การวางแผนเส้นทางสำหรับหุ่นยนต์โดยใช้เจเนติกอัลกอริธึมที่มีฐานความรู้

นางสาวดวงจันทร์ อูเอะตะ 46010235
นายธีวินท์ กิตติทนนท์ชัย 46010326
ผศ. เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการพัฒนาวิธีวางแผนเส้นทางให้หุ่นยนต์อัตโนมัติสามารถเคลื่อนที่จากจุดเริ่มต้นไปยังปลายทางโดยไม่ชนกับอุปสรรคและใช้ระยะทางน้อยที่สุด โดยใช้เจเนติกอัลกอริธึมซึ่งเป็นวิธีการหาคำตอบแบบสุ่มที่เลียนแบบมาจากกระบวนการวิวัฒนาการในธรรมชาติ ปริญญานิพนธ์ฉบับนี้เป็นการทำงานร่วมกันระหว่างเจเนติกอัลกอริธึมกับการเก็บข้อมูลที่เกี่ยวข้องกับการหาคำตอบเป็นฐานความรู้เพื่อสร้างเงื่อนไขเพิ่มเติมในขั้นตอนการหาคำตอบของเจเนติกอัลกอริธึมให้มีความเหมาะสมกับการหาเส้นทางในสภาพแวดล้อมที่มีอุปสรรคโดยเฉพาะด้วยการใช้ประโยชน์จากข้อมูลในฐานความรู้ให้มากขึ้น ลดทอนขั้นตอนที่ไม่จำเป็นในการหาคำตอบ เนื่องจากปริญญานิพนธ์นี้ทำการเลือกเครื่องมือที่ใช้ในขั้นตอนต่างๆให้สอดคล้องกับรูปแบบการทำงานที่เปลี่ยนแปลงไป และกำหนดค่าของตัวแปรในระบบให้เอื้อต่อการหาผลลัพธ์มากที่สุด ทำให้ในแต่ละขั้นตอนการทำงานของเจเนติกอัลกอริธึมจะช่วยพัฒนาคุณภาพของคำตอบมากขึ้น ใช้เวลาในการทำงานลดลง เส้นทางซึ่งเป็นผลลัพธ์ที่ได้จากระบบมีความแม่นยำและน่าเชื่อถือมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot

Miss Duangchan Ueta 46010235

Mr. Theewin Kittithanonchai 46010326

Mr. Kietikul Jiaranaitanakij Advisor

Academic Year 2006

ABSTRACT

The objective of this thesis is to develop algorithm to find the best moving path of a mobile robot using genetic algorithm, a class of stochastic search algorithm imitated biological evolution. This thesis combines genetic algorithm with the knowledge from storing released data for creating additional condition to make this algorithm be suitable for multi-obstacles environment. By using advantage of information in the knowledge base, we can reduce nonessential procedure. Since this thesis chooses tool that matches the change of the work pattern and adjusts the value of variables for best searching, we can improve the quality of the solution and reduce the computation time. Moreover, the solution path is more correct and more reliable.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยการให้คำปรึกษา คำแนะนำ และความเมตตาจาก ผศ. เกียรติกูล เจียรนัยธนะกิจ ผู้ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์ ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์ของท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้กับข้าพเจ้า

ขอขอบคุณทุกคนที่ห้องโปรเจค ที่คอยช่วยเหลือ เป็นกำลังใจ และร่วมทุกข์ร่วมสุขคล้ายๆกัน และขอบคุณห้อง ECC-704, ห้องโปรเจค MML และชุมนุม C-Square ที่มีบรรยากาศดีๆ และเป็นสถานที่ให้ทำงานตลอดมา

ขอขอบคุณทุกคนในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เคยได้พบได้คุย ได้รู้จัก และได้ให้ความช่วยเหลือ ช่วยกล่อมเกลาให้ข้าพเจ้าได้เป็นคนที่มีคุณค่ามากขึ้น

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนข้าพเจ้าอย่างเต็มที่ ทำให้ข้าพเจ้าสามารถทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี

คุณค่าและประโยชน์อันพึงได้รับปริญญานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ดวงจันทร์ อูเอะตะ

ธีวินท์ กิตติทนนท์ชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ

1.1 ความสำคัญและที่มาของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีการดำเนินการ 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

1.6 ส่วนประกอบของปริญญานิพนธ์ 3

บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

2.1 ความหมายของหุ่นยนต์อัตโนมัติ 5

2.2 ทฤษฎีทางด้านเจเนติกอัลกอริธึม 5

2.2.1 วิธีการคัดเลือกโครโมโซม 6

2.2.1.1 วิธีคัดเลือกแบบวงล้อรูเล็ต 6

2.2.1.2 วิธีคัดเลือกแบบจัดอันดับ 7

2.2.1.3 วิธีคัดเลือกแบบการแข่งขัน 8

2.2.2 กระบวนการทางพันธุกรรม 8

2.2.2.1 การสับเปลี่ยนชิ้นส่วน 8

2.2.2.2 การกลายพันธุ์ 10

2.2.3 กระบวนการของเจเนติกอัลกอริธึม 12

บทที่ 3 การทำงานของระบบตามเอกสารอ้างอิง

3.1 บทนำ 15

3.2 การกำหนดปัญหา 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3 การแทนค่าปัญหา 15
3.3.1 การกำหนดคุณสมบัติของสภาพแวดล้อม 16
3.3.2 การกำหนดคุณสมบัติของโครโมโซม 17
3.4 การประเมินค่า 17
3.5 การเลือกโครโมโซมต้นแบบ 18
3.6 กระบวนการทางพันธุกรรม 19
3.6.1 กระบวนการมาตรฐาน 19
3.6.2 กระบวนการเฉพาะทาง 20
3.7 การคัดเลือกโครโมโซมไปเป็นประชากรรุ่นต่อไป 21
3.8 การกำหนดเงื่อนไขเพื่อหยุดการหาคำตอบ 21
3.9 แผนผังการทำงานของเจเนติกอัลกอริธึมแบบมีฐานความรู้ 22
บทที่ 4 การออกแบบระบบส่วนที่โครงงานพัฒนาขึ้น
4.1 การออกแบบวิธีการการหาคำตอบ 24
4.1.1 การออกแบบวิธีการแทนค่าปัญหา 24
4.1.1.1 การออกแบบคุณสมบัติของสภาพแวดล้อม 24
4.1.1.2 การออกแบบคุณสมบัติของโครโมโซม 25
4.1.2 การออกแบบวิธีประเมินค่า 26
4.1.3 การออกแบบวิธีการเลือกโครโมโซมต้นแบบ 27
4.1.4 การออกแบบกระบวนการทางพันธุกรรม 27
4.1.4.1 กระบวนการมาตรฐาน 28
4.1.4.2 กระบวนการเฉพาะทาง 29
4.1.5 การออกแบบวิธีการคัดเลือกโครโมโซมไปเป็นประชากรรุ่นต่อไป 33
4.1.6 การออกแบบเงื่อนไขเพื่อหยุดการหาคำตอบ 34
4.1.7 การออกแบบแผนผังการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้ 34
4.2 การออกแบบโครงสร้างการทำงาน 36
4.2.1 คลาส Grid 36
4.2.2 คลาส Ground 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.2.3 คลาส Obstacle 38

4.2.4 คลาส Genome 39

4.2.5 คลาส Population 41

บทที่ 5 ผลการทดลอง

5.1 การหาค่าที่เหมาะสมของตัวแปรประเภทต่างๆ 44

5.1.1 หาค่าความเป็นไปได้ในการทำกระบวนการทางพันธุกรรมต่างๆที่เหมาะสม 44

5.1.2 หาความยาวมากที่สุดของโครโมโซมที่เหมาะสม 47

5.1.3 หาจำนวนประชากรที่เหมาะสม 47

5.2 การเปรียบเทียบประสิทธิภาพในการหาคำตอบระหว่างระบบเดิม กับระบบที่โครงงานพัฒนาขึ้น 48

5.2.1 การวางแผนเส้นทางในกรณีที่อุปสรรคเป็นรูปตัวยู 49

5.2.2 การวางแผนเส้นทางในกรณีที่สภาพแวดล้อมมีลักษณะซับซ้อนแบบที่ 1 51

5.2.3 การวางแผนเส้นทางในกรณีที่สภาพแวดล้อมมีลักษณะซับซ้อนแบบที่ 2 52

บทที่ 6 บทวิจารณ์และสรุป

6.1 บทสรุป 54

6.2 วิจารณ์สิ่งที่ได้จากโครงงาน 54

6.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข 55

6.4 แนวทางการพัฒนาต่อ 55

บรรณานุกรม 56

เอกสารอ้างอิง 57

ภาคผนวก

ภาคผนวก ก – เอกสารต้นแบบ 59

ภาคผนวก ข – การใช้งานโปรแกรม Knowledge Based Genetic Algorithm For Path Planning ที่โครงงานพัฒนาขึ้น 65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
4.1	แสดง Private Member Datas ของคลาส Grid	36
4.2	แสดง Public Member Functions ของคลาส Grid	36
4.3	แสดง Private Member Datas ของคลาส Ground	37
4.4	แสดง Public Member Functions ของคลาส Ground	37
4.5	แสดง Private Member Datas ของคลาส Obstacle	38
4.6	แสดง Public Member Functions ของคลาส Obstacle	38
4.7	แสดง Private Member Datas ของคลาส Genome	39
4.8	แสดง Private Member Functions ของคลาส Genome	39
4.9	แสดง Public Member Functions ของคลาส Genome	39
4.10	แสดงฟังก์ชั่นที่เกี่ยวข้องกับคลาส Genome	41
4.11	แสดง Private Member Datas ของคลาส Population	41
4.12	แสดง Publie Member Functions ของคลาส Population	42
5.1	แสดงผลลัพธ์ที่ได้จากกำหนดค่าต่างๆให้กับความเป็นไปได้ในการทำกระบวนการต่างๆ	45
5.2	แสดงผลลัพธ์ที่ได้จากกำหนดค่าต่างๆให้กับความยาวมากที่สุดของโครโมโซม	47
5.3	แสดงผลลัพธ์ที่ได้จากกำหนดค่าต่างๆให้กับจำนวนประชากร	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	แสดงตัวอย่างหุ่นยนต์อัตโนมัติ	5
2.2	แสดงแผนภาพสัดส่วนค่าความเหมาะสมของโครโมโซมในวงล้อรูเล็ต	7
2.3	แสดงโอกาสที่แต่ละโครโมโซมจะถูกเลือกด้วยวิธีการคัดเลือกแบบจัดอันดับ	7
2.4	ภาพจำลองการแลกเปลี่ยนชิ้นส่วนโดยระบุเพียงตำแหน่งเดียว	9
2.5	ภาพจำลองการแลกเปลี่ยนชิ้นส่วนโดยระบุ 2 ตำแหน่ง	9
2.6	ภาพจำลองการแลกเปลี่ยนชิ้นส่วนแบบสุ่มสร้างขึ้นใหม่	9
2.7	ภาพจำลองการแลกเปลี่ยนชิ้นส่วนแบบใช้กระบวนการทางคณิตศาสตร์	10
2.8	ภาพจำลองการแลกเปลี่ยนชิ้นส่วนแบบสลับลำดับ	10
2.9	ภาพจำลองการกลายพันธุ์แบบกลับค่า	11
2.10	ภาพจำลองการกลายพันธุ์แบบเปลี่ยนลำดับ	11
2.11	ภาพจำลองการกลายพันธุ์แบบเปลี่ยนค่า	11
2.12	แสดงขั้นตอนการทำงานของเจเนติกอัลกอริธึม	13
3.1	แสดงสภาพแวดล้อมและการแทนเส้นทางด้วยช่องหมายเลขต่างๆบนตาราง	16
3.2	แสดงตัวอย่างของโครโมโซม	17
3.3	แสดงตัวอย่างการปรับปรุงโครโมโซมด้วยกระบวนการทางพันธุกรรม และกระบวนการเฉพาะทาง	20
3.4	โครงสร้างการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้ เพื่อวางแผนเส้นทางให้กับหุ่นยนต์อัตโนมัติ	23
4.1	แสดงวิธีการกำหนดขนาดของอุปสรรคที่จะใช้บนตาราง	25
4.2	แสดงการทำงานของ Crossover Operator	28
4.3	แสดงการทำงานของ Mutation Operator	29
4.4	แสดงการทำงานของ Swap Operator	29
4.5	แสดงการค้นหาเส้นทางในขอบเขตที่มีขนาดเท่ากับขนาดของอุปสรรคบวก เพิ่มอีก 4 ช่อง	30
4.6	แสดงการค้นหาเส้นทางในขอบเขตที่มีขนาดเท่ากับความกว้างและยาวของตำแหน่ง ยีนส์ต้นทางและยีนส์ปลายทาง	31
4.7	แสดงการทำงานของ Linerepair Operator	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.8	แสดงการทำงานของ Delete Operator	32
4.9	แสดงการทำงานของ Improve Operator	33
4.10	แสดงการสร้างประชากรรุ่นใหม่ด้วยวิธีคัดเลือกแบบจัดอันดับ	33
4.11	แสดงขั้นตอนการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้ในการวางแผนเส้นทางที่โครงงานพัฒนาขึ้น	35
5.1	แสดงสภาพแวดล้อมที่ใช้ในการหาค่าตัวแปรที่เหมาะสม	44
5.2	แสดงการหาเส้นทางด้วยระบบเดิมในสภาพแวดล้อมที่มีอุปสรรคเป็นรูปตัวยู	50
5.3	แสดงการหาเส้นทางด้วยระบบที่โครงงานพัฒนาขึ้นในสภาพแวดล้อมที่มีอุปสรรคเป็นรูปตัวยู	50
5.4	แสดงการหาเส้นทางด้วยระบบเดิมในสภาพแวดล้อมที่ซับซ้อนแบบที่ 1	51
5.5	แสดงการหาเส้นทางด้วยระบบที่โครงงานพัฒนาขึ้นในสภาพแวดล้อมที่ซับซ้อนแบบที่ 1	52
5.6	แสดงการหาเส้นทางด้วยระบบเดิมในสภาพแวดล้อมที่ซับซ้อนแบบที่ 2	53
5.7	แสดงการหาเส้นทางด้วยระบบที่โครงงานพัฒนาขึ้นในสภาพแวดล้อมที่ซับซ้อนแบบที่ 2	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ปัญหาเกี่ยวกับการค้นหาเส้นทางมักจะเป็นปัญหาที่ต้องการคำตอบที่ให้ผลลัพธ์ดีที่สุดภายใต้ขอบเขตที่กำหนดให้ (Optimization Problem) เช่น ต้องเป็นเส้นทางที่ไม่ชนกับสิ่งกีดขวางใดๆตั้งแต่ต้นทางจนถึงปลายทาง และมีระยะทางสั้นที่สุดในสภาพแวดล้อมที่กำหนดให้ เป็นต้น ซึ่งมีผู้คิดวิธีแก้ปัญหาเช่นนี้ไว้แล้วหลากหลายวิธี แต่ละวิธีจะมีความเหมาะสมกับปัญหาแตกต่างกันไป บางวิธีสามารถหาคำตอบที่ให้ผลลัพธ์ที่ยอมรับได้ และใช้เวลาในการหาคำตอบพอสมควร เช่น global C-space methods บางวิธีสามารถหาคำตอบที่ให้ผลลัพธ์ดีที่สุดได้ทุกครั้งแต่ใช้เวลาในการหาคำตอบนานมาก เช่น A* search

การแก้ปัญหาเกี่ยวกับการค้นหาเส้นทางด้วยเจเนติกอัลกอริธึม (Genetic Algorithm) จึงเป็นที่นิยมมากขึ้นเนื่องจากสามารถหาช่วงของคำตอบที่มีประสิทธิภาพมากกว่าส่วนอื่นๆได้ในเวลาอันรวดเร็ว และสามารถแก้ปัญหาที่ต้องการคำตอบเกือบดีที่สุดหรือแบบดีที่สุด (Optimal Solution) ได้*แน่นอน โดยมีขั้นตอนการทำงานเลียนแบบหลักวิวัฒนาการของธรรมชาติ* คือ สิ่งมีชีวิตที่แข็งแรงมากที่สุดในสิ่งแวดล้อมเท่านั้นจึงจะเป็นผู้มีชีวิตรอด เติบโต และดำรงเผ่าพันธุ์ได้ต่อไป การค้นหาเส้นทางด้วยเจเนติกอัลกอริธึมเป็นการหาคำตอบจากการคำนวณที่ซับซ้อน และสามารถเลี่ยงปัญหาการได้คำตอบที่ดีในช่วงแคบๆ (Local Optimum) ได้

การใช้เจเนติกอัลกอริธึมโดยปกตินั้นแทบไม่ต่างจากการใช้วิธีหาคำตอบแบบไร้ทิศทาง (Blind Search) เนื่องจากเป็นวิธีการหาคำตอบจากทางเลือกทั้งหมดโดยไม่มีคำแนะนำ ไม่มีกฎเกณฑ์ และจะใช้ทางเลือกที่สามารถแก้ปัญหาได้ดีที่สุด ทำให้การหาคำตอบด้วยวิธีเหล่านี้มีข้อจำกัดในทางปฏิบัติ คือ ใช้เวลาและหน่วยเก็บข้อมูลจำนวนมาก จึงไม่นิยมใช้กับการแก้ปัญหาที่มีขนาดใหญ่ แต่การหาคำตอบที่ใช้ฐานความรู้ (Heuristic Search Approaches) จะพยายามหาหลักเกณฑ์เพื่อจะนำมาใช้ในขบวนการค้นหาและลดปริมาณการคำนวณลง ในทางปฏิบัติการค้นหาแบบนี้จะเร็วและคุ้มค่ากว่าการหาคำตอบแบบไร้ทิศทางแน่นอน และวิธีการแก้ปัญหาที่ได้จะมีประสิทธิภาพ*ใกล้เคียงกับทางแก้ปัญหาที่ดีที่สุดมากกว่า*

ในรายงานฉบับนี้จึงเป็นการหาเส้นทางโดยใช้เจเนติกอัลกอริธึมที่มีฐานความรู้ (Knowledge Based Genetic Algorithm) ที่แทนค่าเส้นทางด้วยการผสมผสานรูปแบบของตาราง (Grid) และคู่ลำดับ (Co-ordinate) ทำให้โครโมโซม (Chromosome) มีโครงสร้างไม่ซับซ้อน การสร้างกระบวนการทางพันธุกรรมเฉพาะสำหรับปัญหา (Problem-specific Operator) จากฐานความรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และการใช้กระบวนการทางพันธุกรรมรูปแบบต่างๆกับโครโมโซมทำได้ง่าย และมีการเลือกใช้วิธีการค้นหาในขอบเขตจำกัดที่มีประสิทธิภาพ ทำให้ระบบสามารถแก้ปัญหาได้ดียิ่งขึ้นไปอีก

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อนำความรู้ความเข้าใจที่เกี่ยวกับความหมาย หลักการ และขั้นตอนการทำงานของเจเนติกอัลกอริธึมมาใช้งานกับการหาเส้นทางที่ดีที่สุดสำหรับหุ่นยนต์เคลื่อนที่ได้
2. เพื่อศึกษาข้อดีข้อเสียของการใช้ฐานความรู้ร่วมกับเจเนติกอัลกอริธึมในการแก้ปัญหาเกี่ยวกับการหาเส้นทาง รวมถึงการพัฒนาให้สามารถแก้ปัญหาได้อย่างมีประสิทธิภาพมากยิ่งขึ้นด้วยการประยุกต์ใช้ความรู้ที่มีอยู่เดิม และการศึกษาหาความรู้ที่เกี่ยวข้องเพิ่มเติม
3. เพื่อศึกษาประโยชน์และความจำเป็นของการปรับปรุงวิธีการค้นหาในขอบเขตจำกัด (Local Search) ให้สามารถนำไปใช้ในกระบวนการทางพันธุกรรมได้เหมาะสมกับการแก้ปัญหาเกี่ยวกับการหาเส้นทางมากยิ่งขึ้น
4. เพื่อพัฒนาให้วิธีการหาเส้นทางที่ดีที่สุดสำหรับหุ่นยนต์เคลื่อนที่ได้ใช้เวลาในการคำนวณลดลง มีความรวดเร็วและเหมาะสมจะนำไปใช้หาเส้นทางในสภาพแวดล้อมจริงมากยิ่งขึ้น

1.3 ขอบเขตของโครงงาน

โครงงาน Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot เป็นการพัฒนาระบบที่มีความรู้เกี่ยวกับสภาพแวดล้อมทั้งหมดที่ต้องใช้เดินทาง และวางแผนเส้นทางให้หุ่นยนต์เคลื่อนที่จากจุดเริ่มต้นไปยังปลายทางโดยไม่ชนกับสิ่งกีดขวางและใช้ระยะทางน้อยที่สุดในสภาพแวดล้อมรูปสี่เหลี่ยมจัตุรัสที่แบ่งออกเป็นช่องขนาดเท่าๆกัน โดยทำการเขียนโปรแกรมด้วยภาษา C++ และเปรียบเทียบความสามารถในการแก้ปัญหาของระบบเดิมกับระบบใหม่ด้วยจำนวนรอบของการทำงานและคุณภาพของคำตอบ แต่ไม่สามารถเปรียบเทียบกันด้วยเวลาที่ใช้ในการหาผลลัพธ์ได้ เนื่องจากเวลาที่ใช้ในการคำนวณหาคำตอบของแต่ละระบบจะแตกต่างกันตามทรัพยากรที่ใช้ในการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีการดำเนินการ

1. ศึกษาทฤษฎีพื้นฐานเกี่ยวกับการแก้ปัญหาทั่วๆไปด้วยเจเนติกอัลกอริธึม
2. ศึกษาลักษณะของปัญหาประเภทค้นหาเส้นทางในสภาพแวดล้อมที่กำหนด
3. ศึกษาทฤษฎีการค้นหาโดยใช้ฐานความรู้
4. สร้างระบบต้นแบบ
5. วิเคราะห์ระบบต้นแบบ
6. กำหนดแนวทางการพัฒนาความสามารถในการแก้ปัญหาของระบบต้นแบบ
7. ศึกษาและวิเคราะห์วิธีการค้นหารูปแบบต่างๆ
8. สร้างสมมุติฐานในการแก้ปัญหา
9. ออกแบบระบบใหม่
10. ทดลองและบันทึกผล
11. ทดสอบความถูกต้องและปรับปรุงแก้ไขส่วนที่ผิดพลาดของระบบ
12. สรุปผลโครงงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้ความเข้าใจอย่างลึกซึ้งเกี่ยวกับกระบวนการทำงานเบื้องต้นของเจเนติกอัลกอริธึม การใช้ฐานความรู้ร่วมกับเจเนติกอัลกอริธึมเพื่อแก้ปัญหาเกี่ยวกับการหาเส้นทาง และวิธีการสร้างและประโยชน์ของการพัฒนาเจเนติกอัลกอริธึมเพื่อใช้ในการแก้ปัญหาเฉพาะด้าน (Problem – Specific Genetic Algorithm)
2. สามารถแก้ปัญหาด้วยการประยุกต์ใช้ความรู้เกี่ยวกับฐานความรู้ การค้นหาในขอบเขตจำกัด และเจเนติกอัลกอริธึมได้
3. สามารถวิเคราะห์ บอกข้อดีข้อเสียในการการแก้ปัญหาเกี่ยวกับการหาเส้นทางด้วยวิธีการแบบต่างๆได้

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บท ได้แก่

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ซึ่งประกอบด้วยความหมายของหุ่นยนต์อัตโนมัติ (Mobile Robot) และทฤษฎีเกี่ยวกับเจเนติกอัลกอริธึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 กล่าวถึงระบบและวิธีการทำงานของระบบต้นแบบตามเอกสารอ้างอิง

บทที่ 4 กล่าวถึงการออกแบบโครงงาน โดยการนำความรู้ที่ได้จากทฤษฎีในบทที่ 2 และแนวทางการทำงานของระบบในบทที่ 3 มาประยุกต์ใช้ในการพัฒนาโครงงานนี้ ซึ่งประกอบด้วยการออกแบบวิธีการหาเส้นทางและการทำงานของระบบที่โครงงานพัฒนาขึ้น

บทที่ 5 กล่าวถึงการทดลองและผลการทดลองเพื่อหาค่าของตัวแปรต่างๆที่เหมาะสมกับการทำงานของระบบมากที่สุด และเปรียบเทียบประสิทธิภาพในการหาคำตอบระหว่างระบบต้นแบบและระบบที่พัฒนาขึ้นเอง

บทที่ 6 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงงาน วิจารณ์สิ่งที่ได้รับจากโครงงาน ปัญหาอุปสรรคพร้อมแนวทางแก้ไขเป็นแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

2.1 ความหมายของหุ่นยนต์อัตโนมัติ

หุ่นยนต์อัตโนมัติ (Mobile Robot) หมายถึง เครื่องยนต์อัตโนมัติที่มีความสามารถในการเคลื่อนที่ภายใต้สภาพแวดล้อมที่กำหนด หุ่นยนต์อัตโนมัติมีความสามารถในการเคลื่อนที่ไปรอบๆ ในอาณาเขตที่เป็นไปได้และไม่ถูกยึดติดอยู่กับตำแหน่งใดตำแหน่งหนึ่ง หุ่นยนต์อัตโนมัติครอบคลุมทั้งรูปแบบที่ขับเคลื่อนบนพื้น บนพื้นน้ำหรือใต้น้ำ ในอากาศ หรือแม้แต่อวกาศ และสามารถแบ่งหุ่นยนต์อัตโนมัติออกเป็น 2 ชนิดใหญ่ๆคือ แบบควบคุมระยะไกล และแบบขับเคลื่อนอัตโนมัติ

หุ่นยนต์ควบคุมระยะไกล คือหุ่นยนต์ที่ถูกสั่งการโดยมนุษย์ที่อยู่ในที่ๆห่างออกไป กระบวนการเรียนรู้ทั้งหมดจะได้รับจากผู้สั่งการโดยอ้างอิงตามข้อมูลที่ได้จากประสาทสัมผัสต่างๆ ทั้งจากตัวผู้สั่งการและตัวหุ่น ตัวอย่างง่ายๆของหุ่นยนต์แบบนี้ก็คือรถบังคับวิทยุนั่นเอง

หุ่นยนต์ขับเคลื่อนอัตโนมัติ เป็นหุ่นยนต์ที่สามารถตัดสินใจทำงานในสภาพแวดล้อมที่ไม่ได้ถูกกำหนดมาตั้งแต่ต้นโดยไม่ต้องอาศัยคำสั่ง(หรือคำแนะนำ)จากมนุษย์ตลอดเวลา หุ่นยนต์ขับเคลื่อนอัตโนมัติที่สมบูรณ์ควรจะรับทราบข้อมูลของสภาพแวดล้อมและเข้าใจได้ด้วยตัวเอง และสามารถตัดสินใจเลือกเส้นทางเดินจากจุดเริ่มต้นไปยังเป้าหมายได้โดยไม่ต้องสามารถความช่วยเหลือเพิ่มเติมจากปัจจัยภายนอก

รูปที่ 2.1 แสดงตัวอย่างหุ่นยนต์อัตโนมัติ

2.2 ทฤษฎีทางด้านเจเนติกอัลกอริธึม

เจเนติกอัลกอริธึม (Genetic Algorithm) เป็นชื่อเรียกวิธีการคิดคำนวณวิธีหนึ่งที่สามารถพัฒนาคำตอบที่เหมาะสมที่สุดขึ้นมาด้วยตัวเองได้ เจเนติกอัลกอริธึมมีพื้นฐานมาจากกระบวนการวิวัฒนาการตัวเองของยีนส์ (Genes) ในสิ่งมีชีวิตในธรรมชาติ ซึ่งจะสร้างวิธีการแก้ปัญหารูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆขึ้นมา เรียกวิธีการแก้ปัญหาแต่ละวิธีว่า "โครโมโซม" (Chromosome) แล้วนำกลุ่มวิธีการแก้ปัญหาผ่านกระบวนการทางวิวัฒนาการต่างๆที่เรียกว่า "กระบวนการทางพันธุกรรม" (Genetic Operator) เพื่อพัฒนาประสิทธิภาพในการแก้ปัญหาให้ดียิ่งขึ้น โดยโครโมโซมที่มีประสิทธิภาพมากจะมีโอกาสได้พัฒนาตัวเองในกระบวนการครั้งต่อๆไปได้มากกว่าโครโมโซมอื่น

ปัญหาสำคัญของการนำเจเนติกอัลกอริธึมไปใช้ก็คือการเปลี่ยนวิธีการแก้ปัญหาให้อยู่ในรูปแบบที่พร้อมต่อการพัฒนาด้วยกระบวนการทางพันธุกรรม โดยทั่วไปแล้ว โครโมโซมจะมีลักษณะเป็นบิตข้อมูลของเลขจำนวนเต็มเรียงต่อกัน แต่ก็สามารถดัดแปลงให้อยู่ในรูปแบบที่ซับซ้อนขึ้นได้เช่นกัน ซึ่งรูปแบบที่เลือกใช้จะส่งผลอย่างมากต่อประสิทธิภาพในการค้นหาคำตอบที่ดีที่สุดของเจเนติกอัลกอริธึม ปัญหาที่สำคัญอีกอย่างหนึ่งคือการสร้างวิธีคำนวณหาความเหมาะสมในการแก้ปัญหาของแต่ละโครโมโซม (Fitness Function) ที่เหมาะสมกับปัญหานั้นๆ ซึ่งค่าความเหมาะสม (Fitness Value) เป็นค่าที่ใช้บ่งบอกว่าแต่ละโครโมโซมมีประสิทธิภาพในการแก้ปัญหามากน้อยแค่ไหน และเป็นค่าที่แสดงโอกาสในการเลือกโครโมโซมที่มีประสิทธิภาพสูงมาใช้งาน

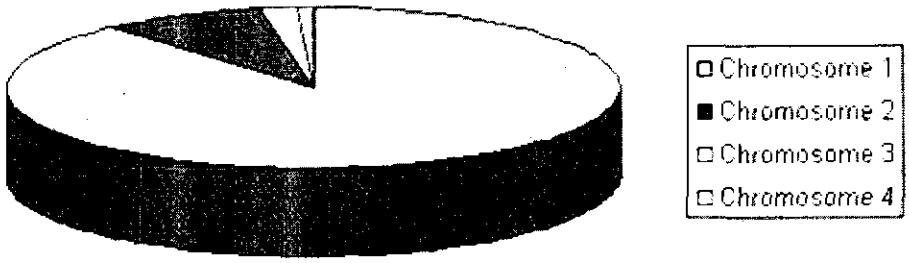
2.2.1 วิธีการคัดเลือกโครโมโซม (Selection)

การคัดเลือกโครโมโซมจะพิจารณาจากค่าความเหมาะสม ซึ่งแบ่งออกเป็นวิธีการต่างๆ ดังต่อไปนี้

2.2.1.1 วิธีคัดเลือกแบบวงล้อรูเล็ต (Roulette Wheel Selection)

คือ กระบวนการคัดเลือกโครโมโซมต้นแบบ (Parent Chromosomes) โดยพิจารณาจากค่าความเหมาะสม (Fitness) โดยมีลักษณะการทำงานเช่นเดียวกับการเล่นวงล้อรูเล็ต (Roulette Wheel) ทุกโครโมโซมจะครองพื้นที่บนวงล้อรูเล็ตตามสัดส่วนค่าความเหมาะสมของตัวเอง เมื่อโยนลูกกลิ้งให้กลิ้งอยู่ในวงล้อรูเล็ต แล้วลูกกลิ้งหยุด ณ ตำแหน่งใด โครโมโซมที่ครองพื้นที่ในบริเวณนั้นจะถูกเลือกเป็นโครโมโซมต้นแบบให้รุ่นต่อไป โครโมโซมที่มีประสิทธิภาพดี มีค่าความเหมาะสมมากกว่าโครโมโซมอื่นในรุ่นเดียวกัน ขนาดของสัดส่วนในวงล้อรูเล็ตก็จะมาก ย่อมมีโอกาสจะถูกเลือกให้เป็นโครโมโซมต้นแบบมากกว่า ดังโครโมโซมหมายเลข 1 ในรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

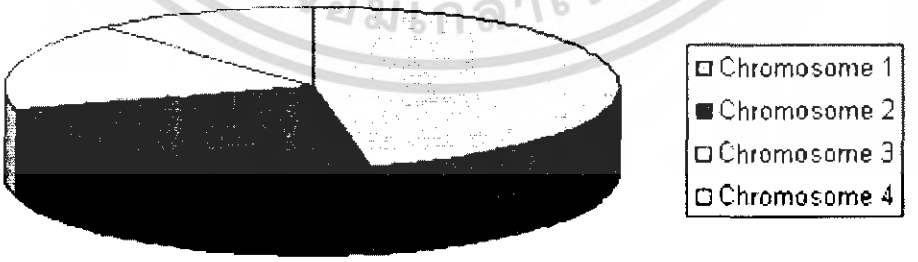


รูปที่ 2.2 แสดงแผนภาพสัดส่วนค่าความเหมาะสมของโครโมโซมในวงล้อรูเล็ต

แต่การใช้วิธีการคัดเลือกแบบวงล้อรูเล็ตจะเกิดปัญหาขึ้นถ้าค่าความเหมาะสมของแต่ละโครโมโซมมีความแตกต่างกันมาก ยกตัวอย่างเช่นถ้าโครโมโซมที่ดีที่สุดมีค่าความเหมาะสมคิดเป็น 90% ของผลรวมค่าความเหมาะสมของโครโมโซมทั้งหมด ในการคัดเลือกครั้งนั้นจะได้แต่โครโมโซมที่ดีที่สุดเกือบทั้งรุ่น ทำให้รูปแบบของโครโมโซม หรือวิธีการแก้ปัญหาขาดความหลากหลาย

2.2.1.2 วิธีคัดเลือกแบบจัดอันดับ (Rank Selection)

เกิดจากการจัดอันดับค่าความเหมาะสมของโครโมโซม โดยโครโมโซมที่ดีที่สุดจะมีโอกาสถูกเลือกเป็นโครโมโซมต้นแบบมากเป็นอันดับหนึ่ง โครโมโซมที่มีค่าความเหมาะสมรองลงมาจะมีโอกาสถูกเลือกมาเป็นอันดับ 2 เป็นเช่นนี้ไปเรื่อยๆ จนถึงโครโมโซมอันสุดท้ายที่มีค่าความเหมาะสมน้อยที่สุดก็จะมีโอกาสถูกเลือกเป็นโครโมโซมต้นแบบน้อยที่สุด จากภาพด้านล่างจะเห็นความแตกต่างของการคัดเลือกโครโมโซมด้วยวิธีการคัดเลือกแบบวงล้อรูเล็ตและวิธีการคัดเลือกแบบจัดอันดับ



รูปที่ 2.3 แสดงโอกาสที่แต่ละโครโมโซมจะถูกเลือกด้วยวิธีการคัดเลือกแบบจัดอันดับ

ซึ่งวิธีการคัดเลือกแบบจัดอันดับนั้นแต่ละโครโมโซมจะมีโอกาสในการถูกเลือกเป็นสัดส่วนพอ ๆ กัน แต่จะทำให้กระบวนการในการหาคำตอบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้เวลามากขึ้นเนื่องจากโครโมโซมที่ดีที่สุดมีโอกาสถูกเลือกไปพัฒนาประสิทธิภาพเท่าๆ กับโครโมโซมอื่น

2.2.1.3 วิธีคัดเลือกแบบการแข่งขัน (Tournament Selection)

เกิดจากการสุ่มคัดเลือกโครโมโซมขึ้นมาจำนวนหนึ่ง แล้วคัดเลือกโครโมโซมที่ดีที่สุดออกจากจำนวนโครโมโซมที่มีอยู่จนกระทั่งได้จำนวนโครโมโซมต้นแบบตามจำนวนที่ต้องการ ทำให้สามารถคัดเลือกโครโมโซมต้นแบบที่มีประสิทธิภาพระดับหนึ่งและยังคงความหลากหลายของรูปแบบการแก้ปัญหาไว้ด้วย

2.2.2 กระบวนการทางพันธุกรรม (Genetic Operators)

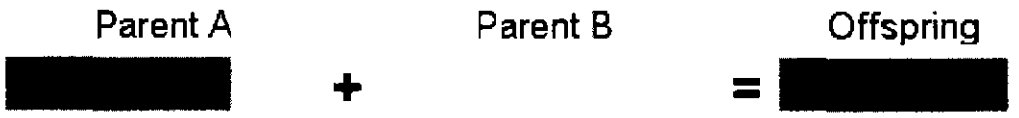
กระบวนการทางพันธุกรรม คือ กระบวนการที่ทำให้โครโมโซม หรือวิธีการแก้ปัญหามีความหลากหลายมากขึ้น ความหลากหลายนี้จะส่งผลดีต่อกระบวนการหาคำตอบเนื่องจากจะเป็นการแก้ปัญหาด้วยค่าต่างๆที่เป็นไปได้ทั้งหมด โดยกระบวนการทางพันธุกรรมถูกออกแบบมาให้มีการทำงานคล้ายกับกระบวนการพัฒนาโครโมโซมที่มีอยู่จริงในธรรมชาติ

2.2.2.1 การสับเปลี่ยนชิ้นส่วน (*Crossover*)

เลียนแบบมาจากการสับเปลี่ยนชิ้นส่วนของยีนส์ ในธรรมชาติ โดยจะเลือกโครโมโซมต้นแบบ (Parent Chromosomes) มาสองตัว แล้วสลับบางส่วนของแต่ละโครโมโซมให้กันและกัน หลังการแลกเปลี่ยนชิ้นส่วนกันแล้ว จะได้โครโมโซมใหม่ที่เรียกว่า โครโมโซมผลลัพธ์ (Off-spring Chromosomes) ซึ่งวิธีแลกเปลี่ยนชิ้นส่วนแบบต่างๆมีดังนี้

1. **Single Point Crossover –** ทำการแลกเปลี่ยนชิ้นส่วนโดยระบุเพียงตำแหน่งเดียว โดยโครโมโซมผลลัพธ์จะเกิดจากการต่อโครโมโซมจากจุดเริ่มต้นโครโมโซมต้นแบบอันแรกไปจนถึงตำแหน่งที่มีการสับเปลี่ยนชิ้นส่วน ต่อด้วยโครโมโซมต้นแบบอันที่สองจากจุดที่ทำการสับเปลี่ยนชิ้นส่วน จนถึงท้ายสุดของโครโมโซม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



11001011+11011**111 = 11001111**

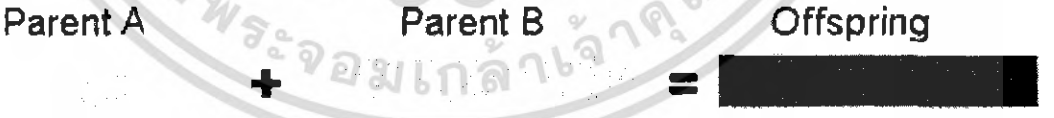
รูปที่ 2.4 ภาพจำลองการแลกเปลี่ยนชิ้นส่วนโดยระบุเพียงตำแหน่งเดียว

2. **Two Point Crossover –** เราจะเลือกจุดเพื่อสับเปลี่ยนชิ้นส่วน 2 จุด จากตำแหน่งเริ่มต้นไบนารี่สตริงถึงจุดสับเปลี่ยนชิ้นส่วน จุดแรกจะคัดลอกมาจากโครโมโซมต้นแบบตัวแรก ส่วนที่อยู่ระหว่างจุดสับเปลี่ยนชิ้นส่วนแรกและจุดสับเปลี่ยนชิ้นส่วนสอง จะคัดลอกมาจากโครโมโซมต้นแบบอื่น ส่วนที่เหลือจะนำมาจากโครโมโซมต้นแบบตัวแรกอีกครั้ง

11001**011** + 1**1**011**111** = **11011111**

รูปที่ 2.5 ภาพจำลองการแลกเปลี่ยนชิ้นส่วนโดยระบุ 2 ตำแหน่ง

3. **Uniform Crossover** – บิตในโครโมโซมของโครโมโซมต้นแบบจะถูกสุ่มขึ้นเพื่อใช้ในการประกอบเป็นลูกหลานใหม่

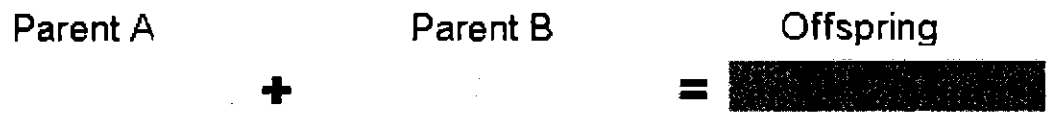


1**10**01**011** + **1**1011**101** = 11011111

รูปที่ 2.6 ภาพจำลองการแลกเปลี่ยนชิ้นส่วนแบบสุ่มสร้างขึ้นใหม่

4. **Arithmetic Crossover** - เป็นวิธีที่ใช้กระบวนการทางคณิตศาสตร์ในการสร้างลูกหลานใหม่ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



11001011 + 11011111 = 11001001 (AND)

รูปที่ 2.7 ภาพจำลองการแลกเปลี่ยนชิ้นส่วนแบบใช้กระบวนการทางคณิตศาสตร์

5. **Permutation Single Point Crossover** – ทำการเลือกตำแหน่งสับเปลี่ยนชิ้นส่วนมาหนึ่งจุด ค่าที่ใช้ในการสลับลำดับจะถูกคัดลอกจากพ่อแม่ตัวแรกไปถึงจุดสับเปลี่ยนชิ้นส่วน จากนั้นพ่อแม่อีกตัวหนึ่งจะทำการค้นหาหมายเลขที่ยังไม่มีในลูกหลานและทำการเพิ่มต่อเข้าไป

(**1 2 3 4 5** 6 7 8 9) + (4 5 3 **6 8 9 7** 2 1) = (**1 2 3 4 5** 6 8 9 7)

รูปที่ 2.8 ภาพจำลองการแลกเปลี่ยนชิ้นส่วนแบบสลับลำดับ

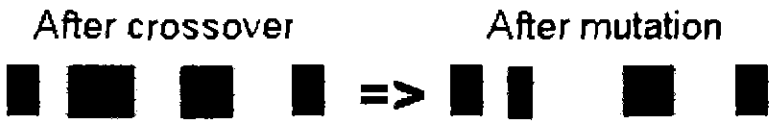
ในขั้นตอนกระบวนการสับเปลี่ยนชิ้นส่วน ถ้าหากไม่มีการสับเปลี่ยนชิ้นส่วนเกิดขึ้น รุ่นลูกจะมีลักษณะเหมือนพ่อแม่ แต่หากเกิดการสับเปลี่ยนชิ้นส่วน ลูกหลานใหม่จะถูกประกอบขึ้นจากโครโมโซมของพ่อแม่ทำให้มีการเปลี่ยนแปลงไปจากรุ่นเดิม ถ้าค่าความเป็นไปได้ในการสับเปลี่ยนชิ้นส่วน (Crossover Probability) เป็น 100% แล้วหมายความว่าลูกที่ได้เกิดจากการสับเปลี่ยนชิ้นส่วนทุกบิตแต่ถ้าความเป็นไปได้ในการสับเปลี่ยนชิ้นส่วนมีค่าเป็น 0% รุ่นใหม่ที่กำเนิดมาจะมีโครโมโซมแบบเดียวกับประชากรเก่า กระบวนการสับเปลี่ยนชิ้นส่วนมีขึ้นเพื่อให้ส่วนดีของโครโมโซมเก่ายังมีการสืบทอดอยู่และทำให้รุ่นใหม่มีประสิทธิภาพดีขึ้น ดังนั้นจึงเป็นการดีที่จะให้มีการหลงเหลือของประชากรเก่าไว้ไปยังรุ่นต่อไป

2.2.2.2 การกลายพันธุ์ (Mutation)

คือ กระบวนการที่เลียนแบบมาจากธรรมชาติเช่นกัน กระบวนการกลายพันธุ์จะสุ่มเปลี่ยนข้อมูลบางส่วนในโครโมโซมที่ผ่านการสับเปลี่ยนชิ้นส่วนมาแล้ว รูปแบบของการกลายพันธ์ต่างๆมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Bit Inversion – เลือกบิตใดมาหนึ่งบิตแล้วเปลี่ยนค่าบิตนั้นใหม่ให้มีค่าตรงกันข้ามกับค่าเดิม



11001001 => 10001001

รูปที่ 2.9 ภาพจำลองการกลายพันธุ์แบบกลับค่า

2. Order Changing – วิธีการนี้จะเลือก 2 หมายเลขขึ้นมาและทำการสลับที่กัน

(1 **2** 3 4 5 6 **8** 9 7) => (1 **8** 3 4 5 6 **2** 9 7)

รูปที่ 2.10 ภาพจำลองการกลายพันธุ์แบบเปลี่ยนลำดับ

3. Value Mutation – จะใช้การบวกหรือลบเลขจำนวนน้อยๆ หรืออาจคิดเป็นเปอร์เซ็นต์ในการเพิ่มหรือลดค่าในจำนวนที่เราทำการเลือกไว้

(1.29 5.68 **2.86 4.11** 5.55) => (1.29 5.68 **2.73 4.22** 5.55)

รูปที่ 2.11 ภาพจำลองการกลายพันธุ์แบบเปลี่ยนค่า

ในขั้นตอนการกลายพันธุ์หากไม่มีการกลายพันธุ์เกิดขึ้น ลูกหลานที่ได้จะเกิดมาหลังจากการสับเปลี่ยนชิ้นส่วนโดยไม่มีการเปลี่ยนแปลง บางกรณีที่เจเนติกอัลกอริธึมไม่สามารถแก้ปัญหาได้ถูกต้อง เช่น ปัญหา Hill Climbing ที่อาจเจอ Local Maxima, Plateau และ Shoulder(สันเขา) โดยมีการกลายพันธุ์ขึ้น ทำให้ส่วนใดส่วนหนึ่งของโครโมโซมจะมีการเปลี่ยนแปลงทำให้ดูเหมือนมีการเปลี่ยนแบบก้าวกระโดด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 กระบวนการของเจเนติกอัลกอริธึม

กระบวนการของ เจเนติกอัลกอริธึมสามารถเขียนเป็นขั้นตอนหลักๆ ได้ดังนี้

Step 1 : ทำการเปลี่ยนข้อมูลต่างๆที่ใช้แก้ปัญหาให้อยู่ในรูปของโครโมโซมที่มีความยาวคงที่ค่าหนึ่ง , กำหนดจำนวนของประชากร (N) , กำหนดค่าความเป็นไปได้ของการสับเปลี่ยนชิ้นส่วน (p_c) และ ความเป็นไปได้ของการกลายพันธุ์ (p_m)

Step 2 : สร้างวิธีการหาค่าความเหมาะสมเพื่อคำนวณหาค่าความเหมาะสมของแต่ละโครโมโซมในประชากร ค่านี้จะใช้ในการเลือกโครโมโซมเป็นคู่เพื่อสร้างประชากรใหม่

Step 3 : สร้างประชากรรุ่นแรกขึ้นมาโดยการสุ่มเป็นจำนวน N ตัว

$$x_1, x_2, x_3, \ldots, x_N$$

Step 4 : คำนวณค่าความเหมาะสมของโครโมโซมแต่ละตัว

$$f(x_1), f(x_2), f(x_3), \ldots, f(x_N)$$

Step 5 : เลือกโครโมโซม 2 ตัวจากประชากรในรุ่นปัจจุบัน โดยโอกาสในการสุ่มได้ขึ้นอยู่กับค่าความเหมาะสมของโครโมโซมแต่ละตัว ยิ่งมีค่าสูงโอกาสที่จะถูกเลือกยิ่งมาก

Step 6 : สร้างโครโมโซมใหม่จำนวน 2 ตัว ด้วยการสับเปลี่ยนชิ้นส่วนและการกลายพันธุ์

Step 7 : เก็บโครโมโซมใหม่ไว้เป็นประชากรของรุ่นต่อไป

Step 8 : ย้อนกลับไปทำตั้งแต่ Step 5 ซ้ำ จนจำนวนของประชากรรุ่นต่อไปมีเท่ากับประชากรรุ่นเดิม N ตัว

Step 9 : แทนที่ประชากรชุดปัจจุบันทั้งหมดด้วยประชากรรุ่นใหม่

Step 10 : ย้อนกลับไปทำตั้งแต่ Step 4 ซ้ำ จนกว่าเงื่อนไขในการสิ้นสุดจะเป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.12 แสดงขั้นตอนการทำงานของเจเนติกอัลกอริธึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าเจเนติกอัลกอริธึมมีลักษณะการทำงานเหมือนเดิมซ้ำๆ (Iterative) โดยโครโมโซมแต่ละชุดที่ถูกสร้างขึ้นใหม่จากการทำงานในแต่ละรอบเรียกว่า รุ่น (Generation) จำนวนรุ่นของโครโมโซมโดยทั่วไปจะมีตั้งแต่ 50 จนถึงมากกว่า 500 รุ่น เมื่อสิ้นสุดการทำงานควรจะได้โครโมโซมที่มีค่าความเหมาะสมสูงกว่าโครโมโซมตัวอื่นๆ อย่างเด่นชัดอย่างน้อย 1 ตัว

เนื่องจากเจเนติกอัลกอริธึมใช้การค้นหาแบบสุ่มในการเลือกคำตอบ บางครั้งค่าความเหมาะสมของกลุ่มประชากรจะคงที่อยู่เป็นเวลานานหลายรุ่นกว่าจะสร้างโครโมโซมที่ดีมากกว่าได้ ทำให้เกิดปัญหาการทำงานที่ไม่สามารถสิ้นสุดได้ วิธีแก้ปัญหาง่ายๆก็คือให้หยุดทำงานหลังจากมีจำนวนรุ่นถึงค่าหนึ่ง และเลือกเอาโครโมโซมที่ดีที่สุดจากประชากร ถ้าไม่ดีพอก็จะให้ระบบทำงานใหม่ทั้งหมดอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การทำงานของระบบตามเอกสารอ้างอิง

3.1 บทนำ

การหาเส้นทางด้วยเจเนติกอัลกอริธึมที่มีฐานความรู้ (Knowledge Based Genetic Algorithm) นั้นเป็นการนำการค้นหาที่ใช้ฐานความรู้ (Heuristic Search) ในขอบเขตข้อมูลที่เราสนใจ (Domain Knowledge) มาใช้กับเจเนติกอัลกอริธึมที่ถูกปรับปรุงกระบวนการทำงานเพื่อใช้สำหรับหาเส้นทางโดยเฉพาะ (Problem-specific Knowledge) ซึ่งจะเป็นการแก้ปัญหาที่ใช้ข้อมูลจากฐานความรู้ และเหมาะกับการหาเส้นทางที่ดีที่สุดในสภาพแวดล้อมที่มีอุปสรรค โดยมีขั้นตอนในการหาคำตอบดังนี้

3.2 การกำหนดปัญหา

กำหนดให้เจเนติกอัลกอริธึมที่มีฐานความรู้หาเส้นทางจากจุดเริ่มต้นไปยังจุดเป้าหมายที่กำหนดโดยไม่มีส่วนใดส่วนหนึ่งของเส้นทางตัดผ่านอุปสรรค แต่ละเส้นทางที่เกิดขึ้นก็จะมีประสิทธิภาพแตกต่างกันไป การพิจารณาว่าแต่ละเส้นทางมีประสิทธิภาพมากน้อยแค่ไหนสามารถพิจารณาได้จากหลายปัจจัย เช่น เวลาที่ใช้ในการเดินทางจุดเริ่มต้นไปยังจุดเป้าหมาย, ระยะทางทั้งหมด หรือ ความคล่องตัวในการเคลื่อนที่ไปตามเส้นทางนั้น ซึ่งโดยปกติแล้ว "ระยะทาง" จะเป็นประเด็นหลักในการพิจารณา

3.3 การแทนค่าปัญหา

การแทนค่าเป็นหัวใจหลักของการแก้ปัญหาด้วยเจเนติกอัลกอริธึม จึงมีการแบ่งประเภทของตัวแปรที่ต้องพิจารณาออกเป็น 2 ประเภทใหญ่ๆ คือ

1) ตัวแปรที่มีความเกี่ยวข้องกับการแก้ปัญหา แต่ไม่ถูกนำไปใช้ในการแก้ปัญหา เรียกว่า สภาพแวดล้อม
2) ตัวแปรที่ถูกนำไปพิจารณาเพื่อการแก้ปัญหา ซึ่งมีทั้งตัวแปรที่เป็นส่วนหนึ่งในการทำงานของเจเนติกอัลกอริธึม เช่น ค่าความเป็นไปได้ของการทำกระบวนการทางพันธุกรรมต่างๆ และตัวแปรที่เอาไว้ใช้แก้ปัญหา เช่น โครโมโซม เป็นต้น

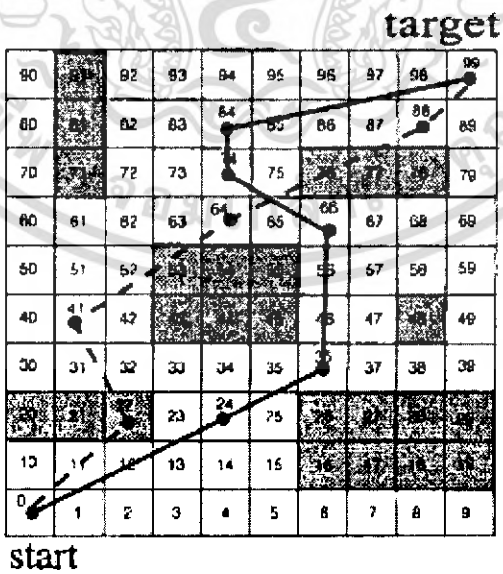
ซึ่งรายละเอียดการแทนค่าของตัวแปรต่างๆเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 การกำหนดคุณสมบัติของสภาพแวดล้อม

บริเวณที่ให้หุ่นยนต์เคลื่อนที่ไปได้จะแสดงด้วยตาราง (Grid) ขนาด 10x 10 ช่อง แต่ละช่อง (Node) มีหมายเลขเป็นจำนวนเต็มตั้งแต่ 0 – 99 กำกับอยู่แบบเรียงลำดับจากน้อย (ด้านซ้ายล่าง) ไปมาก (ด้านขวาบน) โดยทั่วไปการใช้ตารางมีรูปแบบการเคลื่อนที่จากช่องหนึ่งไปอีกช่องหนึ่งได้เฉพาะ 8 ช่องรอบข้างของช่องที่เป็นจุดเริ่มต้นเท่านั้น และจะนำทิศทางนั้นมาใช้เป็นเส้นทางการเคลื่อนที่ แต่การแทนค่าสภาพแวดล้อมในที่นี้เป็นการผสมผสานรูปแบบของตารางและคู่ลำดับ (Co-ordinate) เข้าด้วยกัน หมายเลขที่กำกับไว้ในแต่ละช่องจะคล้ายๆกับการอ้างอิงตำแหน่งแบบคู่ลำดับ คือ หลักหน่วยของหมายเลขประจำแต่ละช่องจะเสมือนการแสดงตำแหน่งตามแกน X และเลขในหลักสิบจะเสมือนการแสดงตำแหน่งตามแกน Y แต่จะแตกต่างกันตรงที่แต่ละช่องจะไม่มีความเกี่ยวพันกัน ทำให้มีรูปแบบการเคลื่อนที่มีไม่จำกัดสามารถจัดการกับโครงสร้างของโครโมโซมและกระบวนการทางพันธุกรรมได้ง่ายขึ้น โดยการแบ่งสภาพแวดล้อมออกเป็นส่วนๆแยกจากกัน

และเนื่องจากการเพิ่มขนาดของอุปสรรคดังที่กล่าวมา ทำให้บริเวณขอบของอุปสรรคเกิดมุมโค้งที่มีรัศมีเท่ากับ r เพื่อให้สะดวกในการคำนวณจึงต้องขยายขอบเขตของอุปสรรคให้พอดีกับเส้นตารางก่อนจะนำไปขนาดของอุปสรรคไปใช้จริง ทำให้ขนาดของอุปสรรคที่ใช้บนตารางมีขนาดใหญ่ขึ้น และทำให้สามารถมองการเคลื่อนที่ของหุ่นยนต์เสมือนเป็นการเคลื่อนที่ของจุดไปตามเส้นทางที่พิจารณาได้ ตัวอย่างตามรูปที่ 3.1

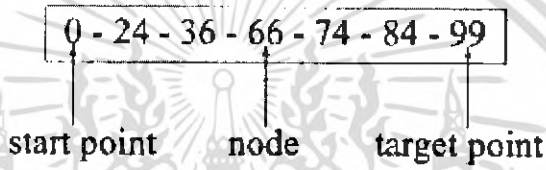


รูปที่ 3.1 แสดงสภาพแวดล้อมและการแทนเส้นทางด้วยช่องหมายเลขต่างๆบนตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การกำหนดคุณสมบัติของโครโมโซม

เนื่องจากเราต้องการหาเส้นทางที่มีประสิทธิภาพมากที่สุด โครโมโซมจึงใช้แทนเส้นทางการเคลื่อนที่ของหุ่นยนต์ โดยให้แต่ละยีนส์แทนด้วยเลขจำนวนเต็ม 1 ค่าซึ่งนำมาจากหมายเลขของแต่ละช่องบนตารางประกอบกันเป็นโครโมโซมที่มีความยาวไม่คงที่ และในแต่ละช่วงของเส้นทางจะเกิดจากการเชื่อมช่อง 2 ช่องเข้าด้วยกัน ดังนั้นการกำหนดเส้นทางจากจุดเริ่มต้นไปที่จุดเป้าหมายจะเกิดจากการนำหมายเลขช่องของจุดเริ่มต้นจากยีนส์อันแรกเป็นที่เริ่มลากเส้นทางการเคลื่อนที่ และหมายเลขช่องต่อๆมาจะถูกเชื่อมเป็นเส้นทางไปจนถึงหมายเลขช่องในยีนส์อันสุดท้ายในโครโมโซม หรือหมายเลขช่องของจุดเป้าหมายนั่นเอง ดังรูปที่ 3.2



รูปที่ 3.2 แสดงตัวอย่างของโครโมโซม

เส้นทางที่เหมาะสมจะต้องไม่ชนกับสิ่งกีดขวาง เช่น ไม่มีช่องใดอยู่ในขอบเขตของสิ่งกีดขวาง หรือไม่มีส่วนหนึ่งส่วนใดของเส้นตัดผ่านบริเวณที่เป็นสิ่งกีดขวาง ความยาวของโครโมโซมไม่คงที่ มีค่าอยู่ระหว่าง 2 (จุดเริ่มต้นกับจุดสุดท้าย) ถึงความยาวที่มากที่สุด คือจำนวนของหมายเลขช่องทั้งหมดในตาราง

3.4 การประเมินค่า (Evaluation)

ถ้ามี 2 เส้นทางที่ไม่เหมือนและไม่ชนกับสิ่งกีดขวางตั้งแต่ต้นทางถึงปลายทางทั้งคู่ หากเราพิจารณาประสิทธิภาพของเส้นทางจากการชนอุปสรรคเพียงอย่างเดียวจะไม่สามารถเห็นความแตกต่างของ 2 เส้นทางนี้ได้ จึงต้องใช้การคำนวณเพื่อประเมินหาค่าความเหมาะสมที่เกิดจากช่องต่างๆในเส้นทาง เพื่อช่วยแยกระหว่างเส้นทางที่เหมาะสมกับเส้นทางที่ไม่เหมาะสม และช่วยบอกความแตกต่างของคุณภาพเส้นทางแต่ละเส้นด้วย โดยการคำนวณเพื่อหาค่าความเหมาะสมเป็นดังนี้

$$F_{cost} = \sum_{i=1}^{N} (d_i + \beta_i C) \qquad (3.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

72744

โดย i คือ ลำดับของเส้นที่เชื่อมระหว่างช่อง

N คือ จำนวนเส้นที่เชื่อมระหว่างช่องในเส้นทางนั้น

d_i คือ ระยะทางแบบยูคลิเดียน (Euclidean's Distance) ระหว่างช่องที่อยู่ติดกัน

C คือ ค่าคงที่

β_i คือ ค่าสัมประสิทธิ์แสดงค่าความลึกที่เกิดการชน ซึ่งหาได้จาก

$$\beta_i = \begin{cases} 0 & \text{เมื่อส่วนของเส้นที่ i นั้นเหมาะสมแล้ว} \\ \sum_{j=1}^{M} \alpha_j & \text{เมื่อส่วนของเส้นที่ i นั้นตัดผ่านสิ่งกีดขวาง} \end{cases} \qquad (3.2)$$

โดย M คือ จำนวนของสิ่งกีดขวางที่ส่วนของเส้นตัดผ่าน

α_j คือ ค่าที่อธิบายความลึกที่ส่วนของเส้นตัดผ่านสิ่งกีดขวางลำดับที่ j ซึ่งหาได้จากการเคลื่อนที่ที่น้อยที่สุดที่ทำให้ส่วนของเส้นพ้นจากสิ่งกีดขวาง

นอกจากนี้ การประเมินค่ายังทำให้เกิดการคลุมเครือกันระหว่างคำตอบที่เหมาะสมกับคำตอบที่ยังไม่เหมาะสม ซึ่งนับว่าเป็นเรื่องที่ดี เนื่องจากในแง่ของวิวัฒนาการ เส้นทางที่เหมาะสมแต่คุณภาพไม่ดี ไม่จำเป็นว่าจะต้องดีกว่าเส้นทางที่เกือบจะมีความเหมาะสมแต่มีคุณภาพดี เพราะถ้าเราให้โอกาสแก่เส้นทางที่มีคุณภาพดีแต่ยังไม่เหมาะสมมากขึ้น เส้นทางแบบนี้จะกลายเป็นคำตอบที่ดีง่ายกว่า และเพื่อให้ใช้เวลาในการคำนวนลดลง ข้อมูลบางอย่างที่ได้จากการประเมินค่าจะต้องถูกบันทึกไว้เป็นคุณสมบัติของโครโมโซมเมื่อต้องการใช้ข้อมูลเหล่านี้เป็นฐานความรู้ในกระบวนการเฉพาะทาง (Specialized Operator) จะได้นำค่ามาใช้ได้ทันที ไม่ต้องทำการคำนวณซ้ำอีก ซึ่งข้อมูลจำพวกนี้ ได้แก่ ค่าความเหมาะสมต่างๆ เช่น ความสมบูรณ์ของเส้นทาง, ความเหมาะสมของแต่ละตำแหน่ง, จำนวนของช่องที่อยู่บริเวณเดียวกับอุปสรรค, จำนวนส่วนของเส้นทางที่ตัดผ่านอุปสรรค และสิ่งกีดขวางที่มีเส้นตัดผ่าน เป็นต้น

3.5 การเลือกโครโมโซมต้นแบบ (Selection Parent Chromosomes)

ใช้วิธีการคัดเลือกแบบแข่งขันในการเลือกโครโมโซมไปเป็นต้นแบบของประชากรรุ่นต่อๆไป โดยโครโมโซมที่แทนเส้นทางที่มีประสิทธิภาพมากกว่าจะมีความเป็นไปได้ที่จะถูกเลือกมาก และโครโมโซมที่แทนเส้นทางที่มีประสิทธิภาพน้อยกว่าก็จะมีความเป็นไปได้น้อยกว่าที่จะถูกเลือกไปเป็นโครโมโซมต้นแบบ ประกอบกับแนวคิดของการคัดเฉพาะโครโมโซมที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Elitism) ซึ่งเป็นการป้องกันไม่ให้เกิดเหตุการณ์ที่มีโครโมโซมคุณภาพดีอยู่ในประชากรแต่ไม่มีโอกาสที่ถูกเลือกไปผ่านกระบวนการทางพันธุกรรมเพื่อสร้างลูกหลานใหม่ ทำให้เกิดการใช้วิธีการคัดลอกโครโมโซมที่ดีที่สุดหลังผ่านกระบวนการต่างๆไปยังประชากรใหม่ก่อน ส่วนประชากรอื่นๆยังคงทำตามขั้นตอนตามปกติที่ได้กล่าวมา การคัดลอกโครโมโซมที่ดีที่สุดเก็บไว้ยังประชากรรุ่นต่อๆไปนี้สามารถเพิ่มประสิทธิภาพของโครโมโซมในแต่ละประชากรได้อย่างรวดเร็วเนื่องจากสามารถป้องกันการสูญหายของโครโมโซมที่มีคุณภาพดีที่สุดได้

3.6 กระบวนการทางพันธุกรรม (Genetic Operators)

ค่าที่ได้จากการประเมินจะบอกความแตกต่างของประสิทธิภาพของแต่ละเส้นทาง เส้นทางที่มีประสิทธิภาพน้อยจะยังเก็บไว้ในประชากร เพราะสามารถกลายเป็นเส้นทางที่เหมาะสมเมื่อถูกปรับปรุงด้วยวิธีทางพันธุกรรมประเภทต่างๆให้มีประสิทธิภาพมากขึ้น

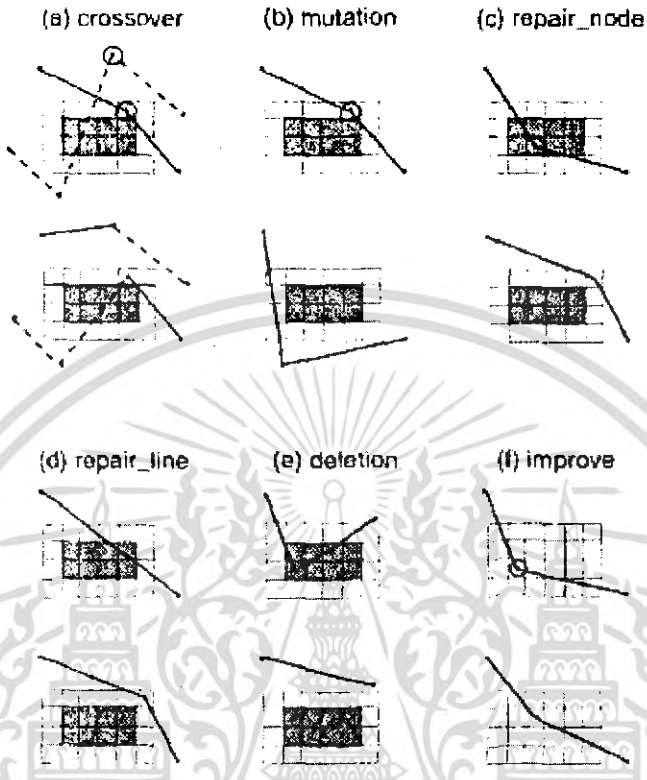
การหาคำตอบโดยการใช้เจเนติกอัลกอริธึมโดยทั่วไปจะใช้กระบวนการทางพันธุกรรม (Genetic Operators) อยู่ 2 ชนิดคือ การสับเปลี่ยนชิ้นส่วน (Crossover) และการกลายพันธุ์ (Mutation) ซึ่งยังไม่เพียงพอในการใช้แก้ปัญหานี้ จึงมีการเพิ่มกระบวนการเฉพาะทาง (Specialized Operator) เพื่อแก้ปัญหาเกี่ยวกับการหาเส้นทางโดยเฉพาะอีก 4 กระบวนการ ทำให้เจเนติกอัลกอริธึมมีประสิทธิภาพในการหาคำตอบมากขึ้น โดยแต่ละกระบวนการมีรายละเอียดดังนี้

3.6.1 กระบวนการมาตรฐาน

1. **Crossover** คือ กระบวนการที่จะทำการสับเปลี่ยนชิ้นส่วนของโครโมโซมโดยการสุ่มเลือก 1 ยีนส์มาจากโครโมโซมต้นแบบทั้งสองอัน แล้วแลกเปลี่ยนชิ้นส่วนที่อยู่หลังจาก 2 ยีนส์ที่ถูกเลือก ซึ่งจะได้โครโมโซมผลลัพธ์ที่มีจำนวนเท่ากับจำนวนของโครโมโซมต้นแบบ แล้วตรวจสอบภายในแต่ละโครโมโซมผลลัพธ์เพื่อลบบางส่วนของเส้นทางที่อาจใช้ยีนส์ที่มีค่าเดียวกัน และเนื่องจากยีนส์ภายในส่วนที่ทำการสับเปลี่ยนชิ้นส่วน และความยาวของส่วนที่ทำการสับเปลี่ยนชิ้นส่วนในแต่ละครั้งมีความแตกต่างกัน ทำให้โครโมโซมผลลัพธ์ที่เกิดขึ้นมีความยาวแตกต่างกัน ผลลัพธ์ที่ได้จึงมีประสิทธิภาพในการหาคำตอบมากกว่า
2. **Mutation** คือ กระบวนการที่จะสุ่มเลือก 1 ยีนส์จากโครโมโซมเพื่อเปลี่ยนค่าของยีนส์เป็นค่าใหม่ที่ไม่ซ้ำกับค่าเดิมของยีนส์ และไม่ใช่ค่าของยีนส์ที่มีอยู่ในเส้นทางแล้ว การทำการกลายพันธุ์เป็นปัจจัยหลักที่ทำให้ประชากรมีความหลากหลาย จึงไม่จำเป็นว่าผลลัพธ์ที่ได้จากการทำกระบวนการนี้จะต้องมีประสิทธิภาพดีขึ้น หาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการเพิ่มความหลากหลายให้กับแต่ละโครโมโซมในประชากรสามารถทำได้โดยเพิ่มค่าความเป็นไปได้ของการทำการกลายพันธุ์ให้มากขึ้น



รูปที่ 3.3 แสดงตัวอย่างการปรับปรุงโครโมโซมด้วยกระบวนการทางพันธุกรรมและกระบวนการเฉพาะทาง

3.6.2 กระบวนการเฉพาะทาง (Specialized Operator)

1. **Node_repair** คือ กระบวนการที่จะย้ายยีนส์ของโครโมโซมที่อยู่บนอุปสรรคไปยังตำแหน่งที่ไม่ใช่อุปสรรค เป็นการค้นหาในขอบเขตจำกัดโดยการหาตำแหน่งใหม่รอบๆอุปสรรคที่ทำให้เส้นทางมีประสิทธิภาพเพิ่มขึ้นมากที่สุด
2. **Line_repair** คือ กระบวนการที่จะปรับปรุงส่วนของเส้นทางที่ตัดผ่านอุปสรรคโดยการเพิ่มยีนส์ที่เหมาะสมลงไประหว่าง 2 ยีนส์ที่ทำให้เกิดส่วนของเส้นทางนั้น โดยตำแหน่งของยีนส์ใหม่ที่ดีที่สุดจะใช้การค้นหาในขอบเขตจำกัดหาจากตำแหน่งที่อยู่รอบๆอุปสรรคที่ถูกส่วนของเส้นทางตัดผ่าน
3. **Deletion** คือ กระบวนการที่ใช้ปรับปรุงคุณภาพของเส้นทางทั้งที่เป็นเส้นทางที่เหมาะสมแล้วและเส้นทางที่ยังไม่เหมาะสม คือ ยังมีบางส่วนที่ติดอยู่กับอุปสรรค โดยการสุ่มเลือก 1 ยีนส์ ขึ้นมาจากโครโมโซม สมมุติให้ยีนส์ที่ถูกเลือกนี้โดนลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วลองเชื่อมเส้นทางระหว่าง 2 ยีนส์ ที่อยู่ใกล้เคียงกับยีนส์ที่ถูกลบ ถ้าการเชื่อมเส้นทางทำให้ประสิทธิภาพของเส้นทางทั้งหมดดีขึ้น ให้ลบยีนส์ที่ถูกเลือกออกไปจริงๆ แล้วใช้เส้นทางใหม่ที่มีการเชื่อมเส้นทางแทน

4. **Improvement** คือ กระบวนการที่ใช้กับเส้นทางที่ไม่มีส่วนใดอยู่บนอุปสรรคให้มีคุณภาพดีขึ้นอีก โดยการสุ่มเลือก 1 ยีนส์จากโครโมโซมแล้วใช้ค้นหาตำแหน่งรอบๆยีนส์ที่ทำให้เส้นทางมีประสิทธิภาพดีขึ้น แล้วย้ายยีนส์ที่ถูกเลือกไปยังตำแหน่งใหม่ แต่ถ้าการย้ายตำแหน่งทำให้ประสิทธิภาพของเส้นทางลดลง ให้คงค่าเดิมของยีนส์ ไว้

แต่การที่กระบวนการทางพันธุกรรมจะสามารถปรับปรุงคุณภาพของเส้นทางได้มากน้อยแค่ไหน จะขึ้นอยู่กับค่าความน่าจะเป็น (Probability) และฐานความรู้ (เช่น ถ้าเส้นทางที่เหมาะสมแล้วให้ใช้ Improve Operator)

3.7 การคัดเลือกโครโมโซมไปเป็นประชากรรุ่นต่อไป (Representation)

ประชากรที่ใช้ในการคำนวณชุดแรกเกิดจากการสุ่มค่าต่างๆสร้างขึ้นมา และนำไปประเมินค่า เพื่อเลือกบางส่วนของประชากรที่มีลักษณะที่ต้องการไปเป็นต้นแบบของการสร้างประชากรรุ่นต่อไป แล้วนำโครโมโซมต้นแบบไปปรับปรุงคุณภาพด้วยกระบวนการต่างๆ ได้เป็นประชากรใหม่ (Off-spring) โดยจะนำโครโมโซมผลลัพธ์ที่มีค่าความเหมาะสมแย่ที่สุดหรือเป็นโครโมโซมที่มีคุณภาพน้อยที่สุดออกจากประชากรใหม่ แทนที่โครโมโซมนั้นด้วยโครโมโซมต้นแบบที่มีค่าความเหมาะสมที่ดีที่สุด จากนั้นจะนำโครโมโซมผลลัพธ์ที่ได้ทั้งหมดไปเป็นประชากรรุ่นต่อไป โดยขั้นตอนต่างๆเหล่านี้จะทำวนไปเรื่อยๆจนกว่าจะได้คำตอบที่พึงพอใจ

3.8 การกำหนดเงื่อนไขเพื่อหยุดการหาคำตอบ

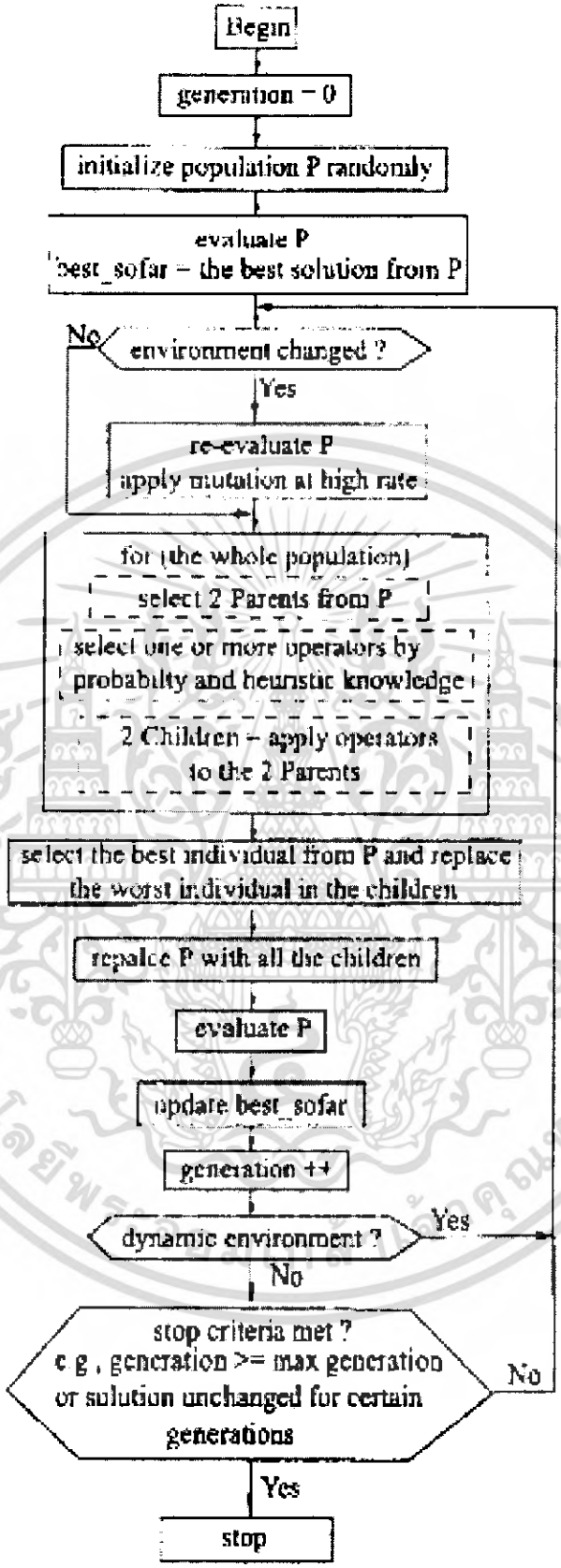
มีการกำหนดเงื่อนไขให้เจเนติกอัลกอริธึมรู้จักคำตอบที่ดีที่สุด โดยการเก็บโครโมโซมที่มีประสิทธิภาพที่ดีที่สุดของทุกรุ่นเอาไว้ ซึ่งโครโมโซมนี้จะกลายเป็นคำตอบของการแก้ปัญหาด้วยเจเนติกอัลกอริธึมที่มีฐานความรู้ เมื่อมีการสร้างประชากรเกินจำนวนรุ่นที่กำหนด หรือโครโมโซมที่ประสิทธิภาพดีที่สุดมีค่าความเหมาะสมคงที่ติดต่อกันหลายๆรุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 แผนผังการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้

แผนผังของการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้เริ่มต้นจากกรอบสี่เหลี่ยมด้านบนสุดของรูปที่ 3.4 ประชากรชุดแรกจะสุ่มสร้างขึ้นมาและถูกประเมินค่าโดยใช้วิธีคัดเลือกแบบวงล้อรูเล็ตในการเลือกโครโมโซมไปเป็นต้นแบบของประชากรรุ่นต่อไป แล้วใช้ค่าความน่าจะเป็นและฐานความรู้เป็นเกณฑ์ในการเลือกใช้กระบวนการทางพันธุกรรมตั้งแต่ 1 ตัวขึ้นไปกับโครโมโซมที่ทำการคัดเลือกไว้ นำโครโมโซมผลลัพธ์ที่ผ่านกระบวนการต่างๆมาแทนประชากรเดิม และนำโครโมโซมที่มีคุณภาพดีที่สุดของประชากรรุ่นเดิมมาแทนโครโมโซมที่มีคุณภาพน้อยที่สุดในรุ่น ได้เป็นประชากรรุ่นใหม่และจะทำการตรวจสอบโครโมโซมของทุกรุ่นเพื่อเก็บโครโมโซมที่มีคุณภาพดีที่สุดเอาไว้ ซึ่งจะกลายเป็นผลลัพธ์สุดท้ายเมื่อเกิดเหตุที่ทำให้การทำงานสิ้นสุดลง ส่วนหลักเกณฑ์ที่ทำให้การหาคำตอบสิ้นสุดลงจะเกิดจากมีจำนวนรุ่นเกินค่าที่กำหนดหรือเป็นเพราะคุณภาพของคำตอบที่ดีที่สุดจากหลายๆรุ่นมีค่าเท่าๆกัน และจะเริ่มต้นทำกระบวนการต่างๆที่กล่าวมาข้างต้นซ้ำอีกเพื่อหาคำตอบที่ถูกปรับปรุงแล้วในรุ่นต่อๆมา แต่ถ้าเป็นการหาคำตอบในสภาพแวดล้อมที่สภาพแวดล้อมเปลี่ยนแปลงได้ (Dynamic Environment) จะต้องเพิ่มขั้นตอนสำหรับตรวจสอบการเปลี่ยนแปลงของสภาพแวดล้อมอยู่เสมอ ถ้าสภาพแวดล้อมเปลี่ยน ก็จะมีการรับข้อมูลของสภาพแวดล้อมที่เปลี่ยนแปลงไป แล้วนำไปประเมินคุณภาพของประชากรใหม่อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 โครงสร้างการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้
เพื่อวางแผนเส้นทางให้กับหุ่นยนต์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบระบบส่วนที่โครงงานพัฒนาขึ้น

4.1 การออกแบบวิธีการหาคำตอบ

เนื่องจากโครงงานนี้เป็นการปรับปรุงประสิทธิภาพในการหาคำตอบด้วยเจเนติกอัลกอริธึมที่มีฐานความรู้ แนวทางการทำงานส่วนใหญ่จึงคล้ายกับการหาคำตอบในเอกสารอ้างอิงเดิม ปัญหาที่ต้องการหาคำตอบคือ เส้นทางจากจุดเริ่มต้นไปยังจุดเป้าหมายที่กำหนดโดยไม่มีส่วนใดส่วนหนึ่งของเส้นทางตัดผ่านอุปสรรค และเป็นเส้นทางที่มีประสิทธิภาพในการแก้ปัญหามากที่สุด การพิจารณาว่าแต่ละเส้นทางมีประสิทธิภาพมากน้อยแค่ไหนสามารถพิจารณาได้จากระยะทางทั้งหมดของเส้นทาง และบริเวณที่เส้นทางตัดผ่านอุปสรรค

4.1.1 การออกแบบวิธีการแทนค่าปัญหา

การแทนค่าต้องแบ่งประเภทของตัวแปรที่ต้องพิจารณาออกเป็น 2 ประเภท ซึ่งรายละเอียดการแทนค่าของตัวแปรต่างๆเป็นดังนี้

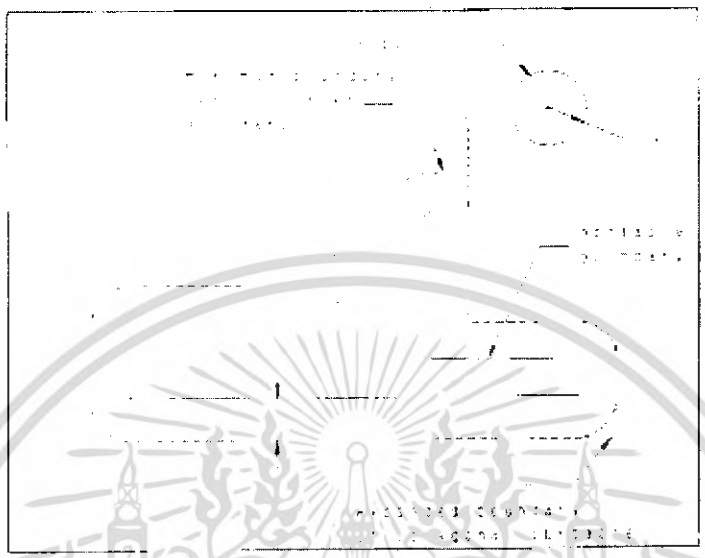
4.1.1.1 การออกแบบคุณสมบัติของสภาพแวดล้อม

ตัวแปรที่มีความเกี่ยวข้องกับการแก้ปัญหา แต่ไม่ถูกนำไปใช้ในการแก้ปัญหา เรียกว่า สภาพแวดล้อม ได้แก่ ตารางทั้งหมดที่แบ่งออกเป็นช่องขนาดเท่าๆกัน ทั้งบริเวณที่หุ่นยนต์เคลื่อนที่ได้ และบริเวณที่เป็นอุปสรรค ในแต่ละช่อง (Node) มีหมายเลขเป็นจำนวนเต็มกำกับอยู่แบบเรียงลำดับจากน้อย (ด้านซ้ายล่าง) ไปหามาก (ด้านขวาบน) เริ่มตั้งแต่ 0 เพิ่มขึ้นเรื่อยๆจนถึงช่องสุดท้าย ขึ้นอยู่กับจำนวนช่องของตารางในขณะนั้น

บริเวณที่เป็นอุปสรรคบนตาราง เกิดจากการสร้างวงกลมที่มีขนาดพอดีกับขนาดจริงของหุ่นยนต์ดังที่แสดงบริเวณมุมขวาบนของรูปที่ 4.1 โดยรัศมีของวงกลม (*r*) จะต้องรวมระยะปลอดภัย (Safety Distance) หรือบริเวณที่อาจทำให้เกิดการชนกับอุปสรรค เนื่องจากการเคลื่อนที่ของหุ่นยนต์ผิดพลาด (Robot Control Error) ด้วย แล้วนำค่า r ที่ได้มาเพิ่มขนาดให้กับแต่ละด้านของขนาดอุปสรรคเดิม ทำให้หุ่นยนต์เคลื่อนที่ไปตามเส้นทางที่ได้จากการหาคำตอบด้วยเจเนติกอัลกอริธึมที่มีฐานความรู้ได้แม่นยำและถูกต้องมากยิ่งขึ้น ไม่เกิดเหตุการณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์เคลื่อนที่ไปตามเส้นทางแล้วชนกับอุปสรรคอย่างแน่นอน ในรูปที่ 4.1 จะแสดงตัวอย่างการกำหนดขอบเขตของอุปสรรค โดยเส้นทึบแสดงขอบเขตเดิมของอุปสรรค เส้นปะแสดงขอบเขตของอุปสรรคหลังเพิ่มระยะปลอดภัยแล้ว



รูปที่ 4.1 แสดงวิธีการกำหนดขนาดของอุปสรรคที่จะใช้บนตาราง

และเนื่องจากการเพิ่มขนาดของอุปสรรคดังกล่าว ทำให้บริเวณขอบของอุปสรรคเกิดมุมโค้งที่มีรัศมีเท่ากับ r เพื่อให้สะดวกในการคำนวณจึงต้องขยายขอบเขตของอุปสรรคให้พอดีกับเส้นตารางก่อนจะนำขนาดของอุปสรรคไปใช้จริง ทำให้สามารถมองการเคลื่อนที่ของหุ่นยนต์ที่มีขนาดเสมือนเป็นการเคลื่อนที่ของจุดไปตามเส้นทางที่พิจารณาได้ ตามรูปตัวอย่างที่ 4.1

4.1.1.2 การออกแบบคุณสมบัติของ โครโมโซม

โครโมโซมใช้แทนเส้นทางการเคลื่อนที่ของหุ่นยนต์ โดยให้แต่ละยีนส์แทนด้วยเลขจำนวนเต็ม 1 ค่าซึ่งนำมาจากหมายเลขช่องบนตารางที่ไม่เป็นอุปสรรคและไม่ซ้ำกับหมายเลขช่องที่มีอยู่แล้วในโครโมโซม เริ่มต้นด้วยยีนส์ที่ระบุหมายเลขช่องต้นทาง แล้วเชื่อมเส้นทางกับยีนส์อันถัดไป เป็นเช่นนี้ไปเรื่อยๆ จนถึงยีนส์อันสุดท้ายที่ระบุหมายเลขช่องปลายทาง ความยาวของแต่ละโครโมโซมเกิดจากการสุ่มค่าระหว่าง 2 ถึง จำนวนช่องที่ไม่เป็นอุปสรรคหารด้วย 2 ซึ่งแตกต่างจากเอกสารอ้างอิงที่มีขนาดได้มากเท่ากับจำนวนช่องทั้งหมดในตาราง เนื่องจากในโครงงานนี้ได้สร้างกระบวนการทางพันธุกรรมใหม่ที่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มความยาวของโครโมโซมโดยการใส่ยีนส์จำนวนหนึ่งที่เพิ่มค่าความเหมาะสมของโครโมโซมได้ด้วยซึ่งจะกล่าวถึงรายละเอียดในหัวข้อเกี่ยวกับกระบวนการทางพันธุกรรม เรื่องกระบวนการซ่อมแซมเส้นทาง (Linerepair operator) ดังนั้นในแต่ละโครโมโซมจึงมีความยาวไม่เท่ากัน

4.1.2 การออกแบบวิธีประเมินค่า (Evaluation)

ถ้ามี 2 เส้นทางที่ไม่เหมือนกันและไม่ชนกับสิ่งกีดขวางตั้งแต่ต้นทางถึงปลายทางทั้งคู่ หากเราพิจารณาประสิทธิภาพของเส้นทางจากการชนอุปสรรคเพียงอย่างเดียวจะไม่สามารถเห็นความแตกต่างของ 2 เส้นทางนี้ได้ จึงต้องใช้สมการคำนวณเพื่อประเมินค่าช่องต่างๆในเส้นทาง เพื่อช่วยแยกประเภทของเส้นทางระหว่างเส้นทางที่เหมาะสมกับเส้นทางที่ไม่เหมาะสม และช่วยบอกความแตกต่างของคุณภาพเส้นทางแต่ละเส้นด้วย โดยมีรายละเอียดดังนี้

$$F_{\cos t} = \sum_{i=1}^{N} (d_i + \beta_i C) \tag{4.1}$$

โดย i คือ ลำดับของเส้นที่เชื่อมระหว่างช่อง

N คือ จำนวนเส้นที่เชื่อมระหว่างช่องในเส้นทางนั้น

d_i คือ ระยะทางแบบ Euclidean (Euclidean's Distance) ระหว่างช่องที่อยู่ติดกัน

C คือ ค่าคงที่

β_i คือ ค่าสัมประสิทธิ์แทนบริเวณที่เส้นทางทับอุปสรรค ซึ่งหาได้จาก

$$\beta_i = \begin{cases} 0 & \text{เมื่อส่วนของเส้นที่ i นั้นเหมาะสมแล้ว} \\ \sum_{j=1}^{M} \alpha_j & \text{เมื่อส่วนของเส้นที่ i นั้นตัดผ่านสิ่งกีดขวาง} \end{cases} \tag{4.2}$$

โดย M คือ จำนวนของสิ่งกีดขวางที่ส่วนของเส้นตัดผ่าน

α_j คือ จำนวนช่องที่ส่วนของเส้นทางนั้นตัดผ่านสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่ายิ่งโครโมโซมมีส่วนเส้นทางที่ตัดผ่านอุปสรรคมากเท่าไหร่ ค่าความเหมาะสมยิ่งมากขึ้นเท่านั้น แสดงว่าโครโมโซมที่มีค่าความเหมาะสมน้อย มีส่วนเส้นทางตัดกับอุปสรรคน้อยหรือไม่มีเลย จะเป็นโครโมโซมที่ระบบต้องการมากกว่า

และเพื่อให้ใช้เวลาในการหาคำตอบลดลง ข้อมูลบางอย่างที่ได้จากการประเมินค่าจะต้องถูกบันทึกไว้เป็นคุณสมบัติของโครโมโซม เมื่อต้องการใช้ข้อมูลเหล่านี้เป็นฐานข้อมูลในกระบวนการต่างๆจะได้นำค่ามาใช้ได้ทันที ไม่ต้องทำการคำนวณซ้ำอีก ซึ่งข้อมูลจำพวกนี้ ได้แก่ โครโมโซมที่ดีที่สุดในรุ่น, โครโมโซมที่ดีที่สุดตั้งแต่เริ่มหาคำตอบ, ค่าความเหมาะสมของโครโมโซม, จำนวนช่องที่ตัดผ่านอุปสรรค, ค่าความเหมาะสมของส่วนเส้นทางที่ตัดผ่านอุปสรรค, ตำแหน่งของยีนส์ที่ทำให้เกิดเส้นทางตัดผ่านอุปสรรค

4.1.3 การออกแบบวิธีการเลือกโครโมโซมต้นแบบ (Selection Parent Chromosomes)

ทำการสุ่มเลือกโครโมโซมในประชากรต้นแบบเพื่อนำไปผ่านกระบวนการต่างๆ โดยไม่พิจารณาค่าความเหมาะสม เนื่องจากการทำงานในส่วนอื่นๆของระบบ มีการคัดเลือกคุณภาพของโครโมโซมอยู่แล้ว ขั้นตอนนี้จึงเปิดโอกาสให้โครโมโซมมีความหลากหลายมากขึ้น เนื่องจากโครโมโซมที่มีบางส่วนตัดผ่านอุปสรรคก็มีโอกาสกลายเป็นโครโมโซมที่มีประสิทธิภาพในแก้ปัญหาได้

4.1.4 การออกแบบกระบวนการทางพันธุกรรม (Genetic Operators)

การหาคำตอบแบบเดิมมีการใช้กระบวนการทางพันธุกรรม (Genetic Operators) แบบการกลายพันธุ์ (Mutation Operator) ที่มักไม่ก่อให้เกิดการปรับปรุงคุณภาพของโครโมโซมเป็นส่วนใหญ่ เนื่องจากเป็นการสุ่มเลือกค่าใหม่จากค่าทั้งหมดที่เป็นไปได้ในตาราง จึงเปลี่ยนเป็นการใช้กระบวนการที่ทำให้เกิดการกลายพันธุ์ภายในโครโมโซมเดียวกัน คือ กระบวนการสลับที่ (Swap Operator) จะทำการสลับค่ายีนส์สองตำแหน่งในโครโมโซมเดียวกัน ซึ่งรับประกันว่าค่าใหม่ที่ได้จากการกลายพันธ์เป็นค่าที่ไม่อยู่บนอุปสรรค และเนื่องจากเริ่มแรกแต่ละค่าในยีนส์เกิดจากการสุ่มเลือกขึ้นมา ต้องผ่านกระบวนการปรับปรุงและคัดเลือกมาแล้วมากมาย แต่การที่ยีนส์นั้นยังคงประกอบอยู่ในโครโมโซมแสดงว่าเป็นค่าที่เหมาะสมในการแก้ปัญหาอยู่บ้าง การกลายพันธุ์โดยใช้ค่าภายในโครโมโซมเดียวกันจึงเป็นการกำหนดการกลายพันธุ์ให้อยู่ในขอบเขตของค่าที่เหมาะสมจะเป็นยีนส์ภายในโครโมโซมด้วย

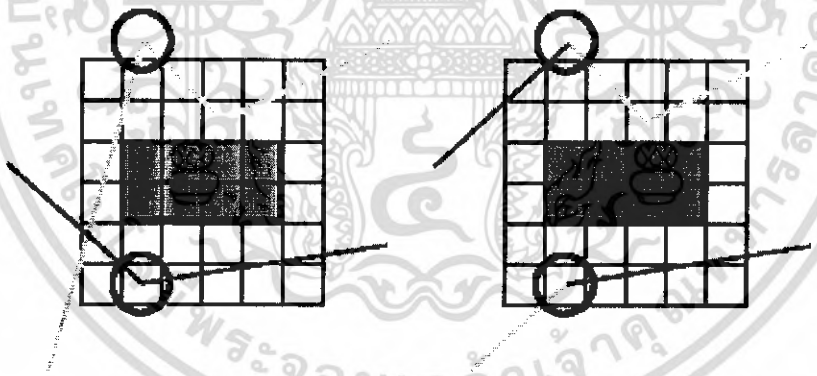
ในกระบวนการพิเศษ (Specialized Operator) ที่สร้างขึ้นมาเพื่อแก้ปัญหาเกี่ยวกับการหาเส้นทางโดยเฉพาะจะไม่มี NodeRepair Operator เหมือนในเอกสารอ้างอิงเนื่องจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยีนส์ในโครโมโซมทุกตัวต้องเป็นหมายเลขช่องที่ไม่เป็นอุปสรรคอยู่แล้ว ส่วนกระบวนการอื่นทั้งสาม คือ Linerepair Operator, Delete Operator และ Improve Operator ยังมีแนวการทำงานแบบเดิม แตกต่างกันเพียงบางรายละเอียด ซึ่งแต่ละกระบวนการมีรายละเอียดดังนี้

4.1.4.1 กระบวนการมาตรฐาน

1. **Crossover** คือ กระบวนการที่จะทำการสับเปลี่ยนชิ้นส่วนของโครโมโซมโดยการสุ่มเลือก 1 ยีนส์ มาจากโครโมโซมต้นแบบทั้งสองอัน แล้วแลกเปลี่ยนชิ้นส่วนที่อยู่หลังจาก 2 ยีนส์ที่ถูกเลือก ซึ่งจะได้โครโมโซมผลลัพธ์ที่มีจำนวนเท่ากับจำนวนของโครโมโซมต้นแบบ แล้วตรวจตรวจสอบภายในแต่ละโครโมโซมผลลัพธ์เพื่อลบบางส่วนของเส้นทางที่อาจใช้ยีนส์ที่มีค่าเดียวกัน และเนื่องจากยีนส์ภายในส่วนที่ทำการสับเปลี่ยนชิ้นส่วนและความยาวของส่วนที่ทำการสับเปลี่ยนชิ้นส่วนในแต่ละครั้งมีความแตกต่างกัน ทำให้โครโมโซมผลลัพธ์ที่เกิดขึ้นมีความยาวแตกต่างกัน ผลลัพธ์ที่ได้จึงมีประสิทธิภาพในการหาคำตอบมากกว่า

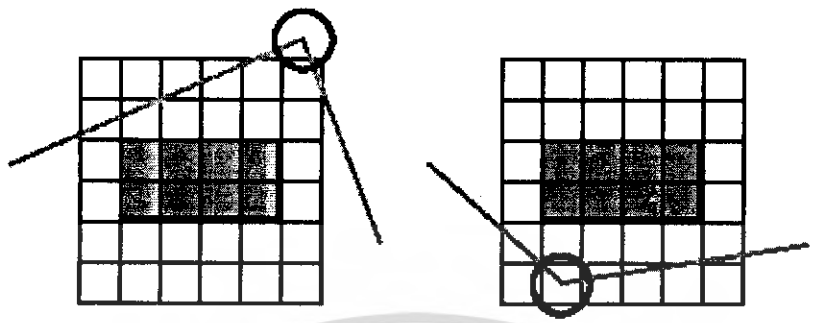


รูปที่ 4.2 แสดงการทำงานของ Crossover Operator

2. **Mutation** คือ กระบวนการที่จะสุ่มเลือก 1 ยีนส์จากโครโมโซมเพื่อเปลี่ยนค่าของยีนส์เป็นค่าใหม่ที่ไม่ซ้ำกับค่าเดิมของยีนส์และไม่ใช่ค่าของยีนส์ที่มีอยู่ในเส้นทางแล้ว การกลายพันธุ์เป็นปัจจัยหลักที่ทำให้ประชากรมีความหลากหลาย จึงไม่จำเป็นว่าผลลัพธ์ที่ได้จากการทำกระบวนการนี้จะต้องมีประสิทธิภาพดีขึ้น หากต้องการเพิ่มความหลากหลายให้กับแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

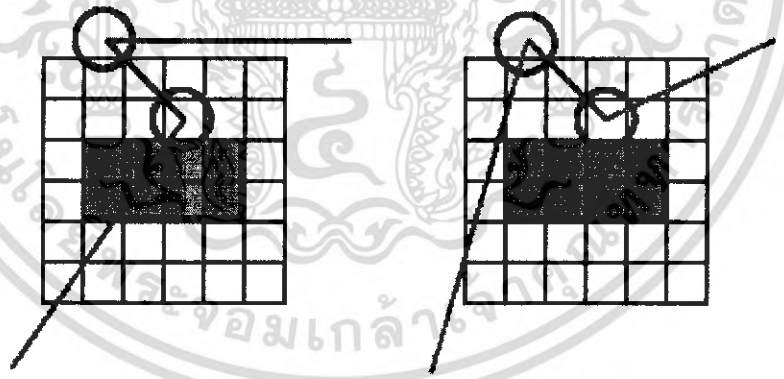
โครโมโซมในประชากรสามารถทำได้โดยเพิ่มค่าความเป็นไปได้ของการทำกระบวนการกลายพันธุ์ให้มากขึ้น



รูปที่ 4.3 แสดงการทำงานของ Mutation Operator

4.1.4.2 กระบวนการเฉพาะทาง (Specialized Operator)

1. **Swap** คือ กระบวนการที่จะทำการสลับค่าของยีนส์ 2 ตำแหน่งภายในโครโมโซม โดยโครโมโซมที่ผ่านกระบวนการนี้ได้ต้องมีขนาดมากกว่า 2 ยีนส์ขึ้นไป

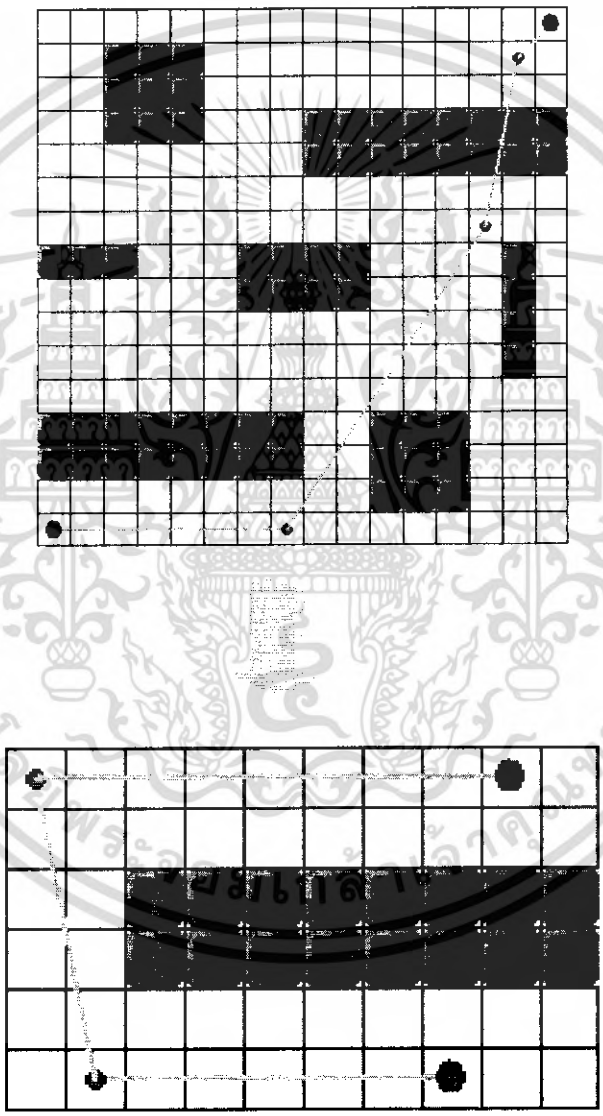


รูปที่ 4.4 แสดงการทำงานของ Swap Operator

2. **Line_repair** คือ กระบวนการที่ปรับปรุงส่วนของเส้นทางที่ตัดผ่านอุปสรรคด้วยการใช้เจเนติกอัลกอริธึมโดยระบุต้นทางและปลายทางเป็นยีนส์ 2 ตำแหน่งที่ทำให้เกิดเส้นทางที่ตัดผ่านอุปสรรค และทำการค้นหาเส้นทางที่สามารถหลบหลีกอุปสรรคในสภาพแวดล้อมที่มีขนาดเท่ากับความกว้างและความยาวมากที่สุดเมื่อเปรียบเทียบกันระหว่างตำแหน่งยีนส์

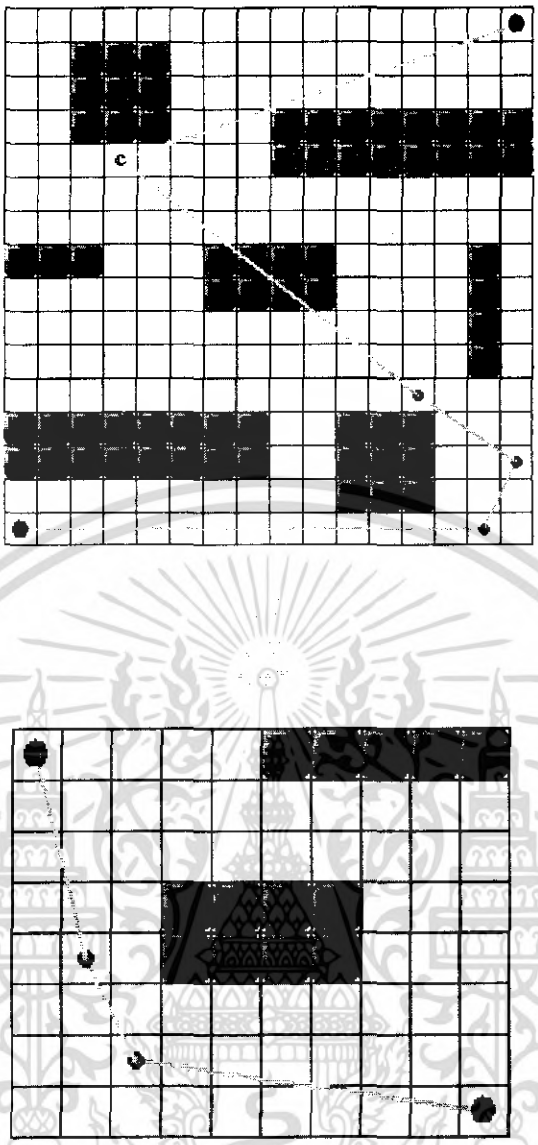
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้นทาง, ตำแหน่งยีนส์ปลายทาง, ความกว้าง และความยาวของอุปสรรค บวกเพิ่มอีก 4 ช่อง เช่น ในรูปที่ 4.5 แสดงการค้นหาเส้นทางที่ไม่ตัดผ่านอุปสรรคด้วยเจเนติกอัลกอริธึม ในกรณีที่ขนาดของอุปสรรคบวกเพิ่มอีก 4 ช่องมีใหญ่กว่าความกว้างและความยาวของตำแหน่งของยีนส์ต้นทางและปลายทาง และในรูปที่ 4.6 แสดงการค้นหาเส้นทางในกรณีที่ความกว้างและความยาวของตำแหน่งของยีนส์ต้นทางและตำแหน่งยีนส์ปลายทางมีขนาดใหญ่กว่าขนาดของอุปสรรคที่บวกเพิ่มอีก 4 ช่อง เป็นต้น



รูปที่ 4.5 แสดงการค้นหาเส้นทางในขอบเขตที่มีขนาดเท่ากับขนาดของอุปสรรคบวกเพิ่มอีก 4 ช่อง

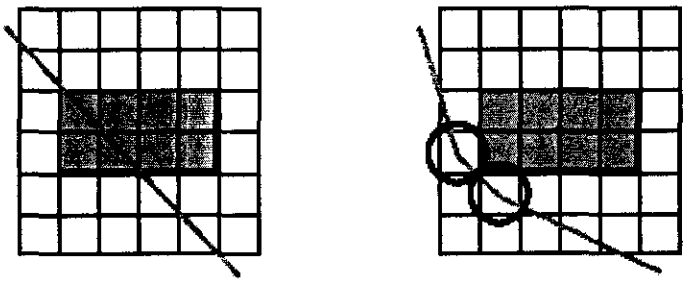
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงการค้นหาเส้นทางในขอบเขตที่มีขนาดเท่ากับความกว้างและยาวของตำแหน่งยีนส์ต้นทางและยีนส์ปลายทาง

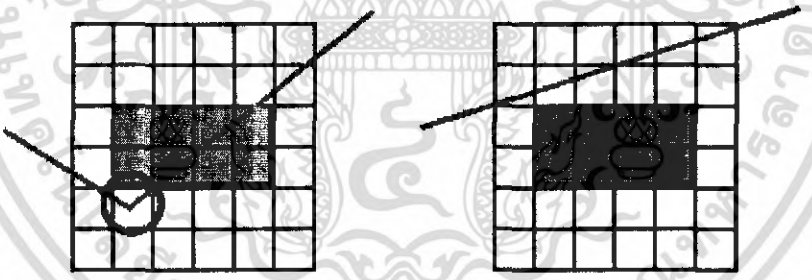
ในแต่ละรุ่นมีจำนวนประชากรเท่ากับ 6 โครโมโซม แต่ละโครโมโซมมีความยาวไม่คงที่ตั้งแต่ 2 ถึง 5 ยีนส์ ใช้กระบวนการทางพันธุกรรมแบบมาตรฐานคือ กระบวนการสับเปลี่ยนชิ้นส่วนและกระบวนการกลายพันธุ์ มีค่าความเป็นไปได้ในการทำกระบวนการสับเปลี่ยนชิ้นส่วนและกระบวนการกลายพันธุ์เท่ากับ 0.5 และ 0.9 ตามลำดับ และจะหยุดการทำงานเมื่อมีโครโมโซมที่ดีที่สุดซ้ำกันมากกว่า 3 รุ่นหรือทำงานมากกว่า 5 รุ่น ซึ่งถ้าผลลัพธ์ที่ได้จากการใช้ เจเนติกอัลกอริธึมมีค่าความเหมาะสมน้อยกว่าค่าความเหมาะสมของส่วนเส้นทางที่ตัดผ่านอุปสรรค ก็จะเพิ่มผลลัพธ์ดังกล่าวเข้าไปแทนที่ส่วนเส้นทางเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงการทำงานของ Linerepair Operator

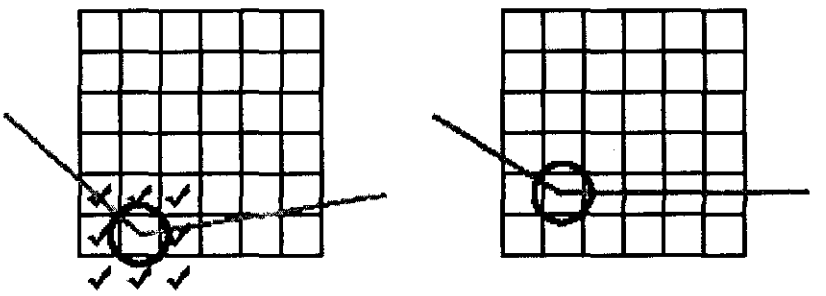
3. **Deletion** คือ กระบวนการที่ใช้ปรับปรุงคุณภาพของเส้นทางทั้งที่เป็นเส้นทางที่เหมาะสมแล้วและเป็นเส้นทางที่ยังมีบางส่วนที่ติดอยู่กับอุปสรรค โดยการสุ่มเลือก 1 ยีนส์ขึ้นมาจากโครโมโซม สมมุติให้ยีนส์ที่ถูกเลือกนี้โดนลบ แล้วลองเชื่อมเส้นทางระหว่าง 2 ยีนส์ที่อยู่ใกล้เคียงกับยีนส์ที่ถูกลบ ถ้าการเชื่อมเส้นทางทำให้ประสิทธิภาพของเส้นทางทั้งหมดดีขึ้น ให้ลบยีนส์ที่ถูกเลือกออกไปจริงๆ แล้วใช้เส้นทางใหม่ที่มีการเชื่อมเส้นทางแทน



รูปที่ 4.8 แสดงการทำงานของ Delete Operator

4. **Improvement** คือ กระบวนการที่ใช้กับเส้นทางที่ไม่มีส่วนใดอยู่บนอุปสรรคให้มีคุณภาพดีขึ้นอีก โดยการสุ่มเลือก 1 ช่องจากโครโมโซม แล้วใช้วิธีการค้นหาในขอบเขตจำกัดหาช่องที่ทำให้เส้นทางมีประสิทธิภาพดีขึ้นในบริเวณ 8 ช่องรอบข้างช่องที่ทำการสุ่มขึ้นมา แล้วย้ายช่องที่ถูกสุ่มเลือกไปยังตำแหน่งใหม่ แต่ถ้าการย้ายตำแหน่งทำให้ประสิทธิภาพของเส้นทางลดลงให้คงค่าเดิมของยีนส์ไว้

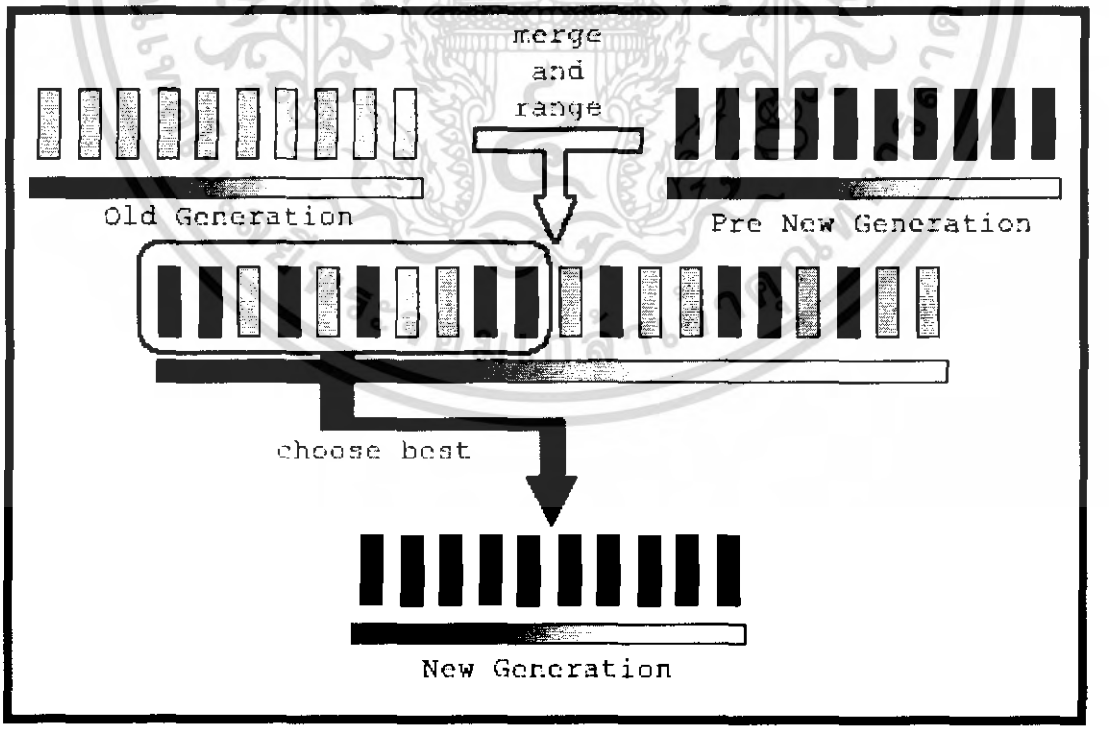
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงการทำงานของ Improve Operator

4.1.5 การออกแบบวิธีการคัดเลือกโครโมโซมไปเป็นประชากรรุ่นต่อไป (Representation)

ใช้วิธีคัดเลือกแบบจัดอันดับในการเลือกโครโมโซมไปเป็นประชากรต้นแบบรุ่นต่อๆไป โดยจะรวมโครโมโซมประชากรต้นแบบของรุ่นและประชากรที่ผ่านกระบวนการต่างๆมาแล้วเข้าด้วยกัน เรียงลำดับค่าความเหมาะสมจากน้อยไปหามาก จากนั้นจึงเลือกโครโมโซมที่มีลำดับอยู่ในจำนวนประชากรที่ต้องการไปเป็นประชากรต้นแบบของรุ่นต่อไป ซึ่งเป็นการคัดเฉพาะโครโมโซมที่ดีที่สุดในการหาคำตอบรอบนั้นไปเป็นประชากรต้นแบบของรุ่นต่อไป ดังรูปที่ 4.10



รูปที่ 4.10 แสดงการสร้างประชากรรุ่นใหม่ด้วยวิธีคัดเลือกแบบจัดอันดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

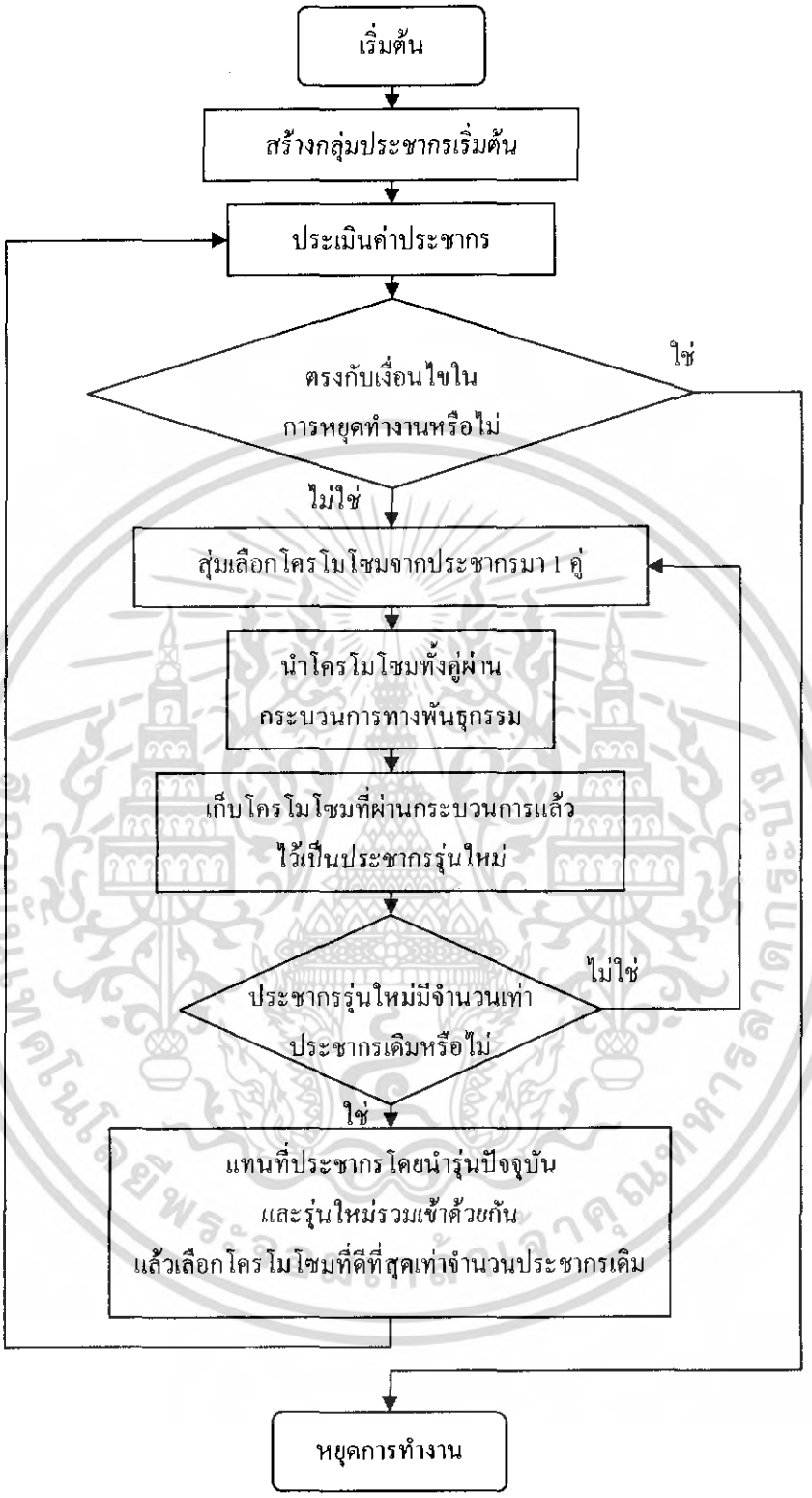
4.1.6 การออกแบบเงื่อนไขเพื่อหยุดการหาคำตอบ (Stop Criterion)

มีการกำหนดเงื่อนไขให้เจเนติกอัลกอริธึมรู้จักคำตอบที่ดีที่สุด โดยการเก็บโครโมโซมที่มีประสิทธิภาพที่ดีที่สุดของทุกรุ่นเอาไว้ ซึ่งโครโมโซมนี้จะกลายเป็นคำตอบของการแก้ปัญหาด้วยเจเนติกอัลกอริธึมที่มีฐานความรู้เมื่อมีโครโมโซมที่ไม่มีส่วนใดส่วนหนึ่งของเส้นทางตัดผ่านอุปสรรค และการทำงานเกินจำนวนรุ่นที่กำหนด หรือโครโมโซมที่ประสิทธิภาพดีที่สุดมีค่าความเหมาะสมคงที่ติดต่อกันหลายรุ่น

4.1.7 การออกแบบแผนผังการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้

แผนผังของการวางแผนเส้นทางด้วยเจเนติกอัลกอริธึมที่มีฐานความรู้ที่โครงงานพัฒนาขึ้นเริ่มต้นจากกรอบสี่เหลี่ยมด้านบนสุดของรูปที่ 4.11 ประชากรชุดแรกจะสุ่มสร้างขึ้นมาจากตำแหน่งในตารางที่ไม่เป็นอุปสรรคและถูกประเมินค่าความเหมาะสม สุ่มเลือกโครโมโซมเพื่อสร้างต้นแบบของประชากรก่อนนำไปผ่านกระบวนการต่างๆ แล้วใช้ค่าความเป็นไปได้และฐานความรู้เป็นเกณฑ์ในการเลือกใช้กระบวนการต่างๆ นำโครโมโซมที่ผ่านกระบวนการต่างๆมาแทนประชากรเดิมด้วยการคัดเลือกแบบจัดอันดับไปเป็นประชากรรุ่นใหม่ และจะเริ่มต้นทำกระบวนการต่างๆที่กล่าวมาข้างต้นซ้ำอีกจนกว่าจะได้คำตอบที่ต้องการ รวมทั้งตรวจสอบประชากรของทุกรุ่นเพื่อเก็บโครโมโซมที่มีคุณภาพดีที่สุดเอาไว้ ซึ่งจะกลายเป็นผลลัพธ์สุดท้ายเมื่อเกิดเหตุที่ทำให้การทำงานสิ้นสุดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงขั้นตอนการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้ในการวางแผนเส้นทางที่โครงงานพัฒนาขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบโครงสร้างการทำงาน

การทำงานของโปรแกรมเกิดจากการทำงานร่วมกัน 5 คลาส ซึ่งมีรายละเอียดดังต่อไปนี้

4.2.1 คลาส Grid

เป็นคลาสหลักของอีก 4 คลาสที่เหลือ มีการกำหนดข้อมูลและฟังก์ชั่นที่สำคัญในการคำนวนของคลาสอื่นๆไว้ในคลาสดังตารางด้านล่างนี้

ตารางที่ 4.1 แสดง Private Member Datas ของคลาส Grid

ประเภท	ชื่อ	ความหมาย
Int	gridX	จำนวนช่องของตารางในแนวแกน X
Int	gridY	จำนวนช่องของตารางในแนวแกน Y
Int	size	จำนวนช่องทั้งหมดของตาราง

ตารางที่ 4.2 แสดง Public Member Functions ของคลาส Grid

คืนค่า	ชื่อ	ค่าพารามิเตอร์	ความหมาย
void	Grid	void	ตั้งค่าเริ่มต้นของคลาส
void	~Grid	void	จบการทำงานคลาส
int	GetGridX	void	คืนค่าความกว้างตามแนวแกน X ของตาราง
int	GetGridY	void	คืนค่าความสูงตามแนวแกน Y ของตาราง
int	GetSize	void	คืนค่าจำนวนช่องทั้งหมดของตาราง
void	SetGridX	int	กำหนดความกว้างตามแนวแกน X ของตาราง
void	SetGridY	int	กำหนดความยาวตามแนวแกน Y ของตาราง
int	GetX	int	คืนค่าตำแหน่งในแนวแกน X ของจุด
int	GetY	int	คืนค่าตำแหน่งในแนวแกน Y ของจุด
Int	GetDistance	int , int	คืนค่าระยะทางระหว่างจุดสองจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vector<int>	GetLineVector	int , int	คืนค่าช่องที่ถูกเส้นทางระหว่างจุดสองจุดตัดผ่าน
Vector<int>	GetNeighbor	int	หา8 ช่องรอบข้างตำแหน่งระบุ
int*	GetArea	int , int , int	หาช่องที่อยู่ในบริเวณที่กำหนด โดยการระบุตำแหน่งเริ่มต้น ความกว้าง และความยาว

4.2.2 คลาส Ground

เป็นคลาสที่สืบทอดคุณสมบัติมาจากคลาส Grid ใช้เก็บข้อมูลเกี่ยวกับสภาพแวดล้อมของโปรแกรม โดยรายละเอียดของคลาสเป็นไปตามตารางด้านล่าง

ตารางที่ 4.3 แสดง Private Member Datas ของคลาส Ground

ประเภท	ชื่อ	ความหมาย
int	countObs	จำนวนอุปสรรค
bool*	Grid	ผังตารางสำหรับหาเส้นทาง
Obstacle*	ObsGrid	ข้อมูลของอุปสรรคในสภาพแวดล้อม

ตารางที่ 4.4 แสดง Public Member Functions ของคลาส Ground

คืนค่า	ชื่อ	ค่าพารามิเตอร์	ความหมาย
void	Ground	void	ตั้งค่าเริ่มต้นของคลาส
void	~Ground	void	จบการทำงานคลาส
void	AddObstacle	int , int , int	สร้างและใส่อุปสรรคเข้าไปในตาราง
bool*	GetGrid	void	คืนค่าผังของตาราง
int	GetNumObs	void	คืนค่าจำนวนของอุปสรรคในตาราง
Obstacle*	GetObstacle	void	คืนข้อมูลของอุปสรรคทั้งหมด
Void	ResetEnvironment	void	กำหนดให้สภาพแวดล้อมมีค่าเป็นค่าเริ่มต้นทั้งหมด
Void	CopyGround	Ground*	กำหนดให้สภาพแวดล้อมมีคุณลักษณะของอุปสรรคเหมือนกับสภาพแวดล้อมต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 คลาส Obstacle

เป็นคลาสที่สืบทอดคุณสมบัติมาจากคลาส Grid ใช้เก็บข้อมูลและฟังก์ชั่นการทำงานของอุปสรรคในสภาพแวดล้อม โดยรายละเอียดของคลาสเป็นไปตามตารางด้านล่าง

ตารางที่ 4.5 แสดง Private Member Datas ของคลาส Obstacle

ประเภท	ชื่อ	ความหมาย
int	point	ตำแหน่งแรกของอุปสรรค
int	width	ความกว้างในแนวแกน X ของอุปสรรค
int	height	ความกว้างในแนวแกน Y ของอุปสรรค
int	osize	ขนาดของอุปสรรค
int*	plain	หมายเลขช่องที่เป็นอุปสรรค

ตารางที่ 4.6 แสดง Public Member Functions ของคลาส Obstacle

คืนค่า	ชื่อ	ค่าพารามิเตอร์	ความหมาย
void	Obstacle	void	ตั้งค่าเริ่มต้นของคลาส
void	Obstacle	int , int , int	สร้างอุปสรรคโดยระบุตำแหน่งเริ่มต้น ความกว้างในแนวแกน X และความยาวในแกน Y
void	~Obstacle	void	จบการทำงานคลาส
int*	GetPlain	void	คืนค่าหมายเลขช่องที่เป็นอุปสรรค
int	GetPoint	void	คืนค่าตำแหน่งแรกของอุปสรรค
int	GetWidth	void	คืนค่าความกว้างในแนวแกน X ของอุปสรรค
int	GetHeight	void	คืนค่าความยาวในแนวแกน Y ของอุปสรรค
int	GetSize	void	คืนค่าขนาดของอุปสรรค
void	CopyObstacle	Obstacle*	กำหนดให้อุปสรรคมีคุณลักษณะเหมือนกับอุปสรรคต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 คลาส Genome

เป็นคลาสที่สืบทอดคุณสมบัติมาจากคลาส Grid ใช้เก็บข้อมูลและฟังก์ชั่นการทำงานของโครโมโซม โดยรายละเอียดของคลาสเป็นไปตามตารางด้านล่าง

ตารางที่ 4.7 แสดง Private Member Datas ของคลาส Genome

ประเภท	ชื่อ	ความหมาย
int	constant	ค่าคงที่ที่ใช้ในการคำนวณค่าความเหมาะสม
int	length	ความยาวของโครโมโซม
int	Fitness	ค่าความเหมาะสมของโครโมโซม
int	linealfa	จำนวนช่องของอุปสรรคที่โครโมโซมตัดผ่าน
int	SectionFitness	ค่าความเหมาะสมของส่วนที่เส้นทางตัดผ่านอุปสรรค
int	LineSegment	ตำแหน่งแรกที่ทำให้เส้นทางตัดผ่านอุปสรรค
vector<int>	GenesList	ค่าของโครโมโซม

ตารางที่ 4.8 แสดง Private Member Functions ของคลาส Genome

คืนค่า	ชื่อ	ค่าพารามิเตอร์	ความหมาย
int	CheckLine	int , int , bool*	คืนค่าจำนวนช่องที่ส่วนของเส้นทางตัดผ่านอุปสรรค
int	CheckLine	int , bool*	คืนค่าจำนวนช่องที่ส่วนของเส้นทางตัดผ่านอุปสรรค

ตารางที่ 4.9 แสดง Public Member Functions ของคลาส Genome

คืนค่า	ชื่อ	ค่าพารามิเตอร์	ความหมาย
void	Genome	void	ตั้งค่าเริ่มต้นของคลาส
void	~Genome	void	จบการทำงานคลาส
void	GetGenomeValue	void	แสดงค่าภายในโครโมโซม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

char*	GetGenomeStrin g	void	คืนค่าที่ใช้แสดงค่าภายในโครโมโซม
int	GetLength	void	คืนค่าความยาวของโครโมโซม
int	GetFitness	void	คืนค่าความเหมาะสม
int	GetLineSegment	void	คืนค่าจำนวนช่องในโครโมโซมที่ตัดกับอุปสรรค
vector<int>	GetGenesList	void	คืนเวคเตอร์ค่าภายในโครโมโซม
void	Initialize	int , int , int , bool*	สร้างโครโมโซมโดยกำหนดจุดเริ่มต้น จุดสุดท้ายของเส้นทาง และความยาวมากที่สุดของโครโมโซม(0 = ความยาวไม่คงที่)
int	CalculateFitness	bool*	คำนวณค่าความเหมาะสมของโครโมโซม
void	Mutate	int , bool*	เรียกการทำงาน Mutate Operator โดยการระบุตำแหน่ง
void	Crossover	int , int , Genome	เรียกการทำงาน Crossover Operator โดยการระบุตำแหน่งของตัวโครโมโซมเอง และตำแหน่งของโครโมโซมที่นำมาแลกเปลี่ยนชิ้นส่วน
void	Linerepair	Obstacle* , int , bool*	เรียกการทำงาน Line Repair Operator โดยการระบุตำแหน่งที่ตัดผ่านอุปสรรค และอุปสรรคที่ถูกเส้นทางตัดผ่าน
void	Deletion	int , bool*	เรียกการทำงาน Deletion Operator โดยการระบุตำแหน่ง
void	Improve	int , bool*	เรียกการทำงาน Improve Operator โดยการระบุตำแหน่ง
void	Swap	int , int	เรียกการทำงาน Swap Operator โดยการระบุ 2 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 แสดงฟังก์ชั่นที่เกี่ยวข้องกับคลาส Genome

คืนค่า	ชื่อ	ค่าพารามิเตอร์	ความหมาย
Genome	NewLocalSearch	int , int , bool*	เป็นการทำ local search ด้วยเจเนติกอัลกอริธึม โดยจะถูกเรียกใช้ในฟังก์ชั่น Linerepair

4.2.5 คลาส Population

เป็นคลาสที่สืบทอดคุณสมบัติมาจากคลาส Grid ใช้เก็บข้อมูลของประชากรและฟังก์ชั่นการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้ โดยรายละเอียดของคลาสเป็นไปตามตารางด้านล่าง

ตารางที่ 4.11 แสดง Private Member Datas ของคลาส Population

ประเภท	ชื่อ	ความหมาย
int	PopulationLimit	จำนวนประชากรในแต่ละรุ่น
int	Generation	หมายเลขรุ่นปัจจุบัน
int	BestGen	หมายเลขรุ่นที่ให้คำตอบที่ดีที่สุดรุ่นแรก
bool*	FreeGrid	ผังของตาราง
float	CrossoverProb	ค่าความเป็นไปได้ในการทำ Crossover Operator
float	SwapProb	ค่าความเป็นไปได้ในการทำ Swap Operator
float	MutateProb	ค่าความเป็นไปได้ในการทำ Mutate Operator
float	LineRepProb	ค่าความเป็นไปได้ในการทำ LineRepair Operator
float	DeleteProb	ค่าความเป็นไปได้ในการทำ Delete Operator
float	ImproveProb	ค่าความเป็นไปได้ในการทำ Improve Operator
Genome	BestInGeneration	โครโมโซมที่ดีที่สุดในรุ่นปัจจุบัน
Genome	BestSofar	โครโมโซมที่ดีที่สุดตั้งแต่เริ่มหาคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Genome*	Genomes	ประชากรที่ประกอบด้วยโครโมโซมต้นแบบ
Genome*	ResultGenomes	ประชากรที่ประกอบด้วยโครโมโซมที่ผ่านการกระบวนการต่างๆแล้ว

ตารางที่ 4.12 แสดง Public Member Functions ของคลาส Population

คืนค่า	ชื่อ	ค่าพารามิเตอร์	ความหมาย
void	Population	void	ตั้งค่าเริ่มต้นของคลาส
void	~Population	void	จบการทำงานคลาส
int	GetGenNumber	void	คืนค่าหมายเลขรุ่นของประชากรปัจจุบัน
void	GetCurrPopulation	void	แสดงโครโมโซมในประชากรต้นแบบ
void	GetResPopulation	void	แสดงโครโมโซมที่ผ่านกระบวนการต่างๆแล้ว
void	GetFitnessMax	void	แสดงค่าความเหมาะสมที่มากที่สุดในรุ่น
void	GetFitnessMin	void	แสดงค่าความเหมาะสมที่น้อยที่สุดในรุ่น
Genome	GetBestGenome	void	คืนค่าโครโมโซมที่ดีที่สุดในประชากร
void	SetAttribute	int,float,float, float, float, float, float	ตั้งค่าจำนวนประชากร ค่าความน่าจะเป็นของกระบวนการต่างๆทั้งหมด
void	DoInitialize	int , int , int , bool*	สร้างประชากรรุ่นแรกโดยการกำหนดจุดเริ่มต้น จุดสุดท้ายของเส้นทาง และความยาวมากที่สุดที่เป็นไปได้ของแต่ละโครโมโซม(0 = ความยาวไม่คงที่)
void	DoLineRepair	Obstacle* , int	เรียกการทำงาน Line Repair Operator โดยการระบุตำแหน่งที่ตัดผ่านอุปสรรคและอุปสรรคที่ถูกเส้นทางตัดผ่าน
void	DoDeletion	void	เรียกการทำงาน Deletion Operator
void	DoImprovement	void	เรียกการทำงาน Improve Operator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

void	DoCrossover	void	เรียกการทำงาน Crossover Operator
void	DoSwap	void	เรียกการทำงาน Swap Operator
void	DoMutation	void	เรียกการทำงาน Mutate Operator
void	DoCurrRanking	void	จัดเรียงประชากรต้นแบบ โดยเรียงตามค่าความเหมาะสมจากน้อยไปหามาก
void	DoResRanking	void	จัดเรียงประชากรที่ผ่านกระบวนการต่างๆแล้ว โดยเรียงตามค่าความเหมาะสมจากน้อยไปหามาก
void	DoSelection	void	คัดเลือกโครโมโซมจากประชากรต้นแบบไปปรับปรุงคุณภาพ
void	DoEvaluation	void	คำนวณค่าความเหมาะสมของแต่ละโครโมโซมในประชากรที่ผ่านกระบวนการต่างๆแล้ว
void	DoRepresentation	void	คัดเลือกโครโมโซมจากประชากรที่ผ่านกระบวนการต่างๆแล้ว ให้เป็นประชากรต้นแบบในรุ่นต่อไป

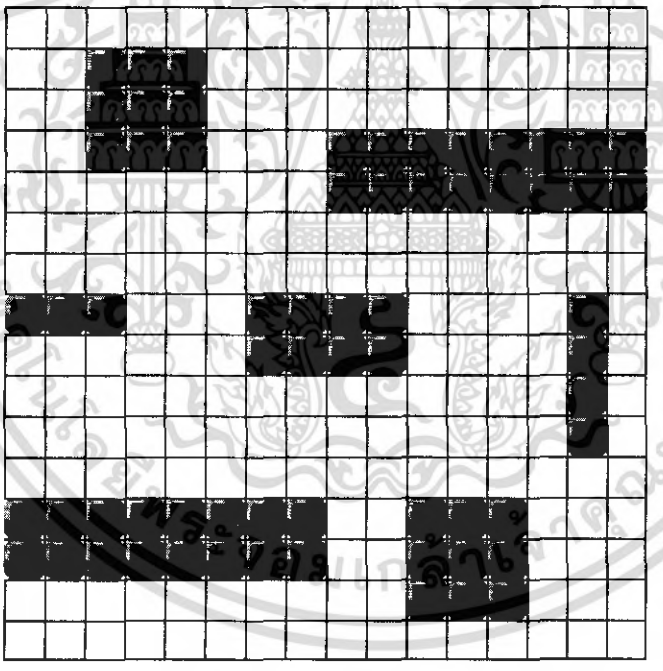
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

5.1 การทดลองเพื่อหาค่าที่เหมาะสมของตัวแปรต่างๆ

เมื่อต้องการหาค่าของตัวแปรใดตัวแปรหนึ่งที่เหมาะสมกับการทำงานของระบบมากที่สุด จะต้องกำหนดให้ตัวแปรตัวอื่นๆ ได้แก่ สภาพแวดล้อม, ตัวแปรที่เกี่ยวกับค่าความเป็นไปได้ในการทำกระบวนการต่างๆ , จำนวนประชากร, ขนาดของโครโมโซม และค่าคงที่ที่ใช้ในการหาค่าความเหมาะสมมีค่าคงที่ แล้วทดลองกำหนดให้ตัวแปรที่เราสนใจมีค่าต่างๆกัน แล้วพิจารณาผลลัพธ์ที่ได้จากระบบว่ามีความสัมพันธ์กับค่าของตัวแปรอย่างไร โดยสภาพแวดล้อมของหุ่นยนต์เป็นดังรูปที่ 5.1 และให้ระบบทำงาน 10 รอบต่อการพิจารณาค่าของตัวแปร 1 ค่า เพื่อให้ผลลัพธ์มีความน่าเชื่อถือ



รูปที่ 5.1 แสดงสภาพแวดล้อมที่ใช้ในการหาค่าตัวแปรที่เหมาะสม

5.1.1 หาค่าความเป็นไปได้ในการทำกระบวนการทางพันธุกรรมต่างๆที่เหมาะสม

กำหนดให้ค่าความเป็นไปได้ของการทำกระบวนการอื่นๆที่ไม่ได้ทำการพิจารณาในรอบนั้นมีค่าเท่ากับ 0.5 มีจำนวนประชากรเท่ากับ 50 โครโมโซม มีความยาวมากที่สุดของโครโมโซมไม่เกินครึ่งหนึ่งของจำนวนช่องที่ไม่เป็นอุปสรรคในตาราง และค่าคงที่ที่ใช้ในการหาค่าความเหมาะสมเท่ากับ 20 ผลลัพธ์ที่ได้เป็นดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 แสดงผลลัพธ์ที่ได้จากกำหนดค่าต่างๆให้กับความเป็นไปได้ในการทำกระบวนการต่างๆ

กระบวนการทางพันธุกรรม / ค่าความเป็นไปได้		ค่าความเหมาะสมโดยเฉลี่ย	จำนวนรุ่นที่ได้คำตอบโดยเฉลี่ย	เวลาที่ใช้คำนวณโดยเฉลี่ย
1. Crossover Operator	0.1	29.8	171.7	34.56
	0.2	29.6	202.6	41.96
	0.3	29.78	199.897	39.69
	0.4	30.2	162.6	34.70
	0.5	29.8	175.3	38.39
	0.6	30.3	161	35.67
	0.7	30	189	42.10
	0.8	30	195.7	43.95
	0.9	29.9	174.5	40.04
2. Mutation Operator	0.1	29.8	157.6	29.80
	0.2	29.1	143.5	29.15
	0.3	29.6	172.4	34.74
	0.4	29.8	160.1	33.99
	0.5	29.7	191	41.49
	0.6	30.3	178.5	39.69
	0.7	30	215.5	48.70
	0.8	30.12	194	45.16
	0.9	30.9	171.8	40.15
3. Swap Operator	0.1	29.7	189.5	36.23
	0.2	29.8	167	32.48
	0.3	30.4	202.7	41.64
	0.4	29.8	174.2	37.17
	0.5	29.8	206.6	45.25
	0.6	30.2	166.2	37.32
	0.7	31.2	184.3	41.96
	0.8	30.56	200	45.49
	0.9	30.5	235	54.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Linerepair Operator	0.1	31.7	207.67	19.04
	0.2	31.1	207.20	26.15
	0.3	29.9	210.20	32.66
	0.4	29.4	211.4	38.83
	0.5	29.8	206.02	43.99
	0.6	29.3	185	45.37
	0.7	30	166.70	45.21
	0.8	27.5	166.78	49.7
	0.9	29.8	174.1	57.37
5. Delete Operator	0.1	30.2	207.8	43.98
	0.2	30	169.3	36.56
	0.3	29.9	201.6	43.5
	0.4	30	193	42.04
	0.5	29.9	171.4	36.89
	0.6	29.7	171.9	37.20
	0.7	29	201	43.80
	0.8	30	203	45.60
	0.9	30	188	40.20
6. Improve Operator	0.1	30.3	216	42.10
	0.2	29.7	253.9	50.69
	0.3	30.2	186.1	38.45
	0.4	29.7	185.7	38.75
	0.5	29.7	181.6	39.01
	0.6	29.5	214.4	47.761
	0.7	29.3	172.4	39.52
	0.8	29.9	172.8	40.17
	0.9	**29.5**	**160**	**39.34**

จากผลการทดลองจะได้ค่าความเป็นไปได้ในการทำกระบวนการที่เหมาะสมดังนี้

- ค่าความเป็นไปได้ของการทำ Crossover Operator เท่ากับ 0.4
- ค่าความเป็นไปได้ของการทำ Mutation Operator เท่ากับ 0.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ค่าความเป็นไปได้ของการทำ Swap Operator เท่ากับ 0.2
- ค่าความเป็นไปได้ของการทำ LineRepair Operator เท่ากับ 0.5
- ค่าความเป็นไปได้ของการทำ Delete Operator เท่ากับ 0.5
- ค่าความเป็นไปได้ของการทำ Improve Operator เท่ากับ 0.7

5.1.2 หาความยาวมากที่สุดของโครโมโซมที่เหมาะสม

กำหนดให้ N คือ จำนวนช่องที่ไม่เป็นอุปสรรคในตาราง มีค่าความเป็นไปได้ของการทำกระบวนการต่างๆเท่ากับ 0.5 มีจำนวนประชากรเท่ากับ 50 โครโมโซม และค่าคงที่ที่ใช้ในการหาค่าความเหมาะสมเท่ากับ 20 ผลลัพธ์ที่ได้เป็นดังตารางต่อไปนี้

ตารางที่ 5.2 แสดงผลลัพธ์ที่ได้จากกำหนดค่าต่างๆให้กับความยาวมากที่สุดของโครโมโซม

ค่าความยาวมากที่สุดของโครโมโซม	ค่าความเหมาะสมโดยเฉลี่ย	จำนวนรุ่นที่ได้คำตอบโดยเฉลี่ย	เวลาที่ใช้คำนวณโดยเฉลี่ย
N	29.5	185.5	40.156
$N/2$	29.3	213	45.62
$N/3$	29	215	44.93
$N/4$	30.2	184.2	38.95

จากการทดลองพบว่าความยาวมากที่สุดของโครโมโซมที่เหมาะสมในการแก้ปัญหา คือ หนึ่งในสามของจำนวนช่องที่ไม่เป็นอุปสรรคในตาราง

5.1.3 หาจำนวนประชากรที่เหมาะสม

กำหนดให้ค่าความเป็นไปได้ของการทำกระบวนการต่างๆเท่ากับ 0.5 มีความยาวมากที่สุดของโครโมโซมไม่เกินครึ่งหนึ่งของจำนวนช่องที่ไม่เป็นอุปสรรคในตาราง และค่าคงที่ที่ใช้ในการหาค่าความเหมาะสมเท่ากับ 20 ผลลัพธ์ที่ได้เป็นดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.3 แสดงผลลัพธ์ที่ได้จากกำหนดค่าต่างๆให้กับจำนวนประชากร

จำนวนประชากร	ค่าความเหมาะสมโดยเฉลี่ย	จำนวนรุ่นที่ได้คำตอบโดยเฉลี่ย	เวลาที่ใช้คำนวณโดยเฉลี่ย
10	29.7	169.3	106.9
30	29.7	162	106.17
50	29.4	163.7	89.625
70	29.4	182.7	97.79
90	29.3	168.8	89.45
110	29.5	145.8	77.05
130	29.8	143.7	76.69
150	29.7	151.7	78.48
170	29.8	160	86.05
190	28.7	154.3	81.74

จากการทดลองพบว่าจำนวนประชากรที่เหมาะสมในการแก้ปัญหา คือ 110 โครโมโซมต่อ 1 ประชากร

5.2 การทดลองเพื่อเปรียบเทียบประสิทธิภาพในการหาคำตอบระหว่างระบบเดิมกับระบบที่โครงงานพัฒนาขึ้น

การเปรียบเทียบผลลัพธ์ที่ได้จากการหาเส้นทางด้วยเจเนติกอัลกอริธึมแบบมีฐานความรู้ตามเอกสารอ้างอิง กับผลลัพธ์ที่ได้จากการหาเส้นทางที่โครงงานนำมาปรับปรุงแล้วในสภาพแวดล้อมที่อุปสรรคมีลักษณะต่างๆกัน จะต้องตั้งค่าให้ตัวแปรในระบบทั้งสองมีความคล้ายคลึงกันมากที่สุดเพื่อให้ผลการทดลองมีความน่าเชื่อถือและถูกต้อง โดยกำหนดจำนวนประชากรเท่ากับ 50 โครโมโซม ค่าความเป็นไปได้ในการทำกระบวนการต่างๆเท่ากับ 0.9 ยกเว้นกระบวนการกลายพันธุ์ให้มีค่าความเป็นไปได้เท่ากับ 0.2 และให้ตารางมีขนาดเท่ากับ 16 X 16 ช่อง มีค่าคงที่ในการคำนวณค่าความเหมาะสมเท่ากับ 20 และเนื่องจากทั้งสองระบบจะมีวิธีการคำนวนค่าความเหมาะสมเหมือนกันก็ต่อเมื่อค่าสัมประสิทธ์ของส่วนเส้นทางที่ตัดกับอุปสรรคมีค่าเป็นศูนย์ ดังนั้นจะสามารถเปรียบเทียบคุณภาพโครโมโซมที่ดีที่สุดของทั้งสองระบบได้ต่อเมื่อโครโมโซมทั้งสองไม่มีส่วนหนึ่งส่วนใดตัดกับอุปสรรค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.1 การวางแผนเส้นทางในกรณีที่อุปสรรคเป็นรูปตัวยู (U-shape Obstacle)

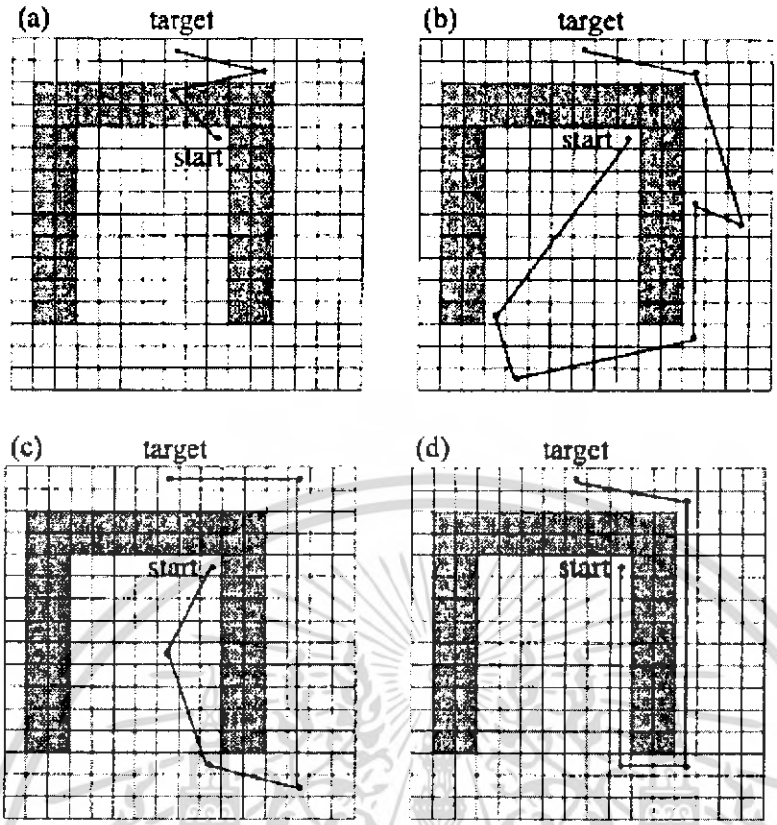
ผลลัพธ์ที่ได้จากระบบต้นแบบ :

เส้นทางที่ดีที่สุดในประชากรแรกมีค่าความเหมาะสมเท่ากับ 80.59 ซึ่งยังไม่มีลักษณะของเส้นทางที่เหมาะสมอยู่เลย เมื่อผ่านการทำงานไปจนถึงรุ่นที่ 8 จะได้เส้นทางดีที่สุดมีค่าความเหมาะสมเท่ากับ 40.02 เป็นเส้นทางที่ไปถึงปลายทางโดยไม่มีส่วนใดส่วนหนึ่งตัดกับอุปสรรค และเมื่อทำงานไปจนถึงรุ่นที่ 22 จะได้เส้นทางที่ดีที่สุดมีค่าความเหมาะสมเท่ากับ 33.98 แต่ยังไม่ใช่เส้นทางที่มีประสิทธิภาพดีที่สุด ระบบจึงทำงานต่อไปเรื่อยๆจนได้คำตอบเป็นเส้นทางที่ดีที่สุดในรุ่นที่ 30 มีค่าความเหมาะสมเท่ากับ 29.10 ดังรูปที่ 5.2

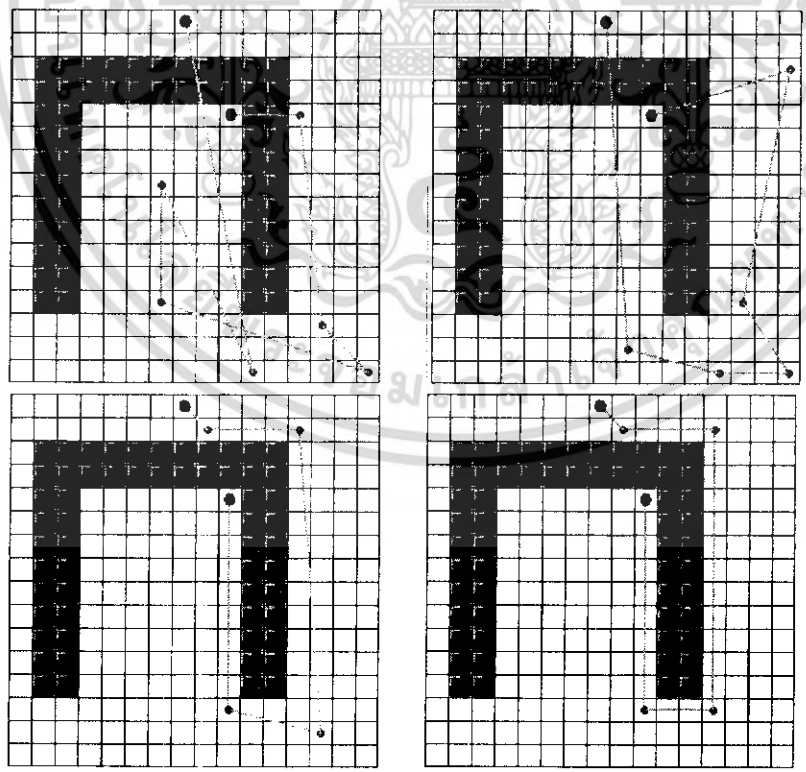
ผลลัพธ์ที่ได้จากระบบที่โครงงานพัฒนาขึ้น :

เส้นทางที่ดีที่สุดในประชากรแรกมีค่าความเหมาะสมเท่ากับ 150 ซึ่งยังไม่มีลักษณะของเส้นทางที่เหมาะสมอยู่เลย เมื่อผ่านการทำงานไปจนถึงรุ่นที่ 3 จะได้เส้นทางดีที่สุดมีค่าความเหมาะสมเท่ากับ 133 ซึ่งเป็นเส้นทางที่ไปถึงปลายทางโดยไม่มีส่วนใดส่วนหนึ่งตัดกับอุปสรรค และเมื่อทำงานไปจนถึงรุ่นที่ 19 จะได้เส้นทางที่ดีที่สุดมีค่าความเหมาะสมเท่ากับ 31 แต่ยังไม่ใช่เส้นทางที่มีประสิทธิภาพดีที่สุด ระบบจึงทำงานต่อไปเรื่อยๆจนได้คำตอบเป็นเส้นทางที่ดีที่สุดในรุ่นที่ 20 มีค่าความเหมาะสมเท่ากับ 29 ดังรูปที่ 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แสดงการหาเส้นทางด้วยระบบเดิมในสภาพแวดล้อมที่มีอุปสรรคเป็นรูปตัวยู



รูปที่ 5.3 แสดงการหาเส้นทางด้วยระบบที่โครงงานพัฒนาขึ้นในสภาพแวดล้อมที่มีอุปสรรคเป็นรูปตัวยู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 การวางแผนเส้นทางในกรณีที่สภาพแวดล้อมมีลักษณะซับซ้อน (Complex Environment) แบบที่ 1

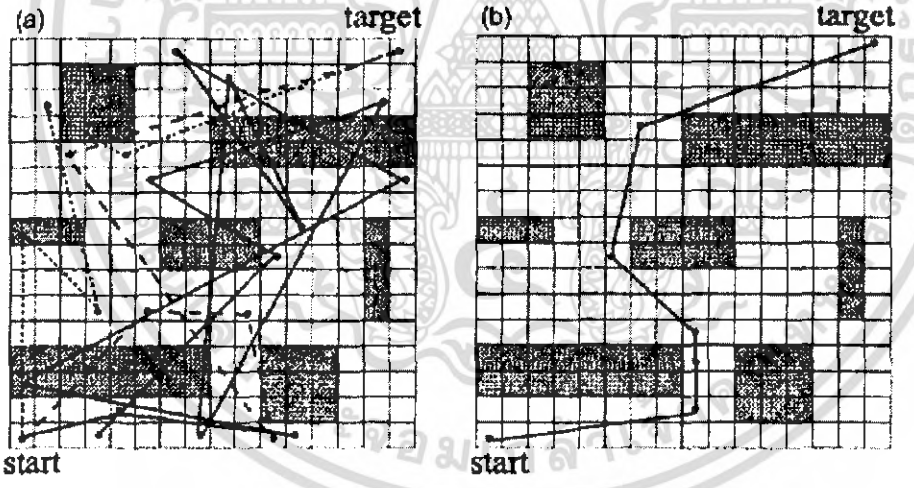
เป็นการดูผลลัพธ์ของระบบเมื่อสภาพแวดล้อมมีอุปสรรคหลายชิ้นและมีรูปทรงแตกต่างกันแบบแรก

ผลลัพธ์ที่ได้จากระบบต้นแบบ :

ในสภาพแวดล้อมแบบที่ 1 ประชากรในรุ่นแรกๆจะมีเส้นทางดีที่สุดที่ยังมีบางส่วนของเส้นทางตัดกับอุปสรรคอยู่ดังรูปที่ 5.4 (a) หลังจากให้ระบบทำงาน 20 ครั้งจะได้เส้นทางที่ดีที่สุดมีค่าความเหมาะสมเท่ากับ 30.85 มีค่าเบี่ยงเบนมาตรฐานเท่ากับ 0.67 ในรุ่นโดยเฉลี่ยเท่ากับ 257 มีค่าเบี่ยงเบนมาตรฐานเท่ากับ 123 ดังแสดงในรูปที่ 5.4 (b)

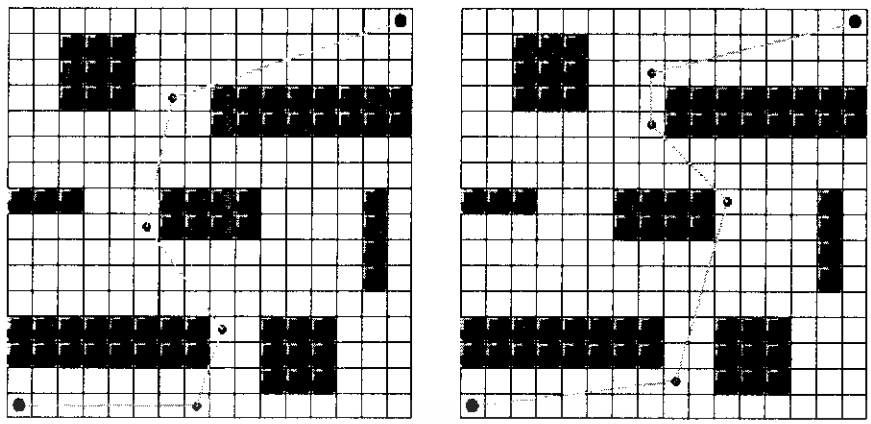
ผลลัพธ์ที่ได้จากระบบที่โครงงานพัฒนาขึ้น :

ในสภาพแวดล้อมแบบที่ 1 หลังจากให้ระบบทำงาน 20 ครั้งจะได้เส้นทางที่ดีที่สุดโดยเฉลี่ยในรุ่นที่ 156.7 มีค่าความเหมาะสมโดยเฉลี่ยเท่ากับ 29.45 และเส้นทางที่ได้จะมีลักษณะต่างๆกัน ดังรูปที่ 5.5



รูปที่ 5.4 แสดงการหาเส้นทางด้วยระบบเดิมในสภาพแวดล้อมที่ซับซ้อนแบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงการหาเส้นทางด้วยระบบที่โครงงานพัฒนาขึ้นในสภาพแวดล้อมที่ซับซ้อนแบบที่ 1

5.2.3 การวางแผนเส้นทางในกรณีที่สภาพแวดล้อมมีลักษณะซับซ้อน (Complex Environment) แบบที่ 2

เป็นการดูผลลัพธ์ของระบบเมื่อสภาพแวดล้อมมีอุปสรรคหลายชิ้นและมีรูปทรงแตกต่างกันแบบที่ 2

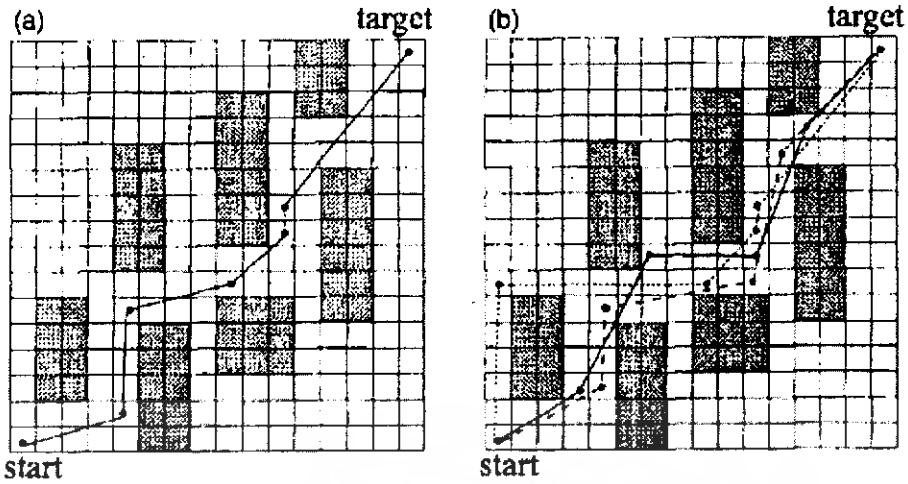
ผลลัพธ์ที่ได้จากระบบต้นแบบ :

ในสภาพแวดล้อมแบบที่ 2 หลังจากให้ระบบทำงานจะได้เส้นทางที่ดีที่สุดดังแสดงในรูปที่ 5.6 (a) และเมื่อทดลองให้ระบบทำงานหลายๆรอบจะได้เส้นทางที่ดีที่สุดมีลักษณะแตกต่างกันเล็กน้อยดังรูปที่ 5.6 (b)

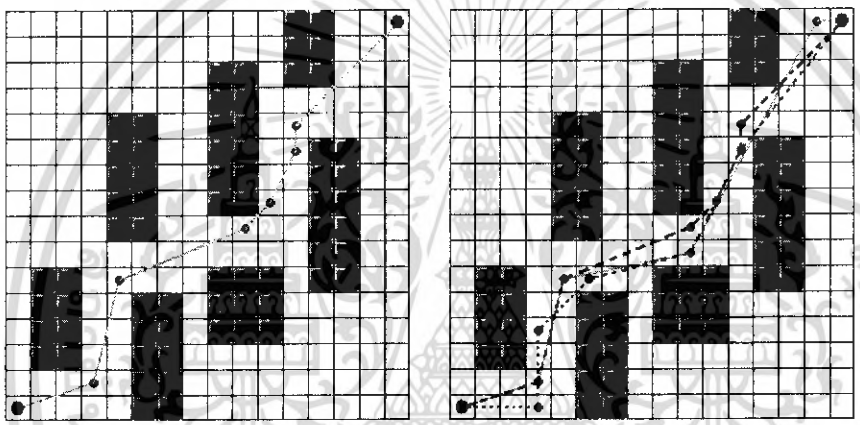
ผลลัพธ์ที่ได้จากระบบที่โครงงานพัฒนาขึ้น :

ในสภาพแวดล้อมแบบที่ 2 หลังจากให้ระบบทำงาน 20 ครั้งจะได้เส้นทางที่ดีที่สุดโดยเฉลี่ยในรุ่นที่ 60.45 และมีค่าความเหมาะสมเท่ากับ 24.75 และเส้นทางที่ได้มีลักษณะแตกต่างกันเล็กน้อยดังรูปที่ 5.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 แสดงการหาเส้นทางด้วยระบบเดิมในสภาพแวดล้อมที่ซับซ้อนแบบที่ 2



รูปที่ 5.7 แสดงการหาเส้นทางด้วยระบบที่โครงงานพัฒนาขึ้นในสภาพแวดล้อมที่ซับซ้อนแบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
บทวิจารณ์และสรุป

6.1 บทสรุป

การพัฒนาวิธีการวางแผนเส้นทางสำหรับหุ่นยนต์ในสภาพแวดล้อมที่มีอุปสรรคแบบต่างๆ ด้วยเจเนติกอัลกอริธึมที่ถูกปรับปรุงให้เหมาะกับการวางแผนเส้นทางโดยใช้ข้อมูลจากฐานความรู้ ทำได้โดยการสร้างให้ทุกยีนส์ของโครโมโซมไม่อยู่ในชิ้นส่วนของอุปสรรค พัฒนากระบวนการกลายพันธุ์ให้ใช้ค่าที่อยู่ภายในโครโมโซมเดียวกัน เปลี่ยนวิธีการค้นหาในขอบเขตจำกัดเป็นการใช้เจเนติกอัลกอริธึมแบบพื้นฐานให้ทำการปรับปรุงเฉพาะบางส่วนของเส้นทางที่ตัดกับอุปสรรค เปลี่ยนเครื่องมือในการคัดเลือกโครโมโซมเพื่อสร้างเป็นประชากรในรุ่นถัดไปให้เป็นการใช้วิธีการคัดเลือกแบบจัดอันดับ และทำการทดลองเพื่อหาค่าของตัวแปรต่างๆที่เอื้อต่อการทำงานของระบบมากที่สุด การค้นหาคำตอบในแต่ละรอบของกระบวนการทำงานของเจเนติกอัลกอริธึมที่มีฐานความรู้จึงสามารถพัฒนาคุณภาพของโครโมโซมหรือวิธีการแก้ปัญหาได้มากขึ้น ทำให้การใช้เวลาในการหาคำตอบลดลง ผลลัพธ์ที่ได้จากระบบยังมีประสิทธิภาพเท่าเดิมหรือมากกว่า มีความเชื่อถือได้มากขึ้น

6.2 วิจารณ์สิ่งที่ได้จากโครงงาน

6.2.1 เข้าใจวิธีการ และประโยชน์ของการพัฒนาเจเนติกอัลกอริธึมเพื่อแก้ปัญหาเฉพาะทาง เช่นการสร้างกระบวนการเฉพาะทาง หรือการใช้งานเจเนติกอัลกอริธึมร่วมกับฐานความรู้ในการแก้ปัญหาการวางแผนเส้นทางสำหรับหุ่นยนต์เคลื่อนที่ได้โดยเฉพาะ

6.2.2 เข้าใจแนวทางและประโยชน์ของการแก้ปัญหาด้วยเจเนติกอัลกอริธึมที่มีฐานความรู้ในการวางแผนเส้นทางสำหรับหุ่นยนต์ สามารถพัฒนาให้สามารถแก้ปัญหาได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ใช้จำนวนรอบในการทำงานลดลง มีค่าความเหมาะสมของผลลัพธ์สุดท้ายค่อนข้างคงที่ และมีค่าดีที่สุด

6.2.3 สามารถแยกแยะข้อดีข้อเสียของการแก้ปัญหาด้วยเจเนติกอัลกอริธึมกับวิธีอื่นๆได้

6.2.4 เข้าใจโครงสร้างของภาษา หลักการทำงาน และวิธีการเขียนโปรแกรมภาษา C++ รวมถึงการสร้าง MFC Application มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข

6.3.1 ช่วงแรกของการพัฒนา ระบบจะใช้เวลาค่อนข้างนานในการหาคำตอบในแต่ละครั้งจึงเปลี่ยนโครงสร้างของข้อมูลภายในโปรแกรมจากที่เดิมจะใช้ vector เพื่อเก็บข้อมูลจำนวนมาก เปลี่ยนเป็นการใช้ pointer ของ array แทนแบบเดิมทั้งหมด ซึ่งทำให้ระบบทำงานเร็วขึ้นมาก และมีความถูกต้องเหมือนเดิม

6.3.2 การทำงานของระบบเมื่อทำงานบน console จะใช้เวลาทำงานเพียงเล็กน้อย แต่ไม่สามารถแสดงผลในรูปแบบที่น่าสนใจได้ แต่เมื่อทำงานใน MFC Application ที่สามารถแสดงหน้าจอติดต่อกับผู้ใช้ได้ตามต้องการจะใช้เวลาในการคำนวณนานกว่า

6.4 แนวทางการพัฒนาต่อ

6.4.1 หน้าจอสำหรับติดต่อกับผู้ใช้ของโปรแกรมยังสามารถปรับปรุงและใส่รายละเอียดให้ดึงดูดผู้ใช้ได้มากขึ้น และอาจเพิ่มส่วนตั้งค่าตัวแปร การแสดงผลของโครโมโซมในแต่ละรุ่นอย่างละเอียด เพื่อให้สะดวกต่อการศึกษาลักษณะการทำงานของเจเนติกอัลกอริธึมก็ได้

6.4.2 ทำการปรับปรุงโครงสร้างของระบบให้ใช้เวลาหาคำตอบลดลงจนอยู่ในระยะเวลาที่สามารถนำระบบไปวางแผนเส้นทางในสภาพแวดล้อมจริงได้ และอาจเพิ่มส่วนตรวจสอบการเปลี่ยนแปลงของสภาพแวดล้อม ทำให้ระบบสามารถวางแผนเส้นทางในสภาพแวดล้อมที่อุปสรรคสามารถเปลี่ยนแปลงได้ เป็นการทำงานแบบ real-time มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Andrew Koenig and Barbara E.Moo. 2000. Accelerated C++ : Practical Programming by Example. New Jersey : Addison-Wesley

[2] Michael Negnevitsky. 2005. Artificial Intelligence : A Guide to Intelligent Systems. Harlow, England : Addison-Wesley

[3] cplusplus.com - The C++ Resources Network
: http://cplusplus.com

[4] The Code Project - Free Source Code and Tutorials
: http://www.codeproject.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Jing Xiao, Zbigniew Michalewicz, Lixin Zhang, Krzysztof Trojanowski "Adaptive Evolutionary Planner/Navigator for Mobile Robots" IEEE Transactions on Evolutionary Computation, VOL. 1, NO. 1, April, 1997 , PP.56-59

[2] Miha KovaEiE, Miran BrezoEnik, Joie BaliE "Genetic Algorithm Approach for Autonomous Vehicles", IEEE-ICIT, Maribor, Slovenia, 2003, PP.974-978

[3] Mohamed B. Trabia "Planning Near-Minimum-Length Collision-Free Paths for Robots" IEEE Transactions on Systems, Man and Cybernetics, VOL. 23, NO. 5, September/October, 1993, PP.1481-1488

[4] Neda Shahidi, Hadi Esmaeilzadeh, Marziye Abdollahi, Caro Lucas "Memetic Algorithm Based Path Planning for a Mobile Robot" Transactions on Engineering, Computing and technology V1, December, 2004, PP.56-59

[5] T. R. Addis "Knowledge Science:A Pragmatic Approach to Research in Expert Systems" Univeristy of Reading, Whiteknights, Computer Science Department, Reading, Berks, 1993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารต้นแบบ

Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot

Yanrong Hu and Simon X. Yang
Advanced Robotics and Intelligent Systems (ARIS) Lab
School of Engineering, University of Guelph
Guelph, Ontario, Canada N1G 2W1
Email: syang@uoguelph.ca

Abstract— In this paper, a knowledge based genetic algorithm (GA) for path planning of a mobile robot is proposed, which uses problem-specific genetic algorithms for robot path planning instead of the standard GAs. The proposed knowledge based genetic algorithm incorporates the domain knowledge into its specialized operators, where some also combine a local search technique. The proposed genetic algorithm also features a unique and simple path representation and a simple but effective evaluation method. The knowledge based genetic algorithm is capable of finding an optimal or near-optimal robot path in both complex static and dynamic environments. The effectiveness and efficiency of the proposed genetic algorithm is demonstrated by simulation studies. The irreplaceable role of the specialized genetic operators in the proposed GA for solving robot path-planning problem is demonstrated by a comparison study.

I. Introduction

Path planning is an important issue in mobile robotics. In an environment with obstacles, path planning is to find a suitable collision-free path for a mobile robot to move from a start location to a target location. Very often this path is highly desirable to be optimal or near-optimal with respect to time, distance or energy. Distance is a commonly adopted criterion. Robot path planning has been an active research area, and many methods have been developed to tackle this problem [1], such as global C-space methods [2], [3], potential field methods [4], [5], and neural networks approaches [6], [7]. Each method has its own strength over others in certain aspects. Generally, the main difficulties for robot path-planning problem are computational complexity, local optimum, and adaptability. Researchers have always been seeking alternative and more efficient ways to solve the problem.

There is no doubt that path planning can be viewed as an optimization problem (e.g., shortest distance) under certain constraints (e.g., the given environment and collision-free condition). Since the appearance of genetic algorithms (GAs) in 1975 [8], GAs have been used in solving many optimization problems successfully. GAs are stochastic search techniques analogous to natural evolution based on the principle of survival of the fittest [9], [10]. Potential solutions of a problem are encoded as chromosomes, which form a population. Each individual of the population is evaluated by a fitness function. A selection mechanism based on the fitness is applied to the population and the individuals strive for survival. The fitter ones have more chance to be selected and to reproduce offspring by means of genetic transformations such as crossover and mutation. The process is repeated and the population is evolved generation by generation. After many generations, the population converges to solutions of good quality, and the best individual has good chance to be the optimal or near-optimal solution. The feature of parallel search and the ability of quickly locating high performance region [9] contribute to the success of GAs on many applications.

It is not surprising that researchers applied GAs on path planning for mobile robots [11]–[13]. However, like most early GA applications, most of those methods adopt classical GAs that use fixed-length binary strings and two basic genetic operators, and few modifications were made to the algorithms. Sugihara *et al.* [11] proposed a genetic algorithm for path planning with fix-length binary string chromosomes based on cell representation of mobile robot environment. Its binary encoding is biased and inefficient. Besides, in order to use the standard GA, the path planning solutions are restricted to X-monotone or Y-monotone. A similar approach was also proposed by Tu *et al.* [14], where no obvious improvement is made in spite of using variable-length chromosomes instead of fix-length chromosomes. It takes hours to evolve a solution due to its inefficiency.

The classical GAs use binary strings and two basic genetic operators. After encoding solutions to a problem, the classical GAs are more like "blind" search, and perform well when very little prior knowledge is available. However, GAs do not have to be "blind" search, when additional knowledge about problem is available, it can be incorporated into GAs to improve the efficiency of GAs [10], [15]. Path planning is such a problem that requires knowledge incorporation into the GAs for the problem. Graph technique is a traditional way of representing the environment where a mobile robot moves around. Shibata *et al.* [16], [17] proposed a genetic algorithm based on MAKLINK graph environment representation [18]. In this genetic algorithm, the path is represented by variable-length chromosomes formed by mid-points of the free-links, which is a more natural way of encoding than binary strings. This graph based method needs to form a configuration space before applying the genetic algorithm, which can be very time consuming. Both Hocaolu *et al* [19] and Xiao *et al* [20]designed specialized genetic operators with some heuristic knowledge. In [19], a multi-paths planning algorithm based on

0-7803-8232-3/04/$17.00 ©2004 IEEE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

an iterative multi-resolution path representation was proposed. A path is represented by a hierarchically ordered set of vectors that define path vertices generated by a modified Gram-Schmidt orthogonalization process [21], [22]. Xiao *et al.* [20] proposed an evolutionary planner for both on-line and off-line planning. However, both approaches are relatively complicate on problem representation, evaluation, or GA structure.

In this paper, a knowledge based genetic algorithm is proposed. It uses a simple yet effective path representation that combines grids and coordinates representations. Unlike other grid methods, the grids adopted here do not limit movement of the path, but simplify the chromosome structure and genetic operation by discretizing the environment. This approach makes it possible to have one number for each gene and to use integer numbers instead of real numbers in chromosomes. The proposed GA also has six knowledge based genetic operators. Problem-specific genetic operators are not only designed with domain knowledge, but also incorporate small-scale local search that improves efficiency of the operators. A relatively simple but effective evaluation method is applied to both feasible and infeasible solutions. The proposed GA is suitable for both static and dynamic environments. The effectiveness of the knowledge based GA for mobile robot path planning is demonstrated by simulation studies.

II. The Proposed Knowledge Based GA for Path Planning

The proposed genetic algorithm features its simple and unique problem presentation, its effective evaluation method and its knowledge based genetic operators. The detailed description of each element is presented below.

A. Problem Representation

Representation is a key issue in the work of GAs. The proposed GA uses a simple yet effective path representation. The mobile robot environment is represented by orderly numbered grids, each of which represents a location in the environment. The boundary of obstacles is formed by their actual boundary plus minimum safety distance considering the size of the mobile robot, which makes it possible to treat the mobile robot a point in the environment. As an example, Fig. 1 shows an environment with 10 × 10 grids applied on. A potential path is formed by line segments connecting the nodes falling on the grids with different numbers. Thus a path is encoded as a sequence of grid numbers starting from the source and ending at the target with a various number of intermediate nodes (Fig. 2). A feasible path is a collision free path, i.e. no nodes fall on any obstacle, or no any of line segments of a path intersects an obstacle. The length of a chromosome is variable and between 2 and maximum length N_{max}. Such a grid representation is different from the one that usually uses grids to limit the movement of a path to be one of its eight adjacent cells and uses relative directions to represent a path [11]. The proposed path representation is more like a coordinate representation, but differs by discretization and using integer numbers instead of coordinates (x, y).

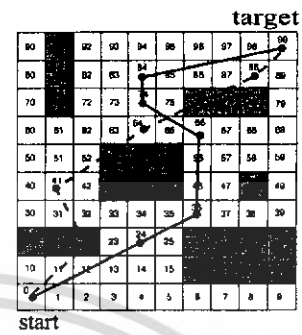


Fig. 1. Mobile robot environment and path representation. Solid line: a feasible path; dashed line: an infeasible path.

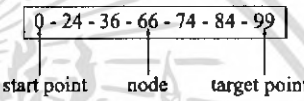


Fig. 2. A sample chromosome: a path represented by nodes falling on grids with different numbers.

B. Evaluation

A path can be either feasible (collision free) or infeasible because intermediate nodes can fall on any of the grids. The evaluation should be able to distinguish feasible and infeasible paths and tell the difference of path qualities within either category. The evaluation function is presented below:

$$F_{cost} = \sum_{i=1}^{N} (d_i + \beta_i C), \tag{1}$$

where N is the number of line segments of a path, d_i is the Euclidean distance of the two nodes forming the line segment, C is a constant, β_i is the coefficient denoting depth of collision, and its definition is given as

$$\beta_i = \begin{cases} 0 & \text{if the } i\text{th line segment is feasible} \\ \sum_{j=1}^{M} \alpha_j & \text{if the line segment intersects obstacle(s)} \end{cases} \tag{2}$$

where M is the number of obstacles the line segment intersects. α_j is determined by considering how deep a line segment intersects an obstacle j. It is defined as the shortest moving distance for escaping the intersected obstacle. This evaluation gives penalty to infeasible paths, but still keep them in the population because they might become good feasible solutions after certain genetic transformations. Besides, this evaluation allows some overlap between fitnesses of feasible and infeasible solutions because a very poor feasible path is

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

not necessarily better than a very good near-feasible path in the sense of evolving solutions. It is beneficial to give more chance to some good infeasible solutions that are easily to be evolved to good solutions. To save computational time, some information obtained by the evaluation needs to be recorded so that later on it can be used by some specialized genetic operators as heuristic knowledge without re-calculation. The information includes feasibility (feasible or infeasible, *node-infeasible or line-infeasible*), *number of infeasible nodes* or line segments, and which obstacle(s) a path intersects.

C. Genetic Operators

Two classical genetic operators: crossover and mutation are not applicable for the problem here. They have to be tailored for the path planning problem and adopted problem representation. Besides, to make the genetic algorithm more effective, four specialized operators are designed to make use of available problem-specific knowledge including knowledge of the environment. These six operators are introduced as following and depicted in Fig. 3.

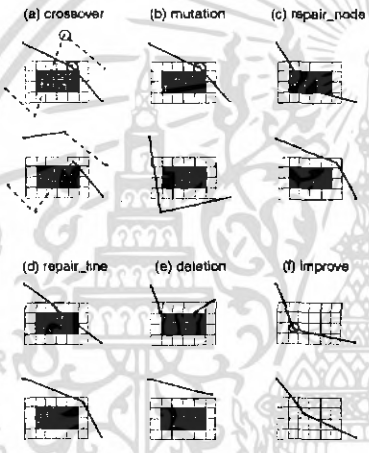


Fig. 3 Six specialized genetic operators that incorporate problem-specific knowledge

Crossover is the operator that randomly choose a node from Parent 1 and the other node from Parent 2. Exchange the part after these two nodes. Check the two offspring, and delete the part between two same nodes if it happens. The choice of different crossover sites in different parents can increase the variability of chromosome length, which benefits exploration of the solution space.

Mutation is that randomly choose a node and replace it with a node that is not included in the path. Mutation is served as a key role to diversify the solution population. Therefore, it is not necessary that a solution is better after it is mutated.

Node_repair is used to move a node falling on an obstacle out of the obstacle and to a best grid around the obstacle. To locate the best grid, a small-scaled local search in the neighbors of the obstacle is applied.

Line_repair is used to repair an infeasible line segment by inserting a suitable node between the two nodes of the *segment*. Again, to locate a best node, similar local search is applied in the all of neighboring grids of the intersected obstacle.

Deletion is applied to both feasible and infeasible path. Randomly choose one node, check its two adjacent nodes, and connected segments, if the deletion of the chosen node is beneficial (turn the infeasible to the feasible, reduce the cost), delete the node.

Improvement is designed for feasible solutions. Randomly chose one node, do a local search in the neighboring grids of the node, move to a best grid. This operator is used for fine tuning of a feasible solution.

These operators are very necessary to evolve feasible and good quality solutions. The firing of these operators depends on two criteria: probability and heuristic knowledge (e.g., if *feasible* then *improve*). The important role of these operators is discussed later.

D. Outline of the Knowledge Based Genetic Algorithm

An outline of the proposed knowledge based genetic algorithm is given in Fig. 4. Initial solutions are generated randomly and are evaluated by the fitness function in Eqn. (1). Two parents are selected according to some selection mechanism. Then one or more genetic operators are selected and applied to the parents according to some probabilities and heuristic knowledge. The whole generation is replaced by children. The best solution so far is updated in each generation, and it will be the final solution when some stop criterion is satisfied. The stop criterion can either be that the preset maximum generation is exceeded, or that the best solution remains unchanged for certain generations. The algorithm is also suitable for dynamic environment. It checks sensing information periodically. If the environment is changed, the algorithm will re-evaluate the current population according to the new environment and starts the process to get a new solution. To increase diversity of the population at the moment, mutation with higher probability is applied to the current *population*.

III. Simulations

The effectiveness of the proposed genetic algorithm is demonstrated by simulations. In the following simulations, parameter setting for the genetic algorithm is: population size = 50, probability for mutation per chromosome is 0.2, and 0.9 for all the rest operators. Tournament selection and elitism are applied. For simplicity, 16 × 16 grids is applied to the environments. All the simulations are conducted on a a Pentium III PC (933MHz).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

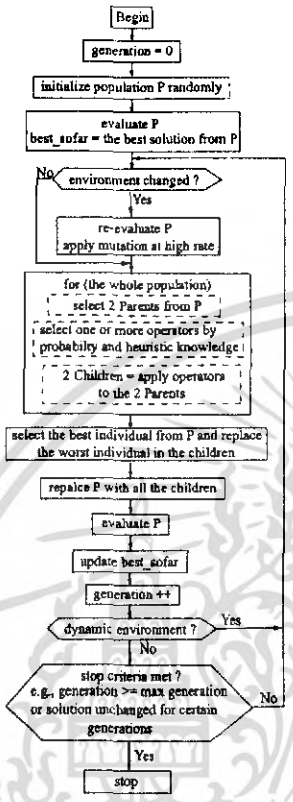


Fig. 4. Outline of the knowledge based genetic algorithm for path planning of a mobile robot.

A. Path Planning in an Environment with a U-shape Obstacle

The proposed genetic algorithm can easily deal with U-shape obstacles. Fig. 5 shows an example. The best solution in the initial population in Fig. 5(a) is far from feasible. Fig. 5(b) shows the best solution after eight generation's evolution, which is feasible but not optimal. The genetic algorithm continues to evolve better solutions (Fig. 5(c)) until the optimal solution (Fig. 5(d)) is found in generation 30.

B. Path Planning in Complex Environments

In this simulation, the proposed genetic algorithm is applied to different mobile robot environments with different obstacle

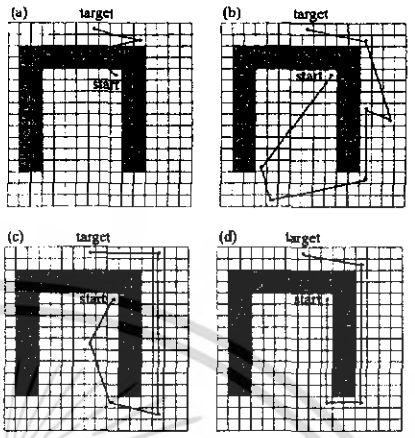


Fig. 5. One typical run of path planning in a U-shape obstacle environment. (a) The best initial path (cost: 80.59); (b) The best solution in generation 8 (cost: 42.02); (c) The best solution in generation 22 (cost: 33.98); (d) The optimal path (cost: 29.10) obtained in generation 30.

layouts. Fig.s 6 and 7 show that the GA is capable of obtaining an optimal/near-optimal collision free path from randomly generated paths. For the environment in Fig. 6, for 20 runs, the average cost is 30.85 with 0.67 Standard Deviation, and the average generation number for obtaining the results in the 20 runs is 257 with 123 Standard Deviation. Fig. 7 shows the result of the GA on a more challenging mobile robot environment.

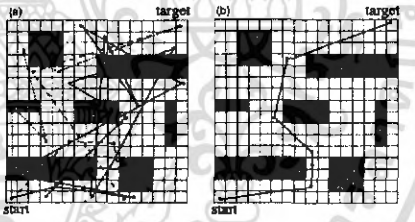


Fig. 6. Path planning in a complex environment. (a) Three randomly generated initial paths. (b) The path obtained by the GA in one typical run.

C. Path Planning in a Dynamic Environment

The proposed genetic algorithm can not only deal with complex static environment, but also be suitable for dynamic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

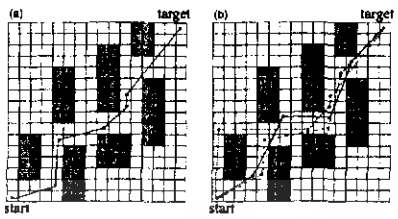


Fig. 7. Path planning in another complex environment. (a) The path obtained by the GA in one typical run; (b) Three alternative paths obtained by the GA from different runs.

environment. When the environment changes, information of the obstacles is updated. The algorithm re-evaluates the current population according to the new information. Therefore, the costs of the solutions in the current population are updated. The quality and feasibility of a path is determined according to the new environment. The original best individual and the better ones may become solutions having bad quality. The GA starts to evolve the population until a new best solution is obtained according to the new environment.

A simulation is presented below to show the adaptability of the proposed genetic algorithm to changing environment. Fig. 8 shows the result of one typical run. In Fig. 8(a), the GA finds a best solution (with cost 28.80) at generation 125 in the environment. Then a new obstacle appears (Fig. 8(b)), and the original solution is no longer feasible. The GA starts to evolve better solutions according to the new environment. After 346 generations, it finds a best path (with cost 29.83) shown in Fig. 8(b). When the obstacle is removed again, after 206 generations the GA evolves an even better new path (with cost 27.77) shown in Fig. 8(c).

IV. Discussion

The above simulation results demonstrate the capability of the proposed GA of evolving satisfactory paths in complex environment. This is mainly contributed by the specialized genetic operators that incorporate heuristic knowledge. To show the contribution of these operators, a comparison study between the GA with and without the specialized operators is conducted as following. We apply the comparison on the same environment as shown in Fig. 6. To see the performance of the GA without the developed specialized operators, we only keep crossover and mutation operators, and shut off all the other operators. Every specialized operator can be viewed as a special mutation operator. Simply shutting off operators makes the two sides of the comparison have different mutation rates. Therefore, we set a best mutation rate for the GA with only crossover and mutation operators. By running the GA for 100 times at different mutation rates, a mutation probability of 0.5 is selected as the best value. Then the GA with and without specialized operators is run for 20 times respectively.

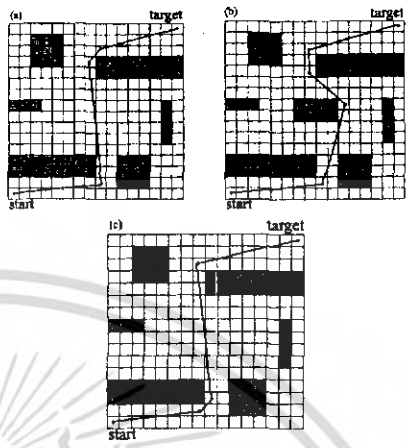


Fig. 8. One typical run of path planning in a dynamic environment. (a) Path obtained in the original environment; (b) Path after adding an obstacle; (c) Path after removal of the obstacle.

Statistic analysis shows that the specialized operators improve the performance of the GA significantly, which is evidently shown in Table I.

TABLE I
Comparison of the GA with and without specialized operators (SD: standard deviation)

Specialized operators		with	without
number of runs		20	20
Best found path cost	Mean	30.86	61.41
	SD	0.67	13.54
Number of generations	Mean	257	709
	SD	125	470

Figure 9(a) shows the best path found in 20 runs by the GA with only crossover and mutation operators. The cost is 39.15, at is found at generation 690. Fig. 9(b) displays the other 3 paths evolved by undergoing only crossover and mutation transformations. Comparing to Fig. 6, it is clear that without the knowledge based operators, the quality of solutions deteriorates dramatically, and for most of the times, it even cannot find feasible path when the environment is complex. The figures and the statistical analysis in Table I prove that the knowledge based genetic operators are essential for solving the path planning problem.

Simulation study also indicates that the proposed knowledge based genetic algorithm is of practical use because the required computational time is quite reasonable. As introduced before, grids are applied to the robot environment to obtain a relatively

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

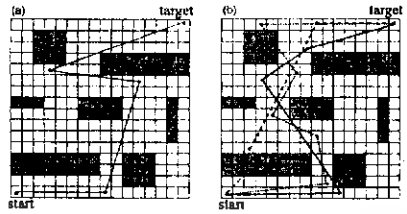


Fig. 9. The path obtained by the GA with only crossover and mutation operators. (a) The best path found in 20 runs; (b) Three other obtained paths.

simple problem representation and to simplify the involved operations of the GA. Obviously, resolution is a main factor affecting the computational time. However, simulation results show that the computational time does not increase exponentially as the resolution increases. For the environment shown in Fig. 6, when resolution increases from 16×16, to 24×24, and to 32×32, the computational time are 5.20, 7.44, and 8.20 respectively, which only increases slighily. Comparing to most of the GAs used for path planning, the proposed GA is fast. Notice that the stop criterion adopted in this simulation is that the best result remains unchanged for certain generations. 100 generations is used here and also counted into computational time.

V. Conclusion

In this paper, a knowledge based genetic algorithm for path planning of a mobile robot is proposed. The GA uses a simple and unique robot path representation that combines grids and coordinates environment representations. The genetic algorithm incorporates the domain knowledge into its problem-specific genetic operators. The developed GA also features its efficient evaluation method that is greatly beneficial for evolving good solutions from infeasible solutions. The effectiveness of the knowledge based operators is demonstrated by simulation studies. The simulation results also show that the proposed genetic algorithm is effective in both complex static and dynamic environments. The efficiency of computational time makes the proposed knowledge based genetic algorithm be able to be applied to real applications.

In this proposed GA, domain knowledge is incorporated into its genetic operators. There are many other ways to utilize additional knowledge besides designing specialized genetic operators. The GA will work better by using the domain knowledge to generate more feasible solutions for initial population. This will be future work. In addition, when the GA deals with dynamic environments, it would be beneficial that some new solutions based on knowledge about the environment change can be injected into the population. Furthermore, the genetic algorithm includes more genetic operators, firing of which depends on their respective probability. On-line tuning of these probabilities is much desirable, which would be another important future work.

Acknowledgment

This work was supported by Natural Sciences and Engineering Research Council (NSERC) and Materials and Manufacturing Ontario (MMO) of Canada.

References

[1] J. C. Latombe, *Robot motion planning*. Boston: Kluwer Academic Publisher, 1991.

[2] T. Lozano-Perez, "Spatial planning: A configuration approach," *IEEE Trans. on Computers*, vol. C-32, no. 2, pp. 108–120, Feb 1983.

[3] M. Sharir, "Algorithmic motion planning in robotics," *Computer*, vol. 22, no. 3, pp. 9–19, March 1989.

[4] P. Khosla and R. Volpe, "Superquadric artificial potentials for obstacle avoidance and approach," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 3, Philadelphia, PA, USA, April 1988, pp. 1778–1784.

[5] E. Rimon and D. E. Dodischek, "Exact robot navigation using artificial potential fields," *IEEE Trans. on Robotics and Automation*, vol. 8, pp. 501–518, 1992.

[6] S. X. Yang and M. Meng, "An efficient neural network approach to dynamic robot motion planning," *Neural Networks*, vol. 13, pp. 143–148, 2000.

[7] ——, "An efficient neural network method for real-time motion planning with safety consideration," *Robotics and Autonomous Systems*, vol. 32, pp. 115–128, 2000.

[8] J. Holland, *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press, 1975.

[9] L. D. (Editor), *Genetic algorithms and simulated annealing*. Los Altos, California: Morgan Kaufman Publishers, 1987.

[10] Z. Michalewicz, *Genetic algorithms + Data structures = Evolution programs, 2nd extended edition*. Springer-Verlag, 1994.

[11] K. Sugihara and J. Smith, "Genetic algorithms for adaptive motion planning of an autonomous mobile robot," in *Proc. of IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, 7 1997, pp. 138–143.

[12] I. Ashiru, C. Czarnecki, and T. Routen, "Characteristics of a genetic based approach to path planning for mobile robots," *J. Network and Computer Applications*, vol. 19, pp. 149–169, 1996.

[13] D. Gallardo, O. Colomina, F. Florez, and R. Rize, "A genetic algorithm for robust motion planning," in *11th Intl. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, vol. 2, Benicassim, Spain, June 1998, pp. 115–121.

[14] J. Tu and S. Yang, "Genetic algorithm based path planning for a mobile robot," in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, Taiwan, September 2003, accepted.

[15] J. J. Grefenstette, "Incorporating problem specific knowledge into genetic algorithms," in *Genetic Algorithm and Simulated Annealing*, L. Davis, Ed. Los Altos, California: Morgan Kaufman Publishers, 1987, pp. 42–60.

[16] T. Shibata and T. Fukuda, "Intelligent motion planning by genetic algorithm with fuzzy critic," in *Proc. of Intl. Symposium on Intelligent Control*, Aug. 1993, pp. 565–570.

[17] T. Shibata, T. Fukuda, K. Kosuge, and F. Arai, "Selfish and coordinative planning for multiple mobile robots by genetic algorithm," in *Proc. of 31th Conf. on Decision and Control*, 12 1992, pp. 2686–2691.

[18] M. K. Habib and H. Asama, "Effecient method to generate collision free paths for autonomous mobile robot based on new free space structuring approach," in *Proc. of IEEE/RSJ Intl. Workshop on Intelligent Robotics and Systems IROS'91*, vol. 2.

[19] C. Hocaoglu and C. Sanderson, "Planning multiple paths with evolutionary speciation," *IEEE Trans. on Evolutionary Computation*, vol. 5, no. 3, pp. 169–191, 6 2001.

[20] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski, "Adaptive evolutionary planner/navigator for mobile robots," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 18–28, 4 1997.

[21] C. Hocaoglu and A. C. Sanderson, "Evolutionary speciation using minimal representation size clustering," in *Evolutionary Programming IV Proc. of the 4th Annual Conf. on Evolutionary Programming*, San Diego, CA, USA, March 1995, pp. 187–203.

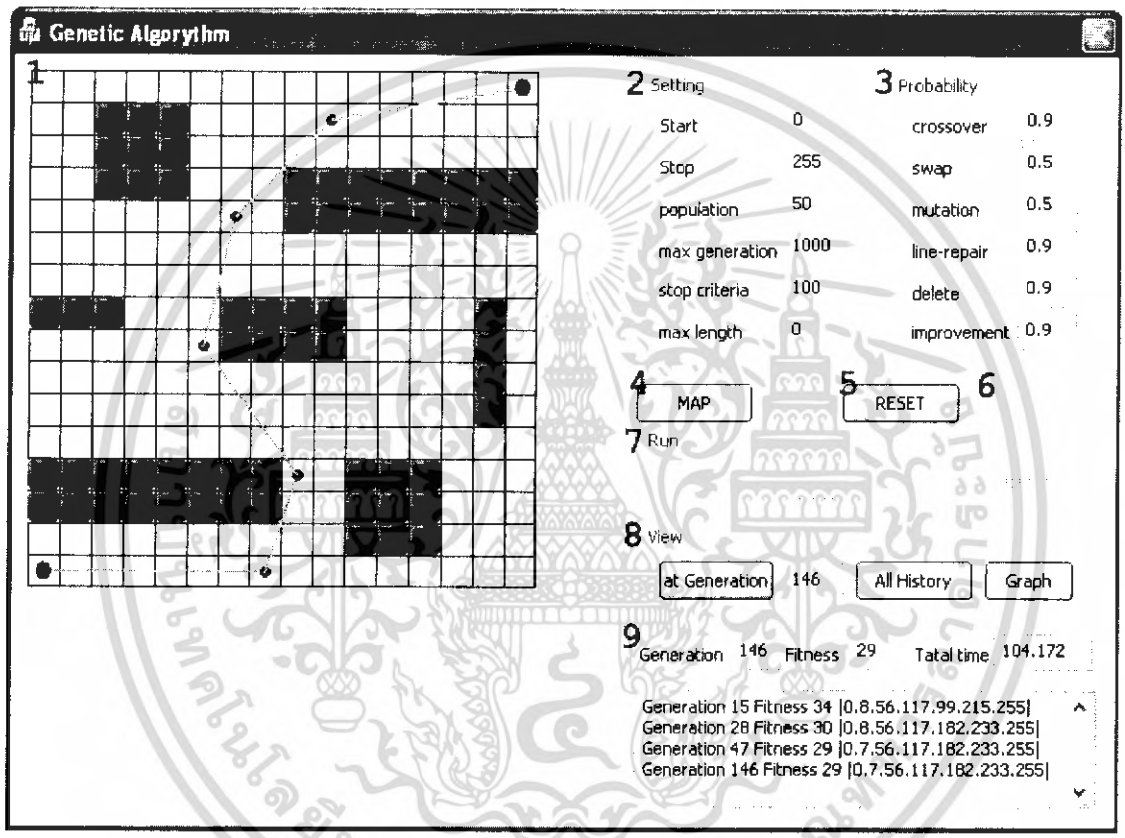
[22] ——, "Multimodal function optimization using minimal representation size clustering and its application to planning multipaths," *J. of Evolutionary Computation*, vol. 5, no. 1, pp. 81–104, 1997.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

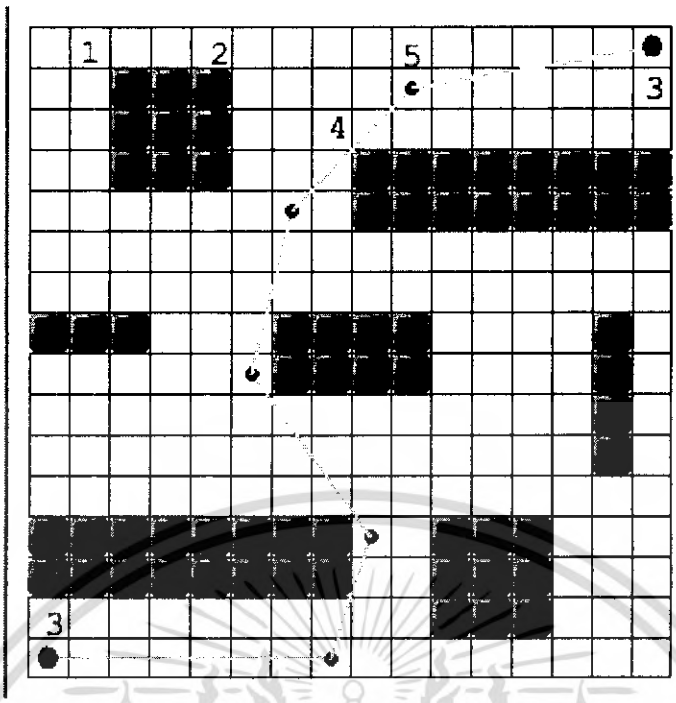
การใช้งานโปรแกรม Knowledge Based Genetic Algorithm For Path Planning ที่โครงงานพัฒนาขึ้น

ส่วนต่างๆของโปรแกรม



1. ส่วนแสดงสภาพแวดล้อม
   1.1. ช่องที่ไม่มีอุปสรรค
   1.2. ช่องที่มีอุปสรรค
   1.3. จุดเริ่มต้นหรือจุดปลายทาง
   1.4. เส้นทาง(แสดงเมื่อเริ่มการทำงาน)
   1.5. จุดเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2. Setting คือส่วนกำหนดค่าพารามิเตอร์ต่างๆของเจเนติกอัลกอริธึม
   2.1. Start กำหนดจุดเริ่มต้นของเส้นทาง
   2.2. Stop กำหนดจุดปลายทางของเส้นทาง
   2.3. Population กำหนดจำนวนโครโมโซมของประชากร
   2.4. max generation กำหนดจำนวนรุ่นสูงสุดที่จะพัฒนา
   2.5. stop criteria กำหนดจำนวนรุ่นจะทำให้หยุดการพัฒนา เมื่อคำตอบที่ดีที่สุดไม่เปลี่ยนแปลงติดต่อกัน
   2.6. max length กำหนดความยาวสูงสุดของโครโมโซม ถ้าตั้งเป็น 0 จะหมายถึง จำนวนช่องหารด้วยสอง

Setting		
Start	1	0
Stop	2	255
population	3	50
max generation	4	1000
stop criteria	5	100
max length	6	0

3. Probability คือส่วนกำหนดค่าความเป็นไปได้ของการทำกระบวนการทางพันธุกรรม (genetic operator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1. crossover กำหนดค่าความเป็นไปได้ของการทำ crossover

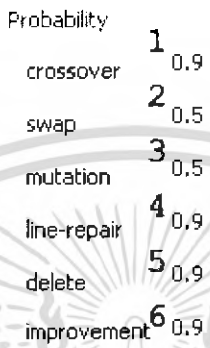
3.2. swap กำหนดค่าความเป็นไปได้ของการทำ swap

3.3. mutation กำหนดค่าความเป็นไปได้ของการทำ mutation

3.4. line-repair กำหนดค่าความเป็นไปได้ของการทำ line-repair

3.5. delete กำหนดค่าความเป็นไปได้ของการทำ delete

3.6. improvement กำหนดค่าความเป็นไปได้ของการทำ improvement



4. MAP เปลี่ยนแปลงตำแหน่งของอุปสรรค และขนาดของสภาพแวดล้อม เมื่อกด จะเปิดหน้าต่าง Map Design ขึ้นมา

4.1. ส่วนแสดงสภาพแวดล้อมที่ออกแบบ

4.2. Use this Map เลือกใช้สภาพแวดล้อมที่ออกแบบ

4.3. Cancel กลับไปใช้สภาพแวดล้อมเดิม

4.4. Reload Map เรียกใช้สภาพแวดล้อมแบบเดิมในการแก้ไข

4.5. Clear Map ลบอุปสรรคทั้งหมดออกจากสภาพแวดล้อมที่ออกแบบ

4.6. Set Map Size กำหนดความละเอียดของสภาพแวดล้อม

4.6.1. จำนวนช่องในแนวนอน

4.6.2. จำนวนช่องในแนวตั้ง

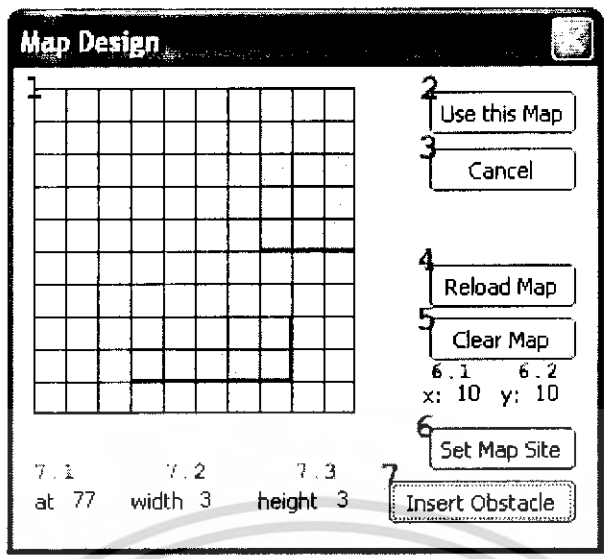
4.7. Insert Obstacle เพิ่มอุปสรรคเข้าไปในสภาพแวดล้อม

4.7.1. ตำแหน่งเริ่มต้นของอุปสรรค

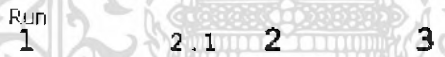
4.7.2. ความกว้างของอุปสรรค

4.7.3. ความยาวของอุปสรรค

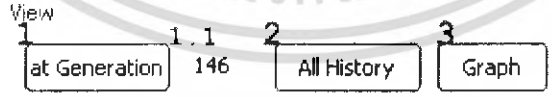
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5. RESET เริ่มการทำงานของเจเนติกอัลกอริธึมหรือเริ่มใหม่
6. STOP หยุดการทำงานของเจเนติกอัลกอริธึม
7. Run คือส่วนควบคุมการทำงานของของเจเนติกอัลกอริธึม
   7.1. run a RUN ให้เจเนติกอัลกอริธึมทำการพัฒนาไป 1 รุ่น
   7.2. run N RUNs ให้เจเนติกอัลกอริธึมทำการพัฒนาไปเป็นจำนวน N รุ่น โดยกำหนดจำนวนรุ่นที่ 7.2.1
   7.3. run til Fin ให้เจเนติกอัลกอริธึมทำการพัฒนาไปจนกว่าจะสิ้นสุดการทำงาน

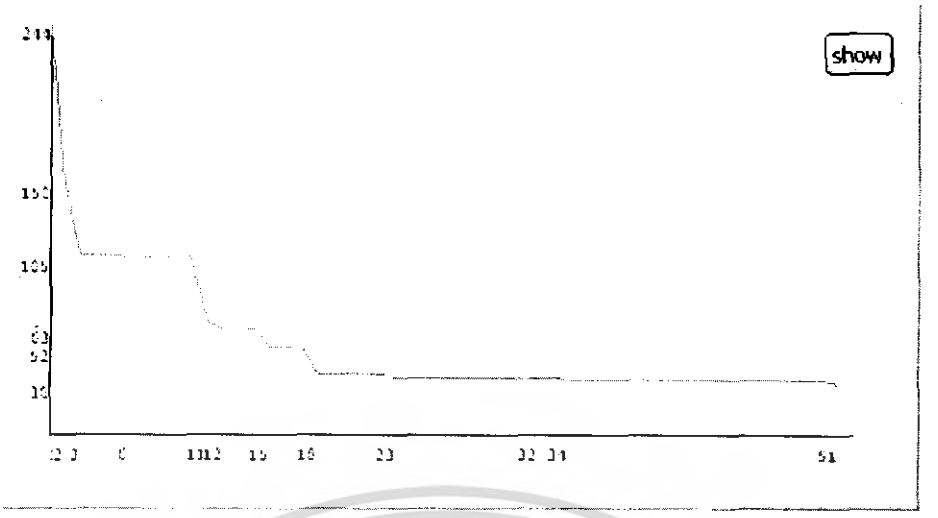


8. View คือส่วนที่เลือกดูผลของการทำงาน
   8.1. at Generation เลือกแสดงเส้นทางที่ดีที่สุดของรุ่นที่ระบุ โดยจะแสดงผลออกมาที่ส่วนแสดงสภาพแวดล้อม เลือกรุ่นที่จะแสดงผลโดยกำหนดค่าที่ 8.1.1



   8.2. All History แสดงเส้นทางที่ดีที่สุดของทุกรุ่นเรียงตามลำดับ โดยจะแสดงผลออกมาที่ส่วนแสดงสภาพแวดล้อม สามารถปรับเวลาดีเลย์ระหว่างการแสดงผลแต่ละรุ่นได้โดยกำหนดที่ 8.1.1
   8.3. Graph แสดงกราฟ กราฟจะแสดงความสัมพันธ์ระหว่างค่าความเหมาะสม(แกนตั้ง)กับจำนวนรุ่น(แกนนอน) โดยถ้าเจเนติกอัลกอริธึมสิ้นสุดการทำงานแล้ว กราฟจะแสดงจนถึงรุ่นที่ได้คำตอบที่ดีที่สุดเท่านั้น

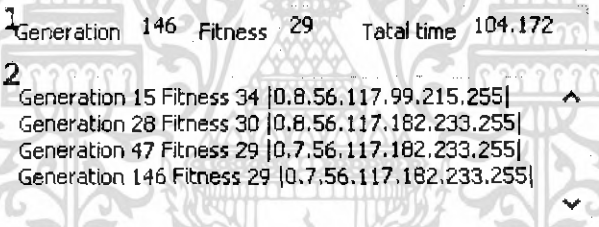
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



9. ส่วนแสดงผลต่างๆ

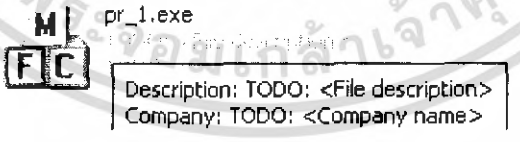
9.1. Generation, Fitness และ Total Time แสดงจำนวนรุ่น, ค่าความเหมาะสมและเวลาที่ใช้ในการคำนวณของการทำงานล่าสุด

9.2. แสดงจำนวนรุ่น, ค่าความเหมาะสม และเส้นทางของโครโมโซมของการทำงานที่ทำอยู่ โดยแสดงเฉพาะรุ่นที่มีการเปลี่ยนแปลงที่น่าสนใจ



การใช้งาน

ดับเบิ้ลคลิกที่ไอคอนของโปรแกรมจะปรากฏหน้าต่างของโปรแกรมขึ้น



เมื่อกำหนดค่าพารามิเตอร์ต่างๆและสภาพแวดล้อมที่ต้องการทดสอบได้แล้ว ให้กดที่ปุ่ม RESET จะเริ่มการทำงานของเจเนติอัลกอริธึมไป 1 รุ่น และแสดงเส้นทางที่ดีที่สุดในรุ่นออกมา สามารถสั่งให้ทำงานต่อโดยคำสั่งใน Run โดยสามารถเลือกดูผลลัพธ์และกราฟได้ตลอดเวลา เมื่อต้องการให้หยุดการทำงานให้กดปุ่ม STOP จะสามารถกำหนดค่าเพื่อเริ่มการทำงานครั้งใหม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้