LAYOUT OPTIMIZATION OF TRUSS STRUCTURES BY FULLY STRESSED DESIGN EVOLUTION STRATEGY

By

Ali Ahrari

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering-Doctor of Philosophy

2016

ABSTRACT

LAYOUT OPTIMIZATION OF TRUSS STRUCTURES BY FULLY STRESSED DESIGN EVOLUTION STRATEGY

By

Ali Ahrari

The field of structural optimization has gained much academic interest in the recent decades. Different streams of optimization methods have been applied to this problem including analytical methods, optimality criteria-based method and gradient-based methods. During the recent decade, there has been a growing interest among researchers to apply stochastic population-based methods, the so-called meta-heuristics, to this class of optimization problems. The motivation is the robustness and capability of meta-heuristics to avoid local minima. On the downside, their required evaluation budget grows fast when the number of design variables is increased, which limits the complexity of problems to which they can be applied. Furthermore, majority of these methods are tailored to optimize only the cross-sectional areas of the members, the potential saving from which is highly limited. At the same time, several factors have diminished practitioners' interests in the academic research on this topic, including simplicity of conventional test problems compared to real structures, variety of design constraints in practice and the complexity of evaluation of the total cost.

This dissertation aims at addressing some of the most critical shortcomings in the available truss optimization methods, both from academic and practical perspectives. It proposes a novel bi-level method for simultaneous optimization of topology, shape and size of truss structures. In the upper level, a specialized evolution strategy (ES) is proposed which follows the principles of contemporary evolution strategies (ESs), although the formulation is modified to handle mixed-

variable highly constrained truss optimization problems. The concept of fully stressed design is employed in the lower level as an efficient method for resizing the sampled solution in the upper level. The concept of fully stressed design is also utilized to define a specialized penalty term based on the estimated required increase in the structural weight such that all constraints are satisfied.

The proposed method, called fully stressed design evolution strategy (FSD-ES), is developed in four stages. It is tested on complicated problems, some of which are developed in this dissertation, as an attempt to reduce the gap between complexity of test problems and real structures. Empirical evaluation and comparison with the best available methods in the literature reveal superiority of FSD-ES, which intensifies for more complicated problems.

Aside from academically interesting features of FSD-ES, it addresses some of the practicing engineers' critiques on applicability of truss optimization methods. FSD-ES can handle large-scale truss optimization problems with more than a thousand design parameters, in a reasonable amount of CPU time. Our numerical results demonstrate that the optimized design can hardly be guessed by engineering intuition, which demonstrates superiority of such design optimization methods. Besides, the amount of material saving is potentially huge, especially for more complicated problems, which justifies simulation cost of the design problem. FSD-ES does not require any user-dependent parameter tuning and the code is ready to use for an arbitrary truss design problem within the domain of the code.

To my parents, for their unconditional support

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Kalyanmoy Deb. I had the privilege to have his support and guidance during my Ph.D. studies. We also collaborated on a few other optimization projects in parallel.

I also would like to thank my committee members,

Professor Erik Goodman,

Professor Xiaobo Tan,

Dr. Ronald Averill,

For their insightful recommendations, which helped me improve my dissertation.

Finally, I would like to thank BEACON, the Center for the Study of Evolution in Action, for funding my research.

TABLE OF CONTENTS

LIST OF TABLES ..... **ix**

LIST OF FIGURES ..... **xi**

KEY TO SYMBOLS ..... **xiv**

KEY TO ABBREVIATIONS ..... **xvii**

CHAPTER 1. INTRODUCTION **1**
1.1. Different Types of Truss Optimization 2
1.2. Academia Versus Practice 5
1.3. Discrete Versus Continuum Consideration of Topology 6
1.4. Failure Criteria in Design 9
1.5. Bi-level Optimization 12
1.6. Contribution of This Dissertation 13

CHAPTER 2. EVOLUTION STRATEGIES **15**
2.1. Recombination Operator 17
2.2. Selection Operator 18
2.3. Mutation Operator 19
2.4. Adjusting the Mutation Strength 20
2.5. State-of-the-Art Evolution Strategies 22
2.6. Handling Discrete Variables 24

CHAPTER 3. RELATED STUDIES **25**
3.1. Fully Stressed Design (FSD) 26
3.2. Metaheuristics for Truss Optimization 29
3.2.1. Topology, Shape and Size Optimization: Challenges and Alternatives 30
3.2.2. Evolution Strategies for Truss Optimization 32

CHAPTER 4. FULLY STRESSED DESIGN EVOLUTION STRATEGY **35**
4.1. Algorithm Details 38
4.1.1. Notation 38
4.1.2. Initial Solution 38
4.1.3. Mutating Shape Variables 39
4.1.4. Mutating Size Variables 39
4.1.5. Evaluation 40
4.1.6. Resizing 43
4.1.7. Recombination 45
4.1.8. Update of Parameters 46
4.1.9. Parameter Tuning 47

4.1.10. Flowchart of the Proposed Algorithm 47
4.2. Numerical Evaluation 47
4.2.1. Test Problems 48
4.2.2. Performance Measures 51
4.2.3. Results and Discussion 54
4.2.4. Importance of Problem Specific Knowledge 58

CHAPTER 5. FULLY STRESSED DESIGN EVOLUTION STRATEGY FOR SIMULTANEOUS TOPOLOGY, SHAPE AND SIZE OPTIMIZATION 61
5.1. Algorithm Details 62
5.1.1. Problem Representation 62
5.1.2. Initial Values 64
5.1.3. Mutating Topology Variables 64
5.1.4. Mutating Shape Variables 65
5.1.5. Mutating Size Variables 66
5.1.6. Evaluation 66
5.1.7. Resizing 67
5.1.8. Recombination 68
5.1.9. Updating Parameters 70
5.1.10. Parameter Tuning 71
5.1.11. Flowchart of the Proposed Algorithm 72
5.2. Numerical Evaluation 73
5.2.1. Test Problems 73
5.2.2. Performance Measures 78
5.2.3. Results and Discussion 78

CHAPTER 6. IMPROVED FULLY STRESSED DESIGN EVOLUTION STRATEGY .. 87
6.1. Algorithm Details 90
6.1.1. Problem Representation 90
6.1.2. ES-based Sampling of New Designs 90
6.1.3. Design Evaluation 92
6.1.4. Resizing 95
6.1.4.1. Stress-Based Resizing 95
6.1.4.2. Displacement-Based Resizing 97
6.1.5. Updating Parameters 99
6.1.5.1. Updating Mutation Parameters 99
6.1.5.2. Updating the Recombinant Design 100
6.1.5.3. Updating the Penalty Coefficients 101
6.1.5.4. Controlling the Move Limit 102
6.1.6. Stopping Criteria and Parameter Tuning 103
6.1.7. Flowchart of the Proposed Algorithm 105
6.2. Numerical Evaluation of FSD-ES II 106
6.2.1. Test Problems 107
6.2.1.1. 47-Bar Transmission Tower 107
6.2.1.2. 68-Bar Truss Problem 109
6.2.1.3. 110-Bar Transmission Tower 109

6.2.1.4. 224-Bar Pyramid 110
6.2.1.5. Bridge Design Problem 111
6.2.2. Performance Measures 115
6.2.3. Results and Discussion 116

CHAPTER 7. FULLY STRESSED DESIGN EVOLUTION STRATEGY WITH ARBITRARY RESIZING BUDGET 123
7.1. New Features 124
7.1.1. Controlling the Lower Loop Budget 125
7.1.2. Biasing the Cross-Sectional Areas 126
7.1.3. Parameter Setting 126
7.2. Numerical Evaluation 127
7.2.1. Size Optimization of 960-Bar Double Grid 128
7.2.2. Physical Design Area Problem 129
7.3. Results and Discussion 131

CHAPTER 8. SUMMARY, CONCLUSIONS AND FUTURE RESEARCH 139

REFERENCES 145

LIST OF TABLES

Table 1. Simulation data for the 18-bar truss problem 49

Table 2. Data for simulation of the 47-bar truss problem..... 50

Table 3. Summary of the best results available in the literature for each problem. For FSD-ES, FEs for two selected target weights are reported 56

Table 4. Data for the best feasible solution found for each problem..... 57

Table 5. Simulation data for the 15-bar truss problem 74

Table 6. Simulation data for the 25-bar spatial truss problem..... 75

Table 7. Data for simulation of the 39-bar truss problem..... 76

Table 8. Data for simulation of the 45-bar truss problem..... 77

Table 9. Data for simulation of the 68-bar truss problem..... 78

Table 10. Comparison of the results from FSD-ES and the best available results in the literature. For FSD-ES, FEs for two selected target weights are reported..... 83

Table 11. The best solutions found for the test problems using FSD-ES. Coordinates and areas are in inch and inch square, respectively. The ratio of the maximum stress, buckling load and displacement to the allowable limit as well as the overall weight are provided in the four last rows..... 85

Table 12. Simulation Data for the 47-bar truss problem 108

Table 13. Simulation Data for the 110-bar truss problem 110

Table 14. Simulation data for the 224-bar pyramid..... 111

Table 15. Simulation Data for the 277-bar bridge design problem 113

Table 16. Default parameter setting for the test problems determined using equations 7 and 8..... 116

Table 17. Parameters of the best solution found for the 77-bar, 224-bar and 277-bar problems..... 121

Table 18. Parameters of the best solution found for the 47-bar, 68-bar and 110-bar truss problems 122

Table 19. Calculation of default values of FSD-ES IIb parameters using Equation 56 132

Table 20 FEs, SR and ERT for some selected values of W_{target} when N_{resize}=3..... 137

Table 21 Data for the best solution found for the 759-bar problem in Variant I (Weight=27,967 Kg) 138

LIST OF FIGURES

Figure 1. A couple of familiar examples for truss structures a) an electricity transmission line pylon and b) a tower crane 2

Figure 2. Different aspects of truss optimization: a) design requirements b) topologically distinct designs can be selected to address the problem c) different shape can be selected for a given topology d) different sizing can be performed for a given shape and topology. 4

Figure 3. A typical optimized solution using a) continuum, b) discrete ground structure. 6

Figure 4. Iso-density contours for a) Isotropic Mutation with one free parameter b) D free parameters and c) N×(N+1)/2 free parameters 20

Figure 5. Flowchart of FSD-ES 48

Figure 6. Ground structure for the 18-bar truss: a= 250 in. 49

Figure 7. Ground structure of the 47-bar power-line problem. The illustration is rotated for better use of space. 49

Figure 8. Ground structure for the 77-bar truss bridge problem 51

Figure 9. ERT, SR and FE_S to reach arbitrary structural weights for the employed test problems: a) 18-bar truss, b) 47-bar truss and c) 77-bar truss bridge problems 56

Figure 10. The best feasible solution found for each problem: a) 18-bar, b) 47-bar and c) 77-bar truss problems 58

Figure 11. Performance of the different variants of FSD-ES on the 47-bar problem a) ERT and b) SR 60

Figure 12. Flowchart for FSD-ES for TSS optimization 73

Figure 13. Ground structure of the 150-bar problem 74

Figure 14. Ground structure of the 3D 25-bar problem 75

Figure 15. Ground structure of the two-tier 39-bar truss 76

Figure 16. Ground structure of the 45-bar truss problem 77

Figure 17. Ground structure of 68-bar truss problem 77

Figure 18. ERT, SR and FE_S to reach arbitrary structural weights for the employed test problems: a) 15-bar, b) 25-bar, c) 39-bar, d) 45-bar and e) 68-bar truss problems. 82

Figure 19. The best final design for a) 15-bar, b) 25-bar (front and side view), c) 39-bar, d) 45-bar and e) 68-bar truss problems. The overlapping member is depicted with curved line segment for the 45-bar problem. 84

Figure 20. Some selected final designs for the 39-bar truss problem which have distinct topology: Topology #2: W=181.02 lb, Topology #3: W=181.38 lb, Topology #4: W=181.60 lb, Topology #7: W=182.37 lb, Topology #8:W=183.34 lb, Topology #10: W=183.89 lb, Topology #13: W=186.91 lb, Topology #15: W=186.96 lb, Topology #18: W=187.30 lb. 86

Figure 21. Flowchart of FSD-ES II..... 106

Figure 22. Ground structure for the 47-bar transmission tower problem 108

Figure 23. Ground structure for the a) 110-bar and b) 224-bar (front and top view) test problems..... 109

Figure 24. a) The proposed module for the bridge design problem. The proposed module can conform to different models such as b) Bailey c) Pratt and d) K-truss. e) The ground structure is posed by joining 10 of these modules side by side. For esthetics, some members of the first and the last modules were removed. 112

Figure 25. FE_S, SR and ERT as a function of the target weight (W_{target}) for a) 47-bar, b)68-bar, c) 110-bar, d) 224-bar, e) 277-bar (Variant I), f) 277-bar (Variant II) and g) 277-bar (Variant III) test problems. 119

Figure 26. The best feasible solution found for different test problems. a) 47-bar (W=1727.6 lb), b) 110-bar (W=1314.0 lb), c) 224-bar (W=3079.4 Kg), d) 68-bar (W=1166.1 lb), e) 277-bar in case I (W=282.03 kip), f) 277-bar in case II (W=236.54 kip), g) 277-bar in case III (W=231.94 kip)..... 120

Figure 27. Flowchart of FSD-ES IIb. The blue dotted rectangle specifies iterative resizing defined in FSD-ES IIb. 125

Figure 28. Ground structure for the 960-bar problem (front, side and top views) 129

Figure 29. Illustration of the 759-bar physical design problem..... 130

Figure 30. Convergence history (best penalized weight and best feasible weight versus FEs) in Variant for different values of N_{resize}. 132

Figure 31. ERT and SR as a function of N_{resize} for the 960-bar problem for some selected values of W_{target} 136

Figure 32. ERT and SR as a function of N_{resize} for the 759-bar problem for some selected values of W_{target}: a) Variant I and b) Variant II 137

Figure 33. The best solution found by FSD-ES IIb with N_{resize}=3 for the 759-bar problem in Variant I 138

KEY TO SYMBOLS

$\boldsymbol{A}$	Vector of size variables
c_A	Move limit controlling parameter
CE	Cost Effectiveness
D	Space of the structure (2 for planar and 3 for spatial structures)
E	Modulus of elasticity
$\boldsymbol{f}$	Ratio of the buckling load in the members to the allowable limit in the most critical load case
$f(\boldsymbol{\theta})$	Objective function (Penalized weight)
$FEpG$	Number of function evaluations per generation
f_{ik}	Axial force in the i-th member when the unit load is applied to the k-th degree of freedom of the structure
F_{il}	Axial force in the i-th member under the l-th load
$\boldsymbol{F_y}$	Allowable stress
$\mathbf{K}$	Stiffness matric of the structure
$\boldsymbol{M}$	Vector of topology variables
$MaxIter$	Maximum number of iterations
N	Number of decision parameters
N_{CON}	Number of constraints in the problem
N_{eff}	Effective number of independent design parameters
N_l	Number of load cases applied to the structure
N_m	Number of members in the structure
N_n	Number of nodes in the structure
N_{resize}	Maximum number of resizing in the lower level per design

N_{shape}	Number of independent shape variables
N_{size}	Number of independent size variables
N_{top}	Number of independent topology variables
null(**K**)	Calculate the nullity of a matrix **K**
N_{VAR}	Overall number of independent design parameters
$\boldsymbol{\mathcal{N}}_{\boldsymbol{DN_n}}(0,1)$	Vector of $D \times N_n$ independent random numbers sampled from the standard normal distribution.
P_i or c_{Pi}	Penalty coefficient for the i-th member
sgn	The standard sign function
$\boldsymbol{s}$	Ratio of axial stress in the members to the allowable limit in the most critical load case
$\boldsymbol{u}$	Vector of displacement of the nodes
$\boldsymbol{u}$	Ratio of nodal displacements to the allowable limit in the most critical load case
W	Raw Structural weight
w_i	Recombination weight for the i-th parent
$\boldsymbol{X}$	Vector of shape variables
$\boldsymbol{Z}$	Variation vector applied to the recombinant design
$\boldsymbol{\sigma}$	Ratio of compressive axial load in the members to the allowable limit in the most critical load case
$\boldsymbol{\sigma}_{\text{mean}}$	Vector of global step sizes
$\boldsymbol{\theta}$	A candidate solution
$\boldsymbol{\theta}_{\text{mean}}$	Intermediate (recombinant) design
$\boldsymbol{\theta}^{\text{up}}$	Upper bound of design parameters
$\boldsymbol{\theta}^{\text{low}}$	Lower bound of design parameters
λ	Population size

μ	Parent size
ρ	Density of the structure material
$\boldsymbol{\tau}$	Learning rate for the global step size
$\boldsymbol{\tau}_{\mathrm{C}}$	Adaptation time interval for the covariance matrix
$\otimes$	Element-wise multiplication

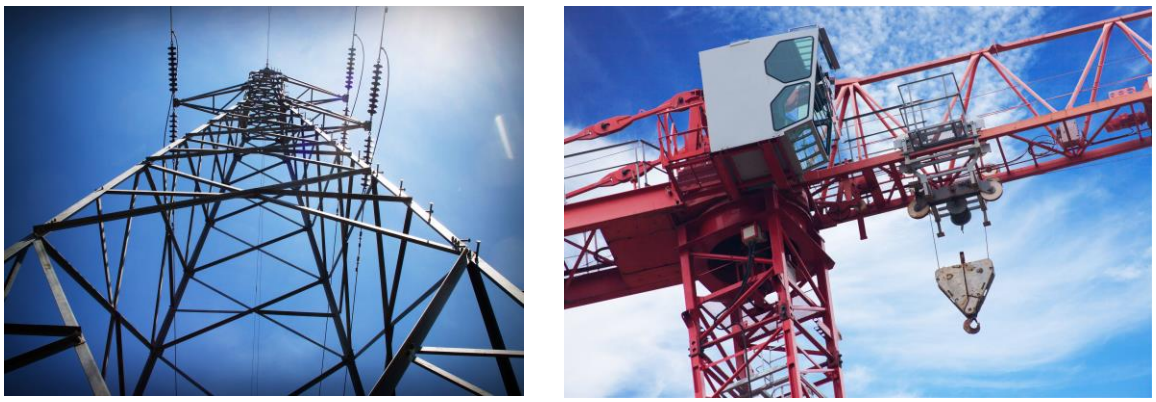
KEY TO ABBREVIATIONS

AISC-ASD	American Institute of Steel Construction -Allowable Stress Design
BBOB	Black-Box Optimization Benchmarking
CEC	Congress on Evolutionary Computation
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CMSA-ES	Covariance Matrix Self-Adaptation Evolution Strategy
CSA	Cumulative Step-size Adaption
EA	Evolutionary Algorithm
ERT	Expected Running Time
ES(s)	Evolution Strategy(ies)
ESO	Evolutionary Structural Optimization
FE	Function Evaluation
FSD	Fully Stressed Design
GECCO	Genetic and Evolutionary Computation Conference
SIMP	Solid Isotropic Microstructures with Penalization
SR	Success Rate
TSS	Topology, Shape and Size

CHAPTER 1. INTRODUCTION

Truss structures are 2D (planar) or 3D (spatial) structures commonly found in aerospace and civil structures such as bridges, cranes, roofs and towers (Figure 1). They consist of pin-connected rods which undergo axial loading when the external loads are applied. Since a rod is much stronger in axial loading than bending, truss structures are an ideal alternative when a minimal weight is desired.

This chapter highlights importance and different aspects of truss optimization. It discusses two distinct trends in structural optimization. Factors that have prevented widespread application of truss optimization methods in practice are discussed. Finally, the contributions of this dissertation are highlighted.



(a) (b)

Figure 1. A couple of familiar examples for truss structures a) an electricity transmission line pylon and b) a tower crane

1.1. Different Types of Truss Optimization

Design of a truss structure can be a tedious task especially if it consists of many bars, which we call members, and joints, commonly referred to as nodes. Figure 2(a) illustrates a simple truss design problem where the structure should carry a vertical load at point 1, while it is anchored at points 2 and 3. Locations of these three nodes are defined by the problem requirements, and thus

they must be present in the final design. Such nodes are usually referred to as the *basic nodes*. A designer may think of several different connectivity plots (topology) to address this issue, such as those illustrated in Figure 2(b). For a given topology, the designer should decide on the coordinates of the nodes, which configure the shape of the truss (Figure 2(c)). Finally, the cross-sections of the members should be selected such that the structure can carry the applied external load, which is known as sizing (Figure 2(d)).

Failure may occur if the stress in a member or displacement of a node exceeds the allowable limit. In practice, more sophisticated failure criteria may govern, and besides, a structure is usually supposed to carry multiple loading conditions (load cases). The best design not only satisfies these constraints, but also optimizes some objectives such as the weight, esthetics, production and maintenance cost. Consequently, finding the best design is a challenging task since it requires deciding on many parameters such that the design goals are fulfilled while many constraints are satisfied.

Because of the complexity of finding the optimum design, a lot of research on utilization of optimization technique in this field has been conducted during the past decades. The early academic research was based on the Mitchel's theorem [1]; however, application of this theorem is limited for structures subjected to one load case and besides, it requires appropriate specification of the strain field. Moreover, the theorem results in structures with infinite number of members having non-standard length, which is not practically acceptable [2]. Alternatively, methods based on mathematical programming were later developed and successfully applied to truss optimization [2]. Subsequently, advent of evolutionary algorithms (EAs) and other population based metaheuristic methods provided a new stream in truss optimization. A huge number of the recent studies focus on application of different metaheuristics for truss optimization.

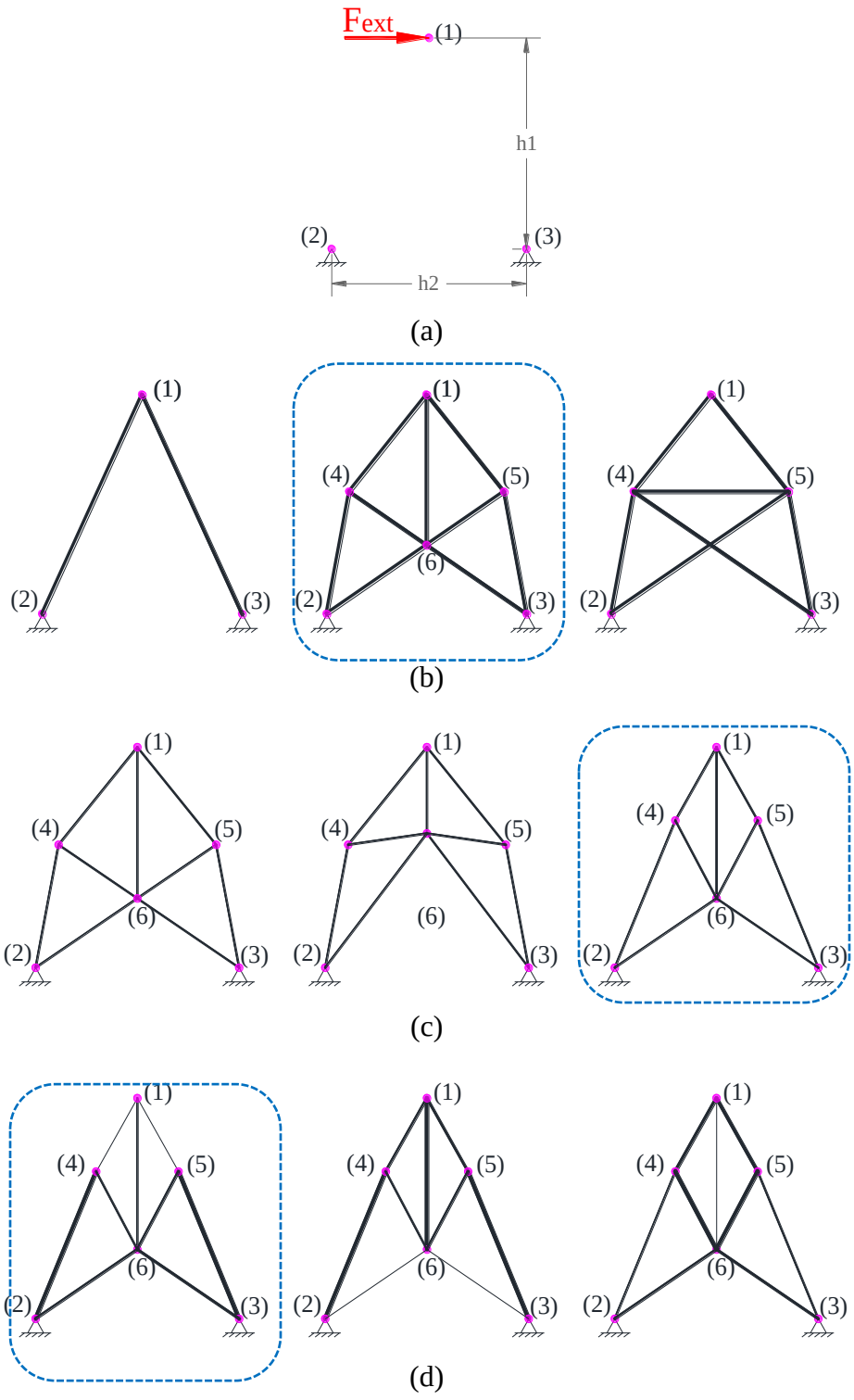


Figure 2. Different aspects of truss optimization: a) design requirements b) topologically distinct designs can be selected to address the problem c) different shape can be selected for a given topology d) different sizing can be performed for a given shape and topology.

1.2. Academia Versus Practice

Despite such a huge number of academic studies on structural optimization, practitioners still prefer intuition-based trial-and-error methods to such rigorous optimization algorithms [3, 4]. There are several factors accountable for such a gap between the amount of academic research and practicing engineers' preferred methods:

- Practitioner believe that the benefits of using optimization algorithms do not pay off the effort required for formulation of the design problem [3].
- Designers prefer to rely on methodologies on which they have a good understanding [3].
- It is difficult to isolate a part of a large structure for optimization; therefore, all parameters in the structure should be considered at the same time [3].
- Most conventional methods minimize the structural weight, while from practitioners' point of view, the cost should be minimized, which is affected by other factors such as constructability, repetition of design (modularity) and material procurement [3][1].
- The academic research lacks practical benchmark problems which may reliably simulate the complexity of practical problems. This complexity may be caused by the problem assumptions, constraints, objectives as well as the number of design parameters [4].
- Satisfaction of different disciplines whose decisions affect optimality of a design may result in successive revisions of the initial design [4].

This lingering gap between truss optimization research in academia and practitioners' preference is questionable, especially considering that the goal of the academic research should be addressing the practical needs.

[1] There are a few studies focusing on such constrains.

1.3. Discrete Versus Continuum Consideration of Topology

Truss topology optimization is commonly performed using the concept of *ground structure* [2], in which the optimum topology is chosen as a subset of an excessively connected structure. The topology optimization problem can be continuum or discrete, depending on the ground structure [5, 6]. In continuum topology optimization, the ground structure is a 2D or 3D region, which is discretized to small elements (Figure 3(a)). Topology optimization removes unnecessary material such that the objective function is minimized. In discrete topology optimization, the ground structure consists of many nodes connected by an excessive number of members. The optimization algorithm selects a subset of these members and nodes such that the objective function is optimized. Shape and size optimization are performed by optimizing coordinates of the nodes and cross-sections of the members, respectively (Figure 3(b)).

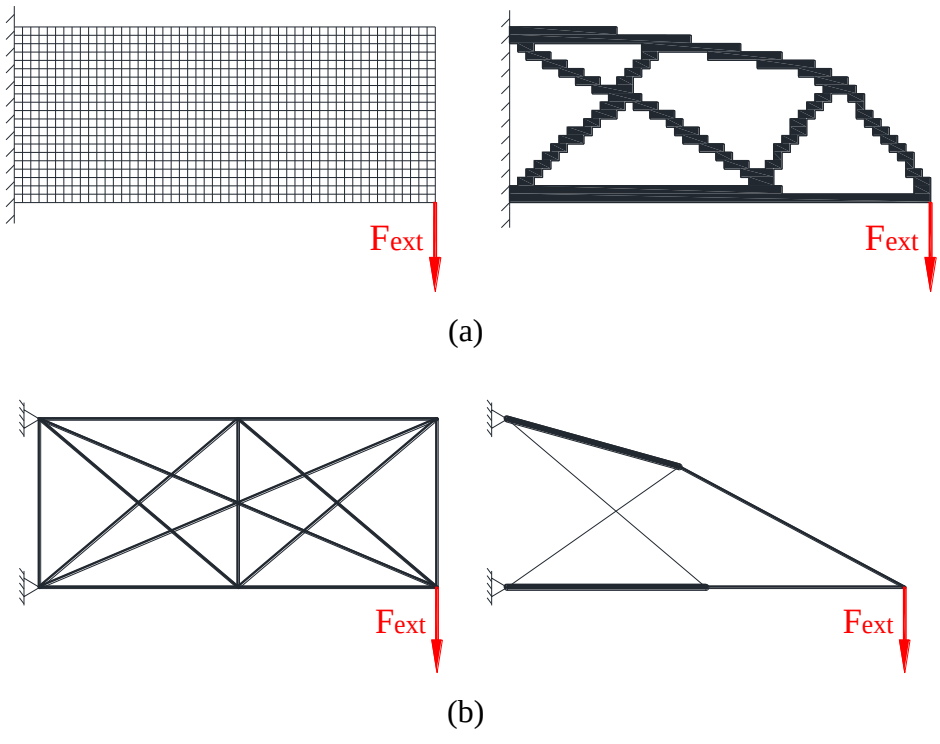


Figure 3. A typical optimized solution using a) continuum, b) discrete ground structure.

Continuum and discrete ground structure approaches and methods used for each class differ fundamentally:

- The optimized design of a continuum ground structure is usually considered as a *truss-like* structure, since it is difficult to distinguish discrete structural members with well-defined length and cross-sectional areas [6]. Such a design may provide useful insight on the optimum layout; however, large-scale practical structures are constructed by distinguishable members with fixed cross-sections, and thus the optimized solution from continuum topology optimization must somehow be interpreted as a system of discrete structural elements [6]. Continuum topology optimization can be utilized to get insight towards a good layout, which may be subsequently optimized by a discrete ground structure to determine the buildable optimal design [6]. Continuum topology optimization is an ideal candidate for parts which are built using additive manufacturing, in which there is no need for distinguishable members and nodes.
- In continuum topology optimization, the stress constraints assure that the maximum stress remains smaller than a fixed value. Standard design specifications codes often introduce more sophisticated failure criteria which may depend on the geometry of the member. For example, American Institute of Steel Constructions-Allowable Stress Design (AISC-ASD) [7] states that the slenderness ratio must be smaller than a fixed value. This ratio is independent of the axial force and depends only on the member length and the radius of gyration of the cross-section. Such requirements can be easily applied when the ground structure is discrete. Since cross-sections of members may vary in the optimized truss-like design, these types of constraints can hardly be applied in continuum topology optimization.

- The number of variables in a continuum grounds structure is much greater, varying from several thousands to even a few millions, which practically hinders application of metaheuristic methods in this category. Although some studies (such as [8, 9, 10]) applied metaheuristics to continuum topology optimization, they have not been widely accepted in the field [11]. Most well-known optimization methods applied to continuum ground structures are mathematical programming-based or gradient-based methods, such as homogenization method [12], solid isotropic microstructures with penalization (SIMP) [13] and evolutionary structural optimization (ESO) [14]. Gradient-based methods can result in better solutions with less computation effort for these problems, when compared to metaheuristics [15]. In contrast, the number of variables in a discrete ground structure is small, varying from tens to a few hundred for comparatively complicated test problems.
- Finite element analysis in the continuum ground structure is performed on 2D or 3D elements, and the number of elements varies from several thousands to a few millions, or even more. Therefore, computation time for a design evaluation can be huge. In contrast, when a discrete ground structure is employed, the number of elements is relatively small, equal to the number of members, and besides, the elements are 1D. Therefore, the finite element simulation is inexpensive. Consequently, the maximum number of evaluations can be much greater when a discrete ground structure is used.
- The required number of evaluations for gradient-based methods is small, about 50-200, almost independent of the number of variables. In contrast, metaheuristic methods require a great number of evaluations which grows polynomially with the number of variables.
- In aerospace engineering, the weight is a reasonable representative for the cost; however, in many infrastructure, the cost is affected by other factors such as the number of joints

[16], similarity of cross-sections [17], and constructability. Although most studies on truss optimization by metaheuristics have considered the weight as the objective, the flexibility of metaheuristics allows for consideration of more sophisticated factors that may affect the cost, as performed in some previous studies [16].

Based on this comparison, we may conclude that a discrete ground structure provides more practically interesting features than the continuum ground structure. Metaheuristics may excel in discrete ground structures, but they are not a reasonable choice for a continuum ground structure. Even for a discrete ground structure, a combination of a metaheuristic and an optimality criteria-based method was demonstrated to outperform purely metaheuristic approaches [18, 19]. For this dissertation, we limit our discussion to discrete ground structures.

It should also be noted that there are a few approaches that do not employ the concept of ground structure. The justification is dependence of the optimized solution on the ground structure. Shape annealing [20] eliminates this dependency and forms new topologies by employing some predefined rules, called shape grammar, on an existing design. Shape and size variables are also modified one-at-a-time using simulated annealing to direct the search process. The sampled topologies are configured by applying logical operators on triangular units. Although triangular units are commonly observed in truss structures, it may limit flexibility of the topology optimization. A revised version was applied to more intricate structures such as a transmission tower [21] by modifying the topology, shape and size. The logical rules were redefined and extended to handle design requirements.

1.4. Failure Criteria in Design

Each structure should satisfy some constraints such as stress, displacement and natural frequency constraints. There are also problem-specific buildability constraints which are usually

overlooked in academic research. Most available studies considered stress, buckling and displacement constraints. For sake of comparison with available methods, similar constraints are considered in this dissertation. It should also be noted that the formulation of the constraints depends on the specifications governed by the design code. Two cases are commonly followed in the literature. The first case, which we call *simplified specifications*, can be represented as follows:

$$u_{kl} \le u^{\text{all}}, k = 1, 2, \dots, DN_n, l = 1, 2, \dots, N_l,$$

$$-\sigma_{il} \le \sigma_{\text{T}}^{\text{all}}, |\sigma_{il}| \le \sigma_{\text{C}}^{\text{all}}, \qquad i = 1, 2, \dots, N_m, l = 1, 2, \dots, N_l,$$

$$-\sigma_{il} \le \frac{\alpha E A_i}{L_i^2}, i = 1, 2, \dots, N_m, l = 1, 2, \dots, N_l,$$

$$A_i \in \mathbb{A}, i = 1, 2, \dots, N_m,$$

N_m, N_n and N_l are the number of members, nodes and load cases respectively. D=2 for planar and D=3 for spatial trusses. σ_{il} is the stress in the i-th member and u_{kl} is the displacement of the k-th degree of the truss under the l-th load case, respectively. u^{all} is the allowable nodal displacement and $\sigma_{\text{C}}^{\text{all}}$ and $\sigma_{\text{T}}^{\text{all}}$ denote the allowable stress in compression and tension, respectively. A_i and L_i are the cross-sectional area and the length of the i-th member, respectively. ρ is the density of the truss material and $\mathbb{A}$ is the given set of available sections. This specification allows for different yield strengths in tension and compression as well as Euler buckling, assuming that the radius of gyration is proportional to the square root of the area of the cross-section, in which α encompasses the proportionality coefficient.

The practical truss structures show some deviations from the ideal pin-connected structures. In the design specifications employ in practice, such as AISC-ASD, more sophisticated constraints govern the problem. For AISC-ASD, these constraints are defined as follows:

$$u_{kl} \leq u^{\text{all}}, k = 1, 2, \dots, DN_n, l = 1, 2, \dots, N_l,$$

$$\begin{cases} \lambda_i \leq 300 & \text{if} \quad \sigma_i \geq 0 \\ \lambda_i \leq 200 & \text{if} \quad \sigma_i < 0' \end{cases} \quad \lambda_i = \frac{K_i L_i}{r_{\text{i}}} \tag{1}$$

where λ_i is the slenderness ratio, L_i is the length of the member, r_i is the smaller radius of gyration and K_i is the effective length factor. For truss members, K_i=1. The allowable tensile stress is calculated as follows:

$$\sigma_{il} \leq \sigma_T^{all} = min\{0.6F_y, 0.5F_u\} \tag{2}$$

where F_y and F_u stand for the yield and ultimate tensile strengths respectively. For members in compression, failure can be elastic or inelastic:

$$|\sigma_i| \leq \sigma_c^{\text{all}} = \begin{cases} F_y \left(1 - \frac{\lambda_i^2}{2C_c^2}\right) \Big/ \left(\frac{5}{3} + \frac{3\lambda_i}{8C_c} + \frac{\lambda_i^3}{8C_c^3}\right) & \text{if } \lambda_i < C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{if } \lambda_i \geq C_c \end{cases}, \; C_c = \sqrt{\frac{2\pi^2 E}{F_y}}, \tag{3}$$

where E is the modulus of elasticity, and C_c *is the critical slenderness ratio. For* $\lambda < C_c$, inelastic failure and for $\lambda \geq C_c$, elastic buckling may occur. The AISC-ASD specifications make the problem harder because:

- Unlike the simplified specifications, the stress constraints can be nonlinear.
- The slenderness constraint is independent of the magnitude of the member force. Furthermore, this constraint is not continuous with respect to member force.
- The allowable stress depends on the length and the radius of gyration of the cross-section.
- The uncertain assumption regarding the relation between r_{i} and A_{i} is avoided.

1.5. Bi-level Optimization

Bi-level (nested) optimization is a type of problem where evaluation of a solution requires solving another optimization problem. For example, design optimization of sensor placement on an artificial lateral line [22]. In the upper level, parameters of the lateral line including location of sensors are optimized such that accuracy of identification of vibrating objects is maximized. To measure accuracy of localization, many sample objects are considered and the artificial lateral line is used to identify them. The identification requires solving an inverse problem, which is solved by converting to an optimization problem.

In general, the bi-level optimization problem can be stated as follows [23]:

$$\underset{\boldsymbol{x}\in\boldsymbol{X},\boldsymbol{y}\in\boldsymbol{Y}}{\text{minimize}}\ F(\boldsymbol{x},\boldsymbol{y})$$

subject to $G_i(\boldsymbol{x},\boldsymbol{y})\leq 0,\ i=1,2,\ldots,N_G$

$H_j(\boldsymbol{x},\boldsymbol{y})=0,\ i=1,2,\ldots,N_H$

$$\mathbf{y}\in \underset{\boldsymbol{y}\in\boldsymbol{Y}}{\text{argmin}}\ f(\boldsymbol{x},\boldsymbol{y})$$

subject to $g_i(\boldsymbol{x},\boldsymbol{y})\leq 0,\ i=1,2,\ldots,N_g$

$h_i(\boldsymbol{x},\boldsymbol{y})=0,\ j=1,2,\ldots,N_h$

where:

- $F(\boldsymbol{x},\boldsymbol{y})$ is the objective function of the upper level.
- $\boldsymbol{x}$ is the vector of independent design variables.
- $\boldsymbol{y}$ is the dependent deign variable, the optimal value of which depends on $\boldsymbol{x}$.
- $G_i(\boldsymbol{x},\boldsymbol{y})$ and $H_j(\boldsymbol{x},\boldsymbol{y})$ are the constraints of the upper level problem.
- $f(\boldsymbol{x},\boldsymbol{y})$ is the objective function of the lower level, where $\boldsymbol{x}$ is a constant.
- $g_i(\boldsymbol{x},\boldsymbol{y})$ and $h_j(\boldsymbol{x},\boldsymbol{y})$ are the constraints of the lower level problem.

In bi-level optimization, the lower loop should be rendered to evaluate a design in the upper loop. This makes the process computationally expensive, unless the lower loop is an efficient algorithm.

Most research on evolutionary computation focused on either discrete or continuous search space [24], while practical optimization problems usually are mixed variable. Therefore, algorithm operators should be redefined if all parameters are optimized at the same time. Alternatively, a bi-level approach may be employed where discrete values are optimized in the upper level while the continuous values are optimized in the lower level, as followed in some studies [24]. The latter has two clear disadvantage: First, the process is very time-consuming and second, the possible correlation among discrete and continuous variable is overlooked [24].

1.6. Contribution of This Dissertation

This dissertation aims at overcoming some general drawbacks in truss optimization by introducing a novel algorithm specialized for optimization of truss structures. The contributions of this study can be summarized as follows:

- A bi-level TSS optimization method is developed for simultaneous topology, shape and size (TSS) optimization of truss structures. A stochastic search on all variables takes place in the upper level using a specialized evolution strategy (ES) while in the lower level, an OC-based approach, fully stressed design (FSD), is developed to find a quite optimally sized structure for the given shape and topology. A novel procedure to solve the resizing problem in the lower level is developed. The maximum variation in the lower level, is controlled to prevent divergence that may happen due to uncertainty of the FSD assumptions.

- A specialized FSD-based penalty term is introduced which estimates the required increase in structural weight such that all constraints are satisfied. This increased weight is considered as the penalized function value. The penalty coefficients are adapted since the FSD assumptions are not always valid.
- Novel TSS problems are proposed to improve common, yet simple, test problems. Such test problems reduce the gap between the complexity of test problems and real structures, which has been one the main barriers that prevent widespread application of structural optimization methods by practitioners. Moreover, it will be demonstrated that the gap between performances of different methods is significant on complicated test problems but usually marginal on simple ones. Therefore, such complicated test problems may provide a more reliable conclusion on superior of different methods.

The rest of dissertation is organized as follows: 0 provides an overview of Evolution Strategies (ESs), one the main stream of Evolutionary Algorithms (EAs). 0 reviews the truss optimization literature. The proposed method is introduced in 0 and subsequently improved in 0, 0 and 0CHAPTER 7. Finally, 0 summarizes the contributions and findings in this dissertation and highlight the main conclusions.

CHAPTER 2. EVOLUTION STRATEGIES

Optimization methods can be divided to deterministic and stochastic. Stochastic optimization methods are generally population-based methods which employ probability distributing functions for sampling new solutions. They can perform a more rigorous global search and therefore find better solution in multimodal problems. Moreover, deterministic methods generally rely on assumptions that can hardly be met in most real-world problems, which hinder their practicality.

Stochastic optimization methods are robust with respect to the problem assumptions. They have been developed for black-box problems, where no information on the system behavior is available except the output for a given input. The advantages of stochastic methods, however, come at cost of a considerable computation effort, which, can be supplied by recent development in computers and parallel computing.

EAs are one the earliest and most well-known stochastic optimization methods, also referred to as metaheuristics, which follow the principles of natural selection, recombination and mutation. Some other metaheuristic methods were introduced and developed subsequently such as particle swarm optimization [25], simulated annealing [26] and ant colony optimization [27]. More recently, there has been a growing interest in proposing novel stochastic optimization methods. Many of these methods have been published in the recent decade, too many to count; nevertheless, they can hardly provide any contribution to the field for the following reasons:

- Quite often, their novelty is only in the metaphor, not the algorithm [28].
- Proper explanation and justification of the steps in the proposed algorithms is overlooked. Similar previous methods are usually ignored.
- Experimental setup is usually biased. Selection of the test problems, performance measures, and the competing methods are not justified or the authors were unware of available standard experimental setups.

The proposed truss optimization method in this dissertation utilizes the principles of ESs, one of the main streams of EAs, which are versatile optimization tools particularly in continuous domain. In the canonical form, λ descendants are generated by recombination and mutation of μ parents. Selection is performed on the recently generated offspring (comma scheme), or on the union of descendants and ancestors (plus scheme), resulting in survival of the μ-best individuals for the next generation. In the following, a short description of each operator is provided.

2.1. Recombination Operator

Prior to mutation, recombination is optionally but preferably executed to generate a descendant by cooperation of ρ parents. Recombination can be employed for object (decision) and endogenous strategy parameters, e.g. the mutation strength. Recombination can be intermediary or discrete (dominant). In intermediary recombination, the recombinant design is the arithmetic average of the selected parents. In discrete recombination, the value of each variable of the recombinant design is selected randomly from the corresponding value of the selected parents. For the special case of $\mu=\rho$, the recombination is global. It is also reasonable to give a greater weight to a fitter parent in intermediary recombination (weighted recombination) and a higher probability of selection for a fitter parent in discrete recombination.

Intermediate recombination is unanimously recommended for strategy parameters, while the apt choice of recombination for object parameters has been discussed [29, 30]. Some researchers preferred discrete recombination [31], while an empirical study [30] demonstrated superiority of intermediate/weighted over discrete recombination.

Contemporary ESs for continuous parameter optimization generally employ global weighted recombination, in which all parents recombine with different weights to form a new solution [32]. Intermediary recombination has a critical advantage: Unlike discrete recombination, it has no bias

to search specific directions. Since the sampled solutions are subsequently generated by mutation of the recombinant design, it can be deduced that discrete recombination favors coordinate-wise search. For problems where minima are along the coordinate directions respect to each other, e.g. separable problems, such a bias is advantageous; however, for the correlated problems, it may drastically deteriorate the search efficiency. A comparison of results of different algorithms participated in CEC'2005 [33] and BBOB'2009 [34] workshops reveals that majority of optimization algorithms suffer from such an inherit bias. When the fitness landscape is linearly rotated, a drastic performance decline is observed for such algorithms, while it remains invariant for a few others. Such an invariant performance, under linear rotation of the fitness landscape in this case, is a feature of robustness [35]. It allows for generalization of empirical results to other landscapes that are generated by linear rotation of the tested function.

2.2. Selection Operator

Unlike GAs, ESs traditionally employ a deterministic selection scheme [32]. The early variants of ESs employed the plus selection scheme, which guarantees convergence to at least a local minimum [36]. The comma scheme was initially criticized since it is counterintuitive to lose the best solutions found in the optimization process [36]. Comma, however, has a significant advantage: It allows the algorithm to accept temporary deteriorations that might help to leave the attraction region of a local optimum and reach a better one [29]. Most well-known strategies for adaptation of the mutation step sizes are theoretically derived for the comma scheme, such as the traditional mutative self-adaptive strategy (also known as *self-adaptation*) or cumulative step size adaptation [36]. Almost all contemporary ESs employ the comma scheme [32]. Comma can deal with noisy [37] and dynamically changing environments [38]. The comma scheme gradually replaced the plus, although some experimental findings indicated the latter performs as well as, or

even better than comma in many practical cases [29]. For unconstrained continuous parameter optimization, the state-of-the-art ESs employ the comma scheme [32], although some empirical results have demonstrated in case of randomly generated multimodal functions, where distribution of local minima is chaotic, comma tends to leave deeper valleys for larger rims, and consequently, it falls behind EAs with elitism [39, 40]. Moreover, the plus scheme has been preferred for constrained, niching and multiobjective problems in most previous studies [29, 41, 42].

It seems that either of the schemes may have some advantages over the other, and thus it was preferred in different types of problems. Schwefel et al. [43] introduced the concept of lifespan for individuals as a generalized form of both extreme schemes, which limits a descendant's life to κ generations. Both extremes can be interpreted as special cases of this strategy: for κ=1, the selection scheme yields (μ,λ)-ES, while it turns into (μ+λ)-ES for κ=∞. This allows for arbitrary scaling of advantages and disadvantages of both extreme cases [43]. Nevertheless, only a little research has been conducted on this subject, including some theoretical investigation through the effects of the finite lifespan for artificial immune system [44, 45, 46, 47]. In a recent study [48], the trade-off between advantages and disadvantages of both selection schemes in coping with different features of multimodal problems were investigated by introduction of the aging rate. The aging rate gradually decays the fitness of individuals and was preferred over sudden elimination in the original definition of the finite lifespan. It was demonstrated that for non-symmetric landscapes, an intermediate selection scheme can outperform both extremes [48].

2.3. Mutation Operator

In contrast to GAs in which a small fraction of genes is mutated, all variables are mutated at the same time in ESs for continuous parameter optimization. In fact, this was one of the main principles since creation of ESs [36]. Variation of all variables at the same time helps the method

cope with possible correlation among variables efficiently. The mutation, however, is not a totally random operator, the strength of mutation is adjusted by the algorithm. Mutation is performed by adding a random perturbation to the recombinant design to generate a sample solution. In the simplest form, Isotropic mutation, Iso-density contours form concentric spheres (Figure 4(a)). In this case, merely a single mutation parameter should be adjusted, which is standard deviation of the normal distribution, the so-called *step size*. In a more general case, an independent step size can be allocated for each design parameter (N step sizes in total), in which Iso-density contours form axis-parallel ellipses (Figure 4(b)). Finally, correlation among design parameters can be considered which adds $N\times(N-1)/2$ rotation angles to the mutation parameters. In this case, Iso-density contours form arbitrary rotated ellipses (Figure 4(c)).

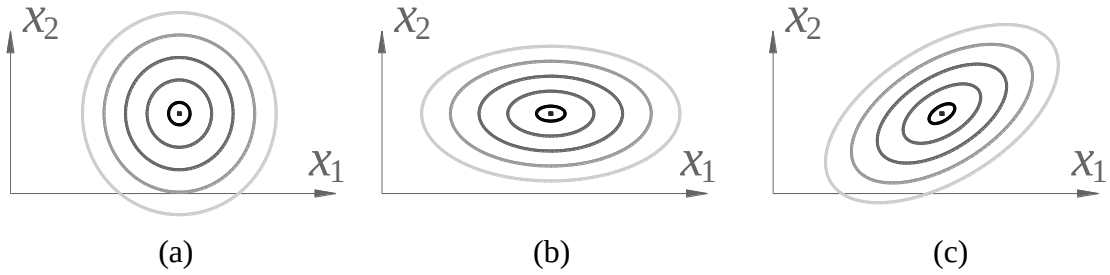


Figure 4. Iso-density contours for a) Isotropic Mutation with one free parameter b) D free parameters and c) $N\times(N+1)/2$ free parameters

2.4. Adjusting the Mutation Strength

Without adjusting the mutation strength, an ES becomes extremely inefficient [36]. Keeping the mutation strength at the (near-) optimal value is critically important in ESs, in contrast to genetic algorithms, in which the mutation rate is conventionally set to a small fixed value.

The one-fifth rule was the first approach proposed for adapting the mutation strength in (1+1)-ES [36], it increases the mutation strength if more than 1/5 of the new offspring are better than the parent. Later, the concept of mutative parameter control (self-adaption) was introduced as a robust

tool which can be used for the general case of ($\mu/\rho,+\lambda$)-ES. In self-adaption, the mutation strength is encoded as a strategy parameter which undergoes recombination and mutation as well. The mutation strength is mutated first, and the new value is used to generate a new solution. Fitness of the new offspring is correlated to the quality of its mutation strength, and thus, the mutation strength is updated similarly to the object parameters. The concept of self-adaptation can be generalized when one independent step size is allocated per decision parameter (Figure 4(b)), or even when N step sizes plus $N\times(N\text{-}1)/2$ rotation angles (Figure 4(c)) are considered to cope with high-conditioned problems efficiently.

One iteration of ES with self-adaption and one step size per object variable can be summarized as follows:

For j=1 to λ

$$\boldsymbol{\sigma}_j = \boldsymbol{\sigma}_{\text{mean}} \otimes \left(\exp\left(\tau_0 \mathcal{N}(0,1)\right) \exp\left(\tau \boldsymbol{\mathcal{N}}_N(0,1)\right)\right)$$

$$\boldsymbol{x}_j = \boldsymbol{X}_{\text{mean}} + \boldsymbol{\sigma}_j \otimes \boldsymbol{\mathcal{N}}_N(0,1)$$

Compute $f(\boldsymbol{x}_j)$

End

Sort individuals ($\boldsymbol{x}_j$'s) based on their function value.

$$\boldsymbol{x}_{\text{mean}} \leftarrow \sum_{j=1}^{\mu} w_j \boldsymbol{x}_j$$

$$\sigma_{\text{mean}_i} \leftarrow \prod_{j=1}^{\mu} \left(\left(\sigma_{ji}\right)^{w_j}\right), i = 1,2,\dots,D$$

In which $\mathcal{N}(0,1)$ is a random number and $\boldsymbol{\mathcal{N}}_N(0,1)$ is a vector of N independent random numbers sampled from standard normal distribution. w_j's are the weights for recombination and symbol $\otimes$ denotes elementwise multiplication. τ and τ_0 are fixed parameters specifying the learning rate for the step sizes. Default values are [36]:

$$\tau_0 = \frac{c}{\sqrt{2N}}, \tau = \frac{c}{\sqrt{2\sqrt{N}}}, \qquad c = 1 \tag{4}$$

Hansen and Ostermeier [35] highlighted some shortcomings of self-adaption, including the indirectness of the process: fitness of a solution is associated with the quality of its step size, while a bad step size may still result in a high-fitness solution and vice-versa. Furthermore, a smaller step size is more likely to results in a fitter individual. Consequently, the self-adaptation usually results in the mutation strength which is smaller than the optimal value, a challenge which is referred to as *favoritism* [36]. The concept of cumulative step-size adaption (CSA) was later proposed to overcome the shortcomings of the self-adaption. CSA analyzes the path of the population center in previous iterations to update the step size to avoid the favoritism. Hansen and Ostermeier [35] also proposed a derandomized approach which outperforms the self-adaption for adaption of the full covariance matrix.

2.5. State-of-the-Art Evolution Strategies

Covariance matrix adaptation evolution strategy (CMA-ES) is known as the state-of-the-art ES [32, 49]. CMA-ES employs global weighted recombination for both strategy and object parameters, comma for selection. It employs CSA to adapt the global step size and can adapts the full covariance matrix for mutation. It can handle badly-scaled functions and its performance remains invariant under rotation of the search space [35]. The parameter-free variants of this algorithm ranked first in CEC'2005 [50] and BBOB'2009 [51] for unconstrained continuous parameter optimization. Several variants of CMA-ES were developed in subsequent studies [32] to enhance efficiency or robustness of the method by adapting the population size or other strategy parameters [52, 53, 54], revising the heuristic for adaptation of the covariance matrix [55] or simplifying the algorithm and reducing the number of strategy parameters [56].

Despite the outstanding features of CMA-ES, the complexity of the adaptation process reduces its flexibility, when applied to constrained mixed-variable problems. In another study [56], a simpler variant of this method, called covariance matrix self-adaptation evolution strategy (CMSA-ES), was proposed which may compete with the original CMA-ES, at least when highly ill-conditioned problems are excluded. Because of its simplicity, it shows more flexibility for specialization for highly constrained mixed-variable problems. A recent niching method which employs CMSA-ES as the core search algorithm demonstrated promising results for multimodal optimization [57], and ranked the first in CEC'2016 and GECCO'2016 competitions on multimodal optimization. One iteration of CMSA-ES run as follows:

For j=1 to λ

$$\sigma_j = \sigma_{\text{mean}} \exp\big(\tau \mathcal{N}(0,1)\big),$$

$$\boldsymbol{z}_j = \sqrt{\mathbf{C}}\, \boldsymbol{\mathcal{N}}_N(0,1)$$

$$\boldsymbol{x}_j = \boldsymbol{x}_{\text{mean}} + \sigma_j \boldsymbol{z}_j$$

Compute $f(\boldsymbol{x}_j)$

End

Sort individuals ($\boldsymbol{x}_j$'s) based on their function value.

$$\boldsymbol{X}_{\text{mean}} \leftarrow \sum_{j=1}^{\mu} w_j \boldsymbol{x}_j$$

$$\sigma_{\text{mean}} \leftarrow \prod_{j=1}^{\mu} \big((\sigma_j)^{w_j}\big)$$

$$\mathbf{C} \leftarrow \left(1 - \frac{1}{\tau_c}\right)\mathbf{C} + \frac{1}{\tau_c}\sum_{j=1}^{\mu} w_j \boldsymbol{z}_j \boldsymbol{z}_j^T$$

In the pseudo code, **C** is the covariance matrix and $\sigma_j \boldsymbol{z}_j$ is the perturbation applied to the recombinant design, $\boldsymbol{X}_{\text{mean}}$, to generate the new solution, $\boldsymbol{x}_j$. w_j's specify the weights of parents in the updating process. In CMSA-ES, these weights are equal (w_j=1/μ), while in the original CMA-

ES, logarithmically decreasing weights [35] were preferred. τ and τ_c are fixed parameters specifying the learning rate for the global step size and the adaption interval for the covariance matrix, respectively. The recommended values are [56]:

$$\tau = \frac{1}{\sqrt{2N}},\ \tau_c = 1 + \frac{N(N+1)}{2\mu} \tag{5}$$

2.6. Handling Discrete Variables

For discrete variables, employing the normal distribution for mutation results in values that do not belong to the given discrete set. This justifies preference of discrete recombination for discrete variables in mixed-variable ESs [24], as well as those proposed for truss optimization [58, 59, 60, 61, 62]. One disadvantage of replacing normal distribution with other distributions is losing Isotropy, or rotation-invariance properties, the importance of which was discussed earlier. Besides, the derandomized approach to update the covariance matrix cannot be applied and the self-adaptation should be used for discrete variables [24]. Alternatively, it is possible to use a continuous distribution for discrete variables provided that the sampled values are subsequently rounded to a value in the given discrete set [36]. If the difference between the values of the discrete set is small, the properties of normal distribution are relatively preserved. Furthermore, the derandomized approach can be employed for discrete variables as well.

CHAPTER 3. RELATED STUDIES

This chapter reviews previous research on the main components of the proposed method in this dissertation, including evolution strategies (ESs), the concept of fully stressed design (FSD) and bi-level optimization. Relevant literature on truss optimization is explored, with emphasis on studies that performed simultaneous topology, shape and size optimization or employed an ES-based method for optimization.

3.1. Fully Stressed Design (FSD)

Methods based on optimality criteria are among the early approaches applied to truss optimization [63, 2], which assume that the optimal design should satisfy some *a priori* conditions [64, 65]. The concept of fully stressed design (FSD) is the most common approach in this group, which assumes that:

- in the optimally sized structure, all members reach the stress limit at least in one of the load cases [64].
- the axial force of members does not change when members are resized.

Accordingly, all members are iteratively resized to reach this goal. Change in topology is also possible by removing members with very small cross-sections; however, for our purpose, the topology is assumed to be fixed. The assumptions of FSD are not flawless. First, the global minimum is not necessarily a fully stressed design [65, 2]. Second, member forces change as soon as their cross-sectional areas are modified, except in determinate structures, in which FSD can find the global minimum in one iteration. When the number of redundant members is small, the error prompted by these assumptions is usually small, and iterative resizing, when the maximum variation during resizing is controlled, the so-called move limit [2], can reach a good design [64]. The required number of design evaluations is almost independent of the number of members [64], and the method usually reaches a good solution after a few iterations [64].

For member-based constraints (stress and buckling constraints), the smallest cross-sectional area that satisfies the constraints can be easily determined by multiplying the current area by the stress ratio, which is the ratio of stress in the member to the allowable limit. When more sophisticated design specifications such as AISC-ASD are imposed, there could be nonlinear stress constraints and slenderness ratio constraints. Even in these situation, the FSD assumptions allows for fast determination of the optimal cross-sections [63].

When there are displacement constraints, FSD can still be utilized by calculating the effect of each member on each displacement using the unit load method:

$$u_{kl}(\boldsymbol{A}) = \left| \sum_{i=1}^{m} \frac{c_{ikl}}{A_i} \right|, \quad c_{ikl} = \frac{f_{ik} F_{il} L_i}{E}, \tag{6}$$

where f_{ik} is the axial force in the i-th member when a unit load is applied to the k-th degree of freedom of the truss and F_{il} is the axial force in the i-th member under the l-th load case. According to Equation (6), each displacement constraint depends on many or even all members, therefore, solving the resizing problem, in general, is not easy. In a study [64], a two-step approach was employed such that in the first step, member sections are increased or decreased so that all stress constraints are satisfied and activated. In the second step, satisfaction of displacement constraints is pursued, while, no reduction in the cross-sectional areas is allowed.

For the case with only one displacement constraint, using optimality criteria leads to [66]:

$$CE_i = \frac{\partial W}{\partial A_i} \Big/ \frac{\partial u}{\partial A_i}, i = 1,2, \dots, N_m, \tag{7}$$

where 'W' denotes the structure weight. CE_i can be interpreted as *cost effectiveness* of the i-th member in reduction of the displacement constraint [66]. According to this equation, in the

optimally sized structure, all members have identical cost effectiveness. When sections are discrete, average cost effectiveness should be used:

$$CE_i \cong \frac{\Delta W}{\Delta A_i} \Big/ \frac{\Delta u}{\Delta A_i}, i = 1,2, \dots, N_m, \tag{8}$$

where ΔA_i is the difference between the current cross-sectional area and the next/previous area in $\mathbb{A}$. For the more general case, when there are multiple displacement constraints, the common approach is to merge all the displacement constraints into one constraint [66]. Schevenels et al. [66] proposed computation of average cost effectiveness of each member for all possible solutions around the current design, and selecting the one with the least average cost effectiveness. This process is repeated until a convergence condition is met. Although it was demonstrated that this strategy can reach a stable point, computation of average cost effectiveness for all possible designs around the current design is exponentially expensive, which limits the number of independent sections in the problem.

In some recent studies for size optimization [67, 68, 69] , the concept of FSD was partially utilized to guide the search. This resulted in a simple heuristic, however, since only a fraction of members are resized at each step and the maximum variation of a cross- sectional area was limited to a small amount, they require several hundreds or thousands of evaluations to reach a near optimally-sized structure. This number is still small compared to the methods that are purely meta-heuristic, but rather large compared to efficient resizing algorithms, which make the most of FSD assumptions. The latter can result in the optimally sized structure in one function evaluation for determinate truss structures and a good solution after a few (say, less than 20) evaluations in general [64], almost independent of the number of design variables [70].

3.2. Metaheuristics for Truss Optimization

Because of high efficiency, FSD used to be preferred over mathematical programming, when the computation resources were limited [64], except for highly indeterminate structures, where FSD risks divergence [64]. FSD does not take the objective function into account and thus, use of more sophisticated objective functions that consider other factors in the overall cost, is not directly applicable. When there are multiple displacement constraints, FSD leads to a resizing problem which is not easy to solve analytically.

Unlike optimality criteria, mathematical programming methods may solve general optimization problems [64]. With recent development in computation tools and parallel computing, the challenge of costly evaluations has been moderated to great extent. Advent of stochastic optimization methods, the so-called metaheuristics, such as EAs and swarm-based methods, introduced a new stream in optimization algorithms. There have been many studies on truss optimization with stochastic methods in the recent decade; however, most of them, even those published recently, can perform only size optimization. For example, size optimization by harmony search algorithm [71, 72], artificial bee colony [73], particle swarm optimization [74], genetic algorithm [75], ant colony optimization [76] evolution strategies [77] and many other apparently novel metaheuristics [78, 79, 80, 81, 82, 83, 67] as well as some hybridized methods [84]. Hasançebi *et al.* [62] compared performance of seven different stochastic optimization techniques for size optimization of truss structures and concluded that evolution strategies and simulated annealing are the most reliable ones.

A more sophisticated scheme optimizes shape and size [85, 18] or topology and size [86, 87]. Considering shape or topology in addition to size optimization can result in substantial saving in material [2]. Topology optimization is particularly a challenging task, since even a small variation

in topology may result in a significant change in member forces and besides, many kinematically unstable structures might be sampled during the search. A few studies compared performance of different metaheuristics for shape and size optimization with frequency constraints [88, 89].

The most sophisticated scheme, and potentially the most rewarding and challenging one [2], performs topology, shape and size (TSS) optimization at the same time. Surprisingly, studies on TSS optimization are comparatively scarce, possibly because of the complexity of the problem nature which demands sophisticated specialization of the standard metaheuristics. Several strategies to circumvent this complexity, in the case of TSS optimization, were proposed in the literature; however, they usually reduce the potential for better solutions [19]. Moreover, the size of the test problems employed to validate the algorithms is usually small or moderate at best [90, 91, 92, 93, 94, 95]. A few studies tried comparatively complicated problems as well [96, 97, 19], but a comprehensive comparison with other methods was not performed. In this chapter, we concentrate on studies that handle simultaneous topology, shape and size optimization and review their strategies to address the problem complexity.

3.2.1. Topology, Shape and Size Optimization: Challenges and Alternatives

Genetic algorithms (GAs) have been widely utilized in TSS optimization of truss structures [98, 90, 91, 99, 100, 101]. When using Binary-coded GAs, continuous variables are discretized [91, 102], for which the discretization step, which determines precision of the optimized results, should be tuned. Alternatively, some researchers applied mixed coding for shape and discrete size variables [103]. Deb and Gulati [90] proposed a real-valued GA in which the search range of member areas is assumed symmetric, for example, [$-A, A$], and members with cross-sectional areas less than a predefined threshold are considered passive. This strategy was also employed in come of the subsequent studies [104, 105], resulting in continuous treatment of all variables.

Another strategy to moderate problem complexity of TSS optimization is to use a two-stage approach. First, the structure topology is optimized while the cross-sectional areas of members and shape of the truss remain fixed. When an optimized topology is found, size and shape of the obtained topology are optimized. Such a strategy greatly alleviates the problem complexity as it reduces the number and type of design parameters at each stage. Luh and Lin exploited this strategy for TSS optimizing using ant colony [104] and particle swarm optimization [92]. Although for the investigated problems this two-stage strategy appeared beneficial, it cannot always provide the global optimum since TSS optimization is not a separable problem [90, 91]. The obtained results were outperformed by another method based on differential evolution [105], which considers the joint effects of topology, shape and size. Nonetheless, in the latter research, the drawbacks of continuous values for cross-sectional areas and specifying the threshold area remained unsolved.

A remarkably efficient strategy is to activate or deactivate a non-basic node or member with similar probabilities [96, 106]. This strategy leads to an inherent bias towards topologies with small number of nodes and members, since the number of acceptable topologies in which a non-basic node is active is much more than those where this node is passive.

Miguel et al. [93] developed a firefly-based multimodal optimization algorithm for TSS problems. The algorithm was demonstrated to find several good solutions in a single run, although the best identified solutions were slightly heavier than the best results reported in the literature. The number of evaluations was underestimated in their work, since they checked positive definiteness of the stiffness matrix to verify stability of the topology, and discarded and resampled unstable topologies. Since computation time for forming and checking the positive definiteness of the stiff matrix is considerable, this results in underestimation of the computation cost.

A recent study [94] proposed sequential cellular particle swarm optimization (SCPSO) for TSS optimization. The method was demonstrated to provide competing solutions for some TSS test problems, although only rather simple problems were tested.

Noilublao and Bureerat [97] applied multi-objective EAs on TSS optimization of a slender truss tower, where the second objective was introduced using either the natural frequencies, frequency response function (FRF), or force transmissibility (FT).

3.2.2. Evolution Strategies for Truss Optimization

Several ES-based methods for structural optimization have been developed in the last decades [98, 107, 61]. Considering that ESs were mainly developed to handle continuous unconstrained problems, their application for truss optimization necessitates some modification of the canonical form to handle discrete variables as well as constraints. These modifications can be categorized to five groups:

- *Distribution function for mutation***:** In most previous ES-based methods, normal distribution was replaced by some other distributions to handle discrete nature of size variables. As a matter of fact, recent ESs also employ discrete distribution for discrete variables [24], the drawbacks of which were discussed earlier. For example, Hasançebi [107] used a locally uniform distribution for discrete size variables. In ESs, larger mutations should be less probable, this property cannot be reflected by uniform distribution over the whole given set of discrete sections. Furthermore, the mutation should be scalable, which means the mutation strength should be controlled and besides, the distribution should be able to perform a perfect trade-off between exploration and exploitation [108] which is not the case for uniform distribution. Geometric distribution, as used by Hasançebi et al. [62] or Poisson distribution as used for size variables in other studies [59,

109] resembles normal distribution in the way that stronger mutations are less probable, yet, it cannot solve the problem of directional bias. These distributions rely on sequence of available sections rather than the difference between them. This make the performance highly dependent on the distribution of values in the available set of discrete sections.

- *Fraction of parameters that undergoes mutation***:** In contrast to GAs in which the mutation probability is low, all variables are simultaneously mutated in continuous optimization by ESs; however, the strength of mutation can be adapted to any arbitrary value. In discrete optimization, the fact that the difference between two adjacent values can be relatively large contradicts the requirement that the mutation strength could be as small as desired. As an alternative, mutation of a small fraction of design variables was pursued in most previous studies. For example, Lagaros et al. [109] proposed mutating about 20% of the size variables. In many other studies an extra parameter, *p*, defines the mutation probability of a size variable which can be self-adapted [58, 61, 106, 107, 62]. This is opposed to one of the earliest rule in ESs, mutating all variables at the same time [36] . The problem of directional bias remains as well.
- *Selection scheme:* Following the recommendations by ES pioneers, comma is preferred over plus (Bäck et al. 1997) for continuous parameter optimization. Contemporary evolution strategies [32] also employs comma. Some of the previous studies in truss optimization by ESs suggested this scheme [62], while some others preferred to preserve the best individual [107] or a more elite case, (μ+1)-ES [61], in which only the worst individual is removed.
- *Recombination scheme:* Previous research has suggested global weighted recombination as the most effective choice in continuous optimization by ESs (Kramer 2010), which is

also successfully employed in CMA-ES. For discrete variables, utilization of global weighted recombination usually leads to values that do not belong to the given set. This justifies preferring discrete recombination for size variables in previous studies [58, 59, 109, 61, 107, 62].

- *Adjusting strategy parameters:* The advantageous of derandomized adaption of the covariance matrix over self-adaption was demonstrated in [35]. A simpler version of CMA-ES that ignore pairwise correlation of variables (diagonal covariance matrix) may still provide competent results [110]. This simplification can be helpful for truss optimization where variables are of different nature. Most previous ES-based truss optimization methods, however, employ the more traditional concept of self-adaption for adjusting the strategy parameters [58, 59, 109, 61, 107, 62].

Such deviations from principles of contemporary ESs for continuous parameter optimization, are reasonable, since the truss optimization is neither continuous nor unconstrained; however, it comes at the price of losing some advantages. Direct implementation of the standard CMA-ES for truss optimization, as performed in a few studies [77, 111], may have its own disadvantages. Size parameters should be assumed to be continuous, a condition which can hardly be met in practice. The CMA-ES method needs a constraint handling strategy, which remains a constant challenge to its users.

CHAPTER 4. FULLY STRESSED DESIGN EVOLUTION STRATEGY

This chapter[2] elaborates the first version of the proposed method, called fully stressed design evolution strategy (FSD-ES). FSD-ES is a bi-level approach: In the upper level, the global search is performed by an ES-based method while in the lower level, FSD is utilized for fast optimization of size parameters of the given design. FSD-ES is thus a combination of a metaheuristic (upper level ES-based optimizer) and an optimality criteria-based method (FSD in the lower loop). For the following reasons, this combination is preferred over a purely metaheuristic approach:

- Although metaheuristics can provide good solutions for hard problems (multimodal, non-separable, ill-condition, discontinuous), the number of variables in the test problems on which contemporary metaheuristics have been evaluated is limited. For example, the highest dimension for benchmarking was 40 in BBOB'2013 [112], 50 in CEC'2013 [113] and 100 in CEC'2015 [114]. For TSS optimization, optimization of a moderate size structures can easily lead to 100 variables, unless members or nodes are grouped. Therefore, even the most sophisticated purely metaheuristic methods cannot efficiently cope with complicity of complicated TSS problems.
- Metaheuristics have usually been developed for either discrete or continuous variables [24] and mostly for unconstrained problems. The highly constrained mixed variable TSS optimization could be much harder for identical number of variables.
- Metaheuristics have mostly been developed to solve black-box optimization problems, where the only knowledge is the fitness of an arbitrary design. Truss optimization is not a black-box problem. A metaheuristic approach overlooking problem specific information would be easily outperformed by a heuristic approach that utilizes such information.

[2] This chapter uses some materials from our previously published work [18], available at: http://dx.doi.org/10.1016/j.compstruc.2013.04.013 . The publisher's policy allows reuse of the materials published by the authors in their dissertation.

The resizing step explores the space of size variables only, therefore, the evaluation budget for resizing of each solution should be limited; otherwise, the algorithm may converge to a poor shape or topology. In FSD-ES, each sampled solution is resized only one time. This setting parallels results in [64], which demonstrated that the maximal gain during the FSD-based resizing is reached during the first step.

In comparison with previous ES-based truss optimization, the upper level abides by the principles of contemporary evolution strategies:

- FSD-ES employs the normal distribution for all variables, in contrast to previous ES-based methods that employ other distributions for discrete variables. The drawbacks of the latter were discussed in Section 2.6.
- FSD-ES employs global weighted recombination for both strategy and object variables, in contrast to previous studies that perform discrete recombination for discrete variables. The motivation for this preference was discussed in section 2.1.
- FSD-ES employs comma as the selection operator, as performed by contemporary evolution strategies for continuous parameter optimization.
- FSD-ES mutates all variables at the same time, including discrete variables. The importance of this scheme was discussed in section 2.3.
- FSD-ES allocates an independent step size for each parameter, which are adjusted using the concept of self-adaptation. From this point of view, it resembles previous ES-based truss optimization methods.

This section presents the basic version of FSD-ES, proposed for shape and size optimization for a given topology. It elaborates the algorithm in details, evaluates it numerically and compares

the obtained results with the best available results in the literature. It also highlights importance of using the problem specific knowledge.

4.1. Algorithm Details

The steps of FSD-ES are explained in details in this section.

4.1.1. Notation

In FSD-ES, each candidate design, $\boldsymbol{\theta}$, is represented by two vectors:

- $\boldsymbol{X}$ is a vector of size DN_n, whose elements are continuous variables that determine nodal coordinates, where D=2 for planar and D=3 for spatial trusses.
- $\boldsymbol{A}$ is a vector of size N_m, whose elements are discrete variables that determine member cross-sectional areas.

Accordingly, $\boldsymbol{\theta}=\{\boldsymbol{X}, \boldsymbol{A}\}$ is a vector of size N_m+DN_n, with upper and lower limits of $\boldsymbol{\theta}^u$ and $\boldsymbol{\theta}^l$ respectively. The number of independent shape and size variables are denoted by N_{shape} and N_{size} respectively. Since grouping of members and coordinates is commonly used, the number of independent size and shape variables may be smaller than N_m and DN_n. An independent mutation step is allotted for each design variable, therefore $\boldsymbol{\sigma}=\{\boldsymbol{\sigma}_X, \boldsymbol{\sigma}_A\}$ is of size N_m+DN_n. Variables and step sizes corresponding dependent members/nodes do not undergo evolution.

4.1.2. Initial Solution

For the first iteration, recombinant design can be randomly selected within the bounds. As this design is not evaluated, it does not necessarily belong to the given discrete set. The recombinant point, denoted by the subscript 'mean', consists of $\boldsymbol{\theta}_{\text{mean}}=\{\boldsymbol{X}_{\text{mean}}, \boldsymbol{A}_{\text{mean}}\}$ and the corresponding vectors of step sizes $\boldsymbol{\sigma}_{\text{mean}}=\{\boldsymbol{\sigma}_{X\text{mean}}, \boldsymbol{\sigma}_{A\text{mean}}\}$. The center of the search range is selected as the the initial design. The values of the step sizes are set to one-third of the corresponding search range.

4.1.3. Mutating Shape Variables

Shape of the design is determined in this step. Step sizes are mutated first:

$$\boldsymbol{\sigma}_{X_j} = \boldsymbol{\sigma}_{X\text{mean}} \otimes \left(\exp\left(\tau_0 N_j\right) \exp\left(\tau \boldsymbol{\mathcal{N}}_{DN_n}(0,1)\right)\right) \tag{9}$$

in which the index j refers to the j-th individual, $\boldsymbol{\sigma}_{Xj}$ is the vector of step sizes for shape variables of the j-th individual, N_j is a random number sampled from the standard normal distribution. The sign $\otimes$ refers to element-wise multiplication. Coordinates of nodes are modified by mutation of the corresponding variables of $\boldsymbol{X}_{\text{mean}}$, which is sampled from the truncated normal distribution. The centre of mutation is $\boldsymbol{X}_{\text{mean}}$, the standard deviation is $\boldsymbol{\sigma}_{X\,j}$ and the truncated range is search range of shape variables. Using the truncated normal distribution, bounded variables are sampled in the range and bound constraints are automatically satisfied:

$$\boldsymbol{X}^{\text{up}} = \boldsymbol{F}\left(\mathbf{1}, \boldsymbol{X}_{\text{mean}}, \boldsymbol{\sigma}_{X_j}\right), \boldsymbol{X}^{\text{low}} = \boldsymbol{F}\left(\mathbf{0}, \boldsymbol{X}_{\text{mean}}, \boldsymbol{\sigma}_{X_j}\right),$$

$$\boldsymbol{X}_j = \boldsymbol{F}^{-1}\left(\boldsymbol{u}_{DN_n}(0,1) \otimes \left(\boldsymbol{X}^{\text{up}} - \boldsymbol{X}^{\text{low}}\right) + \boldsymbol{X}^{\text{low}}, \boldsymbol{X}_{\text{mean}}, \boldsymbol{\sigma}_{X_j}\right), \tag{10}$$

where $\boldsymbol{u}_{DN_n}(0,1)$ represents *DNn* independent random numbers between 0 and 1 sampled from uniform distribution**,** the index j refers to the j-th individual, N_j is a random number sampled from the standard normal distribution, τ_0 and τ are the learning rates. N_{VAR} is the total number of (independent) design variables, equal to the sum of the number of shape (N_{shape}) and size (N_{size}) variables.

4.1.4. Mutating Size Variables

Having determined the shape, the size of the structure is determined in this step. Step sizes are mutated first:

$$\boldsymbol{\sigma}_{A_j} = \boldsymbol{\sigma}_{A\text{mean}} \otimes \left(\exp(\tau_0 N_j)\exp\left(\tau \boldsymbol{\mathcal{N}}_{N_m}(0,1)\right)\right), \tag{11}$$

where $\boldsymbol{\sigma}_{Aj}$ is the vector of step sizes for size variables of j-th individual, and N_j is identical to its value in section 4.1.3. $\boldsymbol{A}_j$, the vector of cross sections of the candidate design, is sampled from the truncated normal distribution. The center of mutation is $\boldsymbol{A}_{\text{mean}}$, $\boldsymbol{\sigma}_{Aj}$ is the standard deviation and the search range of size variables is the truncated range.

$\boldsymbol{A}_j$ obtained from Equation (11) consists of continuous values. Since the cross-section should be a member of the given discrete set, the stochastic rounding technique is applied to size variables to create the actual design. Discrete variables are stochastically rounded to the upper or lower available value. For example, if A_{ji}=0.73 in^2, and the closest smaller and larger available cross-sectional areas are 0.70 and 0.75 in^2, A_{ji} is rounded to 0.70 in^2 with a probability of 0.4 and to 0.75 in^2 with a probability of 0.60. Unlike conventional rounding strategies which replace the continuous variable by the closest (scaled) discrete value [115], the stochastic rounding does not change expectation of A_{ji}, and therefore, the expected change in $\boldsymbol{A}_{\text{mean}}$ design will be zero if parents are selected randomly.

4.1.5. Evaluation

In this step, the generated design is evaluated and member stresses and nodal deflections are computed:

$$\boldsymbol{\sigma}_j = [\sigma_{ji}], \sigma_{ji} = \max_l \left\{ \left| \frac{\sigma_{jil}}{\sigma^{\text{all}}} \right| \right\}, \qquad i = 1,2,\dots,N_m$$

$$\boldsymbol{f}_j = [f_{ji}], f_{ji} = \max_l \left\{ 0, \frac{-L_i^2 \sigma_{jil}}{\alpha E A_i} \right\} \quad , \quad i = 1,2,\dots,N_m$$

$$\boldsymbol{s}_j = [s_{jk}], s_{jk} = \max_l \left\{ \frac{s_{jil}}{s^{all}} \right\}, \quad i = 1,2, \dots, N_m$$

$$\boldsymbol{u}_j = [u_{jk}], u_{jk} = \max_l \left\{ \frac{u_{jkl}}{u_k^{all}} \right\}, \quad k = 1,2, \dots, DN_n \tag{12}$$

Vectors $\boldsymbol{\sigma}_j$, $\boldsymbol{f}_j$, $\boldsymbol{s}_j$ and $\boldsymbol{u}_j$ store the ratios of calculated stress, compressive axial load, slenderness ratio and nodal displacement to their allowable limits in the most critical load case, respectively. For a feasible design, all elements of these vectors are equal to or less than one, otherwise some nodes or members have violated some constraints. If there are members that coupled to each other, the most critical constraint ratio is assigned to all of them.

For constraint violations, a penalty term is used knowing that the optimal design falls on or very close to the boundary of the feasible region, where several constraints are activated. A common approach to treat the constraints is using a very large penalty coefficient for infeasible solutions such that all feasible solutions are fitter than all infeasible solutions (Death penalty), as employed in many previous studies [90, 104, 92]. When the optimum is very close to the boundary of feasible region, a great proportion of samples that fall outside the feasible region are eliminated. The rest of samples form an asymmetric distribution, where under random selection, the recombinant point tends to move far from the global minimum. Therefore, the population cannot converge to the boundary unless the mutation strength is reduced, which risks premature convergence [116, 117, 48].

Utilization of FSD concept, an adaptive penalty term is introduced for truss optimization which enables the population to approximate the feasible region boundary from both sides [61]. The underlying idea is to estimate the required increase in the cross-sections of members that violate some constraints such that the design becomes feasible. The penalty term is calculated

based on this increase. Consequently, the proposed penalty term is specialized for the truss optimization problem, which uses problem specific knowledge. Based on FSD method, the following assumptions are utilized for any arbitrary $\alpha>0$:

- Nodal deflections are divided by α if all cross-sectional areas are multiplied by α.
- Member stress is divided by α if the area of that element is multiplied by α
- Member critical buckling load is divided by α^2, if the area of that element is multiplied by α.
- Slender ratio is divided by α if the corresponding member area is multiplied by α^2

The first assumption is valid for both determinate and indeterminate trusses, since proportional increase in the cross-sections do not change the axial force of members; however, the second and third assumptions are valid only for determinate trusses and an error, usually small but growing with the number of redundant members, originates for indeterminate trusses. The fourth one assumes that radius of gyration is proportional to square root of the cross-sectional area.

These assumptions, although not perfectly accurate, may provide a useful estimate for the required increase in the cross-sectional areas. For example, if $\sigma_{j2}=1.4$, A_{j2} should be multiplied by 1.4 so that the 2nd member of the j-th design satisfies the stress constraint. If so, the overall volume of the structure is increased by $0.4A_{j2}L_{j2}$. Although in AISC-ASD code the allowable compressive stress increases when the radius of gyration is increases, for simplicity, we overlook this effect, which may result in an overestimation of the required increase. When a displacement constraint is violated, it is assumed all cross-sectional areas should increase proportionally such that the displacement constraint is satisfied. The estimated increase in the cross-sectional area is $A_{ji}(p_{ji}-1)$ where:

$$p_{ji} = max\left\{1, \tilde{\sigma}_{ji}\ , \tilde{s}_{ji}^2, \sqrt{\tilde{f}_{ji}}, \max_i\{\tilde{u}_{ji}\}\right\} \tag{13}$$

This leads to an increase of $\sum_{i=1}^{N_m} L_{ji}A_{ji}(p_{ji}-1)$ in the volume of design j which depends not only on the constraint violation amount, but also on the current length and cross-section of that member. This means similar constraint violation for a larger member results in a larger penalty term. The penalty term is defined as follows:

$$f_j = W_j + \rho \sum_{i=1}^{N_m} c_{Pi}A_{ji}l_{ji}\left(p_{ji}^2 - 1\right), \tag{14}$$

Parameter c_{Pi}≥0.5 plays two roles in the algorithm. First, as a penalty coefficient in this equation for evaluation of the objective function. The second role will be explained in section 4.1.6. This parameter is introduced to control reliance on FSD, since assumptions of the FSD, as explained earlier, are not always valid and overlooking this uncertainty may result in performance degradation or even divergence. The adaptation scheme of c_{Pi} will be explained in section 4.1.8. The squared penalty (p_{ji}^2) was preferred since the uncertainty prompted by FSD assumptions increases when the estimated area is significantly larger than the current value. For c_{Pi}=0.5 and p_{ji}≈1, the penalty term is almost equal to $A_{ji}L_{ji}(p_{ji}-1)$.

4.1.6. Resizing

In this step, vectors $\tilde{\boldsymbol{\sigma}}_j$, $\tilde{\boldsymbol{f}}_j$ and $\tilde{\boldsymbol{u}}_j$ calculated in the previous step are utilized to generate an optimally sized design ($\boldsymbol{\theta}_{j+\lambda}$) from $\boldsymbol{\theta}_j$ by changing only the size variables. This step can be interpreted as the lower level of the optimization method, where the engineering knowledge on the system behavior, the concept of FSD in this case, is highly utilized. Accordingly, each member

cross-section is resized such that the stress or buckling constraint becomes activated, while member forces are assumed to be constant. This means the member should be loaded up to its maximum capacity; however, reduction of member areas takes place more conservatively since, as discussed earlier, not all members may reach their stress limit in the optimum design.

If the constraints are governed by simplified speculations (see section 1.4), yield strengths in tension and compression are given and buckling is assumed to follow Euler relation. The section is resized as follows:

$$\check{A}_{ji} = \begin{cases} A_{ji} \times \max\left\{\sigma_{ji}\,,\sqrt{f_{ji}}\right\} & \text{if } \max\left\{\sigma_{ji}\,,\sqrt{f_{ji}}\right\} \geq 1 \\ A_{ji} \times \left(1 + \left(\max\left\{\sigma_{ji}\,,\sqrt{f_{ji}}\right\} - 1\right) \times \exp(1 - 2c_{Pi})\right) & \text{otherwise} \end{cases} \tag{15}$$

where $\check{A}_{ji}$ is the resized section. In the above equation, the term $0<\exp(1-2c_{Pi})\leq 1$ moderates the shrinking rate when the member areas are going to reduce. If $\mathbb{A}$ consist of discrete values, $\check{A}_{ji}$ is rounded to the nearest upper value in $\mathbb{A}$.

When AISC-ASD design specification govern the problem, there is no simple relation between buckling load and member stress. Instead of lowering the shrinking rate, the member force is virtually magnified only if the corresponding stress constraint ratio is smaller than one:

$$\acute{f}_{ji} = \begin{cases} f_{ji} & \text{if } \sigma_{ji} \geq 1 \\ f_{ji} \times \left(\exp(1 - 2c_{Pi}) + \dfrac{1 - \exp(1 - 2c_{Pi})}{\sigma_{ji}}\right) & \text{otherwise} \end{cases} \tag{16}$$

The available section with minimum area that fulfils slenderness and stress constraints is found with respect to the corresponding virtual axial force. Although it is possible to update the current design $\boldsymbol{\theta}_j=\{\boldsymbol{X}_j, \boldsymbol{A}_j\}$with the resized one, our preliminary results suggested considering the

resized design as a completely new one. Therefore, $\boldsymbol{\theta}_{j+\lambda}=\{\boldsymbol{X}_j, \boldsymbol{A}_{j+\lambda}=\breve{\boldsymbol{A}}_j\}$ is considered as a new design which differs from $\boldsymbol{\theta}_j$ only in size variables.

Virtual increase of member force (AISC-ASD specifications) or lowering the area reduction rate (simplified specifications) is performed to shrink section area more conservatively than enlarging it. For example, $\sigma_{ji}=0.5$ means member stress is half of the limit; however, cross area would be halved only if $c_{Pi}=0.5$, which means the assumptions underlying the resizing step were effective. For larger values of c_{Pi}, the area reduction rate is lowered. In the third case for instance, $c_{Pi}=1$ and $g_{ji}=0.5$ lead to $f'_{ji}=1.63f_{ji}$ and the resized area that satisfies stress constraint will be $1.63\times0.5=0.82$ of the original area (Instead of 0.5). Therefore c_{Pi} decreases the shrinkage rate of the cross-sectional areas if the stress is smaller than the allowable limit, which is the second purpose of this parameter in FSD-ES.

Since vector $\boldsymbol{A}_{j+\lambda}$–$\boldsymbol{A}_{\text{mean}}$ is now different from $\boldsymbol{A}_j$–$\boldsymbol{A}_{\text{mean}}$, it is logical to adapt the corresponding step sizes to increase the chance of sampling around $\boldsymbol{A}_{j+\lambda}$ in the subsequent iterations:

$$\sigma_{A_{j+\lambda,i}} = \begin{cases} \left(1.9\sigma_{A_{ji}}\left|A\text{mean}_i - A_{j+\lambda,i}\right|\right)^{0.5} & \text{if } m_{ji}=1 \\ \left(\sigma_{A_{ji}}\right) & \text{if } m_{ji}=0 \end{cases} \tag{17}$$

The step sizes corresponding the new $A_{j+\lambda}$ are the geometric average of the step sizes of A_j and the difference between the resized values and $\boldsymbol{A}_{\text{mean}}$. The coefficient 1.9 in equation was obtained empirically. A smaller value expedites convergence but deteriorates global search. The fitness of the recently generated individuals is calculated according to Equation 14. This process is repeated until $(\lambda+\lambda)$ candidate solutions are generated.

4.1.7. Recombination

Global weighted recombination is performed to update the recombinant point. Having sorted the

offspring, the μ-best individuals (μ= [λ/2]) are selected to form the recombinant point. The new values of the recombinant design $\boldsymbol{\theta}_{\text{mean}}=\{\boldsymbol{X}_{\text{mean}}, \boldsymbol{A}_{\text{mean}}\}$ and the corresponding mutation strength, $\boldsymbol{\sigma}_{\text{mean}}=\{\boldsymbol{\sigma}_{X\text{mean}}, \boldsymbol{\sigma}_{A\text{mean}}\}$ are determined by performing weighted average (arithmetic for object variables and geometric for strategy variables) on the new parents. This is similar to the standard weighted global recombination in continuous ESs. The weight of each individual decrease logarithmically with their rank.

$$\boldsymbol{\theta}_{\text{mean}} \leftarrow \sum_{j=1}^{\mu} w_j \boldsymbol{\theta}_j,\ \boldsymbol{\sigma}_{\text{mean}} \leftarrow \prod_{j=1}^{\mu} \left(\boldsymbol{\sigma}_j\right)^{w_j},\ w_j = \frac{(\ln(\mu+1)-\ln(j))}{\sum_{k=1}^{\mu}(\ln(\mu+1)-\ln(k))} \tag{18}$$

where w_j's are the weights.

4.1.8. Update of Parameters

The penalty coefficients (c_{Pi}'s) are updated in this section. It can be assumed that the center of population is in the infeasible region if more that 50% of the solutions are violating the same constraint. If so, the penalty coefficient, c_{Pi}, is increased, otherwise decreased. Two vectors store the ratio of offspring and parents that violate a member constraint (stress or slenderness).

$$\psi_i^{(\mu)} \leftarrow (1-\tau)\psi_i^{(\mu)} + \tau \sum_{j=1}^{\mu} w_j \text{sgn}\left(p_{ji}-1\right),$$

$$\psi_i^{(\lambda)} \leftarrow (1-\tau)\psi_i^{(\lambda)} + \tau \sum_{j=1}^{\lambda} \frac{\text{sgn}\left(p_{ji}-1\right)}{\lambda}$$

$$c_{Pi} \leftarrow \max\left\{0.5, c_{Pi}\sqrt{0.5+\psi_i^{(\mu)}}, c_{Pi}\sqrt{0.5+\psi_i^{(\lambda)}}\right\} \tag{19}$$

where sgn is the sign function. $\psi_i^{(\mu)}$ and $\psi_i^{(\lambda)}$ represent the fraction of parents and offspring in which the *i*-th member has violated a constraint. If a displacement constraint is violated, all members are assumed responsible. Cumulative information with exponential decay of the past information is utilized to compute the ratio of infeasible parents or offspring.

4.1.9. Parameter Tuning

All the parameters of the algorithm are set to their default values. The default values of τ_0 and τ are $0.5(N_{\mathrm{VAR}})^{-0.5}$ and $0.5(N_{\mathrm{VAR}})^{-0.25}$, respectively, which can be obtained by setting $c=(0.5)^{0.5}$ in Equation (4). In metaheuristics, the recommended population size often grows sub-linearly with the number of design parameters. For example, it grows logarithmically in the default setting in CMA-ES [35]. Based on some parameter study, the near- optimal values of λ and the maximum number of iterations (*MaxIter*) were found to grow proportional to the square root of the number of variables. The recommended values are set as follows:

$$\lambda = 20 + \left\lfloor 5\sqrt{N_{VAR}} + 0.5 \right\rfloor, MaxIter = 200 + \left\lfloor 50\sqrt{N_{VAR}} + 0.5 \right\rfloor \quad (20)$$

MaxIter is set to a conservatively large value. FSD-ES does not utilize this parameter during the optimization. According to performance measures employed in this study (Section 4.2.2), the number of used evaluations to reach the target weight for the first time is considered as the required evaluation budget, which can be much smaller than the maximum allocated budget.

4.1.10. Flowchart of the Proposed Algorithm

Figure 5 illustrates flowchart of FSD-ES.

4.2. Numerical Evaluation

In this section, FSD-ES is numerically evaluated on a number of qualitatively distinct shape

and size optimization problems. Results of FSD-ES are compared to the best available results in the literature for each problem.

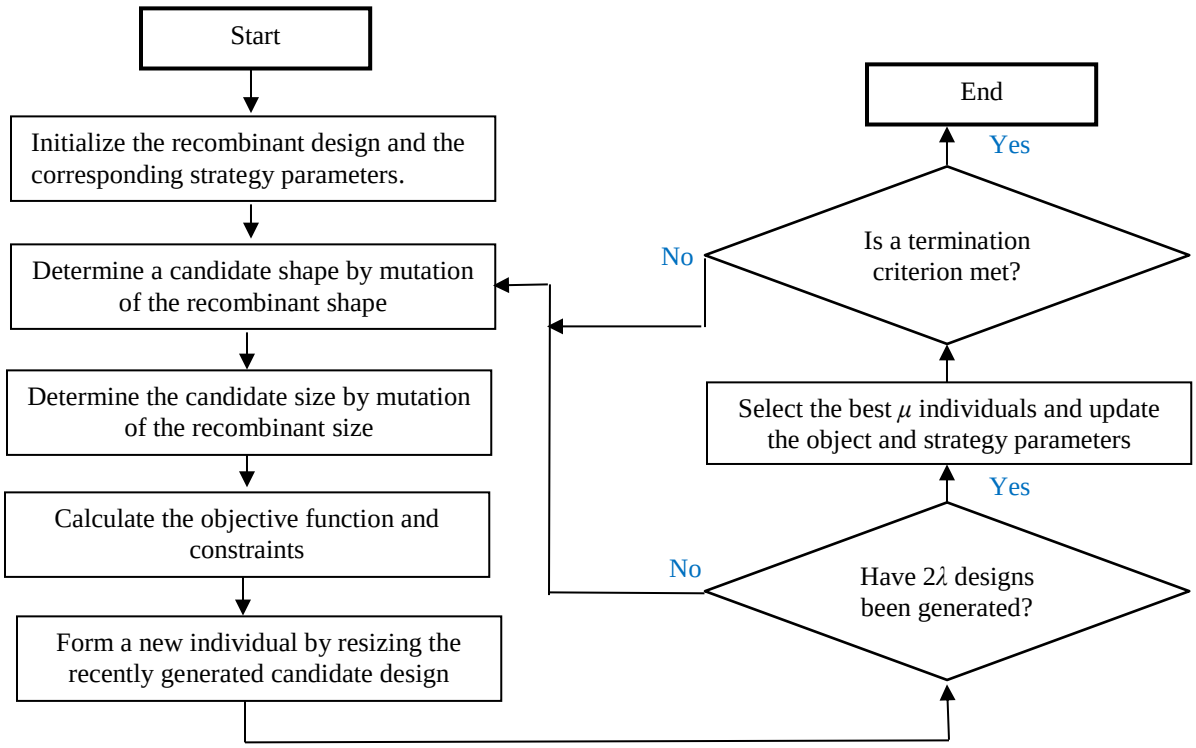


Figure 5. Flowchart of FSD-ES

4.2.1. Test Problems

The first test problem is simultaneous shape and size optimization of an 18-bar planar truss which has been investigated in several papers [118, 91, 119, 120]. Truss members are grouped to four size variables. Figure 6 illustrates the ground structure for this problem. Required data for simulation is presented in Table 1, which are adapted from [91]. For this problem, member areas are taken to be continuous, the structure is determinate and no displacement constraints exist. These factors not only simplify the problem but also make it less interesting, as the assumptions can hardly be met in practice.

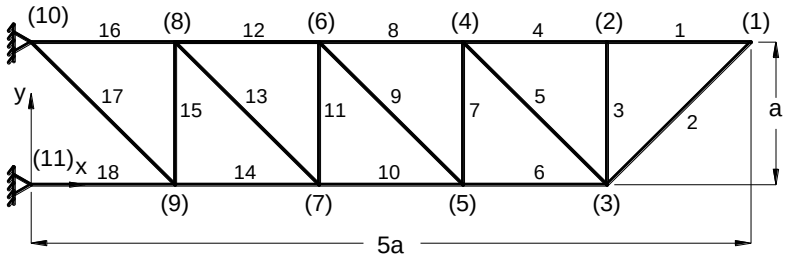


Figure 6. Ground structure for the 18-bar truss: a= 250 in.

Table 1. Simulation data for the 18-bar truss problem

Design Variables	**Shape (8)**	x_3 ; y_3 ; x_5 ; y_5 ; x_7 ; y_7 ; x_9 ; y_9
	Size (4)	$A_1=A_4=A_8=A_{12}=A_{16}$; $A_2=A_6=A_{10}=A_{14}=A_{18}$; $A_3=A_7=A_{11}=A_{15}$; $A_5=A_9=A_{13}=A_{17}$
Constraints	**Stress**	$\|\sigma_i\| \le 20$ ksi; i=1,2,…,18
	Displacement	None
	Buckling	$\|(\sigma_c)_i\| \le 4EA_i/L^2$, i = 1, . . . , 18
Search Range	**Shape Variables**	$-225'' \le y_3, y_5, y_7, y_9 \le 245''$; $775'' \le x_3 \le 1225''$; $525'' \le x_5 \le 975''$; $275'' \le x_7 \le 725''$; $25'' \le x_9 \le 475''$;
	Size Variables	3.5 in$^2 \le A_i \le$ 20 in.^{2}, i = 1, 2, . . . , 18 (continuous values)
Loading		Nodes F_x KN (kips) F_y KN (kips) F_z KN (kips) 1, 2, 4, 6 ,8 0 −89.075 (−20.0) 0
Mechanical Properties		Modulus of elasticity: E=68.95 GPa (1.0×10^4 ksi) Density of the material: ρ=0.0272 N/cm^3 (0.1 lb/in.^{3})

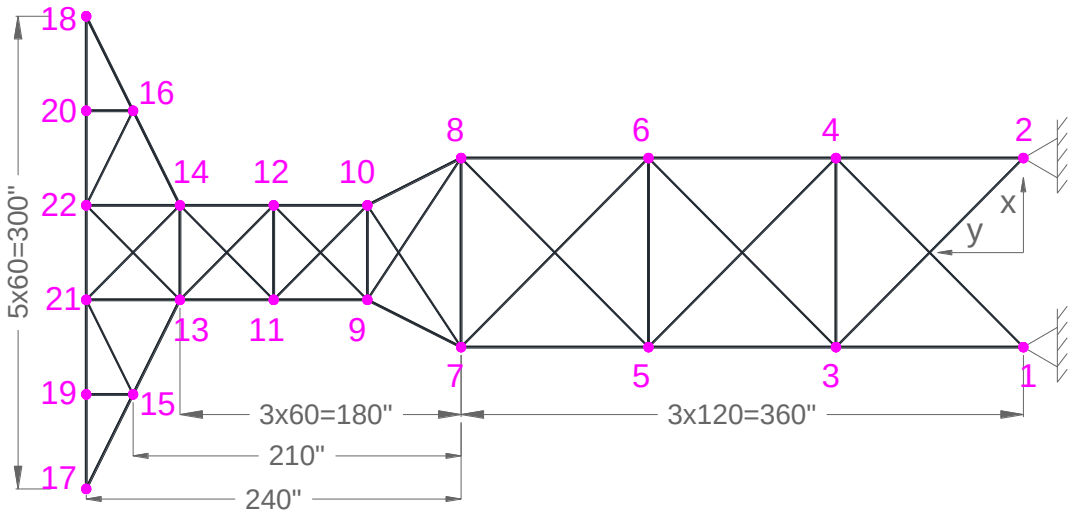


Figure 7. Ground structure of the 47-bar power-line problem. The illustration is rotated for better use of space.

The second problem is size and shape optimization of a 47-bar power line truss adapted from [121]. Problem specifications are presented in Table 2 and the ground structure is depicted in Figure 7. The structure is supposed to be able to carry three load cases. External load is not symmetric, but the structure design including nodal coordinates and member cross-sections are symmetric. Coordinates of nodes 15, 16, 17 and 18 are fixed. The ranges of coordinate variables were not quoted in the referenced paper and hence, a logical range was selected for this study.

Table 2. Data for simulation of the 47-bar truss problem

Design Variables	**Shape (17)**	$-x_1=x_2$; $-x_3=x_4$, $y_3=y_4$; $-x_5=x_6$, $y_5=y_6$; $-x_7=x_8$; $y_7=y_8$; $-x_9=x_{10}$; $y_9=y_{10}$; $-x_{11}=x_{12}$; $y_{11}=y_{12}$; $-x_{13}=x_{14}$; $y_{13}=y_{14}$; $-x_{19}=x_{20}$; $y_{19}=y_{20}$; $-x_{21}=x_{22}$; $y_{21}=y_{22}$		
	Size (27)	$A_i=A_{i-1}$, i=2, 4, 6, …, 20 ; A_{41} , A_{42} , A_{43} , A_{44} , A_{45} , A_{46} , A_{47}		
Constraints	**Stress**	$(\sigma_c)_i \leq$15 ksi; $(\sigma_t)_i \leq$ 20 ksi, i=1, 2,…, 47		
	Displacement	None		
	Buckling	$\lvert(\sigma_c)_i\rvert \leq \alpha E a_i / l_i^2$, i=1, 2, …, 47 , α=3.96		
Search Range	**Shape Variables**	$0 \leq x_2, x_4, x_6, x_8 \leq 150''$; $0 \leq x_{10}, x_{12}, x_{14} \leq 75''$; $0 \leq x_{22} \leq 75''$; $0 \leq x_{20} \leq 150''$; $0 \leq y_4 \leq 240''$; $120'' \leq y_6 \leq 360''$; $240'' \leq y_8 \leq 420''$; $360'' \leq y_{10} \leq 480''$; $420'' \leq y_{12} \leq 540''$; $480'' \leq y_{14} \leq 600''$; $540'' \leq y_{20}$, $y_{22} \leq 660''$		
	Size Variables	$A_i \in \mathbb{A}$, i = 1, . . . , 47 $\mathbb{A}$ ={0.1, 0.2, 0.3, …, 4.9, 5} (in.^{2})		
Loading		Nodes	F_x	F_y
	Case I	17,18	6 kips	-14.0 kips
	Case II	17	6 kips	-14.0 kips
	Case III	18	6 kips	-14.0 kips
Mechanical Properties		Modulus of elasticity: E=3.0×10^4 ksi Density of the material: ρ=0.3 lb/in.^{3}		

The third problem is simultaneous shape and size optimization of the 77-bar truss bridge adapted from [19]. The bridge consists of 20 panels and spans 20×25=500 ft. A force of magnitude 60 kips is applied downward on each node of the lower cord, excluding the supports Among the five candidate topologies studied in the referenced study, the Parker model emerged as the optimum topology and besides, other topologies have no or only a few shape variables. Accordingly, this case is investigated here. The bridge deck is subject to a uniform load downward. A set of 83 discrete sections between W10×12 and W14×730 in W-shape profile list [24] are used to size the truss members, which are governed by the stress and stability provisions of AISC-ASD

[24]. Furthermore, displacements of the panel points in any direction are restricted to 10.0″ and $50'' \leq y_i \leq 1000''$ (i=3, 5, 7, …,21). Symmetry is exploited to group design variables to 39 size and 10 shape variables. In the section list, there are five pair of sections that have identical area but different radius of gyration. For each pair, we remove the one with the smaller radius of gyration, since it cannot provide any advantage over the other one.

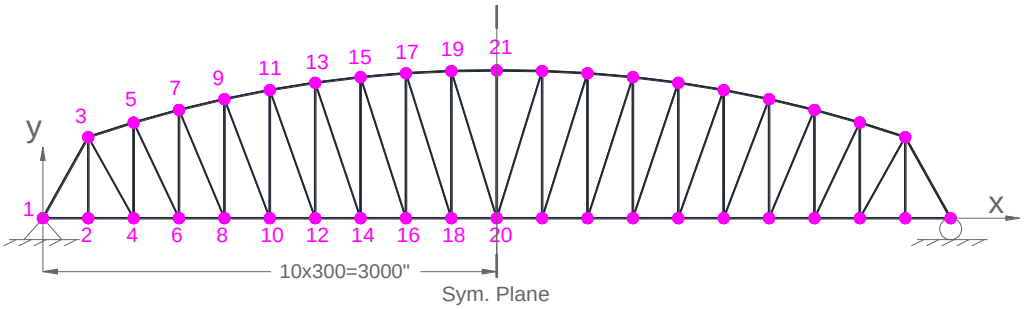


Figure 8. Ground structure for the 77-bar truss bridge problem

4.2.2. Performance Measures

Because of stochastic nature of metaheuristics, comparison should be performed over results of several independent runs. Most studies on metaheuristics employ fixed-cost scenario, in which final solutions after a predefined evaluation budget are compared, usually by comparing the mean and the standard deviation of the final solutions. Alternatively, one may employ a fixed target scenario, where the required computation efforts by different methods to reach a target function value are compared.

The performance measure employed in this study is based on the expected running time (ERT) which measures the expected number of evaluations that an algorithm requires to reach the global minimum for the first time [122]:

$$ERT = \frac{SR \times \overline{FE}_S + (1 - SR) \times \overline{FE}_{US}}{SR}, \quad (21)$$

where *SR* is the success rate, which is the fraction of independent runs in which the algorithm could reach the global minimum. FE_S is the average number of function evaluations of successful runs to reach the global minimum, and FE_{US} represents the average number of function evaluations of unsuccessful runs. *ERT* facilitates comparison of optimization methods since it considers the overall effects of reliability and convergence speed. In comparison with measures based on fixed cost scenario:

- ERT provides quantitative comparison [123], e.g. algorithm A is two times efficient that algorithm B if its *ERT* is half of the *ERT* of algorithm B [123]. Such quantitative comparison cannot be performed in fixed cost scenario, where the function values after a fixed number of evaluations are compared.
- *Mean* is not a robust statistic with respect to outliers. If a few independent runs conclude to poor solutions or diverge, they heavily deteriorate the mean value. Such a huge impact of a few unsuccessful runs could mislead the comparison, since in practice, the outcome of the best run among *k* independent runs is selected.
- It is unclear whether a higher standard deviation in the results is an advantage or disadvantage. Quite often, a high standard deviation is interpreted as erratic performance of an algorithm on a problem, and thus it is considered as a disadvantage. Considering that in practice, the outcome of the best run among *k* independent runs is selected, a greater standard deviation, opposed to intuition, might be desirable, because for identical *mean values*, a higher standard deviation means a lower *min value*.

To utilize Equation (21), the algorithm should stop when it reaches a certain tolerance of the

objective function or when the convergence criterion is met. However, in truss optimization, as with most practical problems, the global optimum is not known and besides, the performance measure should discriminate between converging to a high/low fitness minimum. Accordingly, the optimization algorithm is run for a long time and the required number of evaluations to reach different structural weights is recorded. Therefore, the measured *ERT* depends on the target weight (W_{target}). It can be reasonably assumed $FE_{\text{US}} \sim FE_{\text{S}}$ and thus Equation (21) concludes to:

$$ERT(W_{\text{target}}) = \frac{\overline{FE}_S(W_{\text{target}})}{SR(W_{\text{target}})} \tag{22}$$

In this study, the above equation is used to calculate *ERT* from FE_S and *SR* at any arbitrary target function value. The *ERT* plots in this case can discriminate and compare both the short and long term performance of algorithms. Such discrimination is of great practical interest as the budget of function evaluations determines whether short or long-term success is desired.

Unfortunately, statistical performance measures for reliable comparison of the methods is not provided in many referenced studies. Many studies only reported the best solution found and the parameter setting of their algorithms. The *best weight* is not a robust statistic and repeating the experiments is likely to result in other best values. Therefore, reliable conclusions cannot be made unless the difference between the best solutions is considerable. Furthermore, it is not clear in what fraction of runs their methods could reach the best reported solutions. In many other studies, although the parameter setting including the population size and the maximum number of iterations were reported, the used number of evaluations to reach the reported solutions, which may be smaller than the allocated evaluation budget, is not reported. We further assume that the reported number of evaluations is equivalent to FE_{S}.

4.2.3. Results and Discussion

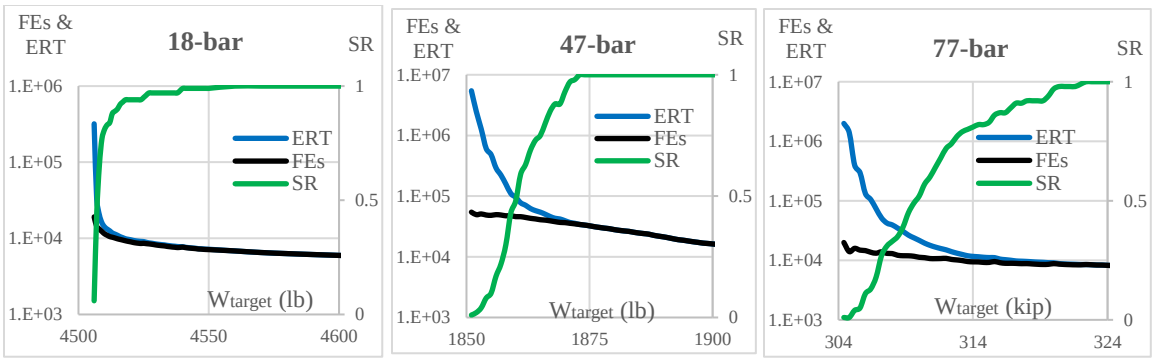
A comparison of results of FSD-ES on the employed test suite with the best available results is performed in the section. Each test problem is solved 100 times independently and the calculated *ERT* accompanied by *SR* and FE_S is plotted versus the target weight. These plots are provided in Figure 9. Only feasible designs that satisfy all constraints are considered in calculating FE_S and *SR*. A summary of the best solutions found by FSD-ES and other available methods is provided in

Table 3. The best feasible solution found in this study is also depicted Figure 10 and the corresponding data are tabulated in Table 4. Based on the results, the following conclusions can be made:

- Reaching lighter structures logically needs more evaluations; however, Figure 9 demonstrates that *ERT* grows much faster when lighter structures are desired. This is since not all independent runs could reach some target weights. Consequently, the gap between the FE_S and *ERT* lines increases when the target weight decreases. When only a few independent runs can reach a target weight, the computed value of *ERT* is prone to unreliability due to the stochastic nature of the runs.
- For the 18-bar problem, plots of *FEs*, *ERT* and *SR* are illustrated in Figure 9(a). The plots show that success rate gradually decreases for target weights smaller than 4520 lb, which results in a sharp increase in *ERT*. For this problem, the best solution of Kaveh and Talatahari [118] weighs 4507.15 lb, reached after 4000 function evaluations and was demonstrated to outperform some other optimizers. Lamberti [119] employed a gradient-based method called CMLPSA to this problem. It could reach a slightly better solution (W=2043.86 Kg = 4505.95 lb) after 550 evaluations, however, the success rate was not reported. FSD-ES could reach the target weight of 4506.0 lb, after 19,129 evaluations with SR=0.06. It is quite identical to the best solution available in the literature, although the required function evaluation is several

times the SA-based method. This problem is determinate, subjected to one load case with no displacement constraint. The fitness landscape is likely to be unimodal in the space of size variables, which may justify the success of the CMLPSA.

- For the 47-bar truss problem, FSD-ES reached W_{target}=1860.0 after 45,893 evaluations with SR=0.48 (Figure 9(b)). The best solution weighs 1850.1 lb, although only one run could reach this target weight.
- Table 3 shows that the closest competent is SCPSO [94], which could reach the best weight of 1864.1 lb after 25,000 evaluations. FSD-ES could find a relatively lighter structure, although the used number of evaluations is somewhat greater. It should be noted a smaller population size improves the short-term performance, and thus, the required number of evaluations to reach an arbitrary target weight decreases, unless a very light structure is desired. There are a few other studies that tried this problem but not included in the comparison, because they used a different section list [124], considered one load case only [125], or their final result highly violates a constraint [126].
- Figure 9(c) illustrates the calculated performance measure for the 77-bar bridge problem. Hasançebi [107] employed a (25,150)-ES with a maximum of 1000 generations (150,000 evaluations) for this problem. His best reported solution weighs 317.13 kip, about 4.2% heavier than the best solution found by FSD-ES, which weighs 304.37 kip. 52% of independent runs could reach the structural weight of 310.00 kips, on average after 11,253 evaluations. For this problem, FSD-ES is not only several times faster, but also able to find a comparatively lighter structure.



(a) (b) (c)

Figure 9. *ERT*, *SR* and *FE*S to reach arbitrary structural weights for the employed test problems: a) 18-bar truss, b) 47-bar truss and c) 77-bar truss bridge problems

Table 3. Summary of the best results available in the literature for each problem. For FSD-ES, *FEs* for two selected target weights are reported

Problem	Method	Year	W_{min}	W_{min}/W^*_{min}	FEs	FEs/ FEs*
18-bar	**CMLPSA** [119]	2008	**4505.95** lb	1.000	**550**	1.000
	ECCS [118]	2011	4507.13 lb	1.000	4000	7.273
	SCPSO [94]	2013	4512.37 lb	1.001	4500	8.182
	FSD-ES [This study]	2013	4506.00 lb	1.000	19,139	34.798
			4508.00 lb	1.000	13,248	24.087
47-bar	**GA** [102]	2001	1925.8 lb	1.040	100,000	4.000
	SA [121]	2002	1871.1 lb	1.012	26,400	1.056
	SCPSO [94]	2013	1864.1 lb	1.007	**25,000**	1.000
	FSD-ES [This study]	2013	**1851.0** lb	1.000	54,483	2.179
			1860.0 lb	1.005	45,893	1.836
77-bar	**ES** [107]	2008	317.13 kip	1.042	150,000	13.330
	FSD-ES [This study]	2013	**304.40** kip	1.000	19,911	1.769
		2013	310.00 kip	1.018	**11,253**	1.000

Table 4. Data for the best feasible solution found for each problem

	18-bar		47-bar				77-bar			
	X_3	911.7713	X_2	98.3336	A_9	2.6	Y_3	224.2682	A_{16}	75.6
	Y_3	185.7973	X_4	83.4020	A_{10}	0.7	Y_5	335.9095	A_{17}	75.6
	X_5	643.8633	Y_4	134.4980	A_{11}	2.5	Y_7	450.8260	A_{18}	75.6
	Y_5	147.5345	X_6	65.6210	A_{12}	0.8	Y_9	544.3938	A_{19}	75.6
	X_7	414.1109	Y_6	259.3571	A_{13}	0.7	Y_{11}	603.2893	A_{20}	3.54
	Y_7	98.4023	X_8	56.7434	A_{14}	1.8	Y_{13}	658.2400	A_{21}	14.4
	X_9	202.3849	Y_8	347.6946	A_{15}	1	Y_{15}	703.2170	A_{22}	7.65
	Y_9	30.5643	X_{10}	50.5018	A_{16}	1.1	Y_{17}	738.1255	A_{23}	9.71
	A_1	12.4778	Y_{10}	413.6941	A_{17}	0.3	Y_{19}	761.0233	A_{24}	19.1
	A_2	17.8260	X_{12}	45.6190	A_{18}	1	Y_{21}	762.0000	A_{25}	14.4
	A_3	5.2707	Y_{12}	472.5069	A_{19}	1.3	A_1	35.3	A_{26}	14.4
	A_5	3.7202	X_{14}	43.0924	A_{20}	0.9	A_2	35.3	A_{27}	14.4
			Y_{14}	513.0640	A_{21}	0.8	A_3	44.7	A_{28}	14.4
			X_{20}	89.9646	A_{22}	1.3	A_4	50	A_{29}	14.4
			Y_{20}	627.8005	A_{23}	0.1	A_5	50	A_{30}	68.5
			X_{22}	5.6310	A_{24}	0.2	A_6	51.8	A_{31}	11.8
			Y_{22}	589.3159	A_{25}	0.1	A_7	55.8	A_{32}	7.65
			A_1	3.3	A_{26}	0.1	A_8	55.8	A_{33}	9.71
			A_2	1.1	A_{27}	0.1	A_9	55.8	A_{34}	14.4
			A_3	3.1			A_{10}	55.8	A_{35}	14.4
			A_4	1.1			A_{11}	68.5	A_{36}	14.4
			A_5	2.9			A_{12}	75.6	A_{37}	15.8
			A_6	0.9			A_{13}	75.6	A_{38}	19.1
			A_7	2.6			A_{14}	75.6	A_{39}	19.1
			A_8	0.8			A_{15}	75.6		
Weight	**4505.92 lb**		**1850.11 lb**				**304.3747 kip**			
$\max_{i,l}\{\sigma_{il}\}$	**1.0000**		**1.0000**				**1.0000**			
$\max_{i,l}\{f_{il}\}$	**1.0000**		**0.9986**				–			
$\max_{k,l}\{u_{kl}\}$	–		–				**1.0000**			
$\max_{i}\{s_i\}$	–		–				**1.0000**			

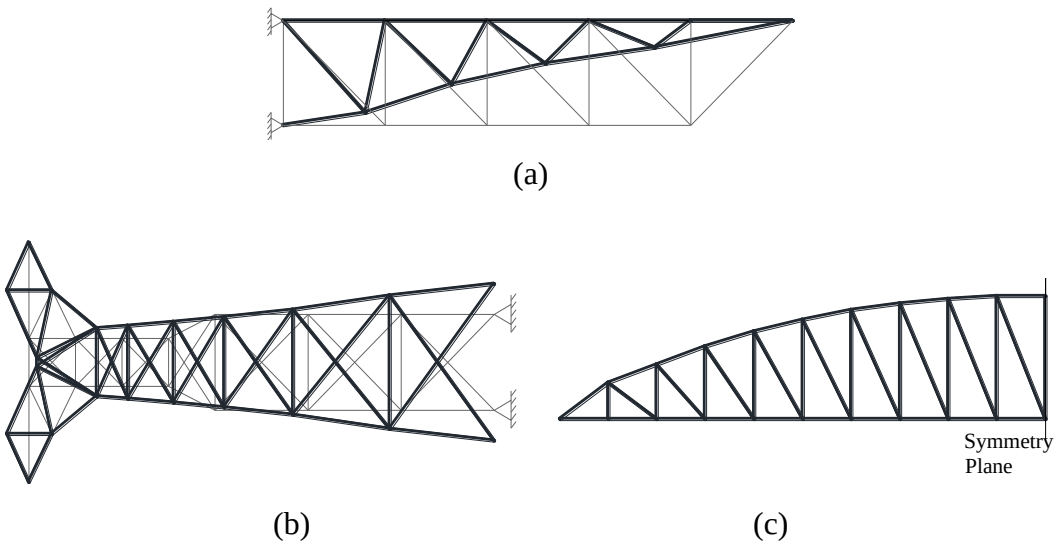


Figure 10. The best feasible solution found for each problem: a) 18-bar, b) 47-bar and c) 77-bar truss problems

4.2.4. Importance of Problem Specific Knowledge

As it was discussed earlier, the concept of FSD utilizes problem specific knowledge for fast size optimization of a solution. FSD-ES utilized the concept of FSD for defining a specialized penalty term and fast resizing of a given truss. This feature discriminates FSD-ES from the large number of purely metaheuristic truss optimization methods in the literature. To explore importance of this feature, four variants of FSD-ES are tested and compared in this section:

- FSD-ES: The standard version of the proposed method which performs resizing and employs the specialized penalty term.
- FSD-ES-{resize}: The proposed method without the resizing step (lower level). The whole 2λ solutions are sampled using the ES-based upper level.
- FSD-ES-{penalty}: The proposed method without the specialized penalty. A conventional penalty term is employed in this case.

- FSD-ES-{resize, penalty}: Both resizing step and the specialized penalty term are suppressed. All solutions are sampled using the ES-based upper level and evaluated using a conventional penalty term.

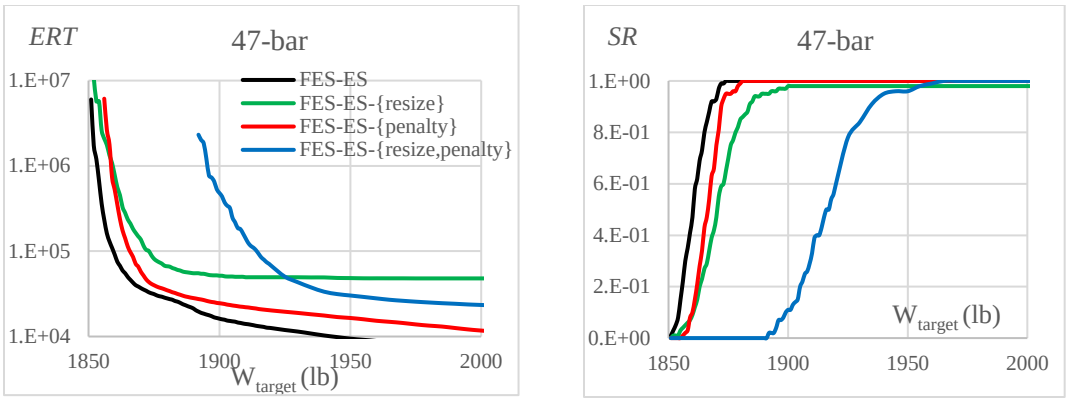
These variants are tested on the 47-bar problem. This structure is indeterminate which may challenge assumptions of FSD. The alternative penalty term for the third and fourth variants is defined as sum of squared constraint violations:

$$f_j = W_j \times \left(1 + \sum_{i=1}^{N_m} \langle \sigma_{ji}^2 - 1 \rangle + \sum_{i=1}^{\mathrm{N}_m} \langle f_{ji}^2 - 1 \rangle + \sum_{k=1}^{DN_n} \langle u_{jk}^2 - 1 \rangle \right), \tag{23}$$

in which:

$$\langle x \rangle = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Each variant is employed to solve this problem 100 times independently. To consider possible long-term advantages of purely metaheuristic variant, the maximum number of evaluations is doubled for all cases, compared to standard parameter setting in FSD-ES. The plots of *ERT* and *SR* for each variant are provided in Figure 11. It reveals that both the specialized penalty term and the resizing step provide significant contribution to FSD-ES. Suppressing either of them results in a detectable performance decline. In particular, suppressing both of them at the same time results in a drastic deterioration in *SR* and *ERT*.



(a) (b)

Figure 11. Performance of the different variants of FSD-ES on the 47-bar problem a) ERT and b) SR

CHAPTER 5. FULLY STRESSED DESIGN EVOLUTION STRATEGY FOR SIMULTANEOUS TOPOLOGY, SHAPE AND SIZE OPTIMIZATION

The decency and competency of FSD-ES was demonstrated in the previous chapter for shape and size optimization. However, as it was discussed in Section 3.2, the most effective and challenging scenario includes topology optimization as well. In this chapter[3], FSD-ES is generalized for TSS optimization problems. The steps of the improved algorithm, FSD-ES^{TSS,} are elaborated in details. Some test problems are selected from literature, on which the performance of FSD-ES^{TSS} is evaluated and compared with the best available methods in the literature.

5.1. Algorithm Details

Apart from topology optimization phase, most component are similarly to the previous version. We limit this code to simplified design specifications as it is the common scenario in most previous studies on TSS optimization.

5.1.1. Problem Representation

In FSD-ES^{TSS}, each design, $\boldsymbol{\theta}$, is represented by a set of three vectors:

- $\boldsymbol{M}$ is a vector of size N_m, whose elements are Boolean variables that determine whether a member is active (M_i=1) or passive (M_i=0) in the design.
- $\boldsymbol{X}$ is a vector of size DN_n, whose elements are continuous variables that determine nodal coordinates, where D=2 for planar and D=3 for spatial trusses.
- $\boldsymbol{A}$ is a vector of size N_m that determines member cross-sectional areas.

As it can be observed, only topology representation is new, in comparison with the earlier version. Such representation resembles the work of Rajan [98], where Boolean, continuous, and

[3] This chapter uses some materials from our previously published work [19], available at http://dx.doi.org/10.1080/0305215X.2014.947972. The publisher's policy allows reuse of the materials published by the authors in their dissertation.

discrete variables were allocated for topology, shape, and size respectively It is also possible to eliminate topology variables by passivating those members whose cross-sectional areas are smaller than a predefined value, as performed in [90]. However, this means that members with smaller cross-sectional areas are more likely to be passivated (stronger mutations are less probable). Similarly, a member with a large cross-section is unlikely to be passivated in the immediate subsequent iteration. When using an independent variable for presence/absence of a member, the probability of presence of a member is independent of the cross-sectional area.

In the employed topology representation, presence/absence of nodes is determined by presence/absence of the members connecting to that node. Furthermore, a sampled topology must satisfy some preliminary conditions. For example, it must be kinematically stable and has all basic nodes, otherwise it is considered as an unacceptable topology. The overall number of independent variables (N_{VAR}) is equal to the sum of the number of independent topology (N_{top}), shape (N_{shape}) and size (N_{size}) variables. N_{shape} and N_{size} are set to the number of independent coordinates and member section variables, respectively. The value of N_{top} is defined as the binary logarithm of the number of topologically distinct acceptable designs[4]. In this case, if all topologies are acceptable, N_{top} is equal to the number of topology variables; however, many sampled topologies will turn out to be unacceptable, which will be discarded without evaluation. This reduces the size of the space of the acceptable topologies. Accordingly, the value of N_{top} is smaller than the number of topology variables.

[4] This can be easily estimated by uniform sampling of a reasonable number of topologies (for example 1,000) subset of the ground structure and calculating the ratio of the acceptable topologies to all sampled topologies. As the overall number of topologies is known, the number of acceptable topologies can easily be estimated.

5.1.2. Initial Values

For the first iteration, recombinant design can be randomly selected within the bounds. As this design is not evaluated, it does not necessarily belong to the given discrete set. The recombinant point, consists of vectors of design variables $\boldsymbol{\theta}_{\text{mean}}=\{\boldsymbol{M}_{\text{mean}}, \boldsymbol{X}_{\text{mean}}, \boldsymbol{A}_{\text{mean}}\}$ and their corresponding vectors of strategy variables $\boldsymbol{\sigma}_{\text{mean}}=\{\boldsymbol{\sigma}_{M\text{mean}}, \boldsymbol{\sigma}_{X\text{mean}}, \boldsymbol{\sigma}_{A\text{mean}}\}$. An independent mutation step is allotted for each design variable. The values of the strategy variables are set to a fraction of the corresponding search range: 1/2, 1/3, and 1/4 for Boolean, discrete, and continuous variables, respectively.

5.1.3. Mutating Topology Variables

First, the strategy parameters of topology variables are mutated:

$$\boldsymbol{\sigma}_{M_j} = \boldsymbol{\sigma}_{M\text{mean}} \otimes \left(\exp(\tau_0 N_j) \exp\left(\tau \boldsymbol{\mathcal{N}}_{N_m}(0,1)\right)\right) \tag{24}$$

where $\boldsymbol{\sigma}_{Mj}$ is the vector of step sizes for topology variables of the j-th solution, N_j is a random number sampled from the standard normal distribution, τ_0 and τ are learning rates which are set to $0.5(N_{\text{VAR}})^{-0.5}$ and $0.5(N_{\text{VAR}})^{-0.25}$ respectively.

Once the value of $\boldsymbol{\sigma}_{Mj}$ is determined, it is used to mutate topology variables. The truncated normal distribution is used to sample $\boldsymbol{M}_j$, as explained in section 4.1.3. The center of mutation is $\boldsymbol{M}_{\text{mean}}$, the standard deviation is $\boldsymbol{\sigma}_{Mj}$, and [0, 1] is the truncated range. Since $\boldsymbol{M}_j$ consists of Boolean variables, the stochastic rounding strategy is performed to round values of $\boldsymbol{M}_j$ to 0 or 1.

After rounding, the generated topology, $\boldsymbol{M}_j$, is accepted if it is stable and includes all basic nodes of the structure, otherwise discarded and this step starts again from the beginning. To check stability, the condition number of the stiffness matrix of the generated topology is calculated while the shape and size of the design are equal to their corresponding values in the recombinant design.

If the condition number is larger than a predefined value, e.g. 10^{10}, the design is considered unstable[5].

5.1.4. Mutating Shape Variables

Having determined the topology, shape of the design is determined in this step. Step sizes are mutated first:

$$\boldsymbol{\sigma}_{X_j} = \boldsymbol{\sigma}_{X\text{mean}} \otimes \left(\exp(\tau_0 N_j) \left[\boldsymbol{c}_j \otimes \exp\left(\tau \boldsymbol{\mathcal{N}}_{DN_n}(0,1) \right) \right] \right) \tag{25}$$

where $\boldsymbol{\sigma}_{Xj}$ is the vector of step sizes for shape variables of the j-th individual, N_j is identical to its value in Section 5.1.3 and, c_{jk}=1 if the node corresponding the k-th coordinate is active and c_{jk}=0 otherwise. This means that step sizes of coordinates of passive nodes do not undergo component-wise mutilation. Similarly, coordinates of passive nodes remain equal to the corresponding values of $\boldsymbol{X}_{\text{mean}}$, since their variation have no effect on fitness and just perturbs current values which are resulted from previous iterations. Coordinates of active nodes are modified by mutation of the corresponding variables of $\boldsymbol{X}_{\text{mean}}$. Similarly to the previous step, $\boldsymbol{X}_j$, the vector of nodal coordinates, is sampled from the truncated normal distribution. The center of mutation is $\boldsymbol{X}_{\text{mean}}$, the standard deviation is $\boldsymbol{\sigma}_{X\,j}$ and the truncated range is the search range of shape variables.

[5] Determinant of the stiffness matrix of an unstable design is zero. Theoretically, this can be used to check stability of a topology, since calculation of determinant is much faster than the condition number; however, since the software calculates the determinant numerically, it is likely that a non-zero value is returned for and unstable truss. Defining a threshold is not a reliable approach since determinant of $[K_{ij}]_{n\times n}$ is multiplied by α^n if the cross-sectional areas are multiplied by α. Such multiplication does not change stability condition of a truss but significantly varies the determinant.

5.1.5. Mutating Size Variables

Having determined the topology and shape, the size of the structure is determined in this step. Strategy parameters are mutated first:

$$\boldsymbol{\sigma}_{A_j} = \boldsymbol{\sigma}_{A\text{mean}} \otimes \left(\exp(\tau_0 N_j)\left[\boldsymbol{M}_j \otimes \exp\left(\tau \boldsymbol{\mathcal{N}}_{N_m}(0,1)\right)\right]\right) \tag{26}$$

where $\boldsymbol{\sigma}_{Aj}$ is the vector of step sizes for size variables of j-th individual. M_{ji} is the topology variable associate with the i-th member, therefore, step sizes of passive members do not undergo component-wise mutation. Similarly, $\boldsymbol{A}_j$, the vector of cross-sectional areas of the candidate design, is sampled from the truncated normal distribution. The center of mutation is $\boldsymbol{A}_{\text{mean}}$, $\boldsymbol{\sigma}_{Aj}$ is the standard deviation and the search range of size variables is the truncated range. The stochastic rounding strategy is employed to round the element values of $\boldsymbol{A}_j$ to a value in the available set of sections. The section of passive members remains equal to the corresponding section of the recombinant design.

5.1.6. Evaluation

In this step, the generated design ($\boldsymbol{\theta j}$) is evaluated and member stresses and nodal displacements are computed:

$$\boldsymbol{\sigma}_j = [\sigma_{ji}], \sigma_{ji} = \begin{cases} \max\limits_{l}\{|\sigma_{jil}/\sigma^{\text{all}}|\} & \text{if } M_{ji} = 1 \\ 0 & \text{if } M_{ji} = 0 \end{cases}, i = 1,2,\dots,N_m$$

$$\boldsymbol{f}_j = [f_{ji}], f_{ji} = \begin{cases} \max\limits_{l}\left\{0, \dfrac{-L_i^2 \sigma_{jil}}{\alpha E A_i}\right\} & \text{if } M_{ji} = 1 \\ 0 & \text{if } M_{ji} = 0 \end{cases}, \quad i = 1,2,\dots,N_m$$

$$\boldsymbol{u}_j = [u_{jk}], u_{jk} = \begin{cases} \max\limits_{l} \left\{ \dfrac{u_{jkl}}{u_k^{all}} \right\} & \text{if the corresponding node is active} \\ 0 & \text{otherwise} \end{cases}$$

$$k = 1,2, \dots, DN_n \tag{27}$$

Vectors $\boldsymbol{\sigma}_j$, $\boldsymbol{f}_j$, $\boldsymbol{s}_j$ and $\boldsymbol{u}_j$ store the ratios of calculated stress, compressive axial load, slenderness ratio and nodal displacement to their allowable limits in the most critical load case, respectively. Similarly to the previous version (Section 4.1.5), the penalty term is based on the estimated required increase in the structural weight if all constraints are to be satisfied:

$$p_{ji} = \max\left\{1, \sigma_{ji}\ , \sqrt{f_{ji}}, \max_i\{u_{ji}\}\right\}$$

$$f_j = W_j + \rho \sum_{i=1}^{e} c_{Pi} A_{ji} l_{ji} \left(p_{ji}^2 - 1\right), \tag{28}$$

This equation implies that if areas of all members are multiplied by the corresponding q_{ji}, the resultant truss supposedly satisfies all constraints.

5.1.7. Resizing

Having analyzed the design, the active members in design are resized. The concept and formulation of the resizing are similar to the earlier version. The resized design ($\boldsymbol{\theta}_{j+\lambda}$) is identical to the original one ($\boldsymbol{\theta}_j$) except for the size variables and the corresponding step sizes. If A_{ji} is passive, then $A_{j+\lambda,i}$=A_{ji}, otherwise it is calculated as follows:

$$A_{j+\lambda,i} = \begin{cases} A_{ji} \times \max\left\{\sigma_{ji}\,,\sqrt{f_{ji}}\right\} & \text{if } \max\left\{\sigma_{ji}\,,\sqrt{f_{ji}}\right\} \geq 1 \\ A_{ji} \times \left(\begin{array}{c} 1+\exp(1-2c_{Pi}) \\ \times\left(\max\left\{\sigma_{ji}\,,\sqrt{f_{ji}}\right\}-1\right) \end{array} \right) & \text{otherwise} \end{cases} \tag{29}$$

If the given set of areas is discrete, each member area is rounded to its upper value. The corresponding step sizes are updated similarly to the earlier version of FSD-ES:

$$\sigma_{A_{j+\lambda,i}} = \begin{cases} \left(1.9\sigma_{A_{ji}}\left|A_{Ri} - A_{j+\lambda,i}\right|\right)^{0.5} & \text{if } m_{ji} = 1 \\ \left(\sigma_{A_{ji}}\right) & \text{if } m_{ji} = 0 \end{cases} \tag{30}$$

The fitness of the recently generated individuals is calculated using equation (28). This process is repeated from Section 5.1.3 until (λ+λ) candidate solutions are generated.

5.1.8. Recombination

The μ-best solutions of 2λ evaluated solutions are selected to update the recombinant design ($\boldsymbol{\theta}_{\text{mean}}$ and $\boldsymbol{\sigma}_{\text{mean}}$). The new values of $\boldsymbol{X}_{\text{mean}}$, $\boldsymbol{A}_{\text{mean}}$, $\boldsymbol{\sigma}_{M\text{mean}}$, $\boldsymbol{\sigma}_{X\text{mean}}$ and $\boldsymbol{\sigma}_{A\text{mean}}$ are specified by weighted average of the selected individuals:

$$\boldsymbol{X}_{\text{mean}} \leftarrow \sum_{j=1}^{\mu} w_j \boldsymbol{X}_j, \quad \boldsymbol{A}_{\text{mean}} \leftarrow \sum_{j=1}^{\mu} w_j \boldsymbol{A}_j,$$

$$\boldsymbol{\sigma}_{M_{\text{mean}}} \leftarrow \prod_{j=1}^{\mu} \boldsymbol{\sigma}_{M_j}{}^{w_j}\,,\ \boldsymbol{\sigma}_{X_{\text{mean}}} \leftarrow \prod_{j=1}^{\mu} \boldsymbol{\sigma}_{X_j}{}^{w_j}\,,\ \boldsymbol{\sigma}_{A_{\text{mean}}} \leftarrow \prod_{j=1}^{\mu} \boldsymbol{\sigma}_{A_j}{}^{w_j} \tag{31}$$

where μ=λ/2 and w_j's are logarithmically decreasing weight.

Updating $\boldsymbol{M}_{\text{mean}}$ is more complicated, since many sampled topologies were rejected without evaluation. This means distribution of topology variables in the population is not isotropic anymore. Topologies with fewer members are more likely to be rejected since they have a smaller chance to be kinematically stable. Therefore, using the same idea for updating topology variable

(i.e. $\boldsymbol{M}_{\text{mean}} \leftarrow \sum_{j=1}^{\mu} w_j \boldsymbol{M}_j$) results in a bias towards larger values for topology variables, such that under random selection of parents, the values of the topology variables in the recombinant design is likely to increase. This is a highly undesirable feature since only the selection operator should guide the population to a specific direction. As an alternative, update of $\boldsymbol{M}_{\text{mean}}$ is performed by comparing the fraction of designs in which M_{ji}=1 in the offspring and parental populations. For example, if the *i*-th member is active in 0.8 of the offspring but 0.6 of the selected parents, it is concluded that presence of this member is a disadvantage (0.6–0.8<0), and hence the *i*-th component of $\boldsymbol{M}_{\text{mean}}$ should decrease to make presence of this member less likely in the offspring of the next generation. Accordingly, the following relation for updating the topology vector of the recombinant point is suggested:

$$\boldsymbol{M}_{\text{mean}} \leftarrow \boldsymbol{M}_{\text{mean}} + c_M \left(\sum_{j=1}^{\mu} w_j \boldsymbol{M}_j - \frac{1}{2\lambda} \sum_{j=1}^{2\lambda} \boldsymbol{M}_j \right) \tag{32}$$

In the above equation, $\sum_{j=1}^{\mu} w_j \boldsymbol{M}_j$ and $\sum_{j=1}^{2\lambda} \boldsymbol{M}_j / 2\lambda$ are the (weighted) average topology variables of the selected parents and the whole population, respectively. Now under random selection of the parents, the expected change of M_{mean} is zero, and only the fitness function decides whether elements of M_{mean} should increase or not. Parameter c_M specifies the learning rate for the topology variables, which is assumed to depend on the number of parents, N_{VAR} and N_{top}. The following equation for updating c_{M} is proposed:

$$\boldsymbol{c_M} = \frac{\mu_{\text{eff}} + 2}{\mu_{\text{eff}} + 5 + \sqrt{N_{\text{VAR}} \times N_{\text{top}}}}, \quad \mu_{\text{eff}} = \left(\sum_{i=1}^{\mu} w_j \right) / \left(\sum_{i=1}^{\mu} w_j^2 \right)^{0.5} \tag{33}$$

5.1.9. Updating Parameters

The penalty coefficients are updated in this section similarly to the earlier version of FSD-ES.

$$\psi_i^{(\mu)} \leftarrow (1-\tau)\psi_i^{(\mu)} + \tau \sum_{j=1}^{\mu} \operatorname{sgn}(q_{ji} - 1) w_j$$

$$\psi_i^{(\lambda)} \leftarrow (1-\tau)\psi_i^{(\lambda)} + \tau \sum_{j=1}^{\lambda} \frac{\operatorname{sgn}(q_{ji} - 1)}{\lambda}$$

$$c_{Pi} \leftarrow \max\left\{0.5, c_{Pi}\sqrt{0.5 + \psi_i^{(\mu)}}, c_{Pi}\sqrt{0.5 + \psi_i^{(\lambda)}}\right\} \tag{34}$$

where $\psi_i^{(\mu)}$ and $\psi_i^{(\lambda)}$ represent the fraction of parents and offspring in which the *i*-th member has violated a constraint. If a displacement constrain is violated, all members are assumed to have violated constraint.

As the optimization process converges, a considerable proportion of acceptable topologies will no longer be produced. Accordingly, some nodes or members may exist or vanish in almost all generated topologies in the next iteration. The effective number of topology variables ($\tilde{N}_{\text{top}}$) is updated iteratively. The term $\left(\frac{1}{2\lambda}\sum_{j=1}^{2\lambda} m_{ji}\right)$ is the fraction of offspring in which the *i*-th member is active. If this value is close to 1, it means that the *i*-th member is active in most offspring. Similarly, if it reaches 0, this member is passive in most offspring. In such situations, it is concluded that the algorithm has decided on the corresponding topology variable. The value of $0 \leq \left|\left(\frac{1}{2\lambda}\sum_{j=1}^{2\lambda} m_{ji}\right) - 0.5\right| \leq 0.5$ can therefore signify the diversity of the values of the *i*-th topology variable in the population. The effective number of topology variables ($\tilde{N}_{\text{top}}$) is measured as follows:

$$\widetilde{N}_{\text{top}} = \left(1 - 4\sum_{i=1}^{e}\left|\left(\frac{1}{2\lambda}\sum_{j=1}^{2\lambda} m_{ji}\right) - 0.5\right|^{2}\right) N_{\text{top}} \tag{35}$$

For the final stages in which m_{ji}'s=0 or m_{ji}'s=1, $\widetilde{N}_{\text{top}} = 0$. Similarly, since the coordinates of passive nodes and the cross-sectional areas of passive members are not modified, the effective number of shape and size variables ($\tilde{N}_{\text{shape}}$ and $\tilde{N}_{\text{size}}$) can be smaller than N_{shape} and N_{size}. These values are the average number of active shape and size variables, respectively. Having computed $\tilde{N}_{\text{shape}}$ and $\tilde{N}_{\text{size}}$, the learning rates are updated:

$$\tau_0 \leftarrow \frac{1}{2\sqrt{\widetilde{N}_{\text{VAR}}}}, \tau \leftarrow \frac{1}{2\sqrt[4]{\widetilde{N}_{\text{VAR}}}}, c_M \leftarrow \frac{2+\mu_{\text{eff}}}{5+\mu_{\text{eff}}+\sqrt{\widetilde{N}_{\text{VAR}}\times\widetilde{N}_{\text{top}}}} \tag{36}$$

where $\tilde{N}_{\text{VAR}}=\tilde{N}_{\text{top}}+\tilde{N}_{\text{shape}}+\tilde{N}_{\text{size}}$ is the updated value for the effective number of design variables.

5.1.10. Parameter Tuning

All control parameters of FSD-ES are set to their recommended values. The population size, should logically be proportional to the problem complexity. A more sophisticated parameter tuning, in comparison with the earlier version, is developed for FSD-ES$^{\text{TSS}}$. The population size (λ) is set using the following relation, which considers the number of design variables and constraints:

$$\lambda=[(0.2\times K_1\times K_2)^{0.5}+5.5]$$

$$K_{VAR} = N_{top}+N_{shape}+\grave{N}_{size}+(N_{top}\times N_{shape})^{0.5}+(N_{shape}\times\grave{N}_{size})^{0.5}+$$

$$+(\grave{N}_{size}\times N_{top})^{0.5}+(N_{top}\times N_{shape}\times\grave{N}_{size})^{1/3}$$

$$K_{CON}= (c_{displacement}\times N_m\times\grave{N}_{size}/N_{size}+c_{stress}\times N_n\times D)\times N_l\times D \quad (37)$$

where $\grave{N}_{size} = N_{size} - 0.5\times N_{top}$ is the adjusted number of size variables. $c_{displacement}$ is 1 if nodal displacements are constrained and 0 otherwise. Similarly, c_{stress} is 1 if members are subject to stress or buckling constraints and 0 otherwise. K_{VAR} and K_{CON} are based on the complexity caused by the number of design variables and constraints, respectively. A complex ground structure with a small number of design variables still has many constraint (e.g. all members should satisfy stress constraints). Therefore, although K_{VAR} is small, K_{CON} is great. The value of λ is set based on both K_{VAR} and K_{CON}; therefore, the computed λ for this problem will be greater than the one computed for a simple ground structure with similar number of design variables.

FSD-ES does not employ data for the maximum number of iterations; however, by default, we set $MaxIter$=20λ+100. Al parameters of FSD-ES^{TSS} are set based on a priori known features of the problem and therefore there is no problem dependent parameter tuning.

5.1.11. Flowchart of the Proposed Algorithm

The flowchart of FSD-ES is provided in Figure 12.

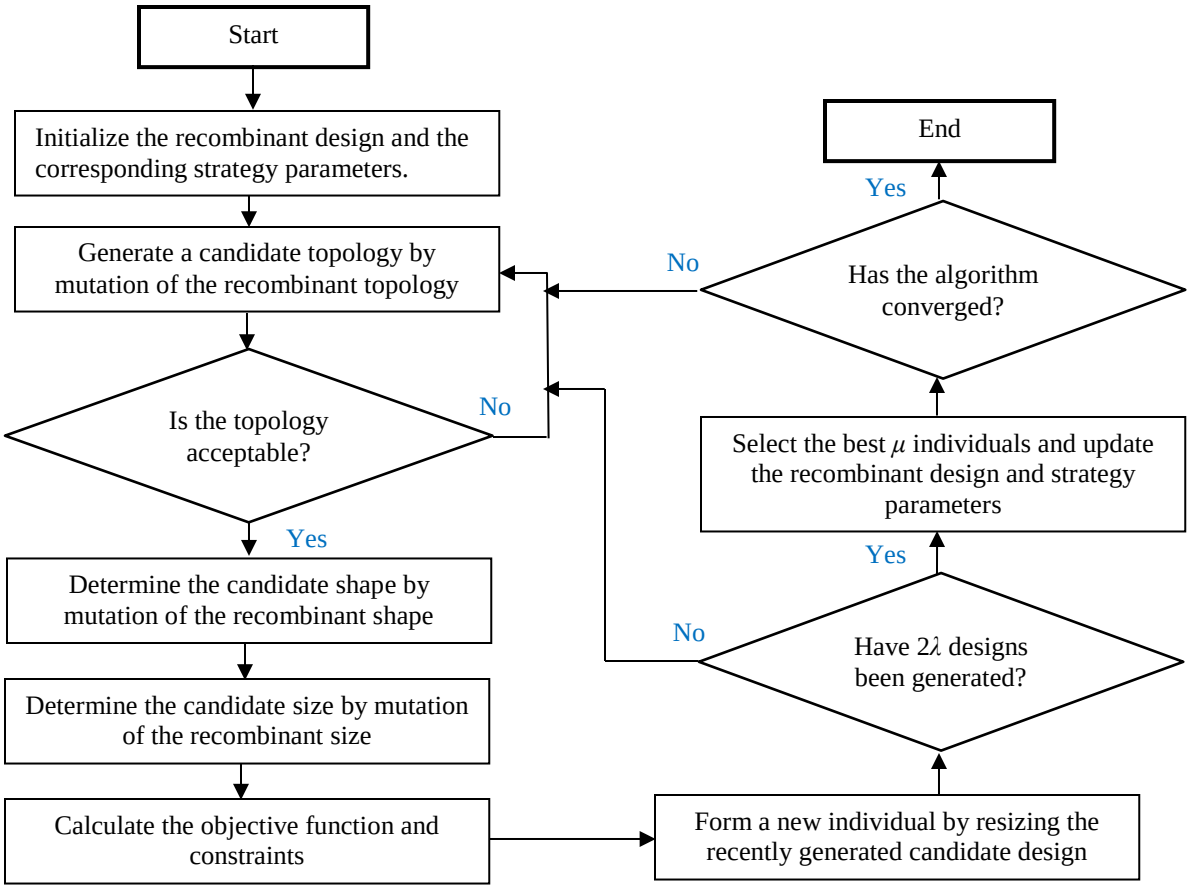


Figure 12. Flowchart for FSD-ES for TSS optimization

5.2. Numerical Evaluation

In this section, several qualitatively distinct problems are selected, on which FSD-ES is evaluated and compared with the most competent methods in the literature.

5.2.1. Test Problems

Test problems considering only one aspect of truss optimization are excluded as the concentration of this article is on TSS optimization problems. Some prevalent, albeit too simple, test problems are excluded and a brief explanation that highlights distinctive features of each selected test problem is presented.

The first test is a TSS optimization of a 15-bar truss adapted from [101]. The ground structure is depicted in Figure 13. This is quite a simple TSS problem since node 4 is intuitively redundant and other nodes may not be eliminated. Data required for simulation of this problem are presented in Table 5.

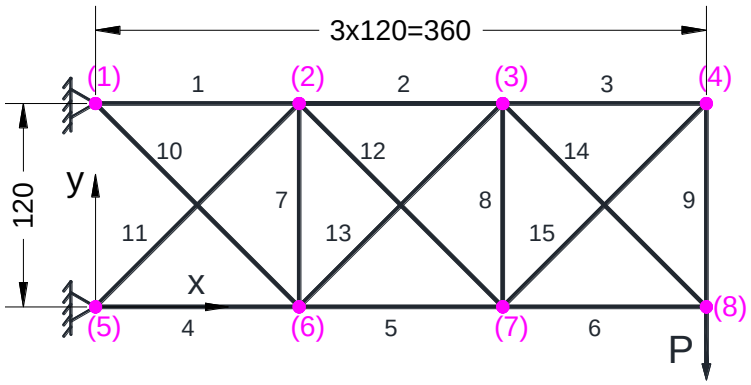


Figure 13. Ground structure of the 150-bar problem

Table 5. Simulation data for the 15-bar truss problem

Design Variables	**Topology (15)**	M_i, i=1,2,…,15		
	Shape (8)	x_2=x_6; x_3=x_7; y_2; y_3; y_4; y_6; y_7; y_8		
	Size (15)	A_i, i=1,2,…,15		
Constraints	**Stress**	$(\sigma_c)_I$, $(\sigma_t)_i \leq$ 25 ksi, i=1,2,…,15		
	Displacement	None		
	Buckling	None		
Search Range	**Shape Variables**	100″ ≤ x_2 ≤ 140″; 220″≤ x_3 ≤ 260″; 100″ ≤ y_2 ≤ 140″; 100″ ≤ y_3 ≤ 140″; 50″ ≤ y_4 ≤ 90″; −20″ ≤ y_6 ≤ 20″; −20″ ≤ y_7 ≤ 20″ ; 20″ ≤ y_8 ≤ 60″;		
	Size Variables	$A_i \in \mathbb{A}$, i = 1, . . . , 15 $\mathbb{A}$ ={0.111, 0.141, 0.174, 0.22, 0.27, 0.287, 0.347, 0.44, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.8, 3.131, 3.565, 3.813, 4.805,5.952, 6.572, 7.192, 8.525, 9.3, 10.85, 13.33, 14.29, 17.17, 19.18} (in^2)		
Loading		Nodes	F_x (kip)	F_y (kip)
		8	0	−10.0
Mechanical Properties		Modulus of elasticity: E=1.0×10^4 ksi Density of the material: ρ=0.1 lb/in^3		

The second problem is simultaneous TSS optimization of a 25-bar spatial truss adapted from [91]. Front and left views of the ground structure are depicted in Figure 14. The structure, but not the loading, is symmetric with respect to the *xz* and *yz* planes, which reduces the number of independent sections and coordinates to 8 and 5 respectively. Table 6 tabulates data required for simulation of this problem.

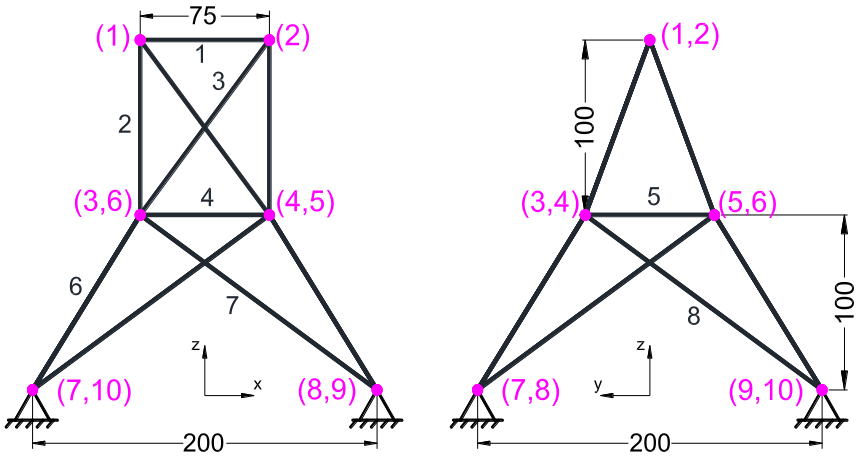


Figure 14. Ground structure of the 3D 25-bar problem

Table 6. Simulation data for the 25-bar spatial truss problem

Design Variables	**Topology (8)**	M_i , i=1,2,…,8 (Presence of other members depends on these 8 members)			
	Shape (5)	$x_4=x_5=-x_3=-x_6$; $x_8=x_9=-x_7=-x_{10}$; $y_3=y_4=-y_5=-y_6$; $y_7=y_8=-y_9=-y_{10}$; $z_3=z_4=z_5=z_6$			
	Size (8)	A_i , i=1,2,…,8 (Other member areas are dependent to these 8 variables)			
Constraints	**Stress**	$(\sigma_c)_i$, $(\sigma_t)_i \le 40$ ksi, i=1,2,…,25			
	Displacement	$u_k \le 0.35$ in, k=1,2,…, 30			
	Buckling	None			
Search Range	**Shape Variables**	$20'' \le x_4 \le 60''$; $40'' \le x_8 \le 80''$; $40'' \le y_4 \le 80''$; $100'' \le y_8 \le 140''$; $90'' \le z_4 \le 130''$			
	Size Variables	$A_i \in \mathbb{A}$, i = 1, . . . , 25, $\mathbb{A}$ ={0.1, 0.2, 0.3, … , 2.6} U {2.8, 3.0, 3.2, 3.4} in^2			
Loading		Nodes	F_x (kip)	F_y (kip)	F_z (kip)
		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
Mechanical Properties		Modulus of elasticity: E=1.0×10^4 ksi Density of the material: ρ=0.1 lb/in^3			

The third test is a two-tier 39-bar truss proposed by Deb and Gulati [90] and subsequently used in many studies [104, 92, 105], as a more comprehensive TSS optimization problem. The symmetric ground structure is depicted in Figure 15, in which overlapping members are illustrated with curved line segments. Structural symmetry is exploited to shrink the number of design parameters. Data for simulation of this problem are tabulated in Table 7. In comparison with the previous test, the number of design variables is greater. The problem has a low-fitness local optimum (W~214 lb) with minimal number of active nodes (7 members), which can trap algorithms that inherently favor simpler topologies.

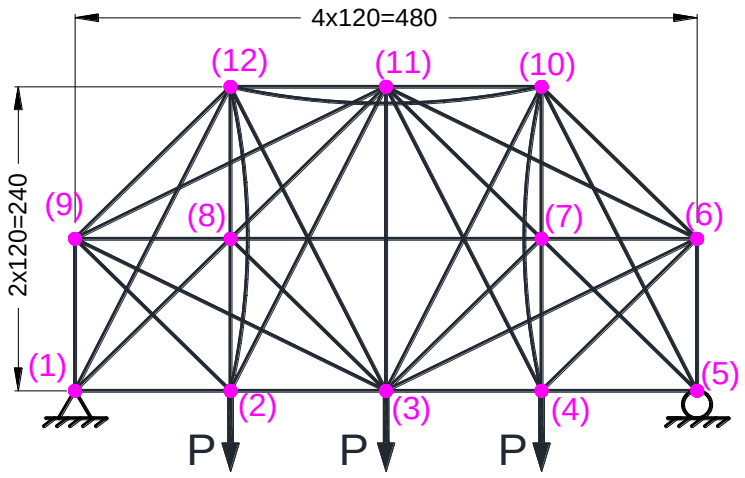


Figure 15. Ground structure of the two-tier 39-bar truss

Table 7. Data for simulation of the 39-bar truss problem

Design Variables	**Topology (21)**	M_i , i=1,2,…, 21 (21 members on the left and symmetry line)		
	Shape (7)	x_8, y_8, x_9, y_9, y_{11}, x_{12}, y_{12}		
	Size (21)	A_i , i=1,2,…,21 (21 members on the left and symmetry line)		
Constraints	**Stress**	$(\sigma_c)_i$, $(\sigma_t)_i \leq 20$ ksi;		
	Displacement	$u_k \leq 2$ in, k=1, 2, …, 24		
	Buckling	None		
Search Range	**Shape Variables**	The coordinate variables may vary up to ±120 inch with respect to their value in the ground structure		
	Size Variables	$A_i \in [0.05, 2.25]$ in^2 (continuous values)		
Loading		Nodes	F_x (kip)	F_y (kip)
		2,3,4	0	–20
Mechanical Properties		Modulus of elasticity: 1.0×10^4 ksi		
		Density of the material: ρ=0.1 lb/in^3		

The fourth problem is a 45-bar 2D truss employed by Deb and Gulati [90]. The structure weight is minimized while symmetry is ignored. The ground structure, depicted in Figure 16, has all pair-wise interconnection (all-to-all scheme). A vertical load of 10 kips is exerted on nodes 7, 8 and 9. Data for simulation of this problem are presented in Table 8.

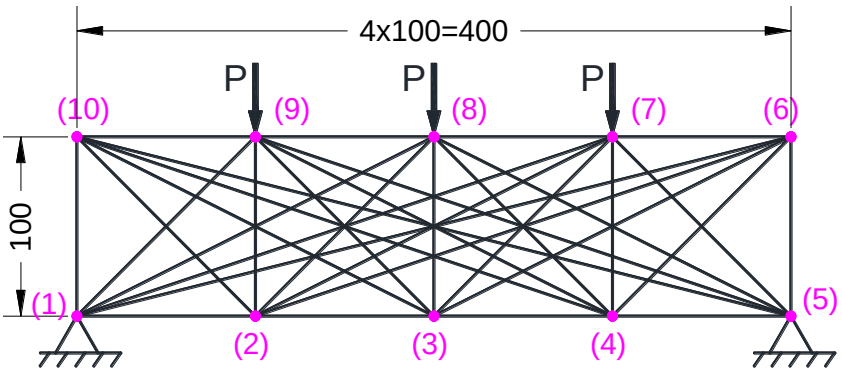


Figure 16. Ground structure of the 45-bar truss problem

Table 8. Data for simulation of the 45-bar truss problem

Design Variables	**Topology (45)**	M_i, i=1,2,…, 45 (one topology variable per member)		
	Shape (0)	None		
	Size (45)	A_i , i=1,2,…,45 (one size variable per member)		
Constraints	**Stress**	$(\sigma_c)_i$, $(\sigma_t)_i \le 25$ ksi;		
	Displacement	$u_k \le 2$ in, k=1, 2, …, 20		
	Buckling	None		
Search Range	**Size Variables**	$A_i \in [0.09, 1]$ in^2 (continuous areas)		
Loading		Nodes	F_x (kip)	F_y (kip)
		7,8,9	0	–10
Mechanical Properties		Modulus of elasticity: 1.0×10^4 ksi Density of the material: ρ=0.1 lb/in^3		

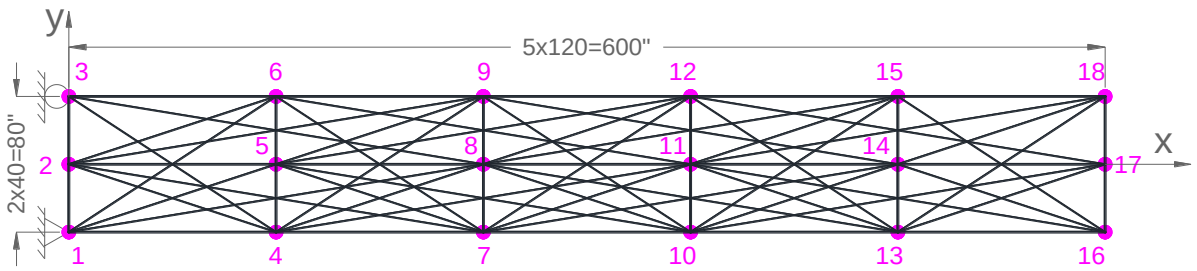


Figure 17. Ground structure of 68-bar truss problem

The final test is TSS optimization of a 68-bar truss subjected to two load cases at the end (Figure 17). Data required for simulation of this problem is presented in Table 9. Horizontal and vertical coordinates of the nodes may vary within 120 and 40 inch of the initial configuration respectively. The ground structure allows for elimination of most members and nodes except nodes

1, 3, and 17. This test problem is introduced in this study and has a significantly larger number of design variables compared to the previous test problems.

Table 9. Data for simulation of the 68-bar truss problem

Design Variables	**Topology (68)**	M_i , i=1,2,…,68 (One topology variable per member)		
	Shape (31)	y_{17}, x_i, y_i, i=2,4,5,6,…, 14,15,16,18		
	Size (68)	A_i , i=1,2,…,68 (One size variable per member)		
Constraints	**Stress**	$(\sigma_c)_i$, $(\sigma_t)_i \leq 20$ ksi, i=1, 2, …, 68		
	Displacement	$u_k \leq 2.5''$, k=1, 2, 3, …, 36		
	Buckling	$\lvert(\sigma_c)_i\rvert \leq \alpha E a_i / l_i^2$, i=1,2,…,68, α=3.96		
Search Range	**Shape Variables**	Horizontal coordinates may vary within ±120 in of their initial value. Vertical coordinates may vary within ±40 in of their initial value.		
	Size Variables	$A_i \in \mathbb{A}$, i = 1, . . . , 68 $\mathbb{A}$ ={0.111, 0.141, 0.174, 0.22, 0.27, 0.287, 0.347, 0.44, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.8, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.3, 10.85} (in^2)		
Loading		Nodes	F_x (kip)	F_y (kip)
	Case I	17	–50	0
	Case II	17	–50	–15
Mechanical Properties		Modulus of elasticity: E=3.0×10^4 ksi Density of the material: ρ=0.3 lb/in^3		

5.2.2. Performance Measures

The same performance measure (Section 4.2.2) is employed for performance evaluation. All control parameters of FSD-ES are set to their default values. Following the procedure explained in Section 5.1.1, the estimated value for N_{top} is 43.4, 8.25, 3.55, 18.94, and 65.5. Equation (37) leads to the population size of 35, 22, 39, 44, and 136 for problems 1 to 5 respectively. Accordingly, FSD-ES requires no ad hoc tuning effort, since all parameters are set based on known features of the problem.

5.2.3. Results and Discussion

Results from FSD-ES on the selected test suite are presented in this section. Each test is repeated 100 times independently and the calculated *ERT* accompanied by *SR* and FE_S is plotted versus the target weight. These plots are provided in Figure 18. Only feasible designs that satisfy

all constraints are considered in calculating *FEs* and *SR*. Table 10 summarizes the results of FSD-ES alongside the best available results in the literature. The best feasible solution found in this study is also plotted in Figure 19 and the corresponding data are presented in Table 11. Based on the results, the following conclusions can be made:

- For the 15-bar truss problem, Figure 18(a) demonstrates that FSD-ES needs 3,859 evaluations to reach W=72.50 lb, however, the *ERT* is more than five times larger (*SR*=0.19). A rapid deterioration of *SR* starts at the target weight of 75.0 lb, resulting in fast growth of the *ERT*. The best design found in this study weighs 69.585 lb, and 5% of independent runs could reach W=70.00 lb after an average of 8,508 evaluations. The best solution of SCPSO [94] weighs W=72.49 lb, and was reached after 12,500 FE, almost 4.2% heavier than the best solution of the proposed FSD-ES. The best solution of FA [93] weighs 74.33 lb, reached after 8,000 evaluations. The GA proposed in Rahami *et al*. [91] reached the best solution of 75.10 lb after 8,000 evaluations, which is about 7.9% heavier than the best solution found by the proposed FSD-ES. For this problem, FSD-ES surpasses the best existing algorithms both in efficiency and quality of the best final solution.
- Figure 18(b) demonstrates that for the 25-bar spatial truss, FSD-ES requires on average 8,660 evaluations to reach W=114.42 lb, however, the corresponding SR is low. To the authors' knowledge, the best reported solution for this problem is 114.36 lb [91], reached after 10,000 FE. The GA of Tang *et al*. [103] and FA by Miguel *et al*. (2013) could reach W=114.74 lb and W=116.58 lb after 6,000 FE respectively. The best solution of SCPSO (Gholaizadeh 2013) weighs 117.23 lb, reached after 8,000 FE. For this problem, performance of FED-ES is quite identical to the best available results.

- Figure 18(c) demonstrates that for the 39-bar truss problem, a gradual drop in *SR* initiates at W=190 lb, caused by the existence of several topologically distinct high fitness designs at this zone. 16% of independent runs could still reach W=182 lb, on average after 40,256 FE. The best feasible solution found in this study weighs 180.98 lb, which is about 3.7% lighter than and topologically different from the best reported solution in the literature. For this problem, Luh and Lin [104, 92] could reach W=188.73 lb and W=188.86 lb after 453,600 and 262,500 evaluations, respectively. A better result was obtained by Wu and Tseng [105], which could reach W=188.45 lb after 137,200 evaluations. The FA of Miguel *et al*. (2013) could reach W=191.30 lb after 50,000 evaluations. For this problem, FSD-ES considerably outperforms the previous methods both in efficiency and quality of the final design.
- There are some notable points regarding this problem pertaining to existence of several topologically distinct high-fitness designs. In 100 independent runs, FSD-ES converged to many distinct topologies. A number of these distinct final designs that weigh less than 188 lb are illustrated in Figure 20. To the author's knowledge, only topology #18 is reported in the literature, meaning the previous algorithms could not detect topologies that potentially result in lighter structures. Second, the best previously reported designs have 13-15 active members, however, topologies that could reach structures lighter than 187.5 lb generally have more members. This can pose a challenge for truss optimizers that are biased toward simpler structures. Finally, topologies of the best designs can hardly be guessed by engineering intuition, which highlights benefits of using a reliable truss optimizer.
- For the 45-bar truss problem, Figure 18(d) demonstrates that FSD-ES can reach a structural weight of 44.000 lb after 19,656 evaluations, if the run is successful. 90% of independent

runs converged to this weight and hence there is only a slight difference between the *ERT* and FE_S for this problem. To the authors' knowledge, the best solution for this problem is 43.99 lb, reported by Wu and Tseng [105], which was reached after 30,800 evaluations. The best design of FSD-ES is quite identical to theirs, but the required number of evaluations for the proposed FSD-ES is comparatively smaller.

- For the 68-bar truss problem, the best final solution (Figure 19(e)) has a reasonable topology where all three types of constraints were activated (see Table 11); however, no comparison with other specialized algorithms can be made at the current moment. Independent runs of FSD-ES converged to a variety of different designs ranging from 1,203.5 lb up to 1,555 lb (Figure 18(e)). This is probably due to the complexity of the ground structure and various reasonable topologies which may challenge truss optimizers. It is expected that a greater variance is observed in performance of different optimization methods on this problem.

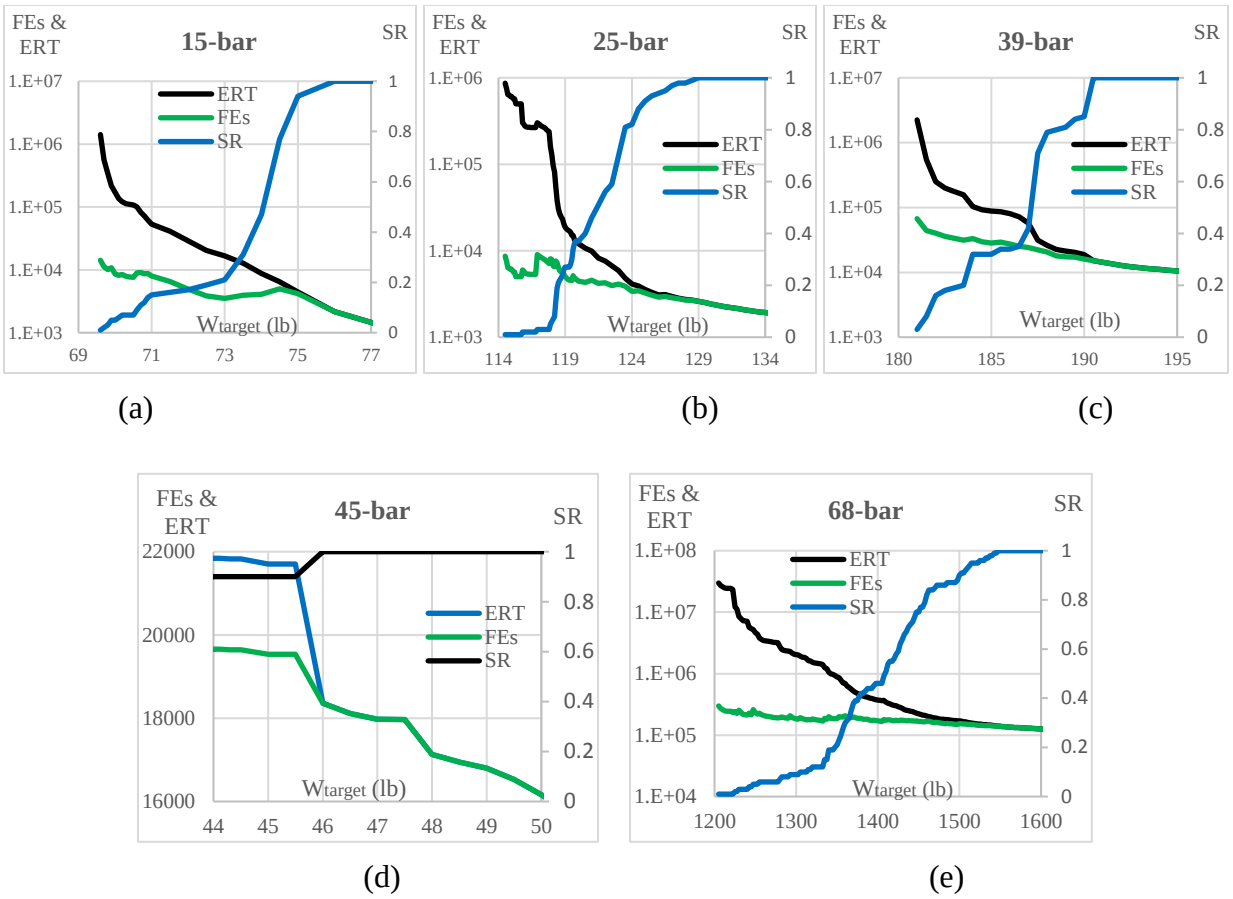


Figure 18. *ERT*, *SR* and *FE*S to reach arbitrary structural weights for the employed test problems: a) 15-bar, b) 25-bar, c) 39-bar, d) 45-bar and e) 68-bar truss problems.

Table 10. Comparison of the results from FSD-ES and the best available results in the literature. For FSD-ES, *FEs* for two selected target weights are reported.

Problem	**Method**	**Year**	W_{min}	W_{min}/W^*	**FEs**	**FEs/ FEs***
15-bar	GA [103]	2005	77.84 lb	1.118	8000	2.073
	SCPSO [94]	2013	72.51 lb	1.042	4500	1.166
	FA [93]	2013	74.33 lb	1.068	8000	2.073
	FSD-ES	2014	**69.60** lb	1.000	14,148	3.666
			72.50 lb	1.042	**3859**	1.000
25-bar	GA [103]	2005	114.74 lb	1.003	6000	1.345
	GA [91]	2008	**114.37** lb	1.000	10,000	2.241
	SCPSO [94]	2013	117.23 lb	1.025	4500	1.009
	FA [93]	2013	116.58 lb	1.019	6000	1.345
	FSD-ES	2014	114.50 lb	1.001	8656	1.940
			120.00 lb	1.049	**4462**	1.000
39-bar	GA [90]	2001	192.19 lb	1.062	504,000	17.908
	AA [104]	2008	188.73 lb	1.043	453,600	16.117
	MPDE [105]	2010	188.45 lb	1.041	137,200	4.875
	PSO [92]	2011	188.86 lb	1.043	262,500	9.327
	FA [93]	2013	191.30 lb	1.057	50,000	1.777
	FSD-ES	2014	**181.00** lb	1.000	67,273	2.390
			185.00 lb	1.022	**28,144**	1.000
45-bar	GA [90]	2001	44.033 lb	1.001	NA	NA
	MPDE [105]	2010	**43.99** lb	1.000	30,800	1.567
	FSD-ES	2014	44.00 lb	1.000	**19,656**	1.000
68-bar	FSD-ES	2014	**1205** lb	1.000	297,994	1.485
			1340 lb	1.112	**200,734**	1.000

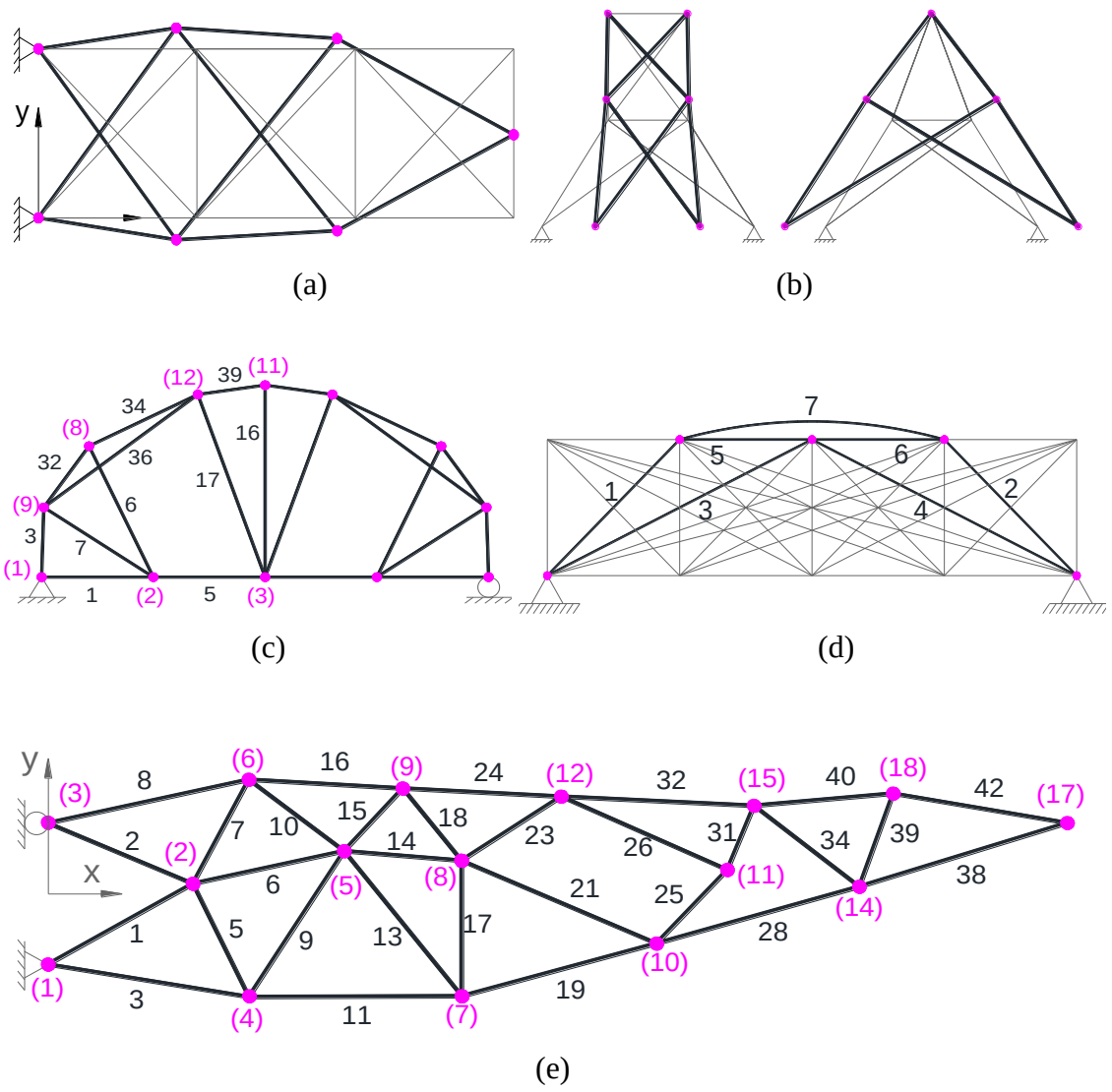


Figure 19. The best final design for a) 15-bar, b) 25-bar (front and side view), c) 39-bar, d) 45-bar and e) 68-bar truss problems. The overlapping member is depicted with curved line segment for the 45-bar problem.

Table 11. The best solutions found for the test problems using FSD-ES. Coordinates and areas are in inch and inch square, respectively. The ratio of the maximum stress, buckling load and displacement to the allowable limit as well as the overall weight are provided in the four last rows.

	45-bar truss	**15-bar truss**	**25-bar truss**	**39-bar truss**	**47-bar truss**	**68-bar truss**
	A_1 0.56569	x_2=x_6 100.000	x_4 38.871	x_8 53.990	x_2 100.972 A_{11} 0.9	x_2 85.217 x_{15} 415.232 A_{21} 0.111
	A_2 0.56569	y_2 135.135	y_4 61.521	y_8 151.253	x_4 80.477 A_{13} 2.7	y_2 5.752 y_{15} 49.954 A_{23} 0.111
	A_3 0.44721	x_3=x_7 229.819	z_4 119.179	x_9 5.191	y_4 136.870 A_{15} 0.8	x_4 119.117 y_{17} 40.000 A_{24} 2.142
	A_4 0.44721	y_3 124.426	x_8 49.415	y_9 84.278	x_6 64.391 A_{17} 2.7	y_4 -58.650 x_{18} 497.452 A_{25} 0.174
	A_5 0.18521	y_6 -16.966	y_8 137.942	y_{11} 219.957	y_6 247.049 A_{19} 0.7	x_5 175.036 y_{18} 56.890 A_{26} 0.111
	A_6 0.18521	y_7 -9.202	A_1 -	x_{12} 172.019	x_8 55.259 A_{21} 2.5	y_5 24.010 A_1 2.697 A_{28} 2.697
	A_7 0.21479	y_8 56.169	A_2 0.9	y_{12} 210.256	y_8 338.453 A_{23} 0.9	x_6 118.841 A_2 1.174 A_{31} 0.111
		A_1 0.954	A_3 0.1	A_1 0.092	x_{10} 48.733 A_{25} 0.7	y_6 64.732 A_3 3.565 A_{32} 2.142
		A_2 0.539	A_4 -	A_3 1.503	y_{10} 409.738 A_{27} 1.8	x_7 243.407 A_5 0.539 A_{34} 0.954
		A_4 0.954	A_5 -	A_5 0.980	x_{12} 43.474 A_{29} 0.9	y_7 -58.276 A_6 1.174 A_{38} 2.697
		A_5 0.539	A_6 1.0	A_6 0.560	y_{12} 472.148 A_{31} 1.1	x_8 243.804 A_7 1.174 A_{39} 0.44
		A_6 0.44	A_7 0.1	A_7 0.823	x_{14} 44.835 A_{33} 0.3	y_8 18.715 A_8 2.142 A_{40} 1.764
		A_{10} 0.44	A_8 0.1	A_{16} 0.311	y_{14} 512.190 A_{35} 1	x_9 209.080 A_9 0.287 A_{42} 1.764
		A_{11} 0.44		A_{17} 0.362	x_{20} 84.504 A_{37} 1.3	y_9 59.759 A_{10} 0.954
		A_{12} 0.22		A_{32} 1.216	y_{20} 630.347 A_{39} 0.9	x_{10} 358.090 A_{11} 3.131
		A_{13} 0.22		A_{34} 1.051	x_{22} 3.841 A_{41} 0.8	y_{10} -28.416 A_{13} 0.539
		A_{14} 0.44		A_{36} 0.050	y_{22} 591.145 A_{42} 1.1	x_{11} 400.205 A_{14} 0.44
				A_{39} 1.102	A_1 3.3 A_{43} 0.1	y_{11} 13.541 A_{15} 0.44
					A_3 1.1 A_{44} 0.1	x_{12} 302.656 A_{16} 1.764
					A_5 3.2 A_{45} 0.1	y_{12} 55.056 A_{17} 0.287
					A_7 1 A_{46} 0.1	x_{14} 477.849 A_{18} 0.44
					A_9 3 A_{47} 0.1	y_{14} 3.919 A_{19} 2.8
Stress	**1.0000**	**1.0000**	**0.4490**	**1.0000**	**1.0000**	**1.0000**
Buckling	**-**	**-**	**-**	**-**	**1.0000**	**1.0000**
Def.	**0.6250**	**-**	**1.0000**	**0.7708**	**-**	**1.0000**
Weight	**44.000 lb**	**69.585 lb**	**114.417 lb**	**180.983 lb**	**1846.52 lb**	**1203.51 lb**

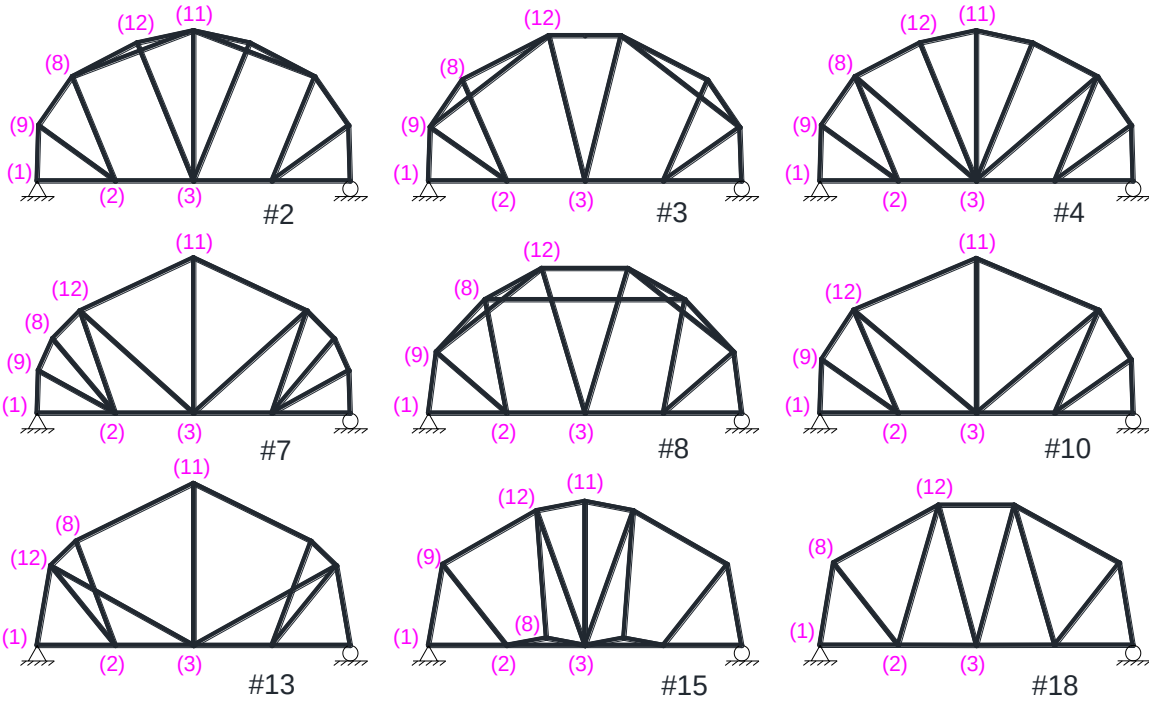


Figure 20. Some selected final designs for the 39-bar truss problem which have distinct topology: Topology #2: W=181.02 lb, Topology #3: W=181.38 lb, Topology #4: W=181.60 lb, Topology #7: W=182.37 lb, Topology #8:W=183.34 lb, Topology #10: W=183.89 lb, Topology #13: W=186.91 lb, Topology #15: W=186.96 lb, Topology #18: W=187.30 lb.

CHAPTER 6. IMPROVED FULLY STRESSED DESIGN EVOLUTION STRATEGY

In the previous chapters, we demonstrated that FSD-ES can outperform or at least compete with the best methods in the literature for each problem. In this chapter[6], we discuss some shortcomings in FSD-ES and alternatively propose an improved version, which is called FSD-ES II:

- The resizing step in FSD-ES does not take displacement constraints into account explicitly. It merely reduces the reduction of cross-sectional areas if displacement constraints are violated in many designs, since all members are assumed responsible for violating a displacement constraint. As discussed in Section 3.1, FSD assumptions provide some useful information to calculate contribution of each member to each nodal displacement constraint. Maximal utilization of such knowledge is not easy since each displacement constraint, in contrast to member-based constraint, is affected by cross-sectional areas of many members. In FSD-ES II, an improved version of the resizing technique is proposed which can explicitly take displacement constraints into account. We also propose an FSD-based heuristic to solve the resizing problem with respect to displacement constraints.
- The evolution strategy in FSD-ES follows the traditional concept of self-adaptation for adjusting mutation strength. The shortcoming of this approach, in compared to derandomized approach, was discussed in Section 2.5. In FSD-ES II, the employed evolution strategy follows principles of CMSA-ES, a contemporary ES that can compete with CMA-ES, the state-of-the-art ES. Because of its simplicity, and thus flexibility for tailoring for highly constrained mixed-variable TSS problems, CMSA-ES is preferred over

[6] This chapter uses some materials from our previously published work [133], available at http://dx.doi.org/10.1016/j.compstruc.2015.11.009. The publisher's policy allows reuse of the materials published by the authors in their dissertation.

CMA-ES in this study. We further simplify the method by assuming only diagonal elements of the covariance matrix are considered. This reduces the covariance matrix to vector $\boldsymbol{d}$, which is of the size of $\boldsymbol{\theta}$. This simplification is mainly performed to allow for using the truncated normal distribution for perturbation of the recombinant design.

- FSD-ES, similarly to a few other studies [93], discard the generated topologies if they are unstable, without counting this process as a function evaluation. There are some necessary conditions for stability of a truss which can be easily checked, such as the minimum number of members. It is possible that an unstable truss satisfies all the necessary conditions for stability and acceptability. In this case, the instability of the truss can be revealed by forming and analyzing the stiffness matrix (e.g. positive definiteness or condition number). The computation time for this process is in the same order of a full function evaluation, therefore, overlooking such computation effort may result in underestimation of the number of evaluations, as performed in a few other studies [93] . Moreover, it is possible that the algorithm sticks in the generation loop since almost all sampled topologies turn out to be unstable. On the other extreme perspective, some studies considered it as a function evaluation even if the instability of a design was revealed by checking the simple necessary conditions (such as [90]), which results in an overestimation of the evaluations. In FSD-ES II, the problem of under-estimated number of evaluations in FSD-ES is addressed by considering formation and analysis of the stiffness matrix a function evaluation.

FSD-ES-II, is explained in the subsequent section. Many components are similar to FSD-ES, therefore, they are briefly explained. New features and formulation are elaborated in details. FSD-ES is evaluated on several complicated test problems and the obtained results are compared with the best available results in the literature.

6.1. Algorithm Details

The FSD-ES II algorithm is explained in this section.

6.1.1. Problem Representation

In FSD-ES, each design, $\boldsymbol{\theta}$, is represented by a set of three vectors.

- $\boldsymbol{M}$ is a vector of size N_m, whose elements are Boolean variables that determine whether a member is active (M_i=1) or passive (M_i=0) in the design.
- $\boldsymbol{X}$ is a vector of size DN_n, whose elements are continuous variables that determine nodal coordinates, where D=2 for planar and D=3 for spatial trusses.
- $\boldsymbol{A}$ is a vector of size N_m, whose elements are discrete variables that determine member cross-sectional areas.

Therefore, a solution is determined by $\boldsymbol{\theta}=\{\boldsymbol{M}, \boldsymbol{X}, \boldsymbol{A}\}$ and the lower and upper bounds of the search space is defined by $\boldsymbol{\theta}^{\mathrm{L}}=\{\boldsymbol{M}^{\mathrm{L}}, \boldsymbol{X}^{\mathrm{L}}, \boldsymbol{A}^{\mathrm{L}}\}$ and $\boldsymbol{\theta}^{\mathrm{U}}=\{\boldsymbol{M}^{\mathrm{U}}, \boldsymbol{X}^{\mathrm{U}}, \boldsymbol{A}^{\mathrm{U}}\}$. The overall number of independent variables (N_{VAR}) is equal to the sum of the number of topology (N_{top}), shape (N_{shape}) and size (N_{size}) variables. In contrast to FSD-ES, N_{top} is simply the number of independent topology variables.

The recombinant ($\boldsymbol{\theta}_{\mathrm{mean}}=\{\boldsymbol{M}_{\mathrm{mean}}, \boldsymbol{X}_{\mathrm{mean}}, \boldsymbol{A}_{\mathrm{mean}}\}$) is deterministically initialized for the first iteration: $\boldsymbol{X}_{\mathrm{mean}}$ and $\boldsymbol{A}_{\mathrm{mean}}$ are the center of the range defined by the search space limits. $\boldsymbol{M}_{\mathrm{mean}}=\mathbf{1}$ so that most sampled solutions in the early iterations have sufficient number of members.

6.1.2. ES-based Sampling of New Designs

Solutions are sampled by applying a perturbation to the recombinant design. The global step size is mutated first and is subsequently used to generate $\boldsymbol{\theta}_j$:

$$\sigma_j = \sigma_{\text{mean}} \exp\big(\tau \mathcal{N}(0,1)\big)$$

$$\boldsymbol{\Theta}^{\text{up}} = \boldsymbol{F}\big(\boldsymbol{\theta}^{\text{U}}, \boldsymbol{\theta}_{\text{mean}}, \sigma_j \boldsymbol{d}\big), \boldsymbol{\Theta}^{\text{low}} = \boldsymbol{F}\big(\boldsymbol{\theta}^{\text{L}}, \boldsymbol{\theta}_{\text{mean}}, \sigma_j \boldsymbol{d}\big),$$

$$\boldsymbol{\theta}_j = \boldsymbol{F}^{-1}\big(\boldsymbol{u}(\boldsymbol{0}, \boldsymbol{1}) \otimes \big(\boldsymbol{\Theta}^{\text{up}} - \boldsymbol{\Theta}^{\text{low}}\big) + \boldsymbol{\Theta}^{\text{low}}, \boldsymbol{\theta}_{\text{mean}}, \sigma_j \boldsymbol{d}\big), \tag{38}$$

where $\boldsymbol{F}(\boldsymbol{\theta}^{\text{U}}, \boldsymbol{\theta}_{\text{mean}}, \sigma_j \boldsymbol{d})$ computes the cumulative function of normal distribution at $\boldsymbol{\theta}^{\text{U}}$ with mean of $\boldsymbol{\theta}_{\text{mean}}$ and standard deviation of $\sigma_j \boldsymbol{d}$ in an element-wise manner. $\boldsymbol{\theta}_j$ is sampled from the truncated normal distribution such that it falls within the limits defined by $\boldsymbol{\theta}^{\text{L}}$ and $\boldsymbol{\theta}^{\text{U}}$. The stochastic rounding technique (see Section 4.1.4) is applied to size and topology variables to create the actual design.

Having generated a sample design, the flowing necessary conditions are checked for kinematic stability and acceptability of the generated design:

- All necessary nodes must be active.
- The truss must have at least the minimum number of members required for kinematic stability: The number of active members plus the number of reactions from the supports must be equal or greater than the number of active nodes multiplied by *D*.
- The number of members connected to an active node plus the number of reactions on that node must be equal or greater than *D*.
- The number of members connected to an active node plus the number of reactions on that node must be *greater* than *D* if no external force is applied on that node.

The first three conditions have been commonly used for checking stability of a truss in the literature. If only *D* members are connected to a node upon which no reaction or external force is applied, the equilibrium conditions necessitate that the axial force be zero in all these members. Such a configuration increases the structural weight without contributing to kinematic stability or

load carrying capacity. The last condition is therefore introduced to assure that such a configuration does not exist in the sampled topology. It is not a necessary condition for stability but for efficiency of a topology.

Checking these necessary, but not sufficient conditions for kinematic stability, acceptability and efficiency of the sampled tropology is computationally inexpensive, since it does not require performing a finite element analysis. If the generated design does not satisfy any of these conditions, it is discarded and a new topology is sampled. This process continues until a design that satisfies all these conditions is generated and accepted. We ignore the computation time required by resampling and checking these necessary conditions as they are negligible in comparison with finite element analysis.

For each accepted design, vector $\boldsymbol{\varpi}_j^X$ of size DN_n and vector $\boldsymbol{\varpi}_j^A$ of size N_m store the data pertaining to activeness or passiveness of shape and size variables in $\boldsymbol{\theta}_j$, respectively. These vectors consist of Boolean variables, in which '1' refers to an active and '0' refers to a passive size or shape variable. These vectors are utilized to exclude the effect of passive shape and size variables during the updating process, which will be explained in Section 6.1.5.

6.1.3. Design Evaluation

In this study, the command '*rcond*'[7] of MATLAB© software is used to calculate an estimate for the condition number of the stiffness matrix. It is assumed that truss is kinematically stable if and only if $rcond(\mathbf{K}_j)>10^{-12}$, where $\mathbf{K}_j$ is the stiffness matrix of the structure. If the truss is found to be kinematically unstable, the cost of the truss is assigned as follows:

[7] This command is faster than command '*cond*', which calculates the exact condition number of the matrix.

$$f(\boldsymbol{\theta}_j) = 10^{100} \times \left(1 + \frac{\text{null}(\mathbf{K}_j) + N_j^{\text{red}}}{N_j^{\text{req}}}\right), \quad \text{if} \quad rcond(\mathbf{K}_j) \leq 10^{-12} \tag{39}$$

where *null*($\mathbf{K}_j$) is nullity of $\mathbf{K}_j$, N_j^{req} is the minimum number of members required for kinematic stability of the truss and N_j^{red} is the number of redundant members. In the defined penalty term:

- Applying the coefficient of 10^{100} ensures all kinematically unstable structures are inferior to all stable ones.
- *null*($\mathbf{K}_j$) is an integer number which shows how much the structure is far from kinematic stability, or equivalently, how kinematically unstable the structure is. This provides a unique tool for comparing even kinematically unstable designs so that by preferring less kinematically unstable designs, the population size is directed towards regions with kinematically stable designs.
- N_j^{red} is the number of redundant members. It increases the penalty for an unstable design that has more redundant members, since it could not provide stability even with having extra members.
- N_j^{req} is the minimum number of members required for kinematic stability of the truss. It increases the penalty for a fixed nullity if the required minimum number of members is small. For example, it calculates a smaller penalty for a truss with many nodes and members and nullity=1 in comparison with a truss with a few nodes and members and the same nullity.

Using Equation (39) is and should be considered as a function evaluation, even though the full analysis of the structure is not performed, since it requires formation of the stiffness matrix and computation of the nullity, the required time for which is in the same order of a full analysis. This is the case with FSD-ES II, in contrast to the earlier version and some other studies such as [93],

in which such designs were simply rejected without considering this process as a function evaluation, resulting in an underestimation of the number of evaluation, and besides, it might encounter stagnation in problems were most members must be active to have a stable topology.

If the design is kinematically stable, the system response to the external load(s), including the member forces and nodal displacements, as well as the response to unit loads (required to compute c_{ikl}'s in Equation (6)), are computed. The same concept is utilized in definition of the penalty term: The penalty term is the estimated required increase in the cross-sectional areas, and thus in the weight, such that all constraints are satisfied. This required increase, when multiplied by the penalty coefficients, form the penalty term; however, some modifications are performed in formulation. First, for each member, the smallest available section area $\hat{A}_{ji} \geq A_{ji}$ is sought such that it satisfies all member-based constraints (slenderness, buckling and stress), if the axial force does not change. If the largest section in $\mathbb{A}$ cannot satisfy these constraints, a virtual $\hat{A}_{ji}$ that satisfies these constraint is determined whose radius of gyration is equal to the radius of gyration of the largest section area in $\mathbb{A}$. $\hat{A}_{ji} - A_{ji}$ is the estimated required increase in the current value of A_{ji} so that the i-th member satisfies all member-based constraints.

To compute the required increase for satisfaction of displacement constraints, cross-sectional areas of all members are increased proportionally:

$$\check{A}_{ji} = A_{ji} \max_{k,l} \left\{ \left| \frac{u_{jkl}}{u_k^{all}} \right| \right\}, \qquad k = 1, 2, \dots, DN_n, l = 1, 2, \dots, N_l \tag{40}$$

According to this equation, if $\check{A}_{ji}$'s replace A_{ji}'s, all displacement constraints will be satisfied. The penalized objective function is calculated considering the estimated required increase in the structure weight as follows:

$$f(\boldsymbol{\theta}_j) = \sum_{i=1}^{N_m} \rho_i M_i A_{ji} l_{ji} + \sum_{i=1}^{N_m} \rho_i l_{ji} \max\{P_i(\hat{A}_{ji} - A_{ji}), (\check{A}_{ji} - A_{ji})\}, \quad (41)$$

where $P_i \geq 1$ is the penalty coefficient for the i-th member (equivalent to c_{Pi} coefficients in FSD-ES), applied to member-based constraint violations only, since the required increase of $\hat{A}_{ji}-A_{ji}$ is computed assuming the member force does not change when the section is modified. If a displacement constraint is violated, all cross-sectional areas are proportionally increased, which does not affect member forces even in indeterminate structures, and thus, the penalty coefficients are not applied to the $\check{A}_{ji}-A_{ji}$ term. P_i's are iteratively adapted at the end of each iteration, as it will be explained in Section 3.5. For the first iteration, P_i=1, i=1, 2, …, N_m.

6.1.4. Resizing

In the resizing step, a new design ($\boldsymbol{\theta}_{j+\lambda}$) is generated by resizing the members of $\boldsymbol{\theta}_j$ utilizing the assumptions of FSD. $\boldsymbol{\theta}_{j+\lambda}$ can be interpreted as a near optimally sized structure whose shape and topology is prescribed by $\boldsymbol{\theta}_j$. Member forces and coefficients c_{ijl}'s in Equation (6) are known from evaluation of $\boldsymbol{\theta}_j$, which are assumed to be independent of member cross-sectional areas. Resizing is performed only if $\boldsymbol{\theta}_j$ is kinematically stable, otherwise we simply set $\boldsymbol{\theta}_{j+\lambda} = \boldsymbol{\theta}_j$. In FSD-ES-II, resizing is performed in two steps. In the first step, members are resized for stress constraints and in the second step, the resized members are resized again so that all displacement constraints are satisfied.

6.1.4.1. Stress-Based Resizing

For each member, a new cross-section is assigned such that the member satisfies all member-based constraints. Search for an acceptable section is performed in a limited range around the current value:

$$\acute{A}_{ji}^{\mathrm{d}} = RoundUpA\left(\frac{A_{ji}}{1+(c_A-1)\exp(1-P_i)}\right)$$

$$\acute{A}_{ji}^{\mathrm{u}} = RoundUpA\left(c_A A_{ji}\right), \tag{42}$$

in which $[\acute{A}_{ji}^{\mathrm{d}}, \acute{A}_{ji}^{\mathrm{u}}]$ is the range in which the search for the new size is performed. In this equation

- function *RoundUpA* rounds the argument to the nearest available section in $\mathbb{A}$
- P_i, the penalty coefficients, may shrink the lower bound of the range (similar to the penalty coefficient in FSD-ES). When P_i has the lowest value of 1, the ratio of maximum allowable increase and decrease are similar.
- c_A controls the maximum possible increase and decrease in A_{ji}, the so-called move limit [2]. As it explained in Section 3.1, without controlling this range, the uncertainty in assumptions of FSD can diverge the resizing process or at least reduce its contribution. This parameter is adapted iteratively based on the success of resizing in reduction of the objective function, which will be explained in Section 6.1.5.4. $c_A^{\mathrm{d}} \leq c_A \leq c_A^{\mathrm{u}}$, where c_A^{d} is the maximum of the ratio of areas of two successive sections in $\mathbb{A}$ and c_A^{u} is the square root of the ratio of the largest to the smallest areas in $\mathbb{A}$. For the first iteration, $c_A = c_A^{\mathrm{u}}$.

Having defined the search range, the new size values, $\acute{A}_{ji}$'s are calculated as follows:

$$\acute{A}_{ji} = FindAccSec\left([F_{i1}, F_{i2}, \dots, F_{iN_l}]; \acute{A}_{ji}^{\mathrm{d}}; \acute{A}_{ji}^{\mathrm{u}}\right), \tag{43}$$

where function *FindAccSec* finds the smallest acceptable cross-sectional area ($\acute{A}_{ji}, \acute{A}_{ji}^{\mathrm{d}} \leq \acute{A}_{ji} \leq \acute{A}_{ji}^{\mathrm{u}}$) such that it satisfies all member-based constraints in all load cases. F_{il}'s are member forces in different load cases. If such a section cannot be found in the provided range, the function simply selects $\acute{A}_{ji} = A_{ji}^{\mathrm{u}}$.

6.1.4.2. Displacement-Based Resizing

$\acute{A}_{ji}$'s determined from the stress-based resizing presumably satisfies all member-based constraints; however, the areas of some members should be increased to satisfy all displacement constraints as well. At this step, only increasing the cross-sectional area is allowed so that further resizing do not deteriorate member-based constraints.

The complicated problem of finding out which member areas and how much to increase is handled by repeatedly solving a simple problem: At each iteration, the most critical displacement constraint (u_c) is selected and cost effectiveness (see Equation (7)) of all members, when reduction of u_c is pursued, is computed. Combining Equations (6) and (7) leads to:

$$CE_{ji} = \left(\frac{\partial w}{\partial \acute{A}_{ji}} \Big/ \frac{\partial u_c}{\partial \acute{A}_{ji}}\right) = -\frac{\rho L \acute{A}_{ji}^2}{c_{jil}} \times \mathrm{sgn}\left(\sum_{i=1}^{m} \frac{c_{ikl}}{\acute{A}_{ji}}\right), i = 1,2,\dots,N_m \qquad (44)$$

The most efficient way to reduce u_0 is to increase the area of the cross-section with minimum CE_{ji}. It is remarkable that the value of CE_{ji} increases when the corresponding cross-sectional area is increased, and hence, the computed value of CE_{ji} should be updated whenever a small change is applied to $\acute{A}_{ji}$. This means that changes to $\acute{A}_{ji}$ should be small and after each change, the optimum member, the one with the minimum CE_{ji} should be selected again which may or may not be the previously selected member. Instead of the computationally expensive process of finding the member with minimum CE_{ji} over and over, a target cost effectiveness ($CE^{\mathrm{T}} \in [CE_0^{\mathrm{L}}, CE_0^{\mathrm{U}}]$) is found such that if all cross-sectional areas with $CE_{ji} < CE^{\mathrm{T}}$ are increased to have $CE_{ji} = CE^{\mathrm{T}}$, then u_c decreases to $\acute{u}_c$. $\acute{u}_c = \max\{u_c^{\mathrm{all}}, u_c/r_{\mathrm{uc}}\}$, where $r_{\mathrm{uc}}>1$ specifics the desired reduction in u_c. A greater value of this parameter reduces the required computation by applying a greater change to member sections; however, the quality of the resized structure may reduce. By default, $r_{\mathrm{uc}}=1.05$.

The search for CE^{T} is performed in the range $\left[CE_0^{\mathrm{L}}, CE_0^{\mathrm{U}}\right]$ using standard bisection method for 10 iterations, where

$$\left[CE_0^{\mathrm{L}}, CE_0^{\mathrm{U}}\right] = \left[\min_i\{CE_{ji}\}, 100\times \min_i\{CE_{ji}\}\right] \tag{45}$$

The coefficient of 100 limits the maximum variation of a cross-sectional area in one step. This is important because modification of a cross-sectional area affects all nodal displacements in all load cases, while in each step, CE^{T} is searched with respect to a specific displacement constraint in a specific load case. Furthermore, since each iteration of bisection method halves the search range, the coefficient of 100 ensures that CE^{T} will be found with accuracy of about 10% of the smallest cost effectiveness. Considering that $CE_{ji} \propto A_{ji}^2$, this means the maximum error in proper modification of a cross-sectional area, as will be performed in the next stage, is limited to about 5% of the current section area.

After finding CE^{T}, the new cross-sectional areas are computed form Equation (44) as follows:

$$\acute{A}_{ji} \longleftarrow \begin{cases} \left|\dfrac{c_{jil}\times CE^T}{\rho_i l_i}\right|^{0.5}, & \text{if } \ CE_{ji} < CE^T, \\ \acute{A}_{ji}, & \text{otherwise.} \end{cases} \tag{46}$$

The structure with the new value of $\acute{A}_{ji}$ computed from the baove equation reduces u_{c} to $\acute{u}_{\mathrm{c}}$, however, $\acute{A}_{ji}$'s are continuous values. The closest upper ($\acute{A}_{ji}^{+}$) and lower ($\acute{A}_{ji}^{-}$) cross-sectional areas in $\mathbb{A}$ are found. Some $\acute{A}_{ji}$'s should be rounded to $\acute{A}_{ji}^{+}$'s to satisfy the desired change in u_{c}, while the rest should be rounded to $\acute{A}_{ji}^{-}$ to avoid unnecessary weight increase. The proper number (N_0) of sections to round to the upper values are sought so that in the resized design, u_{c} is reduced by the

desired amount. These N_0 sections are selected based on their priority, which is the relative closeness to $\acute{A}_{ji}^{+}$:

$$\text{Priority}\left(\acute{A}_{ji}\right) = \frac{\acute{A}_{ji} - \acute{A}_{ji}^{-}}{\acute{A}_{ji}^{+} - \acute{A}_{ji}^{-}} \tag{47}$$

After rounding, $\acute{A}_{ji} \in \mathbb{A}$ and presumably $u_c \leq \acute{u}_c$. All u_{kl}'s are updated using Equation (7) when member sections are $\acute{A}_{ji}$. The most critical displacement constraint is detected again and reduced. This process continues until all displacement constraints are satisfied or the most critical displacement cannot be reduced anymore. The final $\acute{A}_{ji}$ specifies the sections of the new design ($A_{j+\lambda,i} = \acute{A}_{ji}, i = 1,2, \dots, N_m$). The new design, $\boldsymbol{\theta}_{j+\lambda}$=$\{\boldsymbol{M}_j, \boldsymbol{X}_j, \boldsymbol{A}_{j+\lambda}\}$ is analyzed and evaluated.

6.1.5. Updating Parameters

In addition to the strategy and design parameters of the evolution strategy, the penalty coefficients and the learning rates of $\boldsymbol{d}$ and σ_{mean} are updated at the end of each iteration after sorting all 2λ designs based on their objective function values.

6.1.5.1. Updating Mutation Parameters

The mutation parameters are the global step size, σ_{mean}, and coordinate-wise scaling factors, $\boldsymbol{d}$=[$\boldsymbol{d}^{\text{M}}$, $\boldsymbol{d}^{\text{X}}$, $\boldsymbol{d}^{\text{A}}$]. The μ-best designs are selected to update σ_{mean} and $\boldsymbol{d}$. σ_{mean} is updated as follows:

$$\sigma_{\text{mean}} = \prod_{j=1}^{\mu}\left(\left(\sigma_j\right)^{w_j}\right), \tag{48}$$

which is similar to the corresponding step in the original CMSA-ES [56], except that logarithmically decreasing weight (w_j's) are used. $\boldsymbol{d}$ is updated as follows:

$$\boldsymbol{d}^{X} \leftarrow \left(\left(1 - \frac{1}{\tau_c} \sum_{j=1}^{\mu} w_j \boldsymbol{\varpi}_j^{X} \right) \otimes (\boldsymbol{d}^{X} \otimes \boldsymbol{d}^{X}) + \frac{1}{\tau_c} \sum_{j=1}^{\mu} w_j \left(\boldsymbol{\varpi}_j^{X} \otimes \boldsymbol{Z}_j^{\mathrm{X}} \otimes \boldsymbol{Z}_j^{\mathrm{X}} \right) \right)^{0.5}$$

$$\boldsymbol{d}^{\mathrm{A}} \leftarrow \left(\left(1 - \frac{1}{\tau_c} \sum_{j=1}^{\mu} w_j \boldsymbol{\varpi}_j^{A} \right) \otimes (\boldsymbol{d}^{A} \otimes \boldsymbol{d}^{A}) + \frac{1}{\tau_c} \sum_{j=1}^{\mu} w_j \left(\boldsymbol{\varpi}_j^{A} \otimes \boldsymbol{Z}_j^{\mathrm{A}} \otimes \boldsymbol{Z}_j^{\mathrm{A}} \right) \right)^{0.5} \quad (49)$$

where $\boldsymbol{Z}_j = \{\boldsymbol{Z}_j^{\mathrm{M}}, \boldsymbol{Z}_j^{\mathrm{X}}, \boldsymbol{Z}_j^{\mathrm{A}}\} = (\boldsymbol{\theta}_j - \boldsymbol{\theta}_{\mathrm{mean}})/\sigma_j$ and $\boldsymbol{\varpi}_j^{X}$ and $\boldsymbol{\varpi}_j^{A}$ are employed to exclude the effect of passive variables in the update process.

As it was explained in Section 5.1.8, distribution of topology variables is not isotropic. In FSD-ES, a compensation for this anisotropy was considered for topology variables by updating the recombinant based on the difference between population and the selected parents. If FSD-ES II, compensation for this anisotropy is performed for both design variables and step sizes of topology variables, because the perturbation vector, $\boldsymbol{Z}_j$, is calculated differently. Accordingly:

$$\boldsymbol{d}^{\mathrm{M}} \leftarrow \left(\boldsymbol{d}^{\mathrm{M}} \otimes \boldsymbol{d}^{\mathrm{M}} + \frac{1}{\tau_c} \left(\sum_{j=1}^{\mu} w_j \left(\boldsymbol{Z}_j^{\mathrm{M}} \otimes \boldsymbol{Z}_j^{\mathrm{M}} \right) - \frac{1}{2\lambda} \sum_{j=1}^{2\lambda} \left(\boldsymbol{Z}_j^{\mathrm{M}} \otimes \boldsymbol{Z}_j^{\mathrm{M}} \right) \right) \right)^{0.5} \quad (50)$$

6.1.5.2. Updating the Recombinant Design

Shape and size variables of the recombinant design are updated by computing the weighted average of the parental population, however, the effects of passive members and nodes are excluded:

$$\boldsymbol{X}_{\text{mean}} \leftarrow \sum_{j=1}^{\mu} w_j \left(\left(\mathbf{1} - \boldsymbol{\varpi}_j^X \right) \otimes \boldsymbol{X}_{\text{mean}} + \boldsymbol{\varpi}_j^X \otimes \boldsymbol{X}_j \right)$$

$$\boldsymbol{A}_{\text{mean}} \leftarrow \sum_{j=1}^{\mu} w_j \left(\left(\mathbf{1} - \boldsymbol{\varpi}_j^A \right) \otimes \boldsymbol{A}_{\text{mean}} + \boldsymbol{\varpi}_j^A \otimes \boldsymbol{A}_j \right) \tag{51}$$

Topology variables are updated based on difference between their distribution in the offspring and parental populations, to compensate for anisotropy in distribution of topology variables. The proposed relation in FSD-ES-II is as follows:

$$\boldsymbol{M}_{\text{mean}} \leftarrow \boldsymbol{M}_{\text{mean}} + \left(\sum_{j=1}^{\mu} w_j \boldsymbol{M}_j - \frac{1}{2\lambda} \sum_{j=1}^{2\lambda} \boldsymbol{M}_j \right), \tag{52}$$

which is very similar to that of the FSD-ES.

6.1.5.3. Updating the Penalty Coefficients

Adaptation of P_i's is performed based on the fraction of designs in the parental population in which the *i*-th member is responsible for a constraint violation. If this fraction is greater than 0.5, the corresponding penalty coefficient is increased. If a displacement constraint is violated, all members are assumed to be responsible for that. Accordingly, p_{ij} measures responsibility of the *i*-th member in possible constraint violation of the *j*-th design as follows:

- p_{ij}=0, if $\boldsymbol{\theta}_j$ is kinematically stable, the *i*-th member is active in the design, the design satisfies all displacement constraints and the *i*-th member does not violate a stress, slenderness or buckling constraint.

- p_{ij}=1, if $\boldsymbol{\theta}_j$ is kinematically stable, the *i*-th member is active in the design, but the design violates a displacement constraints or the *i*-th member violates a stress, slenderness or buckling constraint.
- p_{ij}=0.5, if $\boldsymbol{\theta}_j$ is kinematically unstable or the *i*-th member is passive in the design.

The value of 0.5 is a neutral value which neither increases nor decreases a penalty term. The weighted average of p_{ij}'s shows whether a penalty term should be increased or decreased:

$$\psi_i^{\text{old}} \leftarrow \psi_i^{\text{new}}$$

$$\psi_i^{\text{new}} \leftarrow \sum_{j=1}^{\mu} w_j \, p_{ji},$$

$$P_i \leftarrow \begin{cases} P_{i,} & \text{if } 0.5 < \psi_i^{\text{new}} < \psi_i^{\text{old}} \\ P_i \exp\left(\tau_P(\psi_i^{\text{new}} - 0.5)\right), & \text{otherwise.} \end{cases} \tag{53}$$

where τ_P specifies the learning rate for the penalty coefficients. As it can be observed, only parental population is considered. By default, $\tau_P=\tau^{0.5}$. ψ_i^{new}>0.5 refers that most of the times the *i*-th member has violated a constraint, and thus the corresponding penalty term should be intensified. It may take the algorithm several iterations to reach regions where $\psi_i^{\text{new}} < 0.5$. To give the algorithm the sufficient time, the corresponding penalty term is not intensified if $\psi_i^{\text{new}} < \psi_i^{\text{old}}$, which demonstrates some progress in moving towards the feasible regions was made.

6.1.5.4. Controlling the Move Limit

FSD assumes that the axial force remains constant when a cross-section is changed. The error caused by this assumption increases when the amount of change increases, therefore, the resized solution might be even worse than the original one. The alternative is to reduce the possible amount of change in the cross-sectional area, the so-called move limit [2], which is controlled by c_A, so

that the error caused by the FSD assumptions is reduced. The adaption of c_A is based on the resizing success, which is proportional to the fraction of times in which the resized solution ($\boldsymbol{\theta}_{j+\lambda}$) is better than the original one ($\boldsymbol{\theta}_j$):

$$R_{\text{eff}} = \frac{1}{\lambda}\sum_{j=1}^{\lambda} \text{sgn}\left(1 - \frac{f\left(\boldsymbol{\theta}_{j+\lambda}\right)}{f\left(\boldsymbol{\theta}_j\right)}\right)$$

$$c_A \leftarrow \max\left\{c_A^{\text{d}}, \min\left\{\text{c}_\text{A}^{\text{u}}, c_A \times \exp\left(R_{\text{eff}} \times \sqrt{\tau}\right)\right\}\right\}, \tag{54}$$

where $-1 \leq R_{\text{eff}} \leq 1$ measures the resizing efficiency. Equation 26 indicates that c_A should increase if the resizing efficiency is high and vice versa.

6.1.6. Stopping Criteria and Parameter Tuning

All parameters of the FSD part are set to their default values. In evolution strategies, τ, τ_c, μ and λ are selected based on the number of design parameters and population size for continuous unconstrained optimization. Our preliminary parameter study, in accordance with previous version of FSD-ES revealed that for highly constrained mixed-variable TSS optimization problems, the types of variables and the number of constraints should also be considered in computing the effective number design parameters. The proposed relation to compute the effective number of variables (N_{eff}) is as follows:

$$N_{\mathrm{VAR}} = \left(\sqrt{N_{\mathrm{top}}} + \sqrt{N_{\mathrm{shape}}} + \sqrt{N_{\mathrm{size}}}\right)^2$$

$$N_{\mathrm{CON}} = \sqrt{N_l} \times \left(\sqrt{N_u} + \sqrt{N_m}\right)^2$$

$$N_{\mathrm{eff}} = (N_{\mathrm{VAR}}) \left(1 + \frac{N_{\mathrm{CON}}}{N_{\mathrm{VAR}}}\right)^{0.5} \tag{55}$$

where N_{u} is the number of displacement constraints per load case. N_{VAR} and N_{CON} integrate the effect of the number of design variables and constraints, respectively. N_{eff} is the effective number of design parameters, which is greater than $N_{\mathrm{top}}+N_{\mathrm{shape}}+N_{\mathrm{size}}$, because:

- The diversity in the type of the design variables makes the problem harder. A problem with $N_{\mathrm{top}}=N_{\mathrm{shape}}=N_{\mathrm{size}}=20$ is reasonably harder than a problem with 60 variables of a single type. Therefore, Equation (55) calculates a greater value for the former ($N_{\mathrm{VAR}}=180$) than the latter ($N_{\mathrm{VAR}}=60$) case.
- Presence of many constraints and activation of many of them in the optimum design make the TSS problem harder than most unconstrained test problems conventionally employed for empirical evaluation of ESs. N_{CON} integrates the effect of the number of constraints. Similarly, presence of different types (member-based or node-based), in addition to the number of constraints, is assumed to affect the problem complexity. Although the number of constraints is proportional to N_l, the square root of N_l was used in Equation (55) since usually a few of the load cases is critical for each member or node.
- The effect of a fixed N_{CON} on problem complexity is assumed to be less for a complicated problem with a great N_{VAR} than a problem with a small N_{VAR}. Consequently, N_{eff} is affected

by N_{CON}/N_{VAR}. The square root was empirically found to be useful to moderate the effect of N_{CON}/N_{VAR} on N_{eff}.

The parameters of the ES part are then set a follows:

$$\tau = \frac{1}{\sqrt{2N_{\text{eff}}}}, \tau_c = \frac{\mu + N_{\text{eff}} \times (N_{\text{eff}} + 1)}{\mu}$$
$$\lambda = 2\sqrt{N_{\text{eff}}}, \mu = \lfloor 0.3\lambda \rfloor, MaxIter = 75\sqrt{N_{\text{eff}}}. \quad (56)$$

τ and τ_c are calculated following the standard CMSA-ES code. In ESs, the optimal population size is problem dependent, but it grows sub-proportionally with respect to the number of variables. For example, the minimum recommended value for CMA-ES grows logarithmically with the problem dimension. Accordingly, the default population size is selected proportional to the square root of N_{eff}. The recommended value for *MaxIter* was determined based on some preliminary parameter study. It is a conservatively large value; however, based on the employed performance measure, the performance is measured by the consumed evaluation budget to reach the desired target weight instead of the allocated budget.

6.1.7. Flowchart of the Proposed Algorithm

The flowchart of FSD-ES II is provided in Figure 21.

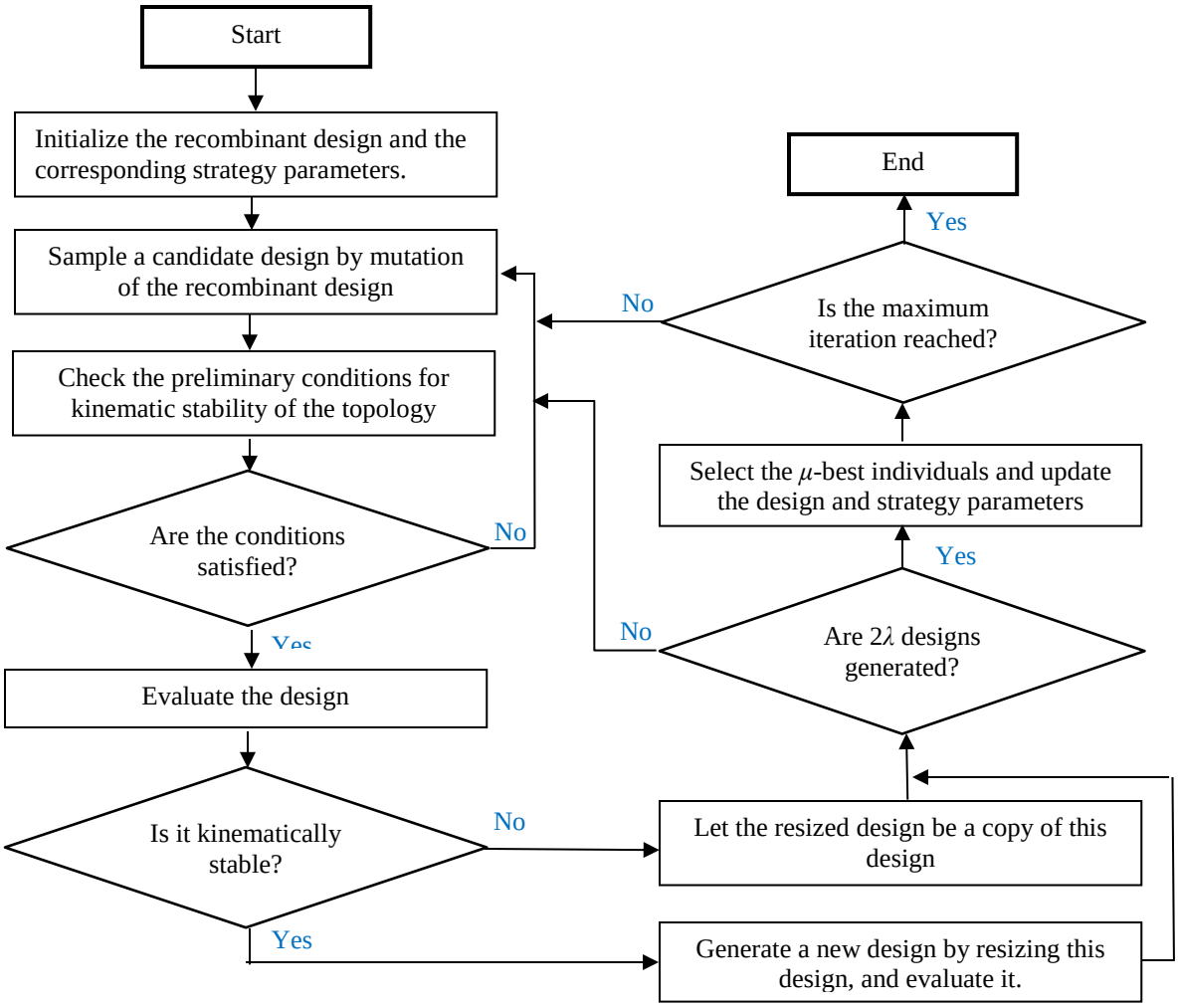


Figure 21. Flowchart of FSD-ES II

6.2. Numerical Evaluation of FSD-ES II[8]

In this section, performance of FSD-ES II is numerically evaluated and compared with the available methods in the literature.

[8] The source code of FSD-ES II (in MATLAB) and the reported solutions are provided as supplementary files for this dissertation. For the latest source codes of the developed methods in this dissertation, please visit the author's ResearchGate page: https://www.researchgate.net/profile/Ali_Ahrari/contributions

6.2.1. Test Problems

To analyze and compare FSD-ES-II with the best TSS optimization methods, a test suite consisting of complicated truss problems is formed. Simple test problems that are commonly employed in most available studies are overlooked. Some of the test problems are introduced in this study, mainly by converting an available shape and size optimization problem to a TSS optimization problem. Two goals are pursued by this modification: First, to measure the potential extra saving when a more flexible and intricate ground structure is used and second, whether employing an optimization method can detect such a potential gain, if any.

It is notable that some of the structures in the employed test problems are modular. For TSS optimization of modular trusses, the specialized formulation proposed in [127] can lead to better results than the conventional concept of the ground structure, nevertheless, in this study, the concept of the ground structure is employed for all problems to have a unified formulation method. For readability, only nodes are numbered in the ground structure, and members are determined by their end nodes. For example, $A_{1\text{-}2}$ denotes the cross-sectional area of the member connecting node 1 to node 2.

6.2.1.1. 47-Bar Transmission Tower

The first problem is the 47-bar transmission tower truss problem, commonly employed as a shape and size optimization in many previous studies [127, 102, 128] including Section 4.2.1 of this dissertation, but rarely as TSS optimization problem [102]. The TSS version of this problem is solved in this section by FSD-ES-II. Symmetry about x=0 is imposed which reduces the number of shape and size variables to 17 and 27 respectively. The search range for shape variables were not explicitly reported in [102], and thus a quite large range, with respect to the original configuration of the nodes in the depicted ground structure (Figure 22) is considered in this study

(Table 12). The ground structure has limited flexibility for topology optimization, since only 7 members ($A_{3\text{-}4}$, $A_{5\text{-}6}$, $A_{7\text{-}8}$, $A_{9\text{-}10}$, $A_{11\text{-}12}$, $A_{13\text{-}14}$, $A_{21\text{-}22}$) might be eliminated without kinematic instability of the structure, however, this information was not given to the method in [102]. For fair comparison, the same procedure is followed in this study and thus 27 topology variables are considered for this problem. Nodes 15, 16, 17 and 18 cannot move, and nodes 1 and 2 can move horizontally only.

Table 12. Simulation Data for the 47-bar truss problem

Design Variables	**Shape (17)**	$x_1, x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{19}, x_{21}, y_3, y_5, y_7, y_9, y_{11}, y_{13}, y_{19}, y_{21}$		
	Size (27)	27 size variables for 27 independent members, cross section of other members is dependent and determined using symmetry about x=0.		
	Topology (27)	One topology variable per size variable		
Constraints	**Stress**	$(\sigma_c)_i \le 103.42$ MPa (15 ksi); $(\sigma_c)_i \le 3.96EA_i/l_i^2$; $(\sigma_t)_i \le 137.90$ MPa (20 ksi), i=1, 2, …, 47		
	Displacement	None		
Search Range	**Shape Variables**	$-180'' \le x_1, x_3, x_5, x_7 \le -10''$; $-90'' \le x_9, x_{11}, x_{13}, x_{21} \le -10''$; $-150'' \le x_{19} \le -10''$; $120\times(i-1)'' \le y_{2i+1} \le 120\times(i+1)''$, i=1, 2, 3; $60\times(i+1)'' \le y_{2i+1} \le 60\times(i+3)''$, i=4, 5, 6; $560'' \le y_{19}, y_{21} \le 660''$;		
	Size Variables	$A \in \mathbb{A}$, $\mathbb{A}$ ={0.1, 0.2, 0.3, …, 4.9, 5} (in²)		
Loading		Nodes	F_x	F_y
	Case I	17,18	26.689 KN (6 kips)	-62.275 KN (-14 kips)
	Case II	17	26.689 KN (6 kips)	-62.275 KN (-14 kips)
	Case III	18	26.689 KN (6 kips)	-62.275 KN (-14 kips)
Mechanical Properties		Modulus of elasticity: E=206.84 GPa (3.0×10⁴ ksi) Density of the material: ρ=8304.0 Kg/m³ (0.3 lb/in.³);		

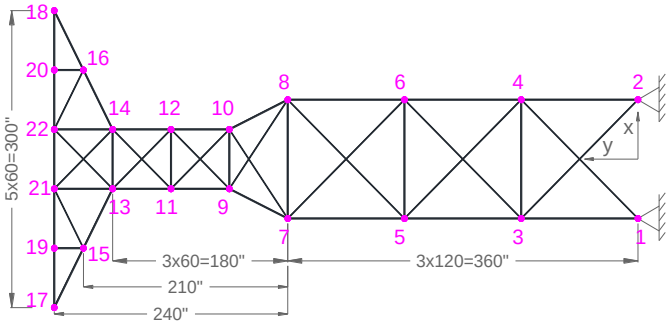
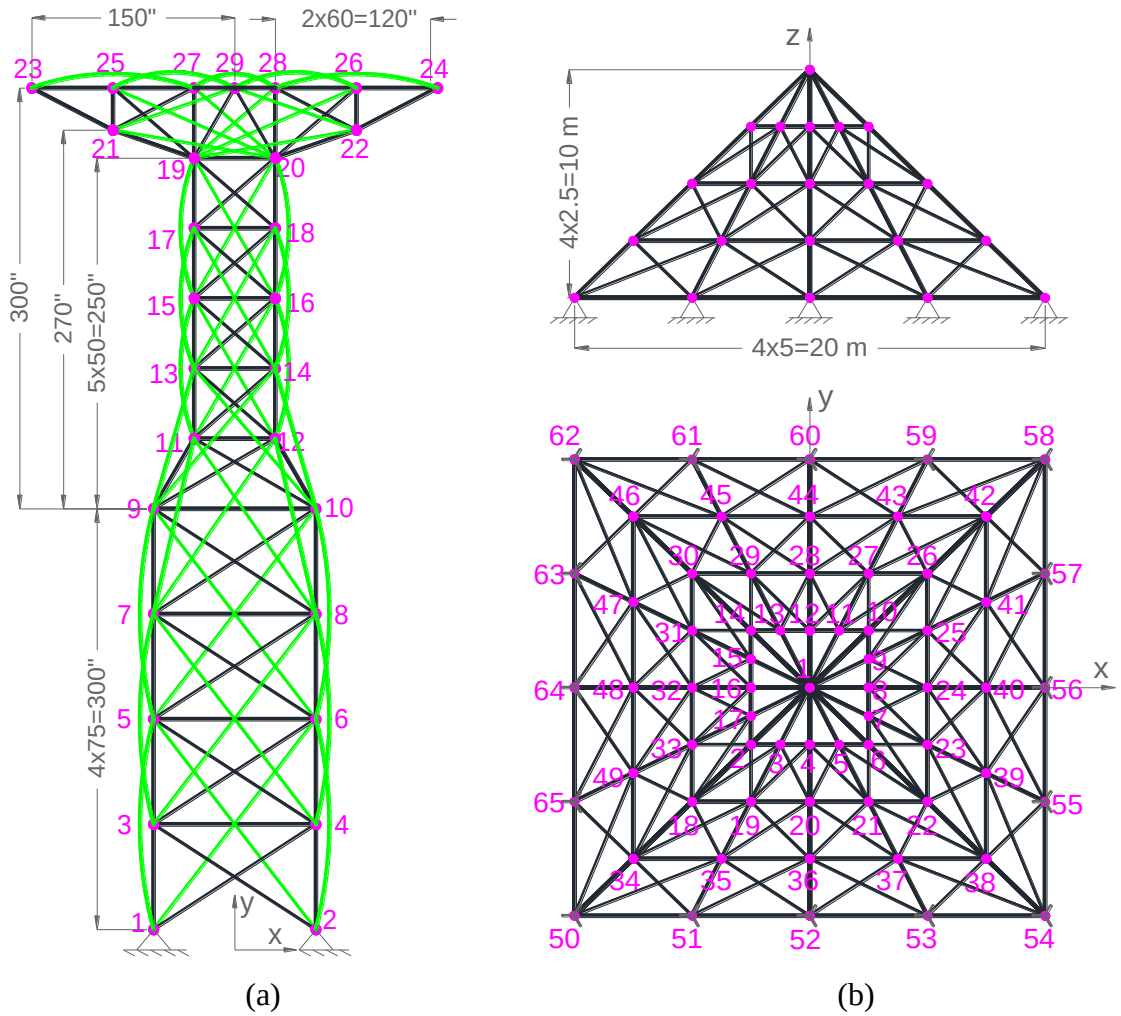


Figure 22. Ground structure for the 47-bar transmission tower problem



(a) (b)

Figure 23. Ground structure for the a) 110-bar and b) 224-bar (front and top view) test problems

6.2.1.2. 68-Bar Truss Problem

The 68-bar test problem introduced in Section 5.2.1 is solved with FSD-ES II.

6.2.1.3. 110-Bar Transmission Tower

The 47-bar transmission tower problem is revisited with a more intricate ground structure to have more flexibility for topology optimization. The alternative ground structure with 110-bar (Figure 23(a)) allows for elimination of redundant nodes as well. Nodes 21, 22, 23 and 24 cannot move and symmetry about x=0 is imposed. Presence of no members seems necessary for kinematic

stability of the structure, and thus a topology variable per independent member is allotted, resulting in 60, 24 and 60 topology, shape and size variables respectively. Data required for simulation of this problem are provided in Table 13.

Table 13. Simulation Data for the 110-bar truss problem

Design Variables	**Shape (24)**	$x_1, x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{15}, x_{17}, x_{19}, x_{25}, x_{27}$ $y_3, y_5, y_7, y_9, y_{11}, y_{13}, y_{15}, y_{17}, y_{19}, y_{25}, y_{27}, y_{29}$		
	Size (60)	60 size variables for 60 independent members, cross-sections of other members is dependent and determined using symmetry about x=0.		
	Topology (60)	One topology variable per size variable.		
Constraints (Variant I)	**Stress**	$(\sigma_c)_i \leq$103.42 MPa (15 ksi); $(\sigma_t)_i \leq$137.90 MPa (20 ksi), $(\sigma_c)_i \leq 3.96EA/l_i^2$		
	Displacement	No displacement constraint		
Search Range	**Shape Variables**	$-180'' \leq x_1, x_3, x_5, x_7, x_9 \leq 0''$; $-90'' \leq x_{11}, x_{13}, x_{15}, x_{17}, x_{19}, x_{27} \leq 0''$; $-150'' \leq x_{25} \leq 0''$; $75\times(i-1)'' \leq y_{2i+1} \leq 75\times(i+1)''$, i=1, 2, 3, 4; $50\times(i+1)'' \leq y_{2i+1} \leq 50\times(i+3)''$, i=5, 6, 7, 8, 9; $550'' \leq y_{25}, y_{27}, y_{29} \leq 650''$;		
	Size Variables	$A \in \mathbb{A}$, $\mathbb{A}$ ={0.1, 0.2, 0.3, …, 4.9, 5} (in²)		
Loading		Nodes	F_x	F_y
	Case I	23, 24	26.689 KN (6 kips)	-62.275 KN (-14 kips)
	Case II	23	26.689 KN (6 kips)	-62.275 KN (-14 kips)
	Case III	24	26.689 KN (6 kips)	-62.275 KN (-14 kips)
Mechanical Properties		Modulus of elasticity: E=206.84 GPa (3.0×10^4 ksi); Density of the material: ρ=8304.0 Kg/m³ (0.3 lb/in³);		

6.2.1.4. 224-Bar Pyramid

The ground structure for the 224-bar pyramid test problem is depicted in Figure 23(b), which is directly adopted from [96, 102]. There are four planes of symmetry (xy=0, $y\pm x$=0) and besides, for structural aesthetics, all nodes that have similar z in the ground structure must always have similar z. Nodes 1, 52, 56, 60 and 64 are basic nodes and cannot move in any direction. Other nodes, including the supports, can be eliminated if necessary. Despite the large number of members, the number of design variables is moderate. This problem has scarcely been used as a TSS optimization problem, even though it is more challenging and realistic than most conventional yet simple TSS problems. The search range of the shape variables and the material density were not

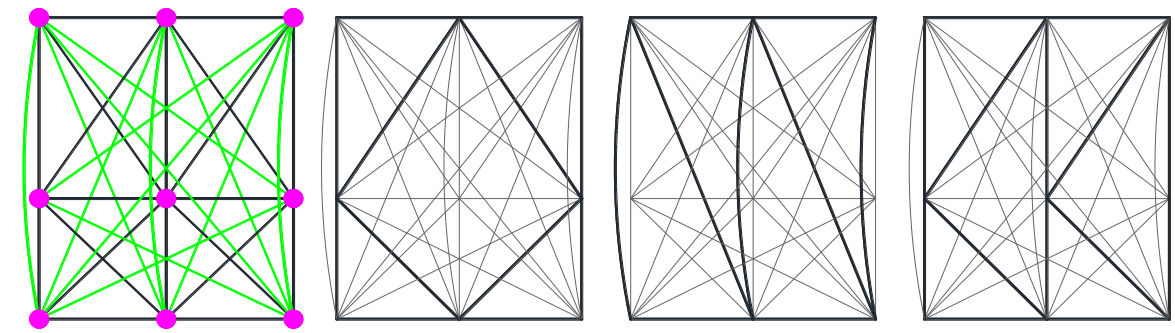
explicitly mentioned in the referenced studies; therefore, a relatively large range is selected in this dissertation considering relative distance of nodes in the ground structure. The material density is set to 7,850 Kg/m³, a commonly used value for structural steel that also matches the results provided in [102]. Simulation data for this problem are provided in Table 14.

Table 14. Simulation data for the 224-bar pyramid

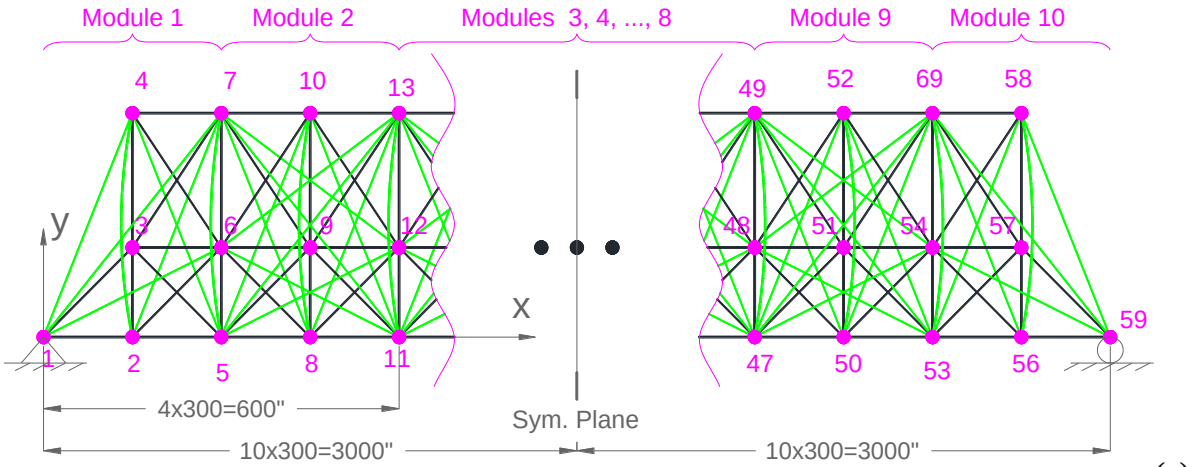
Design Variables	**Shape (18)**	x_2, x_3, y_3, y_4, x_{18}, x_{19}, y_{19}, y_{20}, x_{34}, x_{35}, y_{35}, y_{36}, x_{50}, x_{51}, y_{51}, z_2, z_{18}, z_{34}			
	Size (32)	$A_{1\text{-}2}$, $A_{1\text{-}3}$, $A_{1\text{-}4}$, $A_{2\text{-}3}$ $A_{3\text{-}4}$, $A_{2\text{-}18}$, $A_{2\text{-}19}$, $A_{3\text{-}18}$, $A_{3\text{-}19}$, $A_{3\text{-}20}$, $A_{4\text{-}19}$, $A_{4\text{-}20}$, $A_{18\text{-}19}$, $A_{19\text{-}20}$, $A_{19\text{-}34}$, $A_{18\text{-}35}$, $A_{19\text{-}34}$, $A_{19\text{-}35}$, $A_{19\text{-}36}$, $A_{20\text{-}35}$, $A_{20\text{-}36}$, $A_{34\text{-}35}$, $A_{35\text{-}36}$, $A_{34\text{-}50}$, $A_{34\text{-}51}$, $A_{35\text{-}50}$, $A_{35\text{-}51}$, $A_{35\text{-}52}$, $A_{36\text{-}51}$, $A_{36\text{-}52}$, $A_{50\text{-}51}$, $A_{51\text{-}52}$			
	Topology (32)	One topology variable per size variable.			
Constraints	**Stress**	AISC-ASD design specification [7] with F_y=248.21 MPa (36.0 ksi)			
	Slenderness	AISC-ASD design specification			
	Displacement	$u_k \leq 1$ cm (0.39370″), k=1, 2,…, DN_n			
Search Range	**Shape Variables**	x_2, x_3, y_3, y_4 may vary within ±1.25 m of the default value in the ground structure. x_{18}, x_{19}, y_{19}, y_{20} may vary within ±2.5 m of the default value in the ground structure. x_{34}, x_{35}, y_{35}, y_{36} may vary within ±3.75 m of the default value in the ground structure. x_{50}, x_{51}, y_{51} may vary within ±5.0 m of the default value in the ground structure. z_2, z_{18}, z_{34} may vary within ±2.5 m of the default value in the ground structure.			
	Size Variables	$A \in \mathbb{A}$, where $\mathbb{A}$ is the set of circular hollow sections in AISC-ASD			
Loading		Nodes	F_x (KN)	F_y (KN)	F_z (KN)
		1	500	500	−1000
Mechanical Properties		Modulus of elasticity: E=200 GPa (29000 ksi) Density of the material: ρ=7850 Kg/m³ (0.2836 lb/in³);			

6.2.1.5. Bridge Design Problem

In the final step, the bridge design problem which was solved for shape and size optimization in Section 4.2 is revisited. Since there are many reasonable models for the topology, choosing the optimum one based on intuition is challenging, and besides, it is possible that the optimum topology of a panel depends on its location. For example, the Parker model may be the best choice for the 2nd panel, while the Pettit model might be the optimum choice for the 3rd one.



(a) (b) (c) (d)



(e)

Figure 24. a) The proposed module for the bridge design problem. The proposed module can conform to different models such as b) Bailey c) Pratt and d) K-truss. e) The ground structure is posed by joining 10 of these modules side by side. For esthetics, some members of the first and the last modules were removed.

Table 15. Simulation Data for the 277-bar bridge design problem

Design Variables	**Shape**	**Variant I (1):** $y_4=y_7=\ldots=y_{58}=2y_3=2y_6=2y_9=\ldots=2y_{57}$ **Variant II (10):** $2y_{3i}=y_{3i+1}$, i=1, 2, …, 10 **Variant III (38):** x_{3i}, y_{3i}, x_{3i+1}, y_{3i+1}, i=1, 2, …, 9; y_{30}, y_{31}		
	Size	**All variants (140):** Cross sections of 140 members on the left side of the symmetry plane of the bridge, including those on the symmetry plane.		
	Topology	**Variants I & II (43):** $M_{1\text{-}3}$, $M_{1\text{-}4}$, $M_{1\text{-}6}$, $M_{1\text{-}7}$, $M_{2\text{-}3}$, $M_{2\text{-}4}$, $M_{2\text{-}6}$, $M_{2\text{-}7}$, $M_{3\text{-}4}$, $M_{3\text{-}5}$, $M_{3\text{-}6}$, $M_{3\text{-}7}$, $M_{4\text{-}5}$, $M_{4\text{-}6}$, $M_{4\text{-}7}$ (1st module) $M_{5\text{-}6}$, $M_{5\text{-}7}$, $M_{5\text{-}9}$, $M_{5\text{-}10}$, $M_{5\text{-}12}$, $M_{5\text{-}13}$, $M_{6\text{-}7}$, $M_{6\text{-}8}$, $M_{6\text{-}9}$, $M_{6\text{-}10}$, $M_{6\text{-}11}$, $M_{6\text{-}13}$, $M_{7\text{-}8}$, $M_{7\text{-}9}$, $M_{7\text{-}10}$, $M_{7\text{-}11}$, $M_{7\text{-}12}$, $M_{8\text{-}9}$, $M_{8\text{-}10}$, $M_{8\text{-}12}$, $M_{8\text{-}13}$, $M_{9\text{-}10}$, $M_{9\text{-}11}$, $M_{9\text{-}12}$, $M_{9\text{-}13}$, $M_{10\text{-}11}$, $M_{10\text{-}12}$, $M_{10\text{-}13}$ (2nd module) Apply overlap prevention rule **Variant III (130):** One topology variable per member on the left side of the bridge except members on the lower cord.		
Constraints	**Stress**	AISC-ASD design specification [7], with F_y=248.21 MPa (36.0 ksi)		
	Slenderness	AISC-ASD design specification		
	Displacement	$u_k \leq 25.4$ cm (10″), k=1, 2,…, DN_n		
Search Range	**Shape Variables**	$50 \leq y_{3i+1} \leq 1000''$; i=1, 2, …, 10 $300''\times(i\text{-}1) \leq x_{3i}$, $x_{3i+1} \leq 300''\times(i+1)$, i=1, 2, …, 9		
	Size Variables	$A \in \mathbb{A}$, where $\mathbb{A}$ is the set of 83 sections in W-shape profile list of AISC-ASD between W10×12 and W14×730.		
Loading		Node	F_x (kip)	F_y (kip)
		2, 5, 8, …, 56	0.0	−60.0
Mechanical Properties		Modulus of elasticity: E=200 GPa (29000 ksi) Density of the material: ρ=7850 Kg/m^3 (0.2836 lb/in.^{3});		

To reduce the burden on the designer for deciding on these important factors, this problem is revisited by using an intricate ground structure, consisting of 10 modules. Each module consists of 3×3=9 nodes and 33 members Figure 24(e). The lower cord is pre-designed [107] and cannot be changed. Adjacent modules share 3 nodes and 3 members, thus the grounds structure has 59 nodes and 277 members. The selected module can conform to most models conventionally used for bridge design, including Parker, Bailey and K-truss, and many non-standard models. Three variants of this problem are solved in this study, with distinct amount of flexibility in the design:

- **Variant I**: The 1st and the 2nd modules are independent while modules 3, 4 and 5 are topologically identical to the 2nd module. This reduces the number of topology parameters to 43. There is only one shape variable, vertical position of the upper cord. The height of

nodes on the middle cord is half of the adjacent node on the upper cord. No node may move horizontally. There is one size variable per member on the left part of the bridge.

- **Variant II**: similar to variant I, but vertical position of the nodes on the upper cord are independent of each other, however, the height of nodes on the middle cord is half of the adjacent node on the upper cord, which increases the number of shape variables to ten.
- **Variant III:** Topologies of modules are independent of each other. There is one topology variable per member on the left part of the bridge except for the members on the lower cord, which must be active. Nodes on the middle and upper cord can move in any direction independent of each other, except the nodes on x=3,000″, which can move vertically only. The overall number of design variables is 308 for this case.

In all variants, symmetry about x=3,000″ is imposed, location of nodes on the lower cord is fixed and members on the lower cord (20 members) must remain active. Each variant provides more flexibility in design optimization than the previous one. This increases the potential material saving when optimization is performed; however, since the complexity of the problem exacerbates, the achieved solution might become even heavier, especially if the extra potential saving is small in comparison with the added complexity. Furthermore, modularity of the structure disrupts in later variants. For example, in variant I, shape and topology of modules are similar, which facilitates construction and improve esthetics. In variant II, modules are topologically similar, but differ in shape and cross-sections. Finally, in variant III, the structure is not modular anymore. Whether the extra saving, if achieved, compensates for deterioration of structural esthetics and increase in construction costs depends on the amount of the saving, and is a decision which should be made by the decision maker. Later variants are however more interesting for benchmarking,

since they provide more challenging situations with larger number of design parameters, which can reliably illuminate the gap among different optimization methods.

The problem of possible overlapping members in the final design, which is assumed to be practically undesirable, is handled by imposing an overlap prevention rule: For a set of three nodes that must remain vertically aligned and are connected to one another (two short members and one long member), the long member may be active only if the two short members are passive. For example, this rule is applied to the set {2, 3, 4}. This means the long member can be active ($M_{2\text{-}4}$=1) only if the two short members are passive ($M_{2\text{-}3}$=$M_{3\text{-}4}$=0). A revision is performed to handle sampled designs that violate this rule. If a set of three members violates this rule, first a random number is generated ($r_0 \in [0,1]$) and then, the following correction is applied:

$$\begin{cases} \text{Remove all 3 members,} & \text{if} \quad r_0 \leq 0.2, \\ \text{Remove the long member,} & \text{if} \quad 0.2 \leq r_0 \leq 0.8, \\ \text{Remove the short members,} & \text{if} \quad 0.8 \leq r_0. \end{cases}$$

The probabilities are computed considering that in 8 possible combinations for absence/absence of 3 members, 5 combinations do not violate the overlapping rule. In 1 out of 5, no member is present, in 1 out of 5, only the long member is present and in 3 out of 5, at least one short member is present. This rule is applied to the 19 sets of vertically aligned nodes ({2, 3, 4}, {5, 6, 7},….,{56, 57, 58}) in variants I and II. It is notable that this undesirable feature is unlikely to happen in variant III, since nodes may move horizontally as well. The data required for simulation of this problem are presented in Table 15.

6.2.2. Performance Measures

The same performance measure (Section 4.2.2) is employed for performance evaluation. Table 16 tabulates values of the test problem data and the calculated values of the control

parameters of FSD-ES II according to the Section 6.1.6. Accordingly, FSD-ES requires no ad hoc tuning effort, since all parameters are set based on known features of the problem.

To increase reliability of statistical measures, each problem is solved 500 times independently and the best solution found is reported.

Table 16. Default parameter setting for the test problems determined using equations 7 and 8

Problem	N_{top}	N_{shape}	N_{size}	N_l	N_u	N_m	N_{VAR}	N_{CON}	N_{eff}	λ	*MaxIter*
47-bar	27	17	27	3	0	47	210.7	81.4	248.1	32	1181
68-bar	68	31	68	2	33	68	486.7	276.8	609.5	49	1852
77-bar	0	10	39	1	77	77	88.5	308.0	187.3	27	1026
110-bar	60	24	60	3	0	110	415.8	190.5	502.1	45	1681
224-bar	32	18	32	1	147	224	242.0	733.9	486.0	44	1653
277-bar (I)	43	1	140	1	115	277	376.0	749.0	650.3	51	1913
277-bar (II)	43	10	140	1	115	277	464.5	749.0	750.7	55	2055
277-bar (II)	130	38	140	1	115	277	864.3	749.0	1180.8	69	2577

6.2.3. Results and Discussion

Because of the large number of independent runs, measured values of *SR*, *FE*s and *ERT* are assumed to be reliable if $SR \geq 0.05$. Figure 25 illustrates *FEs*, *SR* and *ERT* curves as a function of W_{target} for each problem. For the 68-bar problem, *ERT* of the earlier version of FSD-ES [19] and the new version excluding explicit consideration of displacement constraints in the resizing step (FSD-ES II-{u}) are also provided. In this problem, the displacement constraint is active which may illuminate the importance of consideration of displacement constraints in the resizing step. The best solution found for each problem is illustrated in Figure 26 and the corresponding design parameters are provided in Table 17 and Table 18. The obtained results demonstrate that:

- For the 47-bar truss problem, Figure 25(a) demonstrates that the best solution found by FSD-ES-II weighs 1,728 lb, reached after about 64,000 evaluations, although *ERT* is much greater. For W_{target}=1,750 lb, *FEs*=38,800 and *SR*=0.24, sufficiently high to draw reliable conclusions. The best previous results in the literature include 1789 lb, reached after about

24,600 evaluations using an SA-based method [121], and 1,885 lb, reached at the end of 100,000 evaluations using a GA-based method [102]. The best solution of FSD-ES-II is about *3.4%* lighter than the best solution available in the literature. In comparison with the best design for shape and size optimization which weighs 1,847 lb [19], performing TSS optimization resulted in an extra *6.4%* reduction in the overall weight, which is quite significant considering the limited flexibility of the topology of the ground structure.

- For the 68-bar problem, the best solution of FSD-ES-II weighs 1,166 lb, about *3.1%* lighter than that found by FSD-ES. More importantly, Figure 25(b) demonstrates that FSD-ES-II is several times faster than FSD-ES when *ERT* of both methods are compared. It should also be highlighted that the older version underestimates the number of evaluations, as explained earlier. Considering that some displacement constraints are active in the optimized solution, this advantage probably originates from the performed improvement in the resizing step such that displacement constraints are explicitly considered. This can be verified by analyzing the *ERT* curve of FSD-ES II-{u}. For the early stage of the optimization, FSD-ES II-{u} slightly outperforms FSD-ES II, however, FSD-ES can reach better solutions at the end.
- For the 110-bar problem, the best solution of FSD-ES weighs 1,314 lb, which demonstrates by providing more flexibility in topology optimization, it is possible to save an extra *24%* in weight, when compared to the best solution of the 47-bar problem. The topology of the best solution could hardly be concluded by engineering intuition (see Figure 26(b)), even though it is only a little more complicated than the optimized solution of 47-bar problem. Figure 25(c) demonstrates that 28.6% of runs could reach W_{target}=1,450 lb, on average after 66,200 evaluations.

- The 224-bar pyramid problem has been solved in a few studies [102, 96]. The best solution in the literature weighs 4,587 Kg [96], reached by an SA-based algorithm, after about 60,000 function evaluations. The best solution of FSD-ES weighs 3,079 Kg, which is *32.8%* lighter than the best reported solution in the literature. The required number of function evaluations for the best run to reach this weight is 113,000, relatively greater than the SA-based method, however, the gap between the qualities of the best designs is huge. For W_{target}=3,400 lb, *FEs*=87,400 and *SR*=0.17 (Figure 25(d)).
- In variant I of the bridge design problem, FSD-ES-II could reach the weight of 282.0 kip, which is surprisingly lighter than the best solution found for shape and size optimization when different topologies were tried [107] including the 77-bar bridge problem optimized in this study (see Section 4.2). This demonstrates the optimized design of variant I is not only lighter, but also has less fabrication and assembly cost, due to similarity of modules (identical shape and topology versus identical topology only). It also excels in esthetics. The topology of the best solution found resembles the Bailey model to some extent (Figure 26(e)). Figure 25(e) demonstrates that 12.6% of runs could each W_{target}=282.5 kip after 125,700 evaluations. The best solution of variant II of this problem weighs 236.5 kip, *16.1%* lighter than best solution of variant I. The arch-shape of the upper cord in the best-found solution of variant II matches engineering intuition. In variant III, there is only *1.9%* achieved extra saving, in comparison with the best solution of variant II, which also comes at the cost of excessive function evaluations. For example, for W_{target}=240 kip, *ERT* in case III (Figure 25(g)) is about 12 times greater than the *ERT* in case II (Figure 25(f)). The structural esthetics and ease of assembly has degraded as well.

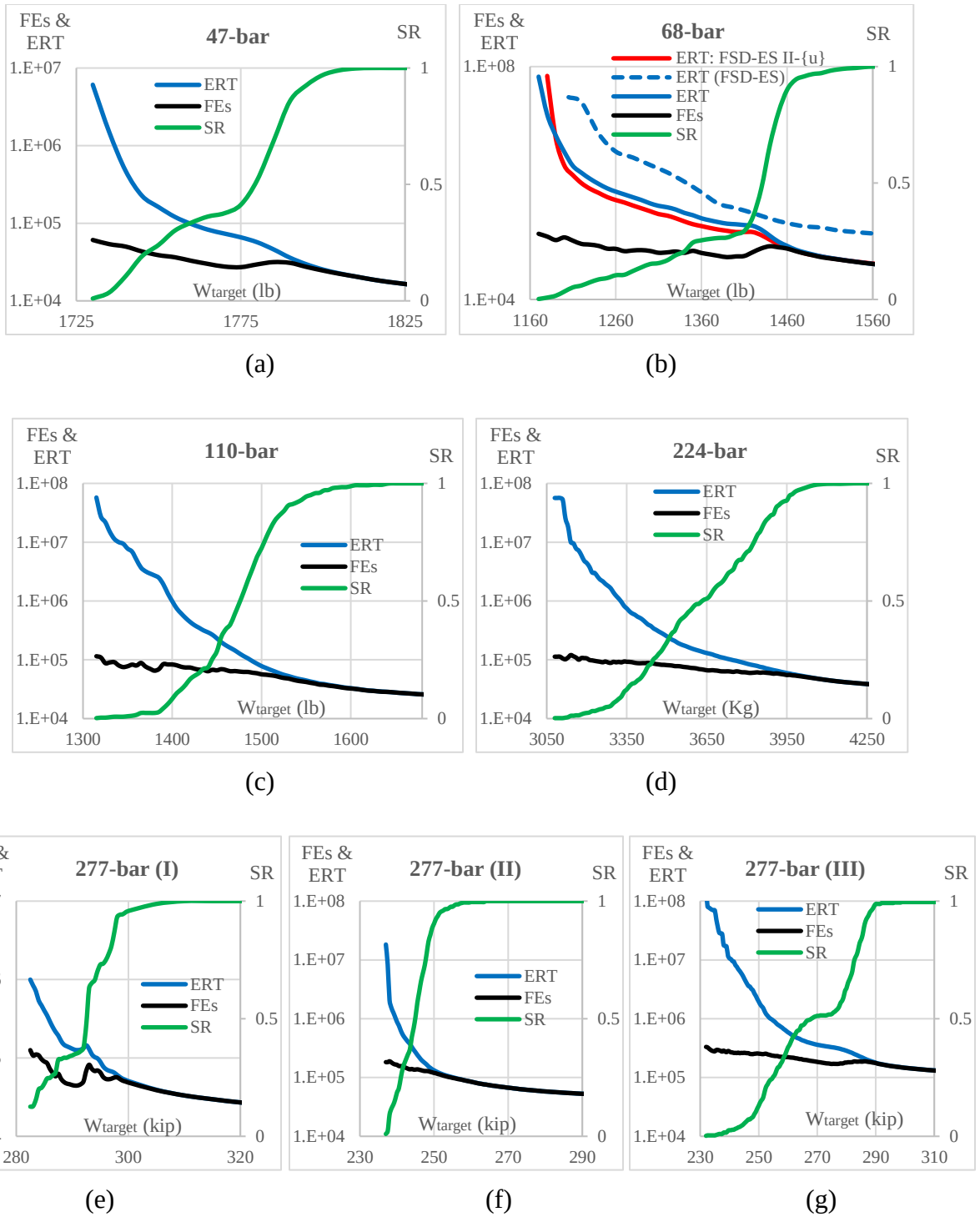


Figure 25. FES, SR and ERT as a function of the target weight (Wtarget) for a) 47-bar, b)68-bar, c) 110-bar, d) 224-bar, e) 277-bar (Variant I), f) 277-bar (Variant II) and g) 277-bar (Variant III) test problems.

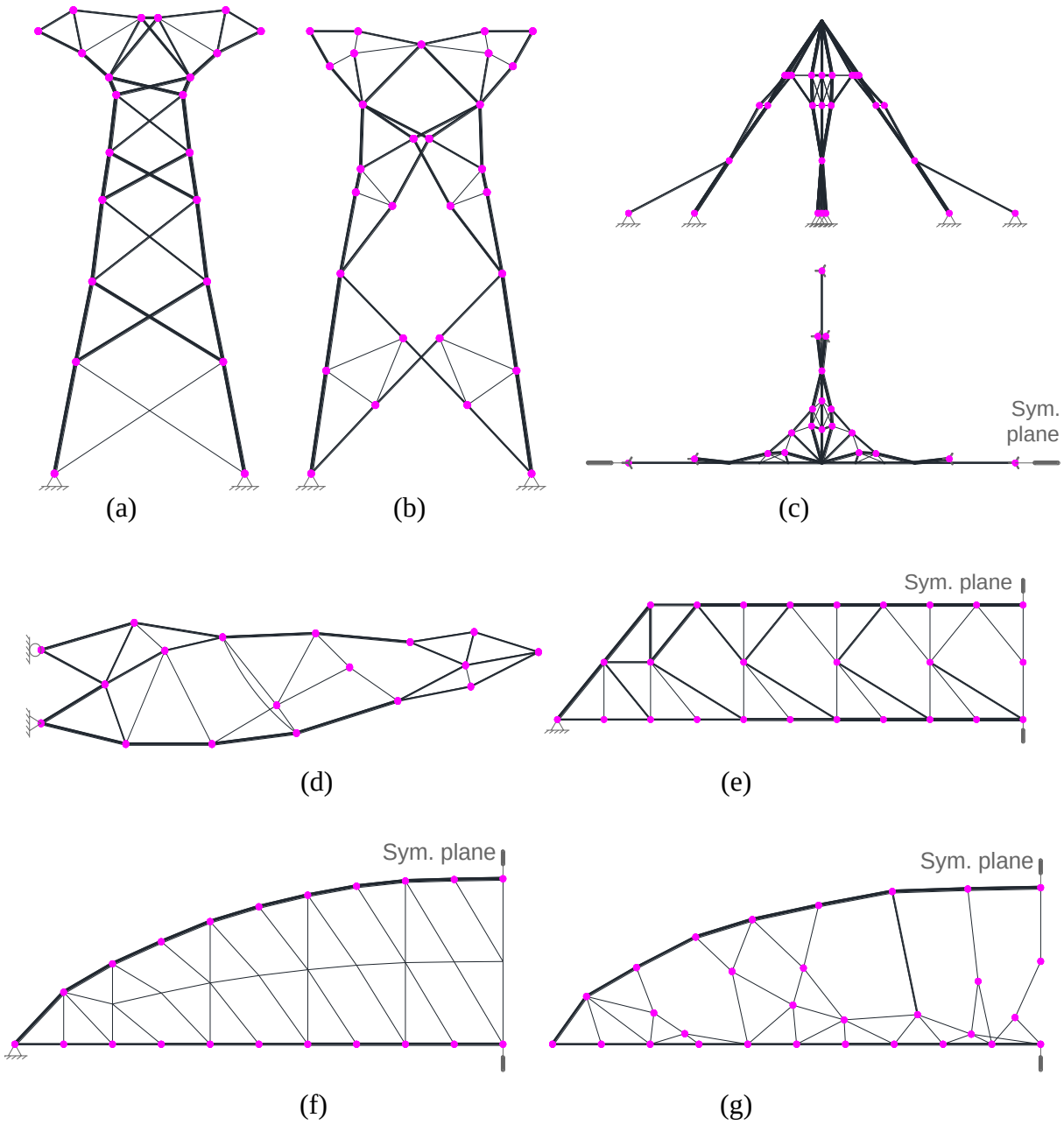


Figure 26. The best feasible solution found for different test problems. a) 47-bar (W=1727.6 lb), b) 110-bar (W=1314.0 lb), c) 224-bar (W=3079.4 Kg), d) 68-bar (W=1166.1 lb), e) 277-bar in case I (W=282.03 kip), f) 277-bar in case II (W=236.54 kip), g) 277-bar in case III (W=231.94 kip)

Table 17. Parameters of the best solution found for the 77-bar, 224-bar and 277-bar problems

	77-bar (in and in^2)		224-bar (cm and cm^2)		277-bar (I) (in and in^2)		277-bar (II) (in and in^2)			277-bar (III) (in and in^2)		
	y_3	224.347	y_2	-150.800	y_{31}	748.714	y_4	314.233	$A_{21\text{-}24}$ 6.49	x_3	208.021	$A_{7\text{-}8}$ 6.49
	y_5	321.370	z_2	718.123	$A_{1\text{-}2}$	21.8	y_7	485.359	$A_{22\text{-}24}$ 9.71	y_3	289.400	$A_{7\text{-}9}$ 5.57
	y_7	431.315	y_3	-46.344	$A_{1\text{-}3}$	68.5	y_{10}	617.425	$A_{22\text{-}25}$ 56.8	x_4	514.544	$A_{8\text{-}11}$ 24.1
	y_9	521.002	x_3	-191.008	$A_{2\text{-}3}$	6.49	y_{13}	739.897	$A_{23\text{-}24}$ 9.71	y_4	463.214	$A_{9\text{-}11}$ 9.71
	y_{11}	592.148	x_4	-172.365	$A_{2\text{-}5}$	21.8	y_{16}	830.438	$A_{23\text{-}26}$ 39.9	x_6	879.829	$A_{10\text{-}11}$ 7.65
	y_{13}	650.329	y_{19}	-44.058	$A_{3\text{-}5}$	9.71	y_{19}	899.228	$A_{24\text{-}25}$ 9.71	y_6	646.499	$A_{10\text{-}12}$ 7.65
	y_{15}	703.318	x_{19}	-280.169	$A_{3\text{-}6}$	11.5	y_{22}	950.765	$A_{24\text{-}26}$ 9.71	x_7	623.346	$A_{10\text{-}13}$ 9.71
	y_{17}	741.956	x_{20}	-322.412	$A_{3\text{-}7}$	61.8	y_{25}	985.011	$A_{24\text{-}27}$ 6.49	y_7	186.673	$A_{11\text{-}12}$ 6.49
	y_{19}	761.865	x_{36}	-478.008	$A_{5\text{-}6}$	6.49	y_{28}	993.227	$A_{25\text{-}27}$ 9.71	x_9	819.550	$A_{11\text{-}14}$ 32.9
	y_{21}	765.677	y_{51}	-17.302	$A_{5\text{-}8}$	21.8	y_{31}	997.973	$A_{25\text{-}28}$ 56.8	y_9	62.113	$A_{12\text{-}14}$ 3.54
	$A_{1\text{-}2}$	35.3	x_{51}	-657.798	$A_{6\text{-}7}$	23.2	$A_{1\text{-}2}$	25.6	$A_{26\text{-}29}$ 42.7	x_{10}	1107.381	$A_{12\text{-}15}$ 3.54
	$A_{1\text{-}3}$	68.5	$A_{36\text{-}51}$	35.9999	$A_{6\text{-}8}$	9.71	$A_{1\text{-}4}$	67.7	$A_{27\text{-}29}$ 9.71	y_{10}	433.965	$A_{12\text{-}19}$ 7.61
	$A_{2\text{-}3}$	3.54	$A_{36\text{-}52}$	6.9032	$A_{6\text{-}10}$	42.7	$A_{2\text{-}4}$	6.49	$A_{27\text{-}30}$ 6.49	x_{12}	1480.945	$A_{13\text{-}15}$ 7.65
	$A_{2\text{-}4}$	35.3	$A_{19\text{-}36}$	27.7419	$A_{6\text{-}11}$	14.4	$A_{2\text{-}5}$	25.6	$A_{28\text{-}30}$ 9.71	y_{12}	233.424	$A_{13\text{-}16}$ 75.6
	$A_{3\text{-}4}$	14.4	$A_{20\text{-}36}$	17.2903	$A_{7\text{-}10}$	26.5	$A_{4\text{-}5}$	7.65	$A_{28\text{-}31}$ 56.8	x_{13}	1227.996	$A_{14\text{-}17}$ 28.2
	$A_{3\text{-}5}$	68.5	$A_{2\text{-}19}$	4.3161	$A_{8\text{-}11}$	24.1	$A_{4\text{-}6}$	6.49	$A_{29\text{-}30}$ 9.71	y_{13}	749.968	$A_{14\text{-}19}$ 6.49
	$A_{4\text{-}5}$	14.4	$A_{3\text{-}19}$	23.7419	$A_{10\text{-}12}$	19.1	$A_{4\text{-}7}$	55.8	$A_{30\text{-}31}$ 9.71	x_{15}	1544.228	$A_{15\text{-}16}$ 6.49
	$A_{14\text{-}16}$	55.8	$A_{4\text{-}19}$	3.1871	$A_{10\text{-}13}$	56.8	$A_{5\text{-}6}$	6.49		y_{15}	455.410	$A_{15\text{-}19}$ 6.49
	$A_{15\text{-}16}$	15.8	$A_{3\text{-}20}$	3.1871	$A_{11\text{-}12}$	9.71	$A_{5\text{-}8}$	29.1		x_{16}	1633.329	$A_{16\text{-}18}$ 83.3
	$A_{15\text{-}17}$	75.6	$A_{4\text{-}20}$	17.2903	$A_{11\text{-}14}$	31.2	$A_{6\text{-}7}$	4.41		y_{16}	836.159	$A_{17\text{-}19}$ 3.54
	$A_{16\text{-}17}$	14.4	$A_{19\text{-}20}$	2.7935	$A_{12\text{-}13}$	6.49	$A_{6\text{-}8}$	6.49		x_{18}	2085.647	$A_{17\text{-}20}$ 28.2
	$A_{16\text{-}18}$	55.8	$A_{1\text{-}2}$	6.9032	$A_{12\text{-}14}$	9.71	$A_{6\text{-}9}$	6.49		y_{18}	921.111	$A_{18\text{-}21}$ 14.4
	$A_{17\text{-}18}$	21.8	$A_{1\text{-}3}$	27.7419	$A_{12\text{-}16}$	21.8	$A_{7\text{-}9}$	6.49		x_{19}	1790.757	$A_{18\text{-}25}$ 83.3
	$A_{17\text{-}19}$	75.6	$A_{2\text{-}3}$	1.6129	$A_{12\text{-}17}$	15.6	$A_{7\text{-}10}$	56.8		y_{19}	146.595	$A_{19\text{-}20}$ 6.49
	$A_{18\text{-}19}$	14.4	$A_{1\text{-}4}$	17.2903	$A_{13\text{-}16}$	56.8	$A_{8\text{-}11}$	32.9		x_{21}	2243.269	$A_{19\text{-}21}$ 10
	$A_{18\text{-}20}$	56.8	$A_{3\text{-}4}$	4.1226	$A_{14\text{-}17}$	32.9	$A_{9\text{-}11}$	7.65		y_{21}	179.840	$A_{20\text{-}21}$ 3.54
	$A_{19\text{-}20}$	21.8			$A_{16\text{-}18}$	9.71	$A_{9\text{-}12}$	6.49		x_{24}	2571.264	$A_{20\text{-}23}$ 32.9
	$A_{19\text{-}21}$	75.6			$A_{16\text{-}19}$	67.7	$A_{10\text{-}12}$	6.49		y_{24}	58.838	$A_{21\text{-}23}$ 4.41
	$A_{20\text{-}21}$	15.8			$A_{17\text{-}18}$	17	$A_{10\text{-}13}$	56.8		x_{25}	2549.469	$A_{21\text{-}24}$ 11.2
					$A_{17\text{-}20}$	46.7	$A_{11\text{-}12}$	6.49		y_{25}	935.135	$A_{23\text{-}24}$ 3.54
					$A_{18\text{-}19}$	6.49	$A_{11\text{-}14}$	35.3		x_{27}	2848.702	$A_{23\text{-}26}$ 32
					$A_{18\text{-}20}$	9.71	$A_{12\text{-}13}$	6.49		y_{27}	162.706	$A_{24\text{-}26}$ 4.41
					$A_{18\text{-}22}$	21.8	$A_{12\text{-}14}$	9.71		x_{28}	2615.871	$A_{24\text{-}28}$ 6.49
					$A_{18\text{-}23}$	14.4	$A_{12\text{-}15}$	6.49		y_{28}	378.151	$A_{24\text{-}29}$ 8.79
					$A_{19\text{-}22}$	67.7	$A_{13\text{-}15}$	7.65		y_{30}	944.284	$A_{25\text{-}28}$ 9.71
					$A_{20\text{-}23}$	50	$A_{13\text{-}16}$	55.8		y_{31}	499.294	$A_{25\text{-}30}$ 83.3
					$A_{22\text{-}24}$	9.71	$A_{14\text{-}17}$	38.8		$A_{1\text{-}2}$	19.1	$A_{26\text{-}27}$ 3.54
					$A_{22\text{-}25}$	75.6	$A_{15\text{-}17}$	9.71		$A_{1\text{-}3}$	50	$A_{26\text{-}28}$ 6.49
					$A_{23\text{-}24}$	9.71	$A_{15\text{-}18}$	6.49		$A_{2\text{-}3}$	6.49	$A_{26\text{-}29}$ 35.3
					$A_{23\text{-}26}$	51.8	$A_{16\text{-}18}$	9.71		$A_{2\text{-}5}$	20	$A_{27\text{-}29}$ 3.54
					$A_{24\text{-}25}$	6.49	$A_{16\text{-}19}$	56.8		$A_{3\text{-}4}$	55.8	$A_{27\text{-}31}$ 6.49
					$A_{24\text{-}26}$	9.71	$A_{17\text{-}18}$	7.65		$A_{3\text{-}5}$	9.71	$A_{30\text{-}31}$ 7.65
					$A_{24\text{-}28}$	14.4	$A_{17\text{-}20}$	38.8		$A_{3\text{-}7}$	8.79	
					$A_{24\text{-}29}$	14.4	$A_{18\text{-}19}$	7.65		$A_{4\text{-}6}$	62	
					$A_{25\text{-}28}$	75.6	$A_{18\text{-}20}$	9.71		$A_{4\text{-}7}$	6.49	
					$A_{26\text{-}29}$	55.8	$A_{18\text{-}21}$	6.49		$A_{5\text{-}7}$	6.49	
					$A_{28\text{-}30}$	9.71	$A_{19\text{-}21}$	9.71		$A_{5\text{-}8}$	20	
					$A_{28\text{-}31}$	75.6	$A_{19\text{-}22}$	55.8		$A_{5\text{-}9}$	4.99	
					$A_{29\text{-}30}$	6.49	$A_{20\text{-}23}$	39.9		$A_{6\text{-}10}$	7.61	
					$A_{30\text{-}31}$	6.49	$A_{21\text{-}23}$	9.71		$A_{6\text{-}13}$	61.8	
Weight	**305.964 kip**		**3079.446 Kg**		**282.033 kip**		**236.543 kip**			**231.943 kip**		
$\max_{i,l}\{\sigma_{il}\}$	1.000		0.993		1.000		1.000			0.999		
$\max_{i}\{s_i\}$	1.000		0.983		0.965		0.999			0.995		
$\max_{k,l}\{u_{kl}\}$	1.000		0.911		0.960		0.753			0.933		

Table 18. Parameters of the best solution found for the 47-bar, 68-bar and 110-bar truss problems

	47-bar (in and in^2)				**68-bar (in and in^2)**				**110-bar (in and in^2)**			
	x_1	-127.426	$A_{12\text{-}13}$	1.4	x_2	77.158	$A_{3\text{-}6}$	2.142	x_1	-147.839	$A_{3\text{-}8}$	0.5
	x_3	-98.277	$A_{13\text{-}21}$	0.7	y_2	2.241	$A_{4\text{-}5}$	0.347	x_3	-61.793	$A_{5\text{-}7}$	0.1
	y_3	151.332	$A_{13\text{-}15}$	1.6	x_4	102.478	$A_{5\text{-}6}$	0.347	y_3	92.508	$A_{5\text{-}9}$	2.6
	x_5	-76.494	$A_{19\text{-}21}$	0.9	y_4	-63.299	$A_{4\text{-}7}$	3.131	x_5	-127.842	$A_{7\text{-}9}$	0.5
	y_5	260.438	$A_{15\text{-}19}$	0.8	x_5	149.668	$A_{5\text{-}7}$	0.44	y_5	139.089	$A_{9\text{-}11}$	1.6
	x_7	-64.180	$A_{15\text{-}21}$	0.2	y_5	39.504	$A_{5\text{-}9}$	1.081	x_7	-24.589	$A_{9\text{-}13}$	1.2
	y_7	370.620	$A_{17\text{-}19}$	0.9	x_6	112.369	$A_{6\text{-}9}$	1.488	y_7	183.323	$A_{11\text{-}13}$	0.1
	x_9	-53.706	$A_{15\text{-}17}$	1.2	y_6	69.877	$A_{7\text{-}8}$	0.27	x_9	-108.653	$A_{11\text{-}15}$	1.6
	y_9	435.519	$A_{14\text{-}21}$	1.1	x_7	206.289	$A_{8\text{-}9}$	0.44	y_9	270.627	$A_{13\text{-}15}$	0.2
	x_{11}	-44.619	$A_{21\text{-}22}$	1.2	x_{12}	331.501	$A_{7\text{-}10}$	2.8	x_{11}	-88.025	$A_{13\text{-}18}$	1.1
	y_{11}	513.943			y_{12}	58.396	$A_{8\text{-}10}$	0.44	y_{11}	381.346	$A_{15\text{-}17}$	0.6
	x_{13}	-53.559			x_{13}	430.570	$A_{8\text{-}11}$	0.111	x_{13}	-39.385	$A_{15\text{-}19}$	1.4
	y_{13}	536.863			y_{13}	-15.651	$A_{8\text{-}12}$	0.27	y_{13}	362.847	$A_{17\text{-}19}$	1.0
	x_{19}	-102.846			x_{14}	512.163	$A_{9\text{-}12}$	2.142	x_{15}	-81.704	$A_{17\text{-}20}$	0.8
	y_{19}	628.453			y_{14}	23.357	$A_{11\text{-}12}$	0.111	y_{15}	412.900	$A_{19\text{-}21}$	0.4
	x_{21}	-10.378			x_{15}	445.326	$A_{10\text{-}13}$	2.697	x_{17}	-10.813	$A_{19\text{-}25}$	1.1
	y_{21}	618.162			y_{15}	48.998	$A_{11\text{-}13}$	0.111	y_{17}	453.908	$A_{19\text{-}29}$	1.1
	$A_{1\text{-}3}$	3.0			x_{16}	518.860	$A_{12\text{-}15}$	2.142	x_{19}	-78.574	$A_{21\text{-}25}$	0.2
	$A_{2\text{-}3}$	0.3			y_{16}	-0.055	$A_{13\text{-}14}$	1.333	y_{19}	500.112	$A_{21\text{-}27}$	0.2
	$A_{3\text{-}5}$	2.6			y_{17}	37.942	$A_{14\text{-}15}$	0.954	x_{25}	-122.764	$A_{21\text{-}29}$	0.2
	$A_{4\text{-}5}$	1.5			x_{18}	522.801	$A_{13\text{-}16}$	1.333	y_{25}	552.032	$A_{23\text{-}25}$	1.1
	$A_{5\text{-}7}$	2.8			y_{18}	60.286	$A_{14\text{-}16}$	0.347	x_{27}	-85.107	$A_{23\text{-}27}$	0.7
	$A_{6\text{-}7}$	0.6			$A_{1\text{-}2}$	3.131	$A_{14\text{-}17}$	1.333	y_{27}	599.788	$A_{27\text{-}29}$	0.7
	$A_{7\text{-}9}$	2.3			$A_{2\text{-}3}$	1.333	$A_{14\text{-}18}$	0.539	y_{29}	582.204		
	$A_{7\text{-}10}$	1.1			$A_{1\text{-}4}$	3.131	$A_{15\text{-}18}$	1.174	$A_{1\text{-}3}$	0.5		
	$A_{9\text{-}11}$	2.5			$A_{2\text{-}4}$	0.539	$A_{16\text{-}17}$	1.333	$A_{1\text{-}5}$	2.6		
	$A_{10\text{-}11}$	0.6			$A_{2\text{-}5}$	1.333	$A_{17\text{-}18}$	1.081	$A_{3\text{-}5}$	0.1		
	$A_{11\text{-}13}$	2.3			$A_{2\text{-}6}$	1.081	$A_{9\text{-}10}$	0.111	$A_{3\text{-}7}$	0.1		
Weight	**1727.62 lb**				**1166.06 lb**				**1314.05 lb**			
$\max_{i,l}\{\sigma_{il}\}$	1.000				1.000				1.000			
$\max_{i,l}\{f_{il}\}$	1.000				0.999				0.999			
$\max_{k,l}\{u_{kl}\}$	-				0.981				-			

CHAPTER 7. FULLY STRESSED DESIGN EVOLUTION STRATEGY WITH ARBITRARY RESIZING BUDGET

In FSD-ES II, the evaluation budget for resizing each solution (N_{resize}) is set to one. This is because of two reasons: First, the resizing step explores the size space only. Excessive allocation of evaluation budget for resizing may result in convergence to a poor shape or topology. Second, this setting was motivated by behavior of FSD, in which the maximal gain is reached during the first iteration [64].

Although this setting is justified, the pitfall is that the resized structure may still violate some constraint, since member forces change during the resizing step. Therefore, for complicated problems with many constraints and members, it may take a long time to find a feasible design. It remains unclear whether a higher value for N_{resize} can provide advantages. Furthermore, this parameter might be problem-dependent. For example, for problems in which most design parameters are size parameters, a greater N_{resize} might be advantageous. It is also possible that an adaptive scheme can outperform any fixed values. The generalized version of FSD-ES II, called FSD-ES IIb, is introduced in this chapter[9] which can control the budget of the lower level by tuning the value of N_{resize}. This generalization particularly allows for arbitrary trade-off between the upper level evolution strategy and the lower level FSD-based resizing.

7.1. New Features

Figure 27 presents the flowchart of FSD-ES IIb. There are a few minor modifications as well, which will be discussed in this section.

[9] This chapter uses some materials from our submitted work to Applied Soft Computing. The probable publisher's policy allows reuse of the materials published by the authors in their dissertation.

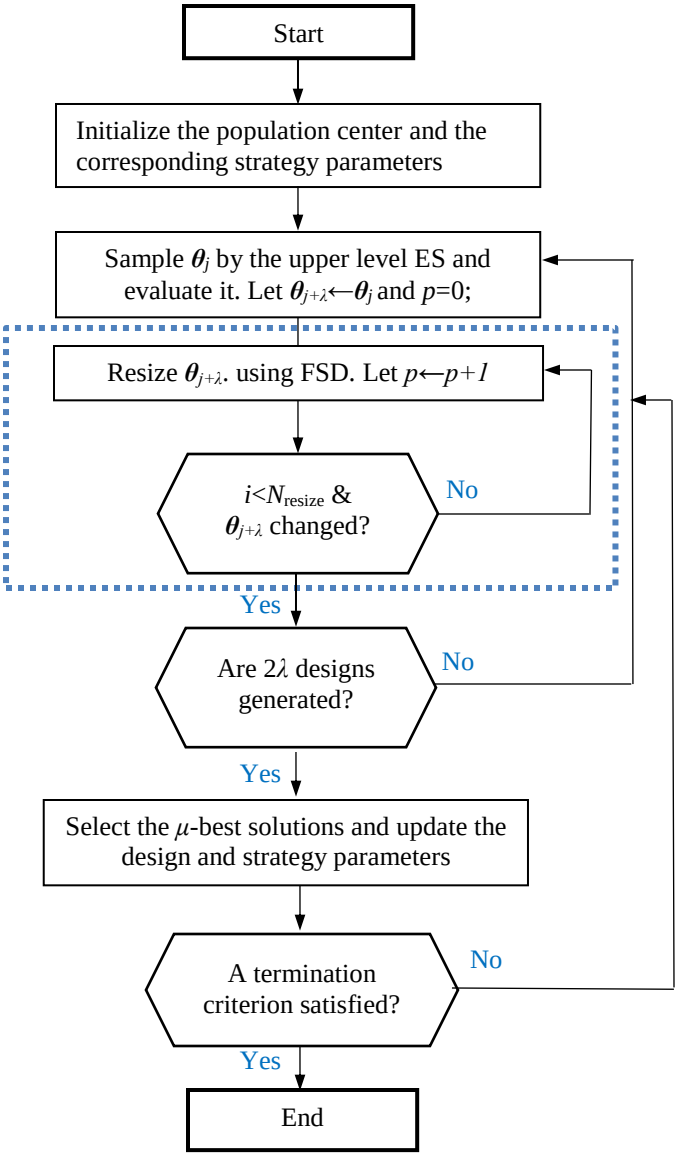


Figure 27. Flowchart of FSD-ES IIb. The blue dotted rectangle specifies iterative resizing defined in FSD-ES IIb.

7.1.1. Controlling the Lower Loop Budget

In FSD-ES IIb, each sample design by the upper loop ($\boldsymbol{\theta}_j$), undergoes iterative resizing up to N_{resize} times to generate the repaired design $\boldsymbol{\theta}_{j+\lambda}$, which differs from $\boldsymbol{\theta}_j$ only in size. Resizing process terminates if no change in two consecutive resizing iterations is observed, and thus the number of evaluations for resizing a sampled design can be less than N_{resize}.

One challenge in comparing different variants is that the depleted evaluation budget in one generation depends on the value of N_{resize}. This means that for a fixed value of λ, variants with a greater N_{resize} have fewer iterations to converge. The exact number of resizing evaluations is not known beforehand because the resizing process terminates if no change is observed in two successive resizing iterations. This means reducing λ by a fixed value cannot reliably address this issue. To handle this challenge, the value of λ is updated iteratively such that the number of function evaluations per generation (*FEpG*) remains constant. The default value of this parameter will be discussed in Section 7.1.3

7.1.2. Biasing the Cross-Sectional Areas

If no resizing is performed, especially when there are many discrete size parameters, the scaling factor might become too small such that increasing the penalty term cannot force the algorithm to increase the cross-sectional areas of the constraint violating member in the subsequent iterations; therefore, the algorithm might converge to an infeasible solution. To alleviate this problem, the size of the sampled solution in the upper level is rounded to the closest upper value (instead of stochastic rounding) if the corresponding penalty coefficient is greater than one, which is the minimal value.

7.1.3. Parameter Setting

All parameters of FSD-ES IIb are set to default values, as performed for the early approach. A few updates in the default setting is performed as follows:

- The default setting in FSD-ES II leads to $FEpG \approx 4N_{eff}^{0.5}$. The same value is used for FSD-ES IIb with an arbitrary N_{resize}. Starting with the initial value of $\lambda \approx 4N_{eff}^{0.5}/(1+N_{resize})$, the number of function evaluations is calculated at the end of generation and if smaller than *FEpG,* λ is

increased accordingly and vice versa.

- A simpler relation for calculation of N_{eff} is suggested as follows:

$$N_{VAR} = \left(\sqrt{N_{top}} + \sqrt{N_{shape}} + \sqrt{N_{size}}\right)^2$$

$$N_{\text{eff}} = (N_{\text{VAR}}) \left(1 + \frac{\sqrt{N_l}(N_m + DN_n)}{N_{\text{VAR}}}\right)^{0.5} \tag{57}$$

in which N_l is the number of load cases and D=2 for planar and D=3 for spatial structures.

- Like FSD-ES II, FSD-ES IIb considers only the diagonal elements of the covariance matrix (scaling factors); therefore, the number of strategy parameters to be adapted is smaller than the case in CMA-ES [35], which adapts the full covariance matrix. This means the scaling factors can be adapted at a higher rate [110]. Accordingly, the covariance learning time constant (τ_C) which specifies the learning rate of the covariance matrix is set proportional to N_{eff}: $\tau_C = 1+N_{\text{eff}}/(4\mu_{\text{eff}})$.
- The initial scaling factors for the topology, shape and size variables are set to 2, 1.5 and 1, respectively.

7.2. Numerical Evaluation[10]

The goal of the experiments presented here is mainly to scrutinize the contributions of the lower level FSD-based resizing and the upper level evolution strategy. Accordingly, different values of N_{resize} are tested for each problem while other parameters are set to their default values. The optimal value of N_{resize} is sought which optimizes the performance measure for each problem

[10] For the latest source codes of the developed methods in this dissertation, please visit the author's ResearchGate page: https://www.researchgate.net/profile/Ali_Ahrari/contributions

and justifications for optimality of a specific value/range of N_{resize} is provided. It is predicted that a greater N_{resize} provides better solutions in short-term (better exploitation) while a smaller one is deemed beneficial in long-term (more exploration).

Two of the most complicated truss problems are selected from the literature. These two problems are solved with conventional grouping of members (Variant I) for comparison with available results in the literature. They are also tested without conventional grouping (Variant II), which allows for testing the method in problems with more design parameters without increasing the evaluation cost of a design. The maximum number of design parameters reaches 1571, which to the authors' knowledge, is the highest number of design parameters tested in discrete truss optimization by metaheuristics. These two problems are size optimization of a 960-bar double layer grid [62, 79, 83] and simultaneous topology, shape and size optimization of a 759-bar truss bridge [129, 96, 130, 131]. For both problems, design constraints are governed by AISC-ASD specifications [132]. The following values are used: F_y=248.21 MPa (36 Ksi), Density=7850 Kg/m^3 and E=200 GPa (29000 Ksi).

7.2.1. Size Optimization of 960-Bar Double Grid

The 960-bar is one the most complicated size optimization problem available in the literature. It is a double layer roof truss which consists of 960 members and 263 joints (Figure 28). Nodes 1, 5, 9, 13, 66, 78, 131, 135, 139, 143 are constrained from moving in any directions, and thus 201 members (about 21%) are redundant. This high redundancy can challenge assumptions, and hence the potential contribution of the FSD part, which is another reason for selection of this problem. Symmetry is conventionally exploited to reduce the number of design parameters to 251. The sections are selected from a list of 28 steel hollow circular sections, stated in ASD-AISC [132]. The structure is supported at 10 nodes on the periphery of the bottom layer, and under the action

of snow load only, with a uniform snow pressure of 754 N/m^2. Member stress and stability limitations are in accordance with ASD-AISC, and nodal displacements are restricted to a maximum of 10.57 cm in any direction. In Variant II, one independent design parameter is considered for each member which increases the number of design parameters to 960.

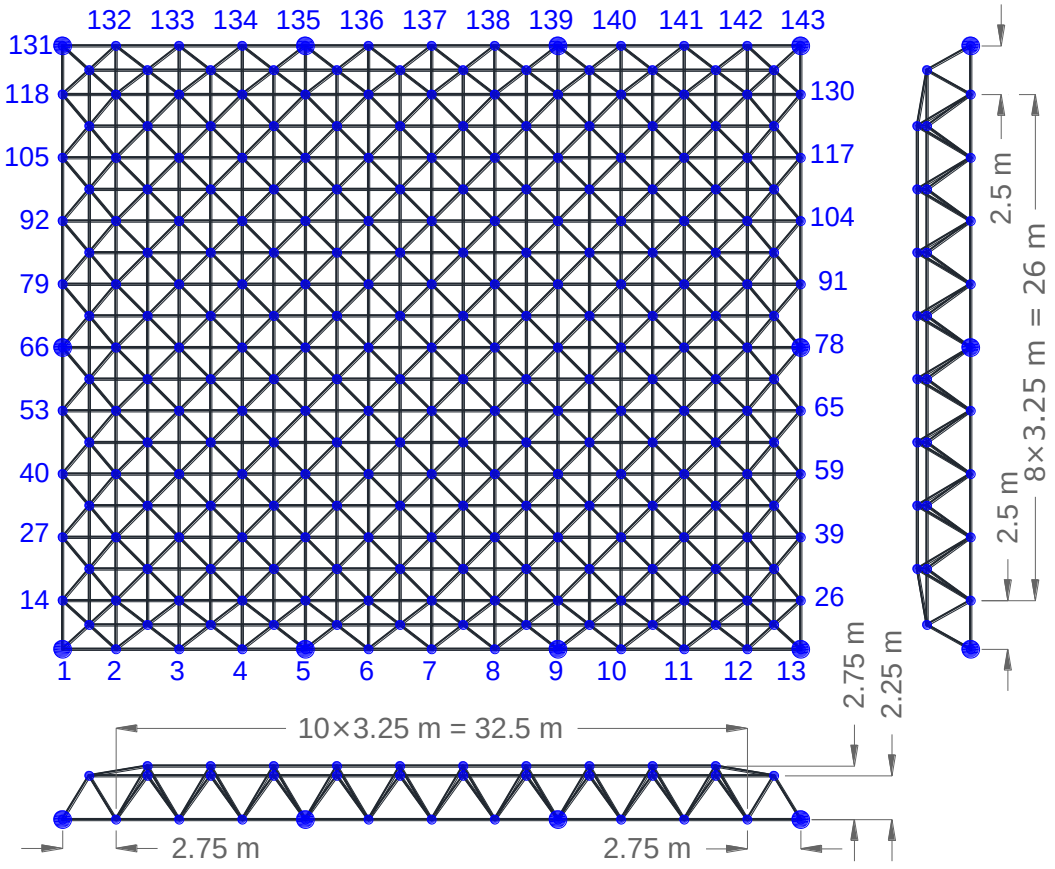


Figure 28. Ground structure for the 960-bar problem (front, side and top views)

7.2.2. Physical Design Area Problem

The physical design area problem is simultaneous optimization of topology, shape and size of 759-bar truss bridge with 40 nodes. The lower cord of the bridge (Figure 29) is defined by seven members and eight nodes. A rectangular physical design area with height of 35 m and length of 70 m is considered. To impose symmetry, the remaining nodes are divided into three groups [96]:

- 14 independent nodes in the left side of the design space area.
- 14 dependent nodes in the right side, which are linked to the nodes in the first group by symmetry.
- 4 independent nodes lying on the symmetry line.

These 32 nodes are connected to each other (16×31=496) and the nodes on the lower cord (32×8=256). Therefore, the overall number of members is 496+256+7=759. One topology variable is assigned per independent member, except for the members on the lower cord which must be present. Therefore, the overall number of topology, shape and size variables is 386+32+390=808. The allowable displacement of all nodes in any direction is 7 cm. The problem necessitates that lengths of the members be within the range of 5-30 meters. To handle this constraint, a member is deactivated if the length is outside of this range. When grouping is ignored (Variant II), the overall number of independent topology, shape and size parameters increases to 752+60+759=1571, which, to the authors' knowledge, is the highest number of design parameters tested in the literature of truss optimization by metaheuristics.

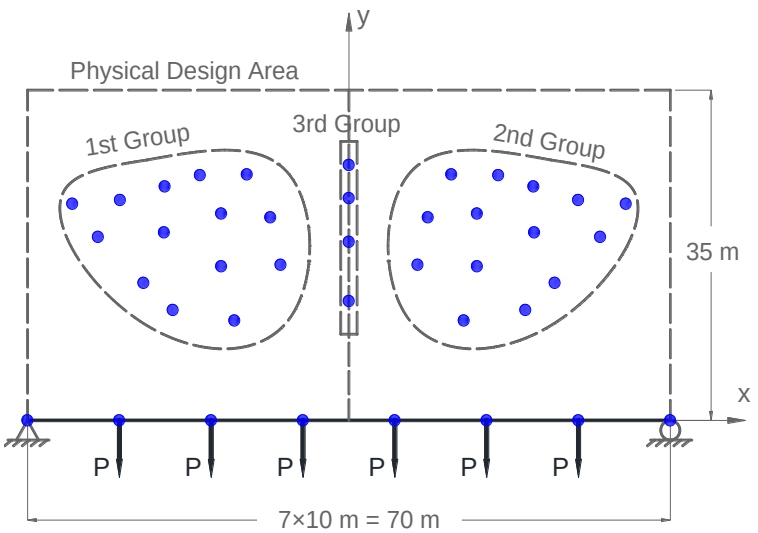
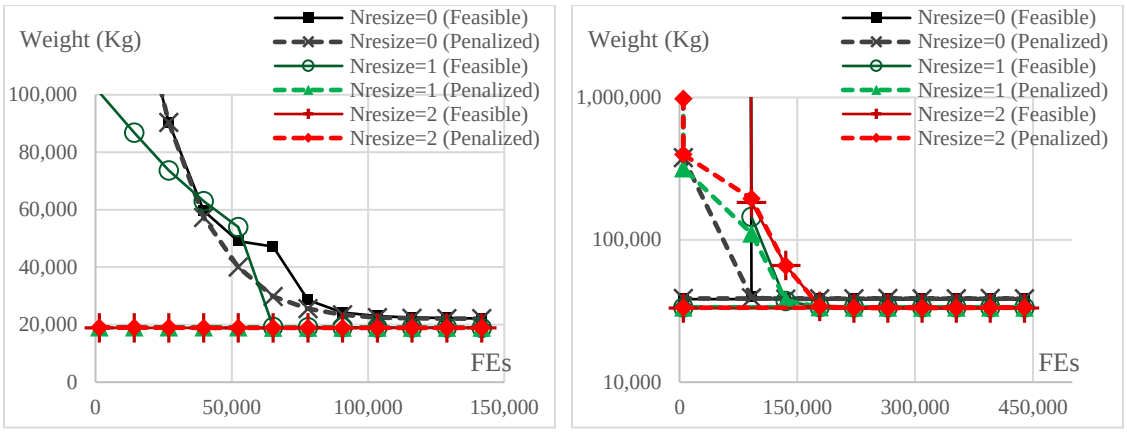


Figure 29. Illustration of the 759-bar physical design problem

7.3. Results and Discussion

Each problem is solved with different values of N_{resize} and the Expected Running Time (*ERT*) [123] to reach selected target feasible weights for different values of N_{resize} is plotted. In addition to $ERT(W_{target})$, convergence history, including the median of the best penalized function and median of the best feasible weight are presented to provide a qualitative analysis of the behavior of different variants.

FEpG and maximum number of evaluations (*MaxFE*) are set following the default parameter setting. These values are summarized in Table 19. For each variant and each value of N_{resize}, 50 independent runs for the 960-bar grid problem and 100 independent runs for the 759-bar truss bridge problem are performed. The reason for a greater number of independent runs for the 759-bar problem is the high variance among the outcomes of independent runs because of the huge number of possible topologies. The convergence history for the 960-bar and 759-bar problems in Variant I are plotted in Figure 30. Figure 31 and Figure 32 illustrates $ERT(W_{target})$, and $SR(W_{target})$ for the 960-bar and 759-bar problems, respectively. Some reasonable values of W_{target} were selected while N_{resize} increases from 0 to 10. Figure 33 illustrates the best solutions found for the physical design problem and the corresponding data are presented in Table 21. Details of the best solution found for the 960-bar grid problem can be found in the supplementary materials.



(a) 960-bar problem (b) 759-bar problem

Figure 30. Convergence history (best penalized weight and best feasible weight versus FEs) in Variant for different values of N_{resize}.

Table 19. Calculation of default values of FSD-ES IIb parameters using Equation 57

	960-bar (I)	**960-bar (II)**	**759-bar (I)**	**759-bar (II)**
N_{top}	0	0	752	386
N_{shape}	0	0	60	32
N_{size}	251	960	759	390
N_{node}	263	263	40	40
N_{member}	960	960	759	759
N_{eff}	709	1613	2413	4333
FEpG	106	161	196	263
MaxFEE	141,700	322,530	482,600	866,570

- Figure 30 demonstrates that for the 960-bar problem, excluding the resizing step results in a slow and inefficient progress towards the global minimum. There is no considerable gap between the best penalized weight and the best feasible weight for N_{resize}=0; however, the final solution is heavier than the other variants with $N_{resize} > 0$. For N_{resize}=1, the best penalized function approaches the value of 19000 Kg in early iterations, but the best feasible design remains far from this value until about 40% of the evaluation budget is used. This is due the fact that the resizing step generates a solution close to the boundary of feasible region; however, since assumptions of FSD are not perfectly valid, the resized solution may still violate some constraints slightly. The change during the resizing step

gradually decreases as the algorithm converges, and hence the assumptions of FSD become more valid. For N_{resize}=2, the algorithm can find a near-optimal solution in the early generations. This demonstrates that for this problem, two iterations of the resizing step may result in a near-optimal feasible solution before any evolutionary operator can affect the search. Consequently, successive resizing of a randomly generated design can result in a near-optimal solution, much better than what the stochastic search method, the case with N_{resize}=0, may achieve after thousands of evaluations.

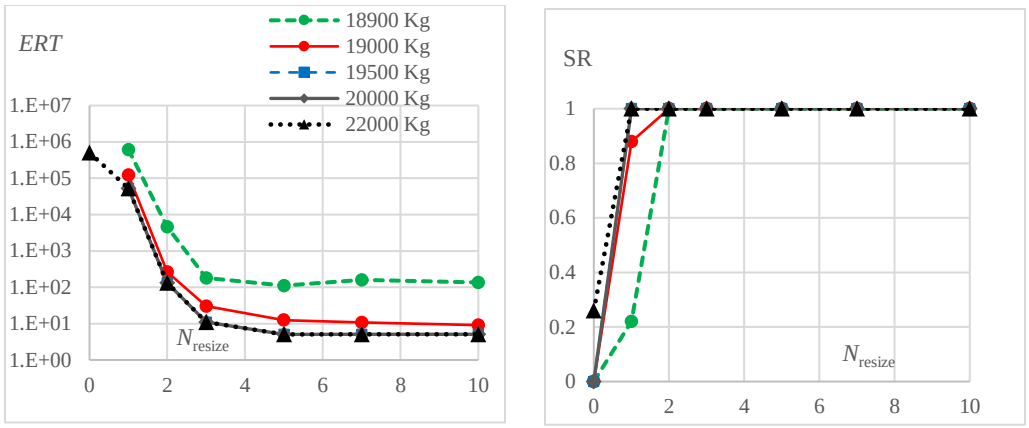
- Figure 31 demonstrates that for the 960-bar problem, increasing N_{resize} continually and significantly reduces *ERT* up to N_{resize}=3. Additionally, the algorithm cannot reach a solution equal to or lighter than 19,000 Kg if N_{resize}=0. For $N_{resize}\geq$2, all runs of the algorithm can reach W_{target}=19,000 Kg in less than 320 evaluations. For this problem, the best available results in the literature are 23,521.5 kg after 10000 evaluations using adaptive dimensional search [79], 24,266.7 kg after 100,000 evaluations by big bang-big crunch [83] and 24,388.3 kg by simulated annealing and 24,780.2 kg by evolution strategies after 100,000 evaluations [62]. In comparison with these methods, Table 20 shows that FSD-ES IIb with N_{resize}=3 could reach a *19.2%* lighter design (W_{target}=19,000 Kg) more than *300 times* faster (*FEs*=*ERT*=29)[11].
- When the grouping is ignored (Variant II), the number of design parameters is almost quadrupled; nevertheless, for $N_{resize}\geq$5, the number of evaluations to reach a near-optimal weight of 19,000 Kg almost doubles (Figure 31). This shows that the required number of evaluations increases at a much slower rate when compared to metaheuristic methods, in

[11] Detailed data for the reported results on this problem are not available in the literature. It is not possible to double check that the same problem is tested; however, the case with N_{resize}=0 can represent a competent metaheuristic method for truss optimization.

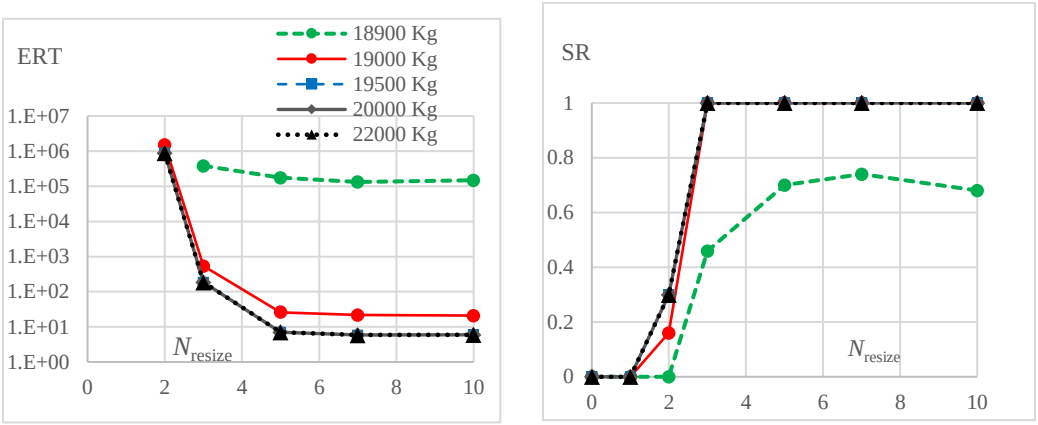
which the required number of evaluations grows polynomially with respect to the number of design variables.

- The contributing of higher values of N_{resize} is not that spectacular for the 759-bar problem (Figure 32). N_{resize}=0 shows a higher rate of weight reduction in early iterations, but falls behind N_{resize} =1 and N_{resize} =2 in the middle stages of the optimization. This was predictable, since the size of possible topologies and to some extent, shape space is huge and the weight cannot be minimized when the topology and the shape are far from the optimum. At the same time, axial forces undergo significant changes especially when topology is modified, and hence the optimality of size parameters vanishes when the topology and the shape, is modified. For these reasons, N_{resize}=0 is advantageous when early progress rate towards the near-optimal topology is considered; however, after that, it falls behind the variants with N_{resize}=1 and N_{resize}=2.
- On remarkable observation is that even for N_{resize} =2, it takes FSD-ES IIb a long time to find a feasible solution. This is since the greatest radius of gyration in the given set of sections cannot satisfy the slenderness constraints for very long members in compression. Such members will be eliminated during the subsequent iterations by evolving the topology/shape. Before that time, there is no other way to find a feasible solution by iterative resizing.
- Figure 32 demonstrates that the best values of N_{resize} are 1, 2, and 3 for the physical design problem. A greater value deteriorates exploration in topology/shape space while a lower value negatively affects finding an optimally sized structure. The optimality of N_{resize} =1, 2, 3 is more remarkable for Variant II and when W_{target} is lower.
- For this problem, the best reported results in the literature (in Variant I) are 35,573 Kg

reached after 500,000 evaluations [130]. FSD-ES IIb with N_{resize}=3 (Table 20) could reach W_{target} of 32,000 Kg after 191,000 evaluations, with a success rate of 0.37. This means FSD-ES IIb could reach a *10*% lighter design with *62*% fewer evaluations. The best solution found by FSD-ES IIb with N_{resize}=3 weighs 27,967 Kg, which is *21.4*% lighter than the best available solution in the literature. This design is illustrated in Figure 33 and the corresponding data are tabulated in Table 21. For this problem, the stress constraint is activated and slenderness and displacement constraint are close to activation in the best-found solution. Unlike previously reported results for this problem which resemble the Parker model (see [130]), the best solution in this study resembles the Baily model. This parallels the finding in another study [133], in which a Baily-like model turned out to be the optimal model for a bridge design problem.

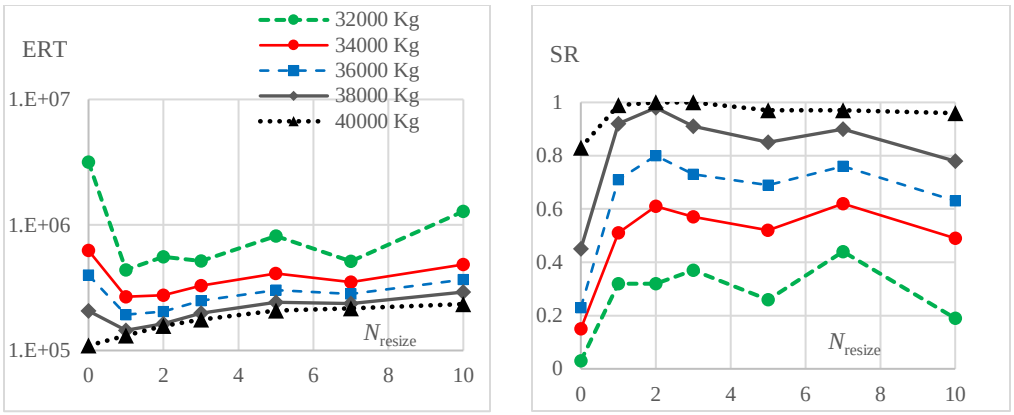


(a) Variant I

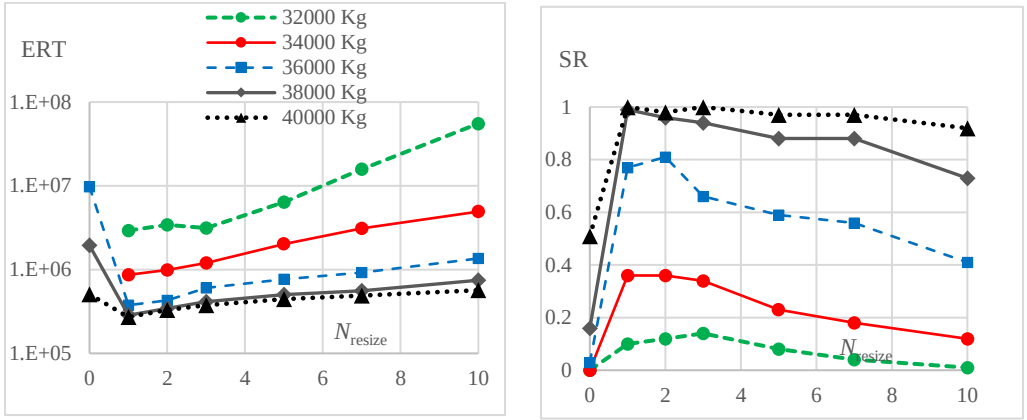


(b) Variant II

Figure 31. *ERT* and *SR* as a function of N_{resize} for the 960-bar problem for some selected values of W_{target}



(a) Variant I



(b) Variant II

Figure 32. *ERT* and *SR* as a function of N_{resize} for the 759-bar problem for some selected values of W_{target}: a) Variant I and b) Variant II

Table 20 *FEs*, *SR* and *ERT* for some selected values of W_{target} when N_{resize}=3.

	Weight (Kg)	ERT	SR	FEs
960-bar (I)	18,870	1,977	1	1,977
	18,900	177	1	177
	18,950	54	1	54
	19,000	29	1	29
759-bar (I)	28,000	22,380,000	0.01	223,800
	30,000	2,060,099	0.09	185,409
	32,000	515,741	0.37	190,824
	35,000	285,165	0.65	185,357
960-bar (II)	18,900	383,720	0.46	176,510
	18,950	4,423	1	4,423
	19,000	545	1	545
	19,100	186	1	186
759-bar (II)	30,000	44,126,000	0.01	441,260
	32,000	3,131,036	0.14	438,345
	35,000	817,445	0.5	408,723
	40,000	376,809	1	376,809

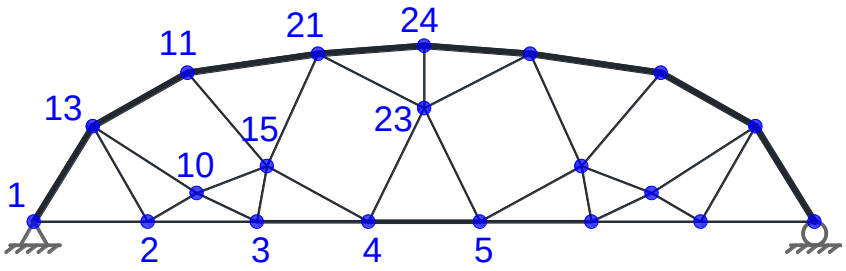


Figure 33. The best solution found by FSD-ES IIb with N_{resize}=3 for the 759-bar problem in Variant I

Table 21 Data for the best solution found for the 759-bar problem in Variant I (Weight=27,967 Kg)

x_{10}	–20.3633	$A_{1\text{-}2}$	10	$A_{10\text{-}15}$	6.49
y_{10}	2.5562	$A_{2\text{-}3}$	10	$A_{11\text{-}13}$	32
x_{11}	–21.0379	$A_{3\text{-}4}$	14.1	$A_{11\text{-}15}$	8.85
y_{11}	13.0219	$A_{4\text{-}5}$	20	$A_{11\text{-}21}$	42.7
x_{13}	–29.6019	$A_{1\text{-}13}$	35.3	$A_{15\text{-}21}$	8.85
y_{13}	8.4274	$A_{2\text{-}10}$	6.49	$A_{21\text{-}23}$	8.85
x_{15}	–14.0427	$A_{2\text{-}13}$	8.85	$A_{21\text{-}24}$	35.30
y_{15}	4.9075	$A_{3\text{-}10}$	6.49	$A_{23\text{-}24}$	6.49
x_{21}	–9.5590	$A_{3\text{-}15}$	6.49		
y_{21}	14.6720	$A_{4\text{-}15}$	8.85		
y_{23}*	9.9145	$A_{4\text{-}23}$	8.85		
y_{24}*	15.3462	$A_{10\text{-}13}$	8.85		
Max. Stress Ratio**=				1.000	
Max. Slenderness Ratio=				0.978	
Max. Displacement Ratio=				0.973	

* Node on the symmetry line

** The maximum ratio of the constraint value to the allowable limit

CHAPTER 8. SUMMARY, CONCLUSIONS AND FUTURE RESEARCH

Optimum design of truss structures is a tedious task which requires decision on diverse parameters such that many constraints are satisfied while the overall cost is minimized. The most effective, yet challenging scheme, optimizes the truss topology, shape and size (TSS) at the same time; nevertheless, a small fraction of recent methods can render this scheme. On the other hand, several factors hinder widespread application of truss optimization methods by practitioners, such as the limited complexity of the conventional test problems which cannot reliably represent the complex practical problems. These factors have resulted in a gap between truss research in academia and practitioners' preference in traditional intuition-based try-and-error methods.

This dissertation develops a truss optimization method aiming at improving and addressing some shortcomings in the current truss optimization methods, both from academic and practical perspectives. The method, called fully stressed design evolution strategy (FSD-ES) is a bi-level method where a specialized state-of-the-art evolution strategy-based method performs global search in the upper level while the concept of fully stressed design (FSD) resizes the design sampled by the upper level. FSD-ES also introduces a specialized penalty term to enhance boundary search of the algorithm based on the estimated required increase in the structural weight such that all constraints are satisfied.

FSD-ES was developed in three stages. The fist variant was proposed for shape and size optimization only, in which the upper level ES-based method allocates an independent step size per variable which undergoes the traditional concept of self-adaptation. The FSD-based resizing lower loop optimizes the size of the sampled design in the upper loop such that all stress and buckling constraints are satisfied. A heuristic for implicit consideration of displacement constraints was also implemented. In the second variant, FSD-ES was extended to handle topology optimization as well. The upper level was specialized to compensate for anisotropy of distribution

of topology variables in the population, caused by rejection of kinematically unstable designs. In the third variant, called FSD-ES II, the resizing step was revised so that it can explicitly handle the displacement constraints. The ES-based method in the upper level was improved as well to follow principles of the contemporary evolution strategies.

Each variant was numerically evaluated on some selected tests problems and the obtained results were compared with the best available results in the literature. All the variants were tested with default parameter setting, therefore, no ad-hoc parameter tuning is required for a new problem. The first variant demonstrated a slow convergence for the simplest problem; however, in the other two problems, it surpassed the best competitors. The numerical results confirmed the superiority of the second variant, since it could outperform or at least compete with the best methods in the literature. Finally, FSD-ES II, was tested on more complicated test problems and demonstrated a significant advantage over available methods.

This dissertation contributes to the truss optimization field, from both academic and practical perspectives. From the academic perspective, these contributions can be summarized as follows:

- *Capabilities:* FSD-ES can handle the most challenging and effective truss optimization scenario, simultaneous topology, shape and size, which can be performed by a small fraction of proposed truss optimization methods in the literature.
- *Superiority*: The numerical results revealed superiority of FSD-ES and FSD-ES II over the best available truss optimization methods in the literature, except for very simple problems. More importantly, the superiority of the proposed method turned out to be more significant for more complicated problems.
- *Benchmarking*: A few more complicated test problems were developed as a complement to conventional yet simple test problems. They can provide a more

reliable tool to compare different truss optimization methods more reliably in more realistic situations and therefore, the researchers are encouraged to test their methods on such problems.

- *Importance of the methodology*: The success of FSD-ES variants, especially in more complicated situations, implies that such a bi-level methodology, where global search is performed in the upper level using a metaheuristic and the problem specific knowledge is utilized in the lower level can be employed to tailor metaheuristics to different classes of engineering problems. Such specialized stochastic methods can surpass purely metaheuristic algorithms when benchmarked over the considered class of problems, especially when the number of design parameters are great.

This dissertation also addresses several challenges in widespread application of the truss optimization methods in practice. Regarding the discussions in Section 1.2, these contributions can be summarized as follows:

- *Set-up time*: FSD-ES is a user-friendly ready-to-use specialized algorithm for simultaneous optimization of topology shape and size. For problems within this category, providing data for running the optimization algorithm takes from a few minutes for easy problems to a few hours for complicated ones, such as the 277-bar bridge design problem with 303 design parameters. Consequently, the required set-up effort to solve a new problem is small and no user-based tuning is required.
- *Constraints and cost function*: In the current format, FSD-ES II optimizes the structural weight and can handle conventional simple constrains, as well as the more practical specifications of AISC-ASD. The option of having more sophisticated cost functions is predictably and to some extent, easily, reachable considering the

flexibility of the evolution strategy in the upper level. Other constraints can also be handled by proper augmentation of the lower loop and the penalty term, following the same principles used for stress and displacement constraints.

- *Problem size*: It was demonstrated that FSD-ES II can optimize complicated and predictably large-scale problems within a reasonable amount of computation time, which mitigates one of the critical factors that prevents widespread application of truss optimization methods in practice.
- *Optimization gain:* The optimized design, especially when topology is considered, can hardly be reached by engineering intuition or try-and-error. The amount of material saving when FSD-ES or FSD-ES II is employed, compared to intuition-based methods, is potentially huge, considering that even the difference between the optimized solutions from different algorithms is significant for more complicated problems.

Despite the demonstrated superiority of FSD-ES over available truss optimization methods, it can still be further enhanced. For example:

- The lower loop of FSD-ES performs a rather exhaustive search in the given section list. It does not increase the number of evaluations, yet it slows than the algorithm. A faster search with a minimum compromise of the contribution of the resizing step can be advantageous.
- The evaluation budget of the lower level is one per design of the upper level. This fixed setting was motivated by contribution of the FSD concept, which is spectacular in the first iteration and diminishes fast in the subsequent iterations; Nevertheless, testing the lower loop with more evaluation budget, or even an adaptive scheme where the evaluation budget is updated iteratively is worth trying.

- Other types of constraints, such as natural frequency, can be considered, as some recent studies have done. The resizing step can take the frequency constraints into account by measuring the contribution of each member to the critical natural frequencies, similarly to the strategy followed by FSD-ES-II for handling displacement constraints. More sophisticated constraints associated with buildability constraints can be considered as well.
- Common factors that affect the overall cost can be simulated in the objective function, which has been overlooked in academic research. Such a realistic objective function can provide valuable contribution to this field of research.

REFERENCES

REFERENCES

[1] A. G. M. Michell, "LVIII. The limits of economy of material in frame-structures," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* , vol. 8, no. 47, pp. 589-597, 1904.

[2] B. H. V. Topping, "Shape optimization of skeletal structures: a review," *Journal of Structural Engineering,* vol. 109, no. 8, pp. 1933-1951, 1983.

[3] K. Mueller, M. Gustafson and J. Ericksen, "A Practicing Engineer's View of Benchmark Problems in Structural Optimization.," in *20th Analysis and Computation Specialty Conference* , 2012.

[4] A. Alimoradi, C. M. Foley and S. Pezeshk, "Benchmark Problems in Structural Design and Performance Optimization: Past, Present, and Future-Part I," in *State of the Art and Future Challenges in Structure*, 2010.

[5] R. Kicinger, T. Arciszewski and K. De Jong, "Evolutionary computation and structural design: A survey of the state-of-the-art," *Computers & Structures,* vol. 83, no. 23, pp. 1943-1978, 2005.

[6] S. A. Burns, Recent advances in optimal structural design (Chapter 5), ASCE Publications, 2002.

[7] American Institute of Steel Construction (AISC). Manual of Steel Construction: Allowable Stress Design, 1989.

[8] R. Balamurugan, C. V. Ramakrishnan and N. Swaminathan, "A two phase approach based on skeleton convergence and geometric variables for topology optimization using genetic algorithm," *Structural and Multidisciplinary Optimization,* vol. 43, no. 3, pp. 381-404, 2001.

[9] G. C. Luh and C. H. Chueh, "Multi-modal topological optimization of structure using immune algorithm," *Computer Methods in Applied Mechanics and Engineering,* vol. 193, no. 36, pp. 4035-4055, 2004.

[10] A. Kaveh, B. Hassani, S. Shojaee and S. M. Tavakkoli, "Structural topology optimization using ant colony methodology," *Engineering Structures,* vol. 30, no. 9, pp. 2559-2565, 2008.

[11] D. J. Munk, G. A. Vio and G. P. Steven, "Topology and shape optimization methods using evolutionary algorithms: a review," *Structural and Multidisciplinary Optimization,* p. in press, 2015.

[12] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer methods in applied mechanics and engineering,* vol. 71, no. 2, pp. 197-224, 1988.

[13] G. I. Rozvany, M. Zhou and T. Birker, "Generalized shape optimization without homogenization," *Structural Optimization,* vol. 4, no. 3-4, pp. 250-252, 1992.

[14] Y. M. Xie and G. P. Steven, "A simple evolutionary procedure for structural optimization," *Computers & structures,* vol. 49, no. 5, pp. 885-896, 1993.

[15] O. Sigmund, "On the usefulness of non-gradient approaches in topology optimization," *Structural and Multidisciplinary Optimization,* vol. 43, no. 5, pp. 589-596, 2011.

[16] E. W. Parkes, "Joints in optimum frameworks," *International Journal of Solids and Structures,* vol. 11, no. 9, pp. 1017-1022, 1975.

[17] H. J. Barbosa, A. C. Lemonge and C. C. Borges, "A genetic algorithm encoding for cardinality constraints and automatic variable linking in structural optimization," *Engineering Structures,* vol. 30, no. 12, pp. 3708-3723, 2008.

[18] A. Ahrari and A. A. Atai, "Fully Stressed Design Evolution Strategy for Shape and Size Optimization of Truss Structures," *Computers & Structures,* vol. 123, pp. 58-67, 2013.

[19] A. Ahrari, A. A. Atai and K. Deb, "Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy," *Engineering Optimization,* p. in press, 2014.

[20] G. Reddy and J. Cagan, "Optimally directed truss topology generation using shape annealing," Carnegie Mellon University, 1993.

[21] K. Shea and I. Smith, "Improving full-scale transmission tower design through topology and shape optimization," *Journal of structural engineering,* vol. 132, no. 5, pp. 781-790, 2006.

[22] A. Ahrari, H. Lei, M. A. Sharif, K. Deb and X. Tan, "Design optimization of artificial lateral Line system under uncertain conditions," in *Proc. of IEEE congress on Evolutionary Computation (CEC 2015),* Sendai, Japan, (to appear), 2015.

[23] L. N. Vicente and P. H. Calamai, "Bilevel and multilevel programming: A bibliography review," *Journal of Global optimization,* vol. 5, no. 3, pp. 291-306, 1994.

[24] R. Li, M. T. Emmerich, J. Eggermont, T. Bäck, M. Schütz, J. Dijkstra and J. H. Reiber, "Mixed integer evolution strategies for parameter optimization," *Evolutionary computation,* vol. 21, no. 1, pp. 29-64, 2013.

[25] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *IEEE International Conference on Neural Networks*, 1995.

[26] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671-680, 1983.

[27] M. Dorigo, "Optimization, Learning and Natural Algorithms," PhD thesis, Milano, Italie, 1992.

[28] K. Sörensen, "Metaheuristics—the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3-18, 2015.

[29] T. Bäck, U. Hammel and H. P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3-17, 1997.

[30] J. Chen, M. T. Emmerich, R. Li, J. Kok and T. Bäck, "How to Do Recombination in Evolution Strategies: An Empirical Study," in *Methods and Models in Artificial and Natural Computation. A Homage to Professor Mira's Scientific Legacy*, Berlin Heidelberg, Springer , 2009, pp. 223-232.

[31] O. M. Shir, C. Siedschlag, T. Bäck and M. J. Vrakking, "Niching in evolution strategies and its application to laser pulse shaping," in *In Artificial Evolution*, Berlin Heidelberg, Springer , 2006, pp. 85-96.

[32] T. Bäck, C. Foussette and P. Krause, Contemporary Evolution Strategies, Springer, 2013.

[33] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," KanGAL report, 2005005, 2005.

[34] A. Auger, S. Finck, N. Hansen and R. Ros, "Comparison Tables of All Algorithms on All Noiseless Functions (RT-0383)," 2010.

[35] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159-195, 2001.

[36] H. G. Beyer and H. P. Schwefel, "Evolution strategies–A comprehensive introduction," *Natural computing*, vol. 1, no. 1, pp. 3-52, 2002.

[37] D. V. Arnold, "Weighted multirecombination evolution strategies," *Theoretical computer science*, vol. 361, no. 1, pp. 18-37, 2006.

[38] S. H.-P. Bäck T, "Evolution strategies I: Variants and their computational implementation," in *Genetic algorithms in engineering and computer science*, Wiley, 1995, p. 111–126.

[39] A. Ahrari, M. R. Saadatmand, M. Shariat-Panahi and A. A. Atai, "On the limitations of classical benchmark functions for evaluating robustness of evolutionary algorithms," *Applied Mathematics and Computation*, vol. 215, no. 9, pp. 3222-3229, 2010.

[40] A. Ahrari and R. Ahrari, "On the utility of randomly generated functions for performance evaluation of evolutionary algorithms," *Optimization letters*, vol. 4, no. 4, pp. 531-541, 2010.

[41] R. Dębski, R. Dreżewski and M. Kisiel-Dorohinicki, "Maintaining Population Diversity in Evolution Strategy for Engineering Problems," in *New Frontiers in Applied Artificial Intelligence*, Berlin Heidelberg, Springer , 2008, pp. 379-387.

[42] E. Mezura-Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *International Journal of General Systems*, vol. 37, no. 4, pp. 443-473, 2008.

[43] H.-P. Schwefel, G. Rudolph and T. Bäck, "Contemporary evolution strategies," in *Advances in artificial life*, Berlin, Springer-Verlag, 1995, pp. 891-907.

[44] C. Horoba, T. Jansen and C. Zarges, "Maximal age in randomized search heuristics with aging," in *11th Annual conference on Genetic and evolutionary computation*, NY, 2009.

[45] Z. C. Jansen T, "Comparing different aging operators," in *Artificial Immune Systems*, Berlin Heidelberg, Springer, 2009, pp. 95-108.

[46] T. Jansen and C. Zarges, "On benefits and drawbacks of aging strategies for randomized search heuristics," *Theoretical Computer Science*, vol. 412, no. 6, pp. 543-559, 2011.

[47] T. Jansen and C. Zarges, "On the role of age diversity for effective aging operators," *Evolutionary Intelligence*, vol. 4, no. 2, pp. 99-125, 2001.

[48] A. Ahrari and O. Kramer, "Finite life span for improving the selection scheme in evolution strategies.," *Soft Computing*, p. in press, 2015.

[49] O. Kramer, "Evolutionary self-adaptation: a survey of operators and strategy parameters," *Evolutionary Intelligence*, vol. 3, no. 2, pp. 51-65, 2010.

[50] S. García, D. Molina, M. Lozano and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617-644., 2009.

[51] N. Hansen, A. Auger, R. Ros, S. Finck and P. Pošík, "Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009," in *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, 2010.

[52] S. Wessing, M. Preuss and G. Rudolph, "When parameter tuning actually is parameter control," in *13th Annual Conference on Genetic and Evolutionary Computation (GECCO'11)*, NY, 2011.

[53] A. Ahrari and M. Shariat-Panahi, "An improved evolution strategy with adaptive population size.," *Optimization*, p. in press, 2013.

[54] T. Liao, M. A. M. de Oca and T. Stützle, "Computational results for an automatically tuned CMA-ES with increasing population size on the CEC'05 benchmark set," *Soft Computing*, vol. 17, no. 6, pp. 1031-1046, 2013.

[55] G. Jastrebski and D. V. Arnold, "Improving evolution strategies through active covariance matrix adaptation. (pp. 2814-2821). IEEE.," in *IEEE Congress on Evolutionary Computation (CEC'06)*, 2006.

[56] H. G. Beyer and B. Sendhoff, "Covariance matrix adaptation revisited–the CMSA evolution strategy," in *Parallel Problem Solving from Nature–PPSN X* , Berlin Heidelberg, Springer, 2008, pp. 123-132.

[57] A. Ahrari, K. Deb and M. Preuss, "Multimodal Optimization by Covariance Matrix Self-Adaptation Evolution Strategy with Repelling Subpopulations," *Evolutionary Computation*, vol. in press, 2016.

[58] M. Papadrakakis, N. D. Lagaros, G. Thierauf and J. Cai, "Advanced solution methods in structural optimization based on evolution strategies," *Engineering Computations*, vol. 15, no. 1, pp. 12-34, 1998.

[59] G. Thierauf and J. Cai, "Parallelization of the evolution strategy for discrete structural optimization problems," *Computer-Aided Civil and Infrastructure Engineering*, vol. 13, no. 1, pp. 23-30, 1998.

[60] N. D. Lagaros, M. Papadrakakis and G. Kokossalakis, "Structural optimization using evolutionary algorithms," *Computers & structures*, vol. 80, no. 7, pp. 571-589, 2002.

[61] C. Ebenau, J. Rottschäfer and G. Thierauf, "An advanced evolutionary strategy with an adaptive penalty function for mixed-discrete structural optimisation," *Advances in Engineering Software*, vol. 36, no. 1, pp. 29-38, 2005.

[62] O. Hasançebi, S. D. E. Çarbaş, F. Erdal and M. P. Saka, "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures," *Computers & Structures,* vol. 87, no. 5, pp. 284-302, 2009.

[63] M. P. Saka, "Optimum design of pin-jointed steel structures with practical applications," *Journal of Structural Engineering,* vol. 116, no. 10, pp. 2599-2620, 1990.

[64] C. Feury and M. Geradin, "Optimality criteria and mathematical programming in structural weight optimization," *Computers & Structures,* vol. 8, no. 1, pp. 7-17, 1978.

[65] R. Razani, "Behavior of fully stressed design of structures and its relationshipto minimum-weight design," *AIAA Journal,* vol. 3, no. 12, pp. 2262-2268, 1965.

[66] M. Schevenels, S. McGinn, A. Rolvink and J. Coenders, "An optimality criteria based method for discrete design optimization taking into account buildability constraints," *Structural and Multidisciplinary Optimization,* p. in press, 2014.

[67] F. Flager, G. Soremekun, A. Adya, K. Shea, J. Haymaker and M. Fischer, "Fully constrained design: a general and scalable method for discrete member sizing optimization of steel truss structures," *Computers & Structures,* vol. 140, pp. 55-65, 2014.

[68] S. K. Azad, O. Hasançebi and M. P. Saka, "Guided stochastic search technique for discrete sizing optimization of steel trusses: A design-driven heuristic approach," *Computers & Structures,* vol. 134, pp. 62-74, 2014.

[69] S. K. Azad and O. Hasançebi, "Discrete sizing optimization of steel trusses under multiple displacement constraints and load cases using guided stochastic search technique," *Structural and Multidisciplinary Optimization,* vol. 52, p. 383–404, 2015.

[70] C. Fleury, "An efficient optimality criteria approach to the minimum weight design of elastic structures," *Computers & Structures,* vol. 11, no. 3, pp. 163-173, 1980.

[71] K. S. Lee, Z. W. Geem, S. Lee and K. Bae, "The harmony search heuristic algorithm for discrete structural optimization," *Engineering Optimization,* vol. 37, no. 7, pp. 663-684, 2005.

[72] S. O. Degertekin, "Improved harmony search algorithms for sizing optimization of truss structures," *Computers & structures,* vol. 92–93, p. 229–41, 2011.

[73] M. Sonmez, "Discrete optimum design of truss structures using artificial bee colony algorithm," *Structural and multidisciplinary optimization,* vol. 43, no. 1, pp. 85-97, 2011.

[74] Y. C. Lu, J. C. Jan, S. L. Hung and G. H. Hung, "Enhancing particle swarm optimization algorithm using two new strategies for optimizing design of truss structures," *Engineering Optimization,* vol. 45, no. 10, pp. 1251-1271, 2013.

[75] V. Toğan and A. T. Daloğlu, "An improved genetic algorithm with initial population strategy and self-adaptive member grouping," *Computers & Structures,* vol. 86, no. 11, pp. 1204-1218., 2008.

[76] a. Kaveh and B. T. S. Farhmand-Azar, "Ant colony optimization for design of space trusses," *International Journal of Space Structures,* vol. 23, no. 3, pp. 167-181, 2008.

[77] A. Kaveh, M. Kalateh-Ahani and M. S. Masoudi, "The CMA evolution strategy based size optimization of truss structures," *International journal of optimization in civil engineering,* vol. 2, pp. 233-256, 2011.

[78] O. Hasançebi, T. Teke and O. Pekcan, "A bat-inspired algorithm for structural optimization.," *Computers & Structures,* vol. 128, pp. 77-90., 2013.

[79] O. Hasançebi and S. K. Azad, "Adaptive dimensional search: A new metaheuristic algorithm for discrete truss sizing optimization," *Computers & Structures,* p. in press, 2015.

[80] A. Kaveh, B. Mirzaei and A. & Jafarvand, "An improved magnetic charged system search for optimization of truss structures with continuous and discrete variables. , 28,.," *Applied Soft Computing,* vol. 28, pp. 400-410, 2015.

[81] C. V. Camp and M. Farshchin, "Design of space trusses using modified teaching–learning based optimization," *Engineering Structures,* vol. 62, pp. 87-97, 2014.

[82] A. H. Gandomi, S. Talatahari, X. S. Yang and S. Deb, "Design optimization of truss structures using cuckoo search algorithm," *The Structural Design of Tall and Special Buildings,* vol. 22, no. 17, pp. 1330-1349, 2013.

[83] O. Hasançebi and S. Kazemzadeh Azad, "Discrete size optimization of steel trusses using a refined big bang–big crunch algorithm," *Engineering Optimization,* vol. 46, no. 1, pp. 61-83, 2014.

[84] A. Kaveh and V. R. Mahdavi, " A hybrid CBO–PSO algorithm for optimal design of truss structures with dynamic constraints," *Applied Soft Computing,* vol. 34, pp. 260-273, 2015.

[85] L. F. F. Miguel and L. F. F. Miguel, "Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms," *Expert Systems with Applications,* vol. 39, no. 10, pp. 9458-9467, 2012.

[86] A. Kaveh and M. Shahrouzi, "Simultaneous topology and size optimization of structures by genetic algorithm using minimal length chromosome," *Engineering Computations*, vol. 23, no. 6, pp. 644-674, 2006.

[87] A. Kaveh and A. Zolghadr, "Topology optimization of trusses considering static and dynamic constraints using the CSS," *Applied Soft Computing*, vol. 13, no. 5, pp. 2727-2734., 2013.

[88] N. Pholdee and S. Bureerat, "Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints," *Advances in Engineering Software*, vol. 75, pp. 1-13, 2014.

[89] A. Kaveh and A. Zolghadr, "Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints," *Advances in Engineering Software*, vol. 76, pp. 9-30, 2014.

[90] K. Deb and S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms," *Finite elements in analysis and design,* vol. 37, no. 5, pp. 447-465, 2001.

[91] H. Rahami, A. Kaveh and Y. Gholipour, "Sizing, geometry and topology optimization of trusses via force method and genetic algorithm," *2008*, vol. 30, no. 9, pp. 2360-2369, Engineering Structures.

[92] G. C. Luh and C. Y. Lin, "Optimal design of truss-structures using particle swarm optimization," *Computers & Structures,* vol. 89, no. 23, pp. 2221-2232, 2011.

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

[94] S. Gholizadeh, "Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization," *Computers & Structures*, vol. 125, pp. 86-99, 2013.

[95] A. Kaveh and B. Ahmadi, "Sizing, geometry and topology optimization of trusses using force method and supervised charged system search," *Structural Engineering and Mechanics*, vol. 50, no. 3, pp. 365-382 , 2014.

[96] O. Hasançebi and F. Erbatur, "Layout optimisation of trusses using simulated annealing," *Advances in Engineering Software,* vol. 33, no. 7, pp. 681-696, 2002.

[97] N. Noilublao and S. Bureerat, "Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multiobjective evolutionary algorithms," *Computers & Structures,*, vol. 89, no. 23, pp. 2531-2538, 2001.

[98] S. D. Rajan, "Sizing, shape, and topology design optimization of trusses using genetic algorithm," *Journal of Structural Engineering*, vol. 121, no. 10, pp. 1480-1487, 1995.

[99] M. Ohsaki, "Simultaneous optimization of topology and geometry of a regular plane truss," *Computers & structures*, vol. 66, no. 1, pp. 69-77, 1998.

[100] S. M. Shrestha and J. Ghaboussi, "Evolution of optimum structural shapes using genetic algorithm," *Journal of Structural Engineering*, vol. 124, no. 11, pp. 1331-1338, 1998.

[101] S. J. Wu and P. T. Chow, "Integrated discrete and configuration optimization of trusses using genetic algorithms. ,," *Computers & structures*, vol. 55, no. 4, pp. 695-702, 1995.

[102] O. Hasançebi and F. Erbatur, "Layout optimization of trusses using improved GA methodologies," *Acta mechanica*, vol. 146, no. 1-2, pp. 87-107, 2001.

[103] W. Tang, L. Tong and Y. Gu, " Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables," *International Journal for Numerical Methods in Engineering*, vol. 62, no. 13, pp. 1737-1762, 2005.

[104] G. C. Luh and C. Y. Lin, "Optimal design of truss structures using ant algorithm," *Structural and Multidisciplinary Optimization*, vol. 36, no. 4, pp. 365-379, 2008.

[105] C. Y. Wu and K. Y. Tseng, "Truss structure optimization using adaptive multi-population differential evolution," *Structural and Multidisciplinary Optimization*, vol. 42, no. 4, pp. 575-590, 2010.

[106] O. Hasançebi, "Optimization of truss bridges within a specified design domain using evolution strategies," *Engineering Optimization*, vol. 39, no. 6, pp. 737-756, 2007.

[107] O. Hasançebi, "Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures," *Computers & structures*, vol. 86, no. 1, pp. 119-132, 2008.

[108] A. Ahrari, "A requirement for the mutation operator in continuous optimization," *Optimization Letters*, vol. 7, no. 8, pp. 1681-1690, 2013.

[109] N. D. Lagaros, M. Papadrakakis and G. Kokossalakis, "Structural optimization using evolutionary algorithms," *Computers & structures*, vol. 80, no. 7, pp. 571-589, 2002.

[110] R. Ros and N. Hansen, "A simple modification in CMA-ES achieving linear time and space complexity," in *Parallel Problem Solving from Nature–PPSN X*, Berlin Heidelberg., Springer , 2008, pp. 296-305.

[111] W. Hare, J. Nutini and S. Tesfamariam, "A survey of non-gradient optimization methods in structural engineering," *Advances in Engineering Software,* vol. 59, pp. 19-28, 2013.

[112] N. Hansen, A. Auger, S. Finck and R. Ros, "Real-Parameter Black-Box Optimization Benchmarking: Experimental Setup," 2013.

[113] J. J. Liang, B. Qu, S. P. N. and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization (Technical Report, 201212)," China and Nanyang Technological University, , Singapor, 2013.

[114] J. J. Liang, B. Y. Qu, P. N. Suganthan and Q. Chen, "Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-based Real-Parameter Single Objective Optimization," Nanyang Technological University, Singapore, 2014.

[115] T. Y. Chen and H. C. Chen, "Discrete Structural Optimization by Using Evolution Strategy," in *6th European Solid Mechanics Conference ESMC,* 2006.

[116] O. Kramer, "A review of constraint-handling techniques for evolution strategies," *Applied Computational Intelligence and Soft Computing,* 2010.

[117] O. Kramer, "Premature convergence in constrained continuous search spaces," in *Parallel Problem Solving from Nature–PPSN X*, Heidelberg, Springer, 2008, pp. 62-71.

[118] A. Kaveh and S. Talatahari, "An enhanced charged system search for configuration optimization using the concept of fields of forces," *Structural and Multidisciplinary Optimization,* vol. 43, no. 3, pp. 339-351, 2011.

[119] L. Lamberti, "An efficient simulated annealing algorithm for design optimization of truss structures," *Computers & Structures,* vol. 86, no. 19, pp. 1936-1953, 2008.

[120] S. R. Hansen and G. N. Vanderplaats, "Approximation method for configuration optimization of trusses," *AIAA journal,* vol. 28, no. 1, pp. 161-168, 1990.

[121] O. Hasançebi and F. Erbatur, "On efficient use of simulated annealing in complex structural optimization problems," *Acta Mechanica,* vol. 157, no. 1-4, pp. 27-50, 2002.

[122] A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm. (Vol. 2, pp. 1777-1784). IEEE.," in *The 2005 IEEE Congress on Evolutionary Computation (CEC'05)*, 2005.

[123] N. Hansen, A. Auger, S. Finck and R. Ros, "Real-Parameter Black-Box Optimization Benchmarking 2010: Experimental Setup," INRIA, 2010.

[124] K. S. Lee, S. W. Han and Z. W. Geem, "Discrete size and discrete-continuous configuration optimization methods for truss structures using the harmony search algorithm," *Int. J. Optim. Civil Eng,* vol. 1, pp. 107-126, 2011.

[125] V. Ho-Huu, T. Nguyen-Thoi, M. H. Nguyen-Thoi and L. Le-Anh, "An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures," *Expert Systems with Applications,* vol. in press, 2015.

[126] N. Ali, K. Behdinan and Z. Fawaz, "Applicability and viability of a GA based finite element analysis architecture for structural design optimization," *Computers & structures,* vol. 81, no. 22, pp. 2259-2271, 2003.

[127] A. Ahrari and A. Atai, "Efficient simulation for optimization of topology, shape and size of modular truss structures," *International Journal of Optimization in Civil Engineering,* vol. 3, no. 2, pp. 209-223, 2013.

[128] E. Salajegheh and G. N. Vanderplaats, "Optimum design of trusses with discrete sizing and shape variables," *Structural Optimization,* vol. 6, no. 2, pp. 79-85, 1993.

[129] S. M. Shrestha and J. Ghaboussi, "Evolution of optimum structural shapes using genetic algorithm," *Journal of Structural Engineering,* vol. 124, no. 11, pp. 1331-1338, 1998.

[130] O. Hasançebi, "Optimization of truss bridges within a specified design domain using evolution strategies," *Engineering Optimization,* vol. 39, no. 6, pp. 737-756, 2007.

[131] Y. Y and C. K. Soh, "Automated optimum design of structures using genetic programming," *Computers & Structures,* vol. 80, no. 18-19, p. 1537–1546, 2002.

[132] Manual of Steel Construction: Allowable Stress Design, American Institute of Steel Construction (AISC), 1989.

[133] A. Ahrari and K. Deb, "An improved fully stressed design evolution strategy for layout optimization of truss structures," *Computers & Structures,* vol. 164, pp. 127-144, 2016.