DEVELOPING A STRUCTURAL OPTIMIZATION SOFTWARE FOR EFFICIENT AND PRACTICAL OPTIMUM DESIGN OF REAL-WORLD STEEL STRUCTURES

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

BY

AYTAÇ KORUCU

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

AUGUST 2022

Approval of the thesis:

DEVELOPING A STRUCTURAL OPTIMIZATION SOFTWARE FOR EFFICIENT AND PRACTICAL OPTIMUM DESIGN OF REAL-WORLD STEEL STRUCTURES

submitted by **AYTAÇ KORUCU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Civil Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Erdem Canbay
Head of the Department, **Civil Engineering**

Prof. Dr. Oğuzhan Hasançebi
Supervisor, **Civil Engineering, METU**

Examining Committee Members:

Prof. Dr. Oğuzhan Hasançebi
Civil Engineering, METU

Prof. Dr. Afşin Sarıtaş
Civil Engineering, METU

Prof. Dr. Eray Baran
Civil Engineering, METU

Prof. Dr. Saeed Gholizadeh
Civil Engineering, Urmia Universitesi

Assoc. Prof. Dr. Ibrahim Aydoğdu
Civil Engineering, Akdeniz Üniversitesi

Date: 10.08.2022

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 : Aytaç Korucu

Signature :

ABSTRACT

DEVELOPING A STRUCTURAL OPTIMIZATION SOFTWARE FOR EFFICIENT AND PRACTICAL OPTIMUM DESIGN OF REAL-WORLD STEEL STRUCTURES

Korucu, Aytaç
Ph. D., Department of Civil Engineering
Supervisor: Prof. Dr. Oğuzhan Hasançebi

August 2022, 163 pages

Structural engineers have to design not only safe and practical buildings in a reasonably short period but also guarantee that the design generated should be cost-efficient. However, the design of real-world structures in a traditional way is an extremely laborious and time-consuming task, and final designs are often uneconomical. In this study, a computationally efficient, design-driven optimization technique (Guided Evolution Strategy Optimization, GES) is developed for the optimal design of real-world steel structures considering the strength, displacement, and geometric constraints in accordance with the conventional design codes.

Structural design engineers demand practical, efficient, and robust software platforms that combine the potential of optimization techniques with design software capabilities. In other words, there is a need for a single master software platform that integrates the optimization software and the design software packages. This way, promising solutions for real-world structures can be obtained practically with reasonable computational power and time. This study also developed a reliable, efficient, flexible, and user-friendly software platform called Structural Optimization Platform Software (SOPS) with the abovementioned capabilities to present a robust infrastructure for designing real-world steel structures. The SOPS's structural analysis and design part (design software component) is carried out externally via the Open Application Programming Interface (OAPI) of the SAP2000 structural analysis program. In addition to GES, various metaheuristic techniques are implemented into the SOPS program package to constitute its optimization software component.

The numerical results demonstrate the computational efficiency of the proposed technique (GES) for the optimal design of real-world steel structures subjected to geometric, strength, and displacement constraints in terms of convergence rate and quality of the optimum solution. Moreover, the application of SOPS to the optimization of real-world steel structures demonstrates that SOPS successfully combines the potential of the optimization techniques and the design software capabilities (SAP2000). Furthermore, due to SOPS's real-time monitoring ability, the performance of the optimization techniques can be easily observed in real-time.

Keywords: Structural Optimization, Optimization Software, Optimization Algorithms, Steel Frame Structure, SAP2000 OAPI

ÖZ

GERÇEK ÇELİK YAPILARIN ETKİN VE PRATİK OPTİMUM TASARIMLARI İÇİN BİR YAPISAL OPTİMİZASYON YAZILIMININ GELİŞTİRİLMESİ

Korucu, Aytaç
Ph. D., Department of Civil Engineering
Supervisor: Prof. Dr. Oğuzhan Hasançebi

Ağustos 2022, 163 sayfa

Yapı mühendisleri binaları kısa bir sürede güvenli ve pratik olarak tasarlamanın yanında, tasarımın maliyetinin de uygun olmasını sağlamalıdır. Bununla birlikte, yapıların geleneksel bir şekilde tasarımı son derece zahmetli ve zaman alıcı bir süreç olmakla birlikte elde edilen tasarımlar genellikle ekonomik olmamaktadır. Bu çalışmada, yapıların dayanım, deplasman kısıtlamalarının yanında, geometrik kısıtlamalarının da göz önünde bulundurulduğu optimal tasarımı için, verimli ve tasarım odaklı optimizasyon tekniği olan GES geliştirilmiştir.

Yapısal tasarım mühendisleri, optimizasyon tekniklerinin potansiyelini tasarım yazılımı yetenekleriyle birleştiren pratik, verimli ve etkin yazılım platformlarına ihtiyaç duymaktadırlar. Başka bir deyişle, optimizasyon yazılımları ile tasarım yazılım paketlerini entegre ederek yapıların optimum tasarımlarının kısa sürede ve etkin kaynak kullanımı ile elde edilmesine olanak sağlayan bir yazılıma ihtiyaç vardır. Bu çalışmada, yapıların optimum tasarımı için, bu bahsedilen özelliklere sahip SOPS adında etkin, pratik ve kullanışlı bir yazılım paketi geliştirilmiştir. SOPS yazılımının, analiz ve tasarım bileşenleri, Açık Uygulama Programlama Arayüzü (OAPI) kullanılarak SAP2000 analiz programıyla entegre edilmiştir. Bu çalışmada geliştirilen GES optimizasyon tekniğinin yanısıra diğer öne çıkan bazı optimizayon teknikleri de SOPS yazılım platformuna entegre edilerek GES tekniğinin performansı diğer tekniklerle kıyaslanmıştır.

Sayısal sonuçlar, geliştirilen optimizasyon tekniğinin (GES), yapıların dayanım, deplasman ve geometrik kısıtlamalarını da göz önüne alan optimum tasarımlarını yakınsama hızı ve çözüm kalitesi açısından, etkin ve hesaplama verimliliği ile yapabildiğini ortaya çıkarmıştır. Sonuçlar aynı zamanda, SOPS'nin optimizasyon tekniklerinin potansiyeli ile tasarım yazılımının (SAP2000) yeteneklerini başarılı bir şekilde biraraya getirdiğini göstermektedir. Bunun yanında, SOPS'un gerçek zamanlı izleme yeteneği sayesinde, optimizasyon tekniklerinin performansı gerçek zamanlı olarak kolayca gözlemlenebilmektedir.

Anahtar Kelimeler: Yapısal Optimizasyon, Optimizasyon Yazılımı, Optimizasyon Algoritmaları, Çelik Yapılar, SAP2000 OAPI

To My Family

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my supervisor Prof. Dr. Oğuzhan Hasançebi, for his patience, guidance, advice, criticism, and continuous encouragement that motivated me at all levels of this research. The completion of this study would not be possible without his endless support and enlightening remarks.

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

I would also like to thank all examining committee members for their invaluable comments and suggestions.

I owe special thanks to my friend, Dr. Hasan Eser, for his valuable support and encouragement during the preparation of the thesis.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xvii

LIST OF ABBREVIATIONS xxiii

1. INTRODUCTION 1
   - 1.1 Structural Optimization 1
   - 1.2 Gradient-Based Approaches and Heuristic Methods 4
   - 1.3 Metaheuristic Search Techniques 5
   - 1.4 Aim and Scope of the Thesis 9

2 OPTIMUM DESIGN PROBLEM FORMULATION 11
   - 2.1 Introduction 11
   - 2.2 Design Variables and Objective Function 11
   - 2.3 Strength Constraints 13
   - 2.4 Deflection and Inter-Story Drift Constraints 14
   - 2.5 Geometric Constraints 15
     - 2.5.1 Girder-to-Column Connections 15
     - 2.5.2 Column-to-Column Connections 17
     - 2.5.3 Beam-to-Girder Connections 20
   - 2.6 Constraint Handling with Penalty Function 20

3 OPTIMIZATION ALGORITHMS 23

3.1 Introduction ... 23
3.2 Big Bang - Big Crunch Optimization Technique (BB-BC) ... 23
    3.2.1 Formulation BB-BC Optimization Algorithm ... 25
3.3 Particle Swarm Optimization Technique (PSO) ... 29
    3.3.1 Formulation of PSO Algorithm ... 31
3.4 Evolution Strategies (ESs) ... 35
    3.4.1 The (μ,λ)- and (μ+λ)-Evolution Strategy ... 37
    3.4.2 The (1+1)-Evolution Strategy ... 43
3.5 The Proposed Guided Evolution Strategy Technique ... 45
    3.5.1 Initialization of the Algorithm ... 46
    3.5.2 Evaluation of the parent individual ... 46
    3.5.3 Mutation ... 47
    3.5.4 Evaluation of the offspring individual ... 52
    3.5.5 Selection ... 52
    3.5.6 Stagnation Control Strategy ... 52
    3.5.7 Termination ... 53
4 STRUCTURAL OPTIMIZATION PLATFORM SOFTWARE (SOPS) ... 55
4.1 Introduction ... 55
4.2 Integration of SOPS with SAP2000 API ... 56
4.3 SOPS Architecture ... 57
    4.3.1 SOPS Input/output Module ... 59
    4.3.2 SOPS Constraints Module ... 61
    4.3.3 SOPS Analysis and Design Module ... 64
    4.3.4 SOPS Optimization Modules ... 65
5 NUMERICAL EXAMPLES ... 67

5.1 Sensitivity Analysis of GES Parameters 68

5.1.1 Problems Used in Sensitivity Analysis of GES Parameters............ 68

5.1.2 Sensitivity Analysis for Guided Mutation Limit (GML) Parameter74

5.1.3 Sensitivity Analysis of Guided Mutation Ratio (GMR) Parameter 92

5.1.4 Discussion of Parameter Sensitivity Analysis Results.................... 97

5.2 Performance Comparison of the GES Algorithm.................................. 99

5.2.1 Optimization Problem 1-A: 160-Member Steel Frame under Strength Constraints... 100

5.2.2 Optimization Problem 1-B: 160-Member Steel Frame Under Strength and Inter-Story Drift Constraints.. 105

5.2.3 Optimization Problem 2: 584-Member Steel Frame..................... 111

5.3 Numerical Examples of Real-World Steel Structures.......................... 117

5.3.1 Optimization Problem 1: 545-Member Steel Frame..................... 118

5.3.2 Optimization Problem 2: 551-Member Steel Frame..................... 129

5.3.3 Optimization Problem 3: 1634-Member Steel Structure 140

6 CONCLUSION..151

6.1 Summary and Concluding Remarks.. 151

REFERENCES..155

CURRICULUM VITAE 163

LIST OF TABLES

TABLES

Table 5.1 Gravity design loads for both 160-member steel frame and 584-member steel frame 73

Table 5.2 Earthquake seismic coefficients for both 160-member steel frame and 584-member steel frame 73

Table 5.3 Constant values of the GML parameter considered for the sensitivity analysis. 76

Table 5.4 Various combinations of the linearly decreasing GML parameter considered for the sensitivity analysis 78

Table 5.5 Various combinations of the exponentially decreasing GML parameter considered for the sensitivity analysis 80

Table 5.6 Parameter setting of the metaheuristic algorithms (EBB-BC, PSO, ES) used in this study. 100

Table 5.7 Optimization statistics for the 160-member steel frame design example (under strength constraints). 102

Table 5.8 The optimum designs obtained for the 160-member steel frame design example (under strength constraints) with different optimization techniques. ... 103

Table 5.9 The member groups' DCR values in the optimum designs of the 160-member steel frame design example (under strength constraints) produced by various optimization techniques. 105

Table 5.10 Optimization statistics for the 160-member steel frame design example (under both strength and inter-story drift constraints). 107

Table 5.11 The optimum designs obtained for the 160-member steel frame design example (under both strength and inter-story drift constraints) with different optimization techniques 107

Table 5.12 The member groups' DCR values in the optimum designs of the 160-member steel frame design example (under both strength and inter-story drift constraints) produced by various optimization techniques 109

Table 5.13 The *x*-direction inter-story drift constraint values in the optimum designs of the 160-member steel frame design example (under both strength and inter-story drift constraints) produced by various optimization techniques........ 110

Table 5.14 The *y*-direction inter-story drift constraint values in the optimum designs of the 160-member steel frame design example (under both strength and inter-story drift constraints) produced by various optimization techniques........ 110

Table 5.15 Optimization statistics for the 584-member steel frame design example. 113

Table 5.16 The optimum designs obtained for the 584-member steel frame design example with different optimization techniques. 113

Table 5.17 The member groups' DCR values in the optimum designs of the 584-member steel frame design example produced by various optimization techniques 115

Table 5.18 The *x*-direction inter-story drift constraint values in the optimum designs of the 584-member steel frame design example produced by various optimization techniques 116

Table 5.19 The *y*-direction inter-story drift constraint values in the optimum designs of the 584-member steel frame design example produced by various optimization techniques 117

Table 5.20 Load combination definitions for the 545-member steel frame design example 125

Table 5.21 Profile Lists for the 545-member steel frame design example 126

Table 5.22 Earthquake seismic coefficients used for calculating earthquake loads acting on the 545-member steel frame. 126

Table 5.23 Optimization statistics for the 545-member steel frame design example. 127

Table 5.24 The initial (original) and optimum designs of the 545-member steel frame design example 127

Table 5.25 The member groups' DCR values in the optimum design of the 545-member steel frame design example. 129

Table 5.26 The inter-story drift constraint values in the optimum design of the 545-member steel frame design example. 129
Table 5.27 Profile Lists for the 551-member steel frame design example.......... 137
Table 5.28 Optimization statistics for the 551-member steel frame design example 137
Table 5.29 The initial (original) and optimum designs of the 551-member steel frame design example 138
Table 5.30 The member groups' DCR values in the optimum design of the 551-member steel frame design example. 139
Table 5.31 Inter-story drift constraint values in the optimum design of the 551-member steel frame design example. 140
Table 5.32 Load combination definitions for the 1634-member steel roof design example.. 145
Table 5.33 Profile Lists for the 1634-member steel roof design example 146
Table 5.34 Profile Lists assigned for member groups for the 1634-member steel roof design example.. 146
Table 5.35 Earthquake seismic coefficients used for calculating earthquake loads acting on the 1634-member steel roof 147
Table 5.36 Optimization statistics for the 1634-member steel roof design example 147
Table 5.37 The initial (original) and optimum designs of the 1634-member steel roof design example.. 148
Table 5.38 The member groups' DCR values in the optimum design of the 1634-member steel roof design example. 149

LIST OF FIGURES

FIGURES

Figure 2.1. Girder-to-Column connections 15
Figure 2.2. Column-to-Column connection (splice connection)........................... 17
Figure 2.3. Column-to-Column connection (cap plate connection)...................... 19
Figure 2.4. Beam to Girder Connection 20
Figure 3.1 An illustration for the preparation of IDL and DDL lists for a design variable, (a) For IDL (b) For DDL... 51
Figure 4.1 Interaction of external application (SOPS) with the SAP2000 API. ... 57
Figure 4.2 Structural Optimization Platform Software (SOPS) architecture 58
Figure 4.3 Interaction with SAP2000 via GUI of SOPS...................................... 59
Figure 4.4 Extracting SAP2000 model parameters into SOPS. 60
Figure 4.5 Section List Form GUI of SOPS 61
Figure 4.6 Constraints Form GUI of SOPS 62
Figure 4.7 Each End of the Member’s Connection Details 63
Figure 4.8 Highlighted Member Connection Details... 63
Figure 4.9 Enforcing Geometric Constraints 63
Figure 4.10 Enforcing Inter-story Drift Constraints 64
Figure 4.11 International design codes implemented in SOPS............................ 64
Figure 4.12 SOPS - Optimization Modules. 65
Figure 5.1 160-member steel frame (the first test example): (a) 3-D view, (b) plan view, (c) side view in x-z plane (d) side view in y-z plane.................................. 70
Figure 5.2 584-member steel frame (the second test example): (a) 3-D view, (b) plan view, (c) side view in x-z plane, (d) side view in y-z plane......................... 71
Figure 5.3 The column member groups in the first two stories of the 584-member steel frame: (a) corner columns, (b) outer y-side columns, (c) outer x-side columns, (d) inner columns.. 72
Figure 5.4 Beam member groups in the first two stories of the 584-member steel frame: (a) inner beams, (b) outer beams. 73

Figure 5.5 Optimization histories for the 160-member steel frame under different constant values of the guided mutation limit (GML) parameter, (a) full-scale view (b) zoomed view 76
Figure 5.6 The variations in the linearly decreasing GML parameters for 16 different combinations considered. 78
Figure 5.7 Optimization histories for the 160-member steel frame under different linearly varying guided mutation limit (GML) schemes: (a) for 2000 analyses, (b) for 5000 analyses 79
Figure 5.8 The variations in the exponentially decreasing GML parameters for 16 different combinations considered. 81
Figure 5.9 Optimization histories for the 160-member steel frame under different exponentially varying guided mutation limit (GML) schemes: (a) for 1000 analyses (b) for 5000 analyses 81
Figure 5.10 Optimization histories for the 160-member steel frame corresponding to the best two performances of the algorithm under three different manipulations of the GML parameter: (a) for 2000 analyses, (b) for 5000 analyses 83
Figure 5.11 Optimization histories for the 160-member steel frame under different exponentially varying guided mutation limit (GML) schemes and initial parent individual generated by using the maximum section: (a) for 2500 analyses (b) for 5000 analyses 85
Figure 5.12 Optimization histories for the 160-member steel frame corresponding to the best two performances of the algorithm under three different manipulations of the GML parameter and initial parent individual that is generated randomly and generated by assigning the largest possible section: (a) for 2000 analyses, (b) for 5000 analyses 86
Figure 5.13 Case-B: Optimization histories for the 160-member steel frame under different exponentially varying guided mutation limit (GML) schemes under a randomly generated initial individual condition: (a) for 1000 analyses (b) for 5000 analyses 88
Figure 5.14 Optimization histories for the 160-member steel frame under different exponentially varying guided mutation limit (GML) schemes and initial parent

individual generated by using the maximum section: (a) for 2000 analyses (b) for 5000 analyses 89

Figure 5.15 Optimization histories for the 160-member steel frame corresponding to the best performances of the algorithm under two initial parent individual conditions (generated randomly and generated by assigning the largest possible section): (a) for 1500 analyses, (b) for 3000 analyses 90

Figure 5.16 Optimization histories for the 584-member steel frame subjected to inter-story drift constraint under different exponentially varying guided mutation limit (GML) schemes: (a) for 2000 analyses (b) for 5000 analyses 92

Figure 5.17 Optimization histories for the 160-member steel frame under exponentially varying (GML_15-5-EXP) guided mutation limit (GML) schemes and five different Guided Mutation Ratio (GMR) values: (a) for 1000 analyses (b) for 5000 analyses.. 94

Figure 5.18 Optimization histories for the 160-member steel frame under exponentially varying (GML_75-3-EXP) guided mutation limit (GML) schemes and four different Guided Mutation Ratio (GMR) values: (a) for 1000 analyses (b) for 5000 analyses.. 94

Figure 5.19 Optimization histories for the 160-member steel frame under exponentially varying (GML_75-3-EXP) guided mutation limit (GML) schemes and four different Guided Mutation Ratio (GMR) values: (a) for 1000 analyses (b) for 5000 analyses.. 96

Figure 5.20 Optimization histories for the 584-member steel frame under exponentially varying (GML_75-3-EXP) guided mutation limit (GML) schemes and four different Guided Mutation Ratio (GMR) values: (a) for 1000 analyses (b) for 5000 analyses.. 97

Figure 5.21 Average convergence curves obtained for the 160-member steel frame design example (under strength constraints) using various optimization techniques: (a) up to 500 analyses, (b) up to 3000 analyses 104

Figure 5.22 Convergence curves obtained for the 160-member steel frame design example (under strength constraints) in the best run of various optimization techniques: (a) up to 1000 analyses, (b) up to 2000 analyses 104

Figure 5.23 Average convergence curves obtained for the 160-member steel frame design example (under both strength and inter-story drift constraints) using various optimization techniques: (a) up to 1000 analyses, (b) up to 3000 analyses 108

Figure 5.24 Convergence curves obtained for the 160-member steel frame design example (under both strength and inter-story drift constraints) in the best run of various optimization techniques: (a) up to 1000 analyses, (b) up to 3000 analyses 109

Figure 5.25 Inter-story drift ratio curves obtained for the 160-member steel frame design example in the best run of various optimization techniques: (a) for x-direction, (b) for y-direction 110

Figure 5.26 Average convergence curves obtained for the 584-member steel frame design example using various optimization techniques: (a) up to 1500 analyses, (b) up to 3000 analyses 114

Figure 5.27 Convergence curves obtained for the 584-member steel frame design example in the best run of various optimization techniques:, (a) up to 1500 analyses, (b) up to 3000 analyses. 115

Figure 5.28 Inter-story drift ratio curves obtained for the 584-member steel frame design example in the best run of various optimization techniques: (a) for x-direction, (b) for y-direction 117

Figure 5.29 545-member steel frame: (a) 3-D view, (b) plan view, (c) side view in x-z plane, (d) side view in y-z plane. 121

Figure 5.30 The first floor girder and beam member groups for the 545-member steel frame: (a) the first floor long-side girders on A and C axes (GR1), (b) the first floor long-side girders on B axis (GR2), (c) the first floor long-side girder on D axis (GR3), (d) the first floor short-side girders between A-C axes (GR4), (e) the first floor short-side beams (BM1), (f) the first floor short-side girders between C-D axes (GR7) 122

Figure 5.31 The roof floor girder and beam member groups for the 545-member steel frame: (a) the roof floor long-side girders and short-side exterior girders

(GR5), (b) the roof floor short-side interior girders (GR6), (c) the roof floor beams (BM2). 123
Figure 5.32 The column member groups for the 545-member steel frame: (a) the columns on axes A and C (CL1), (b) the columns on axis D (CL2). 124
Figure 5.33 Bracing member groups for the 545-member steel frame: (a) horizontal braces (BR1), (b) vertical braces (BR2). 125
Figure 5.34 Average convergence curves obtained for the 545-member steel frame design example using various optimization techniques: (a) up to 500 analyses, (b) up to 1000 analyses 128
Figure 5.35 Convergence curves obtained for the 545-member steel frame design example in the best run of various optimization techniques: (a) up to 500 analyses, (b) up to 1000 analyses 128
Figure 5.36 551-member steel frame: (a) 3-D view, (b) plan view, (c) side view in x-z plane, (d) side view in y-z plane 132
Figure 5.37 The roof floor girder and beam member groups for the 551-member steel frame: (a) roof girders1, (b) roof girders2, (c) roof beams 133
Figure 5.38 The first floor girder and beam member groups for the 551-member steel frame: (a) the first floor girders1, (b) the first floor girders2, (c) the first floor beams1 134
Figure 5.39 the column member groups for the 551-member steel frame: (a) columns1, (b) columns2, (c) columns3 135
Figure 5.40 The brace member groups for the 551-member steel frame: (a) lateral braces1, (b) lateral braces2, (c) vertical braces 136
Figure 5.41 Average convergence curves obtained for the 551-member steel frame design example using various optimization techniques: (a) up to 250 analyses, (b) up to 500 analyses 138
Figure 5.42 Convergence curves obtained for the 551-member steel frame design example in the best run of various optimization techniques: (a) up to 250 analyses, (b) up to 500 analyses 139
Figure 5.43 1634-member steel roof: (a) 3-D view, (b) side view in x-z plane, (c) side view in y-z plane 142

Figure 5.44 The member groups 1-6 for the 1634-member steel roof: (a) group-1, (b) group-2, (c) group-3, (d) group-4, (e) group-5, (f) group-6, (g) group-7, (h) group-8, (i) group-9, (j) group-10, (k) group-11, (l) group-12 143

Figure 5.45 Average convergence curves obtained for the 1634-member steel roof design example using various optimization techniques: (a) up to 500 analyses, (b) up to 1000 analyses 148

Figure 5.46 Convergence curves obtained for the 1634-member steel roof design example in the best run of various optimization techniques: (a) up to 500 analyses, (b) up to 1000 analyses 149

LIST OF ABBREVIATIONS

ABBREVIATIONS	
API	Application Programming Interface
BB-BC	Big Bang-Big Crunch
DDL	Decrease Direction List
EBB-BC	Exponential Big Bang-Big Crunch
ES	Evolution Strategy
GES	Guided Evolution Strategy
GML	Guided Mutation Limit
GMR	Guided Mutation Ratio
GUI	Graphical User Interface
IDL	Increase Direction List
MBB-BC	Modified Big Bang-Big Crunch
PSO	Particle Swarm Optimization
SEP	Stagnation Escape Period
SOPS	Structural Optimization Platform Software
UBS	Upper Bound Strategy

LIST OF SYMBOLS

SYMBOLS

A_i	Cross-sectional area of the section
c_1	Cognitive factor parameter (PSO)
c_2	Social factor parameter (PSO)
C_d	Deflection amplification factor
I_e	The importance factor
L_i	Length of the i^{th}member
M_n	Nominal flexural strength
M_{type}	Mutation type (ES)
M_u	Required flexural strength
N_c	Total number of constraints
N_e	Number of structural members in a group
N_g	Total number of groups in a structure
N_{gen}	Total number of generations (ES)
N_m	Total number of members in a structure
N_p	Total number of particles in the population (BB-BC)
N_s	Total number of stories
N_{sec}	Total number of sections
N_{stag}	Number of stagnation generations
p	Mutation probability parameter (ES)
P_n	Nominal axial strength
P_u	Required axial strength
V_n	Nominal shear strength
V_u	Required shear strength
α_c	Constant used in BB-BC algorithm
δ_j	Deflection of a beam member

Δ_a	The allowable story drift
Δ_j	Drift of the j^{th} story
δ^T	Relative translational displacement of adjacent stories
ψ	Geometric distribution parameter used in ES
γ	Learning rate constant used in ES
w	Inertia weight parameter (PSO)
Δt	Time step value in PSO
β	User-defined constant value (PSO)
λ	Offspring in metaheuristic algorithms
λ_c	Real and positive constant used in exponential distribution
μ	Parents in metaheuristic algorithms
ϕ	Penalized objective function
α_p	Penalty coefficient
α_{SEP}	Predefined ratio of the objective function
ϕ_a	The resistance factors for axial strength
ϕ_b	The resistance factors for flexural strength
ϕ_v	The resistance factors for shear strength
ρ	Unit weight of the steel

CHAPTER 1

1. INTRODUCTION

1.1 Structural Optimization

Structural engineers have to design not only safe and practical buildings in a reasonably short period, but also guarantee that the design generated should be cost-efficient. In this context, the traditional design approach generally results in overdesigned structures, i.e., excessive material usage. Even with the utilization of available engineering software, the design of real-world structures based on a traditional design procedure is usually a laborious and time-consuming task, and the final designs are often uneconomical. Therefore, the use of optimization techniques in structural design is necessary to generate economical designs that yield the minimum weight or cost for a given structure.

On the other hand, the application of optimization tools and methods to the design of real-world structures requires substantial computational power since an optimization procedure requires an iterative process. Fortunately, during the last couple of decades, the power and speed of computers have increased nearly exponentially. This has created an opportunity to incorporate such tools and methods into a standard design process.

Characteristically, an optimization problem has three essential elements: (1) objective function, (2) design variables, and (3) constraints. In structural optimization design problems, the objective function is usually chosen as the weight or cost of a structure. The primary goal of the design process is to minimize this function, i.e., the weight or cost of a structure. It should be noted that for some structural systems there may be a good correlation between the structural weight and cost. However, this statement may not be always true for all kinds of structures. That is to say, a minimum weight design for a structure does not always correspond to a minimum cost design or vice versa. The objective function of an optimization problem is formulated and defined by a set of design variables that are optionally determined by the designer. In fact, design variables refer to a set of parameters that are not assigned to constant values initially, rather they are treated as varying parameters during an optimization process. In general, one can identify different types of design variables in structural optimization problems, such as sizing, shape, and topology design variables. During an optimization process, the design variables cannot be assigned to arbitrary values because the resulting designs must satisfy certain requirements and restrictions which are usually imposed by design codes. These requirements and restrictions are called constraints. In general, constraints can be of two types as behavior constraints and geometric (size) constraints. The behavior constraints are generally determined by a set of strength and serviceability limitations defined in design codes. On the other hand, geometric (size) constraints refer to problem-specific limitations due to construction practices or geometric restrictions that are included in a design process.

Basically, structural optimization can be grouped into the following three categories depending on the type of design variables employed in an optimization process: (i) topology optimization, (ii) shape optimization, and (iii) size optimization. Topology optimization investigates the existence or non-existence of structural members for the best performance of a structure. During a topology optimization, redundant structural members are removed, and sometimes new member connectivities are defined between the nodes. Hence, each member connectivity is associated with a state variable that can assume two values “0” and “1” only during an optimization process. The state variable “1” indicates the existence of the related member, whereas the state variable “0” means its non-existence in the optimum topological design of a structure. In shape optimization, the nodal positions in a structure are considered and defined as design variables, and the aim is to find the best positions of them. Since a change in nodal positions results in modifications in the resulting shape of a structure, this optimization model is referred to as shape optimization.

Finally, in size optimization the cross-sectional areas of structural members are treated as design variables. Sizing design variables can further be classified as continuous and discrete. In continuous-sizing design variables, the cross-sectional areas of structural members can be assigned to any positive arbitrary real value between specified upper and lower bounds. In discrete-sizing design variables, however, cross-sectional areas of structural members can only be selected from a set of predefined values, i.e., sections.

In practice, the topology and shape of structures are usually determined or finalized during an architectural design stage, and therefore are not expected to be modified afterward. Hence, a sizing optimization model is a more common problem type in structural steel design, as compared to the other two types. Moreover, the structural steel design is such that steel members must be assigned from commercially defined steel profile sets. As a result, a structural steel design problem often appears as a discrete sizing optimization problem.

It should be noted that every engineering design has usually more than one acceptable solution. However, finding the best solution which optimizes a chosen objective amongst all these acceptable solutions is not an easy task. For relatively small-scale problems where there exists a limited number of design variables and profile sets, exhaustive search methods can be employed to find the optimum solution. These methods, in principle, search every possible point at a time. However, as the problem scale increases, exhaustive search methods become insufficient because this requires prohibitively long computational time and it is not probable to search the entire candidate design domain in a feasible time. In order to overcome this challenge, various optimization techniques have been proposed and adopted in structural optimization literature. These techniques intelligently search design space to reach the optimum design in an acceptable computational time. In the following sections, the utilization of these techniques in structural design is introduced briefly.

1.2 Gradient-Based Approaches and Heuristic Methods

With the advent and development of computational tools and information technologies, structural optimization has become a popular research subject in the past few decades. The conventional methods used in structural optimization can be roughly categorized into two main divisions: (1) gradient-based methods, and (2) heuristic methods (Aldwaik & Adeli, 2014). The gradient-based approaches, which are used mainly in most of the mathematical programming methods, work on the basis of calculating the first and/or second-order derivatives of the objective function and constraints for determining an advantageous and feasible search direction in the design space. These approaches are also characterized as traditional methods and applied in the early studies in structural optimization. However, in spite of the extensive applications of gradient-based methods, some limitations have been encountered in previous studies.

Firstly, gradient-based methods usually tend to converge to local optima instead of the global optimum. Since structural optimization problems usually have many local optima, the accuracy of the optimum solution located by a gradient-based technique is highly dependent on its starting point. In other words, the performance of the technique depends on how close the starting point is to the global optimum. These techniques can locate the nearest optimum in the neighborhood region of the starting point, and hence they are known as local search techniques. Accordingly, gradient-based methods have difficulties in reaching the global optimum especially for highly complex problems like structural optimization problems. Moreover, due to the requirement of gradient computations, the implementation of gradient-based methods becomes difficult and inefficient for discrete optimization problems in particular (Sigmund, 2011). Therefore, these techniques may not effectively deal with complex, highly nonlinear, and discrete optimization problems.

To circumvent the shortcomings and weaknesses of gradient-based algorithms, heuristic methods, which are based on trial-and-error approaches, are proposed and implemented for structural optimization problems. Despite the fact that heuristic methods offer rather simple and straightforward design methodologies with a high computation performance, they are problem-specific approaches and they also suffer from local optimum traps. Therefore, the ongoing studies in optimization literature have focused on developing generic and robust optimization techniques, which in turn has led to the development of the so-called metaheuristic search techniques. The metaheuristic search techniques have received increasing attention and applied to structural optimization design problems in the last few decades due to (i) their problem-independent nature, (ii) their ability to perform both local and global search, and (iii) their ability to deal with both discrete and continuous design variables, (Yang, 2010).

1.3 Metaheuristic Search Techniques

Structural optimization with metaheuristic search methods has become popular because the use of these methods makes it possible to deal with a variety of practical and complex optimization tasks encountered in the field that cannot be handled formerly by the traditional methods.

Metaheuristics are high-level algorithms such that for a broad range of optimization problems they are capable of finding the optimal solution with limited computation capacity and knowledge about the problem. Besides, the fact that they are problem-independent makes it possible to use them for a variety of problems, including the cases where the search space is large and complex.

As compared to the traditional deterministic and stochastic optimization approaches, metaheuristic search techniques have four main advantages (Saka et al., 2016): (1) they can deal with both discrete and continuous design variables, which in fact makes them more suitable for combinatorial optimization problems as well as discrete structural design optimization problems, (2) they do not need the gradient information for search and implementation, (3) an explicit relation between constraints and the objective function is not needed in metaheuristic search techniques, and (4) they come up with an increased promise for locating the global optimum. Moreover, coding of metaheuristic techniques is rather easy and straightforward compared to traditional structural optimization techniques, and this makes them practical and appropriate design tools for the optimization of structures.

The performance of structural optimization algorithms is fundamentally evaluated according to how fast and how close they reach the optimum solution. The former is regarded as the convergence rate of the algorithm and it is mainly determined by the count of structural analyses implemented to reach the optimum solution. The latter is the accuracy (quality) of the obtained optimum solution and it is assessed by the proximity of the obtained solution to the global optimum.

Despite their advantages stated above, the metaheuristic search algorithms may have some drawbacks due to their random search strategies. They usually investigate a search space randomly by using the strategies inspired by nature, social culture, biology, or laws of physics. Therefore, excessive objective function evaluations are usually required before reaching a global optimum solution. In structural optimization design problems, the evaluations of the objective function regarding the response computations of a structural system generally consume about 85-95% of an optimization task (Hasançebi et al., 2011). Therefore, the need for excessive structural analyses especially for large structures significantly increases the computational time and effort. In order to reduce the computational burden to a manageable size and thus to improve the efficiency of a metaheuristic search technique, mainly two different approaches are followed.

In the first approach, the computational efficiency of a metaheuristic search technique is enhanced by utilizing powerful parallel computing environments. In this approach, the optimization task is distributed amongst several processors located in a computer farm and connected via network devices. This approach has been implemented for structural design optimization design problems in various studies in the literature (Hasançebi et al., 2011; Papadrakakis et al., 2003; Sarma & Adeli, 2001) and it has been reported that the overall computation time required by metaheuristic search algorithms can greatly be reduced by virtue of this approach.

However, extremely laborious and costly hardware environment configuration requirements make this approach somewhat impractical for real-world applications.

The second approach focuses on improving the computational efficiency of a metaheuristic search algorithm by incorporating efficient and intelligent search strategies into the algorithm, resulting in a reduced number of structural analyses required to locate an optimum solution. In other words, the goal of this approach is to obtain a reasonable and acceptable design using fewer structural analyses. In this regard, the information gathered during the structural analysis of a previous design generated in the course of optimization process can be utilized as an effective strategy such that the search process can be guided to more promising and feasible regions of the design space using this information. As a result, an effective search is established through guidance by evaluating the existing solutions and obtaining better solutions in a relatively lesser number of structural analyses. The structural optimization techniques which have implemented this intelligent strategy in their algorithms are referred to as design-driven optimization techniques.

The fully stressed design (FSD) is one of the early optimization methods where this strategy is implemented for structural optimization problems (Razani, 1965; Gallagher, 1973). This method has been successfully used for structural optimization problems where the problem constraints only consist of imposed stress limitations for the members. Accordingly, a search direction is determined by FSD based on the constraint (stress) ratios of the structural members. In case a constraint ratio (stress limit) for a member is well below a specified maximum value, a smaller cross-section is adopted for the member to enable a more effective use of material for sizing the member. If, however, a constraint ratio (stress limit) for a member is well above a specified maximum value, a larger cross-section is assigned to the member to eliminate constraint violation. In other words, when a new (trial) solution is generated, the design variables (cross-sectional area of the members) are assigned to new values regarding the structural analysis of the previous solution so that the demand-to-capacity ratio (DCR) approaches to unity for all the members. Although the FSD has proved to be a very effective method, it is limited to optimization problems with stress or strength constraints only; that is to say, displacement constraints cannot be accounted for by the method. An improvement of the method known as Fully Utilized Design (FUD) has been proposed to handle the displacement constraints as well (Saka, 2003). Nevertheless, despite the fact that the FUD can reach a feasible design solution rapidly, it is shown that this solution in most cases corresponds to an overdesigned structure. Patnaik et al. (1998) proposed a modified (improved) version of the FUD technique, called modified fully utilized design (MFUD), to eliminate the problems associated with the displacement constraints. Therein, the integrated force method (IFM) of structural analysis was integrated into the FUD to identify the significance of a violated displacement constraint and how effectively it was remedied (Patnaik et al., 1991). However, the proposed technique is limited to truss structures only.

In some studies in the literature, the concept of FSD has been combined with the global search abilities of metaheuristic search techniques to create effective hybrid algorithms. For instance, Ahrari & Atai (2013) and Ahrari et al. (2015) have proposed the Fully Stressed Design Evolution Strategy (FSD-ES) that combines the useful features of the FSD and ES optimization techniques to enhance the computational efficiency of the FSD. Later, Ahrari & Deb (2016) proposed an improved version of the FSD-ES, named FSD-ES-II, by introducing displacement constraints into the algorithm. A performance investigation of these hybrid methods (FSD-ES and FSD-ES-II) through some selected benchmark problems of structural optimization has indicated that they exhibit higher efficiencies compared to other techniques. However, it is worth mentioning that these approaches are also limited to truss structures only.

There exist numerous researches that aim to develop an efficient and robust optimization technique for structural design optimization problems. However, up to date, neither of these techniques come up with a promise of finding the global optimum solution, especially for large-scale structures. Moreover, despite a large number of studies and papers that report successful applications of modern metaheuristic search techniques in structural design optimization problems, there is still a gap in the literature as far as the application of these methods to real-world structures is concerned. One reason for this is that high computational time requirements of modern metaheuristic search techniques make practicing engineers reluctant to use them in real-world applications. Indeed, a literature review points out that the metaheuristic search techniques are usually applied to some truss and frame-type structures under a limited number of load combinations. Besides, the full requirements of the design procedure, such as geometric constraints between the members framed into the same joints, are fully or partially omitted in these applications to simplify the search process. Hence, the structural optimization literature entails the development of new and powerful optimization techniques with practical applications of them to real-world problems.

In addition, to accelerate the use of optimization concepts and methods in practical engineering applications, the development of software platforms is necessary where a standard steel design process can be integrated and supported with the potential of structural optimization algorithms. In other words, there is a need for both an efficient optimization technique and a single master software platform that integrate optimization computing software and the design software packages such that promising solutions for real-world structures can be obtained within a reasonable computational time.

1.4 Aim and Scope of the Thesis

The two main objectives of the present study can be outlined as follows: (i) to develop an efficient, fast, and robust design-driven structural optimization algorithm that can easily be used for the design of real-world steel structures in practice, (ii) to develop a reliable, flexible and user-friendly software platform that integrates the optimization techniques with the design software capabilities to present an infrastructure for the design of real-world steel structures.

The chapters of the thesis are organized as follows. Chapter 2 provides a mathematical formulation for discrete sizing optimization of steel frame structures under strength, displacement, and geometric constraints. In Chapter 3, the metaheuristic search algorithms that are implemented in this study are presented and their search algorithms are explained in detail. Besides, a newly developed design-driven hybrid optimization technique called the Guided Evolution Strategy (GES) technique is introduced and its fundamental working principles are explained in detail. In Chapter 4, the Structural Optimization Platform Software (SOPS) developed within the context of this thesis is introduced and its user interface is explained in depth. In Chapter 5, numerical experiments are conducted first to determine the optimum parameter settings of the GES algorithm. Then, several design instances of ordinary moment-resisting steel frames are studied using the GES as well as other metaheuristic search algorithms. This way, the performance of the proposed GES algorithm is identified and quantified in terms of the quality of the optimum solution obtained and the computational efficiency of the algorithm in comparison to those of the other metaheuristic search techniques. Finally, in Chapter 6, the conclusions are presented with a particular emphasis on important findings of the study, and recommendations for future works are articulated.

CHAPTER 2

OPTIMUM DESIGN PROBLEM FORMULATION

2.1 Introduction

The size optimization of a steel frame can be defined as finding the minimum weight of the frame subject to strength and serviceability requirements imposed according to a chosen code of practice as well as geometric constraints between the members framing into each other at the same joints. It should be noted that in practical applications of steel frames, members must be selected from a set of commercially available steel sections only. Therefore, a section pool that consists of a selected number of available steel sections is prepared prior to the initialization of the optimization process. The steel sections are sorted in increasing order of a chosen sectional property; typically, the cross-sectional area. Each steel section in the pool is identified and referenced with a distinct sequence number that varies between 1 and the total number of sections in line with the sorted order of the sections in the pool. During the optimization process, the selection and sizing of member groups are carried out in connection with the sequence numbers. That is to say, when a selection is made for a member group, the cross-sectional properties for the selected section become available from the section pool through the sequence number. In the following section, the relevant formulations are presented in detail for discrete sizing optimization of the steel moment frames.

2.2 Design Variables and Objective Function

The first step in an optimization problem is to identify the design variables. In discrete sizing optimization problems of steel frames, the design variables refer to the cross-sectional areas of steel members. However, as explained in the previous section, the steel members must be selected from a predefined list of ready sections available in the market, i.e., section pool. Hence, the selection and sizing of steel members are carried out in connection with the sequence numbers of the ready sections collected in the section pool. It follows that the design vector **I** in a discrete sizing optimization problem keeps the sequence numbers of the sections assigned to N_m steel members (N_m is the total number of members) from the section pool and is formulated in Equation (2.1):

$$\mathbf{I} = \left[I_1, I_2, \dots, I_{N_m}\right]^T \tag{2.1}$$

Once a selection is made for all members, the corresponding cross-sectional properties for the selected sections become available from the section pool through the sequence numbers. Therefore, the cross-sectional areas (**A**) of the sections assigned to N_m steel members are defined as formulated in Equation (2.2):

$$\mathbf{A} = [A_1, A_2, \dots, A_m]^T \tag{2.2}$$

The objective function is defined as minimizing the weight of the frame, which can be formulated as follows:

$$W = \sum_{i=1}^{N_m} \rho_i A_i \, L_i \tag{2.3}$$

where, ρ_i, A_i, and L_i are the unit weight of the steel, the cross-sectional area of the section assigned to the member i, and the length of the member i, respectively.

Although Equation (2.3) is formulated in a way such that the steel members adopt sections independently, this is not the case in reality. In practice problems, due to architectural, fabrication or constructional requirements, some members of the structure are grouped to have the same section. This is also required to reduce the cost of the construction as the design and fabrication costs increase considerably as the number of different sections used in a structure increase. It should also be stressed that from an optimization point of view, a reduction in the number of design variables through a grouping of the members is advantageous since finding the optimum design becomes an easier task when less number of design variables are considered for a steel frame. Considering the grouping of members in a steel frame, the objective function given in Equation (2.3) can be modified and reformulated as in Equation (2.4)

$$W = \sum_{i=1}^{N_g} \rho_i A_i \sum_{m=1}^{N_e} L_m \tag{2.4}$$

where N_g is the total number of groups in a structure (i.e., the number of independent design variables), and N_e is the number of structural members in a group.

The steel structures must be designed according to a chosen code of practice. In this study, the steel frames are designed in accordance with the provisions of the LRFD-AISC 360-10 (2010) specification, and thus the strength and displacement constraints are imposed accordingly, which are discussed in the following sections.

2.3 Strength Constraints

For steel members that are subjected to bending and/or axial compressive loads, the following strength requirements are imposed:

$$\left(\frac{P_u}{\phi_a P_n}\right)_j + \frac{8}{9}\left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right)_j - 1.0 \le 0 \quad for \left(\frac{P_u}{\phi_a P_n}\right)_j \ge 0.2 \tag{2.5}$$

$$\left(\frac{P_u}{2\phi_a P_n}\right)_j + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right)_j - 1.0 \le 0 \quad for \left(\frac{P_r}{\phi_a P_c}\right)_j < 0.2 \tag{2.6}$$

For steel members that are subjected to shear, the following strength requirement is imposed:

$$\left(\frac{V_u}{\phi_v V_n}\right)_j - 1 \le 0 \tag{2.7}$$

In Equations (2.5-2.7), P_u, M_u, and V_u are the required axial, flexural, and shear strengths calculated using the load combinations given in the ASCE/SEI 7-10 (2010) design load specification, respectively; P_n, M_n, and V_n are the nominal axial, flexural, and shear strengths calculated according to the formulations given in the LRFD-AISC 360-10 (2010) specification; ϕ_a, ϕ_b and ϕ_v are the resistance factors for axial, flexural and shear strengths, respectively and they are all set to 0.9; the subscripts x and y represent the strong and the weak axis of bending for member j.

2.4 Deflection and Inter-Story Drift Constraints

In addition to the strength requirements, the deflection criteria are also considered in the design process. The traditional $1/360$ deflection limit in practice is adopted as a deflection design constraint for the beams subjected to reduced live loads, as formulated in Equation (2.8).

$$\delta_j - \frac{L_j}{360} \leq 0 \tag{2.8}$$

where δ_j represents the calculated deflection of a beam member j under reduced live loads and L_j is the length of the beam member j.

An inter-story drift (Δ) in a typical frame-type steel structure is defined as the relative translational displacement difference between two consecutive floors in the frame. According to ASCE/SEI 7-10 (2010) design load specification, inter-story drift constraint resulting from a seismic action is formulated in Equation (2.9).

$$\frac{\Delta_j}{\Delta_a} - 1 \leq 0 \quad where \quad \Delta_j = \frac{C_d \delta^T}{I_e} \tag{2.9}$$

In Equation (2.9), C_d, I_e, Δ_a, h are the deflection amplification factor, the importance factor, the allowable story drift, and story height, respectively; δ^T is the maximum value of the relative translational displacement of adjacent stories determined by the results of an elastic analysis. The subscript $j = 1,2,\ldots,N_s$ represents the j^{th} story, where N_s is the total number of stories and $\Delta_j$is the drift of the j^{th} story.

2.5 Geometric Constraints

Geometric constraints play an important role in the design of steel structures because any two steel sections cannot be selected arbitrarily and connected together even though they both satisfy strength and displacement constraints. Geometric constraints refer to a set of geometric requirements that must be satisfied by the members that frame into the same joint. Unless these requirements are satisfied, solid connections cannot be produced, leading to impractical and/or unreliable designs. Hence, it is important that geometric constraints are all satisfied from the point of view of the practicality of the optimum design produced. In the following sub-sections, the geometric constraints between various structural members in steel frames are mentioned and formulated.

2.5.1 Girder-to-Column Connections

Girders and columns connected together must satisfy certain geometric constraints depending on whether the girders are connected to the flange or web plate of the columns.

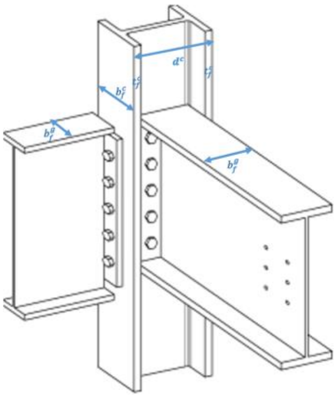


Figure 2.1. Girder-to-Column connections

2.5.1.1 Girder-to-Column flange connections

For a girder connected to the flange of a column member (Figure 2.1), it is required that the flange width of the girder does not exceed the flange width of the column. Accordingly, the associated geometric constraint is defined in Equation (2.10)

$$\frac{b_f^g}{b_f^c} - 1.0 \leq 0 \tag{2.10}$$

where b_f^g is the flange width of the girder, and b_f^c is the flange width of the column.

2.5.1.2 Girder-to-Column web connections

For a girder connected to the web of a column member (Figure 2.1), it is required that the flange width of the girder ideally does not exceed the clear web height of the column. Accordingly, the associated geometric constraint is defined in Equation (2.11)

$$\frac{b_f^g}{(d^c - 2t_f^c)} - 1.0 \leq 0 \tag{2.11}$$

where b_f^g is the flange width of the girder, d^c is the total depth of the column, t_f^c is the flange thickness of the column, and $d^c - 2t_f^c$ is the clear web height of the column.

However, for a girder whose flange width is larger than the clear web height of the connected column, the girder might also be notched/coped by its edges to fit into the column flange at the expense of increased manufacturing cost. In such a case, it is still required that the flange width of the girder does not exceed the total depth of the column. Accordingly, the associated geometric constraint is defined in Equation (2.12)

$$\frac{b_f^g}{d^c} - 1.0 \leq 0 \tag{2.12}$$

where b_f^g is the flange width of the girder, and d^c is the total depth of the column.

2.5.2 Column-to-Column Connections

The column members in multi-story steel frames can be connected together using steel-plated bolted connections called column splices, as depicted in Figure 2.2. For each splice connection, five geometric constraints can be defined as follows.

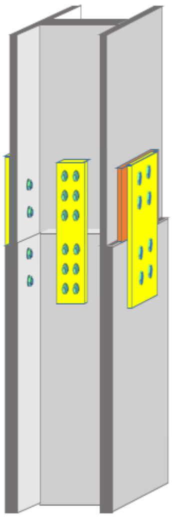


Figure 2.2. Column-to-Column connection (splice connection)

First, it is required that the depth of the upper column does not exceed the depth of the lower column. Accordingly, the associated geometric constraint is defined in Equation (2.13)

$$\frac{d^{uc}}{d^{lc}} - 1.0 \leq 0 \tag{2.13}$$

where d^{uc} is the depth of the upper column and d^{lc} is the depth of the lower column.

Second, it is required that the flange thickness of the upper column does not exceed the flange thickness of the lower column. Accordingly, the associated geometric constraint is defined in Equation (2.14)

$$\frac{t_f^{uc}}{t_f^{lc}} - 1.0 \leq 0 \tag{2.14}$$

where t_f^{uc} is the flange thickness of the upper column, t_f^{lc} is flange thickness of the lower column.

Third, it is required that the web thickness of the upper column does not exceed the web thickness of the lower column. Accordingly, the associated geometric constraint is defined in Equation (2.15)

$$\frac{t_w^{uc}}{t_w^{lc}} - 1.0 \leq 0 \tag{2.15}$$

where t_w^{uc} is the web thickness of the upper column, t_w^{lc} is the web thickness of the lower column.

Fourth, it is required that the flange width of the upper column does not exceed the flange width of the lower column. Accordingly, the associated geometric constraint is defined in Equation (2.16)

$$\frac{b_f^{uc}}{b_f^{lc}} - 1.0 \leq 0 \tag{2.16}$$

where b_f^{uc} is the flange width of the upper column, b_f^{lc} is the flange width of the lower column.

Fifth, it is also required that the clear height of the upper column is not smaller than the clear height of the lower column. Accordingly, the associated geometric constraint is defined in Equation (2.17)

$$\frac{d^{lc} - 2.t_f^{lc}}{d^{uc} - 2.t_f^{uc}} - 1.0 \leq 0 \tag{2.17}$$

where t_f^{uc} is the flange thickness of the upper column, t_f^{lc} is the flange thickness of the lower column, d^{lc} is the depth of the lower column and d^{uc} is the depth of the upper column. It should be noted that this requirement together with Equations (2.13) and (2.16) ensures that the flange of the upper column rests on the flange of the lower column.

Another way of connecting column members in multi-story steel frames is through the use of cap plate connections, in which the end plates welded to the columns are connected to each other by bolts, as depicted inFigure 2.3.

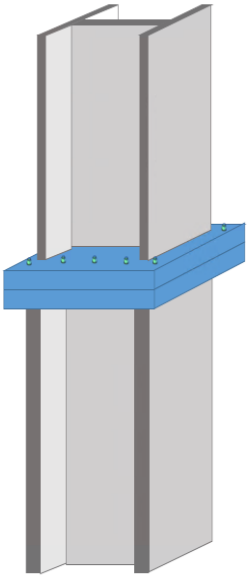


Figure 2.3. Column-to-Column connection (cap plate connection)

It is important to underline that for each cap plate connection, the constraints regarding Equations (2.13) and (2.16) are imposed only.

2.5.3 Beam-to-Girder Connections

Beams connected to girders also need to satisfy certain geometric constraints. For a beam connected to the web of a girder (Figure 2.4), it is required that the depth of the beam does not exceed the clear web height of the girder. Accordingly, the associated geometric constraint is defined in Equation (2.18)

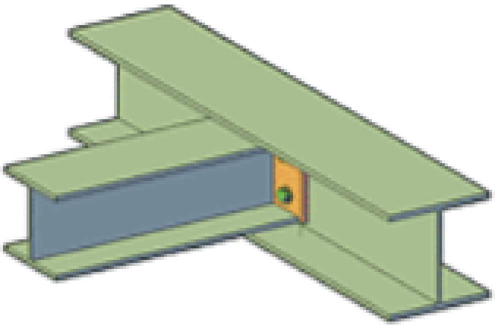


Figure 2.4. Beam to Girder Connection

$$\frac{d^b}{(d^g - 2t_f^g)} - 1.0 \leq 0 \tag{2.18}$$

where d^b is the depth of the beam, d^g is the depth of the girder and t_f^g is the flange width of the girder.

2.6 Constraint Handling with Penalty Function

Metaheuristic search techniques are unconstrained optimization methods. Hence, during the past few decades, various strategies have been proposed for handling constraints with metaheuristic search techniques. One of the most common approaches implemented is to transform a constrained structural optimization problem into an unconstrained one so that unconstrained structural optimization techniques can be employed. Such a transformation is performed by introducing a penalty function and integrating it into the objective function. The penalty function penalizes solutions that violate problem constraints. For minimization problems, the violated constraints lead to an increase in the objective function of that particular solution. The constraint integrated penalty function is referred to as penalized (constrained) objective function. Equation (2.19) presents the constrained objective function used in this study:

$$\phi = W\left[1 + \alpha_p(\sum_{i=1}^{N_c} c_i)\right] \tag{2.19}$$

where ϕ is the penalized objective function, W is the original objective function (i.e., the total weight of the structure), N_c is the total number of constraints, c_i is the value of violation for the i^{th} constraint (if any), and finally the parameter α_p is the penalty coefficient used to to adjust the scale of penalization as a whole and is set to unity in this study.

CHAPTER 3

OPTIMIZATION ALGORITHMS

3.1 Introduction

In this chapter, the metaheuristic search algorithms that are implemented in this study are presented and their search algorithms are explained in detail. These techniques refer to Big Bang-Big Crunch (BB-BC), Particle Swarm Optimization (PSO), and Evolution Strategies (ES) methods. Besides, a newly developed design-driven hybrid optimization technique called the Guided Evolution Strategy (GES) technique is introduced and its fundamental working principles are explained in detail.

3.2 Big Bang - Big Crunch Optimization Technique (BB-BC)

Big Bang-Big Crunch (BB-BC) optimization method introduced by Erol and Eksin (2006) was inspired by the evolution of the universe theory, i.e., the big bang-big crunch theory. According to this theory, the origin of the universe is very rapidly spread out after a big bang, which is followed by shrinkage towards the center of mass of the particles, called big crunch. Similarly, in the BB-BC optimization method, an initial population is generated and spread through the design space randomly identical to the big bang theory, and the population is contracted to the center of mass following the big crunch theory, and this process continues during the optimization process.

The BB-BC optimization method has been successfully applied in various disciplines of optimization, including structural optimization design problems. One of the first applications of the BB-BC method in the area of structural optimization was presented by Camp (2007). Therein the original BB-BC algorithm proposed by Erol and Eksin (2006) was somewhat modified by adding a controlling parameter to adjust the search direction from the center of mass to the position of the global best particle in the population. Using this variant of the BB-BC algorithm, Kaveh and Sabzi (2011) and Kaveh and Abbasgholiha (2011) tested and demonstrated the effectiveness of the method for the optimum design of reinforced concrete structures and planar steel frames, respectively. On the other hand, the original form of the method was used by Kazemzadeh Azad et al. (2011) for solving some benchmark engineering optimization problems and it was demonstrated that the BB-BC was an effective algorithm for complex structural design problems.

Kaveh and Talatahari (2009a) attempted to enhance the performance of the BB-BC method by combining it with Particle Swarm Optimization (PSO) algorithm, resulting in a hybrid method called the HBB-BC method. In the HBB-BC method, the search direction is not only determined by the center of mass but also by the best position of each particle and the global best particle. Kaveh and Talatahari (2010a, 2010b) also applied this hybrid method to structural optimization problems with discrete design variables and achieved satisfactory results.

It should be noted that the original BB-BC algorithm uses a normal distribution for generating new particles (candidate solutions) around the center of mass during the big bang phase. In Hasançebi and Kazemzadeh Azad (2012, 2014) the effect of the implemented statistical distribution on the performance of the BB-BC algorithm was investigated. In their studies, firstly, a normal distribution was followed in line with the original formulation of the BB-BC algorithm; yet, the third power of a normally distributed random number was taken in the relevant formulation, instead of the first power. This new variant of the BB-BC method was named the modified big bang – big crunch method (MBB-BC) method, and they demonstrated that the MBB-BC outperformed the original BB-BC to a large extent in sizing optimum design problems of steel trusses. Secondly, the normal distribution was replaced with an exponential distribution, yet again the third power of the random number sampled according to an exponential distribution was considered in the relevant formulation. This version of the BB-BC method was referred to as exponential big bang – big crunch (EBB-BC), and the numerical experiments performed in the optimum design of steel frames indicated that the EBB-BC outperforms both the original BB-BC method and its variant MBB-BC method.

Kazemzadeh Azad et al. (2013) proposed an Upper Bound Strategy (UBS) which aims to reduce the number of structural analyses required during an optimization process. The principle of UBS lies in the idea of eliminating unnecessary structural analyses for candidate designs that have no chance to improve the best (elite) design. The UBS is very suitable for the BB-BC algorithm and any of its variants because the algorithm implements an elitism rule for design transition, in which the best design is only replaced by an improving candidate solution. This strategy is implemented such that whenever a candidate solution is generated, the structural weight of the resulting design is calculated first. The structural analysis of the candidate design is implemented only if its structural weight is lower than the objective function (penalized weight, ϕ) of the best design; otherwise, the candidate design is automatically eliminated without performing any structural analysis as it has no chance to improve the best design anyway.

In this study, the EBB-BC variant of the method, which was proposed by Hasançebi and Kazemzadeh Azad (2012), is utilized with the UBS approach. In the following sub-sections, a description and related formulations of both the standard BB-BC and EBB-BC algorithms are given briefly.

3.2.1 Formulation BB-BC Optimization Algorithm

The Big Bang-Big Crunch optimization algorithm consists of two successive stages called the big bang and big crunch phases. In the Big Bang phase, the particles are scattered to the search space from a single point called the center of mass. In the Big Crunch phase, the particles shrank to a new center of mass point determined according to the positions of the particles scattered during the big bang phase. The particles represent the candidate solutions while the design variables in a candidate solution correspond to coordinates in the search space. In other words, the position of a particle represents a candidate solution. Similarly, the mass of a particle can be considered as the fitness value of the candidate solution. The main implementation steps of the algorithm can be outlined as follows.

Generate Initial Population: An initial population is generated by randomly spreading (initial Big Bang) the particles (candidate solutions) through the design search space uniformly.

Evaluate the particles: The feasible candidate solutions, which satisfy all the problem constraints without any constraint violations, are evaluated such that their objective function values are directly calculated using Equation (2.4). On the other hand, infeasible candidate solutions, which violate at least one of the problem constraints, are penalized by using a penalty function and their objective function values are determined in accordance with Equation (3.19). In the following the symbol W_p will commonly be used to represent the objective function of a feasible particle and the penalized objective function value of an infeasible particle. Since structural size optimization is a minimization problem, the mass (i.e., fitness value) of a particle is inversely related to its objective function value, and therefore it can be defined using Equation (3.1).

$$mass(m) = \frac{1}{W_p} \tag{3.1}$$

Determine "Center of Mass": This part is called the big crunch phase where the center of mass is calculated by using the weighted mean of the coordinates (design variables) and their corresponding mass values using Equation (3.2)

$$X_i^c = \frac{\sum_{j=1}^{N_p} \frac{1}{W_p} . X_i^j}{\sum_{j=1}^{N_p} \frac{1}{W_p}} \tag{3.2}$$

where X_i^c is the *i*-th component of the center of mass, X_i^j is the *i*-th component (design variable) of the j-th particle, and N_p is the population size, which is the total number of particles in the population.

Generate new solutions: Big Bang is implemented next where new solutions are generated around the center of mass ($\mathbf{X^c}$) using a normal distribution in all components of a particle using Equation (3.3)

$$X_i^{new} = X_i^c + \alpha_c \cdot r_i \frac{\left(X_i^{max} - X_i^{min}\right)}{k} \tag{3.3}$$

where X_i^{max} and X_i^{min} are the specified upper and lower values for the i-th component (design variable) of a particle, respectively, α_c is a constant, k is the iteration number, and r_i is a random number generated according to a standard normal distribution $N(0{,}1)$ with a mean (μ) = 0 and standard deviation (σ) =1.

It should be noted that Equation (3.3) is originally developed and used for continuous variable optimization problems, as it returns a real value. A discrete adaptation of this function is given in Equation (3.4)

$$I_i^{new} = I_i^c + round\left[\alpha_c \cdot r_i \frac{\left(I_i^{max} - I_i^{min}\right)}{k}\right] \tag{3.4}$$

where I_i^c is the i-th the center of mass for a discrete problem, I_i^{max} and I_i^{min} are the specified upper and lower values for the i-th discrete component (design variable) of a particle, respectively. α_c is a constant, k is the iteration number, r_i is a random number generated according to a standard normal distribution $N(0{,}1)$ with a mean (μ) = 0 and standard deviation (σ) =1, and *round* is a function that returns a real number to the nearest integer value.

It should be noted that Equations (3.3) and (3.4) correspond to the formulations of the big bang phase in the original BB-BC algorithm, as developed by Erol and Eksin (2006). In Hasançebi and Kazemzadeh Azad (2012, 2014), certain drawbacks of these formulations are pointed out in the context of discrete sizing optimization of steel trusses and frames, and their reformulations are proposed to improve the efficiency of the algorithm to great extent.

Amongst the two reformulations raised by the related researchers, the exponential big bang – big crunch (EBB-BC) variant proposed by Hasançebi and Kazemzadeh

Azad (2012) is adopted and employed in the present study. In this reformulation, Equation (3.4) is modified such that the use of an exponential distribution is favoured in conjunction with the third power of random number as formulated in Equation (3.5):

$$I_i^{new} = I_i^c \pm round \left[\alpha_c \cdot e_i^3 \frac{\left(I_i^{max} - I_i^{min}\right)}{k} \right] \tag{3.5}$$

where e_i is a random number generated according to an exponential distribution.

An exponential distribution has the probability density function, which is given in Equation (3.6)

$$f(x) = \begin{cases} \lambda_c \cdot e^{-\lambda_c x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{3.6}$$

where λ_c is a real, positive constant. The mean and variance of the exponential distribution are equal to $1/\lambda_c$ and $1/{\lambda_c}^2$, respectively. For various values of λ_c, the shape of an exponential distribution is plotted in Figure 3.1 In the present study, a standard exponential distribution is used by setting λ_c to one. It is important to note that unlike a normal distribution that samples both positive and negative real numbers, an exponential distribution only generates positive numbers. Hence, the rounded term on the right-hand side of Equation (3.5) 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.

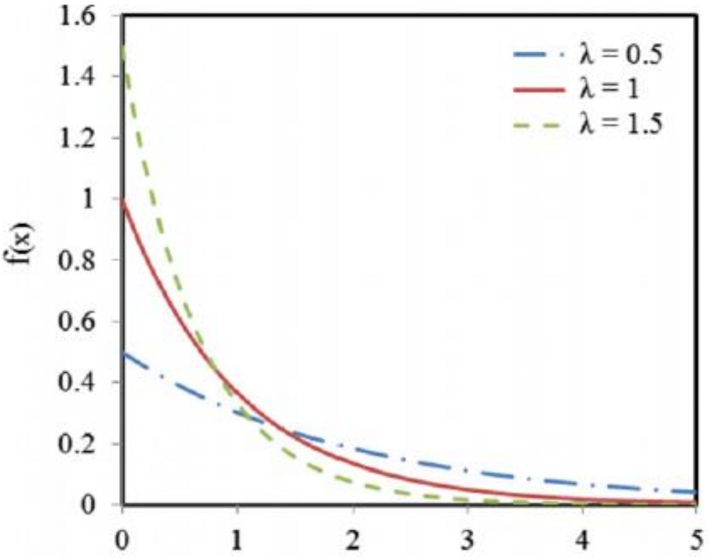


Figure 3.1. Exponential distribution for various values of λ_c (Hasançebi and Kazemzadeh Azad, 2012)

3.3 Particle Swarm Optimization Technique (PSO)

The particle swarm optimization (PSO) algorithm, which was introduced by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995), is a stochastic optimization technique based on the simulation of collective behavior of species like swarm, herd, and flock. These species (i.e., swarms) act collectively to search the food source and share knowledge and experiences among individuals (i.e., particles). This social behavior of swarms is the main inspiration source for the PSO algorithm. The accumulation and sharing of past experiences provide an evolutionary advance in the search pattern of the swarm.

Since the implementation of the method is simple and straightforward and does not involve any problem-specific data to handle the constraints, the PSO algorithm has been applied to various optimization problems, including structural optimization problems. The first applications of the PSO algorithm in the field of structural optimization were carried out by Fourie & Groenwold (2000, 2002) for shape optimization and by Fourie & Groenwold (2001) for topology optimization, and encouraging results were reported.

In Schutte and Groenwold (2003), the original PSO algorithm was modified and used for achieving an optimum sizing design of truss-type structures design. In Camp et al. (2004) the PSO algorithm was implemented in a multi-phase approach such that at the end of each phase the design search space is reduced somewhat and a new search was initiated in a reduced design space located around the global best solution obtained in the previous phase. Bochenek and Forys (2006) introduced a new term to the particle velocity update equation in the algorithm and analyzed the post-buckling behavior of members using the PSO method.

In Perez and Behdinan (2007), the influence of the parameters as well as the functionality of the PSO algorithm was investigated in conjunction with three benchmark problems of structural optimization. Li et al. (2007, 2009) developed modified variants of the technique named heuristic PSO algorithms, which were implemented to optimize truss structures with discrete design. A discrete sizing optimization of truss structures using the PSO algorithm was also studied in Kaveh and Talatahari (2009b), where a hybrid version of the algorithm was introduced by integrating it with other optimization algorithms; namely ant colony optimization (ACO) and harmony search method (HS), to improve its local search and constrained handling performance. Later, in Kaveh and Talatahari (2009c) these hybrid variants of the algorithm were applied to the minimum weight problems of steel frames with discrete design variables.

In Seyedpoor et al. (2010), simultaneous perturbation stochastic approximation (SPSA) and particle swarm optimization (PSO)" methods were hybridized to create an effective algorithm. In this algorithm, the SPSA was utilized only in the first phase, and the outcomes of this phase were used to form an initial swarm for the PSO in the next phase. This way, they reduced the number of structural analyses required and increased the possibility of achieving better solutions.

Dimou and Koumousis (2009) applied PSO to the reliability-based design of truss structures. Gomes (2011) utilized the PSO algorithm in structural size and shape optimization of truss structures under frequency constraints, and they quantified the efficiency of the method using four benchmark structural optimization problems. Plevris and Papadrakakis (2011) combined the PSO algorithm with a gradient-based quasi-Newton sequential quadratic programming method, in which the latter was utilized to accelerate the local search capability of PSO toward the global optimum.

3.3.1 Formulation of PSO Algorithm

Every particle in the swarm has a location (position) in a search space and is represented by $X(i) = x_{i1}, x_{i2}, \ldots, x_{iD}$. The current location of each particle in the swarm is updated with the help of the velocity vector. The velocity of a particle is given as $V(i) = v_{i1}, v_{i2}, \ldots, v_{iD}$.

Therefore, the location-position x of a particle *i* at iteration *k*+1 is updated using Equation (3.7)

$$x_{k+1}^{i} = x_{k}^{i} + v_{k+1}^{i} \Delta t \tag{3.7}$$

where v_{k+1}^{i} is the updated velocity vector of particle *i* at iteration *k*+1, and Δt is the time step value usually taken as unity (Shi and Eberhart, 1998a). The velocity vector is updated by using the experience-memory of each particle and entire swarm’s past experiences (Eberhart and Kennedy, 1995). The velocity vector of each particle is formulated as:

$$v_{k+1}^{i} = w v_{k}^{i} + c_1 r_1 \frac{\left(p_{k}^{i} - x_{k}^{i}\right)}{\Delta t} + c_2 r_2 \frac{\left(p_{k}^{g} - x_{k}^{i}\right)}{\Delta t} \tag{3.8}$$

where v_{k}^{i} is the velocity vector of particle *i* at iteration *k*, p_{k}^{i} is the best position of particle *i* obtained so far, the p_{k}^{g} is the global best position of the whole swarm up to current iteration k, and r_1 and r_2 are random numbers sampled anew according to a uniform distribution in the interval [0,1]

The terms c_1 and c_2 in Equation (3.8) are positive constants which are used to adjust the contribution of the cognitive (the best design solution found by each particle individually) and social (the best design solution found by the whole swarm) components, respectively.

The term w in Equation (3.8) is referred to as the inertia weight, which is introduced to control the exploration capabilities of the swarm. Inertia weight adjusts the current velocity value as well as the updated velocity vector. High values of the w parameter result in higher velocity updates, providing the algorithm an opportunity to explore the design search space globally. On the other hand, low values of this parameter lead to localized search in a near neighborhood within the design space.

The PSO algorithm employs an iterative process, the implementation steps of which can be outlined as follows:

1. Form the initial particle positions x_0^i and velocities v_0^i by distributing them randomly all over the design space.
2. Calculate the objective function values $f\left(x_k^i\right)$ using the design search space locations-positions x_k^i for each particle in the swarm.
3. Keep the fittest particle position (which is optimum) p_k^i at current iteration k and also keep the elitist particle obtained so far as a global optimum particle location/position p_g^i.
4. Modify the location/position of each particle using its previous location-position and updated velocity vector as specified in Equation (3.7) and Equation (3.8).
5. Repeat steps 2-4 until a stopping criterion (generally taken as a number of iterations reached) is accomplished.

The PSO algorithm investigates the search space by updating the velocity term. For each particle, a search direction is determined in consideration of the three factors:

(1) previous iteration velocity, (2) the positions of the individual's previous best location/position, and (3) the best location/position ever found globally by any particle in the swarm. On the other hand, to what extent each of these factors will be effective in the selection of a search direction is established based on the choice of the respective parameters; namely inertia weight parameter (w) for the previous iteration velocity, cognitive factor parameter (c_1) for individual's previous best position, and social factor parameter (c_2) for the best position ever found globally.

It has been shown that the PSO algorithm is quite sensitive to its parameters, and therefore extensive research has been conducted in the literature related to the improvement as well as dynamic adaptation of these parameters during the search process. The adaptive approaches have mainly concentrated on the inertia weight adaptation since a balance between the exploration and exploitation capabilities of the algorithm can easily be established through the adjustment of this parameter.

Exploration is the capability to explore distinct regions of the design space, while exploitation is to improve the current solution by making a more exhaustive search relatively in a small neighborhood of the current solution. The most efficient search process is achieved when two features are combined effectively in an optimization algorithm. In this regard, the inertia weight parameter generally is utilized to balance a search direction between global and local search such that high values of this parameter tend to steer the search towards distinct regions while the small values confine the search to nearby districts.

Generally speaking, at the beginning of the search process, a high value of inertia weight parameter is quite useful and preferable because it allows for a thorough search of the design space. At later optimization stages, when the most promising regions of the design search space are found, the inertia weight should be reduced to decrease the momentum of particles and thereby to focus the search on the most promising regions of the design space. Various approaches have been proposed to achieve this objective. In the following, the time-varying inertia approaches which are adopted and employed in this study are introduced briefly.

In a time-varying strategy, the inertia weight is defined as a function of iteration number or time. Shi and Eberhart (1998, 1999) proposed an approach where inertia weight (w) is reduced from a predefined high value of w_{max} to a low of w_{min} linearly over iterations or time. It was recommended that the values of w_{max} and w_{min} were set to 0.9 and 0.4, respectively. Accordingly, the linearly decreasing inertia weight can be formulated as (Naka et al., 2001):

$$w_{t+1} = w_{max} - (w_{max} - w_{min})\frac{t}{t_{max}} \tag{3.9}$$

where t_{max} is the number of iterations, t is the current iteration, and w_{t+1} is the value of the inertia weight in the t+1 iteration. Accordingly, an initial inertia value w_{max} at t = 0 is linearly reduced towards w_{min} throughout t_{max} iterations.

Another time-varying strategy for the inertia weight is to reduce this parameter in a non-linear fashion from an initially assigned high value. Yang et al. (2015) stated that a nonlinear time-varying inertia weight exhibited higher performance over a linearly decreasing variant and thus they proposed the following nonlinear time-varying inertia weight:

$$w_{t+1} = w_{max} - (w_{max} - w_{min})(\frac{t}{t_{max}})^{\beta} \tag{3.10}$$

where β is a user-defined constant value. The empirical studies indicated that the related parameters of Equation (3.10) could be set to the following values for a satisfactory performance of the algorithm: $\beta = 1/\pi^2$, w_{min} = 0.4, and w_{max} = 0.9.

Chen et al. (2006) studied exponentially decreasing inertia weight using the exponential function and offered two different strategies. The first one is defined as:

$$w_{t+1} = w_{min} + (w_{max} - w_{min})e^{-\frac{10t}{t_{max}}} \tag{3.11}$$

And the second one is defined as:

$$w_{t+1} = w_{min} + (w_{max} - w_{min})e^{-(\frac{4t}{t_{max}})^2} \tag{3.12}$$

The empirical studies indicated that the related parameters of Equations (3.11) and (3.12) could be set to the following values for a satisfactory performance of the algorithm: w_{min} = 0.4, and w_{max} = 0.9.

The logarithmically decreasing inertia weight was integrated into the PSO algorithm by Gao et al. (2008). They formulated a logarithmically decreasing inertia weight as follows:

$$w_{t+1} = w_{max} - (w_{max} - w_{min}) \log_{10}\left(a + \frac{10t}{t_{max}}\right) \tag{3.13}$$

where a is a constant value controlling the evolutionary velocity. The authors empirically suggested the following values for the parameters of Equation (3.13): w_{min} = 0.4, w_{max} = 0.9 and a = 1.

It should be noted that in the original PSO algorithm, the velocity of a particle may frequently rise to an extremely high value, which unfavorably affects the performance of the PSO algorithm. Therefore, the velocity of the particles was limited by defining an upper bound called maximum velocity, V_{max}. Accordingly, If the velocity of a particle exceeds V_{max} during the update of the velocities by Equation (3.8), it is automatically assigned to V_{max}, which is a user-defined parameter.

3.4 Evolution Strategies (ESs)

The evolution strategies (ESs) technique is conceived as one of the three mainstreams of evolution algorithms (EAs). In fact, the term "evolutionary algorithms (EAs)" is used to refer to a group of techniques that employ a crude simulation of natural evolution to evolve a population of individuals (designs) toward the optimum over successive generations. As in any evolutionary algorithm

(EA), evolution strategies (ESs) also require a stochastic and iterative procedure, which endeavours to evolve a population of individuals over a selected number of generations.

The fundamentals of evolution strategies (ESs) were originally laid in the pioneering studies of Rechenberg (1965, 1973) and Schwefel (1965) at the Technical University of Berlin. They developed the first (simplest) variant of ESs, which implements on the basis of two designs; a parent and an offspring individual. Today, the modern variants of the ESs are accepted as $(\mu + \lambda) - \mathrm{ES}$ and $(\mu, \lambda) - \mathrm{ES}$, which were again developed by Schwefel (1977, 1981). Both variants employ design populations consisting of μ parent and λ offspring individuals, and are intended to carry out a self-adaptive search in continuous design spaces. The extensions of these variants to solve discrete optimization problems were put forward in the following studies in the literature: Cai and Thierauf (1993), Bäck and Schütz (1995), Rudolph (1994), and Hasançebi (2007).

In this study, in order to address discrete sizing optimization of steel frames the discrete variant of $(\mu + \lambda) - \mathrm{ES}$ and $(\mu, \lambda) - \mathrm{ES}$ evolution strategies developed in Hasançebi (2007) are adopted and employed. In fact, the performance assessment of various metaheuristic search algorithms in the optimum design of skeletal systems indicated that these discrete variants of the ESs were the most successful ones amongst a total of seven metaheuristic algorithms considered for the related studies (Hasançebi et al. 2009, 2010).

Therefore, in the following first the discrete variants $(\mu + \lambda) - \mathrm{ES}$ and $(\mu, \lambda) - \mathrm{ES}$ of the evolution strategies developed in Hasançebi (2007) are explained in detail first. Next, the fundamentals of the $(1+1) - \mathrm{ES}$ are mentioned very briefly because it forms a basis for the development of the guided evolution strategy proposed in this study.

3.4.1 The (μ,λ)- and (μ+λ)-Evolution Strategy

The modern variants of the ESs, i.e., $(\mu + \lambda) - \text{ES}$ and $(\mu, \lambda) - \text{ES}$, employ an optimization procedure featuring a generation-based iteration of the technique for a population of individuals (potential solutions). Accordingly, the first step in this procedure is to establish an initial population consisting of μ number of parent individuals. Typically, this is performed through a random initialization. Next, the initial population is measured (evaluated), and each parent individual is given a fitness score. The fitness score of an individual quantifies the merit of its solution, and is distributed according to how well the individual satisfies the objective function and constraints. Once the initial population is evaluated, it goes through recombination and mutation evolutionary operators to yield the offspring population. Recombination is applied first, facilitating a trade of design information between the μ parents to generate λ new (offspring) individuals. A variety of different recombination operators exist to carry out this task and their relative effectiveness is an ongoing research subject (Bäck, 1996). Mutation is applied to each offspring individual thereafter, resulting in a new set of design components for the individual. Evaluation of the offspring population is then fulfilled in the same way as the parents. Next, selection operator is implemented to determine the surviving individuals out of parent and offspring populations. The manner the selection is carried out identifies the only difference between the $(\mu + \lambda)$ and (μ, λ) variants of ESs. In the (μ, λ) variant, the parents are all left to die out, and the best μ individuals are chosen deterministically out of λ offspring in reference to their fitness scores. In the $(\mu + \lambda)$ variant, however, the parents are also involved in this mechanism, and the best μ individuals are chosen from μ parents and λ offspring. The selected (surviving) individuals make up the parent population of the next generation. The aforementioned process is iterated in the same way over a certain number of generations, producing a new offspring population from the parent one at each generation. Extensive formulations and computational details of the ES algorithm are given in the following subsections.

3.4.1.1 Initial population

The initial population contains μ parent solutions (individuals). An individual **J** consists of a vector of sizing design variables **I** and two sets of strategy parameters $(p, \boldsymbol{\psi})$, as formulated in Equation (3.14). The strategy parameters are all self-adaptive, which means that they are adjusted to the best suitable values dynamically and automatically by the algorithm to achieve an effective search of the design space during an optimization process.

$$\mathbf{J} = \mathbf{J}(\mathbf{I}, p, \boldsymbol{\psi}) \tag{3.14}$$

In Equation (3.14), p refers to the mutation probability parameter employed for the sizing design variables. It should be noted that this parameter determines the percentages of sizing design variables that will be mutated probabilistically when generating a new (offspring) individual. As explained in the following, in these discrete variants of the ES algorithm, sizing design variables are mutated using geometric distributions. A geometric distribution is a discrete probability distribution with a monotonously decreasing probability curve for a random variable. In general, the flatness of this distribution is controlled by a geometric distribution parameter (ψ). The higher the ψ parameter is, the flatter the distribution becomes, enabling large moves (step sizes) for an effective explorative search. Similarly, the low values of ψ parameter encourage small moves and thus a more exploitative search. For each sizing variable a particular geometric distribution is generated and used by changing its corresponding ψ parameter adaptively during the search for a successful mutation of the variable. Therefore, a different geometric distribution parameter ψ_i is used and associated with each sizing variable I_i and the vector $\boldsymbol{\psi}$ in Equation (3.14) represents the whole set of geometric distribution parameters employed, i.e. $(\psi_i = 1, .., N_g)$. The initial population is formed such that the design vectors are initialized at random, and the strategy parameters are set to suitable initial values $(p(0), \boldsymbol{\psi}(0))$ that are determined through numerical experiments.

3.4.1.2 Evaluation of parent population

The initial population, which is assigned as the parent population at generation zero, is evaluated such that a structural analysis is implemented for each individual in the population to obtain a structural response for the corresponding design. If a solution produced by an individual is feasible, its objective function value is directly calculated from Equation (2.4). Otherwise, the infeasible solution is evaluated using the external penalty function approach, and its objection function is calculated using Equation (2.19).

3.4.1.3 Recombination

After the individuals are evaluated, recombination is implemented next where λ offspring (new) individuals are generated by implementing an exchange of design information between the μ parent individuals. Recombination is performed such that not only the design vector $\mathbf{I}$ but also the strategy parameters $(p, \boldsymbol{\psi})$ are recombined between the randomly selected individuals to create an offspring individual. Generally speaking, recombination can be implemented in different schemes based on the type of design variables. In the present study, sizing design variables are recombined according to a global discrete recombination scheme, whereas an intermediate recombination scheme is adopted for the strategy parameters. Assuming that $\mathbf{s} \in \{\mathbf{I}, p, \boldsymbol{\psi}\}$ represents an arbitrary component of an individual, the recombined $\mathbf{s}'$ can be expressed as follows:

$$\begin{cases} s_i^a \ or \ s_i^{bj} & : g\ lobal\ discrete \\ s_i^a + \frac{s_i^b - s_i^a}{2} & : intermediate \end{cases} \tag{3.15}$$

In Equation (3.15), $\mathbf{s}^a$ and $\mathbf{s}^b$ represent any components of two parent individuals a and b, and s_i^a and s_i^b are typical elements in these components, respectively. In a global discrete recombination of a component **s**, the first parent is chosen and kept fixed, whereas the second parent is chosen all over again for each different element of the component. The recombination is performed by selecting the corresponding element $s_i^{'}$ from that of the first parent s_i^a or the second parent s_i^{bj} under equal probability. In an intermediate recombination scheme, both parents are chosen and held unchanged for all the elements. The recombination is then performed by calculating the arithmetic mean of the elements s_i^a and s_i^b.

3.4.1.4 Mutation

It is essential to emphasize that mutation is conceived of as the main operator of ESs owing to its dominating role in an effective search of the design space. Every offspring individual is subjected to mutation, which arises a new set of design variables ($\mathbf{I}'$) and strategy parameters ($p', \boldsymbol{\psi}'$) for the individual.

Prior to the mutation of sizing variables, the sizing mutation probability parameter p is mutated first. A logistic normal distribution is used for mutation of the parameter in the range of (0,1), as formulated in Equation (3.16).

$$p' = \left(1 + \frac{1-p}{p} \cdot e^{-\gamma \cdot N(0,1)}\right)^{-1} \tag{3.16}$$

In Equation (3.16), p' is the mutated value of the p parameter; $N(0,1)$ is a random number generated from a normal distribution with a mean of 0 and a standard deviation of 1 and it is sampled anew for each individual; and $\gamma = 1/\sqrt{2\sqrt{N_g}}$ is a learning rate constant for the p parameter, where N_g stands for the number of sizing design variables (number of member groups). It is important to mention that upper and lower bounds are also defined for the mutation probability parameter obtained from Equation (3.16), as follows:

$$p_{min} \leq p' \leq p_{max} \tag{3.17}$$

where the minimum mutation probability parameter p_{min} is imposed to ensure the mutability of the individual. In this study, this paramere is set to a value of $p_{min} =$

$1/N_g$ to facilitate mutation of at one design variable probabilistically. Similarly, the maximum mutation probability parameter p_{max} is enforced to eliminate excessive mutation of the individual, In this study, p_{max} is not set to a static value; instead, it is decreased linearly from its initial value of p_{max}^0 at iteration (generation) zero to its final value $p_{max}^{t_{max}}$ at the last iteration in order to promote a more exploitative search towards later stages of optimization (Equation 3.18)

$$p_{max}^{t+1} = p_{max}^0 - \left(p_{max}^0 - p_{max}^{t_{max}}\right)\frac{t}{t_{max}} \tag{3.18}$$

where t_{max} is the maximum number of iterations to be performed during an optimization process, t is the current iteration number, and p_{max}^{t+1} is the maximum value of the mutation probability parameter at iteration $(t+1)$. In this study, p_{max}^0 and $p_{max}^{t_{max}}$ are set to the following values: $p_{max}^0 = 0.25$ and $p_{max}^{t_{max}} = 1/N_g$.

For each sizing variable I_i, a uniform random number r is then sampled in the range [0,1]. If the random number r is smaller than or equal to p', mutation of the sizing variable is performed; otherwise, the variable is not mutated. Before mutating the sizing variable, however, its associated geometric distribution parameter ψ_i must be mutated first using a lognormal distribution given in Equation (3.19).

$$\psi_i' = \psi_i e^{\tau N_i(0,1)} \geq 1.0 \; if \; r \leq p' \in [0,1] \tag{3.19}$$

In Equation (3.19), ψ_i' is the mutated value of the ψ_i parameter; τ is a learning rate constant for the ψ_i parameter and it can be set to $\tau = 1/\sqrt{N_g}$; and a normally distributed random number $N_i(0,1)$ is sampled anew for each sizing variable.

It is important to note that Equation (3.19) may occasionally yield a too small or too big geometric distribution parameter due to mis-adapted strategy parameters, which adversely affect the search performance of the algorithm. Hence, an upper bound ψ_{max} and a lower bound ψ_{min} are also defined for this parameter to facilitate an efficient search, as formulated in Equation (3.20).

$$\psi_{min} \leq \psi'_i \leq \psi_{max} \tag{3.20}$$

Accordingly, if the mutated value of a geometric distribution parameter exceeds ψ_{max}, it is set to $\psi'_i = \psi_{max}$; likewise if it falls below ψ_{min}, it is set to $\psi'_i = \psi_{min}$. In this study, ψ_{min} and ψ_{max} are set to the following values: $\psi_{min} = 2\sqrt{N_{sec}}/5$ and $\psi_{max} = 5\sqrt{N_{sec}}/2$, where N_{sec} is the number of sections (values) in the discrete set used.

Finally, to mutate a sizing variable, two integer random numbers $(z_{i,1}, z_{i,2})$ are sampled according to a geometric distribution with the parameter Ψ'_i, and I_i is mutated using Equation (3.21), where I'_i indicates the mutated value of the sizing variable.

$$I'_i = I_i + z_{i,1}(\Psi'_i) - z_{i,2}(\Psi'_i) \tag{3.21}$$

3.4.1.5 Evaluation of offspring population

The offspring population is evaluated next where structural analysis is implemented for each individual in the offspring population to obtain its force and deformation responses and its objective function value is assigned as per Equation (2.4) or (2.19).

3.4.1.6 Selection

Selection is carried out to determine the surviving individuals. As discussed formerly, the way the selection is carried out identifies the only difference between the ($\mu + \lambda$) and (μ, λ) variants of ESs. In the (μ, λ) variant, the parents are all left to die out, and the best μ individuals are chosen deterministically out of λ offspring in reference to their fitness scores. In the ($\mu + \lambda$) variant, however, the parents are also involved in this mechanism, and the best μ individuals are chosen from μ parents plus λ offspring. It follows that the lifespan of an individual in (μ, λ) – ES is strictly defined as one generation, whereas the individual is permitted to remain alive until it is overwhelmed by subsequent individuals in the ($\mu + \lambda$) variant. At a first glance, ($\mu + \lambda$) − ES seems to be superior in the sense that it increases the selection pressure and comes up with a promise of guaranteed evolution (Bäck, 1996). However, according to Schwefel (1981), this opinion may be deceptive. He hypothesized that the ($\mu + \lambda$) variant is less likely to escape from mis-adapted strategy parameters and local optima, as compared to the (μ, λ) − ES. The selected (surviving) individuals make up the parent population of the next generation.

3.4.1.7 Termination

The procedures described in Sections 3.4.1.3 through 3.4.1.6 are implemented over a predetermined number (N_{gen}) of generations.

3.4.2 The (1+1)-Evolution Strategy

Two member ES, or shortly (1+1)-ES is the initial version of the ESs and includes only two individuals; one parent and one offspring. Therefore, it operates on only two individuals per iteration.

In a (1+1)-ES algorithm, an individual **J** consists of a continuous design variable vector **X** and a single standard deviation σ, as formulated in Equation (3.22)

$$\mathbf{J} = \mathbf{J}(\mathbf{X}, \sigma) \tag{3.22}$$

At each generation (iteration) of the algorithm, an offspring individual is generated from the parent one by mutation only. To do this, the standard deviation parameter σ is first updated according to a so-called 1/5 success rule in which it is adapted deterministically according to the measured success frequency of mutations (Equation 3.23)

$$\sigma' = \begin{cases} \sigma/\sqrt[n]{c} & if\ p > 1/5 \\ \sigma\sqrt[n]{c} & if\ p < 1/5 \\ \sigma & if\ p = 1/5 \end{cases} \tag{3.23}$$

where σ' refers to the adapted (mutated) value of the standard deviation and p is the measured success frequency of mutations. It has been theoretically proven on a sphere model that the parameter c is set to 0.817 for the best performance of the algorithm.

Next, mutation of the design variable vector **X** is carried out where each design variable is added a random number sampled according to a normal distribution with mean 0 and standard deviation σ'

$$X_i' = X_i + N(0, \sigma')_i \qquad i=1, \dots, N_g \tag{3.24}$$

where X_i and X_i' are the values of the i-th design variable in the parent and offspring individuals, respectively, and $N(0, \sigma')$ is a random number sampled according to a normal distribution with a mean 0 and standard deviation σ'.

The better of the two (parent and offspring individuals) is selected deterministically depending on whichever has the lower objective function value or higher fitness, and the selected individual becomes the parent of the next generation, while the other one is discarded. Accordingly, the selection mechanism in (1+1)-ES can mathematically be expressed as follows:

$$\mathbf{J}(t+1) = \begin{cases} \mathbf{J}(t), & if \;\; \phi(\mathbf{J}(t)) \le \phi(\mathbf{J}'(t)) \\ \mathbf{J}'(t), & Otherwise \end{cases} \tag{3.25}$$

where $\mathbf{J}(t+1)$ represents the parent individual selected for the next generation, and $\mathbf{J}(t)$ and $\mathbf{J}'(t)$ are the parent and offspring individuals, respectively, and the fitness function is denoted by ϕ.

3.5 The Proposed Guided Evolution Strategy Technique

In this section, a design-driven guided evolution strategy optimization (GES) technique developed for a time-efficient optimum design of steel skeletal structures is introduced.

A common strategy employed by almost all metaheuristic search algorithms is that they explore the design space stochastically by making random moves based on some search natural, biological or physical principles. These algorithms primarily differ in how a search direction is decided in the design space without using or relying on gradient information. Indeed, a gradient-free search is advantageous in the sense that it prevents an algorithm to stagnate in a local optimum, On the other hand, random search strategies employed by these algorithms cause them to converge towards an optimum solution only after performing a large number of function evaluations. For optimization problems whose objective functions are expressed mathematically as an explicit function of some variables, this is not of much significance because the optimization time will change from a few seconds to a few minutes even for the cases in which millions of function evaluations are necessary. On the other hand, an analysis of a structural system is a computational expansive task since a single analysis may take up to minutes for large-scale problems subjected to many load conditions. Therefore, a time-efficient optimization of large-scale structures may not be always achieved with standard metaheuristic search methods. Therefore, it is crucial to establish intelligent strategies to guide the search direction towards more promising parts of the design space, while implementing metaheuristic search techniques for structural optimization problems

The rationale behind the guided evolution strategy (GES) technique is to improve the convergence characteristics of an evolution strategy by utilizing the information gathered during the structural analysis and design stages. This way, it is aimed to create a time-efficient optimization algorithm having accelerated search abilities.

The GES algorithm has some similarities and differences with (1+1) and (μ, λ) or ($\mu + \lambda$) variants of the technique. Similar to the (1+1)-ES, it works on the basis of two individuals; i.e, a parent and an offspring per generation. The selection scheme in the GES is also implemented in the same way as in the (1+1)-ES. That is to say, if the offspring is better than the parent, it becomes the parent of the next generation; otherwise, the parent survives. On the other hand, the mutation of a parent individual in the GES algorithm is performed along the same line as the (μ, λ) or ($\mu + \lambda$) discrete variants of ES. However, in addition to this mutation operator, the data collected during the evaluation stage of the optimization process (structural analysis and design) are also utilized during mutation to handle strength constraints more efficaciously for guiding the search process. The fundamental implementation steps of the GES algorithm are described in the following sub-sections.

3.5.1 Initialization of the Algorithm

The GES optimization algorithm starts with a random generation of an initial (parent) individual. Similar to (μ, λ) or ($\mu + \lambda$) discrete variants, an individual **J** consists of a vector of sizing design variables **I** and two sets of strategy parameters $(p, \boldsymbol{\psi})$, as formulated in Equation (3.14). The descriptions for these strategy parameters will not be given here to avoid repetition.

3.5.2 Evaluation of the parent individual

The parent individual is evaluated where structural analysis is carried out for the individual to obtain its force and deformation responses and its objective function is calculated using Equation (2.4) or (2.19), depending on the feasibility of the resulting solution.

3.5.3 Mutation

The (μ, λ) or $(\mu + \lambda)$ discrete variants of the ES method employ a stochastic mutation as formulated in Equations (3.16) through (3.21). Indeed, a stochastic mutation is useful since it accommodates an extensive exploration of the design space, preventing entrapment of the algorithm in a local optimum. Therefore, a stochastic mutation is also maintained in the GES algorithm. However, in order to accelerate the convergence speed of the algorithm, a so-called guided mutation is introduced as a supplementary tool to guide the search process using the information obtained during the evaluation stage of the parent individual. In order for the algorithm to take advantage of these two search features together during an optimization process, both mutation schemes are incorporated into the GES algorithm. Accordingly, a stochastic mutation is still followed when generating an offspring individual from the parent one, yet occasionally this conventional mutation scheme is replaced by the guided mutation scheme according to a predefined possibility called guided mutation ratio (*GMR*) in this study. Whether an offspring is generated based on a stochastic or guided mutation is decided using Equation (3.26).

$$M_{type} = \begin{cases} Stochastic, & if\ \ r > GMR \in [0,1] \\ Guided\quad , & if\ \ r \leq GMR \in [0,1] \end{cases} \tag{3.26}$$

Equation (3.22) is implemented such that when a new offspring individual is generated, a uniform random number (r) is sampled anew in the range of $[0,1]$. If $r \leq GMR$, the offspring individual is produced using the guided mutation scheme; otherwise $(r > GMR)$ the conventional stochastic mutation is applied. In Chapter 5, the optimal values of the *GMR* parameter are investigated by implementing numerical examples from the optimum design of steel frames. In the following, these two mutation schemes are explained in detail.

3.5.3.1 Stochastic Mutation

As mentioned previously, the stochastic mutation scheme is implemented along the same line as (μ, λ) or (μ + λ) discrete variants of the method, as formulated in Equations (3.16) through (3.21).

3.5.3.2 Guided Mutation

During the evaluation stage, a parent individual design is analyzed under all design load combinations, and the maximum demand-to-capacity ratio (DCR) is established for each sizing variable concerning the strength constraints. For a sizing design variable, a DCR value below one indicates that related strength constraints are satisfied with the current section selected for the corresponding member or member group, and a smaller (lighter) section might be used to provide economy (i.e., the current section is overdesign). Similarly, On the other hand, for a sizing design variable a DCR value above one implies that related strength constraints are not satisfied; the current section is not adequate (underdesigned) and a stronger section must be used to eliminate the constraint violation.

In guided mutation scheme, search directions are determined for design variables in accordance with their associated DCR values in the parent individual. Accordingly, for design variables with DCR values larger than one, the type of search direction is labeled as $SD_{increase}$, implying that an increase in cross-section (i.e., stronger section) is necessary to remedy the related constraint violations associated with these variables. Conversely, for design variables with DCR values smaller than one, the type of search direction is labeled as $SD_{decrease}$, indicating that a decrease in cross-section (i.e., lighter section) might be useful in terms of more effective use of material, as formulated in Equation (3.27)

$$SD_{type} = \begin{cases} SD_{increase}, & if\ \ DCR > 1.0 \\ SD_{decrease}, & if\ \ DCR \leq 1.0 \end{cases} \tag{3.27}$$

In fact, at times when no global displacement constraint is imposed, ideally, DCRs for all member groups should be adjusted to values around 1.0 to ensure the most effective use of the material in a structure. However, the presence of global displacement constraints might overwrite this rule to some extent because some global displacement quantities might be quite sensitive to the cross-sectional properties of some members, and the selection of larger sections for those members might be necessary to reduce the global displacements to the desired levels at the expense of having low DCR values for those particular members.

Therefore, not all design variables are mutated at a time while generating an offspring individual. Besides, excessive mutation of an individual causes quite a different set of force distributions within the structural members, reducing the exploitative search characteristics of the algorithm. Therefore, the mutation is implemented only on a selected number of design variables and whether a design variable is mutated or not during the guided mutation scheme is decided upon using Equation (3.28)

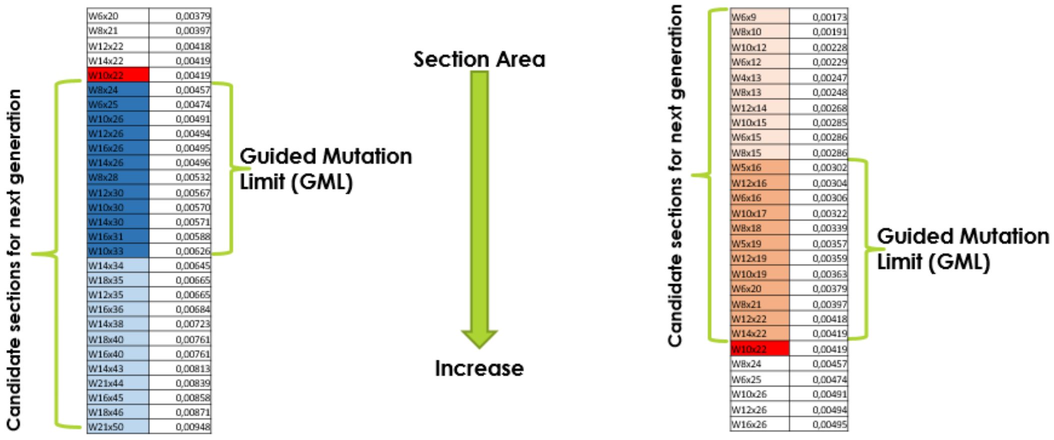
$$Mutation = \begin{cases} Yes, SD_{increase} & if\ DCR > 1.0 \\ Yes;\ SD_{decrease} & if\ DCR < 1.0 \ \text{ and } r < (1 - DCR)^2 \in [0,1] \\ No & if\ DCR < 1.0 \ \text{ and } r \geq (1 - DCR)^2 \in [0,1] \end{cases} \quad (3.28)$$

According to Equation (3.28), the design variables with DCR values above one are strictly mutated in an increased search direction $SD_{increase}$ to have a stronger section and thereby to remedy the related constraint violation. On the other hand, design variables with DCR values below one are mutated probabilistically, where $(1 - DCR)^2$ represents the mutation probability of the design variable. A uniform random r is generated anew in the range of $[0,1]$ and compared to the mutation probability of the design variable. If $r < (1 - DCR)^2$, the design variable is mutated in a decreased search direction $SD_{decrease}$ to have a lighter section; otherwise no mutation is applied to the design variable. It should also be noted in Equation (3.28) that the design variables with lower DCR values are more likely to be mutated as the mutation probability parameter $(1 - DCR)^2$ approaches to one.

One point deserves particular attention. It is important to emphasize that in discrete structural optimization the sections that will be used to size member groups in a structure are sorted in ascending order of cross-sectional areas in a section pool (discrete set). An increase in cross-section in this list results in a heavier section, but not necessarily a stronger section. That is to say, the next section is heavier in cross-section but it may have a smaller moment of inertia about the strong or weak axis or both. On the other hand, in frame-type structures, multiple failure modes are usually available for structural members under the combined effect of axial force and bending moment. For example, one member may fail under shear; another member may fail under bending about the strong axis or weak axis, or local buckling mechanisms may be observed, etc. Therefore, an increase in member size in the discrete set may lead to a heavier yet weaker section; likewise, a decrease in member size in the discrete set might lead to both a lighter and stronger section.

In this study, an increased or a decreased search direction for a design variable is implemented by preparing a so-called “Increase Direction List (IDL)” in the former and “Decrease Direction List (DDL)” in the latter. In this regard, an IDL consists of a predefined number of sections that have sectional properties larger than those of the section selected currently for the design variable, not only in terms of cross-sectional area but also the moments of inertias about strong and weak axes. Hence, to prepare a IDL sections very next to the currently assigned value of the variable in the list are scanned in the increasing direction of the sequence number, and only those larger (heavier) sections which also have larger moment of inertias about both axes are included in the list. On the other hand, a DDL consists of a predefined number of sections that are lighter than the section selected currently for the design variable. As discussed above, sometimes a decrease in member size in the discrete set might lead to a lighter yet stronger section, which is in fact in favour of the design variable. Accordingly, a DDL is prepared such that a predefined number of sections very next to the currently assigned value of the variable in the decreasing direction of the sequence number are included in the list without enforcing any requirement that they will also have smaller moment of inertias. The preparation of

IDL and DDL for a design variable is illustrated in Figure 3.1, where the IDL and DDL lists are marked by blue and orange colors, respectively. Once an IDL or DDL is prepared for a design variable as described above, the mutation of the design variable is performed by stochastically switching to any of these sections in the corresponding list under equal probability.


Candidate sections for next generation
Guided Mutation Limit (GML)
Section Area
Increase
Candidate sections for next generation
Guided Mutation Limit (GML)

W6x20	0,00379
W8x21	0,00397
W12x22	0,00418
W14x22	0,00419
W10x22	0,00419
W8x24	0,00457
W6x25	0,00474
W10x26	0,00491
W12x26	0,00494
W16x26	0,00495
W14x26	0,00496
W8x28	0,00532
W12x30	0,00567
W10x30	0,00570
W14x30	0,00571
W16x31	0,00588
W10x33	0,00626
W14x34	0,00645
W18x35	0,00665
W12x35	0,00665
W16x36	0,00684
W14x38	0,00723
W18x40	0,00761
W16x40	0,00761
W14x43	0,00813
W21x44	0,00839
W16x45	0,00858
W18x46	0,00871
W21x50	0,00948

W6x9	0,00173
W8x10	0,00191
W10x12	0,00228
W6x12	0,00229
W4x13	0,00247
W8x13	0,00248
W12x14	0,00268
W10x15	0,00285
W6x15	0,00286
W8x15	0,00286
W5x16	0,00302
W12x16	0,00304
W6x16	0,00306
W10x17	0,00322
W8x18	0,00339
W5x19	0,00357
W12x19	0,00359
W10x19	0,00363
W6x20	0,00379
W8x21	0,00397
W12x22	0,00418
W14x22	0,00419
W10x22	0,00419
W8x24	0,00457
W6x25	0,00474
W10x26	0,00491
W12x26	0,00494
W16x26	0,00495

(a) (b)

Figure 3.1 An illustration for the preparation of IDL and DDL lists for a design variable, (a) For IDL (b) For DDL

The number of sections included in an IDL or DDL is referred to as “Guided Mutation Limit (*GML*)” in this study. Indeed, the *GML* is identical to the term “step size” in a traditional optimization algorithm. The larger this parameter is; the more explorative search can be achieved at the expense of decreased exploitative search capabilities of the algorithm. Indeed, at the beginning of the optimization process, a large step size is generally beneficial since it allows for an extensive exploration of the design space. However, the exploitation of potentially better solutions in favorable regions requires a smaller step size, especially towards the later stages of the optimization process. Therefore, in Chapter 5 numerical experiments are carried out to test and examine various strategies for determining the suitable values for the GML parameter.

3.5.4 Evaluation of the offspring individual

Upon the generation of the offspring individual, it undergoes an evaluation stage where structural analysis is carried out to obtain its force and deformation responses and its objective function is calculated using Equation (2.4) or (2.19), depending on the feasibility of the resulting solution.

3.5.5 Selection

The selection process is then implemented between the parent and offspring individuals such that whichever is better (having a lower objective function value) among the two survives and the other one dies out. It should be clear that the GES algorithm employs a strict elitism rule (except during a stagnation escape period to be discussed in the following section) such that the offspring individual is only allowed to replace the parent individual if it is better than the latter. This feature of the GES algorithm is very suitable for the application of a so-called upper bound strategy (UBS) proposed by Kazemzadeh Azad et al. (2013) to significantly reduce the number of structural analyses required during an optimization process. In fact, the principle of UBS lies in the idea of eliminating unnecessary structural analysis for an offspring individual that has no chance to defeat the parent individual. This strategy is implemented such that whenever an offspring individual is generated, the structural weight of the resulting design is calculated first. The structural analysis of the offspring individual is implemented only if its structural weight is lower than the objective function (penalized weight, ϕ) of the parent individual; otherwise, the offspring is automatically eliminated without performing any structural analysis as it has no chance to defeat the parent individual anyway.

3.5.6 Stagnation Control Strategy

The GES algorithm utilizes a basic stagnation control strategy based on an uphill move to avoid entrapment at a local optimum. According to this strategy, if the parent individual is not improved over a predetermined number of generations (N_{stag}) a stagnation escape period (SEP) is initiated by the algorithm in the next generation. Before the SEP, the algorithm only allows for a transition to a better solution, and thus the parent individual (also called elite design) represents the best solution obtained thus far in an optimization process. With the start of the SEP, the elitism rule is suspended temporarily, and a transition to a non-improving offspring individual (uphill move) is allowed only once provided that it has an objective function value not more than a predefined ratio (α_{SEP}) of the objective function of the elite design. This offspring solution, which substitutes the elite design, is designated as the first temporary elite design within the SEP. Once the first temporary elite design is established, the elitism rule is re-activated and it is replaced and updated by only an improving offspring individual. On the other hand, when the SEP proceeds, if an offspring individual better than the elite design is produced, the SEP is terminated immediately and it becomes the elite design of the optimization process. If no improvement is achieved within the SEP, another SEP loop is initiated but this time uphill move is performed with reference to the last temporary elite design of the previous SEP.

3.5.7 Termination

The procedures described in Sections 3.5.3 through 3.5.6 are implemented until over a predetermined number (N_{gen}) of generations is completed.

The optimization methods introduced in this chapter are programmed and computerized in a software platform called SOPS (Structural Optimization Platform Software) developed in this study. This software platform is introduced and explained in detail in the following chapter.

CHAPTER 4

STRUCTURAL OPTIMIZATION PLATFORM SOFTWARE (SOPS)

4.1 Introduction

In structural optimization, selecting a suitable software package to manage and execute the optimization process is crucial since software performance directly impacts the efficiency of optimization. Regarding structural optimization, software packages can be categorized as (1) computing software and (2) design software. Computing software covers the optimization routines and codes while the latter performs the structural analysis and design part of the optimization process. In the computing software, an optimization algorithm is executed iteratively. After each iteration, new candidate solutions are generated by assigning a modified set of values (sections from predefined profile lists) to design variables (structural members' cross-sectional areas). This newly developed candidate solution is passed to the analysis and design software to evaluate the candidate solution, i.e., to determine the fitness value of the generated design. The analysis and design results are transferred back into the computing software. Since this optimization cycle is repeated numerous times before reaching the convergence or optimum solution, there is a great demand for developing a software platform where these optimization and design cycles are automated. Therefore, various design optimization software has implemented design software into their codes. In other words, the design software part of the optimization software package is merged into the computing software part. In the early studies, the primary motivation for combining computing software and the design software instead of executing them side-by-side was the absence or inefficient communications with the design software via their Application Programming Interface (API). However, the design optimization software, developed using this approach and hence capable of performing structural analysis, supports generally limited design code specifications and structure types.

On the other hand, in practice, structural analysis and design are generally performed using commercial analysis and design software. SAP2000 is a well-known and extensively used analysis program for structural engineers. It has a complex, detailed, and well-documented API library. Through the API, the package's qualifications and potentials, such as modeling, analysis, and design, are entirely available for the external programs, which is the foremost reason for deciding SAP2000 as a design software component of SOPS. Therefore, instead of merging analysis-design codes into the optimization module, both computing software and design software are executed side-by-side interactively by transferring the related input-output data to each other. As a result, there is a great demand for a software platform that integrates optimization computing software and commercial design software packages to automate the optimization and design procedures. The practical, efficient and robust infrastructure that combines the optimization techniques' potentials with the design software capabilities together in one single master software platform is developed in this study (Structural Optimization Platform Software, SOPS).

To develop the optimization module of the SOPS framework, some of the well-known metaheuristic optimization algorithms are implemented into SOPS, such as big bang big crunch (BB-BC), particle swarm optimization (PSO), and Evolution Strategies (ES). Moreover, a design-driven guided evolution strategy (GES) optimization technique for the optimal design of real-world steel structures, which is also developed in this study and described in the previous chapter, is also integrated into the optimization module of the SOPS. The SOPS framework and its main components are explained in the following sections.

4.2 Integration of SOPS with SAP2000 API

The SAP2000 API (Application Programming Interface) is an in-between software that provides an interface for two applications to communicate with each other. It provides runtime access to the analysis and design modules of the SAP2000 for third-party applications, as shown in Figure 4.1, where the orange-colored part represents the external third-party applications like SOPS and the blue-colored part means the SAP2000 program.

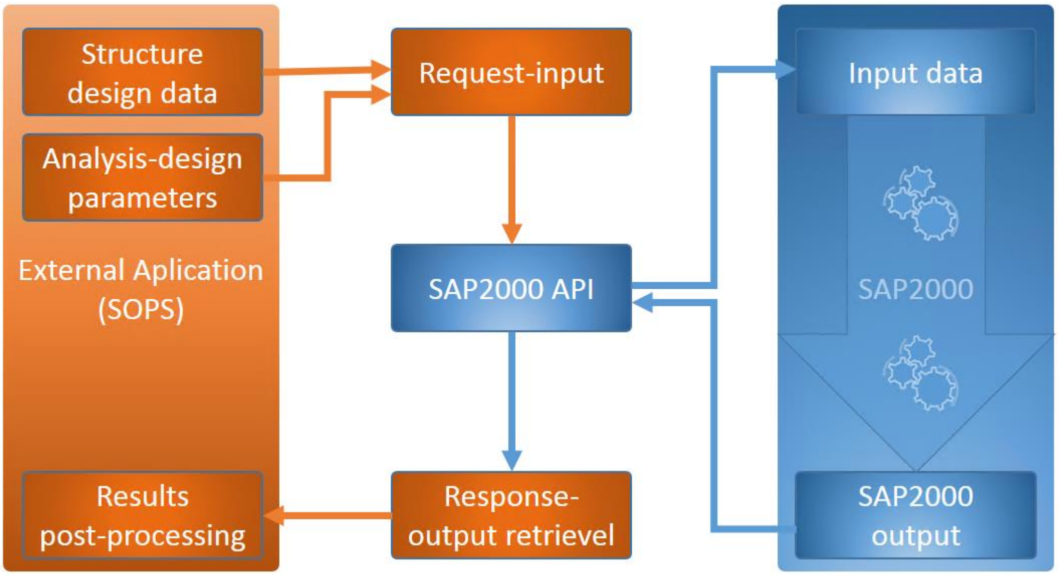


Figure 4.1 Interaction of external application (SOPS) with the SAP2000 API.

The SAP2000 API contains a library through which the program can be utilized and controlled remotely to operate like the classical point-and-click procedure. No additional pre-request of programs is needed to connect to the SAP2000 program through its API except the SAP2000 installation itself. In other words, all the components required for SAP2000 API are automatically installed with the main program setup. Programming language's integrated development environments (IDEs) contain various options for connecting the programming project with the API. The main steps regarding the SAP2000 analysis and design procedures of a structure through its API are very similar to the corresponding point-and-click processes required by the standard SAP2000 program.

4.3 SOPS Architecture

Structural Optimization Platform Software (SOPS) is designed and arranged in modules according to the functions that are intended to perform. SOPS modules can be classified as (i) Input Modules, (ii) Constraints Module, (iii) Analysis-Design

Modules, and (iv) Optimization Modules. These modules are introduced in the following sections. The flow of the program through these modules is managed via an interactive graphical user interface (GUI), which contains the main window that encloses all the modules and components necessary to solve the structural optimization problems. The analysis and design parts, however, are executed entirely inside a separate module that binds the SOPS program with the SAP2000 software through SAP2000 API (Figure 4.2).

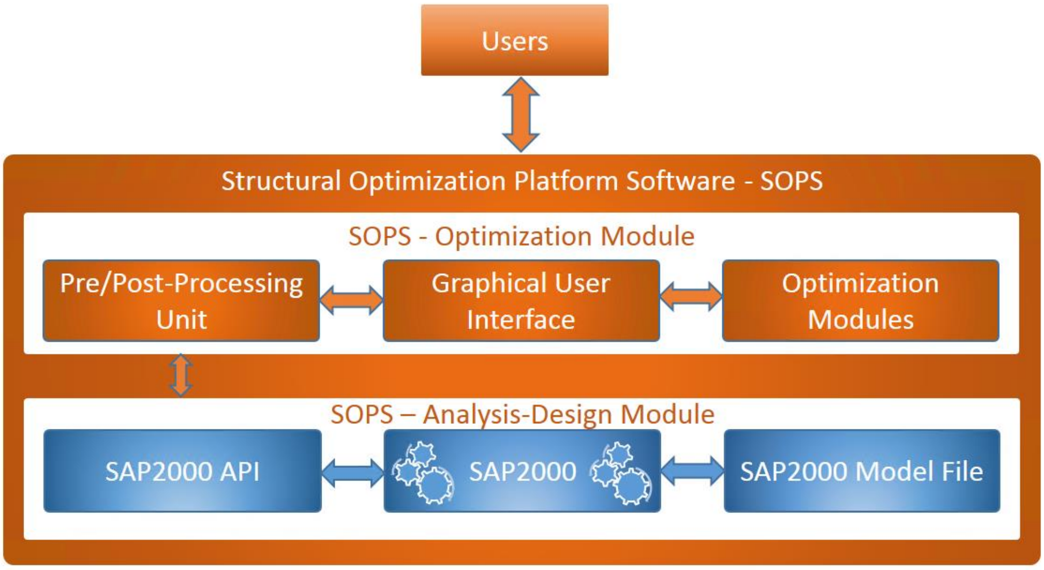
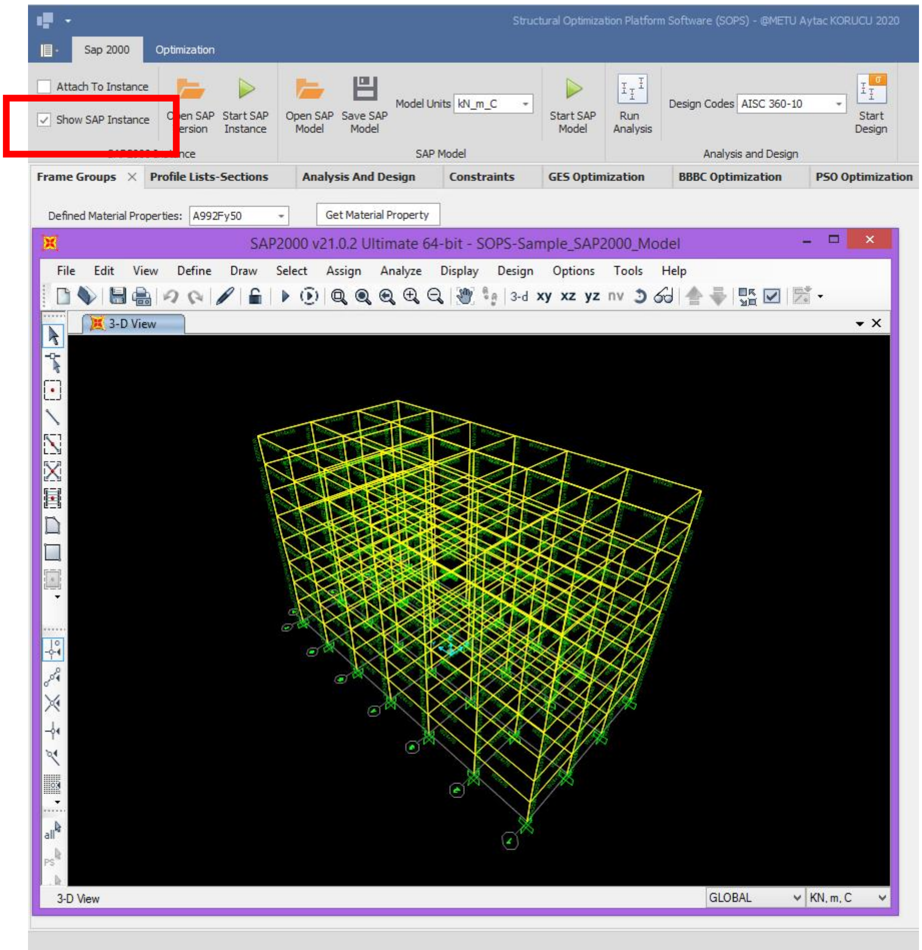


Figure 4.2 Structural Optimization Platform Software (SOPS) architecture

The users are expected to interact SOPS program entirely via the graphical user interface (GUI) provided, while analysis and design parts of the optimization program run silently in the background of the operating system (Figure 4.3). However, if needed, the user can also bring the SAP2000 user interface to the foreground (by selecting the “Show SAP Instance” option as shown in Figure 4.3) and make explicit modifications to the parameters of the SAP2000 program itself. Therefore, the users have complete control over the procedure.

The requests related to the modifications of the SAP2000 model file (assigning new sections to the design variables etc.) are transferred via the API to the SAP2000 program. Similarly, the response (i.e., analysis and design results of individual solutions) are received back again through the SAP2000 API.



☐ Attach To Instance

☑ Show SAP Instance

Model Units: kN_m_C

Design Codes: AISC 360-10

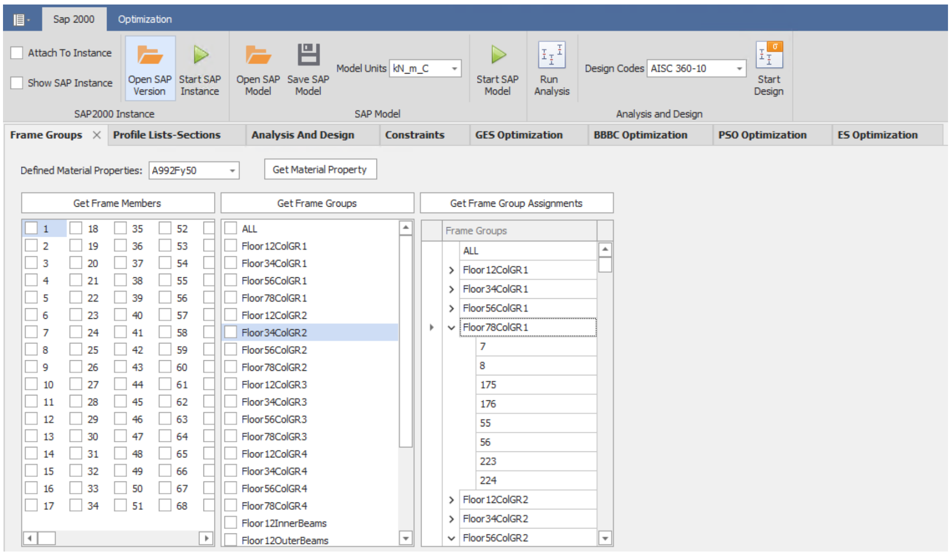
Defined Material Properties: A992Fy50

Figure 4.3 Interaction with SAP2000 via GUI of SOPS

The following sections explain each SOPS module's function in a complete optimization process cycle.

4.3.1 SOPS Input/output Module

SOPS platform receives the data required in the optimization process through the interactive GUI and predefined input files. Since SOPS mainly consists of two parts, i.e., optimization and analysis, two types of input data should be provided. Creation of SAP2000 model file, analysis parameters not modified through optimization, etc., are considered pure analysis-related input data. Similarly, other inputs such as parameters of algorithms, analysis parameters that are modified in the optimization process (i.e., member sections), etc., are categorized as optimization process input data. To minimize the dependency of the SOPS platform to the SAP2000 version updates, pure analysis-related input data are supplied via the SAP2000 program GUI. Other inputs are entered into the SOPS via its interactive GUI and predefined input files. As mentioned, the structural optimization problem model created using the SAP2000 program can be linked to the SOPS platform via SAP2000 API. Therefore, the optimization problem's SAP2000 model can be parsed and transferred to the related optimization module variables such as material property, structural element, frame member groups, and their elements (Figure 4.4).



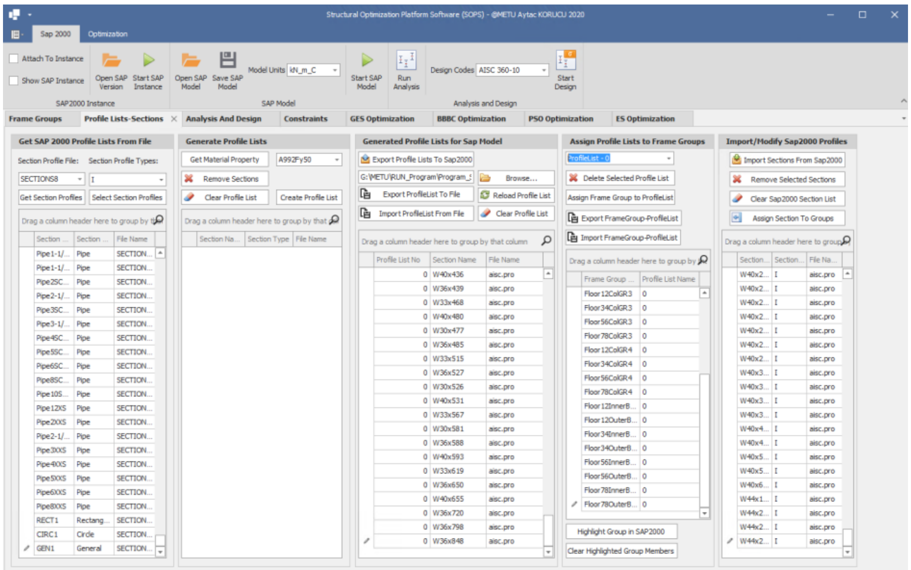
Model Units kN_m_C

Design Codes AISC 360-10

Defined Material Properties: A992Fy50

Figure 4.4 Extracting SAP2000 model parameters into SOPS.

In structural sizing optimization, design variables which are the structural members' cross-sectional areas, are usually picked from a set of predefined profile sections. Therefore, these predefined section lists should be provided to the optimization module of SOPS. There is a form-based input interface where predefined profile lists of SAP2000 can be imported, arranged, and organized as section lists (Figure 4.5). These section lists can also be exported to a file and imported if needed for later runs. In SOPS, each design variable can have its section list, or all design variables can be assigned from a single section list depending on the structure's functional or geometric requirements. SOPS platform presents a flexible infrastructure for section list definitions.


Structural Optimization Platform Software (SOPS) - @METU Aytac KORUCU 2020
Sap 2000
Optimization
Attach To Instance
Show SAP Instance
Open SAP Version
Start SAP Instance
SAP2000 Instance
Open SAP Model
Save SAP Model
SAP Model
Start SAP Model
Run Analysis
Start Design
Analysis and Design
Frame Groups
Profile Lists-Sections
Analysis And Design
Constraints
GES Optimization
BBBC Optimization
PSO Optimization
ES Optimization
Get SAP 2000 Profile Lists From File
Get Section Profiles
Select Section Profiles
Generate Profile Lists
Get Material Property
Remove Sections
Clear Profile List
Create Profile List
Generated Profile Lists for Sap Model
Export Profile Lists To Sap2000
Browse...
Export ProfileList To File
Reload Profile List
Import ProfileList From File
Clear Profile List
Assign Profile Lists to Frame Groups
Delete Selected Profile List
Assign Frame Group to ProfileList
Export FrameGroup-ProfileList
Import FrameGroup-ProfileList
Highlight Group in SAP2000
Clear Highlighted Group Members
Import/Modify Sap2000 Profiles
Import Sections From Sap2000
Remove Selected Sections
Clear Sap2000 Section List
Assign Section To Groups
Drag a column header here to group by that column

☐ Attach To Instance
☐ Show SAP Instance

Model Units: kN_m_C

Design Codes: AISC 360-10

Section Profile File: SECTIONS8

Section Profile Types: I

Get Material Property: A992Fy50

G:\METU\RUN_Program\Program_

ProfileList - 0

Section ...	Section ...	File Name
Pipe1-1/...	Pipe	SECTION...
Pipe1-1/...	Pipe	SECTION...
Pipe2SC...	Pipe	SECTION...
Pipe2-1/...	Pipe	SECTION...
Pipe3SC...	Pipe	SECTION...
Pipe3-1/...	Pipe	SECTION...
Pipe4SC...	Pipe	SECTION...
Pipe5SC...	Pipe	SECTION...
Pipe6SC...	Pipe	SECTION...
Pipe8SC...	Pipe	SECTION...
Pipe10S...	Pipe	SECTION...
Pipe12XS	Pipe	SECTION...
Pipe2XXS	Pipe	SECTION...
Pipe2-1/...	Pipe	SECTION...
Pipe3XXS	Pipe	SECTION...
Pipe4XXS	Pipe	SECTION...
Pipe5XXS	Pipe	SECTION...
Pipe6XXS	Pipe	SECTION...
Pipe8XXS	Pipe	SECTION...
RECT1	Rectang...	SECTION...
CIRC1	Circle	SECTION...
GEN1	General	SECTION...

Section Na...	Section Type	File Name

Profile List No	Section Name	File Name
0	W40x436	aisc.pro
0	W36x439	aisc.pro
0	W33x468	aisc.pro
0	W40x480	aisc.pro
0	W30x477	aisc.pro
0	W36x485	aisc.pro
0	W33x515	aisc.pro
0	W36x527	aisc.pro
0	W30x526	aisc.pro
0	W40x531	aisc.pro
0	W33x567	aisc.pro
0	W30x581	aisc.pro
0	W36x588	aisc.pro
0	W40x593	aisc.pro
0	W33x619	aisc.pro
0	W36x650	aisc.pro
0	W40x655	aisc.pro
0	W36x720	aisc.pro
0	W36x798	aisc.pro
0	W36x848	aisc.pro

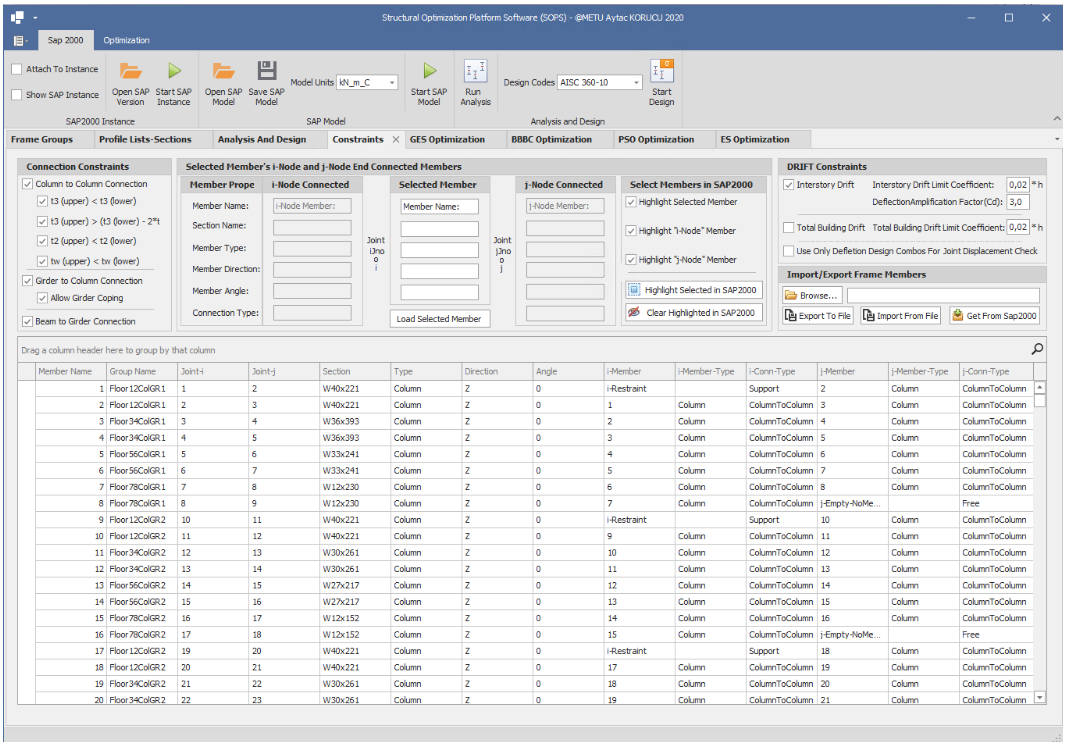
Frame Group ...	Profile List Name
Floor12ColGR3	0
Floor34ColGR3	0
Floor56ColGR3	0
Floor78ColGR3	0
Floor12ColGR4	0
Floor34ColGR4	0
Floor56ColGR4	0
Floor78ColGR4	0
Floor12InnerB...	0
Floor12OuterB...	0
Floor34InnerB...	0
Floor34OuterB...	0
Floor56InnerB...	0
Floor56OuterB...	0
Floor78InnerB...	0
Floor78OuterB...	0

Section...	Section...	File Na...
W40x2...	I	aisc.pro
W40x2...	I	aisc.pro
W40x2...	I	aisc.pro
W40x2...	I	aisc.pro
W40x2...	I	aisc.pro
W40x2...	I	aisc.pro
W40x2...	I	aisc.pro
W40x3...	I	aisc.pro
W40x3...	I	aisc.pro
W40x3...	I	aisc.pro
W40x3...	I	aisc.pro
W40x4...	I	aisc.pro
W40x4...	I	aisc.pro
W40x5...	I	aisc.pro
W40x5...	I	aisc.pro
W40x6...	I	aisc.pro
W44x1...	I	aisc.pro
W44x2...	I	aisc.pro
W44x2...	I	aisc.pro
W44x2...	I	aisc.pro

Figure 4.5 Section List Form GUI of SOPS

4.3.2 SOPS Constraints Module

The constraints subjected to the structural optimization problem should be defined and handled in the optimization process. Therefore, constraints identified in chapter 2 are integrated into the SOPS's constraints module (Figure 4.6). Input data related to the constraints of the optimization problem can be inserted into the SOPS via provided interactive GUI and predefined input files. For regular frame structures, member connection details such as how members are connected to each other and the type of connection (column-to-column, girder-to-column, etc.) can be extracted from the SAP2000 model and presented in a tabular form in the constraints module of the SOPS. This list can be exported to a file for later usage or customizations and imported back into the SOPS.



☐ Attach To Instance
☐ Show SAP Instance
Model Units: kN_m_C
Design Codes: AISC 360-10

Connection Constraints
☑ Column to Column Connection
  ☑ t3 (upper) < t3 (lower)
  ☑ t3 (upper) > (t3 (lower) - 2*t
  ☑ t2 (upper) < t2 (lower)
  ☑ tw (upper) < tw (lower)
☑ Girder to Column Connection
  ☑ Allow Girder Coping
☑ Beam to Girder Connection

Member Name: i-Node Member: Member Name: j-Node Member:
Section Name:
Member Type:
Member Direction:
Member Angle:
Connection Type:

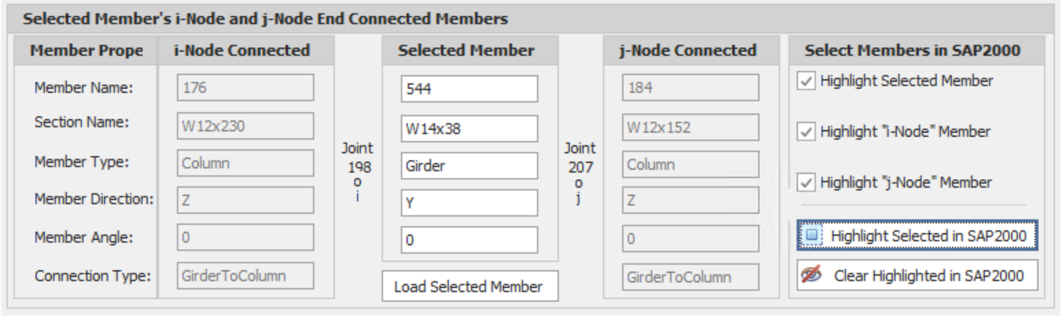
☑ Highlight Selected Member
☑ Highlight "i-Node" Member
☑ Highlight "j-Node" Member

DRIFT Constraints
☑ Interstory Drift Interstory Drift Limit Coefficient: 0,02 *h
DeflectionAmplification Factor(Cd): 3,0
☐ Total Building Drift Total Building Drift Limit Coefficient: 0,02 *h
☐ Use Only Defletion Design Combos For Joint Displacement Check

Member Name	Group Name	Joint-i	Joint-j	Section	Type	Direction	Angle	i-Member	i-Member-Type	i-Conn-Type	j-Member	j-Member-Type	j-Conn-Type
1	Floor12ColGR1	1	2	W40x221	Column	Z	0	i-Restraint		Support	2	Column	ColumnToColumn
2	Floor12ColGR1	2	3	W40x221	Column	Z	0	1	Column	ColumnToColumn	3	Column	ColumnToColumn
3	Floor34ColGR1	3	4	W36x393	Column	Z	0	2	Column	ColumnToColumn	4	Column	ColumnToColumn
4	Floor34ColGR1	4	5	W36x393	Column	Z	0	3	Column	ColumnToColumn	5	Column	ColumnToColumn
5	Floor56ColGR1	5	6	W33x241	Column	Z	0	4	Column	ColumnToColumn	6	Column	ColumnToColumn
6	Floor56ColGR1	6	7	W33x241	Column	Z	0	5	Column	ColumnToColumn	7	Column	ColumnToColumn
7	Floor78ColGR1	7	8	W12x230	Column	Z	0	6	Column	ColumnToColumn	8	Column	ColumnToColumn
8	Floor78ColGR1	8	9	W12x230	Column	Z	0	7	Column	ColumnToColumn	j-Empty-NoMe...		Free
9	Floor12ColGR2	10	11	W40x221	Column	Z	0	i-Restraint		Support	10	Column	ColumnToColumn
10	Floor12ColGR2	11	12	W40x221	Column	Z	0	9	Column	ColumnToColumn	11	Column	ColumnToColumn
11	Floor34ColGR2	12	13	W30x261	Column	Z	0	10	Column	ColumnToColumn	12	Column	ColumnToColumn
12	Floor34ColGR2	13	14	W30x261	Column	Z	0	11	Column	ColumnToColumn	13	Column	ColumnToColumn
13	Floor56ColGR2	14	15	W27x217	Column	Z	0	12	Column	ColumnToColumn	14	Column	ColumnToColumn
14	Floor56ColGR2	15	16	W27x217	Column	Z	0	13	Column	ColumnToColumn	15	Column	ColumnToColumn
15	Floor78ColGR2	16	17	W12x152	Column	Z	0	14	Column	ColumnToColumn	16	Column	ColumnToColumn
16	Floor78ColGR2	17	18	W12x152	Column	Z	0	15	Column	ColumnToColumn	j-Empty-NoMe...		Free
17	Floor12ColGR2	19	20	W40x221	Column	Z	0	i-Restraint		Support	18	Column	ColumnToColumn
18	Floor12ColGR2	20	21	W40x221	Column	Z	0	17	Column	ColumnToColumn	19	Column	ColumnToColumn
19	Floor34ColGR2	21	22	W30x261	Column	Z	0	18	Column	ColumnToColumn	20	Column	ColumnToColumn
20	Floor34ColGR2	22	23	W30x261	Column	Z	0	19	Column	ColumnToColumn	21	Column	ColumnToColumn

Figure 4.6 Constraints Form GUI of SOPS

In the constraints module, the connection details of members can be examined by selecting the related member and loading the information into the connection details part of the form, as shown in Figure 4.7. In this form, every member's end-connected member can be seen. As shown in the figure, the direction of the connected member (X, Y, Z), assigned section name, and type (column, girder, diagonal, etc.) can also be determined. Moreover, the selected member with its connected end points' neighbors can be directly highlighted in the SAP2000 model using SOPS GUI. The request is transferred to the SAP2000 program via API (Figure 4.8).



Selected Member's i-Node and j-Node End Connected Members

Member Prope	i-Node Connected	Selected Member	j-Node Connected
Member Name:	176	544	184
Section Name:	W12x230	W14x38	W12x152
Member Type:	Column	Girder	Column
Member Direction:	Z	Y	Z
Member Angle:	0	0	0
Connection Type:	GirderToColumn		GirderToColumn

Select Members in SAP2000

☑ Highlight Selected Member

☑ Highlight "i-Node" Member

☑ Highlight "j-Node" Member

Figure 4.7 Each End of the Member’s Connection Details

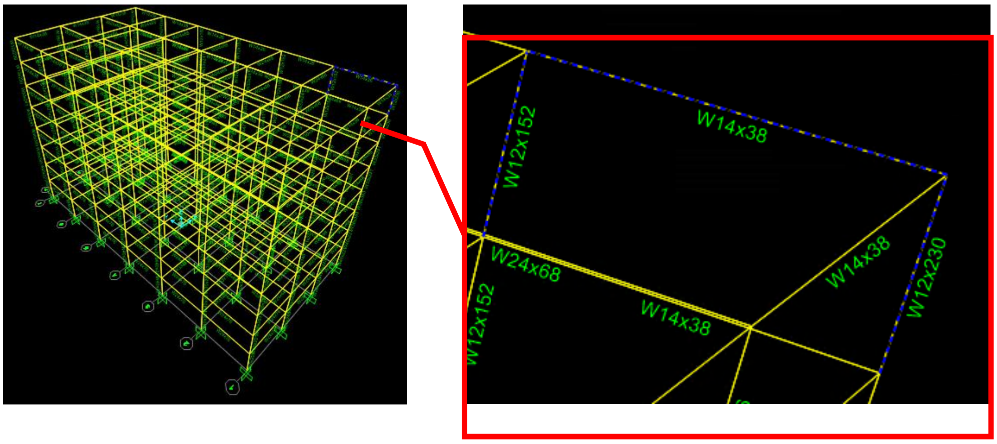
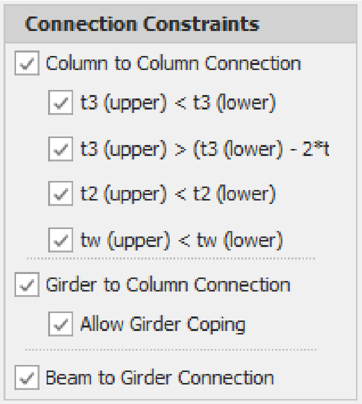


Figure 4.8 Highlighted Member Connection Details

After connections of the frame's structural elements are parsed and transferred into the SOPS program as mentioned above, geometric constraints described in chapter 2 can be enforced using the "Connection Constraints" part of the SOPS module (Figure 4.9).

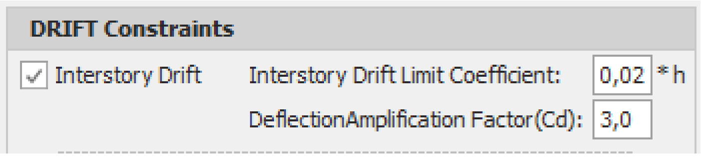


Connection Constraints

- ☑ Column to Column Connection
  - ☑ t3 (upper) < t3 (lower)
  - ☑ t3 (upper) > (t3 (lower) - 2*t
  - ☑ t2 (upper) < t2 (lower)
  - ☑ tw (upper) < tw (lower)
- ☑ Girder to Column Connection
  - ☑ Allow Girder Coping
- ☑ Beam to Girder Connection

Figure 4.9 Enforcing Geometric Constraints

Similarly, inter-story drift constraint is defined in the “Drift Constraints” part of the SOPS. The parameters specified in related design codes, described in chapter 2, can be assigned through the input form GUI of SOPS (Figure 4.10).


DRIFT Constraints

☑ Interstory Drift

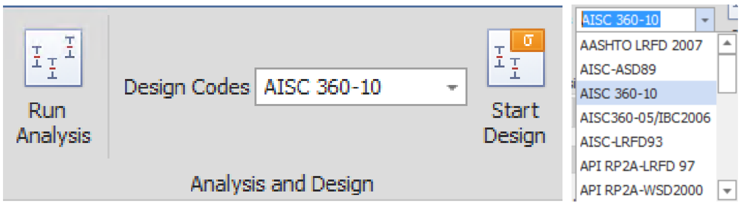
Interstory Drift Limit Coefficient: 0,02 *h

DeflectionAmplification Factor(Cd): 3,0

Figure 4.10 Enforcing Inter-story Drift Constraints

4.3.3 SOPS Analysis and Design Module

International design codes that were implemented in SAP2000 software products are automatically integrated into SOPS via the SAP2000 API (Figure 4.11). Some of the currently available international design codes for steel frames integrated into the SAP2000 is: AISC 360-05, AISC 360-10, AISC 360-16, AISC ASD 89, AISC LRFD 93, API RP-2A LRFD 1997, API RP-2A WSD 2000, API RP2A-WSD 2014, AS 4100-1998, ASCE 10-97, BS 5950-2000, Chinese 2010, Chinese 2018, CSA S16-09, CSA S16-14, Eurocode 3-2005, IS 800:2007, KBC 2009, KBC 2016, Norsok N-004 2013, NTC 2008, NTC 2018, NZS 3404-1997, SP 16.13330.2011, SP 16.13330.2017. Through the SAP2000 API, these design specifications (and also the ones that will be integrated in the future to SAP2000) are completely available to external programs like SOPS. Therefore, SOPS requires no updates or modifications to implement newly announced design specifications into its package.

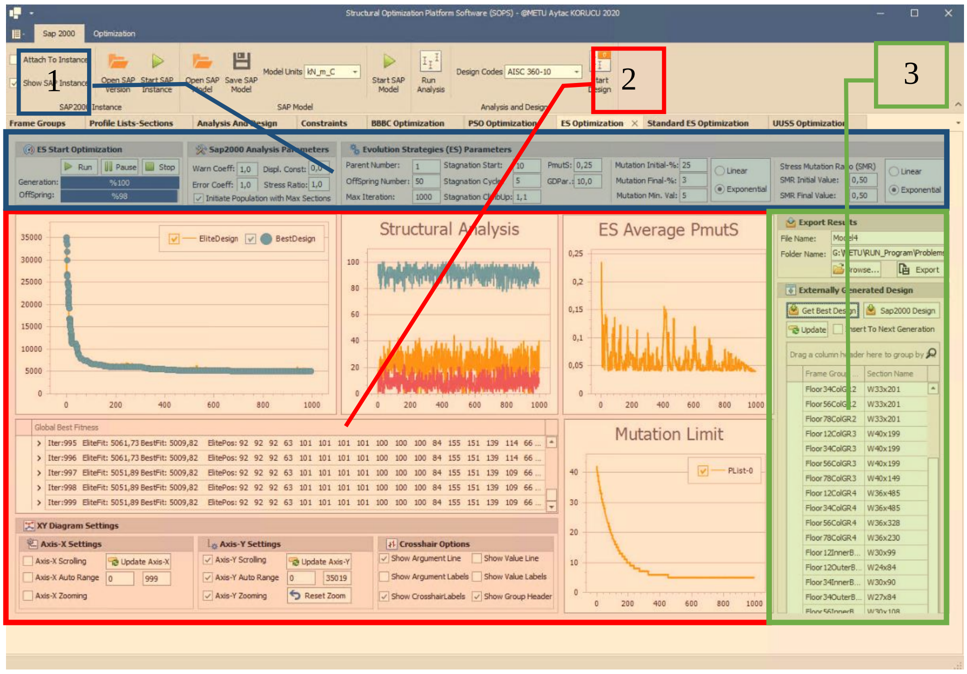


Design Codes AISC 360-10

Figure 4.11 International design codes implemented in SOPS

4.3.4 SOPS Optimization Modules

Structural Optimization Platform Software (SOPS) consists of optimization modules isolated from each other through the modular architecture of SOPS. Therefore, any modifications applied to one optimization technique do not affect the others. SOPS currently contains the optimization techniques such as Evolution Strategies (μ,λ), Evolution Strategies (μ+λ), Particle Swarm Optimization (PSO), Big Bang Big Crunch Optimization (BB-BC), and Guided Evolution Strategy (GES) (Figure 4.12).



Model Units: kN_m_C

Design Codes: AISC 360-10

Generation: %100

OffSpring: %98

Warn Coeff: 1,0

Displ. Const: 0,0

Error Coeff: 1,0

Stress Ratio: 1,0

☑ Initiate Population with Max Sections

Parent Number: 1

OffSpring Number: 50

Max Iteration: 1000

Stagnation Start: 10

Stagnation Cycle: 5

Stagnation ClimbUp: 1,1

PmutS: 0,25

GDPar.: 10,0

Mutation Initial-%: 25

Mutation Final-%: 3

Mutation Min. Val: 5

○ Linear

● Exponential

Stress Mutation Ratio (SMR)

SMR Initial Value: 0,50

SMR Final Value: 0,50

○ Linear

● Exponential

```
> Iter:995  EliteFit: 5061,73 BestFit: 5009,82   ElitePos: 92  92  92  63  101  101  101  101  100  100  100  84  155  151  139  114  66 ...
> Iter:996  EliteFit: 5061,73 BestFit: 5009,82   ElitePos: 92  92  92  63  101  101  101  101  100  100  100  84  155  151  139  114  66 ...
> Iter:997  EliteFit: 5051,89 BestFit: 5009,82   ElitePos: 92  92  92  63  101  101  101  101  100  100  100  84  155  151  139  109  66 ...
> Iter:998  EliteFit: 5051,89 BestFit: 5009,82   ElitePos: 92  92  92  63  101  101  101  101  100  100  100  84  155  151  139  109  66 ...
> Iter:999  EliteFit: 5051,89 BestFit: 5009,82   ElitePos: 92  92  92  63  101  101  101  101  100  100  100  84  155  151  139  109  66 ...
```

☐ Axis-X Scrolling

☐ Axis-X Auto Range 0 999

☐ Axis-X Zooming

☑ Axis-Y Scrolling

☑ Axis-Y Auto Range 0 35019

☑ Axis-Y Zooming

☑ Show Argument Line ☐ Show Value Line

☐ Show Argument Labels ☐ Show Value Labels

☑ Show CrosshairLabels ☑ Show Group Header

File Name: Mo[illegible]el4

Folder Name: G:\[illegible]ETU\RUN_Program\Problems

☐ [illegible]nsert To Next Generation

Frame Grou...	Section Name
Floor34ColG[illegible]2	W33x201
Floor56ColG[illegible]2	W33x201
Floor78ColGR2	W33x201
Floor12ColGR3	W40x199
Floor34ColGR3	W40x199
Floor56ColGR3	W40x199
Floor78ColGR3	W40x149
Floor12ColGR4	W36x485
Floor34ColGR4	W36x485
Floor56ColGR4	W36x328
Floor78ColGR4	W36x230
Floor12InnerB...	W30x99
Floor12OuterB...	W24x84
Floor34InnerB...	W30x90
Floor34OuterB...	W27x84
Floor56InnerB	W30x108

Figure 4.12 SOPS - Optimization Modules.

The optimization module of SOPS is divided into three main parts regarding their functions in the optimization process, as shown in Figure 4.12. In part one, located at the top bar of the form and demonstrated in blue color, parameters of optimization techniques are set through the input fields. Moreover, control of the optimization process, such as start-stop-pause, can also be managed via this part. Default values of the optimization techniques' parameters are provided in the input fields. Hence, unless any custom or problem-specific parameters are necessary, the optimization process can be initiated directly with the start optimization command button.

The real-time monitoring of the optimization process is presented in the second part, located in the middle of the form and demonstrated in red color. The monitoring form of each optimization technique is slightly differentiated from each other due to the techniques' specific demands. However, the best designs of the structural optimization problem obtained at each iteration are presented in the monitoring form for all optimization techniques. Performance of the optimization techniques such as convergence rate and stagnation durations can be easily observed in real-time while the optimization process continues in the background.

Finally, the procedures related to the outcomes of the optimization process are handled in part three, located on the right side of the form and demonstrated in green color. In this part, design variables of the best design found in the optimization process so far can be examined. Optimization interim outcomes generated so far can also be exported to the file without waiting until the end of the optimization process.

CHAPTER 5

NUMERICAL EXAMPLES

In this chapter, numerical examples are presented related to the optimum design of steel frames using the optimization methods discussed in Chapter 3. In fact, the chapter consists of three parts. In the first part, numerical examples are conducted using two test frames for determining the optimal parameter settings of the Guided Evolution Strategy (GES) technique proposed in this study. In the second part, several design instances of ordinary moment-resisting steel frames are studied using the proposed GES technique as well as other metaheuristic search algorithms; such as particle swarm optimization (PSO), exponential big bang big crunch (EBB-BC), (μ,λ)-evolution strategy (ES) and ($\mu+\lambda$)-evolutionary strategy (ES). This way, the performance of the GES algorithm is compared to those of the other metaheuristic search techniques in terms of the quality of the optimum solution as well as the speed of convergence to an optimum solution (i.e. convergence rate). It is important to emphasize that amongst thousands of different metaheuristics, these techniques are deliberately selected and used for comparison based on their successful applications reported formerly in the literature related to the problems of interest. In the last part, some real-world steel structures that have been formerly designed by practicing engineers using a traditional design procedure are optimized and redesigned by the GES algorithm and metaheuristic search techniques. This way the practical applicability of optimization techniques for real-world problems is illustrated. Besides, the amount of material savings that could be achieved through a design optimization procedure is identified with respect to a traditional design procedure implemented by a practicing engineer.

5.1 Sensitivity Analysis of GES Parameters

The numerical studies for parameter sensitivity analysis of the GES technique are carried out in conjunction with two test frames; namely, a 4-story, 160-member steel frame shown in Figure 5.1 and an 8-story, 584-member steel frame shown in Figure 5.2. Since strength constraints are mainly handled by the GES for guiding the search process, first of all, the 4-story steel frame (the first test frame) is subjected to strength constraints only; that is, no displacement constraints are imposed. Next, the same fame is studied under the case for which both strength and inter-story drift constraints are imposed together. Finally, the numerical experiments are performed with the 8-story steel frame (the second test frame) to determine optimal parameter values of the technique under the increased effect (dominance) of the inter-story drift constraints.

The two critical parameters, which significantly affect the performance of the GES algorithm, are Guided Mutation Limit (GML) and Guided Mutation Ratio (GMR). In the following, the effect of GML parameter is examined first, and then the GMR parameter is studied afterward.

5.1.1 Problems Used in Sensitivity Analysis of GES Parameters

The 4-story, 160-member steel frame shown in Figure 5.1 and the 8-story, 584-member steel frame shown in Figure 5.2 are used as the test examples for identifying the numerical performance of the algorithm under different values of the Guided Mutation Limit (GML) and Guided Mutation Ratio (GMR) parameters.

The 4-story, 160-member steel frame consists of 96 beams and 64 columns and has a 3.5 m story height. The stability of the frame in both directions is provided through using moment-resisting connections. For practical fabrication requirements, the 160 members of the frame are collected under 12 member groups. The columns are grouped into four different sizing variables in a plan level as the corner, inner, side plane x-z, and side plane y-z columns. The columns are also grouped every two stories along the height of the frame such that the column groups over the first two (story 1-2) and last two (story 3-4) stories are required to have the same cross-section, resulting in a total of eight column member groups for the frame. Similarly, the beams are grouped into two distinct sizing variables in a plan level as inner and outer beams. The beams are also grouped every two stories along the height of the frame, yielding a total of four different beam member groups.

The 8-story steel frame shown in Figure 5.2 is composed of 584 structural members collected under 24 member groups and has a 3.5 m story height. The member grouping is applied in both plan and elevation levels. At elevation level, the structural members are grouped every two stories along the height of the frame. At the plan level, the columns are collected under 4 different groups, as highlighted in Figure 5.3, and the beams are collected under two groups as inner and outer beams, as demonstrated in Figure 5.4. Therefore, a total of 16 column groups and eight beam groups (24 sizing design variables) are defined for the 584-member steel frame example.

Load combinations used in both 4 Story, 160-member steel frame and 8-story, 584-member steel frame in this study are specified as in Section 2.3 in ASCE 7-10 specification, and they are listed below:

(1) 1.4D

(2) 1.2D + 1.6L

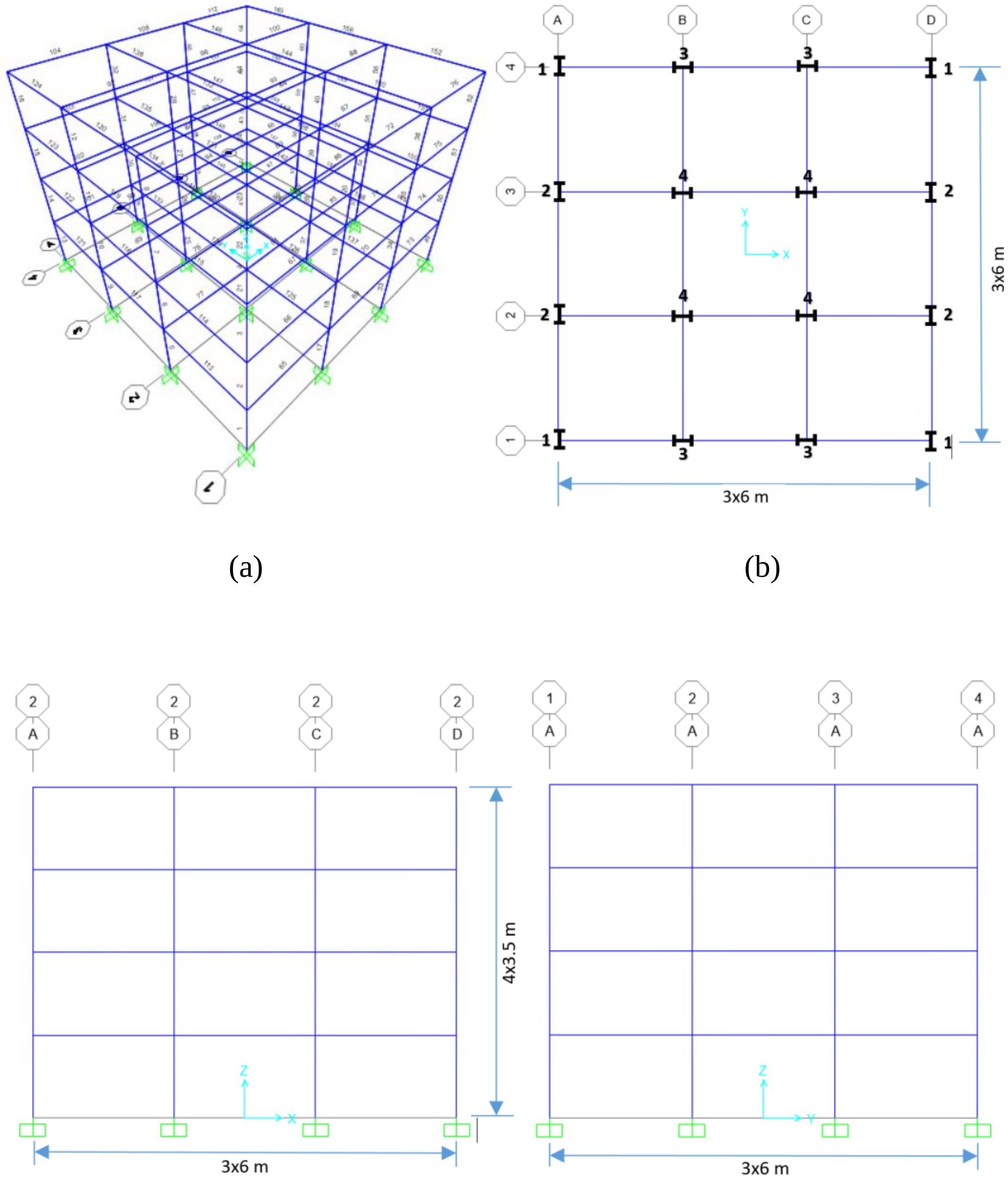
(3) 1.2D + 1.0L ± 1.0E

(4) 0.9D ± 1.0E

where D, L, and E represent the dead, live, and earthquake loads, respectively.

The design gravity loads considered for these two steel frames are given in Table 5.1. The earthquake loads are calculated using the seismic coefficients listed in Table 5.2 and applied to the steel frames in accordance with the equivalent lateral load procedure in ASCE 7-10. The amplified inter-story drift is restricted to 2% of story height. The structural elements are sized under the provisions of AISC 360-

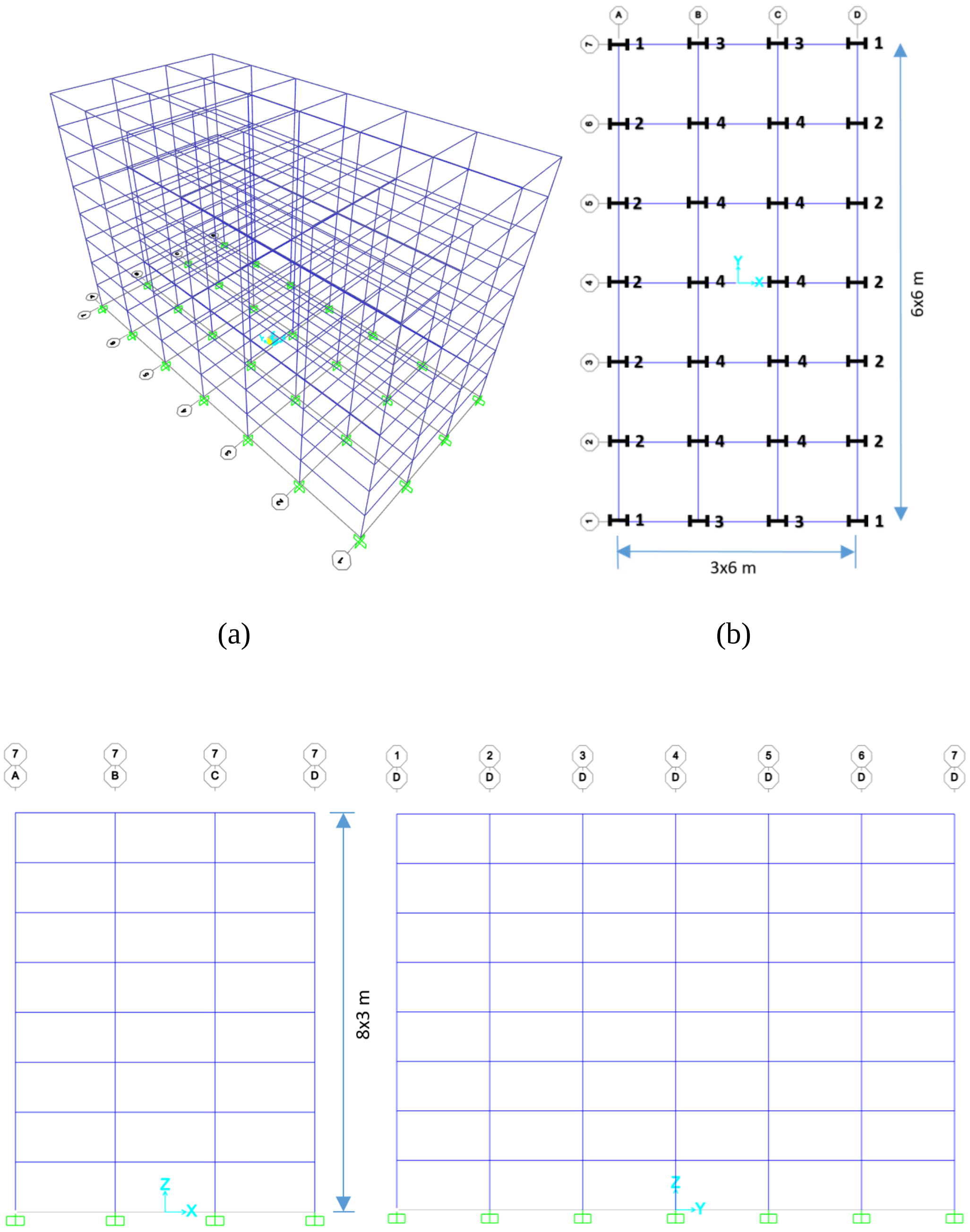
10 design specifications. Moreover, the design constraints are enforced as defined in detail in Chapter 2.



(a) (b)

(c) (d)

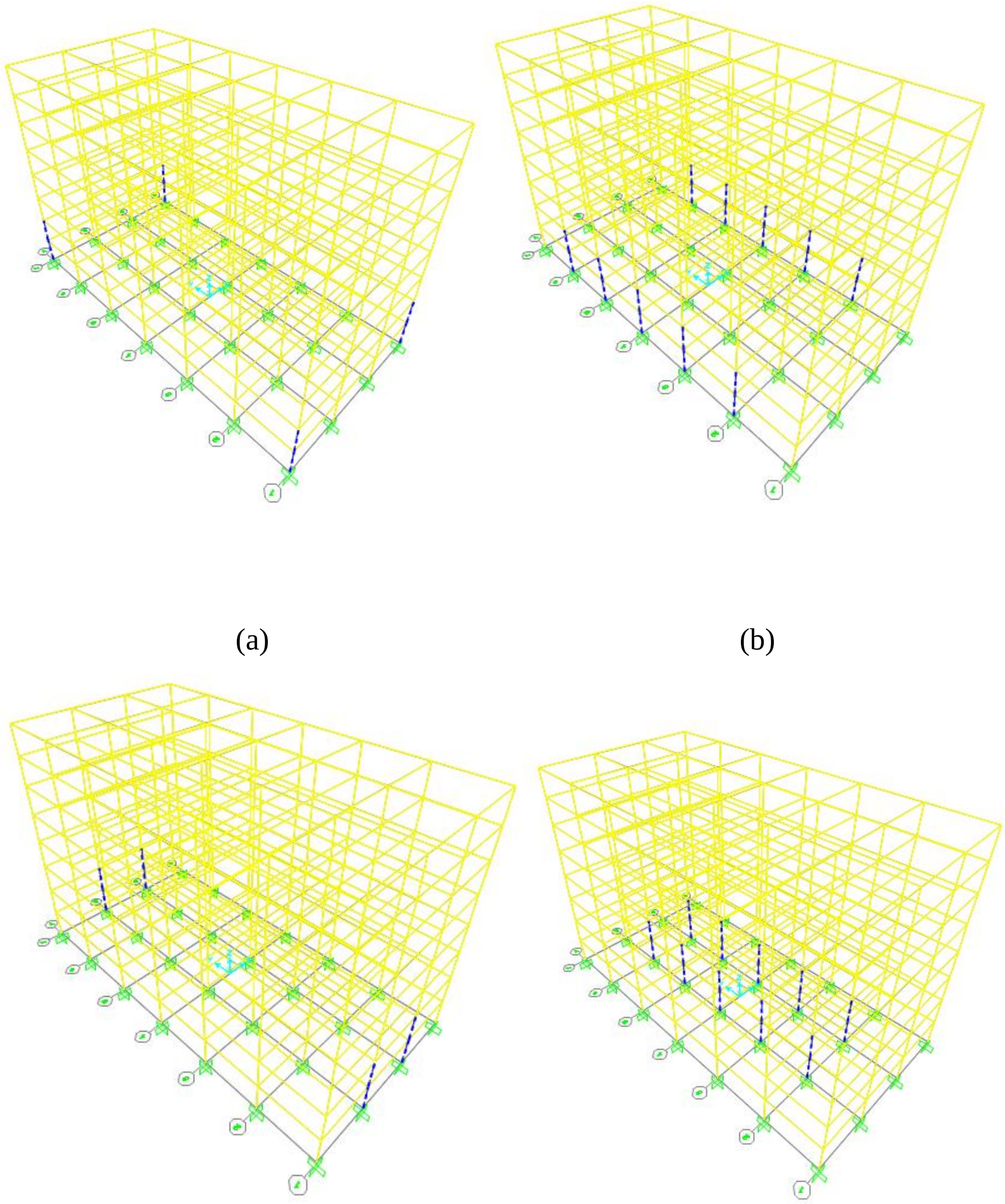
Figure 5.1 160-member steel frame (the first test example): (a) 3-D view, (b) plan view, (c) side view in x-z plane (d) side view in y-z plane



(a) (b)

(c) (d)

Figure 5.2 584-member steel frame (the second test example): (a) 3-D view, (b) plan view, (c) side view in x-z plane, (d) side view in y-z plane



(a) (b)

(c) (d)

Figure 5.3 The column member groups in the first two stories of the 584-member steel frame: (a) corner columns, (b) outer y-side columns, (c) outer x-side columns, (d) inner columns.

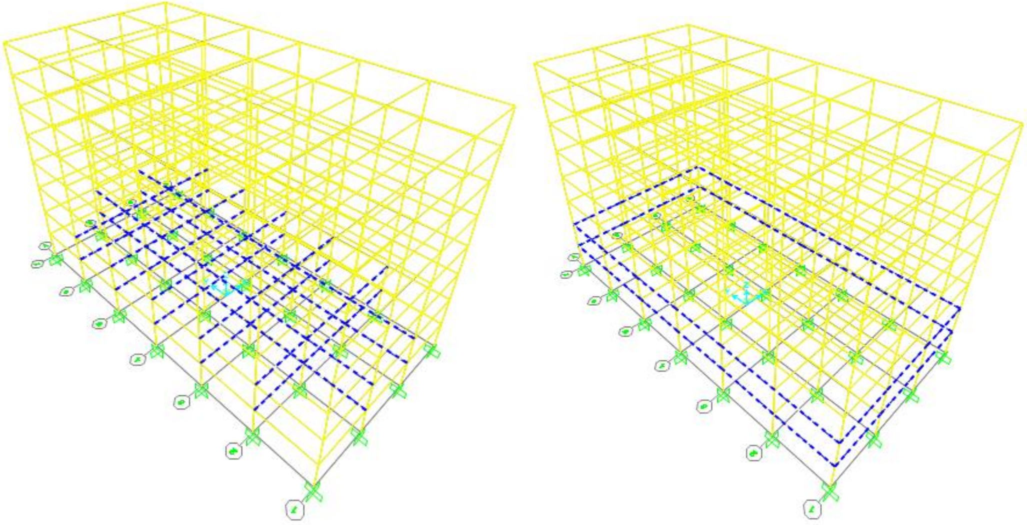

(a) (b)

Figure 5.4 Beam member groups in the first two stories of the 584-member steel frame: (a) inner beams, (b) outer beams.

Table 5.1 Gravity design loads for both 160-member steel frame and 584-member steel frame

Load Type	**Uniformly Distributed Load**	
	Interior Beams (kN/m)	**Exterior Beams (kN/m)**
Dead Load (D)	24	12
Lİve Load (L)	12	6

Table 5.2 Earthquake seismic coefficients for both 160-member steel frame and 584-member steel frame

Seismic Coefficients	
0.2 Sec Spectral Accel, Ss	2.29
0.1 Sec Spectral Accel, S_1	0.869
Long-Period Transition Period	8
Site Class	D
Site Coefficient, F_a	1
Site Coefficient, F_v	1.5
S_{DS}=(2/3) F_a S_s	1.5267
S_{D1}=(2/3) F_v S_1	0.869

5.1.2 Sensitivity Analysis for Guided Mutation Limit (GML) Parameter

The effect of GML parameter is examined under the availability or dominance of various types of constraints. In the following, the 4-story frame, 160-member frame is optimized first under strength constraints using different values of GML parameter (Case-A). Next, the same frame is studied for the case in which both the displacement and strength constraints are imposed together (Case-B). Finally, in order to investigate the variation in the performance of the algorithm for problems under the dominance of displacement constraints, the 8-story, 584-member frame is sized for the minimum weight using different values of the GML parameter (Case-C).

5.1.2.1 Case-A: 4-Story Frame without Story-Drift Constraint

In Case-A, the optimum design of the 4-story, 160-member frame is studied using different values of GML parameter, while the GMR parameter is fixed at a value of 0.50 (50 %). Initially, the constant values of GML parameters are analyzed with six different values of GML, as shown in Table 5.3. Next, dynamic values of the GML are studied where the parameter is decreased by using a timely decreasing function as the iterations proceed. It is important to note that as the GML parameter decreases, exploitative search characteristics of the algorithm are enhanced, while its explorative search potentials are somewhat reduced. The effects of transition rate from explorative to exploitative search are investigated using two different timely decreasing functions; namely linearly decreasing and exponentially decreasing functions. In a linearly decreasing case, there is a smooth transition from explorative to exploitative search. On the other hand, the change is rather sharp and rapid for the exponentially decreasing case. Both linearly decreasing and exponentially decreasing values of the GML parameter are analyzed with 16 cases each, as shown in Table 5.4 and Table 5.5, respectively.

5.1.2.1.1 Constant Guided Mutation Limit

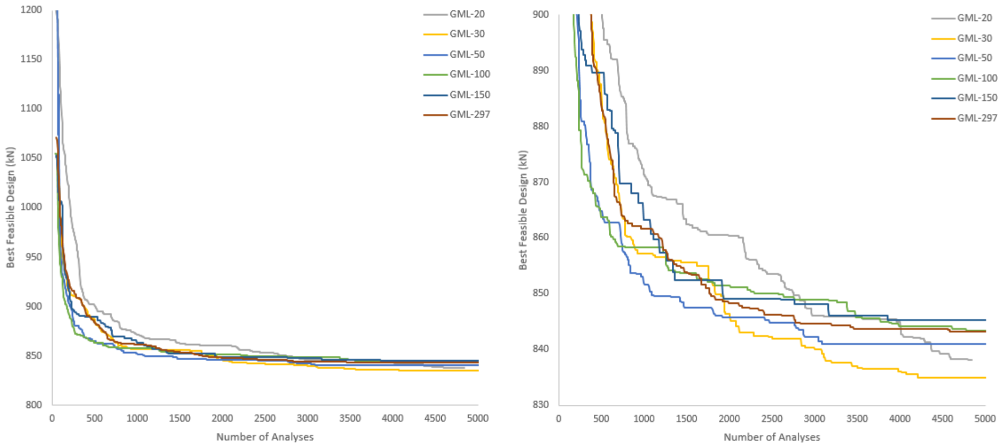
Initially, constant values of the GML parameter are examined. Due to the stochastic nature of the GES technique, a total of ten independent runs are performed with the algorithm for each value of the GML to capture statistically meaningful data. In the optimum design problem, the discrete set used to size the design variables consist of 297 AISC standard wide-flange ready steel sections, and thus $N_s = 297$. Accordingly, the GML values tested here are 20 (7% of N_s), 30 (10% of N_s), 50 (17% of N_s), 100 (34% of N_s), 150 (50% of N_s) and 297 (100% of N_s, i.e., no mutation limit) as shown in Table 5.3. The optimization histories (convergence curves), which show the variation of the best feasible design obtained so far in the optimization process versus the number of structural analyses performed are plotted in Figure 5.5 for the mentioned values of the GML parameter. It is important to mention that each curve in Figure 5.5 is obtained by averaging the results of ten independent runs performed with a particular value of the GML parameter.

In general, Figure 5.5 indicates that at the initial stages of the optimization process, the fastest convergence is exhibited by the guided mutation limit values of 50 (17% of N_s) and 100 (34% of N_s), whereas toward the later stages of the optimization process, mutation limit value of 30 (10% of N_s) demonstrates better performance and yields a better solution. Therefore, it can be deduced that at the initial stages of the optimization process, sufficiently high values of the GML parameter are more successful in terms of a rapid and broader exploration of favorable design regions, which in fact leads to an earlier discovery of the better solutions. On the other hand, low values of the GML parameter – despite their slow rate of convergence - seem to benefit more from the advantages of an exploitative search towards the later stages of the process. Indeed, these results rationalize and form a basis for the use of a timely varying GML parameter, instead of operating the parameter at a constant value the whole time. Therefore, in the following various options are tested for timely varying use of the GML parameter in which the GML parameter is decreased progressively from a large value at the beginning towards a smaller value at the end. For the generality of the results obtained, both, the initial and final values of the

GML parameter are expressed as a certain percentage (ratio) of the N_s, which refers to the number of sections in the discrete set.

Table 5.3 Constant values of the GML parameter considered for the sensitivity analysis.

GML	Initial GML	Final GML	Function	GMR	Run
GML_20	20 (7% of N_s)	20 (7% of N_s)	Constant	50%	10
GML_30	30 (10% of N_s)	30 (10% of N_s)	Constant	50%	10
GML_50	50 (17% of N_s)	50 (17% of N_s)	Constant	50%	10
GML_100	100 (34% of N_s)	100 (34% of N_s)	Constant	50%	10
GML_150	150 (50% of N_s)	150 (50% of N_s)	Constant	50%	10
GML_297	297 (100% of N_s)	297 (100% of N_s)	Constant	50%	10



(a) (b)

Figure 5.5 Optimization histories for the 160-member steel frame under different constant values of the guided mutation limit (GML) parameter, (a) full-scale view (b) zoomed view

5.1.2.1.2 Linearly Decreasing Guided Mutation Limit

In this section, a dynamic use of the GML based on a linearly decreasing value of the parameter will be investigated. Again, the numerical experiments are conducted with the 160-member steel frame subjected to strength constraints, where the optimum designs of the frame are sought using sixteen different combinations of the GML parameter, listed in Table 5.4 and plotted in Figure 5.6. Similarly, the discrete set used to size the design variables consist of 297 AISC standard wide-flange ready steel sections, and thus $N_s = 297$. In the study the initial values of the GML parameter are set to 100% of N_s, 75% of N_s, 50% of N_s, 25% of N_s, 15% of N_s and 10% of N_s, whereas the final values are assigned to 10%, 5, and 3% of N_s. Various combinations of initial and final values of the GML parameter are tested. For example, the so-called GML_50-10-LNR scheme in Table 5.4 represents a combination in which the initial value of GML is set to 149 (50% of 297 sections), and the parameter is decreased to its final value of 30 (10% of 297 sections) linearly at each iteration. As stated before, the second parameter of GES is set constant for all GML cases and taken as 0.5 (i.e., %50), which means that half of the offspring individuals are mutated according to the guided mutation scheme, while the other half are mutated by stochastic mutation scheme. The GES algorithm is executed ten times independently for each combination considered, and the optimization histories corresponding to each case are plotted in Figure 5.7, after averaging the results of ten independent runs pertaining to that case.

The results shown in Figure 5.7 demonstrates that the GML 15-5-LNR scheme (the GML parameter linearly decreasing from 15% of N_s to 5% of N_s) has the fastest convergence rate through the first 2000 analyses, while the GML_10-3-LNR scheme (the GML parameter linearly decreasing from 10% of N_s to 3% of N_s) yields a better solution than all others after 5000 analyses. Since larger design transitions are allowed in the GML_15-5-LNR scheme at the early stages of the optimization process, its explorative search potential results in faster convergence, compared to the GML_10-3-LNR. However, an exploitation-based search, which aims to improve the already obtained designs for reaching better solutions, becomes more essential at later stages, and thereby the GML_10-3-LNR scheme yields better solutions.

Table 5.4 Various combinations of the linearly decreasing GML parameter considered for the sensitivity analysis

GML	Initial GML	Final GML	Function	GMR	Run
GML_100-10-LNR	100% of N_s	10% of N_s	Linear	50%	10
GML_75-10-LNR	75% of N_s	10% of N_s	Linear	50%	10
GML_50-10-LNR	50% of N_s	10% of N_s	Linear	50%	10
GML_25-10-LNR	25% of N_s	10% of N_s	Linear	50%	10
GML_100-5-LNR	100% of N_s	5% of N_s	Linear	50%	10
GML_75-5-LNR	75% of N_s	5% of N_s	Linear	50%	10
GML_50-5-LNR	50% of N_s	5% of N_s	Linear	50%	10
GML_25-5-LNR	25% of N_s	5% of N_s	Linear	50%	10
GML_15-5-LNR	15% of N_s	5% of N_s	Linear	50%	10
GML_10-5-LNR	10% of N_s	5% of N_s	Linear	50%	10
GML_100-3-LNR	100% of N_s	3% of N_s	Linear	50%	10
GML_75-3-LNR	75% of N_s	3% of N_s	Linear	50%	10
GML_50-3-LNR	50% of N_s	3% of N_s	Linear	50%	10
GML_25-3-LNR	25% of N_s	3% of N_s	Linear	50%	10
GML_15-3-LNR	15% of N_s	3% of N_s	Linear	50%	10
GML_10-3-LNR	10% of N_s	3% of N_s	Linear	50%	10

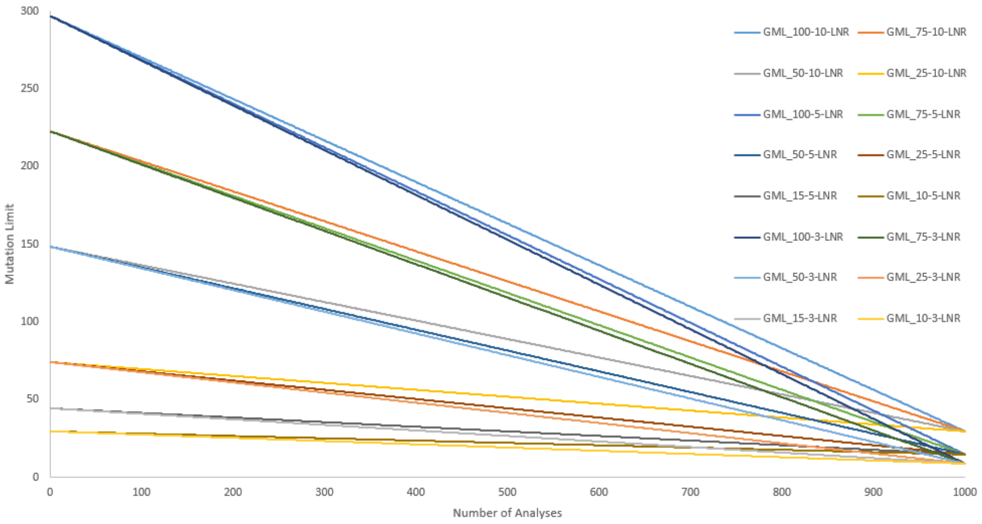
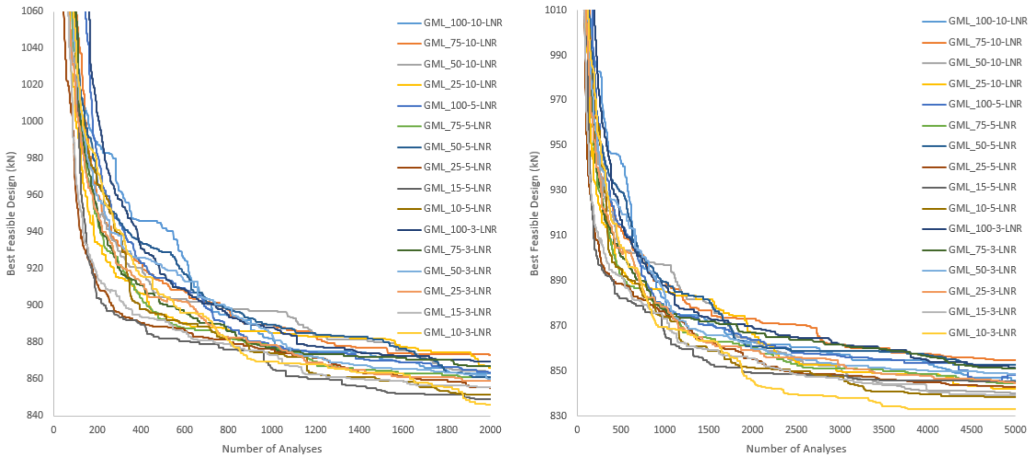


Figure 5.6 The variations in the linearly decreasing GML parameters for 16 different combinations considered.



(a) (b)

Figure 5.7 Optimization histories for the 160-member steel frame under different linearly varying guided mutation limit (GML) schemes: (a) for 2000 analyses, (b) for 5000 analyses

5.1.2.1.3 Exponentially Decreasing Guided Mutation Limit

In this section, a dynamic use of the GML based on exponentially decreasing value of the parameter will be evaluated. The 160-member steel frame (shown in Figure 5.1) that is used for constant and linearly decreasing GML cases is also studied here in the same line. The initial values of the GML parameter are set to 100% of N_s, 75% of N_s, 50% of N_s, 25% of N_s, 15% of N_s and 10% of N_s, whereas the final values are assigned to 10%, 5, and 3% of N_s. For the sake of comparison, the same 16 combinations that are used for a linearly decreasing GML parameter are also tested here under exponentially decreasing value of the parameter. These combinations are listed in Table 5.5 and also plotted in Figure 5.8. The GES algorithm is executed ten times independently for each combination considered, and the optimization histories corresponding to each case are plotted in Figure 5.7, after averaging the results of ten independent runs pertaining to that case.

The results shown in Figure 5.9 demonstrates that the GML_15-5-EXP scheme (the GML parameter exponentially decreasing from 15% of N_s to 5% of N_s) has the fastest convergence rate through the first 1000 analyses, while the GML_75-3-EXP scheme (the GML exponentially decreasing from 75% of N_s to 3% of N_s) yields a better solution than all others after 5000 analyses.

The numerical studies performed so far indicate that the GML parameter has a very strong effect on establishing a tradeoff between the exploration and exploitation capabilities of the GES algorithm. Therefore, a proper selection of this parameter value enables an optimization process that benefits from these search features at the maximum rate for reaching a good optimum in the design space. In the following section, the designs obtained for the 160-member steel frame with various schemes of the GML parameter are compared to determine the most effective value/scheme of the parameter.

Table 5.5 Various combinations of the exponentially decreasing GML parameter considered for the sensitivity analysis

GML	Initial GML	Final GML	Function	GMR	Run
GML_100-10-EXP	100% of Ns	10% of Ns	Exponential	50%	10
GML_75-10-EXP	75% of Ns	10% of Ns	Exponential	50%	10
GML_50-10-EXP	50% of Ns	10% of Ns	Exponential	50%	10
GML_25-10-EXP	25% of Ns	10% of Ns	Exponential	50%	10
GML_100-5-EXP	100% of Ns	5% of Ns	Exponential	50%	10
GML_75-5-EXP	75% of Ns	5% of Ns	Exponential	50%	10
GML_50-5-EXP	50% of Ns	5% of Ns	Exponential	50%	10
GML_25-5-EXP	25% of Ns	5% of Ns	Exponential	50%	10
GML_15-5-EXP	15% of Ns	5% of Ns	Exponential	50%	10
GML_10-5-EXP	10% of Ns	5% of Ns	Exponential	50%	10
GML_100-3-EXP	100% of Ns	3% of Ns	Exponential	50%	10
GML_75-3-EXP	75% of Ns	3% of Ns	Exponential	50%	10
GML_50-3-EXP	50% of Ns	3% of Ns	Exponential	50%	10
GML_25-3-EXP	25% of Ns	3% of Ns	Exponential	50%	10
GML_15-3-EXP	15% of Ns	3% of Ns	Exponential	50%	10
GML_10-3-EXP	10% of Ns	3% of Ns	Exponential	50%	10

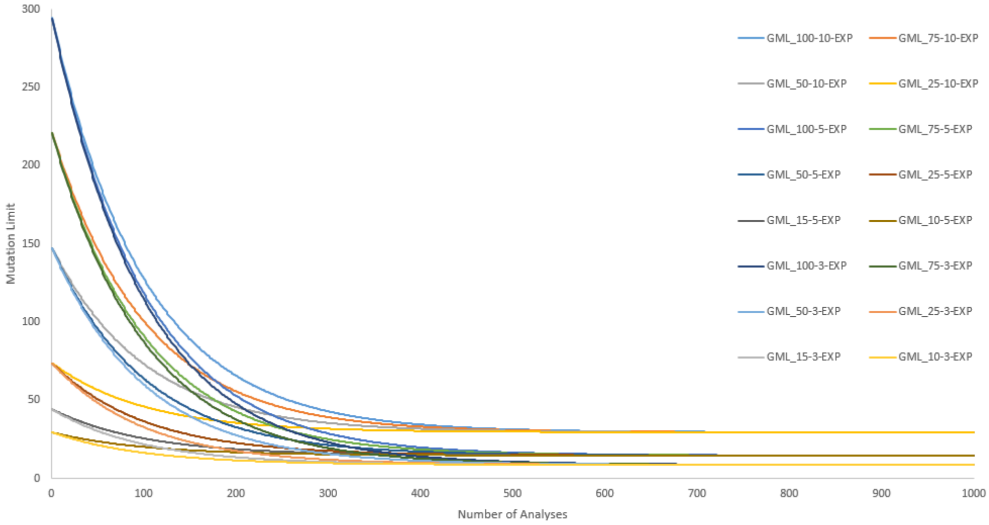
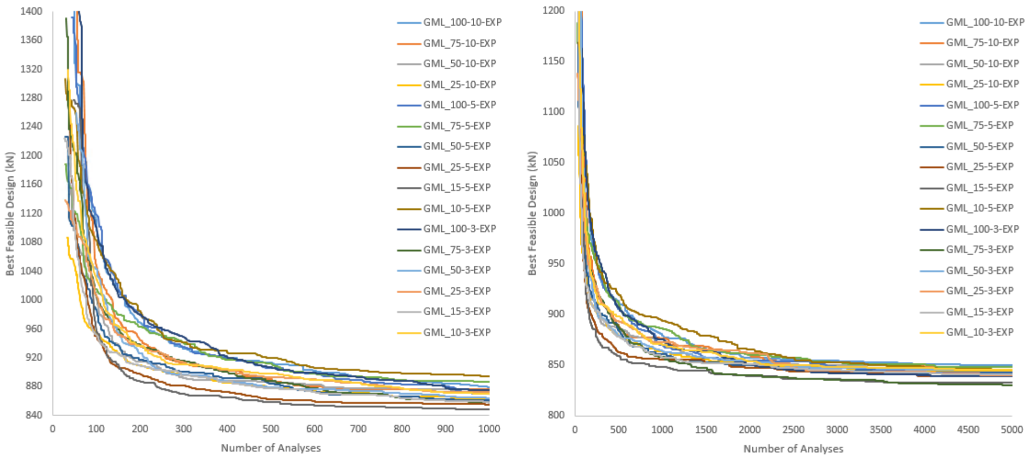


Figure 5.8 The variations in the exponentially decreasing GML parameters for 16 different combinations considered.



(a) (b)

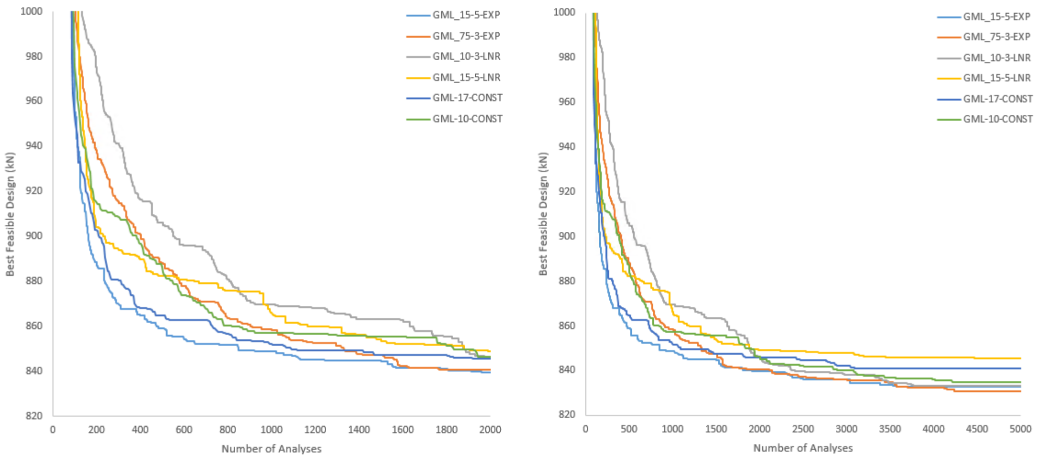
Figure 5.9 Optimization histories for the 160-member steel frame under different exponentially varying guided mutation limit (GML) schemes: (a) for 1000 analyses (b) for 5000 analyses

5.1.2.1.4 Evaluation of Constant, Linear, and Exponential Guided Mutation Limit Results with Random Initial Population

In the previous sections, the use of the Guided Mutation Limit (GML) parameter is tested and experimented based on a static and dynamic manipulation of the parameter. The results obtained under three different manipulations of the GML parameter are combined in Figure 5.10 for the two most successful performances of the algorithm in each case (referred to as constant, linearly decreasing, and exponentially decreasing cases) and their optimization histories are presented.

The results shown in Figure 5.10 reveals that the GML_15-5-EXP scheme (the GML parameter exponentially decreasing from 15% of N_s to 5% of N_s) has the fastest convergence rate up to 2000 analyses, while the GML_75-3-EXP scheme (the GML parameter exponentially decreasing from 75% of N_s to 3% of N_s) yields a better solution than all others after 5000 analyses. The results also indicate that an exponentially decreasing GML parameter is superior to both constant and linearly decreasing GML.

A satisfactory balance between exploration and exploitation search features cannot be established effectively, in case the GML parameter is set to a constant value through the optimization process. High constant values of the GML parameter take advantage of extensive exploration of the design space at the early stages of the optimization process, yet they suffer from an insufficient exploitative search at later stages. Similarly, low constant values of the GML parameter are not able to explore the global search domain sufficiently and experience the local optima traps. On the other hand, it has been observed that both exploration and exploitation search potentials are better utilized by the algorithm when a timely varying use of the parameter is enabled. Favorable design regions and promising solutions discovered by means of explorative search potentials possessed at the early stages of the optimization process are benefited later for reaching even better solutions at later states through exploitative search abilities. The results also reveal that the rate of transition from explorative to exploitative search affects the quality of optimum solution attained by the algorithm. Rather sharp and rapid transition as in the exponentially decreasing GML parameter results in a faster convergence rate than smooth transition like linearly decreasing GML parameter. Therefore, the exponentially decreasing value of the GML will be used and further investigated in the following studies.



(a) (b)

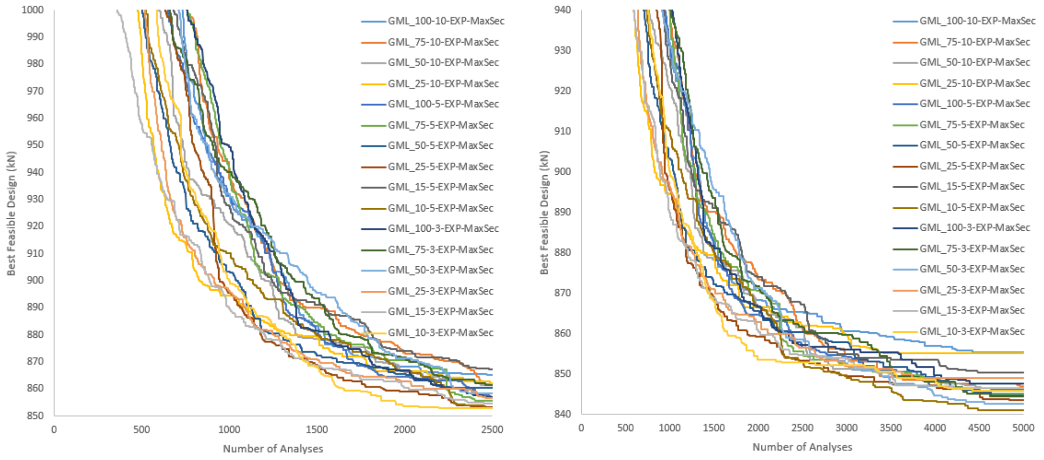
Figure 5.10 Optimization histories for the 160-member steel frame corresponding to the best two performances of the algorithm under three different manipulations of the GML parameter: (a) for 2000 analyses, (b) for 5000 analyses

5.1.2.1.5 Exponentially Decreasing GML with Maximum Section Initial Population

In the previous sections, the effect of the Guided Mutation Limit (GML) parameter is examined using an initial parent individual that is generated at random. Instead of initializing the first parent individual randomly, if all the design variables (member groups) are initially assigned to the largest possible section in the discrete set, then all the strength constraints can be satisfied by the first parent individual. In this section, this strategy is followed and its effect is investigated in conjunction with the exponentially decreasing value of the GML parameter. The 160-member steel frame (shown in Figure 5.1) that is used for an initial parent individual that is generated randomly is also studied here in the same line. The initial values of the GML parameter are set to 100% of N_s, 75% of N_s, 50% of N_s, 25% of N_s, 15% of N_s and 10% of N_s, whereas the final values are assigned to 10%, 5, and 3% of N_s. For the sake of comparison, the same 16 exponentially decreasing value of the GML combinations that are used for a randomly generated initial parent individual are also tested here. These combinations are listed in Table 5.5 and also plotted in Figure 5.8. The GES algorithm is executed ten times independently for each combination considered, and the optimization histories corresponding to each case are plotted in Figure 5.11, after averaging the results of ten independent runs pertaining to that case.

The results shown in Figure 5.11 demonstrates that the GML_10-3-EXP-MaxSec scheme (the GML parameter exponentially decreasing from 10% of N_s to 3% of N_s with maximum section initialization) has the fastest convergence rate through the first 2500 analyses, while the GML_10-5-EXP-MaxSec scheme (the GML exponentially decreasing from 10% of N_s to 5% of N_s with maximum section initialization) yields a better solution than all others after 5000 analyses. Unlike the random initialization, the maximum section initialization case generally starts the optimization process with a solution satisfying the strength constraints. Therefore, balanced exploration and exploitation potential in the early stages demonstrates a faster convergence rate in the maximum section initialization situation than high exploration power.

In the following section, the designs obtained for the 160-member steel frame with various schemes of the GML parameter with two initial parent individual generation types (randomly and maximum-section) are compared to determine the most effective value/scheme of the parameter.



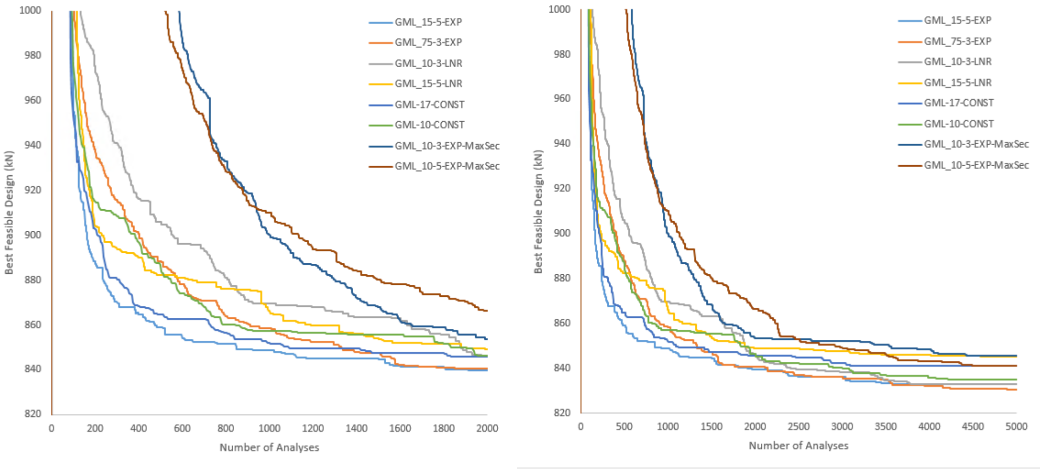
(a) (b)

Figure 5.11 Optimization histories for the 160-member steel frame under different exponentially varying guided mutation limit (GML) schemes and initial parent individual generated by using the maximum section: (a) for 2500 analyses (b) for 5000 analyses

5.1.2.1.6 Comparison of Guided Mutation Limit Results

In the previous sections, the use of the Guided Mutation Limit (GML) parameter is tested and experimented based on a static and dynamic manipulation of the parameter. The results obtained under three different manipulations of the GML parameter and randomly generated initial parent individual are combined in Figure 5.10 for the two most successful performances of the algorithm in each case (referred to as constant, linearly decreasing, and exponentially decreasing cases) and their optimization histories are presented. This section compares these results with results obtained under the different initial parent individual (generated by using maximum sections) condition. The results are combined in Figure 5.12 for the two most successful performances of the algorithm in each case.

The results shown in Figure 5.12 demonstrate that GML under a randomly generated initial individual condition has a faster convergence rate than GML under a largest possible section initialized parent individual case. Since the optimization process has the same initial starting point in the initial parent individual condition that is generated with maximum sections, seeking the entire global search domain takes more time and suffers more from local optima traps. Therefore, compared to random initialization case, it has a slower convergence rate, especially at the early stages of the optimization process. Although, due to the power of exploitation, the difference gets lesser at the final stages of the optimization process after excessive structural analysis, still randomly initialized case yields better optimum solutions.



(a) (b)

Figure 5.12 Optimization histories for the 160-member steel frame corresponding to the best two performances of the algorithm under three different manipulations of the GML parameter and initial parent individual that is generated randomly and generated by assigning the largest possible section: (a) for 2000 analyses, (b) for 5000 analyses

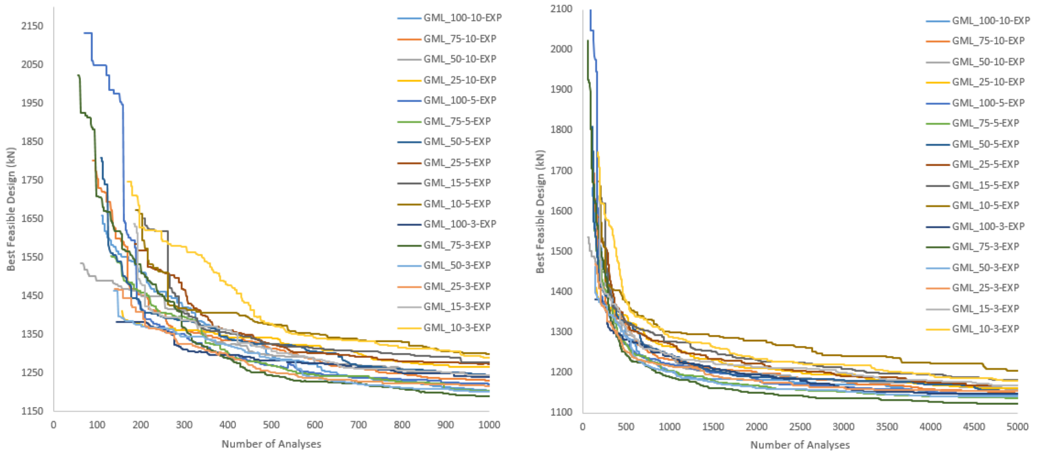
5.1.2.2 Case-B: 4-Story Frame with Story-Drift constraint

In the last part, The Guided Mutation Limit (GML) parameter of GES is examined by studying the steel frame subjected to strength constraints only; that is, no displacement constraints are imposed. In this section, the same frame shown in Figure 5.1 is studied under the case for which both strength and inter-story drift constraints are imposed together.

In Case-B, a dynamic use of the GML based on exponentially decreasing value of the parameter will be evaluated, while the GMR parameter is fixed at a value of 0.50 (50%) as in the Case-A. The initial values of the GML parameter are set to 100% of N_s, 75% of N_s, 50% of N_s, 25% of N_s, 15% of N_s and 10% of N_s, whereas the final values are assigned to 10%, 5, and 3% of N_s. For the sake of comparison, the same 16 combinations that are used for Case-A are also tested here under exponentially decreasing value of the parameter. These combinations are listed in Table 5.5 and also plotted in Figure 5.8. The GES algorithm is executed ten times independently for each combination considered, and the optimization histories corresponding to each case are plotted in Figure 5.13, after averaging the results of ten independent runs pertaining to that case.

Similar to Case-A, both a randomly generated initial individual condition and a largest possible section initialized parent individual condition are investigated. Firstly, a randomly generated initial individual condition is examined in the following section.

The results shown in Figure 5.13 demonstrates that GML_75-3-EXP (the GML parameter exponentially decreasing from 75% of N_s to 3% of N_s) has the fastest convergence rate and yields a better solution than the other ones through all stages of the optimization process. In case-A, at the early stages of the optimization process, the GES technique benefits from intensive guidance, resulting in faster convergence. However, introducing displacement constraints (inter-story drift constraints) to the design problem reduce the guidance efficiency. Since the GES guidance is mainly based on the strength values of the members (DCR of members), it is natural to utilize guidance more effectively in optimization problems where the design is governed mainly by the strength of its members. However, the GES technique uses more exploration potential to overcome the handicap due to the presence of displacement constraints. Therefore, at the initial stages of the optimization process, the high value of GML (GML_75-3-EXP, i.e., % 75 of profile list) allows the GES algorithm to mutate design variables sufficiently large to reduce the displacement constraints interference.

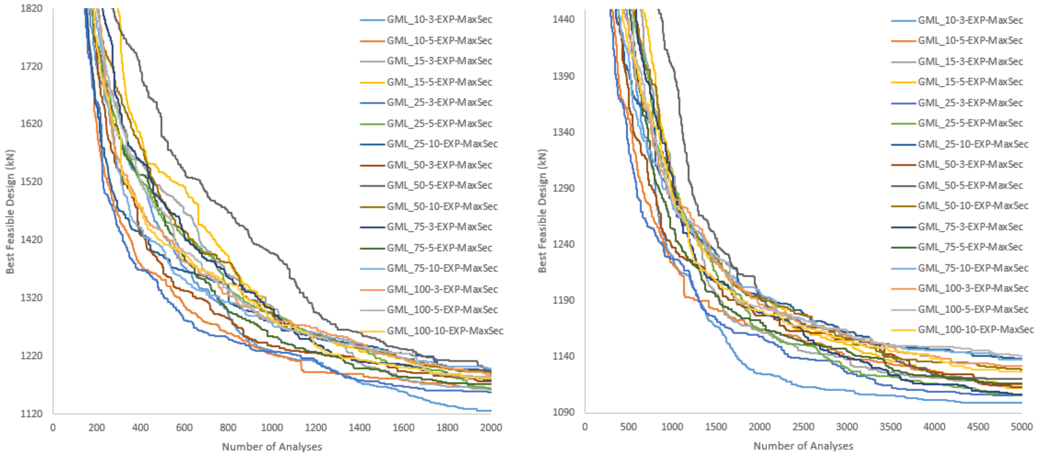


(a) (b)

Figure 5.13 Case-B: Optimization histories for the 160-member steel frame under different exponentially varying guided mutation limit (GML) schemes under a randomly generated initial individual condition: (a) for 1000 analyses (b) for 5000 analyses

In the previous section, the effect of the Guided Mutation Limit (GML) parameter is examined using an initial parent individual that is generated at random. In this part, it is investigated under a largest possible section initialized parent individual condition in conjunction with the exponentially decreasing pattern. The 160-member steel frame (shown in Figure 5.1) that is used previously is also studied here in the same line. The initial values of the GML parameter are set to 100% of N_s, 75% of N_s, 50% of N_s, 25% of N_s, 15% of N_s and 10% of N_s, whereas the final values are assigned to 10%, 5%, and 3% of N_s. For the sake of comparison, the same 16 combinations that are used for the case of randomly generated initial parent individual are also tested here. These combinations are listed in Table 5.5 and also plotted in Figure 5.8. The GES algorithm is executed ten times independently for each combination considered, and the optimization histories corresponding to each case are plotted in Figure 5.14, after averaging the results of ten independent runs pertaining to that case.

The results shown in Figure 5.14 demonstrates that GML_25-3-EXP-MaxSec (the GML parameter exponentially decreasing from 25% of N_s to 3% of N_s with maximum section initialization) has the fastest convergence rate at the initial stage of the optimization process, while GML_10-3-EXP-MaxSec yields better solution and faster convergence rates towards the final stage.



(a) (b)

Figure 5.14 Optimization histories for the 160-member steel frame under different exponentially varying guided mutation limit (GML) schemes and initial parent individual generated by using the maximum section: (a) for 2000 analyses (b) for 5000 analyses

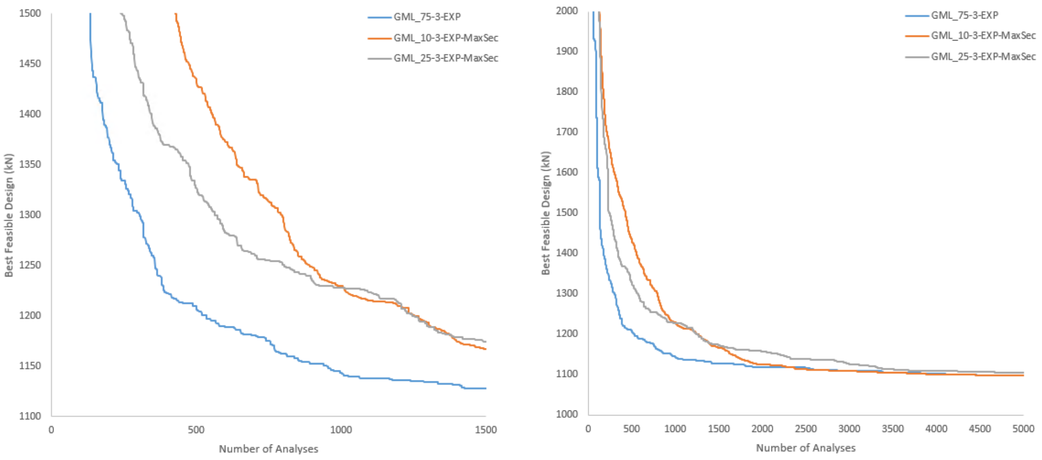
In the following section, the designs obtained for the 160-member steel frame with various schemes of the GML parameter with two initial parent individual generation types (randomly and maximum-section) are compared to determine the most effective value/scheme of the parameter.

The results obtained under these two initial parent individual generation case are combined in Figure 5.15 for the most successful performances of the algorithm in each case (GML_75-3-EXP, GML_25-3-EXP-MaxSec, ML_10-3-EXP-MaxSec) and their optimization histories are presented. It is worth to mention that the Guided Mutation Parameter designated as GML_75-3-EXP is under the randomly generated initial parent individual conditions while the other two GML parameters

(GML_25-3-EXP-MaxSec, ML_10-3-EXP-MaxSec) are under the initial parent individual generated by using largest possible sections in the profile list.

The results shown in Figure 5.15 demonstrate that GML_75-3-EXP (the GML parameter exponentially decreasing from 75% of N_s to 3% of N_s) under a randomly generated initial parent individual condition has the faster convergence rate up to 1500 analyses compared to the initial parent individual generated by using largest possible sections (GML_25-3-EXP-MaxSec, GML_10-3-EXP-MaxSec). On the other hand, after excessive analysis, both initial parent individual generation types (randomly and maximum-section) can reach a comparable quality of solutions.

To summarize, GML_75-3-EXP (the GML parameter exponentially decreasing from 75% of N_s to 3% of N_s) under a randomly generated initial parent individual condition has the faster convergence rate and yields better solutions than GML values/schemes under the initial parent individual generated by using largest possible sections condition.



(a) (b)

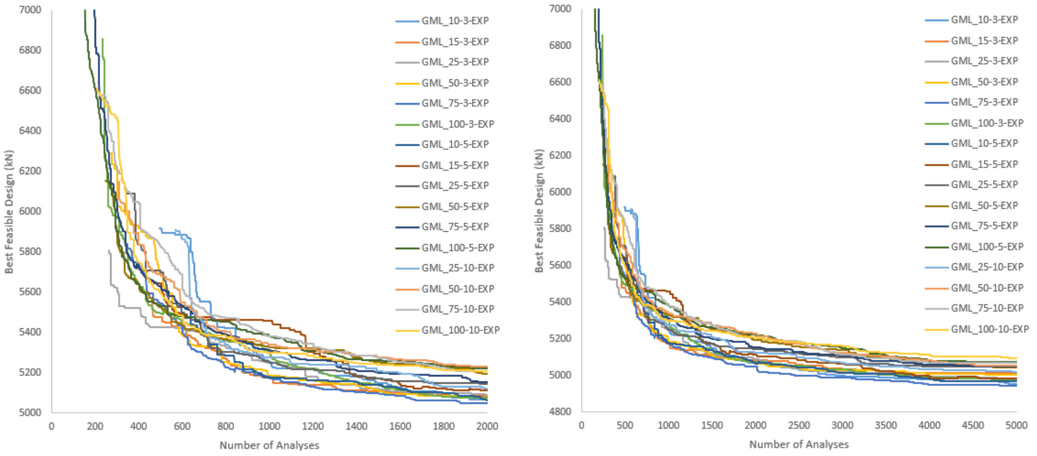
Figure 5.15 Optimization histories for the 160-member steel frame corresponding to the best performances of the algorithm under two initial parent individual conditions (generated randomly and generated by assigning the largest possible section): (a) for 1500 analyses, (b) for 3000 analyses

5.1.2.3 Case-C: 8-Story Frame with Story-Drift constraint

In the previous sections, Guided Mutation Limit (GML) parameter is examined firstly by studying the 4-story steel frame subjected to strength constraints only; that is, no displacement constraints are imposed (Case-A). Next, the same fame is studied under the case for which both strength and inter-story drift constraints are imposed together (Case-B). In this section, the numerical experiments are performed with the 8-story steel frame (shown in Figure 5.2) to determine optimal parameter values of the technique under the increased effect (dominance) of the inter-story drift constraints (Case-C).

In Case-C, a dynamic use of the GML based on exponentially decreasing value of the parameter will be evaluated, while the GMR parameter is fixed at a value of 0.50 (50 %) as in the Case-A and Case-B. The initial values of the GML parameter are set to 100% of N_s, 75% of N_s, 50% of N_s, 25% of N_s, 15% of N_s and 10% of N_s, whereas the final values are assigned to 10%, 5%, and 3% of N_s. For the sake of comparison, the same 16 combinations that are used for other two cases are also tested here under exponentially decreasing value of the parameter. These combinations are listed in Table 5.5 and also plotted in Figure 5.8. The GES algorithm is executed ten times independently for each combination considered, and the optimization histories corresponding to each case are plotted in Figure 5.16, after averaging the results of ten independent runs pertaining to that case.

The results shown in Figure 5.16 demonstrates that GML_75-3-EXP (the GML parameter exponentially decreasing from 75% of N_s to 3% of N_s) has the fastest convergence rate and yields a better solution than the other ones through all stages of the optimization process. This is the same GML pattern (GML_75-3-EXP) obtained for the Case-B. Therefore, introducing more displacement dominance to the optimization problem can still be better handled with the GML_75-3-EXP mutation limit pattern.



(a) (b)

Figure 5.16 Optimization histories for the 584-member steel frame subjected to inter-story drift constraint under different exponentially varying guided mutation limit (GML) schemes: (a) for 2000 analyses (b) for 5000 analyses

5.1.3 Sensitivity Analysis of Guided Mutation Ratio (GMR) Parameter

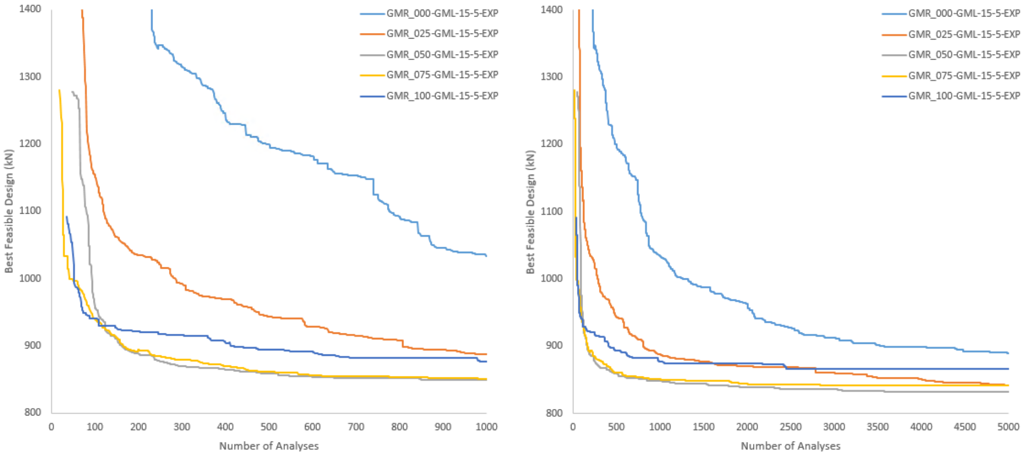
The Guided Mutation Limit (GML) parameter of GES is examined in the previous parts. Another parameter of the GES, called Guided Mutation Ratio (GMR), is studied in this section. GMR is evaluated using the same steel frames (shown in Figure 5.1 and Figure 5.2) employed in assessing the GML parameter. In previous sections, while the evaluation of GML parameter, GMR parameter is taken as 50% for all runs. This section will examine the most effective values/schemes of the GML parameter obtained previously with the different GMR values such as 0%, 25%, 75%, and 100%. As stated before, GMR is the ratio of guided offspring, and hence, for example, 25% means that one-quarter of the offspring individuals are mutated according to the guided mutation scheme while the others are mutated by stochastic mutation scheme.

5.1.3.1 Sensitivity Analysis of GMR Parameter without Story-Drift Constraint

In the previous sections, the use of the Guided Mutation Limit (GML) parameter is tested and experimented based on a static and dynamic manipulation of the parameter. The results obtained under three different manipulations of the GML parameter (referred to as constant, linearly decreasing, and exponentially decreasing cases) demonstrate that GML_15-5-EXP scheme (the GML parameter exponentially decreasing from 15% of N_s to 5% of N_s) has the fastest convergence rate at the early stages of the optimization process, while the GML_75-3-EXP scheme (the GML parameter exponentially decreasing from 75% of N_s to 3% of N_s) yields a better solution than all others. Therefore, GMR values other than 50% are tested for these two prominent GML values in this section.

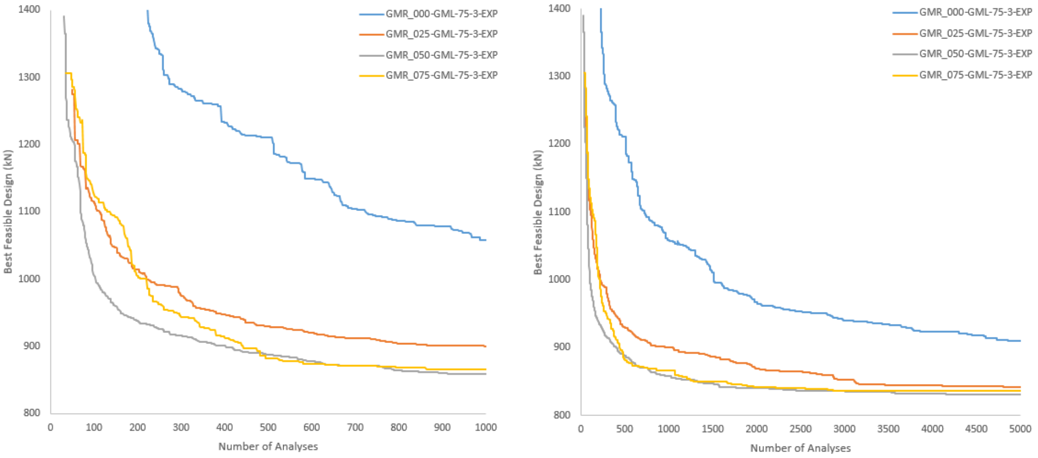
The Guided Mutation Ratio (GMR) parameter is examined by studying the steel frame subjected to strength constraints only; that is, no displacement constraints are imposed. The 160-member steel frame (shown in Figure 5.1) that is used for the Guided Mutation Limit (GML) parameter is also studied here in the same line. Guided Mutation Ratio (GMR) values tested here are 0% (no GMR, i.e., all of the offspring individuals are mutated by stochastic mutation scheme), 25%, 50%, 75%, and 100% (all of the offspring individuals are mutated according to the guided mutation scheme). The GES algorithm is executed ten times independently for GML_15-5-EXP and GML_75-3-EXP, and the optimization histories are plotted in Figure 5.17 and Figure 5.18 respectively, after averaging the results of ten independent runs pertaining to corresponding case.

The results presented in Figure 5.17 and Figure 5.18 demonstrate that the GMR value of 50% has the fastest convergence rate and yields a better solution than all others. Therefore, this GMR value is recommended in GES to design optimization problems without displacement constraints dominance. In the following section, the effects of displacement constraints on the performance of the GMR parameter are investigated by introducing inter-story drift constraints.



(a) (b)

Figure 5.17 Optimization histories for the 160-member steel frame under exponentially varying (GML_15-5-EXP) guided mutation limit (GML) schemes and five different Guided Mutation Ratio (GMR) values: (a) for 1000 analyses (b) for 5000 analyses



(a) (b)

Figure 5.18 Optimization histories for the 160-member steel frame under exponentially varying (GML_75-3-EXP) guided mutation limit (GML) schemes and four different Guided Mutation Ratio (GMR) values: (a) for 1000 analyses (b) for 5000 analyses

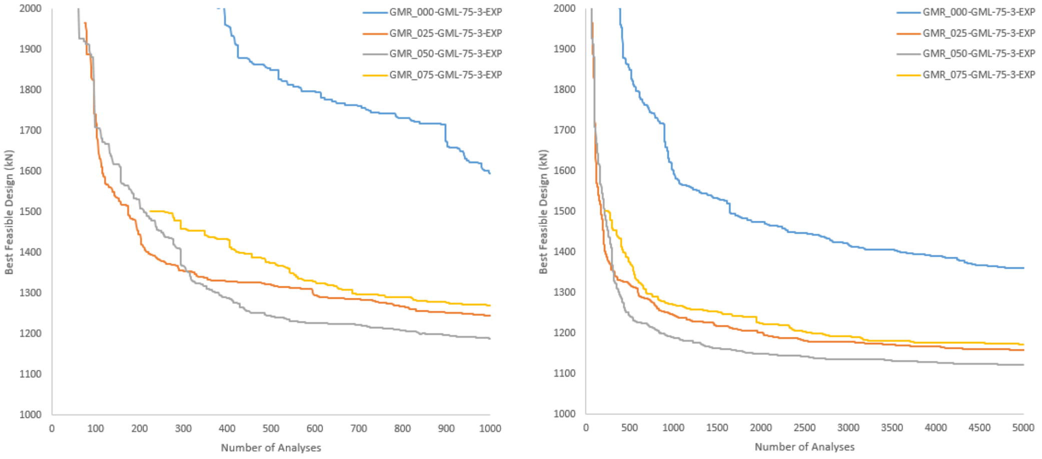
5.1.3.2 Sensitivity Analysis of GMR Parameter with Story-Drift Constraint

In the last part, The Guided Mutation Ratio (GMR) parameter is examined by studying the steel frame subjected to strength constraints only; that is, no displacement constraints are imposed. In this section, first the same frame shown in Figure 5.1 is studied under the case for which both strength and inter-story drift constraints are imposed together. Next, the numerical experiments are performed with the 8-story steel frame (shown in Figure 5.2) to determine optimal GMR parameter values under the increased effect (dominance) of the inter-story drift constraints.

In the previous sections, the use of the Guided Mutation Limit (GML) parameter is tested and experimented based on a static and dynamic manipulation of the parameter under the case for which both strength and inter-story drift constraints are imposed together and the constant value of GMR parameter (%50). The results obtained under three different manipulations of the GML parameter (referred to as constant, linearly decreasing, and exponentially decreasing cases) demonstrate that GML_75-3-EXP scheme (the GML parameter exponentially decreasing from 75% of N_s to 3% of N_s) has the fastest convergence rate and yields a better solution than all others. In this section, GMR values other than 50% are tested for this most effective value/scheme of this GML parameter.

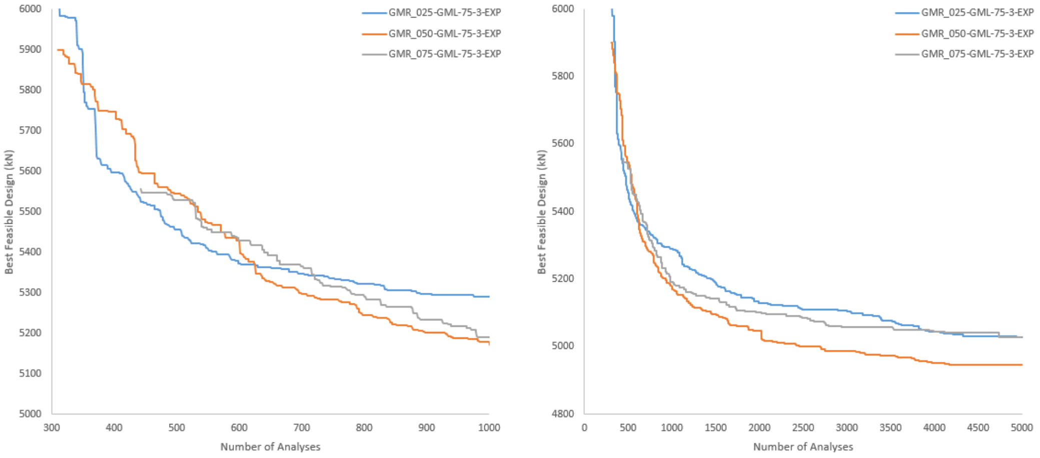
Guided Mutation Ratio (GMR) values tested here are 0% (no GMR, i.e., all of the offspring individuals are mutated by stochastic mutation scheme), 25%, 50% and 75%. First, the GES algorithm is executed ten times independently under the constant GML_75-3-EXP parameter for each GMR combination using 160-member steel frame shown in Figure 5.1. Next the same runs are repeated for the 584-member steel frame shown in Figure 5.2 and the optimization histories are plotted in Figure 5.19 and Figure 5.20 respectively, after averaging the results of ten independent runs pertaining to corresponding case.

The results presented in Figure 5.19 and Figure 5.20 demonstrate that the GMR value of 50% has the fastest convergence rate and yields a better solution than all others. Therefore, this GMR value (50%) is recommended for the solution of the optimization problems.



(a) (b)

Figure 5.19 Optimization histories for the 160-member steel frame under exponentially varying (GML_75-3-EXP) guided mutation limit (GML) schemes and four different Guided Mutation Ratio (GMR) values: (a) for 1000 analyses (b) for 5000 analyses



(a) (b)

Figure 5.20 Optimization histories for the 584-member steel frame under exponentially varying (GML_75-3-EXP) guided mutation limit (GML) schemes and four different Guided Mutation Ratio (GMR) values: (a) for 1000 analyses (b) for 5000 analyses

5.1.4 Discussion of Parameter Sensitivity Analysis Results

In the previous parts, the numerical studies for parameter sensitivity analysis of the GES technique are carried out in conjunction with two test frames. The parameters of the GES, namely Guided Mutation Limit (GML) and Guided Mutation Ratio (GMR), are examined for their most efficient values before applying the GES algorithm to the optimization problems. The results show that the GES algorithm is quite sensitive to its parameters. GML parameter regulates the exploration and exploitation balance of the algorithm. Exploration is the capability to explore distinct regions of the design search space for discovering an acceptable optimum solution, while exploitation is the capability to focus the search nearby a searching region for refining a better solution. Therefore, these two prominent characteristics should be balanced in an optimization algorithm to get a more suitable solution. Guided Mutation Limit parameter can be utilized to balance search direction between global/local search, and higher mutation limit forces algorithm towards to global direction while the smaller value entails local search direction.

At the initial stages of the optimization process, sufficiently high values of the GML parameter are more successful in terms of a rapid and broader exploration of favorable design regions, which in fact leads to an earlier discovery of the better solutions. On the other hand, low values of the GML parameter – despite their slow rate of convergence - seem to benefit more from the advantages of an exploitative search towards the later stages of the process. Therefore, timely varying use of the GML parameter in which the GML parameter is decreased progressively from a large value at the beginning towards a smaller value at the end outcomes a more suitable solution.

Although the relative rate of local/global convergence performance is supplied by GML, the intensity of the GES technique's guidance is determined by the Guided Mutation Ratio (GMR) parameter. To keep the exploration potential and also to avoid the local optima traps, not all offspring are mutated according to the guided mutation scheme. To benefit from guidance potential of the algorithm, guidance intensity should be controlled by adjusting the GMR parameter. The high value of GMR results in reduced exploration potential and local optima traps, while low values of GMR cannot benefit from guidance sufficiently. Therefore, the GMR parameter should be regulated in the GES technique to get a more suitable solution.

The outcomes of the sensitivity analysis of the GES parameters indicate that for the frame, whose design is governed mainly by the strength constraints, GML_15-5-EXP scheme (the GML parameter exponentially decreasing from 15% of N_s to 5% of N_s) has the fastest convergence. However, for the design of frames, which are not only governed by the strength constraints but also displacement limitations, GML_75-3-EXP scheme (the GML parameter exponentially decreasing from 75% of N_s to 3% of N_s) has the fastest convergence rate and yields a better solution. The sensitivity analysis results for the GMR parameter of GES also demonstrate that GMR value of %50 (i.e., guiding half of the offspring) has the fastest convergence rate and yields a more qualified solution. Moreover, the results also demonstrate that GES under a randomly generated initial individual condition has a faster convergence rate and yields more qualified solutions than a largest possible section initialized parent individual case.

5.2 Performance Comparison of the GES Algorithm

In the previous section, the optimal parameter settings of the GES algorithm are examined in the context of structural optimization design problems. In this section, the optimum design problems used for the sensitivity analysis of the GES technique are also solved using some robust metaheuristic algorithms, namely, particle swarm optimization (PSO), exponential big bang big crunch (EBB-BC; enhanced version of the big-bang big crunch algorithm), and the two multi-membered variants of the Evolution Strategies, namely (μ,λ)-ES and (μ+λ)-ES. The solutions previously obtained to these problems using the GES technique are compared with those of the aforementioned metaheuristic algorithms.

The metaheuristic algorithms are implemented with their recommended parameter settings given in Hasançebi et al. (2009). However, it is worth mentioning that the performance of the metaheuristic search techniques might be quite sensitive to their parameter settings and implementation details, and unfortunately, it is not feasible or practical to tune each algorithm to reach its best performance for all the optimum design problems considered in this study. The parameters of the metaheuristic algorithms used to solve optimization design problems in this study are listed in Table 5.6. For each metaheuristic technique, ten independent runs are executed using these parameters over a predefined number of analyses.

In all optimization design examples, the material properties of the steel are taken as follows: yield strength (F_y) = 50 ksi (~ 345 MPa), modulus of elasticity (E) = 29000 ksi (~ 200 GPa) and unit weight (ρ) = 490 lb/ft^3 (~ 76.97 kN/m^3). The design loads (gravity and earthquake) and load combinations are presented in the previous section, and thus will not be repeated here.

Table 5.6 Parameter setting of the metaheuristic algorithms (EBB-BC, PSO, ES) used in this study.

Optimization Methods	**Parameter Values**
EBB-BC	$\mu = 50$, $\lambda = 1.0$, $\alpha=0.25$
PSO	$\mu = 50$, $\Delta t = 1.0$, $c_1=1.5$, $c_2=1.5$, $w=0.5$
(μ,λ)-ES	$\mu = 10$, $\lambda = 50$, $p^{(0)} = 0.25$, $\psi^{(0)} = 10$
$(\mu+\lambda)$-ES	$\mu = 50$, $\lambda = 50$, $p^{(0)} = 0.25$, $\psi^{(0)} = 10$

5.2.1 Optimization Problem 1-A: 160-Member Steel Frame under Strength Constraints

The 160-member steel frame (shown in Figure 5.1) under strength constraints refers to the first design example which is used to make a performance comparison between the GES algorithm and the metaheuristic search techniques. In the previous section, this problem has been studied extensively for sensitivity analysis of the GES algorithm by testing a large set of parameter values. The best solutions to the problem are produced when an exponentially decreasing value of the GML parameter is implemented with the GML_15-5-EXP scheme and also the guided mutation ratio parameter is set as 0.5 (i.e., %50), which implies that half of the offspring are probabilistically mutated according to the guided mutation scheme whereas the other half are mutated according to the stochastic mutation scheme.

Since metaheuristic search techniques have a stochastic nature, each run may end up with a different solution, especially for large and complex structural optimization design problems. Therefore, the problem is solved by executing ten independent runs with each metaheuristic search technique to get statistically meaningful data for performance evaluation. In Table 5.7, the minimum weight designs (best feasible solutions) achieved for the frame are presented in terms of the best, worst, and average (mean) solution attained with each optimization technique up to some selected stages of the optimization process; namely 500, 1000, 2000, 3000 number of structural analyses. Moreover, to assess the reliability of a single run with any technique, the standard deviation (STD) and coefficient of variation (CV) values are also calculated and presented in the related table. In general, a low standard deviation is interpreted as increased reliability of the technique for independent runs, whereas a large standard deviation indicates scarcity and inconsistency of solutions obtained in different runs with the same technique.

As seen in Table 5.7, the GES algorithm demonstrates a superior performance than the metaheuristic search techniques for all stages of the optimization process. In Figure 5.21, the (average) variation of the best design weight against the number of analyses (i.e., average convergence curve) is plotted for each optimization technique, after averaging the results of all runs performed with a particular method. Similarly, in Figure 5.22, convergence curves are plotted considering only the best performance of the methods for the problem of interest. The best performance of a method is referred to as a single run in which the method produces the least design weight for the problem. Rapid convergence characteristics of the GES algorithm can also be clearly observed from these two figures, with respect to the metaheuristics search techniques.

A comparison of the minimum weight designs (the best feasible designs) obtained using each optimization method is carried out in Table 5.8, where the section designations assigned to all member groups are indicated along with the weight of the resulting design. Among the results obtained with various optimization algorithms shown in this table, the GES yields the least design weight for the 160-member steel frame, which is 805.16 kN. The design weights that the other algorithms obtained are 843.86 kN by (μ+λ)-ES, 887.96 kN by (μ,λ)-ES, 837.18 kN by EBB-BC, and 861.14 kN by PSO.

It is also worth mentioning that amongst the two multi-membered variants of Evolution Strategies, Upper Bound Strategy (UBS) is implemented more efficiently with the (μ+λ)-ES variant (Kazemzadeh Azad et al. 2013a, b) since the elitism rule is strictly applied in this variant, the design population at any iteration consists of μ individuals that represent the best solutions captured from the beginning of the optimization process, whereas in (μ,λ)-ES variant, the design population is regenerated at every iteration from the best μ solutions out of λ individuals. Accordingly, an iteration-based implementation of the elitism rule in (μ,λ)-ES requires more individuals to be analyzed during the evaluation stage, resulting in a reduced convergence rate of the algorithm. Hence, this variant of the algorithm is not used for the solution of other design examples.

Table 5.9 presents the DCR (demand-to-capacity ratio) values for member groups in the optimum designs of the 160-member steel frame produced with the aforementioned optimization methods. Since only the strength design constraints are applied in this design problem (i.e., no inter-story drift constrained is enforced), the design variables (cross-sections of the member groups) are controlled by DCRs. Accordingly, DCR values in Table 5.9 mostly appear to be very close to their maximum value of 1.0. However, it is also worth mentioning that geometric constraints between beam and column members presented in Chapter 2 are also applied in this problem. The fact that DCR values for some member groups are not assigned to values around 1.0 in the optimum designs can be attributed to the use of geometric constraints. In order to satisfy these constraints, larger sections might be assigned to member groups, resulting in lower DCR values for some member groups.

Table 5.7 Optimization statistics for the 160-member steel frame design example (under strength constraints).

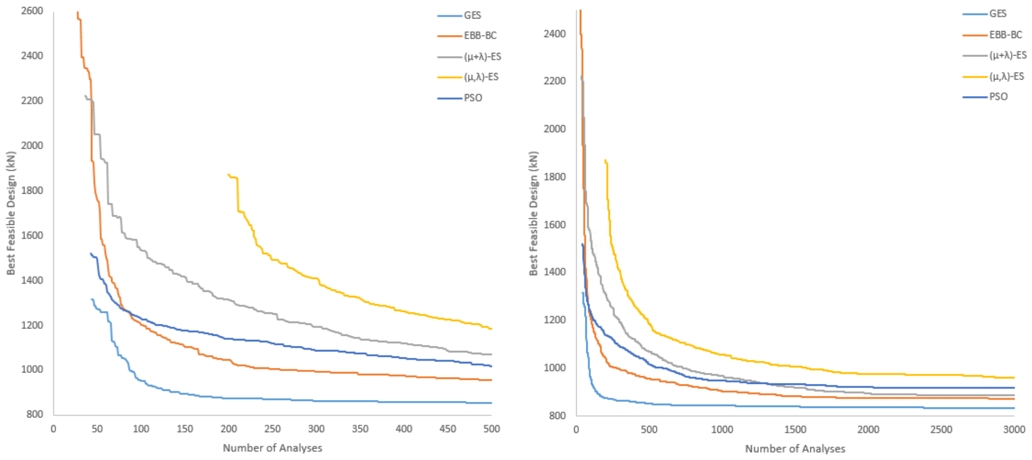
Analysis Count		**PSO**	**EBB-BC**	**(μ,λ)-ES**	**(μ+λ)-ES**	**GES**
500	Best	914.93	916.83	999.49	925.36	831.42
	Mean	1016.69	955.89	1184.64	1069.75	854.12
	Worst	1114.47	1024.26	1328.10	1213.38	904.76
	STD	63.49	33.20	109.26	103.81	21.14
	CV	6.24	3.47	9.22	9.70	2.47
1000	Best	866.38	853.49	942.86	897.51	805.16
	Mean	948.33	904.66	1058.37	969.05	844.91
	Worst	1014.91	952.38	1259.27	1082.82	900.51
	STD	47.19	32.52	89.96	67.96	23.29
	CV	4.98	3.60	8.50	7.01	2.76

Table 5.7. (continued)

2000	Best	861.14	837.18	887.96	843.86	805.16
	Mean	922.85	876.39	976.65	894.04	838.43
	Worst	980.88	909.61	1076.57	942.23	876.77
	STD	37.95	25.94	51.23	30.84	17.33
	CV	4.11	2.96	5.25	3.45	2.07
3000	Best	**861.14**	**837.18**	**887.96**	**843.86**	**805.16**
	Mean	917.18	874.54	961.87	887.83	834.43
	Worst	974.07	909.61	1041.94	942.23	849.90
	STD	36.33	26.67	43.27	31.43	12.37
	CV	3.96	3.05	4.50	3.54	1.48

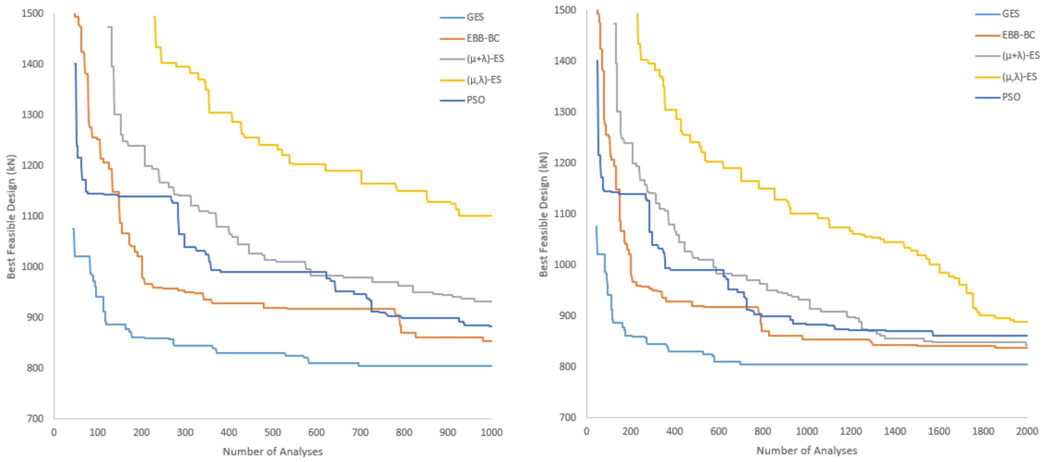
Table 5.8 The optimum designs obtained for the 160-member steel frame design example (under strength constraints) with different optimization techniques.

Stories	**Groups**	**PSO**	**EBB-BC**	**(μ,λ)-ES**	**(μ+λ)-ES**	**GES**
1-2	IB	W24x76	W27x84	W24x68	W24x68	W24x68
	OB	W16x40	W18x50	W14x34	W16x40	W12x40
	CG1	W24x55	W18x50	W27x84	W18x86	W18x76
	CG2	W30x99	W30x90	W36x135	W18x106	W21x101
	CG3	W33x221	W27x102	W27x194	W24x207	W24x146
	CG4	W21x182	W14x233	W24x207	W27x235	W14x211
3-4	IB	W21x62	W21x62	W18x60	W18x55	W18x55
	OB	W16x36	W14x34	W16x40	W16x36	W16x40
	CG1	W24x55	W18x50	W27x84	W18x50	W18x50
	CG2	W30x90	W30x90	W36x135	W18x76	W21x101
	CG3	W33x118	W27x102	W27x146	W24x117	24x104
	CG4	W21x182	W14x145	W24x146	W27x161	W14x132
Weight (kN)		**861.14**	**837.18**	**887.96**	**843.86**	**805.16**



(a) (b)

Figure 5.21 Average convergence curves obtained for the 160-member steel frame design example (under strength constraints) using various optimization techniques: (a) up to 500 analyses, (b) up to 3000 analyses



(a) (b)

Figure 5.22 Convergence curves obtained for the 160-member steel frame design example (under strength constraints) in the best run of various optimization techniques: (a) up to 1000 analyses, (b) up to 2000 analyses

Table 5.9 The member groups' DCR values in the optimum designs of the 160-member steel frame design example (under strength constraints) produced by various optimization techniques.

Stories	**Groups**	**PSO**	**EBB-BC**	**(μ,λ)-ES**	**(μ+λ)-ES**	**GES**
1-2	IB	0.9941	0.8752	0.9934	0.9995	0.9965
	OB	0.9857	0.9969	0.9638	0.9961	0.9995
	CG1	0.9873	0.9261	0.6157	0.6234	0.7277
	CG2	0.9239	0.9579	0.7569	0.7499	0.9181
	CG3	0.9187	0.9880	0.9091	0.8034	0.8925
	CG4	0.9988	0.9927	0.9921	0.9151	0.9546
3-4	IB	0.9071	0.9021	0.9705	0.9705	0.9642
	OB	0.9989	0.9962	0.9182	0.9974	0.9998
	CG1	0.7526	0.8491	0.6250	0.7689	0.8138
	CG2	0.8103	0.9104	0.7161	0.7872	0.8608
	CG3	0.9925	0.9802	0.7794	0.8968	0.9970
	CG4	0.9895	0.9630	0.9930	0.9952	0.9969
Average		0.9383	0.9448	0.8528	0.8753	0.9268
Max.		0.9989	0.9969	0.9934	0.9995	0.9998

5.2.2 Optimization Problem 1-B: 160-Member Steel Frame Under Strength and Inter-Story Drift Constraints

In this example, the 160-member steel frame (shown in Figure 5.1) under both the strength and inter-story drift constraints is studied with the same optimization algorithms mentioned in the previous example. In Section 5.1.2.2, this example has been studied extensively for sensitivity analysis of the GES algorithm by testing a large set of parameter values. The best solutions to the problem are produced when an exponentially decreasing value of the GML parameter is implemented with the GML_75-3-EXP scheme and also the guided mutation ratio parameter is set as 0.5 (i.e., %50).

The frame is designed by executing ten independent runs with the GES algorithm as well as with each metaheuristic search technique employed here. In Table 5.10, the minimum weight designs (best feasible solutions) achieved for the frame are presented in terms of the best, worst, and average (mean) solution attained with each optimization technique up to some selected stages of the optimization process; namely 1000, 1500, 2000, 3000 number of structural analyses. Moreover, the standard deviation (STD) and coefficient of variation (CV) values are also calculated and presented in the related table.

The results presented in Table 5.10 indicate that the GES algorithm demonstrates a superior performance than the metaheuristic search techniques for all stages of the optimization process. In Figure 5.23, the (average) variation of the best design weight against the number of analyses (i.e., average convergence curve) is plotted for each optimization technique, after averaging the results of all runs performed with a particular method. Similarly, in Figure 5.24, convergence curves are plotted considering only the best performance of the methods for the problem of interest. It can be observed from Figure 5.24 that GES has a faster convergence rate compared to other algorithms, such that it reaches 96% of the best feasible design (1073.76 kN) within only the first 600 analyses.

A comparison of the minimum weight designs (the best feasible designs) obtained using each optimization method is carried out in Table 5.11, where the section designations assigned to all member groups are indicated along with the weight of the resulting design. Among the results from the optimization algorithms shown in this table, the GES yields the least design weight for the 160-member steel frame, which is 1073.76 kN. The design weights that the other algorithms obtained are 1092.60 kN by $(\mu+\lambda)$-ES, 1079.11 kN by EBB-BC, and 1081.17 kN by PSO.

In the optimum designs of the 160-member steel frame, the computed DCR (demand-to-capacity ratio) values for member groups, the inter-story drift ratios in the *x*-direction, and the inter-story drift ratios in the *y*-direction are given in Table 5.12, Table 5.13 and Table 5.14, respectively. Moreover, in Figure 5.25, the variation of the best design's inter-story drift ratios against the number of stories is plotted for each optimization technique. It is seen from these tables that mainly inter-story drift limitations (especially due to earthquake loading in the y-direction) are the dominant constraints in this example.

Table 5.10 Optimization statistics for the 160-member steel frame design example (under both strength and inter-story drift constraints).

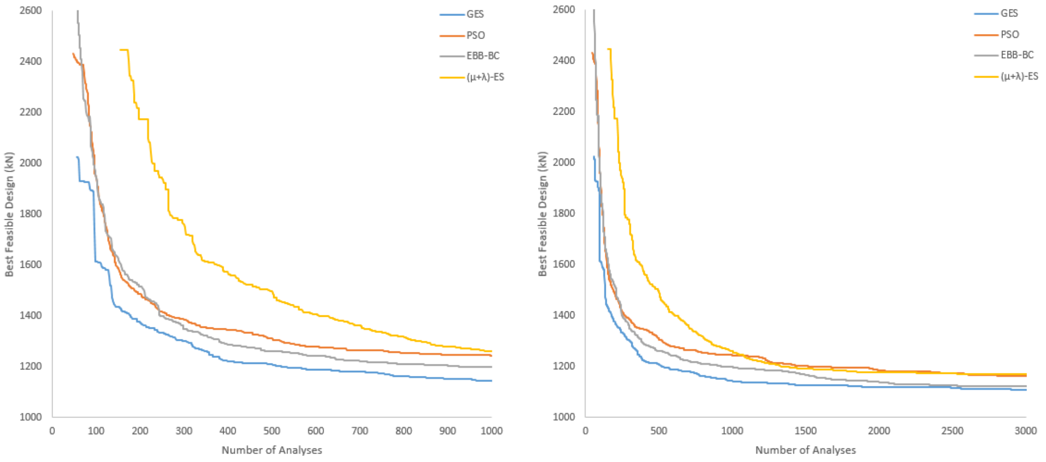
Analysis Count		PSO	EBB-BC	(μ+λ)-ES	GES
1000	MIN	1144.20	1160.07	1178.59	1094.03
	AVERAGE	1244.32	1198.26	1260.99	1143.14
	MAX	1391.15	1288.51	1352.34	1224.15
	STD	72.41	37.80	46.38	42.08
	CV	5.82	3.15	3.68	3.68
1500	MIN	1124.97	1103.22	1151.23	1083.30
	AVERAGE	1202.05	1164.92	1191.49	1127.64
	MAX	1330.92	1239.53	1230.67	1224.15
	STD	51.46	35.20	22.34	41.10
	CV	4.28	3.02	1.87	3.64
2000	MIN	1113.71	1079.11	1136.22	1073.76
	AVERAGE	1186.01	1138.74	1177.96	1119.73
	MAX	1239.59	1176.50	1209.75	1224.15
	STD	35.60	28.51	20.41	43.87
	CV	3.00	2.50	1.73	3.92
3000	MIN	**1081.17**	**1079.11**	**1092.60**	**1073.76**
	AVERAGE	1140.99	1117.98	1162.90	1097.91
	MAX	1174.75	1142.92	1209.75	1149.05
	STD	24.45	16.87	33.58	20.52
	CV	2.14	1.51	2.89	1.87

Table 5.11 The optimum designs obtained for the 160-member steel frame design example (under both strength and inter-story drift constraints) with different optimization techniques

Stories	Groups	PSO	EBB-BC	(μ+λ)-ES	GES
1-2	IB	W30x90	W30x90	W30x99	W30x90
	OB	W21x50	W18x40	W14x30	W18x46
	CG1	W21x62	W24x68	W33x118	W24x55
	CG2	W27x10	W30x90	W30x99	W30x90
	CG3	W40x244	W40x221	W40x221	W40x268
	CG4	W40x328	W40x328	W40x298	W40x298

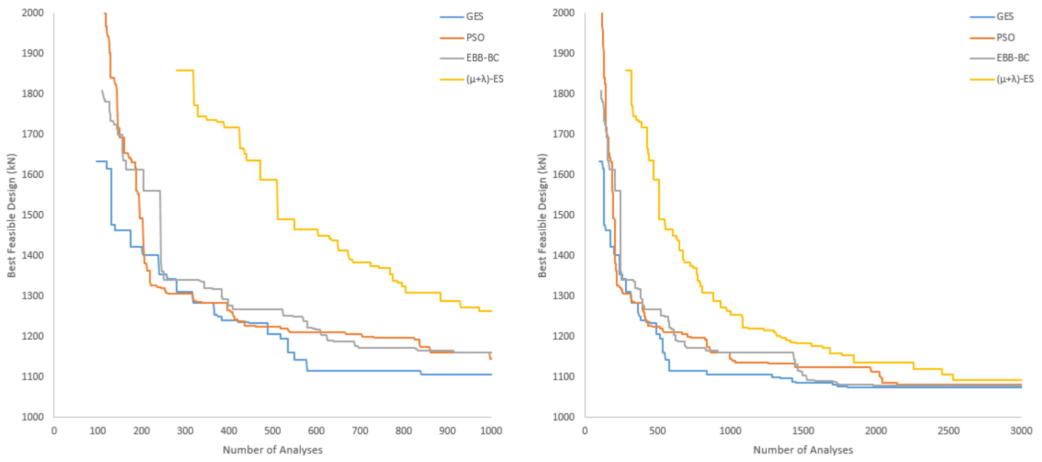
Table 5.11. (continued)

3-4	IB	W24x62	W24x62	W24x62	W24x62
	OB	W16x40	W24x55	W21x50	W18x46
	CG1	W21x62	W24x55	W33x118	W24x55
	CG2	W27x84	W30x90	W30x90	W30x90
	CG3	W40x221	W40x221	W40x192	W40x221
	CG4	W40x298	W40x298	W40x298	W40x298
Weight (kN)		**1081.17**	**1079.11**	**1092.60**	**1073.76**



(a) (b)

Figure 5.23 Average convergence curves obtained for the 160-member steel frame design example (under both strength and inter-story drift constraints) using various optimization techniques: (a) up to 1000 analyses, (b) up to 3000 analyses



(a) (b)

Figure 5.24 Convergence curves obtained for the 160-member steel frame design example (under both strength and inter-story drift constraints) in the best run of various optimization techniques: (a) up to 1000 analyses, (b) up to 3000 analyses

Table 5.12 The member groups' DCR values in the optimum designs of the 160-member steel frame design example (under both strength and inter-story drift constraints) produced by various optimization techniques.

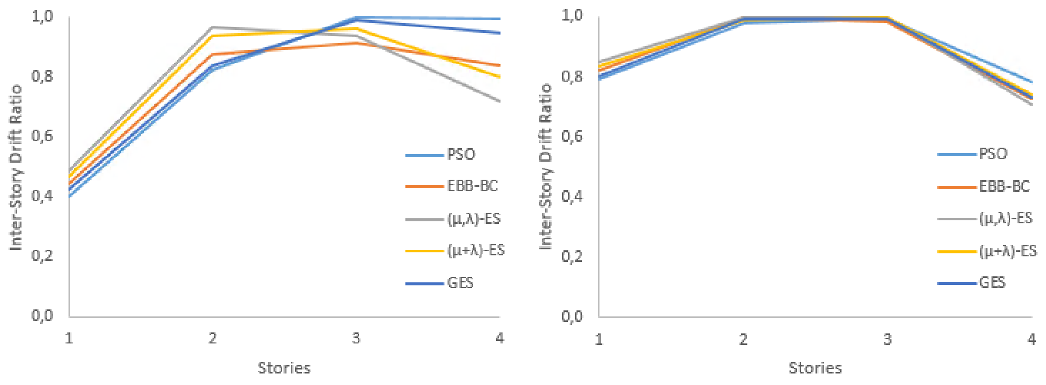
Stories	**Groups**	**PSO**	**EBB-BC**	**(μ+λ)-ES**	**GES**
1-2	IB	0.8106	0.8286	0.7810	0.8250
	OB	0.9970	0.9893	0.8738	0.8741
	CG1	0.7056	0.7100	0.5609	0.9391
	CG2	0.6477	0.7516	0.6729	0.7602
	CG3	0.5973	0.6375	0.6510	0.5819
	CG4	0.6152	0.6316	0.6671	0.6390
3-4	IB	0.9852	0.9948	0.9629	0.9868
	OB	0.7682	0.9659	0.9534	0.9191
	CG1	0.6416	0.6298	0.4953	0.6622
	CG2	0.6205	0.6402	0.6198	0.6395
	CG3	0.4426	0.4455	0.5356	0.4306
	CG4	0.4741	0.4653	0.4883	0.4793
Average		0.6921	0.7242	0.6885	0.7281

Table 5.13 The *x*-direction inter-story drift constraint values in the optimum designs of the 160-member steel frame design example (under both strength and inter-story drift constraints) produced by various optimization techniques.

Stories	**PSO**	**EBB-BC**	**(μ+λ)-ES**	**GES**
1	0.4030	0.4435	0.4665	0.4269
2	0.8242	0.8739	0.9397	0.8378
3	0.9991	0.9145	0.9604	0.9918
4	0.9946	0.8365	0.8021	0.9490
Average	0.8052	0.7671	0.7922	0.8014

Table 5.14 The *y*-direction inter-story drift constraint values in the optimum designs of the 160-member steel frame design example (under both strength and inter-story drift constraints) produced by various optimization techniques.

Stories	**PSO**	**EBB-BC**	**(μ+λ)-ES**	**GES**
1	0.7947	0.8199	0.8347	0.8028
2	0.9794	0.9972	0.9875	0.9911
3	0.9945	0.9808	0.9950	0.9910
4	0.7838	0.7261	0.7387	0.7296
Average	0.8881	0.8810	0.8890	0.8786



(a) (b)

Figure 5.25 Inter-story drift ratio curves obtained for the 160-member steel frame design example in the best run of various optimization techniques: (a) for x-direction, (b) for y-direction

5.2.3 Optimization Problem 2: 584-Member Steel Frame

The third design example refers to the 584-member steel frame shown in Figure 5.2. In Section 5.1.2.3, this example has also been studied extensively for sensitivity analysis of the GES algorithm by testing a large set of parameter values. The best solutions to the problem are produced when an exponentially decreasing value of the GML parameter is implemented with the GML_75-3-EXP scheme and also the guided mutation ratio parameter is set as 0.5 (i.e., %50).

The frame is designed by executing ten independent runs with the GES algorithm as well as with each metaheuristic search technique employed here. In Table 5.15, the minimum weight designs (best feasible solutions) achieved for the frame are presented in terms of the best, worst, and average (mean) solution attained with each optimization technique up to some selected stages of the optimization process; namely 1000, 1500, 2000, 3000 number of structural analyses. Moreover, the standard deviation (STD) and coefficient of variation (CV) values are also calculated and presented in the related table.

Although the GES algorithm provides guidance related to the strength constraints of the member groups only and the considered frame is subjected to high inter-story drift domination design constraints, the superior performance of this algorithm over the employed metaheuristic search techniques is observed in all the stages of the optimization process. In the early stages of the optimization process, the guiding ability of the GES algorithm is utilized more effectively. This is mainly due to the fact that both strength and inter-story drift constraints control the assignment of sections to the member groups. However, as better design solutions are generated through iterations, inter-story drift design constraints dominate the optimization process and mainly control the assignment of sections to the member groups. Since probabilistically half of the individuals (GMR parameter is set as 50%) are mutated according to the guided mutation scheme, the other half which is mutated according to the stochastic mutation handles geometric and inter-story constraints. This way, the GES algorithm retains its exploration and exploitation capabilities through all stages of the optimization process.

In Figure 5.26, the (average) variation of the best design weight against the number of analyses (i.e., average convergence curve) is plotted for each optimization technique, after averaging the results of all runs performed with a particular method. Similarly, in Figure 5.27, convergence curves are plotted considering only the best performance of the methods for the problem of interest. It can be observed from these two figures that GES has a faster convergence rate compared to other algorithms and yields the best design for the 584-member steel frame.

A comparison of the minimum weight designs (the best feasible designs) obtained using each optimization method is carried out in Table 5.16, where the section designations assigned to all member groups are indicated along with the weight of the resulting design. Among the results obtained from the optimization algorithms shown in this table, the GES yields the least design weight for the 584-member steel frame, which is 4771.93 kN. The design weights that the other algorithms obtained are 4877.17 kN by EBB-BC, 4884.44 kN by (μ+λ)-ES, and 5004.58 kN by POS.

In the optimum designs of the 584-member steel frame, the computed DCR (demand-to-capacity ratio) values for member groups, the inter-story drift ratios in the *x*-direction, and the inter-story drift ratios in the *y*-direction are given in Table 5.17, Table 5.18 and Table 5.19, respectively. Moreover, in Figure 5.28, the variation of the best design's inter-story drift ratios against the number of stories is plotted for each optimization technique.It is seen from these tables that like the previous design example mainly inter-story drift constraints (especially due to earthquake loading in the y-direction) are the dominant constraints in this example as well.

Table 5.15 Optimization statistics for the 584-member steel frame design example.

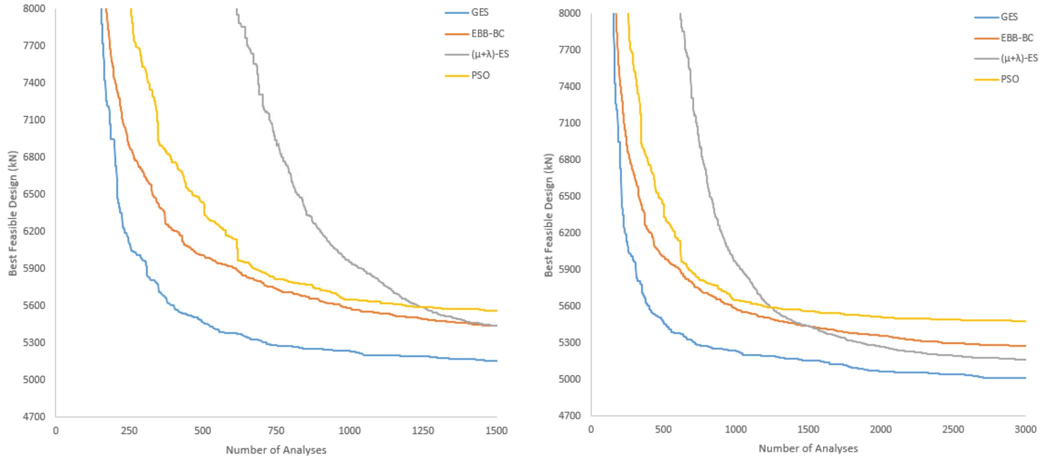
Analysis Count		**PSO**	**EBB-BC**	**(μ+λ)-ES**	**GES**
1000	MIN	5099.92	5095.45	5272.50	4978.12
	AVERAGE	5651.20	5580.88	5958.14	5236.75
	MAX	6522.44	6056.19	6591.63	5498.65
	STD	395.01	283.16	379.36	174.93
	CV	6.99	5.07	6.37	3.34
1500	MIN	5053.08	5028.35	5117.80	4898.68
	AVERAGE	5559.69	5437.94	5438.57	5158.29
	MAX	6522.44	5824.74	5853.46	5421.78
	STD	407.92	211.44	200.52	144.07
	CV	7.34	3.89	3.69	2.79
2000	MIN	5018.64	4919.18	4991.35	4771.93
	AVERAGE	5512.87	5352.90	5268.34	5069.11
	MAX	6522.44	5628.03	5577.43	5247.47
	STD	414.65	184.31	174.52	165.76
	CV	7.52	3.44	3.31	3.27
3000	MIN	**5004.58**	**4877.17**	**4884.44**	**4771.93**
	AVERAGE	5481.44	5272.89	5163.69	5017.01
	MAX	6522.44	5462.96	5540.25	5221.91
	STD	421.88	153.87	187.26	153.18
	CV	7.70	2.92	3.63	3.05

Table 5.16 The optimum designs obtained for the 584-member steel frame design example with different optimization techniques.

Stories	**Groups**	**PSO**	**EBB-BC**	**(μ+λ)-ES**	**GES**
1-2	CG1	W33x201	W33x201	W30x173	W30x173
	CG2	W40x244	W33x241	W40x221	W36x230
	CG3	W33x221	W36x245	W30x326	W40x249
	CG4	W36x359	W36x328	W36x439	W40x268
	IB	W33x118	W33x118	W30x99	W36x135
	OB	W24x68	W24x76	W24x76	W24x76

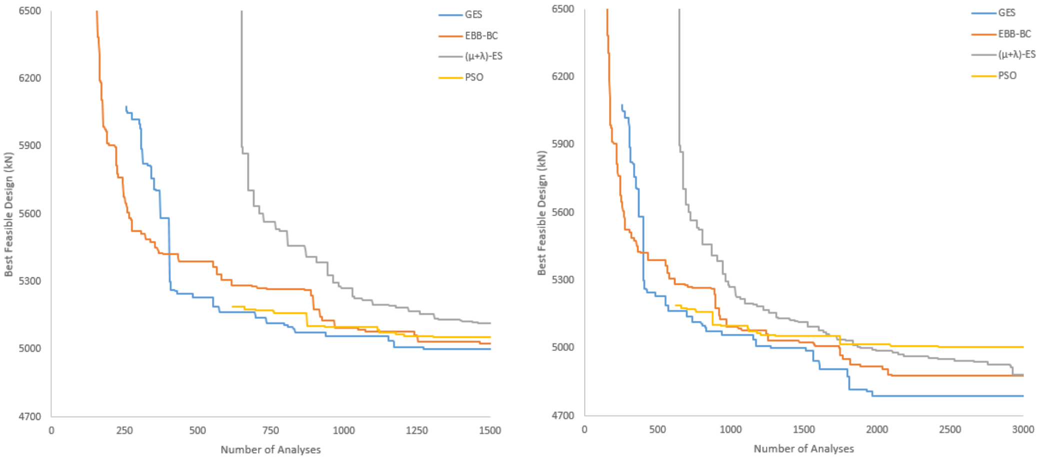
Table 5.16. (continued)

3-4	CG1	W33x201	W33x201	W30x173	W30x173
	CG2	W40x244	W33x221	W40x221	W36x230
	CG3	W33x221	W33x221	W30x326	W40x215
	CG4	W36x300	W36x328	W36x393	W40x268
	IB	W40x149	W33x118	W30x108	W33x118
	OB	W24x62	W24x76	W24x68	W24x68
5-6	CG1	W33x201	W33x118	W30x132	W30x173
	CG2	W40x192	W33x221	W40x221	W33x201
	CG3	W33x221	W33x201	W30x235	W40x199
	CG4	W36x245	W36x280	W36x300	W40x268
	IB	W30x99	W30x90	W30x90	W27x84
	OB	W24x62	W24x62	W24x55	W24x55
7-8	CG1	W33x201	W33x118	W30x90	W30x90
	CG2	W40x192	W33x201	W40x192	W33x201
	CG3	W33x118	W33x201	W30x173	W40x199
	CG4	W36x230	W33x221	W33x201	W40x192
	IB	W24x55	W21x50	W24x55	W24x62
	OB	W14x30	W14x34	W18x35	W18x35
Weight (kN)		**5004.58**	**4877.17**	**4884.44**	**4771.93**



(a) (b)

Figure 5.26 Average convergence curves obtained for the 584-member steel frame design example using various optimization techniques: (a) up to 1500 analyses, (b) up to 3000 analyses



(a) (b)

Figure 5.27 Convergence curves obtained for the 584-member steel frame design example in the best run of various optimization techniques:, (a) up to 1500 analyses, (b) up to 3000 analyses.

Table 5.17 The member groups' DCR values in the optimum designs of the 584-member steel frame design example produced by various optimization techniques

Stories	Groups	PSO	EBB-BC	(μ+λ)-ES	GES
1-2	CG1	0.5360	0.5702	0.6286	0.5870
	CG2	0.6791	0.6235	0.7079	0.5910
	CG3	0.8375	0.8247	0.6323	0.8428
	CG4	0.7741	0.8161	0.6138	0.8705
	IB	0.5840	0.5831	0.6616	0.6036
	OB	0.6869	0.5420	0.5613	0.6146
3-4	CG1	0.4129	0.4623	0.4853	0.5132
	CG2	0.4859	0.5808	0.5623	0.4490
	CG3	0.7475	0.7054	0.4723	0.6842
	CG4	0.7630	0.6749	0.5654	0.7460
	IB	0.5472	0.6191	0.6861	0.6764
	OB	0.9735	0.6158	0.6476	0.8343

Table 5.17. (continued)

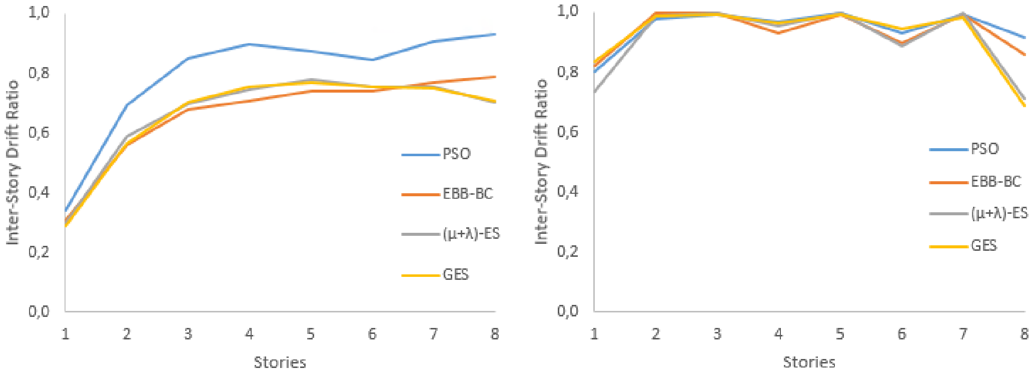
5-6	CG1	0.3781	0.5775	0.5066	0.6237
	CG2	0.5979	0.5134	0.4925	0.4613
	CG3	0.6402	0.6149	0.5200	0.5772
	CG4	0.7615	0.6545	0.6002	0.6413
	IB	0.6572	0.6664	0.6780	0.7071
	OB	0.9546	0.7800	0.8851	0.6304
7-8	CG1	0.2792	0.4803	0.5157	0.5576
	CG2	0.4400	0.3866	0.4153	0.4706
	CG3	0.6048	0.4654	0.5110	0.5900
	CG4	0.5649	0.5367	0.5830	0.5176
	IB	0.9074	0.9188	0.8418	0.9600
	OB	0.8749	0.6857	0.9465	0.7082
Average		0.6537	0.6207	0.6133	0.6441

Table 5.18 The *x*-direction inter-story drift constraint values in the optimum designs of the 584-member steel frame design example produced by various optimization techniques

Story	**PSO**	**EBB-BC**	**(μ+λ)-ES**	**GES**
1	0.3406	0.3066	0.3011	0.2906
2	0.6953	0.5585	0.5909	0.5655
3	0.8503	0.6766	0.6987	0.7036
4	0.8960	0.7094	0.7447	0.7557
5	0.8755	0.7398	0.7770	0.7705
6	0.8466	0.7386	0.7572	0.7567
7	0.9078	0.7684	0.7553	0.7502
8	0.9324	0.7892	0.7026	0.7052
Average	0.7931	0.6609	0.6659	0.6622

Table 5.19 The *y*-direction inter-story drift constraint values in the optimum designs of the 584-member steel frame design example produced by various optimization techniques

Story	PSO	EBB-BC	(μ+λ)-ES	GES
1	0.8033	0.8224	0.7367	0.8344
2	0.9793	0.9950	0.9896	0.9881
3	0.9911	0.9977	0.9967	0.9934
4	0.9667	0.9326	0.9555	0.9642
5	0.9967	0.9917	0.9953	0.9923
6	0.9328	0.8955	0.8882	0.9449
7	0.9915	0.9919	0.9993	0.9829
8	0.9154	0.8581	0.7103	0.6866
Average	0.9471	0.9356	0.9089	0.9233



(a) (b)

Figure 5.28 Inter-story drift ratio curves obtained for the 584-member steel frame design example in the best run of various optimization techniques: (a) for x-direction, (b) for y-direction

5.3 Numerical Examples of Real-World Steel Structures

In this section, three real-world steel structures that have been formerly designed by practicing engineers using a traditional design procedure are optimized and redesigned by the GES algorithm and metaheuristic search techniques. This way the practical applicability of optimization techniques for real-world problems is illustrated. Besides, the amount of material savings that could be achieved through a design optimization procedure is identified with respect to a traditional design procedure implemented by a practicing engineer.

5.3.1 Optimization Problem 1: 545-Member Steel Frame

The first real-world steel structure refers to a trade center constructed at the ropeway station in Beşikdüzü, Trabzon. The trade center consists of three blocks designated as Block-A, Block-B, and Block-C. Each block has a steel frame structural system. The blocks are designed in accordance with the provisions of AISC 360-10 design specifications. The dead, live, snow, and wind loads acting on the blocks are computed according to TS-498 design load specifications. Load combinations applied to this structure are shown in Table 5.20. On the other hand, the earthquake loads are applied to the steel frames by the equivalent lateral load procedure in the Turkish Seismic Code 2018 (TSC-2018), where the seismic coefficients are chosen as listed in Table 5.22. The amplified inter-story drift is restricted to 2% of story height. Moreover, the design constraints are enforced as defined in detail in Chapter 2.

Figure 5.29 shows the structural model prepared for the Block-A of this building, which is used for the optimization studies carried out in this section. The initial design process has been carried out by a design office, resulting in a total design weight of 482.49 kN for the steel frame. However, it has been found that this initial design violates beam-deflection constraints with a maximum constraint violation of 2.117. Besides, beam-column geometric constraint violations are identified in the order of 1.36 in total, and strong column-weak beam constraints are violated in the order of 2.30 in total.

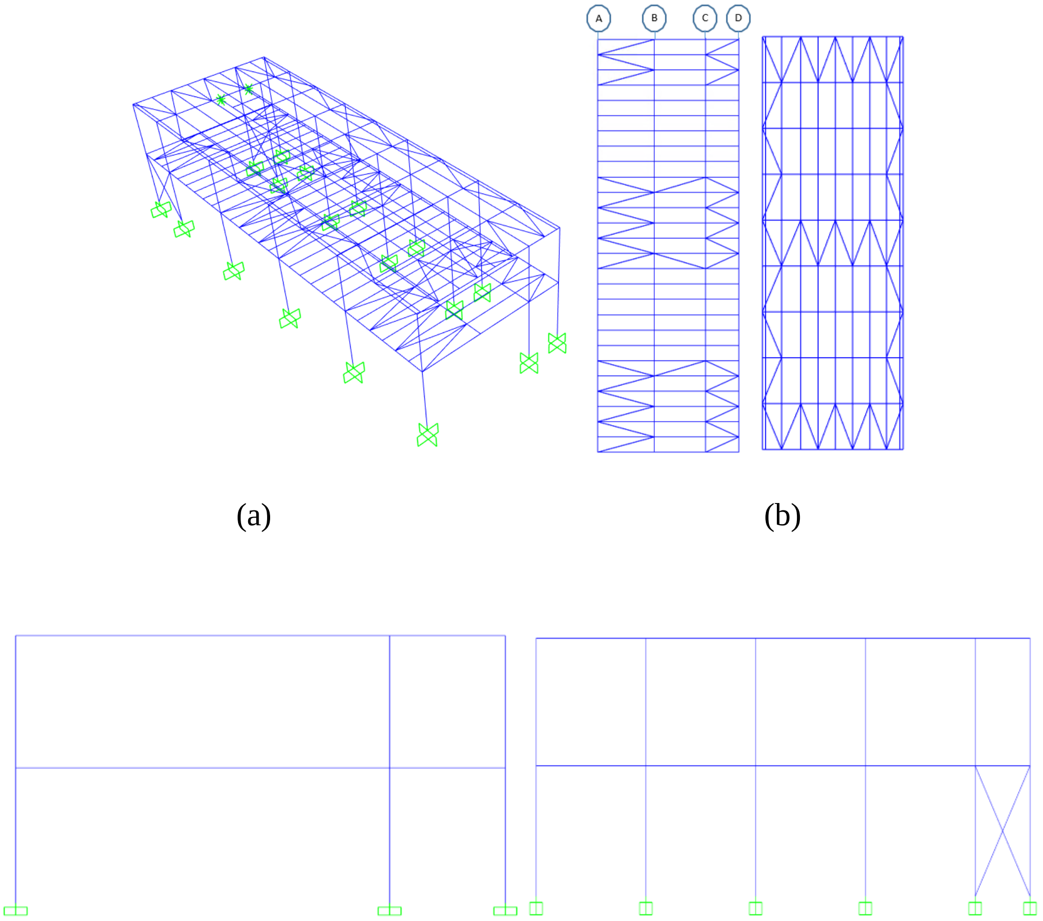
The grouping of the frame members is carried out in the same line with the way the structure has been formerly designed in practice. Accordingly, the 545 members of the frame are collected under 13 member groups. As highlighted in Figure 5.30, the first floor girders and beams are grouped into six sizing variables. On the other hand, the roof girders and beams are grouped into three sizing variables, as depicted in Figure 5.31. Similarly, all the columns are grouped into two sizing variables, as shown in Figure 5.32. Finally, as demonstrated in Figure 5.33, all lateral bracing members of the frame are all grouped into a single sizing variable, and so are all vertical bracing members. Five different profile lists are defined and used for sizing the 13 member groups in accordance with the original treatment of the design problem. The columns and girders are selected from S275JR HE sections, the first floor beams are selected from S275JR IPE sections, the roof floor beams are selected from S235JR RHS (Rectangular Hollow Sections) and the bracing members are selected from S235JR CHS (Circular Hollow Sections). In Table 5.21, the profile lists containing the corresponding sections of the member groups are presented.

The GES algorithm is executed with the most effective parameter settings determined in the previous sections; that is, an exponentially decreasing value of the GML parameter is implemented with the GML_15-5-EXP scheme and also the GMR parameter is set to 0.5 (i.e., %50).

The frame is designed by executing ten independent runs with the GES algorithm as well as with each metaheuristic search technique employed here. In Table 5.23, the minimum weight designs (best feasible solutions) achieved for the frame are presented in terms of the best, worst, and average (mean) solution attained with each optimization technique up to some selected stages of the optimization process; namely 500 and 1000 number of structural analyses. Moreover, the standard deviation (STD) and coefficient of variation (CV) values are also calculated and presented in the related table. The results presented in Table 5.23 indicate that the GES technique's superior performance over the other metaheuristic techniques employed is observed and only GES and EBB-BC algorithms can reach the optimum design before 500 analyses.

In Figure 5.34, the (average) variation of the best design weight against the number of analyses (i.e., average convergence curve) is plotted for each optimization technique, after averaging the results of all runs performed with a particular method. Similarly, in Figure 5.35, convergence curves are plotted considering only the best performance of the methods for the problem of interest. The GES technique reaches the optimum design by performing the least number of structural analyses compared to metaheuristic search methods. Indeed, the GES algorithm seizes the optimum design (448.84 kN) by performing only 209 structural analyses. The minimum design weight of the frame obtained with other methods at the same number of structural analyses (i.e., 209 structural analyses) is 460.43 kN by EBB-BC, 466.89 kN by PSO, and 715.52 kN by (μ+λ)-ES. Although all metaheuristic algorithms can find the optimum design in 1000 analyses, only the EBB-BC manages to reach it in the first 500 analyses. In Table 5.24, the initial (original) and optimum design of the steel frame are reported and compared with the section designations assigned to all member groups.

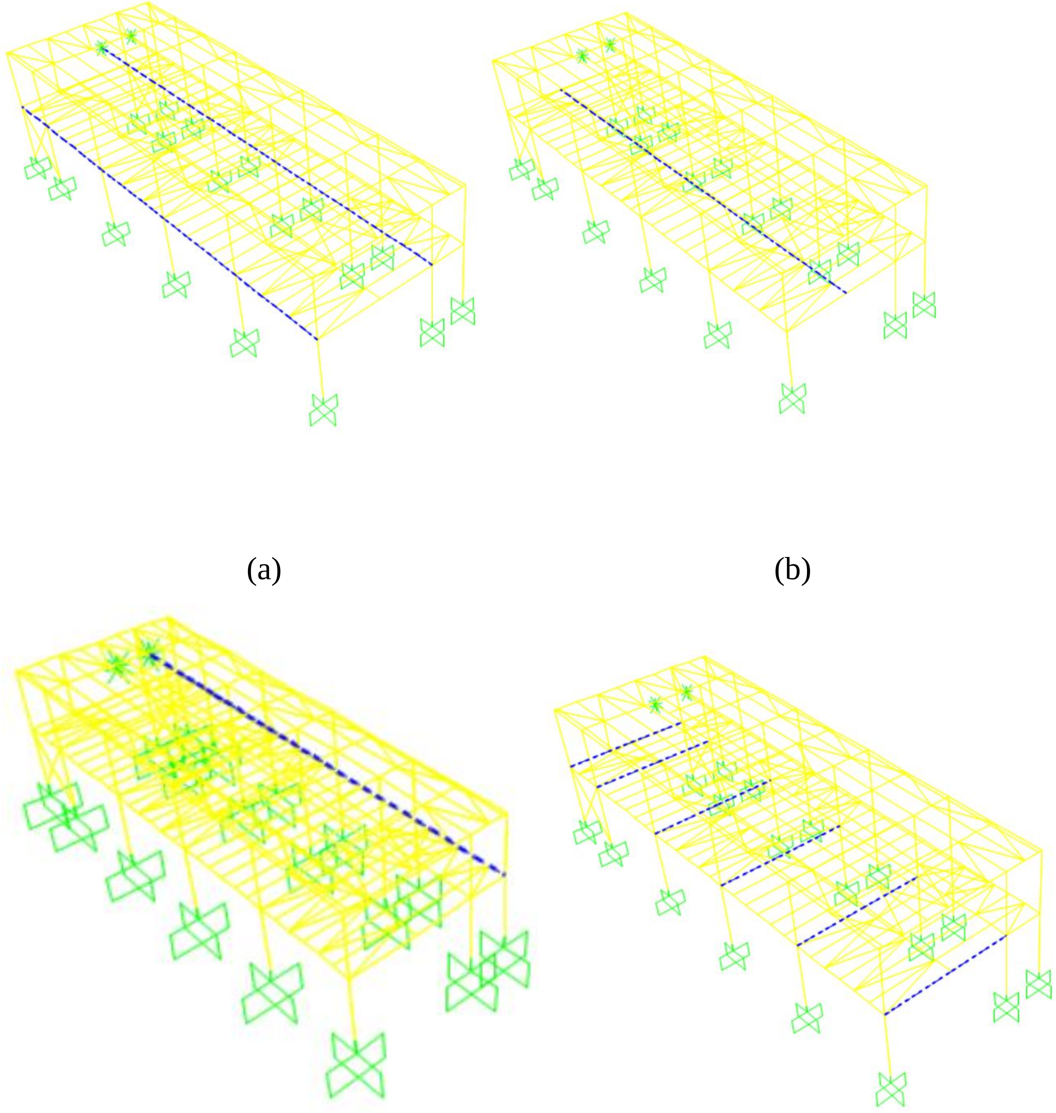
In the optimum design of the steel frame, the computed DCR (demand-to-capacity ratio) values for member groups and inter-story drift ratios in *x* and *y*-directions are given in Table 5.25 and Table 5.26, respectively. The results presented in these tables indicate that the design variables (cross-sections of the member groups) are mainly controlled by DCRs. However, as mentioned in previous sections, geometric constraints presented in Chapter 2 also affect the selection of cross-sections for member groups. In order to satisfy these constraints, larger sections are assigned to member groups, resulting in lower DCR values for some member groups.



(a) (b)

(c) (d)

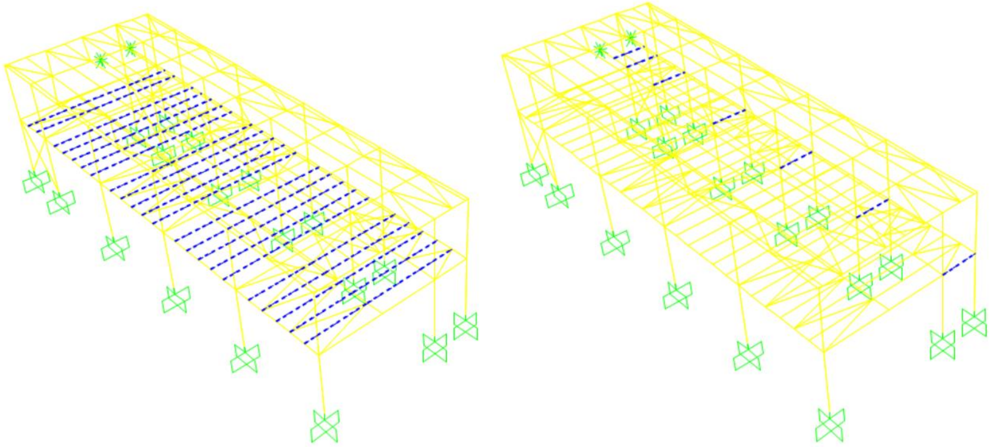
Figure 5.29 545-member steel frame: (a) 3-D view, (b) plan view, (c) side view in x-z plane, (d) side view in y-z plane.



(a) (b)

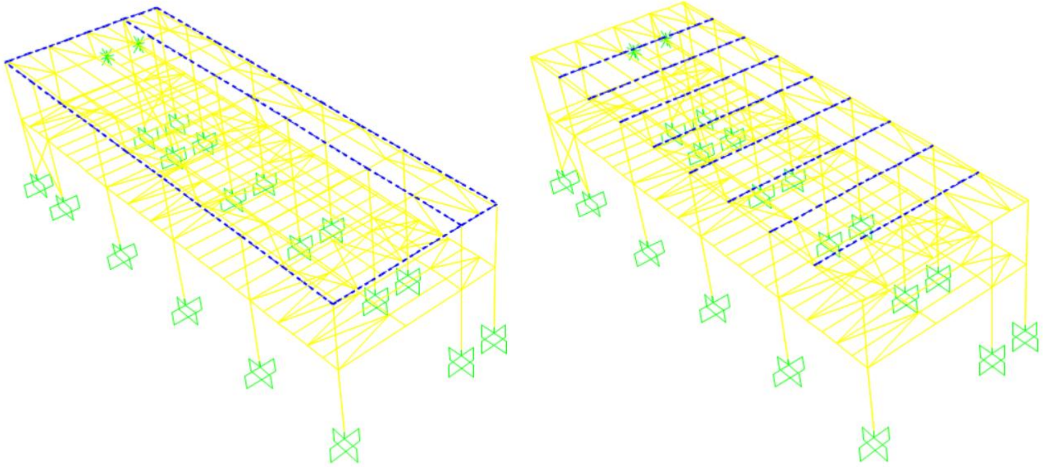
(c) (d)

Figure 5.30 The first floor girder and beam member groups for the 545-member steel frame: (a) the first floor long-side girders on A and C axes (GR1), (b) the first floor long-side girders on B axis (GR2), (c) the first floor long-side girder on D axis (GR3), (d) the first floor short-side girders between A-C axes (GR4), (e) the first floor short-side beams (BM1), (f) the first floor short-side girders between C-D axes (GR7).



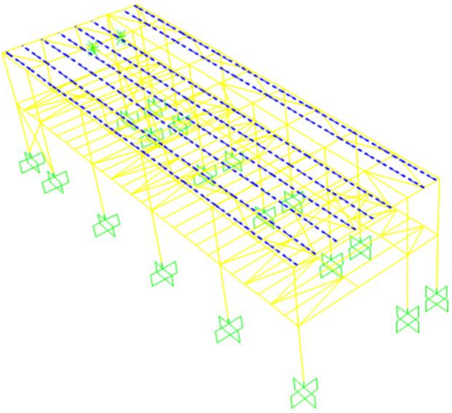
(e) (f)

Figure 5.30 (continued)



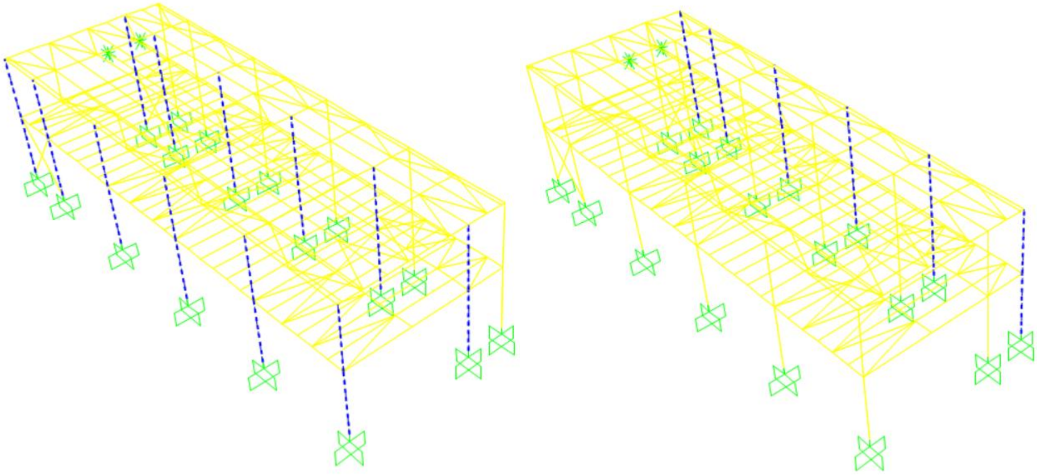
(a) (b)

Figure 5.31 The roof floor girder and beam member groups for the 545-member steel frame: (a) the roof floor long-side girders and short-side exterior girders (GR5), (b) the roof floor short-side interior girders (GR6), (c) the roof floor beams (BM2).



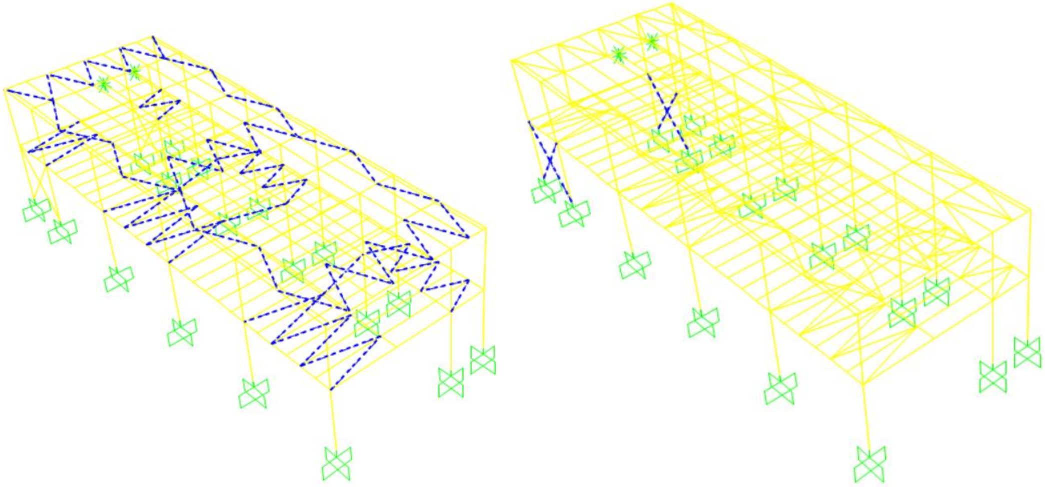
(c)

Figure 5.31 (continued)



(a) (b)

Figure 5.32 The column member groups for the 545-member steel frame: (a) the columns on axes A and C (CL1), (b) the columns on axis D (CL2).



(a) (b)

Figure 5.33 Bracing member groups for the 545-member steel frame: (a) horizontal braces (BR1), (b) vertical braces (BR2).

Table 5.20 Load combination definitions for the 545-member steel frame design example

Load Combination Definitions		
1.4G*	1.2G+1.6Q+0.5S+T	1.2G+Q+0.5S-1.6Wy+T
1.2G+1.6Q+0.5S*	1.2G+1.6Q+0.5S-T	1.2G+Q+0.5S-1.6Wy-T
1.2G+Q+1.6S	1.2G+Q+1.6S+T	0.9G+1.6Wx+T
1.2G+1.6S+0.8Wx*	1.2G+Q+1.6S-T	0.9G+1.6Wx-T
1.2G+1.6S-0.8Wx	1.2G+1.6S+0.8Wx+T	0.9G-1.6Wx+T
1.2G+1.6S+0.8Wy	1.2G+1.6S+0.8Wx-T	0.9G-1.6Wx-T
1.2G+1.6S-0.8Wy	1.2G+1.6S-0.8Wx+T	0.9G+1.6Wy+T
1.2G+Q+0.5S+1.6Wx	1.2G+1.6S-0.8Wx-T	0.9G+1.6Wy-T
1.2G+Q+0.5S-1.6Wx	1.2G+1.6S+0.8Wy+T	0.9G-1.6Wy+T
1.2G+Q+0.5S+1.6Wy	1.2G+1.6S+0.8Wy-T	0.9G-1.6Wy-T
1.2G+Q+0.5S-1.6Wy	1.2G+1.6S-0.8Wy+T	1.2G+Q+0.2S+0.3Ex+Ey
1.2G+Q+0.2S+Ex+0.3Ey*	1.2G+1.6S-0.8Wy-T	0.9G+Ex+0.3Ey
0.9G+1.6Wx	1.2G+Q+0.5S+1.6Wx+T	0.9G+0.3Ex+Ey
0.9G-1.6Wx	1.2G+Q+0.5S+1.6Wx-T	1.2G+Q+0.2S+Ex+0.3Ey
0.9G+1.6Wy	1.2G+Q+0.5S-1.6Wx+T	1.2G+Q+0.2S+0.3Ex+Ey
0.9G-1.6Wy	1.2G+Q+0.5S-1.6Wx-T	0.9G+Ex+0.3Ey
1.4G+T*	1.2G+Q+0.5S+1.6Wy+T	0.9G+0.3Ex+Ey
1.4G-T	1.2G+Q+0.5S+1.6Wy-T	

*G: Dead Loads; Q: Live Loads; Ex, Ey: Earthquake in x-direction, y-direction; S: Snow Load; T: Temperature; Wx, Wy: Wind Load in x-direction, y-direction

Table 5.21 Profile Lists for the 545-member steel frame design example

<table>
<tr><th colspan="8">Profile List 1 - I/Wide Flange</th></tr>
<tr><td>HE100A</td><td>HE100B</td><td>HE120A</td><td>HE120B</td><td>HE140A</td><td>HE140B</td><td>HE160A</td><td>HE160B</td></tr>
<tr><td>HE160M</td><td>HE180A</td><td>HE180B</td><td>HE180M</td><td>HE200A</td><td>HE200B</td><td>HE200M</td><td>HE220A</td></tr>
<tr><td>HE220B</td><td>HE220M</td><td>HE240A</td><td>HE240B</td><td>HE240M</td><td>HE260A</td><td>HE260B</td><td>HE260M</td></tr>
<tr><td>HE280A</td><td>HE280B</td><td>HE280M</td><td>HE300A</td><td>HE300B</td><td>HE300C</td><td>HE300M</td><td>HE320A</td></tr>
<tr><td>HE320B</td><td>HE320M</td><td>HE340A</td><td>HE340B</td><td>HE340M</td><td>HE360A</td><td>HE360B</td><td>HE360M</td></tr>
<tr><td>HE400A</td><td>HE400B</td><td>HE400M</td><td>HE400x107</td><td>HE450A</td><td>HE450B</td><td>HE450M</td><td>HE450x123</td></tr>
<tr><td>HE500A</td><td>HE500B</td><td>HE500M</td><td>HE600x151</td><td>HE550B</td><td>HE550M</td><td>HE600A</td><td>HE600x137</td></tr>
<tr><td>HE600M</td><td>HE600B</td><td>HE550A</td><td>HE600x174</td><td>HE650A</td><td>HE650B</td><td>HE650M</td><td>HE800A</td></tr>
<tr><td>HE700B</td><td>HE700M</td><td>HE700A</td><td>HE700x166</td><td>HE800B</td><td>HE800M</td><td>HE900A</td><td></td></tr>
<tr><th colspan="8">Profile List 2 - I/Wide Flange</th></tr>
<tr><td>IPE100</td><td>IPE120</td><td>IPE140</td><td>IPE160</td><td>IPE180</td><td>IPE200</td><td>IPE220</td><td>IPE240</td></tr>
<tr><td>IPE270</td><td>IPE300</td><td>IPE600</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><th colspan="8">Profile List 3 - Pipe</th></tr>
<tr><td colspan="2">CHS114.3X3</td><td colspan="2">CHS114.3X4</td><td colspan="2">CHS139.7X4</td><td colspan="2">CHS76.1X4</td></tr>
<tr><td colspan="2">CHS88.9X3</td><td colspan="6"></td></tr>
<tr><th colspan="8">Profile List 4 - Pipe</th></tr>
<tr><td colspan="4">TUBO-D168.3X4</td><td colspan="4">TUBO-D273X5.6</td></tr>
<tr><th colspan="8">Profile List 5 - Box/Tube</th></tr>
<tr><td colspan="2">RHS120X60X3.6</td><td colspan="2">RHS160X80X4</td><td colspan="2">RHS200X100X5</td><td colspan="2"></td></tr>
</table>

Table 5.22 Earthquake seismic coefficients used for calculating earthquake loads acting on the 545-member steel frame.

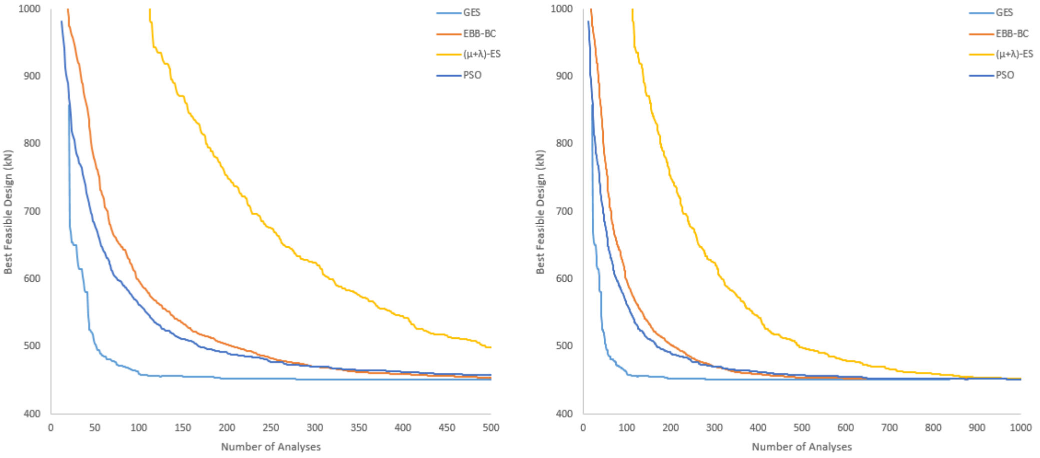
Seismic Coefficients	
0.2 Sec Spectral Accel, S_s	0.459
0.1 Sec Spectral Accel, S_1	0.121
Long-Period Transition Period	8
Site Class	ZB
Site Coefficient, F_a	0.9
Site Coefficient, F_v	0.8
$S_{DS}= F_a S_s$	0.4131
$S_{D1}= F_v S_1$	0.0968

Table 5.23 Optimization statistics for the 545-member steel frame design example.

Analysis Count		**PSO**	**EBB-BC**	**(μ+λ)-ES**	**GES**
500	Best	453.01	**448.84**	451.66	**448.84**
	Mean	456.23	452.56	461.20	451.78
	Worst	464.92	463.64	473.20	457.11
	STD	3.41	3.97	6.58	2.47
	CV	0.75	0.88	1.43	0.55
1000	Best	**448.84**	**448.84**	**448.84**	**448.84**
	Mean	451.82	449.37	452.14	451.22
	Worst	454.37	451.50	457.40	453.23
	STD	2.06	1.06	2.58	1.72
	CV	0.46	0.24	0.57	0.38

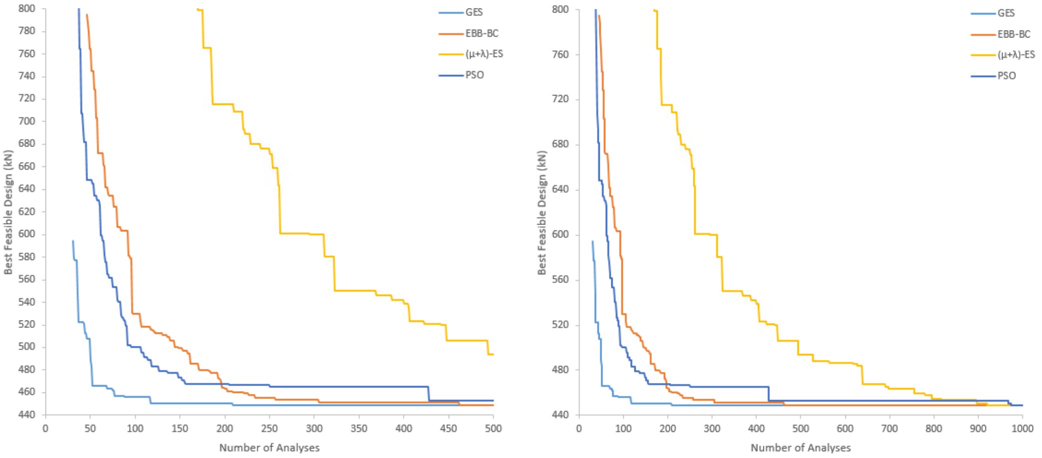
Table 5.24 The initial (original) and optimum designs of the 545-member steel frame design example

Groups	**Sections in the Initial Design**	**Sections in the Optimum Design**
GR1	HE300A	HE260A
GR2	HE320A	HE320A
GR3	HE200A	HE240A
GR4	HE300B	HE400x107
BM1	IPE200	IPE200
BR1	CHS88.9X3	CHS88.9X3
BR2	TUBO-D168.3X4	TUBO-D168.3X4
GR5	HE200A	HE140A
GR6	HE200A	HE160A
BM2	RHS100X100X3	RHS120X60X3.6
CL1	HE300B	HE400x107
CL2	HE200A	HE260A
GR7	HE200A	HE100A
Weight (kN)	**482.49**	**448.84**



(a) (b)

Figure 5.34 Average convergence curves obtained for the 545-member steel frame design example using various optimization techniques: (a) up to 500 analyses, (b) up to 1000 analyses



(a) (b)

Figure 5.35 Convergence curves obtained for the 545-member steel frame design example in the best run of various optimization techniques: (a) up to 500 analyses, (b) up to 1000 analyses

Table 5.25 The member groups' DCR values in the optimum design of the 545-member steel frame design example.

Groups	**DCR**
GR1	0.9158
GR2	0.9009
GR3	0.6293
GR4	0.8648
BM1	0.7013
BR1	0.3475
BR2	0.3013
GR5	0.8532
GR6	0.8048
BM2	0.9562
CL1	0.6468
CL2	0.2673
GR7	0.7784
Average	0.6898

Table 5.26 The inter-story drift constraint values in the optimum design of the 545-member steel frame design example.

Story	**Inter-Story Drift X-Dir.**	**Inter-Story Drift Y-Dir.**
1	0.2619	0.0718
2	0.3186	0.2011
Average	0.2902	0.1365

5.3.2 Optimization Problem 2: 551-Member Steel Frame

As a second problem from the real-world steel structures, the Block-B of the trade center mentioned in the previous section is studied. Figure 5.36 presents a structural model of this block, which consists of 551 steel members. The loads and design conditions are defined as the same in the previous example.

The initial design process has been carried out by a design office, resulting in a total design weight of 428,55 kN for the steel frame. However, it has been found that this initial design violates beam-deflection constraints with a maximum constraint violation of 2.097. Besides, strong column-weak beam constraints are violated in the order of 7.34 in total, whereas the geometric constraints between beams and columns are all satisfied.

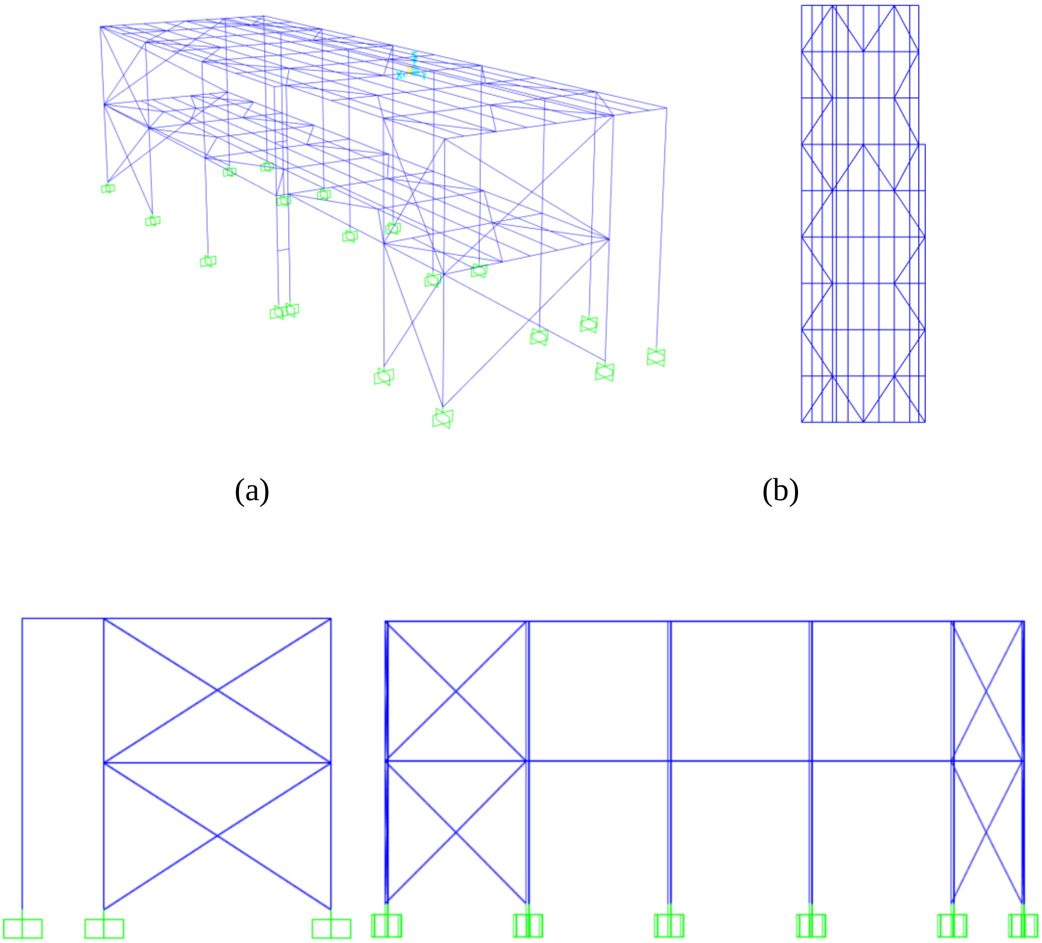
The grouping of the frame members is carried out in the same line with the way the structure has been formerly designed in practice. Accordingly, the 551 members of the frame are collected under 12 member groups. As highlighted in Figure 5.37, the roof floor girders and beams are grouped into three sizing variables. Similarly, the first floor girders and beams are grouped into three sizing variables as in Figure 5.38. All the columns are grouped into three sizing variables, as shown in Figure 5.39. Finally, the bracing members are grouped into three sizing variables, as demonstrated in Figure 5.40. The columns and girders are selected from S275JR HE sections, the first floor beams are selected from S275JR IPE sections, the roof floor beams are selected from S235JR RHS (Rectangular Hollow Sections) and the bracing members are selected from S235JR CHS (Circular Hollow Sections). In Table 5.27, the profile lists containing the corresponding sections of the member groups are presented.

The GES algorithm is executed with the most effective parameter settings determined in the previous sections; that is, an exponentially decreasing value of the GML parameter is implemented with the GML_15-53-EXP scheme and also the GMR parameter is set to 0.5 (i.e., %50).

The frame is designed by executing ten independent runs with the GES algorithm as well as with each metaheuristic search technique employed here. In Table 5.28, the minimum weight designs (best feasible solutions) achieved for the frame are presented in terms of the best, worst, and average (mean) solution attained with each optimization technique up to some selected stages of the optimization process; namely 250 and 500 number of structural analyses. Moreover, the standard deviation (STD) and coefficient of variation (CV) values are also calculated and presented in the related table. The results presented in Table 5.28 indicate a superior performance of the GES technique over the other metaheuristic techniques employed at all stages of the optimization.

In Figure 5.41, the (average) variation of the best design weight against the number of analyses (i.e., average convergence curve) is plotted for each optimization technique, after averaging the results of all runs performed with a particular method. Similarly, in Figure 5.42, convergence curves are plotted considering only the best performance of the methods for the problem of interest. All the optimization algorithms successfully converge to the same optimum design weight, which is 390.94 kN. However, among all the results obtained with different optimization algorithms, the GES algorithm reaches the optimum solution by performing the least number of structural analyses compared to the metaheuristic search techniques. Indeed, the GES algorithm seizes the optimum design (390.94 kN) by performing only 45 structural analyses. The minimum design weight of the frame obtained with other methods at the same number of structural analyses (i.e., 45 structural analyses) is 542.86 kN by EBB-BC, 463.35 kN by PSO, while the $(\mu+\lambda)$-ES does not reach a feasible design yet. The metaheuristic techniques EBB-BC, PSO, and (μ+λ)-ES locate the optimum design (390.94 kN) after 277, 315, and 423 structural analyses, respectively. In Table 5.29, the initial (original) and optimum design of the steel frame are reported and compared with the section designations assigned to all member groups.

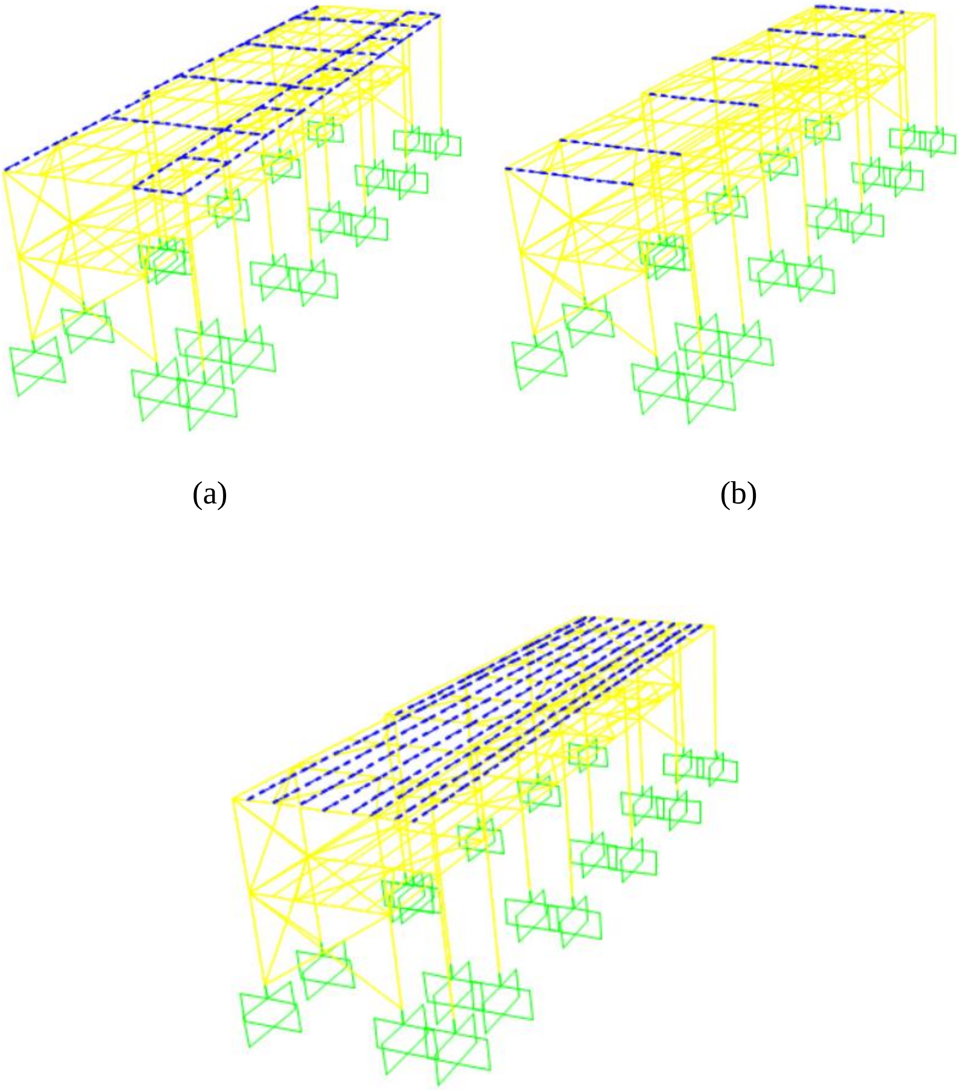
In the optimum design of the steel frame, the computed DCR (demand-to-capacity ratio) values for member groups and inter-story drift ratios in x and y-directions are given in Table 5.30 and Table 5.31, respectively. The results presented in these tables indicate that the design variables (cross-sections of the member groups) are mainly controlled by DCRs. However, as mentioned in previous sections, geometric constraints presented in Chapter 2 also affect the selection of cross-sections for member groups. In order to satisfy these constraints, larger sections are assigned to member groups, resulting in lower DCR values for some member groups.



(a) (b)

(c) (d)

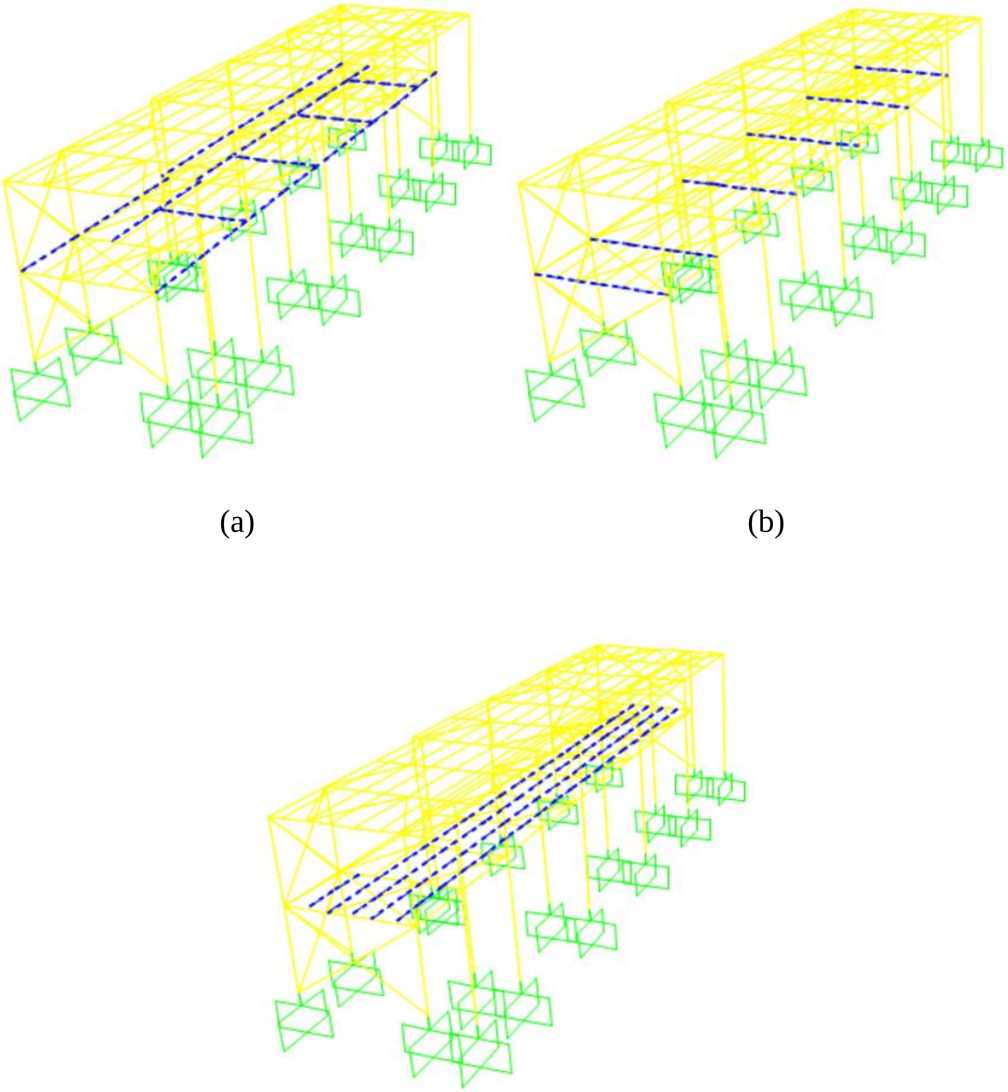
Figure 5.36 551-member steel frame: (a) 3-D view, (b) plan view, (c) side view in x-z plane, (d) side view in y-z plane



(a) (b)

(c)

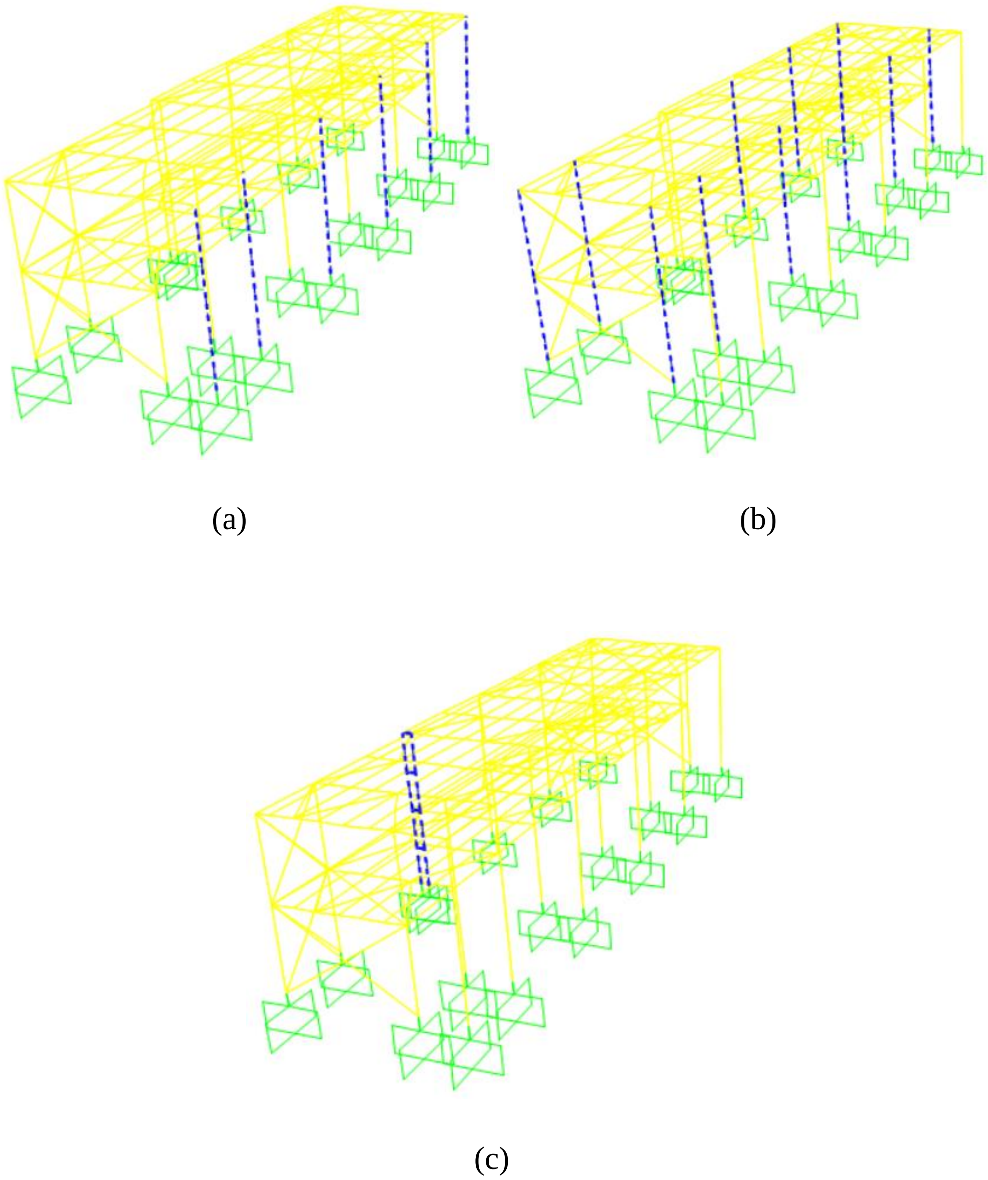
Figure 5.37 The roof floor girder and beam member groups for the 551-member steel frame: (a) roof girders1, (b) roof girders2, (c) roof beams



(a) (b)

(c)

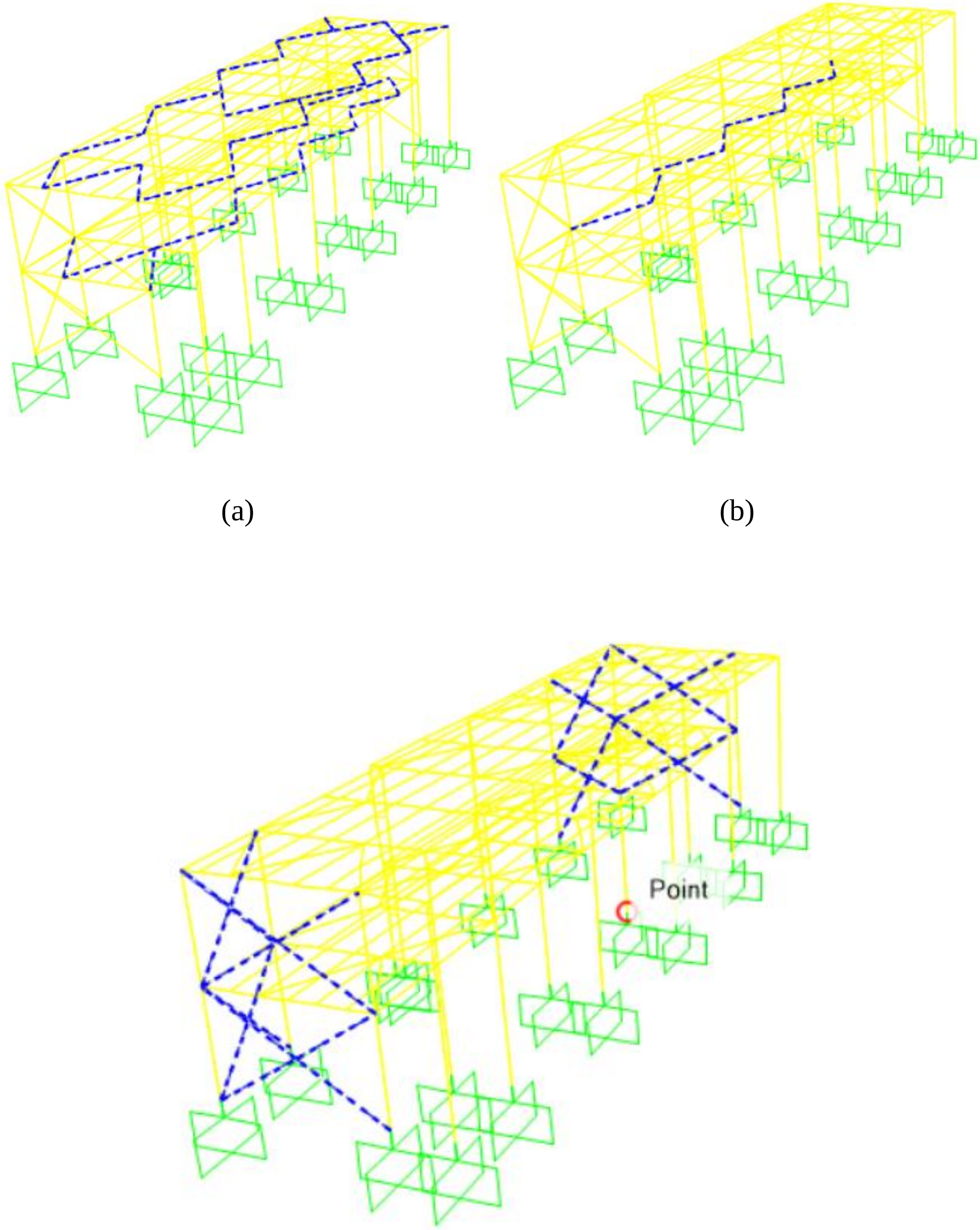
Figure 5.38 The first floor girder and beam member groups for the 551-member steel frame: (a) the first floor girders1, (b) the first floor girders2, (c) the first floor beams1.



(a) (b)

(c)

Figure 5.39 the column member groups for the 551-member steel frame: (a) columns1, (b) columns2, (c) columns3



(a) (b)

(c)

Figure 5.40 The brace member groups for the 551-member steel frame: (a) lateral braces1, (b) lateral braces2, (c) vertical braces

Table 5.27 Profile Lists for the 551-member steel frame design example

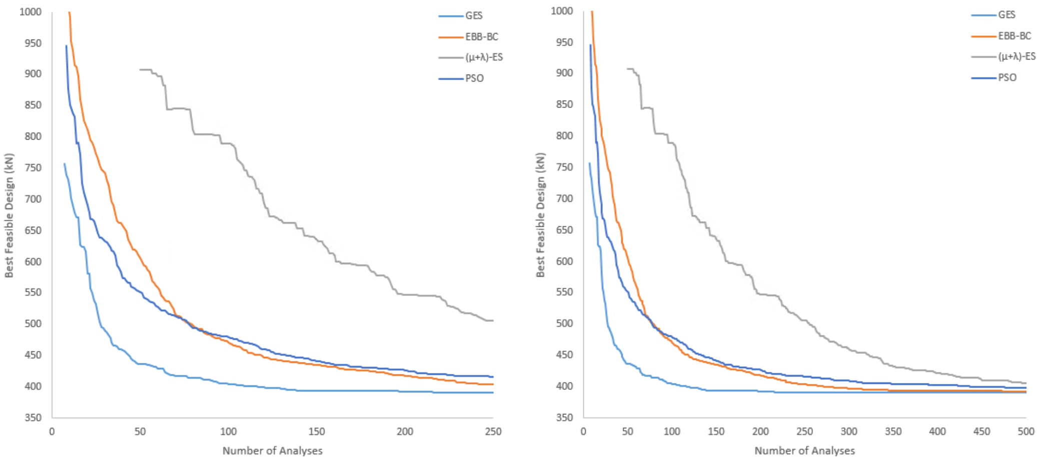
<table>
<tr><th colspan="8">Profile List 1 - I/Wide Flange</th></tr>
<tr><td>HE800M</td><td>HE700M</td><td>HE650M</td><td>HE600M</td><td>HE550M</td><td>HE500M</td><td>HE450M</td><td>HEA700</td></tr>
<tr><td>HE800B</td><td>HE400M</td><td>HE900A</td><td>HE360M</td><td>HE340M</td><td>HE320M</td><td>HE700B</td><td>HE300M</td></tr>
<tr><td>HE800A</td><td>HE650B</td><td>HE600B</td><td>HE700A</td><td>HE550B</td><td>HE650A</td><td>HE280M</td><td>HE500B</td></tr>
<tr><td>HE600A</td><td>HE300C</td><td>HE600x174</td><td>HE260M</td><td>HE450B</td><td>HE550A</td><td>HE700x166</td><td>HE240M</td></tr>
<tr><td>HE500A</td><td>HE400B</td><td>HE600x151</td><td>HE360B</td><td>HE450A</td><td>HE600x137</td><td>HE340B</td><td>HE320B</td></tr>
<tr><td>HE400A</td><td>HE450x123</td><td>HE300B</td><td>HE220M</td><td>HE360A</td><td>HE400x107</td><td>HE340A</td><td>HE200M</td></tr>
<tr><td>HE280B</td><td>HE320A</td><td>HE260B</td><td>HE180M</td><td>HE300A</td><td>HE240B</td><td>HE280A</td><td>HE160M</td></tr>
<tr><td>HE220B</td><td>HE260A</td><td>HE200B</td><td>HE240A</td><td>HE180B</td><td>HE220A</td><td>HE160B</td><td>HE200A</td></tr>
<tr><td>HE180A</td><td>HE140B</td><td>HE160A</td><td>HE120B</td><td>HE140A</td><td>HE100B</td><td>HE120A</td><td>HE100A</td></tr>
<tr><th colspan="8">Profile List 2 - I/Wide Flange</th></tr>
<tr><td>IPE600</td><td>IPE300</td><td>IPE270</td><td>IPE240</td><td>IPE220</td><td>IPE200</td><td>IPE180</td><td>IPE160</td></tr>
<tr><td>IPE140</td><td>IPE120</td><td>IPE100</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><th colspan="8">Profile List 3 - Box/Tube</th></tr>
<tr><td colspan="2">RHS120X60X3.6</td><td colspan="2">RHS160X80X4</td><td colspan="2">RHS200X100X5</td><td colspan="2">RHS100X50X3</td></tr>
<tr><th colspan="8">Profile List 4 - Pipe</th></tr>
<tr><td colspan="2">CHS76.1X4</td><td colspan="2">CHS88.9X3</td><td colspan="2">CHS114.3X3</td><td colspan="2">CHS114.3X4</td></tr>
<tr><td colspan="2">CHS139.7X3.2</td><td colspan="6"></td></tr>
<tr><th colspan="8">Profile List 5 - Pipe</th></tr>
<tr><td colspan="2">TUBO-D273X5.6</td><td colspan="2">TUBO-D168.3X4</td><td colspan="4"></td></tr>
</table>

Table 5.28 Optimization statistics for the 551-member steel frame design example

Analysis Count		PSO	EBB-BC	(μ+λ)-ES	GES
250	Best	**390.94**	**390.94**	400.81	**390.94**
	Mean	409.55	397.52	421.36	391.75
	Worst	433.93	415.04	440.58	399.10
	STD	13.99	6.97	11.61	2.45
	CV	3.42	1.75	2.76	0.62
500	Best	**390.94**	**390.94**	**390.94**	**390.94**
	Mean	396.97	392.26	393.39	391.75
	Worst	428.37	397.71	401.28	399.10
	STD	11.53	2.03	3.63	2.45
	CV	2.91	0.52	0.92	0.62

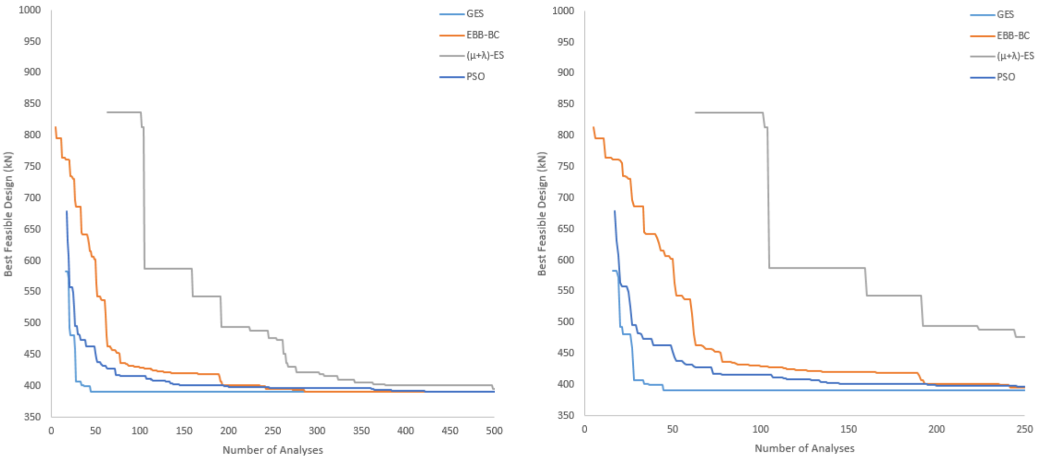
Table 5.29 The initial (original) and optimum designs of the 551-member steel frame design example

Groups	Sections in the Initial Design	Sections in the Optimum Design
RoofGirders1	HE200A	HE200A
RoofGirders2	HE300A	HE140A
RoofBeams1	RHS100X100X3	RHS100X50X3
LateralBraces1	CHS88.9X3	CHS88.9X3
FirstFloorGirders1	HE260A	HE260A
FirstFloorGirders2	HE320A	HE300A
FirstFloorBeams1	IPE200	IPE180
LateralBraces2	CHS114X3	CHS88.9X3
Columns1	HE260A	HE220A
Columns2	HE300A	HE320A
Columns3	HE300A	HE300A
VerticalBraces	TUBO-D168.3X4	TUBO-D168.3X4
Weight (kN)	**428.55**	**390.94**



(a) (b)

Figure 5.41 Average convergence curves obtained for the 551-member steel frame design example using various optimization techniques: (a) up to 250 analyses, (b) up to 500 analyses



(a) (b)

Figure 5.42 Convergence curves obtained for the 551-member steel frame design example in the best run of various optimization techniques: (a) up to 250 analyses, (b) up to 500 analyses

Table 5.30 The member groups' DCR values in the optimum design of the 551-member steel frame design example.

Groups	**DCR**
RoofGirders1	0.8202
RoofGirders2	0.9503
RoofBeams1	0.7257
LateralBraces1	0.5423
FirstFloorGirders1	0.9915
FirstFloorGirders2	0.9434
FirstFloorBeams1	0.8122
LateralBraces2	0.2580
Columns1	0.3416
Columns2	0.4311
Columns3	0.3641
VerticalBraces	0.4463
Average	0.6355

Table 5.31 Inter-story drift constraint values in the optimum design of the 551-member steel frame design example.

Story	Inter-Story Drift X-Dir.	Inter-Story Drift Y-Dir.
1	0.2428	0.0893
2	0.2664	0.0859
Average	0.2546	0.0876

5.3.3 Optimization Problem 3: 1634-Member Steel Structure

The third design example selected for a real-world application of structural optimization refers to the steel roof of a swimming pool complex located in Çankırı, Türkiye. The steel roof is shown in Figure 5.43 and it consists of 608 joints and 1634 structural members. The roof is designed in accordance with the provisions of AISC 360-10 design specifications. The structure is subjected to dead load plus a uniform roof cladding pressure of 50 kg/m^2 and a uniform snow pressure of 200 kg/m^2. The maximum wind load (WL) pressure is assumed as 80 kg/m2 and it is distributed to the roof with varying pressure levels depending on the location-wise slope of the roof, according to TS-EN 1991-1-4. The earthquake loads are applied to the steel roof by the equivalent lateral load procedure in the Turkish Seismic Code 2018 (TSC-2018), where the seismic coefficients are chosen as listed in Table 5.35. Load combinations applied to this structure are shown in Table 5.32. Unlike the previous examples, no inter-story drift constraint is considered in this example structure. Moreover, the design constraints are enforced as defined in detail in Chapter 2.

The original design process of the roof has been carried out by a design office, resulting in a total design weight of 2399.39 kN for the structure. However, it has been found that the original design violates demand-to-capacity ratio (DCR) for one member group in the order of 1.1016.

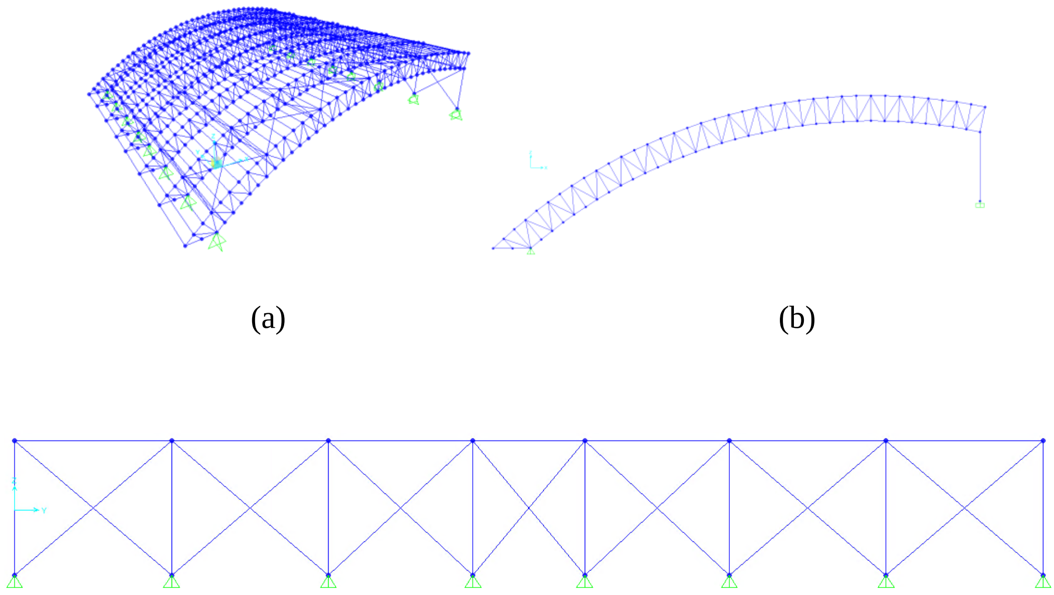
The grouping of the frame members is carried out in the same line with the way the steel has been formerly designed in practice. Accordingly, the 1634 members of the roof are collected under 12 member groups, as highlighted in Figure 5.44. Three different profile lists are defined as presented in Table 5.33 and used for sizing the 12 member groups in accordance with the original treatment of the design problem. Section of the member groups is selected from their assigned profile list and each member group's profile list is shown in Table 5.34. It is worth to mention that one of the member group (group-5) has a predefined section, i.e., its profile list contains only one section.

The GES algorithm is executed with the most effective parameter settings determined in the previous sections; that is, an exponentially decreasing value of the GML parameter is implemented with the GML_15-5-EXP scheme and also the GMR parameter is set to 0.5 (i.e., %50).

The steel roof is designed by executing ten independent runs with the GES algorithm as well as with each metaheuristic search technique employed here. In Table 5.36, the minimum weight designs (best feasible solutions) achieved for the frame are presented in terms of the best, worst, and average (mean) solution attained with each optimization technique up to some selected stages of the optimization process; namely 500 and 1000 number of structural analyses. Moreover, the standard deviation (STD) and coefficient of variation (CV) values are also calculated and presented in the related table. The results presented in Table 5.36 indicate a superior performance of the GES algorithm over the metaheuristic techniques at all stages of the optimization process.

In Figure 5.45, the (average) variation of the best design weight against the number of analyses (i.e., average convergence curve) is plotted for each optimization technique, after averaging the results of all runs performed with a particular method. Similarly, in Figure 5.46, convergence curves are plotted considering only the best performance of the methods for the problem of interest. The GES technique reaches the optimum design by performing the least number of structural analyses compared to metaheuristic search methods. Indeed, the GES algorithm seizes the optimum design (1429.03 kN) by performing only 168 structural analyses. The minimum design weight of the roof obtained with other methods at the same number of structural analysis (i.e., 168 structural analysis) is 1574.34 kN by EBB-BC, 1810.70 kN by PSO, and 2931.70 kN by (μ+λ)-ES. In Table 5.37, the original and optimum design of the steel roof are reported and compared with the section designations assigned to member groups.

In the optimum design of the steel frame, the computed DCR (demand-to-capacity ratio) values for member groups are given in Table 5.38. The DCR values, close to its max value of 1.0, indicate that DCRs mainly control the design variables (cross-sections of the member groups).

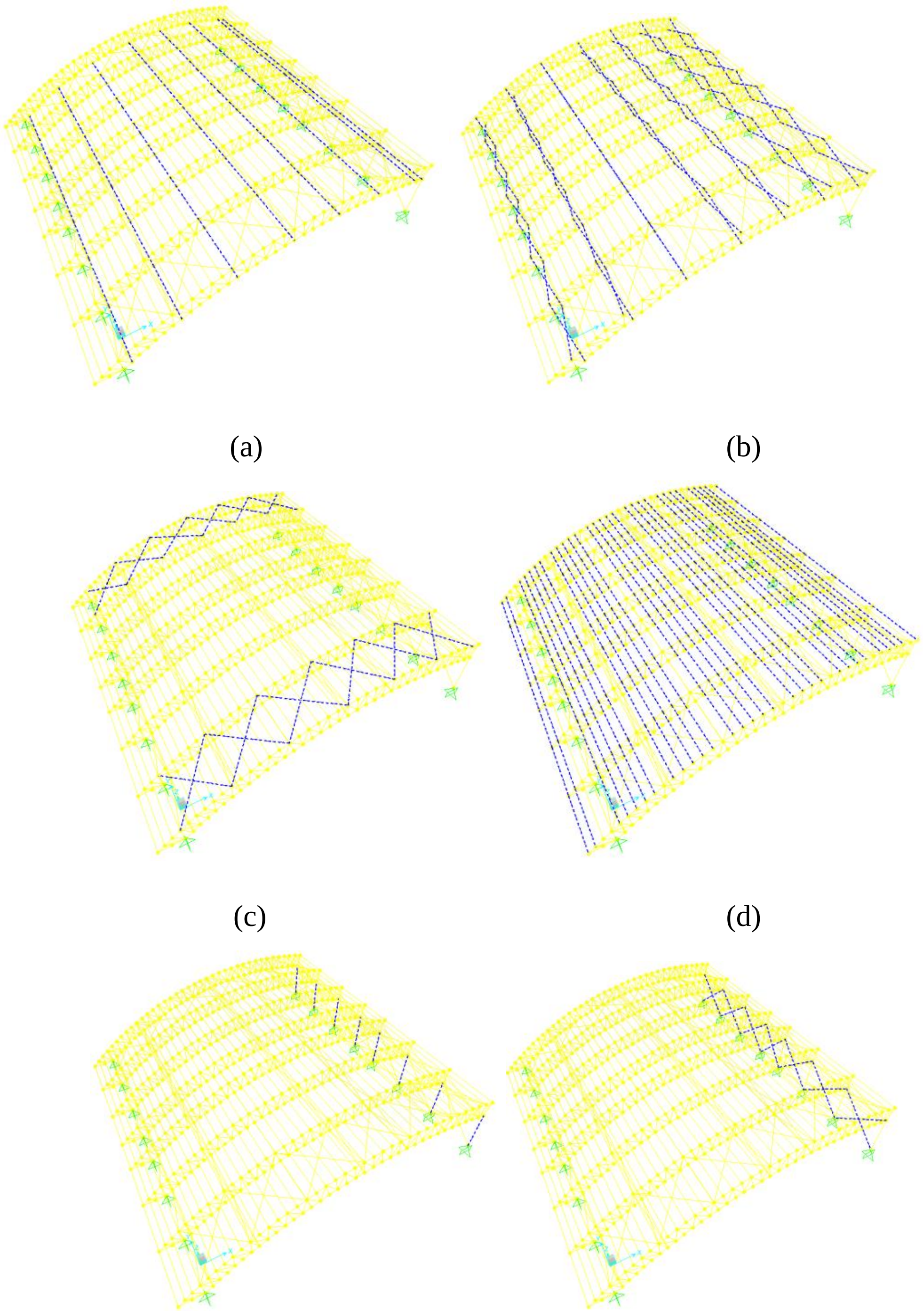


(a)

(b)

(c)

Figure 5.43 1634-member steel roof: (a) 3-D view, (b) side view in x-z plane, (c) side view in y-z plane.

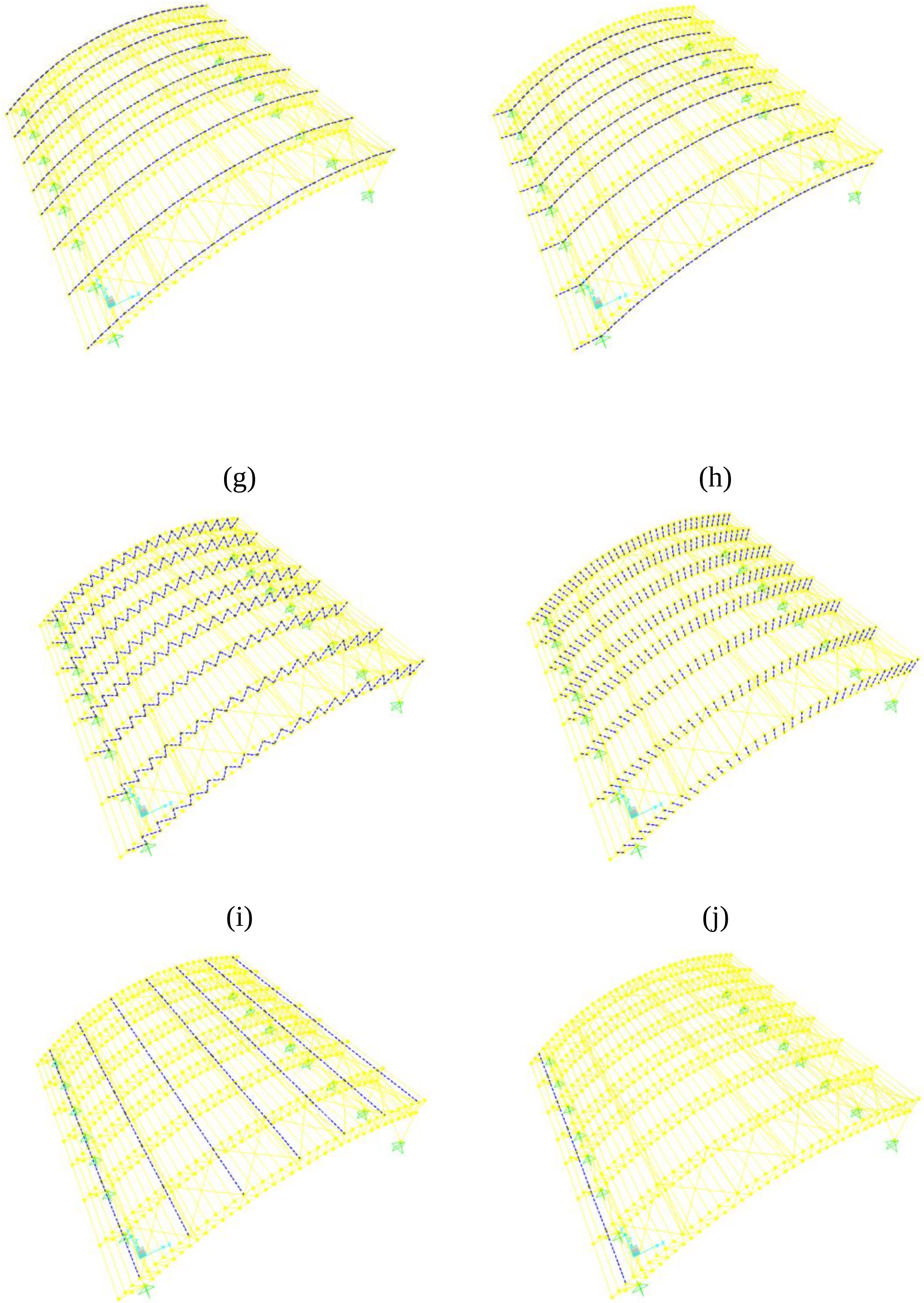


(a) (b)

(c) (d)

(e) (f)

Figure 5.44 The member groups 1-6 for the 1634-member steel roof: (a) group-1, (b) group-2, (c) group-3, (d) group-4, (e) group-5, (f) group-6, (g) group-7, (h) group-8, (i) group-9, (j) group-10, (k) group-11, (l) group-12



(g) (h)

(i) (j)

(k) (l)

Figure 5.44 (continued)

Table 5.32 Load combination definitions for the 1634-member steel roof design example

Load Combination Definitions	
1.4((Gk+t))+1.4G+1T*	1.4(Gk+t)+1.4G-1T
1.2(Gk+t)+1.2G+1.6S+1T	1.2(Gk+t)+1.2G+1.6S-1T
1.2(Gk+t)+1.2G+1.6S+0.8-Wx+1T*	1.2(Gk+t)+1.2G+1.6S+0.8-Wx-1T
1.2(Gk+t)+1.2G+1.6S+0.8+Wx+1T	1.2(Gk+t)+1.2G+1.6S+0.8+Wx-1T
1.2(Gk+t)+1.2G+1.6S+0.8-Wy+1T	1.2(Gk+t)+1.2G+1.6S+0.8-Wy-1T
1.2(Gk+t)+1.2G+1.6S+0.8+Wy+1T	1.2(Gk+t)+1.2G+1.6S+0.8+Wy-1T
1.2(Gk+t)+1.2G+0.5S+1.6-Wx+1T	1.2(Gk+t)+1.2G+0.5S+1.6-Wx-1T
1.2(Gk+t)+1.2G+0.5S+1.6+Wx+1T	1.2(Gk+t)+1.2G+0.5S+1.6+Wx-1T
1.2(Gk+t)+1.2G+0.5S+1.6-Wy+1T	1.2(Gk+t)+1.2G+0.5S+1.6-Wy-1T
1.2(Gk+t)+1.2G+0.5S+1.6+Wy+1T	1.2(Gk+t)+1.2G+0.5S+1.6+Wy-1T
0.9(Gk+t)+0.9G+1.6-Wx+1T	0.9(Gk+t)+0.9G+1.6-Wx-1T
0.9(Gk+t)+0.9G+1.6+Wx+1T	0.9(Gk+t)+0.9G+1.6+Wx-1T
0.9(Gk+t)+0.9G+1.6-Wy+1T	0.9(Gk+t)+0.9G+1.6-Wy-1T
1.351(Gk+t)+1.351G+0.2S+1Ex+0.3Ey+1T*	1.351(Gk+t)+1.351G+0.2S+1Ex+0.3Ey-1T
1.351(Gk+t)+1.351G+0.2S-1Ex-0.3Ey+1T	1.351(Gk+t)+1.351G+0.2S-1Ex-0.3Ey-1T
1.351(Gk+t)+1.351G+0.2S-1Ey+0.3Ex+1T	1.351(Gk+t)+1.351G+0.2S-1Ey+0.3Ex-1T
1.351(Gk+t)+1.351G+0.2S+1Ey+0.3Ex+1T	1.351(Gk+t)+1.351G+0.2S+1Ey+0.3Ex-1T
0.675(Gk+t)+0.675G+1Ex+0.3Ey+1T	0.675(Gk+t)+0.675G+1Ex+0.3Ey-1T
0.675(Gk+t)+0.675G-1Ex+0.3Ey+1T	0.675(Gk+t)+0.675G-1Ex+0.3Ey-1T
0.675(Gk+t)+0.675G-1Ey+0.3Ex+1T	0.675(Gk+t)+0.675G-1Ey+0.3Ex-1T
0.675(Gk+t)+0.675G+1Ey-0.3Ex+1T	0.675(Gk+t)+0.675G+1Ey-0.3Ex-1T
1.351(Gk+t)+1.351G+0.2S+1Ex-0.3Ey+1T	1.351(Gk+t)+1.351G+0.2S+1Ex-0.3Ey-1T
1.351(Gk+t)+1.351G+0.2S-1Ex+0.3Ey+1T	1.351(Gk+t)+1.351G+0.2S-1Ex+0.3Ey-1T
1.351(Gk+t)+1.351G+0.2S-1Ey-0.3Ex+1T	1.351(Gk+t)+1.351G+0.2S-1Ey-0.3Ex-1T
1.351(Gk+t)+1.351G+0.2S+1Ey-0.3Ex+1T	1.351(Gk+t)+1.351G+0.2S+1Ey-0.3Ex-1T
0.675(Gk+t)+0.675G+1Ex-0.3Ey+1T	0.675(Gk+t)+0.675G+1Ex-0.3Ey-1T
0.675(Gk+t)+0.675G-1Ex-0.3Ey+1T	0.675(Gk+t)+0.675G-1Ex-0.3Ey-1T
0.675(Gk+t)+0.675G-1Ey-0.3Ex+1T	0.675(Gk+t)+0.675G-1Ey-0.3Ex-1T

*G:Dead Loads; Gk+t: Cladding; Ex, Ey: Earthquake in x-direction, y-direction; S: Snow Load; T: Temperature; Wx, Wy: Wind Load in x-direction, y-direction

Table 5.33 Profile Lists for the 1634-member steel roof design example

<table>
<tr><th colspan="8">Profile List 1 - I/Wide Flange</th></tr>
<tr><td>HE100A</td><td>HE120A</td><td>HE100B</td><td>HE140A</td><td>HE120B</td><td>HE160A</td><td>HE140B</td><td>HE180A</td></tr>
<tr><td>HE200A</td><td>HE160B</td><td>HE220A</td><td>HE180B</td><td>HE240A</td><td>HE200B</td><td>HE260A</td><td>HE220B</td></tr>
<tr><td>HE280A</td><td>HE240B</td><td>HE300A</td><td>HE260B</td><td>HE320A</td><td>HE280B</td><td>HE340A</td><td>HE360A</td></tr>
<tr><td>HE300B</td><td>HE400A</td><td>HE320B</td><td>HE340B</td><td>HE450A</td><td>HE360B</td><td>HE400B</td><td>HE500A</td></tr>
<tr><td>HE550A</td><td>HE450B</td><td>HE600A</td><td>HE500B</td><td>HE650A</td><td>HE550B</td><td>HE700A</td><td>HE600B</td></tr>
<tr><td>HE650B</td><td>HE800A</td><td>HE700B</td><td>HE900A</td><td>HE800B</td><td>HE1000A</td><td>HE900B</td><td>HE1000B</td></tr>
<tr><th colspan="8">Profile List 2 - Channel</th></tr>
<tr><td>UPN80</td><td>UPN100</td><td>UPN120</td><td>UPN140</td><td>UPN160</td><td>UPN180</td><td>UPN200</td><td>UPN220</td></tr>
<tr><td>UPN240</td><td>UPN260</td><td>UPN280</td><td>UPN300</td><td>UPN320</td><td>UPN350</td><td>UPN380</td><td>UPN400</td></tr>
<tr><th colspan="8">Profile List 3 - Pipe</th></tr>
<tr><td colspan="2">CHS60.3x3.0</td><td colspan="2">CHS76.1x3.0</td><td colspan="2">CHS88.9x3.0</td><td colspan="2">CHS101.6x3.0</td></tr>
<tr><td colspan="2">CHS114.3x3.0</td><td colspan="2">CHS127.0x3.0</td><td colspan="2">CHS133.0x3.0</td><td colspan="2">CHS139.7x3.0</td></tr>
<tr><td colspan="2">CHS159.0x3.0</td><td colspan="2">CHS165.1x4.0</td><td colspan="2">CHS168.3x4.0</td><td colspan="2">CHS177.8x4.0</td></tr>
<tr><td colspan="2">CHS193.7x4.0</td><td colspan="2">CHS168.3x5.0</td><td colspan="2">CHS219.1x4.0</td><td colspan="2">CHS219.1x5.0</td></tr>
<tr><td colspan="2">CHS244.5x5.0</td><td colspan="2">CHS273.0x5.0</td><td colspan="2">CHS323.9x6.0</td><td colspan="2">CHS339.7x6.0</td></tr>
<tr><td colspan="2">CHS273.0x8.0</td><td colspan="2">CHS368.0x8.0</td><td colspan="2">CHS419.0x8.0</td><td colspan="2">CHS457.2x10.0</td></tr>
<tr><th colspan="8">Profile List 4 - SD Section</th></tr>
<tr><td colspan="8">Y900IST</td></tr>
</table>

Table 5.34 Profile Lists assigned for member groups for the 1634-member steel roof design example

Groups	**Profile List Number**
1	1
2	3
3	3
4	2
5	4
6	3
7	1
8	1
9	3
10	3
11	1
12	2

Table 5.35 Earthquake seismic coefficients used for calculating earthquake loads acting on the 1634-member steel roof

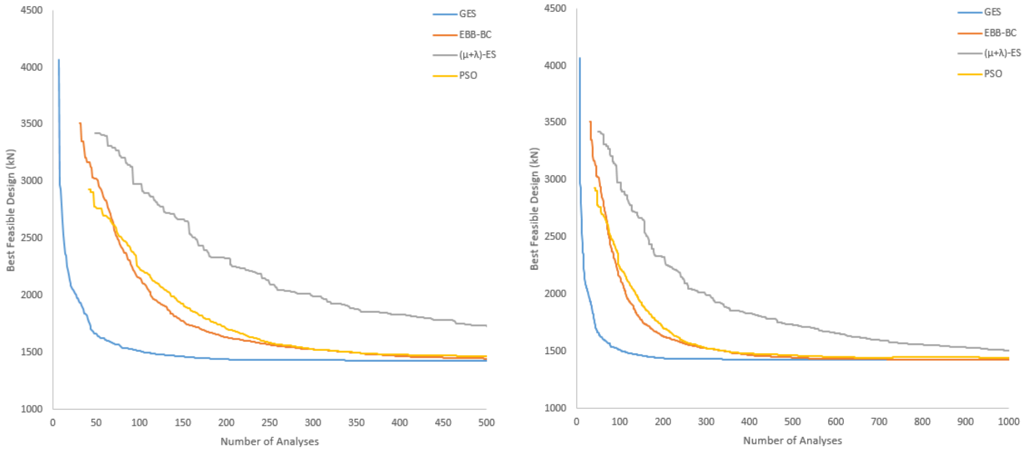
Seismic Coefficients	
0.2 Sec Spectral Accel, S_s	0.695
0.1 Sec Spectral Accel, S_1	0.222
Long-Period Transition Period	6
Site Class	ZE
Site Coefficient, F_a	1.388
Site Coefficient, F_v	3.19
$SD_S = F_a S_s$	0.9647
$SD_1 = F_v S_1$	0.7082

Table 5.36 Optimization statistics for the 1634-member steel roof design example

Analysis Count		**PSO**	**EBB-BC**	**(μ+λ)-ES**	**GES**
500	Best	**1429.03**	**1429.03**	1431.90	**1429.03**
	Mean	1465.44	1442.52	1598.97	1431.26
	Worst	1539.45	1515.36	1673.81	1441.86
	STD	34.93	24.89	65.12	4.53
	CV	23.37	16.92	39.94	3.10
1000	Best	**1429.03**	**1429.03**	**1429.03**	**1429.03**
	Mean	1437.31	1429.03	1471.19	1430.31
	Worst	1511.85	1429.03	1519.24	1441.86
	STD	24.85	0.00	35.34	3.85
	CV	16.95	0.00	23.56	2.64

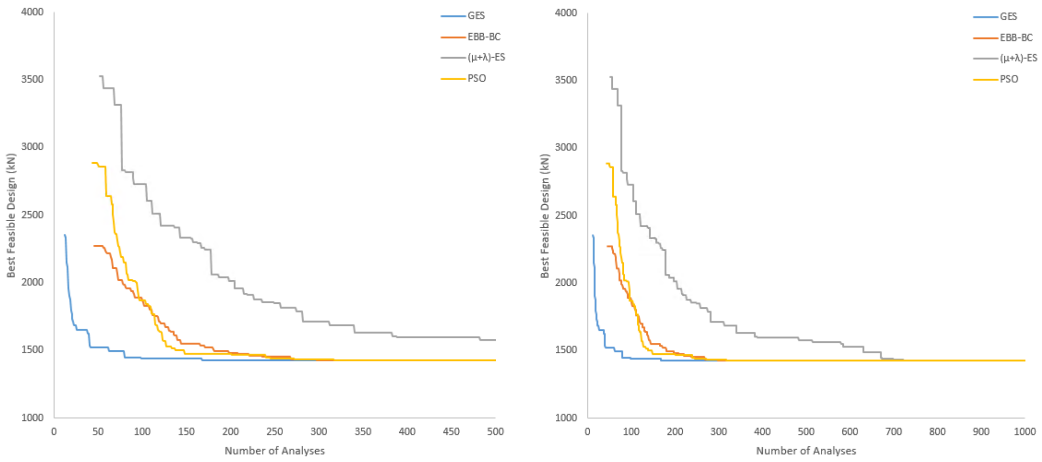
Table 5.37 The initial (original) and optimum designs of the 1634-member steel roof design example

Groups	**Sections in the Initial Design**	**Sections in the Optimum Design**
1	HE240A	HE200A
2	CHS168.3X6.0	CHS 159,0x3,0
3	CHS193X12.0	CHS 219,1x4,0
4	UPN220	UPN200
5	Y900IST	Y900IST
6	CHS168.3X6.0	CHS 177,8x4,0
7	HE300A	HE120A
8	HE300A	HE240A
9	HE160A	CHS 127,0x3,0
10	HE180A	CHS 127,0x3,0
11	HE240A	HE160A
12	UPN220	UPN260
Weight (kN)	**2399.39**	**1429.03**



(a) (b)

Figure 5.45 Average convergence curves obtained for the 1634-member steel roof design example using various optimization techniques: (a) up to 500 analyses, (b) up to 1000 analyses



(a) (b)

Figure 5.46 Convergence curves obtained for the 1634-member steel roof design example in the best run of various optimization techniques: (a) up to 500 analyses, (b) up to 1000 analyses

Table 5.38 The member groups' DCR values in the optimum design of the 1634-member steel roof design example.

Groups	**DCR**
1	0.6494
2	0.7710
3	0.9374
4	0.9998
5	0.9142
6	0.9154
7	0.9846
8	0.9273
9	0.9328
10	0.9632
11	0.8598
12	0.8938
Average	0.8957

CHAPTER 6

CONCLUSION

6.1 Summary and Concluding Remarks

This study covers developing an efficient and robust discrete sizing optimization technique and the software platform for optimum design of real-world steel structures subjected to the strength, displacement, and geometric constraints under the conventional design codes. Firstly, computationally efficient, a novel design-driven discrete sizing optimization technique called Guided Evolution Strategy (GES) is introduced. Secondly, the practical, efficient and robust infrastructure that brings the potential of the optimization techniques with the design software capabilities (SAP2000) together into a single software package called “Structural Optimization Platform Software (SOPS)” is developed. Thirdly, besides the GES technique, various metaheuristic techniques are implemented into SOPS to compare outcomes of GES with the other metaheuristics techniques, such as particle swarm optimization (PSO), exponential big bang big crunch (EBB-BC; enhanced version of the big-bang big crunch algorithm), Evolution Strategies with two variants, namely (μ,λ)-ES and $(\mu+\lambda)$-ES.

In chapter 3, the working principles of the GES technique are explained. The GES involves two tuning parameters that significantly affect its performance. The first one, Guided Mutation Limit (GML), is the upper bound of the design variables subjected to mutation. The second one, the Guided Mutation Ratio (GMR), is the ratio of guided offspring. Since the poor configuration of the GES parameter settings may lead to slow or premature convergence, in the first part of chapter 5, these two parameters are examined for their most efficient values. The results show that the GES algorithm is quite sensitive to its parameters. While GML provides the relative rate of local/global convergence performance, the intensity of the GES technique's guidance is determined by the GMR parameter. Therefore, these parameters should be adjusted as recommended in chapter 5 to get a more qualified solution.

Chapter 4 introduces "Structural Optimization Platform Software (SOPS)" developed in this study. Firstly, the interaction of optimization modules of SOPS with the external design software (SAP2000) through the Application Programming Interface (API) is presented. Secondly, the SOPS architecture and its fundamental working principles are explained. Finally, the modules of SOPS (such as Input/Output, Constraints, and Optimization) are described. The application of SOPS to the optimization problems demonstrates that SOPS successfully combines the potential of the optimization techniques and the design software capabilities (SAP2000). Moreover, due to the ability of its real-time monitoring, the performance of the optimization techniques, convergence rate, and stagnation durations can be easily observed in real-time while the optimization process continues in the background.

In chapter 5, the performance of the proposed Guided Evolution Strategy (GES) is compared with other metaheuristics; particle swarm optimization (PSO), exponential big bang big crunch (EBB-BC; enhanced version of the big-bang big crunch algorithm), Evolution Strategies with two variants, namely (μ,λ)-ES and $(\mu+\lambda)$-ES in terms of convergence rate and quality of the solutions. The numerical outcomes demonstrate the computational efficiency of the proposed technique (GES) for the optimal design of real-world steel structures subjected to the strength, displacement, and geometric constraints following the conventional design codes. Moreover, the results also show that the proposed technique (GES) has a faster convergence rate and reaches an optimum or reasonable near-optimum solution using less analyses. The results also demonstrate that applying optimization techniques to real-world structures can save significant construction materials and costs. The initial designs carried out by a design office are optimized using the optimization technique (GES) and the results show that for the three real-world structures, 6.97%, 8.78%, and 40.44% savings in material (steel) usage are obtained. Therefore, the GES optimization technique with SOPS can satisfy the needs of practical engineers' efficient, functional and robust optimization infrastructure.

As a result of this study, the following remarks can be pointed out:

- The proposed GES algorithm is a fast, robust, easy-to-implement, design-driven optimization algorithm.
- GES can handle not only strength and displacement constraints but also geometric constraints.
- GES can obtain code-compliant, cost-efficient designs with a fast convergence rate and fewer structural analyses than required for other metaheuristics.
- Due to its selection scheme, GES can utilize the UBS efficiently, diminishing unnecessary structural evaluations.
- SOPS successfully integrates optimization computing software (optimization algorithms) and commercial design software packages (SAP2000) to automate the optimization and design procedures.
- Besides the GES technique, various metaheuristic techniques are also implemented into SOPS, such as particle swarm optimization (PSO), exponential big bang big crunch (EBB-BC; enhanced version of the big-bang big crunch algorithm), Evolution Strategies with two variants, namely (μ,λ)-ES and $(\mu+\lambda)$-ES. Therefore, SOPS can solve optimization design problems utilizing various optimization algorithms and compare these solutions.
- SOPS provides real-time monitoring of the optimization process such as convergence rate and stagnation durations to examine the performance of the optimization techniques more efficiently.

REFERENCES

Ahrari, A., & Atai, A. A. (2013). Fully stressed design evolution strategy for shape and size optimization of truss structures. Computers & Structures, 123, 58-67.

Ahrari, A., & Deb, K. (2016). An improved fully stressed design evolution strategy for layout optimization of truss structures. Computers & Structures, 164, 127-144.

Ahrari, A., Atai, A. A., & Deb, K. (2015). Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy. Engineering Optimization, 47(8), 1063-1084.

Aldwaik, M., & Adeli, H. (2014). Advances in optimization of high rise building structures. Structural and Multidisciplinary Optimization, 50(6), 899-919.

American Institute of Steel Construction. (2010). ANSI/AISC 360-10: Specification for Structural Steel Buildings, Chicago, Illinois, USA.

American Society of Civil Engineers. (2010). ASCE/SEI: Minimum design loads for building and other structures.

Bäck, T. (1996). Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York.

Bäck, T., & Schütz, M. (1995). Mixed-Integer Optimization of Optical Multilayer Systems. In Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming (p. 33). MIT Press.

Bochenek, B., & Foryś, P. (2006). Structural optimization for post-buckling behavior using particle swarms. Structural and Multidisciplinary Optimization, 32(6), 521-531.

Cai, J., & Thierauf, G. (1993). Discrete structural optimization using evolution strategies. In: Topping BHV, Khan AI, editors. Neural networks and combinatorial optimization in civil and structural engineering. Edinburgh: Civil-Comp., 95-100.

Camp, C. V. (2007). Design of space trusses using Big Bang–Big Crunch optimization. Journal of Structural Engineering, 133(7), 999-1008.

Camp, C. V., Meyer, B. J., & Palazolo, P. J. (2004). Particle swarm optimization for the design of trusses. In Structures 2004: Building on the Past, Securing the Future (pp. 1-10).

Chen, G., Huang, X., Jia, J., & Min, Z. (2006, June). Natural exponential inertia weight strategy in particle swarm optimization. In 2006 6th world congress on intelligent control and automation (Vol. 1, pp. 3672-3675). IEEE.

Dimou, C. K., & Koumousis, V. K. (2009). Reliability-based optimal design of truss structures using particle swarm optimization. Journal of computing in civil engineering, 23(2), 100-109.

Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In MHS'95. Proceedings of the sixth international symposium on micro machine and human science (pp. 39-43). Ieee.

Erol, O. K., & Eksin, I. (2006). A new optimization method: big bang–big crunch. Advances in Engineering Software, 37(2), 106-111.

Fourie, P. C., & Groenwold, A. A. (2000, August). Particle swarms in size and shape optimization. In Proceedings of the international workshop on multidisciplinary design optimization, Pretoria, South Africa (pp. 97-106).

Fourie, P. C., & Groenwold, A. A. (2002). The particle swarm optimization algorithm in size and shape optimization. Structural and Multidisciplinary Optimization, 23(4), 259-267.

Fourie, P., & Groenwold, A. (2001, May). The particle swarm optimization in topology optimization. In Fourth world congress of structural and multidisciplinary optimization. Paper no. 154, Dalian, China

Gallagher, R. H., & Zienkiewicz, O. C. (1973). Optimum structural design: Theory and applications (Book- Optimum structural design: Theory and applications.). London, John Wiley and Sons, Ltd., 1973.

Gao, Y. L., An, X. H., & Liu, J. M. (2008, December). A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation. In 2008 international conference on computational intelligence and security (Vol. 1, pp. 61-65). IEEE.

Gomes, H. M. (2011). Truss optimization with dynamic constraints using a particle swarm algorithm. Expert Systems with Applications, 38(1), 957-968.

Hasançebi O. (2007). Discrete approaches in evolution strategies based optimum design of steel frames. Structural Engineering & Mechanics; 26(2), 191–210.

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

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

Hasançebi, O. Çarbaş, S., Doğan, E., Erdal, F. & Saka, M. P. (2009). Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. Computers & Structures, 87(5-6), 284-302.

Hasançebi, O. Çarbaş, S., Doğan, E., Erdal, F. & Saka, M. P. (2010). Comparison of non-deterministic search techniques in the optimum design of real size steel frames. Computers & Structures, 88(17-18), 1033-1048.

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 & Structures, 89(21-22), 2037-2051.

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., & Sabzi, O. (2011). A comparative study of two meta-heuristic algorithms for optimum design of reinforced concrete frames, Int J Civil Eng; 9(3): 193-206

Kaveh, A., & Talatahari, S. (2009a). Size optimization of space trusses using Big Bang–Big Crunch algorithm. Computers & structures, 87(17-18), 1129-1140.

Kaveh, A., & Talatahari, S. (2009b). A particle swarm ant colony optimization for truss structures with discrete variables. Journal of Constructional Steel Research, 65(8-9), 1558-1568.

Kaveh, A., & Talatahari, S. (2009c). Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization. In Harmony search algorithms for structural design optimization (pp. 159-198). Springer, Berlin, Heidelberg.

Kaveh, A., & Talatahari, S. (2010). A discrete big bang-big crunch algorithm for optimal design of skeletal structures, Asian J Civil Eng; 11: 103-22.

Kaveh, A., & Talatahari, S. (2010). Optimal design of Schwedler and ribbed domes via hybrid Big Bang–Big Crunch algorithm. Journal of Constructional Steel Research, 66(3), 412-419.

Kazemzadeh Azad, S., Hasançebi, O., & Erol, O. (2011). Evaluating efficiency of big-bang big-crunch algorithm in benchmark engineering optimization problems. Iran University of Science & Technology, 1(3), 495-505.

Kazemzadeh Azad, S., Hasançebi, O., & Kazemzadeh Azad, S. (2013). Upper bound strategy for metaheuristic based design optimization of steel frames. Advances in Engineering Software, 57, 19–32.

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.

Li, L. J., Huang, Z. B., & Liu, F. (2009). A heuristic particle swarm optimization method for truss structures with discrete variables. Computers & structures, 87(7-8), 435-443.

Li, L. J., Huang, Z. B., Liu, F., & Wu, Q. H. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. Computers & structures, 85(7-8), 340-349.

Naka, S., Genji, T., Yura, T., & Fukuyama, Y. (2001, January). Practical distribution state estimation using hybrid particle swarm optimization. In 2001 IEEE power engineering society winter meeting. conference proceedings (Cat. No. 01CH37194) (Vol. 2, pp. 815-820). IEEE.

Papadrakakis, M., Lagaros, N. D., & Fragakis, Y. (2003). Parallel computational strategies for structural optimization. International Journal for Numerical Methods in Engineering, 58(9), 1347-1380.

Patnaik, S. N., Berke, L., & Gallagher, R. H. (1991). Integrated force method versus displacement method for finite element analysis. Computers & Structures, 38(4), 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(7), 1171-1194.

Perez, R. E., & Behdinan, K. (2007). Particle swarm approach for structural design optimization. Computers & Structures, 85(19-20), 1579-1588.

Plevris, V., & Papadrakakis, M. (2011). A hybrid particle swarm—gradient algorithm for global structural optimization. Computer‐Aided Civil and Infrastructure Engineering, 26(1), 48-68.

Razani, R. (1965). Behavior of fully stressed design of structures and its relationshipto minimum-weight design. AIAA journal, 3(12), 2262-2268.

Rechenberg, I. (1965). Cybernetic solution path of an experimental problem. Royal Aircraft Establishment Library Translation 1122.

Rechenberg, I. (1973). Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution, frommann–holzboog. stuttgart Germany.

Saka, M. P., Hasançebi, O., & Geem, Z. W. (2016). Metaheuristics in structural optimization and discussions on harmony search algorithm. Swarm and Evolutionary Computation, 28, 88-97.

Sarma, K. C., & Adeli, H. (2001). Bilevel parallel genetic algorithms for optimization of large steel structures. Computer-Aided Civil and Infrastructure Engineering, 16(5), 295-304.

Seyedpoor, S. M., Gholizadeh, S., & Talebian, S. R. (2010). An efficient structural optimisation algorithm using a hybrid version of particle swarm optimisation with simultaneous perturbation stochastic approximation, Civil Engineering and Environmental Systems, Vol. 27, No. 4, pp. 295–313.

Schutte, J. F., & Groenwold, A. A. (2003). Sizing design of truss structures using particle swarms. Structural and Multidisciplinary Optimization, 25(4), 261-269.

Schwefel H-P. (1965). Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik. Berlin: Diplomarbeit Technische Universität.

Schwefel H-P. (1977). Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie, Interdisciplinary systems research; 26. Birkhäuser, Basel.

Schwefel H-P. (1981). Numerical optimization of computer models. Chichester: Wiley.

Shi, Y., & Eberhart, R. (1998, May). A modified particle swarm optimizer. In 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360) (pp. 69-73). IEEE.

Shi, Y., & Eberhart, R. C. (1999, July). Empirical study of particle swarm optimization. In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 3, pp. 1945-1950). IEEE.

Sigmund, O. (2011). On the usefulness of non-gradient approaches in topology optimization. Structural and Multidisciplinary Optimization, 43(5), 589-596.

Yang, C., Gao, W., Liu, N., & Song, C. (2015). Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight. Applied Soft Computing, 29, 386-394.

Yang, X. S. (2010). Nature-inspired metaheuristic algorithms. Luniver press: Beckington, UK.

CURRICULUM VITAE

PERSONEL INFORMATION

Surname, Name: Korucu, Aytaç

EDUCATION

Degree	**Institution**	**Year of Graduation**
MS	METU Civil Engineering	1999
BS	METU Civil Engineering	1997
High School	Ankara Atatürk High School	1992

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2007-2022	SGK IT Cyber Security Dept.	IT-Expert
2000-2007	Microsoft Certified Education Center	Microsoft Trainer
1997-2000	METU Dept. Of Civil Engineering	Research Assistant

FOREIGN LANGUAGES

Fluent English, Basic German