SIMULTANEOUS SIZE AND SHAPE STRUCTURAL OPTIMIZATION USING ENHANCED COMPREHENSIVE LEARNING PARTICLE SWARM OPTIMIZATION

Mr. Soviphou Muong

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Civil Engineering
Department of Civil Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

การออกแบบขนาดและรูปร่างของโครงสร้างอย่างเหมาะสมที่สุดด้วยวิธีการของกลุ่มอนุภาคที่ปรับปรุงแบบเบ็ดเสร็จ

นายโสวิภู เมือง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมโยธา
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564

Thesis Title	SIMULTANEOUS SIZE AND SHAPE STRUCTURAL OPTIMIZATION USING ENHANCED COMPREHENSIVE LEARNING PARTICLE SWARM OPTIMIZATION
By	Mr. Soviphou Muong
Field of Study	Civil Engineering
Thesis Advisor	Associate Professor SAWEKCHAI TANGARAMVONG, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Engineering

--- Dean of the FACULTY OF ENGINEERING

(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

THESIS COMMITTEE

--- Chairman

(Associate Professor WITHIT PANSUK, Ph.D.)

--- Thesis Advisor

(Associate Professor SAWEKCHAI TANGARAMVONG, Ph.D.)

--- External Examiner

(Assistant Professor Pakawat Sancharoen, Ph.D.)

โสวิภู เมือง : การออกแบบขนาดและรูปร่างของโครงสร้างอย่างเหมาะสมที่สุดด้วยวิธีการของกลุ่มอนุภาคที่ปรับปรุงแบบเบ็ดเสร็จ. (SIMULTANEOUS SIZE AND SHAPE STRUCTURAL OPTIMIZATION USING ENHANCED COMPREHENSIVE LEARNING PARTICLE SWARM OPTIMIZATION) อ.ที่ปรึกษาหลัก : เสวกชัย ตั้งอร่ามวงศ์

การศึกษานี้จะนำเสนอการเพิ่มประสิทธิภาพของโครงสร้างที่มีขนาดพื้นที่หน้าตัดและรูปร่างโครงสร้างให้เหมาะสมพร้อมกันด้วยโครงสร้างข้อหมุนโดยใช้อัลกอริทึมเมตาฮิวริสติกคือ การเพิ่มประสิทธิภาพการเรียนรู้เชิงลึกของกลุ่มอนุภาคซึ่งใช้อัลกอริทึม CLPSO โดย CLPSO นั้นมีที่มาจากการเพิ่มประสิทธิภาพของกลุ่มอนุภาค (PSO) ที่ได้รับความนิยมซึ่งถูกพัฒนามาจากพฤติกรรมการเคลื่อนไหวของฝูงนก และได้รับการปรับปรุงให้มีความสามารถในการหาค่าที่เหมาะสมได้ดี ณ จุดต่ำสุดสมบูรณ์ แต่ให้ค่าไม่ดี ณ จุดต่ำสุดสัมพัทธ์ ดังนั้น ECLPSO จึงแนะนำการปรับปรุงสองประการนี้โดยหาค่าที่เหมาะสมตามการรบกวนและความน่าจะเป็นในการเรียนรู้แบบปรับตัว เพื่อปรับปรุงความสามารถในการหาค่าที่เหมาะสมที่สุดและความน่าจะเป็นในการเรียนรู้ของแต่ละอนุภาคตามลำดับ วัตถุประสงค์ของการศึกษาคือเพื่อให้ได้น้ำหนักที่ต่ำที่สุดของโครงสร้างโครงข้อหมุนตามมาตรฐานภายใต้ข้อจำกัดในการออกแบบที่จำเป็นสำหรับแต่ละปัญหาโดยเฉพาะ การรวมตัวแปรของขนาดพื้นที่หน้าตัดและตัวแปรรูปร่างโครงสร้างของปัญหาเข้าด้วยกัน ผลลัพธ์ที่ได้จะดีกว่าการพิจารณาขนาดพื้นที่หน้าตัดที่เหมาะสมเพียงอย่างเดียว ในงานวิจัยนี้ใช้โปรแกรม Python ด้วยการทำงานซ้ำหลาย ๆ ครั้งเพื่อทำให้ได้ความเสถียรภาพและความถูกต้องของอัลกอริทึม ผลลัพธ์ที่ได้ของน้ำหนักที่ต่ำที่สุดและส่วนเบี่ยงเบนมาตรฐานจากการแก้ปัญหาจะนำมาเปรียบเทียบและสรุปผลกับอัลกอริทึมเมตาฮิวริสติกต่างๆ ในงานวิจัยอื่นๆ

สาขาวิชา	วิศวกรรมโยธา	ลายมือชื่อนิสิต ....
ปีการศึกษา	2564	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6272097121 : MAJOR CIVIL ENGINEERING
KEYWORD: Enhanced Comprehensive Learning, Particle Swarm Optimization, Size and Shape Optimization, Truss Structure

Soviphou Muong : SIMULTANEOUS SIZE AND SHAPE STRUCTURAL OPTIMIZATION USING ENHANCED COMPREHENSIVE LEARNING PARTICLE SWARM OPTIMIZATION. Advisor: Assoc. Prof. SAWEKCHAI TANGARAMVONG, Ph.D.

This study proposes a simultaneous size and shape structural optimization on truss structures using a metaheuristic algorithm, namely Enhanced Comprehensive Learning Particle Swarm Optimization (ECLPSO) which is based on the CLPSO algorithm. CLPSO itself was originated from a famous Particle Swarm Optimization (PSO) that was invented based on behaviour or movement of a bird flock. CLPSO is an improved PSO algorithm which has good exploration ability but is poor in exploitation. Thus, ECLPSO introduces of two enhancements, including perturbation-based exploitation and adaptive learning probabilities, to improve the exploitation ability and to adjust the learning probabilities of each particle, respectively. The objective of the study is to obtain the minimum weight of the benchmark truss structures under the design constraints required specifically for each problem. By combining size variables and shape variables of the problems, the optimizations shall produce better results than those of size optimizations alone. In this research, the optimizations are implemented in Python code and performed several times individually to conclude the robustness and accuracy of the algorithm. The results including the minimum weight and standard deviation of the solutions shall be compared and discussed with those of some other metaheuristic algorithms in the literatures.

Field of Study:	Civil Engineering	Student's Signature ....
Academic Year:	2021	Advisor's Signature ....

ACKNOWLEDGEMENTS

I would like to express my deep and endless gratitude to my parents who always support me, encourage me and care for me, especially during this research and this COVID-19 pandemic, even from afar.

I am extremely grateful to my advisor, Associate Professor Doctor Sawekchai Tangaramvong, for his valuable time, insightful direction, and support since the beginning of the research. This study would not have been possible without his instruction.

My grateful thanks to all committee members for providing suggestions and comments.

I would like to acknowledge the help of the academic staffs who spend their time to response and support me during my master's degree.

Finally, I must also thank my seniors, friends, and colleagues for their precious time, assistance, and ideas they spent on helping me throughout this study.

Soviphou Muong

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Metaheuristic algorithm is an efficient tool to solve the complex optimization problems. In this proposed study, one of the metaheuristic algorithms, namely the enhanced comprehensive learning particle swarm optimization, was addressed to apply to the benchmark structural problems. The scope of work and methodology will be stated with the expected results which can be compared with the outcome of the previous research and relevant studies.

1.1. Motivations

Metaheuristic algorithm consists of trial-and-error approach to solve the optimization problems. To obtain the solution, the metaheuristic algorithm needs two main features including the exploitation phase and the exploration phase. They became the basics features in all metaheuristic algorithms. In the past decades, many metaheuristic algorithms were introduced to improve the performance of algorithm while the optimization problems became more and more complex. Particle swarm optimization (PSO) (Kennedy and Eberhart 1995) is one of a popular swarm-intelligence based algorithms which was used in many real-world optimization problems. Since the algorithm was first introduced, tons of variant of the standard particle swarm optimization algorithm were created and used in many fields of optimization. Unfortunately, the solution of the algorithm might get trapped in the local optimum easily. Thus, the comprehensive learning particle swarm optimization (Liang, Qin et al. 2006) was proposed with the learning probabilities strategy to improve the exploration ability of the optimization performance. Although the CLPSO is good at exploration, its exploitation ability is poor. The enhanced comprehensive learning particle swarm optimization algorithm (ECLPSO) was proposed in 2014 (Yu and Zhang 2014) based on the original concept of the comprehensive learning particle swarm optimization with the application of two novel enhancements to (CLPSO) to improve both exploration and exploitation ability of the algorithm.

Structural optimization became a widely known field for research in structural engineering. The study turns out to be an important part of civil engineering in structural design. Since truss structure is one of a trendy type of structure for optimization, it shall be discussed as the structural optimization problems in this study. To achieve an economic design, size of structural members is an important factor. However, by considering both size and shape variables of structure, the optimization provides the more economical material design than the size optimization alone.

Hence, by employing the enhanced comprehensive learning particle swarm optimization (ECLPSO) on the simultaneous size and shape structural optimization problems, the outcome of optimization shall be expected to be valid and vital for this study.

1.2. Research Objective

The main objectives of this research are:

(1) To implement the enhanced comprehensive learning particle swarm optimization algorithm to the simultaneous size and shape optimization of the benchmark truss structures.

(2) To obtain minimum total weight of the structures under design specifications or design constraints as stated for each problem.

(3) To discuss the robustness and accuracy of the algorithm solution by doing comparison of the results with those processed by some other metaheuristic algorithm in relevant literatures.

1.3. Scope of Research

This research is proposed to execute with the following scope:

(1) A Python program is used for all the process of this study including structural analysis and optimization phase.

(2) Structural analysis shall be performed by utilizing the direct stiffness method.

(3) The enhanced comprehensive learning particle swarm optimization algorithm is implemented into the Python program.

(4) Benchmark 2D and 3D truss structures are selected for this study with design load and design constraints associated with each problem including stress constraints and/or displacement constraints.
(5) Size and shape optimization is performed simultaneously.
(6) Size and shape variables shall be picked from provided lists which consist of continuous variables and/or discrete variables as stated for each problem.
(7) Verify the results of the optimization and compare with those in literatures.

1.4. Methodology

Truss structural optimization is performed by executing the Python code created for this specific task and configured for each optimization problems. Python code includes input phase, analysis phase, optimization phase and output phase. The code starts with inputs of structural data including node coordinates, start and end node of bars, modulus of elasticity and material density. Linear analyses of benchmark truss structures are performed by utilizing the direct stiffness method. Stress of each member and total weight of structure are collected after structural optimization.

Several benchmark truss structures are optimized with the enhanced comprehensive learning particle swarm optimization algorithm which is implemented in Python code with a proposed number of total population and a total number of iterations. The total number of variables is the summation of size variables and shape variables as stated for each benchmark problems. The optimization process initiates from first iteration until it reaches the maximum numbers of iterations to obtain the minimum total of weight of structure. The optimization variables shall be selected within a provided list or the limit. The design constraints are checked at each iteration, so that there are no constraint violations for result of optimization. The optimization process shall be executed many times to find a best optimization result, mean result and the standard deviation of those result as well. To finalize the outcome of optimization, the results shall be discussed and compared with those in literatures.

CHAPTER 2
LITERATURE REVIEW

To utilize the enhanced comprehensive learning particle swarm optimization algorithm to the simultaneous size and shape structural optimization problems, historical background studies shall be presented.

2.1. Direct Stiffness Method

Direct stiffness method or matrix stiffness method has been used in many applications of structural analysis. The method is commonly suited for implementation to computer-aid structural analysis of complex structures including both statically determinate and indeterminate structure. To perform structural analysis, all necessary input data are required and shall be input in the form of matrices or array as specified in Python code.

Those data can be divided into six categories (Kassimali 2011):

- Joint data
- Support data
- Material property data
- Cross-sectional property data
- Member data
- Load data

The direct stiffness method can be performed as the following step-by-step procedures:

(1) Prepare analytical model and collect all input data in matrix forms.

(2) Formulate the structure stiffness matrix K

(3) Determine the joint displacements by substituting the known force P and structure stiffness matrix K into the equation $P = KU$ where U is the joint displacement array.

(4) Compute member end displacements and end forces, and support reactions.

(5) Verify the result by using the equilibrium equations. (For planar frame: $\Sigma F_x = 0$, $\Sigma F_y = 0$, and $\Sigma M = 0$)

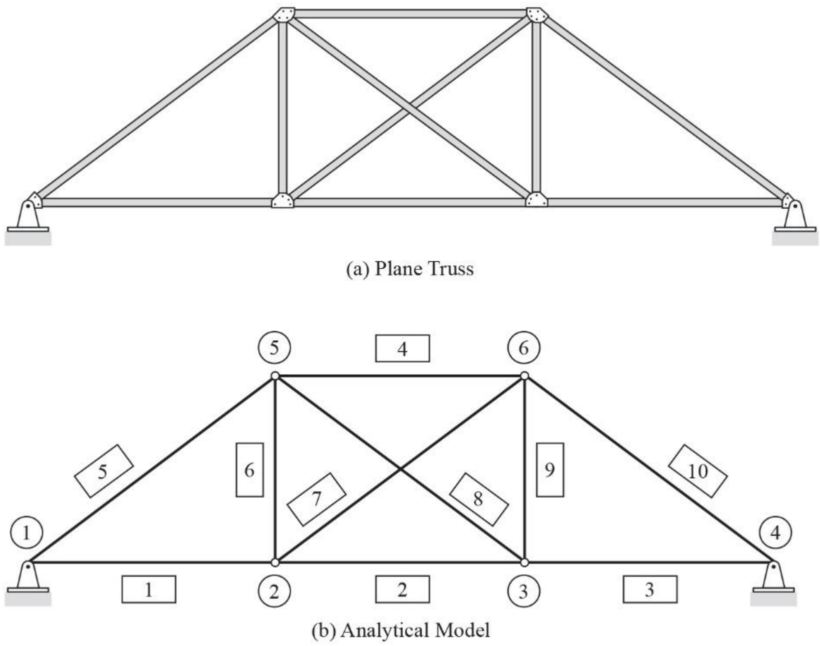

Figure 2.1. Plane truss and analytical model (Kassimali 2011)

2.2. Metaheuristic Algorithm

The optimization algorithm is a vital part of optimization process. "Meta-" in metaheuristic algorithm means beyond or higher level (Gandomi, Yang et al. 2013). All metaheuristic algorithms use trial and error method to deliver satisfactory solutions to optimization problems. The complexity of the problems can cause the performance of the algorithm to produce solution slower or inaccuracy. Many metaheuristic algorithms were proposed by utilizing different strategies to produce better solutions in less time. However, the best solutions may not be found easily.

Metaheuristic algorithms consist of exploitation and exploration phase. Exploitation is an ability to search for solutions in local area by using the information from a good solution in that area. Meanwhile, exploration is an ability to explore the search space on the global scale to produce global optimum. The differences in ability of exploitation and exploration of metaheuristic algorithms can cause the accuracy of the solutions.

Metaheuristic algorithms can be classified as population-based or trajectory-based. Particle swarm optimization (PSO) is considered a swarm-intelligence-based algorithm and a population-bases algorithm.

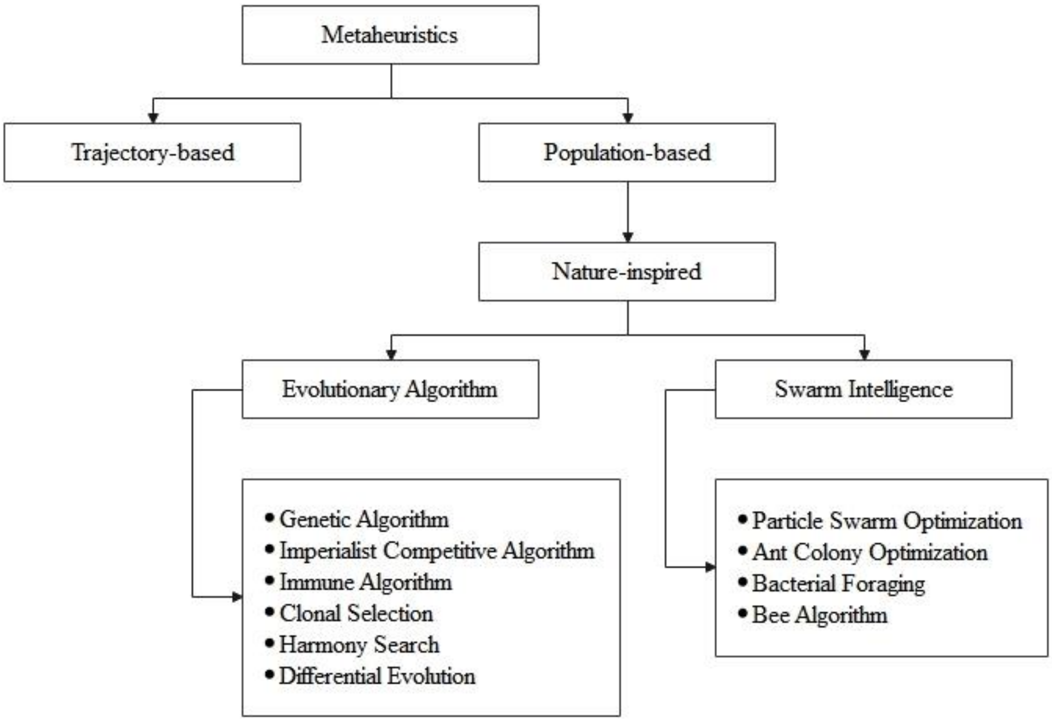

Figure 2.2. Metaheuristic algorithm classification (León-Aldaco, Calleja et al. 2015)

2.3. Particle Swarm Optimization

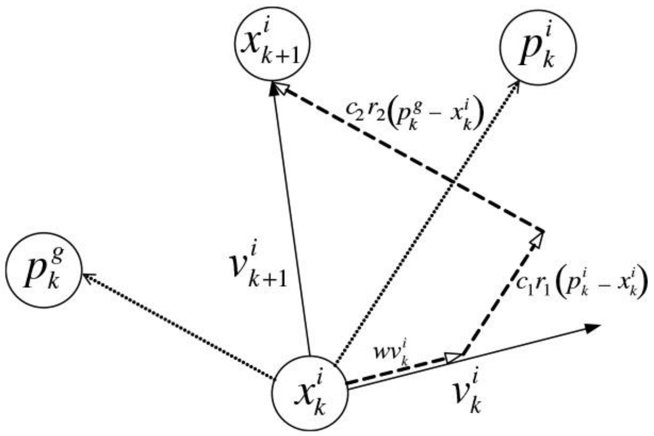

Figure 2.3. PSO position and velocity update (Perez and Behdinan 2007)

Particle swarm optimization algorithm (PSO) was first introduced in 1995 (Kennedy and Eberhart 1995). The algorithm is invented based on the behaviour or movement of a bird flock. PSO utilizes many argents or particles to search for acceptable solution in a search space. The idea of the algorithm is to share information of each particle to one another every iteration. Thus, if one particle has a better position than others, this particle holds a vital information that lets other particles follow its position. To apply the concept of algorithm to the mathematical problems, two essential equations for the d th dimension of particle i are utilized which are a velocity equation (Eq. 2.1) and a position equation (Eq. 2.2) as follows (Shi 2001):

$$V_i^d = wV_i^d + c_1 rand1_i^d (pbest_i^d - X_i^d) + c_2 rand2_i^d (gbest^d - X_i^d) \tag{2.1}$$

$$X_i^d = X_i^d + V_i^d \tag{2.2}$$

where $X_i = (X_i^1, X_i^2, ..., X_i^D)$ is the particle i position.

$V_i = (V_i^1, V_i^2, ..., V_i^D)$ is the particle i velocity.

$rand1_i^d$ and $rand2_i^d$ represent random numbers in the range [0, 1]

$pbest_i^d$ is a personal best position in the previous iteration for particle i

$gbest_i^d$ is the global best position of whole population

$c1$ and $c2$ are the acceleration constants which both equal to 2 in this study

w in the inertial weight which can be computed by using Eq. 2.3

$$w = 0.9 - \frac{k}{k_{max}}(0.9 - 0.4) \tag{2.3}$$

where k and k_{max} are iteration number and maximum number of iterations, respectively.

The algorithm procedure starts with initial positions with associated velocity of each particle which are randomly located anywhere in a search space. The personal best and global best position can be defined using initial position information. The algorithm initializes the first iteration in the next step. The velocity and position of each particle are updated by utilizing Eq. 2.1 and Eq. 2.2. However, the velocity and position of each particle of d th dimension shall be within limit of search space and velocity, respectively. The maximum value of velocity V_{max}^d normally equals to 20% of

$X^d_{max} - X^d_{min}$ and the minimum value V^d_{min} shall equal to $-V^d_{max}$ (Yu and Zhang 2014). Finally, the fitness value shall be compared with previous one to define the personal best position and global best position. The steps shall be repeated until the algorithm reaches the maximum number of iterations.

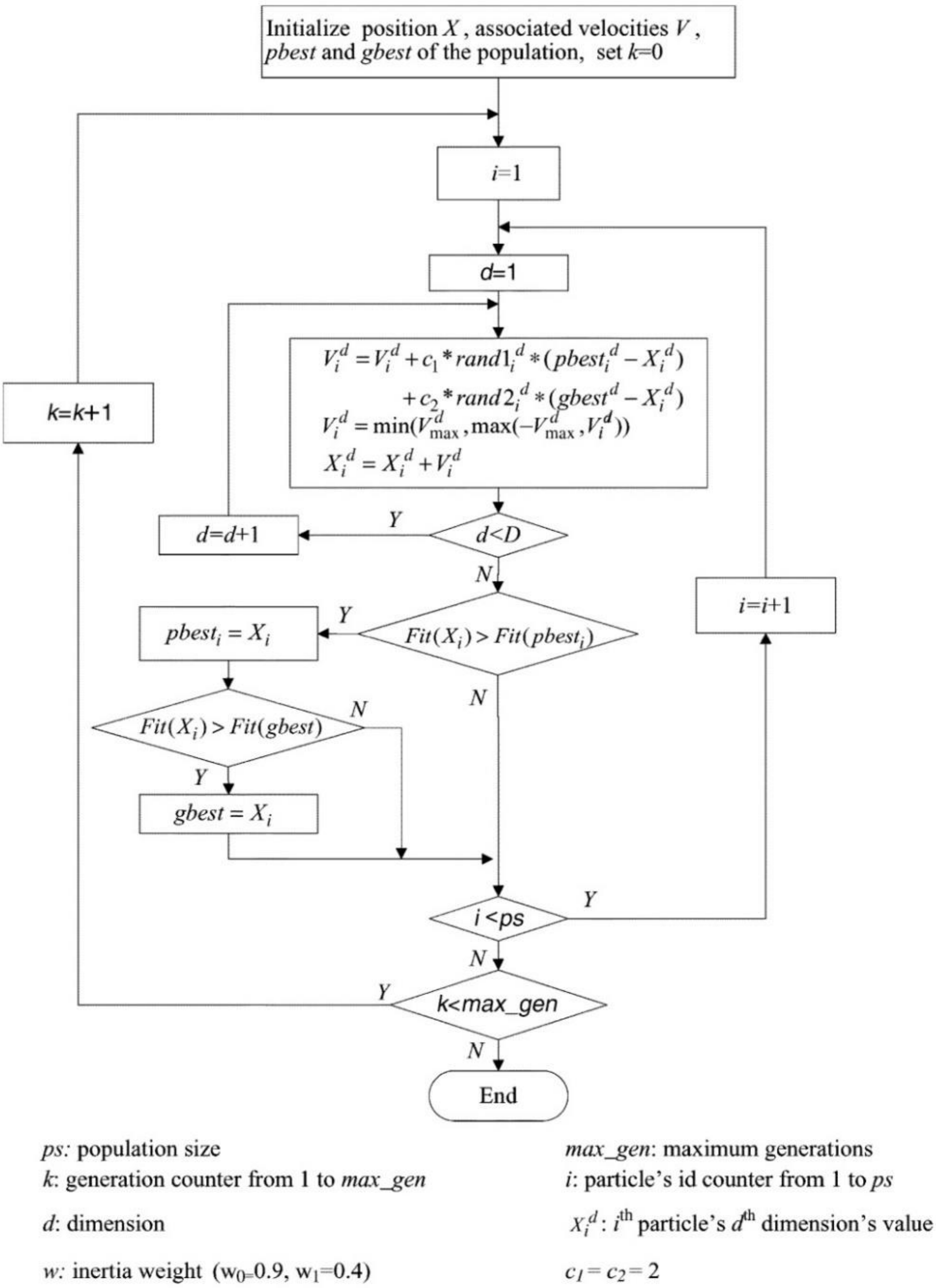

Figure 2.4. Flowchart of the standard PSO algorithm (Liang, Qin et al. 2006)

The simplicity of the standard PSO makes this algorithm popular and it was used in many real-world optimization problems. However, there is a drawback of this algorithm which turns out to be a major disappointment in yielding good solutions of complex optimization problems. The main drawback of this algorithm is the ability to get out of local optimal region. All particles in complex problems where there are many local optimums within a search space can be trapped in those regions forever. Thus, the outcome of optimization might not be good or even acceptable. Since the creation of standard PSO, there are many variants of PSO were proposed in the past decades to solve this problem. Even though new strategies or approaches are implemented to the standard PSO, the concept of algorithm is still the same.

2.3.1. Initialization

The initial positions and velocities of the particles shall be generated randomly within the limits, $[X^d_{\max}, X^d_{\min}]$ and $[V^d_{\max}, V^d_{\min}]$, respectively. To generate the random particles, the following equations can be used (Venter and Sobieszczanski-Sobieski 2004).

$$x_0^i = x_{\min} + r_1\left(x_{\max} - x_{\min}\right) \tag{2.4}$$

$$v_0^i = \frac{x_{\min} + r_2\left(x_{\max} - x_{\min}\right)}{\Delta t} \tag{2.5}$$

where r_1 and r_2 are random numbers between 0 and 1.

2.3.2. Inertia Weight

The inertia weight was not applied in the original equation of PSO but later it was used to improve the convergence speed of the optimization solution which was introduced in 1998 (Shi and Eberhart 1998). The inertia weight balances the global search ability and local search ability of PSO algorithm. When inertia weight is small ($w < 0.8$), PSO becomes a local search algorithm. Thus, if small inertia weight is used within initial search space, the solution can converge fast but might not be global optimum. On the contrary, PSO is more like a global search algorithm when the inertia weight is large ($w > 1.2$). To have the best change of obtaining the global optimum, the value of inertia weight shall be adjusted dynamically by using a large value at initial search and decrease linearly until the end of iteration. Equation below can be employed.

$$w_{k+1} = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} \times k \tag{2.6}$$

where $w_{\max}$ and $w_{\min}$ are set to 0.9 and 0.4, respectively.

2.3.3. Variable Bounds

The limitation of variables is vital for optimization problem to keep the variable stay in search space. The limit of velocities and positions shall be selected to prevent the particles move outside of the bounds. $X_{\max}^{d}$ and $X_{\min}^{d}$ are the position bounds of the particles; whilst $V_{\max}^{d}$ and $V_{\min}^{d}$ are the velocity bounds and they can be limited as shown in equations below.

$$x_i^d = \min\left(x_{\max}^d, \max\left(x_{\min}^d, x_i^d\right)\right) \tag{2.7}$$

$$v_i^d = \min\left(v_{\max}^d, \max\left(v_{\min}^d, v_i^d\right)\right) \tag{2.8}$$

2.4. Comprehensive Learning PSO

In the original PSO, the algorithm let the particles learn from the global best value which may or may not be near the global optimum of objective function. In a case, where the current global best is far from global optimum, the algorithm might not yield a good solution because the particles probably stuck in a local optimum which the current global best located at. To improve the standard PSO algorithm, comprehensive learning particle swarm optimization (CLPSO) was introduced (Liang, Qin et al. 2006). The algorithm is one of numerous variants of the standard PSO. A new learning strategy was proposed to let each particle learns from information of personal best values in the population and to prevent premature convergence. By using this strategy, the velocity equation now is based on the personal best of the particles.

The velocity of each particle can be updated by using the equation as follows:

$$V_i^d = wV_i^d + c_1 \times rand1_i^d (pbest_{fi(d)}^d - X_i^d) + c_2 \times rand2_i^d (gbest^d - X_i^d) \tag{2.9}$$

where $X_i = (X_i^1, X_i^2, ..., X_i^D)$ is the particle i position.

$V_i = (V_i^1, V_i^2, ..., V_i^D)$ is the particle i velocity.

$f_i = [f_i(1), f_i(2), ..., f_i(D)]$ defines which particles' personal best the particle i should follow. The exemplar index f_i can be determined depending on the learning probability Pc.

$pbest_{fi(d)}^{d}$ is personal best position of particle with index $f_i(d)$ which can be its own personal best.

$gbest^d$ is global best position of population.

c_1 and c_2 is the acceleration constant which equals to 2.

$rand1_i^d$ and $rand2_i^d$ is a random number in range [0, 1]. This value is different for each dimension and particle of the population.

w is the inertia weight which can be computed as describe in the standard PSO algorithm.

2.4.1. Exemplar Index

To define the exemplar index f_i, a tournament selection procedure shall be employed. The procedure can be followed as shown in flowchart below.

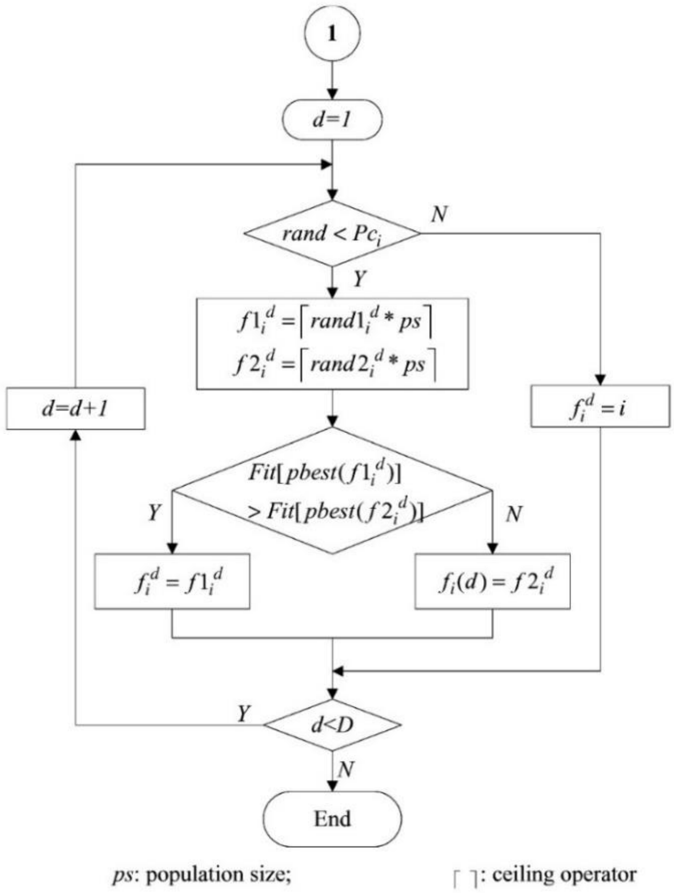

Figure 2.5. Selection of exemplar index for particle i (Liang, Qin et al. 2006)

For each dimension, the tournament selection can be employing as follows:

(1) A random number in range of [0,1] is generated and compared with the learning probability which can be calculated by using Eq. 2.10.

(2) If the learning probability is smaller, the exemplar index f_i is set as particle i. Otherwise, two random particles are selected excluding the particle whose velocity is updated and proceed to next step.

(3) Fitness values of those two particles are compared. The index of particle with better fitness value is selected as the exemplar index.

(4) If all exemplars of a particle are its own personal best, a random dimension of that particle is set to learn from another personal best of the corresponding dimension randomly.

2.4.2. Learning Probabilities

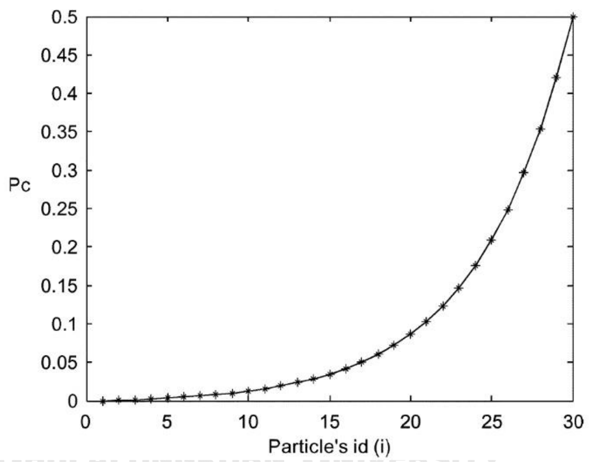

Figure 2.6. Each particle's Pc with a population size of 30.

The learning probability Pc can be computed by using equation as follows:

$$Pc_i = 0.05 + 0.45 \frac{\left(\exp\left(\frac{10(i-1)}{ps-1}\right) - 1\right)}{\left(\exp(10) - 1\right)} \tag{2.10}$$

where ps is the total number of population.

To prevent each particle learns from poor exemplar, a refreshing gap m is employed. The refreshing gap will let the particles learn from one exemplar until the fitness value stops improving for a certain number of generations. In this study, m is set to 7.

The CLPSO can be followed as shown in flowchart below.

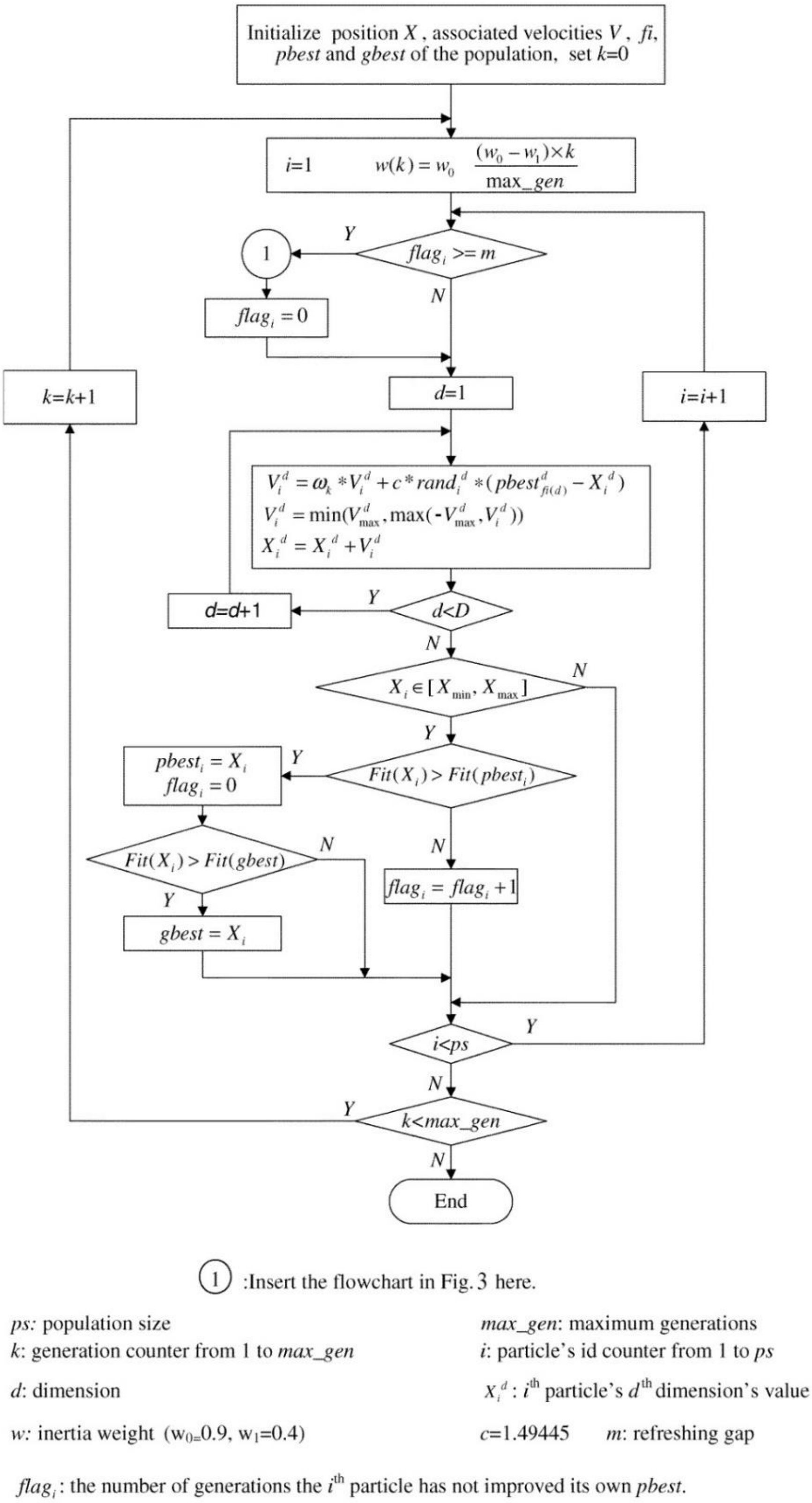

Figure 2.7. Flowchart of CLPSO algorithm (Liang, Qin et al. 2006)

The concept of the comprehensive learning particle swarm optimization algorithm is to let the particles within the search to learn from information of other particles in the population. By doing so, the algorithm improves the exploration ability. Thus, the algorithm can yield a better solution and more accurate than the standard PSO when it faces complex optimization problems.

2.5. Enhanced Comprehensive Learning PSO

Comprehensive learning particle swarm optimization algorithm is good at exploration; however, it is poor at exploitation. As a result, the solution of optimization lacks accuracy. Thus, two enhancements to the algorithm of CLPSO was proposed to improve the algorithm. The enhanced CLPSO algorithm (Yu and Zhang 2014) was introduced in 2014 which was based on the original concept of CLPSO algorithm but with the two enhancement, ECLPSO could yield better results. Two techniques, namely perturbation-based exploitation, and adaptive learning probabilities are developed within the ECLPSO algorithm. These concepts can be described as follows.

2.5.1. Perturbation-based Exploitation

To improve the exploitation of particle swarm optimization, the enhanced comprehensive learning particle swarm optimization algorithm implements a normative knowledge structure as shown in Table 2.1. The normative knowledge is dimensional intervals to all the personal best positions of population. To employ the perturbation-based exploitation-based term, a certain condition based on the normative knowledge shall be made to decide when to perform exploitation effectively and which region to focus on.

Table 2.1. Structure of the normative knowledge (Yu and Zhang 2014)

Dimension	1	2	...	D
Present dimensional lower bound	$\underline{P_1}$	$\underline{P_2}$	...	$\underline{P_D}$
Present dimensional upper bound	$\overline{P_1}$	$\overline{P_2}$	...	$\overline{P_D}$

where $\underline{P_d}$ and $\overline{P_d}$ are respectively the lower and upper bounds of all personal best positions on the d th dimension.

The value of the lower and upper bounds can be calculated as follows.

$$\underline{P_d} = \min\left\{P_{1,d}, P_{2,d}, ..., P_{N,d}\right\} \tag{2.11}$$

$$\overline{P_d} = \max\left\{P_{1,d}, P_{2,d}, ..., P_{N,d}\right\} \tag{2.12}$$

The perturbation-based exploitation term mainly improves the exploitation and accuracy of CLPSO algorithm. However, the procedure of this algorithm is as same as that of CLPSO algorithm. The perturbation-based exploitation term shall be applied when the Eq. 2.13 is true. Thus, the velocity shall be updated by utilizing Eq. 2.14 which includes the perturbation-based exploitation term.

$$\begin{cases} \overline{P_d} - \underline{P_d} \le \alpha\left(\overline{X_d} - \underline{X_d}\right) \\ \overline{P_d} - \underline{P_d} \le \beta \end{cases} \tag{2.13}$$

$$\begin{aligned} V_i^d = w_{PbE}V_i^d + c_1 \times r1_i^d \left(pbest_{fi(d)}^d + \eta\left(\frac{\overline{P_d} - \underline{P_d}}{2} - pbest_{fi(d)}^d\right) - X_i^d \right) \\ + c_2 \times r2_i^d (gbest^d - X_i^d) \end{aligned} \tag{2.14}$$

where α is relative ratio which equals to 0.01.

β is small absolute bound which set as 2.

η is the perturbation coefficient which can be generated randomly from a normal distribution with mean 1 and standard deviation 0.65.

w_{PbE} is the inertia weight but only used for this case which is fixed at 0.5.

If the Eq. 2.13 is not true, we shall use the velocity equation from the CLPSO algorithm which is shown as Eq. 2.9.

2.5.2. Adaptive Learning Probabilities

Learning probabilities strategy is a vital method for finding an exemplar index of the comprehensive learning particle swarm optimization algorithm. However, in CLPSO algorithm, the learning probabilities are only based on particles' index and does not change during optimization iteration. Thus, this static strategy might create difficulties of convergence. To improve this strategy, the ECLPSO algorithm was introduced with a new adaptive learning probabilities which was adjusted dynamically by the ranking information of personal best particles.

The adaptive learning probabilities of ECLPSO will replace the original equation in CLPSO by the Eq. 2.15 as follows.

$$Pc_i = L_{\min} + \left(L_{\max} - L_{\min}\right)\frac{\exp\left(\dfrac{10\left(K_i - 1\right)}{ps - 1}\right) - 1}{\exp\left(10\right) - 1} \tag{2.15}$$

$$L_{\max} = L_{\min} + 0.25 + 0.45\log_{(D+1)}\left(M_k + 1\right) \tag{2.16}$$

where $L_{\max}$ can be computed by using the Eq. 2.16 above.

$L_{\min}$ is fixed at 0.05.

M_k is the number of the dimensions where the Eq. 2.13 is satisfied before or during the iteration k.

The Eq. 2.15 is like Eq. 2.10 of CLPSO algorithm but included with ranking parameter K_i. The ranking information K_i can be defined by sorting the personal best fitness value in ascending order. Thus, if a particle has the best fitness value compared to other particles, its rank shall be 1 (i.e. $K_i = 1$). On the other hand, if it has the worst fitness value, its rank equals to total number of population ps.

To prevent premature convergence, adjustment to the parameter $L_{\max}$ is needed. A small $L_{\max}$ seems good for exploration while a large $L_{\max}$ helps in exploitation. Hence, Eq. 2.16 was proposed to balance exploration and exploitation of the algorithm. As we can see from Eq. 2.16, $L_{\max} = 0.3$ which is the minimum value when $M_k = 0$. Otherwise, $L_{\max} = 0.75$ which is the maximum value when $M_k = D$.

With the adaptive learning probabilities strategy, the algorithm can improve the particles' exploitation ability to accelerate convergence. This is an important role of ECLPSO which can improve the exploitation weakness of CLPSO algorithm.

The enhanced comprehensive learning particle swarm optimization algorithm possesses a similar procedure to CLPSO algorithm. The flowchart of ECLPSO is shown as follows.

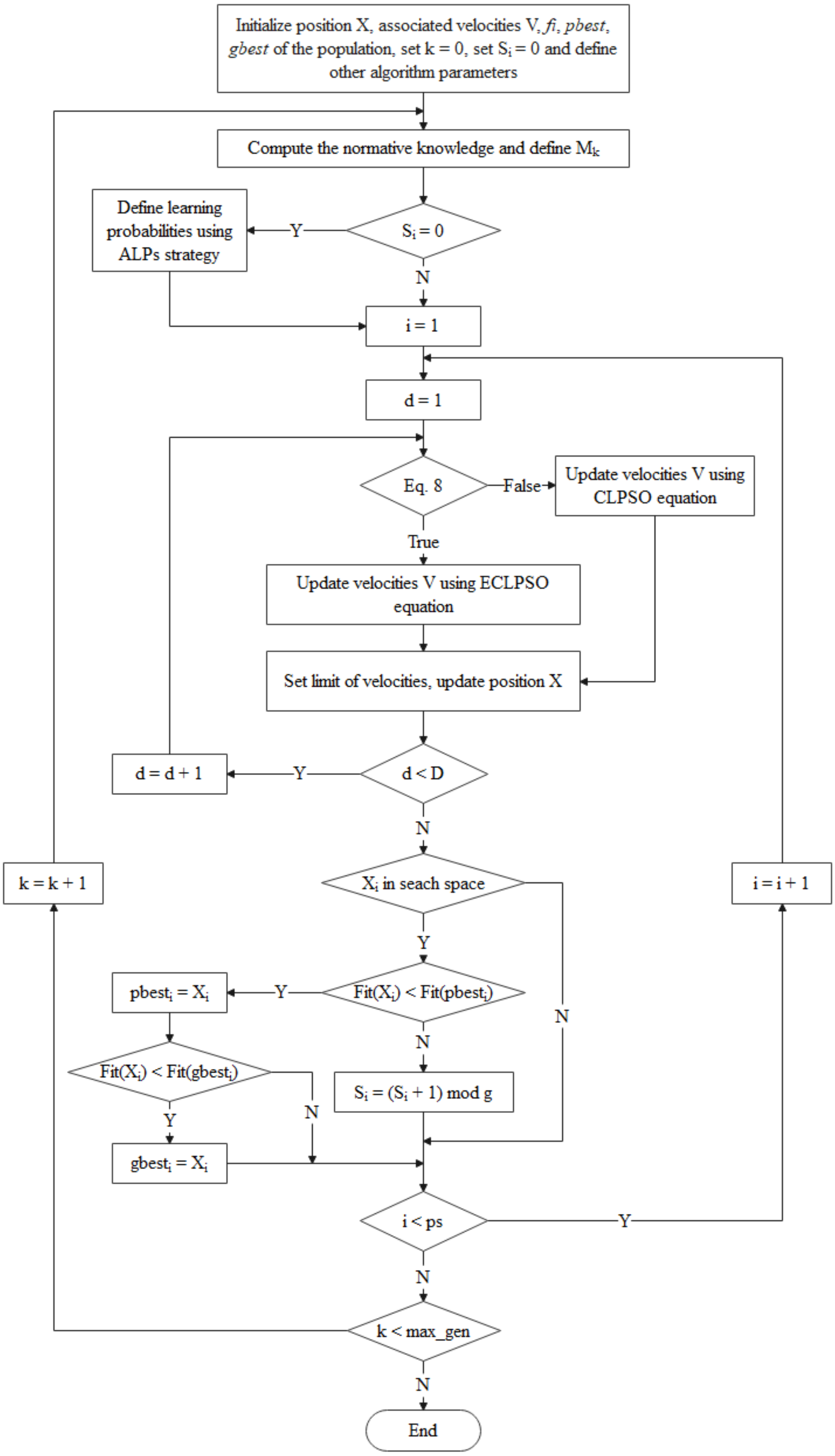

Figure 2.8. Flowchart of ECLPSO algorithm

2.6. Comparison of PSO, CLPSO, and ECLPSO on benchmark function

The improvement of CLPSO and ECLPSO algorithm were tested in many benchmark functions and compared with solution from the conventional PSO algorithm. In this study, the optimization of two benchmark functions, which are sphere function and Rosenbrock's function, will be shown below.

Table 2.2. Expressions and information of the benchmark functions.

Function description	x^*	$f(x^*)$	Search space	Initialization space
Sphere, $f_1(x)=\sum_{i=1}^{D} x_i^2$	$\{0\}^D$	0	$[-100,100]^D$	$[-100,50]^D$
Schwefel's, $f_2(x)=418.9829D-\sum_{i=1}^{D} x_i \sin\left(\lvert x_i\rvert^{\frac{1}{2}}\right)$	$\{420.96\}^D$	0	$[-500,500]^D$	$[-500,500]^D$

The first problem is the sphere function which is considered a unimodal function and is easy to solve. The second problem is a unrotated multimodal problem called Schwefel's function. This function is more complex due the number of local optima being so large. The solution might be hard to obtain if many particles stuck deep in local optima.

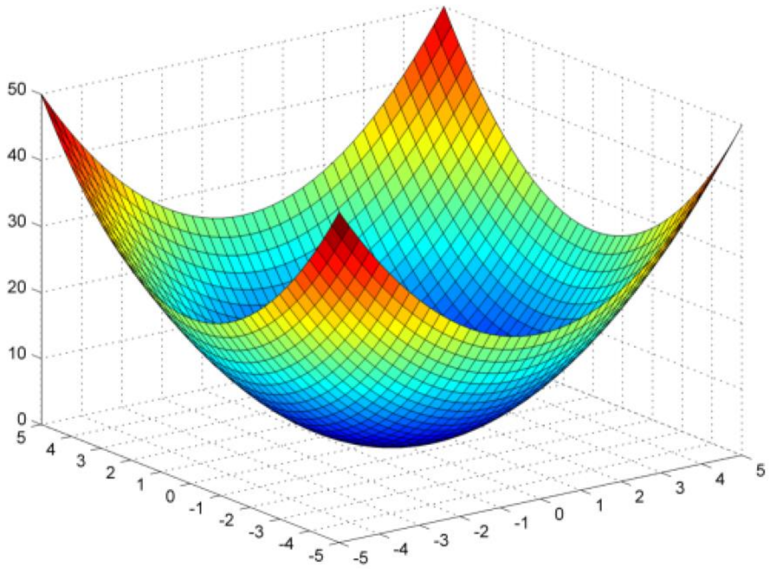

Figure 2.9. Sphere function (Dutu, Mauris et al. 2015)

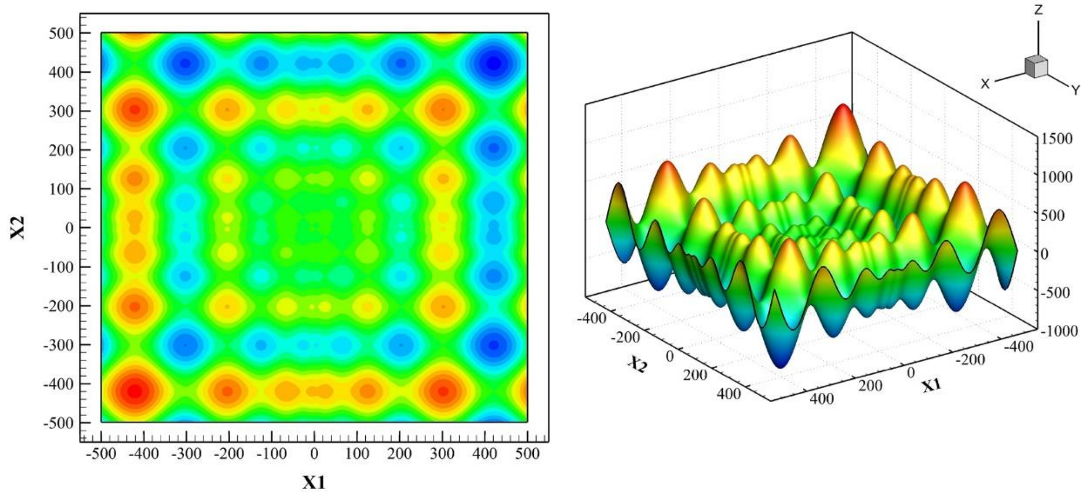

Figure 2.10. Schwefel's function (Omidi and Mazaheri 2020)

2.6.1. CLPSO Algorithm

The optimizations were performed with many PSO algorithms as listed below. For this test, the CLPSO algorithm was used with 10 dimensions and 10 particles or agents to find the global optimum. The maximum evaluations (Fes) were set to 30,000 and the optimization was performed 30 times. The results will be presented for both problems with the graph of convergence curve, the mean values and standard deviation.

The algorithms and parameters settings for the test problems: (Liang, Qin et al. 2006)

- PSO with inertia weight (PSO-w)
- PSO with constriction factor (PSO-cf)
- Local version of PSO with inertia weight (PSO-w-local)
- Local version of PSO with constriction factor (PSO-cf-local)
- UPSO
- Fully informed particle swarm (FIPS)
- FDR-PSO
- CPSO-H
- CLPSO

As we can observe from results below, CLPSO does not yield better for unimodal, sphere function but performs the best for multimodal function, Schwefel's function.

Table 2.3. Results of CLPSO improvement over PSO (Liang, Qin et al. 2006)

Alogorithm	Sphere function	Schwefel's function
PSO-w	7.96e-051 ± 3.56e-050	3.20e+002 ± 1.85e+002
PSO-cf	9.846e-105 ± 4.21e-104	9.87e+002 ± 2.76e+002
PSO-w-local	2.13e-035 ± 6.17e-035	3.26e+002 ± 1.32e+002
PSO-cf-local	1.37e-079 ± 5.60e-079	8.78e+002 ± 2.93e+002
UPSO	**9.84e-118 ± 3.56e-117**	1.08e+003 ± 2.68e+002
FDR	2.21e-090 ± 9.88e-090	8.51e+002 ± 2.76e+002
FIPS	3.15e-030 ± 4.56e-030	7.10e+001 ± 1.50e+002
CPSO-H	4.98e-045 ± 1.00e-044	2.13e+002 ± 1.41e+002
CLPSO	5.15e-029 ± 2.16e-028	**0 ± 0**

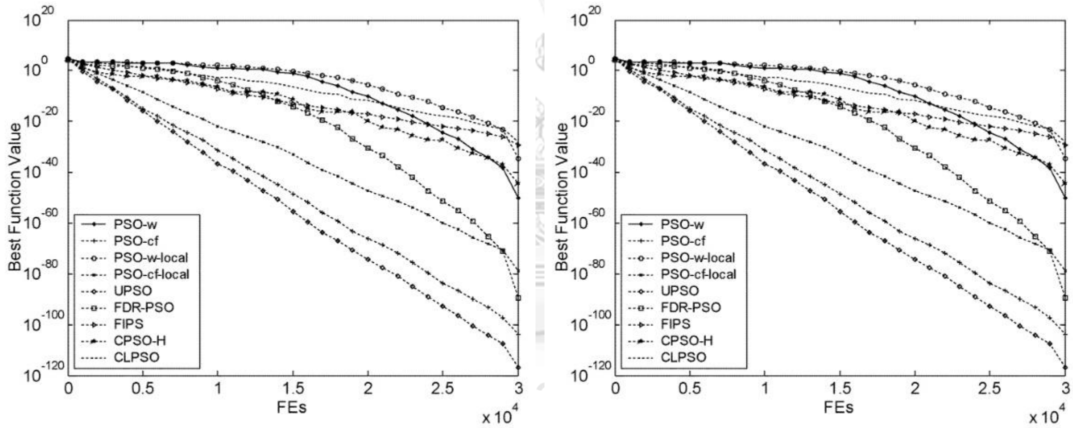

Figure 2.11. Convergence characteristics of sphere function (left) and Schwefel's function (right)

2.6.2. ECLPSO Algorithm

The experiments were tested with 30 dimensions and 40 particles. The maximum number of fitness evaluation (Fes) is 200,000 for each run on each function. The optimization were performed by using many algorithms such as CLPSO, OLPSO-G, OLPSO-L, SPSO from (Zhan, Zhang et al. 2011) and ECLPSO. The problems were solved 25 times independently. The solution of optimization shall be shown below with the mean value and standard deviation.

Table 2.4. Results of ECLPSO improvement over CLPSO (Yu and Zhang 2014)

Algorithm	Sphere function		Schwefel's function	
	Mean	SD	Mean	SD
CLPSO	2.56e-14	8.77e-14	3.82e-4	2.42e-13
OLPSO-G	4.12e-54	6.34e-54	3.84e2	2.17e2
OLPSO-L	1.11e-38	1.28e-38	3.82e-4	0
SPSO	2.29e-96	9.48e-96	3.14e3	7.81e2
ECLPSO	**1.00e-96**	**3.01e-96**	**3.82e-4**	**0**

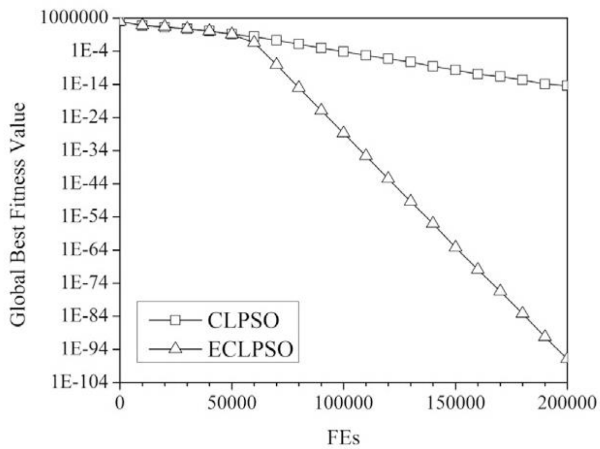

Figure 2.12. Convergence characteristics of CLPSO and ECLPSO of sphere function

The benefit of perturbation-based exploitation makes the solution of optimization problem better. If we look at convergence characteristics of CLPSO and ECLPSO, we can see the difference in solution of the problem is after around 50,000 fitness evaluations, the ECLPSO algorithm starts to produce a better result.

2.7. Size and Shape Truss Structural Optimization

The purpose of the structural optimization is to obtain a minimum total weight of the truss structure by getting the optimal cross-sectional areas and the best nodal coordinates without violating the design constraints. Combined size and shape optimization yields better results than those of size optimization or shape optimization alone. The structural optimization contains two types of variables which are the size (cross-sectional area) variables and shape (nodal location) variables.

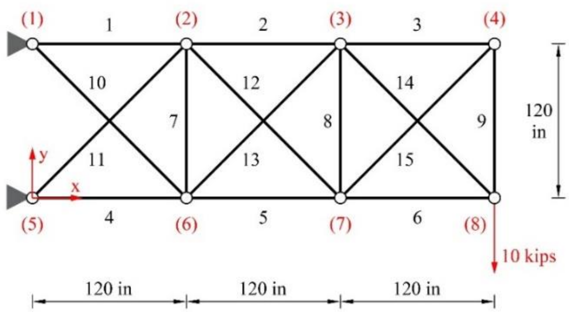

Figure 2.13. Example of optimization problem of 15-bar planar truss structure

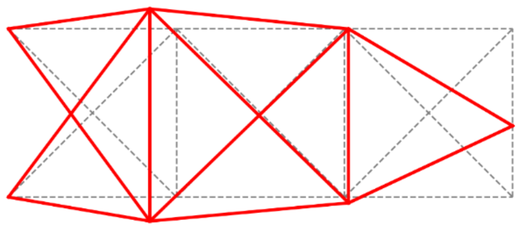

Figure 2.14. Sample of optimal shape of truss structure

After an optimization of truss structure in Figure 2.13, we can obtain the optimal shape of truss structure as shown in Figure 2.14. The red line represents the optimal shape of truss structure while the grey line indicates the original shape. The nodal coordinates move every iteration to find the optimal shape, however they shall be within the geometry limitation. Since the nodal coordinates change, the length of each member shall be computed every structural analysis as well.

2.8. Constraint Handling

The solution of optimization shall be under the design constraint for each problem as stated above. To handle the design constraints, a penalty function method is employed which can be formulated as follows (Jawad, Mahmood et al. 2021).

$$W' = W\left(X_A, X_G\right)\varphi_p K \tag{2.17}$$

$$\varphi_p = \left(1 + C\right)^{\varepsilon} \tag{2.18}$$

where W' is the total weight of truss structure after handling the design constraints.

φ is a multiplication coefficient which can be computed using Eq. 2.18.

K and ε are respectively the penalty constant and penalty exponent which are both fixed at 1 in this study.

The parameter C is for measuring the violation of penalty constraints which is the summation of the design constraints' penalties including stress, displacement and buckling constraints. The parameter C can be calculated by using equation below.

$$C = \sum_{j=1}^{m} C_{\delta}^{j} + \sum_{i=1}^{n} C_{\sigma}^{i} + \sum_{i=1}^{n} C_{\sigma(buckling)}^{i} \tag{2.19}$$

The displacement, stress, and buckling constraints ($C_{\delta}^{j}, C_{\sigma}^{i}, C_{\sigma(buckling)}^{i}$, respectively) shall be penalties for each iteration of the optimization process in the mathematical formula as follows.

2.8.1. Strength Constraint

Penalty of stress constraints:

$$\left.\begin{aligned} C_{\sigma}^{i} &= \left|\frac{\sigma_i - \sigma_{\min}}{\sigma_{\min}}\right| && \text{if} \quad \sigma_i < \sigma_{\min} \\ C_{\sigma}^{i} &= \left|\frac{\sigma_i - \sigma_{\max}}{\sigma_{\max}}\right| && \text{if} \quad \sigma_i > \sigma_{\max} \\ C_{\sigma}^{i} &= 0 && \text{if} \quad \sigma_{\min} \le \sigma_i \le \sigma_{\max} \end{aligned}\right\} \tag{2.20}$$

Penalty of buckling constraints:

$$\left.\begin{aligned} C_{\sigma(buckling)}^{i} &= \left|\frac{\sigma_i^{buckling} - \sigma_{\max}^{buckling}}{\sigma_{\max}^{buckling}}\right| && \text{if} \quad \sigma_i^{buckling} > \sigma_{\max}^{buckling} \\ C_{\delta}^{j} &= 0 && \text{if} \quad \sigma_i^{buckling} \le \sigma_{\max}^{buckling} \end{aligned}\right\} \tag{2.21}$$

2.8.2. Serviceability Constraint

Penalty of displacement constraints:

$$\left.\begin{aligned} C_{\delta}^{j} &= \left|\frac{\delta_j - \delta_{\min}}{\delta_{\min}}\right| && \text{if} \quad \delta_j < \delta_{\min} \\ C_{\delta}^{j} &= \left|\frac{\delta_j - \delta_{\max}}{\delta_{\max}}\right| && \text{if} \quad \delta_j > \delta_{\max} \\ C_{\delta}^{j} &= 0 && \text{if} \quad \delta_{\min} \le \delta_j \le \delta_{\max} \end{aligned}\right\} \tag{2.22}$$

CHAPTER 3
FORMULATION AND PROBLEM STATEMENT

3.1. Problem Statement

In this study proposal, 4 benchmark optimization problems are addressed. Each problems have specific properties as stated. There are 2 types of truss structures in this study which are the planar truss structure and the space truss structure. The planar truss structures include 3 different truss systems with 15, 18 and 47 elements, respectively. The space truss structure consists of 25 elements.

The optimization problems consider size (namely $X_A \in \Re^{n} = \{A_1, A_2, ..., A_n\}$) and shape (namely $X_G \in \Re^{ng} = \{G_1, G_2, ..., G_{ng}\}$) variables simultaneously to obtain the minimum total weight of truss structures. The weight minimization problem for the benchmark truss structures can be mathematically expressed as follows.

$$\left.\begin{array}{lll} \text{find} & X \in \Re^{n+ng} = \{X_A, X_G\} & \\ \text{minimize} & W(X_A, X_G) = \sum_{i=1}^{n} \rho_i L_i A_i & \\ \text{subject to} & \sigma_i^c \le \sigma_i \le \sigma_i^t & \text{for} \quad \forall i \in \{1, 2, ..., n\} \\ & \delta_{\min} \le \delta_j \le \delta_{\max} & \text{for} \quad \forall j \in \{1, 2, ..., m\} \\ & A_{\min} \le A_i \le A_{\max} & \text{for} \quad \forall i \in \{1, 2, ..., n\} \end{array}\right\} \tag{3.1}$$

where W is the total weight of the design structure defined as the function of member density ρ_i.

L_i is the physical length of each member.

A_i defines the cross-sectional area of each member.

m is the total number of degrees of freedom; n and ng are the total number of size and shape variables, respectively.

δ_j is the displacement at the j-th degree of freedom.

σ_i indicates the member stress.

The optimization problem is Eq. 3.1 minimizes the total weight of truss structure under the bounds on permissible compression σ_i^c and tension σ_i^t stresses, minimum $\delta_{\min}$ and maximum $\delta_{\max}$ displacements, and minimum $A_{\min}$ and maximum $A_{\max}$ areas.

3.1.1. 15-bar Planar Truss Structure

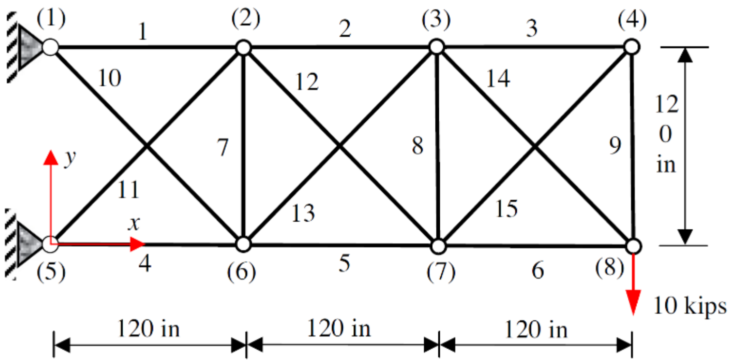

Figure 3.1. 15-bar planar truss structure

Table 3.1. Optimization data for the 15-bar planar truss structure

Objective function:	$\min \quad W\left(X_A, X_G\right)=\sum_{i=1}^{15}\rho_i L_i A_i$
Stress constraints:	$\begin{cases}\sigma_i^t \le 25 \text{ (ksi)} \\ \lvert\sigma_i^c\rvert \le 25 \text{ (ksi)}\end{cases}, \ i=1,2,\ldots,15$
Size variables:	$A_i \ , \ i=1,2,\ldots,15$
Shape variables:	$x_2=x_6 \ , x_3=x_7 \ , y_2 \ , y_3 \ , y_4 \ , y_6 \ , y_7 \ , y_8$
Permissible size variables: $A_i \in S = \{0.111, 0.141, 0.174, 0.220, 0.270, 0.287, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170, 19.180\}$ in ^{2}	
Limitation of shape variables: $100 \le x_2 \le 140$ $220 \le x_3 \le 260$ $100 \le y_2 \le 140$ $100 \le y_3 \le 140$ $50 \le y_4 \le 90$ $-20 \le y_6 \le 20$ $-20 \le y_7 \le 20$ $20 \le y_8 \le 60$	
Young modulus:	$E=10^4$ (ksi)
Material density:	$\rho=0.1$ (lb/in ^{3})

The optimization problem includes a 15-bar planar truss structure subjected to a vertical load (10 kips) at node 8 as shown in Figure 3.1. There are 15 discrete and 8 continuous design variables for the cross-sectional areas and the nodal coordinates, respectively. The discrete variables shall be selected from a list as shown in Table 3.1 and the continuous variables shall be limited as well. All structural members are subjected to stress limitation of ±25 (ksi) for both tension and compression, respectively.

3.1.2. 18-bar Planar Truss Structure

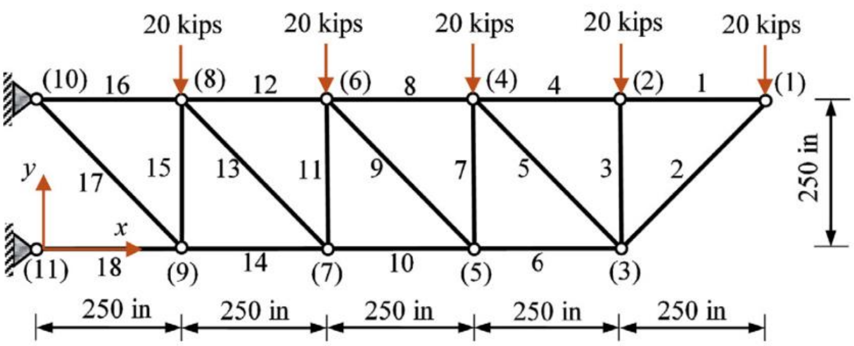

Figure 3.2. 18-bar planar truss structure

This benchmark problem considers 18-bar planar truss structure subjected to vertical forces (20 kips) at nodes 8, 6, 4, 2, and 1 as shown in Figure 3.2. The optimization problem also includes the buckling constraints with the buckling coefficient $K=4$ as stated in Table 3.2 below.

Table 3.2. Optimization data for the 18-bar planar truss structure

Objective function:	$\min \quad W(X_A, X_G)=\sum_{i=1}^{18}\rho_i L_i A_i$
Stress constraints:	$\begin{cases}\sigma_i^t \le 25 \text{ (ksi)} \\ \lvert\sigma_i^c\rvert \le 25 \text{ (ksi)}\end{cases}, \ i=1,2,\ldots,18$
Buckling constraints:	$\lvert\sigma_i^c\rvert \le KEA_i / L_i^2 \ , \ i=1,2,\ldots,18$
Size variables:	$A_1=A_4=A_8=A_{12}=A_{16}$, $A_2=A_6=A_{10}=A_{14}=A_{18}$, $A_3=A_7=A_{11}=A_{15}$, $A_5=A_9=A_{13}=A_{17}$

Shape variables:	x_3 , y_3 , x_5 , y_5 , x_7 , y_7 , x_9 , y_9
Permissible size variables:	$A_i \in S = \{2.00, 2.25, 2.50, ..., 21.25, 21.50, 21.75\}$ in^2
Limitation of shape variables:	
$775 \le x_3 \le 1225$	
$525 \le x_5 \le 975$	
$275 \le x_7 \le 725$	
$25 \le x_9 \le 475$	
$-225 \le y_3 , y_5 , y_7 , y_9 \le 245$	
Young modulus:	$E = 10^4$ (ksi)
Buckling coefficient:	$K = 4$
Material density:	$\rho = 0.1$ (lb/in^3)

3.1.3. 47-bar Planar Truss Structure

The size and shape optimization of the 47-bar planar truss structure consists of 44 design variables under stress and buckling constraints. The forces are applied at node 17 and 22 with value $F_x = 6$ (kips) and $F_y = -14$ (kips).

Table 3.3. Optimization data for the 47-bar planar truss structure

Objective function:	$\min \quad W(X_A, X_G) = \sum_{i=1}^{47} \rho_i L_i A_i$
Stress constraints:	$\begin{cases} \sigma_i^t \le 20 \text{ (ksi)} \\ \lvert\sigma_i^c\rvert \le 15 \text{ (ksi)} \end{cases}, \; i = 1, 2, ..., 47$
Buckling constraints:	$\lvert\sigma_i^c\rvert \le KEA_i / L_i^2$, $i = 1, 2, ..., 47$
Size variables:	$A_3 = A_1$, $A_4 = A_2$, $A_5 = A_6$, A_7 , $A_8 = A_9$, A_{10} , $A_{12} = A_{11}$, $A_{14} = A_{13}$, $A_{15} = A_{16}$, $A_{18} = A_{17}$, $A_{20} = A_{19}$, $A_{22} = A_{21}$, $A_{24} = A_{23}$, $A_{26} = A_{25}$, A_{27} , A_{28} , $A_{30} = A_{29}$, $A_{31} = A_{32}$, A_{33} , $A_{35} = A_{34}$, $A_{36} = A_{37}$, A_{38} , $A_{40} = A_{39}$, $A_{41} = A_{42}$, A_{43} , $A_{45} = A_{44}$, $A_{46} = A_{47}$
Shape variables:	$x_2 = -x_1$, $x_4 = -x_3$, $y_4 = y_3$, $x_6 = -x_5$, $y_6 = y_5$, $x_8 = -x_7$, $y_8 = y_7$, $x_{10} = -x_9$, $y_{10} = y_9$, $x_{12} = -x_{11}$, $y_{12} = y_{11}$, $x_{14} = -x_{13}$, $y_{14} = y_{13}$, $x_{20} = -x_{19}$, $y_{20} = y_{19}$, $x_{21} = -x_{18}$, $y_{21} = y_{18}$

Permissible size variables: $A_i \in S = \{0.1, 0.2, 0.3, ..., 4.8, 4.9, 5.0\}$ in^2	
Limitation of shape variables: x_i , $y_i \in \Re$	
Loads:	
Case 1: Node 17 and 22	$F_x = 6$ (kips), $F_y = -14$ (kips)
Case 2: Node 17	$F_x = 6$ (kips), $F_y = -14$ (kips)
Case 3: Node 22	$F_x = 6$ (kips), $F_y = -14$ (kips)
Young modulus:	$E = 3\times10^4$ (ksi)
Buckling coefficient:	$K = 3.96$
Material density:	$\rho = 0.3$ (lb/in^3)

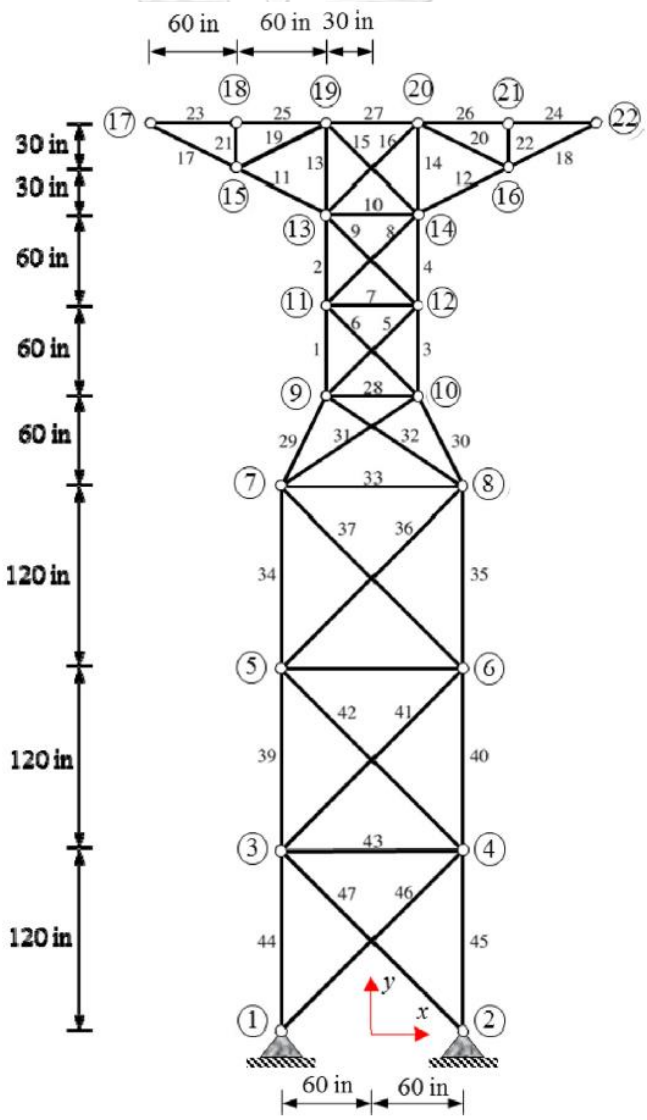

Figure 3.3. 47-bar planar truss structure

3.1.4. 25-bar Space Truss Structure

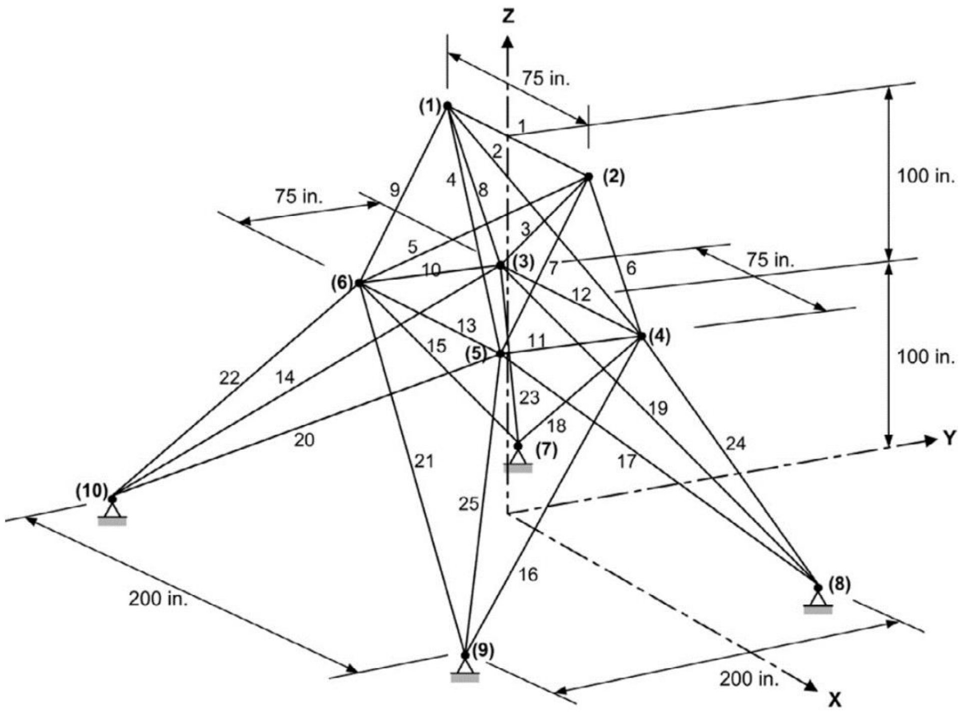

Figure 3.4. 25-bar space truss structure

Table 3.4. Optimization data of the 25-bar space truss structure

Objective function:	$\min \quad W\left(X_A, X_G\right) = \sum_{i=1}^{25} \rho_i L_i A_i$
Stress constraints:	$\begin{cases} \sigma_i^t \le 40 \text{ (ksi)} \\ \left\lvert \sigma_i^c \right\rvert \le 40 \text{ (ksi)} \end{cases}, \ i = 1, 2, ..., 25$
Displacement constraints:	$\delta_j^{(x,y,z)} \le 0.3 \text{ in}, \ j = 1, 2, ..., 6$
Size variables:	$A_3 = A_1$, $A_4 = A_2$, $A_5 = A_6$, A_7, $A_8 = A_9$, A_{10}, $A_{12} = A_{11}$, $A_{14} = A_{13}$, $A_{15} = A_{16}$, $A_{18} = A_{17}$, $A_{20} = A_{19}$, $A_{22} = A_{21}$, $A_{24} = A_{23}$, $A_{26} = A_{25}$, A_{27}, A_{28}, $A_{30} = A_{29}$, $A_{31} = A_{32}$, A_{33}, $A_{35} = A_{34}$, $A_{36} = A_{37}$, A_{38}, $A_{40} = A_{39}$, $A_{41} = A_{42}$, A_{43}, $A_{45} = A_{44}$, $A_{46} = A_{47}$
Shape variables:	$x_2 = -x_1$, $x_4 = -x_3$, $y_4 = y_3$, $x_6 = -x_5$, $y_6 = y_5$, $x_8 = -x_7$, $y_8 = y_7$, $x_{10} = -x_9$, $y_{10} = y_9$, $x_{12} = -x_{11}$, $y_{12} = y_{11}$, $x_{14} = -x_{13}$, $y_{14} = y_{13}$, $x_{20} = -x_{19}$, $y_{20} = y_{19}$, $x_{21} = -x_{18}$, $y_{21} = y_{18}$

Permissible size variables:	
$A_i \in S =$	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.2, 3.4\}$ in^2
Limitation of shape variables:	
$20 \leq x_4 \leq 60$	
$40 \leq x_8 \leq 80$	
$40 \leq y_4 \leq 80$	
$100 \leq y_8 \leq 140$	
$90 \leq z_4 \leq 130$	
Loads (kips):	
Node: 1	$F_x = 1$, $F_y = -10$, $F_z = -10$
Node: 2	$F_x = 0$, $F_y = -10$, $F_z = -10$
Node: 3	$F_x = 0.5$, $F_y = 0$, $F_z = 0$
Node: 6	$F_x = 0.6$, $F_y = 0$, $F_z = 0$
Young modulus:	$E = 10^4$ (ksi)
Material density:	$\rho = 0.1$ (lb/in^3)

The optimization shall include with the displacement constraint for all free nodes (node 1 to 6) and the stress constraint. The loads are applied to nodes 1, 2, 3 and 6 as shown in Table 3.4.

CHAPTER 4
SOLUTION METHODOLOGY AND RESULTS

4.1. Size and Shape Truss Structural Optimization Algorithm

The weight minimization problems of truss structure are implemented in Python code by using the enhanced comprehensive learning particle swarm optimization algorithm (ECLPSO). The procedures of the optimization are the same for all benchmark truss structure in this study even though there are some different parameters or types of truss system. The objective function of the truss structural optimization problems is stated in Chapter 3. For every benchmark problem, there are 2 types of variables which are size variables (discrete) and shape variables (continuous). The design constraints include displacement, stress and buckling constraints as indicated for each problem.

The procedure of the optimization can be expressed step-by-step as follows.

(1) Input truss structural data including node coordinates, bars connectivity data and other parameters.

(2) Input ECLPSO algorithm parameters

(3) Start the optimization with initial positions associated with its velocities and evaluate the weight (fitness value) of truss structure by performing the structural analysis with direct stiffness method.

(4) Define personal best positions and a global best position.

(5) Begin the first iteration ($k = 1$) and follow the process of the optimization using ECLPSO in Chapter 2.

(6) Before updating the personal best position for each iteration, introduce the constraint control using the formula in Chapter 2.

(7) During the process of every iteration, record the global best position and global best fitness value (weight) to plot a convergence curve.

(8) Output the solution of optimization including the optimal cross-sectional areas and the best nodal coordinates, the convergence curve, and optimal shape of truss structure.

The flowchart of simultaneous size and shape optimization of truss structure can be summarized as shown in Figure 4.1.

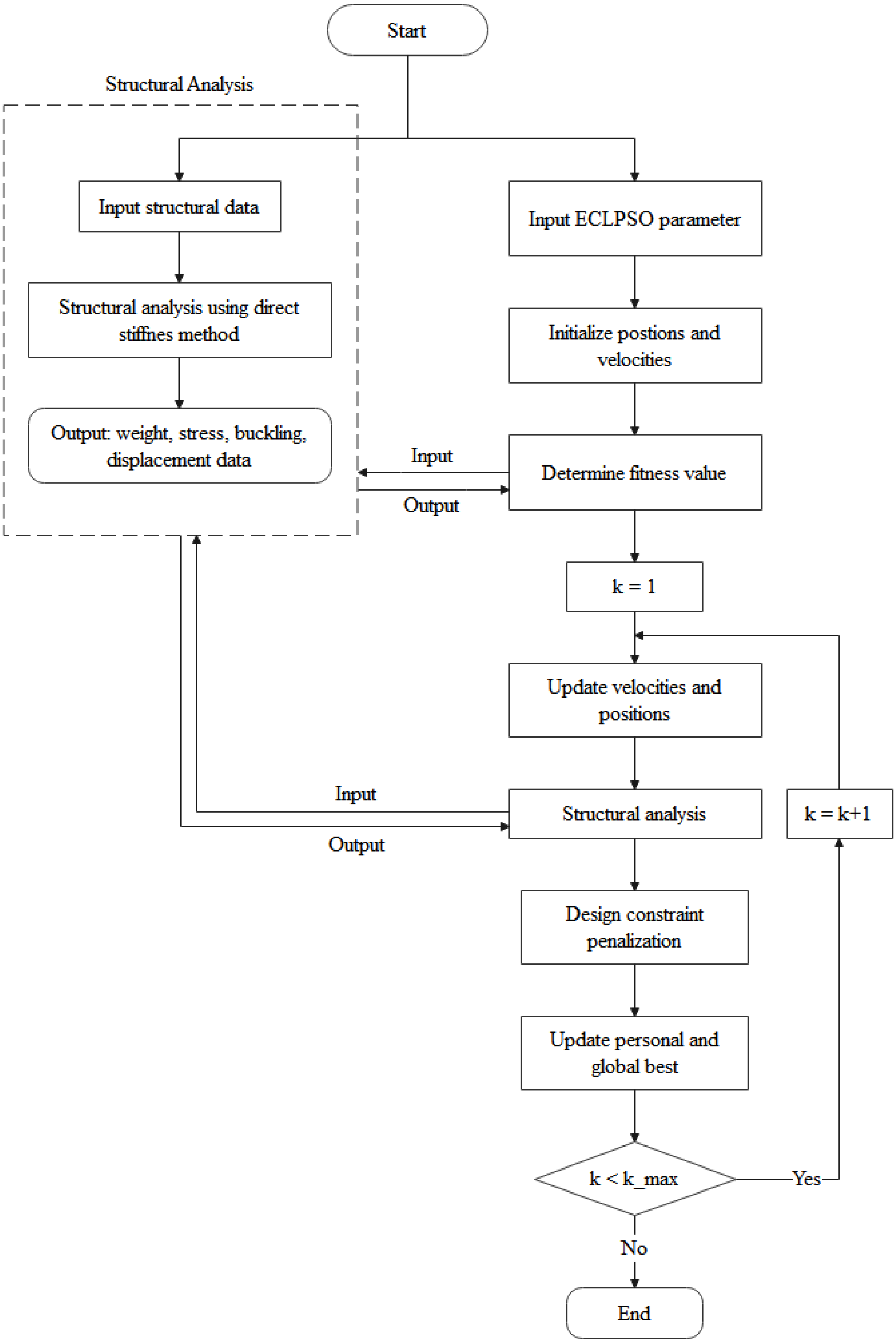

Figure 4.1. Flowchart of the optimization of the benchmark truss structure

4.2. Input Parameters for Optimization

The correct input parameters shall be carefully used in the optimization problems since they might have effect significantly on the solution. The parameters of ELCPSO and CLPSO algorithm as described in literature review are used in the optimization process of the benchmark truss structures in this study.

The total population for the optimization problems is set to 20 particles but the number of iterations varies accordingly to each problem. The dimension of objective function is the total number of variables of the problems which consists of discrete size variables and nodal coordinate variables or shape variables. The design constraints, including strength constraints and serviceability constraints, are different for each problem and handled by techniques as described in literature review.

To obtain the best results and compare fairly to those of other algorithms in the literatures, the optimization is performed 25 times independently for each problem. The best value, constraint violation ratio, number of structural analysis and standard deviation are presented in the result table for each problem.

4.3. Results for 15-bar Planar Truss Structure

The comparison of the solution from ECLPSO algorithm with those of others in literature in show in Table 4.1. There are 23 total number of design variables including 15 discrete size variables and 8 shape variables. The size variables are limit to 25 (ksi) by constraint handling technique.

The best result of ECLPSO algorithm is 74.1723 (lb) which is around 10% lighter than that of the standard PSO which is 82.2344 (lb). The standard deviation of the optimization is of 3.22 for the ECLPSO algorithm. However, the algorithm requires a greater number of structural analysis (i.e., 6000) to yield the result. The maximum stress obtained from structural analysis after the optimization is 24.9964 (ksi). The original and optimal shape of truss structure is shown in Figure 4.2. The convergence of the ECLPSO algorithm is illustrated in Figure 4.3 including the best value and mean value of the optimizations after 25 runs. It is seen that the results start to converge at 200 iterations and fully converge at 300 iterations.

Table 4.1. Optimal discrete size and shape of the 15-bar planar truss structure

Variables	(Rahami, Kaveh et al. 2008)	(Miguel, Lopez et al. 2013)	(Gholizadeh 2013)		This study
		FA	PSO	SCPSO	ECLPSO
A1	1.081	0.954	0.954	0.954	0.954
A2	0.539	0.539	1.081	0.539	0.539
A3	0.287	0.220	0.270	0.270	0.220
A4	0.954	0.954	1.081	0.954	0.954
A5	0.539	0.539	0.539	0.539	0.539
A6	0.141	0.220	0.287	0.174	0.220
A7	0.110	0.111	0.141	0.111	0.111
A8	0.110	0.111	0.111	0.111	0.111
A9	0.539	0.287	0.347	0.287	0.440
A10	0.440	0.440	0.440	0.347	0.440
A11	0.539	0.440	0.270	0.347	0.440
A12	0.270	0.220	0.111	0.220	0.270
A13	0.220	0.220	0.347	0.220	0.220
A14	0.141	0.270	0.440	0.174	0.220
A15	0.287	0.220	0.220	0.270	0.220
X2	101.5775	114.967	106.0521	137.2216	100.9857
X3	227.9112	247.040	239.0245	259.9093	242.8470
Y2	134.7986	125.919	130.3556	123.5006	134.2018
Y3	128.2206	111.067	114.273	110.0020	119.9010
Y4	54.8630	58.298	51.9866	59.9356	50.8212
Y6	-16.4484	-17.564	1.8135	-5.1799	-17.1359
Y7	-13.3007	-5.821	9.1827	4.2193	-4.1215
Y8	54.8572	31.465	46.9087	57.8829	50.7841
Weight (lb)	76.6854	75.55	82.2344	72.5143	**74.1723**
Constraint (%)	0.0	-	0.0	0.0	0.0
Max. stress	24.9992	-	24.9999	24.9912	24.9964
NSA	8000	8000	4500	4500	6000
SD	-	2.96	-	1.922	3.22

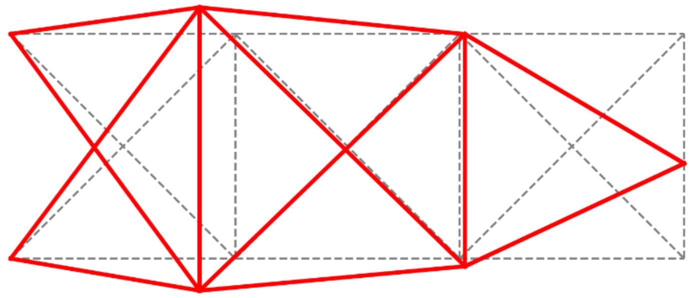

Figure 4.2. Optimal shape of 15-bar planar truss structure

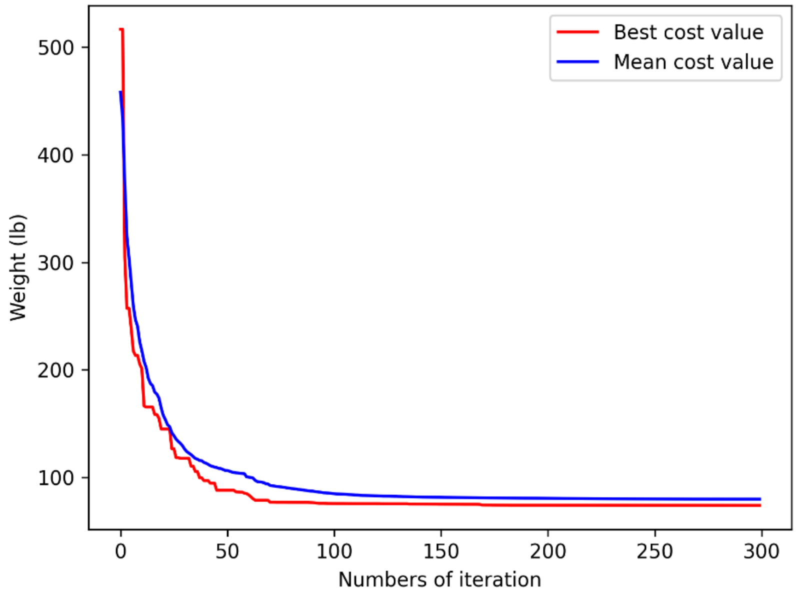

Figure 4.3. Convergence curve of 15-bar planar truss problem

4.4. Results for 18-bar Planar Truss Structure

In this problem, there are 4 discrete area variables and 8 nodal coordinate variables. The constraint restrictions consist of stress constraint for tension members (i.e., 25 ksi) and buckling constraint for compression members with buckling coefficient K = 4.

From the results in Table 4.2, we can see that there are no constraint violations including the stress constraint and buckling constraint. The ECLPSO algorithm need more structural analysis that the standard PSO but the weight from ECLPSO algorithm is the lowest comparing to other algorithms which is 4175.1425 (lb) with standard deviation of 57.32. Figure 4.4 shows the original and optimal shape of truss structure. The convergence curve is illustrated in Figure 4.5. It is seen that the algorithm starts to converge at 200 iterations then fully converges at 300 iterations.

Table 4.2. Optimal discrete size and shape of the 18-bar planar truss structure

Variables	(Rahami, Kaveh et al. 2008)	(Jawad, Ozturk et al. 2021)	(Gholizadeh 2013)		This study
		ABC	PSO	SCPSO	ECLPSO
A1	12.75	12.50	12.00	12.5	10.25
A2	18.50	17.75	18.50	17.5	17.5
A3	4.75	5.75	5.25	5.75	6.25
A5	3.25	3.75	4.50	3.75	2.75
X3	917.4475	912.9974	903.9806	907.2491	909.0566
Y3	193.7899	183.6806	185.7807	179.8671	180.1431
X5	654.3243	642.7143	644.9170	636.7873	636.8079
Y5	159.9436	143.8920	144.9692	141.8271	138.5523
X7	424.4821	411.6918	428.2196	407.9442	406.1961
Y7	108.5779	97.14763	100.5623	94.0559	94.7056
X9	208.4691	200.9087	209.5415	198.7897	197.5999
Y9	37.6349	30.21906	24.3748	29.5157	33.6241
Weight (lb)	4530.68	4537.064	4609.001	4512.365	**4175.1425**
Constraint (%)	0.0	0.0	0.0	0.0	0.0
NSA	8000	2700	4500	4500	6000
SD	-	9.7971	-	37.691	57.32

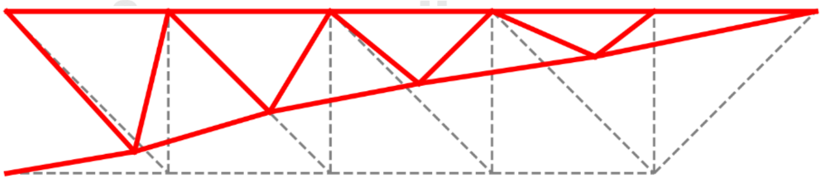

Figure 4.4. Optimal shape of 18-bar planar truss structure

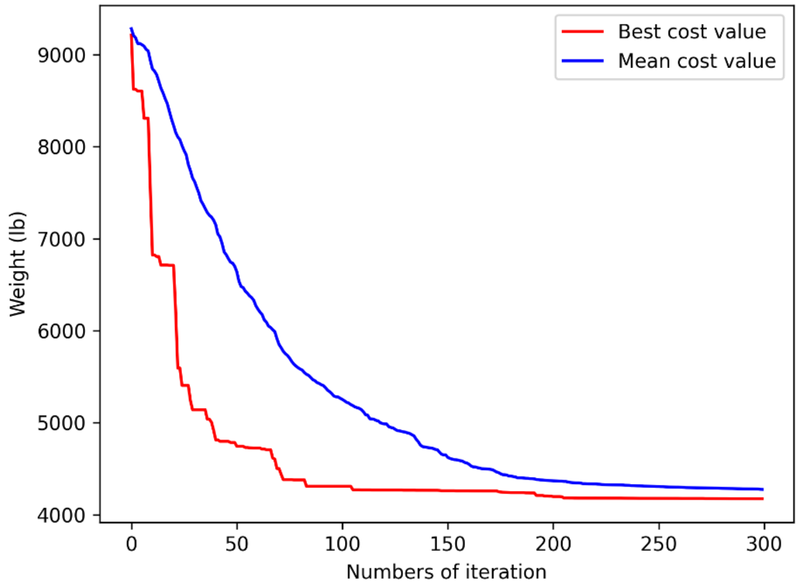

Figure 4.5. Convergence curve of 18-bar planar truss problem

4.5. Results for 47-bar Planar Truss Structure

Table 4.3. Optimal discrete size and shape of the 47-bar planar truss structure

Variables	(Hasançebi and Erbatur 2002)	(Kaveh and Zaerreza 2020)	(Gholizadeh 2013)		This study
			PSO	SCPSO	ECLPSO
A3	2.5	2.8	2.80	2.5	2.7
A4	2.5	2.5	2.70	2.5	1.9
A5	0.8	0.7	0.80	0.8	0.8
A7	0.1	0.1	1.10	0.1	0.5
A8	0.7	1.0	0.80	0.7	1.1
A10	1.3	1.1	1.30	1.4	1.7
A12	1.8	1.8	1.80	1.7	2.2
A14	0.7	0.7	0.90	0.8	0.5
A15	0.9	0.8	1.20	0.9	0.9
A18	1.2	1.5	1.40	1.3	1.9
A20	0.4	0.4	0.30	0.3	0.4
A22	1.3	1.0	1.40	0.9	0.4
A24	0.9	1.1	1.10	1.0	1.7
A26	0.9	1.0	1.20	1.1	1.5
A27	0.7	5.0	1.60	5.0	2.3
A28	0.1	0.1	1.00	0.1	0.3

A30	2.5	2.7	2.80	2.5	3.1
A31	1.0	0.9	0.80	1.0	0.5
A33	0.1	0.1	0.10	0.1	0.1
A35	2.9	3.0	3.00	2.8	3.3
A36	0.8	0.8	0.90	0.9	0.8
A38	0.1	0.1	0.10	0.1	0.1
A40	3.0	3.2	3.30	3.0	3.4
A41	1.2	1.1	0.90	1.0	0.7
A43	0.1	0.1	0.10	0.1	0.2
A45	3.2	3.3	3.30	3.2	3.7
A46	1.1	1.1	1.20	1.2	0.3
X2	104	100.5396	98.8628	101.3393	96.1045
X4	87	81.0279	78.6595	85.9111	75.1729
Y4	128	137.2003	146.7331	135.9645	132.4016
X6	70	63.8334	66.5231	74.7969	52.6328
Y6	259	254.1838	239.0901	237.7447	276.0971
X8	62	56.1445	55.6936	64.3115	45.4036
Y8	3.26	327.9040	327.7882	321.3416	348.3091
X10	53	48.2708	48.8641	53.3345	35.6093
Y10	412	407.5132	398.6775	414.3025	417.0551
X12	47	42.4458	43.1400	46.0277	30.7598
Y12	486	468.8267	464.7831	489.9216	482.1343
X14	45	45.8692	37.8993	41.8353	30.2856
Y14	504	515.2907	511.0450	522.4161	536.6923
X20	2.0	0.0010	18.2341	1.0005	0.1239
Y20	584	586.9443	594.0710	598.3905	594.3145
X21	89	80.7351	90.9369	97.8696	94.5263
Y21	637	621.5769	621.3943	624.0552	604.8316
Weight (lb)	1871.17	1869.876	1975.839	1864.10	**1799.8757**
Constraint (%)	0.0	-	0.0	0.0	0.0
NSA	-	20,020	25,000	25,000	30,000
SD	-	29.55	-	97.478	89.53

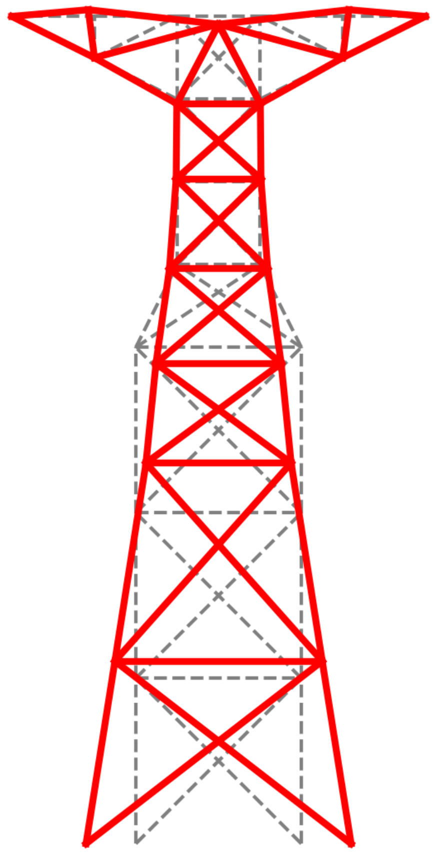

Figure 4.6. Optimal shape of 47-bar planar truss structure

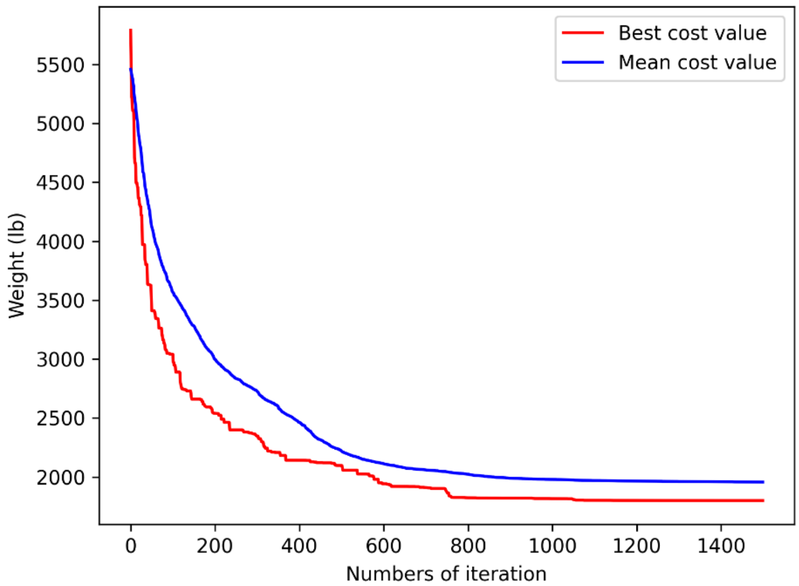

Figure 4.7. Convergence curve of 47-bar planar truss problem

The 47-bar planar truss structure problem consists of the biggest number of variables in this study which is 44 numbers of total variables. There are 17 shape variables and 27 size variable which is limited to the design constraints including stress constraint and buckling constraint. There are 20 particles used to search for global optimum in this problem.

The optimization results are shown in Table 4.3. The best value after 25 runs independently is of 1799.8757 lb with corresponding standard deviation of 89.53. The original and optimal shape and convergence curve are shown in Figure 4.6 and Figure 4.7, respectively. It is seen that the best solution of the problem starts to converge at 1200 iterations and fully converges at 1500 iterations.

4.6. Results for 25-bar Space Truss Structure

Table 4.4. Optimal discrete size and shape of the 25-bar space truss structure

Variables	(Tang, Tong et al. 2005)	(Miguel, Lopez et al. 2013)	(Gholizadeh 2013)		This study
			PSO	SCPSO	ECLPSO
A1	0.1	0.1	0.1	0.1	0.1
A2	0.1	0.1	0.1	0.1	0.1
A3	1.1	0.9	1.1	1.0	0.9
A4	0.1	0.1	0.1	0.1	0.1
A5	0.1	0.1	0.4	0.1	0.1
A6	0.2	0.1	0.1	0.1	0.1
A7	0.2	0.1	0.4	0.1	0.1
A8	0.7	1	0.7	0.9	1
X4	35.47	37.32	27.6169	36.9520	39.9664
Y4	60.37	55.74	51.6196	54.5786	557.433
Z4	129.07	126.62	129.9071	129.9758	124.1553
X8	45.06	50.14	42.5526	51.7317	58.0017
Y8	137.04	136.40	132.7241	139.5316	138.6544
Weight (lb)	124.943	118.83	129.207	117.227	**120.0191**
Constraint (%)	0.0	-	0.0	0.0	0.0
NSA	6000	6000	4500	4500	6000
SD	-	5.5	-	3.671	2.96

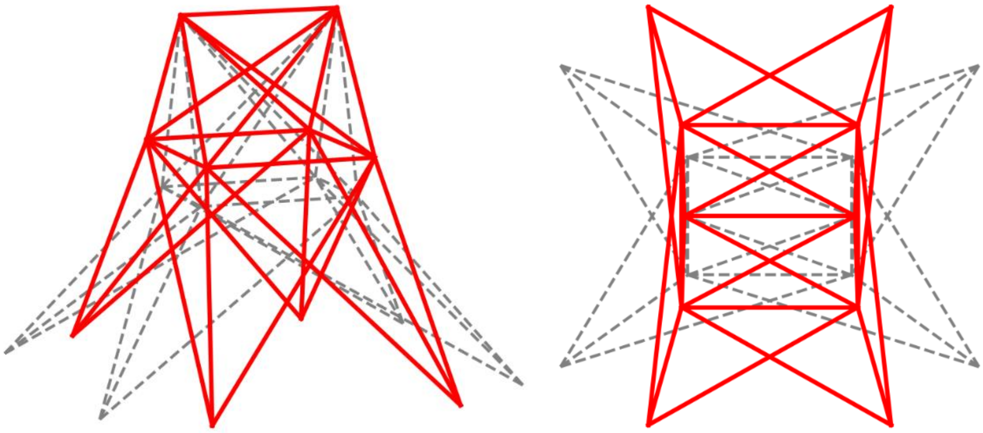

Figure 4.8. Optimal shape of 25-bar space truss structure (top view on right)

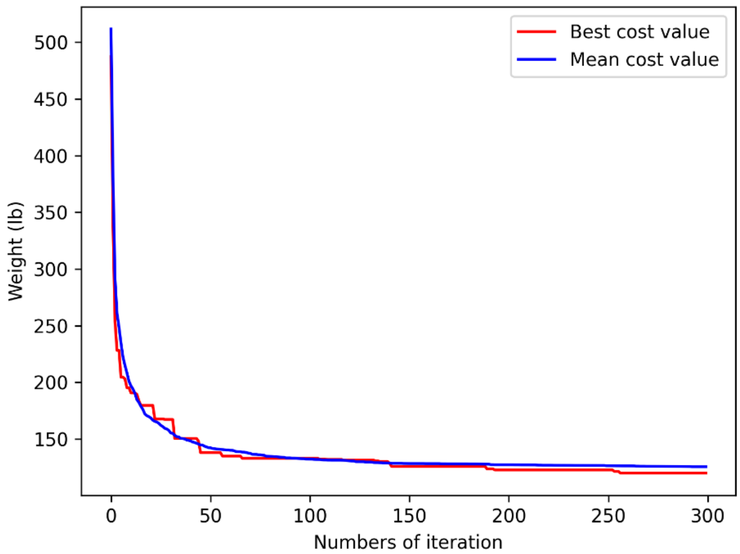

Figure 4.9. Convergence curve of 25-bar space truss problem

The problem consists of 8 discrete area variables and 5 nodal coordinates variables including all axis (i.e., X, Y and Z). The optimization is performed 25 times independently with 20 particles for each run. There are 2 design constraints namely stress constraint and displacement constraint.

The optimization produces an acceptable result where the weight is of 120.0191 lb with standard deviation of 2.96. Figure 4.8 shows 3D view and top view of the original and optimal shape of the space truss structure. The convergence characteristic is shown in Figure 4.9. It is seen that the optimization starts to yield the solution at 280 iterations and completely converge at 300 iterations.

CHAPTER 5
CONCLUSION

This study presents research purposes and solution methodology of the simultaneous size and shape optimization using enhanced comprehensive learning particle swarm optimization algorithm which is a variant of the standard PSO. The objective of this study is to apply the ECLPSO algorithm to the benchmark truss structures under the design constraints including strength constraints and serviceability constraints, and to compare the outcome with those results reported in the literatures. The benchmark problems include planar truss structures and a space truss structure with correspondingly properties. The solutions of the benchmark truss structure problem using ECLPSO algorithm are presented with fast convergence and comparable results. After the optimizations, the ECLPSO is observed to yield better results than those of the standard PSO. This improvement over PSO is the resultant of the introduction of learning probabilities concept of CLPSO algorithm and the two enhancements of ECLPSO algorithm, namely perturbation-based exploitation, and adaptive learning probabilities. This proves that ECLPSO algorithm can be used in many structural applications and is a robust, effective, and reliable optimization method for solving structural optimization problem.

REFERENCES

Dutu, L.-C., G. Mauris and P. Bolon (2015). "A Linear-Complexity Rule Base Generation Method for Fuzzy Systems."

Farshchin, M., C. V. Camp and M. Maniat (2016). "Optimal design of truss structures for size and shape with frequency constraints using a collaborative optimization strategy." Expert Systems with Applications **66**: 203-218.

Gandomi, A. H., X.-S. Yang, S. Talatahari and A. H. Alavi (2013). "Metaheuristic algorithms in modeling and optimization." Metaheuristic applications in structures and infrastructures: 1-24.

Gholizadeh, S. (2013). "Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization." Computers & Structures **125**: 86-99.

Hasançebi, O. and F. Erbatur (2002). "On efficient use of simulated annealing in complex structural optimization problems." Acta Mechanica **157**(1): 27-50.

Ho-Huu, V., T. Nguyen-Thoi, M. H. Nguyen-Thoi and L. Le-Anh (2015). "An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures." Expert Systems with Applications **42**(20): 7057-7069.

Jawad, F. K. J., M. Mahmood, D. Wang, O. Al-Azzawi and A. Al-Jamely (2021). "Heuristic dragonfly algorithm for optimal design of truss structures with discrete variables." Structures **29**: 843-862.

Jawad, F. K. J., C. Ozturk, W. Dansheng, M. Mahmood, O. Al-Azzawi and A. Al-Jemely (2021). "Sizing and layout optimization of truss structures with artificial bee colony algorithm." Structures **30**: 546-559.

Kassimali, A. (2011). Matrix Analysis of Structures, Cengage Learning.

Kaveh, A. and S. Talatahari (2009). "A particle swarm ant colony optimization for truss structures with discrete variables." Journal of Constructional Steel Research **65**(8): 1558-1568.

Kaveh, A. and A. Zaerreza (2020). "Size/Layout Optimization of Truss Structures Using Shuffled Shepherd Optimization Method." Periodica Polytechnica Civil Engineering **64**(2): 408-421.

Kennedy, J. and R. Eberhart (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks.

León-Aldaco, S. E. D., H. Calleja and J. A. Alquicira (2015). "Metaheuristic Optimization Methods Applied to Power Converters: A Review." IEEE Transactions on Power Electronics **30**(12): 6791-6803.

Liang, J. J., A. K. Qin, P. N. Suganthan and S. Baskar (2006). "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions." IEEE Transactions on Evolutionary Computation **10**(3): 281-295.

Ma, Y.-Z., H.-S. Li, K.-F. Tee and W.-X. Yao (2018). "Combined size and shape optimization of truss structures using subset simulation optimization." Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering **233**(7): 2455-2477.

Miguel, L. F. F., R. H. Lopez and L. F. F. Miguel (2013). "Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm." Advances in Engineering Software **56**: 23-37.

Omidi, J. and K. Mazaheri (2020). "Differential evolution algorithm for performance optimization of the micro plasma actuator as a microelectromechanical system." Scientific Reports **10**.

Perez, R. E. and K. Behdinan (2007). "Particle swarm approach for structural design optimization." Computers & Structures **85**(19): 1579-1588.

Plevris, V. and M. Papadrakakis (2011). "A Hybrid Particle Swarm—Gradient Algorithm for Global Structural Optimization." Computer-Aided Civil and Infrastructure Engineering **26**(1): 48-68.

Rahami, H., A. Kaveh and Y. Gholipour (2008). "Sizing, geometry and topology optimization of trusses via force method and genetic algorithm." Engineering Structures **30**(9): 2360-2369.

Rajeev, S. and C. S. Krishnamoorthy (1992). "Discrete Optimization of Structures Using Genetic Algorithms." Journal of Structural Engineering **118**(5): 1233-1250.

Rezaee Jordehi, A. and J. Jasni (2015). "Particle swarm optimisation for discrete optimisation problems: a review." Artificial Intelligence Review **43**(2): 243-258.

Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), IEEE.

Shi, Y. and R. Eberhart (1998). A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360).

Shi, Y. and R. C. Eberhart (1999). Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406).

Tang, W., L. Tong and Y. Gu (2005). "Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables." International Journal for Numerical Methods in Engineering **62**(13): 1737-1762.

Venter, G. and J. Sobieszczanski-Sobieski (2004). "Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization." Structural and Multidisciplinary Optimization **26**(1): 121-131.

Xin, Y., L. Yong and L. Guangming (1999). "Evolutionary programming made faster." IEEE Transactions on Evolutionary Computation **3**(2): 82-102.

Yu, X. and X. Zhang (2014). "Enhanced comprehensive learning particle swarm optimization." Applied Mathematics and Computation **242**: 265-276.

Zhan, Z., J. Zhang, Y. Li and Y. Shi (2011). "Orthogonal Learning Particle Swarm Optimization." IEEE Transactions on Evolutionary Computation **15**(6): 832-847.

VITA

NAME	SOVIPHOU MUONG
DATE OF BIRTH	19 August 1994
PLACE OF BIRTH	Kompong Thom, Cambodia
INSTITUTIONS ATTENDED	1. Institute of Technology of Cambodia
	2. Chulalongkorn University
HOME ADDRESS	Phnom Penh, Cambodia