

DAMAGE-TOLERANT OPTIMAL DESIGN OF STRUCTURES SUBJECTED TO BLAST
LOADING

by

Mustafa Chasib Jasim Al-Bazoon

A thesis submitted in partial fulfillment
of the requirements for the Doctor of Philosophy
degree in Civil and Environmental Engineering in the
Graduate College of
The University of Iowa

August 2019

Thesis Supervisor: Professor Jasbir S. Arora

ProQuest Number:13807220

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 13807220

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Copyright by

Mustafa Chasib Jasim Al-Bazoon

2019

All Rights Reserved

To my beloved Mother, Family, and Friends for supporting me all the way

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Professor Jasbir S. Arora for the continues guidance and support throughout my time at the University of Iowa. I could not have kept pursuing my PhD research without his support. I am lucky and proud to be a student of his. I would also like to express my thanks to my mathematics advisor Professor Laurent O. Jay for his guidance. I would like to thank Professors Colby Swan, Kyung Choi, M. Asghar Bhatti, and Salam Rahmatalla for serving on my thesis defense committee.

I would like to appreciate the Department of Civil and Environmental Engineering at the University of Iowa for giving me the opportunity be part of their graduate program. Also, I am grateful to the Center for Computer-Aided Design at University of Iowa for the support.

ABSTRACT

An explosion is characterized as a sudden release of large energy over a very short duration. As the blast wave travels parallel to a surface, it creates a side-on pressure and when it hits a surface perpendicularly or at an angle, it creates a reflected pressure. Side-on pressure and reflected pressure are much higher than service loads for the structure. Thus, when a blast happens near a building that is not designed to withstand blast loads, it can cause catastrophic damage.

The objective of this study is to present a formulation for the design optimization of framed steel structures subjected to blast loads. Also, a formulation is presented for the design optimization of structures that can withstand some possible damage due to blast loads. To this end, an optimization procedure that includes definitions of design variables, cost function, constraints, and structural analyses is discussed. The design variables for beams and columns are the discrete values of the W-shapes selected from American Institute of Steel Construction (AISC) tables. The optimization problem is to minimize the total structural weight subjected to AISC strength requirements and blast design displacement constraints. Linear static, linear dynamic, and nonlinear dynamic analyses are incorporated in the optimization process and optimum designs are compared. Due to design variables and some constraints discontinuity, gradient-based optimization algorithms cannot be used to solve the optimization problem.

Therefore, metaheuristic algorithms are used that require only simulation results to solve problems with discrete variables and non-differentiable functions. Since the number of simulations and robustness to obtain good designs are important for the class of problems discussed in this research, a new hybrid optimization algorithm based on Harmony Search (HS) and Colliding Bodies Optimization (CBO) is developed and examined. The algorithm is named Hybrid Harmony Search - Colliding Bodies Optimization (HHC). Also, a novel design domain reduction technique

is incorporated in HHC. Some benchmark discrete variable structural design problems are used to evaluate HHC. In comparison with some popular metaheuristic optimization algorithms, HHC is shown to be robust, effective, and needs fewer structural analyses to obtain the best designs.

Depending on the size of the structure to be designed, optimization of structures that require linear or nonlinear dynamic analyses using metaheuristic algorithms can be computationally expensive because these types of algorithms need large number of simulations to reach good designs. Equivalent Static Loads (ESL) approach, which has been used for optimization of structural systems subjected to dynamic loads using gradient-based algorithms, is examined for optimization of structures that have discrete design variables using metaheuristic algorithms. The proposed approach is named global optimization with equivalent static loads (GOESL). Solution of four numerical examples shows that GOESL can drastically reduce the number of dynamic analyses needed to reach the best design compared to an algorithm without the ESL approach. However, the ESL step alone cannot converge to the best design for the current formulation, even with many ESL cycles. Therefore, after a few ESL cycles, the procedure may switch to the original algorithm without the ESL cycles to improve designs further.

HHC and GOESL are used to solve three-dimensional framed steel structures subjected to blast loads with linear and nonlinear dynamic analyses as separate solution cases. The source of the blast loads is a car carrying 250 lbs of Trinitrotoluene (TNT) with 50 ft standoff distance from the front face of a 4-bay x 4-bay x 3-story building. Optimum designs of the structure to withstand blast loads show that penalty on the optimum structural weight is substantial when linear dynamic analysis is used. With nonlinear dynamic analysis, the penalty on the structural weight is substantially reduced. When the stiffness of the walls is included in the analysis model, there is

very little penalty on the optimum structural weight with linear or nonlinear dynamic analysis models.

The best designs obtained with the linear and nonlinear dynamic analysis models are checked for some possible damages due to a blast. Two types of damage conditions are defined: (i) complete removal of some key members from the analysis model, and (ii) reduction of stiffness of some members. It is shown that the best designs using linear or nonlinear dynamic analyses can withstand all damage conditions. Thus, resilience of the designs to withstand blast loads is observed.

PREVIEW

PUBLIC ABSTRACT

Blast-resistant analysis and design of structures has received greater attention during the last few decades as the terrorist attacks showed the great damage that could happen due to an explosion near the structure. In this study, optimum design of steel building to withstand blast loads is studied and discussed. The problem is formulated to minimize total structural cost while requiring sufficient strength to withstand blast loads. Since the computational effort to solve this problem can be substantial, considerable research is conducted to develop and evaluate more efficient procedures for numerical solution of the problem. It is shown that depending on the fidelity of structural analysis models used and the performance requirements for the structure, the penalty on the structural cost to withstand the blast loads can vary from substantial to minimal. Directions for future research to further improve the computational procedures are presented.

TABLE OF CONTENTS

LIST OF TABLES	xiii
LIST OF FIGURES	xvi
LIST OF ACRONYMS	xix
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Blast Phenomena and Blast Loads	1
1.3 Optimization of Framed Structures Subjected to Blast Loading	2
1.4 Optimum Design of Framed Structures for Damage Tolerance	3
1.5 Optimization Algorithms	3
1.6 Optimization of Structures Subjected to Dynamic Loads Using Equivalent Static Loads.....	4
1.7 Motivation and Purpose	4
1.8 Scope of Thesis	5
CHAPTER 2: DESIGN OF STRUCTURES SUBJECTED TO BLAST LOADS: ANALYSIS AND DESIGN REVIEW	7
Abstract.....	7
2.1 Introduction.....	7
2.2 Literature Review.....	8
2.3 Prediction of Blast Loading	9
2.3.1 Blast Phenomena.....	9
2.3.2 Scaling Law	12
2.3.3 Explosion and Blast-Loading Types	13
2.3.4 Blast Wave Reflection	14
2.3.5 Surface Burst and Loading.....	15
2.3.6 Negative Phases	19

2.3.7 Internal Pressure.....	19
2.4 Material Design Strength	19
2.4.1 Material Properties for Steel	20
2.4.2 Material Properties for Reinforce Concrete.....	21
2.4.3 Plastic Hinge	22
2.5 Strength Reduction Factors and Load Combination.....	25
2.6 Blast Load and Structure Interaction (Structural Response).....	25
2.6.1 Pressure-Impulse Charts	26
2.6.2 The Single-Element Analysis Method.....	27
2.6.3 Multi Degree of Freedom Finite Element System.....	28
2.6.4 Equivalent Static Load Method	30
2.7 Criteria for Responses (Response Limits).....	30
2.7.1 Design Criteria for Individual Elements.....	33
2.7.2 Design Criteria for Structural System.....	33
2.8 Progressive Collapse.....	34
2.9 Concluding Remarks.....	34
CHAPTER 3: METAHEURISTIC OPTIMIZATION ALGORITHMS	35
Abstract.....	35
3.1 Introduction.....	36
3.2 Formulation of Discrete Structural Optimization Problems	38
3.3 Metaheuristic Optimization Algorithms	39
3.3.1 Harmony Search Algorithm.....	40
3.3.2 Improved Harmony Search Algorithm (IHS)	43
3.3.3 Colliding Bodies Optimization (CBO)	44
3.4 HHC: Hybrid Improved Harmony Search-Enhanced Colliding Bodies Algorithm	49

3.4.1 Motivation for Hybrid Algorithm	49
3.4.2 Phase 1: Improved Harmony Search (IHS)	49
3.4.3 Phase 2: Enhanced Colliding Bodies Optimization (ECBO).....	51
3.4.4 Domain Adjustment Technique	52
3.4.5 Evaluation of the Algorithms.....	54
3.5 Numerical Examples	55
3.5.1 Planar 10-bar Truss	57
3.5.2 Planar 15-bar Truss	62
3.5.3 Planar 52-bar Truss	65
3.5.4 Spatial 25-bar Truss	68
3.5.5 Spatial 72-bar Truss	71
3.6 Discussion and Conclusions	75
3.7 Reproducing Results	77
CHAPTER 4: DISCRETE VARIABLE OPTIMIZATION OF STRUCTURES SUBJECTED TO DYNAMIC LOADS USING EQUIVALENT STATIC LOADS AND METAHEURISTIC ALGORITHMS	78
Abstract	78
4.1 Introduction.....	78
4.2 Objective of This Work	81
4.3 Transformation of Dynamic Loads into Equivalent Static Loads (ESLs)	82
4.4 Formulation for Discrete Structural Optimization Problems.....	84
4.5 Enhanced Colliding Bodies Optimization (ECBO)	86
4.6 Discrete Variable Optimization Using ESL for Transient Problems.....	87
4.7 Numerical Examples	94
4.7.1 Eighteen-bar Truss	95
4.7.2 Ten-bar Truss with Material Nonlinearity	101

4.7.3 Two-story Two-bay Frame	106
4.7.4 Two-story Two-bay Frame Subjected to Blast Loads with Material and Geometric Nonlinearity	111
4.8 Concluding Remarks.....	115
4.9 Reproducing Results	116
CHAPTER 5: OPTIMIZATION OF FRAMED STRUCTURES SUBJECTED TO BLAST LOADING	
Abstract.....	117
5.1 Introduction.....	118
5.2 Review of Literature	119
5.3 Blast Design	120
5.3.1 Material Design Strength	120
5.3.2 Strength Reduction Factors and Load Combinations	121
5.3.3 Performance Requirements.....	122
5.4 Blast Loading.....	122
5.5 Formulation for Discrete Structural Optimization Problems.....	124
5.5.1 Design Variables.....	127
5.5.2 Cost Function.....	127
5.5.3 Constraints	128
5.6 Discrete Variable Optimization of Structures Subjected to Dynamic Loads Using Equivalent Static Loads	131
5.7 Optimization Algorithms	131
5.7.1 Improved Harmony Search.....	132
5.7.2 Enhanced Colliding Bodies Optimization (ECBO).....	133
5.7.3 Hybrid Improved Harmony Search-Enhanced Colliding Bodies Algorithm (HHC)	134
5.8 Numerical Examples.....	134

5.8.1 Spatial Framed Steel Structure Subjected to Service Loads Only	137
5.8.2 Spatial Framed Steel Structure Subjected to blast Load	138
5.9 Concluding Remarks	147
CHAPTER 6: OPTIMUM DESIGN OF FRAMED STRUCTURES FOR RESILIENCE SUBJECTED TO BLAST LOADS	149
Abstract	149
6.1 Introduction	149
6.2 Review of Literature	151
6.3 Design for Blast Loads	151
6.4 Formulation for Discrete Structural Optimization Problems	152
6.4.1 Design Variables	154
6.4.2 Cost Function	154
6.4.3 Constraints	155
6.5 Optimization Process for Damage Tolerance	157
6.6 Damage-Tolerant Design of Framed Steel Structure	158
6.6.1 Complete Removal of Some Members	160
6.6.2 Strength Reduction of Some Members	162
6.7 Concluding remarks	163
CHAPTER 7: CONCLUSIONS AND FUTURE WORK	165
7.1 Discussion and Conclusion	165
7.2 Future Work	167
BIBLIOGRAPHY	169
APPENDIX A	174
MATLAB code for planar 10-bar truss structure	174
APPENDIX B	183
MATLAB code for the 18-bar truss design example	183

LIST OF TABLES

Table 2.1. Heat of detonation for some explosives (DoD, 2008).	12
Table 2.2. Estimated quantities of explosive (FEMA 426, 2003).	13
Table 2.3. Size of explosive (M. Bangash and T. Bangash, 2006).	13
Table 2.4. Dynamic increasing factor (<i>DIF</i>) for yield stress and ultimate stress for structural steel (DoD, 2008).	21
Table 2.5. Dynamic increase factor (<i>DIF</i>) for reinforced concrete design (DoD, 2008).	22
Table 2.6. Tension-Compression hinge parameters (FEMA, 2000 and Gilsanz et al., 2013).	24
Table 2.7. Damage and response level (ASCE, 2010).	31
Table 2.8. Response limits for different components* (ASCE, 2010).	33
Table 2.9. Side-sway limits for steel frame structures (ASCE, 2010).	33
Table 3.1. Domain adjustment technique illustration for planar 10-bar truss structure.	59
Table 3.2. Comparison of optimal designs for 10-bar truss problem.	61
Table 3.3. Comparison of optimal designs for 15-bar truss problem.	64
Table 3.4. Comparison of optimal designs for 52-bar truss problem.	66
Table 3.5. Load conditions of the spatial 25-bar truss.	68
Table 3.6. Performance comparison for the 25-bar spatial truss.	71
Table 3.7. Load conditions of the spatial 72-bar truss.	72
Table 3.8. Performance comparison for the 72-bar spatial truss.	74
Table 3.9. Comparative data for design examples.	75
Table 4.1. Data for 10 different runs of the 18-bar truss.	99
Table 4.2. Comparison of averages and standard deviations for 10 runs of the 18-bar truss.	100
Table 4.3. Initial and final designs of 18-bar truss for the first run.	100
Table 4.4. Data for 10 different runs of the 10-bar truss.	103
Table 4.5. Comparison of averages and standard deviations for 10 runs of the 10-bar truss.	104

Table 4.6. Initial and final design of 10-bar truss of the first run.	104
Table 4.7. ASIC W-Shapes Database (partial).	107
Table 4.8 Initial and final design of 2-story 2-bay frame of the first run.	110
Table 4.9. 10 individual runs data of the 2-story 2-bay frame.	110
Table 4.10. Initial and final design of 2-story 2-bay frame subjected to blast load of the first run.	114
Table 4.11. 10 individual runs data of the 2-story 2-bay frame subjected to blast load.	114
Table 5.1. Members grouping.	136
Table 5.2. Load combinations (AISC, 2015; Gilsanz et al., 2013).	136
Table 5.3. Final designs for 3D framed steel structure under service load only.	138
Table 5.4. Pressure-time history on faces.	140
Table 5.5. Final designs for 3D framed steel structure under service and blast loads (linear dynamic analysis).	142
Table 5.6. Final designs for 3D framed steel structure under service and blast loads with the mass of external walls (linear dynamic analysis).	143
Table 5.7. Final designs for 3D framed steel structure with external walls under service and blast loads (linear dynamic analysis).	144
Table 5.8. Final designs for 3D framed steel structure under service and blast loads (nonlinear dynamic analysis).	145
Table 5.9. Final designs for 3D framed steel structure under service and blast loads with the mass of the external walls (nonlinear dynamic analysis).	146
Table 5.10. Final designs for 3D framed steel structure with external walls under service and blast loads (nonlinear dynamic analysis).	147
Table 5.11. Comparison of final designs.	148
Table 6.1. Members grouping.	159
Table 6.2. Load combinations (AISC, 2015; Gilsanz et al., 2013).	159
Table 6.3. Damage condition definitions.	160
Table 6.4. Constraints evaluation of the optimal design using linear dynamic analysis for complete removal study case (From Table 5.5).	161

Table 6.5. Constraints evaluation of the optimal design using nonlinear dynamic analysis for complete removal study case (From Table 5.8)..... 161

Table 6.6. Constraints evaluation of the optimal design using linear dynamic analysis for strength reduction study case (From Table 5.5)..... 163

Table 6.7. Constraints evaluation of the optimal design using nonlinear dynamic analysis for strength reduction study case (From Table 5.8)..... 163

PREVIEW

LIST OF FIGURES

Figure 2.1. Idealized pressure-time profile for blast wave: (a) free field pressure (b) reflected pressure (modified from DoD, 2008).....	10
Figure 2.2. Types of explosions: (a) Free-air bursts. (b) Air bursts. (c) Surface bursts (modified from Karlos et al., 2013).	14
Figure 2.3. Influence of angle of incident on the reflection pressure (modified from DoD, 2008).	15
Figure 2.4. The positive phase parameters of hemispherical wave of TNT charges (DoD, 2008).	17
Figure 2.5. The negative phase parameters of hemispherical wave of TNT charges (modified from DoD, 2008).....	17
Figure 2.6. Triangular assumption of pressure time history on the (a) Front walls loading (b) Roof and sides walls loading (c) Rear walls loading (modified from DoD, 2008).	18
Figure 2.7. The effect of high strain rate on mechanical properties of steel (modified from DoD, 2008).	20
Figure 2.8. The effect of high strain rate on mechanical properties of concrete (modified from DoD, 2008).	22
Figure 2.9. Idealized resistance-deflection curve (Cormie et al., 2009).	23
Figure 2.10. Tension-compression hinge properties (FEMA, 2000 and Gilsanz et al., 2013).	24
Figure 2.11. Pressure-impulse diagram for elastic SDOF component (Smith & Hetherington, 1994).	26
Figure 2.12. Member (beam, slab, or panel) support rotations (DoD, 2008).	32
Figure 2.13. Frame support rotations and side-sway deflection (DoD, 2008).....	32
Figure 3.1. Schematic of 10-bar planar truss.	58
Figure 3.2. Comparison of convergence rates for planar 10-bar truss.....	60
Figure 3.3. Comparison of best designs from 50 runs for the 10-bar truss structure.....	62
Figure 3.4. Schematic of 15-bar planar truss.	63
Figure 3.5. Comparison of convergence rates for planar 15-bar truss.....	63
Figure 3.6. Comparison of best designs from 50 runs for the 15-bar truss structure.....	64

Figure 3.7. Schematic of 52-bar space truss.	66
Figure 3.8. Comparison of best designs from 50 runs for the 52-bar truss structure.....	67
Figure 3.9. Comparison of best designs from 50 runs for the 52-bar truss structure.....	67
Figure 3.10. Schematic of 25-bar space truss.	69
Figure 3.11. Comparison of best designs from 50 runs for the 25-bar truss structure.....	70
Figure 3.12. Comparison of best designs from 50 runs for the 25-bar truss structure.....	70
Figure 3.13. Schematic of 72-bar space truss.	72
Figure 3.14. Comparison of convergence rates for 72-bar spatial truss.	73
Figure 3.15. Comparison of best designs from 50 runs for the 72-bar truss structure.....	74
Figure 3.16. Comparison of number of structural analyses to reach the best.	76
Figure 4.1. Dynamic response vs ESL response for a given design (Kim & Park, 2010).....	84
Figure 4.2. GOESL process.	93
Figure 4.3. Schematic of the 18-bar truss and the applied dynamic load.	96
Figure 4.4. Convergence history of 18-bar truss of the first run.....	96
Figure 4.5. Optimum configuration for the 18-bar truss.....	101
Figure 4.6. Schematic of the 10-bar truss and the applied dynamic load.	102
Figure 4.7. Convergence history of 10-bar of the first run.	105
Figure 4.8. Schematic of the 2-story 2-bay frame and the applied dynamic load.	106
Figure 4.9. Convergence history of 2-story 2-bay frame.....	109
Figure 4.10. Schematic of the 2-story 2-bay frame and the applied blast load.....	112
Figure 4.11. Convergence history of 2-story 2-bay frame subjected to blast load.	113
Figure 5.1. Blast wave pressure (Ngo, et al., 2007).....	123
Figure 5.2. The positive phase parameters of hemispherical wave of TNT charges (modified from DoD, 2008).....	124
Figure 5.3. Schematic of 3D framed steel structure.....	136
Figure 5.4. 3D framed steel structure and charge location.	139

Figure 6.1. Schematic of 3D framed steel structure..... 159

PREVIEW

LIST OF ACRONYMS

ACO	Ant Colony Optimization
AISC	American Institute of Steel Construction
ASCE	American Society of Civil Engineers
BB-BC	Big Bang-Big Crunch
CBO	Colliding Bodies Optimization
DIF	Dynamic Increasing Factor
DoD	Department of Defense
DP	Dynamic Programming
ECBO	Enhanced Colliding Bodies Optimization
ESL	Equivalent Static Loads
FEMA	Federal Emergency Management Agency
GA	Genetic Algorithms
GOESL	Global Optimization with Equivalent Static Loads
HS	Harmony Search
HHC	Hybrid Harmony Search-Enhanced Colliding Bodies
HPSO	Harmony Particle Swarm Optimization
IHS	Improved Harmony Search
IRO	Improved Ray Optimization
ISD	Inter-Story Draft
LP	Linear Programming
OAPI	Open Application Programming Interface
NLP	Nonlinear Programming

NSA	Number of Structural Analyses
PSO	Particle Swarm Optimization
DHPSACO	Discrete Heuristic Particle Swarm Ant Colony Optimization
RO	Ray Optimization
SA	Simulated Annealing
SD	Standard Deviation
SDOF	Single Degree of Freedom
SIF	Strength Increasing Factor
TNT	Trinitrotoluene

PREVIEW

CHAPTER 1

INTRODUCTION

1.1 Introduction

Interest in the behavior of structures subjected to blast loading has increased over the last few decades as the terrorist attacks increase around the globe. Attacks on the World Trade Center in New York City in 1993 and Murrah Federal Building in Oklahoma City in 1995 showed the great damage that could happen due to a blast. In both attacks, structural failure caused more casualties and injuries than the blast wave itself (Cormie, 2009). Normally, conventional structures (many are moment resistance frames) are not designed to tolerate blast loads which are very high compared with service loads. For instance, a 10 lbs of TNT at a distance about 50 ft causes roughly peak pressure of 2.5 psi (360 psf) in very short time (less than a second) compared to natural periods of structures. In comparison, the design snow load in the Midwest ranges from 5 psf to 50 psf (Longinow and Alfawakhiri, 2003). Therefore, even a small charge explosion can cause catastrophic local or global failure of the structure. Thus, it is important to design structures (at least some critical ones) that can with stand blast loads.

1.2 Blast Phenomena and Blast Loads

When a blast occurs in the air, it generates hot gas with high pressure and high temperature. A blast wave happens because the air around the explosion expands and its molecules pile-up. The blast wave carries large amount of energy and it travels fast. When the blast wave passes parallel to an object it causes what is called incident pressure. If the blast wave is opposed by an object (such as the façade of a building) it reflects. The reflected pressure is higher than the incident pressure.

The blast pressure is a time history loading. Scaling law is the most popular method to obtain blast pressure-time profile. It is a simplified empirical method that requires only charge weight and stand-off distance. The method is explained thoughtfully in DoD (2008) and it is briefly discussed and used in this study. Computational Fluid Dynamics (CFD) and semi-empirical methods are more accurate than the scaling law to estimate blast pressure, but they are more complex and require substantially more computational time.

In the design of structure subjected to blast loads, like seismic design, some members are allowed to experience nonlinear response for more economical designs. Therefore, the blast design criteria are deformation based. In other words, members end rotations and deflections are the design criteria to design structures subjected to blast loads.

1.3 Optimization of Framed Structures Subjected to Blast Loading

It is desirable to design structures to minimize structural weight while all performance requirements are satisfied. In this study, the optimum design of three-dimensional (3D) framed steel structures subjected to blast loading is considered. Since the blast loads are much larger than the service loads, design of structure to withstand blast loads is expected to be much heavier than the design for service loads. Therefore, it is important to optimize the design for blast loads. In addition, for the design to be useful, constraints of the applicable design code must be included in the formulation. The formulation must also include direct selection of sections available in the commercial catalog. In such a formulation, the design variables are integers representing the section number in the catalog. Also, constraints of the design code are generally discontinuous because the limit values for them are based on checking various failure modes for the members.

It is seen, based on the foregoing discussion, that the gradient-based optimization methods will not be suitable for the practical design optimization formulation. Therefore gradient-free

methods must be used. Stochastic or metaheuristic methods are good gradient-free methods that need only the simulation results. These methods will be explored for the structural optimization problem subjected to blast loads. This problem of optimum design of structures subjected to blast loads has not been studied in the literature before.

1.4 Optimum Design of Framed Structures for Damage Tolerance

It is desirable to anticipate certain amount of damage to the structure under blast loads and include that in the design process. A formulation is needed for optimum design of 3D framed structures that can withstand some future damage due to a blast near the structure. The main idea is that structure should still carry the service loads when some damage happens due to blast loads. However, how to define damage to the 3D framed structure needs to be investigated.

1.5 Optimization Algorithms

Three types of structural analyses need to be carried out in the optimization process: linear static analysis of the framed structure subjected to service loads only, linear dynamic analysis of the framed structure subjected to service and blast loads, and nonlinear dynamic analysis (geometrical and material nonlinearities) of the framed structure subjected to service and blast loads. Depending on the size of the structure and the finite element model used, the computational time for each analysis can be substantial. The metaheuristic algorithms use only analysis results in their calculations, but they require a large number of iterations to reach an optimum point. In addition, dynamic analysis of the structure can require large computational effort. Therefore, the wall-clock time required to solve problem of optimum design of structures subjected to blast loads can be enormous.

Thus, it is desirable to explore ways to reduce the number of iterations of the metaheuristic algorithm so that the wall-clock time needed to solve the problem can be reduced. To achieve this

objective, combination of two metaheuristic algorithms will be explored in order to develop a more efficient hybrid algorithm.

1.6 Optimization of Structures Subjected to Dynamic Loads Using Equivalent Static Loads

Equivalent static loads (ESL) approach has been used successfully for optimizing many structural systems subjected to dynamic loads for continuous variable optimization problems using the gradient-based methods. In the ESL method, the dynamic load is transformed into multiple equivalent static load sets. Then the equivalent static loads are considered as multiple loading conditions in the linear static response optimization process (Kang et al., 2001). This process is repeated a few times until the optimum point is reached.

For the nonlinear dynamics problem, there can be either convergence difficulties in the numerical integration algorithm or nonlinear structural analysis can take long time because of the material nonlinearity, geometrical nonlinearity, and size of the structure. Considering that the metaheuristic algorithms require many simulations (depending on the number of design variables and the number of elements in the discrete set), it is quite inefficient to carry out the nonlinear dynamic analysis in the optimization process with metaheuristic algorithms. Therefore, the ESL approach will be investigated for optimization of structures with discrete design variables using metaheuristic algorithms. The ESL approach has not been investigated with metaheuristic algorithms before.

1.7 Motivation and Purpose

Several research objectives are set up for this study:

- 1- *To develop and study a formulation for optimum design of 3D framed steel structures subjected to blast loads in addition to the service loads.* To this end, design variables, cost function, and

constraints will be defined and discussed. Linear static, linear dynamic, and nonlinear dynamic analyses will be incorporated in the formulation and the optimization process. The formulation for this class of problems are discussed for the first time in this study.

- 2- *To develop a formulation for optimum design of framed structures that includes some possible future damage to the structures due to blast loads.* How to define damage to the structure will be studied and incorporated into the optimum design formulation.
- 3- *To develop a hybrid metaheuristic algorithm by combining good features of two algorithms. Evaluate this new algorithm by solving a set of test structural optimization problems.*
- 4- *To investigate the use of the equivalent static loads (ESL) approach with discrete design variables using metaheuristic algorithms.* Evaluate the approach by solving a set of test problems. The ESL approach with metaheuristic algorithms is discussed for the first time in this study.

1.8 Scope of Thesis

Chapter two is a review of the literature on calculation of blast loads, structural modeling and analysis, and design requirements for structures to resist explosions. The review provides the reader with a concise reference to the analysis and design of structures for blast resistance. In Chapter three, metaheuristic algorithms that are used in this study are discussed. The focus is mainly on the development of a hybrid algorithm. The algorithm is evaluated by solving a set of five truss test structures. In Chapter four, the ESL approach for structures subjected to dynamic loads is investigated with metaheuristic optimization algorithms and discrete design variables. Performance of the new approach is studied using four numerical examples (2 linear and 2 nonlinear dynamic response optimization problems). Chapter five presents a formulation for design optimization (design variables, cost function, and constraints) of 3D framed steel structure

subjected to blast loads. Five numerical design examples are solved, and the results are studied and compared. In Chapter six, a formulation for optimum design of 3D framed steel structures to withstand some possible future damages is presented and discussed. Chapter seven summarize the results obtained in this study and it includes some future work recommendations.

PREVIEW

CHAPTER 2

DESIGN OF STRUCTURES SUBJECTED TO BLAST LOADS: ANALYSIS AND DESIGN REVIEW

Abstract

When designing structures to withstand explosions, the main goals are to minimize the number and extent of occupant injuries and to reduce the chance of catastrophic damage to structures. Although there is uncertainty in the source, extent, and location of explosions, the assessment of blast loading and structural performance is important when designing blast-resistant structures. This chapter is a review of the literature on prediction of blast loads, structural modeling and analysis, and design criteria for structures to resist explosions. The chapter provides in one concise document the general guidelines, references, and tools that structural engineers and researchers need to analyze and design structures subjected to blast loading. References on the topics discussed in this work are provided for more detail.

2.1 Introduction

A small-charge explosion could cause catastrophic local or global failure of the structure. Analysis and design of blast-resistant structures requires good knowledge of the blast phenomena, dynamic response of structures, and design requirements. However, threats cannot be predicted accurately, and it is not possible to design a fully protected structure. Thus, an acceptable damage to the structure is expected according to a predefined level of protection (Goel and Matsagar, 2014).

The purpose of this chapter is to review the literature and provide the reader with a concise reference for analysis and design of structures for blast resistance. It provides basic considerations for blast load calculation, structural modeling and analysis, and design criteria. This study is

limited to surface bursts where the explosive charge is detonated close to ground level and the structure is regularly shaped.

This chapter is organized into eight sections. Following this introduction, Section 2.2 provides an overview of the literature. Section 2.3 discusses the blast phenomena and ways to assess blast load and its duration, and Section 2.4 provides a review of material strength under a high strain rate. Section 2.5 discusses stress increase and reduction factors, and Section 2.6 discusses modeling and analysis of structural components and systems subjected to blast loads. In Section 2.7, design criteria for structural components and system are discussed, and Section 2.8 provides a definition of progressive collapse that designers should be aware of. References on all topics are provided for more detail.

2.2 Literature Review

The subject areas of blast load prediction and blast-resistant design are quite broad. In this review, many references have been used to collect information on these subject areas and provide the reader with a concise document. This section provides a brief overview of the key references used in this study along with some information discussed in each reference. The U.S. Department of Defense (DoD) publication (U.S. Department of Defense, 2008) provides a manual for evaluating blast loads and design criteria for members and structural systems. It is considered one of the most important references for blast-resistant design. The American Society of Civil Engineers (ASCE) prepared a report (ASCE, 2010) to provide guidance for blast resistance of petrochemical facilities. The ASCE also wrote a standard (ASCE, 2011) that provides planning, design, construction, and assessment requirements for existing and new structures subjected to blast loading. Gilsanz et al. (2013) wrote a guide published by the American Institute of Steel Construction that focuses on blast resistance and progressive collapse mitigation of steel structures.

It provides a few detailed design examples. Pape et al. (2010a, 2010b, and 2010c) published a three-part paper on the blast phenomena and its effect on structures. The work provides a practical overview of types of explosions, prediction of explosion effects, and methods for analysis under blast conditions. Goel and Matsagar (2014) discussed different strategies for blast mitigation and the mechanics of sacrificial blast walls using different materials. Books by Smith and Hetherington (1994), Bangash and Bangash (2006), Cormie et al. (2009), and Dusenberry (2010) provide detailed information on the analysis and design of buildings subjected to blast conditions. This chapter summarizes the most important analysis and design information provided in these references and others with a MATLAB code to predict blast loads based on the method described by the DoD (2008).

2.3 Prediction of Blast Loading

This section provides the necessary background and references to calculate external blast loading. Although there is uncertainty in predicting the size, type, and location of the explosive, calculation of blast loads is essential in the design of blast-resistant structures.

2.3.1 Blast Phenomena

The explosion generates hot gas that can be at a pressure of 1450-4351 ksi and a temperature of 3000-4000 °C (Smith and Hetherington, 1994). If a blast happens in the air, the high-temperature gas that is produced by an explosive charge expands spherically to take up the available space. In other words, the violent expansion forces the surrounding air out of its occupied space. Simultaneously, the air around the explosion expands and its molecules pile up. What is known as a *blast wave* occurs next, and it carries a large amount of energy. As the wave front moves away from the source of the explosion, its pressure decreases at an exponential rate until it falls to the normal atmospheric pressure; this is called the positive phase. After that, it decreases

to less than the atmosphere pressure (negative phase) and finally back to the ambient value (see Figure 2.1). Thus, the blast pressure is a time history loading. In Figure 2.1, P_{so} is the peak overpressure or the incident pressure, P_o is the ambient pressure, P_{so}^- is the minimum negative pressure, P_r is the reflected pressure, P_r^- is the minimum negative reflected pressure, t_a is the arrival time, t_o is the positive phase duration, t_o^- is the negative phase duration, i_s is the positive reflected impulse, and i_s^- is the negative incident impulse. When the blast wave travels parallel to a surface and is unimpeded by any object, free-field (side-on or incident) pressure is applied to the surface (see Figure 2.1 (a)). When a surface is struck by a blast wave perpendicularly or at an angle, reflected pressure is applied to the surface.

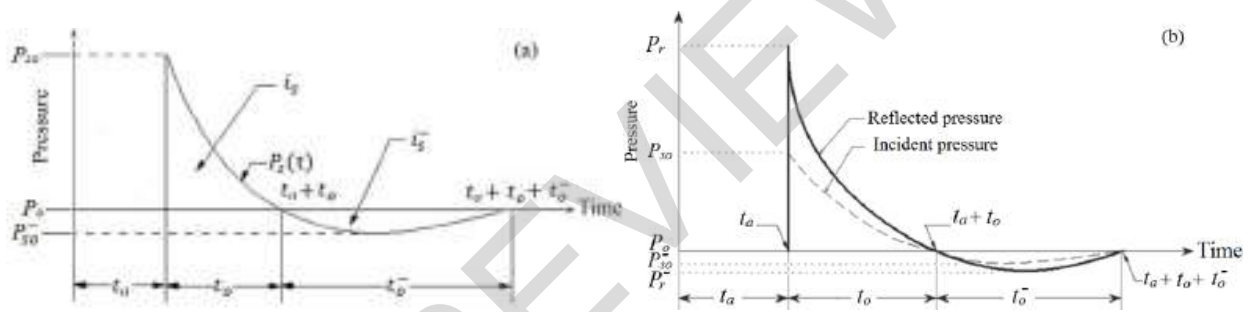


Figure 2.1. Idealized pressure-time profile for blast wave: (a) free field pressure (b) reflected pressure (modified from DoD, 2008).

Friedlander's exponential equation is usually used to describe the pressure-time history of a blast wave (Cormie et al., 2009):

$$P_s(t) = P_{so} \left(1 - \frac{t}{t_o} \right) e^{-bt/t_o} \quad (2.1)$$

where b is a decay coefficient of the waveform (calculated through a nonlinear fitting of an experimental pressure time curve over its positive phase).

There are three techniques to calculate blast loads (Cormie et al., 2009):

1- *First principle methods*: These are the most accurate methods that involve solving the partial differential equations based on computational fluid dynamics (CFD). The CFD models

determine a numerical solution to fluid (air) flow equations. These equations are based on the principles of conservation of mass, momentum, and energy. The reader is referred to the work of Cormie et al. (2009) and Zienkiewicz et al. (2006) for more details on this topic. There are many computer codes available for modeling the detonation of explosives, such as LS-DYNA (2007), ABAQUS (2014), and Air3d (Rose, 2001). The blast loads calculated with CFD are used to compute the structural response. However, when the structure is expected to move significantly due to the blast event, the blast wave and the structural response could be coupled to obtain more accurate results (Pape et al., 2010b).

- 2- *Semi-empirical or phenomenological methods*: These are simplified methods that represent the essential physical phenomena of the explosion.
- 3- *Empirical methods*: These are based on an analysis of the experimental data (Goel and Matsagar, 2014). Scaling Law is the most common empirical method used in the analysis and design of blast-resistant structures. Blast parameters such as incident and reflected pressures are functions of the scaled distance (Z). Report UFC 3-340-02 developed by the DoD (2008) provides guidelines to predict blast loads using the empirical method. ConWep (Hyde, 1992) and ATBlast are examples of computer programs that are widely used to determine blast wave parameters. They are an implementation of the method described by the DoD (2008).

The selection of an analysis method depends on the project requirements and type of components to be designed (Gilsanz et al., 2013).

Blast load decreases rapidly with distance. Therefore, based on the distance from the source of the blast and the angle of incident, blast loads and their durations can change considerably over the surface of the structure. The common approach is to divide the surface into a grid and then calculate blast loads and their durations at the center of each section of the grid.

2.3.2 Scaling Law

The distance of the structure from the detonation point is an important parameter in calculating the blast loads. The Hopkinson-Cranz scaling approach (cube-root scaling) is the most widely used approach for blast wave analysis for spherical explosions. The scaling distance is defined as follows:

$$Z = \frac{R}{\sqrt[3]{W}} \quad (2.2)$$

where Z is the scaled distance, R is the distance from the detonation source to the point of interest expressed in feet (ft.), and W is the charge mass expressed in pounds (lbs) of TNT.

There are many types of explosives. TNT was chosen to be the blast parameter, so an equivalent TNT weight needs to be computed in order to use Eq. (2.2). Equation (2.3) below is used to find the equivalent weight of TNT, and Table 2.1 shows the conversion factors for some explosives (DoD, 2008).

$$W_e = W_{exp} \frac{H_{exp}^d}{H_{TNT}^d} \quad (2.3)$$

where W_e is the equivalent TNT weight, W_{exp} is the weight of the explosive, H_{exp}^d is the heat of detonation of the explosive, and H_{TNT}^d is the heat of detonation of the TNT.

Table 2.1. Heat of detonation for some explosives (DoD, 2008).

Explosive name	Heat of detonation, ft-lb/lb
TNT	1.97 E+06
Composition B	2.15 E+06
Composition C4	2.22 E+06
RDX	2.27 E+06
HMX	2.27 E+06

Identifying explosive size is an important part of the threat assessment process. Table 2.2 shows the estimated ranges of explosives. Bangash and Bangash (2006) categorize explosives as small, medium, large, and very large (Table 2.3).

Table 2.2. Estimated quantities of explosive (FEMA 426, 2003).

type	Charge weight
Luggage	10-100 lb TNT
Automobile	100-450 lb TNT
Van	450-4000 lb TNT
Truck	4000-100000 lb TNT

Table 2.3. Size of explosive (M. Bangash and T. Bangash, 2006).

type	Charge weight
Small	Up to 11 lb TNT
Medium	Up to 44 lb TNT
Large	Up to 220 lb TNT
Very large	Up to 5512 lb TNT

2.3.3 Explosion and Blast-Loading Types

There are three types of explosions, as shown in Figure 2.2 (Karlos et al., 2013):

- 1- *Free-air bursts*: In this case, the charge is detonated in the air away from any reflecting surface. The blast waves can be characterized by a spherical wave that moves outward from the source and impinges directly onto the structure.
- 2- *Air bursts*: The explosive charge is detonated in the air. The blast waves propagate spherically outward from and impinge on the structure after having interacted first with the ground. What is called Mach reflection might occur because of the interaction of the blast wave and the reflected wave.
- 3- *Surface bursts*: The explosive charge is detonated near the ground surface. The blast waves immediately interact locally with the ground and then propagate hemi-spherically outwards, impinging on the structure.

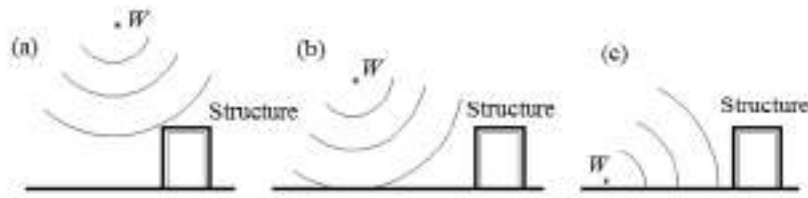


Figure 2.2. Types of explosions: (a) Free-air bursts. (b) Air bursts. (c) Surface bursts (modified from Karlos et al., 2013).

2.3.4 Blast Wave Reflection

The blast waves will reflect when they impact an object made of a medium denser than that carrying the wave. In this case, the pressure acting on the structure is not the same as the incident peak pressure (P_{so}). In fact, the reflected pressure could be several times greater than the incident pressure, as shown in Figure 2.1 (Cormie et al., 2009).

In the discussion above, the angle of the incident (α) is taken as zero. When $\alpha = 90^\circ$, the blast wave travels parallel to the surface. That is, there is no reflection, and the structure is loaded with side-on pressure that is equal to the incident overpressure. If α is between 0° and 90° , either regular or Mach reflection happens. The effect of the angle of the incident on the reflection coefficient ($C_{r\alpha} = \frac{P_r}{P_{so}}$) is shown in Figure 2.3 (Karlos et al., 2013).

The influence of the angle of incident can be ignored for the large pressure, and the structure can be studied under a normal reflected pressure, which is a conservative approach. In general, one can use Figure 2.3 to determine the reflection coefficient.

The Mach reflection is a complex process. When the reflected wave catches up with the incident wave, the so-called Mach stem occurs. This is the reason for the jump in the angle of the incident-reflected pressure curves shown in Figure 2.3.

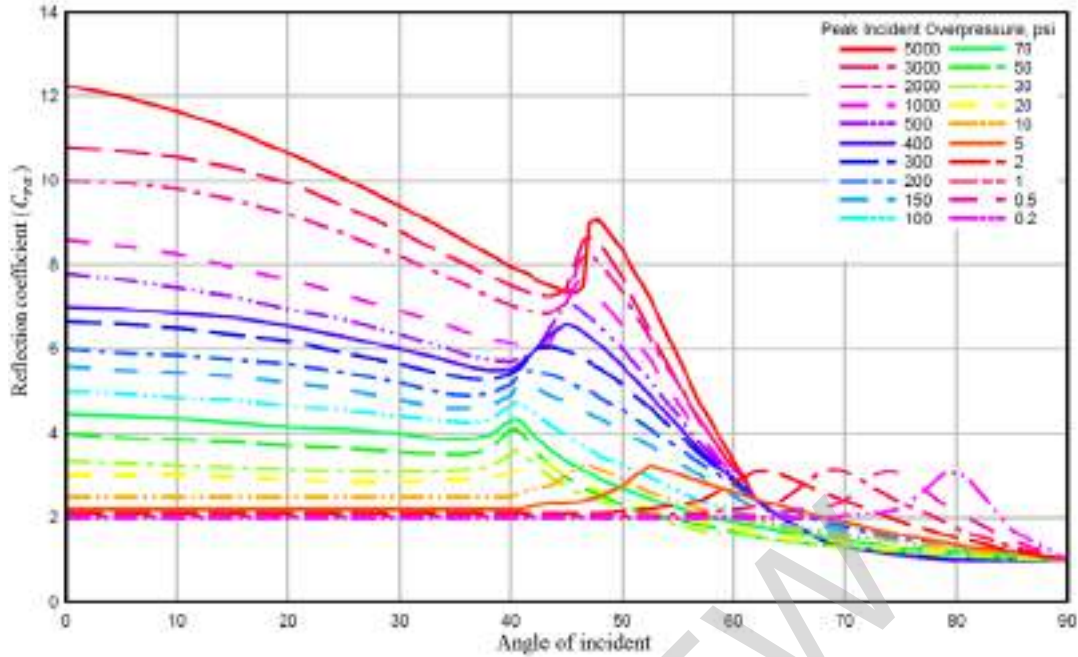


Figure 2.3. Influence of angle of incident on the reflection pressure (modified from DoD, 2008).

Conventionally, facades are assumed to be perfectly rigid so that they perfectly reflect the blast wave front. In reality, however, facades displace when the blast wave impinges on them. This displacement reduces the effectiveness of the reflected pressure.

2.3.5 Surface Burst and Loading

When the explosive charge is placed close to the ground, a modification must be made to the charge weight. The incident wave is reflected immediately from the ground and interacts with the blast wave. This is called hemispherical burst. Practically, due to the creation of a crater, some energy absorption takes place from the ground. Figure 2.4 and 2.5 show the blast wave parameters of a hemispherical wave of TNT charge for the positive and negative phases, respectively. The wave parameters are presented on the y -axis while the x -axis represents the scaled distance (Z).

In, Figure 2.4 W is the weight of the charge, P_{so} is the incident peak overpressure, P_r is the reflected pressure, i_r is the positive reflected impulse, i_s is the positive incident impulse, t_A is the arrival time, t_o is the positive duration, U is the wave speed, and L_w is the wave length. They are

presented on the y -axis, while the x -axis represents the scaled distance Z . In Figure 2.5, the superscript “-” refers to the negative phase.

After calculating the scaled distance for a specified distance and charge weight, Figure 2.4 and 2.5 can be used to determine the positive and negative parameters to plot the equivalent pressure time history for the front, roof, and side and rear walls (Figure 2.6). Numerical examples showing all the steps to find the equivalent load time history are available in the work of the DoD (2008), Gilsanz et. al. (2013), and Karlos et al. (2013). A MATLAB code that follows the methods presented by DoD (2008) is provided*. The code can be used to plot the triangular shape of the pressure time history (similar to those shown in Figure 2.6). Note that the scaled distance must be within the range of Figure 2.4 and 2.5. For close-in explosions, this simplified approach is not allowed. CFD or test data should be used to find the blast loading, and explicit nonlinear dynamic analysis should be performed to consider breach, diagonal tension, direct shear, and spall failure mode.

* https://www.mathworks.com/matlabcentral/fileexchange/70105-matlab_code_blast_load_dod_2008

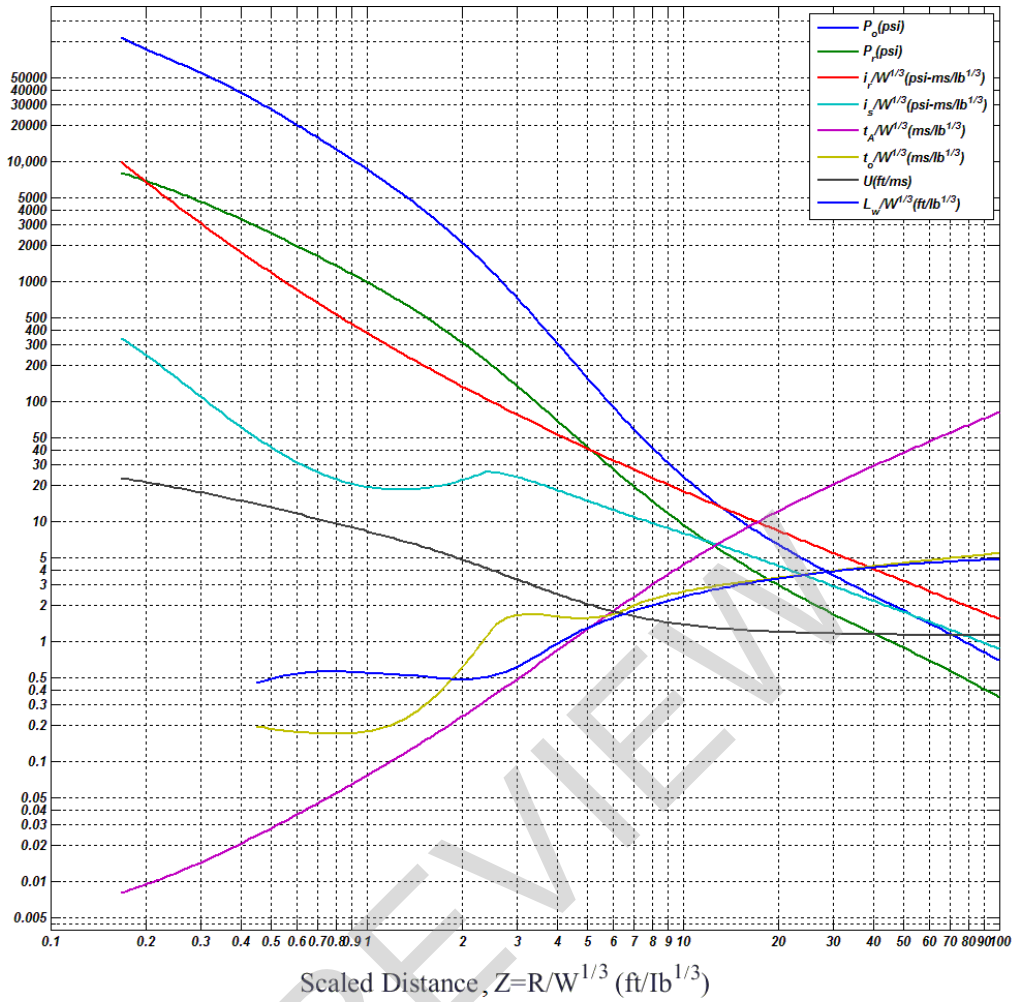


Figure 2.4. The positive phase parameters of hemispherical wave of TNT charges (DoD, 2008).

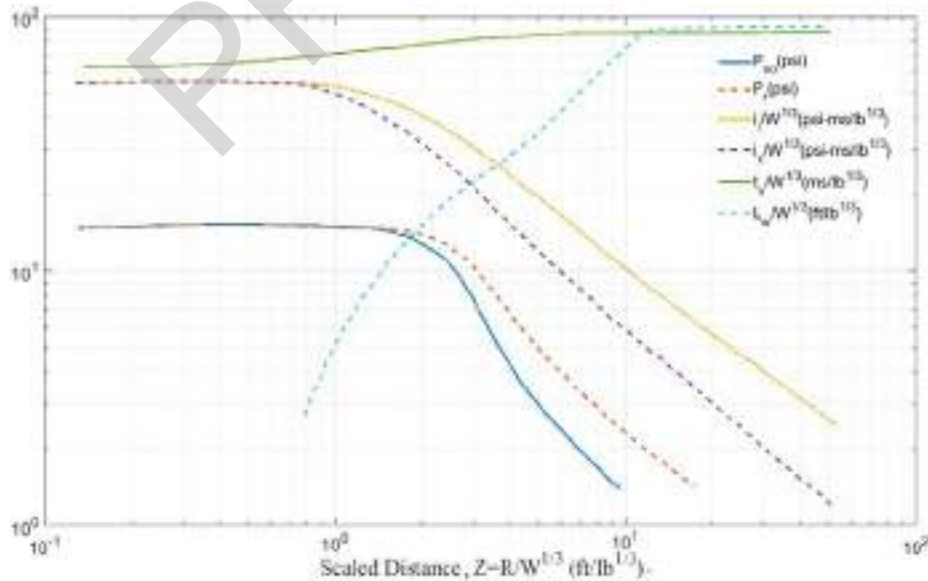


Figure 2.5. The negative phase parameters of hemispherical wave of TNT charges (modified from DoD, 2008).

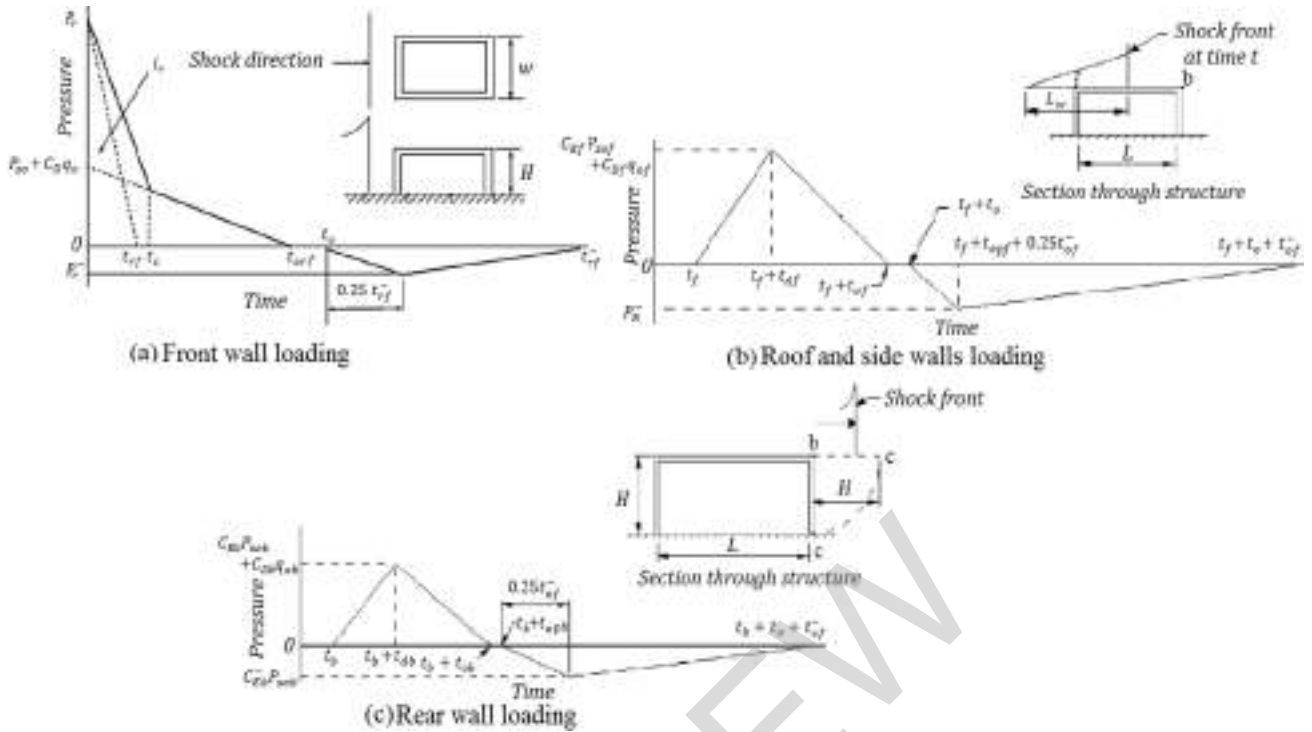


Figure 2.6. Triangular assumption of pressure time history on the (a) Front walls loading (b) Roof and sides walls loading (c) Rear walls loading (modified from DoD, 2008).

Figure 2.6 shows the simplification of the pressure-time history profile of the blast wave (Figure 2.1). In Figure 2.6, w is the width of the front wall and the back wall, H is the height of all walls, L is the length of the side wall, P_r is the reflected pressure, P_{s0} is the incident peak, C_D is the drag coefficient (C_D is 1 for the front wall), q_0 is the incident dynamic pressure, i_r is the total reflected pressure impulse, t_{rf} is the duration of the reflected pressure, t_c is the clearing time, t_{orf} is the actual positive phase duration, and t_o is the positive phase duration. In the roof and side wall loading figure, L_w is the wave length, C_{Ef} is the equivalent load factor, P_{sof} is the incident pressure, C_{Df} is the drag coefficient at point f, q_{of} is the dynamic pressure, t_f is the time when the blast wave reaches the point f, t_{df} is the time when the peak equivalent uniform pressure is reached, t_{of} is the actual positive phase duration, and t_{opf} is the positive phase duration. In the rear wall loading figure, the notations are similar to the roof and side wall loading figure, except that point b is used instead of point f. The superscript “-” refers to the negative phase.

2.3.6 Negative Phases

Compared with the positive phase, the negative wave has a longer duration and a lower pressure magnitude, as shown in Figure 2.1. It reduces the effect of the peak response, and it is usually ignored in design because the main structural damage results from the positive phase loads (Karlos et al., 2013). However, its effect should be examined for members that have a shorter fundamental period in comparison with negative load duration (Gilsanz et al., 2013).

2.3.7 Internal Pressure

In the previous sections, blast pressure has been discussed with the assumption that there are no openings in the walls. Structures, however, have windows and doors that may leak pressure into the building, causing a reduction in the effective new load on the external walls. Internal pressure is important in evaluating the effects on personnel and the internal damage. The internal pressure effect is usually ignored when the openings are small (Gilsanz et al., 2013). The DoD (2008) provides a procedure to evaluate internal pressure.

2.4 Material Design Strength

Steel and reinforced concrete are the most commonly used materials in the construction of blast-resistant structures, but masonry and timber are permitted. For a close and high-impulse blast event, concrete structures are generally used to provide protection against fragments and to limit deformation (DoD, 2008; ASCE, 2011).

The ductility of members (or general structures) is an essential factor in blast design: the greater the ductility, the greater the members' resistance to failure. Low-carbon steel and properly reinforced concrete are suitable to blast resistant design because they can deform beyond the elastic limit without rupturing (ASCE, 2010).

The mechanical properties of material under high strain rate loadings such as blast loads are different from low rate and static loads. Generally, materials become stiffer under high rate loadings, which leads to an improvement in their mechanical properties. Also, in blast design, it is allowable to use the expected actual strength of the material instead of the minimum specified values.

2.4.1 Material Properties for Steel

The effects of high strain rate on some of the mechanical properties of steel are summarized as follows:

- 1- The modulus of elasticity (E_s) remains the same.
- 2- The yield strength (f_y) and ultimate tensile strength (f_u) increase to the dynamic yield strength (f_{dy}) and the dynamic ultimate strength (f_{du}), respectively. Figure 2.7 shows the effect of increasing strain rate on steel.

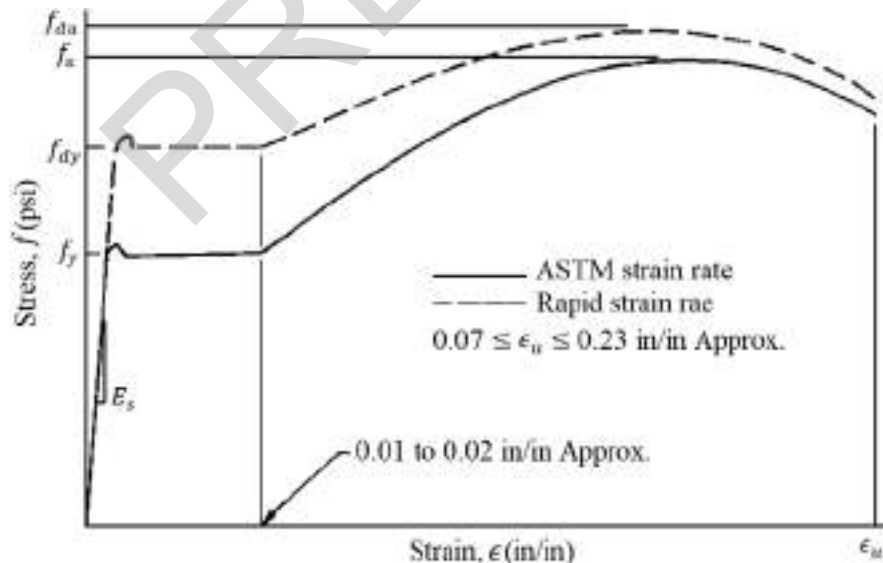


Figure 2.7. The effect of high strain rate on mechanical properties of steel (modified from DoD, 2008).

Dynamic increase factors (*DIF*) are used to modify the static strength due to high rate dynamic loads. Table 2.4 presents the values of *DIF* for different types of steel and different strain rates.

Table 2.4. Dynamic increasing factor (*DIF*) for yield stress and ultimate stress for structural steel (DoD, 2008).

Steel type	Yield DIF				Ultimate stress DIF
	Bendign		Tension or compression		
	Low Pressure ($\dot{\epsilon}=0.1$ in/in/sec)	High Pressure ($\dot{\epsilon}=0.3$)	Low Pressure ($\dot{\epsilon}=0.02$)	High Pressure ($\dot{\epsilon}=0.05$)	
A36	1.29	1.36	1.19	1.24	1.10
A588	1.19	1.24	1.12	1.26	1.05
A514	1.09	1.12	1.05	1.07	1.00

The average yield stress of steel of grades 50 ksi or less is about 10% higher than the stress value specified by ASTM. Thus, for blast-resistant design, the yield stress is 1.1 times the minimum yield stress. This factor is called the strength increase factor (*SIF*) or the average strength factor (*ASF*). The *SIF* should not be used with high-strength steels (Gilsanz et al., 2013).

2.4.2 Material Properties for Reinforce Concrete

Similar to steel, reinforced concrete shows improvements in its mechanical properties when it is subjected to blast loadings. The effect of high strain rates on reinforced steel and concrete are shown in Figure 2.7 and 2.8, respectively. Table 2.5 provides the DIF values of reinforced steel and concrete. The *SIF* of reinforced steel is discussed in Section 2.4.1, and the *SIF* for compressive strength of concrete is 1.1 (ASCE, 2011).

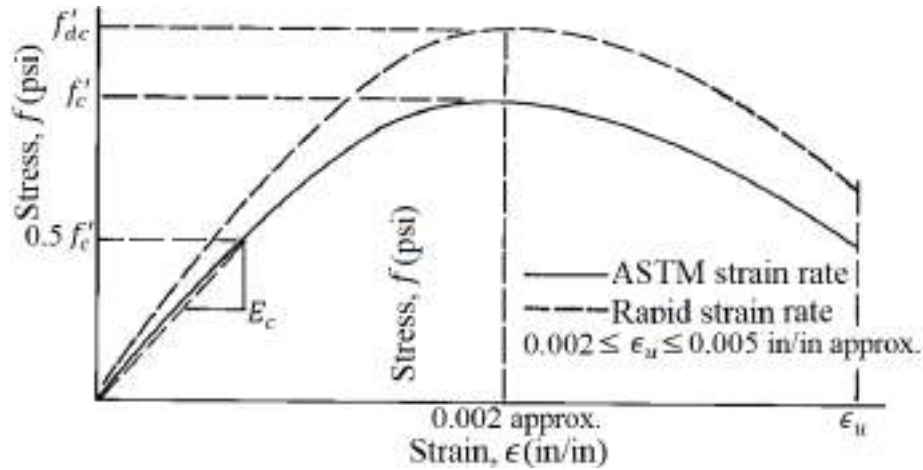


Figure 2.8. The effect of high strain rate on mechanical properties of concrete (modified from DoD, 2008).

Table 2.5. Dynamic increase factor (*DIF*) for reinforced concrete design (DoD, 2008).

Type of stress	Reinforced bars		Concrete
	Yield stress	Ultimate stress	
Bending	1.17	1.05	1.19
Diagonal tension	1.00	-	1.00
Compression	1.10	-	1.12

2.4.3 Plastic Hinge

In design for blast loading, some members are allowed to have plastic behavior to achieve an economical design. Therefore, it is important to understand the local performance of members and the global performance of the structure when one or more plastic hinges start to form. Also, the locations and modeling of the plastic hinges are important. To allow a plastic hinge to form in a component, lateral supports must be provided to prevent premature buckling. It is good practice to design columns to remain elastic to prevent extended structural failure (Gilsanz et al., 2013). This is the “strong column, weak beam” approach. That is, beams are forced to fail before columns.

A plastic hinge is formed at the point of maximum stress. It starts when the outer fiber reaches the material yield limit. Then, the interior of the section starts to yield gradually as the load increases and the stress-strain relationship becomes nonlinear. At other locations, the resistance

continues to increase as the load increases. That is, some points respond plastically while others respond elastically, and elastic-plastic conditions occur (ASCE, 2010).

Modeling the nonlinear behavior of sections depends on the material to be used and the internal force in the section. For example, an ideal elastic-plastic behavior is accepted in the design of a single-degree-of-freedom (SDOF) system. Figure 2.9 shows the idealized resistance-deflection curve, where R_m is the ultimate dynamic resistance, X_E is the deflection at the limit of elastic range, K_e is the elastic stiffness, and X_m is the maximum allowed deflection corresponding to the ductility ratio (u) or rotation (θ) given in Section 2.6.

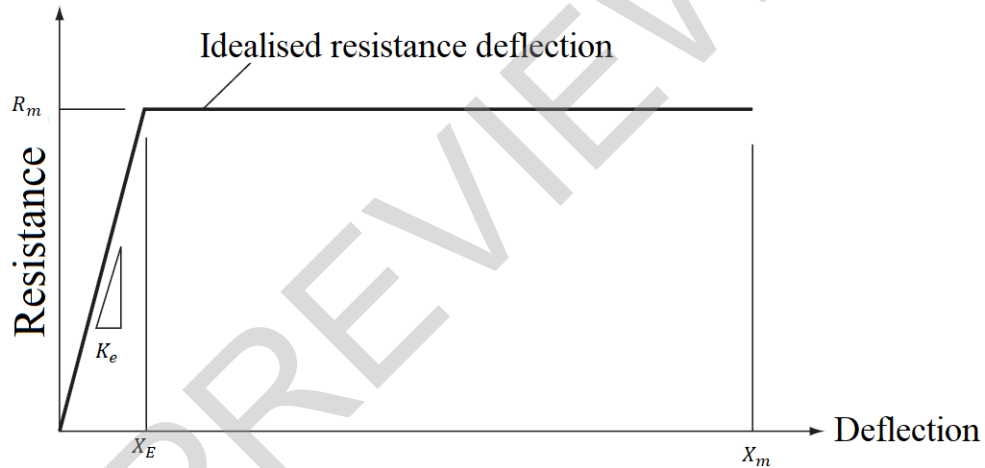


Figure 2.9. Idealized resistance-deflection curve (Cormie et al., 2009).

In more complex scenarios such as a steel member subjected to tension and compression, a plastic hinge can be modeled using FEMA 356 (ASCE and FEMA, 2000), as shown in Figure 2.10 and Table 2.6, where a , b , and c are hinge parameters that are functions of the elongation, P_n is the tensile strength, F_{cr} is the critical buckling load, Δ_T refers to axial deformation at tensile yield load, and Δ_C refers to axial deformation at buckling yield load.

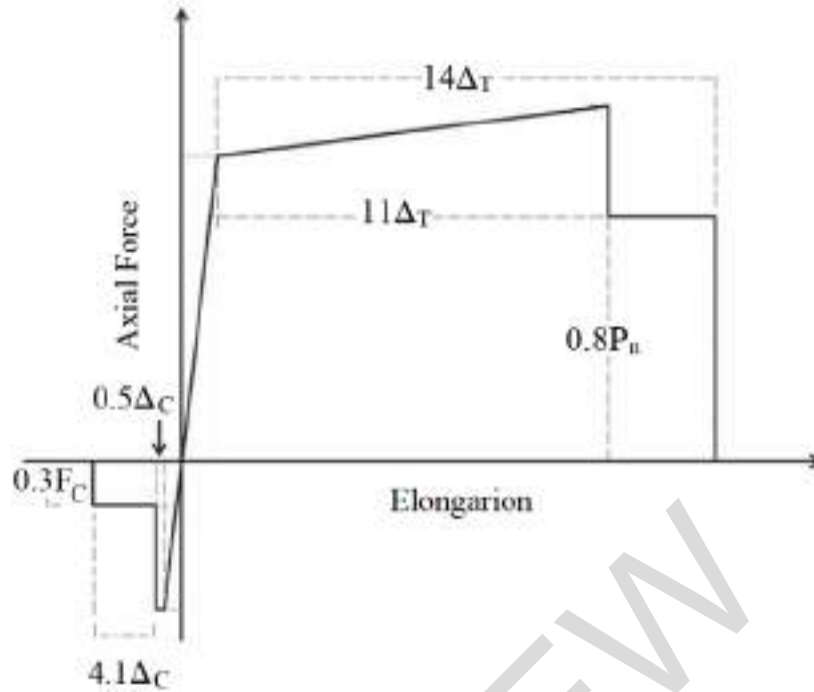


Figure 2.10. Tension-compression hinge properties (FEMA, 2000 and Gilsanz et al., 2013).

Table 2.6. Tension-Compression hinge parameters (FEMA, 2000 and Gilsanz et al., 2013).

loading	a	b	c
Tension	$11\Delta_T$	$14\Delta_T$	$0.8P_n$
Compression	$0.5\Delta_C$	$4.1\Delta_C$	$0.3F_C$

When there are axial force and bending moments in one or two directions, the plastic hinge may be represented using a P-M-M yield surface (El-Tawil and Deierlein, 2001). Here, P is the axial force, and M-M are the minor and major bending moments.

The yield surface defines the strength of the material under biaxial stress. Any elastic-plastic material has a yield surface. When the stress point is on the yield surface, the material has yielded, and its behavior is elastic-plastic. But when the stress point is inside the yield surface, the material is elastic. Stress points outside the yield surface are not allowed. Software such SAP2000 (CSI, 2017) implements what is called Parametric P-M2-M3 based on the P-M-M yield surface method (CSI, 2016).

2.5 Strength Reduction Factors and Load Combination

Because of the nature of the blast load and to achieve economical design, plastic deformations are allowed in the design of structures subjected to blast loads. Also, it is permissible to use the nominal strength without a strength reduction factor (i.e., $\phi = 1$) for all modes of failure (ASCE, 2011). Blast loads are not combined with loads that are not expected to be present when the blast happens. That is, wind, earthquake, part or all the live loads are not combined with blast loads; the basic load combination for all construction materials is as follows (ASCE, 2010):

$$1.0 DL + 1.0 LL + 1.0 BL \quad (2.4)$$

where DL is the dead load, LL is live load, and BL is blast load. In the absence of other governing criteria, Gilsanz et al. (2013) allow the following load combination:

$$1.0 DL + 0.25 LL + 1.0 BL \quad (2.5)$$

2.6 Blast Load and Structure Interaction (Structural Response)

For an isolated building, as the blast wave propagates, its front engulfs the structure. Therefore, all faces of the structure are subjected to positive and negative pressure at different times and for different durations. The structure resists the kinetic energy of moving components by converting it to strain energy in the resisting elements (Dusenberry, 2010). Due to high strain rates, nonlinear inelastic material behavior, time-dependent deformation, and uncertainties of blast load and location, the structural dynamic response is complex (Ngo et al., 2007). Depending on the predicted structural failure mechanism, designers can select the best analytical method to compute the structural response. Pressure-impulse (P-I) charts, single element response analysis, and detailed finite element analyses are the most common approaches to computing structural

response (ASCE, 2011). Designers must select an appropriate analytical approach based on expected failure mechanisms.

2.6.1 Pressure-Impulse Charts

Pressure-impulse or iso-damage curves are based on analytical or experimental data where the peak pressure and impulse represent the explosive loading on the P-I curve to check the performance condition of a target member. This simple method can be used to design secondary elements but not primary elements, and it is limited to flexural modes in response to blast loads (ASCE, 2011). Figure 2.11 shows a typical P-I diagram for an elastic SDOF component, where F is the impulse force, K is the member stiffness, M is the total mass of the member, I is the impulse ($I = \text{peak blast load} \times \text{duration of idealized triangular blast load} / 2$), u is the displacement, and u_{max} is the maximum dynamic response.

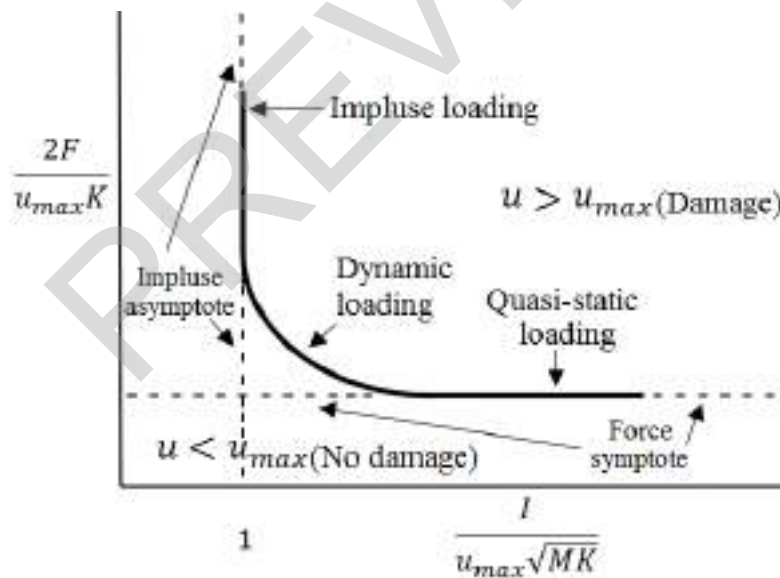


Figure 2.11. Pressure-impulse diagram for elastic SDOF component (Smith & Hetherington, 1994).

Once the maximum response is specified (damage criterion), Figure 2.11 can be used to find the impulse and the load that cause failure or to check whether the section to be designed is damaged. That is, when the combinations of impulse and pressure fall to the right and above, the

curve will result in failure; when the combinations fall to the left and below, the curve will not induce failure. Note that axes $\frac{2F}{u_{max}K}$ and $\frac{I}{u_{max}\sqrt{MK}}$ represent pressure and impulse, respectively, and they have no physical units. Smith and Hetherington (1994) discussed this approach with numerical examples.

2.6.2 The Single-Element Analysis Method

This method involves analyzing and designing individual members subjected to blast loading. This is either an SDOF or multi-degree-of-freedom (MDOF) system with elastic or inelastic dynamic analysis.

The SDOF approach is the most common, and its accuracy depends on selecting a model that adequately represents the failure mechanism. In this approach, the member's mass is concentrated at one point and is allowed to move along a single axis by assuming one response mode. The linear equation of motion for SDOF is:

$$M\ddot{u}(t) + C\dot{u} + Ku(t) = f(t) \quad (2.6)$$

where M is the total mass of the member, C is viscous damping, K is the member stiffness, u is displacement, \dot{u} is velocity, and \ddot{u} is acceleration at time t . Equation (2.6) can be solved by numerical integration using structural analysis software programs such as ABAQUS, ANSYS (2013), LS-DYNA, and SAP2000. This model can be simplified further by considering an elastic undamped SDOF system subjected to a triangular pulse load (just the equivalent positive phase). Thus, Eq. (2.6) becomes (Cormie et al., 2009):

$$M\ddot{u}(t) + Ku(t) = F\left(1 - \frac{t}{t_d}\right) \quad (2.7)$$

where F is peak force and t_d is positive phase duration. To solve Eqs. (2.6) and (2.7), the time increment should not be greater than 1/20 of the natural period of the member or 1/20 of the pulse duration t_d to provide numerical stability (Gilsanz et al., 2013). The reader is referred to UFC 3-340-02 (DoD, 2008) and the works of ASCE (2010) and Gilsanz et al. (2013) for more details.

In Eq. (2.6), the damping effects are commonly ignored because the blast load duration is short and energy dissipates through inelastic deformation (ASCE, 2010). However, it is allowable to include the damping effect when the response is nearly elastic (ASCE, 2011).

The MDOF approach, described in the next section, is more accurate than the SDOF approach because all potential modes of failure can be represented, especially when nonlinear finite element analysis is carried out with geometric nonlinearity.

2.6.3 Multi Degree of Freedom Finite Element System

The single-element modeling discussed above does not represent the actual boundary conditions, nor does it consider the interaction between elements and the phasing of their response or the dissipation of the energy of the whole structure (ASCE, 2010). On the other hand, MDOF modeling of structural systems does not ignore these important parameters. Moreover, the distribution of the mass and stiffness can be modeled throughout the structure instead of for only one member. In this approach, the linear or nonlinear time-history analysis methods can be used to determine the entire structural response. The complexity of the model depends on the type of the element used in finite element analysis, where the spring element is the simplest and the solid element is the most complex.

Discrete System: In this type of structural modeling, a beam element can be used. Depending on the symmetry of the structure and the loading, and the model can be two- or three-

dimensional. The relative flexibility and strength of the connected elements are considered. Moreover, this structural system analysis considers the phasing of the responses between structural elements (ASCE, 2011). Structural analysis outputs that include nodal and elements displacements and plastic hinge(s) rotations (when material nonlinearity is considered) can be used directly to check the design criteria.

Implicit or Explicit Linear or Nonlinear Finite Element Analysis: This approach is necessary for complex structures and to obtain more accurate results. Linear or nonlinear plate/shell elements and solid elements can be used. Implicit, explicit, or mixed-hybrid modeling can be carried out (Bangash and Bangash, 2006). The implicit method involves a numerical solver to invert the stiffness matrix to directly find the displacement vector. Thus, the implicit scheme is not a function of time. This method is unconditionally stable, but it is computationally expensive when the structure is large. Implicit methods are used in software such as ABAQUS and ANSYS. An explicit scheme is a function of time since it involves solving for velocity and acceleration as well as the inverse of the mass matrix (diagonal matrix), but the inverse of the stiffness matrix is not needed. This approach is conditionally stable. That is, small time steps should be used to obtain accurate results. The explicit method is a good choice for large models and blast load problems because the propagation of the blast load through the structure requires small time steps (LS-DYNA, 2007). The explicit method is used in software such as LS-DYNA and ABAQUS.

For both approaches, the interaction between the primary structural system and the nonstructural components can be considered to avoid any possible local failure. ASCE (2011) recommends not directly connecting vertical load-carrying elements to exterior envelope components unless they are designed to have greater strength than the exterior envelope

components they are to be connected to. Also, one-way walls without backing elements can be designed to transfer loads directly into floor diaphragms.

2.6.4 Equivalent Static Load Method

In this method, the blast load is transferred to its equivalent static load, and then the structural static analysis is carried out. This method does not represent the actual response because dynamic parameters such as strain rate, mass, plastic deformation, and time-varying load are ignored. However, when the blast source is far from the structure, the blast loading can be represented as an “equivalent wind” (ASCE, 2010).

2.7 Criteria for Responses (Response Limits)

In static design philosophy (the working stress, ultimate load, and limit state methods), the level of stress in components and deflection are typically the criteria to define failure. In blast design (similar to seismic design), it is expected that some of the components will experience a substantial nonlinear response because designing them to remain elastic is usually uneconomical. However, when a structure is required to be reused following a blast, it must be designed to remain elastic (ASCE, 2010). That is, in designing blast-resistant structures, the maximum dynamic deflection and rotation are the criteria to prevent component failure. The performance of the entire structure is defined by life safety, functionality, and reusability (Dusenberry, 2010). Moreover, designers must check that the failure of key members will not cause any progressive collapse by providing sufficient redundancy (alternate load paths). The level of protection (LOP) (see Table 2.7) for the structure or component, the type of component, and the material to be used define the design criteria (ASCE, 2010). For example, the response limit of individual elements is less than the allowable response of individual frame elements because frames have higher redundancy.

Also, for structural components (such as beams and columns), the response limits are less than nonstructural components (such as purlins).

Table 2.7. Damage and response level (ASCE, 2010).

Damage level	Description	Response level	Description
Low	Localized component damage. The structure can be utilized but it needs repairing. Total cost of repairs is moderate	Low	Component has none to slight visible permanent damage
Medium	Widespread component damage. Building should not be occupied until repaired. Total cost of repairs is significant	Medium	Component has some permanent deflection. It is generally repairable, if necessary, although replacement may be more economical and aesthetic.
High	Component has some permanent deflection. It is generally repairable, if necessary, although replacement may be more economical and aesthetic.	High	Component has not failed, but it has significant permanent deflections causing it to be unrepairable.

There are several sources for response limits, including UFC 3-340-02 (DoD, 2008), *Design of Blast-Resistant Buildings in Petrochemical Facilities* (ASCE, 2010), FEMA 356 (ASCE and FEMA, 2000), and the New York City Building Code (NYCBC, 2008). Although all of these sources define the criteria based on deformation, the limiting values are different, so the designer may need to review these limits. This review, however, is limited a portion of what is provided in *Design of Blast-Resistant Buildings in Petrochemical Facilities* (ASCE, 2010). Before defining the response limit values, three important terminologies are defined:

1- *Ductility ratio* (μ): This is the ratio between the total displacement, X_m , and the elastic displacement, X_E , as follows:

$$\mu = X_m / X_E \quad (2.8)$$

where displacement is the elongation of components subjected to axial load or the deflection of components subjected to bending, as shown in Figure 2.12 (ASCE, 2011). Ductility is a measure of how much a component can carry beyond the elastic range before it drops the load.

2- *Rotation (θ)*: This is the tangent angle at the support caused by the maximum deflection. Figure 2.12 and 2.13 show the rotation of a single element and a frame, respectively. Note that plastic hinge can happen not just at the mid-span of a member but also at other locations. This criterion indicates the degree of stability in a component.

3- *Side-sway deflection or lateral drift (δ)*: This is the movement of a vertical member relative to its bottom (Figure 2.13). Side-sway limits allow framed structures to minimize the P-delta effects on columns and the chance of progressive collapse (ASCE, 2010). Side-sway deflection limit can be defined follows:

$$\delta \leq \text{response limit} \tag{2.9}$$

where the *response limit* is story height H divided by some factor.



Figure 2.12. Member (beam, slab, or panel) support rotations (DoD, 2008).

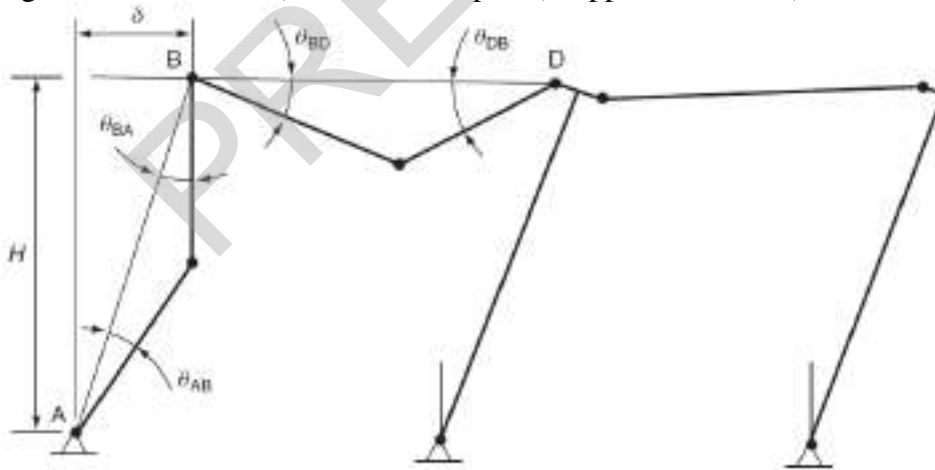


Figure 2.13. Frame support rotations and side-sway deflection (DoD, 2008).

Similar to the modeling and analysis methods discussed in Sections 2.6.2 and 2.6.3, there are two different types of criteria: for elements that are modeled and analyzed as SODF, and for MDOF systems such as framed structures (DoD, 2008).

2.7.1 Design Criteria for Individual Elements

Most of the design criteria are provided for individual components. Table 2.8 shows the response criteria for some steel components for different levels of response.

Table 2.8. Response limits for different components* (ASCE, 2010).

Component	Component response					
	Low		Medium		High	
	μ	θ	μ	θ	μ	θ
Steel Primary Frame Members (with significant compression)**	1.5	1	2	1.5	3	2
Steel Primary Frame Members (without significant compression)	1.5	1	3	2	6	4
R/C Beams, Slabs, & Wall Panels (no shear reinforcement)	-	1	-	2	-	5

* Response limits are for components responding primarily in flexure

** Significant compression is when the axial compressive load is more than 20% of the dynamic axial capacity of the member.

In Table 2.8, *component response* refers to the level of damage. *Low response* means there is no or only slight visible damage. *Medium response* refers to some permanent damage to the component that can be repaired. A component with high response has not failed, but it has experienced permanent damage that cannot be repaired (see Table 2.7).

2.7.2 Design Criteria for Structural System

The ductility ratio criteria concept for individual members is intractable in the design of frame structures because of the wide range and time-varying nature of the end conditions of components (DoD, 2008). That is, in addition to the support rotation criteria, the side-sway limits should be checked for framed structures. Table 2.9 presents side-sway deflection limits for different levels of response for steel-frame structures.

Table 2.9. Side-sway limits for steel frame structures (ASCE, 2010).

Response	Low Response	Medium Response	High Response
δ	$H/50$	$H/35$	$H/25$

2.8 Progressive Collapse

ASCE (2011) defines progressive collapse as “chain-reaction failure of a building’s structural system or elements as a result of, and to an extent disproportionate to, initial localized damage, such as that caused by an explosion.”

As a result of a blast loading, structural components may fail, and their loads may be distributed to neighboring members. If the surrounding members cannot tolerate this extra load, failure can propagate vertically or horizontally. The entire structural system should be evaluated when a blast is expected to cause local failure or plastic hinges of structural components. In blast-resistant design, local damage is expected, but the whole structural system should be stable.

To prevent progressive collapse, the primary members or key elements must be strengthened, and/or the global structural redundancy should be increased, so that only local failures are permitted.

The DoD (2016) requires that buildings of three or more stories must comply with progressive collapse standards. The reader is referred to work by the DoD (2016) and Marchand and Alfawakhiri (2005) for further details.

2.9 Concluding Remarks

In this review chapter, an overview of topics related to the design of blast-resistant structures is provided. Three methods to predict blast loading are discussed, and the modeling of structural response and material behavior under blast loading is reviewed. Design philosophies and criteria are explained, and basic concepts related to the blast-resistant design field are summarized. References on each topic are provided for further details.

CHAPTER 3

METAHEURISTIC OPTIMIZATION ALGORITHMS

Abstract

In recent years, many nature-inspired metaheuristic optimization algorithms have been proposed in an effort to develop efficient and robust algorithms. The drawback in most of them is the large number of simulations required to obtain good designs. To reduce the number of structural analyses to reach the best design, a new two-phase algorithm is proposed and evaluated. This hybrid algorithm is based on the well-known Harmony Search (HS) algorithm and recently developed Colliding Bodied Optimization (CBO). HS analyzes and improves one design in every iteration whereas CBO generates and analyzes a new population of designs in every iteration. Based on the observed behavior of these two algorithms, a Hybrid Harmony Search - Colliding Bodies Optimization (HHC) is proposed. First phase of HHC uses the Improved Harmony Search (IHS) algorithm. A new design domain adjustment technique is also incorporated in IHS that dramatically reduces the number of possible combinations of discrete variables. This improves performance of the IHS algorithm. The second phase uses the Enhanced Colliding Bodies Optimization (ECBO). ECBO receives final designs from the first phase to enhance them further. This makes the second phase needing fewer iterations in comparison with the ECBO alone. The performance of the proposed algorithms is evaluated using some benchmark discrete structural optimization problems although the method is applicable to continuous variable problems as well. The results show HHC with design domain reduction to be quite effective, robust, and needing smaller number of structural analyses to solve optimization problems in comparison with IHS, ECBO, and some other metaheuristic optimization algorithms. HHC with design domain reduction is shown to be quite robust in the sense that different runs for a problem obtain same final design.

This is an important feature that leads to better confidence in the final solution from single run of the algorithm for a problem.

3.1 Introduction

Calculus-based optimization algorithms were developed more than 50 years ago and a vast amount of literature is available on the subject. Linear programming (LP), nonlinear programming (NLP), and dynamic programming (DP) methods need gradient information to improve the solution estimate (to find a search direction). These methods search for the optimum point in a neighborhood of the current estimate. In comparison with metaheuristic algorithms, these methods converge much faster and can find higher accuracy local solutions.

Gradient-based methods are most appropriate for continuous variables and continuous functions. Many engineering problems have non-smooth functions in their formulation. As a result, gradient-based optimization methods cannot be used to solve such problem. On the other hand, stochastic, metaheuristic or nature-inspired algorithms use only simulation results to reach the final solution, such as the well-known Genetic Algorithms (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and many others (Arora, 2017). The search is not near the current point and the discrete variables and non-differentiable functions can be treated routinely. They use an organized random search in the entire design space instead of gradient-based search in a neighborhood of the current point. Therefore, they are likely to converge to a global optimum point rather than a local optimum. Different runs for the same problem can take different paths to the final solution or even a different solution. The methods are suitable for both continuous and discrete variables and with one or more objective functions.

Just like the gradient-based algorithms, stochastic algorithms have drawbacks. They require large computation time to obtain a reasonable solution. The computation time depends on

the number of the design variables and the range for each design variable; a larger design domain needs more iterations and thus more structural analyses. Increasing the number of iterations could be a good way to find a better design but there is no guarantee that a global optimum design will be found. Therefore, the best way is to run the algorithm more than once and choose the best solution from different runs. This, however, means that more computational effort is needed to solve a problem. A good metaheuristic algorithm has the ability to skip local optima, needs less number of simulations to find the best design, is applicable to different types of problems, and can obtain higher accuracy solutions (Kaveh, 2017).

In an effort to reduce the number of structural analyses to reach the final design, a Hybrid Harmony Search - Colliding Bodies Optimization (HHC) algorithm is proposed and evaluated in this study. This proposal is based on the following observations about the behavior of two algorithms while solving some structural design problems: (1) Improved version of the harmony search algorithm (IHS) (Mahdavi, et al., 2007) makes rapid improvements towards the final design in the initial iterations and then its progress slows down once it is in a neighborhood of the best design, and (2) the enhanced version of the colliding bodies optimization (ECBO) makes steady improvement towards the final design requiring more structural analyses to reach a neighborhood of the final design compared to IHS. Therefore, the basic idea to be explored for the proposed hybrid algorithm is to determine if a combination of the two algorithms can reduce the number of structural analyses to reach the final design. That is, since IHS algorithm can reach neighborhood of the final design more rapidly, it will be used in phase one and its iterations will be terminated once the progress towards the final design slows down; in phase two, the improved designs from IHS will be passed on to the ECBO as its initial population (instead of random designs generated from the entire design domain) to improve the best design further. This may lead to fewer structural

analyses to reach the final design. In addition, a new design domain adjustment technique based on statistically analyzing some designs is added to IHS to increase the possibility of rapidly finding better designs.

A new stopping criterion is also introduced in addition to a limit on the number of iterations for terminating phase one iterations. That is, when the algorithm is not able to find a better design for certain number of iterations, it is terminated.

A major motivation for this work is to investigate procedures that can reduce the number of structural analyses to reach the final designs for the class of structural optimization problems that cannot be solved using the gradient-based algorithms. This becomes critically important while solving more complex structural optimization applications, such as nonlinear static response problems, nonlinear dynamic response problems and multidisciplinary problems. Each simulation of such problems can take enormous computational effort making meta-heuristic methods very time consuming.

Some benchmark discrete variable truss optimization problems are solved using the proposed algorithm. These well-known examples are solved previously in the literature using different metaheuristic algorithms. The results are discussed and compared with the available results in the literature to study performance of the proposed algorithm.

3.2 Formulation of Discrete Structural Optimization Problems

In many practical design cases, design variables are discrete because members must be selected from the available sizes in a catalog. The formulation of the discrete design variables optimization problem is slightly different from continuous design variable optimization. In general, the problem can be stated as:

$$\text{Find } \mathbf{X} = [x_1, x_2, \dots, x_{nvar}]; \quad x_j \in D_j; \quad j = 1, 2, \dots, nvar \quad (3.1)$$

$$\text{to minimize } f(\mathbf{X}) \quad (3.2)$$

$$\text{subject to } g_k(\mathbf{X}) \leq 0; \quad k = 1, 2, \dots, p \quad (3.3)$$

where \mathbf{X} is the vector of design variables with $nvar$ unknowns, D_j is a set of discrete values for the j th design variable, $f(\mathbf{X})$ is a cost function (in this study, $f(\mathbf{X})$ is the total weight of the structure), and $g_k(\mathbf{X})$ is a constraint function.

One way of treating constraints in metaheuristic algorithms is to combine constraints with the cost function to define a merit function (also called the penalty function) that is then minimized (Kaveh and Mahdavi, 2015):

$$F(\mathbf{X}) = f(\mathbf{X})[1 + \psi G(\mathbf{X})]^\xi \quad (3.4)$$

$$G(\mathbf{X}) = \sum_{k=1}^p \max(0, g_k(\mathbf{X})) \quad (3.5)$$

where $G(\mathbf{X})$ is a constraint violation function, $\psi \geq 1$ is exploration penalty coefficient ($\psi = 1$ unless another value is mentioned), $\xi > 1$ is penalty function exponent (in this study, $\xi = 2$), and $\max(0, g_k(\mathbf{X})) \geq 0$ is the violation value of the k th inequality constraint. The present problem has just inequality constraints. However, if equality constraints are present in the problem formulation, they are treated by including their violations in Eq. (3.5).

3.3 Metaheuristic Optimization Algorithms

Over the years, many metaheuristic optimization algorithms have been explored. More recent techniques are based on observations about some natural phenomena, such as survival of the fittest and genetic inheritance in Genetic Algorithms (GA), which is inspired by the basic mechanism of natural evolution developed by Goldberg and Holland (1988); Simulated Annealing

(SA) proposed by Kirkpatrick et al. (1983); Particle Swarm Optimization (PSO) proposed by Kennedy and Eberhart (2001); Ant Colony Optimization (ACO) introduced by Dorigo et al. (1996); Harmony Search (HS) algorithm invented by Geem et al. (2001); Big Bang–Big Crunch algorithm (BB–BC) introduced by Erol and Eksin (2006); Colliding Bodies Optimization (CBO) proposed by Kaveh and Mahdavi (2014); and Ray Optimization (RO), developed by Kaveh and Khayat (2012).

In this study, HS algorithm and its improved version, and CBO and its enhanced version are summarized since the proposed hybrid algorithm HHC uses these procedures.

3.3.1 Harmony Search Algorithm

Geem, Kim, and Longanathan (2001) presented the HS algorithm based on music improvisation process of jazz musicians. The following five steps describe the HS algorithm:

Step 1: Parameter setting

The algorithm starts by initially generating a set of random designs from the design domain. Then in every iteration, a new design is generated and analyzed. If this design is better than the worst design in the current population, then it replaces that design; otherwise, another design is generated. The process is continued until a limit on the number of iterations is reached.

HS has four parameters that need to be initialized before starting the algorithm. There are no general guidelines for their selection; they are selected depending on the problem (Degertekin, 2008). Thus, the best way is to try different values to find the best combination for an application.

The parameters are:

- 1- Harmony memory size (*HMS*). It is the initial number of candidate solutions selected randomly from the design domain. For example, if *HMS* is 10, the algorithm starts by selecting 10 designs

and for every design evaluates the merit function F if the problem is constrained or the objective function f if the problem is unconstrained. This information is saved in a matrix called harmony memory (**HM**).

- 2- Harmony memory consideration ratio ($HMCR$): Its value ranges between 0 and 1. It is the probability of selecting design variables from the current **HM** to generate a new design. Variables selected from current **HM** may go through further adjustment depending on the pitch adjustment rate.
- 3- Pitch adjusting rate (PAR): Its value ranges between 0 and 1 and it is the probability of mutation of the design variable selected from **HM** to a neighboring value.
- 4- Maximum improvisations ($MaxIter_{p1}$): It is a limit on the number of iterations for HS.

Step 2: Initialization

The HS starts with HMS random designs to populate the harmony memory matrix **HM** as:

$$\mathbf{HM} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^{nvar} \\ \vdots & \vdots & \ddots & \vdots \\ x_{HMS}^1 & x_{HMS}^2 & \dots & x_{HMS}^{nvar} \end{bmatrix} \quad (3.6)$$

where $nvar$ is the number of design variables. Thus each row of this matrix represents a design point and each column is associated with a design variable.

Step 3: Harmony improvisation:

A new design point is improvised where each design variable is selected from either the current population of designs in the matrix **HM** or from its possible range of values. These selections are based on harmony memory consideration, pitch adjustment, and random numbers.

Using harmony memory consideration parameter $HMCR$, the new value for the j th design variable is chosen as x_i^j from either the j th column of **HM** or from the allowable values for this

design variable (Degertekin, 2008). For each design variable j ($j = 1$ to $nvar$), the row index i is selected randomly as follows:

$$x_i^j \in \{x_1^j, x_2^j, \dots, x_{HMS}^j\}; \quad \text{if } rn_{HMCR}^j \leq HMCR \quad (3.7)$$

$$x_i^j \in D_j; \quad \text{if } rn_{HMCR}^j > HMCR \quad (3.8)$$

where rn_{HMCR}^j is a random number uniformly distributed over the interval [0,1] and D_j is the allowable set of values for the j th design variable.

Every design variable selected from harmony memory is examined further to determine whether it should be pitch-adjusted or not. The parameter PAR is used for this purpose as follows:

$$\text{Pitch adjusting decision for } x_i^j \begin{cases} \text{yes} & \text{if } rn_{PAR}^j \leq PAR \\ \text{No} & \text{if } rn_{PAR}^j > PAR \end{cases} \quad (3.9)$$

where rn_{PAR}^j is a random number uniformly distributed over the interval [0,1]. If the pitch adjustment decision is “yes” x_i^j is replaced as follow:

$$\begin{aligned} x_{i,new}^j &= x_i^j + 1 \quad \text{if } PAR_{rand}^j < 0.5 \\ x_{i,new}^j &= x_i^j - 1 \quad \text{if } PAR_{rand}^j \geq 0.5 \end{aligned} \quad (3.10)$$

where PAR_{rand}^j is random number uniformly distributed over the interval [0,1], and +1 and -1 means moving to the next higher or lower allowable value for this variable (Geem, 2009).

Step 4: Update the harmony memory:

The new design from step 3 is evaluated. If it is better than the worst design in **HM**, the new design replaces the worst design in **HM**; otherwise, a new design is improvised.

Step 5: Termination criteria

If the limit on number of iterations is reached, terminate the algorithm; otherwise, go to step 3.

3.3.2 Improved Harmony Search Algorithm (IHS)

The concept of IHS is the same as HS (the five steps in Section 3.3.1). However, standard HS algorithm uses fixed value of $HMCR$ and PAR . The main drawback of the standard HS algorithm is that it needs a large number of iterations to find an acceptable solution (Mahdavi et al., 2007).

In IHS, $HMCR$ and PAR are adjusted with every iteration using Eqs. (3.11) and (3.12) to improve the performance of the HS algorithm by eliminating its drawbacks (Sun and Chang, 2015).

$$HMCR(iter) = HMCR_{max} - \frac{(HMCR_{max} - HMCR_{min})}{MaxIter_{p1}} \times Iter_{p1} \quad (3.11)$$

$$PAR(iter) = \frac{(PAR_{max} - PAR_{min})}{\pi/2} \times \arctan(Iter_{p1}) + PAR_{min} \quad (3.12)$$

where $Iter_{p1}$ is the current iteration, $HMCR_{max}$ and $HMCR_{min}$ are maximum and minimum harmony memory consideration ratios, respectively, PAR_{max} and PAR_{min} are maximum and minimum pitch adjacent ratios, respectively. Note that $HMCR$ is a linearly decreasing function of iteration number. This increases the probability of selecting a design variable from its allowable range of values rather than from the harmony memory. Also, PAR is an increasing function of the iteration number that increases the probability of pitch adjustment for a design variable when it is selected from the **HM** matrix. These processes introduce more diversity into the population of design in the **HM** matrix.

Just like HS, there are no guidelines that one can follow to select IHS parameters. Therefore, the best way is to start with a set of values then try different values to find the best combination. In this study, $HMCR_{max}$ and PAR_{max} of 0.85 and $HMCR_{min}$ and PAR_{min} of 0.35 show good performance.

3.3.3 Colliding Bodies Optimization (CBO)

3.3.3.1 Background Material

Kaveh and Mahdavi (2014) developed this metaheuristic algorithm that is inspired by the laws of one-dimensional collision. The algorithm works with a population of design at each iteration. Here each design in the population is considered as an object or body with mass and velocity

Using laws of momentum and energy, collision can be simulated between objects such as two balls in a billiard game or two cars in an accident. If there are no external forces acting on the system, the momentum of all objects before the collision equals the momentum of all objects after the collision. Conservation of linear momentum of two bodies in one-dimensional collision is expressed as:

$$m_1 v_1 + m_2 v_2 = m_1 v'_1 + m_2 v'_2 \quad (3.13)$$

where m_1 , v_1 , and v'_1 are mass, initial velocity and final velocity of the first object, respectively, and m_2 , v_2 , and v'_2 are mass, initial velocity and final velocity of the second object, respectively.

For one dimensional collision, let body 1 approach and collide with body 2; therefore $v_1 > v_2$. After the collision, the bodies separate; therefore $v'_2 > v'_1$. The system loses some of its energy during the collision. The Coefficient of Restitution (COR) $\varepsilon \geq 0$ indicates how much kinetic energy remains in the system after collision that is defined as:

$$\varepsilon = \frac{\text{Velocity of separation after collision}}{\text{Velocity of approach before collision}} = \frac{v_2' - v_1'}{v_1 - v_2} \quad (3.14)$$

Using Eqs. (3.13) and (3.14) the velocities after collision are calculated as follows (Kaveh and Mahdavi, 2014):

$$v_1' = \frac{(m_1 - \varepsilon m_2)v_1 + (1 + \varepsilon) m_2 v_2}{m_1 + m_2} \quad (3.15)$$

$$v_2' = \frac{(m_2 - \varepsilon m_1)v_2 + (1 + \varepsilon) m_1 v_1}{m_1 + m_2} \quad (3.16)$$

There are two cases of collision:

- i- A perfect elastic collision. There is no loss of kinetic energy in collision ($\varepsilon = 1$).
- ii- An inelastic collision. There is part of the kinetic energy that is changed to some other form of energy ($\varepsilon < 1$). For the most real bodies, the value of ε is between 0 and 1.

3.3.3.2 Colliding Bodies Optimization

In Colliding Bodies Optimization (CBO), the Colliding Bodies (CBs) (the current population of designs) are divided into two equal groups: stationary and moving objects (Kaveh and Mahdavi, 2014). The moving objects move toward and collide the stationary objects causing:

- 1- Stationary objects to move to another position.
- 2- Moving objects to change their position.

After the collision, the position of both colliding and stationary bodies (population of designs) are updated using new velocities from Eqs. (3.15) and (3.16). The CBO can be executed in following three steps:

Step 1: Initialization. Initializing an array of CBs (initial population of designs) with random positions as follow:

$$x_i^j = x_{j,min} + rn_i^j \times (x_{j,max} - x_{j,min});$$

$$i = 1, 2, \dots, 2n \text{ and } j = 1, 2, \dots, nvar$$
(3.17)

where x_i^j is the j th variable of the i th design in the **CB** matrix, $x_{j,min}$ and $x_{j,max}$ are the lower and the upper bounds of j th design variable, rn_i^j is a random number between 0 and 1, $2n$ is the total number of CBs or the population size, and $nvar$ is number of design variables. To obtain discrete values for design variables, x_i^j is rounded to the nearest permissible discrete value.

Step 2: Search. This step is divided into 4 sub-steps:

1- *CBs ranking*: use the merit function $F(\mathbf{X})$ to compute CBs' masses (Eq. (3.18)), and sort the CBs' in a descending order based on their calculated masses:

$$m_i = \frac{1/F_i(\mathbf{X})}{\sum_{k=1}^{2n} 1/F_k(\mathbf{X})}; \quad i = 1, 2, \dots, 2n$$
(3.18)

where m_i is the mass of the i th body (design), $F_i(\mathbf{X})$ and $F_k(\mathbf{X})$ are the merit function values of the i th and k th bodies (designs), respectively. This way the designs are sorted from the best to the worst. Note that larger mass in Eq. (3.18) corresponds to a smaller value for the merit function.

2- *Groups creation*. CBs are equally divided into two groups:

(i) *Stationary CBs*: these are the upper half of CBs; these better designs that are assigned zero velocities before collision:

$$\mathbf{v}_s = \mathbf{0}; \quad s = 1, 2, \dots, n$$
(3.19)

where \mathbf{v}_s is the velocity of the s th CB in the stationary group.

(ii) *Moving group*: these are the lower part of CBs and they move toward the stationary CBs with velocity before collision as:

$$\mathbf{v}_m = \mathbf{X}_m - \mathbf{X}_s; \quad m = n + 1, \dots, 2n \text{ and } s = m - n \quad (3.20)$$

where \mathbf{v}_m and \mathbf{X}_m are the velocity and position of the m th CB in the moving group, respectively, and \mathbf{X}_s is the s th CB position in the stationary group.

3- *Evaluation after the collision.* After the collision, velocities of stationary and moving CBs are calculated based on inelastic one dimensional collision of two bodies using Eqs. (3.15) and (3.16):

$$\mathbf{v}'_s = \frac{(1 + \varepsilon) m_m \mathbf{v}_m}{m_m + m_s}; \quad s = 1, 2, \dots, n \text{ and } m = s + n \quad (3.21)$$

and the velocity of the moving CBs is obtained as:

$$\mathbf{v}'_m = \frac{(m_m - \varepsilon m_s) \mathbf{v}_m}{m_m + m_s}; \quad m = n + 1, \dots, 2n \text{ and } s = m - n \quad (3.22)$$

$$\varepsilon = 1 - \frac{Iter_{p2}}{MaxIter_{p2}} \quad (3.23)$$

where \mathbf{v}'_s is the velocity of the s th CB of stationary group after collision; \mathbf{v}_m and \mathbf{v}'_m are the velocity of the m th CB of the moving group before and after collision, respectively; m_s is the mass of the s th CB of the stationary group; m_m is the mass of the m th CB of the moving group; ε is the COR parameter; $Iter_{p2}$ is the current iteration of ECBO; and $MaxIter_{p2}$ is the limit on number of iterations for ECBO. Note that ε is a decreasing function of the iteration number.

4- CBs updating. The new position of CBs are calculated as follows:

$$\mathbf{X}_s^{new} = \mathbf{X}_s + [\mathbf{rn}_s] \mathbf{v}'_s; \quad s = 1, 2, \dots, n \quad (3.24)$$

$$\mathbf{X}_m^{new} = \mathbf{X}_m + [\mathbf{rn}_m] \mathbf{v}'_m; \quad m = n + 1, \dots, 2n \quad (3.25)$$

where \mathbf{X}_s^{new} and \mathbf{X}_m^{new} are the new positions of the stationary and moving bodies, respectively, \mathbf{X}_s and \mathbf{X}_m are the old positions of the stationary and moving bodies, respectively, \mathbf{rn}_s and \mathbf{rn}_m are diagonal matrices with diagonal elements as random numbers between -1 and 1. To obtain discrete values of designs, \mathbf{X}_s^{new} and \mathbf{X}_m^{new} are rounded to the nearest permissible discrete values.

Step 3: Terminating criterion control

If the limit on number of iterations ($MaxIter_{p2}$) is reached the algorithm is terminated.

Otherwise, go to Step 2.

3.3.3.3 Enhanced Colliding Bodies Optimization (ECBO)

This metaheuristic algorithm is an enhancement of the standard CBO. It uses memory to save some good designs and a mechanism to escape from local optima to get better solutions faster. This is done by adding two more sub-steps to step 2 of the standard CBO as follows (Kaveh and Mahdavi, 2015):

1- *Saving*: this sub-step is added between sub-steps i and ii in step 2 of the standard CBO. In this sub-step, some historically good designs (having smaller merit function values) and their related information are saved in a matrix called Colliding Memory (**CM**). The good designs saved in **CM** replace the worst designs in the current population at the beginning of every iteration. After that the **CM** is also updated. Number of designs saved is CMS .

2- *Escaping from local optima*: this sub-step is added after the last sub-step of step 2 of the standard CBO. In this sub-step, a parameter called Pro within $[0, 1]$ is introduced. For each colliding body, rnp_i ($i = 1, 2, \dots, 2n$), which is a random number uniformly distributed within $[0, 1]$, is compared with Pro . If $Pro > rnp_i$, one component of the i th CB is selected randomly and its value is regenerated as:

$$x_i^j = x_{j,min} + rnp \times (x_{j,max} - x_{j,min}); \quad i = 1, 2, \dots, 2n \quad (3.26)$$

where x_i^j is the j th variable of the i th design, rnp is a random number between 0 and 1, and $x_{j,min}$ and $x_{j,max}$ are the lower and upper bounds of the j th variable, respectively. The reason to change

just one component of *ith* CB is to protect the structures of CBs. This mechanism was shown to give diversity leading to better designs (Kaveh and Mahdavi, 2015).

3.4 HHC: Hybrid Improved Harmony Search-Enhanced Colliding Bodies Algorithm

3.4.1 Motivation for Hybrid Algorithm

Compared to other metaheuristic algorithms, ECBO is simple, requires just one algorithmic parameter, and performs well in term of the quality of the solution. IHS is easy to implement and it works fine with any kind of problem. However, both have some shortcomings that were observed while solving some problems. IHS needs specification of several algorithmic parameters that can affect performance of the algorithm. ECBO makes steady progress towards the neighborhood of the final design whereas IHS makes quite rapid progress towards a similar neighborhood. Therefore, IHS requires fewer structural analyses compared to ECBO to reach a neighborhood of the final design. However, after reaching the neighborhood of the final design, progress of IHS is quite slow to reach the final design whereas ECBO continues to make good progress towards the solution.

Basic idea of the proposed HHC algorithm is to use IHS in Phase 1 to reach the neighborhood of the solution quickly and then switch to the ECBO to reach the final design. This way ECBO starts with some improved designs in Phase 2. This combination could lead to the final solution in fewer structural analyses which will be very useful while solving more complex problems, such as dynamic response optimization problems with discrete variables and non-differentiable functions.

3.4.2 Phase 1: Improved Harmony Search (IHS)

IHS is used in Phase 1 to obtain a good set of designs quickly for Phase 2. Two additional steps are added to IHS:

i- *Stopping Criteria*: In addition to a maximum number of iteration criterion discussed in step 5 in section 3.1, a new merit function improvement criterion is added. That is, when there is no or small improvement in the current merit function value after many iterations, this phase is terminated.

The pseudo-code of this criterion is as follows:

If₁ $Iter_{p_1} \geq r_1 \times MaxIter_{p_1}$

$$If_2 (Merit(Iter_{p_1}) - Merit(Iter_{p_1} - r_2 \times MaxIter_{p_1})) / Merit(Iter_{p_1}) \leq \varepsilon_{p_1}$$

Terminate Phase 1

End₂

End₁

$$MaxIter_{p_1} = 10 \times nvar \times \text{number of elements in the discrete set} \quad (3.27)$$

where $Iter_{p_1}$ is the current iteration, $MaxIter_{p_1}$ is the limit number of iterations for Phase 1.

Note that the parameters r_1 , r_2 and ε_{p_1} are selected so that premature termination of the algorithm does not occur. They do not affect performance of the algorithm in any other way. The limit on number of iterations, $MaxIter_{p_1}$ in Eq. (3.27), is dependent on the number of design variables and the number of elements in the discrete set. When the number of design variables and/or the number of elements in the discrete set increase, the search space enlarges. Therefore, metaheuristic algorithms need more iterations. Thus, Eq. (3.27) is used instead of a fixed number for each problem.

ii- *Domain adjustment*: During the first few iterations (compared with total number of iterations), IHS improves initial designs rapidly. Although, at this stage, the best design may be far from the final design, it was observed that some of the design variables in **HM** have the same or about the same values from iteration to iteration. These design variables are most likely at their best values

in this phase. That is, the allowable range for these design variables can be reduced based on their mean value and standard deviation. In other words, the design domain can be reduced based on the current state of **HM**. Section 3.4.4 provides more details for this step.

3.4.3 Phase 2: Enhanced Colliding Bodies Optimization (ECBO)

ECBO starts with $2n$ random designs and it keeps colliding them in search for a better solution, as explained earlier. Thus, if the initial population is not reasonably good, the algorithm most likely needs more iterations to find the final design. In each iteration, ECBO needs to evaluate the problem functions $2n$ times, where $2n$ is the population size.

In HHC, some better designs generated by Phase 1 are passed on to the **CB** matrix. Then ECBO collides those designs to enhance them further. That is, starting with better designs, the total number of iterations for the ECBO algorithm can be reduced to obtain the final design. This is quite beneficial since ECBO needs to evaluate the problem functions $2n$ times in one iteration. For example, if the population size is 50 in ECBO and the number of iterations to enhance the initial population are 100, then ECBO alone needs 5000 structural analyses to improve the starting population. However, this improvement may be done with fewer structural analyses by replacing the initial population with some better designs of Phase 1 results. Using the same population sizes of 50, 75, and 100 for both phases (passing all Phase 1 designs to Phase 2) did not improve performance of the algorithm in term of quality of the solutions and the number of structural analyses needed to obtain the final designs. Overall, passing just some better designs of Phase 1 to Phase 2 makes the algorithm obtain the final design more often with smaller number of structural analyses.

In this phase, the stopping criterion is a maximum number of iterations as follows:

$$MaxIter_{P2} = nvar \times \text{number of elements in the discrete set} \quad (3.28)$$

Similar to Eq. (3.27), Eq. (3.28) is based on number of design variables and number of elements in the discrete set.

3.4.4 Domain Adjustment Technique

This additional step is added to Phase 1 to increase the possibility for IHS to find better designs faster to enhance the general performance of HHC in term of the number of structural analyses required to reliably find the best design. Domain reduction can be done by looking at the standard deviation of each design variable values for some better designs in the **HM** matrix. When a design variable has a small standard deviation, its upper and lower limits in the allowable set of discrete values D_j for the j th design variable is changed as follows:

$$x_{j,min} = x_{j,avg} - x_{j,sd} \quad (3.29)$$

$$x_{j,max} = x_{j,avg} + x_{j,sd} \quad (3.30)$$

$$x_{j,avg} = \frac{1}{nd} \sum_{i=1}^{nd} (x_i^j) \quad (3.31)$$

$$x_{j,sd} = \sqrt{\frac{1}{nd-1} \sum_{i=1}^{nd} (x_i^j - x_{j,avg})^2} \quad (3.32)$$

where $x_{j,min}$ and $x_{j,max}$ are the lower and upper bounds of the j th design variable, respectively, $x_{j,avg}$ is the average of j th design variable, $x_{j,sd}$ is the standard deviation of j th design variable, and nd is number of designs that are considered in calculating the average and the standard deviation. Designs that are considered in this step to find the new upper and lower limits are important. Therefore, this step starts only after a certain number of iterations so that IHS has already improved initial designs enough. IHS with domain adjustment shows a better convergence

behavior over IHS without domain reduction. Domain reduction makes IHS obtain better designs in fewer iterations.

Sometimes, the upper and lower limits of a design variables become equal (the standard deviation is zero). That is, to avoid trapping in local optima, at least five elements are kept in the discrete set by modifying the upper and lower limits as follows:

$$x_{j,min} = x_{j,avg} - 2 \quad (3.33)$$

$$x_{j,max} = x_{j,avg} + 2 \quad (3.34)$$

where -2 and +2 imply two elements below and two elements above the average value. Also, when the best design in the current **HM** has a design variable is at the modified lower or the upper bound, the current lower or upper bound is adjusted using Eq. (3.33) or Eq. (3.34), respectively. The proposed domain adjustment technique is dynamic. That is, the lower and upper bounds are adjusted based on the best design and the *nd* better designs in **HM** at each iteration.

The domain reduction step starts when there are feasible and nearly feasible designs in the **HM** matrix (designs that have constraint violation of 5% or less). The minimum number of feasible or nearly feasible designs should not be one so that the upper and lower bounds become the same (in this study, 5% of the population is used as the minimum number of feasible or nearly feasible designs). After sorting of designs in **HM** from the best to the worst, the domain reduction procedure for each design variable is implemented using the following pseudo-code:

If₁ $Iter_{p1} \geq r_3 \times MaxIter_{p1}$

If₂ number of feasible or nearly feasible designs $\geq 5\%$ of *HMS*

change the lower and upper limits using Eqs. (3.29) and (3.30), respectively.

If₃ number of elements in the discrete set of the *jth* design variable < 5

change the lower and upper limits using Eqs. (3.33) and (3.34), respectively.

End₃

If₄ $x_{j,best} \leq x_{j,min}$ or $x_{j,best} \geq x_{j,max}$

change the lower or upper limits using Eq. (3.33) or Eq. (3.34), respectively.

End₄

End₂

End₁

Here r_3 is the percentage of $MaxIter_{p1}$ to start this criterion. r_3 should be selected so that IHS has already improved designs. Based on observing IHS convergence behavior, it is recommended to use $r_3 \geq 10\%$ of $MaxIter_{p1}$. This way, the standard deviation can give more accurate results about the design variable state. $x_{j,best}$ is the value of the j th design variable of the best design in **HM**.

Reduction of the design variable bounds shrinks the feasible set for the problem. This increases the possibility of obtaining better designs at the end of Phase 1 with reduced number of structural analyses. Including this technique in Phase 2 showed no improvement in the performance of the algorithm since ECBO can efficiently treat larger domains for design variables. Therefore domain reduction scheme is not suggested for the ECBO. First numerical example in Section 3.5.1 is used to show how this step reduces the design domain and enhances the performance of the HHC significantly.

3.4.5 Evaluation of the Algorithms

Multiple runs for the same problem will be executed to study performance of the algorithms. Several metrics will be used in evaluations:

- 1- *Average of the final merit function values obtained with different runs.* Average value that is closer to the best solution will indicate ability of the algorithm to obtain the best design more often.
- 2- *Standard deviation of the final values of the merit functions obtained with different runs.* Smaller value of the standard deviation will imply robustness of the algorithm to obtain the best design with different runs for the problem.
- 3- *Average of the number of structural analyses needed to reach the final solution.* Smaller value will indicate more efficient algorithm.
- 4- *Standard deviation of the number of structural analyses.* Smaller value will indicate robustness of the algorithm to obtain the final design in approximately same number of structural analyses with different runs for the same problem.

3.5 Numerical Examples

Before the proposed HHC algorithm can be used to solve more complex and larger problems, it needs to be tested to solve some standard test problems and study its performance. In the following sections, some of popular discrete truss optimization examples are solved for minimum structural weight to compare the performance of HHC with other metaheuristic optimization algorithms. Structures are analyzed using finite element (direct stiffness) method and algorithms are coded using MATLAB. For all problems, Phase 1 parameters are set as follows: HMS is 75, $HMCR_{max}$ and PAR_{max} are 0.85 and $HMCR_{min}$ and PAR_{min} are 0.35. Phase 2 parameters are set as follows: population size ($2n$) is 40, Pro is 0.5, and the number of designs to be saved in **CM** (CMS) is 4 ($2n/10$). Phase 1 improvement criterion ratios, r_1 and r_2 , are 0.25 and 0.10, respectively. Domain reduction ratios, r_3 is 0.10 and ε_{p1} is 10^{-3} . These parameters are

selected based on studying the convergence behavior of IHS. IHS obtains good and diverse designs (in comparison with initial random designs) after about 25% of the maximum number of iterations; then it converges very slowly. HHC's performance was evaluated using population sizes of 50, 75 and 100 for Phase 1 and population sizes of 20, 30, 40, and 50 for Phase 2 with *Pro* of 0.25, 0.4 and 0.5. The results showed that the combinations of these parameters worked well that showed stability and gave good quality solutions. It is noted that these parameters are not problem dependent and are kept fixed for all design examples.

Since the optimization algorithms are stochastic in nature, 50 independent optimization runs were performed for each example to test the performance of HHC

The number of maximum iterations varies based on the number of design variables and the number of elements in the discrete set (Eqs. (3.27) and (3.28)). For IHS and ECBO, the maximum numbers of iterations were set to 50000 and 1000, respectively to allow these algorithms to fully search for the best design. NSA (Number of Structural Analyses) is calculated as follows:

$$NSA_{HHC} = HMS + Niter_{P1} + (Niter_{P2} - 1) \times \text{population size} \quad (3.35)$$

$$NSA_{ECBO} = Niter \times \text{population size} \quad (3.36)$$

$$NSA_{IHS} = HMS + Niter \quad (3.37)$$

where *HMS* is harmony memory size, *Niter* is the number of iterations and the subscript refers to the phase. In Eq. (3.35), $(Niter_{P2} - 1)$ implies that the designs passed to Phase 2 do not need to be evaluated again.

To study the domain adjustment effects on the behavior of HHC, all numerical examples were tested without domain adjustment step as well. In the next sections, HHC refers to the

algorithm without domain adjustment step while HHCD refers to the algorithm with domain adjustment step.

3.5.1 Planar 10-bar Truss

Figure 3.1 shows the configuration of the 10-bar truss. This popular benchmark example has been solved by many researchers, e.g., Rajeev and Krishnamoorthy (1992), Li et al. (2009), Xiang et al. (2009), Camp (2009), and others. For all members, the modulus of elasticity is 10,000 ksi and material density is 0.1 lb/in³. The allowable displacement for all nodes in both vertical and horizontal directions equals ± 2.0 in. All members are subjected to stress limitations of 25 ksi for both tension and compression. The structure is subjected to two vertical downward loads, $P=100$ kips, at joint 2 and 4. Cross-sectional areas of all members are the design variables that are selected from the discrete set of 42 elements as follows:

$$D=[1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, \\ 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, \\ 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, \\ 26.50, 30.00, 33.50] \text{ (in}^2\text{)}. \quad (3.38)$$

As noted earlier, this study case is used to also show how HHCD works. In this illustrative example, $MaxIter_{p_1}=10 \times 10 \times 42=4200$ and $MaxIter_{p_2}=10 \times 42=420$. The rest of the internal parameters are set as mentioned earlier. Phase 1 (IHS) starts with 75 random designs that are evaluated using Eq. (3.4). Table 3.1 gives the best design among these 75 initial designs. The total structural weights (and merit function values) for HHC and HHCD are 4888.346 lb (38724.564) and 4929.869 lb (26734.436) with values of the violation parameter G as 1.815 and 1.329, respectively. After iteration 420 ($0.1 \times MaxIter_{p_1}$), the domain reduction process starts. Table 3.1 shows that at the end of Phase 1 some design variables bounds were reduced based on the method

discussed in Section 3.4.4. Due to this step, the possible design combinations are reduced from $1.708e16$ (42^{10}) to $5.976e9$ in Phase 1. For example, the bounds of the first design variable were changed many times after iteration 420 until its upper and lower limits became sections 28 and 42 in the set D, respectively.

At the end of the Phase 1, the structural weights for HHC and HHCD are 5959.483 lb and 5788.563 lb, respectively. Since there is no violation of constraints, the merit function value is same as the structural weight. Phase 1 terminates at iteration 1051 for HHC and HHCD while the maximum number of iterations allowed for this phase is 4200. This implies that the proposed new stopping criterion terminates this phase due to no improvement in the current best design.

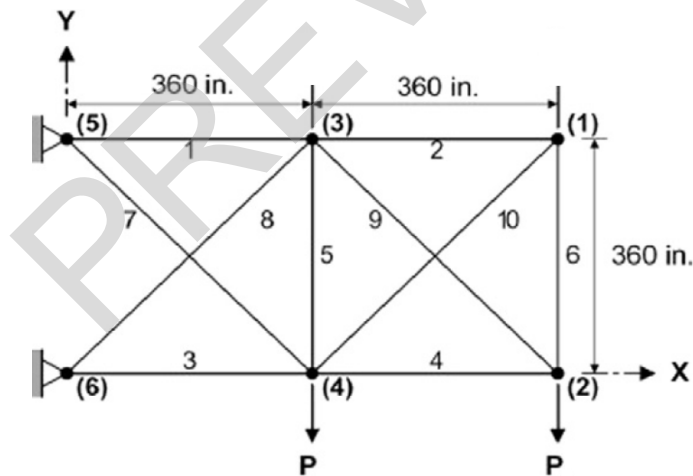


Figure 3.1. Schematic of 10-bar planar truss.

The goal of the Phase 1 is to provide Phase 2 with better designs that may be closer to the best solution so that ECBO requires fewer iterations. Study of designs in **HM** shows that HHCD was able to improve not only the best design but also all the designs in **HM**. This explains the reason that HHCD needs fewer number of structural analyses. At the end of Phase 1, the best 40 designs in **HM** are passed to the **CB** matrix. Therefore, the Phase 2 starts with improved designs instead of random designs. In this example, Phase 1 iterations of 1051 are equivalent to just 26

Table 3.1. Domain adjustment technique illustration for planar 10-bar truss structure.

Design variables (in ²)	Best initial design		Best design at end of Phase 1		Best design at end of Phase 2		Design variables bounds at end of Phase 1 ^b	
	HHC	HHCD	HHC	HHCD	HHC	HHCD	HHCD	
							Lower bound	Upper bound
1 A1	15.50 (33 ^a)	16.90 (35)	33.50 (42)	30.00 (41)	33.50 (42)	33.50 (42)	28	42
2 A2	1.62 (1)	13.9 (31)	2.38 (5)	2.38 (5)	1.62 (1)	1.62 (1)	1	7
3 A3	33.50 (42)	13.5 (30)	26.50 (40)	30.00 (41)	22.90 (39)	22.90 (39)	27	42
4 A4	3.87 (17)	14.20 (32)	13.50 (30)	14.20 (32)	14.20 (32)	14.20 (32)	24	34
5 A5	3.55 (14)	7.97 (28)	3.13 (11)	3.09 (10)	1.62 (1)	1.62 (1)	2	12
6 A6	4.18 (19)	5.12 (25)	3.63 (15)	2.38 (5)	1.62 (1)	1.62 (1)	1	7
7 A7	3.84 (16)	3.84 (16)	14.20 (32)	13.50 (30)	7.97 (28)	7.97 (28)	28	32
8 A8	22.00 (38)	22.90 (39)	22.90 (39)	22.90 (39)	22.90 (39)	22.90 (39)	21	41
9 A9	4.18 (19)	3.47 (13)	19.90 (37)	16.90 (35)	22.00 (38)	22.00 (38)	30	37
10 A10	22.00 (38)	16.00 (34)	1.62 (1)	2.38 (5)	1.62 (1)	1.62 (1)	2	6
Weight (lb)	4888.346	4929.869	5959.483	5788.563	5490.738	5490.738	-	-
<i>G</i> (Eq.5)	1.815	1.329	0.0	0.0	0.0	0.0	-	-
<i>F</i> (Eq.4)	38724.564	26734.436	5959.483	5788.563	5490.738	5490.738	-	-

^aSection number in the set D. ^bLower and upper bounds for HHC remain fixed at 1 and 42 for all members.

iterations of ECBO with a population size of 40. At iteration 1126 (1051 for Phase 1 and 75 for Phase 2), the algorithm obtains the best structural weight of 5490.738 lb with no constraint violation. Phase 2 needs 76 iterations to find the best design that, generally, is less than what ECBO would need (see Table 3.2). HHCD shows similar behavior in all other numerical examples.

Figure 3.2 demonstrates the convergence history of the best run of IHS, ECBO, HHC, and HHCD. It shows that IHS and Phase 1 of HHC and HHCD reach better designs faster than ECBO. HHCD convergences to the best design faster than ECBO and HHC because Phase 2 starts with better designs. When IHS stops improving design at iteration 4101. Note that every iteration in ECBO and Phase 2 of HHC and HHCD requires 40 structural analyses.

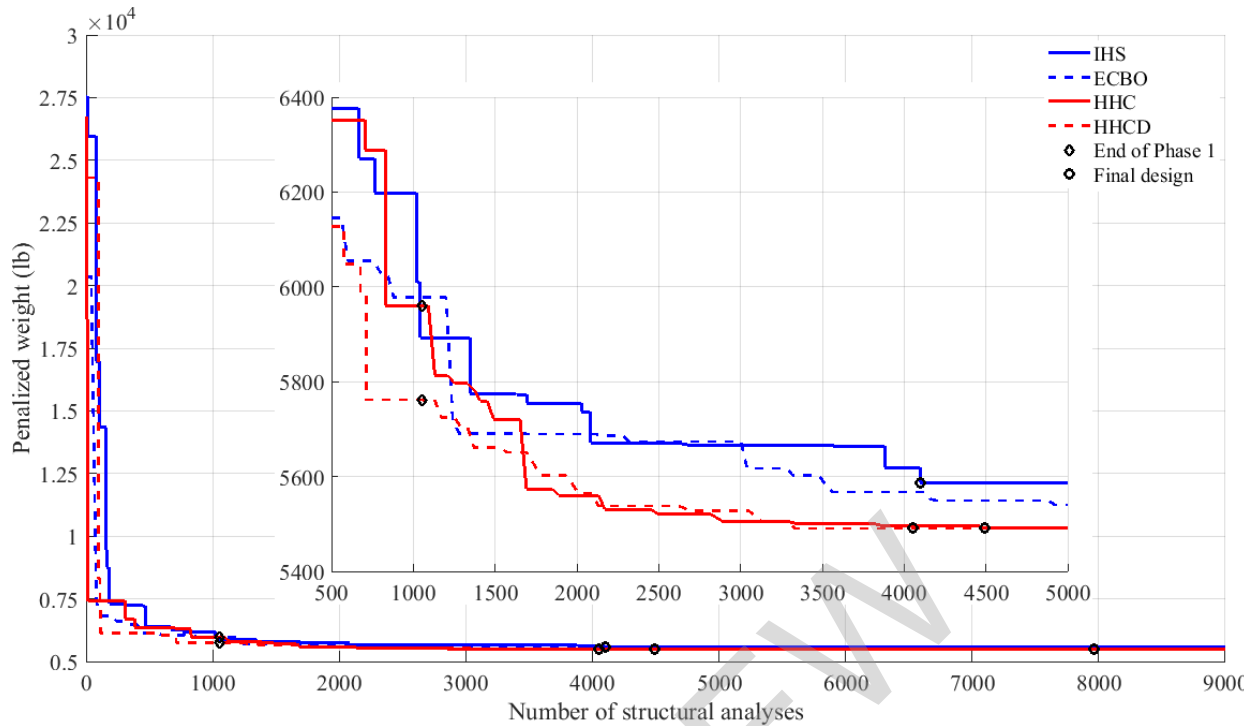


Figure 3.2. Comparison of convergence rates for planar 10-bar truss.

Table 3.2 summarizes results available in the literature with four different algorithms, and results from the present study. It also shows the mean values and standard deviations of the best structural weight from 50 independent runs for IHS, ECBO, HHC, and HHCD. The results show that GA, HPSO, and IHS did not obtain the best design. HHCD was able to find the best design after 4126 structural analyses. This is the same weight as obtained by SA, BB-BC, ECBO, and HHC; however, HHCD needs fewer structural analyses to obtain the best solution.

Figure 3.3 shows the best merit function value for each of the 50 runs for IHS, ECBO, HHC, and HHCD. It is seen that IHS was not able to obtain the final design in any run; HHC was able to reach the final design 42 times; and ECBO reached the final design 26 times. It is seen that HHC performs better than ECBO as well as IHS.

Figure 3.3 shows that HHCD was able to find the best solution 49 times. The average and the standard deviation of 50 runs (Table 3.2) and Figure 3.3 demonstrate that HHCD is very

effective and robust algorithm. Its average for the structural weight is closest to the best solution and its standard deviation is the smallest. This is important because it shows that HHCD does not require multiple runs to find the best solution. The average and standard deviation of number of structural analyses shows that HHCD is efficient (Table 3.2). IHS has the lowest NSA average but the quality of the solution is not good. The complete code for this numerical example is given in the appendix.

Table 3.2. Comparison of optimal designs for 10-bar truss problem.

Design variable (in ²)		GA (Rajeev and Krishna moorthy 1992)	HPSO ^f (Li et al., 2009)	SA (Xiang et al., 2009)	BB-BC (Camp, 2009)	This work			
						IHS ^d	ECBO ^e	HHC	HHCD
1	A1	33.50	30.00	33.50	33.50	30.00	33.50	33.50	33.50
2	A2	1.62	1.62	1.62	1.62	2.62	1.62	1.62	1.62
3	A3	22.00	22.90	22.90	22.90	22.90	22.90	22.90	22.90
4	A4	15.50	13.50	14.20	14.20	14.20	14.20	14.20	14.20
5	A5	1.62	1.62	1.62	1.62	1.80	1.62	1.62	1.62
6	A6	1.62	1.62	1.62	1.62	1.80	1.62	1.62	1.62
7	A7	14.20	7.97	7.97	7.97	11.50	7.97	7.97	7.97
8	A8	19.90	26.50	22.90	22.90	22.00	22.90	22.90	22.90
9	A9	19.90	22.00	22.00	22.00	22.00	22.00	22.00	22.00
10	A10	2.62	1.80	1.62	1.62	2.38	1.62	1.62	1.62
Best weight (lb)		5613.580	5531.984	5490.738	5490.738	5586.289	5490.738	5490.738	5490.738
NSA ^a		800	50000	10500	8694	4176	7960	4566	4126
G (Eq. 5)		3.77×10 ⁻⁴	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Weight (lb)	Average	N/A	N/A	N/A	5494.17 ^c	5680.406	5519.357	5499.116	5490.873
	SD ^b	N/A	N/A	N/A	12.42	40.582	53.183	30.732	0.943
NSA	Average	N/A	N/A	N/A	N/A	6999	19378	9821	8979
	SD	N/A	N/A	N/A	N/A	2728	6215	5038	3890

^aNSA is number of structural analyses. ^bSD is the standard deviation of 50 independent runs. ^cThe average of 100 runs.

^dThe maximum number of iterations is 50000. ^eThe maximum number of iterations is 1000.

^fHPSO is Harmony Particle Swarm Optimization.

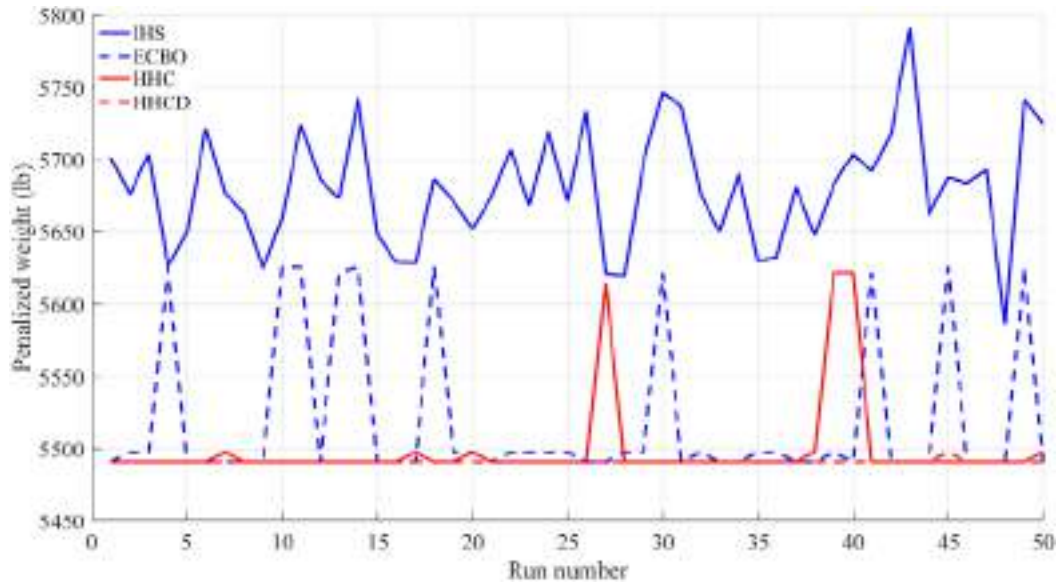


Figure 3.3. Comparison of best designs from 50 runs for the 10-bar truss structure.

3.5.2 Planar 15-bar Truss

Figure 3.4 shows the configuration of the 15-bar truss. Previously, this example was solved in Li et al. (2006) and Li et al. (2009). For all members, the modulus of elasticity is 200 GPa and material density is 7800 kg/m³. The allowable displacement for all nodes in both vertical and horizontal directions is ± 10 mm. All members are subjected to stress limit of 120 MPa for both tension and compression. As shown in Figure 3.4, the structure is subjected to three vertical point loads with three independent load cases: case 1: $P_1 = 35$ kN, $P_2 = 35$ kN, and $P_3 = 35$, case 2: $P_1 = 35$ kN, $P_2 = 0$ kN, and $P_3 = 35$, and case 3: $P_1 = 35$ kN, $P_2 = 35$ kN, and $P_3 = 0$. Design variables are selected from the discrete set of 16 elements:

$$D = [113.2, 143.2, 145.9, 174.9, 185.9, 235.9, 265.9, 297.1, 308.6, 334.3, 338.2, 497.8, 507.6, 736.7, 791.2, 1063.7] \text{ (mm}^2\text{)}. \quad (3.39)$$

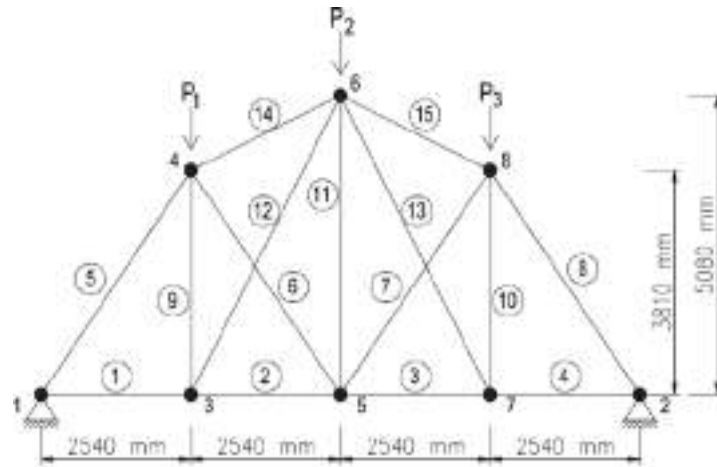


Figure 3.4. Schematic of 15-bar planar truss.

In this study case, ψ is 10 (Eq. (3.4)) because algorithms found some infeasible solutions with very small violation when ψ is 1. Table 3.3 shows comparison of the cross-sectional areas, the best structural weights, NSA when the algorithms reach the best design. It also contains the mean values and standard deviations for structural weight from 50 independent runs for IHS, ECBO, HHC, and HHCD.

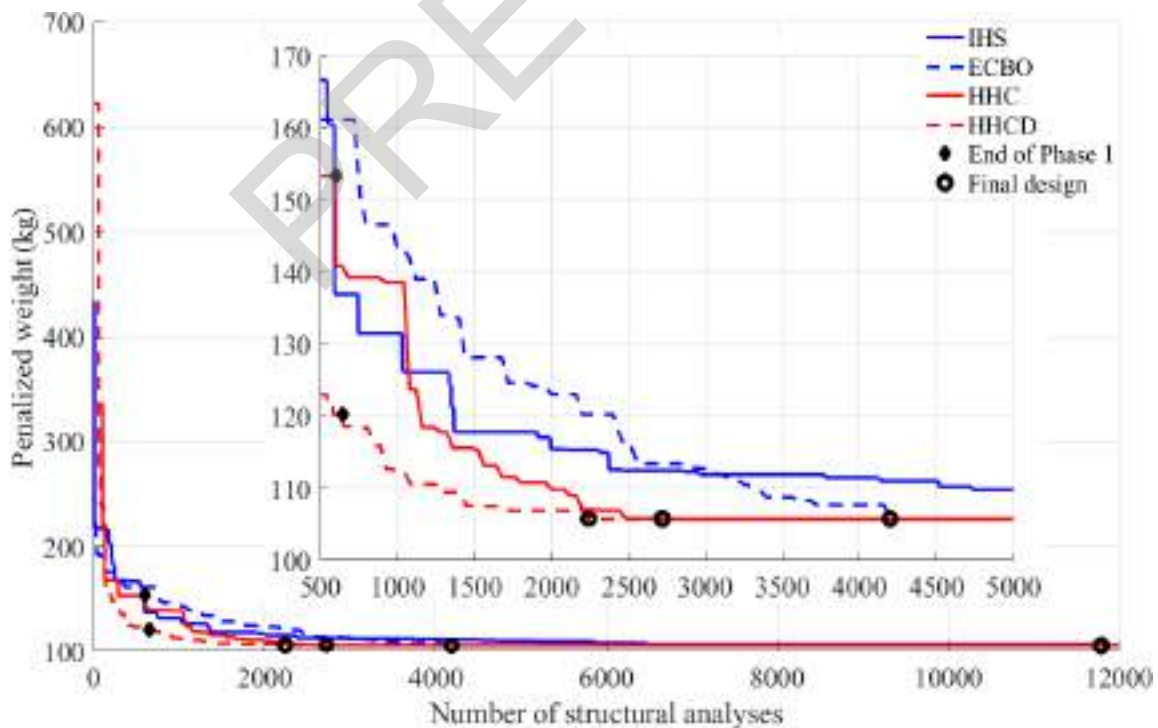


Figure 3.5. Comparison of convergence rates for planar 15-bar truss.

Table 3.3. Comparison of optimal designs for 15-bar truss problem.

Design variables (mm ²)		PSO (Li et al., 2006)	HPSO (Li et al., 2009)	This work			
				IHS	ECBO	HHC	HHCD
1	A1	185.9	113.2	113.2	113.2	113.2	113.2
2	A2	113.2	113.2	113.2	113.2	113.2	113.2
3	A3	143.2	113.2	113.2	113.2	113.2	113.2
4	A4	113.2	113.2	113.2	113.2	113.2	113.2
5	A5	736.7	736.7	736.7	736.7	736.7	736.7
6	A6	143.2	113.2	113.2	113.2	113.2	113.2
7	A7	113.2	113.2	113.2	113.2	113.2	113.2
8	A8	736.7	736.7	736.7	736.7	736.7	736.7
9	A9	113.2	113.2	113.2	113.2	113.2	113.2
10	A10	113.2	113.2	113.2	113.2	113.2	113.2
11	A11	113.2	113.2	113.2	113.2	113.2	113.2
12	A12	113.2	113.2	113.2	113.2	113.2	113.2
13	A13	113.2	113.2	113.2	113.2	113.2	113.2
14	A14	334.3	334.3	334.3	334.3	334.3	334.3
15	A15	334.3	334.3	334.3	334.3	334.3	334.3
Best weight (kg)		108.841	105.735	105.735	105.735	105.735	105.735
NSA		N/A	25000	11861	4240	2836	2402
G (Eq. 5)		N/A	N/A	0.0	0	0	0
Weight (kg)	Average	N/A	N/A	105.993	108.537	105.735	105.735
	SD	N/A	N/A	0.341	5.441	0.0	0.0
NSA	Average	N/A	N/A	15734	4169	3986	3624
	SD	N/A	N/A	4781	790	757	764

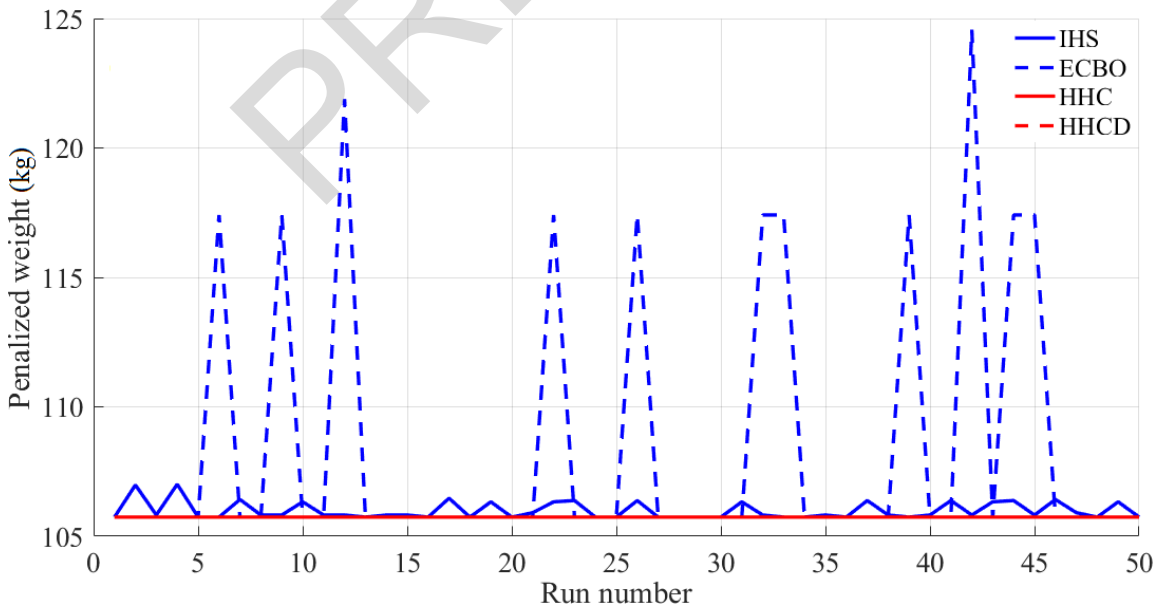


Figure 3.6. Comparison of best designs from 50 runs for the 15-bar truss structure.

Although, HPSO, ECBO and HHC also obtained the best design, HHCD needs fewer structural analyses. Figure 3.6 shows the penalized weight for each of the 50 runs for HIS, ECBO, HHC, and HHCD. In this numerical example, HHCD and HHC were able to find the best solution in every run as shown in Figure 3.6 and Table 3.3.

3.5.3 Planar 52-bar Truss

Figure 3.7 shows the configuration of the 52-bar truss. This example was solved in Lee et al. (2005), Li et al. (2009), and Kaveh and Talatahari (2009). For all members, the modulus of elasticity is 207 GPa and material density is 7860 kg/m³. All members are subjected to stress limitations of 180 MPa in both tension and compression. As shown in Figure 4.5, the structure is subjected to vertical and horizontal point loads at joints 17, 18, 19, and 20, where P_x is 100 kN and P_y is 200 kN. The structure includes 52 members organized into 12 groups (Table 3.4). Design variables are selected from the discrete set of 64 elements:

$$\begin{aligned}
 D = & [71.613, 90.968, 126.451, 161.290, 198.064, 252.258, 285.161, 363.225, \\
 & 388.386, 494.193, 506.451, 641.289, 645.160, 792.256, 816.773, 939.998, \\
 & 1008.385, 1045.159, 1161.288, 1283.868, 1374.191, 1535.481, 1690.319, \\
 & 1696.771, 1858.061, 1890.319, 1993.544, 2019.351, 2180.641, 2238.705, \\
 & 2290.318, 2341.931, 2477.414, 2496.769, 2503.221, 2696.769, 2722.575, \\
 & 2896.768, 2961.284, 3096.768, 3206.445, 3303.219, 3703.218, 4658.055, \\
 & 5141.925, 5503.215, 5999.988, 6999.986, 7419.340, 8709.660, 8967.724, \\
 & 9161.272, 9999.980, 10322.560, 10903.204, 12129.008, 12838.684, \\
 & 14193.520, 14774.164, 15806.420, 17096.740, 18064.480, 19354.800, \\
 & 21612.860] \text{ (mm}^2\text{)}
 \end{aligned} \tag{3.40}$$

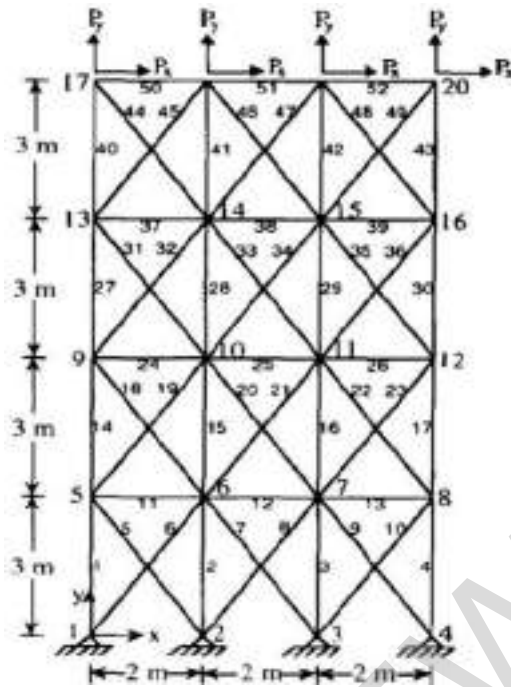


Figure 3.7. Schematic of 52-bar space truss.

Table 3.4. Comparison of optimal designs for 52-bar truss problem.

Design variable (mm ²)		HS (Lee et al., 2005)	HPSO (Li et al., 2009)	DHPSACO (Kaveh and Talatahari, 2009)	This work			
					IHS	ECBO	HHC	HHCD
1	A1-A4	4658.055	4658.055	4658.005	4658.055	4658.055	4658.055	4658.055
2	A5-A10	1161.288	1161.288	1161.288	1161.288	116.1288	116.1288	116.1288
3	A11-A13	506.451	363.225	494.193	494.193	494.193	494.193	494.193
4	A14-A17	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219
5	A18-A23	940	940	1008.385	939.998	939.998	939.998	939.998
6	A24-A30	494.193	494.193	285.161	494.193	494.193	494.193	494.193
7	A31-A34	2290.318	2238.705	2290.318	2238.705	2238.705	2238.705	2238.705
8	A35-A36	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385
9	A37-A39	2290.318	388.386	388.386	363.225	494.193	494.193	494.193
10	A40-A43	1535.481	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868
11	A44-A49	1045.159	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
12	A50-A52	506.451	792.256	506.451	641.289	494.193	494.193	494.193
Best weight (kg)		1906.76	1905.495	1904.83	1903.366	1902.605	1902.605	1902.605
NSA		19104	150,000	5300	20724	33360	36988	32366
G (Eq. 5)		0.027	0.00	0.003	0.00	0.00	0.00	0.00
Weight (kg)	Average	N/A	N/A	N/A	1997.191	1980.129	1935.136	1915.922
	SD ^b	N/A	N/A	N/A	77.96	123.773	55.734	16.950
NSA	Average	N/A	N/A	N/A	21719	37584	36825	36526
	SD	N/A	N/A	N/A	7198	5457	6277	4854

In this study case, ψ is 10 (in Eq. (3.4)) because algorithms found some infeasible solutions with very small violation when ψ is 1. Table 3.4 shows that HHCD has the lowest average and standard division of NSA and the best average and standard division of final designs from 50 runs.

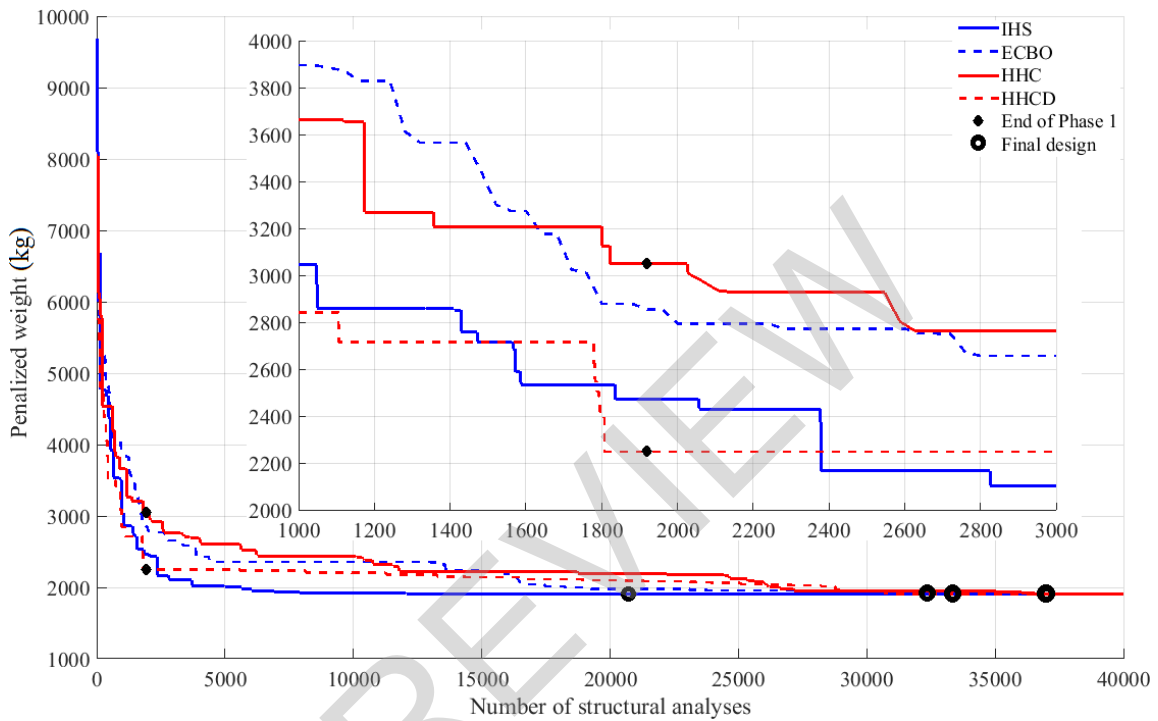


Figure 3.8. Comparison of best designs from 50 runs for the 52-bar truss structure.

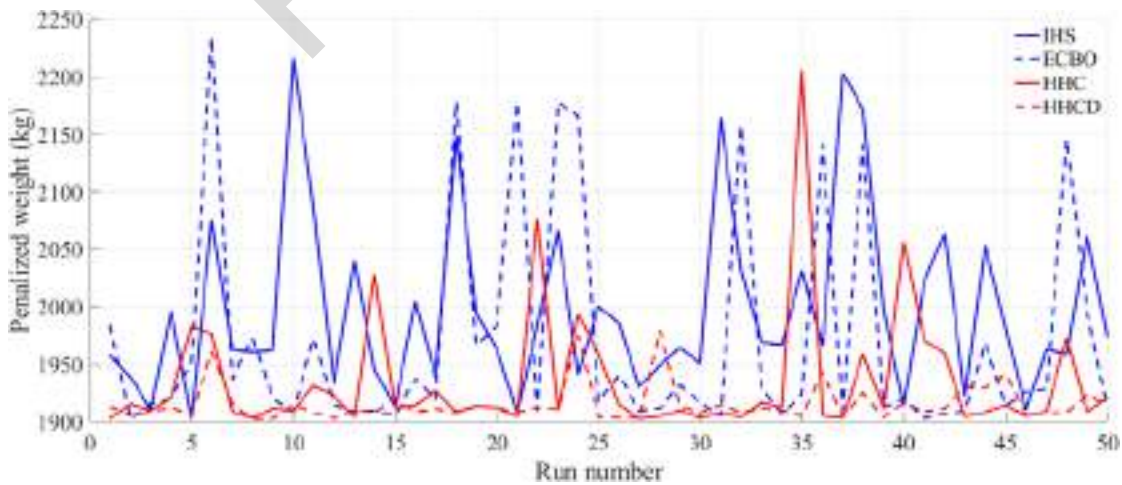


Figure 3.9. Comparison of best designs from 50 runs for the 52-bar truss structure.

3.5.4 Spatial 25-bar Truss

Figure 3.10 shows the configuration of the spatial 25-bar truss. This example was solved in Rajeev and Krishnamoorthy (1992), Lee et al. (2005), Li et al. (2009), Xiang et al. (2009), Camp (2009), and Kaveh and Mahdavi (2015). For all members, the modulus of elasticity is 10,000 ksi and material density is 0.1 lb/in³. The allowable displacement for all nodes in both vertical and horizontal directions is ± 0.35 in. All members are subjected to stress limitations of 40 ksi for both tension and compression. This spatial truss was subjected to the two loading conditions shown in Table 3.5. The structure includes 25 members organized into 8 groups as given in Table 3.6. Design variables are selected from the discrete set of 30 elements as follows:

$$D = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.2, 3.4] \text{ (in}^2\text{)} \quad (3.41)$$

Table 3.5. Load conditions of the spatial 25-bar truss.

Case	Load condition	Nodes	Loads (kips)		
			P _x	P _y	P _z
1	1	1	1.0	-10.0	-10.0
		2	0.0	-10.0	-10.0
		3	0.5	0.0	0.0
		6	0.6	0.0	0.0
2	1	1	0.0	20.0	-5.0
		2	0.0	-20.0	-5.0
	2	1	1.0	10.0	-5.0
		2	0.0	10.0	-5.0
	2	3	0.5	0.0	0.0
		6	0.5	0.0	0.0

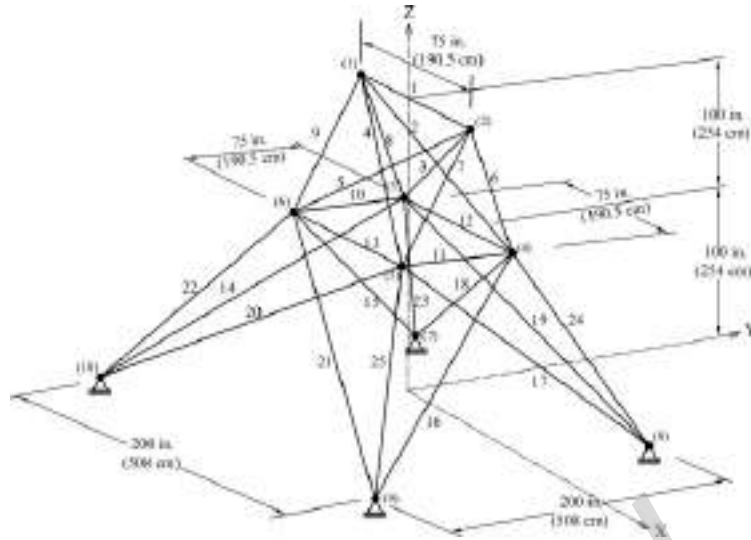


Figure 3.10. Schematic of 25-bar space truss.

In this study case, ψ was 10 (in Eq. (3.4)) because algorithms found some infeasible solutions with very small violation when ψ was 1. HHCD was able to obtain the best design after 2043 structural analyses (759 iterations) as shown in Figure 3.11 and Table 3.6. Figure 3.11 shows the convergence history of the best run of IHS, ECBO, HHC, and HHCD. It shows that HHCD and HHC convergence faster than ECBO to the best design where IHS did not obtain the best design. Table 3.6 shows that although HS, HPSO, SA, BB-BC and ECBO obtained the best design, both HHC and HHCD need fewer structural analyses. Also, Table 3.6 explains that HHCD has the lowest average and standard division of NSA and the best average and standard division of final designs from 50 runs while HHC is close second. Although, BB-BC shows slightly better stability than HHCD, it needs more structural analyses compared to HHCD to obtain the best design. GA did not reach the final design.

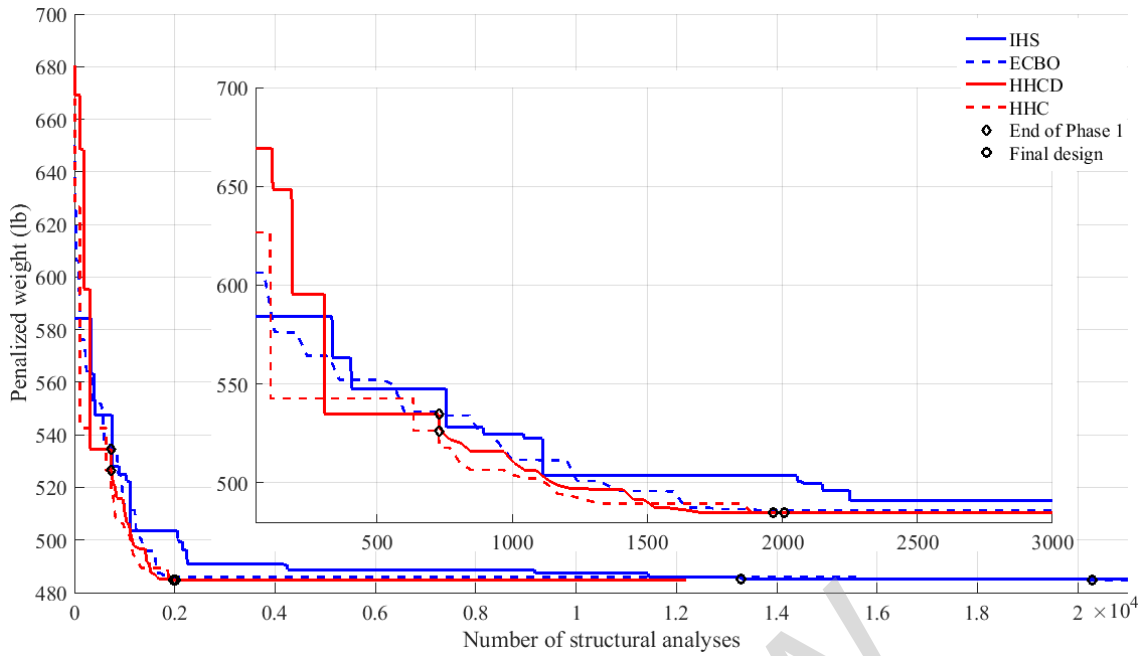


Figure 3.11. Comparison of best designs from 50 runs for the 25-bar truss structure.

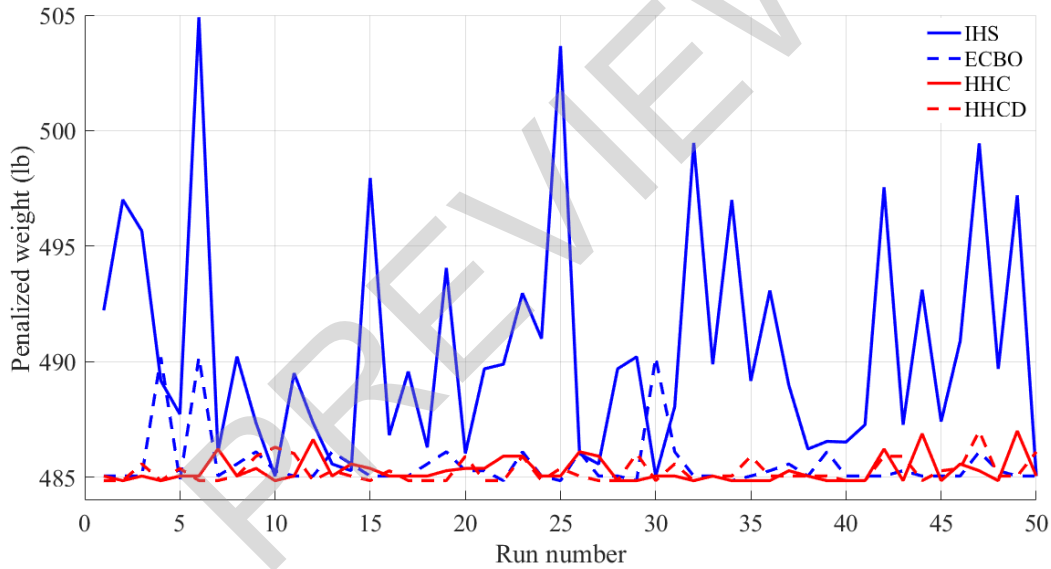


Figure 3.12. Comparison of best designs from 50 runs for the 25-bar truss structure.

Figure 3.12 shows the penalized weight for each of the 50 runs for IHS, ECBO, HHC, and HHCD. It shows that IHS did not reach the final design in any run. HHCD reached the final design more than HHC and ECBO.

Table 3.6. Performance comparison for the 25-bar spatial truss.

Design variable (in2)	GA (Rajeev and Krishn- amoory, 1992)	HS (Lee, et al., 2005)	HPSO (Li et al., 2009)	SA (Xiang et al., 2009)	BB-BC (Camp, 2009)	ECBO (Kaveh and Mahdavi, 2015)	This work				
							IHS ^e	ECBO ^f	HHC	HHCD	
1 A1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2 A2-A5	1.8	0.3	0.3	0.3	0.3	0.3	0.5	0.3	0.3	0.3	0.3
3 A6-A9	2.3	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
4 A10-A11	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5 A12-A13	0.1	2.1	2.1	2.1	2.1	2.1	1.9	2.1	2.1	2.1	2.1
6 A14-A17	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7 A18-A21	1.8	0.5	0.5	0.5	0.5	0.5	0.4	0.5	0.5	0.5	0.5
8 A22-A25	3.0	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
Best weight (lb)	546.013	484.854	484.854	484.854	484.854	484.854	485.054	484.854	484.854	484.854	484.854
NSA	800	13523	25000	7900	9090	61200 ^a	13368	20280	2083	2043	
G (Eq. 5)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Weight (lb)	Average	N/A	N/A	N/A	486.354 ^b	485.10 ^c	485.89 ^d	490.547	485.575	485.480	485.252
	SD	N/A	N/A	N/A	N/A	0.44	N/A	4.986	1.244	0.850	0.505
NSA	Average	N/A	N/A	N/A	N/A	N/A	N/A	23791	17694	8288	7045
	SD	N/A	N/A	N/A	N/A	N/A	N/A	12039	10876	4194	3233

^aPopulation size is 30. ^bThe average of 12 runs. ^cThe average of 100 runs. ^dThe average of 20 runs.

^eThe maximum number of iterations is 50000. ^fThe maximum number of iterations is 1000.

3.5.5 Spatial 72-bar Truss

Figure 3.13 shows the configuration of the spatial 72-bar truss. This example was solved in Li et al. (2009), Kaveh and Khayat (2012), and Kaveh and Mahdavi (2015). For all members, the modulus of elasticity is 10000 ksi and material density is 0.1 lb/in³. The allowable displacement for all nodes in both vertical and horizontal directions equals ± 0.25 in. All members are subjected to stress limitations of 25 ksi for both tension and compression. This spatial truss was subjected to the two loading conditions as shown in Table 3.7. The structure includes 72 members organized into 16 groups (Table 3.8). Design variables are selected from the discrete set of 64 elements as follows:

$$D=[0.111, 0.141, 0.196, 0.250, 0.307, 0.391, 0.442, 0.563, 0.602, 0.766, 0.785, 0.994, 1.000, 1.228, 1.266, 1.457, 1.563, 1.620, 1.800, 1.990, 2.130, 2.380, 2.620, 2.630, 2.880, 2.930, 3.090, 3.13, 3.380, 3.470, 3.550, 3.630, 3.840, 3.870, 3.880, 4.180, 4.220, 4.490, 4.590, 4.800, 4.970, 5.120, 5.740, 7.220, 7.970, 8.530, 9.300, \quad (3.42)$$

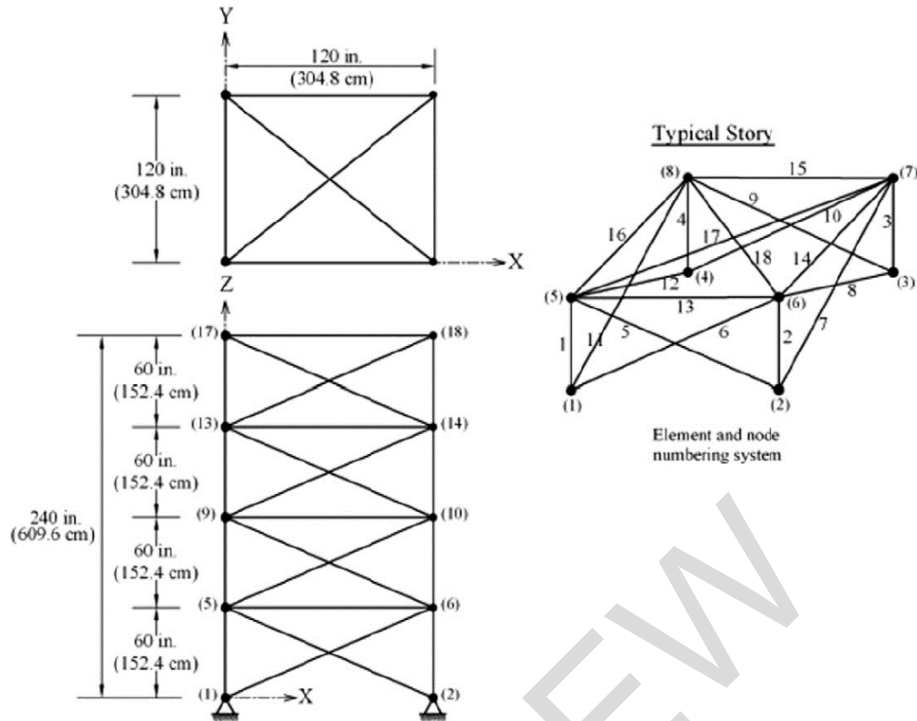


Figure 3.13. Schematic of 72-bar space truss.

10.850, 11.500, 13.500, 13.900, 14.200, 15.500, 16.000, 16.900, 18.800, 19.900, 22.000, 22.900, 24.500, 26.500, 28.000, 30.000, 33.500] (in²).

Table 3.7. Load conditions of the spatial 72-bar truss.

Case	Nodes	Loads (kips)		
		P _x	P _y	P _z
1	17	5.0	5.0	-5.0
	18	0.0	0.0	-5.0
2	18	0.0	0.0	-5.0
	19	0.0	0.0	-5.0
	20	0.0	0.0	-5.0

Table 3.8 provides a comparison between some best designs reported in the literature along with those obtained in this study. HHCD obtained best design after 20836 structural analyses (3017 iteration). The best structural weight of 389.334 lb was obtained by IRO, ECBO, and HHC after 17925, 95400 (35100 in this study), 26476 structural analyses, respectively. It is seen that both HHC and HHCD performed better than IHS and ECBO. Note that in ECBO (this work), HHC and HHCD, design variables 2 and 6 values are 0.442 and 0.563, respectively, whereas in IRO and ECBO (Kaveh and Mahdavi, 2015), they are 0.563 and 0.442. In this study case, HHCD needs

more analyses than IRO; however, HHCD has smaller average value of 50 independent runs (see Table 3.8).

Figure 3.14 shows the convergence history of IHS, ECBO, HHC, and HHCD. It shows at HHCD convergences faster than IHS, ECBO, and HHC. Figure 3.15 and Table 3.8 demonstrate that HHCD is more stable than IRO, CBO, ECBO, IHS, and HHC in term of the quality of final designs. For a similar quality of design other algorithms needed more simulations (NSA averages in Table 3.8).

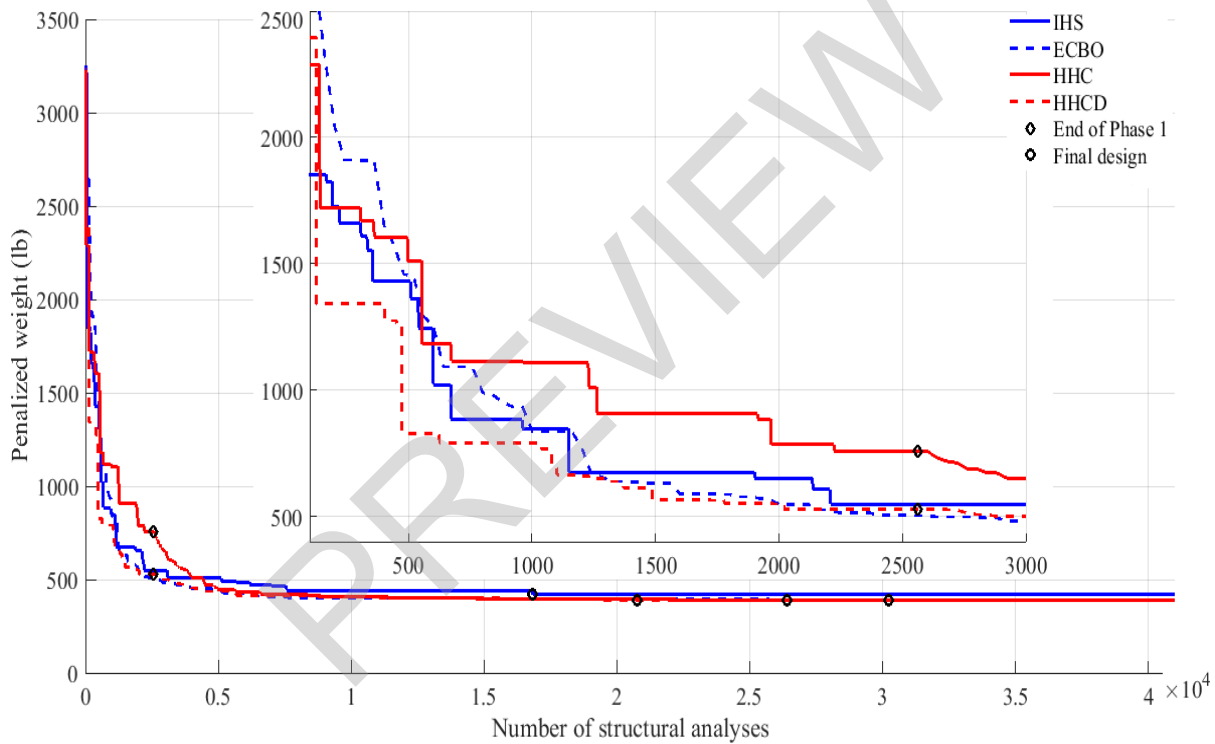


Figure 3.14. Comparison of convergence rates for 72-bar spatial truss.

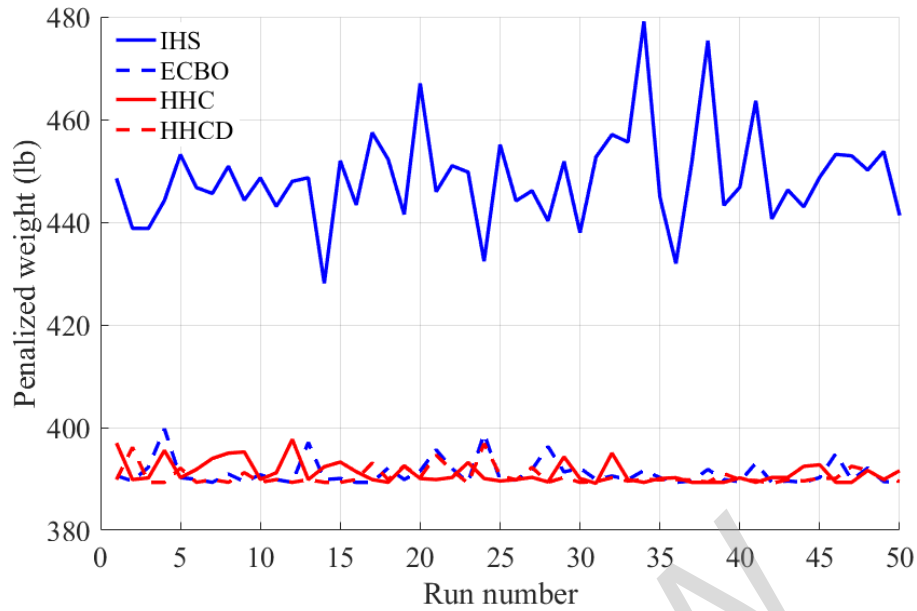


Figure 3.15. Comparison of best designs from 50 runs for the 72-bar truss structure.

Table 3.8. Performance comparison for the 72-bar spatial truss.

Design variable (in2)		HPSO (Li, 2009)	IRO (Kaveh et al., 2013)	CBO (Kaveh and Mahdavi, 2015)	ECBO (Kaveh and Mahdavi, 2015)	This work			
						IHS ^c	ECBO ^d	HHC	HHCD
1	A1-A4	4.970	1.990	2.130	1.990	2.62	1.990	1.990	1.990
2	A5-A12	1.228	0.563	0.563	0.563	0.442	0.442	0.442	0.442
3	A13-A16	0.111	0.111	0.111	0.111	0.141	0.111	0.111	0.111
4	A17-A18	0.111	0.111	0.111	0.111	0.141	0.111	0.111	0.111
5	A19-A22	2.880	1.228	1.228	1.228	1.457	1.228	1.228	1.228
6	A23-A30	1.457	0.442	0.442	0.442	0.563	0.563	0.563	0.563
7	A31-A34	0.141	0.111	0.141	0.111	0.141	0.111	0.111	0.111
8	A35-A36	0.111	0.111	0.111	0.111	0.196	0.111	0.111	0.111
9	A37-A40	1.563	0.563	0.442	0.563	0.442	0.563	0.563	0.563
10	A41-A48	1.228	0.563	0.563	0.563	0.602	0.563	0.563	0.563
11	A49-A52	0.111	0.111	0.111	0.111	0.141	0.111	0.111	0.111
12	A53-A54	0.196	0.111	0.111	0.111	0.141	0.111	0.111	0.111
13	A55-A58	0.391	0.196	0.196	0.196	0.25	0.196	0.196	0.196
14	A59-A66	1.457	0.563	0.563	0.563	0.563	0.563	0.563	0.563
15	A67-A70	0.766	0.391	0.391	0.391	0.442	0.391	0.391	0.391
16	A71-A72	1.563	0.563	0.563	0.563	0.391	0.563	0.563	0.563
Best weight (lb)		933.094	389.334	391.230	389.334	418.380	389.334	389.334	389.334
NSA		50000	17925	160200	95400 ^a	16918	30240	26476	20836
G (Eq. 5)		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Weight (lb)	Average	N/A	408.17	456.69	391.59 ^b	448.554	391.173	391.242	390.632
	SD	N/A	N/A	N/A	N/A	9.412	2.073	2.105	1.679
NSA	Average	N/A	N/A	N/A	N/A	14135	37531	32208	27442
	SD	N/A	N/A	N/A	N/A	6037	8858	7994	6884

^aPopulation size is 30. ^bThe average of 20 runs. ^cThe maximum number of iterations is 50000.

^dThe maximum number of iterations is 1000.

3.6 Discussion and Conclusions

A new two-phase metaheuristic optimization algorithm was presented in this study. Phase 1 used Improved Harmony Search (IHS) with a new domain reduction technique that used statistical analysis of some of the better designs in the current population. Phase 2 used the Enhanced Colliding Bodies Optimization (ECBO) where the initial population consisted of some of the better designs from Phase 1. With this better initial population, ECBO obtained the best design more efficiently. Also, in Phase 1, an improved stopping criterion was proposed that terminated the phase when there was no or small improvement in the best design after many iterations.

Table 3.9. Comparative data for design examples.

Design Example			Optimization algorithm			
			IHS	ECBO	HHC	HHCD
Planar 10-bar truss	Best weight (lb)		5586.289	5490.738	5490.738	5490.738
	Weight (lb)	Average	5680.406	5519.357	5499.116	5490.873
		SD	40.582	53.183	30.732	0.943
	NSA	Average	6999	19378	9821	8979
SD		2728	6215	5038	3890	
Planar 15-bar truss	Best weight (lb)		105.735	105.735	105.735	105.735
	Weight (kg)	Average	105.993	108.537	105.735	105.735
		SD	0.341	5.441	0	0
	NSA	Average	15734	4169	3986	3624
SD		4781	790	757	764	
Planar 52-bar truss	Best weight (lb)		1903.366	1902.605	1902.605	1902.605
	Weight (kg)	Average	1997.191	1980.129	1935.136	1915.922
		SD	77.96	123.773	55.734	16.95
	NSA	Average	21719	37584	36825	36526
SD		7198	5457	6277	4854	
25-bar spatial truss	Best weight (lb)		485.054	484.854	484.854	484.854
	Weight (lb)	Average	490.547	485.575	485.480	485.252
		SD	4.986	1.244	0.850	0.505
	NSA	Average	23791	17694	8288	7045
SD		12039	10876	4194	3233	
72-bar spatial truss	Best weight (lb)		418.380	389.334	389.334	389.334
	Weight (lb)	Average	448.554	391.173	391.242	390.632
		SD	9.412	2.073	2.105	1.679
	NSA	Average	14135	37531	32208	27442
SD		6037	8858	7994	6884	

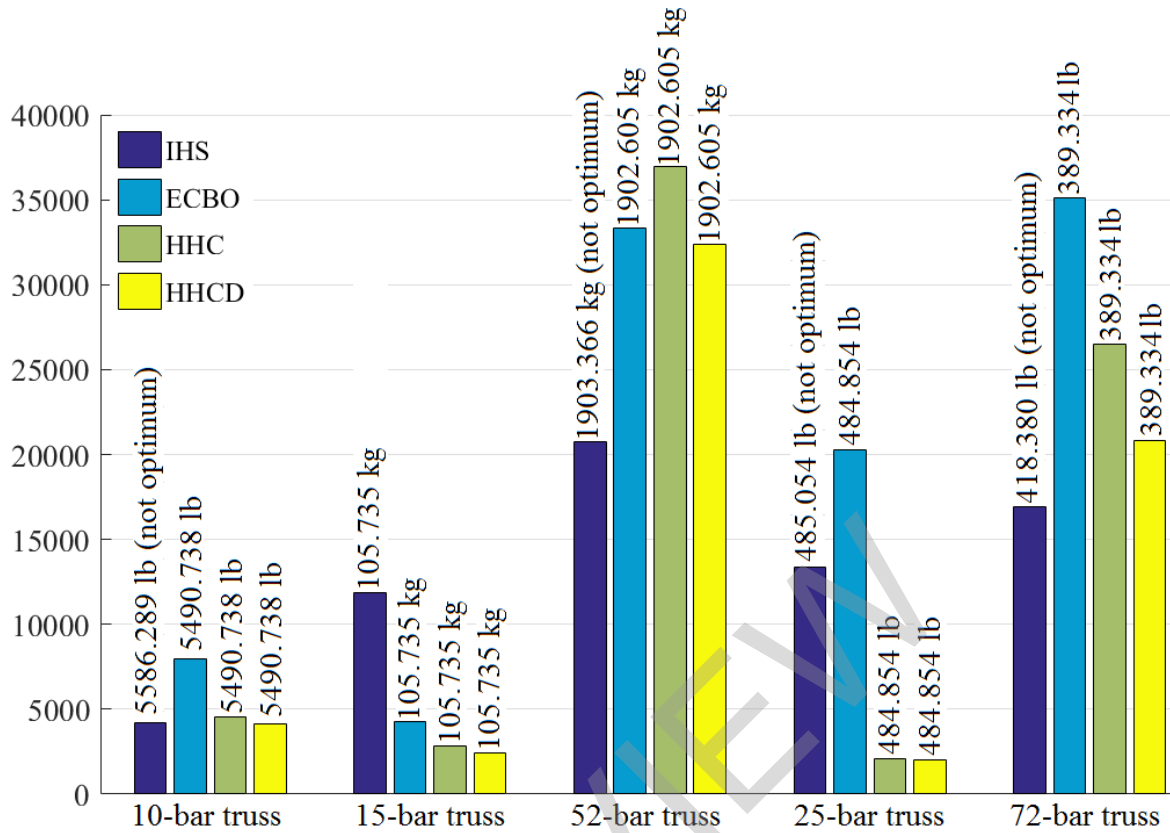


Figure 3.16. Comparison of number of structural analyses to reach the best.

Detailed results for three standard truss test structures were presented and discussed. Table 3.9 summarize comparative data obtained with IHS, ECBO, HHC and HHCD for the three design examples. It shows, in term of the quality of the solution, HHCD obtained the best designs with the lowest averages and standard deviations from 50 independent runs. HHC is close second behind HHCD. Figure 3.16 is a bar chart representation of the number of structural analyses needed to reach final designs of the three numerical examples with the four methods. The best structural weight values are also shown with bar charts. It shows that IHS does not reach the best design for any of the examples. ECBO needs the largest number of analyses to obtain the final designs. However, HHC and HHCD need a smaller number of simulations to reach the final designs. Table 3.9 and Figure 3.16 show that HHCD is quite stable and more efficient among all metaheuristic algorithms that are discussed in this study. For the 3-D truss problems (the last two examples),

HHCD shows an outstanding performance in terms of the number of structural analyses needed to obtain the best design. This is an attractive feature of the proposed metaheuristic algorithm with domain adjustment.

Based on the comparison with other metaheuristic optimization algorithms for the numerical examples, the following conclusions are drawn:

- 1- The 50 independent runs for each example showed that the proposed HHC algorithm was quite reliable in obtaining the best designs for each run. Also, HHCD had the lowest averages and standard deviations for the final cost function values. This implies that fewer runs are needed to obtain the best design compared to many other stochastic algorithms.
- 2- The proposed domain adjustment approach worked very well with IHS.
- 3- The proposed hybrid algorithm with domain adjustment was able to find the best design with fewer structural analyses, by substantial amount in some cases. This efficiency is critically important for solving more complex applications, such as nonlinear structural analysis problems, dynamic response optimization problems and multidisciplinary optimization problems.

3.7 Reproducing Results

To reproduce results provided in this work, all the necessary information about design examples are described in Section 3.5. Appendix A includes MATLAB code for the 10-bar truss design example.

CHAPTER 4

DISCRETE VARIABLE OPTIMIZATION OF STRUCTURES SUBJECTED TO DYNAMIC LOADS USING EQUIVALENT STATIC LOADS AND METAHEURISTIC ALGORITHMS

Abstract

Equivalent static loads (ESL) approach has been used successfully for optimizing many structural systems subjected to dynamic loads. The approach has been used for continuous variable optimization problems using the gradient-based methods. It has been shown that the approach drastically reduces the number of dynamic analyses of the structure to reach a local optimum point. In this chapter, the ESL approach is investigated for optimization of structures with discrete design variables using metaheuristic algorithms. The focus is on a class of problems that cannot be solved using the gradient-based optimization methods. It is shown that for this class of problems, the ESL approach reaches near the best design with a drastically reduced number of dynamic analyses of the structure. However, it cannot converge to the best design because the ESLs calculated for a member of the population are not suitable for the remaining members of the population in metaheuristic algorithms. Moreover, the assumption of small change to design variables near the solution point does not hold in metaheuristic algorithms. Therefore, after a few ESL cycles, the procedure may switch to a full dynamic analysis of each member of the population, if desired, to further improve designs and reach the best design. Overall, better results are obtained by incorporating the ESL approach and the number of dynamic analyses is substantially reduced to solve this class of discrete variable optimization problems.

4.1 Introduction

It is important to consider transient dynamic loads in the design process of many structures in engineering applications since many loads in the real-world act dynamically. At the same time,

it is important to consider minimizing the total cost while achieving all the safety and performance requirements by optimizing the design of structures (Arora, 1999).

Optimization of structures subjected to dynamic loads using gradient-based algorithms includes calculating the gradients of all the problem functions provided the functions are differentiable. Several methods can be used to calculate the gradients such as: direct method, the adjoint method, the modal approximation method (Kang et al., 2006). Then a gradient-based optimization algorithm can be used to determine the design improvement by solving a subproblem. This process involves the integration of the equations of motion and sensitivity equations. Numerical integration of these equations is computationally expensive. Moreover, for some problem with material and/or geometrical nonlinearities, the numerical integration methods can have convergence difficulties. Therefore, it can be difficult to optimize structures subjected to dynamic loads in a mathematical optimization process (Kang et al., 2001). To overcome these difficulties, efforts have been made to transform the dynamic load into static loads.

One of the well-known dynamic to static loads transformation methods is based on the displacement field obtained using dynamic analysis of the structure (Kang et al., 2001). That is, the dynamic load is transformed into multiple equivalent static load sets. Then the equivalent static loads (ESL) are considered as multiple loading conditions in the linear static response optimization process. This is called an ESL cycle of the optimization process. These cycles are repeated until the final design is obtained. More details of this process are provided in Section 4.3.

Calculus-based local optimization algorithms are applicable for continuous variables and differentiable functions. To solve a differentiable problem with discrete variables, many gradient-based optimization strategies are available (Arora, 2017). One strategy is to initially treat the discrete variables as continuous (if possible) and then round-off their values at the optimum point

to get their discrete values. With such an approach, the final solution may be infeasible or far from the true optimum. Moreover, for some engineering problems with discrete variables, it is not possible to compute gradient information because the problem functions are not differentiable. Frame design optimization examples presented and discussed later in Sections 4.7.3 and 4.7.4 are a class of problems where gradient-based optimization methods are not applicable.

Stochastic, metaheuristic or nature-inspired algorithms based on simulations do not require gradient information, such as the well-known Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Harmony Search (HS), and many others. In these algorithms, the search is not limited to a neighborhood of the current point, and the discrete variables and nondifferentiable functions can be treated routinely. They use random search in the whole design space instead of gradient-based search in a neighborhood of the current point (this is why they are sometimes called global optimization methods (Weise, 2009)). Therefore, they are applicable for both continuous and discrete variables and with one or more, simple or complex objective functions. Also they tend to converge to a global minimum (although there is no guarantee of this) for the problem instead of a local minimum as with the gradient-based methods. Since only the structural response is required in the optimization process, these methods can handle any kind of problems (linear, nonlinear, static, dynamic, differentiable, nondifferentiable). Similar to gradient-based optimization method, the computation cost of linear or nonlinear dynamic analysis is more than that for linear static analysis. Therefore, using metaheuristic algorithms could be impractical for dynamic response optimization problems since they generally require many structural analyses to reach the final design.

4.2 Objective of This Work

In this study, the ESL approach for structures subjected to dynamic loads is investigated with metaheuristic optimization algorithms and discrete design variables. This has not been investigated before in the literature. Also, the problem functions are assumed to be nondifferentiable which is the case with some practical problems as discussed in Sections 4.7.3 and 4.7.4. The idea is to study if the number of transient structural analyses required to reach the best design can be reduced compared to those with a standard metaheuristic algorithm. The method is named global optimization with equivalent static loads (GOESL). That is, the dynamic load for linear or nonlinear transient problems will be transformed into multiple equivalent static load sets using the ESL approach. Then the linear static problem will be optimized using a metaheuristic optimization algorithm. These ESL cycles will be repeated until the best design is reached. Enhanced Colliding Bodies Optimization (ECBO) algorithm will be used as the metaheuristic algorithm, although any other such algorithm may also be used.

ESL method with gradient-based optimization obtains one solution at the end of an ESL cycle. That solution is used to generate new ESLs for the next cycle. Metaheuristic algorithms, however, deal with a population of designs. Therefore, at the end of an ESL cycle, there is a population of designs that has been improved based on linear static analysis process. For the next cycle, only one design should be used to generate new ESLs. The question is which one? There are several possibilities for this. Three approaches are examined to select the design that is used to generate the ESLs for the next cycle (Section 4.7). Example problems are solved to evaluate these approaches and, in general, the ESL approach with metaheuristic algorithms. It is important to note that since the focus of this work is on discrete variable problems with nondifferentiable functions, example problems will not be solved or compared with the gradient-based methods.

4.3 Transformation of Dynamic Loads into Equivalent Static Loads (ESLs)

Dynamic analysis is needed when the magnitude of the loads changes with time. In this section, we describe the basic concepts and steps of the *ESL method* for continuous design variables using the gradient-based optimization algorithms (Kang, Choi & Park, 2001). The dynamic response of a structure subjected to dynamic load is described by the following differential equation obtained after a finite element model for the structure has been developed:

$$\mathbf{M}(\mathbf{X})\ddot{\mathbf{u}}(t) + \mathbf{C}(\mathbf{X})\dot{\mathbf{u}}(t) + \mathbf{K}(\mathbf{X}, \mathbf{u}(t))\mathbf{u}(t) = \mathbf{p}(t); t = t_1, t_2, \dots, t_n \quad (4.1)$$

where \mathbf{M} is the mass matrix, \mathbf{K} is the stiffness matrix (\mathbf{K} is a function of the design variables and displacement vector for nonlinear dynamic analysis and just the design variables for linear dynamic analysis), \mathbf{C} is the damping matrix, \mathbf{u} is the dynamic displacements vector, $\dot{\mathbf{u}}$ is the velocities vector, $\ddot{\mathbf{u}}$ is the accelerations vector, \mathbf{X} is the vector of design variables, $\mathbf{p}(t)$ is the applied load vector, t is time (generally discretized for numerical integration), and n is the total number of the time steps.

The static analysis with the finite element method is described by the following equation:

$$\mathbf{K}(\mathbf{X})\mathbf{z} = \mathbf{p}_s \quad (4.2)$$

where \mathbf{z} is the static displacement vector and \mathbf{p}_s is the external static load vector. ESLs are static loads that generate the same displacement field as from dynamic loads at a given design \mathbf{X} . Using Eq. (4.2), an ESL vector at an arbitrary time (t_α) is calculated as follows:

$$\mathbf{p}_\alpha = \mathbf{K}(\mathbf{X})\mathbf{u}(t_\alpha); \quad \alpha = 1, 2, \dots, n \quad (4.3)$$

Figure 4.1 describes the concept of ESL approach. That is, after linear or nonlinear dynamic analysis of the structure, an equivalent load vector (\mathbf{p}_α) is generated at each time step using Eq. (4.3). It is seen that for a given design \mathbf{X} , the linear static response from the α th load set (\mathbf{p}_α) is

the same as the dynamic response at the α th time step. Therefore, the displacement profile of the dynamic response is exactly the same as the displacement profile calculated from the linear static analysis (Kim & Park, 2010). However, the profile of the ESLs is quite different from those of the dynamic loads because the ESLs are applied at each degree of freedom of the model even if the dynamic load is applied along only one degree of freedom. After the design is changed during the optimization iterations, the static and dynamic displacement profiles would be different because the ESLs are based on the starting design.

Optimum design of structures subjected to dynamic loads using the ESLs proceeds as follows (this will be called the *ESL method*):

Step 1. Select an initial design for the structure. Perform dynamic analysis of the structure to generate the displacement profile $\mathbf{u}(t)$ using Eq. (4.1).

Step 2. Calculate the ESLs using Eq. (4.3).

Step 3. Perform static response optimization of the structure using ESLs calculated in Step 2. These loads are kept fixed during this optimization process. This is called an *ESL cycle* of the ESL method.

Step 4. Check the stopping criteria; if satisfied stop; otherwise continue.

Step 5. Since the final design from Step 3 is different from the starting design, the static displacements will be different from dynamic displacements for the final design. Therefore, perform the dynamic analysis of the structure and go to Step 2.

After a few cycles of the above process, the design changes are quite small such that the ESLs do not change much and a solution to the original dynamic response optimization problem is achieved.

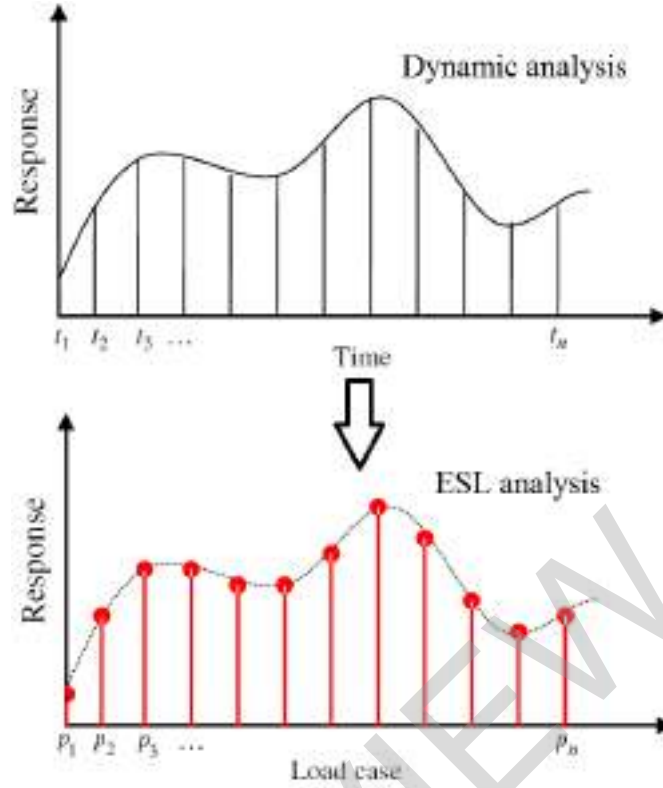


Figure 4.1. Dynamic response vs ESL response for a given design (Kim & Park, 2010).

4.4 Formulation for Discrete Structural Optimization Problems

In many practical design cases, design variables are discrete because members must be selected from the available sizes in a catalog. The formulation of the discrete design variables optimization problem is different from the continuous design variables optimization. In general, the nonlinear dynamic response optimization problem with discrete design variables can be stated as:

$$\text{Find } \mathbf{X} = [x_1, x_2, \dots, x_{nvar}]; \quad x_i \in D_i; \quad i = 1, 2, \dots, nvar \quad (4.4)$$

$$\text{to minimize } f(\mathbf{X}) \quad (4.5)$$

$$\text{subject to } \mathbf{M}(\mathbf{X})\ddot{\mathbf{u}}(t) + \mathbf{C}(\mathbf{X})\dot{\mathbf{u}}(t) + \mathbf{K}(\mathbf{X}, \mathbf{u}(t))\mathbf{u}(t) = \mathbf{p}(t) \quad (4.6)$$

$$g_k(\mathbf{X}, \mathbf{u}(t), \dot{\mathbf{u}}(t), \ddot{\mathbf{u}}(t), t) \leq 0; \quad \text{for all } t \text{ and } k = 1, 2, \dots, l$$

where \mathbf{X} is the vector of design variables with $nvar$ unknowns, D_i is a set of discrete values for the i th design variable, $f(\mathbf{X})$ is a cost function (in this study, $f(\mathbf{X})$ is the total weight of the structure), and g_k is a constraint function that needs to be imposed at all time points.

One way of treating constraints in metaheuristic algorithms is to combine constraints with the cost function to define a merit function (also called the penalty function) $F(\mathbf{X})$ that is then minimized:

$$F(\mathbf{X}) = f(\mathbf{X})[1 + \psi G(\mathbf{X})]^\xi \quad (4.7)$$

$$G(\mathbf{X}) = \sum_{i=1}^n \sum_{k=1}^l \max(0, g_k(t_i)) \quad (4.8)$$

where $G(\mathbf{X})$ is a constraint violation function, $\psi \geq 1$ is exploration penalty coefficient (in this study, $\psi = 1$), $\xi > 1$ is penalty function exponent (in this study, $\xi = 2$), and $\max(0, g_k(t_i)) \geq 0$ is the violation value of the k th inequality constraint at the time point t_i . The present problem has just inequality constraints. However, if equality constraints are present in the problem formulation, they are treated by including their violations in Eq. (4.8). The linear dynamic response problem is the same as the nonlinear dynamic response problem except that \mathbf{K} is not a function of the displacement vector \mathbf{u} .

The linear static response optimization problem subjected to ESLs can be stated as:

$$\text{Find } \mathbf{X} = [x_1, x_2, \dots, x_{nvar}]; \quad x_i \in D_i; \quad i = 1, 2, \dots, nvar \quad (4.9)$$

$$\text{to minimize } f(\mathbf{X}) \quad (4.10)$$

$$\text{subject to } \mathbf{K}(\mathbf{X})\mathbf{u}_\alpha = \mathbf{p}_\alpha \quad (4.11)$$

$$g_{k\alpha}(\mathbf{X}) \leq 0; \quad k = 1, 2, \dots, l; \quad \alpha = 1, 2, \dots, n$$

For this problem, the merit function $F(\mathbf{X})$ for the metaheuristic algorithms is defined as:

$$F(\mathbf{X}) = f(\mathbf{X})[1 + \psi G]^\xi \quad (4.12)$$

$$G(\mathbf{X}) = \sum_{\alpha=1}^n \sum_{k=1}^p \max(0, g_{k\alpha}) \quad (4.13)$$

4.5 Enhanced Colliding Bodies Optimization (ECBO)

Kaveh and Mahdavi (2014) developed this metaheuristic algorithm that is inspired by the laws of one-dimensional collision. The algorithm works with a population of designs at each iteration. The initial population is generated randomly, and the designs are stored in a matrix **CB**, called the colliding bodies' matrix. Each design in the population is considered as an object or body having pseudo-mass that is calculated using the merit function value for each design. The entire population is ranked and divided into moving objects and stationary objects. Using the conservation law of linear momentum and the coefficient of restitution, one dimensional collision between the bodies is simulated. Based on that, new velocities of the stationary and moving objects are calculated. Using these velocities and random numbers, each design in the population is updated. This process is repeated until a limit on the iterations is reached or there is very little change in the best design for several iterations.

In the enhanced version of the colliding bodies optimization (ECBO), a colliding memory matrix called **CM** is used to store some good designs. These designs replace the worst designs in the **CB** matrix at every iteration. This way the good designs are always preserved. In addition, a parameter $Pro \in [0,1]$ is introduced that is used along with random numbers to regenerate a component of selected designs in the **CB** matrix. This mechanism is shown to give diversity to the design population leading to a better final design (Kaveh & Ghazaan, 2014).

Many metaheuristic algorithms need selection of several algorithmic parameters in their calculations. This is a major drawback of these algorithms because their performance depends on the values for the parameters. ECBO, however, requires just one algorithmic parameter specification, and performs well in term of the quality of solutions and convergence time. In addition, ECBO has been used to solve truss, frame, and other engineering optimization problems. It has shown very good convergence behavior compared to other metaheuristic algorithms such as genetic algorithm, particle swarm and harmony search (Kaveh & Mahdavi, 2015). Therefore, this metaheuristic algorithm is elected for use in this study.

4.6 Discrete Variable Optimization Using ESL for Transient Problems

As mentioned earlier, metaheuristic optimization algorithms search not only in the neighborhood of the current design point but also in the entire design space. That is, small changes in design variables are not guaranteed which is an important assumption in the ESL method (at least near the local optimum point) with gradient-based optimization (Kang et al., 2001). Also, in ESL method with gradient-based optimization, there is one solution at the end of an ESL cycle. That solution is used to generate new ESLs for the next cycle. In metaheuristic algorithms, however, there is a population of designs at the end of an ESL cycle. Since most metaheuristic optimization algorithms deal with a population of designs, it is not obvious which design should be used to calculate the ESLs for the static response optimization cycle. One choice could be to use the best design from the population based on the merit function value. However, the ESLs calculated for the best design may not be suitable for the remaining designs of the population. Therefore, three approaches are investigated:

1- The best design from static analysis – ESL1.

Design that has the lowest merit function value based on linear static analyses (the best design at the end of an ESL cycle) is used to generate ESLs for the next cycle. In each cycle, only one dynamic analysis is needed in this approach.

2- The best design from dynamic analysis – ESL2.

It was observed that the best design from the ESL cycle (which is based on linear static analyses) may not be the best design when a transient analysis is performed for the final population. Therefore, dynamic analyses are performed for designs in **CM** (4 designs in this study) and the first 25% of **CB** (the first 10 designs in this study). Just the first 25% of **CB** is used instead of the entire population because it is expected that the best design will be in this range. Then design that has the lowest merit function is used to generate ESLs for the next cycle. In each cycle, the number of dynamic analyses is 14 in this approach.

3- The heaviest feasible design from dynamic analysis – ESL3.

This approach is similar to ESL2 except that heaviest feasible design is used to generate ESLs for the next cycle. This design usually generates smaller ESLs values because the heavier structure is usually stiffer giving smaller displacements. If there is no feasible design (which usually happens in the first few cycles), ESL2 is used to generate ESLs for the next cycle. In each cycle, the number of dynamic analyses is 14 in this approach.

In the proposed algorithm, ESL method is used to transform the problem to linear static response optimization problem subjected to load cases that give the same displacement field as for the transient problem for the selected design (Section 4.3). Then the linear static problem is optimized using ECBO. This results in fewer transient structural analyses for the metaheuristic

optimization algorithm to find the best design. Figure 5.2 shows the flowchart of GOESL that is explained as follows:

Step 1. Generation of an initial population.

A population of designs is randomly generated from the design domain and saved in the **CB** matrix.

Step 2. Evaluation of designs in **CB**.

In this step, designs in **CB** are analyzed using a transient solver. Using the simulation results and Eqs. (4.7) and (4.8), the merit function $F(\mathbf{X})$ is calculated for each design. Then the designs are arranged in an ascending order based on their merit function values. The colliding memory matrix **CM** is generated. The best design of the population is used to generate the ESLs; **CB** and **CM** are passed to ECBO with ESL method block in Figure 4.2. Also, two matrices **CB**_{ESL} and **CM**_{ESL} are set to **CB** and **CM**, respectively. At the end of ECBO with ESL method, **CB**_{ESL} and **CM**_{ESL} will be passed to ECBO without the ESL block.

Step 3. Optimum design with the calculated ESLs.

Using linear static analyses of the structure, optimum design is found with the formulation given in Eqs. (4.9) to (4.13). This completes a cycle of the ESL method.

The termination criteria for one ESL cycle are as follows:

$$\text{If}_1 \text{ } Iter_{ESL} \geq 0.25 \times MaxIter_{ESL}$$

$$\text{If}_2 \text{ } (Merit(Iter_{ESL}) - Merit(Iter - 0.1 \times MaxIter_{ESL})) / Merit(Iter_{ESL}) \leq \epsilon$$

Terminate the current cycle

End₂

End₁

$$MaxIter_{ESL} = 0.5 \times MaxIter_{transient} \quad (4.14)$$

$$MaxIter_{transient} = \sum_i^{nvar} N_i \quad (4.15)$$

where $Iter_{ESL}$ is the current iteration, $MaxIter_{ESL}$ is the limit on number of iterations for the ESL cycle, ϵ is a small number (in this study $\epsilon=10^{-3}$), N_i is the number of elements in the discrete set D_i , and $nvar$ is number of design variables. That is, when there is no or small improvement in the current merit function value after several iterations, the current ESL cycle is terminated.

Step 4. Transient analysis of final design(s).

Perform transient analysis of a design or multiple designs depending on ESL1, ESL2, or ESL3 approach used.

Step 5. Updating \mathbf{CB}_{ESL} and \mathbf{CM}_{ESL} .

In this step, \mathbf{CB}_{ESL} and \mathbf{CM}_{ESL} matrices are updated depending on the approach as follows:

1- ESL1: if the transient analysis for the best design from static analysis at the end of an ESL cycle shows this design to be better than the best design in \mathbf{CM}_{ESL} , update \mathbf{CB}_{ESL} and \mathbf{CM}_{ESL} as follows:

$$\begin{aligned} \mathbf{CB}_{ESL} &= \mathbf{CB} \text{ (colliding bodies matrix of the current ESL cycle)} \\ \mathbf{CM}_{ESL} &= \mathbf{CM} \text{ (colliding memory matrix of the current ESL cycle)} \end{aligned} \quad (4.16)$$

2- ESL2: in this approach, the design that has the lowest merit function is used to generate ESLs for the next cycle (as described above). If this design is better than the best design in \mathbf{CM}_{ESL} , update \mathbf{CB}_{ESL} and \mathbf{CM}_{ESL} as follows:

$\mathbf{CB}_{ESL}=\mathbf{CB}$ (colliding bodies matrix of the current ESL cycle)

$\mathbf{CM}_{ESL}=\text{best 4 designs from \{previous } \mathbf{CM}_{ESL} \text{ (4 designs), } \mathbf{CM} \text{ (4 designs) of}$ (4.17)

the current ESL cycle, or 25% of \mathbf{CB} of the current ESL cycle (10 designs)}

3- ESL3: in this approach, the heaviest feasible design is used to generate ESLs for the next cycle (as described above). If the design that has the lowest merit function is better than the best design in \mathbf{CM}_{ESL} , update \mathbf{CB}_{ESL} and \mathbf{CM}_{ESL} as follows:

$\mathbf{CB}_{ESL}=\mathbf{CB}$ (colliding bodies matrix of the current ESL cycle)

$\mathbf{CM}_{ESL}=\text{best 4 designs from \{previous } \mathbf{CM}_{ESL} \text{ (4 designs), } \mathbf{CM} \text{ (4 designs)}$ (4.18)

of the current ESL cycle, or 25% of \mathbf{CB} of the current ESL cycle (10 designs)}

This way, the population that generate the best design (\mathbf{CB}_{ESL}) and the best designs that saved from cycle to cycle (\mathbf{CM}_{ESL}) are passed to ECBO at the end of ESL method.

To terminate the ESL method, the following criterion is used (note that the minimum number of ESL cycles is set to 5): no better design is found for two ESL cycles. The stopping criteria are checked at this stage; if satisfied, the ESL method is terminated and we go to the ECBO block (Step 7) with full transient analyses; otherwise, we continue to Step 6.

Step 6. Initialization for a new ESL cycle.

In this step, new ESLs are re-calculated based on ESL1, ESL2, or ESL3 approach, new population of designs is generated from the design domains in the \mathbf{CB} matrix. This shows better convergence behavior than passing the last \mathbf{CB} to the next cycle because new designs are explored by generating new \mathbf{CB} when the best designs (so far) are preserved by setting $\mathbf{CM}=\mathbf{CM}_{ESL}$. The updated \mathbf{CM}_{ESL} is passed to the next cycle as \mathbf{CM} .

Step 7. ECBO without ESL cycles.

If the stopping criteria for the ESL step are satisfied, \mathbf{CB}_{ESL} matrix and the \mathbf{CM}_{ESL} are passed to ECBO block with full transient analyses. These two matrices have improved designs using ECBO with ESL method. Then, the formulation given in Eqs. (4.4) to (4.8) is used to find the final best design. It was found that with just the ESL cycles, the algorithm could not reach the best design. Therefore, *Step 7* was necessary to further improve the design. It is observed however, that the best design at the end of ESL cycles is usually close to the final best design. Therefore, for practical applications, it may be appropriate to stop the algorithm after the ESL cycles.

The foregoing procedure significantly reduces the number of transient analyses needed to reach the best design compared to the procedure without the use of ESLs (as shown in Section 7). In other words, the time needed to reach the best design is reduced because computation times for linear static analyses (solving system of linear equations) are much shorter than those for the transient analyses (numerically solving system of differential equations). For large problems, one transient analysis might require hours which makes the metaheuristic optimization algorithms very time consuming.

In nonlinear dynamic problems, ESLs generate the same displacements as those from nonlinear dynamic analysis; however, they do not generate the same stress responses because of the nonlinear relationship between stress and strain and strain and displacement (Kim & Park, 2010). Therefore, when there are stress constraints, the difference in stresses can be adjusted to $\bar{\sigma}_{L\alpha}$ as follows:

$$\beta_{\alpha,i} = \frac{\sigma_{N\alpha,i}}{\sigma_{L\alpha,i}} \quad (4.19)$$

$$\bar{\sigma}_{L\alpha,i}^j = \sigma_{L\alpha,i}^j \times \beta_{\alpha,i}; \alpha = 1, 2, \dots, n$$

where β the stress correction factor, $\sigma_{L\alpha,i}$ and $\sigma_{N\alpha,i}$ are the linear and nonlinear stress responses, respectively, i is the element number, and j is the iteration number. This procedure is used in nonlinear truss design example (Section 4.7.2).

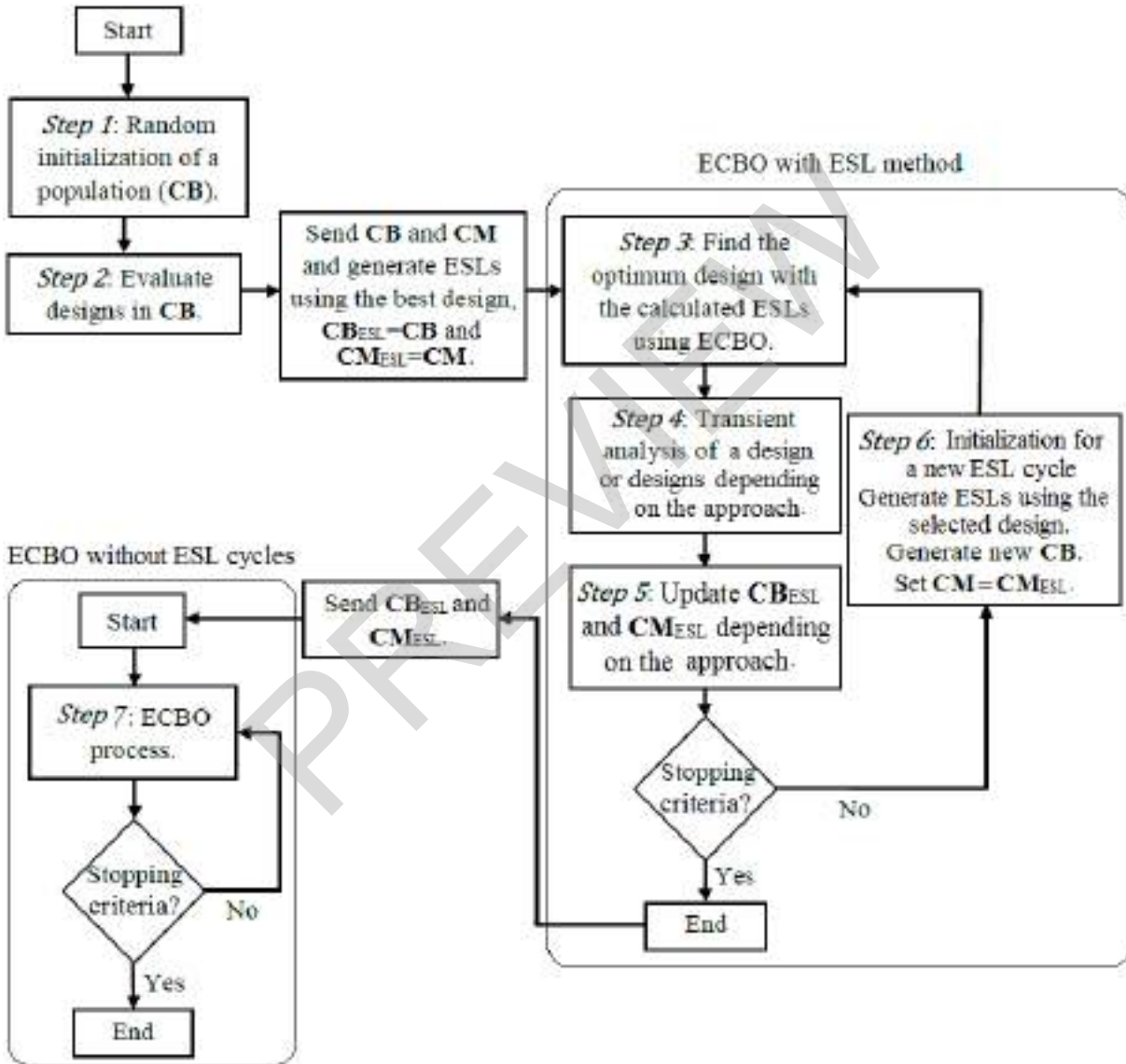


Figure 4.2. GOESL process.

4.7 Numerical Examples

In the following sections, four discrete structural optimization examples are solved for minimum structural weight to test the performance of the proposed algorithm. ECBO and the first two design examples (truss structures) are coded using MATLAB and the models and simulation are verified using the commercial finite element analysis program ANSYS (Bhatti, 2006). The frame design examples are coded in MATLAB and interfaced with the structural analysis program SAP2000 using the Open Application Programming Interface (OAPI).

The first numerical example is solved using the two simultaneous single-step Runge-Kutta method (ODE23 MATLAB function). For the rest of the examples, Newmark's method ($\beta=1/4$ and $\gamma=1/2$) is used for linear and nonlinear dynamic analysis while direct stiffness method is used for linear static analysis.

ECBO parameters are set as follows: population size is 40, *Pro* is 0.4, and the number of designs to be saved in **CM** (*CMS*) is 4 (10% of the population) (Kaveh & Mahdavi, 2015). For all design examples, the time duration for dynamic analysis is set so that the maximum response is covered.

Since the optimization algorithms are stochastic in nature, 10 independent optimization runs were performed for each case to test the performance of ECBO with ESL. In each individual run, the initial population was the same for ECBO without the ESL cycles and GOESL to make a fair comparison.

The quality of proposed method is determined based on the final cost function value, the cost function value after the ESL step, and the total number of transient structural analyses needed to reach the best design.

4.7.1 Eighteen-bar Truss

Figure 4.3 shows the configuration of the 18-bar truss subjected to a half sine wave load at nodes 1, 2, 4, 6, 8. This example was solved in Choi and Park (2002) for continuous design variables and gradient-based optimization. The modulus of elasticity and the density are 69 GPa and 2765 kg/m³, respectively. All members are subjected to stress limitations of 138 MPa in both tension and compression. The allowable displacement for all nodes in both vertical and horizontal directions is ± 203 mm. The optimization problem is to minimize the total mass of the structure. Four size variables and eight shape variables are selected as the design variables.

To test performance of the proposed algorithm, this example is re-formulated as a discrete variable optimization problem. The sizing variables are selected from the discrete set of 100 elements where the range of the cross-sectional area is from 1 to 150 cm² with 1.505 cm² increment. The shape variables are the x and y coordinates of nodes 3, 5, 7, 9. The shape variables are selected from the discrete set of 100 elements where the range is from -317.5 (half the span of 635 cm) to 317.5 cm with 6.141 cm increment. All members of the truss are divided into 4 groups giving 4 sizing design variables (Choi and Park, 2002): all top chord members, all bottom chord members, all vertical members and all diagonal members. Considering the peaks of the displacements and the stresses, the time duration for dynamic analysis is set from 0 to 8 second. The time interval is divided into 100 increments giving 100 loading conditions for static response optimization with the ESL approach. Each loading condition vector has 18 elements since there are 18 degrees of freedom for the truss.

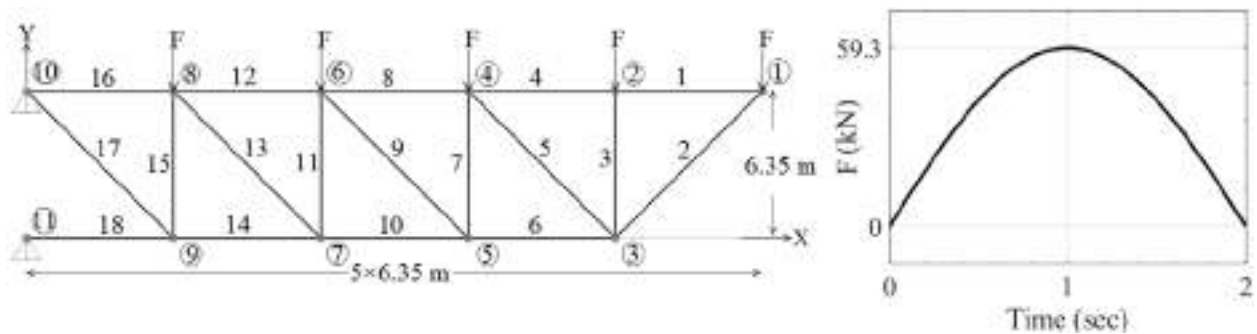


Figure 4.3. Schematic of the 18-bar truss and the applied dynamic load.

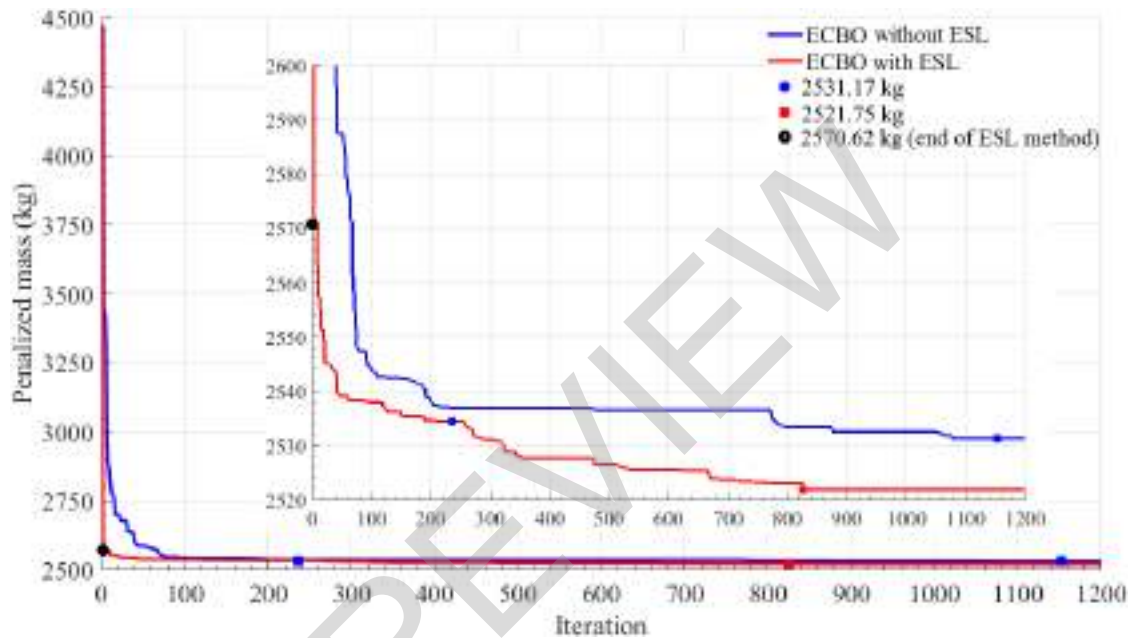


Figure 4.4. Convergence history of 18-bar truss of the first run.

Table 4.1 summarizes results for 10 different runs for ECBO without the ESL cycles and for the three ESL methods. The data in the table for the 10 runs for each ESL method includes: Final mass (kg), mass at the end of ESL cycles (kg), number of ECBO iterations without the ESL cycles, number of dynamic analyses, and number of static analyses. It is interesting to note that of the total 30 runs, 6 runs converged to the best mass value, 17 runs converged to the mass that was within 0.1% of the best value and the remaining 7 runs converged to within 0.2% of the best value. This shows robustness of the proposed algorithm for this example because all the designs would be acceptable from practical applications point of view.

It is noted from the data in Table 4.1 that at the end of ESL cycles, the best design has not been reached for all runs. Therefore, the algorithm must switch to ECBO with dynamic analysis of the entire population to obtain the final design. It is noted that many more ESL cycles beyond the ones shown in Table 4.1, did not result in improved designs.

To compare the ECBO with and without ESL cycles, averages and standard deviations of some key parameters for 10 runs for each method are examined. These data are summarized in Table 4.2. The averages of the final masses and the total number of dynamic analyses show that the proposed method (GOESL with ESL1, ESL2 or ESL3) obtains not only better final designs but also needs a significantly smaller number of dynamic analyses compared to ECBO without the ESL cycles. That is, the average of dynamic analyses of ECBO without ESL cycles is 42524 analyses whereas ESL1, ESL2, and ESL3 have averages of 20964, 22616, and 23000 analyses, respectively.

To study the performance of three proposed ESL approaches, averages and standard deviations for the 10 runs of each ESL method given in Table 4.2 are examined. It is seen that ESL2 has the smallest averages and standard deviations for the final mass as well as the mass at the end of ESL cycles. This shows that ESL2 approach is more reliable in obtaining the final solution. Although ESL2 approach has a slightly higher average for the number of dynamic analyses, it is preferred because of its reliability in obtaining the final design. Performance of ESL1 is a close second to ESL2 for this example.

The best initial and final designs of the first run of ECBO without the ESL cycles and GOESL using ESL2 approach are shown in Table 4.3. For the same initial population, GOESL found a lighter design of 2521.75 kg. After 6 ESL cycles ($6 \times 14 + 40 = 124$ dynamic analyses), the

total structure mass became 2571.70 kg (this is just 1.98 % heavier than the best design). As shown in Figure 4.4, GOESL converges faster than ECBO without ESLs. That is, when ECBO obtains the total mass of 2531.17 kg at iteration 1153, and ECBO with ESL needs just 236 iterations to reach the same mass. That is, with a population of 40 designs, ECBO without ESL cycles needs 917 iterations ($917 \times 40 - 6 \times 14 = 36596$ dynamic analyses) more than GOESL to reach a mass of 2531.17 kg. GOESL final design configuration is depicted in Figure 4.5. The total time needed for ECBO without ESL cycles and for GOESL to obtain the total mass of 2531.17 are 37.21 minutes and 9.13 minutes, respectively (ESL step took only 1.51 minutes) on a desk top computer.

PREVIEW

Table 4.1. Data for 10 different runs of the 18-bar truss.

		Run										
		1	2	3	4	5	6	7	8	9	10	
ECBO without ESL cycles	Mass (kg)	2531.2	2535.7	2525.4	2526.3	2521.8	2524.0	2525.6	2522.7	2523.0	2527.5	
	No. of iterations ^a	1153	1198	1163	1182	840	1097	1133	1186	482	1197	
	No. of dynamic analyses	46120	47920	46520	47280	33600	43880	45320	47440	19280	47880	
GOESL	ESL1	Final mass (kg)	2523.5	2521.2	2521.1	2521.1	2523.5	2523.9	2520.5	2520.5	2522.9	2520.8
		Mass at end of ESL cycles (kg)	2594.2	2607.5	2591.0	2594.8	2588.4	2600.5	2585.3	2590.1	2589.4	2585.7
		No. of iterations ^a	452	591	568	424	568	583	573	545	433	475
		No. of cycles	7	7	12	7	13	5	9	6	6	11
		No. of dynamic analyses	18178	23738	22888	17058	22902	23390	23046	21884	17404	19154
		No. of static analyses	61360	51240	84960	71840	87000	32960	71880	45960	44400	78280
	ESL2	Final mass (kg)	2521.8	2523.0	2520.5	2521.5	2522.9	2521.5	2521.4	2522.2	2523.9	2520.6
		Mass at end of ESL cycles (kg)	2570.6	2589.3	2589.1	2580.7	2585.5	2580.0	2590.4	2579.3	2577.7	2578.7
		No. of iterations ^a	826	539	379	600	563	598	480	597	600	442
		No. of cycles	6	6	11	11	8	7	7	9	7	14
		No. of dynamic analyses	33124	21644	15314	24154	22632	24018	19298	24006	24098	17876
		No. of static analyses	45920	43240	82480	67840	50320	56960	46320	56240	50440	104320
	ESL3	Final mass (kg)	2522.3	2522.5	2525.5	2524.1	2520.8	2523.8	2525.5	2523.3	2524.8	2522.6
		Mass at end of ESL cycles (kg)	2594.9	2618.8	2607.4	2613.4	2597.8	2603.8	2619.4	2610.2	2598.8	2615.3
		No. of iterations ^a	559	535	588	561	594	598	541	565	589	600
No. of cycles		7	5	6	5	7	5	5	5	5	7	
No. of dynamic analyses		22458	21470	23604	22510	23858	23990	21710	22670	23630	24098	
No. of static analyses		45720	30920	52160	45160	43240	35880	38640	38720	38880	52080	

^a ECBO iterations without ESL cycles.

Table 4.2. Comparison of averages and standard deviations for 10 runs of the 18-bar truss.

Metric	Averages				Standard deviation			
	ECBO Alone	ESL1	ESL2	ESL3	ECBO Alone	ESL1	ESL2	ESL3
Final mass (kg)	2526.32	2521.92	2521.85	2523.5	4.28	1.37	1.10	1.50
Mass at end of ESL cycles (kg)	-	2592.68	2583.41	2607.97	-	6.90	5.13	8.90
No. of dynamic analyses	42524	20964	22616	23000	9208	2691	4791	962
No. of static analyses	-	62988	60408	42140	-	18644	19488	6848

Table 4.3. Initial and final designs of 18-bar truss for the first run.

Design variables	Best initial design	ECBO	GOESL (ESL2)						Final design
		Final design	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	
1 Area _{top} (mm ²)	9431.31	10635.35	14548.48	10484.85	10484.85	10785.86	11237.37	11237.37	10334.34
2 Area _{bottom} (mm ²)	13946.46	10635.35	10033.33	10785.86	10785.86	11688.89	11538.38	11538.38	11387.88
3 Area _{vertical} (mm ²)	14698.99	4013.13	2207.07	3561.62	3561.62	3561.62	3260.61	3260.61	3561.62
4 Area _{diagonal} (mm ²)	7324.24	4314.14	4163.64	5518.18	5518.18	4314.14	4314.14	4314.14	4163.64
5 X ₃ (mm)	2533.59	737.63	-3175.00	1250.76	1250.76	-3110.86	-2854.29	-2854.29	416.92
6 Y ₃ (mm)	-2277.02	3175.00	2726.01	2148.74	2148.74	2148.74	2277.02	2277.02	3175.00
7 X ₅ (mm)	-96.21	-865.91	-3175.00	-3046.72	-3046.72	-3110.86	-3175.00	-3175.00	-994.19
8 Y ₅ (mm)	-1186.62	481.06	1250.76	865.91	865.91	930.05	1058.33	1058.33	673.48
9 X ₇ (mm)	2341.16	-2341.16	-1956.31	-2405.30	-2405.30	-2597.73	-2533.59	-2533.59	-2212.88
10 Y ₇ (mm)	-1571.46	-224.49	224.49	-32.07	-32.07	-32.07	224.49	224.49	-160.35
11 X ₉ (mm)	-2084.60	-3175.00	-1699.75	-2341.16	-2341.16	-2148.74	-2277.02	-2277.02	-2854.29
12 Y ₉ (mm)	-1250.76	-96.21	-288.64	-288.64	-288.64	-224.49	-160.35	-160.35	-160.35
Max. Displacement (mm)	14.76 (1 ^a)	203.00 (1)	205.12 (1)	202.50 (1)	202.50 (1)	202.46 (1)	202.76 (1)	202.76 (1)	203.00 (1)
Max. Stress (MPa)	86.98 (17 ^b)	111.18 (18)	130.13 (15)	110.11 (18)	110.11 (18)	116.93 (17)	109.86 (17)	109.86 (17)	114.01 (17)
Mass (kg)	4470.86	2531.17	2621.58	2621.58	2621.58	2577.38	2570.62	2570.62	2521.75
Merit	4470.86	2531.17	2621.58	2621.58	2621.58	2577.38	2570.62	2570.62	2521.75
Iteration	-	1153 ^c	233 ^d	233 ^d	232 ^d	150 ^d	150 ^d	150 ^d	826 ^c
Dynamic, static analyses	-	46120, 0	14, 9320	14, 9320	14, 9280	14, 6000	14, 6000	14, 6000	33040, 0

Top members: 1, 4, 5, 12, and 16. Bottom members: 2, 6, 10, 14, and 18. Vertical members: 3, 7, 11, and 15. Diagonal members: 5, 9, 13, and 17.

^a Node number where the maximum displacement occurs. ^b Member number where the maximum stress occurs. ^c Transient analysis.

^d Linear static analysis.

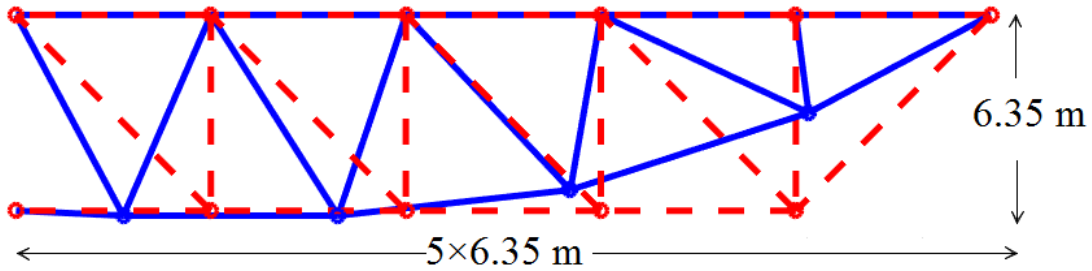


Figure 4.5. Optimum configuration for the 18-bar truss.

4.7.2 Ten-bar Truss with Material Nonlinearity

Figure 4.6 shows the configuration of the 10-bar truss subjected to a half sine wave load at nodes 2 and 4. This example was solved in Kim and Park (2010) for continuous design variables using a gradient-based optimization algorithm. The material nonlinearity is considered in this problem. The Young's modulus is 200 GPa, the tangent modulus is 50 GPa, the yield stress is 200 MPa, the Poisson ratio is 0.3, and the mass density is 7860 kg/m³.

To evaluate the proposed algorithm, this problem is also re-formulated as a discrete variable problem. The optimization problem is to minimize the total mass of the structure. The design variables are the cross-sectional areas of the members (Table 4.6). The size variables are selected from the discrete set of 100 elements where the range of the cross-sectional areas is from 78.5 to 2826 cm² with 27.752 mm² increment. All members are subjected to stress limitations of 250 MPa in both tension and compression. Considering the peaks of the displacements and the stresses, the time duration for the analysis is set from 0 to 0.03 second with time step of 0.0002 second. This gives 150 loading conditions for static response optimization with the ESL approach. Each loading condition vector has 8 elements since there are 8 degrees of freedom for the truss.

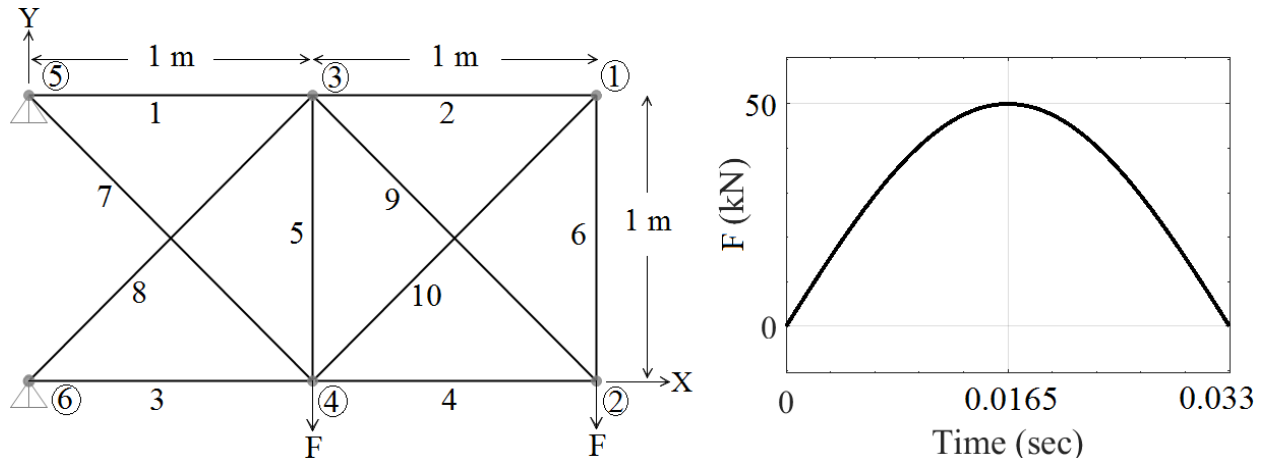


Figure 4.6. Schematic of the 10-bar truss and the applied dynamic load.

Table 4.4 shows results for 10 different runs of ECBO without the ESL cycles and the results of GOESL with the three approaches, ESL1, ESL2 and ESL3. The data in the table for all the 30 runs includes: final mass (kg), mass at end of ESL cycles (kg), number of ECBO iterations without the ESL cycles, number of dynamic analyses, and number of static analyses. It is interesting to note that of the 30 runs, 4 runs converged to the best mass value, 10 runs converged to the mass that was within 2.5% of the best value and 8 runs converged to within 5.0% of the best value.

An examination of the averages and standard deviations in Table 4.5 for this example leads to the same conclusion as for Example 1: GOESL obtains better designs with less number of dynamic analyses compared to ECBO without the ESL cycles, and ESL2 approach performs more reliably in obtaining the final design than ESL1 and ESL3.

Table 4.4. Data for 10 different runs of the 10-bar truss.

		Run										
		1	2	3	4	5	6	7	8	9	10	
ECBO without ESL cycles	Mass (kg)	26.9	30.0	26.2	25.6	25.4	31.2	25.9	26.0	25.5	25.4	
	No. of iterations ^a	765	395	624	936	406	772	833	380	422	664	
	No. of dynamic analyses	30600	15800	24960	37440	16240	30880	33320	15200	16880	26560	
GOESL	ESL1	Final mass (kg)	29.6	25.8	25.2	25.4	25.4	25.6	27.7	25.8	25.9	25.6
		Mass at end of ESL cycles (kg)	61.7	58.3	47.0	63.6	43.5	87.4	77.0	45.9	140.4	54.9
		No. of iterations ^a	465	252	313	187	219	202	180	351	363	194
		No. of cycles	6	7	14	8	7	7	5	7	7	5
		No. of dynamic analyses	18606	10087	12534	7488	8767	8087	7205	14047	14527	7765
		No. of static analyses	37640	44800	81360	42960	50160	45240	30720	48480	59120	33720
	ESL2	Final mass (kg)	26.1	25.2	26.1	25.7	26.1	26.1	25.2	25.3	25.7	25.9
		Mass at end of ESL cycles (kg)	41.6	29.6	39.0	36.1	45.2	61.1	68.9	35.0	51.6	39.6
		No. of iterations ^a	374	294	227	251	267	258	417	162	339	345
		No. of cycles	7	21	9	10	6	6	5	11	5	14
		No. of dynamic analyses	15058	12054	9206	10180	10764	10404	16750	6634	13630	13996
		No. of static analyses	46200	119080	59720	63200	47640	33120	29000	66000	39720	80120
	ESL3	Final mass (kg)	27.7	25.2	27.0	25.9	27.6	30.2	28.7	27.1	25.3	26.2
		Mass at end of ESL cycles (kg)	54.3	79.4	63.7	74.6	83.8	86.1	77.7	63.2	65.2	75.3
		No. of iterations ^a	255	242	341	204	283	354	138	113	324	409
No. of cycles		5	5	5	7	6	5	5	5	5	5	
No. of dynamic analyses		10270	9750	13710	8258	11404	14230	5590	4590	13030	16430	
No. of static analyses		29280	25480	27160	35280	30560	25520	27240	26120	35200	33560	

^a ECBO iterations without ESL cycles.

Table 4.5. Comparison of averages and standard deviations for 10 runs of the 10-bar truss.

Metric	Averages				Standard deviation			
	ECBO Alone	ESL1	ESL2	ESL3	ECBO Alone	ESL1	ESL2	ESL3
Final mass (kg)	26.8	26.2	25.73	27.08	2.1	1.36	0.39	1.56
Mass at end of ESL cycles (kg)	-	67.96	44.75	72.32	-	28.97	12.33	10.26
No. of dynamic analyses	24788	10911	11869	10726	8268	3849	3025	3812
No. of static analyses	-	47420	58380	29540	-	14496	26569	3909

Table 4.6. Initial and final design of 10-bar truss of the first run.

Design variables (mm ²)		Best initial design	ECBO	GOESL (ESL2)								
			Final design	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Final design	
1	A ₁	439.28	439.28	855.57	855.57	661.30	661.30	661.30	661.30	661.30	661.30	605.80
2	A ₂	494.79	300.52	411.53	272.77	134.01	134.01	161.76	189.51	189.51	189.51	134.01
3	A ₃	1632.64	661.30	550.29	550.29	550.29	550.29	550.29	550.29	550.29	550.29	522.54
4	A ₄	1771.40	78.50	439.28	411.53	217.26	217.26	217.26	189.51	189.51	189.51	161.76
5	A ₅	1632.64	106.25	411.53	217.26	134.01	134.01	134.01	134.01	134.01	134.01	161.76
6	A ₆	1382.87	217.26	1438.37	356.03	161.76	161.76	161.76	217.26	217.26	217.26	106.25
7	A ₇	383.78	550.29	1188.60	800.07	689.06	689.06	689.06	661.30	661.30	661.30	328.27
8	A ₈	134.01	189.51	1327.36	1327.36	1299.61	1299.61	1299.61	1299.61	1299.61	1299.61	439.28
9	A ₉	605.80	78.50	189.51	633.55	245.02	245.02	245.02	217.26	217.26	217.26	217.26
10	A ₁₀	1299.61	328.27	494.79	439.28	217.26	217.26	189.51	189.51	189.51	189.51	161.76
Max. Stress (MPa)		242.60 (7 ^a)	249.88 (9)	243.92 (3)	248.57 (3)	252.37 (2)	252.37 (2)	250.50 (3)	246.16 (3)	246.16 (3)	246.16 (3)	249.15 (7)
Mass (kg)		84.74	26.92	67.85	56.51	41.83	41.83	41.76	41.58	41.58	41.58	26.05
Merit		84.74	26.92	67.85	56.51	43.10	43.10	41.93	41.58	41.58	41.58	26.05
Iteration		-	765	285	163	125	207	125	125	125	125	374
Dynamic, static analyses		-	30600, 0	14, 11400	14, 6520	14, 5000	14, 8280	14, 5000	14, 5000	14, 5000	14, 5000	14960, 0

^a Member number where the maximum stress occurs.

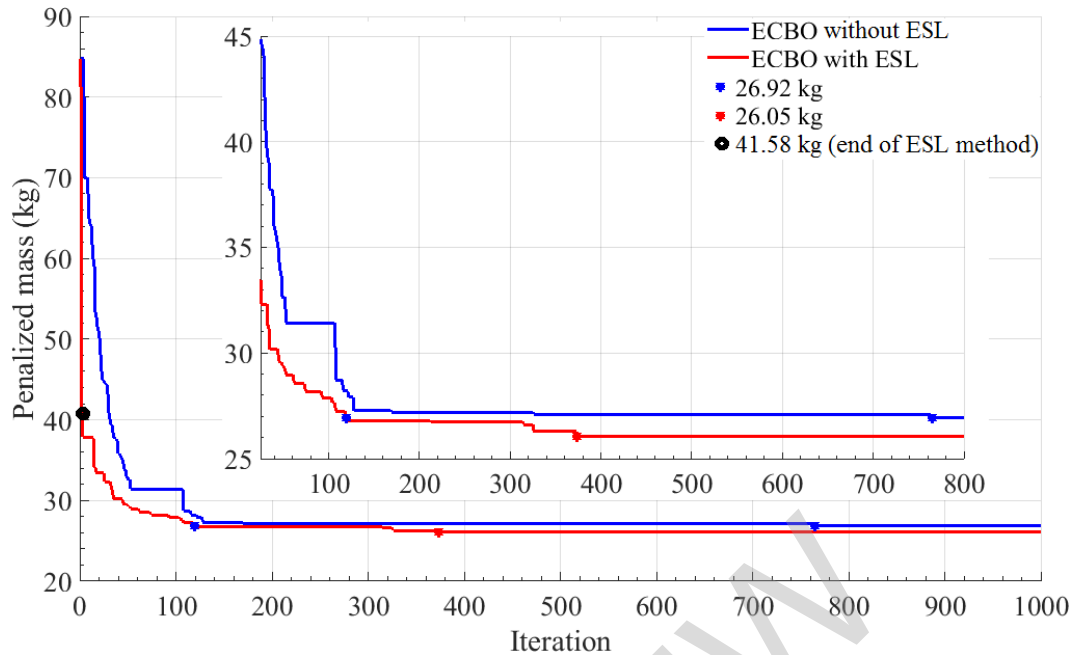


Figure 4.7. Convergence history of 10-bar of the first run.

The best initial and final designs of the first run of ECBO without ESL cycles and GOESL using ESL2 are shown in Table 4.6. For the same initial population ECBO with ESL2 found a lighter design of 26.05 kg. After 8 ESL cycles, the total structure mass became 41.58 kg (the best design is 59.62% lighter). As shown in Figure 4.7, GOESL converges faster than ECBO without ESL cycles. ECBO reaches the best design of 26.92 kg at iteration 765 and GOESL needs just 119 iterations to obtain a similar mass. That is, with a population of 40 designs, ECBO needs 25728 more dynamic analyses (646 iterations) than GOESL. The total time needed for ECBO without ESL cycles and for ECBO with ESL2 to reach the best design are 18.36 minutes and 10.16 minutes, respectively (ESL step took only 1.18 minutes).

The average of final masses and the average of total number of dynamic analyses of the proceeding two examples show that the proposed method (GOESL) gives better results. The best design from dynamic analysis approach (the second approach, ESL2) shows better convergence

behavior and final designs of the three approaches. Therefore, this approach is used in the next two examples.

4.7.3 Two-story Two-bay Frame

This design example is a 2-story, 2 bays planar steel frame having 4 beams and 6 columns and has not been solved in the literature before. It is modeled using SAP2000 and MATLAB with 19 nodes and 20 elements. Note that in order to get more accurate analysis results intermediate nodes are introduced for each member of the frame. The frame has 48 degrees of freedom that is subjected to a half sine wave load at nodes 2 and 3 and uniformly distributed static load of 5 kip/ft on members 13 to 20 as shown in Figure 4.8. All ground supports are fixed. Material properties are: Young's modulus, $E=29000$ ksi, yield stress, $F_y=50$ ksi, and Poisson's ratio, $\nu=0.3$.

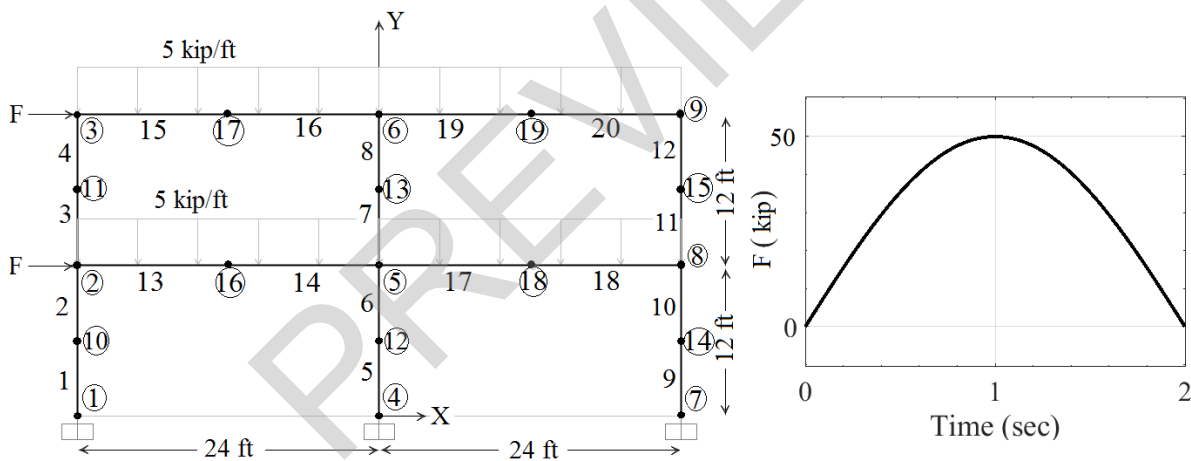


Figure 4.8. Schematic of the 2-story 2-bay frame and the applied dynamic load.

Columns and beams are selected from the first lightest 100 standard W-shapes provided in AISC tables (ASIC, 2017). The sections are rearranged in an ascending order based on their weight. The problem is formulated as an integer variable optimization problem where the section number is treated as a design variable. To further explain the design variables, consider a small part of the AISC (2017) wide-flange sections table shown in **Error! Reference source not found..**

Once an integer value is assigned to a design variable, a section is specified. For example, if a design variable is assigned value of 4, then the section from the Table 4.7 is W44X230. For this section, the weight per foot is 230 lbs, the cross-sectional area is 67.8 inch², total depth is 42.9 inches, and so on. In other words, all the cross-sectional properties are available to formulate and check the performance constraints. It is seen that there is no continuous functional relationship between the section number and the cross-sectional properties. Therefore, it is not possible to formulate and differentiate the problem functions with respect to the design variables.

Table 4.7. ASIC W-Shapes Database (partial).

Section number	Shape	W (lb/ft)	A (in ²)	d (in)	Web		Flange		Axis X-X				...	h _o (in)	P _A (in)	P _B (in)
					t _w (in)	t _w /2 (in)	b _f (in)	t _f (in)	I _x (in ⁴)	Z _x (in ³)	S _x (in ³)	r _x (in)				
1	W44X335	335	98.5	44.0	1.030	1/2	15.9	1.77	31100	1620	1410	17.8	...	42.2	132	148
2	X290	290	85.4	43.6	0.865	7/16	15.8	1.58	27000	1410	1240	17.8	...	42.0	131	147
3	X262	262	77.2	43.3	0.785	7/16	15.8	1.42	24100	1270	1110	17.7	...	41.9	131	147
4	X230	230	67.8	42.9	0.710	3/8	15.8	1.22	20800	1100	971	17.5	...	41.7	130	146
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
273	W4X13	13	2.83	4.16	0.280	1/8	4.06	0.345	11.3	6.28	5.46	1.72	...	97	495	599

The strength requirement for the members is based on the AISC interaction ratio constraint expressed as follows:

$$\frac{P_u}{\phi P_n} + \frac{8}{9} \left(\frac{M_{uz}}{\phi_b M_{nz}} \right) - 1 \leq 0 \quad \text{if } \frac{P_u}{\phi P_n} \geq 0.2 \quad (4.20)$$

$$\frac{P_u}{2\phi P_n} + \left(\frac{M_{uz}}{\phi_b M_{nz}} \right) - 1 \leq 0 \quad \text{if } \frac{P_u}{\phi P_n} < 0.2$$

here ϕ is the resistance factor ($\phi_c = 0.85$ and $\phi_t = 0.90$ for compression and tension, respectively). $\phi_b = 0.90$ is the flexural resistance factor. P_u and P_n are the required and the nominal axial strengths (compression or tension) (kips), respectively. M_{uz} is the required flexural strength (kip-ft). M_{nz} is the nominal flexural strength (kip-ft). Constraints in Eq. (4.20) needs to be imposed at each point along the axis of every member in the structure. Thus, the equation represents infinite constraints.

In the numerical process, the constraints are evaluated at several points along the axis of the member and they imposed at the point where they have maximum value. These constraint values are then used to evaluate the penalty function.

In Eq. (4.20), evaluation of P_n and M_{nz} is an involved process (AISC, 2017) that requires checking of several failure modes (i.e., several “if then else” requirements). For example, to find P_n , first one needs to find whether the member force is tensile or compressive. For tension members, P_n is calculated based on whether the gross section yields or the net section ruptures. For compression members, P_n is calculated based on consideration of several failure modes, such as yielding of the material, local buckling of flanges or the web (elastic or inelastic), and global buckling (elastic or inelastic). Similarly, calculation of M_{nz} involves checking several flexural failure modes. All the foregoing calculations involve various cross-sectional properties of the sections that are available in Table 4.7.

Thus, it is concluded that it is not possible to obtain a functional expression for the constraints in Eq. (4.20) in terms of the design variables, the integer number of the sections. Even if that were somehow possible, there would be several discontinuities in the functions due to all the “if then else” requirements mentioned in the foregoing paragraph. Also notice that constraints in Eq. (4.20) have a discontinuity at $\frac{P_u}{\phi P_n} = 0.2$. Therefore, Due to all these reasons, the gradient-based methods are not applicable for this class of applications.

Considering the peaks of the displacements and the stresses, the time range for the analysis is set from 0 to 4 second with time step of 0.04 second. This gives 100 loading conditions for static response optimization with the ESL approach. Each loading condition vector has 48 elements since there are 48 degrees of freedom for the frame.

This example was also solved without the intermediate nodes for the members. That model was more efficient to solve. However, the final designs were not as good as with the increased degrees of freedom. The reason is that with more degrees of freedom a more accurate dynamic response is obtained resulting in better ESLs as well.

Table 4.9 Table 4.8 gives the best initial and final designs of the first run. For the same initial population GOESL found a lighter design of 9108 lb. After 6 ESL cycles, the total structure weight becomes 9996 lb (the best design is 9.75% lighter). As shown in Figure 4.9, GOESL converges faster than ECBO without ESL cycles. ECBO reaches the best design of 9492 lb at iteration 207 whereas GOESL needs 43 iterations to obtain a similar design of 9492 lb. From Table 4.9, the average of 10 individual runs for final weight is better with GOESL than with ECBO without the ESL approach. The average of total number of dynamic analyses shows that GOESL needs about half the number dynamic analyses than ECBO without ESL cycles to reach the final design.

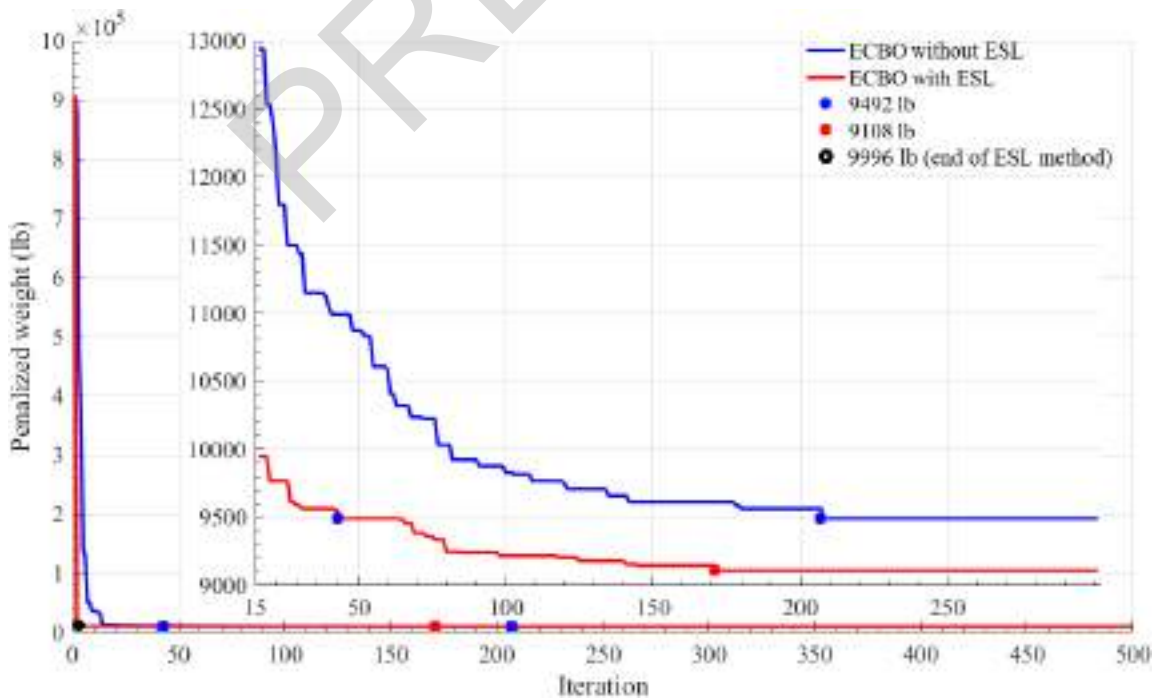


Figure 4.9. Convergence history of 2-story 2-bay frame.

Table 4.8 Initial and final design of 2-story 2-bar frame of the first run.

Design variable no.	Member no.	Best initial design	ECBO		GOESL (ESL2)					
			Final design	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Final design
1	1-2	W27X281	W24X229	W40X167	W24X229	W30X90	W27X281	W36X160	W36X160	W40X167
2	3-4	W30X261	W36X529	W36X302	W27X129	W33X354	W36X182	W33X354	W33X354	W36X302
3	5-6	W36X135	W24X229	W24X306	W24X229	W27X129	W30X108	W24X306	W24X306	W24X306
4	7-8	W40X324	W33X241	W36X282	W27X539	W27X307	W30X90	W36X182	W36X182	W36X282
5	9-10	W36X160	W36X282	W36X247	W36X135	W30X99	W33X130	W36X256	W36X256	W36X247
6	11-12	W27X539	W33X354	W36X160	W33X318	W30X173	W36X256	W36X182	W36X182	W36X160
7	13-14	W27X258	W27X281	W24X306	W24X306	W24X250	W27X217	W24X306	W24X306	W24X306
8	15-16	W30X148	W27X336	W27X258	W27X258	W27X336	W27X281	W27X258	W27X258	W27X258
9	17-18	W24X335	W33X318	W33X130	W27X217	W27X258	W30X108	W30X90	W30X90	W33X130
10	19-20	W24X250	W30X235	W30X211	W27X161	W30X108	W30X124	W30X90	W30X90	W30X211
Max. interaction ratio		3.953 (6 ^a)	0.999 (4)	1.050 (8)	0.920 (5)	3.531 (6)	1.155 (14)	0.985 (10)	0.985 (10)	0.998 (5)
Weight (lb)		10476	9492	10596	11832	11184	9948	9996	9996	9108
Merit		907201.4	9492	12928.15	11832	424985.4	17378.3	9996	9996	9108
Iteration		-	207	142	214	121	154	154	63	171
Dynamic, static analyses		-	8280, 0	14, 5680	14, 8560	14, 4840	14, 6160	14, 6160	14, 2520	6840, 0

^a Member number where the maximum interaction ratio occurs.

Table 4.9. 10 individual runs data of the 2-story 2-bay frame.

		Run										Average	SD
		1	2	3	4	5	6	7	8	9	10		
ECBO without ESL cycles	Weight (lb)	9492	9708	9720	9972	9240	9168	9744	9924	9468	9708	9614.4	254.0
	No. of iteration	207	271	297	303	214	215	277	278	328	347	274	46
No. of dynamic analyses		8280	10840	11880	12120	8560	8600	11080	11120	13120	13880	10948	1843
GOESL ESL2	Final weight (lb)	9108	9216	9372	9264	9288	9132	9144	9552	9132	8988	9219.6	159.0
	ESL final weight (lb)	9996	10152	10488	11360	10100	10440	10020	10224	9996	10572	10334.8	417.8
	No. of iterations	171	160	117	158	175	121	124	90	160	178	145	30
	No. of cycles	6	5	7	5	5	8	9	5	5	7	6.2	1.5
	No. of dynamic analyses	6924	6470	4778	6390	7070	4952	5086	3670	6470	7218	5903	1194
No. of static analyses		33920	19440	36480	29720	28880	42040	35800	18640	17600	33920	29644	8470

4.7.4 Two-story Two-bay Frame Subjected to Blast Loads with Material and Geometric Nonlinearity

The configuration of this numerical example is the same as the previous example except that it is subjected to blast load. This example has also not been solved in the literature. For simplicity, the blast load is modeled as triangle and the negative pressure phase is neglected. In addition, a uniformly distributed static load of 5 kip/ft on members 13 to 20 is added as shown in Figure 4.10. All ground supports are fixed. Young's modulus, $E=29000$ ksi, yield stress, $F_y=50$ ksi, ultimate stresses, $F_u=65$ ksi and Poisson's ratio, $\nu=0.3$. Due to the dynamic effects resulting from the high strain rates, the dynamic increase factors (*DIF*) for yield and ultimate stresses of 1.19 and 1.05, respectively, are used (Gilsanz et al., 2013). Since the average yield stress for structural steels having a specified minimum yield stress of 50 ksi or less is generally higher than the specified minimum, it is recommended that the minimum design yield stress, as specified by the AISC (2011) specification, be increased by 10 percent. This increasing factor is called the strength increase factor (*SIF*). Also, for all modes of failure, it shall be permissible to use a strength reduction factor (ϕ) of 1.0 instead of smaller than 1 value (ASCE, 2011). The reader is referred to ASCE (2010), ASCE (2011), Dusenberry (2010), Gilsanz et al. (2013), and DoD (2008) for more details. Therefore, the new strength values are as follows:

$$F_{dy} = (SIF)(DIF)F_y = (1.1)(1.19)(50) = 65.45 \text{ ksi} \quad (4.21)$$

$$F_{du} = (DIF)F_u = (1.05)(65) = 68.25 \text{ ksi} \quad (4.22)$$

Columns and beams are selected from the first lightest 150 standard W-shapes provided in AISC tables (ASIC, 2017) after rearranging sections in an ascending order based on their weight. Columns are designed to remain elastic and subjected to the AISC interaction ratio constraint (Eq. (4.20)) while beams are allowed to develop plastic hinges. Steel Beams-Flexure elastic-perfectly

plastic hinges provided by SAP2000 v.20 are modeled near the start of members 13, 15, 17, and 19 and near the end of members 14, 16, 18, and 20 as shown in Figure 4.10. The locations of these plastic hinges are chosen based on observing where maximum bending moments occur. The maximum member end rotation shall be 1 degree and the maximum side-sway deflection (or inter-story drift (*ISD*)) is limited to 1/50 of the story height (low response design (ASCE, 2010)).

$$ISD \leq \frac{H}{50} (2.9 \text{ in}) \quad (4.23)$$

where H is the height of the story. It was noticed that when there are many plastic hinges the numerical solver stops converging before reaching the maximum analysis time which indicates the structure becomes unstable. In this case, G (in Eq. (4.7)) is set to 10 to eliminate structurally unstable designs in the optimization process.

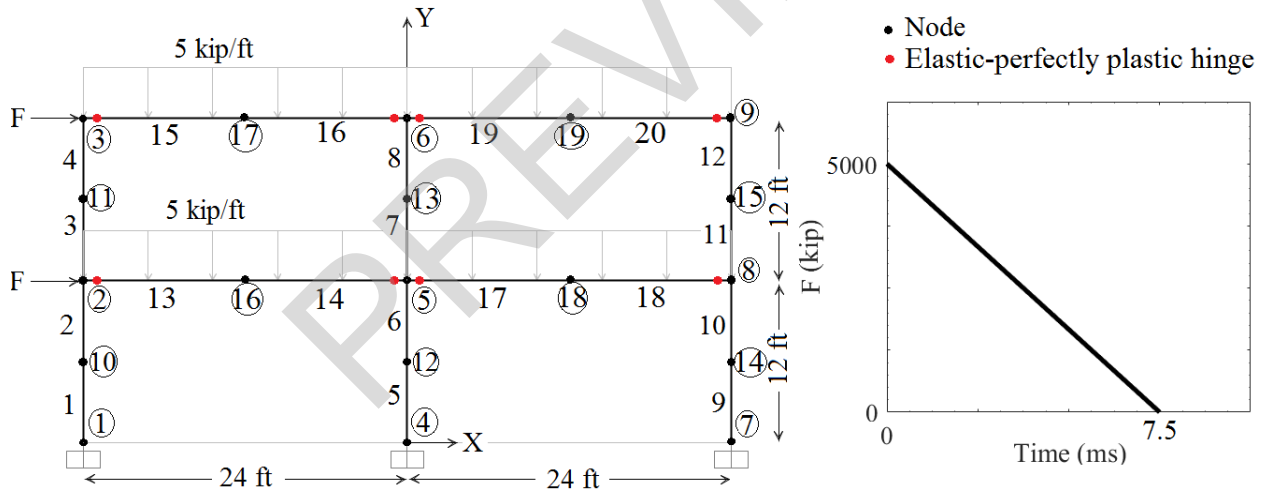


Figure 4.10. Schematic of the 2-story 2-bay frame and the applied blast load.

Considering the blast load duration and the peaks of the response, the time range from the analysis is set from 0 to 1.25 second with time step of 0.0025. This gives 500 loading conditions for static response optimization with the ESL approach. Each loading condition vector has 48 elements since there are 48 degrees of freedom for the frame.

This example was solved with 9 joints and 10 members (no intermediate nodes). Similar to the previous example, it was observed that increasing the degrees of freedom makes GOESL converges to better designs for the same reasons as noted earlier.

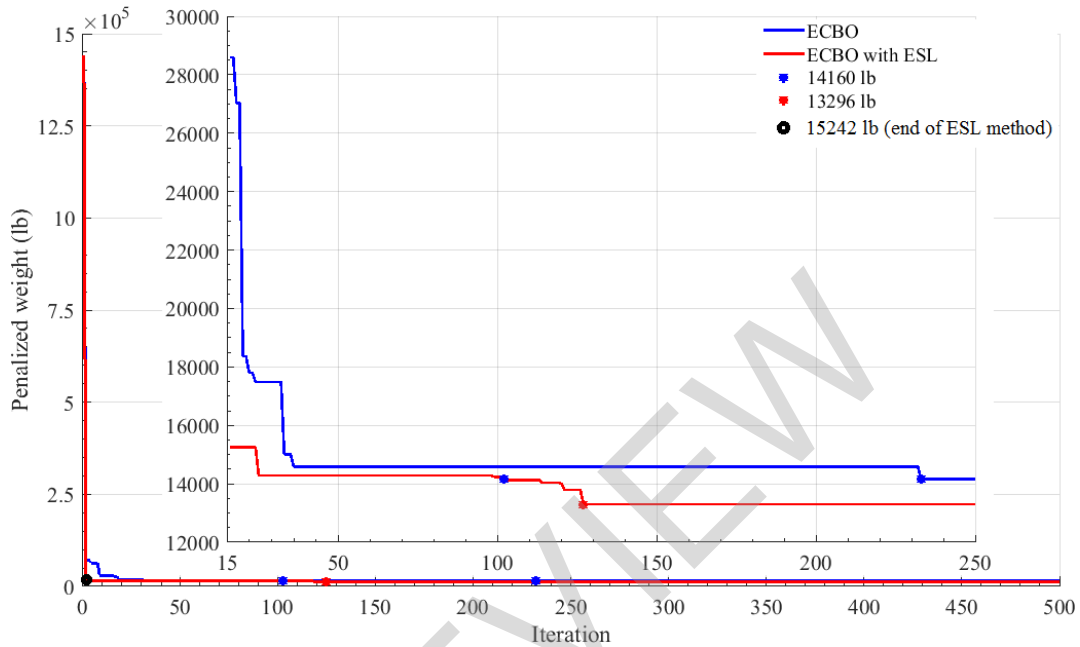


Figure 4.11. Convergence history of 2-story 2-bay frame subjected to blast load.

Table 4.10 gives the best initial and final designs for the first run of the problem. For the same initial population, GOESL found lighter design of 13296 lb. After 5 ESL cycles, the total structure mass becomes 15264 lb (the best design is 14.8% lighter). As shown in Figure 4.11, GOESL converges faster than ECBO without ESL cycles. That is, when ECBO without ESL cycles obtains the total weight of 14160 lb at iteration 232, GOESL needs 102 iterations to reach a similar structural weight. That is, with population of 40 designs, ECBO without ESL cycles needs 130 more iterations (5130 dynamic analyses) than ECBO with ESLs. Table 4.11 summarizes results for 10 different runs for ECBO without the ESL cycles and for GOESL. The average of the final weights and the total number of dynamic analyses show that GOESL obtains not only better average but also needs significantly smaller number of dynamic analyses (less than half) compared to ECBO without the ESL cycles.

Table 4.10. Initial and final design of 2-story 2-bar frame subjected to blast load of the first run.

Design variable no.	Member no.	Best initial design	GOESL (ESL2)						
			ECBO Final design	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Final design
1	1-2	W21X101	W21X147	W10X77	W24X76	W36X150	W36X150	W36X150	W18X130
2	3-4	W21X57	W33X130	W21X83	W18X76	W36X150	W36X150	W36X150	W36X135
3	5-6	W21X44	W14X82	W21X111	W14X109	W36X150	W36X150	W36X150	W16X89
4	7-8	W14X120	W18X50	W18X86	W24X84	W12X120	W12X120	W12X120	W27X102
5	9-10	W10X39	W14X61	W24X104	W24X103	W10X68	W10X68	W10X68	W12X50
6	11-12	W10X100	W33X118	W30X90	W10X88	W10X68	W10X68	W10X68	W24X62
7	13-14	W33X130	W27X129	W12X45	W21X44	W40X149	W40X149	W40X149	W40X149
8	15-16	W12X53	W14X48	W6X25	W10X26	W12X45	W12X45	W12X45	W18X50
9	17-18	W14X74	W8X40	W21X50	W10X54	W10X45	W10X45	W10X45	W12X35
10	19-20	W18X65	W12X79	W21X50	W21X50	W21X44	W21X44	W21X44	W16X36
Max. interaction ratio		2.798 (6 ^a)	0.979 (7)			1.000 (3)	1.000 (3)	1.000 (3)	0.934 (5)
Max. rotation (degree)		1.449 (2 ^b)	1.000 (2)	Unstable ^c	Unstable ^c	0.964 (6)	0.964 (6)	0.964 (6)	0.8697 (2)
Max. ISD (in)		2.578	2.343			2.155	2.155	2.155	1.964
Weight (lb)		13260	14160	10692	10608	15264	15264	15264	13296
Merit		1441347	14160	1293732	1283568	15264	15264	15264	13296
Iteration		-	232	136	92	133	84	152	125
Dynamic, static analyses		-	9280, 0	14, 5440	14, 3680	14, 5320	14, 3360	14, 6080	5000, 0

^a Member number where the maximum interaction ratio occurs. ^b Node number where the maximum rotation occurs.

^c The numerical solver stops converging ($G=10$).

Table 4.11. 10 individual runs data of the 2-story 2-bar frame subjected to blast load.

		Run										Average	SD
		1	2	3	4	5	6	7	8	9	10		
ECBO without ESL cycles	Weight (lb)	14160	10680	11868	13080	15785	12840	11496	11724	12960	12468	12706	1382
	No. of iteration	232	388	400	188	296	187	366	370	353	393	317	81
	No. of dynamic analyses	9280	15520	16000	7520	11840	7480	14640	14800	14120	15720	12692	3235
GOESL ESL2	Final weight (lb)	13296	11976	12180	12348	12192	11604	12160	13956	13980	11940	12563	858
	ESL final weight (lb)	15264	15044	21177	19774	15636	18183	18028	20182	16566	15108	17496	2303
	No. of iterations	125	126	174	200	209	165	171	162	96	87	152	41
	No. of cycles	5	5	7	5	5	5	5	5	5	5	5.2	0.6
	No. of dynamic analyses	5070	5110	7058	8070	8430	6670	6910	6550	3910	3550	6132.8	1658
No. of static analyses		23880	29240	24920	19200	20320	19120	23400	15960	20640	15760	21244	4176

4.8 Concluding Remarks

Metaheuristic algorithms are often used to optimize problems when it is not possible to compute gradients of the cost and/or constraints functions and/or design variables are not continuous. However, depending on the number of design variables and number of elements in the allowable discrete set, these stochastic algorithms require too many structural analyses. Also, more than one individual run is needed to ensure that the best design has been obtained (since these algorithms are stochastic in nature). That is, optimizing transient problems using metaheuristic algorithms is computationally expensive because every simulation requires solving a system of differential equations. One way to reduce the wall-clock time to solve problems using metaheuristic algorithms is to use parallel processing. This aspect has not been addressed in the current research.

In search for a more efficient method for dynamic response structural optimization, the Equivalent Static Load (ESL) approach with gradient-free algorithms was examined in this study. In the proposed method, the transient problem was transformed to ESL sets that generated the same displacement field as with the transient analysis for a given design. Then, the sets of generated ESLs were used as a multiple loading conditions in the static response structural optimization process. Since it was not clear which design should be used at the end of each ESL cycle to generate ESLs in metaheuristic algorithms, three approaches were studied: the best design from static analysis (ESL1), the best design from dynamic analysis (ESL2), and the heaviest feasible design from dynamic analysis (ESL3).

Based on the analysis of results of four numerical examples (2 linear and 2 nonlinear), the following conclusions are drawn:

- 1- ESL approach with metaheuristic algorithms is not able to obtain the best design because the ESLs calculated for the chosen member of the population are not suitable for the remaining members of the population. Also, a small change in design variables is not guaranteed in metaheuristic algorithms from one ESL cycle to the next. This violates the assumption of small changes in design from one ESL cycles to the next with the gradient-based methods (at least near the local minimum point).
- 2- At the end of ESL cycles, improved designs are obtained although not the best design.
- 3- At the end of ESL cycles, the better designs and the improved population may be passed on to the metaheuristic method without the ESL cycles to improve these designs further, if desired.
- 4- In most cases, it is shown that the proposed method can reach the best design with substantially less number of dynamic analyses than with the metaheuristic algorithm without the ESL cycles.
- 5- Among the three ESL approaches investigated, ESL2 ranked first, ESL1 was close second and ESL3 was third based on reliability of obtaining the best design.

4.9 Reproducing Results

To reproduce results provided in this work, all the necessary information about design examples are described in Section 4.7, the steps to implement GOESL is shown in Section 6, and the implementation of ECBO is provided in Kaveh & Ghazaa (2014). Appendix B includes MATLAB code for the 18-bar truss design example.

CHAPTER 5

OPTIMIZATION OF FRAMED STRUCTURES SUBJECTED TO BLAST LOADING

Abstract

In this chapter, optimum design of three-dimensional (3D) framed steel structure subjected to blast loading is considered. The basic idea of this research is to develop a practical formulation for the design optimization problem and to study the effect of including blast loads in the design process. The optimization problem is formulated to minimize the total weight of the structure subjected to American Institution of Steel Construction (AISC) strength requirements and blast design displacement constraints. The design variables for beams and columns are the discrete values of the W-shapes selected from the AISC tables. A car carrying 250 lbs of Trinitrotoluene (TNT) with 50 ft standoff distance from the front face is modeled as the source of the blast loading. Pressure-time histories are calculated on the front, sides, roof, and rear faces of the structure. Then linear and nonlinear dynamic analyses are carried out in the optimization process. Since the problem functions are not differentiable with respect to the design variables, the gradient-based optimization algorithms cannot be used to solve the problem. Therefore, metaheuristic algorithms are used to solve the optimization problem. These algorithms are coded in MATLAB and interfaced with the structural analysis program SAP2000 using its Open Application Programming Interface (OAPI). Example problems are solved to study the formulation of the optimization problem and its solutions. The problems are 4-bay x 4-bay x 3-story frames under serviceability and blast loading. It is shown that penalty on the optimum structural weight is substantial for designing structures to withstand blast loads

5.1 Introduction

As explained in Chapter 2, a small charge explosion could cause catastrophic local or global failure of the structure. The attacks on the World Trade Center in New York City in 1993 and Murrah Federal Building in Oklahoma City in 1995 showed the great damage that could happen due to a blast. In both attacks, structural failure caused more casualties and injuries than the blast wave itself (Cormie, 2009).

The main objective of this work is to present a practical formulation for optimum design of 3D framed steel structures subjected to blast loading. To this end, design variables, cost function, and constraints are studied and explained. The design variables are frame members (beams and columns) which are considered to be discrete (specifically, W-shapes selected from the AISC tables (AISC, 2017)) and are organized in groups based on structural symmetry. The objective function is the total weight of the structure which depends on the discrete design variable values. Constraints are the AISC code strength requirements and DoD (2008) displacement requirements. They are also dependent on the discrete design variables; however, their functional form is not possible. Thus, the gradient-based optimization algorithms are unsuitable for this application because the problem functions cannot be differentiated with respect to the design variables. Therefore, metaheuristics (stochastic) optimization algorithms are used. In these algorithms, gradients are not needed to find an optimum solution. Instead, they search the entire design space for the best solution based on some stochastic strategy. There are many metaheuristics algorithms; however, in this study, Hybrid Harmony Search - Colliding Bodies Optimization (HHC) is utilized to find the minimum weight structure.

MATLAB is used to implement the algorithms and to model design examples by interfacing with the structural analysis program SAP2000 using its Open Application

Programming Interface (OAPI). That is, algorithms start with random design vectors that are sent to SAP2000 for structural analysis; then SAP2000 sends back information needed (nodal displacement, interaction ratio, etc.) to evaluate problem functions. Following this, MATLAB uses this data to arrange and update design vectors using optimization algorithms and then send them back to SAP2000 for re-evaluation.

5.2 Review of Literature

Blast-resistant analysis of structures has been pursued in the literature for many years. Most of the research has focused on dynamic analysis, progressive collapse situations, and members (such as columns, beams or slabs) subjected to blast loading. However, no study is available that considers the optimum design of 3D structures subjected to blast loading.

Stea et al. (1977) presented a report that provided criteria and procedure for the design of framed steel structures subjected to blast loading based on dynamic analysis. Inelastic behavior of the frame members and second order effects were considered in the analysis. Numerical examples were discussed and solved using a FORTRAN computer program called Dynamic Nonlinear Frame Analysis (DYNFA). Lee et al. (2011) studied the dynamic collapse behavior of two moment steel frames using nonlinear finite element analysis. The first model represented a blast and post-blast scenario and the second frame was modeled with a missing column. The study showed that the strain rate should be considered to predict more exact progressive collapse response. Jeyarajan et al. (2015) investigated the response of a 10-story framed steel building subjected high blast pressure using ABAQUS. The source of the blast load was a charge of 500 kg TNT placed at a distance of 20 m from the building. Various lateral bracing systems were studied to show their contributions to progressive collapse analysis. The study showed that higher redundancy in frames could redistribute the damaged members' loads to other floor levels and the vertical displacement

could be reduced. On the other hand, unbraced frame needs rigid beam-column connections to avoid very large displacement due to members' loss. Khaledy et al. (2018) study the optimum design of 2D steel moment frames under blast loading using three techniques: Nonlinear Programming by Quadratic Lagrangian (NLPQL), Particle Swarm Optimization (PSO), and Multi Island Genetic Algorithm (MIGA). The weight of the structure is considered as the objective function and design variables are cross-sectional area of members. Design variables are continuous and other geometrical properties of members are formulated based on polynomial functions of the cross-sectional areas. Only displacement constraints are considered. It is seen that this formulation of the problem is not practical since the members cannot be directly selected from the AISC tables and the design code constraints cannot be imposed.

5.3 Blast Design

Similar to seismic design, it is expected that some of components will experience substantial nonlinear response because designing structures subjected to blast loading to remain elastic is usually uneconomical (ASCE, 2010). Therefore, in designing blast-resistance structures, the maximum dynamic deflection and rotation are the criterion to prevent components failure.

5.3.1 Material Design Strength

Material under high strain rate loadings, such as blast loads, behaves differently from low rate and static loads. Generally, materials become stiffer under high rate loadings which means improvement in their mechanical properties. In addition, in design for blast loads, it is allowed to use the expected actual strength of the material instead of the minimum specified values in blast design.

The high strain rate effect on some mechanical properties of steel is summarized as follows:

- 1- The modulus of elasticity (E_s) remains the same.

2- The yield strength (f_y) and ultimate tensile strength (f_u) increase to the dynamic yield strength (f_{dy}) and the dynamic ultimate strength (f_{du}), respectively.

Dynamic increase factors, DIF , are used to modify the static strength due to high rate dynamic loads (DoD, 2008).

The average yield stress of steel grades 50 ksi or less is about 10% higher than the stress value specified by ASTM. Thus, for blast-resistant design, the yield stress is 1.1 times the minimum yield stress. This factor is called the strength increase factor (SIF) or average strength factor (ASF). SIF should not be used with high strength steels (Gilsanz et al., 2013).

5.3.2 Strength Reduction Factors and Load Combinations

As mentioned above, plastic deformations are allowed in the design of structures subjected to blast loads because of the nature of the blast load and to achieve an economical design. Also, it can use the nominal strength without the strength reduction factor (i.e. $\phi = 1$) for all modes of failure (ASCE, 2011). Blast loads are not combined with the loads that are not expected to be present when the blast happens. That is, wind, earthquake, part or all the live loads are not combined with blast loads. The basic load combination for all construction materials is as follows (ASCE, 2010):

$$1.0 DL + 1.0 LL + 1.0 BL \quad (5.1)$$

where DL is the dead load, LL is live load, and BL is blast load. In the absence of other governing criteria, Gilsanz et al. (2013) allow the following load combination:

$$1.0 DL + 0.25 LL + 1.0 BL \quad (5.2)$$

5.3.3 Performance Requirements

There are many sources for response limits such as UFC 3-340-02 (DoD, 2008), Design of Blast Resistant Buildings in Petrochemical Facilities (ASCE, 2010), FEMA 356 (ASCE and FEMA, 2000), and New York City Building Code (NYCBC, 2008). In this study, design criteria for a structural system are used with a medium response design (ASCE, 2010). That is, the maximum member end rotation shall be 2 degrees and the maximum side-sway deflection (or inter-story drift (*ISD*)) is limited to 1/25 of the story height. To prevent extended structural collapse, beams are allowed to develop plastic hinges when columns are designed to remain elastic (Gilsanz et al., 2013).

5.4 Blast Loading

When a blast occurs in the air, it forces the surrounding air out of its volume it occupies and the air molecules pile-up. A blast wave happens after that and it carries a huge amount of energy (Cormie et al., 2009). The blast wave travels fast and its pressure decays exponentially until it falls to the atmospheric pressure (positive phase). After that, the front wave pressure decreases further to be less than the atmospheric pressure (negative phase) and finally back to ambient value (Figure 5.1). In Figure 5.1, P_{s0} is the peak overpressure or the incident pressure, P_0 is the ambient pressure, and P_{s0}^- is the minimum negative pressure, P_r is the reflected pressure, P_r^- is the minimum negative reflected pressure, t_a is the arrival time, t_o is the positive phase duration, and t_o^- is the negative phase duration.

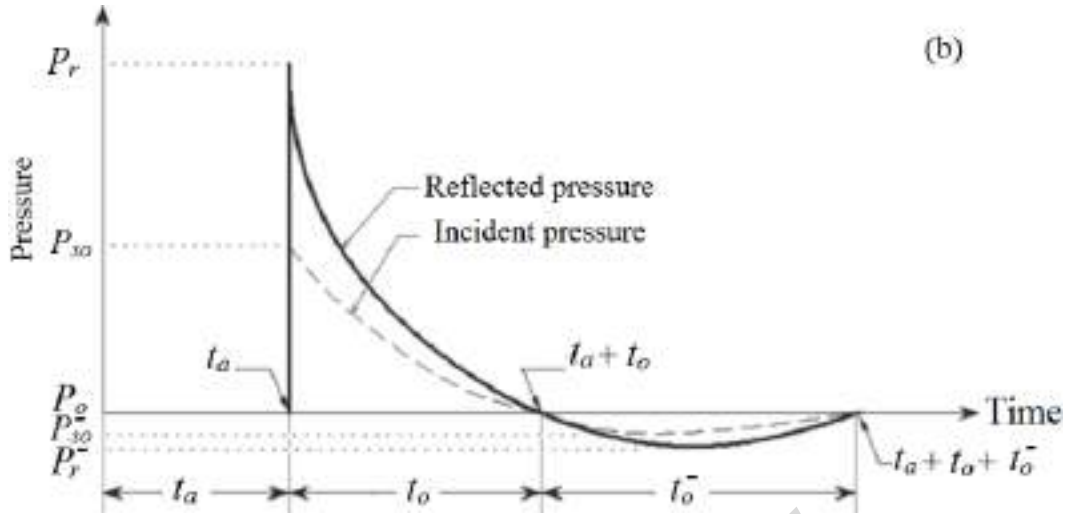


Figure 5.1. Blast wave pressure (Ngo, et al., 2007).

The most commonly used approach for blast wave scaling is Hopkinson-Cranz scaling (or cube-root scaling). It is expressed as follows (Cormie, et al., 2009):

$$Z = \frac{R}{\sqrt[3]{W}} \quad (5.3)$$

where Z is the scaled distance, R is the distance from the detonation source center to the point of interest, and W is the charge mass expressed in pounds of TNT. There are many types of explosives. TNT was chosen to be the blast load source. If another explosive is used, an equivalent TNT weight needs to be computed to use Eq. (5.3); these are provided in conversion tables for different explosives (Cormie, et al., 2009).

Hemispherical burst is considered in this work. It happens when an explosive charge is close to the ground, so the incident wave reflects immediately from the ground and interacts with the blast wave. To find blast wave properties of a hemispherical burst for a given scaled distance (Z), one can use Figure 5.2.

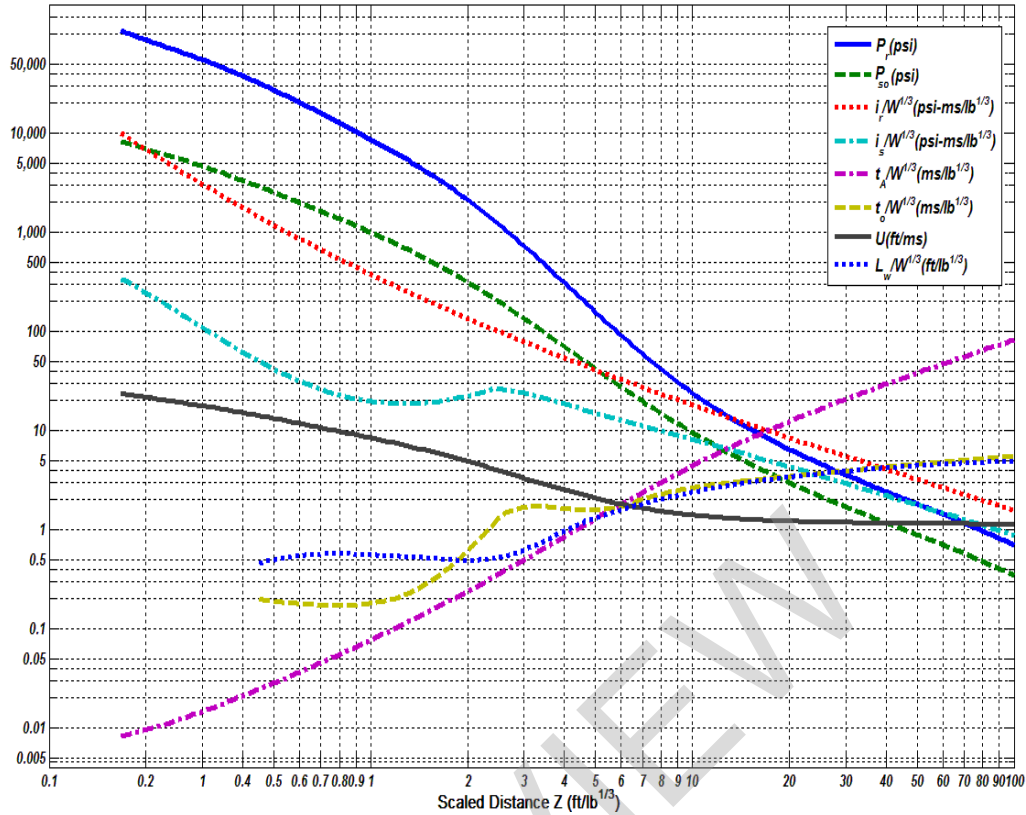


Figure 5.2. The positive phase parameters of hemispherical wave of TNT charges (modified from DoD, 2008).

In Figure 5.2, P_{so} is the incident peak overpressure, P_r is the reflected pressure, i_r is the positive reflected impulse, i_s is the positive incident impulse, t_A is the arrival time, t_o is the positive duration, U is the wave speed, and L_w is the wavelength. They are presented on the y-axis while the x-axis represents the scaled distance Z . Blast loading calculations used in this study follow the methods presented in DoD (2008). For simplicity, a triangular simplification of pressure-time history profile is used and the negative phase is ignored as shown in Section 5.8.2.

5.5 Formulation for Discrete Structural Optimization Problems

The problem is to find American Institute of Steel Construction (AISC) standard W-shape for each member of a framed steel structure and optimize its performance. In general, the nonlinear

undamped dynamic response optimization problem with discrete design variables can be expressed as:

$$\text{Find } \mathbf{X} = [x_1, x_2, \dots, x_{nvar}]; \quad x_i \in D_i; \quad i = 1, 2, \dots, nvar \quad (5.4)$$

$$\text{to minimize } f(\mathbf{X}) \quad (5.5)$$

$$\text{subject to } \mathbf{M}(\mathbf{X})\ddot{\mathbf{u}}(t) + \mathbf{K}(\mathbf{X}, \mathbf{u}(t))\mathbf{u}(t) = \mathbf{p}(t); t = t_1, t_2, \dots, t_n \quad (5.6)$$

$$g_k(\mathbf{X}, \mathbf{u}(t), \dot{\mathbf{u}}(t), \ddot{\mathbf{u}}(t), t) \leq 0; \quad \text{for all } t \text{ and } k = 1, 2, \dots, l$$

where \mathbf{X} is the vector of design variables with $nvar$ unknowns, D_i is a set of discrete values for the i th design variable, $f(\mathbf{X})$ is a cost function (in this study, $f(\mathbf{X})$ is the total weight of the structure), \mathbf{M} is the mass matrix, \mathbf{K} is the stiffness matrix (\mathbf{K} is a function of the design variables and displacement vector for nonlinear dynamic analysis and just the design variables for linear dynamic analysis), \mathbf{u} is the dynamic displacements vector, $\dot{\mathbf{u}}$ is the velocity vector, $\ddot{\mathbf{u}}$ is the acceleration vector, $\mathbf{p}(t)$ is the applied load vector, t is time (generally discretized for numerical integration), n is the total number of the time steps, and g_k is the k th constraint function that needs to be imposed at all time points. The linear dynamic response problem is the same as the nonlinear dynamic response problem except that \mathbf{K} is not a function of the displacement vector \mathbf{u} .

The constrained optimization problem defined in Eqs. (5.4) to (5.6) needs to be transformed into an unconstrained problem so that the metaheuristic algorithms can be used to solve the problem. This can be done by defining a modified cost function $F(\mathbf{X})$ to account for the constraint violations, as follows:

$$F(\mathbf{X}) = f(\mathbf{X})[1 + \psi G(\mathbf{X})]^\xi \quad (5.7)$$

$$G(\mathbf{X}) = \sum_{i=1}^n \sum_{k=1}^l \max(0, g_k(t_i)) \quad (5.8)$$

where $G(\mathbf{X})$ is a constraint violation function, $\psi \geq 1$ is exploration penalty coefficient (in this study, $\psi = 10$), $\xi > 1$ is penalty function exponent (in this study, $\xi = 2$), and $\max(0, g_k(t_i)) \geq 0$ is the violation value of the k th inequality constraint at the time point t_i . The present problem has just inequality constraints.

A linear static response optimization formulation is used in Section 5.8.1 and in equivalent static loads and metaheuristic optimization (see Section 5.6). The linear static response optimization problem subjected to n loading conditions can be stated as:

$$\text{Find } \mathbf{X} = [x_1, x_2, \dots, x_{nvar}]; \quad x_i \in D_i; \quad i = 1, 2, \dots, nvar \quad (5.9)$$

$$\text{to minimize } f(\mathbf{X}) \quad (5.10)$$

$$\text{subject to } \mathbf{K}(\mathbf{X})\mathbf{u}_\alpha = \mathbf{p}_\alpha \quad (5.11)$$

$$g_{k\alpha}(\mathbf{X}) \leq 0; \quad k = 1, 2, \dots, l; \quad \alpha = 1, 2, \dots, n$$

$$F(\mathbf{X}) = f(\mathbf{X})[1 + \psi G]^\xi \quad (5.12)$$

$$G(\mathbf{X}) = \sum_{\alpha=1}^n \sum_{k=1}^p \max(0, g_{k\alpha}) \quad (5.13)$$

where \mathbf{X} is the vector of design variables with $nvar$ unknowns, D_i is a set of discrete values for the i th design variable, $f(\mathbf{X})$ is a cost function (in this study, $f(\mathbf{X})$ is the total weight of the structure), \mathbf{K} is the stiffness matrix, \mathbf{p}_α is the α th loading condition, k is the total number of constraints, n is the total number of the loading conditions (time steps), $F(\mathbf{X})$ is a modified cost function, $G(\mathbf{X})$ is a constraint violation function, $\psi \geq 1$ is exploration penalty coefficient (in this

study, $\psi = 10$), $\xi > 1$ is penalty function exponent (in this study, $\xi = 2$), and $\max(0, g_k(t_i)) \geq 0$ is the violation value of the k th inequality constraint of the α th loading condition.

5.5.1 Design Variables

In this study, the AISC (2017) W-shapes available in manufacturer's catalog are desired for beams and columns. Since all sections are chosen from AISC tables and assignment of a section specifies several cross-sectional properties for the member, the design variables are classified as linked discrete variables (Arora, 2017).

Huang and Arora (1997) defined three types of discrete design variables for this kind of a problem; each one requiring a specific optimization strategy. In this study, design variable type 3 is appropriate. That is, one design variable is assigned for each member (the AISC section number). Once the section number is known, all the cross-sectional properties are known from the tables. This way the design variables Eqs. (5.1) and (5.2) become:

$$\text{Find } \mathbf{X} = [S_1, S_2, \dots, S_{nvar}] \quad (5.14)$$

$$S_{imin} \leq S_i \leq S_{imax} \quad (5.15)$$

where S_i is an AISC W-shape number, $i \in [1, 2, \dots, nvar]$, S_{imin} and S_{imax} are the lightest and the heaviest sections, respectively. In numerical calculations, the W-shapes from the AISC table are re-arranged in an ascending order based on their weights.

5.5.2 Cost Function

The cost function is the criterion that is used to compare feasible designs to find the optimum solution (Arora, 2017). In this study, the problem is to minimize the total weight of the structure (in kips). Thus, Eqs. (5.5) and (5.10) become:

$$W_s(\mathbf{X}) = \sum_{ng=1}^{NG} w_{ng} \sum_{mk=1}^{MK} L_{mk} \quad (1)$$

where W_s is the total weight of the structure, \mathbf{X} is the design vector, NG is the total number of member groups for the structure, w_{ng} is the weight per unit length (kips/ft) of the members in the ng th group (available in AISC tables), MK is the number of members in the ng th group, and L_{mk} is the length of the mk th member (ft).

5.5.3 Constraints

Restrictions imposed on the structural members are: the strength requirements given in AISC manual, inter-story displacement constraints, and geometrical requirements. These constraints are implicit functions of the design variables and are explained in the following paragraphs.

5.5.3.1 Strength Constraints

According to the AISC (2017), symmetric members subjected to axial force and bending must satisfy the interaction ratio and shear force strength requirements:

$$\frac{P_u}{\phi P_n} + \frac{8}{9} \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right) - 1 \leq 0 \quad \text{if } \frac{P_u}{\phi P_n} \geq 0.2 \quad (5.16)$$

$$\frac{P_u}{2\phi P_n} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right) - 1 \leq 0 \quad \text{if } \frac{P_u}{\phi P_n} < 0.2$$

$$V_u \leq \phi_v V_n$$

$$\frac{V_u}{\phi_v V_n} - 1 \leq 0 \quad (5.17)$$

Here ϕ is the resistance factor ($\phi_c = 0.85$ and $\phi_t = 0.90$ for compression and tension, respectively).

$\phi_b = 0.9$ is the flexural resistance factor. P_u and P_n are the required and the nominal axial strengths

(compression or tension) (kips), respectively. M_{ux} and M_{uy} are the required flexural strengths about the major and the minor axes (kip-ft), respectively. M_{nx} and M_{ny} are the nominal flexural strengths about the major and the minor axes (kip-ft), respectively. M_u and M_n are required and the nominal flexural strengths about major or minor axes. V_u and V_n are required and the nominal shear strengths (kips), respectively. $\phi_v = 0.9$ is the resistance factor for shear.

Evaluation of P_n , M_{nx} and M_{ny} in Eqs. (5.16) is an involved process that requires checking of several failure modes (i.e., several “if then else” statements). For example, to find P_n , first one needs to find whether the member force is tensile or compressive. For tension members, P_n is calculated based on whether the gross section yields or the net section ruptures. For compression members, P_n is calculated based on consideration of several failure modes, such as yielding of the material, local buckling of flanges or the web (elastic or inelastic), and global buckling (elastic or inelastic). The nominal strength for flexure of major or minor axis bending (M_{nx} or M_{ny} in Eq. (5.16)) depends on categorization of the member as compact, noncompact, or slender. Compact sections can develop full plastic strength before local buckling happens. Plastic moment (yielding) and lateral-torsional buckling are considered in calculating M_n . Noncompact sections can develop partial yielding in compression but they buckle inelastically before full plastic strength. Lateral-torsional buckling and compression flange local buckling are considered in calculating M_n . Slender sections buckle elastically before yield under compression. Compression flange yielding, lateral-torsional buckling, and compression flange local buckling are considered in calculating M_n . The nominal strength for flexure of minor axis bending is calculated considering plastic moment and flange local buckling. V_n in Eq. (5.17) is calculated according to the limit states of shear yield and shear buckling. That is, nominal strengths (P_n , M_{nx} , M_{ny} and V_n) require a lot of calculations that can be found in AISC (2017).

Constraints in Eqs. (5.16), and (5.17) need to be imposed at each point along the axis of every member in the structure. Thus, each equation represents infinite constraints. In the numerical process, the constraints are evaluated at several points along the axis of the member and they are imposed at the point where they have maximum value. These constraint values are then used to evaluate the penalty function defined earlier in the chapter. Thus, the total number of interaction ratio constraints (Eq. (5.16)) equals the total number of members. Same is true for shear force constraints (Eq. (5.17)). Also notice that constraints in Eq. (5.16) have a discontinuity at $\frac{P_u}{\phi P_n}=0.2$. In addition, the nominal strength calculations have several discontinuities as explained in the previous paragraph. That is, gradient-based optimization algorithms are not suitable for this class of optimization problems.

5.5.3.2 Displacement Constraints

The maximum member end rotation shall be 2 degree and the maximum side-sway deflection (or inter-story drift (*ISD*)) is limited to 1/25 of the story height (high response design (ASCE, 2010)).

$$\frac{|\delta_r - \delta_{r-1}|}{\delta_{ru}} - 1 \leq 0 \quad (5.18)$$

$$\delta_{ru} = h_r/25 \quad (5.19)$$

where δ_r and δ_{r-1} are lateral displacements of two adjacent stories (in), δ_{ru} is the allowable lateral displacement, and h_r is the r th story height (in). At each node, SAP2000 evaluates displacements and rotations in 3-dimensions. Displacements in x and y directions are extracted to evaluate Eq. (5.19) to impose these constraints.

5.6 Discrete Variable Optimization of Structures Subjected to Dynamic Loads Using Equivalent Static Loads

Design of structures subjected to blast loads requires nonlinear dynamic analysis (as described in Section 5.4). Depending on the size of the structure to be designed, the nonlinear dynamic analysis (numerical integration of system of nonlinear differential equations) might need very long time. Metaheuristic algorithms require many structural analyses to reach the final design. Using metaheuristic algorithms could be impractical for this type of a problem. Therefore, optimization by transforming dynamic to static loads is more efficient.

One of the well-known dynamic to static loads transformation methods is based on the displacement field obtained using dynamic analysis of the structure (Kang et al., 2001). That is, the dynamic load is transformed into multiple equivalent static load sets. Then the equivalent static loads (ESLs) are considered as multiple loading conditions in the linear static response optimization process. This is called an ESL cycle of the optimization process. These cycles are repeated until the final design is obtained. This method works fine for gradient-based algorithms (Kang et al., 2001) and it is shown that ESL with metaheuristic algorithm (called GOESL) can reduce the number of linear or nonlinear dynamic analyses drastically (Chapter 5). Thus, GOESL is used in this study to optimize nonlinear dynamic problems.

5.7 Optimization Algorithms

Stochastic, metaheuristic or nature-inspired algorithms are based only on simulations and do not require gradient information, such as the well-known genetic algorithms (GA) and ant colony methods (AC). They use random search in the entire design space instead of just in the neighborhood of the current design as in the gradient search techniques. Also, the discrete variables

can be treated routinely. Therefore, they are suitable for both continuous and discrete design variables and differentiable and non-differentiable problem functions.

In this study, Hybrid Harmony Search - Colliding Bodies Optimization (HHC) is utilized to find an optimum design of every case of study. HHC uses two phases: first phase of HHC uses the Improved Harmony Search (IHS) algorithm with a new design domain reduction technique. This improves the performance of IHS. The second phase uses Enhanced Colliding Bodies Optimization (ECBO). ECBO receives final designs from the first phase to enhance them further (see Chapter 4).

5.7.1 Improved Harmony Search

Geem, Kim, and Longanathan (2001) developed the harmony search (HS) algorithm based on music improvisation process of jazz musicians. The algorithm starts by initially generating a set of random designs from the design domain. Then in every iteration, a new design is generated and analyzed. If this design is better than the worst design in the current population, then it replaces that design; otherwise, another design is generated. This optimization process is continued until a limit on the number of iterations is reached. HS has 4 parameters that need to be turned on before starting the algorithm: harmony memory size (*HMS*), harmony memory consideration ratio (*HMCR*), pitch adjusting rate (*PAR*), and maximum improvisations (or maximum number of iterations).

Improved harmony search (IHS) is the same as HS, however, standard HS algorithm uses fixed value of *HMCR* and *PAR* while in IHS *HMCR* and *PAR* are adjusted with every iteration. The main drawback of the standard HS algorithm is that it needs a large number of iterations to find an acceptable solution (Mahdavi et al., 2007).

5.7.2 Enhanced Colliding Bodies Optimization (ECBO)

This metaheuristic algorithm is developed by Kaveh and Mahdavi (2014). It is inspired by the laws of one-dimensional collision. The algorithm works with a population of designs at each iteration. It starts with random designs that are stored in a matrix called the colliding bodies' matrix (**CB**). Each design in the population is considered as an object or body having pseudo-mass that is calculated using the merit function value for each design. Then the entire population is ranked and divided into stationary objects and moving objects. One dimensional collision between the bodies is simulated using the conservation law of linear momentum and the coefficient of restitution. Based on that, new velocities of the stationary and moving objects are evaluated. Each design in the population is updated using the new velocities and random numbers. This process is repeated until a limit on the iterations is reached.

The enhanced version of the colliding bodies optimization (ECBO) uses a colliding memory matrix (**CM**) to store some good designs. These designs replace the worst designs in the **CB** matrix at every iteration. This way the good designs are always preserved. Also, a parameter $Pro \in [0,1]$ is introduced that is used along with random numbers to regenerate a component of selected designs in the **CB** matrix. This mechanism leads to a better final design (Kaveh & Ghazaan, 2014).

Many metaheuristic algorithms need selection of several parameters in their calculations which is a major drawback of these algorithms. ECBO, however, is simple, requires just one internal parameter, and performs well in term of the quality of the solution and convergence time (Kaveh & Mahdavi, 2015).

5.7.3 Hybrid Improved Harmony Search-Enhanced Colliding Bodies Algorithm (HHC)

Compared to other metaheuristic algorithms, IHS is easy to implement and it works with any kind of problem. ECBO requires just one algorithmic parameter and it performs well in term of the quality of final designs. However, IHS and ECBO have some shortcomings that were observed while solving some problems. IHS requires specification of several algorithmic parameters that can affect the performance of the algorithm. ECBO makes steady progress towards the final design whereas IHS makes quite rapid progress towards a similar neighborhood. Therefore, IHS needs fewer simulations compared to ECBO to reach a neighborhood of the final design. However, after reaching the neighborhood of the final design, progress of IHS becomes slow to reach the final design whereas ECBO continues to make steady progress towards the solution.

HHC algorithm uses IHS in Phase 1 to reach the neighborhood of the solution quickly and then switches to the ECBO to reach the final design. This way ECBO starts with some improved designs in Phase 2. This combination could lead to the final solution in fewer simulations which is very useful while solving more complex problems.

5.8 Numerical Examples

In this study, HHC is applied for optimum design of 3D framed steel structures. Phase 1 parameters, r_1 , r_2 , and r_3 are 25%, 10%, and 10%, respectively. When there is no or small improvement in the current function value, Phase 2 is terminated using a stopping criterion similar to the one used in Phase 1. The pseudo-code of this criterion is as follows:

$$\text{If}_1 \text{ } Iter_{p2} \geq r_4 \times \text{MaxIter}_{p2}$$

$$\text{If}_2 (\text{Merit}(Iter_{p2}) - \text{Merit}(Iter_{p2} - r_5 \times \text{MaxIter}_{p2})) / \text{Merit}(Iter_{p2}) \leq \varepsilon_{p2}$$

Terminate Phase 2

End₂

End₁

where $Iter_{p2}$ is the current iteration in Phase 2, $MaxIter_{p2}$ is the limit on number of iterations of Phase 2, $r_4=10\%$, $r_5=5\%$ and $\varepsilon_{p2}=10^{-3}$. These parameters are selected so that premature termination of the algorithm does not occur.

As mentioned earlier design variables are W-shapes selected from the AISC tables. The design variables are linked discrete variables. That is, when the section number is selected, all the cross-sectional properties are known from the AISC tables.

Figure 5.3 shows 3D view and the dimensions of the structure (slabs and external walls are not shown). It is a 3-story, 4 bays in x and y directions with 4 in concrete slab consisting of 197 members modeled using SAP2000 and MATLAB. All ground supports are fixed. Steel properties are: Young's modulus, $E=29000$ ksi, yield stress, $F_y=50$ ksi, ultimate strength, $F_u=65$ ksi, and Poison's ratio, $\nu=0.3$. Concrete properties are: Young's modulus, $E=3605$ ksi, density, $\rho=150$ lb/ft³, $f'_c=4000$ psi, and Poison's ratio, $\nu=0.2$.

The frame members are divided into 9 groups as shown in Figure 5.3 and Table 5.1. Each group is treated as a design variable. Gravity loads are assigned as uniformly distributed loads on the first and second floor slabs consisting of a design dead load of 60 psf and a design live load of 50 psf and on the roof slab consisting of a design dead load of 60 psf and a design live load of 25 psf. Table 5.2 gives design load combinations.

Since the algorithm is stochastic in nature, three independent optimization runs were performed for each case study. Design variables bounds are selected based on testing each problem with different design variables values.

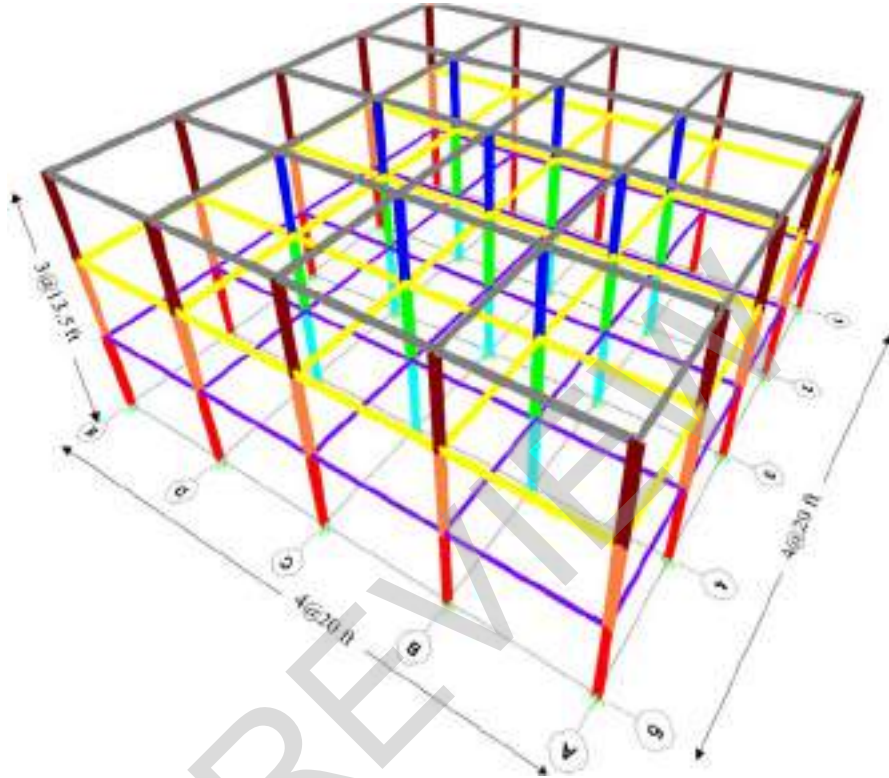


Figure 5.3. Schematic of 3D framed steel structure.

Table 5.1. Members grouping.

Group Number	Members	Number of members
1	1st floor external columns	16
2	2nd floor external columns	16
3	3rd floor external columns	16
4	1st floor internal columns	9
5	2nd floor internal columns	9
6	3 floor internal columns	9
7	1st floor beams	40
8	2nd floor beams	40
9	3rd floor beams	40

Table 5.2. Load combinations (AISC, 2015; Gilsanz et al., 2013).

Load combination	Scale factor
Comb 1	1.2 DL + 1.6 LL
Comb 2	1.4 DL
Comb 3	1.0 DL + 0.25 LL + 1.0 BL

DL is dead load, LL is live load, and BL is blast load.

5.8.1 Spatial Framed Steel Structure Subjected to Service Loads Only

In this design example, only the service loads (no blast loads) and strength constraints are considered (only load combinations Comb 1 and Comb 2 in Table 5.2 are used). It is used as reference to compare designs with examples and to study the penalty of designing framed steel structure to resist blast loads.

Columns and beams are selected from the first 100 lightest standard W-shape sections provided in AISC tables (AISC, 2017) after rearranging sections in an ascending order based on their weight. This example is solved using linear static analysis (direct stiffness method).

The final designs for the three runs are reported in Table 5.3 along with total structural weight and maximum values of interaction and shear ratios. The second run gives the best design with a total structural weight of 60.549 kips.

Even though the number of the possible combination is tremendously large (9^{100} , HHC domain adjustment technique reduces the possible combination by reducing design variables bounds in Phase 1 (see Section 3.5.1). In this design example, HHC was able to obtain designs that have similar structural weight. That is, the first design is about 11% heavier than the best design and the third design is about 10% heavier than the best design.

Table 5.3. Final designs for 3D framed steel structure under service load only.

Design variable (group number)	Run		
	1	2	3
1	W8X31	W8X28	W21X44
2	W12X26	W8X24	W14X34
3	W6X25	W10X22	W6X25
4	W18X50	W14X48	W12X40
5	W8X40	W16X36	W16X36
6	W10X22	W10X26	W16X31
7	W5X16	W4X13	W4X13
8	W5X16	W4X13	W4X13
9	W6X15	W4X13	W4X13
Max. interaction ratio (Eq. (5.16))	0.984	0.979	0.993
Max. shear ratio (Eq. (5.17))	0.270	0.191	0.202
Weight (kips)	67.665	60.549	66.449
No. of linear static analyses	11946	11740	11752

To investigate the effectiveness of HHC for this type of problems, the algorithm was run for 45,035 structural analyses (9,000 iterations for the first phase and 900 iterations for the second phase with population size of 40). This is more than 3 times the number iterations needed by each of the 3 runs above. For this run, the best structural weight obtained is 64.496 kips. This design is 6.5% heavier than the best design in Table 5.3. Studying the designs in HM matrix (first phase) and CB matrix (second phase) shows that:

- 1- HM matrix has diverse designs but it stops improving design after about 3000 iteration.
- 2- The second phase stops improving designs after about 200 iterations and CB matrix starts to have less diverse designs.

5.8.2 Spatial Framed Steel Structure Subjected to blast Load

This design example has the same dimensions, design variables and material properties as the previous study case. However, in addition to gravity loads and load combinations described in the previous section, the blast loading is considered. The source of the blast loads is an automobile carrying a large charge of 250 lb of TNT. The structure has a stand-off distance of 50 ft from the charge's center as shown in Figure 5.4. The structure is isolated with no opening (conservative

assumption). In real cases, however, there are windows and door openings that (if not designed to resist blast loading) vent some of the blast wave inside the building depending on the size of those opening. The façade of the structure is divided into 12 panels and the blast reflected pressure is evaluated at the center of each panel and distributed uniformly on that panel as shown in Figure 5.4 (Karlos and Solomos, 2013). Side, roof, and rear blast loads are calculated at the center of each face and distributed uniformly on the surface. Table 5.4 shows the pressure-time history on the front, sides, roof, and rear faces. Using *SIF* and *DIF* parameters (Section 5.3.1), the updated material strength values are as follows:

$$F_{df} = (SIF)(DIF)F_y = (1.1)(1.19)(50) = 65.45 \text{ ksi} \quad (5.20)$$

$$F_{du} = (DIF)(F_u) = (1.05)(65) = 68.25 \text{ ksi} \quad (5.21)$$

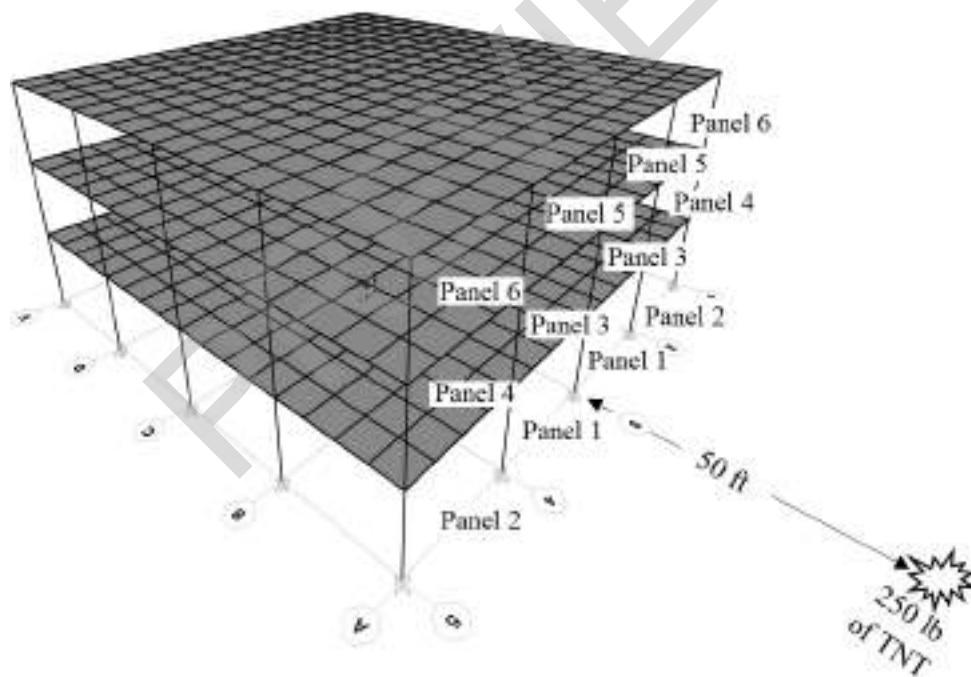
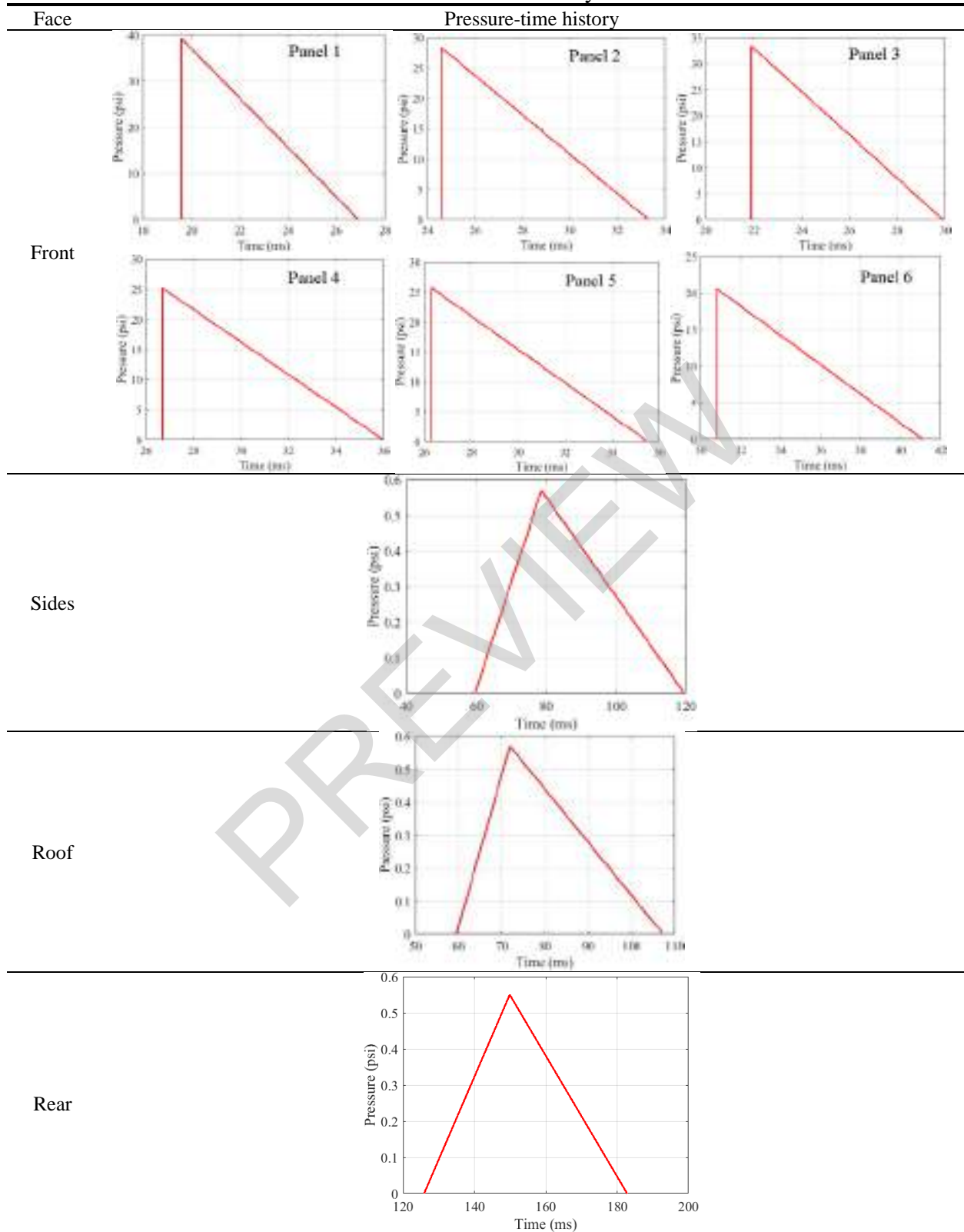


Figure 5.4. 3D framed steel structure and charge location.

Table 5.4. Pressure-time history on faces.



5.8.2.1 Optimum Design with Linear Dynamic Analysis

In this study case, beams and columns are designed according to AISC (2017) strength requirements (Section 5.5.3.1). That is, all members are designed to remain elastic. The following examples are solved using Hilber-Hughes-Taylor method (linear direct integration) and the total analysis time is 1 second with time step of 0.0025. The analysis time was selected after different designs indicated that the maximum response occurs between 0 to 1 second.

In calculating blast loads on the structure surfaces, the assumption was that all surfaces are rigid enough to reflect the pressure with no energy dissipation. Then blast loads are transported to beams and columns as distributed loads. The common design approach is to neglect the outer periphery walls in the analysis model. In this study, three approaches are investigated.

5.8.2.1.1 No External Walls

In this example, the stiffness of the external walls is not considered. This conservative procedure is used in most of blast design references such as AISC (2013). Columns and beams are selected from the first 173 heaviest standard W-shape sections provided in AISC tables (AISC, 2017) after rearranging sections in a descending order based on their weight.

The final designs for the three runs are reported in Table 5.5 along with total structural weight and maximum values of the interaction and shear ratios. It shows that the second run reaches the best design with a total structural weight of 853.469 kips. This is about 14 times heavier than the best design found for same structure subjected to gravity loads only.

Table 5.5. Final designs for 3D framed steel structure under service and blast loads (linear dynamic analysis).

Design variable (group number)	Run		
	1	2	3
1	W36X262	W44X262	W24X229
2	W40X277	W12X210	W14X211
3	W14X283	W21X201	W33X118
4	W30X148	W14X211	W14X159
5	W14X233	W18X158	W18X175
6	W14X311	W12X152	W36X150
7	W36X262	W12X336	W40X362
8	W14X211	W33X241	W12X305
9	W36X262	W24X229	W27X194
Max. interaction ratio (Eq. (5.16))	0.994	0.996	0.968
Max. shear ratio (Eq. (5.17))	0.400	0.456	0.518
Weight (kips)	892.988	853.469	868.134
No. of linear dynamic analyses	15125	15102	14259

5.8.2.1.2 No External Walls with Mass

In this example, the stiffness of the external walls is not considered. However, the mass of the outer periphery walls (thickness of 4 in) is added as a dead load on beams. Columns and beams are selected from the first 173 heaviest standard W-shape sections provided in AISC tables (AISC, 2017) after rearranging sections in a descending order based on their weight.

The final designs for the three runs are reported in Table 5.6 along with total structural weight and maximum values of the interaction and shear ratios. It shows that the first run reaches the best design with a total structural weight of 827.182 kips. This is about 13.7 times heavier than the best design found for same structure subjected to gravity loads only.

Table 5.6. Final designs for 3D framed steel structure under service and blast loads with the mass of external walls (linear dynamic analysis).

Design variable (group number)	Run		
	1	2	3
1	W36X232	W27X235	W36X247
2	W12X279	W18X158	W24X192
3	W33X169	W14X132	W24X229
4	W40X211	W27X146	W30X173
5	W18X234	W14X145	W40X211
6	W40X183	W40X211	W30X173
7	W36X330	W40X297	W33X291
8	W36X247	W40X324	W36X302
9	W27X178	W33X201	W24X192
Max. interaction ratio (Eq. (5.16))	1.000	1.001	0.997
Max. shear ratio (Eq. (5.17))	0.421	0.459	0.461
Weight (kips)	827.182	831.993	839.964
No. of linear dynamic analyses	16055	17955	16985

5.8.2.1.3 With External Walls

The stiffness of the external walls that are connected to the framed steel structure is added to the structural model. The outer periphery wall is a concrete wall with a thickness of 4 in. The outer periphery wall is pinned to the ground and attached to external beams only which are attached to the roof system to transfer loads directly into floor diaphragms to reduce the risk of progressive collapse as recommended in ASCE (2011). Columns and beams are selected from the first 100 lightest standard W-shape sections provided in AISC tables (AISC, 2017) after rearranging sections in an ascending order based on their weight.

Table 5.7 gives the final designs of three optimization runs. The best design has a total structural weight of 77.818 kips. This design is about 28.5% heavier than the best design found for same structure subjected to gravity loads only.

The external walls add quite amount of stiffness to the structure. They act as shear walls that resist the lateral blast loads. This is unconservative design approach. Although, the structure quite lighter than the previous case study, adding wall stiffness in the model makes the structure less robust to progressive collapse when a wall fails during the blast event.

Table 5.7. Final designs for 3D framed steel structure with external walls under service and blast loads (linear dynamic analysis).

Design variable (group number)	Run		
	1	2	3
1	W8X40	W16X40	W12X35
2	W18X40	W18X40	W18X40
3	W18X40	W18X40	W16X31
4	W10X26	W12X26	W8X24
5	W8X24	W14X26	W8X24
6	W6X15	W6X25	W10X22
7	W6X20	W5X19	W5X19
8	W6X20	W6X20	W8X24
9	W6X15	W6X15	W5X19
Max. interaction ratio (Eq. (5.16))	0.9345	0.977	0.997
Max. shear ratio (Eq. (5.17))	0.161	0.161	0.235
Weight (kips)	77.818	78.476	81.001
No. of linear dynamic analyses	11804	12518	11922

5.8.2.2 Optimum Design with Nonlinear Dynamic Analysis

In this study case, columns are designed according to AISC (2017) strength requirements (Section 5.5.3.1) but beams can develop plastic hinges and blast design requirements are applied (Section 5.5.3.2). Steel Columns-Flexure elastic-perfectly plastic hinges provided by SAP2000 v.20 are modeled near the start and the end of each beam (CSI, 2017). The following examples are solved using Hilber-Hughes-Taylor method (Nonlinear direct integration with P-delta). Considering the blast load duration and peaks of the response, the time range from the analysis is set from 0 to 1 second with time step of 0.0025 (Similar to linear dynamic analysis study case).

Testing the nonlinear dynamic models (Sections 5.8.2.2.1 and 5.8.2.2.2) with different designs shows that in most cases there is either numerical convergence difficulty or the nonlinear structural analysis takes long time because of the material nonlinearity, geometrical nonlinearity, and the size of the structure. This makes metaheuristic algorithms inconvenient to use since they require many structural analyses to obtain the best design. Therefore, ESL1 method (Section 5.6) is used. The algorithm starts with 75 random designs from the design domain. These designs are evaluated using nonlinear dynamic analyses, ranked in an ascending order based on their merit function values, and passed to the ESL step. In the ESL step, HHC is used to find the best design

using linear static analyses (note that in Chapter 4 ECBO is used). After few ESL cycles, the best 2 designs in **CM** matrix and the best 20 designs in **CB** matrix are improved further using ECBO with just 10 iterations using nonlinear dynamic analyses (total of 200 analyses). That is, ECBO has a population size of 20.

5.8.2.2.1 No External Walls

This study case is like the one in Section 5.8.2.1.1. That is, the stiffness of the external walls is not considered. However, beams can develop plastic hinges as discussed in the previous section. Beams are selected from the first 100 lightest standard W-shape sections and columns are selected from the heaviest 173 standard W-shape sections provided in AISC tables (AISC, 2017) after rearranging sections in an ascending order based on their weight.

The final designs are shown in Table 5.8 along with total structural weight and maximum values of interaction ratio, shear ratio, member end rotation, and inter-story drift. The first run reaches the best design with a total structural weight of 386.148 kips. This design is about 6 times heavier than the best design found for same structure subjected to gravity loads only.

Table 5.8. Final designs for 3D framed steel structure under service and blast loads (nonlinear dynamic analysis).

Design variable (group number)	Run		
	1	2	3
1	W24X370	W33X263	W36X262
2	W27X235	W30X211	W40X211
3	W36X232	W40X324	W24X306
4	W30X116	W33X291	W44X290
5	W40X362	W36X395	W40X392
6	W27X258	W30X391	W33X387
7	W8X31	W10X30	W16X31
8	W12X45	W16X57	W21X57
9	W8X35	W8X21	W12X22
Max. interaction ratio (Eq. (5.16))	1.000	0.995	0.995
Max. shear ratio (Eq. (5.17))	0.910	0.938	0.796
Max. rotation (degree)	0.718	0.594	1.865
Max. ISD (Eq. (5.18))	0.199	0.164	0.303
Weight (kips)	359.016	389.624	386.148
Total number of linear static analyses	43660	45340	46530
No. of nonlinear dynamic analyses	205	205	205

5.8.2.2.2 No External Walls with Mass

This study case is similar to Section 5.8.2.1.25.8.2.2.1. However, the mass of the external walls is considered but their stiffness is not considered. Columns and beams are selected from the first 173 lightest standard W-shape sections provided in AISC tables (AISC, 2017) after rearranging sections in the descending order based on their weight.

The final designs are shown in Table 5.9 along with total structural weight and maximum values of interaction ratio, shear ratio, member end rotation, and inter-story drift. The first run reaches the best design with a total structural weight of 399.215 kips. This design is about 6.6 times heavier than the best design found for same structure subjected to gravity loads only.

Table 5.9. Final designs for 3D framed steel structure under service and blast loads with the mass of the external walls (nonlinear dynamic analysis).

Design variable (group number)	Run		
	1	2	3
1	W27X368	W36X232	'W40X264'
2	W27X281	W30X357	'W40X362'
3	W36X231	W40X503	'W27X217'
4	W33X152	W36X194	'W40X372'
5	W40X397	W40X149	'W40X362'
6	W27X307	W33X152	'W36X361'
7	W8X40	W10X54	'W14X74'
8	W12X53	W18X35	'W12X45'
9	W8X40	W8X40	'W12X53'
Max. interaction ratio (Eq. (5.16))	0.823	0.988	0.987
Max. shear ratio (Eq. (5.17))	0.905	0.960	0.940
Max. rotation (degree)	0.326	0.323	0.285
Max. ISD (Eq. (5.18))	0.311	0.306	0.460
Weight (kips)	400.484	399.215	452.731
Total number of linear static analyses	37575	43025	41150
No. of nonlinear dynamic analyses	205	205	205

5.8.2.2.3 With External Walls

This study case is similar to Section 5.8.2.1.3. That is, the stiffness of the external walls is considered. Columns and beams are selected from the first 100 lightest standard W-shape sections provided in AISC tables (AISC, 2017) after rearranging sections in an ascending order based on their weight.

The best designs are shown in Table 5.10 along with total structural weight and maximum values of interaction ratio, shear ratio, member end rotation, and inter-story drift. The third run obtains the best design with a total structural weight of 77.626 kips. This design is about 28% heavier than the best design found for same structure subjected to gravity loads only.

Table 5.10. Final designs for 3D framed steel structure with external walls under service and blast loads (nonlinear dynamic analysis).

Design variable (group number)	Run		
	1	2	3
1	W14X34	W10X39	W14X38
2	W8X35	W16X36	W8X35
3	W8X35	W8X35	W12X35
4	W8X21	W8X24	W10X22
5	W8X21	W10X22	W12X22
6	W6X20	W14X22	W8X21
7	W14X22	W8X18	W10X17
8	W8X21	W16X26	W6X25
9	W8X18	W6X15	W12X16
Max. interaction ratio (Eq. (5.16))	0.984	0.966	0.988
Max. shear ratio (Eq. (5.17))	0.801	0.725	0.350
Max. rotation (degree)	0.415	0.267	0.625
Max. ISD (Eq. (5.18))	0.177	0.184	0.176
Weight (kips)	78.797	79.222	77.626
Total number of linear static analyses	30749	29425	30444
No. of nonlinear dynamic analyses	205	205	205

5.9 Concluding Remarks

Optimum design of 3D framed steel structures subjected to service and blast loads are studied using metaheuristic optimization algorithms. The problem is formulated to minimize total weight of the structure subjected to AISC strength requirements and DoD displacement constraints. The design variables for beams and columns are W-shapes selected from the AISC tables. Depending on the problem, three types of analyses are carried out in the optimization process: linear static analysis, linear dynamic analysis, and nonlinear dynamic analysis. Hybrid Harmony Search - Colliding Bodies Optimization (HHC) with domain adjustment for each design variable is used to find an optimum design of every case of study. Global optimization with

equivalent static load (GOESL) is used to find the optimum design for the nonlinear dynamic study case.

Table 5.11 shows final designs of the structure with and without blast loading. It is seen that when beams and columns are designed to remain elastic, the optimum structure is about 14 times heavier to withstand the blast loads compared to the optimum design without the consideration of blast loading. However, when columns are designed to remain elastic and beams can develop plastic hinges (with displacement requirements), the optimum structure is about 6 times heavier to withstand the blast loads compared to the optimum design without the consideration of blast loading. When the stiffness of the external walls is considered, the final designs for linear and nonlinear dynamic analyses are only slightly heavier than the design without the blast load considerations.

Table 5.11. Comparison of final designs.

Design variable (group number)	Service load only using linear static analysis	Service and blast loads using linear dynamic analysis			Service and blast loads using nonlinear dynamic analysis		
		No external walls	No external walls with mass	With external walls	No external walls	No external walls with mass	With external walls
1	W8X28	W44X262	W36X232	W8X40	W24X370	W36X232	W14X38
2	W8X24	W12X210	W12X279	W18X40	W27X235	W30X357	W8X35
3	W10X22	W21X201	W33X169	W18X40	W36X232	W40X503	W12X35
4	W14X48	W14X211	W40X211	W10X26	W30X116	W36X194	W10X22
5	W16X36	W18X158	W18X234	W8X24	W40X362	W40X149	W12X22
6	W10X26	W12X152	W40X183	W6X15	W27X258	W33X152	W8X21
7	W4X13	W12X336	W36X330	W6X20	W8X31	W10X54	W10X17
8	W4X13	W33X241	W36X247	W6X20	W12X45	W18X35	W6X25
9	W4X13	W24X229	W27X178	W6X15	W8X35	W8X40	W12X16
Max. interaction ratio	0.979	0.996	1.000	0.9345	1.000	0.988	0.988
Max. shear ratio	0.191	0.456	0.421	0.161	0.910	0.960	0.350
Max. rotation (degree)	-	-	-	-	0.718	0.323	0.625
Max. ISD	-	-	-	-	0.199	0.306	0.176
Weight (kips)	60.549	853.469	827.182	77.818	359.016	399.215	77.626

CHAPTER 6

OPTIMUM DESIGN OF FRAMED STRUCTURES FOR RESILIENCE SUBJECTED TO BLAST LOADS

Abstract

In this study, a formulation is presented for the optimum design of 3D framed structures that can withstand some future damage due to a blast near the structure. That is, after the blast event occurs the structure should still carry the service loads (or at least some part of them) so that no further damage can happen beyond the designed damage conditions. A least weight structure is desired that also meets the American Institution of Steel Construction (AISC) strength requirements and displacement constraints. In addition, the formulation includes some possible future damage to the structures due to a blast. The possible damage conditions are defined as complete removal of certain members and reduction of stiffness of some members. The design variables for beams and columns are the discrete values of the W-shapes selected from the AISC tables. Since the cost function and the constraints are not differentiable with respect to the discrete design variables, the gradient-based optimization algorithms cannot be used to solve the problem. Therefore, metaheuristic optimization algorithms are used to find optimum or near optimum designs. As an example, problem, a 4-bay x 4-bay x 3-story framed steel building under serviceability loading. Several different scenarios of damage to the structure are considered and the optimum designs from Chapter 5 are checked.

6.1 Introduction

It is desirable to design structures to minimize a measure of cost while all performance requirements are satisfied. When a structure is designed by application of an optimal design technique, it is expected that many performance constraints for the structure are at their limit values

(or close to them). Therefore, a small change in the performance environment may cause a catastrophic failure of the structure. A damage-tolerant structure must avoid failure by providing an alternate load path(s). In other words, fail-safe structures must have enough redundancy to withstand possible damages and perform normally even when a member fails.

In this study, some possible damage conditions of building structure due to blast loads are studied. When blast happens, it generates hot gas that makes the air around the explosion expand and its molecules pile-up. After that, a blast wave occurs that carries a large amount of energy; also, it can carry some objects that can cause damage.

The main hypothesis of this work is that a 3D building structure can be designed for minimum weight according to AISC (2017) requirements that is also able to sustain some damage. Structures that are designed to minimize a cost function subject to constraints that must hold for intact and damaged structure are called optimal damage-tolerant structures. A damage condition is defined as complete or partial removal of members (Arora et al., 1979).

The prime objective of this study is to present a practical formulation for optimum design of 3D framed steel structures subjected to some damage due to blast loading. In the proposed design method, the structure remains stable after the damage happens. In the formulation, the cost function is the total weight of the structure. The design variables are frame members (columns and beams) which are discrete variables (specifically, W-shapes selected from the AISC tables). Constraints are AISC strength requirements. The structure is designed to withstand service loads (dead and live loads) and some projected damage conditions.

6.2 Review of Literature

Sun et al. (1976) discussed the fail-safe optimal design concept of truss structures subjected to stress, displacement, buckling, natural frequency, and design variables constraints under one or more predetermined damage conditions. Three bar plane truss, four bar space truss, and seventy-two-member space truss were optimized for minimum weight. Two types of damage conditions were considered. The first was complete removal of member(s) and the second type was a partial reduction in cross-section area of member(s). Mistree (1983) presented a mathematical model for continuous design variables, constraints, and system goals to design structures that were damage tolerant. Damage tolerance was represented as reserve strength and residual strength that the intact structures must have in order to avoid failure and to minimize the consequences of failure. Arora et al. (1980) formulated fail-safe designs of open truss and closed helicopter tail-boom structures. Damage conditions were defined as total or partial damage to chosen members. Some joints of the truss could be removed because of the damage. The optimum solutions for five cases, no damage and combinations of 6 damage conditions under total and reduced normal operating conditions, were obtained and the effect on structural weight was studied. It was shown that the structure could be designed to withstand possible future damage. Ming and Fleury (2016) established a mathematical model and formulation for fail-safe topology design optimization. 2D and 3D continuum structural examples subjected to different damage scenarios were discussed. The optimization problem was to minimize the compliance with a constraint on the material volume.

6.3 Design for Blast Loads

Designing structures to withstand blast loads that also remain elastic is usually uneconomical. That is, in design for blast loads, it is expected that some of the components will experience substantial nonlinear response, and the maximum dynamic deflection and rotation are

the criterion to prevent component failure. However, when a structure is required to be reused following a blast, it must be designed to remain elastic (ASCE, 2010). In addition, designers must provide sufficient redundancy (alternate load paths) to ensure that the failure of key members will not cause a progressive collapse of the structure. The reader is referred to DoD (2008), ASCE (2010), ASCE (2011), and Gilsanz et al. (2013) for more details about the design of structures for blast loads.

6.4 Formulation for Discrete Structural Optimization Problems

In structural design practice, members must be selected from the available sections in a catalog. Thus, design variables are discrete/integers (section number in the list). The formulation of the discrete design variables optimization problem with the nonlinear undamped dynamic response can be stated as:

$$\text{Find } \mathbf{X} = [x_1, x_2, \dots, x_{nvar}]; \quad x_i \in D_i; \quad i = 1, 2, \dots, nvar \quad (6.1)$$

$$\text{to minimize } f(\mathbf{X}) \quad (6.2)$$

$$\text{subject to } \mathbf{M}(\mathbf{X})\ddot{\mathbf{u}}(t) + \mathbf{K}(\mathbf{X}, \mathbf{u}(t))\mathbf{u}(t) = \mathbf{p}(t); \quad t = t_1, t_2, \dots, t_n$$

$$g_k(\mathbf{X}, \mathbf{u}(t), \dot{\mathbf{u}}(t), \ddot{\mathbf{u}}(t), t) \leq 0; \quad \text{for all } t \text{ and } k = 1, 2, \dots, l \quad (6.3)$$

where \mathbf{X} is the vector of design variables with $nvar$ unknowns, D_i is a set of discrete values for the i th design variable, $f(\mathbf{X})$ is a cost function (in this study, $f(\mathbf{X})$ is the total weight of the structure), \mathbf{M} is the mass matrix, \mathbf{K} is the stiffness matrix (\mathbf{K} is a function of the design variables and displacement vector for nonlinear dynamic analysis and just the design variables for linear dynamic analysis), \mathbf{u} is the dynamic displacements vector, $\dot{\mathbf{u}}$ is the velocities vector, $\ddot{\mathbf{u}}$ is the accelerations vector, n is the total number of the time steps, g_k is the k th constraint function that needs to be imposed at all time points, and l is the total number of constraints.

One way of treating constraints in metaheuristic algorithms is to combine constraints with the cost function to define a merit function (also called the penalty function) that is then minimized:

$$F(\mathbf{X}) = f(\mathbf{X})[1 + \psi G(\mathbf{X})]^\xi \quad (6.4)$$

$$G(\mathbf{X}) = \sum_{i=1}^n \sum_{k=1}^l \max(0, g_k(t_i)) \quad (6.5)$$

where $G(\mathbf{X})$ is a constraint violation function, $\psi \geq 1$ is exploration penalty coefficient (in this study, $\psi = 10$), $\xi > 1$ is penalty function exponent (in this study, $\xi = 2$), and $\max(0, g_k(t_i)) \geq 0$ is the violation value of the k th inequality constraint at the time point t_i . The present problem has just inequality constraints. The linear dynamic response problem is the same as the nonlinear dynamic response problem except that \mathbf{K} is not a function of the displacement vector \mathbf{u} . The formulation above (Eqs. (6.1) to (6.5)) is needed to solve the linear or nonlinear dynamic response optimization problems.

The linear static response optimization problem subjected to loading conditions can be stated as:

$$\text{Find } \mathbf{X} = [x_1, x_2, \dots, x_{nvar}]; \quad x_j \in D_j; \quad j = 1, 2, \dots, nvar \quad (6.6)$$

$$\text{to minimize } f(\mathbf{X}) \quad (6.7)$$

$$\text{subject to } g_k(\mathbf{X}) \leq 0; \quad k = 1, 2, \dots, p \quad (6.8)$$

$$F(\mathbf{X}) = f(\mathbf{X})[1 + \psi G(\mathbf{X})]^\xi \quad (6.9)$$

$$G(\mathbf{X}) = \sum_{k=1}^p \max(0, g_k(\mathbf{X})) \quad (6.10)$$

The linear static response formulation (Eqs. (6.6) to (6.10)) is needed to solve the optimization problems of structures subjected to service load only and some defined damages conditions.

6.4.1 Design Variables

In this work, the AISC (2017) W-shapes available in manufacturer's catalog are the design variables for beams and columns. All sections are chosen from AISC tables and assignment of a section specifies several cross-sectional properties for the member. Such design variables are classified as linked discrete variables (Arora, 2017). This way the design variables Eqs. (6.1) and (6.6) become:

$$\text{Find } \mathbf{X} = [S_1, S_2, \dots, S_{nvar}] \quad (6.11)$$

$$S_{imin} \leq S_i \leq S_{imax}; \quad i = 1, 2, \dots, nvar, \quad (6.12)$$

where S_i is an AISC W-shape number, and S_{imin} and S_{imax} are the lightest and the heaviest sections, respectively. In numerical calculations, W-shapes from the AISC table are re-arranged in an ascending order based on their weights.

6.4.2 Cost Function

In this study, the problem is to minimize the total weight of the structure (in kips). Thus, Eqs. (6.4) and (6.9) become:

$$W_s(\mathbf{X}) = \sum_{ng=1}^{NG} w_{ng} \sum_{mk=1}^{MK} L_{mk} \quad (6.13)$$

where W_s is the total weight of the structure, \mathbf{X} is the design vector, NG is the total number of member groups for the structure, w_{ng} is the weight per unit length (kips/ft) of the members in the

$ngth$ group (available in AISC's tables), MK is the number of members in the $ngth$ group, and L_{mk} is the length of the $mkth$ member (ft).

6.4.3 Constraints

6.4.3.1 Strength Constraints

According to the AISC (2017), symmetric members subjected to axial force and bending must satisfy the interaction ratio and shear force strength requirements:

$$\frac{P_u}{\phi P_n} + \frac{8}{9} \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right) - 1 \leq 0 \quad \text{if } \frac{P_u}{\phi P_n} \geq 0.2 \quad (6.14)$$

$$\frac{P_u}{2\phi P_n} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right) - 1 \leq 0 \quad \text{if } \frac{P_u}{\phi P_n} < 0.2$$

$$V_u \leq \phi_v V_n \quad (6.15)$$

$$\frac{V_u}{\phi_v V_n} - 1 \leq 0$$

Here ϕ is the resistance factor ($\phi_c = 0.85$ and $\phi_t = 0.90$ for compression and tension, respectively). $\phi_b = 0.9$ is the flexural resistance factor. P_u and P_n are the required and the nominal axial strengths (compression or tension) (kips), respectively. M_{ux} and M_{uy} are the required flexural strengths about the major and the minor axes (kip-ft), respectively. M_{nx} and M_{ny} are the nominal flexural strengths about the major and the minor axes (kip-ft), respectively. V_u and V_n are required and the nominal shear strengths (kips), respectively. $\phi_v = 0.9$ is the resistance factor for shear.

Evaluating P_n , M_{nx} and M_{ny} in Eqs. (6.14) is an involved process that requires checking of several failure modes (i.e., several "if then else" statements). For example, to find P_n , first one needs to find whether the member force is tensile or compressive. For tension members, P_n is calculated based on whether the gross section yields or the net section ruptures.

Constraints in Eqs. (6.14) and (6.15) need to be imposed at each point along the axis of every member in the structure. Thus, each equation represents infinite constraints. In the numerical process, the constraints are evaluated at several points along the axis of the member and they imposed at the point where they have maximum value. Therefore, the total number of interaction ratio constraints (Eq. (6.14)) equals the total number of members. Same is true for shear force constraints (Eq. (6.15)).

In addition to that the nominal strength calculations have several discontinuities as explained in the previous paragraph, that constraints in Eq. (6.14) has a discontinuity at $\frac{P_u}{\phi P_n}=0.2$. Thus, the gradient of these constraints is not possible and consequently, gradient-based optimization algorithms are not suitable for this class of optimization problems.

6.4.3.2 Displacement Constraints

In blast design, the maximum member end rotation shall be 2 degree and the maximum side-sway deflection (or inter-story drift (*ISD*)) is limited to 1/25 of the story height (high response design (ASCE, 2010)).

$$\frac{|\delta_r - \delta_{r-1}|}{\delta_{ru}} - 1 \leq 0 \quad (6.16)$$

$$\delta_{ru} = h_r/25 \quad (6.17)$$

where δ_r and δ_{r-1} are lateral displacements of two adjacent stories (in), δ_{ru} is the allowable lateral displacement, and h_r is the *r*th story height (in). At each node, SAP2000 evaluates displacements and rotations in 3-dimensions. Displacements in *x* and *y* directions are extracted to evaluate Eq. (6.17) to impose these constraints.

6.5 Optimization Process for Damage Tolerance

In the optimization of structures subjected to service and blast loads, the nonlinear dynamic formulation is used (Eqs. (6.1) to (6.5)) with strength and displacement constraints (Sections 6.4.3.1 and 6.4.3.2). That is, columns must remain elastic and beams are allowed to develop plastic hinges. The same formulation can be used in optimization of structures subjected to service and blast loads when columns and beams are to remain elastic except that \mathbf{K} (Eq. (6.3)) is a function of only the design variables and strength constraints are imposed.

In the optimization of structures subjected to service load only, the linear static formulation is used (Eqs. (6.6) to (6.10)) with strength constraints only. That is, columns and beams must remain elastic. The same is true in the optimization of structures subjected to service loads and some defined damages. However, the structure to be designed for some damages has a different stiffness any may be subjected to different service loads. For example, if the damage is defined as total removal of a column with 50% of live loads, the formulation is like intact structure subject to service load (say dead and live loads). In other words, for the structure with a missed column, live load is reduced to half of its original value.

Formulation for optimization of structures subjected to blast loading and to withstand some damage is a combination of the foregoing formulations. That is, two structures must be analyzed and all constraints must be evaluated for the same design vector. First, the intact structure subjected to blast loads with linear or nonlinear dynamic formulation and the intact structure subjected to service load with linear static formulation are analyzed. Second, the damaged structure subjected to service loads (or a part of them) with linear static formulation is analyzed. Namely, in every optimization iteration, the design vector is sent to two independent simulations. Then all the constraints are evaluated for use in Eqs. (6.5) and (6.10).

In nonlinear dynamic analysis, some members might be in their plastic region and the stiffness of these members is different from elastic stiffness. That is, a strength modification should be done on these members depending on their plastic deformation level. Since in this study the main focus is on formulating for the optimum design of framed structures and providing optimization procedure, the assumption is the structure is elastic.

6.6 Damage-Tolerant Design of Framed Steel Structure

The optimal design formulation for resilience presented in this study is evaluated using a moderate size 3D framed steel structure. The structure is a 4-bay x 4-bay x 3-story under serviceability loading. Figure 6.1 shows 3D view and the dimensions of the intact structure. It is a 3-story, 4 bays in x and y directions with 4 in concrete slab consisting of 197 members modeled using SAP2000 and MATLAB. All ground supports are fixed. Steel properties are: Young's modulus, $E=29000$ ksi, yield stress, $F_y=50$ ksi, ultimate strength, $F_u=65$ ksi, and Poisson's ratio, $\nu=0.3$. Concrete properties are: Young's modulus, $E=3605$ ksi, $f'_c=4000$ psi, and Poisson's ratio, $\nu=0.2$.

The frame members are divided into 9 groups as shown in Table 6.1. Each group is treated as a design variable. Gravity loads are assigned as uniformly distributed loads on the first floor and the second floor slabs consisting of design dead load of 60 psf and a design live load of 50 psf and on the roof slab, consisting of 60 psf and 25 psf for dead and live loads respectively. Load combinations are given in Table 6.2.

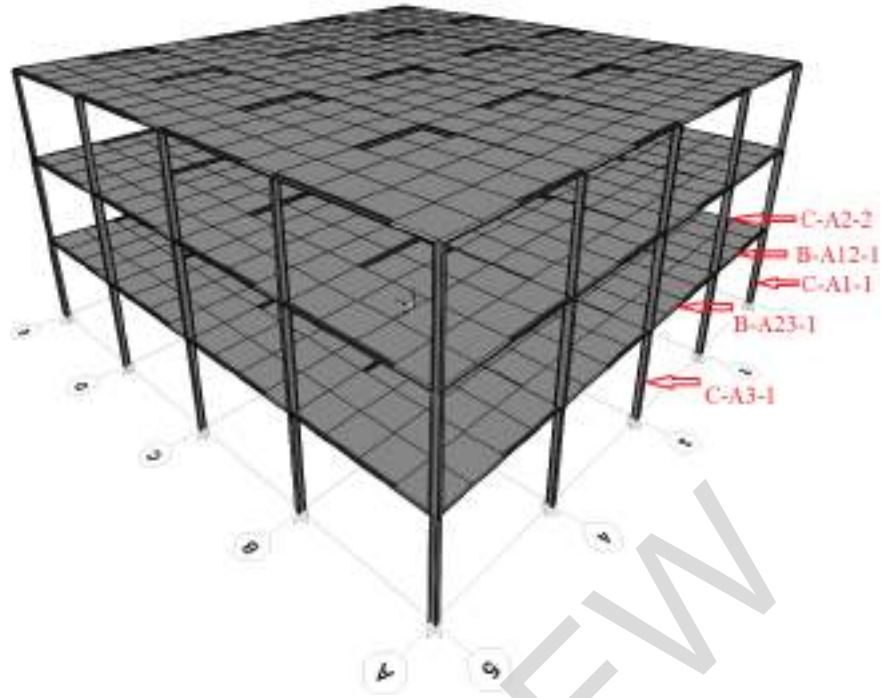


Figure 6.1. Schematic of 3D framed steel structure.

Table 6.1. Memembers grouping.

Group Number	Members	Number of members
1	1st floor external columns	16
2	2nd floor external columns	16
3	3rd floor external columns	16
4	1st floor internal columns	9
5	2nd floor internal columns	9
6	3 floor internal columns	9
7	1st floor beams	40
8	2nd floor beams	40
9	3rd floor beams	40

Table 6.2. Load combinations (AISC, 2015; Gilsanz et al., 2013).

Load combination	Scale factor
Comb 1	1.2 DL + 1.6 LL
Comb 2	1.4 DL
Comb 3	1.0 DL + 0.25 LL + 1.0 BL

DL is dead load, LL is live load, and BL is blast load.

This design example is solved in Chapter 5 using linear and nonlinear dynamic analyses. Therefore, the best designs obtained are checked for the damage conditions defined in the next paragraphs. If there is a violation in the strength constraints, the optimization process described in section 6.5 must be used to design the structure.

As mentioned earlier, there are two types of damages that will be discussed in this study: complete removal of some members and reduction in some members' strength. Table 6.3 shows damage condition definitions. In Table 6.3, 100% reduction in strength refers to complete removal of damaged member(s) and 50% reduction in strength means the damaged member is still in action but it has lost half of its normal strength.

The aim of this work is to introduce a general formulation for damage tolerance of framed structures. Therefore, just 6 damage conditions are discussed. One, however, any other damage condition may be modeled in a similar way.

Table 6.3. Damage condition definitions.

Damage condition	Members	Reduction in strength, %
1	C-A3-1*	100
2	C-A1-1	100
3	C-A2-2, B-A23-1, and B-A12-1	100
4	C-A3-1	50
5	C-A1-1	50
6	C-A2-2, B-A23-1, and B-A12-1	50

* Figure 6.1 shows the location of each member.

6.6.1 Complete Removal of Some Members

In this type of damage, the assumption is that a member or more are totally damaged, and they are not able to carry gravity loads. Thus, damaged members are removed from the model of the structure depending on the damage condition defined in Table 6.3 (damage conditions 1, 2, and 3). The best designs of spatial steel frame subjected to blast load without external walls with linear and nonlinear dynamic analyses (Sections 5.8.2.1.1 and 5.8.2.2.1) are checked for the following six cases:

Case I: Structure with damage condition 1.

Case II: Structure with damage condition 2.

Case III: Structure with damage condition 3.

Case IV: Case I except live load is 50% of the normal condition.

Case V: Case II except the live load is 50% of the normal condition.

Case VI: Case III except the live load is 50% of the normal condition.

Tables 4 and 5 show the constraints evaluation for the six damages for linear and nonlinear dynamic analysis study cases. In both cases, there is no constraints violation. Thus, these designs (Table 5.5 and Table 5.8) are safe for the proposed damage conditions and there is no need to solve the optimization problems again and include the anticipated damage conditions. For other problems, however, the procedure described in Section 6.5 may need to be followed.

Table 6.4 shows that damage case II has the highest maximum interaction ratio of 0.120. This ratio is far from 1 which indicates the best design using linear dynamic analysis can tolerate the defined damages.

Table 6.5 shows that damage case I has the highest maximum interaction ratio of 0.973. This ratio is close to 1 which indicates that the best design using nonlinear dynamic analysis is more critical than the best design using linear dynamic analysis.

In all study cases, shear ratios are not critical.

Table 6.4. Constraints evaluation of the optimal design using linear dynamic analysis for complete removal study case (From Table 5.5).

Constraint	No damage	Damage case					
		I	II	III	IV	V	VI
Max. interaction ratio (Eq. (6.14))	0.061	0.102	0.120	0.109	0.093	0.110	0.100
Max. shear ratio (Eq. (6.15))	0.035	0.049	0.039	0.045	0.041	0.034	0.038

Table 6.5. Constraints evaluation of the optimal design using nonlinear dynamic analysis for complete removal study case (From Table 5.8).

Constraint	No damage	Damage case					
		I	II	III	IV	V	VI
Max. interaction ratio (Eq. (6.14))	0.672	0.973	0.713	0.864	0.806	0.607	0.725
Max. shear ratio (Eq. (6.15))	0.287	0.307	0.288	0.302	0.252	0.234	0.244

6.6.2 Strength Reduction of Some Members

For fixed material and structural geometry, reduction in strength of truss structure members can be directly defined as a reduction in cross-sectional areas because they are the only variables involved in calculating stiffness matrix and members' strength. However, for W-shape sections and AISC strength requirement, the reduction in strength must be defined differently because cross-sectional areas are not the only variables in the stiffness matrix and sections' strength calculations. Frame member stiffness matrix calculation requires cross-sectional area, moment of inertia in x direction, and moment of inertia in y direction. Also, Section 6.4.3.1 shows that estimating members' strength require calculating some other quantities such as local buckling which involves finding the radius of gyration value and other factors. Thus, in this study reducing damaged members' dimensions is the way to reduce sections' strength. That is, when a section is signed to a group (Table 6.1), all dimensions of damage members are reduced by a percentage of the reduction in strength value (Table 6.3). Other possible ways of strength reductions are:

- 1- Reducing the modulus of elasticity and strength reduction factors in Eqs. (6.14) and (6.15) values for damaged members only.
- 2- Reducing areas, moment of inertia, and strength reduction factors in Eqs. (6.14) and (6.15) values for the damaged members only.

The best designs of spatial steel frame subjected to blast load without external walls with linear and nonlinear dynamic analyses (Sections 5.8.2.1.1 and 5.8.2.2.1) are checked for the following two cases:

Case I: Structure with damage conditions 4-6.

Case II: Case I except the live load is 50% of the normal condition.

Table 6.6 and Table 6.7 show the constraints evaluation for the two damage conditions for linear and nonlinear dynamic analysis study cases. The results lead to the same conclusions as for previous study case: the best designs of linear and nonlinear dynamic analyses (Table 5.5 and Table 5.8) are safe for the proposed damages and there is no need to solve the design optimization problems again to include the damage conditions. Also, the best design using nonlinear dynamic analysis is more critical than the best design using linear dynamic analysis.

Table 6.6. Constraints evaluation of the optimal design using linear dynamic analysis for strength reduction study case (From Table 5.5).

Constraint	No damage	Damage case	
		I	II
Max. integration ration (Eq. (6.14))	0.061	0.331	0.286
Max. shear ratio (Eq. (6.15))	0.035	0.042	0.035

Table 6.7. Constraints evaluation of the optimal design using nonlinear dynamic analysis for strength reduction study case (From Table 5.8).

Constraint	No damage	Damage case	
		I	II
Max. integration ration (Eq. (6.14))	0.672	0.826	0.611
Max. shear ratio (Eq. (6.15))	0.287	0.296	0.240

6.7 Concluding remarks

In this research, formulation and procedure for optimization of framed structure to endure some possible damages caused by a blast are explained and discussed. Two types of damages are considered: complete removal of members with six damage cases and strength reduction of members with two damage cases. The problem is formulated to minimize the total structural weight subjected to strength and displacement constraints. The design variables for beams and columns are W-shapes sections selected from the AISC tables.

The problem is 4-bay x 4-bay x 3-story framed steel building that is solved in Chapter 5 using linear and nonlinear dynamic analyses. Therefore, the best designs are examined for the

damage cases. Based on evaluating constraints, the results show that the best design using linear dynamic analysis is less critical than the best design using nonlinear dynamic analysis.

Further research will be needed to study the following cases:

- 1- Study the effect of strength reduction of all façade's members based on the distance from the blast location.
- 2- For the nonlinear dynamic analysis, a new formulation based on strength reduction of members that develop plastic hinges may be studied.

PREVIEW

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Discussion and Conclusion

Optimum design of 3D framed steel structures subjected to service and blast loads are studied using metaheuristic optimization algorithms. The main purpose of this research is to develop a formulation for the design optimization problem to withstand blast loads and a formulation for the design optimization problem to withstand some possible damages due to blast loads. The optimization formulations for this class of problems are presented for the first time in this study. The problems are formulated to minimize total weight of framed structures subjected to American Institution of Steel Construction (AISC) strength requirements and blast design displacement constraints. The design variables for beams and columns are the discrete values of the W-shapes selected from the AISC tables. All optimization algorithms and structures are coded in MATLAB and interfaced with the structural analysis program SAP2000 using its Open Application Programming Interface (OAPI). Three types of structural analyses are investigated: linear static analysis of the framed structure subjected to service loads only or equivalent static loads, linear dynamic analysis of the framed structure subjected to service and blast loads, and nonlinear dynamic analysis (geometrical and material nonlinearities) of the framed structure subjected to service and blast loads.

The thesis has made five main contributions in an effort to develop a practical formulation for the design optimization of framed steel structures subjected to blast loads:

- 1- *Review of topics related to the design of blast-resistant structures.* Methods to predict blast loads, modeling of structures under blast loading, and design requirements are discussed from

a structural design optimization perspective. References on different subjects are provided for further details. Chapter 2 presents a concise state-of-the-art document on the subject.

- 2- *Robust and efficient metaheuristic optimization algorithm.* A two-phase metaheuristic algorithm based on the well-known Harmony Search (HS) algorithm and recently developed Colliding Bodied Optimization (CBO) is developed. The algorithm is called Hybrid Harmony Search-Colliding Bodies Optimization (HHC). Also, a new design domain reduction technique is integrated in IHS that reduces the number of possible combinations of discrete variables. The results comparing HHC with other popular metaheuristic algorithms using some benchmark discrete structural optimization problems shows that HHC is quite reliable in obtaining the best designs with fewer structural analyses.
- 3- *Investigation of the Equivalent Static Load (ESL) approach for optimization of structures subjected to dynamic loads with discrete design variables using metaheuristic algorithms.* Since optimizing transient response problems (specifically, problems requiring nonlinear dynamic analysis) using metaheuristic algorithms is computationally very expensive, the ESL method with gradient-free algorithms is examined. The method is named global optimization with equivalent static load (GOESL). The results of four numerical examples show that ESL step is not able to obtain the best design; however, it reaches near the best design with a drastically reduced number of transient analyses of the structure. Thus, after a few ESL cycles, the procedure may switch to dynamic analysis of each member of the population to improve designs further and reach the best design.
- 4- *Formulation for the design optimization of three-dimensional framed steel structures subjected to blast loads.* Linear and nonlinear dynamic analyses are carried out in the optimization process. It is shown that for linear dynamic analysis (beams and columns are designed to

remain elastic) the optimum structure is about 14 times heavier than the optimum design of the structure subjected to service loads only. When columns are only designed to remain elastic and beams are allowed to develop plastic hinges with displacement requirements, the optimum structure is about 6 times heavier than the optimum design of the structure subjected to service loads only. Inclusion of outer wall in the analysis model reduces the weight of the structure dramatically to withstand blast loads. This give a very practical design solution for blast resistant design of structures. However, it also implies that the outer walls must be designed and properly anchored to the beam and columns to fully contribute to the stiffness of the structure.

- 5- *Formulation for the design optimization of three-dimensional framed steel structures subjected to some possible damages due to blast loads.* The optimization procedure for two types of damages are discussed: complete removal of members with six damage cases and strength reduction of members with two damage cases. It is shown the best design using linear and nonlinear dynamic analyses (from Chapter 5) can endure the defined damage conditions without any further damage.

7.2 Future Work

While the formulation of optimum design of framed steel structures subjected to blast loads is studied that includes the definition of design variables, cost function, constraints, optimization algorithms, and structural analysis type, further research is suggested as follows:

- 1- *Improving HHC algorithm performance.* While average and standard deviation are used in the domain reduction step to modify design variables bounds in the quest to speed-up the search process, other methods such as modern machine learning methods might be used to speed-up the entire optimization process. Also, it is noticed that Phase 2 (ECBO) shows less

diverse designs after about 25% of the total number of allowed iterations. Methods to increase diversity might improve HHC performance in term of the number of structural analyses and quality of final designs.

- 2- *Improving GOESL algorithm.* Although GOESL reduces the number of dynamic analyses drastically to obtain the best design, ESL step is not able to reach the best design. Further research is needed to improve performance of the ESL approach with discrete design variables and gradient-free methods.
- 3- *Formulation for the design optimization of three-dimensional framed steel structures subjected to blast loads with glass curtain.* While in this study the conservative assumption is considered, that is, the external envelope of the structure is stiff enough to transfer all the blast loads to the frame system, other models such as buildings with external glass curtain can be studied. Also, the effect of openings in the walls needs to be considered.
- 4- *Formulation for optimum design of 3D framed steel structures subjected to damages based on strength reduction of members that developed plastic hinges.* In nonlinear dynamic analysis, some members may develop plastic hinges. Therefore, stiffness of these members is different from elastic stiffness. Thus, these members may be considered as damaged members and their strength modification factors should be developed depending on the plastic deformation level.

BIBLIOGRAPHY

- ABAQUS/Standard Version 6.14 User's Manual. (2014). Dassault Systèmes Simulia Corp., Providence, RI, USA.
- AISC (2017). Steel construction manual (15th Ed.). Chicago: American Institute of Steel Construction.
- Ansys, Inc., Version 15.0 User's Manual. (2013). Canonsburg, PA, USA.
- Arora, J. (1999). Optimization of structures subjected to dynamic loads. *Structural dynamic systems computational techniques and optimization*, 7, 1-73.
- Arora, J. S. (2017). Introduction to Optimum Design (4th Ed.). Elsevier Inc.
- Arora, J. S., Haskell, D. F., & Govil, A. K. (1980). Optimal design of large structures for damage tolerance. *AIAA Journal*, 18(5), 563-570.
- American Society of Civil Engineers, Task Committee on Blast-Resistant Design. (2010). *Design of blast-resistant buildings in petrochemical facilities* (2nd ed), American Society of Civil Engineers, Reston, VA.
- American Society of Civil Engineers. (2011). *Blast protection of buildings: Standard ASCE/SEI 59-11*, American Society of Civil Engineers, Reston, VA.
- American Society of Civil Engineers and United States Federal Emergency Management Agency. (2000). *Prestandard and commentary for the seismic rehabilitation of buildings* (FEMA 356), American Society of Civil Engineers, Reston, VA.
- Bangash, T., Bangash, M. (2006). Explosion-resistant buildings: design, analysis, and case studies. Springer Science & Business Media.
- Bhatti, M. A. (2006). Advanced topics in finite element analysis of structures: with Mathematica and MATLAB computations. John Wiley New York.
- Camp, C. V. (2007). Design of space trusses using big bang–big crunch optimization. *Journal of Structural Engineering*. 133(7): 999–1008.
- Choi, W., & Park, G. (2002). Structural optimization using equivalent static loads at all time intervals. *Computer Methods in Applied Mechanics and Engineering*, 191(19-20), 2105-2122.
- Cormie, D., Mays, G., Smith, P. (2009). Blast Effects on Buildings. Thomas Telford Ltd.
- Computer and Structures Inc. (CSI). (2017). CSI Analysis Reference Manual. Berkeley, CA, USA.
- Computer and Structures Inc. (CSI). (2016). Technical Note: Parametric P-M2-M3 Hinge Model.
- Degertekin, S. O. (2008). Optimum design of steel frames using harmony search algorithm. *Structural and Multidisciplinary Optimization*, 36: 393–401.
- DoD (U.S. Department of Defense) (2008). Structures to resist the effects of accidental explosions. UFC 3-340-02, Washington, DC.

- DoD (U.S. Department of Defense) (2016). Design of buildings to resist progressive collapse. UFC 4-023-03.
- Dorigo, M., Maniezzo, V., Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1):29–41.
- Draganić, H., Sigmund, V. (2012). Blast Loading on Structures. Tech. vjesn., Croatia.
- Dusenberry, D., O. (2010). Handbook for Blast-Resistant Design of Buildings. John Wiley & Sons, Inc., Hoboken, New Jersey, USA.
- El-Tawil, S. and Deierlein, G. (2001). Nonlinear Analysis of Mixed Steel-Concrete Frames, I: Element Formulation. *Journal of Structural Engineering*, Vol. 126, No. 6.
- Erol, O., Eksin, I. (2006). A new optimization method: Big Bang-Big Crunch. *Advances in Engineering Software*. 37(2), 106–111.
- Federal Emergency Management Agency (FEMA). (2000). Prestandard and Commentary for the Seismic Rehabilitation of Buildings, FEMA 356. Washington, D.C., USA.
- Federal Emergency Management Agency (FEMA). (2003). Reference Manual to Mitigate Potential Terrorist Attacks Against Buildings, FEMA 426. Washington, D.C., USA.
- Geem, Z. W. (2009). Harmony Search Algorithms for Structural Design Optimization. Springer.
- Geem, Z. W., Kim, J. H., Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *SAGE Journals*. 76(2): 60-68.
- Gilsanz, R., Hamburger, R., Barker, D., Smith, J., Rahimian, A. (2013). Design of blast resistant structures. Steel Design Guide No. 26. American Institute of Steel Construction, Chicago, IL, USA.
- Goel, M. D., Matsagar, V. A. (2014). Blast-Resistant Design of Structures. Practice Periodical on Structural Design and Construction. ASCE, Vol. 19, Issue 2.
- Goldberg, D. E., Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2), 95–99.
- Huang, M. W., Arora, J. S. (1997). Optimal design of steel structures using standard sections. *Structural and Multidisciplinary Optimization*. 14: 24-35.
- Hyde, D. (1992). ConWep - Application of TM 5-855-1. Fundamentals of protective design for conventional weapons. Structural Mechanics Division, Structures Laboratory, USACE Waterways Experiment Station, Vicksburg, MS, USA.
- Jeyarajan, S., Richard, J., Koh, C. (2015). Vulnerability of simple braced steel building under extreme load. *The IES Journal Part A: Civil and Structural Engineering*. 8(4): 219-231.
- Kang, B., Choi, W., & Park, G. (2001). Structural optimization under equivalent static loads transformed from dynamic loads based on displacement. *Computers & Structures*, 79(2), 145-154.

- Kang, B., Park, G. , & Arora, J. (2006). A review of optimization of structures subjected to transient loads. *Structural and Multidisciplinary Optimization*, 31(2), 81-95.
- Karlos, V., Solomos, G. (2013). Calculation of Blast Loads for Application to Structural Components. JRC Technical Reports, Joint Research Centre, European Commission, Publication Office of the European Union, Luxembourg. ISBN 978-92-79-35158-9.
- Kaveh, A. (2017). *Advances in Metaheuristic Algorithms for Optimal Design of Structures* (2nd Ed.). Springer.
- Kaveh, A., Bakhshpoori, T., Azimi, M. (2015). Seismic optimal design of 3D steel frames using cuckoo search algorithm. *The Structural Design of Tall and Special Building*. 24: 210-227.
- Kaveh, A., Ghazaan, M., (2014). Enhanced colliding bodies optimization for design problems with continuous and discrete variables. *Advances in Engineering Software*. 77: 66-75.
- Kaveh, A., Ghazaan, M. I., & Bakhshpoori, T. (2013). An improved ray optimization algorithm for design of truss structures. *Periodica Polytechnica Civil Engineering*, 57(2), 97-112.
- Kaveh, A., Khayat, A. M. (2012). A novel meta-heuristic method: ray optimization. *Computer and Structures*. 112–113:283–294.
- Kaveh, A., Mahdavi, V. R. (2014). Colliding bodies optimization: a novel meta-heuristic method. *Computers and Structures*. 139: 18-27.
- Kaveh, A., Mahdavi, R. (2015). *Colliding Bodies Optimization: Extension and Application*. Springer.
- Kaveh, A., Talatahari, S. (2009) Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *J. Constr. Steel Res.* 65 (8–9): 1558–1568.
- Kennedy, J., Eberhart, R. (2001). *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers.
- Khaledy, N., Habibi, A., Memarzadeh, P. (2018). A Comparison between different techniques for optimum design of steel frames subjected to blast. *Latin American Journal of Solids and Structures*. 15 (9), e106.
- Khatavakar, N., et. al. (2016). Response of High Rise Structures Subjected to Blast Loads. *International Journal of Science, Engineering and Technology Research*, Volume 5, Issue 7.
- Kim, Y., & Park, G. (2010). Nonlinear dynamic response structural optimization using equivalent static loads. *Computer Methods in Applied Mechanics and Engineering*, 199(9-12), 660-676.
- Kirkpatrick, S., Gelatt, C., Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220 (4598): 671–80.
- Kralik, J., Baran Michal (2012). Protection of the Buildings Against the Explosion Effects. *Applied Mechanics and Materials*, Vol. 390, p. 230-234.

- Lee, K., Chung, L., Lee S., Park, T., Rho, J. (2011). Evaluation of Dynamic Collapse Behavior of Steel Moment Frames Damaged by Blast. *Applied Mechanics and Materials*. 82: 404-409.
- Lee, K. S., Geem, Z. W., Lee, S. H., Bae, K. W. (2005). The harmony search heuristic algorithm for discrete structural optimization. *Eng. Optimization*. 37(7): 663–684.
- Li, L. J., Huang, Z. B., Liu, F. (2009). A heuristic particle swarm optimization method for truss structures with discrete variables. *Computer and Structures*. 87(7–8): 435–443.
- Li L., Ren F., Liu F, Wu Q. (2006). An improved particle swarm optimization method and its application in civil engineering. In: The eighth international conference on computation and structures technology. Las Palmas de Gran Canaria, Spain. paper 42.
- Longinow, A., Alfawakhiri, F. (2003). Blast Resistant Design with Structural Steel. Modern Steel Construction.
- LS-Dyna Version 971 User's Manual. (2007). Livermore Software Technology, Co.
- Mahdavi, M., Fesanghary, M., Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*. 188(2), 1567–1579.
- Marchand, K. A., & Alfawakhiri, F. (2004). *Facts for steel buildings: Blast and progressive collapse*: American Institute of Steel Construction.
- Mistree, F. (1983). Design of damage tolerant structural systems. *Engineering Optimization*, 6(3), 141-144.
- Ngo, T., Mendis, P., Gupta, A., & Ramsay, J. (2007). Blast loading and blast effects on structures—an overview. *Electronic Journal of Structural Engineering*, 7(S1), 76-91.
- NYCBC (2008), Building Code of the City of New York, New York City Building Code, New York, NY.
- Pape, R., Mniszewski, K. R., Longinow, A., & Kenner, M. (2010a). Explosion phenomena and effects of explosions on structures. I: Phenomena and Effects. *Practice Periodical on Structural Design and Construction*, 15(2), 135-140.
- Pape, R., Mniszewski, K. R., Longinow, A., & Kenner, M. (2010b). Explosion phenomena and effects of explosions on structures. II: Methods of Analysis (Explosion Effects). *Practice Periodical on Structural Design and Construction*, 15(2), 141-152.
- Pape, R., Mniszewski, K. R., Longinow, A., & Kenner, M. (2010c). Explosion phenomena and effects of explosions on structures. III: Methods of analysis (explosion damage to structures) and example cases. *Practice Periodical on Structural Design and Construction*, 15(2), 153-169.
- Rajeev, S., Krishnamoorthy, C. (1992). Discrete optimization of structures using genetic algorithms. *Journal of Structural Engineering*. 118(5): 1233–1250.
- Rose T.A. An approach to the evaluation of blast loads on finite and semi-infinite structures. PhD thesis, Engineering Systems Department, Cranfield University, Royal Military College of Science, February 2001.

- Stea, W., Tseng, G., Kossover, D. (1977). Nonlinear Analysis of Frame Structures Subjected to Blast Overpressures. US Army Armament Research and Development Command. Contractor Report ARLCD-CR-77008. Dover, New Jersey.
- Smith, P. D. and Hetherington J. G. (1994). Blast and Ballistic Loading of Structures. Butterworth-Heinemann Ltd.
- Sun, P. F., Arora, J. S., & Haug Jr, E. J. (1976). Fail-safe optimal design of structures. *Engineering Optimization*, 2(1), 43-53.
- Sun, W., Chang, X. (2015). An improved harmony search algorithm for power distribution network planning. *Journal of Electrical and Computer Engineering*. Vol. 2015, Article ID 753712.
- Weise, T. (2009). Global optimization algorithms-theory and application. *Self-published*, 2.
- Xiang, B. W., Chen, R. Q., Zhang, T. (2009). Optimization of trusses using simulated annealing for discrete variables, image analysis and signal processing. *International Conference on Image analysis and signal processing*. 2009, pp. 410.
- Zhou, M., & Fleury, R. (2016). Fail-safe topology optimization. *Structural and Multidisciplinary Optimization*, 54(5), 1225-1243.
- Zienkiewicz O.C., Taylor R.L. and Nithiarasu P. (2006). The finite element method for fluid dynamics (6th Ed.). Butterworth-Heinemann.

APPENDIX A

MATLAB CODE FOR PLANAR 10-BAR TRUSS STRUCTURE

```

%% 10-bar planar truss
NVAR=10; % number of design variables
% Print results %
pr=1; % pr=0 do not print
pr500=0; pr50=0; % counting to print results
HMS=75; % Harmony memory size Phase 1
popSize=40; % Population size Phase 2
% Choose an algorithm
Method=3; % Method= 1=IHS 2=ECBO 3=HHC
DR=1; % DR=1 domain reduction DR=0 no domain reduction
% HHCD parameters
Eps=10e-3; r1=0.25; r2=0.1; r3=0.1;
% constraints limits %
MaxS=25*10^3; % stress limit ksi
Maxd=2; % allowable displacement in
% Cross-sectional areas %
Sections= [1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, ...
3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, ...
4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, ...
15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50] ;
% IHS %
% IHS parameters %
MaxItrIHS=10*NVAR*length(Sections); % IHS Maximum number of iterations
if Method==1
MaxItrIHS=50000;
end
HMCRmax=0.85; % minimum harmony consideration rate
HMCRmin=0.35; % maximum harmony consideration rate
PARmin=0.35; % minimum pitch adjusting rate
PARmax=0.85; % maximum pitch adjusting rate
% range of variables %
for i=1:NVAR
PVB(i,:)= [1,length(Sections)];
end
% Initiate Matrices %
HM=zeros(HMS,NVAR); % harmony memory matrix
NCHV=zeros(1,NVAR); % updated design vector
BestGen=zeros(1,NVAR); % best design
fitness=zeros(1,HMS); % merit function values
for i=1:HMS
for j=1:NVAR % random initial designs
HM(i,j)=round(rand(1)*(PVB(j,2)-PVB(j,1))+PVB(j,1));
end % end "for j=1:NVAR"
A=Sections(HM(i,:));
[weight, Stress, Disp] = TenBarTrussCase (A);
Sum=0;
GM=[abs(Stress)/MaxS;abs(Disp)/Maxd];
for g=1:length(GM)

```

```

G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
merit=weight*(1+Sum)^2;           % merit function value
fitness(i)=merit;SUM(i)=Sum;
end % end "for i=1:HMS"
FHM1=[fitness',HM];
FHM1=sortrows(FHM1);             % ascending order based on merit value
DV1=FHM1(1,2:end);              % best design of initial designs
if Method==1 || Method==3
% MainHarmony
iterIHS = 0;
for itr=1:MaxItrIHS
PAR=(PARmax-PARmin)/(pi/2)*atan(itr)+PARmin;
HMCR=HMCRmax-(HMCRmax-HMCRmin)*itr/MaxItrIHS;
for i =1:NVAR
ran = rand(1);
if( ran < HMCR )                 % memory consideration
index=round(rand(1)*(HMS-1)+1);
NCHV(i) = HM(index,i);
pvbRan = rand(1);
if( pvbRan < PAR)                % pitch adjusting
pvbRan1 = rand(1);
result = NCHV(i);
if( pvbRan1 < 0.5)
result =result+ 1;
if( result < PVB(i,1))
NCHV(i) = PVB(i,1);
end
else
result =result- 1;
if( result > PVB(i,2))
NCHV(i) = PVB(i,2);
end
end
else
NCHV(i) = round(rand(1)*(PVB(i,2)-PVB(i,1))+PVB(i,1)); % random selection
if NCHV(i)<1
NCHV(i)=1;
end
if NCHV(i)>length(Sections)
NCHV(i)=length(Sections);
end
end
end
for i=1:HMS
ts=isequal(NCHV,HM(i,:));
if ts==1
for j=1:NVAR                    % random design
NCHV(j)=round(rand(1)*(PVB(j,2)-PVB(j,1))+PVB(j,1));
end
end
end

```

```

end
% evaluating the new design
A=Sections(NCHV);
[Weight, Stress, Disp] = TenBarTrussCase (A);
Sum=0;
GM=[abs(Stress)/MaxS;abs(Disp)/Maxd];
for g=1:length(GM)
G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
merit=Weight*(1+Sum)^2;
newFitness=merit;newSum=Sum;
% If this design is better than the worst design in the current %
% population, then it replaces that design %
if(iterIHS==0)
BestFit=fitness(1);
for i = 1:HMS
if( fitness(i) <= BestFit )
BestFit = fitness(i);
BestIndex =i;
end
end

WorstFit=fitness(1);
for i = 1:HMS
if( fitness(i) >= WorstFit )
WorstFit = fitness(i);
WorstIndex =i;
end
end
end
if (newFitness< WorstFit)
if( newFitness < BestFit )
HM(WorstIndex,:)=NCHV;
BestGen=NCHV;
fitness(WorstIndex)=newFitness;SUM(WorstIndex)=newSum;
BestIndex=WorstIndex;
else
HM(WorstIndex,:)=NCHV;
fitness(WorstIndex)=newFitness;SUM(WorstIndex)=newSum;
end
WorstFit=fitness(1);
WorstIndex =1;
for i = 1:HMS
if( fitness(i) > WorstFit )
WorstFit = fitness(i);
WorstIndex =i;
end
end
end
iterIHS=iterIHS+1;
[val, loc]=min(fitness);
Merit(iterIHS)=val;

```

```

%% Domain Reduction Technique
if DR==1
if itr>round(r3*MaxItrIHS)
FHM=[fitness',HM];
FHM=sortrows(FHM);
isum=1;
Fsum(1)=1;
for i=1:HMS
if SUM(i)<=0.05
Fsum(isum)=i;
isum=isum+1;
end
end
if length(Fsum)>HMS*0.05
for i=1:NVAR
Std(i)=std(Sections(HM(Fsum,i)));
Mean(i)=mean(Sections(HM(Fsum,i)));
pvb(i,1)=Mean(i)-Std(i);
[pvbval pvbloc]=min(abs(pvb(i,1)-Sections));
PVB(i,1)=pvbloc;
if PVB(i,1)<1
PVB(i,1)=1;
end
pvb(i,2)=Mean(i)+Std(i);
[pvbval pvbloc]=min(abs(pvb(i,1)-Sections));
PVB(i,2)=pvbloc;
if PVB(i,2)>length(Sections)
PVB(i,2)=length(Sections);
end
if PVB(i,2)-PVB(i,1)<4
PVB(i,1)=round(mean(PVB(i,2)+PVB(i,1))-2);
PVB(i,2)=round(mean(PVB(i,2)+PVB(i,1))+2);
if PVB(i,1)<1
PVB(i,1)=1;
end
if PVB(i,2)>length(Sections)
PVB(i,2)=length(Sections);
end
end
end
end
for i=1:NVAR
if PVB(i,1)>=FHM(1,1+i)
PVB(i,1)=FHM(1,1+i)-2;
end
if PVB(i,2)<=FHM(1,1+i)
PVB(i,2)=FHM(1,1+i)+2;
end
if PVB(i,1)<1
PVB(i,1)=1;
end
if PVB(i,2)>length(Sections)
PVB(i,2)=length(Sections);
end
end
end

```

```

end
end
%%% End Domain Reduction Technique %%%
% Stopping criteria
if Method==3
if iterIHS>r1*MaxItrIHS
if (Merit(iterIHS-round(r2*iterIHS))-Merit(iterIHS))/Merit(iterIHS) <=Eps
break
end
end
end
FHM=[fitness',HM];
ITER=iterIHS;
if pr==1
if ITER==1 || ITER==pr500*500
pr500=pr500+1;
fprintf('iter(IHS)=%4.0f   Merit=%6.2f \n',iterIHS,Merit(end)) % Print results at each step
end
end
end % end "for itr=1:MaxItrIHS"
%%% End of Phase 1 %%%
FHM=sortrows([fitness',HM]); % ascending order based on merit value
DV2=FHM(1,2:end);
end
if Method==1
[a b]=min(Merit);
fprintf('iter(IHS) = %4.0f   Merit = %6.2f \n',b,a)
hold on
plot(Merit,'Linewidth',2)
plot(b,a,'r*','Linewidth',2)
hold off
return
end
MaxItrECBO=NVAR*length(Sections); % ECBO Maximum number of iteration
if Method==2
FHM=FHM1;
iterIHS=0;
MaxItrECBO=1000;
end
if Method==3
fprintf('Phase1 = %4.0f   iteration Merit = %6.2f \n',iterIHS,Merit(end))
end
% ECBO %
for i=1:NVAR % re-set the range of variables
PVB(i,:)= [1,length(Sections)];
end
if Method==2 || Method==3
% ECBO parameters %
cms=0.1*popSize; % Colliding memory size
pro=0.5; % Pro parameters
CB=FHM(1:popSize,2:end); % Colliding Bodies matrix
iterECBO=0; % counter iterations
agentCost=zeros(popSize,2); % array of agent costs
agentCost(:,2)=[1:popSize];

```



```

agentCost(:,1)=FHM(1:popSize,1);
% Colliding memory; The first column contains CB costs and the remaining
% columns include CB positions
cm=zeros(cMs,NVAR+1);           % Colliding Memory matrix
tm=zeros(2*cMs,NVAR+1);        % Temporary memory
cm(1:cMs,:)=FHM(1:cMs,:);
while iterECBO < MaxItrECBO
iterECBO=iterECBO+1;
% Evaluating the population
if iterECBO > 1
for e=1:popSize
A=Sections(CB(e,:));
[Weight, Stress, Disp] = TenBarTrussCase (A);
Sum=0;
GM=[abs(Stress)/Maxs;abs(Disp)/Maxd];
for g=1:length(GM)
G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
merit=Weight*(1+Sum)^2;
% cost=eval(CB(e,:)); % evaluating objective function for each agent
agentCost(e,1)=merit;
agentCost(e,2)=e;
end %for
end
% Updating colliding memory
agentCost=sortrows(agentCost);
if iterECBO>1
for e=1:cMs
agentCost(popSize-cMs+e,1)=cm(e,1);
for ee=1:NVAR
CB(agentCost(popSize-cMs+e,2),ee)=cm(e,ee+1);
end
end
end
for e=1:cMs
tm(e,1)=agentCost(e,1);
tm(e+cMs,1)=cm(e,1);
for ee=1:NVAR
tm(e,ee+1)=CB(agentCost(e,2),ee);
tm(e+cMs,ee+1)=cm(e,ee+1);
end
end
tm=sortrows(tm);
for e=1:cMs
cm(e,:)=tm(e,:);
end
agentCost=sortrows(agentCost);
% Evaluating the mass
mass=zeros(popSize,1);
for e=1:popSize
mass(e,:)=1/(agentCost(e,1));
end

```

```

% Updating CB positions
for e=1:popSize/2
    indexS=e; % index of stationary bodies
    indexM=popSize/2+e; % index of moving bodies
    COR=(1-(iterECBO/MaxItrECBO)); % coefficient of restitution
    % velocity of moving bodies before collision
    velMb=(((CB(agentCost(indexS,2),:)-CB(agentCost(indexM,2),:)))));
    % velocity of stationary bodies after collision
    velSa=(((1+COR)*mass(indexM,1))/(mass(indexS,1)+mass(indexM,1))*velMb));
    % velocity of moving bodies after collision
    velMa=(((mass(indexM,1)-COR*mass(indexS,1))/(mass(indexS,1)+mass(indexM,1))*velMb));
    CB(agentCost(indexM,2),:)=round(CB(agentCost(indexS,2),:)+2*(0.5-rand(1,NVAR)).*velMa);
    CB(agentCost(indexS,2),:)=round(CB(agentCost(indexS,2),:)+2*(0.5-rand(1,NVAR)).*velSa);
    if rand<pro
        tmp=ceil(rand*NVAR);
        CB(agentCost(indexS,2),tmp)=round(PVB(tmp,2)+rand*(PVB(tmp,2)-PVB(tmp,1)));
    end
    if rand<pro
        tmp=ceil(rand*NVAR);
        CB(agentCost(indexM,2),tmp)=round(PVB(tmp,2)+rand*(PVB(tmp,2)-PVB(tmp,1)));
    end
    for i=1:popSize
        for j=1:NVAR
            if CB(i,j)> PVB(j,2)
                CB(i,j)=PVB(j,2);
            end
            if CB(i,j)< PVB(j,1)
                CB(i,j)=PVB(j,1);
            end
        end
    end
    ITER=iterECBO+iterIHS;
    Merit(ITER)=cm(1,1);
    if pr==1
        if iterECBO==1 || iterECBO==pr50*50
            pr50=pr50+1;
            fprintf('iter(ECBO)=%4.0f Merit=%6.2f \n',iterECBO+iterIHS,Merit(end)) % Print results at each step
        end
    end
end % end "while iterECBO < MaxItrECBO"
%%%% end of Phase 2 %%%%
[a, b]=min(Merit);
hold on
plot(Merit,'Linewidth',2)
plot(b,a,'r*','Linewidth',2)
hold off
if Method==3
    fprintf('Phase2 = %4.0f iteration Merit=%6.2f \n',b-iterIHS,a)
else
    fprintf('iter(ECBO) = %4.0f iteration Merit=%6.2f \n',b,a)
end
end
function [weight, Stress, Disp] = TenBarTrussCase (A)
% Ten bar truss case 1

```

```

e=10*10^6;Ro=0.1; P = 100*10^3;
nodes = 360*[2, 1; 2, 0; 1, 1; 1, 0; 0, 1; 0, 0];
conn = [5,3; 3,1; 6,4; 4,2; 3,4; 1,2; 5,4; 6,3; 3,2; 4,1];
for j=1:2:3;
    for i=1:length(conn)
        if j==1
            lmm(i,j) =2*conn(i,j)-1;
            lmm(i,j+1)=2*conn(i,j);
        else
            lmm(i,j) =2*conn(i,j-1)-1;
            lmm(i,j+1)=2*conn(i,j-1);
        end
    end
end
K=zeros(2*length(nodes));
% Generate stiffness matrix for each element and assemble it.
for i=1:length(conn)
    lm=lmm(i,:);
    con=conn(i,:);
    k=PlaneTrussElement(e, A(i), nodes(con,:));
    K(lm, lm) = K(lm, lm) + k;
end

% Define the load vector
R = zeros(2*length(nodes),1); R(4)=-P; R(8)=-P;
% Nodal solution and reactions
[Disp, reactions] = NodalSoln(K, R, [9,10,11,12], zeros(4,1));
results=[];
for i=1:length(conn)
    results = [results; PlaneTrussResults(e, A(i), ...
        nodes(conn(i,:),:), Disp(lmm(i,:)))];
end
weight=Ro*360*(A(1)+A(2)+A(3)+A(4)+A(5)+A(6)+sqrt(2)*(A(7)+A(8)+A(9)+A(10)));
Stress=results(:,2);
end
function [d, rf] = NodalSoln(K, R, debc, ebcvals)
% [nd, rf] = NodalSoln(K, R, debc, ebcvals)
% Computes nodal solution and reactions
% K = global coefficient matrix
% R = global right hand side vector
% debc = list of degrees of freedom with specified values
% ebcvals = specified values
dof = length(R);
df = setdiff(1:dof, debc);
Kf = K(df, df);
Rf = R(df) - K(df, debc)*ebcvals;
dfvals = Kf\Rf;
d = zeros(dof,1);
d(debc) = ebcvals;
d(df) = dfvals;
rf = K(debc,:)*d - R(debc);
end

function k = PlaneTrussElement(e, A, coord)
% PlaneTrussElement(e, A, coord)

```

```

% Generates stiffness matrix for a plane truss element
% e = modulus of elasticity
% A = area of cross-section
% coord = coordinates at the element ends
x1=coord(1,1); y1=coord(1,2);
x2=coord(2,1); y2=coord(2,2);
L=sqrt((x2-x1)^2+(y2-y1)^2);
ls=(x2-x1)/L; ms=(y2-y1)/L;
k = e*A/L*[ls^2, ls*ms,-ls^2,-ls*ms;
           ls*ms, ms^2,-ls*ms,-ms^2;
           -ls^2,-ls*ms,ls^2,ls*ms;
           -ls*ms,-ms^2,ls*ms,ms^2];
end

function results = PlaneTrussResults(e, A, coord, disps)
% results = PlaneTrussResults(e, A, coord, disps)
% Compute plane truss element results
% e = modulus of elasticity
% A = Area of cross-section
% coord = coordinates at the element ends
% disps = displacements at element ends
% The output quantities are eps = axial strain
% sigma = axial stress and force = axial force.
x1=coord(1,1); y1=coord(1,2);
x2=coord(2,1); y2=coord(2,2);
L=sqrt((x2-x1)^2+(y2-y1)^2);
ls=(x2-x1)/L; ms=(y2-y1)/L;
T=[ls,ms,0,0; 0,0,ls,ms];
d = T*disps;
eps= (d(2)-d(1))/L;
sigma = e.*eps;
force = sigma.*A;
results=[eps, sigma, force];
end

```

APPENDIX B

MATLAB CODE FOR THE 18-BAR TRUSS DESIGN EXAMPLE

```

%% ECBO ESL for 18 bar truss dis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ESL=1; % Linear=1 % Best dynamic=2 % heaviest feasible dynamic=3
% / **** ECB ****/
% Initializing variables
NVAR=12;           % Number of design variables
popSize=40;        % Size of the population
cMs=0.1*popSize;   % Colliding memory size
maxIt=50;          % Max. number of interation (without ESL method)
pro=0.4;           % Pro prameter
% limits
MaxS=138;          % Max. stress (MPa)
Maxd=203;          % Max. displacement (mm)
% range of variables
Sections = linspace(1*100,150*100,100);
x=6.35*10^3/2; y=6.35*10^3/2; X = linspace(-x,x,100); Y = linspace(-y,y,100);
PVB=[1 length(Sections); 1 length(Sections); 1 length(Sections);...
1 length(Sections); 1 length(X); 1 length(Y); 1 length(X); 1 length(Y);...
1 length(X); 1 length(Y); 1 length(X); 1 length(Y)];
% random initial designs
for i=1:popSize
for j=1:NVAR
CB(i,j)=round(rand(1)*(PVB(j,2)-PVB(j,1))+PVB(j,1));
end
end
agentCost=zeros(popSize,2); % array of agent costs
% Colliding memory;
% The first column contains CB costs and the remaining columns include CB positions
Inf=1e100;          % infinity
cm=zeros(cMs,NVAR+1);
tm=zeros(2*cMs,NVAR+1); % Temporary memory
for e=1:cMs
cm(e,1)=Inf;
end
% Start iteration
iter=0;             % counter
while iter < maxIt
iter=iter+1;
% Evaluating the population
parfor e=1:popSize
DV=[Sections(CB(e,1:4)),X(CB(e,5:8)),Y(CB(e,9:12))];
[Mass, Stress, Max_d, nodes, Disp, Kf] = Eigh10BarTruss (DV);
Sum=0;
GM=[abs(Stress(:))/MaxS;abs(Disp(:))/Maxd];
for g=1:length(GM)

```

```

G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
merit(e)=Mass*(1+Sum)^2; % evaluating merit function for each agent
end %for
agentCost(:,1)=merit(:); agentCost(:,2)=1:popSize;
% Updating colliding memory
agentCost=sortrows(agentCost);
if iter>1
for e=1:cMs
agentCost(popSize-cMs+e,1)=cm(e,1);
for ee=1:NVAR
CB(agentCost(popSize-cMs+e,2),ee)=cm(e,ee+1);
end
end
end
for e=1:cMs
tm(e,1)=agentCost(e,1); tm(e+cMs,1)=cm(e,1);
for ee=1:NVAR
tm(e,ee+1)=CB(agentCost(e,2),ee); tm(e+cMs,ee+1)=cm(e,ee+1);
end
end
tm=sortrows(tm);
for e=1:cMs
cm(e,:)=tm(e,:);
end
agentCost=sortrows(agentCost);
% Evaluating the mass
mass=zeros(popSize,1);
for e=1:popSize
mass(e,:)=1/(agentCost(e,1));
end
% Updating CB positions
for e=1:popSize/2
indexS=e; % index of stationary bodies
indexM=popSize/2+e; % index of moving bodies
COR=(1-(iter/maxIt)); % coefficient of restitution
% velocity of moving bodies before collision
velMb=(((CB(agentCost(indexS,2),:)-CB(agentCost(indexM,2),:))));
% velocity of stationary bodies after collision
velSa=(((1+COR)*mass(indexM,1))/(mass(indexS,1)+mass(indexM,1))*velMb));
% velocity of moving bodies after collision
velMa=(((mass(indexM,1)-COR*mass(indexS,1))/(mass(indexS,1)+mass(indexM,1))*velMb));
CB(agentCost(indexM,2),1:4)=round(CB(agentCost(indexS,2),1:4)+2*(0.5-rand(1,4)).*velMa(1:4));
CB(agentCost(indexM,2),5:NVAR)=round(CB(agentCost(indexS,2),5:NVAR)+2*(0.5-rand(1,NVAR-4)).*velMa(5:NVAR));
CB(agentCost(indexS,2),1:4)=round(CB(agentCost(indexS,2),1:4)+2*(0.5-rand(1,4)).*velSa(1:4));
CB(agentCost(indexS,2),5:NVAR)=round(CB(agentCost(indexS,2),5:NVAR)+2*(0.5-rand(1,NVAR-4)).*velSa(5:NVAR));
if rand<pro

```

```

tmp=ceil(rand*NVAR);
CB(agentCost(indexS,2),tmp)=round(PVB(tmp,1)+rand*(PVB(tmp,2)-PVB(tmp,1)));
end
if rand<pro
tmp=ceil(rand*NVAR);
CB(agentCost(indexM,2),tmp)=round(PVB(tmp,1)+rand*(PVB(tmp,2)-PVB(tmp,1)));
end
for i=1:popSize
for j=1:NVAR
if CB(i,j)> PVB(j,2)
CB(i,j)=PVB(j,2);
end
if CB(i,j)< PVB(j,1)
CB(i,j)=PVB(j,1);
end
end
end
end
Merit(iter)=cm(1,1);
% Print results at each step
if iter==1 || iter==10 || iter==50 || iter==100 || iter==200 || iter==400 || iter==800
fprintf('iter=%d merit= %6.3f \n',iter,Merit(iter))
end
if iter==1
maxiter=600; % Max. number of iteration (ESL method)
maxCy=25; % Max. number of cycles
bestCost=cm(1,1);bestDesign=cm(1,2:end);
[CBes1,CMes1,bestD,bestM,MERIt,ESLIter]=ESLsubdis(CB,cm,bestDesign,bestCost,maxiter,maxCy,ESL);
CB=CBes1;
cm=CMes1;
end
end % while
[bestcost, interation]=min(Merit); bestDesign=cm(1,2:end);
plot(Merit)
Dnsa=popSize*interation; Lnsa=popSize*sum(ESLIter);
fprintf('BestCost= %6.3f(kg), Number of Dyn Ana.= %d, Number of Sta Ana.= %d, Time= %1.1f
(min)\n',...
bestcost, Dnsa, Lnsa, toc/60)

```

```

function [Mass, Stress, Max_d, nodes, d, Kf]=Eigh10BarTruss(DV)
% Transient analysis of a plane truss
global Mf Kf Rf
% g = 386.4;
e = 69*10^3; rho = 2765/1000^3; P=-59.3*10^3;
At = DV(1); Ab = DV(2); Av = DV(3); Ad = DV(4);
X = 6.35*10^3; Y = 6.35*10^3;
x3=X*4+DV(5);y3=Y*0+DV(6) ;x5=X*3+DV(7);y5=Y*0+DV(8); x7=X*2+DV(9);
y7=Y*0+DV(10); x9=X*1+DV(11);y9=Y*0+DV(12);
nodes = [X*5, Y*1; X*4, Y*1; x3, y3; X*3, Y*1; x5, y5; X*2, Y*1; x7, y7;...
X*1, Y*1; x9, y9; 0, Y*1; 0, 0];
conn = [1, 2; 1, 3; 2, 3; 2, 4; 3, 4; 3, 5; 4, 5; 4, 6; 5, 6; 5, 7; 6, 7; 6, 8; 7, 8;...
7, 9; 8, 9; 8, 10; 9, 10; 9, 11];

```

```

elems = size(conn,1);
lmm=[];
A=zeros(elems,1);
A(1)=At;A(4:4:16)=At; A(6:4:18)=Ab; A(2)=Ab; A(5:4:17)=Ad; A(3:4:15)=Av;
for i=1:elems
lmm = [lmm; [2*conn(i,1)-1, 2*conn(i,1),2*conn(i,2)-1, 2*conn(i,2)]];
end
debc = [19,20,21,22]; ebcVals=zeros(length(debc),1);
dof=2*size(nodes,1);
M=zeros(dof); K=zeros(dof);
R = zeros(dof,1); R(2)=P; R(4)=P; R(8)=P; R(12)=P; R(16)=P;
% Generate equations for each element and assemble them.
Mass=0;
for i=1:elems
con = conn(i,:);
lm = lmm(i,:);
[m, k, mass] = TransientPlaneTrussElement(e, A(i), rho, nodes(con,:));
M(lm, lm) = M(lm, lm) + m;
K(lm, lm) = K(lm, lm) + k;
Mass=Mass+mass;
end
% Adjust for essential boundary conditions
dof = length(R);
df = setdiff(1:dof, debc);
Mf = M(df, df);
Kf = K(df, df);
Rf = R(df) - K(df, debc)*ebcVals;
% Setup and solve the resulting first order differential equations
u0 = zeros(length(Mf),1);
v0 = zeros(length(Mf),1);
[t,d] = ode23('TrussODE',[0,8],[u0; v0]);
d=d(:,1:18);d(:,19:22)=0;
v=d(:,1:18);v(:,19:22)=0;
Max_S=0;
for i=1:elems
con = conn(i,:);
for j=1:length(t)
disps=d(j,lmm(i,:));
results = PlaneTrussResults(e, A(i), nodes(con,:), disps');
Stress(j,i)= results;
if abs(Stress(j,i))>Max_S
Max_S=abs(Stress(j,i));
a=j;b=i;
end
end
end
Max_d=max(max(abs(d)));
[Max,d_i]=max(abs(d));
[Max_d,d_j]=max(Max);

```

Function

```
[CBes], CMes], bestD, bestM, MERIT, Es], Iter]=ESLsubdis(CB, CM, bestDesign, MERIT, maxIt, maxCy, Method)
```



```

% This function is the ESL step
%%%% input
% CB: Colliding bodies matrix
% CM: Colliding memory matrix
% bestDesign: the best design what will be used to generate ESLs
% MERIT: merit value of the best design
% maxiter: maximum number of iterations for each cycle
% Method: ESL1, ESL2, or ESL3
%%%% output
% CBesl: the best colliding bodies matrix
% CMesl: the best colliding memory matrix
% bestD: matrix that stores the best design at the end of each cycle
% bestM: vector that stores the best merit value at the end of each cycle
% MERIT: best merit value
% EsIter: vector that stores number of iterations of each cycle
NVAR=12; popSize=40; cms=0.1*popSize; pro=0.4;
% limits
MaxS=138; Maxd=203;
% range of variables
Sections = linspace(1*100,150*100,100); x=6.35*10^3/2;y=6.35*10^3/2;
X = linspace(-x,x,100); Y = linspace(-y,y,100);
PVB=[1 length(Sections); 1 length(Sections); 1 length(Sections);...
1 length(Sections); 1 length(X); 1 length(Y); 1 length(X); 1 length(Y);...
1 length(X); 1 length(Y); 1 length(X); 1 length(Y)];
Design=[Sections(bestDesign(1:4)),X(bestDesign(5:8)),Y(bestDesign(9:12))];
[Mass, Stress, Max_d, nodes, d, K] = Eigh10BarTruss (Design);
CMesl=CM;
for cycles=1:maxCy
clear Merit; clear FDM
ESL=K*d(:,1:18)';
agentCost=zeros(popSize,2);
if cycles>1
for i=1:popSize
for j=1:NVAR
CB(i,j)=round(rand(1)*(PVB(j,2)-PVB(j,1))+PVB(j,1));
end
end
end
cm=CMesl; iter=0;
while iter < maxIt
iter=iter+1;
parfor e=1:popSize
[Mass, stress, disp]=StaticEigh10BarTruss([Sections(CB(e,1:4)),X(CB(e,5:8)),Y(CB(e,9:12))],ESL);
GM=[abs(stress)/MaxS;abs(disp)/Maxd];
Sum=0;
for g=1:length(GM)
G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
merit(e)=Mass*(1+Sum)^2;

```

```

end
agentCost(:,1)=merit(:); agentCost(:,2)=1:popSize; agentCost=sortrows(agentCost);
if iter>1
for e=1:cMs
agentCost(popSize-cMs+e,1)=cm(e,1);
for ee=1:NVAR
CB(agentCost(popSize-cMs+e,2),ee)=cm(e,ee+1);
end
end
end
for e=1:cMs
tm(e,1)=agentCost(e,1); tm(e+cMs,1)=cm(e,1);
for ee=1:NVAR
tm(e,ee+1)=CB(agentCost(e,2),ee); tm(e+cMs,ee+1)=cm(e,ee+1);
end
end
tm=sortrows(tm);
for e=1:cMs
cm(e,:)=tm(e,:);
end
agentCost=sortrows(agentCost); mass=zeros(popSize,1);
for e=1:popSize
mass(e,:)=1/(agentCost(e,1));
end
for e=1:popSize/2
indexS=e; indexM=popSize/2+e;
COR=(1-(iter/maxIt));
ve1Mb=(((CB(agentCost(indexS,2),:)-CB(agentCost(indexM,2),:)))));
ve1Sa=(((1+COR)*mass(indexM,1))/(mass(indexS,1)+mass(indexM,1))*ve1Mb));
ve1Ma=(((mass(indexM,1)-COR*mass(indexS,1))/(mass(indexS,1)+mass(indexM,1))*ve1Mb));
CB(agentCost(indexM,2),1:4)=round(CB(agentCost(indexS,2),1:4)+2*(0.5-rand(1,4)).*ve1Ma(1:4));
CB(agentCost(indexM,2),5:NVAR)=round(CB(agentCost(indexS,2),5:NVAR)+2*(0.5-rand(1,NVAR-
4)).*ve1Ma(5:NVAR));
CB(agentCost(indexS,2),1:4)=round(CB(agentCost(indexS,2),1:4)+2*(0.5-rand(1,4)).*ve1Sa(1:4));
CB(agentCost(indexS,2),5:NVAR)=round(CB(agentCost(indexS,2),5:NVAR)+2*(0.5-rand(1,NVAR-
4)).*ve1Sa(5:NVAR));
if rand<pro
tmp=ceil(rand*NVAR);
CB(agentCost(indexS,2),tmp)=round(PVB(tmp,1)+rand*(PVB(tmp,2)-PVB(tmp,1)));
end
if rand<pro
tmp=ceil(rand*NVAR);
CB(agentCost(indexM,2),tmp)=round(PVB(tmp,1)+rand*(PVB(tmp,2)-PVB(tmp,1)));
end
for i=1:popSize
for j=1:NVAR
if CB(i,j)> PVB(j,2)
CB(i,j)=PVB(j,2);
end
if CB(i,j)< PVB(j,1)
CB(i,j)=PVB(j,1);
end
end

```

```

end
end
end
Merit(iter)=cm(1,1);
% Stopping criteria
if iter>0.25*maxIt
if (Merit(iter-round(0.1*maxIt))-Merit(iter))/Merit(iter) <=10^-3
break
end
end
end
DV=[Sections(bestDesign(1:4)),X(bestDesign(5:8)),Y(bestDesign(9:12))];
bestDesign=cm(1,2:end); bestcost=cm(1,1);
if Method==2 || Method==3
AA=[cm(:,2:end);CB(1:10,:)];
fd=0;
for i=1:10+cms
DV=[Sections(AA(i,1:4)),X(AA(i,5:8)),Y(AA(i,9:12))];
[Mass, Stressnew,Max_d, nodes, dnew, Knew]=Eigh10BarTruss(DV);
GM=[abs(Stressnew(:))/MaxS;abs(dnew(:))/Maxd];
Sum=0;
for g=1:length(GM)
G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
Meritt(i)=Mass*(1+Sum)^2;
if Sum==0
fd=fd+1;
FD(fd,:)=DV; FDM(fd,1)=Meritt(i);
end
end
AC(:,1)=Meritt;AC(:,2)=1:10+cms; AC=sortrows(AC); MeriT=AC(1,1);
if Method==2
Design=[Sections(AA(AC(1,2),1:4)),X(AA(AC(1,2),5:8)),Y(AA(AC(1,2),9:12))];
else
if fd>0
FDM(:,2)=1:fd; FDM=sortrows(FDM); Design=FD(FDM(end,2),:);
else
Design=[Sections(AA(AC(1,2),1:4)),X(AA(AC(1,2),5:8)),Y(AA(AC(1,2),9:12))];
end
end
bestD(cycles,:)=Design;
[Mass, Stressnew,Max_d, nodes, dnew, Knew]=Eigh10BarTruss(Design);
elseif Method==1
Design=[Sections(bestDesign(1:4)),X(bestDesign(5:8)),Y(bestDesign(9:12))];
bestD(cycles,:)=bestDesign;
[Mass, Stressnew,Max_d, nodes, dnew, Knew]=Eigh10BarTruss(Design);
GM=[abs(Stressnew(:))/MaxS;abs(dnew(:))/Maxd];
Sum=0;
for g=1:length(GM)

```

```

G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
Merit=Mass*(1+Sum)^2;
else
AA=[cm(:,2:end);CB(1:10,:)];
fd=0;
for i=1:10+cms
DV=[Sections(AA(i,1:4)),X(AA(i,5:8)),Y(AA(i,9:12))];
[Mass, stress, disp]=StaticEigh10BarTruss(DV,ESL);
GM=[abs(stress)/MaxS;abs(disp)/Maxd];
Sum=0;
for g=1:length(GM)
G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
Merit(i)=Mass*(1+Sum)^2;
if Sum==0
fd=fd+1; FD(fd,:)=DV; FDM(fd,1)=Merit(i);
end
end
AC(:,1)=Merit(:); AC(:,2)=1:10+cms; AC=sortrows(AC);
if fd>0
FDM(:,2)=1:fd; FDM=sortrows(FDM); Design=FD(FDM(end,2),:);
else
Design=[Sections(bestDesign(1:4)),X(bestDesign(5:8)),Y(bestDesign(9:12))];
end
bestD(cycles,:)=Design;
[Mass, Stressnew,Max_d, nodes, dnew,
Knew]=Eigh10BarTruss([Sections(bestDesign(1:4)),X(bestDesign(5:8)),Y(bestDesign(9:12))]);
GM=[abs(Stressnew(:))/MaxS;abs(dnew(:))/Maxd];
Sum=0;
for g=1:length(GM)
G=GM(g)-1;
if G>0
Sum=Sum+G;
end
end
Merit=Mass*(1+Sum)^2;
[Mass, Stressnew,Max_d, nodes, dnew, Knew]=Eigh10BarTruss(Design);
end
fprintf('Cycle= %d iter= %d Merit= %4.2f \n',cycles,iter,Merit)
clear d; clear K
d=dnew; K=Knew; bestD(cycles,:)=Design; bestM(cycles)=Merit;EsIter(cycles)=iter;
if (Merit < MERIT)
MERIT=Merit;
if Method==1
CMesl=cm; CBesl=CB;

```

```

else
CBes1=CB; CMes1(:,1)=AC(1:cms,1); CMes1(:,2:end)=AA(AC(1:cms,2),:);
end
end
if cycles>=5
if MERIT>=bestM(end)
if bestM(end)==bestM(end-2)
break
end
else
if MERIT<bestM(end) & MERIT<bestM(end-1)
break
end
end
end
end
end
end
end

```

```

function sigma = PlaneTrussResults(e, A, coord, disps)
% results = PlaneTrussResults(e, A, coord, disps)
% Compute plane truss element results
% e = modulus of elasticity
% A = Area of cross-section
% coord = coordinates at the element ends
% disps = displacements at element ends
% The output quantities are eps = axial strain
% sigma = axial stress and force = axial force.
x1=coord(1,1); y1=coord(1,2); x2=coord(2,1); y2=coord(2,2);
L=sqrt((x2-x1)^2+(y2-y1)^2); ls=(x2-x1)/L; ms=(y2-y1)/L;
T=[ls,ms,0,0; 0,0,ls,ms];
d = T*disps;
eps= (d(2,:)-d(1,:))/L; sigma = e.*eps;

```

```

function [Mass, stress, d]=StaticEigh10BarTrusstest(DV,ESL)
e = 69*10^3; rho = 2765/1000^3;
At = DV(1); Ab = DV(2); Av = DV(3); Ad = DV(4);
X = 6.35*10^3; Y = 6.35*10^3;
x3=X^4+DV(5);y3=Y^0+DV(6) ;x5=X^3+DV(7);y5=Y^0+DV(8); x7=X^2+DV(9);
y7=Y^0+DV(10); x9=X^1+DV(11);y9=Y^0+DV(12);
nodes = [X^5, Y^1; X^4, Y^1; x3, y3; X^3, Y^1; x5, y5; x^2, Y^1; x7, y7;...
X^1, Y^1; x9, y9; 0, Y^1; 0, 0];
conn = [1, 2; 1, 3; 2, 3; 2, 4; 3, 4; 3, 5; 4, 5; 4, 6; 5, 6; 5, 7; 6, 7; 6, 8; 7, 8;...
7, 9; 8, 9; 8, 10; 9, 10; 9, 11];
elems = size(conn,1);
lmm=[];
A=zeros(elems,1);
A(1)=At;A(4:4:16)=At; A(6:4:18)=Ab; A(2)=Ab; A(5:4:17)=Ad; A(3:4:15)=Av;
for i=1:elems
lmm = [lmm; [2*conn(i,1)-1, 2*conn(i,1),2*conn(i,2)-1, 2*conn(i,2)]];
end
debc = [19,20,21,22]; ebcVals=zeros(length(debc),1);
dof=2*size(nodes,1);
M=zeros(dof); K=zeros(dof);

```

```

[ies1,jes1]=size(ESL);
R = zeros(dof,jes1) ; R(1:ies1,:)=ESL;
% Generate equations for each element and assemble them.
Mass=0;
for i=1:elems
con = conn(i,:);
lm = lmm(i,:);
[m, k, mass] = TransientPlaneTrussElement(e, A(i), rho, nodes(con,:));
M(lm, lm) = M(lm, lm) + m;
K(lm, lm) = K(lm, lm) + k;
Mass=Mass+mass;
end
% Adjust for essential boundary conditions
df = setdiff(1:dof, debc);
Mf = M(df, df); Kf = K(df, df); Rf = R(df,:);
d=lin_solve(Kf,Rf);d(19:22,:)=0;
for i=1:elems
results(i,:) = PlaneTrussResults(e, A(i), nodes(conn(i,:),:), d(lmm(i,:),:));
end
stress=results(:); d=d(:);

```

```

function [m, k, mass] = TransientPlaneTrussElement(e, A, rho, coord)
% Generates mass & stiffness matrices for a plane truss element
% rho = mass density
% e = modulus of elasticity
% A = area of cross-section
% coord = coordinates at the element ends
x1=coord(1,1); y1=coord(1,2);
x2=coord(2,1); y2=coord(2,2);
L=sqrt((x2-x1)^2+(y2-y1)^2);
ls=(x2-x1)/L; ms=(y2-y1)/L;
k = e*A/L*[ls^2, ls*ms, -ls^2, -ls*ms;
ls*ms, ms^2, -ls*ms, -ms^2;
-ls^2, -ls*ms, ls^2, ls*ms;
-ls*ms, -ms^2, ls*ms, ms^2];
mass=rho*A*L;
m = ((mass)/6)*[2, 0, 1, 0; 0, 2, 0, 1;
1, 0, 2, 0; 0, 1, 0, 2];

```

```

function ddot = TrussODE(t, d)
% function to set up equations for a transient truss problem
global Mf Kf Rf
if t>=2
ft=0;
else
ft=sin(pi*t/2);
end
n=length(d);
u = d(1:n/2); v = d(n/2+1:n);
vdot = inv(Mf)*(Rf*ft - Kf*u);
udot = v;
ddot = [udot; vdot

```