดขวางด้วยการสุ่ม เป็นจำนวนตามระดับของการรบกวน และทำการเปลี่ยนตำแหน่งของสิ่งกีดขวางแต่ละชิ้นในแนวแกนใดแกนหนึ่ง หรือทั้งสองแกนเป็นระยะทาง 8 หน่วย หรือประมาณร้อยละ 40 ของเส้นผ่าศูนย์กลางของสิ่งกีดขวาง ทิศทางการเปลี่ยนตำแหน่งนี้มี 8 วิธีดังรูป 5.7 โดยตำแหน่งใหม่ของสิ่งกีดขวางต้องไม่เข้าใกล้จุดเริ่มต้นหรือเป้าหมายจนเกินไป

สำหรับสมการที่ใช้ในการวัดความทนทาน $(R)$ เป็นดังนี้

$$R = \frac{N_s}{N_t} \times 100 \tag{5.3}$$

$N_s$ เป็นจำนวนสิ่งแวดล้อมที่โปรแกรมประสบความสำเร็จในการเข้าหาเป้าหมาย

$N_t$ เป็นจำนวนสิ่งแวดล้อมทั้งหมด

## 5.2 การปรับปรุงความทนทานโดยเพิ่มจำนวนสิ่งแวดล้อม

จากงานวิจัย [29] แสดงให้เห็นว่าโปรแกรมควบคุมหุ่นยนต์ที่ได้จากกำหนดการเชิงพันธุกรรมยังขาดความทนทาน เนื่องจากโปรแกรมที่ได้จากการเรียนรู้ในโลกจำลอง เมื่อนำไปใช้กับหุ่นยนต์จริง ในการทำงานบางครั้งโปรแกรมไม่ประสบความสำเร็จ เนื่องมาจากความไม่แน่นอน และความคลาดเคลื่อนของโลกจริง

งานวิจัย [1] ได้เสนอวิธีปรับปรุงความทนทาน โดยการเพิ่มจำนวนสิ่งแวดล้อมที่ใช้สำหรับการเรียนรู้ในกำหนดการเชิงพันธุกรรม ผลลัพท์ที่ได้แสดงให้เห็นว่าวิธีการนี้สามารถเพิ่มความทนทานของโปรแกรมควบคุมหุ่นยนต์ได้ การเพิ่มจำนวนสิ่งแวดล้อมสำหรับการเรียนรู้ทำได้โดย ค่าความเหมาะสมของประชากรแต่

ละตัวเป็นผลรวมของค่าความเหมาะสมในแต่ละสิ่งแวดล้อมตามจำนวนสิ่งแวดล้อมที่ใช้ สิ่งแวดล้อมที่เพิ่มขึ้นมาเป็นสิ่งแวดล้อมที่ได้จากการใส่การรบกวนลงไปในสิ่งแวดล้อมดั้งเดิม ซึ่งการใส่การรบกวนนี้จะทำอย่างระมัดระวัง เพื่อให้เหลือเส้นทางที่หุ่นยนต์สามารถเคลื่อนที่ไปสู่เป้าหมายได้จริง ดังนั้นอาจกล่าวได้ว่าวิธีการนี้เป็นการเพิ่มความทนทานโดยใช้การเติมการรบกวนลงไปในกระบวนการเรียนรู้ของหุ่นยนต์ โดยสมการที่ใช้ในการหาค่าความเหมาะสม ($f$) จะเปลี่ยนเป็นดังนี้

$$f = \sum_{i=1}^{n} (10000 \times d_i + m_i) \tag{5.4}$$

$n$ เป็นจำนวนสิ่งแวดล้อมที่ใช้ในกระบวนการเรียนรู้
$d_i$ เป็นระยะทางที่เข้าใกล้เป้าหมายมากที่สุดในสิ่งแวดล้อมที่ $i$
$m_i$ จำนวนบัพที่ใช้ในการทำงานในสิ่งแวดล้อมที่ $i$

## 5.3 การออกแบบการประมวลผลแบบขนานและผลการทดลอง

ในส่วนนี้เป็นการนำเสนอการออกแบบ และผลการทดลองของการใช้การประมวลผลแบบขนานกับปัญหาการนำร่องหุ่นยนต์โดยเพิ่มจำนวนสิ่งแวดล้อม เพื่อให้การเปรียบเทียบประสิทธิภาพที่ได้จากการประมวลผลแบบขนาน และการประมวลผลแบบเชิงลำดับเป็นไปอย่างเหมาะสม คุณภาพคำตอบที่ได้จากการประมวลผลแบบขนานต้องเหมือนกับคุณภาพคำตอบที่ได้จากการประมวลผลแบบเชิงลำดับ ในปัญหาการนำร่องหุ่นยนต์คุณภาพของคำตอบวัดได้จากค่าความทนทานของคำตอบ ดังนั้นผลการทดลองที่จะนำเสนอต่อไปนี้และในบทที่ 6 จะมีการเปรียบเทียบความทนทานในทุกการทดลอง ก่อนที่จะทำการเปรียบเทียบประสิทธิภาพในเรื่องของเวลาในการหาคำตอบ

การทดลองแบ่งออกเป็น 2 ส่วน ในส่วนแรกเป็นการทดลองแบบพื้นฐานคือ ทุกหน่วยประมวลผลใช้ค่ากำหนดและสิ่งแวดล้อมสำหรับกระบวนการเรียนรู้เหมือนกัน และการอพยพจะกระทำในแบบประสานเวลา ในส่วนที่สองเป็นการหาแนวทางในการเพิ่มประสิทธิภาพของการทดลองในส่วนแรก โดยแนวทางในการเพิ่มประสิทธิภาพมี 2 ข้อดังนี้ ข้อแรกเป็นการใช้สิ่งแวดล้อมสำหรับกระบวนการเรียนรู้ที่แตกต่างกันในแต่ละหน่วยประมวลผล เนื่องจากการใช้สิ่งแวดล้อมที่แตกต่างกันทำให้ความทนทานของคำตอบเพิ่มขึ้น ในทางกลับกันสามารถลดจำนวนสิ่งแวดล้อมในแต่ละหน่วยประมวลผลลง เพื่อให้ความทนทานของคำตอบจากการประมวลผลแบบขนานมีค่าเท่ากับหรือมากกว่าการประมวลผลแบบเชิงลำดับ ข้อสองเป็นการทดลองการอพยพแบบไม่ประสานเวลาเปรียบเทียบกับการอพยพแบบประสานเวลา เนื่องจากการทดลองในส่วนแรกพบว่าเวลาในการให้ทุกหน่วยประมวลผลพร้อมสำหรับการอพยพ เป็นเวลาส่วนใหญ่ของเวลาที่เสียไปในการสื่อสารระหว่างหน่วยประมวลผล ดังนั้นการอพยพแบบไม่ประสานเวลาน่าจะช่วยในการประมวลผลแบบขนานมีประสิทธิภาพสูงขึ้น

เครื่องคอมพิวเตอร์ที่ใช้ในการทดลองมีลักษณะดังนี้ เครื่องที่ทำหน้าที่ประมวลผลมีสองรุ่น โดยหน่วยประมวลผลจะแตกต่างกัน แบบแรกเป็นเครื่องที่มีหน่วยประมวลผล Pentium 133 MHz และแบบที่สองเป็นเครื่องที่มีหน่วยประมวลผล Pentium II 350 MHz หน่วยความจำของเครื่องทั้งสองรุ่นเท่ากัน คือ 32 Mbyte ทุกเครื่องใช้ระบบปฏิบัติการ RedHat Linux 5.1 และเชื่อมต่อด้วยเครือข่ายอีเทอร์เน็ต (Ethernet) ที่ใช้ฮับ (hub) ความเร็ว 10 Mbps นอกจากนี้มีเครื่องที่เป็นทางเข้าเพื่อทำหน้าที่แยกเครือข่ายการประมวล

I19239300

ผลจากเครือข่ายภายนอก โดยใช้เครื่องที่มีหน่วยประมวลผล 486DX 66 MHz หน่วยความจำ 16 Mbyte ระบบปฏิบัติการ RedHat Linux 5.1 โดยเชื่อมต่อเครือข่ายสองทางคือ เครือข่ายการประมวลผล และเครือข่ายภายนอกด้วยความเร็ว 10 Mbps ทั้งสองทาง ลักษณะการติดตั้งเครื่องคอมพิวเตอร์แสดงในรูป 5.8

เนื่องจากบางการทดลองใน 3 ปัญหา การเปรียบเทียบเวลาในการหาคำตอบระหว่างการประมวลผลแบบเชิงลำดับและแบบขนานกระทำที่คุณภาพของคำตอบระดับเดียวกัน โดยที่ขนาดของงานในแง่จำนวนฟังก์ชันที่เรียกใช้โดยขั้นตอนวิธีเชิงพันธุกรรม และกำหนดการเชิงพันธุกรรมของการทำงานแบบเชิงลำดับและแบบขนานมีจำนวนไม่เท่ากัน ทำให้ไม่ตรงกับนิยามที่ถูกต้องของค่าสปีดอัพ (speedup) ดังนั้นการทดลองจึงนิยามการเปรียบเทียบเวลาให้เป็นค่าเวลาสัมพัทธ์ (relative time) ซึ่งคำนวณได้จากเวลาในการหาคำตอบของการประมวลผลแบบเชิงลำดับหารด้วยเวลาของการประมวลผลแบบขนาน เมื่อคุณภาพของคำตอบจากการทำงานทั้งสองแบบอยู่ในระดับเดียวกัน การนำเสนอเป็นค่าเวลาสัมพัทธ์นี้จะใช้กับทุกการทดลอง

ในส่วนแรกของการทดลองของปัญหาการนำร่องหุ่นยนต์โดยเพิ่มจำนวนสิ่งแวดล้อม ใช้เครื่องที่มีหน่วยประมวลผล Pentium 133 MHz เป็นเครื่องที่ทำหน้าที่ประมวลผล การทดลองที่เหลือรวมทั้งการทดลองกับปัญหาในบทที่ 6 และ 7 ใช้เครื่องที่มีหน่วยประมวลผล Pentium II 350 MHz เป็นเครื่องที่ทำหน้าที่ประมวลผล

### 5.3.1 การทดลองโดยแบ่งประชากรและใช้สิ่งแวดล้อมเหมือนกัน

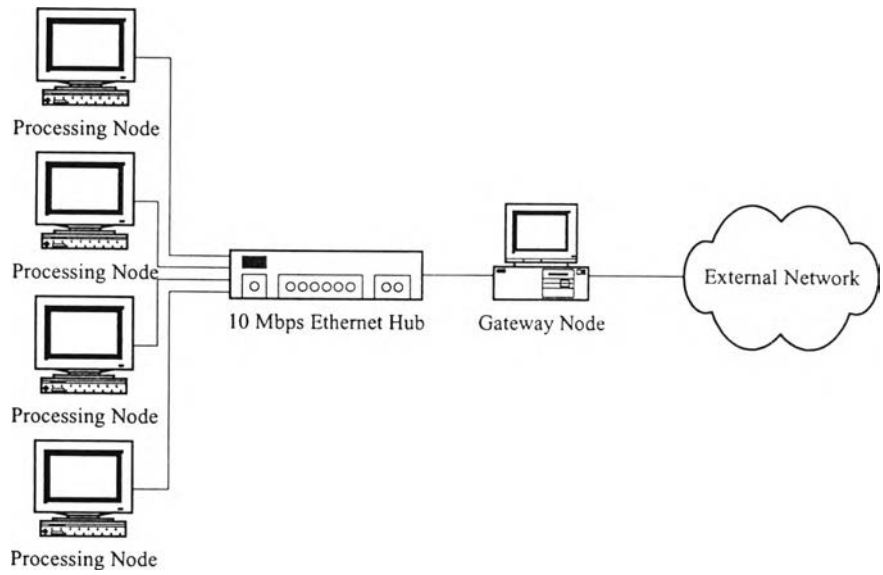
#### 5.3.1.1 การออกแบบการทดลอง

โปรแกรมที่ใช้ในการทดลองเป็นการนำโปรแกรมจากงานวิจัย [1] มาดัดแปลงและเพิ่มเติม การอพยพจะกระทำในทุกรุ่นของการวิวัฒนาการ โดยเป็นแบบประสานเวลา การส่งประชากรในการอพยพจะทำในลักษณะวงแหวนดังรูป 5.9 เนื่องจากการรับส่งข้อมูลกระทำในแบบประสานเวลา การใช้ฟังก์ชันรับและส่งข้อมูลต้องมีการหยุดรอจนกว่าการรับส่งจะเสร็จเรียบร้อย ทำให้ต้องมีการกำหนดให้ชัดเจนว่าหน่วยประมวลผลใดทำการรับข้อมูล และหน่วยประมวลผลใดทำการส่งข้อมูล ในการทดลองจึงให้การแลกเปลี่ยนประชากรเป็นสองจังหวะ จังหวะแรกจะให้หน่วยประมวลผลที่เป็นเลขคู่ทำการส่งประชากรให้หน่วยประมวลผลที่เป็นเลขคี่ จากนั้นในจังหวะที่สองหน่วยประมวลที่เป็นเลขคี่จะทำการส่งประชากรให้หน่วยประมวลผลที่เป็นเลขคู่ในทิศทางเดียวกัน สามารถเขียนเป็นรหัสเทียมได้ดังรูป 5.10

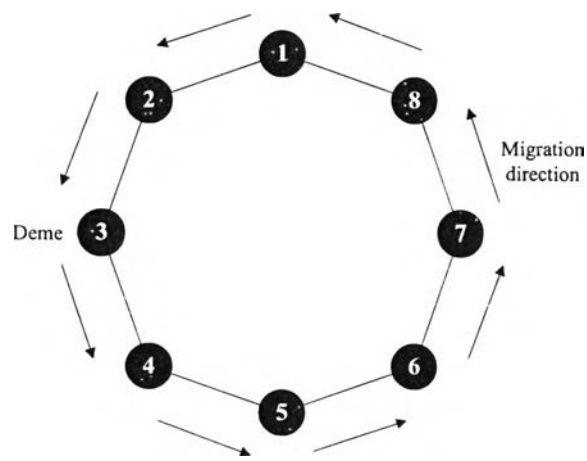
จากรหัสเทียม ทุกหน่วยประมวลผลหยุดรอด้วยฟังก์ชันการกีดขวางในตอนเริ่มต้นการอพยพ ซึ่งไม่จำเป็นต้องมีก็ได้เนื่องจากการรับส่งทำงานแบบประสานเวลา แต่วัตถุประสงค์ในการเพิ่มฟังก์ชันการกีดขวางเพื่อทำให้ทราบเวลาที่หน่วยประมวลผลใช้ในการรอ

ในงานวิจัย [1] กระบวนการวิวัฒนาการจะเกิดขึ้นเป็นจำนวน 200 รุ่น โดยในแต่ละรุ่นใช้ขนาดประชากรเท่ากับ 960 ตัว หลังจากเรียงลำดับโปรแกรมหุ่นยนต์ตามค่าความเหมาะสม จะทำการเลือกโปรแกรมไว้ 30 โปรแกรม โดยเป็นโปรแกรมที่ดีที่สุดหนึ่งโปรแกรม และอีก 29 โปรแกรมเลือกโดยการสุ่มซึ่งโอกาสที่แต่ละโปรแกรมจะถูกเลือกแปรผันโดยตรงกับค่าความเหมาะสม จากนั้นการสร้างประชากรรุ่นใหม่จะให้โปรแกรมทั้ง 30 โปรแกรมที่เลือกไว้กำหนดให้เป็นประชากรรุ่นใหม่ ประชากรที่เหลือจะสร้างด้วยการเปลี่ยนไขว้เป็นจำนวน 900 ตัว โดยการจับคู่ทุกแบบจากโปรแกรมที่เลือกไว้ 30 โปรแกรม[1] ประชากรที่เหลืออีก 30 ตัวจะสร้างด้วยการกลายพันธุ์ โดยการนำโปรแกรมที่เลือกไว้ 30 โปรแกรมมาทำการกลายพันธุ์

[1] รวมทั้งการเปลี่ยนไขว้กับประชากรตัวเดียวกัน




รูปที่ 5.8 การติดตั้งเครื่องคอมพิวเตอร์ที่ใช้ในการทดลอง




รูปที่ 5.9 การเชื่อมต่อในลักษณะวงแหวน

```
procedure Migration
begin
    barrier wait all nodes ready
    if (my_process_id =odd number ) then
    begin
        Immigrate from the previous node
        Emigrate to the next node
    end
    else
    begin
        Emigrate to the next node
        Immigrate from the previous node
    end
end
```

รูปที่ 5.10 รหัสเทียมของการแลกเปลี่ยนประชากรแบบวงแหวน

จะเห็นได้ว่าการสร้างประชากรตามงานวิจัยเดิม [1] ไม่เหมาะกับการประมวลผลแบบขนาน เนื่องจากในการประมวลผลแบบขนานจะใช้การแบ่งประชากรโดยให้ประชากรรวมของทุกหน่วยประมวลผลมีค่าคงที่ การสร้างประชากรตามงานวิจัยเดิมจะกำหนดจำนวนประชากรที่สร้างจากตัวดำเนินการต่างๆ เป็นค่าคงที่ เมื่อทำการแบ่งประชากรจะไม่สามารถรักษาจำนวนประชากรที่สร้างจากตัวดำเนินการต่างๆ ได้ เช่น เมื่อใช้ 2 หน่วยประมวลผล แต่ละหน่วยประมวลผลมีประชากรเท่ากับ 480 ตัว ถ้าใช้การเลือกประชากรในอัตราเท่าเดิม แต่ละหน่วยประมวลผลต้องเลือกประชากรเป็นจำนวน 15 ตัว ดังนั้นการสร้างประชากรโดยใช้การเปลี่ยนไขว้ในแต่ละหน่วยประมวลผลจะได้ประชากรเป็นจำนวน 225 ตัว และโดยใช้การกลายพันธุ์จะได้ประชากรเป็นจำนวน 15 ตัว รวมแล้วจะได้ประชากรเพียง 255 ตัว

ดังนั้นในการทดลองจึงทำการเปลี่ยนลักษณะการสร้างประชากรรุ่นใหม่ โดยกำหนดจำนวนให้เป็นอัตราส่วนกับขนาดของประชากร ประชากรจะสร้างโดยการเปลี่ยนไขว้ด้วยความน่าจะเป็นเท่ากับ 0.9 และสร้างโดยการกลายพันธุ์ด้วยความน่าจะเป็นเท่ากับ 0.1 ขนาดของประชากรที่ทำการเลือกเพื่อใช้สร้างประชากรรุ่นใหม่มีจำนวนร้อยละ 5 ของจำนวนประชากร แต่การเลือกประชากรเพื่อใช้สร้างประชากรรุ่นใหม่เป็นจำนวนที่น้อยนี้อาจทำให้เกิดปัญหา เนื่องจากเมื่อทำการทดลองการประมวลผลแบบขนาน ประชากรจะถูกแบ่งทำให้ขนาดประชากรในแต่ละหน่วยประมวลผลจะมีขนาดเล็กลง ประชากรที่ถูกเลือกมีขนาดเล็กลงตามขนาดของประชากรต่อหน่วยประมวลผล ซึ่งอาจจะไม่เพียงพอกับการทำงานของกำหนดการเชิงพันธุกรรม ดังนั้นในการทดลองจึงเพิ่มขนาดประชากรให้เป็น 6000 ตัว

การสร้างประชากรรุ่นใหม่ในแต่ละรุ่นจะได้จากประชากรที่ถูกเลือกส่วนหนึ่ง ถ้าเป็นการประมวลผลแบบขนานจะมีประชากรอีก 10 ตัวที่ได้จากการอพยพซึ่งกำหนดให้เป็นประชากรรุ่นใหม่ ส่วนที่เหลือสร้างจากการเปลี่ยนไขว้และการกลายพันธุ์จากประชากรที่ถูกเลือก โดยใช้ความน่าจะเป็นดังที่ได้กล่าวมา จำนวนการสร้างประชากรรุ่นใหม่ของการทดลองแสดงในตาราง 5.1

การทำงานจะสิ้นสุดเมื่อผ่านการวิวัฒนาการเป็นจำนวน 200 รุ่นเหมือนกับงานวิจัย [1] จำนวนสิ่งแวดล้อมที่ใช้ในกระบวนการเรียนรู้มีค่าเท่ากับ 5 โดยแต่ละหน่วยประมวลผลจะใช้สิ่งแวดล้อมเหมือนกัน

ตารางที่ 5.1 จำนวนการสร้างประชากรรุ่นใหม่ของการทดลองโดยใช้สิ่งแวดล้อมเหมือนกัน

| | จำนวนหน่วยประมวลผล | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 4 | 6 | 8 | 10 |
| ขนาดประชากร (ต่อหน่วยประมวลผล) | 6000 | 3000 | 1500 | 1000 | 750 | 600 |
| ประชากรที่ถูกเลือก (5%) | 300 | 150 | 75 | 50 | 38 | 30 |
| การอพยพ | ไม่มี | 10 | 10 | 10 | 10 | 10 |
| การเปลี่ยนไขว้และการกลายพันธุ์ | 5700 | 2840 | 1415 | 940 | 702 | 560 |

#### 5.3.1.2 ผลการทดลอง

ผลการทดลองจะเฉลี่ยจากการทดลองซ้ำ 20 ชุด ซึ่งเป็นจำนวนเดียวกับที่ใช้ในงานวิจัย [30] ซึ่งเป็นการศึกษาเพิ่มเติมจากงานวิจัย [1] โดยใช้ปัญหาเดียวกัน

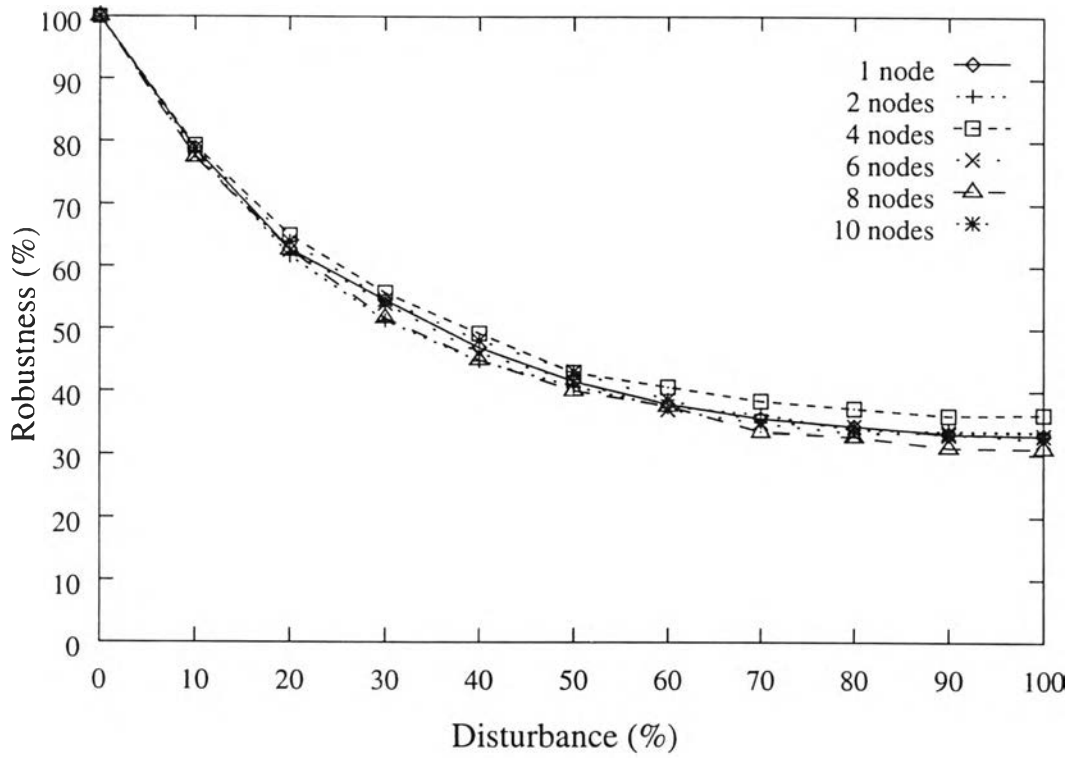
กราฟความทนทานแสดงในรูป 5.11 เฉลี่ยจากโปรแกรมที่ดีที่สุดของแต่ละชุดในการทดลองซ้ำ แกนตั้งของกราฟเป็นความทนทานและแกนนอนเป็นค่าเปอร์เซ็นต์การรบกวนตั้งแต่ 0% ถึง 100% แต่ละค่าเปอร์เซ็นต์การรบกวน จะใช้สิ่งแวดล้อมจำนวน 1000 ชุดในการวัดความทนทาน จากกราฟจะเห็นว่าการประมวลผลแบบขนาน และการประมวลผลแบบเชิงลำดับให้ความทนทานได้ใกล้เคียงกัน อาจกล่าวได้ว่าคุณภาพคำตอบนั้นเหมือนกัน

ค่าเวลาสัมพัทธ์ของการประมวลผลแบบขนานแสดงในรูป 5.12 แกนนอนของกราฟเป็นจำนวนหน่วยประมวลผลที่ใช้ และแกนตั้งเป็นค่าเวลาสัมพัทธ์ จากกราฟจะเห็นได้ว่าค่าเวลาสัมพัทธ์ที่ได้ใกล้เคียงเชิงเส้นโดยมีการลดลงเล็กน้อยเมื่อหน่วยประมวลผลเพิ่มขึ้น

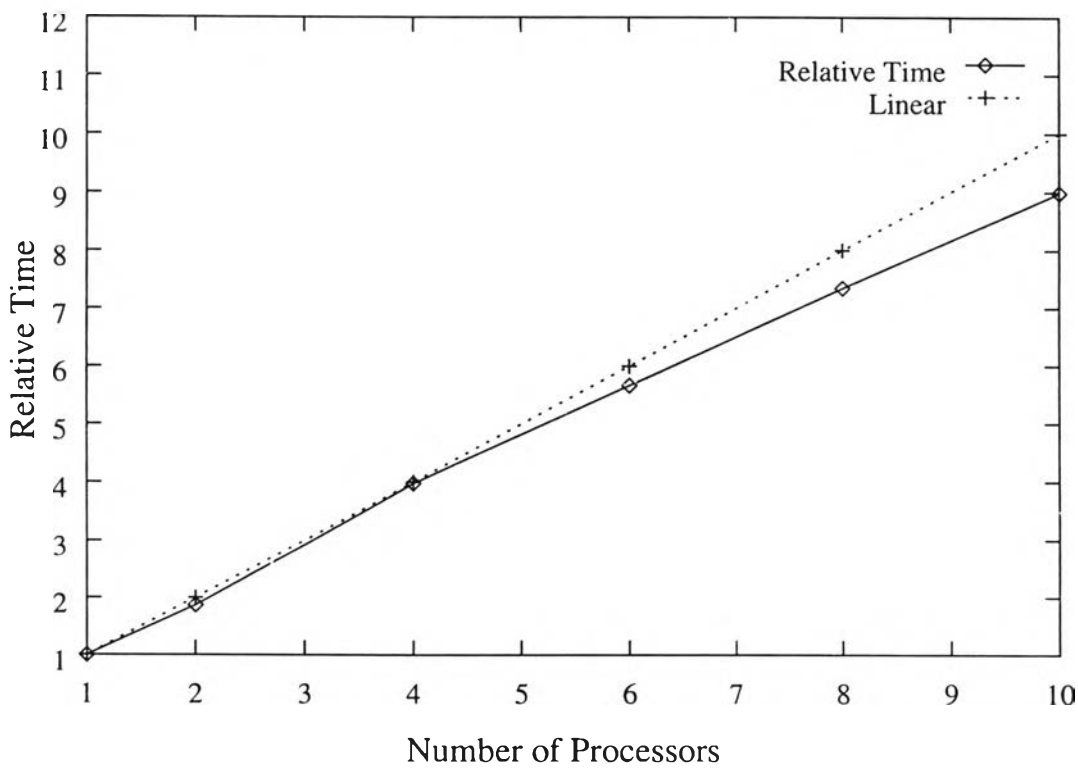
รูป 5.13 แสดงเวลาที่เสียไปในการสื่อสาร เวลาที่ใช้ในการสื่อสารจะประกอบด้วยเวลาที่ใช้ในการรับส่งข้อมูล และเวลาที่เสียไปกับฟังก์ชันการกีดขวาง (barrier) ซึ่งใช้ในการรอให้ทุกหน่วยประมวลผลพร้อมสำหรับการรับส่งข้อมูล จากกราฟแสดงให้เห็นว่าเวลาที่ใช้ในการรับส่งข้อมูลเพิ่มขึ้นตามจำนวนหน่วยประมวลผลที่ใช้ และเวลาส่วนใหญ่ของการสื่อสารจะเป็นเวลาในการรอของฟังก์ชันการกีดขวาง เวลาในการรอนี้มีค่าไม่แน่นอน เนื่องมาจากความแตกต่างของประชากรในแต่ละหน่วยประมวลผล ทำให้เวลาที่ใช้ในการดำเนินการตามโปรแกรมในขั้นตอนการหาค่าความเหมาะสมไม่เท่ากัน โดยประชากรที่มีประสิทธิภาพดีกว่าสามารถควบคุมหุ่นยนต์ให้ถึงเป้าหมายเร็วกว่าประชากรที่ไม่มีประสิทธิภาพ

การเปรียบเทียบเปอร์เซ็นต์ของเวลาที่ใช้ในการสื่อสารและการคำนวณแสดงในรูป 5.14 เปอร์เซ็นต์ของเวลาที่ใช้ในการสื่อสารจะค่อยๆ เพิ่มตามจำนวนหน่วยประมวลผลที่เพิ่มขึ้น ถึงแม้ว่าจากรูป 5.13 การเพิ่มจำนวนหน่วยประมวลบางครั้งไม่ได้เพิ่มเวลาที่ใช้ในการสื่อสาร เช่น จาก 8 ไปเป็น 10 หน่วยประมวลผล แต่เวลาที่ใช้ในการสื่อสารเมื่อคิดเป็นเปอร์เซ็นต์เทียบกับเวลาทั้งหมดก็ยังคงเพิ่มขึ้น เนื่องมาจากการลดลงของเวลาที่ใช้ในส่วนของการคำนวณ เพราะขนาดของประชากรเล็กลงเมื่อเพิ่มหน่วยประมวลผล

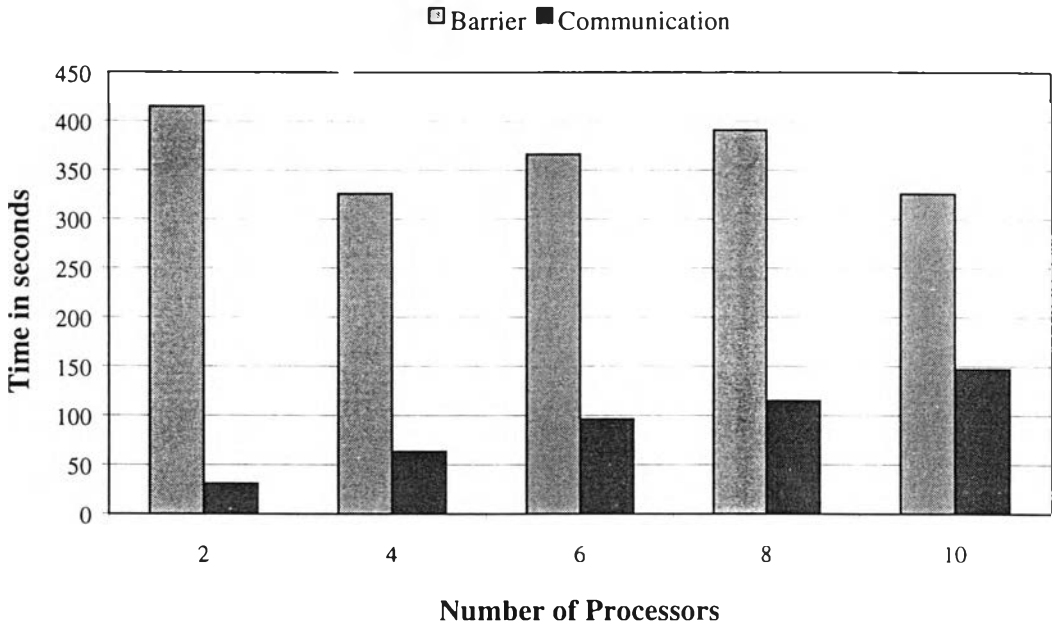
จากหนังสือ [31] ได้อธิบายถึงอัตราส่วนระหว่างเวลาในการคำนวณ ($R$) และเวลาในการสื่อสาร ($C$) ว่าเกี่ยวข้องกับประสิทธิภาพการประมวลผลแบบขนาน โดยการประมวลผลแบบขนาน ที่ค่า $R/C$ มีค่ามากจะมีประสิทธิภาพสูงกว่าการประมวลผลแบบขนานที่ $R/C$ มีค่าน้อย




รูปที่ 5.11 ความทนทานของการทดลองโดยใช้สิ่งแวดล้อมเหมือนกัน




รูปที่ 5.12 ค่าเวลาสัมพัทธ์ของการทดลองโดยใช้สิ่งแวดล้อมเหมือนกัน




รูปที่ 5.13 เวลาที่เสียไปในการสื่อสารของการทดลองโดยใช้สิ่งแวดล้อมเหมือนกัน

จากผลการทดลองในรูป 5.14 จะเห็นว่าเมื่อเพิ่มจำนวนหน่วยประมวลผล ค่า $R/C$ มีค่าลดลง ซึ่งจะส่งผลให้ประสิทธิภาพการประมวลผลลดลงเมื่อมีการเพิ่มจำนวนหน่วยประมวลผล สำหรับการวิเคราะห์เพิ่มเติมจะนำเสนอในส่วนการวิเคราะห์ผลในหัวข้อ 5.4.1

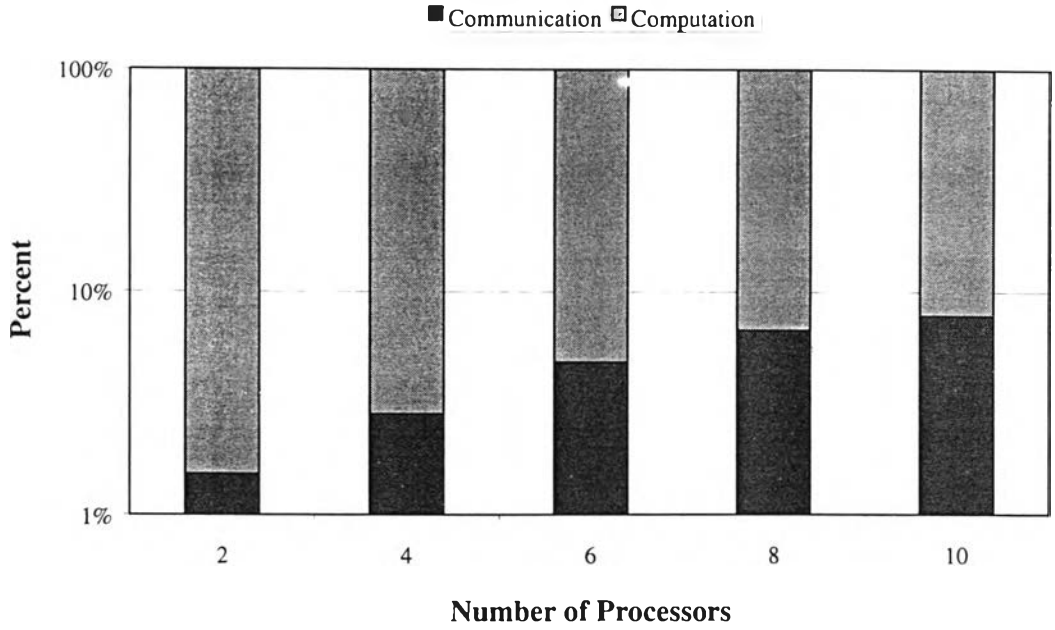
### 5.3.2 การทดลองโดยแบ่งประชากรและใช้สิ่งแวดล้อมต่างกัน

ในการทดลองชุดที่แล้วในหัวข้อ 5.3.1 จำนวนสิ่งแวดล้อมที่ใช้ของการประมวลผลแบบเชิงลำดับจะเท่ากับที่ใช้ในการประมวลผลแบบขนาน นอกจากนี้สิ่งแวดล้อมที่ใช้ในการประมวลผลแบบขนานจะเหมือนกันทุกหน่วยประมวลผล และเหมือนกับที่ใช้ในการประมวลผลแบบเชิงลำดับ ในการทดลองชุดนี้จะใช้วิธีที่แตกต่างจากการทดลองชุดที่แล้ว โดยให้หน่วยประมวลผลแต่ละหน่วยใช้สิ่งแวดล้อมที่ต่างกัน จากการทดลองพบว่าความทนทานของคำตอบที่ได้จะสูงขึ้น ในทางกลับกันถ้าเราต้องการให้คำตอบของการประมวลผลแบบขนานมีความทนทานเท่ากับการประมวลผลแบบเชิงลำดับ เราสามารถใช้จำนวนสิ่งแวดล้อมที่น้อยกว่าเพื่อให้ได้ความทนทานของคำตอบเท่ากัน ผลที่ได้ก็คือเวลาที่ใช้ในการประมวลผลลดลง

เนื่องจากการทดลองชุดที่แล้ว เวลาในการรอให้แต่ละหน่วยประมวลผลพร้อมในการรับส่งข้อมูลเป็นเวลาส่วนใหญ่ของการสื่อสาร ดังนั้นในการทดลองชุดนี้จึงได้ทำการทดลองเป็นสองแบบคือ แบบประสานเวลาและแบบไม่ประสานเวลา โดยหวังว่าการสื่อสารแบบไม่ประสานเวลาจะช่วยลดเวลาในการรอให้แต่ละหน่วยประมวลผลพร้อมในการรับส่งข้อมูล

#### 5.3.2.1 การออกแบบการทดลอง

โปรแกรมที่ใช้ในการทดลองดัดแปลงและเพิ่มเติมจากการทดลองในหัวข้อ 5.3.1 ในโปรแกรมเติมการอพยพจะกระทำทุกรุ่นของการวิวัฒนาการ โดยใช้ประชากรจำนวนน้อยในการอพยพแต่ละครั้ง ในการทดลองนี้สิ่งแวด




รูปที่ 5.14 เปอร์เซ็นต์ของเวลาในการสื่อสารและการคำนวณของการทดลองโดยใช้สิ่งแวดล้อมเหมือนกัน

ล้อมกระจายอยู่ในหน่วยประมวลผลต่างๆ โดยจำนวนสิ่งแวดล้อมในแต่ละหน่วยประมวลผลมีจำนวนน้อยกว่าที่ใช้ในการประมวลผลเชิงลำดับ ดังนั้นการออกแบบจึงพยายามให้ประชากรในแต่ละหน่วยประมวลผลมีการแลกเปลี่ยนได้มากที่สุด เพื่อให้ประชากรในแต่ละหน่วยประมวลผลได้มีโอกาสในการเรียนรู้ในสิ่งแวดล้อมที่กระจายอยู่ในหน่วยประมวลผลอื่นๆ

ในการอพยพ ประชากรที่ถูกเลือกเพื่อสร้างประชากรรุ่นใหม่ทั้งหมดจะถูกส่งไปยังหน่วยประมวลผลอื่นๆ เมื่อใช้ประชากรที่มีจำนวนมากในการส่ง จำนวนการอพยพจึงลดลงเพื่อให้เวลาที่ใช้ในการสื่อสารไม่มากจนเกินไป การอพยพจะกระทำเป็นช่วงขึ้นกับจำนวนหน่วยประมวลผล ดังตาราง 5.2 การอพยพมีจำนวนเพียงพอที่ประชากรแต่ละกลุ่มจะสามารถไปถึงทุกหน่วยประมวลผล โดยถ้าเพิ่มจำนวนหน่วยประมวลผล ความถี่ของการอพยพก็จะเพิ่มขึ้น การส่งประชากรในการอพยพจะทำในลักษณะที่แต่ละหน่วยส่งประชากรไปยังหน่วยประมวลผลอื่นๆ ทุกหน่วย และรับประชากรจากหน่วยประมวลผลอื่นๆ ทุกหน่วย หลังจากนั้นลบประชากรเก่าในหน่วยประมวลผล ดังรูป 5.15

พารามิเตอร์ในการทดลองยังคงใช้เหมือนการทดลองในหัวข้อ 5.3.1 ยกเว้น จำนวนสิ่งแวดล้อมที่ใช้ในกระบวนการเรียนรู้จะเทียบกับขั้นตอนวิธีเชิงลำดับที่ใช้สิ่งแวดล้อมจำนวน 8 ชุดในการทดลอง ซึ่งมากว่าในการทดลองในหัวข้อ 5.3.1 ซึ่งสาเหตุจะอธิบายต่อไป นอกจากนี้สิ่งที่แตกต่างจากการทดลองในหัวข้อ 5.3.1 คือ จำนวนประชากรที่อพยพจะใช้ประชากรทั้งหมดที่ถูกเลือก[2] หลังจากการแลกเปลี่ยนประชากรในการอพยพ ประชากรส่วนที่เหลือสร้างขึ้นโดยการการเปลี่ยนไขว้ และการกลายพันธุ์ด้วยความน่าจะเป็นเท่ากับที่ใช้ในการสร้างประชากรในรุ่นที่ไม่มีการอพยพ

การใช้สิ่งแวดล้อมที่ต่างกันทำให้สามารถลดจำนวนสิ่งแวดล้อมที่ใช้ในการเรียนรู้ของหุ่นยนต์ ในตาราง 5.2 แสดงจำนวนสิ่งแวดล้อมที่น้อยที่สุดที่ทำให้ความทนทานของคำตอบจากการประมวลผลแบบขนานมีค่า

[2]ร้อยละ 5 ของประชากรทั้งหมด

ตารางที่ 5.2 จำนวนสิ่งแวดล้อมและช่วงของการอพยพ

| | จำนวนหน่วยประมวลผล | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 4 | 6 | 10 |
| ขนาดประชากร (ต่อหน่วยประมวลผล) | 6000 | 3000 | 1500 | 1000 | 600 |
| จำนวนสิ่งแวดล้อม (ต่อหน่วยประมวลผล) | 8 | 7 | 4 | 3 | 2 |
| ช่วงของการอพยพ (รุ่น) | ไม่มี | 100 | 50 | 34 | 20 |

ไม่น้อยกว่าแบบเชิงลำดับ ซึ่งได้จากการทดลองและปรับค่าหลายครั้ง สาเหตุที่การทดลองนี้เปรียบเทียบกับขั้นตอนวิธีเชิงลำดับที่ใช้สิ่งแวดล้อม 8 ชุดในการเรียนรู้ เนื่องมาจากในการการปรับจำนวนสิ่งแวดล้อมให้พอดี เพื่อให้ได้ความทนทานที่ต้องการ จำนวนสิ่งแวดล้อมอาจจะมีค่ามากเกินไปทำให้ความทนทานสูงกว่าการประมวลผลเชิงลำดับมาก หรือจำนวนสิ่งแวดล้อมอาจจะมีค่าน้อยเกินไปทำให้ความทนทานต่ำกว่าการประมวลผลเชิงลำดับมาก การเพิ่มจำนวนสิ่งแวดล้อมทำให้การปรับจำนวนสิ่งแวดล้อมเป็นไปได้ละเอียดกว่าสามารถทำให้ความทนทานมีค่าใกล้เคียงกับการประมวลผลเชิงลำดับ

การทดลองแบ่งได้เป็นสองแบบคือ

1. แบบประสานเวลา จะทำการรอให้หน่วยประมวลผลทุกหน่วยดำเนินการถึงจำนวนรุ่นที่กำหนดถึงทำการอพยพ หน่วยประมวลผลที่ดำเนินการถึงจำนวนรุ่นที่กำหนดก่อนต้องทำการหยุดรอ ดังนั้นความเร็วในการทำงานของทั้งระบบจะเท่ากับความเร็วของหน่วยประมวลผลที่ช้าที่สุด

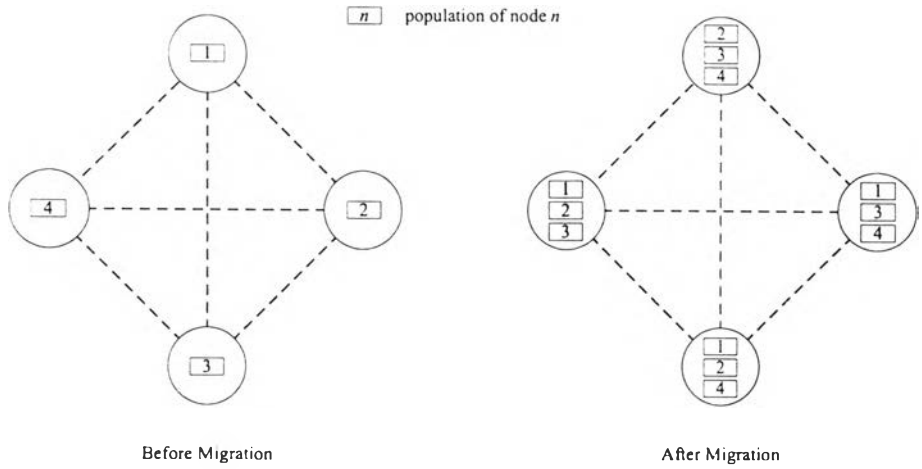
2. แบบไม่ประสานเวลา ในแต่ละรอบของการวิวัฒนาการ หน่วยประมวลผลแต่ละหน่วยจะส่งจำนวนรุ่นที่กำลังวิวัฒนาการให้หน่วยประมวลผลแรก (หมายเลขศูนย์) ในลักษณะไม่หยุดรอ (non-blocking) หน่วยประมวลผลแรกจะทำการตรวจสอบจำนวนรุ่นของแต่ละหน่วยประมวลผลกับเงื่อนไขในการอพยพ ถ้าเงื่อนไขเป็นจริง หน่วยประมวลผลแรกจะทำการส่งข้อความในลักษณะไม่หยุดรอให้กับทุกหน่วยประมวลผล เพื่อให้ทุกหน่วยประมวลผลทำการอพยพประชากร สำหรับเงื่อนไขของการอพยพที่ใช้ในการทดลองนี้คือ เมื่อหน่วยประมวลผลใดดำเนินการถึงจำนวนรุ่นที่กำหนดให้ทำการอพยพทันที ดังนั้นความเร็วในการทำงานจะเท่ากับความเร็วของหน่วยประมวลผลที่เร็วที่สุด

   เนื่องจากการตรวจสอบข้อมูลที่ส่งจากหน่วยประมวลผลแรกของแต่ละหน่วยประมวลผล จะกระทำในลักษณะการหยั่งสัญญาณ (polling) ถ้าหน่วยประมวลผลใดไม่ได้รับข้อความให้ดำเนินการอพยพก็จะดำเนินการตามกระบวนการของกำหนดการเชิงพันธุกรรมในรุ่นถัดไป ดังนั้นการอพยพจะเกิดขึ้นหลังจากที่หน่วยประมวลผลแรกทำการส่งข้อความไปยังทุกหน่วยประมวลผลอย่างช้าที่สุดเป็นจำนวน 1 รุ่น

การทดลองทั้งสองแบบจะต่างกันในการรอให้เกิดการอพยพ แต่การส่งประชากรในการอพยพจะเหมือนกัน โดยทำในแบบประสานเวลา สามารถเขียนเป็นรหัสเทียมได้ดังรูป 5.16

#### 5.3.2.2 ผลการทดลอง

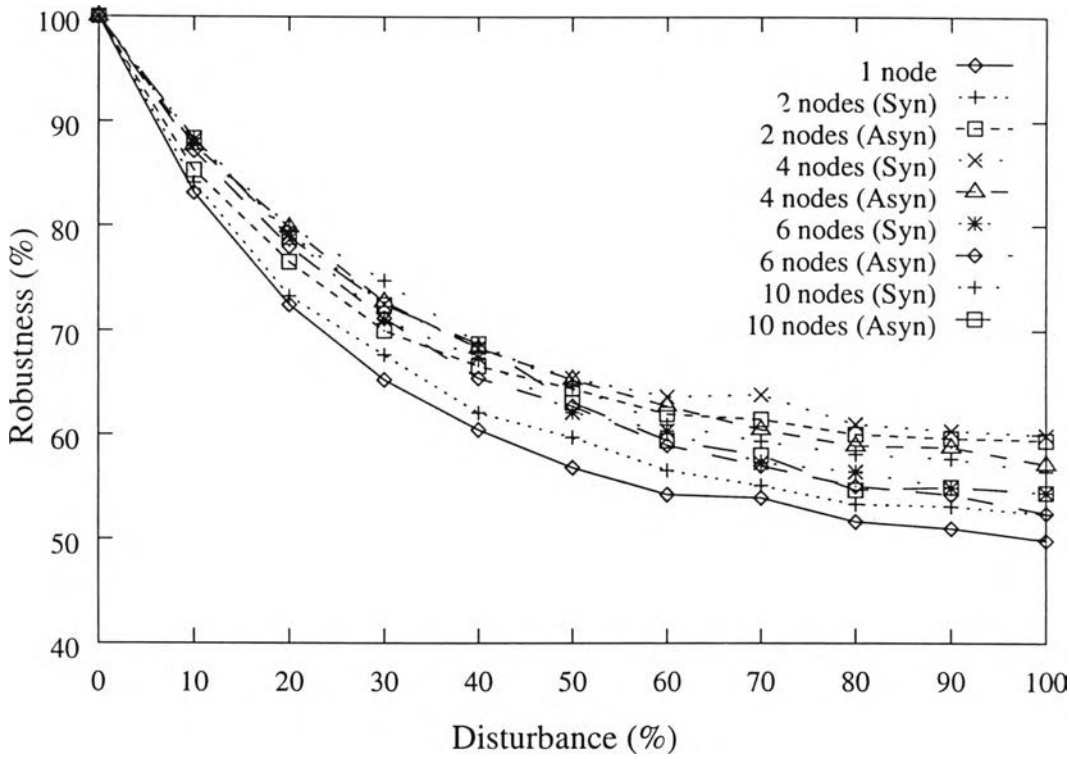
ผลการทดลองจะเฉลี่ยจากการทดลอง 20 ชุด กราฟความทนทานแสดงในรูป 5.17 เฉลี่ยจากโปรแกรมที่ดีที่สุดของแต่ละชุด แกนตั้งของกราฟเป็นความทนทานและแกนนอนเป็นค่าเปอร์เซ็นต์การรบกวนตั้งแต่ 0%




รูปที่ 5.15 การส่งประชากรของการทดลองโดยใช้สิ่งแวดล้อมต่างกัน

```
procedure Migration
begin
    barrier[1] wait all nodes ready
    for i = 1 to n
    begin
        if (my_process_id = i)
            broadcast send
        else
            broadcast receive
        barrier[2] wait for the next broadcast
    end
end
```

รูปที่ 5.16 รหัสเทียมของการแลกเปลี่ยนประชากรสำหรับการทดลองโดยใช้สิ่งแวดล้อมที่แตกต่างกันในแต่ละหน่วยประมวลผล



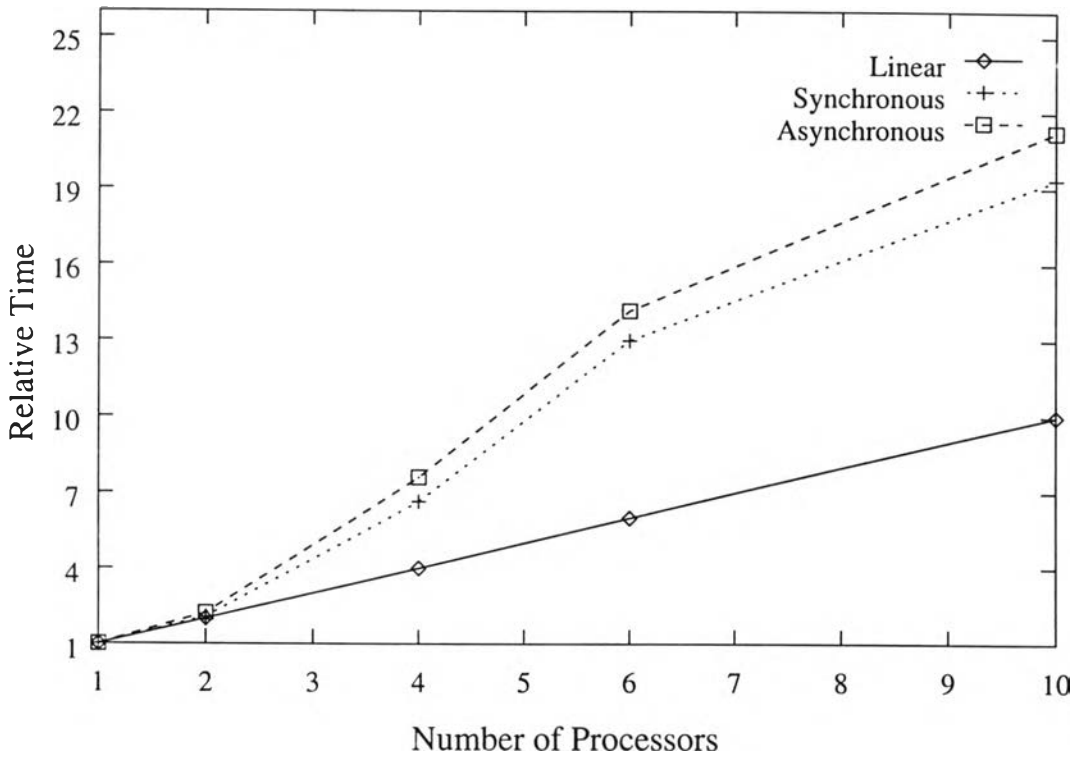

รูปที่ 5.17 ความทนทานของการทดลองโดยใช้สิ่งแวดล้อมต่างกัน

ถึง 100% แต่ละค่าเปอร์เซ็นต์การรบกวนจะใช้สิ่งแวดล้อม 1000 ชุด จากกราฟจะเห็นได้ว่าความทนทานของคำตอบจากการประมวลผลแบบขนานมีค่ามากกว่าการประมวลผลแบบเชิงลำดับ เนื่องจากเลือกใช้จำนวนสิ่งแวดล้อมอย่างน้อยที่สุดที่ความทนทานไม่น้อยกว่าจากการประมวลผลแบบเชิงลำดับ

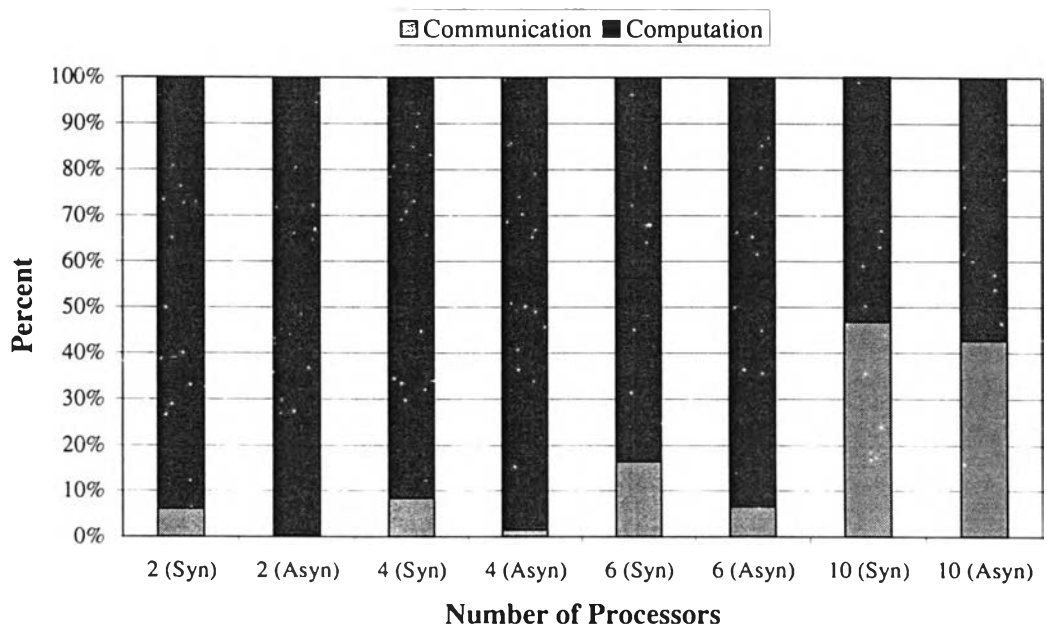
รูป 5.18 แสดงค่าเวลาสัมพัทธ์ของการทดลองทั้งสองแบบ ถึงแม้ว่าค่าเวลาสัมพัทธ์จะมีค่ามากกว่าจำนวนหน่วยประมวลผลที่ใช้ แต่ในช่วงที่ใช้จำนวนหน่วยประมวลเท่ากับ 10 ค่าเวลาสัมพัทธ์จะลดลงอย่างเห็นได้ชัด และประสิทธิภาพของการทดลองแบบไม่ประสานเวลาดีกว่าแบบประสานเวลาเพียงเล็กน้อย ดังนั้นเพื่อวิเคราะห์สาเหตุของประสิทธิภาพที่ได้ จึงทำการวิเคราะห์ในส่วนของเวลาที่เสียไปกับการสื่อสาร

เปอร์เซ็นต์ของเวลาที่ใช้ในการคำนวณและการสื่อสารแสดงในรูป 5.19 จะเห็นได้ว่าในช่วงที่หน่วยประมวลผลมีจำนวนน้อย การทำงานแบบไม่ประสานเวลาสามารถช่วยลดเปอร์เซ็นต์ของเวลาที่ใช้ในการสื่อสารได้อย่างชัดเจน เช่น ที่ 4 หน่วยประมวลผล เวลาของการสื่อสารเมื่อการทำงานเป็นแบบประสานเวลาคิดเป็น 8.4 เปอร์เซ็นต์ของเวลาทั้งหมด เมื่อใช้การทำงานแบบไม่ประสานเวลา ส่วนของเวลาในการสื่อสารเป็น 1.5 เปอร์เซ็นต์ของเวลาทั้งหมด การทำงานแบบไม่ประสานเวลาช่วยลดเวลาในการสื่อสารลง 6.9 เปอร์เซ็นต์ ในขณะที่เมื่อเพิ่มจำนวนหน่วยประมวลผล เปอร์เซ็นต์ของเวลาที่ใช้ในการสื่อสารในการทำงานแบบไม่ประสานเวลาลดลงเพียงเล็กน้อย โดยเฉพาะเมื่อใช้ 10 หน่วยประมวลผล เวลาในการสื่อสารจาก 46.7 เปอร์เซ็นต์เหลือ 42.7 เปอร์เซ็นต์ ส่วนของเวลาในการสื่อสารลดลง 4.0 เปอร์เซ็นต์ การที่เปอร์เซ็นต์ของเวลาที่ใช้ในการสื่อสารมีค่าสูงทำให้ประสิทธิภาพของทำงานแบบขนานลดลง เนื่องจากอัตราส่วนระหว่างเวลาในการคำนวณ ($R$) และเวลาในการสื่อสาร ($C$) มีค่าน้อยลง เหมือนกับในการทดลอง 5.3.1

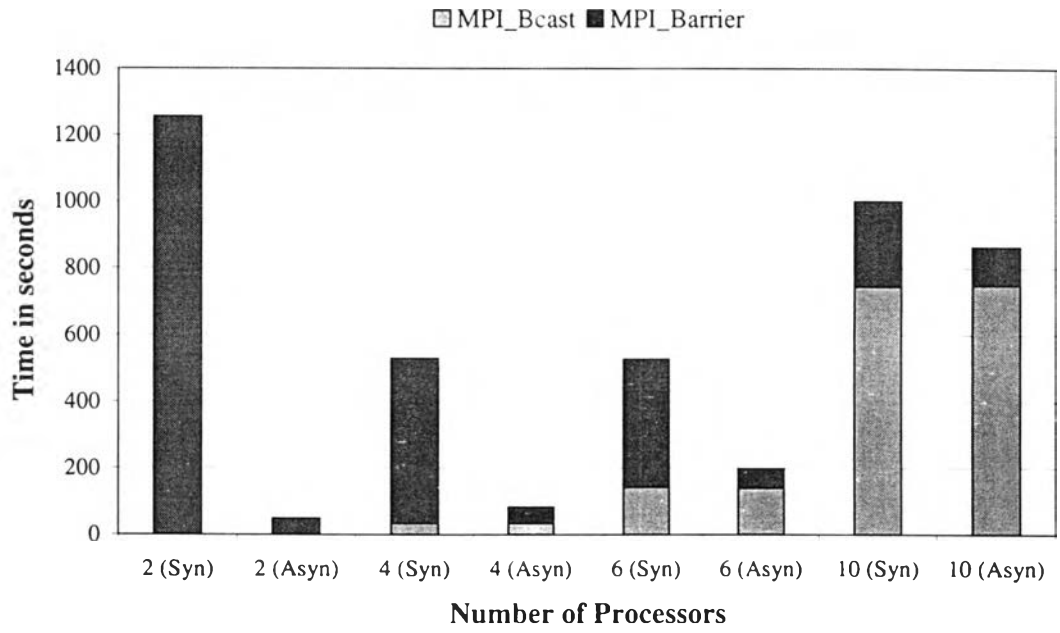
รูป 5.20 แสดงเวลาที่ใช้ในฟังก์ชันหลักของการสื่อสาร การทำงานแบบไม่ประสานเวลาช่วยลดเวลาในการรอเนื่องจากฟังก์ชันการกีดขวาง โดยการทำงานแบบไม่ประสานเวลาช่วยลดเวลาในการรอได้อย่างมี




รูปที่ 5.18 ค่าเวลาสัมพัทธ์ของการทดลองโดยใช้สิ่งแวดล้อมต่างกัน



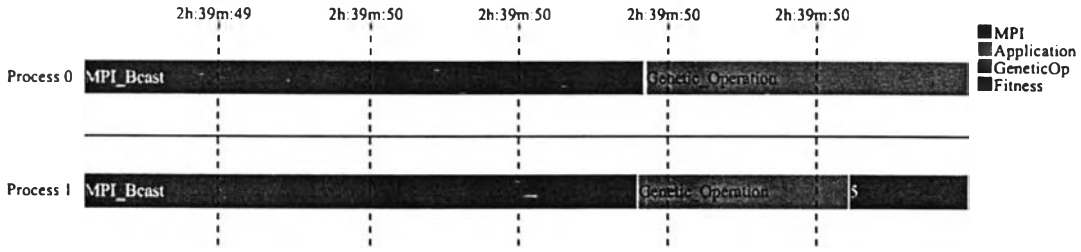

รูปที่ 5.19 เปอร์เซ็นต์ของเวลาในการสื่อสารและการคำนวณของการทดลองโดยใช้สิ่งแวดล้อมต่างกัน



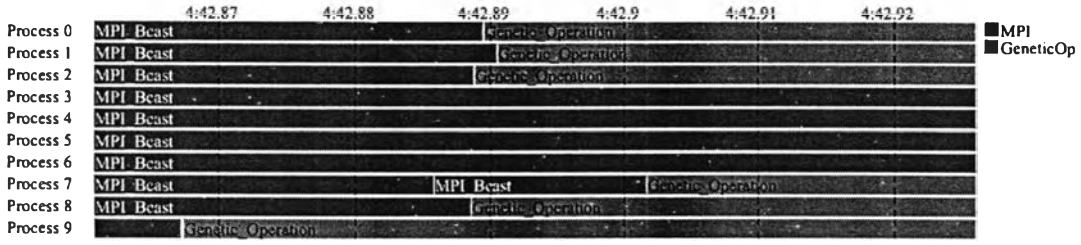

รูปที่ 5.20 เวลาที่ใช้ในการสื่อสารของการทดลองโดยใช้สิ่งแวดล้อมต่างกัน

ประสิทธิภาพเฉพาะในช่วงที่หน่วยประมวลผลมีจำนวนน้อย เช่น เวลาในการรอ (MPI_Barrier) ของ 4 หน่วยประมวลผลเมื่อทำงานแบบประสานเวลาเท่ากับ 496.4 วินาที และเมื่อทำงานแบบไม่ประสานเวลา เวลาในการรอลดลงเหลือ 48.2 วินาที เวลาลดลง 448.2 วินาที ในขณะที่เวลาในการส่งกระจายข้อมูล (MPI_Bcast) ของแบบประสานเวลา และไม่ประสานเวลามีค่าเท่ากับ 33.7 และ 34.3 วินาทีตามลำดับ ซึ่งมีค่าใกล้เคียงกัน เมื่อพิจารณาจากคำสั่งเทียมของฟังก์ชันการอพยพจะเห็นได้ว่าเวลาของการรอประกอบด้วย เวลาที่รอให้ทุกหน่วยประมวลผลพร้อมทำการอพยพ (barrier[1]) และให้ทุกหน่วยประมวลผลพร้อมสำหรับการกระจายข้อมูลครั้งต่อไป (barrier[2]) การทำงานแบบไม่ประสานเวลาช่วยลดเวลาในการรอครั้งแรก (barrier[1]) เท่านั้น เวลาในการรอครั้งที่สอง (barrier[2]) ยังคงเหลืออยู่ (ดูรหัสเทียมหน้า 44)

นอกจากนี้ในรูป 5.20 แสดงให้เห็นว่าเวลาที่ใช้ในการส่งกระจายข้อมูล (MPI_Bcast) มีการเพิ่มขึ้นอย่างรวดเร็ว โดยมากกว่าเชิงเส้น เพื่อวิเคราะห์สาเหตุจึงทำการตรวจสอบโดยใช้โปรแกรมจินตทัศน์ข้อมูล รูป 5.21 เป็นการส่งกระจายข้อมูลครั้งสุดท้ายก่อนจะเข้าสู่ช่วงของการคำนวณของ 2 หน่วยประมวลผล จากรูปจะเห็นเส้นประในแนวตั้งเป็นตัวที่บอกเวลาในแต่ละตำแหน่ง และแถบเส้นในแนวนอนที่บอกเหตุการณ์ที่เกิดขึ้นในแต่ละช่วงเวลา โดยแยกตามโปรเซส ลำดับเหตุการณ์จะเรียงจากทางซ้ายไปทางขวา ในช่วงแรกทั้งสองโปรเซสทำการส่งกระจายข้อมูล หลังจากนั้นจะเข้าสู่การทำงานของกำหนดการเชิงพันธุกรรม จากตำแหน่งที่แสดงการเปลี่ยนจากการส่งกระจายข้อมูลมาเป็นการทำงานของกำหนดการเชิงพันธุกรรม จะเห็นว่าการส่งกระจายข้อมูลเสร็จพร้อมกันทุกหน่วยประมวลผล รูป 5.22 เป็นการส่งกระจายข้อมูลครั้งสุดท้ายก่อนจะเข้าสู่ช่วงของการคำนวณของ 10 หน่วยประมวลผล จะเห็นได้ว่าที่ 10 หน่วยประมวลผล การส่งกระจายข้อมูลไม่ได้เสร็จสิ้นพร้อมกันทุกหน่วยประมวลผล ซึ่งแตกต่างจากใน 2 หน่วยประมวลผล หลังจากทำการจินตทัศน์ สามารถสรุปได้ว่าปัญหาของการส่งกระจายข้อมูลเกิดจากการส่งข้อมูลซ้ำหลายครั้งต่อการส่งกระจายข้อมูลหนึ่งครั้ง




รูปที่ 5.21 การส่งกระจายข้อมูลเมื่อใช้ 2 หน่วยประมวลผล




รูปที่ 5.22 การส่งกระจายข้อมูลเมื่อใช้ 10 หน่วยประมวลผล

การที่ใช้การส่งกระจายข้อมูลซ้ำหลายครั้งนี้ มีผลให้เวลาในการรอครั้งที่สอง (barrier[2]) เพิ่มขึ้นเช่นกัน ดังนั้นจะเห็นว่าในการประมวลผลแบบไม่ประสานเวลา ซึ่งเวลาในการรอที่เหลืออยู่เป็นเวลาในการรอครั้งที่สอง เวลาในการรอจะมีการเพิ่มขึ้นตามการเพิ่มจำนวนหน่วยประมวลผล

หลังจากทำการวิเคราะห์ด้วยการตรวจสอบเวลาที่เสียไปในส่วนของการสื่อสาร สามารถอธิบายข้อสงสัยสองประการที่เกี่ยวข้องกับประสิทธิภาพที่ได้ คือ ในเรื่องการการลดลงของประสิทธิภาพอย่างชัดเจนที่ 10 หน่วยประมวลผล และในเรื่องการทำงานแบบไม่ประสานเวลาช่วยเพิ่มประสิทธิภาพได้เพียงเล็กน้อย

ประสิทธิภาพที่ลดลงเมื่อใช้ 10 หน่วยประมวลผลเกิดจากการเพิ่มขึ้นของเวลาที่เสียไปในฟังก์ชันการส่งกระจายข้อมูลอย่างรวดเร็ว และมีค่าสูงกว่าเวลาในการรอเป็นจำนวนมาก เช่น เมื่อใช้การทำงานแบบประสานเวลา 6 หน่วยประมวลผลเวลาที่ใช้ในการส่งกระจายข้อมูลเท่ากับ 142.7 วินาที และเวลาในการรอเท่ากับ 385.1 วินาที และเมื่อใช้ 10 หน่วยประมวลผลโดยเป็นการทำงานแบบประสานเวลาเช่นกัน เวลาที่ใช้ในการส่งกระจายข้อมูลเพิ่มขึ้นเป็น 746.5 วินาที และเวลาในการรอเท่ากับ 253.2 วินาที ซึ่งการทำงานแบบไม่ประสานเวลาจะช่วยลดเวลาในการรอได้เพียงอย่างเดียว ดังนั้นเวลาในการสื่อสารยังคงเหลืออยู่มากประสิทธิภาพที่ได้จึงไม่เพิ่มเท่าที่ควร

สำหรับสาเหตุที่การทำงานแบบไม่ประสานเวลาช่วยเพิ่มประสิทธิภาพได้เพียงเล็กน้อย สามารถแบ่งได้เป็น 2 กรณี ในกรณีที่หนึ่งเมื่อหน่วยประมวลผลมีจำนวนไม่มาก ถึงแม้ว่าการทำงานแบบไม่ประสานเวลาสามารถลดเวลาในการรอได้อย่างมีประสิทธิภาพ แต่เนื่องจากเปอร์เซ็นต์ของเวลาในการสื่อสารมีค่าไม่สูงนัก การทำงานแบบประสานเวลาและไม่ประสานเวลาจึงไม่แตกต่างกันมากนัก ในกรณีที่สองเมื่อหน่วยประมวลผลมีจำนวนมาก เวลาส่วนใหญ่ของการสื่อสารเป็นเวลาในการส่งกระจายข้อมูล ซึ่งมีการเพิ่มอย่างรวดเร็ว การทำงานแบบไม่ประสานเวลาไม่ได้ช่วยลดเวลาในการส่งกระจายข้อมูล ทำให้การทำงานแบบไม่ประสานเวลาไม่ได้ช่วยเพิ่มประสิทธิภาพมากนัก

## 5.4 การวิเคราะห์ผล

### 5.4.1 ผลของเวลาในการสื่อสารต่อประสิทธิภาพ

ในส่วนนี้จะทำการวิเคราะห์ผลของอัตราส่วนระหว่างเวลาในการคำนวณ ($R$) และเวลาในการสื่อสาร ($C$) ว่ามีความเกี่ยวข้องกับค่าเวลาสัมพัทธ์ ($Rt$) อย่างไร โดยใช้สมการที่ดัดแปลงจากกฎของ Amdahl จากหนังสือ [8] ในช่วงแรกจะกล่าวถึงสมการของค่าสปีดอัพ ($S$) จากนั้นจะเป็นการดัดแปลงสมการของค่าสปีดอัพเพื่อทำการวิเคราะห์ความเกี่ยวข้องของค่าเวลาสัมพัทธ์กับค่า $R/C$

การวัดค่าสปีดอัพ ($S$) ใช้สมการดังต่อไปนี้

$$S = \frac{T_s}{T_p} \tag{5.5}$$

$T_s$ เป็นเวลาของการประมวลผลแบบเชิงลำดับ

$T_p$ เป็นเวลาของการประมวลผลแบบขนาน

ถ้างานทั้งหมดคิดเป็นหนึ่งหน่วย ส่วนของงานที่ต้องทำแบบเชิงลำดับบน 1 หน่วยประมวลผลมีค่าเป็น $\alpha$ และส่วนของงานที่สามารถทำแบบขนานบน $n$ หน่วยประมวลผลจะมีค่าเท่ากับ $(1-\alpha)$ โดยการแทนค่าลงในสมการ 5.5 สามารถสร้างเป็นสมการตามกฎของ Amdahl ได้ดังนี้

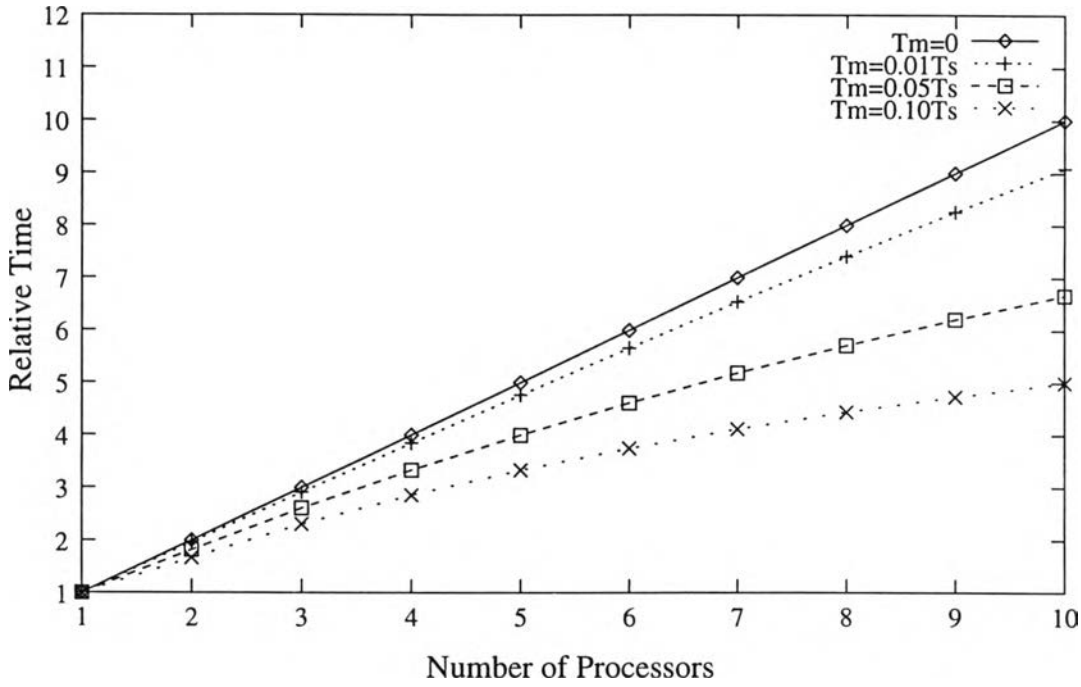
$$\begin{aligned} S &= \frac{T_s}{T_p} \\ &= \frac{T_s}{(1-\alpha)T_s/n + \alpha T_s} \\ &= \frac{n}{1+\alpha(n-1)} \end{aligned} \tag{5.6}$$

จากสมการ 5.5 จะเห็นได้ว่าสมการค่าสปีดอัพนี้เหมือนกับที่ใช้คำนวณค่าเวลาสัมพัทธ์ แต่การคำนวณค่าสปีดอัพโดยใช้สมการ 5.5 ถือว่าขนาดของงานในการทำงานแบบเชิงลำดับและแบบขนานเท่ากัน เนื่องจากขนาดของงานในการทำงานแบบเชิงลำดับและแบบขนานไม่เท่ากันในบางการทดลอง ทำให้ไม่สามารถคำนวณเป็นค่าสปีดอัพได้ จึงใช้ค่าเวลาสัมพัทธ์แทนในการนำเสนอผลการทดลองโดยใช้คุณภาพของคำตอบที่ระดับเดียวกันเป็นตัวเปรียบเทียบ ทั้งนี้ในการทดลองที่ขนาดของงานในการทำงานแบบเชิงลำดับเท่ากับแบบขนาน เช่น การทดลองในหัวข้อ 5.3.1 ผลการทดลองสามารถแสดงเป็นค่าสปีดอัพได้เช่นเดียวกัน

จากการที่ค่าเวลาสัมพัทธ์สามารถคำนวณได้เช่นเดียวกับสมการ 5.5 ดังนั้นเราสามารถสร้างสมการค่าเวลาสัมพัทธ์ได้เช่นเดียวกับสมการ 5.6 ดังนี้

$$\begin{aligned} Rt &= \frac{T_s}{T_p} \\ &= \frac{T_s}{(1-\alpha)T_s/n + \alpha T_s} \\ &= \frac{n}{1+\alpha(n-1)} \end{aligned} \tag{5.7}$$

จากสมการ 5.6 ถ้าการประมวลผลแบบขนานทำงานแบบหัวหน้า-ผู้ช่วย โดยการกระจายส่วนที่เป็นการหาค่าความเหมาะสมให้ทำงานแบบขนาน และส่วนที่เหลือทำงานแบบเชิงลำดับบนหน่วยประมวลผลที่เป็นหัว




รูปที่ 5.23 ผลของเวลาในการสื่อสารของฟังก์ชันการอพยพต่อค่าเวลาสัมพัทธ์

หน้า ค่า $\alpha$ จะเป็นอัตราส่วนระหว่างงานที่ต้องทำแบบเชิงลำดับกับงานทั้งหมด ซึ่งงานที่ต้องทำแบบเชิงลำดับได้แก่ การเลือกประชากรและการสร้างประชากรรุ่นใหม่

แต่ในการทดลอง การประมวลผลแบบขนานทำงานแบบหน่วยหยาบ ซึ่งการทำงานในแต่ละหน่วยประมวลผลเป็นอิสระต่อกัน และงานที่ต้องทำบนหน่วยประมวลผลเดียวมีค่าน้อยมาก ดังนั้นค่า $\alpha$ มีค่าเข้าใกล้ศูนย์ นอกจากนี้การประมวลผลแบบขนานจะเพิ่มฟังก์ชันการอพยพที่ไม่มีในการประมวลผลเชิงลำดับ ถ้าเวลาที่ใช้ในการสื่อสารของฟังก์ชันการอพยพเป็น $T_m$ และ $\alpha$ มีค่าเข้าใกล้ศูนย์ สามารถประมาณค่าเวลาสัมพัทธ์ $(Rt)$ ได้ดังนี้

$$\begin{aligned} Rt &= \frac{T_s}{T_p} \\ &= \frac{T_s}{(1-\alpha)T_s/n + \alpha T_s + T_m} \\ &= \frac{T_s}{T_s/n + T_m} \\ &= \frac{nT_s}{T_s + nT_m} \end{aligned} \tag{5.8}$$

จากสมการ 5.8 ถ้า $T_m$ มีค่าน้อยมาก $(T_m \to 0)$ ค่าเวลาสัมพัทธ์จะเท่ากับจำนวนหน่วยประมวลผลที่ใช้ และถ้าค่า $T_m$ มีค่ามากขึ้นจะมีผลให้ค่าเวลาสัมพัทธ์ที่ได้มีค่าลดลง ดังแสดงในรูป 5.23 โดยการกำหนดให้ $T_m$ เป็น 0, 1, 5 และ 10 เปอร์เซ็นต์ของ $T_s$

จากผลการทดลอง การลดลงของค่า $R/C$ เกิดจากสองสาเหตุ คือ การลดลงของค่า $R$ โดยเมื่อเพิ่มจำนวนหน่วยประมวลผล ทำให้ขนาดของงานต่อ 1 หน่วยประมวลผลมีค่าลดลง จากรูป 5.23 ถ้าเวลาในการอพยพ $(T_m)$ มีค่าคงที่ จะเห็นได้ว่าการเพิ่มหน่วยประมวลผล ทำให้ค่าเวลาสัมพัทธ์ที่ได้เมื่อเทียบกับจำนวน

หน่วยประมวลผลที่ใช้มีการลดลง และอีกสาเหตุหนึ่งคือการเพิ่มขึ้นของค่า $C$ ซึ่งจากสมการ 5.8 การเพิ่มขึ้นของค่า $C$ นี้ทำให้ค่าเวลาสัมพัทธ์ที่ได้เมื่อเทียบกับจำนวนหน่วยประมวลผลที่ใช้ลดลง

### 5.4.2 ค่าเวลาสัมพัทธ์ของการทดลองโดยใช้สิ่งแวดล้อมแตกต่างกัน

ในการทดลองโดยใช้สิ่งแวดล้อมแตกต่างกัน ค่าเวลาสัมพัทธ์ที่ได้มีค่ามากกว่าจำนวนหน่วยประมวลผลที่ใช้ เนื่องมาจากจำนวนสิ่งแวดล้อมต่อหน่วยประมวลผลมีค่าลดลงเมื่อทำการเพิ่มจำนวนหน่วยประมวลผล ดังนั้นการประมวลผลแบบขนานจึงทำงานน้อยกว่าการประมวลผลแบบเชิงลำดับ

ถ้า $M_p$ เป็นจำนวนสิ่งแวดล้อมต่อหน่วยประมวลผล ของการทำงานแบบขนานโดยใช้ $n$ หน่วยประมวลผล และ $M_s$ เป็นจำนวนสิ่งแวดล้อมของการทำงานแบบเชิงลำดับ เวลาในส่วนของการคำนวณจะลดลง $M_p/M_s$ ของเวลาเดิม เมื่อแทนค่าลงในสมการ 5.8 จะได้สมการสำหรับการประมาณค่าเวลาสัมพัทธ์ของการทดลองโดยแบ่งประชากร และใช้สิ่งแวดล้อมแตกต่างกันได้ดังนี้

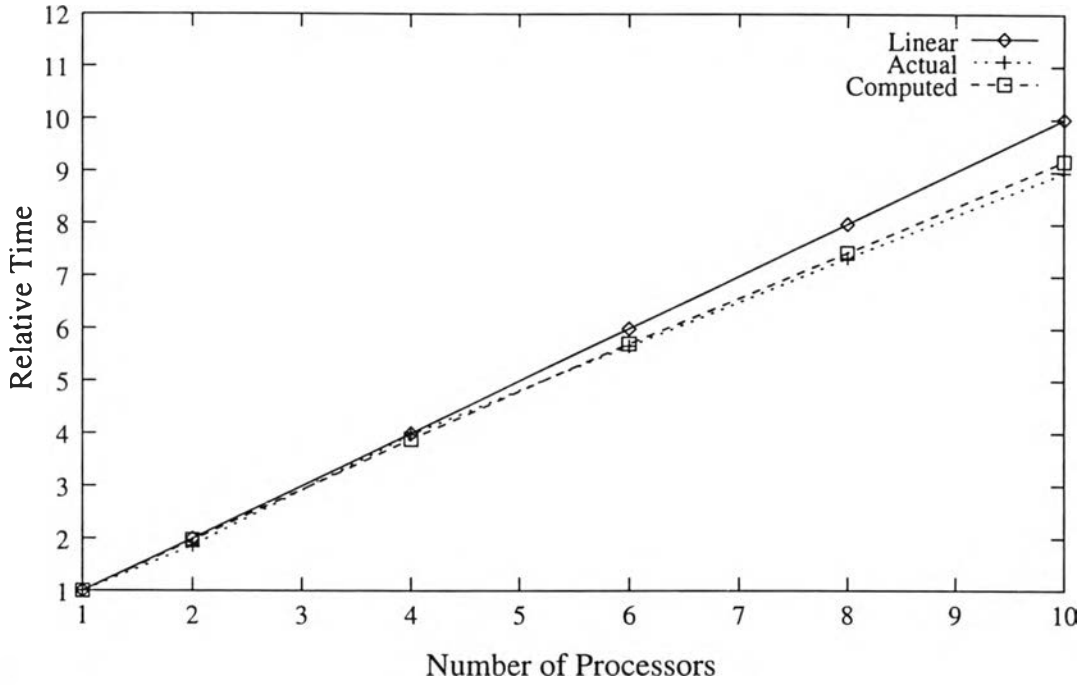
$$\begin{aligned} Rt &= \frac{T_s}{(M_p/M_s)(T_s/n) + T_m} \\ &= \frac{nT_s}{(M_p/M_s)T_s + nT_m} \end{aligned} \tag{5.9}$$

เพื่อเปรียบเทียบค่าเวลาสัมพัทธ์ที่คำนวณได้จากสมการ 5.8 และ 5.9 กับค่าเวลาสัมพัทธ์จากการทดลอง ในรูป 5.24 แสดงการเปรียบเทียบค่าเวลาสัมพัทธ์จากการทดลองโดยใช้สิ่งแวดล้อมเหมือนกัน และจากการคำนวณด้วยสมการ 5.8 และรูป 5.25 แสดงการเปรียบเทียบค่าเวลาสัมพัทธ์จากการทดลองโดยใช้สิ่งแวดล้อมต่างกัน และจากการคำนวณด้วยสมการ 5.9 โดยค่า $T_s$ และ $T_m$ ที่ใช้ในการคำนวณได้มาจากผลการทดลอง จากกราฟจะเห็นได้ว่าค่าเวลาสัมพัทธ์จากการทดลองจะต่ำกว่าที่ได้จากการคำนวณเล็กน้อย เนื่องมาจากในการคำนวณละเลยเวลาบางส่วนที่ต้องทำในการประมวลผลแบบขนาน

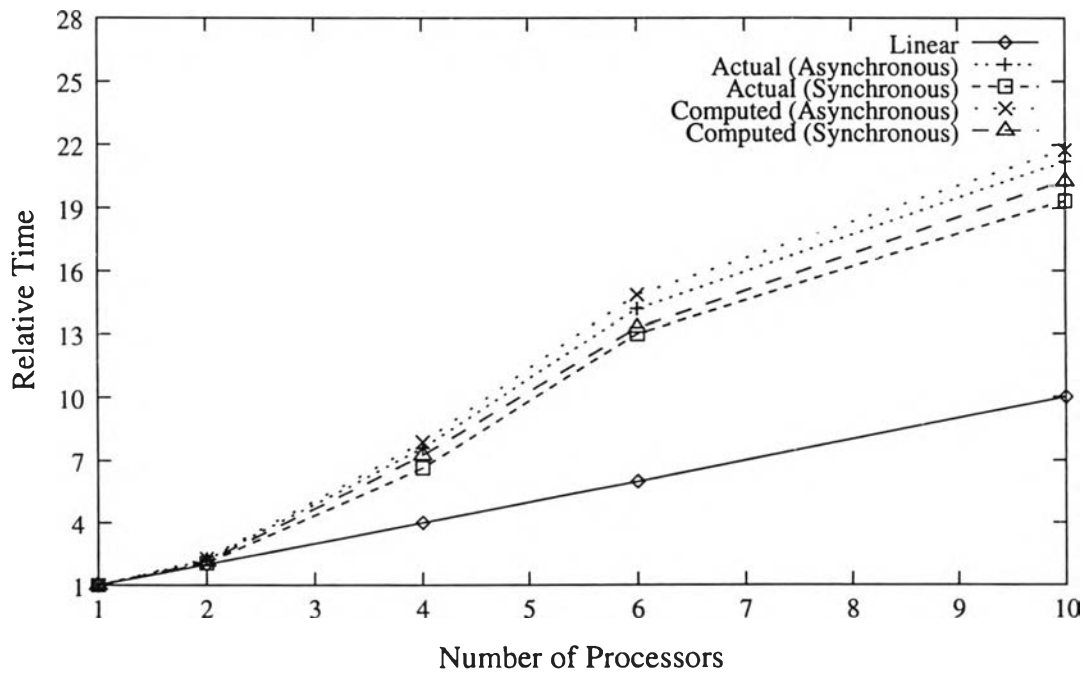
## 5.5 สรุปท้ายบท

การทดลองการประมวลผลแบบขนานสำหรับปัญหาการนำร่องหุ่นยนต์ โดยเพิ่มจำนวนสิ่งแวดล้อมแบ่งออกเป็นสองส่วน ผลการทดลองมีการเปรียบเทียบทั้งในด้านคุณภาพของคำตอบ และประสิทธิภาพการประมวลผลแบบขนาน ในส่วนแรกเป็นการทดลองที่แต่ละหน่วยประมวลผลใช้สิ่งแวดล้อมสำหรับกระบวนการเรียนรู้ที่เหมือนกัน คุณภาพของคำตอบที่ได้ระหว่างการประมวลผลแบบเชิงลำดับ และการประมวลผลแบบขนานมีค่าใกล้เคียงกัน ประสิทธิภาพการประมวลผลแบบขนานโดยพิจารณาจากค่าเวลาสัมพัทธ์มีค่าใกล้เคียงกับจำนวนหน่วยประมวลผลที่ใช้ นอกจากนี้เวลาในการสื่อสารส่วนใหญ่คือ เวลาในการหยุดรอ (barrier) เพื่อให้ทุกหน่วยประมวลผลพร้อมสำหรับการสื่อสาร และเวลาส่วนที่เหลือของการสื่อสารคือ เวลาในการรับส่งข้อมูล

ในส่วนที่สองเป็นการทดลองที่แต่ละหน่วยใช้สิ่งแวดล้อมที่แตกต่างกันสำหรับกระบวนการเรียนรู้ เนื่องจากการใช้สิ่งแวดล้อมที่แตกต่างกันทำให้สามารถลดจำนวนสิ่งแวดล้อมในแต่ละหน่วยประมวลผล ผลที่ได้คือความเร็วในการทำงานที่เพิ่มขึ้น การทดลองเปรียบเทียบการทำงานสองแบบคือ แบบประสานเวลา และแบบไม่ประสานเวลา จำนวนสิ่งแวดล้อมของการทำงานแบบขนานเลือกจากการทดลองปรับค่าหลายครั้ง จน




รูปที่ 5.24 ค่าเวลาสัมพัทธ์จากการทดลองและการคำนวณของการทดลองโดยใช้สิ่งแวดล้อมเหมือนกัน




รูปที่ 5.25 ค่าเวลาสัมพัทธ์จากการทดลองและการคำนวณของการทดลองโดยใช้สิ่งแวดล้อมต่างกัน

คุณภาพของคำตอบไม่น้อยกว่าคุณภาพของคำตอบจากการทำงานแบบเชิงลำดับ ในส่วนของประสิทธิภาพการทำงานแบบขนาน ค่าเวลาสัมพัทธ์ของการทำงานทั้งแบบแบบประสานเวลา และแบบไม่ประสานเวลามีค่ามากกว่าจำนวนหน่วยประมวลผลที่ใช้ โดยการทำงานแบบไม่ประสานเวลาสามารถเพิ่มประสิทธิภาพได้เพียงเล็กน้อย เนื่องจากการทำงานแบบไม่ประสานเวลาสามารถลดเวลาในการสื่อสารได้เพียงบางส่วนเท่านั้น นอกจากนี้เมื่อเพิ่มจำนวนหน่วยประมวลผลถึง 10 หน่วย ค่าเวลาสัมพัทธ์ที่ได้เมื่อเทียบกับจำนวนหน่วยประมวลผลที่ใช้มีการลดลงอย่างชัดเจน จากการวิเคราะห์พบว่าเกิดจากเวลาที่ใช้ในการส่งกระจายข้อมูล (broadcast) มีการเพิ่มขึ้นเป็นอย่างมาก

ในส่วนของการวิเคราะห์ได้แสดงการคำนวณเพื่อประมาณค่าเวลาสัมพัทธ์ สมการแสดงความสัมพันธ์ของขนาดของงานที่จะใช้กับการประมวลผลแบบขนาน (ในรูปของเวลาในการประมวลผลแบบเชิงลำดับ), เวลาในการสื่อสาร (เวลาในการอพยพ) และค่าเวลาสัมพัทธ์ที่ประมาณได้ จากสมการแสดงให้เห็นว่าถ้าขนาดของงานที่จะนำมาใช้กับการประมวลผลแบบขนานมีขนาดใหญ่ขึ้น หรือเวลาในการสื่อสารลดลง ค่าเวลาสัมพัทธ์ที่ได้จะสูงขึ้น