OPTIMUM DESIGN OF STEEL STRUCTURES VIA ARTIFICIAL BEE COLONY (ABC) ALGORITHM AND SAP2000

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CENGİZ ESER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

FEBRUARY 2014

Approval of the thesis:

OPTIMUM DESIGN OF STEEL STRUCTURES VIA ARTIFICIAL BEE COLONY (ABC) ALGORITHM AND SAP2000

submitted by **CENGİZ ESER** in partial fulfillment of the requirements for the degree of **Master of Science in Civil Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ahmet Cevdet Yalçıner ____________________
Head of Department, **Civil Engineering**

Assoc. Prof. Dr. Oğuzhan Hasançebi ____________________
Supervisor, **Civil Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Cem Topkaya ____________________
Civil Engineering Dept., METU

Assoc. Prof. Dr. Oğuzhan Hasançebi ____________________
Civil Engineering Dept., METU

Assoc. Prof. Dr. Ayşegül Askan Gündoğan ____________________
Civil Engineering Dept., METU

Dr. Onur Pekcan ____________________
Civil Engineering Dept., METU

Dr. Cenk Tort ____________________
Civil Engineer, Miteng Engineering

Date: 07.02.2014

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Cengiz ESER

Signature :

ABSTRACT

OPTIMUM DESIGN OF STEEL STRUCTURES VIA ARTIFICIAL BEE COLONY (ABC) ALGORITHM AND SAP2000

Eser, Cengiz

M.Sc., Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. Oğuzhan Hasançebi

February 2014, 81 pages

Over the past few years, metaheuristic optimization techniques have received considerable attention from engineering researchers. Under metaheuristics, swarm intelligence based algorithms have been used in the solution of various structural optimization problems where the main goal is to minimize the weight of structures while satisfying all design constraints imposed by design codes. In this study, artificial bee colony algorithm (ABC) is utilized to optimize four truss structures from real life and literature. ABC algorithm is one of those popular techniques which has proved to be effective when solving combinatorial and nonlinear optimization problems such as scheduling, routing, financial product design and other problem areas. In this thesis, the results of the ABC algorithm are compared with the results of other optimization algorithms from the literature to investigate the use and efficiency of this technique for solving steel truss design problems. Artificial bee colony algorithm is computerized in VB.NET platform to develop software called ABC-SOP2014. ABC-SOP2014 is capable to interact with well-known structural analysis and design software SAP2000 through the Open Application Programming Interface (OAPI) for size optimum design of steel structures. In this study the program is used only for discrete size optimization of steel truss structures with penalty function implementation aiming minimum weight according to design limitations imposed by AISC-ASD (Allowable Stress Design Code of American Institute of Steel Construction) or limitations specified for the problem without any code requirement. The results reveal that the ABC algorithm can be used effectively as an optimization technique for truss structures, resulting significant savings.

Key Words: Artificial Bee Colony, Structural Optimization, Size Optimization, Discrete Optimization, Steel Truss Structures

ÖZ

ÇELİK YAPILARIN YAPAY ARI KOLONİSİ (ABC) ALGORİTMASI VE SAP2000 İLE OPTİMUM TASARIMI

Eser, Cengiz
Yüksek Lisans, İnşaat Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Oğuzhan Hasançebi

Şubat 2014, 81 sayfa

Son birkaç yılda, metasezgisel optimizasyon teknikleri mühendislik araştırmacılarının önemli ölçüde dikkatini çekmiştir. Metasezgisel teknikler altında, sürü zekası tabanlı algoritmalar, temel amacı tasarım kodlarıyla dayatılan tüm tasarım kısıtlamalarını sağlarken yapıların ağırlığını en aza indirmek olan çeşitli yapısal optimizasyon problemlerinin çözümünde kullanılmıştır. Bu çalışmada, yapay arı koloni algoritması (ABC) , gerçek hayat ve literatürden alınan dört kafes sistem yapısını optimize etmek için kullanılmaktadır. ABC algoritması, zamanlama, rotalama, finansal ürün tasarımı ve diğer problem alanları gibi kombinasyonel ve doğrusal olmayan optimizasyon problemlerini çözmek için etkili olduğu kanıtlanmış bu popüler tekniklerden biridir.

Bu tezde, ABC algoritmasının sonuçları, çelik kafes tasarım problemlerini çözmek için bu tekniğin kullanımı ve etkinliğini araştırmak amacıyla, literatürdeki diğer optimizasyon algoritmalarının sonuçları ile karşılaştırılmıştır. Yapay arı koloni algoritması ABC-SOP2014 olarak adlandırılan bir yazılımı geliştirmek üzere

VB.NET platformunda programlanmıştır. ABC-SOP2014 çelik yapıların boyutu optimum tasarımı için tanınmış yapısal analiz ve tasarım yazılımı SAP2000 ile Açık Uygulama Programlama Arayüzü (OAPI) aracılığıyla etkileşim yeteneğine sahiptir. Bu çalışmada program AISC - ASD (Amerikan Çelik Konstrüksiyon Enstitüsünün Emniyet Gerilmesi Tasarım Kuralları)'nın dayattığı tasarım kısıtlamalarına veya herhangi bir kod gereksinimi olmadan problemin kendisince belirtilen sınırlamalara göre, ceza fonksiyonu uygulanması ile asgari ağırlığı amaçlayarak sadece çelik kafes yapıların ayrık boyut optimizasyonu için kullanılmıştır. Sonuçlar, ABC algoritmasının çelik kafes yapılar için önemi tasarruflarla birlikte bir optimizasyon yöntemi olarak etkin bir şekilde kullanılabileceğini ortaya koymaktadır.

Anahtar Kelimeler: Yapay Arı Kolonisi, Yapısal Optimizasyon, Boyut Optimizasyonu, Ayrık Optimizasyon, Çelik Kafes Yapılar

To My Family

ACKNOWLEDGMENTS

I would like to offer my deepest gratitude to my supervisor Assoc. Prof. Dr. Oğuzhan Hasançebi, for his invaluable guidance, support and encouraging approach that kept my motivation alive throughout the research. It was a great opportunity to be accompanied by his enlightening guidance during the progress of this thesis.

I wish also to express my sincere gratitude to the examining committee for their contributive suggestions and efforts in reviewing the thesis.

I am grateful to my family for their guidance, their encouragement and their endless support throughout my life.

I wholeheartedly thank my friends for their continuous motivation and inspiration.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF FIGURES xiv

LIST OF TABLES xvi

LIST OF SYMBOLS xvii

CHAPTERS

1. INTRODUCTION 1
   - 1.1 Truss Structures 1
   - 1.2 Artificial Bee Colony (ABC) Algorithm 2
   - 1.3 Software Development 3
   - 1.4 Outline of the Thesis 3

2. STRUCTURAL OPTIMIZATION 5
   - 2.1 Introduction 5
   - 2.2 The Design Process 6
   - 2.3 Elements of Optimization 7
     - 2.3.1 Design Variables 7
     - 2.3.2 Objective Function 8
     - 2.3.3 Constraints 8
     - 2.3.4 Design Space 9
   - 2.4 Mathematical Formulation 9
   - 2.5 Classification of the Structural Optimization Tasks 10
     - 2.5.1 Size Optimization 11

2.5.2 Shape Optimization 12
2.5.3 Topology Optimization 13
2.5.4 Selection of Material Properties 15
2.5.5 Selection of Construction 15
2.6 Optimization Methods 16
3. PROBLEM STATEMENT 19
3.1 Introduction 19
3.2 Design Variables 19
3.3 Objective Function 20
3.4 Constraints 21
3.5 Constraint Handling Procedure 23
4. ARTIFICIAL BEE COLONY (ABC) ALGORITHM AND SOFTWARE DEVELOPMENT FOR STRUCTURAL OPTIMIZATION 25
4.1 Introduction 25
4.2 Literature Survey 26
4.2.1 Studies Related to Structural Optimization 26
4.2.2 Studies Related to Other Applications of ABC in Civil Engineering ... 29
4.2.3 Studies Related to Other Areas of Engineering Optimization 32
4.3 Swarm Intelligence 32
4.3.1 Behavior of Honey Bee Swarm 33
4.4 Artificial Bee Colony (ABC) Algorithm 35
4.5 ABC Algorithm for Discrete Optimization 38
4.5.1 Diversification Generation Method for Initial Population 40
4.6 Constraint Handling 41
4.7 Sample Problem (Two-bar Truss) 42
4.7.1 General Statement of the Design Variables 42
4.7.2 Derivation and Formulation of the Problem 43
4.7.3 Solution of the Problem 45
4.7.3.1 Solution of the Problem with ABC Algorithm 45
4.7.3.2 Solution of the Problem with Augmented Lagrangian Method (ALM) 50

4.7.3.3 Solution of the Problem with Zoutendijk's Method of Feasible Directions 51
4.7.3.4 Solution of the Problem with the Generalized Reduced Gradient Method (GRG) (Nonlinear Constraints) 51
4.7.3.5 Solution of the Problem with the Sequential Quadratic Programming (SQP) Method 52
4.7.4 Discussion of the Results 53
4.8 Structural Optimization Software Development with ABC Algorithm 53
5. APPLICATION OF ABC ALGORITHM VIA ABC-SOP2014 SOFTWARE 57
5.1 Introduction 57
5.2 Truss Problems 57
5.2.1 22-Bar Cantilever Truss 57
5.2.2 25-Bar Space Truss 62
5.2.3 160-Bar Space Truss Pyramid 65
5.2.4 693-BarBraced Barrel Vault 68
6. CONCLUSIONS 73
6.1 Conclusion 73
6.2 Final Recommendations 74
6.3 Future Work 75
REFERENCES 77

LIST OF FIGURES

FIGURES

Figure 2-1 Three-bar truss problem and design space (Adapted from Schmit, 1981) 10

Figure 2-2 Classification of structural optimization tasks according to design variables (Schumacher, 2013) ... 11

Figure 2-3 Size optimization of a steel tower structure ... 12

Figure 2-4 Shape optimization defined by Galileo in 1638 (Crew and Salvio, 2010) ... 13

Figure 2-5 The flow of computations for topology design (Bendsoe and Sigmund, 2003) ... 14

Figure 2-6 Types of topology optimization, (Maute, 1998) ... 14

Figure 2-7 Mimicking of actual industrial design process. Rib structure in front part of airplane wing at EADS (courtesy of EADS Military Aircraft) ... 15

Figure 2-8 Classification of numerical optimization techniques (Gandomi et al, 2013) ... 17

Figure 4-1 Basic elements of foraging behavior ... 34

Figure 4-2 Encoding information between honeybees (Seeley, 2010) ... 35

Figure 4-3 Detailed Pseudo code of the ABC Algorithm ... 36

Figure 4-4 Flowchart of ABC algorithm ... 37

Figure 4-5 The two bar truss problem ... 42

Figure 4-6 ABC Two Bar Optimization Results ... 50

Figure 4-7 The algorithm of ABC-SOP2014 ... 55

Figure 5-1 Statically determinate 22-bar plane truss ... 58

Figure 5-2 Axial forces on elements of 22-bar planar cantilever truss 59

Figure 5-3 Element stresses within limitations of 22-bar planar cantilever truss 59

Figure 5-4 ABC-SOP2014 optimization results of 22-bar planar cantilever truss 60

Figure 5-5 ABC-SOP2014 design history of 22-bar planar cantilever truss 61

Figure 5-6 25-bar space truss 62

Figure 5-7 ABC-SOP2014 design history of 25-bar space truss 64

Figure 5-8 160-bar pyramid 66

Figure 5-9 ABC-SOP2014 optimization results of 160-bar truss pyramid................ 67

Figure 5-10 ABC-SOP2014 design history of 160-bar space pyramid 68

Figure 5-11 The platform shelter at Thirumailai Station, LUZ, Chennai, India........ 69

Figure 5-12 The cross-section of the parallel vault and railway tracks 70

Figure 5-13 The 693-bar braced vault 3-D view, Front view and Plan view (Hasançebi, 2011) 71

Figure 5-14 ABC-SOP2014 design history of 693 bar braced vault 72

LIST OF TABLES

TABLES

4-1. The pre-assigned analysis variables 43

4-2. The bounds of design variables 43

4-3. The initial artificial bee colony 45

4-4. The employed bees for the two-bar truss 46

4-5. The objective function values for the employed bees and onlooker bees for the-two-bar truss 48

4-6. The new objective function values for the employed bees and onlooker bees for the-two-bar truss at the second cycle 49

4-7. The final solution at the 50. cycle for the-two-bar truss 49

4-8. Results of ALM solution 50

4-9. Results of Zoutendijk's Method of Feasible Directions solution 51

4-10. Results of GRG solution 52

4-11. Results of SQP solution 52

5-1. Comparison of the optimum designs for 22-bar planar truss. 61

5-2. Nodal loading conditions 8kips) for the 25-bar space truss. 63

5-3. Comparison of the optimum designs for 25-bar space truss. 64

5-4. Comparison of optimization results for 160-bar space pyramid 67

5-5. Comparison of ABC with other optimization techniques for 693-bar braced barrel vault 72

LIST OF SYMBOLS

x	: design variable vector
f(x)	: objective function
g(x)	: equality constraints
h(x)	: inequality constraints
$x^{(l)}$	: the lower bound of the side constraints
$x^{(u)}$	: the upper bound of the side constraints
I	: design space for each member group
W	: weight
A	: cross-sectional area
ρ_m	: unit weight
L_m	: length of the *m*th member
N_d	: number of structural members
N_g	: number of member groups
N_k	: number of members in member group k
g	: constraints on stresses
s	: constraints on slenderness ratios
δ	: constraints on displacements
σ	: computed axial stresses
$(\sigma)_{all}$	: allowable axial stresses
λ	: slenderness ratio
$(\lambda)_{all}$	: allowable slenderness ratio
N_j	: total number of joints
d	: computed displacement, mean diameter
$(d)_{all}$	: allowable displacement
$(\sigma_t)_{all}$	: allowable tensile stress
E	: modulus of elasticity

C_c : critical slenderness ratio

K_m : effective length factor of *m*th member

L_m : length of *m*th member

r_m : minimum radii of gyration

f_a : calculated axial stress

F_y : yield strength of the material

F_u : ultimate tensile strength of the material

Φ : fitness score

α : penalty function coefficient

sn : total number of bees in the colony

D : number of design variables

$\vec{x}_i$: initial food source

$\vec{v}_i$: new food source in the neighborhood of the food source $\vec{x}_i$

$P(\vec{x}_k)$: the selection probability

${}_i x$: the i^{th} food source in the population

I_j : the index of area A_j in the available profile list

$I_j^{\min}$: the first profile index in section pool

$I_j^{\max}$: the last profile index in section pool

H : the height

t : the thickness

DL : dead load

WL : wind load

CHAPTER 1

INTRODUCTION

The concept of optimization is a basic part of our daily lives. To increase the company profit implies an objective of economy or to produce the best quality of life with the resources available is an objective of engineering. The tool to be used to achieve the best in a timely and economical way is optimization.

There have been developed numerous optimization techniques for optimum design of structural systems. With the availability of computer codes, new and more sophisticated optimization techniques have been emerged against the conventional methods like optimality criteria, dynamic programming and steepest descent. Structural optimization with meta-heuristic search methods have become more popular as a consequence of acquiring extensive accomplishment in dealing with a variety of practical and complex optimization tasks, where it is nearly impossible to come up with the optimum solution by traditional deterministic design procedures.

These new meta-heuristic optimization techniques developed during last three decades enable to engineers and designers find the most proper and efficient solution amongst thousands of design alternatives.

1.1 Truss Structures

Truss structures can be used in buildings as support to roofs and floors. Moreover, they can be also used for rail and road bridges or for cranes. A truss structure consists of triangular units made up of connections of straight and slender bars. Because a truss is not able to transfer moments, bars are subjected to only axial compressive or tensile forces. Cross-sectional area is a basic property to characterize a truss element to resist these axial forces apart from material properties like modulus of elasticity. The length of a truss member can be determined by the end node coordinates.

A planar truss element has two local and four global dof, a space truss element has also two local dof, whereas it has six global dof.

Trusses offer efficient solutions where material use is considered. However, fabrication and maintenance costs must be taken into account. Therefore, a simple design with maximum repetition is preferred.

1.2 Artificial Bee Colony (ABC) Algorithm

Computational researchers have been greatly interested in the natural sciences to model and solve complex optimization problems by employing nature- and bio-inspired algorithms. This is mainly due to complexity and/or non-linearity of the problems. Classical algorithms generally require making several assumptions. Researchers fascinated by the swarm behavior in nature such ant colonies, honey bees, bird flocking, animal herding, and many more, have developed population based algorithms such as Ant Colony Optimization, Bee Colony Optimization, Particle Swarm Optimization, Fish Schooling, etc. These algorithms have been successfully applied to solve computational, complex and non-linear problems from different disciplines.

Swarm Intelligence (Beni and Wang, 1989) is the area of Artificial Intelligence that is based on study of actions in various decentralized systems. Swarm intelligence (Bonabeau et al. 1999) is the part of artificial intelligence on the basis of studying actions of individuals in various decentralized systems. These decentralized systems (multi agent systems) are composed of physical individuals (robots, for example) or “virtual” (artificial) ones that communicate among themselves, cooperate, collaborate, exchange information and knowledge and perform some tasks in their environment.

Few algorithms from the swarm intelligence class, inspired by bees’ behavior, appeared during the last decade. An excellent survey of the algorithms inspired by bees’ behavior in the nature is given in (Baykasoglu et al. 2007).

The artificial bee colony optimization algorithm belongs to the class of stochastic swarm optimization methods. The proposed algorithm is inspired by the foraging habits of bees in nature. The communication systems between individual insects contribute to the configuration of the collective intelligence of the social insect colonies.

Artificial bee colony (ABC) algorithm has been widely used for all types of optimization problems in various civil engineering disciplines and other disciplines, since it has been introduced originally by Karaboga (2005) for solving numerical optimization problems based on simulating real bees social behavior, foraging behavior as a heuristic. In this study it is aimed to implement ABC algorithm to discrete size optimization of real size steel truss structures, which leads to minimum weight design. Details of the ABC algorithm are discussed in chapter 4.

1.3 Software Development

A computer program called ABC-SOP2014 is developed specially for this study as a size optimization tool that is capable of finding the appropriate combination of ready sections with optimum cross-sectional areas for the minimum weight design of steel truss structures using artificial bee colony algorithm. The software developed by using VB.NET programming language interacts with SAP2000 v14 through Open Application Programming Interface (OAPI), which is released by Computers and Structures, Inc. Artificial Bee Colony Algorithm is embedded in ABC-SOP2014 program to implement the optimization procedure. ABC-SOP2014 is a user-friendly, easy to use program, which enables users to perform structural optimization under various constraints such as stress, stability and displacement imposed by problems or by specific design codes. The optimization problem in this study is called as discrete structural optimization, since the cross-sections of steel members can be selected only from a prescribed discrete set of values.

1.4 Outline of the Thesis

Chapter 2 deals with the basic concepts of optimum structural design. After classification of the design process, elements and mathematical formulation of structural optimization are described. Types of the optimization tasks and classification of numerical optimization techniques are outlined. In Chapter 3 is mathematical statement of the structural optimization problem for the structural model is defined, in other words the objective function and the constraints are described in details. In chapter 4, the literature survey is done firstly, and then swarm intelligence is introduced. Consequently, the main principles of artificial bee colony (ABC) algorithm is presented that is used in this study as optimization method. Next constraint handling method is outlined. Consequently, a sample problem is solved by ABC algorithm and by four classical methods comparatively. Thereafter, the optimization program ABC-SOP2014 written in VB.NET programming language and developed to find the optimum weight for truss structures by means of ABC algorithm is introduced. The main features, capabilities and algorithm of the software are also expressed. In chapter 5, four numerical test examples from literature and the results obtained by ABC-SOP2014 using ABC algorithm are studied and discussed in details. Chapter 6 presents the conclusion, recommendations based on the results of the study and issues of future work.

CHAPTER 2

STRUCTURAL OPTIMIZATION

2.1 Introduction

The optimization concept became popular with significant progress in capabilities of computers as well as structural analysis and optimization techniques in recent decades. Minimum weight optimum design of basic aircraft structural components such as columns and stiffened panels, subject to compressive loads was initially developed during World War II (Kirsch, 1993). After Schmit offered in 1960 a comprehensive statement of the use of mathematical programming techniques to solve the non-linear-inequality-constrained problem of designing elastic structures, his work indicated the feasibility of coupling finite element analysis and nonlinear mathematical programming to create automated optimum design capabilities for structural systems. Today most engineers who design structures employ complex general-purpose structural analysis software and the major challenge for researchers in structural optimization is to develop user-friendly methods that are suitable for use with such software packages. Another major challenge is to reduce the high computational cost of complex real-life problems.

Haftka & Gürdal (1992) paraphrases Douglas Wilde’s optimal design definition as “being the best feasible design according to a preselected quantitative measure of effectiveness”. Recently, Christensen & Klarbring (2008) defined structural optimization as “the subject of making an assemblage of materials sustain loads in the best way.” Both of the definitions address the term “best”, therefore an objective should be defined to specify the best. To design a structure with best performance, we can make the structure as stiff as possible or as insensitive to buckling or instability as possible, or to obtain the lightest structure, we could minimize the weight. Structural optimization problem can be formulated by picking one of the preselected quantitative measures like weight, stiffness, critical load, stress, displacement and geometry as an objective function that should be minimized or maximized using some other measures as constraints.

Functionality, economy and esthetics can also be considered as the objective in the design process.

This study addresses the solution of constrained optimization problems of steel truss structures with stress, stability, displacement and some other constraints by using an effective optimization algorithm called artificial bee colony (ABC) algorithm, by determining the cross-sectional areas of the structural members for minimizing the weight of a given structure.

In this chapter the design process and the elements of the optimization in the structural design process are introduced, to provide a general understanding on the subject. Mathematical formulation of nonlinear constrained optimization problem is also given. Then, the classification of structural optimization tasks are defined.

2.2 The Design Process

The design process may be divided into four stages as follows (Kirsch, 1981):

1. ***Functionality*****:** The required lanes on a bridge, the required space in an industrial building, loads expected to be carried on a truss bridge etc. are examples of functional requirements, which are often established before entering the design process.
2. ***Conceptual design*****:** It is the critical part of the design stage, because the designer should select the overall topology, type of structure, and materials by his ingenuity, creativity, and engineering judgment to serve the structural systems functional purposes. For a bridge deciding whether it should be a truss bridge, an arch bridge or perhaps a cable-stayed bridge with selected materials is an example to conceptual design.
3. ***Optimization*****:** Within the selected concept considering desired constraints, satisfying the functional requirements achieving the optimal design. For a bridge it would be selection of the best geometry of a truss or the cross-sections of the members or minimizing the cost by using

least possible amount of material. Utilizing computer with optimization algorithms and software is most suitable to this step.

4. ***Detailing*****:** After completion of optimization stage, results must be checked and modified if necessary. Engineering judgment, experience and decision-making process is necessary at this stage. This stage is usually controlled by market, social and esthetic factors.

Iterative procedures for the four stages are often required to find an acceptable final design. At the end, even the conceptual requirements are fulfilled, the final design may not be optimal. At that point, optimization techniques and computer aided design utilizing finite element method based software become the helpful and effective tools to make the best possible decision.

2.3 Elements of Optimization

2.3.1 Design Variables

Design variables are the parameters used in the formulation of the objective function to define the structural system. They can be size design variable related with cross-sectional quantities like area of a truss member, the moment inertia of a flexural member, area of a beam, and thickness of a plate or a shell. The coordinates of joints, the location of supports, and the span lengths are examples for configurational or geometric layout variables. Some uneconomical members are eliminated during the optimization process. Therefore, some design variables are defined as integer variables to declare the existence or absence of a structural element. For example a truss member joining two nodes which is limited to the values 1 and 0 is an integer topological variable. 1 represents the existence of member and 0 represents the absence of the member. Number of elements in a grillage system, number of spans in a bridge or number of columns supporting a slab are some other examples of topological variables.

Besides integer variables, design variables can be also continuous or discrete. Continuous variables are selected between lower and upper bounds of the variable, whereas discrete variables are selected from a prescribed set of values. The selection of the design variables must be consistent with the structural model and optimization algorithm for the success of the optimization process.

2.3.2 Objective Function

The objective function is a criterion to determine the quality of the solution and the effectiveness of the design. For that reason, a great deal of care, judgment, and experience are required for determining the objective function. The common engineering objectives involve minimization of overall cost, or minimization of total weight, or maximization of mechanical quality, or maximization of net profit, or others. In some cases there could be more than one objective that the designer may want to optimize simultaneously, called multiobjective or multicriterion optimization. However, multiobjective optimization algorithms are more complex and computationally expensive. Therefore, in most cases single criterion optimization is preferred and other objectives are included as constraints.

The most common objective in structural optimization applications is the weight minimization of the structure due to fact that is readily quantified, but the minimum weight concept is not always the cheapest. When we consider the interaction of design and technology, we should not forget that cost is practically important than the weight, but obtaining the objective function for the cost of the construction is more complicated, since it includes parameters such as cost of materials, fabrication, transportation, operating and maintenance cost. These factors have direct effect on the sizes, shape or topology of the structure. Furthermore, displacements, average stiffness of the structure, maximum stress and strain, buckling load, collapse load, vibration frequencies or any combination of these can be used as objective function.

2.3.3 Constraints

The special conditions that must be satisfied in order to produce a feasible design are called constraints. The set of solutions that satisfy all constraints is called the feasible design. Constraints may be categorized in two groups in structural optimization problems: side (design) constraints and behavior constraints. Side constraints arise from various considerations such as functionality, fabrication, or aesthetics. These constraints are generally related to the lower and upper bounds of design variables. Examples of side constraints include minimum value of a cross-sectional dimension, minimum thickness of a plate, minimum slope of a roof structure, maximum height of a truss.

Contrary to side constraints, the behavioral constraints derive from mechanical response of the structural system under application of loading and impose restrictions on the behavior or performance of the system according to the provisions of the design codes such as displacement, stress, strength, cracking. Both side and behavior constraints may be formulated in the form of inequalities and equalities.

A problem stated with some constraints is called constrained optimization problem, whereas, problems do not include any limitations are called unconstrained optimization problems. In some cases, constrained optimization problems are converted to unconstrained optimization problems by means of penalty functions or other constraint handling methods.

2.3.4 Design Space

Design space is a region or domain that is described by design variables in the objective function. Each design variable is one dimension in a design space, where any particular set of variables is indicated as a point. A design space with n variables is a n-dimensional hyperspace. A design which satisfies all the constraints is a feasible design and the set of all feasible designs form the feasible region. In Fig. 2-1 the design space of a three bar truss problem is illustrated, which was first presented by Fox in 1960.

2.4 Mathematical Formulation

The nonlinear constrained optimization problem can be stated mathematically as follows:

Find:

$$\boldsymbol{x} = [x_1, x_2, \dots, x_N]^T \qquad \text{design variables} \qquad (2.1)$$

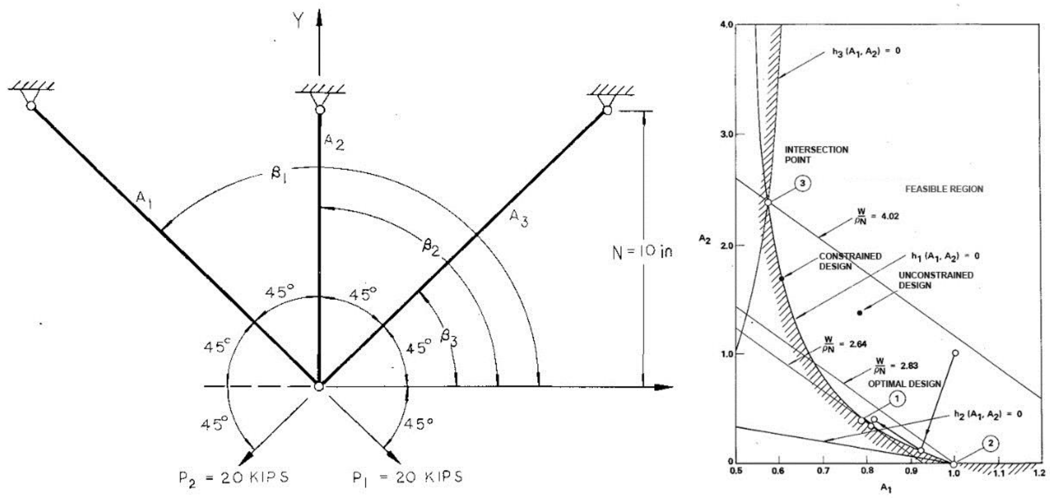


Figure 2-1: Three-bar truss problem and design space (Adapted from Schmit, 1981)

To minimize:

$$\min f(\boldsymbol{x}) \qquad \text{objective function} \qquad (2.2)$$

Subject to:

$$\begin{aligned} g_j(\boldsymbol{x}) &\leq 0 & j &= 1, \ldots, J & &\text{inequality constraints} \\ h_k(\boldsymbol{x}) &= 0 & k &= 1, \ldots, K & &\text{equality constraints} \qquad (2.3) \\ x_i^{(l)} \leq x_i &\leq x_i^{(u)} & i &= 1, \ldots, N & &\text{side constraints} \end{aligned}$$

2.5 Classification of the Structural Optimization Tasks

Structural optimization tasks can be classified according to type of design variables, since the applicable solution strategies are also chosen according to them as shown in Fig. 2-2 (Schumacher, 2013):

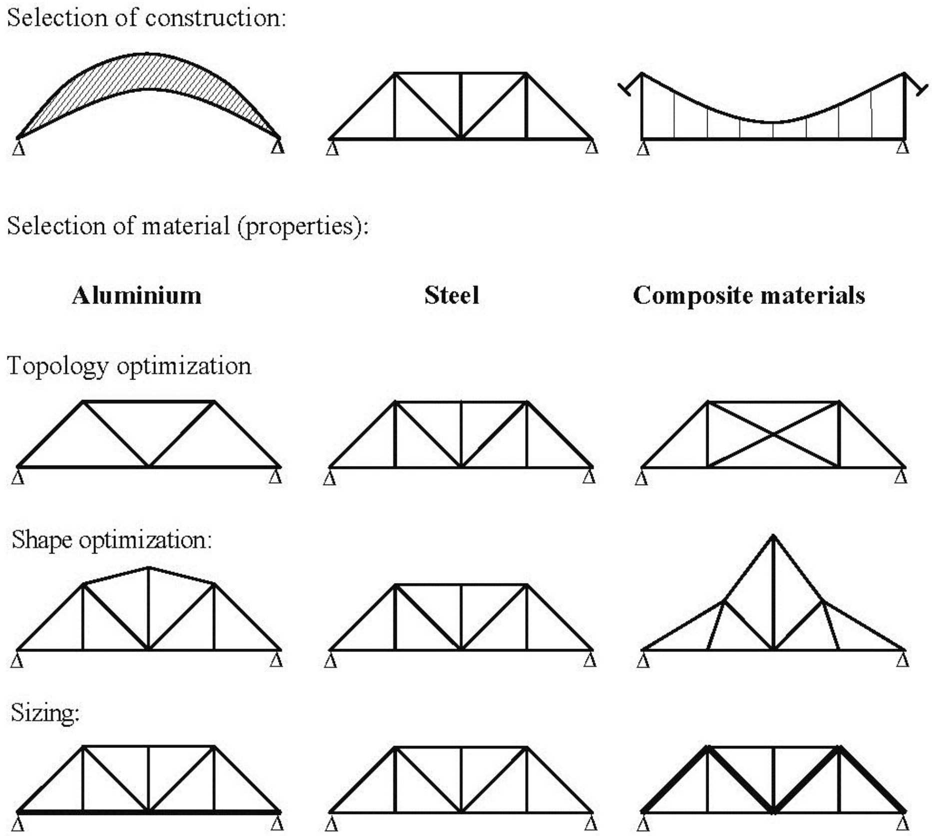


Figure 2-2: Classification of structural optimization tasks according to design variables (Schumacher, 2013).

2.5.1 Size Optimization

In size optimization problems, the goal may be to find the optimal thickness distribution of plate and shell segments or the optimal member cross-sectional areas of bars in a truss structure. The optimal thickness distribution minimizes (or maximizes) a physical quantity such as the peak stress, deflection, etc. while imposed constraints on the state and design variables are satisfied. The thickness of a plate or the cross-sectional area of a bar is the design variable and the state variable may be their deflection. The main feature of the sizing problem is that the layout of the structure and the state variables are prescribed and fixed throughout the optimization process. A size optimization problem for a truss tower structure is shown in Fig. 2-3.

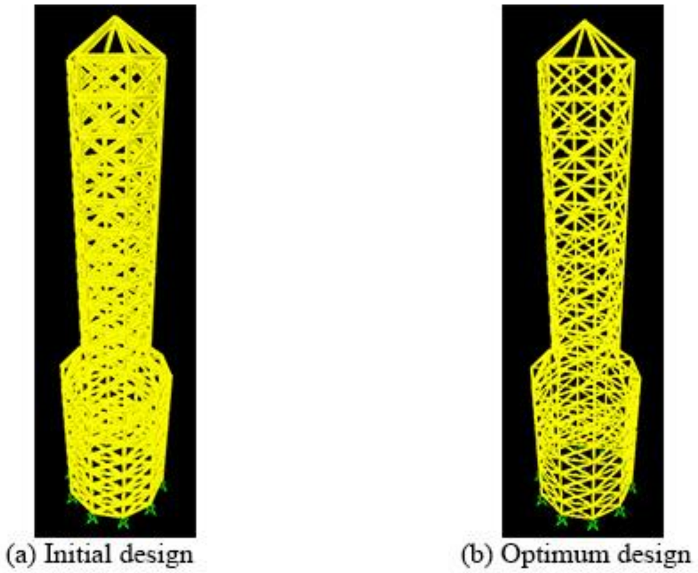


Figure 2-3: Size optimization of a steel tower structure

2.5.2 Shape Optimization

In shape optimization, the aim is to find the best possible geometrical arrangement of the structural members. The optimum node locations in a finite element model of the structure has to be determined, without changing the connectivity of structural elements.

In the engineering field, the first shape optimization problem was defined by Galileo in 1638 at his famous book titled 'Dialogues Concerning Two New Sciences,' where he used the simple bending theory of beams considering the uniform strength criterion to find the optimum shape of a cantilever beam with constant width and under tip loading as given in Fig. 2-4. He was often credited with the first published theory of the strength of beams in bending, but it was discovered in 1967 in the National Library of Spain that this theory was initiated by Leonardo da Vinci in his work "The Codex Madrid" published in 1493. Galileo proved that under imposed constraints, the optimum shape of a cantilever beam should be parabolic. This statement can be recognized as a fundamental model for fully stressed design (FSD) concept.

Structural optimization methods have been implemented into commercial finite element programs to treat large shape optimization problems. However, considerable manual efforts to define design variable and added constraints, and to integrate with CAD system and optimizer are vitally necessary.

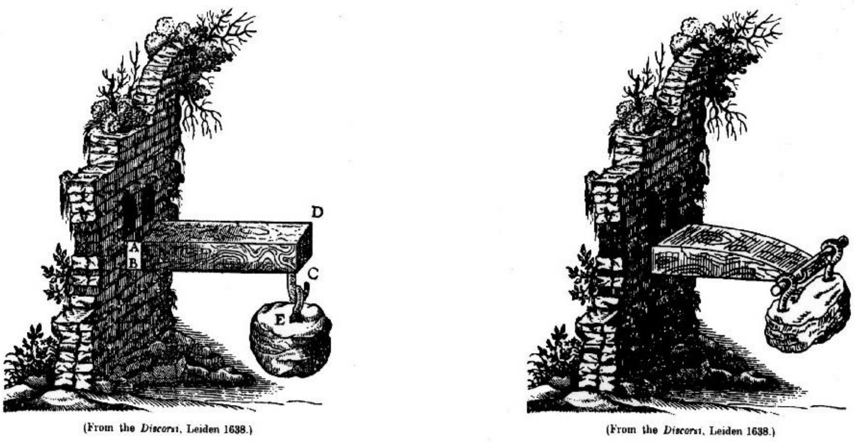


Figure 2-4: Shape optimization defined by Galileo in 1638 (Crew and Salvio, 2010)

2.5.3 Topology Optimization

The topology optimization method solves the most general structural optimization problem of distributing a given amount of material freely in the design space such that performance is optimized (Sigmund, 2000). Before topology optimization, the physical size and the shape and the connectivity of the structure, which define together the topology of the structure are unknown. In 1904, Michell derived the mathematics behind structures of least volume, or optimal structures and his work provided a basis for topology optimization of structures. The computations for topology design is shown in Fig. 2-5.

Topology optimization can be classified as geometrical topology optimization and material topology optimization as shown in Fig. 2-6.

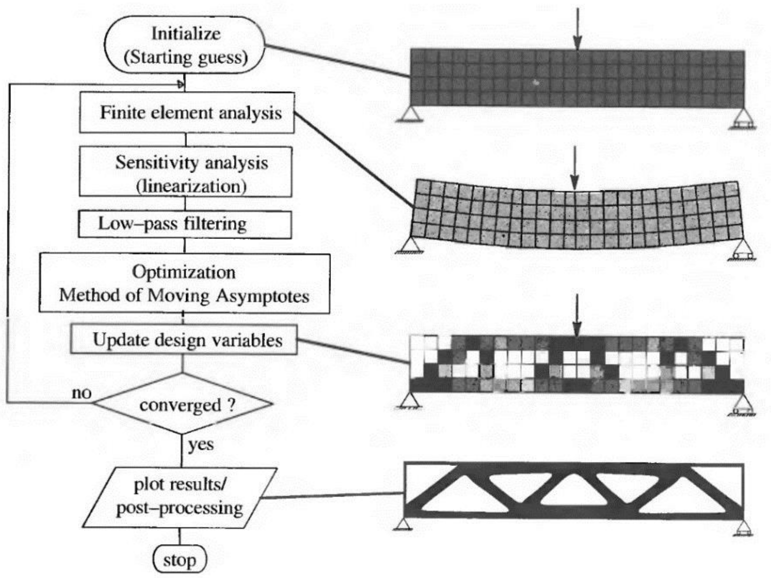


Figure 2-5: The flow of computations for topology design (Bendsoe and Sigmund, 2003)

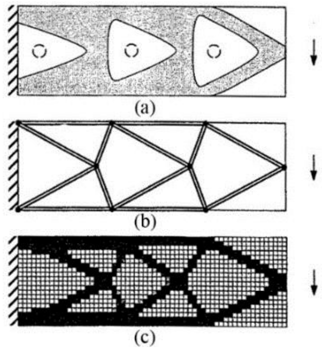


Figure 2-6: Types of topology optimization, (a) geometrical (bubble-method), (b) discrete material distribution, (c) continuous material distribution (Maute, 1998)

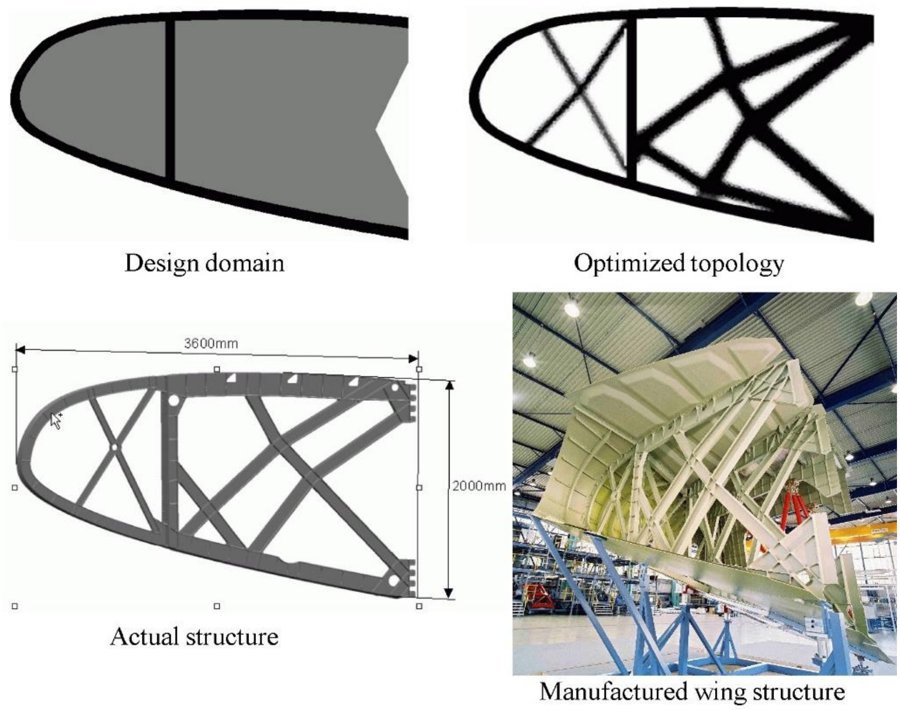


Figure 2-7: Mimicking of actual industrial design process. Rib structure in front part of airplane wing at EADS (courtesy of EADS Military Aircraft)

As shown in Fig. 2-7, after topology design in aeronautics for the design of integrally stiffened machined ribs for the inboard inner fixed leading edge of the new airliner, the Airbus 380, a new type of structure was devised for the ribs which gave a weight benefit against traditional (up to 40%) and competitive honeycomb / composite designs.

2.5.4 Selection of Material Properties

Materials can be selected as steel, aluminum, magnesium, composite etc.

2.5.5 Selection of Construction

Construction type should be selected as plate girder, truss-like structure or composite structure etc.

2.6 Optimization Methods

Optimization methods can be categorized in various ways, but in a very general way they can be classified in two categories as function and parameter optimization methods.

In function optimization, the problem under consideration is formulated by a number of unknown functions and through the optimization process, where the main goal is to find the optimum form of these functions. For function optimization methods such as differential calculus, the calculus of variations, etc. are used. On the other hand, parameter optimization methods search the optimum values of design variables for the specified problem. Mathematical programming, quadratic programming, methods of feasible directions, optimality criteria (OC), and metaheuristic methods are some subsets of parameter optimization methods.

Finally, a classification of various numerical optimization methods is shown in Fig. 2-8.

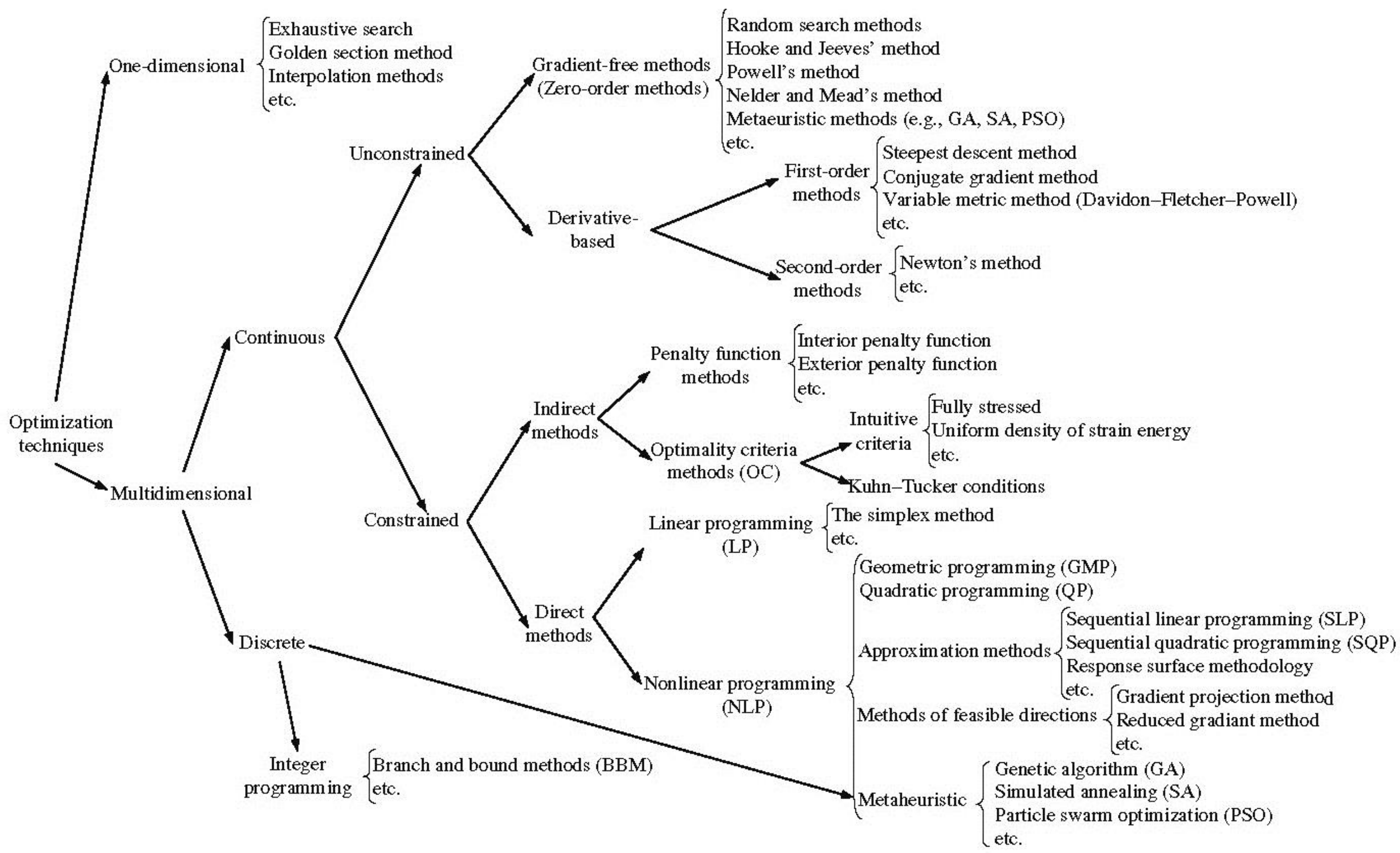


Figure 2-8: Classification of numerical optimization methods (Gandomi et al, 2013)

CHAPTER 3

PROBLEM STATEMENT

3.1 Introduction

The objective of this study is to investigate the use and application of ABC algorithm in the realm of structural optimization, and analyze the performance of ABC algorithm. The investigations will unveil the capabilities and potentials of this algorithm in practical problems from structural optimization literature and provide a guidance to potential users with object-oriented software implementation ABC-SOP2014.

This chapter describes the mathematical statement of the structural optimization problem for the structural model based on the formulation and definitions aforementioned in Chapter 2. Constraint implementation is done according the provisions of American Steel Institute of Steel Construction-Allowable Stress Design (AISC-ASD) code specifications.

3.2 Design Variables

Mostly in practical sizing design optimization problems of steel structures, the cross-sectional areas of structural members are chosen from a list of standard sections available on the market such that the final design satisfies the design constraints determined by technical specifications of standards. Consequently, the cross-sections of structural steel members refer to discrete sizing design variables and the optimization process is called as discrete structural optimization.

The optimum design procedure begins by first deciding the initial values of area variables. They can be selected in any way; feasible or infeasible, for simplicity equal to each other or not, obtained from a structural design software or from literature, even just by engineering judgment or design experience. Benchmark problems solved have shown the algorithm achieves promising results with all these initial design point selections.

During modeling the structural optimization system, members are grouped strategically to reduce the population size and computational cost, as well as fabrication time. For small systems each member cross-section can be handled as design variable without grouping.

A vector of discrete integer values **I** corresponding to the sequence numbers of standard steel sections in a given section table for N_d members of the structural system

$$\mathbf{I}^T = \left[I_1, I_2, \ldots, I_{N_d}\right] \tag{3.1}$$

constitutes the design space for each member group. A cross-section from vector **I** is assigned randomly to each member group in every iteration step. After N_g iteration the assignment procedure is completed for one step. N_g represents total number of member groups.

Mathematically, design variables can be formulated as a vector of cross-sectional areas **A** for each member group:

$$\mathbf{A}^T = \left[A_1, A_2, \ldots, A_{N_g}\right] \tag{3.2}$$

3.3 Objective Function

In this thesis, the constrained weight (W) minimization of steel structures is defined as the objective function as formulated below:

$$min W = \sum_{k=1}^{N_g} A_k \sum_{m=1}^{N_k} \rho_m L_m \tag{3.3}$$

where W is weight, N_g is the total number of member groups in the structure, A_k is cross-sectional area of the kth member group, N_k is number of members in member group k and ρ_m, L_m are unit weight and length of the mth member in the kth member group, respectively Saka (1990).

3.4 Constraints

Typically in any optimization problem, the design variables accordingly the final solution are controlled by the constraints imposed on the problem. In the present study, constraints are defined according to the provisions of AISC-ASD (1989) design code for pin-jointed truss type structures.

For truss structures, constraints can be shown in general form as follows:

$$g_m = \frac{\sigma_m}{(\sigma_m)_{all}} - 1 \leq 0 \quad ; \quad m = 1, \dots, N_m \tag{3.4}$$

$$s_m = \frac{\lambda_m}{(\lambda_m)_{all}} - 1 \leq 0 \quad ; \quad m = 1, \dots, N_m \tag{3.5}$$

$$\delta_{j,l} = \frac{d_{j,l}}{(d_{j,l})_{all}} - 1 \leq 0 \quad ; \quad j = 1, \dots, N_j \tag{3.6}$$

In Eqns. (3.4-3.6), the functions g_m , s_m and $\delta_{j,l}$ are referred as constraints being bounds on stresses, slenderness ratios and displacements, respectively; σ_m and $(\sigma_m)_{all}$ are the computed and allowable axial stresses for the mth member, respectively; λ_m and $(\lambda_m)_{all}$ are the slenderness ratio and allowable value for mth member, respectively; N_j is the total number of joints; and $d_{j,l}$, and $(d_{j,l})_{all}$, are the computed displacements and allowable displacement, respectively; lastly, l and j represent direction and joint id, respectively.

The allowable tensile stress for the members subjected to axial tension force shall not exceed the values calculated in Eq. (3.7):

$$(\sigma_t)_{all}=0.60F_y \text{ (on the gross area)}$$
$$(\sigma_t)_{all}=0.50F_u \text{ (on the effective net area)} \tag{3.7}$$

where F_y is the yield strength and F_u is the ultimate tensile strength of the material.

The allowable tensile stress of structural members under axial compression force is calculated considering the two possible failure modes of the members known as elastic and inelastic buckling.

$$C_c = \sqrt{\frac{2\pi^2 E}{F_y}} \tag{3.8}$$

$$(\sigma_c)_{all} = \frac{\left[1 - \frac{(K_m L_m / r_m)^2}{2C_c^2}\right] F_y}{\frac{5}{3} + \frac{3(K_m L_m / r_m)}{8C_c} - \frac{(K_m L_m / r_m)^3}{8C_c^3}} \quad ; \quad \lambda_m < C_c \qquad \text{(inelastic buckling)} \tag{3.9}$$

$$(\sigma_c)_{all} = \frac{12\pi^2 E}{23(K_m L_m / r_m)^2} \quad ; \quad \lambda_m \geq C_c \qquad \text{(elastic buckling)} \tag{3.10}$$

The maximum slenderness ratio is limited to 300 for tension members, and it is recommended to be 200 for compression members. Hence, the design constraints related with slenderness for structural members under axial tension or compression can be expressed as in Eqn. (3.11)

$$\lambda_m = \frac{K_m L_m}{r_m} \leq 300 \text{ (for tension members)}$$

$$\lambda_m = \frac{K_m L_m}{r_m} \leq 200 \text{ (for compression members)} \tag{3.11}$$

In Eqns. (3.8-3.11), E represents the modulus of elasticity, and C_c stands for critical slenderness ratio parameter. K_m is the effective length factor and is taken as 1 for all truss members, Lm is the length of the *m*th member, and r_m represents minimum radii of gyration.

where, K_m , Lm and r_m are mentioned before.

3.5 Constraint Handling Procedure

In this thesis, the constraints are handled by integrating a penalty function term into the objective function. The constraint integrated objective function is defined to evaluate the feasible and infeasible designs, which proportionally penalizes the designs with respect to the sum of constraint violations as shown in Eqn. (3.12).

$$\Phi = W\left[1 + \alpha \sum_{i=1}^{N_g} \left(\sum_{k=1}^{N_m} g_m + \sum_{k}^{N_m} s_m\right) + \alpha \sum_{j=1}^{N_j} \sum_{j=1}^{3} d_{j,k}\right] \tag{3.12}$$

In Eqn. (3.12), Φ represents the fitness score which is the penalized objective function and α is referred to as the penalty coefficient to be used to adjust the intensity of penalization. The details of constraints will be presented in section 4.6.

CHAPTER 4

ARTIFICIAL BEE COLONY (ABC) ALGORITHM AND SOFTWARE DEVELOPMENT FOR STRUCTURAL OPTIMIZATION

4.1 Introduction

Artificial bee colony algorithm has been widely used for all types of optimization problems in various civil engineering disciplines and other disciplines, since it has been introduced originally by Karaboga (2005) for solving numerical optimization problems based on simulating real bees social behavior, foraging behavior as a heuristic.

Further modifications and improvements of the ABC algorithm have been carried out by Karaboga and Basturk (2007a). The main outlines of the ABC algorithm have been presented by Karaboga and Basturk (2007b). Later on, ABC algorithm has been applied by Akay and Karaboga (2009) on various numerical test functions and the results have been compared with other well-known optimization algorithms such as the GA, PSO and HS. Recently, a modified version of ABC algorithm for constrained optimization problems has been proposed by Karaboga and Akay (2011).

In this study it is aimed to implement ABC algorithm to discrete size optimization of real size steel truss structures. To find the minimum weight of steel structures by most appropriate cross-sections of structural elements, while satisfying the constraints imposed to the structure, software called ABC-SOP2014 has been developed. The developed software provides minimum weight design of both truss and frame structures, but in this study it is applied only to truss structures.

In this chapter the literature survey is done firstly, and then swarm intelligence is introduced. Consequently, artificial bee colony (ABC) algorithm is discussed broadly. Thereafter, the constraint handling method is presented. An example two-bar benchmark problem is solved to clarify the use of artificial bee in structural optimization comparing the results with other four optimization techniques. Finally the details of the developed structural optimization software have been overviewed.

4.2 Literature Survey

In the literature, there are a huge number of studies about Artificial Bee Colony (ABC) Algorithm ranging from normal equations to structural design problems in a variety of engineering, finance and other areas. In the following, the applications of the technique in structural optimization and civil engineering are reviewed first.

4.2.1 Studies Related to Structural Optimization

A modified ABC (MABC) algorithm was proposed by Hadidi et al. (2010) for size optimization of planar and space truss structures under stress, displacement and buckling constraints by applying the concept of probability to modify neighborhood search method and by modifying the onlooker and scout phase. Their results outperformed the classic ABC algorithm in all benchmark problems.

Sonmez (2011a) integrated an adaptive penalty function approach (ABC-AP) into the ABC algorithm in order to minimize the weight of steel truss structures. Constraints were handled with the adaptive penalty function method within ABC to get rid of the drawbacks of Deb's selection method and the static penalty function methods. The efficiency of the ABC-AP for optimum design of truss structures was studied in five test problems up to 200 elements with fixed-geometry and continuous sizing variables subjected to multiple loading conditions. When the results of the proposed enhanced algorithm were compared with other optimization methods in the literature, it was shown that the approach is efficient as an optimization technique for structural designs.

Sonmez (2011b) made modifications in original ABC with improved performance of the algorithm for discrete optimum design of truss problems. Four truss problems with up to 582 structural elements and 29 design variables were solved to test the effectiveness of the modified algorithm and the results demonstrated the robustness and effectiveness of the proposed method for discrete optimization design of truss structural problems

Aydogdu et al. (2012) used ABC algorithm for solving two discrete constrained structural optimization problems. They designed a four story, three-bay 132 member irregular steel space frame and an eight-story, 1024 member regular steel space frame selecting the sequence numbers of W steel sections from a section pool as design variables. The design constraints were implemented according to the provisions of LRFD-AISC (Load and Resistance Factor Design [LRFD], 2001) which covers the displacement limitations, inter-story drift restrictions, ultimate strength requirements and geometric constraints. They obtained lighter designs with better performance than both of the optimum designs determined by the dynamic harmony search and ant colony optimization algorithms. Finally, they concluded that, artificial bee colony algorithm is a robust and efficient approach that can be effectively used to determine the optimum designs of large scale, real size steel space frames.

Gerhardt and Gomes (2012) applied ABC algorithm to three classic benchmark problems, spring design and optimization of a 10 bars plane truss and optimization of a 52 bars space truss. They used the same architecture as in Akay and Karaboga (2010) but with the minimum penalty rule, that says that the penalty should be kept as low as possible so that an infeasible solution could not be optimal Coello Coello (2002). Their results indicate that the ABC algorithm is an effective global optimizer with relative high computational cost.

Fiouz et al. (2012) used ABC algorithm with the fly-back mechanism to impose the constraints for discrete optimization of 10-bar plane truss and 25-bar and 72-bar space truss. In some cases they found same results as other methods used in literature, and in other applications their method produced significantly better results. The fly-back mechanism technique significantly improved the rate of convergence and the accuracy in comparison with other methods. They concluded also, that the ability to reduce the structural weight and the computational cost proves that this algorithm is one of the most powerful algorithms available for structural truss weight optimization.

Talatahari et al. (2012) coded ABC algorithm in MATLAB to study four skeletal structure benchmark problems and to show the efficiency of the ABC algorithm. The results of their study revealed that the ABC algorithm offers results as good as or better than other optimization methods and can be used effectively for solving such type of problems satisfying various constraint conditions.

Degertekin (2012) developed an ABC algorithm for the optimum design of geometrically non-linear steel frames. He investigated the numerical efficiency of the ABC algorithm by solving weight minimization problems of three steel frames taken from literature and found better designs than the GA and the HS algorithms in shorter time under strength, displacement and size constraints.

Degertekin and Hayalioglu (2013) developed an improved artificial bee colony algorithm (IABC) for size optimization of truss structures in order to enhnce the efficiency of the ABC algorithm. Solving a twenty-five bar space truss from literature by IABC they verified that the convergence capability of the IABC algorithm is significantly better than that of the artificial bee colony algorithm with an adaptive penalty function (ABC-AP) and it also obtained better design than the ABC-AP and other heuristic search algorithms compared. Less than 0.1% standard deviation of 20 independent runs in comparison with the average weight in the design example proves that the IABC algorithm converges to near global optimum and it is not sensitive to the initial designs.

Three metaheuristic algorithms, namely harmony search (HS), artificial bee colony (ABC) and firefly algorithm (FA) have been evaluated by Miguel and Miguel (2013). They solved seven benchmark truss problems and performed the optimization of a realistic transmission tower. To optimize these structures on shape and sizing under multiple loading conditions, they used penalty approach dealing with different type of constraints. They obtained better results than the literature in three of the seven examples considered, and in the other four examples the results were approximately equal to the best one obtained in literature, without constraint violation. In complex examples, involving shape and size optimization with multiple natural frequency and buckling constraints, the results of used algorithms were better than the results in literature and parameter fine-tuning was not necessary to obtain these results. The computational time was relatively short to find the optimal solution.

Joubari et al. (2013) performed a structural truss mass optimization on size under frequency constraints using ABC and gravitation search algorithm (GSA). Their results showed that both algorithms reached, better results than the literature in three of the four examples considered, and in the other example the structure is approximately equal to the best one found, emphasizing the excellent capacity of both methods.

Carbas et al. (2013) carried out a comparative study of three metaheuristics for optimum design of engineering structures. They selected the Firefly Algorithm (FFA), Artificial Bee Colony (ABC), and Cuckoo Search (CS) algorithms and designed 132-members space steel frame by using these three different algorithms to investigate the minimum weight design. In the design example considered, the design constraints include the displacement limitations, inter-story drift restrictions, strength requirements for beams and beam-columns which are formulated according to provisions of LRFD-AISC (Load and Resistance Factor Design of American Institute of Steel Institution). They compared the optimum designs obtained by FFA, ABC, and CS algorithms by each other as well as by those attained by the Dynamic Harmony Search and Ant Colony Optimization algorithms. The lightest optimum design is attained by the artificial bee colony algorithm. Adaptive Firefly algorithm (AFFA) is the second bests. It is also noticed the performances of CS and DHS algorithms are close to each other in this particular problem. The optimum design obtained by Ant Colony Optimization algorithm is the heaviest weight among the other algorithms which is 8.73% heavier than the one determined by ABC algorithm.

4.2.2 Studies Related to Other Applications of ABC in Civil Engineering

Kang et al. (2009a, 2009b) proposed the hybrid simplex artificial bee colony algorithm (HSABCA) which combines artificial bee colony algorithm with the Nelder-Mead simplex search (NMSS) method for inverse analysis problems. They applied the new algorithm which combines the local search ability of NMSS and the global search ability of ABC algorithm to parameter identification of concrete dam-foundation systems. They compared the overall search ability of HSABCA with the basic ABC algorithm and a real coded genetic algorithm (RCGA) on two examples: a gravity dam and an arc dam. Obtained results indicate that the proposed algorithm is an efficient tool for inverse analysis of dam-foundation systems and it performs much better than the other two stochastic algorithms on such problems.

Li et al. (2011a) showed using four examples the reliability and accuracy of the ABC algorithm in reliability analysis of engineering structures.

Li et al. (2011b) combined fuzzy c-means clustering (FCM) with ABC algorithm to overcome the sensitivity to the initialization of clustering centers and to prevent trapping into local optima of FCM during risk analysis of dam. Results show that it is more accurate and robust than FCM, and it is an efficient tool for risk analysis of dams.

Su et al. (2012) used ABC algorithm for pile group load optimization. They concluded that ABC is feasible and has the advantages of high efficiency and easy implementation for pile group load optimization.

Prakash (2012) tried to improve the exploitation capability which in turn accelerates the convergence of ABC by embedding convex linear combination in its onlooker bee phase, because a poor balance between exploration and exploitation may result in a weak optimization method which may cause premature convergence, trapping in a local optima, and stagnation. The modified variant called BABC was applied to determine and improve the seismic location in the Earth's crust and upper mantle. The proposed variant gave good results and enhanced the accuracy of the hypocentral parameters.

Mandal et al. (2012) used an integrated approach of rough set theory and ABC trained support vector machine leak detection of pipeline with maximum accuracy.

Hossain and El-shafie (2013) presented a paper on developing an optimum reservoir release policy by using ABC algorithm. The paper presents a study on developing an optimum reservoir release policy by using ABC algorithm for the Aswan high dam of Egypt. After using the actual historical inflow, the release policy succeeded in meeting demand for about 98% of total time period.

Jahjouh et al. (2013) studied to obtain the optimum design for reinforced concrete continuous beams in terms of cross section dimensions and reinforcement details using a fine tuned Artificial Bee Colony (ABC) Algorithm while still satisfying the constraints of the ACI Code (2008). Four RC beams of varying complexity are presented and optimized.

Sun et al. (2013) utilized a modified ABC algorithm to identification of structural models and demonstrated the effectiveness, robustness and efficiency of the method.

Kang et al. (2013) proposed an artificial bee colony algorithm with a multi-slice adjustment method for locating the critical slip surfaces of soil slopes. They tested the proposed algorithm on six benchmark problems and showed its reliable performance by solving these problems. Compared with several other population-based algorithms like PSO, MHS, spline-based GAand RCGA, their method demonstrated strong competitive capabilities in terms of convenience, efficiency and accuracy.

Ozturk and Durmus (2013) investigated optimum cost design of columns subjected to axial force and uniaxial bending moment via ABC algorithm implementing the design constraints according to ACI 318-08 and studies in the literature. They selected the height and width of the column, diameter and number of reinforcement bars as design variables and the cost of unit length of the column consisting the cost of concrete, steel, and shuttering as the objective function. Deb's constraint handling method was used. They obtained nearly same values as the existing values in the literature.

Yahya and Saka (2014) used in their study a multi objective artificial bee colony (MOABC) algorithm via Levy flights algorithm to obtain the solution of the construction site layout planning (CSLP) problem. The objective of the study was to optimize the dynamic layout problem under two objective functions of minimizing the safety hazards/environmental concerns and the total handling cost of interaction flows between facilities. The performance of MOABC with Levy flights is demonstrated on a real benchmark construction engineering of construction site layout planning problem and the optimum solution obtained is compared with Basic-MOABC model, max–min Ant system (MMAS) model, and the original construction site layout of the studied problem. They concluded that, the results indicated that MOABC via Levy flights performs better than the above mentioned algorithms and the proposed model was successfully applied to practical case studies and proved to be robust and efficient.

4.2.3 Studies Related to Other Areas of Engineering Optimization

Standard ABC algorithm is effectively used for solving unimodal and multimodal numerical optimization problems even for solving unconstrained optimization problems. Improved versions of ABC algorithm are developed to handle constrained problems in various areas such as economics, engineering design, allocation and location problems, visual target recognition, image clustering, reactive power optimization, protein structure prediction, data mining, software testing, vehicle routing, neural network training, job shop scheduling, bioinformatics.

For further applications in a wide range information can be found in Baykasoglu et al. (2007), Karaboga et al. (2012), Bolaji et al. (2013), Balasubramani and Marcus (2013), Bansal et al. (2013).

4.3 Swarm Intelligence

Karaboga (2005) defines two basic concepts, self-organization and division of labor, as necessary and sufficient properties for obtaining swarm intelligent behavior.

According to Bonabeau et al. (1999) self-organization can be defined as a set of dynamical mechanisms which result in structures at the global level of a system from interactions among its lower-level components. Between the components of the system the rules specifying the interactions are executed on the basis of purely local information, without global knowledge or global planning. However insect colonies can collectively build fascinating structures and achieve goals which individual insects are incapable of achieving alone. A behavioral model of self-organization is proposed for a colony of honey bees by Seeley (1995). Self-organization can be described with four basic properties:

i) Positive feedback promotes the creation of convenient structures. Recruitment and reinforcement are typical examples of positive feedback. Trail laying and following in some ant species or dances in bees can be shown as the examples of recruitment. Seeley, Camazine and Sneyd (1991) have confirmed that foragers can home in on the best food source through a positive feedback.

ii) Negative feedback counterbalances positive feedback, which leads to the stabilization of the collective pattern. Limited number of available foragers, saturation, food source exhaustion, crowding or competition at the food sources might give rise to a negative feedback mechanism.

iii) Fluctuations such as random walks, errors, random task switching among swarm individuals enable the discovery of new solutions.

iv) All cases of self-organization rely on multiple interactions. In general, self-organization requires a minimal density of mutually tolerant individuals capable to make use of the results of their own activities as well as of others' activities.

Different tasks inside a swarm are often performed simultaneously by specialized individuals and this phenomenon is called division of labor. Simultaneous task performance by specialized workers is believed to be more efficient than sequential task performance by unspecialized workers. While parallelism avoids task switching, specialization allows greater efficiency of individuals in task performance.

The basic properties related to self-organization of honey bees are as follows:

i) Positive feedback: With the increase of nectar amount, the number of onlookers visiting them increases, too.

ii) Negative feedback: The exploration process of a food source abandoned by bees is stopped and it helps to labor allocation.

iii) Fluctuations: The scouts carry out a random search process, which enables the discovery of new food sources.

iv) Multiple interactions: Information about food source locations is shared on the dance area.

4.3.1 Behavior of Honey Bee Swarm

The foraging behavior of honeybees, the process of seeking for nectar in flowers is an optimization process in nature. This social behavior has been modeled successfully as an optimization technique.

The nectar gathering process of honeybees, which emerges their collective intelligence consists of three essential elements (Karaboga, 2005) as shown in Fig. 4-1:

Food sources: The value of a food source depends on many factors, such as the proximity to the hive, energy concentration of nectar and the ease to extract it. For the sake of simplicity, it is possible to represent the profitability of a source with a single quantity, its fitness.

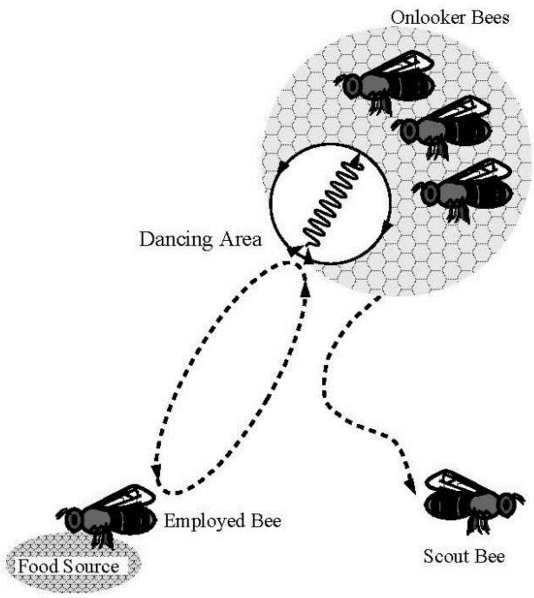


Figure 4-1: Basic elements of foraging behavior

Employed foragers: These bees are associated with a particular food source which is exploited by them. Employed bees carry information about their food source, such as distance, direction and profitability to other collector bees and share it with a certain probability.

Unemployed foragers: They are constantly looking for a food source to exploit. They can be classified as the scouts, searching for new food sources in the neighborhood of the hive and as the onlookers, waiting in the hive and choosing a food source according to the information shared by employed foragers.

The encoding of information is discovered by Karl von Frisch is as follows: The duration of waggle dance represents the distance and profitability of the food source, namely one second of a waggle dance represents 1,000 meters of flight. And the angle of the dance with respect to the comb indicates the location of the food source relative to the sun as shown in Fig.4-2 (Seeley, 2010).

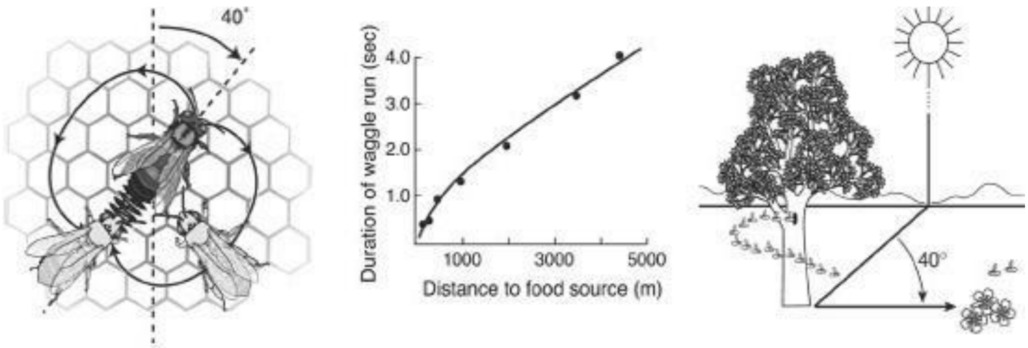


Fig. 1.4 How a dancing bee encodes information about the distance and direction to a rich patch of flowers. Distance coding: The duration of each waggle run is proportional to the length of the outbound flight. Direction coding: Outside the hive the bee notes the angle of her outbound flight relative to the sun's direction, and then inside the hive she orients her waggle runs at the same angle relative to straight up on the comb. Two followers are acquiring the dancing bee's information.

Figure 4-2: Encoding information between honeybees (Seeley, 2010)

4.4 Artificial Bee Colony (ABC) Algorithm

The pseudo-code of the basic ABC algorithm, can be stated step by step as in Fig. 4-3 and a basic outline of ABC algorithm is presented in Fig. 4-4.

Parameters: *sn*, limit
Initialize the food sources $\vec{x}_i$ randomly
Evaluate fitness $f(\vec{x}_i)$ of the population
cycle= 1
repeat
    for *i* = 1 to s*n* / 2 **do** {*Employed phase*}
        for *j* = 1 to *D* do
            Produce a new food source $\vec{v}_i$ in the neighborhood of the food source $\vec{x}_i$for the employed bee by using
            $\vec{v}_i = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right)$
            Select *k* at random such that
            $k \in \{1, 2, \ldots, sn\}$, $k \neq i$, $\phi \in [-1, 1]$
        end for
        Evaluate solutions $\vec{v}_i$ and $\vec{x}_i$
        if $f(\vec{v}_i)$ is better than $f(\vec{x}_i)$ **then**
            Greedy selection
        else
            $count_i = count_i + 1$
        end if
    end for
    for *i* = *sn* / 2 + 1 **to** *n* **do** {*Onlooker phase*}
        Calculate selection probability
        $P(\vec{x}_k) = \frac{f(\vec{x}_k)}{\sum_{k-i}^{sn} f(\vec{x}_k)}$
        Select a bee using the selection probability
        Produce a new solution $\vec{v}_i$ from the selected bee
        Evaluate solutions $\vec{v}_i$ and $\vec{x}_i$
        if $f(\vec{v}_i)$ is better than $f(\vec{x}_i)$ **then**
            Greedy selection
        else
            counti = *counti* + 1
        end if
    end for
    for *i* = 1 **to** *sn* **do** {*Scout phase*}
        if $count_i > limit$ **then**
            $\vec{x}_i$=random
        end if
    end for
    Memorize the best solution achieved so far
cycle=cycle+1
until cycle=Maximum Cycle Number (MCN)
Post process results and visualization

Figure 4-3: Detailed Pseudo code of the ABC Algorithm

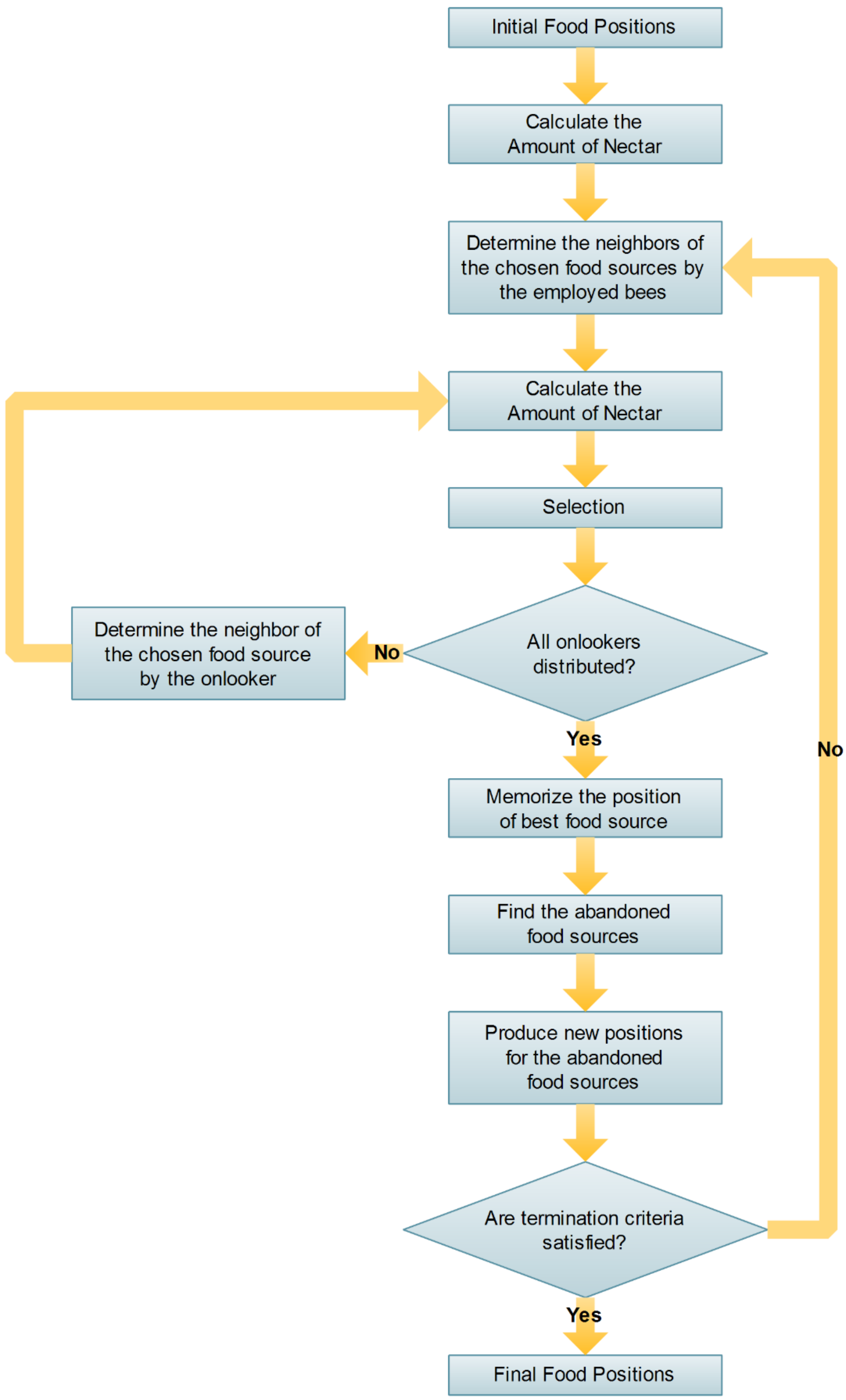


Figure 4-4: Flowchart of ABC algorithm

4.5 ABC Algorithm for Discrete Optimization

ABC algorithm was firstly proposed to solve unconstrained optimization problems. Karaboga and Akay (2011) proposed a modified ABC algorithm for constrained optimization problems. At first step, a randomly distributed initial population of *sn* solutions is generated, where each solution is a *D*-dimensional vector. *D* represents the number of design variables and ${}_{i}x$ represents the i^{th} food source in the population.

$${}_{i}x_j^0 = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}), \quad i = 1,2,...,sn \quad j = 1,2,...,D \tag{4.1}$$

where $x_j^{\min} \le x_j \le x_j^{\max}$

When the design variables are selected from a continuous design space Eqn. (4.1) is used. If the design variables are to be selected from a set of discrete section pool, Eqn. (4.1) is modified as follows:

$${}_{i}I_j^0 = I_j^{\min} + INT\left[rand(0,1)(I_j^{\max} - I_j^{\min})\right], \quad i = 1,2,...,sn \quad j = 1,2,...,D \tag{4.2}$$

where $I_j^{\min} \le I_j \le I_j^{\max}$

In Eqn. (4.2) the integer value of I_j denotes the index of the design variable. In truss optimization I_j represents the index of cross-sectional area A_j in the available profile list vector. $I_j^{\min}$ and $I_j^{\max}$ refer to the first and last profile index in section pool.

After all the bees return to the hive with a certain amount of nectar, the first half (sn/2) that found the best food sources become “employed bees”. While performing truss optimization, the amount of nectar refers to the weight of the truss and the best food sources are the lightest trusses explored by foragers. The remaining bees are called “unemployed bees” or “onlooker bees”. They watch the waggle dance to decide which of the employed bees to follow.

Each food source possesses one employed bee. The number of onlooker bees which will fly to a food source depends on the amount of nectar at that source. The onlooker bees select the food source according to a probability proportional with the amount of nectar existing at the food source. Singh (2009). The probability p_i for i^{th} source is calculated as shown below:

$$p_i = \frac{\frac{1}{_i f(x)}}{\sum_{j=1}^{sn/2} \frac{1}{_j f(x)}} \qquad \text{or} \qquad p_i = \frac{\frac{1}{_i fW(I)}}{\sum_{j=1}^{sn/2} \frac{1}{_j W(I)}} \tag{4.3}$$

A new candidate food source is calculated based on information inherited from the employed bee:

$$_i I_j^{new} = I_j^{best} + INT\left[\phi(_i I_j^{best} - {_k I_j^{best}})\right], \; i = 1, 2, ..., sn/2 \qquad j = 1, 2, ..., D \tag{4.4}$$

In Eqn. (4.4) ϕ is a random number between -1 and 1. The left hand subscript (*i*) indicates the solution number while the right hand script (*i*) represents the design variable number. *k* is a randomly chosen integer number between 1 and *sn*/2 that has to be different from *i*. As the difference between the parameters $_i I_j^{best}$ and $_k I_j^{best}$ decreases, the perturbation on the new candidate food position $_i I_j^{new}$ decreases. Therefore, as the search approaches to the optimum solution on the search space, the step length adaptively decreases. If the food level in the new location is better then the old one, the new position becomes best food source; otherwise, the old location is preserved as the best food source.

Since the ABC algorithm is iterative, a food source is discarded by its employed bee, if there is no improvement in the amount of nectar from a certain food source after a predefined iteration (LIMIT). If a scout accidentally discovers a rich, unexplored food source, it becomes an employed bee.

4.5.1 Diversification Generation Method for Initial Population

In this thesis, the diversification generation method is used for generating the diverse initial solution set. The diversification generation method does not consider the objective function, it only focuses on diversification.

First, the last design produced by design module of SAP2000 is used for initial seed for the method with an acceptable design, which enables the algorithm to find the optimum results very rapidly by reducing number of iteration cycles and avoiding redundant computations. The initial seed is randomized by using geometric distribution during generation of other members of the population as follows:

$$x_i' = x_i^{ini} + g_i \tag{4.5}$$

The probability density function of the geometric distribution is given by Eqn. (4.6):

$$P(g) = \frac{1}{\varphi+1}\left(1 - \frac{1}{\varphi}\right)^g, g \in \{0,1,2, \dots, +\infty\} \tag{4.6}$$

where g represents a geometrically distributed random integer number and φ corresponds to the average of this specific distribution (Hasançebi, 2007).

It is pointed out by Hasançebi (2007) that most programming language falls short of a library to satisfy a function to sample the geometrically distributed random numbers and suggests using the following equation to generate a geometrically distributed number:

$$g_{i,1}, g_{i,2} = \left[\frac{log(1-r_i)}{log\left(1-1/(1+\varphi_i)\right)}\right] \tag{4.7}$$

where r_i is a uniform random number generated between 0 and 1 for each design variable, and φ can formulated as follows:

$$\varphi = \sqrt{number\ of\ selected\ ready\ section} \tag{4.8}$$

The ready sections from SAP2000 section list library ordered according to their cross-sectional areas and enumerated starting from 1 represent the candidate list for design variables. According to SAP2000 design results integer Id numbers of the assigned sections are assigned to initial values of design variables. Subsequently, for each design variable a random number is generated by using Eqn. (4.7) and it is added or subtracted to the initial value of that design variable. The new random number that represents the ready section id from the list is assigned to the design variable. This process is repeated for all the variables until population size is reached.

4.6 Constraint Handling

In most of the previous structural optimization studies, constraint handling has been achieved using the death penalty method. In this approach, an initial parent population is formed by creating only feasible individuals and all infeasible solutions are automatically eliminated. Although this approach is simple to apply, it has some drawbacks:

i) Firstly, the search process may get stuck at initial stage, since the initial population is randomly generated and there is a high possibility of constraint violation occurrence for every individual for problems subject to heavy constraints.

ii) Secondly, searching through both feasible and infeasible regions is usually more efficient than searching through only feasible regions with death penalty implementation, because the first approach enables reaching the optimum from both regions.

In contrast with the death penalty approach, penalty function implementation prevents the search stagnation and infeasible candidate solutions are not disregarded. Since the abovementioned shortcomings are eliminated and also penalty functions are relatively easier to implement and efficient with a proper parameterization, the use of a penalty function method is preferred in the present study. Subsequently, a constrained objective function is defined to evaluate infeasible individuals in proportion to the sum of the constraint violation as in Eqn. (4.9).

$$\Phi = W[1 + Penalty(a)] = W\left[1 + \alpha\left(\sum_{j=1}^{n_j}(c)\right)\right] \qquad (4.9)$$

In Eqn. (4.9), W symbolizes the unconstrained and Φ symbolizes the constrained objective functions; c refers to the whole set of normalized constraints, and α refers to the penalty coefficient, used to adjust the intensity of penalization as a whole.

4.7 Sample Problem (Two-bar Truss)

4.7.1 General Statement of the Design Variables

The structural optimization problem is the two bar truss problem (Fox 1971, Schmit 1981) as shown in Fig. 4-5. The objective function is the weight minimization of two tubular truss elements which is one of the most used objective functions in structural optimization.

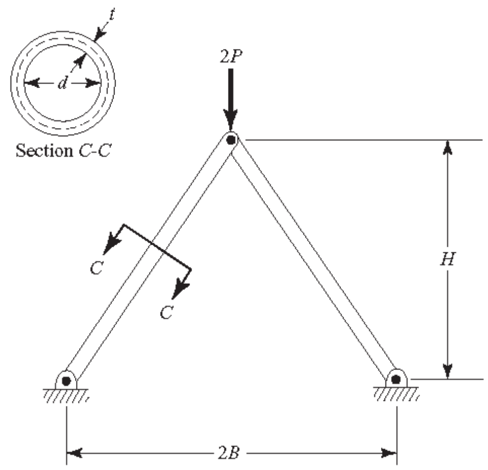


Figure 4-5: The two bar truss problem

The design of the symmetrical truss is specified by a unique set of values for the pre-assigned analysis variables summarized in Table 4-1.

Table 4-1: The pre-assigned analysis variables

Parameter	**Description**	**Unit**	**Value**
$2P$	Applied load	lb	66,000
$2B$	Horiz. dist. btw. supports	in.	60
t	Wall thickness of tube	in.	0.1
E	Young's modulus	psi	30×10^6
ρ	Density	lb/in^3	0.3
σ_y	Yield stress of material	psi	100,000

The bounds of the two independent design variables are shown in Table 4-2. below.

Table 4-2: The bounds of design variables

Design Variables	**Description**	**Lower bound**	**Upper bound**
d	The mean diameter of the tubes	0.1 in.	5 in.
H	The height of the truss	5 in.	50 in.

4.7.2 Derivation and Formulation of the Problem

Using elementary analysis, the dependent parameters are tabulated below:

$$\text{Member force (k): } F = \frac{PL}{H} = P\frac{\left(B^2 + H^2\right)^{1/2}}{H} \qquad (4.11)$$

$$\text{Second moment of inertia (in.}^4\text{): } I = \frac{\pi}{64}\left[\left(d+t\right)^4 - \left(d-t\right)^4\right] = \frac{\pi t d}{8}\left(d^2 + t^2\right) \qquad (4.12)$$

Member stress (psi): $\sigma = \frac{F}{A} = \frac{P}{\pi t}\frac{\left(B^2+H^2\right)^{1/2}}{Hd}$ (4.13)

Buckling stress (psi): $\sigma_e = \frac{\pi^2 EI}{L^2}\frac{1}{A} = \frac{\pi^2 E\left(d^2+t^2\right)}{8\left(B^2+H^2\right)}$ (4.14)

The problem posed is to find d and H such that the weight of the truss system

$$W = 2\rho\pi dt\left(B^2+H^2\right)^{1/2} \rightarrow Min. \qquad (4.15)$$

while requiring that the following imposed behavior constraints be satisfied:

$$g_1(d,H) = \sigma_y - \sigma = \sigma_y - \frac{P}{\pi t}\frac{\left(B^2+H^2\right)^{1/2}}{Hd} \geq 0 \qquad (4.16)$$

$$g_2(d,H) = \sigma_e - \sigma = \frac{\pi^2 EI}{L^2}\frac{1}{A} = \frac{\pi^2 E\left(d^2+t^2\right)}{8\left(B^2+H^2\right)} - \frac{P}{\pi t}\frac{\left(B^2+H^2\right)^{1/2}}{Hd} \geq 0 \qquad (4.17)$$

Formally, the objective function f, and the normalized constraints g_1, g_2 can now be written as

$$f(d,H) = 2\rho\pi dt\left(B^2+H^2\right)^{1/2} \qquad (4.18)$$

$$g_1(d,H) = \frac{P}{\sigma_y \pi t}\frac{\left(B^2+H^2\right)^{1/2}}{Hd} - 1 \leq 0 \qquad (4.19)$$

$$g_2(d,H) = \frac{8P\left(B^2+H^2\right)^{1.5}}{\pi^3 EtHd\left(d^2+t^2\right)} - 1 \leq 0 \qquad (4.20)$$

Note that after obtaining a solution (H*, d*), we must ensure that the tubular cross-sections are indeed thin-walled (i.e., d/t »1).

4.7.3 Solution of the Problem

4.7.3.1 Solution of the Problem with ABC Algorithm

The structural optimization problem is solved first by artificial bee colony (ABC) algorithm and then by four different techniques. Afterwards the results are compared.

1. Initially the control parameters of ABC algorithm, population size (the total number of bees in the colony=N) and maximum number of cycles (MNC) are set as N=10 and MNC=50.
2. A random initial bee colony (N different trusses having similar shape but different member cross-sections) is generated using Eqn. (4.1). Here the initial values of the design variables, constrained objective function, the values of normalized design constraints and unconstrained objective function for every bees (each truss) are calculated. A static penalty coefficient r=1 is used to solve the problem. The results of this step are indicated in Table 4-3. In the ABC algorithm, every food source exploited by the bees represents a possible solution to a given optimization problem. The location and amount of the nectar from the flower patch correspond to the design variables and the fitness function (weight of the truss).

Table 4-3: The initial bee colony

Bee No	λ_1	$0.1 \le d \le 5$	λ_2	$5 \le H \le 50$	f(d,H)	g_1(d,H)	g_2(d,H)	F(d,H)
1	0.7210	3.6330	0.8308	42.3841	35.5598	-0.6458	-0.8046	35.5598
2	0.4589	2.3487	0.1514	11.8120	14.2739	0.2208	-0.3796	17.4251
3	0.4104	2.1111	0.0430	6.9359	12.2532	1.2089	0.2667	30.3341
4	0.9792	4.8982	0.2401	15.8042	31.3073	-0.5399	-0.9404	31.3073
5	0.0904	0.5430	0.1781	13.0162	3.3474	3.8599	45.0587	167.0950
6	0.1902	1.0318	0.7609	39.2389	9.6063	0.2815	6.8614	78.2232
7	0.6894	3.4779	0.4040	23.1789	24.8535	-0.5060	-0.8415	24.8535
8	0.8698	4.3622	0.9262	46.6791	45.6250	-0.7138	-0.8749	45.6250
9	0.5304	2.6988	0.5639	30.3733	21.7175	-0.4529	-0.6306	21.7175
10	0.8216	4.1256	0.4199	23.8960	29.8262	-0.5913	-0.9046	29.8262

3. The trusses are sorted by weight from the lightest to the heaviest, then the first half of the trusses (N/2) is selected. The selected trusses are referred as "employed bees", which found the best food sources.

Table 4-4: The employed bees for the two-bar truss

Bee No	λ_1	$0.1 \leq d \leq 5$	λ_2	$5 \leq H \leq 50$	f(d,H)	g_1(d,H)	g_2(d,H)	F(d,H)
2	0.4589	2.3487	0.1514	11.8120	14.2739	0.2208	-0.3796	17.4251
9	0.5304	2.6988	0.5639	30.3733	21.7175	-0.4529	-0.6306	21.7175
7	0.6894	3.4779	0.4040	23.1789	24.8535	-0.5060	-0.8415	24.8535
10	0.8216	4.1256	0.4199	23.8960	29.8262	-0.5913	-0.9046	29.8262
3	0.4104	2.1111	0.0430	6.9359	12.2532	1.2089	0.2667	30.3341

4. At the beginning of first cycle we loop over each truss (food source) (i=1, 2, 3, …,, N/2).

5. The remainder of the bees called "unemployed bees" or "onlooker bees" watch the waggle dance to decide which of the employed bees should be followed. They decide according to the probability proportional to the amount of nectar existing at the food source. So we determine how many solution(s) must be performed on the selected i<u>th</u> truss based on the probability p(i) using Eqn. (4.3) and recruit the onlooker bees.

$$p_1 = \frac{\frac{1}{17.4251}}{\frac{1}{17.4251} + \frac{1}{21.7175} + \frac{1}{24.8535} + \frac{1}{29.8262} + \frac{1}{30.3341}} = 0.2731$$

$$\Rightarrow \textit{number of onlooker bees} = 1.3653 \approx 1$$

$$p_2 = \frac{\frac{1}{21.7175}}{\frac{1}{17.4251}+\frac{1}{21.7175}+\frac{1}{24.8535}+\frac{1}{29.8262}+\frac{1}{30.3341}} = 0.2191$$

$$\Rightarrow number\ of\ onlooker\ bees = 1.0955 \approx 1$$

$$p_3 = \frac{\frac{1}{24.8535}}{\frac{1}{17.4251}+\frac{1}{21.7175}+\frac{1}{24.8535}+\frac{1}{29.8262}+\frac{1}{30.3341}} = 0.1914$$

$$\Rightarrow number\ of\ onlooker\ bees = 0.9572 \approx 1$$

$$p_4 = \frac{\frac{1}{29.8262}}{\frac{1}{17.4251}+\frac{1}{21.7175}+\frac{1}{24.8535}+\frac{1}{29.8262}+\frac{1}{30.3341}} = 0.1595$$

$$\Rightarrow number\ of\ onlooker\ bees = 0.7977 \approx 1$$

$$p_5 = \frac{\frac{1}{30.3341}}{\frac{1}{17.4251}+\frac{1}{21.7175}+\frac{1}{24.8535}+\frac{1}{29.8262}+\frac{1}{30.3341}} = 0.1569$$

$$\Rightarrow number\ of\ onlooker\ bees = 0.7843 \approx 1$$

Accordingly we send 1 employed bee and 1 onlooker bee to each food source.

6. For the first food source the initial values were given in Table 4-4. The new values of the design variables for employed bee and onlooker bee are calculated as

$$^{new}_{1}d = 2.3487 + 0.2853 * (2.3487 - 4.1256) = 6.9359$$

$$^{new}_{1}H = 11.8120 - 0.4145 * (11.8120 - 6.9359) = 9.7910$$

The new value of the constrained objective function is

$$f_{new} = 2 * 0{,}25 * \pi * d * (50^2 + H^2) = 14.2739$$

New values of the normalized constraints g_1, g_2 are

$$g_1(d, H) = 0.8384$$
$$g_2(d, H) = 0.4542$$

Finally the value of the unconstrained objective function is calculated.

$$F(d, H) = 25.1143$$

Table 4-5: The objective function values for the employed bees and onlooker bees for the-two-bar truss

Bee No	Bee ID	d			H			f(d,H)	g_1	g_2	F(d,H)
		k	ϕ1	0.1≤d≤5	k	ϕ2	5≤H≤50				
1 (Former 2)	Old Bee			2.3487			11.8120	14.2739	0.2208	-0.3796	17.4251
	Employed Bee	4	0.2853	1.8416	5	-0.4145	9.7910	10.9548	0.8384	0.4542	25.1143
	Onlooker Bee 1	4	-0.6548	3.5123	2	-0.0102	12.0014	21.3917	-0.1948	-0.8160	21.3917
2 (Former 9)	Old Bee			2.6988			30.3733	21.7175	-0.4529	-0.6306	21.7175
	Employed Bee	1	-0.1835	2.6345	4	-0.8580	24.8155	19.3343	-0.3745	-0.6314	19.3343
	Onlooker Bee 1	3	-0.9972	3.4757	5	0.6160	44.8109	35.3302	-0.6363	-0.7637	35.3302
3 (Former 7)	Old Bee			3.4779			23.1789	24.8535	-0.5060	-0.8415	24.8535
	Employed Bee	5	-0.3372	3.0170	4	0.8077	22.5996	21.3601	-0.4214	-0.7580	21.3601
	Onlooker Bee 1	2	-0.5908	2.9796	1	0.0508	23.7560	21.4923	-0.4321	-0.7472	21.4923
4 (Former 10)	Old Bee			4.1256			23.8960	29.8262	-0.5913	-0.9046	29.8262
	Employed Bee	5	0.3478	4.8262	2	0.1994	23.7126	34.7876	-0.6490	-0.9405	34.7876
	Onlooker Bee 1	3	0.5057	4.6862	1	0.0634	24.6616	34.3046	-0.6470	-0.9345	34.3046
5 (Former 3)	Old Bee			2.1111			6.9359	12.2532	1.2089	0.2667	30.3341
	Employed Bee	4	-0.3369	2.7899	1	0.4426	5.0000	15.9942	1.2902	-0.2656	36.6300
	Onlooker Bee 1	3	-0.4984	2.5626	3	0.2137	5.0000	14.6910	1.4934	-0.0525	36.6300

7. At the second cycle same calculations are performed as at the first cycle.

Table 4-6: The new objective function values for the employed bees and onlooker bees for the-two-bar truss at the second cycle.

Bee No	Bee ID	d			H			f(d,H)	g_1	g_2	F(d,H)
		k	ϕ1	0.1≤d≤5	k	ϕ2	5≤H≤50				
1 (Former 2)	Old Bee			2.3487			11.8120	14.2739	0.2208	-0.3796	17.4251
	Employed Bee	3	0.5924	1.9527	5	0.0856	12.2297	11.9247	0.4250	0.0570	17.6723
	Onlooker Bee 1	5	-0.9121	2.1320	3	-0.0708	12.5759	13.0727	0.2744	-0.2002	16.6600
2 (Former 9)	Old Bee			2.6345			24.8155	19.3343	-0.3745	-0.6314	19.3343
	Employed Bee	4	0.4084	2.0256	1	-0.2201	22.1216	14.2322	-0.1262	-0.2026	14.2322
	Onlooker Bee 1	3	0.9951	2.2539	3	0.7987	26.5853	17.0303	-0.2973	-0.4007	17.0303
3 (Former 7)	Old Bee			3.0170			22.5996	21.3601	-0.4214	-0.7580	21.3601
	Employed Bee	5	-0.4556	2.6043	2	0.1000	22.6474	18.4519	-0.3306	-0.6237	18.4519
	Onlooker Bee 1	2	-0.5604	2.4615	5	-0.4512	15.5314	15.6740	-0.0718	-0.5284	15.6740
4 (Former 10)	Old Bee			4.1256			23.8960	29.8262	-0.5913	-0.9046	29.8262
	Employed Bee	5	-0.4917	3.1351	1	0.6472	31.2221	25.5875	-0.5353	-0.7608	25.5875
	Onlooker Bee 1	5	0.0872	4.3012	1	-0.0057	23.8316	31.0629	-0.6074	-0.9159	31.0629
5 (Former 3)	Old Bee			2.1111			6.9359	12.2532	1.2089	0.2667	30.3341
	Employed Bee	2	-0.5711	2.0623	4	0.3165	5.0000	11.8230	2.0982	0.8163	46.2814
	Onlooker Bee 1	3	0.2045	2.0395	4	0.5225	5.0000	11.6921	2.1329	0.8779	46.8942

8. At 50th cycle final objective function values are obtained as follows:

Table 4-7: The final solution at the 50. cycle for the-two-bar truss

Bee No	λ_1	0.1≤d≤5	λ_2	5≤H≤50	f(d,H)	g_1(d,H)	g_2(d,H)	F(d,H)
1	0.0559	1.8784	-0.5285	20.2369	12.8126	0.0000	0.0000	12.8126
2	-0.8768	1.8784	-0.8000	20.2369	12.8126	0.0000	0.0000	12.8126
3	0.6723	1.8784	-0.6683	20.2369	12.8126	0.0000	0.0000	12.8126
4	-0.1997	1.8784	-0.0729	20.2369	12.8126	0.0000	0.0000	12.8126
5	0.4351	1.8784	0.4523	20.2369	12.8126	0.0000	0.0000	12.8126

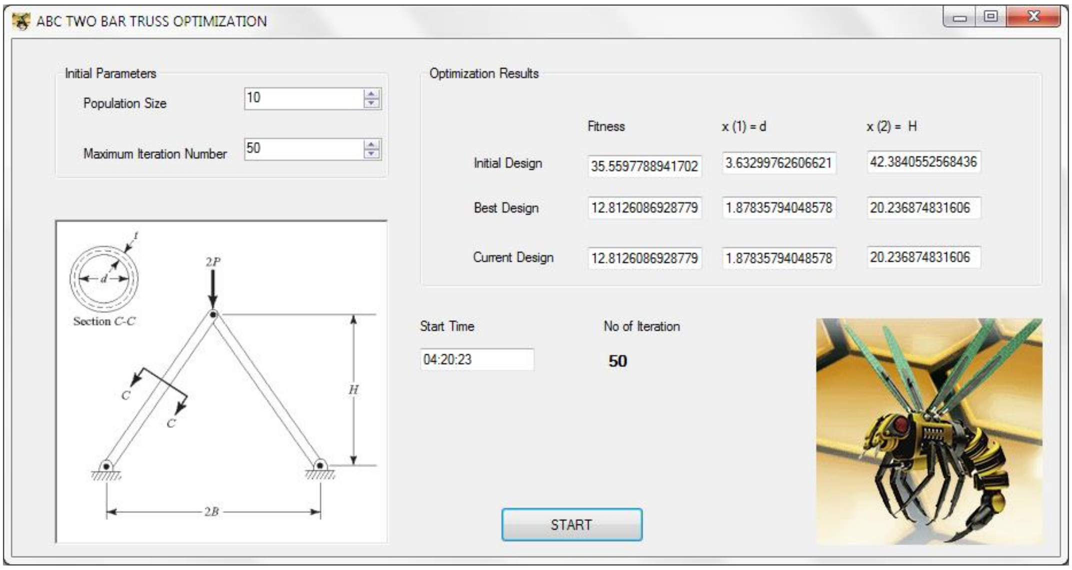


Figure 4-6: ABC Two Bar Optimization Results

4.7.3.2 Solution of the Problem with Augmented Lagrangian Method (ALM)

Powell, Hestenes, and Rockafellar developed the Augmented Lagrangian Method, which is based on combining duality with (exterior) penalty functions. The two bar truss optimization is solved by means of an Excel VBA Code AUGLAG based on Fletcher-Reeves. At the end of iteration 6, the number of function evaluations was 314.

Table 4-8: Results of ALM solution

Iter. No	No Of Func. Ev.	f(d,H)	Violation	d	H	Lagrange Multipliers	
1	1	10.99	10.14	1.00	50.00	0.00	0.00
2	97	12.52	0.05	1.87	19.15	4.62	2.50
3	165	12.81	0.00	1.88	20.20	5.88	2.33
4	237	12.81	0.00	1.88	20.24	5.61	2.40
5	276	12.81	0.00	1.88	20.24	3.07	3.41
6	314	12.81	0.00	1.88	20.24	3.07	3.41
Final Constraint Values:				g1 =	-2.5E-05	g2 =	1E-05

4.7.3.3 Solution of the Problem with Zoutendijk's Method of Feasible Directions

Zoutendijk developed the Method of Feasible Directions, which is still considered as one of the most robust methods for optimization. The method is suitable for solving problems with inequality constraints.

The starting point is x_0 = (5, 50), which is feasible. Table 4-9 presents the iterations. At the end of iteration 4, the number of function evaluations was 51.

Table 4-9: Results of Zoutendijk's Method of Feasible Directions solution

Iter. No	No Of Func. Ev.	f(d,H)	d	H	Active Set
1	0	54.96	5.00	50.00	-
2	17	24.50	2.24	49.80	2
3	34	16.88	2.76	12.34	1
4	51	12.82	1.88	20.28	1,2
Final Constraint Values:		g1 =	-0.001566226	g2 =	-8.999E-05

4.7.3.4 Solution of the Problem with the Generalized Reduced Gradient Method (GRG) (Nonlinear Constraints)

The Generalized Reduced Gradient (GRG) Method is another popular technique for constrained minimization and is well suited to handle nonlinear equality constraints. The inequality constraints are transformed to equality constraints as require, through the addition of slack variables. Excel Solver is based on a GRG algorithm.

Slack variables, x_3 and x_4, have been added to g_1 and g_2, respectively, as is required by the GRG method. The starting point is x_0 = (5, 50, 0.755, 0.91) which is feasible. Table 4-10 presents the iterations. At the end of iteration 11, the number of function evaluations was 20.

Table 4-10: Results of GRG solution

Iter. No	No Of Func. Ev.	f(d,H)	d	H	x3	x4
1	1	54.96	5.00	50.00	0.755	0.91
2	3	30.15	2.75	49.83	0.55	0.46
3	5	25.83	2.36	49.82	0.48	0.14
4	7	24.51	2.24	49.81	0.45	0.00
5	9	14.60	1.91	27.41	0.18	0.00
6	11	13.35	1.88	22.80	0.08	0.00
7	13	13.04	1.88	21.39	0.04	0.00
8	15	12.92	1.88	20.78	0.02	0.00
9	17	12.86	1.88	20.50	0.01	0.00
10	19	12.84	1.88	20.37	0.00	0.00
11	20	12.81	1.88	20.24	0.00	0.00

4.7.3.5 Solution of the Problem with the Sequential Quadratic Programming (SQP) Method

Sequential quadratic programming (SQP) methods have been very popular in recent years due to their superior rate of convergence. The method used for solution of the problem was first published by Pshenichny in 1970 in Russian and later in a book by Pshenichny and Danilin in 1978. SQP is the principal algorithm for NLP in the Matlab “fmincon” optimizer.

The starting point is $x_0 = (0.1, 5.0)$, which is infeasible. Table 4-11 presents the iterations. At the end of iteration 34, the number of function evaluations was 68.

Table 4-11: Results of SQP solution

Iter. No	No Of Func. Ev.	f(d,H)	d	H	Violation
1	0	0.57	0.10	5.00	7983.48
2	2	1.13	0.20	5.10	1646.79
3	4	2.14	0.37	5.29	273.53
5	8	5.62	0.98	5.90	13.67
8	14	9.89	1.71	6.54	1.89
13	24	15.97	2.58	13.29	0.00
17	32	14.33	2.24	15.92	0.00
30	60	12.82	1.88	20.19	0.00
34	68	12.81	1.88	20.23	0.00

4.7.4 Discussion of Solutions of the Problem

A classical benchmark problem is solved with ABC algorithm and with four other classical techniques. ABC has found the optimum solution at 37. iteration.

With ALM method, at the end of iteration 6, the number of function evaluations was 314. With ZMFD method, at the end of iteration 4, the number of function evaluations was 51. With GRG method at the end of iteration 11, the number of function evaluations was 20. With SQP method, at the end of iteration 34, the number of function evaluations was 68. When the five techniques are compared, GRG method was the fast converging method with 20 function evaluations, ABC outperformed well after GRG method. The slowest method was the ALM method with 314 function evaluations. As can be seen from the results ABC has a high convergence speed.

4.8 Structural Optimization Software Development with ABC Algorithm

In this study it is aimed to implement ABC algorithm to discrete size optimization of real size steel truss structures. To find the minimum weight of steel structures by most appropriate cross-sections of structural elements, while satisfying the constraints imposed to the structure, software called ABC-SOP2014 has been developed to evaluate the performance of the ABC algorithm in structural optimization and to give a visual sense to algorithm with a user-friendly interface. The developed software provides minimum weight design of both truss and frame structures, but in this study it is applied only to truss structures. In the present form, the software can handle only size optimization, further developments can be made to it to perform shape and topology optimization also.

ABC-SOP2014 is a size optimization tool capable to interact with commercial structural analysis and design software SAP2000. Although SAP2000 is one of the most used analysis and design software, it does not always find the best design possible. The software is coded on the basis of ABC algorithm on VB.NET platform, which is compatible with the programming language of Open Application Programming Interface (OAPI) released by Computers and Structures, Inc. The OAPI functions have been used to access and communicate with SAP2000 v14. This OAPI provides designers a fast and efficient method to access all of the analysis and design options of SAP2000 enabling model transfer, control of data input and output to and from SAP2000 software environment. The modeling and structural analysis of the structure to be optimized is carried out by SAP2000 software, which has been verified the reliability of its analysis results by various benchmark examples. The integrated ABC- SOP2014 module provides the structural optimization power of ABC algorithm to the designers.

The initial design of SAP2000 can be used to converge rapidly to the optimum point by decreasing the number of iterations, however it is not vitally necessary.

Users of the software must install SAP2000 v14, because the references used in the programming environment were taken from v14. The detailed programming algorithm of ABC-SOP2014 is shown in Fig. 4-7

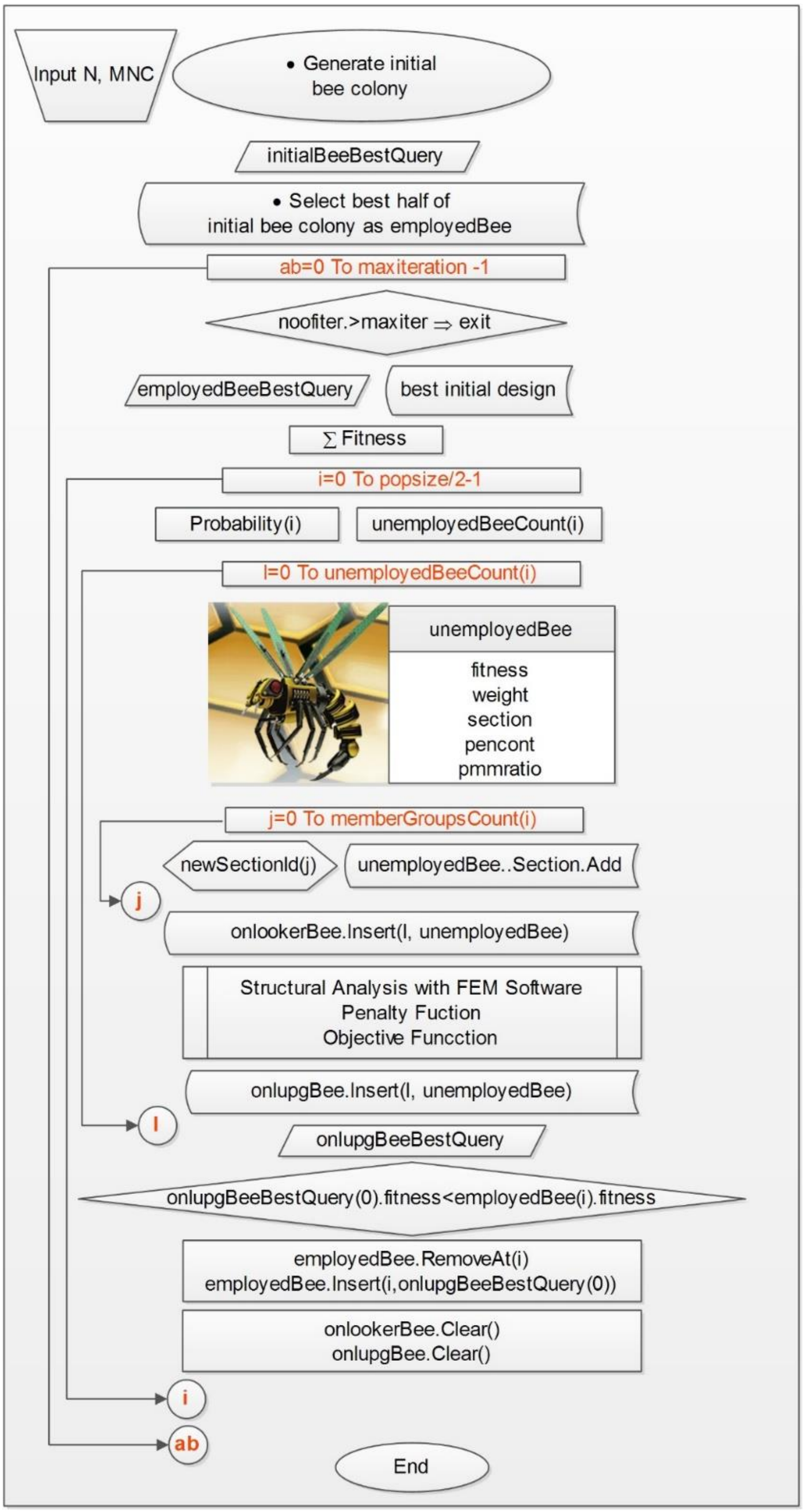


Figure 4-7: The algorithm of ABC-SOP2014

CHAPTER 5

APPLICATION OF ABC ALGORITHM VIA ABC-SOP2014 SOFTWARE

5.1 Introduction

The numerical correctness, efficiency and validation of the ABC algorithm in structural size optimization problems has been investigated and experimented using a test suite consisting of four steel trusses in all. For all the examples presented in this study, a bee colony of bee size N= 50 was set. Maximum number of cycles (MNC) was set different in each problem due to variety of design variables.

The software ABC-SOP2014 discussed in the previous section has been used for performing numerical tests with ABC algorithm.

5.2 Truss Problems

Four different pin-jointed truss examples are solved with ABC-SOP2014. The design constraints in these problems are stress, stability, and displacement type arranged according to AISC-ASD (1989) design code specifications. For all the numerical examples, discrete sets of AISC-ASD ready standard steel sections available in the section database of SAP2000 or discrete sets of problem specific cross-sectional area values are used.

5.2.1 22-Bar Cantilever Truss Structure

The first test problem considered is a statically determinant cantilever truss structure shown in Fig. 5-1, which has been studied by Erbatur et al. (2000) for discrete design variables.

The objective function is to minimize the mass of the structure under stress AISC stress constraints imposed on all bars, for which circular hollow pipe sections from AISC-ASD design specification are adapted. No member grouping is used to make the problem harder to optimize. Besides, the member forces are independent of member sizes due to statical determinacy of the truss, which means no compensation with the distribution of internal forces amongst the members is possible in case any design variable is exceeded.

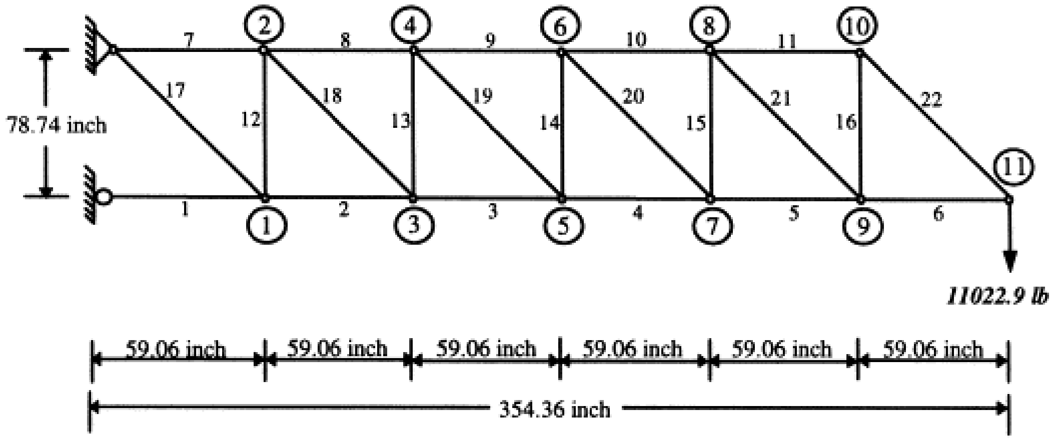


Figure 5-1: Statically determinate 22-bar plane truss

Sizing optimization of the 22-bar plane truss structure is carried out using ABC algorithm setting the initial bee colony size as N=50. Maximum number of cycles (MNC) was set to 1200, but we terminated program at cycle 810. The structural steel members are selected from a database of 37 circular hollow sections issued in AISC-ASD (1989) design specification. All of the design variables were set the same initial cross-sections as PXX8 for the initial design. The results obtained are compared to the previous work done from the literature (Table 5-1). The result of the ABC algorithm used in this study yields a weight of 524.5 lb and a volume of 1849.90 in.^{3}, which is exactly the optimum solution of the problem. The GAOS level2 has a weight of 524.5 lb, which is 4.49% heavier than the result of ABC algorithm. No kind of constraint violation occurred as seen in Figs. 5-2 and 5-3. The screenshot of the optimization results of ABC-SOP2014 program is shown in Fig. 5-4 and the convergence history of optimization process is shown in Fig. 5-5.

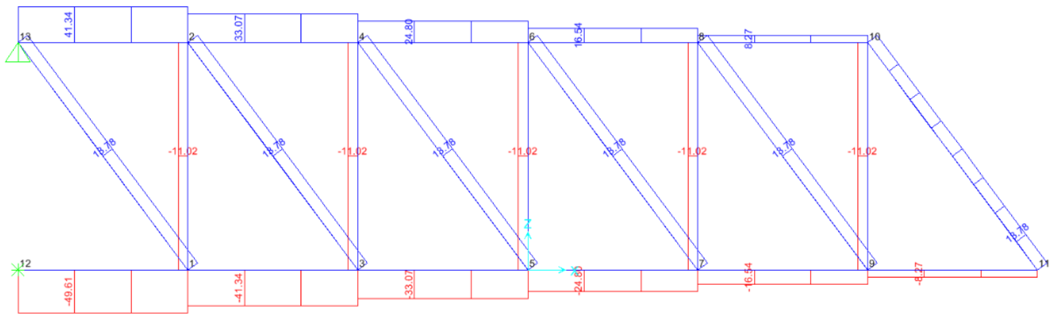


Figure 5-2: Axial forces on elements of 22-bar planar cantilever truss

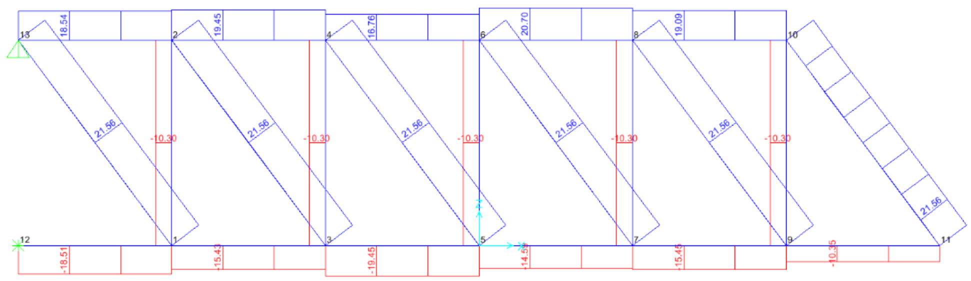


Figure 5-3: Element stresses within limitations of 22-bar planar cantilever truss

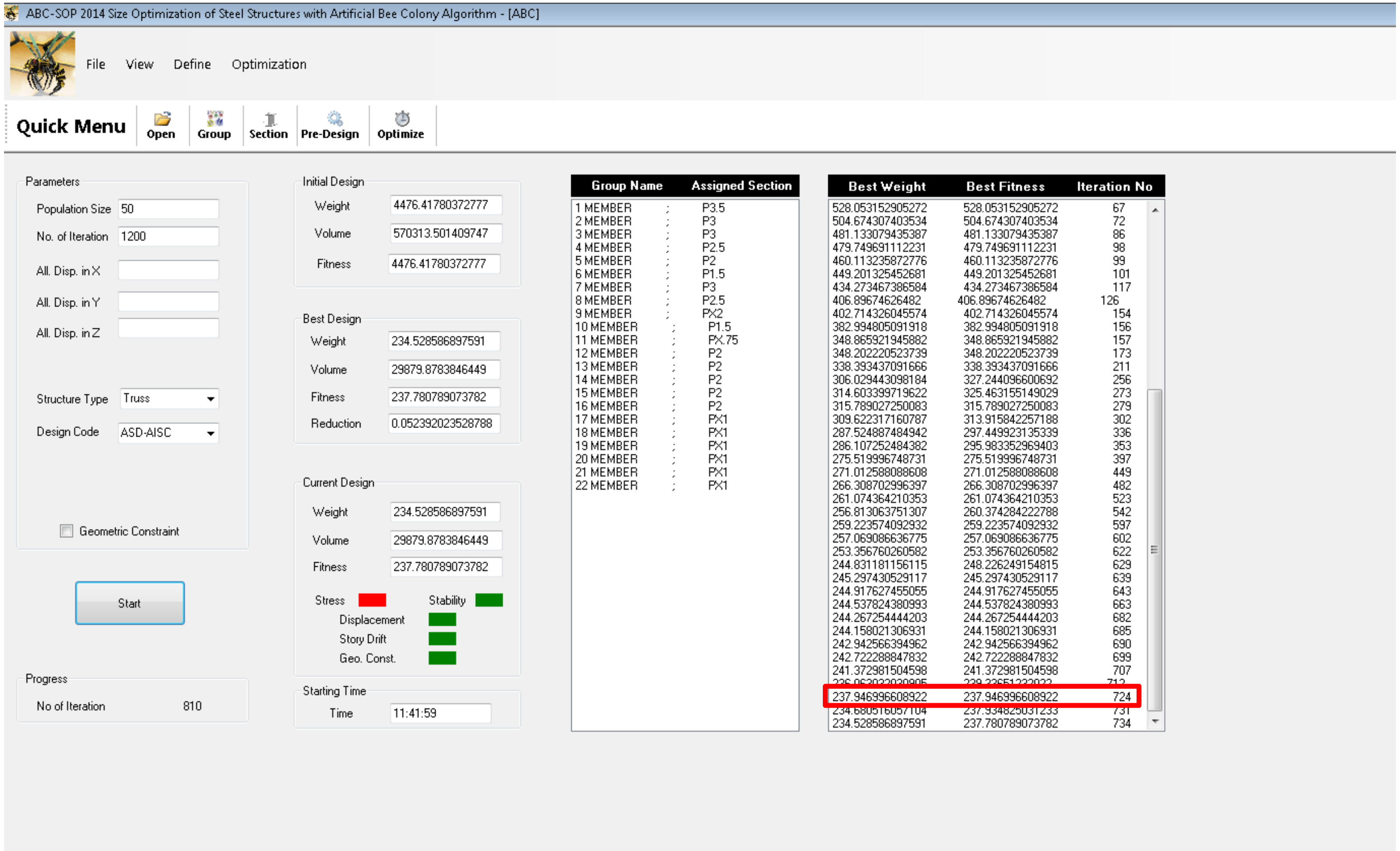


Figure 5-4: ABC-SOP2014 optimization results of 22-bar planar cantilever truss

Table 5-1: Comparison of the optimum designs for 22-bar planar truss.

Truss member	Axial force (kip) SAP2000	Applied stress (ksi)	Allowable stress (ksi)	True optimum Profile	GAOS level2 Profile	This study ABC Profile
1	-49.607	-18.51	- 18.85	P3.5	P3.5	P3.5
2	-41.339	- 15.43	- 18.85	P3.5	P3.5	P3.5
3	-33.071	-14.83	-18.26	P3	P3	P3
4	-24.803	-14.59	-17.2	P2.5	P2.5	P2.5
5	-16.535	-15.45	-15.89	P2	P2	P2
6	-8.268	-10.35	-13.62	P1.5	P1.5	P1.5
7	41.339	18.54	21.6	P3	P3	P3
8	33.071	19.45	21.6	P2.5	P2.5	P2.5
9	24.804	16.76	21.6	PX2	P2.5	PX2
10	16.536	20.69	21.6	P1.5	P1.5	P1.5
11	8.268	19.09	21.6	PX.75	P1.25	PX.75
12	-11.023	-10.3	-12.97	P2	P2	P2
13	-11.023	-10.3	-12.97	P2	P2	P2
14	-11.023	-10.3	-12.97	P2	P2	P2
15	-11.023	-10.3	-12.97	P2	P2	P2
16	-11.023	-10.3	-512.97	P2	P2	P2
17	13.779	21.56	21.6	PX1	PX1	PX1
18	13.779	21.56	21.6	PX1	P 11/2	PX1
19	13.779	21.56	21.6	PX1	PX1	PX1
20	13.779	21.56	21.6	PX1	PX1.25	PX1
21	13.779	21.56	21.6	PX1	PX1	PX1
22	13.779	21.56	21.6	PX1	P1.5	PX1
Weight, lb (kg)				524.5 (237.95)	548.06 (248.64)	524.5 (237.95)
Volume, in.^{3}(cm.^{3})				1849.9 (30314)	1932.21 (31663)	1849.9 (30314)

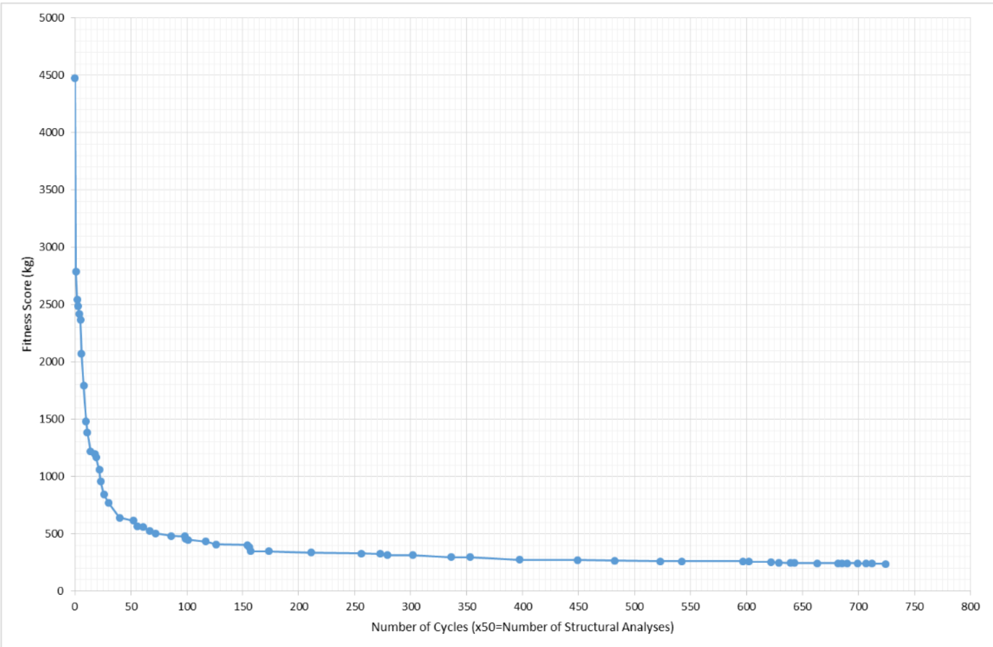


Figure 5-5: ABC-SOP2014 design history of 22-bar planar cantilever truss

5.2.2 25-Bar Space Truss

The second design example considered is a 25-bar space steel truss (Fig. 5-6). This example has been studied by many researchers with different structural optimization techniques: Rajeev and Krishnamoorthy (1992) used GA, Li et al. (2009) utilized HPSO, Camp and Bichon (2004) used ACO, Kripka (2004) used SA and Sonmez (2009) used ABC.

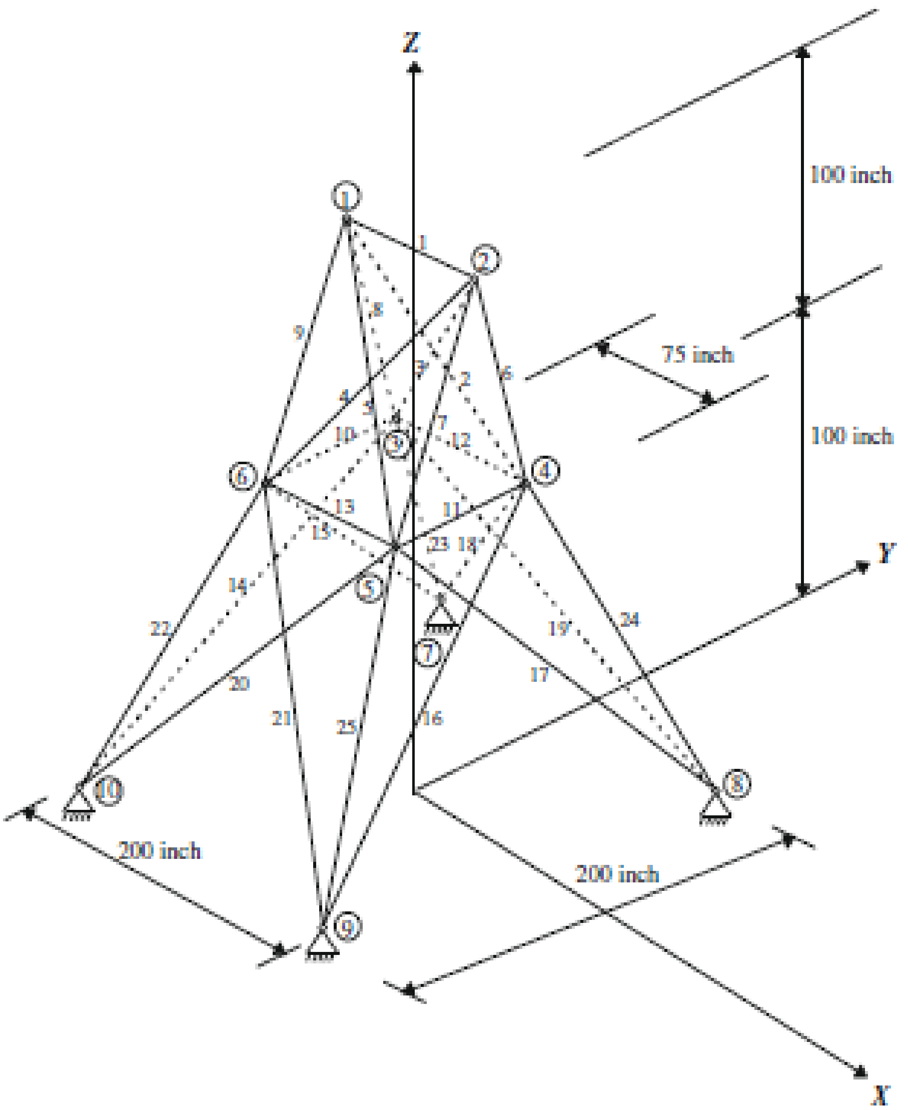


Figure 5-6: 25-bar space truss

All structural members were assumed to be constructed from a material with a mass density of 0.1 lb/ in.^{3} (2,768 kg/m^3) and a modulus of elasticity of 10,000 ksi (68.971 MPa). The stress limitations of the members are ±40 ksi (275.6 MPa). The top nodes 1 and 2 are subjected to displacement limitations of ±0.35 in. in three directions. The structure includes 25 members, which are divided into 8 groups, as follows: (1) A1, (2) A2~A5, (3) A6~A9, (4) A10~A11, (5) A12~A13, (6) A14~A17, (7) A18~A21 and (8) A22~A25. The discrete variables are selected from the set D= {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). A single loading condition was imposed to the structure as shown in Table 5-2.

Table 5-2: Nodal loading conditions (kips) for the 25-bar space truss

Node	**Directions**		
	x	**y**	**z**
1	1.0	-10	-10
2	0	-10	-10
3	0.5	0	0
6	0.6	0	0

The results of ABC algorithm are shown in Table 5-3 and compared with those previously reported in the literature. The best design results obtained by means of all the optimization methods listed in Table 5-3 are identical except of GA. 26946 structural analyses were performed in 539 cycles to find the best feasible design (Fig. 5-7).

Table 5-3: Comparison of the optimum designs for 25-bar space truss

Variables		Optimal cross section area (in.^{2})					
No	**Des. Vars.**	**GA**	**SA**	**HPSO**	**ACO**	**ABC (Sonmez)**	**ABC (This study)**
1	A_1	0.1	0.1	0.1	0.1	0.1	0.1
2	A_2 ~A_5	1.8	0.4	0.3	0.3	0.3	0.3
3	A_6 ~A_9	2.3	3.4	3.4	3.4	3.4	3.4
4	A_{10} ~A_{11}	0.2	0.1	0.1	0.1	0.1	0.1
5	A_{12} ~A_{13}	0.1	2.2	2.1	2.1	2.1	2.1
6	A_{14} ~A_{17}	0.8	1.0	1.0	1.0	1.0	1.0
7	A_{18} ~A_{21}	1.8	0.4	0.5	0.5	0.5	0.5
8	A_{22} ~A_{25}	3.0	3.4	3.4	3.4	3.4	3.4
Weight (lb)		**546.010**	**484.330**	**484.85**	**484.85**	**484.85**	**484.85**
Evaluation (#)		840	40,000	25,000	7,700	24,250	26,946
Constraint violation		None	193.8 x 10"6	None	None	None	None

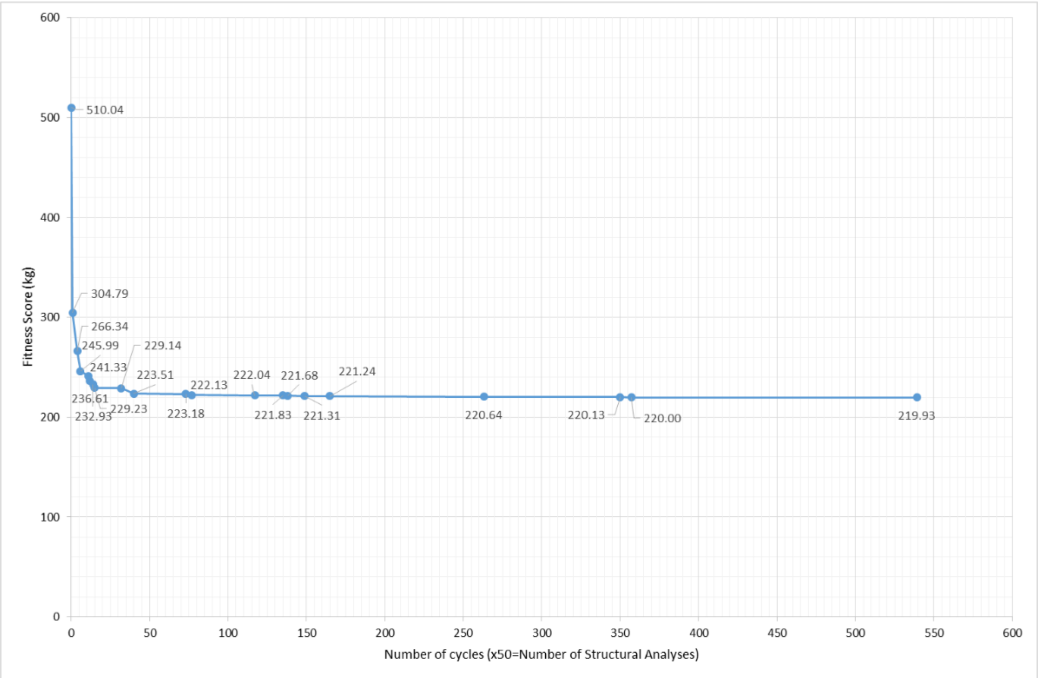


Figure 5-7: ABC-SOP2014 design history of 25-bar space truss

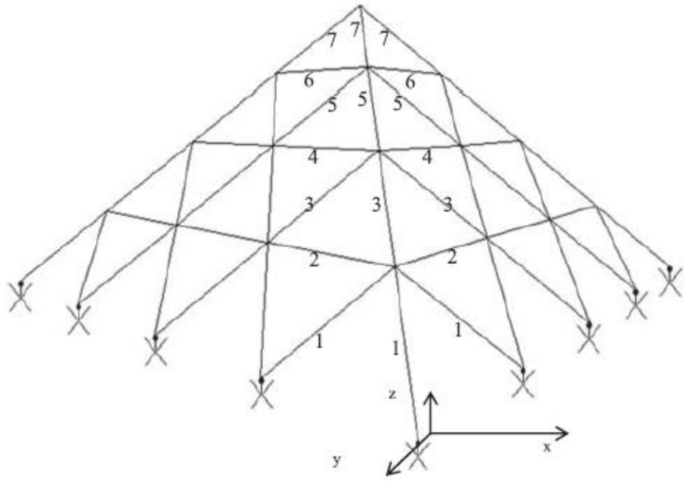
5.2.3 160-Bar Space Truss Pyramid

The third design example considered is a 160-bar space steel pyramid (Fig. 5-8) with a square base diameter of 16 m (52.5 ft) along both the x and y axis and a total height of 8 m (26.25 ft). This problem was studied in Carbas et al. (2013) and Hasançebi and Çarbaş (2011) using the standard (ACO1) and ranked (ACO2) ant colony optimization algorithms and recently in Hasançebi and Azad (2014) using BB-BC and MBB-BC algorithms. The structure contains 55 joints and 160 members that are linked into seven independent sizing design variables. The grouping scheme of members is shown in Fig. 5-8 (a). The structural steel members are selected from a list of 37 hollow pipe sections issued in AISC-ASD (1989) design specification.

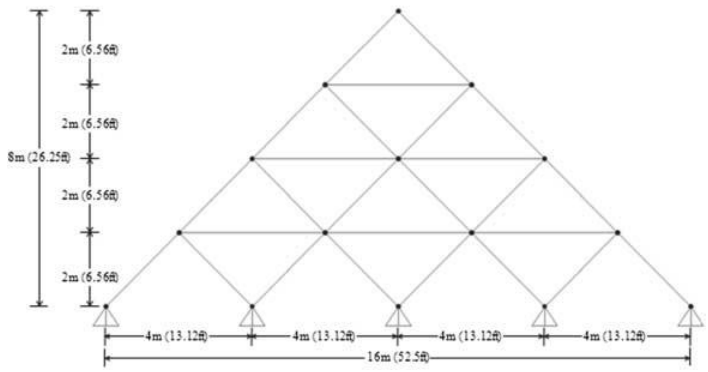
The imposed constraints to the structural elements are stress and stability type, which are limitations of the members are computed according to the provisions of AISC-ASD (1989). The displacements of all nodes are limited to 4.45 cm (1.75 in) in each direction. A vertical load of −8.53 kN (−1.92 kips) applied in the z-direction at all nodes of the pyramid. Consequently, a single load case is considered for design purpose.

Sizing optimization of the 160-bar space pyramid is carried out using ABC algorithm setting the initial bee colony size as N=50. Maximum number of cycles (MNC) was set to 1000. Two runs executed with different initial design points. At first run all of the design variables were set the heaviest circular hollow section (PXX8), at second run a lighter pipe section (P10) was selected for the initial design variables. The results obtained are compared to the previous work of Hasançebi and Çarbaş (2011) and to recent work of Hasançebi and Azad (2014) (Table 5-4).

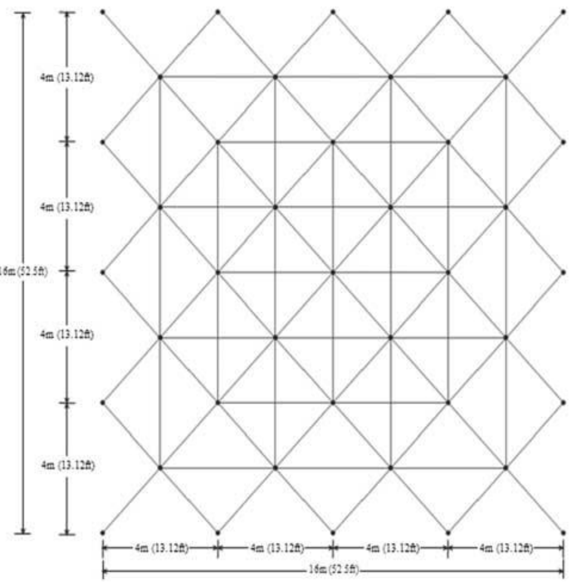
It should be noted that for all the solutions reported in Table 5-4, no kind of constraint violation occurred. Similar to the result of the MBB-BC algorithm ABC algorithm yields an identical design weight of 2788.84 kg (6148.35 lb) after 56 cycles in first run and 61 cycles in second run, which is the best solution to the problem found so far. The final designs attained using BB-BC, ACO1, and ACO2 techniques are slightly heavier; namely 2821.27 kg (6219.83 lb), 2875.01 kg (6338.31 lb) and 2817.56 kg (6211.65 lb), respectively. The screenshot of the optimization results of ABC-SOP2014 program is shown in Fig. 5-9.



(a)



(b)



(c)

Figure 5-8: 160-bar pyramid: (a) 3-dimensional view; (b) front view; (c) plan view (Hasançebi and Çarbaş, 2011).

Table 5-4: Comparison of optimization results for 160-bar space pyramid.

Sizing variables	Optimal cross-sectional areas (in.^{2})				
	ACO2	ACO1	BB-BC	MBB-BC	This study ABC
1	1.07	1.07	1.07	1.07	1.07
2	0.669	0.669	0.669	0.669	0.669
3	1.07	1.07	1.07	1.07	1.07
4	0.669	0.799	0.669	0.669	0.669
5	1.07	1.07	1.07	1.07	1.07
6	0.669	0.669	1.07	0.669	0.669
7	1.48	1.70	1.07	1.07	1.07
Weight, lb (kg)	6211.65 (2817.56)	6338.31 (2875.01)	6219.83 (2821.27)	6148.35 (2788.84)	6148.35 (2788.84)

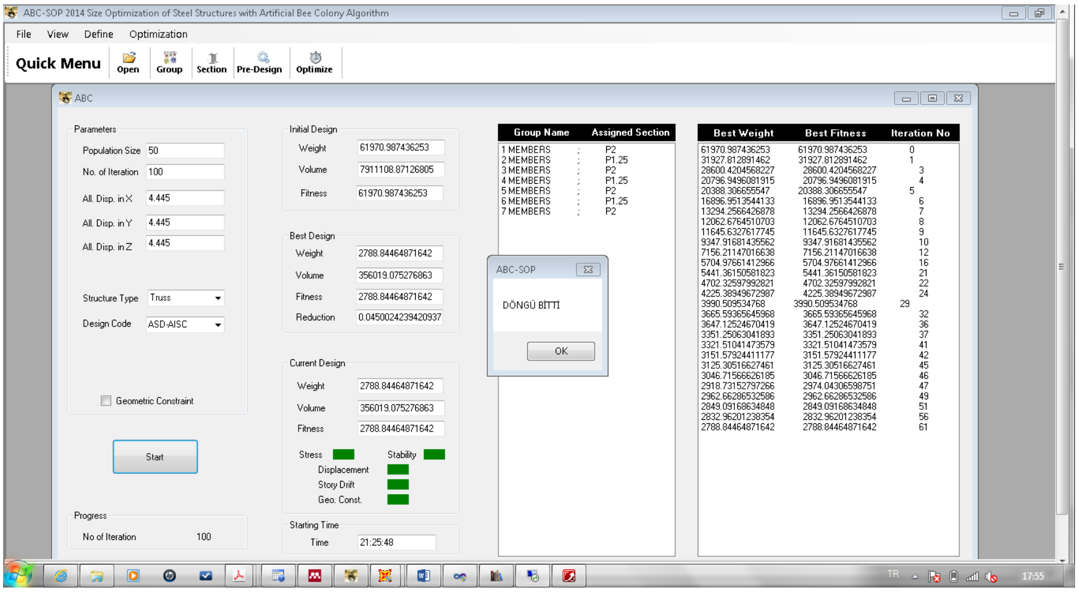


Figure 5-9: ABC-SOP2014 optimization results of 160-bar truss pyramid

The fitness scores of the solutions attained at two independent runs after different cycles are plotted in Fig. 5-10. Although the initial design points were different, identical designs are attained after 61 cycles and 2991 structural analyses for the first run and after 56 cycles 2787 structural analyses for the second run.

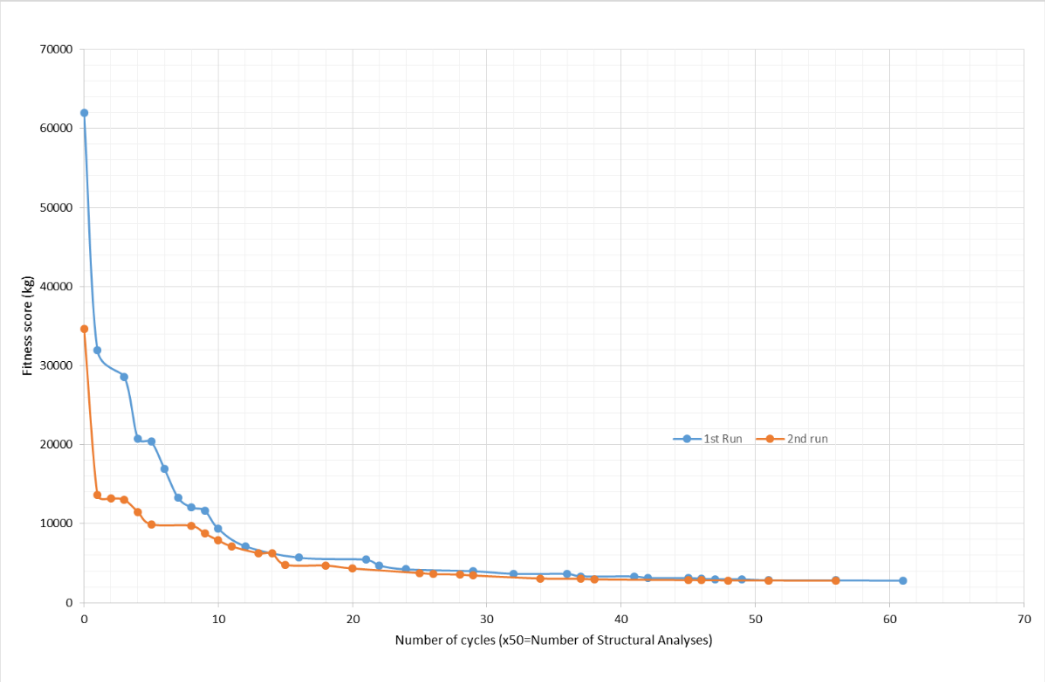


Figure 5-10: ABC-SOP2014 design history of 160-bar space pyramid

5.2.4 693-Bar Braced Barrel Vault

The last example is a three dimensional braced barrel vault structure which was already built for roofing the platform shelters at the Thirumailai Railway Station in Chennai, India as shown in Figs. 5-11 and 5-12.

The braced barrel vault contains 259 joints and 693 bars which are linked into 23 independent size variables considering the symmetry about centerline as shown in

Fig. 5-13. The member grouping scheme is shown in Fig. 5-13 (a) and the front and plan view are provided in Fig. 5-13 (b) and Fig. 5-13 (c), respectively.

Figure 5-11: The platform shelter at Thirumailai Station, LUZ, Chennai, India

The braced barrel vault is subjected to a uniform dead load (DL) pressure of 35 kg/m^2, a positive wind load (WL) pressure of 160 kg/m^2, and a negative wind load (WL) pressure of 240 kg/m^2 which are combined under two separate load cases for design purposes as follows:

(i) 1.5(DL+WL) = 1.5(35 +160) = + 292.5 kg/m^2 (+2.87 kN/m^2)

(ii) 1.5(DL–WL) = 1.5(35–240) = –307.5 kg/m^2 (–3.00 kN/m^2), along z direction.

The displacements of all nodes are limited to a maximum value of ±0.254 cm (0.1 in) in x, y and z directions. The strength and stability requirements of steel members are imposed according to the provisions of AISC-ASD (1989). The structural steel members are selected from a database of 37 circular hollow sections issued in AISC-ASD (1989) design specification.

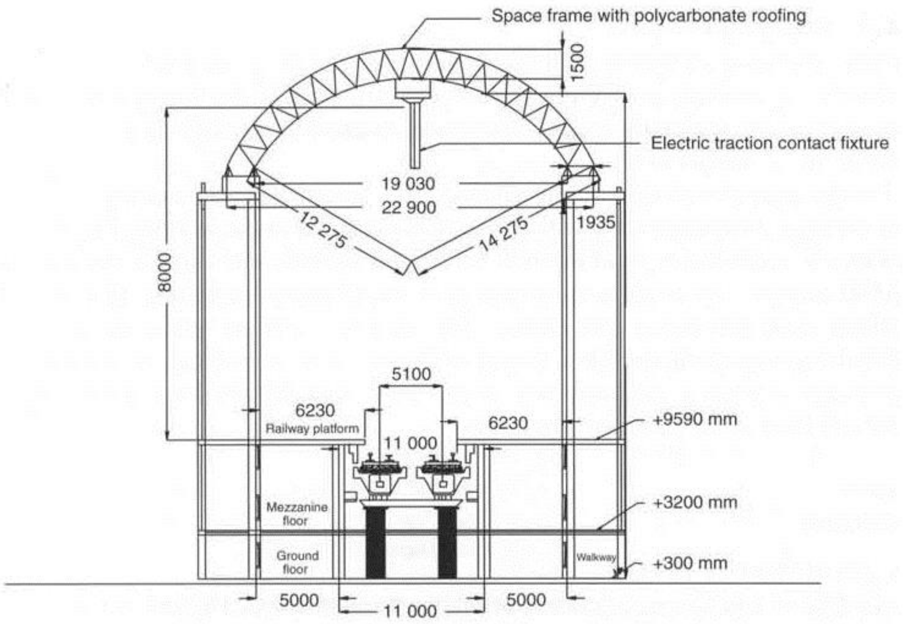


Figure 5-12: The cross-section of the parallel vault and railway tracks

Standart auto design procedure of SAP2000 guided to a feasible design weight of 15691.35 kg (34593.5 lb) that is far from the optimum. The real weight of the structure was 8250 kg (18188.10 lb). On the other hand, an optimum design weight of 5001.8 kg (11027.1 lb) is achieved by the ABC algorithm without any constraint violation. This best design is tabulated in Table 5-5 with section designations assigned to each member group and the convergence history of the algorithm is plotted in Fig. 5-14.

The optimal design of the 693-bar braced barrel vault was first presented by Hasançebi and Çarbaş (2011) using a standart ant colony optimization (ACO1) and ranked ant colony optimization (ACO2) stating that the minimum weight are found 6068.69 kg (13379.19 lb) and 5503.65 kg (12133.47 lb).

The solutions to this problem obtained with ABC, ACO1, and ACO2 are tabulated in Table 5-5. ABC result takes the first place when it is compared to the results of ACO2 and ACO1 with a 5001.8 kg weight, 9.12% lighter than ACO2 and 17.58% lighter than ACO1. ABC result is 68.12% lighter than SAP2000 auto design procedure weight.

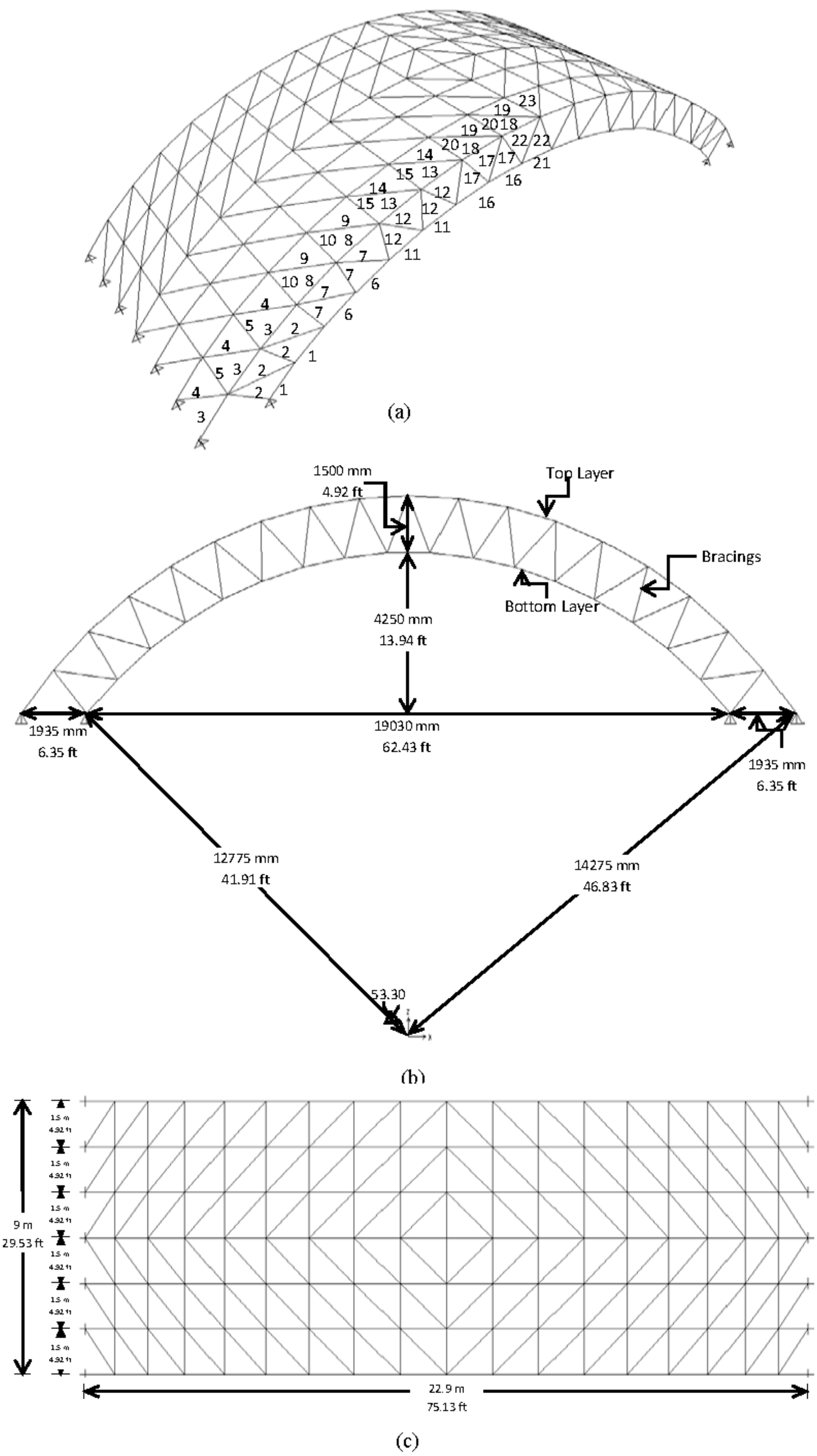


Figure 5-13: The 693-bar braced vault, a) 3-D view, b) Front view, c) Plan view (Hasançebi and Çarbaş, 2011).

Table 5-5: Comparison of ABC with other optimization methods for 693-bar braced barrel vault.

Size variables	ACO2		ACO1		ABC	
	Ready Section	Area, cm^2 (in^2)	Ready Section	Area, cm^2 (in^2)	Ready Section	Area, cm^2 (in^2)
1	P4	20.45 (3.17)	PX3	3.02 (19.48)	PX3	3.02 (19.48)
2	P1	3.18 (0.494)	PX1.5	6.90 (1.07)	P1	3.18 (0.494)
3	P1.25	4.32 (0.669)	P1	3.18 (0.494)	P.75	2.15 (0.333)
4	PX1.25	5.68 (0.881)	PX1.25	5.68 (0.881)	P1	3.18 (0.494)
5	P. 75	2.15 (0.333)	P1.25	4.32 (0.669)	P.75	2.15 (0.333)
6	P5	27.74 (4.3)	PX4	28.45 (4.41)	PXX2.5	25.99 (4.03)
7	P1	4.32 (0.669)	P1.25	4.32 (0.669)	PX1	4.12 (0.669)
8	PX1.25	5.68 (0.881)	PX1.5	6.90 (1.07)	P1	3.18 (0.494)
9	PX3.5	23.74 (3.68)	PXX2	17.16 (2.66)	P1	3.18 (0.494)
10	P1	4.32 (0.669)	PX1.25	5.68 (0.881)	P.75	2.15 (0.333)
11	P1.25	2.79 (0.433)	P1	3.18 (0.494)	P3	14.39 (2.23)
12	P1.5	5.16 (0.799)	PX1	4.12 (0.669)	P2	6.90 (1.07)
13	P1.5	5.16 (0.799)	PX1.25	5.68 (0.881)	P2	6.90 (1.07)
14	P1	4.32 (0.669)	PX2	9.55 (1.48)	P1	3.18 (0.494)
15	PX.75	2.79 (0.433)	P.75	2.15 (0.333)	PX.75	2.79 (0.433)
16	P1.5	5.16 (0.799)	P1.5	5.16 (0.799)	P1.25	4.32 (0.669)
17	PX2	9.55 (1.48)	P2.5	10.97 (1.70)	PX1	4.12 (0.669)
18	P1.25	4.32 (0.669)	P1.25	4.32 (0.669)	PXX2	17.16 (2.66)
19	P1	4.32 (0.669)	P1.5	5.16 (0.799)	P1	3.18 (0.494)
20	P.75	2.15 (0.333)	PX1.5	6.90 (1.07)	P.75	2.15 (0.333)
21	PX2.5	14.52 (2.25)	P4	20.45 (3.17)	P1	3.18 (0.494)
22	P1.5	5.16 (0.799)	P1	3.18 (0.494)	P.75	2.15 (0.333)
23	P.75	2.15 (0.333)	PX.75	2.79 (0.433)	P.75	2.15 (0.333)
Weight	**5503.65 kg (12133.47 lb)**		**6068.69 kg (13379.19 lb)**		**5001.8 kg (11027.1 lb)**	

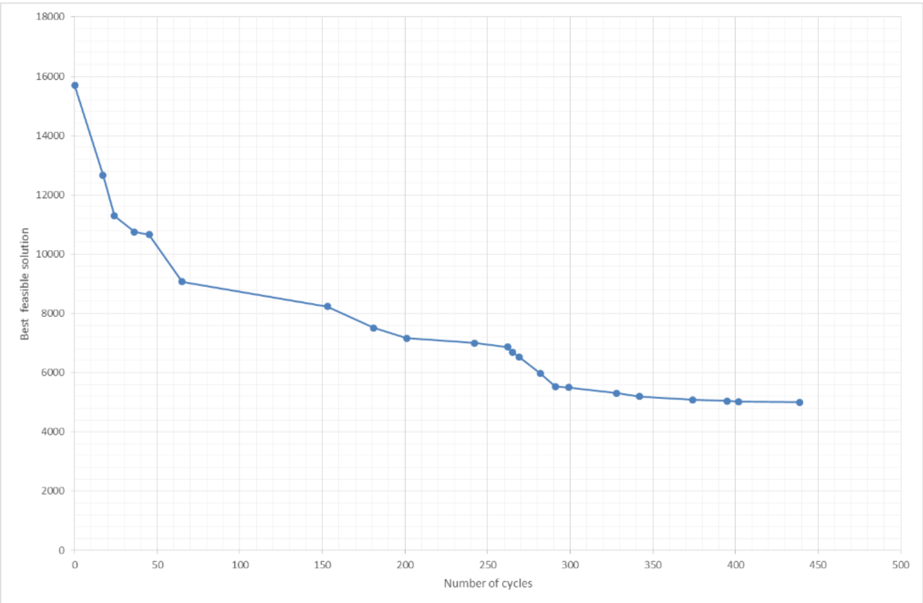


Figure 5-14: ABC-SOP2014 design history of 693 bar braced vault

CHAPTER 6

CONCLUSIONS

6.1 Conclusion

The objective of this thesis is to investigate the use and efficiency of artificial bee colony (ABC) algorithm in structural optimization. The used version of ABC algorithm is a modified version of the original algorithm. Diversification generation method has been used to initialize the population. A software called ABC-SOP2014 has been developed to evaluate the performance of the ABC algorithm in structural optimization on real size steel truss structures and to give a visual sense to algorithm with a user-friendly interface. The software is capable to outperform size optimization program that is interacting with a commercial structural analysis and design software SAP2000 by evaluating the analysis data to find the optimum for the minimum weight design of truss structures.

ABC-SOP2014 offers a practical optimization tool to the ABC algorithm and to the designer.

Most of the studies about structural optimization with ABC algorithm have been used continuous design variables, whereas in this thesis three discrete examples from literature are covered. This thesis contributes to extend the discrete structural sizing optimization field of ABC algorithm providing satisfactory performance with feasible and near optimal solutions compared to previously work done.

Besides discrete sizing optimization of steel structures, there are also some other useful features of the developed program such as:

- It requires a small amount of input due to nature of ABC algorithm selected to optimization. Only colony size and maximum number of cycles constitute the algorithmic input parameters. Other input data, namely SAP2000 file of the model, group data, section list, penalty coefficient α and some design parameters can be imposed easily.

- Material properties can be modified by ABC-SOP2014.
- The grouping property of the software decreases the fabrication requirements and increases the convergence speed.
- It enables the user to create his section lists from ready steel profile lists of SAP2000 or user defined sections.
- Structural analysis, design and optimization algorithm can be managed simultaneously by ABC-SOP2014.
- The optimization history is kept in a specified array to make comparison and measure its performance.
- Language integrated query (LINQ) is used to perform the queries for the best solutions in ABC algorithm, which is a recent approach of programming technique.

As a conclusion, considering the results of the presented study it can be inferred that ABC algorithm with penalty implementation interactively working with SAP2000 is a reliable and efficient discrete sizing optimization technique under the constraints problem specific or imposed by design code provisions with the objective of weight minimization. This functional optimization tool provides optimized designs saving material, construction time and reducing the cost of the structure significantly.

6.2 Final Recommendations

Artificial bee colony algorithm is a nature-inspired heuristic search technique, which refers to experience-based techniques for problem solving, learning, and discovery that give a solution which is not guaranteed to be optimal. To obtain satisfactory results considering the random nature of the search algorithm, the initial parameters should be carefully selected according to the number of design variables to be optimized. Besides proper population size selection, minimizing the number of design variables and the size of discrete section pool is very important to reduce the computational cost. It has been observed that computation time of the structural systems with lesser number of size variables and reduced discrete sets is significantly shorter.

The optimization process via ABC algorithm interacting with SAP2000 takes more time than expected, because of huge amount of data transfer. The structural analysis performed for the initial population is equal to population size and after selecting the best half of the colony, at every iteration cycle half the structural analysis number is the same as population size. Using parallel computing and utilizing a local search algorithm may help to come up with this drawback.

An initial design obtained by SAP2000 with a small fitness score provides better performance. The use of results from previous runs also supplies better performance.

Lastly, it should be considered, that this program is not an analysis and design software alone, the results must be checked by experienced and authorized consultants. It can be used also for educational and research purposes to discover and evaluate new and possibly better design options.

6.3 Future Work

The developed software is working simultaneously with SAP2000 software. Each interaction and each analysis consumes time depending on the problem size, hardware performance and also source code optimization performance. To save computational time an integrated finite element module can be added to the software.

Further development can be made with adding new design codes to provide users the opportunity to compare the effect of various constraints imposed by different design code specifications. The algorithm also can be modified or adapted properly for better convergence. A hybridization with other algorithms can be a good example for modification. Some control parameters can be assigned to increase the efficiency of the algorithm.

REFERENCES

AISC-ASD (American Institute of Steel Construction-Allowable Stress Design). (1989). *Manual of steel construction-allowable stress design.*

Akay, B., & Karaboga, D. (2010). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing, 23*(4), 1001–1014.

Aydogdu, I., Akin, A, & Saka, M. P. (2012). Optimum design of steel space srames by artificial bee colony algorithm, *10th International Congress on Advances in Civil Engineering,* Ankara, Turkey.

Balasubramani, K., & Marcus, K. (2013). A comprehensive review of artificial bee colony algorithm. *International Journal Of Computers&Technology, 5*(1), 15–28.

Bansal, J.C., Sharma, H. and Jadon, S.S. (2013). Artificial beecolony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms* 5(1-2) 123-159.

Baykasoglu, A., Özbakır, L., & Tapkan, P. (2007). Artificial bee colony algorithm and its application to generalized assignment problem. *Swarm Intelligence: Focus on Ant and particle swarm optimization*, 113-144.

Bendsoe, M. P., & Sigmund, O. (2003). *Topology Optimization: Theory, Methods and Applications* (p. 370). Springer.

Bolaji, A. L. A., Khader, A. T., Al-Betar, M. A., & Awadallah, M. A. (2013). Artificial bee colony algorithm, its variants and applications: a survey. *Journal of Theoretical & Applied Information Technology,* 47(2), 434-459

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence* (pp. 32-77). Oxford University Press, NY, USA

Carbas, S., Aydogdu, I., & Saka, M. P. (2013). A comparative study of three metaheuristics for optimum design of engineering structures. In *Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization, Orlando, Florida, USA* (pp. 1-10).

Christensen, P. W., & Klarbring, A. (2008). *An Introduction to Structural Optimization*.

Coello Coello, C. a. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, *191*(11-12), 1245–1287.

Computers and Structures, Inc. (2008). *SAP2000 API Documentation*, California.

Degertekin, S. O. (2012). Optimum design of geometrically non-linear steel frames using artificial bee colony algorithm. *Steel & Composite Structures*, *12*(6), 505–522.

Degertekin, S.O. & Hayalioglu, M. S. (2013). An improved artificial bee colony algorithm for sizing optimization of space truss structures, *Proceedings of the Fourth International Conference on Mathematical and Computational Applications,* Manisa, Turkey (pp. 1–10).

Erbatur, F., Hasançebi, O., Tütüncü, I., & Kılıç, H. (2000). Optimal design of planar and space structures with genetic algorithms. *Computers & Structures*.

Fiouz, A.R., Obeydi, M., Forouzani, H., & Keshavarz, A. (2012). Discrete optimization of trusses using an artificial bee colony (ABC) algorithm and the fly-back mechanism. *Structural Engineering and Mechanics*, *44*(4), 501–519.

Fox, R. L. (1971). *Optimization methods for engineering design* (p. 270). Addison-Wesley Pub. Co.

Galilei, G. (1991). *Dialogues Concerning Two New Sciences*, translated by Henry Crew and Alfondo de Salvio, William Andrew Publishing.

Gandomi, A. H., Yang, X.-S., Talatahari, S., & Alavi, A. H. (2013). *Metaheuristic Applications in Structures and Infrastructures* (p. 577). Elsevier.

Gerhardt, E., & Gomes, H. M. (2012). Artificial bee colony (ABC) algorithm for engineering optimization problems. *EngOpt 2012 – 3rd International Conference on Engineering Optimization,* Rio de Janeiro, Brazil

Hadidi, A., Azad, S. K., & Azad, S. K. (2010). Structural optimization using artificial bee colony algorithm.

Haftka, R. T., & Gürdal, Z. (1992). *Elements of Structural Optimization* (p. 481). Springer.

Hasançebi, O. (2007). Discrete approaches in evolution strategies based optimum design of steel frames. *Structural Engineering and Mechanics*, 26, 191-210.

Hasançebi, O., & Çarbaş, S. (2011). Ant colony search method in practical structural optimization. *Internatıonal Journal of Optımızatıon In Cıvıl Engıneerıng*, *1*, 91-105.

Hasançebi, O., & Kazemzadeh Azad, S. (2014). Discrete size optimization of steel trusses using a refined big bang–big crunch algorithm. *Engineering Optimization*, *46*(1), 61-83.

Hossain, M. S., & El-shafie, A. (2013). Application of artificial bee colony (ABC) algorithm in search of optimal release of Aswan High Dam. *Journal of Physics: Conference Series*, (Vol. 423, No. 1, p. 012001). IOP Publishing.

Jahjouh, M. M., Arafa, M. H., & Alqedra, M. A. (2013). Artificial Bee Colony (ABC) algorithm in the design optimization of RC continuous beams. *Structural and Multidisciplinary Optimization*, *47*(6), 963–979.

Joubari, M. M., Pashaei, M. H., & Fathi, A. (2013). Artifical bee colony algorithm and gravitation search algorithm for optimization of truss structures with multiple frequency constraints, 3131–3141.

Kang F, Li J, Xu Q (2009a) Hybrid simplex artificial bee colony algorithm and its application in material dynamic parameter back analysis of concrete dams. *J Hydraul Eng* 40(6):736–742

Kang, F., Li, J., & Xu, Q. (2009b). Structural inverse analysis by hybrid simplex artificial bee colony algorithms. *Computers & Structures*, *87*(13), 861–870.

Kang, F., Li, J., & Ma, Z. (2013). An artificial bee colony algorithm for locating the critical slip surface in slope stability analysis. *Engineering Optimization*, *45*(2), 207–223.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*.

Karaboga, D., & Basturk, B. (2007a). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459-471.

Karaboga, D., & Basturk, B. (2007b). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. *Foundations of Fuzzy Logic and Soft Computing* (pp. 789-798). Springer Berlin Heidelberg.

Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, *214*(1), 108-132.

Karaboga, D., & Akay, B. (2011). A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing*, *11*(3), 3021–3031.

Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2012). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*.

Kirsch, U. (1981). *Optimum structural design: concepts, methods, and applications* (p. 441). McGraw-Hill.

Kirsch, U. (1993). *Structural optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg.

La, A., Bolaji, A. R. O., Khader, A. T., Al-betar, M. A., & Awadallah, M. A. (2013). Artificial bee colony algorithm , its variants and applications : A Survey, *47*(2).

Li, H. J., Li, J. J., & Kang, F. (2011a). Artificial bee colony algorithm for reliability analysis of engineering structures. *Advanced Materials Research*, *163-167*, 3103–3109.

Li, H., Li, J., & Kang, F. (2011b). Risk analysis of dam based on artificial bee colony algorithm with fuzzy c -means clustering. *Canadian Journal of Civil Engineering*, *38*(5), 483–492.

Mandal, S. K., Chan, F. T. S., & Tiwari, M. K. (2012). Leak detection of pipeline: An integrated approach of rough set theory and artificial bee colony trained SVM. *Expert Systems with Applications*, *39*(3), 3071–3080.

Maute, K. K. C. (1998). *Topologie- und Formoptimierung von dünnwandigen Tragwerken* (p. 253). Univ., Inst. für Baustatik.

Miguel, L. F. F., & Miguel, L. F. F. (2013). Assessment of modern metaheuristic algorithms-HS, ABC and FA-in shape and size optimisation of structures with different types of constraints. *International Journal of Metaheuristics*, *2*(3), 256-293.

Ozturk, H. T., Durmus, A. (2013). Optimum cost design of RC columns using artificial bee colony algorithm. *Structural Engineering and Mechanics*, 45(5) 643-654

Prakash, D. (2012). Bespoke artificial bee colony algorithm to determine the earthquake locations, *2*(3), 207–212.

Saka, M. (1990). Optimum design of pin-jointed steel structures with practical applications. *Journal of Structural Engineering*, *116*(10), 2599–2620.

Schmit, L. A. (1981). Structural synthesis - Its genesis and development. *AIAA Journal*, *19*(10), 1249–1263.

Schumacher, A. (2013). *Optimierung mechanischer Strukturen* (2nd ed.). Springer Vieweg.

Seeley, T. D., Camazine, S., & Sneyd, J. (1991). Collective decision-making in honey bees: how colonies choose among nectar sources. *Behavioral Ecology and Sociobiology*, *28*(4), 277-290.

Seeley, T. D. (1995). *The wisdom of the hive: the social physiology of honey bee colonies*. Harvard University Press.

Seeley, T. D. (2010). *Honeybee democracy*. Princeton University Press.

Sigmund, O. (2000). Topology optimization : a tool for the tailoring of structures and materials. *Phil. Trans. R. Soc. Lond. A*, *358*, 211–227.

Singh, A. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing*, *9*(2), 625–631.

Sonmez, M. (2011a). Artificial Bee Colony algorithm for optimization of truss structures. *Applied Soft Computing*, *11*(2), 2406–2418.

Sonmez, M. (2011b). Discrete optimum design of truss structures using artificial bee colony algorithm. *Structural and Multidisciplinary Optimization*, *43*(1), 85–97.

Su, G. S., Qian, K., & Zhang, Y. (2012). Load distribution calculation of pile group using artificial bee colony algorithm. *Applied Mechanics and Materials*, *204-208*, 4878–4883.

Sun, H., Luş, H., & Betti, R. (2013). Identification of structural models using a modified Artificial Bee Colony algorithm. *Computers & Structures*, *116*, 59–74.

Talatahari, S., Nouri, M., & Tadbiri, F. (2012). Optimization of skeletal structural using artificial bee colony algorithm, *Int. J. Optim. Civil Eng*, *2*(4), 557–571.

Yahya, M., & Saka, M. P. (2014). Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights. *Automation in Construction*, *38*, 14–29.