ABSTRACT

KRIPAKARAN, PRAKASH. COMPUTATIONAL APPROACHES FOR DECISION SUPPORT IN STRUCTURAL DESIGN AND PERFORMANCE EVALUATION. (Under the direction of Dr. Abhinav Gupta).

Structural design and performance evaluation involves complex decision-making tasks that are both time and cost intensive. Decision Support Systems (DSS) that combine engineer interaction with computer-based tools to provide a joint-cognitive decision-making environment can significantly help engineers. In this study, three different computational approaches that can become part of such a DSS are proposed. Two of these approaches use simulation-based design to assist engineers with the design of (1) bridge trusses and (2) moment-resisting steel frames, and the third is intended for internet-enabled remote control and observation of experimental facilities. A hybrid optimization approach that is composed of two algorithms - an algorithm that generates a good starting solution and a local search, is proposed for truss optimization. The performance of the approach is compared with existing optimization approaches. Then, a GA-based optimization approach is presented for the design of moment-resisting steel frames with rigid connections. The GA uses a unique crossover scheme to perform trade-off study between the number of rigid connections and the total cost. It is observed from the trade-off study that the total cost of the frame is minimum when using only a few rigid connections if they are placed at certain locations in the frame. “Modeling to Generate Alternatives - MGA” is used to generate good design alternatives that may be more suitable for design than the optimal solution. A generic computational framework for internet-enabled remote control and observation of laboratory experiments is proposed. The framework is illustrated for a shaketable experiment. The framework uses LabVIEW and web technologies to form a secure framework

that permits multiple observers to simultaneously view and download measurements from the experiment while a single user controls the shaketable.

COMPUTATIONAL APPROACHES FOR DECISION SUPPORT IN STRUCTURAL DESIGN AND PERFORMANCE EVALUATION

by

PRAKASH KRIPAKARAN

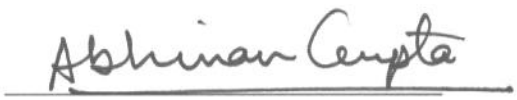
A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

CIVIL ENGINEERING

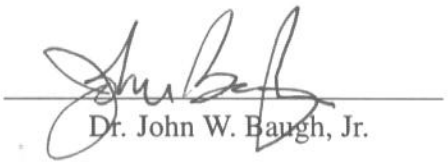
Raleigh, NC

2005

Approved By:



Dr. Abhinav Gupta
Chair of Advisory Committee



Dr. John W. Baugh, Jr.

Dr. G. Kumar Mahinthakumar

Dr. E. Downey Brill, Jr.

Biography

Prakash Kripakaran was born on June 16, 1979 in Coimbatore, India. His family moved to Madras, India in the May of 1993. He joined the undergraduate program in civil engineering at Indian Institute of Technology, Madras in August 1996 and received the Bachelors of Technology degree in May 2000. Then, he came to the U.S. for graduate study. He received the Masters degree in civil engineering from North Carolina State University in the summer of 2002. He continued his studies at North Carolina State University as a doctoral student in the department of civil engineering from Fall 2002. He has accepted a research position in the applied computing and mechanics laboratory at Ecole Polytechnique Federale de Lausanne, Switzerland. He plans to move to Switzerland in January 2006, immediately after receiving his doctorate degree.

Acknowledgments

I would like to thank my advisor, Prof. Abhinav Gupta, for giving me the opportunity to perform graduate research at North Carolina State University. His guidance and support played a significant role in the successful completion of my study. I would like to thank Prof. J. W. Baugh Jr., Prof. G. M. Kumar and Prof. E. D. Brill Jr. for serving on my graduate committee and for their valuable comments and suggestions on the research. I would like to thank Dr. Vernon Matzen with whom I interacted on several occasions during the work on remote control and observation of laboratory experiments. I would also like to thank Brian Hall for his help during the study on moment-resisting steel frames.

I would like to express my gratitude to the helpful staff of the department of civil engineering. I have made many friends during my time at NCSU. I have enjoyed their company and appreciate all their help.

Lastly, this research would not have been possible without the support and encouragement of my parents. I appreciate the sacrifices they made for me to have this education. Also, a special thanks goes out to my younger brother!

Table of Contents

List of Tables

PART II - OPTIMIZATION APPROACHES FOR TRUSS MODELS WITH REAL-WORLD COST FUNCTIONS

PART III - GENETIC ALGORITHMS FOR DESIGN OF MOMENT-RESISTING STEEL FRAMES

List of Figures

PART II - OPTIMIZATION APPROACHES FOR TRUSS MODELS WITH REAL-WORLD COST FUNCTIONS

PART III - GENETIC ALGORITHMS FOR DESIGN OF MOMENT-RESISTING STEEL FRAMES

PART IV - COMPUTATIONAL FRAMEWORK FOR REMOTELY OPERABLE LABORATORIES

Part I

INTRODUCTION

1 Introduction

Like any decision-making problem, structural engineering design poses a variety of challenges to the decision-maker. Typically, engineers face the following tasks during design - (1) Search for alternatives, (2) Evaluation of the alternatives and (3) Selection of a desired alternative. Engineers iteratively perform these time- and computation-intensive tasks to arrive at a good and feasible solution. Simulation-based approaches that combine optimization with recent advances in computing can formalize and expedite the design process [16, 12]. Researchers have made great strides in structural optimization with both traditional mathematical optimization [8] and modern heuristic optimization, e.g., using genetic algorithms [25]. In spite of all the progress, optimization approaches have made little impact in structural engineering practice because of the way in which existing optimization methodologies have been applied for decision support.

The premise of simulation-based design is that existing knowledge is sufficient to accurately represent the behavior of the structure using a mathematical model. In the absence of such knowledge, engineers resort to experimental validation for evaluating the design. The time taken for experimental validation can be significant, since large-scale laboratories that support these experiments are geographically distributed. Advances in internet and information technology have generated tremendous interest towards networking experimental research laboratories that are geographically far apart so as to make them available for engineering research and decision-making. While the internet provides the infrastructure for communication, computational approaches consistent with hardware are required for data acquisition, control, security and safety.

The proposed research focuses on developing computational approaches that can become part of structural engineering Decision Support Systems (DSS) for simulation-based design and internet-enabled experimental testing. This study can be broadly categorized into three parts:

1. Simulation-based design of bridge trusses
2. Simulation-based design of moment-resisting steel frames
3. Internet-enabled remote operation of laboratory experiments

These topics are discussed briefly in the following sections.

2 Design of Bridge Trusses

The design of a bridge truss involves finding truss topology, node locations and member sizes such that the truss can withstand the given loading for a minimal total cost. Over the past two decades, researchers have proposed numerous optimization approaches for truss design. Rajeev and Krishnamoorthy [23], Deb and Gulati [6] and Shrestha and Ghaboussi [24] have studied the use of genetic algorithms for truss design. Hansen and Vanderplaats [13] have suggested an approximation method based on Taylor series expansion of the member forces. Elperin [7] has proposed a Monte Carlo algorithm. Studies that use other optimization techniques like evolutionary strategies [21] and simulated annealing [14] also exist.

Adeli and Sarma [1] reviewed the available literature in optimization of steel frame structures and concluded that a very small number of structural steel optimization studies attempt to include any cost other than the weight of the structure. The same is also true in the case of truss optimization studies. The optimization approaches in literature are developed primarily for design problems with simple minimum-weight cost functions. In addition to the material cost that is not necessarily proportional to the weight of the truss, real-world costs also include (i) explicit costs dependent on the number of different products used in the design, (ii) fabrication cost of each joint in the truss and (iii) implicit cost related to the constraints of erecting a truss. Studies using real-world cost models are likely to encourage the use of optimization approaches in structural engineering practice. In this study, we consider a bridge truss problem with a real-world cost function.

3 Design of Moment-Resisting Steel Frames

Design of rigid steel frames that are intended to resist large lateral loads is a complex decision problem. While the use of rigid connections in the frame can considerably reduce the moments in the beams and save significant material costs, the cost of fabrication of a rigid connection is often very high due to excessive labor requirements for on-site welding. In the

current practice, all connections in a moment frame are considered rigid and designed accordingly. Depending on the relative cost of connection and the cost of material, an optimal solution that has rigid connections only at certain locations may exist. Most rigid frame optimization studies, even those attempting to model the semi-rigid behavior of connections [18, 3, 15], focus only on minimizing the total weight of the structure and avoid incorporating the fabrication cost of the connection in their optimization model. Optimization approaches that account for the cost of connections can serve as a good tool for engineers.

Lavrich [17] points out an important drawback in the current trend of using optimization to find least cost solutions. The structures resulting from optimization are often fully-stressed and have little redundancy. Engineers though are willing to compromise the cost to a certain extent if the additional benefits from a design outweigh the increase in cost. While designing moment-resisting steel frames, engineers compare different solutions on the basis of factors like the number of rigid connections, the location of the rigid connections (interior or exterior bays) and the structure's lateral load resistance capacity before choosing a solution for final design. Engineers can greatly benefit from computational approaches for *trade-off* and *what-if* analyses.

In this study, we consider a real-life steel frame formulation. The cost estimates are obtained from practicing engineers and erectors. We propose using Genetic Algorithms (GA) [9, 19] to perform trade-off study between the number of rigid connections and the total cost of the frame. The use of penalty functions [20], which can be detrimental to the performance of the GAs, is avoided using novel crossover schemes and algorithms that correct infeasible solutions. We also develop techniques that use optimization to generate design alternatives.

4 Remote Operation of Laboratory Experiments

Decision-makers opt to conduct experiments to validate their alternatives especially when mathematical models that accurately depict the structural behavior are lacking. Since large-scale experimental laboratories are cost-intensive to build and are usually situated at locations that are geographically distant for a decision-maker, significant cost as well as time may be spent during the decision-making stage. The NSF is currently sponsoring the Network for

Earthquake Engineering Simulation (NEES), which is a simulation resource that is composed of geographically distributed state-of-the-art experimental research equipment sites, to advance earthquake engineering research and education. Its objective is to make experimental laboratories with facilities like shaketables for simulating earthquakes, available for seismic performance evaluation and testing through the internet.

While internet is available as the underlying infrastructure, new IT architectures that interconnect web technologies with a framework for data acquisition and control are required. Remote monitoring of the response of a structure under testing requires sophisticated computer-controllable data-acquisition hardware. Similarly, remote control of a shaketable requires computer-controllable hardware that drives the shaketable. Another facet of an internet-enabled experiment is to provide the remote user with real-time video of the experiment. For an experiment that involves dynamic behavior, the decision-maker is interested in directly viewing the structure's motion. Shaketable experiments often involve motions in the frequency region of 5-20 Hz. Capturing this motion and communicating it in real-time requires a high frame-rate digital camera and appropriate hardware for acquiring images from the camera.

In this study, we present a generic computational framework that can aid in the creation of research collaboratories like NEES. We illustrate the fundamental concepts of the framework by implementing it for remote control and observation of a shaketable experiment. Such a framework is also applicable for making laboratory experiments available in distance-education courses at universities [5, 4, 22, 10]. The framework supports a single-controller multiple-observer concept so that many users can simultaneously observe video and download measurements in real-time as a single controller controls the test parameters.

5 Objectives and Proposed Research

5.1 Design of Bridge Trusses

The objective of this study is to develop new truss optimization methodologies that combine methodologies based on current understanding of structural behavior, and heuristic optimization approaches to solve truss problems with realistic cost functions in significantly less

computational time than existing optimization approaches. The specific tasks that are required to achieve this objective are listed below.

1. Formulate a bridge truss design problem that incorporates a real-life like cost function and a set of member products.
2. Develop new sizing optimization methodologies that include structural engineering knowledge to generate good starting solutions and to reduce the size of search space.
3. Explore strategies to explicitly handle costs on the number of product types in the truss.
4. Study the performance of the developed methodology for representative examples.
5. Compare the sizing algorithm with other recently proposed optimization techniques. This step would require modification of the cost model to that for minimum-weight design problems.
6. Implement a truss geometry optimization algorithm, which wraps around the proposed sizing approach to evaluate each geometry solution generated during the search process.
7. Evaluate the performance of the geometry optimization algorithm with respect to other geometry optimization approaches in literature.

5.2 Design of Moment-Resisting Steel Frames

The purpose of this study is to explore the use of optimization for finding the best locations of rigid connections in moment-resisting steel frames. Generating the trade-off relationship between the number of rigid connections and the total cost of the frame can provide insight into whether the total cost is minimized by minimizing the number of rigid connections in the frame. The study aims to develop new optimization-based computational approaches to perform *trade-off* and *what-if* analyses and hence, support decision-making in the design of steel frames. The following tasks are performed in this study.

1. Identify the various strength and serviceability constraints from the Manual of Steel Construction - Load and Resistance Factor Design [2].
2. Formulate the objective function using cost estimates obtained from discussion with practicing engineers and erectors.
3. Implement computer code for analyzing rigid steel frames. This task involves implementing

programs for performing finite element analyses of frames and incorporating the design checks as specified by AISC [2].

4. Evaluate the applicability of GA-based optimization approach that was developed for discrete optimization problem encountered in pipe support optimization [11]. Modify it as appropriate for application to the design of rigid steel frames.
5. Choose an appropriate real-life like rigid steel frame to study the proposed optimization approach.
6. Perform a trade-off study between the number of rigid connections in the frame and its total cost. Generate the trade-off curve for two types of support conditions in the frame - (a) all supports are fixed and (b) all supports are hinged.
7. Analyze the generated trade-off curves. In particular, observe if significant cost savings can be achieved by minimizing the number of rigid connections in the frame.
8. Generate alternative solutions with a cost comparable to that of the optimal solution using "Modeling to Generate Alternatives - MGA."
9. Comment on relative quality of alternative solutions by examining the generated designs for factors like practical feasibility, lateral load resistance and fabrication.

5.3 Remote Operation of Laboratory Experiments

The goal of this study is to develop a computational framework that can enable remote observation and control of laboratory experiments. For portability to different experiments, it is required that the software components for data acquisition and control have little dependency with the components that handle web technologies. The main tasks that are identified to complete this study are listed below.

1. Setup a proxy server that prevents direct access to the LabVIEW computers from the internet.
2. Run a webserver on the proxy server that will authenticate remote users and setup communication channels for serving data to the remote users.
3. Outline alternatives for communicating measurements and video over the internet. Examples include LabVIEW's remote front panels and direct TCP/IP implementations.

4. Create provision for remotely viewing the sensor data using web-based oscilloscope displays. Each user shall be able to individually adjust the oscilloscope settings like its scale or origin coordinates without affecting the view of other users.
5. Develop mechanisms that ensure that only a single user registered as controller is permitted to control the experiment.
6. Implement real-time video transmission such that multiple users are able to view the structural motion simultaneously.
7. Provide forms on the webserver that allows users to schedule for accessing the experiment on a future date.

6 Organization

This dissertation consists of 5 parts, some of which are primarily manuscripts prepared for submission to various peer-reviewed journals for possible future publication. Part I gives an introduction to the dissertation and provides an outline of the different research topics considered in this study. It gives the various tasks necessary to achieve the objectives of this study. It also informs the reader about the organization of the dissertation.

Part II presents a hybrid-optimization approach for truss optimization. A bridge truss formulation with real-life cost functions and member types is described. The performance of the optimization approach is compared with conventional genetic algorithms for real-life truss problems. The superiority of the optimization approach over existing approaches is illustrated for traditional minimum-weight problems.

Part III describes computational approaches for design of moment-resisting steel frames. New approaches to perform trade-off study between the number of rigid connections in the frame and its total cost are proposed. Results from the trade-off study are presented. The use of "Modeling to Generate Alternatives - MGA" to generate valuable alternatives is demonstrated.

Part IV illustrates a generic computational framework for making laboratory experiments remotely operable via the internet. The hardware and software architecture of the proposed framework are detailed. The interaction between the LabVIEW technologies and the web technologies for observation and control of the experiment are shown using schematic diagrams

and flowcharts.

Part V of the dissertation provides the summary and conclusions of this search. Recommendations for possible future research in these areas are also provided.

References

[1] H. Adeli and K. C. Sarma. 2000. *Engineering Optimization*, 32:777–802, 2000.

[2] AISC. *Manual of Steel Construction - Load Resistance Factor Design*. AISC, 3^{rd} edition, 2001.

[3] Y. A. Al-Salloum and T. H. Almusallam. Optimality and safety of rigidly- and flexibly-jointed steel frames. *Journal of Constructional Steel Research*, 35:189–215, 1995.

[4] C. L. Bohus, A. Crowl, B. Aktan, and M. H. Shor. Running control engineering experiments over the internet. In *Proceedings of the 13th IFAC World Congress*, San Francisco, CA, 1996. paper no. 4c-03.

[5] M. L. Corradini, G. Ippoliti, T. Leo, and S. Longhi. An internet based laboratory for control education. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, FL, December 2001.

[6] K. Deb and S. Gulati. Design of truss-structures for minimum weight using genetic algorithms. *Finite Elements in Analysis and Design*, 37:447–465, 2001.

[7] T. Elperin. Monte-carlo structural optimization in discrete variables with annealing algorithm. *International Journal for Numerical Methods in Engineering*, 26:815–821, 1988.

[8] O. N. Ghattas and I. E. Grossman. MINLP and MILP for discrete sizing structural optimization problems. In *In Electronic Computation: Proceedings of the Tenth Conference*, pages 197–204. ASCE, New York, NY, 1991.

[9] D. E. Goldberg. *Genetic algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing Company, 1989.

[10] A. Gupta, M. A. Gabr, and V. C. Matzen. Alternatives in the implementation of internet-enabled laboratory. In *2004 ASEE Annual Conference and Exposition*, Salt Lake City, June 2004.

[11] A. Gupta, P. Kripakaran, G. Kumar, and J. W. Baugh Jr. Genetic Algorithm-based decision support for optimizing seismic response of piping systems. *Journal of Structural Engineering*, 131(3):389–398, March 2005.

[12] R. T. Haftka and Z. Gurdal. *Elements of structural optimization*. Kluwer Academic Publishers, Boston, third edition, 1992.

[13] S. R. Hansen and G. N. Vanderplaats. Approximation method for configuration optimization of trusses. *AIAA journal*, 28(1):161–168, January 1990.

[14] O. Hasancebi and F. Erbatur. Layout optimization of trusses using simulated annealing. *Advances in Engineering Software*, 33:681–696, July-October 2002.

[15] E. S. Kameshki and M. P. Saka. Optimum design of nonlinear steel frames with semi-rigid connections using a genetic algorithm. *Computers and Structures*, 79(17):1593–1604, 2001.

[16] U. Kirsch. *Structural optimization: fundamentals and applications*. Springer-Verlag, New York, 1993.

[17] D. L. Lavrich. Basic education: A practitioner's point of view. *Structure*, pages 9–11, April 2003. A joint publication of NCSEA, CASE and SEI.

[18] E. S. B. Machaly. Optimum weight analysis of steel frames with semi-rigid connections. *Computers and Structures*, 23(4):565–574, 1986.

[19] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, third, revised and extended edition, 1996.

[20] Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*, chapter 9. Springer-Verlag, 2002.

[21] M. Papadrakakis, N. D. Lagaros, G. Thierauf, and J. Cai. Advanced solution methods in structural optimization based on evolution strategies. *Engineering Computations*, 15(1):12–34, 1998.

[22] S. E. Poindexter and B. S. Heck. Using the web in your courses: What can you do? what should you do? *IEEE Control System*, 9(1):83–92, 1999.

[23] S. Rajeev and C. S. Krishnamoorthy. Genetic algorithms-based methodologies for design optimization of trusses. *Journal of Structural Engineering*, pages 350–358, 1997.

[24] S. M. Shrestha and J. Ghaboussi. Evolution of optimum structural shapes using genetic algorithm. *Journal of Structural Engineering*, pages 1331–1338, 1998.

[25] M. S. Voss and C. M. Foley. Evolutionary algorithm for structural optimization. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO)*, 1999.

[26] S. Wirgau, A. Gupta, and V. Matzen. Internet-enabled remote observation and control of a shake table experiment. *Journal of Computing in Civil Engineering, ASCE*, 2005. In press.

Part II

NOVEL OPTIMIZATION APPROACHES FOR TRUSS MODELS WITH REAL-WORLD COST FUNCTIONS

Prakash Kripakaran, Abhinav Gupta,

and John W. Baugh, Jr.

To be submitted to the Journal of Structural Engineering, ASCE

Novel Optimization Approaches for Truss Models with Real-World Cost Functions

Prakash Kripakaran, Abhinav Gupta, and John W. Baugh Jr.

Abstract

This paper describes new optimization strategies that can be incorporated within a joint-cognitive decision-making environment for bridge truss design. Conventional truss optimization approaches, while often sophisticated and computationally intensive, have been applied to simple minimum-weight cost models. These approaches do not perform well when applied to real-world trusses, which have cost models that are complex and often involve multiple objectives. In this paper, we consider a real-world cost function consisting of costs on the weight of the truss, the number of products in the design, and the number of joints in the truss. We propose a new sizing approach that involves two algorithms applied in sequence - (1) a novel approach to generate a "good" initial solution and (2) a simple local search that attempts to generate the optimal solution by starting with the final solution from the previous algorithm. A clustering technique, which identifies members that are likely to have the same product type, is used with real-world cost functions that consider a cost on the number of products. The proposed approach gives solutions that are much lower in cost compared to those generated in a comprehensive study of the same problem using Genetic Algorithms (GA). Also, the number of evaluations needed to arrive at the optimal solution are an order of magnitude lower than that needed in GAs. Since existing optimization techniques use simpler cost functions like those of minimum-weight truss problems to illustrate their performance, the proposed approach is also applied to the same examples in order to compare its relative performance. The proposed approach is shown to generate solutions of not only better quality but also much more efficiently. To highlight the use of this sizing approach in a broader optimization framework, a simple geometry optimization algorithm that uses the sizing approach is also presented. This algorithm is also shown to provide solutions better than the existing results in literature.

1 Introduction

Structural engineers, faced with the challenge of designing reliable structures at a minimum cost, continue to seek formal techniques to assist them in obtaining cost-optimal designs. A key feature of the decision-making process is identification of alternatives for solving the given problem. Then, whether an alternative is satisfactory typically requires analysis by one or more computer models and the expertise and judgment of the engineer. Of course, such a scenario is iterative, and many such alternatives may be considered in the process. The search for both feasible and good alternatives can be tedious, and numerous optimization approaches have been developed over the years to formalize and expedite the process in structural engineering [11, 7]. Research in structural optimization has included computational complexity studies [19], traditional mathematical optimization [5], and modern heuristic optimization, e.g., using genetic algorithms [18, 6].

The computing related advances over the past decade have created a tremendous interest in heuristic optimization techniques for structural engineering design. The availability of faster computers with large memory has made it possible to employ these complex approaches that involve large-scale computing for structural optimization problems. Yet, a major drawback of most optimization approaches is that they are developed primarily for solving minimum-weight truss problems. As a result, most approaches while computation-intensive, seldom give good results when applied to truss problems with real-world cost functions. Hybrid approaches that couple structural engineering-based algorithms to find good initial solutions with traditional optimization algorithms are likely to exhibit better performance. Such hybrid optimization approaches are often employed in an alternative decision-making paradigm in which a computer-based decision support system (DSS) is used [12, 13, 10, 2].

To illustrate this, let us consider the case of truss design. The topology of the truss is often not a function of cost alone. It is also dictated by practical issues and aesthetics that may not be quantifiable. The engineer may wish to obtain optimal solutions for different topologies and then select one of these for final design. In this process, optimization can serve as a tool to evaluate the sizing and geometry for a given topology. The engineer can then compare the solutions and make additional changes before considering one for the final design. In this pa-

per, we propose computational approaches to support such a framework for truss design with conventional as well as real-world cost functions. A new method is developed for determining optimal member sizes (sizing optimization) and the location of nodes (geometry optimization) in a given topology. An attempt has been made to illustrate the efficiency and accuracy of the proposed method with respect to some of the existing and widely popular heuristic techniques. The proposed method is a hybrid approach that combines the following two key steps in sequence (i) a search that mimics professional intuition to generate a near-optimal sizing, and (ii) a non-deterministic local search that identifies the final solution. For real-world cost functions in which additional costs are incurred based on the number of different products used in the design, a "clustering" approach is implemented to identify groups of similar sized members. Examples are used for illustrating the capability of the proposed approaches to support joint-cognitive decision-making through the involvement of the designer.

2 Definition of Truss Optimization Problem

Truss design can be a complex decision problem in real world applications that involves minimization of various costs. Researchers have proposed numerous optimization approaches for truss design. Rajeev and Krishnamoorthy [15], Deb and Gulati [3] and Shrestha and Ghaboussi [17] have studied the use of genetic algorithms. Hansen and Vanderplaats [8] have suggested an approximation method based on Taylor series expansion of the member forces. Elperin [4] has proposed a Monte Carlo annealing algorithm. Other approaches like evolutionary strategies [14] and simulated annealing [9] have also been studied.

Adeli and Sarma [1] reviewed the available literature in optimization of steel frame structures and concluded that a very small number of structural steel optimization studies attempt to include any cost other than the weight of the structure. The same is true in the case of truss optimization studies. In addition to the material cost that is proportional to the weight of the truss, real-world costs also include (i) explicit costs dependent on the number of different products used in the design, (ii) fabrication cost of each joint in the truss and (iii) implicit cost related to the constraints of erecting a truss. According to Adeli and Sarma [1], studies using real-world cost models are likely to encourage the use of optimization approaches in structural

engineering practice.

In case of truss bridges, one such realistic cost function is considered in the bridge design contest [16] offered by West Point military academy. This cost function can be expressed as:

$$\text{Total cost, } T_c = C_m + C_p n_p + C_j n_j \tag{1}$$

where n_p and n_j represent the number of products and joints in the truss respectively. C_p and C_j are the cost per product and the fabrication cost per joint, respectively. C_m is the total material cost and is evaluated as

$$C_m = \sum_{i=1}^{n} c(p_i) l_i \tag{2}$$

l_i is the length of member i; n represents the number of members in the truss; and $c(p_i)$ is the cost per unit length of the product p_i used for member i. $c(p_i)$ is computed as

$$c(p) = \rho(m_p) A_p c_w(m_p, s_p) \tag{3}$$

m_p represents the material type of product p; $\rho(m_p)$ gives the material density for m_p; A_p is the cross-sectional area of product p; and $c_w(m_p, s_p)$ is the cost per unit weight of the product p. $c_w(m_p, s_p)$ of a product p depends on its material m_p and the shape of its cross-section s_p. The products p_i are chosen from a discrete set S of products such that $p_i = S(j)$ and $S(j)$ refers to the j^{th} product in set S. The products in set S are sorted in increasing order of cost per unit length such that, $c(S(i)) \leq c(S(j)) \implies i \leq j$.

Equation 1 is a fairly generic representation of the costs in real-world bridge trusses. It reduces to the minimum-weight problem, a commonly used cost function in structural optimization studies, when $C_p = C_j = 0$ and $c(p_i) = \rho(m_{pi}) A_i$. Consequently, the approaches presented in this paper are intended to work well not only for real-world cost functions but also for minimum-weight truss problems.

In this paper, we propose a new truss optimization formulation that uses the cost function given in Equation 1. Like most other existing formulations, the constraints for a truss design problem in the proposed formulation prescribe upper limits on member stresses and structural displacements. To verify the stress constraints, the maximum tensile and compressive loads for each of the members are computed and compared against the tensile and compressive strengths of the corresponding members. The displacement constraints are given as limits on the horizontal and vertical displacements at the different nodes of the truss. The stress violation, P_s

and the displacement violation, P_d, are calculated using the following equations.

$$P_s = \sum_{i=1}^{i=n} P_{si}, \text{ and } P_{si} = \{\max(F_t(i) - S_t(p_i), 0) + \max(F_c(i) - S_c(p_i), 0)\} \tag{4}$$

$$P_d = \max \left\{ \begin{array}{l} (d_j - s_j), \ \ \forall \ \ j = 0..m \\ 0 \end{array} \right\} \tag{5}$$

$S_c(p_i)$ and $S_t(p_i)$ are the compressive and tensile force capacity of product p_i; $F_c(i)$ and $F_t(i)$ are the maximum compressive and tensile forces in member i, respectively; d_j and s_j are the maximum evaluated displacement and the specified maximum limit at degree of freedom j; and m is the total number of degrees of freedom in the truss. $S_c(p_i)$ and $S_t(p_i)$ are evaluated as,

$$S_c(p_i) = \left\{ \begin{array}{l} 0.9(0.66^{\lambda}\sigma_{yi}A_i), \ \ \lambda \leq 2.25 \\ 0.9 \left\{ \dfrac{0.88\sigma_{yi}A}{\lambda} \right\}, \ \ \lambda > 2.25 \end{array} \right. \tag{6}$$

$$\lambda_i = \frac{\sigma_{yi}A_iL_i^2}{\pi^2 E_i I_i} \tag{7}$$

$$S_t(p_i) = 0.95(\sigma_{yi}A_i) \tag{8}$$

p_i refers to the product assigned to the i^{th} bar; E_i and σ_{yi} are the Young's modulus and yield stress of the bar material; and A_i and I_i are the area and moment of inertia of product p_i. L_i is the length of member i. In Equations 6 and 8, 0.90 and 0.95 correspond to the resistance factors used for compression and tension, respectively. The stress and displacement constraints are enforced during the optimization process by using penalty costs with P_s and P_d. The constraint-free objective function for this problem is given as

$$\text{Minimize } \{Z = aP_s + bP_d + T_c\} \tag{9}$$

a and b are constants that serve as weights to the stress and displacement violations, respectively, and can vary depending on the problem. The specific values of a and b used for various examples are presented later along with the discussion on a particular example.

3 Sizing Algorithm

The proposed hybrid sizing strategy involves two algorithms - (i) a novel approach to generate a good initial solution, which is used as a starting point for (ii) a non-deterministic local

search. The first algorithm uses a special "clustering" technique to intelligently determine a good initial set of clusters. This technique is described next.

3.1 Clustering

Clustering is proposed to explicitly handle the problem of minimizing the number of products n_p in the truss. The idea behind this technique is that members of the truss that exhibit similar behavior will most likely have the same product. Based on the forces that members may experience, one can arrive at the clusters of members that are likely to have the same products. Before this technique is formally presented as part of the algorithm to generate the starting solution, we present a simple graphical illustration to explain its basic premise.

Draw a graph in which the x and y axes represent the compressive and tensile forces, respectively. For the given truss topology, product types for each of its members can be selected arbitrarily based on professional experience as an initial solution. Evaluate the maximum tensile forces $F_t(i)$ and compressive forces $F_c(i)$ in different members by analyzing the truss for various load cases. Using the evaluated forces, plot the point corresponding to each member on this graph. Then, consider a rectangular area that encloses all the plotted points such that each side of the rectangle is parallel to one of the two axes. Divide this rectangular area into a grid using equally spaced vertical and horizontal lines. The members of the truss that are present in the same unit of the grid are considered as belonging to the same cluster. As an example, Figure 1 gives the clusters in the optimal solution for the 21-bar truss shown in Figure 2. As is evident from the figure, the grid size determines the number of clusters that will be generated. The finer the grid, the larger is the number of clusters, and vice versa. The formal algorithm for this clustering technique in the proposed hybrid approach to sizing is outlined next.

3.2 Algorithm 1 - Clustering to Generate Starting Solution

The formal representation for a truss sizing solution is presented to facilitate the development and explanation of the proposed algorithm. For a given topology, a sequence of cluster and product pairs - (c_i, p_i) can be used to represent a sizing solution. Here, c_i is given by,

$$c_i = \{e \mid 1 \leq e \leq n \ \& \ e \notin c_j \ \forall \ j \neq i\} \tag{10}$$

All members in cluster c_i use the same product p_i. The algorithm is outlined in the following steps.

Step 1 : Assign the product having the least cost to all the members as follows:

$$p_i = S(0) \ \forall \quad i = 1 \ldots n \tag{11}$$

n is the number of members in the truss.

Step 2 : Analyze the truss to obtain the maximum tensile force $F_t(i)$ and compressive force $F_c(i)$ experienced by each member i.

Step 3 : For each member i, determine the least cost product (q_i) such that, $S_t(i) > F_t(i)$ and $S_c(i) > F_c(i)$ if q_i is assigned to the i^{th} member.

Step 4 : Find the minimum and maximum of both compressive and tensile forces over all the members. Let us call them $F_{c\ min}$ and $F_{c\ max}$, and $F_{t\ min}$ and $F_{t\ max}$, respectively.

Step 5 : In a graph in which the x and y axes represent the $F_c(i)$ and $F_t(i)$ respectively, consider the rectangular area bounded by $x = F_{c\ min}$, $x = F_{c\ max}$, $y = F_{t\ min}$ and $y = F_{t\ max}$.

Step 6 : Divide the rectangular area into a $r_1 \times r_2$ grid using vertical and horizontal lines that are equally-spaced. Number the grids as $1, 2, \ldots, r_1 r_2$ sequentially starting from the left-bottom corner and proceeding row-wise.

Step 7 : Identify the grid number g_i for each member i based on its $F_c(i)$ and $F_t(i)$.

Step 8 : Form clusters c_j by grouping all members that have the same grid number g_i.

Step 9 : Assign the same product r_j to all members of each cluster c_j, where r_j is the median of the products q_i found in step 3 for the members in this cluster.

Step 10 : Repeat steps $2 - 9$ until the clusters obtained over two consecutive iterations are the same.

Step 11 : Enforce the stress constraints by repetitively trying products of next higher cost for clusters that have members with force violation. This process is illustrated using a flowchart in Figure 3.

Step 12 : Enforce the displacement constraints using the iterative procedure given as a flowchart in Figure 4. For a solution that currently violates displacement constraints, its cost either remains the same or increases when it is made to satisfy certain specified displacement constraints. This algorithm attempts to find a solution in the neighborhood of the current solution such that it satisfies displacement constraints with minimal increase in cost. The algorithm

given in Figure 4 generates the final solution by performing the following steps iteratively - (i) For each cluster with displacement violation ($P_d > 0$), find the least cost product that reduces the displacement violation and evaluate the ratio (θ) of the amount by which the displacement violation decreases (ΔP_d) to the amount by which the cost increases (ΔT_c) and (ii) Accept the change in product for only that cluster which has the highest value for the ratio (θ), i.e., for cluster corresponding to θ_{max}.

Step 13 : Return the solution. This solution serves as a very good initial solution in a broader search procedure.

When attempting to solve a simple minimum-weight problem that does not have costs on the number of products, skip steps $4 - 8$ and assume that each member belongs to a cluster of its own. In other words, $c_i = \{i\}$ and $\{(c_0, q_0) \ldots (c_n, q_n)\}$ forms the solution to the problem.

3.3 Algorithm 2 - Local Search

The solution generated from the previous algorithm is used as a starting point in this local search that finds the optimal solution. The algorithm is given using a flowchart in Figure 5. The key components of the algorithm are described below:

- *Transition*: A transition is any one of the following two alternatives -
  1. Change the product $p = S(j)$ of a randomly chosen cluster c to a new product q such that $\delta = |c(p) - c(q)|$ is small. Since S is already sorted in the order of increasing cost, randomly choose a product q near the current product p as follows,

$$q = \text{rand}(S_c) \ , S_c = \{S(j - R), S(j - R + 1) \ldots S(j + R)\} \tag{12}$$

     rand() is a function that randomly chooses any element from S_c with equal probability. R is a positive integer value that prescribes the maximum allowed change.
  2. Combine any two or more randomly chosen clusters into one cluster. This is an explicit attempt to reduce the number of products used in the design. Suppose that clusters - $d_1 \ldots d_m$, are randomly chosen for combination. Let $q_1 \ldots q_m$ represent their corresponding products. Create a new cluster-product pair (d, q) such that $d = \{d_1 \cup d_2 \cdots \cup d_m\}$ and $q = \max(q_1 \ldots q_m)$.

  Selection : If $A1$ and $A2$ denote the two alternatives described above, then during any

transition T, only one of $A1$ and $A2$ is chosen as follows,

$$T = \begin{cases} A1, & X \geq P \\ A2, & X < P \end{cases} \tag{13}$$

X is a random variable uniformly distributed in the interval $[0, 1]$. Parameter P defines the probability with which one of the alternatives is chosen over the other. It is set such that $A1$ is chosen with a higher probability than $A2$. We observed that a value of P between 0.70 and 0.90 gave consistent and good results in the various examples considered.

- *Memory*: The algorithm uses a memory that keeps track of the solutions already visited. Specifically, in the event of combining two clusters, it is possible that the new solution generated has already been explored. This memory can only hold a finite number of solutions. Therefore, it is possible that the same solution is examined multiple times in the search. This inefficiency of the proposed algorithm may be avoided by using a larger memory.

The performance of this sizing algorithm is analyzed in the following section.

4 Sizing Algorithm Validation

The performance of the algorithm is evaluated using two examples. Its performance is first discussed for a truss problem with real-world cost function. Next, its performance is studied for a minimum-weight truss problem that is commonly cited in literature to compare its performance with respect to existing algorithms.

4.1 Real-World Truss Problem: 21-Bar Truss Example

The problem details such as the available products, the load cases, and the cost model are borrowed from a problem offered as a bridge-truss design contest by Westpoint Military Academy [16]. The truss configuration is shown in Figure 2. The length of the various members of the truss are also indicated in the figure. The cost function for this problem is same as given in Equation 1 with $C_p = \$600.00$ and $C_j = \$150.00$. The material cost C_m is calculated using Equation 2. For this problem, the product set S consists of 210 elements. S has an equal number of solid bar and hollow tube cross-sections. The cross-section dimensions of

the products are listed in Table 1. A cross-section of particular shape (hollow tube or bar) and size is available in three different materials - carbon steel, high tensile steel and tempered steel. The properties of all three materials are listed in Table 2. The cross-section data in Table 1 and the material data in Table 2 are used in Equation 3 to find the cost per unit length $c(p)$ of any product p. It is observed that the cost per unit length $c()$ for the products do not monotonically increase with increasing area. Instead, the values of $c()$ are such that hollow tubes are usually cost-effective for compression members compared to solid bars of same area whereas the solid bars are cost-effective for tension members, as is the case in conventional structural engineering practice.

The bridge is subject to six different load cases. All the loads in each of the load cases act at the nodes of the bottom chord. Table 3 gives the dead load of the deck acting on the truss. In addition to the deck load given in the table, the dead load corresponding to the weight of the truss members is included in each load case by assigning half the weight of a member to each of its nodes. Table 4 gives the six different cases of live loads acting on the truss. The truss is analyzed for various load combinations such that each combination involves all of the dead load and a single live load case.

The proposed algorithm is used to obtain the optimal solution to the 21-bar truss. In Equation 9, the penalty cost for stress violation a is set as 200000 and the penalty cost for displacement violation b is set to zero. The latter is set to zero because there are no constraints on displacements for this problem. The best solution generated over a sequence of 50 runs using different random seeds is found to have a cost of $\$\ 41,538$. The clusters in the optimal solution are $c_1 = \{1, 2, 5, 6, 14, 19\}, c_2 = \{3, 4\}, c_3 = \{7, 8, 9, 10, 11\}, c_4 = \{12, 21\}$ and $c_5 = \{13, 15, 16, 17, 18, 20\}$. The products that are assigned to the various clusters are listed in Table 5. The average cost of the optimal solution and the average number of evaluations over the 50 different runs are $\$\ 41,600$ and 6905 respectively.

We can make some important observations from the generated optimal solutions that give engineering logic to the generated design. The member composition of the different clusters in the optimal solution reflect engineering intuition. The solutions have clusters that directly correspond to the top chord, the bottom chord and the two sets of diagonal members of the truss. In conventional practice, engineers often arrive at designs that have this property. From

a structural engineer's perspective, one can also visualize that the top chord of the truss is in compression and the bottom chord is in tension. Correspondingly, the optimal solution has tubular sections for the top chord and bar cross-sections for the bottom chord. Although there is symmetry in geometry and loading, symmetry in member sizing is not enforced explicitly. But the final design given by the sizing algorithm does show symmetry. All these properties show that solutions generated using this optimization approach will prove useful towards decision-making by a truss designer.

While the above discussion provides some engineering logic to the generated optimal solution, the overall performance of the approach can be shown only by comparing the generated solution with the global optimal solution or by comparing the approach to other commonly-used optimization techniques. In this case, a complete enumeration of all possibilities to find the global optimal solution is not feasible, since the problem space (210^{21}) is too large. Therefore, we compare the performance of the proposed approach with Genetic Algorithms (GA), a widely-used optimization technique in the recent past. Symmetry in sizing is explicitly imposed in the GA formulation. Consequently, there are 11 decision variables in the formulation. Each decision variable is modeled using 8-bits in the GA. GAs with large population sizes are run for several generations to find the best results one can obtain with a GA. Two different crossover schemes - one-point crossover and uniform crossover, are separately used in the GA. The GAs are assumed to have converged if the optimal solution does not change over 100 generations. The results are tabulated in Table 6. The table lists results for four different cases characterized by differing population sizes and crossover schemes. For each case, the GAs are run 20 times with each run using a different random number seed. The table lists the cost of the best solution, the average cost of the solutions and the average number of evaluations over the 20 different runs for each case. From the table, it is seen that the best solution obtained using GA has a cost of $\$\ 65,482$, which is $\$\ 23,944$ (about 60%) more than the cost of the best solution produced by the proposed approach. The GAs not only take a much higher number of evaluations, the cost of the final solution is also significantly higher.

5 Minimum-Weight Truss Problem: 10-Bar Truss Example

In this section, the performance of the proposed sizing technique is compared with that of a genetic algorithm-based optimization approach. In particular, we consider the GA-based approach proposed by Rajeev and Krishnamoorthy [15]. They have illustrated the approach for the ten-bar truss shown in Figure 6. As this is a minimum-weight truss problem, $C_p = C_j = 0$ in Equation 1. The product set S consists of 79 different products. All the products are aluminium bars having cross-section areas from $[0.6452$ cm$^2; 3.226K$ cm^2 $(K = 1, 2 \ldots 76); 256.854$ cm$^2; 258.08$ cm$^2]$. It is assumed that the bars do not buckle and the compressive strength of the bar is same as its tensile strength. Instead of Equations 6-8, the following equation is used to determine $S_c(p_i)$ and $S_t(p_i)$.

$$S_c(p_i) = S_t(p_i) = \sigma_{yi} A_i \tag{14}$$

Displacement constraints are specified using upper limits on vertical displacements at the nodes in the bottom chord. The displacement constraints and material properties are summarized in Table 7. Interested readers are referred to Rajeev and Krishnamoorthy [15] and Elperin [4] for comprehensive details on this problem.

The values $a = b = 200000$ are used for the parameters in Equation 9. The best solutions obtained for this truss and the number of solutions evaluated in a sequence of 10 runs with different random seeds in the proposed approach are given in Table 8. While the best solutions given by Rajeev and Krishnamoorthy [15] and Elperin [4] have a weight of 22653.55 N and 25829.73 respectively, the proposed approach generates a solution with a weight of 22573.68 N. Also, while $6,050$ and $80,000$ solutions are evaluated in the GA-based approach and Monte-Carlo method, respectively, the proposed approach requires evaluation of only 826 solutions on an average. All three solutions are given in Table 9.

The 10-bar truss problem is much simpler than the previously-discussed 21-bar truss example. The features that contribute to a significant reduction in complexity of the 10-bar truss problem are summarized below,

1. The cost model is only made up of the weight. Costs on the number of products if present would have resulted in a more complex objective function.
2. Members are assumed to exhibit the same behavior in both tension and compression which

is contrary to the bridge truss formulation, in which buckling is considered. Due to buckling, the compressive strength of a member can be significantly lower than its tensile strength depending on the shape of its cross-section and the member length. As a result, it is important to handle member design for compression and tension differently.

3. All products available for the design are made of the same material - aluminium. For the real-world cost function, products of three different materials that have different allowable strengths are available.
4. Cost (or weight) increases monotonically with increase in area of the member. For the real-world cost function, increase in member area does not correspond to increase in cost. The cost also depends on the material type and the basic shape of the product.

Most of the sizing optimization approaches in the literature are designed for academic problems that suffer from one or more of the above described simplifications. As illustrated earlier with a GA-based approach for the 21-bar truss, existing approaches do not give good results when real-world cost models are considered because such models produce a more complex decision space. The proposed approach is designed with heuristics to handle such real-world cost models. In the following section, we illustrate a relatively simple way of incorporating the proposed sizing optimization approach in a geometry optimization algorithm.

6 Geometry Optimization Algorithm

The goal of geometry optimization is to find the optimal shape of the truss for a given topology. Therefore, decision variables of the problem include the coordinates of certain nodes of the truss in addition to the sizing variables for its different members. We consider a discrete design space for nodal coordinates using a rectangular grid such that any node in the truss can only be present at one of the grid-points. We propose a layered-optimization framework in which the geometry optimization algorithm evaluates the quality of a geometry solution by using the sizing approach to find the least cost solution for that particular geometry.

The proposed geometry optimization algorithm, which is simply a local search, is given using a flowchart in Figure 7. The key elements of the search procedure are now described in detail.

- *Transition* : In each iteration, the algorithm attempts a certain specified number of transitions (n_t). The coordinates of the truss nodes undergo small random changes in each transition. In any transition, the coordinates of one or multiple nodes may be shifted. Suppose that the design space for the truss is discretized in into a grid by equally-spaced lines in x and y-directions. Let the spacing between the lines in both the directions be equal to d. Then, the random shifts in x and y-direction for a particular node can be given by $\Delta x = Xd$ and $\Delta y = Yd$, where X and Y are random variables that return a integer value in the interval [-R,+R] and R is an integer value that determines the maximum allowed change. By starting the search procedure with trial solutions that have nodes only at the grid-points, it is ensured that all solutions generated by the transitions also have nodes only at the grid-points. The best solution generated in all attempted transitions is taken as the initial solution for the next iteration.
- *Solution evaluation* : The objective value of any geometry solution is defined as the cost of the least-cost solution for the truss with that particular geometry. Thus, an evaluation of a solution in the geometry optimization procedure actually involves using the previously described sizing approach to find the least cost solution for the particular geometry.
- *Convergence criteria* : The algorithm keeps track of the best solution, O, generated during the entire course of the search procedure. The algorithm converges when the best solution, O, has not changed over a specified number of iterations (n_f).

In the following section, we show results that illustrate that this simple algorithm performs better than other common optimization approaches proposed for shape optimization.

7 Performance of Geometry Optimization Algorithm

The performance of the geometry optimization is studied on an 18-bar cantilever truss used by Rajeev and Krishnamoorthy [15] and Hansen and Vanderplaats [8]. The truss topology is shown in Figure 8. There is only one load case and it consists of five loads of value 88.97 kN acting at nodes 1, 2, 4, 6 and 8 respectively in the truss. The nodes on the top chord are fixed. Only the coordinates of the four nodes of the bottom chord - 3, 4, 7 and 9, are allowed to move. The available products have areas between 1250 and 12500 mm^2 (2 and 20 in^2) in steps

of 156.25 mm^2 (0.25 in^2). The allowable compressive and tensile forces for a member i using product p_i are calculated using,

$$S_c(p_i) = \frac{-K_i E_i A_i^2}{L_i^2}, \text{ and } S_t(p_i) = \sigma_{yi} A_i \tag{15}$$

K_i is a constant determined from the cross-sectional geometry of p_i. $K_i = 4$ for this problem. E_i is the Young's modulus of the material of product p_i. A_i is the cross-section area of product p_i. L_i is the length of the member i. There are no displacement constraints specified for this problem. As in the case of 10-bar truss, the cost is once again defined as the weight of the truss. For complete details on the problem, the readers are referred to Hansen and Vanderplaats [8].

Rajeev and Krishnamoorthy [15] have used a GA-based approach, which is an extension of the approach mentioned in the sizing discussion, for shape optimization of this truss. In their approach, the following assumptions were made on the member areas: $A_1 = A_2 = A_3 = A_4 = A_5; A_6 = A_7 = A_8 = A_9 = A_{10}; A_{11} = A_{12} = A_{13} = A_{14}; A_{15} = A_{16} = A_{17} = A_{18}$. The optimal solution generated by the GA-based approach has a weight of $20,537$ N ($4,616.82$ lb). This solution is shown in Figure 9. To use our approach, we discretize the design space into a rectangular grid using vertical and horizontal lines that are 0.254 m (10 in) apart. The search parameters are defined as follows: $d = 0.254$ m $(10$ in$)$ and $R = 3$. $d = 10$ ensures that the new node locations found during the search process lie only on the grid-points of the discretized space. We perform two different case studies for this truss. In case A, the assumption made on the member areas by Rajeev and Krishnamoorthy [15] is enforced during the search. Consequently, there are only four area variables in the search for case A whereas in case B, each member is permitted to have a different member area leading to 18 area variables. Our approach generates optimal solutions with weights of 20,468 N and 16,156 N for cases A and B, respectively. These solutions are shown in Figures 10 and 11. The node locations and member areas of the optimal solutions generated by Rajeev and Krishnamoorthy [15] and our approach are given in Tables 10 and 11 respectively. The solution generated in case A is very similar to the solution obtained by Rajeev and Krishnamoorthy. On the other hand, the solution obtained in case B is much different from the other two in both appearance and member areas. The solutions for both cases have weights that are lower than the solutions evaluated by Rajeev and Krishnamoorthy.

To illustrate the performance of the proposed approach for real-world problems, we seek the geometry optimal solution for the 21-bar truss shown in Figure 2. Only nodes 8 and 9 are permitted to move. Node locations of 11 and 10 are constrained using symmetry relationship with 8 and 9 respectively. The node locations and the member details of the optimal solution generated using the proposed approach are listed in Tables 12 and 13. The cost of the optimal solution is $\$\ 37,296$.

8 Summary and Conclusions

Existing structural optimization studies have focussed their attention primarily on optimizing truss problems with weight as the cost. Real-world applications involve numerous other costs like those dependent on the number of joints and the number of products used in the design, in addition to the weight of the truss. This paper considers a truss problem with a real-world cost function. Specifically, the cost function includes costs on the number of products used in the truss. A clustering technique is proposed to identify members that are likely to have the same product type. A novel hybrid sizing approach that uses the following two algorithms is presented. The first algorithm in the hybrid approach uses the clustering technique to arrive at a near-optimal initial solution. The second algorithm begins with the solution generated by the previous algorithm and performs a local search to find the optimal solution. This approach is shown to generate good results when applied to real-world truss problems. The performance of the sizing approach is also compared with the results obtained from some recently developed optimization schemes. The performance comparison is attempted in the context of minimum-weight truss problems as results for the other approaches are available only for minimum-weight trusses. It is observed that the proposed approach outperforms the other approaches with respect to computational efficiency as well as quality of optimal solution. Finally, a simple way of integrating the proposed sizing approach into a geometry optimization technique is demonstrated. A layered-optimization framework, in which each geometry obtained in the process of geometry optimization is evaluated by finding the least-cost solution with the sizing approach, is proposed. This algorithm is also compared to a GA-based geometry optimization algorithm. The solutions generated by the proposed geometry optimization approach are better

than the solutions generated by the GA-based approach. The approach presented in this paper is suitable for a Decision Support System (DSS) for truss design. The designer can evaluate various topologies by using the proposed approaches. Often, designers prefer to keep the discretion of deciding a truss topology, based on their experience, judgment and aesthetics. The proposed approach provides an ideal environment for joint-cognitive decision-making that involves the designer for generating topologies and utilizes the computational approaches to generate the best sizing and geometry solution for each topology.

The main conclusions from this study are listed below:

- The clustering technique mimics the professional structural engineering knowledge to handle the cost associated with the number of products. The number of clusters generated using this technique is governed by the size of the grids in the rectangular area bounded by the maximum forces.
- The first algorithm in the hybrid approach generates a near-optimal solution. This algorithm contributes significantly to the overall computational efficiency of the proposed approach.
- The proposed hybrid sizing approach generates consistently good results for real-world cost functions.
- The sizing algorithm performs better than existing optimization techniques with respect to minimum - weight truss problems.
- The power of the sizing approach is also demonstrated in the geometry optimization algorithm. While the algorithm is an extension of simple local search, it generates much better results than existing optimization techniques.

References

[1] H. Adeli and K. C. Sarma. 2000. *Engineering Optimization*, 32:777–802, 2000.

[2] J. W. Baugh, Jr., S. C. Caldwell, and E. D. Brill, Jr. A mathematical programming approach to generate alternatives in discrete structural optimization. *Engineering Optimization*, 28:1–31, 1997.

[3] K. Deb and S. Gulati. Design of truss-structures for minimum weight using genetic

algorithms. *Finite Elements in Analysis and Design*, 37:447–465, 2001.

[4] T. Elperin. Monte-carlo structural optimization in discrete variables with annealing algorithm. *International Journal for Numerical Methods in Engineering*, 26:815–821, 1988.

[5] O. N. Ghattas and I. E. Grossman. MINLP and MILP for discrete sizing structural optimization problems. In *In Electronic Computation: Proceedings of the Tenth Conference*, pages 197–204. ASCE, New York, NY, 1991.

[6] A. Gupta, P. Kripakaran, G. Kumar, and J. W. Baugh Jr. Genetic Algorithm-based decision support for optimizing seismic response of piping systems. *Journal of Structural Engineering*, 131(3):389–398, March 2005.

[7] R. T. Haftka and Z. Gurdal. *Elements of structural optimization*. Kluwer Academic Publishers, Boston, third edition, 1992.

[8] S. R. Hansen and G. N. Vanderplaats. Approximation method for configuration optimization of trusses. *AIAA journal*, 28(1):161–168, January 1990.

[9] O. Hasancebi and F. Erbatur. Layout optimization of trusses using simulated annealing. *Advances in Engineering Software*, 33:681–696, July-October 2002.

[10] H. M. Karandikar and F. Mistree. The integration of information from design and manufacture through decision support problems. *Applied Mechanics Reviews*, 44(10), 1991.

[11] U. Kirsch. *Structural optimization: fundamentals and applications*. Springer-Verlag, New York, 1993.

[12] D. H. Loughlin, J. K. Neal, S. Ranjithan, E. D. Brill, Jr., and J. W. Baugh, Jr. Decision support system for air quality management. In *Computing in Civil Engineering: Proceedings of the 2nd Congress, ASCE*, pages 1367–1374, 1995.

[13] F. Mistree, W. F. Smith, B. Bras, J. K. Allen, and D. Muster. Decision-based design: A contemporary paradigm for ship design. In *Transactions, Society of Naval Architects and Marine Engineers*, volume 98, pages 565–587, Jersey City, NJ, 1990.

[14] M. Papadrakakis, N. D. Lagaros, G. Thierauf, and J. Cai. Advanced solution methods in structural optimization based on evolution strategies. *Engineering Computations*, 15(1):12–34, 1998.

[15] S. Rajeev and C. S. Krishnamoorthy. Genetic algorithms-based methodologies for design optimization of trusses. *Journal of Structural Engineering*, pages 350–358, 1997.

[16] S. Ressler. West point bridge design contest, united states military academy, 2004. http://bridgecontest.usma.edu/.

[17] S. M. Shrestha and J. Ghaboussi. Evolution of optimum structural shapes using genetic algorithm. *Journal of Structural Engineering*, pages 1331–1338, 1998.

[18] M. S. Voss and C. M. Foley. Evolutionary algorithm for structural optimization. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO)*, 1999.

[19] D. F. Yates, A. B. Templeman, and T. B. Boffey. The complexity of procedures for determining minimum weight trusses with discrete member sizes. *International Journal of Solids and Structures*, 18(6):487–495, 1982.

Table 1: Product details: 21-bar truss

Product dimensions	
Outer dimensions (Bar/Tube) (mm×mm)	Wall thickness (Tube) (mm)
10×10	2
15×15	2
.	”
55×55	2
60×60	3
.	”
75×75	3
80×80	4
90×90	4
100×100	5
110×110	5
120×120	6
130×130	6
140×140	7
150×150	7
160×160	8
170×170	8
180×180	9
190×190	9
200×200	10
210×210	10
220×220	11
230×230	11
240×240	12
250×250	12
300×300	15
400×400	20
500×500	25

Table 2: Material properties - 21-bar truss

Property	Carbon Steel (CS)	High Strength Steel (HS)	Quenched /tempered Steel (QS)
Elastic modulus	2×10^8kN/m^2		
Density (ρ)	7850 kg/m^3		
Yield stress $(\times 10^4$kN/m$^2)$	25	34.5	48.5
c_w (\$/kg) Bar shapes	3150	4200	5700
c_w (\$/kg) Tube shapes	5040	6250	7950

Table 3: Dead load details - 21-bar truss

Load Node	Load value (kN)
2	189.34
3	189.34
4	189.34
5	189.34
6	189.34

Table 4: Live load details - 21-bar truss

Load case	Node	Load value (kN)
1	2	418.95
2	2	418.95
	3	418.95
3	3	418.95
	4	418.95
4	4	418.95
	5	418.95
5	5	418.95
	6	418.95
6	6	418.95

Table 5: Optimal solution for the 21-bar truss shown in Figure 2

Cluster No.	Product details		Member numbers
	Cross-section	Material	
1	Bar 55×55	HS	1,2,5,6,14,19
2	Bar 55×55	QS	3,4
3	Tube $220 \times 220 \times 11$	CS	7,8,9,10,11
4	Bar 30×30	QS	12,21
5	Tube $110 \times 110 \times 5$	CS	13,15,16,17,18,20

Table 6: Details of the GA runs for the 21-bar truss shown in Figure 2

Case No.	Population size	Crossover	Average number of evaluations	Optimal solution ($)	Average cost ($)
1	1000	one-point	176600	70,991	86,670
2	5000	one-point	943500	65,482	78,889
3	1000	uniform	181600	73,408	93,218
4	5000	uniform	856250	66,773	84,349

Table 7: Problem details for 10-bar truss in Figure 6

Material	Aluminium
Young's modulus	68.95×10^6 kN/m^2
Specific weight	2768 Kg/m^3
Allowable stress	0.172375×10^6 kN/m^2
Displacement limits (nodes 1-4)	$\delta = 5.08$ cm
Length	$l = 9.144$ m

Table 8: Minimum weight values for the 10-bar truss shown in Figure 6 over 10 different runs

Minimum weight (N)	Number of evaluations
22716.57	836
22655.97	886
22653.70	836
22655.97	786
22716.57	786
23147.73	786
22573.68	736
22622.82	886
22722.26	886
22722.26	836

Table 9: Sizing results for the 10-bar truss shown in Figure 6

Member. No.	Our approach (cm^2)	Rajeev and Krishnamoorthy (cm^2)	Elperin (cm^2)
1	200.01	206.46	200.01
2	0.65	0.65	0.65
3	154.85	151.62	129.04
4	93.55	103.23	90.33
5	0.65	0.65	0.65
6	3.23	0.65	0.65
7	48.39	54.84	51.62
8	138.72	129.04	145.17
9	135.49	132.27	145.17
10	0.65	0.65	96.78

Table 10: Node locations of the geometry solutions for the 18-bar cantilever truss given in Figure 8

Node No.	Our approach Case A		Rajeev and Krishnamoorthy		Our approach Case B	
	x (m)	y (m)	x (m)	y (m)	x (m)	y (m)
3	23.5	5.0	22.3	4.1	24.1	2.9
5	16.8	3.8	15.5	3.0	17.7	2.3
7	10.9	2.5	9.8	1.8	11.4	1.8
9	5.3	0.5	4.7	0.6	5.5	0.5

Table 11: Member areas of the geometry solutions for the 18-bar cantilever truss given in Figure 8

Member No.	Our approach Case A (mm^2)	Rajeev and Krishnamoorthy (mm^2)	Our approach Case B (mm^2)
1	8064.50	7741.92	1451.61
2	8064.50	7741.92	1774.19
3	8064.50	7741.92	3709.67
4	8064.50	7741.92	6129.02
5	8064.50	7741.92	7580.63
6	10483.85	11935.46	7419.34
7	10483.85	11935.46	8225.79
8	10483.85	11935.46	10483.85
9	10483.85	11935.46	11774.17
10	10483.85	11935.46	12096.75
11	5161.28	3387.09	2258.06
12	5161.28	3387.09	4032.25
13	5161.28	3387.09	5000.00
14	5161.28	3387.09	5483.86
15	2580.64	2903.22	1774.19
16	2580.64	2903.22	2419.35
17	2580.64	2903.22	1612.90
18	2580.64	2903.22	3225.80
Weight (N)	20468.39	20536.64	16155.54

Table 12: Node locations in the geometry solution for 21-bar truss

Node No.	Initial Coordinates		Optimized Coordinates	
	x (m)	y (m)	x (m)	y (m)
8	3.50	4.00	2.50	4.25
9	9.25	5.50	8.50	7.50
10	14.75	5.50	15.50	7.50
11	20.50	4.00	21.50	4.25

Table 13: Products in the geometry solution for 21-bar truss

Cluster No.	Product details		Member numbers
	Cross-section	Material	
1	Bar 40×40	QS	1,2,5,6
2	Bar 55×55	HS	3,4
3	Tube $200 \times 200 \times 10$	CS	7,8,9,10,11
4	Bar 45×45	CS	12,14,15,18,19,21
5	Tube $120 \times 120 \times 6$	CS	16,17
6	Bar 30×30	CS	13,20

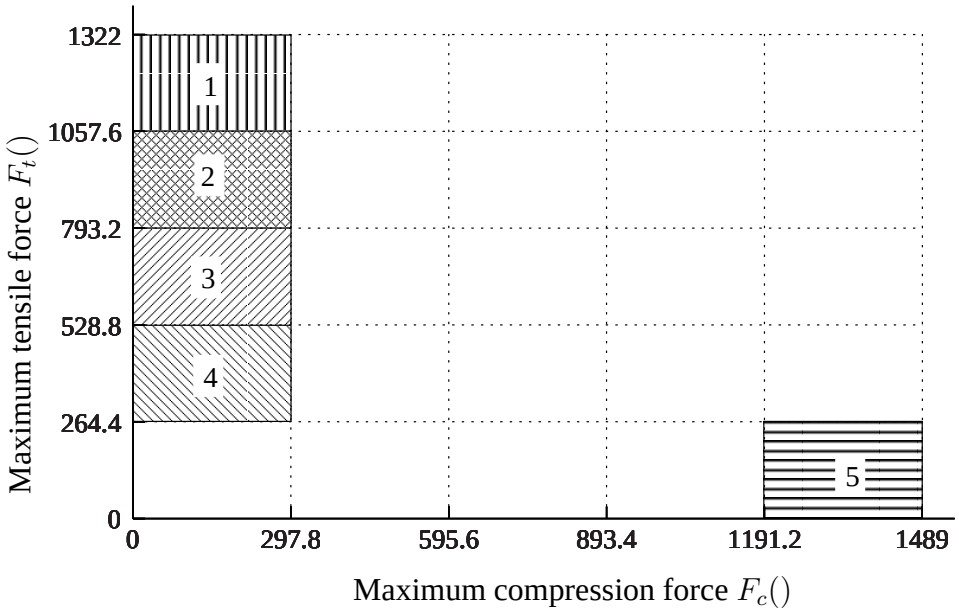


Figure 1: Formation of clusters in the optimal solution for the 21-bar truss shown in Figure 2. The members belonging to each cluster are given in Table 5

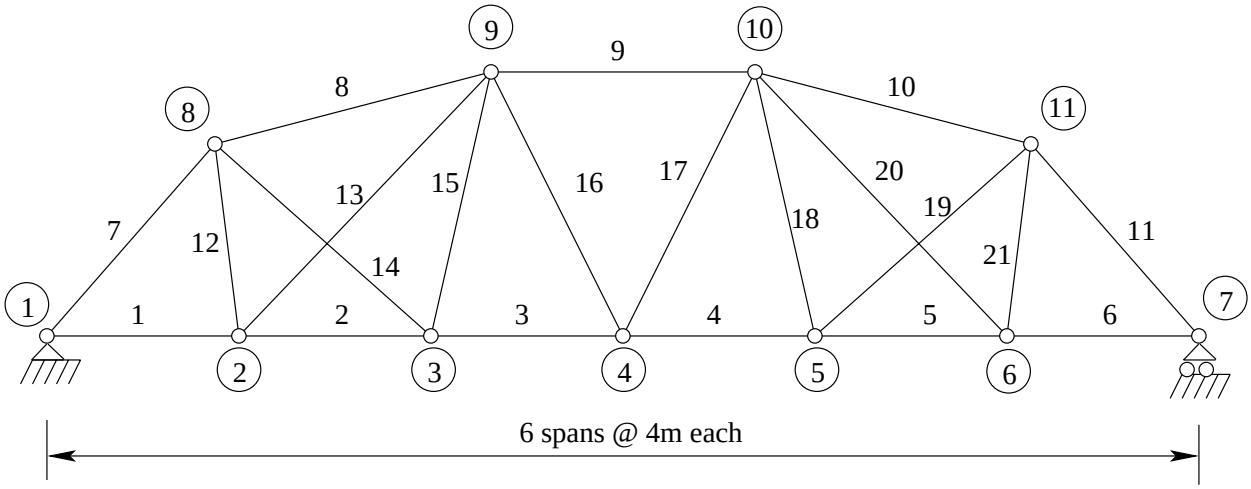


Figure 2: Topology of the 21-bar truss

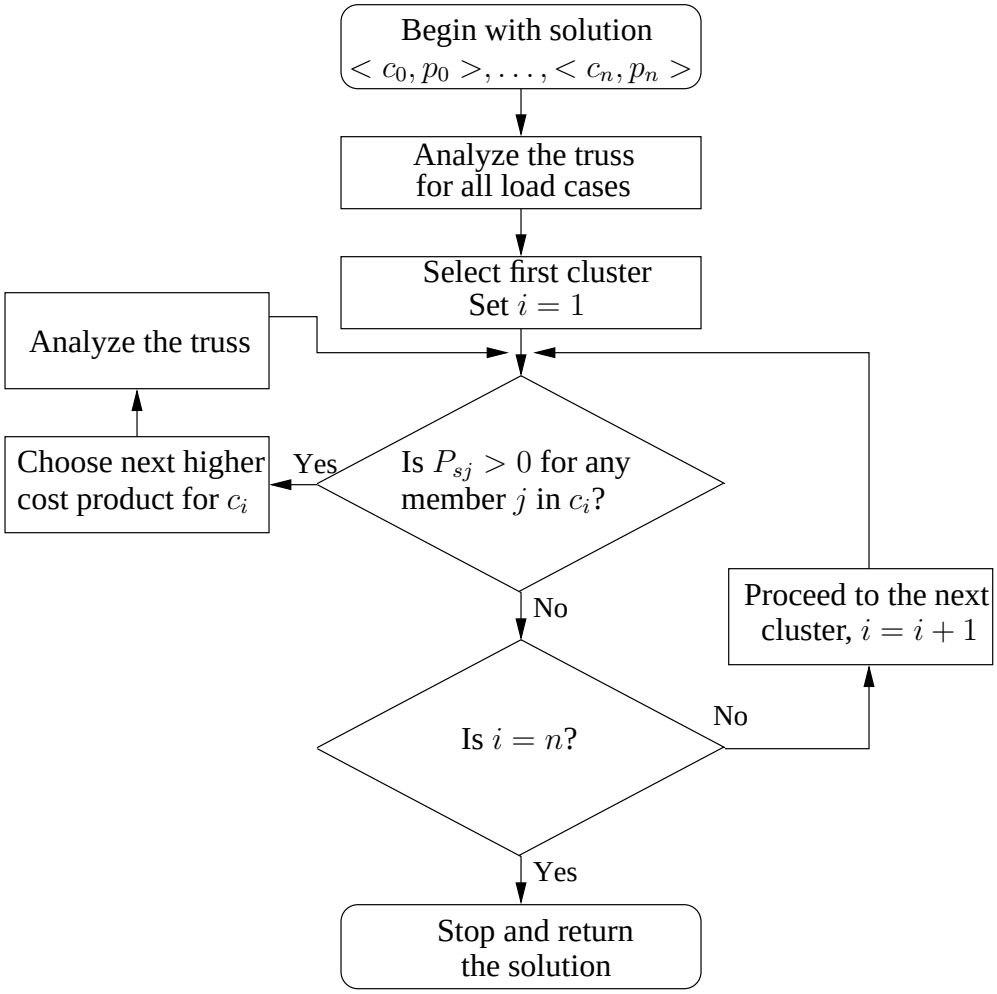


Figure 3: Flowchart for algorithm that forces solution to satisfy stress constraints

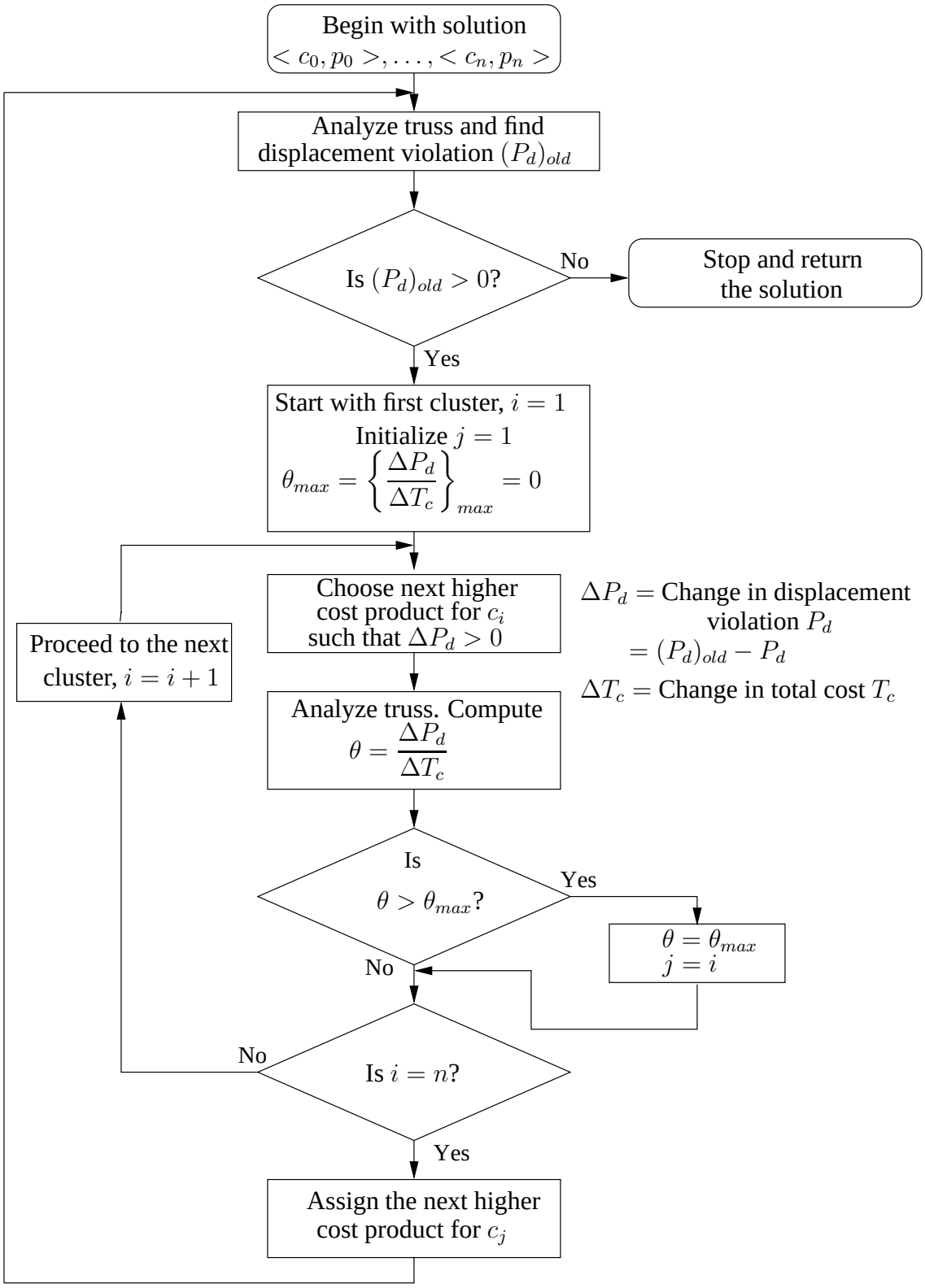


Figure 4: Flowchart for algorithm that forces solution to satisfy displacement constraints

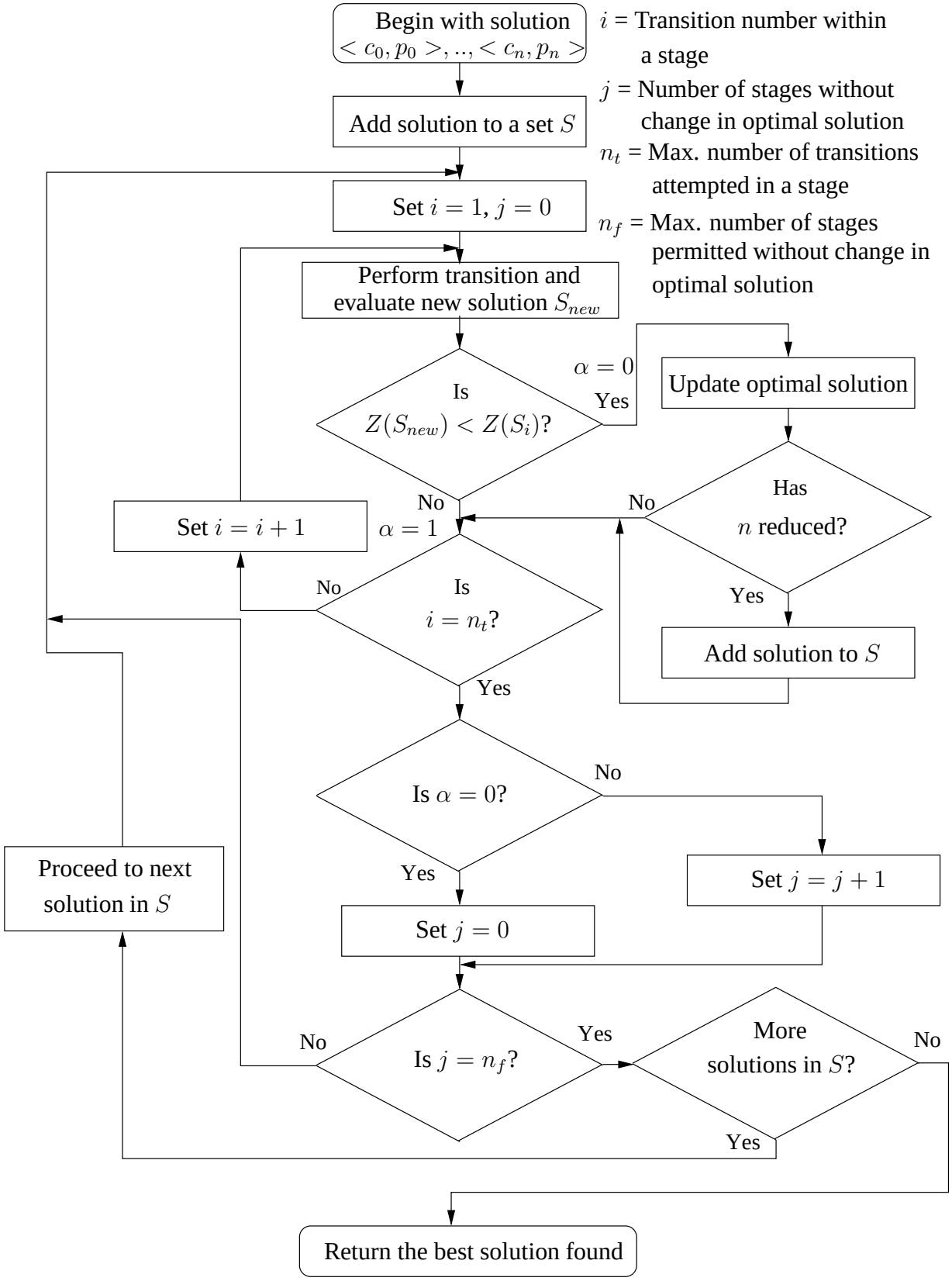


Figure 5: Flowchart for algorithm 2

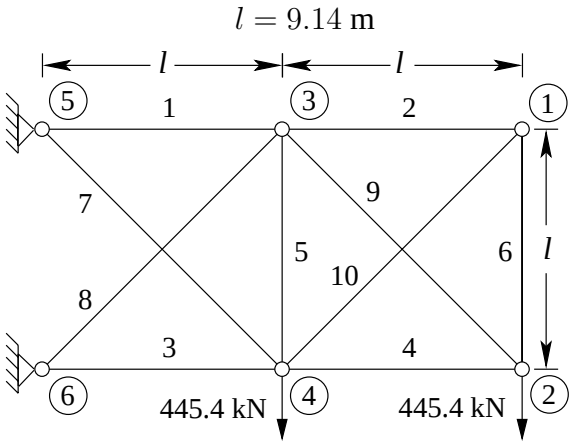


Figure 6: Topology of 10-bar truss

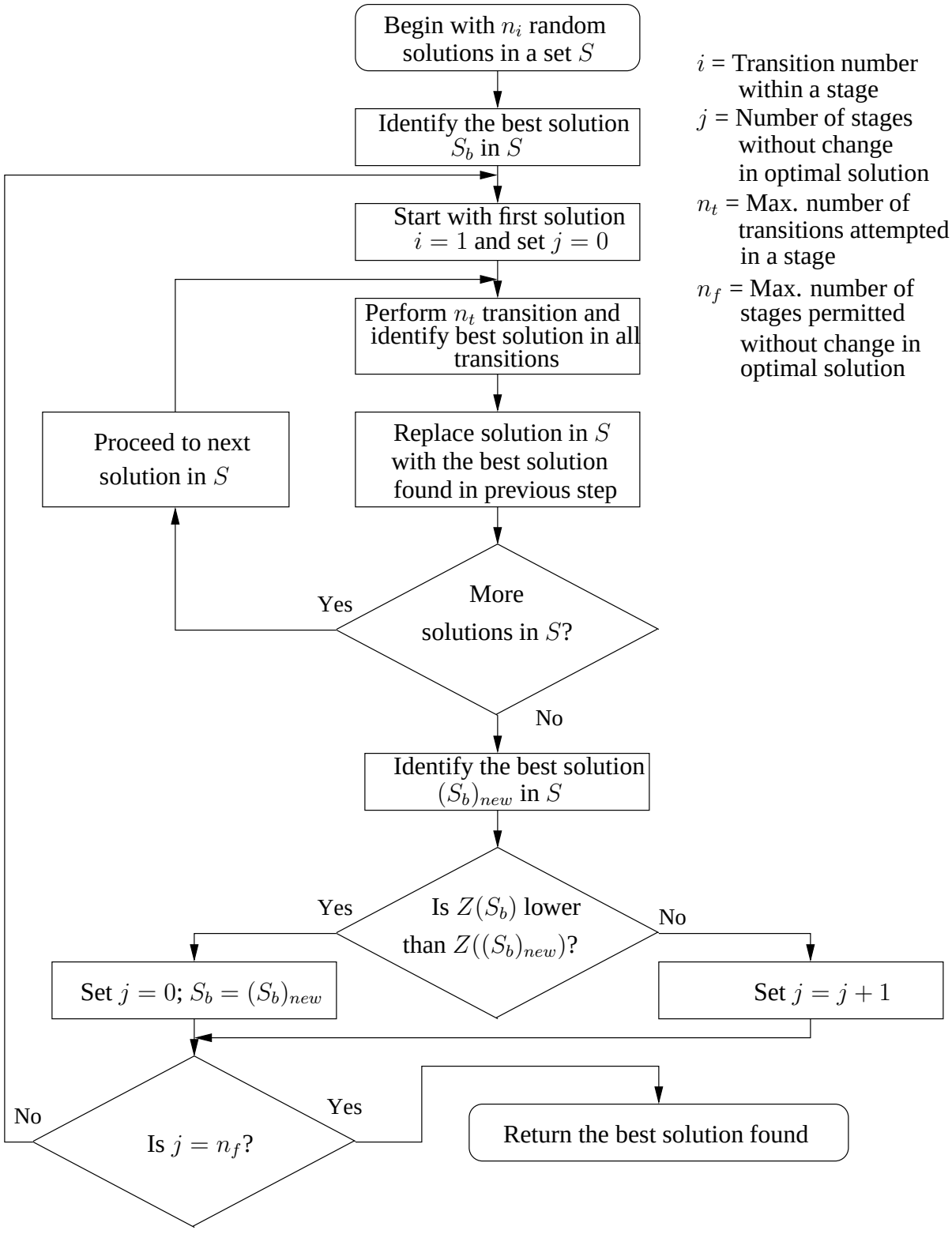


Figure 7: Flowchart of geometry optimization algorithm

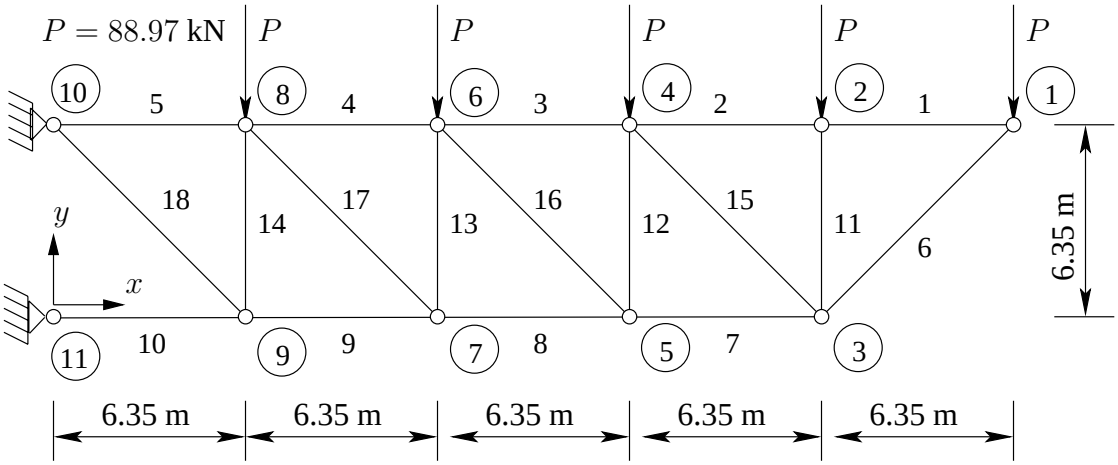


Figure 8: 18-bar truss

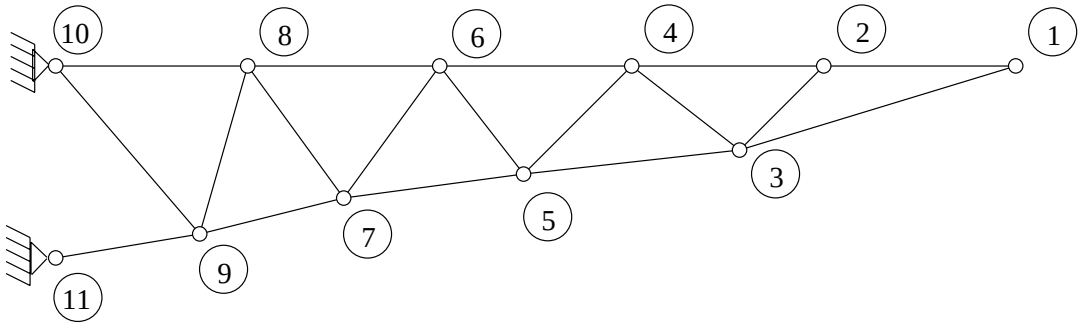


Figure 9: Solution for 18-bar truss evaluated by Rajeev and Krishnamoorthy [15]

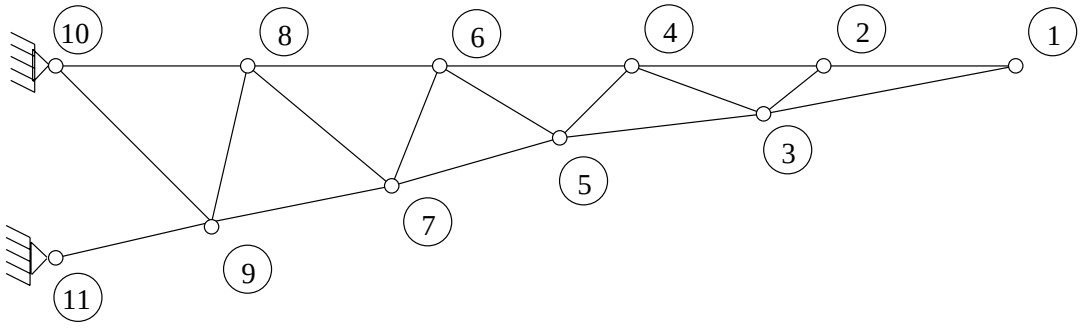


Figure 10: Solution for 18-bar truss obtained using the proposed approach - Case A

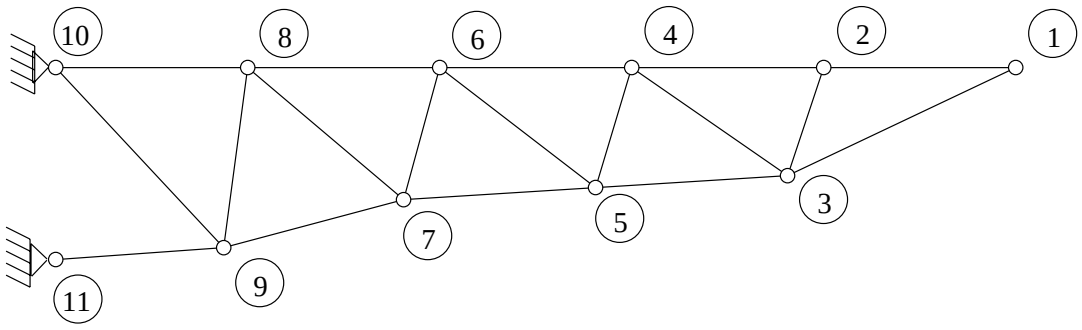


Figure 11: Solution for the 18-bar truss obtained using the proposed approach - Case B

Part III

GENETIC ALGORITHMS FOR DESIGN OF MOMENT-RESISTING STEEL FRAMES

Prakash Kripakaran, Brian Hall, and Abhinav Gupta

To be submitted to Computers & Structures

Genetic Algorithms for Design of Moment-Resisting Steel Frames

Prakash Kripakaran, Brian Hall, and Abhinav Gupta

Abstract

Computational approaches that can be implemented in a Decision Support System (DSS) for design of moment-resisting steel frames are presented in this study. Genetic Algorithms are used to find the rigid connection locations in the frame that minimize a cost function, which includes the connection cost as well as the material cost of the frame. The cost estimates used in the study are obtained from consultation with practicing engineers and erectors. The problem formulation includes two sets of decision variables - (1) binary variables that represent the presence or absence of a rigid connection at a particular location and (2) integer variables that represent the product type assigned to a particular frame member. As engineers obtain an idea of the redundancy of a frame from the number of rigid connections, GA-based approaches for performing trade-off study between the number of rigid connections and the total cost of the frame are proposed to provide insight during the decision-making process. The trade-off curve is obtained by finding the optimal cost solutions for different specified values of number of rigid connections. A seeding technique that creates an initial population of solutions that have exactly the specified number of rigid connections is developed. A unique crossover scheme, which ensures that the offspring have the same number of rigid connections as the parents, is implemented. Algorithms that correct solutions violating strength and serviceability constraints are proposed to avoid using penalty costs in the fitness function. The trade-off study is performed for a real-life 5-bay 5-story frame under two different support conditions - all the supports are fixed and all are hinged. The results indicate that the total cost of the frame is minimal when the rigid connections are present only at certain locations. Finally, “Modeling to Generate Alternatives - MGA,” is proposed to generate good design alternatives as the solution from optimization may not be optimal with respect to the unmodeled objectives and constraints.

The optimization formulation is modified to generate alternatives using “Hamming distance” as a difference metric. The engineer’s professional judgment and expertise can ultimately decide the alternative used for final design.

1 Introduction

Design of moment-resisting rigid steel frames for resisting lateral loads represents one particular class of problem that can significantly benefit from novel approaches for decision support. While the use of rigid connections in the frame can considerably reduce the moments in the beams and save significant material costs, the cost of fabrication of a rigid connection is often very high due to excessive labor requirements for on-site welding. Depending on the relative cost of connection and the cost of material, an optimal solution is likely to have rigid connections only at certain locations within a frame. In the current practice, all the connections within a rigid frame are assumed to be rigid. Even if an attempt is made to reduce the number of connections within the frame, all the connections within certain bays of a frame are assumed to be rigid. The bays with rigid connections in the final design are identified by a hands-on trial based on professional judgment without any formal consideration of the connection cost. Such frames are not likely to be optimal when connection cost is included.

A recent state-of-the-art publication by ASCE on optimal structural design [5] recommends optimization formulations to consider the cost of fabrication and erection. In general, moment connections are modeled as either fully-rigid or hinged for analysis purposes. Recent studies [6, 7, 8] have found that modeling the semi-rigid behavior of connections can lead to lighter structures. Most of the rigid frame optimization studies, even those that model the semi-rigid behavior of connections [12, 2, 11], focus on minimizing the total weight of the structure and do not incorporate the fabrication cost of the connection in their optimization model. Only a few frame optimization studies have incorporated both the connection cost and the cost of material. Simoes [13] developed a linear programming approach for cost optimization whereas Xu and Grierson [14] formulated an optimal semi-rigid frame design problem in which the cost of connections was represented by an equivalent weight of steel. Hayalioglu and Degertekin [10] used Genetic Algorithms for optimization with this cost model.

While minimizing the total cost is a primary objective of design optimization, often other objectives like redundancy are given consideration during the decision-making process. Since the number of rigid connections in a frame is directly related to its redundancy, a natural engineering judgment is to select designs that have a higher number of rigid connections leading to a higher total cost. Trade-off curves between these competing objectives can provide valuable insight into the problem. In addition to these quantifiable objectives like redundancy and cost, design problems often involve certain objectives or constraints that are excluded from the optimization model because of their unquantifiable nature. As a result, the final solution is rarely the final design adopted by a decision-maker. An example of such an unquantifiable parameter can be an additional cost related to erecting a frame in a congested urban setting. This cost can be difficult to quantify but is often a critical factor in determining the design chosen for implementation. Therefore, it would be preferable to first evaluate a set of alternatives that are good with respect to the modeled objectives and constraints. Then, select an alternative that is also good with respect to the unmodeled objectives and constraints based on professional judgment. Computational approaches that use optimization to not only find the "best" solution, but also generate a set of "good" alternatives permit the incorporation of professional expertise for arriving at a final design and for conducting investigative studies like *trade-off* and *what-if* analyses.

In this paper, we propose computational approaches that can become part of a decision support system for the design of moment-resisting frames. The study uses a cost function that considers the material cost of steel as well as the cost of rigid connections in the frame. Practicing engineers and erectors were consulted to arrive at the cost estimates considered in this study. For simplicity, connections are considered either fully-rigid or hinged. Semi-rigid behavior is not considered in this exploratory study. Unlike many optimization formulations in which the beam and column member types are the only decision variables, the proposed formulation uses binary decision variables that correspond to the presence or absence of a rigid connection at each beam-column junction. We propose Genetic Algorithms (GA), that use a unique crossover scheme, for finding the minimum cost design. The use of penalty functions for stress and displacement constraints is avoided by developing novel algorithms to change infeasible solutions into feasible solutions. Trade-off studies between the number of rigid

connections and the total cost are performed using the GA with this unique crossover scheme. "Modeling to Generate Alternatives - MGA," which is an optimization-based technique, is used to identify alternatives that are far apart in decision space but close to the optimal solution in objective space. The solutions from the MGA are examined with respect to various design aspects that are not formally considered in the optimization model.

2 Problem Definition

An m-story, n-bay frame consists of $(m \times n)$ beams. Each beam has two connections, one at each junction with the columns. Let us number the possible rigid connection locations from 1 to R where $R = 2(m \times n)$ is the total number of rigid beam-column connection locations in the frame. Let c_i represent the decision variable corresponding to the presence or absence of a rigid connection at location i, $i = 1..R$. The presence of a hinged connection at location i is represented by $c_i = 0$ and the presence of a rigid connection by $c_i = 1$. Then, the binary string $< c_1, c_2, c_3, c_4 ... c_R >$ represents the decision variables that correspond to the various beam-column connections in the frame. In many cases, the number of connection decision variables may be less than $2(m \times n)$ because of considerations of symmetry in the frame. In such a case, R would represent the total number of connection locations after considering symmetry. The other set of decision variables in the problem is related to the product types p_j for the various members in the frame. Let $< p_1, p_2, ... p_{2mn+m} >$ represent the decision variables that correspond to the product type for each of the $(2mn + m)$ members in the frame. The total cost C of the frame is expressed as,

$$\text{Total cost}, C = T_s + T_c \tag{1}$$

in which T_s and T_c correspond to the material cost and the cost of connections, respectively. T_s is computed as,

$$T_s = C_s \sum_{j=1}^{2mn+m} w_j(p_j) l_j \tag{2}$$

where w_j is the weight in tonnes per unit length of the product p_j, which is the product (or member type) assigned to member j. l_j is the length of member j. C_s is the cost per tonne of

steel. The total cost of the connections, T_c, is computed as,

$$T_c = C_r \sum_{i=1}^{R} c_i \tag{3}$$

C_r is the cost of a single rigid connection. In order to determine C_r, detailed discussions were held with practicing engineers, steel fabricators and erectors. The actual cost of a connection depends upon various factors including the type of connections, amount of welding, web stiffening, doubler plates, etc. However, we adopted a fixed cost of $\$900$ per connection, which is representative of the average value for fabricating a rigid connection in the state of North Carolina. Furthermore, we consider $C_s = \$600$ per tonne of steel. Since the material cost of steel varies over time, we evaluate the sensitivity of the results calculated in our study by varying the ratio of connection cost to material cost, C_r/C_s, from 1.0 to 2.0.

The constraints for the design problem are prescribed by the strength and serviceability requirements specified in the Manual of Steel Construction, Load and Resistance Factor Design [1]. According to the manual, the members in the frame must conform to the following design equations for each load case.

For $P_u/\phi P_n \geq 0.2$,

$$\frac{P_u}{\phi P_n} + \frac{8}{9}\frac{M_u}{\phi_b M_n} \leq 1.0 \tag{4}$$

For $P_u/\phi P_n \leq 0.2$,

$$\frac{P_u}{2\phi P_n} + \frac{M_u}{\phi_b M_n} \leq 1.0 \tag{5}$$

P_u is the factored axial load (tensile or compressive) in the member, P_n is the nominal (tensile or compressive) strength in the member and ϕ the corresponding resistance factor. For tension, $\phi = 0.75$ in Equation 4 and $\phi = 0.9$ in Equation 5. For compression, $\phi = 0.85$ in both equations. M_u and M_n are the required and nominal flexural strength for major axis bending, respectively. ϕ_b is the resistance factor for flexure and is equal to 0.9. Since the proposed study considers only plane frames, Equations 4 and 5 have only those terms that correspond to major axis bending. P_n is calculated using the area of the member and either the yield stress or the buckling stress depending upon the nature of the axial force in the member, i.e., tension or compression, respectively. M_n is calculated using certain equations that evaluate the ability of the member to resist the following buckling possibilities - local flange buckling, local web

buckling and lateral torsional buckling. These equations are described in detail in AISC [1] and are not provided in this paper for brevity. The serviceability constraint given below governs the maximum side sway δ for any story of height H in the frame.

$$\delta \leq \frac{H}{400} \tag{6}$$

3 Optimization Approach

The goal of optimization is to find values for c_i and p_j that minimize T_c. In a conventional Genetic Algorithm, all variables - c_i as well as p_j, are modeled into the GA representation. The constraints are normally applied by having penalty costs in the objective function of the GA. A key limitation of such an approach is the large number of decision variables when combined with the numerous types of products available for the members result in an excessively large search space that diminishes the efficiency as well as the quality of the search process. However, a closer inspection of this particular problem reveals that the size of the search space can be reduced significantly because there exist certain products corresponding to each p_j that will minimize the total cost of that frame. Therefore, it is not necessary to consider p_j as the decision variable in the GA representation. The optimization formulation needs to model only c_i as decision variables.

We propose a GA-based optimization approach that is focused on finding the optimal values of c_i and hence, the best locations for rigid connections in a frame. The proposed approach involves two key steps. In the first step, we determine a minimum-weight frame. To do so, we assume that all connections in the frame are rigid and find the products q_j for the various members that minimize the total weight of the frame. In the second step, we use a GA to perform trade-off study between the number of rigid connections and total cost. Trade-off study is performed by conducting a series of GA runs, where each GA run is aimed at finding the solution with minimal total cost for a specified number r_{req} of rigid connections. During a GA run for a specified r_{req}, the GA explores the solution space with exactly r_{req} rigid connections. To evaluate the quality of a particular connection configuration, the GA analyzes the frame with the particular connection configuration and q_j as its products. Since the solution has rigid connections at only certain locations, the members in the frame may violate the sway

and/or LRFD requirements. Solutions that violate these constraints are transformed into feasible solutions by using algorithms that change the member sizes (product types) to find the minimal-cost feasible solution for that connection configuration. The various components of the proposed approach are described in the following sections.

4 Algorithm for Least-Weight Frame

As stated earlier, we first find the minimum weight solution when all the connections in the frame are rigid connections, i.e, $\{c_i = 1, \forall\ i\}$. We generate this solution by following the simple iterative procedure outlined below. This procedure is similar to the one that can be performed within a commercial finite element analysis package like SAP2000. While the algorithm presented here is not guaranteed to find the true least-weight solution, its purpose is to evaluate an initial set of product types q_j for the members in the frame that can later facilitate finding p_j within the GA implementation.

Step 1: Model the frame with only rigid connections.

Step 2: Assign the smallest W-shape member to all the beams and columns.

Step 3: Analyze the structure considering P-δ effects for the various load combinations and evaluate the acceptability with respect to the various strength design requirements.

Step 4: For members that violate the design requirements, assign larger W-shapes such that the members satisfy the design requirements for the forces evaluated in the previous step.

Step 5: Repeat steps 3-4 until all members satisfy the strength design requirements.

Step 6: Check if the sway requirements are met for each floor. Commercial packages like SAP2000 do not implement floor sway constraints in their current versions.

Step 7: Assign larger W-shapes for all beams and columns in a floor that violate sway requirements such that the increase in moment of inertia (ΔI_{req}) for each member satisfies the following inequality.

$$\Delta I_{req} = \frac{(\delta_u - \delta)}{\delta} I(p_j) \tag{7}$$

δ is the floor sway obtained from analysis and δ_u is the maximum allowable floor sway. $I(p_j)$ is the moment of inertia of product p_j that is currently assigned to member j. The premise for using Equation 7 is based on the direct relationship of the displacement to bending stiffness of

beams and columns.

Step 8: Repeat steps 6-7 until sway requirements are met.

The above algorithm is not computationally intensive and converges to a feasible solution after only a few iterations over both the strength and the sway constraints. Additional constraints may also be considered. For example, the W-shape of a column member in any floor must be larger than that of any column member above it. Such rules can also be appropriately implemented in the above algorithm while performing steps 4 and 7. Let q_j represent the W-shape assigned to member j in the solution that is returned by this algorithm. The set of q_j is used in the algorithms of the GA implementation, which is described in the following section.

5 GA Implementation

The GA approach presented in this paper is developed to generate the trade-off curve between the number of rigid connections, r_{req} and the total cost, T_c. As mentioned previously, this curve is generated using a series of GA runs, where each GA run is aimed at minimizing the total cost C for a specified r_{req}. The variables c_i are the only decision variables that are affected by the GA operators. We use a binary string representation in the GA that is given as $< c_1, c_2, ...c_R >$.

5.1 Fitness Evaluation

Given a particular binary string, the GA evaluates its fitness in the following manner.

Step 1: Consider a frame with the products q_j, as determined using the algorithm to find least-weight frame, assigned to its various members and having the connection configuration represented by the binary string.

Step 2: Determine if the frame satisfies strength constraints. If not, use the algorithm given in Figure 1 to transform this solution into one that satisfies strength constraints.

Step 3: Determine if the frame satisfies floor sway constraints. If not, use the algorithm given in Figure 2 to transform the solution into one that satisfies floor sway constraints.

Step 4: Evaluate the fitness of the solution using the following fitness function,

$$Z = -(C + P_s + P_d + P_c) \tag{8}$$

in which C is the total cost of the frame whereas P_s, P_d and P_c are the penalty costs that correspond to the various constraint violations, i.e., strength, sway and number of rigid connections, respectively. P_s and P_d are applied only in the event that the algorithms used in steps 2 and 3 are unable to force the solution to satisfy the respective constraints. These penalty costs are calculated as follows.

$$P_s = a \sum_{j=0}^{2mn+m} p_s \tag{9}$$

$$p_s = \begin{cases} 0, \text{ if the corresponding design equation is satisfied} \\ \text{LHS of corresponding design equation, if member violates design equation} \end{cases} \tag{10}$$

The design equations that are referred to above are Equations 4 and 5. a is a constant multiplier and is set equal to $100,000$. Penalty cost P_d is evaluated as,

$$P_d = b \sum_{j=0}^{2mn+m} p_d \tag{11}$$

$$p_d = \begin{cases} 0, \text{ if member } j \text{ is a beam or } \delta_j - \delta_{max} \leq 0 \\ \delta_j - \delta_{max}, \text{ if member } j \text{ is a column and } \delta_j - \delta_{max} \geq 0 \end{cases} \tag{12}$$

δ_j is the floor sway for column j and is simply the absolute difference of the horizontal displacements at the two end nodes of the member. δ_{max} is the maximum allowable sway as specified by Equation 6. b is a constant multiplier and is equal to $100,000$. The penalty cost, P_c, for enforcing the number of rigid connections is calculated as,

$$P_c = c|\sum_{i=1}^{R} c_i - c_{req}| \tag{13}$$

where c is a constant multiplier. c_{req} represents the number of decision variables c_i that must equal 1 for the string to represent exactly r_{req} number of rigid connections in the frame. If symmetry is not used to reduce the number of decision variables, $c_{req} = r_{req}$. Otherwise, $c_{req} < r_{req}$. P_c is necessary to perform a trade-off study when using traditional crossover schemes within the GA. As discussed in detail in the following sections, we implement an unique crossover and mutation scheme to avoid the use of P_c.

5.2 Seeding in GA

Since a particular GA run must consider only those solutions that have a specified number of rigid connections, r_{req}, strings in the GA must satisfy the following property, $\sum c_i = c_{req}$ or a corresponding penalty P_c is applied as given in Equation 13. Seeding the GA appropriately can minimize the number of solutions on which this penalty is used. We use the following seeding technique that ensures that all solutions in the initial population have exactly the specified r_{req} number of rigid connections.

$$c_i = \begin{cases} 1, \text{ for } i \in S \\ 0, \text{ for } i \notin S \end{cases} \tag{14}$$

S is a set of cardinality c_{req} with elements randomly chosen from the collection $\{1, 2, 3...R\}$ such that no two elements in S are equal.

5.3 GA Operators

In an attempt to improve the quality of solutions in a subsequent generation, the GA uses the crossover operator to produce two offspring by combining two parent strings. Traditional GA approaches use uniform crossover or one-point crossover. These conventional crossover operators generate solutions in which the number of rigid connections is different from r_{req}. This behavior results in the application of penalty, P_c, to many solutions even though the seeding technique ensured that all the solutions at the start of the GA only have a total of r_{req} rigid connections. Such a situation is avoided by using the crossover operator proposed by Gupta et al. [9]. This crossover operator has the unique characteristic of generating offspring that have an equal number of ones when each of the parent strings also have the same equal number of ones. This crossover is illustrated in Figure 3. Let us consider two parent strings, which have exactly c_{req} number of ones, given by $< c_{a1}, c_{a2}, \ldots, c_{aR} >$ and $< c_{b1}, c_{b2}, \ldots, c_{bR} >$. Let us define the set $S = \{i : c_{ai} \neq c_{bi}\}$. Note that S, whose cardinality is always even, can be randomly divided into two sets of equal sizes S_1 and S_2. The offspring, $< c_{c1}, c_{c2}, \ldots, c_{cR} >$ and $< c_{d1}, c_{d2}, \ldots, c_{dR} >$, are then given by:

$$c_{ci} = \begin{cases} c_{ai}, \ \forall\, i \notin S \\ 1, \ \forall\, i \in S_1 \end{cases} \tag{15}$$

$$c_{di} = \begin{cases} c_{ai}, \ \forall \, i \notin S \\ 1, \ \forall \, i \in S_2 \end{cases} \tag{16}$$

This crossover scheme effectively reduces the size of the search space, which is 2^R when using uniform crossover, to ${}^{R}C_{c_{req}}$. A crossover probability of 0.75 is used for the study.

GAs use the mutation operator to make random changes to existing solutions in a population. The premise for mutation is that random changes may result in solutions that lie in those regions of the decision space that were not explored in the previous generations. Mutation probability is kept very small at 0.1% to avoid any significant interference with the GA convergence. For the trade-off study, the decision space consists of all solutions that have exactly r_{req} number of rigid connections, i.e., c_{req} number of ones. Therefore, we use the following mutation operator, which makes random changes but does not alter the number of ones in the string. Let $< c_1, c_2, \ldots, c_R >$ represent the solution. Let $S_1 = \{i : c_i = 1\}$ and $S_2 = \{i : c_i = 0\}$ represent the locations that have rigid connections and hinged connections in the original solution. For mutation, set $c_k = 0$, where k is randomly chosen from S_1 and set $c_l = 1$, where l is randomly chosen from S_2.

Binary tournament selection with a selection probability of 0.75 is employed in the GA. The convergence criteria is determined based on the number of consecutive generations for which there is no change in the global optimal solution. If this number exceeds a specified value, then the GA is considered to have converged to the optimal solution. We have used 10 generations as the criterion for convergence.

6 Plane Frame Example - Description

We study the performance of the proposed optimization approach for a real-life 5-story 5-bay frame. This frame is one of two end frames of a commercial building and is primarily used to resist the lateral forces arising due to wind in one direction. The elevation view of the frame is shown in Figure 4. The figure shows the numbering of the members and the joints in the frame.

Load factors and load combinations for design are taken from the 3^{rd} edition of AISC Manual of Steel Construction for Load and Resistance Factor Design [1]. The strength of

moment frame is evaluated for the following load combinations - $1.4D$, $(1.2D + 1.6L + 0.5L_r)$ and $(1.2D+1.6W+0.5L+0.5L_r)$, where D, L, W and L_r represent dead loads, live loads, wind loads and roof live loads respectively. Note that there are effectively four load combinations as the wind load results in two different load cases that correspond to the wind blowing in either direction. The details of all the loads are summarized in Tables 1, 2 and 3. The serviceability of the building is evaluated for two load cases corresponding to only the wind loads acting on either side of the frame.

The W-shapes for the members of the frame are chosen from the shapes listed in the AISC Manual of Steel Construction for LRFD [1]. Engineers follow certain guidelines for assigning products to different frame members. The following guidelines are used in this study.

- The AISC manual has a table of W-shapes that have the highest moment of inertia (I_x) for lowest weight per unit length among groups of W-shapes with similar order of magnitude of I_x. These W-shapes that have the largest I_x for minimal weight are best for flexural design. Beams are assigned products from a separate product set with only these W-shapes.
- All members that belong to a particular column in the frame use W-shapes that have the same web depth. Engineers advise using W14 shapes for columns and hence, we have a separate product set that is composed of only W14 shapes for columns.

These two product sets are given in Tables 4 and 5.

There are 50 possible locations for rigid connections in the frame. Due to the symmetry in both the loading and the geometry, only 25 decision variables - $< c_1, c_2 ... c_{25} >$ are needed in the optimization formulation. While the frame has 55 members, only 30 decision variables for the products - $< p_1, p_2 ... p_{30} >$, are needed due to symmetry.

7 Results

As a first step, we use the algorithm to find the least weight solution when the frame consists of only the rigid connections, i.e., all 50 connections are considered to be rigid. The W-shapes q_j assigned to the various members in the solution from this step are provided in Table 6. The weight of steel in this solution is approximately 22 tonnes. Since all 50 connections are rigid, the total cost of the frame = $50 \times 900 + 22 \times 600 = \$58,200$. During this process, it is observed

that the floor sway constraints govern the search for this solution. This solution will violate the sway constraints if the frame is assigned a fewer number of rigid connections.

In the second step, we generate the trade-off curve using the GA. A population size of 200 is used in the GA. The trade-off curve is generated for $r_{req} = 4...36$. Due to the use of symmetry, $c_{req} = r_{req}/2$ in the GA. The resulting curve is shown in Figure 5. The solutions plotted on the curve are the best solutions generated over a series of five GA runs each with a different random starting population. The trade-off curve clearly shows that there is a certain number of rigid connections, r_{opt}, for which the total cost is optimal. As $|r_{req} - r_{opt}|$ increases, the total cost of the solution also increases. From the figure, $r_{opt} = 10$. This optimal solution has a weight of 33 tonnes and a total cost of $\$30,609.45$. The rigid connection locations in the optimal solution are shown in Figure 6. The W-shapes assigned to the various members in the optimal solution are given in Table 7.

Often, high costs associated with footings may force the designers to consider the connections at the base of the frame as hinged. The trade-off study is repeated for such a case. Instead of using the algorithm for finding the least-weight frame, values for q_j from the previous case, which are given in Table 6, are used. The results of this trade-off study are also plotted in Figure 5. In this case, the trade-off curve is linear and the total cost appears to decrease with the number of rigid connections. It is observed that there are no feasible solutions for $r_{req} < 6$. The cost optimal solution is for $r_{req} = 6$ and its details are given in Figure 7 and Table 8.

Since the values for C_r/C_s are expected to change with time, we examine the sensitivity of the study for different specified values of C_r/C_s. We perform the study only for the case when all the supports in the frame are fixed. The trade-off study is repeated for five different values of C_r/C_s between 1.0 and 2.0. Figure 8 and Table 9 summarize the results obtained. The table gives the least cost solution and the particular values of C_s and C_r for each curve shown in the figure. As can be perceived by intuition, for increasing values of C_r/C_s, significant cost savings can be achieved by minimizing the number of rigid connections. Since practicing engineers are predicting that C_r/C_s is bound to keep increasing in the future, optimization approaches to minimize the number of rigid connections will have a key role to play in the design process. In the following sections, we present approaches to generate design alternatives.

8 Generating Design Alternatives

The solutions generated using optimization are not necessarily the best for adoption as a final design because such solutions are optimal only with respect to the modeled costs and constraints. Often, many objectives and constraints are not explicitly stated in the problem formulation due to their unquantifiable nature. The solutions produced using an optimization approach may therefore be sub-optimal with respect to these unmodeled issues. Traditionally, this has been cited as a major limitation of optimization applications. Modeling to Generate Alternative (MGA) techniques [3, 4] use optimization to generate a small set of very different solutions. These are an extension of single and multiple-objective mathematical programming techniques with an emphasis on generating a set of alternatives that are "good" but "as different as possible."

To illustrate MGA, consider the following formulation of a design optimization problem:

$$\begin{aligned} \text{Maximize } z = x + 2y, \quad \text{subject to } x + y &\leq 30 \\ y &\leq 20 \\ x, y &\geq 0 \end{aligned}$$

The decision space for the problem is shown as the shaded region in Figure 9a and the slope of the objective function can be seen from a typical contour line at an arbitrary value, 80. Mathematical optimization correctly produces z = 50 at point A as the solution. Thus, all other things being equal, the design would be $< x, y >=< 10, 20 >$. The premise of MGA, however, is that all other things may not be equal, and that there may be features not completely captured by the model. When those issues are considered, point A may be less desirable overall than a point, call it B, originally deemed inferior by the model. The issues involved in finding B and establishing the computer assistance needed in this process are discussed below.

- Generating all feasible solutions suggests that one has no confidence in the model - one would expect that point B optimizes the objective function nearly as well as A, so only these good solutions need to be examined. This idea is illustrated in Figure 9b, where only those solutions that are within 10% of the optimal are retained.
- Available solutions should represent a cross-section of good solutions, so that, if B is not actually among them, perhaps one of them is close enough from which to begin "tinkering."

- Since a decision maker can reasonably consider only a small number of designs, a subset of these good solutions should be presented for further consideration.

For the problem defined above, the first alternative can be found by changing the formulation to maximize a difference metric δ, instead of z and imposing the constraint, $x + 2y \geq 0.9(50)$. The difference metric δ for this problem can be defined as $\delta^2 = (x-10)^2 + (y-20)^2$, which is the distance between the two points in the decision space. Similarly, MGA techniques can be applied for the design of steel moment frames if an appropriate difference metric can be devised.

9 MGA for Moment Frame Design

We suggest using the "Hamming distance," which is a metric used in binary computation, as the difference metric for the MGA. If two solution strings are represented as $< c_{1a}, c_{2a}, .., c_{Ra} >$ and $< c_{1b}, c_{2b}, \ldots, c_{Rb} >$, we can write,

$$\delta_{ab} = \sum_{i=1}^{R} |c_{ia} - c_{ib}| \tag{17}$$

where δ_{ab} is the difference between the two solutions. Let us consider that we have generated n alternatives and are currently in the process of generating the $(n+1)$th alternative. Then, we attempt to maximize the following difference metric to generate the $(n+1)$th alternative.

$$\delta = \sum_{j=1}^{n} \delta_{aj} \tag{18}$$

Since the original model is accurate to a large extent, designers are interested in alternatives for which the costs are only slightly higher than the optimal cost. Therefore, we add the following constraint to the optimization formulation.

$$C_{mga} \leq kC_{opt} \tag{19}$$

where $k > 1$ and, C_{mga} and C_{opt} are the costs associated with the alternative and the optimal solution respectively;

The alternatives are generated by modifying Equation 8 to maximize δ instead of minimizing cost and adding an additional penalty to account for the constraint given by Equation 19.

The value of k is taken to be 1.1, i.e., the search is for alternatives whose total costs are higher than the cost of the optimal solution by not more than 10%. Using this formulation, we generate alternatives for $r_{req} = 12$, when all the supports in the frame are fixed. These alternatives are evaluated on the basis of the following criteria - (1) preferences to certain locations for rigid connections, (2) margins against excessive lateral loads, and (3) number of W-shapes used in a single column of the frame.

The optimal cost solution for $r_{req} = 12$ is given in Figure 10. The alternatives - $MGA1$ and $MGA2$, and their costs for $r_{req} = 12$ are shown in Figures 11 and 12. The member types for these solutions are given in Tables 10, 11 and 12. These two solutions along with the cost-optimal solution constitute a small solution set which the designer can explore with respect to the unmodeled objectives. All three solutions have significantly different locations for rigid connections. While the model assumes that all the rigid connections have the same cost, this is not entirely true in practice. The connection cost varies depending on the products used for the corresponding beam and column which in turn determine the required weld specifications, fabrication and stiffness requirements. In the optimal solution, the largest beam with a rigid connection uses W18×35. In $MGA1$, the largest beam with a rigid connection is member 19 with W24×68. In $MGA2$, the largest beam with a rigid connection is also member 19 but with W24×62. $MGA1$ and $MGA2$ have costs of $\$33,613$ and $\$30,884$ respectively. Also, $MGA1$ has most of its rigid connections on the innermost bay while $MGA2$ and the optimal solution have the rigid connections primarily in the outermost bays. Since rigid connections require on-site welding, the engineer may consider construction aspects to determine if one alternative is superior to another. Specifically, engineers may prefer to have the rigid connections on the inner bays in an urban setting.

Another parameter that an engineer may consider to evaluate the relative quality of the alternatives is related to the ability of the frames to withstand increased lateral loads. While the generated solutions satisfy the strength constraint, it is possible that certain solutions have greater margins than others. An elegant way to compare the alternatives on this basis is by gradually increasing the wind load and identifying the wind load at which the different design alternatives fail. It is observed that the optimal solution fails when the wind load is increased by 95%. On the other hand, the alternatives $MGA1$ and $MGA2$ fail when the wind load is

increased by 123% and 117%, respectively. Figures 10, 11 and 12, which give the optimal solution and the alternatives when the base is fixed and $r_{req} = 12$, also show the left-hand side values of Equations 4 and 5 for the increased wind loads. The optimal solution, $MGA1$, and $MGA2$ have costs of $\$30,609$, $\$33,613$ and $\$30,884$ respectively. Expertise and judgment can be used to evaluate the alternative that is best suited for final design.

The engineer may also wish to tinker with these solutions and arrive at a better solution. To illustrate this, let us consider the optimal solution and an alternative for the case when $r_{req} = 6$ and the base of the frame is hinged. The respective solutions are shown in Figures 7 and 13. The solutions tend to have different products at each floor level for the same column. Engineers prefer to run the same column for atleast two floor levels as it reduces the the cost of splicing them. Therefore, the engineer can modify the solution so that the bottom two floors have the same W-shape and the top three floors have the same W-shape. The costs of the solutions are re-evaluated and the engineer may use discretion to identify the best among all the alternatives. In this manner, MGA and optimization can be combined to create a joint-cognitive decision-making environment, where the engineer uses computing tools to perform analyses, study trade-off curves and generate alternatives but employs engineering knowledge and experience to make the final decision.

10 Summary and Conclusions

Computational approaches that can assist decision-makers in the design of rigid steel frames are proposed in this paper. Development of optimization-based approaches that enable designers to identify the best locations for rigid connections in a moment frame is presented. The optimization approach involves two components - (1) A simple iterative procedure to find the least weight solution when the frame is considered to have only rigid connections and (2) A Genetic Algorithm (GA) to perform trade-off study between the number of rigid connections and the total cost of the frame. The GA starts with the solution from the first algorithm and attempts to find the cost-optimal solutions as the number of rigid connections are gradually decreased. The GA uses a unique crossover scheme that enables it to find the least cost solution for a specified number of rigid connections by limiting the search to only those solutions in

which the number of rigid connections is exactly equal to a specified number. The performance of this optimization technique is studied for a realistic 5-bay 5-story frame. The trade-off study is conducted for two support cases at the base - (1) all the supports at the base are fixed and (2) all the supports at the base are hinged. For the former case, it is observed that there is a certain number of rigid connections for which the total cost is optimal. For the latter case, it appears that minimizing the total number of rigid connections can also minimize the total cost. Finally, MGA is applied to generate design alternatives since the solutions generated through optimization may not be optimal with respect to certain unmodeled objectives and constraints. We use the "Hamming distance" as the difference metric in the MGA to generate a handful of alternatives for moment frame design. These alternatives are compared with respect to the following criteria - (1) inherent margin in a particular design with respect to increased lateral loads, (2) preference of certain rigid connection locations in the frame based on construction aspects and (3) the number of product changes in a single column of the frame.

The following conclusions are derived from this study.

- The trade-off study shows that the total cost of the frame is optimal when the rigid connections are placed at only a few locations in the frame.
- When all the supports of the frame are fixed, there exists a trade-off between the number of rigid connections and the total cost.
- When all the supports in the frame are hinged, the total cost is proportional to the number of rigid connections in the frame.
- MGA using "hamming distance" generates design alternatives that can prove valuable in arriving at a final design.

References

[1] AISC. *Manual of Steel Construction - Load Resistance Factor Design*. AISC, 3^{rd} edition, 2001.

[2] Y. A. Al-Salloum and T. H. Almusallam. Optimality and safety of rigidly- and flexibly-jointed steel frames. *Journal of Constructional Steel Research*, 35:189–215, 1995.

[3] J. W. Baugh, Jr., S. C. Caldwell, and E. D. Brill, Jr. A mathematical programming approach to generate alternatives in discrete structural optimization. *Engineering Optimization*, 28:1–31, 1997.

[4] E. D. Brill, Jr., J. M. Flach, L. D. Hopkins, and S. Ranjithan. MGA: A decision support for complex, incompletely defined problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(4):745–757, 1990.

[5] S. A. Burns, editor. *Recent Advances in Optimal Structural Design*. ASCE, 2002.

[6] B. S. Dhillon and J. W. O'Malley III. Interactive analysis and design of flexibly connected frames. *Journal of Structural Engineering*, 125(5):556–564, 1999.

[7] C. M. Foley and S. Vinnakota. Inelastic behavior of multistory partially restrained steel frames, Part I. *Journal of Structural Engineering*, 125(8):854–861, 1999.

[8] C. M. Foley and S. Vinnakota. Inelastic behavior of multistory partially restrained steel frames, Part II. *Journal of Structural Engineering*, 125(8):862–869, 1999.

[9] A. Gupta, P. Kripakaran, G. Kumar, and J. W. Baugh Jr. Genetic Algorithm-based decision support for optimizing seismic response of piping systems. *Journal of Structural Engineering*, 131(3):389–398, March 2005.

[10] M. S. Hayalioglu and S. O. Degertekin. Design of non-linear steel frames for stress and displacement constraints with semi-rigid connections via genetic optimization. *Structural and Multidisciplinary Optimization*, 27(4):259–271, 2004.

[11] E. S. Kameshki and M. P. Saka. Optimum design of nonlinear steel frames with semi-rigid connections using a genetic algorithm. *Computers and Structures*, 79(17):1593–1604, 2001.

[12] E. S. B. Machaly. Optimum weight analysis of steel frames with semi-rigid connections. *Computers and Structures*, 23(4):565–574, 1986.

[13] L. M. C. Simoes. Optimization of frames with semi-rigid connections. *Computers and Structures*, 60(4):531–539, 1996.

[14] L. Xu and D. E. Grierson. Computer-automated design of semirigid steel frameworks. *Journal of Structural Engineering*, 119(6):1740–1760, 1993.

Table 1: Horizontal loads (x-direction) for frame given in Figure 4

Load type	Joint	Load (kips)
W	1	7.22
	2	14.43
	3	14.43
	4	14.43
	5	14.43
	6	7.22

Table 2: Vertical distributed loads (y-direction) for frame given in Figure 4

Load type	Members	Load (kips/ft)
D	7-10,17-20,37-40,47-50	1.10
	27-30	0.54
	6,16,36,46	0.30
	26	0.08
L	7-10,17-20,37-40,47-50	1.20
	27-30	0.30
L_r	6,16,26,36,46	0.19

Table 3: Vertical concentrated loads (y-direction) for frame given in Figure 4

Load type	Joints	Load (kips)
D	6,36	1.50
	12,30	6.00
	18,24	11.34
	2,3,4,5,32,33,34,35	10.10
	8,9,10,11,26,27,28,29	14.40
	14,15,16,17,20,21,22,23	43.90
L	2,3,4,5,32,33,34,35	6.00
	8,9,10,11,26,27,28,29	24.00
	14,15,16,17,20,21,22,23	45.50
L_r	6,36	3.84
	12,30	15.36
	18,24	29.12

Table 4: Product set for the beams in frame shown in Figure 4

W10 × 12	W16 × 40	W24 × 68	W33 × 130
W12 × 14	W18 × 35	W24 × 76	W36 × 135
W12 × 16	W18 × 40	W24 × 84	W36 × 194
W12 × 19	W21 × 44	W27 × 84	W40 × 149
W12 × 22	W21 × 48	W30 × 90	W40 × 167
W12 × 26	W21 × 50	W30 × 99	W40 × 183
W14 × 22	W21 × 55	W30 × 108	W40 × 199
W14 × 26	W24 × 55	W30 × 116	W40 × 211
W16 × 26	W24 × 62	W33 × 118	W40 × 215
W16 × 31			

Table 5: Product set for the columns in frame shown in Figure 4

W14 × 22	W14 × 74	W14 × 176	W14 × 398
W14 × 26	W14 × 82	W14 × 193	W14 × 426
W14 × 30	W14 × 90	W14 × 211	W14 × 455
W14 × 34	W14 × 99	W14 × 233	W14 × 500
W14 × 38	W14 × 109	W14 × 257	W14 × 550
W14 × 43	W14 × 120	W14 × 283	W14 × 605
W14 × 48	W14 × 132	W14 × 311	W14 × 665
W14 × 53	W14 × 145	W14 × 342	W14 × 730
W14 × 61	W14 × 159	W14 × 370	W14 × 808
W14 × 68			

Table 6: Least weight solution for fully rigid frame shown in Figure 4

W-shape	Members
W14 × 120	1,2,11,12,21,22 31,32,41,42,51,52
W14 × 53	3,4,5,13,14,15,23,24,25 33,34,35,43,44,45,53,54,55
W18 × 35	8,9,10,18,19,20,28,29 30,38,39,40,48,49,50
W12 × 14	6,16,36,46
W14 × 22	7,17,26,37,47
W16 × 26	27

Table 7: p_j for optimal solution shown in Figure 6, when base is fixed and $r_{req} = 10$

W-shape	Members
W14 × 193	1,2,11,12,21,22 31,32,41,42,51,52
W14 × 99	3,4,13,14,24 34,43,44,53,54
W14 × 120	23,33
W14 × 90	5,15,25,35,45,55
W12 × 22	6,16,36,46
W21 × 44	7,9,27,29,47,49
W21 × 55	8,18,28,38,48
W18 × 35	10,20,30,40,50
W18 × 40	17,37
W24 × 68	19,39
W16 × 31	26

Table 8: p_j for optimal solution shown in Figure 7, when base is hinged and $r_{req} = 6$

W-shape	Members
W14 × 211	1,11,21,31,41,51
W14 × 145	2,12,22,32,42,52
W14 × 90	3,13,23,33,43,53
W14 × 53	4,14,24,34,44,54 5,15,25,35,45,55
W12 × 14	6,16,36,46
W16 × 26	7,17,37,47
W21 × 48	8,18,28,38,48 10,20,40,50
W18 × 40	9,49
W21 × 44	19,39
W14 × 22	26
W33 × 118	29
W40 × 199	27,30

Table 9: Optimal cost for different values of C_r/C_s, when the base is fixed for 5-bay 5-story frame

Cost of rigid connection (C_r)	Cost of steel per ton (C_s)	Cost Ratio (CR)	Least-weight design cost ($)	Cost-optimal solution		% reduction in total cost
				r_{req}	Cost ($)	
750.00	750.00	1.00	54,147.75	12	33,761.81	38%
825.00	660.00	1.25	55,900.00	12	31,690.40	43%
900.00	600.00	1.50	58,318,65	10	30,425.25	48%
945.00	540.00	1.75	59,236.40	10	28,732.73	51%
1,000.00	500.00	2.00	62,098.50	10	27,854.38	55%

Table 10: p_j for optimal solution shown in Figure 10, when base is fixed and $r_{req} = 12$

W-shape	Members
W14 × 193	1,2,11,12,21,22 31,32,41,42,51,52
W14 × 82	3,4,5,13,14,15,23,24,25 33,34,35,43,44,45,53,54,55
W21 × 44	8,18,19,28 29,38,39,48
W12 × 22	6,16,36,46
W18 × 35	7,17,37,47
W24 × 55	9,39
W16 × 31	26
W18 × 40	27
W16 × 40	10,20,30,40,50

Table 11: p_j for $MGA1$ solution shown in Figure 11, when base is fixed and $r_{req} = 12$

W-shape	Members
W14 × 193	1,2,11,12,21,22 31,32,41,42,51,52
W14 × 120	3,4,13,14,23,24 33,34,43,44,53,54
W14 × 99	4,15,25,35,45,55
W14 × 22	6,16,36,46
W21 × 48	7,9,17,29,37,47,49
W21 × 55	8,18,38,48
W18 × 35	10,20,26,30,40,50
W24 × 68	19,39
W24 × 62	27
W24 × 55	28

Table 12: p_j for $MGA2$ solution shown in Figure 12, when base is fixed and $r_{req} = 12$

W-shape	Members
W14×193	1,2,11,12,21,22 31,32,41,42,51,52
W14×82	3,4,5,13,14,15,24,25 34,35,43,44,45,53,54,55
W14×99	23,33
W12×22	6,16,36,46
W18×35	7,17,37,47
W21×44	8,18,28,38,48
W21×48	9,29,49
W18×40	10,20,27,30,40,50
W24×62	19,39
W16×32	26

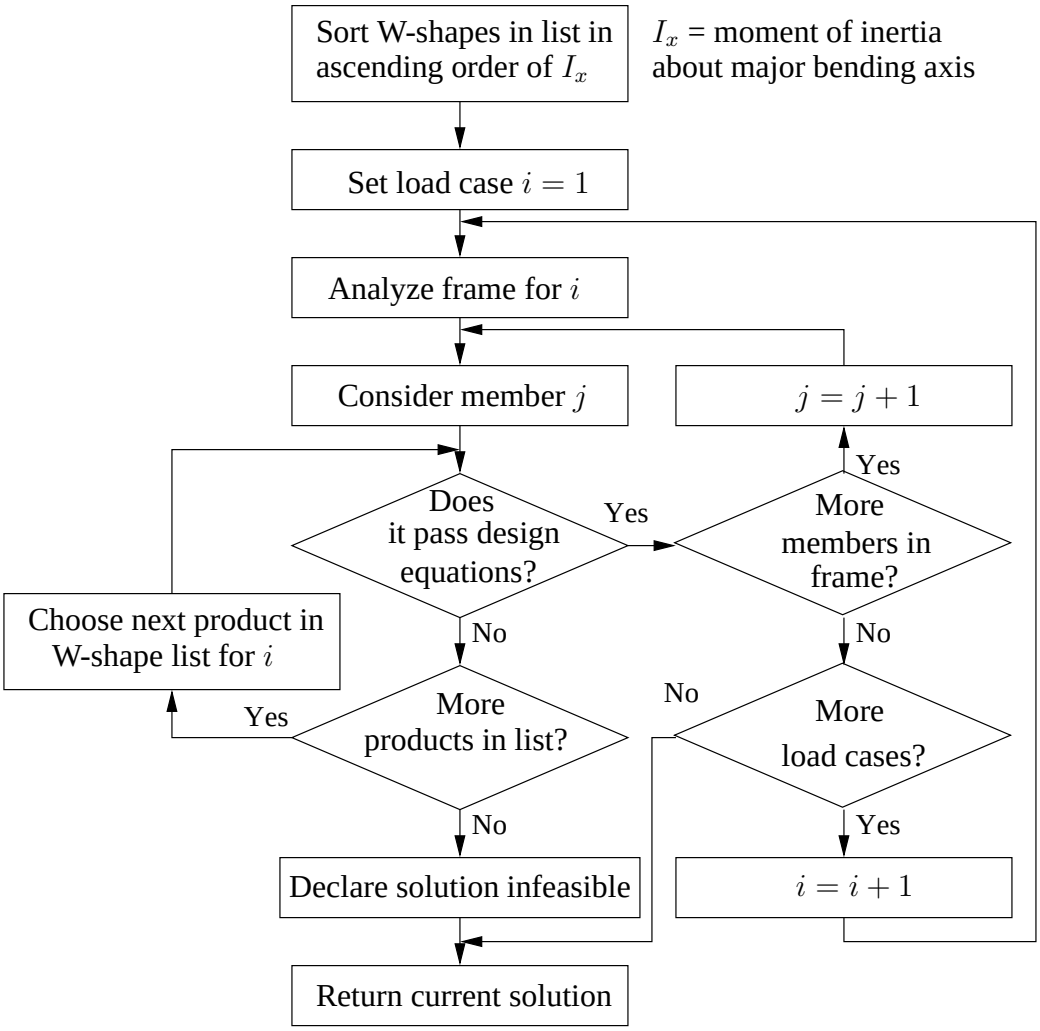


Figure 1: Algorithm for correcting solutions that violate equation 4 and 5

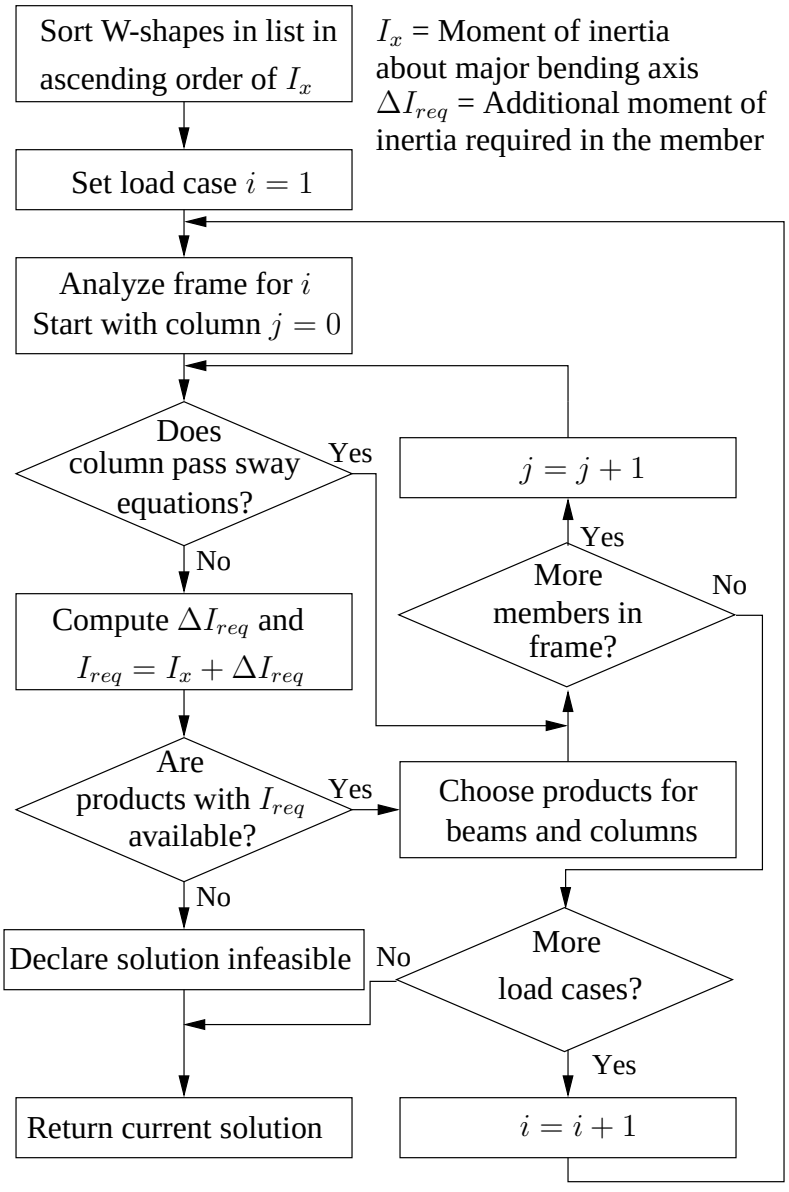


Figure 2: Algorithm for correcting solutions that violate sway constraints

1	2	3	4	5	6	7	8	
1	1	0	1	0	0	1	0	Parent 1
1	0	0	1	0	1	0	1	Parent 2
1	1	0	1	0	0	0	1	Offspring 1
1	0	0	1	0	1	1	0	Offspring 2

$S = \{2, 6, 7, 8\}; S_1 = \{2, 8\}; S_2 = \{6, 7\}$

Figure 3: Illustration of the crossover operator

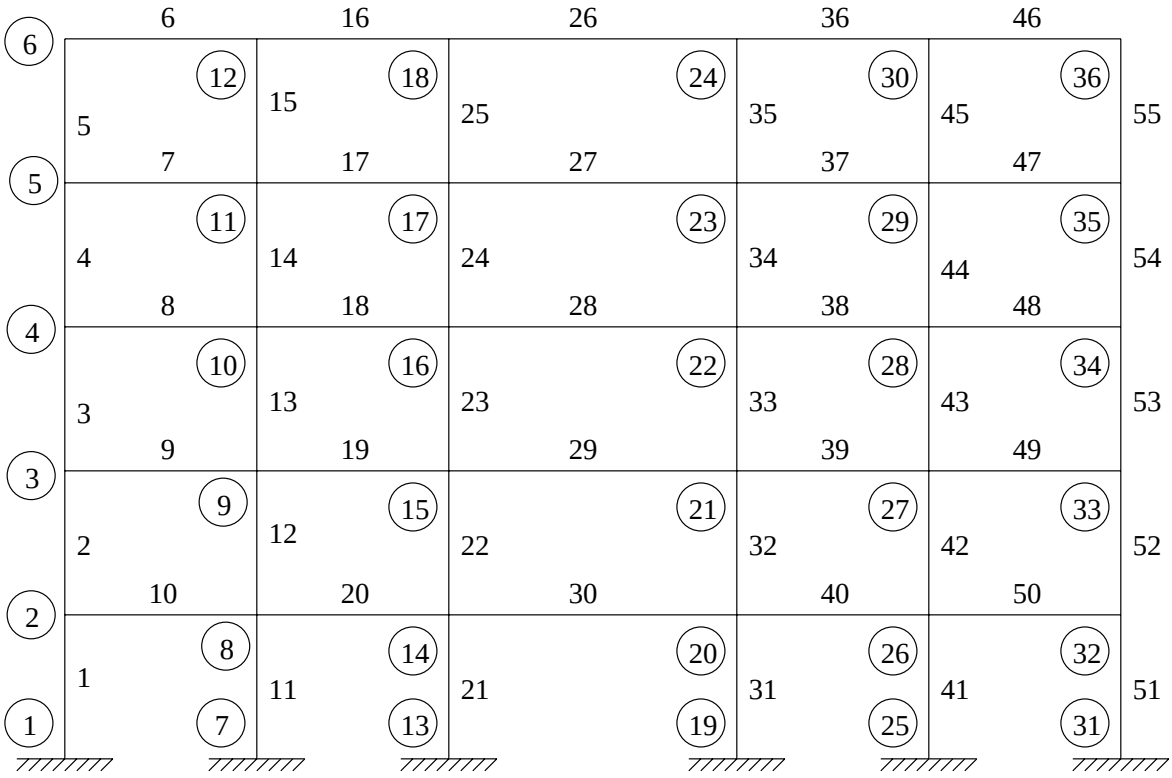


Figure 4: Elevation of 5-bay 5-story frame

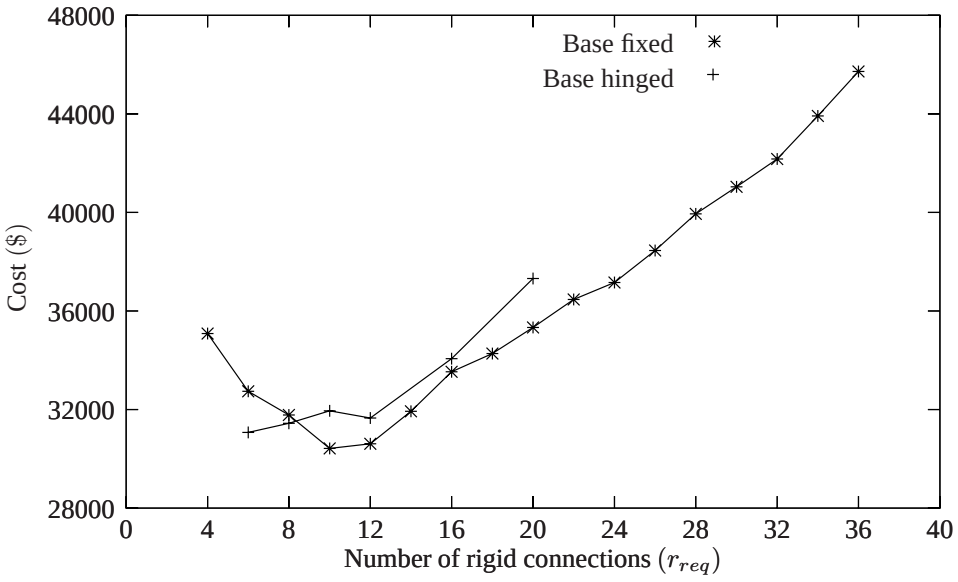


Figure 5: Trade-off curve between r_{req} and C for frame shown in Figure 4

Cost = $\$30,425$

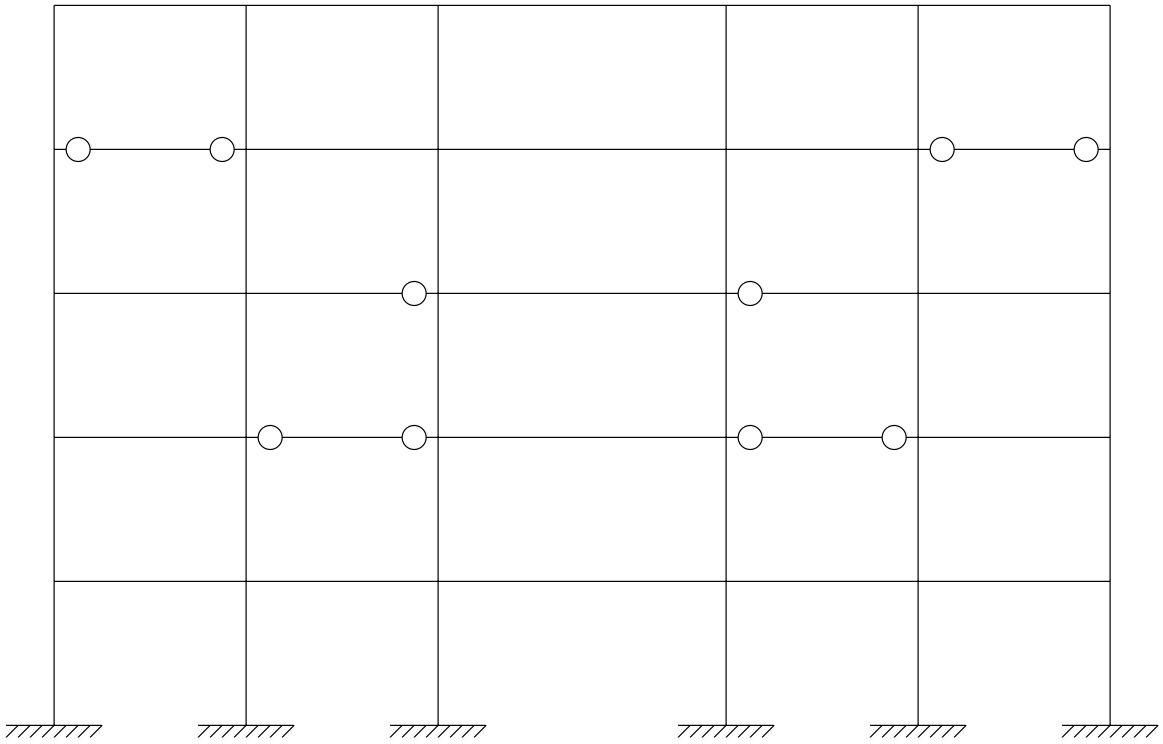


Figure 6: Rigid connection locations in cost optimal solution for 5-bay 5-story frame with fixed base

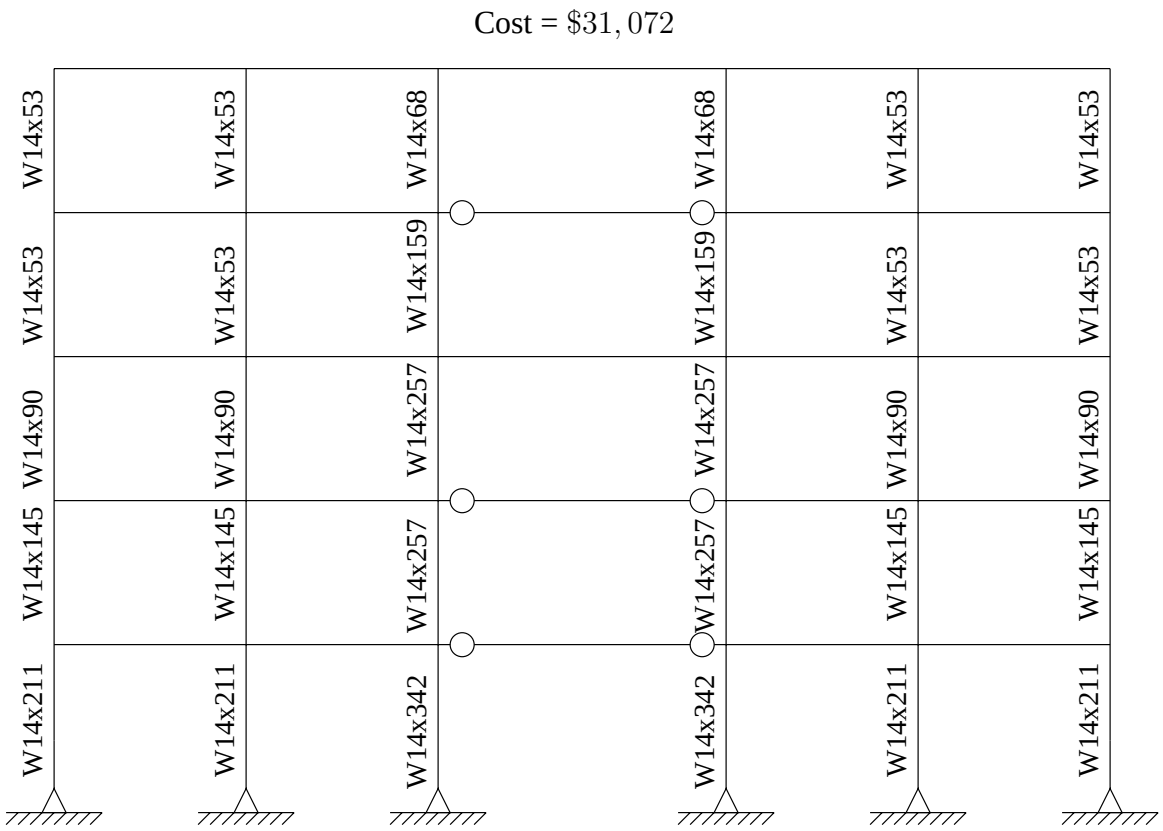


Figure 7: Column member types and rigid connection locations in cost optimal solution for 5-bay 5-story frame with hinged base

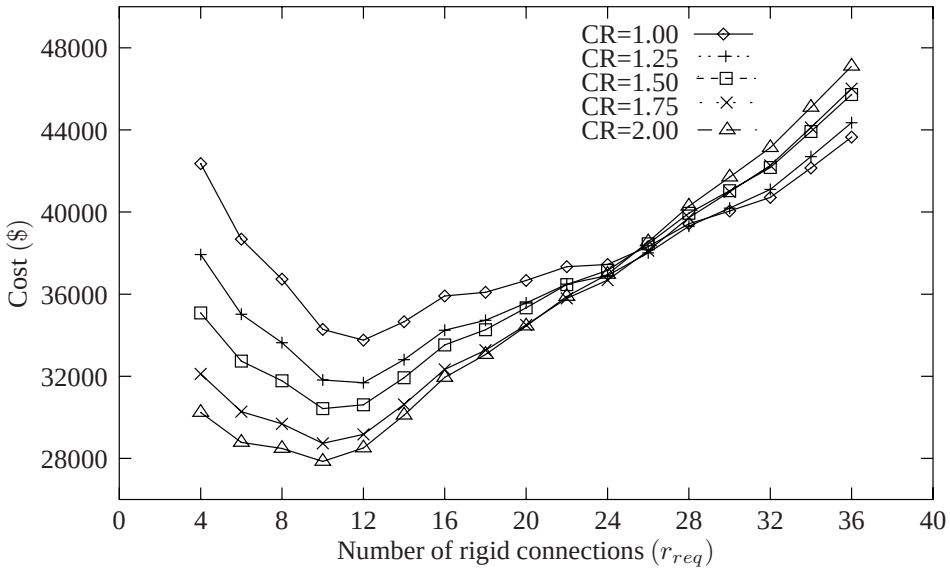


Figure 8: Plot showing trade-off curve for different values of CR=C_r/C_s

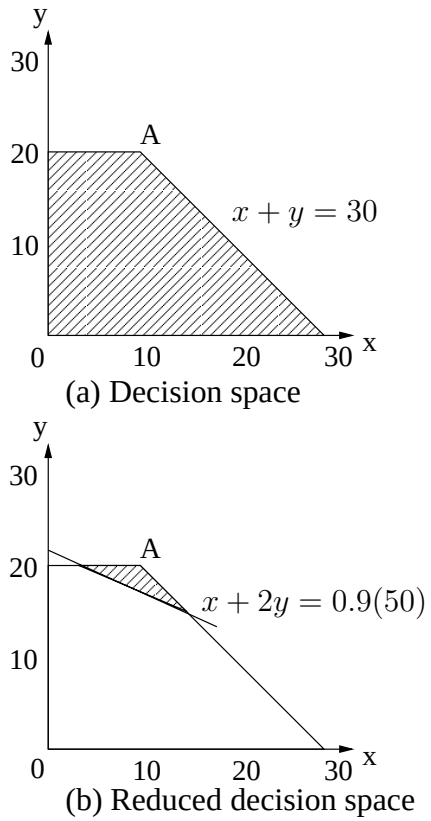


Figure 9: Illustration of MGA

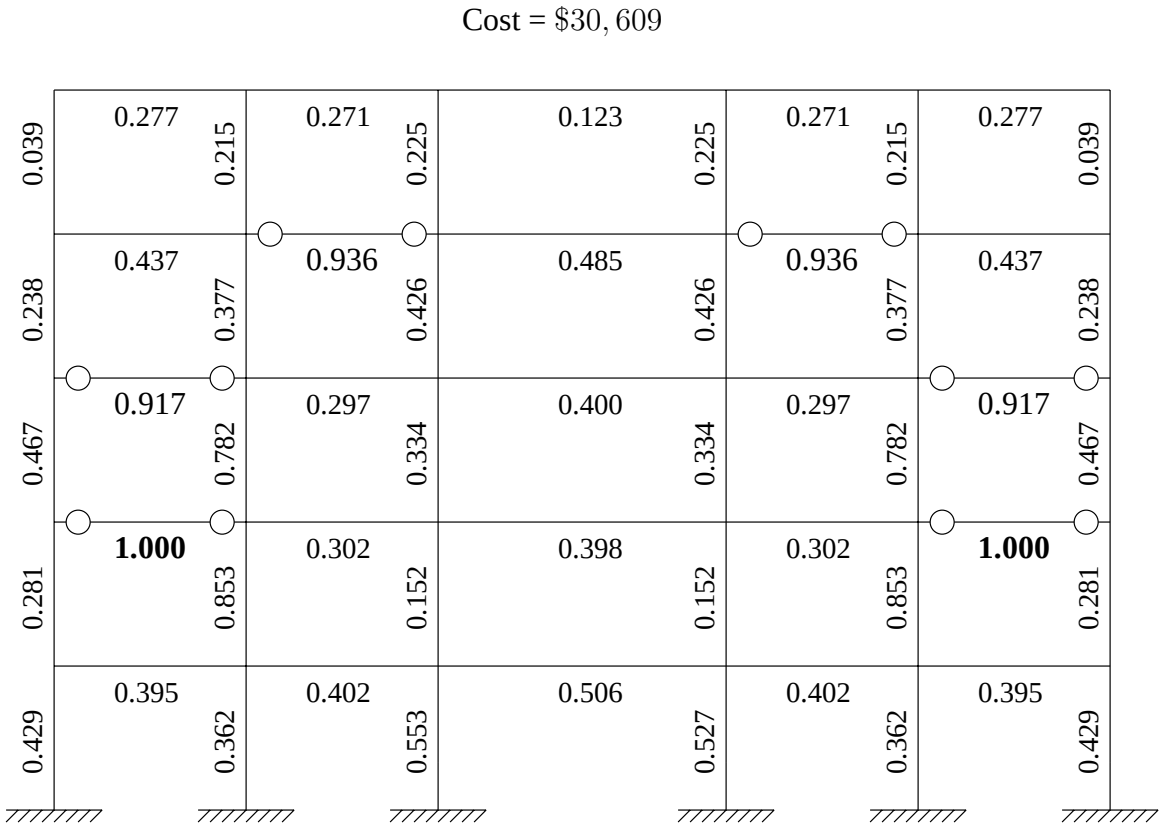


Figure 10: Design equation values for increased wind loads and rigid connection locations in optimal cost solution when all supports are fixed and $r_{req} = 12$

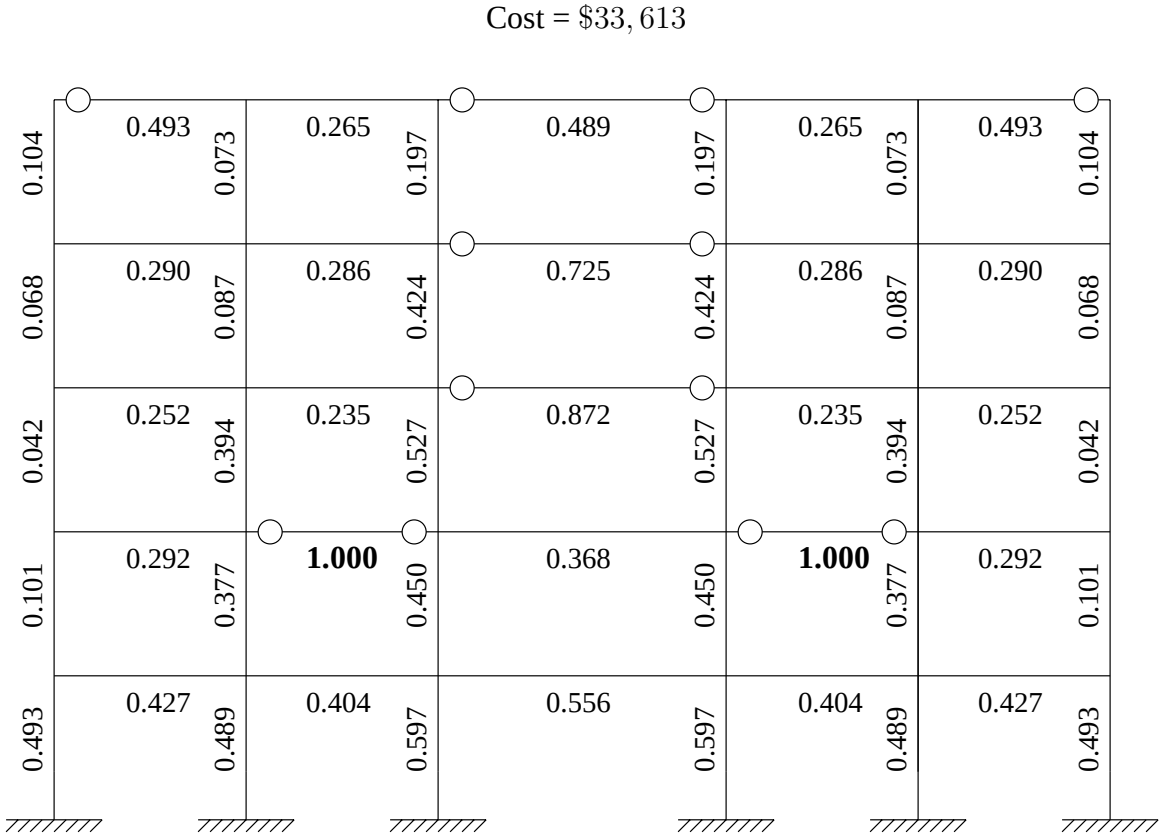


Figure 11: Design equation values for increased wind loads and rigid connection locations in $MGA1$ when all supports are fixed and $r_{req} = 12$

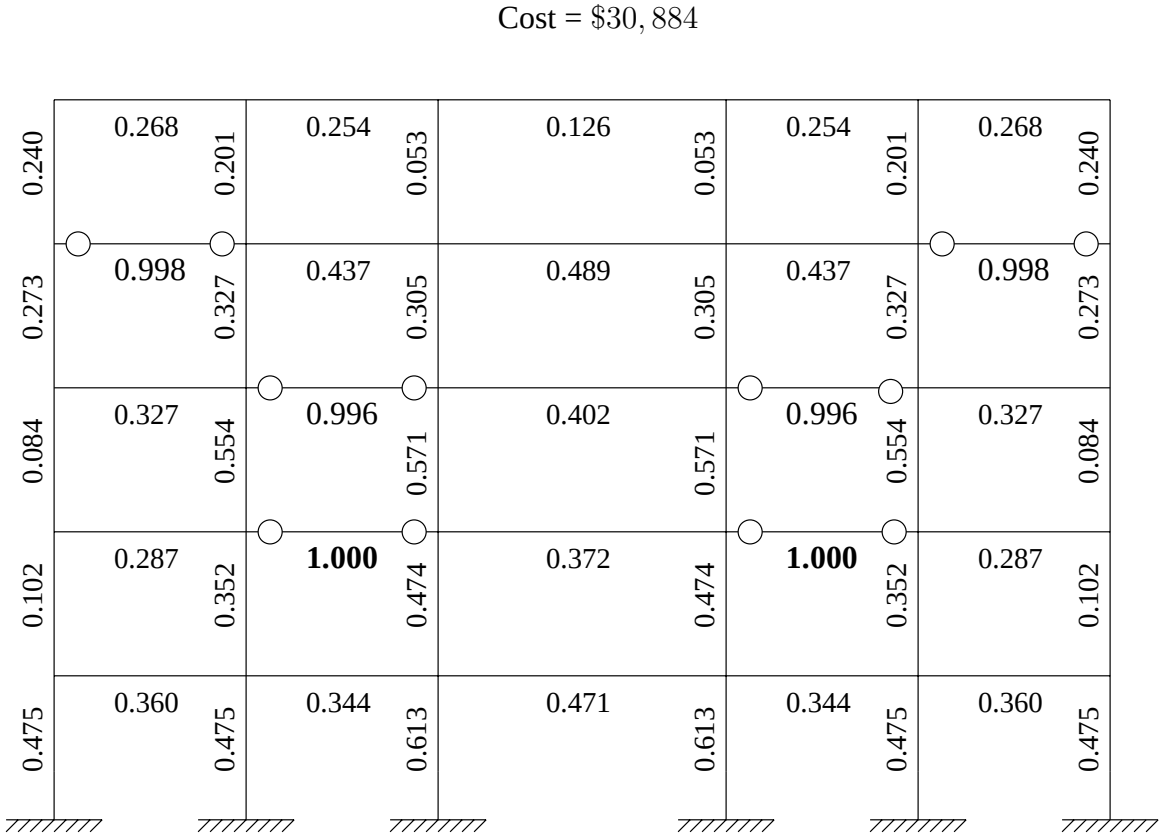


Figure 12: Design equation values for increased wind loads and rigid connection locations in $MGA2$ when all supports are fixed and $r_{req} = 12$

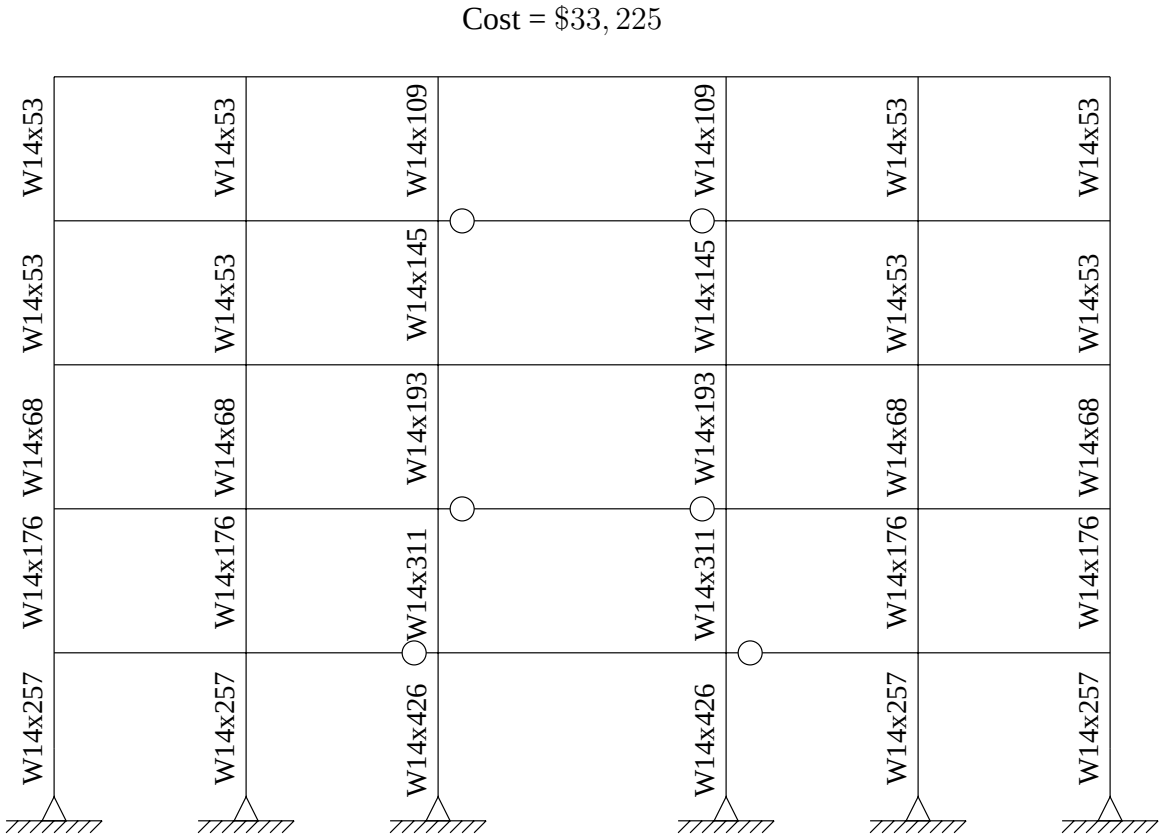


Figure 13: Column member types and rigid connection locations in $MGA1$ when all supports are hinged and $r_{req} = 6$

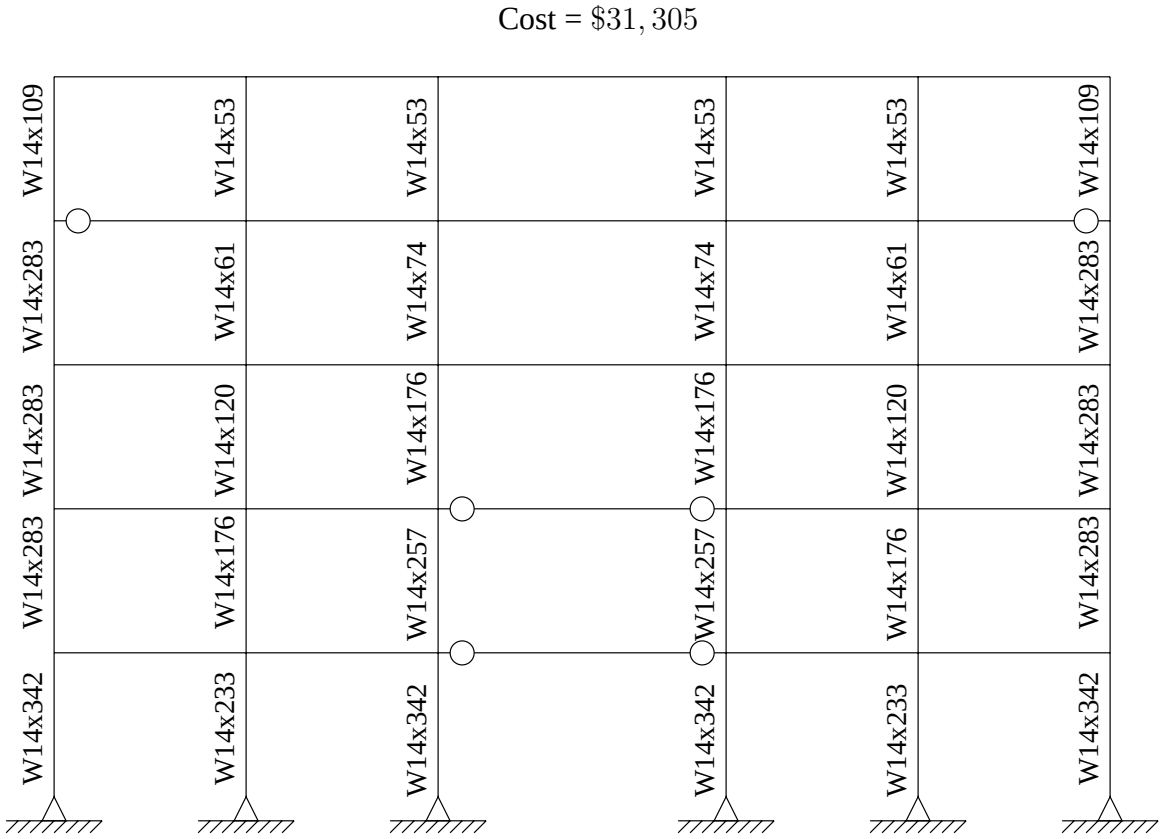


Figure 14: Column member types and rigid connection locations in $MGA2$ when all supports are hinged and $r_{req} = 6$

Appendix

Calculation of $\phi \mathbf{P_n}$

For tensile strength of member,

$$\phi P_n = \text{minimum of } \begin{cases} 0.9 F_y A_g \\ 0.75 F_u A_e \end{cases} \tag{1a}$$

For compressive strength of member,

$$\phi P_n = 0.85 A_g F_{cr} \tag{2a}$$

F_y and F_u are the yield and tensile stress respectively. A_g and A_e are the gross area and the effective net area of the member. F_{cr} is calculated as,

$$F_{cr} = \begin{cases} (0.658^{\lambda_c^2}) F_y, \text{ for } \lambda_c \leq 1.5 \\ \dfrac{0.877}{\lambda_c^2} F_y, \text{ for } \lambda_c > 1.5 \end{cases} \tag{3a}$$

$$\lambda_c = \frac{kl}{r\pi} \sqrt{\frac{F_y}{E}} \tag{4a}$$

k is the effective length factor. l is the laterally unbraced length of member. E is the modulus of elasticity of the material. r is the radius of gyration about the axis of buckling (r_y).

Calculation of $\mathbf{M_n}$

M_n is dependent on three sets of the following parameters - λ, λ_p and λ_r. Each set corresponds to one of the three major buckling criteria. For local flange buckling (LFB),

$$\lambda = \frac{b_f}{2t_f} \tag{5a}$$

$$\lambda_p = 0.38 \sqrt{\frac{E}{F_y}} \tag{6a}$$

$$\lambda_r = 0.83 \sqrt{\frac{E}{F_y}} \tag{7a}$$

For local web buckling (LWB),

$$\lambda = \frac{h}{t_w} \tag{8a}$$

$$\lambda_p = \begin{cases} 3.76\sqrt{\frac{E}{F_y}}(1 - \frac{2.75P_u}{\phi P_y}), \text{ if } \frac{P_u}{\phi P_y} \leq 0.125 \\ \max(1.12\sqrt{\frac{E}{F_y}}(2.33 - \frac{P_u}{\phi P_y}), 1.49\sqrt{\frac{E}{F_y}}), \text{ if } \frac{P_u}{\phi P_y} > 0.125 \end{cases} \tag{9a}$$

$$\lambda_r = 5.70\sqrt{\frac{E}{F_y}}(1 - \frac{0.74P_u}{\phi P_y}) \tag{10a}$$

$$\phi P_y = 0.9F_yA_g \tag{11a}$$

For lateral torsional buckling (LTB),

$$\lambda = \frac{L_b}{r_y} \tag{12a}$$

$$\lambda_r = \frac{L_r}{r_y} \tag{13a}$$

$$\lambda_p = \frac{L_p}{r_y} \tag{14a}$$

$$L_r = \frac{r_yX_1}{F_L}\sqrt{1 + \sqrt{1 + X_2(F_L^2)}} \tag{15a}$$

$$L_p = 1.76r_y\sqrt{\frac{E}{F_y}} \tag{16a}$$

b_f and t_f represent the breadth and the thickness of the flange respectively. F_y is the yield stress. E is the modulus of elasticity. t_w is the thickness of the web. P_u is the factored axial load (compressive). r_y is the radius of gyration about minor axis. $F_L = F_y - F_r = (F_y - 10)$ksi for hot-rolled shapes. L_b is the unbraced length of the member. X_1 and X_2 are specified for the different product types in the AISC Manual of Steel Construction.

M_n is calculated as,

$$M_n = \min \begin{cases} M_n(LFB) \\ M_n(LWB) \\ M_n(LTB) \end{cases} \tag{17a}$$

$M_n(LFB)$, $M_n(LWB)$ and $M_n(LTB)$ represent the nominal flexural strength determined by local flange buckling, local web buckling and lateral torsional buckling, respectively. These are determined using the following equations.

Case 1:

If $\lambda < \lambda_p$ for all buckling criteria, then the beam is identified as a compact beam and

$M_n(LFB) = M_n(LWB) = M_n(LTB) = M_p = F_y Z_x$. F_y is the yield stress and Z_x is the section modulus of the product type.

Case 2:

If $\lambda_r < \lambda < \lambda_p$ for one buckling criteria but $\lambda < \lambda_r$ for all other criteria, then the beam is a non-compact beam. For this case, M_n are specified by the following equations.

$$M_n(LFB) = M_n(LWB) = \begin{cases} M_p = F_y Z_x \\ M_p - (M_p - M_r)\dfrac{(\lambda - \lambda_p)}{(\lambda_r - \lambda_p)} \end{cases} \tag{18a}$$

$$M_r = F_L S_x = (F_y - F_r) S_x \tag{19a}$$

$$M_n(LTB) = \begin{cases} M_p = F_y Z_x \\ C_b(M_p - (M_p - M_r)\dfrac{(\lambda - \lambda_p)}{(\lambda_r - \lambda_p)}) \end{cases} \tag{20a}$$

$$M_r = F_y S_x \tag{21a}$$

F_r is the compressive residual stress in the flange and is equal to 10 ksi for rolled shapes. S_x and Z_x are the elastic and plastic section modulus about the major bending axis. C_b is a modification factor for non-uniform moment diagrams when both ends of the beam segment are braced. It is calculated as follows.

$$C_b = \frac{12.5 M_{max}}{2.5 M_{max} + 3M_a + 4M_b + 3M_c} \tag{22a}$$

M_{max} is the absolute value of the maximum moment in the unbraced segment. M_a is the absolute value of the moment at $0.25L$, where L is the length of the unbraced segment. Similarly, M_b and M_c are the absolute value of the moments at $0.5L$ and $0.75L$, respectively. In this study, we conservatively assume $C_b = 1.0$, which is the value specified for cantilever beams.

Case 3:

If $\lambda > \lambda_r$ for lateral torsional buckling and $\lambda < \lambda_r$ for local web buckling and local flange buckling, then M_n are calculated using the following equations.

$$M_n(LFB) = M_n(LWB) = \min \begin{cases} M_p = F_y Z_x \\ M_p - (M_p - M_r)\dfrac{(\lambda - \lambda_p)}{(\lambda_r - \lambda_p)} \end{cases} \tag{23a}$$

$$M_r = F_L S_x = (F_y - F_r) S_x \tag{24a}$$

$$M_n(LTB) = \min \begin{cases} M_p = F_y Z_x \\ C_b \dfrac{\pi}{L_b} \sqrt{EI_y GJ + \left\{ \dfrac{\pi E}{L_b} \right\}^2 I_y C_w} \end{cases} \tag{25a}$$

F_y is the yield stress. S_x is the section modulus of the member product type about its major bending axis. F_r is the compressive residual stress in the flange and is equal to 10 ksi for rolled shapes. G and J are the shear modulus and the torsional constant for the product type. L_b is the unbraced length and C_b is calculated as described above in Equation 22a. C_w is the warping constant for the member product. S_x and Z_x are the elastic and plastic section modulus for the member. I_y is the moment of inertia of the member product about its minor axis of bending.

Part IV

COMPUTATIONAL FRAMEWORK FOR REMOTELY OPERABLE LABORATORIES

Prakash Kripakaran, Abhinav Gupta and Vernon C. Matzen

To be submitted to the Journal of Computing in Civil Engineering, ASCE

Computational Framework for Remotely Operable Laboratories

Prakash Kripakaran, Abhinav Gupta and Vernon C. Matzen

Abstract

Decision-makers envision a significant role for remotely operable laboratories in advancing research in structural engineering, as seen from the tremendous support for the National Earthquake Engineering Simulation (NEES) framework. This paper proposes a generic computational framework that uses LabVIEW and web technologies to enable observation and control of laboratory experiments via the internet. The framework, which is illustrated for a shaketable experiment, consists of two key hardware components - (1) A local network that has an NI-PXI with hardware for measurement acquisition and shaketable control, and a Dell Precision that acquires images from a high-speed camera for video and (2) A proxy server that controls access to the local network. The software for shaketable control and data/video acquisition are developed in the form of Virtual Instruments (VI) using LabVIEW development system. The proxy server employs a user-based authentication protocol to provide security to the experiment. The user can run perl-based CGI scripts on the proxy server for scheduling to control or observe the experiment in a future timeslot as well as gain access to control or observe the experiment during that timeslot. The proxy server implements single-controller multiple-observer architecture so that many users can simultaneously observe and download measurements as a single controller decides the waveform input into the shaketable. A provision is also created for users to simultaneously view the real-time video of the experiment. Two different methods to communicate the video are studied. It is concluded that a JPEG compression of the images acquired from the camera offers the best performance over a wide range of networks. The framework is accessible by a remote user with a computer that has access to a high-speed internet connection and has the LabVIEW runtime engine that is available at no cost to the user. Care is taken to ensure that the implementation of the LabVIEW applications and the perl scripts have little dependency for ease of portability to other experiments.

1 Introduction

Recent developments in web technologies provide an excellent opportunity to form a network of laboratories that can enable control and observation of experiments via the internet. Experimental research laboratories that are geographically far apart can be networked together to create research collaboratories that promote engineering research and decision-making. Moreover, such a network of laboratories can also be useful for distance-education programs. As greater number of students are beginning to enroll in distance-education classes, educators are striving to provide a complete education experience in which theory is complemented with hands-on experiments [2, 1, 7, 3]. With these goals in mind, the NSF is currently sponsoring the Network for Earthquake Engineering Simulation (NEES). NEES *(http:// www.nees.org)* is a simulation resource that is composed of geographically distributed state-of-the-art experimental research equipment sites that are specifically designed to advance earthquake engineering research and education. Its objective is to make experimental laboratories with facilities like shaketables for simulating earthquakes, available for remote participation and control.

Even though internet is readily available as the underlying infrastructure for communication, significant modifications to the existing experiment configurations are required. Sophisticated hardware for data acquisition and control are needed to interface measurement and control devices with a computer. Appropriate software that graphically display acquired measurements and allow manipulation of control and measurement devices must be developed. Novel IT architectures that create a secure framework by integrating web technologies with the system for data acquisition and control are needed. Moreover, sufficient safety and security protocols are required to prevent intentional or unintentional damage to the experiment and its surroundings. The framework should also support the single controller-multiple observers concept. This capability will enable geographically-distributed engineers to simultaneously monitor and download measurements from the experiment as one engineer controls the loading on the test specimen.

Another facet of an internet-enabled experiment is to provide the remote user with real-time video of the experiment. Typically, structural engineering-related experiments that involve earthquake motion evaluate structural behavior at a frequency of 5-20 Hz. Capturing

this motion and communicating it in real-time requires a high frame-rate digital camera and appropriate hardware for acquiring images from the camera. Image compression techniques have to be explored to reduce the data transmitted to the remote user. In most cases, there is a trade-off between image compression and image quality. As with the measured data, multiple users may want to simultaneously watch the video of the experiment and download them on to their personal computer. Also, decision-makers often employ strobe-lights while doing vibration tests in laboratory for viewing mode shapes of the test structure. A similar capability in an internet-enabled experiment is highly desirable for educational purposes.

In this study, we propose a generic computational framework that enables the creation of such research collaboratories. We illustrate the framework by implementing it for the remote control and observation of a shaketable experiment. The framework has two key components - (1) A local network with computers that perform data/video acquisition and shaketable control using LabVIEW Virtual Instrument (VI) applications, and (2) A proxy server that controls access to the local network. The proxy server has forms that allow users to schedule for observing or controlling the experiment on a future date. A user can access the experiment as per the scheduled timeslot by using the forms provided on it. The proxy server uses the schedule and a user-based authentication protocol to control access to the LabVIEW host machines. The implementation of the forms on the proxy server and that of the applications for data acquisition and shaketable control have little dependency. The system ensures that only the user scheduled as controller can decide the waveform input into the shaketable. On the other hand, multiple users are permitted to schedule for simultaneously observing and downloading the measured data from the experiment. Provision is also created for multiple users to watch a live video feed of the experiment. The images for the video are captured using a high frame-rate camera. We have evaluated two different approaches to transmit real-time video through LabVIEW applications. One approach utilizes JPEG compression of the acquired images and the other employs converting the image into an array of numerical values that represent the intensities of the gray-scale image. The former, while providing faster video than the latter, requires the remote user to download additional software. The only requirements this framework places on the remote user's computer are the following two - (1) A high-speed internet connection like DSL or cable and (2) The LabVIEW run-time engine [6] which is available at no cost to the

user.

2 Shaketable Experiment

Seismic behavior of structural systems and various structural control architectures is studied by using a shaketable to simulate earthquake excitations. The shaketable has a platform to mount the structural system under consideration. Then, accelerations and displacements at different locations on the structural system are measured while the ground motion is applied through the shaketable. In this study, we consider a shaketable experiment that is primarily used as part of a laboratory course to illustrate the concepts in structural dynamics for undergraduate and graduate students. However, the illustrated computational framework is sufficiently generic and can be easily adapted to other laboratory experiments.

Figure 1 shows the laboratory setup of the shaketable experiment considered for illustration of the computational framework. It consists of a 12”$\times$34” one-dimensional shaketable and a 100 lb electro-magnetic shaker. The test specimen is a single or multi-story shear building having wide but thin aluminium columns and heavy steel girders. Forced vibration tests are conducted by applying a harmonic excitation to the table using a function generator. The input frequency of the excitation is increased in steps from a value that is lower than the natural frequency of the structure to one that is much higher. For each input frequency, the table is excited for a reasonable duration to ensure that the structure vibrates in steady state motion. The acceleration response is measured using accelerometers mounted on the different floors and can be viewed in an oscilloscope.

To transform this experiment to one that can be controlled and monitored through the internet, significant monetary investment on hardware as well as programming effort for appropriate software is needed. Remote monitoring of the response of the test specimen requires sophisticated data-acquisition hardware that can enable communication between the measurement devices like strain gauges and accelerometers, and a computer. Similarly, remote operation of a shaketable requires hardware that enables the computer to operate the electro-magnetic shaker.

Wirgau et al. [9] studied the viability of using National Instruments (NI) hardware and

software for data acquisition and control. They developed programs to communicate with the NI hardware in the form of Virtual Instruments (VI) using NI's LabVIEW development system [8]. In particular, two VIs were created for generation of waveform input and for display of measurements. For remote control, a LabVIEW technology referred to as Remote Panels [5] was employed. In this method, the host machine runs the LabVIEW webserver that serves the previously described two VIs to computers whose IP addresses are registered on the host machine. A remote user was required to register the IP address of his or her computer with the LabVIEW host machine by providing it to the administrator. Then only could a user control the experiment or observe the response through an internet browser from that computer. Real-time video of the experiment was provided by using Microsoft's netmeeting software with a web camera.

The implementation by Wirgau et al. [9] was restricted in generality due to the following reasons:

- The Remote Panels technology allows only one user to use the VI for viewing and downloading the response of the structure.
- The system is designed to accept connections only from computers whose IP addresses are on a list maintained by the LabVIEW webserver. Such a scheme is inflexible as the system is intended for use by authorized engineers via the internet irrespective of their IP address.
- A higher likelihood of intentional damage to the system by malignant users on the internet existed, since the LabVIEW host PC was directly connected to the internet.
- A scheduling facility did not exist. Such a facility would allow users to sign up for controlling or monitoring the experiment on a future date.
- A web camera does not sufficiently capture the high-frequency motions resulting from a shaketable experiment.
- Netmeeting software does not ensure a high video transmission rate that is required for remotely viewing the real-time video of a dynamics experiment.

In the following sections, we propose a computational framework that addresses these aspects. First, we describe the hardware and network setup of the framework. Later, we discuss the software architecture implemented over this hardware.

3 Hardware Setup of Framework

The computational framework is composed of a network of three computers - (1) A linux-based proxy server, (2) An NI-PXI for data acquisition/control and (3) A windows-based PC with hardware for video acquisition. The networking of the computers is illustrated in Figure 2. An IBM thinkpad with the fedora linux operating system is used as the proxy server. Figure 2 shows that the proxy server serves as the gateway to the experiment for a remote user on the internet. It has two network adapters one each for communication with the internet and the computers on the local network, respectively.

The PXI uses an NI 8176 controller and runs LabVIEW in a windows environment. The host NI-PXI does not directly perform the data acquisition or real-time control. Instead, it has a control board with a dedicated processor and memory. LabVIEW programs can be embedded into the board for real-time control as well as data acquisition. This facility protects the experiment from any software or hardware related accidental failure in the host PXI as the embedded program can safely shutdown the experiment during such an event. The accelerometers mounted on the test specimen are connected to this board through a BNC connector. Similarly, the board is also wired to the shaketable so that the generated waveforms may be communicated to the shaketable for real-time control. It must be noted that the same PXI was previously used in the study by Wirgau et al. [9].

Real-time video of the experiment is obtained using a Dell precision 670 that has an NI-IMAQ card (PCI-1429) mounted on its PCI express bus. The IMAQ card offers a high data transfer rate of 680 MB/s. A high-speed Basler camera (Model: A504K) that uses a progressive scan CMOS sensor is used to capture the video. The camera produces 8-bit monochrome images. The camera can capture images at a maximum speed of 500 frames per second and a resolution of 1280×1024 pixels. This frame rate is easily sufficient for shaketable experiments as the frequencies that are of interest often lie within 20 Hz. A zoom lens, which can be controlled using software on the computer, is attached to the basler camera. The camera, which is mounted on a tripod, is kept at a distance of 4 ft from the shaketable. Since the camera sensor requires a bright setting for producing good images, powerful lamps are employed for additional lighting.

4 Software Architecture

The software architecture essentially consists of two types of components - (1) LabVIEW applications for data acquisition, video and control, and (2) Web technologies for experiment scheduling and authentication. The interaction between these components is illustrated in Figure 3. While the LabVIEW applications are completely independent of the web technologies, the web technologies, on the other hand, use some information about the implementation of the LabVIEW applications. Consequentially, we first describe the LabVIEW applications, which is followed by a description of the web technologies.

4.1 Data Acquisition and Control

As described earlier in this paper, Wirgau et al. [9] implemented VIs for data acquisition and control and used them in combination with LabVIEW remote panels for remote observation of the experiment. The architecture of their VIs is schematically illustrated in Figure 4. For control of the shaketable, a VI that generates the waveform as per the input amplitude and frequency is run on the host PXI. Another VI that is embedded on the real-time board of the PXI receives the generated waveform and communicates it to the shaketable. These two VIs always maintain an active TCP/IP connection for communication. The VI on the host PXI is made available for the remote user using LabVIEW remote panels technology. The LabVIEW webserver is activated on the host PXI and it is configured to accept connections from a specific remote computer. The remote user can control this VI through a internet browser by connecting to the LabVIEW webserver on the host PXI. Our framework uses a similar implementation for granting control of the experiment to the remote user.

Wirgau et al. [9] also proposed a similar implementation for data acquisition. This is schematically shown in Figure 5. A VI was run on the real-time board of the PXI to read measurements from the accelerometers. These were communicated using TCP/IP to another VI on the host PXI. The VI on the host PXI displays the measured data using an oscilloscope. It also provides other interfaces to manipulate the display in the oscilloscope. For instance, the user can freeze the display to examine a specific frame, say, for calculating the dynamic magnification factor. The remote user gains access to this VI using LabVIEW remote front

panels. When implemented in this manner, only one remote user can control this VI and hence, observe the measurements. While it was suitable for controlling the shaketable, it is desirable to provide the capability for multiple observers to simultaneously monitor the experiment. Therefore, we have modified the implementation to use LabVIEW's datasocket technology. The new implementation enables multiple users to simultaneously visualize and download the acceleration measurements.

LabVIEW's datasocket technology is specifically aimed at distributing measurements to geographically distributed users. The computer that is used to make the measurements writes them to a datasocket server that maybe running on the same computer or on a different computer. All users irrespective of geographical location can obtain the measurements by using a LabVIEW application that subscribes to the datasocket server. Thus, our implementation of data acquisition using datasockets consists of two types of applications. One for the PXI that will gather data using the appropriate devices and publish it to a LabVIEW datasocket server. A second that will be used by each of the remote users for observing the measurements.

The proposed framework for data acquisition is given in Figure 6. The VI that is embedded on the real-time control board reads data from the accelerometers. It always maintains a TCP/IP connection with the VI on the host PC. The VI sets the control board to scan the data channels at a rate of 1000 scans/second. During each scan, the control board acquires the voltage values returned by the two accelerometers. The VI uses a loop structure such that it acquires data corresponding to 100 scans during every loop iteration. The acquired data is stored in a $100 \times m$ array, where the number of rows correspond to the number of scans and the number of columns m to the number of accelerometers, respectively. This array of data acquired in each loop iteration is communicated to the VI on the host PXI. The VI on the host PC uses the accelerometer calibration information to convert the acquired voltage values to meaningful acceleration values. It then writes the acceleration values to the datasocket server that is running on the dell precision PC. The datasocket server is configured to accept connections from computers with VIs that wish to read the acceleration values.

To view the accelerations, the remote user downloads a LabVIEW executable that is created from a VI. This executable connects to the datasocket server running on the PXI and reads the accelerations. The executable also possesses various graphical interfaces that enable the user

to examine the acceleration measurements. A snapshot of the interface of the executable is provided in Figure 7. In particular, there is a knob control that the user can adjust to set the number of data points and hence, the number of wavelengths to be shown in the oscilloscope. The user can visualize a fewer number of datapoints in the oscilloscope using this control, even though the datasocket server provides acceleration measurements in chunks of only 1000. The client executable makes this possible by breaking the data read from the datasocket server into smaller pieces as determined by the setting on the knob control. These pieces are sequentially processed by the oscilloscope. Additional interfaces like a button control to freeze a particular dataframe in the oscilloscope and a tool for moving the scales in the oscilloscope are also present. The VI also provides facilities to download the acquired data in a format that can be viewed in a spreadsheet application like Microsoft Excel. The application writes the acquired data to a specified spreadsheet file whenever the corresponding switch is turned on.

4.2 Video Acquisition

Programs for video acquisition are developed using NI's image acquisition (IMAQ) drivers. The camera produces images at a rate that can be set using IMAQ drivers. We provide real-time video by continuously streaming these images to the remote user. These images, which are obtained through the IMAQ hardware, are in a unique format that requires installation of IMAQ drivers for viewing. This can be a hindrance to the remote users as they may have to download a fairly large-sized set of drivers. To avoid this issue, the images can be converted to a different format that the remote user can view without installing additional software. Another important consideration while implementing real-time video, is related to the size of the images transmitted to the remote user. If the images are large, the remote user may experience significant delay between consecutive images due to the increased communication time. The images can be sent in a compressed form as a solution. Multiple alternatives are available with regard to handling these issues. These alternatives determine the factors like size and quality of the image that is communicated to the remote user. For example, when the images are converted to a JPEG form, the reduction in size of the image is often achieved at the expense of some information loss in the final JPEG image. These factors play a significant role in determining

the performance of the real-time video of the experiment.

In this study, we have evaluated two techniques to provide real-time video. These two methods only differ in the image processing algorithms used to modify the images before communicating them to the remote user. Otherwise, both the methods function in a similar manner. The working of this framework is illustrated in Figure 8. As in the case of measurement communication, both methods use LabVIEW's datasocket technology for communicating real-time video of the experiment. There are two VIs - (1) The VI on the Dell Precision continuously acquires images from the IMAQ interface, processes them and writes the resulting images to the datasocket server, and (2) The VI on the remote client connects to the datasocket server and displays the images that are received. The image processing techniques that differentiate the two methods are described below.

- *JPEG streaming:* In this method, the VI on the host PXI converts the acquired IMAQ images to a JPEG stream. The conversion algorithm accepts an integer parameter between 0 and 1000 that corresponds to the desired quality of the resulting image. The image quality increases with increase in the value of the parameter but so does the size of the resulting image. Since the camera produces an 8-bit monochrome image with a resolution of 1280 pixels, the size of the original image is approximately 1 MB. We have specified a value of 90 for the parameter in the conversion algorithm. The size of the image after compression is close to 42 kB, which is considerably smaller than the size of the original image. This image is written to the datasocket server. The main advantages of this method are: (a) faster communication because of the smaller image size and (b) absence of any noticeable delay between consecutive images at the remote computer. The main disadvantage of this implementation is that it requires the remote computer to have IMAQ drivers to display the received images even though the images were converted to a JPEG stream format at the PXI.
- *Intensity graph:* One approach that avoids having IMAQ drivers at the remote computer involves using LabVIEW's intensity graph interface. The acquired image is converted to a two-dimensional array of integer values that represent the color intensities of the corresponding pixel in the image. The VI on the host PXI writes these arrays to the datasocket server. The application on the remote computer continuously downloads these arrays from

the datasocket server. The application generates the images by plotting these arrays in an intensity graph. The color scale of the intensity graph is adjusted to gray-scale as the camera generates a monochrome image. The intensity graph is continuously refreshed with new arrays from the datasocket server. Thus, the remote user will see the real-time video of the experiment in the intensity graph. While the IMAQ drivers are not required for the remote user, this implementation can involve higher communication time between the remote client and the server than the previous method. The reason is that the size of the communicated arrays can be significantly large, i.e., of the order of 1 MB. One way of reducing the size of the communicated data is to shrink the acquisition window for the camera. For example, if the experiment can be captured in a window of 850×800 pixels, then the size of the communicated data is only 425 kB. This is considerably smaller the data generated from a full acquisition window.

Table 1 summarizes the differences between the two techniques. The table gives estimates of the amount of data required to communicate a single image to the remote user when using the two techniques. It is observed that the method using intensity graph sends more data and hence requires significantly large network bandwidth. Moreover, the JPEG compression does not significantly reduce the quality of the image. Therefore, the JPEG compression technique is a better overall solution and will deliver better performance over a wide variety of networks. The intensity graph method is a better alternative only in the presence of very high bandwidth such as in the case when the remote user and the experimental site are located in the same LAN.

5 Web Architecture

The LabVIEW applications described earlier in the paper use one of the following two technologies for communication with the remote user - (1) Datasockets and (2) Remote panels. Datasockets are used for sensor and video data communication while the remote panels are used for remote shaketable control. The datasocket server and the LabVIEW webserver, which are the programs that support the two technologies, operate by listening on certain network ports. The proxy server controls access to these network ports using perl-based CGI

scripts within a webserver and thereby provides security to the experiment. The key actions performed by the proxy server from the moment a user logs into the system to observe or control the experiment until the time when a user disconnects from the system is illustrated using a flowchart in Figure 9. As seen from the flowchart, the following sequence of steps are involved:

1. The remote user is authenticated by the proxy server using WRAP [4], a web-based authentication mechanism.
2. The user schedules a particular timeslot for controlling or observing the experiment on a future date.
3. On the scheduled date, the user connects to the proxy server and is authenticated.
4. The user requests access to control or monitor the experiment by running a CGI script on the webserver.
5. The CGI script checks if the user is scheduled to monitor or control the experiment during the current timeslot.
6. If the user is scheduled, the script then sets up port forwarding on the proxy server so that the user can access the appropriate server on the LabVIEW host.
7. The user runs a LabVIEW application that is downloadable from the website to observe the experiment. To control the experiment, the user uses a browser as explained previously in the paper.
8. Port forwarding that was setup in the previous step is disabled when the current timeslot expires.

These steps are performed using the following components, which constitute the web architecture.

5.1 User Authentication

North Carolina State University uses a cookie-based authentication protocol referred to as WRAP [4]. This authentication protocol is used to verify the authenticity of users with access to the university's computing facilities. Students and university employees have a username and password that can be used in various computing labs around campus. Since one of the goals of this study was to make the shaketable experiment available for distance-education

students, we have used the university's authentication protocol for providing internet security to the experiment.

WRAP is a web-based authentication mechanism to verify the identity of a user without requiring the user to login to each individual webserver within the university domain. In this authentication mechanism, the user obtains an encrypted cookie, called the WRAP cookie, from a SSL-secured server by using his/her username and password. Whenever the user visits a WRAP-protected website, the browser sends the WRAP cookie to the website. The website verifies if the cookie is genuine and also obtains the username for that user. If the user does not already possess a WRAP cookie or possesses an invalid cookie, the user will be forwarded to the SSL-secured server that will issue a WRAP cookie to the user. WRAP cookie components like the username are available as environment variables within CGI scripts. The CGI scripts can, therefore, recognize the user making the request.

The apache webserver on the proxy server is configured to use WRAP. File directories that contain the CGI scripts are protected using WRAP. Thus, the remote user is forced to obtain a WRAP cookie before attempting to run the scripts. If an authorized user is making a request, the CGI scripts use the environment variables to recognize the user. This information is later used in scheduling the user for the experiment as well as setting up remote access to the experiment for the user.

5.2 Experiment Scheduling

A user can schedule to either control or observe the experiment in a timeslot within the next ten days. The system provides 8 timeslots for the experiment each day. Each timeslot is of $2\frac{1}{2}$ hours duration. There is a 30 minute interval between the timeslots for the laboratory personnel to make adjustments, if necessary. The system maintains two timetables that keep track of the users currently registered to use the experiment. One timetable is for users registered to observe the experiment while the other is for users registered to conduct the experiment. The latter timetable has only a single user for a given timeslot whereas the former timetable can have multiple users registered in any given timeslot. This is consistent with the objective of allowing multiple observers but only a single controller.

When a user wants to schedule for controlling the experiment, the user can navigate to the two-week timetable from the homepage of the proxy server website. Figure 10 shows the table appearing in an internet browser. The user can choose a timeslot by filling in the two text boxes on the form and hitting the submit button. If the timeslot is currently unused, the user will be registered into the timeslot and the updated table will be listed. If the timeslot is already taken, the user will be displayed a corresponding message. To schedule as an observer to the experiment, the procedure is similar with the only difference being that the forms do not check for availability of a selected timeslot because multiple observers are permitted.

5.3 Single-User Control

Users can control the experiment within their scheduled timeslot by visiting the website. The activities performed by the proxy server to setup access for the remote user are given in a flowchart in Figure 11. As shown in the figure, once the user is authenticated using WRAP, the proxy server checks whether the user is already scheduled to control the experiment. If scheduled, the script permits the remote user to access the LabVIEW webserver on the host PXI. The remote user uses a browser to access a http link returned by the script. The user must use internet explorer as the browser as the LabVIEW runtime engine is only configured to work with it.

Access to the LabVIEW webserver is setup using the networking concept of packet forwarding. The LabVIEW webserver on the host PXI is setup to listen on TCP/IP port 8080. All TCP/IP packets arriving from the remote user's computer on port 8080 of the proxy server are forwarded to port 8080 on the host PXI. There are two software components involved in setting up packet forwarding. One is the CGI script that is run on the webserver when the user visits the website. The other is a perl script that is always running on the proxy server. Figure 12 shows the interaction between the two scripts in a sequence diagram. The CGI script identifies the IP address of the remote user's computer. It then verifies if remote access is already enabled for the user. For this purpose, the script uses a data file that keeps track of the various users currently using the system. If the user is already logged in, the system simply informs the user that the connections are already active and that the user needs to visit the returned http link. If

the user is not logged into the system, it communicates the IP address and the duration left in the timeslot to the perl script. The perl script uses the linux networking command - 'iptables,' to make the proxy server appropriately forward packets arriving on port 8080 from the remote user's computer. The perl script also ensures that the packet forwarding is disabled after the expiry of the allotted time for the user.

The reason for using a perl script different from the CGI script to setup packet forwarding lies in concerns regarding network security. The 'iptables' command is a very low-level system command and it is essential to prevent direct access to it from a user on the internet. Also, using a single script that is always active on the proxy server provides a way of queuing multiple requests. Even if the user runs the CGI script multiple times, the requests are processed sequentially as all the CGI requests are finally processed by a single perl script.

5.4 Multi-User Observation

The implementations for access to monitor the measurements and to view the real-time video are similar in many ways to the implementation discussed above for an access to control the experiment. As discussed earlier in the paper, the remote user will view the video and acceleration measurements using two different LabVIEW applications. These applications will attempt to connect to the datasocket server on the Dell Precision computer in the local network. Therefore, the scripts in this case enable access to the TCP/IP port on which the datasocket server listens.

Figure 13 shows a sequence diagram representing the interaction between the CGI script activated by the remote user and the perl script that is always active on the proxy server. The key difference between Figures 13 and 11 lies in the TCP/IP port to which packet forwarding is enabled. Since the datasocket server listens on TCP/IP port 3055, the scripts in this case setup packet forwarding to that port. It is noted that this mechanism enables access to both video and acceleration measurements as both are delivered via the datasocket server.

5.5 Experiment Administration

The software architecture, while automating the operation of the laboratory, should also provide sufficient administrative tools to the laboratory administrator. In the event that the experiment needs to be stopped, the laboratory administrator must be able to gain control of the experiment. Since the remote user who is controlling the experiment, accesses the corresponding VI through remote front panels, the laboratory administrator can easily gain control of the VIs by directly accessing them through the host PXI. Also, the CGI scripts record the username and the IP address of the remote users currently using the system in a data file. The administrator can use the information in the file to disconnect any user from the system by disabling packet forwarding between the remote user's computer and the appropriate computer on the local network. The administrator can modify the weekly schedule as well as the daily schedule to make any changes.

6 Summary and Conclusions

Recent computing advances in web technology have given rise to new ideas in web-based experimentation for supporting engineering research and education. The National Science Foundation (NSF) is sponsoring the development of a national Network for Earthquake Engineering Simulation (NEES) that will make large-scale laboratories with seismic testing facilities available for control and observation to geographically distributed researchers. Numerous hurdles exist in the various tasks from creating applications that will enable a remote user to perform data acquisition and control of the experiment to developing novel security and safety protocols for preventing intentional or unintentional damage to the laboratory experiment. In this study, we have proposed a generic computational framework as a solution. The framework uses a combination of National Instruments hardware and software, and web technologies to create a comprehensive internet-based environment that implements a single-user multiple-observer model for remote control and observation of laboratory experiments. The proposed framework is illustrated for a shaketable experiment. LabVIEW applications that use datasocket technology and remote front panels are developed for data acquisition and control respectively. Video of the experiment is acquired using high-speed cameras and NI's IMAQ

software. Video transmission using two techniques - JPEG compression and intensity graph, are explored. The study concludes that the JPEG compression method is relatively more reliable. The computers that operate the shaketable experiment are in a local network, which is protected using a proxy server. The webserver uses NC State University's WRAP authentication protocol to provide web security. Remote users are able to access the laboratory schedules on the website, which is hosted by the proxy server, and schedule for controlling or observing the experiment on a future date. Within a scheduled timeslot, the appropriate remote user is given permission to control or monitor the experiment after the user is authenticated by the website. The main conclusions from this study are:

- LabVIEW Remote Panels technology is an efficient solution if only a single user needs to remotely access the laboratory experiment. Such an implementation is suitable for the controller in a single-controller multiple-observer framework.
- LabVIEW Datasockets technology enables simultaneously communicating data to users distributed across the internet. This technology is ideal for communicating sensor and video data to remote observers in a single-controller multiple-observer framework.
- The JPEG compression technique is relatively better than the intensity graph method for video transmission. It provides a suitable compromise between the size of the communicated data and the image quality received at the remote user.
- A webserver with suitable user-authentication protocol, CGI scripts and linux networking concepts are sufficient to create a secure web-based environment to control access to laboratory experiments.
- The web technologies and LabVIEW VIs are decoupled to a large extent. The dependencies between the two components are primarily with respect to the TCP/IP listening ports of the LabVIEW webserver and the datasocket server.
- The computational framework is fairly generic and easily extensible to other laboratory experiments. The major task in adapting the framework to a new experiment is the development of new LabVIEW VIs that are necessary for control and data acquisition with respect to the new experiment.

References

[1] C. L. Bohus, A. Crowl, B. Aktan, and M. H. Shor. Running control engineering experiments over the internet. In *Proceedings of the 13th IFAC World Congress*, San Francisco, CA, 1996. paper no. 4c-03.

[2] M. L. Corradini, G. Ippoliti, T. Leo, and S. Longhi. An internet based laboratory for control education. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, FL, December 2001.

[3] A. Gupta, M. A. Gabr, and V. C. Matzen. Alternatives in the implementation of internet-enabled laboratory. In *2004 ASEE Annual Conference and Exposition*, Salt Lake City, June 2004.

[4] NCSU. WRAP, 2005. https://webauth.ncsu.edu/wrap/.

[5] NI. NI LabVIEW Remote Panels, 2005. http://sine.ni.com/nips/cds/view/p/lang/en/nid/11-017.

[6] NI. NI LabVIEW Runtime Engine, 2005. http://digital.ni.com/softlib.nsf/websearch/.

[7] S. E. Poindexter and B. S. Heck. Using the web in your courses: What can you do? what should you do? *IEEE Control System*, 9(1):83–92, 1999.

[8] L. Wells and J. Travis. *LabVIEW for Everyone*. Prentice Hall, 1997.

[9] S. Wirgau, A. Gupta, and V. Matzen. Internet-enabled remote observation and control of a shake table experiment. *Journal of Computing in Civil Engineering, ASCE*, 2005. In press.

Table 1: Comparison of two video transmission techniques

Features	JPEG Compression	Intensity Array
IMAQ drivers	Required at both host and remote computer	Required only at host
Data size	42 kB	450 kB
Effect of reducing acquisition window	Reduces data size	Reduces data size
Computation	Compression is computation intensive	Computationally inexpensive
Suggested use	Over any network	Suitable only for highbandwidth networks

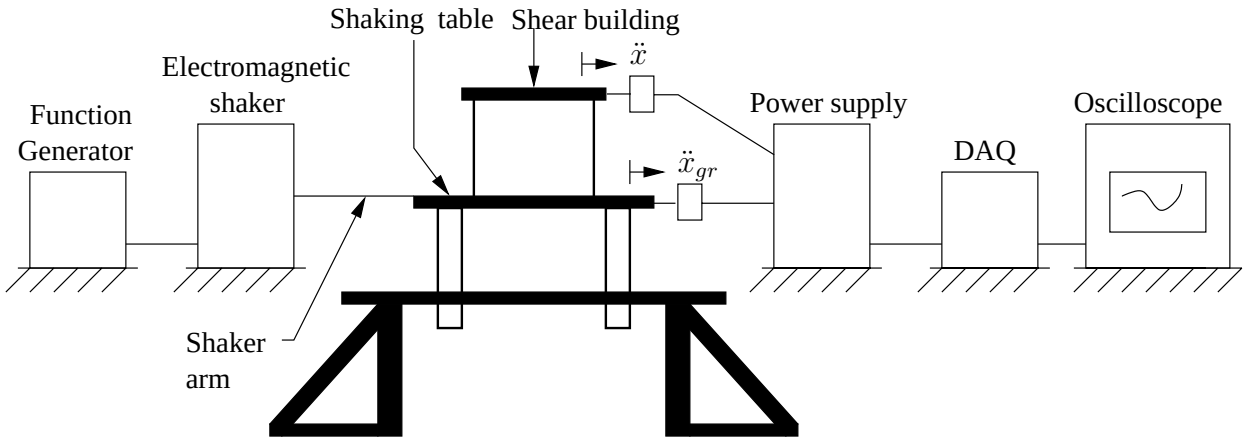


Figure 1: Laboratory setup of shaketable experiment

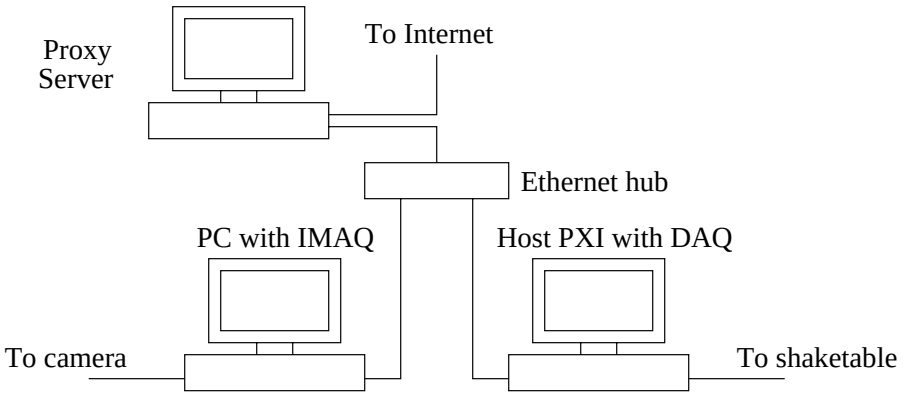


Figure 2: Network setup of the computers

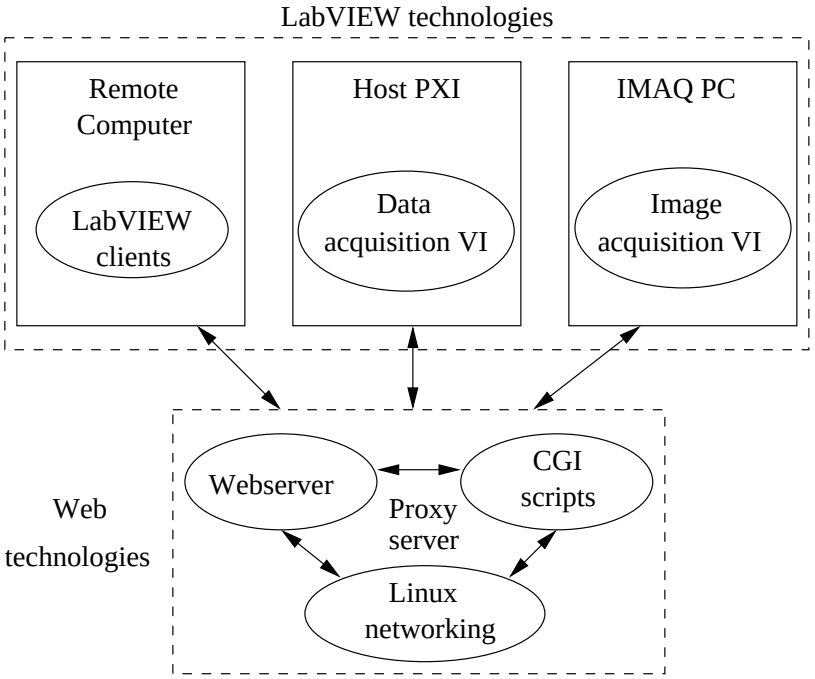


Figure 3: Software architecture

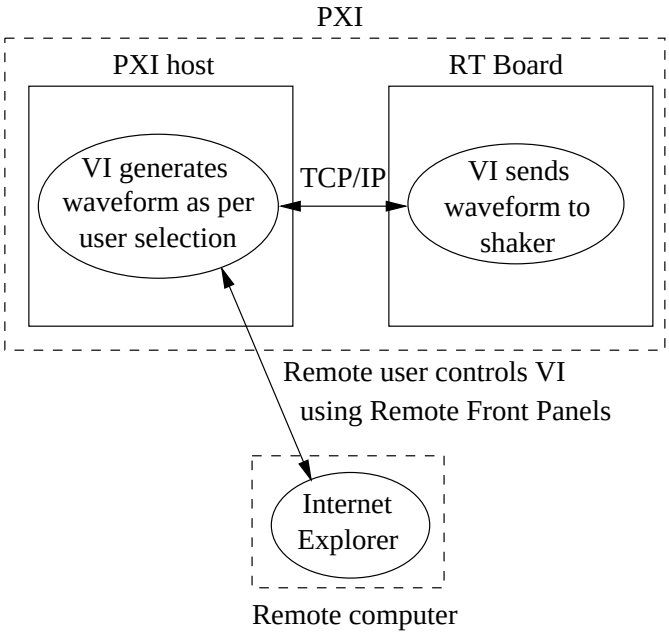


Figure 4: Wirgau et al.'s framework for shaketable control

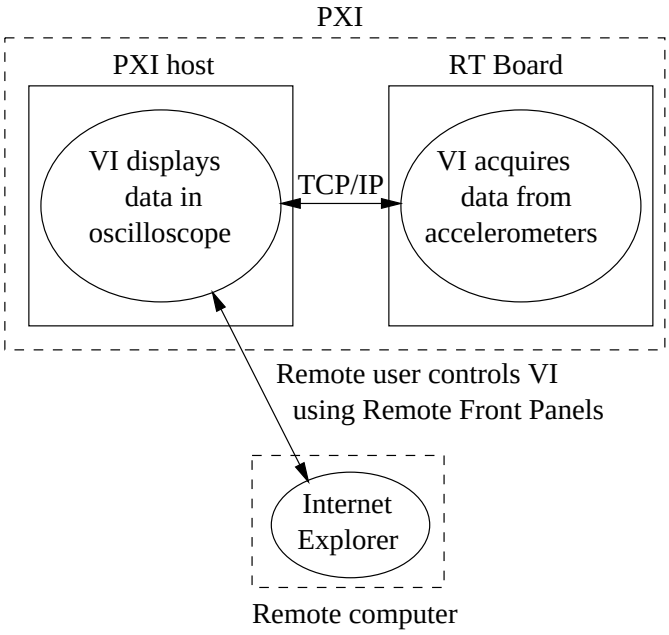


Figure 5: Wirgau et al.'s framework for data acquisition

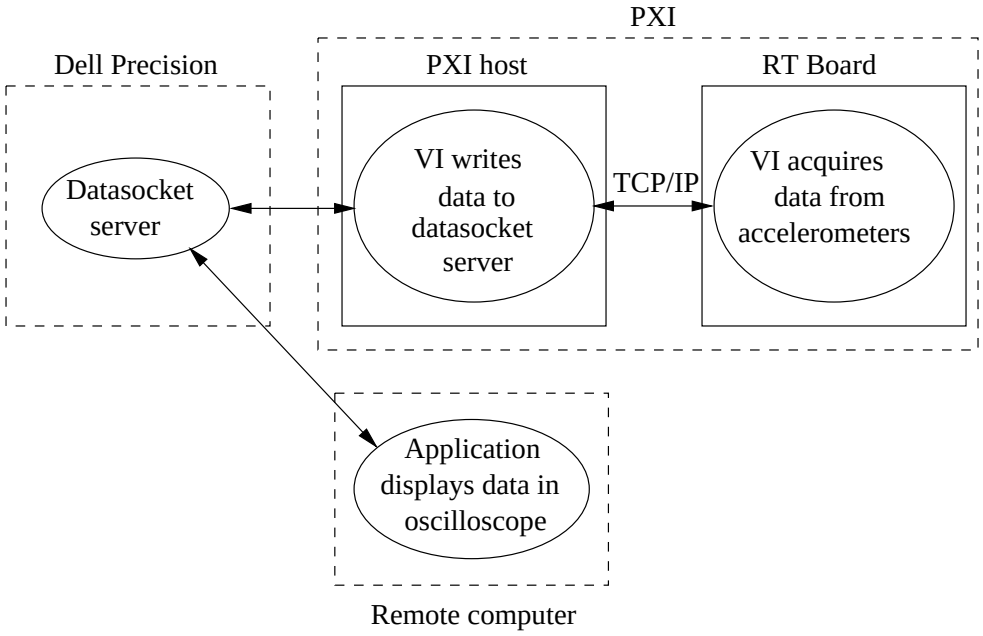


Figure 6: LabVIEW components for data acquisition

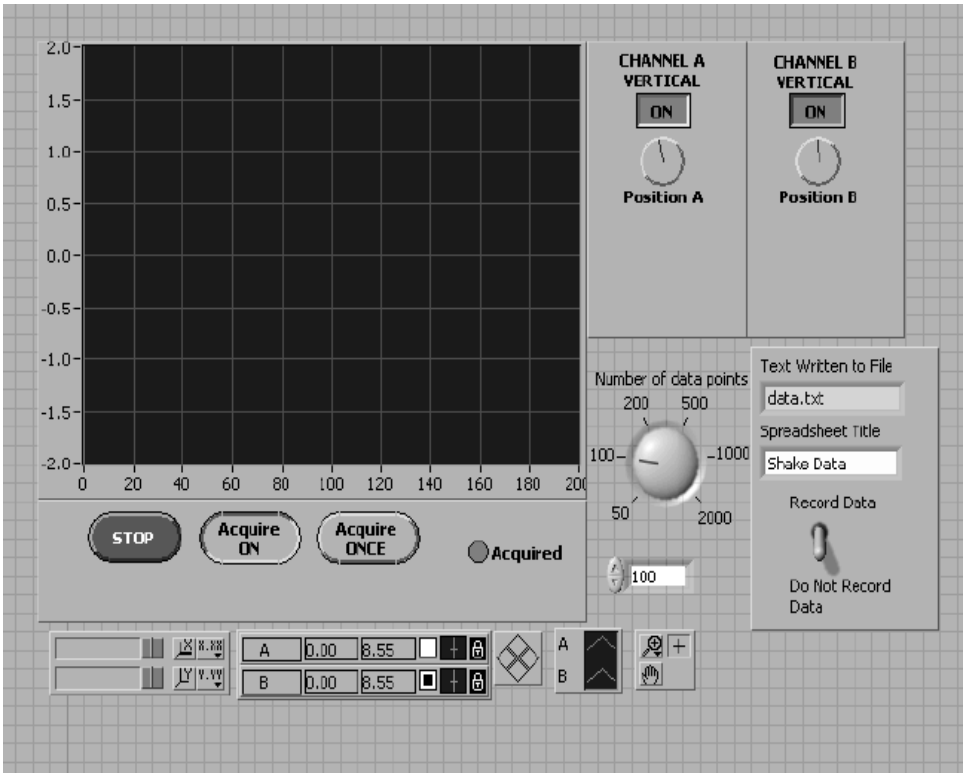


Figure 7: Front panel of remote user's data display VI

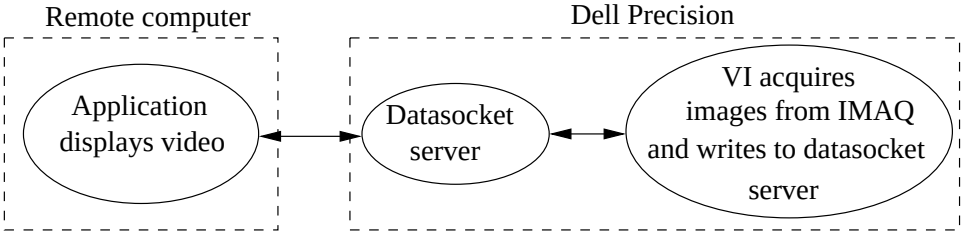


Figure 8: LabVIEW components for video acquisition

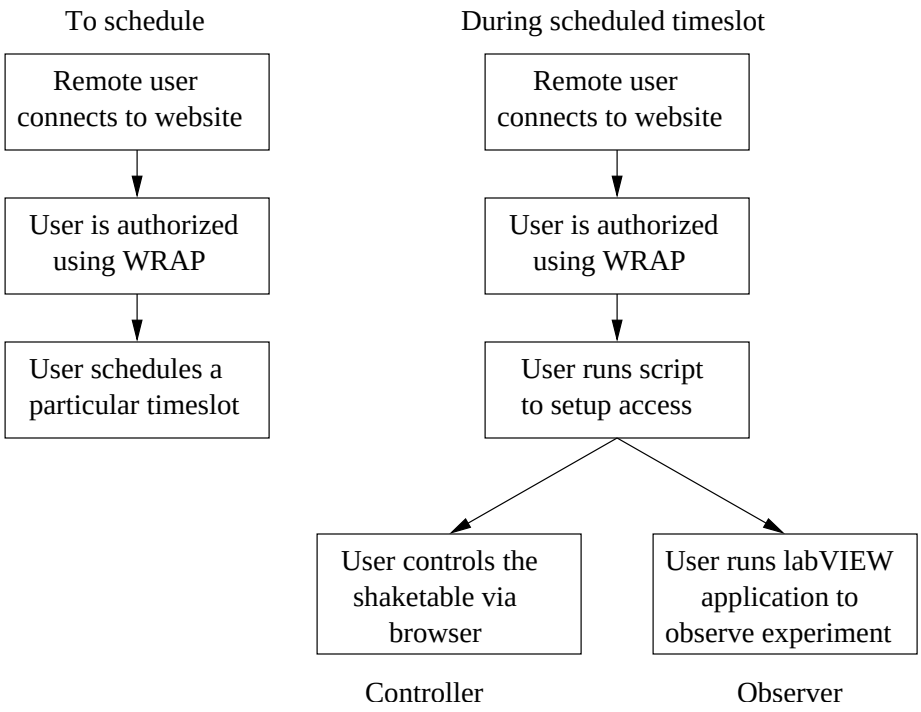


Figure 9: Flowchart showing remote user actions to access the experiment

Table lists the users scheduled to control the experiment for the next ten days

Timeslots	11/29/2005 (1)	11/30/2005 (2)	12/1/2005 (3)	12/2/2005 (4)	12/3/2005 (5)	12/4/2005 (6)	12/4/2005 (7)	12/5/2005 (8)	12/6/2005 (9)	12/7/2005 (10)
00:00-02:30 (1)	0	0	0	0	0	0	0	0	0	0
03:00-05:30 (2)	0	0	0	0	0	0	0	0	0	0
06:00-08:30 (3)	0	0	0	0	0	0	0	0	0	0
09:00-11:30 (4)	0	0	0	0	0	0	0	0	0	0
12:00-14:30 (5)	0	0	0	0	0	0	0	0	0	0
15:00-17:30 (6)	0	0	0	0	0	0	0	0	0	0
18:00-20:30 (7)	0	0	0	0	0	0	0	0	0	0
21:00-23:30 (8)	0	0	0	0	0	0	0	0	0	0

To reserve a particular timeslot,
To specify the date, enter the value in parantheses beside the date:
Enter timeslot number, given in parantheses along with the times:
SUBMIT

PLEASE CLICK ONLY ONCE AND THEN WAIT FOR THE TABLE TO GET UPDATED

Figure 10: Snapshot of control timetable

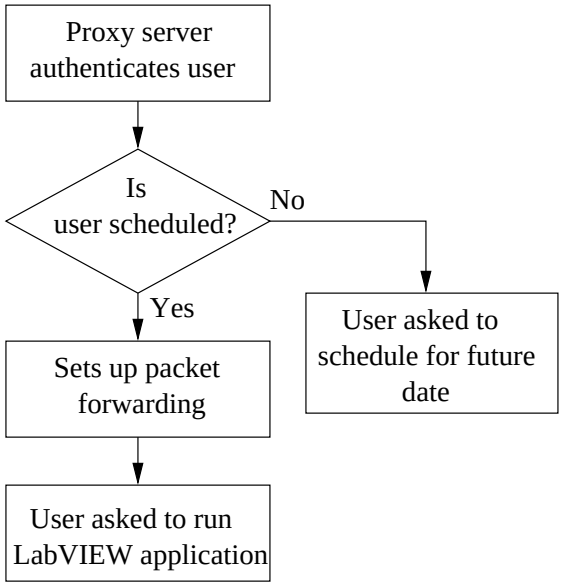


Figure 11: Flowchart showing proxy server actions for enabling remote access

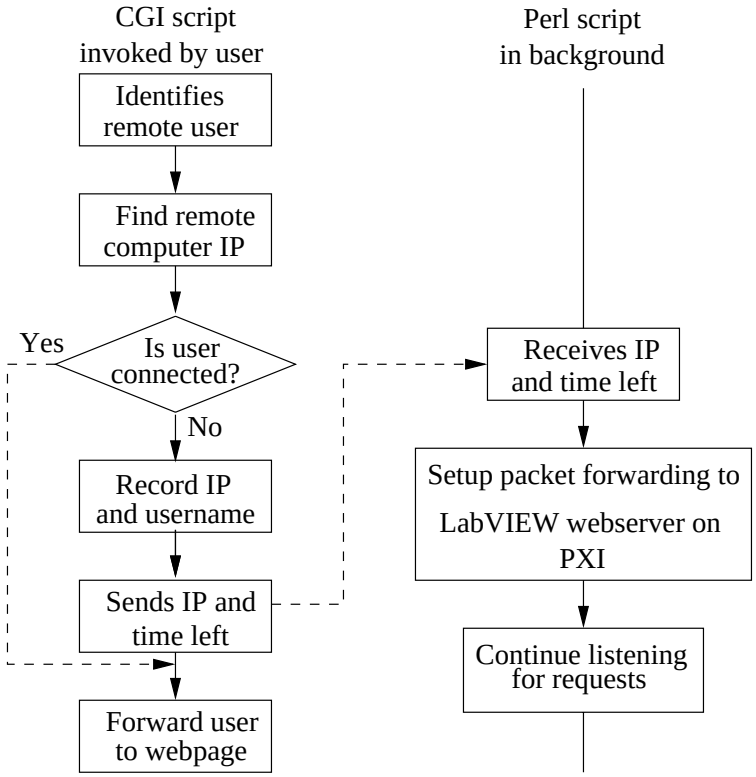


Figure 12: Sequence diagram of actions in giving remote user access to control the experiment

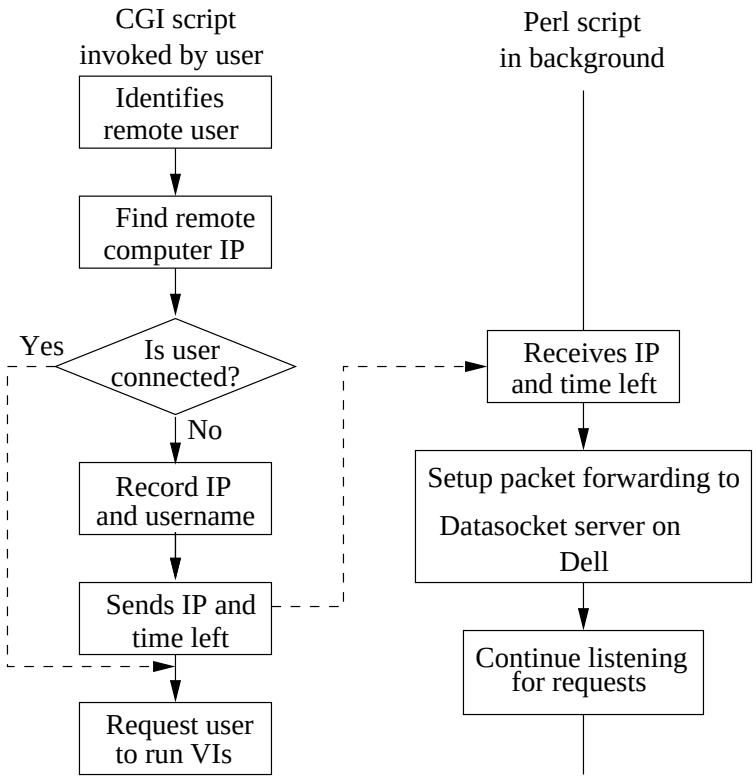


Figure 13: Sequence diagram of actions in giving remote user access to monitor the experiment

Part V

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

1 Summary and Conclusions

This research is focused on developing computational approaches for implementation in structural engineering Decision Support Systems (DSS). The study proposes approaches in two primary areas of decision-making - (1) simulation-based design and (2) experimental validation. These approaches are presented in the form of manuscripts, which are planned for submission to peer-reviewed journals for publication.

1.1 Design of Bridge Trusses

A real-life cost function is considered that has costs for the material and the number of products used in the design. A hybrid optimization approach is proposed that combines an algorithm based on engineering knowledge with a local search. The superior performance of the approach is demonstrated for real-life truss problems and minimum-weight truss problems. The following conclusions are obtained from this study.

- The clustering technique mimics the professional structural engineering knowledge to handle the cost associated with the number of products. The number of clusters generated using this technique is governed by the size of the grids in the rectangular area bounded by the maximum forces.
- The first algorithm in the hybrid approach generates a near-optimal solution. This algorithm contributes significantly to the overall computational efficiency of the proposed approach.
- The proposed hybrid sizing approach consistently generates good results for real-world cost functions.
- The sizing algorithm performs better than existing optimization techniques with respect to minimum-weight truss problems.
- The power of the sizing approach is also demonstrated in the geometry optimization algorithm. While the algorithm is an extension of simple local search, it generates much better results than existing optimization techniques.

1.2 Design of Moment-Resisting Steel Frames

A GA-based optimization approach is presented for the optimization of moment-resisting steel frames with rigid connections. The optimization approach uses a unique crossover scheme to perform trade-off study between the total cost and the number of rigid connections in the frame. “Modeling to Generate Alternatives - MGA” with “Hamming distance” as the difference metric is proposed for finding good design alternatives. We arrived at the following conclusions from this study.

- The trade-off study shows that the total cost of the frame is optimal when the rigid connections are placed at only a few locations in the frame.
- When all the supports of the frame are fixed, there exists a trade-off between the number of rigid connections and the total cost.
- When all the supports in the frame are hinged, the total cost is proportional to the number of rigid connections in the frame.
- MGA using “hamming distance” generates design alternatives that can prove valuable in arriving at a final design.

1.3 Remote Operation of Laboratory Experiments

A generic computational framework for remotely observing and controlling laboratory experiments is proposed. The framework supports a single-controller multiple-observer architecture in which multiple users simultaneously download and observe the measurements and video from the experiment and only one user is given control of the experiment. The main conclusions from this study are given below.

- LabVIEW Remote Panels technology is an efficient solution if only a single user needs to remotely access the laboratory experiment. Such an implementation is suitable for the controller in a single-controller multiple-observer framework.
- LabVIEW Datasockets technology enables simultaneously communicating data to users distributed across the internet. This technology is ideal for communicating sensor and video data to remote observers in a single-controller multiple-observer framework.
- The JPEG compression technique is relatively better than the intensity graph method for

video transmission. It provides a suitable compromise between the size of the communicated data and the image quality received at the remote user.

- A webserver with suitable user-authentication protocol, CGI scripts and linux networking concepts are sufficient to create a secure web-based environment to control access to laboratory experiments.
- The web technologies and LabVIEW VIs are decoupled to a large extent. The dependencies between the two components are primarily with respect to the TCP/IP listening ports of the LabVIEW webserver and the datasocket server.
- The computational framework is fairly generic and easily extensible to other laboratory experiments. The major task in adapting the framework to a new experiment is the development of new LabVIEW VIs that are necessary for control and data acquisition with respect to the new experiment.

2 Recommendations for Future Research

The following suggestions are provided for future research on the different topics considered in this research.

2.1 Design of Bridge Trusses

- The cost on the number of products used in the truss is taken from the westpoint bridge design software. The truss model can be improved by obtaining the actual values for this cost from practicing engineers.
- Extending the algorithms developed for geometry and sizing optimization to truss topology optimization is the natural follow-up to this study. This may require development of new algorithms that wrap around the algorithms developed for geometry and sizing optimization.

2.2 Design of Moment-Resisting Steel Frames

- In this study, the cost of a rigid connection is specified as a fixed value. In practice, the cost is dependent upon factors like the weld size and doubler-plates. Incorporating this

knowledge into the formulation can improve the quality of the results from optimization.

- Since space frame (three-dimensional) structures may also benefit from similar algorithms, it is useful to extend this study on plane frames to include space frame structures.
- The cost of the optimal solution is dependent on the type of supports at the base of the frame. If the cost of the footings is considered along with the material cost of the frame and its connection cost, optimization can provide a better estimate of the overall least-cost solution.

2.3 Remote Operation of Laboratory Experiments

- In this study, the proposed framework is tested primarily over the high-speed networks within the university. Testing its performance over a broad range of networks is essential before putting into practice.
- The system currently uses ASCII files to store the timetables with the controllers and observers. Using databases to store this information may result in a more robust architecture.
- Shaketable experiments involve both free and forced vibration. Currently, the framework permits only forced vibration tests. Studying the free vibration of a structure may require new hardware and a modification of the proposed framework.