CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Fully Constrained Design: a Scalable Method for Discrete Member Sizing Optimization of Steel Frame Structures

By

Forest Flager, Grant Soremekun, Akshay Adya, Kristina Shea, John Haymaker, and Martin Fischer

CIFE Technical Report #TR201
September 2011

STANFORD UNIVERSITY

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
The Jerry Yang & Akiko Yamazaki Environment & Energy Building
473 Via Ortega, Room 292, Mail Code: 4020
Stanford, CA 94305-4020

Fully constrained design: a general and scalable method for discrete member sizing optimization of steel frame structures

Forest Flager[a], Grant Soremekun[b], Akshay Adya[c], Kristina Shea[d], John Haymaker[e], Martin Fischer[f]

[a] Stanford University, CIFE, 473 Via Ortega, Room 292, Stanford, CA 94305-4020, United States, forest@stanford.edu (corresponding author)
[b] Phoenix Integration, Inc., 2921 Washington St. #6, San Francisco, CA 94115, United States, grant@phoenix-int.com
[c] Bentley Systems, 1600 Riviera Avenue Suite 300, Walnut Creek, CA 94596, United States, akshay.adya@alumni.stanford.edu
[d] ETH Zurich, Engineering Design and Computing Laboratory, CLA F 34.2, Tannenstrasse 3, 8092, Zurich, Switzerland, kshea@ethz.ch
[e] Schools of Architecture and Building Construction, Georgia Tech, 280 Ferst Dr., First Floor, Atlanta, GA 30332, United States, johnrhaymaker@gmail.com
[f] Stanford University, CIFE, 473 Via Ortega, Room 297, Stanford, CA 94305-4020, United States, fischer@stanford.edu

ABSTRACT

Fully Constrained Design (FCD) is a new method for discrete sizing optimization of steel frames. Based on the optimality criteria approach, FCD handles constraints and generates designs in a new way that enables it to be readily applied to different problem formulations, even when the search space is discontinuous. The quality of solution produced by the proposed method is superior to optimality criteria (>7% difference) and comparable to leading heuristic methods (<2% difference), based on the benchmarking

studies we conducted. We present a successful industry application of FCD that yields cost savings of 19% compared to conventional design methods.

Keywords: structural optimization; size optimization; discrete variables; steel structures; frame; truss

1. Introduction

Engineers are often challenged to design steel structures that use the least amount of economic and environmental resources possible to satisfy the system's functional requirements. The design of these structures can be decomposed into three components: (i) *topology*, which concerns the number and connectivity of members; (ii) *shape*, which pertains to the location of structural joints; and (iii) *sizing*, which involves defining member cross-sections [1]. This paper presents a flexible, general, and scalable algorithm to optimize the sizing of steel members given a fixed topology and shape. The objective of the optimization process is to minimize the cost of the structure while satisfying design performance requirements for safety and serviceability. In this case, the total weight of the structure is used to estimate cost. Steel weight is commonly used as a surrogate for cost in the structural design industry and has been demonstrated to be accurate, provided that the construction methods required do not become too expensive or impractical [2]. To achieve this objective, engineers select steel sections from a discrete set which contains certain designations of steel profiles that are produced by steel mills [3].

Member sizing optimization is traditionally an iterative process that is performed manually by the engineer. The number of possible design alternatives (i.e., the design space) for sizing problems is an exponential function of the number of design variables and the number of possible choices for each variable. Even for a relatively simple 10-bar truss problem as described in Section 4.1, the number of possible sizing configurations is greater than 1.0E+10. Finding optimum designs within such a large design space using manual methods is very difficult. Often engineers leave vast areas of the design space unexplored that potentially contain better performing design configurations [4, 5].

Optimization algorithms enable engineers to leverage computer processing power to systematically search the design space for optimal member size configurations. Researchers have developed and applied a variety of optimization algorithms to discrete sizing problems for steel truss and frame structures over the past 50 years as surveyed by Arora [6]. These algorithms can be broadly categorized as deterministic or non-deterministic. Deterministic methods such as mathematical programming [7-11] and optimality criteria [12-14] were first applied to discrete sizing problems in the 1960s. These algorithms need an initial design configuration to begin the search and require gradient computations in the exploration process, namely the calculation of the first derivative of the objective and constraint functions with respect to the design variables. In some cases, the objective and/or constraint functions are discontinuous or irregular, making the gradient search difficult [15]. In addition, the constraint functions may vary depending on local regulatory requirements and stakeholder preferences [16], thus requiring the customization of the algorithm for each unique set of constraint functions. The implementation of the algorithm can be time consuming and error prone in such cases [17].

Another group of optimization techniques that have emerged recently do not require gradient information for the objective and constraint functions and use probabilistic transition rules rather than deterministic ones. The basic idea behind these stochastic techniques is to simulate a natural phenomenon, such as survival of the fittest, the immune system, swarm intelligence and the cooling process of molten metal through annealing. A detailed review of these algorithms as well as a comparison of their performance for discrete sizing problems is provided by Hasancebi [18, 19]. These heuristic search and optimization methods have a couple of advantages when compared to the deterministic methods discussed above. First, they separate domain knowledge from search, making them generally applicable to a wide variety of problem formulations without customization. Second, there is no limitation on the continuity of the search space since no gradient information is required.

A disadvantage of heuristic methods, however, is that they require significantly more computational resources than deterministic techniques [20]. Research on the convergence of these algorithms has shown that the number of evaluations required to reach a given solution quality grows as a function of the square root of the size of the problem [21]. To keep computation times manageable, researchers have focused on applying heuristic methods to truss and frame structures involving fewer than 100 sizing variables. Further research is required to compare the performance of these methods to deterministic techniques for large-scale member sizing problems involving hundreds or even thousands of variables which are common in industry practice.

The goal of the research presented in this paper was to develop a discrete member sizing optimization method that is (i) flexible (i.e., can accommodate different objective and constraint functions without modification); (ii) general (i.e., does not require the search space to be continuous) and (iii) scalable (i.e., can be applied to large structures involving greater than 100 sizing variables in a time frame that is at least comparable to conventional design practice). To achieve these objectives, the proposed optimization algorithm, which we call the Fully Constrained Design method, employs a new way of handling constraints and generating new designs that is presented in Section 3. We benchmark the method against the best performing existing deterministic and heuristic optimization methods in Section 4. In Section 5, we benchmark the method against conventional industry practice on a large stadium roof structure to demonstrate the scalability of the method. Finally, we summarize the benchmarking results and discuss the suitability of the method for general industry application in Section 6.

2. Mathematical model for discrete sizing optimization introduction

A general discrete sizing structural optimization problem can be formulated as:

Minimize: $W = f(x^1, x^2, \dots, x^d), \;\; d = 1,2,\dots,D$ (1)

Satisfying: $G_q = f(x^1, x^2, \dots, x^d) \leq 1, \;\; d = 1,2,\dots,D$ and $q = 1,2,\dots,M$ (2)

$$x^n \in S_n \left\{X_1, X_2, \dots, X_p\right\} \quad (3)$$

Where W is the weight of the structure, which is a scalar function. The set of design variables are represented as $x^1, x^2, \dots, x^d$. The design variable x^n belongs to the set S_n, which describes the available list of discrete member section values. The inequality $G \leq 1$ represents the constraint functions, which must be less than unity in this case. The structural constraints considered in the numerical examples in Section 4 include member stresses and nodal displacements. The letters D and M are the number of design variables and constraint functions, respectively. The letter p is the number of available section size choices for a given design variable.

3. Fully constrained design method

3.1. Description

The Fully Constrained Design (FCD) method for member sizing optimization is based on the optimality criteria approach discussed in Section 1. FCD possesses a new approach to constraint handling and the generation of new designs that overcomes the observed limitations to the flexibility and generality of the optimality criteria method, namely (1) the requirement that the objective and constraint functions are continuously differentiable in terms of the design variables, and (2) the requirement that the algorithm be customized for each unique problem formulation.

The proposed method does not require gradient information. It involves creating a one-to-one mapping between each member size design variable and a governing constraint. Based on the value of the governing constraint, the section size of each member variable is adjusted incrementally from an ordered list of choices.

Figure 1 provides an overview of the FCD process. Steps 1-5 are identical to the optimality criteria method; steps 6-10 are unique. Each process step is described in more detail below.

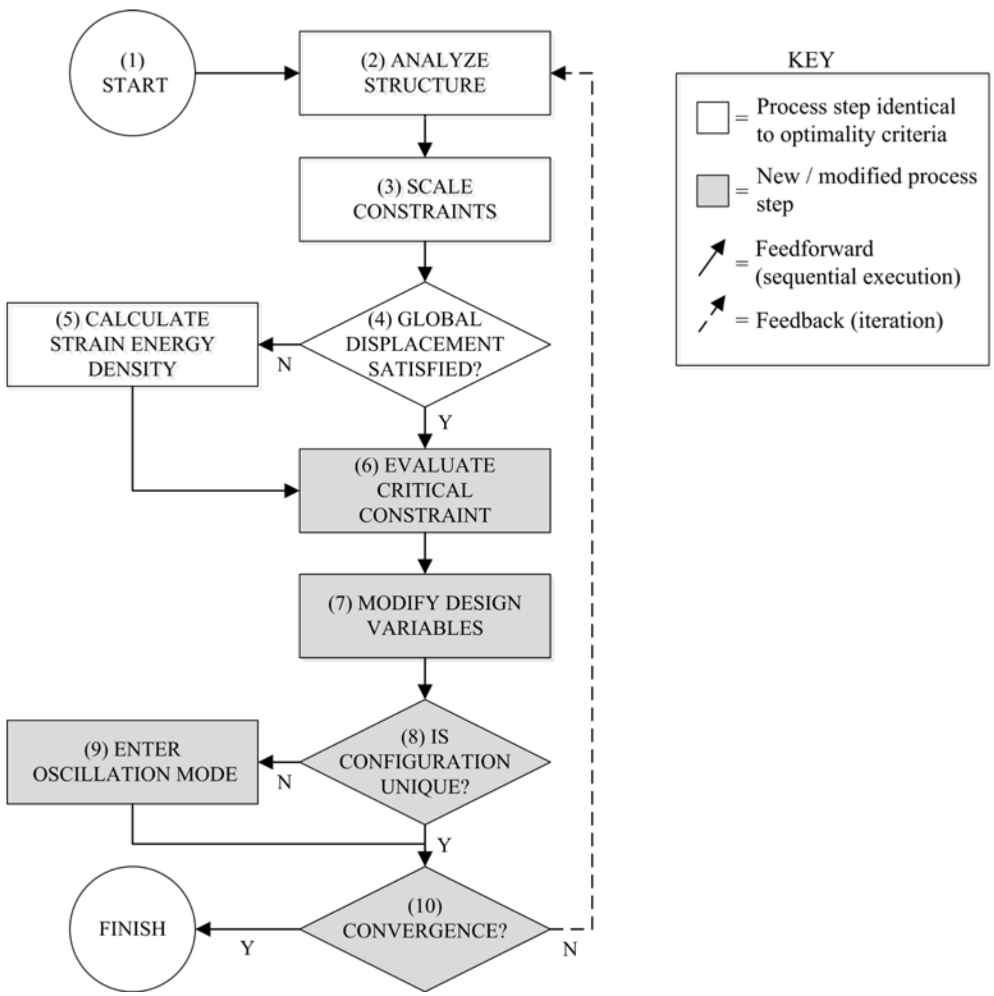

Figure 1: Overview of Fully Constrained Design (FCD) method

Step 1 – Start: The optimization process begins with the creation of an analytical model that contains an initial configuration of member sizes. This initial configuration of member sizes can either be chosen at random or be based on a previous design solution.

Step 2 – Analyze structure: The analytical model is used to calculate the structure's response to the defined loading. The responses calculated in the numerical examples discussed in Section 4 include the maximum stress (σ_{max}^{n}) for each member, the maximum deflection (Δ_{max}^{n}) for each member, and the global deflection (Δ_{Gmax}), considering all of the members in the structure. The value of the objective function, total steel weight (W) in this case, is also calculated.

Step 3 – Scale constraints: The structural responses calculated in the previous step are then normalized to unity based on the allowable values for each design constraint, as specified in the problem formulation. The constraint function for stress (G_{σ}) therefore can be expressed as follows:

$$G_{\sigma} = g_{\sigma}(x_{\sigma}^{1}, x_{\sigma}^{2}, x_{\sigma}^{3}, \ldots, x_{\sigma}^{d}) \leq 1, \text{ where } x_{\sigma}^{n} = \sigma_{max}^{n}/\sigma_{allow}^{n} \tag{4}$$

Similarly, for member deflection ($G_{\Delta m}$):

$$G_{\Delta m} = g_{\Delta m}\left(x_{\Delta m}^{1}, x_{\Delta m}^{2}, x_{\Delta m}^{3}, \ldots, x_{\Delta m}^{d}\right) \leq 1, \text{ where } x_{\Delta m}^{n} = \Delta_{max}^{n}/\Delta_{allow}^{n} \tag{5}$$

Finally, the normalized global displacement (Δ_{Gn}) scalar is calculated as follows:

$$\Delta_{Gn} = \Delta_{Gmax}/\Delta_{Gallow} \leq 1 \tag{6}$$

Step 4 – Global displacement satisfied? If the normalized global displacement constraint is satisfied ($\Delta_{Gn} \leq 1$), it is unnecessary to calculate strain energy density as described in Step 5. If the global displacement constraint is violated ($\Delta_{Gn} > 1$), Step 5 is required.

Step 5 – Calculate strain energy density: In cases where the global displacement constraint is violated ($\Delta_{Gn} > 1$), member strain energy density is used to map this scalar function to each design variable. The maximum strain energy density for each member is calculated for a single load case in which a unit displacement is applied to the particular node in the structure where the maximum global displacement is observed. The strain energy density for each member (SED^n) is then normalized by the member with the maximum strain energy density (SED_{max}). Finally, the normalized strain energy density values are multiplied by the normalized global deflection scalar to calculate the constraint function for global displacement ($G_{\Delta g}$) as described in Eq. (7).

$$G_{\Delta g} = g_{\Delta g}\left(x_{\Delta g}^1, x_{\Delta g}^2, \dots, x_{\Delta g}^d\right), \text{ where } x_{\Delta g}^n = \Delta_{Gn}(SED^n/SED_{max}) \qquad (7)$$

Step 6 – Evaluate critical constraint: Once all of the constraint functions have been calculated, the values must be compared to identify the critical constraint for each design variable. From Eq. 2, the inequality $G \leq 1$ represents the constraint functions where the letter D is the number of design variables and the letter M is the number of constraint functions (e.g., stress, global displacement). The critical constraint function (G_c) is calculated by comparing the different constraint values for each design variable and taking the maximum:

$$G_c = g_c(x_c^1, x_c^2, \dots, x_c^d), \text{ where } x_c^n = \max_d[x_1^n, x_2^n \dots x_M^n] \quad (8)$$

Step 7 – Modify design variables: Each design variable $(x^1, x^2, \dots, x^d)$ has a corresponding set of possible values $(S_1, S_2, \dots, S_d)$, which describes the available list of discrete member section values for each variable as described in Eq. 3. These sets $(S_1, S_2, \dots, S_d)$ are ordered by section area from low to high. New designs are generated iteratively by adjusting the variable values up or down the corresponding ordered list of section sizes based on the critical constraint value calculated for each design variable. The design variables are organized into three groups according to their critical constraint value. This grouping determines the sequence in which the variables are modified as well as the increment as described in Table 1.

Variable group name	Critical constraint range (% of allowable)	Rank	Discrete section size increment
Margin Range 1	0-90%	2	X_{n-1}
Constant Range	90-100%	3	X_n
Violation Range 1	100%-inf.	1	X_{n+1}

Table 1: FCD process for modifying design variables based on critical constraint values

The discrete section size increment determines how the variable values in a particular group are to be adjusted. If the critical constraint is in the violation range (i.e., greater than allowable) the section with the next largest area is selected for the next iteration. If the critical constraint is in the margin range (i.e., less than allowable) the section with the next smallest area is selected. The rank of the variable group determines the sequence of adjustment. Variables with a lower rank are adjusted first. The algorithm continues to adjust the variable values in a particular group until there are no longer any variables in that group. The rationale for prioritizing the adjustment of the variable group in the violation range is to identify a feasible design configuration (i.e., set of design variable values that satisfy all of the problem constraints) with as little iteration as possible.

Step 8 – Is configuration unique? If the current design configuration (i.e., set of design variable values) is unique, the process proceeds to Step 10. If the current design configuration is identical to a previous iteration of the optimization process, the optimizer enters 'oscillation mode', which is discussed in the following step.

Step 9 – Enter oscillation mode: If a repeated design configuration is detected, the oscillation mode perturbs individual design variables to avoid an infinite loop of repeated configurations. This is achieved by first reverting back to the 'best' sizing configuration (i.e., the least weight configuration that satisfies the constraints), considering all previous iterations. Next, the design configuration is adjusted using similar logic to that described in Table 1. The only difference compared to Step 7 is that a single design variable is adjusted per iteration rather than an entire group of variables. The variable with a critical constraint value that is farthest from the allowable limit in terms of absolute value is adjusted. Oscillation mode continues until an improved design configuration is found or the convergence criteria described in Step 10 are met.

Step 10 – Convergence? The optimization process is concluded in one of four possible ways: (a) a fully constrained design is achieved, meaning that all of the design variables have critical constraint values in the Constant Range as described in Table 1; (b) a local / global optimum is reached, meaning that the optimizer manipulates all of the variables while in oscillation mode and is unable to find an improved design; (c) the number of iterations without improvement specified by the user is met; or (d) the maximum number of iterations specified by the user is met.

3.2. Implementation

The proposed method was implemented in ModelCenter® [22], a commercial software package. It allows users to bring commercial or proprietary software tools into a common environment using software “wrappers” or “plug-ins”. Four software components were created in ModelCenter as shown on the diagonal in Figure 2. The four components are described in more detail below.

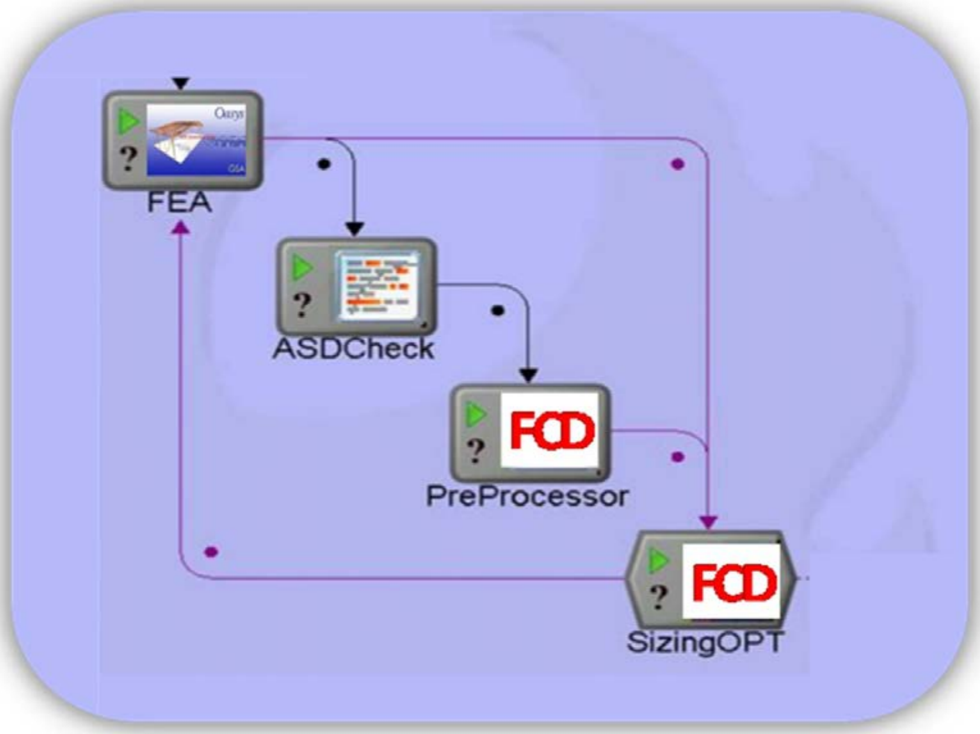

Figure 2: ModelCenter® interface showing the implementation of the FCD process. Arrows above the diagonal represent data dependencies for sequential execution while arrows below the diagonal represent iteration.

Step 1 – FEA: (i) Reads an existing Finite Element Analysis (FEA) model and allows the user to specify the desired design variables and the corresponding discrete set of candidate section sizes for each variable; and (ii) executes the FEA and stores the desired structural responses (e.g., deflections, member forces and moments). See Section 3.1, step 2.

Step 2 – ASD Check: Calculates the strength utilization ratio for each member based on the applicable building code. A utilization ratio of less than unity indicates that the strength of the member is adequate for the defined loading. See Section 3.1, step 2.

Step 3 – PreProcessor: (i) Scales each constraint type (e.g., strength utilization, member deflection, global deflection) to unity based on the allowable value; (ii) determines if the global displacement constraint is satisfied; and (iii) calculates the critical constraint for each design variable. See Section 3.1, steps 3-6.

Step 4 – SizingOPT: (i) Modifies the design variables based on the critical constraint values; (ii) checks whether the design configuration is unique and enters 'oscillation mode' if necessary; and (iii) concludes the optimization process if the convergence criteria have been met. See Section 3.1, steps 7-10.

4. Numerical examples

Three standard member sizing optimization problems are used to benchmark the performance of the FCD method: a 10-bar truss, a 25-bar truss and a 200-bar truss. The objective of each problem is to minimize the total steel weight of the structure while satisfying local stress and global displacement constraints. FCD is compared to other methods in terms of solution quality and computational efficiency. Solution quality is measured in terms of the total steel weight of the lightest design configuration that satisfies the design constraints. Computational efficiency is measured in terms of the number of finite element analyses required to arrive at the 'optimal' solution. The "leading heuristic methods" listed in Table 6, Table 7, Table 11, and Table 14 refer to the methods that have the highest solution quality of the algorithms surveyed for that particular numerical example.

The FCD results are based on conducting six different optimization runs for each problem using different initial design configurations. Each starting point is described in Table 2.

Start Point	Description of design variable values
1	uniform: smallest section area
2	uniform: largest section area
3	uniform: median section area
4	mixed: smallest and largest section areas
5	mixed: smallest and median section areas
6	mixed: median and largest section areas

Table 2: Initial variable configurations used by FCD for the numerical examples

4.1. 10-bar truss

The 10-bar truss geometry is shown in Figure 3. A single load case is applied to the structure as described in Table 3. The members are subjected to a stress limitation of ±25 ksi, and a displacement limitation of 2.0 in. is imposed at each node in both directions. These design constraints as well as the material properties are summarized in Table 4. There are 10 independent design variables in the problem corresponding to the cross-sectional area for each structural member. In this example, variable values must be selected from one of two discrete sets, which are enumerated in Table 5.

Several heuristic methods have been applied to this problem, including genetic algorithms [2, 3, 20, 23, 24], evolutionary strategies [25, 26], and heuristic particle swarm optimization [27]. In addition, benchmarking studies using SODA [28] have been conducted. SODA is a design and engineering software that utilizes the optimality criteria method to perform discrete sizing optimization. No additional benchmarking results involving discrete design variables could be found in the literature for the optimality criteria method. Therefore, the continuous solutions presented [29, 30] have been rounded to the nearest discrete section of equal or larger area to facilitate comparison. The benchmarking results are reported in Table 6 and Table 7 for Cases 1 and 2, respectively.

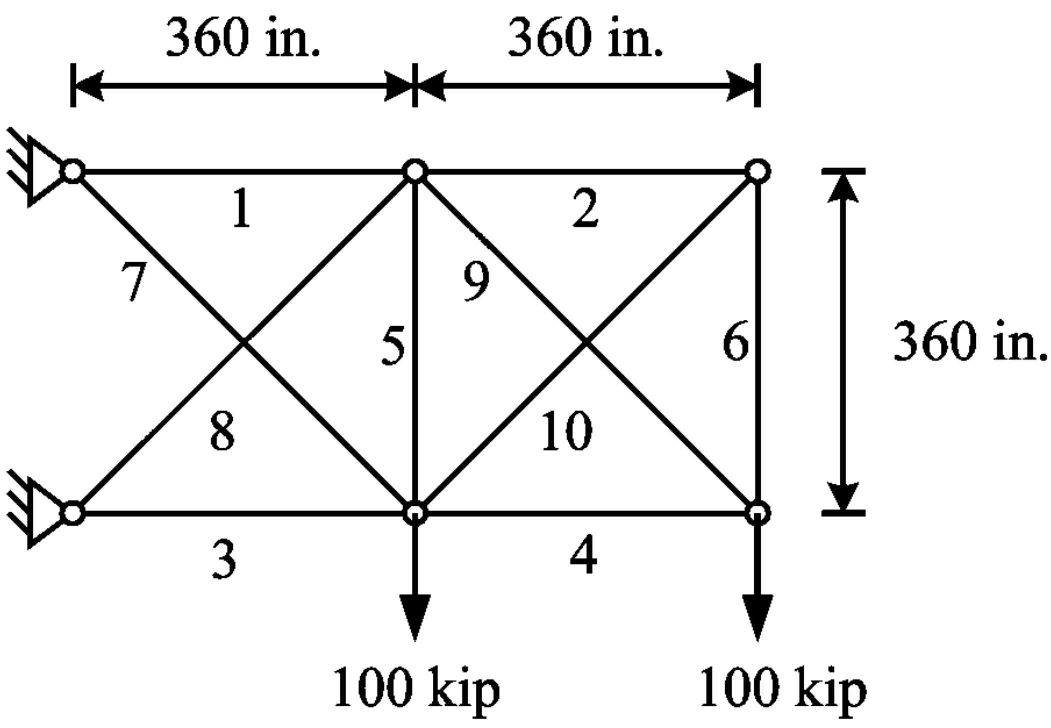

Figure 3: 10-bar truss [2]

Name	Magnitude (kips)	Direction	Nodes
Case 1	-100	y-axis	2,4

Table 3: 10-bar truss loading

Material properties	
Density (lbs/in^3)	0.1
Modulus of elasticity (ksi)	10,000
Constraints	
Allowable tensile stress (ksi)	25
Allowable compressive stress (ksi)	-25
Maximum displacement (in)	2

Table 4: 10-bar truss design parameters

Case	Reference	Set of cross sectional areas (in^2)
1	[26]	{0.100, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170, 19.180, 23.680, 28.080, 33.700}
2	[31]	{1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50}

Table 5: 10-bar truss discrete section set

Reference	LEADING HEURISTIC METHODS			OPTIMALITY CRITERIA			FCD
	[26]	[25]	[24]	[29]	[30]	[28]	(this study)
Weight (lb)	5,153	5,100	5,046	5,734 (5,060)	5,716 (5,067)	5,356	5,109
Max. defl. (in)	2.00	-	2.00	-	2.00	1.97	2.01
Num analyses	4,000	-	30,000	14	18	6	461
Variable	**Optimal area (in^2)**						
A_1	33.700	-	28.080	33.700 (30.520)	33.700 (30.980)	33.700	33.700
A_2	0.100	-	0.100	0.100 (0.100)	0.100 (0.100)	0.347	0.100
A_3	23.680	-	23.680	28.080 (23.200)	28.080 (24.170)	19.180	23.680
A_4	14.290	-	17.170	17.170 (15.220)	17.170 (14.810)	19.180	13.330
A_5	0.347	-	0.100	0.100 (0.100)	0.100 (0.100)	0.347	0.100
A_6	0.100	-	0.100	0.954 (0.550)	0.440 (0.410)	0.539	0.100
A_7	7.192	-	7.192	8.525 (7.460)	8.525 (7.550)	10.850	7.192
A_8	19.180	-	19.180	23.680 (21.040)	23.680 (21.050)	23.000	19.180
A_9	23.680	-	23.680	23.680 (21.530)	23.680 (20.940)	19.180	23.680
A_{10}	0.100	-	0.100	0.100 (0.100)	0.100 (0.100)	0.347	0.100

Table 6: 10-bar truss results – case 1. Continuous solutions are shown in parenthesis

Reference						OPTIMALITY CRITERIA	FCD
	LEADING HEURISTIC METHODS						
	[8]	[27]	[31]	[24]	[20]	[28]	(this study)
Weight (lb)	5,557	5,532	5,499	5,480	5,448	5,760	5,559
Max. defl. (in)	-	-	-	2.00	-	1.95	1.99
Num analyses	-	50,000	-	30,000	40,000	5	94
Variable	Optimal area (in^2)						
A_1	30.00	30.00	33.50	33.50	33.50	30.00	33.50
A_2	1.62	1.62	1.62	1.62	1.62	3.13	1.62
A_3	26.50	22.90	22.90	22.90	22.00	30.00	26.50
A_4	13.50	13.50	15.50	13.90	13.90	13.50	14.20
A_5	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A_6	1.62	1.62	1.62	1.62	1.62	3.13	1.62
A_7	7.22	7.97	7.22	7.97	7.97	13.50	11.50
A_8	22.90	26.50	22.90	22.90	22.90	18.80	19.19
A_9	22.00	22.00	22.00	22.00	22.90	18.80	19.19
A_{10}	1.62	1.80	1.62	1.62	1.62	4.49	1.99

Table 7: 10-bar truss results – case 2

4.2. 25-bar truss

The next example is a 25-bar space truss as shown in Figure 4. The loading for the structure is summarized in Table 8. The members are subjected to a stress limitation of ±40 ksi and nodes 1 and 2 are limited to a maximum displacement of 0.35 in. The design constraints and material properties are summarized in Table 9.

The structural members are aggregated into eight groups, making the structure doubly symmetric about the X and Y axes. All members constituent to a particular group must assume the same variable value. The discrete cross-sectional area values for the design variables are enumerated in Table 10. The benchmarking results are shown in Table 11.

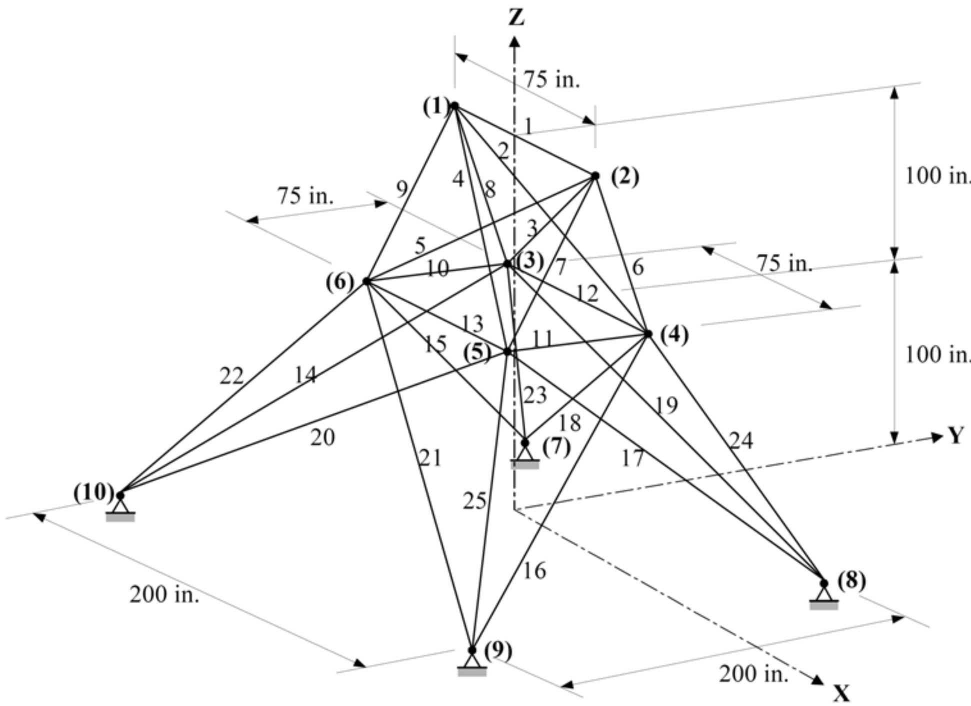

Figure 4: 25-bar truss [32]

Name	Magnitude (kips)	Direction	Nodes
Case 1	-1	y-axis	1,2
Case 1	-1	z-axis	1,2
Case 1	1	x-axis	1
Case 1	0.5	x-axis	3
Case 1	0.6	x-axis	6

Table 8: 25-bar truss loading

Material properties	
Density (lbs/in^3)	0.1
Modulus of elasticity (ksi)	10,000
Constraints	
Allowable tensile stress (ksi)	40
Allowable compressive stress (ksi)	-40
Maximum displacement (in)	0.35 (nodes 1,2)

Table 9: 25-bar truss design parameters

Case	Reference	Set of cross sectional areas (in^2)
1	[31]	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.2, 3.4}

Table 10: 25-bar discrete section set

Reference		LEADING HEURISTIC METHODS				OPTIMALITY CRITERIA	FCD
		[33]	[32]	[27]	[24]	[28]	(this study)
Total weight (lb)		493.8	484.9	484.9	483.4	562.9	526.8
Max. defl. (in)		-	0.351	-	0.351	0.342	0.3369
Num analyses		-	13,523	25,000	17,500	4	45
Variable	Constituent members	Optimal area (in^2)					
A_1	1	0.1	0.1	0.1	0.1	0.1	0.1
A_2	2-5	1.2	0.3	0.3	0.3	1.9	0.1
A_3	6-9	3.2	3.4	3.4	3.4	2.6	3.4
A_4	10,11	0.1	0.1	0.1	0.1	0.1	0.1
A_5	12,13	1.1	2.1	2.1	2.0	0.1	0.1
A_6	14-17	0.9	1.0	1.0	1.0	0.8	0.8
A_7	18-21	0.4	0.5	0.5	0.5	2.1	2.5
A_8	22-25	3.4	3.4	3.4	3.4	2.6	2.5

Table 11: 25-bar truss results

4.3. 200-bar truss

The final benchmarking problem is the 200-bar plane truss shown in Figure 5. The truss is subjected to three loading conditions as described in Table 12. The design parameters are summarized in Table 13. For this problem, the members are aggregated into 29 groups as enumerated in Table 14 and subjected to a stress limitation of ±10 ksi. The discrete section choices can be found in Table 5, Case 1. The benchmarking results are shown in Table 14.

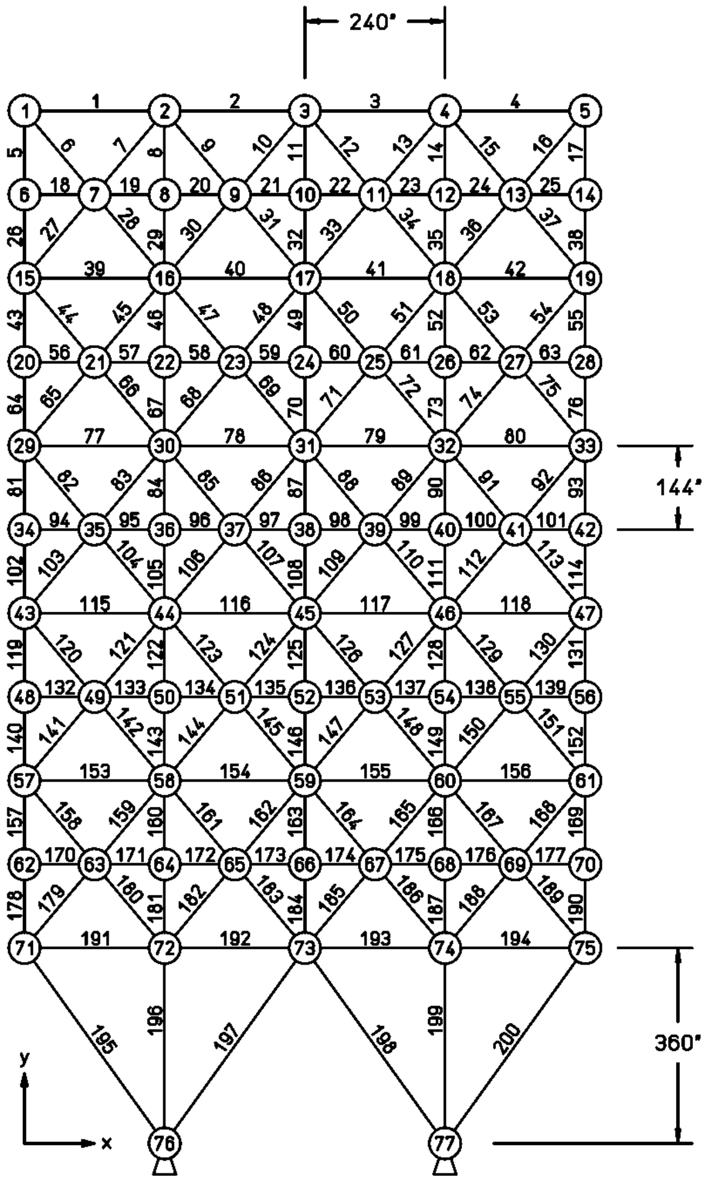

Figure 5: 200-bar truss [25]

Name	Magnitude (kips)	Direction	Nodes
Case 1	1	x-axis	1,2
Case 2	-10	y-axis	1,2
Case 3	Case 1 + Case 2		

Table 12: 200-bar truss loading

Material properties	Value
Density (lbs/in^3)	0.283
Modulus of elasticity (ksi)	30,000
Constraints	**Value**
Allowable tensile stress (ksi)	10
Allowable comp. stress (ksi)	-10
Maximum displacement (in)	N/A

Table 13: 200-bar truss design parameters

Reference		LEADING HEURISTIC METHOD [24]	OPTIMALITY CRITERIA [28]	FCD (this study)
Total weight (lb)		28,544	28,806	27,151
Num analyses		51,360	7	327
Variable	Constituent members	Optimal area (in^2)		
A_1	1,2,3,4	0.347	0.347	0.100
A_2	5,8,11,14,17	1.081	0.954	0.954
A_3	19,20,21,22,23,24	0.100	0.347	0.100
A_4	18,25,56,63,94,101, 132,139,170,177	0.100	0.100	0.100
A_5	26,29,32,35,38	2.142	2.142	2.142
A_6	6,7,9,10,12,13,15,16, 27,28,30,31,33,34,36, 37	0.347	0.440	0.347
A_7	39,40,41,42	0.100	0.347	0.100
A_8	43,46,49,52,55	3.565	3.131	3.131
A_9	57,58,59,60,61,62	0.347	0.347	0.100
A_{10}	64,67,70,73,76	4.805	4.805	4.805
A_{11}	44,45,47,48,50,51,53, 54,65,66,68,69,71,72, 74,75	0.440	0.954	0.440
A_{12}	77,78,79,80	0.440	0.347	0.347
A_{13}	81,84,87,90,93	5.952	5.952	5.952
A_{14}	95,96,97,98,99,100	0.347	0.347	0.347
A_{15}	102,105,108,111,114	6.572	6.572	6.572
A_{16}	82,83,85,86,88,89,91, 92,103,104,106,107, 109,110,112,113	0.954	0.954	0.954
A_{17}	115,116,117,118	0.347	0.347	0.347

A_{18}	119,122,125,128,131	8.525	8.525	8.525
A_{19}	133,134,135,136,137, 138	0.100	0.347	0.100
A_{20}	140,143,146,149,152	9.300	10.850	9.300
A_{21}	120,121,123,124,126, 127,129,130,141,142, 144,145,147,148,150, 151	0.954	1.081	1.081
A_{22}	153,154,155,156	1.764	9.300	0.347
A_{23}	157,160,163,166,169	13.330	13.330	13.330
A_{24}	171,172,173,174,175, 176	0.347	0.954	0.954
A_{25}	178,181,184,187,190	13.330	13.330	13.330
A_{26}	158,159,161,162,164,1 65,167,168,179,180, 182,183,185,186,188, 189	2.142	2.142	1.764
A_{27}	191,192,193,194	4.805	3.565	3.813
A_{28}	195,197,198,200	9.300	8.525	8.525
A_{29}	196,199	17.170	17.170	17.170

Table 14: 200-bar truss results

4.4. Discussion of numerical results

The results of these numerical examples indicate that the computational efficiency of the FCD method falls between that of the leading deterministic and heuristic methods. Generally, the number of analyses required for convergence of the optimization process is on the order of 100 for the proposed method, compared to 10 for optimality criteria and 10,000 for heuristic methods such as genetic algorithms (Figure 6). Similar to optimality criteria, the scale of the problem does not have a significant impact on the efficiency of the method.

With regard to solution quality, the best designs produced by the FCD method are comparable to leading heuristic methods (<2% difference) and superior to optimality criteria (>7% difference) based on averaging the numerical example results (Table 15). The data also suggest that deterministic methods such as optimality criteria and FCD perform relatively better as the scale of the problem increases.

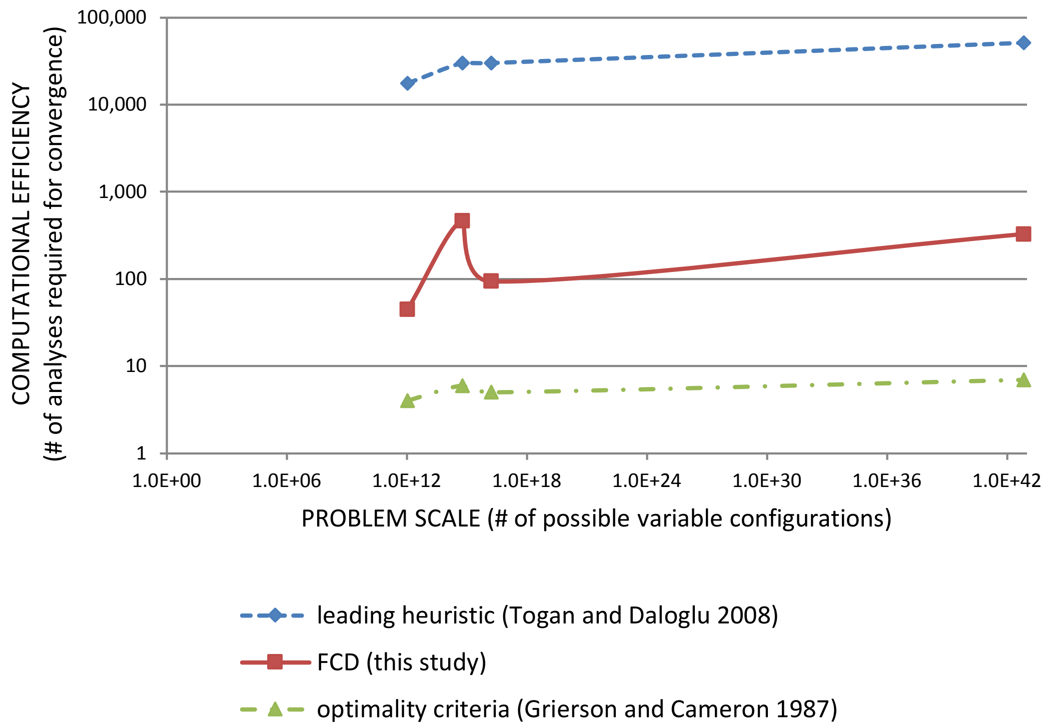

Figure 6: Comparison of computational efficiency by method for numerical examples

Numerical Example	Problem Scale (number of possible combinations)	LEADING HEURISTIC [24] Total weight (lb)	OPTIMALITY CRITERIA [28] Weight difference (% of total)	FCD (this study) Weight difference (% of total)
25-bar truss	1.10E+12	483.4	+16.4	+9.0
10-bar truss: case 1	5.90E+14	5046	+6.1%	+1.2%
10-bar truss: case 1	1.71E+16	5480	+5.1%	+1.4%
200-bar truss	6.86E+42	28,544	+0.9%	-4.9%
	Avg. weight difference	-	+7.2%	+1.7%

Table 15: Comparison of solution quality by method for numerical examples

5. Case study: stadium space frame roof structure

5.1. Background

The roof structure of a 65,000 seat athletics stadium was selected to test the scalability of the FCD method and to compare the performance of the method to the results achieved in conventional industry practice. The roof structure consists of two identical arched steel space frames that cover the north and south main stands for the stadium (Figure 7). The arches span 210 m (689 ft) and reach a maximum height of 72 m (236 ft). Two design methods were applied in parallel to optimize the member sizing configuration for the space frames: (1) the conventional design method of a leading engineering firm and (2) the FCD optimization method. The implementation of the FCD method is described in Section 3.2 and the implementation of the conventional method is explained below. The results for each method are then compared in terms of solution quality and process efficiency.

5.2. Problem specification

The objective of the optimization process was to minimize the total weight of the roof structure while satisfying the structural performance criteria for strength and serviceability as summarized in Table 16. The section size for each member in the roof structure was a design variable (1,955 total variables). The variables were aggregated into 34 groups. The candidate section sizes within each group were chosen from the British Standards Institution [34] catalogue and possessed a consistent depth or outer diameter, depending on the type of section being considered. This was done for two reasons: (1) to ensure symmetry, continuity, and proportion for the structural elements since the roof structure was exposed, and (2) to standardize member connections to some degree to simplify the fabrication and erection process. Sample member grouping and associated section types are shown in Figure 7.

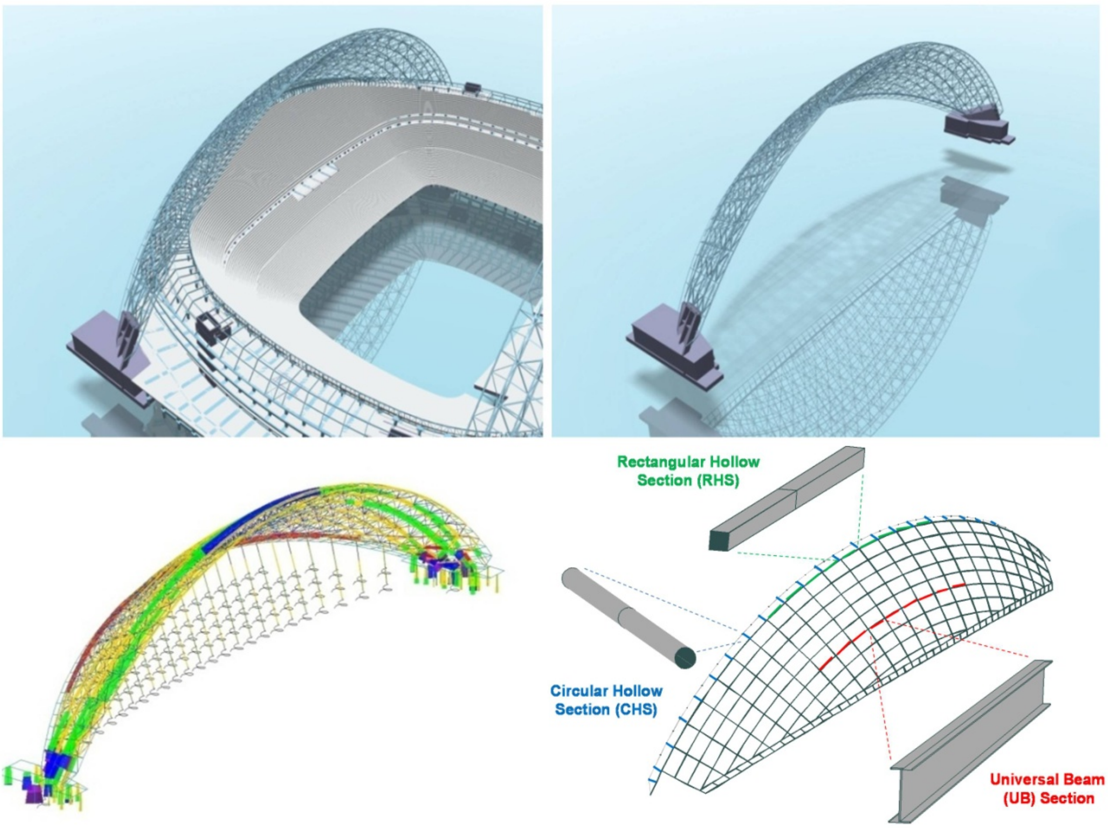

Figure 7: Building Information Model of the case study roof structure (TOP). Finite element analysis model and sample member grouping with associated section types (BOTTOM)

Objective	*Minimize* steel weight
Variables	• 1,955 member sizing variables aggregated into 34 groups • 10-30 candidate sections per variable
Constraints	• Member strength [34] • Member deflection (span / 360 for SLS load cases) • Global deflection limit (450mm at mid-span)
Design Space	≈ 1.7E2435 possible configurations of sizing variables

Table 16: Overview of case study sizing optimization problem formulation

5.3. Conventional process

The conventional member sizing optimization method of a leading design firm was an iterative process performed manually by the engineering team. First, a detailed finite element analysis model of the structure was created. It included 150 unique design loading combinations consisting of the weight of the roof as well as wind, snow, and seismic loading. The initial configuration of member sizes was determined based on the best judgment of the engineering team from their past experience with similar stadium roof structures. After completing the finite element analysis of the structure, the engineering team post-processed the structural responses to calculate a strength utilization ratio for each structural member based on the British engineering code of practice [35]. The member and global deflections of the structure were also checked against the allowable values specified in Table 16. After reviewing the performance of the structure relative to the design constraints, the engineering team then selected a new configuration of member section sizes with the goal of minimizing the weight of the structure while satisfying the design constraints. This process was repeated until the engineering team arrived at a satisfactory design configuration.

5.4. Results

The performance of the FCD method on the project was compared to the conventional design process in terms of process efficiency and solution quality. In terms of process efficiency, the FCD method took substantially longer to set up. The additional 80 man hours of set-up time was divided between developing the software 'wrapper' necessary to integrate the finite element analysis software into the ModelCenter environment (≈30 hrs) and implementing / testing the FCD optimization algorithm (≈50 hrs). These components were designed for general use such that the finite element model of the structure could be modified or replaced without modifying the underlying software or process. As a result, the same implementation of the FCD method that was used for the case study was also used for all of the numerical examples described in Section 4 without further modification. Therefore, the set-up time required to implement the FCD process in this case study would likely be significantly reduced or eliminated in subsequent applications.

Once implemented, the FCD optimization method required 340 iterations to complete, resulting in a total run time of 32 hours and 35 minutes using a Dell workstation with 2.83 GHz processor and 4 Mb of memory (Table 17). The total time required by the FCD process, including set-up, was 172 hours and 35 minutes versus 216 hours for conventional practice, or about 20% less time. In less time, the FCD method generated over eight times the number of design alternatives as the conventional practice method.

Design method	Set-up time (man hours)	Design cycle time (avg.)	Number of alternatives evaluated	Total design time
Conventional Practice	60	4 hrs	39	216hrs
FCD Method	140	5min45sec	340	172hrs35min

Table 17: Comparison of process efficiency for case study project

In terms of solution quality, the best design found by FCD method possessed a total steel weight of 2,292 metric tons while satisfying all of the design constraints. This design represents a weight reduction of 19% compared to the best design found using the conventional practice method. The weight reduction achieved equates to an estimated cost savings of approximately US $4 million for the total cost of the steelwork (US $2 million per roof structure) assuming a unit cost of structural steel of US $4,500 per metric ton. The professional engineering firm responsible for the structural design of the project chose to submit the FCD design to the contractor for tender with only a few minor modifications.

6. Summary and Conclusions

This paper presents the Fully Constrained Design (FCD) method for discrete member sizing optimization of steel truss and frame structures. FCD is different from other deterministic methods, such as optimality criteria, in that it does not require the first derivative of the objective and constraint functions with respect to the design variables. This feature improves the flexibility and robustness of the algorithm. Since the search logic for the algorithm is independent of the specific objective and constraint functions used, the FCD method can be applied to different problem formulations without modification of the algorithm to suit a particular problem formulation as required to implement the optimality criteria method. Also, there is no limitation on the continuity of the search space as there is with the deterministic methods discussed in Section 1. The FCD method can readily be applied to problems where the objective and constraint functions are discontinuous or not easily expressed in terms of the design variables.

The performance of the FCD method was compared to other optimization approaches found in the literature using three standard truss problems: a 10-bar truss, a 25-bar truss and a 200-bar truss. The results of these numerical examples indicate that the computational efficiency of the FCD method falls between that of the leading deterministic and heuristic methods. With regard to solution quality, the best designs produced by the FCD method are comparable to leading heuristic methods (<2% difference) and superior to optimality criteria (>7% difference) based on averaging the numerical example results. The benchmark studies also suggest that the scale of the problem does not have a significant impact on the efficiency of the FCD method. The numerical examples presented in this paper included only truss structures. Additional benchmarking studies using frame structures are required to generalize the claims made in this paper regarding the performance of the FCD method in comparison with other optimization methods.

We also compared the FCD method to the conventional design process used by a leading engineering firm by conducting a parallel case study: the sizing of 1,955 structural members for a steel space frame roof structure. In terms of process efficiency, the FCD implementation required an additional 80 man-hours to set up compared to the conventional (manual) process, but reduced design cycle time by approximately 40 times, once in place. The set up time required to implement the FCD process in this case study would likely be significantly reduced or eliminated in subsequent projects for the reasons discussed in Section 5.4. With regard to the quality of the final solution, the FCD solution required 19% less steel than the conventional solution, resulting in an estimated construction cost savings of US $4 million. Further case studies will be required to comment more generally on the performance and robustness of the FCD method in comparison with manual design iteration methods that are commonly used in industry.

The results we present in this paper support the claim that the FCD method has certain advantages with regard to flexibility, generality, and solution quality compared to the deterministic methods surveyed. In addition, the method is demonstrated to be scalable to structures involving greater than 100 sizing variables in a time frame that is comparable to conventional design practice. The significant savings achieved in the case study project demonstrate the potential of the FCD method to substantially improve design process efficiency and product performance.

Acknowledgements

The authors wish to thank Martin Simpson for his support of the industry case study project. In addition, we thank Dennis Shelden, and Greg Deierlein for their input to this research. The authors would also like to thank the following companies for their technical support: Arup, Oasys, and Phoenix Integration. Finally, we thank the Stanford Graduate Fellowship (SGF) program and the Center for Integrated Facility Engineering (CIFE) at Stanford University for the financial support of this research.

REFERENCES

[1] Kicinger R, Arciszewski T, Jong KD. Evolutionary computation and structural design: A survey of the state-of-the-art. Computers & Structures. 2005;83:1943-78.

[2] Nanakorn P, Meesomklin K. An adaptive penalty function in genetic algorithms for structural design optimization. Computers & Structures. 2001;79:2527-39.

[3] Rajeev S, Krishnamoorthy CS. Discrete Optimization of Structures Using Genetic Algorithms. Journal of Structural Engineering. 1992;118:1233-50.

[4] Coello C. Discrete optimization of trusses using genetic algorithms. In: Cheng J, Attia F, Crabtree D, editors. Expert Systems Applications and Artificial Intelligence JG Cheng, FG Attia and DL Crabtree: IITT International; 1994. p. 331-6.

[5] Shea K, Aish R, Gourtovaia M. Towards integrated performance-driven generative design tools. Automation in Construction. 2005;14:253-64.

[6] Arora JS. Methods for discrete variable structural optimization. In: Burns S, editor. Recent advances in optimal structural design. USA: ASCE; 2002. p. 1-40.

[7] Moses F. Optimum structural design using linear programming. J Struct Div. 1964;90:89-104.

[8] Camp C, Pezeshk S, Cao G. Optimized design of two-dimensional structures using a genetic algorithm. Structural and Multidisciplinary Optimization. 1998;124:551-9.

[9] Arora J, Huang M, Hsieh C. Methods for optimization of nonlinear problems with discrete variables: a review. Structural and Multidisciplinary Optimization. 1994;8:69-85.

[10] Svanberg K. The method of moving asymptotes-a new method for structural optimization. International Journal for Numerical Methods in Engineering. 1987;24:359-73.

[11] Lasdon L, Waren A, Jain A, Ratner M. Design and testing of a generalized reduced gradient code for nonlinear programming. ACM Transactions on Mathematical Software (TOMS). 1978;4:34-50.

[12] Barnett R. Minimum weight design of beams for deflection. Journal of the Engineering Mechanics Division. 1961:75—109.

[13] Prager W. Optimality criteria in structural design. Proceedings of the National Academy of Sciences of the United States of America. 1968;61:794-6.

[14] Venkayya V. Design of optimum structures. Computers & Structures. 1971;1:265-309.

[15] Horst R, Tuy H. Global optimization: Deterministic approaches: Springer; 2003.

[16] Sriram D, Maher ML. The representation and use of constraints in structural design. Applications of Artificial Intelligence in Engineering Problems. 1986;1:355-68.

[17] Cohn MZ, Dinovitzer AS. Application of Structural Optimization. Journal of Structural Engineering. 1994;120:617-50.

[18] Hasancebi O, Carbas S, Dogan E, Erdal F, Saka MP. Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. Computers & Structures. 2009;87:284-302.

[19] Hasancebi O, Carbas S, Dogan E, Erdal F, Saka MP. Comparison of non-deterministic search techniques in the optimum design of real size steel frames. Computers & Structures. 2010;88:1033-48.

[20] Ghasemi M, Hinton E, Wood R. Optimization of trusses using genetic algorithms for discrete and continuous variables. Engineering Computations. 1997;16:272-301.
[21] Harik G, Cantú-Paz E, Goldberg DE, Miller BL. The gambler's ruin problem, genetic algorithms, and the sizing of populations. Evolutionary Computation. 1999;7:231-53.
[22] PHX. ModelCenter 8.0. Blacksburg, VA, USA: Phoenix Integration, Inc. (http://www.phoenix-int.com/); 2008.
[23] Camp C, Pezeshk S, Cao G. Optimized design of two-dimensional structures using a genetic algorithm. Journal of Structural Engineering. 1998;124:551-9.
[24] Togan V, Daloglu AT. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. Computers & Structures. 2008;86:1204-18.
[25] Thierauf G, Cai J. Parallelization of the evolution strategy for discrete structural optimization problems. Computer-Aided Civil and Infrastructure Engineering. 1998;13:23-30.
[26] Jenkins W. A decimal-coded evolutionary algorithm for constrained optimization. Computers & Structures. 2002;80:471-80.
[27] Li L, Huang Z, Liu F. A heuristic particle swarm optimization method for truss structures with discrete variables. Computers & Structures. 2009;87:435-43.
[28] Grierson DE, Cameron GE. SODA--structural optimization design and analysis. Waterloo, Ontario, Canada: Waterloo Engineering Software; 1987.
[29] Khot NS, Berke L, Venkayya VB. Comparison of optimality criteria algorithms for minimum weight design of structures. AIAA Journal. 1979;17:182-90.
[30] Khan MR, Thornton WA, Willmert KD. An optimality criterion method for large-scale structures. AIAA Journal. 1979;17:753-61.
[31] Nanakorn P, Meesomklin K. An Adaptive Penalty Function in Genetic Algorithms for Structural Design Optimization. Computer and Structure. 2001;79:2527-39.
[32] Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. Computers & Structures. 2004;82:781-98.
[33] Erbatur F, Hasançebi O, Tütüncü I, KilIç H. Optimal design of planar and space structures with genetic algorithms. Computers & Structures. 2000;75:209-24.
[34] BS. Structural steel sections. Specification for hot-rolled sections. In: Institution BS, editor. supersedes BS 4-1:19932005.
[35] BSI. BS5950 - Structural Use of Steelwork in Building - Part 1: Code of Practice for Design - Rolled and Welded Sections. London, UK: British Standards Institution; 1999.