บทที่ 4

การออกแบบการทดลอง

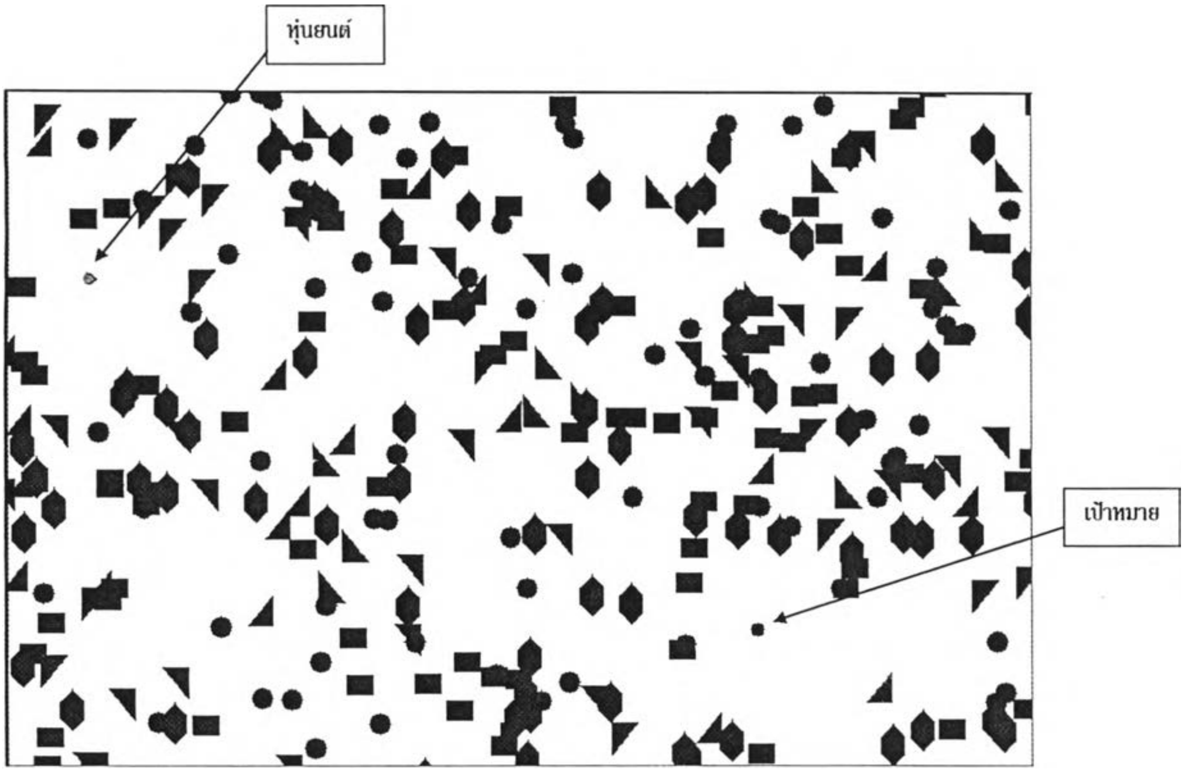
ในบทนี้กล่าวถึง รายละเอียดของการทดลองเพื่อใช้วิธีกำหนดการเชิงพันธุกรรมในการค้นหาโปรแกรมควบคุมหุ่นยนต์ ในปัญหาหุ่นยนต์เดินหลบหลีกสิ่งกีดขวางเพื่อไปหาเป้าหมาย ในสภาพจำลองบนคอมพิวเตอร์ โดยอธิบายถึงลักษณะของปัญหา และลักษณะของหุ่นยนต์ในการทดลอง รวมไปถึงรายละเอียดต่างๆ ของกำหนดการเชิงพันธุกรรม ซึ่งยังไม่มีการปรับปรุงกระบวนการวิวัฒนาการที่ใช้ในการทดลอง วิธีการทดสอบความทนทานของโปรแกรมควบคุมหุ่นยนต์ที่ได้จากวิธีกำหนดการเชิงพันธุกรรม ผลการทดสอบความทนทาน และสรุปท้ายบท

4.1 <u>ลักษณะของปัญหา</u>

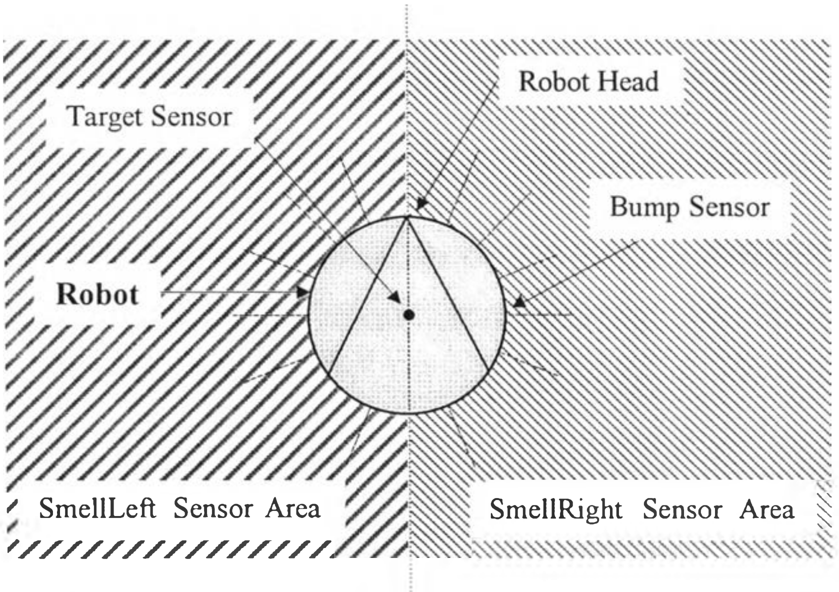
ปัญหาที่นำมาใช้ในการทดลองนี้ คือ ปัญหาการควบคุมหุ่นยนต์ให้เดินในสภาพแวดล้อมที่มีสิ่งกีดขวางวางกระจัดกระจายอยู่เป็นจำนวนมากเพื่อจะไปหาเป้าหมาย ในการทดลองนี้ได้ทำการสร้างสภาพจำลองขึ้นมาบนคอมพิวเตอร์ ซึ่งปัญหาจะคล้ายกับปัญหาที่ใช้ในงานวิจัยของ ธีระ โตสุโขวงศ์ และทัศสิน บัวชื่น (2540) และ Chongstitvatana (1998) แต่ในงานวิจัยนี้ได้เพิ่มขนาดของสนาม และหุ่นยนต์จะมีลักษณะการทำงานแตกต่างออกไปเล็กน้อย เพื่อให้เหมาะสมและให้ปัญหามีความยากขึ้น ลักษณะของปัญหา และสภาพแวดล้อมที่ใช้ในการทดลองเป็นดังรูปที่ 4.1

สำหรับรายละเอียดของสภาพแวดล้อมที่ในการทดลองจะเป็นดังนี้ สภาพแวดล้อมมีขนาด 500x750 หน่วย บริเวณโดยรอบจะเป็นกำแพง ซึ่งถือเป็นสิ่งกีดขวางล้อมรอบทั้งหมด ภายในสภาพแวดล้อมมีสิ่งกีดขวางวางกระจัดกระจายอยู่เป็นจำนวนมาก สิ่งกีดขวางมีลักษณะเป็นรูปเรขาคณิตหลายชนิด เช่น สี่เหลี่ยม วงกลม สามเหลี่ยม และ รูปหกเหลี่ยม ซึ่งสามารถวางเหลื่อมซ้อนกันได้ สิ่งกีดขวางแต่ละชิ้นมีขนาดโดยเฉลี่ยประมาณ 15x20 หน่วยและพื้นที่ของสิ่งกีดขวางทั้งหมดโดยรวม คิดเป็นพื้นที่ประมาณ 20% ของพื้นที่สภาพแวดล้อมทั้งหมด สำหรับตำแหน่งเริ่มต้นของหุ่นยนต์ และตำแหน่งของเป้าหมายได้ถูกกำหนดไว้ในสภาพแวดล้อมแล้ว โดยตำแหน่งเริ่มต้นของหุ่นยนต์อยู่ที่พิกัด (60, 140) ในทิศทางหันไปด้านขวาและเป้าหมายอยู่ที่พิกัด (550, 400) สิ่งกีดขวางจะอยู่ห่างจากตำแหน่งเริ่มต้นของหุ่นยนต์ และตำแหน่งเป้าหมาย ในระยะไม่น้อยกว่า 40 หน่วย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.1 แสดงสภาพแวดล้อมของปัญหาหุ่นยนต์เดินหลบหลีกสิ่งกีดขวางไปยังเป้าหมายที่ใช้ในการทดลองนี้



รูปที่ 4.2 แสดงลักษณะของหุ่นยนต์ในสภาพจำลองที่ใช้ในการทดลองนี้

สำหรับเป้าหมายจะมีลักษณะเป็นวงกลมรัศมี 5 หน่วย ซึ่งอยู่ตำแหน่งเดิมตลอดการทดลอง

ลักษณะของหุ่นยนต์ในสภาพจำลองที่ใช้จะมีลักษณะดังรูป 4.2 ลำตัวเป็นวงกลมรัศมี 5 หน่วย สามารถหมุนหัวไปทางซ้าย หรือขวาได้ครั้งละ 22.5 องศา โดยที่ตำแหน่งยังคงอยู่ที่เดิม และสามารถเดินไปข้างหน้าตามทิศทางที่หันหัวอยู่ได้ครั้งละ 1 หน่วย โดยตำแหน่งหุ่นยนต์จะอยู่บนพิกัดจำนวนจริง นอกจากนี้ยังมีเซนเซอร์ติดอยู่ที่ตัวหุ่นยนต์อีก 3 ชนิด ได้แก่

- เซนเซอร์บอกการชน (Bump Sensor) มีอยู่ 16 ชิ้นติดอยู่รอบๆที่ตัวหุ่นยนต์ แต่ละชิ้นวางห่างกัน 22.5 องศา เซนเซอร์ทั้งหมดนี้จะทำงานโดยอัตโนมัติทุกครั้งที่หุ่นยนต์มีการเดิน โดยไม่ต้องรับคำสั่งภายนอกมาสั่งจะทำการตรวจสอบว่ามีการชนกับสิ่งกีดขวางหรือไม่ โดยเซนเซอร์บอกการชนจะรายงานทุกครั้งที่เกิดการชนกับสิ่งกีดขวาง
- เซนเซอร์บอกเป้าหมาย (Smell Sensor) มีอยู่ 2 ชิ้นคือ เซนเซอร์บอกเป้าหมายทางด้านซ้าย (SmellLeft Sensor) และเซนเซอร์บอกเป้าหมายทางด้านขวา (SmellRight Sensor) เซนเซอร์ทั้ง 2 ชิ้นนี้ต้องรับคำสั่งจากภายนอกมาสั่งการให้เซนเซอร์ทำงาน โดยที่เซนเซอร์บอกเป้าหมายทางด้านซ้าย จะรายงานว่ามีเป้าหมายอยู่ในพื้นที่ทางด้านซ้ายของตัวหุ่นยนต์หรือไม่ ไม่ว่าเป้าหมายอยู่ไกลเท่าไรหรือมีสิ่งกีดขวางขวางอยู่หรือไม่ ส่วนเซนเซอร์บอกเป้าหมายทางด้านขวา จะรายงานว่ามีเป้าหมายอยู่ในพื้นที่ทางด้านขวาของตัวหุ่นยนต์หรือไม่
- เซนเซอร์ตรวจจับเป้าหมาย (Target Sensor) มีอยู่ 1 ชิ้นที่ตำแหน่งจุดศูนย์กลางของตัวหุ่นยนต์ ซึ่งจะทำงานโดยอัตโนมัติทุกครั้งที่หุ่นยนต์มีการเดิน โดยไม่ต้องรับคำสั่งภายนอกมาสั่ง จะทำการตรวจจับว่าขณะนี้หุ่นยนต์เดินไปพบเป้าหมายหรือยัง โดยถือว่าหุ่นยนต์ทำงานได้สำเร็จเมื่อเซนเซอร์นี้รายงานว่าหุ่นยนต์เดินไปพบเป้าหมายแล้ว

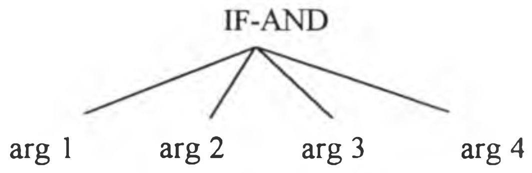
4.2 <u>รายละเอียดของกำหนดการเชิงพันธุกรรม</u>

ในส่วนนี้อธิบายถึงรายละเอียดต่างๆ ของการนำวิธีกำหนดการเชิงพันธุกรรมเพื่อมาใช้แก้ปัญหาดังกล่าว โดยแยกเป็นหัวข้อย่อยที่กล่าวถึงฟังก์ชัน และเทอร์มินอลที่ประกอบขึ้นมาเป็นโปรแกรมควบคุม รูปแบบและวิธีประมวลผลโปรแกรมควบคุม รวมทั้งกระบวนการต่างๆ ในกำหนดการเชิงพันธุกรรม

4.2.1 ฟังก์ชัน และเทอร์มินอล

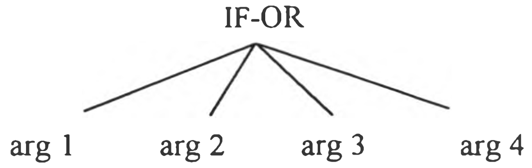
การเลือกฟังก์ชันและเทอร์มินอลพยายามจะเลือกใช้ฟังก์ชันที่เป็นมาตรฐานและง่าย และเลือกเทอร์มินอลเพื่อให้สอดคล้องกับลักษณะของหุ่นยนต์ เพื่อให้เชื่อมั่นได้ว่าไม่มีปัจจัยอื่นมารบกวนหรือมีผลต่อความทนทานของคำตอบที่ได้ รายละเอียดของฟังก์ชันและเทอร์มินอลที่ใช้มีดังต่อไปนี้

ฟังก์ชันที่เลือกใช้ เป็นตัวดำเนินการแบบเงื่อนไข ซึ่งเป็นมาตรฐานทั่วๆ ไป สามารถประยุกต์ใช้ได้กับปัญหาเกือบทุกประเภท จะไม่เลือกใช้ฟังก์ชันที่มีความเฉพาะเจาะจงกับปัญหานี้เท่านั้น ซึ่งฟังก์ชันที่ใช้มีอยู่ 3 ชนิด ได้แก่



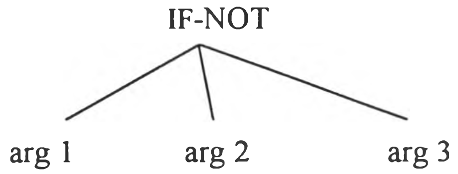
รูปที่ 4.3 แสดงต้นไม้ซึ่งแทนโปรแกรมที่ประกอบด้วยฟังก์ชัน IF-AND

- ฟังก์ชัน IF-AND แสดงในรูปต้นไม้ดังในรูปที่ 4.3 เป็นฟังก์ชันที่มี 4 อาร์กิวเมนต์ โดยจะประมวลผลในอาร์กิวเมนต์ที่ 1 และ 2 ตามลำดับเสียก่อน ถ้าผลการดำเนินการตรรกแบบ AND ของค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ทั้งสองเป็นจริง ก็จะทำการประมวลผลอาร์กิวเมนต์ที่ 3 และค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ที่ 3 ก็จะเป็นค่าส่งคืนของการประมวลผลฟังก์ชัน IF-AND นี้ นอกเหนือจากนั้นก็จะไปประมวลผลอาร์กิวเมนต์ที่ 4 และค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ที่ 4 ก็จะเป็นค่าส่งคืนของการประมวลผลฟังก์ชัน IF-AND นี้



รูปที่ 4.4 แสดงต้นไม้ซึ่งแทนโปรแกรมที่ประกอบด้วยฟังก์ชัน IF-OR

- ฟังก์ชัน IF-OR แสดงในรูปต้นไม้ดังในรูปที่ 4.4 เป็นฟังก์ชันที่มี 4 อาร์กิวเมนต์เช่นกัน โดยจะประมวลผลในอาร์กิวเมนต์ที่ 1 และ 2 ตามลำดับ ถ้าผลการดำเนินการตรรกแบบ OR ของค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ทั้งสองเป็นจริงก็จะทำการประมวลผลอาร์กิวเมนต์ที่ 3 และค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ที่ 3 ก็จะเป็นค่าส่งคืนของการประมวลผลฟังก์ชัน IF-OR นี้ นอกเหนือจากนั้นก็จะไปประมวลผลอาร์กิวเมนต์ที่ 4 และค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ที่ 4 ก็จะเป็นค่าส่งคืนของการประมวลผลฟังก์ชัน IF-OR นี้



รูปที่ 4.5 แสดงต้นไม้ซึ่งแทนโปรแกรมที่ประกอบด้วยฟังก์ชัน IF-NOT

- ฟังก์ชัน IF-NOT แสดงในรูปต้นไม้ดังในรูปที่ 4.5 เป็นฟังก์ชันที่มีเพียง 3 อาร์กิวเมนต์ โดยจะประมวลผลในอาร์กิวเมนต์ที่ 1 ก่อน ถ้านิเสธของค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ที่ 1 เป็นจริง ก็จะทำการประมวลผลอาร์กิวเมนต์ที่ 2 และค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ที่ 2 ก็จะเป็นค่าส่งคืนของการประมวลผลฟังก์ชัน IF-NOT นี้ แต่ถ้านิเสธของค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ที่ 1 เป็นเท็จจะไปประมวลผลอาร์กิวเมนต์ที่ 3 และค่าส่งคืนของการประมวลผลในอาร์กิวเมนต์ที่ 3 ก็จะเป็นค่าส่งคืนของการประมวลผลฟังก์ชัน IF-NOT นี้

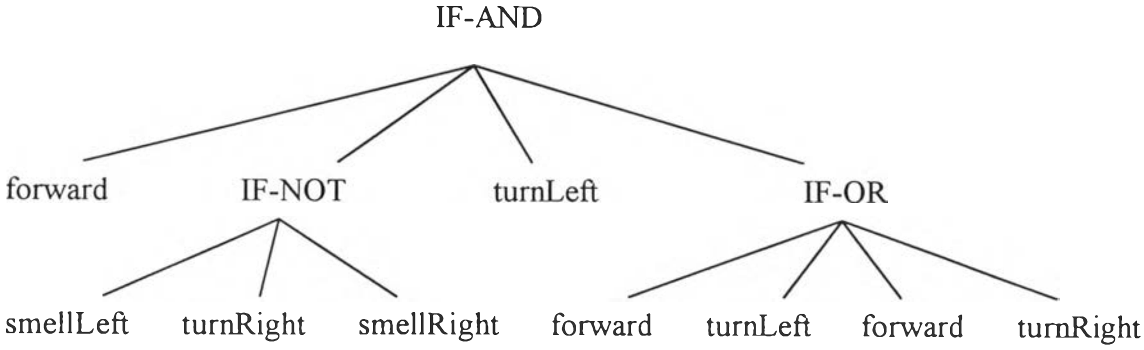
สำหรับเทอร์มินอลที่ใช้ จะเป็นคำสั่งที่ใช้สั่งงานหุ่นยนต์โดยตรง เมื่อประมวลผลเทอร์มินอลแต่ละตัวจะต้องส่งคืนค่าให้กับโปรแกรม รูปแบบของเทอร์มินอลที่ใช้มี 5 ชนิด ได้แก่

- forward เป็นคำสั่งให้หุ่นยนต์เดินหน้าไป 1 หน่วยในทิศทางที่หุ่นยนต์หันหัวอยู่ การสั่งคำสั่งนี้จะมีผลให้เซนเซอร์บอกการชนมีการทำงานโดยอัตโนมัติ และค่าส่งคืนของคำสั่งนี้มีค่าเป็นจริงเมื่อเซนเซอร์บอกการชน รายงานว่าไม่มีการชนกับสิ่งกีดขวาง และค่าส่งคืนของคำสั่งนี้มีค่าเป็นเท็จและหุ่นยนต์จะไม่มีการเดินเมื่อเซนเซอร์บอกการชนรายงานว่าเกิดการชนกับสิ่งกีดขวาง
- turnLeft เป็นคำสั่งให้หุ่นยนต์หันทิศทางไปทางซ้าย 22.5 องศา โดยที่ตำแหน่งของหุ่นยนต์ยังคงอยู่ที่เดิม ค่าส่งคืนของคำสั่งนี้จะมีค่าเป็นจริงเสมอ
- turnRight เป็นคำสั่งให้หุ่นยนต์หันทิศทางไปทางขวา 22.5 องศา โดยที่ตำแหน่งของหุ่นยนต์ยังคงอยู่ที่เดิม ค่าส่งคืนของคำสั่งนี้จะมีค่าเป็นจริงเสมอ
- smellLeft เป็นคำสั่งให้เซนเซอร์บอกเป้าหมายทางด้านซ้ายทำงาน เพื่อตรวจดูว่ามีเป้าหมายอยู่ในพื้นที่ทางด้านซ้ายของตัวหุ่นยนต์หรือไม่ ค่าส่งคืนของคำสั่งนี้จะมีค่าเป็นจริงเมื่อเซนเซอร์บอกเป้าหมายทางด้านซ้ายรายงานว่าพบเป้าหมายอยู่ในพื้นที่ทางด้านซ้ายของหุ่นยนต์ ถ้าไม่พบก็จะส่งคืนค่าเท็จให้กับโปรแกรม
- smellRight เป็นคำสั่งให้เซนเซอร์บอกเป้าหมายทางด้านขวาทำงานเพื่อตรวจดูว่ามีเป้าหมายอยู่ในพื้นที่ทางด้านขวาของตัวหุ่นยนต์หรือไม่ ค่าส่งคืนของคำสั่งนี้จะมีค่าเป็นจริงเมื่อเซนเซอร์บอกเป้าหมายทางด้านขวารายงานว่าพบเป้าหมายอยู่ในพื้นที่ทางด้านขวาของหุ่นยนต์ ถ้าไม่พบก็จะส่งคืนค่าเท็จให้กับโปรแกรม

4.2.2 รูปแบบของโปรแกรมควบคุมหุ่นยนต์ และวิธีการประมวลผล

โปรแกรมควบคุมหุ่นยนต์จะประกอบขึ้นด้วย ฟังก์ชันและเทอร์มินอลที่ได้กำหนดไว้ โดยในการประมวลผลโปรแกรมจะทำจนกระทั่งไปถึงตำแหน่งของเทอร์มินอล ซึ่งเทอร์มินอลอาจเป็นการกระทำอย่างใดอย่างหนึ่งเช่น การเคลื่อนที่ การเปลี่ยนทิศทาง การตรวจเป้าหมาย เป็นต้น ก็ให้ทำการกระทำนั้น แล้วทำการประมวลผลต่อไป ถ้าประมวลผลไปจนสุดต้นไม้แล้วให้ย้อนกลับไปเริ่มต้นที่รากของต้นไม้ใหม่จนกว่าจะสิ้นสุดเงื่อนไขในการวนซ้ำที่กำหนด ตัวอย่างของโปรแกรมควบคุมดังรูปที่ 4.6 ซึ่งแทนโปรแกรม

(IF-AND forward (IF-NOT smellLeft turnRight smellRight) turnLeft (IF-OR forward turnLeft forward turnRight))



รูปที่ 4.6 แสดงต้นไม้ซึ่งแทนโปรแกรมคอมพิวเตอร์ (IF-AND forward (IF-NOT smellLeft turnRight smellRight) turnLeft (IF-OR forward turnLeft forward turnRight))

เมื่อนำโปรแกรมนี้ไปประมวลผลจะทำการประมวลผลคำสั่งในต้นไม้ โดยเริ่มจากฟังก์ชัน IF-AND ซึ่งจะไปประมวลผลอาร์กิวเมนต์ที่ 1 คือ forward และ อาร์กิวเมนต์ที่ 2 คือ ฟังก์ชัน IF-NOT ซึ่งจะไปประมวลผลคำสั่ง smellLeft ที่เป็นอาร์กิวเมนต์ที่ 1 ของฟังก์ชันนี้ ถ้านิเสธของค่าส่งคืนของคำสั่ง smellLeft เป็นจริง ก็จะสั่งให้ไปประมวลผลคำสั่ง turnRight ซึ่งส่งคืนค่าจริงให้กับฟังก์ชัน IF-NOT ถ้าเป็นเท็จก็จะไปประมวลผลคำสั่ง smellRight และส่งคืนค่าจากคำสั่งนี้ให้กับฟังก์ชัน IF-NOT ในที่นี้ถ้าค่าส่งคืนของอาร์กิวเมนต์ทั้ง 2 ของฟังก์ชัน IF-AND คือ forward และ IF-NOT เป็นจริงทั้งคู่ ก็จะไปประมวลผลอาร์กิวเมนต์ที่ 3 คือ turnleft ไม่เช่นนั้นจะไปประมวลผลอาร์กิวเมนต์ที่ 4 ต่อไปเมื่อประมวลผลเสร็จแล้วถือเป็นการประมวลผล 1 รอบ และเริ่มต้นในรอบใหม่ การประมวลผลแต่ละโปรแกรมจะสิ้นสุดลงเมื่อถึงเกณฑ์ของเงื่อนไขในการวนซ้ำที่กำหนด ดังจะกล่าวต่อไป

4.2.3 การดำเนินการกำหนดการเชิงพันธุกรรม

ส่วนนี้อธิบายถึงกระบวนการค้นหาคำตอบโดยกำหนดการเชิงพันธุกรรม แบ่งเป็นขั้นตอนหลัก คือ การสร้างประชากรผลเฉลยเริ่มต้น การประเมินค่าความเหมาะของผลเฉลย การสร้างประชากรของผลเฉลยรุ่นใหม่ และการหาคำตอบ ดังที่กล่าวในบทที่ 2 ซึ่งมีรายละเอียดดังนี้

4.2.3.1 การสร้างประชากรผลเฉลยเริ่มต้น

ทำการสุ่มสร้างประชากรของโปรแกรมควบคุม ซึ่งจะเรียกว่าเป็นประชากรของผลเฉลยขึ้นมาเป็นจำนวน 1,000 ตัว แต่ละตัวประกอบด้วยฟังก์ชัน และเทอร์มินอลที่แตกต่างกันเนื่องมาจากการสุ่ม โดยขนาดเริ่มต้นของผลเฉลยที่สุ่มสร้างมาแต่ละตัวมีจำนวนฟังก์ชัน และเทอร์มินอลไม่เกิน 110 คำสั่ง หรือกล่าวอีกอย่างได้ว่าต้นไม้ที่แทนผลเฉลยเริ่มต้นแต่ละต้นมีจำนวนโหนดไม่เกิน 110 โหนด (Node) ไม่จำกัดระดับความสูงและรูปร่างของต้นไม้ที่แทนผลเฉลย

4.2.3.2 การประเมินค่าความเหมาะของผลเฉลย

แต่ละรุ่นของกระบวนการวิวัฒนาการเพื่อหาคำตอบโปรแกรม หรือผลเฉลยแต่ละตัวจะถูกนำไปประเมินประสิทธิภาพในการแก้ปัญหาที่กำหนด โดยนำผลเฉลยแต่ละตัวไปประมวลผลกับสภาพแวดล้อมที่กำหนดทีละตัว ผลเฉลยแต่ละตัวจะถูกวนรอบปฏิบัติซ้ำไปซ้ำมา การประมวลผลแต่ละตัวจะสิ้นสุดลงเมื่อถึงเกณฑ์ของเงื่อนไขในการวนซ้ำที่กำหนด ซึ่งมีดังนี้

- เมื่อมีการประมวลผลเทอร์มินอล หรือคำสั่งต่างๆ ในผลเฉลยเกิน 5,000 คำสั่ง
- เมื่อผลเฉลยนั้นสามารถควบคุมหุ่นยนต์ได้ประสบความสำเร็จ คือให้หุ่นยนต์เคลื่อนที่ไปพบเป้าหมายได้

เมื่อสิ้นสุดการประมวลผลแต่ละผลเฉลยแล้ว นำมาหาค่าความเหมาะเพื่อหาว่าผลเฉลยนั้นมีความประสิทธิภาพในการแก้ปัญหาเช่นไร โดยมีฟังก์ชันวัดความเหมาะ (Fitness Function) ดังสมการ

$$FITNESS = k \times finalDist. + numTermPass \qquad \ldots\ldots (1)$$

โดยที่ *k* เป็นค่าคงที่ในที่นี้ มีค่าเท่ากับ 10,000

finalDist. เป็นระยะห่างในแนวตรงระหว่างตำแหน่งของหุ่นยนต์ กับตำแหน่งของเป้าหมาย หลังจากสิ้นสุดการประมวลผลผลเฉลยนั้น

numTermPass เป็นจำนวนเทอร์มินอลที่ผลเฉลยนั้นมีการประมวลไปหลังจากสิ้นสุดการประมวลผลผลเฉลยนั้น

จะเห็นได้ว่าค่าความเหมาะที่ยิ่งน้อย ผลเฉลยยิ่งมีคุณภาพดี คือ ควบคุมให้หุ่นยนต์ให้เข้าใกล้เป้าหมายได้มากและมีจำนวนครั้งในการปฏิบัติคำสั่งที่น้อย หลังจากนั้นก็เริ่มประมวลผลผลเฉลยตัวอื่นๆ ต่อไป โดยนำหุ่นยนต์ไปเริ่มที่ตำแหน่งเริ่มต้นที่กำหนดไว้ในสภาพแวดล้อมเหมือนเดิม ทำเช่นนี้ไปจนครบทั้งหมด จะได้ค่าความเหมาะของผลเฉลยทุกตัว

4.2.3.3 การสร้างประชากรผลเฉลยรุ่นใหม่

เพื่อคัดเลือกประชากรที่มีประสิทธิภาพให้อยู่รอด และสามารถขยายพันธุ์ได้ในรุ่นต่อๆไป โดยเรียงลำดับผลเฉลยตามค่าความเหมาะแล้วคัดเลือกไว้เป็นจำนวน 10% ของจำนวนประชากรทั้งหมด ซึ่งในที่นี้คือ 100 ตัว แล้วนำผลเฉลยที่ถูกคัดเลือกเหล่านี้มาทำการสืบพันธุ์ (Reproduction) หรือคัดลอกไว้ในรุ่นถัดไป แล้วทำการสร้างประชากรในรุ่นใหม่นี้ โดยนำผลเฉลย 100 ตัวเหล่านี้มาดำเนินการทางพันธุกรรมจนมีจำนวนครบ 1,000 ตัว สำหรับในการทดลองนี้จะเลือกใช้ตัวดำเนินการแบบการไขว้เปลี่ยน เท่านั้น ไม่ใช้ตัวดำเนินการแบบการกลาย

รายละเอียดของตัวดำเนินการไขว้เปลี่ยนที่ใช้ คือ ทำการคัดเลือกผลเฉลยมา 2 ตัวแบบสุ่มจากผลเฉลย 100 ตัวในตอนแรก หลังจากนั้นเลือกตำแหน่งในการไขว้เปลี่ยนแบบสุ่ม แต่มีข้อกำหนดว่าให้เลือกตำแหน่งที่เป็นฟังก์ชันเท่านั้น จะไม่มีการไขว้เปลี่ยนที่ตำแหน่งเทอร์มินอล แล้วทำการแลกเปลี่ยนส่วนของโปรแกรมกัน โดยไม่จำกัดระดับความสูงและรูปร่างของต้นไม้ที่แทนผลเฉลย ผลจากการไขว้เปลี่ยนแต่ละครั้งจะได้ประชากรผลเฉลยใหม่อีก 2 ตัว ทำไปเช่นนี้จนกระทั่งได้จำนวนประชากรครบ 1,000 ตัว

4.2.3.4 การหาคำตอบ

กระบวนการวิวัฒนาการเพื่อหาคำตอบ คือการประเมินค่าความเหมาะของผลเฉลย และการสร้างผลเฉลยในรุ่นใหม่ จะดำเนินการไปเรื่อยๆ จนกระทั่งจะหยุดเมื่อพบเงื่อนไขแบบใดแบบหนึ่ง ดังต่อไปนี้

- จำนวนรุ่นมากที่สุดเป็น 125 รุ่น
- ผลเฉลยที่ถูกคัดเลือกเพื่อมาทำการสืบพันธุ์ 100 ตัวมีค่าความเหมาะซ้ำกันทั้งหมดเป็นเช่นนี้ไปจำนวน 5 รุ่นติดต่อกัน

เมื่อหยุดการดำเนินการแล้วผลเฉลยตัวที่มีค่าความเหมาะดีที่สุด(น้อยที่สุด) และสามารถควบคุมหุ่นยนต์ให้ไปพบเป้าหมายได้ จะถือว่าผลเฉลย หรือโปรแกรมควบคุมตัวนั้นเป็นคำตอบของปัญหาสำหรับการทดลองในครั้งนั้นๆ ในกรณีที่มีผลเฉลยที่ค่าความเหมาะที่ดีที่สุดเท่ากันหลายตัว จะถือลำดับของผลเฉลยเป็นสำคัญเพราะต้องการคำตอบเพียงตัวเดียวเท่านั้น เป็นที่น่าสังเกตว่าคำตอบที่ได้มาในการทดลองแต่ละครั้งนั้นจะมีลักษณะแตกต่างกันคือเป็นคนละคำตอบกัน สาเหตุที่เป็นเช่นนี้อันเนื่องมาจากการสร้างประชากรผลเฉลยเริ่มต้นที่แตกต่างกันเนื่องมาจากการสุ่ม

วิธีการกำหนดการเชิงพันธุกรรมที่ได้กล่าวมาเหล่านี้ เป็นวิธีที่ยังไม่มีการปรับปรุงกระบวนการวิวัฒนาการในการหาคำตอบ สำหรับสรุปรายละเอียดต่างๆ ของวิธีการกำหนดการเชิงพันธุกรรมที่ใช้ในการทดลองนี้แสดงดังตารางที่ 4.1

วัตถุประสงค์	เพื่อค้นหาโปรแกรมควบคุมหุ่นยนต์ที่สามารถควบคุมให้หุ่นยนต์เดินหลบหลีกสิ่งกีดขวางเพื่อไปหาเป้าหมายในสภาพแวดล้อมที่กำหนด
ขนาดของประชากร	1,000 โปรแกรม
จำนวนรุ่นมากที่สุดที่ใช้ในการทดลอง	125 รุ่น
จำนวนโหนดเริ่มต้นของแต่ละโปรแกรม	110 nodes
ฟังก์ชัน	IF-AND , IF-OR , IF-NOT
เทอร์มินอล	forward , turnLeft , turnRight, smellLeft , smellRight
เงื่อนไขในการวนซ้ำ ประมวลผลผลเฉลยแต่ละตัวจะสิ้นสุดเมื่อ	- เมื่อมีการประมวลผลเทอร์มินอล หรือคำสั่งต่างๆในผลเฉลยเกิน 5,000 คำสั่ง หรือ - เมื่อผลเฉลยนั้นสามารถควบคุมหุ่นยนต์ได้ประสบความสำเร็จ คือให้หุ่นยนต์เคลื่อนที่ไปพบเป้าหมายได้

ตารางที่ 4.1 แสดงรายละเอียดต่างๆของวิธีการกำหนดการเชิงพันธุกรรมที่ใช้ในการทดลอง

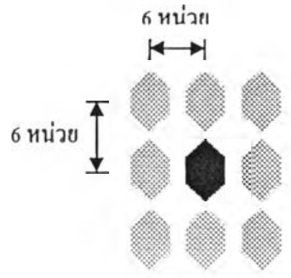
ฟังก์ชันวัดความเหมาะ	$FITNESS = k \times finalDist. + numTermPass$ *k* เป็นค่าคงที่มีค่า 10,000 *finalDist.* เป็นระยะห่างในแนวตรงระหว่างตำแหน่งของหุ่นยนต์ กับตำแหน่งของเป้าหมาย หลังจากสิ้นสุดการประมวลผลผลเฉลยนั้น *numTermPass* เป็นจำนวนเทอร์มินอลที่ผลเฉลยนั้นมีการประมวลผลไปหลังจากสิ้นสุดการประมวลผลผลเฉลยนั้น ค่าความเหมาะจากสมการนี้ ยิ่งน้อย ยิ่งแสดงประสิทธิภาพของผลเฉลยในการแก้ปัญหาที่ยิ่งดี
ตัวดำเนินการทางพันธุกรรมที่ใช้	จำนวนผลเฉลยที่สร้างขึ้นใหม่โดยใช้ตัวดำเนินการนี้
- การสืบพันธุ์	- 10 % ของจำนวนประชากรทั้งหมด (100 ตัว)
- การไขว้เปลี่ยน	- 90 % ของจำนวนประชากรทั้งหมด (900 ตัว)
- การกลาย	- ไม่ใช้ตัวดำเนินการกลาย
วิธีการสืบพันธุ์	เรียงลำดับผลเฉลยทั้ง 1,000 ตัวตามค่าความเหมาะจากน้อยไปมาก แล้วคัดเลือก 100 ตัว ที่มีค่าความเหมาะใน 100 อันดับแรก ให้อยู่รุ่นต่อไป
วิธีการไขว้เปลี่ยน	- คัดเลือกผลเฉลยที่ได้ผ่านการสืบพันธุ์ มาดำเนินการไขว้เปลี่ยนทีละ 2 ตัว แต่ละตัวมีโอกาสถูกคัดเลือกเท่าๆ กัน - ทำการสุ่มที่จะเกิดการไขว้เปลี่ยน และยอมให้มีการไขว้เปลี่ยนเฉพาะตำแหน่งที่เป็นฟังก์ชันเท่านั้น - การไขว้เปลี่ยนแต่ละครั้งได้ประชากรผลเฉลยเพิ่มขึ้นอีก 2 ตัว
เงื่อนไขในกระบวนการวิวัฒนาการเพื่อหาคำตอบจะสิ้นสุดเมื่อ	- จำนวนรุ่นเกิน 125 รุ่น หรือ - ผลเฉลยที่ถูกคัดเลือกเพื่อมาทำการสืบพันธุ์ 100 ตัว มีค่าความเหมาะซ้ำกันทั้งหมด เป็นเช่นนี้ไปเป็นจำนวน 5 รุ่นติดต่อกัน

ตารางที่ 4.1 (ต่อ) แสดงรายละเอียดต่างๆของวิธีการกำหนดการเชิงพันธุกรรมที่ใช้ในการทดลอง

4.3 <u>การทดสอบความทนทานของคำตอบ</u>

ดังได้กล่าวมาแล้วว่า การนำวิธีกำหนดการเชิงพันธุกรรมมาใช้แก้ปัญหา หรือหาคำตอบของปัญหาการควบคุมดังเช่นการทดลองนี้ พบว่าถึงแม้ว่าคำตอบที่ได้นั้นสามารถควบคุมให้หุ่นยนต์ทำงานเดินหลบหลีกสิ่งกีดขวางไปหาเป้าหมายได้ในสภาพแวดล้อมที่หุ่นยนต์เรียนรู้มา แต่เมื่อสภาพแวดล้อมเปลี่ยนแปลงไป เช่นมีการขยับสิ่งกีดขวางบางชิ้นไปจากตำแหน่งเดิม คำตอบนั้นก็อาจจะไม่ประสบความสำเร็จได้ ในการทดลองนี้ได้นิยามความทนทานของคำตอบ หมายถึง ความสามารถในการนำคำตอบนั้นไปทำงานในสภาพแวดล้อมที่ไม่เคยพบหรือไม่เคยเรียนรู้มาก่อน หรือเป็นโอกาสที่คำตอบนั้นสามารถควบคุมหุ่นยนต์ไปหาเป้าหมายได้สำเร็จในสภาพแวดล้อมที่ไม่เคยพบมาก่อน และได้ออกแบบวิธีทดสอบความทนทานของคำตอบ ดังนี้

สร้างสภาพแวดล้อมที่ใช้ในการทดสอบความทนทานขึ้นมาเป็นจำนวนมาก โดยนำสภาพแวดล้อมที่เป็นโจทย์ปัญหาซึ่งได้เรียนรู้มา มาทำการดัดแปลงไปโดยสุ่มเปลี่ยนตำแหน่งของสิ่งกีดขวางบางชิ้นในสภาพแวดล้อมนั้นไปเป็นระยะทาง 6 หน่วย คิดเป็นระยะประมาณ 40% ของขนาดของสิ่งกีดขวาง จากตำแหน่งเดิมของมันในทิศทางใดก็ได้ขึ้นอยู่กับการสุ่มดังรูปที่ 4.7 สำหรับจำนวนสิ่งกีดขวางที่ถูกขยับตำแหน่งไปจากเดิมแทนด้วยตัวแปร d_{Test} (Test Disturbance) ซึ่งคิดเป็นร้อยละของจำนวนสิ่งกีดขวางทั้งหมด โดยที่ตำแหน่งเริ่มต้นของหุ่นยนต์และตำแหน่งเป้าหมายยังอยู่ตำแหน่งเดิม



รูปที่ 4.7 แสดงทิศทางการเปลี่ยนตำแหน่งของสิ่งกีดขวางชิ้นหนึ่งๆ ที่เป็นไปได้

สภาพแวดล้อมที่สร้างขึ้นมาจะมีจำนวน 10,000 สภาพแวดล้อม แบ่งออกเป็น 10 กลุ่มซึ่งแต่ละกลุ่มแตกต่างกันตามค่า d_{Test} หรือค่าเปอร์เซนต์สิ่งกีดขวางที่ถูกขยับตำแหน่งไป มีจำนวนกลุ่มละ 1,000 สภาพแวดล้อม ค่า d_{Test} ของแต่ละกลุ่มมีค่าเป็น 10% , 20%, 30% ,..... จนถึง 100% นั่นคือขยับสิ่งกีดขวางไปทุกชิ้น

หลังจากนั้นนำคำตอบที่ได้มาทดสอบควบคุมหุ่นยนต์ในสภาพแวดล้อมแต่ละกลุ่มเหล่านี้ โดยที่เงื่อนไขในการสิ้นสุดการวนซ้ำเพื่อประมวลผลคำตอบในแต่ละสภาพแวดล้อม คือ เมื่อประมวลผลเทอร์มินอลเกิน 5,000 คำสั่ง หรือพบเป้าหมาย ซึ่งเหมือนกับเงื่อนไขในระหว่างการประมวลผลเพื่อหาคำตอบ โดยค่าความทนทานของคำตอบจะนับจากจำนวนสภาพแวดล้อมที่คำตอบนั้นสามารถควบคุมหุ่นยนต์ให้พบเป้าหมายได้ประสบความสำเร็จ คิดเป็นร้อยละของจำนวนสภาพแวดล้อมทดสอบทั้งหมดในกลุ่มนั้นๆ แทนด้วยตัวแปร $R(d_{Test})$ คือความทนทานของคำตอบนั้นในสภาพแวดล้อมทดสอบที่มีค่าเปอร์เซนต์สิ่งกีดขวางที่ถูกขยับตำแหน่งไปเป็น d_{Test} %

วิธีการทดสอบความทนทานและสภาพแวดล้อมที่ใช้ในการทดสอบเหล่านี้จะใช้ในการทดลองเพื่อทดสอบความทนทานในวิทยานิพนธ์ฉบับนี้ เพื่อใช้เปรียบเทียบค่าความทนทานของคำตอบ ทั้งก่อนและหลังการปรับปรุงกระบวนการวิวัฒนาการในการหาคำตอบ

4.4 ผลการทดสอบความทนทานของคำตอบ

การทดลองหาคำตอบโดยใช้กำหนดการเชิงพันธุกรรมดังที่กล่าวมา เป็นวิธีแบบปกติที่ยังไม่ได้มีการปรับปรุงกระบวนการวิวัฒนาการเพื่อหาคำตอบ ผลการทดสอบความทนทานที่ได้นี้จะนำไปเปรียบเทียบกับวิธีหลังจากการปรับปรุงกระบวนการวิวัฒนาการเพื่อหาคำตอบ เพื่อดูว่าความทนทานของคำตอบเพิ่มขึ้นหรือไม่และเช่นไร

ดังที่ได้กล่าวในหัวข้อ 4.2.3.4 แล้วว่า ในการหาคำตอบโดยกำหนดการเชิงพันธุกรรม คำตอบที่ได้มาในการทดลองแต่ละครั้งจะแตกต่างกันอันเนื่องมาประชากรที่สุ่มสร้างในตอนเริ่มต้นแตกต่างกัน ดังนั้นเพื่อเป็นการเพิ่มความถูกต้องให้กับการทดลอง จะทำการทดลองเพื่อหาคำตอบทั้งหมด 20 ครั้ง ได้โปรแกรมควบคุมที่ประสบความสำเร็จที่เป็นคำตอบมา 20 ตัว (การทดลองละ 1 ตัว) แล้วนำคำตอบแต่ละตัวมาทดสอบความทนทานตามวิธีที่กล่าวมา หลังจากนั้นนำมาหาค่าเฉลี่ยแล้วนำมาวาดกราฟแสดงความทนทานเฉลี่ยของคำตอบ ในวิธีก่อนปรับปรุงกระบวนการวิวัฒนาการ

ให้ $R(i,d_{Test})$ คือ ความทนทานของคำตอบจากการทดลองครั้งที่ i ในสภาพแวดล้อมทดสอบที่มีค่าเปอร์เซนต์สิ่งกีดขวางที่ถูกขยับตำแหน่งไปเป็น d_{Test} % เป็นจำนวน 1,000 สภาพแวดล้อม

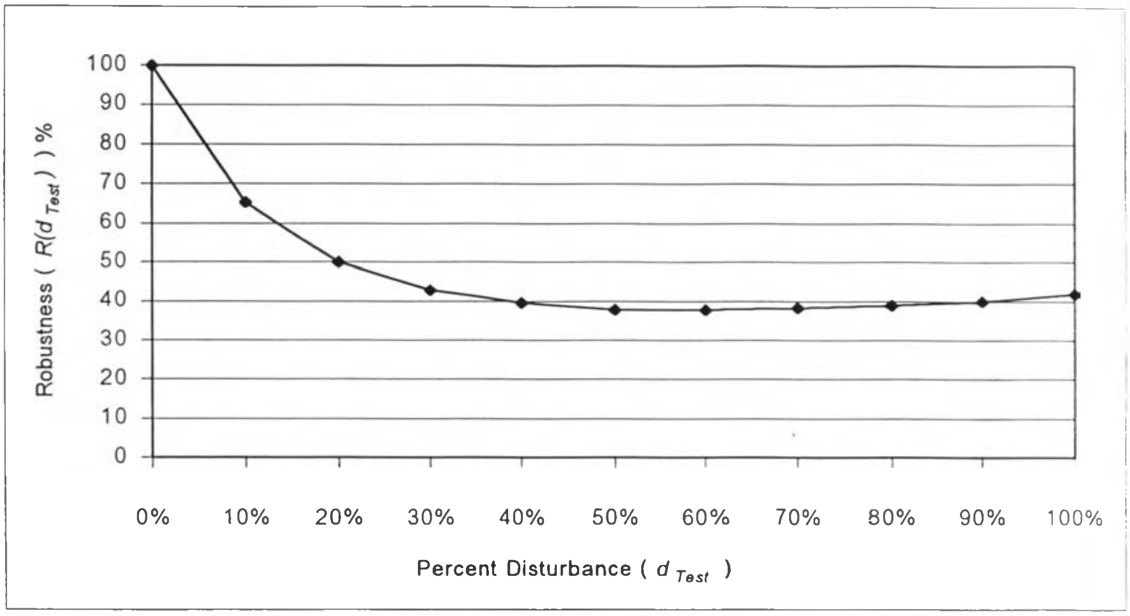
ดังนั้น $R(d_{Test})$ ซึ่งคือความทนทานของคำตอบโดยเฉลี่ยในสภาพแวดล้อมทดสอบที่มีค่าเปอร์เซนต์สิ่งกีดขวางที่ถูกขยับตำแหน่งไปเป็น d_{Test} % มีค่าดังสมการ

$$R(d_{Test}) = \frac{\sum_{i=1}^{20} R(i, d_{Test})}{20} \qquad \text{......(2)}$$

d_{Test}	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
$R(d_{Test})$%	65.19	50.11	42.90	39.55	38.00	38.02	38.45	39.11	40.11	41.93

ตารางที่ 4.2 แสดงค่าความทนทานของคำตอบ (%) ในสภาพแวดล้อมต่างๆ กัน

ตารางที่ 4.2 เป็นผลการทดสอบความทนทานของคำตอบที่ได้ โดยแสดงค่าความทนทานของคำตอบ $R(d_{Test})$ ในสภาพแวดล้อมที่สิ่งกีดขวางถูกขยับตำแหน่งไปเป็นจำนวนต่างๆ กัน d_{Test} ซึ่งสามารถนำมาวาดเป็นกราฟแสดงความสัมพันธ์ได้ในดังรูปที่ 4.8 เป็นกราฟแสดงความทนทานเฉลี่ยของคำตอบที่ได้จากวิธีการกำหนดการเชิงพันธุกรรมสำหรับปัญหาการควบคุมหุ่นยนต์ให้เดินหลบหลบหลีกสิ่งกีดขวางเพื่อไปหาเป้าหมายในสภาพแวดล้อมดังรูปที่ 4.1 โดยใช้กระบวนการวิวัฒนาการเพื่อหาคำตอบแบบปกติ ยังไม่มีการปรับปรุงกระบวนการใดๆ



รูปที่ 4.8 กราฟแสดงความทนทานของคำตอบ (%) ก่อนการปรับปรุงกระบวนการวิวัฒนาการ

แกนนอนแสดงค่า d_{Test} ซึ่งเป็นกลุ่มของสภาพแวดล้อมที่ใช้ทดสอบที่มีค่าเปอร์เซนต์สิ่งกีดขวางที่ถูกขยับตำแหน่งไปเป็น d_{Test} % ส่วนแกนตั้งแสดงค่าความทนทาน หรือโอกาสที่คำตอบนั้นสามารถควบคุมหุ่นยนต์ไปหาเป้าหมายได้สำเร็จในสภาพแวดล้อมนั้นๆ

จากกราฟสังเกตได้ว่า ความทนทานมีค่าค่อนข้างต่ำ เช่นที่ d_{Test} = 30% มีค่า $R(d_{Test})$ = 43% นั่นคือ เมื่อนำคำตอบไปทดสอบควบคุมหุ่นยนต์ให้เดินในสภาพแวดล้อมที่สิ่งกีดขวางถูกขยับตำแหน่ง ให้แตกต่างจากเดิมเป็นจำนวน 30% จำนวน 1,000 สภาพแวดล้อม พบว่าโปรแกรมคำตอบนั้นควบคุมหุ่นยนต์ให้ไปหาเป้าหมายได้สำเร็จโดยเฉลี่ยเพียง 430 สภาพแวดล้อม เท่านั้น คิดเป็นเปอร์เซนต์ที่สำเร็จประมาณ 43% ซึ่งถือว่าต่ำมาก

สำหรับที่ค่า d_{Test} เป็น 0% นั่นคือสภาพแวดล้อมที่ใช้ทดสอบเป็นสภาพแวดล้อมเดียวกับสภาพแวดล้อมที่ได้เรียนรู้มา ไม่มีการเปลี่ยนแปลงใดๆ ค่าความทนทานของคำตอบที่จุดนี้จึงมีค่าเป็น 100%

4.5 <u>สรุปท้ายบท</u>

ในบทนี้ได้แสดงถึงวิธีการออกแบบการทดลองเพื่อหาคำตอบโดยวิธีกำหนดการเชิงพันธุกรรมในปัญหาการควบคุมหุ่นยนต์เดินหลบหลีกสิ่งกีดขวางเพื่อไปหาเป้าหมาย รายละเอียดของการทดลอง และวิธีทดสอบความทนทานของคำตอบ รวมทั้งผลการทดสอบความทนทานของคำตอบสำหรับวิธีก่อนปรับปรุงกระบวนการวิวัฒนาการ เพื่อใช้เปรียบเทียบกับความทนทานของคำตอบหลังจากปรับปรุงกระบวนการวิวัฒนาการเพื่อหาคำตอบ ดังจะกล่าวในบทต่อไป