

# การออกแบบอย่างมีประสิทธิภาพของคลังสินค้าเหล็กด้วยวิธีแบบกลุ่มอนุภาคโดยมีการเรียนรู้อย่างครอบคลุม

## Comprehensive Learning Particle Swarm Optimization of Steel Warehouse Structures

กวิน ไชยฤกษ์ และ นที จิ้งสวัสดิ์เมธา และ อ.ดร. เสวกชัย ตั้งอร่ามวงศ์

ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย จ.กรุงเทพฯ

### บทคัดย่อ

ในปัจจุบันในการออกแบบโครงสร้างเหล็กหล่อรูปพรรณนั้น มีจุดประสงค์คือต้องการให้โครงสร้างนั้นปลอดภัย, มีความสามารถในการใช้งานและประหยัดค่าใช้จ่ายมากที่สุดเท่าที่จะสามารถเป็นไปได้ ดังนั้นการเลือกหน้าตัดเหล็กจึงเป็นเรื่องที่สำคัญมากเพราะราคาโครงสร้างนั้นมาจากค่าวัสดุถึง 40% แต่การออกแบบโครงสร้างให้ได้ตามเงื่อนไขที่ต้องการนั้นใช้เวลาเป็นจำนวนมาก ดังนั้นจึงเริ่มมีการใช้โปรแกรม OAPI ร่วมทำงานกับโปรแกรมวิเคราะห์โครงสร้างเพื่อที่จะวิเคราะห์โครงสร้างนั้นเป็นไปอย่างรวดเร็วและมีประสิทธิภาพนอกจากนี้แล้วเพื่อการทำงานที่รวดเร็วมากยิ่งขึ้นจึงเริ่มมีการนำอัลกอริทึมเข้ามาใช้กับการออกแบบโครงสร้างด้วยโปรแกรม OAPI และ โปรแกรมวิเคราะห์โครงสร้างแต่การออกแบบโครงสร้างรูปพรรณเหล็กนั้นเป็นปัญหาที่ความซับซ้อนสูงผู้วิจัยจึงเลือกอัลกอริทึม CLPSO มาประยุกต์ใช้กับการออกแบบโครงสร้างรูปพรรณเหล็กตัวอย่างคือ โครงสร้างคลังสินค้าเหล็กหน้าตัดกว้าง 20 เมตร ด้วยจุดประสงค์การออกแบบคือ มีน้ำหนักน้อยที่สุด ภายใต้เงื่อนไขที่มีกำลังรับแรงมากเพียงพอ และ มีความสามารถในการใช้งานที่ดี โดยใช้เวลาให้น้อยที่สุดเท่าที่สามารถทำได้ ซึ่งหลังจากการได้ลองประยุกต์ใช้โปรแกรมในการเพิ่มประสิทธิภาพการออกแบบโครงสร้างนั้นผลออกมาว่าสามารถออกแบบโครงสร้างได้ตามวัตถุประสงค์ และ ประหยัดเวลาเป็นอย่างมาก

คำสำคัญ: OAPI, CLPSO, การเพิ่มประสิทธิภาพการออกแบบโครงสร้าง

### Abstract

In the present day, the purpose of steel structure design is to strengthen structure to be safe, serve enough serviceability and most economical. That's why, selection of steel section becomes an important role in steel structure design because of 40% of construction cost come from this part. However, designing steel structure to meet the criteria consumes a lot of time. Therefore, the OAPI (Open Application Programming Interface) and structural analysis software are used to improve

design procedure as well as to implement optimal results. In addition, applying algorithm with OAPI and structural analysis software is a way to solve structure design problem. Due to the high complexity of design problem we decided to use CLPSO (Comprehensive Learning Particle Swarm Optimization) to applied with this structure design problem. In this thesis, we used CLPSO algorithm applied with Visual Basic and SAP2000 to design 20m span length steel arches warehouse with minimum weight under two constraint conditions are strength constraints and serviceability constraints. After we ran a structural optimization algorithm, the result met with the objectives and saved a lot of time.

Keywords: OAPI, CLPSO, Structural optimization

### 1. บทนำ

ในปัจจุบันนั้นการออกแบบโครงสร้างเหล็กหล่อรูปพรรณนั้นมีความท้าทายเป็นอย่างมากเพราะแค่การออกแบบให้รับแรงได้และรู้สึกลดภัยนั้นไม่เพียงพอต้องประหยัดเวลาและค่าใช้จ่ายด้วยจึงเริ่มมีการเพิ่มประสิทธิภาพในการออกแบบในโครงสร้างโดยการมองย้อนกลับไปปัญหาเรื่องของการใช้เวลามากเกินไปนั้นขั้นตอนการเลือกหน้าตัดของเหล็กรูปพรรณใช้เวลาสูงอย่างมากทำให้ในการทำโครงงานนี้นั้นเลือกใช้โปรแกรมในการเลือกหน้าตัดสำหรับโครงสร้างโดยมีจุดประสงค์ให้น้ำหนักเบาที่สุดเท่าที่สามารถเป็นไปได้เนื่องจากยิ่งน้ำหนักน้อยปริมาณเหล็กที่ใช้จะน้อยลงส่งผลให้ราคาวัสดุลดลงภายใต้ข้อจำกัดคือต้องมีความสามารถในการรับแรงและความสามารถในการใช้งานที่ดีโดยใช้เวลาให้น้อยที่สุดเท่าที่สามารถเป็นไปได้

เนื่องในในสถานการณ์โลกในปัจจุบันเกิดวิกฤตไวรัสโคโรนาสายพันธุ์ใหม่ 2019 ทำให้เกิดข้อจำกัดด้านการใช้โปรแกรมและเทคโนโลยีทำให้ผู้วิจัยนั้นได้จำกัดขอบเขตการวิจัยโดยพิจารณาอัลกอริทึมที่เหมาะสมหนึ่ง

ตัวคือ CLPSO โดยประยุกต์ใช้กับปัญหาที่เหมาะสมโดยจะเป็นปัญหาที่ความซับซ้อนซึ่งในที่นี้คือ การออกแบบโครงสร้างคลังสินค้าเหล็ก

## 2. ความรู้และทฤษฎีที่เกี่ยวข้อง

### 2.1 โครงสร้างช่องโค้ง (Arch Structure)

โครงสร้างช่องโค้งสามารถนำไปใช้ได้อย่างหลากหลายในเชิงสถาปัตยกรรม เช่น หลังคา, ห้องนิรภัย, โครงสร้างร้านค้า, ห้องโถงนิทรรศการ, สนามบิน, โรงเก็บเครื่องบิน, และ สะพาน โดยตัวอย่างจะแสดงดังรูปที่ 1



รูป 1 Sydney Harbour Bridge, New South Wales, Australia

### 2.2 อัลกอริทึม meta-heuristic

Meta-heuristic ประกอบไปด้วยคำว่า “meta” แปลว่า “ขั้นสูง” และ “heuristic” แปลว่า “การแก้ปัญหา” ดังนั้น meta-heuristic คือ การแก้ปัญหาขั้นสูง ถูกออกแบบมาเพื่อแก้ปัญหาคำตอบที่เพิ่มประสิทธิภาพโดยการปรับปรุงผลลัพธ์ที่มีโอกาสอย่างมีนัยสำคัญให้ดีขึ้นหลายๆรอบโดยมีการวัดผลของคุณภาพของผลลัพธ์นั้นด้วย แต่ถึงอย่างไรก็ตามวิธี meta-heuristic ก็ไม่สามารถการันตีได้ว่าผลลัพธ์ที่ได้มามีประสิทธิภาพสูงสุดจากผลลัพธ์ที่เป็นไปได้ทั้งหมดโดยคุณสมบัติของ meta-heuristic มีดังนี้

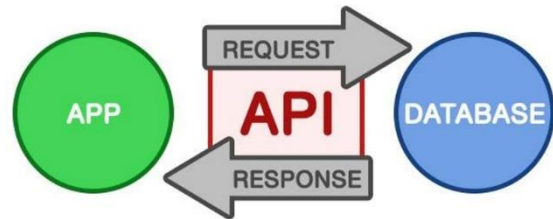
1. Meta-heuristic กุญแจที่นำทางกระบวนการค้นหาคำตอบที่เป็นไปได้ โดยเป้าหมายคือการตามหาผลลัพธ์ที่ใกล้เคียงผลลัพธ์ที่มีประสิทธิภาพสูงสุด
2. เทคนิคที่สามารถพัฒนาเป็นอัลกอริทึม meta-heuristic ได้นั้นมีตั้งแต่การค้นหาขั้นตอนที่ไม่ซับซ้อนจนไปถึงกระบวนการการศึกษาที่ซับซ้อน
3. อัลกอริทึม meta-heuristic นั้นเป็นการประมาณ และ มักไม่เป็นไปตามกำหนด (non-deterministic)

### 2.3 Stochastic Search Method

เทคนิค stochastic search มีประสิทธิภาพสูงมากสำหรับการหาผลลัพธ์ของทั้งปัญหาการเสริมประสิทธิภาพแบบต่อเนื่องและไม่ต่อเนื่อง ความคิดพื้นฐานของเทคนิคนี้คือการจำลองปรากฏการณ์ทางธรรมชาติขึ้นมาเช่น การอยู่รอดของสิ่งมีชีวิตที่เหมาะสม ระบบภูมิคุ้มกัน และการฉลาดแบบกลุ่ม วิธีการนี้เป็นการค้นหาแบบสุ่มที่ไม่มีแบบแผนและไม่ต้องการข้อมูลหรือความโค้งงอของฟังก์ชันเป้าหมายและขอบเขตโดยวิธีการนี้ใช้กฎของการเปลี่ยนแปลงความน่าจะเป็น

### 2.4 โปรแกรม API

โปรแกรม API หรือ Application Program Interface (API) ซึ่งคือ คำสั่ง (Code) ที่อนุญาตให้ software program สามารถสื่อสารระหว่างกันได้ ถ้าจะพูดในภาษาคนเขียน program แล้ว API เป็นช่องทางสำหรับขอใช้บริการคำสั่ง จาก operation system (OS) หรือ application อื่น ๆ ซึ่งมันใช้งานโดยติดตั้ง function และเรียกใช้งานตามคำสั่งที่เขียนไว้โดยในปัจจุบันการใช้ประโยชน์มากมายจากโปรแกรมประเภทนี้เช่น การสื่อสารกันระหว่าง Application 2 อัน เป็นต้นโดยบริษัท CSI ที่พัฒนาโปรแกรม SAP 200 ก็สร้างตัว CSI API ขึ้นมาเพื่อให้ผู้พัฒนาและวิศวกรใช้โปรแกรมได้อย่างมีประสิทธิภาพสูงสุดนอกจากนี้บริษัท Microsoft ก็พัฒนาโปรแกรม Visual basic ขึ้นมาซึ่งตัวโปรแกรมนั้นสามารถเขียนคำสั่งเพื่อนำเข้าและส่งออกข้อมูลจากโปรแกรม Microsoft office (เช่น Microsoft Excel ) ได้ และยังสามารถใช้ประโยชน์ได้มากมาย



รูป 2 การทำงานของโปรแกรม API

### 2.5 การออกแบบโครงสร้างเหล็กภายใต้มาตรฐาน AISC-LRFD

การออกแบบโครงสร้างเหล็กด้วยวิธี LRFD นั้นเป็นการมองว่าตัว Load มีความผันผวนเนื่องจากพฤติกรรมของโครงสร้างในเชิงวัสดุและในเชิงพฤติกรรมการใช้โดยตัวจะมีความผันผวนมากกว่าคือ Live load เพราะแค่เปลี่ยนรูปแบบการใช้ของพื้นที่ Live load ก็เปลี่ยนดังนั้นใน Load combinations นั้น Live จะมีตัวคูณเผื่อที่สูงกว่าและเมื่อเรานำทั้ง Live load และ Dead load ที่คูณด้วยตัวคูณเผื่อแล้วจะได้ค่า Load สูงสุดที่เกิดขึ้นและเป็นค่าที่เราต้องออกแบบโครงสร้างของเราให้รับ Load นั้นได้ และนอกจากนั้นโครงสร้างเหล็กที่เราออกแบบนั้นยังต้องมีการลดทอนกำลังต่อนจำนวนเพื่อเผื่อสำหรับค่าผิดพลาดตอนก่อสร้างโดยตัวคูณลดกำลังนั้น จะ ขึ้น อยู่กับ ประเภท ของ โครงสร้าง เหล็ก นั้น ๆ

- 1.4D
- 1.2D + 1.6L
- 1.2D + 1.6W + 0.5L
- 1.2D ± 1.0E + 0.5L
- 0.9D ± (1.6W or 1.0E)

D = dead load  
L = live load  
W = wind load  
E = earthquake load

รูป 3 LRFD Load Combinations

AISC-LRFD Resistance Factors

member	resistance factor	limit states
Tension	$\phi = 0.90$	yielding
	$\phi = 0.75$	fracture
Compression	$\phi = 0.85$	buckling or yielding
Beams	$\phi_b = 0.9$	bending
	$\phi_v = 0.9$	shear
Welds	same as for member actions	
Fasteners	$\phi = 0.75$	all

รูป 4 LRFD Resistance factors

2.6 การเสริมประสิทธิภาพการออกแบบโครงสร้าง (Structural Optimization)

มีวัตถุประสงค์ คือ การทำให้คุณสมบัติทางกายภาพของโครงสร้างมีค่าน้อยที่สุดเท่าที่สามารถเป็นไปได้ เช่น ให้โครงสร้างนั้นรับแรงได้โดยมี strength ratio ใกล้เคียง 1 มากที่สุดและยังมีคุณสมบัติทางกายภาพอื่น ๆ อีกมากมายเช่น ปริมาตร น้ำหนัก พื้นที่ เป็นต้นโดยการเลือกคุณสมบัติทางกายภาพนี้จะขึ้นอยู่กับจุดประสงค์ของงานซึ่งถ้าพูดถึงการ Optimization ในเชิงคณิตศาสตร์จะสามารถอธิบายได้ดังนี้

$$X = \{x_1, x_2, \dots, x_n\} \quad (1)$$

$$W(X) \quad (2)$$

$$g_j(X) \leq 0; j = (1, n) \quad (3)$$

$$x_i^L \leq x_i \leq x_i^U; i = (1, n) \quad (4)$$

โดยสมการที่ (1),(2),(3), และ (4) คือ ตัวแปรที่ต้องการทราบค่า, ฟังก์ชันเป้าหมาย, ฟังก์ชันข้อจำกัดของการเสริมประสิทธิภาพ, และ ฟังก์ชันของเขตของตัวแปรที่ต้องการทราบ ตามลำดับ

2.7 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) อัลกอริทึมนี้เป็นการเสริมประสิทธิภาพประเภท population-based stochastic โดยใช้ความรู้พื้นฐานของแนวคิดการฉลาดแบบกลุ่ม อัลกอริทึมได้รับแรงบันดาลใจจากการไหลเวียนของข้อมูลและการศึกษาพฤติกรรมกรหาปลาของฝูงนกเงือก โดยพวกมันจะปรับเปลี่ยนการเคลื่อนไหวของฝูงเพื่อที่จะตามหาอาหาร ตามหาสมาชิกในฝูง และ หลีกเลียงนักล่า โดยพฤติกรรมนี้จะเป็นพฤติกรรมที่เกิดขึ้นกับกลุ่มขึ้นอยู่กับ 2 ปัจจัยคือ ความจำของแต่ละตัว (Particle) และ ข้อมูลหรือความรู้ที่มีอยู่ในบ่อ (Swarm) ดังนั้นสามารถสรุปได้ว่าการทำงานของอัลกอริทึมนี้มีอยู่ 3 หลักการได้แก่ การประเมินผล, การเปรียบเทียบ, และ เลียนแบบโดยส่วนประกอบของ PSO นั้นจะประกอบด้วย จุดประสงค์, ข้อจำกัด และ ขอบเขตของการทำงานซึ่งการทำงานของ PSO มีความรวดเร็วและแม่นยำแต่ PSO นั้นจะมีปัญหาด้านการลู่เข้าของคำตอบ

2.8 Comprehensive Learning Particle Swarm Optimization (CLPSO)

จากข้อบกพร่องของ PSO คือ การลู่เข้าสู่คำตอบของปัญหาก่อนเวลาที่กำหนดไว้โดย พื้นฐานแล้วอนุภาคในอัลกอริทึมจะเรียนรู้ค่า Gbest เป็นหลักโดยค่า Gbest จะทำให้การลู่สู่คำตอบไวขึ้น ถึงแม้จะเป็นค่าล่าสุดใน

ปัจจุบันก็ยังห่างไกลจากจุดที่มีประสิทธิภาพสูงสุดโดยรวม(global optimum) และ อนุภาคจะถูกดึงดูดเข้าสู่ช่วงที่เป็นจุดที่มีประสิทธิภาพเฉพาะที่ (local optimum)

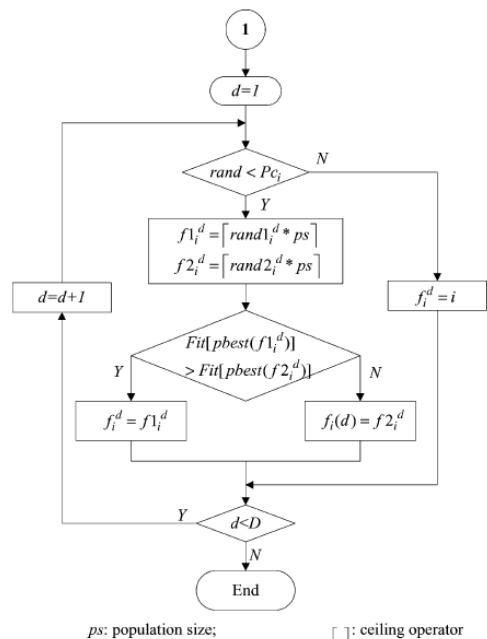
เราใช้แนวคิดเพื่อที่จะใช้ประโยชน์จาก PSO คือการเรียนรู้แบบครอบคลุม (Comprehensive Learning) ใน CLPSO Pbest ของทุกอนุภาคจะถูกทำการเรียนรู้ค่าความเร็วของอนุภาคใหม่ ๆ เสมอเพื่อเป็นตัวนำร่องให้กับอนุภาคอื่น ๆ

ในการเรียนรู้แบบครอบคลุม (Comprehensive Learning) เราจะใช้สมการการปรับปรุงความเร็วดังสมการที่ (5)

$$V_i^d \leftarrow wV_i^d + c(rand_i^d)(Pbest_{f_i(d)}^d - X_i^d) \quad (5)$$

โดย  $f_i = [f_i(1), f_i(2), \dots, f_i(D)]$  หมายถึงค่า Pbest ของอนุภาคที่  $i$  ที่ควรใช้ตาม,  $Pbest_{f_i(d)}^d$  คือมิติที่ตรงกันกับ Pbest ของอนุภาคอื่น ๆ หรือ ตัวมันเอง, การตัดสินใจขึ้นอยู่กับความน่าจะเป็น  $P_c$  ซึ่ง  $P_c$  คือความน่าจะเป็นการเรียนรู้ของแต่ละอนุภาคที่จะมีค่าแตกต่างกัน แต่ละมิติของอนุภาค  $i$  เราจะทำการสุ่มตัวเลข ถ้าเลขมีค่ามากกว่า  $P_c$  แล้วนั้น มิติที่ตรงกันจะใช้ค่า Pbest ของตัวมันเอง ในทางตรงกันข้ามถ้าตัวเลขที่ถูกสุ่มมีค่าน้อยกว่า  $P_c$  ต้องเรียนรู้ค่า Pbest ของอนุภาคอื่น โดยจะมีขั้นตอนการเลือก Pbest ที่ต้องเรียนรู้จากอนุภาคอื่นดังนี้

1. ทำการสุ่มเลือกอนุภาคมา 2 ตัวจากประชากรทั้งหมดในบ่อ โดยที่มีค่าความเร็วเป็นปัจจุบันของอนุภาคนั้น ๆ
2. ทำการเปรียบเทียบค่า Pbest ของทั้ง 2 อนุภาค ใน CLPSO ค่าที่มีความเหมาะสมมากกว่าคือค่าที่มีค่ามากกว่า
3. ใช้ค่าที่มี Pbest ที่ได้รับการคัดเลือกตามข้อ 1 และ 2 ให้เป็นแบบอย่างให้อนุภาคในมิติอื่น ๆ เรียนรู้ ถ้าอนุภาคอื่น ๆ มีค่า Pbest ของตัวมันเท่ากับตัวต้นแบบ จะต้องทำการสุ่มเลือกให้หนึ่งมิติของอนุภาคนั้นทำการเรียนรู้ Pbest จากอนุภาคอื่น มิติเดียวกัน โดยลายระเอียดการเลือกค่า  $f_i$  จะแสดงดังรูป 5



รูป 5 การคัดเลือกต้นแบบสำหรับอนุภาคที่  $i$

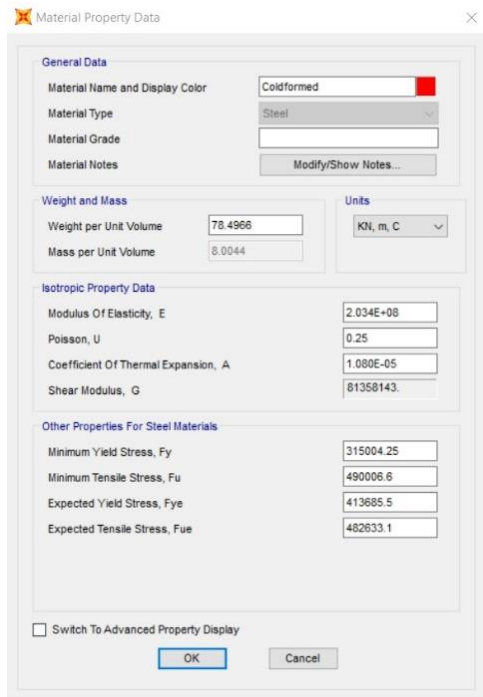
โดยข้อแตกต่างระหว่าง PSO และ CLPSO คือ PSO จะทำการค้นหาตั้งแต่ขอบเขตสูงสุดถึงขอบเขตต่ำสุด แต่ใน CLPSO จะทำการแบ่งช่วงการค้นหาหาระหว่างขอบเขตสูงสุดและขอบเขตต่ำสุดเป็นหลายกรณีทำให้การทำงานภายใต้ข้อจำกัดต่าง ๆ มีความซับซ้อนน้อยลงส่งผลให้การดำเนินการนั้นมีความเร็วมากกว่า PSO แบบปกติ

นอกจากนี้ยังมีผลการแข่งขันจากการแข่งขันรายการ CEC2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization ของ Computational Intelligence Laboratory, Zheng Zhou University, China เป็นข้อพิสูจน์ว่า CLPSO มีความสามารถในแก้ปัญหาที่มีความซับซ้อนได้ดี โดยจากผลการแข่งขันนั้นอัลกอริทึมแบบ CLPSO นั้นมีประสิทธิภาพในการแก้ปัญหาแบบ Hybrid Function และ Composition Function มากกว่า PSO แบบอื่น ๆ แต่ในทางกลับกันประสิทธิภาพในการแก้ปัญหาแบบ Unimodal Function และ Simple Multimodal Function มีไม่มากเท่า PSO แบบอื่น

### 3. ขั้นตอนการแก้ปัญหา

#### 3.1 การตั้งค่าโปรแกรม SAP 2000

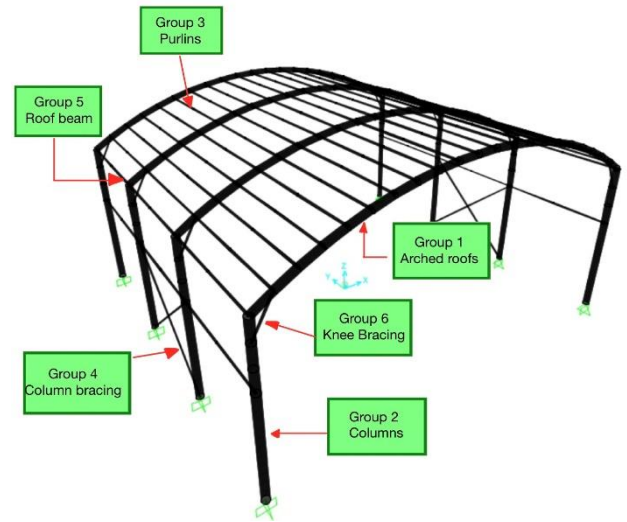
เพื่อการแก้ปัญหาที่ถูกต้องเราจึงต้องตั้งค่าของโปรแกรม SAP 2000 ให้ตรงกับโจทย์ปัญหาโดยขั้นแรกคือต้องตั้งค่าคุณสมบัติทางวัสดุของเหล็กรูปพรรณเย็นอย่างถูกต้องซึ่งประกอบด้วยค่า กำลังรับแรง, ค่าโมดูลัสยืดหยุ่น, ค่าโมดูลัสเฉือน, ความหนาแน่น และ อัตราส่วนปัวซอง โดยมีค่าตามรูป 6



รูป 6 ค่าคุณสมบัติทางวัสดุของโครงสร้างที่จะออกแบบ

หลังจากนั้นจึงเริ่มใช้โปรแกรม SAP2000 ทำการโมเดลโครงสร้างที่จะออกแบบและเพื่อเพิ่มความรวดเร็วในการทำงานของอัลกอริทึม CLPSO

จากโปรแกรม Visual Basic เราจึงทำการจัดกลุ่มโครงสร้างตามประเภทการรับแรงขององค์อาคารตามรูป 7



รูป 7 Group ขององค์อาคารภายในโครงสร้างที่ใช้การออกแบบ

#### 3.2 การประยุกต์ใช้โปรแกรม Visual basic ในการออกแบบโครงสร้างร่วมกับโปรแกรม SAP 2000

หลังจากที่ตั้งค่าที่ตั้งค่า SAP 2000 แล้วเราจึงเริ่มมาจัดการกับตัวโปรแกรม Visual Basic ที่เปิดจาก Microsoft Excel โดยขั้นแรกเราต้องตั้งขอบเขตสูงสุดและขอบเขตต่ำสุดของค่าหน้าตัดเหล็กไว้ใน Search Space Sheet ของ Microsoft excel ซึ่งในแต่ละ Group ขององค์อาคารนั้นจะมีขอบเขตการค้นหาที่แตกต่างกันตามความเหมาะสมซึ่งต้องโดยข้อมูลที่ประกอบนั้นต้องระบุรายละเอียดของขนาดในหน้าตัดด้วยตามรูป 8

	A	B	C	D	E	F	G	H
	Group 1	Section	Area	h	bf	tf	tw	Ixx
1	1	CHS406.4(16")x12.7	157.0793	40.64	40.64	1.27	1.27	30465.73
2	2	CHS406.4(16")x12	148.6853	40.64	40.64	1.2	1.2	28937.01
3	3	CHS406.4(16")x9.5	118.4553	40.64	40.64	0.95	0.95	23338.64
4	4	CHS406.4(16")x9	112.3622	40.64	40.64	0.9	0.9	22192.62
5	5	CHS355.6(14")x12.7	136.811	35.56	35.56	1.27	1.27	20135.45
6	6	CHS406.4(16")x7.9	98.90205	40.64	40.64	0.79	0.79	19640.05
7	7	CHS355.6(14")x12	129.5341	35.56	35.56	1.2	1.2	19139.47
8	8	CHS355.6(14")x9.5	103.294	35.56	35.56	0.95	0.95	15478.02
9	9	CHS406.4(16")x6	75.47362	40.64	40.64	0.6	0.6	15128.33
10	10	CHS355.6(14")x9	97.99884	35.56	35.56	0.9	0.9	14725.86
11	11	CHS355.6(14")x7.9	86.29421	35.56	35.56	0.79	0.79	13047.44
12	12	CHS318.5(12")x10.3	99.7286	31.85	31.85	1.03	1.03	11854.41
13	13	CHS355.6(14")x6.4	70.21083	35.56	35.56	0.64	0.64	10705.54
14	14	CHS318.5(12")x9	87.50906	31.85	31.85	0.9	0.9	10487
15	15	CHS355.6(14")x6	65.89805	35.56	35.56	0.6	0.6	10070.55
16	16	CHS318.5(12")x8	78.03716	31.85	31.85	0.8	0.8	9410.721
17	17	CHS318.5(12")x6.9	67.5455	31.85	31.85	0.69	0.69	8201.895
18	18	CHS318.5(12")x6	58.90486	31.85	31.85	0.6	0.6	7193.186
19	19	CHS267.4(10")x9.3	75.40859	26.74	26.74	0.93	0.93	6287.389
20	20	CHS267.4(10")x9	73.06088	26.74	26.74	0.9	0.9	6105.292
21	21	CHS267.4(10")x8	65.19433	26.74	26.74	0.8	0.8	5488.74
22	22	CHS318.5(12")x4.5	44.3907	31.85	31.85	0.45	0.45	5472.056
23	23	CHS267.4(10")x7	57.26495	26.74	26.74	0.7	0.7	4857.296
24	24	CHS267.4(10")x6.6	54.07561	26.74	26.74	0.66	0.66	4600.496
25	25	CHS267.4(10")x6	49.27274	26.74	26.74	0.6	0.6	4210.723
26	26	CHS267.4(10")x4.5	37.16661	26.74	26.74	0.45	0.45	3211.969
27	27	CHS216.3(8")x8.2	53.60877	21.63	21.63	0.82	0.82	2906.456
28	28	CHS216.3(8")x8	52.3515	21.63	21.63	0.8	0.8	2843.53
29	29	CHS216.3(8")x7	46.02747	21.63	21.63	0.7	0.7	2523.197
30	30	CHS216.3(8")x6	39.64062	21.63	21.63	0.6	0.6	2193.221
31	31	CHS216.3(8")x5.8	38.3557	21.63	21.63	0.58	0.58	2126.051
32	32	CHS216.3(8")x4.5	29.94252	21.63	21.63	0.45	0.45	1679.756
33								
34								
35								

รูป 8 ขอบเขตการค้นหาของ Group1

ต่อมาเมื่อเราเตรียมพร้อมสำหรับการทำงานของโปรแกรม Visual Basic เสร็จแล้วนั้นเราจึงเข้าไปตั้งรายละเอียดของการออกแบบโดยจะใช้โค้ดที่เขียนตามฟังก์ชันจุดประสงค์

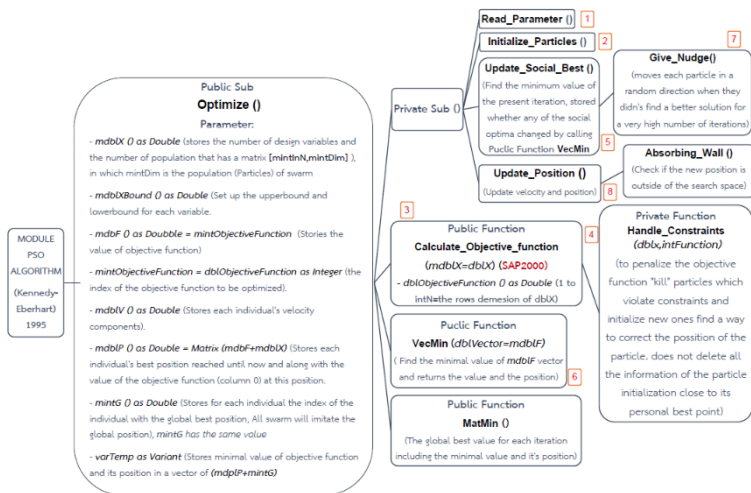
$$\text{Minimize } W(X) = \sum_{i=1}^{ng} A_i \sum_{j=1}^{mk} \rho_j L_j \quad (6)$$

โดย  $W(X)$  คือ ค่ารวมของน้ำหนักโครงสร้างข้อแข็งเหล็กทั้งหมดในโครงสร้าง,  $ng$  คือ จำนวน Group ขององค์อาคารที่เราได้ตั้งไว้,  $mk$  คือ จำนวนขององค์อาคารที่อยู่ใน Group นั้นๆ,  $A_j$  คือ พื้นที่หน้าตัดขององค์อาคารใน Group ที่สนใจ,  $L_j$  คือ ความยาวขององค์อาคารใน Group นั้นๆ และตามเงื่อนไขการทำงานของโครงสร้างเมื่อประยุกต์เข้ากับ CLPSO ดังนั้นจะเกิดข้อจำกัดขึ้นโดยจะเกิดจาก ข้อจำกัดด้านกำลังรับแรงและข้อจำกัดด้านความสามารถในการใช้งานโดนจะเขียนเป็นฟังก์ชันได้ตามนี้

$$\text{Subject to } g_t(x) = \begin{cases} \lambda_j^r \leq 0 ; j = 1, 2, \dots, nc \\ \lambda_d^r \leq 0 ; r = 1, 2, \dots, ns \end{cases} \quad (7)$$

โดย  $nc$  คือ จำนวนของ เสา - คานภายใน Group ขององค์อาคารนั้นๆ  $ns$  คือ จำนวนรวม Stories  $\lambda_j^r$  คือข้อจำกัดด้านกำลังรับแรง  $\lambda_d^r$  คือข้อจำกัดด้านกำลังใช้งาน

ซึ่งฟังก์ชันทั้งหมดที่กล่าวมานั้นจะถูกเขียนออกเป็นโค้ดร่วมกับวิธีการทำงานของอัลกอริทึม CLPSO ตาม Flowchart ในรูป 9



รูป 9 หลักการทำงานของโปรแกรม Visual basis ตามอัลกอริทึม CLPSO

และตามหลักการที่กล่าวมานั้นผู้ใช้โปรแกรมนี้ต้องกรอกจำนวนครั้งที่ทำซ้ำในแต่ละอนุภาคและจำนวนอนุภาคในการดำเนินแต่ละครั้งโดยผู้วิจัยได้เลือกการทำซ้ำ 100 ครั้งในแต่ละอนุภาคและการดำเนินการหนึ่งครั้งจะมีอนุภาคทั้งหมด 15 อนุภาคสรุปได้ดังรูป 10

Parameters	
Inertia	0.4
Cognitive Acceleration	2
Social Acceleration	2
Number of Iterations	100
Number of Particles	15
Social Interaction type	1
"Nudge" criterion (move particles randomly if global optimum didn't)	5
Handling of constraint violations	1
<b>CHS</b>	
Objective Function	100
Number of variables	6
Number of Run	2

รูป 10 รายละเอียดของค่าต่าง ๆ ในอัลกอริทึม CLPSO ตัวอย่าง

#### 4. ผลการศึกษา

การออกแบบโครงสร้างค้ำสินค้าเหล็กด้วยโปรแกรม SAP 2000 และ Visual Basic ภายใต้อัลกอริทึม CLPSO โดยมีจุดประสงค์ให้ได้โครงสร้างที่น้ำหนักรวมขององค์อาคารรวมน้อยที่สุดพร้อมกับมีความสามารถในการรับ

แรงและความสามารถในการใช้งานที่ดีได้ผลของหน้าตัดเหล็กโดยค่าตอบที่

The best solution found is	
$f(x^*) =$	<b>10.5364-(Tons)</b>
<b>at position</b>	
$x1^* =$	<b>CHS318,5(12")x4,5</b>
$x2^* =$	<b>CHS355,6(14")x6,4</b>
$x3^* =$	<b>CHS101,6(3.1/2")x3,2</b>
$x4^* =$	<b>CHS89,1(3")x2,8</b>
$x5^* =$	<b>CHS89,1(3")x3,2</b>
$x6^* =$	<b>CHS89,1(3")x2,8</b>

รูป 11 ผลเฉลยหน้าตัดเหล็กที่ใช้จากการออกแบบ

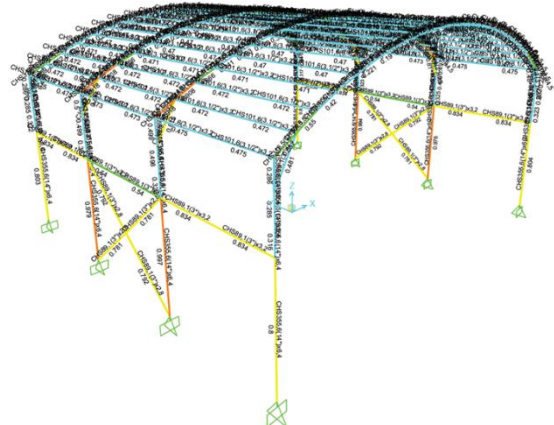
ได้จะแบ่งตาม Group ขององค์อาคารดังนี้

โดยข้อมูลที่ได้นั้นมาจากการดำเนินการโปรแกรม 2 ครั้งซึ่งถ้าพิจารณาจากการที่ใช้ปริมาณการทำซ้ำ 100 ครั้งในแต่ละอนุภาคและการดำเนินการหนึ่งครั้งจะมีอนุภาคทั้งหมด 15 อนุภาคจะคำนวณการทำวิเคราะห์ทั้งหมดเพียง 3,000 ครั้งโดยทุกอนุภาคนั้นจะมีการเก็บค่า Stress Ratio ของทุก Group ไว้โดยผลที่บันทึกไว้นั้นมีค่าดังรูป 12

No.Run	Total_Weight(Lb)	Max_Ratio_Gr1	Max_Ratio_Gr2	Max_Ratio_Gr3	Max_Ratio_Gr4	Max_Ratio_Gr5	Max_Ratio_Gr6
1	12.278	0.92	0.73	1.00	0.75	0.91	0.78
	12.0068	0.70	0.82	1.03	0.97	0.11	0.52
	16.9128	1.01	0.94	1.08	0.38	1.25	0.65
	12.6032	1.00	1.03	0.79	0.97	0.84	0.49
	12.8888	0.59	0.94	0.93	0.26	0.94	0.47
	11.3764	0.74	1.02	0.87	0.40	0.93	0.66
	15.024	1.34	1.31	1.31	0.27	0.60	0.37
	12.8884	0.68	1.03	0.77	0.98	0.93	0.59
	11.3544	0.94	0.98	0.95	0.12	0.84	0.51
	15.3396	0.86	0.52	0.93	1.01	0.80	0.67
	11.7632	0.96	0.89	0.94	0.38	0.84	0.32
	12.6744	0.95	0.85	0.80	0.50	0.31	0.46
	12.0396	0.91	0.88	1.01	0.77	0.83	0.79
	10.7344	0.93	0.99	0.96	0.40	0.84	0.60
	10.9688	0.73	1.01	1.03	0.80	0.84	0.57
2	10.7524	0.88	0.86	0.92	0.78	0.92	0.88
	11.258	0.70	0.82	1.03	0.95	0.59	0.52
	11.6712	0.59	0.99	0.78	0.80	0.93	0.82
	10.626	0.95	0.99	0.95	0.79	0.83	0.46
	11.4628	0.59	0.96	0.98	0.27	0.84	0.43
	11.2932	0.72	0.97	1.03	0.39	0.93	0.78
	14.5788	1.30	1.33	1.35	0.42	0.94	0.91
	12.0808	0.58	0.99	1.01	0.98	0.84	0.56
	11.3544	0.94	0.98	0.95	0.12	0.84	0.51
	11.4464	0.90	0.87	0.82	0.71	0.92	0.71
	10.8512	0.89	0.86	0.91	0.50	0.92	0.70
	11.928	0.92	0.87	0.79	0.50	0.44	0.63
	11.5808	0.89	0.89	0.99	0.95	0.59	0.76
	10.5364	0.92	1.00	0.97	0.79	0.83	0.82
	10.9688	0.73	1.01	1.03	0.80	0.84	0.57

รูป 12 ค่า Stress ratio ที่มีค่ามากที่สุดของแต่ละอนุภาค

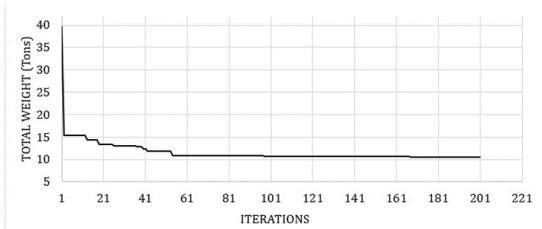
และเพื่อเป็นการยืนยันความถูกต้องและเพิ่มความน่าเชื่อถือของคำตอบการดำเนินการของโปรแกรมผู้วิจัยจึงนำหน้าตัดที่ได้วิเคราะห์ในโปรแกรม SAP 2000 ด้วยตนเองเพื่อยืนยันความถูกต้องซึ่งผลที่ได้ออกมาคือถูกต้อง 100%



รูป 13 การตรวจสอบความถูกต้องของการออกแบบ

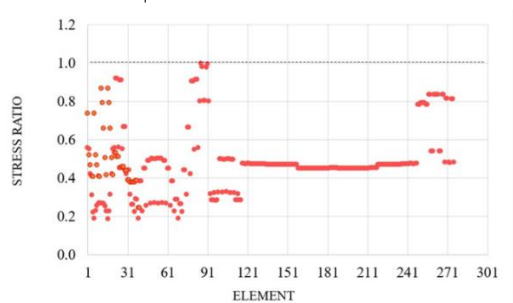
## 5. สรุปผลและวิเคราะห์ผลการทำงาน

จากการดำเนินงานของการเพิ่มประสิทธิภาพการออกแบบโครงสร้างนั้นถือว่าเป็นไปตามจุดประสงค์เพราะทำให้การออกแบบนั้นประหยัดเวลาโดยจากเดิมที่ต้องวิเคราะห์โครงสร้างทั้งหมด  $32^6$  หรือ 1,073,741,824 กรณีเหลือเพียงแค่ 3,000 กรณีทำให้ประหยัดเวลาขึ้นถึง 357913.9 เท่า นอกจากนี้การออกแบบยังบรรลุตามวัตถุประสงค์อีกด้านคือได้น้ำหนักที่เบาที่สุดเท่าที่เป็นไปได้และถ้ามองจากกราฟแต่การทำซ้ำน้ำหนักจะลดลง



รูป 14 น้ำหนักรวมขององค์อาคารทั้งหมดในแต่ละรอบการทำซ้ำ

และโครงสร้างนั้นอยู่ภายใต้ฟังก์ชันข้อกำหนดโดยมี Stress ratio ไม่เกิน 1 และมีค่ามากที่สุดเท่าที่สามารถทำได้เพื่อเพิ่มความสามารถในการใช้งานและเพิ่มความคุ้มค่าของการก่อสร้าง



รูป 15 Stress ratio ของแต่ละองค์อาคาร

โดยในอนาคตนั้นอาจนำโปรแกรมนี้ไปใช้กับคลังสินค้าที่มีหน้ากว้างเพิ่มขึ้นและใช้กับโครงสร้างเหล็กรูปพรรณโครงสร้างอื่นนอกจากนี้อาจขยายความสามารถของโปรแกรมโดยทำให้ใช้ได้กับโปรแกรมวิเคราะห์โครงสร้างหรือโปรแกรมOAPIที่มีความสามารถในการวิเคราะห์สูงกว่านี้

## 6. เอกสารอ้างอิง

- [1] ทักษิณ เทพชาติตรีและอัครวัชร เล่นวารี. พฤติกรรมและการออกแบบโครงสร้างเหล็ก. พิมพ์ครั้งที่ 5. กรุงเทพฯ: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2562
- [2] American Institute of Steel Construction, 2016. Specification for Structural Steel Building. American National Standard ANSI/AISC 360-16.
- [3] Amir Hossein Gandomi, Xin-She Yang, Amir Hossien Alavi (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems.

- [4] Arora, J. S. (2000). Method for discrete variable structural optimization Advanced Technology in Structural Engineering.
- [5] ASCE/SEI 7-10 Minimum Design Loads for Buildings and Other Structures, Structural Engineering Institute.
- [6] Fourie, P., & Groenwold, A. A. (2002). The particle swarm optimization algorithm in size and shape optimization.
- [7] J.J. Liang, A. K. Qin, Ponnuthurai Nagaratnam Syganthan, and S. Bas (2004). Evaluation of Comprehensive Learning Particle Swarm Optimizer
- [8] J.J. Liang, A. K. Qin, Student Member, IEEE, Ponnuthurai Nagaratnam Syganthan, Senior Member, and S. Baskar (2005). Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions.
- [9] K. Mahadevan, P.S. Kannan (2010). Comprehensive Learning Particle Swarm Optimizer for reactive power dispatch
- [10] MBMA-2012 Metal Building Manufacturers Association, Base on the 2012 IBC and ASCE 7-10.
- [11] Perez, R. L., & Behdinan, K. (2007). Particle swarm approach for structure design optimization.
- [12] Plevris, V., Batavanis, A., & Papadrakakis, M. (2011). Optimum design of steel structures with the Particle Swarm Optimization method based on EC3.
- [13] Sachin Desale, Akhtar Rasool, Sushil Andhale, Priti Rane (2015). Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey.
- [14] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer.
- [15] Shi, Y., & Eberhart, R. (1998). Parameter selection in particle swarm optimization
- [16] Tangaramvong, S., & Tin-Loi, F. (2013). Automatic identification of the worst load combination for structural safety assessment using an optimization approach.
- [17] Thu Huynh Van (2018). Analysis and optimum design of cold-formed steel arch structures.
- [18] Venter, G., Sobieszczanski-Sobieski, J. (2003). Particle swarm optimization.
- [19] Yan-Fei, Student member, IEEE, Zhi-Hui Zhan. Yan-Fei, Student member, IEEE, Zhi-Hui Zhan (2014). Comparisons Study of APSO OLPSO and CLPSO on CEC2005 and CEC2014 Test Suit.