

การวางแผนเส้นทางที่ดีที่สุดแบบคงทนเพื่อหลบสิ่งกีดขวางด้วยตัวเอง
ของยานยนต์อัจฉริยะอัตโนมัติในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์

OPTIMAL ROBUST PATH PLANNING WITH SELF-OBSTACLE AVOIDANCE OF
INTELLIGENT VEHICLE FOR HARD DISK DRIVE MANUFACTURING



รวินันท์ ประเสริฐทวีลาภ

RAWINUN PRASERTTAWEELAP

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2564

KMITL-2021-EN-D-018-021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPTIMAL ROBUST PATH PLANNING WITH SELF-OBSTACLE
AVOIDANCE OF INTELLIGENT VEHICLE FOR HARD DISK DRIVE
MANUFACTURING



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2021
KMUTL-2021-EN-D-018-021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2021

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การวางแผนเส้นทางที่ดีที่สุดแบบคงทนเพื่อหลบสิ่งกีดขวางด้วยตัวเอง ของยานยนต์อัจฉริยะอัตโนมัติในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์
นักศึกษา	นางสาวรวินันท์ ประเสริฐทวีลาภ
รหัสประจำตัว	59601010
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2564
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.สมยศ เกียรติวนิชวิไล

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ นำเสนอการออกแบบเส้นทางการเคลื่อนที่ของยานยนต์อัจฉริยะภายใต้สภาพแวดล้อมของอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ โดยการออกแบบเส้นทางการเคลื่อนที่ที่ปลอดภัยที่สุดเพื่อรองรับการทำงานในระบบอัตโนมัติภายในอุตสาหกรรม ทั้งนี้ การออกแบบเส้นทางที่ปลอดภัยที่สุดอาจส่งผลกระทบต่อประสิทธิภาพในการทำงานโดยรวมของระบบ ซึ่งจะทำให้การทำงานต่อวันมีประสิทธิภาพช้าลงเพื่อให้ได้การเคลื่อนที่ของยานยนต์อัจฉริยะมีความปลอดภัย งานวิจัยนี้จึงออกแบบการค้นหาการเคลื่อนที่ที่ปลอดภัยที่สุดโดยการสร้างอัลกอริทึมในการค้นหาที่สามารถหาคำตอบได้อย่างรวดเร็วและถูกต้อง นอกจากนี้ งานวิจัยนี้ยังเพิ่มประสิทธิภาพของแบบจำลองด้วยการเพิ่มความผิดพลาดลงไปแบบจำลอง เพื่อทดสอบกรณีที่ปัจจัยต่างๆมีผลทำให้เกิดค่าผิดพลาดมากที่สุด จากผลการทดลองในแบบจำลองพบว่า ภายใต้สภาพแวดล้อมของอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ สามารถค้นหาเส้นทางที่ดีที่สุดที่จำนวนหุ่นยนต์รวมมากกว่าหรือเท่ากับ 5 หุ่นยนต์ โดยไม่พบการชนกันของหุ่นยนต์กับสิ่งกีดขวาง และไม่พบการชนกันระหว่างหุ่นยนต์ เวลาในกระบวนการทำงานที่เร็วที่สุดอยู่ที่ระบบ 5 หุ่นยนต์ เมื่อทำงานที่ 30 งาน คือ 85.6650 วินาที นอกจากนี้ วิธีการที่งานวิจัยนี้นำเสนอ นั้น สามารถคาดคะเนจำนวนยานยนต์ต่อพื้นที่นั้นๆ ค่าความปลอดภัย และเวลาที่ใช้ในการทำงานได้ดี ซึ่งจะเป็นแบบจำลองต้นแบบของอุตสาหกรรมในการเลือกการลงทุนภายในโรงงานอุตสาหกรรม ภายใต้ความต้องการในการทำงานนั้นๆ ได้เป็นอย่างดี

Thesis	OPTIMAL ROBUST PATH PLANNING WITH SELF-OBSTACLE AVOIDANCE OF INTELLIGENT VEHICLE FOR HARD DISK DRIVE MANUFACTURING
Student	Miss Rawinun Praserttaweelap
Student ID.	59601010
Degree	Doctor of Engineering
Program	Electrical Engineering
Year	2021
Thesis Advisor	Assoc.Prof.Dr.Somyot Kiatwanidvilai

ABSTRACT

This thesis proposes a new design for optimal path planning of intelligent vehicle based on the hard disk drive manufacturing environment. The proposed method is the new safety path searching for low risk of collision in manufacturing. However, the maximum capability of systems may be lower due to the safety path searching. The processing time is higher for the low risk of collision. This research presents the safety path searching with the fast optimal convergence solution. Moreover, this research increases the efficiency of the prediction model with error built-in. The prediction model of this research can represent the worst case scenario. From the simulation results, the proposed method of this research illustrates a good prediction model. The maximum robot running in this model is 5 robots with 85.6650 seconds on 30 tasking simulations. The prediction parameters are the number of robots in area, the safety value, and the processing time. This model is appropriate for investment prediction in manufacturing. In addition, the model of this research is flexible for any upgrades.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ.ดร.สมยศ เกียรติวนิชวิไล ที่ให้ความช่วยเหลือ ให้คำชี้แนะ ช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณ บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) นายสัญญาชัย ทองจันทร์ นายเกรียงชัย เลิศรุ่งเรือง และ นายภูมิศักดิ์ งามวงษ์วาน ที่ให้การสนับสนุนทุนสำหรับงานวิจัยนี้ รวมทั้งให้การสนับสนุน ห้องปฏิบัติการและข้อมูลต่างๆที่เป็นประโยชน์กับทุนวิจัยนี้เป็นอย่างมาก

ขอขอบพระคุณ โครงการพัฒนานักวิจัยและงานวิจัยเพื่ออุตสาหกรรม ที่ให้การสนับสนุนทุน สำหรับงานวิจัยนี้เป็นอย่างดี

สุดท้ายต้องขอขอบคุณ ครอบครัว คุณแม่ นางศรภมล แซ่พั้ว พี่สาว นางสาว โศรยา ประเสริฐทวี ลาก พี่ชาย นาย พศิน ประเสริฐทวีลาก และนายสมคิด ปัญญาพัฒนศิริ ที่คอยให้กำลังใจ ให้คำแนะนำ และสนับสนุนการดำเนินการวิจัยนี้อย่างดีเสมอมา

รวินันท์ ประเสริฐทวีลาก

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	2
1.5 ขอบเขตการวิจัย.....	3
1.6 ขั้นตอนของการศึกษา.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 อุตสาหกรรมฮาร์ดดิสก์ไดรฟ์.....	4
2.2 การวางแผนเส้นทางการเคลื่อนที่ของยานยนต์อัจฉริยะ.....	11
2.2.1 แผนที่.....	12
2.2.1.1 topological maps.....	12
2.2.1.2 occupancy grid maps.....	12
2.2.2 วิธีการวางแผนเส้นทางการเคลื่อนที่.....	13
2.2.2.1 Dijkstra’s algorithm.....	13
2.2.2.2 A* algorithm.....	14
2.2.2.3 D* algorithm.....	15
2.2.2.4 Rapidly-Exploring Random Trees (RRT) algorithm	16
2.3 การค้นหาค่าที่ดีที่สุด.....	16
2.3.1 Genetic algorithm.....	19

สารบัญ(ต่อ)

	หน้า
2.3.2 Particle swarm optimization.....	21
2.3.3 Ant Colony Optimization (ACO).....	24
2.3.4 Artificial Bee Colony (ABC).....	26
2.4 งานวิจัยที่เกี่ยวข้อง.....	28
บทที่ 3 การออกแบบการทดลอง.....	31
บทที่ 4 ผลการทดลองและการวิเคราะห์ผลการทดลอง.....	51
บทที่ 5 สรุปผลการทดลอง ข้อเสนอแนะ และแนวทางการพัฒนา	66
เอกสารอ้างอิง.....	68



สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงเงื่อนไขของค่าพีโรโมนที่เปลี่ยนไป.....	25
3.1 แสดงเงื่อนไขของค่า OA.....	35
3.2 แสดงเงื่อนไขของค่า RC.....	47



สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบภายในฮาร์ดดิสก์ไดรฟ์.....	4
2.2 กระบวนการปัจจุบันในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์.....	6
2.3 กระบวนการใหม่ที่วางแผนในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์.....	7
2.4 กระบวนการตรวจสอบวัตถุดิบนำเข้า.....	8
2.5 กระบวนการตรวจสอบวัตถุดิบนำเข้าแบบปรับปรุงการเคลื่อนย้ายชิ้นงาน.....	9
2.6 หัวอ่าน.....	10
2.7 แชนยี่ด.....	10
2.8 แบบเบื้องต้นของการจัดวางพื้นที่ทำงานของยานยนต์อัจฉริยะ.....	11
2.9 แผนที่แบบ topological maps.....	12
2.10 แผนที่แบบ occupancy grid maps.....	13
2.11 ตัวอย่างการค้นหาเส้นทางที่ดีที่สุดด้วย Dijkstra's algorithm.....	14
2.12 ตัวอย่างการค้นหาเส้นทางที่ดีที่สุดด้วย A* algorithm.....	15
2.13 ตัวอย่างการค้นหาเส้นทางที่ดีที่สุดด้วย Rapidly-Exploring Random Trees (RRT).....	16
2.14 โครงสร้างของวิธีการค้นหาที่ดีที่สุด (optimization method).....	18
2.15 ขั้นตอนของวิธีเชิงพันธุกรรม (Genetic Algorithm).....	20
2.16 เวกเตอร์คำนวณของวิธี PSO.....	22
2.17 ขั้นตอนของ PSO.....	23
2.18 การทดลอง double bridge เพื่อศึกษาพฤติกรรมกรหาอาหารของมด.....	24
3.1 แผนการทำงานวิจัย.....	31
3.2 แบบจำลองแผนที่ที่ออกแบบโดยอ้างอิงจากแบบเบื้องต้นของโรงงานอุตสาหกรรม...	33
3.3 ระยะจากจุดขอบของแต่ละสถานีกับตำแหน่งของยานยนต์.....	33
3.4 ยานยนต์อัจฉริยะและอาณาเขตวงกลม.....	33
3.5 อาณาเขตวงกลมของสิ่งกีดขวางและยานยนต์.....	34
3.6 เปรียบเทียบการสร้างเส้นเชื่อมโยงของวิธี Linear กับ Cubic-spline.....	37
3.7 ขั้นตอนของวิธีการที่ต้องการนำเสนอของงานวิจัย.....	38
3.8 กราฟการค้นหาค่า OA ที่น้อยที่สุดกับค่า β	39

สารบัญญรูป(ต่อ)

รูปที่	หน้า
3.9 ขั้นตอนของวิธีการหาเส้นทางการเคลื่อนที่ที่ดีที่สุดบนสภาพแวดล้อมที่มีหุ่นยนต์.....	41
มากกว่า 1 หุ่นยนต์ด้วยวิธี Conventional.....	
3.10 ตำแหน่งของสิ่งกีดขวางบนแผนที่.....	42
3.11 ตำแหน่งช่องกริด 347 เริ่มต้นและทางเลือกเส้นทางไปยังตำแหน่งช่องกริด 267.....	44
3.12 ตำแหน่งช่องกริด 327 ที่ถูกเลือกและทางเลือกเส้นทางไปยังตำแหน่งช่องกริด.....	
267.....	45
3.13 ตำแหน่งช่องกริด 307 ที่ถูกเลือก และทางเลือกเส้นทางไปยังตำแหน่งช่องกริด.....	
267.....	46
3.14 ตำแหน่งช่องกริด 287 ที่ถูกเลือก และทางเลือกเส้นทางไปยังตำแหน่งช่องกริด	
267.....	46
3.15 ขั้นตอนของวิธีการหาเส้นทางการเคลื่อนที่ที่ดีที่สุดบนสภาพแวดล้อมที่มีหุ่นยนต์	
มากกว่า 1 หุ่นยนต์แบบใช้ OA ในการหลบสิ่งกีดขวางแบบคงที่ และใช้ RC สำหรับหา	
เส้นทางที่มีความเสี่ยงในการชนระหว่างหุ่นยนต์สูงสุด	48
3.16 กระบวนการการค้นหาเส้นทางที่มีความเสี่ยงต่อการชนหุ่นยนต์อื่นมากที่สุดด้วย	
Artificial Bee Colony.....	50
4.1 จุดเริ่มต้นและจุดเป้าหมายในแผนที่ในแบบจำลอง.....	51
4.2 จุดเริ่มต้นและจุดเป้าหมายในแผนที่ในแบบจำลองด้วย linear interpolation.....	51
4.3 จุดเริ่มต้นและจุดเป้าหมายในแผนที่ในแบบจำลองด้วย cubic-spline	
interpolation.....	52
4.4 ค่าของ fitness function ในแต่ละ iteration เปรียบเทียบระหว่าง linear	
interpolation กับ cubic-spline interpolation.....	52
4.5 ค่าของ OA ในแต่ละ iteration เปรียบเทียบระหว่าง linear interpolation กับ	
cubic-spline interpolation.....	53
4.6 ค่าของ fitness function เปรียบเทียบระหว่าง 100 จุดและ 200 จุด.....	53
4.7 ค่าของ OA เปรียบเทียบระหว่าง 100 จุดและ 200 จุด.....	53

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.8 ลักษณะของเส้นทางการค้นหาระหว่างการค้นหาเส้นทางที่สั้นที่สุดกับการค้นหา..... เส้นทางที่ปลอดภัยที่สุด แบบที่ 1.....	54
4.9 ลักษณะของเส้นทางการค้นหาระหว่างการค้นหาเส้นทางที่สั้นที่สุดกับการค้นหา..... เส้นทางที่ปลอดภัยที่สุด แบบที่ 2.....	55
4.10 ลักษณะของเส้นทางการค้นหาระหว่างการค้นหาเส้นทางที่สั้นที่สุดกับการค้นหา.... เส้นทางที่ปลอดภัยที่สุด แบบที่ 3.....	55
4.11 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ..... proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 1..... หุ่นยนต์.....	56
4.12 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ..... proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 2 หุ่นยนต์.....	57
4.13 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ..... proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 3 หุ่นยนต์..	58
4.14 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ..... proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 4..... หุ่นยนต์.....	59
4.15 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ..... proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 5..... หุ่นยนต์.....	60
4.16 การทดสอบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 2 หุ่นยนต์.....	61
4.17 การทดสอบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 3 หุ่นยนต์.....	62
4.18 การทดสอบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 4 หุ่นยนต์.....	63
4.19 การทดสอบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 5 หุ่นยนต์.....	64
4.20 กราฟเส้นแสดงผลการทดลองของการทดสอบ 30 งานด้วยวิธีการที่งานวิจัยนี้..... นำเสนอ.....	65

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน โรงงานอุตสาหกรรมอิเล็กทรอนิกส์เป็นหนึ่งในโรงงานที่มีความสำคัญต่อการพัฒนาทางเศรษฐกิจและเทคโนโลยีเป็นอย่างมาก การพัฒนากระบวนการผลิตจึงเป็นการพัฒนาอย่างต่อเนื่องเพื่อตอบสนองความต้องการในอนาคต บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัดเป็นหนึ่งในโรงงานอุตสาหกรรมอิเล็กทรอนิกส์ด้านหน่วยจัดเก็บข้อมูลที่มีการพัฒนากระบวนการผลิตอย่างสม่ำเสมอ เพราะความต้องการการจัดเก็บข้อมูลมีอัตราเพิ่มสูงขึ้นอย่างต่อเนื่องในทุกๆปี [1] ซึ่งหากยังดำเนินการผลิตด้วยการใช้แรงงานมนุษย์ต่อไปนั้น จะไม่สามารถดำเนินการผลิตได้ตามแผนที่วางไว้ หนึ่งในแผนการดำเนินการพัฒนาอย่างต่อเนื่องคือการนำยานยนต์อัจฉริยะเข้ามาใช้งานในโรงงานอุตสาหกรรมแทนการใช้แรงงานมนุษย์ เพื่อลดความผิดพลาดหรือความเสียหายที่อาจเกิดขึ้นระหว่างกระบวนการและเพิ่มกำลังการผลิตหน่วยจัดเก็บข้อมูลที่มีความต้องการเพิ่มขึ้นอย่างมากในปัจจุบัน [1] ยานยนต์อัจฉริยะจะเข้ามา มีบทบาทในการเคลื่อนย้ายตำแหน่งของสินค้าไปยังสถานที่ที่ต้องการ อาทิเช่น การให้ยานยนต์อัจฉริยะรับสินค้าจากตำแหน่งหนึ่งเพื่อไปเข้าระบบทดสอบที่อีกตำแหน่งหนึ่งและให้ยานยนต์อัจฉริยะรับสินค้าที่ทดสอบเสร็จเรียบร้อยแล้วมายังตำแหน่งจัดเก็บ โดยเป็นการเคลื่อนที่ภายในโรงงานอุตสาหกรรมที่เป็นระบบห้องสะอาด (Clean room) ที่มีการควบคุมความสะอาดอยู่ที่คลาส 1000 ซึ่งหมายถึง จำนวนสิ่งปนเปื้อนที่มีขนาดมากกว่า 0.5 ไมครอนต่อหนึ่งตารางเมตรและมีค่าไม่เกิน 1000 สิ่งปนเปื้อนเพื่อปรับปรุงกระบวนการภายในสายการผลิต และด้วยข้อจำกัดภายในโรงงานอุตสาหกรรม การจัดการการวางแผนเส้นทางการเคลื่อนที่อัตโนมัติของยานยนต์อัจฉริยะจะต้องมีประสิทธิภาพ ปลอดภัย และมีความถูกต้องแม่นยำ ระบบที่ออกแบบจึงมีความซับซ้อนเพื่อรองรับการจัดการระบบอย่างสมบูรณ์แบบ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

จุดมุ่งหมายของการทำวิจัยนี้ คือการศึกษาการวางแผนเส้นทางการเคลื่อนที่ที่ดีที่สุดของยานยนต์อัจฉริยะที่ใช้ภายในโรงงานอุตสาหกรรม และสร้างเป็นแบบจำลองต้นแบบเพื่อการคาดคะเนการลงทุนของโรงงานอุตสาหกรรมตามเงื่อนไขที่โรงงานอุตสาหกรรมต้องการ

1.2.1 เพื่อศึกษาความต้องการในการใช้ยานยนต์อัจฉริยะภายในโรงงานอุตสาหกรรม

1.2.2 เพื่อศึกษาข้อจำกัดภายในโรงงานอุตสาหกรรม

- 1.2.3 เพื่อศึกษาการทำงานของยานยนต์อัจฉริยะ
- 1.2.4 เพื่อศึกษาการวางแผนเส้นทางของยานยนต์อัจฉริยะ
- 1.2.5 เพื่อศึกษาวิธีการในการวางแผนเส้นทางที่ดีที่สุดของยานยนต์อัจฉริยะ
- 1.2.6 เพื่อศึกษาวิธีการในการค้นหาค่าของฟังก์ชันที่ดีที่สุด
- 1.2.7 เพื่อออกแบบการวางแผนเส้นทางที่ดีที่สุดของยานยนต์อัจฉริยะให้กับโรงงานอุตสาหกรรม
- 1.2.8 เพื่อออกแบบการวางแผนเส้นทางที่ปลอดภัยที่สุดให้กับโรงงานอุตสาหกรรม
- 1.2.9 เพื่อออกแบบการวางแผนเส้นทางที่ดีที่สุดของยานยนต์อัจฉริยะภายใต้สภาพแวดล้อมแบบหลายหุ่นยนต์
- 1.2.10 เพื่อออกแบบแบบจำลองต้นแบบเพื่อการนำไปประยุกต์ใช้จริงในโรงงานอุตสาหกรรม

1.3 สมมติฐานของการศึกษา

งานวิจัยนี้นำเสนอการสร้างแบบจำลองใหม่เพื่อโรงงานอุตสาหกรรมสำหรับการวางแผนเส้นทางเคลื่อนที่ของยานยนต์อัจฉริยะโดยมุ่งเน้นที่การค้นหาเส้นทางที่ปลอดภัยที่สุด อย่างไรก็ตามการค้นหาเส้นทางที่ปลอดภัยที่สุดอาจส่งผลกระทบต่อประสิทธิภาพโดยรวมของระบบนั้นลดลง เนื่องจากการเลือกเส้นทางที่ปลอดภัยที่สุดอาจจะส่งผลให้ระยะเวลาในการเคลื่อนที่รวมของระบบเพิ่มขึ้นและเพื่อลดความเสี่ยงให้การชนกันของยานยนต์อัจฉริยะในระบบ งานวิจัยนี้จึงค้นหาวิธีการที่สามารถใช้ในการค้นหาเส้นทางที่ปลอดภัยที่สุดโดยที่มีผลกระทบต่อความเร็วในการทำงานของทั้งระบบน้อยที่สุด

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

งานวิจัยนี้เริ่มต้นจากการศึกษาข้อมูลจากงานวิจัยที่ได้รับการยอมรับในวารสารและการประชุมนานาชาติ ในเรื่องของการหาเส้นทางเคลื่อนที่ของยานยนต์อัจฉริยะและการค้นหาค่าฟังก์ชันที่ดีที่สุดในรูปแบบต่างๆ โดยทฤษฎีที่งานวิจัยนี้นำมาประยุกต์ใช้ได้แก่

- 1.4.1 A* path planning ซึ่งเป็นการประมาณตำแหน่งถัดไปของยานยนต์อัจฉริยะ โดยใช้ค่า heuristic ซึ่งเป็นที่นิยมใช้อย่างแพร่หลายในงานวิจัย
- 1.4.2 Particle Swarm Optimization (PSO) เป็นวิธีการค้นหาค่าที่ดีที่สุดจากฟังก์ชันวัตถุประสงค์ที่ออกแบบไว้
- 1.4.3 Ant Colony Algorithm (ACO) เป็นวิธีการค้นหาค่าที่ดีที่สุดโดยใช้หลักการของพฤติกรรมหาอาหารของมด
- 1.4.4 Artificial Bee Colony Algorithm (ABC) เป็นวิธีการค้นหาค่าที่ดีที่สุดโดยใช้หลักการของพฤติกรรมหาอาหารของผึ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขอบเขตการวิจัย

งานวิจัยนี้จัดทำขึ้นเพื่อศึกษาและออกแบบแบบจำลองการวางแผนเส้นทางการเคลื่อนที่ของยานยนต์อัจฉริยะในโรงงานอุตสาหกรรม โดยเริ่มต้นจากการศึกษาข้อมูลความต้องการภายในโรงงานอุตสาหกรรมรวมทั้งข้อจำกัดต่างๆ และเริ่มต้นออกแบบแบบจำลองจากพื้นที่แบบตัวอย่างที่ต้องการใช้จริงภายในโรงงานอุตสาหกรรม หลังจากนั้น เริ่มต้นการออกแบบการวางแผนเส้นทางการเคลื่อนที่ของยานยนต์อัจฉริยะเพียง 1 ยานยนต์ก่อน และเริ่มการออกแบบฟังก์ชันใหม่สำหรับการหาเส้นทางที่ปลอดภัยที่สุด เพื่อตรวจสอบว่าวิธีการใหม่ที่งานวิจัยนี้นำเสนอ นั้น ให้ผลของการวางแผนเส้นทางที่ดีที่สุดกับยานยนต์อัจฉริยะ ต่อมาจึงเริ่มออกแบบแบบจำลองให้รองรับการทำงานเมื่อยานยนต์อัจฉริยะมีมากกว่า 1 ยานยนต์ ซึ่งเป็นระบบที่เรียกว่า Multi-robot systems โดยการเพิ่มค่าความคลาดเคลื่อนเข้าไปในระบบเพื่อให้แบบจำลองมีความใกล้เคียงกับสภาวะจริงมากขึ้น ซึ่งจำนวนหุ่นยนต์ที่มากที่สุดที่ใช้ในแบบจำลองนี้คือ 5 หุ่นยนต์ตามความต้องการในแผนการของภาคอุตสาหกรรม ค่าที่วัดผลคือการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ภายใต้สภาพแวดล้อมจำลองของภาคอุตสาหกรรมที่ไม่มีการชนกันระหว่างหุ่นยนต์กับสิ่งกีดขวางและไม่มีการชนกันระหว่างหุ่นยนต์กับหุ่นยนต์ในแบบจำลอง พร้อมทั้งประเมินค่าความเสี่ยงในการชนสิ่งกีดขวางและหุ่นยนต์โดยเปรียบเทียบวิธีการใหม่ที่งานวิจัยนี้นำเสนอ กับวิธี conventional รวมทั้งเปรียบเทียบเวลาที่ใช้ในการทำงาน ผลลัพธ์ทั้งหมดที่ได้จะถูกนำไปใช้เป็นแบบจำลองในการประเมินการทำงานจริงในโรงงานอุตสาหกรรมก่อนการลงทุนจริงและสามารถนำไปประยุกต์ใช้ในการปรับเปลี่ยนการออกแบบภายในโรงงานอุตสาหกรรมหรือประยุกต์ใช้กับสภาพแวดล้อมจริงในอนาคตด้วยเช่นกัน

1.6 ขั้นตอนของการศึกษา

1.6.1 จัดเตรียมหาข้อมูลที่เกี่ยวข้องกับการทำวิจัย ศึกษาข้อมูลจากโรงงานอุตสาหกรรมและงานวิจัยที่เกี่ยวข้องที่ได้รับการตีพิมพ์ในวารสารหรือการประชุมวิชาการที่เป็นที่ยอมรับ

1.6.2 จัดเตรียมเครื่องมือที่เกี่ยวข้องกับงานวิจัยทั้งหมด รวมทั้งศึกษาการใช้เครื่องมือต่างๆ

1.6.3 ออกแบบและสร้างโปรแกรมสำหรับการวิเคราะห์การเคลื่อนที่ของยานยนต์อัจฉริยะแบบ 1 ยานยนต์ ภายใต้พื้นที่จำลองตามแบบแผนของโรงงานอุตสาหกรรม

1.6.4 ทดสอบและปรับปรุงการตั้งค่าให้เหมาะสม

1.6.5 ออกแบบและสร้างโปรแกรมต่อยอดของการเคลื่อนที่ของยานยนต์อัจฉริยะแบบมากกว่า 1 ยานยนต์ เพื่อทดสอบประสิทธิภาพของแบบจำลองที่ออกแบบให้ใกล้เคียงกับโรงงานอุตสาหกรรม

1.6.6 ทดสอบและปรับปรุงการตั้งค่าให้เหมาะสมในแบบจำลองพร้อมทั้งเพิ่มความแปรผันในระบบ

1.6.7 สรุปผลของงานวิจัย เพื่อจัดทำวิทยานิพนธ์ต่อไป

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 อุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive Manufacturing)

อุตสาหกรรมฮาร์ดดิสก์ไดรฟ์เป็นหนึ่งในอุตสาหกรรมอิเล็กทรอนิกส์ที่มีการนำระบบอัตโนมัติเข้ามาประยุกต์ใช้ภายในโรงงาน เนื่องด้วยความต้องการในการผลิตชิ้นงานที่มีขนาดเล็กในหน่วยไมโครเมตร เครื่องจักรอัตโนมัติจึงมีบทบาทสำคัญในการทำงานในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ เพื่อให้กระบวนการมีประสิทธิภาพสูง และรองรับต่อความต้องการของตลาดได้อย่างเหมาะสม [2]

ฮาร์ดดิสก์ไดรฟ์ คือ อุปกรณ์อิเล็กทรอนิกส์ที่ใช้สำหรับการจัดเก็บข้อมูล โดยใช้หลักการการอ่านและเขียนข้อมูลด้วยหัวอ่านแม่เหล็กจากการเปลี่ยนแปลงของสนามแม่เหล็กที่อยู่บนแผ่นดิสก์ [2] แผ่นดิสก์นี้จะอยู่ภายในฮาร์ดดิสก์ไดรฟ์ซึ่งมีลักษณะเป็นแผ่นจานโลหะแม่เหล็ก มีสารแม่เหล็กเคลือบอยู่บนแผ่นแก้วหรือแผ่นอลูมิเนียมที่มีความเรียบ ในฮาร์ดดิสก์ไดรฟ์อาจมีแผ่นดิสก์มากกว่า 1 แผ่น ขึ้นอยู่กับการออกแบบเพื่อความต้องการของปริมาณการจัดเก็บข้อมูลในชิ้นงานนั้นๆ ในการทำงานของฮาร์ดดิสก์ไดรฟ์ แผ่นดิสก์จะมีการเคลื่อนที่แบบหมุน โดยมีความเร็วในการหมุนอยู่ที่ 7,200 ถึง 15,000 รอบต่อนาที หัวอ่านแม่เหล็กที่เคลื่อนที่อยู่เหนือแผ่นดิสก์นั้น จะอยู่เหนือแผ่นดิสก์ด้วยระยะประมาณ 10 ไมโครเมตรเท่านั้น [2] ส่วนประกอบภายในฮาร์ดดิสก์ไดรฟ์ แสดงดังรูปที่ 2.1

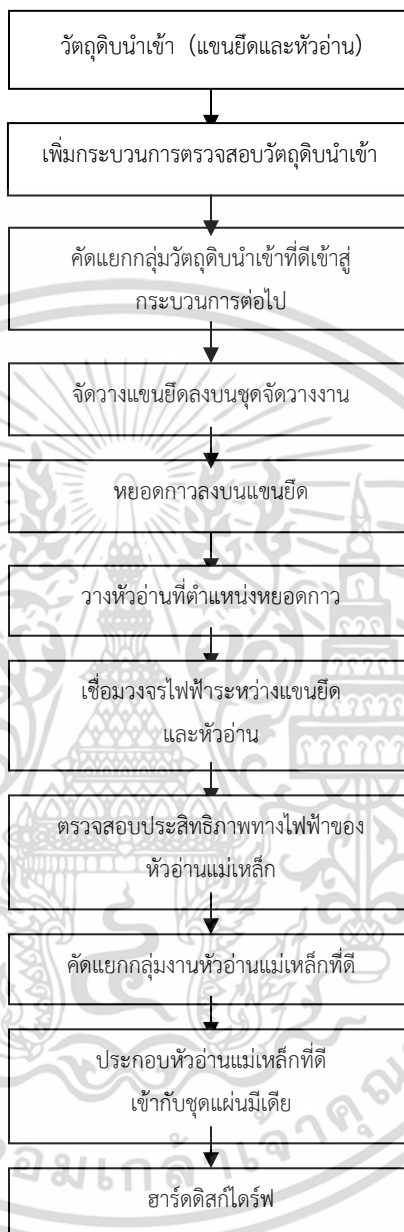


รูปที่ 2.1 ส่วนประกอบภายในฮาร์ดดิสก์ไดรฟ์ [2]

จากรูปที่ 2.1 ส่วนประกอบภายในฮาร์ดดิสก์ไดรฟ์จะประกอบไปด้วยส่วนประกอบหลักคือ แผ่นดิสก์ ส่วนของแขนและส่วนของหัวอ่านแม่เหล็ก โดยส่วนที่มีความสำคัญในกระบวนการอ่านเขียน

ข้อมูลคือ หัวอ่านแม่เหล็ก (Head Gimbal Assembly- HGA) หน้าที่ของหัวอ่านแม่เหล็กคือการอ่านและเขียนข้อมูลลงบนแผ่นดิสก์ โดยส่วนประกอบของหัวอ่านแม่เหล็กคือ แขนยึด (suspension) และหัวอ่าน (slider) ในกระบวนการผลิตในปัจจุบัน จะมีการใช้เครื่องจักรอัตโนมัติสำหรับการประกอบหัวอ่านเข้ากับแขนยึด โดยมีเครื่องจักรอัตโนมัติสำหรับจับยึดชิ้นงาน หยอดกาวลงบนชิ้นงาน และวางหัวอ่านลงบนแขนยึดด้วยค่าความผิดพลาดระดับไมโครเมตร หลังจากนั้น เริ่มกระบวนการเชื่อมวงจรไฟฟ้าระหว่างหัวอ่านและแขนยึดด้วยตะกั่วบัดกรีที่มีขนาดเพียง 45 ไมโครเมตร ซึ่งหลังจากกระบวนการข้างต้นนี้จะมีการย้ายตำแหน่งของชิ้นงานไปยังเครื่องทดสอบชิ้นงานทางไฟฟ้าเพื่อคัดแยกชิ้นงานที่ดีและชิ้นงานที่เสียออกจากกัน เพื่อเข้าสู่กระบวนการส่งชิ้นงานที่ดีให้ลูกค้าเท่านั้น ปัญหาของกระบวนการในปัจจุบันคือ ในกระบวนการคัดแยกชิ้นงานที่ดีและชิ้นงานที่เสียด้วยเครื่องทดสอบชิ้นงานทางไฟฟ้านั้น พบว่าชิ้นงานที่เสียส่วนใหญ่เป็นชิ้นงานที่ไม่ได้เสียหายจากกระบวนการในการผลิตแต่เป็นชิ้นงานที่เสียหายก่อนเข้ากระบวนการผลิต ซึ่งปัญหาของกระบวนการนี้คือการที่นำชิ้นงานที่เสียก่อนกระบวนการผลิตมาผ่านกระบวนการผลิตจริงหลายขั้นตอน จึงมีการปรับปรุงแบบการทำงานของอุตสาหกรรม โดยการเพิ่มกระบวนการตรวจสอบชิ้นงานก่อนประกอบก่อน แล้วจึงนำเพียงชิ้นงานที่ดี เข้าสู่กระบวนการประกอบต่อไป

จากรูปที่ 2.2 กระบวนการปัจจุบันในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ เริ่มต้นจากการนำวัสดุดิบนำเข้ามาคือแขนยึดและหัวอ่านเข้ามาใช้ในกระบวนการผลิตโดยตรง และเริ่มต้นกระบวนการอัตโนมัติภายในโรงงานอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ด้วยการจัดวางชิ้นงานแขนยึดให้อยู่ในตำแหน่งที่เหมาะสม จึงเริ่มการหยอดกาวลงบนตำแหน่งที่ต้องการลงบนแขนยึด หลังจากนั้น จึงใช้แขนกลอัตโนมัติหยิบชิ้นงานหัวอ่านเพื่อมาวางลงบนแขนยึด ณ ตำแหน่งที่มีการหยอดกาวไว้ หลังจากกระบวนการวางหัวอ่านลงบนแขนยึดและเชื่อมด้วยกาวแล้วนั้น กระบวนการถัดไปคือการเชื่อมวงจรหัวอ่านกับแขนยึดด้วยตะกั่วบัดกรี ด้วยการใช้พลังงานความร้อนเลเซอร์ เมื่อเสร็จสิ้นกระบวนการประกอบและเชื่อมวงจร เมื่อเสร็จสิ้นกระบวนการนี้ จะได้ชิ้นงานหลังการประกอบที่เรียกว่า หัวอ่านแม่เหล็ก กระบวนการถัดไปคือกระบวนการตรวจสอบประสิทธิภาพของหัวอ่านแม่เหล็ก ซึ่งวิธีการนี้จะเป็นการตรวจสอบทางไฟฟ้าโดยมีการนำหัวอ่านแม่เหล็กไปเคลื่อนที่เหนือแผ่นดิสก์ เพื่อทดลองอ่านเขียนและวัดค่าประสิทธิภาพผ่านผลลัพธ์ทางไฟฟ้า ซึ่งกระบวนการนี้ในปัจจุบันพบว่าหัวอ่านแม่เหล็กที่ไม่ผ่านประสิทธิภาพทางไฟฟ้าส่วนใหญ่เป็นหัวอ่านเพียงส่วนเดียวแต่แขนยึดไม่มีการเสียหายใดๆ หลังจากนั้น จะมีการคัดเลือกหัวอ่านแม่เหล็กที่ดีเท่านั้นไปประกอบเข้ากับชุดแผ่นมีเดียเป็นชุดฮาร์ดดิสก์ที่สมบูรณ์แบบก่อนนำส่งให้ลูกค้า

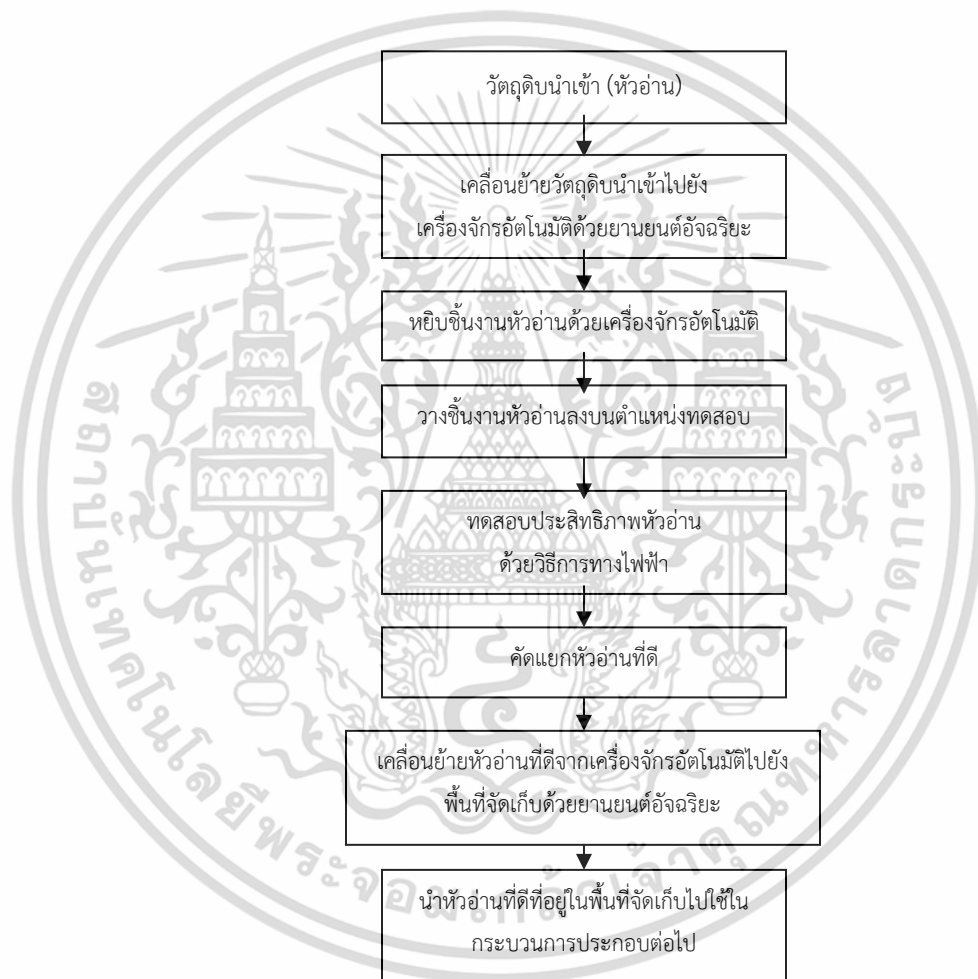


รูปที่ 2.3 กระบวนการใหม่ที่วางแผนในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์

ในส่วนของการเพิ่มขั้นตอนการตรวจสอบวัตถุดิบนำเข้านั้น มีรายละเอียดขั้นตอนส่วนนี้แสดงในรูปที่ 2.4

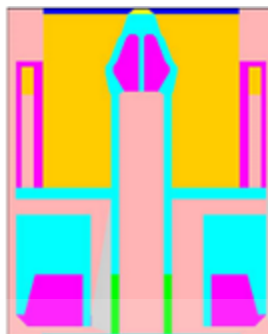
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเดินของoperator กระบวนการนี้จึงถูกปรับปรุงอีกครั้ง ในรูปที่ 2.5 เพื่อให้เหมาะสมกับการใช้งานจริงในโรงงานอุตสาหกรรม ด้วยการเปลี่ยนวิธีการเคลื่อนย้ายชิ้นงานด้วย operator เป็นการเคลื่อนย้ายชิ้นงานด้วยยานยนต์อัจฉริยะ เพื่อควบคุมการทำงานของระบบให้มีความแม่นยำมากขึ้น และลดโอกาสในความเสียหายของชิ้นงานระหว่างขนส่งได้ถ้ามีระบบควบคุมและการออกแบบที่ดี ทั้งนี้ ระบบควบคุมการทำงานของยานยนต์อัจฉริยะจะต้องมีประสิทธิภาพ เพื่อให้ระบบการทำงานภายในโรงงานอุตสาหกรรมมีประสิทธิภาพมากยิ่งขึ้น

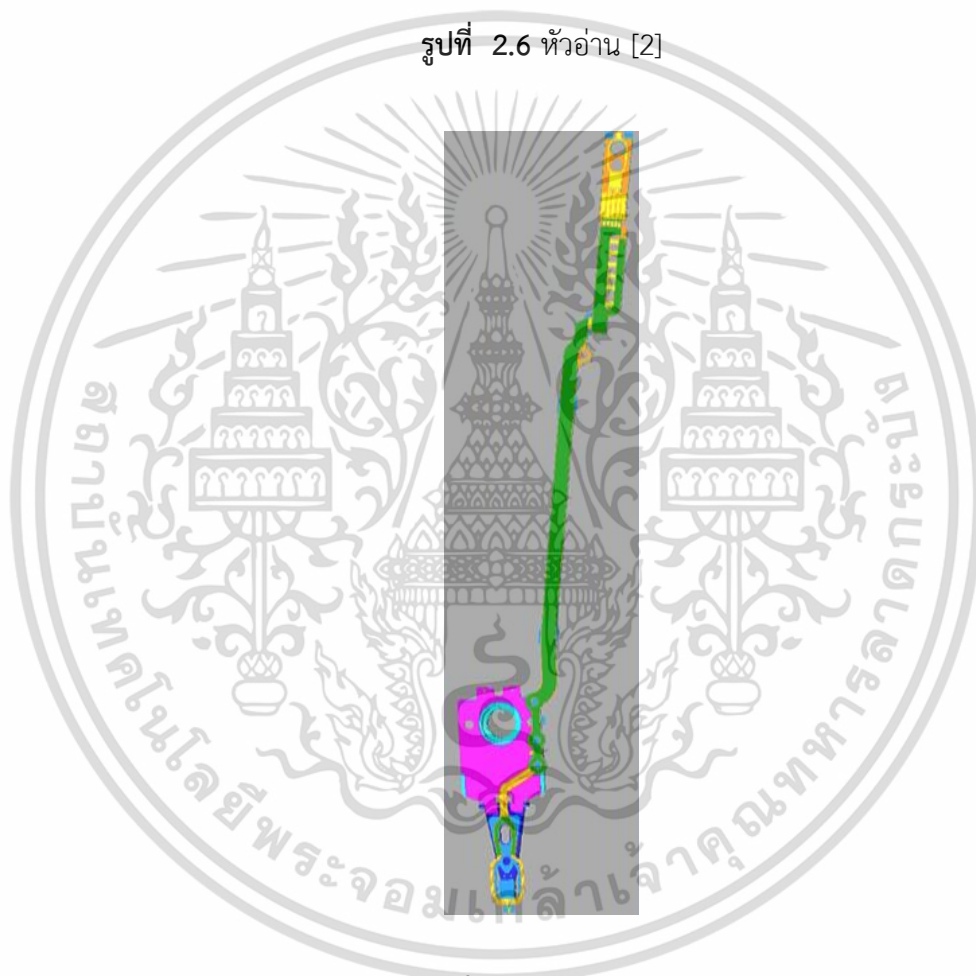


รูปที่ 2.5 กระบวนการตรวจสอบวัตถุดิบนำเข้าแบบปรับปรุงการเคลื่อนย้ายชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



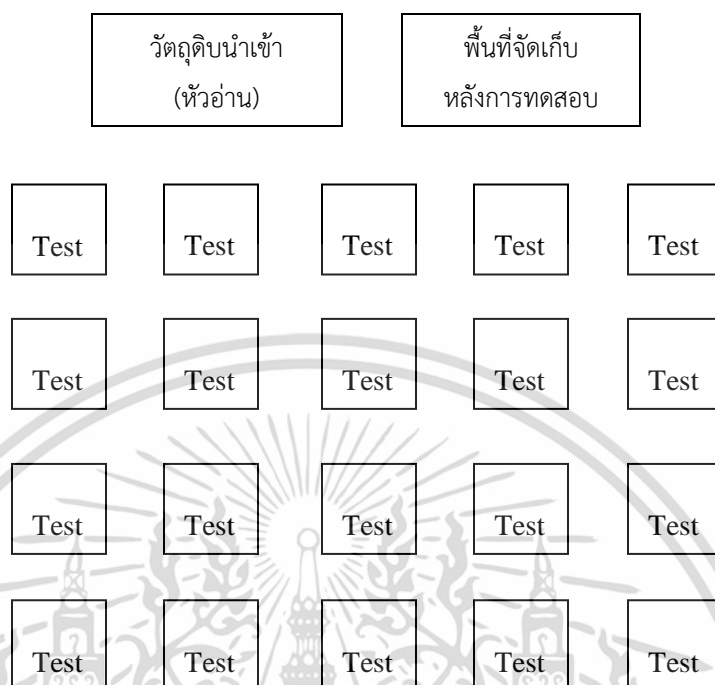
รูปที่ 2.6 หัวอ่าน [2]



รูปที่ 2.7 แขนยึด [2]

จากรูปที่ 2.5 และ 2.6 คือรูปตัวอย่างของหัวอ่านและแขนยึด ซึ่งเมื่อประกอบกันด้วยกระบวนการอัตโนมัติในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์จะได้ผลลัพธ์ออกมาเป็นชุดหัวอ่านแม่เหล็ก (HGA) โดยหัวอ่านเป็นส่วนประกอบที่สำคัญเพราะมีวงจรสำหรับกระบวนการอ่านเขียนอยู่ภายในหัวอ่าน แต่แขนยึดเป็นเส้นไฟสำหรับเชื่อมวงจรของหัวอ่านแม่เหล็กไปยังส่วนประกอบอื่นๆในฮาร์ดดิสก์ไดรฟ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แบบเบื้องต้นของการจัดวางพื้นที่ทำงานของยานยนต์อัจฉริยะ

จากรูปที่ 2.8 เป็นแบบเบื้องต้นโดยประมาณของการจัดวางพื้นที่แต่ละสถานีทำงานของยานยนต์อัจฉริยะภายในโรงงานอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ จากแบบดังกล่าวยานยนต์อัจฉริยะจะเริ่มต้นจากการเคลื่อนที่ที่สถานีที่จัดเก็บวัตถุดิบนำเข้า (หัวอ่าน) แล้วเคลื่อนย้ายหัวอ่านที่ยังไม่ได้รับการทดสอบมายังเครื่องทดสอบ (test) ซึ่งเครื่องทดสอบจะต้องใช้เวลาในการทดสอบ ในระหว่างนั้น ยานยนต์อัจฉริยะสามารถที่จะนำส่งหัวอ่านที่ยังไม่ได้ทดสอบไปให้กับเครื่องทดสอบอื่นๆ ได้ เมื่อกระบวนการทดสอบเสร็จสิ้น ยานยนต์อัจฉริยะจะเคลื่อนย้ายหัวอ่านที่ได้รับการทดสอบผ่านแล้วไปยังพื้นที่จัดเก็บหัวอ่านหลังการทดสอบ เพื่อเข้าสู่กระบวนการประกอบในกระบวนการถัดไป

2.2 การวางแผนเส้นทางการเคลื่อนที่ของยานยนต์อัจฉริยะ (Path planning)

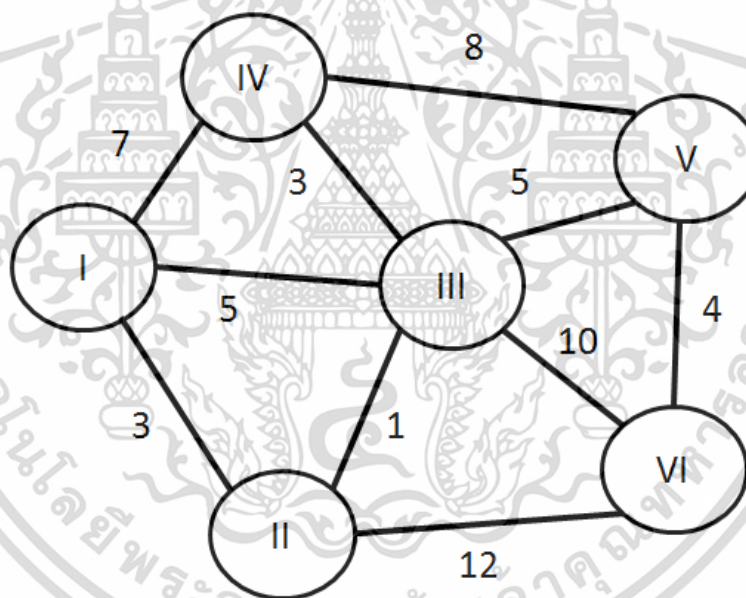
การวางแผนเส้นทางการเคลื่อนที่ของยานยนต์อัจฉริยะ คือการกำหนดเส้นทางการเคลื่อนที่ที่สั้นที่สุดให้กับยานยนต์อัจฉริยะ จากตำแหน่งเริ่มต้นไปยังตำแหน่งเป้าหมาย เส้นทางการที่ดีที่สุดของการวางแผนเส้นทางให้ยานยนต์อัจฉริยะคือการค้นหาเส้นทางที่มีจำนวนรอบในการทำงานน้อยที่สุด หรือเส้นทางที่มีอัตราการเบรคน้อยที่สุด ในการวางแผนเส้นทางการเคลื่อนที่ สิ่งสำคัญเป็นอันดับแรกคือแผนที่ภายใต้สภาพแวดล้อมนั้นๆ และยานยนต์ที่ต้องการเคลื่อนที่ในแผนที่นั้น

2.2.1 แผนที่ (Map representation)

แผนที่เป็นส่วนสำคัญของกระบวนการวางแผนเส้นทางการเคลื่อนที่ของยานยนต์อัจฉริยะ เพื่อใช้ในการเป็นต้นแบบของแบบจำลองในคอมพิวเตอร์ ลักษณะของแผนที่ที่นำเสนอมีอยู่สองประเภท

2.2.1.1 Topological maps

Topological maps [3] คือแผนที่แบบ continuous coordinates มีลักษณะเหมือนกราฟเชื่อมโยงโดยเชื่อมต่อแต่ละจุดยอด (vertex) เข้าด้วยกัน แต่ละจุดยอดนั้นอาจเรียกว่า node ซึ่งมีการเชื่อมโยงกันผ่านเส้นเชื่อม (edges) โดยสามารถรวมพื้นที่พื้นที่หนึ่ง เป็น 1 จุดยอดได้ และใช้เส้นเชื่อมเพื่อเชื่อมโยงระหว่างเส้นทาง โดยภายใต้พื้นที่ที่มีสิ่งกีดขวาง จะถูกระบุไว้เป็นขอบเขตในรูปหลายเหลี่ยม (polygon) ชื่อข้อเสียของแผนที่แบบนี้คือ การแสดงผลที่อาจจะไม่ชัดเจนเท่ากับแบบ Occupancy grid maps



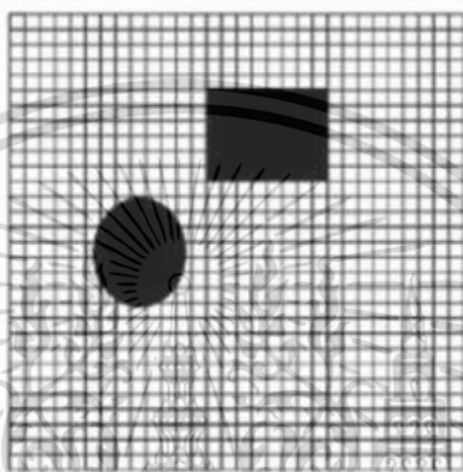
รูปที่ 2.9 แผนที่แบบ topological maps [3]

2.2.1.2 Occupancy grid maps

Occupancy grid maps [3] คือแผนที่แบบ discrete coordinates ซึ่งจะมีการแบ่งพื้นที่ย่อยๆ ออกเป็นพื้นที่ที่เท่ากันในลักษณะแบบตาราง หรืออาจจะเป็นพื้นที่ที่มีขนาดแตกต่างกันเช่น ในห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรืออาคาร ซึ่งการแสดงแผนที่แบบนี้เป็นแผนที่ที่นิยมใช้มากที่สุดในงานวิจัยของหุ่นยนต์ภายใต้วัตถุประสงค์ต่างๆกันแต่ละจุดสามารถกำหนดเงื่อนไขการมีสิ่งกีดขวางหรือกำหนดค่าความน่าจะเป็นในการมีสิ่งกีดขวาง ข้อเสียของแผนที่แบบนี้คือการจัดเก็บข้อมูลขนาดใหญ่ภายใต้การคำนวณแต่ละช่องกริดตามรูปที่ 2.10 ที่มีจำนวนมาก



รูปที่ 2.10 แผนที่แบบ occupancy grid maps [3]

2.2.2 วิธีการวางแผนเส้นทางการเคลื่อนที่ (path planning algorithm)

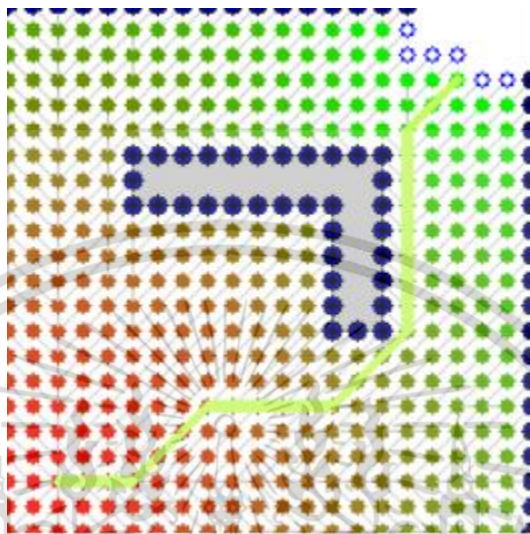
วิธีการวางแผนเส้นทางการเคลื่อนที่คือการหาเส้นทางที่สั้นที่สุดจากจุดยอดหนึ่งผ่านการเชื่อมโยงเส้นกราฟไปยังตำแหน่งต่างๆที่ต้องการ การคำนวณเส้นทางที่สั้นที่สุดคือการค้นหาผลรวมของฟังก์ชันของเส้นเชื่อมที่น้อยที่สุด ซึ่งในงานวิจัยที่เกี่ยวข้องกับหุ่นยนต์คือเส้นทางการเคลื่อนที่ของหุ่นยนต์หรืออาจจะเป็นเวลาหน่วงที่น้อยที่สุดเมื่อเป็นงานวิจัยที่เกี่ยวข้องกับระบบ network หรืออาจจะเป็นค่าฟังก์ชันที่น้อยที่สุดในงานวิจัยเฉพาะทางนั้นๆ วิธีการในการหาเส้นทางที่ดีที่สุดในปัจจุบัน มีดังนี้

2.2.2.1 Dijkstra's algorithm

วิธีการ Dijkstra's algorithm [4] คือวิธีการที่ใช้กันมายาวนานและเป็นวิธีการอย่างง่าย ถูกนำเสนอโดย Dutch ในปี 1959 โดยการเริ่มต้นจากจุดยอดเริ่มต้น ณ ตำแหน่งเส้นทางเริ่มต้น วิธีการนี้จะทำการ คำนวณค่าเส้นทางจากจุดเริ่มต้นเทียบกับจุดที่ใกล้เคียงทั้งหมด และเลือกใช้เส้นทางที่ได้ผลลัพธ์การคำนวณน้อยที่สุดจากทุกๆจุดยอดข้างเคียง หลังจากมีการตรวจสอบครบทุกจุดยอดข้างเคียงแล้วนั้น วิธีการนี้จะทำการเลือกจุดต่อไปที่ทำให้ผลลัพธ์การคำนวณน้อยที่สุดจนถึงตำแหน่งจุดยอดเป้าหมาย (goal) วิธีการนี้จะสั้นสุดและหุ่นยนต์จะเดินในเส้นทางที่ผลลัพธ์ของการคำนวณตลอดเส้นทางมีค่าน้อยที่สุด ซึ่งจะเห็นได้ว่า วิธีการข้างต้นนี้ ใช้การคำนวณในทุกๆจุดภายในแผนที่ โดยที่บางจุดยอดอาจให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ที่สามารถตัดออกได้อยู่แล้ว จึงทำให้ส่วนนี้เป็นหนึ่งในข้อเสียของวิธีการนี้ที่อาจจะส่งผลต่อเวลาในการประมวลผลที่นานจากการเลือกคำนวณทุกๆจุดทั้งจุดที่จำเป็นและจุดที่ไม่จำเป็น



รูปที่ 2.11 ตัวอย่างการค้นหาเส้นทางที่ดีที่สุดด้วย Dijkstra's algorithm [4]

2.2.2.2 A* algorithm

วิธีการ A* algorithm [5] ถูกนำเสนอโดย Peter Hart, Nils Nilsson ,และ Bertram Raphael ในปี 1968 คือวิธีการที่นำข้อเสียของวิธีการ Dijkstra's algorithm มาปรับปรุงโดยการใช้ heuristic function โดยการจัดกลุ่มของ nodes ที่มีค่าประมาณของเส้นทางถึงเป้าหมายที่สั้นที่สุดเมื่อเทียบกับกลุ่ม nodes อื่นๆ ซึ่งกระบวนการนี้ต้องทำการคิดทุก node แบบที่ไม่ใช่แค่ระยะทางของเส้นทางจริงเท่านั้น นั่นคือจะมีการคำนวณด้วยค่าระยะทางโดยประมาณโดยการคำนวณด้วย Euclidean distance ในสมการที่ (2.1) หรือ Manhattan distance ในสมการที่ (2.2) ของตำแหน่งจุดยอดนั้นๆ (x_i, y_i) ถึงเป้าหมาย (x_g, y_g) ซึ่งด้วยวิธีการคำนวณข้างต้นนี้จะทำให้หุ่นยนต์สามารถถึงเป้าหมายได้เร็วกว่าแบบ Dijkstra's algorithm

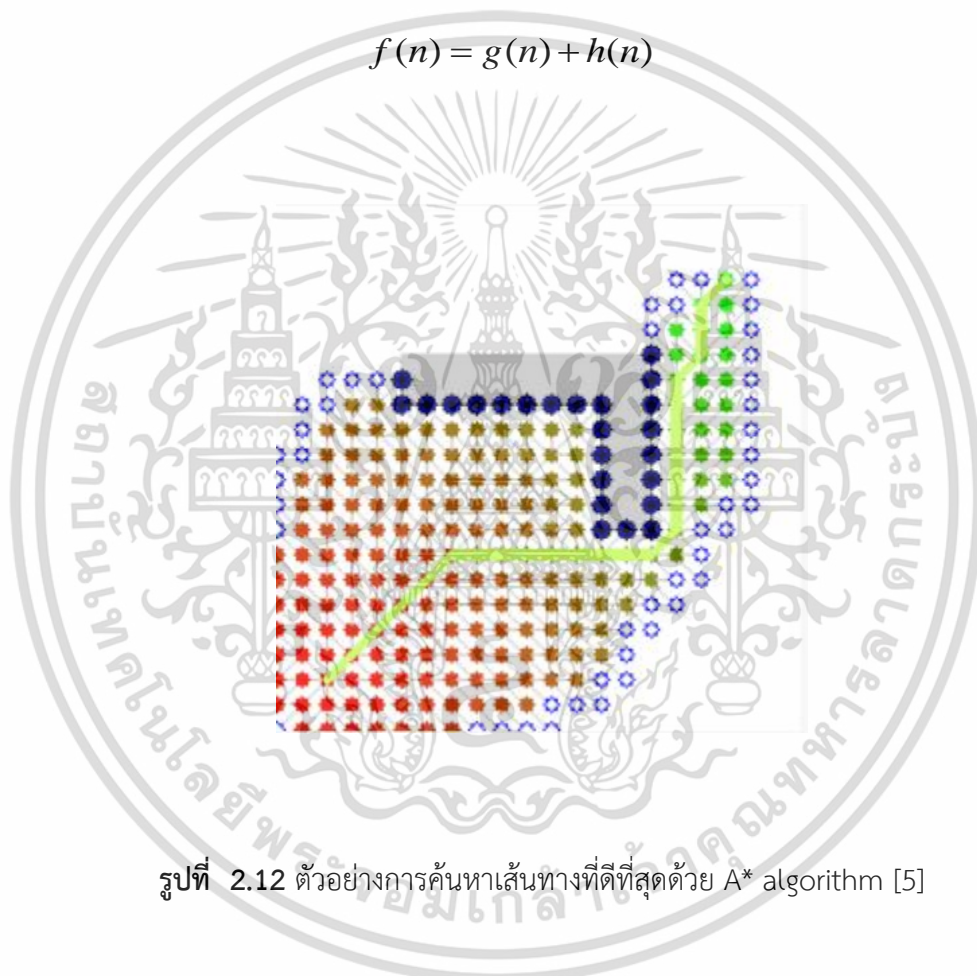
$$D = \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2} \quad (2.1)$$

$$D = |x_i - x_g| + |y_i - y_g| \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณเส้นทางของ A* algorithm คือการคำนวณเส้นทางจากจุดเริ่มต้นถึงเป้าหมายด้วยผลรวมของค่าระยะทางจริงกับค่าของระยะทางที่ประมาณการ จากสมการที่ 2.3 $f(n)$ แทนค่าของเส้นทางรวมของ A* , $g(n)$ คือค่าระยะทางของเส้นทางจากจุดเริ่มต้นไปยังจุดใดๆ และค่า $h(n)$ คือค่าของระยะทางโดยประมาณจากจุดใดๆ ไปยังตำแหน่งเป้าหมาย ซึ่งจะทำให้วิธีการของ A* นั้นมีความรวดเร็วในการวางแผนเส้นทางที่เคลื่อนที่มากกว่าวิธีการ Dijkstra เพราะการพิจารณาจำนวน node ที่ลดน้อยลงจากการประเมินค่าของผลรวมเส้นทางด้วย $h(n)$ หรือ heuristic term

$$f(n) = g(n) + h(n) \quad (2.3)$$



รูปที่ 2.12 ตัวอย่างการค้นหาเส้นทางที่ดีที่สุดด้วย A* algorithm [5]

2.2.2.3 D* algorithm

วิธีการ D* algorithm [6] ถูกนำเสนอโดย Anthony Stentz ในปี 1994 คือวิธีการที่นำ A* algorithm มาปรับปรุงแก้ไข เมื่อมีสิ่งกีดขวาง โดยสามารถปรับค่าผลลัพธ์ได้ในเส้นทางที่มีสิ่งกีดขวาง นั่นคือ D* สามารถที่จะปรับเปลี่ยนแผนของเส้นทางที่เคลื่อนที่ของหุ่นยนต์รอบๆ สิ่งกีดขวางได้ในขณะที่ยังคงไว้ซึ่งการคำนวณเส้นทางอื่นๆ ที่คำนวณไว้แล้ว หรือกล่าวได้ว่า D* คือ Dynamic A* ซึ่งทั้งวิธี A* และ D* จะทำได้ยาก เมื่อแผนที่มีความซับซ้อน การค้นหาและการคำนวณจะใช้เวลานาน

2.2.2.4 Rapidly-Exploring Random Trees(RRT) algorithm

วิธีการ Rapidly-Exploring Random Trees(RRT) algorithm [7] ถูกนำเสนอโดย Steven M. La Valle ในปี 1998 คือวิธีการที่ใช้การสุ่มค่าเพื่อค้นหาค้นหาพื้นที่ขนาดใหญ่ วิธีการนี้จะเลือกจุดแบบสุ่มภายใต้สภาพแวดล้อมนั้นๆ และเชื่อมโยงจุดแบบสุ่มนั้นกับจุดยอดเริ่มต้น หลังจากนั้นจุดแบบสุ่มถัดไปจะเชื่อมโยงกับจุดยอดที่ใกล้ที่สุดในกราฟจนเชื่อมโยงไปถึงจุดเป้าหมาย มีลักษณะการสร้างเหมือนต้นไม้ที่แตกกิ่งก้านสาขาออกไปเรื่อยๆตามค่า iteration ในแบบจำลองนั้นๆ การเพิ่มขึ้นของ RRT หรือการเพิ่มขึ้นของรากที่แตกกิ่งก้านสาขานั้น ขึ้นกับค่าความน่าจะเป็นในแต่ละส่วนของพื้นที่นั้นๆ บนพื้นที่ที่มีความน่าจะเป็นน้อยจะถูกจัดอยู่ในกลุ่มตัวอย่าง ส่วนบนพื้นที่ที่มีความน่าจะเป็นสูงจะมีการเติบโตของต้นไม้ไปยังจุดยอดเป้าหมายต่อไป รูปที่ 2.13 แสดงตัวอย่างการค้นหาเส้นทางที่ดีที่สุดด้วย Rapidly-Exploring Random Trees(RRT) เมื่อ iteration เพิ่มขึ้น ข้อเสียของวิธีการนี้คือจะมีการใช้เวลาในการคำนวณที่นานเนื่องจากการค้นหาเส้นทางที่ดีที่สุดภายใต้จำนวนกิ่งก้านสาขาจำนวนมาก วิธีการนี้จึงเหมาะกับแผนที่ที่มีขนาดใหญ่



รูปที่ 2.13 ตัวอย่างการค้นหาเส้นทางที่ดีที่สุดด้วย Rapidly-Exploring Random Trees(RRT) [7]

2.3 การค้นหาค่าที่ดีที่สุด (Optimization methods)

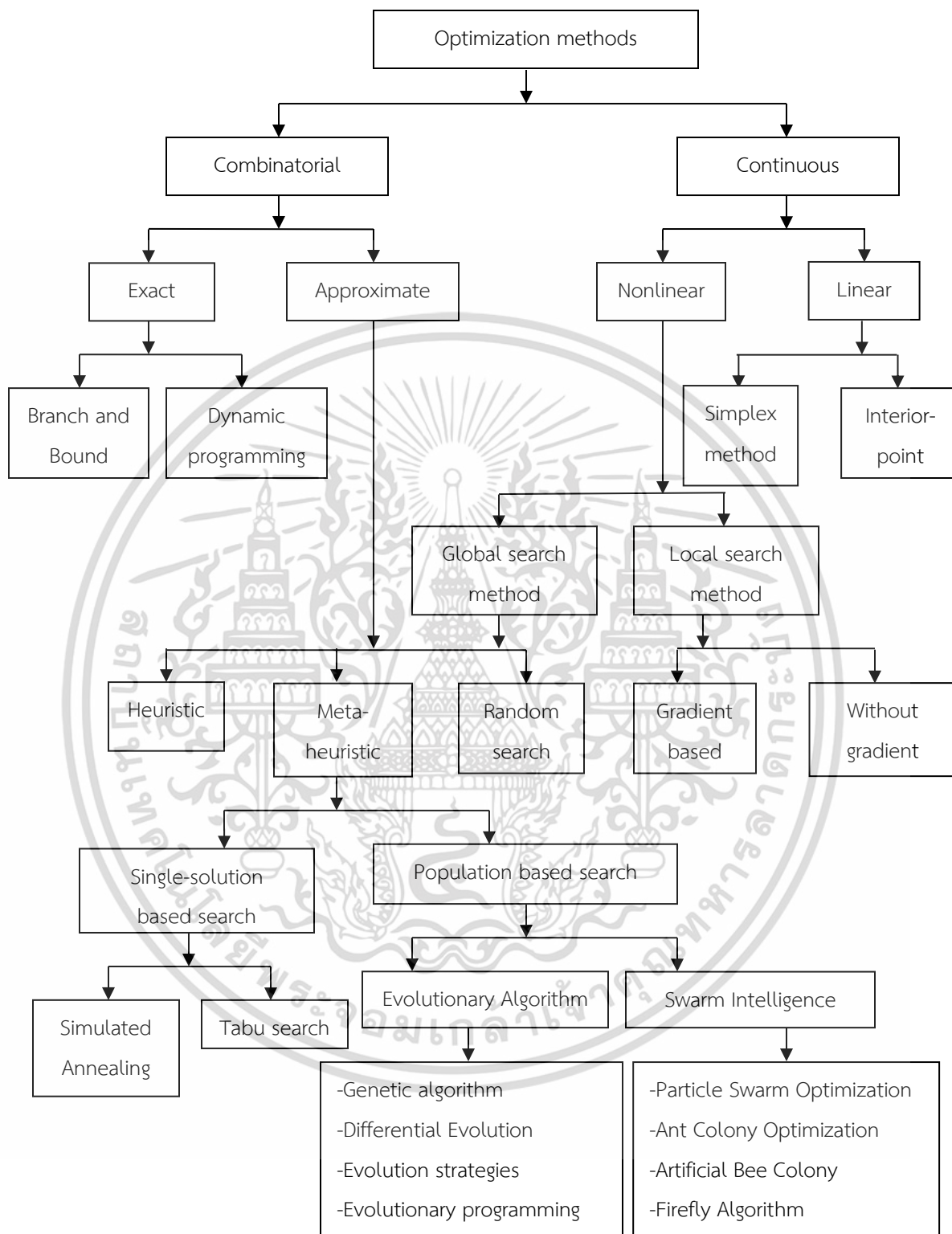
ในปัจจุบัน มีวิธีการค้นหาค่าที่ดีที่สุดหลากหลายวิธี โดยการเลือกใช้ จะต้องเลือกให้เหมาะสมกับลักษณะข้อมูลของปัญหา โดยหลักการเบื้องต้นของการค้นหาค่าที่ดีที่สุดคือการสร้างฟังก์ชันวัตถุประสงค์ (Objective function) ที่สนใจ โดยฟังก์ชันนี้จะเป็ฟังก์ชันที่นำไปใช้ในกระบวนการค้นหาค่าที่ดีที่สุด ซึ่งอาจจะเป็นค่ามากที่สุด หรือค่าน้อยที่สุดก็ได้ ทั้งนี้ขึ้นกับความต้องการของผู้ออกแบบ นอกจากนี้ ลักษณะของข้อมูลนำเข้าเป็นหนึ่งในข้อมูลสำคัญในการเลือกใช้วิธีการค้นหาค่าที่ดีที่สุดให้เหมาะสมกับงานนั้นๆ

จากสมการที่ (2.4) ฟังก์ชันวัตถุประสงค์จะแปรตามตัวแปรที่มีหลายชั้น โดยตัวแปรดังกล่าวเรียกว่าตัวแปรต้น (x_1, x_2, \dots, x_n) โดยการหาค่าที่ดีที่สุดนี้จะเป็นการหาค่าตัวแปรต้นที่เหมาะสมที่สุดเพื่อให้ได้ค่าของฟังก์ชันวัตถุประสงค์ที่เหมาะสมที่สุดซึ่งอาจจะเป็นค่ามากที่สุดหรือน้อยที่สุดตามที่ต้องการ ซึ่งในการแก้ปัญหา ตัวแปรต้นแต่ละตัวอาจมีเงื่อนไข หรือข้อจำกัด เช่นเงื่อนไขที่ห้ามเป็นค่าติดลบ ($x \geq 0$) หรือค่าตัวแปรต้นมากกว่า 3 ($x > 3$)

$$f = f(x_1, x_2, \dots, x_n) \quad (2.4)$$

รูปที่ 2.14 แสดงโครงสร้างของวิธีการหาค่าที่ดีที่สุด (Optimization method) [8] ซึ่งประกอบไปด้วย 2 ส่วนหลักๆคือ combinatorial และ continuous โดยวิธีการที่เป็นที่นิยมในปัจจุบันคือการประมาณค่าแบบ meta-heuristic ภายใต้ population based search ในงานวิจัยนี้ได้เลือกนำวิธีการ meta-heuristic ซึ่งเป็นการประมาณค่าตอบที่มีความน่าเชื่อถือ โดยแบ่งออกเป็น 2 แบบหลักๆ คือแบบ Single-solution based search ซึ่งเป็นการค้นหาแบบ local search ภายใต้ตัวอย่างหนึ่งกับกลุ่มใกล้เคียงตัวอย่างอื่นๆ กับ Population based search ซึ่งสามารถค้นหาได้ทั้ง local search และ global search เนื่องจากเป็นการค้นหาแบบหลายตำแหน่งในพื้นที่ค้นหา โดยในงานวิจัยนี้เลือกแบบ Population based search เพราะการเลือกใช้ Single-solution based search อาจจะทำให้ไม่สามารถค้นหาเจอค่า global optimum ได้

วิธีการของ population based search นั้น จะใช้จุดเริ่มต้นที่แตกต่างกัน กระจายตัวออกไป ภายใต้พื้นที่ค้นหา ซึ่งทำให้ค่าความน่าจะเป็นในการค้นหา global optimum นั้นมีค่าสูงกว่าแบบ Single-solution based search ในงานวิจัยนี้ขอเสนอทฤษฎีเบื้องต้นของ 4 วิธีการที่เป็นที่นิยมใช้ในปัจจุบัน คือ Genetic Algorithm , Particle Swarm Optimization, Ant Colony Optimization และ Artificial Bee Colony



รูปที่ 2.14 โครงสร้างของวิธีการค้นหาที่ดีที่สุด (Optimization method) [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 Genetic algorithms

วิธีการค้นหาค่าที่ดีที่สุด ด้วยวิธี Genetic algorithms (GA) [8] เป็นกระบวนการ meta-heuristic แบบ Evolutionary algorithm ที่นิยมใช้อย่างแพร่หลาย วิธีการ Genetic Algorithms ถูกนำเสนอขึ้นโดย John Holland ในปี 1992 โดยวิธีการนี้ ใช้หลักการของวิวัฒนาการในเชิงชีววิทยา (biological evolution) สำหรับการคำนวณหาผลเฉลยที่เหมาะสมที่สุด หรือที่เรานิยมเรียกกันว่า Optimal Solution ในขั้นตอนเชิงพันธุกรรมนั้น มีการเปรียบเทียบสิ่งต่างๆทางชีววิทยา

- โครโมโซม หมายถึง ลำดับของยีนซึ่งเป็นตัวแทนของผลเฉลย
- ยีน หมายถึง ค่าที่เป็นไปได้ในแต่ละหน่วยย่อยโครโมโซม
- ประชากร หมายถึง กลุ่มของโครโมโซม

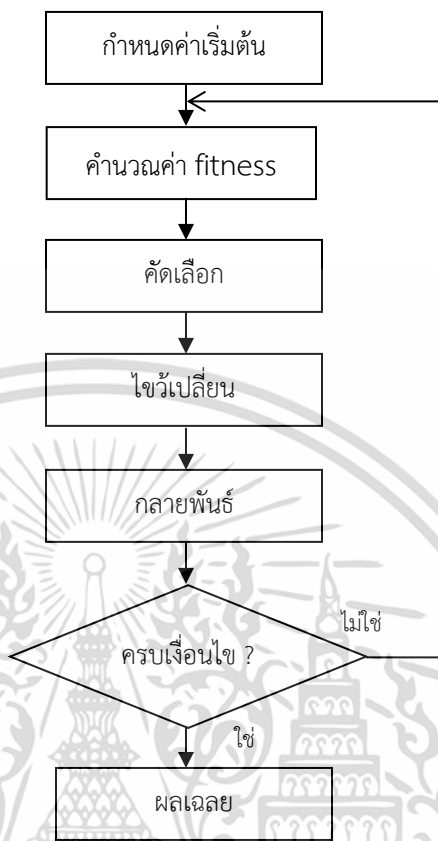
โดยขั้นตอนเชิงพันธุกรรมนั้น อาศัยการอยู่รอดและสืบพันธุ์มาใช้

- การสืบพันธุ์แบบใช้เพศ หมายถึง การที่ประชากรรุ่นต่อไปเกิดจากความแตกต่างของ 2 ครอบครั้ว หรือการไขว้เปลี่ยน (crossover)
- การสืบพันธุ์แบบไม่อาศัยเพศ หมายถึง การที่ประชากรรุ่นต่อไปเกิดจากครอบครัวเดียว เช่นการกลายพันธุ์ (Mutation)
- การคัดเลือก หมายถึง การอาศัยค่า fitness หรือค่าความเหมาะสมค่าหนึ่ง ในการคัดเลือกรุ่นต่อไป

ในส่วนของวิธีการคัดเลือกลักษณะนั้น ในเชิงของขั้นตอนเชิงพันธุกรรมคือ การเปรียบเทียบผู้ที่แข็งแรงกว่า เป็นผลเฉลยที่เหมาะสม หรือค่า fitness สูง โดยการเลือกใช้ fitness function นั้นๆ จะต้องเลือกให้เหมาะสมและสอดคล้องกับเป้าหมายที่สนใจ ตัวอย่างวิธีการในกระบวนการคัดเลือก เช่น

- 1) **การเก็บเฉพาะอันดับต้น** วิธีการนี้เป็นการคัดเลือกที่นำโครโมโซมมาเรียงลำดับกันตามค่า fitness จากน้อยไปมาก แล้วทำการคัดเลือกเฉพาะ fitness ที่มีค่ามากเท่านั้น สำหรับการเป็นโครโมโซมในรุ่นถัดไป ในส่วนของโครโมโซมที่เหลือที่มีค่า fitness น้อย จะถูกกำจัดทิ้งไป
- 2) **วงล้อรูเล็ต (Roulette wheel)** วิธีการนี้เป็นการใช้ค่า fitness แทนความน่าจะเป็นในการดำรงอยู่ของโครโมโซม นั่นคือ โดยโครโมโซมที่มีค่า fitness สูง จะมีโอกาสดำรงอยู่มากกว่าโครโมโซมที่มีค่า fitness ต่ำ ซึ่งวิธีการนี้อาจจะมีโครโมโซมที่มีค่า fitness น้อย ผ่านไปยังรุ่นถัดไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ขั้นตอนของวิธีเชิงพันธุกรรม (Genetic algorithm)

จากรูปที่ 2.15 แสดงขั้นตอนวิธี Genetic algorithm (GA) หรือขั้นตอนวิธีเชิงพันธุกรรม เริ่มต้นด้วยการสร้างประชากร หรือผลเฉลยของปัญหาที่เป็นไปได้จำนวนมากก่อน โดยใช้วิธีการสุ่ม หลังจากนั้นทำการคำนวณหาค่า fitness ของแต่ละผลเฉลยนั้นๆ เพื่อเข้าสู่กระบวนการคัดเลือก และแทนที่ผลเฉลยที่ถูกกำจัดออกด้วยกลุ่มประชากรใหม่ ทำซ้ำแบบนี้ไปเรื่อยๆจนกว่าจะได้ผลเฉลยที่ค่า fitness มีค่าสูงที่สุดตามที่ต้องการหรือถึงจำนวนการวนซ้ำที่กำหนดไว้ ทั้งนี้ หากค่า fitness ที่ตั้งไว้ไม่สามารถหาค่าได้จริงในระหว่างกระบวนการ ค่ากำหนดของการวนซ้ำสูงสุดจะทำให้โปรแกรมสามารถสิ้นสุดลงได้โดยไม่ใช่เวลายาวนานเกินไปหรือเกิดเหตุการณ์ที่โปรแกรมไม่สามารถวนออกจากการคำนวณได้ นอกจากนี้ การกำหนดประชากรในช่วงแรกมีความสำคัญเป็นอย่างมาก โดยปกติจะมีการกำหนดกลุ่มประชากรให้ไม่ซ้ำกันเลย ซึ่งหากประชากรมีการซ้ำกัน จะทำการเก็บค่าเป็นประชากรเดียว โดยภาพรวมแล้วนั้น วิธี Genetic algorithm อาศัยการสุ่มในกระบวนการ ซึ่งจะแตกต่างจากวิธีการ Brute force ที่เป็นการคำนวณทุกเงื่อนไขที่เกิดขึ้นได้แต่ใช้เวลาในการคำนวณนาน จากข้อสรุปนี้ แปลว่า วิธี Genetic algorithm จะให้ผลลัพธ์การคำนวณที่รวดเร็วกว่าวิธีการ Brute force เป็นอย่างมาก แต่อาจจะไม่ได้คำตอบที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จริงๆออกมาก็ได้ เนื่องจากโปรแกรมถูกกำหนดการสิ้นสุดด้วยค่า fitness ที่ต้องการ หรือจำนวนรอบมากที่สุดในการคำนวณ ลักษณะดังกล่าวนี้เรียกว่าการติดกับ local optima หรือ local trap

2.3.2 Particle Swarm Optimization

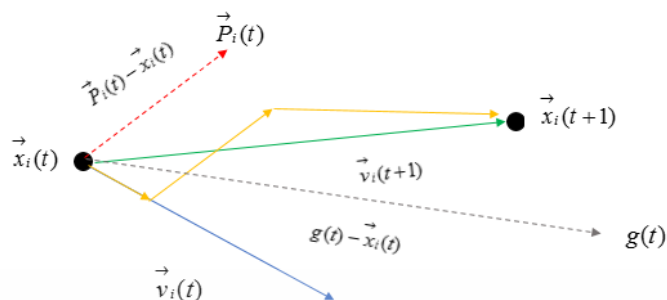
วิธีการค้นหาค่าที่ดีที่สุด ด้วยวิธี Particle Swarm Optimization (PSO) [8] เป็นกระบวนการ meta-heuristic แบบ swarm Intelligence โดยมีการค้นหาคำตอบที่ดีที่สุดแบบไม่มีเงื่อนไข คำตอบของกระบวนการคือค่าของฟังก์ชันวัตถุประสงค์ (objective function) ซึ่งผู้ออกแบบสามารถออกแบบการหาค่ามากที่สุดหรือน้อยที่สุดก็ได้ วิธีการ PSO ถูกคิดค้นโดย Kennedy และ Eberhart ในปี 1995 โดยการศึกษาจากลักษณะการบินหาอาหารของฝูงนก ซึ่งนกจะบินไปหาดำแหน่งที่สมาชิกในฝูงเจอแหล่งอาหารที่อุดมสมบูรณ์เสมอ และมีลักษณะการบินแบบเกาะกลุ่มกัน ฝูงนกเหล่านั้นมีการส่งสัญญาณเพื่อใช้ในการสื่อสารกันถึงตำแหน่งที่มีอาหาร และเคลื่อนที่ไปยังแหล่งอาหารนั้นๆ จากแนวคิดนี้ จึงเป็นที่มาของวิธีการ PSO ตัวแปรหลักที่ใช้ในกระบวนการ PSO นั่นคือ ค่าตำแหน่ง และค่าความเร็ว โดยมีการค้นหาคำตอบด้วยการใช้ออนุภาค หรือ particles จำนวนมาก ให้มีการเคลื่อนที่ไปยังพื้นที่ค้นหา (search space) เพื่อค้นหาคำตอบที่ดีที่สุด หลักการการคำนวณคือเปรียบเทียบแต่ละตัวเป็นอนุภาค ที่สามารถจดจำตำแหน่งและทิศทางของตนเองได้ (x_i) และสามารถจดจำความเร็วในการเคลื่อนที่ของตนเองได้ด้วย (v_i) นอกจากนั้นวิธีการ PSO จะมีการคำนวณค่าของฟังก์ชันวัตถุประสงค์ตลอดเวลา เมื่อพบว่าเป็นค่าที่ดีที่สุดจะจัดเก็บข้อมูลตำแหน่งที่ดีที่สุดของตนเอง หรือเรียกว่า P_{Best} และข้อมูลตำแหน่งที่ดีที่สุดจากอนุภาคทุกตัว จะเรียกว่า G_{Best}

$$v_{ij}^{t+1} = wv_{ij}^t + r_1^t c_1 (p_{ij}^t - x_{ij}^t) + r_2^t c_2 (g_{ij}^t - x_{ij}^t)(t) \quad (2.5)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (2.6)$$

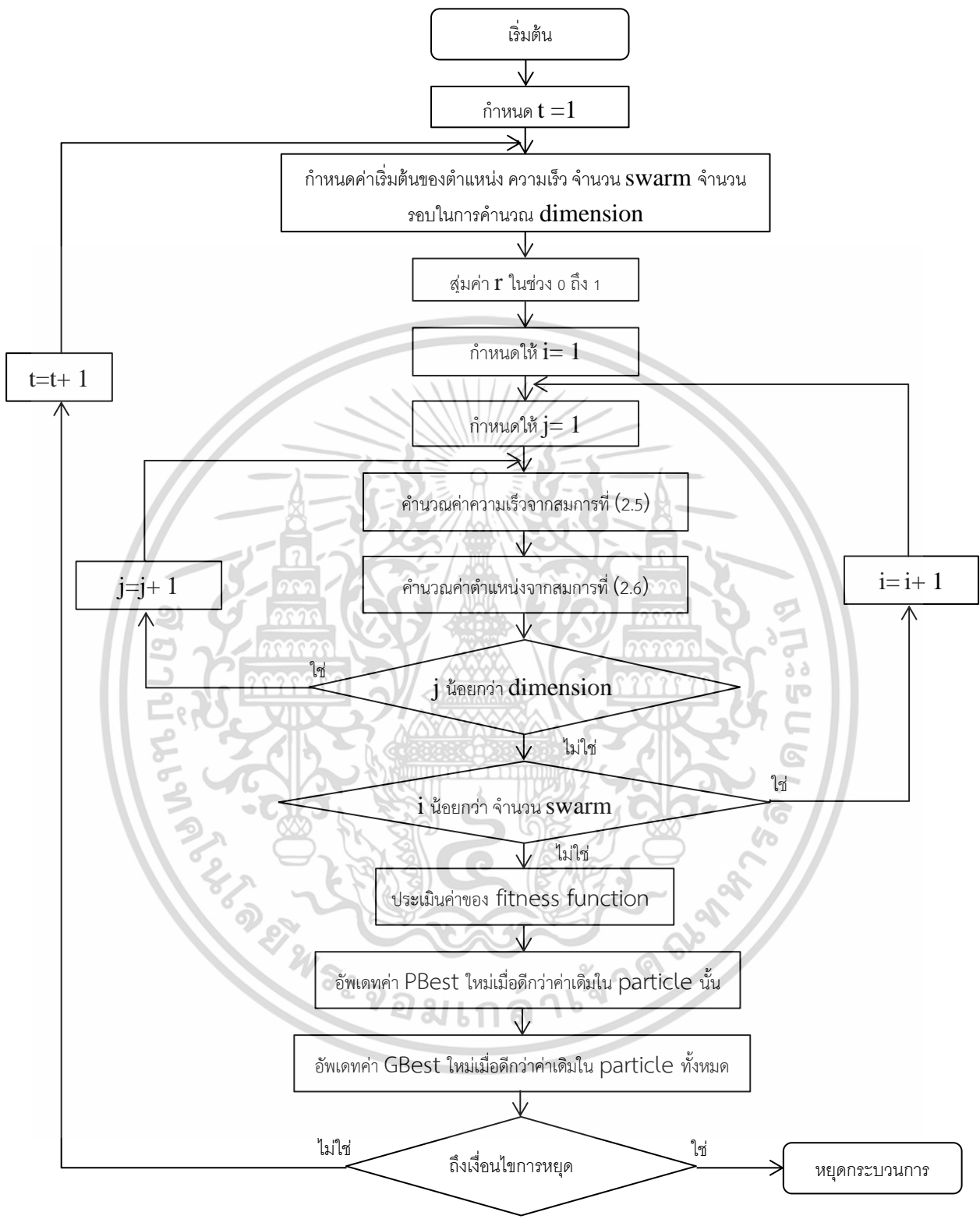
สมการที่ (2.5) แสดงถึงการคำนวณความเร็วในการเคลื่อนที่ของอนุภาค ณ ตำแหน่งถัดไป (v_{ij}^{t+1}) เมื่อ w คือค่าความเฉื่อย (inertia) , r_1^t และ r_2^t แทนค่าตัวเลขแบบสุ่มที่เป็นบวก และ c_1 กับ c_2 แทนค่าของสัมประสิทธิ์ของความเร่ง p_{ij}^t แทนค่าของตำแหน่งที่ดีที่สุดของอนุภาคนั้น และ g_{ij}^t แทนค่าของตำแหน่งที่ดีที่สุดของทุกอนุภาค

ในสมการที่ (2.6) แสดงถึงการคำนวณตำแหน่งถัดไปของอนุภาค (x_{ij}^{t+1}) จากผลบวกของตำแหน่งปัจจุบัน (x_{ij}^t) กับความเร็วถัดไปของอนุภาค (v_{ij}^{t+1}) โดยเป็นผลบวกในรูปของเวกเตอร์ ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 เวกเตอร์ค่านวนของวิธี PSO

ขั้นตอนการค้นหาค่าที่ดีที่สุดด้วยวิธี PSO แสดงดังรูปที่ 2.17 โดยเริ่มต้นจากการกำหนดค่าเริ่มต้นของตำแหน่งอนุภาค ความเร็ว จำนวนอนุภาค และจำนวนรอบมากที่สุดในการคำนวณ หลังจากนั้นเริ่มกระบวนการสุ่มค่า r ในช่วง 0 ถึง 1 และเริ่มต้นการคำนวณค่าความเร็วถัดไปของอนุภาคนั้นๆ ตามสมการที่ (2.5) แล้วจึงคำนวณค่าตำแหน่งถัดไปของอนุภาคนั้นๆตามสมการที่ (2.6) ตรวจสอบการเข้าถึงขอบเขตที่กำหนดของกลุ่มอนุภาค แล้วจึงเริ่มการคำนวณค่า fitness function ทำการค้นหาค่าที่ดีที่สุดของอนุภาคแต่ละตัวแล้วกำหนดค่าดังกล่าวเป็นค่า P_{Best} หลังจากนั้นหาค่าที่ดีที่สุดจากอนุภาคทั้งหมดแล้วกำหนดให้เป็นค่า G_{Best} ขั้นตอนสุดท้ายคือการตรวจสอบว่าการคำนวณถึงเงื่อนไขการสิ้นสุดของการคำนวณหรือยัง นั่นคือจำนวนรอบมากที่สุดของการวนซ้ำ หากยังไม่ถึงจำนวนรอบมากที่สุดของการวนซ้ำ ให้นำกลับไปทำที่กระบวนการค้นหาต่อไป แต่หากถึงจำนวนรอบมากที่สุดของการวนซ้ำแล้ว จะเป็นการเสร็จสิ้นกระบวนการค้นหาค่าที่ดีที่สุดด้วย PSO

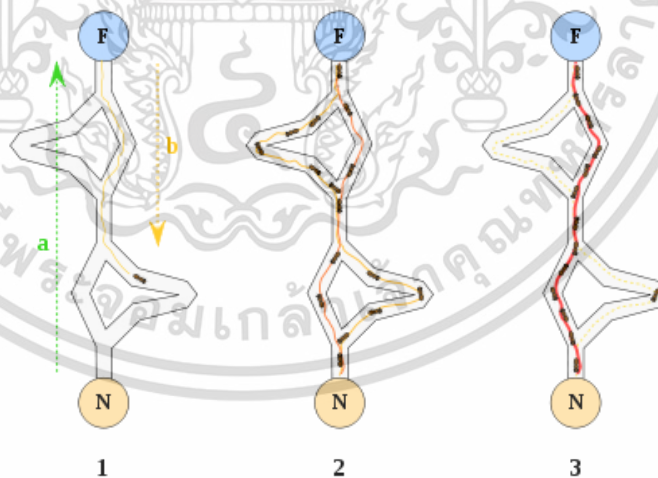


รูปที่ 2.17 ขั้นตอนของ PSO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 Ant Colony Optimization (ACO)

วิธีการค้นหาที่ดีที่สุดด้วยวิธี Ant Colony Optimization (ACO) [9] เป็นกระบวนการ meta-heuristic แบบ Swarm Intelligence เป็นวิธีการค้นหาที่ดีที่สุดโดยใช้เทคนิคของความน่าจะเป็น วิธีการ ACO ถูกคิดค้นขึ้นโดย Marco Dorigo ในปี 1992 รูปแบบการค้นหาของวิธี ACO มาจากแนวคิดของการค้นหาอาหารของมด ในขณะที่มดกำลังค้นหาอาหาร มดแต่ละตัวจะมีการปล่อยฟีโรโมน (pheromone) ทิ้งไว้ระหว่างทางที่เดิน เพื่อนำทางให้มดตัวถัดไป จากการทดลอง double bridge โดยมีอาหารอยู่ที่ระยะห่างหนึ่งกับรังมด มีการสร้างเส้นทางที่แบ่งออกเป็นสองฝั่งและทำการทดลองให้มดเดินจากตำแหน่งของรังมดไปยังตำแหน่งที่มีอาหาร โดยทางที่แบ่งออกเป็นสองฝั่งนี้มีการทำทดลองให้ระยะทางของแต่ละฝั่งไม่เท่ากัน จากการทดลองนี้พบว่า ในช่วงแรกมดจะเลือกเดินทั้งสองเส้นทางก่อนซึ่งเมื่อพบว่าเส้นทางในที่ดีกว่าจะมีอัตราการทิ้งฟีโรโมนที่มากขึ้นตามไปด้วย เมื่อมดตัวถัดไปพบว่าเส้นทางใดมีฟีโรโมนที่เยอะกว่าก็จะเลือกเส้นทางนั้นสำหรับการเดินทางไปยังตำแหน่งที่มีอาหาร ซึ่งเมื่อปริมาณมดเพิ่มมากขึ้นปริมาณฟีโรโมนก็จะมากขึ้นที่เส้นทางนั้นจนยกรวดเร็วเช่นกัน รูปที่ 2.18 แสดงผลการทดลอง double bridge เพื่อศึกษาพฤติกรรมการหาอาหารของมด จะเห็นได้ว่า เมื่อเวลาที่ผ่านไป ดังตำแหน่งในรูป 1, 2, 3 จะพบว่าที่เวลาหนึ่งกลุ่มของมดจะเลือกเดินเพียงเส้นทางเดียวเท่านั้น ซึ่งเป็นเส้นทางที่ดีที่สุดของกลุ่มมด ในที่นี้อาจเป็นเส้นทางที่สั้นที่สุดและมีคุณภาพของอาหารดีที่สุ่มนั่นเอง



รูปที่ 2.18 การทดลอง double bridge เพื่อศึกษาพฤติกรรมการหาอาหารของมด [9]

ในส่วนของการคำนวณของวิธี ACO เริ่มต้นด้วยการกำหนดค่าเริ่มต้นต่างๆ และสร้างคำตอบของเส้นทางทั้งหมดที่มดแต่ละตัวจะหาได้ หลังจากนั้นเริ่มการค้นหาเส้นทางเดินถัดไปของมด พร้อมทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปรับปรุงค่าฟีโรโมนไปตามวิธีการหรือเงื่อนไขที่กำหนดไว้ และขั้นตอนสุดท้ายคือการตรวจสอบการเข้าเงื่อนไขการสิ้นสุดของกระบวนการนั่นเอง

$$\tau_{i,j}^k = (1 - \rho)\tau_{i,j} + \sum \Delta\tau_{i,j}^k \quad (2.7)$$

สมการที่ (2.7) แสดงถึงการคำนวณค่าของฟีโรโมน $\tau_{i,j}^k$ ของมดตัวที่ k เมื่อ ρ คือค่าในช่วง 0 ถึง 1 $\sum \Delta\tau_{i,j}^k$ คือค่าของฟีโรโมนที่เปลี่ยนไปหรือค่าของฟีโรโมนที่ถูกปล่อยออกมาระหว่างทางนั่นเอง ทั้งนี้ค่าของฟีโรโมนที่เปลี่ยนไปนั้น ขึ้นอยู่กับเงื่อนไขดังตารางที่ 2.1 เมื่อมดเดินทางไปตามขอบ i, j ที่ถูกต้องแล้วค่าของฟีโรโมนจะเปลี่ยนไปในเชิงผกผันกับค่าระยะทางที่ใช้ในการเคลื่อนที่ (L_k) เมื่อระยะทางที่มดต้องเดินไปหาอาหารไกล ค่าของฟีโรโมนที่เปลี่ยนไปจะมีค่าน้อย แต่หากระยะทางที่มดต้องเดินไปหาอาหารเป็นระยะทางที่สั้น ค่าของฟีโรโมนที่เปลี่ยนไปจะมีค่าสูง แต่หากมดเดินออกนอกจากเส้นทางตามขอบ i, j แล้วนั้น ค่าของฟีโรโมนที่เปลี่ยนไปจะกลายเป็นค่า 0

ตารางที่ 2.1 แสดงเงื่อนไขของค่าฟีโรโมนที่เปลี่ยนไป

ลำดับที่	เงื่อนไข	ค่าของฟีโรโมนที่เปลี่ยนไป
1	มดเดินอยู่บนเส้นทางตามขอบ i, j	$\Delta\tau_{i,j}^k = \frac{1}{L_k}$
2	อื่นๆ	$\Delta\tau_{i,j}^k = 0$

ขั้นตอนสุดท้ายหลังจากการคำนวณค่าฟีโรโมนคือการคำนวณหาค่าความน่าจะเป็นของเส้นทางของมดแต่ละตัว สมการแสดงการคำนวณแสดงในสมการที่ (2.8) เมื่อ $\tau_{i,j}$ คือค่าของฟีโรโมนในเส้นทางนั้นๆ $\eta_{i,j}$ คือ heuristic information หรือ $\eta_{i,j} = \frac{1}{L_{i,j}}$ α คือค่าสัมประสิทธิ์ของฟีโรโมนและ β คือค่าสัมประสิทธิ์ของ heuristic เส้นทางที่ดีที่สุดซึ่งเป็นผลลัพธ์ของวิธี ACO คือเส้นทางที่ค่าความน่าจะเป็นมีค่าสูงที่สุด

$$P_{i,j} = \frac{(\tau_{i,j})^\alpha (\eta_{i,j})^\beta}{\sum ((\tau_{i,j})^\alpha (\eta_{i,j})^\beta)} \quad (2.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 Artificial Bee Colony (ABC)

วิธีการค้นหาที่ดีที่สุด ด้วยวิธี Artificial Bee Colony (ABC) [32] เป็นกระบวนการ meta-heuristic แบบ Swarm Intelligence โดย Dervis Karaboga ในปี 2005 การคำนวณของวิธีการ ABC นั้น อาศัยพฤติกรรมกรรมการหาอาหารของผึ้งที่อยู่ไนรัง เมื่อผึ้งเจอแหล่งอาหารและส่งสัญญาณการเจอแหล่งอาหารที่ดีที่สุดด้วยลักษณะการเต้นของผึ้งตัวนั้นๆ โดยภายในรังผึ้ง สามารถแบ่งออกได้เป็น 3 กลุ่มดังนี้

1. กลุ่มผึ้งงาน (Employed bee) ทำหน้าที่ในการค้นหาแหล่งอาหารพร้อมทั้งประเมินคุณภาพของอาหารที่พบ และแสดงออกด้วยการเต้นภายในรังผึ้ง
2. กลุ่มผึ้งเฝ้าดู (Onlookers bees) ทำหน้าที่ในการรับข้อมูลที่ได้จากผึ้งงานผ่านลักษณะการเต้นของผึ้งงาน แล้วนำมาใช้ในการเลือกแหล่งอาหาร ซึ่งแหล่งอาหารที่มีคุณภาพและมีปริมาณมาก จะเป็นแหล่งอาหารที่มีความน่าจะเป็นในการถูกเลือกสูง
3. กลุ่มผึ้งสำรวจ (Scout bees) ทำหน้าที่ในการค้นหาแหล่งอาหารใหม่ๆ เมื่อพบว่าปริมาณอาหารที่ผึ้งงานหาได้นั้นไม่มีการเปลี่ยนแปลงไปในทิศทางที่ดีขึ้น

การใช้วิธีการ ABC นี้ ในขั้นตอนที่กลุ่มผึ้งเฝ้าดูทำงานเพื่อเลือกแหล่งอาหารที่มีความน่าจะเป็นสูง นั้น เป็นวิธีการแบบสุ่มวงล้อรูเล็ต ซึ่งอาจจะมีโอกาสที่จะได้แหล่งอาหารที่ความน่าจะเป็นที่จะมีอาหารน้อยเช่นกัน

$$X_{mi} = l_i + rand(0,1)*(u_i - l_i) \quad (2.9)$$

สมการที่ (2.9) แสดงถึงการคำนวณค่าของกลุ่มของแหล่งอาหาร (X_{mi}) เมื่อ l_i แทนค่าขอบล่าง u_i แทนค่าขอบบน และ $rand(0,1)$ แทนค่าสุ่มของตัวเลขในช่วง 0 ถึง 1

$$V_{mi} = X_{mi} + \phi_{mi}(X_{mi} - X_{ki}) \quad (2.10)$$

สมการที่ (2.10) แสดงถึงการคำนวณค่าของกลุ่มของแหล่งอาหารใหม่ (V_{mi}) เมื่อ X_{mi} แทนกลุ่มของแหล่งอาหารเดิม ϕ_{mi} แทนค่าสุ่มของตัวเลขในช่วง $[-1,1]$ และ X_{ki} แทนค่าของแหล่งอาหารแบบสุ่ม

$$fit_m(X_m) = \frac{1}{1 + f_m(X_m)}; \text{ if } f_m(X_m) \geq 0 \quad (2.11)$$

$$fit_m(X_m) = 1 + abs(f_m(X_m)); \text{ if } f_m(X_m) < 0 \quad (2.12)$$

สมการที่ (2.11) และ (2.12) แสดงถึงการคำนวณค่า fitness function ในกรณีที่ฟังก์ชันของแหล่งอาหาร ($f_m(X_m)$) มีค่าเป็นบวก และมีค่าติดลบตามลำดับ

หลังจากการคำนวณแหล่งอาหารจนถึงการคำนวณ fitness function ซึ่งเป็นกระบวนการที่เกิดขึ้นบนกลุ่มฝูงงาน กลุ่มฝูงจะสนใจที่ค่าความน่าจะเป็นในการคัดเลือกเส้นทางที่เหมาะสมที่สุด

$$P_m = \frac{fit_m(X_m)}{\sum_{m=1}^M fit_m(X_m)} \quad (2.13)$$

การคำนวณในสมการที่ (2.13) จะเป็นการคำนวณความน่าจะเป็นจากอัตราส่วนของ fitness function ของแหล่งอาหารนั้นๆ กับผลรวมของ fitness function ซึ่งผลลัพธ์สุดท้ายที่ถูกเลือกจะมีความน่าจะเป็นที่จะถูกเลือกมากที่สุด บนผลลัพธ์ที่มีค่า fitness มากที่สุด หลังจากกระบวนการวนซ้ำทั้งหมดนี้ เมื่อกลุ่มฝูงงานไม่สามารถหาแหล่งอาหารที่ดีขึ้นได้ กลุ่มฝูงงานเหล่านั้นจะถูกเปลี่ยนเป็นกลุ่มฝูงสำรวจเพื่อสุ่มแหล่งอาหารต่อไปจนกว่าจะถึงเงื่อนไขการสิ้นสุดการทำงานของโปรแกรม

2.4 งานวิจัยที่เกี่ยวข้อง (Literature review)

จากการศึกษางานวิจัยที่เกี่ยวข้อง พบว่า มีงานวิจัยที่เกี่ยวข้องกับหุ่นยนต์หลากหลายงานวิจัย และมีการปรับปรุงวิธีการให้ระบบการทำงานของหุ่นยนต์ดีขึ้นในหลากหลายวิธี Daniel D. และคณะ [10] ทำการศึกษาการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ โดยพวกเขาเริ่มจากการศึกษาวิธีการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์จากวิธี Dijkstra's algorithm จนมีการพัฒนาปรับปรุงเป็น A* และปรับปรุง A* เป็นวิธี D* จนนำมาสู่วิธีการที่พวกเขานำเสนอ โดยเป้าหมายคือวางแผนเส้นทางที่สั้นที่สุด ซึ่งได้ผลดีทั้งใน simulation และ experiment บางงานวิจัยมีการนำวิธีการ RRT มาปรับปรุงการวางแผนเส้นทางการเคลื่อนที่สำหรับหุ่นยนต์ในโรงงานอุตสาหกรรมที่แผนที่มีความซับซ้อน การนำวิธีการ RRT มาใช้ยังคงให้ผลที่ไม่ดีนัก และบางครั้งยังได้ผลลัพธ์เป็น local optimal ซึ่งไม่ใช่ global optimal ในงานวิจัย [11] ได้นำเสนอวิธีการปรับปรุงวิธี RRT เพื่อลดการทำงานที่ซ้ำซ้อนและไม่เข้าสู่เป้าหมายที่สนใจ โดยทดลองภายใต้สภาพแวดล้อมที่ซับซ้อน แต่ได้ผลลัพธ์เป็นที่น่าพอใจ ซึ่งใน [12] มีการออกแบบเส้นทางการเคลื่อนที่ของหุ่นยนต์ภายใต้สภาพแวดล้อมที่ซับซ้อนเช่นกัน โดยทดลองในสภาพแวดล้อมที่เป็น dynamic obstacle และเลือกใช้ Any Colony Algorithm และ Clustering Algorithm เพื่อสร้างการหลบสิ่งกีดขวางแบบอัตโนมัติ ซึ่งผล simulation แสดงให้เห็นถึงเส้นทางที่ปลอดภัยในการหลบสิ่งกีดขวาง Xu Tong และคณะ นำเสนอการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ด้วย Support Vector Machine (SVM) และได้ผลลัพธ์ในแบบจำลองที่ดี [13] ในงานวิจัย [14] มีการนำ Neural และ RRT มาประยุกต์ใช้ในการหาวางแผนเส้นทางการเคลื่อนที่ที่ดีที่สุด โดยเลือกใช้ A* ในการสร้าง training data set ซึ่งประกอบไปด้วยแผนที่และเส้นทางที่ดี แล้วจึงนำไปทดสอบด้วย Neural-RRT ซึ่งได้ผลลัพธ์ที่ดี Min Luo และคณะ เลือกใช้การปรับปรุงวิธี Dijkstra's algorithm เพื่อการวางแผนเส้นทางที่ดีที่สุด และเปรียบเทียบผลลัพธ์ของแบบจำลองด้วยวิธีการดั้งเดิมและวิธีการที่ปรับปรุงแล้ว หากมีการเพิ่มความซับซ้อนของสภาพแวดล้อม จะทำให้ผลการทดลองนี้น่าสนใจมากยิ่งขึ้น [15] Raouf F. และคณะ นำเสนอการเน้นการสนใจในพื้นที่วิกฤตภายในแผนที่ เพื่อลดเวลาในกระบวนการ โดยมีการนำวิธีการพื้นฐานเช่น A* ,D* และ Probabilistic Roadmap (PRM) มาใช้เพื่อหาเส้นทางที่เหมาะสม โดยได้ทดลองบนหุ่นยนต์จริงด้วยและพบว่าเวลาในกระบวนการมีการปรับปรุงที่ดีมาก [16] ในงานวิจัยเกี่ยวกับหุ่นยนต์ Genetic algorithm (GA) จัดเป็นหนึ่งใน algorithm ที่มีการนำไปประยุกต์ใช้ Mao L. และคณะ ได้เลือกใช้ GA สำหรับการหลบสิ่งกีดขวางของหุ่นยนต์ โดยนำมาประยุกต์ใช้ด้วยการใช้เงื่อนไขการหลบสิ่งกีดขวางเป็นเงื่อนไขหลัก ผลในแบบจำลองแสดงให้เห็นถึงความสามารถในการหลบสิ่งกีดขวางได้เป็นอย่างดี [17] Giohong และคณะ ทำการศึกษาเกี่ยวกับ การปรับปรุงเวลาการวางแผนเส้นทางให้หุ่นยนต์ในโรงงานอุตสาหกรรมโดยการเลือกใช้ GA โดยมีการเลือกใช้ cubic polynomial เพื่อเชื่อมโยงระหว่างจุดและทำให้เส้นทางมีความราบรื่น โดยฟังก์ชันวัตถุประสงค์ที่ถูกเลือกใช้คือฟังก์ชันของการหา

ระยะทางที่สั้นที่สุด [18] ในงานวิจัย [19] มีการเลือกใช้ GA สำหรับแก้ไขปัญหาของการเปลี่ยนงานให้หุ่นยนต์กรณีที่มีหลายหุ่นยนต์ เพื่อให้หุ่นยนต์ได้รับงานที่เฉลี่ยเท่าๆกันไป Yanhui และคณะ เลือกใช้ GA ในการวางแผนเส้นทางการเคลื่อนที่ให้หุ่นยนต์ โดยเลือกใช้ GA เพื่อแก้ปัญหาเวลาประมวลผลที่นานและการค้นหาที่เจอเพียง local optimum ซึ่งยังคงเลือกฟังก์ชันวัตถุประสงค์เป็นฟังก์ชันของระยะทางที่สั้นที่สุด [20] นอกเหนือจากวิธี GA วิธี Particle Swarm Optimization (PSO) ก็เป็นหนึ่งในวิธีที่เป็นที่นิยมมากเช่นกัน Tong Y. และคณะ นำวิธี GA และ PSO มาเปรียบเทียบกัน สำหรับการหาเส้นทางการเคลื่อนที่ให้หุ่นยนต์ โดยเลือกใช้ GA สำหรับการทำให้กระบวนการประมวลผลทำได้อย่างรวดเร็ว แต่ในส่วนของ PSO ก็สามารถทำได้เช่นกันด้วยการ ปรับปรุงค่า inertia weight ซึ่งผลลัพธ์จากแบบจำลองของพวกเขาแสดงให้เห็นถึงประสิทธิภาพของทั้งสองวิธีการที่ตีตื้นคู่ [21] ในงานวิจัย [22] พวกเขาเลือกใช้ PSO สำหรับการปรับการหาเส้นทางที่ดีที่สุด ทดลองภายใต้สภาพแวดล้อมที่แตกต่างกัน และได้ผลลัพธ์ที่ดี Xiangjun K. และคณะ เลือกใช้ PSO และ Greedy algorithm สำหรับระบบจัดการงานให้หุ่นยนต์ที่มีมากกว่า 1 หุ่นยนต์ โดยการใช้ PSO สำหรับค้นหางานกับหุ่นยนต์นั้นๆ และใช้ Greedy algorithm สำหรับการจัดเรียงลำดับของงาน ซึ่งได้ผลลัพธ์ในแบบจำลองที่ดี [23] Syed Irfan Ali Meerza และคณะ ได้นำ PSO และ Q-value มาผสมผสานกันสำหรับการหาเส้นทางที่ดีที่สุดให้หุ่นยนต์ ภายใต้สิ่งแวดล้อมที่ไม่ทราบลักษณะของสิ่งแวดล้อมนั้น ผลลัพธ์ที่ได้แสดงให้เห็นว่ากระบวนการมีการทำงานที่รวดเร็วขึ้นและยังได้ผลลัพธ์ที่ถูกต้อง [24] Changyun W. เลือกใช้ PSO สำหรับการจัดการงานให้หุ่นยนต์ที่มีมากกว่า 1 หุ่นยนต์ โดยวัตถุประสงค์คือเพื่อให้มีกระบวนการที่ให้งานของแต่ละหุ่นยนต์ที่เท่าๆกัน โดยยังคงเป็นเส้นทางที่น้อยที่สุดของหุ่นยนต์ทั้งหมดอยู่ พวกเขานำเสนอการใช้ 1 PSO กับหลายฟังก์ชันวัตถุประสงค์ [25] ในงานวิจัย [26] มีการออกแบบ Improved PSO เพื่อใช้กับหุ่นยนต์ในโรงงานอุตสาหกรรม โดยเป้าหมายของงานวิจัยของพวกเขาคือ ถ้าสามารถหาเวลาที่เหมาะสมในการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ได้นั้น จะช่วยประหยัดพลังงานโดยรวมและยังคงได้เส้นทางที่ดีที่สุดด้วยเช่นกัน พวกเขาปรับปรุงค่าของ ความเร็วและความเร่งของ PSO โดยที่ไม่มีผลกระทบต่อความสามารถในการค้นหาของ PSO เดิม ซึ่งได้ผลลัพธ์ที่ดีบนแบบจำลอง นอกเหนือจากวิธี GA และ PSO วิธีการ Ant Colony Optimization (ACO) ก็เป็นหนึ่งในวิธีที่มีการนำมาประยุกต์ใช้ในหลากหลายงานวิจัย Huangfu S. และคณะ เลือกใช้การนำ ACO มาปรับปรุงเพื่อการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ [27] Xue Y. และคณะ นำวิธี ACO มาผสมผสานกับ A* โดยหน้าที่ของ A* คือการหาเส้นทางและให้ ACO ทำการค้นหาลำดับของการเคลื่อนที่ไปยังเป้าหมายเพื่อให้ได้เวลาที่รวดเร็วที่สุด โดยพวกเขาได้ผลการทดลองที่เห็นถึงประสิทธิภาพที่ดีขึ้นบน ACO-A* [28] Xu L. และ Zhengyan L. นำเสนอการปรับปรุงวิธี ACO สำหรับการจัดการงานให้หุ่นยนต์ที่มีมากกว่า 1 หุ่นยนต์ พวกเขาปรับปรุงค่าของฟีโรโมนให้เป็นไปตาม local และ global พร้อมกับมีการ cross path เพื่อปรับปรุงค่าของผลลัพธ์สุดท้าย ผลจากแบบจำลองให้

ผลลัพธ์ที่ถูกต้องและรวดเร็ว [29] ในงานวิจัย [30] พวกเขาเสนอ ACO สำหรับการหาเส้นทาง การเคลื่อนที่ให้หุ่นยนต์ที่มีมากกว่า 1 หุ่นยนต์ในโรงงานอุตสาหกรรม โดย ACO เข้ามา มีบทบาทในการค้นหา จากการเปรียบเทียบเส้นทางที่ดีที่สุดก่อนหน้ากับเส้นทางปัจจุบัน พร้อมทั้งแก้ปัญหาการเข้าถึง local optimum อย่างรวดเร็วโดยการเพิ่มกระบวนการ mutation อีกด้วย Farouq Z. และคณะนำเสนอการใช้ Consensus-Based Bundle Algorithm กับ Ant Colony System สำหรับการแก้ปัญหาการจัดวางงาน ให้หุ่นยนต์ที่มีมากกว่า 1 หุ่นยนต์ โดยการใช้ Ant Colony สำหรับการแก้ไขการจัดการงาน ซึ่งผลลัพธ์ใน แบบจำลองให้ผลที่ดี [31] อีกหนึ่งในวิธีการค้นหาที่ดีที่สุดที่เป็นที่นิยมคือวิธี Artificial Bee Colony (ABC) Xiangmin L. และคณะได้ปรับปรุงวิธี ABC เพื่อการค้นหาเส้นทางที่ดีที่สุดของหุ่นยนต์ โดยมีการ นำ ABC มาค้นหาเส้นทางที่สั้นที่สุด [32] Necmettin A. และคณะ นำเสนอการผสมผสานของวิธี ABC และวิธี Probabilistic roadmap (PRM) เพื่อใช้ในการค้นหาเส้นทางที่สั้นที่สุด โดยใช้ heuristic optimization method เพื่อการปรับค่าให้ได้เส้นทางที่เหมาะสมที่สุด [33]

จากงานวิจัยที่เกี่ยวข้องกับการวางแผนเส้นทางเคลื่อนที่ของหุ่นยนต์ พบว่า มีการนำวิธีการ วางแผนเส้นทางเคลื่อนที่ของหุ่นยนต์และวิธีการค้นหาที่ดีที่สุดมาประยุกต์ใช้ในหลากหลายงานวิจัย โดยแต่ละวิธีจะมีข้อดี ข้อด้อยแตกต่างกัน โดยงานวิจัยส่วนใหญ่มุ่งเน้นไปที่การค้นหาเส้นทางที่ดีที่สุดของ หุ่นยนต์จากการค้นหาเส้นทางที่สั้นที่สุด เพื่อให้ได้ระยะเวลาในการทำงานที่รวดเร็วพร้อมกับการประหยัด พลังงานโดยรวมของระบบ และในงานวิจัยที่ผ่านมาที่มีการวางแผนเส้นทางเคลื่อนที่ของหุ่นยนต์ที่ดี ที่สุดนั้น เป็นการวางแผนเส้นทางภายใต้ 1 หุ่นยนต์เท่านั้น

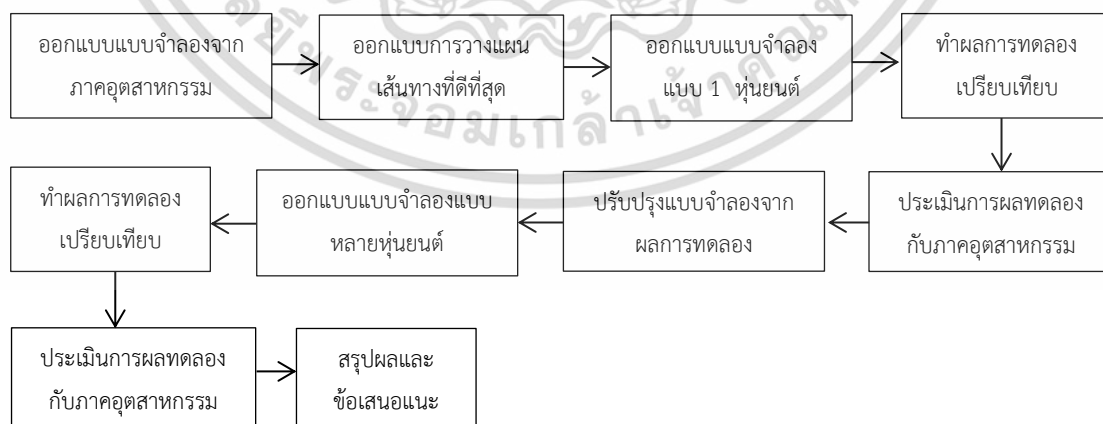
งานวิจัยนี้แนะนำเสนอการวางแผนเส้นทางเคลื่อนที่ให้กับหุ่นยนต์แบบใหม่ด้วยมุ่งเน้นไปที่การ ค้นหาเส้นทางที่ปลอดภัยที่สุดให้กับหุ่นยนต์เพื่อใช้ในโรงงานอุตสาหกรรม พร้อมทั้งวางแผนเส้นทางที่ดีที่สุด ภายใต้อุปกรณ์แบบ multi-robot systems ซึ่งในงานวิจัยที่ผ่านมาไม่พบว่ามีงานวิจัยเรื่องการ วางแผนเส้นทางที่ดีที่สุดให้กับหุ่นยนต์ด้วยการวางแผนเส้นทางที่ปลอดภัยที่สุดและไม่พบงานวิจัยที่ศึกษา สภาพแวดล้อมในแบบจำลองจากภาคอุตสาหกรรมโดยตรงสำหรับการวางแผนเส้นทางที่ดีที่สุด รวมทั้งไม่ พบงานวิจัยที่ศึกษาการวางแผนเส้นทางที่ดีที่สุดโดยการออกแบบภายใต้ multi-robot systems

วิธีการที่งานวิจัยนี้แนะนำเสนอคือการนำ A* มา modified เพื่อการวางแผนเส้นทางที่ดีที่สุดด้วยการ ค้นหาเส้นทางที่ปลอดภัยที่สุดจากการสร้างฟังก์ชัน OA เพื่อแสดงค่าความเสี่ยงในการชนสิ่งกีดขวางที่อยู่ หนึ่งและงานวิจัยนี้ยังสร้างฟังก์ชัน RC เพื่อแสดงค่าความเสี่ยงในการชนสิ่งกีดขวางที่เคลื่อนที่ได้โดยเลือกใช้ ABC สำหรับการค้นหาเส้นทางที่ RC มีความเสี่ยงในการชนสูงที่สุดแล้วจึงนำออกจากการพิจารณา เส้นทางให้กับหุ่นยนต์ พร้อมทั้งออกแบบการจัดการงานให้กับหุ่นยนต์ในระบบที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์ภายใต้แบบจำลองต้นแบบจากภาคอุตสาหกรรม

บทที่ 3

การออกแบบการทดลอง

จากการศึกษางานวิจัยที่เป็นที่ยอมรับในการประชุมนานาชาติหรือวารสารนานาชาติ พบว่ามีการพัฒนาการวางแผนเส้นทางที่ดีที่สุดให้หุ่นยนต์ในสภาพแวดล้อมที่แตกต่างกันหลากหลายวิธี [10-22] ซึ่งหนึ่งในวิธีการที่เห็นได้ชัดเจนจากผลการวิจัยหลากหลายคือ วิธีการวางแผนเส้นทางเคลื่อนที่ด้วย A* [5] ซึ่งถูกพัฒนามาจากวิธีการ Dijkstra [4] และได้ผลลัพธ์ที่ดีขึ้นอย่างเห็นได้ชัดเจน งานวิจัยนี้จึงเลือก A* สำหรับการวางแผนเส้นทางเคลื่อนที่ อีกหนึ่งส่วนคือการวางแผนเส้นทางเคลื่อนที่ที่เหมาะสมที่สุด จากการศึกษางานวิจัยพบว่า งานวิจัยส่วนใหญ่ เลือกใช้การวางแผนเส้นทางที่เหมาะสมที่สุดจากการหาเส้นทางที่สั้นที่สุดเท่านั้น ซึ่งการหาเส้นทางที่สั้นที่สุด มีข้อดีในเรื่องของการประหยัดพลังงานโดยรวมของระบบ และยังเป็น การเพิ่มประสิทธิภาพการทำงานโดยรวมของระบบให้สามารถทำงานได้มากขึ้นต่อวันด้วย แต่มีสิ่งหนึ่งที่ขาดไปจากการศึกษางานวิจัยที่ผ่านมา งานวิจัยนี้จึงขอเสนอการวางแผนเส้นทางที่เหมาะสมที่สุดด้วยการหาเส้นทางที่ปลอดภัยที่สุดก่อนแล้วจึงค้นหาเส้นทางที่สั้นที่สุดในลำดับต่อมา งานวิจัยนี้ จึงเลือกวิธีการค้นหาที่ดีที่สุดโดยสามารถค้นหา Global optimum ได้ ซึ่งจากการศึกษางานวิจัย พบว่าวิธีที่ได้รับความนิยมเป็นอย่างมากคือ GA, PSO, ACO และ ABC งานวิจัยนี้จึงทำการศึกษาในวิธีการเหล่านี้และนำมาทดลองในแบบจำลอง งานวิจัยนี้ เริ่มต้นด้วยการสร้างแบบจำลองก่อน โดยการศึกษาค้นคว้าแผนการออกแบบของโรงงานอุตสาหกรรมของบริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด เป็นต้นแบบในแบบจำลองของงานวิจัยนี้



รูปที่ 3.1 แผนการทำงานวิจัย

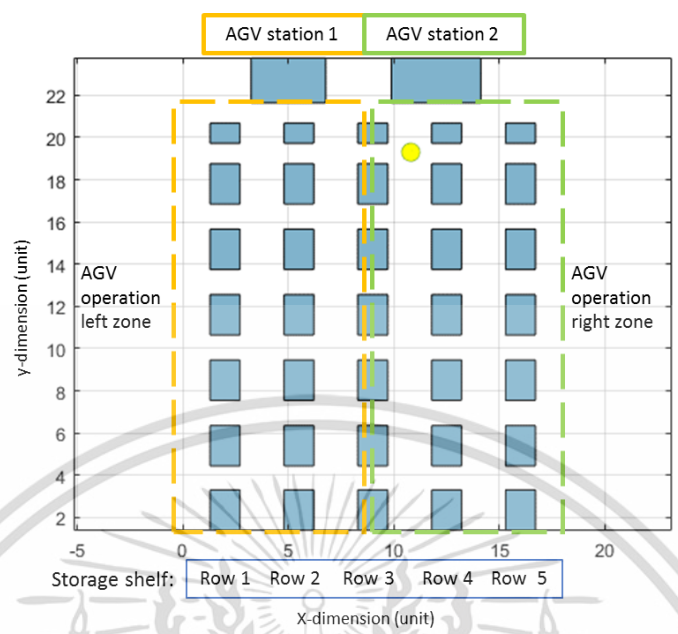
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 แสดงแผนการทำงานวิจัยนี้ โดยเริ่มต้นจากการศึกษาโจทย์ปัญหาที่ได้รับจากโรงงานอุตสาหกรรมก่อน ซึ่งมีโจทย์อยู่ว่า ต้องการแบบจำลองที่สามารถค้นหาเส้นทางที่ดีที่สุดในระบบที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์ได้ จึงเริ่มดำเนินการค้นหาข้อมูลของเงื่อนไขภายในโรงงานอุตสาหกรรมและข้อจำกัดต่างๆในโรงงานอุตสาหกรรม และพบว่า ความปลอดภัยเป็นหนึ่งในเงื่อนไขสำคัญภายในโรงงานอุตสาหกรรมเพราะเส้นทางที่ปลอดภัยจะทำให้การทำงานจริงภายในอุตสาหกรรมนั้นไม่เกิดปัญหาร้ายแรงที่คาดไม่ถึงที่อาจจะส่งผลเสียต่อการดำเนินงานอื่นๆภายในภาคอุตสาหกรรม จึงเริ่มดำเนินการศึกษาวรรณกรรมที่ได้รับการตีพิมพ์ เกี่ยวกับการค้นหาเส้นทางของหุ่นยนต์ในแบบต่างๆ และเริ่มการออกแบบเส้นทางเคลื่อนที่ที่ดีที่สุดในระบบที่มีเพียง 1 หุ่นยนต์ก่อน โดยที่การออกแบบเส้นทางที่ดีที่สุดในงานวิจัยนี้ มีความแตกต่างออกไปจากงานวิจัยอื่น คือเพิ่มการค้นหาเส้นทางที่มีความปลอดภัยที่สุด โดยที่เมื่อหุ่นยนต์เจอเส้นทางที่ปลอดภัยที่สุดแล้วจากการค้นพบค่าความเสี่ยงที่น้อยที่สุด จึงค้นหาเส้นทางที่สั้นที่สุดในลำดับต่อมา หลังจากออกแบบแบบจำลองภายใต้ 1 หุ่นยนต์ ในสภาพแวดล้อมอ้างอิงจากโรงงานอุตสาหกรรมดังรูปที่ 3.2 แล้วนั้น จึงได้ทำการเปรียบเทียบกับวิธีการแบบ conventional คือการค้นหาเส้นทางที่สั้นที่สุด เพื่อเปรียบเทียบแบบจำลองของวิธีการดั้งเดิม และวิธีการใหม่ในงานวิจัยนี้ นำเสนอ หลังจากนั้นจึงทำการนำเสนอเพื่อปรับปรุงระบบ โดยเมื่อปรับปรุงระบบแล้ว จึงเริ่มดำเนินการสร้างแบบจำลองเพิ่ม สำหรับระบบที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์ และยังเพิ่มค่าความผิดพลาดเข้าไปในระบบเพื่อทำการทดลองในแบบจำลองในกรณีที่บางอย่างในระบบมีสิ่งรบกวน ซึ่งอาจทำให้ผลบางอย่างผิดเพี้ยนไปจากผลในอุดมคติได้ แล้วจึงทำการเปรียบเทียบผลของวิธีการแบบดั้งเดิมและวิธีการใหม่ในงานวิจัยนี้ นำเสนอ เพื่อนำเสนอให้แก่ผู้ให้ทุนวิจัย และนำมาปรับปรุงระบบ จนถึงขั้นตอนของการสรุปผลงานวิจัย

จากรูปที่ 3.2 เป็นการสร้างแบบจำลองที่อ้างอิงจากการศึกษาค้นคว้าแผนการจัดวางสถานีต่างๆ ในโรงงานอุตสาหกรรมของบริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด โดยแต่ละสถานีแทนฟังก์ชันการทำงานในสถานีนั่นๆ เพื่อให้ยานยนต์เคลื่อนที่จากตำแหน่งเริ่มต้นไปยังตำแหน่งเป้าหมาย นอกจากนี้ งานวิจัยนี้ยังระบุระยะห่างเป็นตัวแปรเพื่อรองรับการปรับเปลี่ยนแผนที่ในอนาคตสำหรับโรงงาน

อุตสาหกรรมดังรูปที่ 3.3 โดย $a = \frac{6}{5}r$ และ $b = \frac{9}{5}r$ เมื่อ r คือค่ารัศมีวงกลมของพื้นที่ยานยนต์ดังรูปที่

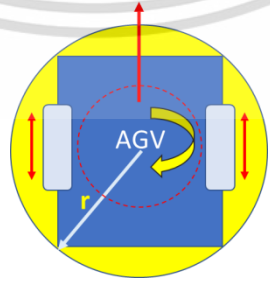
3.4



รูปที่ 3.2 แบบจำลองแผนที่ที่ออกแบบโดยอ้างอิงจากแบบเบื้องต้นของโรงงานอุตสาหกรรม



รูปที่ 3.3 ระยะจากจุดขอบของแต่ละสถานีกับตำแหน่งของยานยนต์

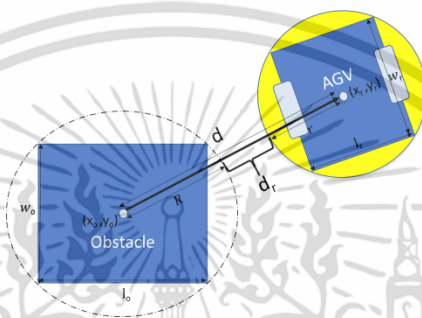


รูปที่ 3.4 ยานยนต์อัจฉริยะและอาณาเขตวงกลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการหลบสิ่งกีดขวาง งานวิจัยนี้เลือกใช้อาณาเขตวงกลมสำหรับยานยนต์อัจฉริยะ และสิ่งกีดขวาง เพื่อใช้แทนการคำนวณระยะที่ปลอดภัย จากรูปที่ 3.5 เมื่อให้ (x_r, y_r) แทนตำแหน่งของยานยนต์ และให้ (x_o, y_o) แทนตำแหน่งของสิ่งกีดขวางแล้วนั้น จะสามารถคำนวณระยะห่างของยานยนต์ถึงสิ่งกีดขวางได้ตาม Euclidean distance ดังแสดงในสมการที่ (3.1)

$$d = \sqrt{(x_r - x_o)^2 + (y_r - y_o)^2} \quad (3.1)$$



รูปที่ 3.5 อาณาเขตวงกลมของสิ่งกีดขวางและยานยนต์

จากรูปที่ 3.5 แสดงถึงระยะที่ปลอดภัยระหว่างสิ่งกีดขวางและยานยนต์ โดยแทนในรูปของตัวแปร

d_r สมการการคำนวณค่า d_r แสดงในสมการที่ (3.2) เมื่อ $R = \sqrt{\left[\frac{w_o}{2}\right]^2 + \left[\frac{l_o}{2}\right]^2}$,

$r = \sqrt{\left[\frac{w_r}{2}\right]^2 + \left[\frac{l_r}{2}\right]^2}$ w_o คือความกว้างของสิ่งกีดขวาง l_o คือความยาวของสิ่งกีดขวาง w_r คือความกว้างของยานยนต์ และ l_r คือความยาวของยานยนต์

$$d_r = d - R - r \quad (3.2)$$

งานวิจัยนี้นำเสนอสมการใหม่ที่ชื่อว่า Obstacle avoidance function (OA) แสดงในสมการที่ (3.3) โดยค่า α แทนจำนวนเต็มบวกที่อยู่ในช่วง (0,1)

$$OA = \max\left(1 - \alpha \cdot \frac{d}{R + r}, 0\right) \quad (3.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงเงื่อนไขของค่า OA

ลำดับที่	เงื่อนไข	ค่าของ OA	การประเมินความเสี่ยง
1	$d < R+r$	$\Rightarrow 1$	มีความเสี่ยงต่อการชนมาก
2	$d > R+r$	$\Rightarrow 0$	มีความเสี่ยงต่อการชนน้อย
3	$d \gg R+r$	0	อยู่ในระยะปลอดภัย

จากตารางที่ 3.1 แสดงถึงเงื่อนไขของค่า OA ในกรณีต่างๆ จะพบว่า เมื่อค่า $d < R+r$ หนูนยนต์กับสิ่งกีดขวางจะมีความเสี่ยงในการชนกัน ซึ่งค่าความเสี่ยงต่อการชนที่มากหรือน้อยนั้น สามารถวัดได้จากค่าของ OA เมื่อค่า OA เข้าใกล้ 1 จะมีความเสี่ยงต่อการชนสูงมาก แต่ถ้าค่า OA เข้าใกล้ค่า 0 จะมีความเสี่ยงต่อการชนที่น้อยกว่า ซึ่งในอุดมคติ เราต้องการระบบที่ปลอดภัย หรือต้องการค่า OA ที่เป็น 0 ตลอดเวลา แต่ด้วยสภาพแวดล้อมจริง ในเวลาที่มีการเคลื่อนที่ของยานยนต์ในบางพื้นที่ อาจมีเหตุการณ์ที่ทำให้ค่า OA นั้นมากกว่า 0 ซึ่งในกรณีนี้ เราต้องการให้ค่า OA เข้าใกล้ 0 ที่สุด เพื่อให้เส้นทางการเคลื่อนที่นั้น เป็นเส้นทางที่ปลอดภัยที่สุด

อีกหนึ่งในส่วนสำคัญของการวางแผนเส้นทางการเคลื่อนที่ คือการสร้างเส้นทางเชื่อมต่อระหว่างจุดให้เส้นทางมีความราบรื่น ซึ่งในปัจจุบันมี 2 แบบหลักๆ ดังนี้

1. Linear interpolation

วิธีการแบบ Linear interpolation คือวิธีอย่างง่ายในการเชื่อมระหว่างจุดให้เป็นเส้นตรงดังแสดงในสมการที่ (3.4) เมื่อ A แทนค่าของความชัน, B แทนค่าของจุดตัดแกน Y

$$y = f(x) = Ax + B \quad (3.4)$$

เมื่อเราทราบช่วงของค่า x อยู่ในช่วง $[a, c]$ เราสามารถที่จะสร้างเส้นตรงโดยประมาณได้ โดยการแทนค่า a, c ลงไปในสมการที่ (3.4) จะได้สมการที่ (3.5) และ (3.6) ตามลำดับ

$$f(a) = Aa + B \quad (3.5)$$

$$f(c) = Ac + B \quad (3.6)$$

หลังจากนั้น แก้มการคำนวณหาค่า A และ B ได้ตั้งสมการที่ (3.7) และ (3.8) แล้วจึงแทนที่ค่า A และ B ในสมการที่ (3.4) อีกครั้ง จะได้สมการเส้นตรงที่สร้างจาก Linear interpolation ตั้งสมการที่ (3.9) เมื่อ $a \leq b \leq c$ ซึ่งจะเห็นได้ว่า วิธีการนี้ ต้องมีจุดอย่างน้อย 2 จุดจึงสามารถหาค่าสัมประสิทธิ์ เพื่อใช้ในการสร้าง Linear interpolation ได้

$$A = \frac{f(c) - f(a)}{c - a} \quad (3.7)$$

$$B = f(c) - \frac{f(c) - f(a)}{c - a} c \quad (3.8)$$

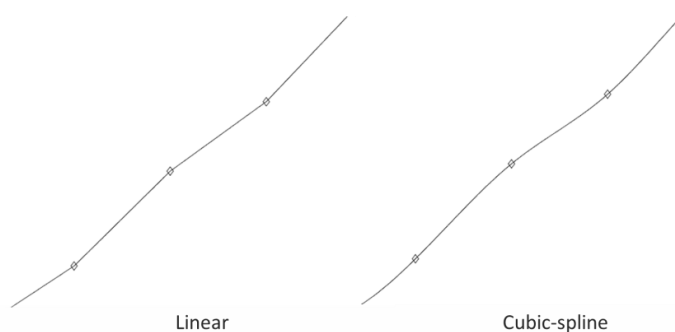
$$f(b) = \frac{f(c) - f(a)}{c - a} b + \left[f(c) - \frac{f(c) - f(a)}{c - a} c \right] \quad (3.9)$$

2. Cubic-spline interpolation

วิธี Cubic-spline interpolation เป็นการสร้างเส้นจากสมการพหุนามดีกรี วิธีการนี้จะทำให้ได้เส้นเชื่อมโยงจุดที่ราบรื่นกว่าวิธีการ Linear interpolation ยกตัวอย่างเช่นสมการพหุนามดีกรี 3 ในสมการที่ (3.10) ซึ่งจะเห็นได้ว่ามีค่าสัมประสิทธิ์อยู่ถึง 4 ค่า นั่นคือ $(\alpha_3, \alpha_2, \alpha_1, \beta_1)$ ซึ่งแปลว่า จะต้องมีย่าน้อย 4 จุด จึงจะสามารถหาค่าสัมประสิทธิ์ได้ครบทั้งหมดเพื่อใช้ในการสร้างเส้น Cubic-spline interpolation ได้

$$y = f(x) = \alpha_3 x^3 + \alpha_2 x^2 + \alpha_1 x + \beta_1 \quad (3.10)$$

จากรูปที่ 3.6 แสดงให้เห็นถึงการเปรียบเทียบยาระหว่างการสร้างเส้นเชื่อมโยงของวิธี Linear กับวิธี Cubic-spline ซึ่งจะเห็นได้ว่าพจน์ของพหุนามดีกรี ส่งผลให้ผลลัพธ์ของเส้นเชื่อมแบบ Cubic-spline นั้นมีความราบรื่นมากกว่าแบบวิธี Linear



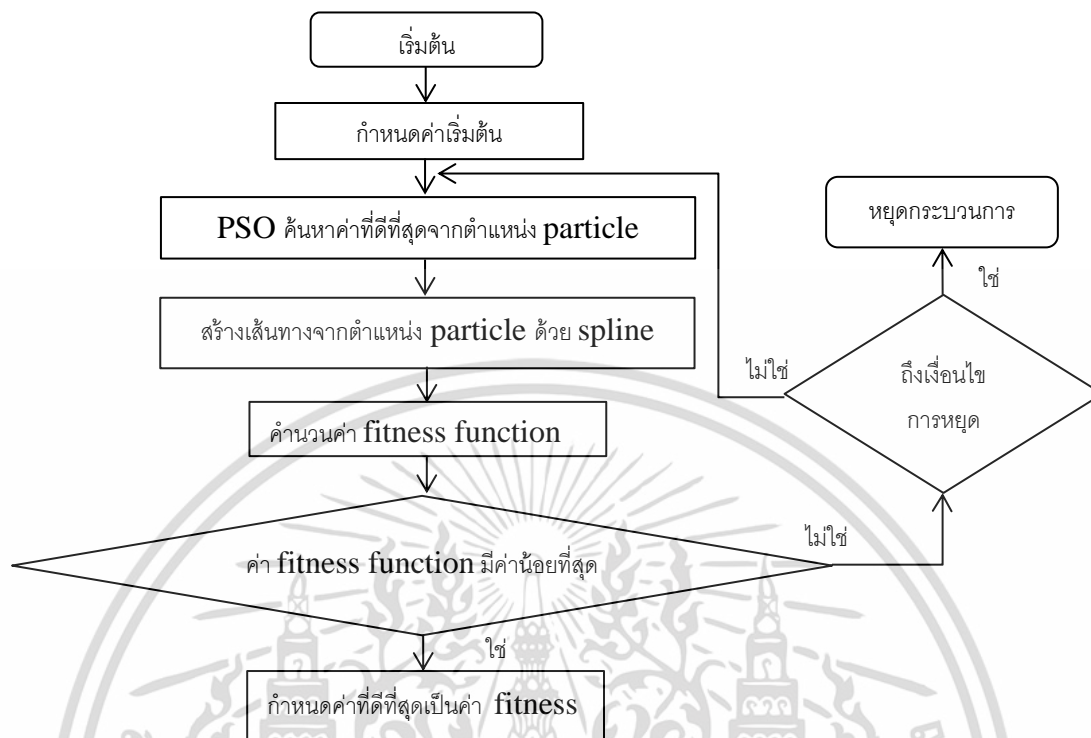
รูปที่ 3.6 เปรียบเทียบการสร้างเส้นเชื่อมโยงของวิธี Linear กับ Cubic-spline

งานวิจัยนี้เริ่มต้นที่การทดลองภายใต้แบบจำลองแบบ 1 หน่วยงานก่อน โดยการเลือกใช้วิธี PSO มาทำการทดลองในวิธีการที่ต้องการนำเสนอในงานวิจัยนี้ เริ่มต้นด้วยการเลือก fitness function สำหรับ PSO โดยการออกแบบฟังก์ชัน Z ดังแสดงในสมการที่ (3.11) เพื่อใช้ในการค้นหาเส้นทางที่ปลอดภัยที่สุด และยังคงได้ระยะทางที่สั้นด้วยเช่นกัน เมื่อ γ คือค่าสัมประสิทธิ์ของระยะทาง ,

$$L = \sum_{i=1}^{n-1} \left[\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \right], \text{ และ } \beta \text{ คือค่าสัมประสิทธิ์ของค่า OA}$$

$$Z = \gamma * L + \beta * OA \quad (3.11)$$

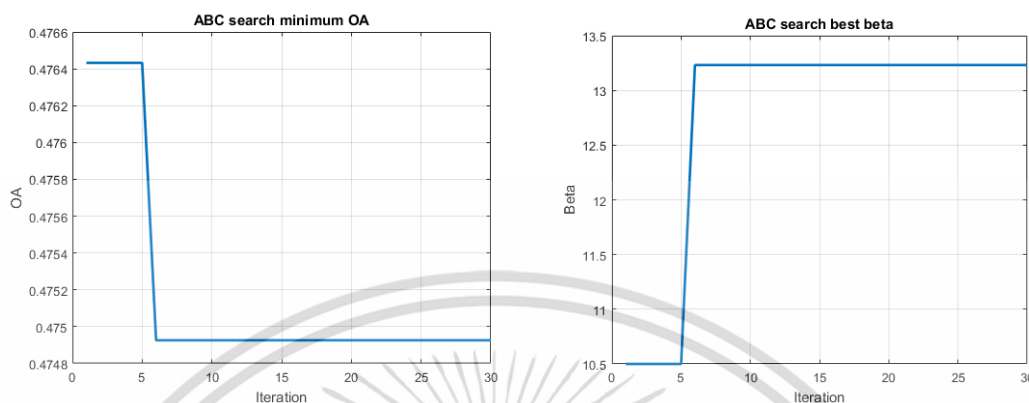
จากรูปที่ 3.7 แสดงขั้นตอนของวิธีการที่ต้องการนำเสนอของงานวิจัยนี้ โดยเริ่มต้นจากการตั้งค่าเริ่มต้นของระบบ แล้วให้ PSO ทำการค้นหาตำแหน่งของอนุภาค หลังจากนั้น ใช้ Cubic-spline ในการสร้างเส้นจากจุดเริ่มต้นที่จุดสิ้นสุดระหว่างตำแหน่งของอนุภาค นำค่าตำแหน่งของอนุภาคมาทำการคำนวณค่า Z ในสมการที่ (3.11) และตรวจสอบค่าของ OA ว่าเป็นค่าน้อยที่สุดหรือยัง ถ้าใช่ปรับค่าน้อยที่สุดเป็นค่านั้นเพื่อใช้ในการคำนวณรอบถัดไป และตรวจสอบค่าของ Z (fitness function) ว่าเป็นค่าน้อยที่สุดหรือยัง ถ้าใช่ปรับค่า fitness ที่ดีที่สุดเป็นค่านั้น ถ้ายังไม่ใช่ ให้ทำการค้นหาค่าต่อไปจนกว่าจะสิ้นสุดกระบวนการ



รูปที่ 3.7 ขั้นตอนของวิธีการที่ต้องการนำเสนอของงานวิจัย

การออกแบบงานวิจัยในส่วนแรกเป็นการออกแบบในระบบ 1 หุ่นยนต์ ซึ่งในสภาพแวดล้อมจริงภายในอุตสาหกรรมเป็นสภาพแวดล้อมแบบที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์ โดยที่การเคลื่อนที่ในระบบที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์นั้น เส้นทางที่ปลอดภัยที่สุดอาจจะแตกต่างไปจากแบบ 1 หุ่นยนต์ ค่า OA ในระบบ 1 หุ่นยนต์ที่งานวิจัยนี้ออกแบบ เป็นการค้นหาเส้นทางที่ปลอดภัยที่สุดเมื่อหลบสิ่งกีดขวางที่อยู่นิ่ง (static obstacle) แต่ในระบบที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์นั้น เส้นทางที่ปลอดภัยที่สุดควรเป็นเส้นทางที่หลบสิ่งกีดขวางที่อยู่นิ่งและ สิ่งกีดขวางที่เคลื่อนที่อยู่ (หุ่นยนต์ตัวอื่นๆ) งานวิจัยนี้จึงดำเนินการออกแบบเพื่อรองรับการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ที่มากกว่า 1 หุ่นยนต์ โดยเริ่มต้นจากศึกษาการค้นหาเส้นทางที่ดีที่สุดจากงานวิจัยที่เกี่ยวข้อง โดยพบว่าในระบบการวางแผนเส้นทางการเคลื่อนที่ที่เป็นที่นิยมคือ วิธีการ A* จึงเลือกใช้วิธี A* สำหรับวิธีการในการค้นหาเส้นทางการเคลื่อนที่หลักของหุ่นยนต์ แล้วจึงนำการค้นหาเส้นทางที่ปลอดภัยที่สุดด้วยฟังก์ชัน OA เข้ามาเป็นเงื่อนไขในการหาเส้นทางที่ดีที่สุดด้วย A* จากสมการที่ (3.12) พบว่า ฟังก์ชัน OA ถูกนำไปบวกกับฟังก์ชันของการหาเส้นทางรวมในการเคลื่อนที่เป็นวิธีการ modified A* โดยค่า β แทนค่า safety factor

$$f(n) = g(n) + h(n) + \beta * OA(n) \quad (3.12)$$

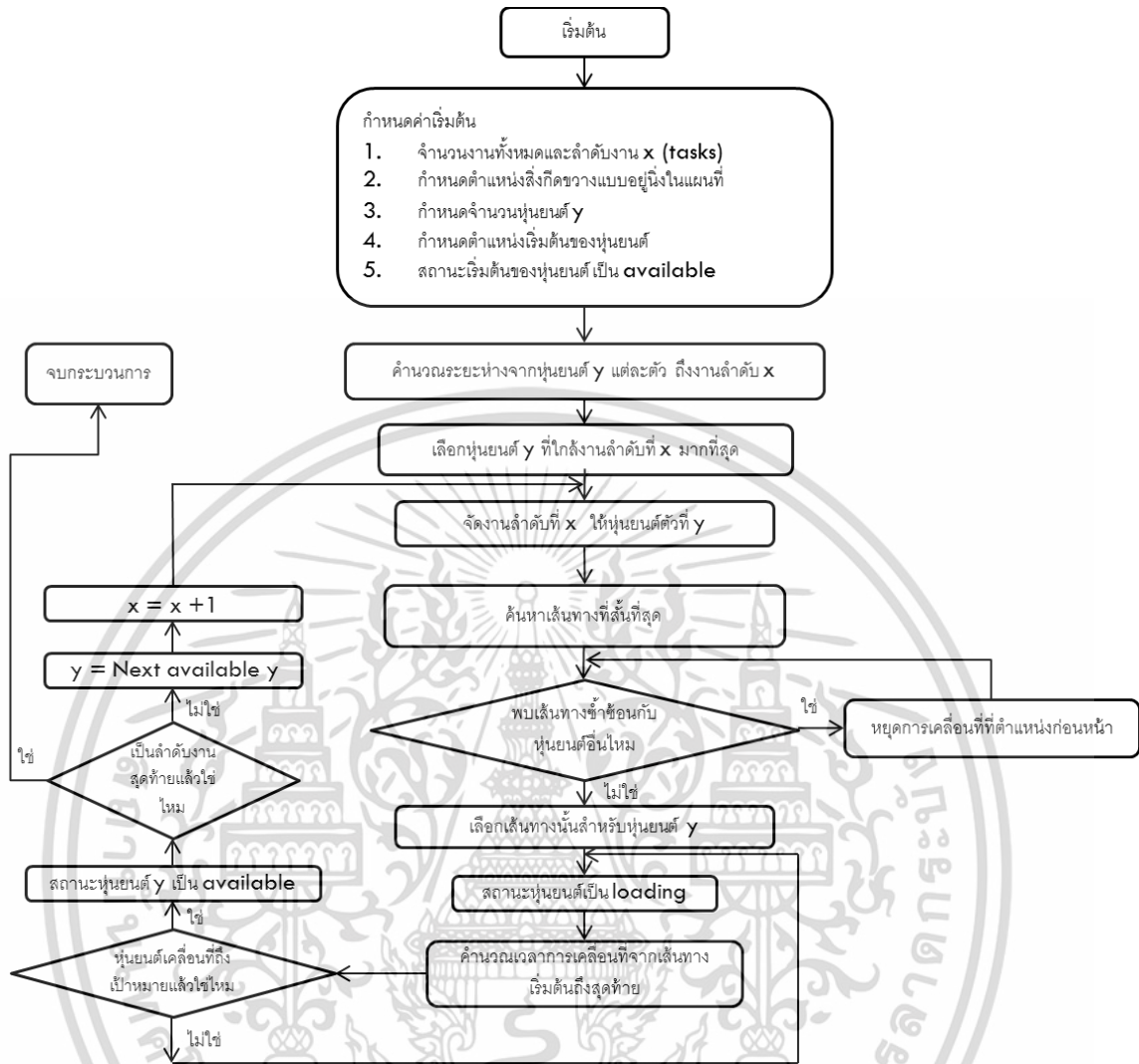


รูปที่ 3.8 กราฟการค้นหาค่า OA ที่น้อยที่สุดกับค่า β

งานวิจัยนี้ทำการค้นหาค่า β ที่เหมาะสมจากการทดลองทดสอบด้วยการหาค่า OA ที่ดีที่สุดจากช่วง $[1, 15]$ จำนวน 100,000 กรณี พบว่าค่า OA ที่น้อยที่สุดมีค่าเป็น 0.4748 ที่ค่า β เท่ากับ 13.2541 ซึ่งค่านี้จะเป็นค่า global minimum หลังจากนั้นจึงทำการทดลองค้นหาค่าที่ดีที่สุดด้วย ABC ดังรูปที่ 3.8 เพื่อให้การค้นหามีความรวดเร็วขึ้นพบว่า OA ที่น้อยที่สุดมีค่าเป็น 0.4749 ที่ค่า β เท่ากับ 13.2315 ซึ่งมีค่าเข้าใกล้ global minimum ซึ่งแสดงให้เห็นว่า ABC สามารถค้นหาค่าที่ดีที่สุดได้ใกล้เคียงกับค่า global minimum งานวิจัยนี้จึงเลือกใช้ค่า β ที่ 13.2315 ในสมการที่ (3.12)

จากรูปที่ 3.9 แสดงถึงขั้นตอนการหาเส้นทางการเคลื่อนที่ที่ดีที่สุดบนสภาพแวดล้อมที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์ โดยเริ่มต้นจากการกำหนดจำนวนงาน (tasks) และลำดับงาน กำหนดตำแหน่งสิ่งกีดขวางแบบอยู่ยั้งยั่งทั้งหมดในแผนที่โดยการกำหนดให้ตำแหน่งช่องกริดในแผนที่ที่มีสิ่งกีดขวางมีค่าเป็น 1 และตำแหน่งช่อง กริดในแผนที่ที่ไม่มีสิ่งกีดขวางมีค่าเป็น 0 กำหนดจำนวนหุ่นยนต์ทั้งหมดที่ต้องการใช้ในแผนที่ กำหนดค่าความปลอดภัย (safety factor) กำหนดตำแหน่งเริ่มต้นของหุ่นยนต์ และกำหนดสถานะเริ่มต้นของหุ่นยนต์ ซึ่งหุ่นยนต์จะมีอยู่สองสถานะคือ สถานะว่าง (available) กับสถานะทำงาน (loading) หลังจากนั้น ทำการคำนวณระยะห่างของหุ่นยนต์แต่ละตัวถึงตำแหน่งงานลำดับที่ x เมื่อ x คือ ลำดับของงานที่ถูกกำหนดเริ่มต้นไว้ หุ่นยนต์ที่อยู่ใกล้กับงานลำดับที่ x มากที่สุด จะได้รับเลือกให้ทำงาน ลำดับที่ x เมื่อได้รับงานแล้ว จะเข้าสู่กระบวนการการคำนวณหาเส้นทางที่ดีที่สุด ด้วยวิธีการ modified A* โดยแต่ละตำแหน่งในการคำนวณใน modified A* นั้น จะมีการเลือกตำแหน่งที่มีค่าของผลรวมของ $g(n)$ กับ $h(n)$ และ $OA(n)$ ที่น้อยที่สุด เพื่อให้ได้เส้นทางที่ปลอดภัยและระยะทางรวมที่เหมาะสม ในระหว่างการคำนวณ จะมีการเพิ่มการตรวจสอบเส้นทางซ้ำซ้อนของหุ่นยนต์ในแต่ละคู่ด้วยการใช้วิธีการจัด

หมู่ $C_{n,r} = \frac{n!}{(n-r)!r!}$ เมื่อ n คือ จำนวนหุ่นยนต์ทั้งหมด และ $r = 2$ ตัวอย่างเช่น ในระบบมีหุ่นยนต์ 4 ตัว จะได้ $C_{4,2} = \frac{4!}{(4-2)!2!} = 6$ แปลว่าจะมีการตรวจสอบเส้นทางเข้าซ้อนของหุ่นยนต์เมื่อระบบมีหุ่นยนต์ 4 ตัว ทั้งหมด 6 เงื่อนไขการตรวจสอบ คือ (R_1, R_2) , (R_1, R_3) , (R_1, R_4) , (R_2, R_3) , (R_2, R_4) , (R_3, R_4) เมื่อ R_x คือหุ่นยนต์ตัวที่ x ซึ่งเป็นการคำนวณให้ครบทุกเงื่อนไขการพบกัน หากพบเส้นทางที่เข้าซ้อนกันบนตำแหน่งถัดไปของการเคลื่อนที่ ณ เวลาเดียวกันนั้น จะมีการตรวจสอบว่าหุ่นยนต์ตัวไหนที่เหลือเส้นทางเคลื่อนที่มากกว่า จาก ณ ตำแหน่งนั้น หุ่นยนต์ที่เหลือเส้นทางเคลื่อนที่มากกว่า จะถูกนำเส้นทางที่เข้าซ้อนออกจากการคำนวณและเข้าสู่กระบวนการวางแผนเส้นทางเคลื่อนที่ด้วย modified A* อีกครั้ง หรือเรียกว่าการวางแผนเส้นทางใหม่ ส่วนหุ่นยนต์อีกตัวที่เหลือเส้นทางเคลื่อนที่น้อยกว่า จะถูกสั่งให้หยุด ณ ตำแหน่งนั้นจนกว่าจะไม่พบเส้นทางเข้าซ้อน แล้วจึงเคลื่อนที่ต่อ เมื่อหุ่นยนต์เคลื่อนที่ จะถูกเปลี่ยนสถานะจากสถานะว่าง เป็นสถานะทำงาน และเมื่อหุ่นยนต์เคลื่อนที่ถึงตำแหน่งงานเป้าหมายแล้ว หุ่นยนต์จะถูกเปลี่ยนสถานะจากสถานะทำงานเป็นสถานะว่าง และเข้าสู่การตรวจสอบจำนวนงานที่เหลือ หากพบว่ามีงานที่เหลืออยู่ จะเข้าสู่กระบวนการค้นหาหุ่นยนต์ที่มีสถานะว่างแล้วนำหุ่นยนต์เข้าไปทำงานตามกระบวนการข้างต้นนี้ จนกว่างานทั้งหมดจะถูกทำจนครบและสิ้นสุดกระบวนการ



รูปที่ 3.9 ขั้นตอนของวิธีการหาเส้นทางเคลื่อนที่ที่ดีที่สุดบนสภาพแวดล้อมที่มีหน่วยย่มากกว่า 1 หน่วยย่ด้วยวิธี Conventional

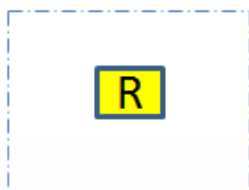
381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

รูปที่ 3.10 ตำแหน่งของสิ่งกีดขวางบนแผนที่

จากรูปที่ 3.10 แสดงตำแหน่งของสิ่งกีดขวางบนแผนที่ที่กำหนด โดยบนแผนที่จะมีตำแหน่งช่องกริดจาก 1 ถึง 400 ซึ่งตำแหน่งของช่องกริด 5, 8, 11, 14, 17, 25, 28, 31, 34, 37, 65, 68, 71, 74, 77, 85, 88, 91, 94, 97, 125, 128, 131, 134, 137, 145, 148, 151, 154, 157, 185, 188, 191, 194, 197, 205, 208, 211, 214, 217, 245, 248, 251, 254, 257, 265, 268, 271, 274, 277, 305, 308, 311, 314, 317, 367, 368, 369, 373, 374, 375, 387, 388, 389, 393, 394, 395 จะถูกกำหนดค่าเป็น 1 แต่ตำแหน่งของช่องกริดอื่นๆ นอกเหนือจากนี้จะถูกกำหนดค่าเป็น 0 ซึ่งในการหลบสิ่งกีดขวางที่อยู่หนึ่งทำได้โดยการนำตำแหน่ง ช่องกริดที่มีค่าเป็น 1 ออกจากกระบวนการคำนวณเส้นทางที่ดีที่สุดด้วย modified A*

ค่าของ OA ภายใต้แผนที่ในแต่ละตำแหน่งช่องกริด จะมีอยู่ 3 แบบด้วยกัน

1. แบบไม่มีสิ่งกีดขวางรอบตัว : ค่า OA = 0



2. แบบมีสิ่งกีดขวางฝั่งเดียว: ค่า OA จะอยู่ในช่วง 0 ถึง 0.15



3. แบบมีสิ่งกีดขวางสองฝั่ง: ค่า OA จะอยู่ในช่วง 0.15 ถึง 0.30



การคำนวณ modified A* ในแบบจำลองของงานวิจัยนี้ คือการเลือกเส้นทางที่ปลอดภัยที่สุดและระยะทางสั้นที่สุด ตัวอย่างการคำนวณการเคลื่อนที่ในแผนที่จากจุดเริ่มต้นที่ตำแหน่งช่องกริด 347 ไปยังจุดเป้าหมายที่ตำแหน่งช่องกริด 267 จากรูปที่ 3.10 เมื่อหุ่นยนต์อยู่ที่ตำแหน่งช่องกริด 347 หุ่นยนต์จะมีทางเลือกสามทางคือ ตำแหน่งช่องกริด 346 ตำแหน่งช่องกริด 327 และตำแหน่งช่องกริด 348 สำหรับเป็นทางเลือกในการเดินทางไปยังตำแหน่งช่องกริด 267 ซึ่งเป็นตำแหน่งเป้าหมาย

381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

รูปที่ 3.11 ตำแหน่งชองกริด 347 เริ่มต้นและทางเลือกเส้นทางไปยังตำแหน่งชองกริด 267

รูปที่ 3.11 แสดงตำแหน่งชองกริด 347 เริ่มต้นและทางเลือกเส้นทางไปยังตำแหน่งชองกริด 267 จากการคำนวณ $f(n)$ ในสมการที่ (3.12) และกำหนดให้ $OA = 0.1200$ เมื่อพบตำแหน่งกริดที่มีสิ่งกีดขวางฝั่งเดียว และ $OA = 0.2400$ เมื่อพบตำแหน่งกริดที่มีสิ่งกีดขวางสองฝั่ง จะได้ว่า $f(346) = 1+5+0.1200 = 6.1200$ ที่ตำแหน่งกริด 346 ที่ตำแหน่งกริด 327 จะได้ $f(327) = 1+3+0.1200 = 4.1200$ และตำแหน่งกริด 348 จะได้ $f(348) = 1+5+0.1200 = 6.1200$ ซึ่งจะเห็นได้ว่าตำแหน่งกริด 327 มีค่า $f(n)$ น้อยที่สุด จึงได้รับเลือกให้เป็นตำแหน่งถัดไปใน modified A*

381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

รูปที่ 3.12 ตำแหน่งชองกริต 327 ที่ถูกเลือก และทางเลือกเส้นทางไปยังตำแหน่งชองกริต 267

จากรูปที่ 3.12 จากการที่ตำแหน่งกริต 327 ได้รับเลือก จะพบว่าเส้นทางที่จะเดินทางจากตำแหน่งกริต 327 ไปยังตำแหน่งเป้าหมาย 267 จะมี 3 ทางเลือกคือตำแหน่งกริต 326 ตำแหน่งกริต 307 และตำแหน่งกริต 328 จะได้ว่า $f(326) = 2+4+0.1200 = 6.1200$, $f(307) = 2+2+0.1200 = 4.1200$ และ $f(328) = 2+4+0.1200 = 6.1200$ ซึ่งจะได้ตำแหน่งกริต 307 ที่มีค่า $f(n)$ น้อยที่สุด จึงได้รับเลือกให้เป็นตำแหน่งถัดไปใน modified A* รูปที่ 3.13 แสดงตำแหน่งกริต 307 จากการที่ตำแหน่งกริต 307 ได้รับเลือก จะพบว่าเส้นทางที่จะเดินทางจากตำแหน่งกริต 307 ไปยังตำแหน่งเป้าหมาย 267 จะมี 2 ทางเลือกคือตำแหน่งกริต 306 และตำแหน่งกริต 287 โดยที่ตำแหน่งกริต 308 นั้นถูกจัดอยู่ในตำแหน่งสิ่งกีดขวาง จึงไม่นำมาพิจารณาในการคำนวณเส้นทางทางเลือก จะได้ว่า $f(306) = 3+3+0.1200 = 6.1200$ และ $f(287) = 3+1+0.1200 = 4.1200$ ซึ่งจะได้ตำแหน่งกริต 287 ที่มีค่า $f(n)$ น้อยที่สุด จึงได้รับเลือกให้เป็นตำแหน่งถัดไปใน modified A* รูปที่ 3.14 แสดงตำแหน่งกริต 287 จากการที่ตำแหน่งกริต 287 ได้รับเลือก จะพบว่าเส้นทางที่จะเดินทางจากตำแหน่งกริต 287 ไปยังตำแหน่งเป้าหมาย 267 จะมี 3 ทางเลือกคือตำแหน่งกริต 286 ตำแหน่งกริต 267 และตำแหน่งกริต 288 จะได้ว่า $f(286) = 4+2+0.1200 = 6.1200$, $f(267) = 4+0+0.1200 = 4.1200$ และ $f(288) = 4+2+0.1200 = 6.1200$ ซึ่งจะได้ตำแหน่งกริต 267 ที่มีค่า $f(n)$ น้อยที่สุด จึงได้รับเลือกให้เป็นตำแหน่งถัดไปใน modified A* และเป็นตำแหน่งเป้าหมายด้วย จึงสิ้นสุดกระบวนการการคำนวณ modified A*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

รูปที่ 3.13 ตำแหน่งชองกริด 307 ที่ถูกเลือก และทางเลือกเส้นทางไปยังตำแหน่งชองกริด 267

381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

รูปที่ 3.14 ตำแหน่งชองกริด 287 ที่ถูกเลือก และทางเลือกเส้นทางไปยังตำแหน่งชองกริด 267

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อเพิ่มประสิทธิภาพของแบบจำลอง งานวิจัยนี้จึงทำการศึกษาค่าความคลาดเคลื่อนที่อาจเกิดขึ้นได้ในระบบ ซึ่งแบ่งออกเป็นสองส่วนคือ ค่าความคลาดเคลื่อนที่เกิดจากเซนเซอร์วัดระยะห่างของหุ่นยนต์ถึงสิ่งกีดขวาง และค่าความคลาดเคลื่อนของตำแหน่งการเคลื่อนที่ของหุ่นยนต์ งานวิจัยนี้จึงเพิ่มค่าความคลาดเคลื่อนในระบบดังนี้

1. เพิ่มค่าความคลาดเคลื่อนจากเซนเซอร์วัดระยะห่างที่ค่า $\pm 0.2000\%$

ค่า d_r คือระยะห่างจากหุ่นยนต์ถึงสิ่งกีดขวาง ในสมการที่ (3.2) จะถูกปรับเป็นค่าดังสมการที่ (3.13) เมื่อ d_{r_ideal} คือค่าในสมการที่ (3.2) และค่า $rand(-0.2000\%, 0.2000\%)$ คือค่าสุ่มในช่วง $\pm 0.2000\%$

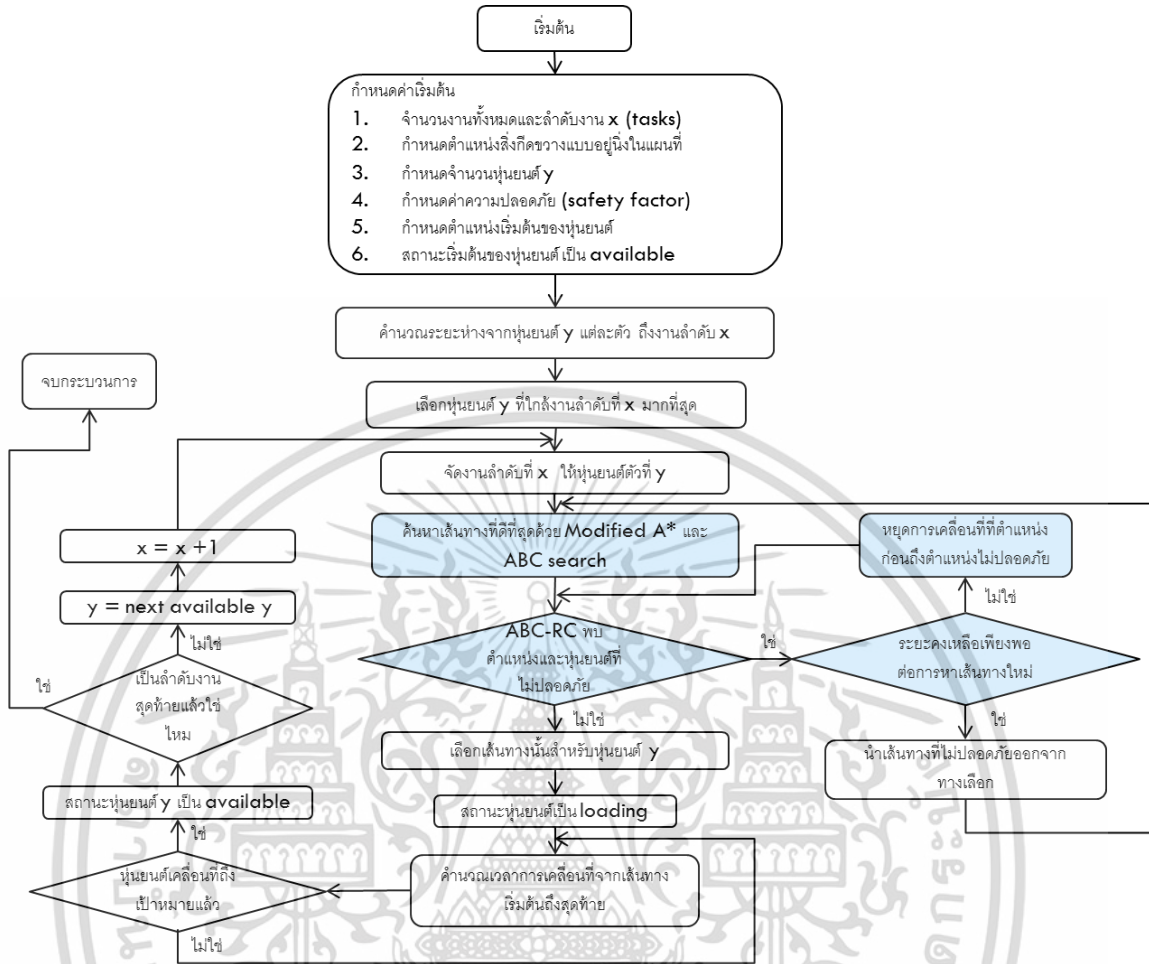
$$d_r = d_{r_ideal} + d_{r_ideal} * rand(-0.2000\%, 0.2000\%) \quad (3.13)$$

2. เพิ่มค่าความคลาดเคลื่อนของตำแหน่งการเคลื่อนที่ของหุ่นยนต์ที่ค่า $\pm 1\%$

ค่าตำแหน่งของหุ่นยนต์ (robot_position) ณ ขณะนั้น จะถูกปรับให้เป็นค่าดังสมการที่ (3.14) เมื่อ $robot_pos_{i,j_ideal}$ คือค่าตำแหน่งของหุ่นยนต์ตัวที่ i ตำแหน่งที่ j ณ ขณะนั้น และค่า $rand(-1\%, 1\%)$ คือค่าสุ่มในช่วง $\pm 1\%$

$$robot_pos_{i,j} = robot_pos_{i,j_ideal} + robot_pos_{i,j_ideal} * rand(-1\%, 1\%) \quad (3.14)$$

จากขั้นตอนของวิธีการดังรูปที่ 3.9 จะเห็นว่า เงื่อนไขการคำนวณเส้นทางที่ซ้ำซ้อนกันของหุ่นยนต์ และเลือกให้หุ่นยนต์หยุดการเคลื่อนที่นั้น หุ่นยนต์อาจหยุดการเคลื่อนที่ ณ ตำแหน่งที่ไม่เหมาะสม นอกจากนี้ การเลือกใช้การค้นหาเส้นทางที่สั้นที่สุดอาจไม่ใช่เส้นทางที่ปลอดภัย งานวิจัยนี้จึงปรับปรุงวิธีการวางแผนเส้นทางเคลื่อนที่ให้กับหุ่นยนต์ใหม่ด้วยการใช้ modified A* เพื่อค้นหาเส้นทางที่ปลอดภัยที่สุดแต่ยังคงได้ระยะทางที่เหมาะสมและนำ artificial bee colony (ABC) มาใช้ในการค้นหาเส้นทางที่มีความเสี่ยงในการชนกับหุ่นยนต์ตัวอื่นมากที่สุดด้วยฟังก์ชัน RC ณ ขณะนั้น ให้เป็นหุ่นยนต์ที่ต้องนำไปเข้าสู่กระบวนการค้นหาเส้นทางใหม่ด้วย modified A* ดังแสดงในรูปที่ 3.15



รูปที่ 3.15 ขั้นตอนของวิธีการหาเส้นทางเคลื่อนที่ที่ดีที่สุดบนสภาพแวดล้อมที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์แบบใช้ OA ในการหลบสิ่งกีดขวางแบบคงที่ และใช้ RC สำหรับหาเส้นทางที่มีความเสี่ยงในการชนระหว่างหุ่นยนต์สูงสุด

ขั้นตอนการคำนวณการค้นหาเส้นทางที่มีความเสี่ยงในการชนกับหุ่นยนต์ตัวอื่นมากที่สุดด้วย ABC นั้น เริ่มต้นด้วยการกำหนดฟังก์ชันวัตถุประสงค์ (cost function) โดยงานวิจัยนี้ออกแบบฟังก์ชัน Risk of Collision (RC) เพื่อแทนความสัมพันธ์ของระยะห่างระหว่างหุ่นยนต์ ดังแสดงในสมการที่ (3.15) เมื่อ β เป็นค่าคงที่ของ RC ในงานวิจัยนี้เลือกใช้ที่ค่า 0.8000 d_i คือระยะห่างระหว่างหุ่นยนต์ด้วยการคำนวณ Manhattan distance ในสมการที่ (2.2) และ η คือค่าคงที่ของอัตราส่วน RC งานวิจัยนี้เลือกใช้ค่า η คือ 100 เป็นค่าเริ่มต้น

$$RC_i = \max(1 - \beta * \frac{d_i}{\eta}, 0) \quad (3.15)$$

ตารางที่ 3.2 แสดงเงื่อนไขของค่า RC

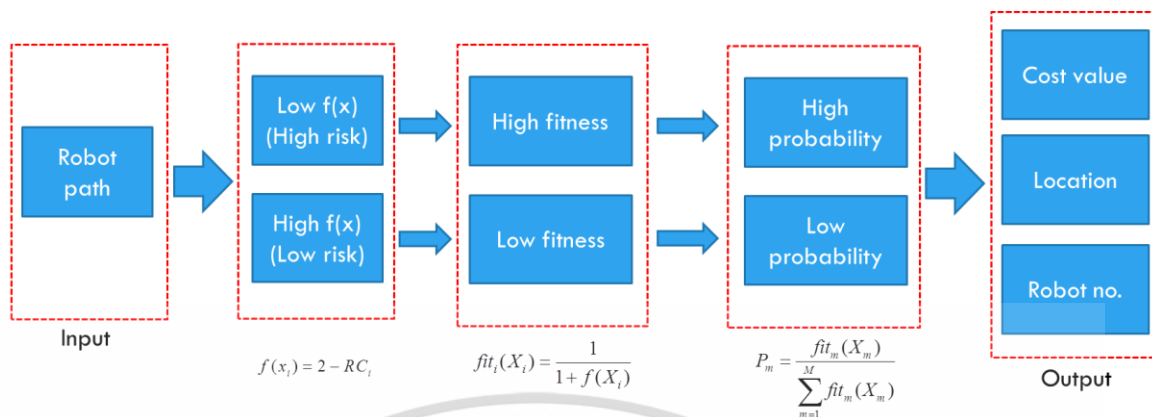
ลำดับที่	เงื่อนไข	ค่าของ RC	การประเมินความเสี่ยง
1	d น้อย	เข้าใกล้ 1	มีความเสี่ยงต่อการชนมาก
2	d มาก	0	มีความเสี่ยงต่อการชนน้อย

เมื่อกำหนด RC ให้เป็น cost function ของ ABC งานวิจัยนี้จึงทำการออกแบบข้อมูลขาเข้าและข้อมูลขาออกที่ต้องการจากการคำนวณในกระบวนการของ ABC ดังแสดงในสมการที่ (3.16) ซึ่งกำหนดข้อมูลขาเข้าให้เป็นเส้นทางของหุ่นยนต์แต่ละตัว ณ ขณะนั้น และข้อมูลขาออกเป็นค่าของ cost function ที่น้อยที่สุด

$$Output(Cost, Location, Robot_no.) = ABC(Robot_path) \quad (3.16)$$

การค้นหาของ ABC จะใช้วิธีการคำนวณ fitness function ดังสมการที่ (2.11) ซึ่งเมื่อประยุกต์เข้ากับฟังก์ชัน RC แล้ว จะได้ว่า $f(x) = 2 - RC(x)$ เมื่อแทนค่า จะได้สมการ fitness function ดังสมการที่ (3.17)

$$fit_i(X_i) = \frac{1}{1 + f(X_i)} \quad (3.17)$$



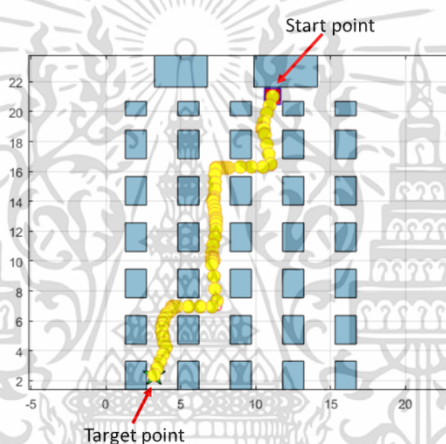
รูปที่ 3.16 กระบวนการการค้นหาเส้นทางที่มีความเสี่ยงต่อการชนหุ่นยนต์อื่นมากที่สุดด้วย Artificial Bee Colony

จากรูปที่ 3.16 แสดงถึงกระบวนการค้นหาเส้นทางที่มีความเสี่ยงต่อการชนหุ่นยนต์อื่นมากที่สุดด้วย Artificial Bee Colony โดยเริ่มต้นจากการนำข้อมูลขาเข้าเป็นข้อมูลของเส้นทางหุ่นยนต์ทุกตัว ณ ขณะนั้น แล้วนำมาคำนวณหาค่า RC ซึ่งค่า RC ที่มาก แสดงถึงความเสี่ยงในการชนที่สูง และค่า RC ที่น้อย แสดงถึงความเสี่ยงในการชนที่น้อย หลังจากนั้น นำค่า RC ซึ่งถูกกำหนดเป็น cost function ไปใส่ในสมการการหาค่า fitness ของ ABC ซึ่งจะได้ว่า เส้นทางที่มีความเสี่ยงในการชนที่สูง จะได้ค่า fitness ที่สูง และเส้นทางที่มีความเสี่ยงในการชนที่ต่ำ จะมีค่า fitness ที่น้อย ซึ่งเมื่อนำค่า fitness ไปคำนวณค่าความน่าจะเป็น จะได้ค่าความน่าจะเป็นสูงที่ fitness ที่มีค่ามาก และ ความน่าจะเป็นมีค่าน้อย เมื่อ fitness มีค่าน้อย ซึ่งผลเฉลยของการค้นหาคือเส้นทางที่มีความเสี่ยงในการชนมากที่สุด เพื่อเข้าสู่กระบวนการค้นหาเส้นทางใหม่ต่อไป

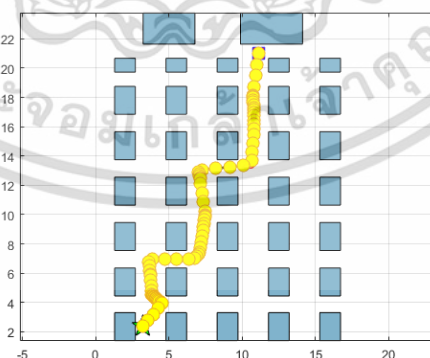
บทที่ 4

ผลการทดลองและการวิเคราะห์ผลการทดลอง

ผลลัพธ์จากการทดลองแบบ 1 ยานยนต์ ด้วยการค้นหาเส้นทางที่ปลอดภัยที่สุดด้วย PSO ค่าเริ่มต้นของ PSO ที่ถูกใช้ในการทดลองนี้คือ ค่า iterations = 200 รอบ จำนวนจุดที่ใช้ต่อ 1 เส้นทางคือ 30 จุดและจำนวนประชากรใน PSO หรือ swarm size คือ 200 ภายใต้สภาพแวดล้อมจำลองขนาด 20×22 unit² ค่าของ inertia weight = 1 และค่าของ inertia weight damping ratio ที่ 0.9900 ค่า acceleration coefficient ถูก set ที่ 1.4000 เพราะเป็นค่าที่ทำให้ระบบมีการค้นหาซ้ำที่คงที่ที่สุด



รูปที่ 4.1 จุดเริ่มต้นและจุดเป้าหมายในแผนที่ในแบบจำลอง



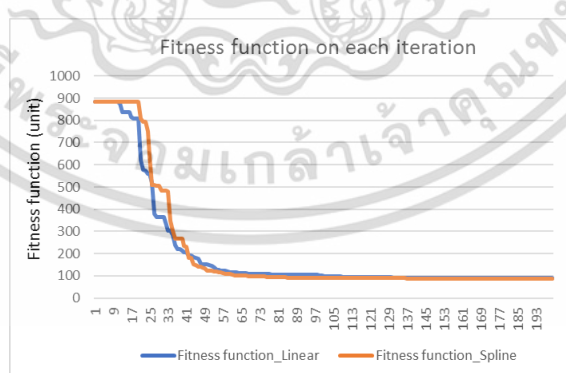
รูปที่ 4.2 จุดเริ่มต้นและจุดเป้าหมายในแผนที่ในแบบจำลองด้วย linear interpolation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



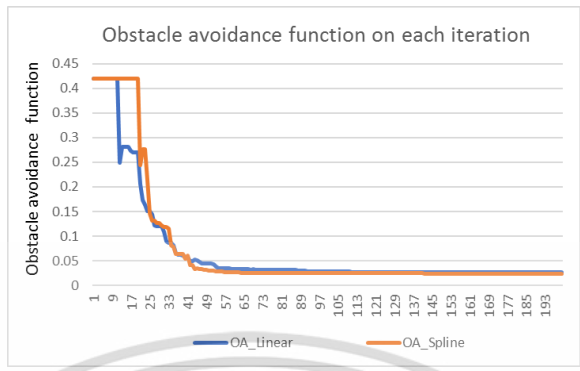
รูปที่ 4.3 จุดเริ่มต้นและจุดเป้าหมายในแผนที่ในแบบจำลองด้วย cubic-spline interpolation

จากรูปที่ 4.2 และ 4.3 พบว่าการเลือกใช้ cubic-spline interpolation นั้นให้เส้นทางที่ราบรื่นกว่าในแบบจำลองเช่นกัน และจากรูปที่ 4.4 และ 4.5 ก็พบว่าแบบ cubic-spline interpolation นั้นให้ค่า fitness และค่า OA ที่ดีกว่าเช่นกัน เนื่องจากการคำนวณการสร้างเส้นของ cubic-spline interpolation ใช้พหุนามดีกรี 3 ซึ่งมีลักษณะเป็นเส้นโค้ง ซึ่งมีผลต่อค่า $\frac{d}{R+r}$ จากตารางที่ 3.1 จึงทำให้ค่า OA ลู่เข้าค่า 0 ได้มากกว่า แต่ข้อเสียของ cubic-spline interpolation คือในจุดแรกๆ จะยังไม่สามารถหาเส้นได้เนื่องจากจำนวนจุดมีน้อยเกินไป ซึ่งอาจส่งผลให้การเคลื่อนที่ในช่วงจุดแรกๆ นั้น มีความเสี่ยงสูงต่อการชน งานวิจัยนี้จึงทำการเพิ่มจำนวนจุดทั้งหมดตลอดช่วงจาก 100 จุดเป็น 200 จุด เพื่อให้ระยะระหว่างจุดสั้นลงและโอกาสผิดพลาดในช่วงแรกมีน้อยมากๆ ซึ่งผลการทดลองในรูปที่ 4.6 และ 4.7 แสดงให้เห็นถึงประสิทธิภาพที่ดีขึ้นในการค้นหา global minimum

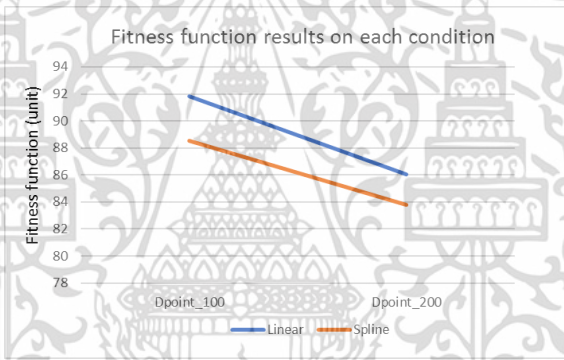


รูปที่ 4.4 ค่าของ fitness function ในแต่ละ iteration เปรียบเทียบระหว่าง linear interpolation กับ cubic-spline interpolation

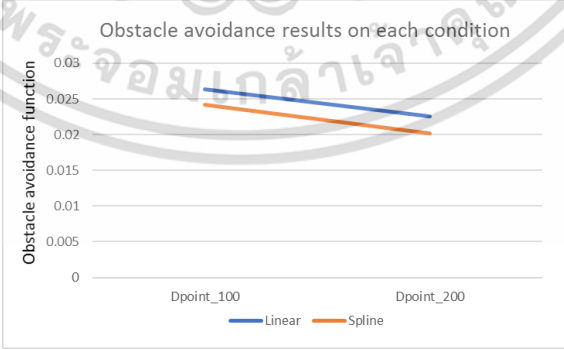
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ค่าของ OA ในแต่ละ iteration เปรียบเทียบระหว่าง linear interpolation กับ cubic-spline interpolation



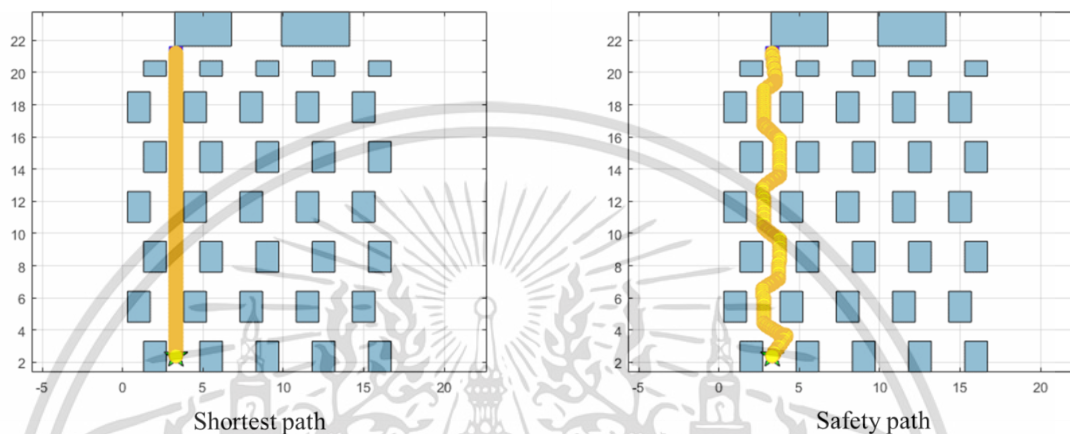
รูปที่ 4.6 ค่าของ fitness function เปรียบเทียบระหว่าง 100 จุดและ 200 จุด



รูปที่ 4.7 ค่าของ OA เปรียบเทียบระหว่าง 100 จุดและ 200 จุด

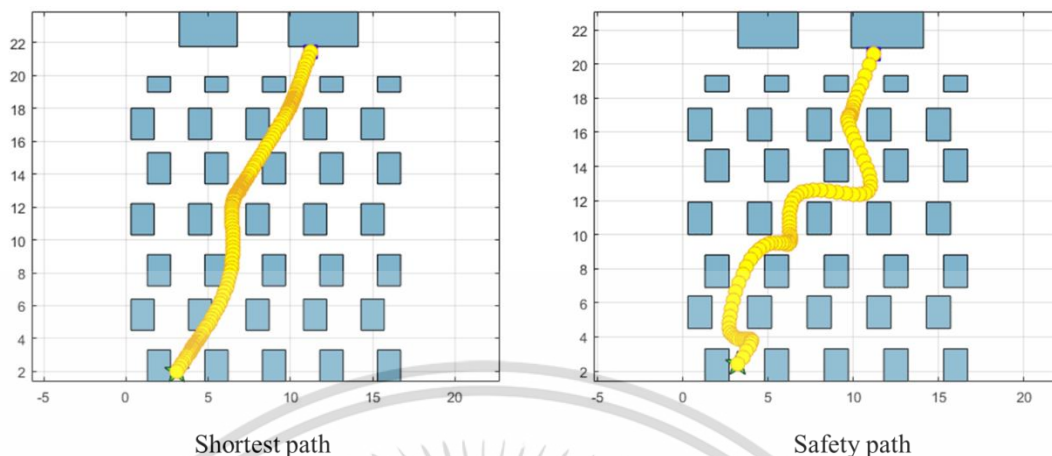
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากได้ค่า setting ที่เหมาะสมจากการเลือกใช้ cubic-spline interpolation จึงทำการทดลองเปรียบเทียบระหว่างระบบที่ค้นหาเส้นทางที่ดีที่สุดจากเส้นทางที่สั้นที่สุดกับวิธีการค้นหาเส้นทางที่ปลอดภัยที่สุดที่งานวิจัยนี้นำเสนอ

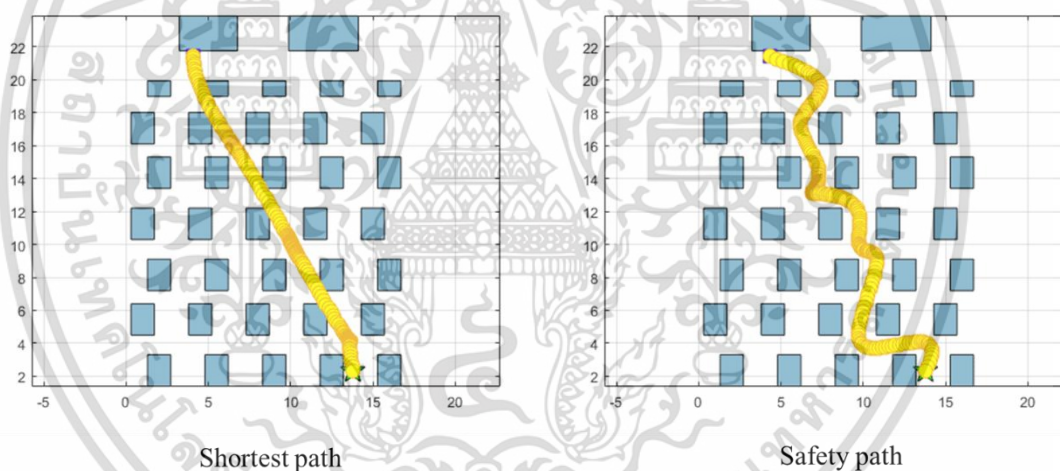


รูปที่ 4.8 ลักษณะของเส้นทางการค้นหาระหว่างการค้นหาเส้นทางที่สั้นที่สุดกับการค้นหาเส้นทางที่ปลอดภัยที่สุด แบบที่ 1

จากรูปที่ 4.8 แสดงถึงลักษณะของเส้นทางการค้นหาจากการทดลองในแบบจำลองของงานวิจัยนี้ แล้วพบว่าระยะทางรวมของการหาเส้นทางที่สั้นที่สุดเป็น 18.8100 แต่การหาเส้นทางที่ปลอดภัยที่สุดเป็น 22.2900 ซึ่งมีค่ามากกว่า แต่ค่าความเสี่ยงต่อการชน หรือค่า OA นั้น ในการหาเส้นทางที่สั้นที่สุดมีค่า 0.1650 แต่การค้นหาด้วยเส้นทางที่ปลอดภัยที่สุดนั้นมีค่าเพียง 0.0070 จึงทำการทดลองเพิ่มในแบบที่แตกต่างออกไป จากรูปที่ 4.9 ได้ระยะทางรวมเป็น 21.8700 และ 25.5100 บนวิธีการค้นหาเส้นทางที่สั้นที่สุดและการค้นหาเส้นทางที่ปลอดภัยที่สุดตามลำดับ และ มีค่าของความเสี่ยงต่อการชนหรือค่า OA ของวิธีการค้นหาเส้นทางที่สั้นที่สุดและการค้นหาเส้นทางที่ปลอดภัยที่สุดมีค่าเป็น 0.2244 และ 0.0081 ตามลำดับ ในรูป 4.10 ก็ให้ผลลัพธ์ในทิศทางเดียวกัน วิธีการค้นหาเส้นทางที่สั้นที่สุดและการค้นหาเส้นทางที่ปลอดภัยที่สุด เป็น 21.7500 และ 28.9700 และค่าความเสี่ยงต่อการชนเป็น 0.2062 และ 0.0083 ตามลำดับ



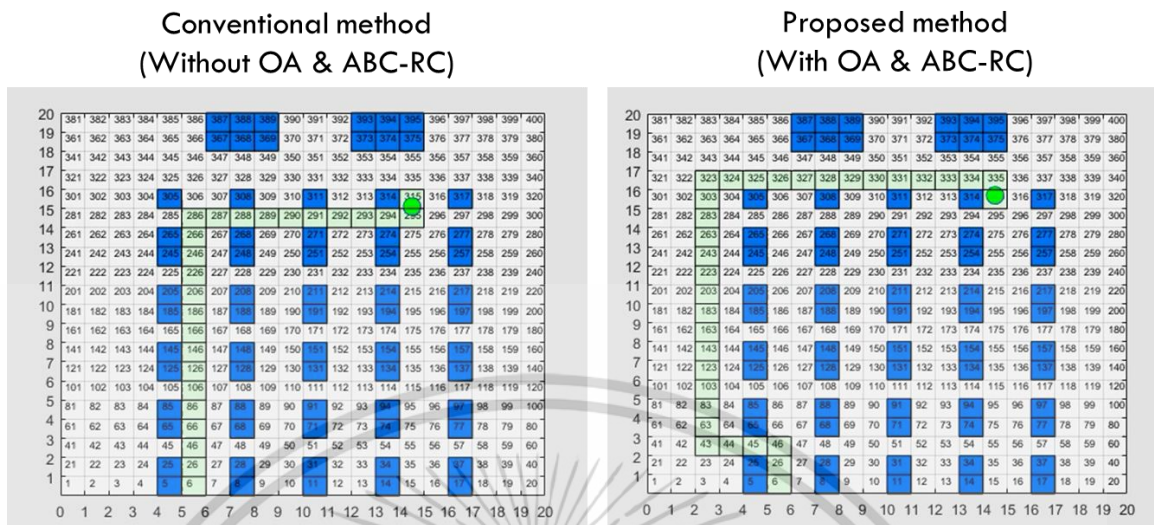
รูปที่ 4.9 ลักษณะของเส้นทางการค้นหาระหว่างการค้นหาเส้นทางที่สั้นที่สุดกับการค้นหาเส้นทางที่ปลอดภัยที่สุด แบบที่ 2



รูปที่ 4.10 ลักษณะของเส้นทางการค้นหาระหว่างการค้นหาเส้นทางที่สั้นที่สุดกับการค้นหาเส้นทางที่ปลอดภัยที่สุด แบบที่ 3

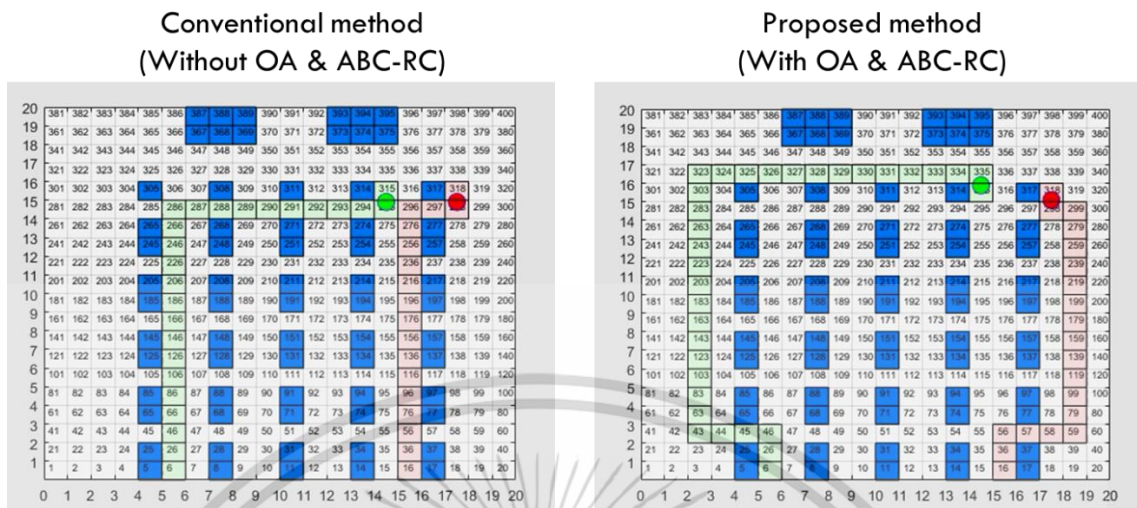
หลังจากได้ผลลัพธ์ที่ตีบนการวางแผนเส้นทางการเคลื่อนที่ที่ 1 หุ่นยนต์ จึงทำการออกแบบแบบจำลองเพิ่มเพื่อรองรับสภาพแวดล้อมจริงของโรงงานอุตสาหกรรม โดยเพิ่มจำนวนหุ่นยนต์เข้ามาในระบบ และออกแบบเงื่อนไขการหลบหลีกเส้นทางที่ซ้อนทับกัน นอกจากนี้ งานวิจัยนี้ยังเพิ่มสิ่งรบกวนแบบค่าสุ่ม เข้าไปในระบบเพื่อตรวจสอบประสิทธิภาพของแบบจำลอง ผลการทดลองที่ได้เมื่อนำค่า OA และ RC มาใช้ในการค้นหาเส้นทางการเคลื่อนที่ด้วยวิธีการใหม่ที่งานวิจัยนี้นำเสนอ นั้น แสดงดังรูปที่ 4.11 ถึง 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 1 หุ่นยนต์

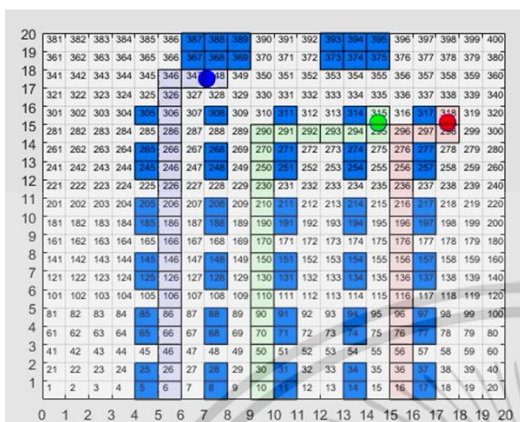
จากรูปที่ 4.11 แสดงให้เห็นถึงลักษณะของการวางแผนเส้นทางด้วยวิธีการ A* โดยทั่วไปซึ่งไม่มีการพิจารณาค่า OA กับ ABC-RC และการวางแผนเส้นทางด้วยวิธีการ OA และ ABC-RC ของงานวิจัยนี้ ซึ่งมีการเพิ่มการพิจารณาค่า OA และ RC เข้าไปในระบบ ผลลัพธ์ที่ได้พบว่า ค่าผลรวม OA ตลอดเส้นทางของวิธีการ conventional จะมีค่าเป็น 2.0465 ค่าเฉลี่ย RC เป็น 0 เพราะมี 1 หุ่นยนต์ ด้วยจำนวนตำแหน่งกริดที่เคลื่อนที่ 25 ตำแหน่ง และใช้เวลาในกระบวนการ 16.6445 วินาที ในส่วนของผลการทดลองเมื่อระบบมีการพิจารณาค่า OA กับ ABC-RC ที่งานวิจัยนี้นำเสนอ จะได้ค่าผลรวม OA ตลอดเส้นทาง มีค่าเป็น 1.0827 ซึ่งเป็นค่าที่แสดงถึงความเสี่ยงในการชนสิ่งกีดขวางที่อยู่หนึ่งในแผนที่ จะพบว่ามีค่า OA ที่น้อยกว่าการวางแผนเส้นทางแบบ conventional ซึ่งเป็นการพิจารณาเส้นทางที่สั้นที่สุดเท่านั้น แต่อย่างไรก็ตาม จำนวนตำแหน่งกริดที่เคลื่อนที่ในวิธีการใหม่ของงานวิจัยนี้คือ 33 ตำแหน่งและใช้เวลาในกระบวนการ 23.9610 วินาที ซึ่งจะมีค่าที่สูงกว่าวิธี conventional



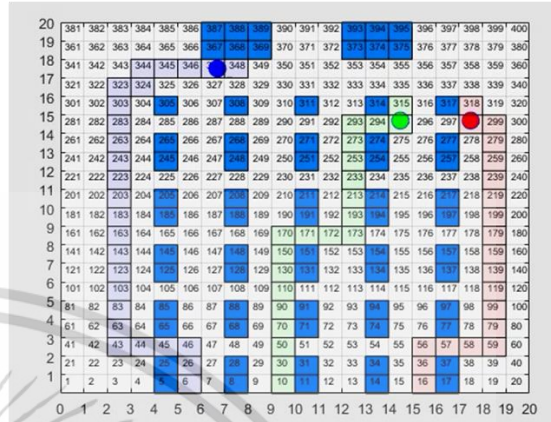
รูปที่ 4.12 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 2 หุ่นยนต์

จากรูปที่ 4.12 แสดงให้เห็นถึงลักษณะของการวางแผนเส้นทางของทั้ง 2 วิธีภายใต้สภาพแวดล้อมแบบ 2 หุ่นยนต์ จะเห็นได้ชัดเจนว่า การวางแผนเส้นทางที่มี OA กับ RC จะได้เส้นทางที่อ้อมกว่าเพื่อลดความเสี่ยงในการเข้าใกล้สิ่งกีดขวางที่อยู่ภายในระบบ ผลการทดลองพบว่า ที่หุ่นยนต์สีเขียวเริ่มทำงานจากตำแหน่งช่องกริด 315 ไปยังตำแหน่งช่องกริด 6 ได้ว่าวิธี conventional มีค่า OA ที่ 2.0451 ค่า RC ที่ 0.8662 จำนวนตำแหน่งเคลื่อนที่ 25 แต่วิธีการที่นำเสนอในงานวิจัยนี้มีค่า OA ลดน้อยลงเหลือเพียง 1.0802 กับค่า RC ลดน้อยลงเหลือ 0.8147 โดยจำนวนตำแหน่งเคลื่อนที่เพิ่มขึ้นเป็น 33 ที่หุ่นยนต์สีแดงเริ่มทำงานจากตำแหน่งช่องกริด 318 ไปยังตำแหน่งช่องกริด 16 ได้ว่าวิธี conventional มีค่า OA ที่ 1.5571 ค่า RC ที่ 0.9075 จำนวนตำแหน่งเคลื่อนที่ 18 แต่วิธีการที่นำเสนอในงานวิจัยนี้มีค่า OA ลดน้อยลงเหลือเพียง 0.7192 กับค่า RC ลดน้อยลงเหลือ 0.8504 โดยจำนวนตำแหน่งเคลื่อนที่เพิ่มขึ้นเป็น 20 จะเห็นได้ว่าผลของวิธีที่นำเสนอในงานวิจัยนี้ให้ค่าความเสี่ยงที่น้อยลงแต่จำนวนตำแหน่งเคลื่อนที่เพิ่มขึ้นและพบว่าเวลาที่ใช้ในกระบวนการของวิธี conventional เป็น 15.9647 วินาที แต่วิธีการที่งานวิจัยนี้เสนอจะใช้เวลาเพิ่มขึ้นเป็น 20.4999 วินาที

Conventional method
(Without OA & ABC-RC)



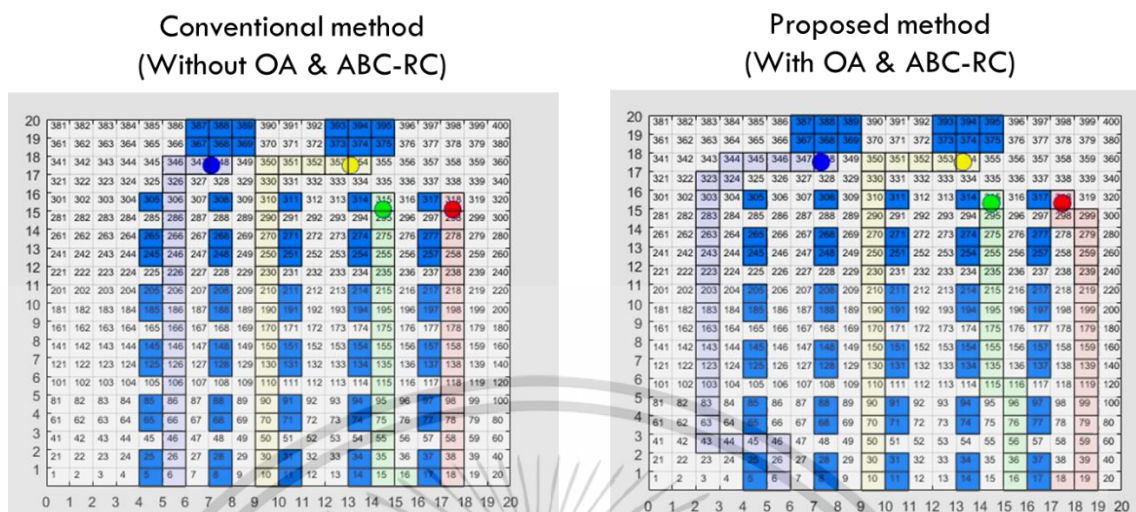
Proposed method
(With OA & ABC-RC)



รูปที่ 4.13 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 3 หุ่นยนต์

จากรูปที่ 4.13 แสดงให้เห็นถึงลักษณะของการวางแผนเส้นทางของทั้ง 2 วิธีภายใต้สภาพแวดล้อมแบบ 3 หุ่นยนต์ หุ่นยนต์สีเขียวเริ่มทำงานจากตำแหน่งช่องกริด 315 ไปยังตำแหน่งช่องกริด 10 มีค่า OA ของวิธี conventional ที่ 1.7959 ค่า RC ที่ 0.9385 และมีค่า OA ของวิธีที่งานวิจัยนี้เสนอที่ 1.7949 ค่า RC ที่ 0.9371 ด้วยจำนวนตำแหน่งเคลื่อนที่ที่ 21 เท่ากัน ซึ่งแปลว่าที่หุ่นยนต์สีเขียวให้ผลทั้งสองวิธีที่ค่าใกล้เคียงกัน ที่หุ่นยนต์สีแดงเคลื่อนที่จากตำแหน่งช่องกริด 318 ไปยังตำแหน่งช่องกริด 16 มีค่า OA ของวิธี conventional ที่ 1.5644 ค่า RC ที่ 0.9450 แต่มีค่า OA ลดลงเป็น 0.7184 กับ RC ที่ลดลงเป็น 0.9400 บนวิธีการที่นำเสนอในงานวิจัยนี้ และจำนวนตำแหน่งเคลื่อนที่มีค่าเป็น 18 และ 20 ตำแหน่งบนวิธี conventional และวิธีที่นำเสนอในงานวิจัยนี้ตามลำดับ หุ่นยนต์สีน้ำเงินเคลื่อนที่จากตำแหน่ง 348 ไปยังตำแหน่ง 6 โดยมีค่า OA, RC และตำแหน่งเคลื่อนที่ของวิธี conventional และวิธีที่นำเสนอในงานวิจัยนี้อยู่ที่ 1.5623, 0.9355 และ 20 กับ 0.7180, 0.8834 และ 26 ตามลำดับ เวลารวมในกระบวนการอยู่ของวิธี conventional อยู่ที่ 13.0446 วินาที และแบบวิธีที่นำเสนอในงานวิจัยนี้อยู่ที่ 16.1407 วินาที จึงเห็นได้ว่าผลการทดลองแบบ 3 หุ่นยนต์นั้น วิธีการที่นำเสนอในงานวิจัยนี้ให้ค่าความเสี่ยงในการชนสิ่งกีดขวางและความเสี่ยงในการชนหุ่นยนต์น้อยกว่าวิธีการ conventional แต่ใช้เวลาที่เพิ่มขึ้นกับจำนวนตำแหน่งที่เพิ่มขึ้น

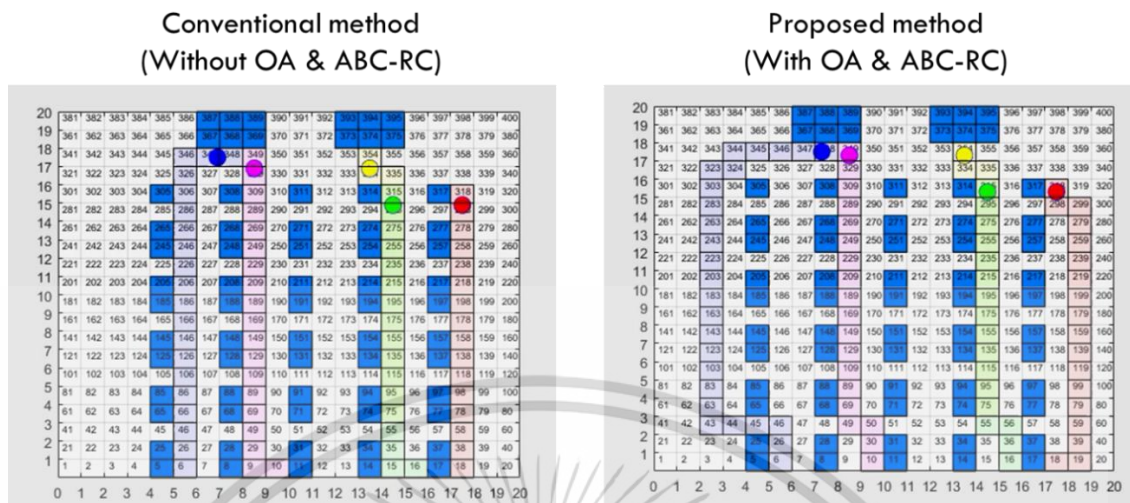
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 4 หุ่นยนต์

จากรูปที่ 4.14 แสดงให้เห็นถึงลักษณะของการวางแผนเส้นทางของทั้ง 2 วิธีภายใต้สภาพแวดล้อมแบบ 4 หุ่นยนต์ หุ่นยนต์สีเขียวเคลื่อนที่จากตำแหน่ง 315 ไปยังตำแหน่ง 16 โดยมีค่า OA เท่ากับ 1.4386 ค่า RC เท่ากับ 0.9678 และจำนวนตำแหน่งเคลื่อนที่เท่ากับ 17 บนวิธี conventional แต่บนวิธีการที่งานวิจัยนี้นำเสนอมีค่า OA เท่ากับ 1.3161 ค่า RC เท่ากับ 0.9715 และจำนวนตำแหน่งเคลื่อนที่เท่ากับ 17 เท่ากัน ซึ่งหมายความว่าทั้งสองวิธีให้ผลของเส้นทางที่ไม่แตกต่างกันบนหุ่นยนต์สีเขียว ที่หุ่นยนต์สีแดงเคลื่อนที่จากตำแหน่ง 318 ไปยังตำแหน่ง 18 โดยมีค่า OA เท่ากับ 1.3240 ค่า RC เท่ากับ 0.9790 และจำนวนตำแหน่งเคลื่อนที่เท่ากับ 16 บนวิธี conventional แต่บนวิธีการที่งานวิจัยนี้นำเสนอ มีค่า OA เท่ากับ 0.3585 ค่า RC เท่ากับ 0.9615 และจำนวนตำแหน่งเคลื่อนที่เท่ากับ 18 ที่หุ่นยนต์สีน้ำเงิน เคลื่อนที่จากตำแหน่ง 348 ไปยังตำแหน่ง 6 โดยมีค่า OA เท่ากับ 1.5625 ค่า RC เท่ากับ 0.9244 และจำนวนตำแหน่งเคลื่อนที่เท่ากับ 20 บนวิธี conventional แต่บนวิธีการที่งานวิจัยนี้นำเสนอ มีค่า OA เท่ากับ 0.7192 ค่า RC เท่ากับ 0.8939 และจำนวนตำแหน่งเคลื่อนที่เท่ากับ 26 หุ่นยนต์สีเหลืองเคลื่อนที่จากตำแหน่ง 354 ไปยังตำแหน่ง 10 โดยมีค่า OA เท่ากับ 1.5592 ค่า RC เท่ากับ 0.9134 และจำนวนตำแหน่งเคลื่อนที่เท่ากับ 22 บนวิธี conventional แต่บนวิธีการที่งานวิจัยนี้นำเสนอ มีค่า OA เท่ากับ 1.5636 ค่า RC เท่ากับ 0.9141 และจำนวนตำแหน่งเคลื่อนที่เท่ากับ 22 เท่ากัน เวลาที่ใช้ในกระบวนการคือ 13.8457 วินาที บนวิธี conventional และ 16.1685 บนวิธีการที่งานวิจัยนี้นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

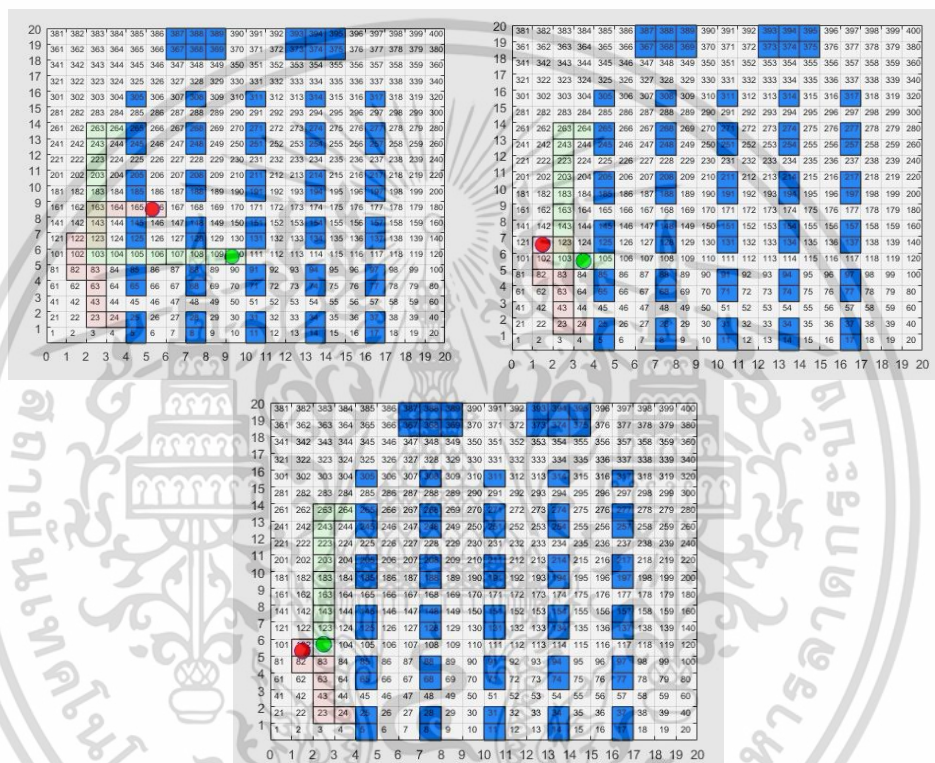


รูปที่ 4.15 ลักษณะของการวางแผนเส้นทางด้วย A* (without OA & ABC-RC) และ proposed method (With OA and ABC-RC) ภายใต้สภาพแวดล้อมแบบ 5 หุ่นยนต์

จากรูปที่ 4.15 แสดงให้เห็นถึงลักษณะของการวางแผนเส้นทางของทั้ง 2 วิธีภายใต้สภาพแวดล้อมแบบ 5 หุ่นยนต์ หุ่นยนต์สีเขียวเคลื่อนที่จากตำแหน่ง 315 ไปยังตำแหน่ง 16 ที่วิธี conventional มีค่า OA อยู่ที่ 1.4321 ค่า RC อยู่ที่ 0.9701 จำนวนตำแหน่งเคลื่อนที่เป็น 17 ส่วนวิธีการทำงานวิจัยนี้นำเสนอ มีค่า OA อยู่ที่ 1.3146 ค่า RC อยู่ที่ 0.9701 จำนวนตำแหน่งเคลื่อนที่เป็น 17 เป็น เท่ากัน หุ่นยนต์สีแดงเคลื่อนที่จากตำแหน่ง 318 ไปยังตำแหน่ง 18 ที่วิธี conventional มีค่า OA อยู่ที่ 1.3271 ค่า RC อยู่ที่ 0.9805 จำนวนตำแหน่งเคลื่อนที่เป็น 16 ส่วนวิธีการทำงานวิจัยนี้นำเสนอมีค่า OA อยู่ที่ 0.3607 ค่า RC อยู่ที่ 0.9523 จำนวนตำแหน่งเคลื่อนที่เป็น 18 หุ่นยนต์สีน้ำเงินเคลื่อนที่จากตำแหน่ง 348 ไปยังตำแหน่ง 6 ที่วิธี conventional มีค่า OA อยู่ที่ 1.5639 ค่า RC อยู่ที่ 0.9369 จำนวนตำแหน่งเคลื่อนที่เป็น 20 ส่วนวิธีการทำงานวิจัยนี้นำเสนอมีค่า OA อยู่ที่ 0.7194 ค่า RC อยู่ที่ 0.8890 จำนวนตำแหน่งเคลื่อนที่เป็น 26 ที่หุ่นยนต์สีเหลืองเคลื่อนที่จากตำแหน่ง 354 ไปยังตำแหน่ง 75 มีค่า OA อยู่ที่ 1.3177 ค่า RC อยู่ที่ 0.9805 จำนวนตำแหน่งเคลื่อนที่เป็น 16 ส่วนวิธีการทำงานวิจัยนี้นำเสนอมีค่า OA อยู่ที่ 1.3191 ค่า RC อยู่ที่ 0.9805 จำนวนตำแหน่งเคลื่อนที่เป็น 16 ที่หุ่นยนต์สีชมพูเคลื่อนที่จากตำแหน่ง 349 ไปยังตำแหน่ง 10 มีค่า OA อยู่ที่ 1.5600 ค่า RC อยู่ที่ 0.9438 จำนวนตำแหน่งเคลื่อนที่เป็น 19 ส่วนวิธีการทำงานวิจัยนี้นำเสนอมีค่า OA อยู่ที่ 1.4330 ค่า RC อยู่ที่ 0.9350 จำนวนตำแหน่งเคลื่อนที่เป็น 19 โดยเวลาที่ใช้กระบวนการเป็น 13.5341 วินาทีและ 16.6050 วินาทีบนวิธี conventional และวิธีการทำงานวิจัยนี้ นำเสนอตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

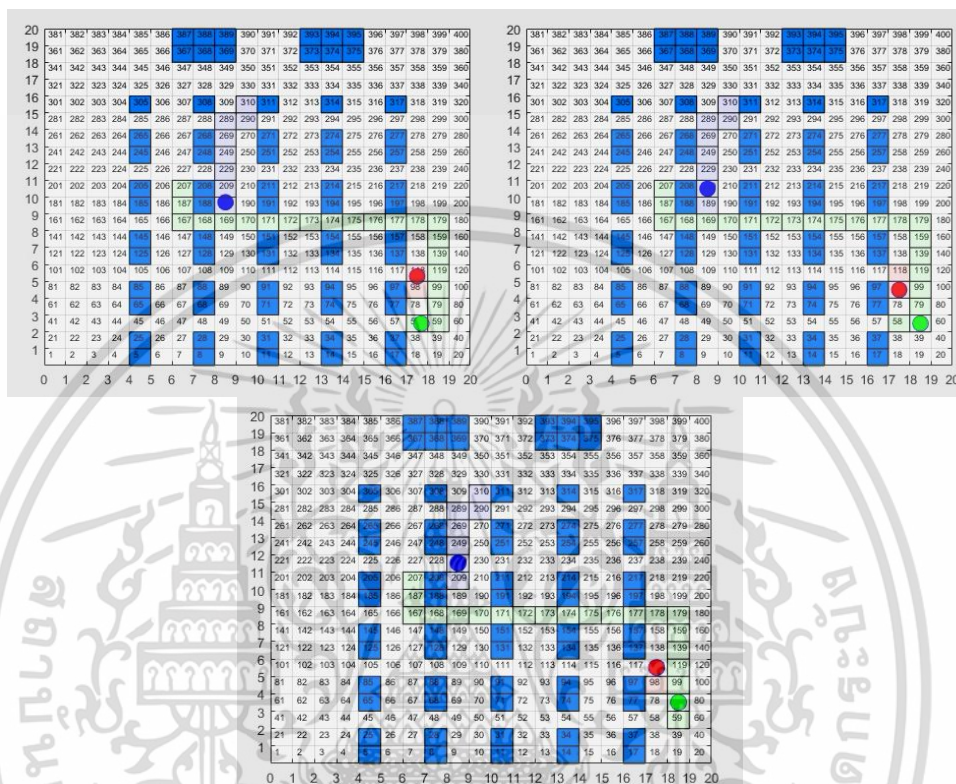
งานวิจัยนี้ทำการทดลองเพิ่มด้วยการจัดวางงานให้หุ่นยนต์จำนวน 30 งาน โดยมีลำดับงานเป็นตำแหน่งดังนี้ [304, 267, 213, 318, 195, 30, 27, 36, 93, 158, 310, 38, 98, 207, 264, 24, 258, 210, 184, 318, 306, 4, 10, 72, 150, 247, 313, 204, 96, 250] แล้วทดสอบการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ด้วยวิธีการที่นำเสนอใหม่ในงานวิจัยนี้ ในกรณีที่มีจำนวนงานในการทำงานของหุ่นยนต์มีเพิ่มขึ้น



รูปที่ 4.16 การหลบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 2 หุ่นยนต์

จากรูปที่ 4.16 แสดงการหลบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 2 หุ่นยนต์แล้วพบเหตุการณ์ที่มีเส้นทางซ้ำซ้อนกัน ระบบจะทำการหาเส้นทางใหม่เพื่อหลบหุ่นยนต์อีกตัว ในกรณีนี้ หุ่นยนต์สีแดงทำการวางแผนเส้นทางใหม่เพื่อหลบหุ่นยนต์สีเขียวที่กำลังเคลื่อนที่มาในเวลาเดียวกัน จึงเห็นได้ว่าขณะเคลื่อนที่หุ่นยนต์ไม่มีการชนกัน ณ ตำแหน่งที่มีโอกาสเป็นเส้นทางที่ซ้ำซ้อนกัน เวลารวมทั้งหมดที่ใช้ในการเคลื่อนที่ในระบบ 2 หุ่นยนต์ ด้วยการกำหนดงานทั้งหมด 30 งาน พบว่าใช้เวลาในกระบวนการ

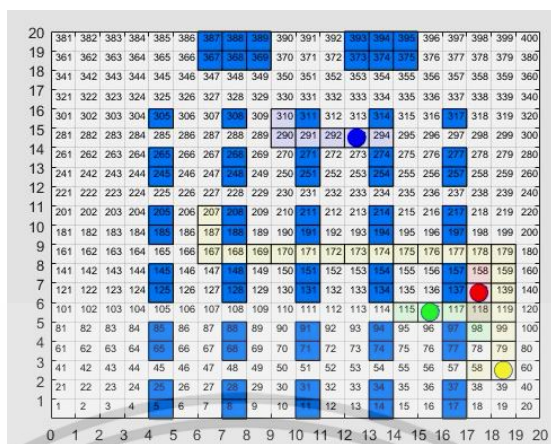
173.2814 วินาที โดยหุ่นยนต์สี่เหลี่ยมทำงาน 16 งาน หุ่นยนต์สี่แฉกทำงาน 14 งาน ค่าเฉลี่ยของผลรวม OA เป็น 1.5567 และค่าเฉลี่ยของ RC เป็น 0.9440



รูปที่ 4.17 การหลบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 3 หุ่นยนต์

จากรูปที่ 4.17 แสดงการหลบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 3 หุ่นยนต์แล้วพบเหตุการณ์ที่มีเส้นทางซ้ำซ้อนกัน ซึ่งให้ผลเหมือนกันกับการทดสอบในระบบ 2 หุ่นยนต์ แต่พบว่า เมื่อมีจำนวนหุ่นยนต์เพิ่มขึ้น เวลารวมที่ใช้ในระบบมีค่าลดลงเป็น 139.9298 วินาที เมื่อทำงานที่ 30 งาน เท่ากัน หุ่นยนต์สี่เหลี่ยมทำงานทั้งหมด 12 งาน หุ่นยนต์สี่แฉกทำงานทั้งหมด 8 งาน หุ่นยนต์ตัวสุดท้าย หุ่นยนต์สีน้ำเงิน ทำงานทั้งหมด 10 งาน จึงเห็นได้ว่า เมื่อจำนวนหุ่นยนต์ที่เพิ่มขึ้น จะมีการกระจายงานให้กับหุ่นยนต์แต่ละตัว เวลารวมในกระบวนการจึงรวดเร็วขึ้น ค่าเฉลี่ยของผลรวม OA เป็น 1.6806 และค่าเฉลี่ยของ RC เป็น 0.9835 ซึ่งมีค่าความเสี่ยงสูงขึ้นกว่าระบบ 2 หุ่นยนต์

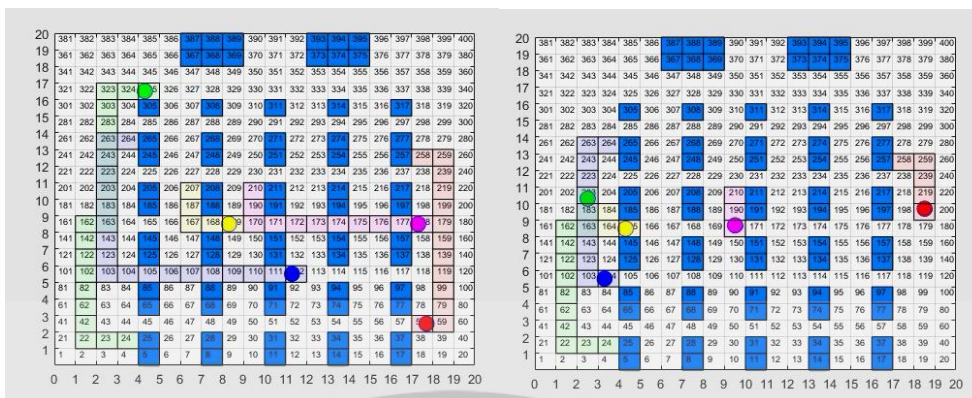
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 การหลบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 4 หุ่นยนต์

จากรูปที่ 4.18 แสดงการหลบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 4 หุ่นยนต์แล้วพบเหตุการณ์ที่มีเส้นทางซ้ำซ้อนกัน โดยหุ่นยนต์สี่เหลี่ยมหลบเส้นทางที่หุ่นยนต์สีแดงกำลังเคลื่อนที่และหุ่นยนต์สี่เหลี่ยมกำลังเคลื่อนที่ตามมา เวลาารวมที่ใช้ในระบบมีค่าลดลงเป็น 97.6985 วินาที เมื่อทำงานที่ 30 งานเท่ากัน หุ่นยนต์สี่เหลี่ยมทำงานทั้งหมด 7 งาน หุ่นยนต์สีแดงทำงานทั้งหมด 7 งาน หุ่นยนต์สีน้ำเงินทำงานทั้งหมด 9 งาน และหุ่นยนต์ตัวสุดท้าย หุ่นยนต์สี่เหลี่ยม ทำงานทั้งหมด 7 งาน ค่าเฉลี่ยของผลรวม OA เป็น 1.6807 และค่าเฉลี่ยของ RC เป็น 0.9996

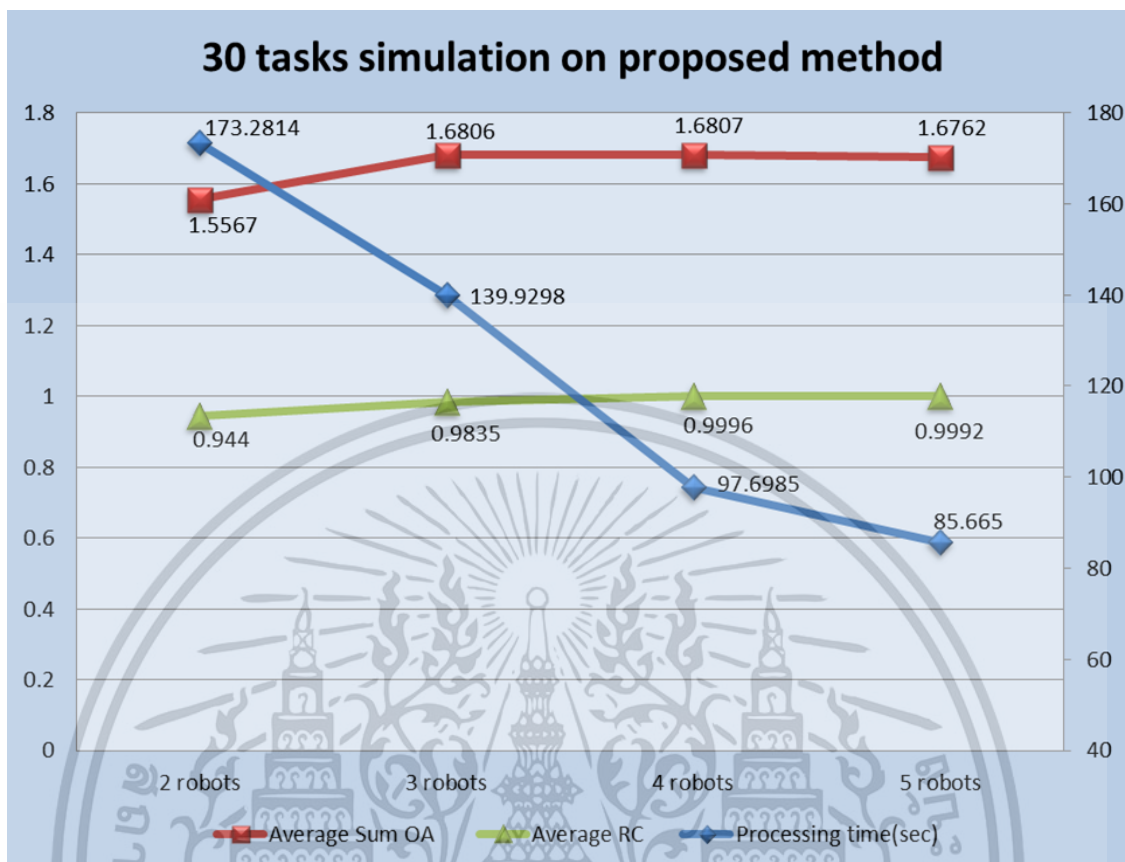
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 การหลบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 5 หุ่นยนต์

จากรูปที่ 4.19 แสดงการหลบการชนกันระหว่างหุ่นยนต์เมื่อทดสอบในระบบ 5 หุ่นยนต์แล้วพบเหตุการณ์ที่มีเส้นทางซ้ำซ้อนกัน โดยหุ่นยนต์สีเขียวพบเส้นทางที่ซ้ำซ้อนกับหุ่นยนต์สีน้ำเงิน หุ่นยนต์สีเขียวจึงทำการค้นหาเส้นทางใหม่เพื่อหลบการชนกันระหว่างหุ่นยนต์ ณ ตำแหน่งที่หุ่นยนต์สีฟ้าจะเคลื่อนที่มา เวลารวมที่ใช้ในระบบมีค่าลดลงเป็น 85.6650 วินาที เมื่อทำงานที่ 30 งานเท่ากัน หุ่นยนต์สีเขียวทำงานทั้งหมด 6 งาน หุ่นยนต์สีแดงทำงานทั้งหมด 6 งาน หุ่นยนต์สีน้ำเงินทำงานทั้งหมด 5 งาน หุ่นยนต์สีเหลืองทำงานทั้งหมด 7 งาน และหุ่นยนต์สีชมพูทำงานทั้งหมด 6 ค่าเฉลี่ยของผลรวม OA เป็น 1.6762 และค่าเฉลี่ยของ RC เป็น 0.9992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 กราฟเส้นแสดงผลการทดลองของการทดสอบ 30 งานด้วยวิธีการที่งานวิจัยนี้นำเสนอ

จากรูปที่ 4.20 แสดงกราฟเส้นของผลการทดลองของการทดสอบ 30 งานด้วยวิธีการที่งานวิจัยนี้นำเสนอซึ่งจำแนกได้ว่าเมื่อเพิ่มจำนวนหุ่นยนต์เข้าไปในระบบ เวลาที่ใช้ในกระบวนการจะมีค่าลดน้อยลงตามลำดับเพราะหุ่นยนต์มีการแบ่งงานกันมากขึ้น แต่การเพิ่มจำนวนหุ่นยนต์เข้าไปในระบบส่งผลให้ค่าความเสี่ยงสูงขึ้นซึ่งจะเห็นได้จากแนวโน้มของค่า OA ซึ่งเป็นความเสี่ยงในการชนสิ่งกีดขวางที่อยู่นิ่งและค่า RC ซึ่งเป็นความเสี่ยงในการชนสิ่งกีดขวางที่เคลื่อนที่ได้หรือหุ่นยนต์ตัวอื่นนั้น มีค่าความเสี่ยงที่สูงเพิ่มขึ้นเมื่อจำนวนหุ่นยนต์ในระบบเพิ่มขึ้น โดยหุ่นยนต์ที่มีความเสี่ยงน้อยที่สุดคือจำนวนหุ่นยนต์ที่เท่ากับ 2 หุ่นยนต์

บทที่ 5

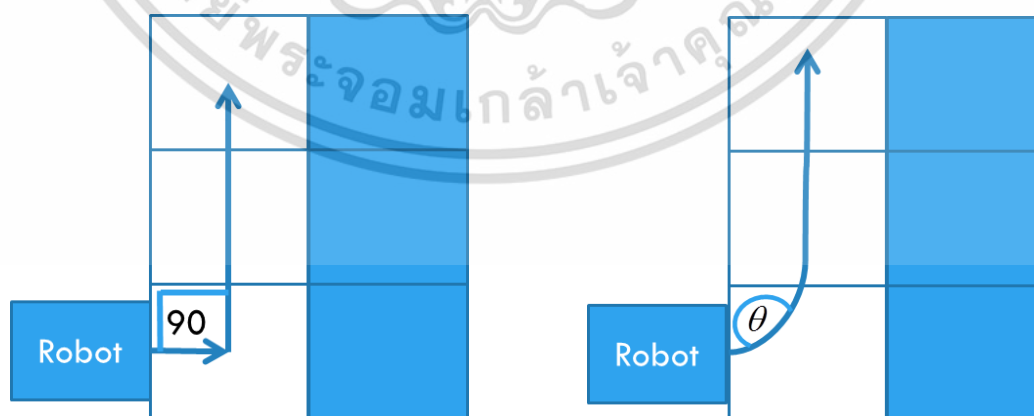
สรุปผลการทดลอง ข้อเสนอแนะ และแนวทางการพัฒนา

สรุปผลการทดลอง จากการทดลองพบว่าในการวางแผนเส้นทางที่ดีที่สุดในงานวิจัยที่ได้ศึกษามา นั้น จะเลือกใช้การวางแผนเส้นทางที่ดีที่สุดให้หุ่นยนต์จากการค้นหาเส้นทางที่สั้นที่สุด โดยมีการนำวิธีการค้นหาเส้นทางในแบบต่างๆมาพัฒนาต่อยอด ซึ่งงานวิจัยนี้มีแนวคิดที่แตกต่างออกไปคือการค้นหาเส้นทางที่ดีที่สุดจากเส้นทางที่มีความเสี่ยงในการชนที่น้อยที่สุดแล้วจึงค้นหาเส้นทางที่สั้นที่สุดตามลำดับ ทั้งนี้ เพราะเส้นทางที่ดีที่สุดภายในโรงงานอุตสาหกรรม จำเป็นต้องคำนึงถึงความปลอดภัยเป็นส่วนสำคัญ เพื่อให้การทำงานจริงในภาคอุตสาหกรรมนั้นมีโอกาสในการเกิดอุบัติเหตุหรือโอกาสในการเกิดความเสียหายในการผลิตที่น้อยที่สุด จากผลการทดลองในช่วงแรกจะพบได้ว่าการวางแผนเส้นทางที่ดีที่สุดโดยคำนึงถึงความปลอดภัยเป็นลำดับแรกและระยะทางที่สั้นที่สุดเป็นลำดับที่สองนั้น ในกรณีที่มีหุ่นยนต์เพียง 1 หุ่นยนต์ หุ่นยนต์สามารถที่จะเลือกเส้นทางที่ปลอดภัยที่สุดได้ผ่านการค้นหาค่าที่ดีที่สุดของค่า OA ซึ่งเป็นค่าฟังก์ชันที่งานวิจัยนี้สร้างขึ้นมาเพื่อค้นหาค่าความเสี่ยงของการชนสิ่งกีดขวางที่อยู่หนึ่งในแผนที่ จากผลของแบบจำลองใน 1 หุ่นยนต์พบว่าเส้นทางที่ปลอดภัยต่อการชนสิ่งกีดขวางที่อยู่หนึ่ง จะมีการใช้เวลาที่เพิ่มขึ้นไปจากการเคลื่อนที่จากการค้นหาเส้นทางที่สั้นที่สุด แต่ในทางกลับกัน เส้นทางที่งานวิจัยนี้นำเสนอให้ค่าความเสี่ยงต่อการชนสิ่งกีดขวางที่อยู่หนึ่งน้อยกว่า จึงนำวิธีการที่งานวิจัยนี้แนะนำเสนอบน 1 หุ่นยนต์ ไปพัฒนาต่อยอดในแบบจำลองที่มีมากกว่า 1 หุ่นยนต์ โดยจากการทบทวนวรรณกรรมพบว่าวิธีการค้นหาเส้นทางให้หุ่นยนต์ที่เป็นที่นิยมอย่างแพร่หลายคือวิธี A* และวิธีการค้นหาค่าที่ดีที่สุดที่เป็นวิธีการที่ได้รับความนิยมคือ Artificial bee colony (ABC) งานวิจัยนี้จึงเลือกที่จะนำสองวิธีการนี้มาประยุกต์ใช้ร่วมกัน โดยใช้ในการค้นหาเส้นทางของหุ่นยนต์ด้วยการปรับปรุงค่า OA ไปในการค้นหาของ A* เพื่อการหลบสิ่งกีดขวางแบบอยู่หนึ่ง และเพิ่มเงื่อนไขการค้นหาค่า RC ซึ่งเป็นฟังก์ชันใหม่ที่งานวิจัยนี้สร้างขึ้นมาเพิ่มเพื่อค้นหาค่าความเสี่ยงต่อการชนสิ่งกีดขวางที่เคลื่อนที่ได้หรือความเสี่ยงต่อการชนระหว่างหุ่นยนต์ในระบบที่มีหุ่นยนต์มากกว่า 1 หุ่นยนต์ภายใต้สภาพแวดล้อมต้นแบบจากโรงงานอุตสาหกรรม ซึ่งจากผลการทดลองพบว่าเมื่อหุ่นยนต์พบว่ามีความเสี่ยงในการชนหุ่นยนต์อีกตัวที่กำลังเคลื่อนที่อยู่ หุ่นยนต์จะถูกย้ายสถานะเข้าไปอยู่ใน Conflict loop เพื่อเข้าสู่การวางแผนเส้นทางใหม่ที่ไม่มีเส้นทางที่มีความเสี่ยงอยู่ในการคำนวณเส้นทางใหม่หรือเพื่อเข้าสู่สภาวะอยู่หนึ่งและรอการเคลื่อนที่ต่อไปเมื่อปลอดภัย ซึ่งจะเห็นได้ว่าผลการทดลองทั้งในระบบ 2 หุ่นยนต์ 3 หุ่นยนต์ 4 หุ่นยนต์ หรือ 5 หุ่นยนต์นั้น ถ้ามีระยะทางมากพอ หุ่นยนต์จะได้รับการคำนวณเส้นทางใหม่ที่ปลอดภัยกว่า เพื่อหลบการชนกันระหว่างหุ่นยนต์ แต่ด้วย

วิธีการข้างต้นนี้ เวลาที่ใช้อาจจะเพิ่มมากขึ้น นอกจากนี้ เพื่อให้แบบจำลองมีผลลัพธ์ที่ใกล้เคียงกับสภาพแวดล้อมจริงมากขึ้น งานวิจัยนี้จึงเพิ่มการออกแบบสิ่งรบกวนเข้าไปในระบบสำหรับการทดลองด้วย

ผลจากแบบจำลองในงานวิจัยนี้ แสดงให้เห็นว่าการใช้จำนวนหุ่นยนต์ที่เพิ่มขึ้น ระบบยังคงสามารถหาเส้นทางที่ปลอดภัยได้ทั้งการหลบสิ่งกีดขวางที่อยู่นิ่งจาก OA และการหลบการชนกันระหว่างหุ่นยนต์จาก RC ระบบ 2 หุ่นยนต์ให้ค่าความเสี่ยงในการชนสิ่งกีดขวางที่อยู่นิ่ง (OA) อยู่ที่ 1.5567 และค่าความเสี่ยงในการชนสิ่งกีดขวางที่เคลื่อนที่ได้ (RC) อยู่ที่ 0.9835 ซึ่งเป็นค่าความเสี่ยงที่น้อยที่สุดเมื่อเปรียบเทียบจากการทดลอง 30 งานภายใต้ 2 ถึง 5 หุ่นยนต์ และแบบ 5 หุ่นยนต์ใช้เวลาที่น้อยที่สุดเมื่อทำงาน 30 งานอยู่ที่ 85.6650 วินาที โดยไม่พบการชนกันระหว่างหุ่นยนต์เลย ซึ่งเมื่อเปรียบเทียบกับวิธี conventional จะเห็นได้ว่าวิธีการที่นำเสนอในงานวิจัยนี้ ให้เส้นทางที่ปลอดภัยที่สุดจากผลลัพธ์ของเส้นทางในรูปที่ 4.11 – 4.19 และค่าความเสี่ยงในการชนที่น้อยกว่าทั้งความเสี่ยงในการชนสิ่งกีดขวางที่อยู่นิ่งและค่าความเสี่ยงในการชนสิ่งกีดขวางที่เคลื่อนที่ได้ภายใต้สภาพแวดล้อมแบบมากกว่า 1 หุ่นยนต์ของแบบจำลองต้นแบบจากภาคอุตสาหกรรม

ข้อเสนอแนะและแนวทางในการพัฒนาต่อยอดของงานวิจัยนี้คือการลด turning angle เพราะผลจากงานวิจัยนี้ จะเห็นได้ว่า turning angle มีค่าเท่ากับ 90 องศาเสมอ ซึ่งอาจส่งผลกระทบต่อหุ่นยนต์ขณะเคลื่อนที่จริงได้ เพราะหุ่นยนต์อาจจะเคลื่อนที่ได้ไม่ stable ณ ขณะที่เกิดการหมุนด้วย turning angle สูง งานวิจัยนี้จึงมีข้อเสนอแนะว่า หากนำ grid map นี้ไปปรับปรุงให้เป็นแผนที่ที่มีความละเอียดมากขึ้น และเชื่อมโยงเส้นทางจุดต่อจุดด้วย spline interpolation จะทำให้เส้นทางมีความ smooth เพิ่มขึ้น และ turning angle ลดลง โดยเมื่อ turning angle ลดลง จะทำให้หุ่นยนต์เคลื่อนที่ขณะเลี้ยวได้ดีขึ้น



รูปที่ 4.21 turning angle เท่ากับ 90 องศา และ turning angle ที่น้อยกว่า 90 องศา

เอกสารอ้างอิง

- [1] อรรถสิทธิ์ แจ่มฟ้า. **อุตสาหกรรมเครื่องใช้ไฟฟ้าและอิเล็กทรอนิกส์**. พฤษภาคม 2563
- [2] Seagate. 2015. **Process training for engineer**. Samutprakarn: Seagate technology Thailand
- [3] Esther Cole. **Autonomous Robots Robot Path Planning**. Manfred Huber. 2008.
- [4] Wikipedia. **“Dijkstra's algorithm.”** [Online]. Available: https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm.
- [5] Wikipedia. **“A* search algorithm.”** [Online]. Available: https://en.wikipedia.org/wiki/A*_search_algorithm.
- [6] Wikipedia. **“D*.”** [Online]. Available: https://en.wikipedia.org/wiki/D*.
- [7] Wikipedia. **“Rapidly-exploring random tree.”** [Online]. Available: https://en.wikipedia.org/wiki/Rapidly-exploring_random_tree.
- [8] M. Janga R. and D. Nagesh K. “Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review.” **H2 Open Journal**, vol. 3, no.1,2020.
- [9] Pablo J. V. **Swarm Intelligence Metaheuristics, part 1: Ant Colony Optimization**. Stratio. 2018.
- [10] Daniel D., Scott K., and Eugene C. “Mobile Robot Path Planning With a Moving Goal.” **IEEE Access.**, vol.6, Jan. 2018. pp. 12800-12814.
- [11] Haojian Z., Yunkuan W., Jun Z., and Junzhi Y. “Path Planning of Industrial Robot Based on Improved RRT Algorithm in Complex Environments.” **IEEE Access.**, vol.6, Sep. 2018. pp. 53296-53306.
- [12] Xinyu L., Yun L., Jing Z., Jian Z. and Chunxi Y. “Self-Adaptive Dynamic Obstacle Avoidance and Path Planning for USV Under Complex Maritime Environment.” **IEEE Access.**, vol.7, Aug. 2019. pp. 114945-114954.
- [13] Xu T., Chen S., Wang D., Wu T., Xu Y., and Zhang W. “A Novel Path Planning Method for Articulated Road Roller Using Support Vector Machine and Longest Accessible Path With Course Correction.” **IEEE Access.**, vol.7, Dec. 2019. pp. 182784-182795.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [14] Jiankun W., Wenzheng C., Chenming L., Chaoqun W., and Max Q.-H. M. “Neural RRT*: Learning-Based Optimal Path Planning.” **IEEE Transactions on Automation Science and Engineering.**, vol.17, no. 4, Mar. 2020. pp. 1748-1758.
- [15] Min L., Xiaorong H., and Jing Y. “Surface Optimal Path Planning Using an Extended Dijkstra Algorithm.” **IEEE Access.**, vol.8, Aug. 2020. pp. 147827-147838.
- [16] Raouf F., Mohammed B., Tamer R., and Maamar B. “Enhancing Path Quality of Real-Time Path Planning Algorithms for Mobile Robots: A Sequential Linear Paths Approach.” **IEEE Access.**, vol.8, Aug. 2020. pp. 167090-167104.
- [17] Mao L., Ji X., and Qin F. 2018. “A Robot Obstacle Avoidance Method Based on Improved Genetic Algorithm.”in **2018 11th International Conference on Intelligent Computation Technology and Automation (ICICTA)**. China.
- [18] Giohong L.,and Yuanliang W. 2019. “Industrial Robot Optimal Time Trajectory Planning Based on Genetic Algorithm.”in **2019 IEEE International Conference on Mechatronics and Automation (ICMA)**. China.
- [19] Mohadese S-M.,and Hossein N-P. 2020. “Discrete Genetic Algorithm for Solving Task Allocation of Multi-robot Systems.”in **2020 4th Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)**. Iran.
- [20] Yanhui L., Zhonghua H., and Ya X. 2020. “Path planning of mobile robot based on improved genetic algorithm.”in **2020 3rd International Conference on Electron Device and Mechanical Engineering (ICEDME)**. China.
- [21] Tong Y., Zhong M., Li J., Li D., and Wang Y. “Research on Intelligent Welding Robot Path Optimization Based on GA and PSO Algorithms.” **IEEE Access.**, vol.6, Oct. 2018. pp. 65397-65404.
- [22] Syed Irfan Ali M., Moinul I., and Md. Mohiuddin U. 2018. “Optimal Path Planning Algorithm for Swarm of Robots Using Particle Swarm Optimization Technique.” in **2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)**. Indonesia.
- [23] Xiangjun K., Yunpeng G., Tianyi W., Jihong L., and Wenting X. 2019. “Multi-robot Task Allocation Strategy based on Particle Swarm Optimization and Greedy Algorithm.”

in **2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)**. China.

[24] Syed Irfan Ali M., Moinul I., and Md. Mohiuddin U. 2019. “Q-Learning Based Particle Swarm Optimization Algorithm for Optimal Path Planning of Swarm of Mobile Robots.” in **2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)**. Bangladesh.

[25] Changyun W., Ze J., and Boliang C. “Particle Swarm Optimization for Cooperative Multi-Robot Task Allocation: A Multi-Objective Approach.” **IEEE Robotics and Automation Letters.**, vol.5, no.2, Apr. 2020. pp. 2530-2537.

[26] Buhai S., and Jiaxiang X. 2020. “Time-Optimal Trajectory Planning of Industrial Robot based on Improved Particle Swarm Optimization Algorithm.” in **2020 39th Chinese Control Conference (CCC)**. China.

[27] Huangfu S., Tang S., Song B., Tong M. and Ji M. 2018. “Robot Path Planning Based on Improved Ant Colony Optimization.” in **2018 International Conference on Robots & Intelligent System (ICRIS)**. China.

[28] Xue Y., Wei-Neng C, Tianlong G., Huaqiang Y. , Huaxiang Z. and Jun Z. “ACO-A*: Ant Colony Optimization Plus A* for 3-D Traveling in Environments With Dense Obstacles.” **IEEE Transactions on Evolutionary Computation.**, vol.23, no.4, Aug. 2019. pp. 617-631.

[29] Xu L., and Zhengyan L. 2019. “Research on Improvement of Ant Colony Algorithm for Multi-Robot Task Allocation.” in **2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)**. China.

[30] Zhang Y., Zhnag Y., and Liu X. 2019. “Path Planning of Multiple Industrial Mobile Robots Based on Ant Colony Algorithm.” in **2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing**. China.

[31] Farouq Z., Saad H., and Ramdane M. “A Distributed Approach to the Multi-Robot Task Allocation Problem Using the Consensus-Based Bundle Algorithm and Ant Colony System.” **IEEE Access.**, vol.8, Feb. 2020. pp. 27479-27494.

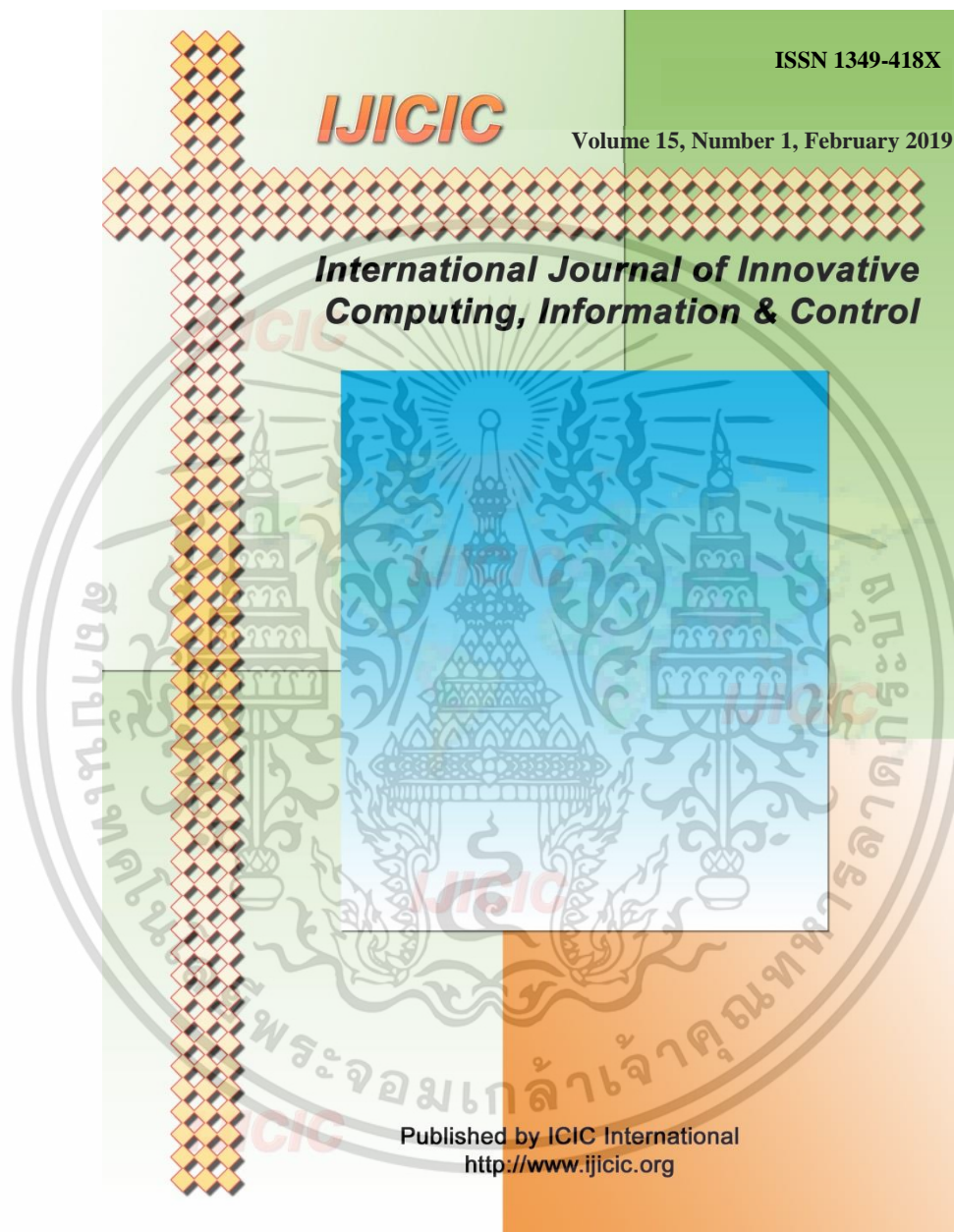
- [32] Xiangmin L., Yonghui H., Yijia Z. and Xiaojin Z. 2018. “Robot Path Planning Using Improved Artificial Bee Colony Algorithm.” in **2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)**. China.
- [33] Necmettin A., Yunis T., and Oguz K. 2018. “Probabilistic Roadmap and Artificial Bee Colony Algorithm Cooperation For Path Planning.” in **2018 International Conference on Artificial Intelligence and Data Processing (IDAP)**. Turkey.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SAFETY PATH PLANNING WITH OBSTACLE AVOIDANCE USING PARTICLE SWARM OPTIMIZATION FOR AGV IN MANUFACTURING LAYOUT

RAWINUN PRASERTTAWHEELAP¹, SOMYOT KAITWANIDVILAI¹
AND HISAYUKI AOYAMA²

¹Department of Electrical Engineering
Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Rd., Ladkrabang, Bangkok 10520, Thailand
rawinun.prasertaweelap@hotmail.com; drsomyotk@gmail.com

²Department of Mechanical and Intelligent Systems Engineering
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
aoyama@mce.uec.ac.jp

Received April 2018; revised September 2018

ABSTRACT. *In robotic systems, path planning is the one of important processes for robot motion. The best path planning is required for shortest path searching that can make fast movement of robot. However, the real environment is not only the path from point to point but it has obstacles which are the one of constraints for best path searching. The obstacle avoidance is concerned to avoid the crashing between robot and obstacle under environment. In Hard Disk Drive manufacturing, the first priority is safety constraint for non-collision and second priority is shortest path for processing time saving. This research designed the algorithm for path planning and obstacle avoidance for AGV in Hard Disk Drive Manufacturing of Seagate Technology (Thailand) Ltd by using particle swarm optimization. The fitness function on particle swarm optimization process for particle searching has been integrated with obstacle avoidance function to find the best path for robot without collision and total distance to find the shortest path. This algorithm is applied to verifying the model performance. The simulation results of this research are done by MATLAB 2016b and illustrate the good performance on different cases with controlled parameter.*

Keywords: Path planning, Particle swarm optimization, Obstacle avoidance, Hard disk drive, Safety constraint

1. Introduction. Currently, robotic system is the one of popular systems for manufacturing. Many factories apply the robotic system to increasing manufacturing performance such as unit per hour, accuracy, and reliability. One of robotic types is AGV which is the automated guided vehicle. Normally, it has been used for transferring process because it can move faster than people and it can work all the time without rest. In addition, the robotic system can be controlled and designed easier than people. There is the main reason why the robotic system is very popular in manufacturing. Hard Disk Drive factory is one manufacturing that applied the robotics for manufacturing process. Their purpose is to transfer the object from station to target point. Path planning is an important section to make robot movement in manufacturing layout. In addition, the obstacle avoidance is

DOI: 10.24507/ijicic.15.01.351

the key feature because it is related with safety requirement. Seagate Technology (Thailand) Ltd is the Hard Disk Drive manufacturing. This factory designs the robot system with path planning in manufacturing layout. Due to constraint on the total space, the width of robot pathway in layout is not a large width. The robot cannot move through robot pathway easier. In term of safety requirement in this factory, the minimum width of robot pathway must be more than 20% of robot's safety region radius. The robot can move along the path without crashing. When the crashing problem occurred, the transfer object (product) and robot can be damaged. It will impact the factory on rework cost and timeline commitment to customer. Safety path is the first priority for robot movement on this factory. This research focuses on algorithms for the best safety path of robot movement based on the layout on this factory. The layout of this factory is provided by industrial engineer in Seagate Technology (Thailand) Ltd. From Figure 1, it is the estimated layout of storage shelf. All of rectangles in Figure 1 are the storage shelf (obstacles) and AGV station. The small circle is the AGV. The comparing size of distance between the edge of obstacle (gap) are shown in Figure 1. The AGV station and AGV operation zone are created from model of factory floor. 2 AGV stations are the AGV parking zone before operation. It will be the start point of AGV in the model. AGV station 1 will be used to support AGV operation left zone (left dash line rectangle) and AGV station 2 will be used to support AGV operation right zone (right dash line rectangle). The concept of this design is to reduce the path of AGV to make high capability on process. The AGV size is shown in Figure 2. a is $a = \frac{6}{5}r$ and gap b is $b = \frac{9}{5}r$ where r is radius of AGV region.

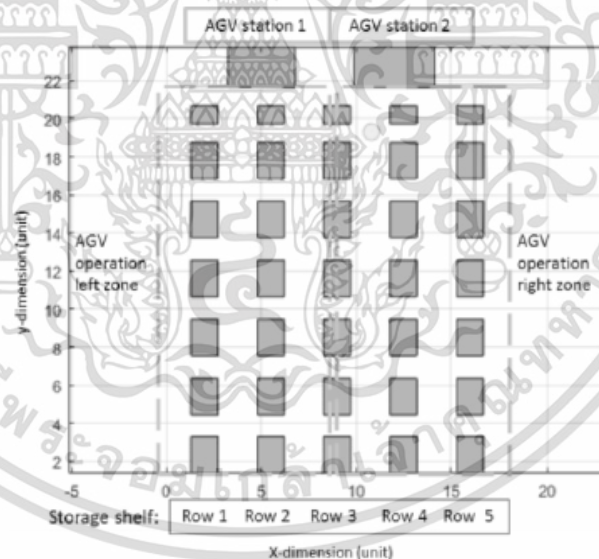


FIGURE 1. Estimated layout in production with AGV

Figure 3 shows the 2 wheels AGV inside the circle. Radius (r) is calculated from width and length of AGV by using Pythagoras theorem. The circle zone in Figure 3 represents the AGV region. This research used the simulation model as Figures 1-3 based on nonholonomic robot. The constraint on the model is the narrow gap (a) that is close to AGV region. In term of algorithm, many researchers studied the algorithm for robot movement. Probabilistic concept was applied in [1]. P. Alexandros et al. combined the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

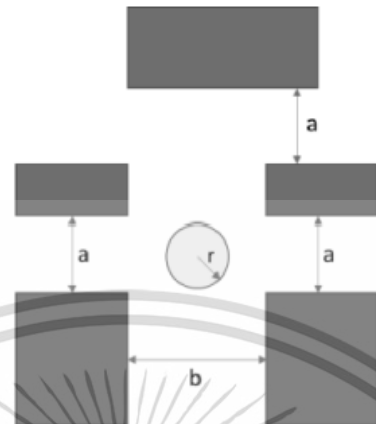


FIGURE 2. Distance between the edge of obstacle (gap) and AGV size



FIGURE 3. Circle and AGV

probabilistic movement primitives and Bayesian task prioritization but they still need to expand the evaluations on more complex real-world scenarios. In [2], G. Han et al. studied the probabilistic neighborhood location-point for data collection latency optimization and obstacle avoidance. The simulation shows better performance than the other existing underwater data collection algorithms. C. Evers and P. A. Naylor derive the framework in general for GEM-SLAM which is Probability Hypothesis Density (PHD)-based in SLAM algorithm. The simulation shows good on model-specific by using range-bearing sensors [3]. In [4], the obstacle avoidance is main objective by using Nonlinear Model Predictive Control (NMPC) in real time which represents good performance in various situations. A. K. Kar et al. presented the navigation techniques on 2 different environments which are normal and artificial potential function. Recalculated path is required for optimal path to avoid the obstacle which shows high risk to effect high processing time [5]. For motion path planning with additional tool, the aerial videography was selected in [6]. The purpose is to avoid the obstacle in real time but the constraint is the position and framing by user only. About the path planning algorithm, the algorithm for optimal path planning is popular because of the various environments. D. Devaurs et al. combined the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

principle of RRT and T-RRT. This method is faster than the RRT on complex problem [7]. A. Wan et al. presented a GPR-based (Gaussian Process Regression) prediction of deformation and method of compensation for robot motion accuracy improvement. They can achieve at high accuracy on both simulation and experimental prototype [8]. Furthermore, development phase for path planning by using integrated circuits was developed in [9]. Reconfigurable Analog Very Large Scale Integrated (AVLSI) circuits are Application Specific Integrated Circuits (ASICs) for path planning. The algorithm for mapping robot's environment needs to be loaded onto Field-Programmable Analog Arrays (FPAAs). They tested the proposed method on 24 environment scenarios. This is one of good studies about the path planning. P. Grosch and F. Thomas studied the geometric path planning for nonholonomic robots to find smooth paths but it is an open-loop method. The noise and errors on the systems still need to consider for next improvement [10]. One of techniques that many researchers selected for robotics system is Particle Swarm Optimization (PSO) but the fitness function is different because it is dependent on the design concept for each researcher. S. Yeasmin et al. proposed the point-to-point motion and planned the trajectory for robot arm using the Enhanced Particle Swarm Optimization (EPSO). The simulation results illustrate the optimal trajectory for different conditions of robot arm [11]. In [12], N. Setyawan et al. developed adaptive Gaussian parameter particle swarm optimization because of difficulties balancing on basic PSO algorithm. The simulation results show the better processing time with smooth path planning. In term of application, hybrid of particle swarm optimization with Tabu Search (TABU) were developed in [13] for mobile robot. The results show the best performance on hybrid PSO-TABU. In addition, PSO is widely used for multi optimal objectives. In [14], A. D. Falehi and M. Rafiee studied about the harmonic mitigation optimization and they applied the PSO on algorithm with multi-objective for low THD and harmonic elimination pulse width modulation. However, the results can represent good performance on desired situation only. In multiple robot systems, A. Ayari and S. Bouamama studied the advanced artificial intelligence. A new Dynamic Distributed Particle Swarm Optimization (D2PSO) is the main concept of their research. The results of proposed method perform better performance than normal [15]. L. Liao et al. improved the path planning on complex environment by Dynamic Double Mutation Particle Swarm Optimization (DDPSO) algorithm but it is not simulation results from various situations [16]. N. Mizuno and C. H. Nguyen used PSO to estimate parameters for high accuracy tracking control. Their concept is validated by several trajectories with good performance [17]. M. K. Rath and B. B. V. L. Deepak studied the path planning for mobile robot by using particle swarm optimization. The purpose is the obstacle avoidance with shortest path but the environment on simulation is not complex [18]. In [19], Z. Nie et al. combined the particle swarm optimization with annealing algorithm. The incremental work is required. Y. Guo et al. designed the path planning for robot for collision-free path by using fuzzy neural network on obstacle avoidance strategy and improved fuzzy parameters by particle swarm optimization. Verification phase shows the effectiveness from proposed method [20]. In addition, many researchers studied about the Spline for smooth path planning. Z. Wang et al. combined the Astar and B-Spline for path planning on autonomous underwater vehicle. Their purpose is to generate fit path based on the motion constraint. However, their concept works on low-dimension only. New method is required for high-dimension environments [21]. A. Khan et al. designed the coverage path planning for mobile robots by using rational quadratic Spline. They focus on smoothing of coverage path which is created by rational quadratic spline. The results show good on simulation only [22]. Y. Wang et al. used Spline interpolation to map 2D dubins path for 3D multi-vehicle path planning. They proposed to design the system to support multiple targets. Simulation

results show shortest dubins with small probability of collision [23]. D. Lee et al. designed the optimal path planning by using the Spline-RRT in 3-D environments of UAVs. Simulation results show good for their systems [24]. K. Yang et al. presented the optimal spline-based RRT path planning by using probabilistic map. RRT is good for quick path but it is not good for quality path. Spline-based RRT is their concept to get the feasible path. Simulation results illustrated feasibility for optimal path on system [25]. N. Arana-Daniel et al. designed smooth path planning by particle swarm optimization, radial basis function, Splines, and B'ezier curves. The results from their experiments show good on complex environments [26]. M. Neubauer and A. Müller designed the smooth path planning with Quaternions using B-Splines. They compared the angular velocities between standard method (SLERP) and smooth path generation. The appropriate method for continuity of motor torque on robotic manipulator is required. Smooth path by Splines can support their purpose [27]. S. Zhang et al. presented the smooth path planning using η^3 -splines for home service robot. Known maps with static obstacles are created. They used 2 steps in algorithm which are MAKLINK graph for shortest linear path and the smoothly connected path using η^3 -splines. Simulation and experiments results show good performance [28]. D. Lee and D. H. Shim used the Spline-RRT for optimal path planning of Fixed-Wing UAVs flights. Simulation results show that their concept can be utilized [29]. From many researches, the new algorithm for path planning optimization and obstacle avoidance are the important topics for robotics systems due to the various constraints such as robot specifications, robot performance, and environments.

In this research, the purpose is to design the new algorithm for path planning and obstacle avoidance to support AGV in Hard Disk Drive manufacturing. Normally, the main purpose to apply the robotics system in manufacturing is for replacing human workers because they can reduce high labor cost in operation. In addition, the static obstacles on factory are the normal condition that it will have more than one obstacle as station to support process operation in process. Non-collision system is main key of this research to avoid the accidents. This research is focused on the real layout of Hard Disk Drive in Seagate Technology (Thailand) Ltd, where AGV is working in factory operation. Moreover, the path planning searching is the one of main processes. W. Ojenge et al. studied about the Particle Swarm Optimization and Genetic Algorithm (GA) on mobile traffic jam times prediction. Their results indicated the better performance on PSO than GA in case of model prediction performance [30]. C.-C. Chiu et al. compared the PSO and GA in an urban area for the path loss reduction. They used GA and PSO for excitation voltage optimization on high order nonlinear optimization problem. The results show that the PSO is better than GA in term of performance in reduction of path loss [31]. V. Kachitvichyanukul studied the differences of Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution (DE). The results show that GA is appropriate for discrete but PSO and DE are appropriate for continuous optimization system. PSO shows better performance in terms of tendency for premature convergence and influence of best solution on population. In addition, the performances without local search to reach good solution and homogeneous sub-grouping convergence improvement are the one of good points from PSO also [32]. This main reason why this research selects the PSO is the good point and good performance from previous study on many researchers such as the performance to find the one of the best solutions and premature convergence trends. However, the path planning and obstacle avoidance algorithm cannot finish by using only the total distance along the path for PSO processing. It required the additional condition for obstacle avoidance. The fitness function inside PSO is modified on this research to support the shortest path without collision condition.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Particle Swarm Optimization. Particle Swarm Optimization (PSO) algorithm is introduced by Dr. Eberhart and Dr. Kennedy [1995] [15]. It is the one of optimization techniques which used nature bird convergence concept. The particle randomly moves through the space at each iteration. The objective is to search the best solution which can be defined by objective function. The properties of each particle are position (x_t) and velocity (v_t). Swarm is the set of particles. p_t (P_{Best}) is best fitness value so far and g_t (G_{Best}) is best value so far from population. The velocity and position are calculated by Equation (1) and (2) respectively.

$$v_{ij}^{t+1} = wv_{ij}^t + r_1^t c_1 (p_{ij}^t - x_{ij}^t) + r_2^t c_2 (g_{ij}^t - x_{ij}^t) \quad (1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (2)$$

where w is inertia, r_1 and r_2 are positive number in random, and c_1 and c_2 are acceleration coefficients. The inertia weight is decreasing function in linearity.

The PSO algorithm process flow is shown in Figure 4. The process starts from parameter initialization. The position, velocity, particle, swarm, dimensions, swarm size and number of iterations have been set in initial phase. The position number (r) has been run in random condition between 0 to 1. Then, the PSO process will start calculation on Equation (1) and Equation (2) until it meets criteria. The fitness function or objective function is calculated to check the best value before continuing to the next process. All of PSO process will be stopped when it reaches the stopping criteria.

The vector of PSO calculation on Figure 5 is related with Equation (1) and (2). The positive number (r) and acceleration coefficient (c) are the important parameters on PSO process because there are the multiplication factors on Equation (1). The weight of vector will be changed by these parameters. It means that the next particle position will be changed also.

3. Obstacle Avoidance with PSO. One of key processes of this research is the obstacle avoidance and the obstacle avoidance function is designed by this research. In addition, this research combined the PSO method and obstacle avoidance function to find the best solution. Equation (3) is an equation for distance calculation between AGV's position (x_r, y_r) and obstacle's position (x_o, y_o).

$$d = \sqrt{(x_r - x_o)^2 + (y_r - y_o)^2} \quad (3)$$

Figure 6 shows the safety region between AGV and obstacle from Equation (3) but it is calculated from the center point of AGV and obstacle only. This condition is enough for model calculation in point to point but it cannot represent the real condition of robot in the real environment.

The equation for real gap calculation is shown in Equation (4) where R is the radius of obstacle region and r is radius of AGV region from Pythagoras theorem.

$$d_r = d - R - r \quad (4)$$

where $R = \sqrt{[\frac{w_o}{2}]^2 + [\frac{l_o}{2}]^2}$, $r = \sqrt{[\frac{w_r}{2}]^2 + [\frac{l_r}{2}]^2}$, w_o is width of obstacle, l_o is length of obstacle, w_r is width of AGV, and l_r is length of AGV.

Obstacle Avoidance function (OA) is shown in Equation (5) where α is positive number in range (1, 2).

$$OA = \max \left(1 - \alpha * \frac{d}{R + r}, 0 \right) \quad (5)$$

From Table 1, the OA value is convergent to 1 on high risk for crashing condition. When the OA value is convergent to 0, it is the low risk for crashing condition. However, the best condition is the condition that OA value shows 0 but it is only normal condition

on empty area. The OA value on area with obstacle will converge to 0. On this research, the results from obstacle avoidance function will be taken into account for fitness function calculation on PSO.

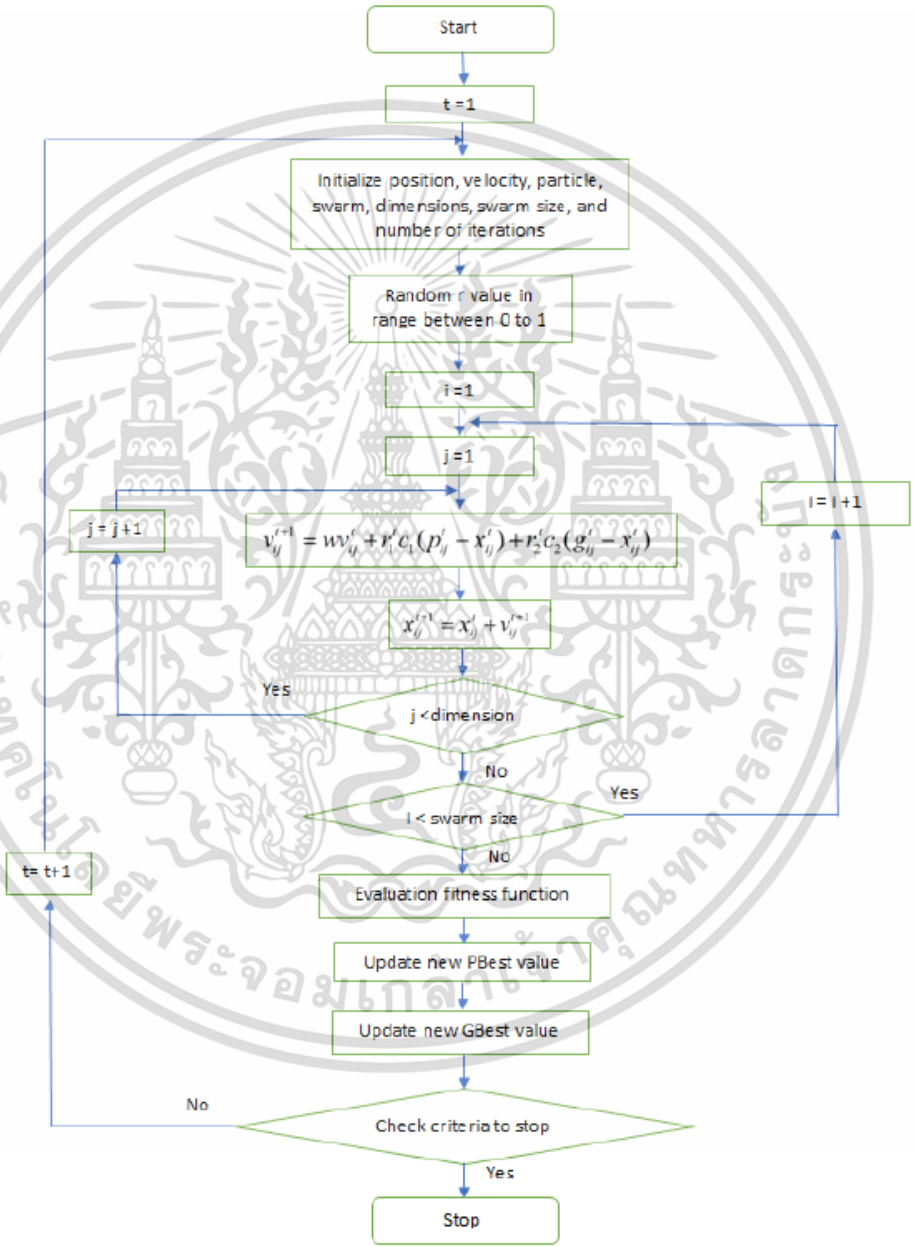


FIGURE 4. PSO algorithm process flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

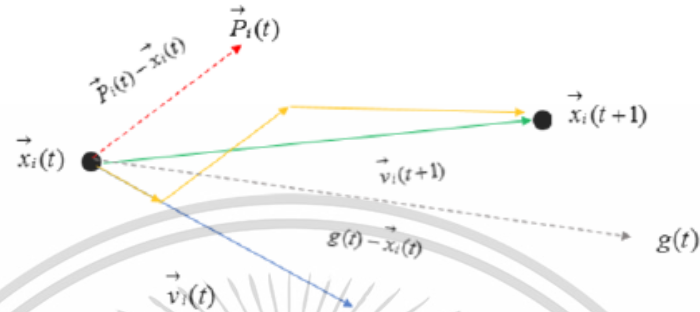


FIGURE 5. Vector of PSO calculation

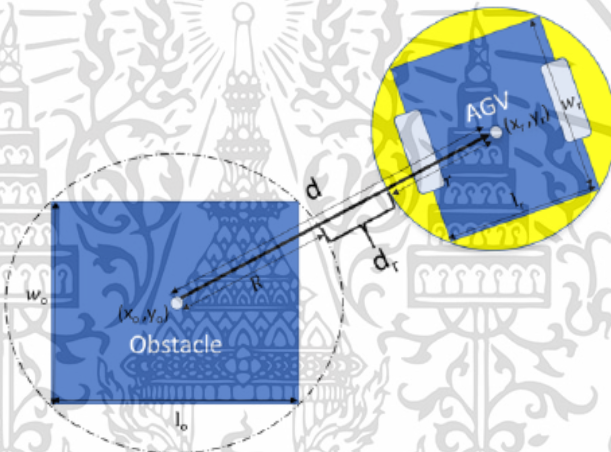


FIGURE 6. Safety region model

TABLE 1. Condition of obstacle avoidance function

Condition	OA value	AGV and Obstacle
$d < R + r$	$\rightarrow 1$	High risk for crashing
	$\rightarrow 0$	Low risk for crashing
$d = R + r$	0	Restriction zone
$d > R + r$	0	Safety zone

4. **Interpolation.** Interpolation is the new data points construction from the range of known data points. The data missing between points will be filled from interpolation by estimation.

4.1. **Linear interpolation.** The simplest form of interpolation is linear interpolation. The concept is to connect straight line with data points. The linear equation is shown in Equation (6).

$$y = f(x) = Ax + B \tag{6}$$

where A is slope, and B is y -intercept. The data in $[a, c]$ are estimated for straight line creation. From Equation (6), it can be substituted by a and c as Equations (7) and (8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

respectively.

$$f(a) = Aa + B \quad (7)$$

$$f(c) = Ac + B \quad (8)$$

After solving the problem, A and B are shown in Equations (9) and (10).

$$A = \frac{f(c) - f(a)}{c - a} \quad (9)$$

$$B = f(c) - \frac{f(c) - f(a)}{c - a}c \quad (10)$$

Substitute A and B into Equation (6) to find $f(b)$ in Equation (11) where $a \leq b \leq c$.

$$f(b) = \frac{f(c) - f(a)}{c - a}b + \left[f(c) - \frac{f(c) - f(a)}{c - a}c \right] \quad (11)$$

Linear interpolation needs at least 2 data points for calculation to find the coefficient of Equation (11).

4.2. Cubic-spline interpolation. For cubic-spline interpolation, it is a piecewise polynomial using not-a-knot end conditions. About the third order polynomial, the equation will be shown as Equation (12).

$$y = f(x) = \alpha_3x^3 + \alpha_2x^2 + \alpha_1x + \beta_1 \quad (12)$$

The coefficients for Equation (12) are computed on each interval by using the adjacent data points to make smooth line from point to point. However, the cubic-spline needs adjacent data points at least 4 points for calculation of 4 unknown coefficients ($\alpha_3, \alpha_2, \alpha_1, \beta_1$).

The graph of data points with interpolation between linear and cubic-spline is shown in Figure 7. The graph from cubic-spline will be smoother than linear due to polynomial term.

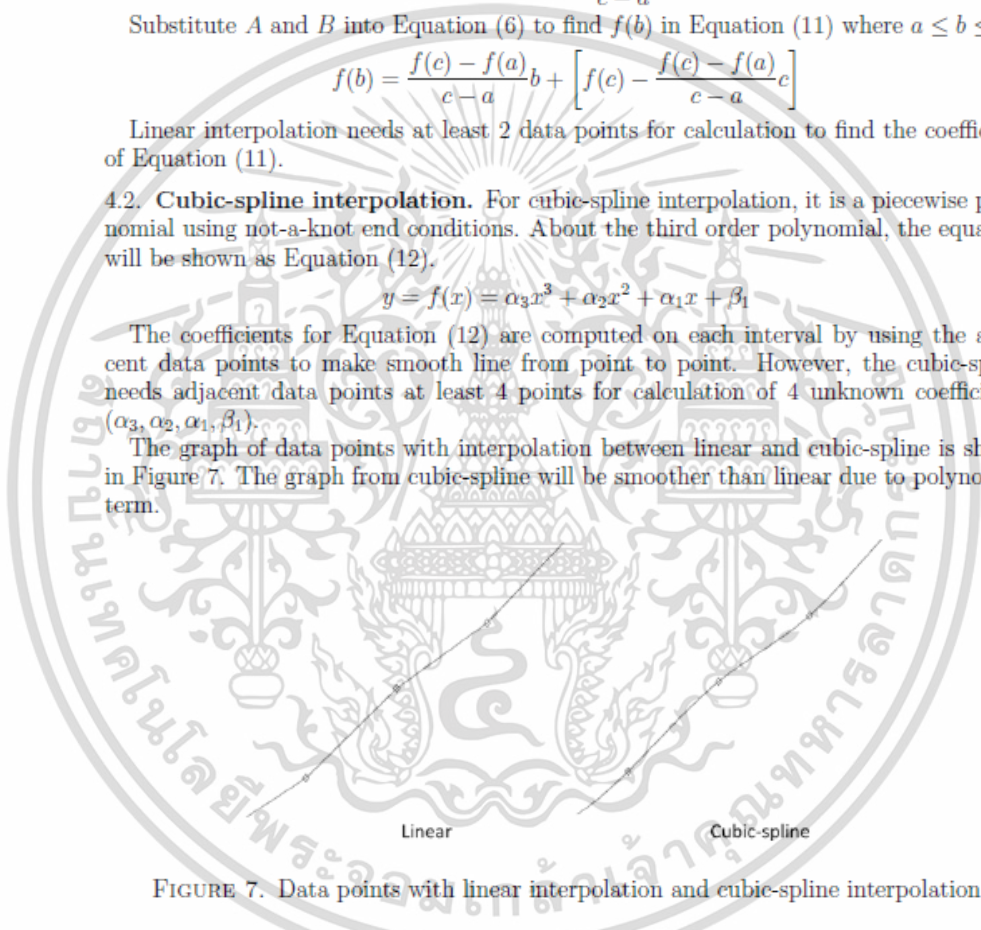


FIGURE 7. Data points with linear interpolation and cubic-spline interpolation

5. Proposed Method. In this research, the proposed method is the Particle Swarm Optimization for path planning with obstacle avoidance based on the layout from Seagate Technology (Thailand) Ltd (Hard Disk Drive manufacturing). The fitness function and obstacle avoidance are the new ones which are developed in this research. The fitness function inside PSO for path planning calculation is shown in Equation (13)

$$L = \sum_{t=1}^{n-1} \left[\sqrt{(x_t - x_{t+1})^2 + (y_t - y_{t+1})^2} \right] \quad (13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The fitness function on this research is the distance between start point to target point (goal) for path planning. Equation (13) represents the distance which is calculated sum of distance between point to point on the path from start point to target point. (x_i, y_i) is the position in 2D space. However, the results from Equation (13) can be used for path planning only. This research designed the fitness function for this system. New fitness function is shown in Equation (14) where OA is Equation (5), L is Equation (13), γ is coefficient for distance and β is coefficient for OA value (safety factor).

$$Z = \gamma * L * + \beta * OA \quad (14)$$

The inertia weight for PSO calculation will use value from the multiplication between weight and weight damping ratio as shown in Equation (15).

$$w_{t+1} = wdamp * w_t \quad (15)$$

The searching process will be started until it found minimum value of fitness function and OA is in minimum value in range (0,1). It means that the algorithm needs to find minimum value on both fitness function and OA function to conclude the best solution.

The process flow of proposed algorithm is shown in Figure 8. The algorithm will be stopped when it met stopping criteria. The minimum value of OA and fitness are concerned for each iteration loop to find the best solution.

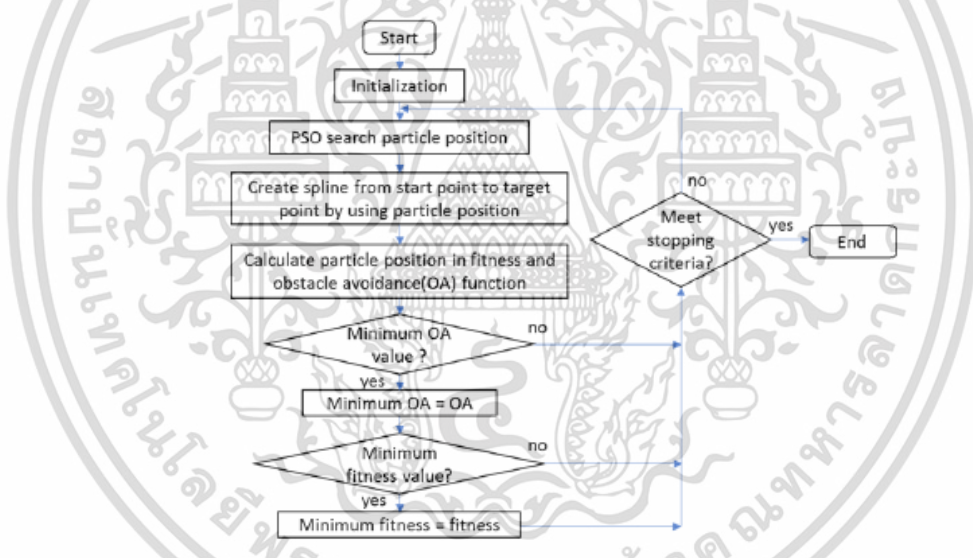


FIGURE 8. Process flow of proposed algorithm

6. Simulation Results. The simulation is done by MATLAB with 200 iterations because the value will be saturated after 100 iterations. More than 100 iterations are required to find the accurate value. The number of handling points is set to 30 with 200 population sizes (swarm size) to support $20 \times 22 \text{ unit}^2$ on model environments. The inertia weight is set to 1 with 0.99 inertia weight damping ratio as literature review. The acceleration coefficient is set to 1.4 because it represents the repeatability of fitness value after testing with model. In simulation, the start point and target point will be shown as Figure 9 which shows the path planning results on the model. This research simulates the results of path planning from PSO and obstacle avoidance function. However, the connecting path

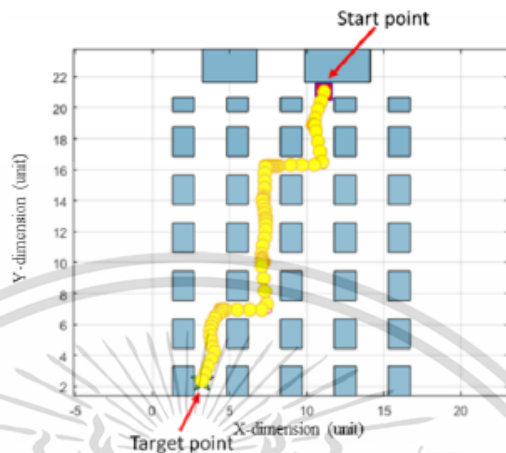


FIGURE 9. Path planning results with start and target point

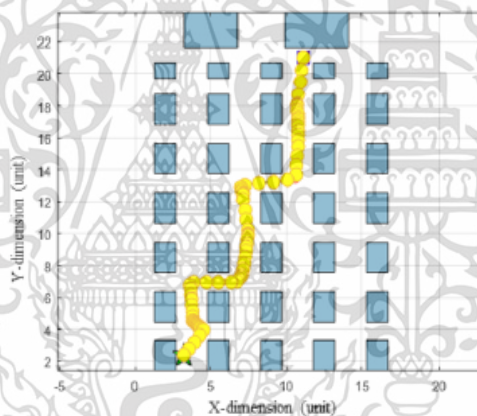


FIGURE 10. Path results on model with linear interpolation

is one of important sections to create path from point to point. The linear interpolation and cubic spline interpolation are simulated on the model of this research.

Figures 10 and 11 show the path results from proposed method on the simulation model by using linear interpolation and cubic-spline interpolation. Figure 12 shows the value of fitness function on each iteration. Fitness function values of spline (orange line) in Figure 12 are under the value of linear (blue line) after 43th iteration until 200th iteration. In Figure 13, the spline shows the obstacle avoidance function less than linear interpolation also.

The results from cubic-spline interpolation with proposed method are better than the results from linear interpolation in terms of minimum fitness value and minimum OA value. However, the total distance (L) from linear interpolation is 25.2801 units but the total distance from cubic-spline interpolation is 25.8663. It means that the total distance from linear interpolation is less than the total distance from cubic-spline interpolation.

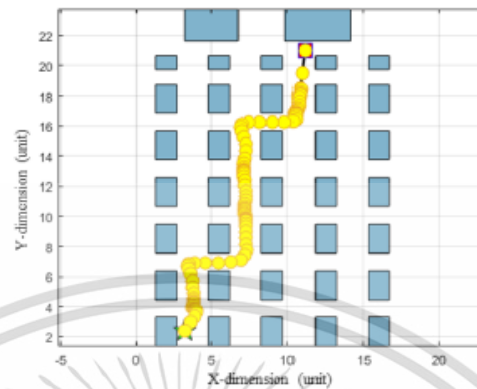


FIGURE 11. Path results on model with cubic-spline interpolation

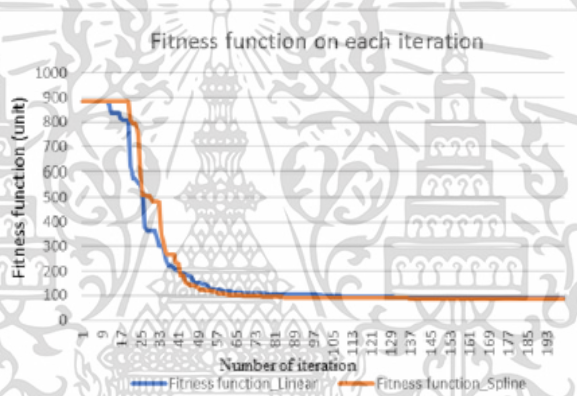


FIGURE 12. Fitness function on each iteration of linear interpolation and cubic-spline interpolation

The reason why the fitness value from cubic-spline is smaller than linear is the OA value from Equation (5). Cubic-spline can make curve path due to the third order polynomial calculation and curve path effect the high ratio of $\frac{d}{R+r}$. From Table 1, the OA value will converge to 0 on this case. Moreover, OA is in Equation (14) for fitness function calculation. Minimum OA value will affect minimum fitness function also.

In addition, the first path of cubic-spline in Figure 11 is still one problem because cubic-spline cannot generate the point along the first path (start point to first point). However, this result can affect the path results for AGV. The AGV has a risk to crash to the obstacle on this case because no data point is taken into fitness function calculation. The reason is that the cubic-spline concept needs to know data points at least 4 points to solve the coefficient of Equation (12). From this reason, the samples of data point need to increase to solve first phase problem from cubic-spline interpolation results. Total data points on interpolation process are changed from 100 to 200 for simulation. The new path results on model with cubic-spline interpolation at 200 total data points for interpolation process are shown in Figure 14 and it shows that the start point to first path can be

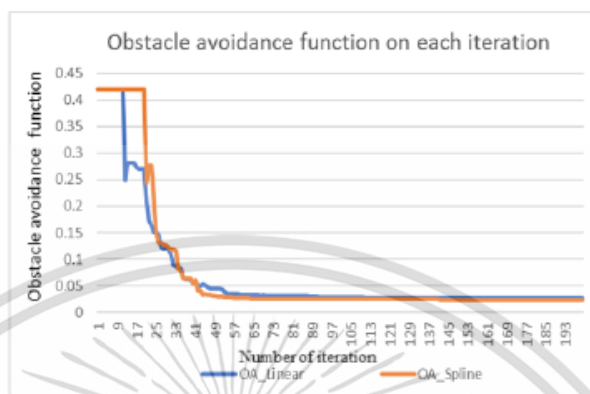


FIGURE 13. Obstacle avoidance function on each iteration of linear interpolation and cubic-spline interpolation

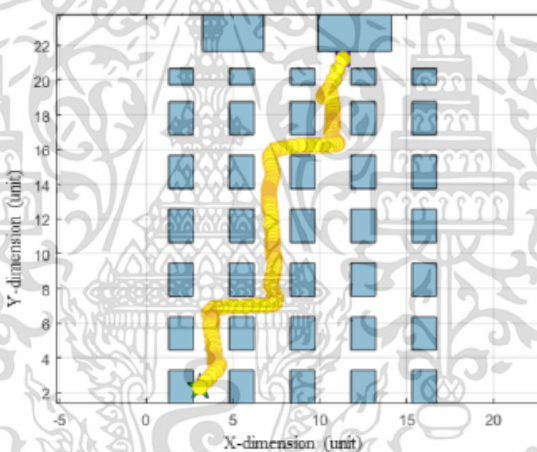


FIGURE 14. Path results on model with cubic-spline interpolation with 200 data points for interpolation

connected together after data points increasing. Then, the new data point results are compared in terms of fitness function, obstacle avoidance function and total distance as shown in Figures 15-17.

In Figure 15, the results of fitness function between linear and cubic-spline interpolation with 100 and 200 data points show the minimum fitness function on cubic-spline at 200 data points. In Figure 16, the results of obstacle avoidance show the same trend as fitness function due to the relationship on Equation (14).

In Figure 17, the total distance results show the opposite trend. The total distance results from linear interpolation are less than the cubic-spline interpolation and the increasing of data point can affect the high value of total distance on both interpolation method but the OA value is reduced after data point increasing. 200 data points are better than the previous in term of risk for crashing. This research selects cubic-spline interpolation with proposed method for next simulation.

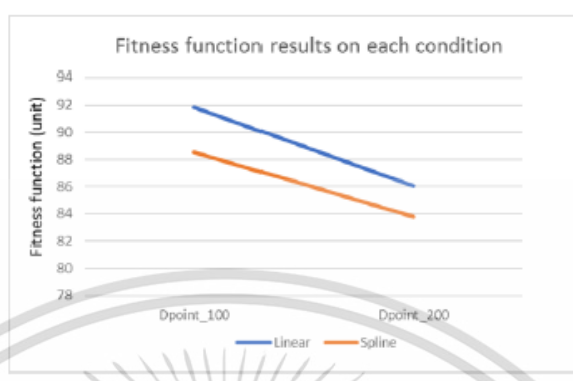


FIGURE 15. Fitness function results between linear and cubic-spline interpolation with 100 and 200 data points

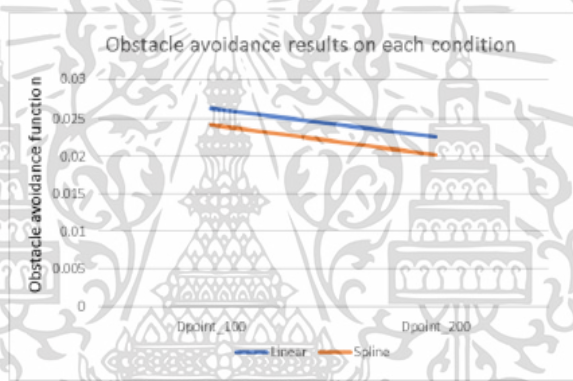


FIGURE 16. Obstacle avoidance function results between linear and cubic-spline interpolation with 100 and 200 data points

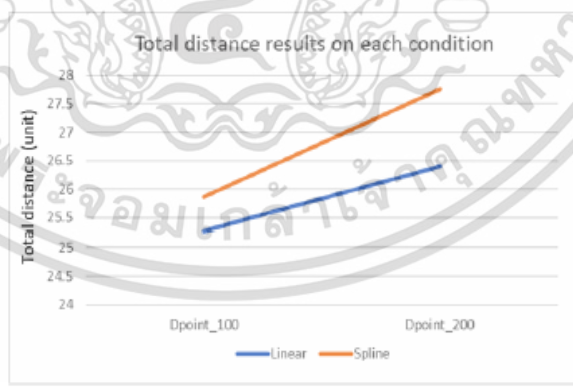


FIGURE 17. Total distance results between linear and cubic-spline interpolation with 100 and 200 data points

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In practical situation of manufacturing, the layout on manufacturing can be changed due to factory requirements. This research simulated the layout which is not straight line as previous layout. The layout changed condition is referred to layout plan of Seagate Technology (Thailand) Ltd. Figure 18 shows the path results between the shortest path and safety path on layout changed condition. The total distance on shortest path is 18.81 and safety path is 22.29. However, the OA value that it can represent the risk of collision shows 0.165 on shortest path and 0.007 on safety path. The result shows that the shortest path is better in term of minimum total distance but it shows high risk for collision. In case of safety path, it shows higher total distance but the risk of collision is very low and it is a good choice for factory because the safety is first priority.

On layout changed condition, this research creates new layout by adjusting the gap to verify the model performance. Then, the results have been compared between shortest path and safety path in Figure 19.

Figure 19 shows the right to left path results between shortest path and safety path on new layout. The shortest path shows high risk of collision and the safety path shows low risk for collision as same as the results from Figure 18. For numeric data, the total distance from shortest path is 21.87 and safety path is 25.51 but the OA value from shortest path is 0.2244 and safety path is only 0.0081.

The left to right path results between shortest path and safety path are shown in Figure 20. In Figure 20, the shortest path still shows high risk of collision on this path and the safety path shows low risk for collision as Figure 18 and Figure 19. The total distance of shortest path is 21.75 and safety path is 28.97. The OA value of shortest path is 0.2062 and safety path is only 0.0083. The results in Figures 18-20 illustrate the better performance for obstacle avoidance on safety path from proposed method of this research than the conventional shortest path searching.

7. Conclusions. This research considered path planning on particle swarm optimization with obstacle avoidance for AGV in Hard Disk Drive manufacturing. The main purpose is to develop the new algorithm for safety constraint to avoid the accident in manufacturing by using particle swarm optimization for low risk of collision path searching. The process step for model calculation starts at PSO and stops when it meets the stopping criteria. However, the goal for searching is combined between the obstacle avoidance function and total distance to find the minimum results. The minimum of fitness function can represent

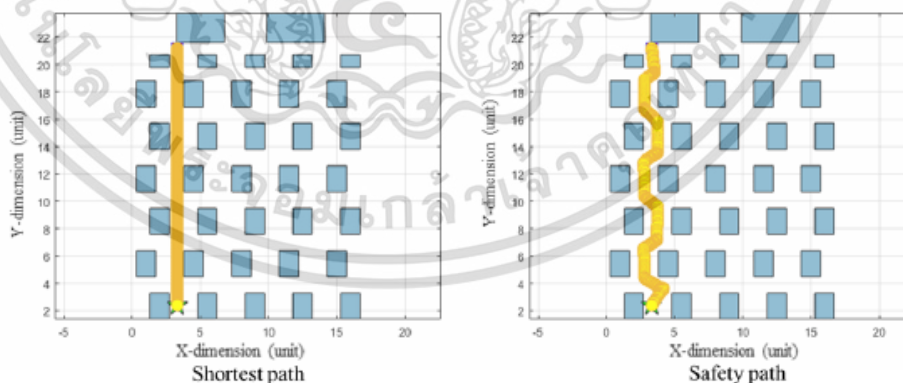


FIGURE 18. Path results between shortest path and safety path

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

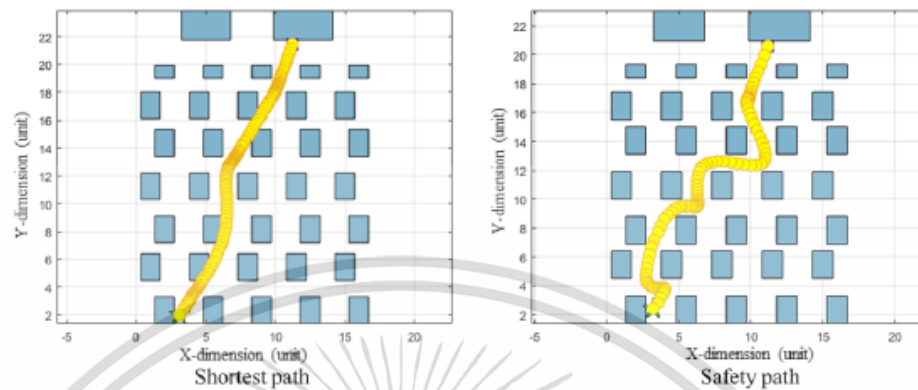


FIGURE 19. Right to left path results between shortest path and safety path on new layout

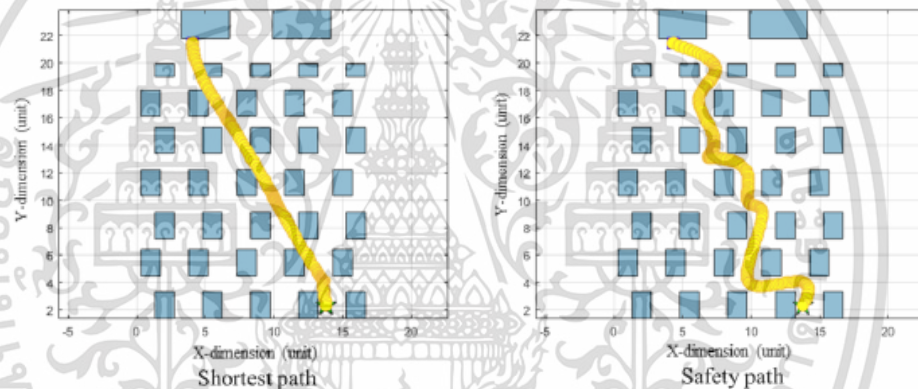


FIGURE 20. Left to right path results between shortest path and safety path on new layout

the low risk for collision and shortest path but the priority is set to non-collision. The minimum value of obstacle avoidance function shows the low risk for collision between AGV and obstacle. Inside the model, the PSO results will report the particle position on each point. The model is required interpolation on the system to connect path between point to point for robot path from start point to target point. 2 interpolations which are linear and cubic-spline are tested. The performance from cubic-spline interpolation is better than linear interpolation for obstacle avoidance on layout from Seagate Technology (Thailand) Ltd, and changeable layout plan. In addition, the results from proposed method of this research illustrate good performance on safety path when it is compared between the conventional shortest path searching and safety path searching on this research. This model can be applied for Hard Disk Drive manufacturing of Seagate Technology (Thailand) Ltd with safety constraint.

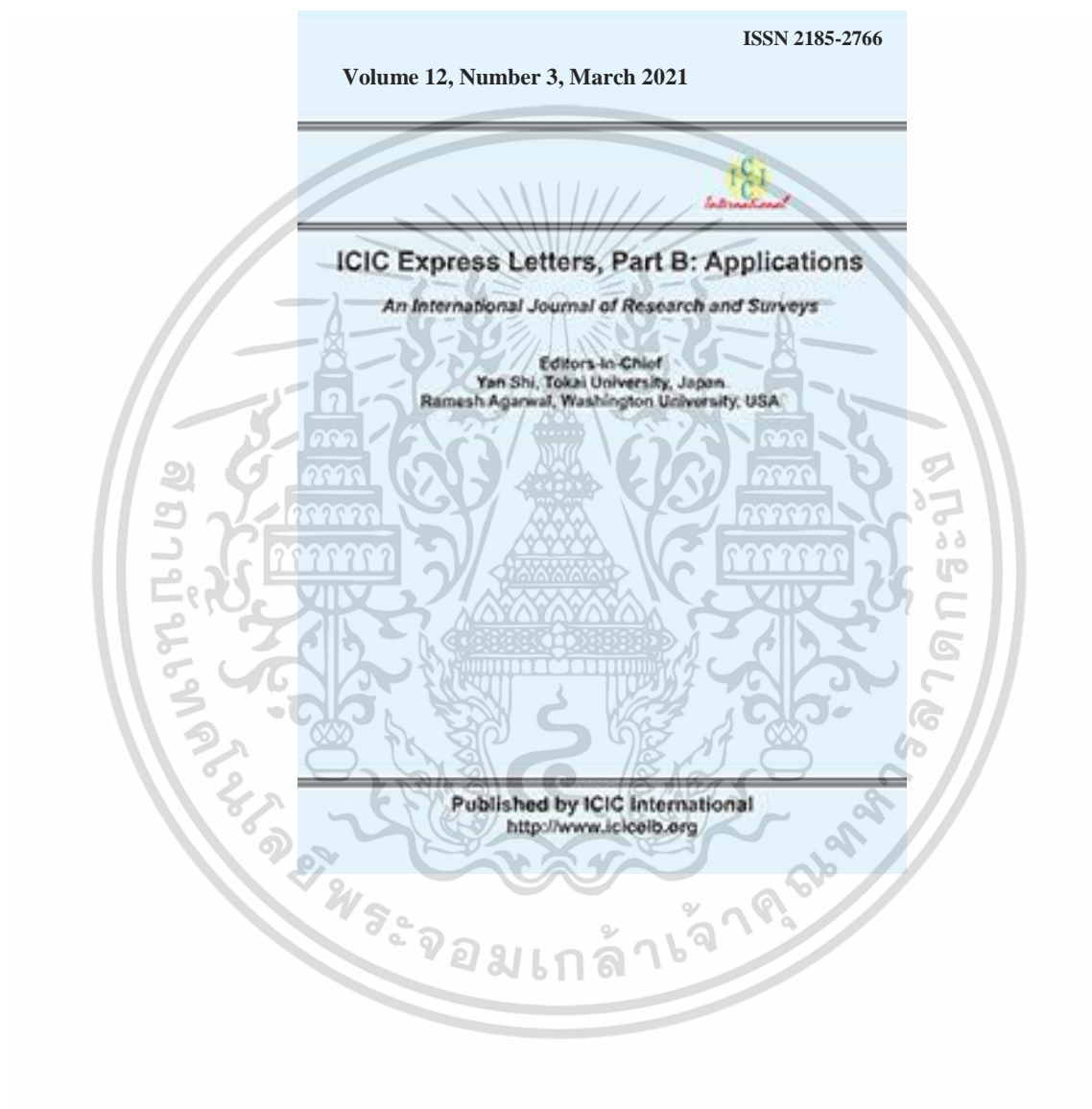
Acknowledgment. This work is supported by the Thailand Research Fund under the research grant No. PHD59I0059. This work is also supported by the Faculty of Engineering KMITL and Seagate Technology (Thailand) Company Ltd.

REFERENCES

- [1] P. Alexandros, L. Rudolf, P. Jan and N. Gerhard, Probabilistic prioritization of movement primitives, *IEEE Robotics and Automation Letters*, vol.2, no.4, pp.2294-2301, 2017.
- [2] G. Han, H. Wang, S. Li, J. Jiang and W. Zhang, Probabilistic neighborhood location-point covering set-based data collection algorithm with obstacle avoidance for three-dimensional underwater acoustic sensor networks, *IEEE Access*, vol.5, pp.24785-24796, 2017.
- [3] C. Evers and P. A. Naylor, Optimized self-localization for SLAM in dynamic scenes using probability hypothesis density filters, *IEEE Trans. Signal Processing*, vol.66, no.4, pp.863-878, 2018.
- [4] M. A. Abbas, R. Milman and J. M. Eklund, Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles, *Canadian Journal of Electrical and Computer Engineering*, vol.40, no.1, pp.12-22, 2017.
- [5] A. K. Kar, N. K. Dhar, S. S. F. Nawaz, R. Chandola and N. K. Verma, Automated guided vehicle navigation with obstacle avoidance in normal and guided environments, *The 11th International Conference on Industrial and Information Systems (ICIS)*, India, 2016.
- [6] T. Nageli, J. Alonso-Mora, A. Domahidi, D. Rus and O. Hilliges, Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization, *IEEE Robotics and Automation Letters*, vol.2, no.3, pp.1696-1703, 2017.
- [7] D. Devaurs, T. Siméon and J. Cortés, Optimal path planning in complex cost spaces with sampling-based algorithms, *IEEE Trans. Automation Science and Engineering*, vol.13, no.2, pp.415-424, 2016.
- [8] A. Wan, J. Xu, H. Chen, S. Zhang and K. Chen, Optimal path planning and control of assembly robots for hard-measuring easy-deformation assemblies, *IEEE/ASME Trans. Mechatronics*, vol.22, no.4, pp.1600-1609, 2017.
- [9] S. Koziol, R. Wunderlich, J. Hasler and M. Stilman, Single-objective path planning for autonomous robots using reconfigurable analog VLSI, *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol.47, no.7, pp.1301-1314, 2017.
- [10] P. Grosch and F. Thomas, Geometric path planning without maneuvers for nonholonomic parallel orienting robots, *IEEE Robotics and Automation Letters*, vol.1, no.2, pp.1066-1072, 2016.
- [11] S. Yeasmin, P. C. Shill and A. K. Paul, A new method for smooth trajectory planning for 3R robot arm using enhanced particle swarm optimization algorithm, *The 3rd International Conference on Electrical Information and Communication Technology (EICT)*, Bangladesh, 2017.
- [12] N. Setyawan, R. E. A. Kadir and A. Jazidie, Adaptive gaussian parameter particle swarm optimization and its implementation in mobile robot path planning, *International Seminar on Intelligent Technology and Its Application*, Indonesia, 2017.
- [13] M. R. Panda, R. Priyadarshini and S. K. Pradhan, Autonomous mobile robot path planning using hybridization of particle swarm optimization and tabu search, *IEEE International Conference on Computational Intelligence and Computing Research (ICIC)*, India, 2016.
- [14] A. D. Falehi and M. Rafiee, Novel optimal harmonic mitigation based on MOPSO to control switching of ODD-nary multilevel inverter, *International Journal of Innovative Computing, Information and Control*, vol.14, no.1, pp.243-260, 2018.
- [15] A. Ayari and S. Bouamama, Collision-free optimal paths for multiple robot systems using a new dynamic distributed particle swarm optimization algorithm, *Proc. of the 18th International Conference on Advanced Robotics (ICAR)*, China, 2017.
- [16] L. Liao, X. Cai, H. Huang and Y. Liu, Improved dynamic double mutation particle swarm optimization for mobile robot path planning, *Chinese Control and Decision Conference (CCDC)*, China, 2016.
- [17] N. Mizuno and C. H. Nguyen, Parameters identification of robot manipulator based on particle swarm optimization, *The 13th IEEE International Conference on Control & Automation (ICCA)*, Macedonia, 2017.
- [18] M. K. Rath and B. B. V. L. Deepak, PSO based system architecture for path planning of mobile robot in dynamic environment, *Proc. of Global Conference on Communication Technologies (GCCT 2015)*, India, 2015.
- [19] Z. Nie, X. Yang, S. Gao, Y. Zheng, J. Wang and Z. Wang, Research on autonomous moving robot path planning based on improved particle swarm optimization, *IEEE Congress on Evolutionary Computation (CEC)*, Canada, 2016.
- [20] Y. Guo, W. Wang and S. Wu, Research on robot path planning based on fuzzy neural network and particle swarm optimization, *The 29th Chinese Control and Decision Conference (CCDC)*, China, 2017.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [21] Z. Wang, X. Xiang, J. Yang and S. Yang, Composite Astar and B-spline algorithm for path planning of autonomous underwater vehicle, *IEEE the 7th International Conference on Underwater System Technology: Theory and Applications (USYS)*, Malaysia, 2017.
- [22] A. Khan, I. Noreen and Z. Habib, Coverage path planning of mobile robots using rational quadratic B'ezier spline, *International Conference on Frontiers of Information Technology*, Pakistan, 2016.
- [23] Y. Wang, W. Cai and Y. R. Zheng, Dubins curves for 3D multi-vehicle path planning using spline interpolation, *OCEANS*, Anchorage, USA, 2017.
- [24] D. Lee, H. Song and D. H. Shim, Optimal path planning based on spline-RRT* for fixed-wing UAVs operating in three-dimensional environments, *The 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, South Korea, 2014.
- [25] K. Yang, S. K. Gan, J. Huh and S. Joo, Optimal spline-based RRT path planning using probabilistic map, *The 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, South Korea, 2014.
- [26] N. Arana-Daniel, A. A. Gallegos, C. L'opez-Franco and A. Y. Alanis, Smooth global and local path planning for mobile robot using particle swarm optimization, radial basis functions splines and B'ezier curves, *IEEE Congress on Evolutionary Computation (CEC)*, China, 2014.
- [27] M. Neubauer and A. Müller, Smooth orientation path planning with quaternions using B-splines, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Germany, 2015.
- [28] S. Zhang, L. Sun, Z. Chen, X. Lu and J. Liu, Smooth path planning for a home service robot using η^3 -Splines, *IEEE International Conference on Robotics and Biomimetics*, Indonesia, 2014.
- [29] D. Lee and D. H. Shim, Spline-RRT* based optimal path planning of terrain following flights for fixed-wing UAVs, *The 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2014)*, Malaysia, 2014.
- [30] W. Ojenge, W. Okelo-Odongo and P. Ogao, Comparing PSO and GA optimizers in MLP to predict mobile traffic jam times, *International Journal of Computer Science and Information Security (IJCSIS)*, vol.13, no.10, pp.19-30, 2015.
- [31] C.-C. Chiu, Y.-T. Cheng and C.-W. Chang, Comparison of particle swarm optimization and genetic algorithm for the path loss reduction in an urban area, *Journal of Applied Science and Engineering*, vol.15, no.4, pp.371-380, 2012.
- [32] V. Kachitvichyanukul, Comparison of three evolutionary algorithms: GA, PSO, and DE, *Industrial Engineering & Management Systems*, vol.11, no.3, pp.215-223, 2012.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPTIMAL PATH PLANNING WITH A-STAR OPTIMIZATION AND MULTI-ROBOT TASK ALLOCATION IN HARD DISK DRIVE MANUFACTURING LAYOUT

RAWINUN PRASERTTAWHEELAP AND SOMYOT KIATWANIDVILAI

Department of Electrical Engineering
School of Engineering
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang District, Bangkok 10520, Thailand
rawinun.praserttaweeap@hotmail.com; drsomyotk@gmail.com

Received September 2020; accepted November 2020

ABSTRACT. *Hard Disk Drive Manufacturing is the manufacturing industry in electronics components. The automation processes for process capability improvement, process control improvement, and process stability improvement are important. A robotics system in this manufacturing is Multi-Robot System. The new development for Multi-Robot System depends on manufacturing requirements. This research is focused on the Optimal Path Planning for Multi-Robot Systems with Multi-Robot Task Allocation. A-star searching has been selected for path planning and modified for manufacturing requirements. The simulation model of this research has been developed by using Matlab R2016b as program based. Error built-in has been integrated in this simulation mode for validation. The simulation results for optimal path planning with modified A-star optimization and Multi-Robot Task Allocation can help to predict the suitable completion tasks/hour for Hard Disk Drive Manufacturing.*

Keywords: Optimal path planning, Multi-Robot System, A-star, Multi-Robot Task Allocation, Hard Disk Drive Manufacturing

1. Introduction. This research is studied about the optimal path planning for robotics systems based on Hard Disk Drive Manufacturing layout. Seagate Technology (Thailand) is the electronic industry in Hard Disk Drive Manufacturing which is required the development of Multi-Robot Systems support to an improvement on manufacturing process. Completion task per hour is the key requirement for manufacturing process because number of robot planning, tasks planning, and area control are very important. The major contributions and significance of this research is the optimal path planning with safety constraint on Multi-Robot Systems of Hard Disk Drive Manufacturing layout. Devaurs et al. studied about RRT method for path planning by RRT and T-RRT combination. The result is shown the good results on complex problem [1]. Wan et al. developed the GPR-based (Gaussian Process Regression) prediction for optimal path planning. They can perform high accuracy from both simulation and experiments [2]. Many researchers selected A-star for path planning and this method can represent good performance on variant problem. In [3], Votion and Cao studied new navigation algorithms on Multi-Robots by using A-star and the result is shown good performance on some examples. It should be better on systems reliability. For path planning on cooperative robot, key matrix is not only the path planning. The task allocation is also important for task arrangement of each robot. Huang et al. studied on Multi-Robot Task Allocation. They focused on the robot to target distance. In addition, they have checked model and their model can represent effectiveness [4]. In manufacturing, the key matrix for path planning is not only the shortest path. The obstacle avoidance for safety path is very important in

DOI: 10.24507/icicelb.12.03.223

manufacturing. In [5], new development of path planning has been developed for shortest path and safety path for manufacturing with good performance but the results are in 1 mobile robot only. For Multi-Robot Systems (MRS), the Multi-Robot Task Allocation (MRTA) is the one of the key matrices for systems. All tasks must be completed by robot tasks arrangement. Farouq et al. developed bundle algorithm with Ant Colony on Multi-Robot Task Allocation algorithm and it shows good performance results [6]. In [7], Chen et al. developed the 3-PRS ankle for force/position control by using the combination of position control and impedance control. Their purposed method can represent the effectiveness of flexible control. Solyman et al. studied about the robot task planning for pick-and-place task. Their purposed method shows the effectiveness for robot task planning in unstructured environments [8]. Yazeed studied the locomotion mechanism on small robot. He developed the hybrid locomotion mechanism with the highly effective results [9]. In [10], Watanuki et al. developed the deep reinforcement learning in multi-robot environment for obstacle avoidance based on the image. After learning, the performance looks better. Some researcher studied about the robot on human posture. In [11], the effect of exoskeleton robot has been studied during manual lifting task. The result can represent the exoskeleton robot can help users for the correct posture. From the literature review, many researchers developed the optimal path planning for the shortest path searching only. This research has developed the new modified A-star algorithm for safety path planning as a first priority and shortest path planning as a second priority. In addition, this research has developed this concept in the Multi-Robot Systems based on the Hard Disk Drive Manufacturing layout.

In this research, the manufacturing requirement is shown in the first section. The layout and manufacturing process were built in the model. Then, the path planning algorithm has been developed by the modified A-star algorithm. After that, the condition for Multi-Robot System needs to develop and add in the model. The output of this simulation is the completion tasks per hour between conventional method and purposed method with error built-in.

2. Layout. In the model of this research, layout and manufacturing requirements have been integrated for optimal path planning model creation. The concept of robotic path in manufacturing is to have the loading station for incoming part load, unloading station for outgoing part, and testing station for incoming part testing and outgoing part checking with data. Robots need to perform task to support the manufacturing process in layout as plan.

In Figure 1, 4 robots are moving in parallel and 3 boxes at the top of layout are the loading station and unloading station. The testing stations are 24 boxes in which it can perform both testing and checking condition in electrical testing of hard disk drive components. The material will be loaded from loading station and transferred from loading station to testing station by using robot and planning algorithm. After that, the next process is to load the tested material from the testing station to the unloading station. The path planning algorithm must perform each robot task for waiting time reduction. Then, the capability will be increased or we can get high completion task/hour.

3. A-Star Path Planning. One of popular algorithms for path planning is A-star which is the method to find the shortest path. Key concept is to calculate distance from starting point to any point and distance from any point to goal. Based on the concept, program can calculate the distance from starting point to any point but the distance from any point to goal needs to be estimated. The prediction model is required. The A-star uses the heuristic function for distance estimation from any point to goal to support starting point to goal point calculation. Equation (1) is the objective function of A-star algorithm. The minimum objective function will be selected for next robot path. $g(n)$ is the distance

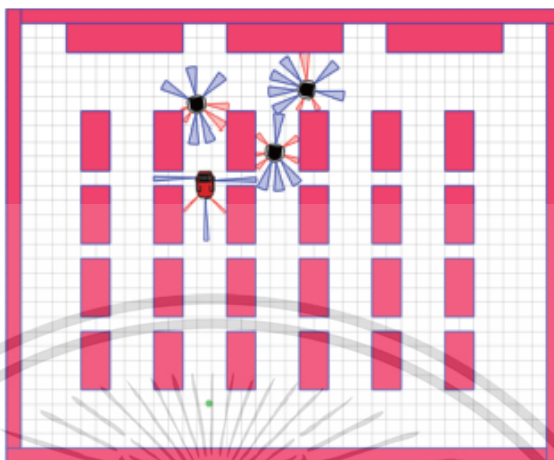


FIGURE 1. Hard Disk Drive Manufacturing layout and robots

from starting point to any node n . $h(n)$ is the estimated distance from any node n to goal. $f(n)$ is the sum of $g(n)$ and $h(n)$ to represent distance from starting point to goal based on the actual distance and estimated distance.

$$f(n) = g(n) + h(n) \quad (1)$$

For estimated distance, the heuristic function ($h(n)$) is the calculation of sum of difference of x and difference of y between two locations. For path planning, two locations are the location $n (x_n, y_n)$ and goal (x_g, y_g) where location n is any point and goal is goal point in Equation (2).

$$h(n) = |x_n - x_g| + |y_n - y_g| \quad (2)$$

4. Purposed Method for Optimal Path Planning. This research designed new A-star function by modified A-star function with safety path constraint. The new method is ASOA in Equation (3). The purpose is to find the optimal path planning for manufacturing layout and manufacturing constraint.

$$f(n) = g(n) + h(n) + OA(n) \quad (3)$$

From Equation (3), the objective function will be combined between the path planning for shortest path function and safety path function. The minimum value is required. Sum of these functions is required minimum value selection during searching to support optimal path planning also.

Obstacle avoidance function is OA function in [11]. From Equation (4), OA is obstacle avoidance function, α is safety factor, d is distance between robot and obstacle, R is radius of obstacle region and r is radius of robot region. If the OA value is close to 1, it will be the high risk condition of crashing between robots to any obstacle. If the OA value is close to 0, it will be the low risk of crashing or safety zone for robot path.

$$OA = \max \left(1 - \alpha * \frac{d}{R+r}, 0 \right) \quad (4)$$

For Multi-Robot Systems, the Multi-Robot Task Allocation or MRTA is the robot task assignment where R is set of robot and T is set of tasks.

In general problem, the single task is good enough for task allocation. The system will have robot with task in pairs as Equation (5). One robot will get one task for this case.

$$(r_1, t_1), (r_2, t_2), (r_3, t_3), \dots, (r_k, t_k) \quad (5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For complex tasks, the task allocation will be in group assignment as tasks of the bundles which are the sum of costs of individual tasks.

$$b_r(G) = \sum_{k=1}^f b_r(t_k) \quad \{t_k \in G\} \quad (6)$$

where G is the bundle, f is the number of tasks of bundle, b is the cost of tasks, and t is the tasks.

The robot is assigned to group assignment by the cost of individual tasks. This research selected the bundle task and designed new algorithm for optimal path planning on Multi-Robot Systems. The first step of this research is the reviewing process with the requirements of the Hard Disk Drive Manufacturing first which are the completion task/hour must be more than the conventional method with 0% robot crashing and robot can move within space constraint in specific layout from manufacturing plan. The A* has been selected for path planning because of effectiveness of A* method on many path planning research from literature review.

In Figure 2, the purposed method of this research is to start on whole task defining. Then, the model will start on distance calculation of each robot with all tasks. After that, it will select the minimum distance as 1st task for each robot and continue the A* path planning process. During the path planning process, it will have path conflict checking

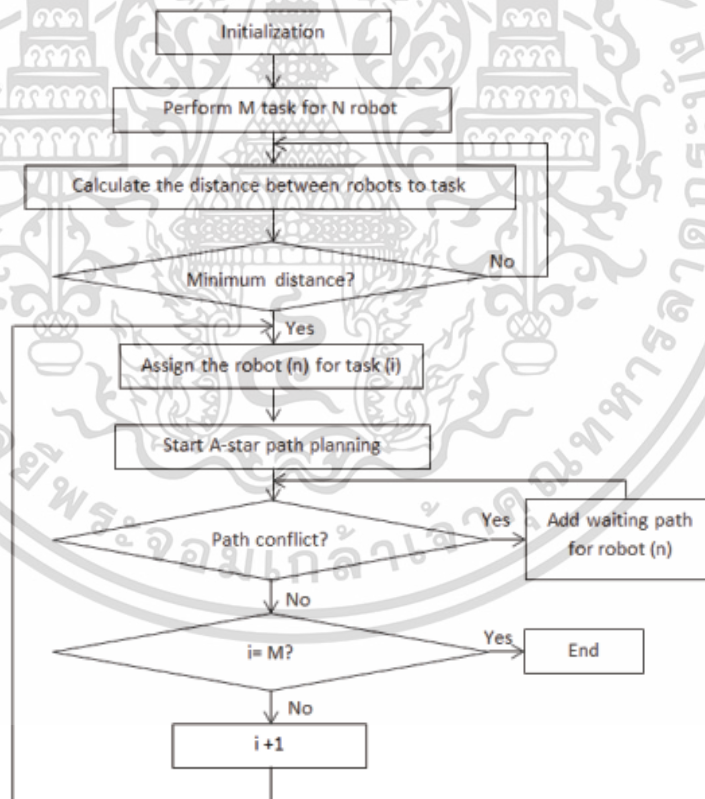


FIGURE 2. Purposed method of optimal path planning and Multi-Robot Task Allocation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

between robots due to cooperative robot working at the same time. If the path conflict = true, the waiting path process will start on that robot until conflict = false. Finally, the process will continue as this in continuous loop until all tasks have been completed.

Conditions for conflict path checking are

- Check conflict path between 2 robot paths;
- Current point of robot_a (x_i, y_i) is not equal to the next point of robot_b (x_{i+1}, y_{i+1});
- For not equal condition, continue the process as normal;
- For equal condition, add waiting time on the robot_b in 1 second and compare the robot point again in continuous loop.

5. Simulation Results. The simulation results show that the maximum robot for this model is 5 robots during maximum 30 tasks. If the robot number is more than 6, the simulation program will wait without stopping. The reason is the waiting path process on purposed method that it has limitation to support this model.

From Table 1, the longest processing time is 2 robots condition but the total step movement is smaller than 3, 4, 5 robots because robot can be managed the task in queue from 2 robots selection only. Shortest path will be the 1st priority when systems found only 2 robots. However, 5 robots can be shown the fastest processing time because program can share task for more robots but the total step movement may be increased to allow alternative robots movement path with safety constraint.

TABLE 1. Simulation results

No. of robots	Processing time (min)	Total step movement
2	2.49.02	1081
3	1.58.06	1129
4	1.43.19	1245
5	1.23.54	1216
6	Infinity	Cannot identify

During 6 robots test, program cannot continue until the end of process because of waiting path. All robots move in to waiting path loop at the same time and cannot find the solution by themselves.

In addition, this research has built the error on systems for checking model performance. 4 built-in error models are constant in Equation (7), proportional in Equation (8), and exponential in Equation (9) where f is objective function and y is the output after built-in error inside.

$$y = f + ae \quad (7)$$

$$y = f + b|f|e \quad (8)$$

$$y = f * \exp(ae) \quad (9)$$

From Table 2, no. of robots = 5 cannot be tested during random error built-in. Robot will stop at waiting path process as same as the results of no. of robots = 6 in Table 1. Maximum robot test on simulation is 4 robots condition.

TABLE 2. Simulation results with random error built-in at no. of robots = 4

Built-in error	Processing time (min)	Total step movement
Constant	1.50.28	1236
Proportional	1.50.38	1329
Exponential	2.01.15	1378

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

After built-in error in model, we can see the higher processing time and the higher trend of total step movement. The worst case a result is when error built-in is “exponential”. Processing time will be 2.01.15 minutes. It means that the completion tasks per hour will be around 895 tasks/hour for worst case of 4 robots condition.

The completion tasks/hour calculation is in Equation (10) where Ta = count of total tasks, t is time in minute and 60 is the constant value for time converting from minute to hour.

$$\text{Completion tasks/hour} = \frac{Ta}{t} \times 60 \quad (10)$$

From Figure 3, the completion tasks/hour is reduced after built-in error in models by using $Ta = 30$ tasks. The best completion tasks/hour is the model without built-in error that it can reach to 1258 completion tasks/hour but it might not be accurate for actual condition. The 2nd is model with constant error built-in that it can be 1200 completion tasks/hour. The 3rd is model with proportional error built-in and the 4th is the model with exponential error built-in. From these results, the worst case completion tasks/hour is 895 tasks/hour. It is higher than the conventional method which is equal to 800 tasks/hour.

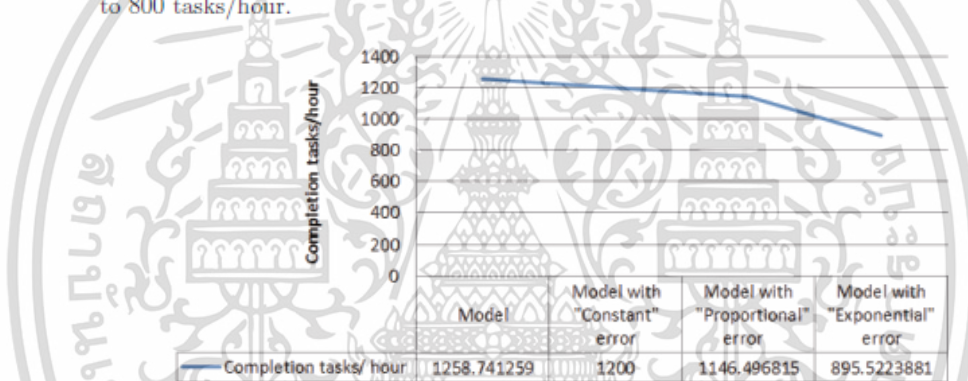


FIGURE 3. Completion tasks/hour during 4 robots running in the model

6. Conclusions. This research contributes new proposed method for optimal path planning based on Hard Disk Drive Manufacturing layout by using the combination of A-star path finding and safety path constraint with Multi-Robot Task Allocation. The conventional method by using Genetic Algorithm (GA) shows 800 tasks/hour. With the same condition, the proposed method results can represent 5 robots/30 tasks with 1.23.54 minutes which are the simulation results without error built-in. With 3 built-in errors in the model, it cannot run in 5 robots condition due to waiting time loop constraint. The best case for simulation test is 4 robots and worst case completion tasks/hour is 895 tasks/hour which is more than 800 tasks/hour. This result can help to support Hard Disk Drive Manufacturing for prediction and planning and this model can be applied for future development of the multi-robot path planning on manufacturing as well.

Acknowledgment. This research was supported by RRI and Seagate Technology (Thailand) Ltd. under Grant No. PHD59I0059.

REFERENCES

- [1] D. Devaurs, T. Siméon and J. Cortés, Optimal path planning in complex cost spaces with sampling based algorithms, *IEEE Trans. Automation Science and Engineering*, vol.13, no.2, pp.415-424, 2016.

- [2] A. Wan, J. Xu, H. Chen, S. Zhang and K. Chen, Optimal path planning and control of assembly robots for hard-measuring easy-deformation assemblies, *IEEE/ASME Trans. Mechatronics*, vol.22, no.4, pp.1600-1609, 2017.
- [3] J. Votion and Y. Cao, Diversity-based cooperative multi-vehicle path planning for risk management in costmap environments, *IEEE Trans. Industrial Electronics*, vol.66, 2019.
- [4] Y. Huang, Y. Zhang and H. Xiao, Multi-robot system task allocation mechanism for smart factory, *IEEE the 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, China, 2019.
- [5] R. Prasertaweelap, S. Kaitwanidvilai and H. Aoyama, Safety path planning with obstacle avoidance using particle swarm optimization for AGV in manufacturing layout, *International Journal of Innovative Computing, Information and Control*, vol.15, no.1, pp.351-368, 2019.
- [6] Z. Farouq, H. Saad and M. Ramdane, A distributed approach to the multi-robot task allocation problem using the consensus-based bundle algorithm and ant colony system, *IEEE Access*, vol.8, no.2, pp.27479-27494, 2020.
- [7] G. Chen, H. Zhou and P. Yang, Force/position control strategy of 3-PRS ankle rehabilitation robot, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp.481-494, 2020.
- [8] A. E. Solyman, K. M. Ibrahim, M. R. Atia, H. I. Saleh and M. R. Roman, Perceptive augmented reality-based interface for robot task planning and visualization, *International Journal of Innovative Computing, Information and Control*, vol.16, no.5, pp.1769-1785, 2020.
- [9] Y. Ghadi, A novel hybrid locomotion mechanism for small mobile robot, *ICIC Express Letters, Part B: Applications*, vol.10, no.11, pp.1031-1036, 2019.
- [10] R. Watanuki, T. Horiuchi and T. Aodai, Vision-based behavior acquisition by deep reinforcement learning in multi-robot environment, *ICIC Express Letters, Part B: Applications*, vol.11, no.3, pp.237-244, 2020.
- [11] M. Hussain, J. Park, N. Kim, H. K. Kim, J. Lee and J. Lee, Effects of exoskeleton robot on human posture and lumbar pressure during manual lifting tasks, *ICIC Express Letters, Part B: Applications*, vol.11, no.5, pp.439-445, 2020.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11th International Conference on Industrial Engineering and Operations Management
March 7 - 11, 2021, Singapore
A Virtual Conference

IEOM Society

SUSS
SINGAPORE UNIVERSITY OF SOCIAL SCIENCES

Operational Research Society of Singapore

Home Authors Program Registration Committee Competitions Keynote Global Engineering Education Industry4.0 GBME Panels Sponsors

Preliminary Program Proceedings

11th Annual International Conference on Industrial Engineering and Operations Management
Singapore, March 7-11, 2021

Scopus[®]
 Scopus Indexing of IEOM Papers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Optimal Path planning on Multi-Robot Task Allocation for Manufacturing with Artificial Bee Colony Algorithm

Rawinun Praserttaweelap and Somyot Kiatwanidvilai

School of Engineering

King Mongkut's Institute of Technology Ladkrabang

Bangkok, Thailand

rawinun.praserttaweelap@hotmail.com, drsomvotk@gmail.com

Abstract

This research presents the optimal path planning for manufacturing. The new contribution is the safety path searching by using artificial bee colony algorithm in multi-robot systems. The new RC function was introduced in this paper to represent the risk of collision path. The best safety path searching is the low risk of collision path from automatic searching on RC function. This research designs the artificial bee colony algorithm for the safety path searching based on the manufacturing environment. The proposed is to find the optimal safety path with fast convergence. With the built-in error, the simulation results illustrate the effectiveness of the optimal path planning in multi-robot systems for manufacturing.

Keywords

Safety path planning, Artificial Bee Colony Algorithm, Multi-Robot Systems, Manufacturing

1. Introduction

Nowadays, the multi-robot system in manufacturing is the key process for the capabilities and the reliabilities. In dynamic environments, the multi-robot movement is the complex systems. The new development for multi-robot path planning is required. Seagate Technology (Thailand) Ltd. is the one of the multi-robot manufacturing in the electronics industry. This research designs the optimal path planning in multi-robot systems based on the manufacturing layout of Seagate Technology (Thailand) Ltd. Many researchers studied the path planning with the different condition. An Wan et al. (2017) designed the GPR-based deformation compensation method for the assembly robots. Both simulation results and experimental prototype illustrate the good performance on their proposed method. Johnathan V. and Yongcan C. (2019) proposed the new path planning algorithm by using A*. The comparison results show the effectiveness of the modified A* (Johnathan and Yongcan 2019). Rong-Jong Wai et al. (2019) designed the optimal path planning and adaptive neural network for energy consumption prediction. This combination algorithm represents the efficiency UAV surveillance systems. Shuai D. H. et al. (2020) proposed new multi-robot path planning algorithm by using DDM. The results show good performance at high levels with the optimum quality. Jiankun Wang et al. (2020) created the neural RRT* for optimal path planning. The results show better performance than RRT in term of the convergence speed.

About the algorithm for path planning, Guozun Tian et al. (2018) improved the artificial bee colony algorithm for the faster convergence in Multi-UAV systems. The results represent the better efficiency. Xiangmin Li et al. (2018) improved the artificial bee colony algorithm for robot path planning. The results illustrate the faster optimal path searching. Necmettin et al. (2018) designed the optimal path planning by using artificial bee colony algorithm and probabilistic roadmap. The results show good performance in simulation only. Fateh B. and Beyza G. (2018) selected the artificial bee colony for robotic path planning and they compared the performance with the genetic programming. X. Bai et al. (2019) proposed the hybrid algorithm between the artificial bee colony algorithm and A* for Multi-UAV systems. The hybrid performance illustrate the better results than A* algorithm. Chengfang W. et

Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management Singapore, March 7-11, 2021

al.(2020) selected the artificial bee colony algorithm for the aerial vehicle path planning. Their simulation results represent the faster convergence than the conventional method. About the multiple vehicle research, Kiwon Y. (2020) developed the monitoring mobile sensors of multiple vehicles. He presented the new arrangement technique to support the particle matters in the urban street. His results represent the correlation of the particulate matters and metrological parameters.

From literature review, many researchers studied the robotic systems and designed the smart algorithm for complex problem solving. The path planning concept from previous research are the shortest path searching. The popular algorithm is the A* and artificial bee colony algorithm. The goodness of the artificial bee colony algorithm is the good effectiveness for fast convergence and this algorithm still performs the accuracy for optimal search.

This paper presents the new safety path searching for manufacturing in multi-robot environment. At first, this research designs the model from the manufacturing layout. The path planning for this research is A* path planning. The new contribution is the new risk of collision function (RC) for the safety path searching. However, the safety path searching concept can impact the higher processing time. From this problem, this research proposes the artificial bee colony for the fast convergence of optimal safety path searching. The results show a good prediction model for manufacturing.

2. Manufacturing

This research studied the manufacturing layout of Seagate Technology (Thailand) Ltd. Figure 1 shows the 3 rectangle boxes which are the loading station and unloading station. The 24 rectangle boxes are the operation test station. In automation process, the robotic system of this manufacturing is the multi-robot systems. The robotics path planning algorithm is very important for manufacturing capabilities with 0% collision.

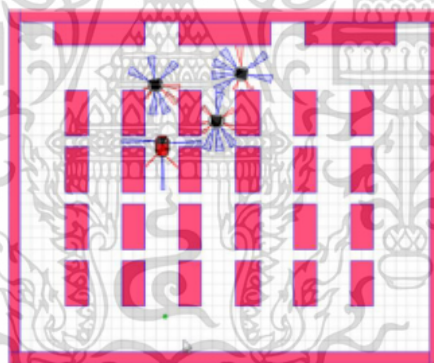


Figure 1. Manufacturing layout

3. A* Path planning

The A* path planning is the algorithm to search the shortest path. The distance between the starting point and the goal point $f(n)$ in (1) is the A* path planning function where $g(n)$ is the distance between starting point and current point and $h(n)$ is the estimated distance between current point and goal point.

$$f(n) = g(n) + h(n) \quad (1)$$

Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management Singapore, March 7-11, 2021

The distance calculation of $g(n)$ is the Euclidean distance in (2) where (x_s, y_s) is the starting point and (x_c, y_c) is the current point.

$$g(n) = \sqrt{(x_s - x_c)^2 + (y_s - y_c)^2} \quad (2)$$

At the current point, the distance between the current point to the goal point is the estimated distance in the heuristic value ($h(n)$). There are 3 types of heuristic calculations where (x_c, y_c) is the current point and (x_g, y_g) is the goal point.

3.1 Manhattan distance

Manhattan distance for $h(n)$ in (3) is the sum of the absolute value between the current point and the goal point.

$$h(n) = |x_c - x_g| + |y_c - y_g| \quad (3)$$

3.2 Diagonal distance

Diagonal distance for $h(n)$ in (4) is the max value of Manhattan distance.

$$h(n) = \max(|x_c - x_g|, |y_c - y_g|) \quad (4)$$

3.3 Euclidean distance

Euclidean distance for $h(n)$ in (5) is the distance calculation between the current point and the goal point.

$$h(n) = \sqrt{(x_c - x_g)^2 + (y_c - y_g)^2} \quad (5)$$

4. Artificial Bee Colony Algorithm

Artificial bee colony algorithm is an optimization algorithm based on the behavior of honey bee swarm. The bee colony has three groups of bees. The 1st group is the employed bees. The 2nd group is the onlookers and the 3rd group is the scouts. The unemployed bees are the onlookers and the scouts. Firstly, the employed bees start on the specific sources of food. Then, the onlooker bees observe the dance of employed bees within the hive. The dance of employed bees can represent the food source. For the scout bees, they search the random food source and they can find all food source locations.

In artificial bee colony algorithm, the food source position represents the solution quality. The algorithm process is as follows.

4.1 Initialization phase

Firstly, the initialization phase is to initiate the population of food source and population of scout bees with control parameters. The optimization process is to minimize the fitness function. The minimum fitness function represents the best food source. The population of food source is in (6) where l_i is the lower bound of X_{mi} , u_i is the upper bound of X_{mi} , and $rand(0,1)$ is random value between 0 to 1. The index m is the population of bees and i is an optimization variable.

$$X_m = l + rand(0,1)*(u, -l) \quad (6)$$

4.2 Employed bees phase

The employed bees search the high nectar in the neighbor food source. The neighbor food source in (7) is the new food source (V_m) where ϕ_m is random number within [-a, a] and X_m is a random food source.

$$V_m = X_m + \phi_m(X_m - X_m) \quad (7)$$

The fitness of the solution is shown in (8) and (9) where $f_m(X_m)$ is the objective function value of X_m .

$$fit_m(X_m) = \frac{1}{1 + f_m(X_m)}; \text{ if } f_m(X_m) \geq 0 \quad (8)$$

$$fit_m(X_m) = 1 + abs(f_m(X_m)); \text{ if } f_m(X_m) < 0 \quad (9)$$

4.3 Onlooker bees phase

The onlooker bees get the food source information from employed bees. Then, the onlooker bees choose their food sources based on the probability value of the fitness values from the employed bees. The probability value calculation is in (10).

$$P_m = \frac{fit_m(X_m)}{\sum_{m=1}^M fit_m(X_m)} \quad (10)$$

4.4 Scout bees phase

The scout bees choose the food source randomly. The employed bees become the scout bees when the solution is in hard limit criteria.

4.5 Final solution phase

The final solution phase is to select the best solution after the repeating process. The hard limit of repeating process is the maximum cycle number.

5. Proposed method

The proposed method of this research is to design the optimal path planning for manufacturing. Firstly, this research designs the manufacturing model as manufacturing layout from Seagate Technology (Thailand) Ltd. Based on the manufacturing requirements, the 0% collision is very important for manufacturing in multi-robot systems environment. This research proposes a new safety path by using the artificial bee colony algorithm for fast convergence. The lowest risk of collision path is the final solution of the proposed method. Nowadays, many researchers design the shortest path for the best path planning. The safety path planning with fast convergence is the new contribution of this research.

Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management Singapore, March 7-11, 2021

The risk of collision function or RC function is in (11) where γ is the safety factor, d_{ij} is the maximum distance of robot i (r_i) to each robot, and $\sum_{j=1}^n d_{ij}$ is sum of maximum distance of robot i (r_i) to each robot.

$$RC_{ij}(r_i) = \max(1 - \gamma(d_{ij}) / \sum_{j=1}^n d_{ij}, 0) \quad (11)$$

The minimum RC value is the low risk of collision path. This research applies the RC function in the artificial bee colony algorithm. The new fitness function in artificial bee colony algorithm of each robot i is shown in (12).

$$fit_{ij}(r_i) = \frac{1}{1 + RC_{ij}(r_i)}; \text{ if } RC_{ij}(r_i) \geq 0 \quad (12)$$

The best safety path is the minimum RC value or the maximum fitness function in artificial bee colony algorithm. The maximum fitness function represents the highest probability for onlooker bees.

The proposed method process is as follows.

1. Initialization process
2. Calculate the Euclidean distance of robot n and task t
3. Assign the task t for robot n from minimum distance
4. Start A* path planning searching
5. Start the safety path searching by using artificial bee colony algorithm
 - Case 1: Best safety path
 - Continue the next process
 - Case 2: Conflict path
 - Conflict loop process (waiting path)
6. Repeat the process until it reaches criteria

The end of process criteria is the completed tasks or out of the processing time limit.

6. Results

From the simulation results, the without RC searching shows the high risk of collision. With RC searching, the result shows the low risk of collision. Figure 2 shows the RC value of with and without RC searching from 100 iterations. The RC searching algorithm reduces the risk of collision value during the iteration process

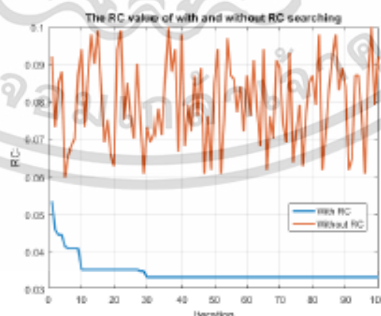


Figure 2. RC value of with and without RC searching

This research assigns 30 tasks for this simulation. The simulation conditions are 2 to 6 robots with a built-in error. Figure 3 is the location of 30 tasks.

In 6 robots condition, the model cannot complete the process. All robots stop because of the conflict loop in case 2 of process step 5.

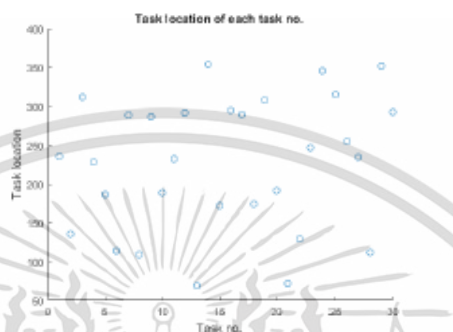


Figure 3. Task location of each task no.

For the model validation, this research built the error in the model on each condition. The built-in errors in this model are the constant, the proportional, and the exponential.

1. Built-in error: Constant

For the constant, the output of built-in error (y_{out}) in (13) is the sum of the desired function ($f_{desired}$) and the constant error value (c).

$$y_{out} = f_{desired} + c \quad (13)$$

2. Built-in error: Proportional

For the proportional, the output of built-in error (y_{out}) in (14) is the sum of the desired function ($f_{desired}$) and the multiplication of the desired function and the absolute of the constant error value (c).

$$y_{out} = f_{desired} + |c|f_{desired} \quad (14)$$

3. Built-in error: Exponential

For the exponential, the output of built-in error (y_{out}) in (15) is the multiplication of the desired function ($f_{desired}$) and the exponential function.

$$y_{out} = f_{desired} * \exp(ce) \quad (15)$$

Figure 4 is the result of 2 robots condition. Without error in the model shows the lowest moving step. The exponential built-in error shows the highest moving step. The lowest moving step is the without error.

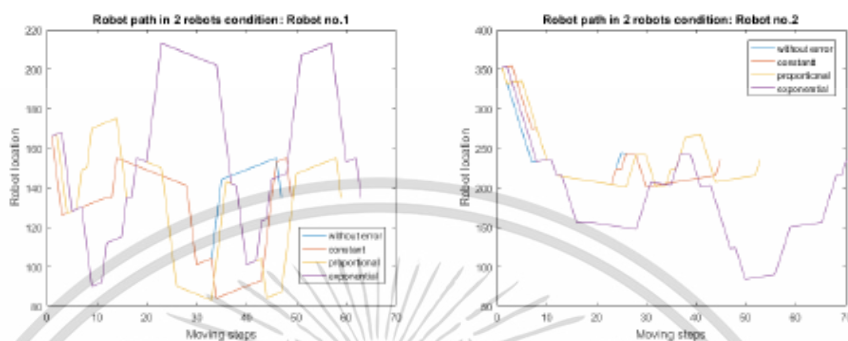


Figure 4. Robot path in 2 robots condition

With the built-in error, the moving step is higher than the without error. The reason is the built-in error in the heuristic value for the prediction variation simulation. The heuristic value in the A* path planning is the prediction of the robot path from the current point to the goal point. The next robot path can change along the time from the built-in error.

Figure 5 is the results of 3 robots condition. The last robot location for without error and built-in error show the same location. The robot steps show the over traveling in built-in error. The worst case is the exponential error. Without error is the best case. The maximum moving step is 74 steps in robot no.3 with an exponential error. The maximum processing time is 1.595712 minutes.

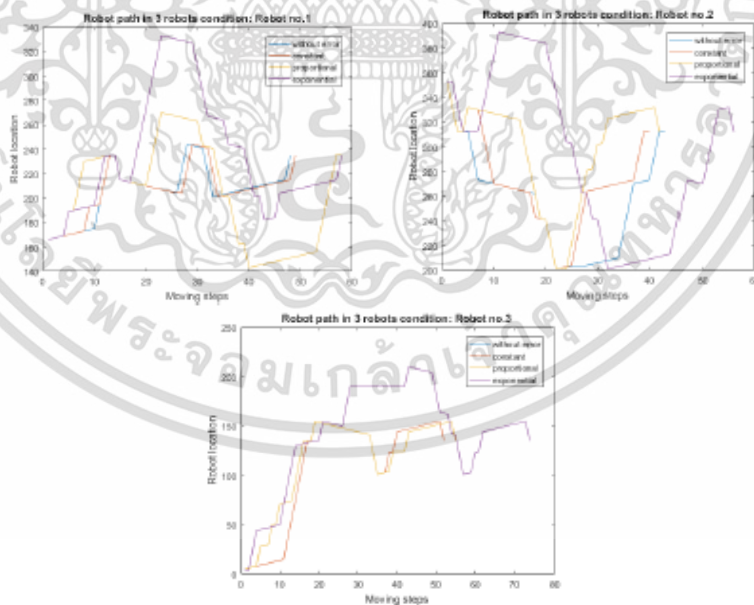


Figure 5. Robot path in 3 robots condition

Figure 6 is the results of 4 robots condition. The results are similar to 3 robots condition. The maximum moving step is 68 steps in robot no.4. The maximum processing time is 1.963558 minutes.

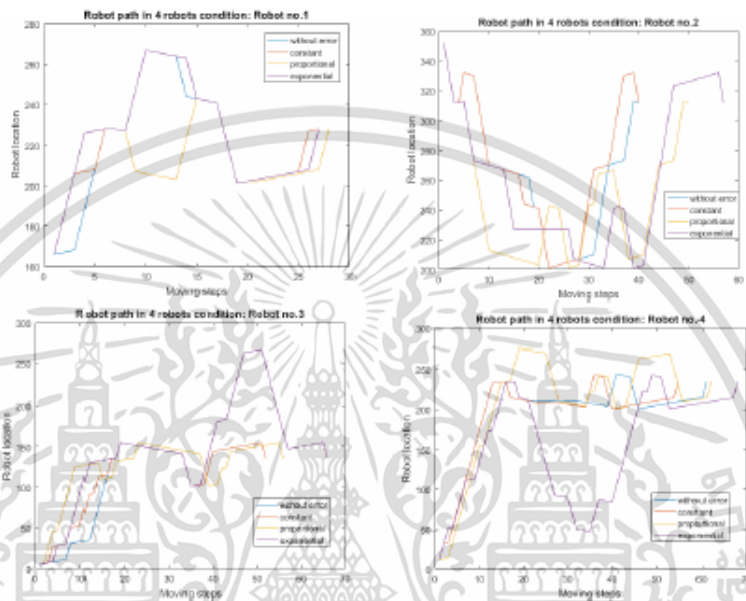


Figure 6. Robot path in 4 robots condition

In 5 robots condition, the result is shown in Figure 7. The results are similar to 3 robots condition and 4 robots condition. The maximum moving step is 57 steps in robot no.1. The maximum processing time is 2.645825 minutes. The increasing of robot no. in manufacturing layout reduces the moving step of each robot but the processing time is higher. The reason is the conflict loop process for safety path searching.

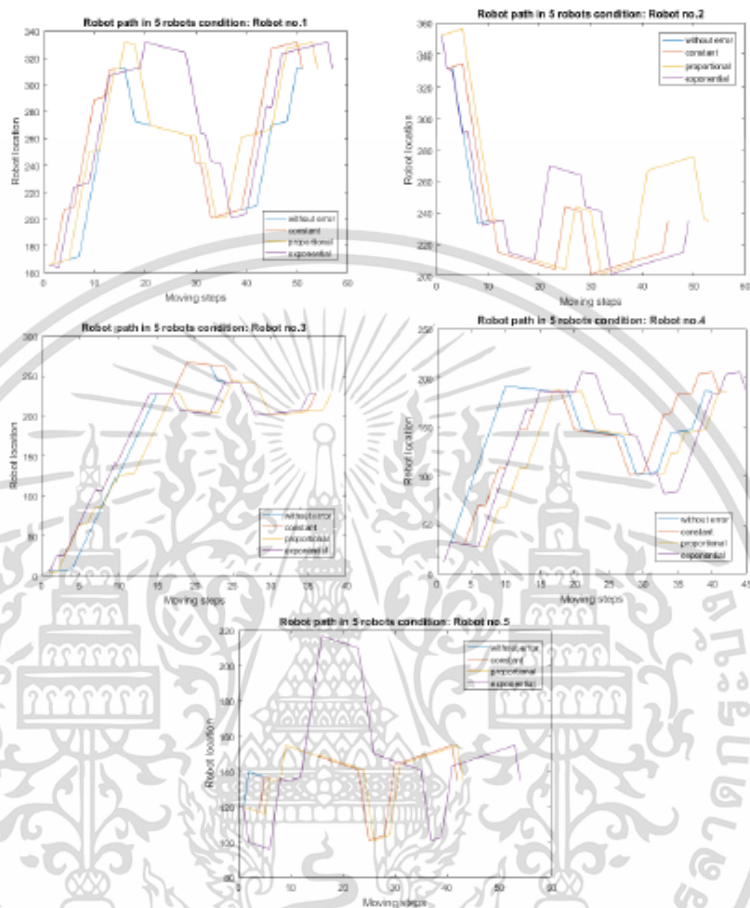


Figure 7. Robot path in 5 robots condition

Fig. 8 is the processing time in a minute of each condition. The minimum processing time is the 2 robots condition and the maximum processing time is the 5 robots condition.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

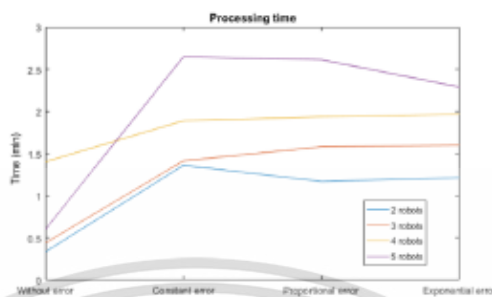


Figure 8. Processing time of each condition

Fig.9 is the maximum RC value of each condition. The minimum RC value is the 2 robots condition and the maximum RC value is the 5 robots condition.

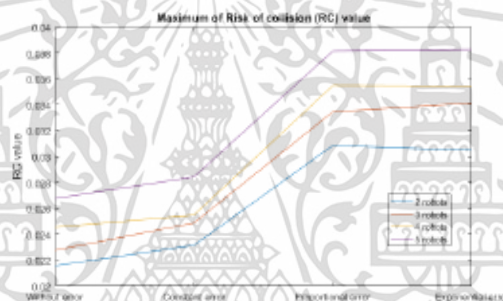


Figure 9. The maximum RC value of each condition

From the results, the best safety path searching is the 2 robots condition because of the lowest risk of collision and the fastest processing time. The optimal path planning in manufacturing is for 0% of collision with less impact on capability. The proposed method of this paper illustrates a good prediction model for manufacturing.

7. Conclusion

The proposed method of this research is to design the new optimal path planning for manufacturing. The environment in manufacturing is the multi-robot systems. The safety path is the 1st priority. The shortest path for fast processing time is the 2nd priority. This research presents the new safety path searching with fast convergence on the optimal value. The best condition is 2 robots condition for this manufacturing layout. The maximum risk of collision is only 0.0304789 and the maximum processing time is 1.358885 minutes. The results show the effectiveness of a prediction model for multi-robot systems in manufacturing.

Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management Singapore, March 7-11, 2021

References

- A. Wan, J. Xu, H. Chen, S. Zhang and K. Chen, Optimal path planning and control of assembly robots for hard-measuring easy-deformation assemblies, *IEEE/ASME Trans. Mechatronics*, vol.22, no.4, pp.1600-1609, 2017.
- J. Votion, and Y. Cao, Diversity-Based Cooperative Multivehicle Path Planning for Risk Management in Costmap Environments, *IEEE Transactions on Industrial Electronics*, vol. 66, no.8, pp.6117-6127, 2019.
- R-Jong Wai, and A. S. Prasefia, Adaptive Neural Network Control and Optimal Path Planning of UAV Surveillance System With Energy Consumption Prediction, *IEEE Access*, vol. 7, pp. 126137-126153, Aug 2019.
- S. D. Han, and J. Yu, DDM: Fast Near-Optimal Multi-Robot Path Planning Using Diversified-Path and Optimal Sub-Problem Solution Database Heuristics, *IEEE Robotics and Automation Letters*, vol.5, no.2, pp.1350-1357, 2020.
- J. Wang, W. Chi, C. Li, C. Wang, and M.Q.-H. Meng, Neural RRT*: Learning-Based Optimal Path Planning, *IEEE Transactions on Automation Science and Engineering*, vol.17, no.4, pp.1748-1758, Mar 2020.
- G. Tian, L. Zhang, X. Bai, and B. Wang, Real-time Dynamic Track Planning of Multi-UAV Formation Based on Improved Artificial Bee Colony Algorithm, *37th Chinese Control Conference (CCC)*, China, 2018, pp. 1934-1768.
- X. Li, Y. Huang, Y. Zhou, and X. Zhu, Robot Path Planning Using Improved Artificial Bee Colony Algorithm, *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, China, 2018.
- N. Alpkıray, Y. Torun, and O. Kaynar, Probabilistic Roadmap and Artificial Bee Colony Algorithm Cooperation For Path Planning, *International Conference on Artificial Intelligence and Data Processing (IDAP)*, Turkey, 2018.
- F. Boudardara, and B. Gorkemli, Application of Artificial Bee Colony Programming to Two Trails of the Artificial Ant Problem, *2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Turkey, 2018.
- X. Bai, P. Wang, Z. Wang, and L. Zhang, 3D Multi-UAV Collaboration Based on the Hybrid Algorithm of Artificial Bee Colony and A*, *Chinese Control Conference*, China, 2019.
- C. Wu, X. Huang, Y. Luo, and S. Leng, An Improved Fast Convergent Artificial Bee Colony Algorithm for Unmanned Aerial Vehicle Path Planning in Battlefield Environment, *IEEE 16th International Conference on Control & Automation (ICCA)*, Singapore, 2020.
- K. Yeom, Real-Time Particle Matters Monitoring Based on Mobile Sensors from Multiple Vehicles in Urban Street, Daejeon-City, South Korea, *International Journal of Electrical and Electronics Engineering & Telecommunications*, vol.9, no. 1, 2020.

Acknowledgements

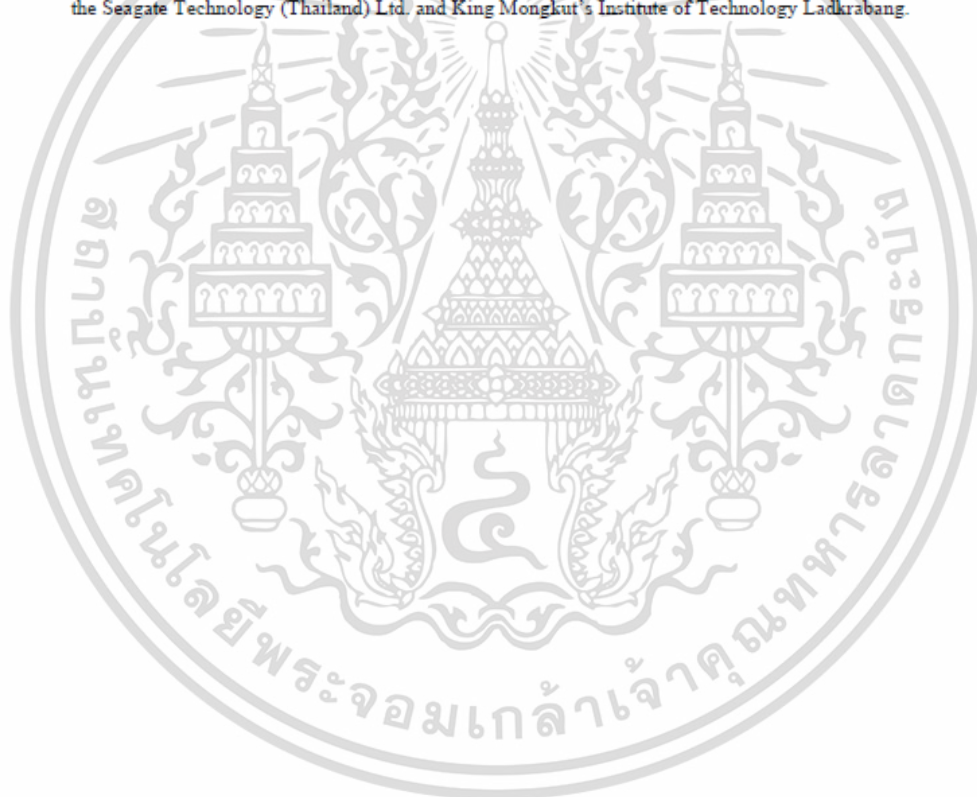
This research was supported by RRI and Seagate Technology (Thailand) Ltd. under grant no. PHD59I0059.

*Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management
Singapore, March 7-11, 2021*

Biographies

Rawinun Praserttaweelap is a Ph.D. student in the School of Engineering at King Mongkut's Institute of Technology Ladkrabang. She has completed her bachelor's degree in electrical engineering at King Mongkut's University of Technology North Bangkok and master's degree in electrical engineering at King Mongkut's Institute of Technology Ladkrabang. She has an experience in Hard Disk Drive Manufacturing for automation systems. She is interested in machine learning, robotics systems, automation machine, modeling, and prediction model.

Dr. Somyot Kiatwanidvilai is Assoc. Prof. Dr. in the School of Engineering at King Mongkut's Institute of Technology Ladkrabang. He has more experience in artificial intelligence and machine learning. His publications appeared in such journals as Computational Intelligence and Neuroscience, Sensors and Materials, International Journal of Robotics and Automation, Engineering Letters, and International Journal of Innovative Computing, Information and Control. He has authored several technical papers. He carried out the collaborative research with the Seagate Technology (Thailand) Ltd. and King Mongkut's Institute of Technology Ladkrabang.



ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาว รวินันท์ ประเสริฐทวีลาภ
วัน เดือน ปีเกิด	31 มีนาคม 2533 ที่นนทบุรี
ที่อยู่	หมู่บ้านอินดี 2 ศรีนครินทร์ ถ.ทรัพย์พัฒนา ต.บางเมือง อ.เมือง จ.สมุทรปราการ 10270
ประวัติการศึกษา	2554 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า (เกียรตินิยมอันดับ2) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้า พระนครเหนือ 2558 วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ความชำนาญเฉพาะด้าน	1.) ระบบควบคุมด้วยโปรแกรมอัตโนมัติ 2.) เทคนิคการจัดกลุ่ม (Clustering techniques) 3.) เครื่องจักรอัตโนมัติในอุตสาหกรรม
ผลงานวิจัย	[1] Rawinun Prasertaweelap, and Somyot Kiatwanidvilai, “ Head Gimbal Assembly circuit with vision technique and Fuzzy C-Means Clustering ”, 15th International Conference on Control, Automation and Systems (ICCAS), 2015 [2] Rawinun Prasertaweelap, Somyot Kiatwanidvilai, and Hisayuki Aoyama, “ Safety Path Planning with Obstacle Avoidance using Particle Swarm Optimization for AGV in Manufacturing Layout ”, International Journal of Innovative Computing Information and Control, vol. 15, no. 1, Feb 2019. [3] Rawinun Prasertaweelap, and Somyot Kiatwanidvilai, “Optimal Path Planning with A-Star Optimization and Multi-Robot Task Allocation in Hard Disk Drive Manufacturing Layout”, ICIC Express Letters, Part B: Applications An International Journal of Research and Surveys, vol. 12, no. 3, 2021