

COMPUTATIONALLY ENHANCED TECHNIQUES FOR PRACTICAL  
OPTIMUM DESIGN OF STEEL STRUCTURES

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

BY

SAEID KAZEMZADEH AZAD

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
CIVIL ENGINEERING

APRIL 2014



Approval of the thesis:

**COMPUTATIONALLY ENHANCED TECHNIQUES FOR PRACTICAL  
OPTIMUM DESIGN OF STEEL STRUCTURES**

submitted by **SAEID KAZEMZADEH AZAD** in partial fulfillment of the requirements  
for the degree of **Doctor of Philosophy 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 Yalçiner  
Head of Department, **Civil Engineering**

\_\_\_\_\_

Assoc. Prof. Dr. Oğuzhan Hasaebi  
Supervisor, **Civil Engineering Dept., METU**

\_\_\_\_\_

**Examining Committee Members:**

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

\_\_\_\_\_

Assoc. Prof. Dr. Oğuzhan Hasaebi  
Civil Engineering Dept., METU

\_\_\_\_\_

Assoc. Prof. Dr. Murat Altuğ Erberik  
Civil Engineering Dept., METU

\_\_\_\_\_

Assoc. Prof. Dr. Ahmet Türer  
Civil Engineering Dept., METU

\_\_\_\_\_

Asst. Prof. Dr. İzzet Özdemir  
Manufacturing Engineering Dept., Atılım University

\_\_\_\_\_

Date: 11.04.2014

**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: Saeid Kazemzadeh Azad

Signature:

## **ABSTRACT**

### **COMPUTATIONALLY ENHANCED TECHNIQUES FOR PRACTICAL OPTIMUM DESIGN OF STEEL STRUCTURES**

Kazemzadeh Azad, Saeid

Ph. D., Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. Oğuzhan Hasançebi

April 2014, 201 pages

Practical optimum design of structural systems via modern metaheuristic algorithms suffers from enormously time-consuming structural analyses to locate a reasonable design. This study is an attempt to reduce the computational effort of optimization process involved in real-life applications through development of alternative techniques to the existing computationally expensive methods. Basically two main approaches are considered as (i) investigating the algorithmic structure of the existing metaheuristics and enhancing their performances in sizing optimization problems (ii) developing new design-driven optimization techniques based on the principles of structural mechanics.

In the first approach enhanced reformulations of modern metaheuristics are developed and tested using real-life instances. Furthermore, an upper bound strategy is proposed wherein non-improving candidate designs are identified and excluded from the structural analysis stage, diminishing the total computational effort.

In the second approach a guided stochastic search (GSS) technique is developed wherein the search direction is determined by the principle of virtual work and response computations of the generated designs. In the GSS, the information provided in the structural analysis and design check stages are utilized for handling strength constraints. Moreover, the principle of virtual work is used to detect the most effective structural members for satisfying displacement constraints. The optimum sizing of a structure is then performed where both strength and displacement criteria are taken into account for reduction of the member sizes along the way the aforementioned constraints are handled.

The numerical results indicate the computational efficiency of the proposed techniques in sizing optimization of steel skeletal structures.

**Keywords:** Structural Optimization, Discrete Sizing, Truss Structures, Frame Structures, Steel Design

## ÖZ

### GELİŞTİRİLMİŞ HESAPSAL YÖNTEMLERLE ÇELİK YAPILARIN UYGULANABİLİR OPTİMUM TASARIMLARI

Kazemzadeh Azad, Saeid

Doktora, İnşaat Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Oğuzhan Hasançebi

Nisan 2014, 201 sayfa

Modern metabulğusal yöntemlerle yapı sistemlerinin optimum tasarımları, makul sayılabilecek bir çözümün ancak çok sayıda yapısal analiz gerçekleştirilerek ulaşılabilmesi açısından son derece zaman alıcı bir işlemdir. Bu çalışmada amaç, mevcut pahalı hesaplama yöntemlerine alternatif olarak, gerçek hayattaki yapıların optimum tasarımları için gereken hesap yükünü azaltacak yeni ve alternatif yöntemler geliştirmektir. Bu doğrultuda genel olarak şu iki temel yaklaşım takip edilmektedir: (i) mevcut metabulğusal yöntemlerin algoritmik yapılarının incelenerek bu tekniklerin optimum boyutlandırma problemlerindeki performanslarının geliştirilmesi, (ii) yapı mekaniği prensiplerini kullanarak çalışan tasarım odaklı yeni optimizasyon tekniklerinin geliştirilmesi.

İlk yaklaşımda modern metabulğusal yöntemler yeniden formüle edilip gerçek hayat uygulamaları kullanılarak test edilmiştir. İlaveeten, bu yöntemlerdeki toplam hesap yükünü azaltmak açısından; optimizasyon işlemi boyunca denenen kötü tasarımların tespit edilerek yapı analizi aşamasından hariç tutulmasını sağlayan “üst sınır stratejisi” isimli bir yaklaşım önerilmektedir.

İkinci yaklaşımda ise, arama yönünü virtüel iş prensipi ve optimizasyon boyunca oluşturulan tasarımların tepki hesaplamalarını kullanarak belirleyen GSS isimli bir rehberli stokastik arama tekniği geliştirilmiştir. GSS’de yapısal analiz ve tasarım kontrol aşamaları boyunca elde edilen bilgiler, oluşturulacak tasarımların mukavemet sınırlayıcılarını düzenlemek için kullanılmaktadır. Öte yandan, virtüel iş prensipi deplasman sınırlayıcılarının sağlanması açısından en etkin elemanların belirlenmesi için kullanılmaktadır. Buna göre, bir yapının optimum boyutlandırılması mukavemet ve deplasman sınırlayıcılarının bahsi geçen yöntemler kapsamında ele alınarak yapı eleman kesitlerinin azaltılması suretiyle gerçekleştirilmektedir.

Sayısal sonuçlar çelik iskelet yapılarının optimum boyutlandırma işleminde önerilen tekniklerle elde edilecek hesap verimliliğini ortaya koymaktadır.

Anahtar Kelimeler: Yapısal Optimizasyon, Ayrık Boyutlandırma, Kafes Yapılar, Çerçeve Yapılar, Çelik Tasarım



To My Family

## **ACKNOWLEDGMENTS**

My first debt of gratitude must go to my supervisor Assoc. Prof. Dr. Oğuzhan Hasançebi for his patience, guidance, and continuous encouragement that motivated me at all levels of this research. It was an invaluable opportunity for me to work under his kind and enlightening supervision.

I would like to express my sincere appreciation to Prof. Dr. Mehmet Polat Saka for his helpful and constructive comments on my research work.

I am thankful to my family for their encouragement and unconditional support throughout my Ph. D. studies.

In particular, I would like to thank my brother Sina Kazemzadeh Azad. Indeed it would not have been possible to write this doctoral thesis without his help and support.

## TABLE OF CONTENTS

ABSTRACT .....	v
ÖZ .....	vii
ACKNOWLEDGMENTS .....	x
TABLE OF CONTENTS .....	xi
LIST OF TABLES .....	xiv
LIST OF FIGURES .....	xvi
CHAPTERS	
1. INTRODUCTION .....	1
1.1 Basic Elements of Structural Optimization .....	1
1.2 Structural Optimization Techniques .....	2
1.3 Optimization of Skeletal Structures using Metaheuristics: A Survey of the State-of-the-Art .....	5
1.4 Today's Dilemma in Practical Structural Optimization .....	11
1.5 Aim and Scope of the Thesis .....	13
2. OPTIMUM DESIGN PROBLEM FORMULATION .....	17
2.1 Discrete Sizing Optimization Problem to AISC-ASD .....	17
2.1.1 Design Constraints for Steel Trusses .....	18
2.1.2 Design Constraints for Steel Frames .....	19
2.2 Discrete Sizing Optimization to AISC-LRFD .....	21
2.2.1 Nominal Strengths .....	23
2.2.2 Effective Length Factor K .....	27
2.2.3 Considerations for Steel Trusses .....	28
3. REFORMULATIONS OF BIG BANG-BIG CRUNCH ALGORITHM .....	29
3.1 Introduction .....	29
3.2 Standard BB-BC Algorithm .....	30
3.3 Search Dimensionality and Step Size .....	33
3.4 Numerical Investigations with BB-BC Algorithm .....	34

3.5 Reformulations of BB-BC Algorithm .....	36
3.6 Numerical Examples .....	42
3.6.1 Performance Evaluation of MBB-BC Algorithm .....	42
3.6.1.1 Example 1: 160-Bar Space Pyramid .....	43
3.6.1.2 Example 2: 354-Bar Braced Dome Truss .....	49
3.6.1.3 Example 3: 693-Bar Braced Barrel Vault .....	54
3.6.1.4 Example 4: 960-Bar Double Layer Grid .....	58
3.6.2 Performance Evaluation of EBB-BC Algorithm .....	61
3.6.2.1 Example 1: 132-Member Unbraced Space Steel Structure...	62
3.6.2.2 Example 2: 209-Member Industrial Factory Building .....	67
3.7 Summary .....	74
4. UPPER BOUND STRATEGY .....	75
4.1 Introduction .....	75
4.2 The UBS in Metaheuristic Based Design Optimization .....	76
4.3 Numerical Examples .....	79
4.3.1 Example 1: 135-Member Steel Frame .....	80
4.3.2 Example 2: 1026-Member Steel Frame .....	88
4.3.3 Example 3: 10-Bar Truss Structure .....	97
4.3.4 Example 4: 45-Bar Truss Bridge .....	102
4.3.5 Example 5: 120-Bar Truss Dome .....	105
4.4 Summary .....	109
5. GUIDED STOCHASTIC SEARCH TECHNIQUE FOR DISCRETE SIZING OPTIMIZATION OF STEEL TRUSSES .....	111
5.1 GSS for Single Displacement Constraint and Single Load Case .....	111
5.1.1 Introduction .....	111
5.1.2 Application of the Principle of Virtual Work .....	112
5.1.3 The GSS Algorithm .....	114
5.1.4 Numerical Examples .....	122
5.1.4.1 Example 1: 117-Member Cantilever Truss .....	123
5.1.4.2 Example 2: 130-Member Transmission Tower .....	126
5.1.4.3 Example 3: 392-Member Double Layer Grid .....	129

5.1.4.4	Example 4: 354-Member Truss Dome .....	133
5.1.5	Summary .....	137
5.2	GSS for Multiple Load Cases and Displacement Criteria .....	138
5.2.1	Introduction .....	138
5.2.2	Enhancement of GSS .....	138
5.2.2.1	First Approach (GSS <sub>A</sub> ) .....	139
5.2.2.2	Second Approach (GSS <sub>B</sub> ) .....	139
5.2.3	Integrated Force Method .....	140
5.2.4	Numerical Examples .....	142
5.2.4.1	Example 1: 117-Member Cantilever Truss .....	143
5.2.4.2	Example 2: 130-Member Transmission Tower .....	145
5.2.4.3	Example 3: 368-Member Truss Dome .....	148
5.2.5	Summary .....	153
6.	GUIDED STOCHASTIC SEARCH TECHNIQUE FOR DISCRETE SIZING OPTIMIZATION OF STEEL FRAMES .....	155
6.1	Introduction .....	155
6.2	Sensitivity Index of Frame Members .....	155
6.3	The GSS for Discrete Sizing Optimization of Steel Frames .....	157
6.4	Local Search and Move-Back Mechanisms in GSS .....	162
6.5	Numerical Examples .....	164
6.5.1	Example 1: 135-Member Steel Frame .....	166
6.5.2	Example 2: 3860-Member Steel Frame .....	168
6.5.3	Example 3: 11540-Member Steel Frame .....	175
6.6	Summary .....	181
7.	CONCLUSION .....	183
7.1	Summary and Concluding Remarks .....	183
7.2	Recommendations for Future Research .....	185
	REFERENCES .....	187
	CURRICULUM VITAE .....	200

## LIST OF TABLES

### TABLES

Table 3.1 Cross sectional properties of the ready sections .....	47
Table 3.2 Number of analyses and size of the design space for the test problems .....	48
Table 3.3 Comparison of optimization results for 160-bar space pyramid .....	48
Table 3.4 Comparison of optimization results for 354-bar braced dome .....	53
Table 3.5 Comparison of optimization results for 693-bar braced barrel vault .....	58
Table 3.6 The gravity and lateral loading on 132-member space steel frame .....	65
Table 3.7 Comparison of results for 132-member space steel frame .....	66
Table 3.8 Member grouping details for 209-member industrial factory building .....	73
Table 3.9 Comparison of results for 209-member industrial factory building .....	73
Table 4.1 Optimum designs obtained for 135-member steel frame .....	87
Table 4.2 Computational details for 135-member steel frame .....	88
Table 4.3 Optimum designs obtained for 1026-member steel frame .....	94
Table 4.4 Computational details for 1026-member steel frame .....	95
Table 4.5 Optimal cross sectional areas for the 10-bar truss structure (case-1) .....	101
Table 4.6 Optimal cross sectional areas for the 10-bar truss structure (case-2) .....	101
Table 4.7 Optimal cross sectional areas for the 45-bar truss bridge .....	104
Table 4.8 Optimal cross sectional areas for the 120-bar truss dome (case-1) .....	108
Table 4.9 Optimal cross sectional areas for the 120-bar truss dome (case-2) .....	109
Table 5.1 Comparison of optimum designs for 117-member cantilever truss .....	125
Table 5.2 Loading of 130-member transmission tower .....	129
Table 5.3 Comparison of optimum designs for 130-member transmission tower ...	129
Table 5.4 Comparison of the optimum designs for 392-member double layer grid..	132
Table 5.5 Comparison of the optimum designs for 354-member truss dome .....	136
Table 5.6 Comparison of optimum designs for 117-member cantilever truss .....	145
Table 5.7 Loading of 130-member transmission tower .....	147
Table 5.8 Comparison of optimum designs for 130-member transmission tower ...	148
Table 5.9 Comparison of optimum designs for 368-member truss dome .....	152

Table 6.1 Optimum designs obtained for 135-member steel frame .....	167
Table 6.2 Optimum designs obtained for 3860-member steel frame .....	170
Table 6.3 Optimum designs obtained for 11540-member steel frame .....	177

## LIST OF FIGURES

### FIGURES

Figure 3.1 The variations of (a) $(SDR)_{ave}$ and, (b) $(SS)_{ave}$ in a typical run of BB-BC .....	36
Figure 3.2 The exponential distribution for various values of $\lambda$ .....	39
Figure 3.3 The variations of (a) $(SDR)_{ave}$ and, (b) $(SS)_{ave}$ in a typical run of MBB-BC .....	41
Figure 3.4 The variations of (a) $(SDR)_{ave}$ and, (b) $(SS)_{ave}$ in a typical run of EBB-BC .....	41
Figure 3.5 160-bar pyramid; (a) 3-D view, (b) Front view, (c) Plan view .....	45
Figure 3.6 Optimization histories for 160-bar space pyramid .....	46
Figure 3.7 160-bar pyramid test problem; variations of (a) $(SDR)_{ave}$ and, (b) $(SS)_{ave}$ .....	47
Figure 3.8 354-bar braced truss dome; (a) 3D view, (b) top view and (c) side view..	50
Figure 3.9 The three load cases considered for 354-bar braced truss dome .....	52
Figure 3.10 Optimization histories for 354-bar braced dome .....	53
Figure 3.11 693-bar braced barrel vault; a) 3-D view, b) Front view, c) Plan view ..	55
Figure 3.12 Optimization histories for 693-bar braced barrel vault .....	57
Figure 3.13 693-bar barrel vault test problem; variations of (a) $(SDR)_{ave}$ and, (b) $(SS)_{ave}$ .....	57
Figure 3.14 960-bar double layer grid .....	60
Figure 3.15 Optimization histories for 960-bar double layer grid .....	61
Figure 3.16 132-member space steel frame a) 3D view, b) front view, c) top view ..	63
Figure 3.17 $(SDR)_{ave}$ variations for 132-member space steel frame example; (a) BB-BC, and (b) EBB-BC variants .....	65



Figure 3.18 ( $SS$ ) <sub>ave</sub> variations for 132-member space steel frame example; (a) BB- BC, and (b) EBB-BC variants .....	65
Figure 3.19 209-member industrial factory building .....	70
Figure 3.20 ( $SDR$ ) <sub>ave</sub> variations for 209-member industrial factory building example; (a) BB-BC, and (b) EBB-BC variants .....	72
Figure 3.21 ( $SS$ ) <sub>ave</sub> variations for 209-member industrial factory building example; (a) BB-BC, and (b) EBB-BC variants .....	73
Figure 4.1 135-member steel frame, (a) 3-D view (b) side view of frames 1 and 3 (c) side view of frame 2 (d) side view of frames A, B, C, D, and E (e) plan view .....	84
Figure 4.2 Columns grouping of 135-member steel frame in plan level .....	86
Figure 4.3 Optimization histories of the 135-member steel frame example .....	86
Figure 4.4 Saving in structural analyses using the UBB-BC algorithm in the 135-member steel frame example .....	86
Figure 4.5 Saving in structural analyses using the UMBB-BC algorithm in the 135-member steel frame example .....	87
Figure 4.6 Saving in structural analyses using the UEBB-BC algorithm in the 135-member steel frame example .....	87
Figure 4.7 1026-member steel frame, (a) 3-D view (b) side view of frames 2, 3, and 4 (c) side view of frames 1 and 5 (d) side view of frames A, B, C, D, E, F, and G (e) plan view .....	91
Figure 4.8 Columns grouping of 1026-member steel frame in plan level .....	92
Figure 4.9 Optimization histories of the 1026-member steel frame example .....	92
Figure 4.10 Saving in structural analyses using the UBB-BC algorithm in the 1026-member steel frame example .....	93
Figure 4.11 Saving in structural analyses using the UMBB-BC algorithm in the 1026-member steel frame example .....	93
Figure 4.12 Saving in structural analyses using the UEBB-BC algorithm in the 1026-member steel frame example .....	93
Figure 4.13 Net weights of individuals generated using the UBB-BC algorithm .....	96

Figure 4.14 Net weights of individuals generated using the UMBB-BC algorithm .. 96

Figure 4.15 Net weights of individuals generated using the UEBB-BC algorithm ... 97

Figure 4.16 10-bar truss structure ..... 99

Figure 4.17 10-bar truss example; saving in structural analyses using the UBB-BC algorithm ..... 99

Figure 4.18 10-bar truss example; variations of the total number of structural analyses using the UBB-BC and BB-BC algorithms ..... 100

Figure 4.19 10-bar truss example; optimization histories of the UBB-BC and BB-BC algorithms (a) case-1 (b) case-2 ..... 100

Figure 4.20 45-bar truss bridge ..... 102

Figure 4.21 45-bar truss bridge example; saving in structural analyses using the UBB-BC algorithm ..... 103

Figure 4.22 45-bar truss bridge example; variations of the total number of structural analyses using the UBB-BC and BB-BC algorithms ..... 103

Figure 4.23 45-bar truss example; optimization histories of the UBB-BC and BB-BC algorithms ..... 104

Figure 4.24 120-bar truss dome ..... 106

Figure 4.25 120-bar truss dome example; saving in structural analyses using the UBB-BC algorithm (a) case-1 (b) case-2 ..... 107

Figure 4.26 120-bar truss dome example; variations of the total number of structural analyses using the UBB-BC and BB-BC algorithms ..... 107

Figure 4.27 120-bar truss example; optimization histories of the UBB-BC and BB-BC algorithms (a) case-1 (b) case-2 ..... 108

Figure 5.1 Flowchart of truss optimization process using GSS ..... 121

Figure 5.2 117-member cantilever truss ..... 124

Figure 5.3 Convergence history of the best feasible generated design for 117-member cantilever truss using GSS ..... 124

Figure 5.4 Convergence histories for 117-member cantilever truss using some metaheuristics ..... 125

Figure 5.5 130-member transmission tower ..... 127

Figure 5.6 Loaded nodes of 130-member transmission tower ..... 127

Figure 5.7 Convergence history of the best feasible generated design for 130-member transmission tower using GSS .....	128
Figure 5.8 Convergence histories for 130-member transmission tower using some metaheuristics .....	128
Figure 5.9 392-member double layer grid (a) 3-D view (b) top view .....	130
Figure 5.10 Convergence history of the best feasible generated design for 392-member double layer grid using GSS .....	131
Figure 5.11 Convergence histories for 392-member double layer grid using some metaheuristics .....	132
Figure 5.12 354-member truss dome (a) 3-D view (b) top view (c) side view .....	134
Figure 5.13 Convergence history of the best feasible generated design for 354-member truss dome using GSS .....	135
Figure 5.14 Convergence histories for 354-member truss dome using some metaheuristics .....	136
Figure 5.15 Convergence history of the best feasible generated design for 117-member cantilever truss using (a) GSS <sub>A</sub> , (b) GSS <sub>B</sub> .....	144
Figure 5.16 Convergence histories for 117-member transmission tower using some metaheuristics .....	144
Figure 5.17 Convergence history of the best feasible generated design for 130-member transmission tower using (a) GSS <sub>A</sub> , (b) GSS <sub>B</sub> .....	146
Figure 5.18 Convergence histories for 130-member transmission tower using some metaheuristics .....	147
Figure 5.19 368-member truss dome (a) 3-D view (b) top view (c) side view .....	150
Figure 5.20 Convergence history of the best feasible generated design for 368-member truss dome using (a) GSS <sub>A</sub> , (b) GSS <sub>B</sub> .....	151
Figure 5.21 Convergence histories for 368-member truss dome using some metaheuristics .....	152
Figure 6.1 Flowchart of frame optimization process using GSS .....	165
Figure 6.2 Optimization history of 135-member steel frame using GSS .....	167
Figure 6.3 Optimization histories of 135-member steel frame using some metaheuristics .....	168

Figure 6.4 3860-member steel frame, (a) 3-D view (b) side view of frames B, D, F, and H (c) side view of frames C, E, and G (d) side view of frames A, and I (e) side view of frames 1, 3, 5, and 7 (f) side view of frames 2, 4, and 6 (g) plan view ..... 171

Figure 6.5 Column grouping in plan level for 3860-member steel frame ..... 173

Figure 6.6 Optimization history of 3860-member steel frame using GSS ..... 174

Figure 6.7 Optimization history of 3860-member steel frame using UEBC-BC ..... 174

Figure 6.8 11540-member steel frame, (a) 3-D view (b) side view of frames 2, 3, 4, 6, 7, 8, 10, 11, 12, B, C, D, F, G, H, J, K and L (c) side view of Frames 1, 5, 9, 13, A, E, I, and M (d) plan view ..... 178

Figure 6.9 Columns' orientations of 11540-member steel frame ..... 179

Figure 6.10 Outline of column grouping in plan level for 11540-member steel frame ..... 180

Figure 6.11 Optimization history of 11540-member steel frame using GSS ..... 180

Figure 6.12 Optimization history of 11540-member steel frame using UEBC-BC.. 181

# CHAPTER 1

## INTRODUCTION

### 1.1 Basic Elements of Structural Optimization

Due to limitations in natural resources, optimization has always been an inseparable component of structural design. This fact has resulted in development of a large number of optimization techniques in the past few decades to achieve robust and reliable design tools for dealing with complicated structural optimization problems. Typically, an optimal design problem is composed of three basic elements: (i) objective function, (ii) design variables, and (iii) constraints. In structural design optimization usually weight or cost of the structure is taken as the objective function of the problem. Here, the fundamental aim is to minimize the final weight or cost of the structure which is a function of the design variables. The design variables are those parameters which are to be determined by the designer in order to generate an optimal solution. Furthermore, in practical applications achieving an optimum design should be carried out with respect to a set of strength and serviceability limitations i.e. design constraints.

In the literature, classifying the structural optimization problems is basically carried out regarding the type of design variables involved. It is worth mentioning that since optimum design of continuum structures is out of the scope of this study, the followings explicitly cover practical design optimization of skeletal structures (i.e. truss and frame structures). Generally, optimum design of skeletal structures is divided into three main categories as sizing, shape, and topology optimization. In sizing optimization the cross sectional areas of structural members are considered as design variables. This can further be divided into two subcategories in terms of the

nature of the design variables employed: continuous and discrete. In continuous sizing optimization any positive value can be assigned to cross sectional areas of elements. However, this is usually not the case in practical applications, where structural members should be adopted from a set of available sections. The latter is addressed to discrete sizing optimization. In shape optimization, the best positions of a selected group of joints in a structure are determined. Due to practical aspects this type of design optimization is usually involved in optimum design applications of truss structures rather than frames. In both the aforementioned optimization categories topology of a structure is assumed to be fixed. However, it is sometimes more expedient to search for the optimum topology of a structure, which entails considering the presence or absence of structural components, such as elements and nodes.

Since in practical sizing optimization of skeletal structures usually the structural members are to be adopted from a set of available sections, the design problem turns into a discrete sizing optimization. Here the aim is to seek for the best set of ready sections which yield the optimum design. Although for a given structure in fact the number of candidate designs is numerically limited, however, in real world applications performing an exhaustive search is not possible in a reasonable computational time. Therefore, structural optimization techniques have been developed for locating the optimum or a reasonably good near optimum solution through investigating a portion of the design space in a reasonable computational time. These techniques and their applications in optimal design of structural systems are outlined in the following sections.

## **1.2 Structural Optimization Techniques**

During the past decades, inherent complexity of practical structural optimization problems motivated the researchers to develop efficient and robust optimization techniques. Basically, structural optimization methods can be divided into two main categories: (i) traditional methods (ii) modern techniques. Two main categories of traditional structural optimization techniques include mathematical programming and

optimality criteria approaches, whereas heuristics or metaheuristic search techniques are referred to as the modern structural optimization methods. An overview of these techniques is provided in this section as follows. Mathematical programming techniques are amongst the well known classes of structural optimization techniques which work based on gradients of the objective function. The basic idea is to move in the negative direction of the gradient of the objective function to find a more promising candidate design. Many studies have been conducted on application of mathematical programming techniques in structural design optimization so far (Belegundu and Arora 1985, Rashedi and Moses 1986, Hall et al. 1989, Erbatur and Al-Hussainy 1992). However, it is generally conceived that mathematical programming techniques are not efficient for optimum design of structural systems having numerous design variables.

Another class of traditional structural optimization techniques covers optimality criteria methods. Typically in optimality criteria methods first a set of necessary optimality criteria (such as Kuhn-Tucker conditions) are derived for the design. Next, in order to generate an optimum design a recursive algorithm is employed to update the structural members for satisfying the optimality criteria. Early works on optimality criteria methods are due to Prager et al. (1967), Prager (1968), Venkayya et al. (1973). Later, numerous variants of the optimality criteria methods are applied to optimum design of pin-jointed (Feury and Geradin 1978, Fleury 1980, Saka 1984) and frame structures (Tabak and Wright 1981, Khan 1984, Chan et al. 1995). It is worth mentioning that the well known fully stressed design (FSD) (Gallagher 1973, Patnaik et al. 1998) can be also considered as a simple stress-ratio optimality criteria technique which can only deal with stress constraints. An extension of FSD to handle both stress and displacement constraints is fully utilized design (FUD). The FUD includes two main steps (i) providing a FSD considering stress constraints (ii) prorating the FSD to obtain the FUD. The proportion parameter is computed with respect to the most violated displacement constraint. Although FUD is capable of generating a feasible solution through a small number of structural analyses, the obtained solution can be an overdesign. Patnaik et al. (1998) developed a modified

fully utilized design (MFUD) using advantage of the integrated force method (IFM) of structural analysis (Patnaik et al. 1991), however the study was limited to truss structures.

Basically, traditional structural optimization techniques i.e. mathematical programming and optimality criteria are developed for handling continuous design variables, hence, they are not effective for tackling practical discrete optimization instances. Furthermore, gradient based formulations of such methods entail different types of approximations which sometimes are far from the reality. Therefore, due to the shortcomings of traditional techniques in handling real world design optimization instances, in the recent decades, stochastic search techniques or metaheuristics have received an increasing attention and found plenty of applications in structural optimization field.

In general, metaheuristic techniques, such as genetic algorithms (GAs) (Goldberg and Samtani 1986), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), ant colony optimization (ACO) (Dorigo 1992), etc., borrow their working principles from natural phenomena (Yang 2008); and follow non-deterministic search strategies in locating the optimum solutions. The rising popularity of these techniques arise from (i) the lack of dependency on gradient information; (ii) inherent capability to deal with both discrete and continuous design variables; and (iii) automated global search features to produce near-optimum solutions (if not the global optimum) for complicated problems. In addition, the simplicity in their coding makes it possible to avoid cumbersome formulations frequently encountered with traditional structural optimization technique, and renders them ideal and prevalent tools for structural optimization applications. The state-of-the-art reviews of metaheuristic techniques and their applications in structural design optimization problems are provided in Saka (2007a), Lamberti and Pappalettere (2011a), Saka and Geem (2013), and Hare et al. (2013).



### **1.3 Optimization of Skeletal Structures using Metaheuristics: A Survey of the State-of-the-Art**

Undoubtedly, in the recent years, most of the optimization algorithms developed for optimum design of truss and frame structures belong to the class of stochastic search algorithms or metaheuristics. Besides various inspiration resources reported in the literature for development of metaheuristic search techniques, in fact these techniques have similar characteristics. Basically, a metaheuristic structural optimization algorithm aims to locate the global optimum in the solution space through generating candidate solutions in an iterative way. Roughly speaking, the fundamental idea is to seek the vicinity of more promising candidate designs found so far to drive the search towards more reliable portions of the solution space. Since working principles of these techniques are somewhat identical, a general outline is provided here. Detailed descriptions of various types of metaheuristic algorithms can be found in (Yang 2008).

Typically, a metaheuristic structural optimization algorithm initiates with a population of randomly generated candidate designs. The initial population is mostly composed of infeasible solutions due to this random generation. Then in order to investigate the quality of generated designs, each candidate design is evaluated with respect to a merit function which can be the objective function of the problem. Once merit or fitness of each candidate design is computed, new candidate designs can be generated using the obtained information from the formerly generated designs. Generally, different mechanisms and operators are utilized for generating a new population of solutions to guide the search towards the optimum. In fact the key difference between the algorithms is in the way that they propose the next move in the solution space i.e. the new candidate design. In metaheuristics generation of new populations is iteratively performed until a predefined termination criterion, which is usually the maximum number of iterations, is met. The last iteration of a metaheuristic algorithm is expected to include the optimum or a reasonably good near optimum design. Many studies have been conducted on structural optimization using metaheuristics so far.

Some of the most recent applications of these algorithms in optimum design of truss and frame structures are outlined below.

In fact GAs are the most well known metaheuristic algorithms frequently employed for structural optimization applications. Kameshki and Saka (2001a) studied the effect of bracing on the optimum design of planar steel frames using GAs. In their study, first, a 15 story frame was designed assuming rigid beam column connections and fix supports, as well as rigid frame with pin supports. Then the same frame, with pin supports, was designed assuming pin beam-column connections with four types of frequently used bracing systems. Through investigating X-bracing, X-bracing with outrigger truss, V-bracing and Z-bracing systems they presented a clear numerical comparison among the minimum weights obtained for all the aforementioned systems. According to the reported results X-bracing provided the minimum weight frame in comparison to the other considered structural systems.

Kameshki and Saka (2001b) presented a GA for optimum design of steel frames with semi-rigid connections based on BS 5950 (1990) specifications. Considering the optimum design of two unbraced steel frames with end plate connection without column stiffeners, it was deducted that the semi-rigid connection modeling produces lighter designs. In their work the number of investigated semi-rigid connections was limited to one type. In a more comprehensive study, Kameshki and Saka (2003) evaluated the effect of connection flexibility and the geometric non-linearity of the frame members in the optimum design of planer steel frames. A GA based approach was employed for optimization of three unbraced steel frames with rigid and three different types of semi-rigid connections. It was demonstrated that considering the geometric nonlinearity, in the analysis stage, leads to lighter frames in case of rigid connections.

Kaveh and Kalatjari (2002) employed the force method for structural analysis stage of a genetic algorithm based structural optimization technique. Considering optimum design of truss structures they demonstrate the computational efficiency of the

proposed method. Hayalioglu and Degertekin (2005) developed a genetic algorithm for cost efficient design of steel frames with semi-rigid connections and column bases. The optimization results of three planar steel frames using eight different types of semi-rigid connections and column bases was compared to those obtained using rigid connections. According to the presented numerical results, instead of using rigid connections, sometimes choosing specific types of semi-rigid connections could to be more economical. It should be noted that, since the study does not cover seismic loading, the conclusion cannot be generalized for the steel frames exposed to seismic loads. Later, Degertekin et al. (2008) investigated the efficiency of the tabu search (Glover 1989) and genetic algorithm in design optimization of geometrically nonlinear steel space frames based on the AISC-LRFD (1995) specifications. According to the investigated examples, tabu search algorithm resulted in lighter designs.

Recently, Kazemzadeh Azad et al. (2012) developed a mutation based genetic algorithm for sizing and shape optimization of planar and spatial truss structures. An adaptive tournament selection mechanism in combination with adaptive Gaussian mutation operators were used to achieve an effective search in the design space. The efficiency of the proposed GA was demonstrated using design examples of truss structures with both discrete and continuous design variables.

The PSO algorithm proposed by Kennedy and Eberhart (1995) is another popular metaheuristic search technique with extensive applications in the field of structural design optimization. Fourie and Groenwold (2002) applied the PSO algorithm to design optimization instances with sizing and shape variables and compared its performance to that of GA as well as three gradient based techniques. Considering optimum design of three truss structures and a torque arm the authors demonstrated the suitability of the PSO in tackling structural optimization problems. Perez and Behdinan (2007) investigated the effect of different parameter settings on the efficiency of the PSO algorithm through optimal design of classical truss optimization instances. Li et al. (2007, 2009) proposed improved variants of the PSO as heuristic

PSO algorithms for optimum design of truss structures. Further, Kaveh and Talatahari (2009a) developed a hybrid version of the PSO algorithm for discrete sizing optimization of truss structures and demonstrated its promising performance. Luh and Lin (2011) used a two stage PSO algorithm for minimum weight design of truss structures. In their approach first a topology optimization is performed using a ground structure and next sizing and shape optimizations are carried out to locate the minimum weight design. Recently, Gomes (2011) employed the PSO algorithm for sizing and shape optimization of truss structures with frequency constraints and reported promising performance of the technique.

Geem et al. (2001) developed the harmony search (HS) algorithm as a new meta-heuristic technique. Later, Lee and Geem (2004) used the algorithm for sizing optimization of truss structures and demonstrated its efficiency compared to conventional mathematical methods as well as genetic algorithm. They concluded that the algorithm can be also employed for optimum design of other types of structures such as frame, plate or shell structures. Saka (2007b) demonstrated the efficiency of the HS algorithm in optimum geometry design of single layer geodesic domes. In their approach the height of the dome crown was treated as a design variable along with the cross-sectional designations of dome members. Later, Carbas and Saka (2009) employed the algorithm for design optimization of single layer network domes. Saka (2009) used the HS algorithm for optimum design of steel sway frames. Recently, Hasancebi et al. (2010a) presented an adaptive harmony search algorithm for structural optimization and employed it for sizing optimization of a 162-member braced planar steel frame and a 744-member unbraced space steel frame. Unlike the standard HS algorithm where the control parameters are typically set to constant values, in their algorithm these parameters are adaptively tuned during the search to establish a tradeoff between the exploration and exploitation in the design space. They illustrated the efficiency of their adaptive approach through comparison of the obtained numerical results with those of four other metaheuristic algorithms.

Another novel metaheuristic algorithm is a nature-inspired method so called artificial bee colony (ABC) algorithm which is proposed by Karaboga (2005). One recent application of this algorithm in sizing optimization of planar and space truss structures is due to Hadidi et al. (2010). The authors proposed some modifications in the original algorithm and reported satisfactory results of optimum design of four truss structure examples. Sonmez (2011a) used a discrete ABC algorithm for optimum design of truss structures with up to 582 members and reported promising performance of the algorithm compared to the other well known metaheuristic techniques. Furthermore, Sonmez (2011b) employed the ABC algorithm with an adaptive penalty function approach for minimum weight design of truss structures with fixed geometries. Besides the available works in the literature on application of the ABC in optimum design of truss structures, further research is required to investigate the performance of the algorithm in optimum design of steel frames.

Erol and Eksin (2006) introduced a new metaheuristic optimization method called Big Bang–Big Crunch (BB-BC) algorithm. Due to the simple algorithmic outline of the method as well as its efficiency in tackling practical optimization instances, it has become one of the popular metaheuristics of the recent years. The first application of the algorithm in optimum design of skeletal structures was carried out by Camp (2007). In his work the optimum design of planar and spatial truss structures was performed using a modified version of the algorithm. In order to increase the efficiency of the BB–BC algorithm, Camp (2007) introduced a weighting parameter to control the influence of both the center of mass and the current global best solution on new candidate solutions. Further, a multiphase search strategy was employed to increase the quality of final solution. The study demonstrated the efficiency of the BB–BC algorithm in comparison to the previously reported GA, PSO, and ACO based approaches. Later, Kaveh and Abbasgholiha (2011) adopted the Camp’s strategy of generating new candidate solutions for design optimization of planar steel sway frames. Lamberti and Pappalettere (2011b) proposed an improved version of the BB-BC algorithm for weight minimization of truss structures and reported promising results using four benchmark truss optimization examples. Kaveh and Talatahari

(2009b,c , 2010a) developed hybrid versions of the BB-BC algorithm for optimum design of different types of skeletal structures. Recently, in Kazemzadeh Azad et al. (2011) the success of BB-BC algorithm in benchmark problems of engineering optimization is investigated.

Charged system search (CSS) is a very recent meta-heuristic optimization algorithm proposed by Kaveh and Talatahari (2010b). The authors employed the algorithm for optimum design of skeletal structures including three truss and two frame structures (Kaveh and Talatahari 2010c). The study revealed the efficiency of the CSS in comparison to the other heuristic methods. In another study the authors applied a discrete version of the algorithm to sizing optimization of different types of truss structures with fixed configurations (Kaveh and Talatahari 2010d). Furthermore, an enhanced version of the CSS algorithm is used for configuration optimization of truss structures (Kaveh and Talatahari 2011). Additionally, the efficiency of the charged system search is demonstrated considering the optimum design of three benchmark examples of frame structures with up to 290 members in Kaveh and Talatahari (2012). Besides the above-mentioned algorithms, numerous metaheuristic search techniques have been also developed in the recent years to deal with challenging optimization problems (Yang 2010, Hasançebi and Kazemzadeh Azad 2012, Kaveh and Khayatizad 2012). Due to the variety of metaheuristic techniques available in the literature of structural optimization, adopting an appropriate method for practical applications may turn into a confusing task. Therefore, comparison and performance evaluation of optimization algorithms can lessen the burden of choosing an efficient algorithm to deal with a given design optimization problem. In this regard, Hasançebi et al. (2009) evaluated the performance of seven metaheuristic optimization algorithms in optimal design of truss structures. The investigated algorithms include GA, SA, ES, PSO, tabu search (TS), ant colony optimization (ACO) and HS. The algorithms were compared through design optimization of four real size truss structures according to the design limitations of AISC-ASD (1989). The study revealed the superiority of SA and ES to the other techniques in design optimization of truss structures. Later, Hasancebi et al. (2010b) investigated the performance of the

above mentioned algorithms in design optimization of steel frames with rigid connections. In the studied three steel frame examples, the two best performances were related to ES and SA algorithms amongst the other techniques. The study provides general guidelines for future practical applications of metaheuristics in design optimization of steel frames.

#### **1.4 Today's Dilemma in Practical Structural Optimization**

Despite many studies conducted on developing efficient optimization algorithms for structural optimization applications, no unique method is accepted to be the most successful approach for optimum design of skeletal structures so far. Basically, two main factors determine the efficiency of a design optimization algorithm. The first criterion is optimality of the obtained final design and the second measure is speed of the algorithm in finding the optimum solution. The latter is highly dependent on the number of structural analysis required in the optimization process to locate the optimum or a relatively good near optimum solution.

In spite of many advantageous characteristics of modern structural optimization techniques namely metaheuristics, the slow rate of convergence towards the optimum as well as the need for a high number of structural analyses are conceived as the downside of the search features of these techniques in structural optimization applications. It is known that response computations of designs sampled during a search process usually occupies 85-95% workload of a metaheuristic technique (Hasançebi et al. 2011a), and thus large number of structural analyses substantially increases the total computing time. Here, one solution to this is to reduce the total computational time by taking advantage of high performance computing methods, such as parallel or distributed computing methods. The idea in this approach is to distribute the total workload of the algorithm amongst multiprocessors of a single computer or within a cluster of computers connected to each other via local area network. In Hasançebi et al. (2011) it is shown that a maximum speedup ratio between

12.2 and 16.8 can be achieved for three large-scale design examples solved using a cluster computing system consisting of 32 processors.

Another approach, which is more straightforward and easier to apply, is to develop efficient strategies for diminishing the number of structural analyses required in the course of optimization. The latter, can be carried out by proposing efficient optimization techniques that are able to locate a reasonable solution using fewer numbers of structural analyses, i.e. less computational effort.

It is worth mentioning that although numerous studies demonstrate the applicability of modern optimization techniques in structural design optimization, still optimization is not established well in typical design of skeletal structures basically due to the following reasons. On one hand enormously time consuming procedures of modern techniques make structural engineers reluctant to use them in real world applications. The computational inefficiency of modern techniques makes it almost impossible to use them for large scale applications without utilizing high performance computing techniques. As a result of this, structural engineers generally do not receive benefits of optimization in large scale applications wherein optimality of final designs are much more important in comparison to small size structures.

On the other hand since in practical applications structural analysis and design check stages are generally carried out by commercial software such as SAP2000, optimization tools should be developed such that they can work integrated with the analyzer. In other words, automated design optimization platforms are needed for simultaneous modeling, analysis, and design optimization of structures. However, generally the existing design optimization software, capable of performing design optimization integrated with a structural analysis program, support only one design code specifications or one type of skeletal structures only. Hence, considering the above-mentioned issues, the new trend of structural design optimization is towards developing automated optimum design techniques capable of locating promising solutions in a reasonable computational time for practical applications. There is also a



great demand for robust and efficient algorithms capable of handling large scale systems without employing expensive high performance computing techniques.

### **1.5 Aim and Scope of the Thesis**

The theoretical and practical objectives of the study are as follows: (i) developing an automated design optimization environment for practical applications; (ii) improving the performance of existing modern structural optimization algorithms through reformulations and modifications in their algorithmic structures; (iii) proposing efficient strategies for diminishing the total computational effort of the existing structural optimization techniques; (iv) developing computationally efficient structural design optimization techniques based on the principles of structural mechanics; (v) facilitating practical optimum design of large scale structural systems and removing the need for expensive high performance computing techniques such as parallel or distributed computing methods. The following chapters of the thesis are organized as follows.

Chapter 2 provides a mathematical formulation of the considered discrete sizing optimization of steel truss and frame structures. Since AISC-ASD and AISC-LRDF specifications are both used in the investigated optimum design examples, for the sake of clarity, the corresponding criteria are outlined in this chapter for both the aforementioned design codes.

Chapter 3 covers performance enhancement of a novel metaheuristic technique, i.e., the BB-BC algorithm for discrete sizing optimization. It is shown that the standard version of the algorithm is sometimes unable to produce reasonable solutions to problems from discrete design optimization of steel skeletal structures. Hence, through investigating the shortcomings of the BB-BC algorithm, it is aimed to reinforce the performance of the technique for this class of problems in particular. Reformulations of the BB-BC algorithm are proposed, where the formula used by the standard algorithm for generating new candidate solutions around the center of mass

is efficaciously reformulated, resulting in the so-called modified BB-BC (MBB-BC) and exponential BB-BC (EBB-BC) variants for truss and frame type structures, respectively. The performances of the proposed algorithms are compared to those of other well-known metaheuristic techniques using various practical design optimization instances.

Chapter 4 proposes an upper bound strategy (UBS) for reducing the total computational effort in metaheuristic based structural optimization algorithms. The idea behind the UBS is to detect those candidate designs which have no chance to improve the search during the iterations of the algorithm. After identifying the non-improving candidate designs, they are directly excluded from the structural analysis stage, resulting in diminishing the total computational effort. The well-known big bang-big crunch algorithm as well as its two enhanced variants, namely MBB-BC and EBB-BC algorithms, are selected as typical metaheuristic algorithms to investigate the effect of the UBS on computational efficiency of these techniques. The numerical results obtained from optimum design of steel truss and frame structures clearly reveal that the UBS can significantly lessen the total computational time in metaheuristic based design optimization of steel structures.

Chapter 5 describes the development of a guided stochastic search (GSS) technique for computationally efficient design optimization of steel trusses. The GSS offers a stochastic procedure where the optimization process is guided by the principle of virtual work and response computations of the generated designs, resulting in an efficient and rapid search. In the proposed method, the information provided through the structural analysis and design check stages are used for handling strength constraints. Furthermore, the well-known principle of virtual work is utilized to detect the most effective structural members for satisfying displacement constraints. The optimum sizing of a structure is then performed using an integrated approach wherein both strength and displacement criteria are taken into account for reduction of the member sizes along the way the aforementioned constraints are handled.

In chapter 5 first the performance of the proposed GSS technique is evaluated in discrete sizing optimization problems of truss structures having a single displacement constraint under a single load case. Next, the GSS is further improved for tackling a more general class of truss optimization problems subject to multiple displacement constraints and load cases. To this end, enhancements of the GSS are proposed in the form of two alternative approaches that enable the technique to deal with multiple displacement/load cases. The first approach implements a methodology in which the most critical displacement direction is considered only when guiding the search process. The second approach, however, takes into account the cumulative effect of all the critical displacement directions in the course of optimization. Here, advantage of the integrated force method of structural analysis is utilized for further reduction of the computational effort in these approaches. The performance of the proposed GSS technique is evaluated and compared with some selected techniques of metaheuristics through real-size trusses that are sized for minimum weight. The numerical results indicate the computational efficiency and robustness of the proposed technique compared to metaheuristic algorithms.

Chapter 6 provides a GSS based computationally enhanced algorithm for frame optimization. In this chapter the GSS technique is refined and reformulated for tackling discrete sizing optimization problems of steel frames. In order to increase the efficiency of the method local search and move-back mechanisms are integrated with the original GSS. On the one hand, the proposed local search mechanism helps the GSS increase the quality of solutions found during the search process without deteriorating its global search features. On the other hand, the move-back mechanism is introduced to avoid additional iterations required for modifying probable highly oversized solutions in the course of optimization. The proposed refined and reformulated GSS is compared with some contemporary metaheuristics using three real-size steel frames. The numerical results obtained demonstrate the computational efficiency of the proposed approach in optimum design of steel frames.

Chapter 7 presents a brief conclusion of the thesis and highlights the most significant outcomes of the study. At last, some recommendations are provided for further research.

## CHAPTER 2

### OPTIMUM DESIGN PROBLEM FORMULATION

#### 2.1 Discrete Sizing Optimization Problem to AISC-ASD

Typically in practical design optimization of steel structures the aim is to find a minimum weight design by selecting cross-sectional areas of structural members from a table of available sections, such that the final design satisfies strength and serviceability requirements according to a chosen code of design practice. According to AISC-ASD (1989) specifications for a structure composed of  $N_m$  members that are collected in  $N_g$  groups, this problem can be formulated as follows. The objective is to find a vector of integer values  $\mathbf{I}$ , Eq. (2.1), representing the sequence numbers of standard sections for  $N_g$  member groups in the section table

$$\mathbf{I}^T = [I_1, I_2, \dots, I_{N_d}] \quad (2.1)$$

to minimize the weight ( $W$ ) of the structure

$$W = \sum_{i=1}^{N_g} \rho_i A_i \sum_{m=1}^{N_s} L_m \quad (2.2)$$

where  $A_i$  and  $\rho_i$  are the length and unit weight of a standard section adopted for member group  $i$ , respectively,  $N_s$  is the total number of members in group  $i$ , and  $L_m$  is the length of the member  $m$  which belongs to group  $i$ . The design constraints, which are formulated separately for steel trusses and frames in the following subsections, consist of certain limitations imposed on overall structural response as well as individual members.

### 2.1.1 Design Constraints for Steel Trusses

The constraints usually involved in the design of steel trusses consist of following inequalities:

$$\frac{f_a}{F_a} - 1 \leq 0 \quad (2.3)$$

$$\frac{\lambda}{\lambda_a} - 1 \leq 0 \quad (2.4)$$

$$\frac{d}{D_a} - 1 \leq 0 \quad (2.5)$$

where  $f_a$  and  $F_a$  are the computed axial stress and its allowable value for a member, respectively;  $\lambda$  and  $\lambda_a$  are the slenderness ratio and its allowable value for a member, respectively; and  $d$  and  $D_a$  are the displacement computed at a joint in a certain direction and its allowable value, respectively.

In Eq. (2.3), the allowable axial stress in tension is taken as the smaller of  $0.60F_y$  and  $0.50F_u$ , where  $F_y$  and  $F_u$  stand for the yield and ultimate tensile strengths, respectively. On the other hand, the allowable stress in compression is calculated based on two possible failure modes known as elastic and inelastic buckling using Eqs. (2.6) and (2.7)

$$F_a = \frac{\left[1 - \frac{(KL/r)^2}{2C_c^2}\right] F_y}{\frac{5}{3} + \frac{3(KL/r)}{8C_c} - \frac{(KL/r)^3}{8C_c^3}}, \quad \lambda < C_c \text{ (inelastic buckling)} \quad (2.6)$$

$$F_a = \frac{12\pi^2 E}{23(KL/r)^2}, \quad \lambda \geq C_c \text{ (elastic buckling)} \quad (2.7)$$

where  $E$  is the elasticity modulus,  $K$  is the effective length factor,  $r$  is the radius of gyration about buckling axis, and  $C_c = \sqrt{2\pi^2 E / F_y}$  is the critical slenderness ratio. Finally, the maximum slenderness ratio  $\lambda_a$  in Eq. (2.4) is limited to 300 for tension members, whereas it is taken as 200 for compression members.

### 2.1.2 Design Constraints for Steel Frames

In steel frames the structural members that are subjected to a combination of axial compression and flexural stress must be sized to meet the following stress constraints:

$$\text{if } \frac{f_a}{F_a} > 0.15 ; \quad \left[ \frac{f_a}{F_a} + \frac{C_{mx} f_{bx}}{\left(1 - \frac{f_a}{F_{ex}}\right) F_{bx}} + \frac{C_{my} f_{by}}{\left(1 - \frac{f_a}{F_{ey}}\right) F_{by}} \right] - 1.0 \leq 0 \quad (2.8)$$

$$\left[ \frac{f_a}{0.60F_y} + \frac{f_{bx}}{F_{bx}} + \frac{f_{by}}{F_{by}} \right] - 1.0 \leq 0 \quad (2.9)$$

$$\text{if } \frac{f_a}{F_a} \leq 0.15 ; \quad \left[ \frac{f_a}{F_a} + \frac{f_{bx}}{F_{bx}} + \frac{f_{by}}{F_{by}} \right] - 1.0 \leq 0 \quad (2.10)$$

In case the flexural member is under tension, the following formula is employed:

$$\left[ \frac{f_a}{0.60F_y} + \frac{f_{bx}}{F_{bx}} + \frac{f_{by}}{F_{by}} \right] - 1.0 \leq 0 \quad (2.11)$$

In Eqs. (2.8-2.11),  $f_a$  represents the computed stress due to axial compression force, and the computed flexural stresses due to bending of the member about its major ( $x$ )

and minor (y) principal axes are denoted by  $f_{bx}$  and  $f_{by}$ , respectively.  $F'_{ex}$  and  $F'_{ey}$  denote the Euler stresses about principal axes of the member divided by a factory of safety of 23/12.  $F_a$  is the allowable compressive stress under axial compression force alone, and is calculated using Eq. (2.6) or (2.7) depending on elastic or inelastic buckling failure mode of the member. The allowable bending compressive stresses about major and minor axes are designated by  $F_{bx}$  and  $F_{by}$ , which are computed using the Formulas 1.5-6a or 1.5-6b and 1.5-7 given in AISC-ASD (1989). It is important to note that while calculating allowable bending stresses, a newer formulation, Eq. (2.12), of moment gradient coefficient  $C_b$  given in ANSI/AISC 360-05 (2005) is employed in the study to account for the effect of moment gradient on lateral torsional buckling resistance of the elements,

$$C_b = \frac{12.5M_{\max}}{2.5M_{\max} + 3M_A + 4M_B + 3M_C} R_m \leq 3.0 \quad (2.12)$$

where  $M_{\max}$ ,  $M_A$ ,  $M_B$  and  $M_C$  are the absolute values of maximum, quarter-point, midpoint, and three-quarter point moments along the unbraced length of the member, respectively, and  $R_m$  is a coefficient which is equal to 1.0 for doubly symmetric sections.  $C_{mx}$  and  $C_{my}$  are the reduction factors, introduced to counterbalance overestimation of the second-order moments by the amplification factor  $(1 - f_a / F'_e)$ . For braced frame members without transverse loading between their ends, they are calculated from  $C_m = 0.6 - 0.4(M_1 / M_2)$ , where  $M_1 / M_2$  is the ratio of smaller end moment to the larger end moment. For braced frame members having transverse loading between their ends, they are determined from the formula  $C_m = 1 + \psi(f_a / F'_e)$  based on a rational approximate analysis outlined in AISC-ASD (1989) Commentary-H1, where  $\psi$  is a parameter that considers maximum deflection and maximum moment in the member.

For the computation of allowable compression and Euler stresses, the effective length



factors ( $K$ ) are required. For beam and bracing members,  $K$  is taken equal to unity. For column members, alignment charts furnished in AISC-ASD (1989) can be utilized. In this study, however, the effective length factors of columns in braced steel frames are calculated from the following approximate formula developed by Dumonteil (1992), which are accurate to within about -1.0 and +2.0 % of the exact results (Hellesland 1994):

$$K = \frac{3G_A G_B + 1.4(G_A + G_B) + 0.64}{3G_A G_B + 2.0(G_A + G_B) + 1.28} \quad (2.13)$$

where  $G_A$  and  $G_B$  refer to stiffness ratio or relative stiffness of a column at its two ends.

In addition to axial force and bending, the members must be sized to have sufficient strength against shear. Eq. (2.14) ensures that the calculated shear stresses ( $f_v$ ) in members are smaller than the allowable shear stresses ( $F_v$ )

$$f_v \leq F_v = 0.40C_v F_y \quad (2.14)$$

where  $C_v$  is referred to as web shear coefficient calculated from Formulas G2-3, G2-4 and G2-5 in ANSI/AISC 360-05 (2005). Additionally the maximum slenderness ratio is limited to 300 and 200 for tension and compression members, respectively.

## 2.2 Discrete Sizing Optimization to AISC-LRFD

This section covers the design procedure based on the AISC-LRFD (1994) code. The design criteria for steel frame structures are described first; then in the next subsection the required amendments are given to employ the same procedure in case of truss structures. For steel frames, according to AISC-LRFD (1994) code of practice, the following design constraints ( $C_{IEL}^i$  and  $C_{IEL}^v$ ) must be satisfied for the strength requirements.

$$C_{IEL}^i = \left[ \frac{P_{uJ}}{\phi P_n} \right]_{IEL} + \frac{8}{9} \left( \frac{M_{uxJ}}{\phi_b M_{nx}} + \frac{M_{uyJ}}{\phi_b M_{ny}} \right)_{IEL} - 1 \leq 0 \quad \text{for} \quad \left[ \frac{P_{uJ}}{\phi P_n} \right]_{IEL} \geq 0.2 \quad (2.15)$$

$$C_{IEL}^i = \left[ \frac{P_{uJ}}{2\phi P_n} \right]_{IEL} + \left( \frac{M_{uxJ}}{\phi_b M_{nx}} + \frac{M_{uyJ}}{\phi_b M_{ny}} \right)_{IEL} - 1 \leq 0 \quad \text{for} \quad \left[ \frac{P_{uJ}}{\phi P_n} \right]_{IEL} < 0.2 \quad (2.16)$$

$$C_{IEL}^v = (V_{uJ})_{IEL} - (\phi_v V_n)_{IEL} \leq 0 \quad (2.17)$$

In Eqs. (2.15) to (2.17),  $IEL=1, 2, \dots, NEL$  is the element number,  $NEL$  is the total number of elements,  $J=1, 2, \dots, N$  is the load combination number and  $N$  is the total number of design load combinations.  $P_{uJ}$  is the required axial (tensile or compressive) strength, under  $J$ -th design load combination.  $M_{uxJ}$  and  $M_{uyJ}$  are the required flexural strengths for bending about  $x$  and  $y$ , under the  $J$ -th design load combination, respectively; where subscripts  $x$  and  $y$  are the relating symbols for strong and weak axes bending, respectively. On the other hand,  $P_n$ ,  $M_{nx}$  and  $M_{ny}$  are the nominal axial (tensile or compressive) and flexural (for bending about  $x$  and  $y$  axes) strengths of the  $IEL$ -th member under consideration.  $\phi$  is the resistance factor for axial strength, which is 0.85 for compression and 0.9 for tension (based on yielding in the gross section) and  $\phi_b$  is the resistance factor for flexure, which is equal to 0.9. Here, Eq. (2.17) is used for checking members' shear capacity wherein  $V_{uJ}$  is the required shear strength under  $J$ -th load combination and  $V_n$  is the nominal shear strength of the  $IEL$ -th member under consideration. In order to calculate the design shear strength the nominal shear strength is multiplied by a resistance factor  $\phi_v$  of 0.9.

In addition to the strength requirements, the serviceability criteria should be considered in the design process. The serviceability constraints ( $C_D^t$  and  $C_F^d$ ) considered in this research for frame structures are formulated as follows:

$$C_D^t = \Delta_{MaxJ} - \Delta_{Max}^a \leq 0 \quad (2.18)$$

$$C_F^d = [\delta_J]_F - [\delta^a]_F \leq 0 \quad (2.19)$$

Eq. (2.18) compares the maximum lateral displacement of the structure in the D-th direction (D=1, ..., ND) under J-th load combination  $\Delta_{MaxJ}$  with the maximum allowable lateral displacement  $\Delta_{Max}^a$ . Similarly, Eq. (2.19) checks the interstory drift of the F-th story (F=1, 2, ..., NF) under the J-th load combination  $[\delta_J]_F$  against the related permitted value  $[\delta^a]_F$ ; here NF is the total number of stories.

### 2.2.1 Nominal Strengths

Based on AISC-LRFD (1994) specification, the nominal tensile strength of a member, based on yielding in the gross cross section, is equal to:

$$P_n = F_y A_g \quad (2.20)$$

where  $F_y$  is the member's specified yield stress and  $A_g$  is the gross cross section of the member.

The nominal compressive strength of a member is the smallest value obtained from the limit states of flexural buckling, torsional buckling, and flexural-torsional buckling. For members with compact and/or non-compact elements, the nominal compressive strength of the member for the limit state of flexural buckling is as follows:

$$P_n = F_{cr} A_g \quad (2.21)$$

where  $F_{cr}$  is the critical stress based on flexural buckling of the member, calculated as:

$$\text{for } \lambda_c = \frac{Kl}{r\pi} \sqrt{\frac{F_y}{E}} \leq 1.5 \quad F_{cr} = (0.658^{\lambda_c^2}) F_y \quad (2.22)$$

$$\text{for } \lambda_c = \frac{Kl}{r\pi} \sqrt{\frac{F_y}{E}} > 1.5 \quad F_{cr} = \left[ \frac{0.877}{\lambda_c^2} \right] F_y \quad (2.23)$$

In the above equations,  $l$  is the laterally unbraced length of the member,  $K$  is the effective length factor,  $r$  is the governing radius of gyration about the axis of buckling and  $E$  is the modulus of elasticity.

The AISC-LRFD (1994) code addresses the nominal compressive strength based on the limit state of torsional and flexural-torsional buckling, for doubly symmetric members with compact and/or non-compact elements. For this limit state, Eq. (2.21) is still applicable with the following modifications:

$$\text{for } \lambda_e \leq 1.5 \quad F_{cr} = (0.658^{\lambda_e^2}) F_y \quad (2.24)$$

$$\text{for } \lambda_e > 1.5 \quad F_{cr} = \left[ \frac{0.877}{\lambda_e^2} \right] F_y \quad (2.25)$$

where

$$\lambda_e = \sqrt{F_y / F_e} \quad (2.26)$$

$$F_e = \left[ \frac{\pi^2 EC_w}{(K_z l_z)^2} + GJ \right] \frac{1}{I_x + I_y} \quad (2.27)$$

In Eq. (2.27),  $C_w$  is the warping constant,  $G$  is the shear modulus,  $J$  is the torsional constant,  $I_x$  and  $I_y$  are moments of inertia about principal axes,  $l_z$  is the unbraced

length for torsional buckling, and  $K_z$  is the effective length factor for torsional buckling. In this study  $K_z$  is conservatively taken as unity.

The nominal flexural strength of a member is the minimum value obtained according to the limit states of yielding, lateral-torsional buckling, flange local buckling, and web local buckling. The flexural capacity based on the limit state of yielding is as follows:

$$M_n = M_p = Z F_y \leq 1.5 S F_y \quad (2.28)$$

where  $Z$  is the plastic modulus and  $S$  is the section modulus of the member for the axis of bending. For doubly symmetric sections, the flexural capacity considering the limit state of lateral-torsional buckling is as follows:

$$M_n = \begin{cases} M_p & \text{if } L_b \leq L_p \\ C_b \left[ M_p - (M_p - M_r) \left( \frac{L_b - L_p}{L_r - L_p} \right) \right] \leq M_p & \text{if } L_p < L_b \leq L_r \\ M_{cr} \leq M_p & \text{if } L_b > L_r \end{cases} \quad (2.29)$$

where  $L_b$  is the laterally unbraced length of the member,  $L_p$  is the limiting laterally unbraced length for full plastic bending capacity,  $L_r$  is the limiting laterally unbraced length for inelastic lateral-torsional buckling,  $M_r$  is the limiting buckling moment, and  $M_{cr}$  is the critical elastic moment for the lateral-torsional buckling. The modification factor for non-uniform moment diagram,  $C_b$ , is defined by Eq. (2.30),

$$C_b = \frac{12.5 M_{\max}}{2.5 M_{\max} + 3 M_A + 4 M_B + 3 M_C} \quad (2.30)$$

where  $M_{\max}$ ,  $M_A$ ,  $M_B$ , and  $M_C$  are absolute values of maximum moment, moment at quarter point, centerline, and three-quarter point of the unbraced segment, respectively.

The nominal flexural strength of members with doubly symmetric sections and non-compact flanges, considering the limit state of flange local buckling, is given below:

$$M_n = \begin{cases} M_p & \text{if } \lambda_f \leq \lambda_p^f \\ M_p - (M_p - M_r^f) \left( \frac{\lambda_f - \lambda_p^f}{\lambda_r^f - \lambda_p^f} \right) & \text{if } \lambda_p^f < \lambda_f \leq \lambda_r^f \\ M_{cr}^f \leq M_p & \text{if } \lambda_f > \lambda_r^f \end{cases} \quad (2.31)$$

where  $\lambda_f$  is the flange slenderness parameter,  $\lambda_p^f$  is the limiting value of  $\lambda_f$  for full plastic bending capacity,  $\lambda_r^f$  is the limiting value of  $\lambda_f$  for inelastic flange local buckling,  $M_r^f$  is the limiting moment for flange buckling, and  $M_{cr}^f$  is the critical elastic moment for flange local buckling.

The nominal flexural strength of members with doubly symmetric sections and non-compact webs, considering the limit state of flange web buckling, is given below:

$$M_n = \begin{cases} M_p & \text{if } \lambda_w \leq \lambda_p^w \\ M_p - (M_p - M_r^w) \left( \frac{\lambda_w - \lambda_p^w}{\lambda_r^w - \lambda_p^w} \right) & \text{if } \lambda_p^w < \lambda_w \leq \lambda_r^w \end{cases} \quad (2.32)$$

where  $\lambda_w$  is the web slenderness parameter,  $\lambda_p^w$  is the limiting value of  $\lambda_w$  for full plastic bending capacity,  $\lambda_r^w$  is the limiting value of  $\lambda_w$  for inelastic web local buckling,  $M_r^w$  is the limiting moment for web buckling, and  $M_{cr}^w$  is the critical elastic moment for web local buckling.

The nominal shear strength of unstiffened webs of doubly symmetric members, subjected to shear in the plane of the web, is as follows:

$$\text{for } h/t_w \leq 418/\sqrt{F_{yw}} \quad V_n = 0.6F_{yw} A_w \quad (2.33)$$

$$\text{for } 418/\sqrt{F_{yw}} < h/t_w \leq 523/\sqrt{F_{yw}} \quad V_n = 0.6F_{yw} A_w (418/\sqrt{F_{yw}}) / (h/t_w) \quad (2.34)$$

$$\text{for } 523/\sqrt{F_{yw}} < h/t_w \leq 260 \quad V_n = 132000 A_w / (h/t_w)^2 \quad (2.35)$$

where  $h$  is the clear distance between flanges less the fillet or corner radius for rolled shapes,  $t_w$  is the web thickness,  $A_w$  is the shear area, and  $F_{yw}$  is the yield stress of the web in ksi; also  $V_n$  in Eq. (2.35) is in ksi. Here, to keep the original formulation of the code, Eqs. (2.33) to (2.35) are presented in British units. For members subjected to shear perpendicular to the plane of the web, the nominal shear strength is calculated through Eq. (2.33) as well.

### 2.2.2 Effective Length Factor K

In order to calculate the nominal compressive strength, the effective length factor,  $K$ , should be determined for each member. This factor can be computed using the frame buckling monograph developed by Jackson and Moreland as cited in McGuire (1968). For sway frames, the effective length factor for columns is computed as follows:

$$\frac{\alpha^2 G_i G_j - 36}{6(G_i + G_j)} = \frac{\alpha}{\tan \alpha} \quad (2.36)$$

$$G_i = \frac{\sum I_{ci}/l_{ci}}{\sum I_{bi}/l_{bi}}, \quad G_j = \frac{\sum I_{cj}/l_{cj}}{\sum I_{bj}/l_{bj}} \quad (2.37)$$

where  $\alpha = \pi/K$ ,  $i$  and  $j$  subscripts correspond to end- $i$  and end- $j$  of the compression member, and subscripts  $c$  and  $b$ , in building structures, refer to columns and beams connecting to the joint under consideration, respectively. Parameters  $I$  and  $l$  in the

above equations, represent the moment of inertia and unbraced length of the member, respectively. Here,  $K$  factor for beam, bracings and non-sway column elements is taken as 1. It is worth mentioning that in case of the examples solved before in the literature, calculation of the effective length factor,  $K$ , is performed using Eq. (2.13) for the sake of conformity. In case of the new instances Eqs. (2.36) and (2.37) are used.

### 2.2.3 Considerations for Steel Trusses

Since truss members carry only axial forces, the design criteria for truss structures can be considered as a simplified form of the aforementioned relations for frames. According to AISC-LRFD (1994) code, for each truss member,  $i$ , the following relation must be satisfied for the strength requirement.

$$\left[ \frac{P_u}{\phi P_n} \right]_i - 1 \leq 0 \quad (2.38)$$

In Eq. (2.38),  $P_u$  and  $P_n$  are the required and nominal axial (tensile or compressive) strengths of the  $i$ -th member under consideration, respectively. Here,  $\phi$  is the resistance factor for axial strength, which is 0.85 for compression and 0.9 for tension.

In addition to the strength requirements, the displacement criterion considered in this study for truss structures is formulated as follows:

$$\frac{d_{j,k}}{(d_{j,k})_{all}} - 1 \leq 0 \quad (2.39)$$

where  $j = 1, 2, \dots, N_j$  is the joint number,  $N_j$  is the total number of joints,  $d_{j,k}$ , and  $(d_{j,k})_{all}$ , are the displacements computed in the  $k$ -th direction of the  $j$ -th joint and its allowable value, respectively.



## CHAPTER 3

### REFORMULATIONS OF BIG BANG-BIG CRUNCH ALGORITHM

#### 3.1 Introduction

Big bang–big crunch (BB-BC) algorithm (Erol and Eksin 2006) is a novel metaheuristic optimization method based on the BB-BC theory of the universe evolution. The algorithm has become one of the most popular metaheuristics of the recent years because of its simple algorithmic structure and efficiency in solving real world design optimization problems. Different applications of the BB-BC algorithm in engineering design optimization problems have been reported in the literature so far. Afshar and Motaei (2011) used the BB-BC algorithm to find the optimal solution of large-scale reservoir operation problems. Tang et al. (2010) utilized the algorithm for parameter estimation in structural systems. An early work on application of the BB-BC algorithm in optimum design of skeletal structures was reported by Camp (2007). Later studies considering the use of the BB-BC algorithm in structural design optimization are due to Kaveh and Abbasgholiha (2011), Lamberti and Pappalettere (2011b), and Kaveh and Talatahari (2009b, 2009c, 2010a). Recently, Kazemzadeh Azad et al. (2011) evaluated the performance of the BB-BC algorithm in benchmark problems of engineering optimization and reported promising results.

This chapter presents a thorough investigation on the efficiency of BB-BC algorithm in discrete size optimum design of steel truss and frame structures. It is shown that a standard formulation of the BB-BC algorithm may sometimes fail to provide acceptable solutions to discrete sizing problems in structural optimization. The observed deficiencies of the algorithm are attributed to ineffective manipulation of the two search parameters; namely, search dimensionality and step size. Reformulations

of the BB-BC algorithm are then proposed, where the formula used by the standard algorithm for generating new candidate solutions around the center of mass is simplistically yet efficaciously reformulated, resulting in the so-called modified BB-BC (MBB-BC) and exponential BB-BC (EBB-BC) variants for truss and frame type structures, respectively. The performances of the proposed algorithms are experimented and compared to its standard version as well as some other metaheuristic techniques using several practical design examples. In these examples the steel trusses and frames are sized for minimum weight subject to stress, stability and displacement limitations according to the provisions of AISC-ASD (1989). The numerical results confirm the efficiency of the proposed approaches in practical design optimization of steel structures. In the following sections, the main steps involved in implementation of a standard BB-BC algorithm are outlined. The observed deficiencies of the standard BB-BC algorithm in discrete design optimization of steel structures are discussed and the proposed reformulations of BB-BC algorithm (i.e., MBB-BC and EBB-BC) are described in details. Finally the performance evaluations of the proposed algorithms through numerical examples are presented.

### **3.2 Standard BB-BC Algorithm**

The BB-BC optimization algorithm is based on continuous application of two successive stages, namely big bang and big crunch phases. During big bang phase, new solution candidates are randomly generated around a point called center of mass. This point is recalculated and updated every time in the big crunch phase with respect to the solution candidates generated. The main steps in the implementation of a standard BB-BC algorithm are outlined as follows.

**Step1. Initial Population:** Form an initial population by randomly spreading individuals (solutions) over all the search space (first big bang) in a uniform manner. This step is applied once.

**Step 2. Evaluation:** The initial population is evaluated, where structural analyses of all individuals are performed with the set of steel sections selected for design variables, and force and deformation responses are obtained under the loads. The objective function values of the feasible individuals that satisfy all problem constraints are directly calculated from Eq. (2.1). However, infeasible individuals that violate some of the problem constraints are penalized using an external penalty function approach, and their objective function values are calculated according to Eq. (3.1).

$$\phi = W \left[ 1 + p \left( \sum_i c_i \right) \right] \quad (3.1)$$

In Eq. (3.1),  $\phi$  is the penalized weight,  $c_i$  is the  $i$ -th design constraint and  $p$  is the penalty coefficient. The fitness scores of the individuals are then calculated by taking the inverse of their objective function values (i.e. fitness =  $1/W$  or  $1/\phi$  for feasible and infeasible solutions, respectively). The fitness scores are assigned as the mass values for the individuals.

**Step 3. Big Crunch Phase:** Calculate the “center of mass” by taking the weighted average using the coordinates (design variables) and the mass values of every single individual or choose the fittest individual among all as their center of mass. In the numerical investigations performed it is seen that selecting the center of mass based on the weighted average may lead to some convergence problems, increasing the computational time of the optimization dramatically. Hence, the latter approach is adopted in the present study.

**Step 4. Big Bang Phase:** Generate new individuals by using normal distribution (big bang phase). For a continuous variable optimization problem, Eq. (3.2) is used at each iteration to generate new solutions around the center of mass.

$$x_i^{new} = x_i^c + \alpha \cdot N(0,1)_i \frac{(x_i^{\max} - x_i^{\min})}{k} \quad (3.2)$$

where  $x_i^c$  is the value of  $i$ -th continuous design variable in the fittest individual,  $x_i^{\min}$  and  $x_i^{\max}$  are the lower and upper bounds on the value of  $i$ -th design variable, respectively,  $N(0,1)_i$  is a random number generated according to a standard normal distribution with mean ( $\mu$ ) zero and standard deviation ( $\sigma$ ) equal to one,  $k$  is the iteration number, and  $\alpha$  is a constant.

In this research, however, a discrete list of ready sections is used for sizing members of a steel structure. Hence, Eq. (3.3) is employed instead to round off the real values to nearest integers representing the sequence number of ready sections in a given section list

$$I_i^{new} = I_i^c + \text{round} \left[ \alpha \cdot N(0,1)_i \frac{(I_i^{\max} - I_i^{\min})}{k} \right] \quad (3.3)$$

where  $I_i^c$  is the value of  $i$ -th discrete design variable in the fittest individual, and  $I_i^{\min}$  and  $I_i^{\max}$  are its lower and upper bounds, respectively.

**Step 5. Elitism:** Keep the fittest individual found so far in a separate place or as a member of the population.

**Step 6. Termination:** Go to Step 2 until a stopping criterion is satisfied, which can be imposed as a maximum number of iterations or no improvement of the best design over a certain number of iterations.

### 3.3 Search Dimensionality and Step Size

Metaheuristic search techniques offer a general solution methodology for solving a wide range of optimization problems from different disciplines. On the other hand, each optimization problem has unique features of its own, and in most cases a problem-wise reformulation is necessary to achieve the best performance of the algorithm for a particular class of problems. In the following the observed deficiencies of the standard BB-BC algorithm in discrete design optimization of steel structures are discussed in detail. The poor performance of the standard algorithm is attributed to ineffective manipulations of the two search parameters; namely search dimensionality and step size.

Search dimensionality ( $SD$ ) is defined as the number of design variables that are perturbed to generate a new design through Eq. (3.3). Perhaps a more general term to quantify the degree of search dimensionality irrespective of problem size will be search dimensionality ratio ( $SDR$ ), which is computed by proportioning the number of perturbed design variables ( $N_p$ ) to the total number of design variables ( $N_d$ ) used in a problem, Eq. (3.4).

$$SDR = \frac{N_p}{N_d} \quad (3.4)$$

It may be expected that  $SDR$  will be different for each individual in the population, and the average search dimensionality ratio for a population,  $(SDR)_{ave}$ , is obtained by averaging  $SDR$  values of all the individuals, Eq. (3.5), where  $(SDR)_j$  is search dimensionality ratio for individual  $j$  and  $N_{pop}$  is the population size referring to the total number of individuals in the population.

$$(SDR)_{ave} = \frac{\sum (SDR)_j}{N_{pop}} \quad (3.5)$$

For continuous optimization problems  $(SDR)_{ave}$  will always have a value equal or close to 1.0, since all design variables -except those on the value bounds- are subjected to perturbation during generation of a new individual. That is to say an N-dimensional search is performed by the algorithm at any time during the search process. However, for discrete optimization problems some design variables will remain unchanged owing to the fact that the second term on the right hand side of Eq. (3.3) is rounded off to zero when the random number  $r_i$  generated by normal distribution is too small, which implicitly drive  $(SDR)_{ave}$  to low values especially when the iteration number  $k$  increases.

On the other hand, the step size for a single design variable is equal to  $I_i^{new} - I_i^c$ . Hence at any iteration one can define an average step size for an individual and the entire population as formulated in Eqs. (3.6) and (3.7)

$$(SS)_{j,ave} = \frac{\sum (I_i^{new} - I_i^c)_j}{N_d} \quad (3.6)$$

$$(SS)_{ave} = \frac{\sum (SS)_{j,ave}}{N_{pop}} \quad (3.7)$$

where  $(SS)_{j,ave}$  and  $(SS)_{ave}$  denote the average step size for  $j$ -th individual and entire population, respectively.

### 3.4 Numerical Investigations with BB-BC Algorithm

While using the BB-BC algorithm for discrete optimization, it is noted that both the average search dimensionality ratio  $(SDR)_{ave}$  and average step size  $(SS)_{ave}$  parameters will tend to approach zero after a certain number of iterations. Once this happens, no

new solutions are generated; namely the subsequent solutions simply replicate the former one, i.e. center of mass. As a remedy to this situation, the routine given below is integrated into the algorithm to make sure that a newly generated solution will differ from the former one at least by one variable.

```

Set  $\sigma := 1.0$ ;
Quitloop:= False;
Repeat
    Generate  $\mathbf{I}^{new}$  from  $\mathbf{I}^c$  using Eq. (3.3)
    If  $\mathbf{I}^{new} \neq \mathbf{I}^c$  then Quitloop := true;
        else  $\sigma := \sigma + 1.0$ ;
Until Quitloop;

```

When applying this routine, if all the design variables in a newly generated solution remain unchanged after applying Eq. (3.3), the generation process is iterated in the same way by increasing the standard deviation of normal distribution  $\sigma$  by one every time till a different design is produced. Apparently, the increased standard deviation facilitates occurrence of larger step sizes and increases probability of design variable change.

Typical results obtained from numerical investigations with the BB-BC algorithm on discrete sizing optimization problem are reflected in Figure 3.1, which shows the variations of  $(SDR)_{ave}$  and  $(SS)_{ave}$  parameters in the course of search process. It is noted that the average search dimensionality ratio is generally in the order of 0.9 in the first iterations, which results in extreme changes in the individuals. Although this helps provide a diverse population, this amount of diversity is more likely to result in convergence difficulties in case of discrete design optimization of truss structures. It is stated in Hasançebi (2008) that a useful starting value of  $(SDR)_{ave}$  will be in the range

of [0.25, 0.50] based on experiments with evolution strategies integrated search process. On the other hand, towards the later stages, only a single design variable is perturbed mostly to generate a new solution, implying that a unidirectional search is performed per design. Although a somewhat decreased search dimensionality towards the latest stage might be useful in the sense that it boosts more exploitative search in the design space, the search capability of the algorithm is significantly restricted when it is limited too much, as observed in the BB-BC algorithm.

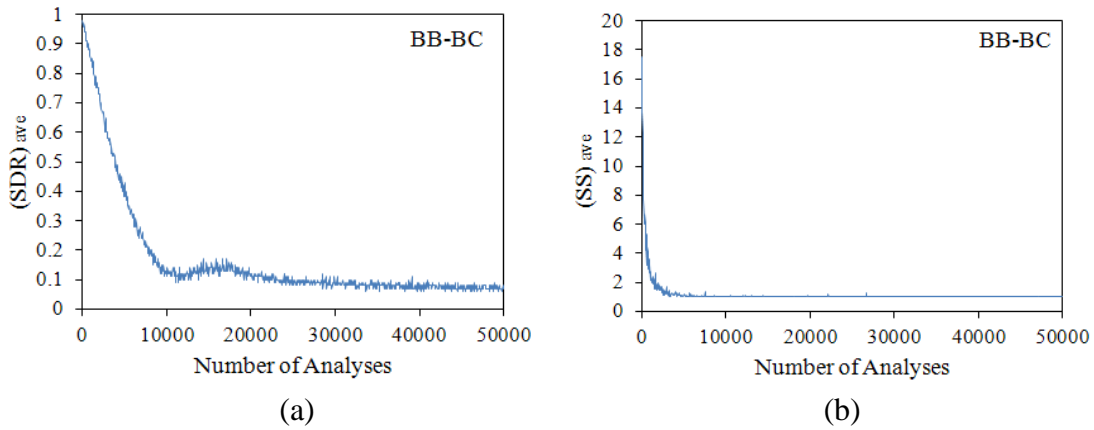


Figure 3.1: The variations of (a)  $(SDR)_{ave}$  and, (b)  $(SS)_{ave}$  in a typical run of BB-BC

### 3.5 Reformulations of BB-BC Algorithm

Noticing the drawbacks of the standard algorithm discussed above, Eq. (3.8) is proposed in lieu of Eq. (3.3) to improve the efficiency of the BB-BC algorithm in discrete structural design optimization. In the new formulation the use of  $n$ -th power ( $n \geq 2$ ) of a random number  $r_i$  is motivated using any appropriate statistical distribution.

$$I_i^{new} = I_i^c + \text{round} \left[ \alpha \cdot r_i^n \frac{(I_i^{\max} - I_i^{\min})}{k} \right] \quad (3.8)$$



The rationale behind Eq. (3.8) is to achieve a satisfactory trade-off or compromise between the following two conflicting requirements needed to eliminate the shortcomings of the standard formulation: (i) diminishing search dimensionality in the beginning of the search process and increasing it somewhat towards the latest stage and (ii) enabling large step size from time to time at later optimization stages to facilitate design transitions to new design regions and thereby preventing entrapment of the search in local optima. In fact, at times when the random number is sampled at values below 1, taking  $n$ -th power of  $r_i$  makes it even much smaller, which helps to fulfill the first requirement. On the other hand, at times when it is sampled at values above 1, it might be amplified to fairly large values by taking its  $n$ -th power, helping to satisfy the second requirement. It should be noted that for a general continuous optimization problem having a large dimensionality ratio is useful in the sense that it encourages exploration ability of an algorithm. However, a discrete structural optimization problem is the one such that even a small change in one design variable causes significant changes on the whole response of the structure and thus on the feasibility of a design generated. Accordingly, having a very large dimensionality ratio for such problems increases randomness of the search process, resulting in the slow convergence of the algorithm. It is important to limit the search dimensionality of the algorithm for these problems as far as the efficiency of the algorithm is concerned. By taking the  $n$ -th power of  $r$  in Eq. (3.8), the randomness of the algorithm is indeed diminished while higher step sizes are encouraged for large transitions individually at times when  $r$  is sampled above 1.

Two instances of Eq. (3.8) are generated in Eqs. (3.9) and (3.10) by selecting the type of statistical distribution used to sample the random number, where the power of random number  $n$  is set to 3 based on extensive numerical experiments.

$$I_i^{new} = I_i^c + \text{round} \left[ \alpha \cdot N(0,1)_i^3 \frac{(I_i^{\max} - I_i^{\min})}{k} \right] \quad (3.9)$$

$$I_i^{new} = I_i^c \pm \text{round} \left[ \alpha \cdot E(\lambda = 1)_i^3 \frac{(I_i^{\max} - I_i^{\min})}{k} \right] \quad (3.10)$$

Eq. (3.9) refers to the third power reformulation of big crunch phase according to a normally distributed random number. This reformulation will be referred to as modified BB-BC (MBB-BC) hereafter, and is introduced in relation to discrete design optimization of truss structures (Hasançebi and Kazemzadeh Azad 2014). The second reformulation, Eq. (3.10), referred to as the exponential BB-BC (EBB-BC), where the use of an exponential distribution in conjunction with the third power of random number is favored particularly when tackling problems from discrete design optimization of steel frames (Hasançebi and Kazemzadeh Azad 2012). The probability density function for an exponential distribution is given as follows:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (3.11)$$

where  $\lambda$  is a real, positive constant. The mean and variance of the exponential distribution are given as  $1/\lambda$  and  $1/\lambda^2$ , respectively. For various values of  $\lambda$ , the shape of exponential distribution is plotted in Figure 3.2. In this study, a standard exponential distribution is used by setting  $\lambda$  to one. It is important to note that unlike normal distribution which samples both positive and negative real numbers, exponential distribution only generates positive numbers. Hence, the rounded term on the right hand side of Eq. (3.10) should be added to or subtracted from  $I_i^c$  under equal probability to allow for both increase and decrease in the value of a design variable.

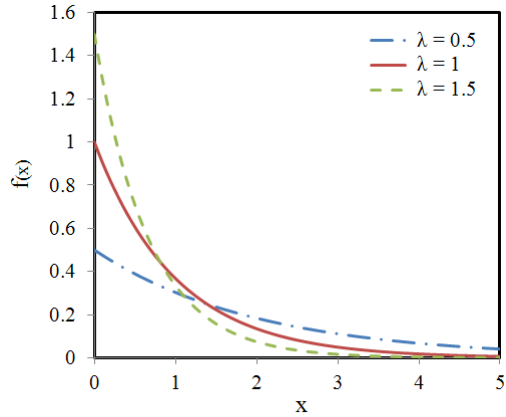


Figure 3.2: The exponential distribution for various values of  $\lambda$

When employing Eq. (3.10), the following routine is integrated into the EBB-BC algorithm to make sure that a new solution will differ from the former one at least by one variable.

```

Set  $\lambda := 1.0$ ;
Quitloop:= False;
Repeat
    Generate  $\mathbf{I}^{new}$  from  $\mathbf{I}^c$  using Eq. (3.10)
    If  $\mathbf{I}^{new} \neq \mathbf{I}^c$  then Quitloop := true;
    else  $\lambda := \lambda/2.0$ ;
Until Quitloop;

```

Accordingly, if all the design variables in a new solution remain unchanged after applying Eq. (3.10), i.e.  $\mathbf{I}^{new} = \mathbf{I}^c$ , the generation process is iterated in the same way by decreasing the  $\lambda$  parameter of the exponential distribution by half each time, and this is repeated until a different solution is produced, i.e.  $\mathbf{I}^{new} \neq \mathbf{I}^c$ . It is apparent from Figure 1 that the decreased value of  $\lambda$  parameter leads to a flatter probability distribution curve facilitating occurrence of larger changes.

Figures 3.3 and 3.4 are displayed to demonstrate the influence of the proposed reformulation on a BB-BC integrated search process. They show typical variations of  $(SDR)_{ave}$  and  $(SS)_{ave}$  parameters during a search captured while performing numerical investigations with the MBB-BC and EBB-BC algorithms, respectively. Figures 3.3a and 3.4a indicate that the starting value of average search dimensionality ratio in both MBB-BC and EBB-BC algorithms is around 0.60-0.70. Although this is slightly more than the upper limit of the recommended range, it leads to a more appropriate diversity in the population as compared to the standard algorithm, and provides a more suitable search mechanism in the initial iterations. As the iterations increase, the  $(SDR)_{ave}$  parameter is dragged to smaller values, implying that an explorative search is progressively replaced and dominated by an exploitative one. However, unlike the standard algorithm where a rapid and linear reduction is observed in  $(SDR)_{ave}$  towards unfavorably too low values, the reduction happens to be slower and more gradual in both the MBB-BC and EBB-BC algorithms. Besides, it is observed that  $(SDR)_{ave}$  is always kept at sufficiently high values in both MBB-BC and EBB-BC algorithms, which in turn prevents the search from becoming inefficient or restricted. The EBB-BC algorithm usually provides greater  $(SDR)_{ave}$  values in comparison to the MBB-BC. The rate of decrease of  $(SDR)_{ave}$  is slower and steadier in EBB-BC algorithm, whereas in the MBB-BC algorithm  $(SDR)_{ave}$  is brought down to its minimum value more rapidly and it is practically stabilized around this minimum value thereafter.

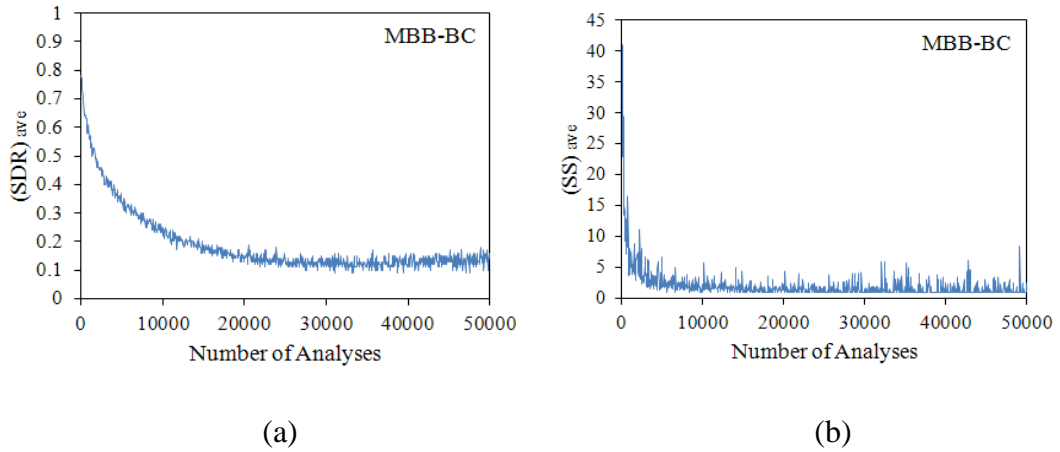


Figure 3.3: The variations of (a)  $(SDR)_{ave}$  and, (b)  $(SS)_{ave}$  in a typical run of MBB-BC

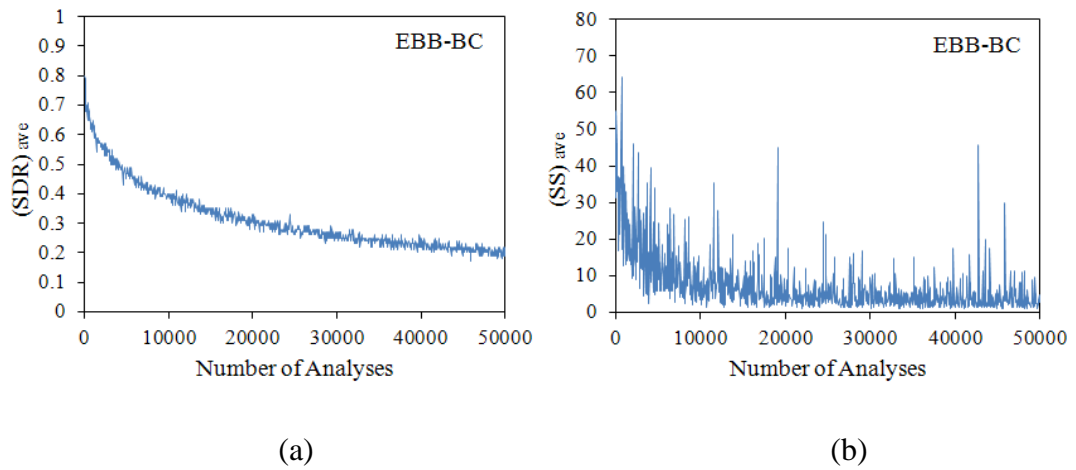


Figure 3.4: The variations of (a)  $(SDR)_{ave}$  and, (b)  $(SS)_{ave}$  in a typical run of EBB-BC

Figures 3.3b and 3.4b show that both MBB-BC and EBB-BC algorithms accommodate fairly larger step sizes as compared to the standard algorithm. As the search process goes on, while  $(SS)_{ave}$  parameter is strictly reduced to one in the standard algorithm, it takes place in the range of  $[2, 8]$  for MBB-BC and of  $[2, 40]$  for EBB-BC even in the latest iterations of the optimization. These occasional large step sizes are quite useful for steering the search towards new design regions when the search gets stuck in local optima. This characteristic of the proposed reformulation

provide an efficient mechanism to avoid local optima while the standard algorithm is likely to get trapped in local optima.

### **3.6 Numerical Examples**

This section covers performance evaluation of developed MBB-BC and EBB-BC design optimization algorithms in discrete sizing optimization of truss and frame structures. For the sake of clarity the numerical investigations of MBB-BC and EBB-BC algorithms are provided in two distinct sections.

#### **3.6.1 Performance Evaluation of MBB-BC Algorithm**

This section analyzes the performance of the proposed MBB-BC algorithm that is applied to four different discrete truss optimization problems: (i) 160-bar space pyramid; (ii) 354-bar braced dome; (iii) 693-bar braced barrel vault; (iv) 960-bar double layer grid. In these test problems, the structures are optimized for minimum weight by selecting members either from the whole or a certain portion of a profile list consisting of 37 pipe (circular hollow) sections issued in AISC-ASD (1989) standard section tables, which is reproduced in Table 3.1 for convenience. The test problems, which are challenging real-size optimization problems with the number of sizing design variables ranging between 7 and 251, are all selected from earlier studies available in the literature (Hasançebi et al. 2009, Hasançebi et al. 2011b). The idea here is to eliminate any kind of factor, such as the starting point of optimization, the number of independent runs performed, the way of constraint handling, additional strategies for reducing the design space, etc., that may have an impact on the performance of an algorithm in terms of both convergence velocity and accuracy. The results obtained to these problems using the proposed methodology are compared to those of the standard BB-BC and other metaheuristic methods, such as simulated annealing (SA), evolution strategies (ESs), particle swarm optimization (PSO), ant colony optimization (ACO), tabu search (TS), harmony search (HS) and simple genetic algorithms (SGA) using the algorithms developed for them in Hasançebi et al.

(2009). It is important to mention that verifications of these algorithms are conducted using benchmark problems of engineering and structural optimization in Kazemzadeh Azad et al. (2011), and Hasançebi et al. (2009). All these algorithms share the same integrated programming unit for defining optimum design problem (i.e., objective function, design variables, design set, constraints, penalization, etc.), which in turn provides an ideal and unbiased platform for comparison of different techniques.

For a fair comparison of results, the maximum number of structural analyses in each example is limited to the values formerly reported in the literature, which are 50,000 for the examples 1, 2 and 3, and 100,000 for the example 4. However, in cases where no progress in the best feasible design is recorded over a certain number of successive iterations, the search process is terminated before the maximum number of structural analyses is reached. For each example the actual number of structural analyses performed to attain the optimum design by the proposed MBB-BC algorithm is reported in Table 3.2, including the ratio of design points searched during the optimization process to the total size of the design space.

For all test problems the parameter  $\alpha$  in Eqs. (3.3) and (3.9) is taken as 0.5 and a population size of 50 is used for both the MBB-BC and BB-BC algorithms. It is worth mentioning that large values of parameter  $\alpha$  results in larger  $(SDR)_{ave}$  and  $(SS)_{ave}$  values which consequently increases the exploration properties of algorithms. Conversely, low values of  $\alpha$  limit the  $(SDR)_{ave}$  and  $(SS)_{ave}$  to lower values which results in more exploitation of algorithms. As mentioned before in this study the value of parameter  $\alpha$  is determined based on numerous numerical investigations. The material properties of steel used for all test problems are as follows: modulus of elasticity ( $E$ ) = 29,000 ksi (199,948 MPa) and yield stress ( $F_y$ ) = 36 ksi (248.2 MPa).

### **3.6.1.1 Example 1: 160-Bar Space Pyramid**

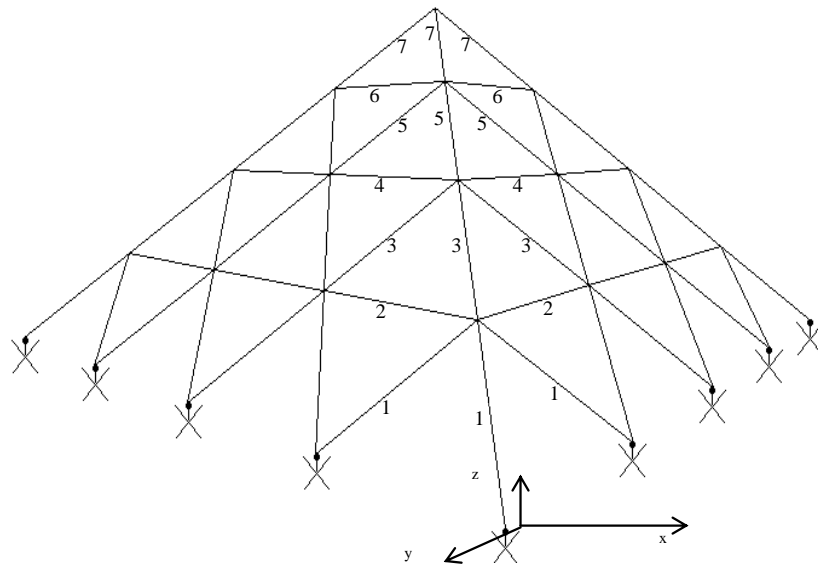
The first test problem considered is a 160-bar space steel pyramid (Figure 3.5) with a square base diameter of 16 m (52.5 ft) along both the x and y axes and a total height

of 8 m (26.25 ft). This problem was studied in Hasaebi et al. (2011b) using PSO and HS techniques as well as the three variants of ACO method referred to as sACO, rACO and eACO therein. The structure is composed of 55 joints and 160 members that are grouped into 7 independent sizing design variables. The grouping of members is shown in Figure 3.5a. The sizing variables are selected from the entire database of 37 standard pipe (circular hollow) sections in Table 3.1. The stress and stability limitations of the members are computed according to the specifications of AISC-ASD (1989). The displacements of all nodes are limited to 4.45 cm (1.75 in) in all the x, y, and z directions. For design purpose, a single load case is considered such that it consists of a vertical load of  $-8.53$  kN ( $-1.92$  kips) applied in the z-direction at all nodes of the pyramid.

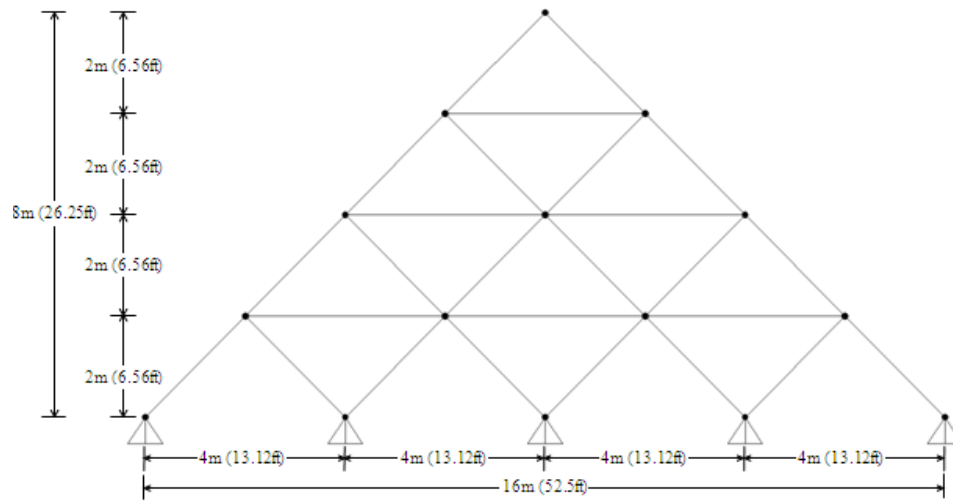
Sizing optimization of 160-bar space pyramid is carried out using both the BB-BC and MBB-BC algorithms and the results obtained are compared to the previously reported solutions by Hasaebi et al. (2011b) in Table 3.3. It should be noted that all the solutions reported in Table 3.3 as well as in the similar tables of the following examples correspond to only the best feasible designs attained by the techniques, and hence any kind of constraint violation is not expected in these solutions. Similar to the results of sACO, PSO and HS techniques, the MBB-BC algorithm yields an identical design weight of 2788.84kg (6148.35 lb) for the truss, which is the best known solution of the problem so far. The final designs attained using BB-BC, rACO, and eACO techniques are slightly heavier; namely 2821.27 kg (6219.83 lb), 2817.56 (6211.65 lb) and 2875.01 kg (6338.31 lb), respectively.

The variations of the best feasible design obtained thus far in the search processes with BB-BC and MBB-BC algorithms are plotted in Figure 3.6. Figure 3.7 shows variations of  $(SDR)_{ave}$  and  $(SS)_{ave}$  parameters in the implementations of the two algorithms. This figure clearly indicates how the aforementioned problems associated with implementation of these two parameters in the BB-BC algorithm are surmounted with the MBB-BC algorithm.



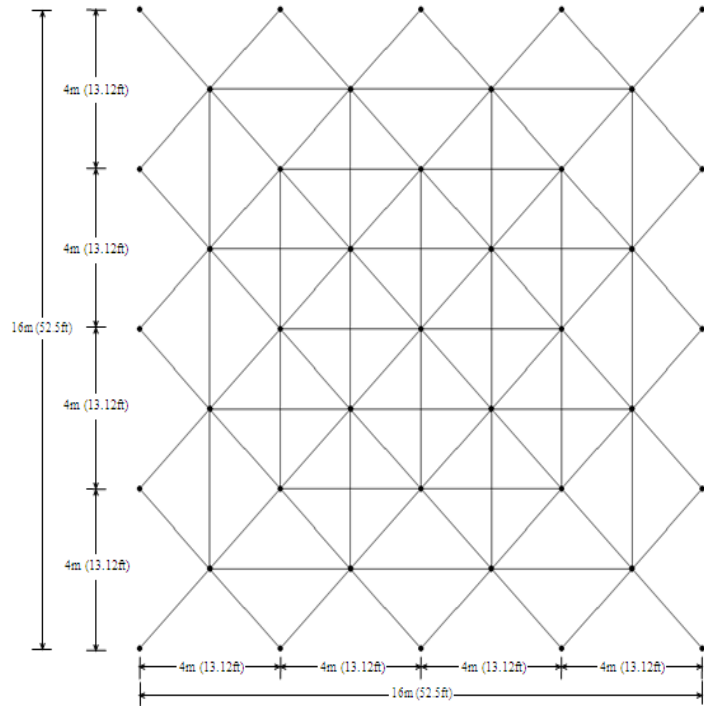


(a) 3-D view



(b) Front view

Figure 3.5: 160-bar pyramid; (a) 3-D view, (b) Front view, (c) Plan view  
(Hasançebi et al. 2011b)



(c) Plan view

Figure 3.5 (continued)

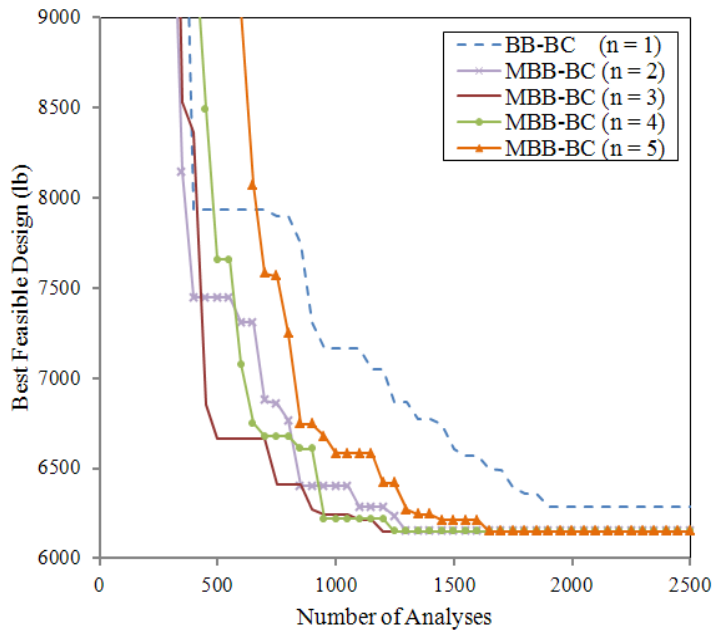


Figure 3.6: Optimization histories for 160-bar space pyramid

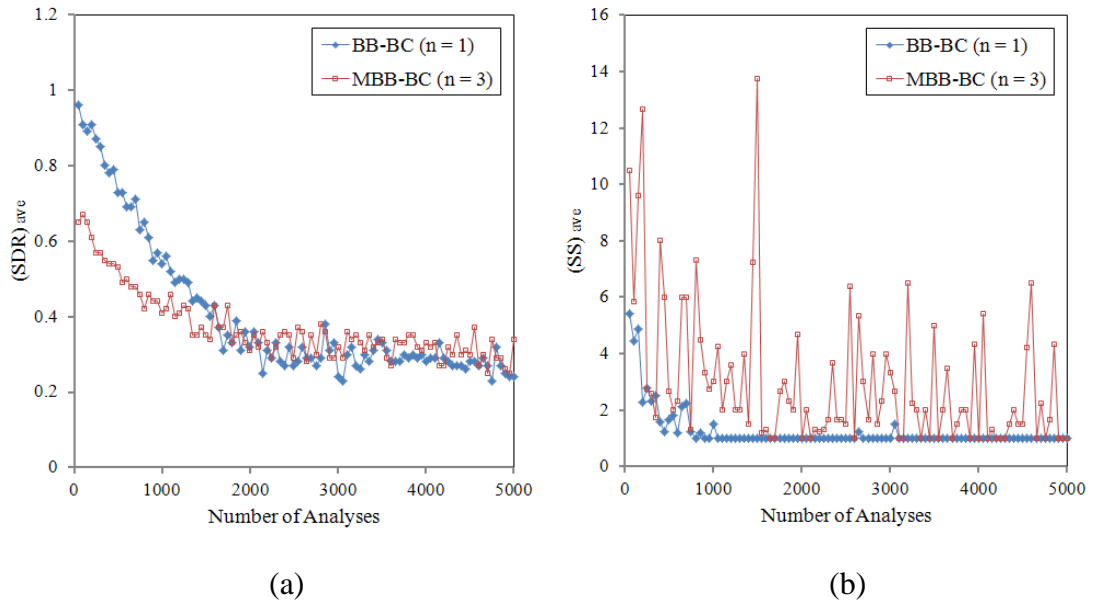


Figure 3.7: 160-bar pyramid test problem; variations of (a)  $(SDR)_{ave}$  and, (b)  $(SS)_{ave}$

Table 3.1. Cross sectional properties of the ready sections

Section number	Ready section	Area , cm <sup>2</sup> (in <sup>2</sup> )	Section number	Ready section	Area , cm <sup>2</sup> (in <sup>2</sup> )
1	P1	3.187 (0.494)	20	PX5	39.419 (6.11)
2	P2	6.903 (1.07)	21	PX6	54.193 (8.4)
3	P3	14.387 (2.23)	22	PX8	82.581 (12.8)
4	P4	20.452 (3.17)	23	PX10	103.871 (16.1)
5	P5	27.472 (4.3)	24	PX12	123.871 (19.2)
6	P6	36 (5.58)	25	PX1.5	6.903 (1.07)
7	P8	54.193 (8.4)	26	PX2.5	14.516 (2.25)
8	P10	76.774 (11.9)	27	PX3.5	23.742 (3.68)
9	P12	94.193 (14.6)	28	PX1.25	5.684 (0.881)
10	P1.5	5.155 (0.799)	29	PX.5	2.065 (0.32)
11	P2.5	10.968 (1.7)	30	PX.75	2.794 (0.433)
12	P3.5	17.290 (2.68)	31	PXX2	17.161 (2.66)
13	P1.25	4.3161 (0.669)	32	PXX3	35.290 (5.47)
14	P.5	1.613 (0.25)	33	PXX4	52.258 (8.1)
15	P.75	2.148 (0.333)	34	PXX5	72.903 (11.3)

Table 3.1. (continued)

16	PX1	4.123 (0.639)	35	PXX6	100.645 (15.6)
17	PX2	9.548 (1.48)	36	PXX8	137.419 (21.3)
18	PX3	19.484 (3.02)	37	PXX2.5	26 (4.03)
19	PX4	28.452 (4.41)			

Table 3.2. Number of analyses and size of the design space for the test problems

Test problem	Number of analyses	Design space size	Portion of the design space searched
160-bar space pyramid	5000	$37^7$	$5.27 \times 10^{-8}$
354-bar braced dome	16350	$37^{22}$	$5.17 \times 10^{-31}$
693-bar braced barrel vault	27150	$37^{23}$	$2.32 \times 10^{-32}$
960-bar double layer grid	100000	$28^{251}$	$100000/28^{251}$

Table 3.3. Comparison of optimization results for 160-bar space pyramid

Sizing variables	Optimal cross sectional areas (in <sup>2</sup> )						
	sACO*	rACO *	eACO *	PSO *	HS *	BB-BC (n=1)	MBB-BC (n=2,3,4,5)
1	1.07	1.07	1.07	1.07	1.07	1.07	1.07
2	0.669	0.669	0.669	0.669	0.669	0.669	0.669
3	1.07	1.07	1.07	1.07	1.07	1.07	1.07
4	0.669	0.669	0.799	0.669	0.669	0.669	0.669
5	1.07	1.07	1.07	1.07	1.07	1.07	1.07
6	0.669	0.669	0.799	0.669	0.669	1.07	0.669
7	1.07	1.48	1.7	1.07	1.07	1.07	1.07
Weight, lb (kg)	6148.35 (2788.84)	6211.65 (2817.56)	6338.31 (2875.01)	6148.35 (2788.84)	6148.35 (2788.84)	6219.83 (2821.27)	6148.35 (2788.84)

\* The algorithms are provided in Hasançebi et al. (2011b).

### 3.6.1.2 Example 2: 354-Bar Braced Dome Truss

The second test problem is a 354-bar braced dome truss shown in Figure 3.8 (Hasançebi et al. 2009). The structural members are grouped into 22 discrete sizing variables (Figure 3.8), which are selected from the entire set of 37 standard sections presented in Table 3.1. The dome is subjected to the following three load cases considering various combinations of dead (D), snow (S) and wind (W) loads computed according to the specifications of ASCE (1998): (i) D + S, (ii) D + S + W (with negative internal pressure), and (iii) D + S + W (with positive internal pressure). Figure 3.9 presents the considered three load cases. The complete details of load calculations and others for this instance can be found in Hasançebi et al. (2009, 2010c). The stress and stability constraints of the members are computed regarding the specifications of AISC-ASD (1989). The displacements of nodes are restricted to 11.1 cm in any direction. It is worth mentioning that an overall structural stability check or constraint is not explicitly implemented in this example. There are two main reasons for this. Firstly, even though the overall structural stability check is not explicitly implemented, the displacement constraints induced at every joint of the structure implicitly prevent the occurrences of large displacements and hence the possibility of a failure such as a snap-through buckling. In fact, when the final optimum design is analyzed it can be seen that the displacement constraints are all inactive; in fact the actual displacements are limited to much smaller values than even the limits imposed by codes. One can see that a related failure such as a snap-through buckling cannot occur under such small displacements. Secondly, a detailed analysis of snap-through buckling requires a detailed finite element analysis that accounts for geometric non-linearity, which is out of the scope of this study.

In Table 3.4 the minimum weight designs of 354-bar braced dome obtained by the BB-BC and MBB-BC algorithms are compared to the previously reported results by Hasançebi et al. (2009) with different metaheuristic techniques. It is noted that the results published formerly by Hasançebi et al. (2009) are corrected here due to mis-grouping of one bracing member in the previous work.

The MBB-BC algorithm with  $n=3, 4,$  and  $5$  powers produces a design weight of  $14775.7$  kg ( $32574.9$  lb) for the dome, which is the best solution of the problem reported so far and which has been attained using SA, ESs and PSO methods as well. Relatively higher design weights have been attained for the structure with other metaheuristic algorithms: respectively,  $15221.4$  kg ( $33557.5$  lb) by ACO,  $15251.16$  kg ( $33623.06$  lb) by MBB-BC with a power of  $n=2,$   $16043.6$  kg ( $35370.1$  lb) by TS,  $15850.5$  kg ( $34944.3$  lb) by HS and  $16485$  kg ( $36343.3$  lb) by SGA. The BB-BC algorithm shows a very poor performance and produces a final design weight of  $18784.8$  kg ( $41413.5$  lb). Such a significant difference between the results clearly indicates usefulness of the proposed reformulation on the performance of the BB-BC algorithm.

The variations of the best feasible design obtained so far in the search processes with BB-BC and MBB-BC algorithms are plotted in Figure 3.10. Considering the design weights as well as the convergence histories attained for 354-bar braced dome and 160-Bar space pyramid examples using five different values of power  $n$  (i.e.  $n=1, 2, 3, 4,$  and  $5$ ), it is observed that adopting the value of  $n=3$  can be reasonable. Hence, the next design examples are investigated using the third power formulation.

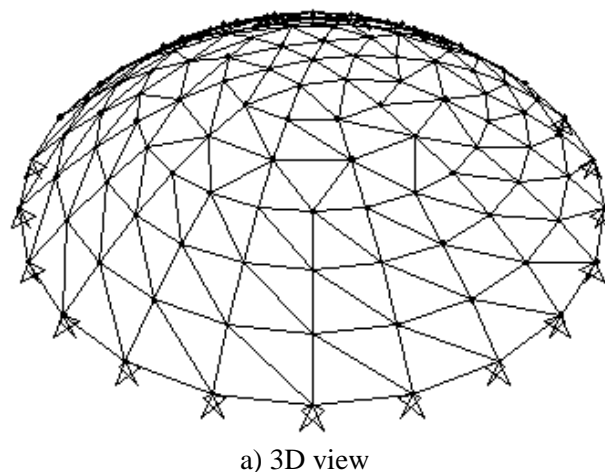
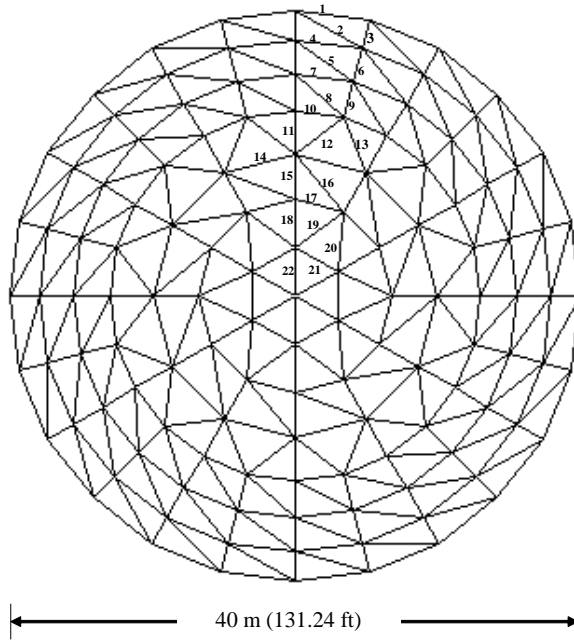
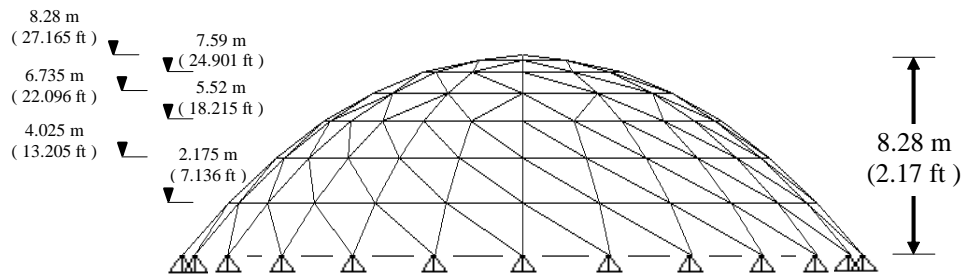


Figure 3.8: 354-bar braced truss dome; (a) 3D view, (b) top view and (c) side view (Hasançebi et al. 2009)



b) top view



c) side view

Figure 3.8 (continued)

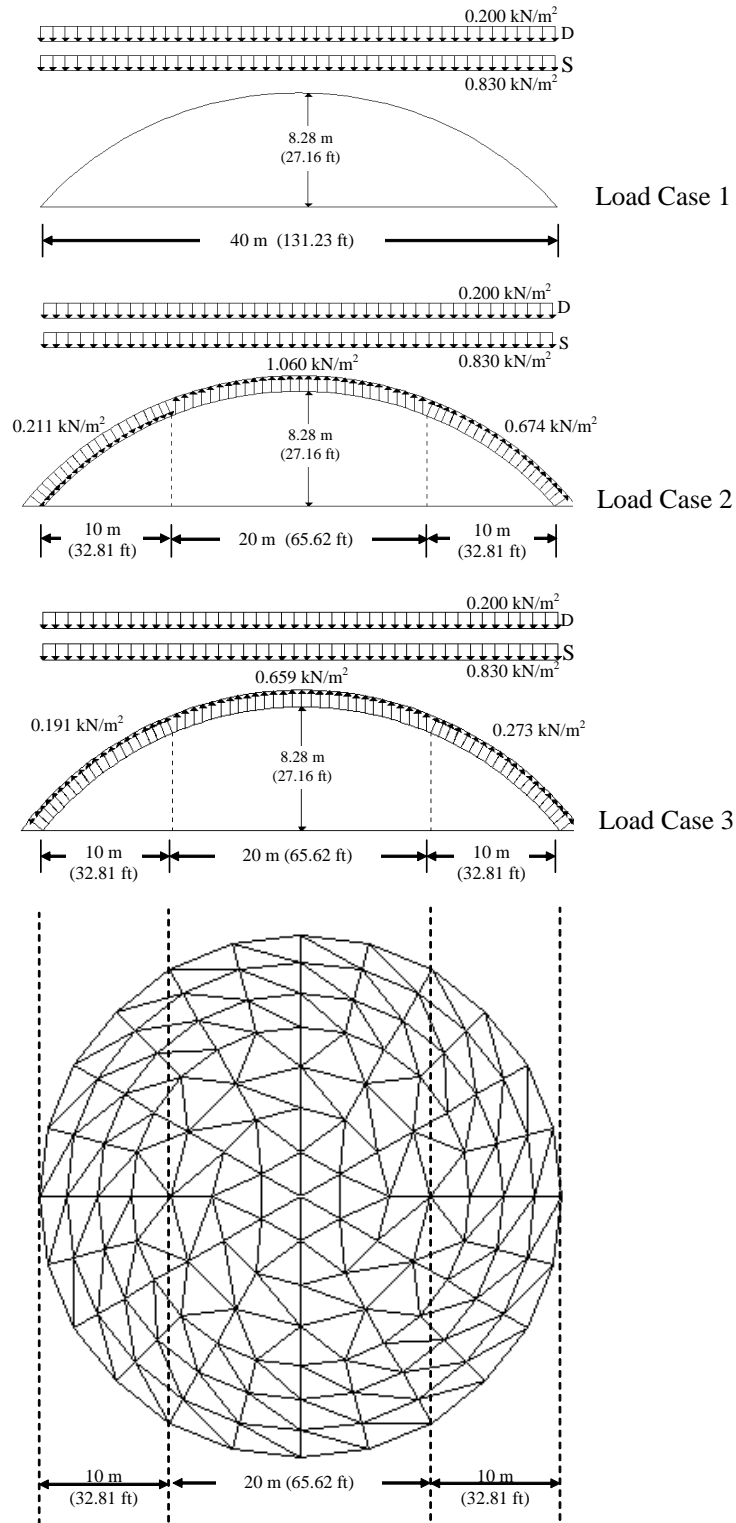


Figure 3.9: The three load cases considered for 354-bar braced truss dome (Hasançebi et al. 2009)



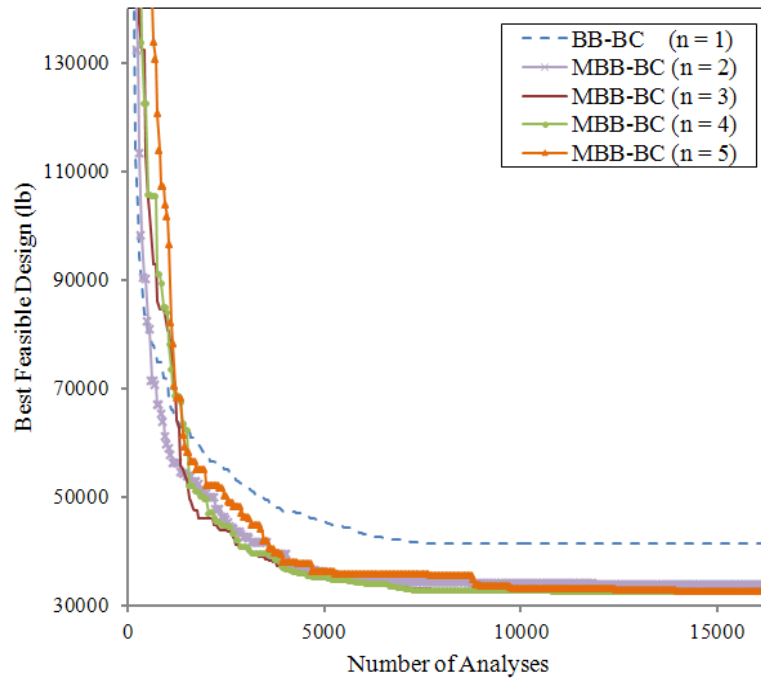


Figure 3.10: Optimization histories for 354-bar braced dome

Table 3.4. Comparison of optimization results for 354-bar braced dome

Sizing variables	Optimal cross sectional areas (in <sup>2</sup> )									
	SA *	ESs *	PSO *	ACO *	TS *	HS *	SGA*	BB-BC (n=1)	MBB-BC (n=2)	MBB-BC (n=3,4,5)
1	1.07	1.07	1.07	1.07	1.07	1.07	1.48	1.07	1.07	1.07
2	3.17	3.17	3.17	3.17	3.17	3.17	2.68	3.17	3.17	3.17
3	2.23	2.23	2.23	2.23	2.68	2.23	4.3	2.68	2.68	2.23
4	2.68	2.68	2.68	2.68	2.68	2.68	2.68	4.3	2.68	2.68
5	2.23	2.23	2.23	2.23	2.68	2.23	2.23	2.23	2.23	2.23
6	2.23	2.23	2.23	2.23	2.23	2.25	2.23	2.68	2.23	2.23
7	2.23	2.23	2.23	2.23	2.68	2.23	2.23	2.68	2.23	2.23
8	2.23	2.23	2.23	2.23	2.68	2.68	2.23	4.3	2.23	2.23
9	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7
10	2.23	2.23	2.23	2.23	2.23	2.25	2.25	2.68	2.23	2.23
11	1.7	1.7	1.7	2.66	1.7	2.25	2.66	1.7	1.7	1.7
12	1.7	1.7	1.7	2.23	1.7	1.7	2.23	2.68	1.7	1.7
13	1.7	1.7	1.7	1.7	1.7	1.7	2.23	2.68	1.7	1.7
14	1.7	1.7	1.7	1.7	1.7	2.23	1.7	1.7	1.7	1.7
15	1.7	1.7	1.7	1.7	1.7	1.7	2.25	1.7	1.7	1.7

Table 3.4. (continued)

16	1.7	1.7	1.7	1.7	2.68	1.7	1.7	2.68	2.68	1.7
17	1.48	1.48	1.48	1.48	1.7	2.66	1.7	1.48	1.48	1.48
18	1.48	1.48	1.48	2.68	1.48	3.02	3.17	1.48	1.48	1.48
19	1.07	1.07	1.07	1.07	1.07	1.7	1.48	2.68	1.07	1.07
20	1.07	1.07	1.07	1.07	1.07	1.7	1.48	2.68	1.07	1.07
21	1.07	1.07	1.07	1.07	1.07	1.7	1.48	1.48	1.07	1.07
22	1.07	1.07	1.07	1.48	1.07	1.07	1.7	1.48	1.07	1.07
Weight, lb (kg)	32574.9 (14775.7)	32574.9 (14775.7)	32574.9 (14775.7)	33557.5 (15221.4)	35370.1 (16043.6)	34944.3 (15850.5)	36343.3 (16485)	41413.5 (18784.8)	33623.06 (15251.16)	32574.9 (14775.7)

\* The algorithms are provided in Hasańcebi et al. (2009).

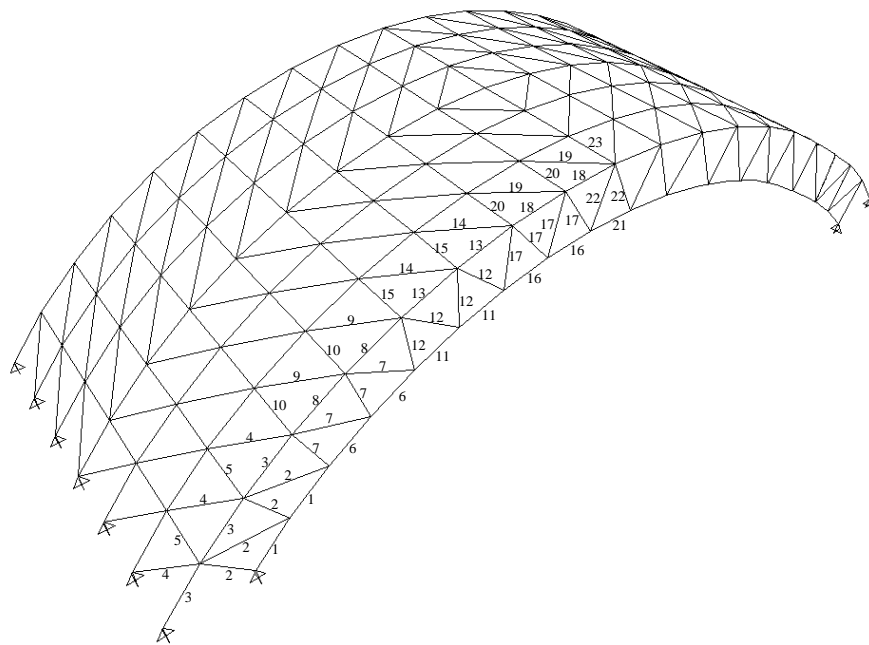
### 3.6.1.3 Example 3: 693-Bar Braced Barrel Vault

The third test problem depicted in Figure 3.11 is a spatial braced barrel vault (Ramaswamy et al. 2002) composed of 259 joints and 693 members which are grouped into 23 discrete sizing variables due to the symmetry of the structure about the centerline. The member grouping scheme is outlined in Figure 3.11a and the main geometric dimensions of the structure are shown in Figures 3.11b and 3.11c. The structure is subjected to a uniform dead load (DL) pressure of  $35 \text{ kg/m}^2$ , a positive wind load (WL) pressure of  $160 \text{ kg/m}^2$  ( $32.77 \text{ lb/ft}^2$ ) and a negative wind load (WL) pressure of  $240 \text{ kg/m}^2$  ( $49.16 \text{ lb/ft}^2$ ). Here, these loads are combined under two separate load cases as follows: (i)  $1.5\text{DL}+1.5\text{WL}$ , and (ii)  $1.5\text{DL}-1.5\text{WL}$ . The displacements of joints in all the x, y, and z directions are restricted to a maximum value of 0.254 cm. Further details about this design optimization instance can be found in Hasańcebi et al. (2011b).

In Table 3.5 the design optimization results of 693-bar barrel vault obtained using the MBB-BC and BB-BC algorithms are compared to the previously reported results by Hasańcebi et al. (2011b) with different metaheuristic techniques. According to these results, the best solution is attained by the MBB-BC algorithm, which is 4805.96 kg (10595.33 lb). In this example, the BB-BC algorithm shows a more promising performance compared to the other metaheuristic techniques and yields the second

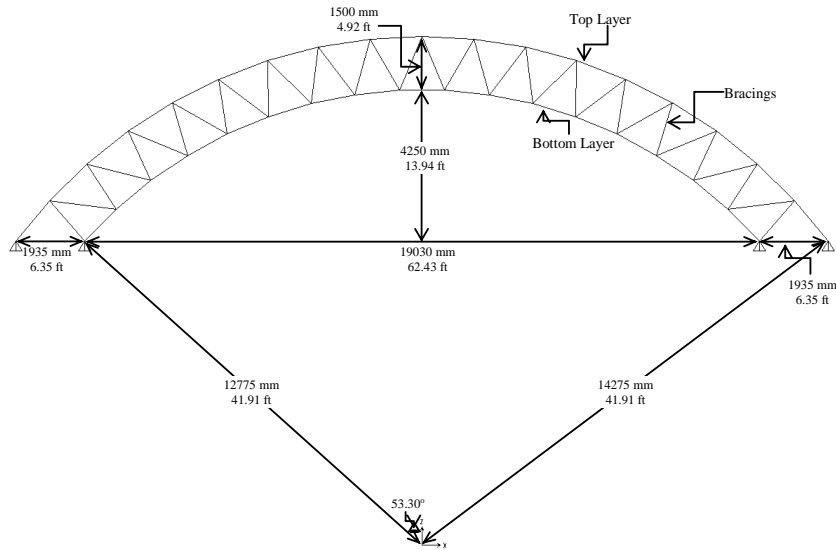
best solution, which is 4925.75 kg (10859.42 lb). It is noted that unlike the first two design examples, displacement constraints are active in this example. Amongst the other solutions are 4989.15 kg (10999.20 lb) by sACO, 5095.07 kg (11232.71 lb) by HS, 5456.48 kg (12029.49 lb) by SGA, 5503.65 kg (12133.47 lb) by rACO and 6068.69 kg (13379.19 lb) by eACO.

The variations of the best feasible design obtained so far in the search processes with BB-BC and MBB-BC algorithms are plotted in Figure 3.12. Figure 3.13 shows the variations of  $(SDR)_{ave}$  and  $(SS)_{ave}$  parameters in the implementations of the two algorithms.

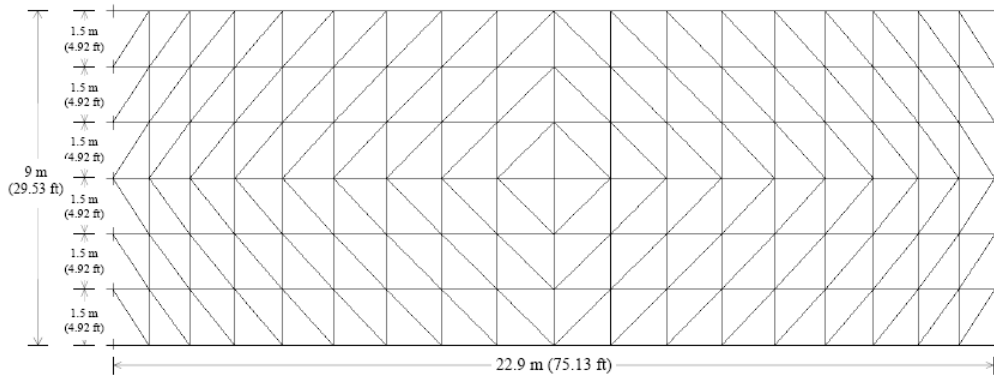


a) 3-D view

Figure 3.11: 693-bar braced barrel vault; a) 3-D view, b) Front view, c) Plan view (Hasançebi et al. 2011b)



b) Front view



c) Plan view

Figure 3.11 (continued)

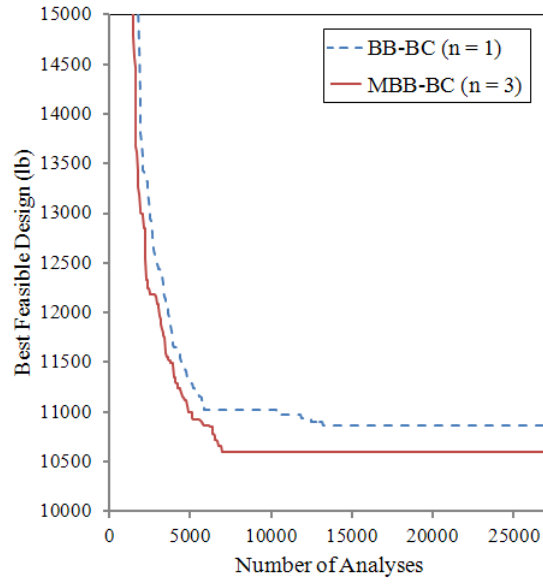


Figure 3.12: Optimization histories for 693-bar braced barrel vault

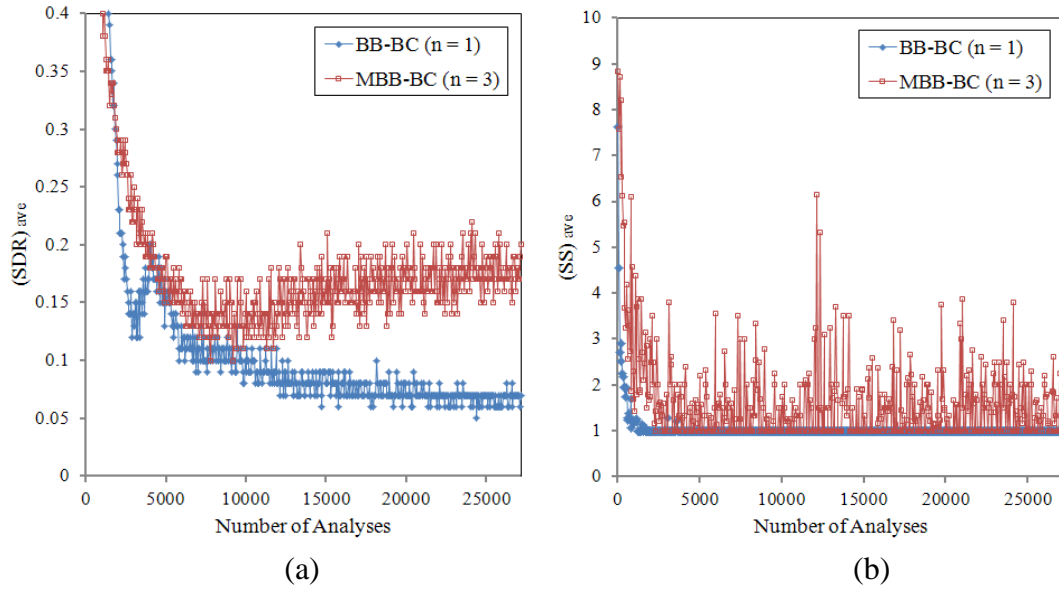


Figure 3.13: 693-bar barrel vault test problem; variations of (a)  $(SDR)_{ave}$  and, (b)  $(SS)_{ave}$

Table 3.5. Comparison of optimization results for 693-bar braced barrel vault

Sizing variables	Optimal cross sectional areas (in <sup>2</sup> )						
	sACO *	rACO *	eACO *	HS *	GA *	BB-BC (n=1)	MBB-BC (n=3)
1	4.03	3.17	2.23	3.68	3.02	4.03	3.68
2	0.494	0.494	1.07	0.433	0.669	0.494	0.494
3	0.494	0.669	0.494	0.494	0.639	0.333	0.333
4	0.494	0.881	0.881	0.494	0.494	0.669	0.494
5	0.494	0.333	0.669	0.433	0.333	0.333	0.333
6	0.333	4.3	4.41	3.17	4.41	3.68	3.68
7	0.639	0.494	0.669	0.669	0.639	0.494	0.494
8	0.333	0.639	1.07	0.333	0.333	0.494	0.494
9	2.68	2.23	2.66	2.68	2.66	0.494	0.494
10	4.03	0.494	0.881	0.494	0.639	0.333	0.333
11	0.494	0.669	0.494	0.669	0.669	2.25	2.23
12	0.639	0.799	0.639	0.881	0.799	0.799	0.799
13	0.881	0.799	0.881	1.07	1.07	1.07	1.07
14	0.639	0.494	1.48	0.881	0.799	0.494	0.494
15	0.333	0.433	0.333	0.333	0.494	0.333	0.333
16	0.639	0.799	0.799	0.881	0.669	1.07	0.881
17	0.881	1.48	1.7	0.881	1.07	0.669	0.669
18	0.494	0.669	0.669	0.669	0.799	1.7	2.23
19	0.669	0.494	0.799	0.639	0.669	0.669	0.494
20	0.333	0.333	1.07	0.333	0.333	0.333	0.333
21	1.7	2.25	3.17	1.7	2.23	0.494	0.494
22	0.669	0.799	0.494	0.494	0.669	0.494	0.333
23	0.494	0.333	0.433	0.639	0.433	0.333	0.333
Weight, lb (kg)	10999.20 (4989.15)	12133.47 (5503.65)	13379.19 (6068.69)	11232.71 (5095.07)	12029.49 (5456.48)	10859.42 (4925.75)	10595.33 (4805.96)

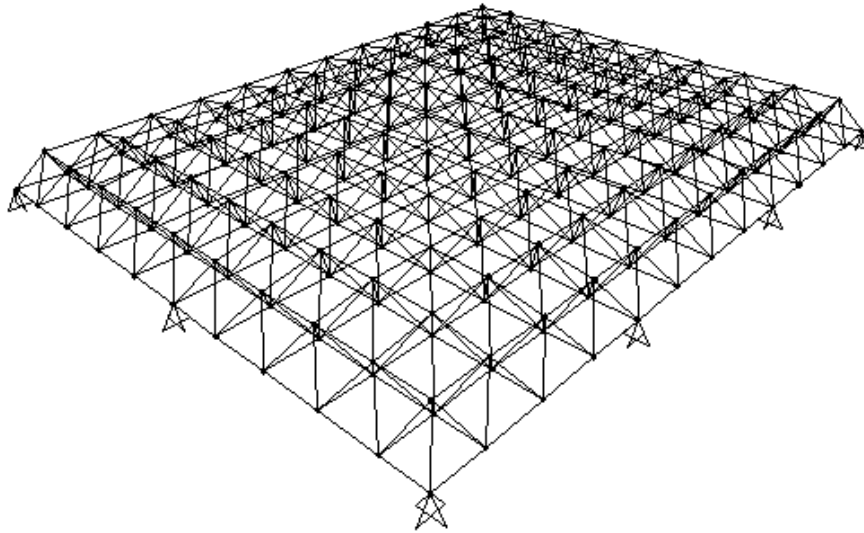
\* The algorithms are provided in Hasaebi et al. (2011b).

### 3.6.1.4 Example 4: 960-Bar Double Layer Grid

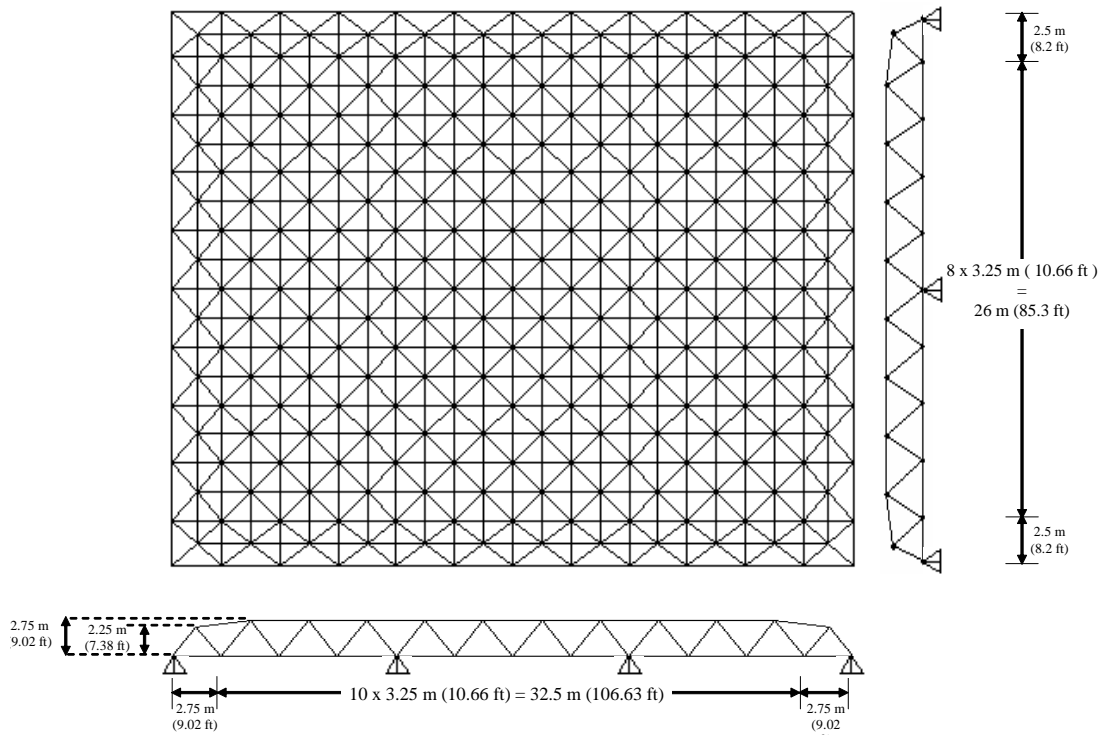
Figure 3.14 illustrates the last test problem which regards a double layer grid

consisting of 263 joints and 960 members (Hasançebi et al. 2009). The symmetry of the structure around x and y axes is used to group the 960 members into 251 independent sizing variables. They are selected from a list of 28 standard sections with cross-sectional areas between 1.07 in.<sup>2</sup> (6.90 cm<sup>2</sup>) and 21.3 in.<sup>2</sup> (137.42 cm<sup>2</sup>) in Table 3.1. The structure is subjected to a single load case resulting from snow load. This load case is considered based on a design snow pressure of 0.754 kN/m<sup>2</sup> (15.75 lb/ft<sup>2</sup>) calculated per ASCE (1998) manual. The stress and stability limitations of the members are computed in accordance with the specifications of AISC-ASD (1989). Further, the displacements of all nodes are limited to a maximum value of 4.16 in (10.57 cm) in any direction.

The considered 960-bar double layer grid is a challenging design example due to the large number of design variables considered. It is noted that only the stress and slenderness ratio constraints are active for this example. The lightest design for the double layer grid system is attained by MBB-BC algorithm, which is 24266.7 kg (53498.8 lb). The other designs are 24388.3 kg (53656.7 lb) by SA, 24780.2 kg (54631.0 lb) by ESs, 24973.5 kg (55057.1 lb) by PSO, 25320.0 kg (55821.1 lb) by TS, 29556.6 kg (65161.2 lb) by ACO, 32338.5 kg (71294.2 lb) by SGA and 40133.8 kg (88479.9 lb) by HS according to Hasançebi et al. (2009). No feasible solution is obtained with the BB-BC algorithm when the initial population is generated randomly. To facilitate design transitions to feasible regions during the search, the algorithm is started from one feasible design point such that all the member groups are assigned to the strongest section of the discrete set in one individual, while all other individuals in the initial population are created randomly in a usual manner. The BB-BC algorithm employed under this case produces a final design weight of 31119.8 kg (68607.4 lb). The variations of the best feasible design obtained so far in the search processes with BB-BC and MBB-BC algorithms are plotted in Figure 3.15.



a ) 3D view



b ) Top view

Figure 3.14: 960-bar double layer grid; a) 3D view, b) Top view (Hasançebi et al. 2009)



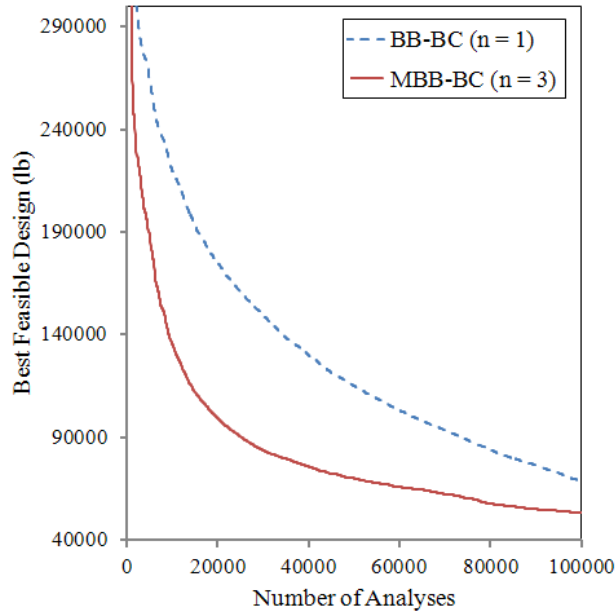


Figure 3.15: Optimization histories for 960-bar double layer grid

### 3.6.2 Performance Evaluation of EBB-BC Algorithm

In order to experiment and quantify the performance of EBB-BC algorithm in optimum design of steel frames, two numerical examples are investigated in this section. The design examples include a 132-member unbraced steel frame and a 209-member industrial factory building and are selected regarding their practical characteristics. The optimum solutions produced for these frames with EBB-BC algorithm are compared to those achieved using the standard version (BB-BC algorithm) as well as other metaheuristic techniques. In addition, the effects of new reformulation on the variations of average search dimensionality ratio and average step size parameters are presented in each design example. For a fair comparison of results, the maximum number of structural analyses is limited to the previously reported values in the literature, which is 50,000 for both examples. For numerical applications, the value of parameter  $\alpha$  in Eqs. (3.3) and (3.10) is taken as 0.25 and a

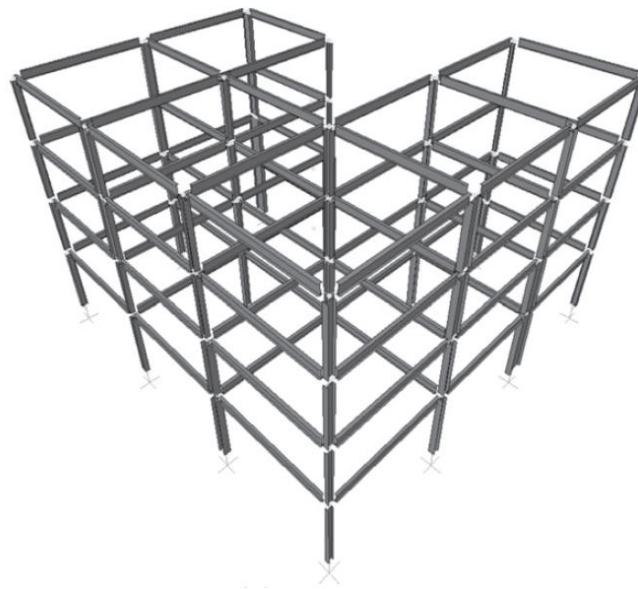
population size of 50 is used for all the algorithms. The material properties of steel used for all test problems are as follows: modulus of elasticity ( $E$ ) = 29,000 ksi (199,948 MPa) and yield stress ( $F_y$ ) = 36 ksi (248.2 MPa).

### **3.6.2.1 Example 1: 132-Member Unbraced Space Steel Structure**

The first example depicted in Figure 3.16 is a three dimensional unbraced (swaying) steel frame composed of 70 joints and 132 members that are grouped into 30 independent sizing variables to satisfy practical fabrication requirements (Hasançebi et al. 2010). The columns are adopted from the complete W-shape profile list consisting of 297 ready sections, whereas a discrete set of 171 economical sections selected from W-shape profile list based on area and inertia properties is used to size beam members. Here, both gravity and lateral loads are considered in designing the structure. Gravity loads ( $G$ ) consisting dead, live and snow loads are calculated according to ASCE 7-05 (2005) based on the following design values: a design dead load of 60.13 lb/ft<sup>2</sup> (2.88 kN/m<sup>2</sup>), a design live load of 50 lb/ft<sup>2</sup> (2.39 kN/m<sup>2</sup>), and a ground snow load of 25 lb/ft<sup>2</sup> (1.20 kN/m<sup>2</sup>). This yields the uniformly distributed loads on the outer and inner beams of the roof and floors given in Table 3.6. As for the lateral forces, earthquake loads ( $E$ ) are considered. These loads are calculated based on the equivalent lateral force procedure outlined in ASCE 7-05 (2005), resulting in the values given in Table 3.6 that are applied at the center of gravity of each story as joint loads. Gravity ( $G$ ) and earthquake ( $E$ ) loads are combined under two loading conditions for the frame: (i) 1.0G + 1.0E (in x-direction), and (ii) 1.0G + 1.0E (in y-direction). The joint displacements in  $x$  and  $y$  directions are limited to 1.53 in (3.59 cm) which is obtained as height of frame/400. Additionally, story drift constraints are applied to each story of the frame which is equal to height of each story/400.

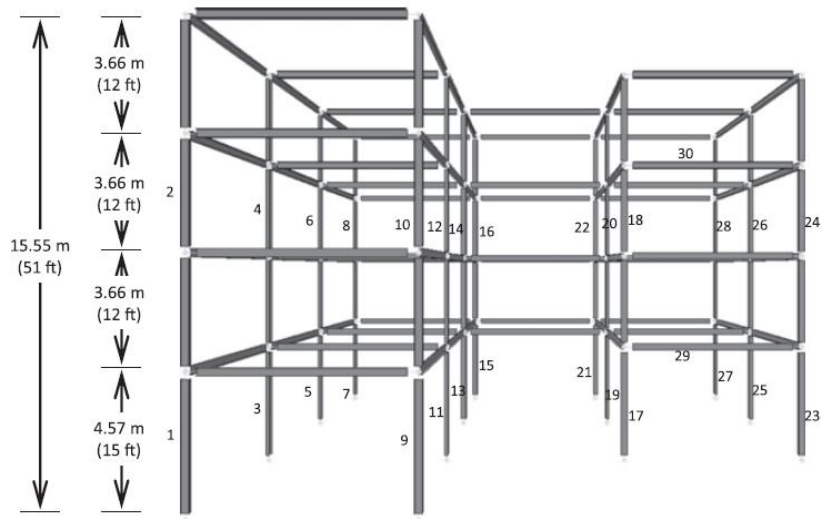
The BB-BC and EBB-BC algorithms are employed to minimize the weight of the 132-member steel frame. In Table 3.7 the minimum weight designs of the frame obtained by these algorithms are compared to the previously reported results by Hasançebi et

al. (2010) using different metaheuristic techniques; namely improved simulated annealing (iSA), tabu search (TS) and harmony search (HS). The EBB-BC algorithm produces a design weight of 60804.31 kg (134050.55 lb) for the frame which is the best solution of this problem reported so far. Relatively higher design weights have been obtained for the frame with other metaheuristic algorithms; namely 62993.55 kg (138874.67 lb) by iSA, 64733.69 kg (142710.96) by TS, 64926.17 kg (143135.29) by HS. The BB-BC algorithm exhibits a very poor performance and produces a final design weight of 87468.21 kg (192834.39 lb). Such a significant difference between the results clearly demonstrates the usefulness of the proposed refinement on the performance of the standard algorithm. The variations of the  $(SDR)_{ave}$  and  $(SS)_{ave}$  parameters in the implementations of the BB-BC and EBB-BC algorithms are shown in Figures 3.17 and 3.18, respectively.

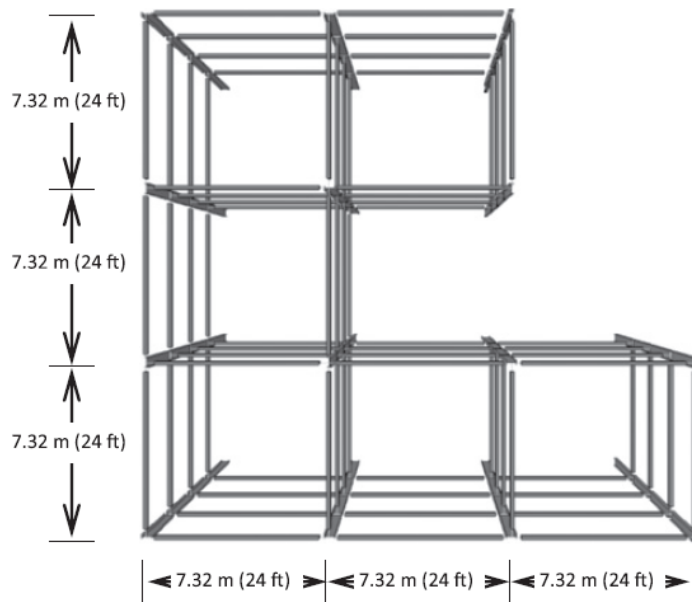


a) 3D view

Figure 3.16: 132-member space steel frame a) 3D view, b) Front view, c) Top view



b) Front view



c) Top view

Figure 3.16 (continued)

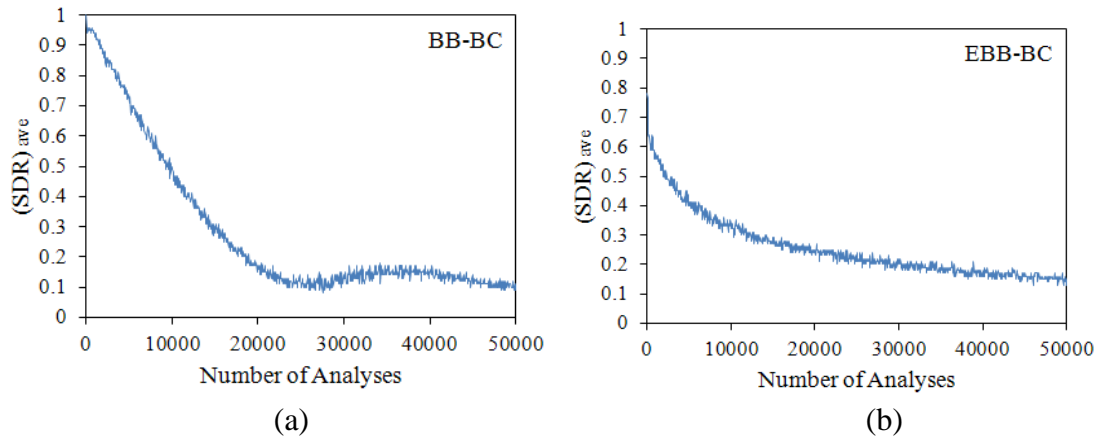


Figure 3.17:  $(SDR)_{ave}$  variations for 132-member space steel frame example; (a) BB-BC, and (b) EBB-BC variants

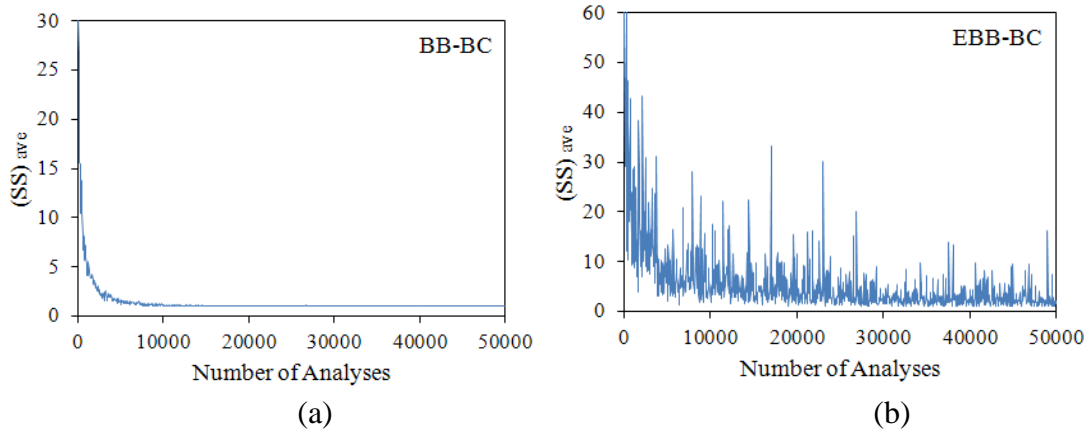


Figure 3.18:  $(SS)_{ave}$  variations for 132-member space steel frame example; (a) BB-BC, and (b) EBB-BC variants

Table 3.6. The gravity and lateral loading on 132-member space steel frame

Beam Type	Gravity Loads			
	Uniformly Distributed Load			
	Outer Span Beams (lb/ft) (kN/m)		Inner Span Beams (lb/ft) (kN/m)	
Roof Beams (Dead + Snow Loads)	1011.74	14.77	1193.84	17.42
Floor Beams (Dead + Live Loads)	1468.40	21.49	1732.70	25.29

Table 3.6. (continued)

Floor Number	Lateral Loads	
	Earthquake Design Load	
	(kips)	(kN)
1	6.57	29.23
2	12.43	55.28
3	18.52	82.35
4	24.76	110.15

Table 3.7. Comparison of results for 132-member space steel frame

Sizing variables	iSA	TS	HS	BB-BC	EBB-BC
1	W8X35	W8X31	W14X53	W24X176	W10X33
2	W18X86	W12X65	W12X120	W21X132	W12X79
3	W12X79	W27X129	W30X48	W27X336	W40X167
4	W18X65	W8X58	W16X77	W24X279	W12X65
5	W12X65	W12X79	W18X119	W14X193	W14X120
6	W27X161	W12X106	W24X104	W14X109	W14X109
7	W24X117	W18X97	W30X148	W12X87	W14X99
8	W10X54	W8X58	W10X68	W27X94	W14X90
9	W18X86	W12X72	W18X158	W30X292	W10X100
10	W12X96	W14X90	W12X120	W18X283	W12X106
11	W10X60	W36X135	W36X150	W10X49	W33X152
12	W10X49	W10X49	W16X67	W21X62	W12X53
13	W12X87	W12X96	W10X112	W18X311	W14X90
14	W12X50	W10X49	W24X117	W33X141	W36X160
15	W24X55	W24X55	W18X40	W18X40	W18X40
16	W24X55	W10X33	W14X61	W12X210	W12X53
17	W12X58	W18X76	W12X65	W16X67	W21X111
18	W12X67	W21X83	W18X119	W12X65	W12X65
19	W12X40	W8X40	W14X82	W14X211	W14X43
20	W10X49	W14X61	W18X86	W14X211	W10X60
21	W12X72	W18X76	W14X90	W40X277	W12X106
22	W12X79	W12X72	W18X97	W33X141	W10X88
23	W8X48	W12X40	W21X73	W12X65	W8X48

Table 3.7. (continued)

24	W24X68	W24X76	W12X87	W30X326	W27X84
25	W14X61	W10X77	W18X71	W12X72	W14X61
26	W21X50	W16X50	W27X102	W8X28	W10X39
27	W8X40	W10X49	W8X48	W30X124	W12X40
28	W8X67	W14X61	W24X117	W24X94	W18X76
29	W10X39	W18X97	W18X97	W16X89	W24X68
30	W21X44	W16X45	W16X40	W21X44	W18X40
Weight, lb (kg)	138874.67 (62993.55)	142710.96 (64733.69)	143135.29 (64926.17)	192834.39 (87468.21)	134050.55 (60804.31)

### 3.6.2.2 Example 2: 209-Member Industrial Factory Building

In this example, sizing optimization of a factory structure (Figure 3.19) composed of 100 joints and 209 members is considered (Saka and Hasańcebi 2009). The main structural system includes five similar frameworks lying 6.1 m (20ft) apart from each other. As shown in Figure 3.19b, each framework includes two side frames and a gable roof truss in between them. The lateral stability of the structure against wind loads in  $x$ - $z$  plane is provided through columns fixed at the base as well as the rigid connections of the side frames. Hence, the beams and columns of the side frames are considered as axial-flexural structural elements. On the other hand, the gable roof truss transmits axial forces only using pin-jointed connections. The member grouping is carried out considering the symmetry of building, resulting in 14 different member groups. Table 3.8 and Figure 3.19e give the member grouping details.

In order to design the industrial building three different types of loads namely dead, crane and wind loads are considered. Here, uniformly distributed loads of 14.63 kN/m (1004.55 lb/ft) and 7.32 kN/m (502.27 lb/ft) on the interior and exterior beams of the side frames are considered as dead loads. As shown in Figure 3.19d, the crane load is modeled as two pairs of moving live loads where each pair consists of a concentrated load of 280 kN (62.9 kip) and a couple moment of 75 kN.m (5532 kip.ft). In the

present study the crane load is shown by two different load cases namely CL1 and CL2. As shown in Figure 3.19d, in CL1 the crane is positioned at points A and A' whereas in CL2 it is positioned at points B and B'.

For design purpose, the wind in the  $x$ -direction is considered only and the resulting wind loads are computed based on a basic wind speed of  $V = 46.94$  m/s (105 mph) per ASCE 7-05 (2005). Here, WL1 and WL2 load cases are generated considering the sign of the internal wind pressure exerted on the external faces of the structure. Further details about this design example can be found in Saka and Hasançebi (2009). Regarding all the load cases, 6 load combinations are considered as follows:

- (i)  $1.0DL + 1.0CL1$
- (ii)  $1.0DL + 1.0CL1 + 1.0WL1$
- (iii)  $1.0DL + 1.0CL1 + 1.0WL2$
- (iv)  $1.0DL + 1.0CL2$
- (v)  $1.0DL + 1.0CL2 + 1.0WL1$
- (vi)  $1.0DL + 1.0CL2 + 1.0WL2$

In this design optimization example, the combined stress, stability and geometric constraints are imposed with respect to AISC-ASD (1989) specifications. Further, displacements of all the joints in  $x$  and  $y$  directions are limited to 3.43 cm, and the maximum allowable value for inter-story drifts is taken as 1.52 cm.

The BB-BC and EBB-BC algorithms are employed to minimize the weight of the industrial factory building. In Table 3.9 the minimum weight designs of the structure obtained by these algorithms are compared to the previously reported results by Saka and Hasançebi (2009) using harmony search (HS) and its adaptive variant (AHS)

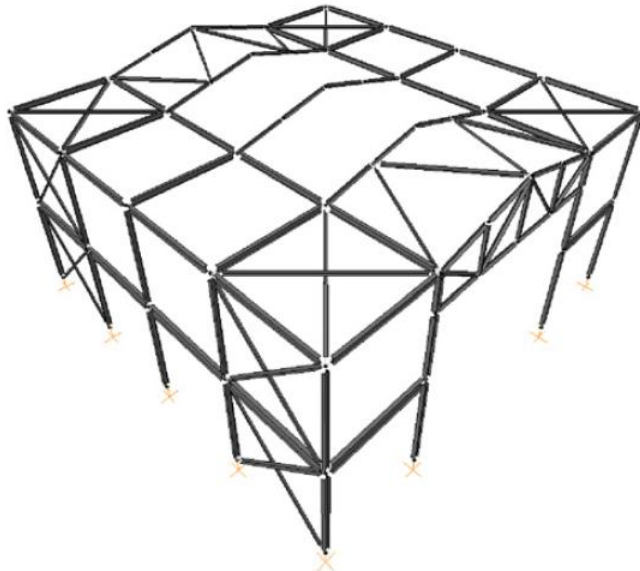


techniques. Again the EBB-BC algorithm performs very well and produces the best known solution of the problem, which is 42924.07 kg (94631.38 lb). The final designs attained for this problem with AHS and HS techniques were 44053.45 kg (97121.3 lb) and 46685.83 kg (102924.73 lb), respectively. On the other hand, a substandard performance is exhibited by BB-BC algorithm, in which the structural weight could only be decreased to 73375.37 kg (161764.99 lb).

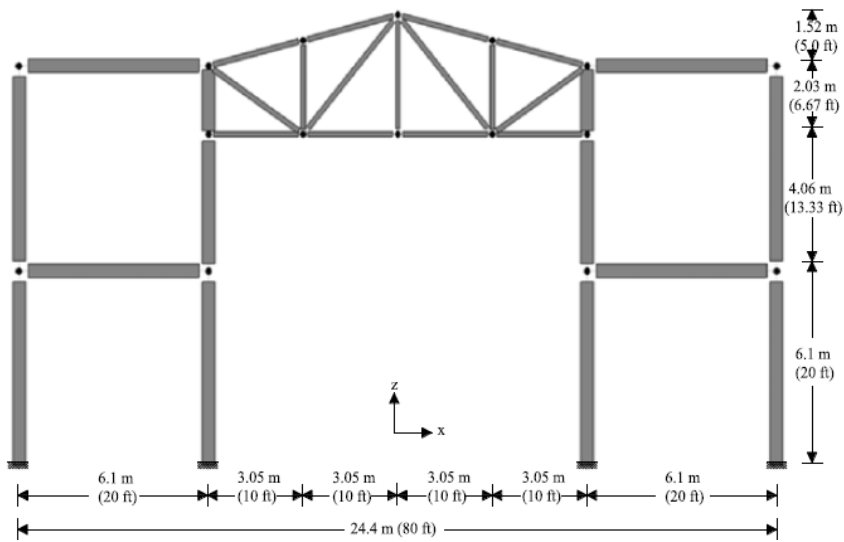
The variations of  $(SDR)_{ave}$  and  $(SS)_{ave}$  parameters in the implementations of these algorithms are shown in Figures 3.20 and 3.21, respectively. It is seen from these figures that the BB-BC algorithm quickly drops the average step size to one, and tends to search only the very near neighborhood of the solutions found during the former iterations. This exploiter characteristic of the algorithm makes it difficult to explore the solution space effectively, resulting in a premature convergence to a local minimum.

On the other hand, the EBB-BC algorithm follows a more successful strategy in searching the optimum solution in the design space by adjusting an optimal balance between exploration and exploitation features of the algorithm. In comparison to the BB-BC, the EBB-BC makes greater changes in the  $(SS)_{ave}$  values while decreasing the  $(SDR)_{ave}$  parameter in a gradual way, resulting in the minimum weight design of the industrial building.

It is expedient to highlight that performance enhancement of the existing metaheuristic algorithms for tackling specific optimization problems has become one of the most frequent strategies in the recent years. The key issue in creating a new metaheuristic optimization technique as well as modification of an existing algorithm is how to adjust a balance between the exploration and exploitation characteristics of the technique. In this regard, the main concern is to investigate the effect of changes in the formulation of the algorithm on the quality of final solutions.

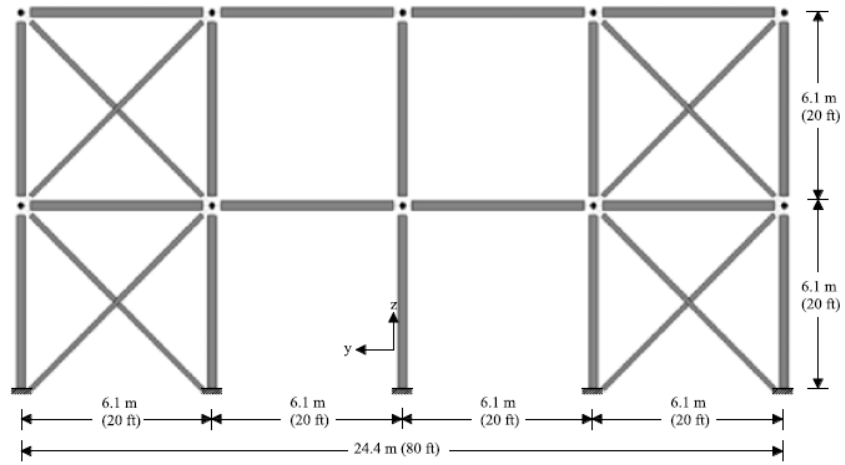


a) 3D view

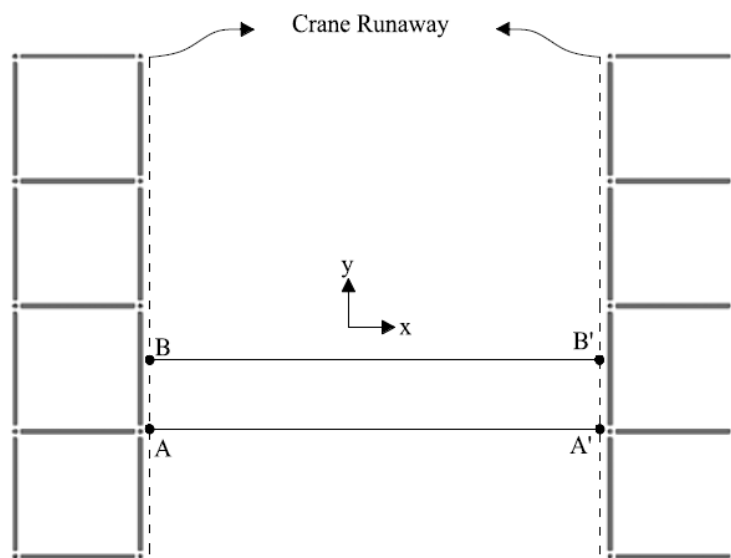


b) Front view

Figure 3.19: 209-member industrial factory building

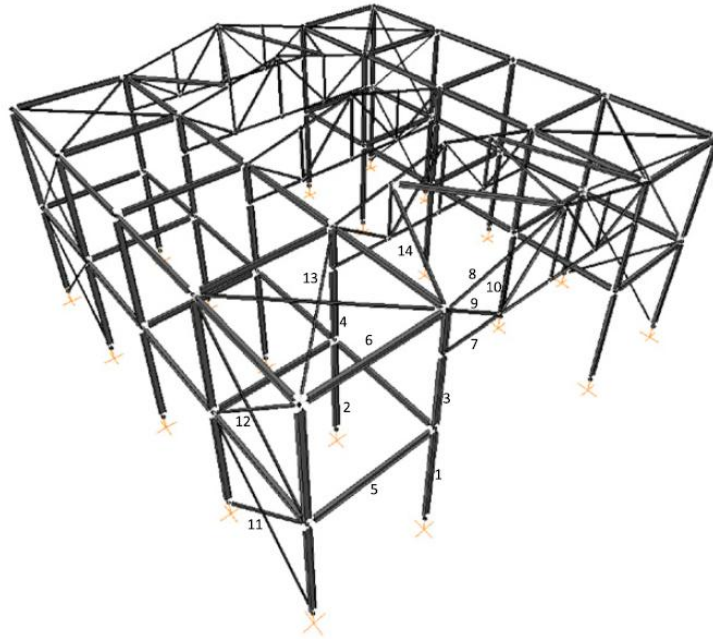


c) Side view



d) First floor plan view

Figure 3.19 (continued)



e) Member grouping

Figure 3.19 (continued)

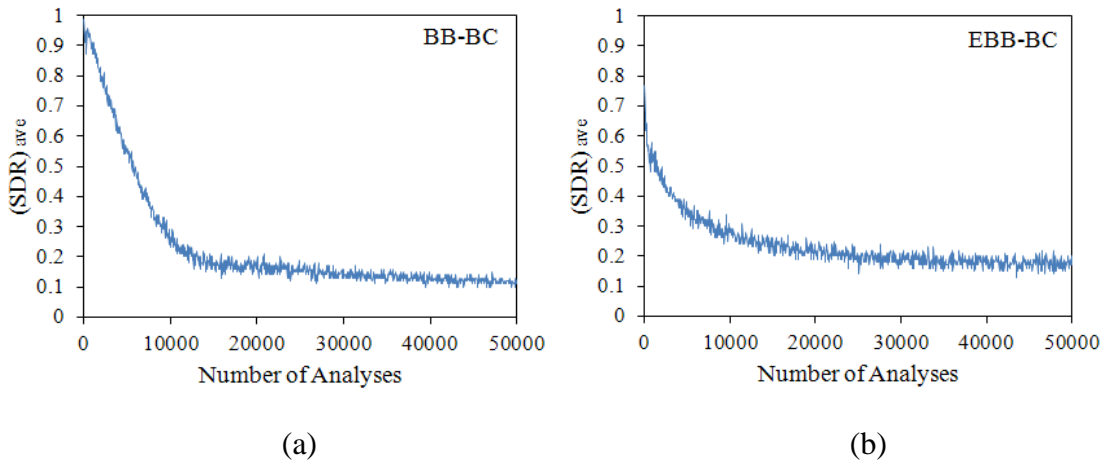


Figure 3.20:  $(SDR)_{ave}$  variations for 209-member industrial factory building example;  
 (a) BB-BC, and (b) EBB-BC variants

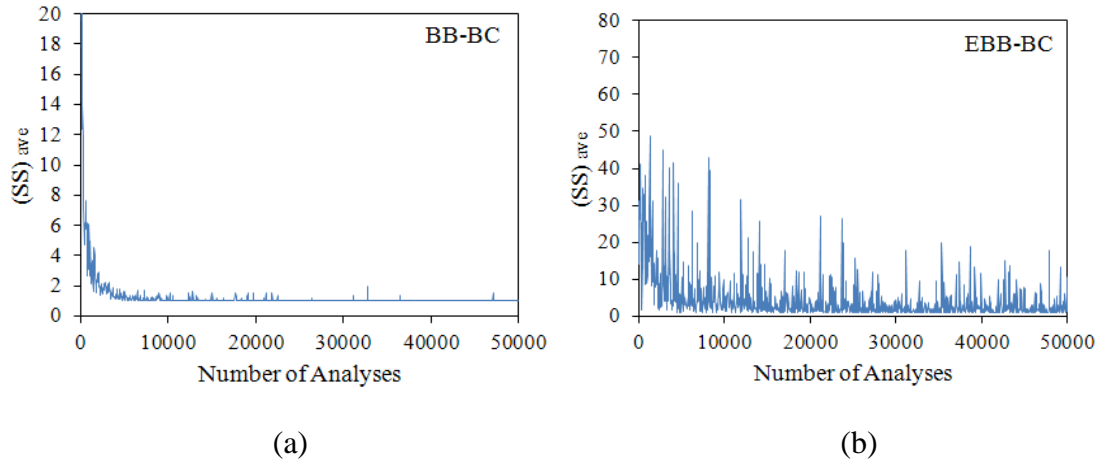


Figure 3.21:  $(SS)_{ave}$  variations for 209-member industrial factory building example; (a) BB-BC, and (b) EBB-BC variants

Table 3.8. Member grouping details for 209-member industrial factory building

Member	Group Name	Member	Group Name
1	1st floor external columns	8	Truss top chord
2	1st floor internal columns	9	Truss web diagonals
3	2nd floor external columns	10	Truss web verticals
4	2nd floor internal columns	11	1st floor wall braces
5	1st floor beams	12	2nd floor wall braces
6	2nd floor beams	13	Floor frames braces
7	Truss bottom chord	14	Floor truss braces

Table 3.9. Comparison of results for 209-member industrial factory building

Sizing	HS	AHS	BB-BC	EBB-BC
1	W8X31	W8X31	W16X57	W10X33
2	W12X40	W10X39	W16X57	W10X33
3	W8X31	W12X26	W8X28	W8X24
4	W8X40	W8X40	W21X68	W10X33
5	W24X62	W24X62	W24X62	W24X62
6	W12X26	W10X26	W21X44	W12X26

Table 3.9. (continued)

7	2L2.5X2X3/16	2L2X2X1/8	2L5X5X5/8	2L2X2X1/8
8	2L2X2X1/8	2L2X2X1/8	2L2X2X1/8	2L2X2X1/8
9	2L3X3X3/16	2L3X3X3/16	2L4X4X5/8	2L3X3X3/16
10	2L3X2.5X5/16	2L2X2X1/8	2L2.5X2.5X3/16	2L2X2X1/8
11	2L6X6X7/16	2L6X6X5/16	2L6X6X3/4	2L6X6X5/16
12	2L6X6X3/8	2L6X6X5/16	2L8X8X3/4	2L6X6X5/16
13	2L6X6X5/16	2L6X6X5/16	2L6X6X5/8	2L6X6X5/16
14	2L6X6X5/16	2L5X5X5/16	2L5X5X7/16	2L5X5X5/16
Weight, lb	102924.73	97121.3	161764.99	94631.38
(kg)	(46685.83)	(44053.45)	(73375.37)	(42924.07)

### 3.7 Summary

In this chapter efficient structural design optimization algorithms based on a big bang-big crunch algorithm are implemented for discrete sizing optimization of steel structures. Through modifications of the standard algorithm (BB-BC) two enhanced variants namely MBB-BC and EBB-BC algorithms are introduced for design optimization of steel trusses and frames, respectively. The numerical efficiencies of the proposed algorithms are quantified using different optimum design instances. In all the examples, the steel structures are designed for minimum weight subject to strength and serviceability limitations of AISC-ASD (1989). Based on the results obtained in the investigated examples it is shown that the performance of the BB-BC algorithm can be improved to a great extent with the proposed reformulations. Furthermore, in comparison to the previously reported results with various meta-heuristic techniques, the MBB-BC and EBB-BC algorithms obtained promising designs, demonstrating the robustness of the optimization process with the latter. The success of the proposed algorithms are basically due to their abilities to provide more advantageous mechanism for adjusting search dimensionality ratio when dealing with discrete design optimization problems as well as making it possible to have occasional increments in the step size values throughout the optimization process.

## CHAPTER 4

### UPPER BOUND STRATEGY

#### 4.1 Introduction

Optimum design of structural systems using metaheuristic algorithms suffers from enormously time-consuming structural analyses to locate a reasonable design. Basically there are three different types of remedies for this problem. One approach is to reduce the total computational effort through employing computationally more efficient structural analysis techniques such as taking advantages of the force method of structural analysis (Kaveh and Kalatjari 2002) or employing approximate structural analysis techniques (Salajegheh 1997). Alternatively, a relatively expensive approach can be employing high performance computing techniques, such as parallel or distributed computing methods (Hasançebi et al. 2011a). Another approach, which is more straightforward and easier to apply, is to develop efficient strategies for reducing the total number of structural analyses required in the course of optimization. The latter, can be performed by investigating the algorithmic structure of the existing metaheuristics and proposing new strategies capable of diminishing their total computational effort without affecting their performances in locating the global optima.

In this chapter an upper bound strategy (UBS) is proposed for reducing the total number of structural analyses in metaheuristic based design optimization of steel structures. The idea behind the UBS is to detect those candidate designs which have no chance to improve the search during the iterations of the optimization algorithm. After identifying the non-improving candidate designs, they are directly excluded from the structural analysis stage, resulting in diminishing the total computational

effort. The well-known big bang-big crunch algorithm as well as its two enhanced variants, namely MBB-BC and EBB-BC algorithms, are adopted as typical metaheuristic techniques to evaluate the effect of the UBS on computational efficiency of these algorithms. The numerical results clearly reveal that the UBS can significantly lessen the total computational cost in metaheuristic based design optimization of steel structures.

## 4.2 The UBS in Metaheuristic Based Design Optimization

The UBS can be used in conjunction with all metaheuristic algorithms that employ a  $\mu + \lambda$  selection scheme in their algorithmic models (Kazemzadeh Azad et al. 2013a, b). This selection scheme is first characterized by the well-known variant of evolution strategies (ES) technique referred to as  $(\mu + \lambda)$ -ES in the literature (Schwefel 1981). Typically, at each generation of the  $(\mu + \lambda)$ -ES,  $\mu$  parents generate  $\lambda$  offspring; and then a deterministic selection is performed by selecting the  $\mu$  best individuals out of  $\mu$  parents and  $\lambda$  offspring in reference to the individuals' fitness scores (Michalewicz 1996). This way, the number of individuals to produce the next generation is reduced back to  $\mu$  every time. It should be noted that the evolutionary scheme employed in the BB-BC algorithm works on the basis of the same principle. At each iteration of the BB-BC algorithm  $\mu = 1$  parent generates  $\lambda$  candidate solutions ( $\lambda = 50$  in this research), and only one individual survives out of  $\lambda + 1$  solutions after implementing the selection and elitism schemes.

In the BB-BC, MBB-BC, and EBB-BC algorithms discussed in the previous chapter, the current best design found during the optimization process is used to generate new candidate solutions for the next iteration. Then, every candidate solution generated is subjected to evaluation such that a conventional structural analysis is first carried out per design and then its penalized weight (which is the base of comparison between the solutions) is computed through the application of Eq. (3.1). The idea behind the UBS is to impose the current best design as the *upper bound* for the forthcoming candidates



to eliminate unnecessary structural analysis and associated fitness computations for those candidates that have no chance of surpassing the best solution. The pseudo-code for the UBS integrated algorithms is outlined below, where `ite_cnt` and `ite_max` stand for the current and maximum iteration numbers, respectively.

```

Repeat
Generate  $\mathbf{I}_i$  ( $i:=1,..,\lambda$ ) from  $\mathbf{I}_{best}$  using Eqs. (3.3), (3.9) or (3.10)

For  $i:=1$  to  $\lambda$  do
begin
    Calculate  $W(\mathbf{I}_i)$ 
If  $W(\mathbf{I}_i) \leq f(\mathbf{I}_{best})$ 
then
        Perform structural analysis of  $\mathbf{I}_i$ 
        Compute  $P(\mathbf{I}_i)$  and  $f(\mathbf{I}_i)$ 
else
        Activate UBS
        Eliminate  $\mathbf{I}_i$ 
end
Set new  $\mathbf{I}_{best}$ 
ite_cnt:=ite_cnt+1;
until ite_cnt>ite_max

```

In the proposed strategy,  $\lambda$  number of candidate solutions  $\mathbf{I}_i$  are first generated from the current best design in a usual manner, i.e. through application of Eqs. (3.3), (3.9) or (3.10). Then, in the first step the net weight  $W(\mathbf{I}_i)$  of each candidate solution is calculated only; not the penalized weight. This computation is straightforward and can be done with a trivial computational effort. If a candidate solution has a net weight  $W(\mathbf{I}_i)$  smaller than or equal to the penalized weight of the current best design  $f(\mathbf{I}_{best})$ ,

the structural analysis of the candidate solution is processed and its penalized weight is computed. In the opposite case, i.e.  $W(\mathbf{I}_i) > f(\mathbf{I}_{best})$ , however, the UBS is activated and the candidate solution is automatically removed from the population without undergoing structural analysis phase for response computations, since such a candidate is unlikely to improve the current best design  $\mathbf{I}_{best}$ .

Basically, the key factor in this approach is to define the penalized weight of the current best solution found during the previous iterations as an upper bound for the net weight of the newly generated candidate solutions. Thus, any new candidate solution with a net weight greater than this upper bound will not be analyzed and this will lessen the computational burden of the optimization algorithm. It is worthwhile to mention that the UBS is applicable to all other instances of  $\mu + \lambda$  selection scheme in which  $\mu$  is greater than one. For such cases, the upper bound for the net weight of the newly generated candidate solutions would be the penalized weight of the worst design among the  $\mu$  parents.

It should be noted that the UBS does not affect exploration and exploitation characteristics of the employed algorithms; however, using this strategy it is possible to perform a computationally more efficient design optimization. Basically, at each iteration of the utilized optimization algorithms, a portion of the whole design space is investigated through the newly generated individuals. Once the algorithm is integrated with UBS, the same portion of design space is investigated; but mostly only few individuals undergo a structural analysis and non-improving individuals, which are not analyzed, contribute in the search process only with their net weights.

The main question here is the amount of saving in structural analyses through the proposed strategy. This is answered in the next section through numerical examples of real size steel structures.

### 4.3 Numerical Examples

In this section the efficiency of the UBS in reducing the number of structural analyses is investigated through design optimization examples of steel skeletal structures. To this end, the optimization algorithms are coded in MATLAB and employed in conjunction with SAP2000 v14.1 structural analysis package using application programming interface (API) for analysis and design of structural systems sampled during the course of optimization process. The optimization runs are performed using a regular PC with AMD Athlon II X4 620, 2.60 GHz CPU and 2 GB RAM. It should be noted that in the investigated examples the shear deformations as well as second order effects are neglected.

The population size of the algorithms is set to 50 and the value of parameter  $\alpha$  in Eqs. (3.3), (3.9) and (3.10) is selected as 0.5. The value of penalty constant  $p$  is taken as 1. Further, the maximum number of iterations (`ite_max`) is considered as the termination criterion. Here, the wide-flange (W) profile list composed of 268 ready sections is used to size the frame members; and the material properties of steel are taken as follows: modulus of elasticity ( $E$ ) = 200 GPa, yield stress ( $F_y$ ) = 248.2 MPa, and unit weight of the steel ( $\rho$ ) = 7.85 ton/m<sup>3</sup>.

In the following examples the notations UBB-BC, UMBB-BC and UEBB-BC are used to refer to the UBS integrated versions of the BB-BC, MBB-BC, and EBB-BC algorithms, respectively. It should be underlined that an improvement of the algorithms in terms of quality of the optimum solution located is not intended in this study. Instead, the objective of the study is to accelerate computational efficiency of the algorithms by reducing their computing time through a smaller number of structural analyses performed in the course of optimization. It is worth mentioning that the reported optimum solutions refer to the feasible best solutions attained by the algorithms, rather than the final solutions. In this research, when the algorithms are implemented, the feasible best solutions located by the algorithms are stored in a

separate data file and updated when a better feasible solution is generated. Hence, all the reported solutions are completely feasible and have zero violation of constraints.

#### 4.3.1 Example 1: 135-Member Steel Frame

The 3-story steel frame shown in Figure 4.1 is adopted as the first example. The frame is composed of 135 structural members including 66 beam, 45 column and 24 bracing elements. The stability of structure is provided using moment resisting connections as well as inverted V-type bracing systems along the  $x$  direction. For practical fabrication requirements, the 135 members of the frame are collected under 10 member groups. As depicted in Figure 4.2, the columns are grouped into four sizing variables in a plan level as corner, inner, side  $x$ - $z$  and side  $y$ - $z$  columns, and they are assumed to have the same cross-section over the three stories of the frame. On the other hand, all the beams in each story are grouped into one sizing variable, resulting in three beam-sizing design variables for the frame. Similarly, all the bracings in each story are grouped into one sizing variable, resulting in three bracing-sizing design variables for the frame. It is worth mentioning that floor slabs shown in Figure 4.1(a) are just for better illustration of the structure; and are not modeled in the analysis stage. For design purpose, the frame is subjected to the following 10 load combinations per ASCE 7-98 (2000):

(1)  $1.4D$

(2)  $1.2D + 1.6L$

(3)  $1.2D + 1.0E_x + 0.5L$

(4)  $1.2D + 1.0E_{ex} + 0.5L$

(5)  $1.2D + 1.0E_y + 0.5L$

(6)  $1.2D + 1.0E_{ey} + 0.5L$

(7)  $0.9D + 1.0E_x$

(8)  $0.9D + 1.0E_{ex}$

(9)  $0.9D + 1.0E_y$

(10)  $0.9D + 1.0E_{ey}$

where  $D$  and  $L$  denote the dead and live loads, respectively;  $E_x$  and  $E_y$  are the earthquake loads applied to the center of mass in  $x$  and  $y$  directions, respectively;  $E_{ex}$  and  $E_{ey}$  are the earthquake loads applied considering the effect of accidental eccentricity of the center of mass in  $x$  and  $y$  directions, respectively. Based on ASCE 7-98 (2000) the amount of eccentricity is set to 5% of the dimension of the structure perpendicular to the direction of the applied earthquake load.

The live loads acting on the floor and roof beams are 12 and 7 kN/m, respectively. In the case of dead loads, besides the uniformly distributed loads of 20 and 15 kN/m applied on floor and roof beams, respectively, the self-weight of the structure is also considered.

The earthquake loads, are calculated based on the equivalent lateral force procedure outlined in ASCE 7-98 (2000). Here, the resulting seismic base shear ( $V$ ) is taken as  $V = 0.15W_s$  where  $W_s$  is the total dead load of the building. The computed base shear is distributed to each floor based on the following equation:

$$F_x = \frac{w_x h_x^k V}{\sum_{i=1}^n w_i h_i^k} \quad (4.1)$$

where  $F_x$  is the induced lateral seismic force at level  $x$ ;  $w$  is portion of the total gravity load assigned to the related level (i.e. level  $i$  or  $x$ ); and  $h$  is the height from base to the related level. Here,  $k$  is determined based on the structure period. It is equal to 1 for structures with a period of 0.5 sec or less; and 2 for structures with a period of 2.5 sec or more. For structures with a period in range of 0.5 to 2.5 sec,  $k$  is calculated through linear interpolation per ASCE 7-98 (2000). It is worth mentioning that the approximate fundamental period of the structure is calculated using the following equation given in ASCE 7-98 (2000).

$$T = C_T h_n^{3/4} \quad (4.2)$$

where  $C_T$  is taken as 0.0853 and  $h_n$  is the height of the building; namely 12 m for this example. Hence, the period of the structure,  $T$ , is approximately taken as 0.55 sec. It is worth mentioning that modal analysis can be also carried out in order to obtain a more accurate value for  $T$ . Here, based on the above-mentioned period, the value of parameter  $k$  in Eq. (4.1) is set to 1.025 for this example. It should be noticed that since the self-weight of the structure changes during the course of optimization, apparently, the values of dead and earthquake loads change accordingly.

The beam elements are continuously braced along their lengths by the floor system; and columns and bracings are assumed to be unbraced along their lengths. The effective length factor,  $K$ , is taken as 1 for all beams and bracings. The  $K$  factor is conservatively taken as 1.0 for buckling of columns about their minor (weak) direction, since the frame is assumed to be non-swaying in that direction owing to inverted V-type bracing systems. However, for buckling of columns about their major direction the  $K$  factor is calculated according to section 2.2.

The maximum lateral displacement of the top story is limited to 0.03 m and the upper limit of interstory drift is taken as  $h/400$ , where  $h$  is the story height. The interstory drifts are calculated based on the displacement of center of mass of each story. The maximum lateral displacement of the top story is calculated with respect to the maximum displacements of the ends of the structure. Here, horizontal displacements of all joints of each story are constrained to each other based on a rigid diaphragm assumption.

Optimum desing of the frame is performed using the UBB-BC, UMBB-BC and UEBC-BC algorithms and the results obtained are tabulated in Table 4.1. The algorithms are executed until the termination condition, which is the maximum number of iterations, i.e. 350, is met. The optimization histories of the algorithms are presented in Figure 4.3, which shows the variation of the penalized weight of the current best design obtained so far in the search process.

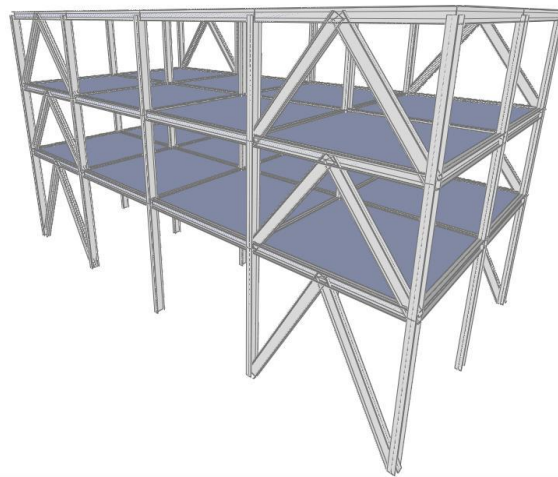
Since each UBS integrated algorithm (e.g. UMBB-BC) and its corresponding original algorithm (e.g. MBB-BC) utilizes the same formulation for the search procedure, the optimum designs reported with each UBS integrated algorithm is also valid for the corresponding original algorithm. However, the number of structural analyses to reach the optimum design will be different as a result of the UBS employed in the former. Here, the number of structural analyses performed in the UBS integrated algorithms is calculated by counting candidate solutions that undergo structural analysis. For the original algorithms (i.e. BB-BC, MBB-BC, and EBB-BC) this can be simply obtained through multiplying the total number of iterations ( $ite\_cnt$ ) by the population size  $\lambda$ . It should be noted that comparing the performances of the employed algorithms is not the aim of this study. The main concern is to demonstrate the effect of UBS on performance of each algorithm.

Figures 4.4 to 4.6 depict the amount of saving in structural analyses at each iteration of the UBB-BC, UMBB-BC, and UEBB-BC algorithms, respectively. The markers on the graphs show the amount of structural analyses saved out of a total number of 50 analyses at each iteration. For instance a saving of 80% at an iteration of a UBS integrated algorithm means that only 10 candidate solutions are analyzed and the remaining 40 candidate solutions are removed from the population (without performing an structural analysis) at that iteration due to their violation of the upper bound limit. Further, as shown in Figure 4.4 since all the individuals of the initial population are analyzed, there is no saving in the first iteration of the UBS integrated algorithms.

Bearing in mind that a population size of  $\lambda = 50$  is employed over a maximum number of 350 iterations ( $ite\_max = 350$ ), the number of structural analyses performed by the BB-BC, MBB-BC and EBB-BC algorithms is equal to 17500. However, when UBB-BC, UMBB-BC, and UEBB-BC algorithms are employed, it is found that a total of 880, 1794, and 1235 structural analyses are performed, respectively. This implies that using UBB-BC, UMBB-BC, and UEBB-BC algorithms, the amount of saving in the number of structural analyses is 94.97%, 89.75%, and 92.94%, respectively. As

presented in Table 4.2, using the BB-BC algorithm without UBS the total computational time is 24.31 hrs; whereas UBS integrated algorithms, namely UBB-BC, UMBB-BC, and UEBB-BC, perform the design optimization of the frame only in 1.22, 2.49, and 1.72 hrs, respectively. It should be noted that since the MBB-BC and EBB-BC algorithms also perform 17500 structural analyses during the optimization, the CPU time of these algorithms is similar to that of reported for the BB-BC algorithm and is not repeated here.

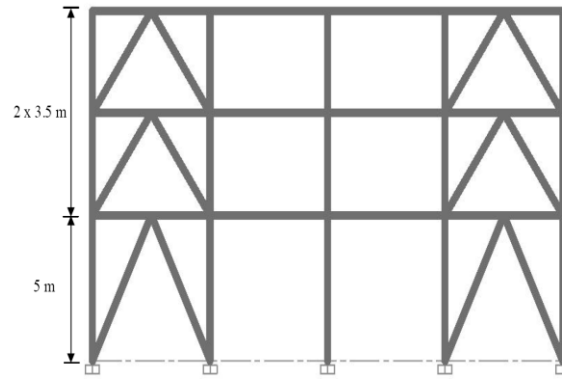
It is observed that in the optimum design located by the UEBB-BC algorithm the maximum interstory drift is 99.80% of the allowable limit value; the maximum lateral displacement of the top story is 91.09% of the allowable displacement; and the load to capacity ratio for the most critical frame member is 0.93.



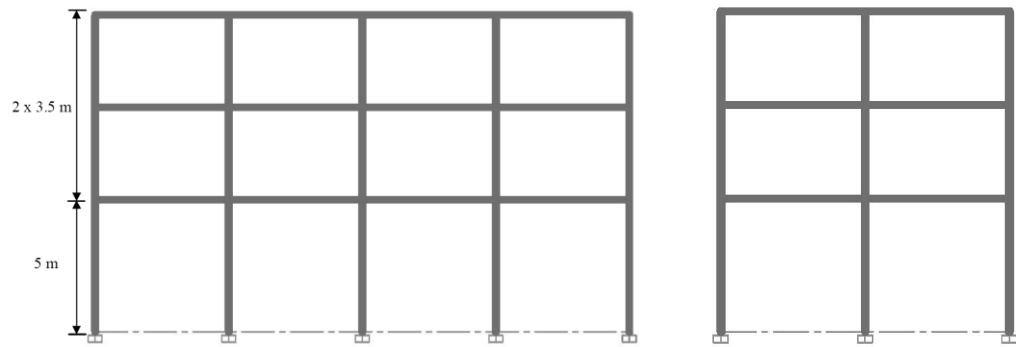
(a)

Figure 4.1: 135-member steel frame, (a) 3-D view (b) side view of frames 1 and 3 (c) side view of frame 2 (d) side view of frames A, B, C, D, and E (e) plan view



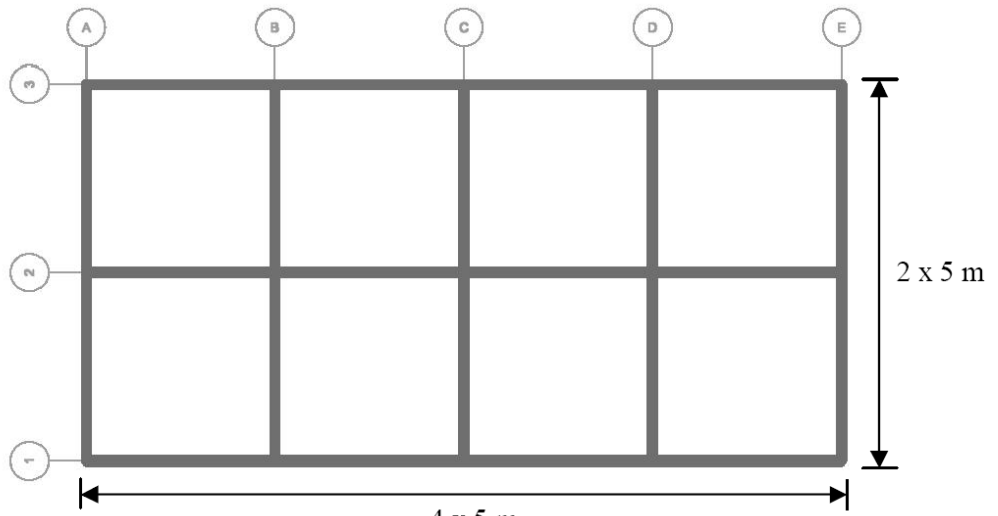


(b)



(c)

(d)



(e)

Figure 4.1 (continued)

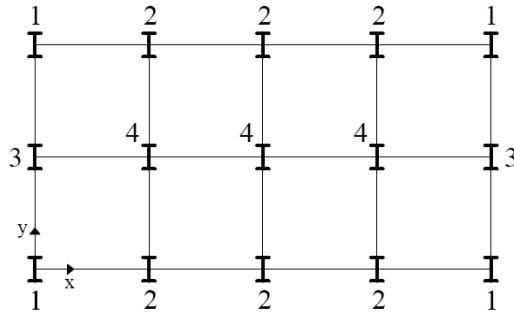


Figure 4.2: Columns grouping of 135-member steel frame in plan level

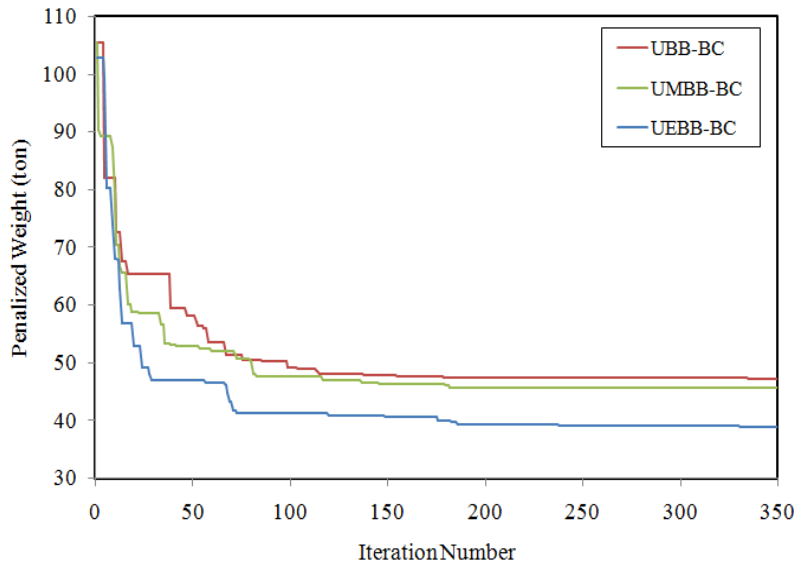


Figure 4.3: Optimization histories of the 135-member steel frame example

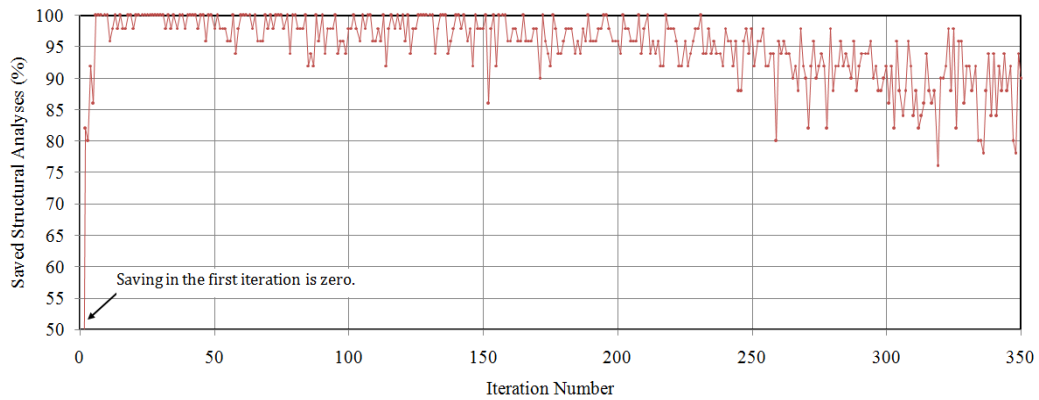


Figure 4.4: Saving in structural analyses using the UBB-BC algorithm in the 135-member steel frame example

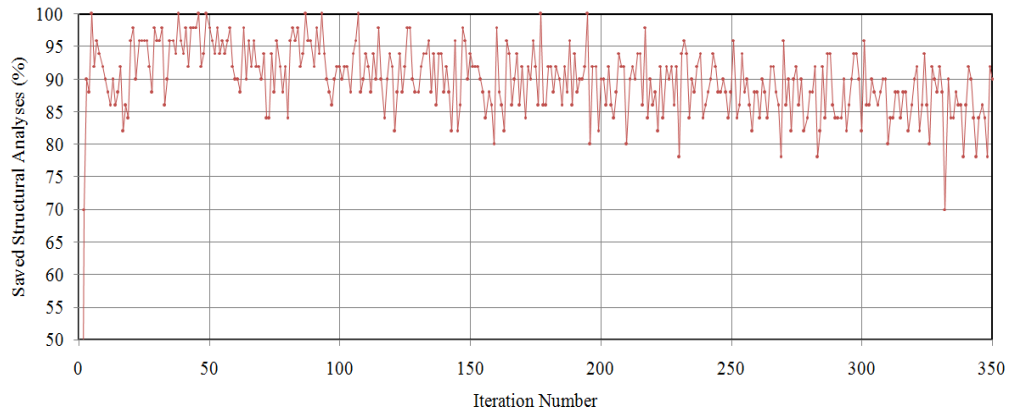


Figure 4.5: Saving in structural analyses using the UMBB-BC algorithm in the 135-member steel frame example

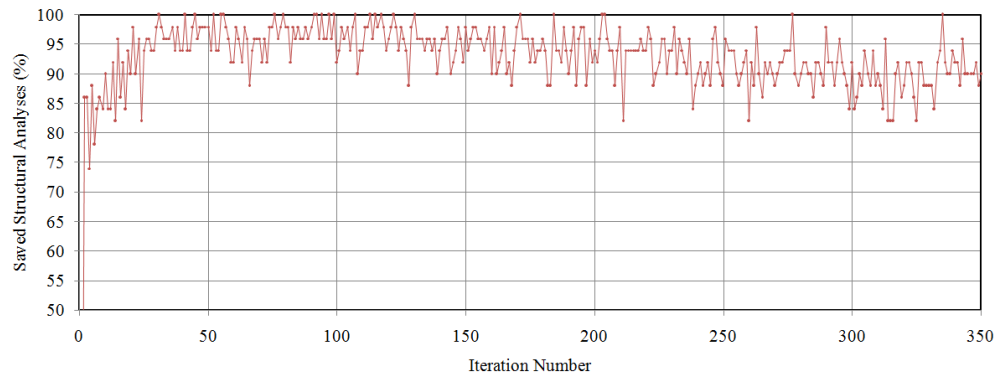


Figure 4.6: Saving in structural analyses using the UEBC-BC algorithm in the 135-member steel frame example

Table 4.1. Optimum designs obtained for 135-member steel frame

Groups	UBB-BC	UMBB-BC	UEBB-BC
CG <sub>1</sub> *	W10X39	W30X90	W21X62
CG <sub>2</sub>	W27X84	W14X48	W14X48
CG <sub>3</sub>	W40X149	W40X215	W36X150
CG <sub>4</sub>	W18X65	W27X84	W21X68
B <sub>1</sub> *	W21X44	W14X34	W18X40
B <sub>2</sub>	W16X40	W12X35	W18X35
B <sub>3</sub>	W10X22	W18X35	W16X26
BR <sub>1</sub> *	W27X84	W21X44	W8X24
BR <sub>2</sub>	W16X26	W10X22	W16X26
BR <sub>3</sub>	W21X44	W6X15	W6X15
Weight (ton)	47.3	45.67	38.91

\*CG denotes column group with respect to Figure 4.2, B<sub>i</sub>: beams and BR<sub>i</sub>: bracings of the i-th story

Table 4.2. Computational details for 135-member steel frame

<b>Computational Details</b>	<b>BB-BC</b>	<b>UBB-BC</b>	<b>UMBB-BC</b>	<b>UEBB-BC</b>
Generated Individuals	17500	17500	17500	17500
Analyzed Individuals	17500	880	1794	1235
CPU Time (hrs.)	24.31	1.22	2.49	1.72
Saving in Structural Analyses (%)	0	94.97	89.75	92.94

### 4.3.2 Example 2: 1026-Member Steel Frame

The 10-story steel frame shown in Figure 4.7 is selected as the second example. The frame is composed of 1026 structural members, including 580 beam, 350 column and 96 bracing elements. The stability of structure is provided through moment resisting connections as well as X-type bracing systems along the  $x$  direction. Considering practical fabrication requirements, the 1026 members of the frame are collected under 32 member groups. The member grouping is carried out in both plan and elevation levels. In elevation level the structural members are grouped in every three stories except the first story. In plan level, columns are considered in 5 different groups as depicted in Figure 4.8; beams are divided into two groups as outer and inner beams; and bracings are assumed to be in one group. Therefore, based on both elevation and plan level groupings, there are totally 20 column groups, 8 beam groups, and 4 bracing groups considered as 32 sizing design variables in this example. It is worth mentioning that floor slabs shown in Figure 4.7(a) are just for better illustration of the structure; and are not modeled in the analysis stage.

For design purpose, the frame is subjected to the same 10 load combinations described in the first example. The live loads acting on the floor and roof beams are 12 and 7 kN/m, respectively. In the case of dead loads, besides the uniformly distributed loads of 20 and 15 kN/m applied on floor and roof beams, respectively, the self-weight of the structure is also considered. The earthquake loads, are calculated based on the

same procedure described in the first example. Here, the resulting seismic base shear ( $V$ ) is taken as  $V = 0.1W_s$  where  $W_s$  is the total dead load of the building. Further, in Eq. (34),  $C_T$  is taken as 0.0853 and  $h_n$  is 36.5 m. Hence, the period of the structure,  $T$ , is approximately computed as 1.267 sec. Based on the obtained period the value of parameter  $k$  in Eq. (33) is taken as 1.38 for this example.

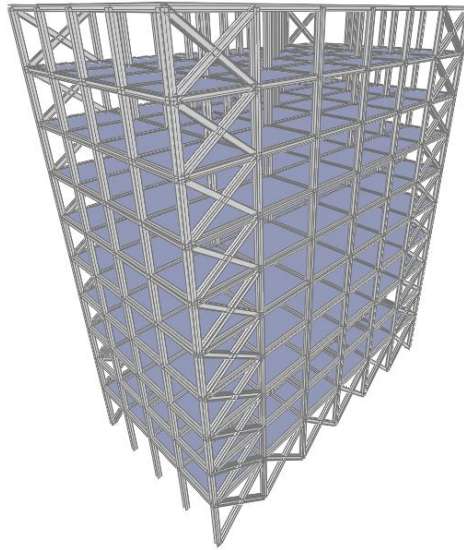
The unbraced lengths of all beam elements are set to the one fifth of their lengths; and columns and bracings are assumed to be unbraced along their lengths. The effective length factor,  $K$ , for buckling of columns about their minor direction as well as beams and bracings is taken as 1. For buckling of columns about their major direction the  $K$  factor is calculated according to relations given in chapter 2. The maximum lateral displacement of the top story is limited to 0.1 m and the upper limit of interstory drift is taken as  $h/400$ , where  $h$  is the story height.

Although typically the algorithms initiate with an initial population composed of randomly generated individuals, it was observed during the numerical investigations that a randomly generated initial population mostly is composed of highly infeasible and poor designs. On the other hand, it is more likely to generate some feasible individuals to initiate the optimization process simply by using strong sections of the available profile list. Hence, in this example the initial population is generated more deterministically to utilize candidate solutions some of which are expected to be feasible. The initial population is generated with respect to a profile list sorted based on sections' depth. The first individual which is expected to be feasible is one with the deepest section for all design variables (structural members). Accordingly, next individuals are generated using shallower sections of the list with 5 section intervals. For instance, since the structural members are to be selected from a list of 268 ready sections, the first individual of initial population is a candidate design with section number 268 (deepest section) for all structural members; the second individual will be composed of section number 263; and others will be generated similarly. The rationale behind this approach is to start the optimization process from a more promising point.

Optimum desing of the frame is carrid out using the UBB-BC, UMBB-BC and UEBB-BC algorithms and the results obtained are tabulated in Table 4.3. The optimization histories of the algorithms are presented in Figure 4.9. Figures 4.10 to 4.12 depict the amount of saving in structural analyses at each iteration of the UBB-BC, UMBB-BC, and UEBB-BC algorithms, respectively.

Bearing in mind that a population size of  $\lambda = 50$  is employed over a maximum number of 500 iterations ( $ite\_max = 500$ ), the number of structural analyses performed by the BB-BC, MBB-BC and EBB-BC algorithms is equal to 25000. However, when UBB-BC, UMBB-BC, and EBB-BC algorithms are employed, it is found that a total of 1069, 1476, and 724 structural analyses are performed, respectively. This implies that using UBB-BC, UMBB-BC, and UEBB-BC algorithms, the amount of saving in the number of structural analyses is 95.72% and 94.1%, and 97.1%, respectively. As presented in Table 4.4, using the BB-BC algorithm without UBS the total computational time is 194.44 hrs; whereas UBS integrated algorithms, namely UBB-BC, UMBB-BC, and UEBB-BC, perform the design optimization of the frame only in 8.31, 11.48, and 5.63 hrs, respectively. It should be noted that since the MBB-BC and EBB-BC algorithms also perform 25000 structural analyses during the optimization, the CPU time of these algorithms is similar to that of reported for the BB-BC algorithm and is not repeated here. Considering the examples with 6 loads (i.e. D, L,  $E_x$ ,  $E_{ex}$ ,  $E_y$ , and  $E_{ey}$ ), and 10 load combinations, the computational times obtained using the UBS are promising.

Here, it is observed that in the optimum design located by the UEBB-BC algorithm the maximum interstory drift is 97.58% of the allowable limit value; the maximum lateral displacement of the top story is 68.35% of the allowable displacement; and the load to capacity ratio for the most critical frame member is 0.97.



(a)

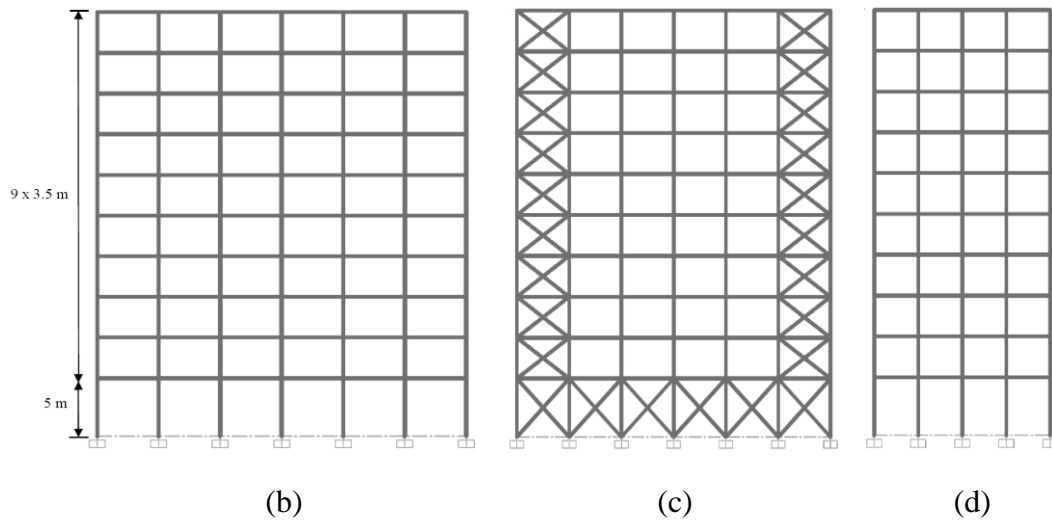


Figure 4.7: 1026-member steel frame, (a) 3-D view (b) side view of frames 2, 3, and 4 (c) side view of frames 1 and 5 (d) side view of frames A, B, C, D, E, F, and G (e) plan view

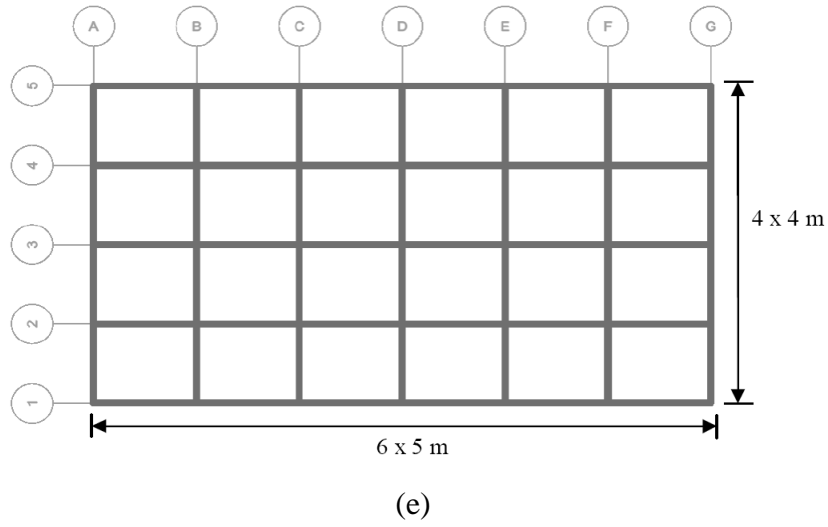


Figure 4.7 (continued)

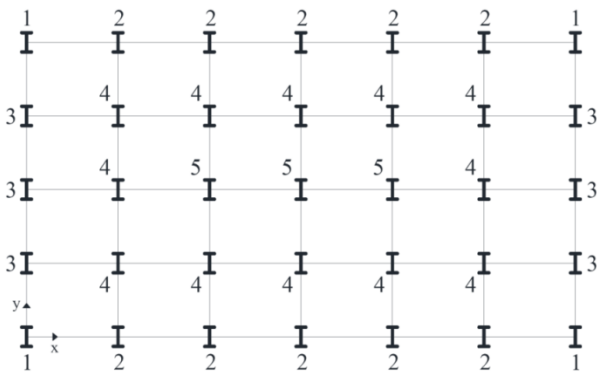


Figure 4.8: Columns grouping of 1026-member steel frame in plan level

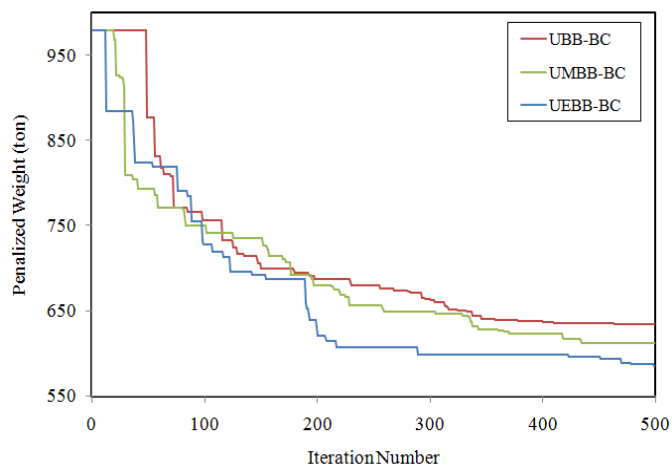


Figure 4.9: Optimization histories of the 1026-member steel frame example



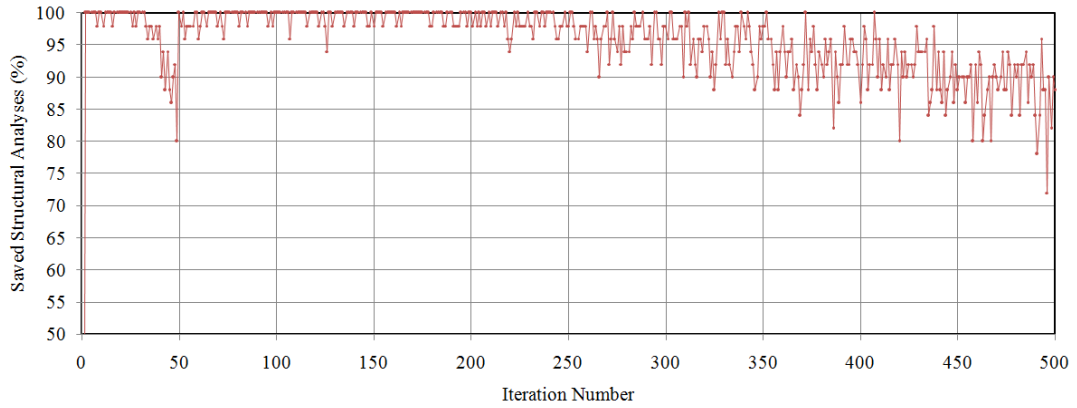


Figure 4.10: Saving in structural analyses using the UBB-BC algorithm in the 1026-member steel frame example

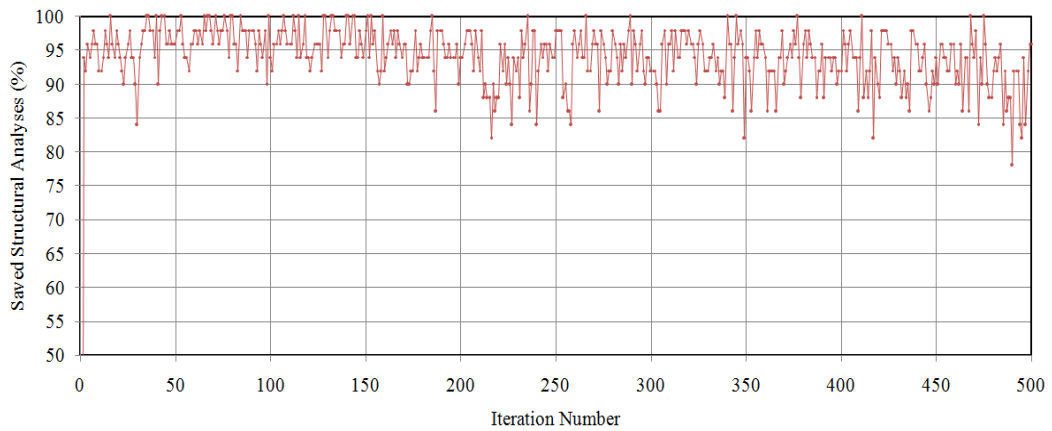


Figure 4.11: Saving in structural analyses using the UMBB-BC algorithm in the 1026-member steel frame example

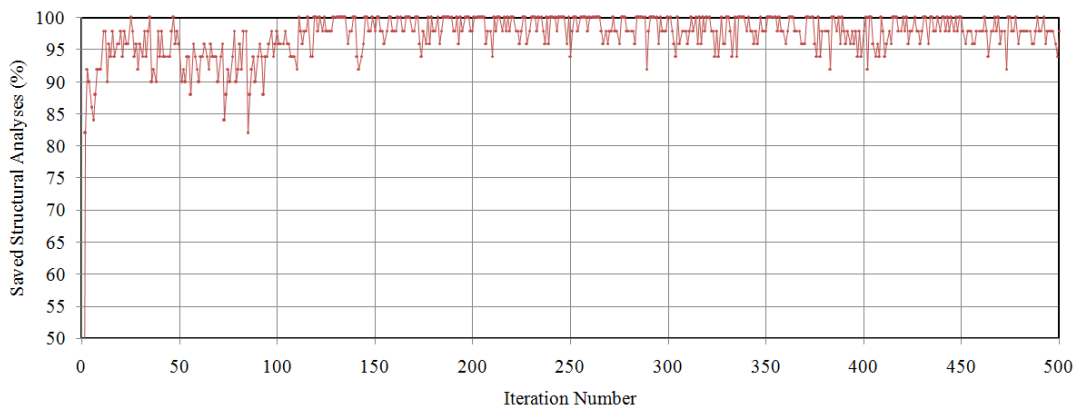


Figure 4.12: Saving in structural analyses using the UEBC-BC algorithm in the 1026-member steel frame example

Table 4.3. Optimum designs obtained for 1026-member steel frame

Stories	Groups	UBB-BC	UMBB-BC	UEBB-BC
1	CG <sub>1</sub> *	W27X258	W24X492	W33X201
	CG <sub>2</sub>	W27X161	W27X146	W24X146
	CG <sub>3</sub>	W27X102	W21X101	W24X104
	CG <sub>4</sub>	W27X146	W27X161	W40X174
	CG <sub>5</sub>	W27X146	W27X258	W40X321
	IB*	W27X84	W21X44	W27X84
	OB*	W27X84	W27X84	W27X84
	BR*	W27X94	W30X90	W18X76
2-4	CG <sub>1</sub>	W27X258	W21X201	W36X328
	CG <sub>2</sub>	W27X146	W24X162	W36X245
	CG <sub>3</sub>	W27X84	W24X131	W36X135
	CG <sub>4</sub>	W27X102	W40X174	W33X118
	CG <sub>5</sub>	W27X114	W27X102	W44X262
	IB	W27X84	W27X84	W16X26
	OB	W27X84	W30X90	W36X135
	BR	W27X84	W40X149	W21X62
5-7	CG <sub>1</sub>	W27X161	W40X235	W27X258
	CG <sub>2</sub>	W27X114	W24X131	W18X106
	CG <sub>3</sub>	W27X84	W30X90	W33X130
	CG <sub>4</sub>	W27X84	W18X86	W27X94
	CG <sub>5</sub>	W30X99	W14X90	W24X192
	IB	W27X84	W21X44	W21X44
	OB	W27X84	W30X108	W21X73
	BR	W27X94	W33X118	W30X90
8-10	CG <sub>1</sub>	W27X84	W36X194	W18X86
	CG <sub>2</sub>	W27X146	W27X146	W21X50
	CG <sub>3</sub>	W27X84	W40X174	W36X135
	CG <sub>4</sub>	W27X84	W21X62	W33X201
	CG <sub>5</sub>	W27X84	W24X76	W30X108
	IB	W27X84	W14X30	W21X57
	OB	W27X84	W16X31	W16X26
	BR	W27X84	W33X118	W18X76
Weight (ton)		634.12	612.05	584.93

\*CG denotes column group with respect to Figure 4.8, IB: inner beams, OB: outer beams, BR: bracings

Table 4.4. Computational details for 1026-member steel frame

<b>Computational Details</b>	<b>BB-BC</b>	<b>UBB-BC</b>	<b>UMBB-BC</b>	<b>UEBB-BC</b>
Generated Individuals	25000	25000	25000	25000
Analyzed Individuals	25000	1069	1476	724
CPU Time (hrs.)	194.44	8.31	11.48	5.63
Saving in Structural Analyses (%)	0	95.72	94.1	97.1

In order to visualize the performance of the UBS, especially in iterations with high savings of structural analyses, the current best design of the 35th iteration of UMBB-BC algorithm located for the 1026-member frame during the numerical investigations is selected as a typical current best solution. The penalized weight of the design namely 829.86 ton determines the upper bound. Here, using the UBB-BC, UMBB-BC and UEBB-BC algorithms, a new population of size 50 is generated for each algorithm. Next, the net weights of generated individuals are computed and compared with the upper bound value to detect the non improving individuals. As shown in Figure 4.13 , all the 50 individuals generated using the UBB-BC algorithm have net weights greater than the upper bound value. Hence, none of the individuals will be analyzed; and a saving of 50 structural analyses is obtained.

In case of the UMBB-BC algorithm, it is seen from Figure 4.14 that only one individual with a net weight of 827.49 ton remains below the upper bound limit. Similarly, as Figure 4.15 depicts, considering the UEBB-BC algorithm, only two individuals with net weights of 822.50 and 819.08 will be considered in the structural analysis stage. Therefore, the number of savings in the structural analyses for UMBB-BC and UEBB-BC algorithms are 49 and 48, respectively.

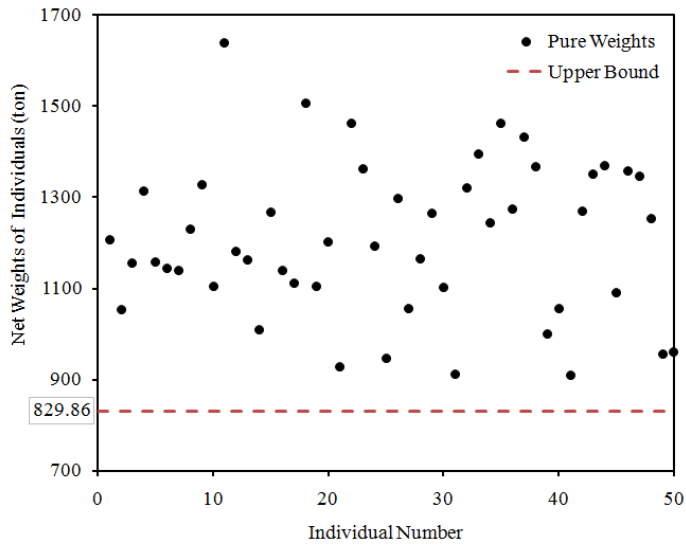


Figure 4.13: Net weights of individuals generated using the UBB-BC algorithm

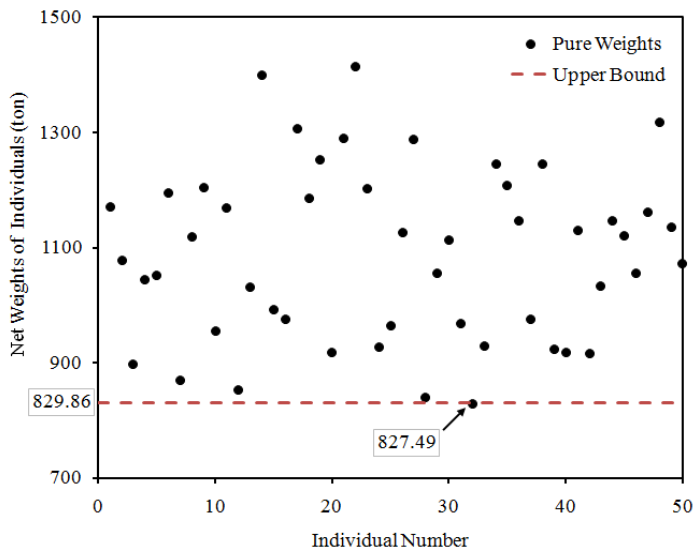


Figure 4.14: Net weights of individuals generated using the UMBB-BC algorithm

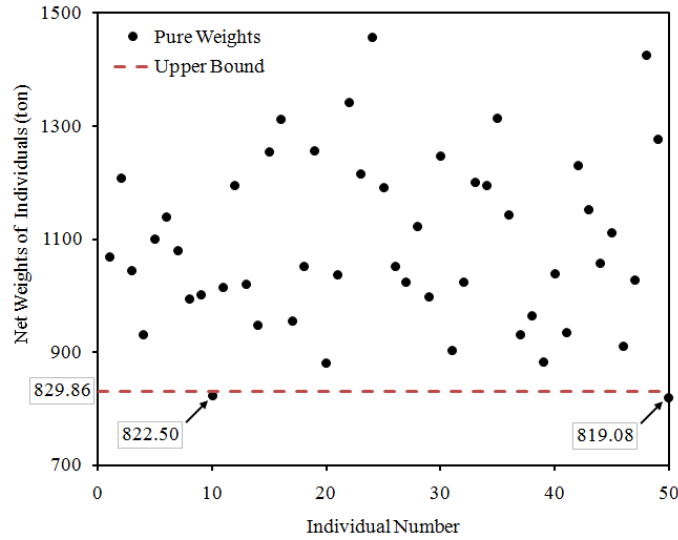


Figure 4.15: Net weights of individuals generated using the UEBC-BC algorithm

### 4.3.3 Example 3: 10-Bar Truss Structure

In order to investigate the efficiency of the UBS in optimal design of truss structures, sizing optimization of the benchmark 10-bar truss (Figure 4.16) is considered here. A total of 10 design variables are used corresponding to cross-sectional areas of the truss members. The members are subjected to the stress limits of  $\pm 25$  ksi (172.369 MPa) and the displacements of all nodes in both lateral and vertical directions are limited to  $\pm 2.0$  in (5.08 cm). For the design purpose, two different load cases are considered (Figure 4.16) in two separate test cases: case-1:  $P_1 = 100$  kips (444.82 kN) and  $P_2 = 0$ , and case-2:  $P_1 = 150$  kips (667.23 kN) and  $P_2 = 50$  kips (222.41 kN). The minimum allowable cross-sectional area of each member is limited to  $0.1 \text{ in}^2$  ( $0.6452 \text{ cm}^2$ ). The density of the material is  $0.1 \text{ lb/in}^3$  ( $2767.99 \text{ kg/m}^3$ ) and the modulus of elasticity is 10,000 ksi (68,947.6 MPa).

The optimum designs found by the UBB-BC algorithm are presented in Tables 4.5 and 4.6 in comparison to the previously obtained designs using ARCGA (Koohestani and Kazemzadeh Azad 2009), ABC and MABC (Hadidi et al. 2010) algorithms. It is accentuated that producing a solution for the problem which is better than the previously reported ones is not a matter of interest here. This could be achieved by

implementing the algorithm for a large number of iterations. Instead, the purpose is to demonstrate that the proposed upper bound strategy integrated optimization algorithm is able to locate an acceptable solution that has some level of comparability with the formerly reported ones by implementing a reasonable (fewer) number of structural analyses. In addition, it is intended to investigate the effectiveness of the proposed strategy in reducing the computational burden by comparing the number of structural analyses required by the UBB-BC algorithm and the standard version, i.e. BB-BC algorithm. To this end the UBB-BC algorithm is executed until the termination condition is met, which is the maximum number of iterations. Since both the UBB-BC and BB-BC algorithms implement the same formulations for the search procedure, the optimum designs reported with the UBB-BC algorithm are also valid for the BB-BC algorithm. However, the number of structural analyses to reach the optimum designs will be different as a result of the upper bound strategy employed in the former. The number of structural analyses performed in UBB-BC algorithm is calculated by counting candidate solutions that undergo structural analysis. For the BB-BC algorithm, this is simply obtained by multiplying the total number of iterations ( $ite\_cnt$ ) by the population size  $\lambda$ .

Figure 4.17 shows the amount of saving in structural analyses at each iteration of the UBB-BC algorithm in both test cases. The markers on the graphs show the number of structural analyses saved out of a total number of 50 analyses at each iteration. For example a saving of 40 at an iteration of the UBB-BC algorithm means that only 10 candidate solutions are analyzed and the remaining 40 candidate solutions are removed from the population (without performing a structural analysis) at that iteration due to their violation of the upper bound limit. The cumulative number of structural analyses performed with UBB-BC and BB-BC algorithms up to each iteration number is displaced in curves given in Figure 4.18. The iteration histories for the algorithms are reproduced in Figure 4.19 which shows the variation of the penalized weight of the current best design obtained thus far in the search process versus the number of structural analyses performed. Bearing in mind that a population size of  $\lambda=50$  is employed over a maximum number of 100 iterations ( $ite\_max = 100$ ),

the number of structural analyses performed by the BB-BC algorithm is equal to 5000 in the both test cases. However, when UBB-BC algorithm is employed, it is found that a total of 2000 and 1957 structural analyses are performed in case-1 and case-2, respectively. This implies that the amount of saving in the number of structural analyses is 60% and 61% for case-1 and case-2, respectively.

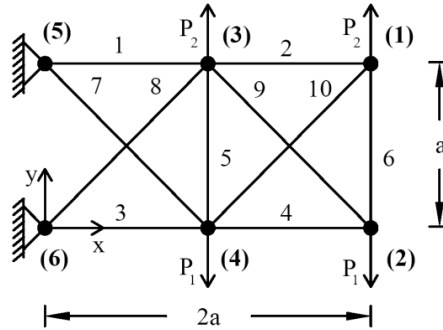


Figure 4.16: 10-bar truss structure,  $a = 360$  in.(914.4cm)

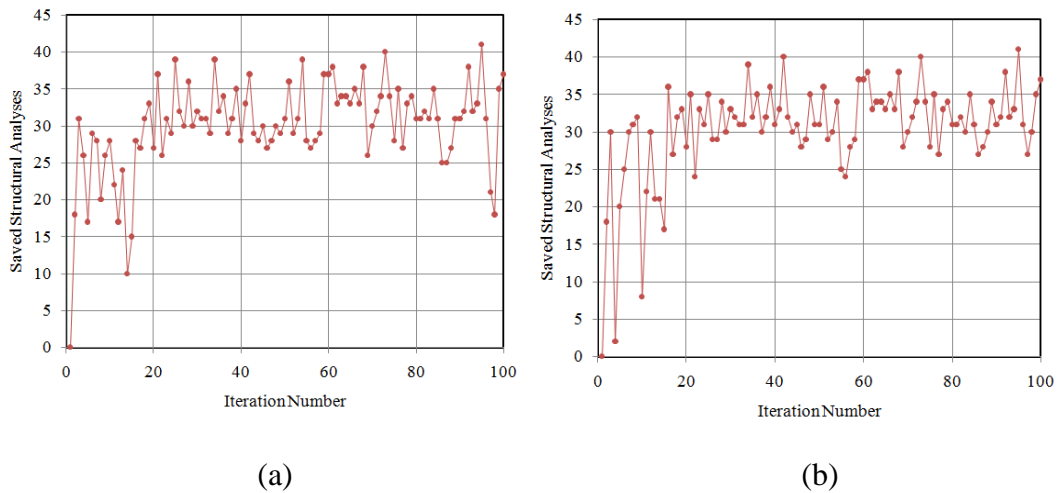
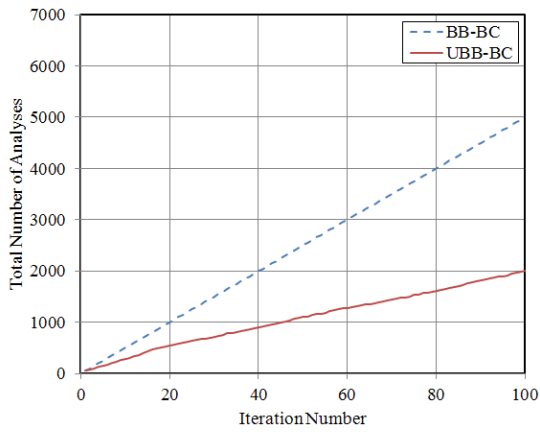
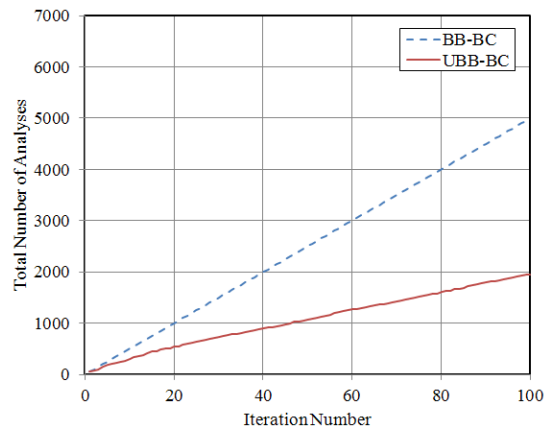


Figure 4.17: 10-bar truss example; saving in structural analyses using the UBB-BC algorithm (a) case-1 (b) case-2

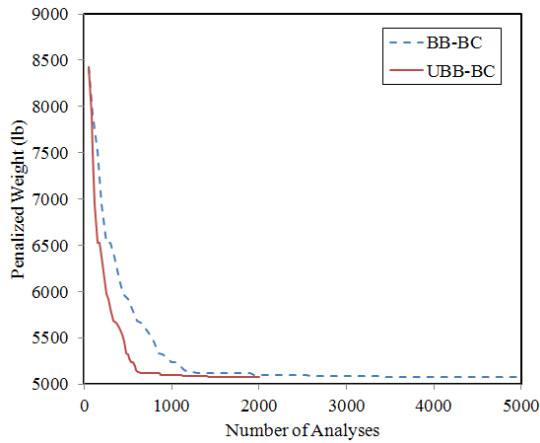


(a)

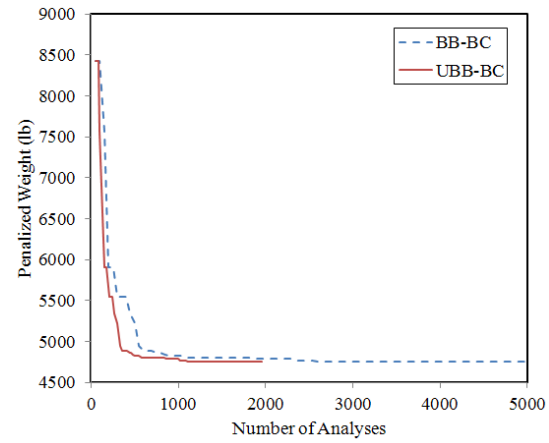


(b)

Figure 4.18: 10-bar truss example; variations of the total number of structural analyses using the UBB-BC and BB-BC algorithms (a) case-1 (b) case-2



(a)



(b)

Figure 4.19: 10-bar truss example; optimization histories of the UBB-BC and BB-BC algorithms (a) case-1 (b) case-2



Table 4.5. Optimal cross sectional areas (in.<sup>2</sup>) for 10-bar truss structure (case-1)

Design Variables	ARCGA	ABC	MABC	UBB-BC
A1	30.5984	34.3057	30.6573	28.5580
A2	0.1002	0.1	0.1	0.1
A3	23.1714	20.6728	23.0429	24.7442
A4	15.1958	14.5074	15.2821	15.7132
A5	0.1	0.1	0.1	0.1
A6	0.5409	0.6609	0.5626	0.6015
A7	7.4625	7.8696	7.4721	7.2635
A8	21.0346	20.3461	21.0084	20.8941
A9	21.5182	22.0232	21.5094	22.0193
A10	0.1	0.1	0.1	0.1
Weight (lb)	5060.9	5095.33	5060.97	5073.1
Structural Analyses	10000	20000	20000	2000

Table 4.6. Optimal cross sectional areas (in.<sup>2</sup>) for 10-bar truss structure (case-2)

Design Variables	ARCGA	ABC	MABC	UBB-BC
A1	23.5986	24.8143	23.6383	26.812
A2	0.1009	0.1	0.1	0.1
A3	25.1175	26.0480	25.3237	26.9013
A4	14.5383	14.8772	14.4108	18.0765
A5	0.1001	0.1	0.1001	0.1
A6	1.9713	2.0055	1.9707	1.995
A7	12.3923	12.4467	12.3781	11.7854
A8	12.7439	12.6835	12.7739	10.9272
A9	20.3697	18.8669	20.2678	18.2826
A10	0.1	0.1	0.1	0.1
Weight (lb)	4677.24	4691.07	4677.06	4755.7
Structural Analyses	10000	20000	20000	1957

### 4.3.4 Example 4: 45-Bar Truss Bridge

In this example sizing optimization of the 45-bar truss bridge shown in Figure 4.20 is carried out. Here, nine vertical loads of 10 kips (44.48 kN) are applied at nodes 3, 5, 7, 9, 11, 13, 15, 17 and 19. The stress limit is 30 ksi (206.843 MPa) in both tension and compression for all the members of the structure. The displacements of all nodes in both lateral and vertical directions are limited to  $\pm 2.0$  in (5.08 cm). The material density is  $0.283 \text{ lb/in}^3$  ( $7833.41 \text{ kg/m}^3$ ) and the modulus of elasticity is 30,000 ksi (206,842.8 MPa). For design purpose, the members of the structure are linked into 23 groups (sizing design variables) taking into account the symmetry of the structure. The lower bound of all sizing variables is  $0.1 \text{ in}^2$  ( $0.6452 \text{ cm}^2$ ).

The optimum design located by the UBB-BC algorithm is given in Table 4.7 in comparison to the formerly reported result based on the MABC algorithm (Hadidi et al. 2010). Figure 4.21 presents the amount of saving in the number of structural analyses at each iteration of the UBB-BC algorithm. The cumulative number of structural analyses performed with UBB-BC and BB-BC algorithms up to each iteration number is displaced in curves given in Figure 4.22. The iteration histories for the algorithms are reproduced and compared in Figure 4.23. According to these results, a total of 15000 structural analyses required by the BB-BC algorithm is reduced to 6169, when the UBB-BC algorithm is employed, leading to a saving in the number of structural analyses as much as 59%.

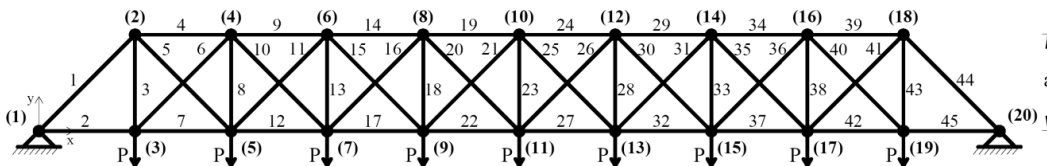


Figure 4.20: 45-bar truss bridge,  $a = 200$  in. (508 cm)

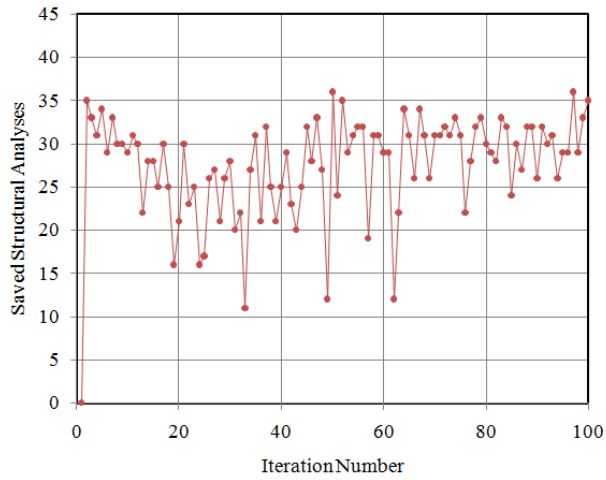


Figure 4.21: 45-bar truss bridge example; saving in structural analyses using the UBB-BC algorithm

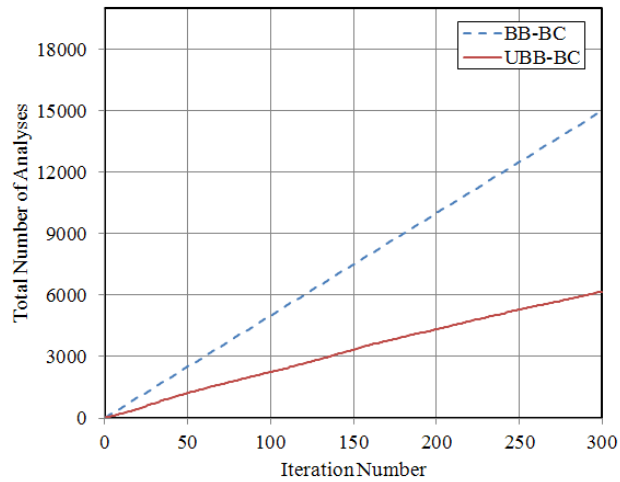


Figure 4.22: 45-bar truss bridge example; variations of the total number of structural analyses using the UBB-BC and BB-BC algorithms

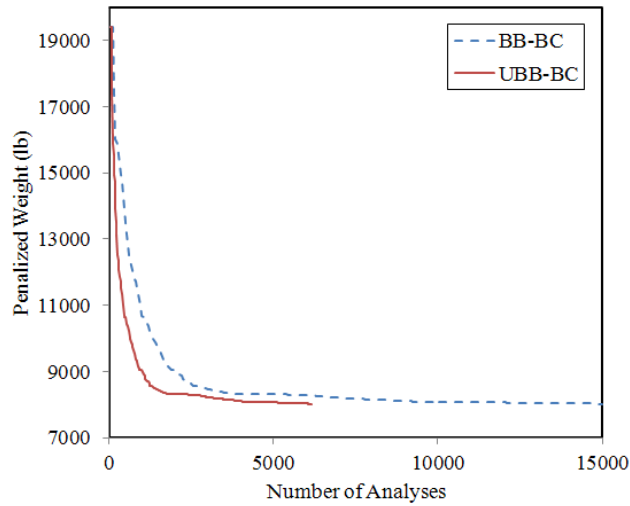


Figure 4.23: 45-bar truss example; optimization histories of the UBB-BC and BB-BC algorithms

Table 4.7. Optimal cross sectional areas (in.<sup>2</sup>) for 45-bar truss bridge

Design Variables	Members	MABC	UBB-BC	Design Variables	Members	MABC	UBB-BC
G1	1, 44	4.5996	4.6243	G13	13, 33	0.1008	0.1
G2	2, 45	3.7966	4.1520	G14	14, 29	9.5360	9.4787
G3	3, 43	3.0497	2.0432	G15	15, 31	1.2173	1.2004
G4	4, 39	3.2841	4.6309	G16	16, 30	1.4190	1.4504
G5	5, 41	0.1069	1.5864	G17	17, 32	2.5513	2.4293
G6	6, 40	3.9279	2.1517	G18	18, 28	0.1	0.1
G7	7, 42	0.9649	2.4574	G19	19, 24	11.5439	11.6481
G8	8, 38	1.2133	0.1033	G20	20, 26	1.2807	1.4056
G9	9, 34	7.6553	7.3455	G21	21, 25	0.101	0.1
G10	10, 36	2.1993	2.0886	G22	22, 27	3.7598	3.4959
G11	11, 35	1.1929	1.2330	G23	23	0.1017	0.7221
G12	12, 37	0.1001	0.1				
Weight (lb)						7968.95	8019.86
Structural Analyses						40000	6169

### 4.3.5 Example 5: 120-Bar Truss Dome

Design optimization of the 120-bar truss dome shown in Figure 4.24 is carried out in Soh and Yang (1996) using both shape and sizing design variables. Here, only the sizing optimization of the structure is performed. The dome is subjected to vertical loading at all unsupported nodes. The loads are taken as -13.49 kips (-60 kN) at node 1, -6.744 kips (-30 kN) at nodes 2 to 13, and -2.248 kips (-10 kN) in the rest of the nodes. The minimum allowable cross sectional area of each member is limited to 0.775 in<sup>2</sup> (5 cm<sup>2</sup>). For design purpose the allowable tensile stress is taken as 0.6  $F_y$  and the compressive stress constraint  $\sigma_i^b$  of the  $i$ -th member is computed as follows (AISC-ASD 1989):

$$\sigma_i^b = \begin{cases} \left[ \left( 1 - \frac{\lambda_i^2}{2C_c} \right) F_y \right] / \left( \frac{5}{3} + \frac{3\lambda_i}{8C_c} - \frac{\lambda_i^3}{8C_c^3} \right) & \text{for } \lambda_i < C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \geq C_c \end{cases} \quad (4.3)$$

where  $F_y$  is the yield stress of steel,  $E$  is the modulus of elasticity,  $\lambda_i$  is the slenderness ratio ( $\lambda_i = kL_i / r_i$ ),  $k$  is the effective length factor,  $L_i$  is the length of the  $i$ -th member,  $r_i$  is the radius of gyration, and  $C_c = \sqrt{2\pi^2 E / F_y}$ . Here, the material density is 0.288 lb/in<sup>3</sup> (7971.81 kg/m<sup>3</sup>),  $F_y = 58$  ksi (400 MPa),  $E = 30,450$  ksi (210,000 MPa). In this example, two different displacement constraints are considered in two separate test cases: case-1: no displacement constraints are imposed; and case-2: the displacements of all nodes in directions  $x$ ,  $y$  and  $z$  directions are limited to  $\pm 0.1969$  in ( $\pm 0.5$  cm).

The optimum designs found by the UBB-BC algorithm are given in Tables 4.8 and 4.9 together with the previously reported solutions based on the FA (Kazemzadeh Azad and Kazemzadeh Azad 2011), ABC and MABC (Hadidi et al. 2010) techniques.

Figure 4.25 presents the amount of saving in structural analyses at each iteration of the UBB-BC algorithm in both test cases. The cumulative number of structural analyses performed with UBB-BC and BB-BC algorithms up to each iteration number is displaced in curves given in Figure 4.26. The iteration histories for the algorithms are reproduced and compared in Figure 4.27. Noting that a population size of  $\lambda = 50$  is employed over a maximum number of 300 iterations ( $ite\_max = 300$ ), the number of structural analyses performed by the BB-BC algorithm is equal to 15000 in both test cases. However, when UBB-BC algorithm is employed, a total of 7739 and 7802 structural analyses are performed in case-1 and case-2, respectively. Hence the amount of saving in the number of structural analyses is 48% for the both cases.

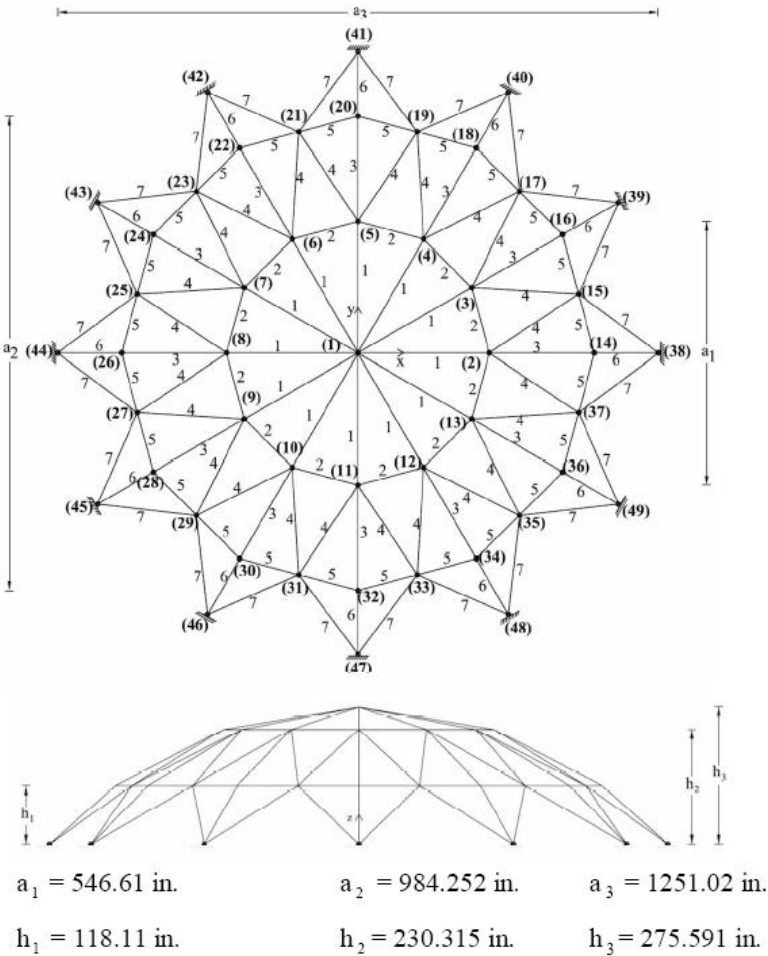
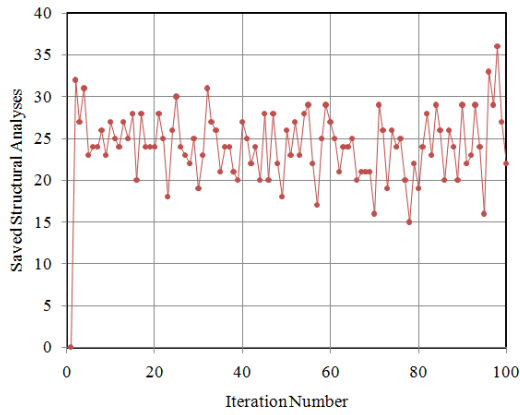
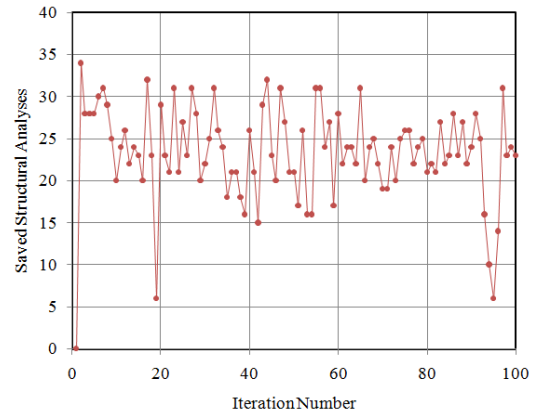


Figure 4.24: 120-bar truss dome

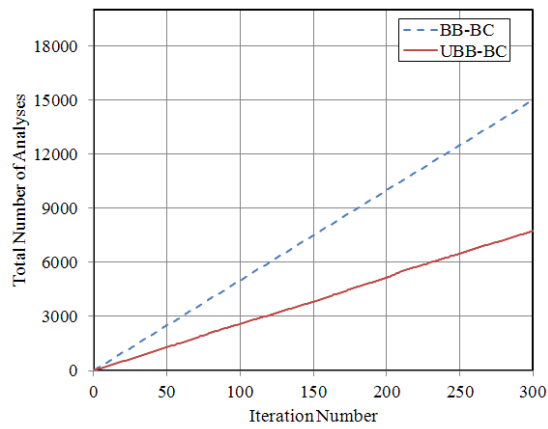


(a)

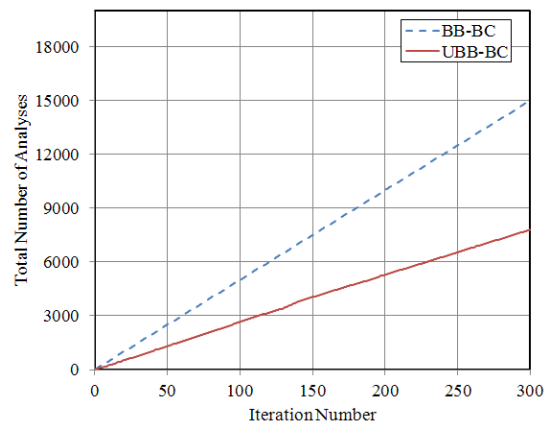


(b)

Figure 4.25: 120-bar truss dome example; saving in structural analyses using the UBB-BC algorithm (a) case-1 (b) case-2



(a)



(b)

Figure 4.26: 120-bar truss dome example; variations of the total number of structural analyses using the UBB-BC and BB-BC algorithms (a) case-1 (b) case-2

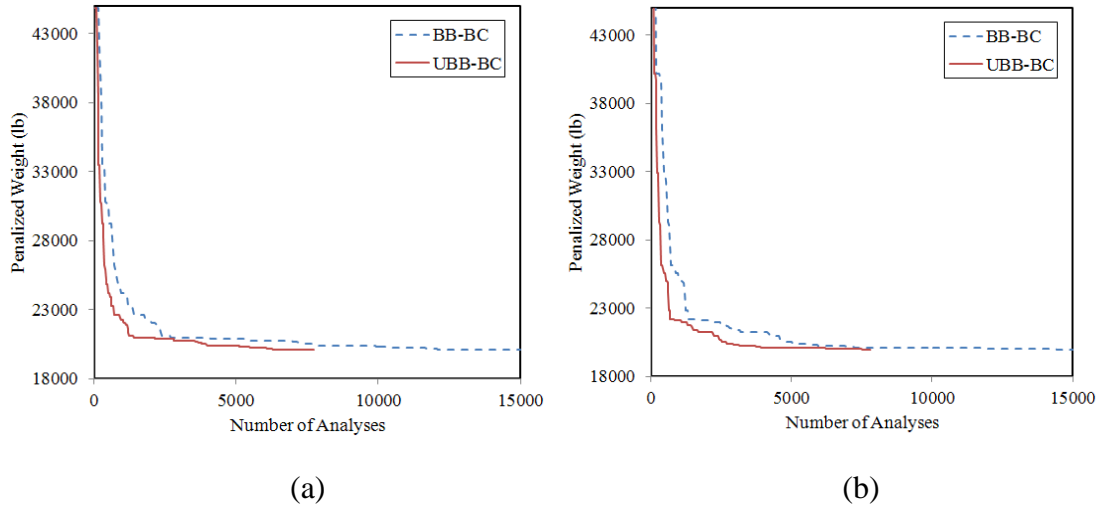


Figure 4.27: 120-bar truss example; optimization histories of the UBB-BC and BB-BC algorithms (a) case-1 (b) case-2

Table 4.8. Optimal cross sectional areas (in.<sup>2</sup>) for 120-bar truss dome (case-1)

Design Variables	FA	ABC	MABC	UBB-BC
A1	3.3293	3.2977	3.2976	3.3072
A2	2.4384	2.3964	2.3964	2.5319
A3	4.0168	3.8737	3.8736	3.9035
A4	2.5918	2.5710	2.5710	2.5863
A5	1.1823	1.1514	1.1513	1.1554
A6	3.4513	3.3324	3.3323	3.8227
A7	2.7854	2.7850	2.7848	2.7746
Weight (lb)	20016.67	19707.19	19706.62	20122.71
Structural Analyses	15000	20000	20000	7739

It is worth mentioning that, traditionally, metaheuristic algorithms tend to perform a local search in the last iterations to increase the quality of final solutions. Although this entails numerous iterations, generally, the improvements in the solutions are not significant with respect to the large number of structural analyses performed. The UBS removes this burden and makes it possible to perform a computationally efficient local search through avoiding unnecessary analyses. This can be considered as one of



the beneficial features of the UBS, which enables the optimization algorithms to perform a fine tuning in the last iterations with considerably less number of structural analyses. Although, here, only BB-BC algorithm based approaches are employed for demonstrating the usefulness of the UBS, recently other UBS integrated metaheuristics are also implemented and successfully applied to structural design optimization problems (Hasançebi et al. 2013, Kazemzadeh Azad and Hasançebi 2014).

Table 4.9. Optimal cross sectional areas (in.<sup>2</sup>) for 120-bar truss dome (case-2)

Design Variables	FA	ABC	MABC	UBB-BC
A1	3.3005	3.2984	3.2985	3.3318
A2	2.7481	2.7894	2.7928	2.7983
A3	3.9036	3.8743	3.8748	3.8947
A4	2.5713	2.5719	2.5719	2.5719
A5	1.2889	1.1549	1.1501	1.1680
A6	3.4089	3.3341	3.3328	3.3498
A7	2.8150	2.7860	2.7838	2.7861
Weight (lb)	20125.35	19908.03	19901.42	19982.91
Structural Analyses	15000	20000	20000	7802

#### 4.4 Summary

In this chapter the UBS is proposed as a novel strategy to reduce the number of structural analyses in all metaheuristic techniques that employ a  $\mu + \lambda$  selection scheme in their algorithmic models. The BB-BC algorithm as well as its two enhanced variants are integrated with the UBS for optimum design of steel structures. Based on the UBS the penalized weight of the current best solution is considered as the upper bound limit for the net weights of the newly generated candidate solutions. Accordingly, the new candidate solutions with net weights greater than this limit are excluded from the structural analysis stage.

The numerical results obtained through design optimization instances of skeletal structures, clearly reveal that the UBS is capable of reducing the computational effort required to approach a reasonable design. The significant reduction in computational effort achieved via this strategy can pave the way for practical optimum design of large scale steel structures using regular computers.

## CHAPTER 5

### GUIDED STOCHASTIC SEARCH TECHNIQUE FOR DISCRETE SIZING OPTIMIZATION OF STEEL TRUSSES

In the present chapter a guided stochastic search (GSS) technique is proposed for code based design optimization of steel trusses. The GSS offers a stochastic procedure where the optimization process is guided by the principle of virtual work and response computations of the generated designs, resulting in an efficient and rapid search. Here, first the proposed technique is implemented for handling discrete sizing optimization problems of truss structures having a single displacement constraint under a single load case. Next, the GSS is further improved for tackling a more general class of truss optimization problems subject to multiple displacement constraints and load cases.

#### 5.1 GSS for Single Displacement Constraint and Single Load Case

##### 5.1.1 Introduction

In general common stochastic search techniques perform random moves in the design space using strategies borrowed from nature to locate the optimum solution using a single or a population of candidate designs. The basic difference between these techniques lies in the way that they decide on the next move in the design space. This can significantly affect both the quality of final solution as well as the computational time of the optimization process. Therefore, it becomes vital to make an effective search in a timely manner by developing some robust strategies as a guide to stochastic moves in the design space. Since response computations are performed for generated designs at each iteration of a stochastic optimization algorithm, it is possible

to utilize such valuable information collected during response computations to guide the optimization process.

In this chapter a design-driven optimization technique named GSS is proposed for discrete sizing optimization of steel truss structures (Kazemzadeh Azad et al. 2014a). Here, the information provided through the structural analysis and design check stages are utilized for handling strength constraints. Besides, the well-known principle of virtual work is employed to detect the most effective structural members for satisfying displacement constraints. The weight minimization of a structure is then performed using an integrated approach wherein both strength and displacement criteria are taken into account for reduction of the member sizes along the way the aforementioned constraints are handled. The performance of the proposed method is investigated using four real-size steel truss structures with 25, 130, 392 and 354 sizing variables designed for minimum weight according to AISC-LRFD (1994) specifications. The comparison of numerical results obtained using the GSS to those of different metaheuristic algorithms indicates that the proposed technique is able to locate promising solutions using lesser computational effort.

### **5.1.2 Application of the Principle of Virtual Work**

In order to guide the design optimization process, member wise information should be computed and utilized for determining a useful direction of search. In the case of strength criteria this information can be provided from the load capacity of members. However, in the case of displacement criteria one requires a measure to identify contribution of each structural member to the total displacement for each considered direction, referred to as displacement participation factor (DPF) (Charney 1991). In the present study a procedure based on the principle of virtual work is used to determine the DPF of each member in a truss structure. In order to compute the DPF of a truss member in the  $k$ -th direction of the  $j$ -th joint, in addition to a common structural analysis performed under the applied real loads, the truss structure should be analyzed under a unit load (virtual load) applied at the same joint and in the same

direction as well. Next, the DPF of the  $i$ -th member in the  $k$ -th direction,  $DPF_{i,k}$ , can be calculated as follows.

$$DPF_{i,k} = \frac{P_i^r P_i^u}{E_i A_i} L_i \quad (5.1)$$

In Eq. (5.1),  $P_i^r$  and  $P_i^u$  are the internal axial forces in the  $i$ -th member under the real and unit load, respectively; and  $E_i$ ,  $A_i$ , and  $L_i$  are the modulus of elasticity, cross-sectional area, and length of the  $i$ -th truss member, respectively. Accordingly, the total displacement of the structure in the  $k$ -th direction,  $\delta_{Total,k}$ , can be computed by summing the DPF values as follows.

$$\delta_{Total,k} = \sum_{i=1}^{N_m} DPF_{i,k} \quad (5.2)$$

Park and Park (1997) considered the DPF for optimum design of high-rise buildings. However, a more general approach can be achieved by taking into account the volume of structural members. The concept of sensitivity index (SI) (Charney 1993) given in Eq. (5.3) provides a more rational measure for identifying the effectiveness of members in satisfying the displacement constraints,

$$SI_{i,k} = \frac{DPF_{i,k}}{V_i} \quad (5.3)$$

where  $SI_{i,k}$  and  $V_i$  are the sensitivity index in the  $k$ -th direction, and volume of the  $i$ -th member, respectively. Once the SI value of each member is calculated, these values can be employed to decide on the structural members that should be increased in size to satisfy the required displacement limits using the approach provided in the next section.

### 5.1.3 The GSS Algorithm

The GSS works on the basis of guiding the optimization process using DPF as well as the information collected during the structural analysis and design stages. To this end, first the critical truss elements that have the highest impact on the response of a given structure should be detected; and next their sizes should be changed appropriately. The whole process should be guided such that the final minimum weight design leads to an optimum or reasonable near-optimum solution, satisfying all the predefined design constraints. The following steps outline the main procedure in the implementation of the GSS.

**Step 1. Initialization:** The optimization process with the GSS initiates with a randomly generated design. This type of initialization, which is similar to the initialization of metaheuristic techniques with a population size of one, indicates the independency of the algorithm on the starting point of the design optimization process. It is worth mentioning that, since the members are to be adopted from a predefined list of sections, while generating a new candidate design, the design variables created outside of the allowable range are moved back to their lower/upper limits.

**Step 2. Evaluation under real loading:** The generated design is evaluated next under the applied real loads, where structural analysis of the design is carried out with the set of steel sections adopted for the design variables, and the force and deformation responses are obtained. Once the internal forces and nodal displacements are known, the structure undergoes a design check where the amounts of strength and displacement violations are calculated. Evaluation stage reveals the quality of the generated design at each iteration.

**Step 3. Evaluation under virtual loading:** In addition to the abovementioned evaluation of the design under the real loads, an additional analysis is carried out at this step to capture structural response of the design under the virtual loading. This

step is required in order to compute the SI values using Eq. (5.3).

**Step 4. Detection of critical members:** In the GSS both increase and decrease in member sizes are carried out simultaneously. Hence, it is required to identify two main groups of members. The first group includes those members which should be increased in size for eliminating strength and/or displacement violations. Oppositely, the second group contains the members which should be decreased in size in line with the weight minimization objective of the optimization process. These critical member groups are detected in this step as follows.

**Step 4.1. Increase-group for constraint satisfaction:** The increase-group (IG) contains members that are to be increased in size in an effort to eliminate constraint violations. This group can further be divided into two subgroups as IG<sub>s</sub> and IG<sub>d</sub> based on the type of constraint violations, where the former is used for satisfying the strength constraints while the latter is adopted to handle the displacement criteria. All the members violating the strength constraints are directly included in the IG<sub>s</sub> subgroup. These members can simply be detected based on their load to capacity ratios (*LCRs*), Eq. (5.4), which exceeds 1.0 for under designed members.

$$LCR = \left[ \frac{P_u}{\phi P_n} \right] \quad (5.4)$$

On the other hand, in order to detect the IG<sub>d</sub> members SI values described through Eq. (5.3) should be computed. Once SI values are calculated, the most critical members for reducing the displacement violations, i.e. members with higher SI values, are identified. Here, the number of IG<sub>d</sub> members is determined using an adaptive ratio parameter R<sub>d</sub> (see step 6) based on the total number of design variables. For instance, for a truss structure with 100 sizing design variables if R<sub>d</sub> is set to 0.1 in an arbitrary iteration of the GSS, then the number of IG<sub>d</sub> members will be equal to 10. This implies that only 10 members with highest SI values are to be increased in size to

satisfy the displacement criteria.

**Step 4.2. Decrease-group for weight reduction:** The decrease-group ( $DG_w$ ) contains the elements that are to be decreased in size with the purpose of achieving a certain weight reduction. Selection of  $DG_w$  members is carried out to identify those members which seem to be oversized under the applied loads. For this purpose, first the most oversized members considering the strength criteria are detected using LCR values. Next, these members are further evaluated based on their SI values in order to select those that have the least impact on the displacement criteria. Similar to  $IG_d$  members, the number of  $DG_w$  members is also determined in conjunction with an adaptive ratio parameter  $R_w$  (see step 6) based on the total number of design variables. The  $R_w$  ratio determines the number of most oversized members having the least effect on the displacement constraints. In case the most oversized members detected are also amongst the least effective members on the displacement constraint, they are chosen as  $DG_w$  members. Hence, the selected  $DG_w$  members will be those oversized members for which the size reduction will have the least effect on the displacement of the structure in the direction of interest. For instance, for a truss structure with 100 sizing design variables if  $R_w$  is set to 0.1 in an arbitrary iteration of the GSS, then at most the number of  $DG_w$  members will be equal to 10. Here, by decreasing the size of  $DG_w$  members it is expected to have no or minimum violation in the displacement constraints while reducing the total weight of the truss structure.

**Step 5. Stochastic member resizing:** The resizing strategy used in this study is based on a simple stochastic approach, where the  $IG_s$  and  $IG_d$  members are stochastically increased in size based on a predefined maximum incremental step size, which is taken as 5 for most practical problems. A random move towards larger sections is then performed for each IG member using Eq. (5.5),

$$I_i^{new} = I_i^{pre} + Rand^{IG} \quad (5.5)$$



where  $I_i^{pre}$  is the value of a design variable in the previous iteration;  $I_i^{new}$  is the new value of the design variable, representing the sequence number of the new section adopted for an IG member in the current iteration; and  $Rand^{IG}$  is an integer random number selected between 1 and 5 according to a uniform distribution. It follows that in the new design each IG member is replaced stochastically by any of the five stronger sections next to its previous value in a profile list.

On the other hand, the  $DG_w$  members are stochastically decreased in size based on a predefined bound for the reduction step size, which is taken as 3 for most practical problems. A random move towards smaller sections is then performed for each of the  $DG_w$  members as follows:

$$I_i^{new} = I_i^{pre} - Rand^{DG_w} \quad (5.6)$$

where  $Rand^{DG_w}$  is an integer random number selected between 1 and 3 according to a uniform distribution.

The resizing strategy followed in this study provides stochastic yet guided moves in the design space to approach the optimum. It is worth mentioning that the boundary values adopted for increase and decrease in the size of members are determined based on extensive numerical experiments. Here, it is attempted to avoid large changes in the structure of candidate designs specially when decreasing the size of members to prevent intensive constraint violations.

**Step 6. Updating the rate of member resizing:** This step is implemented to determine the number of structural members considered for resizing at each iteration. In this regard an adaptive approach is employed based on the feasibility of the generated design at each iteration using some rules. The first rule is that  $IG_s$  members, those violating strength constraints, are all increased in size due to the high importance of strength criteria for producing acceptable designs. Secondly, as

mentioned before, not all but rather a certain percentage of structural members are selected as  $DG_w$  and  $IG_d$  members for section change based on the associated  $R_w$ , and  $R_d$  ratios. Both of these parameters are initially set to a minimum value of 0.1, implying that only 10% of the members (design variables) are subjected to a section decrease due to oversized strength capacity, and another 10% of the members will be considered for a section increase due to violation of displacement criteria.

During the numerical investigations it was seen that some of the  $DG_w$  members eventually reach their critical sizes (i.e. the lower bounds on profile sizes) when the iterations continue, and no further improvement in the solution becomes possible using only 10% of the design variables. To avoid this problem and hence to ensure an effective weight reduction process in the following iterations, the  $R_w$  ratio is increased by  $\Delta R$  at each iteration provided that no constraint violation takes place in the generated design. It is apparent that, increasing the value of  $R_w$  results in contribution of more structural elements in weight reduction. However, in case any kind of constraint violation (either displacement or strength criteria violation) takes place in the generated design, the value of  $R_w$  is set back to its initial value to expedite the repair of the infeasible design.

In the case of  $IG_d$  members the design optimization of investigated examples has revealed that the initial value of  $R_d = 0.1$  can be kept constant throughout the optimization process, and no increment of this parameter is indeed necessary. However, in case no improvement is provided for reducing the displacement violations, the same adaptive approach used for  $R_w$  can be employed for  $R_d$  as well to increase the number of structural members contributing to satisfy the displacement criteria. It is worth mentioning that an upper bound value of  $R_{max} = 0.7$  is used for both  $R_w$  and  $R_d$  ratios in the present study.

**Step 7. Termination:** The aforementioned procedure is iteratively carried out, starting from the last generated design at the end of each iteration, until a stopping criterion is satisfied. The termination criterion can be imposed as a maximum number

of iterations or no improvement of the best design over a certain number of iterations. For the sake of clarity the flowchart of truss optimization procedure using the proposed GSS is outlined in Figure 5.1.

It should be noted that in case there are no displacement constraints present, the proposed algorithm will employ the LCR values only to guide the search and will use a similar logic to that of the classical fully stressed design (FSD) (Gallagher 1973) which is a simple yet efficient stress-ratio technique for handling such cases in practical applications. However, in these cases there will be some differences between the proposed technique and the FSD as follows. Firstly, in the GSS only a portion of elements are resized based on an adaptive approach which takes into account the feasibility of the generated solution as well. Secondly, the GSS follows a stochastic procedure which increases the global search capacity of the technique and helps the algorithm avoid local minima. Finally, in the GSS only the direction of the search is determined (i.e. increase or decrease) and the amount of resizing is carried out stochastically within a limited boundary. Since using the FSD is well established in the literature for handling problems with no displacement constraint, the main focus of this study is on developing a computationally efficient technique especially for dealing with problems having displacement constraint together with strength limitations. Although problems having no displacement constraint can be also handled by the GSS, such problems are not covered in this study.

In general, parameters of the GSS include resizing ratios (i.e.  $R_d$ , and  $R_w$ ), corresponding step size and upper bound (i.e.  $\Delta R$  and  $R_{max}$ ), as well as boundary values for stochastic resizing of members (i.e. boundary values for  $Rand^{IG}$  and  $Rand^{DG_w}$ ). Basically, since an adaptive procedure is employed in the GSS which takes into account the feasibility of the generated design, the algorithm is able to adjust the resizing ratios  $R_d$ , and  $R_w$  during the search process. Hence initial values of these parameters do not have significant effects on the performance of the algorithm. For step size of these ratios,  $\Delta R$ , using larger values than those adopted in this study

could impose abrupt changes in the structure of a generated design at each iteration resulting in an inefficient search in the solution space. Similarly taking larger values of  $R_{\max}$  than that of used in this study could result in a similar problem by allowing simulations resizing of a large number of members. Furthermore, in order to set the boundary values for  $Rand^{IG}$  and  $Rand^{DG_w}$  typically square root of the number of available sections in a profile list can provide an approximate idea. Accordingly since a list of 37 pipe sections (AISC-LRFD 1994) is used in this study, in order not to disturb the generated solutions to a great extent, smaller boundary values of 5 and 3 (compared to  $\sqrt{37}$  which gives an approximation of 6) are used for  $Rand^{IG}$  and  $Rand^{DG_w}$  based on numerical experiments by which increasing the size of members to repair an infeasible design can be carried out faster than decreasing the member sizes for weight reduction. Indeed, further studies are required for evaluating the effect of different parameter settings on the general performance of the GSS. Additionally, since parameter setting is somehow a problem dependent issue, in case of new instances, the speed of the proposed technique makes it possible to perform numerous runs in a reasonable time for setting its parameters to some appropriate values.

In addition, it should be noted that there is no limitation for application of the GSS to the instances having member grouping. In such cases the same procedure described for the GSS can be employed wherein for each group of members LCR and SI values are assigned. For instance, in a group of members the most critical member's LCR can be considered as the LCR of the group. Furthermore, in order to determine the SI value of a group, in light of Eq. (5.3) sum of the DPFs of group members should be divided by the total volume of the group members. Once LCR and SI values are obtained for member groups, these values can be used by the GSS to determine the resizing scheme.

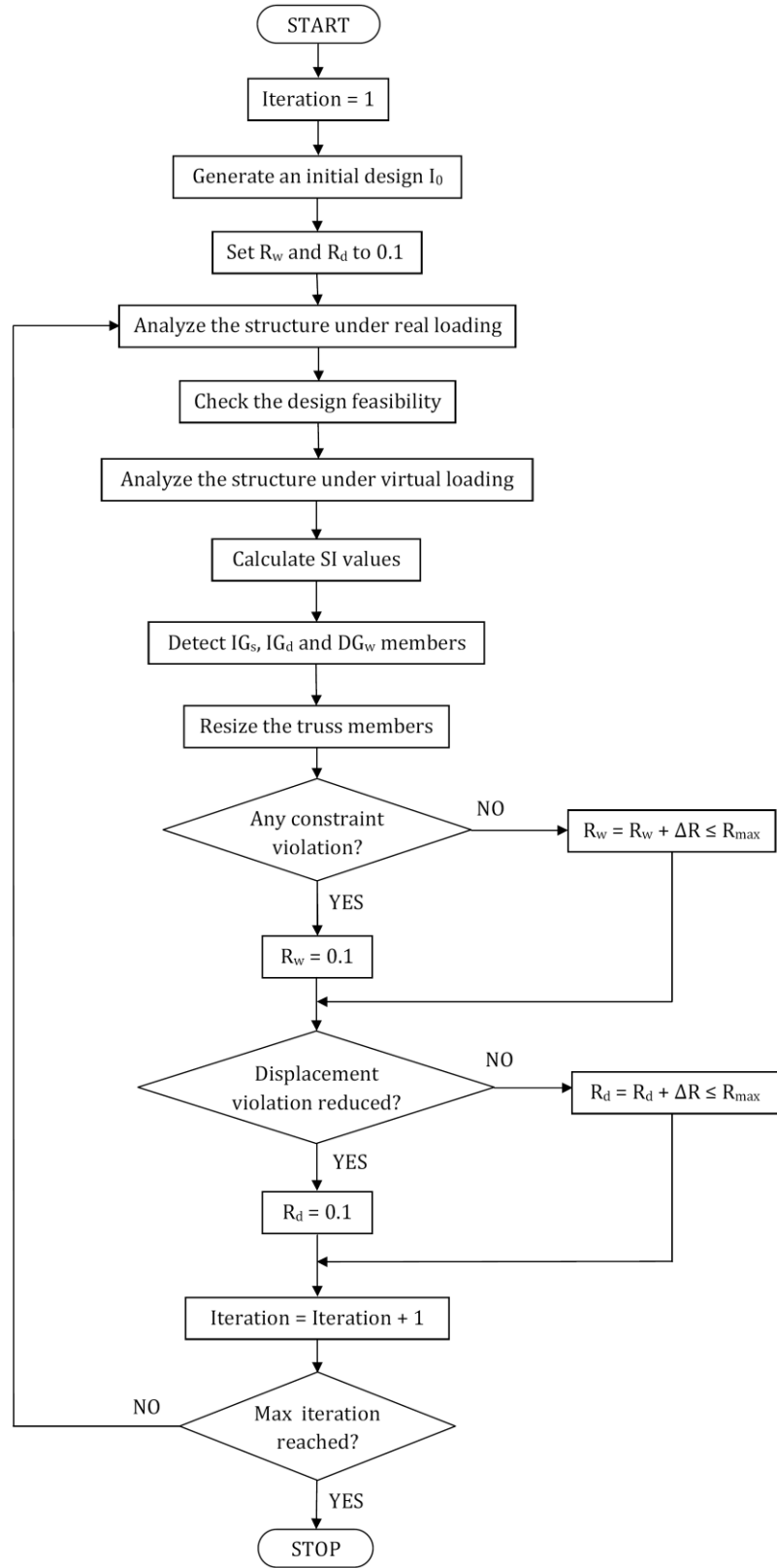


Figure 5.1: Flowchart of truss optimization process using GSS

### 5.1.4 Numerical Examples

This section covers performance evaluation of the GSS through four sizing optimization examples. The investigated instances consist of a 117-member cantilever truss, a 130-member transmission tower, a 392-member double layer grid, and a 354-member truss dome, which are challenging real-size optimization problems with 25, 130, 392 and 354 sizing design variables, respectively. The optimum designs produced for these examples using the GSS are compared to those obtained using some metaheuristics. It should be highlighted that there are a large number of metaheuristic techniques available in the literature nowadays, and the ones used for comparison purposes here are particle swarm optimization (PSO), the big-bang big-crunch (BB-BC) algorithm and two enhanced variants of the latter, namely modified (Hasançebi and Kazemzadeh Azad 2014) and exponential BB-BC (Hasançebi and Kazemzadeh Azad 2012). These algorithms are particularly selected based on their successful performances on discrete truss optimization problems. In the present study the BB-BC algorithm and its two enhanced variants are employed based on the formulations provided in chapter 3. On the other hand, PSO is implemented with a slight modification in its formulations in Hasançebi et al. (2009), such that the additional velocity term utilized therein is adopted here with a probability of 0.01 for both positive and negative changes in the velocities of particles. Due to the stochastic nature of the techniques, each design example is independently solved ten times with each technique and the computational results collected are used to carry out a performance evaluation of the techniques for comparison purposes. The value of parameter  $\Delta R$  is set to 0.1 for the first and second examples, and 0.2 for the last example. Here, the maximum number of iterations is adopted as the termination criterion of the optimization process. For all the examples, the sizing variables are to be selected from a database of 37 pipe sections, and the material properties of steel are taken as follows: modulus of elasticity ( $E$ ) = 200 GPa, yield stress ( $F_y$ ) = 248.2 MPa, and unit weight of the steel ( $\rho$ ) = 7.85 ton/m<sup>3</sup>.

#### **5.1.4.1 Example 1: 117-Member Cantilever Truss**

The steel cantilever truss depicted in Figure 5.2 is considered as the first design example. The structure is composed of 117 members and 30 joints. The truss members are linked into 25 groups, so that a design optimization problem with 25 design variables is generated. For design purpose, downward loads of 15 kN are applied at all the unsupported nodes. The vertical displacement of the truss tip is limited to 2 cm and the maximum number of iterations for the GSS is set to 200. For the PSO and BB-BC algorithms a population size of 50 individuals and a maximum number of 200 iterations are used.

Design optimization of the cantilever truss is performed, and the results obtained are presented in Table 5.1. In this example, the GSS gives a promising design weight of 5026.4 kg. The other design weights are 5026.4 kg by PSO, 5025.2 kg by standard BB-BC, 5025.2 kg by modified BB-BC, and 5026.4 kg by exponential BB-BC algorithm. Here, the GSS locates a good near optimum solution over 200 iterations by performing only 400 structural analyses. On the other hand, the abovementioned design weights are obtained by other algorithms through 10000 analyses, which is simply calculated by multiplying the population size (i.e. 50) by the maximum number of iterations (i.e. 200). The performance of the GSS can also be observed from Table 2 in terms of the worst, mean, and standard deviation of the design weights attained using a particular technique. The convergence history showing the variation of the best feasible generated design throughout the optimization process in the best run of GSS is depicted in Figure 5.3. The convergence histories of the best runs of the other algorithms are shown in Figure 5.4.

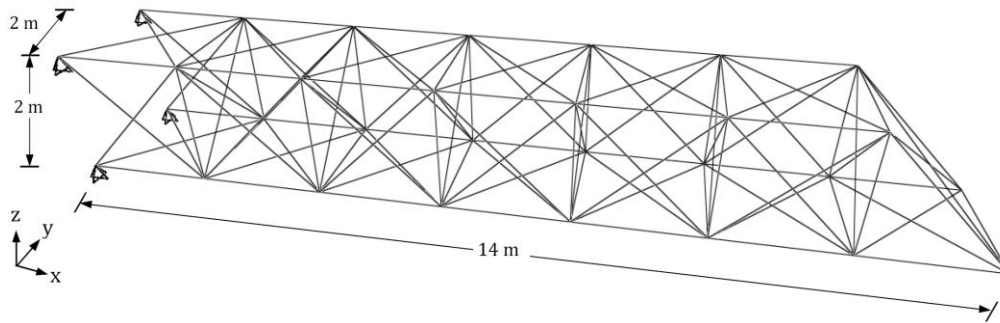


Figure 5.2: 117-member cantilever truss

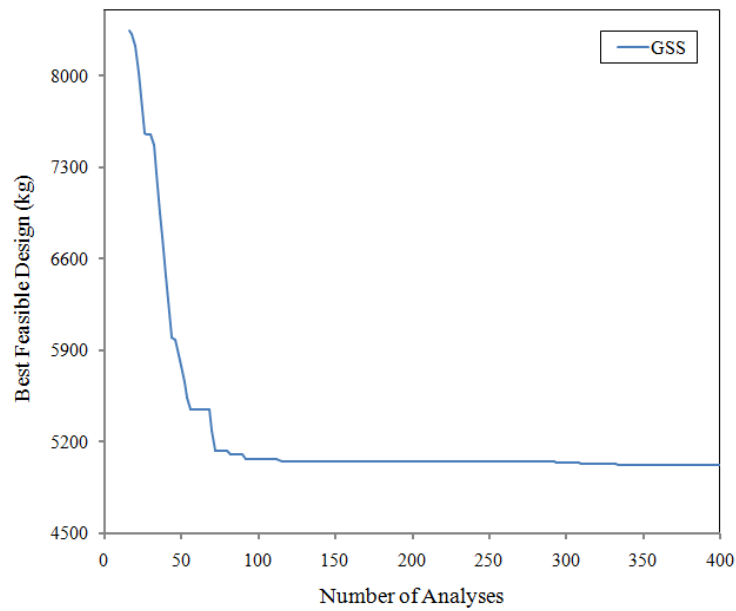


Figure 5.3: Convergence history of the best feasible generated design for 117-member cantilever truss using GSS



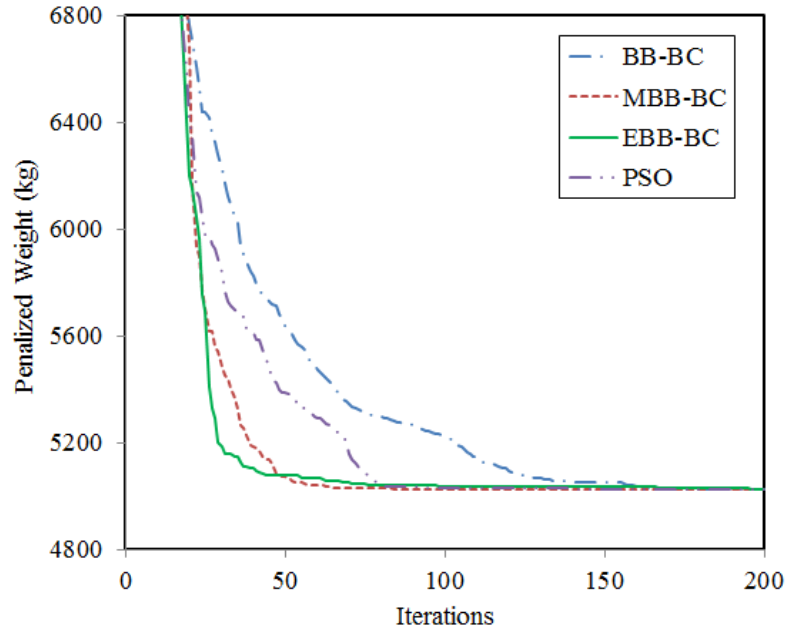


Figure 5.4: Convergence histories for 117-member cantilever truss using some metaheuristics

Table 5.1. Comparison of optimum designs for 117-member cantilever truss

Computational details	PSO	BB-BC			GSS
		Standard	Modified	Exponential	
Population size	50	50	50	50	1
Maximum no. iterations	200	200	200	200	200
Maximum no. structural analyses	10000	10000	10000	10000	400
Optimum weight (kg)	5026.4	5025.2	5025.2	5026.4	5026.4
Worst weight	5132.8	5169.5	5030.2	5033.6	5042.5
Mean weight	5040.1	5050.9	5027	5027.9	5037
Standard deviation	32.8	46.6	2.31	2.33	5.7

#### 5.1.4.2 Example 2: 130-Member Transmission Tower

The steel transmission tower depicted in Figure 5.5 is selected as the second design example. The structure is composed of 130 members and 33 joints. To keep the size of design space as large as possible, no member grouping is performed. Hence, a challenging design optimization problem with 130 design variables is investigated. The applied loads on the tower are given in Table 5.2 (see Figure 5.6 as well). In this example, the displacement at the tip of the tower in  $x$ -direction is limited to 3 cm, and the maximum number of iterations for GSS is set to 200. For the PSO, and BB-BC algorithms a population size of 50 individuals and a maximum number of 1000 iterations are used.

Sizing optimization of the transmission tower is carried out, and the results obtained are tabulated in Table 5.3. In this example, the GSS gives the least weight, which is 5801.3 kg. The other design weights are 6059.6 kg by PSO, 6427.8 kg by standard BB-BC, 5973.5 kg by modified BB-BC, and 5853.9 kg by exponential BB-BC algorithm. The GSS locates the minimum weight design over 200 iterations by performing only 400 structural analyses. On the other hand, the above-mentioned heavier design weights are obtained by other algorithms through 50000 analyses, which is simply calculated by multiplying the population size (i.e. 50) by the maximum number of iterations (i.e. 1000). The performance of the GSS can also be observed from Table 5.3 in terms of the worst, mean and standard deviation of the design weights attained using a particular technique. The convergence history showing the variation of the best feasible generated design throughout the optimization process in the best run of GSS is depicted in Figure 5.7. The convergence histories of the best runs of the other algorithms are depicted in Figure 5.8.

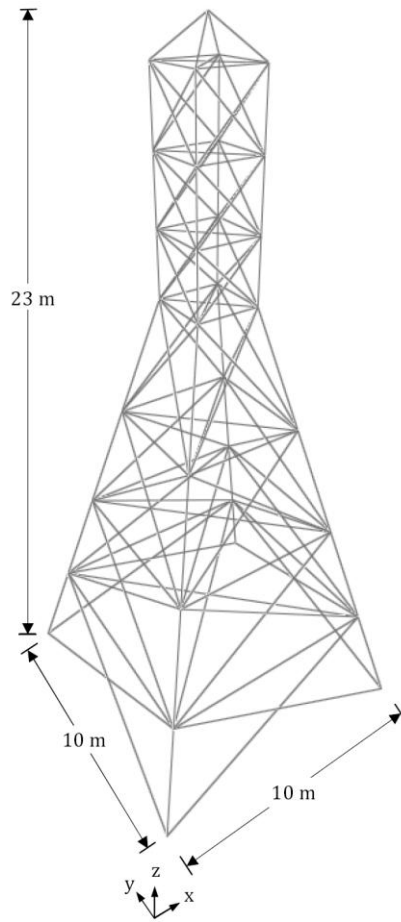


Figure 5.5: 130-member transmission tower

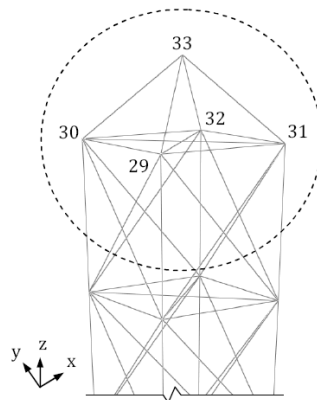


Figure 5.6: Loaded nodes of 130-member transmission tower

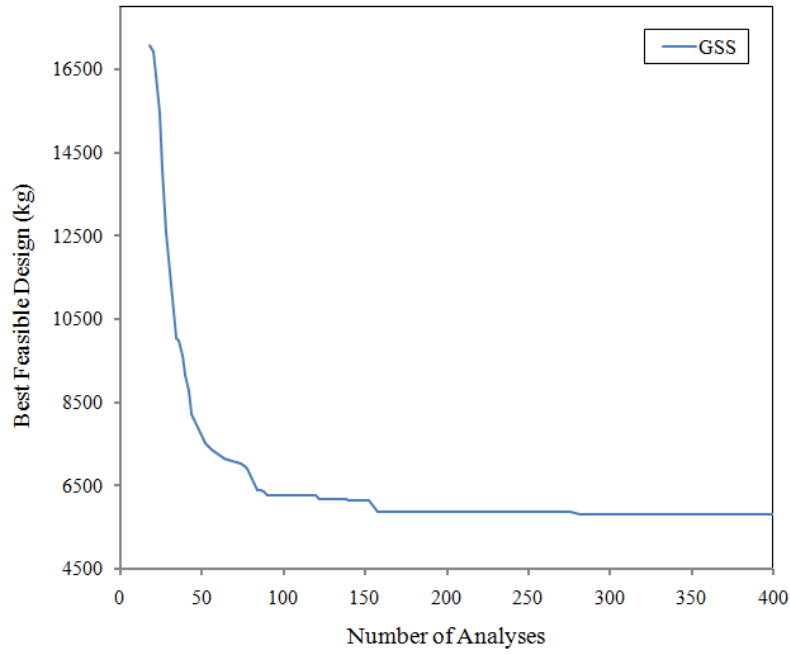


Figure 5.7: Convergence history of the best feasible generated design for 130-member transmission tower using GSS

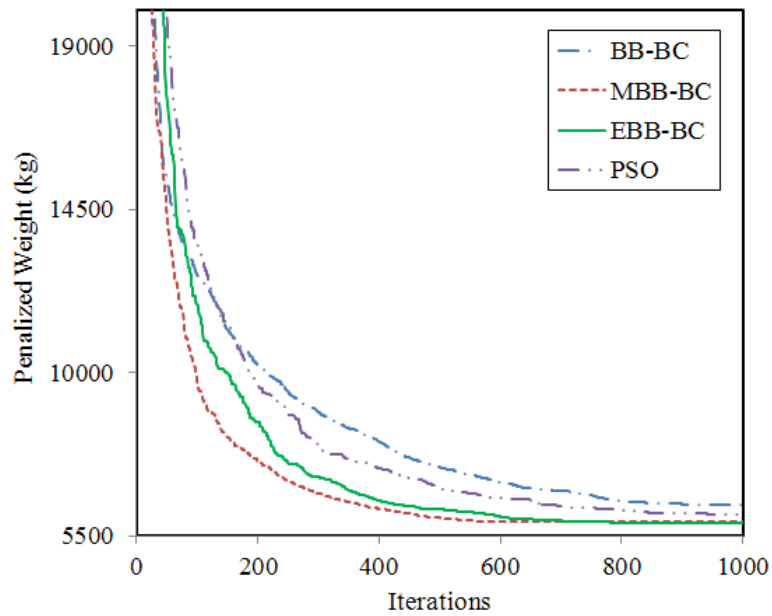


Figure 5.8: Convergence histories for 130-member transmission tower using some metaheuristics

Table 5.2. Loading of 130-member transmission tower

Node	x-direction (kN)	y-direction (kN)	z-direction (kN)
29	100	0	0
30	100	0	0
31	0	25	0
32	0	25	0
33	0	0	-50

Table 5.3. Comparison of optimum designs for 130-member transmission tower

Computational details	PSO	BB-BC			GSS
		Standard	Modified	Exponential	
Population size	50	50	50	50	1
Maximum no. iterations	1000	1000	1000	1000	200
Maximum no. structural analyses	50000	50000	50000	50000	400
Optimum weight (kg)	6059.6	6427.8	5973.5	5853.9	5801.3
Worst weight	6611.4	7172.6	6434.7	6526.4	6118.5
Mean weight	6364.3	6742.9	6144.5	6059.5	6004.4
Standard deviation	227.3	303.8	188.9	240.4	87.6

### 5.1.4.3 Example 3: 392-Member Double Layer Grid

The steel double layer grid shown in Figure 5.9 is chosen as the third design optimization instance. The structure is composed of 392 members and 113 joints. Similar to the first example no member grouping is performed to create a challenging design optimization problem with 392 sizing design variables. For design purpose, downward loads of 15 kN are applied at all nodes of the top grid. In this example, the

vertical displacement at the center of the bottom grid is limited to 1 cm and the maximum number of iterations for GSS is set to 200. For the PSO and BB-BC algorithms a population size of 50 individuals and a maximum number of 2000 iterations are used.

Design optimization of the 392-member double layer grid is performed and the results attained using different techniques are presented in Table 5.4. In this example, the GSS gives the least weight again, which is 2113.8 kg. The other design weights are 3272.7 kg by PSO, 3645.1 kg by standard BB-BC, 2574.4 kg by modified BB-BC, and 2116.3 kg by exponential BB-BC algorithm. The GSS locates the minimum weight design through only 400 structural analyses; while the other algorithms need 100000 analyses to locate the abovementioned heavier design weights. The performance of the GSS in terms of the worst, mean, and standard deviation of attained design weights can be seen from Table 5.4 as well. The convergence history in the best run of GSS is depicted in Figure 5.10. The convergence histories of the best runs of the other algorithms are shown in Figure 5.11.

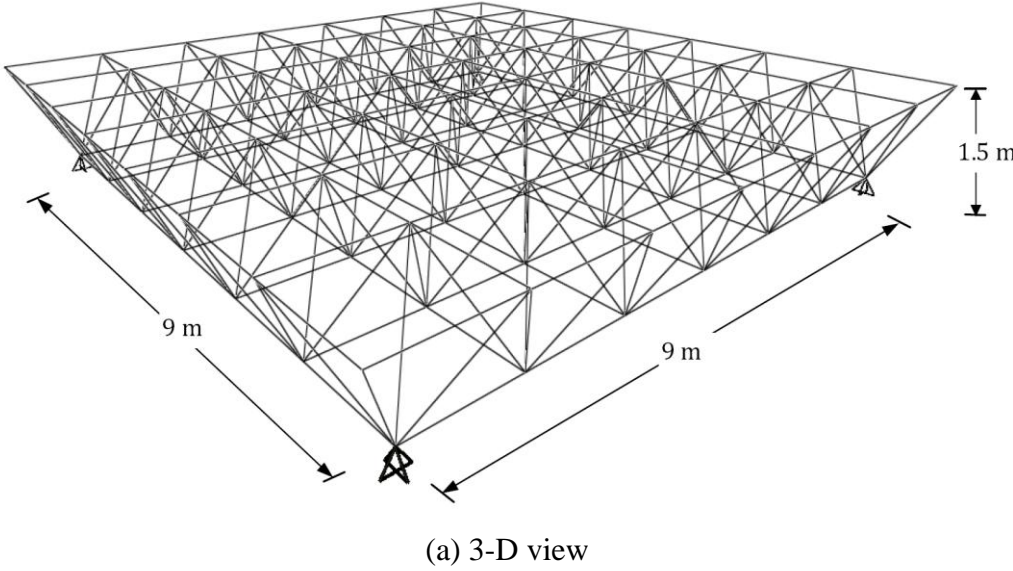
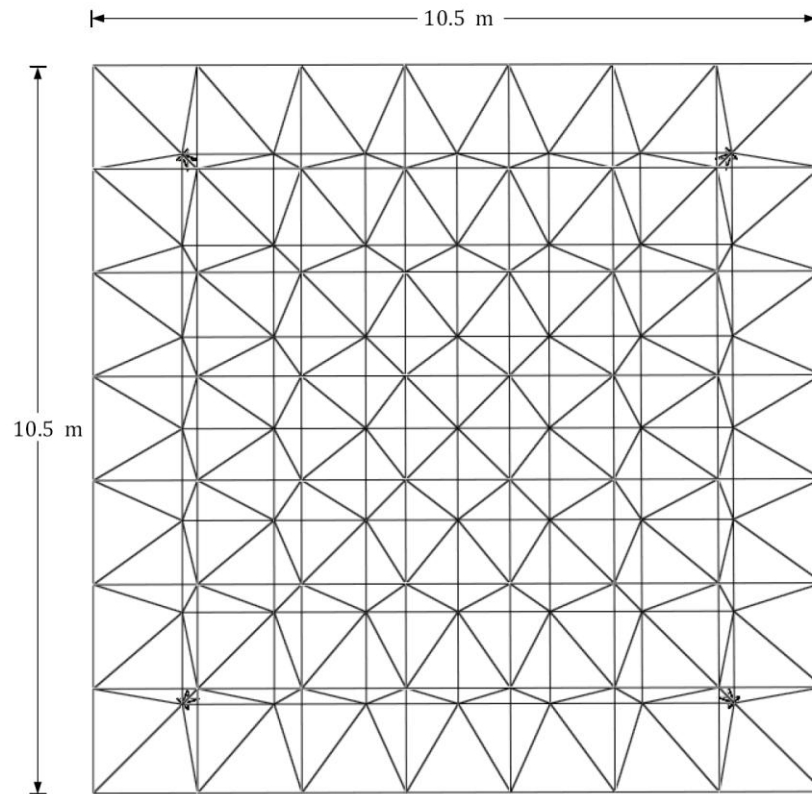


Figure 5.9: 392-member double layer grid (a) 3-D view (b) top view



(b) top view

Figure 5.9 (continued)

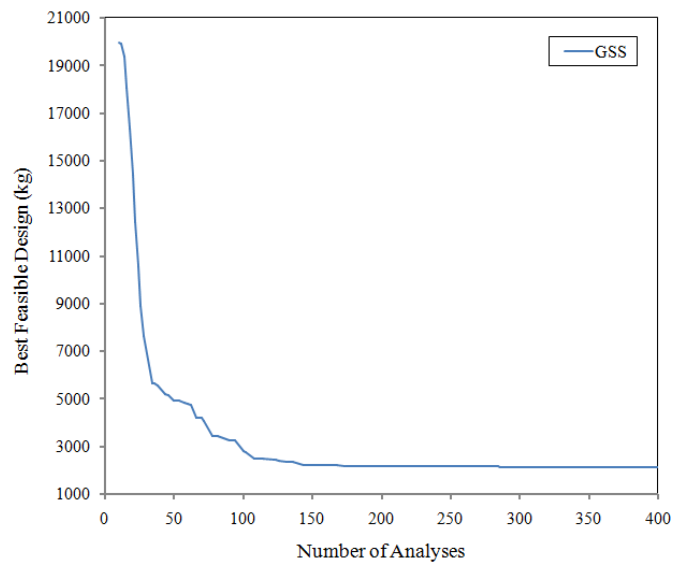


Figure 5.10: Convergence history of the best feasible generated design for 392-member double layer grid using GSS

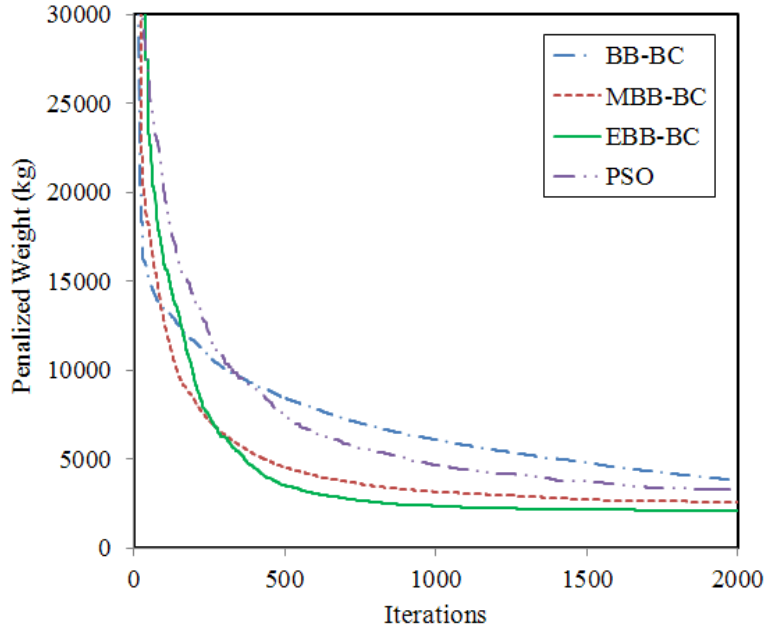


Figure 5.11: Convergence histories for 392-member double layer grid using some metaheuristics

Table 5.4. Comparison of the optimum designs for 392-member double layer grid

Computational details	PSO	BB-BC			GSS
		Standard	Modified	Exponential	
Population size	50	50	50	50	1
Maximum no. iterations	2000	2000	2000	2000	200
Maximum no. structural analyses	100000	100000	100000	100000	400
Optimum weight (kg)	3272.7	3645.1	2574.4	2116.3	2113.8
Worst weight	3603.1	4501.2	2775.9	2290.3	2146.9
Mean weight	3406.3	4013.1	2669.1	2169.6	2127.5
Standard deviation	159.7	376.5	83.3	81.5	9.6



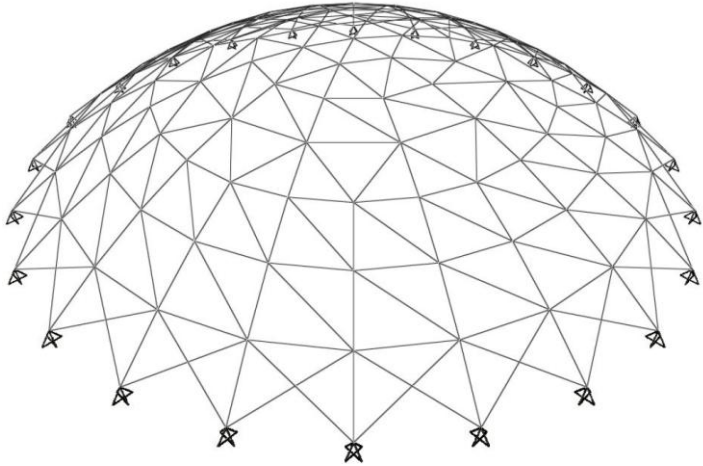
#### 5.1.4.4 Example 4: 354-Member Truss Dome

The steel truss dome shown in Figure 5.12 is adopted as the last design optimization example. The dome is composed of 354 members and 127 joints. Similar to the former examples no member grouping is performed and a challenging design optimization problem including 354 sizing design variables is considered. For design purpose, downward loads of 15 kN are applied at all the unsupported nodes. In addition to these loads, a single downward load of 100 kN is also acting at the tip of the dome. The vertical displacement of the dome tip is limited to 2 cm and the maximum number of iterations for GSS is set to 250. For the PSO and BB-BC algorithms a population size of 50 individuals and a maximum number of 4000 iterations are used.

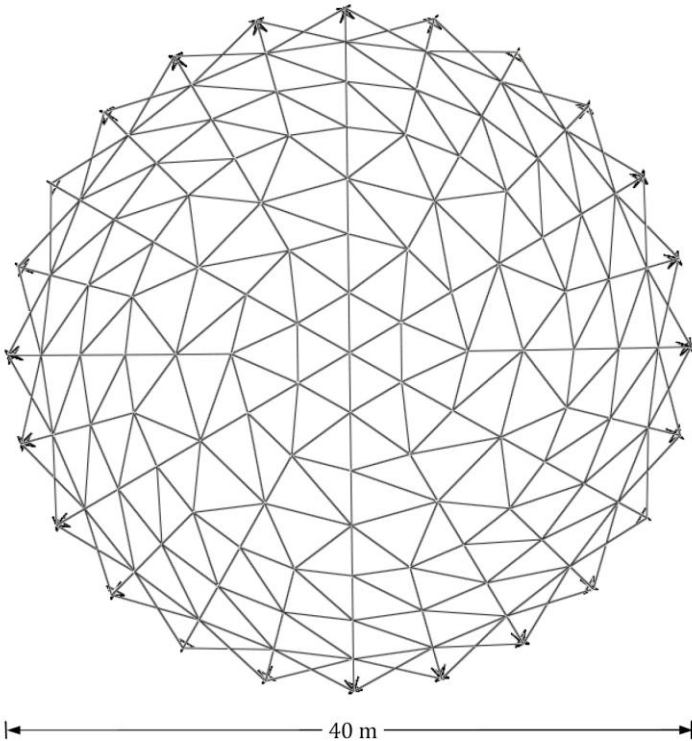
Design optimization of the dome is carried out and the results obtained using different techniques are given in Table 5.5. Once again the GSS yields the least weight for this example, which is 14252.5 kg. The other design weights are 21822.3 kg by PSO, 15786.2 kg by modified BB-BC, and 14312.4 kg by exponential BB-BC algorithm. The GSS locates the minimum weight design through only 500 structural analyses; while the other algorithms need 200000 analyses to locate the above-mentioned heavier design weights.

For this example, no feasible design is located with the standard BB-BC algorithm when the initial population is generated randomly. To facilitate design transitions to feasible regions during the search, the standard BB-BC algorithm is initiated from one feasible design point such that the strongest section of the discrete profile set is assigned to all the truss members in one individual, while all other individuals in the initial population are created randomly in a usual manner. This way it is ensured that the search process is performed in more promising regions of the design space throughout the optimization process. The BB-BC algorithm employed under this case locates a final design weight of 16320.9 kg. The performance of the GSS in terms of the worst, mean, and standard deviation of attained design weights can be seen from

Table 5.5. The convergence history in the best run of GSS is depicted in Figure 5.13. The convergence histories of the best runs of the other algorithms are depicted in Figure 5.14

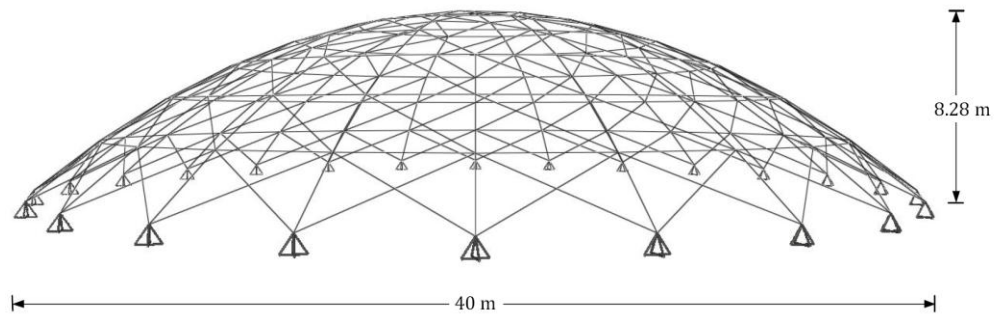


(a) 3-D view



(b) top view

Figure 5.12: 354-member truss dome (a) 3-D view (b) top view (c) side view



(c) side view

Figure 5.12 (continued)

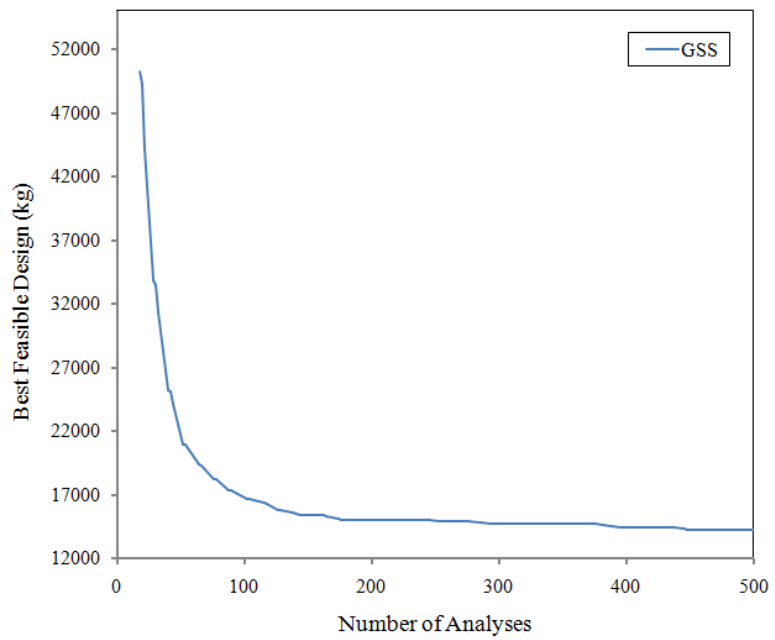


Figure 5.13: Convergence history of the best feasible generated design for 354-member truss dome using GSS

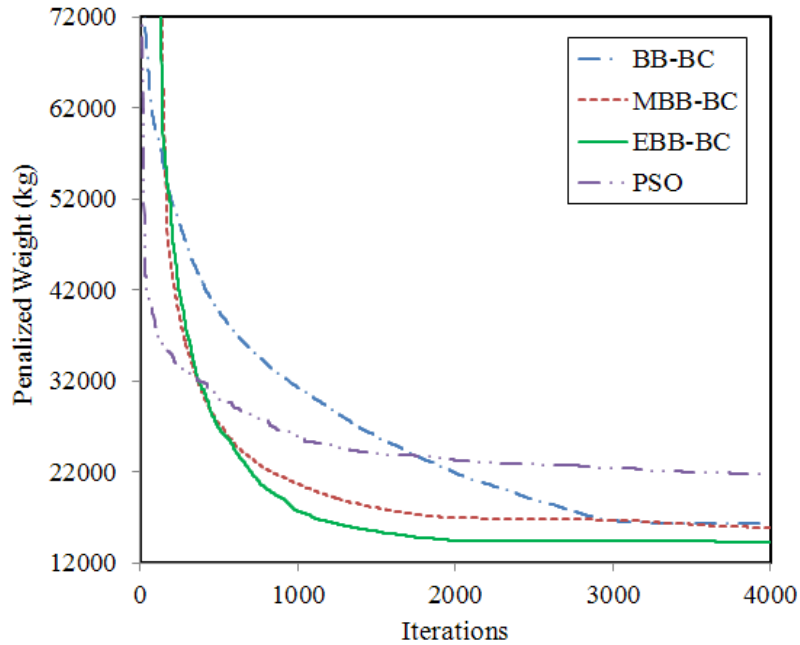


Figure 5.14: Convergence histories for 354-member truss dome using some metaheuristics

Table 5.5. Comparison of the optimum designs for 354-member truss dome

Computational details	PSO	BB-BC			GSS
		Standard	Modified	Exponential	
Population size	50	50	50	50	1
Maximum no. iterations	4000	4000	4000	4000	250
Maximum no. structural analyses	200000	200000	200000	200000	500
Optimum weight (kg)	21822.3	16320.9	15786.2	14312.4	14252.5
Worst weight	24253.7	17442.5	17217.9	14895.8	14711.4
Mean weight	23456.9	16704.1	16344.5	14533.2	14515.4
Standard deviation	972.3	519.1	600.3	236.8	142.6

### 5.1.5 Summary

In this section a novel guided stochastic technique named GSS is developed for discrete sizing optimization of steel truss structures. For design optimization, first those members which should be resized are detected using the well-known principle of virtual work together with the load to capacity ratios of structural members. Next, the resizing of members is carried out in a stochastic way. Basically, two critical groups of truss members (IG, and  $DG_w$ ) are detected such that IG members are increased in size to satisfy strength and/or displacement constraints, and meanwhile  $DG_w$  members are decreased in size to minimize the structural weight in the course of optimization. The performance evaluation of the proposed GSS is carried out based on a comparison with robust representatives of metaheuristic approaches using four real-size truss structures, namely a 117-member cantilever truss, a 130-member transmission tower, a 392-member double layer grid, and a 354-member truss dome. The results obtained in these examples clearly documented the efficiency of the proposed technique in discrete sizing optimization of truss structures. Besides the promising performance of the GSS in design optimization, its performance is not affected by random initiation. Furthermore, no penalty function or gradient information is needed to guide the optimization process.

It should be noted that, in comparison to the general purpose metaheuristic techniques, the computational efficiency of the GSS in discrete sizing optimization of truss structures is basically due to the use of domain knowledge (knowledge obtained using the principle of virtual work as well as the information collected during the structural analysis and design stages) in the course of optimization. The low computational effort required for design optimization using the GSS can pave the way for development of a new class of stochastic design techniques guided by principles of structural mechanics.

## **5.2. GSS for Multiple Load Cases and Displacement Criteria**

### **5.2.1 Introduction**

The GSS developed in the previous section is basically suitable for handling optimal sizing problems of steel truss structures with a single displacement constraint under a single load case. Nevertheless, practical designs of structures usually come up with numerous displacement criteria in several load cases, and it is crucial that an optimization algorithm can address such practical requirements of design problems in reality. The present section aims to investigate the GSS in a more general class of truss sizing optimization problems subject to multiple displacement constraints and load cases. To this end, enhancements of the GSS are proposed in the form of two alternative approaches that enable the technique to deal with multiple displacement/load cases. The first approach implements a methodology in which the most critical displacement direction is considered only when guiding the search process. The second approach, however, takes into account the cumulative effect of all the critical displacement directions in the course of optimization. Advantage of the integrated force method of structural analysis is utilized for further reduction of the computational effort in these approaches. The proposed enhancements of GSS are investigated and compared with some selected techniques of metaheuristics through three real-size trusses that are sized for minimum weight per AISC-LRFD (1994) specifications. The numerical results reveal that both enhancements generally provide better solutions than do metaheuristics using lesser computational effort.

### **5.2.2 Enhancement of GSS**

The original GSS is developed for discrete size optimization of steel truss structures with single displacement constraint under a single load case. However, as stated before, a practical design problem usually incorporates numerous displacement criteria under several load cases. Hence, enhancements of the GSS are carried out in this section to address a more general class of truss optimization problems using the

algorithm. Apparently, the two main inputs used for decision making in the GSS are the LCR and SI values of structural members. When multiple load cases are present, the LCR values are first calculated for all the members under each load case. The maximum LCR value of a member amongst all the load cases is selected as the critical LCR of this member. The critical LCR values of the members determined this way are then used to select the  $IG_s$  and  $DG_w$  members in the optimization process as described in the previous section.

Unlike a single displacement constraint, where computing the SI values of members is rather straightforward, an appropriate method should be adopted to generate a suitable SI value for each member while dealing with multiple displacement criteria. To this end, the following two approaches are proposed and evaluated in the present study.

#### **5.2.2.1 First Approach (GSS<sub>A</sub>)**

In the first approach the problem is turned into as a single displacement constraint problem by focusing solely on the most critical displacement direction even though there are several constraint violations regarding displacements. If there is any displacement constraint violation, the critical displacement direction refers to the one that violates the constraint with the largest value. Otherwise, for a feasible design, it is the one for which the amount of nodal displacement is closest to the corresponding allowable limit. Once the critical displacement direction is determined, calculation of SI values for truss members can be carried out in the same way described in the previous chapter. The computed SI values are then employed for determining the  $IG_d$  and  $DG_w$  members in the course of design optimization as outlined in chapter 5.

#### **5.2.2.2 Second Approach (GSS<sub>B</sub>)**

Alternatively, in the second approach multiple displacement constraints are taken into account on a cumulative basis. In this approach SI values of members are calculated at all the critical directions where displacement violations occur, and they are added to

each other in a cumulative manner as formulated in Eq. (5.7).

$$SI_{i,cu} = \frac{\sum_{k=1}^{Ncr} DPF_{i,k}}{V_i} \quad (5.7)$$

In Eq. (5.7),  $SI_{i,cu}$  and  $V_i$  are the cumulative sensitivity index, and volume of the  $i$ -th member, respectively; and  $Ncr$  is the total number of critical directions (i.e., displacement directions with constraint violation) used for calculation of the  $DPF$  values. It is worth mentioning that for a generated design with no constraint violation, similarly the critical direction is taken as the one for which the amount of nodal displacement is closest to the corresponding allowable limit. Once the SI values for truss members are computed using Eq. (5.7), these values can be used for determining the  $IG_d$  and  $DG_w$  members.

### 5.2.3 Integrated Force Method

As mentioned before, the GSS requires that structural response of a generated candidate design is performed under both real and virtual loads. Because the  $GSS_A$  focuses on most critical displacement direction only, two response computations are required at each iteration of this algorithm, if a displacement based structural analysis is used. On the other hand, multiple constraint violations are simultaneously accounted for in the  $GSS_B$  and thus numerous response computations maybe needed at an arbitrary iteration of the algorithm to calculate internal forces under virtual loads for each critical direction. To this end, the use of integrated force method (IFM) (Patnaik et al. 1991) of structural analysis is adopted here for further reduction of computational effort in both approaches.

In the commonly used displacement method of structural analyses the force-displacement relationship for the structure is given as follows,



$$[K]\{U\} = \{P\} \quad (5.8)$$

where  $[K]$  is the stiffness matrix of the structure, and  $\{U\}$  and  $\{P\}$  are the nodal displacement and applied load vectors, respectively. Once nodal displacements are determined the internal forces are then calculated based on the obtained displacements. Alternatively, the IFM employs the relation below for a structure with  $m$  members and  $n$  displacement degrees of freedom.

$$[S]\{F\} = \{P^*\} \quad (5.9)$$

where  $[S]$  is the IFM governing unsymmetrical square matrix of dimension  $m$ ,  $\{F\}$  is the vector of internal forces and  $\{P^*\}$  is defined as,

$$\{P^*\} = \begin{Bmatrix} \{P\} \\ \{0\} \end{Bmatrix} \quad (5.10)$$

where  $\{P\}$  is the applied load vector and  $\{0\}$  denotes a zero vector of dimension  $r$ , which is equal to the degree of indeterminacy of a structure. Using the IFM formulation, once internal forces  $\{F\}$  are determined, the nodal displacements can then be calculated using the computed forces.

Accordingly, at each iteration of the GSS, based on Eq. (5.9) the internal forces  $\{F\}$  are first computed under real loads using the following relation:

$$\{F\} = [S]^{-1}\{P^{r*}\} \quad (5.11)$$

where  $P^r$  denotes the applied real loading on the structure. It is important to note that each column in matrix  $[S]^{-1}$  contains internal member forces caused by a unit load in the direction of a particular degree of freedom (DOF). By definition, this is identical

to virtual internal forces in members developed under an applied virtual load. Hence, when the structure is analyzed under the applied real loads at each iteration, the virtual internal forces can directly be extracted from  $[S]^{-1}$  matrix without a need for additional analyses. It is worth mentioning that in this approach if a virtual load is applied in a negative direction with respect to the corresponding DOF, the attained virtual internal forces using  $[S^{-1}]$  should be multiplied by -1 to account for coordinates sign convection. It should be noted that a comparison of displacement method versus IFM in terms of computational efficiency is out of the scope of this study. The IFM is employed here since it removes the need for additional response computations in case of virtual loading.

#### **5.2.4 Numerical Examples**

This section carries out performance evaluation of the GSS in discrete sizing optimization of truss structures subject to multiple displacement constraints and load cases. In the investigated examples the notations  $GSS_A$ , and  $GSS_B$  are used to refer to the first and second approaches proposed for enhancements of the GSS, respectively. The investigated examples consist of a 117-member cantilever truss, a 130-member transmission tower, and a 368-member truss dome, which are challenging real-size optimization problems with 117, 130, and 368 sizing design variables, respectively. The optimum designs obtained for these examples using the  $GSS_A$  and  $GSS_B$  are compared to those located using some metaheuristics. Due to the stochastic nature of the techniques, each problem is independently solved ten times with each technique and the numerical results collected are used for comparisons.

Here, the value of parameter  $\Delta R$  is set to 0.1 and the maximum number of iterations is taken as the termination criterion of the optimization process. For all the investigated examples, the sizing variables are selected from a database of 37 pipe sections, and the material properties of steel are taken as follows: modulus of elasticity ( $E$ ) = 200 GPa, yield stress ( $F_y$ ) = 248.2 MPa, and unit weight of the steel ( $\rho$ ) = 7.85 ton/m<sup>3</sup>.

#### 5.2.4.1 Example 1: 117-Member Cantilever Truss

The steel cantilever truss shown in Figure 5.2 is studied in the previous section with a single displacement constraint under a single load case. For design purpose the structural members were linked into 25 member groups in the previous section. However, in order to keep the size of design space as large as possible, no member grouping is performed here so that a challenging design optimization problem with 117 design variables is generated. The structure is sized under three independent load cases, where the loads are applied at all unsupported nodes of the truss in the following cases: (i) horizontal loads of 15 kN applied in positive x-direction, (ii) horizontal loads of 15 kN applied in positive y-direction, (iii) vertical loads of 15 kN applied in negative z-direction. The displacements of all nodes in x, y, and z directions are limited to a maximum value of 4 cm.

The maximum number of iterations for  $GSS_A$  and  $GSS_B$  is set to 200, whereas a population size of 50 individuals and a maximum number of 500 iterations are used for PSO and BB-BC algorithms. The results obtained for sizing optimization of the 117-member cantilever truss are tabulated in Table 5.6. As seen from this table, the least design weight of the cantilever truss is attained by  $GSS_B$ , which is 3072.2 kg. The second best design is located by  $GSS_A$ , which is 3100.9 kg. The other design weights are 3123.6 kg by exponential BB-BC algorithm, 3125.4 kg by modified BB-BC, 3476 kg by PSO, and 3586.5 kg by standard BB-BC. The  $GSS_B$  locates the minimum weight design through only 317 structural analyses, of which 200 analyses are performed at the iterations of the GSS, while 117 analyses are performed during a local search around the best feasible design following the last iteration. On the other hand, the abovementioned heavier design weights are obtained by metaheuristic algorithms through 25000 analyses, calculated by multiplying the population size (i.e. 50) with the maximum number of iterations (i.e. 500). The promising performance of the  $GSS_B$  can also be observed from Table 6.1 in terms of the worst, mean and standard deviation of the attained design weights. The convergence histories showing the variation of the feasible generated design throughout the optimization process in

the best runs of the  $GSS_A$  and  $GSS_B$  are depicted in Figure 5.15. The convergence histories of the best runs of the metaheuristic algorithms are depicted in Figure 5.16.

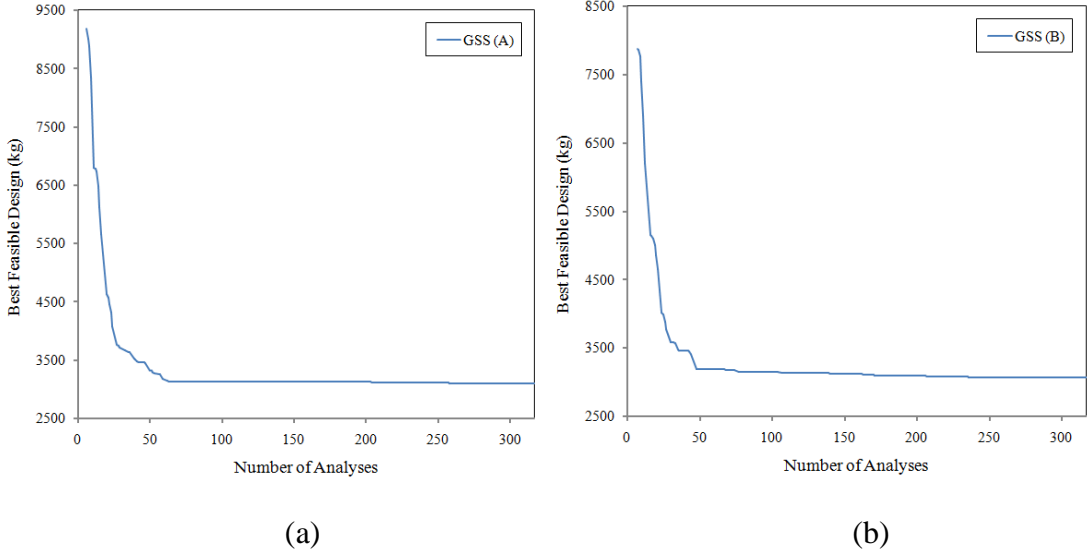


Figure 5.15: Convergence history of the best feasible generated design for 117-member cantilever truss using (a)  $GSS_A$ , (b)  $GSS_B$

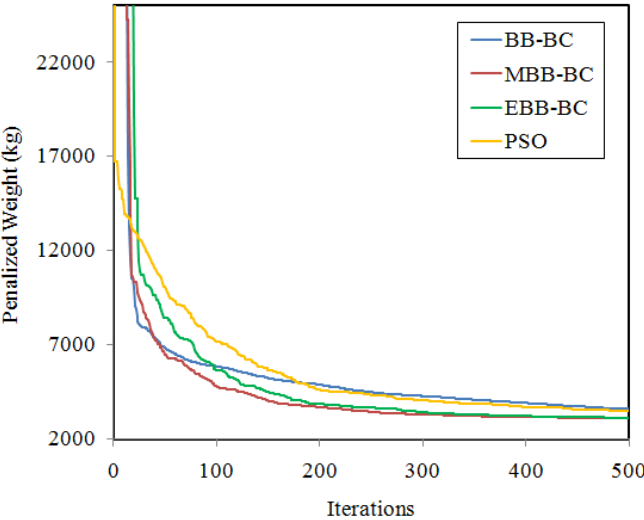


Figure 5.16: Convergence histories for 117-member transmission tower using some metaheuristics

Table 5.6. Comparison of optimum designs for 117-member cantilever truss

Computational details	PSO	BB-BC			GSS <sub>A</sub>	GSS <sub>B</sub>
		Standard	Modified	Exponential		
Population size	50	50	50	50	1	1
Maximum no. iterations	500	500	500	500	200	200
No. analyzed designs	25000	25000	25000	25000	317	317
Optimum weight (kg)	3476	3586.5	3125.4	3123.6	3100.9	3072.2
Worst weight	3828.2	4265.5	3253.5	3277.3	3133.5	3085.9
Mean weight	3600.7	3855.9	3205.8	3209.9	3112.3	3078.3
Standard deviation	141.1	249.8	60.7	66.8	12.6	6.6
Optimum weight rank	5	6	4	3	2	1

#### 5.2.4.2 Example 2: 130-Member Transmission Tower

The steel transmission tower shown in Figure 5.5 is studied in the previous section under a single load case and a single displacement constraint. Here, the truss tower is sized under three independent load cases given in Table 5.7 (see Figure 5.6 as well). Further, the displacements of all nodes in x, y, and z directions are limited to a maximum value of 3 cm.

The maximum number of iterations for GSS<sub>A</sub> and GSS<sub>B</sub> is set to 200, whereas a population size of 50 individuals and a maximum number of 1000 iterations are used for PSO and BB-BC algorithms. The results obtained for sizing optimization of the transmission tower are tabulated in Table 5.8. In this example the least weight of the truss tower, which is 6448.1 kg, is again produced by GSS<sub>B</sub>. The second best design is located by GSS<sub>A</sub>, which is 6485.6 kg. The other design weights are 7060.7 kg by

exponential BB-BC algorithm, 7011.1 kg by modified BB-BC, 7344.9 kg by PSO, and 8686.1 kg by standard BB-BC. The  $GSS_B$  locates the minimum weight design by performing only 330 structural analyses. On the other hand, the above-mentioned heavier design weights are obtained by metaheuristic algorithms through 50000 analyses. The promising performance of the  $GSS_B$  can also be observed from Table 5.8 in terms of the worst, mean and standard deviation of the attained design weights. The convergence histories showing the variation of the feasible generated design throughout the optimization process in the best runs of the  $GSS_A$  and  $GSS_B$  are depicted in Figure 5.17. The convergence histories of the best runs of the metaheuristic algorithms are depicted in Figure 5.18.

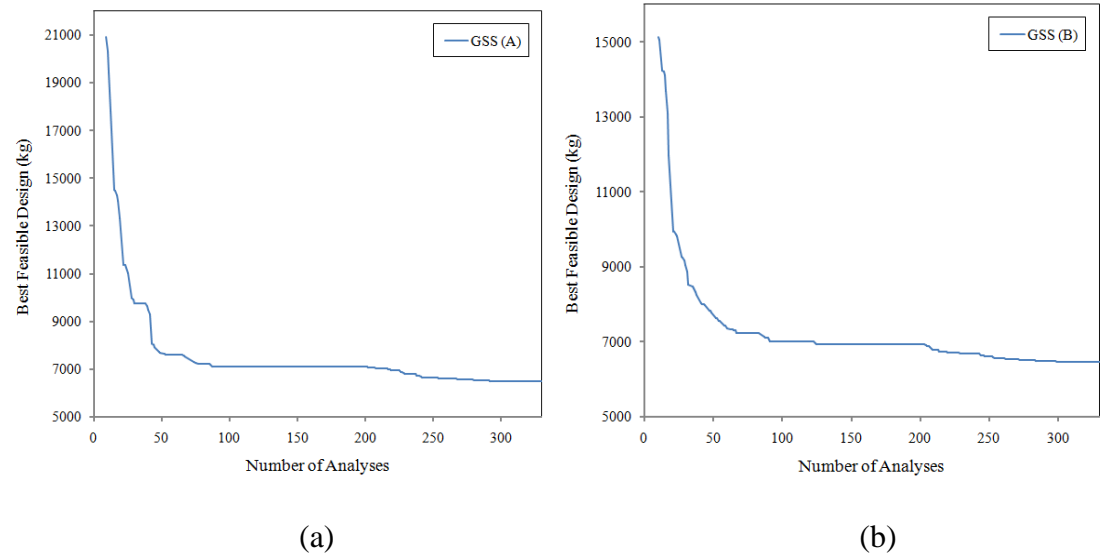


Figure 5.17: Convergence history of the best feasible generated design for 130-member transmission tower using (a)  $GSS_A$ , (b)  $GSS_B$

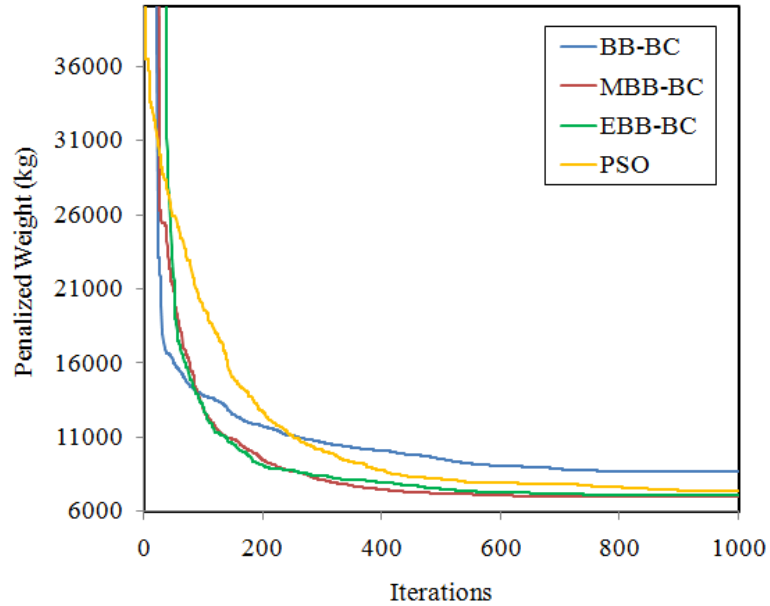


Figure 5.18: Convergence histories for 130-member transmission tower using some metaheuristics

Table 5.7. Loading of 130-member transmission tower

Node	x-direction (kN)			y-direction (kN)			z-direction (kN)		
	Load Case			Load Case			Load Case		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
29	100	0	0	0	100	85	0	0	0
30	100	0	0	0	0	0	0	0	0
31	0	0	0	25	100	0	0	0	0
32	0	0	0	25	0	0	0	0	0
33	0	0	50	0	0	100	-50	-100	0

Table 5.8. Comparison of optimum designs for 130-member transmission tower

Computational details	PSO	BB-BC			GSS <sub>A</sub>	GSS <sub>B</sub>
		Standard	Modified	Exponential		
Population size	50	50	50	50	1	1
Maximum no. iterations	1000	1000	1000	1000	200	200
No. analyzed designs	50000	50000	50000	50000	330	330
Optimum weight (kg)	7344.9	8686.1	7011.1	7060.7	6485.6	6448.1
Worst weight	8869	9492.4	7808.2	7706.6	6706.2	6557.6
Mean weight	7749.4	9127.1	7280.2	7307.2	6573.6	6509.9
Standard deviation	633.3	377.4	368.1	247.8	83.7	44.9
Optimum weight rank	5	6	3	4	2	1

### 5.2.4.3 Example 3: 368-Member Truss Dome

The steel truss dome shown in Figure 5.19 is composed of 368 members and 129 joints. Similar to the former examples, no member grouping is performed to generate a challenging design optimization problem with 368 sizing design variables. The dome is sized under three independent load cases, where the loads are applied at all unsupported nodes of the truss in the following cases: (i) horizontal loads of 15 kN applied in positive x-direction, (ii) horizontal loads of 15 kN applied in positive y-direction, (iii) vertical loads of 15 kN applied in negative z-direction. The displacements of all nodes in x, y, and z directions are limited to a maximum value of 1.5 cm.

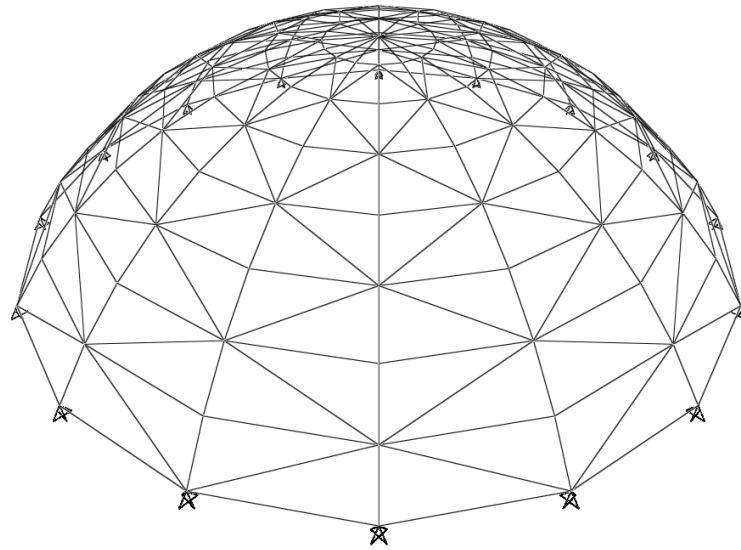
The maximum number of iterations for GSS is set to 400, whereas a population size of 50 individuals and a maximum number of 2000 iterations are used for PSO and BB-BC algorithms. The results obtained for sizing optimization of the dome are given in



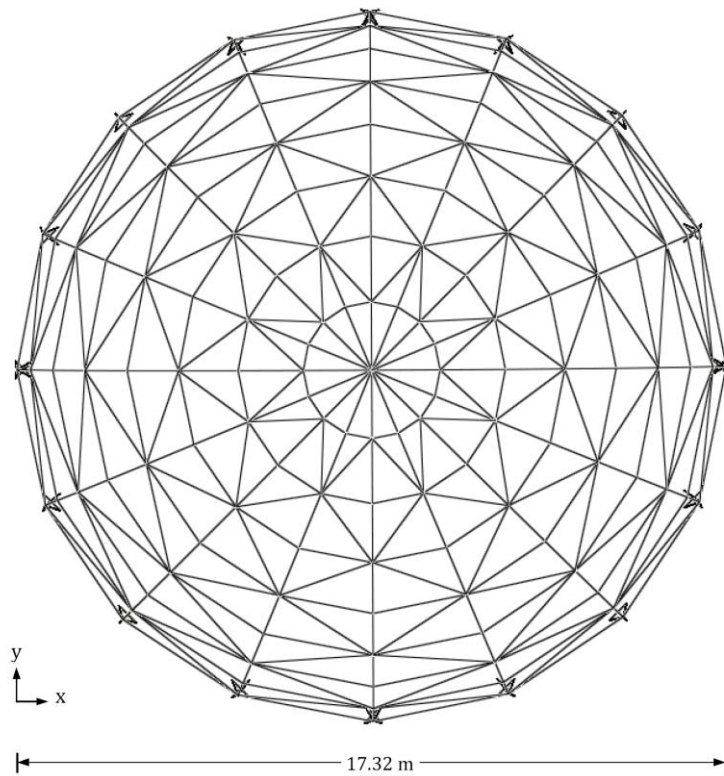
Table 5.9. Once again the least weight of the dome is attained by  $GSS_B$ , which is 4850.1 kg. The  $GSS_A$  gets to find a design weight of 4917.1 kg for the dome, which is comparatively heavier than the design weight of 4856.2 kg located by the exponential BB-BC algorithm. This reveals that sometimes modern metaheuristics can provide solutions better than those of  $GSS_A$  at the expense of enormously increased computation time. The other design weights are 5462.2 kg by modified BB-BC, and 8998.3 kg by PSO. The  $GSS_B$  locates the minimum weight design by performing only 768 structural analyses. However, the above-mentioned design weights are obtained by metaheuristic algorithms through 100000 structural analyses.

For this example, no feasible design is located with the standard BB-BC algorithm when the initial population is generated randomly. To facilitate design transitions to feasible regions during the search, the standard BB-BC algorithm is initiated from one feasible design point such that the strongest section of the discrete profile set is assigned to all the truss members in one individual, while all other individuals in the initial population are created randomly in a usual manner. This way it is ensured that the search process is performed in more promising regions of the design space throughout the optimization process.

The BB-BC algorithm employed under the above-mentioned case locates a final design weight of 8186.7 kg. The convergence histories showing the variation of the feasible generated design throughout the optimization process in the best runs of the  $GSS_A$  and  $GSS_B$  are presented in Figure 5.20. The convergence histories of the best runs of the metaheuristic algorithms are shown in Figure 5.21.

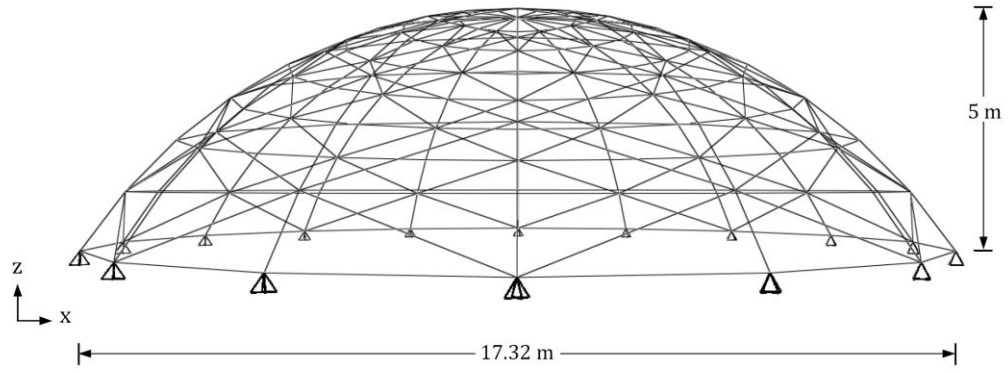


(a) 3-D view



(b) top view

Figure 5.19: 368-member truss dome (a) 3-D view (b) top view (c) side view



(c) side view

Figure 5.19 (continued)

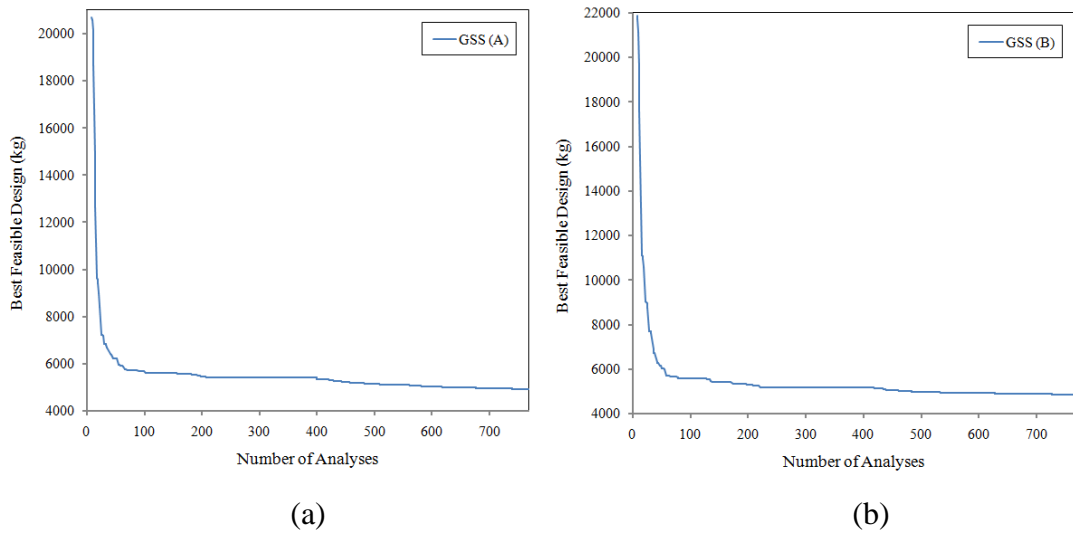


Figure 5.20: Convergence history of the best feasible generated design for 368-member truss dome using (a)  $GSS_A$ , (b)  $GSS_B$

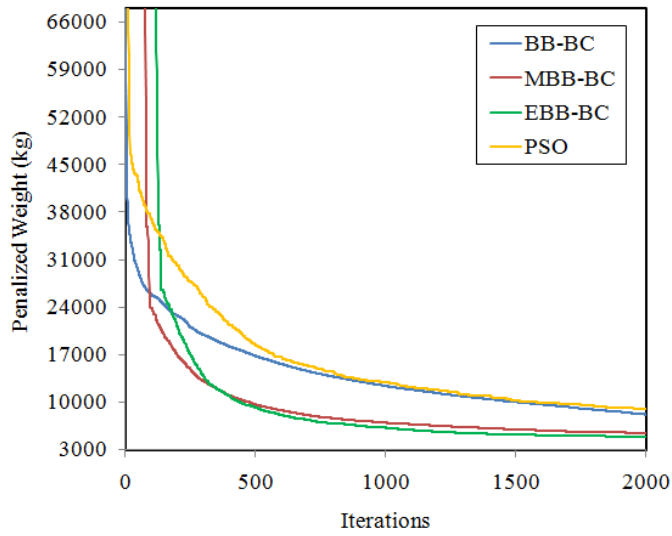


Figure 5.21: Convergence histories for 368-member truss dome using some metaheuristics

Table 5.9. Comparison of optimum designs for 368-member truss dome

Computational details	PSO	BB-BC			GSS <sub>A</sub>	GSS <sub>B</sub>
		Standard	Modified	Exponential		
Population size	50	50	50	50	1	1
Maximum no. iterations	2000	2000	2000	2000	400	400
No. analyzed designs	100000	100000	100000	100000	768	768
Optimum weight (kg)	8998.3	8186.7	5462.2	4856.2	4917.1	4850.1
Worst weight	9961.4	8995.3	6113.7	4968.3	5043.5	5021.6
Mean weight	9514.9	8604.4	5785.2	4924.5	4967.6	4918.7
Standard deviation	397.4	404.9	305.7	42.7	53.5	69.1
Optimum weight rank	6	5	4	2	3	1

### 5.2.5 Summary

The GSS is a new design optimization algorithm for sizing optimization of steel trusses with a single displacement constraint under a single load case. The attributes of the GSS, which in fact makes it very attractive for practical applications, are that it can be applied to problems with discrete size variables, plus it produces comparable and often better solutions to challenging design optimization problems using lesser computational effort compared to today's popular global optimization techniques referred to as metaheuristics.

This section investigated the application and performance of the GSS in a more general class of truss sizing optimization problems subject to multiple displacement constraints and multiple load cases. Here, enhancements of the GSS are achieved in the form of two alternative approaches referred to as  $GSS_A$  and  $GSS_B$ . The  $GSS_A$  turns a multi displacement constraint problem into as a single one by focusing solely on the most critical displacement direction at an iteration. On the other hand, all the critical directions where displacement constraint violations occur are simultaneously accounted for by  $GSS_B$  using a formulation on the basis of cumulative effect. Both approaches are implemented in conjunction with the integrated force method (IFM) of structural analysis to further accelerate the computational efficiency of the algorithms. In fact, the use of IFM in  $GSS_A$  is useful but not mandatory because it gets to reduce the number of structural optional to half by eliminating the response analysis due to the virtual load at the most critical direction. On the other hand, the use of IFM in  $GSS_B$  is of much significance as far as computational efficiency of the technique is concerned, because response computations may be excessive due to a need for calculating internal forces under virtual loads for every critical direction where a displacement constraint violation occurs.

The efficiency of the proposed enhancements of the GSS is numerically experimented and quantified using three real-size trusses that are sized for minimum weight according to AISC-LRFD (1994) specifications. The solutions produced to these

problems using  $GSS_A$  and  $GSS_B$  are compared to those obtained using some selected methods of metaheuristics to achieve some level of comparability for the numerical accuracy and computational efficiency of the GSS with metaheuristics. The numerical results reveal that both enhancements generally provide better solutions than do metaheuristics using significantly lesser computational effort. The  $GSS_B$  produces slightly improved solutions compared to the  $GSS_A$ , although the difference may not be of much consequence. On the other hand, the latter comes up with an advantage of also being employed with a displacement based structural analysis at no significant increase in computational cost, unlike  $GSS_B$ .

## CHAPTER 6

### GUIDED STOCHASTIC SEARCH TECHNIQUE FOR DISCRETE SIZING OPTIMIZATION OF STEEL FRAMES

#### 6.1 Introduction

The GSS technique is proposed for computationally efficient optimum design of steel trusses in chapter 5. Regarding the promising performance of the GSS in sizing optimization of truss structures, in the present chapter it is extended and reformulated for discrete sizing optimization of steel frames subject to design provisions of AISC-LRFD (1994). As described in the previous chapter, the GSS offers a stochastic procedure where the optimization process is guided by the principle of virtual work as well as response computations of the generated designs, resulting in an efficient and rapid search in the design space. In this context, the information provided through the structural analysis and design check stages are utilized for handling strength constraints. On the other side, the well-known principle of virtual work is employed to detect the most effective structural members for satisfying displacement criteria. The performance of the proposed technique is numerically evaluated through optimum design of three real-size steel frame structures with 135, 3860, and 11540 structural members. The numerical results reveal the success of the GSS in locating promising solutions for this kind of problems through a reasonable computational effort.

#### 6.2 Sensitivity Index of Frame Members

An efficient design optimization of steel frames entails that members are sized in a candidate design in an effort to improve strength and displacement criteria of the current design. To achieve this, one needs to figure out how the selection of members

will affect the strength and displacement criteria of the resulting structure. In the case of strength criteria this information is readily available through the load to capacity ratios of members given in Eqs. (2.15-2.17), which provide direct information regarding whether the members are oversized or undersized. However, in the case of displacement criteria DPF of members is employed to identify the contribution of each structural member to the total displacement for each considered direction. In the present chapter a procedure based on the principal of virtual work is employed to determine the DPF of each member in a structure (Charney 1991). In order to compute the DPF of a frame member in the  $k$ -th direction, in addition to performing a usual structural analysis under the applied real loads, the structure should be first analyzed under a unit load (virtual load) applied in the  $k$ -th direction. Then, the DPF of the  $i$ -th member in the  $k$ -th direction,  $DPF_{i,k}$ , can be computed as follows.

$$DPF_{i,k} = \int_0^{L_i} \left( \frac{P_i^r P_i^u}{E_i A_i} + \frac{M_i^r M_i^u}{E_i I_i} + \frac{V_i^r V_i^u}{G_i A_{w,i}} + \frac{T_i^r T_i^u}{G_i J_i} \right) dx \quad (6.1)$$

In Eq. (6.1),  $P_i^r$ ,  $M_i^r$ ,  $V_i^r$ , and  $T_i^r$  are the axial, flexural, shear and torsional forces in the  $i$ -th member under the real loads, respectively, and the notational counterparts of these quantities with superscript u refer to the corresponding member forces under the unit load.  $E_i$ ,  $I_i$ ,  $G_i$ ,  $J_i$ ,  $A_i$ ,  $A_{w,i}$  and  $L_i$  are the modulus of elasticity, moment of inertia, shear modulus, polar moment of inertia, cross-sectional area, shear area and length of the  $i$ -th frame member, respectively. A problem associated with the formulation of DPF in Eq. (6.1) is that it is greatly influenced by the current selection of members at a time. As mentioned before, a more general approach to DPF has been introduced by the SI (Charney 1993), Eq. (5.3), which normalizes DPF by the volume of each structural member to provide a more rational measure for identifying effectiveness of members in satisfying the displacement constraint. Once the SI values for frame members are computed, these values can be used to determine the resizing scheme as described in the following section.



### 6.3 The GSS for Discrete Sizing Optimization of Steel Frames

The GSS works on the basis of guiding the optimization process for candidate designs using the information collected on strength and displacement criteria of the current design, i.e, DPF values and load to capacity ratios. To this end, first the critical elements that have the highest impact on the response of a given structure should be detected; and next their sizes should be changed appropriately. The whole process should be guided such that the final minimum weight design leads to an optimum or reasonable near-optimum solution, satisfying all the predefined design constraints. The following steps outline the main procedure in the implementation of the GSS for frame optimization.

**Step 1. Initialization:** The optimization process with GSS initiates with a randomly generated design. This type of initialization, which is similar to initialization of metaheuristic techniques with a population size of one, indicates independency of the algorithm on the starting point of design optimization process. It is worth mentioning that, since the members are to be selected from a predefined list of sections, while generating a new candidate design, the design variables created outside the predefined ranges are moved back to their lower/upper limits.

**Step 2. Evaluation under real loading:** The generated design is evaluated next under the applied real loads, where structural analysis of the design is carried out with the set of steel sections adopted for the design variables, and the force and deformation responses are obtained. Once the values of forces and displacements are known, the structure undergoes a design check where the amounts of strength and serviceability violations are calculated through Eqs. (2.15-2.21). Evaluation stage reveals the quality of the generated design at each iteration.

**Step 3. Evaluation under virtual loading:** In addition to the abovementioned evaluation of the design under the real loads, an additional analysis is carried out at this step to capture structural response of the design under the virtual loading. This

step is required in order to compute the SI values using Eq. (5.3). It is worth mentioning that in case of a single displacement constraint, computing the SI values of members is straightforward since there is a single direction of interest. However, while dealing with multiple displacement criteria, an appropriate method can be selected depending on whether all or only the critical displacement are to be considered in guiding the search process. The critical displacement direction refers to the one that violates Eqs. (2.18) and (2.19) with the largest value, in case a displacement constraint violation takes place. Otherwise, it is the one for which the amount of displacement is closest to the corresponding allowable limit. Once the critical displacement direction is determined, calculation of SI values for the members can be carried out for that particular displacement direction only using Eq. (5.3). In chapter 5 both methods are examined in detail and it is shown that the critical displacement approach works almost as well as the other without a need for an increase in computational effort with a displacement based finite element analysis. Hence this approach is adopted here for handling multiple displacement criteria.

**Step 4. Detection of critical members:** At each iteration of GSS, two main groups of member should be identified for simultaneous member resizing. The first group includes members which are to be increased in size for eliminating strength and/or displacement violations. Oppositely, the second group consists of members which are to be decreased in size in line with the weight minimization objective of the optimization process. The selection of members for these two groups is carried out as follows.

**Step 4.1. Increase-group for constraint satisfaction:** Members that are to be increased in size in an effort to eliminate constraint violations are included in the first group referred to as increase-group (IG). This group can further be divided into two subgroups as  $IG_s$  and  $IG_d$  depending on whether an increase in member size is required due to strength criteria ( $IG_s$ ) or displacement criteria ( $IG_d$ ). All the members violating the strength constraints are directly included in the  $IG_s$  subgroup. These members can simply be detected based on their load to capacity ratios ( $LCRs$ ), which

exceeds 1.0 for under designed members in the light of Eqs. (2.15-2.17).

On the other hand,  $IG_d$  members are selected on the basis of SI values described through Eq. (5.3). In principle, members with highest SI values are identified as the most critical members for reducing or eliminating a displacement violation. The number of  $IG_d$  members is determined using an adaptive ratio parameter  $R_d$  (see step 6) employed as a percentage of the total number of design variables.

**Step 4.2. Decrease-group for weight reduction:** Members that are to be decreased in size with the purpose of achieving a certain weight reduction are included in the second group referred to as decrease-group (DG). This group can further be divided into two subgroups as  $DG_w$  and  $DG_r$  depending on whether the selection is carried out amongst most oversized members ( $DG_w$ ) or others ( $DG_r$ ). The  $DG_w$  members are selected such that the most oversized members are first identified using LCR values. Next, these members are further evaluated based on their SI values, and those that have the least impact on the displacement criteria are selected as  $DG_w$  members. It follows that the  $DG_w$  group consists of oversized members for which the size reduction will have the least effect on displacement of a structure in a direction of interest. Similar to  $IG_d$  members, the number of  $DG_w$  members is also determined in conjunction with an adaptive ratio parameter  $R_w$  (see step 5) based on the total number of design variables. By reducing  $R_w$  (and thus decreasing the number of  $DG_w$  members), it is expected to have no or minimum violation in the displacement constraints while reducing the total weight of the frame structure.

During the numerical experiments it is observed that if only  $DG_w$  members are used for weight reduction, some members rarely (if not at all) find a chance for size reduction during the iterations of the GSS, even though they might have LCR values well below 1.0. These members do not violate strength criteria ( $LCR < 1.0$ ), yet they are not amongst the most oversized members having the least effect on the displacement constraint; nor are amongst the most critical members for handling the displacement criteria. It follows that these members do not belong to any of the

aforementioned  $IG_s$ ,  $IG_d$  or  $DG_w$  groups, and remain unchanged while generating a new candidate solution. In this regard a selected number of such members, named as  $DG_r$  members, are also allowed to reduce their sizes in conjunction with an adaptive ratio of  $R_r$ . It should be emphasized that only members that have LCR values below 0.95 are involved in  $DG_r$  group because size reduction of members having LCR values between 0.95 and 1.0 is more likely to yield an infeasible solution.

**Step 5. Stochastic member resizing:** A resizing strategy is used in the GSS based on a simple stochastic approach, where the  $IG_s$  and  $IG_d$  members are stochastically increased in size using a maximum incremental step size, which is initially set to  $\sqrt{N_s}$  for most practical problems, where  $N_s$  is the number of discrete sections used to size the frame members. Assuming that the members are selected from a wide-flange (W) profile list consisting of 268 discrete sections, the maximum incremental step size is set to 16 in the first iteration, and it is linearly decreased to 3 during the optimization process. For each IG member a random move towards larger sections is then performed using Eq. (5.5) where  $Rand^{IG}$  in this equation is an integer random number selected between 1 and the maximum incremental step size (i.e. 16 to 3) according to a uniform distribution. It follows that in the new design each IG member is replaced stochastically by any of the stronger sections next to its previous value in a profile list. On the other hand, a size increase for a member can be carried out according to different cross-sectional properties, such as cross sectional area,  $A_g$ , moments of inertia about principal axes,  $I_x$  and  $I_y$ , or the governing radius of gyration about the axis of buckling,  $r$ . To this end, in this study four section lists are generated where the available steel sections are ordered according to only one of the aforementioned sectional properties in each list. Then for each IG member one of these four lists is selected randomly, and the member adopts a stronger section in the selected list in terms of the corresponding cross-sectional property.

On the other hand, the DG members are stochastically decreased in size based on a predefined maximum decremental step size, which is defined in the same way as the

maximum incremental step size in this study. A random move towards smaller sections is then performed for each of the DG members through Eq. (5.6) where  $Rand^{DG}$  in this equation is an integer random number selected between 1 and the maximum decremental step size (i.e. 16 to 3) according to a uniform distribution. The resizing strategy followed in the GSS provides stochastic yet guided moves in the design space to approach the optimum.

**Step 6. Updating the rate of member resizing:** This step is implemented to determine the number of structural members considered for resizing at each iteration. In this regard an adaptive approach is employed based on the feasibility of the generated design at each iteration using some rules. The first rule is that  $IG_s$  members, those violating strength constraints, are all increased in size due to the high importance of strength criteria for producing acceptable designs. Secondly, as mentioned before, not all but rather a certain percentage of structural members are selected as  $DG_w$ ,  $DG_r$ , and  $IG_d$  members for size change based on the associated  $R_w$ ,  $R_r$ , and  $R_d$  ratios. For weight reduction  $R_w$  and  $R_r$  are initially set to a minimum value of 0.05, implying that only 5% of the members (design variables) are subjected to a section decrease due to excessive strength capacity. For handling the displacement criteria  $R_d$  is taken as 0.1, implying that 10% of the members will be increased in size due to violation of displacement criteria.

In the GSS, to ensure an effective weight reduction process, the  $R_w$  and  $R_r$  ratios are increased by  $\Delta R$  at each iteration provided that no constraint violation takes place in the generated design. It is apparent that, increasing the value of these ratios results in contribution of more structural elements in weight reduction. However, in case any kind of constraint violation (either displacement or strength criteria violation) takes place in the generated design, the values of  $R_w$  and  $R_r$  are gradually reduced towards their initial values twice faster (i.e. reduced by  $2\Delta R$ ) to expedite repair of the infeasible design.

In the case of  $IG_d$  members the initial value of  $R_d = 0.1$  can be kept constant

throughout the optimization process, and no increment of this parameter is indeed necessary. However, in case no improvement is provided for reducing the displacement violations, the same adaptive approach used for  $R_w$  and  $R_r$  can be employed for  $R_d$  as well to increase the number of structural members contributing to satisfy the displacement criteria. It is worth mentioning that an upper bound value of  $R_{max} = 0.7$  is taken for both  $R_w$  and  $R_d$  ratios, and an upper bound of  $0.5R_{max}$  is taken for  $R_r$  in this study.

**Step 8. Termination:** The aforementioned procedure is iteratively carried out, starting from the last generated design at the end of each iteration, until a stopping criterion is satisfied. The termination criterion can be imposed as a maximum number of iterations or no improvement of the best design over a certain number of iterations.

#### 6.4 Local Search and Move-Back Mechanisms in GSS

Local search capability of an optimization method plays an important role in the quality of final solutions generated. Here, a local search mechanism is integrated with the GSS to enhance its search performance. The local search mechanism is planned to work complementarily and parallel to the GSS procedure, such that at the end of each iteration cycle the vicinity of the best design found so far by GSS is investigated by decreasing a small fraction of frame members in size. The fraction of design variables to be changed during local search is limited to %5 of the total variables in the first iteration and it is linearly decreased to %1 in the last iteration. Further, the maximum decrement step size during local search is set to  $\sqrt{N_s} / 2 = 8$  in the first iteration and it is linearly decreased towards a final value of 3 during the iterations of design optimization procedure. To avoid duplicated solutions, a simple mechanism is employed where if the newly generated design is same as the original solution, the local search is forced to choose and update one design variable randomly.

Care should be taken not to deteriorate the global search features of the GSS

technique by local search. To this end, in case a better design is located during the local search, the current design is updated in the GSS only if the amount of improvement in the objective function value is more than a predefined limit named as transfer criteria (TC). For instance setting  $TC=0.05$  means that if the solution found by the local search mechanism is 5% better than the best solution found by the GSS so far, the better solution can be transferred to the GSS, otherwise GSS continues its search process in the solution space regardless of the solution found by the local search mechanism. In this study the value of TC is set to 0.05 in the first iteration and it is reduced linearly to 0.01 in the last iteration. This gradual reduction makes it possible to perform a more efficient local search in the last iterations since the probability of feeding the GSS by local search increases by decreasing the value of TC. It should be noted that the local search's best solution is always updated when a better solution is located by the GSS. However, local search cannot always update the GSS unless its solution meets the TC criterion. This strategy makes it possible to perform a parallel local search along with the GSS technique without degenerating its global search features.

The GSS inherently moves between the feasible and infeasible regions of the design space when searching for the optimum. While repairing an infeasible solution, sometimes it generates some oversized solutions that are feasible yet have a high design weight. Even though carried on from such a solution, GSS will eventually locate a good design in the next iterations. However, this may take some additional iterations, resulting in slower convergence of the algorithm especially if the search is continued from a highly oversized solution. In order to overcome this problem, a move-back mechanism is incorporated into the GSS in conjunction with an overdesign criterion parameter ( $OC=0.15$ ). Accordingly, in GSS the objective function value of the best design obtained thus far from the beginning of the optimization process is continuously stored and updated. If any time GSS locates a solution with a structural (net) weight which is 15% greater than the best objective function value, this solution is automatically replaced by the-so-far-best solution, and the search is carried on from this point over again.

In the GSS, infeasible designs that violate some of the problem constraints are penalized using an external penalty function approach, and their objective function values are computed according to Eq. (3.1). Furthermore, in case of member grouping the cumulative effect of all the group members should be taken into account to determine the SI value of a group. In such cases, in light of Eq. (5.3) sum of the DPFs of group members can be divided by the total volume of the group members. Once LCR and SI values are obtained, these values are used by the GSS to determine the resizing scheme. For the sake of clarity, the overall flowchart of frame optimization procedure via the GSS is depicted in Figure 6.1.

## 6.5 Numerical Examples

This section covers performance evaluation of the proposed GSS in discrete sizing optimization of steel frames. The investigated examples consist of three real-size steel frame structures with 135, 3860, and 11540 structural members. The optimum designs to these frames with the GSS are sought by implementing the algorithm over a predefined number of iterations. In order to evaluate the accuracy of the final solutions obtained with the GSS, the optimum solutions are also attained using some robust metaheuristic algorithms, and the results are compared. The value of parameter  $\Delta R$  is set to 0.1 for the first example, and 0.05 for the last two examples. For all the investigated instances the upper limit of interstory drift is taken as  $h/400$ , where  $h$  is the story height. The material properties of steel are taken as follows: modulus of elasticity ( $E$ ) = 200 GPa, yield stress ( $F_y$ ) = 248.2 MPa, and unit weight of the steel ( $\rho$ ) = 7.85 ton/m<sup>3</sup>.



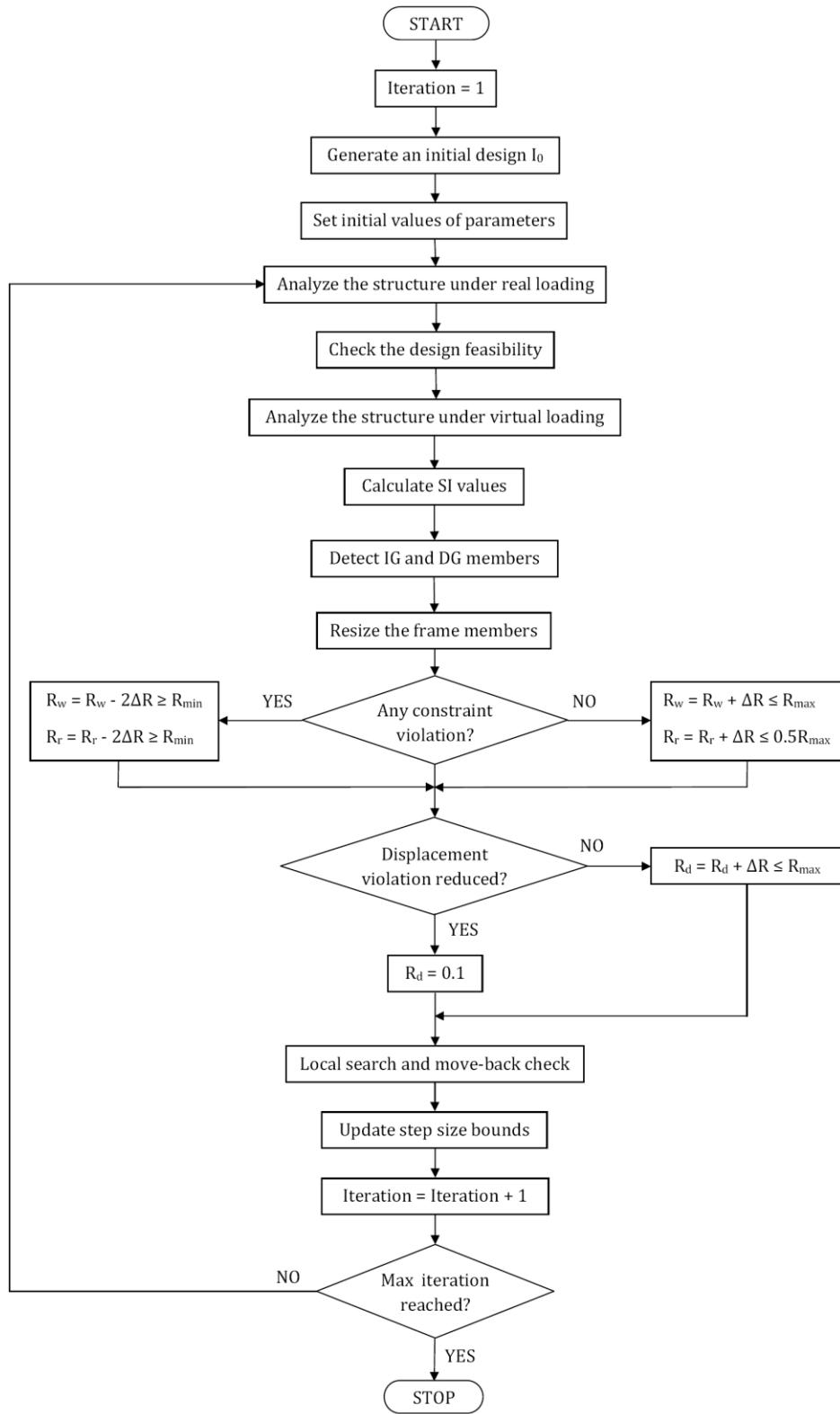


Figure 6.1: Flowchart of frame optimization process using GSS

### 6.5.1 Example 1: 135-Member Steel Frame

The 135-member steel frame shown in Figure 4.1 is studied in chapter 4 using the proposed computationally enhanced variants of the BB-BC algorithm (i.e. UBB-BC, UMBB-BC, and UEBB-BC). Here, the same example is used for performance evaluation of the GSS in discrete sizing optimization of frame structures. Furthermore, the optimization results obtained using a UBS integrated particle swarm optimization algorithm (UPSO) (Kazemzadeh Azad and Hasançebi 2013) is also considered for comparison purpose. All the above-mentioned metaheuristic algorithms are executed over a predefined number of maximum iterations, which is set to 350 for this example.

A comparison of optimal solutions attained using different algorithms is carried out in Table 6.1. As shown in this table, the GSS yields the least design weight for this example, which is only 37.12 ton. Other solutions obtained are 38.91 ton by UEBBBC, 45.67 ton by UMBB-BC, 47.3 ton by UBB-BC, and 55.66 ton by UPSO. Totally 750 structural analyses are performed by the GSS over 250 iterations, in which at each iteration of the algorithm the generated design is analyzed three times (two analyses for real and virtual loading plus one analysis for local search). On the other hand, the abovementioned heavier design weights are obtained using 1235 analyses by UEBBBC, 1794 analyses by UMBB-BC, 880 analyses by UBB-BC, and 1574 analyses by UPSO. Here, for the optimum design located by the GSS the maximum interstory drift is 99.41% of the allowable limit value; the maximum lateral displacement of the top story is 96.67% of the allowable displacement; and the load to capacity ratio for the most critical frame member is 0.999.

The variation of the best feasible design located in the optimization process with GSS is depicted in Figure 6.2. Figure 6.3 shows the convergence curves obtained with the metaheuristic algorithms, where the variation of penalized weight of the best design located is plotted against the iterations for each technique.

Table 6.1. Optimum designs obtained for 135-member steel frame

Groups	UPSO	UBB-BC	UMBB-BC	UEBB-BC	GSS
CG <sub>1</sub> *	W8X28	W10X39	W30X90	W21X62	W16X36
CG <sub>2</sub>	W33X118	W27X84	W14X48	W14X48	W18X60
CG <sub>3</sub>	W40X167	W40X14	W40X215	W36X150	W12X40
CG <sub>4</sub>	W14X53	W18X65	W27X84	W21X68	W30X10
B <sub>1</sub> *	W14X30	W21X44	W14X34	W18X40	W24X55
B <sub>2</sub>	W24X55	W16X40	W12X35	W18X35	W18X35
B <sub>3</sub>	W16X26	W10X22	W18X35	W16X26	W12X19
BR <sub>1</sub> *	W14X30	W27X84	W21X44	W8X24	W8X24
BR <sub>2</sub>	W40X149	W16X26	W10X22	W16X26	W6X15
BR <sub>3</sub>	W27X84	W21X44	W6X15	W6X15	W4X13
Weight (ton)	55.66	47.3	45.67	38.91	37.12
No. Analyses	1574	880	1794	1235	750

\*CG denotes column group with respect to Figure 4.2, B<sub>i</sub>: beams and BR<sub>i</sub>: bracings of the i-th story

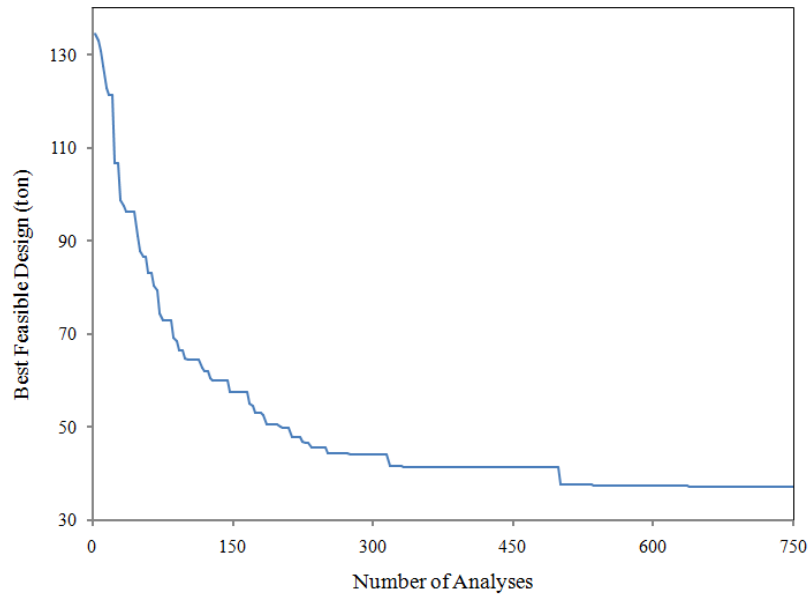


Figure 6.2: Optimization history of 135-member steel frame using GSS

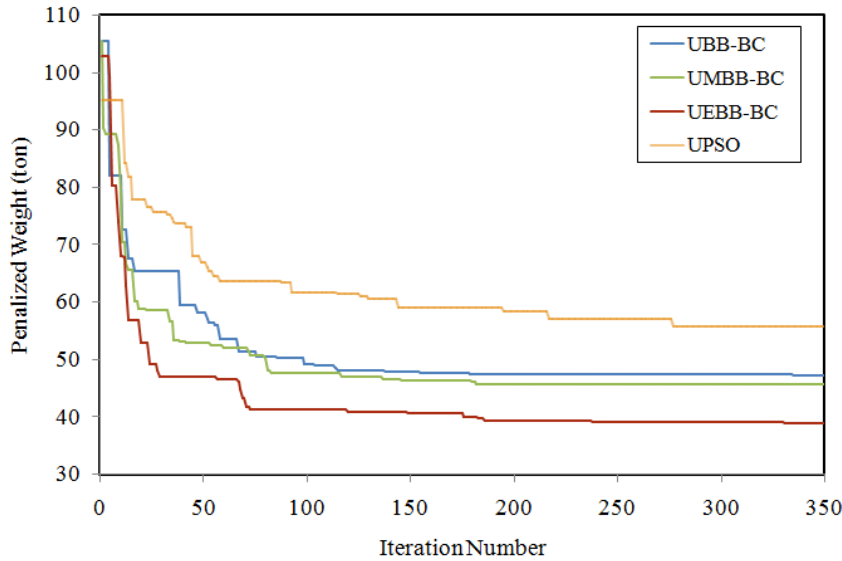


Figure 6.3: Optimization histories of 135-member steel frame using some metaheuristics

### 6.5.2 Example 2: 3860-Member Steel Frame

The second design example refers to a 20-story steel frame shown in Figure 6.4, consisting of 3860 structural members, including 1836 beam, 1064 column and 960 bracing elements. The stability of structure is provided through moment resisting connections as well as X-type bracing systems along the  $x$  and  $y$  directions. The 3860 members of the frame are collected under 73 member groups owing to practical fabrication requirements. The member grouping is performed in both plan and elevation levels. In elevation level the structural members are grouped in every two stories. In plan level columns are collected in 5 different column groups ( $CG_1$  through  $CG_5$ ) as depicted in Figure 6.5; beams are divided into two groups as outer and inner beams; and bracings are assumed to be in one group. Therefore, based on both elevation and plan level groupings, there are totally 43 column groups, 20 beam groups, and 10 bracing groups, resulting in 73 sizing design variables in all for this example.

For design purpose, the frame is subjected to the same 10 load combinations used for the first example. The live loads acting on the floor and roof beams are 10 and 7 kN/m, respectively. The dead loads consist of the self-weight of the structure in addition to the uniformly distributed loads of 14 and 12 kN/m applied on floor and roof beams, respectively.

The earthquake loads, are calculated based on the same procedure used for the first example. Here, the resulting seismic base shear ( $V$ ) is taken as  $V = 0.1W_s$  where  $W_s$  is the total dead load of the building. Further, in Eq. (4.2),  $C_T$  is taken as 0.0488 and  $h_n$  is 70 m. Hence, the period of the structure,  $T$ , is approximately 1.181 sec. Based on the period obtained the value of parameter  $k$  in Eq. (4.1) is taken as 1.341 for this example.

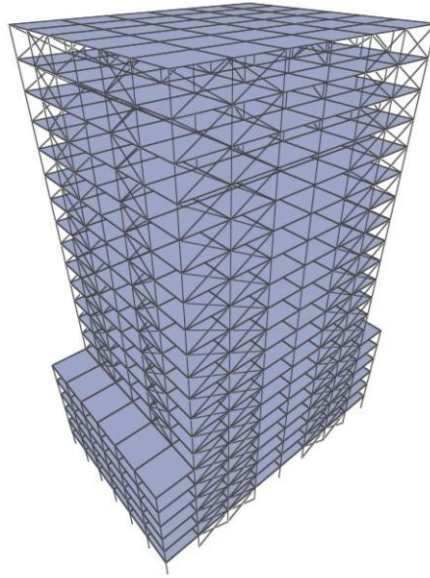
The beam elements are continuously braced along their lengths by the floor system; and columns and bracings are assumed to be unbraced along their lengths. The effective length factor,  $K$ , for buckling of columns as well as beams and bracings is taken as 1. For this design instance the maximum lateral displacement of the top story is limited to 0.18 m.

Optimum desing of the frame is carried out using the GSS algorithm. In Table 6.2, the minimum weight design obtained with GSS is tabulated and compared with the solution obtained using the UEBC algorithm (Kazemzadeh Azad et al. 2014b). In this example, the GSS produces the least weight of 3539.83 ton for the frame. The design weight attained by UEBC is 4117.43 ton which is much heavier than the solution located by the GSS. The GSS locates the minimum weight design over 300 iterations by performing only 900 structural analyses, whereas, the above-mentioned heavier design is obtained by UEBC using 9979 structural analyses. The optimization histories of the algorithms are presented in Figures 6.6 and 6.7.

Table 6.2. Optimum designs obtained for 3860-member steel frame

Stories	Groups	UEBB-BC	GSS	Stories	Groups	UEBB-BC	GSS
1-2	CG <sub>1</sub> *	W27X146	W24X94	11-12	CG <sub>1</sub>	N.A.	N.A.
	CG <sub>2</sub>	W12X210	W27X368		CG <sub>2</sub>	W30X99	W21X182
	CG <sub>3</sub>	W36X359	W30X477		CG <sub>3</sub>	W12X305	W12X210
	CG <sub>4</sub>	W40X593	W40X199		CG <sub>4</sub>	W30X261	W24X192
	CG <sub>5</sub>	W8X67	W30X191		CG <sub>5</sub>	W44X230	W24X162
	IB*	W40X183	W12X152		IB	W44X230	W40X149
	OB*	W14X22	W10X30		OB	W40X149	W10X54
	BR*	W40X167	W14X82		BR	W16X77	W16X67
3-4	CG <sub>1</sub>	W40X167	W18X65	13-14	CG <sub>1</sub>	N.A.	N.A.
	CG <sub>2</sub>	W12X230	W36X439		CG <sub>2</sub>	W16X89	W14X159
	CG <sub>3</sub>	W36X650	W44X335		CG <sub>3</sub>	W40X174	W40X264
	CG <sub>4</sub>	W24X335	W36X170		CG <sub>4</sub>	W16X100	W27X102
	CG <sub>5</sub>	W14X68	W30X191		CG <sub>5</sub>	W10X100	W27X129
	IB	W16X26	W14X176		IB	W14X43	W10X77
	OB	W40X235	W12X40		OB	W36X135	W33X318
	BR	W12X53	W18X119		BR	W33X152	W16X67
5-6	CG <sub>1</sub>	W24X335	W18X71	15-16	CG <sub>1</sub>	N.A.	N.A.
	CG <sub>2</sub>	W27X178	W44X335		CG <sub>2</sub>	W12X79	W18X86
	CG <sub>3</sub>	W27X539	W40X297		CG <sub>3</sub>	W44X262	W36X232
	CG <sub>4</sub>	W36X439	W36X160		CG <sub>4</sub>	W24X250	W14X90
	CG <sub>5</sub>	W30X99	W27X161		CG <sub>5</sub>	W33X263	W12X79
	IB	W44X230	W18X71		IB	W30X132	W27X129
	OB	W14X26	W27X114		OB	W14X68	W10X26
	BR	W12X72	W33X152		BR	W21X62	W21X83
7-8	CG <sub>1</sub>	N.A.	N.A.	17-18	CG <sub>1</sub>	N.A.	N.A.
	CG <sub>2</sub>	W27X368	W40X235		CG <sub>2</sub>	W24X117	W21X83
	CG <sub>3</sub>	W18X234	W24X207		CG <sub>3</sub>	W40X167	W18X119
	CG <sub>4</sub>	W33X221	W14X159		CG <sub>4</sub>	W36X245	W16X67
	CG <sub>5</sub>	W40X321	W12X210		CG <sub>5</sub>	W30X292	W18X65
	IB	W12X72	W30X99		IB	W40X149	W18X35
	OB	W33X130	W8X35		OB	W33X141	W18X35
	BR	W12X72	W12X53		BR	W14X48	W10X45
9-10	CG <sub>1</sub>	N.A.	N.A.	19-20	CG <sub>1</sub>	N.A.	N.A.
	CG <sub>2</sub>	W30X326	W36X194		CG <sub>2</sub>	W36X300	W10X30
	CG <sub>3</sub>	W14X455	W24X250		CG <sub>3</sub>	W18X60	W10X33
	CG <sub>4</sub>	W12X120	W40X183		CG <sub>4</sub>	W44X230	W36X210
	CG <sub>5</sub>	W18X86	W21X147		CG <sub>5</sub>	W10X54	W10X100
	IB	W27X94	W40X211		IB	W21X50	W36X150
	OB	W44X230	W30X124		OB	W30X116	W16X45
	BR	W24X84	W18X76		BR	W24X62	W16X45
Weight (ton)						4117.43	3539.83
No. Analyses						9979	900

\*CG denotes column group with respect to Figure 6.5, IB: inner beams, OB: outer beams, BR: bracings



(a)

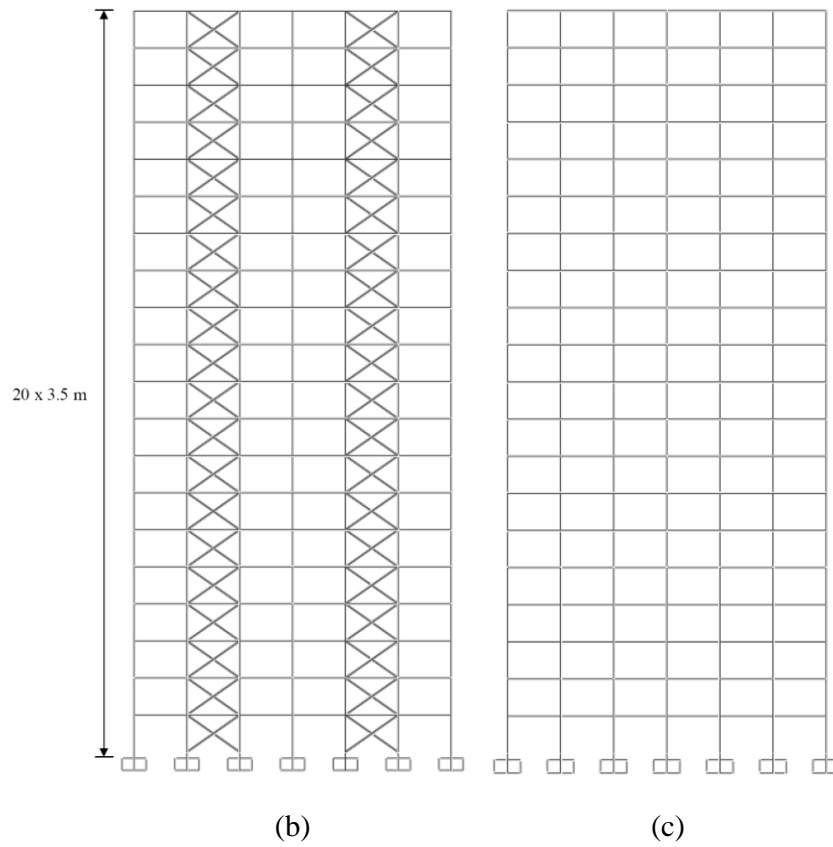
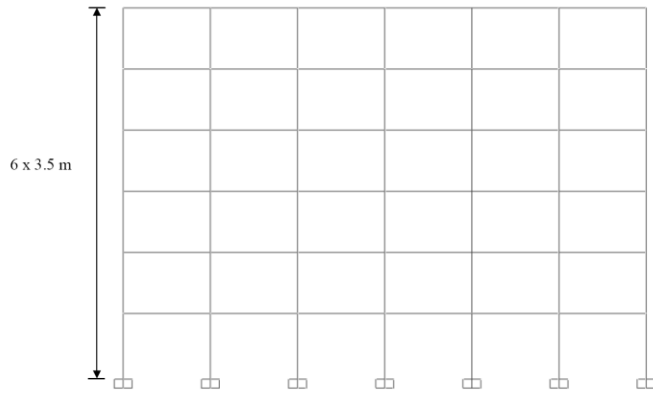
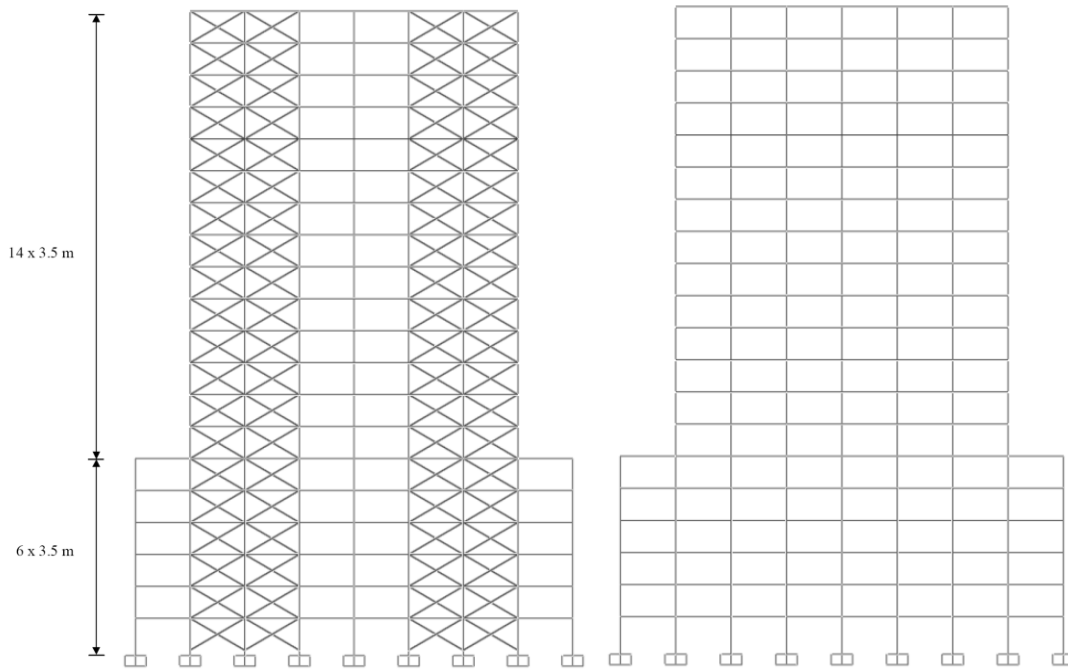


Figure 6.4: 3860-member steel frame, (a) 3-D view (b) side view of frames B, D, F, and H (c) side view of frames C, E, and G (d) side view of frames A, and I (e) side view of frames 1, 3, 5, and 7 (f) side view of frames 2, 4, and 6 (g) plan view



(d)

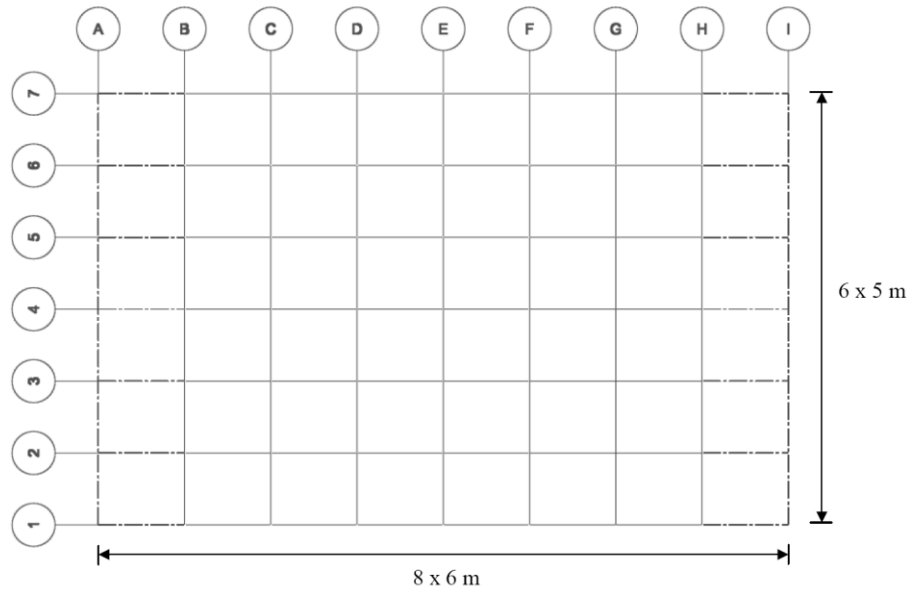


(e)

(f)

Figure 6.4 (continued)





(g)

Figure 6.4 (continued)

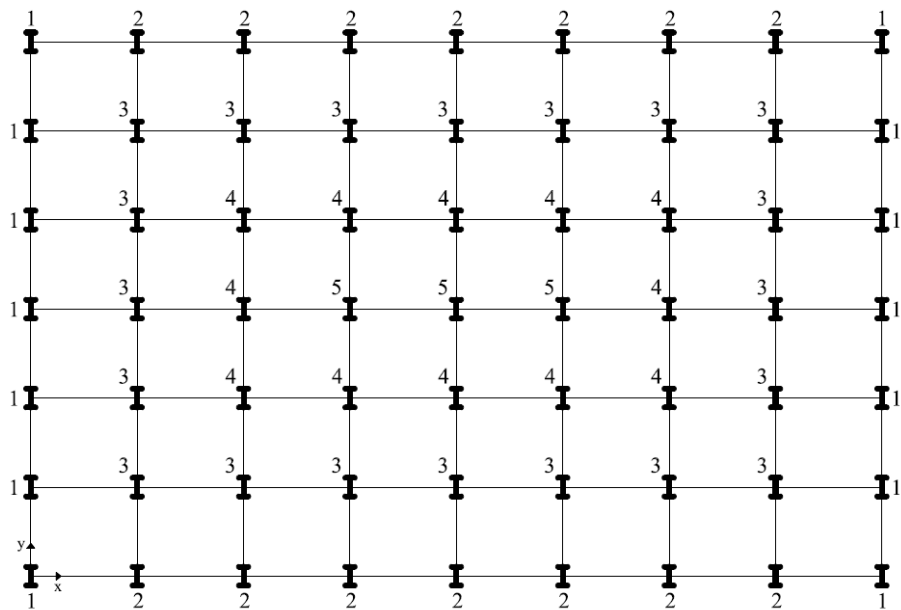


Figure 6.5: Column grouping in plan level for 3860-member steel frame

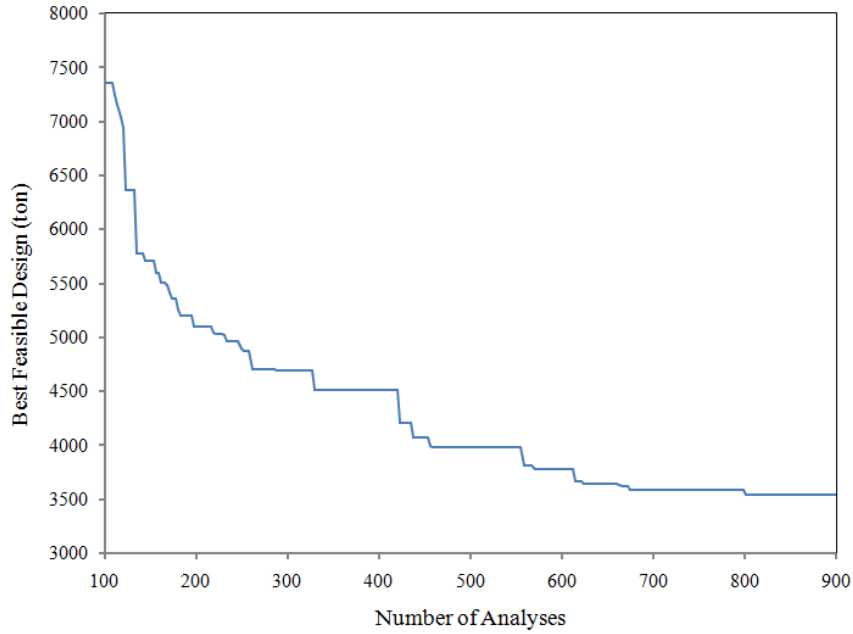


Figure 6.6: Optimization history of 3860-member steel frame using GSS

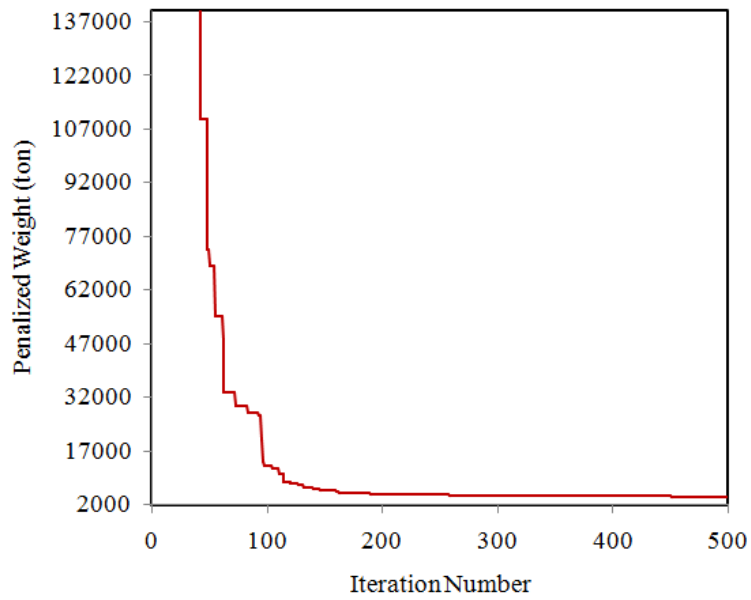


Figure 6.7: Optimization history of 3860-member steel frame using UEBB-BC

### 6.5.3 Example 3: 11540-Member Steel Frame

The third design example refers to a 20-story steel frame shown in Figure 6.8. The frame, which is one of the largest steel frame instances investigated so far in the literature, is composed of 11540 structural members, including 6240 beam, 3380 column and 1920 bracing elements. The stability of structure is provided through moment resisting connections as well as X-type bracing systems along the  $x$  and  $y$  directions. The 11540 members of the frame are collected under 100 member groups owing to practical fabrication requirements. The member grouping is performed in both plan and elevation levels. In elevation level the structural members are grouped in every two stories. In plan level, columns are collected in 7 different column groups ( $CG_1$  through  $CG_7$ ) as depicted in Figure 6.10 (where all columns located on each square are treated as one column group); beams are divided into two groups as outer and inner beams; and bracings are assumed to be in one group. Therefore, based on both elevation and plan level groupings, there are totally 70 column groups, 20 beam groups, and 10 bracing groups, resulting in 100 sizing design variables in all for this example. For the sake of clarity, columns' orientations of the frame are shown in Figure 6.9.

For design purpose the frame is subjected to the same 10 load combinations described in the first example. The live loads acting on the floor and roof beams are 12 and 7 kN/m, respectively. The dead loads consist of the self-weight of the structure in addition to the uniformly distributed loads of 15 and 12 kN/m applied on floor and roof beams, respectively.

The earthquake loads, are calculated based on the same procedure employed in the previous two examples. Here, the resulting seismic base shear ( $V$ ) is taken as  $V = 0.1W_s$  where  $W_s$  is the total dead load of the building. Further, in Eq. (4.2),  $C_T$  is taken as 0.0488 and  $h_n$  is 70 m. Hence, the period of the structure,  $T$ , is approximately

computed as 1.181 sec. Based on the obtained period the value of parameter  $k$  in Eq. (4.1) is taken as 1.341 for this example.

The beam elements are continuously braced along their lengths by the floor system; and columns and bracings are assumed to be unbraced along their lengths. The effective length factor,  $K$ , for buckling of columns as well as beams and bracings is taken as 1. In this example the maximum lateral displacement of the top story is limited to 0.18 m.

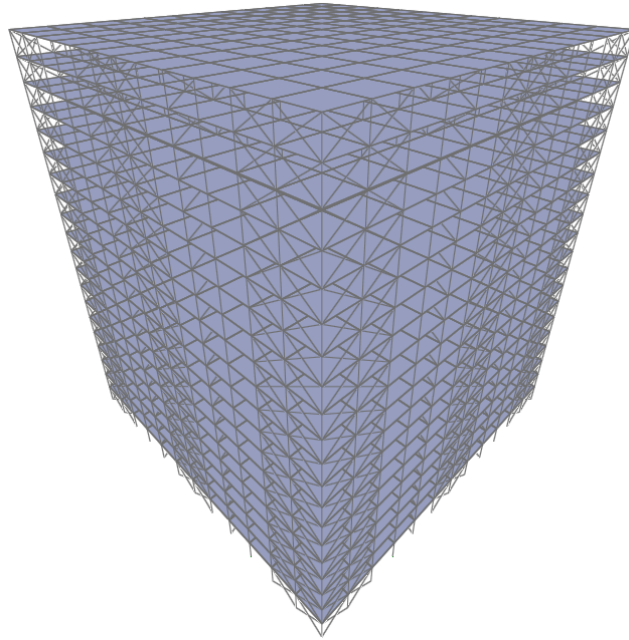
It is worth mentioning that optimum sizing of the frame is formerly studied in Kazemzadeh Azad et al. (2014b) using a wide-flange (W) profile list consisting of 162 ready sections between W16 and W44. A more challenging case including all the 268 ready sections is investigated here using the GSS.

In Table 6.3, the minimum weight design obtained with GSS is tabulated and compared with the solution obtained using UEBB-BC algorithm (Kazemzadeh Azad et al. 2014b). In this example, the GSS produces the least weight of 10707.53 ton while a heavier design weight of 10756.63 ton is achieved by UEBB-BC algorithm. The GSS locates the minimum weight design over 700 iterations by performing only 2100 structural analyses, whereas, the above-mentioned heavier design is attained by UEBBBC using 7616 structural analyses. The optimization histories of the algorithms are presented in Figures 6.11 and 6.12.

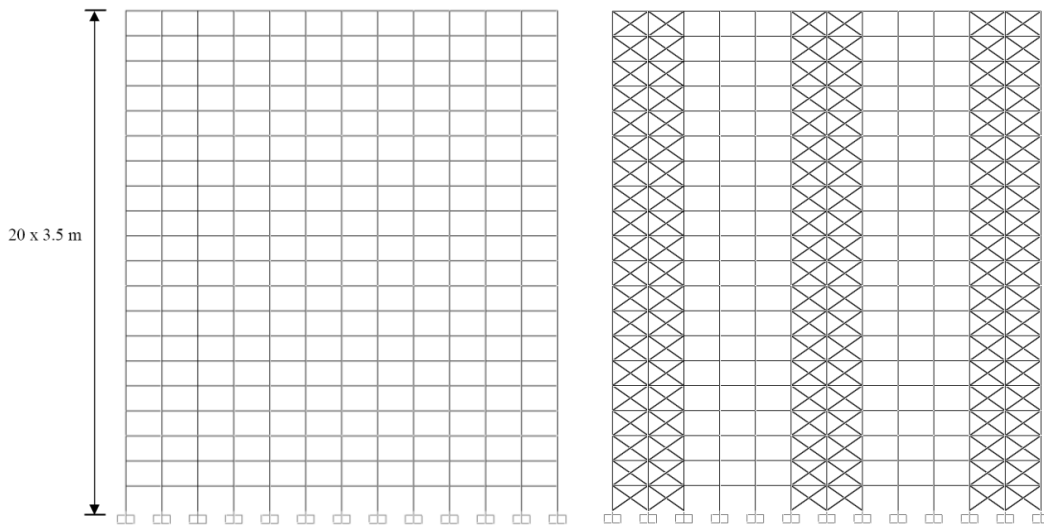
Table 6.3. Optimum designs obtained for 11540-member steel frame

Stories	Groups	UEBB-BC	GSS	Stories	Groups	UEBB-BC	GSS
1-2	CG <sub>1</sub> *	W36X393	W40X431	11-12	CG <sub>1</sub>	W24X408	W14X176
	CG <sub>2</sub>	W40X249	W33X291		CG <sub>2</sub>	W33X118	W12X106
	CG <sub>3</sub>	W36X393	W36X300		CG <sub>3</sub>	W18X234	W30X235
	CG <sub>4</sub>	W40X297	W12X190		CG <sub>4</sub>	W36X328	W40X431
	CG <sub>5</sub>	W44X335	W36X393		CG <sub>5</sub>	W30X235	W36X256
	CG <sub>6</sub>	W18X258	W12X279		CG <sub>6</sub>	W36X170	W33X221
	CG <sub>7</sub>	W40X431	W40X264		CG <sub>7</sub>	W27X146	W36X170
	IB*	W18X40	W18X35		IB	W40X199	W40X183
	OB*	W21X57	W27X146		OB	W33X291	W8X28
BR*	W18X211	W18X130	BR	W21X62	W30X124		
3-4	CG <sub>1</sub>	W24X250	W30X477	13-14	CG <sub>1</sub>	W24X250	W27X307
	CG <sub>2</sub>	W40X372	W40X199		CG <sub>2</sub>	W33X118	W24X131
	CG <sub>3</sub>	W44X262	W30X261		CG <sub>3</sub>	W30X191	W33X118
	CG <sub>4</sub>	W36X256	W40X174		CG <sub>4</sub>	W40X264	W33X118
	CG <sub>5</sub>	W40X277	W33X263		CG <sub>5</sub>	W21X93	W36X170
	CG <sub>6</sub>	W27X258	W27X178		CG <sub>6</sub>	W33X241	W36X280
	CG <sub>7</sub>	W33X291	W12X190		CG <sub>7</sub>	W16X89	W21X132
	IB	W16X26	W24X84		IB	W27X84	W24X55
	OB	W18X40	W36X182		OB	W33X291	W27X235
BR	W33X130	W30X211	BR	W27X129	W21X62		
5-6	CG <sub>1</sub>	W44X290	W24X335	15-16	CG <sub>1</sub>	W24X117	W30X211
	CG <sub>2</sub>	W18X234	W40X211		CG <sub>2</sub>	W24X84	W14X68
	CG <sub>3</sub>	W44X335	W18X283		CG <sub>3</sub>	W36X256	W12X96
	CG <sub>4</sub>	W36X245	W27X307		CG <sub>4</sub>	W36X232	W14X159
	CG <sub>5</sub>	W33X241	W36X280		CG <sub>5</sub>	W16X89	W14X74
	CG <sub>6</sub>	W18X258	W40X199		CG <sub>6</sub>	W33X130	W14X74
	CG <sub>7</sub>	W44X290	W36X170		CG <sub>7</sub>	W40X167	W12X58
	IB	W40X174	W40X235		IB	W24X62	W18X35
	OB	W40X199	W12X26		OB	W18X60	W16X36
BR	W27X94	W12X72	BR	W21X68	W14X61		
7-8	CG <sub>1</sub>	W21X182	W36X359	17-18	CG <sub>1</sub>	W18X192	W33X318
	CG <sub>2</sub>	W33X152	W24X117		CG <sub>2</sub>	W24X117	W12X96
	CG <sub>3</sub>	W18X175	W44X262		CG <sub>3</sub>	W33X118	W40X167
	CG <sub>4</sub>	W33X291	W40X215		CG <sub>4</sub>	W44X262	W30X90
	CG <sub>5</sub>	W27X178	W14X283		CG <sub>5</sub>	W44X335	W21X62
	CG <sub>6</sub>	W40X199	W27X194		CG <sub>6</sub>	W40X297	W18X65
	CG <sub>7</sub>	W24X250	W30X173		CG <sub>7</sub>	W36X160	W14X48
	IB	W36X135	W10X22		IB	W36X150	W14X22
	OB	W24X68	W36X280		OB	W24X62	W40X321
BR	W40X174	W36X135	BR	W21X62	W12X58		
9-10	CG <sub>1</sub>	W40X372	W40X297	19-20	CG <sub>1</sub>	W18X119	W30X132
	CG <sub>2</sub>	W33X169	W36X135		CG <sub>2</sub>	W33X152	W27X102
	CG <sub>3</sub>	W24X250	W12X152		CG <sub>3</sub>	W24X76	W24X131
	CG <sub>4</sub>	W30X261	W33X291		CG <sub>4</sub>	W16X36	W30X124
	CG <sub>5</sub>	W24X335	W30X191		CG <sub>5</sub>	W16X36	W12X53
	CG <sub>6</sub>	W27X178	W36X280		CG <sub>6</sub>	W18X143	W8X28
	CG <sub>7</sub>	W27X448	W18X119		CG <sub>7</sub>	W30X108	W6X15
	IB	W24X55	W33X130		IB	W24X55	W40X278
	OB	W40X277	W10X30		OB	W40X167	W12X26
BR	W27X94	W24X94	BR	W18X40	W8X31		
Weight (ton)						10756.63	10707.53
No. Analyses						7616	2100

\*CG denotes column group with respect to Figure 6.10, IB: inner beams, OB: outer beams, BR: bracings



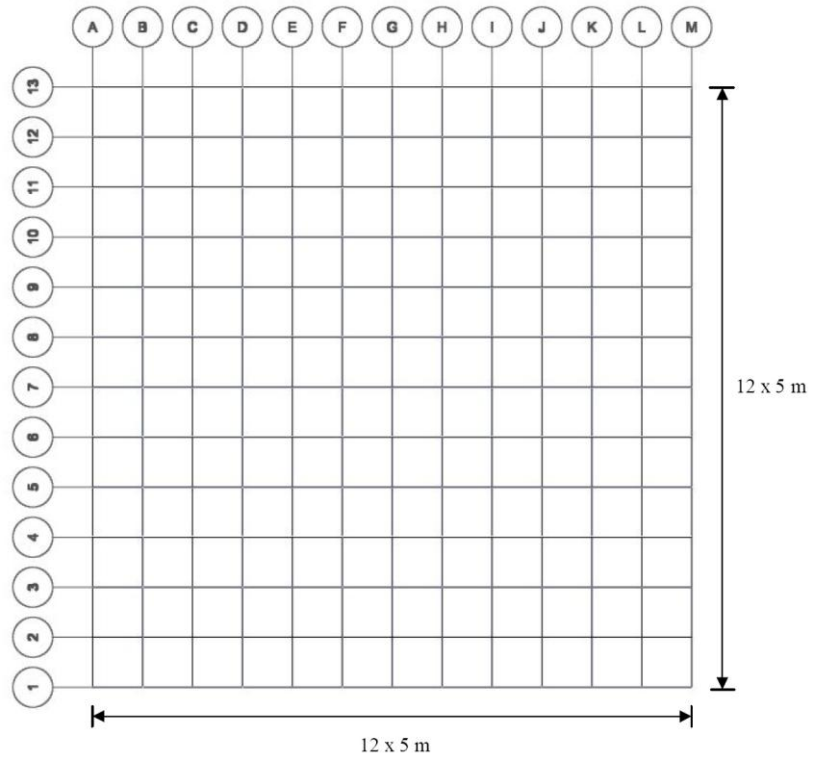
(a)



(b)

(c)

Figure 6.8: 11540-member steel frame, (a) 3-D view (b) side view of frames 2, 3, 4, 6, 7, 8, 10, 11, 12, B, C, D, F, G, H, J, K and L (c) side view of frames 1, 5, 9, 13, A, E, I, and M (d) plan view



(d)

Figure 6.8 (continued)

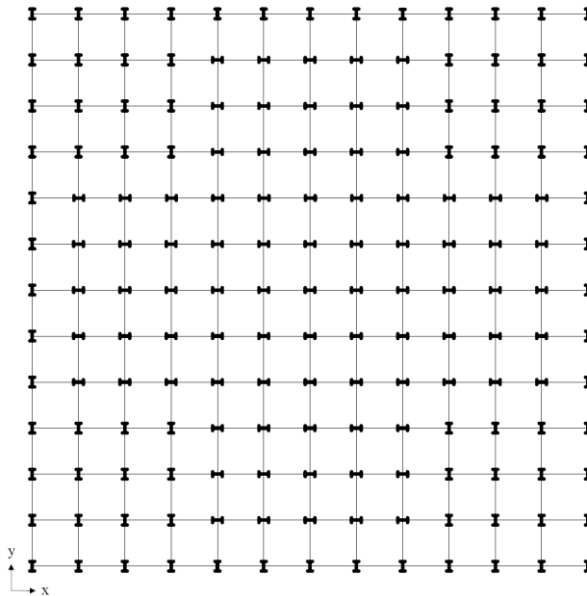


Figure 6.9: Columns' orientations of 11540-member steel frame

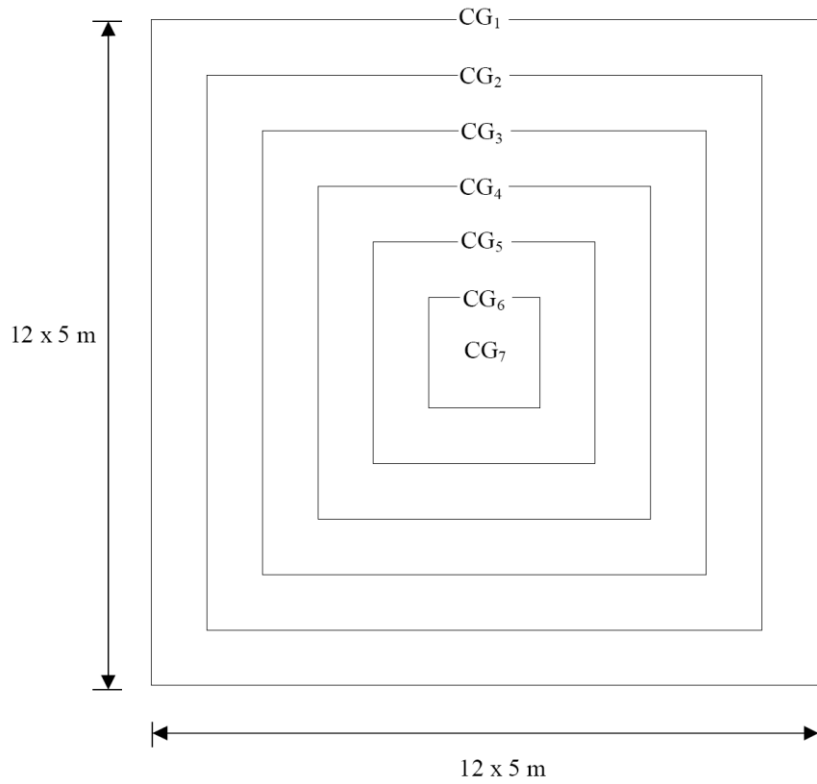


Figure 6.10: Outline of column grouping in plan level for 11540-member steel frame

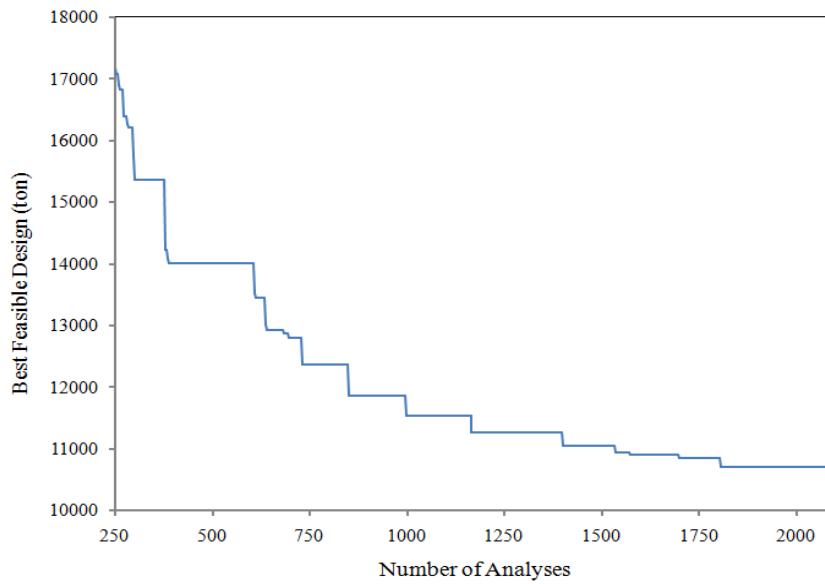


Figure 6.11: Optimization history of 11540-member steel frame using GSS



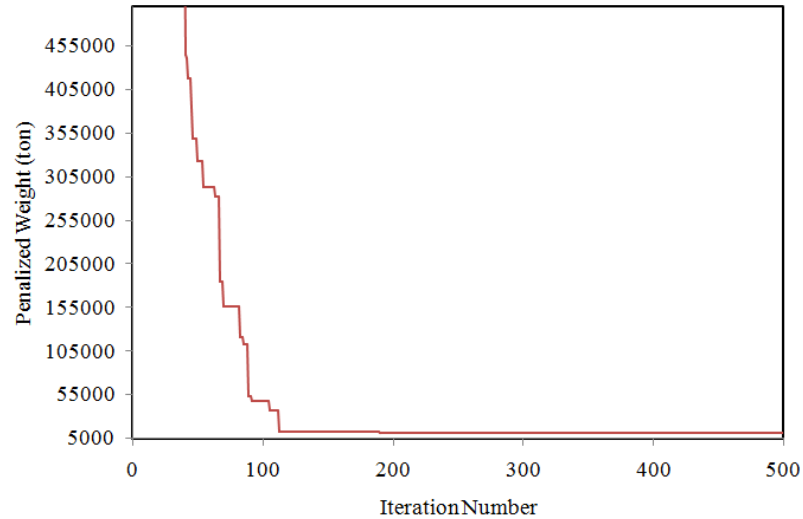


Figure 6.12: Optimization history of 11540-member steel frame using UEBC-BC

## 6.6 Summary

In this chapter the GSS technique is refined and reformulated for discrete sizing optimization of steel frame structures under practical design considerations. The method works on the basis of first identifying the members to resize using the well-known principal of virtual work and load to capacity ratios of structural members. The resizing of members is then carried out in a stochastic way. Basically, two critical groups of frame members (IG, and DG) are detected at each iteration such that IG members are increased in size to satisfy strength and/or displacement constraints, and meanwhile DG members are decreased in size to minimize the structural weight in the course of optimization.

Furthermore, local search and move-back mechanisms are incorporated into the GSS in order to enhance its efficiency in discrete sizing optimization problems. The performance evaluation of the GSS is performed based on a comparison with some contemporary metaheuristic approaches using three real-size steel frame structures, namely 135, 3860, and 11540-member frames that are designed according to AISC-LRFD (1994) specifications. The results obtained in these examples indicate the efficiency of the proposed GSS technique in discrete sizing optimization of frame

structures. Aside from the promising performance of the GSS, the fact that its performance is not affected by random initiation, and no gradient information is needed to guide the optimization process makes it very favorable for design optimization of steel frames. Besides, it requires considerably low computational effort, which in turn shows how domain knowledge can be utilized to achieve robust and computationally efficient design optimization of steel frames in practical applications.

## CHAPTER 7

### CONCLUSION

#### 7.1 Summary and Concluding Remarks

The thesis covers developing computationally efficient discrete sizing optimization techniques for optimal design of steel truss and frame structures subject to strength and serviceability constraints imposed by well known standard design codes. Basically two main strategies are followed in the study as (i) investigating the algorithmic structure of the existing metaheuristic techniques and enhancing their performances in discrete sizing optimization problems (ii) developing new design-driven optimization techniques. As outlined below, chapters 3 and 4 focus on the first approach whereas chapters 5 and 6 cover the latter.

In chapter 3 through investigating the shortcomings of the well known BB-BC algorithm, it is shown that the standard version of the algorithm is sometimes unable to locate reasonable solutions to problems from discrete sizing optimization problems. Accordingly, improved variants of the BB-BC algorithm for this class of problems are proposed, where the formula used by the standard algorithm for generating new candidate solutions around the center of mass is efficiently reformulated, resulting in the so called MBB-BC and EBB-BC algorithms. The performance evaluation of the proposed algorithms using discrete sizing optimization instances of steel skeletal structures demonstrated their efficiency and robustness in locating promising solutions.

Although, in locating reasonable solutions, the MBB-BC and EBB-BC algorithms have shown superior performances compared to the standard BB-BC algorithm as

well as other metaheuristic techniques, the number of structural analysis required still is not ideal, especially for large scale applications. Hence, an upper bound strategy (UBS) for reducing the total number of structural analysis in metaheuristic based design optimization algorithms is proposed in chapter 4.

The basic idea behind the UBS is to detect those candidate designs which have no chance to improve the search during the iterations of the optimization algorithm. After identifying the non-improving candidate designs, they are directly excluded from the structural analysis stage, diminishing the total computational effort. The numerical results obtained from optimum design of two real size steel frames using the UBS integrated optimization algorithms clearly demonstrated the usefulness of this strategy in reducing the total computational time of the design optimization.

In chapter 5 a guided stochastic search (GSS) technique is developed for computationally efficient optimum design of steel trusses. In the GSS a stochastic procedure is employed wherein the search direction is determined by the principle of virtual work and response computations of the generated designs, resulting in an efficient and rapid search. In the proposed method, the information provided in the structural analysis and design check stages are utilized for handling strength constraints. Moreover, the well-known principle of virtual work is used to detect the most effective structural members for satisfying displacement constraints. The optimum sizing of a structure is then performed using an integrated approach wherein both strength and displacement criteria are taken into account for reduction of the member sizes along the way the aforementioned constraints are handled.

The GSS technique developed is first evaluated through optimal sizing problems of steel truss structures with a single displacement constraint under a single load case. Nevertheless, practical designs of structures usually come up with numerous displacement criteria in several load cases, and it is crucial that an optimization algorithm can address such practical requirements of design problems in reality. Therefore, the GSS technique is further improved for handling a more general class of

truss optimization problems subject to multiple displacement constraints and load cases. For this purpose, enhancements of the GSS are proposed in the form of two alternative approaches that enable the technique to deal with multiple displacement/load cases. The first approach implements a methodology in which the most critical displacement direction is considered only when guiding the search process. The second approach, however, takes into account the cumulative effect of all the critical displacement directions in the course of optimization. Furthermore, advantage of the integrated force method of structural analysis is utilized for further reduction of the computational effort in the aforementioned approaches. The performance of the proposed GSS technique is evaluated through challenging optimum design instances of steel truss structures. The comparison of numerical results obtained using the GSS to those of different metaheuristic techniques revealed its computational efficiency in practical discrete sizing optimization problems.

Refinement and reformulation of the GSS technique is carried out in chapter 6 to make the algorithm suitable for handling sizing optimization problems of steel frame structures. On the one hand, a local search mechanism is proposed to increase the quality of solutions found during the search process without deteriorating the global search features of the GSS. On the other hand, a move-back mechanism is employed to avoid additional iterations required for modifying probable highly oversized solutions in the course of optimization. The developed refined and reformulated GSS technique is compared with some modern metaheuristics using three practical instances of steel frames. The attained numerical results demonstrated the computational efficiency of the proposed method in discrete sizing of steel frames.

## **7.2 Recommendations for Future Research**

When dealing with optimum design of structural systems, besides the size of the considered structure, another important criterion is the dimension of the solution space which significantly affects the difficulty of the optimum design problem. Basically, by increasing the number of design variables consequently the dimension of solution

space increases drastically such that locating the optimum or even a good near optimum solution in a timely manner becomes a cumbersome task. This type of difficulty that arises when dealing with high dimensional problems is referred to as the “curse of dimensionality” introduced by Bellman (1957, 1961). Fortunately, by emergence of modern computing technologies, indeed there is no lack of optimization algorithms capable of handling small size structures having a few design variables (say 10 or 25). However, the optimum design problem of large scale structural systems including numerous design variables (say more than 1000 variables) is not properly addressed in the literature of structural optimization. Therefore, in order to overcome the curse of dimensionality in large scale structural optimization, investigating the performance of the developed techniques (in particular the GSS) as well as adapting design-driven optimization methods for handling high dimensional problems can be a good research topic.

Finally, since this research covers discrete sizing optimization instances only, investigating the applicability of the proposed techniques to the other types of structural optimization instances such as problems including frequency constraints, geometrical or material nonlinearities as well as shape or topology variables can be fruitful.

## REFERENCES

Afshar, M.H., and Motaei, I. (2011). Constrained big bang–big crunch algorithm for optimal solution of large scale reservoir operation problem. *International Journal of Optimization in Civil Engineering*, 1, 357–375.

AISC-ASD. (1989). *Manual of Steel Construction-Allowable Stress Design*, 9th ed., Chicago, Illinois, USA.

American Institute of Steel Construction. (1994). *Manual of Steel Construction, Load and Resistance Factor Design*, 2nd ed., AISC, Chicago.

American Institute of Steel Construction. (1995). *Manual of steel construction, Load and Resistance Factor Design*, AISC, Chicago.

ANSI/AISC 360-05. (2005). *Specification for structural steel buildings*, Chicago, Illinois, USA, 2005.

American Society of Civil Engineers (ASCE). (1998). *Minimum design loads for buildings and other structures*.

American Society of Civil Engineers (ASCE 7-98). (2000). *Minimum design loads for buildings and other structures: Revision of ANSI/ASCE 7-95*.

American Society of Civil Engineers (ASCE 7-05). (2005), *Minimum design loads for building and other structures*.

Belegundu, A.D., and Arora J.S. (1985). A study of mathematical programming methods for structural optimization. Part II: Numerical results. *International Journal for Numerical Methods in Engineering*, 21(9), 1601–1623.

Bellman, R. (1957). *Dynamic programming*, Princeton University Press.

Bellman, R. (1961). *Adaptive control processes: a guided tour*, Princeton University Press.

British Standards Institution. (1990). *Structural use of steel works in building, Part 1, Code of practice for design in simple and continuous construction, hot rolled sections. BS 5950*, London.

Carbas, S., and Saka, M.P. (2009). Optimum design of single layer network domes using harmony search method. *Asian Journal of Civil Engineering*, 10(1), 97–112.

Camp, C.V. (2007). Design of space trusses using Big Bang–Big Crunch optimization. *Journal of Structural Engineering, ASCE*, 133, 999–1008.

Chan, C.-M., Grierson, D.E., and Sherbourne A.N. (1995). Automatic optimal design of tall steel building frameworks. *Journal of Structural Engineering*, 121(5), 838–847.

Charney, F.A. (1991) The use of displacement participation factors in the optimization of drift controlled buildings. In *Proc. 2nd Conf. on Tall Buildings in Seismic Regions, 55th Regional Conference*, Los Angeles, California, USA, 91–98.

Charney, F.A. (1993). Economy of steel framed buildings through identification of structural behavior, *National Steel Construction Conference*, Orlando, FL, AISC, 1–33.

Degertekin, S. O., Saka, M. P., and Hayalioglu, M.S. (2008). Optimal load and resistance factor design of geometrically nonlinear steel space frames via tabu search and genetic algorithm. *Engineering Structures*, 30(1), 197–205.



- Dorigo, M. (1992). Optimization, learning and natural algorithms, PhD thesis. Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy.
- Dumonteil, P. (1992). Simple equations for effective length factors. *Engineering Journal*, AISC, 29, 111–115.
- Erbatur, F., and Al-Hussainy, M.M. (1992). Optimum design of frames. *Computers and Structures*, 45(5–6), 887–891.
- Erol, O.K., and Eksin, I. (2006). A New optimization method: Big Bang–Big Crunch. *Advances in Engineering Software*, 37, 106–11.
- Feury, C., and Geradin, M. (1978). Optimality criteria and mathematical programming in structural weight optimization. *Computers and Structures*, 8(1), 7–17.
- Fleury, C. (1980). An efficient optimality criteria approach to the minimum weight design of elastic structures. *Computers and Structures*, 11(3), 163–173.
- Fourie, P.C., and Groenwold, A.A. (2002). The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization*, 23, 259–267.
- Gallagher, R.H. (1973). Fully stressed design. In: Gallagher RH, Zienkiewicz OC (eds.) *Optimum structural design: theory and applications*, Wiley, Chichester, 19–32.
- Geem, Z.W., Kim, J.H., and Loganathan, G.V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2), 60–68.
- Glover, F. (1989). Tabu Search-Part I. *ORSA Journal on Computing*, 1, 190–206.

Goldberg, D.E., Samtani, M.P. (1986). Engineering optimization via genetic algorithm. In Proceeding of the Ninth Conference on Electronic Computation, ASCE, 471–482.

Gomes, H.M. (2011). Truss optimization with dynamic constraints using a particle swarm algorithm. *Expert Systems with Applications*, 38, 957–968.

Hadidi, A., Kazemzadeh Azad, S., Kazemzadeh Azad, S. (2010). Structural optimization using artificial bee colony algorithm. The 2nd International Conference on Engineering Optimization (Eng Opt), Lisbon, Portugal.

Hall, S.K., Cameron, G.E., and Grierson, D.E. (1989). Least-weight design of steel frameworks accounting for P- $\Delta$  effects. *Journal of Structural Engineering*, ASCE, 115(6), 1463–1475.

Hare, W., Nutini, J., and Tesfamariam, S. (2013). A survey of non-gradient optimization methods in structural engineering. *Advances in Engineering Software*, 59, 19–28.

Hasançebi, O. (2008). Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures. *Computers and Structures*, 86 (1–2), 119–132.

Hasançebi, O., Çarbas, S., Dogan, E., Erdal, F., and Saka, M.P. (2009). Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. *Computers and Structures*, 87, 284–302.

Hasançebi, O., Erdal, F., and Saka, M.P. (2010a). Adaptive harmony search method for structural optimization. *Journal of Structural Engineering*, ASCE, 136(4), 419–431.

Hasançebi, O., Çarbas, S., Dogan, E., Erdal, F., and Saka, M.P. (2010b). Comparison of non-deterministic search techniques in the optimum design of real size steel frames. *Computers and Structures*, 88, 1033–1048.

Hasançebi, O., Erdal, F., and Saka, M.P., (2010c). Optimum design of geodesic steel domes under code provisions using metaheuristic techniques. *International Journal of Engineering and Applied Sciences*, 2 (2), 88–103.

Hasançebi, O., Bahçecioğlu, T., Kurç, Ö., Saka, M.P. (2011a) Optimum design of high-rise steel buildings using an evolution strategy integrated parallel algorithm. *Computers and Structures*, 89, 2037–2051.

Hasançebi, O., Çarbas, S., and Saka, M.P. (2011b). A reformulation of the ant colony optimization algorithm for large scale structural optimization. In Tsompanakis Y., Topping B.H.V., (Editors), *Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, Civil-Comp Press, Stirlingshire, UK.

Hasançebi, O., and Kazemzadeh Azad, S. (2012). An efficient metaheuristic algorithm for engineering optimization: SOPT. *International Journal of Optimization in Civil Engineering*, 2(4), 479–87.

Hasançebi, O., Kazemzadeh Azad, S. (2012). An exponential big bang-big crunch algorithm for discrete design optimization of steel frames. *Computers and Structures*, 110–111, 167–179.

Hasançebi, O., Kazemzadeh Azad, S., and Kazemzadeh Azad, S. (2013). Automated sizing of truss structures using a computationally improved SOPT algorithm. *International Journal of Optimization in Civil Engineering*, 3(2), 209–221.

Hasançebi, O., and Kazemzadeh Azad, S. (2014). Discrete Size Optimization of Steel Trusses using a Refined Big Bang–Big Crunch Algorithm. *Engineering Optimization*, 46(1), 61–83.

Hayalioglu, M. S., and Degertekin, S. O. (2005). Minimum cost design of steel frames with semi-rigid connections and column bases via genetic optimization. *Computers and Structures*, 83(21–22), 1849–1863.

Hellesland, J. (1994) Review and evaluation of effective length formulas, Research Report, No. 94–2, University of Oslo.

Kameshki, E. S., and Saka, M. P. (2001a). Genetic algorithm based optimum bracing design of non-swaying tall plane frames. *Journal of Constructional Steel Research*, 57, 1081–1097.

Kameshki, E. S., and Saka, M. P. (2001b). Optimum design of nonlinear steel frames with semi-rigid connections using a genetic algorithm. *Computers and Structures*, 79, 1593–1604.

Kameshki, E. S., and Saka, M. P. (2003). Genetic algorithm based optimum design of nonlinear planar steel frames with various semi-rigid connections. *Journal of Constructional Steel Research*, 59, 109–134.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey.

Kaveh, A., Abbasgholiha, H. (2011). Optimum design of steel sway frames using big bang–big crunch algorithm. *Asian Journal of Civil Engineering*, 12, 293–317.

Kaveh, A., and Kalatjari, V. (2002). Genetic algorithm for discrete-sizing optimal design of trusses using the force method. *International Journal for Numerical Methods in Engineering*, 55(1), 55–72.

Kaveh, A., and Khayatazad M. (2012). A new meta-heuristic method: Ray Optimization. *Computers and Structures*, 112–113, 283–294.

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

Kaveh, A., and Talatahari, S. (2009b). Size optimization of space trusses using big bang-big crunch algorithm. *Computers and Structures*, 87, 1129–1140.

Kaveh, A., and Talatahari, S. (2009c). Optimal design of Schwedler and ribbed domes via hybrid big bang-big crunch algorithm. *Journal of Constructional Steel Research*, 66, 412–419.

Kaveh, A., and Talatahari S. (2010a). A discrete big bang-big crunch algorithm for optimal design of skeletal structures. *Asian Journal of Civil Engineering*, 11, 103–123.

Kaveh, A., Talatahari, S. (2010b). A novel heuristic optimization method: charged system search. *Acta Mechanica*, 213, 267–289.

Kaveh, A., Talatahari, S. (2010c). Optimal design of skeletal structures via the charged system search algorithm. *Structural and Multidisciplinary Optimization*, 41, 893–911.

Kaveh, A., Talatahari, S. (2010d). A charged system search with a fly to boundary method for discrete optimum design of truss structures. *Asian Journal of Civil Engineering*, 11(3), 277–293.

Kaveh, A., Talatahari, S. (2011). An enhanced charged system search for configuration optimization using the concept of fields of forces, *Structural and Multidisciplinary Optimization*, 43(3), 339–51.

Kaveh, A., and Talatahari, S. (2012). Charged system search for optimal design of frame structures. *Applied Soft Computing*, 12(1), 382–393.

Kazemzadeh Azad, S., Hasançebi, O., and Erol, O.K. (2011). Evaluating efficiency of big bang-big crunch algorithm in benchmark engineering optimization problems. *International Journal of Optimization in Civil Engineering*, 1, 495–505.

Kazemzadeh Azad, S., and Kazemzadeh Azad, S. (2011). Optimum Design of Structures Using an Improved Firefly Algorithm. *International Journal of Optimization in Civil Engineering*, 1(2), 327–340.

Kazemzadeh Azad, S., Kazemzadeh Azad, S., and Kulkarni, A.J. (2012). Structural optimization using a mutation based genetic algorithm. *International Journal of Optimization in Civil Engineering*, 2 (1), 81–101.

Kazemzadeh Azad S., Hasançebi O., and Kazemzadeh Azad, S. (2013a). Upper Bound Strategy for Metaheuristic Based Design Optimization of Steel Frames. *Advances in Engineering Software*, 57, 19–32.

Kazemzadeh Azad S., Hasançebi O., Kazemzadeh Azad S., and Erol O.K. (2013b). Upper Bound Strategy in Optimum Design of Truss Structures: A Big Bang-Big Crunch Algorithm Based Application. *Advances in Structural Engineering*, 16(6), 1035–1046.

Kazemzadeh Azad S., and Hasançebi O. (2013). Improving Computational Efficiency of Particle Swarm Optimization for Optimal Structural Design. *International Journal of Optimization in Civil Engineering*, 3, 563–574.

Kazemzadeh Azad S., and Hasańcebi O. (2014). An Elitist Self-Adaptive Step-Size Search for Structural Design Optimization, *Applied Soft Computing*, 19, 226–235.

Kazemzadeh Azad S., Hasańcebi O., and Saka M.P. (2014a). Guided stochastic search technique for discrete sizing optimization of steel trusses: a design-driven heuristic approach. *Computers and Structures*, 134, 62–74.

Kazemzadeh Azad S., Hasańcebi O., and Kazemzadeh Azad S. (2014b). Computationally Efficient Optimum Design of Large Scale Steel Frames. *Advances in Structural Engineering*, under review.

Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. In: *IEEE international conference on neural networks*, IEEE Press, 1942–1948.

Khan, M.R. (1984). Optimality criterion techniques applied to frames having general cross-sectional relationships, *AIAA Journal*, 22(5), 669–676.

Koohestani, K., and Kazemzadeh Azad S. (2009). An Adaptive real-coded genetic algorithm for size and shape optimization of truss structures, In: Topping B.H.V., Tsompanakis Y. (Eds.), *The First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, Civil-Comp Press, Stirlingshire, UK, Paper 13.

Lamberti, L., and Pappalettere, C. (2011a). Metaheuristic design optimization of skeletal structures: a review. *Computational Technology Reviews*, 4, 1–32.

Lamberti, L., and Pappalettere, C. (2011b). A fast big bang-big crunch optimization algorithm for weight minimization of truss structures. In: Tsompanakis Y., Topping B.H.V., (Eds), *Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, Stirlingshire, UK: Civil-Comp Press.

Lee, K.S., and Geem, Z.W. (2004). A new structural optimization method based on the harmony search algorithm. *Computers and Structures*, 82, 781–98.

Li, L.J., Huang, Z.B., Liu, F., Wu, Q.H. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers and Structures*, 85, 340–349.

Li, L.J., Huang, Z.B., Liu, F. (2009). A heuristic particle swarm optimization method for truss structures with discrete variables. *Computers and Structures*, 87, 435–443.

Luh, G.C., and Lin, C.Y. (2011). Optimal design of truss-structures using particle swarm optimization. *Computers and Structures*, 89, 2221–2232.

MATLAB. Version 7.8, The MathWorks Inc., 3 Apple Hill Drive, Natick.

McGuire, W. (1968). *Steel Structures*. Prentice-Hall.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*. 3rd ed., Springer-Verlag.

Park, H.S., and Park, C.L. (1997). Drift control of high-rise buildings with unit load method, *The Structural Design of Tall Buildings*, 6, 23–35.

Patnaik, S.N., Berke, L., Gallagher, R.H. (1991). Integrated force method versus displacement method for finite element analysis. *Computers and Structures*, 38, 377–407.

Patnaik, S.N., Gendy, A.S., Berke, L., Hopkins, D.A. (1998). Modified fully utilized design (MFUD) method for stress and displacement constraints. *International Journal for Numerical Methods in Engineering*, 41, 1171–1194.



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

Prager, W., and Shield, R. T. (1967). A general theory of optimal plastic design. *J. Appl. Mech.* 34(1), 184–186.

Prager, W. (1968). Optimality criteria in structural design. In *Proceedings of National Academy for Science*, 61(3), 794–796.

Rashedi, R., Moses, F. (1986). Application of linear programming to structural system reliability. *Computers and Structures*, 24(3), 375–384.

Ramaswamy, G.S., Eekhout, M., and Suresh, G.R. (2002). *Analysis, design and construction of steel space frames*, Thomas Telford Publishing.

SAP2000. Version 14.1, Computers and Structures Inc., Berkeley, CA.

Saka, M.P. (1984). Optimum design of space trusses with buckling constraints. In *Proceedings of 3rd International Conference on Space Structures*, University of Surrey, Guildford, U.K., September 1984.

Saka, M.P. (2007a). Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review. In Topping B.H.V., (Editor), *Civil Engineering Computations: Tools and Techniques*, Saxe-Coburg Publications, Stirlingshire, UK.

Saka, M.P. (2007b). Optimum geometry design of geodesic domes using harmony search algorithm. *Advances in Structural Engineering*, 10, 595–606.

Saka, M.P. (2009). Optimum design of steel sway frames to BS5950 using harmony search algorithm. *Journal of Constructional Steel Research*, 65(1), 36–43.

Saka, M.P., and Hasańcebi, O. (2009). Adaptive harmony search algorithm for design code optimization of steel structures. In: Z. Geem (Ed.) Harmony search algorithms for structural design optimization. Springer-Verlag, Berlin, Heidelberg, 79–120.

Saka, M.P., and Geem, Z.W. (2013). Mathematical and Metaheuristic Applications in Design Optimization of Steel Frame Structures: An Extensive Review. *Mathematical Problems in Engineering*, vol. 2013, Article ID 271031, doi:10.1155/2013/271031.

Salajegheh, E. (1997). Structural optimization using response approximation and optimality criteria methods. *Engineering Structures*, 19(7), 527–532.

Schwefel, H.-P. (1981). Numerical optimization of computer models. Chichester, UK, John Wiley & Sons.

Soh, C.K., and Yang, J. (1996). Fuzzy controlled genetic algorithm search for shape optimization, *Journal of Computing in Civil Engineering*, 10, 143–150.

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

Sonmez, M. (2011b). Artificial bee colony algorithm for optimization of truss structures. *Applied Soft Computing*, 11, 2406–2418.

Tabak, E.I., and Wright, P.M. (1981). Optimality criteria method for building frames. *Journal of Structural Division*, 107(7), 1327–1342.

Tang, H., Zhou, J., Xue, S., and Xie, L. (2010). Big bang-big crunch optimization for parameter estimation in structural systems. *Mechanical Systems and Signal Processing*, 24, 2888–2897.

Venkayya, V. B., Khot, N. S., and Berke, L. (1973). Application of optimality criteria approaches to automated design of large practical structures. In Proceedings of the 2nd Symposium on Structural Optimization, AGARD-CP-123, 3, 1–19.

Yang, X.-S., (2008). Nature-inspired metaheuristic algorithms, Luniver Press.

Yang, X-S. (2010). A new metaheuristic bat-Inspired algorithm, In: J. R. Gonzalez et al. (Eds.), Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence, Springer Berlin, Springer, 65–74.

## CURRICULUM VITAE

### PERSONAL INFORMATION

Surname, Name: Kazemzadeh Azad, Saeid  
Nationality: Iranian  
Date and Place of Birth: 13 September 1981, Tabriz  
Marital Status: Single  
Phone: +90 537 910 86 22  
email: saeid.azad@metu.edu.tr

### EDUCATION

Degree	Institution	Year of Graduation
MS	University of Tabriz, Structural Engineering	2009
BS	University of Tabriz, Civil Engineering (Ranked 1 <sup>st</sup> )	2004

### FOREIGN LANGUAGES

English, Turkish, Persian, and Azerbaijani

### ORGANIZED EVENT

Organizer and scientific committee member of the International Student Competition in Structural Optimization (ISCSO 2011, 2012 and 2013).

### SELECTED PUBLICATIONS

1. Kazemzadeh Azad S., and Hasançebi O. (2014). An Elitist Self-Adaptive Step-Size Search for Structural Design Optimization, *Applied Soft Computing*, 19, 226–235.
2. Kazemzadeh Azad S., Hasançebi O., and Saka M.P. (2014). Guided stochastic search technique for discrete sizing optimization of steel trusses: a design-driven heuristic approach. *Computers and Structures*, 134, 62–74.

3. Hasaebi, O., and Kazemzadeh Azad, S. (2014). Discrete Size Optimization of Steel Trusses using a Refined Big Bang–Big Crunch Algorithm. *Engineering Optimization*, 46(1), 61–83.
4. Kazemzadeh Azad S., Hasaebi O., and Kazemzadeh Azad, S. (2013). Upper Bound Strategy for Metaheuristic Based Design Optimization of Steel Frames. *Advances in Engineering Software*, 57, 19–32.
5. Kazemzadeh Azad S., Hasaebi O., Kazemzadeh Azad S., and Erol O.K. (2013). Upper Bound Strategy in Optimum Design of Truss Structures: A Big Bang-Big Crunch Algorithm Based Application, *Advances in Structural Engineering*, 16(6), 1035–1046.
6. Hasaebi, O., Kazemzadeh Azad, S., and Kazemzadeh Azad, S. (2013). Automated sizing of truss structures using a computationally improved SOPT algorithm. *International Journal of Optimization in Civil Engineering*, 3(2), 209–221.
7. Hasaebi, O., Kazemzadeh Azad, S. (2012). An exponential big bang-big crunch algorithm for discrete design optimization of steel frames. *Computers and Structures*, 110–111, 167–179.
8. Hasaebi, O., and Kazemzadeh Azad, S. (2012). An efficient metaheuristic algorithm for engineering optimization: SOPT. *International Journal of Optimization in Civil Engineering*, 2(4), 479–87.
9. Kazemzadeh Azad, S., Kazemzadeh Azad, S., and Kulkarni, A.J. (2012). Structural optimization using a mutation based genetic algorithm, *International Journal of Optimization in Civil Engineering*, 2 (1), 81–101.
10. Kazemzadeh Azad, S., Hasaebi, O., and Erol, O.K. (2011). Evaluating efficiency of big bang-big crunch algorithm in benchmark engineering optimization problems. *International Journal of Optimization in Civil Engineering*, 1, 495–505.
11. Kazemzadeh Azad, S., and Kazemzadeh Azad, S. (2011). Optimum Design of Structures Using an Improved Firefly Algorithm. *International Journal of Optimization in Civil Engineering*, 1(2), 327–340.

## **HOBBIES**

Computer Technologies, Movies, Swimming