

การปรับปรุงประสิทธิภาพของปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์
ด้วยขั้นตอนวิธีความฉลาดแบบกลุ่ม

EFFICIENCY IMPROVEMENT OF CONTAINER LOADING PROBLEM
USING SWARM INTELLIGENCE ALGORITHMS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2564

KMITL-2021-SC-M-002-008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EFFICIENCY IMPROVEMENT OF CONTAINER LOADING PROBLEM
USING SWARM INTELLIGENCE ALGORITHMS



A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2021

KMITL-2021-SC-M-002-008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2021

SCHOOL OF SCIENCE

เอกสารนี้ KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การปรับปรุงประสิทธิภาพของปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ด้วยขั้นตอนวิธีความฉลาดแบบกลุ่ม
ชื่อนักศึกษา	นางสาว ศศิธร คุณสินธนานันต์
รหัสประจำตัว	58605107
ปริญญา	วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2564
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.วรางคณา กัมปาน

บทคัดย่อ

การบรรจุผลิตภัณฑ์เพื่อการขนส่งสินค้าเป็นปัจจัยสำคัญอย่างหนึ่งของวงการอุตสาหกรรมโลจิสติกส์ ซึ่งในปัจจุบันการส่งสินค้าให้ถึงมือลูกค้ายังเกิดความล่าช้า และเกิดความผิดพลาดบ่อยครั้ง ดังนั้นเพื่อให้เกิดประสิทธิภาพต่อการขนส่งสินค้า จึงต้องมีการจัดการตั้งแต่การบรรจุผลิตภัณฑ์และ การใช้พื้นที่ภายในตู้คอนเทนเนอร์ให้มีความเหมาะสม ซึ่งช่วยลดค่าใช้จ่ายและระยะเวลาในการขนส่ง ดังนั้นงานวิจัยนี้เสนอการแก้ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์แบบตู้เดียว ซึ่งใช้กลยุทธ์การจัดตำแหน่งของสินค้าโดยปราศจากรูปแบบการจัดวาง (Layout) เพื่อลดพื้นที่ว่างระหว่าง Layout เงื่อนไขที่นำมาพิจารณาคือ สินค้าที่นำมาบรรจุต้องอยู่ภายในตู้คอนเทนเนอร์ ไม่ซ้อนทับกัน การวางสินค้าต้องวางขนานไปกับตู้คอนเทนเนอร์หรือสินค้าอื่นๆ นอกจากนี้ยังพิจารณาน้ำหนัก การหมุนของสินค้า โดยใช้ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (PSO) และขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบนกกาเหว่า (CS) สำหรับเลือกรูปแบบการบรรจุสินค้าลงในตู้คอนเทนเนอร์ เพื่อให้เหลือพื้นที่ว่างน้อยที่สุดและจำลองการวางสินค้าลงในตู้คอนเทนเนอร์ด้วยภาพ 3 มิติ โดยผลลัพธ์ที่ได้ของแต่ละขั้นตอนวิธีถูกนำมาเปรียบเทียบกัน ซึ่งผลลัพธ์การทดลองด้วยวิธี PSO ดีกว่าวิธี CS ในเรื่องของการบรรจุสินค้าต่างชนิดกันลงในตู้คอนเทนเนอร์ขนาด 20 และ 40 ฟุต โดยมีเปอร์เซ็นต์การใช้พื้นที่อยู่ที่ 93.09% และ 92.23% ตามลำดับ ในขณะที่วิธี CS ดีกว่าวิธี PSO ในเรื่องของการบรรจุสินค้าชนิดเดียวกันลงในตู้คอนเทนเนอร์ขนาด 20 และ 40 ฟุต โดยมีเปอร์เซ็นต์การใช้พื้นที่อยู่ที่ 93.92% และ 93.85% ตามลำดับ อย่างไรก็ตามทั้งสองขั้นตอนวิธีมีการใช้พื้นที่ภายในตู้คอนเทนเนอร์โดยเฉลี่ยมากกว่า 90% ซึ่งให้ผลลัพธ์ที่น่าพอใจอย่างมาก

คำสำคัญ : การเพิ่มประสิทธิภาพพื้นที่การบรรจุสินค้า ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่ม

เอกสารนี้เป็นอนุภาค ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบนกกาเหว่า ปัญหาการบรรจุผลิตภัณฑ์ โลจิสติกส์การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Efficiency Improvement of Container Loading Problem Using Swarm Intelligence Algorithms
Student Name	Miss Sasithorn Koonsintananan
Student ID	58605107
Degree	Master of Science (Computer Science)
Department	Computer Science
Year	2021
Thesis Advisor	Asst.Prof.Dr.Warangkhana Kimpan

Abstract

Container loading problem (CLP) is one of a major aspect of logistics industrial. Recently, products delivery to customers is still delayed or mistaken. Therefore, to achieve efficiency in shipping, the management process should be handled from the products packing to cargo space utilization which reduce transportation costs and time. Therefore, the thesis proposed single container loading problem solving using a strategy for positioning non placement heuristics to reduce space between different layouts. The condition for this solution is packing must be placed within the container with no part of the cargo overhanging the container, and the cargoes must be placed parallel to the container surface area or other cargoes. Moreover, the conditions: weight and orientation of the cargo are also considered. Particle Swarm Optimization (PSO) and Cuckoo Search (CS) algorithms are used to select the orientation of cargoes to be packed into single container to minimize the space which simulated in 3D. The experimental results of both methods were compared and it can be indicated that Particle Swarm Optimization outperformed Cuckoo Search in terms of packing different of box sizes into the 20 and 40 feet container at 93.09% and 92.23%, respectively, while Cuckoo Search outperformed Particle Swarm Optimization in terms of packing same of box sizes into the 20 and 40 feet container at 93.92% and 93.85%, respectively. However, both algorithms can optimize the space utilization in the container more than 90 percent in average which is very satisfying.

Keywords : Space Optimization, Particle Swarm Optimization, Cuckoo Search, Container Loading Problem, Logistics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความกรุณาชี้แนะ และช่วยเหลือจาก ผู้ช่วยศาสตราจารย์ ดร.วรางคณา กัมปาน อาจารย์ที่ปรึกษาที่ได้ให้คำแนะนำ แนวคิด และตรวจแก้ไข ข้อบกพร่องต่างๆ มาโดยตลอด จนวิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์ จึงขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณะกรรมการตรวจสอบวิทยานิพนธ์ทุกท่านได้แก่ ประธานกรรมการ และ กรรมการสอบวิทยานิพนธ์ที่กรุณาตรวจสอบแก้ไขและให้คำแนะนำ จนทำให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์ยิ่งขึ้น

ขอบคุณเพื่อนๆ ที่ช่วยให้คำแนะนำที่ดี เกิดประโยชน์เพื่อจุดประกายความคิด เกี่ยวกับ วิทยานิพนธ์ฉบับนี้

สุดท้ายนี้ขอขอบพระคุณครอบครัวที่สนับสนุน และผลักดันเพื่อให้มีแรงบันดาลใจในการทำงานวิจัยให้สำเร็จ คุณค่าและประโยชน์อันพึงมีจากการศึกษาวิจัยนี้ ขอน้อมบูชาพระคุณบิดามารดา บุรพจารย์ทุกท่านที่ได้อบรมสั่งสอนวิชาความรู้ ให้ความเมตตาโดยตลอด และเป็นกำลังใจสำคัญที่ทำให้การศึกษาระดับนี้สำเร็จลุล่วงได้ด้วยดี

นางสาว ศศิธร คุณสินธนานันต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญภาพ	ช
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ความเป็นมาของระบบตู้คอนเทนเนอร์	4
2.1.1 การขนส่งด้วยระบบตู้คอนเทนเนอร์.....	4
2.1.2 การพัฒนาและความสำคัญของระบบตู้คอนเทนเนอร์.....	5
2.1.3 ขนาดและชนิดตู้คอนเทนเนอร์.....	5
2.1.4 ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์.....	6
2.2 ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization).....	8
2.3 ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบนกกาเหว่า (Cuckoo Search).....	11
2.3.1 พฤติกรรมการเลี้ยงลูกของนกกาเหว่า	12
2.3.2 Lévy Flights	12
2.3.3 การค้นหาแบบนกกาเหว่า.....	13
2.3.4 การเลือกกำหนดพารามิเตอร์ของการค้นหาแบบนกกาเหว่า	15
บทที่ 3 วิธีการดำเนินงานวิจัย	17
3.1 นิยามและเงื่อนไข	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2 กลยุทธ์สำหรับการบรรจุกล่องสินค้า.....	19
3.3 กลยุทธ์สำหรับการบรรจุกล่องสินค้ากับขั้นตอนวิธีการหาค่าเหมาะสมที่สุด แบบกลุ่มอนุภาค	23
3.4 กลยุทธ์สำหรับการบรรจุกล่องสินค้ากับขั้นตอนวิธีการหาค่าเหมาะสมที่สุด แบบนกกาเหว่า	26
บทที่ 4 ผลการวิจัยและการอภิปรายผล	31
4.1 ผลลัพธ์ของการทดลองด้วยขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค.....	32
4.2 ผลลัพธ์ของการทดลองด้วยขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบนกกาเหว่า	34
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	39
5.1 สรุปผลการวิจัย	39
5.2 ข้อเสนอแนะ	39
เอกสารอ้างอิง	41
ภาคผนวก.....	45
ภาคผนวก ก	46
ประวัติผู้เขียน.....	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 รายละเอียดของกล่องแต่ละชนิด	19
3.2 ตัวอย่างแนวการวางกล่อง	23
3.3 การกำหนดค่าเริ่มต้นสำหรับ PSO.....	24
3.4 การกำหนดค่าเริ่มต้นสำหรับ CS.....	27
4.1 กรณีของการทดลองทั้งหมด	31
4.2 ผลการทดลองด้วยวิธี PSO กับคอนเทนเนอร์ขนาด 20'	33
4.3 ผลการทดลองด้วยวิธี PSO กับคอนเทนเนอร์ขนาด 40'	33
4.4 ผลการทดลองด้วยวิธี CS กับคอนเทนเนอร์ขนาด 20'	36
4.5 ผลการทดลองด้วยวิธี CS กับคอนเทนเนอร์ขนาด 40'	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การเคลื่อนที่ของอนุภาค โดยเคลื่อนที่ไปทาง gbest g^* และค่าที่ดีที่สุดในปัจจุบัน x_i^* สำหรับแต่ละอนุภาค i	9
2.2 ซูโตโค้ดของ PSO	10
2.3 ซูโตโค้ดของการค้นหาแบบนกกาเหว่าสำหรับปัญหาในการหาค่าน้อยสุด	15
3.1 ตู้คอนเทนเนอร์ในรูปแบบ 3 มิติ	18
3.2 ทิศการหมุนของกล่องทั้ง 6 ทาง	18
3.3 แผนภาพกลยุทธ์สำหรับการบรรจุกล่องสินค้า	20
3.4 พื้นที่ใหม่ที่ได้จากการตัดสเปซ	21
3.5 อัลกอริทึมที่ใช้สำหรับตัดสเปซ	22
3.6 ลิสต์แนวการวางทั้งหมด	23
3.7 อัลกอริทึมของ PSO ที่ใช้สำหรับการเลือกลิสต์แนวการวางกล่องสินค้าที่ดีที่สุด	25
3.8 แผนภาพการทำงานของวิธีการค้นหาแบบนกกาเหว่า	26
3.9 กระบวนการเลือกรังสำหรับการวางไข่	28
3.10 ขั้นตอนการทำ Lévy Flights (λ) ของนกกาเหว่า	29
3.11 รังของนกเจ้าของรังที่มีไข่แปลกปลอมของนกกาเหว่า	29
3.12 อัลกอริทึมของ CS ที่ใช้สำหรับการเลือกลิสต์แนวการวางกล่องสินค้าที่ดีที่สุด	30
4.1 การจำลองการวางกล่องด้วยวิธี PSO ในตู้คอนเทนเนอร์ขนาด 20'	32
4.2 การจำลองการวางกล่องด้วยวิธี CS ในตู้คอนเทนเนอร์ขนาด 20'	35
4.3 การกระจายตัวของน้ำหนักกล่องภายในตู้คอนเทนเนอร์ขนาด 20' ด้วยวิธี PSO	37
4.4 การกระจายตัวของน้ำหนักกล่องภายในตู้คอนเทนเนอร์ขนาด 40' ด้วยวิธี PSO	37
4.5 การกระจายตัวของน้ำหนักกล่องภายในตู้คอนเทนเนอร์ขนาด 20' ด้วยวิธี CS	38
4.6 การกระจายตัวของน้ำหนักกล่องภายในตู้คอนเทนเนอร์ขนาด 40' ด้วยวิธี CS	38
4.7 สรุปการใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 20'	38
4.8 สรุปการใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 40'	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กระบวนการบรรจุผลิตภัณฑ์เพื่อการกระจายสินค้ามีการพัฒนาอย่างกว้างขวางเป็นผลมาจากการเติบโตอย่างรวดเร็วของเศรษฐกิจ ดังนั้นการวางแผนโลจิสติกส์สำหรับการบรรจุผลิตภัณฑ์ จึงเป็นปัจจัยสำคัญอย่างหนึ่งต่อวงการธุรกิจอุตสาหกรรม โดยปัญหาพื้นฐานของการบรรจุผลิตภัณฑ์ ขึ้นอยู่กับความหลากหลายของสถานการณ์ในการบรรจุผลิตภัณฑ์เช่น คุณลักษณะของผลิตภัณฑ์ จำนวนและชนิดของยานพาหนะ ความจุของคอนเทนเนอร์ที่ใช้ ดังนั้นควรมีการวางแผนสำหรับการบรรจุผลิตภัณฑ์ และพัฒนาในเรื่องของการใช้พื้นที่ภายในคอนเทนเนอร์ให้มีความเหมาะสมซึ่งช่วยลดค่าใช้จ่ายและระยะเวลาในการขนส่ง

โดยทั่วไปเงื่อนไขหลักที่ใช้สำหรับการจัดวางผลิตภัณฑ์ ภายในภาชนะที่บรรจุคือน้ำหนักของผลิตภัณฑ์ ความแข็งแรงในการรับน้ำหนักของภาชนะ (Bearing Strength) การซ้อนทับกันของผลิตภัณฑ์ ความเปราะบางหรือความสามารถในการรับแรงกดทับของผลิตภัณฑ์ จำนวนของภาชนะที่ใช้ (Single or Multiple Container) จำนวนมิติ (One, Two, or Three Dimensions) และรูปแบบการจัดวาง (Layout) ของผลิตภัณฑ์ เช่น Wall Building, Stack, Horizontal Layer Building, Block Building และ Guillotine Cutting ซึ่งนำไปสู่ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ (Container Loading Problem: CLP)

กลุ่มการแก้ไขปัญหาการบรรจุผลิตภัณฑ์สามารถแบ่งได้เป็น 3 กลุ่มดังนี้ วิธีที่ 1 วิธีฮิวริสติกปกติ (Conventional Heuristics) ประกอบด้วยเงื่อนไขและการจัดวาง ยกตัวอย่างเช่น การจัดวางผลิตภัณฑ์โดยใช้ขั้นตอนวิธีแบบละโมภ (Greedy Algorithm) และใช้ขั้นตอนวิธีการค้นหาเฉพาะที่ (Local Search Algorithm) สำหรับเพิ่มประสิทธิภาพในการจัดวาง [1-3] วิธีที่ 2 ใช้วิธีการค้นหาแบบต้นไม้ (Tree Search) หรือการค้นหาแบบกราฟ (Graph Search) [4-7] แต่วิธีนี้จะใช้เวลาในการค้นหานานหากมีปริภูมิการค้นหา (Search Space) กว้าง และวิธีที่ 3 วิธีเมตาฮิวริสติก (Metaheuristics) เช่นขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm: GA) และการค้นหาแบบทาบู (Tabu Search) [8-11] วิธีนี้เป็นวิธีที่ใช้กันหลายหลายเพราะเป็นวิธีที่ยืดหยุ่นและรวดเร็ว

อย่างไรก็ตามการแก้ไขปัญหามีขึ้นอยู่กัแต่ละแนวคิด ซึ่งนักวิจัยส่วนใหญ่เสนอวิธีการจัดวางสินค้าและ ปรับพารามิเตอร์ตามอัลกอริทึมที่นำเสนอ ผลลัพธ์ที่ได้คือมีการจัดสรรพื้นที่ภายในตู้คอนเทนเนอร์ได้อย่างเหมาะสม ทำให้สามารถลดจำนวนการใช้ตู้คอนเทนเนอร์ และเพิ่มจำนวนสินค้าลงในตู้คอนเทนเนอร์ได้มากขึ้น ซึ่งนำไปสู่การลดค่าใช้จ่ายและระยะเวลาที่ใช้ในการขนส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อให้เกิดประสิทธิภาพต่อการขนส่งสินค้า จึงต้องมีการวางแผนการบรรจุผลิตภัณฑ์ซึ่งเป็นการช่วยลดต้นทุนและเวลาในการขนส่ง โดยมีวัตถุประสงค์ดังนี้

1. เสนอวิธีการแก้ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ ซึ่งใช้กลยุทธ์สำหรับจัดตำแหน่งของสินค้าโดยปราศจากรูปแบบการจัดวาง (Layout) เพื่อลดพื้นที่ว่างระหว่าง Layout ซึ่งเงื่อนไขที่นำมาพิจารณาคือ น้ำหนัก การหมุนของสินค้า และแรงกดทับระหว่างสินค้า
2. ศึกษาการใช้ขั้นตอนวิธีฉลาดแบบกลุ่มด้วยวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization) และวิธีการหาค่าเหมาะสมที่สุดแบบนกกาเหว่า (Cuckoo Search) สำหรับจัดวางสินค้าภายในตู้คอนเทนเนอร์ให้เหมาะสม เพื่อเพิ่มจำนวนการบรรจุสินค้าลงในตู้คอนเทนเนอร์ตู้เดียว (Single Container) ให้ได้จำนวนมากที่สุด
3. การใช้พื้นที่ของสินค้าภายในตู้คอนเทนเนอร์สำหรับกรณีการทดลองมีเปอร์เซ็นต์การใช้พื้นที่มากกว่า 90%
4. เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีความฉลาดแบบกลุ่มในการบรรจุสินค้าลงในตู้คอนเทนเนอร์

1.3 ขอบเขตของงานวิจัย

1. ศึกษากลยุทธ์และขั้นตอนวิธีสำหรับการจัดวางสินค้าให้อยู่ภายในตู้คอนเทนเนอร์
2. วิเคราะห์จุดเด่นและจุดด้อยของแต่ละกลยุทธ์
3. เลือกใช้กลยุทธ์และขั้นตอนวิธีสำหรับการจัดวางสินค้า
4. กำหนดเงื่อนไขสำหรับการจัดวางสินค้า
5. นำกลยุทธ์และขั้นตอนวิธีที่เลือกมาสร้างโปรแกรมเพื่อค้นหาคำตอบของปัญหา
6. ปรับปรุงพารามิเตอร์เพื่อค้นหาคำตอบที่เหมาะสมที่สุด
7. นำผลลัพธ์ที่ได้จากการทดลองมาสร้างการจำลองการจัดวางสินค้าภายในตู้คอนเทนเนอร์
8. สรุปผลการทดลอง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ขั้นตอนวิธีฉลาดแบบกลุ่มที่เหมาะสมที่สุดในการจัดวางสินค้าภายในตู้คอนเทนเนอร์
2. สามารถนำขั้นตอนวิธีความฉลาดแบบกลุ่มที่หาค่าเหมาะสมที่สุดในการจัดวางสินค้าภายในตู้คอนเทนเนอร์มาพัฒนาเป็นโปรแกรมเพื่อเพิ่มประสิทธิภาพในการจัดวางสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งผลต่อการจัดสรรพื้นที่ภายในตู้คอนเทนเนอร์ได้อย่างเหมาะสม เพื่อเพิ่มจำนวนและลดเวลาในการบรรจุสินค้า

3. แสดงการจำลองการจัดวางสินค้าภายในตู้คอนเทนเนอร์ด้วยภาพแบบ 3 มิติ เพื่อให้ง่ายต่อการจัดวางสินค้าจริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้กล่าวถึงความเป็นมาของตู้คอนเทนเนอร์ ตลอดจนปัญหาเรื่องการขนส่งโดยตู้คอนเทนเนอร์ จนกลายเป็นปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ และทฤษฎีที่เป็นแนวคิดสำหรับการแก้ปัญหาคือ ความฉลาดแบบกลุ่มที่มาจากธรรมชาติได้แก่ ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคและขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบนกกาเหว่า

2.1 ความเป็นมาของระบบตู้คอนเทนเนอร์

ระบบขนส่งสินค้ามีปัจจัยสำคัญคือ การส่งของให้สอดคล้องกับพื้นที่และปริมาตรของตู้คอนเทนเนอร์ ระบบการขนส่งด้วยตู้คอนเทนเนอร์หรือ การกำหนดมาตรฐานที่ตู้คอนเทนเนอร์มีมาตรฐานเดียวกัน ไม่ว่าจะเป็นพื้นที่บนเรือเดินสมุทร รถบรรทุกส่งของ รถไฟส่งสัมภาระ ตลอดจนการไหลของเข้าเครื่องบิน อุปกรณ์ทุกอย่างจะต้องทำงานสอดคล้องกับมาตรฐานของตู้คอนเทนเนอร์ที่มีมาตรฐานเดียวกันทั่วโลก จึงเป็นที่มาของคำว่า Containerization หรือการทำทั้งระบบให้สอดคล้องกับมาตรฐานตู้คอนเทนเนอร์นั่นเอง

มาตรฐานของระบบการขนส่งด้วยคอนเทนเนอร์เกิดขึ้นมาจาก Malcom McLean ในราวปี ค.ศ. 1956 [12] ซึ่งสังเกตว่า เดิมทีหากเวลาขนส่งของขึ้นเรือ ต้องขนของทีละชิ้นขึ้นไปบนเรือ เมื่อถึงที่หมายต้องขนของลงทีละชิ้นๆ บางทีเรืออาจต้องจอดถึง 7 วันเพื่อขนของขึ้น และอีก 7 วันเพื่อขนของลง อีกทั้งรถที่มาส่งของก็ต้องจอดรอนานๆ ทำให้เสียเวลาเป็นอย่างมาก จนเกิดแนวคิดที่ยกรถบรรทุกทั้งคันขึ้นไปไว้บนเรือเสียเลย โดย Malcom ได้ออกแบบเรือและคอนเทนเนอร์ หลังจากนั้นได้มีการดำเนินการส่งของด้วยคอนเทนเนอร์ เพื่อประหยัดเวลาและค่าใช้จ่าย อีกทั้งยังเป็นการอำนวยความสะดวกต่อการขนส่งสินค้าในสมัยนั้นเป็นอย่างมาก ลักษณะการขนส่งด้วยตู้คอนเทนเนอร์จึงเริ่มเป็นที่นิยมแพร่หลายเป็นอย่างมาก ด้วยความแพร่หลายจึงมีการสร้างตู้คอนเทนเนอร์ที่มีขนาดมาตรฐานต่างๆ เพื่อให้รองรับกับการขนส่งหลากหลายประเภท เช่นการขนส่งทางน้ำ ทางบกและทางอากาศ

2.1.1 การขนส่งด้วยระบบตู้คอนเทนเนอร์

คำจำกัดความของการขนส่งด้วยระบบตู้คอนเทนเนอร์ (Container Logistics) [13] หมายถึง การวางแผนแผนการขนส่ง การไหล การประสานงาน การดำเนินการ และการควบคุมการทำงานทั้งหมดตามมาตรฐานองค์การมาตรฐานสากล (ISO 668) และข้อมูลที่เกี่ยวข้องจากต้นทางไปยังปลายทาง เมื่อเปรียบเทียบกับขนส่งสินค้าแบบทั่วไป การใช้ตู้คอนเทนเนอร์เป็นภาชนะบรรจุมีข้อได้เปรียบคือ เวลาที่ใช้การบรรจุสินค้าลดลง ทำให้เกิดความเสียหายน้อยลงและมีประสิทธิภาพมากขึ้น โดยทุกวันนี้การขนส่งในต่างประเทศใช้ตู้คอนเทนเนอร์เป็นมาตรฐานในการขนส่งสินค้าสินค้าอุปโภค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริโภคและสินค้าสำเร็จรูป นอกจากนี้สินค้าที่มีขนาดเล็กและขนาดใหญ่ รวมถึงสินค้าที่เป็นของเหลว ถูกจัดส่งในตู้คอนเทนเนอร์แบบพิเศษกำลังเพิ่มขึ้นเช่นกัน นอกจากการขนส่งในต่างประเทศโดยเรือบรรทุกคอนเทนเนอร์แล้ว การขนส่งด้วยระบบตู้คอนเทนเนอร์ยังประกอบด้วยกระบวนการบรรจุ (Stuffing) การนำออก (Stripping) การจัดเก็บ (Storing) และการจัดการกับตู้คอนเทนเนอร์อีกด้วย

2.1.2 การพัฒนาและความสำคัญของระบบตู้คอนเทนเนอร์

การขนส่งด้วยระบบตู้คอนเทนเนอร์เริ่มเมื่อปี ค.ศ. 1956 โดย Malcolm McLean ซึ่งเป็นผู้ให้บริการ ด้านการขนส่งที่ชื่อว่า Sea-Land ได้ทำการซื้อกองเรือบรรทุกน้ำมันเก่า ทำให้กลายเป็นจุดเปลี่ยนของโลกการขนส่งและโลจิสติกส์ทันที ในวันที่ 26 เมษายน ค.ศ. 1956 เรือบรรทุกน้ำมันดัดแปลง 'Ideal-X' ของ McLean ออกจากท่าเรือวอร์ก (รัฐนิวเจอร์ซีย์) ไปเมืองฮุสตัน (เท็กซัส) โดยบรรทุกตู้คอนเทนเนอร์ 56 ตู้ ต่อจากนั้นเขาได้ก่อตั้งบริการการขนส่งระหว่างสหรัฐอเมริกา ตะวันออกและฝั่งตะวันตก ทำให้เกิดการบริการจัดส่งระหว่างประเทศในทศวรรษที่ 1960 จนถึงสิ้นทศวรรษ เรือคอนเทนเนอร์ลำแรกมีกำลังการบรรจุสูงถึง 700 ตู้คอนเทนเนอร์ ในปี ค.ศ. 2006 Maersk Line นำเสนอ 'Emma Maersk' ซึ่งมีกำลังการบรรจุสูงถึง 11,000 ตู้คอนเทนเนอร์

ในช่วงทศวรรษที่ผ่านมาปริมาณบรรจุภัณฑ์ทั่วโลกเพิ่มขึ้นอย่างต่อเนื่อง อันเป็นผลมาจากโลกาภิวัตน์ การเติบโตทางเศรษฐกิจและการกระจายกิจกรรมทางภูมิศาสตร์ ก่อนเกิดวิกฤตเศรษฐกิจ ในปี ค.ศ. 2008 และ 2009 คาดว่าการเติบโตนี้จะดำเนินต่อไปอีกหลายทศวรรษด้วยอัตรา 5-8% ต่อปี ต้นทุนการขนส่งในต่างประเทศด้วยตู้คอนเทนเนอร์ลดลงอย่างมากเนื่องจากการเพิ่มขนาดของเรือ ทำให้เรือสามารถบรรทุกตู้คอนเทนเนอร์ได้มากขึ้น อย่างไรก็ตามเรือบรรทุกตู้คอนเทนเนอร์ได้สูงทำให้มีอายุการใช้งานค่อนข้างสั้น การจัดส่งตู้คอนเทนเนอร์และโลกาภิวัตน์ขึ้นอยู่กับส่วนอื่นๆ หากไม่ประสบความสำเร็จในการขนส่งด้วยระบบตู้คอนเทนเนอร์อาจจะทำให้การค้าระหว่างประเทศลดน้อยลง แต่ในขณะเดียวกันการเติบโตของการค้าโลกด้วยการแบ่งสรรแรงงานความต้องการใช้บริการจัดส่งตู้คอนเทนเนอร์และความจุตู้คอนเทนเนอร์เพิ่มขึ้น

2.1.3 ขนาดและชนิดตู้คอนเทนเนอร์

การจำแนกประเภทและขนาดของตู้คอนเทนเนอร์ที่เป็นมาตรฐานและใช้กันทั่วโลก ได้รับการจัดการตามมาตรฐานองค์การมาตรฐานสากลโดยความยาว ความกว้างและความสูงของตู้คอนเทนเนอร์มักแสดงเป็นฟุตและนิ้ว ความยาวของตู้คอนเทนเนอร์ที่ใช้กันทั่วไปคือ 20', 40' หรือ 45' ส่วนความสูงของตู้คอนเทนเนอร์ที่ใช้กันคือ 0', 8', 8'6" และ 9'6" ตู้คอนเทนเนอร์ตามมาตรฐานของ ISO นั้นมีความกว้าง 8' ถ้าตู้คอนเทนเนอร์สูง 9'6" ฟุต เรียกว่า High-cube ในขณะที่ความสูง 0' เรียกว่า แพลตฟอรม์ซึ่งเป็นเพียงผนังที่สามารถพับเก็บได้หรือไม่มีผนัง ขนาดและความจุของเรือและลานวางตู้ (Container Terminal) วัดเป็น TEU (Twenty-Equivalent Unit) ซึ่งหมายถึงตู้คอนเทนเนอร์ขนาด

ของ 20' เท่ากับ 1 TEU ดังนั้นหากตู้คอนเทนเนอร์ขนาด 40' จะเท่ากับ 2 TEU การรับน้ำหนักของตู้
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ในการศึกษา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนเทนเนอร์ขนาด 20' อยู่ที่ประมาณ 2,250 กิโลกรัมและน้ำหนักที่สามารถบรรทุกสูงสุดคือ 22,750 กิโลกรัม

นอกจากขนาดของตู้คอนเทนเนอร์แล้วยังสามารถจำแนกตู้คอนเทนเนอร์ได้ตามคุณสมบัติของการขนส่งสินค้า ซึ่งสามารถแบ่งเป็นประเภทหลักๆ ได้ดังนี้ [14]

1. ตู้แห้ง (Dry Container) เป็นตู้มาตรฐานแบบปิดที่มีสองประตู ใช้สำหรับบรรทุกสินค้าที่เป็นของแข็ง สินค้าทั่วไปที่มีการบรรจุหีบห่อหรือบรรจุสินค้าที่ไม่ต้องการรักษาอุณหภูมิ
2. ตู้แท็งก์เกอร์ (Tank Container) เป็นถังที่ใช้สำหรับการบรรทุกของเหลวหรือก๊าซ ที่มีโครงสร้างขนาดเท่าตู้แห้งทั่วไปครอบอยู่ ช่วยให้วางซ้อนทับกันได้ เหมาะกับการขนส่งร่วมกับตู้คอนเทนเนอร์ทั่วไป
3. ตู้เปิดหลังคา (Open Container) เป็นตู้คอนเทนเนอร์แบบเปิด ไม่มีหลังคาหรือผนังบางส่วน ส่วนใหญ่มีขนาด 40' ตู้ประเภทนี้ถูกออกแบบมาสำหรับการขนส่งสินค้า OOG (Out of Gauge) ตู้คอนเทนเนอร์แบบเปิดที่ใช้กันทั่วไปบางตัวเป็นแบบเปิดด้านบน (เช่น ไม่มีหลังคา) เปิดด้านข้าง (เช่น ไม่มีผนังด้านข้าง) ชั้นวางแบบแบน (เช่น มีผนังปลายที่พับเก็บได้เท่านั้น) และแพลตฟอร์ม (เช่น ไม่มีผนัง) ใช้ขนส่งสินค้าขนาดใหญ่ที่ไม่สามารถขนย้ายผ่านประตูตู้ได้
4. ตู้ห้องเย็น (Reefer) เป็นตู้แห้งที่ติดตั้งเครื่องปรับอากาศแบบมาสำหรับการบรรทุกสินค้าที่ต้องการความเย็น สามารถแยกแยะความแตกต่างของตู้ออกเป็น 2 ประเภทคือ Conair-container มีหน่วยทำความเย็นพิเศษแบบ clip-on อยู่ในตู้และ Integral reefer มีหน่วยทำความเย็นไฟฟ้าที่ใช้กระแสไฟฟ้าเสียจากนอกตู้

ทุกวันนี้การขนส่งสินค้าอุปโภคบริโภคสำเร็จรูปในต่างประเทศส่วนใหญ่ใช้ในตู้แห้งมาตรฐานขนาด 20' และ 40' ในการขนส่งสินค้า และตู้ประเภทอื่นๆ อีกประมาณ 15% ของมูลค่าการซื้อขายของที่ทำเรือ

2.1.4 ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์

ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ (Container Loading Problem: CLP) เป็นกลุ่มปัญหาเอ็นพีแบบยาก (NP-HARD) [15] ที่ไม่สามารถแก้ไขปัญหภายในเวลาที่เป็นฟังก์ชันพหุนาม (Polynomial Time) การบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์นั้น เป็นการบรรจุวัตถุ (Object) หรือ สินค้าที่มีรูปทรงสี่เหลี่ยมลงในตู้คอนเทนเนอร์ที่มีรูปทรงสี่เหลี่ยมเหมือนกัน สินค้าที่บรรจุนั้นต้องวางอยู่ในตู้คอนเทนเนอร์ไม่ทะลุหรือออกนอกตู้คอนเทนเนอร์ และต้องวางขนานไปกับตู้คอนเทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนอร์ ซึ่งการบรรจุผลิตภัณฑ์ไม่ได้จำกัดเพียงบรรจุอยู่ในตู้คอนเทนเนอร์เท่านั้น แต่ยังสามารถบรรจุผลิตภัณฑ์ลงในกระเป๋า ตะกร้า หรือพาเลทสินค้าได้อีกด้วย เป้าหมายหลักของการแก้ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์มี 2 ชนิดคือ 1) มุ่งเน้นไปที่การแก้ปัญหาเกี่ยวกับค่ามากที่สุดหมายถึงจะทำอย่างไรเพื่อบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ให้ได้มากที่สุดและ 2) การพิจารณาเรื่องการหาค่าน้อยที่สุดหมายถึง จะทำอย่างไรเมื่อบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์แล้วใช้พื้นที่ให้น้อยที่สุด รวมถึงเงื่อนไขที่ใช้ในการวางผลิตภัณฑ์ด้วย ได้แก่ น้ำหนักของผลิตภัณฑ์ที่อยู่ภายในตู้คอนเทนเนอร์ต้องไม่เกินน้ำหนักที่ตู้คอนเทนเนอร์รับได้ ขนาดของผลิตภัณฑ์ หรือการจัดตำแหน่งการวางของผลิตภัณฑ์ เป็นต้น

ในอดีตจนถึงปัจจุบันการแก้ปัญหาการบรรจุผลิตภัณฑ์ มีนักวิจัยคิดค้นอัลกอริทึมที่ใช้ในการแก้ปัญหาการบรรจุผลิตภัณฑ์แบบ 3 มิติ (3D Bin Packing: 3D-BPP) ไว้มากมายเช่น ในปี ค.ศ. 1999 Chien และ Wu [16] ได้พัฒนาโมดูลฮิวริสติก (Heuristic Module) ที่ใช้ในการแบ่งพื้นที่ว่างของตู้คอนเทนเนอร์ออกเป็นคอลัมน์เพื่อจัดสินค้าตามปริมาตรและชนิดของสินค้าซึ่งสินค้าถูกจัดลงในพื้นที่ว่างที่เหลืออยู่ทำให้มีการใช้พื้นที่ว่างภายในตู้คอนเทนเนอร์ได้อย่างเหมาะสม ในปี ค.ศ. 2000 Martello และคณะ [17] ใช้ Exact Branch และ Bound Algorithm เพื่อบรรจุสินค้าลงใน Multiple - Container ซึ่งเงื่อนไขในการวางสินค้าคือสินค้าต้องถูกวางภายในคอนเทนเนอร์โดยไม่มี การหมุนสินค้า สมมติให้สินค้ามีการเคลื่อนที่เป็นมุมฉาก ด้านข้างของสินค้าต้องวางขนานไปกับสินค้า ชั้นอื่นและตู้คอนเทนเนอร์ ในปี ค.ศ. 2001 Bortfeldt และ Gehring [18] พัฒนา Hybrid Generic Algorithm เพื่อแก้ปัญหา 3D Container Loading โดยมีเงื่อนไขคือ ด้านข้างของสินค้าขนานไปกับผนังของตู้คอนเทนเนอร์โดยที่สินค้าอยู่ภายในตู้คอนเทนเนอร์ ไม่ซ้อนทับกับสินค้าชั้นอื่นๆ สินค้าหมุนได้ 6 ทางและการวางสินค้าใช้วิธีวางเป็น Stack โดยคำนึงถึงน้ำหนัก ความคงทน และความสมดุลของสินค้า ในปี ค.ศ. 2006 Bischoff [19] พิจารณา Load Bearing Strength เพื่อหลีกเลี่ยงการแตกหักของสินค้าที่อยู่ในตู้คอนเทนเนอร์โดยจัดเรียงน้ำหนักหรือปริมาตรของสินค้า ดังนั้นสินค้าหนักจะอยู่ด้านล่างและสินค้าที่มีน้ำหนักเบาจะอยู่ชั้นบน ส่วนพื้นที่ว่างระหว่างกล่องที่รับ Load Bearing Strength สามารถนำสินค้ามาใส่ในพื้นที่ว่าง โดยสินค้านั้นสามารถเปลี่ยนแนวการวางเพื่อให้เหมาะสมกับพื้นที่ว่างที่เหลืออยู่ได้ ต่อมาในปี ค.ศ. 2010 Wu และคณะ [20] ได้นำ Genetic Algorithm มาใช้เพื่อแก้ปัญหา 3D-BPP โดยใช้ภาชนะบรรจุ (Bin) มากกว่าหนึ่งชนิดและใช้วิธีฮิวริสติก สำหรับบรรจุกลุ่มสินค้า (Packing Batch) ซึ่งมีโครงสร้างในการบรรจุแบบ Building Block จากการทดลองพบว่าอัลกอริทึมทำงานได้อย่างมีประสิทธิภาพและให้ผลลัพธ์ที่น่าพอใจเมื่อเทียบกับ

เอกสารนี้ ผลลัพธ์การบรรจุจริงและผลลัพธ์ที่ได้จาก MIP model ในปี ค.ศ. 2012 Kang และคณะ [21] เสนอ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการบรรจุอ็อบเจกต์ที่มีขนาดต่างกันลงในคอนเทนเนอร์ตู้เดียว โดยให้มีจำนวนอ็อบเจกต์ที่ถูกบรรจุมากที่สุด ใช้กลยุทธ์ I-DBLF ในการบรรจุเพื่อให้เหลือพื้นที่ว่างที่น้อยที่สุด โดยการพิจารณาจากความกว้าง ความยาว และความสูงของอ็อบเจกต์กับพื้นที่ว่าง (Overlap Packable Spaces) และใช้ Hybrid Genetic Algorithm ในการแก้ปัญหา 3D-BPP ในปี ค.ศ. 2013 W. Kimpan และ N. Tantawarak [22] พิจารณาการจัดเรียงสินค้าที่มีขนาดแตกต่างกันลงในคอนเทนเนอร์ที่มีการแบ่งคอนเทนเนอร์ออกเป็นพาร์ทิชันเพื่อเป็นลำดับในการวางสินค้าทำให้ง่ายต่อการส่งและใช้ Best-Fit Decreasing Algorithm สำหรับโพลิดสินค้ำลงคอนเทนเนอร์

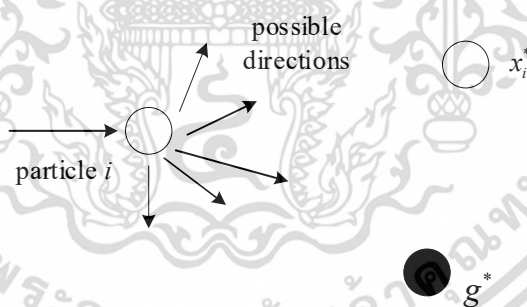
2.2 ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization)

ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization: PSO) [23] พัฒนาขึ้นโดย J.Kennedy และ R.Eberhart ในปี ค.ศ. 1995 มีแนวคิดมาจากการจำลองวิธีการหาค่าเหมาะสมที่สุดจากพฤติกรรมการเดินทางหรือการหาอาหารของฝูงสัตว์ โดยเฉพาะฝูงนกและฝูงปลา ตั้งแต่นั้นเป็นต้นมา PSO ได้สร้างความน่าสนใจในวงกว้างมากขึ้น เนื่องจากความเรียบง่ายและความยืดหยุ่น รวมทั้งสร้างหัวข้อการวิจัยที่น่าตื่นเต้นและต่อเนื่องที่เรียกว่า ความฉลาดแบบกลุ่ม (Swarm Intelligence) PSO ได้นำไปใช้ในการหาค่าเหมาะสมที่สุดเกือบทุกด้าน มีความฉลาดในการคำนวณ การออกแบบ และการนำ PSO มาผสมผสานรวมเข้ากับอัลกอริทึมอื่นๆ ที่มีอยู่ กลายเป็นอัลกอริทึมไฮบริด (Hybrid Algorithm) ยังถูกนำไปใช้อย่างกว้างขวางอีกด้วย

PSO [24] เป็นการจำลองการมีปฏิสัมพันธ์แบบสังคมของนก (Social Interaction) และพฤติกรรมที่ใช้อัจฉริยะ (Cognitive Behavior) ที่อยู่ร่วมกันเป็นฝูงซึ่งการมีปฏิสัมพันธ์แบบสังคมของนกและพฤติกรรมที่ใช้อัจฉริยะเป็นกระบวนการเรียนรู้ที่ใช้ในการพัฒนาของนกแต่ละตัว (Individual) นกแต่ละตัวถูกเรียกว่าอนุภาค (Particle) ซึ่งแต่ละอนุภาคมีข้อมูล 2 ชนิดที่ติดตัวอยู่แล้วคือ ประสบการณ์ของอนุภาค (หมายถึงรู้สถานะที่เคยเลือก) และรู้ประสิทธิภาพของอนุภาคตัวอื่นภายในสภาวะแวดล้อมเดียวกัน ซึ่งข้อมูลเหล่านี้บ่งบอกว่าอนุภาคจะเคลื่อนที่ไปอย่างไร เพื่อหาคำตอบในปริภูมิการค้นหา (Search Space) PSO มี 3 หลักการคือ การประเมิน (Evaluate) การเปรียบเทียบ (Compare) และการเลียนแบบ (Imitate) ในแต่ละอนุภาค มีการตอบสนองต่อสิ่งรอบข้างโดยขึ้นอยู่กับความน่าดึงดูดหรือความน่ารังเกียจซึ่งมีผลกระทบต่อการทำงานและตัดสินใจ มีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบประสิทธิภาพกับตัวอื่นๆ ว่ามีค่าฟิตเนส (Fitness Value) ที่ดีกว่าหรือแย่กว่า ถ้าหากว่าแย่กว่าตัวอื่นๆ ก็จะมีการเลียนแบบเพื่อให้ได้ดีกว่า ตัวอย่างเช่นหากอนุภาค A เปรียบเทียบกับอนุภาค B ที่อยู่ในพื้นที่ที่ใกล้เคียงกัน และอนุภาค B มีประสิทธิภาพดีกว่าอนุภาค A อนุภาค A จะทำการเลียนแบบเพื่อให้มีประสิทธิภาพดีขึ้น ซึ่งประสิทธิภาพการทำงานของแต่ละอนุภาคประกอบด้วยทิศทาง และความเร็ว โดยทิศทางของอนุภาคเกิดขึ้นเกิดจากการสุ่มตำแหน่งเวกเตอร์ ในอนุภาคแต่ละตัวมีค่าฟิตเนสที่บ่งบอกถึงระยะห่างของตัวมันจากแหล่งอาหาร การเคลื่อนที่ของแต่ละอนุภาคจะเคลื่อนที่ไปยังตำแหน่งปัจจุบันที่ดีที่สุดสากล (Global Best: g^*) ซึ่งเป็นค่าที่ดีที่สุดของฝูง g^* และตำแหน่งที่ดีที่สุดของตัวเอง x^* (Particle Best: pbest) อนุภาคทั้งหมดจะเคลื่อนที่ตามค่าที่ดีที่สุดของฝูง และอนุภาคที่มีค่าฟิตเนสที่ดีที่สุด ในเวลาเดียวกันก็มีแนวโน้มที่จะเคลื่อนที่แบบสุ่ม เมื่ออนุภาคพบตำแหน่งที่ดีกว่าตำแหน่งที่เคยพบก่อนหน้านี้ ทำการปรับปรุงตำแหน่งนั้นให้เป็นตำแหน่งใหม่ที่ดีที่สุดสำหรับอนุภาค i และตำแหน่งที่ดีที่สุดสำหรับอนุภาค n ทั้งหมด ที่เวลา t ใดๆ ในระหว่างการวนซ้ำ จุดมุ่งหมายคือการหาค่า g^* ที่ดีที่สุดในปัจจุบันจนกว่าวัตถุประสงค์ที่กำหนดไว้ไม่ปรับปรุงแล้ว หรือครบรอบจากการทำซ้ำ การเคลื่อนที่ของอนุภาคแสดงในรูปที่ 2.1 และสามารถสรุปได้เป็นรหัสเทียม หรือ ซูโดโค้ด (Pseudo Code) ในรูปที่ 2.2



รูปที่ 2.1 การเคลื่อนที่ของอนุภาค โดยเคลื่อนที่ไปทาง g^* และค่าที่ดีที่สุดในปัจจุบัน x_i^* สำหรับแต่ละอนุภาค i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Algorithm : Particle Swarm OptimizationObjective function $f(x), x = (x_1, \dots, x_d)^T$ Initialize locations x_i and velocity v_i of n particlesFind g^* from $\min \{f(x_1), \dots, f(x_n)\}$ (at $t = 0$)**while** (criterion) **for** loop over all n particles and all d dimensions Generate new velocity v_i^{t+1} using equation (7.1) Calculate new locations $x_i^{t+1} = x_i^t + v_i^{t+1}$ Evaluate objective functions at new locations x_i^{t+1} Find the current best for each particle x_i^* **end for** Find the current global best g^* Update $t = t + 1$ (pseudo time or iteration counter)**end while**Output the final results x_i^* and g^*

รูปที่ 2.2 ชุดโค้ดของ PSO

โดยที่

 $x_i^{*(t)}$ คือค่าที่ดีที่สุดในปัจจุบันของอนุภาค i $g^* \approx \min \{f(x_i)\}$ โดยที่ $(i = 1, 2, \dots, n)$ คือค่าที่ดีที่สุดในปัจจุบันของทั้งกลุ่มที่เวลา t

ใดๆ

 x_i และ v_i เป็นเวกเตอร์ตำแหน่งและความเร็วของอนุภาค i ตามลำดับ การปรับปรุง

ความเร็วใหม่สามารถคำนวณได้โดยสมการที่ (2.1)

$$v_i^{t+1} = v_i^t + \alpha \varepsilon_1 [g^* - x_i^t] + \beta \varepsilon_2 [x_i^{*(t)} - x_i^t] \quad (2.1)$$

ให้ ε_1 และ ε_2 เป็นเวกเตอร์สุ่มสองตัวและมีค่าอยู่ระหว่าง 0 ถึง 1 พารามิเตอร์ α และ β คือพารามิเตอร์การเรียนรู้หรือค่าคงที่ของการเร่งความเร็วซึ่งโดยทั่วไปมีค่า $\alpha \approx \beta \approx 2$ ตำแหน่งเริ่มต้นของอนุภาคทั้งหมดนั้น ควรกระจายอย่างสม่ำเสมอเพื่อให้สามารถสุ่มตัวอย่างผ่านพื้นที่ส่วนใหญ่ ซึ่งถือเป็นสิ่งสำคัญอย่างยิ่งสำหรับปัญหาต่อเนื่องหลายรูปแบบ ความเร็วเริ่มต้นของอนุภาคสามารถใช้เป็นศูนย์ได้ นั่นคือ $v_i^{t=0} = 0$ การปรับปรุงตำแหน่งใหม่สามารถปรับปรุงได้โดยสมการที่ (2.2)

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้ว่า v_i สามารถเป็นค่าใดๆ ได้ แต่มักจะมีขอบเขตอยู่ในช่วง $[0, v_{\max}]$ มีอัลกอริทึมมากมายที่ปรับปรุงจากอัลกอริทึม PSO และการปรับปรุงที่เห็นได้ชัดเจนนที่สุดคือการใช้ฟังก์ชันความเฉื่อย $\theta(t)$ ดังนั้น v_i' ถูกแทนที่ด้วย $\theta(t)v_i'$ ตามสมการที่ (2.3)

$$v_i'^{t+1} = \theta v_i' + \alpha \varepsilon_1 [g^* - x_i'] + \beta \varepsilon_2 [x_i^{*(k)} - x_i'] \quad (2.3)$$

ค่า θ มีค่าระหว่าง 0 ถึง 1 โดยทั่วไปฟังก์ชันความเฉื่อยสามารถใช้เป็นค่าคงที่คือ $\theta \approx 0.5 \sim$ ซึ่งเทียบเท่ากับการใช้มวลเสมือนจริง เพื่อให้การเคลื่อนที่ของอนุภาคมีความเสถียรและคาดว่าอัลกอริทึมจะเร็วขึ้น

จากการศึกษาพบว่านักวิจัยได้นำ PSO มาประยุกต์ใช้เพื่อการแก้ปัญหาการบรรจุผลิตภัณฑ์ ในปี ค.ศ. 2016 มีการนำ PSO มาเพื่อใช้ในการแก้ปัญหาการบรรจุผลิตภัณฑ์แบบ 2 มิติ สำหรับจัดสินค้าประเภทต่างๆ ลงบนเรือหรือในตู้คอนเทนเนอร์โดย Shingade และคณะ [25] นำวิธี Bottom Left with Fill (BLF) เพื่อใช้สำหรับการจัดวางสินค้า และการใช้ PSO เพื่อเปรียบเทียบตำแหน่งที่ให้ค่าฟิตเนสที่ดีที่สุด ซึ่งหมายถึงการใช้พื้นที่ที่น้อยที่สุดสำหรับการบรรจุสินค้า ในปี ค.ศ. 2012 Domingo และคณะ [24] แก้ไขปัญหา CLP โดยใช้กล่องสี่เหลี่ยมขนาดต่างๆ ซึ่งบรรจุในตู้คอนเทนเนอร์ตู้เดียว โครงสร้างการบรรจุเป็น Layer และนำ PSO มาใช้ในการประเมินค่า สร้างลำดับการบรรจุที่เป็นไปได้และคำนวณปริมาตรที่ใช้ในแต่ละลำดับเพื่อค้นหาลำดับการบรรจุที่ดีที่สุด ต่อมาในปี ค.ศ. 2017 Francois และคณะ [26] เสนอแนวคิดเรื่องการตัด (Cutting) สำหรับปัญหาการบรรจุผลิตภัณฑ์ เพื่อลดการใช้ปริมาตรของวัตถุทั้งหมด และวางวัตถุเหล่านั้นลงในพื้นที่ว่างไม่ต่อเนื่อง (Discrete Space) โดยการใช้พิกัด มีการสร้างแบบจำลองพื้นที่ว่างไม่ต่อเนื่องเพื่อจำลองการโหลดวัตถุ โดยใช้อัลกอริทึม P.S.O. - Starcraft และฟังก์ชันวัตถุประสงค์ Waterfalls (W.O.F.) สำหรับปรับลดการใช้ปริมาตรให้เหมาะสม

2.3 ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบนกกาเหว่า (Cuckoo Search)

การค้นหาแบบนกกาเหว่า (Cuckoo Search: CS) [23, 27, 28] เป็นหนึ่งในวิธีเมตาฮิวริสติกทางธรรมชาติล่าสุด ที่พัฒนาขึ้นในปี ค.ศ. 2009 โดย X. S. Yang และ S. Deb CS ได้นำพฤติกรรมการฝากเลี้ยง (Brood Parasitism) ของนกกาเหว่าบางสปีชีส์มาเป็นแนวคิดในการแก้ปัญหา นอกจากนี้อัลกอริทึมยังถูกปรับปรุงโดยการใช้ Lévy Flights ซึ่งเป็นพฤติกรรมการบินของสัตว์และแมลงหลายชนิด แทนที่จะใช้วิธีการเดินสุ่มแบบไอโซโทรปิก (Isotropic Random Walks)

2.3.1 พฤติกรรมการเลี้ยงลูกของนกกาเหว่า

นกกาเหว่าเป็นนกที่น่าสนใจไม่เพียงเพราะมีเสียงอันไพเราะ แต่ยังมีพฤติกรรมการสืบพันธุ์แบบก้าวร้าว นกกาเหว่าบางสปีชีส์เช่น *Ani* และ *Guira* เมื่อวางไข่ในรังอื่น พวกมันอาจจะกำจัดไข่ของนกเจ้าของรังออกไป เพื่อเพิ่มความน่าจะเป็นในการฟักไข่ของตัวเอง และมีหลายสายพันธุ์ที่มีพฤติกรรมการฝากเลี้ยงโดยแม่นกวางไข่ของตัวเองไว้ในรังของนกตัวอื่น โดยทั่วไปพฤติกรรมการฝากเลี้ยงของนกกาเหว่าแบ่งได้เป็น 3 ประเภทคือ พฤติกรรมการฝากเลี้ยงระหว่างสิ่งมีชีวิตชนิดเดียวกัน (Intraspecific Brood Parasitism) พฤติกรรมการผสมพันธุ์แบบมีนกผู้ช่วย (Cooperative Breeding) และการครอบครองรัง นกเจ้าของรังบางตัวสามารถมีส่วนร่วมในความขัดแย้งโดยตรงกับลูกนกกาเหว่า หากนกเจ้าของรังพบไข่ที่ไม่ใช่ของตัวเอง พวกมันจะกำจัดไข่แปลกปลอมออก หรือทิ้งรังไปเฉยๆ และสร้างรังใหม่ที่อื่นแทน ในนกกาเหว่าบางสปีชีส์เช่น *Tapera* มีการพัฒนาพฤติกรรมการฝากเลี้ยงคือ นกกาเหว่าตัวเมียมักมีความเชี่ยวชาญในการเลียนแบบสี และรูปแบบของไข่ไม่กินชนิดตามนกเจ้าของรังที่เลือก สิ่งนี้จะช่วยลดความน่าจะเป็นของไข่ที่ถูกทอดทิ้งและเพิ่มการสืบพันธุ์

นอกจากนี้ช่วงเวลาในการวางไข่ของนกกาเหว่าบางสายพันธุ์ก็น่าทึ่งเช่นกัน นกกาเหว่ามักเลือกรังที่นกเจ้าของรังวางไข่เอง โดยทั่วไปแล้วไข่นกกาเหว่ามักจะฟักออกมาเร็วกว่าไข่ของนกเจ้าของรังเล็กน้อย เมื่อนกกาเหว่าฟักออกมาเป็นตัวแรก สัญชาตญาณแรกที่ทำก็คือการขบไล่ไข่นกเจ้าของรังโดยการผลัดไข่ออกจากรังอย่างสุ่มๆ ซึ่งเป็นการเพิ่มส่วนแบ่งอาหารของลูกนกกาเหว่าที่จัดหาให้โดยนกเจ้าของรัง จากการศึกษาซึ่งพบอีกว่า ลูกนกกาเหว่ายังสามารถเลียนแบบเสียงร้องของลูกนกเจ้าของรังเพื่อให้มีโอกาสในการได้รับอาหารมากขึ้น

2.3.2 Lévy Flights

การศึกษาพฤติกรรมการบินของสัตว์และแมลงหลายชนิด แสดงให้เห็นถึงลักษณะทั่วไปของ Lévy Walk คำว่า Lévy Walk [29] นั้นประกาศใช้ครั้งแรกโดย Schlesinger และคณะ [30] เพื่ออธิบายการแพร่กระจายของอนุภาคที่ผิดปกติซึ่งไม่ได้อยู่ภายใต้การเคลื่อนที่แบบบราวเนียน (Brownian Motion: BM) BM แสดงลักษณะการแพร่กระจายของอนุภาคขนาดเล็กที่มีเส้นทางอิสระ (หรือเที่ยวบิน) และเวลาหยุดพักเฉลี่ยระหว่างเที่ยวบิน เที่ยวบินถูกกำหนดให้เป็นการเดินทางเป็นเส้นตรงที่ยาวที่สุดจากที่หนึ่งไปอีกที่หนึ่ง โดยอนุภาคไม่มีการเปลี่ยนทิศทางหรือหยุดชั่วคราว จากการศึกษาล่าสุดโดย Reynolds และ Frye [31] แสดงให้เห็นว่าแมลงวันผลไม้หรือ *Drosophila melanogaster* สำนวณวิถีการบินโดยใช้เส้นทางการบินเป็นเส้นตรงที่คั่นด้วยการเลี้ยว 90 องศาแบบฉับพลัน ซึ่งนำไปสู่ลักษณะของ Lévy Flights รูปแบบการค้นหาอิสระเป็นระยะมาตราส่วน จากการศึกษาพฤติกรรมของมนุษย์เช่น รูปแบบการหาอาหารของนักล่าสัตว์ Ju/' ho Hoansi นั้นแสดงให้เห็นคุณลักษณะทั่วไปของ Lévy Flights [32] แม้แต่แสงก็มีความเกี่ยวข้องกับ Lévy Flights ได้เช่นกัน ต่อจากนั้นพฤติกรรมดังกล่าวได้ถูกนำไปใช้กับการเพิ่มประสิทธิภาพและการค้นหาที่ดีที่สุด ซึ่งผลลัพธ์ที่ได้แสดงถึงความสามารถที่มีแนวโน้มดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 การค้นหาแบบนกกาเหว่า

CS เป็นหนึ่งในอัลกอริทึมเมตาฮีวิริสติกล่าสุด ที่ได้รับแรงบันดาลใจจากธรรมชาติ พัฒนาขึ้นในปี ค.ศ. 2009 โดย X. S. Yang และ S. Deb CS โดยมีพื้นฐานมาจากพฤติกรรมการฟ্লাกเลี้ยงของนกกาเหว่าบางสายพันธุ์ นอกจากนี้ขั้นตอนการค้นหาแบบนกกาเหว่าได้รับการปรับปรุงโดยใช้ Lévy Flights แทนที่จะใช้รูปแบบการเดินสุ่มแบบไอโซโทริก จากการศึกษาล่าสุดแสดงให้เห็นว่า CS อาจมีประสิทธิภาพมากกว่า PSO และ GA เพื่อความง่ายในการอธิบายมาตรฐานของ CS ใช้กฎอุดมคติสามข้อต่อไปนี้

1. นกกาเหว่าแต่ละตัววางไข่ครั้งละฟอง และทิ้งไว้ในรังที่สุ่มเลือก
2. รังที่ดีที่สุดกับไข่ที่มีคุณภาพสูงจะถูกส่งต่อไปยังรุ่นต่อไป
3. กำหนดจำนวนรังของนกเจ้าของรังที่ใช้ได้ และไข่ที่วางโดยนกกาเหว่านั้น นกเจ้าของรังสามารถค้นพบไข่ของนกกาเหว่าด้วยความน่าจะเป็น $p_a \in (0,1)$ ในกรณีนี้นกเจ้าของรังสามารถกำจัดไข่ หรือทิ้งรังไปและสร้างรังใหม่

จากกฎข้อสุดท้ายนี้สามารถประมาณค่าความน่าจะเป็นได้โดยแทนที่ p_a ของรังนกเจ้าของรัง n รัง ด้วยรังใหม่ (ด้วยคำตอบใหม่แบบสุ่ม) สำหรับปัญหาในการหาค่ามากที่สุด (Maximization Problem) คุณภาพหรือค่าฟิตเนสของคำตอบอาจเป็นสัดส่วนกับค่าของฟังก์ชันวัตถุประสงค์ รูปแบบอื่นๆ ของค่าฟิตเนสสามารถกำหนดในลักษณะที่คล้ายกับฟังก์ชันฟิตเนสในขั้นตอนวิธีทางพันธุกรรม

ในมุมมองของการนำไปใช้ สามารถใช้การแทนแบบง่ายๆ ดังนี้ ไข่แต่ละฟองในรังแสดงถึงคำตอบของปัญหาและนกกาเหว่าแต่ละตัวสามารถวางไข่ได้เพียงไข่เดียว (ดังนั้นจึงเป็นตัวแทนของคำตอบของปัญหาเพียงคำตอบเดียว) เป้าหมายคือการหาคำตอบใหม่ของปัญหา และคำตอบที่ดีกว่า (ไข่นกกาเหว่า) เพื่อแทนที่คำตอบของปัญหาที่ไม่ดีในรัง อัลกอริทึมนี้ยังสามารถนำไปใช้ในกรณีที่ซับซ้อนมากขึ้น โดยแต่ละรังสามารถมีไข่ได้หลายฟองซึ่งแสดงถึงชุดคำตอบของปัญหา การใช้วิธีที่ง่ายที่สุดคือแต่ละรังมีเพียงไข่เดียว ในกรณีนี้ไม่มีความแตกต่างระหว่างไข่ รังหรือนกกาเหว่าเนื่องจากแต่ละรังสอดคล้องกับไข่หนึ่งฟองซึ่งยังหมายถึงนกกาเหว่าหนึ่งตัว

อัลกอริทึมนี้ใช้การผสมผสานที่สมดุลของการเดินแบบสุ่มในพื้นที่และการเดินแบบสำรวจทั่วโลกซึ่งควบคุมโดยการสลับพารามิเตอร์ p_a การเดินสุ่มเฉพาะที่สามารถเขียนเป็นสมการที่ (2.4)

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \varepsilon) \otimes (x_j^t - x_k^t), \quad (2.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

x_j' และ x_k' เป็นการแก้ปัญหาที่แตกต่างกันสองแบบที่เลือกโดยการสุ่มเปลี่ยน

$H(u)$ เป็นฟังก์ชัน Heaviside

ε คือตัวเลขสุ่มที่ถูกสุ่มจากการแจกแจงแบบเดียวกัน

s คือระยะกระโดด (Step size)

⊗ หมายถึงผลคูณของสองเวกเตอร์

ในอีกทางหนึ่งการเดินสุ่มทั่วพื้นที่นั้นดำเนินการโดยใช้ Lévy Flights ดังสมการที่ (2.5) และ (2.6)

$$x_i'^{t+1} = x_i' + \alpha L(s, \lambda), \quad (2.5)$$

โดยที่

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{s^{1+\lambda}}, (s \gg 1) \quad (2.6)$$

$\alpha > 0$ หมายถึงขนาดมาตราส่วนของระยะกระโดดซึ่งควรมีความเกี่ยวข้องกับขนาดของปัญหาที่สนใจ ในกรณีอื่นสามารถใช้ $\alpha = O(L/10)$ โดยที่ L แทนขนาดของปัญหาที่สนใจ ในบางกรณี $\alpha = O(L/10)$ มีประสิทธิภาพมากขึ้นและหลีกเลี่ยงการเดินทางไกลเกินไป เห็นได้ชัดว่าค่า α ในสมการการอัปเดตทั้งสองนี้อาจแตกต่างกันดังนั้นจึงนำไปสู่พารามิเตอร์ที่แตกต่างกันสองแบบ ค่า α_1 และ α_2 ดังนั้น $\alpha_1 = \alpha_2 = \alpha$ ตามเงื่อนไขพื้นฐานทั้ง 3 ข้อของ CS สามารถสรุปเป็นชุดโค้ดดังรูปที่ 2.3

การปรับปรุงสมการ (2.5) เป็นหลักสมการสโตแคสติก (Stochastic Equation) สำหรับการเดินแบบสุ่ม โดยทั่วไปการเดินแบบสุ่มเป็นห่วงโซ่มาร์คอฟ (Markov Chains) ซึ่งสถานะหรือสถานที่ต่อไปขึ้นอยู่กับตำแหน่งปัจจุบันเท่านั้น (เทอมแรกในสมการก่อนหน้า) และความน่าจะเป็นในการเปลี่ยนแปลง (เทอมที่สอง) อย่างไรก็ตามโซลูชันใหม่ควรมีการสร้างโดยการสุ่มพิดในพื้นที่ห่างไกลและสถานที่ควรจะไกลพอกจากคำตอบที่ดีที่สุดในปัจจุบัน สิ่งนี้จะทำให้แน่ใจว่าระบบจะไม่ติดอยู่ในค่าที่ดีที่สุดเฉพาะที่ (Local Optimum)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Algorithm : Cuckoo Search via Lévy FlightsObjective function $f(x), x = (x_1, \dots, x_d)^T$ Generate initial population of n host nest x_i **while** ($t < \text{MaxGeneration}$) or (stop criterion)

Get a cuckoo randomly

Generate a solution by Lévy Flights [e.g., Eq.(2.4)]

 Evaluate its solution quality or objective value f_i Choose a nest among n (say, j) randomly **if** ($f_i < f_j$) Replace j by the new solution i **end** A fraction (p_a) of worse nests are abandoned

New nests / solutions are build / generated by Eq.(2.3)

Keep best solutions (or nests with quality solutions)

Rank the solutions and find the current best

 Update $t \leftarrow t + 1$ **end while**

Postprocess results and visualization

รูปที่ 2.3 ชุดโค้ดของการค้นหาแบบนกกาเหว่าสำหรับปัญหาในการหาค่าน้อยสุด

การศึกษาเกี่ยวกับการค้นหาแบบนกกาเหว่ากำลังขยายตัวอย่างรวดเร็ว เป็นที่ได้รับความสนใจเป็นอย่างมากและจากการศึกษาเมื่อไม่นานมานี้โดยใช้การค้นหาแบบนกกาเหว่าถูกนำไปใช้การแก้ปัญหาที่หลากหลาย ในปี ค.ศ. 2009 Yang และ Deb [33] ใช้ CS และ Lévy Flights เพื่อแก้ปัญหาการหาค่าที่เหมาะสมที่สุด โดยตรวจสอบอัลกอริทึมกับฟังก์ชันทดสอบและเปรียบเทียบประสิทธิภาพกับ GA และ PSO จากการจำลองและการเปรียบเทียบฟังก์ชันวัตถุประสงค์หลายรูปแบบแสดงให้เห็นว่า CS เหนือกว่า GA และ PSO ส่วนหนึ่งเนื่องจาก CS มีการปรับพารามิเตอร์น้อยกว่าใน GA และ PSO อีกส่วนเพราะขนาดของประชากรมี p_a เป็นพารามิเตอร์หลัก และอัตราการลู่เข้าไม่มีผลต่อพารามิเตอร์ p_a ซึ่งหมายความว่าไม่ต้องปรับพารามิเตอร์ CS จึงเหมาะกับการแก้ไขปัญหาค่าที่เหมาะสมที่สุดเมื่อเทียบกับอัลกอริทึมเมตาฮีวิริสติกอื่นๆ

2.3.4 การเลือกกำหนดพารามิเตอร์ของการค้นหาแบบนกกาเหว่า

ใน CS นอกจากขนาดของประชากร (n) และค่าความน่าจะเป็นในการค้นพบไข่ของนกกาเหว่า (p_a) ที่เป็นพารามิเตอร์หลักแล้ว ยังประกอบไปด้วยค่าระยะกระโดด (α) และค่าที่เปลี่ยนแปลงจากพฤติกรรมของ Lévy Flights (λ) เป็นพารามิเตอร์อีกด้วย เนื่องจากค่า α และ λ เป็นค่าคงที่ [23] จึงสามารถมีได้หลายค่าและจากการศึกษาพบว่าสามารถกำหนดค่า $\alpha = 1.5$ และ

$\lambda = 0.01$ สำหรับปัญหาส่วนใหญ่ได้ ซึ่งการกำหนดค่า α และ λ เป็นการสรุปผลจากการทดลอง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการกำหนดพารามิเตอร์หลักคือ จำนวนรังของนกเจ้าของรัง หรือขนาดของประชากร (n) ได้มีการใช้ $n = 5, 10, 15, 20, 30, 40, 50, 100, 150, 250, 500$ และค่าความน่าจะเป็น (p_a) ได้มีการใช้ $p_a = 0, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5$ ซึ่งจากการทดลองพบว่า $n = 15$ ถึง 40 และ $p_a = 0.25$ เพียงพอสำหรับปัญหาการเพิ่มประสิทธิภาพส่วนใหญ่ จากการวิเคราะห์ผลลัพธ์สามารถบอกเป็นนัยได้ว่า อัตราการลู่เข้าในระดับหนึ่งไม่ไวต่อพารามิเตอร์ที่ใช้ ซึ่งหมายความว่าไม่จำเป็นต้องมีการปรับค่าสำหรับปัญหาใดๆ ยกตัวอย่างฟังก์ชันง่ายๆ ซึ่งเป็นหนึ่งในฟังก์ชันที่ใช้ในการทดสอบคือ ฟังก์ชัน Michalewicz สองตัวแปรดังสมการที่ (2.7)

$$f(x, y) = -\sin(x) \sin^{2m} \left(\frac{x^2}{\pi} \right) - \sin(y) \sin^{2m} \left(\frac{2y^2}{\pi} \right) \quad (2.7)$$

โดยที่ $m = 10$ และ $(x, y) \in [0, 5] \times [0, 5]$ ฟังก์ชันนี้มีค่าที่ต่ำที่สุดสากล (Global Minimum) คือ $f_* \approx -1.8013$ ที่ $(2.20319, 1.57049)$ ค่าที่ดีที่สุดสากล (Global Optimum) นี้สามารถพบได้ง่ายโดยใช้ CS โดยการกำหนดให้ $n = 15$, $\alpha = 1$ และ $p_a = 0.25$ นอกจากนี้ในการทดลองยังพบว่า รังกระจายอยู่ที่ local optima ที่แตกต่างกันในกรณีของฟังก์ชันหลายรูปแบบ ซึ่งหมายความว่า CS สามารถค้นหาอพติมาทั้งหมดได้พร้อมกันหากจำนวนรังสูงกว่าจำนวน local optima ข้อได้เปรียบนี้อาจมีความสำคัญมากขึ้นเมื่อกำลังจัดการกับปัญหาการเพิ่มประสิทธิภาพหลายรูปแบบและหลายวัตถุประสงค์ อีกทั้ง CS ได้มีการพัฒนาไปหลายรูปแบบในช่วงไม่กี่ปีที่ผ่านมา ถึงแม้จะมีเพียงเนื้อหาสั้น ๆ ที่เกี่ยวกับอัลกอริทึมนี้ แต่ CS ก็ดึงดูดความสนใจอย่างมากเนื่องจากมีความเรียบง่าย มีประสิทธิภาพและความยืดหยุ่นเป็นผลให้มีการศึกษาเรื่องนี้กันอย่างแพร่หลาย รวมทั้งได้รับการพัฒนาเพื่อเพิ่มประสิทธิภาพอย่างต่อเนื่อง ส่วนการศึกษาที่มีประโยชน์อย่างหนึ่งคือการสร้าง CS เวอร์ชันที่ไม่ต่อเนื่องเพื่อให้สามารถแก้ปัญหาการจัดตารางเวลาและปัญหาการเพิ่มประสิทธิภาพ Combinatorial ได้อย่างมีประสิทธิภาพ

สำหรับการนำ CS มาประยุกต์ใช้กับการแก้ปัญหาการบรรจุผลิตภัณฑ์ ยังมีจำนวนน้อย เนื่องจาก CS เป็นแนวคิดแบบใหม่ที่มีเมื่อไม่นานมานี้ แต่ก็ยังมีบางนักวิจัยนำ CS มาปรับใช้สำหรับการแก้ปัญหาการบรรจุผลิตภัณฑ์เช่น ในปี ค.ศ. 2012 Layeb และ Boussalia [34] เสนอวิธีการที่ใช้ Quantum Inspired Cuckoo Search เพื่อแก้ปัญหาการบรรจุผลิตภัณฑ์ลงตู้คอนเทนเนอร์ตู้เดียว โดยการกำหนดควอนตัมที่เหมาะสมโดยขึ้นอยู่กับคิวบิก (Qubit) ที่แสดงถึงผลลัพธ์ของการบรรจุผลิตภัณฑ์ลงตู้คอนเทนเนอร์ และการกำหนดตำแหน่งการจัดวางก่อนสำหรับวิธีที่นำเสนอ (Hybrid Quantum) ซึ่งวัดโดยใช้ First Fit เพื่อบรรจุออบเจ็กต์ที่ยังไม่ได้บรรจุโดยการวัดแบบพื้นฐาน ทำให้

ช่วยลดขนาดของประชากรและจำนวนรอบของการวนซ้ำ ผลลัพธ์ที่ได้แสดงความเป็นไปได้และความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักผู้แต่งเห็นว่าเว็บไซต์บนอินเทอร์เน็ตมีการนำ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีประสิทธิภาพ ในปี ค.ศ. 2012 Yesodha [35] แสดงอัลกอริทึมฮิวริสติก First Fit และ Best Fit เพื่อใช้ในแก้ไขปัญหาการหาค่าเหมาะที่สุดเชิงการจัด (Combinatorial Optimization Problem) รวมทั้งสรุปการเปรียบเทียบประสิทธิภาพของ First Fit และ Best Fit คือผลลัพธ์ของ Best Fit ดีกว่า จากปัญหาการบรรจุผลิตภัณฑ์ โดยเงื่อนไขที่นำมาใช้ทดสอบประสิทธิภาพของอัลกอริทึมคือ การลดจำนวนของภาชนะที่บรรจุ การลดพื้นที่ที่เสียเปล่า และการลดเวลาในการประมวลผล ซึ่ง Best Fit ดีกว่าในเรื่องของจำนวนภาชนะที่ใช้ พื้นที่ที่เสียไปในภาชนะและเวลาในการประมวลผลเมื่อเทียบกับ First Fit และในปี ค.ศ. 2016 Zendaoui และ Layeb [36] เสนอวิธีการแก้ปัญหาการบรรจุผลิตภัณฑ์ลงในภาชนะแบบ 1 มิติ (1-BPP) โดยใช้วิธี Adaptive Cuckoo search (ACS) และการใช้เทคนิค Rank Order Value (ROV) เพื่อเปลี่ยน Continuous Solutions เป็น Discrete Integer Permutations โดยทดสอบกับอินสแตนซ์มาตรฐานของ 1-BPP สามคลาส และเปรียบเทียบประสิทธิภาพกับอัลกอริทึม First Fit Decreasing (FFD), Ant System Algorithm (AS), Firefly Algorithm based Ant Colony Optimization (FCO) และ Quantum Cuckoo Search Algorithm (QICS) พบว่า ASC มีประสิทธิภาพเหนือกว่า เนื่องจาก ACS มีพารามิเตอร์จำนวนน้อยจึงง่ายต่อการปรับใช้และการปรับพารามิเตอร์

บทที่ 3

วิธีการดำเนินงานวิจัย

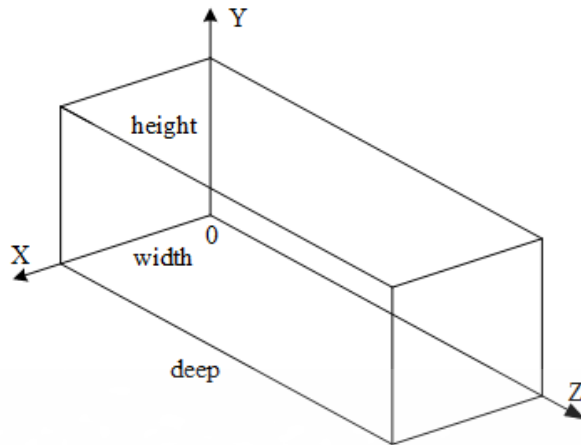
งานวิจัยนี้เป็นการแก้ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ตู้เดียวแบบ 3 มิติ (3D-BPP) ใช้กลยุทธ์สำหรับการจัดตำแหน่งการวางสินค้าโดยปราศจากรูปแบบการจัดวาง (Layout) เพื่อลดพื้นที่ว่างระหว่าง Layout ใช้วิธี PSO และ CS เพื่อหาตำแหน่งการวางสินค้าภายในตู้คอนเทนเนอร์ หลังจากนั้นนำผลลัพธ์ที่ได้ของแต่ละวิธีมาเปรียบเทียบประสิทธิภาพการจัดวางสินค้าต่อการใช้พื้นที่ภายในตู้คอนเทนเนอร์ได้อย่างเหมาะสมมากที่สุด และจำลองการจัดวางสินค้าแต่ละวิธีด้วยภาพ 3 มิติ ซึ่งในบทนี้กล่าวถึงขั้นตอนวิธีการดำเนินงานวิจัยสำหรับการแก้ปัญหา 3D-BPP โดยแบ่งหัวข้อออกเป็น การกำหนดนิยามและเงื่อนไข กลยุทธ์สำหรับการบรรจุกล่องสินค้า กลยุทธ์สำหรับการบรรจุกล่องสินค้ากับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค และกลยุทธ์สำหรับการบรรจุกล่องสินค้ากับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบนกกาเหว่า

3.1 นิยามและเงื่อนไข

การแก้ปัญหา 3D-BPP สามารถกำหนดเงื่อนไขได้มากมายตามแต่ละแนวคิด ซึ่งเงื่อนไขสำหรับการแก้ปัญหา 3D-BPP นี้คือ การวางสินค้าต้องวางอยู่ภายในตู้คอนเทนเนอร์ ไม่ทะลุหรือออกนอกตู้คอนเทนเนอร์ ไม่ซ้อนทับกัน การวางสินค้าต้องวางขนานไปกับตู้คอนเทนเนอร์หรือ สินค้าชิ้นอื่นๆ

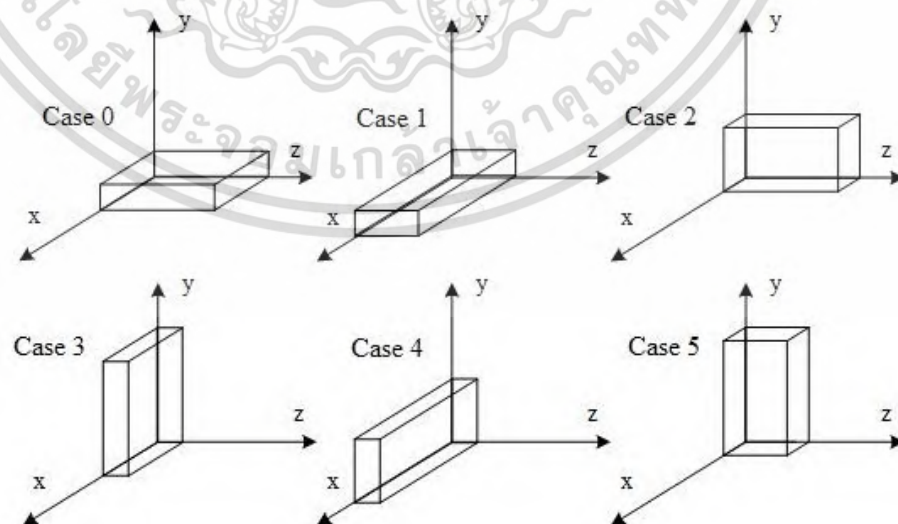
ตู้คอนเทนเนอร์ประกอบด้วยค่า Dimension Volume (ความกว้าง ความสูง และความลึก) แสดงโดย Coordinate Vector (X_w, Y_H, Z_L) และ Dimension Position (X, Y, Z) ตำแหน่งเริ่มต้นเท่ากับ $(0, 0, 0)$ แสดงดังรูปที่ 3.1 โดยขนาดของตู้คอนเทนเนอร์ที่ใช้ในการทดลองอ้างอิงจากมาตรฐานเลขที่ มอก. 587-2528 ขนาดตู้คอนเทนเนอร์ที่นิยมใช้มี 2 ขนาดคือ 20 ฟุต $(243 \times 259 \times 609 \text{ cm}^3)$ และ 40 ฟุต $(243 \times 259 \times 1219 \text{ cm}^3)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ตู้คอนเทนเนอร์ในรูปแบบ 3 มิติ

สินค้าหรือกล่องมีรูปทรงสี่เหลี่ยมประกอบด้วยค่า Dimension Volume (ความกว้าง ความสูง และความลึก) แสดงโดย Coordinate Vector (x_w, y_h, z_l) ค่า Dimension Position (x, y, z) สถานะของกล่อง ถ้ากล่องไม่ถูกบรรจุลงในตู้คอนเทนเนอร์สถานะของกล่องนั้นมีค่าเป็น 0 และถ้าสถานะของกล่องมีค่าเท่ากับ 1 แสดงว่ากล่องถูกบรรจุลงในตู้คอนเทนเนอร์ id แสดงหมายเลขของกล่อง ซึ่งแต่ละกล่องมี id ที่ไม่ซ้ำกันเป็นของตัวเอง ทิศการหมุนของกล่อง มี 6 ทาง (มีค่า 0-5) แสดงดังรูปที่ 3.2 และขนาดของกล่องที่ใช้ทำการทดลองมี 5 ชนิดคือ S, M, L, XL และ OVER (อ้างอิงจากเว็บไซต์อย่างเป็นทางการของ Kerry) รายละเอียดของกล่องแต่ละชนิด แสดงดังตารางที่ 3.1



รูปที่ 3.2 ทิศการหมุนของกล่องทั้ง 6 ทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

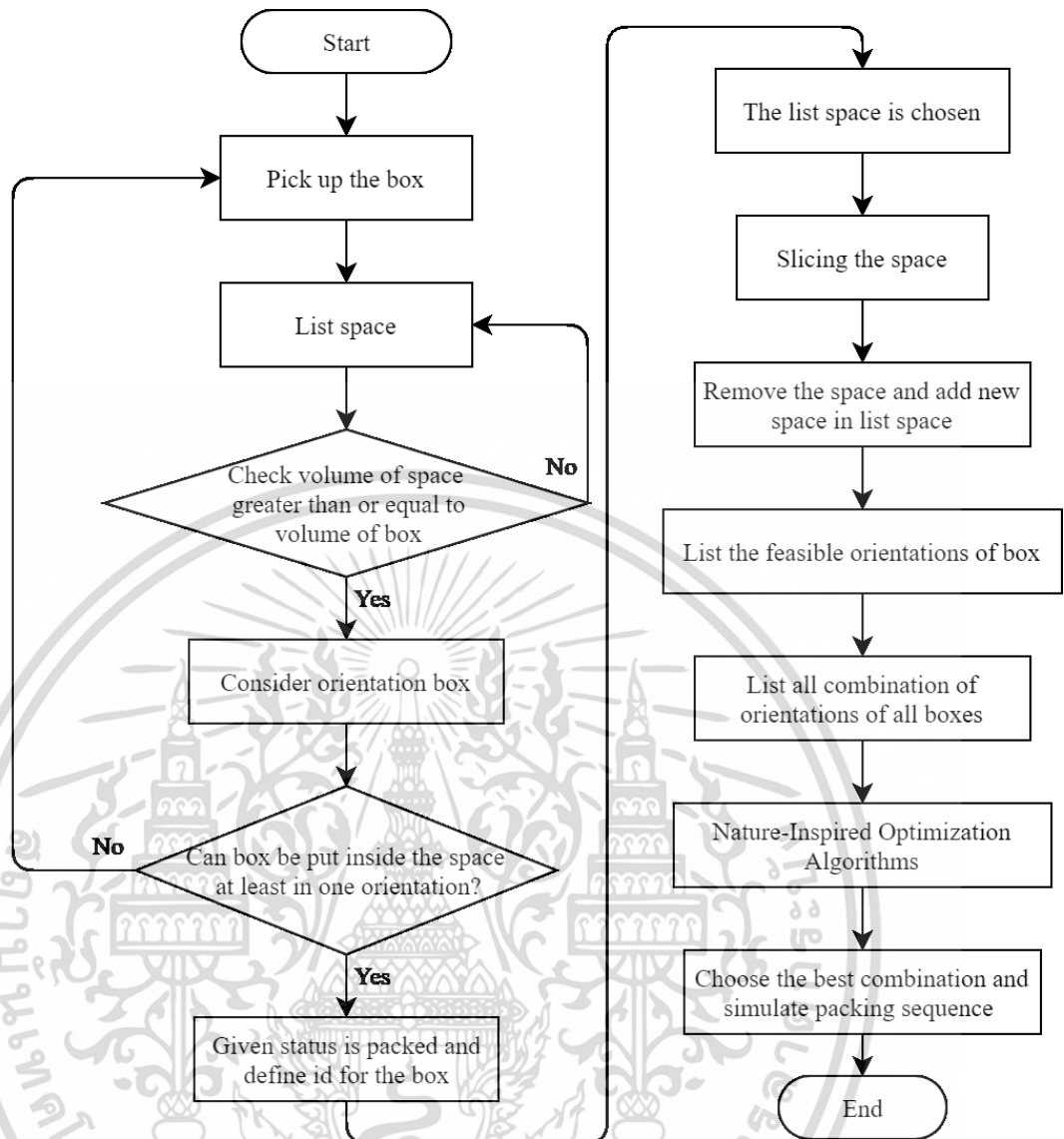
ตารางที่ 3.1 รายละเอียดของกล่องแต่ละชนิด

Type	Size (cm ³)	Weight (kg)
S	30x20x10	≤ 5
M	40x30x20	≤ 10
L	50x40x30	≤ 15
XL	60x50x40	≤ 20
OVER	70x60x50	≤ 25

การพิจารณาเงื่อนไขและกระบวนการวางกล่องที่ทำการบรรจุทั้งหมด ต้องเรียงตามน้ำหนัก จากมากไปน้อย โดยกล่องที่มีน้ำหนักมากต้องถูกนำมาเรียงก่อนด้านล่าง เพื่อป้องกันการแตกหักของกล่อง ในขณะที่ทำการวางเรียงลงในตู้คอนเทนเนอร์ และหลังจากเรียงกล่องตามน้ำหนักแล้วทำการกำหนด id ให้กับแต่ละกล่อง เพื่อให้สอดคล้องกับที่กล่าวมาทั้งหมด จะทำการทดลองโดยจำลอง 12 สถานการณ์ โดยมีวัตถุประสงค์คือการหาจำนวนกล่องที่มากที่สุดที่สามารถบรรจุลงในตู้คอนเทนเนอร์ ตู้เดียวทั้งสองขนาด โดยแยกตามขนาดของกล่องแต่ละชนิด และคละขนาดของกล่อง

3.2 กลยุทธ์สำหรับการบรรจุกล่องสินค้า

ข้อบ่งชี้ที่เกี่ยวข้องกับการบรรจุสินค้าประกอบด้วยตู้คอนเทนเนอร์ กล่องและสเปซ (Space) ซึ่งตู้คอนเทนเนอร์ และกล่องประกอบไปด้วยพารามิเตอร์ดังที่ได้กล่าวไปในหัวข้อก่อนหน้านี้แล้ว สเปซ ประกอบไปด้วยค่า Dimension Volume (X_{sw}, Y_{sh}, Z_{sl}) และ Dimension Position (X_s, Y_s, Z_s) ตำแหน่งเริ่มต้นเท่ากับ (0,0,0) และ Volume เริ่มต้นคือ Volume ของตู้คอนเทนเนอร์ กระบวนการทำงานแสดงดังรูปที่ 3.3



รูปที่ 3.3 แผนภาพกลยุทธ์สำหรับการบรรจุกล่องสินค้า

จากรูปที่ 3.3 สามารถอธิบายรายละเอียดในแต่ละขั้นตอนได้ดังนี้

- 1) ขั้นตอนที่ 1 : ทำการหยิบกล่องที่ได้เรียงลำดับตามน้ำหนักเอาไว้ก่อนแล้ว หลังจากนั้นเลือกสเปซที่มีพื้นที่น้อยที่สุดในการบรรจุกล่องดังกล่าวที่ (3.1)

$$S_{\min} = \min \sum_{j=0}^s X_{sw_j} \cdot Y_{sh_j} \cdot Z_{sl_j} \quad (3.1)$$

- 2) ขั้นตอนที่ 2 : การวางกล่อง โดยพิจารณาเงื่อนไขในการบรรจุกล่องลงในสเปซที่เลือก ซึ่งเงื่อนไขการบรรจุคือพื้นที่ในสเปซต้องมากกว่าหรือเท่ากับพื้นที่กล่อง เงื่อนไขคือ

$X_{sw_j}, Y_{sh_j}, Z_{sl_j}$ ของสเปซต้องมากกว่าหรือเท่ากับกล่อง มิเช่นนั้นกล่องอาจจะทะลุหรือเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์โดยไม่ผ่านการอนุมัติจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกนอกสเปซได้ และรวมไปถึงเงื่อนไขการหมุนกล่องทั้ง 6 ทาง เมื่อทำการหมุนกล่อง
กล่องยังต้องสามารถวางอยู่ภายในสเปซที่เลือกได้อย่างน้อย 1 ทาง แสดงดังสมการที่
(3.2) (3.3) และ (3.4) ตามลำดับ

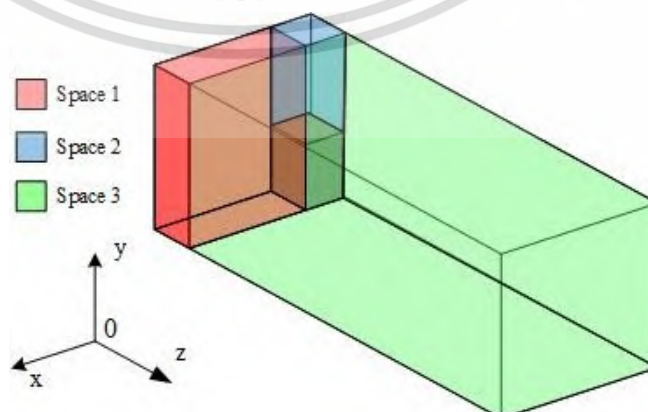
$$V_i = \begin{cases} 1, \sum X_{sw_i} \cdot Y_{sh_i} \cdot Z_{sl_i} \geq x_{w_i} \cdot y_{h_i} \cdot z_{l_i} \\ 0, otherwise \end{cases} \quad (3.2)$$

$$v_i = \begin{cases} 1, X_{sw_i} \geq x_{w_i} \cup Y_{sh_i} \geq y_{h_i} \cup Z_{sl_i} \geq z_{l_i} \\ 0, otherwise \end{cases} \quad (3.3)$$

$$R_i = \sum_{c=0}^5 r_c \quad (3.4)$$

จากเงื่อนไขของการหมุนกล่องให้ R_i คือ ผลรวมทศการหมุนของกล่องหากมีค่าเท่ากับ 0
แสดงว่าไม่สามารถวางกล่องลงในสเปซที่เลือกได้ เพราะหากวางกล่องลงในสเปซ ทำให้แนวกล่องทะลุ
หรือออกนอกขอบเขต ถ้า R_i มีค่ามากกว่า 1 หมายความว่า กล่องสามารถวางในสเปซที่เลือกได้และ
เปลี่ยนสถานะของกล่องเป็น 1 พร้อมทั้งกำหนด id ให้กับกล่อง

- 3) ขั้นตอนที่ 3 : กระบวนการตัดสเปซ ซึ่งจะทำการตัดสเปซที่เลือกโดยการตัดออกเป็น 3
ส่วนตามแกน x, y และ z ของกล่องแสดงดังรูปที่ 3.4 พื้นที่ของสเปซใหม่ที่ได้ต้องไม่
ซ้อนทับกัน โดยสเปซใหม่จะถูกเพิ่มลงในลิสต์สเปซและสเปซเก่าจะถูกนำออกจาก
ลิสต์สเปซส่วนอัลกอริทึมที่ใช้ตัดสเปซแสดงในรูปที่ 3.5



รูปที่ 3.4 พื้นที่ใหม่ที่ได้จากการตัดสเปซ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Algorithm : Slice Space Algorithm

```

For each axis do
  if (X axis)
    PositionX0 = abs(spacePosition0 + cargoVolume0)
    VolumeX0 = abs(spaceVolume0 - cargoVolume0)
    PositionX1 = spacePosition1
    VolumeX1 = spaceVolume1
    PositionX2 = spacePosition2
    VolumeX2 = cargoVolume2
  else if (Y axis)
    PositionY0 = spacePosition0
    VolumeY0 = cargoVolume0
    PositionY1 = spacePosition1 + cargoVolume1
    VolumeY1 = spaceVolume1 - cargoVolume1
    PositionY2 = spacePosition2
    VolumeY2 = cargoVolume2
  else
    PositionZ0 = spacePosition0
    VolumeZ0 = spaceVolume0
    PositionZ1 = spacePosition1
    VolumeZ1 = spaceVolume1
    PositionZ2 = abs(spacePosition2 + cargoVolume2)
    VolumeZ2 = abs(spaceVolume2 - cargoVolume2)
  End if
End for
if (accumulateVolumeX > 0)
  addnewSpace
End if
if (accumulateVolumeY > 0) {
  addnewSpace
End if
if (accumulateVolumeZ > 0) {
  addnewSpace
End if

```

รูปที่ 3.5 อัลกอริทึมที่ใช้สำหรับตัดสเปซ

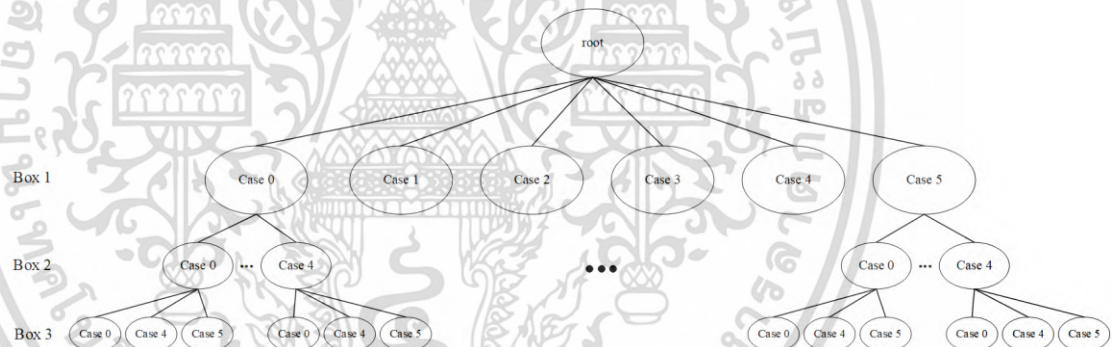
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการตัดสเปซของทุกแนวการวางกล่องเช่น กล่องที่หนึ่งมีแนวการวางทั้งหมด 6 ทาง ($R_1 = 6$) อยู่ในสเปซที่เลือก กล่องที่สอง $R_2 = 5$ และกล่องที่สาม $R_3 = 3$ แสดงในตารางที่ 3.2

ตารางที่ 3.2 ตัวอย่างแนวการวางกล่อง

id	Rotate Way						Total Way
	0	1	2	3	4	5	
box 1	1	1	1	1	1	1	6
box 2	1	1	1	1	1	0	5
box 3	1	0	0	0	1	1	3

- 4) ขั้นตอนที่ 4 : ทำการเก็บทุกแนวการหมุนของกล่องให้อยู่ในลิสต์ เมื่อได้แนวการวางของกล่องทั้งหมดแล้วทำการรวมวิธีการหมุนของกล่องเข้าด้วยกัน ดังนั้นจากตัวอย่างจะได้แนวการวางของกล่องทั้งหมด 90 ลิสต์ ($6 \times 5 \times 3$) แสดงดังรูปที่ 3.6



รูปที่ 3.6 ลิสต์แนวการวางทั้งหมด

3.3 กลยุทธ์สำหรับการบรรจุกล่องสินค้ากับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

การกำหนดค่าเริ่มต้นสำหรับ PSO หลังจากที่ได้ดำเนินการครบทั้ง 4 ขั้นตอนแล้วจึงเริ่มกำหนดค่าเริ่มต้น โดยในแต่ละลิสต์ที่ได้จากขั้นตอนที่ 4 หมายถึง อนุภาค (i) ซึ่งแต่ละอนุภาคกำหนดค่าเริ่มต้น แสดงดังตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 การกำหนดค่าเริ่มต้นสำหรับ PSO

Parameters	Value
particle_index	i
particle_position	volume_list
particle_best_position	particle_position
particle_velocity	random float number [0,1]
particle_fitness	คำนวณจากสมการที่ (3.5)
particle_best_fitness	null
number_of_particles	จำนวนอนุภาคทั้งหมด
speed_limit	9
number_of_iteration	10
fitness_threshold	0
inertia_weight	0.729
cognitive_weight (c_1)	1.49445
social_weight (c_2)	1.49445

โดยสมการสำหรับการคำนวณเพื่อหาค่าฟิตเนสแสดงดังสมการที่ (3.5)

$$particle_fitness = volume_container - \sum_{i=0}^N (x_{w_i} \cdot y_{h_i} \cdot z_{l_i}) \quad (3.5)$$

เมื่อกำหนดค่าเริ่มต้นให้แต่ละอนุภาคเรียบร้อยแล้ว อนุภาคทำการเปรียบเทียบเพื่อหาค่าฟิตเนสที่มากที่สุดในช่วง และนำมาใช้ในการปรับเปลี่ยนความเร็วและทิศทาง แสดงดังสมการที่ (3.6) และ (3.7) ตามลำดับ

$$v_i(t+1) = wv_i(t) + c_1r_1[x_i(t) - x_i(t)] + c_2r_2[g(t) - x_i(t)] \quad (3.6)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.7)$$

โดยที่

- i particle index
- w inertial coefficient
- c_1, c_2 acceleration coefficients ($0 \leq c_1, c_2 \leq 2$)
- r_1, r_2 ค่าสุ่ม ($0 \leq r_1, r_2 \leq 1$) ซึ่งถูกสุ่มใหม่ทุกครั้งที่มีการเปลี่ยนแปลงความเร็ว
- $v_i(t)$ ความเร็วของอนุภาคที่เวลา t
- $x_i(t)$ ตำแหน่งของอนุภาคที่เวลา t
- $x_i(t)$ ตำแหน่งที่ดีที่สุดของอนุภาค ที่เวลา t (pbest)
- $g(t)$ ตำแหน่งที่ดีที่สุดของฝูงที่เวลา t (gbest)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำซ้ำตามจำนวนรอบที่ได้กำหนดเอาไว้ใน `number_of_iterations` ซึ่งผลลัพธ์สุดท้ายคือ ปริมาตร (`swam_best_position`) พื้นที่ว่าง (`swam_best_fitness`) และลิสต์ที่เหลือพื้นที่ว่างน้อยที่สุดในตู้คอนเทนเนอร์ (`swam_best_index`) โดยอัลกอริทึมของ PSO ที่ใช้สำหรับการเลือกลิสต์แนวการวางกล่องสินค้าที่ดีที่สุด แสดงในรูปที่ 3.7

Algorithm : PSO Algorithm for Choosing the Best Combination List

```

For all particle p do
  initialParticle()
  if (swarm_best_fitness is null or particle p particle_fitness > swarm_best_fitness)
    swarm_best_index = p
    swarm_best_fitness = particle p particle_fitness
    swarm_best_position = particle p particle_position
  End if
End for
iteration()
Function iteration
  if (i < number_of_iterations and swarm_best_fitness > fitness_threshold
    For all particle p do
      particle p particle_velocity = UpdateVelocity()
      particle p particle_position = UpdatePosition()
      particle p particle_fitness = CalculateFitness()
      if (particle p particle_fitness > particle p particle_best_fitness
        particle p particle_best_fitness = particle p particle_fitness
        particle p particle_best_position = particle p particle_position
      End if
      if (particle p particle_fitness > swarm_best_fitness)
        swarm_best_index = p
        swarm_best_fitness = particle p particle_fitness
        swarm_best_position = particle p particle_position
      End if
    End for
    i++
  iteration()
else
  return [swarm_best_position , swarm_best_fitness, swarm_best_index]
End if
End function

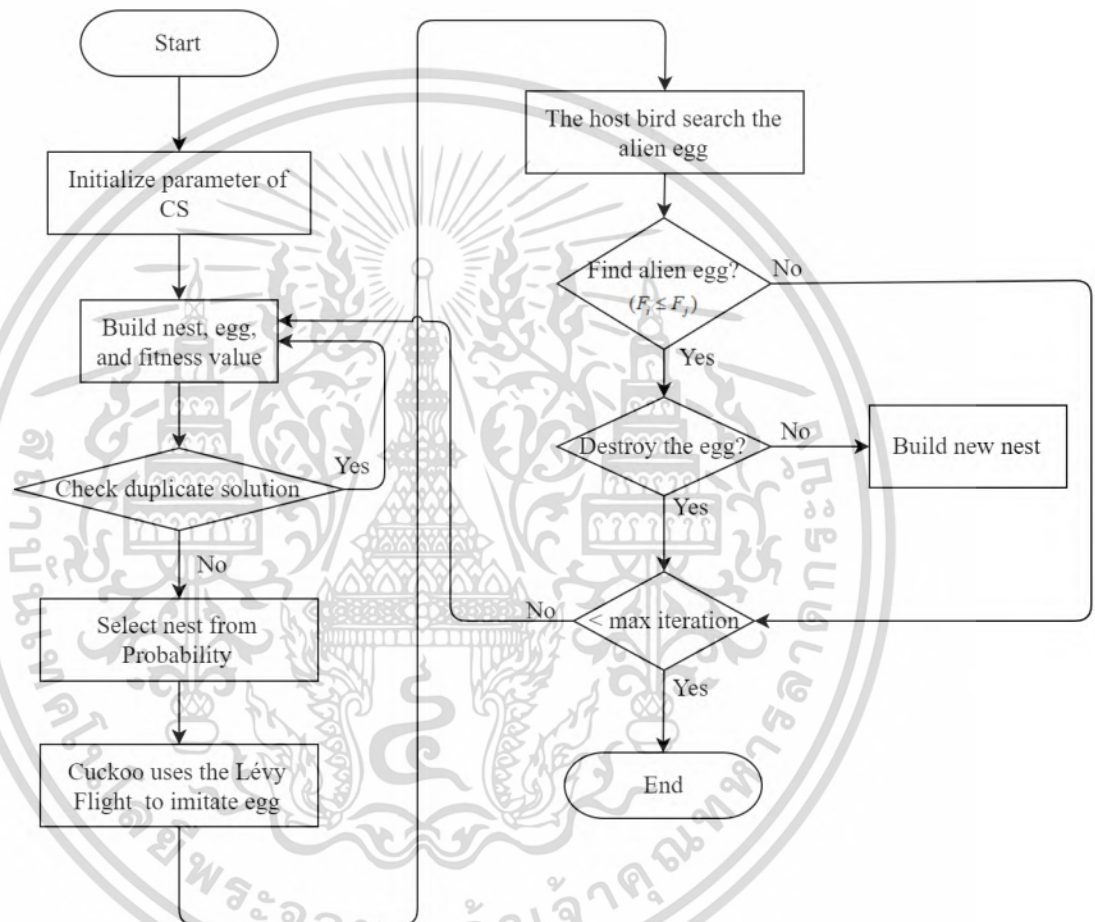
```

รูปที่ 3.7 อัลกอริทึมของ PSO ที่ใช้สำหรับการเลือกลิสต์แนวการวางกล่องสินค้าที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 กลยุทธ์สำหรับการบรรจุกล่องสินค้ากับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบนกกาเหว่า

ในหัวข้อนี้กล่าวถึงการใช้ CS เพื่อหาแนวการบรรจุกล่องที่เหมาะสมที่สุด โดยการกำหนดนิยาม เงื่อนไขและกลยุทธ์สำหรับการบรรจุกล่องสินค้า พิจารณาตามหัวข้อ 3.1 และ 3.2 แผนภาพการทำงานของวิธีการค้นหาแบบนกกาเหว่า แสดงในรูปที่ 3.8



รูปที่ 3.8 แผนภาพการทำงานของวิธีการค้นหาแบบนกกาเหว่า

จากรูปที่ 3.8 สามารถอธิบายรายละเอียดในแต่ละขั้นตอนได้ดังนี้

- 1) ขั้นตอนที่ 1 : เริ่มต้นด้วยการขั้นตอนการกำหนดค่าพารามิเตอร์เริ่มต้นเช่น จำนวนรอบการคำนวณ (I) จำนวนไข่ในแต่ละรัง (NE) อัตราการทิ้งรังของนกเจ้าของรัง (p_a) จำนวนการเลียนแบบไข่นกเจ้าของรัง Lévy Flights (λ) และจำนวนรัง (N) โดยแต่ละลิสต์แนวการวางของกล่องหมายถึง ไข่ (i) (จากหัวข้อที่ 3.2) ดังนั้นการกำหนดค่า

เริ่มต้นสำหรับ CS แสดงดังตารางที่ 3.4 ดังนี้
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 การกำหนดค่าเริ่มต้นสำหรับ CS

Parameters	Value
egg	i
iteration	100
egg_in_nest	1
probability_abandon_nest	0.25
host_nests	15

- 2) ขั้นตอนที่ 2 : การสร้างรังและไข่ของนกเจ้าของรัง สามารถสร้างไข่แต่ละฟองโดยการสุ่มเลือกจากลิสต์การหมุนทั้งหมดจากหัวข้อที่ 3.2 ซึ่งแต่ละลิสต์แทนไข่ของนกเจ้าของรัง โดยจัดเก็บแต่ละลิสต์ และเริ่มสร้างไข่ฟองต่อไปจนครบ หลังจากนั้นนำไปวางในแต่ละรังจนครบตามจำนวนที่กำหนดจากขั้นตอนที่ 1
- 3) ขั้นตอนที่ 3 : ขั้นตอนการป้องกันการสร้างไข่เริ่มต้นซ้ำจากลิสต์ทั้งหมด เพื่อให้ไข่เริ่มต้นเกิดการกระจายค่าของคำตอบ เพื่อเพิ่มโอกาสสำหรับการสร้างคำตอบที่ดีขึ้น โดยการจัดเก็บลิสต์ที่ถูกเลือกในลักษณะของอาร์เรย์ ถ้าสร้างไข่ใหม่ต้องนำลิสต์ที่ถูกเลือกเข้าไปเช็คหาในลิสต์เคยถูกเลือกเพื่อกันไม่ให้คำตอบที่ซ้ำกัน กรณีซ้ำกันให้สุ่มเลือกลิสต์จากลิสต์การหมุนทั้งหมดเพื่อสร้างไข่เริ่มต้นขึ้นมาใหม่ ถ้าไม่ซ้ำกันให้เก็บลิสต์ที่ถูกเลือกนั้นลงในอาร์เรย์
- 4) ขั้นตอนที่ 4 : การคำนวณสมการวัตถุประสงค์ เมื่อได้ลิสต์แล้วก็นำกลับมาคำนวณปริมาตรที่ใช้ ซึ่งก็คือค่าฟิตเนส คำนวณได้จากสมการที่ (3.5)
- 5) ขั้นตอนที่ 5 : การเลือกรัง การกำหนดจำนวนรังซึ่งโดยปกติกำหนด 15 รัง ถึง 50 รัง [37] หมายถึงกลุ่มของคำตอบของสมการวัตถุประสงค์ ตัวอย่างเช่น รังนกมีไข่ของนกเจ้าของรัง 4 ฟอง นั้นหมายความว่าในกลุ่มของคำตอบนี้ (รัง) ประกอบด้วย ลิสต์ของกลุ่ม (ไข่) 4 ลิสต์ มีการคำนวณการใช้ปริมาตร 4 คำตอบ ซึ่งต้องไม่ตรงกับลิสต์การหมุนกลุ่มที่อยู่ในอาร์เรย์ที่เคยถูกเลือก และทำการเรียงคำตอบซึ่งหมายถึงการใช้ปริมาตรที่น้อยที่สุดจะถูกเรียงไว้ลำดับแรก โดยให้นกกาเหว่าที่ต้องการฝากไข่ไว้ที่รังของนกชนิดอื่นให้วิธีสุ่มเลือกรังที่จะไปวางไข่ ตามความน่าจะเป็นของไข่ (การใช้ปริมาตรที่น้อยที่สุด) ที่ดีที่สุดในแต่ละรังนั้น การคำนวณความน่าจะเป็นจากการเปลี่ยนผันให้ปริมาตรน้อยมีความน่าจะเป็นมาก แสดงดังสมการ (3.8) และตัวอย่างการเลือกรังสำหรับการวางไข่แสดงดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_n = \frac{V_n}{\sum_{n=1}^N V_n}; n=1,2,3,\dots,N \quad (3.8)$$



รูปที่ 3.9 กระบวนการเลือกครั้งสำหรับการวางไข่

- 6) ขั้นตอนที่ 6 : การทำ Lévy Flights (λ) เป็นขั้นตอนการเลียนแบบไข่ที่ดีที่สุดของรังนั้น ซึ่งทำได้โดยการนำไข่ที่ดีที่สุดเป็นโครงสร้างหลัก และทำการสุ่มแนวการหมุนของกล่อง แทนที่แนวการหมุนเดิมเพื่อเป็นคำตอบใหม่และคำนวณปริมาตรตามลำดับ กำหนดตำแหน่งสำหรับการสุ่มคือ 1-3 ตำแหน่ง ซึ่งรูปแบบคล้ายกับวิธีการมิวเตชันของวิธทางพันธุกรรม [38] โดยทำการสุ่มตำแหน่งมา 2 ตำแหน่ง ซึ่งทั้ง 2 ตำแหน่งจะถูกแทนที่ด้วยแนวการหมุนของกล่องแบบสุ่ม แสดงดังรูป 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ตำแหน่งสุ่มที่ 1		ตำแหน่งสุ่มที่ 2	
		↓		↓	
เส้นทางต้นแบบ	0	2	1	3	5
เส้นทาง Lévy Flights	0	0	1	3	1

รูปที่ 3.10 ขั้นตอนการทำ Lévy Flights (2) ของนกกาเหว่า

- 7) ขั้นตอนที่ 7 : นกเจ้าของรังค้นหาไข่แปลกปลอมในรัง เนื่องจากว่านกกาเหว่านำไข่มาฝากไว้ในรังและทำการเลียนแบบไข่ของนกเจ้าของรังดังรูปที่ 3.11 ทำให้ไข่ในรังมีจำนวนเพิ่มมากขึ้น นกเจ้าของรังพิจารณาจากค่าของสมการวัตถุประสงค์ว่าจะทำลายไข่ที่ให้ค่าผลเฉลี่ยต่ำสุดหรือทิ้งรังนั้นไปเพื่อสร้างรังใหม่



รูปที่ 3.11 รังของนกเจ้าของรังที่มีไข่แปลกปลอมของนกกาเหว่า

- 8) ขั้นตอนที่ 8 : ขั้นตอนของการนำผลเฉลี่ยที่ดีที่สุดจากไข่ทุกรังมาเก็บเป็นค่าผลเฉลี่ยที่ดีที่สุดตั้งแต่การคำนวณมา (Best so far) ต่อจากนั้นเช็คเงื่อนไขของรอบการคำนวณที่กำหนดไว้ว่าครบหรือไม่ ถ้ายังไม่ครบตามจำนวนที่กำหนดให้กลับไปทำซ้ำกระบวนการเดิม ถ้าครบแล้วให้เลือกคำตอบที่เป็นผลเฉลี่ยที่ดีที่สุดมาเป็นคำตอบของรอบการคำนวณนั้น อัลกอริทึมสำหรับ CS ที่ใช้สำหรับการเลือก-list แนวการวางกล่องสินค้าที่ดีที่สุดแสดงในรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Algorithm : CSO Algorithm for Choosing the Best Combination List

```

Initialize for parameter
Generate initial population of  $N$  host nests, eggs, and fitness value
for loop to check duplicate solution ( $l = 1, 2, 3, \dots, n$ )
  if (is_duplicate solution)
    Regenerate population of  $l$  host nests, eggs, and fitness value
  end if
end for loop
while (  $t < \text{iteration}$  )
  Choose a nest from maximum probability value (say,  $i$ )
  In Lévy Flights, Random 2 box in list for change orientations then add the alien egg
  into the nest and calculate fitness value  $F_j$ 
  if ( $F_i \leq F_j$ )
    Replace  $i$  by the new solution  $j$ 
  end if
  A fraction ( $p_a$ ) of worse nests are abandoned and new one are build
  Keep the best solution (nests)
  Pass the current best solution to next generation
end while

```

รูปที่ 3.12 อัลกอริทึมของ CS ที่ใช้สำหรับการเลือกलिस्टแนวการวางกล่องสินค้าที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัยและการอภิปรายผล

ในบทนี้จะกล่าวถึงผลการวิจัยและการอภิปรายผลของทั้ง 2 กลุ่มก็คือกลุ่มสำหรับการบรรจุกล่องสินค้ากับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค และกลุ่มสำหรับการบรรจุกล่องสินค้ากับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบนกกาเหว่า ตามที่ได้นำเสนอไปในหัวข้อก่อนหน้า

ในกรณีของการทดลองแบ่งตามขนาดของตู้คอนเทนเนอร์ทั้ง 2 ขนาด โดยใส่กล่องชนิดเดียวกัน และต่างชนิดกันลงในตู้คอนเทนเนอร์ ซึ่งกรณีของการทดลองสามารถแบ่งได้ทั้งสิ้น 12 กรณี แสดงดังตารางที่ 4.1

ตารางที่ 4.1 กรณีของการทดลองทั้งหมด

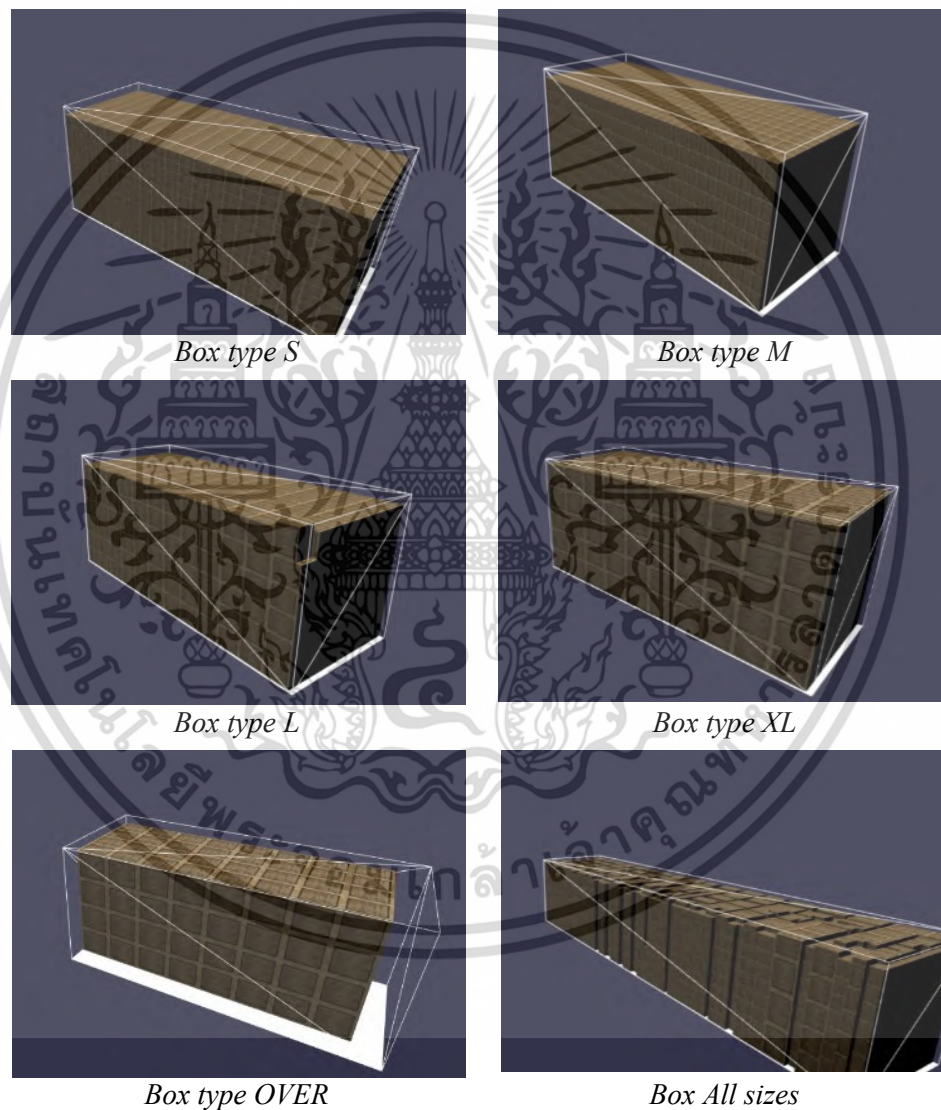
กรณีที่	ขนาดของกล่อง	ขนาดของคอนเทนเนอร์
1	S	20'
2	M	
3	L	
4	XL	
5	OVER	
6	All sizes	
7	S	40'
8	M	
9	L	
10	XL	
11	OVER	
12	All sizes	

ผลลัพธ์ที่ได้จากการทดลอง คือแนวการวางในแต่ละกล่อง และจำนวนกล่องที่มากที่สุดในแต่ละชนิดที่สามารถวางภายในตู้คอนเทนเนอร์ทั้ง 2 ขนาด ซึ่งหมายถึงการจัดสรรพื้นที่ว่างภายในตู้คอนเทนเนอร์ได้เหมาะสมที่สุด หลังจากนั้นนำแนวการวางกล่องของแต่ละกรณีมาจำลองในรูปแบบ 3 มิติ โดยใช้ Library ที่ชื่อว่า BabylonJS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ผลลัพธ์ของการทดลองด้วยขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

ผลลัพธ์ของการทดลองด้วยวิธี PSO สำหรับตู้คอนเทนเนอร์ขนาด 20' บรรจุกล่องขนาด S ได้ 5496 กล่องคิดเป็น 86.03% บรรจุกล่องขนาด M ได้ 1440 กล่องคิดเป็น 90.17% บรรจุกล่องขนาด L ได้ 576 กล่องคิดเป็น 90.17% บรรจุกล่องขนาด XL ได้ 300 กล่องคิดเป็น 93.92% บรรจุกล่องขนาด OVER ได้ 128 กล่องคิดเป็น 70.13% และบรรจุกล่องทุกขนาดได้ 996 กล่องคิดเป็น 93.09% ดังรูปที่ 4.1



รูปที่ 4.1 การจำลองการวางกล่องด้วยวิธี PSO ในตู้คอนเทนเนอร์ขนาด 20'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตู้คอนเทนเนอร์ขนาด 40' บรรจุกล่องขนาด S ได้ 11621 กล่องคิดเป็น 90.88% บรรจุกล่องขนาด M ได้ 2880 กล่องคิดเป็น 90.09% บรรจุกล่องขนาด L ได้ 1151 กล่องคิดเป็น 90.02% บรรจุกล่องขนาด XL ได้ 600 กล่องคิดเป็น 93.85% บรรจุกล่องขนาด OVER ได้ 272 กล่องคิดเป็น 74.45% และบรรจุกล่องทุกขนาดได้ 2653 กล่องคิดเป็น 92.13% ผลการทดลองทั้งหมดแสดงในตารางที่ 4.2 และ 4.3

ตารางที่ 4.2 ผลการทดลองด้วยวิธี PSO กับคอนเทนเนอร์ขนาด 20'

Type	Case					
	1	2	3	4	5	6
S	5496	0	0	0	0	780
M	0	1440	0	0	0	13
L	0	0	576	0	0	37
XL	0	0	0	300	0	64
OVER	0	0	0	0	128	99
Total box	5496	1440	576	300	128	993
Utilized area (%)	86.03	90.17	90.17	93.92	70.13	93.09

ตารางที่ 4.3 ผลการทดลองด้วยวิธี PSO กับคอนเทนเนอร์ขนาด 40'

Type	Case					
	1	2	3	4	5	6
S	11621	0	0	0	0	2120
M	0	2880	0	0	0	145
L	0	0	1151	0	0	105
XL	0	0	0	600	0	125
OVER	0	0	0	0	272	158
Total box	11621	2880	1151	600	272	2653
Utilized area (%)	90.88	90.09	90.02	93.85	74.45	92.13

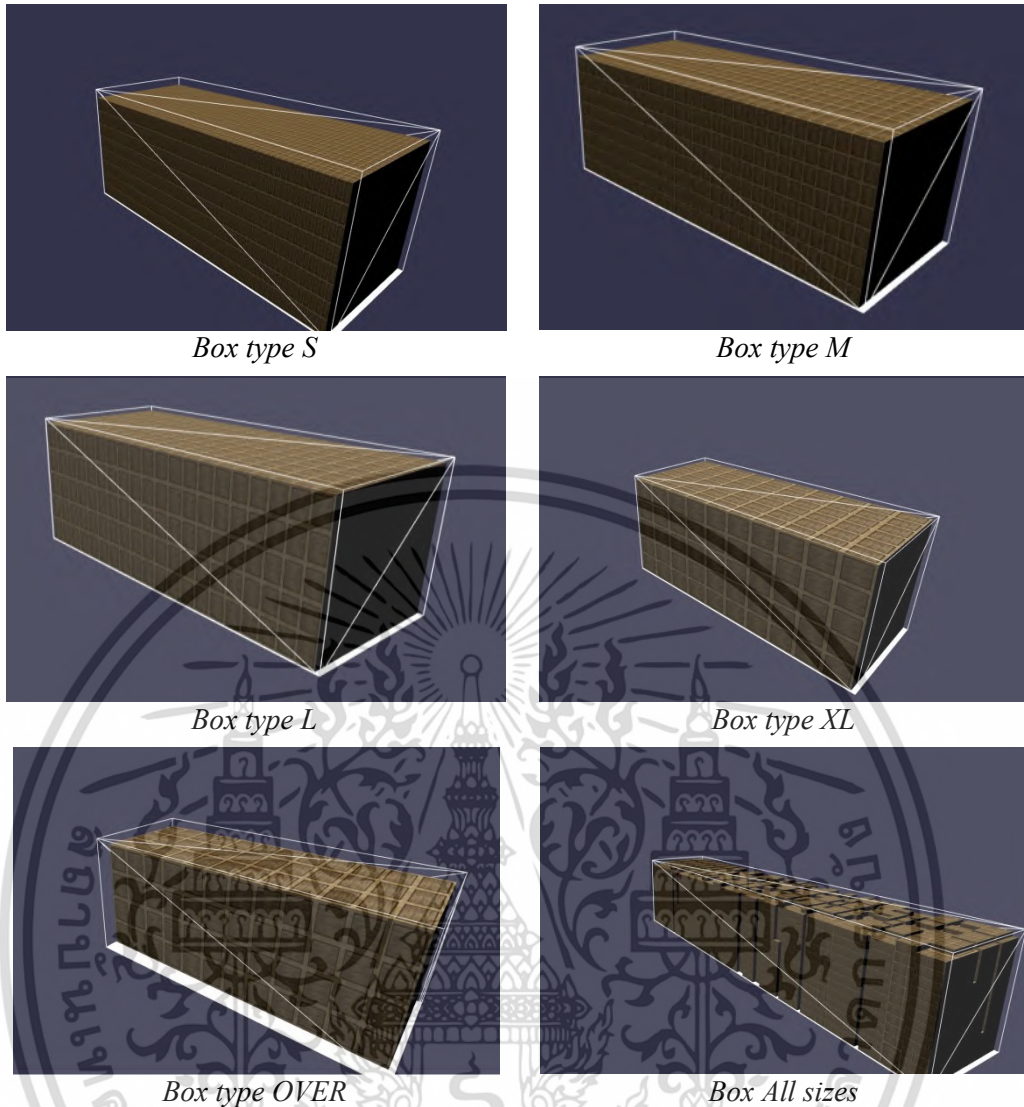
จากผลของการทดลองด้วยวิธี PSO โดยนำกล่องแต่ละขนาดวางเรียงในตู้คอนเทนเนอร์ที่มีขนาดความยาวต่างกัน 1 เท่า จะเห็นว่าจำนวนกล่องในแต่ละขนาดที่วางลงในตู้คอนเทนเนอร์ทั้งสองชนิดต่างกัน 1 เท่า ยกเว้นกล่องขนาด S และ OVER เนื่องจากการบรรจุกล่องขนาด S ลงในตู้คอนเทนเนอร์ขนาด 20' กล่องขนาด S มีขนาดที่เล็กเกินไป ทำให้เมื่อบรรจุกล่องขนาด S ลงในตู้คอนเทนเนอร์จึงเหลือเศษของพื้นที่มาก ทำให้ไม่สามารถบรรจุกล่องลงไปได้อีก แต่เมื่อนำกล่องขนาด S มาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรจุลงในคอนเทนเนอร์ขนาด 40' จะเห็นว่าจำนวนที่สามารถบรรจุลงไปได้มากกว่าประมาณ 1 เท่า เนื่องจากขนาดตู้คอนเทนเนอร์เพิ่มขึ้น ทำให้เหลือเศษของพื้นที่มากขึ้นจึงทำให้สามารถบรรจุกล่องขนาด S ลงไปได้มากขึ้นด้วย ส่วนการบรรจุกล่องขนาด OVER ลงในตู้คอนเทนเนอร์ขนาด 20' กล่องขนาด OVER มีขนาดใหญ่เกินไปจึงทำให้เหลือเศษของพื้นที่ที่ไม่พอให้กล่องขนาด OVER บรรจุลงไปได้ แต่เมื่อนำกล่องขนาด OVER มาบรรจุลงในตู้คอนเทนเนอร์ขนาด 40' จะเห็นว่าจำนวนที่สามารถบรรจุลงไปได้มากกว่า 1 เท่า เนื่องจากขนาดของตู้คอนเทนเนอร์เพิ่มขึ้น ทำให้เหลือเศษของพื้นที่มากขึ้นจึงทำให้สามารถบรรจุกล่องขนาด OVER ลงไปได้มากขึ้นด้วย

เมื่อนำจำนวนกล่องในแต่ละขนาดมาเปรียบเทียบเป็นเปอร์เซ็นต์การใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 20' และ 40' จะเห็นว่าเปอร์เซ็นต์การใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 20' ที่บรรจุกล่องขนาด S มีการใช้พื้นที่ภายในตู้คอนเทนเนอร์คิดเป็น 86.03 % เหตุผลที่ใช้พื้นที่ได้น้อยคือกล่องขนาด S มีขนาดที่เล็กเกินไปเมื่อบรรจุลงในตู้คอนเทนเนอร์แล้ว เหลือเศษของพื้นที่มากขึ้นจึงทำให้เปอร์เซ็นต์การใช้พื้นที่น้อยลงไปด้วย เมื่อนำกล่องขนาด S มาบรรจุลงในตู้คอนเทนเนอร์ขนาด 40' ทำให้เศษของพื้นที่ที่เหลือขยายออกตามขนาดของตู้คอนเทนเนอร์จึงทำให้เปอร์เซ็นต์การใช้พื้นที่เพิ่มขึ้นไปด้วย แต่กล่องขนาด OVER จากการบรรจุลงในตู้คอนเทนเนอร์ทั้ง 2 ขนาด มีเปอร์เซ็นต์การใช้พื้นที่เพียงประมาณ 72% เท่านั้น เนื่องจากขนาดของกล่องที่ใหญ่เกินไปทำให้เหลือช่องว่างแคบๆ ที่ด้านบนและด้านข้าง ส่วนกล่องขนาด M, L, XL และ All sizes มีเปอร์เซ็นต์การใช้พื้นที่ภายในตู้คอนเทนเนอร์ทั้งขนาด 20' และ 40' ประมาณ 90 % ซึ่งให้ผลลัพธ์เป็นที่น่าพอใจอย่างมาก

4.2 ผลลัพธ์ของการทดลองด้วยขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบนกกาเหว่า

ผลลัพธ์ของการทดลองด้วยวิธี CS สำหรับตู้คอนเทนเนอร์ขนาด 20' บรรจุกล่องขนาด S ได้ 5760 กล่องคิดเป็น 90.17% บรรจุกล่องขนาด M ได้ 1440 กล่องคิดเป็น 90.17% บรรจุกล่องขนาด L ได้ 600 กล่องคิดเป็น 93.92% บรรจุกล่องขนาด XL ได้ 300 กล่องคิดเป็น 93.92% บรรจุกล่องขนาด OVER ได้ 144 กล่องคิดเป็น 78.90% และบรรจุกล่องทุกขนาดได้ 1911 กล่องคิดเป็น 89.60% ดังรูปที่ 4.2



รูปที่ 4.2 การจำลองการวางกล่องด้วยวิธี CS ในตู้คอนเทนเนอร์ขนาด 20'

ตู้คอนเทนเนอร์ขนาด 40' บรรจุกล่องขนาด S ได้ 12000 กล่องคิดเป็น 93.85% บรรจุกล่องขนาด M ได้ 2880 กล่องคิดเป็น 90.09% บรรจุกล่องขนาด L ได้ 1200 กล่องคิดเป็น 93.85% บรรจุกล่องขนาด XL ได้ 600 กล่องคิดเป็น 93.85% บรรจุกล่องขนาด OVER ได้ 288 กล่องคิดเป็น 78.83% และบรรจุกล่องทุกขนาดได้ 4011 กล่องคิดเป็น 90.92% ผลการทดลองทั้งหมดแสดงในตารางที่ 4.4 และ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ผลการทดลองด้วยวิธี CS กับคอนเทนเนอร์ขนาด 20'

Type	Case					
	1	2	3	4	5	6
S	5760	0	0	0	0	1732
M	0	1440	0	0	0	13
L	0	0	600	0	0	37
XL	0	0	0	300	0	63
OVER	0	0	0	0	144	66
Total box	5760	1440	600	300	144	1911
Utilized area (%)	90.17	90.17	93.92	93.92	78.90	89.60

ตารางที่ 4.5 ผลการทดลองด้วยวิธี CS กับคอนเทนเนอร์ขนาด 40'

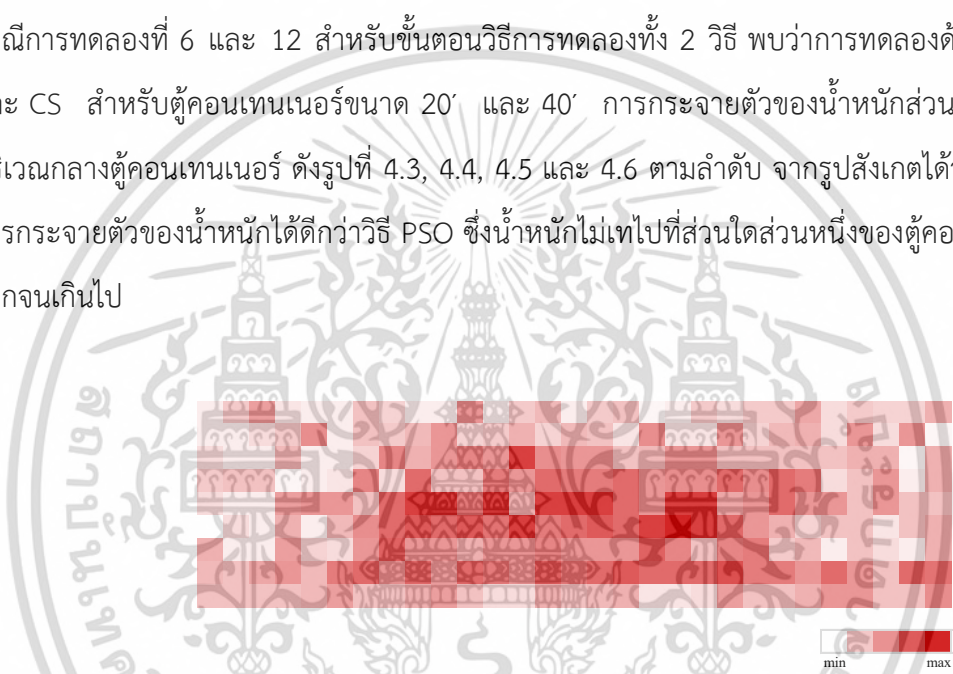
Type	Case					
	1	2	3	4	5	6
S	12000	0	0	0	0	3551
M	0	2880	0	0	0	130
L	0	0	1200	0	0	98
XL	0	0	0	600	0	103
OVER	0	0	0	0	288	129
Total box	12000	2880	1200	600	288	4011
Utilized area (%)	93.85	90.09	93.85	93.85	78.83	90.92

จากผลของการทดลองด้วยวิธี CS โดยนำกล่องแต่ละขนาดวางเรียงในตู้คอนเทนเนอร์ที่มีขนาดความยาวต่างกัน 1 เท้า จะเห็นว่าจำนวนกล่องในแต่ละขนาดที่วางลงในตู้คอนเทนเนอร์ทั้งสองชนิดต่างกันประมาณ 1 เท้าด้วย และเมื่อนำจำนวนกล่องในแต่ละขนาดมาเปรียบเทียบเป็นเปอร์เซ็นต์การใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 20' และ 40' จะเห็นว่าเปอร์เซ็นต์การใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 20' ที่บรรจุกล่องขนาด S, L และ OVER มีการใช้พื้นที่ภายในตู้คอนเทนเนอร์คิดเป็น 90.17 %, 93.92% และ 78.90% ตามลำดับ ซึ่งมีเปอร์เซ็นต์การใช้พื้นที่ที่มากกว่าวิธี PSO เนื่องจากว่าวิธี CS มีการปรับปรุงคำตอบที่ดีในปัจจุบัน ให้เป็นคำตอบที่ดีที่สุดที่เคยพบ โดยการสุ่มเลือกตำแหน่งจากลิสต์ของกล่องเพื่อเปลี่ยนแนวการหมุนของกล่องใหม่ โดยกล่องที่เปลี่ยนแนวการหมุนนั้นเป็นประเภทเดียวกันและมีขนาดเท่ากัน ซึ่งเป็นการปรับปรุงสเปซที่ใช้ให้ลดลง ทำให้เปอร์เซ็นต์การใช้พื้นที่ด้วยวิธี CS มากกว่า ส่วนกล่องขนาด OVER จากการบรรจุลงในตู้คอนเทน

เนอร์ทั้ง 2 ขนาด มีเปอร์เซ็นต์การใช้พื้นที่เพียงประมาณ 78% เท่านั้น เนื่องจากขนาดของกล่องที่ใหญ่เกินไปทำให้เหลือช่องว่างแคบๆ ที่ด้านบนและด้านข้าง

สำหรับกรณี All sizes ด้วยวิธี PSO มีเปอร์เซ็นต์การใช้พื้นที่ภายในตู้คอนเทนเนอร์ทั้งขนาด 20' และ 40' อยู่ที่ประมาณ 92 % ซึ่งมากกว่าวิธี CS เนื่องจากว่าวิธี PSO ทำการบรรจุกล่องโดยการเปรียบเทียบกับกล่องข้างๆ ที่มีประเภทและขนาดที่ต่างกัน ทำให้ในแต่ละครั้งที่มีการบรรจุกล่อง มีการเปรียบเทียบกับกล่องอื่นๆ ที่อยู่ภายในลิสต์ตลอด ส่งผลให้มีการจัดสรรพื้นที่ได้ดีกว่า

นอกจากนี้ยังเพิ่มการสังเกตเรื่องการกระจายน้ำหนักของกล่องภายในตู้คอนเทนเนอร์ จากกรณีการทดลองที่ 6 และ 12 สำหรับขั้นตอนวิธีการทดลองทั้ง 2 วิธี พบว่าการทดลองด้วยวิธี PSO และ CS สำหรับตู้คอนเทนเนอร์ขนาด 20' และ 40' การกระจายตัวของน้ำหนักส่วนใหญ่จะอยู่ในบริเวณกลางตู้คอนเทนเนอร์ ดังรูปที่ 4.3, 4.4, 4.5 และ 4.6 ตามลำดับ จากรูปสังเกตได้ว่าวิธี CS มีการกระจายตัวของน้ำหนักได้ดีกว่าวิธี PSO ซึ่งน้ำหนักไม่เทไปที่ส่วนใดส่วนหนึ่งของตู้คอนเทนเนอร์มากนักเกินไป

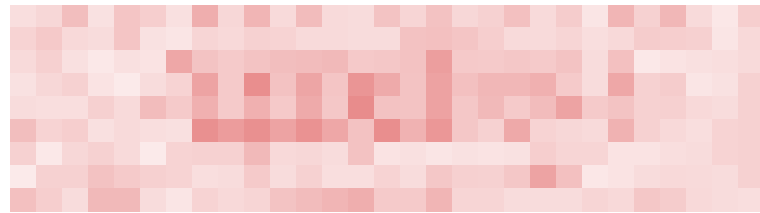


รูปที่ 4.3 การกระจายตัวของน้ำหนักกล่องภายในตู้คอนเทนเนอร์ขนาด 20' ด้วยวิธี PSO



รูปที่ 4.4 การกระจายตัวของน้ำหนักกล่องภายในตู้คอนเทนเนอร์ขนาด 40' ด้วยวิธี PSO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



min max

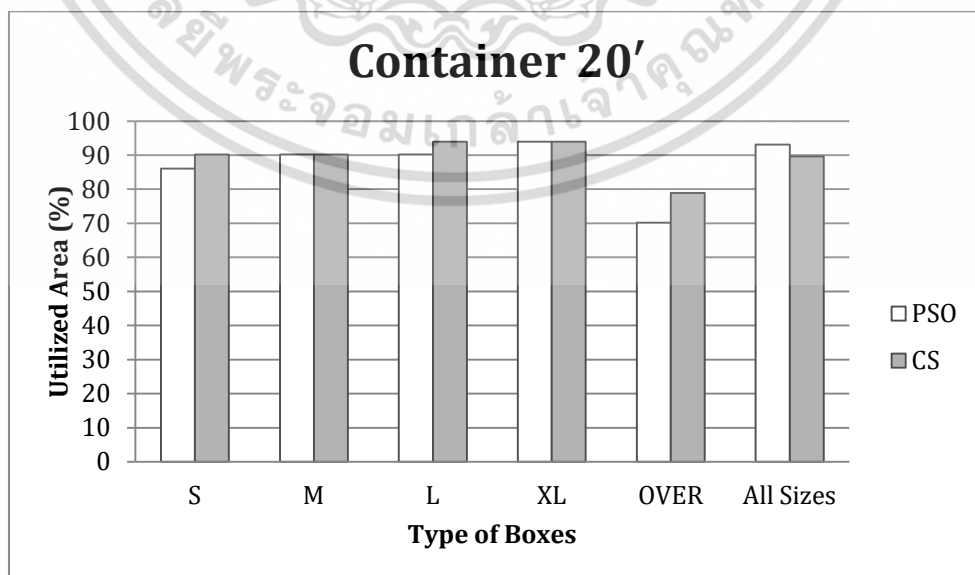
รูปที่ 4.5 การกระจายตัวของน้ำหนักกล่องภายในตู้คอนเทนเนอร์ขนาด 20' ด้วยวิธี CS



min max

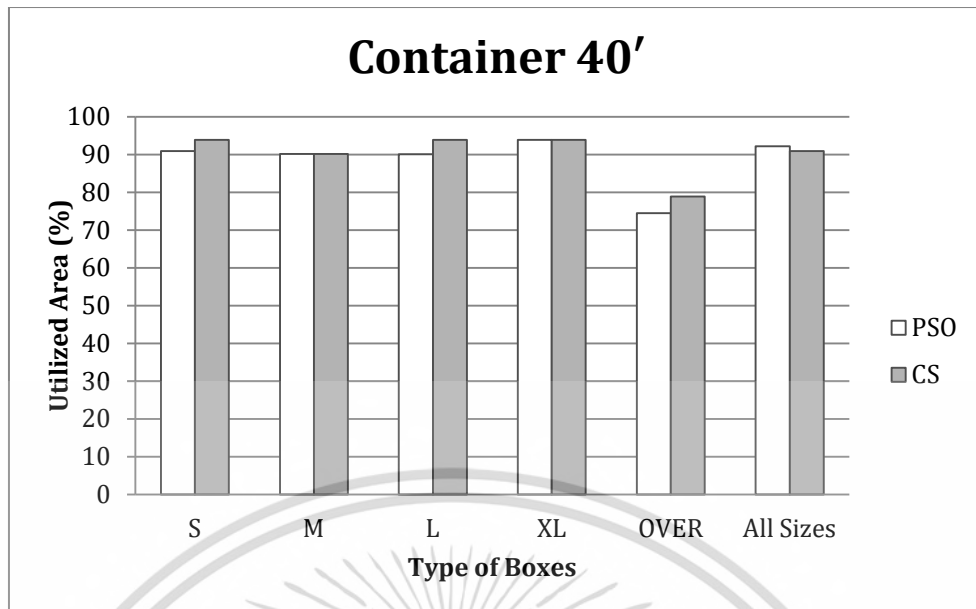
รูปที่ 4.6 การกระจายตัวของน้ำหนักกล่องภายในตู้คอนเทนเนอร์ขนาด 40' ด้วยวิธี CS

การเปรียบเทียบเปอร์เซ็นต์การใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 20' และ 40' ด้วยวิธี PSO และ CS โดยวิธี CS จำนวนกล่อง และเปอร์เซ็นต์การใช้พื้นที่ของกล่องขนาด S, L และ OVER มีเปอร์เซ็นต์การใช้พื้นที่ที่มากกว่าวิธี PSO ส่วนขนาด M และ XL มีเปอร์เซ็นต์การใช้พื้นที่เท่ากับวิธี PSO และส่วนขนาด All sizes วิธี CS มีเปอร์เซ็นต์การใช้พื้นที่ที่น้อยกว่าวิธี PSO ซึ่งแสดงดังรูปที่ 4.3 และ 4.4 ตามลำดับ

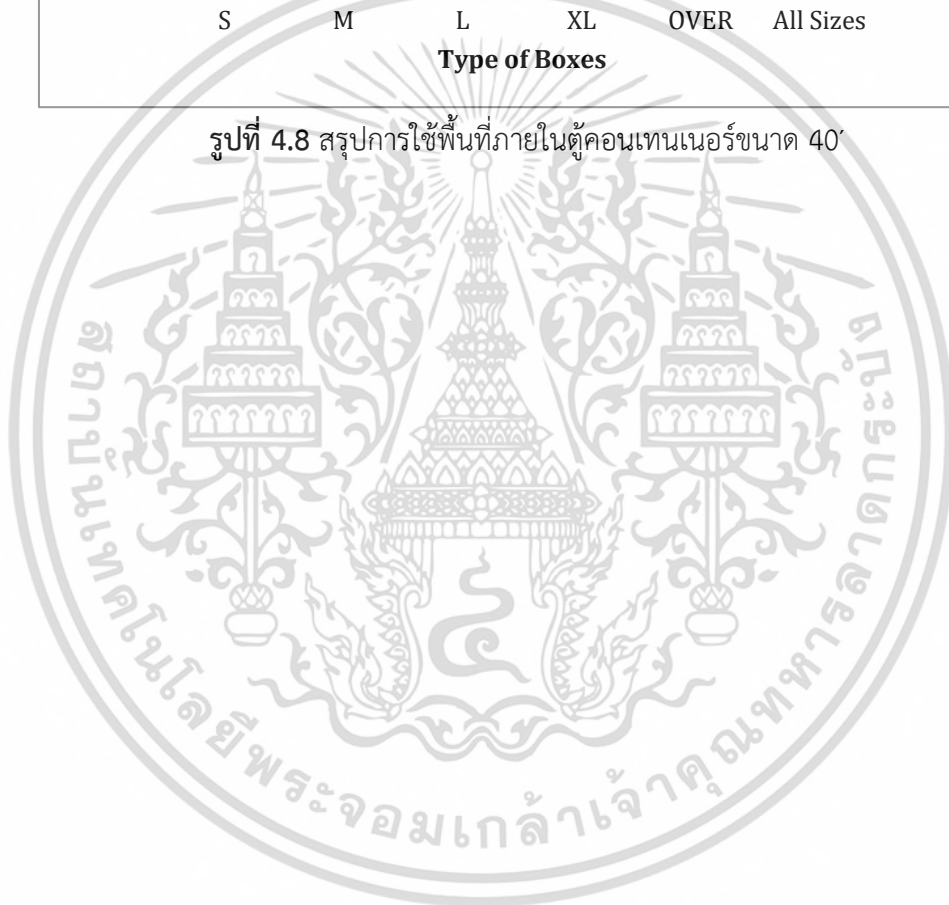


รูปที่ 4.7 สรุปการใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 20'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงพาณิชย์เท่านั้น มิใช่เอกสารที่เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 สรุปการใช้พื้นที่ภายในตู้คอนเทนเนอร์ขนาด 40'



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้เป็นการแก้ปัญหาการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ตู้เดียวแบบ 3 มิติ (3D-BPP) ใช้กลยุทธ์สำหรับการจัดตำแหน่งการวางสินค้าโดยปราศจากรูปแบบการจัดวาง (Layout) เพื่อลดพื้นที่ว่างระหว่างรูปแบบการจัดวาง โดยใช้วิธี PSO และ CS เพื่อหาดำแหน่ง รูปแบบการวางของกล่อง และจำนวนกล่องที่ถูกบรรจุลงในตู้คอนเทนเนอร์ตู้เดียวได้มากที่สุดโดยให้เหลือพื้นที่ว่างน้อยที่สุด ซึ่งเงื่อนไขที่นำมาพิจารณา ประกอบด้วยน้ำหนัก แนวการหมุนของกล่อง การวางกล่องรูปทรงสี่เหลี่ยมลงในตู้คอนเทนเนอร์ กล่องที่วางต้องไม่ทะลุหรือออกนอกตู้คอนเทนเนอร์ ไม่ซ้อนทับกัน และการวางกล่องต้องวางขนานไปกับตู้คอนเทนเนอร์หรือกล่องอื่นๆ หลังจากนั้นนำผลลัพธ์ที่ได้ของแต่ละวิธีมาเปรียบเทียบประสิทธิภาพการจัดวางสินค้าต่อการใช้พื้นที่ภายในตู้คอนเทนเนอร์ได้อย่างเหมาะสมมากที่สุด และจำลองการจัดวางสินค้าแต่ละวิธีด้วยภาพ 3 มิติ ซึ่งในการทดลองแบ่งกรณีการทดลองตามขนาดของกล่องและตู้คอนเทนเนอร์ออกเป็น 12 กรณี โดยการจำลองวางกล่องหลายชนิดลงในตู้คอนเทนเนอร์ที่มีหลายขนาด ซึ่งผลลัพธ์ที่ได้จากการทดลองโดยพิจารณาจากการใช้พื้นที่ภายในตู้คอนเทนเนอร์ด้วยวิธี PSO สำหรับกล่องขนาด M, L, XL และ All sizes ที่บรรจุภายในตู้คอนเทนเนอร์ทั้งขนาด 20' และสำหรับกล่องขนาด S, M, L, XL และ All sizes ที่บรรจุภายในตู้คอนเทนเนอร์ทั้งขนาด 40' มีเปอร์เซ็นต์การใช้พื้นที่มากกว่า 90 % ส่วนการใช้พื้นที่ภายในตู้คอนเทนเนอร์ด้วยวิธี CS สำหรับกล่องขนาด S, M, L, XL และ All sizes ที่บรรจุภายในตู้คอนเทนเนอร์ทั้งขนาด 20' และ 40' มีเปอร์เซ็นต์การใช้พื้นที่มากกว่า 90 % ซึ่งให้ผลลัพธ์เป็นที่น่าพอใจอย่างมาก และเมื่อเปรียบเทียบการวางกล่องของทั้ง 2 วิธี พบว่าวิธี CS เหมาะสำหรับการใช้วางกล่องชนิดเดียวกัน โดยสังเกตจากการวางกล่องขนาด S, L และ OVER ให้ผลลัพธ์ที่ดีกว่าวิธี PSO ทั้งขนาดตู้คอนเทนเนอร์ 20' และ 40'

5.2 ข้อเสนอแนะ

จากการศึกษาวิทยานิพนธ์ที่ได้นำเสนอนี้ ถึงแม้ว่าการบรรจุผลิตภัณฑ์ลงในตู้คอนเทนเนอร์ที่ได้นำเสนอนั้นสามารถทำงานได้ตามวัตถุประสงค์ แต่ละกลยุทธ์มีข้อดีที่สามารถนำไปใช้ในการจัดวางสินค้า เพื่อให้เกิดประสิทธิภาพต่อการขนส่งสินค้า ซึ่งเป็นการช่วยลดต้นทุนและเวลาในการขนส่ง โดยเงื่อนไขที่นำมาพิจารณาเป็นเงื่อนไขเพื่อการแก้ปัญหาในกรณีเฉพาะ ภายใต้สถานการณ์ที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่านั้น ซึ่งเงื่อนไขที่กำหนดในการทดลองไม่ยืดหยุ่นต่อการนำไปใช้จริงในเรื่องของขนาดสินค้า เพราะในการขนส่งจริงนั้น สินค้ามีมากมายหลายประเภทที่มีขนาดแตกต่างกัน โดยในอนาคตสามารถเพิ่มความยืดหยุ่น โดยปรับเปลี่ยนประเภทของสินค้าเพื่อให้เหมาะสมกับสถานการณ์ที่หลากหลายมากยิ่งขึ้น จำนวนภาชนะที่ใช้อาจเป็นแบบตู้คอนเทนเนอร์แบบหลายตู้ และอาจเพิ่มเงื่อนไขในการพิจารณาเรื่องการขนส่งสินค้าให้ถึงจุดหมายตามใบสั่งซื้อสินค้า ซึ่งการเรียงลำดับของสินค้ามีความสำคัญต่อการหยิบสินค้าออกจากตู้คอนเทนเนอร์ โดยสินค้าที่นำส่งก่อนต้องอยู่ใกล้กับประตูตู้คอนเทนเนอร์และสินค้าที่นำส่งในลำดับหลังต้องอยู่ภายในตู้คอนเทนเนอร์เพื่อทำให้การขนส่งสินค้าเกิดประสิทธิภาพสูงสุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Bischoff, E. and Ratcliff, M. 1995. "Issues in the development of approaches to container loading." *Omega*. vol. 23 no. 4 : pp. 377-390.
- [2] Bischoff, E. Janetz, F. and Ratcliff, M. 1995. "Loading pallets with non-identical items." *European Journal of Operational Research*. vol. 84 no. 3 : pp. 681-692.
- [3] Lim, A. Rodrigues, B. and Wang, Y. 2003. "A multi-faced buildup algorithm for three-dimensional packing problems." *Omega*. vol. 31 no. 6 : pp. 471-481.
- [4] Pisinger, D. 1999. "A Tree Search Heuristic for the Container Loading Problem." *Ricerca Operativa*. vol. 28 : pp. 31-48.
- [5] Fanslau, T. and Bortfeldt, A. 2010. "A Tree Search Algorithm for Solving the Container Loading Problem." *INFORMS Journal on Computing*. vol. 22 no. 2 : pp. 222-235.
- [6] Eley, M. 2002. "Solving container loading problems by block arrangement." *European Journal of Operational Research*. vol. 141 no. 2 : pp. 393-409.
- [7] Morabito, R. and Arenalest, M. 1994. "An AND/OR-graph Approach to the Container Loading Problem." *International Transactions in Operational Research*. vol. 1 no. 1 : pp. 59-73.
- [8] Gehring, H. and Bortfeldt, A. 2002. "A Parallel Genetic Algorithm for Solving the Container Loading Problem." *International Transactions in Operational Research*. vol. 9 no. 4 : pp. 497-511.
- [9] Gonçalves, J. and Resende, M. 2012. "A parallel multi-population biased random-key genetic algorithm for a container loading problem." *Computers & Operations Research*. vol. 39 no. 2 : pp. 179-190.
- [10] Bortfeldt, A. and Gehring, H. 1998. "Applying Tabu Search to Container Loading Problems." *Operations Research Proceedings 1997*. pp. 533-538.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [11] Bortfeldt, A. Gehring, H. and Mack, D. 2003. "A parallel tabu search algorithm for solving the container loading problem." *Parallel Computing*. vol. 29 no. 5 : pp. 641–662.
- [12] Levinson, M. 2006. **The box : how the shipping container made the world smaller and the world economy bigger.** New Jersey : Princeton University Press.
- [13] Kemme, N. 2013. **Design and Operation of Automated Container Storage Systems.** Heidelberg : Physica-Verlag HD.
- [14] กมลชนก สุทธิวาทนฤพุดิ. 2540. หน้า 82. ใน **กลยุทธ์บริหารธุรกิจการขนส่งทางเรือ.** จุฬาลงกรณ์มหาวิทยาลัย.
- [15] Pisinger, D. 1999. "A Tree Search Heuristic for the Container Loading Problem." *Ricerca Operativa*. vol. 28 : pp. 31-48.
- [16] Chien, C. and Wu, W. 1999. "A framework of modularized heuristics for determining the container loading patterns." *Computers & Industrial Engineering*. vol. 37 no. 1-2 : pp. 339-342.
- [17] Martello, S. Pisinger, D. and Vigo, D. 2000. "The Three-Dimensional Bin Packing Problem." *Operations Research*. vol. 48 no. 2 : pp. 256–267.
- [18] Bortfeldt, A. and Gehring, H. 2001. "A hybrid genetic algorithm for the container loading problem." *European Journal of Operational Research*. vol. 131 no. 1 : pp. 143-161.
- [19] Bischoff, E. 2006. "Three-dimensional packing of items with limited load bearing strength." *European Journal of Operational Research*. vol. 168 no. 3 : pp. 952–966.
- [20] Wu, Y. Li, W. Goh, M. and Souza, R. D. 2010. "Three-dimensional bin packing problem with variable bin height." *European Journal of Operational Research*. vol. 202 no. 2 : pp. 347–355.
- [21] Kang, K. Moon, I. and Wang, H. 2012. "A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem." *Applied Mathematics and Computation*. vol. 219 no. 3 : pp. 1287– 1299.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [22] Kimpan, W. and Tantawarak, N. 2013. “Simulation model of space optimization in container arranging and sequencing for logistics.” *IADIS International Conference Information Systems PostImplementation and Change Management, Czech Republic.* pp. 5157.
- [23] Yang, X. S. 2014. **Nature-Inspired Optimization Algorithms.** Waltham : Elsevier Science.
- [24] Domingo, B. M. Ponnambalam, S. and Kanagaraj, G. 2012. “Particle Swarm Optimization for the single container loading problem.” *2012 IEEE International Conference on Computational Intelligence and Computing Research.*
- [25] Shingade, V. Kumbhar, S. and Khandve, S. 2016 “2D Packaging Problem using Swarm Intelligence.” *International Journal of Innovative Research in Computer and Communication Engineering.* vol. 4 no. 3.
- [26] François, S. Sebastien, S. Dominique, C. Thomas, B. and Yann, M. 2017. **A new loading problem: product size reduction.** [Online]. Available : <https://cfm2017.sciencesconf.org/133143/document>. [Accessed: Aug-2018].
- [27] Rajabioun, R. 2011. “Cuckoo Optimization Algorithm.” *Applied Soft Computing* 11(8) : pp.5508-5518.
- [28] Mareli, M. and Twala, B. 2018. “An adaptive Cuckoo search algorithm for optimisation.” *Applied Computing and Informatics.* 14(2) : pp.107-115.
- [29] Rhee, I. Shin, M. Hong, S. Lee, K. Kim, S. and Chong, S. 2011. “On the Levy-Walk Nature of Human Mobility.” *IEEE/ACM Transactions on Networking.* 19(3) : pp.630-643.
- [30] Shlesinger, M. F. Klafter, J. and Wong, Y. M. 1982 “Random walks with infinite spatial and temporal moments.” *J. Stat. Phys.* vol. 27 : pp. 499– 512.
- [31] Reynolds, A. and Frye, M. 2007. “Free-Flight Odor Tracking in *Drosophila* Is Consistent with an Optimal Intermittent Scale-Free Search.” *PLoS ONE.* 2(4) :

- [32] Brown, C., Liebovitch, L. and Glendon, R. 2006. "Lévy Flights in Dobe Ju/'hoansi Foraging Patterns." *Human Ecology*. 35(1) : pp.129-138.
- [33] Xin-She, Y. and Deb, S. 2014. "Cuckoo Search via Levy flights." *In World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009*. pp. 210-214.
- [34] Layeb, A. and Boussalia, S. 2012. "A Novel Quantum Inspired Cuckoo Search Algorithm for Bin Packing Problem." *International Journal of Information Technology and Computer Science*. 4(5) : pp.58-67.
- [35] Yesodha, R. 2012. "A Comparative Study On Heuristic Procedures To Solve Bin Packing Problems." *International Journal in Foundations of Computer Science & Technology*. 2(6) : pp.37-49.
- [36] Zendaoui, Z., & Layeb, A. 2016. "Adaptive Cuckoo Search Algorithm for the Bin Packing Problem." *Modelling and Implementation of Complex Systems*. 107–120.
- [37] Glover, F. 1986. "Future paths for integer programming and link to artificial intelligence." *Computers and Operation Research*. 13 : 533-49.
- [38] Sujaree, K. 2017. "Blood vehicle routing network using chemical reaction optimization algorithm." *In Mahidol University editors. The 2017 Technology Innovation Management and Engineering Science international conference*. p. 189-95.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Applied Particle Swarm Optimization in Solving Container Loading Problem for Logistics

Sasithorn Koonsintananan
 Department of Computer Science, Faculty of Science
 King Mongkut's Institute of Technology Ladkrabang
 Bangkok, Thailand
 58605107@kmitl.ac.th

Warangkhan Kimpan, Member, IEEE
 Department of Computer Science, Faculty of Science
 King Mongkut's Institute of Technology Ladkrabang
 Bangkok, Thailand
 warangkhan.ki@kmitl.ac.th

Abstract— Container loading problem (CLP) is a major problem for logistics in distributing cargoes to customers. As cost and time savings are important in shipping management, the process of cargo handling is essential. Cargoes must be effectively packed into a container. This paper, therefore, proposes container loading problem solving using a strategy for positioning non placement heuristics to reduce space between different buildings. Particle Swarm Optimization (PSO) is used to select the orientation of boxes to be packed into a single container to minimize the space which simulated in 3D. The condition for this solution is packing must be placed within the container with no part of the box overhanging the container, and the boxes must be placed parallel to the container surface area or other boxes. Weight and orientation of the box condition are also considered. The experiments are 12 cases. Different sizes of boxes and different sizes of containers are used. The experimental results are satisfied and it can be concluded that varied sizes of boxes are applicable for the space utilization in the container.

Keywords—Container Loading Problem, Logistics, Particle Swarm Optimization, Space Optimization

I. INTRODUCTION

The rapidly growing economy has led the way to widespread development using containers for merchandise distribution. Thus, container logistics plan has become one of the key factors for the arrangement of shipping containers within a vessel or transport mode. The logistics plan may include the order in which containers must be loaded, along with the cargo handling which is the effective way to place cargoes into containers. For effective shipping, the packing of cargoes in containers need to be improved in terms of space utilization with an optimized container load plan prior to the cargo delivery. This will help reducing the transportation cost and time.

Previous researchers have proposed various solutions for packing problems. The packing is classified into 3 groups: the first is conventional heuristics based on conditions and orientation, for example orientation of cargo using a Greedy algorithm and improving the efficiency of a local search algorithm [1-3]. The second is the tree search and graph search method [4-7], which will take a long time if there is a wide search space. The third group is the metaheuristics such as Genetic algorithm and Tabu search [8-11]. Many researchers used this method because it is flexible and fast.

Researchers generally focus on the conditions of container loading for orientation. The main conditions are considering the weight of a container, overlapping of cargoes, fragility or pressure of cargoes, a number of

different containers (single or multiple container), a number of dimensions (one, two, or three dimensions). Additionally, the use of layout in the cargo placement such as wall building, stack, horizontal layer building, block building, guillotine cutting also led to the Container Loading Problem (CLP). However, the problem solving depends on each researcher. Most researchers proposed the cargo placement and adjusted the appropriate parameters for their algorithms and the result is space allocation to reduce transportation cost and time.

This paper proposes a solution for CLP which can arrange the cargo without placement layout. In order to reduce space between different layouts, the conditions focus on weight, pressure, and orientation of cargoes in containers. Particle Swarm Optimization (PSO) was used to choose suitable orientation of each cargo to be packed in a single container. Then, BabylonJS was used to render the cargo placement using 3D simulation.

The arrangement of this paper is as follows: Section II discusses the concepts and related works. Section III described the proposed method and packing conditions. Section IV is the experiment and results. Finally, Section V is the conclusion.

II. RELATED WORKS

A. Container Loading Problem

Container Loading Problem is NP-HARD problem [4] that cannot be solved in polynomial time. As the container has a rectangular shape, all items must be packed in rectangular containers or bins as cargoes. The cargoes must be placed in the container under these conditions: there is no part of the cargo overhanging the container and the items must be placed parallel to the container surface area. The main key of the packaging solution is divided into 2 types: the first is to focus on solving the problem of maximum values which means, how to pack the cargoes into a container as many as possible. The second is to consider the minimum value that means how to pack the cargoes into the minimum number of containers. There are the conditions used for cargo packing, such as the weight of the cargoes inside the container which must not exceed the limited weight of the container, the size of the cargoes, the orientations, the placement methods of the cargoes and so on.

In the past, solving bin packing problems has led researchers to devise many algorithms for solving. For example, in 1999, C. Chien and W.Wu [12] developed a heuristic module. The empty space of the container was divided into columns to pack cargoes by volume and type of

cargos. Therefore, the cargos are placed in the remaining space, in order to appropriately use the space in the container. Later, in 2000, Martello et al. [13] used exact branch and bound algorithm to pack the cargos into a multiple container. The conditions for packing the cargo are: the cargo must be placed within the container without rotating, assuming the cargo moves orthogonally, and the cargo edges must be in paralleled position with other cargos and the container. In 2001, A. Bortfeldt and H. Gehring [14] developed a hybrid GA to solve the CLP. The conditions are: the edges of the box is placed in parallel position to the container wall, each box located inside the container must not be overlapped with other boxes, the box can be rotated to any one of six orientations, stack layout structure is used for placing the boxes, taking into account the weight, stability, and balance. In 2006, E. Bischoff [15] considered the load bearing strength to avoid breaking cargo within the container. The cargos were sorted by weight or volume which means that the heavier cargo is packed below the lighter cargo, and if the cargo can be packed in the empty space between the cargoes to load bearing strength, the packing alignment can be changed within the remaining space. In 2010, Y. Wuet et al. [16] solved the problem of 3D-BPP using more than one type of bin. They used heuristic for packing a batch of cartons, which has a building block structure, including GA algorithm. The experiments showed that the algorithm worked effectively and yielded the satisfactory results when compared to the actual packaging practice and Mixed Integer Programming model (MIP). In 2012, K. Kanget et al. [17] proposed the packing of different size of objects into a single container using the Improved deepest bottom left with fill (I-DBLF) strategy. This strategy was used to fill in the least amount of space, considering the width, length, and height of objects with overlap packable spaces. Then Hybrid genetic algorithm was used to solve 3D-BPP problems. In 2013, Kimpan W. and Tantawarak N. [18] considered the arrangement of different size of items into a container. The container is divided into many partitions, which is easy to deliver the cargoes as in their orders. They used Best-Fit Decreasing algorithm to load the items into the containers.

B. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) [19] is a simulation of social interaction and cognitive behavior in flock of bird, which are the learning processes used in each individual particle. A swarm consists of a number of particles, each of which contains two types of information which are the experience of the particle (known status that has ever chosen) and the performance of other particles within the same environment. This information will tell you how the particle flies to find answers in the search space. PSO has three principles: evaluate, compare, and imitate. Each particle responds to its surroundings based on its attractiveness or abhorrence, which affects its actions and decisions. The particles in swarm compare performance between their neighborhoods to find the maximum fitness. For example, if particle A is compared with its neighborhood which is particle B and particle B performed better performance, then particle A will imitate. Performance of each particle consists of speed and velocity.

In order to update the speed and the direction of the particles, the equations which used to update the two values are shown in equation (1) and equation (2), respectively.

$$v_i(t+1) = wv_i(t) + c_1r_1[\hat{x}_i(t) - x_i(t)] + c_2r_2[g(t) - x_i(t)] \quad (1)$$

where

i	particle index
w	inertial coefficient
c_1, c_2	acceleration coefficients ($0 \leq c_1, c_2 \leq 2$)
r_1, r_2	random values ($0 \leq r_1, r_2 \leq 1$) regenerated every velocity update
$v_i(t)$	velocity of particle at time t
$x_i(t)$	position of particle at time t
$\hat{x}_i(t)$	the individual best of particle at time t (pbest)
$g(t)$	the best solution of swarm at time t (gbest)

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

Previous researchers applied the PSO to solve the bin packing problems. In 2016, PSO was used to solve the 2D bin packing problems of the different cargo types on ship or container by VidyaShingade et al. [20]. The cargo was packed using Bottom Left with Fill (BLF) method to locate the cargo. PSO was used to compare the locations that provide the best fitness value, which means using the minimum space for cargo packing. In 2012, B. M. Domingo et al. [19] solved the CLP problem using square boxes of various sizes, which were packed into a single container. The packing structure was a layer and PSO was used for evaluation, arrangement of the possible packing sequences, and calculating the volume used in each in order to find the best packing sequence. Later in 2017, S. François and et al. [21] proposed a new kind of cutting and packing problem in order to minimize the volume occupied by all objects and placed the objects in discrete space by setting coordinate. A discrete space model has been built to simulate the objects loading. The P.S.O.-starcraft algorithm based on Waterfalls objective-function (W.O.F.) was used in optimization process to find a solution to reduce the volume.

III. PROPOSED METHOD

A. Problem Definition and Constraints

This research aim is to solve CLP problem. It uses placement strategies and PSO to find the feasible packing in order to place cargoes or boxes in the container. The conditions for this solution are: a packed box must be placed inside the container, no part of the box overhanging the container, the packed box must be placed parallel to the container surface area or other boxes. The container contains the volume dimension (width, height, and length) expressed by the coordinate vector (X_H, Y_H, Z_L) . The initial position of the dimension (X, Y, Z) is set to $(0, 0, 0)$, as shown in Fig. 1. The two sizes of the container used in these experiments, based on the standard TIS 587-2528, are 20 feet $(243 \times 259 \times 609 \text{ cm}^3)$ and 40 feet $(243 \times 259 \times 1219 \text{ cm}^3)$.

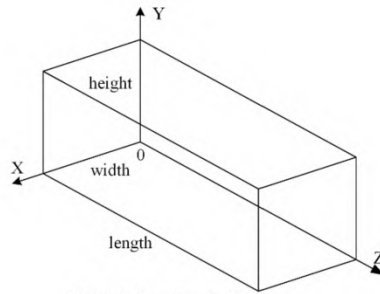


Figure 1. Container in 3D

The box has a rectangular shape. Parameters of the box consists of volume dimension (width, height, and length) expressed by the coordinate vector (x_w, y_h, z_l) , position dimension expressed by the coordinate vector (x, y, z) , status is 0 or 1. If the box is not packed into the container, the status is set to 0, otherwise the status is equal to 1. Each box has its unique id and the orientations are 6 cases as shown in Fig. 2. According to the official website of Kerry, the box sizes used are five types: S, M, L, XL, and OVER, as shown in TABLE I.

CLP is based on the conditions and the process of the box packing which weight arrangement is considered. The heavy boxes are placed in a container floor to prevent breakage. After all boxes are sorted by weight, each box id is set. In order to make it consistent, we will perform simulations of 12 scenarios, as shown in TABLE II. The objective is to find out the maximum number of boxes which are separated by a certain size of boxes, and assorted size of boxes that can be packed into a single container of two sizes.

TABLE I. DETAILS OF EACH BOX

Type	Size (cm ³)	Weight (kg)
S	30x20x10	≤ 5
M	40x30x20	≤ 10
L	50x40x30	≤ 15
XL	60x50x40	≤ 20
Over	70x60x50	≤ 25

TABLE II. DETAILS OF SCENARIOS

Case	Size of Box	Size of Container
1	S	20 feet
2	M	
3	L	
4	XL	
5	OVER	
6	All sizes	
7	S	40 feet
8	M	
9	L	
10	XL	
11	OVER	
12	All sizes	

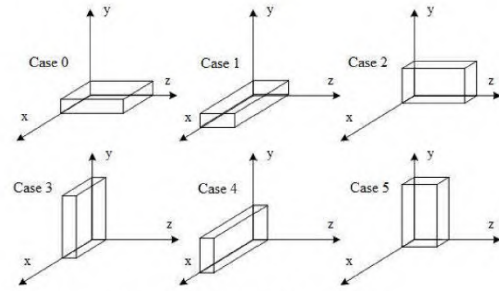


Figure 2. Six directions of orientations

B. Packing Strategy

The relevant objects consist of a container, boxes, and space. The container and the box consist of the parameters as mentioned earlier. In addition, the space contains the volume dimension of (X_{sw}, Y_{sh}, Z_{sl}) and the position dimension of (X_s, Y_s, Z_s) . The initial position equals $(0, 0, 0)$ and the volume is a volume of container. The first step is picking up the first box which is sorted by the weight. Second, choosing the least volume of space as in equation (3)

$$S_{\min} = \min \sum_{j=0}^s X_{sw_j} Y_{sh_j} Z_{sl_j} \quad (3)$$

Then, considering the conditions of placing which is the volume of the space must be greater than or equals to the volume of the box and the volume dimension of the space must be greater than or equals to the volume dimension of the box. Otherwise, some parts of the box may be overhanging the container as in equation (4) and (5).

$$V_i = \begin{cases} 1, & \sum X_{sw_i} Y_{sh_i} Z_{sl_i} \geq x_w, y_h, z_l \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$V_i = \begin{cases} 1, & X_{sw} \geq x_w \cup Y_{sh} \geq y_h \cup Z_{sl} \geq z_l \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

In addition, considering six directions of orientation conditions in Fig. 2, the box must be at least placed within one direction of the selected space as shown in equation (6).

$$R_i = \sum_{c=0}^5 r_c \quad (6)$$

Where R_i is the sum of the rotation directions of the box. If it is equal to 0, it means that the space makes the box overhanging or out of bounds. Therefore, the box cannot be placed in the selected space. On the other hand, if R_i is greater than 1, it means that less than 1 direction that the box can be rotated, so the box can be placed in the selected space. After that the box is changed to packed status and the unique id to the box is set.

After placing the box, the next step is slicing process. The selected space from the previous process will be prepared to slice into 3 parts along with the x, y, z axis of

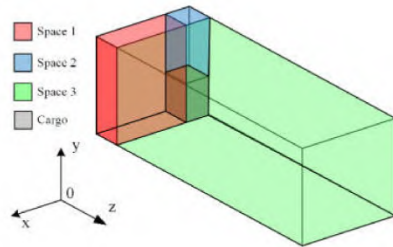


Figure 3. The new spaces from slicing space

the box as shown in Fig. 3. The new spaces will not overlap the volume dimension of other boxes then it will be added into the list space and the old space will be removed from the list space. The algorithm used to slice the space is shown in Algorithm 1.

Next, slice the space of all box alignment, for example, the first box has a total possible rotations in the space selected which are six directions ($R_1 = 6$). The second box is $R_2 = 5$ and the third box is $R_3 = 3$.

Algorithm 1: Slice Space Algorithm

```

For each axis do
  if (X axis)
    PositionX0 = abs(spacePosition0 + cargoVolume0)
    VolumeX0 = abs(spaceVolume0 - cargoVolume0)
    PositionX1 = spacePosition1
    VolumeX1 = spaceVolume1
    PositionX2 = spacePosition2
    VolumeX2 = cargoVolume2
  else if (Y axis)
    PositionY0 = spacePosition0
    VolumeY0 = cargoVolume0
    PositionY1 = spacePosition1 + cargoVolume1
    VolumeY1 = spaceVolume1 - cargoVolume1
    PositionY2 = spacePosition2
    VolumeY2 = cargoVolume2
  else
    PositionZ0 = spacePosition0
    VolumeZ0 = spaceVolume0
    PositionZ1 = spacePosition1
    VolumeZ1 = spaceVolume1
    PositionZ2 = abs(spacePosition2 + cargoVolume2)
    VolumeZ2 = abs(spaceVolume2 - cargoVolume2)
  End if
End if
if ((VolumeX0 * VolumeX1 * VolumeX2) > 0)
  addnewSpace
End if
if ((VolumeY0 * VolumeY1 * VolumeY2) > 0) {
  addnewSpace
End if
if ((VolumeZ0 * VolumeZ1 * VolumeZ2) > 0) {
  addnewSpace
End if

```

Then, keep all the rotation values of the box in the list, and combine the rotation directions of the boxes together. So, the lists of all combinations are 90 lists (6x5x3), as shown in Fig. 4. After all lists of orientation are acquired, preparing parameters for PSO is started which are shown in Algorithm 2. Each list is a particle (i), total volume of each list (volume_list) and each particle defines the initial value as follows:

```

particle_index      = i
particle_position   = volume_list
particle_best_position = particle_position
particle_velocity    = random float number [0,1]
particle_fitness     = is calculate by equation (7)
particle_best_fitness is null

```

$$fit = volume_container - \sum_{i=0}^N (x_{w_i} \cdot y_{h_i} \cdot z_{l_i}) \quad (7)$$

After initializing each particle, the particle is compared to find the maximum fitness, according to equation (1) and equation (2). Repeat the number of iteration defined as a number of iterations. The final result is volume ($swam_best_position$), space ($swam_best_fitness$), and the list of the minimum remaining space in the container ($swam_best_index$).

Algorithm 2: PSO Algorithm

```

For all particle p do
  initialParticle()
  if (swarm_best_fitness is null or particle p particle_fitness >
  swarm_best_fitness)
    swarm_best_fitness = p
    swarm_best_index = p
    swarm_best_fitness = particle p particle_fitness
    swarm_best_position = particle p particle_position
  End if
End for
iteration()
Function iteration
  if (i < number_of_iterations and swarm_best_fitness >
  fitness_threshold)
    For all particle p do
      particle p particle_velocity = UpdateVelocity()
      particle p particle_position = UpdatePosition()
      particle p particle_fitness = CalculateFitness()
      if (particle p particle_fitness > particle p particle_best_fitness)
        particle p particle_best_fitness = particle p particle_fitness
        particle p particle_best_position = particle p particle_position
      End if
      if (particle p particle_fitness > swarm_best_fitness)
        swarm_best_index = p
        swarm_best_fitness = particle p particle_fitness
        swarm_best_position = particle p particle_position
      End if
    End for
    i++
  iteration()
  else
    return [swarm_best_position , swarm_best_fitness,
    swarm_best_index]
  End if
End function

```

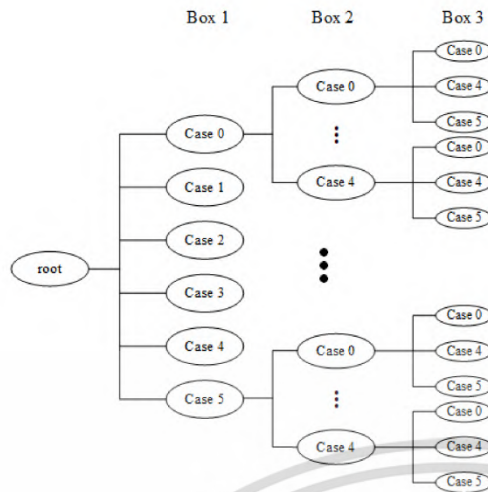


Figure 4. List of all orientations

IV. EXPERIMENTS AND RESULTS

The containers used in this experiment were divided into two sizes by inserting the same and different types of boxes into a container. There were 12 cases, in case 1, we used randomly weighed boxes of size S and sorted the weight in descending order. Then, define the initial space from the container position and volume. Afterward, the boxes that were sorted by weight were picked up and placed into the space and the space was sliced as described in the previous section. After that, the lists of possible rotations of the box were taken into the PSO process by the definition of the PSO values as described in TABLE III. The experimental results indicated that the orientations of each box and the number of boxes in each type can be placed into the container as many as possible, which means that the remaining space is minimal. After that the orientations of boxes in a 3D model using a library called BabylonJS are simulated.

The results of 6 cases for packing boxes into the 20 feet container are: 5,496 boxes of S size can be packed with the 86.03 percent of space utilization, 1,440 boxes of M size can be packed with the 90.17 percent of space utilization. 576 boxes of L size can be packed with the 90.17 percent of space utilization. 300 boxes of XL size can be packed with the 93.92 percent of space utilization. 128 boxes of OVER size can be packed with the 70.13 percent of space utilization. In the last case, 993 boxes of all sizes can be packed with the 93.09 percent of space utilization.

In case of a 40 feet container, the results of 6 cases for packing boxes are: 11,621 boxes of S size can be packed with the 90.88 percent of space utilization. 2,880 boxes of M size can be packed with the 90.09 percent of space utilization. 1,151 boxes of L size can be packed with the 90.02 percent of space utilization. 600 boxes of XL size can be packed with the 93.85 percent of space utilization. 272 boxes of OVER size can be packed with the 74.45 percent of space utilization. In the last case, 2,653 boxes of all sizes can be packed with the 92.13 percent of space utilization.

All results are shown in TABLE IV and TABLE V respectively.

According to the experiments, each box size was placed into both sizes of containers. It can be indicated that the number of boxes that can be packed into different containers are also in different amount. In case of the certain size of boxes are packed into a container, for example, on S size or only OVER size of boxes, the space utilization is ineffective. The box of size S is too small when packed in a container because it left plenty of space, and the box of OVER size is too large when packed into a container as it leaves some space at the top and at the sides of container, as shown in Fig. 5. The best space utilization found in this experiment is XL sizes of boxes arrangement in the both sizes of containers. The arrangement used 93.92 percent at 93.85 percent of space utilization in 20 feet and 40 feet containers, respectively.

V. CONCLUSION

This paper presents a solution for container loading problem without placement heuristics that reduces space between different layouts. The placement process is determined by packing conditions which are: a packed box must be placed inside the container, no part of the box

TABLE III. DEFINITION OF THE PSO VALUES

Parameters	Values of parameters
number_of_particles	all lists of possible rotations box
speed_limit	9
number_of_iteration	10
fitness_threshold	0
inertia_weight	0.729
cognitive_weight (c_1)	1.49445
social_weight (c_2)	1.49445

TABLE IV. RESULTS OF EXPERIMENT FOR CONTAINER SIZE 20'

Type	Case					
	1	2	3	4	5	6
S	5496	0	0	0	0	780
M	0	1440	0	0	0	13
L	0	0	576	0	0	37
XL	0	0	0	300	0	64
OVER	0	0	0	0	128	99
Total box	5496	1440	576	300	128	993
Utilize area (%)	86.03	90.17	90.17	93.92	70.13	93.09

TABLE V. RESULTS OF EXPERIMENT FOR CONTAINER SIZE 40'

Type	Case					
	7	8	9	10	11	12
S	11621	0	0	0	0	2120
M	0	2880	0	0	0	145
L	0	0	1151	0	0	105
XL	0	0	0	600	0	125
OVER	0	0	0	0	272	158
Total box	11621	2880	1151	600	272	2653
Utilize area (%)	90.88	90.09	90.02	93.85	74.45	92.13

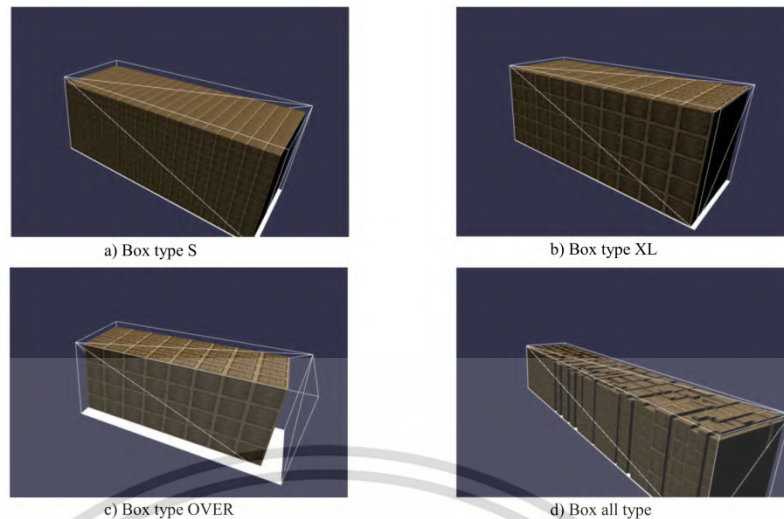


Figure 5. Simulation of packing boxes into container

overhanging the container, the packed box must be placed parallel to the container surface area or other boxes. In addition, weight and orientation conditions are considered for packing boxes into the container. Then, PSO is used to find minimum space of utilization. The experimental results are the orientation of boxes and the maximum number of boxes can be packed into a single container. Moreover, it shows how the multiple boxes packed into the container with 3D simulation. The results indicate that the best space utilization is XL sizes of boxes arrangement in the both sizes of containers.

REFERENCES

- [1] E. Bischoff and M. Ratliff, "Issues in the development of approaches to container loading", *Omega*, vol. 23, no. 4, pp. 377-390, 1995.
- [2] E. Bischoff, F. Janetz and M. Ratliff, "Loading pallets with non-identical items", *European Journal of Operational Research*, vol. 84, no. 3, pp. 681-692, 1995.
- [3] A. Lim, B. Rodrigues and Y. Wang, "A multi-faced buildup algorithm for three-dimensional packing problems", *Omega*, vol. 31, no. 6, pp. 471-481, 2003.
- [4] D. Pisinger, "A Tree Search Heuristic for the Container Loading Problem", *Ricerca Operativa*, vol. 28, pp. 31-48, 1999.
- [5] T. Fanslau and A. Bortfeldt, "A Tree Search Algorithm for Solving the Container Loading Problem", *INFORMS Journal on Computing*, vol. 22, no. 2, pp. 222-235, 2010.
- [6] M. Eley, "Solving container loading problems by block arrangement", *European Journal of Operational Research*, vol. 141, no. 2, pp. 393-409, 2002.
- [7] R. Morabito and M. Arenalest, "An AND/OR-graph Approach to the Container Loading Problem", *International Transactions in Operational Research*, vol. 1, no. 1, pp. 59-73, 1994.
- [8] H. Gehring and A. Bortfeldt, "A Parallel Genetic Algorithm for Solving the Container Loading Problem", *International Transactions in Operational Research*, vol. 9, no. 4, pp. 497-511, 2002.
- [9] J. Gonçalves and M. Resende, "A parallel multi-population biased random-key genetic algorithm for a container loading problem", *Computers & Operations Research*, vol. 39, no. 2, pp. 179-190, 2012.
- [10] A. Bortfeldt and H. Gehring, "Applying Tabu Search to Container Loading Problems", *Operations Research Proceedings 1997*, pp. 533-538, 1998.
- [11] A. Bortfeldt, H. Gehring, and D. Mack, "A parallel tabu search algorithm for solving the container loading problem", *Parallel Computing*, vol. 29, no. 5, pp. 641-662, 2003.
- [12] C. Chien and W. Wu, "A framework of modularized heuristics for determining the container loading patterns", *Computers & Industrial Engineering*, vol. 37, no. 1-2, pp. 339-342, 1999.
- [13] S. Martello, D. Pisinger, and D. Vigo, "The Three-Dimensional Bin Packing Problem", *Operations Research*, vol. 48, no. 2, pp. 256-267, 2000.
- [14] A. Bortfeldt and H. Gehring, "A hybrid genetic algorithm for the container loading problem", *European Journal of Operational Research*, vol. 131, no. 1, pp. 143-161, 2001.
- [15] E. Bischoff, "Three-dimensional packing of items with limited load bearing strength", *European Journal of Operational Research*, vol. 168, no. 3, pp. 952-966, 2006.
- [16] Y. Wu, W. Li, M. Goh, and R. D. Souza, "Three-dimensional bin packing problem with variable bin height", *European Journal of Operational Research*, vol. 202, no. 2, pp. 347-355, 2010.
- [17] K. Kang, I. Moon, and H. Wang, "A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem", *Applied Mathematics and Computation*, vol. 219, no. 3, pp. 1287-1299, 2012.
- [18] Kimpan W. and Tantawarak N., "Simulation model of space optimization in container arranging and sequencing for logistics", *IADIS International Conference Information Systems Post-Implementation and Change Management*, Czech Republic, pp. 51-57, 2013.
- [19] B. M. Domingo, S. Ponnambalam, and G. Kanagaraj, "Particle Swarm Optimization for the single container loading problem," *2012 IEEE International Conference on Computational Intelligence and Computing Research*, 2012.
- [20] Vidya Shingade, Shital Kumbhar, and Sonali Khandve, "2D Packaging Problem using Swarm Intelligence," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 3, Mar. 2016.
- [21] S. François, S. Sebastien, C. Dominique, B. Thomas and M. Yann, 2017. [Online]. Available: <https://cfm2017.sciencesconf.org/133143/document>. [Accessed: Aug- 2018].

ประวัติผู้เขียน

ชื่อ	นางสาว ศศิธร คุณสินธนานันต์
วัน เดือน ปีเกิด	20 เมษายน พ.ศ. 2534
ที่อยู่ปัจจุบัน	20/75 ซอยรามคำแหง 172 ถนนรามคำแหง แขวง/เขต มีนบุรี กรุงเทพมหานคร 10510
ประวัติการศึกษา	(2555) วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ เกรดเฉลี่ย 3.27 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานทางวิชาการ	Koonsintananan, S., & Kimpan, W. 2019. “Applied Particle Swarm Optimization in Solving Container Loading Problem for Logistics.” <i>2019 11th International Conference on Knowledge and Smart Technology (KST)</i> . pp. 88-93.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้